Instruction Set Reference Manual

Allen-Bradley

MicroLogix 1400 Programmable Controllers

Bulletin 1766

Allen-Bradley • Rockwell Software

Rockwell Automation

Important User Information

Solid-state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://www.rockwellautomation.com/literature/) describes some important differences between solid-state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid-state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

	**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
	**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
	**SHOCK HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
	**BURN HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

To help you locate new and updated information in this release of the manual, we have included change bars as shown to the right of this paragraph.

The table below lists the sections that document new features and additional or updated information about existing features.

Summary of Change

Topic	Page
Password Protection for Series B	30

Firmware Revision History

Features are added to the controllers through firmware upgrades. See the latest release notes, 1766-RN001, to be sure that your controller's firmware is at the level you need. Firmware upgrades are not required, except to allow you access to the new features. See "Firmware Upgrades" below.

Firmware Upgrades

Enhanced features are added to the controllers through a firmware upgrade. This firmware upgrade is not required, except to allow you access to the latest features and corrected anomalies.

The following table illustrates which firmware revision is compatible with the various product series available.

Product Revision Compatibility

Processor Type Series	Controller Series	Firmware Revision	Software Version
A	A	7 or earlier	8.10.00 or later
B	B	11 or later'	8.30.00 or later
B	C	21 or later	8.30.00 or later
B (Enhanced Password Security)	B, C	21 or later	11.00.00 or later

To upgrade the firmware for a MicroLogix controller visit the MicroLogix web site at http://www.rockwellautomation.com/global/support/firmware/overview.page.

For firmware upgrade

Table of Contents

Summary of Changes

Firmware Revision History. iii
Firmware Upgrades iii

Preface

Who Should Use this Manual xxix
Purpose of this Manual xxix
Common Techniques Used in this Manual xxix
Related Documentation xxx
Rockwell Automation Support xxx

Chapter 1

I/O Configuration

Embedded I/O. 2
MicroLogix 1400 Expansion I/O 2
Expansion I/O Modules 2
Addressing Expansion I/O Slots 3
MicroLogix 1400 Expansion I/O Memory Mapping 3
Discrete I/O Configuration 3
1762-IA8 ,1762-IQ8, and 1762-IQ8OW6 Input Image. 3
1762-IQ16 Input Image. 3
1762-IQ32T Input Image 4
1762-OX6I and 1762-IQ8OW6 Output Image. 4
1762-OA8, 1762-OB8, and 1762-OW8 Output Image 4
1762-OB16 and 1762-OW16 Output Image. 4
1762-OV32T, 1762-OB32T Output Image. 5
Analog I/O Configuration 6
1762-IF2OF2 Input Data File 6
1762-IF2OF2 Output Data File 7
1762-IF4 Input Data File. 7
1762-OF4 Input Data File. 8
1762-OF4 Output Data File. 9
Specialty I/O Configuration. 9
1762-IR4 RTD/Resistance Module Input Data File 9
1762-IT4 Thermocouple Module Input Data File 10
I/O Addressing 11
Addressing Details. 11
I/O Forcing 12
Input Forcing 12
Output Forcing 12
Input Filtering. 13
Analog Inputs. 13
Analog Input Filter and Update times. 14
Input Channel Filtering. 14
Converting Analog Data 15
Converting Analog Input Data 15
Analog Outputs. 15

Latching Inputs 15
Converting Analog Output Value to Actual Output Voltage . 15
Rising Edge Behavior - Example 1 17
Rising Edge Behavior - Example 2 17
Falling Edge Behavior – Example 1 18
Falling Edge Behavior – Example 2 18
Configure Expansion
I/O Using RSLogix 500/RSLogix Micro 18

Chapter 2

Controller Memory and File Types

Controller Memory. 21
File Structure. 21
User Memory 23
MicroLogix 1400 User Memory 24
Viewing Controller Memory Usage 25
Data Files 26
Protecting Data Files During Download. 26
Data File Download Protection 26
Setting Download File Protection 27
User Program Transfer Requirements 27
Static File Protection 28
Using Static File Protection with Data File Download Protection. 28
Setting Static File Protection. 29
Program Password Protection 30
Program Password Protection for Series B (Enhanced Password Security) 32
Clearing the Controller Memory 34
Allow Future Access Setting (OEM Lock) 34
Web View Disable 35
LCD Edit Disable 35

Chapter 3

Function Files

Overview 37
Real-Time Clock Function File. 38
Writing Data to the Real-Time Clock 38
Real-Time Clock Accuracy 39
RTC Battery Operation 40
RTA - Real Time Clock Adjust Instruction 40
Memory Module Information Function File 42
FT - Functionality Type 42
MP - Module Present. 42
WP - Write Protect 43
FO - Fault Override 43
LPC - Load Program Compare 43

LE - Load on Error 43
LA - Load Always 44
MB - Mode Behavior 44
Base Hardware Information Function File 44
Communications Status File 45
General Status Block of Communications Status File 45
Diagnostic Counter Block of Communications Status File . . . 47
Active Node Table Block of Communications Status File 59
Ethernet Communications Status File 60
General Status Block of Ethernet Communications Status File 60
Diagnostic Counter Block of Communications Status File . . . 64
Input/Output Status File 67

Chapter 4

Programming Instructions Overview

Instruction Set 69
Using the Instruction Descriptions 70
Addressing Modes 71
Immediate Addressing 71
Direct Addressing 71
Indirect Addressing 72
Example – Using Indirect Addressing to Duplicate Indexed Addressing 74
Indexed Addressing Example 74
Indirect Addressing Example 74

Chapter 5

Using the High-Speed Counter and Programmable Limit Switch

High-Speed Counter Overview 77
Programmable Limit Switch Overview 77
High-Speed Counter (HSC) Function File 78
High-Speed Counter Function File Sub-Elements Summary 80
HSC Function File Sub-Elements 81
Program File Number (PFN) 81
Error Code (ER) 81
Function Enabled (FE) 82
Auto Start (AS) 82
Error Detected (ED) 82
Counting Enabled (CE) 83
Set Parameters (SP) 83
User Interrupt Enable (UIE) 84
User Interrupt Executing (UIX) 84
User Interrupt Pending (UIP) 85
User Interrupt Lost (UIL) 85
Low Preset Mask (LPM) 85
Low Preset Interrupt (LPI) 86
Low Preset Reached (LPR) 86

High Preset Mask (HPM) 86
High Preset Interrupt (HPI) 87
High Preset Reached (HPR) 87
Underflow (UF) 87
Underflow Mask (UFM) 88
Underflow Interrupt (UFI) 88
Overflow (OF) 89
Overflow Mask (OFM) 89
Overflow Interrupt (OFI) 89
Count Direction (DIR) 90
Mode Done (MD) 90
Count Down (CD) 90
Count Up (CU) 90
HSC Mode (MOD) 91
HSC Mode 0 - Up Counter 93
HSC Mode 1 - Up Counter with External Reset and Hold . . . 93
HSC Mode 2 - Counter with External Direction 94
HSC Mode 3 - Counter with External Direction, Reset, and Hold 94
HSC Mode 4 - Two Input Counter (up and down) 95
HSC Mode 5 - Two Input Counter (up and down) with External Reset and Hold 95
Using the Quadrature Encoder 96
HSC Mode 6 - Quadrature Counter (phased inputs A and B) 96
HSC Mode 7 - Quadrature Counter (phased inputs A and B) With External Reset and Hold 97
HSC Mode 8 - Quadrature X4 Counter 97
HSC Mode 9 - Quadrature X4 Counter with External Reset and Hold 98
Accumulator (ACC) 99
High Preset (HIP) 99
Low Preset (LOP) 99
Overflow (OVF) 100
Underflow (UNF) 100
Output Mask Bits (OMB) 101
High Preset Output (HPO) 101
Low Preset Output (LPO) 102
HSL - High-Speed Counter Load 103
RAC - Reset Accumulated Value 104
Programmable Limit Switch (PLS) File 105
PLS Data File 105
PLS Operation 105
Addressing PLS Files 106
PLS Example 107
Setting up the PLS File 107
Configuring the HSC for Use with the PLS 109

PLS Operation for This Example. 110

Chapter 6

Using High-Speed Outputs

PTO - Pulse Train Output 111
Pulse Train Output Function 111
Conditions Required to Start the PTO 113
Momentary Logic Enable Example 113
Standard Logic Enable Example 114
Pulse Train Outputs (PTOX) Function File. 115
Pulse Train Output Function File Sub-Elements Summary 116
PTOX Output (OUT) 117
PTOX Done (DN) 118
PTOX Decelerating Status (DS) 118
PTOX Run Status (RS) 118
PTOX Accelerating Status (AS) 119
PTOX Ramp Profile (RP) 119
PTOX Idle Status (IS) 119
PTOX Error Detected (ED) 120
PTOX Normal Operation Status (NS) 120
PTOX Enable Hard Stop (EH) 120
PTOX Enable Status (EN) 121
PTOX Output Frequency (OF) 121
PTOX Operating Frequency Status (OFS) 121
PTOX Total Output Pulses To Be Generated (TOP) 122
PTOX Output Pulses Produced (OPP) 122
PTOX Accel/Decel Pulses Independent (ADI) 122
PTOX Accel / Decel Pulses (ADP) (ADI=0) or File:Elem (ADI=1) 123
PTOX Controlled Stop (CS) 124
PTOX Jog Frequency (JF) 125
PTOX Jog Pulse (JP) 125
PTOX Jog Continuous (JC) 126
PTOX Jog Continuous Status (JCS) 126
PTOX Error Code (ER) 127
PWM – Pulse Width Modulation 128
PWM Function 128
Pulse Width Modulation (PWMX) Function File 129
Pulse Width Modulated Function File Elements Summary 130
PWMX Output (OUT) 131
PWMX Decelerating Status (DS) 131
PWMX Run Status (RS) 131
PWMX Accelerating Status (AS) 132
PWMX Profile Parameter Select (PP) 132
PWMX Idle Status (IS) 132
PWMX Error Detected (ED) 133

PWMX Normal Operation (NS) 133
PWMX Enable Hard Stop (EH) 133
PWMX Enable Status (ES) 133
PWMX Output Frequency (OF) 134
PWMX Operating Frequency Status (OFS) 134
PWMX Duty Cycle (DC) 134
PWMX Duty Cycle Status (DCS) 135
PWMX Accel/Decel Delay (ADD) 135
PWMX Error Code (ER) 135

Chapter 7

Relay-Type (Bit) Instructions

XIC - Examine if Closed
XIO - Examine if Open 137
OTE - Output Energize 139
OTL - Output Latch
OTU - Output Unlatch 140
ONS - One Shot 141
OSR - One Shot Rising
OSF - One Shot Falling 142

Chapter 8

Timer and Counter Instructions

Timer Instructions Overview 145
Timer Accuracy 146
Repeating Timer Instructions 147
TON - Timer, On-Delay 148
TOF - Timer, Off-Delay 149
RTO - Retentive Timer, On-Delay 150
How Counters Work 151
Using the CTU and CTD Instructions 151
Using Counter File Control and Status Bits 152
CTU - Count Up
CTD - Count Down 153
RES - Reset 154

Chapter 9

Compare Instructions

Using the Compare Instructions 155
EQU - Equal
NEQ - Not Equal 156
GRT - Greater Than
LES - Less Than 157
GEQ - Greater Than or Equal To
LEQ - Less Than or Equal To 157
MEQ - Mask Compare for Equal 158
LIM - Limit Test 159

Math Instructions

Chapter 10

General Information 161
Instructions 161
Using the Math Instructions 162
Updates to Math Status Bits 163
Overflow Trap Bit, S:5/0 164
Using the Floating Point (F) Data File 164
File Description 164
Definitions 165
Floating Point Exception Values 165
LSB Round-to-Even Rule 165
Addressing Floating Point Files 166
Programming Floating Point Values 166
ADD - Add
SUB - Subtract 167
MUL - Multiply
DIV - Divide 168
NEG - Negate 168
CLR - Clear 168
ABS - Absolute Value 169
SCL - Scale 170
SCP - Scale with Parameters 171
Special Considerations when Using Floating Point Parameters . . 172
Other Considerations 172
SQR - Square Root 173
SIN - Sine 173
Instruction Operation 174
MATH FLAGS EFFECTS 175
COS - Cosine 175
Instruction Operation 176
MATH FLAGS EFFECTS 177
TAN - Tangent 177
Instruction Operation 178
MATH FLAGS EFFECTS 179
ASN - Arc Sine 179
Instruction Operation 180
ACS - Arc Cosine 181
Instruction Operation 182
ATN - Arc Tangent 183
Instruction Operation 184
DEG - Radians to Degrees 185
Instruction Operation 186
RAD - Degrees to Radians 187
Instruction Operation 188
LN - Natural Log 189
Instruction Operation 190

Instruction Operation. 192
XPY - X Power Y 193
Instruction Operation. 194
CPT - Compute 196
Instruction Operation. 197
MATH FLAGS EFFECTS 197

Chapter 11

Application Specific Instructions

RHC - Read High Speed Clock. 199
Instruction Operation. 200
RPC - Read Program Checksum 201
Instruction Operation. 202
TDF - Compute Time Difference. 202
Instruction Operation. 203

Chapter 12

Conversion Instructions

Using Decode and Encode Instructions 205
DCD - Decode 4 to 1-of-16 206
ENC - Encode
1-of-16 to 4 206
Updates to Math Status Bits 207
FRD - Convert from Binary Coded Decimal (BCD) 208
FRD Instruction Source Operand. 208
Updates to Math Status Bits 209
Example. 209
TOD - Convert to Binary Coded Decimal (BCD) 211
TOD Instruction Destination Operand 211
Updates to Math Status Bits 212
Changes to the Math Register. 212
Example. 212
GCD - Gray Code 213
Updates to Math Status Bits 213

Chapter 13

Logical Instructions

Using Logical Instructions 215
Updates to Math Status Bits. 216
AND - Bit-Wise AND 216
OR - Logical OR 217
XOR - Exclusive OR 217
NOT - Logical NOT 218

Chapter 14

Move Instructions

MOV - Move 219
Using the MOV Instruction 219

Updates to Math Status Bits 220
MVM - Masked Move 221
Using the MVM Instruction 221
Updates to Math Status Bits 223

Chapter 15

File Instructions

CPW - Copy Word 225
COP - Copy File 226
FLL - Fill File 228
BSL - Bit Shift Left 229
BSR - Bit Shift Right 231
FFL - First In, First Out (FIFO) Load 233
FFU - First In, First Out (FIFO) Unload 235
LFL - Last In, First Out (LIFO) Load 237
LFU - Last In, First Out (LIFO) Unload 239
SWP - Swap 241

Chapter 16

Sequencer Instructions

SQC- Sequencer Compare 243
SQO- Sequencer Output 246
SQL - Sequencer Load 249

Chapter 17

Program Control Instructions

JMP - Jump to Label 253
LBL - Label 254
JSR - Jump to Subroutine 254
SBR - Subroutine Label 254
RET - Return from Subroutine 255
SUS - Suspend 255
TND - Temporary End 255
END - Program End 256
MCR - Master Control Reset 256

Chapter 18

Input and Output Instructions

IIM - Immediate Input with Mask 259
IOM - Immediate Output with Mask 260
REF- I/O Refresh 261

Chapter 19

Using Interrupts

Information About Using Interrupts 263
What is an Interrupt? 263
When Can the Controller Operation be Interrupted? 264
Priority of User Interrupts 265

User Fault Routine 265
Status File Data Saved 266
Creating a User Fault Subroutine 266
Controller Operation 266
User Interrupt Instructions 267
INT - Interrupt Subroutine 267
STS - Selectable Timed Start 267
UID - User Interrupt Disable 268
UIE - User Interrupt Enable 269
UIF - User Interrupt Flush 270
Using the Selectable Timed Interrupt (STI) Function File 272
Selectable Time Interrupt (STI) Function File Sub-Elements Summary 273
STI Function File Sub-Elements 273
STI Program File Number (PFN) 273
STI Error Code (ER) 273
STI User Interrupt Executing (UIX) 274
STI User Interrupt Enable (UIE) 274
STI User Interrupt Lost (UIL) 274
STI User Interrupt Pending (UIP) 274
STI Timed Interrupt Enabled (TIE) 275
STI Auto Start (AS) 275
STI Error Detected (ED) 275
STI Set Point Milliseconds Between Interrupts (SPM) 275
Using the Event Input Interrupt (EII) Function File 276
Event Input Interrupt (EII) Function File Sub-Elements Summary 276
EII Function File Sub-Elements 277
EII Program File Number (PFN) 277
EII Error Code (ER) 277
EII User Interrupt Executing (UIX) 278
EII User Interrupt Enable (UIE) 278
EII User Interrupt Lost (UIL) 278
EII User Interrupt Pending (UIP) 278
EII Event Interrupt Enable (EIE) 279
EII Auto Start (AS) 279
EII Error Detected (ED) 279
EII Edge Select (ES) 279
EII Input Select (IS) 280

Chapter 20

Process Control Instruction

The PID Concept 281
The PID Equation 282
PD Data File 282
PID - Proportional Integral Derivative 283

Input Parameters 284
Setpoint (SPS) 284
Process Variable (PV) 284
Setpoint MAX (MAXS) 285
Setpoint MIN (MINS) 285
Old Setpoint Value (OSP) 285
Output Limit (OL) 285
Control Variable High Limit (CVH) 286
Control Variable Low Limit (CVL) 286
Output Parameters 286
Control Variable (CV) 287
Control Variable Percent (CVP) 287
Scaled Process Variable (SPV) 288
Tuning Parameters 288
Controller Gain (Kc) 289
Reset Term (Ti) 289
Rate Term (Td) 290
Time Mode (TM) 291
Loop Update Time (LUT) 291
Zero Crossing Deadband (ZCD) 292
Feed Forward Bias (FF) 292
Scaled Error (SE) 292
Automatic / Manual (AM) 292
Control Mode (CM) 293
PV in Deadband (DB) 293
PLC 5 Gain Range (RG) 293
Setpoint Scaling (SC) 294
Loop Update Too Fast (TF) 294
Derivative Action Bit (DA) 294
CV Upper Limit Alarm (UL) 294
CV Lower Limit Alarm (LL) 295
Setpoint Out Of Range (SP) 295
PV Out Of Range (PV) 295
Done (DN) 295
PD10:0.19Enable (EN) 295
Integral Sum (IS) 296
Altered Derivative Term (AD) 296
Runtime Errors 297
Analog I/O Scaling 298
Application Notes 299
Input/Output Ranges 299
Scaling to Engineering Units 300
Zero-Crossing Deadband DB 301
Output Alarms 301
Output Limiting with Anti-Reset Windup 302
The Manual Mode 302

PID Rung State 302
Feed Forward or Bias 303
Application Examples 303
PID Tuning 303
Procedure 303
Verifying the Scaling of Your Continuous System 305
Determining the Initial Loop Update Time 306

ASCII Instructions

Chapter 21

General Information 309
ASCII Instructions 309
Instruction Types and Operation 310
ASCII String Control 310
ASCII Port Control 310
Programming ASCII Instructions 311
Protocol Overview 311
Using the Full ASCII Instruction Set 311
Using AWA and AWT Instructions with Other Serial Channel Drivers 311
String (ST) Data File 312
File Description 312
Addressing String Files 312
Control Data File 313
File Description 313
Addressing Control Files 314
ACL - ASCII Clear Buffers 314
Entering Parameters 315
Instruction Operation 316
AIC - ASCII Integer to String 316
AWA - ASCII Write with Append 316
Programming AWA Instructions 317
Entering Parameters 317
Example 318
AWT - ASCII Write 319
Programming AWT Instructions 319
Entering Parameters 319
Example 320
ABL - Test Buffer for Line 321
Entering Parameters 321
Instruction Operation 322
ACB - Number of Characters in Buffer 322
Entering Parameters 322
Instruction Operation 323
ACI - String to Integer 323
Entering Parameters 324

Instruction Operation 324
ACN - String Concatenate 325
Entering Parameters 325
Instruction Operation 325
AEX - String Extract 326
Entering Parameters 326
Instruction Operation 326
AHL - ASCII Handshake Lines 327
Entering Parameters 327
Instruction Operation 328
ARD - ASCII Read Characters 328
Entering Parameters 329
Instruction Operation 329
ARL - ASCII Read Line 330
Entering Parameters 330
Instruction Operation 331
ASC - String Search 331
Entering Parameters 331
Example 332
Error Conditions 332
ASR - ASCII String Compare 333
Entering Parameters 333
Instruction Operation 333
Timing Diagram for ARD, ARL, AWA, and AWT Instructions 334
Using In-line Indirection 334
Examples 334
ASCII Instruction Error Codes 335
ASCII Character Set 336

Chapter 22

Communications Instructions

Messaging Overview 337
SVC - Service Communications 339
Channel Select 339
Communication Status Bits 340
Application Example 340
MSG - Message 341
The Message Element 342
Message File Sub-Elements 342
“Control Bits” Parameters 346
Ignore if Timed Out (TO) 346
Enable (EN) 347
Enabled and Waiting (EW) 347
Error (ER) 347
Done (DN) 348
Start (ST) 348

UnConnected(UC) 348
Break Connection (BK)) 348
Timing Diagram for the MSG Instruction 348
Communication Servicing Selection and Message Servicing Selection 352
Communication Servicing Selection 352
Message Servicing Selection 353
MSG Instruction Ladder Logic 353
Enabling the MSG Instruction for Continuous Operation 353
Enabling the MSG Instruction Via User Supplied Input 354
Local Messages 355
Local Networks 355
Example 1 - Local DH-485 Network 356
Example 2 - Local DeviceNet Network with DeviceNet Interface (1761-NET-DNI) 356
Example 3 - Local DF1 Half-Duplex Network 357
Configuring a Local Message 357
Message Setup Screen 357
“This Controller” Parameters 358
Channel 358
Communication Command 359
Modbus Command 360
Data Table Address 360
Size in Elements 361
“Target Device” Parameters 363
Message Timeout 363
Data Table Address/Offset 363
Modbus - MB Data Address (1...65536) 364
Local/Slave Node Address 365
Local/Remote 365
Local Messaging Examples 365
Example 1 - Local Read from a 500CPU 367
Message Instruction Setup 367
Valid File Type Combinations 367
Example 2 - Local Read from a 485CIF 368
Message Instruction Setup 368
Valid File Type Combinations 368
Example 3 - Local Read from a PLC-5 369
Message Instruction Setup 369
Valid File Type Combinations 369
Example 4 - Configuring a Modbus Message for Channel 0 or Channel 2 370
Message Setup Screen 370
“This Controller” Parameters 372
Target Device 372
Example 5 - Configuring an Ethernet/IP Message 373

Message Setup Screen 373
"This Controller" Parameters 374
"Target Device" Parameters 375
Example 6 - Configuring Local Write message with ST file 378
Remote Messages 379
Remote Networks 379
DH-485 and DH+ Networks 379
DeviceNet and Ethernet Networks 381
Configuring a Remote Message 382
Example Configuration Screen and Network 382
"This Controller" Parameters 383
"Control Bits" Parameters 383
"Target Device" Parameters 383
Message Timeout 383
Data Table Address 383
Local Bridge Address 383
Remote Bridge Address 384
Remote Station Address 384
Remote Bridge Link ID 384
Network Link ID 384
Configuring a Multi-hop Remote Message on EtherNet/IP Communication Channel 385
Network Message Example 1: 385
MicroLogix 1400 Ethernet to SLC5/04 DH+ via ENET & DHRIO 385
DHRIO Routing table creation 386
Network Message Example 2: 391
MicroLogix 1400 Ethernet to SLC 5/03 DH485 via ENET, DHRIO and 1785-KA5 bridge device 391
Adding 1785-KA5 bridge module 392
DHRIO Routing table creation 392
ML1400 Channel1 Configuration 393
Network Message Example 3: 395
MicroLogix 1400 Unsolicited Write Message to RSLinx via Ethernet 395
Configuring a MicroLogix 1400 CIP Generic Message via Ethernet . 400
"This Controller" Parameters 401
"Target Device" Parameters 401
Service Type and Service Code 402
CIP Generic Error Codes/Internal Fail Codes 404
MSG Instruction Error Codes 404
Special Function with MSG instruction 407
Ethernet Channel Configuration Change Functionality 407
Set up MSG for Changing Channel Configuration 408
Considerations for Changing Ethernet Channel Configuration. . 410

Considerations for Changing SMTP Configuration 410
Email Functionality 412
Setup SMTP Configuration File 412
Configure MSG Setup Screen to send SMTP message 416
SMTP Error Codes/Internal Fail Codes 418
Inline Indirection in String File for Subject and Body 418
SMTP Authentication Encoding 418
Sending email in User Fault Routine 419

Chapter 23

Modbus TCP

Modbus TCP Architecture 421
Channel Configuration for Modbus TCP 421
Modbus TCP Server Configuration 424
Modbus TCP Server Configuration Parameters 424
Modbus TCP Enable 424
Modbus Data Table File Numbers 425
Expanded for Holding Registers 425
Enable Access Control for IP Addresses 425
Client IP Address0 ..Client IP Address4 425
Local TCP Port Number 425
Diagnostic File Number 426
Modbus TCP Client Configuration 426
Modbus TCP Client Configuration Parameters 427
Modbus TCP Enable 427
Diagnostic File Number 427
Messaging for Modbus TCP Client 427
MSG Configuration Parameters 427
Message Instruction Timeouts 429
Change between Executing and Non-Executing Controller Modes... 429
Diagnostics for Modbus TCP 430
Range of Error codes for Modbus TCP Server and TCP Client .. 432

Chapter 24

Socket Interface Using CIP Generic Messaging

Overview 435
Socket Interface Architecture 435
..... Number and Type of Sockets 435
..... Typical Sequence of Transactions For a TCP Client 436
..... Typical Sequence of Transactions For a TCP Server 437
..... Typical Sequence of Transactions For UDP Without OpenConnection 438
Typical Sequence of Transactions For UDP With OpenConnection. 439
Communicate With the Socket Object Via a MSG Instruction 441

Message Transfer Sizes 443
Service Timeouts 443
Message Instruction Timeouts 444
Socket Interface Timeouts 444
Programming Considerations 444
TCP Connection Loss 445
Change Controller Mode Between Executing and Non-Executing . . . 445
Application Messages and TCP 446
Partial Reads 446
Partial Writes 447
Socket Object Services 447
CreateSocket 448
MSG Configuration Parameters 449
OpenConnection 450
MSG Configuration Parameters 451
AcceptConnection 452
MSG Configuration Parameters 453
Read 455
MSG Configuration Parameters 455
Write 457
MSG Configuration Parameters 457
Inline Indirection functionality 459
DeleteSocket 459
MSG Configuration Parameters 460
DeleteAllSockets 461
MSG Configuration Parameters 462
Possible Error Codes for Socket Services 463

Recipe and Data Logging

Chapter 25

RCP - Recipe 465
Recipe File and Programming Example 467
Configuring the RCP file 467
Application Explanation of Operation 469
Calculation of Consumed Memory 470
Data Logging 471
Queues and Records 471
Example Queue 0 472
String Length of Record 472
Number of Records 473
Example Queue 5 473
String Length of Record 473
Number of Records 474
Example Maximum Record of String Data 475
String Length of Record 475

Number of Records 475
Configuring Data Log Queues 476
DLG - Data Log Instruction 478
Data Log Status File 479
Data Logging Enable (EN) 479
Data Logging Done (DN) 479
Data Logging Overflow (OV) 479
Data Logging ClearQueue (CQ) 479
File Size (FSZ) 480
Records Stored (RST) 480
Retrieving (Reading) Records 481
Accessing the Retrieval File 481
Retrieval Tools 481
Information for Creating Your Own Application 481
Controller Receives Communications Packet 481
Controller Responds with Reply 482
Conditions that Will Erase the Data Retrieval File 483

Chapter 26

LCD - LCD Information

LCD Overview 485
LCD Function File 486
LCD Function File Sub-Elements Summary 487
LCD Function File Sub-Elements 488
Customized Boot Logo ASCII File Address Offset (CBL) .. 488
Start with Customized Display (SCD) 488
Data Input Timeout of LCD instruction (TO) 488
LCD Instruction Job Done (DN) 489
LCD Display Operation Error Bit (ERR) 489
LCD Module Operation Error Number (ERN) 489
Target User Defined File Number (TUF) 490
Jog data update Mode set (JOG) 490
Trimpot 0 Data (TMIN – TMAX) (POT0),
Trimpot 1 Data (TMIN – TMAX) (POT1) 491
Instruction Display Window (WND) 491
OK key in Customized Display (OK) 491
ESC key in Customized Display (ESC) 492
LCD Backlight On/Off (BACKON) 492
LCD Backlight Time (BACKTIME) 492
LCD contrast (CNST) 492
LCD - LCD Instruction 493
Getting Value with Keypad 494
Displaying Special Characters 494

Appendix A

MicroLogix 1400 Memory Usage and Instruction Execution Time

Programming Instructions Memory usage and Execution Time 497
MicroLogix 1400 Indirect Addressing 501
MicroLogix 1400 Scan Time Calculation 501
Communication Channels Inactive 501
One Or More Communication Channels Active 501

Appendix B

System Status File

Status File Overview 503
Status File Details 504
Arithmetic Flags 504
Carry Flag 504
OverFlow Flag 504
Zero Flag 505
Sign Flag 505
Controller Mode 505
User Application Mode 505
Forces Enabled 506
Forces Installed 506
Fault Override At Power-Up 506
Startup Protection Fault 506
Load Memory Module On Error Or Default Program 507
Load Memory Module Always 507
Power-Up Mode Behavior 507
Major Error Halted 509
Future Access (OEM Lock) 509
First Scan Bit 510
STI Mode 510
STI Pending 510
STI Enabled 510
STI Executing 510
Memory Module Program Compare 511
Math Overflow Selection 511
Watchdog Scan Time 512
Free Running Clock 512
S:4 Free Running Clock Comparison for SLC 500 and MicroLogix Controllers 512
Minor Error Bits 513
Overflow Trap Bit 513
Control Register Error 513
Major Error Detected in User Fault Routine 514
Memory Module Boot 514
Memory Module Password Mismatch 514
STI Lost 514
Processor Battery Low 514

Input Filter Selection Modified 515
ASCII String Manipulation Error 515
Major Error Code 515
Suspend Code 515
Suspend File 516
Active Nodes (Nodes 0 to 15) 516
Active Nodes (Nodes 16 to 31) 516
Math Register 516
Node Address 517
Baud Rate 517
Maximum Scan Time 517
User Fault Routine File Number 517
STI Set Point 518
STI File Number 518
Channel 0 Communications 518
Incoming Command Pending 518
Message Reply Pending 518
Outgoing Message Command Pending 519
Communications Mode Selection 519
Communications Active 519
Scan Toggle Bit 519
Last 100 µSec Scan Time 519
Data File Overwrite Protection Lost 520
RTC Year 520
RTC Month 520
RTC Day of Month 520
RTC Hours 521
RTC Minutes 521
RTC Seconds 521
RTC Day of Week 522
OS Catalog Number 522
OS Series 522
OS FRN 522
Processor Catalog Number 522
Processor Series 523
Processor Revision 523
User Program Functionality Type 523
Compiler Revision - Build Number 523
Compiler Revision - Release 523

Appendix C

Fault Messages and Error Codes

Identifying Controller Faults 525
Automatically Clearing Faults 525
Manually Clearing Faults Using the Fault Routine 526
Fault Messages 526

Contacting Rockwell Automation for Assistance 532

Appendix D

Protocol Configuration

DH-485 Communication Protocol 536
DH-485 Network Description 536
DH-485 Token Rotation 536
DH-485 Broadcast Messages 536
DH-485 Configuration Parameters 537
Software Considerations 537
Number of Nodes 537
Setting Node Addresses 538
Setting Controller Baud Rate 538
Setting Maximum Node Address 538
MicroLogix 1400 Remote Packet Support 538
DF1 Full-Duplex Protocol 539
DF1 Full-Duplex Operation 539
DF1 Half-Duplex Protocol 540
DF1 Half-Duplex Master Driver Broadcast Messages 540
DF1 Half-Duplex Slave Driver Broadcast Messages 540
Choosing a Polling Mode for DF1 Half-Duplex Master 540
Message-Based Polling Mode 540
Standard Polling Mode 541
About Polled Report-by-Exception 543
About Slave-to-Slave Messaging 543
Addressing Tips 543
DF1 Half-Duplex Master Standard Polling Mode 543
DF1 Half-Duplex Master MSG-based Polling Mode Operation 544
DF1 Half-Duplex Master Channel Status 546
Monitor Active Stations 547
DF1 Half-Duplex Slave Configuration 548
DF1 Radio Modem Protocol 549
Using the DF1 Radio Modem 550
Using Store and Forward Capability 551
Configuring the Store and Forward Table 552
DF1 Radio Modem Channel Status 553
DF1 Radio Modem System Limitations 554
Modbus RTU Protocol 555
Modbus RTU Master 555
Modbus RTU Slave 556
Modbus RTU Master Configuration 556
Modbus RTU Slave Configuration 559
Modbus Slave Memory Map 561
Modbus Commands 562
Modbus Error Codes 564
ASCII Driver 566

Ethernet Driver 567
SNMP MIB II Data Groups 570

Appendix E

Knowledgebase Quick Starts

17444 Pulse Train Output (PTOX) Quick Start 577
Example 578
General Information on the PTO 579
17446 Pulse Width Modulation (PWMX) Quick Start 580
Example 581
17447 High Speed Counter (HSC) Quick Start 582
General Information 582
Getting Started 582
Example 583
Troubleshooting 585
17465 Message (MSG) Quick Start 586
Communications Specifications 586
Message (MSG) instruction 587
Continuous Message Example 587
Micrologix 1000 (Node 4) Ladder Logic 588
Verify Data Sending 589
17501 Selectable Timed Interrupt (STI) Quick Start 590
What is an Interrupt? 590
Selectable Timed Interrupt 590
Getting Started 590
Example 591
Some Guidelines on Using Interrupt bits 591
17503 Real Time Clock (RTC) Quick Start 592
General Information 592
Getting Started 592
17558 User Interrupt Disable (UID) Quick Start 593
Some Guidelines on Using Interrupt bits 594
18465 RTC Synchronization Between Controllers Quick Start 594
Minimum Hardware/Software Requirements 594
Example 595
18498 Data Logging (DLG) Quick Start 597
General Information 597
Configure the DLG Instruction in MicroLogix 1400 597
Use the Data Logging Utility 600
Frequently Asked Questions 602
Using the Datalog Utility 603
Establish Connections 603
Disconnect the Modem 604

How to Use 40kHz PTOX/PWMX of MicroLogix 1400 Series B Controller

Appendix F

Basic requirements to use 40KHz PTOX and PWMX in MicroLogix Controller 605
PTOX and PWMX function file changes in Series B Controller 605
RSLogix500 display issues 606
Instruction issues 607

Glossary G

Index

MicroLogix 1400 List of Instructions and Function Files

Read this preface to familiarize yourself with the rest of the manual. It provides information concerning:

- who should use this manual
- the purpose of this manual
- related documentation
- conventions used in this manual
- Rockwell Automation support

Who Should Use this Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use MicroLogix 1400 controller.

You should have a basic understanding of electrical circuitry and familiarity with relay logic. If you do not, obtain the proper training before using this product.

Purpose of this Manual

This manual is a reference guide for MicroLogix 1400 controller. It describes the procedures you use to program and troubleshoot your controller. This manual:

- gives you an overview of the file types used by the controllers
- provides the instruction set for the controllers
- contains application examples to show the instruction set in use

Common Techniques Used in this Manual

The following conventions are used throughout this manual:

- Bulleted lists such as this one provide information, not procedural steps.
- Numbered lists provide sequential steps or hierarchical information.
- Change bars appear beside information that has been changed or added since the last revision of this manual. Change bars appear in the margin as shown to the right of this paragraph.

Related Documentation

The following documents contain additional information concerning Rockwell Automation products. To obtain a copy, contact your local Rockwell Automation office or distributor.

For	Read this Document	Document Number
Information on mounting and wiring the MicroLogix 1400 Programmable Controller, including a mounting template and door labels.	MicroLogix 1400 Programmable Controllers Installation Instructions	1766-IN001
Detailed information on planning, mounting, wiring, and troubleshooting your MicroLogix 1400 system.	MicroLogix 1400 Programmable Controllers User Manual	1766-UM001
A description on how to install and connect an AIC+. This manual also contains information on network wiring.	Advanced Interface Converter (AIC+) User Manual	1761-UM004
Information on how to install, configure, and commission a DNI	DeviceNet Interface User Manual	1761-UM005
Information on DF1 open protocol.	DF1 Protocol and Command Set Reference Manual	1770-6.5.16
In-depth information on grounding and wiring Allen-Bradley programmable controllers	Allen-Bradley Programmable Controller Grounding and Wiring Guidelines	1770-4.1
A description of important differences between solid-state programmable controller products and hard-wired electromechanical devices	Application Considerations for Solid-State Controls	SGI-1.1
An article on wire sizes and types for grounding electrical equipment	National Electrical Code - Published by the National Fire Protection Association of Boston, MA.	
A glossary of industrial automation terms and abbreviations	Allen-Bradley Industrial Automation Glossary	AG-7.1

Rockwell Automation Support

Before you contact Rockwell Automation for technical assistance, we suggest you review the troubleshooting information contained in this publication first.

If the problem persists, call your local distributor or contact Rockwell Automation in one of the following ways:

Phone	United States/Canada	1.440.646.3434
	Outside United States/Canada	You can access the phone number for your country via the Internet: 1. Go to http://www.ab.com 2. Click on *Product Support* (http://support.automation.rockwell.com) 3. Under *Support Centers*, click on *Contact Information*
Internet	⇒	1.Go to http://www.ab.com 2.Click on *Product Support* (http://support.automation.rockwell.com)

I/O Configuration

This section discusses the various aspects of Input and Output features of the MicroLogix 1400 controllers. Each controller comes with a certain amount of embedded I/O, which is physically located on the controller. The controller also allows for adding expansion I/O.

This section discusses the following I/O functions:

- Embedded I/O on page 2
- MicroLogix 1400 Expansion I/O on page 2
- MicroLogix 1400 Expansion I/O Memory Mapping on page 3
- I/O Addressing on page 11
- I/O Forcing on page 12
- Input Filtering on page 13
- Latching Inputs on page 15

Embedded I/O

The MicroLogix 1400 provide discrete I/O and analog input that is built into the controller as listed in the following table. These I/O points are referred to as Embedded I/O.

MicroLogix 1400 Embedded I/O

Catalog Number	Description				
	Input Power	User Power	Embedded Discrete I/O	Embedded Analog I/O	Comm. Ports
1766-L32BWA	100/240V AC	24V DC	12 Fast 24V DC Inputs 8 Normal 24V DC Inputs 12 Relay Outputs	None	1 RS232/RS485[1] 1 Ethernet 1 RS232[2]
1766-L32AWA		None	20 120V AC Inputs 12 Relay Outputs		
1766-L32BXB	24V DC		12 Fast 24V DC Inputs 8 Normal 24V DC Inputs 6 Relay Outputs 3 Fast DC Outputs 3 Normal DC Outputs		
1766-L32BWAA	100/240V AC	24V DC	12 Fast 24V DC Inputs 8 Normal 24V DC Inputs 12 Relay Outputs	4 Voltage Inputs 2 Voltage Outputs	
1766-L32AWAA		None	20 120V AC Inputs 12 Relay Outputs		
1766-L32BXBA	24V DC		12 Fast 24V DC Inputs 8 Normal 24V DC Inputs 6 Relay Outputs 3 Fast DC Outputs 3 Normal DC Outputs		

(1) Isolated RS-232/RS-485 combo port. Same as ML1100 Comm 0

(2) Non-isolated RS-232. Standard D-sub connector

AC embedded inputs have fixed input filters. DC embedded inputs have configurable input filters for a number of special functions that can be used in your application. These are: high-speed counting, event input interrupts, and latching inputs. The 1766-L32BXB and 1766-L32BXBA have three high-speed outputs for use as pulse train output (PTO) and/or pulse width modulation (PWM) outputs.

MicroLogix 1400 Expansion I/O

If the application requires more I/O than the controller provides, you can attach I/O modules. These additional modules are called expansion I/O.

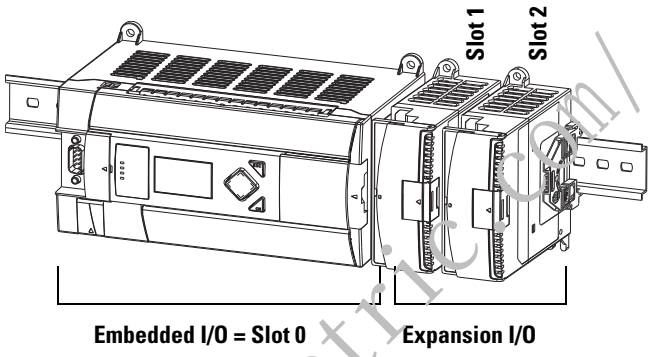
Expansion I/O Modules

For the MicroLogix 1400, Bulletin 1762 expansion I/O is used to provide discrete and analog inputs and outputs, and specialty modules. You can attach up to seven expansion I/O modules in any combination.

Addressing Expansion I/O Slots

The figure below shows the addressing for the MicroLogix 1400 and its I/O.

The expansion I/O is addressed as slots 1...7 (the controller's embedded I/O is addressed as slot 0). Modules are counted from left to right as shown below.



TIP In most cases, you can use the following address format:
X:s/b (X = file type letter, s = slot number, b = bit number)

See I/O Addressing on page 11 for complete information on address formats.

MicroLogix 1400 Expansion I/O Memory Mapping

Discrete I/O Configuration

1762-IA8 ,1762-IQ8, and 1762-IQ8OW6 Input Image

For each input module, the input data file contains the current state of the field input points. Bit positions 0...7 correspond to input terminals 0...7.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	x	x	x	x	x	x	x	x	r	r	r	r	r	r	r	r

r = read only, x = not used, always at a 0 or OFF state

1762-IQ16 Input Image

For each input module, the input data file contains the current state of the field input points. Bit positions 0...15 correspond to input terminals 0...15.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

r = read only

1762-IQ32T Input Image

For each input module, the input data file contains the current state of the field input points. Bit positions 0...15 together with word 0/1 correspond to input terminals 0...31.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
1	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

r = read only

1762-OX6I and 1762-IQ8OW6 Output Image

For each output module, the output data file contains the controller-directed state of the discrete output points. Bit positions 0...5 correspond to output terminals 0...5.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	r/w	r/w	r/w	r/w	r/w	r/w

r/w = read and write, 0 = always at a 0 or OFF state

1762-OA8, 1762-OB8, and 1762-OW8 Output Image

For each output module, the output data file contains the controller-directed state of the discrete output points. Bit positions 0...7 correspond to output terminals 0...7.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

r/w = read and write, 0 = always at a 0 or OFF state

1762-OB16 and 1762-OW16 Output Image

For each output module, the output data file contains the controller-directed state of the discrete output points. Bit positions 0...15 correspond to output terminals 0...15.

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

r/w = read and write

1762-OV32T, 1762-OB32T Output Image

For each output module, the output data file contains the controller-directed state of the discrete output points. Bit positions 0...15 together with word 0/1 correspond to output terminals 0...31.

Word	Bit Position															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
1	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

r/w = read and write

Analog I/O Configuration

The following table shows the data ranges for 0...10V dc and 4...20 mA.

Valid Input/Output Data Word Formats/Ranges

Normal Operating Range	Full Scale Range	Raw/Proportional Data	Scaled-for-PID
0…10V DC	10.5V DC	32,760	16,380
	0.0V DC	0	0
4…20 mA	21.0 mA	32,760	16,380
	20.0 mA	31,200	15,600
	4.0 mA	6240	3120
	0.0 mA	0	0

1762-IF2OF2 Input Data File

For each input module, slot x, words 0 and 1 contain the analog values of the inputs. The module can be configured to use either raw/proportional data or scaled-for-PID data. The input data file for each configuration is shown below.

Raw/Proportional Format

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	Channel 0 Data 0 to 32,768												0	0	0
1	0	Channel 1 Data 0 to 32,768												0	0	0
2	Reserved															
3	Reserved															
4	Reserved														S1	S0
5	U0	O0	U1	O1	Reserved											

Scaled-for-PID Format

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	Channel 0 Data 0 to 16,383												0	0
1	0	0	Channel 1 Data 0 to 16,383												0	0
2	Reserved															
3	Reserved															
4	Reserved														S1	S0
5	U0	O0	U1	O1	Reserved											

The bits are defined as follows:

- Sx = General status bits for channels 0 and 1. This bit is set when an error (over- or under-range) exists for that channel, or there is a general module hardware error.
- Ox = Over-range flag bits for channels 0 and 1. These bits can be used in the control program for error detection.
- Ux = Under-range flag bits for channels 0 and 1. These bits can be used in the control program for error detection.

1762-IF2OF2 Output Data File

For each module, slot x, words 0 and 1 contain the channel output data.

Raw/Proportional Format

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	Channel 0 Data 0 to 32,768												0	0	0
1	0	Channel 1 Data 0 to 32,768												0	0	0

Scaled-for-PID Format

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	Channel 0 Data 0 to 16,383												0	0
1	0	0	Channel 1 Data 0 to 16,383												0	0

1762-IF4 Input Data File

For each module, slot x, words 0 and 1 contain the analog values of the inputs. The module can be configured to use either raw/proportional data or scaled-for-PID data. The input data file for either configuration is shown below.

1762-IF4 Input Data File

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SGN0	Channel 0 Data														
1	SGN1	Channel 1 Data														
2	SGN2	Channel 2 Data														
3	SGN3	Channel 3 Data														
4	Reserved												S3	S2	S1	S0
5	U0	O0	U1	O1	U2	O2	U3	O3	Reserved							
6	Reserved															

The bits are defined as follows:

- Sx = General status bits for channels 0...3. This bit is set when an error (over- or under-range) exists for that channel, or there is a general module hardware error.
- Ox = Over-range flag bits for channels 0...3. These bits are set when the input signal is above the user-specified range. The module continues to convert data to the maximum full range value during an over-range condition. The bits reset when the over-range condition clears.
- UIx = Under-range flag bits for input channels 0...3. These bits are set when the input signal is below the user-specified range. The module continues to convert data to the maximum full range value during an under-range condition. The bits reset when the under-range condition clears.
- SGNx = The sign bit for channels 0...3.

1762-OF4 Input Data File

For each module, slot x, words 0 and 1 contain the analog output module status data for use in the control program.

1762-OF4 Input Data File

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved												SO3	SO2	SO1	SO0
1	Reserved								UO0	OO0	UO1	OO1	UO2	OO2	UO3	OO3

The bits are defined as follows:

- SOx = General status bits for output channels 0...3. These bits are set when an error (over- or under-range) exists for that channel, or there is a general module hardware error.
- OOx = Over-range flag bits for output channels 0...3. These bits indicate an input signal above the user range and can be used in the control program for error detection. The module continues to convert analog data to the maximum full range value while these bits are set (1). The bits is reset (0) when the error clears.
- UOx = Under-range flag bits for output channels 0...3. These bits indicate an input signal below the user range. They can be used in the control program for error detection. The module continues to convert analog data to the minimum full range value while these bits are set (1). The bits are reset (0) when the error clears.

1762-OF4 Output Data File

For each module, slot x, words 0...3 contain the channel output data.

Raw/Proportional Format

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	Channel 0 Data 0 to 32,760												0	0	0
1	0	Channel 1 Data 0 to 32,760												0	0	0
2	0	Channel 2 Data 0 to 32,760												0	0	0
3	0	Channel 3 Data 0 to 32,760												0	0	0

Words 0...3 contain the analog output data for channels 0...3, respectively. The module ignores the "don't care" bits (0...2), but checks the sign bit (15). If bit 15 equals one, the module sets the output value to 0V or 0 mA.

Scaled-for-PID Format

Word	Bit Position															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	Channel 0 Data 0 to 16,380												0	0
1	0	0	Channel 1 Data 0 to 16,380												0	0
2	0	0	Channel 2 Data 0 to 16,380												0	0
3	0	0	Channel 3 Data 0 to 16,380												0	0

Words 0...3 contain the analog output data for channels 0...3, respectively. The module ignores the "don't care" bits (0 and 1), but checks the sign bit (15), and bit 14. If bit 15 equals one, the module sets the output value to 0V or 0 mA. If bit 15 equals zero and bit 14 equals one, the module sets the output value to 10.5V DC or 21 mA.

Specialty I/O Configuration

1762-IR4 RTD/Resistance Module Input Data File

For each module, slot x, words 0...3 contain the analog values of the inputs. Words 4 and 5 provide sensor/channel status feedback. The input data file for each configuration is shown below.

Word /Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Analog Input Data Channel 0															
1	Analog Input Data Channel 1															
2	Analog Input Data Channel 2															

Word /Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
3	Analog Input Data Channel 3															
4	Reserved				OC3	OC2	OC1	OC0	Reserved				S3	S2	S1	S0
5	U0	O0	U1	O1	U2	O2	U3	O3	Reserved							

The bits are defined as follows:

- Sx = General status bits for input channels 0...3. These bits are set (1) when an error (over- or under-range, open-circuit or input data not valid condition) exists for that channel, or there is a general module hardware error. An input data not valid condition is determined by the user program. See MicroLogix 1200 RTD/Resistance Input Module User Manual, publication 1762-UM003, for details.
- OCx = Open-circuit indication for channels 0...3, using either RTD or resistance inputs. Short-circuit detection for RTD inputs only. Short-circuit detection for resistance inputs is not indicated because 0 is a valid number.
- Ox = Over-range flag bits for input channels 0...3, using either RTD or resistance inputs. These bits can be used in the control program for error detection.
- Ux = Under-range flag bits for channels 0...3, using RTD inputs only. These bits can be used in the control program for error detection. Under-range detection for direct resistance inputs is not indicated because 0 is a valid number.

1762-IT4 Thermocouple Module Input Data File

For each module, slot x, words 0...3 contain the analog values of the inputs. The input data file is shown below.

Word /Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SGN	Analog Input Data Channel 0														
1	SGN	Analog Input Data Channel 1														
2	SGN	Analog Input Data Channel 2														
3	SGN	Analog Input Data Channel 3														
4	Reserved			OC4	OC3	OC2	OC1	OC0	Reserved			S4	S3	S2	S1	S0
5	U0	O0	U1	O1	U2	O2	U3	O3	U4	O4	Reserved					

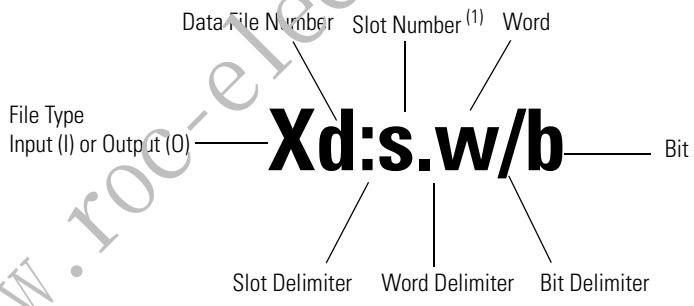
The bits are defined as follows:

- Sx = General status bits for channels 0...3 (S0...S3) and the CJC sensor (S4). This bit is set (1) when an error (over-range, under-range, open-circuit, or input data not valid) exists for that channel. An input data not valid condition is determined by the user program. Refer to MicroLogix 1200 I/O Thermocouple/mV Input Module User Manual, publication 1762-UM002 for additional details.
- OCx = Open-circuit indication for channels 0...3 (OC0...OC3) and the CJC sensor (OC4).
- Ox = Over-range flag bits for channels 0...3 (O0...O3) and the CJC sensor (O4). These bits can be used in the control program for error detection.
- Ux = Under-range flag bits for channels 0...3 (U0...U3) and the CJC sensor (U4). These bits can be used in the control program for error detection.

I/O Addressing

Addressing Details

The I/O addressing scheme and examples are shown below.



(1) I/O located on the controller (embedded I/O) is slot 0.
I/O added to the controller (expansion I/O) begins with slot 1.

I/O Addressing Scheme

Format	Explanation		
Od:s.w/b **Id:s.w/b**	**X**	File Type	Input (I) or Output (O)
	d	Data File Number *(optional)*	0 = output, 1 = input
	:	Slot delimiter *(optional, not required for Data Files 2…255)*	
	s	Slot number (decimal)	Embedded I/O: slot 0
			Expansion I/O: slots 1…7 for MicroLogix 1400 (See page 3 for an illustration.)
	.	Word delimiter. Required only if a word number is necessary as noted below.	
	w	Word number	Required to read/write words, or if the discrete bit number is above 15. Range: 0…255
	/	Bit delimiter	
	b	Bit number	0 to 15

Addressing examples

Addressing Level	Example Address[1]	Slot	Word	Bit
Bit addressing	O:0/4[2]	Output slot 0 (embedded I/O)	Word 0	Output bit 4
	O:2/7[2]	Output slot 2 (expansion I/O)	Word 0	Output bit 7
	I:1/4[2]	Input slot 1 (expansion I/O)	Word 0	Input bit 4
	I:0/15[2]	Input slot 0 (embedded I/O)	Word 0	Input bit 15
Word addressing	O:1.0	Output slot 1 (expansion I/O)	Word 0	
	I:7.3	Input slot 7 (expansion I/O)	Word 3	
	I:3.1	Input slot 3 (expansion I/O)	Word 1	

(1) The optional Data File Number is not shown in these examples.

(2) A word delimiter and number are not shown. Therefore, the address refers to word 0.

I/O Forcing

I/O forcing is the ability to override the actual status of the I/O at the user's discretion.

Input Forcing

When an input is forced, the value in the input data file is set to a user-defined state. For discrete inputs, you can force an input "on" or "off". When an input is forced, it no longer reflects the state of the physical input or the input LCD indicator. For embedded inputs, the controller reacts as if the force is applied to the physical input terminal.

TIP When an input is forced, it has no effect on the input device connected to the controller.

Output Forcing

When an output is forced, the controller overrides the status of the control program, and sets the output to the user-defined state. Discrete outputs can be forced "on" or "off". The value in the output file is unaffected by the force. It maintains the state determined by the logic in the control program. However, the state of the physical output and the output LCD indicator will be set to the forced state.

TIP If you force an output controlled by an executing PTOX or PWMX function, an instruction error is generated.

Input Filtering

The MicroLogix 1400 controllers allow users to configure groups of DC inputs for high-speed or normal operation. Users can configure each input group's response time. A configurable filter determines how long the input signal must be "on" or "off" before the controller recognizes the signal. The higher the value, the longer it takes for the input state to be recognized by the controller. Higher values provide more filtering, and are used in electrically noisy environments. Lower values provide less filtering, and are used to detect fast or narrow pulses. The filters can be set to a lower value when using high-speed counters, latching inputs, and input interrupts.

Input filtering is configured using RSLogix 500/RSLogix Micro programming software. To configure the filters using RSLogix 500/RSLogix Micro:

1. Open the Controller folder.
2. Open the I/O Configuration folder.
3. Open slot 0 (controller).
4. Select the Embedded I/O Configuration tab.

The input groups are pre-arranged. Select the filter time required for each input group. Apply a unique input filter setting to each of the input groups:

MicroLogix 1400 Input Groups

Controller	MicroLogix 1400
Input Groups	• 0 and 1 • 2 and 3 • 4 and 5 • 6 and 7 • 8 and 9 • 10 and 11 • 12 and 13 • 14 and 15 • 16to xxxx (reserved)

The minimum and maximum response times associated with each input filter setting can be found in your controller's User Manual.

Analog Inputs

The MicroLogix 1400 1766-L32BWAA, 1766-L32AWAA, and 1766-L32BXBA controllers support 4-channel, 12-bit resolution analog input with four 12-bit resolution analog input channels. These channels are single-ended (unipolar) circuits and accept 0...10V DC.

Input words 4...7 contain the value of analog inputs (Word 4: analog input channel 0, Word 5: analog input channel 1, Word 6: analog input channel 2, Word 7: analog input channel 3).

Analog Input Filter and Update times

The MicroLogix 1400 analog input filter is programmable. The slower the filter setting, the more immune the analog inputs are to electrical noise. The more immune the analog inputs are to electrical noise, the slower the inputs will be to update. Similarly, the faster the filter setting, the less immune the analog inputs are to electrical noise. The less immune the analog inputs are to electrical noise, the faster the inputs will be to update.

Programmable Filter Characteristics

Filter Setting Value(Hz)	Filter Bandwidth (-3 dB Freq Hz)	Setting Time (mSec)	Resolution (Bits)
10	10	100.00	12
50	50	20.00	12
60	60	16.67	12
250	250	4	12

TIP
- 10 Hz is the default setting
- The total update time is one ladder scan time plus the settling time.

EXAMPLE If a 250 Hz filter is selected, the maximum update Time = ladder scan time + 4ms.

Input Channel Filtering

The analog input channels incorporate on-board signal conditioning, to distinguish AC power line noise from normal variations in the input signal. Frequency components of the input signal at the filter frequency are rejected. Frequency components below the filter bandwidth (-3 dB frequency) are passed with under 3 dB of attenuation. This pass band allows the normal variation of sensor inputs such as temperature, pressure and flow transducers to be input data to the processor. Noise signals coupled in at frequencies above the pass band are sharply rejected. An area of particular concern is the 50/60 Hz region, where pick-up from power lines can occur.

Converting Analog Data

The analog input circuits are able to monitor voltage signals and convert them to digital data. There are five terminals assigned to the input channels that provide four voltage inputs, and a return signal (commons).

The following table shows sample Analog Signal and Data Word values using the nominal transfer function formula:

N=Vin x 4095/10 where Vin (analog signal) is in volts (V)

Converting Analog Input Data

Analog to data word conversion

Analog Signal	Data Word
0V	0
5V	2048
10V	4095

Analog inputs convert voltage signals into 12-bit values. To determine an approximate voltage that an input value represents, use the equation shown.

$$\frac{10V}{4095} \times inputvalue = inputvoltage(V)$$

For example, if an input value of 1200 is in the input image, the calculated value is as follows:

$$\frac{10V}{4095} \times 1200 = 2.9304(V)$$

Analog Outputs

The MicroLogix 1400 -L32BWAA, -L32AWAA, and -L32BXBA support 2-channel, 12-bit resolution analog output. These channels have 0...10V DC output range. Output words 4 and 5 contain the value of analog outputs (Word 4 : analog output channel 0, Word 5 : analog output channel 1).

Latching Inputs

Converting Analog Output Value to Actual Output Voltage

Analog outputs convert voltage signals into 12-bit values. To determine an approximate voltage that an output value represents, use the equation shown.

$$\frac{10V}{4095} \times outputvalue = outputvoltage(V)$$

For example, if an input value of 3000 is in the output image, the calculated value is as follows:

$$\frac{10V}{4095} \times 3000 = 7.326(V)$$

The MicroLogix 1400 controller provides the ability to individually configure inputs to be latching inputs (sometimes referred to as pulse catching inputs). A latching input is an input that captures a very fast pulse and holds it for a single controller scan. The pulse width that can be captured is dependent upon the input filtering selected for that input.

The following inputs can be configured as latching inputs.

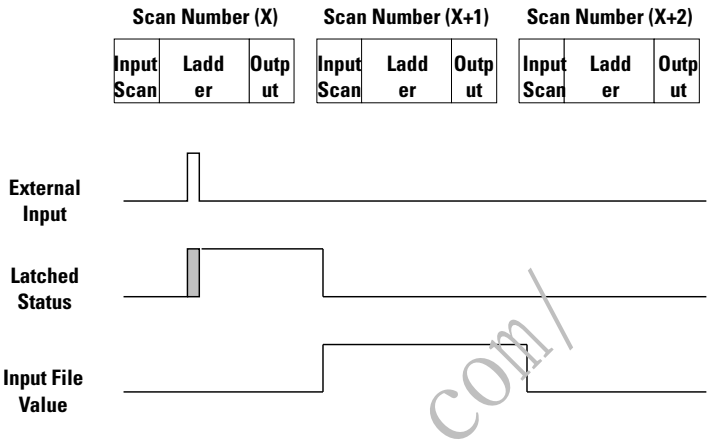
Controller	MicroLogix 1400
DC Inputs	0…11

Enable this feature using RSLogix 500/RSLogix Micro. With an open project:

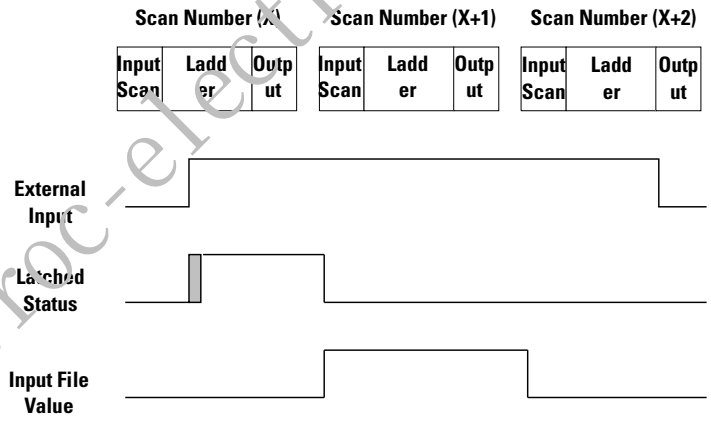
1. Open the Controller folder.
2. Open the I/O Configuration folder.
3. Open slot 0 (controller).
4. Select the Embedded I/O Configuration tab.
5. Select the mask bits for the inputs that you want to operate as latching inputs.
6. Select the state for the latching inputs. The controller can detect both "on" (rising edge) and "off" (falling edge) pulses, depending upon the configuration selected in the programming software.

The following information is provided for a controller looking for an "on" pulse. When an external signal is detected "on", the controller "latches" this event. In general, at the next input scan following this event, the input image point is turned "on" and remains "on" for the next controller scan. It is then set to "off" at the next input scan. The following figures help demonstrate this.

Rising Edge Behavior - Example 1



Rising Edge Behavior - Example 2



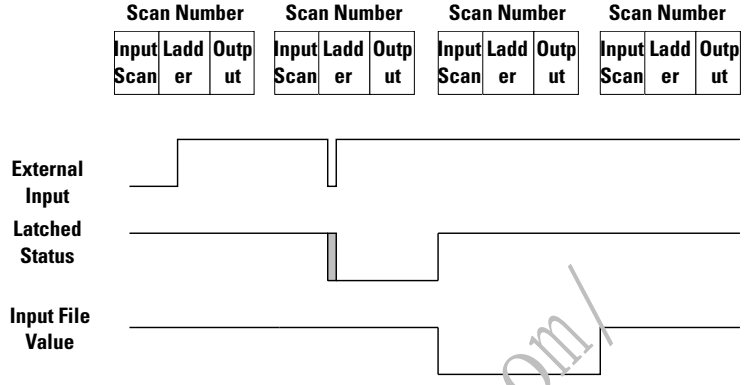
TIP The "gray" area of the Latched Status waveform is the input filter delay.

IMPORTANT The input file value does not represent the external input when the input is configured for latching behavior. When configured for rising edge behavior, the input file value is normally "off" ("on" for 1 scan when a rising edge pulse is detected).

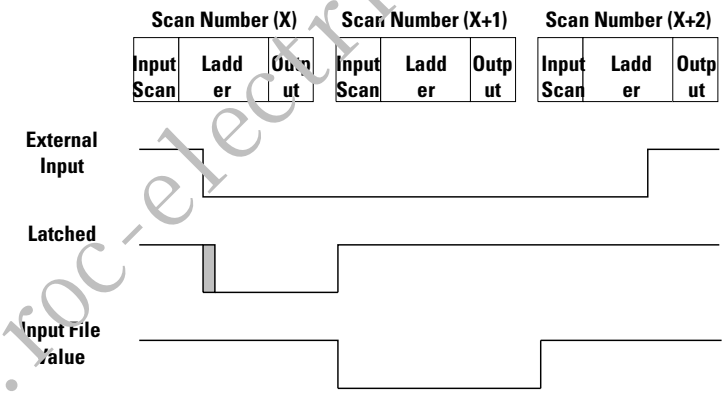
The previous examples demonstrate rising edge behavior. Falling edge behavior operates exactly the same way with these exceptions:

- The detection is on the "falling edge" of the external input.
- The input image is normally "on" (1), and changes to "off" (0) for one scan.

Falling Edge Behavior – Example 1



Falling Edge Behavior – Example 2



TIP The "gray" area of the Latched Status waveform is the input filter delay.

IMPORTANT The input file value does not represent the external input when the input is configured for latching behavior. When configured for falling edge behavior, the input file value is normally "on" ("off" for 1 scan when a falling edge pulse is detected).

Configure Expansion I/O Using RSLogix 500/ RSLogix Micro

Expansion I/O must be configured for use with the controller. Configuring expansion I/O can be done either manually, or automatically. Using RSLogix 500/RSLogix Micro:

1. Open the Controller folder.
2. Open the I/O Configuration folder.
3. For manual configuration, drag the Compact I/O module to the slot.

For automatic configuration, you must have the controller connected online to the computer (either directly or over a network). Click the Read I/O Config button on the I/O configuration screen. RSLogix 500/ RSLogix Micro will read the existing configuration of the controller's I/O.

Some I/O modules support or require configuration. To configure a specific module, double-click on the module, an I/O configuration screen will open that is specific to the module.

Notes:

Controller Memory and File Types

This chapter describes controller memory and the types of files used by the MicroLogix 1400 controller. The chapter is organized as follows:

- Controller Memory on page 21
- Data Files on page 26
- Protecting Data Files During Download on page 26
- Static File Protection on page 28
- Program Password Protection on page 30
- If a memory module is installed, the user program has the "Load Always" functionality enabled or the program is loaded from the memory module using RSLogix500 or LCD and the controller is password protected, then the controller compares the passwords before transferring the user program from the memory module to the controller. If the passwords do not match, the user program is not transferred and the Program Mismatch bit (S:5/9) is set. on page 33
- Allow Future Access Setting (OEM Lock) on page 34
- Web View Disable on page 35

Controller Memory

File Structure

MicroLogix 1400 controller user memory comprises Data Files, Function Files, and Program Files.

TIP The file types shown for data files 3…8 are the default file types for those file numbers and cannot be changed. Data files 9…255 can be added to your program to operate as bit, timer, counter, or other files shown below.

Controller User Memory File Types

Data Files		Function Files		Program Files		Specialty Files	
0	Output File	HSC	High Speed Counter	0	System File 0	0	Data Log Queue 0
1	Input File	PTOX	Pulse Train Output	1	System File 1	1	Data Log Queue 1
2	Status File	PWMX	Pulse Width Modulation	2	Program File 2	2…255	Data Log Queues 2…255

Controller User Memory File Types

Data Files		Function Files		Program Files		Specialty Files	
3	Bit File	STI	Selectable Timed Interrupt	3…255	Program Files 3…255	0	Recipe File 0
4	Timer File	EII	Event Input Interrupt			1	Recipe File 1
5	Counter File	RTC	Real Time Clock			2…255	Recipe Files 2…255
6	Control File						
7	Integer File	MMI	Memory Module Information				
8	Floating Point File						
9…255	(B) Bit (T) Timer (C) Counter (R) Control (N) Integer (F) Floating Point (ST) String (A) ASCII (L) Long Word (MG) Message (PD) PID (PLS) Programmable Limit Switch (RI) Routing Information (RIX) Extended Routing Information	BHI	Base Hardware Information				
		CS0	Communications Status for Channel 0				
		CS2	Communications Status for Channel 2				
		IOS	I/O Status				
		DLS	Data Log Status				
		LCD	LCD				
		ES1	Ethernet Status for Channel 1				

User Memory

User memory is the amount of controller storage available to store data such as ladder logic, data table files, and I/O configuration.

User data files consist of the system status file, I/O image files, and all other user-creatable data files (bit, timer, counter, control, integer, string, long word, MSG, and PID).

A word is defined as a unit of memory in the controller. The amount of memory available to the user for data files and program files is measured in user words. Memory consumption is allocated as follows:

- For *data files*, a word is the equivalent of 16 bits of memory. For example,
  - 1 integer data file element = 1 user word
  - 1 long word file element = 2 user words
  - 1 timer data file element = 3 user words

    TIP Each input and output data element consumes 3 user words due to the overhead associated with I/O forcing.

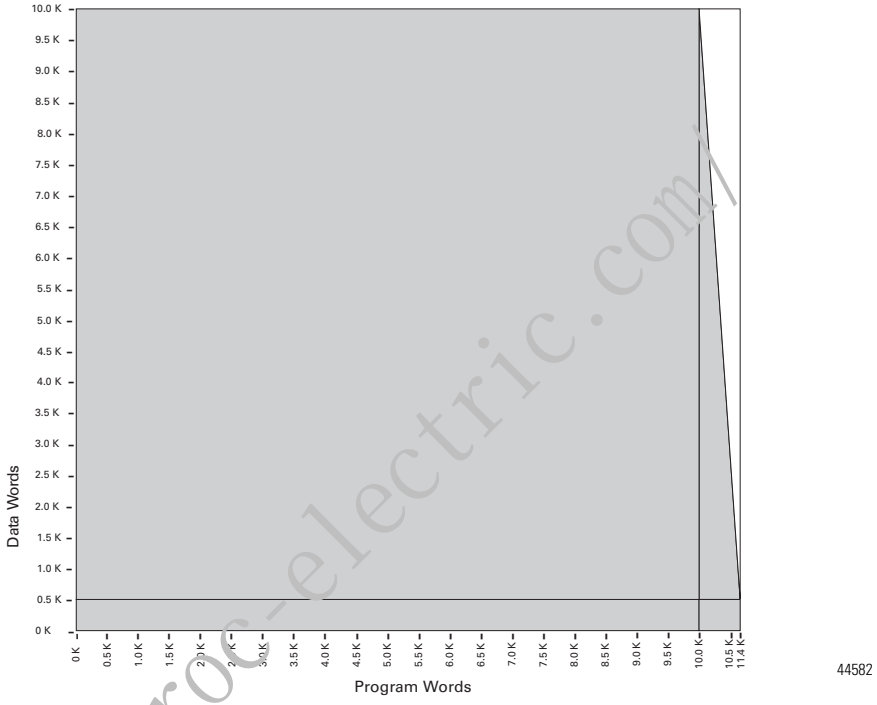
- For *program files*, a word is the equivalent of a ladder instruction with one operand. For example(1),
  - 1 XIC instruction, which has 1 operand, consumes 1 user word
  - 1 EQU instruction, which has 2 operands, consumes 2 user words
  - 1 ADD instruction, which has 3 operands, consumes 3 user words
- *Function files* do not consume user memory.

TIP Although the controller allows up to 256 elements in a file, it may not actually be possible to create a file with that many elements due to the user memory size in the controller.

(1) These are approximate values. For actual memory usage, see the tables in Appendix A of this manual.

MicroLogix 1400 User Memory

The MicroLogix 1400 controller supports 20K of memory. Memory can be used for program files and data files. The maximum data memory usage is 10K words as shown.



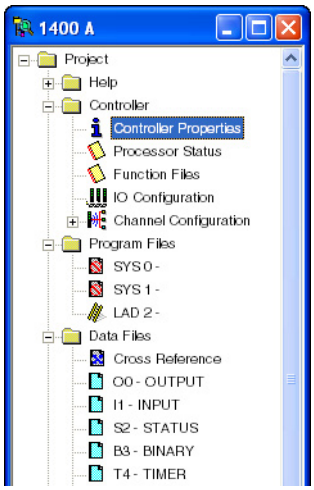
To find the memory usage for specific instructions, see MicroLogix 1400 Memory Usage and Instruction Execution Time on page 497.

The MicroLogix 1400 controller also supports 128K bytes of battery backed memory for data logging or recipe operations. See Chapter 25 for Data Logging and Recipe information.

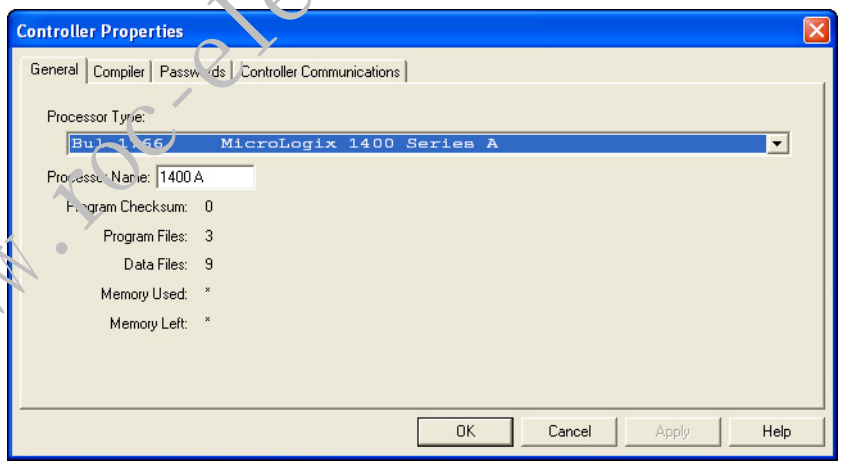
To find the memory usage for specific instructions See System Status File on page 503.

Viewing Controller Memory Usage

1. Highlight and open *Controller Properties*.



2. The amount of *Memory Used* and *Memory Left* will appear in the *Controller Properties* window once the program has been verified.



Data Files

Data files store numeric information, including I/O, status, and other data associated with the instructions used in ladder subroutines. The data file types are:

Data File Types

File Name	File Identifier	File Number(1)	Words per Element	File Description
Output File	O	0	1	The Output File stores the values that are written to the physical outputs during the Output Scan.
Input File	I	1	1	The Input File stores the values that are read from the physical inputs during the Input Scan.
Status File	S	2	1	The contents of the Status File are determined by the functions which utilize the Status File. See System Status File on page 503 for a detailed description.
Bit File	B	**3**, 9…255	1	The Bit File is a general purpose file typically used for bit logic.
Timer File	T	**4**, 9…255	3	The Timer File is used for maintaining timing information for ladder logic timing instructions. See Timer and Counter Instructions on page 145 for instruction information.
Counter File	C	**5**, 9…255	3	The Counter File is used for maintaining counting information for ladder logic counting instructions. See Timer and Counter Instructions on page 145 for instruction information.
Control File	R	**6**, 9…255	3	The Control Data file is used for maintaining length and position information for various ladder logic instructions. See Control Data File on page 313 for more information.
Integer File	N	**7**, 9…255	1	The Integer File is a general purpose file consisting of 16-bit, signed integer data words.
Floating Point File	F	**8**, 9…255	2	The Floating Point File is a general purpose file consisting of 32-bit IEEE-754 floating point data elements. See Using the Floating Point (F) Data File on page 164 for more information.
String File	ST	9…255	42	The String File is a file that stores ASCII characters. See String (ST) Data File on page 312 for more information.
ASCII File	A	9…255	1	The ASCII File is a file that stores ASCII characters.
Long Word File	L	9…255	2	The Long Word File is a general purpose file consisting of 32-bit, signed integer data words.
Message File	MG	9…255	25	The Message File is associated with the MSG instruction. See Communications Instructions on page 337 for information on the MSG instruction.
Programmable Limit Switch File	PLS	9…255	6	The Programmable Limit Switch (PLS) File allows you to configure the High-Speed Counter to operate as a PLS or rotary cam switch. See Programmable Limit Switch (PLS) File on page 105 for information.
PID File	PD	9…255	23	The PID File is associated with the PID instruction. See Process Control Instruction on page 281 for more information.
Routing Information File	RI	9…255	20	The Routing Information File is associated with the MSG instruction. See Communications Instructions on page 337 for information on the MSG instruction.
Extended Routing Information File	RIX	9…255	25	The extended Routing Information File is associated with the MSG instruction. See Communications Instructions on page 337 for information on the MSG instruction.

(1) File Number in BOLD is the default. Additional data files of tha type can be configured using the ramaining numbers.

Protecting Data Files During Download

Data File Download Protection

Once a user program is in the controller, there may be a need to update the ladder logic and download it to the controller without destroying user-configured

variables in one or more data files in the controller. This situation can occur when an application needs to be updated, but the data that is relevant to the installation needs to remain intact.

This capability is referred to as *Data File Download Protection*. The protection feature operates when:

- A User Program is downloaded via programming software
- A User Program is downloaded from a Memory Module

Setting Download File Protection

Download File Protection can be applied to the following data file types:

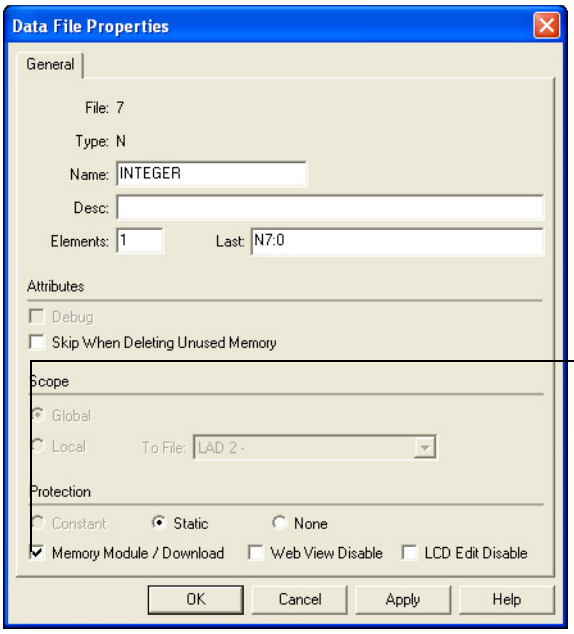
- Output (O)
- Input (I)
- Binary (B)
- Timer (T)
- Counter (C)
- Control (R)
- Integer (N)
- Floating Point (F)
- String (ST)
- ASCII (A)
- Long Word (L)
- Proportional Integral Derivative (PD)
- Message (MG)
- Programmable Limit Switch (PLS)
- Routing Information (RI)
- Extended Routing Information (RIX)
- Recipe (Series B only)

TIP The data in the Status File cannot be protected.

User Program Transfer Requirements

Data File Download Protection only operates when the following conditions are met during a User Program or Memory Module download to the controller:

- The controller contains protected data files.
- The program being downloaded has the same number of protected data files as the program currently in the controller.
- All protected data file numbers, types, and sizes (number of elements) currently in the controller exactly match that of the program being downloaded to the controller.



Access the Download Data File Protect feature using RSLogix 500/RSLogix Micro programming software. For each data file you want protected, check the Memory Module/Download item within the protection box in the Data File Properties screen as shown in this illustration. To access this screen, right click the desired data file.

If all of these conditions are met, the controller will not write over any data file in the controller that is configured as Download Protected when a program is downloaded from a memory module or programming software.

If any of these conditions are not met, the entire User Program is transferred to the controller. Additionally, if the program in the controller contains protected files, the Data Protection Lost indicator (S:36/10) is set to indicate that protected data has been lost. For example, a control program with protected files is transferred to the controller. The original program did not have protected files or the files did not match. The data protection lost indicator (S:36/10) is then set. The data protection lost indicator represents that the protected files within the controller have had values downloaded and the user application may need to be re-configured.

TIP The controller will not clear the Data Protection Lost indicator. It is up to the user to clear this bit.

Static File Protection

When a data file is Static File Protected, the values contained in it cannot be changed via communications, except during a program download to the controller.

Using Static File Protection with Data File Download Protection

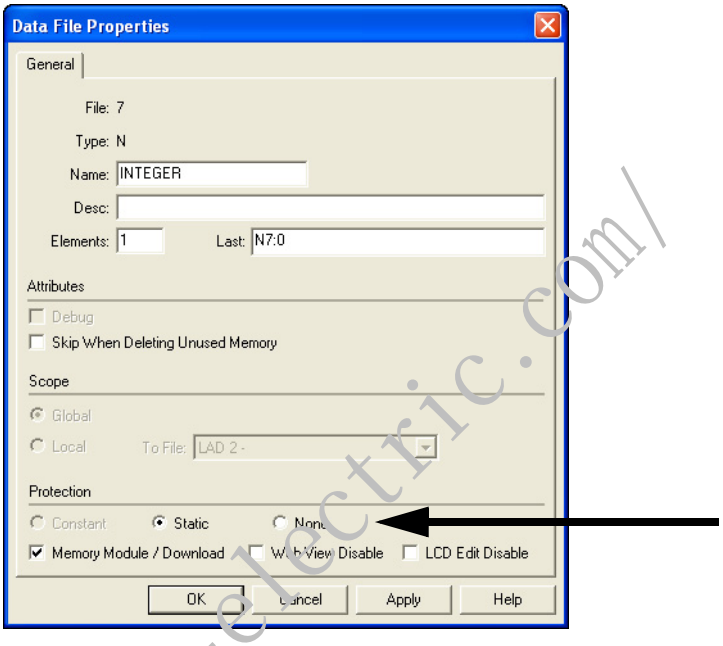
Static File Protection and Data File Download Protection can be used in combination with MicroLogix 1400 Controller Series A and higher.

Setting Static File Protection

Static File Protection can be applied to the following data file types:

- Output (O)
- Input (I)
- Status (S)
- Binary (B)
- Timer (T)
- Counter (C)
- Control (R)
- Integer (N)
- Floating Point (F)
- String (ST)
- ASCII (A)
- Long Word (L)
- Proportional Integral Derivative (PD)
- Message (MG)
- Programmable Limit Switch (PLS)
- Routing Information (RI)
- Extended Routing Information (RIX)

Access the Static File Protect feature using RSLogix 500/RSLogix Micro programming software. For each data file you want protected, select the Static protection in the Data File Properties screen as shown in this illustration. To access this screen, right mouse click on the desired data file.



Program Password Protection

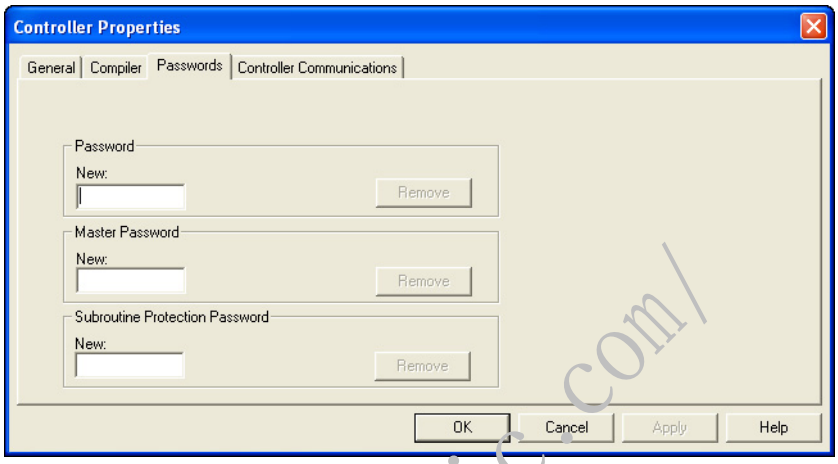
Programming software access to MicroLogix programs and controllers can be restricted through password protection. Each controller program may contain two passwords, the Password and the Master Password. (The Master Password is ignored unless a Password has also been configured.) The Master Password takes precedence over the Password. That way, all controllers on an assembly line, for instance, can each have a different Password, but have the same Master Password, allowing access to all controllers for supervisory or maintenance purposes.

MicroLogix 1400 controllers also have a third password, the Subroutine Protection Password. This password can be configured to restrict viewing of selected ladder files.

IMPORTANT Starting with RSLogix 500/Micro version 8.40, these passwords may optionally be encrypted so that as they are transmitted between the controller and PCs running RSLogix 500/Micro, someone with physical access to the network where communications is occurring over will not be able to "sniff" the network and capture the password(s). Even so, password protection should not be solely relied upon to prevent unintended changes to the program and data table. Refer to other sections in this chapter for additional steps that you may be able to take with your specific MicroLogix controller to prevent unintended changes.

By default, no passwords are configured in the controller. You can create, change or delete a password using the Controller Properties dialog box. Unencrypted

passwords consist of up to 10 numeric characters (0...9). Encrypted passwords contain at least one non-numeric character.



Once a Password has been created and downloaded into the controller, any time the user attempts to open that offline file or go online with that controller, they will be prompted to provide either the Password or the Master Password before the software proceeds.

If a password is lost or forgotten, there is no way to bypass the password to recover the program. The only option is to clear the controller's memory.

If a memory module is installed, the user program has the "Load Always" functionality enabled and the user program is password protected, then the controller compares the passwords before transferring the user program from the memory module to the controller. If the passwords do not match, the user program is not transferred and the Program Mismatch bit (S:5/9) is set.

TIP If a password is lost or forgotten, there is no way to bypass the password to recover the program. The only option is to clear the controller memory. See Clearing the Controller Memory on page 34.

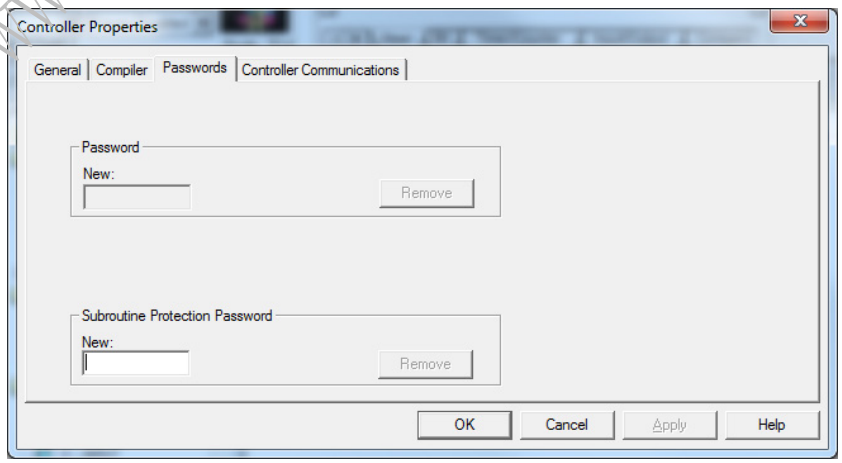
Program Password Protection for Series B (Enhanced Password Security)

RSLogix 500 version 11.00.00 or higher access to MicroLogix 1400 Series B (Enhanced Password Security) programs and controllers can be restricted through online password protection. Each Series B (Enhanced Password Security) program contains one password which can be set/change/clear when program is online with controller.

IMPORTANT	Master Password is not available in Series B (Enhanced Password Security) programs. LCD configuration changes are protected by a separate LCD password. For more information on LCD Password Setup, refer to the MicroLogix 1400 Programmable Controllers User Manual, publication 1766-UM001.

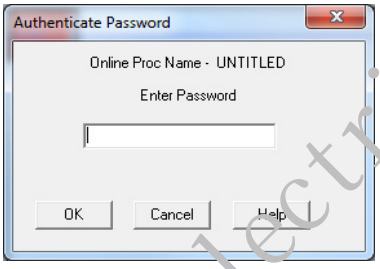
Series B (Enhanced Password Security) programs also have a second password, the Subroutine Protection Password similar to Series A and Series B program. This password can be set, changed, or cleared offline or when the program is online with the controller. This password can be configured to restrict the viewing of selected ladder files whether online or offline.

By default, no passwords are configured in the controller. You can create, change or delete a password using the Controller properties dialog box. The length of the Password should be 8 to 32 and it can contain the combination of integers (0-9), alphabets (upper and lower cases) and symbols (valid symbols are `~!@#$%^&*()-_=+[{]}\\|;:'\",<.>/?). The length of the Subroutine Protection Password should be 1 to 8 characters. It must contain at least one letter (upper or lower case) and can contain integers.



IMPORTANT Starting with RSLogix 500 version 11, these passwords are always encrypted so that as they are transmitted between the controller and PCs running RSLogix 500, someone with physical access to the network where communications is occurring over will not be able to "sniff" the network and capture the password(s). Even so, password protection should not be solely relied upon to prevent unintended changes to the program and data table. Refer to other sections in this chapter for additional steps that you may be able to take with your specific MicroLogix controller to prevent unintended changes.

Once a Password has been created and downloaded into the controller, any attempt to download, upload, or go online to, from or with that controller, will prompt for the Password before the software proceeds.



If you forget or lose a password there is no way to bypass the password prompt to download, upload, or go online to, from or with that controller.

IMPORTANT If the controller is password protected and programming software is authenticated by the controller and communication has started:

- Another instance of programming software on a different host machine cannot access the same controller.
- Another instance of programming software on the same host machine but through different channels cannot access the same controller
- Another instance of programming software on the same host machine through the same channel can access the same controller

If a memory module is installed, the user program has the "Load Always" functionality enabled or the program is loaded from the memory module using RSLogix500 or LCD and the controller is password protected, then the controller compares the passwords before transferring the user program from the memory module to the controller. If the passwords do not match, the user program is not transferred and the Program Mismatch bit (S:5/9) is set.

IMPORTANT If the controller is password protected, the program can be transferred from Memory Module (MM) to controller only when both controller and MM have "Series A or B" programs or both have "Series B (Enhanced Password Security)" programs.

IMPORTANT A memory module containing a Series B (Enhanced Password Security) program is compatible with firmware revision 21.00 or higher.

Clearing the Controller Memory

If you are locked out because you do not have the password for the controller, you can clear the controller memory and download a new User Program.

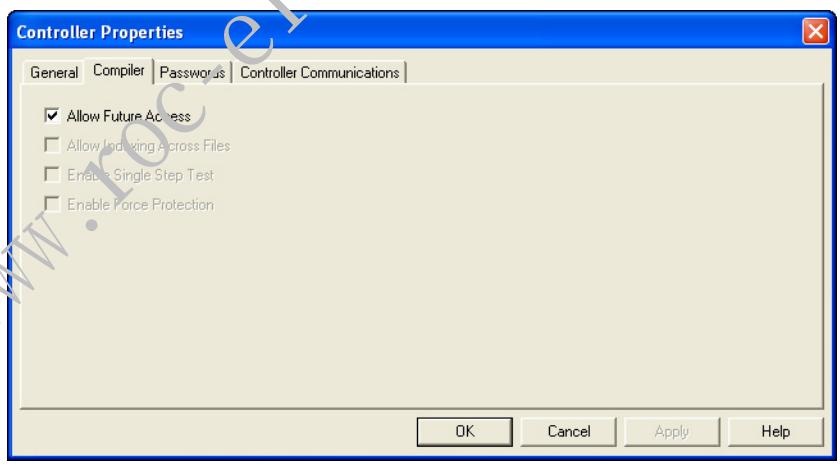
For controllers with firmware revision 14 and earlier, you can clear the memory when the programming software prompts you for a System or Master Password to go online with the controller. To do so:

1. Enter 65257636 (the telephone keypad equivalent of MLCLRMEM, MicroLogix Clear Memory).
2. When the Programming Software detects this number has been entered, it asks if you want to clear the memory in the controller.
3. If you reply "yes" to this prompt, the programming software instructs the controller to clear Program memory.

Allow Future Access Setting (OEM Lock)

The controller supports a feature which allows you to select if future access to the User Program should be allowed or disallowed after it has been transferred to the controller.

The Allow Future Access setting is found in the Controller Properties window as shown below.



When Allow Future Access is deselected, the controller requires that the User Program in the controller is the same as the one in the programming device. If the programming device does not have a matching copy of the User Program, access to the User Program in the controller is denied. To access the User Program, clear controller memory and reload the program.

TIP Functions such as change mode, clear memory, restore program, and transfer memory module are allowed regardless of this selection.

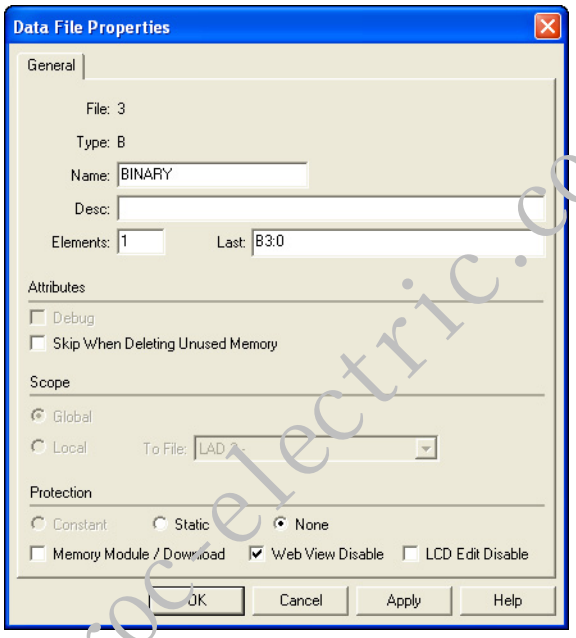
Controller passwords are not associated with the Allow Future Access setting.

IMPORTANT The Clear Controller Memory feature is not supported from FRN 15 onwards.

Web View Disable

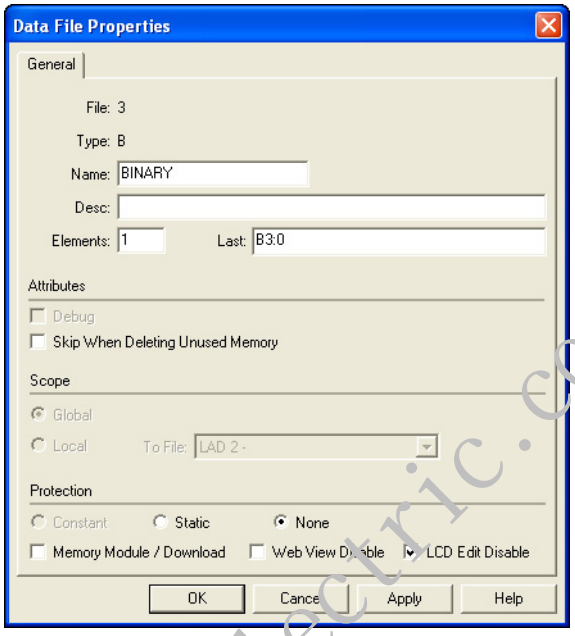
This allows selective disabling individual Data Files from Web View.

Using RSLogix 500/RSLogix Micro V8.10 and higher, you can disable individual data files from being viewed via any web browser by selecting the data file's properties page and checking the Web View Disable check box as shown below. Any data file property changes must be made offline and downloaded to the processor.



LCD Edit Disable

This allows selective protection of individual Data Files on the LCD. Using RSLogix 500/RSLogix Micro V8.10 and higher, select the data file's properties page and check the LCD Edit Disable check box as shown below. Any data file property changes must be made offline and downloaded to the processor.



Function Files

This chapter describes controller function files. The chapter is organized as follows:

- Overview on page 37
- Real-Time Clock Function File on page 38
- Memory Module Information Function File on page 42
- Base Hardware Information Function File on page 44
- Communications Status File on page 45
- Ethernet Communications Status File on page 60
- Input/Output Status File on page 67

Overview

Function Files are one of the three primary file structures within the MicroLogix 1400 controller (Program Files and Data Files are the others). Function Files provide an efficient and logical interface to controller resources. Controller resources are resident (permanent) features such as the Real-Time Clock and High-Speed Counter. The features are available to the control program through either instructions that are dedicated to a specific function file, or via standard instructions such as MOV and ADD. The Function File types are:

File Name	File Identifier	File Description
High-Speed Counter	**HSC**	This file type is associated with the High-Speed Counter function. See Using the High-Speed Counter and Programmable Limit Switch on page 77 for more information.
Extended Pulse Train Output	**PTOX**	This file type is associated with the Pulse Train Output Instruction. See Pulse Train Outputs (PTOX) Function File on page 115 for more information.
Extended Pulse Width Modulation	**PWMX**	This file type is associated with the Pulse Width Modulation instruction. See Pulse Width Modulation (PWMX) Function File on page 129 for more information.
Selectable Timed Interrupt	**STI**	This file type is associated with the Selectable Timed Interrupt function. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.
Event Input Interrupt	**EII**	This file type is associated with the Event Input Interrupt instruction. See Using the Event Input Interrupt (EII) Function File on page 276 for more information.
Real-Time Clock	**RTC**	This file type is associated with the Real-Time Clock (time of day) function. See Real-Time Clock Function File on page 38 for more information.
Memory Module Information	**MMI**	This file type contains information about the Memory Module. See Memory Module Information Function File on page 42 for more information.
Base Hardware Information	**BHI**	This file type contains information about the controller's hardware. See Base Hardware Information Function File on page 44 for the file structure.
Communications Status File for Channel 0	**CS0**	This file type contains information about the Communications with the controller. See Communications Status File on page 45 for the file structure.
Communications Status File for Channel 2	**CS2**	

File Name	File Identifier	File Description
I/O Status File	**IOS**	This file type contains information about the controller I/O. See Input/Output Status File on page 67 for the file structure.
Ethernet Status File for Channel 1	**ES1**	The file type contains information about the Ethernet Communications with the controller.
LCD Information File	**LCD**	This file type is associated with the LCD screen, keypads, and trimpot.

Real-Time Clock Function File

The real-time clock provides year, month, day of month, day of week, hour, minute, and second information to the Real-Time Clock (RTC) Function File in the controller.

The Real-Time Clock parameters and their valid ranges are shown in the table below.

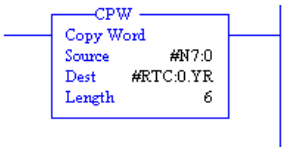
Real-Time Clock Function File

Feature	Address	Data Format	Range	Type	User Program Access
YR - RTC Year	RTC:0.YR	word	1998…2097	status	read/write
MON - RTC Month	RTC:0.MON	word	1…12	status	read/write
DAY - RTC Day of Month	RTC:0.DAY	word	1…31	status	read/write
HR - RTC Hours	RTC:0.HR	word	0…23 (military time)	status	read/write
MIN - RTC Minutes	RTC:0.MIN	word	0…59	status	read/write
SEC - RTC Seconds	RTC:0.SEC	word	0…59	status	read/write
DOW - RTC Day of Week	RTC:0.DOW	word	0…6 (Sunday to Saturday)	status	read-only
DS - Disabled	RTC:0/DS	binary	0 or 1	status	read-only
BL - RTC Battery Low	RTC:0/BL	binary	0 or 1	status	read-only

Writing Data to the Real-Time Clock

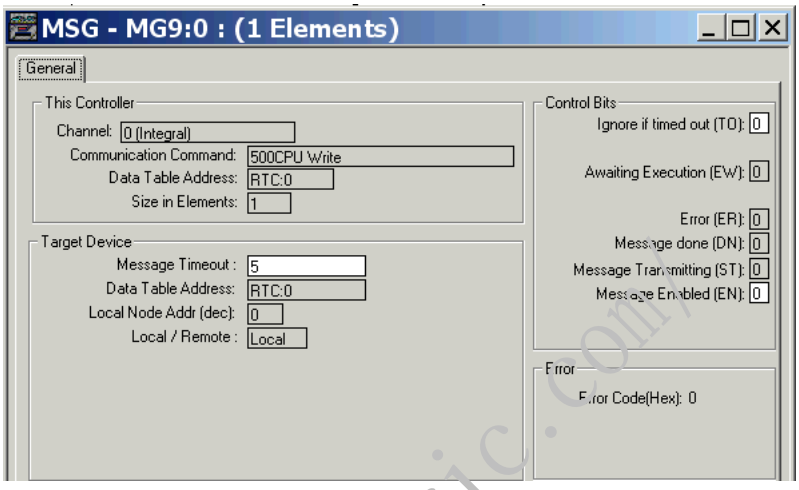
The RTC settings may be changed by either the user program, a write MSG instruction from another MicroLogix controller, or the programming software.

Use the Copy Word (CPW) instruction to adjust the RTC settings within the ladder logic as follows:

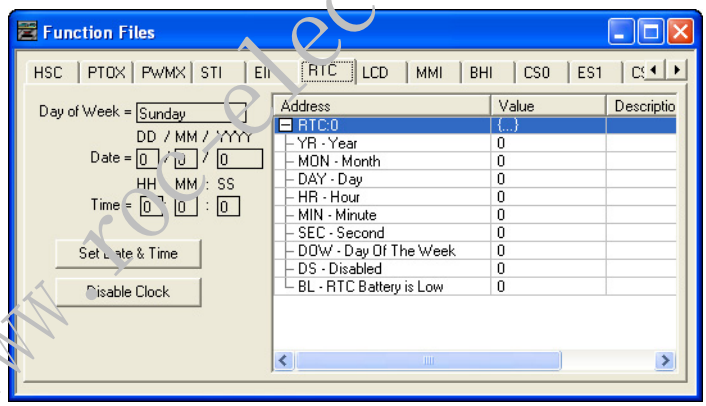


IMPORTANT A Major fault (44h) is generated if any of the data being written to the RTC function file is invalid. For example, setting the Seconds to 61 or setting the Day of Month to 32.

An example write MSG from another MicroLogix controller to synchronize their RTCs is shown here:



The programming screen is shown below:



When valid data is sent to the real-time clock from the programming device or another controller, the new values take effect immediately. In RSLogix 500/ RSLogix Micro, click on *Set Date & Time* in the RTC Function File screen to set the RTC time to the current time on your PC.

The real-time clock does not allow you to load or store invalid date or time data.

TIP Use the *Disable Clock* button in your programming device to disable the real-time clock before storing a module. This decreases the drain on the battery during storage.

Real-Time Clock Accuracy

The following table indicates the expected accuracy of the real-time clock for various temperatures.

Real-Time Clock Accuracy at Various Temperatures

Ambient Temperature	Accuracy[1]
0°C (+32°F)	-13…-121 seconds/month
+25°C (+77°F)	+54…-54 seconds/month
+40°C (+104°F)	+29…-78 seconds/month
+55°C (+131°F)	-43…-150 seconds/month

(1) These numbers are worst case values over a 31 day month.

RTC Battery Operation

The real-time clock uses the same replaceable battery that the controller uses. The RTC Function File features a battery low indicator bit (RTC:0/BL), which shows the status of the replacement battery. When the battery is low, the indicator bit is set (1). This means that the battery wire connector could be disconnected or if the battery is connected, the battery may be ready to fail in the next two weeks. In the latter case, the replacement battery needs to be replaced with a new one. When the battery low indicator bit is clear (0), the battery level is acceptable.

ATTENTION: Operating with a low battery indication for more than 14 days may result in invalid RTC data if power is removed from the controller.

RTA - Real Time Clock Adjust Instruction

Instruction Type: output

Execution Time for the RTA Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	999.8510 µs	0.4090 µs

The RTA instruction is used to synchronize the controllers Real-Time Clock (RTC) with an external source. The RTA instruction will adjust the RTC to the nearest minute. The RTA instruction adjusts the RTC based on the value of the RTC Seconds as described below.

IMPORTANT The RTA instruction will only change the RTC when the RTA rung is evaluated true, after it was previously false (false-to-true transition). The RTA instruction will have no effect if the rung is always true or false.

RTA is set:

- If RTC Seconds are less than 30, then RTC Seconds is reset to 0.

- If RTC Seconds are greater than or equal to 30, then the RTC Minutes are incremented by 1 and RTC Seconds are reset to 0.

The following conditions cause the RTA instruction to have no effect on the RTC data:

- RTC is disabled
- An external (via communications) message to the RTC is in progress when the RTA instruction is executed. (External communications to the RTC takes precedence over the RTA instruction.)

To re-activate the RTA instruction, the RTA rung must become false, and then true.

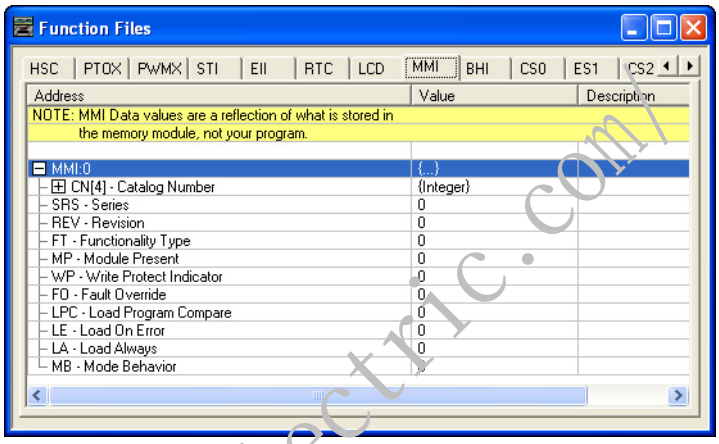
TIP There is only one internal storage bit allocated in the system for this instruction. Do not use more than one RTA instruction in your program.

TIP You can also use a MSG instruction to write RTC data from one controller to another to synchronize time. To send (write) RTC data, use RTC:0 as the source and the destination.

Memory Module Information Function File

The controller has a Memory Module Information (MMI) File which is updated with data from the attached memory module. At power-up or on detection of a memory module being inserted, the catalog number, series, revision, and type are identified and written to the MMI file in the user program. If a memory module is not attached, zeros are written to the MMI file.

The memory module function file programming screen is shown below:



The parameters and their valid ranges are shown in the table below.

MMI Function File Parameters

Feature	Address	Data Format	Type	User Program Access
FT - Functionality Type	MMI:0.FT	word (INT)	status	read-only
MP - Module Present	MMI:0/MP	binary (bit)	status	read-only
WP - Write Protect	MMI:0/WP	binary (bit)	control	read-only
FO - Fault Override	MMI:0/FO	binary (bit)	control	read-only
LPC - Program Compare	MMI:0/LPC	binary (bit)	control	read-only
LE - Load On Error	MMI:0/LE	binary (bit)	control	read-only
LA - Load Always	MMI:0/LA	binary (bit)	control	read-only
MB - Mode Behavior	MMI:0/MB	binary (bit)	control	read-only

FT - Functionality Type

The LSB of this word identifies the type of module installed:

- 1 = Memory Module (MM1)

MP - Module Present

The MP (Module Present) bit can be used in the user program to determine when a memory module is present on the controller. This bit is updated once per scan, provided the memory module is first recognized by the controller. To be

recognized by the controller, the memory module must be installed prior to power-up or when the controller is in a non-executing mode. If a memory module is installed when the controller is in an executing mode, it is not recognized. If a recognized memory module is removed during an executing mode, this bit is cleared (0) at the end of the next ladder scan.

WP - Write Protect

When the WP (Write Protect) bit is set (1), the module is write-protected and the user program and data within the memory module cannot be overwritten

IMPORTANT	Once the WP bit is set (1), it cannot be cleared. Only set this bit if you want the contents of the memory module to become permanent.

FO - Fault Override

The FO (Fault Override) bit represents the status of the fault override setting of the program stored in the memory module. It enables you to determine the value of the FO bit without actually loading the program from the memory module.

IMPORTANT	The memory module fault override selection in the Memory Module Information (MMI) file does not determine the controller's operation. It merely displays the setting of the user program's Fault Override bit (S:1/8) in the memory module.

See Fault Override At Power-Up on page 506 for more information.

LPC - Load Program Compare

The LPC (Load Program Compare) bit shows the status of the load program compare selection in the memory module's user program status file. It enables you to determine the value without actually loading the user program from the memory module.

See Memory Module Program Compare on page 511 for more information.

LE - Load on Error

The LE (Load on Error) bit represents the status of the load on error setting in the program stored in the memory module. It enables you to determine the value of the selection without actually loading the user program from the memory module.

See Load Memory Module On Error Or Default Program on page 507 for more information.

LA - Load Always

The LA (Load Always) bit represents the status of the load always setting in the program stored in the memory module. It enables you to determine the value of the selection without actually loading the user program from the memory module.

See Load Memory Module Always on page 507 for more information.

MB - Mode Behavior

The MB (Mode Behavior) bit represents the status of the mode behavior setting in the program stored in the memory module. It enables you to determine the value of the selection without actually loading the user program from the memory module.

See Power-Up Mode Behavior on page 507 for more information.

Base Hardware Information Function File

The base hardware information (BHI) file is a read-only file that contains a description of the MicroLogix 1400 Controller.

Base Hardware Information Function File (BHI)

Address	Description
BHI:0.CN	CN – Catalog Number
BHI:0.SRS	SRS – Series
BHI:0.REV	REV – Revision
BHI:0.FT	FT – Functionality Type

Communications Status File

The Communications Status (CS) File is a read-only file that contains information on how the controller communication parameters are configured and status information on communications activity.

The communications status file uses:

Communications Status File Size

Controller	Number of Word Elements
MicroLogix 1400	71 1-word elements

There are three Communications Status Files for each communications port. Communications Status File CS0 and CS2 correspond to Channel 0 and Channel 2 on the controller. Ethernet Communications Status File ES corresponds to Channel 1 on the controller.

TIP You can use the Communications Status File information as a troubleshooting tool for communications issues.

The data file is structured as shown below:

Communications Status File

Word	Description	Applies to Controller	Details on Page
0…5	General Channel Status Block	MicroLogix 1400	45
6…22	DLL Diagnostic Counters Block	MicroLogix 1400	47
23…42	DLL Active Node Table Block	MicroLogix 1400	59
words 43…70 when using DF1 Full-Duplex, DF1 Half-Duplex, DH-485, or ASCII:			
43	End of List Category Identifier Code (always 0)	MicroLogix 1400	--
43…70	Reserved	• MicroLogix 1400	--
words 43…70 when using Modbus RTU Slave, Master or DF1 Half-Duplex Master:			
43…69	Modbus Slave Diagnostic Counters Block	• MicroLogix 1400	53
70	End of List Category Identifier Code (always 0)	• MicroLogix 1400	--

The following tables show the details of each block in the Communications Status File.

General Status Block of Communications Status File

General Channel Status Block

Word	Bit	Description
0	-	Communications Channel General Status Information Category Identifier Code
1	-	Length
2	-	Format Code

General Channel Status Block

3	-	Communications Configuration Error Code
4	0	ICP – Incoming Command Pending Bit This bit is set (1) when the controller determines that another device has requested information from this controller. Once the request has been satisfied, the bit is cleared (0).
	1	MRP – Incoming Message Reply Pending Bit This bit is set (1) when the controller determines that another device has supplied the information requested by a MSG instruction executed by this controller. When the appropriate MSG instruction is serviced (during end-of-scan, SVC, or REF), this bit is cleared (0).
	2	MCP – Outgoing Message Command Pending Bit This bit is set (1) when the controller has one or more MSG instructions enabled and in the communication queue. This bit is cleared (0) when the queue is empty.
	3	SSB – Selection Status Bit This bit indicates that the controller is in the System Mode. It is always set.
	4	CAB – Communications Active Bit This bit is set (1) when at least one other device is on the DH-485 network. If no other devices are on the network, this bit is cleared (0).
	5…14	Reserved
	15	Communications Toggle Push Button Communications Defaults Active. This bit is set (1) whenever Channel 0 is in the default communications mode. The bit is cleared (0) when Channel 0 is in user configured communications mode.
5	0…7	Node Address – This byte value contains the node address of your controller on the network.
	8…15	Baud Rate – This byte value contains the baud rate of the controller on the network.

Diagnostic Counter Block of Communications Status File

With RSLogix 500/RSLogix Micro version 8.10.00 and later, formatted displays of the diagnostic counters for each configured channel are available under Channel Status. These displays include a Clear button that allows you to reset the diagnostic counters while monitoring them online with the programming software.

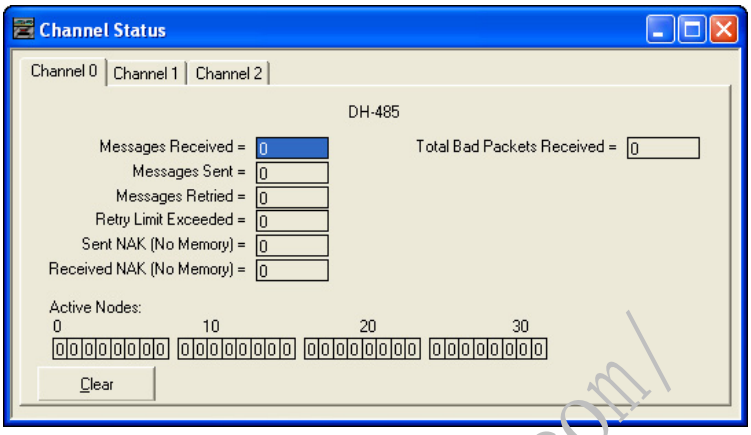
TIP Clicking the Clear button while online monitoring Channel Status of any channel will reset the channel status diagnostic counters for all three channels to zero.

Diagnostic Counter Blocks are shown for:

- DH-485 (on page 47)
- DF1 Full-Duplex (on page 49)
- DF1 Half-Duplex Slave (on page 50)
- DF1 Half-Duplex Master (on page 51)
- DF1 Radio Modem (on page 52)
- Modbus RTU Slave (on page 53)
- Modbus RTU Master (on page 54)
- ASCII (on page 56)
- DNP3 (on page 56)

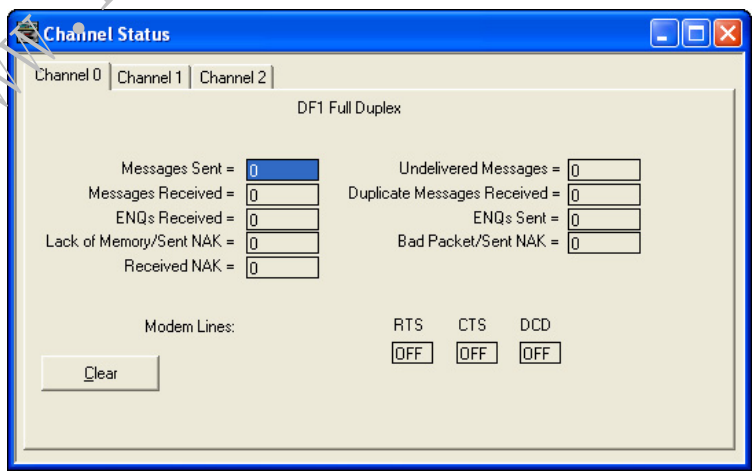
DH-485 Diagnostic Counters Block

Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 0)
9	-	Total Message Packets Received
10	-	Total Message Packets Sent
11	0…7	Message Packet Retries
	8…15	Retry Limit Exceeded (Non-Delivery)
12	0…7	NAK – No Memories Sent
	8…15	NAK – No Memories Received
13	0…7	Total Bad Message Packets Received
	8…15	Reserved
14…22	-	Reserved



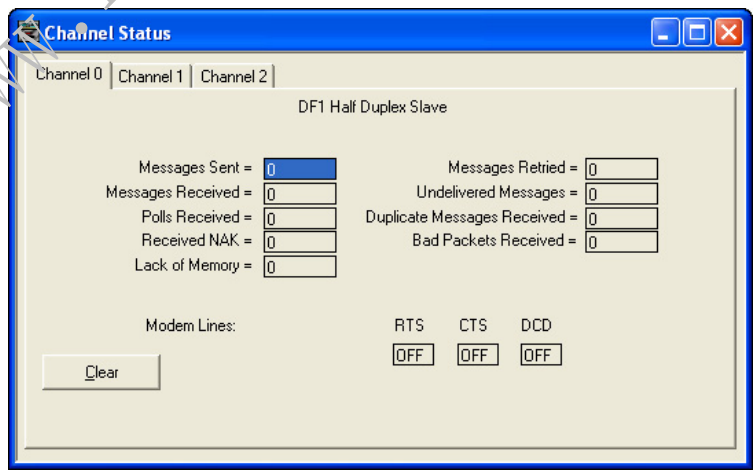
DF1 Full-Duplex Diagnostic Counters Block

Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 1)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	-	Total Message Packets Sent
11	-	Total Message Packets Received
12	-	Undelivered Message Packets
13	-	ENQuiry Packets Sent
14	-	NAK Packets Received
15	-	ENQuiry Packets Received
16	-	Bad Message Packets Received and NAKed
17	-	No Buffer Space and NAK'ed
18	-	Duplicate Message Packets Received
19…22	-	Reserved



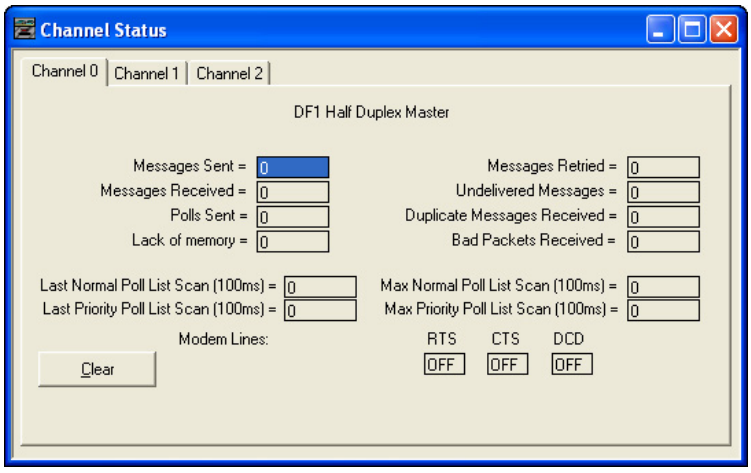
DF1 Half-Duplex Slave Diagnostic Counters Block

Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 2)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	-	Total Message Packets Sent
11	-	Total Message Packets Received
12	-	Undelivered Message Packets
13	-	Message Packets Retried
14	-	NAK Packets Received
15	-	Polls Received
16	-	Bad Message Packets Received
17	-	No Buffer Space
18	-	Duplicate Message Packets Received
19…22	-	Reserved



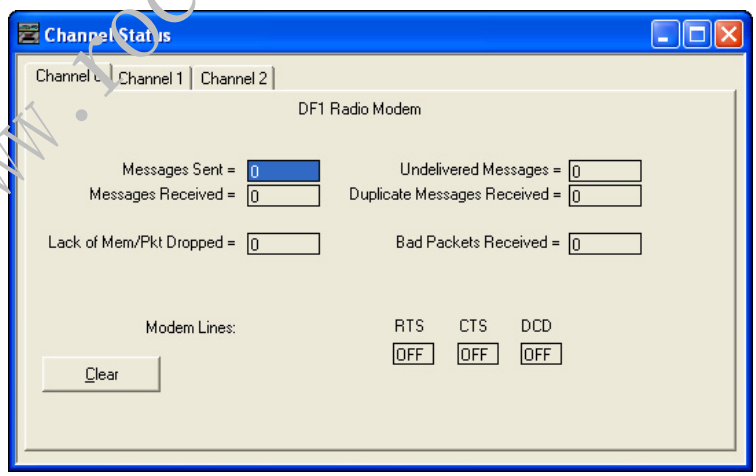
DF1 Half-Duplex Master Diagnostic Counters Block

Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 3)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	-	Total Message Packets Sent
11	-	Total Message Packets Received
12	-	Undelivered Message Packets
13	-	Message Packets Retried
14	-	Reserved
15	-	Polls Sent
16	-	Bad Message Packets Received
17	-	No Buffer Space, Received Packet Dropped
18	-	Duplicate Message Packets Received
19	-	Last Normal Poll List Scan
20	-	Max. Normal Poll List Scan
21	-	Last Priority Poll List Scan
22	-	Max. Priority Poll List Scan



DF1 Radio Modem Diagnostic Counters Block

Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 1)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	-	Total Message Packets Sent
11	-	Total Message Packets Received
12	-	Undelivered Message Packets
13…15	-	Reserved
16	-	Bad Message Packets Received
17	-	No Buffer Space, Received Packet Dropped
18	-	Duplicate Message Packets Received
19…22	-	Reserved



Modbus RTU Slave Diagnostic Counters Block (Data Link Layer)

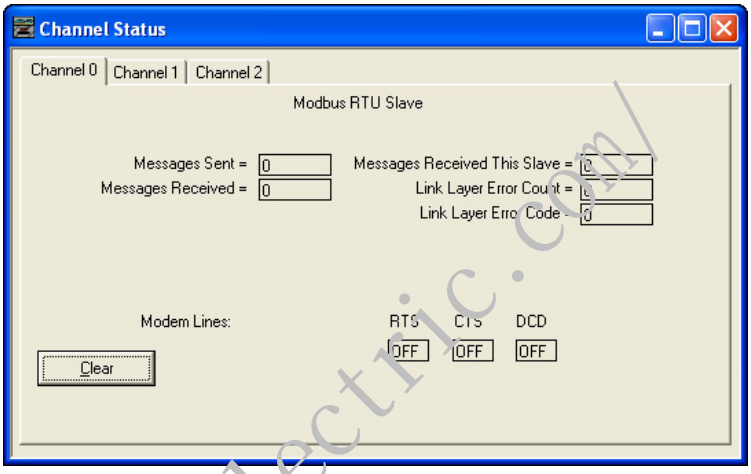
Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 4)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	-	Total Message Packets Sent
11	-	Total Message Packets Received for This Slave
12	-	Total Message Packets Received
13	-	Link Layer Error Count
14	-	Link Layer Error Code
15…22	-	Reserved

Modbus RTU Slave Diagnostic Counters Block (Presentation Layer)

Word	Bit	Description
43	-	Diagnostic Counters Category Identifier Code (always 10)
44	-	Length (always 14)
45	-	Format Code (always 0)
46	-	Pre-Send Time Delay
47	0…7	Node Address
	8…15	Reserved
48	-	Inter-Character Timeout
49	-	RTS Send Delay
50	-	RTS Off Delay
51	0…7	Baud Rate
	8 and 9	Parity
	10…15	Reserved
52	-	Diagnostic Counters Category Identifier Code (always 6)
53	-	Length (always 32)
54	-	Format Code (always 0)
55	-	Presentation Layer Error Code
56	-	Presentation Layer Error Count
57	-	Execution Function Error Code
58	-	Last Transmitted Exception Code
59	-	Data File Number of Error Request
60	-	Element Number of Error Request
61	-	Function Code 1 Message Counter
62	-	Function Code 2 Message Counter
63	-	Function Code 3 Message Counter
64	-	Function Code 4 Message Counter
65	-	Function Code 5 Message Counter

Modbus RTU Slave Diagnostic Counters Block (Presentation Layer)

Word	Bit	Description
66	-	Function Code 6 Message Counter
67	-	Function Code 8 Message Counter
68	-	Function Code 15 Message Counter
69	-	Function Code 16 Message Counter



Modbus RTU Master Diagnostic Counters Block (Data Link Layer)

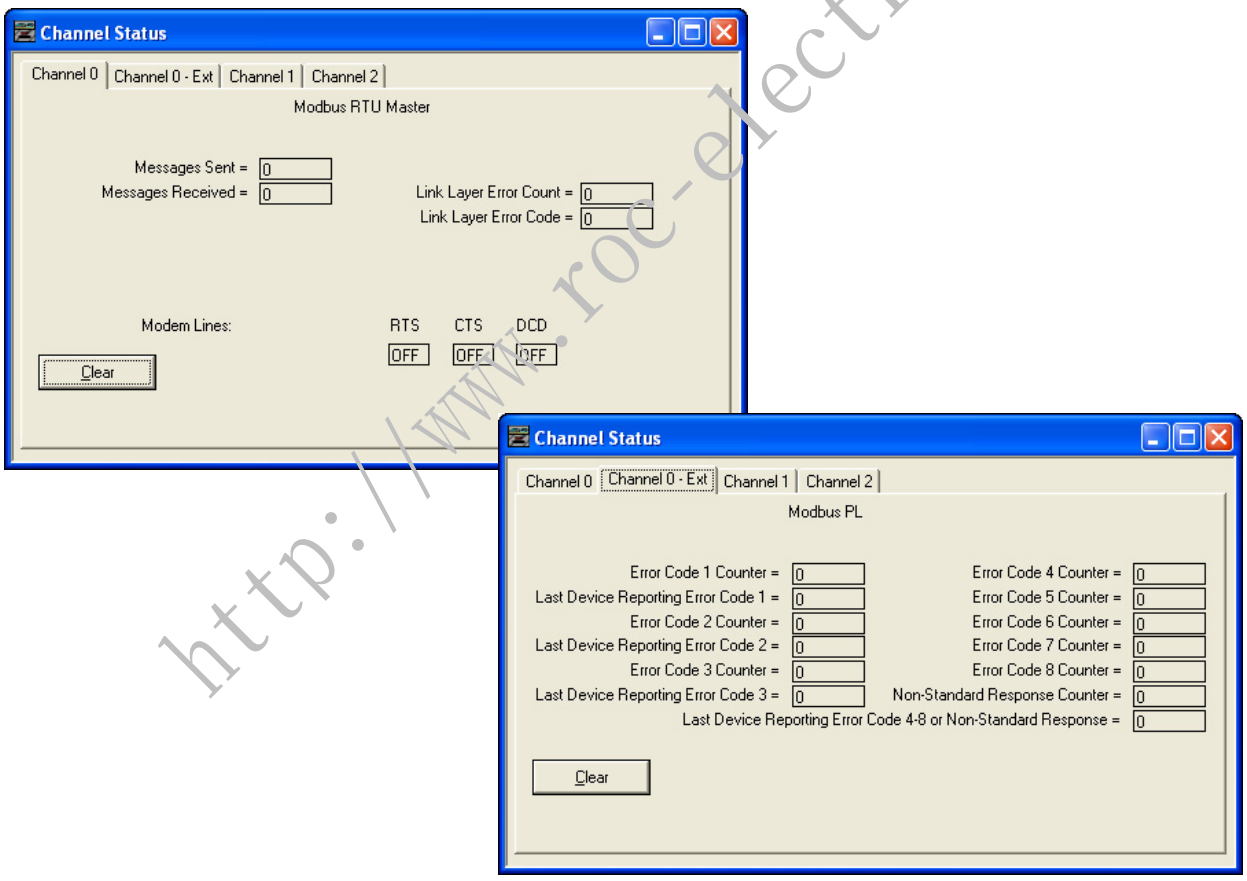
Word	Bit	Description
6	-	Diagnostic Counters Category Identifier Code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 9)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	-	Total Message Packets Sent
11	-	Reserved
12	-	Total Message Packets Received
13	-	Link Layer Error Count
14	-	Link Layer Error Code
15…22	-	Reserved

Modbus RTU Master Diagnostic Counters Block (Presentation Layer)

Word	Bit	Description
52	-	Diagnostic Counters Category Identifier Code (always 6)
53	-	Length (always 32)
54	-	Format Code (always 0)

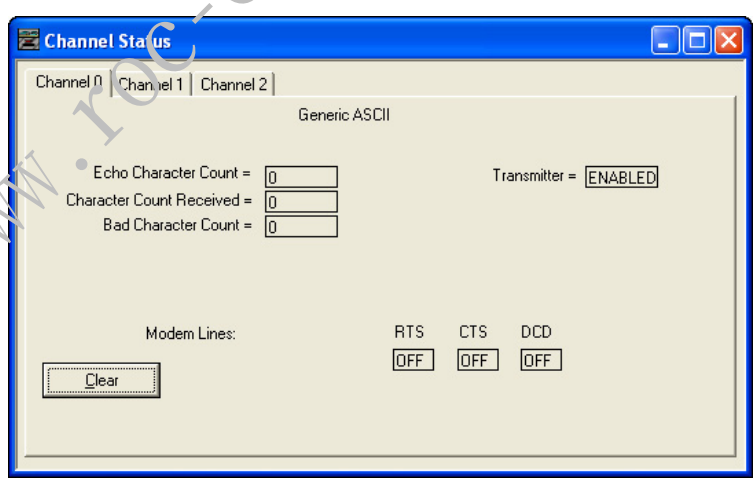
Modbus RTU Master Diagnostic Counters Block (Presentation Layer)

Word	Bit	Description
55	-	ERR 1: Illegal Function
56	-	Last Device Reporting ERR 1
57	-	ERR 2: Illegal Data Address
58	-	Last Device Reporting ERR 2
59	-	ERR 3: Illegal Data Value
60	-	Last Device Reporting ERR 3
61	-	ERR 4: Slave Device Failure
62	-	ERR 5: Acknowledge
63	-	ERR 6: Slave Device Busy
64	-	ERR 7: Negative Acknowledgement
65	-	ERR 8: Memory Parity Error
66	-	Non-Standard Response
67	-	Last Device Reporting ERR 4 to ERR 8 or Non-Standard Response
68 and 69	-	Reserved (always 0)



ASCII Diagnostic Counters Block

Word	Bit	Description
6	-	DLL Diagnostic Counters Category Identifier code (always 2)
7	-	Length (always 30)
8	-	Format Code (always 5)
9	0	CTS
	1	RTS
	2	Reserved
	3	Reserved
	4…15	Reserved
10	0	Software Handshaking Status
	1…15	Reserved
11	-	Echo Character Count
12	-	Received Character Count
13…18	-	Reserved
19	-	Bad Character Count
20…22	-	Reserved

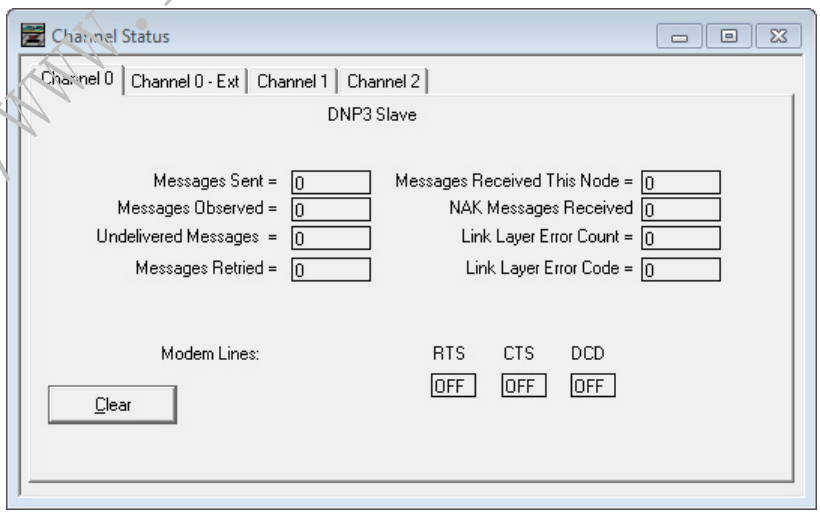


DNP3 Slave Diagnostic Counters Block (Data Link Layer)

Word	Bit	Description
6	-	DLL Diagnostic Counters Category Identifier code (2)
7	-	Length: 30 (15 words to follow including format code)
8	-	Counters Format Code: 11 - DNP3 Slave

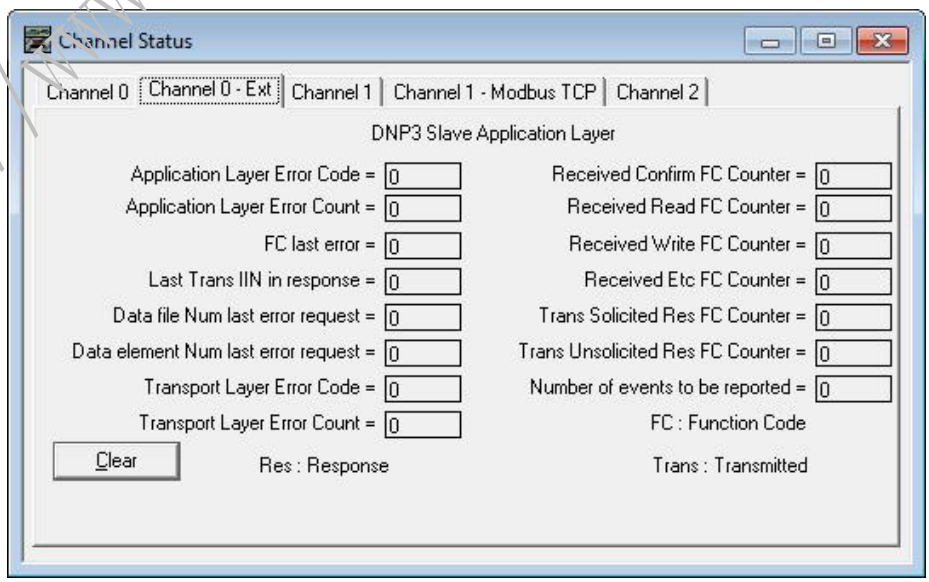
DNP3 Slave Diagnostic Counters Block (Data Link Layer)

Word	Bit	Description
9	15…4	Reserved Modem Control Line States - Always zero
	3	Channel 0 - DCD Channel 2 - DCD
	2	Reserved Modem Control Line States - Always zero
	1	RTS
	0	CTS
10	0	Total Message Packets Sent
11	-	Total Message Packets Received for this node
12	-	Total Packets Observed
13	-	Undelivered Message Packets
14	-	Message Packets Retried
15	-	NAK Packets Received
16	-	Link Layer Error Count
17	-	Link Layer Error Code
18	-	Reserved - Always zero
19	-	Reserved - Always zero
20	-	Reserved - Always zero
21	-	Reserved - Always zero
22	-	Reserved - Always zero



DNP3 Slave Diagnostic Counters Block (Application Layer)

Word	Bit	Description
52	-	PL Diagnostic Counters Category Identifier Code (6))
53	-	Length: 32 (16 words to follow including format code)
54	-	Category Block Format Code - 2
55	-	Application Layer Error Code
56	-	Application Layer Error Count
57	-	Function Code that caused the last error
58	-	Last Transmitted IIN in the response
59	-	Data File Number of Error Request
60	-	Element Number of Error Request
61	-	Received Confirm Function Code Counter
62	-	Received Read Function Code Counter
63	-	Received Write Function Code Counter
64	-	Received Etc Function Code Counter
65	-	Transmitted Solicited Response Function Code Counter
66	-	Transmitted Unsolicited Response Function Code Counter
67	-	Number of events to be reported
68	-	Transport layer error code
69	-	Transport layer error count

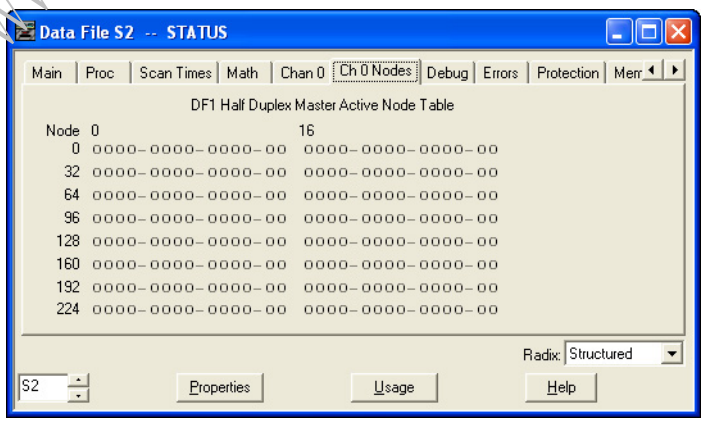


Active Node Table Block of Communications Status File

Active Node Table Block

Word	Description
23	Active Node Table Category Identifier Code (always 3)
24	Length: • always 4 for DH-485 • always 18 for DF1 Half-Duplex Master • always 0 for DF1 Full-Duplex, DF1 Half-Duplex Slave, Modbus RTU Slave, Modbus RTU Master, ASCII, and DNP3 Slave
25	Format Code (always 0)
26	Number of Nodes: • always 32 for DH-485 • always 255 for DF1 Half-Duplex Master • always 0 for DF1 Full-Duplex, DF1 Half-Duplex Slave, Modbus RTU Slave, Modbus RTU Master, ASCII, and DNP3 Slave
27	Active Node Table (DH-485 and DF1 Half-Duplex Master) – Nodes 0…15 (CS0:27/1 is node 1, CS0:27/2 is node 2, etc.) This is a bit-mapped register that displays the status of each node on the network. If a bit is set (1), the corresponding node is active on the network. If a bit is clear (0), the corresponding node is inactive.
28	Active Node Table (DH-485 and DF1 Half-Duplex Master) – Nodes 16…31 (CS0:28/1 is node 16, CS0:28/2 is node 17, etc.)
29	Active Node Table (DF1 Half-Duplex Master) – Nodes 32…47 (CS0:29/1 is node 32, CS0:29/2 is node 33, etc.)
⋮	
42	Active Node Table (DF1 Half-Duplex Master) – Nodes 240…255 (CS0:42/1 is node 240, CS0:42/2 is node 241, etc.)

If you are using RSLogix 500/RSLogix Micro version 8.10.00 or higher, you can view the active node table by clicking on "Processor Status" and then selecting the tab for the configured channel.



Ethernet Communications Status File

The Ethernet Communications Status (ES) File is a read-only file that contains information on how the controller Ethernet communication parameters are configured and status information on Ethernet communications activity.

The Ethernet communications status file uses 178 1-word elements.

TIP You can use the Ethernet Communications Status File information as a troubleshooting tool for Ethernet communications issues.

The data file is structured as:

Communications Status File

Word	Description	Applies to Controller	Details on Page
0…119	General Channel Status Block	MicroLogix 1400	45
120…176	DLL Diagnostic Counters Block	MicroLogix 1400	47
177	End of List Category Identifier Code (always 0)	MicroLogix 1400	59

The following tables show the details of each block in the Ethernet Communications Status File.

General Status Block of Ethernet Communications Status File

General Channel Status Block

Word	Bit	Description
0	-	Communications Channel General Status Information Category Identifier Code (1)
1	-	Length: 236
2	-	Format Code
3	-	Communications Configuration Error Code

General Channel Status Block

Word	Bit	Description
4	0	ICP – Incoming Command Pending Bit This bit is set (1) when the controller determines that another device has requested information from this controller. Once the request has been satisfied, the bit is cleared (0).
	1	MRP – Incoming Message Reply Pending Bit This bit is set (1) when the controller determines that another device has supplied the information requested by a MSG instruction executed by this controller. When the appropriate MSG instruction is serviced (during end-of-scan, SVC, or REF), this bit is cleared (0).
	2	MCP – Outgoing Message Command Pending Bit This bit is set (1) when the controller has one or more MSG instructions enabled and in the communication queue. This bit is cleared (0) when the queue is empty.
	3 to 4	Reserved – Always zero
	5	SNMP Server Status This bit is set (1) when the SNMP server is enabled. The cleared bit (0) means that the SNMP server is disabled.
	6	HTTP Server Status This bit is set (1) when the internal web server is enabled. The cleared bit (0) means that the internal web server is disabled.
	7	SMTP Client Status This bit is set (1) when the SMTP client (for email) is enabled. The cleared bit (0) means that the SMTP client is disabled.
	8…15 (Series A)	Reserved – Always zero
	8…11 (Series B)	Reserved – Always zero
	12 (Series B	Modbus TCP Status This bit is set (1) when the Modbus TCP Server/Client feature is enabled. The cleared bit (0) means that the Modbus TCP Server/Client is disabled.
	13 (Series B	DNP3 over IP Status This bit is set (1) when the DNP3 over IP feature is enabled. The cleared bit (0) means that the DNP3 over IP feature is disabled.
	14 (Series B	Reserved – Always zero
	15 (Series B	Disable Ethernet/IP Incoming Connection Status This bit is set (1) when the Ethernet/IP Incoming connection is not allowed. The cleared bit (0) means that the Incoming connection is allowed.

General Channel Status Block

Word	Bit	Description
5	0	Ethernet Port Link Status This bit is set (1) when the Ethernet link is active.
	1	Ethernet Port Connection Speed This bit is valid when the Auto Negotiation function is enabled. This bit indicates the speed of the link layer driver operating at Ethernet port: • 0: 10 Mbps • 1: 100 Mbps
	2	Reserved – Always zero
	3	Duplex Mode This bit is valid when the Auto Negotiation function is enabled. This bit indicates the duplex mode of the Ethernet port: • 0: Half Duplex • 1: Full Duplex
	4	Auto Negotiate Status This bit is set (1) when the Auto Negotiation function is enabled.
	5	Forced Speed Mode Status This bit set (1) when the Auto Negotiation function is disabled and the Ethernet port speed is 100Mbps.
	6	Forced Duplex Mode Status This bit set (1) when the Auto Negotiation function is disabled and the Ethernet port's duplex mode is Full Duplex.
	7	Reserved - Always zero
	8	BOOTP Valid Flag (Default: 0, False) This bit is set (1) when the appropriate BOOTP response has been received. If BOOTP Enable Flag in Ethernet Port Communications Configuration File is set (1, Yes) and this flag is cleared (0, False), then network-related information is invalid.
	9	DHCP Valid Flag (Default: 0, False) This bit is set (1) when the appropriate DHCP response has been received. If DHCP Enable Flag in Ethernet Port Communications Configuration File is set (1, Yes) and this flag is cleared (0, False), then network-related information is invalid.
	10	BOOTP Status Flag This bit is set (1) if BOOTP is selected as configuration method.
	11	DHCP Status Flag This bit is set (1) if DHCP is selected as configuration method.

General Channel Status Block

Word	Bit	Description
5	12	Advertise 100 MB Full Duplex Flag This bit indicates advertisement status if Auto negotiate enabled: • 0: 100 MB Full Duplex was not advertised during auto negotiation • 1: 100 MB Full Duplex was advertised during auto negotiation
	13	Advertise 100 MB Half Duplex Flag This bit indicates advertisement status if Auto negotiate enabled: • 0: 100 MB Half Duplex was not advertised during auto negotiation • 1: 100 MB Half Duplex was advertised during auto negotiation
	14	Advertise 10 MB Full Duplex Flag This bit indicates advertisement status if Auto negotiate enabled: • 0: 10 MB Full Duplex was not advertised during auto negotiation • 1: 10 MB Full Duplex was advertised during auto negotiation
	15	Configuration End Flag This bit is set (1) when the Ethernet boot-up sequence is completed, including IP address, gateway address, subnet mask and etc.
6…8	-	Ethernet Hardware Address (6-byte string) A unique Ethernet hardware address assigned to this processor.
9…10	-	IP Address (in network byte order) Internet address that is specified for this processor.
11…12	-	Subnet Mask (in network byte order) Subnet mask that is specified for this processor.
13…14	-	Gateway Address (in network byte order) Gateway address that is specified for this processor
15…16	-	Broadcast Address (in network byte order) NOT SUPPORTED AT THIS TIME. The Broadcast Address is used in sending multicast messages. A Broadcast Address of all zeros indicates that no broadcast address was configured. In this case, the network code chooses a valid broadcast address when needed for that current subnet.
17…18	-	Primary Name Server(in network byte order) Primary Name Server that is specified for this processor.
19…20	-	Secondary Name Server(in network byte order) Secondary Name Server that is specified for this processor.
21…52	-	Default Domain Name Default domain name that is specified for this processor.
53…84	-	SNMP Contact Contact string that is specified for this processor.
85…116	-	SNMP Location Location string that is specified for this processor.
117	-	Message Connection Timeout The amount of time (in ms) allowed for a MSG instruction to establish a connection with the destination node. The MSG Connection Timeout has a range of 250 ms...65,500 ms.
118	-	Message Reply Timeout The amount of time (in ms) that the MicroLogix 1400 processor waits for a reply to a command that it has initiated via a MSG instruction. The MSG Reply Timeout has a range of 250 ms...65,500 ms.
119	-	Inactivity Timeout The amount of time (in minutes) that a MSG connection may remain inactive before it is terminated. The Inactivity Timeout has a 1 minute resolution and a range of 1...65,500 minutes.

Diagnostic Counter Block of Communications Status File

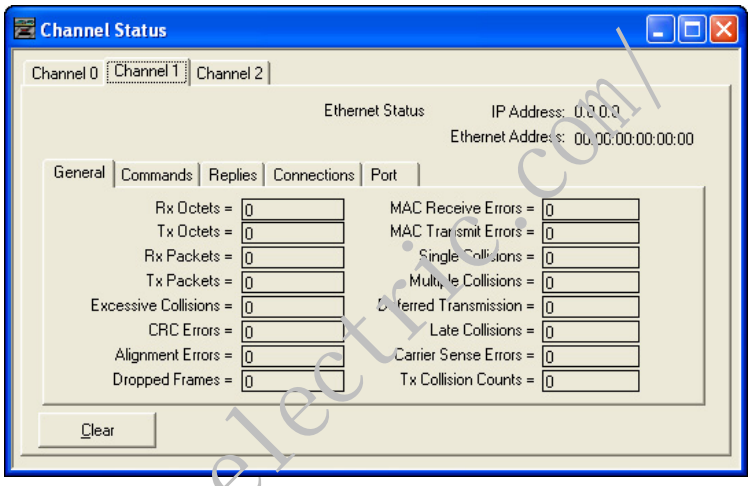
With RSLogix 500/RSLogix Micro version 8.10.00 and later, formatted displays of the diagnostic counters for Ethernet communications channel are available under Channel Status. These displays include a Clear button that allows you to reset the diagnostic counters while monitoring them online with the programming software.

Ethernet Diagnostic Counters Block

Word	Bit	Description
120	-	DLL Diagnostic Counters Category Identifier Code (always 2)
121	-	Length: 110 (55 words to follow including format code)
122	-	Counters Format Code: Ethernet (always 0)
123 124	Low word High word	RMON Rx Octets (RMON_R_OCTETS)
125 126	Low word High word	RMON Tx Octets (RMON_T_OCTETS)
127 128	Low word High word	RMON Rx Packets (RMON_R_PACKETS)
129 130	Low word High word	RMON Tx Packets (RMON_T_PACKETS)
131 132	Low word High word	Frames Transmitted with Excessive Collisions (IEEE_T_EXCOL)
133 134	Low word High word	Frames Received with CRC Error (IEEE_R_CRC)
135 136	Low word High word	Frames Received with Alignment Error (IEEE_R_ALIGN)
137 138	Low word High word	Count of frames not counted correctly (RMON_T_DROP)
139 140	Low word High word	Receive FIFO Overflow Count (IEEE_R_MACERR)
141 142	Low word High word	Frames transmitted with Tx FIFO Under-run (IEEE_T_MACERR)
143 144	Low word High word	Frames Transmitted with Single Collision (IEEE_T_1COL)
145 146	Low word High word	Frames Transmitted with Multiple Collisions (IEEE_T_MCOL)
147 148	Low word High word	Frames Transmitted with Deferral Delay (IEEE_T_DEF)
149 150	Low word High word	Frames Transmitted with Late Collisions (IEEE_T_LCOL)

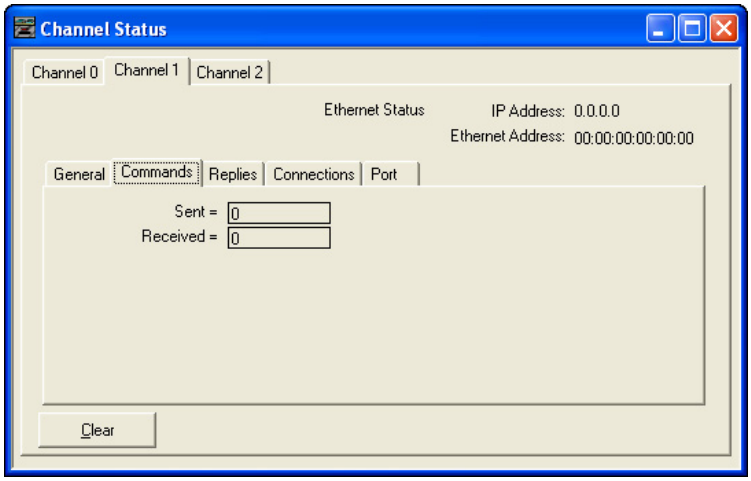
Ethernet Diagnostic Counters Block

Word	Bit	Description
151 152	Low word High word	Frames Transmitted with Carrier Sense Errors (IEEE_T_CSERR)
153 154	Low word High word	RMON Tx Collision Count (RMON_T_COL)



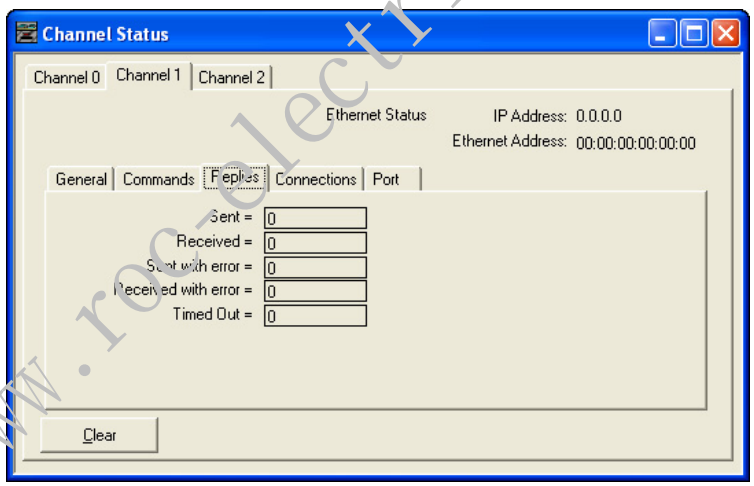
Ethernet Diagnostic Counters Block (continued)

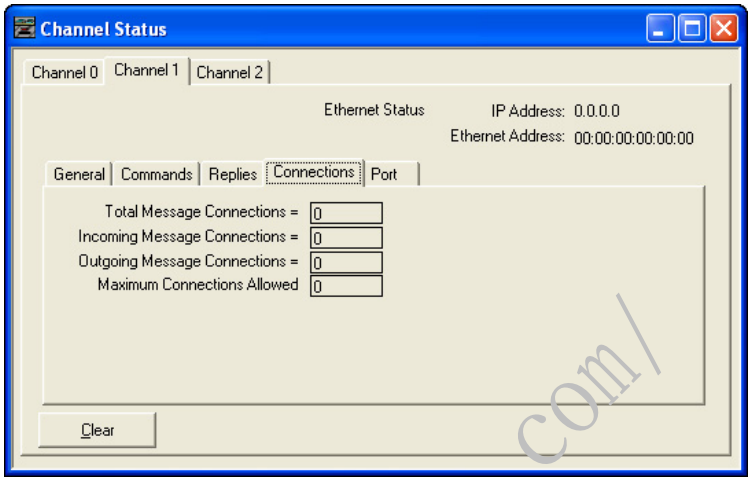
Word	Bit	Description
155 156	Low word High word	Total Commands Sent
157 158	Low word High word	Total Commands Received



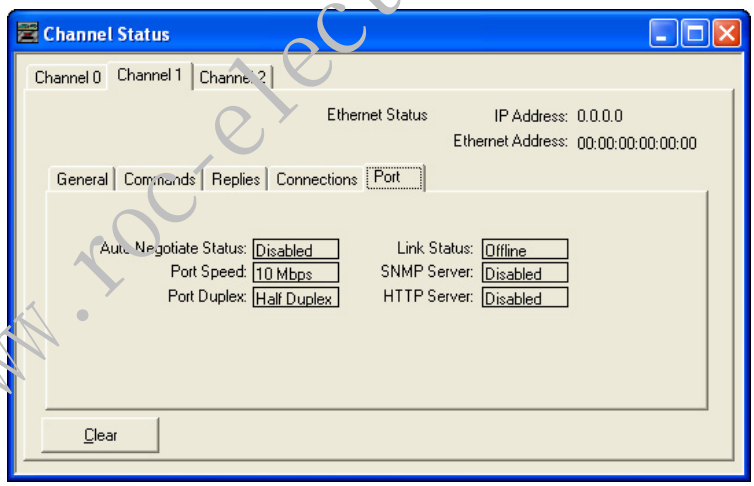
Ethernet Diagnostic Counters Block (continued)

Word	Bit	Description
159 160	Low word	Total Replies Sent
	High word	
161 162	Low word	Total Replies Received
	High word	
163 164	Low word	Total Replies Sent with Error
	High word	
165 166	Low word	Total Replies Received with Error
	High word	
167 168	Low word	Total Replies Timed Out
	High word	





The last Port tab will show the current states of Ethernet communications port according to word 5 of Ethernet Communications Status File.



Input/Output Status File

The input/output status (IOS) file is a read-only file in the controller that contains information on the status of the embedded and local expansion I/O. The data file is structured as:

Input/Output Status File

Word	Description
0	Embedded Module Error Code – Always zero
1…4	Expansion Module Error Code – The word number corresponds to the module's slot number. Refer to the I/O module's documentation for specific information. *(MicroLogix 1400)*

Notes:

Programming Instructions Overview

Instruction Set

The following table shows the MicroLogix 1400 programming instructions listed within their functional group.(1)

Functional Group	Description	Page
High-Speed Counter	HSL, RAC — The high-speed counter instructions (along with the HSC function file) allow you to monitor and control the high-speed counter. Generally used with DC inputs.	77
High-Speed Outputs	PTOX, PWMX — The high-speed output instructions (along with the PTOX and PWMX function files) allow you to monitor and control the high-speed outputs. Generally used with FET outputs BXB and BXBA units).	111
Relay-Type (Bit)	XIC, XIO, OTE, OTL, OTU, OSR, ONS, OSF — The relay-type (bit) instructions monitor and control the status of bits.	137
Timer and Counter	TON, TOF, RTO, CTU, CTD, RES — The timer and counter instructions control operations based on time or the number of events.	145
Compare	EQU, NEQ, LES, LEQ, GRT, GEQ, MEQ, LIM — The compare instructions compare values by using a specific compare operation.	155
Math	ADD, SUB, MUL, DIV, NEG, CLR, ABS, SQR, SCL, SCP, SWP, CPT, COS, ATN, ASN, ACS, SIN, TAN, XPY, LN, LOG, DEG, RAD — The math instructions perform arithmetic operations.	161
Application Specific	RHC, RPC, TDF — The instructions aid in calculating performance diagnostics.	199
Conversion	DCD, ENC, TOD, FRD, GCD — The conversion instructions multiplex and de-multiplex data and perform conversions between binary and decimal values.	205
Logical	AND, OR, XOR, NOT — The logical instructions perform bit-wise logical operations on words.	215
Move	MOV, MVM — The move instructions modify and move words.	219
File	CPW, COP, FLL, BSL, BSR, FFL, FFU, LFL, LFU, SWP — The file instructions perform operations on file data.	225
Sequencer	SQC, SQO, SQL — Sequencer instructions are used to control automatic assembly machines that have consistent and repeatable operations.	243
Program Control	JMP, LBL, JSR, SBR, RET, SUS, TND, MCR, END — The program flow instructions change the flow of ladder program execution.	253
Input and Output	IIM, IOM, REF — The input and output instructions allow you to selectively update data without waiting for the input and output scans.	259
User Interrupt	STS, INT, UID, UIE, UIF — The user interrupt instructions allow you to interrupt your program based on defined events.	263
Process Control	PID — The process control instruction provides closed-loop control.	281
ASCII	ABL, ACB, ACI, ACL, ACN, AEX, AHL, AIC, ARD, ARL, ASC, ASR, AWA, AWT — The ASCII instructions convert and write ASCII strings.	309

(1) The Memory Module Information Function File appears on page 42 following the Real-Time Clock Function File information.

Functional Group	Description	Page
Communications	MSG, SVC — The communication instructions read or write data to another station.	337
Recipe	RCP — The recipe instruction allows you to transfer a data set between the recipe database and a set of user-specified data table elements.	465
Data Logging	DLG — The data logging instruction allow you to capture time-stamped and date-stamped data.	465
LCD	LCD - The LCD instruction transfers data from a data file to the LCD and receives a value from the LCD keypad.	485

Using the Instruction Descriptions

Throughout this manual, each instruction (or group of similar instructions) has a table similar to the one shown below. This table provides information for all sub-elements (or components) of an instruction or group of instructions. This table identifies the type of compatible address that can be used for each sub-element of an instruction or group of instructions in a data file or function file. The definitions of the terms used in these tables are listed below this example table.

Valid Addressing Modes and File Types - Example Table

Parameter	Data Files														Function Files									CS - Comms	IOS - I/O	DLS - Data Log	Address Mode (1)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	ASCII	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Source B	•	•	•	•	•	•	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•	•	•	•	•	•		•		•	•	•	•	•								•	•		•	•	

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

The terms used within the table are defined as follows:

- Parameter – The parameter is the information you supply to the instruction. It can be an address, a value, or an instruction-specific parameter such as a timebase.
- Data Files – See Data Files on page 26.
- Function Files – See Function Files on page 37.
- CS – See Communications Status File on page 45.
- IOS – See Input/Output Status File on page 67.
- DLS – See Data Log Status File on page 479.
- Address Mode – See Addressing Modes on page 71.

- Addressing Level – Address levels describe the granularity at which an instruction allows an operand to be used. For example, relay type instructions (XIC, XIO, and so on) must be programmed to the bit level, timer instructions (TON, TOF, and so on) must be programmed to the element level (timers have 3 words per element) and math instructions (ADD, SUB, and so on) must be programmed to the word or long word level.

Addressing Modes

The MicroLogix 1400 supports three types of data addressing:

- Immediate
- Direct
- Indirect

The MicroLogix 1400 do not support indexed addressing. Indexed addressing can be duplicated with indirect addressing. See Example – Using Indirect Addressing to Duplicate Indexed Addressing on page 74.

How or when each type is used depends on the instruction being programmed and the type of element specified within the operands of the instruction. By supporting these three addressing methods, the MicroLogix 1400 allows incredible flexibility in how data can be monitored or manipulated. Each of the addressing modes are described below.

Immediate Addressing

Immediate addressing is primarily used to assign numeric constants within instructions. For example: You require a 10 second timer, so you program a timer with a 1 second time base and a preset value of 10. The numbers 1 and 10 in this example are both forms of immediate addressing.

Direct Addressing

When you use direct addressing, you define a specific data location within the controller. Any data location that is supported by the elements of an operand within the instruction being programmed can be used. In this example we are illustrating a limit instruction, where:

- Low Limit = Numeric value (from -32,768...32,767) entered from the programming software.
- Test Value = LCD:0.POT0 (This is the current position/value of trim pot 0.)
- High Limit = N7:17 (This is the data resident in Integer file 7, element 17.)

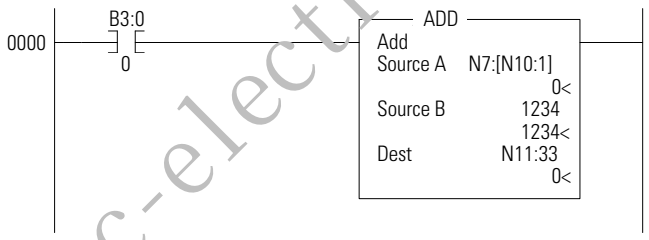
The Test Value (LCD:0.POT0) and High Limit (N7:17) are direct addressing examples. The Low Limit is immediate addressing.

Indirect Addressing

Indirect addressing allows components within the address to be used as pointers to other data locations within the controller. This functionality can be especially useful for certain types of applications, recipe management, batch processing and many others. Indirect addressing can also be difficult to understand and troubleshoot. It is recommended that you only use indirect addressing when it is required by the application being developed.

The MicroLogix 1400 supports indirection (indirect addressing) for Files, Words and Bits. To define which components of an address are to be indirected, a closed bracket "[ ]" is used. The following examples illustrate how to use indirect addressing.

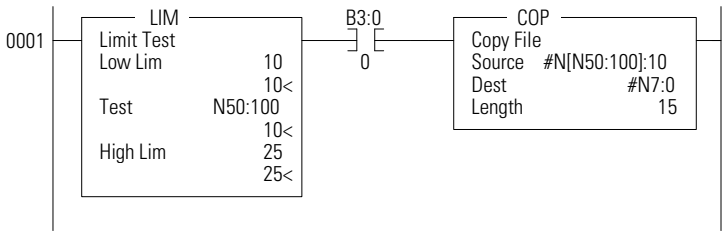
Indirect Addressing of a Word



- Address: N7:[N10:1]
- In this example, the element number to be used for source A in the ADD instruction is defined by the number located in N10:1. If the value of location N10:1 = 15, the ADD instruction operates as "N7:15 + Source B".
- In this example, the element specified by N10:1 must be between 0 and 255, because all data files have a maximum individual size of 256 elements.

TIP If a number larger than the number of elements in the data file is placed in N10:1 (in this example), data integrity cannot be guaranteed, because a file boundary will be crossed. This may not generate a controller fault, but the data location is invalid/unknown.

Indirect Addressing of a File



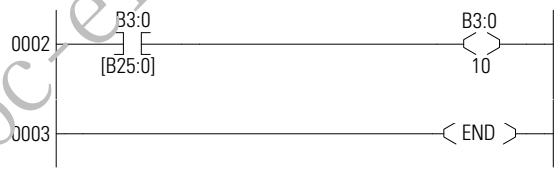
- Address: N[N50:100]:10

- Description: In this example, the source of the COP instruction is indirected by N50:100. The data in N50:100 defines the data file number to be used in the instruction. In this example, the copy instruction source A is defined by N[N50:100]:10. When the instruction is scanned, the data in N50:100 is used to define the data file to be used for the COP instruction. If the value of location N50:100 = 27, this instruction copies 15 elements of data from N27:10 (N27:10 to N27:24) to N7:0 (N7:0 to N7:14)

TIP If a number larger than 255 is placed in N50:100 in this example, a controller fault occurs. This is because the controller has a maximum of 255 data files. In addition, the file defined by the indirection should match the file type defined by the instruction, in this example an integer file.

TIP This example also illustrates how to perform a limit check on the indirect address. The limit instruction at the beginning of the rung is monitoring the indirect element. If the data at N50:100 is less than 10 or greater than 25, the copy instruction is not processed. This procedure can be used to make sure an indirect address does not access data an unintended location.

Indirect Addressing of Bit



- Address: B3/[B25:0]
- Description: In this example, the element to be used for the indirection is B25:0. The data in B25:0 defines the bit within file B3. If the value of location B25:0 = 1017, the XIC instruction is processed using B3/1017.

TIP If a number larger than 4096 (or larger than the number of elements in the data file) is placed in B25:0 in this example, data integrity cannot be guaranteed. Exceeding the number of elements in the data file would cause the file boundary to be crossed.

These are only some of the examples that can be used; others include:

- File and Element Indirection: N[N10:0]:[N25:0]
- Input Slot Indirection: I1:[N7:0].0

Each group of instructions may or may not allow indirection. Please review the compatibility table for each instruction to determine which elements within an instruction support indirection.

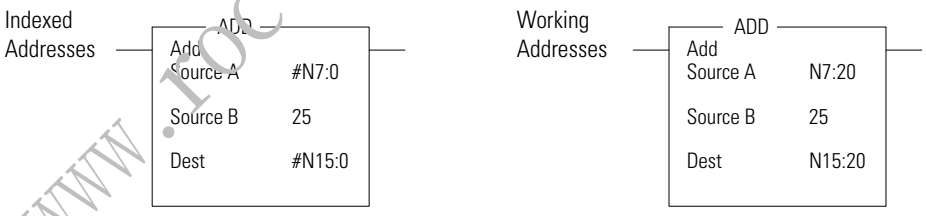
IMPORTANT	You must exercise extreme care when using indirect addressing. Always be aware of the possibility of crossing file boundaries or pointing to data that was not intended to be used.

Example – Using Indirect Addressing to Duplicate Indexed Addressing

In this section, an indexed addressing example is shown first. Then an equivalent indirect addressing example is shown. Indexed addressing is supported by SLC 500 and MicroLogix 1000 programmable controllers. The MicroLogix 1100, 1200, 1400, and 1500 do not support indexed addressing. This example is shown for comparison purposes.

Indexed Addressing Example

The following ADD instruction uses an indexed address in the Source A and Destination addresses. If the indexed offset value is 20 (stored in S:24), the controller uses the data stored at the base address plus the indexed offset to perform the operation.



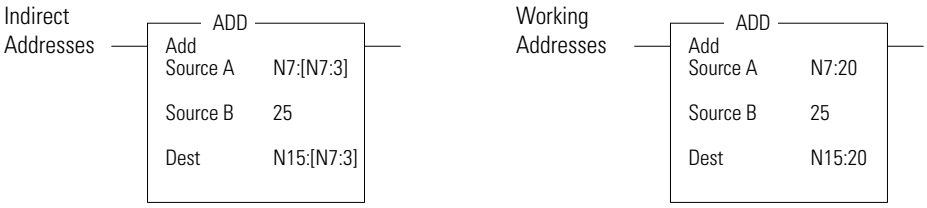
In this example, the controller uses the following addresses:

Operand	Base Address	Offset Value in S:24	Working Address
Source A	N7:0	20	N7:20
Destination	N15:0	20	N15:20

Indirect Addressing Example

An equivalent example using indirect addressing is shown below. In place of using the index register, S:24, the user can designate any other valid word address as the indirect address. Multiple indirect addresses can be used within an instruction.

The following ADD instruction uses an indirect address in the Source A and Destination addresses. If the indirect offset value is 20 (stored in N7:3), the controller uses the data stored at the base address plus the indirect offset to perform to instruction.



In this example, the controller uses the following addresses:

Operand	Base Address	Offset Value in N7:3	Working Address
Source A	N7:0	20	N7:20
Destination	N7:0	20	N15:20

Notes:

Using the High-Speed Counter and Programmable Limit Switch

High-Speed Counter Overview

All MicroLogix 1400, except the 1766-L32AWA and 1766_L32AWAA, have six 100kHz high-speed counters. There are three main high-speed counters (counter 0, 1, 2) and three sub high speed counters (counter 3, 4, 5). Each main high-speed counter has four dedicated inputs and each sub high-speed counter has two dedicated inputs. HSC0 utilizes inputs 0...3, HSC1 utilizes inputs 4...7, HSC2 utilizes inputs 8...11, HSC3 utilizes inputs 2 and 3, HSC4 utilizes inputs 6 and 7 and HSC5 utilizes inputs 10 and 11. In some cases, a sub counter will be disabled by master counter mode. See the section HSC Mode (MOD) on page 91.

TIP HSC0 is used in this document to define how any HSC works.

IMPORTANT The HSC function can only be used with the controller's embedded I/O. It cannot be used with expansion I/O modules.

This chapter describes how to use the HSC function and also contains sections on the HSL and RAC instructions, as follows:

- High-Speed Counter (HSC) Function File on page 78.
- HSL - High-Speed Counter Load on page 103.
- RAC - Reset Accumulated Value on page 104.

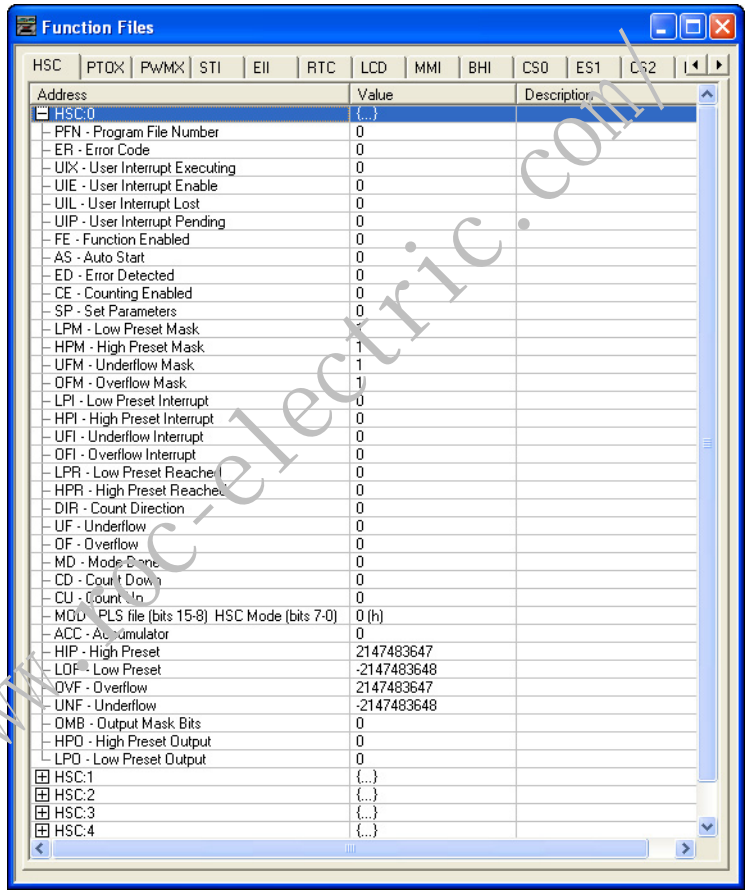
Programmable Limit Switch Overview

The Programmable Limit Switch function allows you to configure the High-Speed Counter to operate as a PLS (programmable limit switch) or rotary cam switch. See page 105 for more information.

High-Speed Counter (HSC) Function File

Within the RSLogix 500/RSLogix Micro Function File Folder, you see a HSC Function File. This file provides access to HSC configuration data, and also allows the control program access to all information pertaining to the High-Speed Counter.

TIP If the controller is in the run mode, the data within sub-element fields may be changing.

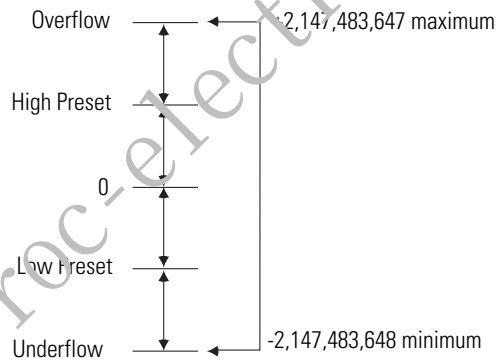


The HSC function, along with the PTOX and PWMX instructions, are different than most other controller instructions. Their operation is performed by custom circuitry that runs in parallel with the main system processor. This is necessary because of the high performance requirements of these functions.

The HSC is extremely versatile; the user can select or configure the master HSC for any one of ten (10) modes and the sub HSC for any one of five (5) modes of operation.
Operating Modes are discussed later in this chapter. See the section HSC Mode (MOD) on page 91. Some of the enhanced capabilities of the High-Speed Counters are:

- 100 kHz operation
- High-speed direct control of outputs
- 32-bit signed integer data (count range of ± 2,147,483,647)
- Programmable High and Low presets, and Overflow and Underflow setpoints
- Automatic Interrupt processing based on accumulated count
- Run-time editable parameters (from the user control program)

The High-Speed Counter function operates as described in the following diagram.



High-Speed Counter Function File Sub-Elements Summary

The HSC is comprised of 36 sub-elements. These sub-elements are either bit, word, or long word structures that are used to provide control over the HSC function, or provide HSC status information for use within the control program. Each of the sub-elements and their respective functions are described in this chapter. A summary of the sub-elements is provided in the following table. All examples illustrate HSC0.

High-Speed Counter Function File (HSC:0, HSC:1, HSC:2, HSC3, HSC4 or HSC5)

Sub-Element Description	Address	Data Format	HSC Modes[1]	Function	User Program Access	For More Information
PFN - Program File Number	HSC:0.PFN	word (INT)	0…9	control	read only	81
ER - Error Code	HSC:0.ER	word (INT)	0…9	status	read only	81
UIX - User Interrupt Executing	HSC:0/UIX	bit	0…9	status	read only	84
UIE - User Interrupt Enable	HSC:0/UIE	bit	0…9	control	read/write	84
UIL - User Interrupt Lost	HSC:0/UIL	bit	0…9	status	read/write	85
UIP - User Interrupt Pending	HSC:0/UIP	bit	0…9	status	read only	85
FE - Function Enabled	HSC:0/FE	bit	0…9	control	read/write	82
AS - Auto Start	HSC:0/AS	bit	0…9	control	read only	82
ED - Error Detected	HSC:0/ED	bit	0…9	status	read only	82
CE - Counting Enabled	HSC:0/CE	bit	0…9	control	read/write	83
SP - Set Parameters	HSC:0/SP	bit	0…9	control	read/write	83
LPM - Low Preset Mask	HSC:0/LPM	bit	2…9	control	read/write	85
HPM - High Preset Mask	HSC:0/HPM	bit	0…9	control	read/write	86
UFM - Underflow Mask	HSC:0/UFM	bit	2…9	control	read/write	88
OFM - Overflow Mask	HSC:0/OFM	bit	0…9	control	read/write	89
LPI - Low Preset Interrupt	HSC:0/LPI	bit	2…9	status	read/write	86
HPI - High Preset Interrupt	HSC:0/HPI	bit	0…9	status	read/write	87
UFI - Underflow Interrupt	HSC:0/UFI	bit	2…9	status	read/write	88
OFI - Overflow Interrupt	HSC:0/OFI	bit	0…9	status	read/write	89
LPR - Low Preset Reached	HSC:0/LPR	bit	2…9	status	read only	86
HPR - High Preset Reached	HSC:0/HPR	bit	2…9	status	read only	87
DIR - Count Direction	HSC:0/DIR	bit	0…9	status	read only	90
UF - Underflow	HSC:0/UF	bit	0…9	status	read/write	87
OF - Overflow	HSC:0/OF	bit	0…9	status	read/write	89
MD - Mode Done	HSC:0/MD	bit	0 or 1	status	read/write	90
CD - Count Down	HSC:0/CD	bit	2…9	status	read only	90
CU - Count Up	HSC:0/CU	bit	0…9	status	read only	90
MOD - HSC Mode	HSC:0.MOD	word (INT)	0…9	control	read only	91
ACC - Accumulator	HSC:0.ACC	long word (32-bit INT)	0…9	control	read/write	99
HIP - High Preset	HSC:0.HIP	long word (32-bit INT)	0…9	control	read/write	99
LOP - Low Preset	HSC:0.LOP	long word (32-bit INT)	2…9	control	read/write	99
OVF - Overflow	HSC:0.OVF	long word (32-bit INT)	0…9	control	read/write	100

High-Speed Counter Function File (HSC:0, HSC:1, HSC:2, HSC3, HSC4 or HSC5)

Sub-Element Description	Address	Data Format	HSC Modes(1)	Function	User Program Access	For More Information
UNF - Underflow	HSC:0.UNF	long word (32-bit INT)	2…9	control	read/write	100
OMB - Output Mask Bits	HSC:0.OMB	word (16-bit binary)	0…9	control	read only	101
HPO - High Preset Output	HSC:0.HPO	word (16-bit binary)	0…9	control	read/write	101
LPO - Low Preset Output	HSC:0.LPO	word (16-bit binary)	2…9	control	read/write	102

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.
n/a = not applicable

HSC Function File Sub-Elements

All examples illustrate HSC0.

Program File Number (PFN)

Description	Address	Data Format	HSC Modes(1)	Type	User Program Access
PFN - Program File Number	HSC:0.PFN	word (INT)	0…9	control	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The PFN (Program File Number) variable defines which subroutine is called (executed) when HSC0 counts to High Preset or Low Preset, or through Overflow or Underflow. The integer value of this variable defines which program file will run at that time. A valid subroutine file is any program file (3...255).

Error Code (ER)

Description	Address	Data Format	HSC Modes(1)	Type	User Program Access
ER - Error Code	HSC:0.ER	word (INT)	0…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The ERs (Error Codes) detected by the HSC sub-system are displayed in this word. Errors include:

HSC Error Codes

Error Code	Name	Mode(1)	Description
1	Invalid File Number	n/a	Interrupt (program) file identified in HSC:0.PFN is less than 3, greater than 255, or does not exist
2	Invalid Mode	n/a	Invalid Mode(1)

HSC Error Codes

Error Code	Name	Mode[1]	Description
3	Invalid High Preset	0,1	High preset is less than or equal to zero (0)
		2…9	High preset is less than or equal to low preset
4	Invalid Overflow	0…9	High preset is greater than overflow
5	Invalid Underflow	2…9	Low preset is less than underflow

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

Function Enabled (FE)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
FE - Function Enabled	HSC:0/FE	bit	0…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The FE (Function Enabled) is a status/control bit that defines when the HSC interrupt is enabled, and that interrupts generated by the HSC are processed based on their priority.

This bit can be controlled by the user program or is automatically set by the HSC sub-system if auto start is enabled.
See also:Priority of User Interrupts on page 265.

Auto Start (AS)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
AS - Auto Start	HSC:0/AS	bit	0…9	control	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The AS (Auto Start) is configured with the programming device and stored as part of the user program. The auto start bit defines if the HSC function automatically starts whenever the controller enters any run or test mode. The CE (Counting Enabled) bit must also be set to enable the HSC.

Error Detected (ED)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
ED - Error Detected	HSC:0/ED	bit	0…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The ED (Error Detected) flag is a status bit that can be used in the control program to detect if an error is present in the HSC sub-system. The most common type of error that this bit represents is a configuration error. When this

bit is set (1), you should look at the specific error code in parameter HSC:0.ER. This bit is maintained by the controller and is set and cleared automatically.

Counting Enabled (CE)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
CE - Counting Enabled	HSC:0/CE	bit	0…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The CE (Counting Enabled) control bit is used to enable or disable the High-Speed Counter. When set (1), counting is enabled; when clear (0, default) counting is disabled. If this bit is disabled while the counter is running, the accumulated value is held; if the bit is then set, counting resumes.

This bit can be controlled by the user program and retains its value through a power cycle. This bit must be set for the high-speed counter to operate.

Set Parameters (SP)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
SP - Set Parameters	HSC:0/SP	bit	0…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The SP (Set Parameters) control bit is used to load new variables to the HSC sub-system. When an OTE instruction with the address of HSC:0/SP is solved true (off-to-on rung transition), all configuration variables currently stored in the HSC function are checked and loaded into the HSC sub-system. The HSC sub-system then operates based on those newly loaded settings.

This bit is controlled by the user program and retains its value through a power cycle. It is up to the user program to set and clear this bit. SP can be toggled while the HSC is running and no counts are lost.

User Interrupt Enable (UIE)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UIE - User Interrupt Enable	HSC:0/UIE	bit	0…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UIE (User Interrupt Enable) bit is used to enable or disable HSC subroutine processing. This bit must be set (1) if the user wants the controller to process the HSC subroutine when any of the following conditions exist:

- Low preset reached
- High preset reached
- Overflow condition - count up through the overflow value
- Underflow condition - count down through the underflow value

If this bit is cleared (0), the HSC sub-system does not automatically scan the HSC subroutine. This bit can be controlled from the user program (using the OTE, UIE, or UID instructions).

ATTENTION: If you enable interrupts during the program scan via an OTL, OTE, or UIE, this instruction *must* be the *last* instruction executed on the rung (last instruction on last branch). It is recommended this be the only output instruction on the rung.

User Interrupt Executing (UIX)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UIX - User Interrupt Executing	HSC:0/UIX	bit	0…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UIX (User Interrupt Executing) bit is set (1) whenever the HSC sub-system begins processing the HSC subroutine due to any of the following conditions:

- Low preset reached
- High preset reached
- Overflow condition - count up through the overflow value
- Underflow condition - count down through the underflow value

The HSC UIX bit can be used in the control program as conditional logic to detect if an HSC interrupt is executing.

The HSC sub-system will clear (0) the UIX bit when the controller completes its processing of the HSC subroutine.

User Interrupt Pending (UIP)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UIP - User Interrupt Pending	HSC:0/UIP	bit	0…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UIP (User Interrupt Pending) is a status flag that represents an interrupt is pending. This status bit can be monitored or used for logic purposes in the control program if you need to determine when a subroutine cannot be executed immediately.

This bit is maintained by the controller and is set and cleared automatically.

User Interrupt Lost (UIL)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UIL - User Interrupt Lost	HSC:0/UIL	bit	0…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UIL (User Interrupt Lost) is a status flag that represents an interrupt has been lost. The controller can process 1 active and maintain up to 2 pending user interrupt conditions.

This bit is set by the controller. It is up to the control program to utilize, track if necessary, and clear the lost condition.

Low Preset Mask (LPM)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
LPM - Low Preset Mask	HSC:0/LPM	bit	2…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The LPM (Low Preset Mask) control bit is used to enable (allow) or disable (not allow) a low preset interrupt from occurring. If this bit is clear (0), and a Low Preset Reached condition is detected by the HSC, the HSC user interrupt is not executed.

This bit is controlled by the user program and retains its value through a power cycle. It is up to the user program to set and clear this bit.

Low Preset Interrupt (LPI)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
LPI - Low Preset Interrupt	HSC:0/LPI	bit	2…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The LPI (Low Preset Interrupt) status bit is set (1) when the HSC accumulator reaches the low preset value and the HSC interrupt has been triggered. This bit can be used in the control program to identify that the low preset condition caused the HSC interrupt. If the control program needs to perform any specific control action based on the low preset, this bit would be used as conditional logic.

This bit can be cleared (0) by the control program and is also be cleared by the HSC sub-system whenever these conditions are detected:

- High Preset Interrupt executes
- Underflow Interrupt executes
- Overflow Interrupt executes
- Controller enters an executing mode

Low Preset Reached (LPR)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
LPR - Low Preset Reached	HSC:0/LPR	bit	2…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The LPR (Low Preset Reached) status flag is set (1) by the HSC sub-system whenever the accumulated value (HSC:0.ACC) is less than or equal to the low preset variable (HSC:0.LOP).

This bit is updated continuously by the HSC sub-system whenever the controller is in an executing mode.

High Preset Mask (HPM)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
HPM - High Preset Mask	HSC:0/HPM	bit	0…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The HPM (High Preset Mask) control bit is used to enable (allow) or disable (not allow) a high preset interrupt from occurring. If this bit is clear (0), and a High Preset Reached condition is detected by the HSC, the HSC user interrupt is not executed.

This bit is controlled by the user program and retains its value through a power cycle. It is up to the user program to set and clear this bit.

High Preset Interrupt (HPI)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
HPI - High Preset Interrupt	HSC:0/HPI	bit	0…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The HPI (High Preset Interrupt) status bit is set (1) when the HSC accumulator reaches the high preset value and the HSC interrupt is triggered. This bit can be used in the control program to identify that the high preset condition caused the HSC interrupt. If the control program needs to perform any specific control action based on the high preset, this bit is used as conditional logic.

This bit can be cleared (0) by the control program and is also cleared by the HSC sub-system whenever these conditions are detected:

- Low Preset Interrupt executes
- Underflow Interrupt executes
- Overflow Interrupt executes
- Controller enters an executing mode

High Preset Reached (HPR)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
HPR - High Preset Reached	HSC:0/HPR	bit	2…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The HPR (High Preset Reached) status flag is set (1) by the HSC sub-system whenever the accumulated value (HSC:0.ACC) is greater than or equal to the high preset variable (HSC:0.HIP).

This bit is updated continuously by the HSC sub-system whenever the controller is in an executing mode.

Underflow (UF)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UF - Underflow	HSC:0/UF	bit	0…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UF (Underflow) status flag is set (1) by the HSC sub-system whenever the accumulated value (HSC:0.ACC) has counted through the underflow variable (HSC:0.UNF).

This bit is transitional and is set by the HSC sub-system. It is up to the control program to utilize, track if necessary, and clear (0) the underflow condition.

Underflow conditions do not generate a controller fault.

Underflow Mask (UFM)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UFM - Underflow Mask	HSC:0/UFM	bit	2…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UFM (Underflow Mask) control bit is used to enable (allow) or disable (not allow) a underflow interrupt from occurring. If this bit is clear (0), and a Underflow Reached condition is detected by the HSC, the HSC user interrupt is not executed.

This bit is controlled by the user program and retains its value through a power cycle. It is up to the user program to set and clear this bit.

Underflow Interrupt (UFI)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
UFI - Underflow Interrupt	HSC:0/UFI	bit	2…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The UFI (Underflow Interrupt) status bit is set (1) when the HSC accumulator counts through the underflow value and the HSC interrupt is triggered. This bit can be used in the control program to identify that the underflow condition caused the HSC interrupt. If the control program needs to perform any specific control action based on the underflow, this bit is used as conditional logic.

This bit can be cleared (0) by the control program and is also cleared by the HSC sub-system whenever these conditions are detected:

- Low Preset Interrupt executes
- High Preset Interrupt executes
- Overflow Interrupt executes
- Controller enters an executing mode

Overflow (OF)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
OF - Overflow	HSC:0/OF	bit	0…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The OF (Overflow) status flag is set (1) by the HSC sub-system whenever the accumulated value (HSC:0.ACC) has counted through the overflow variable (HSC:0.OF).

This bit is transitional and is set by the HSC sub-system. It is up to the control program to utilize, track if necessary, and clear (0) the overflow condition.

Overflow conditions do not generate a controller fault.

Overflow Mask (OFM)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
OFM - Overflow Mask	HSC:0/OFM	bit	0…9	control	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The OFM (Overflow Mask) control bit is used to enable (allow) or disable (not allow) an overflow interrupt from occurring. If this bit is clear (0), and an overflow reached condition is detected by the HSC, the HSC user interrupt is not executed.

This bit is controlled by the user program and retains its value through a power cycle. It is up to the user program to set and clear this bit.

Overflow Interrupt (OFI)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
OFI - Overflow Interrupt	HSC:0/OFI	bit	0…9	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The OFI (Overflow Interrupt) status bit is set (1) when the HSC accumulator counts through the overflow value and the HSC interrupt is triggered. This bit can be used in the control program to identify that the overflow variable caused the HSC interrupt. If the control program needs to perform any specific control action based on the overflow, this bit is used as conditional logic.

This bit can be cleared (0) by the control program and is also cleared by the HSC sub-system whenever these conditions are detected:

- Low Preset Interrupt executes
- High Preset Interrupt executes

- Underflow Interrupt executes
- Controller enters an executing mode

Count Direction (DIR)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
DIR - Count Direction	HSC:0/DIR	bit	0…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The DIR (Count Direction) status flag is controlled by the HSC sub-system. When the HSC accumulator counts up, the direction flag is set (1). Whenever the HSC accumulator counts down, the direction flag is cleared (0).

If the accumulated value stops, the direction bit retains its value. The only time the direction flag changes is when the accumulated count reverses.

This bit is updated continuously by the HSC sub-system whenever the controller is in a run mode.

Mode Done (MD)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
MD - Mode Done	HSC:0/MD	bit	0 or 1	status	read/write

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The MD (Mode Done) status flag is set (1) by the HSC sub-system when the HSC is configured for Mode 0 or Mode 1 behavior, and the accumulator counts up to the High Preset.

Count Down (CD)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
CD - Count Down	HSC:0/CD	bit	2…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The CD (Count Down) bit is used with the bidirectional counters (modes 2...9). If the CE bit is set, the CD bit is set (1). If the CE bit is clear, the CD bit is cleared (0).

Count Up (CU)

Description	Address	Data Format	HSC Modes[1]	Type	User Program Access
CU - Count Up	HSC:0/CU	bit	0…9	status	read only

(1) For Mode descriptions, see HSC Mode (MOD) on page 91.

The CU (Count Up) bit is used with all of the HSCs (modes 0...9). If the CE bit is set, the CU bit is set (1). If the CE bit is clear, the CU bit is cleared (0).

HSC Mode (MOD)

Description	Address	Data Format	Type	User Program Access
MOD - HSC Mode	HSC:0.MOD	word (INT)	control	read only

The MOD (Mode) variable sets the High-Speed Counter to one of 10 types of operation. This integer value is configured through the programming device and is accessible in the control program as a read-only variable.

HSC0's sub counter is HSC3, HSC1's sub counter is HSC4 and HSC2's sub counter is HSC5. Each set of counters share the input. The following table shows the dedicated inputs for the HSCs depending on the mode.

HSC Input Assignments

	I:0.0/0	I:0.0/1	I:0.0/2	I:0.0/3	I:0.0/4	I:0.0/5	I:0.0/6	I:0.0/7	I:0.0/8	I:0.0/9	I:0.0/10	I:0.0/11
HSC:0	A/C	B/D	Reset	Hold								
HSC:1					A/C	B/D	Reset	Hold				
HSC:2									A/C	B/D	Reset	Hold
HSC:3			A/C	B/D								
HSC:4							A/C	B/D				
HSC:5											A/C	B/D

HSC Operating Modes

Mode Number	Type
0	Up Counter - The accumulator is immediately cleared (0) when it reaches the high preset. A low preset cannot be defined in this mode.
1	Up Counter with external reset and hold - The accumulator is immediately cleared (0) when it reaches the high preset. A low preset cannot be defined in this mode.
2	Counter with external direction
3	Counter with external direction, reset, and hold
4	Two input counter (up and down)
5	Two input counter (up and down) with external reset and hold
6	Quadrature counter (phased inputs A and B)
7	Quadrature counter (phased inputs A and B) with external reset and hold
8	Quadrature X4 counter (phased inputs A and B)
9	Quadrature X4 counter (phased inputs A and B) with external reset and hold

The main high-speed counters support 10 types of operation mode and the sub high-speed counters support 5 types (mode 0, 2, 4, 6, 8). If the main high-speed counter is set to mode 1, 3, 5, 7 or 9, then belong the sub high-speed counter will be disabled.

HSC Function Operating Modes & Input Assignments

Modes of Operation	Input 0 (HSC:0) Input 4 (HSC:1) Input 8 (HSC:2) Input 2 (HSC:3) Input 6 (HSC:4) Input 10 (HSC:5)	Input 1 (HSC:0) Input 5 (HSC:1) Input 9 (HSC:2) Input 3 (HSC:3) Input 7 (HSC:4) Input 11 (HSC:5)	Input 2 (HSC:0) Input 6 (HSC:1) Input 10 (HSC:2)	Input 3 (HSC:0) Input 7 (HSC:1) Input 11 (HSC:2)	Mode Value in User Program
Counter with Internal Direction (mode 1a)[1]	Count	Not Used	Not Used	Not Used	0
Counter with Internal Direction, External Reset and Hold (mode 1b)[2]	Count	Not Used	Reset	Hold	1
Counter with External Direction (mode 2a) [1]	Count	Direction	Not Used	Not Used	2
Counter with External Direction, Reset and Hold (mode 2b) [2]	Count	Direction	Reset	Hold	3
Two Input Counter (mode 3a) [1]	Count Up	Count Down	Not Used	Not Used	4
Two Input Counter with External Reset and Hold (mode 3b) [2]	Count Up	Count Down	Reset	Hold	5
Quadrature Counter (mode 4a) [1]	A Type Input	B Type Input	Not Used	Not Used	6
Quadrature Counter with External Reset and Hold (mode 4b) [2]	A Type Input	B Type Input	Z Type Reset	Hold	7
Quadrature X4 Counter (mode 5a) [1]	A Type Input	B Type Input	Not Used	Not Used	8
Quadrature X4 Counter with External Reset and Hold [2]	A Type Input	B Type Input	Z Type Reset	Hold	9

(1) HSC:3, HSC:4, and HSC:5 support mode 1a, 2a, 3a, 4a, and 5a only.

(2) Only valid for HSC:0, HSC:1, and HSC:2

HSC Mode 0 - Up Counter

HSC Mode 0 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count**				**Not Used**				**Not Used**				**Not Used**					
Example 1	⇑																on (1)	HSC Accumulator + 1 count
Example 2	⇑	on (1)	⇓	off (0)													off (0)	Hold accumulator value

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

HSC Mode 1 - Up Counter with External Reset and Hold

HSC Mode 1 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count**				**Not Used**				**Reset**				**Hold**					
Example 1	⇑									on (1)	⇓	off (0)				off (0)	on (1)	HSC Accumulator + 1 count
Example 2										on (1)	⇓	off (0)		on (1)				Hold accumulator value
Example3										on (1)	⇓	off (0)					off (0)	Hold accumulator value
Example 4		on (1)	⇓	off (0)						on (1)	⇓	off (0)						Hold accumulator value
Example 5									⇑									Clear accumulator (=0)

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.t

HSC Mode 2 - Counter with External Direction

HSC Mode 2 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count**				**Direction**				**Not Used**				**Not Used**					
Example 1	⇑							off (0)									on (1)	HSC Accumulator + 1 count
Example 2	⇑					on (1)											on (1)	HSC Accumulator - 1 count
Example3																	off (0)	Hold accumulator value

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

HSC Mode 3 - Counter with External Direction, Reset, and Hold

HSC Mode 3 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count**				**Direction**				**Reset**				**Hold**					
Example 1	⇑							off (0)		on (1)	⇓	off (0)				off (0)	on (1)	HSC Accumulator + 1 count
Example 2	⇑					on (1)				on (1)	⇓	off (0)				off (0)	on (1)	HSC Accumulator - 1 count
Example3										on (1)	⇓	off (0)		on (1)				Hold accumulator value
Example 4										on (1)	⇓	off (0)					off (0)	Hold accumulator value
Example 5		on (1)	⇓	off (0)						on (1)	⇓	off (0)						Hold accumulator value
Example 6									⇑									Clear accumulator (=0)

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

HSC Mode 4 - Two Input Counter (up and down)

HSC Mode 4 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count Up**				**Count Down**				**Not Used**				**Not Used**					
Example 1	⇑					on (1)	⇓	off (0)									on (1)	HSC Accumulator + 1 count
Example 2		on (1)	⇓	off (0)	⇑												on (1)	HSC Accumulator - 1 count
Example3																	off (0)	Hold accumulator value

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

HSC Mode 5 - Two Input Counter (up and down) with External Reset and Hold

HSC Mode 5 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count**				**Direction**				**Reset**				**Hold**					
Example 1	⇑					on (1)	⇓	off (0)		on (1)	⇓	off (0)				off (0)	on (1)	HSC Accumulator + 1 count
Example 2		on (1)	⇓	off (0)	⇑					on (1)	⇓	off (0)				off (0)	on (1)	HSC Accumulator - 1 count
Example3										on (1)	⇓	off (0)		on (1)				Hold accumulator value
Example 4										on (1)	⇓	off (0)					off (0)	Hold accumulator value
Example 5		on (1)	⇓	off (0)						on (1)	⇓	off (0)						Hold accumulator value
Example 6									⇑									Clear accumulator (=0)

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

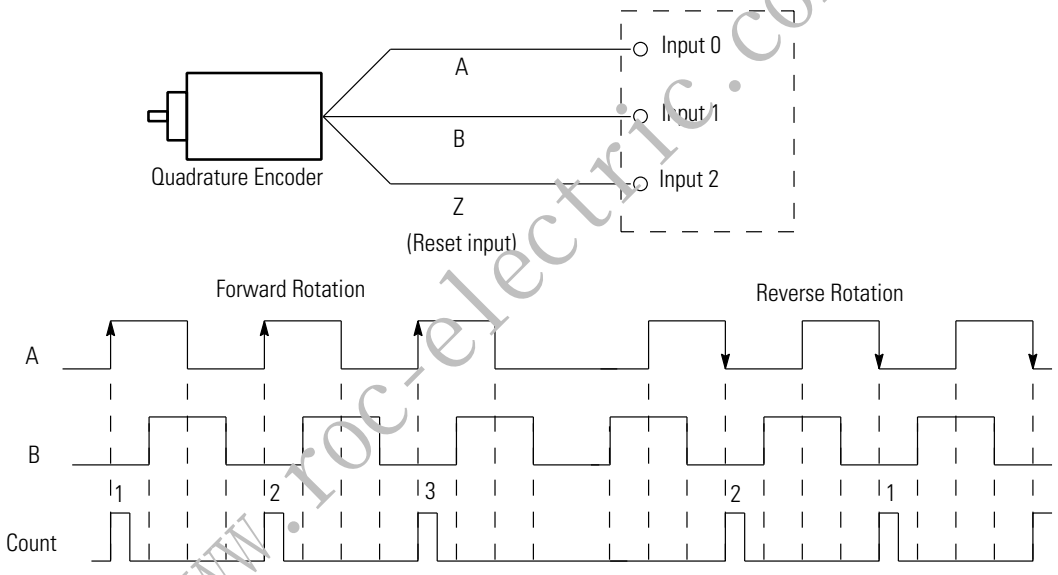
TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

Using the Quadrature Encoder

The Quadrature Encoder is used for determining direction of rotation and position for rotating, such as a lathe. The Bidirectional Counter counts the rotation of the Quadrature Encoder.

The figure below shows a quadrature encoder connected to inputs 0, 1, and 2. The count direction is determined by the phase angle between A and B. If A leads B, the counter increments. If B leads A, the counter decrements.

The counter can be reset using the Z input. The Z outputs from the encoders typically provide one pulse per revolution.



HSC Mode 6 - Quadrature Counter (phased inputs A and B)

HSC Mode 6 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count A**				**Count B**				**Not Used**				**Not Used**					
Example 1[1]	⇑							off (0)									on (1)	HSC Accumulator + 1 count
Example 2[2]			⇓					off (0)									on (1)	HSC Accumulator - 1 count
Example3				off (0)														Hold accumulator value
Example 4		on (1)																Hold accumulator value
Example 5						on (1)												Hold accumulator value
Example 6																	off (0)	Hold accumulator value

(1) Count input A leads count input B.

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

HSC Mode 7 - Quadrature Counter (phased inputs A and B) With External Reset and Hold

(2) Count input B leads count input A.
Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

HSC Mode 7 Examples

Input Terminals	I1:0.0/0 (HSC0)				I1:0.0/1 (HSC0)				I1:0.0/2 (HSC0)				I1:0.0/3 (HSC0)				**CE Bit**	**Comments**
Function	**Count A**				**Count B**				**Z reset**				**Hold**					
Example 1(1)	⇑							off (0)								off (0)	on (1)	HSC Accumulator + 1 count
Example 2(2)			⇓					off (0)				off (0)				off (0)	on (1)	HSC Accumulator - 1 count
Example3			⇓	off (0)				off (0)		on (1)								Reset accumulator to zero
Example 4		on (1)																Hold accumulator value
Example 5						on (1)												Hold accumulator value
Example 6												off (0)		on (1)				Hold accumulator value
Example 7												off (0)					off (0)	Hold accumulator value

(1) Count input A leads count input B.

(2) Count input B leads count input A.

Blank cells = don't care, ⇑ = rising edge, ⇓ = falling edge

TIP Inputs I1:0.0/0 through I1:0.0/11 are available for use as inputs to other functions regardless of the HSC being used.

HSC Mode 8 - Quadrature X4 Counter

HSC Mode 8 Examples

I1:0.0/1(HSC0) (A)	**I1:0.0/1(HSC0) (B)**	**Value of CE Bit**	**Accumulator and Counter Action**
▲	OFF	TRUE	Count Up Acc. Value
▲	ON	TRUE	Count Down Acc. Value
▼	OFF	TRUE	Count Down Acc. Value
▼	ON	TRUE	Count Up Acc. Value

HSC Mode 8 Examples

I1:0.0/1(HSC0) (A)	I1:0.0/1(HSC0) (B)	Value of CE Bit	Accumulator and Counter Action
OFF	▲	TRUE	Count Down Acc. Value
ON	▲	TRUE	Count Up Acc. Value
OFF	▼	TRUE	Count Up Acc. Value
ON	▼	TRUE	Count Down Acc. Value
OFF or ON	OFF or ON	X	Hold Acc. Value
X	X	FALSE	Hold Acc. Value

HSC Mode 9 - Quadrature X4 Counter with External Reset and Hold

HSC Mode 9 Examples

I1:0.0/0(HSC0) (A))	I1:0.0/1(HSC0) (B)	I1:0.0/2(HSC0) (Reset)	I1:0.0/3(HSC0) (Hold)	Value of CE Bit	Accumulator and Counter Action
▲	OFF	X	-	TRUE	Count Up Acc. Value
▲	ON	X	-	TRUE	Count Down Acc. Value
▼	OFF	X	-	TRUE	Count Down Acc. Value
▼	ON	X	-	TRUE	Count Up Acc. Value
OFF	▲	X	-	TRUE	Count Down Acc. Value
ON	▲	X	-	TRUE	Count Up Acc. Value
OFF	▼	X	-	TRUE	Count Up Acc. Value
ON	▼	X		TRUE	Count Down Acc. Value
OFF or ON	OFF or ON	OFF	X	X	Hold Acc. Value
OFF	OFF	ON	X	X	Reset Acc. to Zero
X	X	OFF	ON	X	Hold Acc. Value
X	X	OFF	X	FALSE	Hold Acc. Value

Accumulator (ACC)

Description	Address	Data Format	Type	User Program Access
ACC - Accumulator	HSC:0.ACC	long word (32-bit INT)	control	read/write

The ACC (Accumulator) contains the number of counts detected by the HSC sub-system. If either mode 0 or mode 1 is configured, *the value of the software accumulator is cleared* (0) when a high preset is reached or when an overflow condition is detected.

High Preset (HIP)

Description	Address	Data Format	Type	User Program Access
HIP - High Preset	HSC:0.HIP	long word (32-bit INT)	control	read/write

The HIP (High Preset) is the upper setpoint (in counts) that defines when the HSC sub-system generates an interrupt. To load data into the high preset, the control program must do one of the following:

- Toggle (low to high) the Set Parameters (HSC:0/SP) control bit. When the SP bit is toggled high, the data currently stored in the HSC function file is transferred/loaded into the HSC sub-system.
- Load new HSC parameters using the HSL instruction. See HSL - High-Speed Counter Load on page 103.

The data loaded into the high preset must be less than or equal to the data resident in the overflow (HSC:0.OVF) parameter or an HSC error is generated.

Low Preset (LOP)

Description	Address	Data Format	Type	User Program Access
LOP - Low Preset	HSC:0.LOP	long word (32-bit INT)	control	read/write

The LOP (Low Preset) is the lower setpoint (in counts) that defines when the HSC sub-system generates an interrupt. To load data into the low preset, the control program must do one of the following:

- Toggle (low to high) the Set Parameters (HSC:0/SP) control bit. When the SP bit is toggled high, the data currently stored in the HSC function file is transferred/loaded into the HSC sub-system.
- Load new HSC parameters using the HSL instruction. See HSL - High-Speed Counter Load on page 103.

The data loaded into the low preset must greater than or equal to the data resident in the underflow (HSC:0.UNF) parameter, or an HSC error is generated. (If the underflow and low preset values are negative numbers, the low preset must be a number with a smaller absolute value.)

Overflow (OVF)

Description	Address	Data Format	Type	User Program Access
OVF - Overflow	HSC:0.OVF	long word (32-bit INT)	control	read/write

The OVF (Overflow) defines the upper count limit for the counter. If the counter's accumulated value increments past the value specified in this variable, an overflow interrupt is generated. When the overflow interrupt is generated, the HSC sub-system rolls the accumulator over to the underflow value and the counter continues counting from the underflow value (counts are not lost in this transition). The user can specify any value for the overflow position, provided it is greater than the underflow value and falls between -2,147,483,648 and 2,147,483,647.

To load data into the overflow variable, the control program must toggle (low to high) the Set Parameters (HSC:0.0/SP) control bit. When the SP bit is toggled high, the data currently stored in the HSC function file is transferred/loaded into the HSC sub-system.

TIP Data loaded into the overflow variable must be greater than or equal to the data resident in the high preset (HSC:0.HIP) or an HSC error is generated.

Underflow (UNF)

Description	Address	Data Format	Type	User Program Access
UNF - Underflow	HSC:0.UNF	long word (32-bit INT)	control	read/write

The UNF (Underflow) defines the lower count limit for the counter. If the counter's accumulated value decrements past the value specified in this variable, an underflow interrupt is generated. When the underflow interrupt is generated, the HSC sub-system resets the accumulated value to the overflow value and the counter then begins counting from the overflow value (counts are not lost in this transition). The user can specify any value for the underflow position, provided it is less than the overflow value and falls between -2,147,483,648 and 2,147,483,647.

To load data into the underflow variable, the control program must toggle (low to high) the Set Parameters (HSC:0.0/SP) control bit. When the SP bit is toggled high, the data currently stored in the HSC function file is transferred/loaded into the HSC sub-system.

TIP Data loaded into the underflow variable must be less than or equal to the data resident in the low preset (HSC:0.LOP) or an HSC error is generated.

Output Mask Bits (OMB)

Description	Address	Data Format	Type	User Program Access
OMB - Output Mask Bits	HSC:0.OMB	word (16-bit binary)	control	read only

The OMB (Output Mask Bits) define which outputs on the controller can be directly controlled by the high-speed counter. The HSC sub-system has the ability to directly (without control program interaction) turn outputs ON or OFF based on the HSC accumulator reaching the High or Low presets. The bit pattern stored in the OMB variable defines which outputs are controlled by the HSC and which outputs are not controlled by the HSC.

The bit pattern of the OMB variable directly corresponds to the output bits on the controller. Bits that are set (1) are enabled and can be turned on or off by the HSC sub-system. Bits that are clear (0) cannot be turned on or off by the HSC sub-system. The mask bit pattern can be configured only during initial setup.

This table illustrates this relationship:

Affect of HSC Output Mask on Base Unit Outputs

Output Address	16-Bit Signed Integer Data Word															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSC:0.HPO (high preset output)					0	0	1	1	0	0	0	1	1	0	0	1
HSC:0.OMB (output mask)					0	1	1	0	0	0	1	1	0	0	1	1
O0:0.0					**0**	**1**					**0**	**1**			**0**	**1**

The outputs shown in the black boxes are the outputs under the control of the HSC sub-system. The mask defines which outputs can be controlled. The high preset output or low preset output values (HPO or LPO) define if each output is either ON (1) or OFF (0). Another way to view this is that the high or low preset output is written through the output mask, with the output mask acting like a filter.

The bits in the gray boxes are unused. The first 6 bits of the mask word are used and the remaining mask bits are not functional because they do not correlate to any physical outputs on the base unit.

The mask bit pattern can be configured only during initial setup.

High Preset Output (HPO)

Description	Address	Data Format	Type	User Program Access
HPO - High Preset Output	HSC:0.HPO	word (16-bit binary)	control	read/write

The HPO (High Preset Output) defines the state (1 = ON or 0 = OFF) of the outputs on the controller when the high preset is reached. See Output Mask Bits (OMB) on page 101 for more information on how to directly turn outputs on or off based on the high preset being reached.

The high output bit pattern can be configured during initial setup, or while the controller is operating. Use the HSL instruction or the SP bit to load the new parameters while the controller is operating.

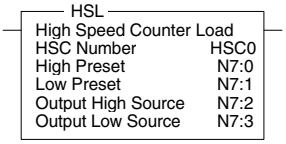
Low Preset Output (LPO)

Description	Address	Data Format	Type	User Program Access
LPO - Low Preset Output	HSC:0.LPO	word (16-bit binary)	control	read/write

The LPO (Low Preset Output) defines the state (1 = "on", 0 = "off") of the outputs on the controller when the low preset is reached. See Output Mask Bits (OMB) on page 101 for more information on how to directly turn outputs on or off based on the low preset being reached.

The low output bit pattern can be configured during initial setup, or while the controller is operating. Use the HSL instruction or the SP bit to load the new parameters while the controller is operating.

HSL - High-Speed Counter Load



Instruction Type: output

Execution Time for HSL Instruction

Controller	Data Size	Execution Time When Rung Is:	
		True	False
MicroLogix 1400	word	18.8260 µs	0.2910 µs
	long word	18.6510 µs	0.4690 µs

The HSL (High-Speed Load) instruction allows the high and low presets, and high and low output source to be applied to a high-speed counter. These parameters are described below:

- Counter Number - Specifies which high-speed counter is being used; 0 = HSC0, 1 = HSC1, 2 = HSC2, 3 = HSC3, 4 = HSC4 and 5 = HSC5
- High Preset - Specifies the value in the high preset register. The data ranges for the high preset are -32768...32767 (word) and -2,147,483,648...2,147,483,647 (long word).
- Low Preset - Specifies the value in the low preset register. The data ranges for the low preset are -32768...32767 (word) and -2,147,483,648...2,147,483,647 (long word).
- Output High Source - Specifies the value in the HPO - high preset output register. The data range for the output high source is from 0...65,535.
- Output Low Source - Specifies the value in the LPO - low preset output register. The data range for the output low source is from 0...65,535.

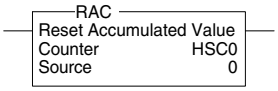
Valid Addressing Modes and File Types are shown below:

HSL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files													Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Counter Number												•													•						
High Preset	•	•		•	•	•				•		•													•	•	•		•	•	
Low Preset	•	•		•	•	•				•		•													•	•	•		•	•	
Output High Source	•	•		•	•	•				•		•													•	•	•		•	•	
Output Low Source	•	•		•	•	•				•		•													•	•	•		•	•	

RAC - Reset Accumulated Value



Instruction Type: output

Controller	Execution Time When Rung Is:	
	True	False
MicroLogix 1400	8.3310 µs	0.2030 µs

The RAC instruction resets the high-speed counter and allows a specific value to be written to the HSC accumulator. The RAC instruction uses the following parameters:

- Counter Number - Specifies which high-speed counter is being used:
  - Counter Number 0 = HSC0, 1 = HSC1, 2 = HSC2, 3 = HSC3, 4 = HSC4, 5 = HSC5
- Source - Specifies the location of the data to be loaded into the HSC accumulator. The data range is from -2,147,483,648...2,147,483,647.

Valid Addressing Modes and File Types are shown below:

RAC Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70

Parameter	Data Files													Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CSF - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Counter Number																									•						
Source																									•	•	•			•	

Programmable Limit Switch (PLS) File

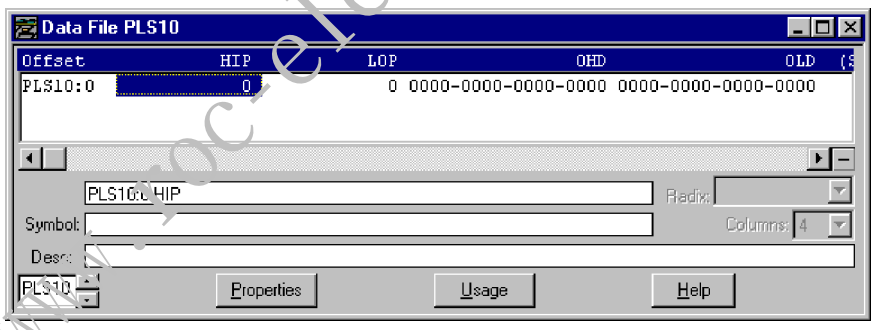
The Programmable Limit Switch function allows you to configure the High-Speed Counter to operate as a PLS (programmable limit switch) or rotary cam switch.

When PLS operation is enabled, the HSC (High-Speed Counter) uses a PLS data file for limit/cam positions. Each limit/cam position has corresponding data parameters that are used to set or clear physical outputs on the controller's base unit. The PLS data file is illustrated below.

IMPORTANT	The PLS Function only operates in tandem with the HSC of a MicroLogix 1400. To use the PLS function, an HSC must first be configured.

PLS Data File

Data files 9...255 can be used for PLS operations. Each PLS data file can be up to 256 elements long. Each element within a PLS file consumes 6 user words of memory. The PLS data file is shown below:



PLS Operation

When the PLS function is enabled, and the controller is in the run mode, the HSC will count incoming pulses. When the count reaches the first preset (High - HIP or Low - LOP) defined in the PLS file, the output source data (High - OHD or Low - OLD) will be written through the HSC mask.

At that point, the next preset (High - HIP or Low - LOP) defined in the PLS file becomes active.

When the HSC counts to that new preset, the new output data is written through the HSC mask. This process continues until the last element within the PLS file is loaded. At that point the active element within the PLS file is reset to zero. This behavior is referred to as circular operation.

TIP The Output High Data (OHD) is only written when the High preset (HIP) is reached. The Output Low Data (OLD) is written when the low preset is reached.

TIP Output High Data is only operational when the counter is counting up. Output Low Data is only operational when the counter is counting down.

If invalid data is loaded during operation, an HSC error is generated (within the HSC function file). The error will not cause a controller fault. If an invalid parameter is detected, it will be skipped and the next parameter will be loaded for execution (provided it is valid).

You can use the PLS in Up (high), Down (low), or both directions. If your application only counts in one direction, simply ignore the other parameters.

The PLS function can operate with all of the other HSC capabilities. The ability to select which HSC events generate a user interrupt are not limited.

Addressing PLS Files

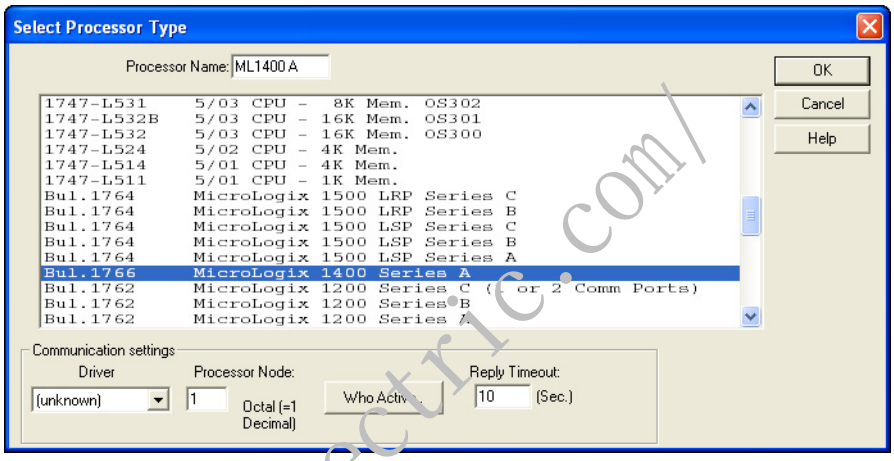
The addressing format for the PLS file is shown below.

Format	Explanation		
PLSf:e.s	**PLS**	Programmable Limit Switch file	
	f	File number	The valid file number range is from 9…255.
	:	Element delimiter	
	e	Element number	The valid element number range is from 0…255.
	.	Sub-Element delimiter	
	s	Sub-Element number	The valid sub-element number range is from 0…5
Examples:	**PLS10:2** **PLS12:36.5**		**PLS File 10, Element 2** **PLS File 12, Element 36, Sub-Element 5 (Output Low Source)**

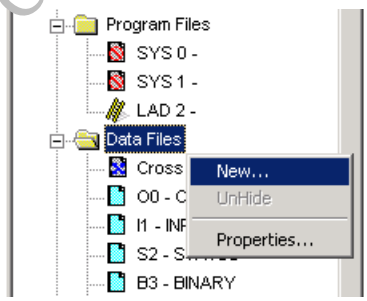
PLS Example

Setting up the PLS File

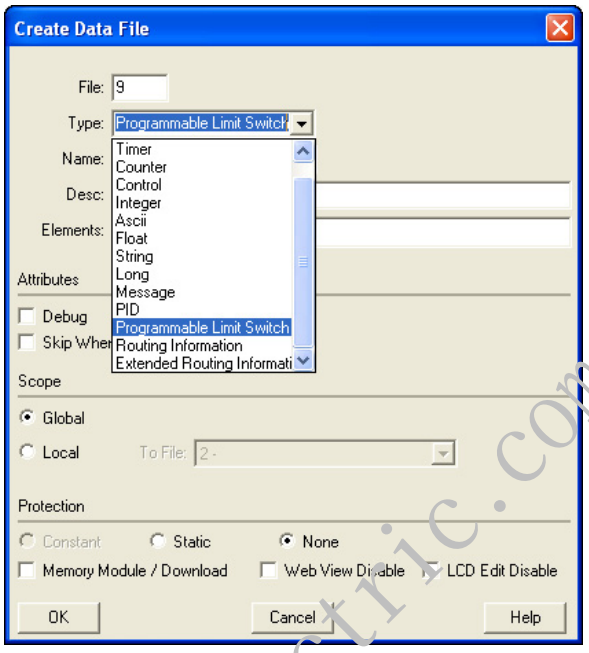
1. Using RSLogix 500/RSLogix Micro, create a new project, give it a name and select the appropriate controller.



2. Right click on *Data Files* and select *New*.



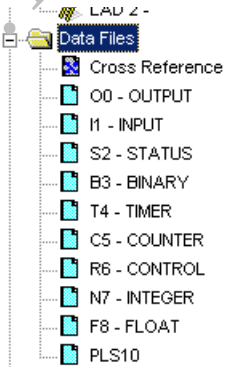
3. Enter a file number (9...255) and select *Programmable Limit Switch* as the type. A Name and/or Description may be entered as well, but is not required.



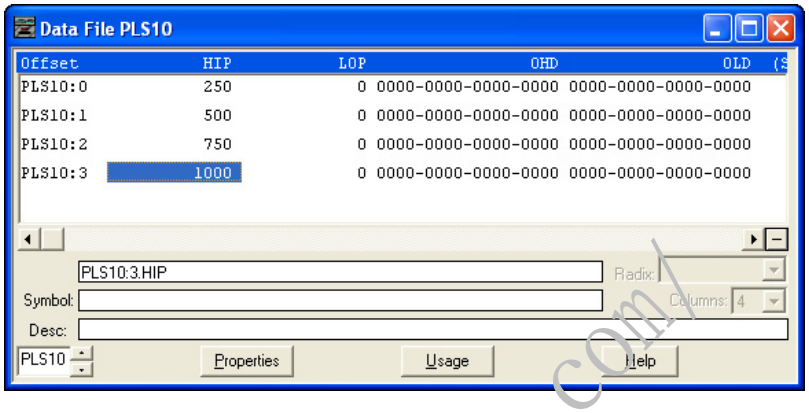
4. *Elements* refers to the number of PLS steps. For this example enter a value of 4.

   If more steps are required at a later time, simply go to the properties for the PLS data file and increase the number of elements.

5. Under Data Files, *PLS10* should appear as shown here:



6. Double-click on *PLS10* under Data Files. For this example, enter the values as illustrated below.



PLS Data File Definitions:

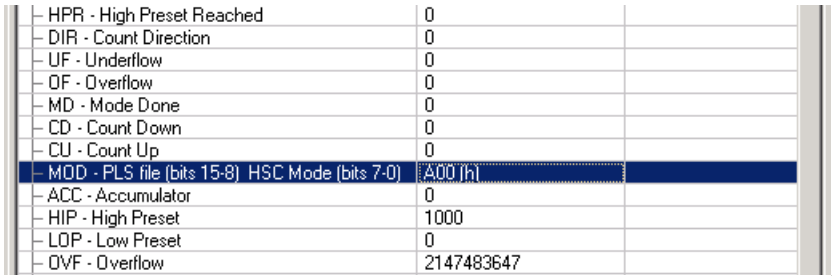
Data	Description	Data Format
HIP	High Preset	32-bit signed integer
LOP	Low Preset	
OHD	Output High Data	16-bit binary (bit 15--> 0000 0000 0000 0000 <--bit 0)
OLD	Output Low Data	

Once the values above have been entered for HIP and OHD, the PLS is configured.

Configuring the HSC for Use with the PLS

1. Under Controller, double-click on *Function Files*.
2. For *HSC:0*, configure the HSC.MOD to use PLS10 and for the HSC to operate in mode 00.

IMPORTANT The value for MOD must be entered in Hexadecimal.
For example, PLS10 = 0A and HSC Mode = 00



PLS Operation for This Example

When the ladder logic first runs, HSC.ACC equals 0, therefore PLS10:0.OLD's data is sent through the HSC.OMB mask and sets all the outputs off.

When HSC.ACC equals 250, the PLS10:0.OHD is sent through the HSC.OMB mask and energizes the outputs.

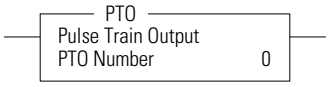
This will repeat as the HSC.ACC reaches 500, 750, and 1000. Once completed, the cycle resets and repeats.

Using High-Speed Outputs

The high-speed output instructions allow you to control and monitor the PTO and PWM functions which control the physical high-speed outputs.

Instruction	Used To:	Page
PTO - Pulse Train Output	Generate stepper pulses	111
PWM - Pulse Width Modulation	Generate PWM output	128

PTO - Pulse Train Output



IMPORTANT The PTO function can only be used with the controller's embedded I/O. It cannot be used with expansion I/O modules.

IMPORTANT The PTO instruction should only be used with MicroLogix 1400 BXB or BXBA units. Relay outputs are not capable of performing very high-speed operations.

Instruction Type: output

Controller	When Rung Is:	
	True	False
MicroLogix 1400	11.0210 µs	5.5115 µs

Pulse Train Output Function

The MicroLogix 1400 1766-L32BXB and 1766-L32BXBA controller supports three high-speed outputs. These outputs can be used as standard outputs (not high-speed) or individually configured for PTO or PWM operation. The PTO functionality allows a simple motion profile or pulse profile to be generated directly from the controller. The pulse profile has three primary components:

- Total number of pulses to be generated
- Accelerate/decelerate intervals
- Run interval

The PTO instruction, along with the HSC and PWM functions, are different than most other controller instructions. Their operation is performed by custom circuitry that runs in parallel with the main system processor. This is necessary because of the high performance requirements of these functions.

In this implementation, the user defines the total number of pulses to be generated (which corresponds to distance traveled), and how many pulses to use for each acceleration/deceleration period. The number of pulses not used in the acceleration/deceleration period defines how many pulses are generated during the run phase. In this implementation, the accelerate/decelerate intervals are not required to be the same. Independent values can be defined for these intervals. The ADI bit in the PTOX function file is used to enable this feature. See page 116.

Within the PTOX function file, there are PTO element(s). An element can be set to control either output 2 (O0:0/2 on 1766-L32BXB or 1766-L32BXBA), output 3 (O0:0/3 on 1766-L32BXB or 1766-L32BXBA) or output 4 (O0:0/4 on 1766-L32BXB or 1766-L32BXBA).

The interface to the PTOX sub-system is accomplished by scanning a PTOX instruction in the main program file (file number 2) or by scanning a PTOX instruction in any of the subroutine files. A typical operating sequence of a PTOX instruction is as follows:

1. The rung that a PTO instruction is on is solved true.
2. The PTO instruction is started, and pulses are produced based on the accelerate/decelerate (ACCEL) parameters, which define the number of ACCEL pulses and the type of profile: s-curve or trapezoid.
3. The ACCEL phase completes.
4. The RUN phase is entered and the number of pulses defined for RUN are output.
5. The RUN phase completes.
6. Decelerate (DECEL) is entered, and pulses are produced based on the accelerate/decelerate parameters, which define the number of DECEL pulses and the type of profile: s-curve or trapezoid.
7. The DECEL phase completes.
8. The PTO instruction is DONE.

While the PTO instruction is being executed, status bits and information are updated as the main controller continues to operate. Because the PTO instruction is actually being executed by a parallel system, status bits and other information are updated each time the PTO instruction is scanned while it is running. This provides the control program access to PTO status while it is running.

TIP PTO status is only as fresh as the scan time of the controller. Worst case latency is the same as the maximum scan of the controller. This condition can be minimized by placing a PTO instruction in the STI (selectable timed interrupt) file, or by adding PTO instructions to your program to increase how often a PTO instruction is scanned.

The charts in the following examples illustrate the typical timing sequence/behavior of a PTO instruction. The stages listed in each chart have nothing to do with controller scan time. They simply illustrate a sequence of events. In actuality, the controller may have hundreds or thousands of scans within each of the stages illustrated in the examples.

Conditions Required to Start the PTO

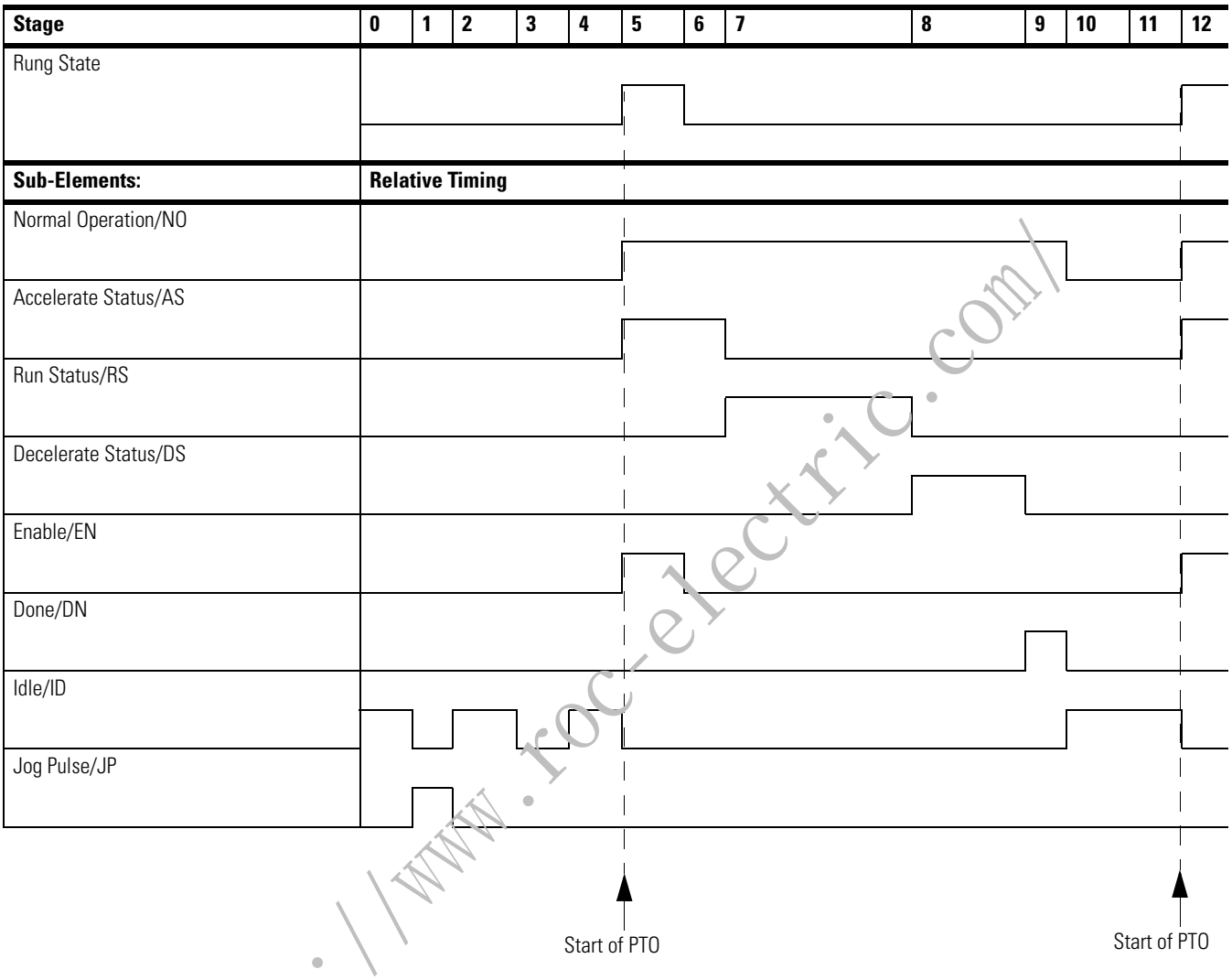
The following conditions must exist to start the PTO:

- The PTO instruction must be in an idle state.
- For idle state behavior, all of the following conditions must be met:
  - Jog Pulse (JP) bit must be off
  - Jog Continuous (JC) bit must be off
  - Enable Hard Stop (EH) bit must be off
  - Normal Operation (NS) bit must be off
  - The output cannot be forced
- The rung it is on must transition from a False state (0) to a True state (1).

Momentary Logic Enable Example

In this example, the rung state is a momentary or transitional type of input. This means that the false-to-true rung transition enables the PTO instruction and then returns to a false state prior to the PTO instruction completing its operation.

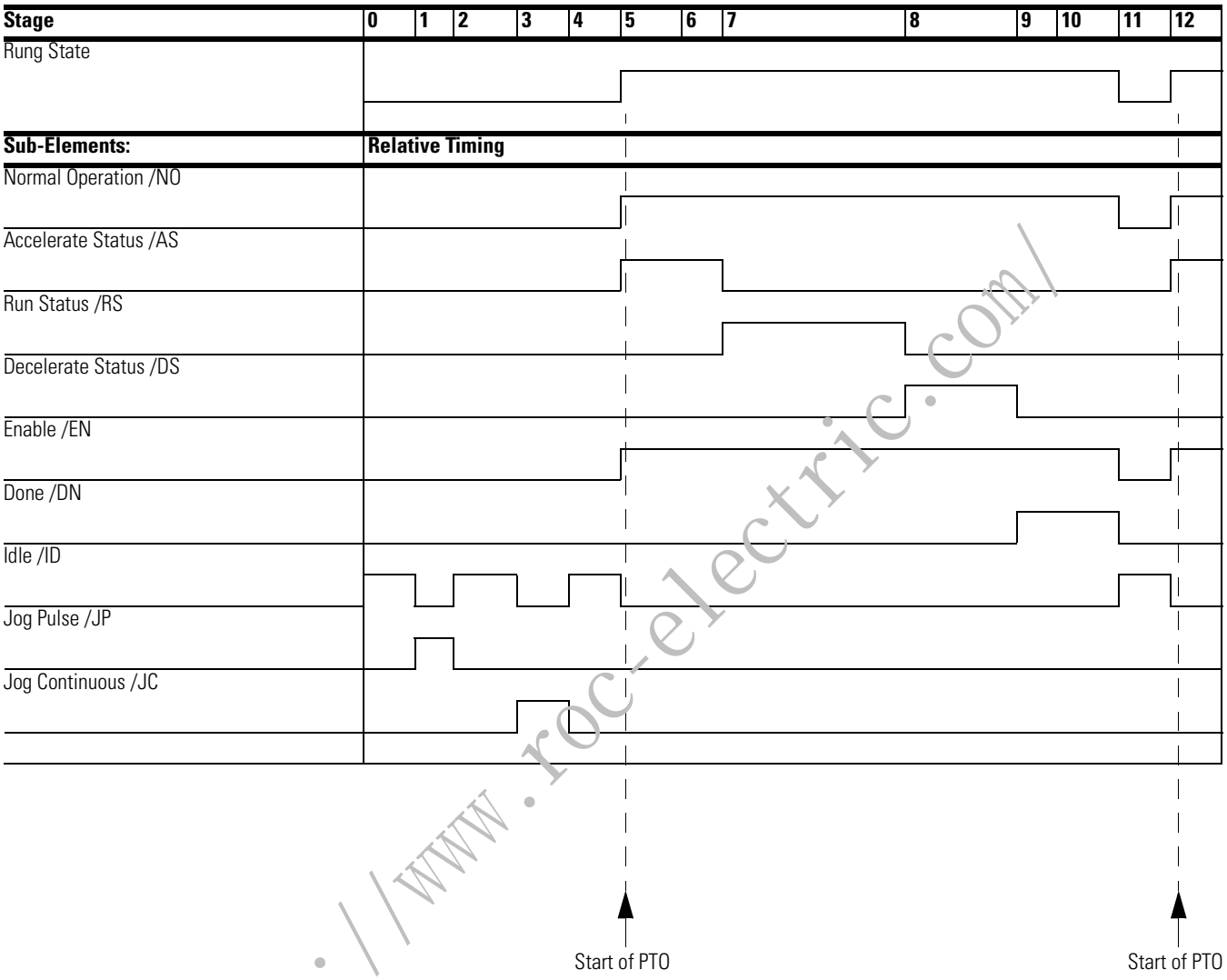
If a transitional input to the PTO instruction is used, the Done (DN) bit turns on when the instruction completes, but only remains on until the next time the PTO instruction is scanned in the user program. The structure of the control program determines when the DN bit goes off. So, to detect when the PTO instruction completes its output, you can monitor the Done (DN), Idle (ID), or Normal Operation (NO) status bits.



Standard Logic Enable Example

In this example, the rung state is a maintained type of input. This means that it enables the PTO instruction Normal Operation (NO) and maintains its logic state until after the PTO instruction completes its operation. With this type of logic, status bit behavior is as follows:

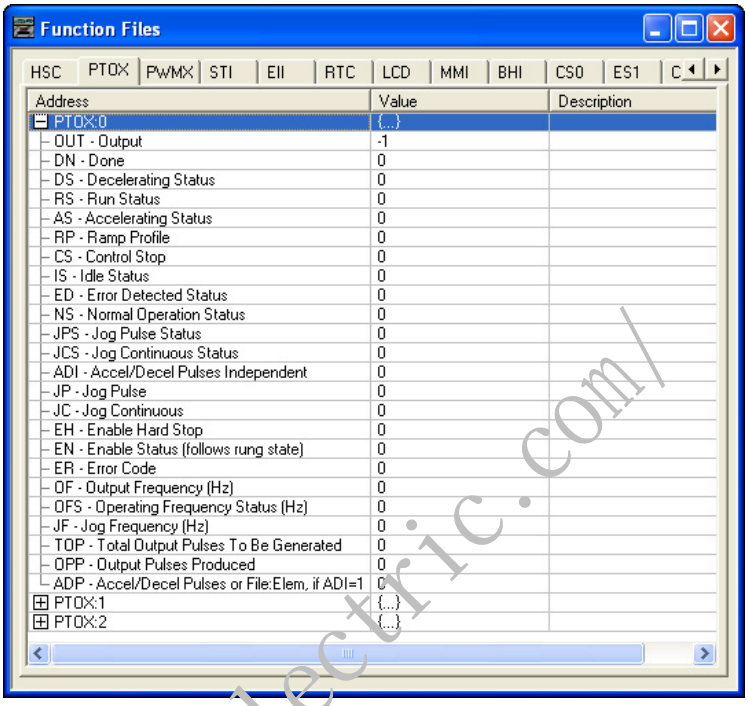
The Done (DN) bit becomes true (1) when the PTO completes and remains set until the PTO rung logic is false. The false rung logic re-activates the PTO instruction. To detect when the PTO instruction completes its output, monitor the done (DN) bit.



Pulse Train Outputs (PTOX) Function File

Within the RSLogix 500/RSLogix Micro Function File Folder, you see a PTOX Function File with three elements. These elements provide access to PTO configuration data and also allow the control program access to all information pertaining to each of the Pulse Train Outputs.

TIP If the controller mode is run, the data within sub-element fields may be changing.



Pulse Train Output Function File Sub-Elements Summary

The variables within each PTOX sub-element, along with what type of behavior and access the control program has to those variables, are listed individually below. All examples illustrate PTOX:0. Terms and behavior for PTOX:1 and PTOX:2 are identical.

PTO File Sub-elements

Sub-Element Description	Address	Data Format	Range	Type	User Program Access	For More Information
OUT - Output	PTOX:0.OUT	word (INT)	2…4	control	read only	117
DN - Done	PTOX:0/DN	bit	0 or 1	status	read only	118
DS - Decelerating Status	PTOX:0/DS	bit	0 or 1	status	read only	118
RS - Run Status	PTOX:0/RS	bit	0 or 1	status	read only	118
AS - Accelerating Status	PTOX:0/AS	bit	0 or 1	status	read only	119
RP - Ramp Profile	PTOX:0/RP	bit	0 or 1	control	read/write	119
CS - Controlled Stop	PTOX:0/CS	bit	0 or 1	control	read/write	124
IS - Idle Status	PTOX:0/IS	bit	0 or 1	status	read only	119
ED - Error Detected Status	PTOX:0/ED	bit	0 or 1	status	read only	120
NS - Normal Operation Status	PTOX:0/NS	bit	0 or 1	status	read only	120
JPS - Jog Pulse Status	PTOX:0/JPS	bit	0 or 1	status	read only	125
JCS - Jog Continuous Status	PTOX:0/JCS	bit	0 or 1	status	read only	126
ADI - Accel/Decel Pulses Independent	PTOX:0/ADI	bit	0 or 1	control	read/write	122
JP - Jog Pulse	PTOX:0/JP	bit	0 or 1	control	read/write	125

PTO File Sub-elements

Sub-Element Description	Address	Data Format	Range	Type	User Program Access	For More Information
JC - Jog Continuous	PTOX:0/JC	bit	0 or 1	control	read/write	126
EH - Enable Hard Stop	PTOX:0/EH	bit	0 or 1	control	read/write	120
EN - Enable Status (follows rung state)	PTOX:0/EN	bit	0 or 1	status	read only	121
ER - Error Code	PTOX:0.ER	word (INT)	-2…7	status	read only	127
OF - Output Frequency (Hz)	PTOX:0.OF	long word (32-bit INT)	0…100,000	control	read/write	121
OFS - Operating Frequency Status (Hz)	PTOX:0.OFS	long word (32-bit INT)	0…100,000	status	read only	121
JF - Jog Frequency (Hz)	PTOX:0.JF	long word (32-bit INT)	0…100,000	control	read/write	125
TOP - Total Output Pulses To Be Generated	PTOX:0.TOP	long word (32-bit INT)	0…2,147,483,647	control	read/write	122
OPP - Output Pulses Produced	PTOX:0.OPP	long word (32-bit INT)	0…2,147,483,647	status	read only	122
ADP - Accel/Decel Pulses	PTOX:0.ADP	long word (32-bit INT)	see p. 123	control	read/write	123

PTOX Output (OUT)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
OUT - Output	PTOX:0.OUT	word (INT)	2…4	control	read only

The PTOX OUT (Output) variable defines the output (O0:0/2, O0:0/3 or or O0:0/4) that the PTOX instruction controls. This variable is set within the function file folder when the control program is written and cannot be set by the user program.

- When OUT = 2, PTOX pulses output 2 (O0:0.0/2) of the embedded outputs.
- When OUT = 3, PTOX pulses output 3 (O0:0.0/3) of the embedded outputs
- When OUT = 4, PTOX pulses output 4 (O0:0.0/4) of the embedded outputs.

TIP Forcing an output controlled by the PTOX while it is running stops all output pulses and causes a PTOX error.

PTOX Done (DN)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
DN - Done	PTOX:0/DN	bit	0 or 1	status	read only

The PTOX DN (Done) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program. The DN bit operates as follows:

- Set (1) - Whenever a PTOX instruction has completed its operation successfully.
- Cleared (0) - When the rung the PTOX is on is false. If the rung is false when the PTOX instruction completes, the Done bit is set until the next scan of the PTOX instruction.

PTOX Decelerating Status (DS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
DS - Decelerating Status	PTOX:0/DS	bit	0 or 1	status	read only

The PTOX DS (Decel) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program. The DS bit operates as follows:

- Set (1) – Whenever a PTO instruction is within the deceleration phase of the output profile.
- Cleared (0) – Whenever a PTOX instruction is not within the deceleration phase of the output profile.

PTOX Run Status (RS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
RS - Run Status	PTOX:0/RS	bit	0 or 1	status	read only

The PTOX RS (Run Status) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program. The RS bit operates as follows:

- Set (1) – Whenever a PTOX instruction is within the run phase of the output profile.
- Cleared (0) – Whenever a PTOX instruction is not within the run phase of the output profile.

PTOX Accelerating Status (AS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
AS - Accelerating Status	PTOX:0/AS	bit	0 or 1	status	read only

The PTOX AS (Accelerating Status) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program. The AS bit operates as follows:

- Set (1) – Whenever a PTOX instruction is within the acceleration phase of the output profile.
- Cleared (0) – Whenever a PTOX instruction is not within the acceleration phase of the output profile.

PTOX Ramp Profile (RP)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
RP - Ramp Profile	PTOX:0/RP	bit	0 or 1	control	read/write

The PTOX RP (Ramp Profile) bit controls how the output pulses generated by the PTOX sub-system accelerate to and decelerate from the Output Frequency that is set in the PTOX function file (PTOX:0.OF). It can be used by an input or output instruction on any rung within the control program. The RP bit operates as follows:

- Set (1) – Configures the PTOX instruction to produce an S-Curve profile.
- Cleared (0) – Configures the PTOX instruction to produce a Trapezoid profile.

PTOX Idle Status (IS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
IS - Idle Status	PTOX:0/IS	bit	0 or 1	status	read only

The PTOX IS (Idle Status) is controlled by the PTOX sub-system. It can be used in the control program by an input instruction. The PTOX sub-system must be in an idle state whenever any PTOX operation needs to start.

The IS bit operates as follows:

- Set (1) – PTOX sub-system is in an idle state. The idle state is defined as the PTOX is not running and no errors are present.
- Cleared (0) – PTOX sub-system is not in an idle state (it is running)

PTOX Error Detected (ED)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
ED - Error Detected Status	PTOX:0/ED	bit	0 or 1	status	read only

The PTOX ED (Error Detected Status) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program to detect when the PTOX instruction is in an error state. If an error state is detected, the specific error is identified in the error code register (PTOX:0.ER). The ED bit operates as follows:

- Set (1) – Whenever a PTOX instruction is in an error state
- Cleared (0) – Whenever a PTOX instruction is not in an error state

PTOX Normal Operation Status (NS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
NS - Normal Operation Status	PTOX:0/NS	bit	0 or 1	status	read only

The PTOX NS (Normal Operation Status) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program to detect when the PTOX is in its normal state. A normal state is ACCEL, RUN, DECEL or DONE, with no PTOX errors. The NS bit operates as follows:

- Set (1) – Whenever a PTOX instruction is in its normal state
- Cleared (0) – Whenever a PTOX instruction is not in its normal state

PTOX Enable Hard Stop (EH)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
EH - Enable Hard Stop	PTOX:0/EH	bit	0 or 1	control	read/write

The PTOX EH (Enable Hard Stop) bit is used to stop the PTOX sub-system immediately. Once the PTOX sub-system starts a pulse sequence, the only way to stop generating pulses is to set the enable hard stop bit. The enable hard stop aborts any PTOX sub-system operation (idle, normal, jog continuous or jog pulse) and generates a PTOX sub-system error. The EH bit operates as follows:

- Set (1) – Instructs the PTOX sub-system to stop generating pulses immediately (output off = 0)
- Cleared (0) – Normal operation

PTOX Enable Status (EN)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
EN - Enable Status (follows rung state)	PTOX:0/EN	bit	0 or 1	status	read only

The PTOX EN (Enable Status) is controlled by the PTOX sub-system. When the rung preceding the PTOX instruction is solved true, the PTOX instruction is enabled and the enable status bit is set. If the rung preceding the PTOX instruction transitions to a false state before the pulse sequence completes its operation, the enable status bit resets (0). The EN bit operates as follows:

- Set (1) – PTOX is enabled
- Cleared (0) – PTOX has completed, or the rung preceding the PTOX is false

PTOX Output Frequency (OF)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
OF - Output Frequency (Hz)	PTOX:0.OF	long word (32-bit INT)	0…100,000	control	read/write

The PTOX OF (Output Frequency) variable defines the frequency of the PTOX output during the RUN phase of the pulse profile. This value is typically determined by the type of device that is being driven, the mechanics of the application, or the device/components being moved. In the MicroLogix 1400 controller, the data less than zero or greater than 100,000 generates a PTOX error.

PTOX Operating Frequency Status (OFS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
OFS - Operating Frequency Status (Hz)	PTOX:0.OFS	long word (32-bit INT)	0…100,000	status	read only

The PTOX OFS (Output Frequency Status) is generated by the PTOX sub-system and can be used in the control program to monitor the actual frequency being produced by the PTOX sub-system.

TIP The value displayed may not exactly match the value entered in the PTOX:0.OF. This is because the PTOX sub-system may not be capable of reproducing an exact frequency at some of the higher frequencies. For PTOX applications, this is typically not an issue because, in all cases, an exact number of pulses are produced.

PTOX Total Output Pulses To Be Generated (TOP)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
TOP - Total Output Pulses To Be Generated	PTOX:0.TOP	long word (32-bit INT)	0…2,147,483,647	control	read/write

The PTOX TOP (Total Output Pulses) defines the total number of pulses to be generated for the pulse profile (accel/run/decel *inclusive*).

PTOX Output Pulses Produced (OPP)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
OPP - Output Pulses Produced	PTOX:0.OPP	long word (32-bit INT)	0…2,147,483,647	status	read only

The PTOX OPP (Output Pulses Produced) is generated by the PTOX sub-system and can be used in the control program to monitor how many pulses have been generated by the PTOX sub-system.

PTOX Accel/Decel Pulses Independent (ADI)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
ADI - Accel/Decel Pulses Independent	PTOX:0/ADI	bit	0 or 1	control	read/write

The PTOX ADI (Accel/Decel Pulses Independent) bit is used to define whether the acceleration and deceleration intervals will be the same, or if each will have a unique value. When this bit is set (1), separate profiles are used. When this bit is clear (0), the PTOX will operate with the deceleration profile as a mirror of the acceleration profile.

If separate acceleration and deceleration profiles are desired, you must choose a long integer file number and a starting element. There must be four long elements available in the file:

Element 1:	Acceleration Count
Element 2:	Deceleration Count
Elements 3 and 4:	Reserved

The choice of selecting a common profile or separate profiles must be made at the time of programming. This cannot be changed once the program is downloaded into the controller. The selection of the ramp type must be made prior to going to run. The acceleration and deceleration counts must be entered before the PTOX is enabled. If the four long elements are not properly identified, the controller will return a -3 error in the PTOX function file when going to run.

PTOX Accel / Decel Pulses (ADP) (ADI=0) or File:Elem (ADI=1)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
ADP - Accel/Decel Pulses	PTOX:0.ADP	long word (32-bit INT)	0…1,073,741,824 (ADI=0) 0…2,147,483,647 (ADI=1)	control	read/write

The PTOX ADP (Accel/Decel Pulses) defines how many of the total pulses (TOP variable) will be applied to each of the ACCEL and DECEL components. The ADP will determine the acceleration and deceleration rate from 0 to the PTOX Output Frequency (OF). The PTOX Output Frequency (OF) defines the operating frequency in pulses/second during the run portion of the profile.

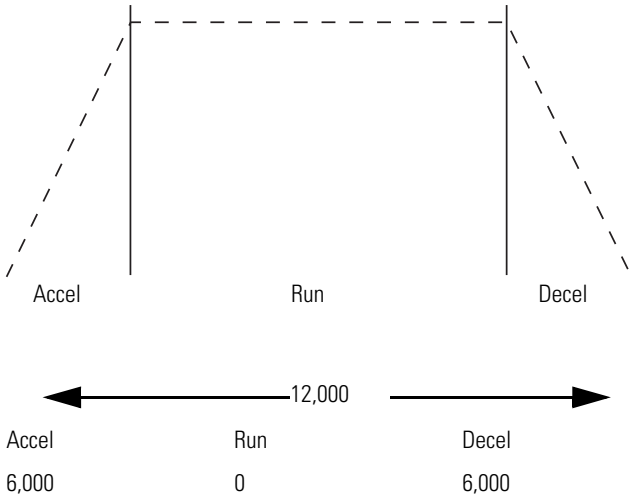
TIP When entering the ADP parameters, the PTOX will generate an Accel/Decel Error if one of the following conditions occur:

- The PTOX ADP for accel and/or decel is negative.
- The total pulses for the acceleration and deceleration phases is greater than the total output pulses to be generated (TOP).

Acceleration and deceleration values can either be identical (ADI = 0), or a unique value for each (ADI = 1).

In the example below (when ADI=0),

- TOP (total output pulses) = 12,000
- ADP (accelerate/decelerate pulses)= 6,000 (This is the maximum ADP value that may be entered without causing a fault. The run portion will equal 0.)



In this example, the maximum value that could be used for accelerate/decelerate is 6000, because if both accelerate and decelerate are 6000, the total number of pulses = 12,000. The run component would be zero. This profile would consist of an acceleration phase from 0...6000. At 6000, the output frequency (OF variable) is generated and immediately enters the deceleration phase,

6000...12,000. At 12,000, the PTOX operation would stop (output frequency = 0).

If you need to determine the ramp period (accelerate/decelerate ramp duration):

- 2 x ADP/OF = duration in seconds (OF = output frequency)

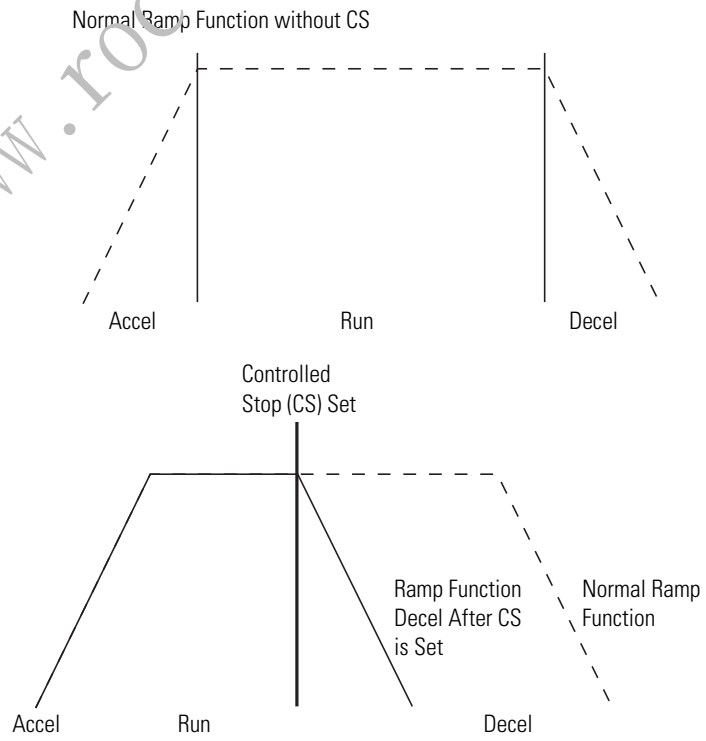
The following formulas can be used to calculate the maximum value that could be used for accelerate/decelerate for both profiles. The maximum pulses of accel/decel = the integer which is less than or equal to the result found below (OF = output frequency):

- For Trapezoid Profiles: [OF x (OF/4)] + 0.5
- For S-Curve Profiles: 0.999 x OF x SQRT(OF/6)

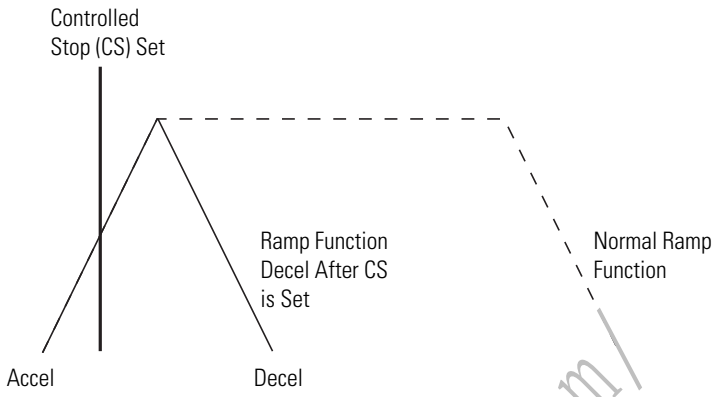
PTOX Controlled Stop (CS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
CS - Controlled Stop	PTOX:0/CS	bit	0 or 1	control	read/write

The PTOX CS (Controlled Stop) bit is used to stop an executing PTOX instruction, in the run portion of the profile, by immediately starting the decel phase. Once set, the decel phase completes without an error or fault condition.



If the CS bit is set during the accel phase, the accel phase completes and the PTOX immediately enters the decel phase.



PTOX Jog Frequency (JF)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
JF - Jog Frequency (Hz)	PTOX:0.JF	long word (32-bit INT)	0…100,000	control	read/write

The PTOX JF (Jog Frequency) variable defines the frequency of the PTOX output during all Jog phases. This value is typically determined by the type of device that is being driven, the mechanics of the application, or the device/components being moved. In the MicroLogix 1400 controller, the data less than zero or greater than 100,000 generates a PTOX error.

PTOX Jog Pulse (JP)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
JP - Jog Pulse	PTOX:0/JP	bit	0 or 1	control	read/write

The PTOX JP (Jog Pulse) bit is used to instruct the PTOX sub-system to generate a single pulse. The width is defined by the Jog Frequency parameter in the PTOX function file. Jog Pulse operation is only possible under the following conditions:

- PTOX sub-system in idle
- Jog continuous not active
- Enable not active

The JP bit operates as follows:

- Set (1) – Instructs the PTOX sub-system to generate a single Jog Pulse
- Cleared (0) – Arms the PTOX Jog Pulse sub-systemPTOX Jog Pulse Status (JPS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
JPS - Jog Pulse Status	PTOX:0/JPS	bit	0 or 1	status	read only

The PTOX JPS (Jog Pulse Status) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program to detect when the PTOX has generated a Jog Pulse.

The JPS bit operates as follows:

- Set (1) – Whenever a PTOX instruction outputs a Jog Pulse
- Cleared (0) – Whenever a PTOX instruction exits the Jog Pulse state

TIP The output (jog) pulse is normally complete with the JP bit set. The JPS bit remains set until the JP bit is cleared (0 = off).

PTOX Jog Continuous (JC)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
JC - Jog Continuous	PTOX:0/JC	bit	0 or 1	control	read/write

The PTOX JC (Jog Continuous) bit instructs the PTOX sub-system to generate continuous pulses. The frequency generated is defined by the Jog Frequency parameter in the PTOX function file. Jog Continuous operation is only possible under the following conditions:

- PTOX sub-system in idle
- Jog Pulse not active
- Enable not active

The JC bit operates as follows:

- Set (1) – Instructs the PTOX sub-system to generate continuous Jog Pulses
- Cleared (0) – The PTOX sub-system does not generate Jog Pulses

When the Jog Continuous bit is cleared, the current output pulse is truncated.

PTOX Jog Continuous Status (JCS)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
JCS - Jog Continuous Status	PTOX:0/JCS	bit	0 or 1	status	read only

The PTOX JCS (Jog Continuous Status) bit is controlled by the PTOX sub-system. It can be used by an input instruction on any rung within the control program to detect when the PTOX is generating continuous Jog Pulses. The JCS bit operates as follows:

- Set (1) – Whenever a PTOX instruction is generating continuous Jog Pulses
- Cleared (0) – Whenever a PTOX instruction is not generating continuous Jog Pulses.

PTOX Error Code (ER)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
ER - Error Code	PTOX:0.ER	word (INT)	-3…7	status	read only

PTOX ER (Error Codes) detected by the PTOX sub-system are displayed in this register. The error codes are shown in the table below:

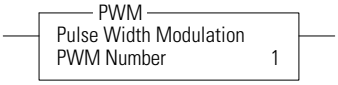
Pulse Train Output Error Codes

Error Code	Non-User Fault	Recoverable Fault	Instruction Errors	Error Name	Description
-3	No	Yes	Yes	Undefined Accel/ Decel	Acceleration Count and Deceleration not defined during going to run mode when Accel/Decel Pulses Independent (ADI) is set (1).
-2	Yes	No	No	Overlap Error	An output overlap is detected. Multiple functions are assigned to the same physical output. This is a configuration error. The controller faults and the User Fault Routine does not execute. Example: PTO0 and PTO1 are both attempting to use a single output.
-1	Yes	No	No	Output Error	An invalid output has been specified. Output 2, output 3 and output 4 are the only valid choices. This is a configuration error. The controller faults and the User Fault Routine does not execute.
0	---	---		Normal	Normal (0 = no error present)
1	No	No	Yes	Hardstop Detected	This error is generated whenever a hard stop is detected. This error does not fault the controller. To clear this error, scan the PTOX instruction on a false rung and reset the EH (Enable Hard Stop) bit to 0.
2	No	No	Yes	Output Forced Error	The configured PTOX output (2, 3 or 4) is currently forced. The forced condition *must* be removed for the PTOX to operate. This error does not fault the controller. It is automatically cleared when the force condition is removed.
3	No	Yes	No	Frequency Error	The operating frequency value (OFS) is less than 0 or greater than 100,000. This error faults the controller. It can be cleared by logic within the User Fault Routine.
4	No	Yes	No	Accel/ Decel Error	The accelerate/decelerate parameters (ADP) are: • less than zero • greater than half the total output pulses to be generated (TOP) • Accel/Decel exceeds limit (See page 123.) This error faults the controller. It can be cleared by logic within the User Fault Routine.

Pulse Train Output Error Codes

Error Code	Non-User Fault	Recoverable Fault	Instruction Errors	Error Name	Description
5	No	No	Yes	Jog Error	PTOX is in the idle state and two or more of the following are set: • Enable (EN) bit set • Jog Pulse (JP) bit set • Jog Continuous (JC) bit set This error does not fault the controller. It is automatically cleared when the error condition is removed.
6	No	Yes	No	Jog Frequency Error	The jog frequency (JF) value is less than 0 or greater than 100,000. This error faults the controller. It can be cleared by logic within the User Fault Routine.
7	No	Yes	No	Length Error	The total output pulses to be generated (TOP) is less than zero. This error faults the controller. It can be cleared by logic within the User Fault Routine.

PWM – Pulse Width Modulation



IMPORTANT The PWM function can only be used with the controller's embedded I/O. It cannot be used with expansion I/O modules.

IMPORTANT The PWM instruction should only be used with MicroLogix 1400 BXB or BXBA unit. Relay outputs are not capable of performing very high-speed operations.

Instruction Type: output

Execution Time for the PWM Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	13.2160 µs	7.1710 µs

PWM Function

The PWM function allows a field device to be controlled by a PWM wave form. The PWM profile has two primary components:

- Frequency to be generated
- Duty Cycle interval

The PWM instruction, along with the HSC and PTO functions, are different than all other controller instructions. Their operation is performed by custom

circuitry that runs in parallel with the main system processor. This is necessary because of the high performance requirements of these instructions.

The interface to the PWM sub-system is accomplished by scanning a PWM instruction in the main program file (file number 2), or by scanning a PWM instruction in any of the subroutine files. A typical operating sequence of a PWM instruction is as follows:

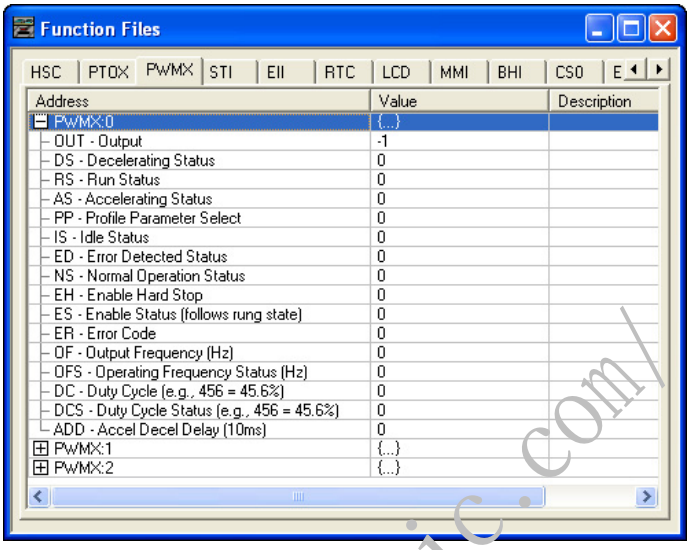
1. The rung that a PWM instruction is on is solved true (the PWM is started).
2. A waveform at the specified frequency is produced.
3. The RUN phase is active. A waveform at the specified frequency with the specified duty cycle is output.
4. The rung that the PWM is on is solved false.
5. The PWM instruction is IDLE.

While the PWM instruction is being executed, status bits and data are updated as the main controller continues to operate. Because the PWM instruction is actually being executed by a parallel system, the status bits and other information are updated each time the PWM instruction is scanned while it is running. This provides the control program access to PWM status while it is running.

TIP PWM status is only as fresh as the scan time of the controller. Worst case latency is the maximum scan of the controller. This condition can be minimized by placing a PWM instruction in the STI (selectable timed interrupt) file, or by adding PWM instructions to your program to increase how often a PWM instruction is scanned.

Pulse Width Modulation (PWMX) Function File

Within the PWM function file are three PWM elements. Each element can be set to control either output 2 (O0:0/2), output 3 (O0:0/3), or output 4 (O0:0/4). Function file element PWMX:0 is shown below.



Pulse Width Modulated Function File Elements Summary

The variables within each PWMX element, along with what type of behavior and access the control program has to those variables, are listed individually below.

PWM Function File Elements

Element Description	Address	Data Format	Range	Type	User Program Access	For More Information
OUT - PWMX Output	PWMX:0.OUT	word (INT)	2…4	status	read only	131
DS - Decelerating Status	PWMX:0/DS	bit	0 or 1	status	read only	131
RS - PWMX Run Status	PWMX:0/RS	bit	0 or 1	status	read only	131
AS - Accelerating Status	PWMX:0/AS	bit	0 or 1	status	read only	132
PP - Profile Parameter Select	PWMX:0/PP	bit	0 or 1	control	read/write	132
IS - PWMX Idle Status	PWMX:0/IS	bit	0 or 1	status	read only	132
ED - PWMX Error Detection	PWMX:0/ED	bit	0 or 1	status	read only	133
NS - PWMX Normal Operation	PWMX:0/NS	bit	0 or 1	status	read only	133
EH - PWMX Enable Hard Stop	PWMX:0/EH	bit	0 or 1	control	read/write	133
ES - PWMX Enable Status	PWMX:0/ES	bit	0 or 1	status	read only	133
OF - PWMX Output Frequency	PWMX:0.OF	long word (32-bit INT)	0…40,000	control	read/write	134
OFS - PWMX Operating Frequency Status	PWMX:0.OFS	long word (32-bit INT)	0…40,000	status	read only	134
DC - PWMX Duty Cycle	PWMX:0.DC	word (INT)	1…1000	control	read/write	134
DCS - PWMX Duty Cycle Status	PWMX:0.DCS	word (INT)	1…1000	status	read only	135
ADD - Accel/Decel Delay	PWMX:0.ADD	word (INT)	0…32,767	control	read/write	135
ER - PWMX Error Codes	PWMX:0.ER	word (INT)	-2…5	status	read only	135

PWMX Output (OUT)

Element Description	Address	Data Format	Range	Type	User Program Access
OUT - PWMX Output	PWMX:0.OUT	word (INT)	2…4	status	read only

The PWMX OUT (Output) variable defines the physical output that the PWMX instruction controls. This variable is set within the function file folder when the control program is written and cannot be set by the user program. The outputs are defined as O0:0/2, O0:0/3 or O0:0/4 as listed below:

- O0:0.0/2: PWMX modulates output 2 of the embedded outputs.
- O0:0.0/3: PWMX modulates output 3 of the embedded outputs.
- O0:0.0/4: PWMX modulates output 4 of the embedded outputs.

PWMX Decelerating Status (DS)

Element Description	Address	Data Format	Range	Type	User Program Access
DS - Decelerating Status	PWM:0/DS	bit	0 or 1	status	read only

The PWMX DS (Decel) bit is controlled by the PWMX sub-system. It can be used by an input instruction on any rung within the control program. The DS bit operates as follows:

- Set (1) - Whenever a PWMX output is within the deceleration phase of the output profile.
- Cleared (0) - Whenever a PWMX output is not within the deceleration phase of the output profile.

PWMX Run Status (RS)

Element Description	Address	Data Format	Range	Type	User Program Access
RS - PWMX Run Status	PWMX:0/RS	bit	0 or 1	status	read only

The PWMX RS (Run Status) bit is controlled by the PWMX sub-system. It can be used by an input instruction on any rung within the control program.

- Set (1) - Whenever the PWMX instruction is within the run phase of the output profile.
- Cleared (0) - Whenever the PWMX instruction is not within the run phase of the output profile.

PWMX Accelerating Status (AS)

Element Description	Address	Data Format	Range	Type	User Program Access
AS - Accelerating Status	PWM:0/AS	bit	0 or 1	status	read only

The PWMX AS (Accelerating Status) bit is controlled by the PWMX sub-system. It can be used by an input instruction on any rung within the control program. The AS bit operates as follows:

- Set (1) – Whenever a PWMX output is within the acceleration phase of the output profile.
- Cleared (0) – Whenever a PWMX output is not within the acceleration phase of the output profile.

PWMX Profile Parameter Select (PP)

Element Description	Address	Data Format	Range	Type	User Program Access
PP - Profile Parameter Select	PWM:0/PP	bit	0 or 1	control	read/write

The PWMX PP (Profile Parameter Select) selects which component of the waveform is modified during a ramp phase:

- Set (1) – selects Frequency
- Cleared (0) – selects Duty Cycle

The PWMX PP bit cannot be modified while the PWMX output is running/enabled. See PWMX ADD on page 135 for more information.

PWMX Idle Status (IS)

Element Description	Address	Data Format	Range	Type	User Program Access
IS - PWMX Idle Status	PWMX:0/IS	bit	0 or 1	status	read only

The PWMX IS (Idle Status) is controlled by the PWMX sub-system and represents no PWMX activity. It can be used in the control program by an input instruction.

- Set (1) – PWMX sub-system is in an idle state.
- Cleared (0) – PWMX sub-system is not in an idle state (it is running).

PWMX Error Detected (ED)

Element Description	Address	Data Format	Range	Type	User Program Access
ED - PWMX Error Detection	PWMX:0/ED	bit	0 or 1	status	read only

The PWMX ED (Error Detected) bit is controlled by the PWMX sub-system. It can be used by an input instruction on any rung within the control program to detect when the PWMX instruction is in an error state. If an error state is detected, the specific error is identified in the error code register (PWMX:0.ER).

- Set (1) – Whenever a PWMX instruction is in an error state.
- Cleared (0) – Whenever a PWMX instruction is not in an error state.

PWMX Normal Operation (NS)

Element Description	Address	Data Format	Range	Type	User Program Access
NS - PWMX Normal Operation	PWMX:0/NS	bit	0 or 1	status	read only

The PWMX NS (Normal Operation) bit is controlled by the PWMX sub-system. It can be used by an input instruction on any rung within the control program to detect when the PWMX is in its normal state. A normal state is defined as ACCEL, RUN, or DECEL with no PWMX errors.

- Set (1) – Whenever a PWMX instruction is in its normal state.
- Cleared (0) – Whenever a PWMX instruction is not in its normal state.

PWMX Enable Hard Stop (EH)

Element Description	Address	Data Format	Range	Type	User Program Access
EH - PWMX Enable Hard Stop	PWMX:0/EH	bit	0 or 1	control	read/write

The PWMX EH (Enable Hard Stop) bit stops the PWMX sub-system immediately. A PWMX hard stop generates a PWMX sub-system error.

- Set (1) – Instructs the PWMX sub-system to stop its output modulation immediately (output off = 0).
- Cleared (0) – Normal operation.

PWMX Enable Status (ES)

Element Description	Address	Data Format	Range	Type	User Program Access
ES - PWMX Enable Status	PWMX:0/ES	bit	0 or 1	status	read only

The PWMX ES (Enable Status) is controlled by the PWMX sub-system. When the rung preceding the PWMX instruction is solved true, the PWMX

instruction is enabled, and the enable status bit is set. When the rung preceding the PWMX instruction transitions to a false state, the enable status bit is reset (0) immediately.

- Set (1) – PWMX is enabled.
- Cleared (0) – PWMX has completed or the rung preceding the PWMX is false.

PWMX Output Frequency (OF)

Element Description	Address	Data Format	Range	Type	User Program Access
OF - PWMX Output Frequency	PWMX:0.OF	long word (32-bit INT)	0…40,000	control	read/write

The PWMX OF (Output Frequency) variable defines the frequency of the PWMX function. This frequency can be changed at any time. In the MicroLogix 1400 controller, the data less than zero or greater than 40,000 generates a PWMX error.

PWMX Operating Frequency Status (OFS)

Element Description	Address	Data Format	Range	Type	User Program Access
OFS - PWMX Operating Frequency Status	PWMX:0.OFS	long word (32-bit INT)	0…40,000	status	read only

The PWMX OFS (Output Frequency Status) is generated by the PWMX sub-system and can be used in the control program to monitor the actual frequency produced by the PWMX sub-system.

PWMX Duty Cycle (DC)

Element Description	Address	Data Format	Range	Type	User Program Access
DC - PWMX Duty Cycle	PWMX:0.DC	word (INT)	1…1000	control	read/write

The PWMX DC (Duty Cycle) variable controls the output signal produced by the PWMX sub-system. Changing this variable in the control program changes the output waveform. Typical values and output waveform:

- DC = 1000: 100% Output ON (constant, no waveform)
- DC = 750: 75% Output ON, 25% output OFF
- DC = 500: 50% Output ON, 50% output OFF
- DC = 250: 25% Output ON, 75% output OFF
- DC = 0: 0% Output OFF (constant, no waveform)

PWMX Duty Cycle Status (DCS)

Element Description	Address	Data Format	Range	Type	User Program Access
DCS - PWMX Duty Cycle Status	PWMX:0.DCS	word (INT)	1…1000	status	read only

The PWMX DCS (Duty Cycle Status) provides feedback from the PWMX sub-system. The Duty Cycle Status variable can be used within an input instruction on a rung of logic to provide PWMX system status to the remaining control program.

PWMX Accel/Decel Delay (ADD)

Element Description	Address	Data Format	Range	Type	User Program Access
ADD - Accel/Decel Delay	PWMX:0.ADD	word (INT)	0…32,767	control	read/write

PWMX ADD (Accel/Decel Delay) defines the amount of time in 10 millisecond interval to ramp from zero to 20 kHz frequency. Also specifies the time to ramp down to zero.

The PWMX ADD value is loaded and activated immediately (whenever the PWMX instruction is scanned on a true rung of logic). This allows multiple steps or stages of acceleration or deceleration to occur.

PWMX Error Code (ER)

Element Description	Address	Data Format	Range	Type	User Program Access
ER - PWMX Error Codes	PWMX:0.ER	word (INT)	-2…5	status	read only

PWMX ER (Error Codes) detected by the PWMX sub-system are displayed in this register. The table identifies known errors.

PWMX Error Codes

Error Code	Non-User Fault	Recoverable Fault	Instruction Errors	Error Name	Description
-2	Yes	No	No	Overlap Error	An output overlap is detected. Multiple functions are assigned to the same physical output. This is a configuration error. The controller faults and the User Fault Routine does not execute. Example: PWM0 and PWM1 are both attempting to use a single output.
-1	Yes	No	No	Output Error	An invalid output has been specified. Output 2, output 3, and output 4 are the only valid choices. This is a configuration error. The controller faults and the User Fault Routine does not execute.
0				Normal	Normal (0 = no error present)
1	No	No	Yes	Hardstop Error	This error is generated whenever a hardstop is detected. This error does not fault the controller. It is automatically cleared when the hardstop condition is removed.

PWMX Error Codes

2	No	No	Yes	Output Forced Error	The configured PWMX output (2, 3, or 4) is currently forced. The forced condition *must* be removed for the PWMX to operate. This error does not fault the controller. It is automatically cleared when the force condition is removed.
3	Yes	Yes	No	Frequency Error	The frequency value is less than 0 or greater than 40,000. This error faults the controller. It can be cleared by logic within the User Fault Routine.
4	Reserved				
5	Yes	Yes	No	Duty Cycle Error	The PWMX duty cycle is either less than zero or greater than 1000. This error faults the controller. It can be cleared by logic within the User Fault Routine.

Relay-Type (Bit) Instructions

Use relay-type (bit) instructions to monitor and/or control bits in a data file or function file, such as input bits or timer control-word bits. The following instructions are described in this chapter:

Instruction	Used To:	Page
XIC - Examine if Closed	Examine a bit for an ON condition	137
XIO - Examine if Open	Examine a bit for an OFF condition	137
OTE - Output Enable	Turn ON or OFF a bit (non-retentive)	139
OTL - Output Latch	Latch a bit ON (retentive)	140
OTU - Output Unlatch	Unlatch a bit OFF (retentive)	140
ONS - One Shot	Detect an OFF to ON transition	141
OSR - One Shot Rising	Detect an OFF to ON transition	142
OSF - One Shot Falling	Detect an ON to OFF transition	142

These instructions operate on a single bit of data. During operation, the processor may set or reset the bit, based on logical continuity of ladder rungs. You can address a bit as many times as your program requires.

XIC - Examine if Closed
XIO - Examine if Open

Instruction Type: input

Execution Time for the XIC Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	0.2646 µs	0.2512 µs

Execution Time for the XIO Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	0.2513 µs	0.2775 µs

Use the XIC instruction to determine if the addressed bit is on. Use the XIO instruction to determine if the addressed bit is off.

When used on a rung, the bit address being examined can correspond to the status of real world input devices connected to the base unit or expansion I/O, or internal addresses (data or function files). Examples of devices that turn on or off:

- a push button wired to an input (addressed as I1:0/4)
- an output wired to a pilot light (addressed as O0:0/2)
- a timer controlling a light (addressed as T4:3/DN)
- a bit in the bit file (addressed as B3/16)

The instructions operate as follows:

XIO and XIC Instruction Operation

Rung State	Addressed Bit	XIC Instruction	XIO Instruction
True	Off	Returns a False	Returns a True
True	On	Returns a True	Returns a False
False	--	Instruction is not evaluated	Instruction is not evaluated

Addressing Modes and File Types can be used as shown in the following table:

XIC and XIO Instructions Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Operand Bit	•	•	•	•	•	•			•	•			•	•	•	•	•	•	•	•	•	•	•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, LCD, and DLS files.

OTE - Output Energize

Instruction Type: output

Execution Time for the OTE Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.2685 µs	0.2629 µs

Use an OTE instruction to turn a bit location on when rung conditions are evaluated as true and off when the rung is evaluated as false. An example of a device that turns on or off is an output wired to a pilot light (addressed as O0:0/4). OTE instructions are reset (turned OFF) when:

- You enter or return to the program or remote program mode or power is restored.
- The OTE is programmed within an inactive or false Master Control Reset (MCR) zone.

TIP A bit that is set within a subroutine using an OTE instruction remains set until the OTE is scanned again.

ATTENTION: If you enable interrupts during the program scan via an OTL, OTE, or UIE, this instruction *must* be the *last* instruction executed on the rung (last instruction on last branch). It is recommended this be the only output instruction on the rung.

ATTENTION: Never use an output address at more than one place in your logic program. Always be fully aware of the load represented by the output coil.

Addressing Modes and File Types can be used as shown in the following table:

OTE Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Destination Bit	•	•	•	•	•	•			•	•			•	•	•	•	•						•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, LCD, CS, IOS, and DLS files.

OTL - Output Latch
OTU - Output Unlatch

Instruction Type: output

Execution Time for the OTL and OTU Instructions

Controller	OTL - When Rung Is:		OTU - When Rung Is:	
	True	False	True	False
MicroLogix 1400	0.2541 µs	0.1882 µs	0.2830 µs	0.1732 µs

The OTL and OTU instructions are retentive output instructions. OTL turns on a bit, while OTU turns off a bit. These instructions are usually used in pairs, with both instructions addressing the same bit.

ATTENTION: If you enable interrupts during the program scan via an OTL, OTE, or UIE, this instruction *must* be the *last* instruction executed on the rung (last instruction on last branch). It is recommended this be the only output instruction on the rung.

Since these are latching outputs, once set (or reset), they remain set (or reset) regardless of the rung condition.

ATTENTION: In the event of a power loss, any OTL controlled bit (including field devices) energizes with the return of power if the OTL bit was set when power was lost.

ATTENTION: Under error conditions, physical outputs are turned off. Once the error conditions are cleared, the controller resumes operation using the data table value.

Addressing Modes and File Types can be used as shown in the following table:

OTL and OTU Instructions Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files(1)										Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Operand Bit	•	•	•	•	•	•			•	•			•	•	•	•	•						•		•	•	•			

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, LCD, and DLS files.

ONS - One Shot

N7:1
ONS
0

Instruction Type: input

Execution Time for the ONS Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.2776 µs	0.3110 µs

TIP The ONS instruction for the MicroLogix 1400 provides the same functionality as the OSR instruction for the MicroLogix 1000 and SLC 500 controllers.

The ONS instruction is a retentive input instruction that triggers an event to occur one time. After the false-to-true rung transition, the ONS instruction remains true for one program scan. The output then turns OFF and remains OFF until the logic preceding the ONS instruction is false (this re-activates the ONS instruction).

The ONS Storage Bit is the bit address that remembers the rung state from the previous scan. This bit is used to remember the false-to-true rung transition.

ONS Instruction Operation

Rung Transition	Storage Bit	Rung State after Execution
false-to-true (one scan)	storage bit is set	true
true-to-true	storage bit remains set	false
true-to-false, false-to-false	storage bit is cleared	false

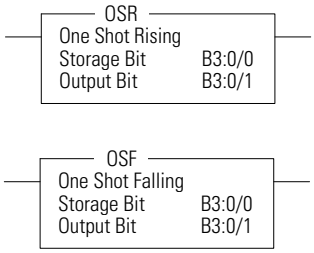
Addressing Modes and File Types can be used as shown in the following table:

ONS Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	NMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Storage Bit				•		•																			•		•			

OSR - One Shot Rising
OSF - One Shot Falling



Instruction Type: output

Execution Time for the OSR and OSF Instructions

Controller	OSR - When Rung Is:		OSF - When Rung Is:	
	True	False	True	False
MicroLogix 1400	1.3766 µs	1.3724 µs	1.3672 µs	2.0952 µs

TIP The OSR instruction for the MicroLogix 1400 *does not* provide the same functionality as the OSR instruction for the MicroLogix 1000 and SLC 500 controllers. For the same functionality as the OSR instruction for the MicroLogix 1000 and SLC 500 controllers, use the ONS instruction.

Use the OSR and OSF instructions to trigger an event to occur one time. These instructions trigger an event based on a change of rung state, as follows:

- Use the OSR instruction when an event must start based on the false-to-true (rising edge) change of state of the rung.
- Use the OSF instruction when an event must start based on the true-to-false (falling edge) change of state of the rung.

These instructions use two parameters, Storage Bit and Output Bit.

- Storage Bit - This is the bit address that remembers the rung state from the previous scan.
- Output Bit - This is the bit address which is set based on a false-to-true (OSR) or true-to-false (OSF) rung transition. The Output Bit is set for one program scan.

To re-activate the OSR, the rung must become false. To re-activate the OSF, the rung must become true.

OSR Storage and Output Bit Operation

Rung State Transition	Storage Bit	Output Bit
false-to-true (one scan)	bit is set	bit is set
true-to-true	bit is set	bit is reset
true-to-false and false-to-false	bit is reset	bit is reset

OSF Storage and Output Bits Operation

Rung State Transition	Storage Bit	Output Bit
true-to-false (one scan)	bit is reset	bit is set
false-to-false	bit is reset	bit is reset
false-to-true and true-to-true	bit is set	bit is reset

Addressing Modes and File Types can be used as shown in the following table:

OSR and OSF Instructions Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Storage Bit			•			•																			•		•			
Output Bit	•		•		•	•			•																•		•			

Notes:

Timer and Counter Instructions

Timers and counters are output instructions that let you control operations based on time or a number of events. The following Timer and Counter Instructions are described in this chapter:

Instruction	Used To:	Page
TON - Timer, On-Delay	Delay turning on an output after a true rung	148
TOF - Timer, Off-Delay	Delay turning off an output on a false rung	149
RTO - Retentive Timer On	Delay turning on an output from a true rung. The accumulator is retentive.	150
CTU - Count Up	Count up	153
CTD - Count Down	Count down	153
RES - Reset	Reset the RTO and counter's ACC and status bits (not used with TOF timers).	154

For information on using the High-Speed Counter output(s), see Using the High-Speed Counter and Programmable Limit Switch on page 77.

Timer Instructions Overview

Timers in a controller reside in a timer file. A timer file can be assigned as any unused data file. When a data file is used as a timer file, each timer element within the file has three sub-elements. These sub-elements are:

- Timer Control and Status
- Preset - This is the value that the timer must reach before the timer times out. When the accumulator reaches this value, the DN status bit is set (TON and RTO only). The preset data range is from 0...32767. The minimum required update interval is 2.55 seconds regardless of the time base.
- Accumulator - The accumulator counts the time base intervals. It represents elapsed time. The accumulator data range is from 0...32767.

Timers can be set to any one of three time bases:

Timer Base Settings

Time Base	Timing Range
0.001 seconds	0…32.767 seconds
0.01 seconds	0…327.67 seconds
1.00 seconds	0…32,767 seconds

Each timer address is made of a 3-word element. Word 0 is the control and status word, word 1 stores the preset value, and word 2 stores the accumulated value.

Timer File

Word	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN	TT	DN	Internal Use												
Word 1	Preset Value															
Word 2	Accumulated Value															

EN = Timer Enable Bit

TT = Timer Timing Bit

DN = Timer Done Bit

ATTENTION: Do not copy timer elements while the timer enable bit (EN) is set. Unpredictable machine operation may occur.

Addressing Modes and File Types can be used as shown in the following table:

Timer Instructions Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Timer					•																				•					•
Time Base																								•						•
Preset																								•				•		
Accumulator																								•				•		

(1) Valid for Timer Files only.

TIP Use an RES instruction to reset a timer's accumulator and status bits.

Timer Accuracy

Timer accuracy refers to the length of time between the moment a timer instruction is enabled and the moment the timed interval is complete.

Timer Accuracy

Time Base	Accuracy
0.001 seconds	-0.001…0.00
0.01 seconds	-0.01…0.00
1.00 seconds	-1.00…0.00

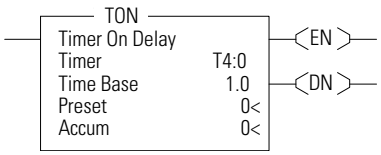
If your program scan can exceed 2.5 seconds, repeat the timer instruction on a different rung (identical logic) in a different area of the ladder code so that the rung is scanned within these limits.

Repeating Timer Instructions

Using the enable bit (EN) of a timer is an easy way to repeat its complex conditional logic at another rung in your ladder program.

TIP Timing could be inaccurate if Jump (JMP), Label (LBL), Jump to Subroutine (JSR), or Subroutine (SBR) instructions skip over the rung containing a timer instruction while the timer is timing. If the skip duration is within 2.5 seconds, no time is lost; if the skip duration exceeds 2.5 seconds, an undetectable timing error occurs. When using subroutines, a timer must be scanned at least every 2.5 seconds to prevent a timing error.

TON - Timer, On-Delay



Instruction Type: output

Execution Time for the TON Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	2.0338 µs (DN=0) 1.2608 µs (DN=1)	0.8608 µs (DN=0)

Use the TON instruction to delay turning on an output. The TON instruction begins to count time base intervals when rung conditions become true. As long as rung conditions remain true, the timer increments its accumulator until the preset value is reached. When the accumulator equals the preset, timing stops.

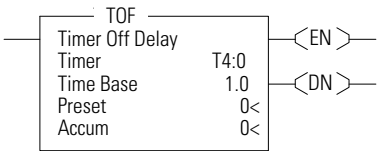
The accumulator is reset (0) when rung conditions go false, regardless of whether the timer has timed out. TON timers are reset on power cycles and mode changes.

Timer instructions use the following control and status bits:

Timer Control and Status Bits, Timer Word 0 (Data File 4 is configured as a timer file for this example.)

Bit		Is Set When:	And Remains Set Until One of the Following Occurs:
bit 13 - T4:0/DN	**DN - timer done**	accumulated value ≥ preset value	rung state goes false
bit 14 - T4:0/TT	**TT - timer timing**	rung state is true and accumulated value < preset value	• rung state goes false • DN bit is set
bit15 - T4:0/EN	**EN - timer enable**	rung state is true	rung state goes false

TOF - Timer, Off-Delay



Instruction Type: output

Execution Time for the TOF Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.5203 µs	1.0962 µs (DN=0) 0.5322 µs (DN=1)

Use the TOF instruction to delay turning off an output. The TOF instruction begins to count time base intervals when rung conditions become false. As long as rung conditions remain false, the timer increments its accumulator until the preset value is reached.

The accumulator is reset (0) when rung conditions go true, regardless of whether the timer is timed out. TOF timers are reset on power cycles and mode changes.

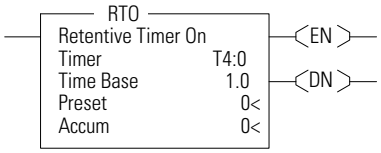
Timer instructions use the following control and status bits:

Timer Control and Status Bits, Timer Word 0 (Data File 4 is configured as a timer file for this example.)

Bit		Is Set When:	And Remains Set Until One of the Following Occurs:
bit 13 - T4:0/DN	**DN - timer done**	rung conditions are true	rung conditions go false and the accumulated value is greater than or equal to the preset value
bit 14 - T4:0/TT	**TT - timer timing**	rung conditions are false and accumulated value is less than the preset value	rung conditions go true or when the done bit is reset
bit15 - T4:0/EN	**EN - timer enable**	rung conditions are true	rung conditions go false

ATTENTION: Because the RES instruction resets the accumulated value and status bits, do not use the RES instruction to reset a timer address used in a TOF instruction. If the TOF accumulated value and status bits are reset, unpredictable machine operation may occur.

RTO - Retentive Timer, On-Delay



Instruction Type: output

Execution Time for the RTO Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	1.1710 µs (DN=0) 0.6100 µs (DN=1)	0.5480 µs

Use the RTO instruction to delay turning "on" an output. The RTO begins to count time base intervals when the rung conditions become true. As long as the rung conditions remain true, the timer increments its accumulator until the preset value is reached.

The RTO retains the accumulated value when the following occur:

- rung conditions become false
- you change the controller mode from run or test to program
- the processor loses power
- a fault occurs

When you return the controller to the RUN or TEST mode, and/or the rung conditions go true, timing continues from the retained accumulated value. RTO timers are retained through power cycles and mode changes.

Timer instructions use the following control and status bits:

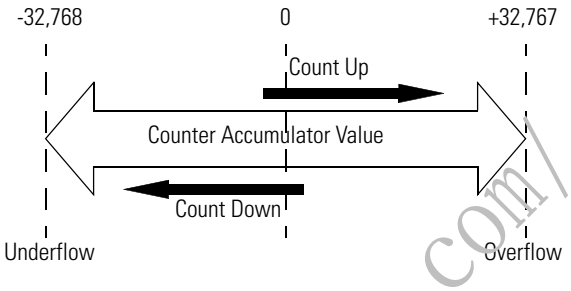
Counter Control and Status Bits, Timer Word 0 (Data File 4 is configured as a timer file for this example.)

Bit		Is Set When:	And Remains Set Until One of the Following Occurs:
bit 13 - T4:0/DN	**DN - timer done**	accumulated value ≥ preset value	the appropriate RES instruction is enabled
bit 14 - T4:0/TT	**TT - timer timing**	rung state is true and accumulated value < preset value	• rung state goes false, or • DN bit is set
bit15 - T4:0/EN	**EN - timer enable**	rung state is true	rung state goes false

To reset the accumulator of a retentive timer, use an RES instruction. See RES - Reset on page 154.

How Counters Work

The figure below demonstrates how a counter works. The count value must remain in the range of -32,768...+32,767. If the count value goes above +32,767, the counter status overflow bit (OV) is set (1). If the count goes below -32,768, the counter status underflow bit (UN) is set (1). A reset (RES) instruction is used to reset (0) the counter.



Using the CTU and CTD Instructions

Counter instructions use the following parameters:

- Counter - This is the address of the counter within the data file. All counters are 3-word data elements. Word 0 contains the Control and Status Bits, Word 1 contains the Preset, and Word 2 contains the Accumulated Value.

Word	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	CU	CD	DN	OV	UN	Not Used										
Word 1	Preset Value															
Word 2	Accumulated Value															

CU = Count Up Enable Bit

CD = Count Down Enable Bit

DN = Count Done Bit

OV = Count Overflow Bit

UN = Count Underflow Bit

- Preset - When the accumulator reaches this value, the DN bit is set. The preset data range is from -32768...32767.
- Accumulator - The accumulator contains the current count. The accumulator data range is from -32768...32767.

  The accumulated value is incremented (CTU) or decremented (CTD) on each false-to-true rung transition. The accumulated value is retained when the rung condition again becomes false, and when power is cycled on the

controller. The accumulated count is retained until cleared by a reset (RES) instruction that has the same address as the counter.

TIP The counter continues to count when the accumulator is greater than the CTU preset and when the accumulator is less than the CTD preset.

Addressing Modes and File Types can be used as shown in the following table:

CTD and CTU Instructions Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]												Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Counter					•																				•					•
Preset																								•				•		
Accumulator																								•				•		

(1) Valid for Counter Files only.

Using Counter File Control and Status Bits

Like the accumulated value, the counter status bits are also retentive until reset, as described below.

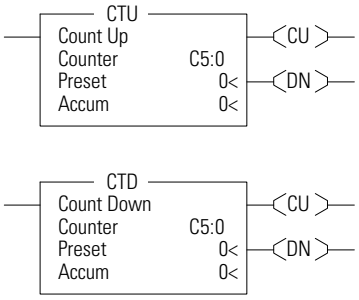
CTU Instruction Counter Control and Status Bits, Counter Word 0 (Data File 5 is configured as a timer file for this example.)

Bit		Is Set When:	And Remains Set Until One of the Following Occurs:
bit 12 - C5:0/OV	**OV - overflow indicator**	the accumulated value wraps from +32,767…-32,768 and continues to count up	a RES instruction with the same address as the CTU instruction is enabled
bit 13 - C5:0/DN	**DN - done indicator**	accumulated value ≥ preset value	• accumulated value < preset value or, • a RES instruction with the same address as the CTU instruction is enabled
bit 15 - C5:0/CU	**CU - count up enable**	rung state is true	• rung state is false • a RES instruction with the same address as the CTU instruction is enabled

CTD Instruction Counter Control and Status Bits, Counter Word 0
(Data File 5 is configured as a timer file for this example.)

Bit		Is Set When:	And Remains Set Until One of the Following Occurs:
bit 11 - C5:0/UN	**UN - underflow indicator**	the accumulated value wraps from -32,768…+32,767 and continues to count down	a RES instruction with the same address as the CTD instruction is enabled
bit 13 - C5:0/DN	**DN - done indicator**	accumulated value ≥ preset value	• accumulated value < preset value or, • a RES instruction with the same address as the CTU instruction is enabled
bit 14 - C5:0/CD	**CD - count down enable**	rung state is true	• rung state is false • a RES instruction with the same address as the CTD instruction is enabled

CTU - Count Up
CTD - Count Down



Instruction Type: output

Execution Time for the CTU and CTD Instructions

Controller	CTU - When Rung Is:		CTD - When Rung Is:	
	True	False	True	False
MicroLogix 1400	0.4849 µs	0.3812 µs	0.4350 µs	0.3803 µs

The CTU and CTD instructions are used to increment or decrement a counter at each false-to-true rung transition. When the CTU rung makes a false-to-true transition, the accumulated value is incremented by one count. The CTD instruction operates the same, except the count is decremented.

TIP If the signal is coming from a field device wired to an input on the controller, the on and off duration of the incoming signal must not be less than twice the controller scan time (assuming 50% duty cycle). This condition is needed to enable the counter to detect false-to-true transitions from the incoming device.

RES - Reset

Instruction Type: output

Execution Time for the RES Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.6320 µs	0.4305 µs

The RES instruction resets timers, counters, and control elements. When the RES instruction is executed, it resets the data defined by the RES instruction.

The RES instruction has no effect when the rung state is false. The following table shows which elements are modified:

RES Instruction Operation

When using a RES instruction with a:		
Timer Element	**Counter Element**	**Control Element**
The controller resets the: ACC value to 0 DN bit TT bit EN bit	The controller resets the: ACC value to 0 OV bit UN bit DN bit CU bit CD bit	The controller resets the: POS value to 0 EN bit EU bit DN bit EM bit ER bit UL bit

ATTENTION: Because the RES instruction resets the accumulated value and status bits, do not use the RES instruction to reset a timer address used in a TOF instruction. If the TOF accumulated value and status bits are reset, unpredictable machine operation or injury to personnel may occur.

Addressing Modes and File Types can be used as shown in the following table:

RES Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files													Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Structure					•																				•					•

Compare Instructions

Use these input instructions when you want to compare values of data.

Instruction	Used To:	Page
EQU - Equal	Test whether two values are equal (=)	156
NEQ - Not Equal	Test whether one value is not equal to a second value (≠)	156
LES - Less Than	Test whether one value is less than a second value (<)	157
LEQ - Less Than or Equal To	Test whether one value is less than or equal to a second value (≤)	157
GRT - Greater Than	Test whether one value is greater than a second value (>)	157
GEQ - Greater Than or Equal To	Test whether one value is greater than or equal to a second value (≥)	157
MEQ - Mask Compare for Equal	Test portions of two values to see whether they are equal	158
LIM - Limit Test	Test whether one value is within the range of two other values	159

Using the Compare Instructions

Most of the compare instructions use two parameters, Source A and Source B (MEQ and LIM have an additional parameter and are described later in this chapter). Both sources cannot be immediate values. The valid data ranges for these instructions are:

- -32,768...32,767 (word)
- -2,147,483,648...2,147,483,647 (long word)

Addressing Modes and File Types can be used as shown in the following table:

EQU, NEQ, GRT, LES, GEQ and LEQ Instructions
Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC[2]	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	
Source B	•	•	•	•	•	•	•		•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) Only use the High Speed Counter Accumulator (HSC.ACC) for Source A in GRT, LES, GEQ and LEQ instructions.

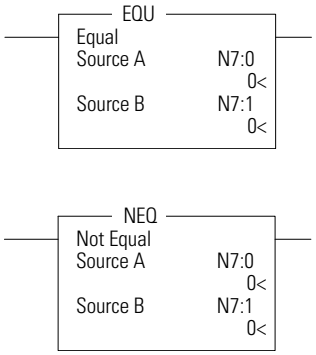
(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

When at least one of the operands is a Floating Data Point value:

- For EQU, GEQ, GRT, LEQ, and LES - If either Source is not a number (NAN), then rung state changes to false.
- For NEQ - If either Source is not a number (NAN), then rung state remains true.

EQU - Equal
NEQ - Not Equal



Instruction Type: input

Execution Time for the EQU and NEQ Instructions

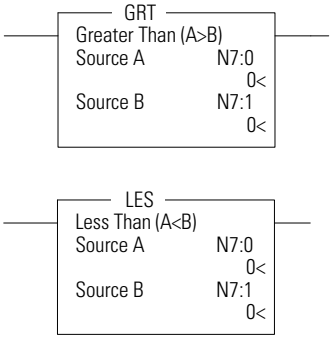
Controller	Instruction	Data Size	When Rung Is:	
			True	False
MicroLogix 1400	EQU	word	1.0814 µs	1.0854 µs
		long word	1.0674 µs	1.0828 µs
	NEQ	word	1.5056 µs	0.1880 µs
		long word	1.3892 µs	0.2070 µs

The EQU instruction is used to test whether one value is equal to a second value. The NEQ instruction is used to test whether one value is not equal to a second value.

EQU and NEQ Instruction Operation

Instruction	Relationship of Source Values	Resulting Rung State
EQU	A = B	true
	A ≠ B	false
NEQ	A = B	false
	A ≠ B	true

GRT - Greater Than
LES - Less Than



Instruction Type: input

Execution Time for the GRT and LES Instructions

Controller	Instruction	Data Size	When Rung Is:	
			True	False
MicroLogix 1400	GRT	word	1.0682 µs	0.2414 µs
		long word	1.0942 µs	0.2212 µs
	LES	word	1.0772 µs	0.2106 µs
		long word	1.0935 µs	0.2137 µs

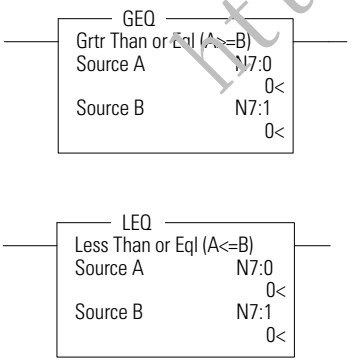
The GRT instruction is used to test whether one value is greater than a second value. The LES instruction is used to test whether one value is less than a second value.

GRT and LES Instruction Operation

Instruction	Relationship of Source Values	Resulting Rung State
GRT	A > B	true
	A ≤ B	false
LES	A ≥ B	false
	A < B	true

IMPORTANT Only use the High Speed Counter Accumulator (HSC.ACC) for Source A in GRT, LES, GEQ and LEQ instructions.

GEQ - Greater Than or Equal To
LEQ - Less Than or Equal To



Instruction Type: input

Execution Time for the GEQ and LEQ Instructions

Controller	Instruction	Data Size	When Rung Is:	
			True	False
MicroLogix 1400	GEQ	word	1.0710 µs	0.2228 µs
		long word	1.0601 µs	0.2242 µs
	LEQ	word	1.0640 µs	0.1847 µs
		long word	1.0364 µs	0.1851 µs

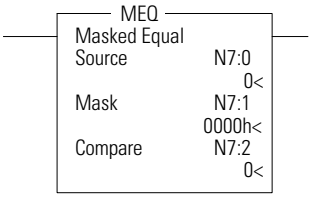
The GEQ instruction is used to test whether one value is greater than or equal to a second value. The LEQ instruction is used to test whether one value is less than or equal to a second value.

GEQ and LEQ Instruction Operation

Instruction	Relationship of Source Values	Resulting Rung State
GEQ	A ≥ B	true
	A < B	false
LEQ	A > B	false
	A ≤ B	true

IMPORTANT Only use the High Speed Counter Accumulator (HSC.ACC) for Source A in GRT, LES, GEQ and LEQ instructions.

MEQ - Mask Compare for Equal



Instruction Type: input

Execution Time for the MEQ Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	6.2730 µs	0.1934 µs
	long word	7.1602 µs	0.1780 µs

The MEQ instruction is used to compare whether one value (source) is equal to a second value (compare) through a mask. The source and the compare are logically ANDed with the mask. Then, these results are compared to each other. If the resulting values are equal, the rung state is true. If the resulting values are not equal, the rung state is false.

For example:

Source:																	Compare:															
1	1	1	1	1	0	1	0	0	0	0	0	1	1	0	0		1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Mask:																	Mask:															
1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1		1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1
Intermediate Result:																	Intermediate Result:															
1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0		1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0
Comparison of the Intermediate Results: not equal																																

The source, mask, and compare values must all be of the same data size (either word or long word). The data ranges for mask and compare are:

- -32,768...32,767 (word)
- -2,147,483,648...2,147,483,647 (long word)

The mask is displayed as a hexadecimal unsigned value from 0000...FFFF FFFF.

Addressing Modes and File Types can be used as shown in the following table:

MEQ Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

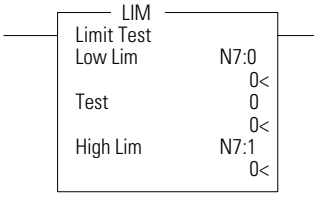
Parameter	Data Files												Function Files[1]											Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	
Mask	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	
Compare	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

LIM - Limit Test



Instruction Type: input

Execution Time for the LIM Instructions

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	7.0970 µs	0.2086 µs
	long word	7.3803 µs	0.2009 µs

The LIM instruction is used to test for values within or outside of a specified range. The LIM instruction is evaluated based on the Low Limit, Test, and High Limit values as shown in the following table.

LIM Instruction Operation Based on Low Limit, Test, and High Limit Values

When:	And:	Rung State
Low Limit ≤ High Limit	Low Limit ≤ Test ≤ High Limit	true
Low Limit ≤ High Limit	Test < Low Limit or Test > High Limit	false
High Limit < Low Limit	High Limit < Test < Low Limit	false
High Limit < Low Limit	Test ≥ High Limit or Test ≤ Low Limit	true

The Low Limit, Test, and High Limit values can be word addresses or constants, restricted to the following combinations:

- If the Test parameter is a constant, both the Low Limit and High Limit parameters must be word or long word addresses.
- If the Test parameter is a word or long word address, the Low Limit and High Limit parameters can be either a constant, a word, or a long word address.

When mixed-sized parameters are used, all parameters are put into the format of the largest parameter. For instance, if a word and a long word are used, the word is converted to a long word.

The data ranges are:

- -32,768...32,767 (word)
- -2,147,483,648...2,147,483,647 (long word)

Addressing Modes and File Types can be used as shown in the following table:

LIM Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]										Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Low Limit	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Test	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
High Limit	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 RXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Math Instructions

General Information

Before using math instructions, become familiar with the following topics at the beginning of this chapter:

- Using the Math Instructions
- Updates to Math Status Bits
- Using the Floating Point (F) Data File

This chapter also explains how advanced math instructions and application specific instructions function in your logic program. Each of the advanced math instructions include information on:

- instruction symbol
- instruction usage

Instructions

Use these output instructions to perform computations using an expression or a specific arithmetic instruction.

Math Instructions

Instruction	Used To:	Page
ADD - Add	Add two values	167
SUB - Subtract	Subtract two values	167
MUL - Multiply	Multiply two values	168
DIV - Divide	Divide one value by another	168
NEG - Negate	Change the sign of the source value and place it in the destination	168
CLR - Clear	Set all bits of a word to zero	168
ABS - Absolute Value	Find the absolute value of the source value	169
SQR - Square Root	Find the square root of a value	173
SCL - Scale	Scale a value	170
SCP - Scale with Parameters	Scale a value to a range determined by creating a linear relationship	171

Advanced Math Instructions

Instruction	Used To:	Page
SIN	Take the sine of a number and store the result in the destination.	173
COS	Take the cosine of a number and store the result in the destination.	175
TAN	Take the tangent of a number and store the result in the destination.	177
ASN	Take the arc sine of a number and store the result(in radians) in the destination.	179
ACS	Take the arc cosine of a number and store the result (in radians) in the destination.	181
ATN	Take the arc tangent of a number and store the result (in radians) in the destination.	183
DEG	Convert radians (source) to degrees and store the result in the destination.	185
RAD	Convert degrees (source) to radians and store the result in the destination.	187
LN	Take the natural log of the value in the source and store it in the destination.	189
LOG	Take the log base 10 of the value in the source and store it in the destination.	190
XPY	Raise a value to a power and stores the result in the destination.	193
CPT	Evaluate an expression and store the result in the destination.	196

Using the Math Instructions

Most math instructions use three parameters, Source A, Source B, and Destination (additional parameters are described where applicable, later in this chapter). The mathematical operation is performed using both Source values. The result is stored in the Destination.

When using math instructions, observe the following:

- Source and Destination can be different data sizes. Sources are evaluated at the highest precision (word or long word) of the operands. Then the result is converted to the size of the destination. If the signed value of the Source does not fit in the Destination, the overflow shall be handled as follows:
  - If the Math Overflow Selection Bit is clear, a saturated result is stored in the Destination. If the Source is positive, the Destination is +32,767 (word) or +2,147,483,647 (long word). If the result is negative, the Destination is -32,768 (word) or -2,147,483,648 (long word).
  - If the Math Overflow Selection Bit is set, the unsigned truncated value of the Source is stored in the Destination.
- Sources can be constants or an address, but both sources cannot be constants.
- Valid constants are -32,768...32,767 (word) and -2,147,483,648...2,147,483,647 (long word).

- Long File Type Address, Constant and Float File Type Address cannot be used together in Source A, Source B and Destination.

Addressing Modes and File Types can be used as shown in the following table:

Math Instructions (ADD, SUB, MUL, DIV, NEG, CLR) Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log[2]	Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Source B	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•			•					•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Updates to Math Status Bits

After a math instruction is executed, the arithmetic status bits in the status file are updated. The arithmetic status bits are in word 0 in the processor status file (S2).

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	sets if carry is generated; otherwise resets
S:0/1	Overflow	sets when the result of a math instruction does not fit into the destination, otherwise resets
S:0/2	Zero Bit	sets if result is zero, otherwise resets
S:0/3	Sign Bit	sets if result is negative (MSB is set), otherwise resets
S:2/14	Math Overflow Selected[1]	examines the state of this bit to determine the value of the result when an overflow occurs
S:5/0	Overflow Trap[1]	sets if the Overflow Bit is set, otherwise resets

(1) Control bits.

Overflow Trap Bit, S:5/0

Minor error bit (S:5/0) is set upon detection of a mathematical overflow or division by zero. If this bit is set upon execution of an END statement or a Temporary End (TND) instruction, the recoverable major error code 0020 is declared.

In applications where a math overflow or divide by zero occurs, you can avoid a controller fault by using an unlatch (OTU) instruction with address S:5/0 in your program. The rung must be between the overflow point and the END or TND statement.

The following illustration shows the rung you can use to unlatch the overflow trap bit.

Using the Floating Point (F) Data File

File Description

Floating point files contain IEEE-754 floating point data elements. One floating point element is shown below. You can have up to 256 of these elements in each floating point file.

Floating Point Data File Structure

Floating Point Element																															
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
S(1)	Exponent Value								Mantissa																						
High Word																Low Word														

(1) S = Sign Bit

Floating point numbers are represented using the IEEE-754 format, where:

- Bit 31 is the sign bit. This bit is set for negative numbers (note that negative zero is a valid value).
- Bits 23...30 are the exponent.
- Bits 0...22 are the mantissa.

The value represented by a 32-bit floating point number (not one of the exception values defined on page 165) is given by the following expression. Note the restoration of the suppressed most significant bit of the mantissa.

$$(-1)^{s} \times 2^{e - 127} \times (1 + m)$$

where:

s is the sign bit (0 or 1)

e is the exponent (1...254)

m is the mantissa ($0 \leq f < 1$)

The valid range for floating point numbers is from -3.4028×10^{38}...$+3.4028 \times 10^{38}$.

Definitions

Overflow - occurs when the result of an operation produces an exponent that is greater than 254.

Underflow - occurs when the result of an operation produces an exponent that is less than one.

Floating Point Exception Values

Zero - represented by an exponent and a mantissa of zero. Both positive and negative zero are valid.

Denormalized - represented by an exponent of zero and a non-zero mantissa part. Since denormalized numbers have very small, insignificant values, they are treated as zero when used as source operand for most instructions. This reduces execution time. Denormalized numbers are not generated by the instructions (but are propagated by some instructions). Zero is generated on an underflow.

Infinity - represented by an exponent of 255 and a mantissa part of zero. Both positive and negative infinity are generated when operations overflow. Infinity is propagated through calculations.

NAN (not a number) - is represented by an exponent of 255 and a non-zero mantissa part. NANs are used to indicate results that are mathematically undefined such as 0/0 and adding plus infinity to minus infinity. All operations given a NAN as input must generate a NAN as output.

LSB Round-to-Even Rule

Floating point operations are rounded using the round-to-even rule. If the bits of the result to the right of the least significant bit (LSB) represent a value less than one-half of the LSB, then the result remains as is. If the bits to the right of the LSB represent a value greater than one-half of the LSB, the result is rounded up

by adding one LSB. If the bits to the right of the LSB represent a value of exactly one-half LSB, the result is rounded up or down so that the LSB is an even number.

Addressing Floating Point Files

The addressing format for floating point data files is shown below.

Format	Explanation		
Ff:e	**F**	Floating Point file	
	f	File number	The valid file number range is from 8 (default) to 255.
	:	Element delimiter	
	e	Element number	The valid element number range is from 0…255.
Examples:	**F8:2** **F10:36**		**Floating Point File 8, Element 2** **Floating Point File 10, Element 36**

Programming Floating Point Values

The following table shows items to consider when using floating point data.

IMPORTANT These rules do not apply to the SCP instruction. See page 172 for the rules for that instruction.

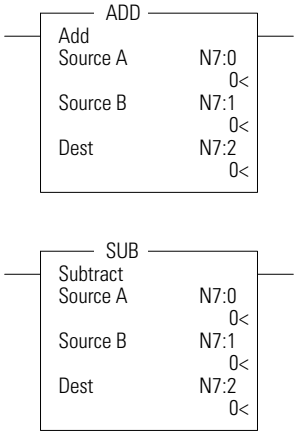
Considerations When Using Floating Point Data

When at least one of the operands is a Floating Data Point value:

- If either Source is NAN, then the result is NAN.
- All overflows result in infinity with the correct sign.
- All underflows result in plus zero.
- All denormalized Source values are treated as plus zero.
- Results are always rounded using the Round to Even rule.
- If Destination is an integer and the result is NAN or infinity, a saturated result (-32768 or +32767 for word or -2,147,836,648 or +2,147,836,647 for long word) is stored in Destination and the Math Overflow Selection Bit is ignored.
- If Destination is an integer, the rounded result is stored. If an overflow occurs after rounding, a saturated result is stored in Destination and the Math Overflow Selection Bit is ignored. The saturated results are:
  - If Destination is an integer and the result is positive, overflow Destination is +32767 (word) or +2,147,483, 648 (long word).
  - If Destination is an integer and the result is negative, overflow Destination is -32767 (word) or -2,147,483, 648 (long word).

Considerations When Using Floating Point Data
Updates to Math Status Bits: • Carry - is reset • Overflow - Is set if the result is infinity, NAN, or if a conversion to integer overflows; otherwise it is reset. • Zero - Is set if the lower 31 bits of the Floating Point Data result is all zero's, otherwise it is reset. • Sign - Is set if the most significant bit of the Destination is set (bit 15 for word, bit 31 for long word or floating point data); otherwise it is reset. • Overflow Trap - The Math Overflow Trap Bit is only set if the Overflow bit is set. Otherwise, it remains in its last state.

ADD - Add
SUB - Subtract



Instruction Type: output

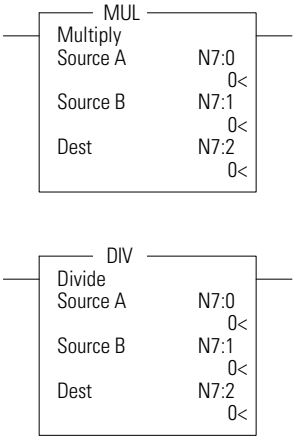
Execution Time for the ADD and SUB Instructions

Controller	Instruction	Data Size	When Rung Is:	
			True	False
MicroLogix 1400	ADD	word	1.8868 µs	0.3540 µs
		long word	1.7807 µs	0.3546 µs
	SUB	word	1.8426 µs	0.3767 µs
		long word	1.7651 µs	0.3758 µs

Use the ADD instruction to add one value to another value (Source A + Source B) and place the sum in the Destination.

Use the SUB instruction to subtract one value from another value (Source A - Source B) and place the result in the Destination.

MUL - Multiply
DIV - Divide



Instruction Type: output

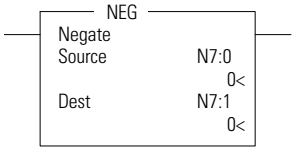
Execution Time for the MUL and DIV Instructions

Controller	Instruction	Data Size	When Rung Is:	
			True	False
MicroLogix 1400	MUL	word	3.3260 µs	0.3920 µs
		long word	3.3476 µs	0.3918 µs
	DIV	word	2.3124 µs	0.3914 µs
		long word	2.3636 µs	0.3914 µs

Use the MUL instruction to multiply one value by another value (Source A x Source B) and place the result in the Destination.

Use the DIV instruction to divide one value by another value (Source A/Source B) and place the result in the Destination. If the Sources are single words and the Destination is directly addressed to S:13 (math register), then the quotient is stored in S:14 and the remainder is stored in S:13. If long words are used, then the results are rounded.

NEG - Negate



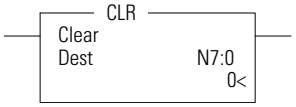
Instruction Type: output

Execution Time for the NEG Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	1.3570 µs	0.3548 µs
	long word	1.3660 µs	0.3413 µs

Use the NEG instruction to change the sign of the Source and place the result in the Destination.

CLR - Clear



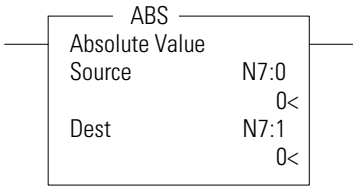
Instruction Type: output

Execution Time for the CLR Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	2.0522 µs	0.3714 µs
	long word	2.0125 µs	0.3691 µs

Use the CLR instruction to set the Destination to a value of zero.

ABS - Absolute Value



Instruction Type: output

Execution Time for the ABS Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	1.4410 µs	0.3750 µs
	long word	1.5390 µs	0.3730 µs

The ABS instruction takes the absolute value of the Source and places it in the Destination. The data range for this instruction is -2,147,483,648...2,147,483,647 or IEEE-754 floating point value.

Source and Destination do not have to be the same data type. However, if the signed result does not fit in Destination, the following will occur.

ABS Result Does Not Fit in Destination

When Both Operands Are Integers	When At Least One Operand is Floating Point Data
• If the Math Overflow Selection Bit is clear, a saturated result (32767 for word or 2,147,836,647 for long word) is stored in the Destination. • If the Math Overflow Selection Bit is set, the unsigned truncated value of the result is stored in the Destination.	• The ABS instruction clears the sign bit. No operation is performed on the remaining bits. • If Destination is an integer and Source is NAN or infinity, a saturated result (32767 for word or 2,147,836,647 for long word) is stored in Destination and the Math Overflow Selection Bit is ignored. • If Destination is an integer, the rounded result is stored. If an overflow occurs after rounding, a saturated result (32767 for word or 2,147,836,647 for long word) is stored in Destination and the Math Overflow Selection Bit is ignored.

The following table shows how the math status bits are updated upon execution of the ABS instruction:

Updates to Math Status Bits

When Both Operands Are Integers	When At Least One Operand is Floating Point Data
• Carry - Is set if input is negative, otherwise resets. • Overflow - Is set if the signed result cannot fit in the Destination; otherwise it is reset. • Zero - Is set if Destination is all zero's, otherwise it is reset. • Sign - Is set if the most significant bit of the Destination is set, otherwise it is reset. • Overflow Trap - The Math Overflow Trap Bit is only set if the Overflow bit is set. Otherwise, it remains in its last state.	• Carry - Is reset. • Overflow - Is set if the signed result is infinity, NAN, or cannot fit in the Destination; otherwise it is reset. • Zero - Is set if Destination is all zero's, otherwise it is reset. • Sign - Is set if the most significant bit of the Destination is set, otherwise it is reset. • Overflow Trap - The Math Overflow Trap Bit is only set if the Overflow bit is set. Otherwise, it remains in its last state.

Addressing Modes and File Types are shown in the following table:

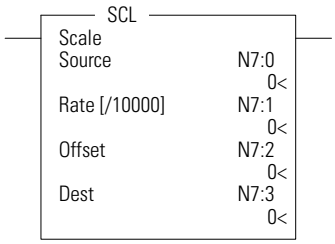
ABS Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 4-2.

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[1]			Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Floating Point	Element
Source	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•		•	•	•	
Destination	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•								•	•		•	•	•	

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX,t PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

SCL - Scale



Instruction Type: output

Execution Time for the SCL Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	10.9080 µs	0.3608 µs

The SCL instruction causes the value at the Source address to be multiplied by the Rate (slope) value. The resulting value is added to the Offset and the rounded result is placed in the Destination.

The following equations express the linear relationship between the input value and the resulting scaled value:

scaled value = [(rate x source)/10000] + offset, where

- rate = (scaled max. - scaled min.)/(input max. - input min.)
- offset = scaled min. - (input min. x rate)

Rate and Offset can both be immediate values. The data range for rate and offset is -32768...32767.

Addressing Modes and File Types can be used as shown in the following table:

SCL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

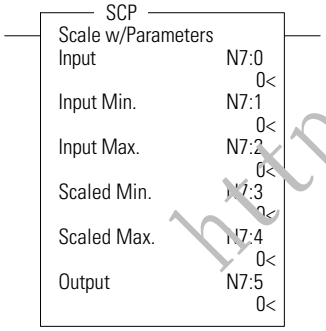
Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address[1] Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•					•														•	•		•		
Rate	•	•		•	•	•					•													•	•	•		•		
Offset	•	•		•	•	•					•														•	•		•		
Destination	•	•		•	•	•					•														•	•		•		

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

IMPORTANT Do not use the High Speed Counter Accumulator (HSC.ACC) for the Destination parameter in the SCL instruction.

SCP - Scale with Parameters



Instruction Type: output

Execution Time for the SCP Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	83.2977 µs	0.3878 µs
	long word	87.0493 µs	0.2910 µs

The SCP instruction produces a scaled output value that has a linear relationship between the input and scaled values. This instruction solves the following equation listed below to determine scaled output:

$$y = [(y_1 - y_0)/(x_1 - x_0)](x - x_0) + y_0$$

Addressing Modes and File Types can be used as shown in the following table:

SCP Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Input (x)	•	•	•	•	•	•	•		•	•	•		•	•	•	•	•	•	•	•	•	•			•	•		•	•	
Input Min. (x_0)	•	•		•	•	•	•		•		•														•	•		•	•	
Input Max. (x_1)	•	•		•	•	•	•		•		•														•	•		•	•	
Scaled Min. (y_0)	•	•		•	•	•	•		•		•													•	•	•		•	•	
Scaled Max. (y_1)	•	•		•	•	•	•		•		•													•	•	•		•	•	
Output (y)	•	•	•	•	•	•	•		•	•	•		•		•	•	•			•					•	•		•	•	

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

IMPORTANT Do not use the High Speed Counter Accumulator (HSC.ACC) for the Scaled Output parameter in the SCP instruction.

Special Considerations when Using Floating Point Parameters

If any of the parameters (except Output) are NAN (not a number), Infinity, or De-normalized; then the result is -NAN.

If $y_1 - y_0$ or $x_1 - x_0$ result in an overflow, then the result is -NAN

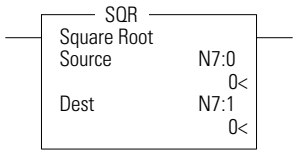
Other Considerations

If $y_1 - y_0 = 0$, the Result becomes the Scaled Start value

If $x_1 - x_0 = 0$ and $x = x_0$, the Result becomes the Scaled Start value

If $x_1 - x_0 = 0$ and x does not equal x_0, The Result becomes a negative overflow (for integer values) or a negative NAN (for floating point values)

SQR - Square Root



Instruction Type: output

Execution Time for the SQR Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	54.8140 µs	0.3561 µs
	long word	45.1450 µs	0.3732 µs

The SQR instruction calculates the square root of the absolute value of the source and places the rounded result in the destination.

The data ranges for the source is -32768...32767 (word) and -2,147,483,648...2,147,483,647 (long word). The Carry Math Status Bit is set if the source is negative. See Updates to Math Status Bits on page 163 for more information.

SQR Instruction Valid Addressing Modes and File Types

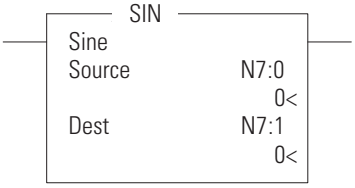
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files								Comms			Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	ST	F	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•		•	•		•													•	•	•		•	•	
Destination	•	•		•	•	•		•	•		•														•	•		•	•	

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

SIN - Sine



Instruction Type: output

Execution Time for the SIN Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	92.8635 µs	0.4210 µs
	long word	95.0760 µs	0.4210 µs

The SIN instruction places the sine of the Source (in radians) in the Destination.

Enter the following parameters when programming this instruction:

- Source is the address to compute the sine.
- Destination is the address to store the sine of the Source.

Address Levels for the operands involved in the SIN can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

SIN Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log[2]	Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute the sine of the Source (in radians) and place the result in Destination. If the Destination is floating point, the result shall always be rounded using the round to even rule.

SIN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Sin(Source) > 0	0,0,0,0	L	Sin(Source)
(1)	DNRM	X	X	X	0,1,0,0	L	0
	W,DW, F	F	X	Sin(Source) < 0	1,0,0,0	L	Sin(Source)
	W,DW, F	W, DW	X	Sin(Source) >= 0.5	0,0,0,0	L	1
	W,DW, F	W, DW	X	-0.5 < Sin(Source) < 0.5	0,1,0,0	L	0

SIN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	W, DW	X	Sin(Source) <= 0.5	1,0,0,0	L	-1
(2)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
(3)	X	W	0	Source is NAN or INF	0,0,1,0	1	32767
(3)	X	DW	0	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

(2) If the Destination is floating point, all overflows(Source is NAN or infinity) shall produce NAN(0x7FFFFFFF).

(3) If the Destination is word or double-word, overflow occurs, a saturated result is stored in Destination. Destination shall be 32767 for Word and 2147483647 for Double Word.

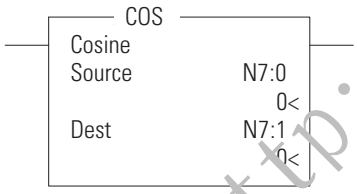
MATH FLAGS EFFECTS

- Carry: shall be reset.
- Overflow: shall be set if the result is infinity, or NAN, otherwise reset.
- Zero: shall be set if the lower 31 bits of float result are all zero (handles negative zero), otherwise reset.
- Sign: shall be set if the most significant bit of Destination is set (bit 31 for float), otherwise reset.

The Math Overflow Trap Bit shall ONLY be set if the Overflow bit is set. Otherwise, it remains in last state.

COS - Cosine

Instruction Type: output



Execution Time for the COS Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	112.7110 µs	0.7686 µs
	long word	19.8070 µs	0.7694 µs

The COS instruction places the sine of the Source (in radians) in the Destination.

Enter the following parameters when programming this instruction:

- Source is the address to compute the cosine.
- Destination is the address to store the cosine of the Source.

Address Levels for the operands involved in the COS can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a

conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

COS Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log[2]	Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•		•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute cosine of the Source (in radians) and place the result in Destination. If the Destination is floating point, the result shall always be rounded using the round to even rule.

COS Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Cos(Source) > 0	0,0,0,0	L	Cos(Source)
(1)	DNRM	X	X	X	0,1,0,0	L	1
	W,DW, F	F	X	Cos(Source) < 0	1,0,0,0	L	Cos(Source)
	W,DW, F	W, DW	X	Cos(Source) >= 0.5	0,0,0,0	L	1
	W,DW, F	W, DW	X	-0.5 < Cos(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	W, DW	X	Cos(Source) <= 0.5	1,0,0,0	L	-1
(2)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
(3)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(3)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

(2) If the Destination is floating point, all overflows(Source is NAN or infinity) shall produce NAN(0x7FFFFFFF).

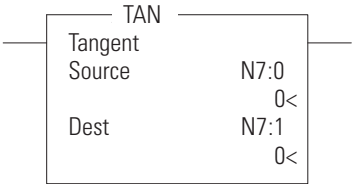
(3) If the Destination is word or double-word, overflow occurs, a saturated result is stored in Destination. Destination shall be 32767 for Word and 2147483647 for Double Word.

MATH FLAGS EFFECTS

- Carry: shall be reset.
- Overflow: shall be set if the result is infinity, or NAN, otherwise reset.
- Zero: shall be set if the lower 31 bits of float result are all zero (handles negative zero), otherwise reset.
- Sign: shall be set if the most significant bit of Destination is set (bit 31 for float), otherwise reset.

The Math Overflow Trap Bit shall ONLY be set if the Overflow bit is set. Otherwise, it remains in last state.

TAN - Tangent



Instruction Type: output

Execution Time for the TAN Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	122.6760 µs	0.3915 µs
	long word	126.9135 µs	0.4234 µs

The TAN instruction places the sine of the Source (in radians) in the Destination.

Enter the following parameters when programming this instruction:

- Source is the address to compute the tangent.
- Destination is the address to store the tangent of the Source.

Address Levels for the operands involved in the TAN can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

TAN Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]										Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute tangent of the Source (in radians) and place the result in Destination. If the Destination is floating point, the result shall always be rounded using the round to even rule.

TAN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Tan(Source) > 0	0,0,0,0	L	Tan(Source)
(1)	DNRM	X	X	X	0,1,0,0	L	0
	W,DW, F	F	X	Tan(Source) < 0	1,0,0,0	L	Tan(Source)
	W,DW, F	W, DW	X	Tan(Source)>= 0.5 && no overflow	0,0,0,0	L	Rnd(Tan(Source))
	W,DW, F	W, DW	X	-0.5 < Tan(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	W, DW	X	Tan(Source) <= 0.5 && no overflow	1,0,0,0	L	Rnd(Tan(Source))
(2)	W,DW, F	W	X	Tan(Source) >= 32767.5	0,0,1,0	1	32767
(2)	W,DW, F	W	X	Tan(Source) <= -32768.5	1,0,1,0	1	-32768
(2)	W,DW, F	DW	X	Tan(Source) >= 214748.647.5	0,0,1,0	1	2147483647
(2)	W,DW, F	DW	X	Tan(Source) <= -214748.648.5	1,0,1,0	1	-2147483648
(3)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
(4)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(4)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

(2) If the Destination is word or double-word and an overflow occurs(except source is NAN or INF):
If the result is positive, Destination is 32767 for Word and 2147483647 for Double Word. If the result is negative, Destination is -32768 for Word and -2147483648 for Double Word.

(3) If the Destination is floating point, all overflows(Source is NAN or infinity) shall produce NAN(0x7FFFFFFF).

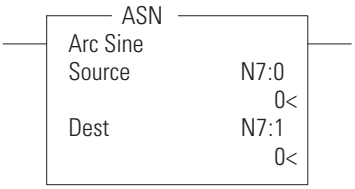
(4) If Destination is an integer, the Source is NAN or infinity, a saturated result (32767 for Word and 2147483647 for Double Word) shall be stored

MATH FLAGS EFFECTS

- Carry: shall be reset.
- Overflow: shall be set if the result is infinity, or NAN, otherwise reset.
- Zero: shall be set if the lower 31 bits of float result are all zero (handles negative zero), otherwise reset.
- Sign: shall be set if the most significant bit of Destination is set (bit 31 for float), otherwise reset.

The Math Overflow Trap Bit shall ONLY be set if the Overflow bit is set. Otherwise, it remains in last state.

ASN - Arc Sine



Instruction Type: output

Execution Time for the ASN Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	42.4610 µs	0.3870 µs
	long word	43.1010 µs	0.3790 µs

Use the ASN instruction to take the arc sine of a number and store the result (in radians) in the destination. The source must be greater than or equal to -1 and less than or equal to 1. The resulting value in the destination is always greater than or equal to $-\pi/2$ and less than or equal to $\pi/2$, where $\pi = 3.141592$.

Enter the following parameters when programming this instruction:

- Source is the address to compute the arc sine.
- Destination is the address to store the arc sine of the Source.

Address Levels for the operands involved in the ASN can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

ASN Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]										Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute arc-sine of the Source and place the result in Destination. Valid input range is from -1.0...1.0 and output range is from $-\pi/2...\pi/2$.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

ASN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	0 < Source <= 1.0	0,0,0,0	L	Asin(Source)
(1)	DNRM	X	X	X	0,1,0,0	L	0
	W,DW, F	F	X	-1.0 < Source < 0	1,0,0,0	L	Asin(Source)
	W,DW, F	W, DW	X	Source <=1 && Asin(Source) >= 0.5	0,0,0,0	L	Rnd(Asin(Source))
	W,DW, F	W, DW	X	-0.5 < Asin(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	W, DW	X	Source >= -1 && Asin(Source) <= -0.5	1,0,0,0	L	Rnd(Asin(Source))
(2)	X	F	X	Source < -1 or Source > 1	0,0,1,0	1	0x7FFFFFFF
(2)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
	X	W	X	Source < -1 or Source > 1	0,0,1,0	1	32767

ASN Instruction Operation

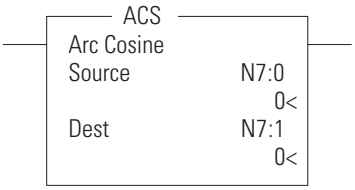
	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	X	DW	X	Source < -1 or Source > 1	0,0,1,0	1	2147483647
(3)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(3)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

(2) If the Destination is floating point, all overflows(Source is NAN or infinity) shall produce NAN(0x7FFFFFFF).

(3) If the Destination is word or double-word, and an overflow occurs, destination shall be 32767 for Word and 2147483647 for Double Word.

ACS - Arc Cosine



Instruction Type: output

Execution Time for the ACS Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	18.0150 µs	0.3750 µs
	long word	18.3070 µs	0.4150 µs

Use the ACS instruction to take the arc cosine of a number (source in radians) and store the result (in radians) in the destination. The source must be greater than or equal to -1 and less than or equal to 1. The resulting value in the destination is always greater than or equal to 0 and less than or equal to π, where π = 3.141592.

Enter the following parameters when programming this instruction:

- Source is the address to compute the arc cosine.
- Destination is the address to store the arc cosine of the Source.

Address Levels for the operands involved in the ACS can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

ACS Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]											Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute arc cosine of the Source and place the result in Destination. Valid input range is from -1.0...1.0 and output range is from 0...π.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

ACS Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	-1.0 <= Source < 1.0	0,0,0,0	L	Acos(Source)
	W,DW, F	F	X	Source= 1.0	0,1,0,0	L	0
(1)	DNRM	F	X	X	0,0,0,0	L	π/2
	W,DW, F	W, DW	X	Acos(Source) >= 0.5	0,0,0,0	L	Rnd(Acos(Source))
	W,DW, F	W, DW	X	Acos(Source) < 0.5	0,1,0,0	L	0
(2)	X	F	X	Source < -1 or Source > 1	0,0,1,0	1	0x7FFFFFFF
(2)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
	X	W	X	Source < -1 or Source > 1	0,0,1,0	1	32767

ACS Instruction Operation

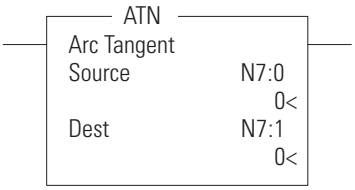
	Inputs			Conditions	Output		
	Source Type	**Destination Type**	**S:2/14**		**S:0/3-0 S, Z, V, C**	**S:5/0**	**Destination**
	X	DW	X	Source < -1 or Source > 1	0,0,1,0	1	2147483647
(3)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(3)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce π/2.

(2) If the Destination is floating point, all overflows shall produce NAN(0x7FFFFFFF).

(3) If the Destination is word or double-word, and an overflow occurs, destination shall be 32767 for Word and 2147483647 for Double Word.

ATN - Arc Tangent



Instruction Type: output

Execution Time for the ATN Instruction

Controller	Data Size	When Rung Is:	
		True	**False**
MicroLogix 1400	word	146.7510 µs	0.3740 µs
	long word	146.4885 µs	0.4088 µs

Use the ATN instruction to take the arc tangent of a number (source) and store the result (in radians) in the destination. The resulting value in the destination is always greater than or equal to -π/2 and less than or equal to π/2, where π = 3.141592.

Enter the following parameters when programming this instruction:

- Source is the address to compute the arc tangent.
- Destination is the address to store the arc tangent of the Source.

Address Levels for the operands involved in the ATN can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

ATN Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]										Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute arc tangent of the Source and place the result in Destination. Valid input range is from $-\infty...+\infty$ and produced output range is from $-\pi/2...\pi/2$.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

ATN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Source >= 0	0,0,0,0	L	Atan(Source)
(1)	DNRM	F	X	X	0,1,0,0	L	0
	W,DW, F	F	X	Source < 0	1,0,0,0	L	Atan(Source)
	W,DW, F	W, DW	X	Atan(Source) >= 0.5	0,0,0,0	L	Rnd(Atan(Source))
	W,DW, F	W, DW	X	-0.5 < Atan(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	W, DW	X	Atan(Source) < -0.5	1,0,0,0	L	Rnd(Atan(Source))
	X	F	X	Source = +INF	0,0,0,0	1	$\pi/2$
	X	F	X	Source = -INF	1,0,0,0	1	$-\pi/2$

ATN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	**Destination Type**	**S:2/14**		**S:0/3-0 S, Z, V, C**	**S:5/0**	**Destination**
(2)	X	F	X	Source is NAN	0,0,1,0	1	0x7FFFFFFF
(3)	X	W	X	Source is NAN	0,0,1,0	1	32767
(3)	X	DW	X	Source is NAN	0,0,1,0	1	2147483647

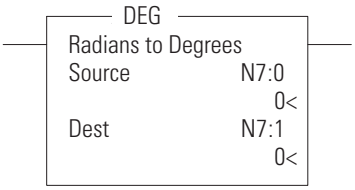
(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

(2) If the Destination is floating point, all overflows shall produce NAN(0x7FFFFFFF).

(3) If the Destination is word or double-word, and an overflow occurs, destination shall be 32767 for Word and 2147483647 for Double Word.

DEG - Radians to Degrees

Instruction Type: output



Execution Time for the DEG Instruction

Controller	Data Size	When Rung Is:	
		True	**False**
MicroLogix 1400	word	27.7310 µs	0.4106 µs
	long word	31.2470 µs	0.4098 µs

The DEG instruction converts the Source(in radians) to degrees and store the result in the Destination.

The following formula applies:

Source * 180/Π
where Π = 3.141592

Enter the following parameters when programming this instruction:

- Source is the address to compute the degrees.
- Destination is the address to store the degrees of the Source.

Address Levels for the operands involved in the DEG can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

DEG Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log[2]	Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall convert the Source(in radians) in degrees and place the result in Destination.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

DEG Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Source >= 0	0,0,0,0	L	Deg(Source)
(1)	DNRM	F	X	X	0,1,0,0	L	0
	W,DW, F	F	X	Source < 0	1,0,0,0	L	Deg(Source)
	W,DW, F	W	X	-0.5 <= Deg(Source) < 32767.5	0,0,0,0	L	Rnd(Deg(Source))
	W,DW, F	W	X	-0.5 < Deg(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	W	X	-32768.5 < Deg(Source) <= -0.5	1,0,0,0	L	Rnd(Deg(Source))
	W,DW, F	DW	X	0.5 <= Deg(Source) < 2147483647.5	0,0,0,0	L	Rnd(Deg(Source))
	W,DW, F	DW	X	-0.5 < Deg(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	DW	X	-2147483648.5 < Deg(Source) <= -0.5	1,0,0,0	L	Rnd(Deg(Source))
(2)	W,DW, F	W	X	Deg(Source) >= 32767.5	0,0,1,0	1	32767
(2)	W,DW, F	W	X	Deg(Source) <= 32767.5	1,0,1,0	1	-32768

DEG Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
(2)	W,DW, F	DW	X	Deg(Source) >= 2147483647.5	0,0,1,0	1	2147483647
(2)	W,DW, F	DW	X	Deg(Source) <= -2147483648.5	1,0,1,0	1	-2147483648
(3)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
(4)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(4)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

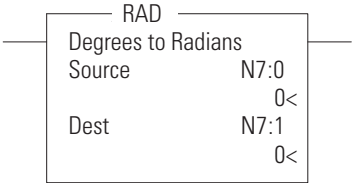
(2) If the Destination is word or double-word and an overflow occurs(except source is NAN or INF):
If the result is positive, Destination is 32767 for Word and 2147483647 for Double Word. If the result is negative, Destination is -32768 for Word and -2147483648 for Double Word.

(3) If the Destination is floating point, all overflows (Source is NAN or infinity) shall produce NAN (0x7FFFFFFF).

(4) If Destination is an integer and the Source is NAN or infinity, a saturated result (32767 for Word and 2147483647 for Double Word) shall be stored.

RAD - Degrees to Radians

Instruction Type: output



Execution Time for the RAD Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	23.0610 µs	0.4070 µs
	long word	26.211 µs	0.3790 µs

The RAD instruction converts the Source (in degrees) to radians and stores the result in the Destination.

The following formula applies:

Source * Π/180
where Π = 3.141592

Enter the following parameters when programming this instruction:

- Source is the address to compute the radians.
- Destination is the address to store the radians of the Source.

Address Levels for the operands involved in the RAD can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

RAD Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log[2]	Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•	•	•	•		•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall convert the Source (in degrees) in radians and place the result in Destination.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

RAD Instruction Operation

	Inputs		Conditions		Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Source >= 0	0,0,0,0	L	Rad(Source)
(1)	DNRM	F	X	X	0,1,0,0	L	0
	W,DW, F	F	X	Source < 0	1,0,0,0	L	Rad(Source)
	W,DW, F	W	X	-0.5 <= Rad(Source) < 32767.5	0,0,0,0	L	Rnd(Rad(Source))
	W,DW, F	W	X	-0.5 < Rad(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	W	X	-32768.5 < Rad(Source) <= -0.5	1,0,0,0	L	Rnd(Rad(Source))
	W,DW, F	DW	X	0.5 <= Rad(Source) < 2147483647.5	0,0,0,0	L	Rnd(Rad(Source))
	W,DW, F	DW	X	-0.5 < Rad(Source) < 0.5	0,1,0,0	L	0
	W,DW, F	DW	X	-2147483648.5 < Rad(Source) <= -0.5	1,0,0,0	L	Rnd(Rad(Source))
(2)	W,DW, F	W	X	Rad(Source) >= 32767.5	0,0,1,0	1	32767
(2)	W,DW, F	W	X	Rad(Source) <= 32767.5	1,0,1,0	1	-32768
(2)	W,DW, F	DW	X	Rad(Source) >= 2147483647.5	0,0,1,0	1	2147483647
(2)	W,DW, F	DW	X	Rad(Source) <= -2147483648.5	1,0,1,0	1	-2147483648

RAD Instruction Operation

	Inputs			Conditions	Output		
	Source Type	**Destination Type**	**S:2/14**		**S:0/3-0 S, Z, V, C**	**S:5/0**	**Destination**
(3)	X	F	X	Source is NAN or INF	0,0,1,0	1	0x7FFFFFFF
(4)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(4)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

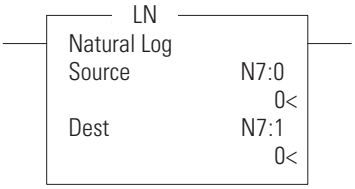
(1) All denormalized inputs shall be treated as plus zero. Any underflow result shall produce plus zero.

(2) If the Destination is word or double-word and an overflow occurs(except source is NAN or INF):
If the result is positive, Destination is 32767 for Word and 2147483647 for Double Word. If the result is negative, Destination is -32768 for Word and -2147483648 for Double Word.

(3) If the Destination is floating point, all overflows (Source is NAN or infinity) shall produce NAN (0x7FFFFFFF).

(4) If Destination is an integer and the Source is NAN or infinity, a saturated result (32767 for Word and 2147483647 for Double Word) shall be stored.

LN - Natural Log



Instruction Type: output

Execution Time for the LN Instruction

Controller	Data Size	When Rung Is:	
		True	**False**
MicroLogix 1400	word	127.3260 µs	0.4094 µs
	long word	130.3635 µs	0.4094 µs

Use the LN instruction to take the natural log of the value in the source and store the result in the destination. The source must be greater than zero.

Enter the following parameters when programming this instruction:

- Source is the address to compute the natural log.
- Destination is the address to store the natural log of the Source.

Address Levels for the operands involved in the LN can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

LN Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]												Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•					•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

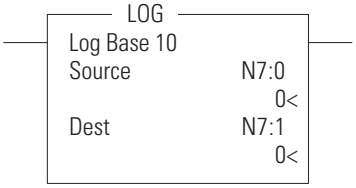
IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute the natural logarithm of the Source and place the result in Destination.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

LOG - Base 10 Logarithm



LN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Source > 1	0,0,0,0	L	Ln(Source)
	W,DW, F	F	X	Source = 1	1,0,0,0	L	0
	W,DW, F	F	X	0 < Source < 1	1,0,0,0	L	Ln(Source)
	W,DW, F	W, DW	X	Source >= Sqrt(e)	0,0,0,0	L	Rnd(Ln(Source))
	W,DW, F	W, DW	X	1/Sqrt(e) < Source < Sqrt(e)	1,0,0,0	L	0
	W,DW, F	W, DW	X	0 < Source <= 1/Sqrt(e)	1,0,0,0	L	Rnd(Ln(Source))

LN Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
(1)	+DNRM	F	X	X	1,0,1,0	1	-INF
(1)(2)	+DNRM	W	X	X	1,0,1,0	1	-32768
(1) (2)	+DNRM	DW	X	X	1,0,1,0	1	-2147483648
(1)(3)	W,DW, F	F	X	Source <= 0 (including -DNRM)	0,0,1,0	1	0x7FFFFFFF
(2)	W,DW, F	F	X	Source = +INF	0,0,1,0	1	+INF
(3)	X	F	X	Source is NAN or -INF	0,0,1,0	1	0x7FFFFFFF
(4)	X	W	X	Source is NAN or INF	0,0,1,0	1	32767
(4)	X	DW	X	Source is NAN or INF	0,0,1,0	1	2147483647

(1) All denormalized inputs shall be treated as plus zero.

(2) If the Destination is word or double-word and an overflow occurs (except source is NAN or INF), destination is -32768 for Word and -2147483648 for Double Word.

(3) If the Destination is floating point, all overflows (except +INF) shall produce NAN (0x7FFFFFFF).

(4) If Destination is an integer and the Source is NAN or infinity (+INF or -INF), a saturated result (32767 for Word and 2147483647 for Double Word) shall be stored.

Instruction Type: output

Execution Time for the LOG Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	112.7110 µs	0.7686 µs
	long word	19.8070 µs	0.7694 µs

Use the LOG instruction to take the log base 10 of the value in the source and store the result in the destination. The source must be greater than zero.

Enter the following parameters when programming this instruction:

- Source is the address to compute the base 10 logarithm.
- Destination is the address to store the base 10 logarithm of the Source.

Address Levels for the operands involved in the LOG can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

LN Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]												Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source	•	•		•	•	•	•		•			•												•	•	•					•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute the natural logarithm of the Source and place the result in Destination.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

LOG Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
	W,DW, F	F	X	Source > 1	0,0,0,0	L	Log(Source)
	W,DW, F	F	X	Source = 1	0,1,0,0	L	0
	W,DW, F	F	X	0 < Source < 1	1,0,0,0	L	Log(Source)
	W,DW, F	W, DW	X	Source >= Sqrt(10)	0,0,0,0	L	Rnd(Log(Source))
	W,DW, F	W, DW	X	1/Sqrt(10) < Source < Sqrt(10)	0,1,0,0	L	0
	W,DW, F	W, DW	X	0 < Source <= 1/Sqrt(10)	1,0,0,0	L	Rnd(Log(Source))
(1)	+DNRM	F	X	X	1,0,1,0	1	-INF
(1)(2)	+DNRM	W	0	X	1,0,1,0	1	-32768
(1) (2)	+DNRM	DW	0	X	1,0,1,0	1	-2147483648
(1)(3)	W,DW, F	F	X	Source <= 0 (including -DNRM)	0,0,1,0	1	0x7FFFFFFF

LOG Instruction Operation

	Inputs			Conditions	Output		
	Source Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
(2)	W,DW, F	F	X	Source = +INF	0,0,1,0	1	+INF
(3)	X	F	X	Source is NAN or -INF	0,0,1,0	1	0x7FFFFFFF
(4)	X	W	0	Source is NAN or INF	0,0,1,0	1	32767
(4)	X	DW	0	Source is NAN or INF	0,0,1,0	1	2147483647

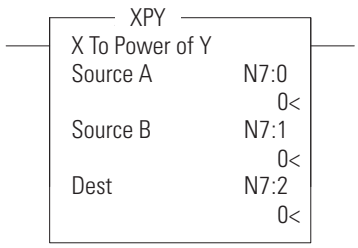
(1) All denormalized inputs shall be treated as plus zero.

(2) If the Destination is word or double-word and an overflow occurs (except source is NAN or INF), destination is -32768 for Word and -2147483648 for Double Word.

(3) If the Destination is floating point, all overflows (except +INF) shall produce NAN (0x7FFFFFFF).

(4) If Destination is an integer and the Source is NAN or infinity (+INF or -INF), a saturated result (32767 for Word and 2147483647 for Double Word) shall be stored.

XPY - X Power Y



Instruction Type: output

Execution Time for the XPY Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	66.2050 µs	0.3920 µs
	long word	69.0550 µs	0.3548 µs

Use the XPY instruction to raise a value (source A) to a power (source B) and store the result in the destination. If the value in source A is negative, the exponent (source B) should be a whole number. If it is not a whole number, the overflow bit is set and the absolute value of the base is used in the calculation

Enter the following parameters when programming this instruction:

- Source A is a value of base to power
- Source B is a value or address to compute exponent
- Destination is the address to store the result of computation

Address Levels for the operands involved in the XPY can be ALL word, ALL double word, ALL float, or a combination. These operands shall undergo a conversion to float. The calculation of the source (in float) is then performed, and the result is then cast to the data type of Destination.

XPY Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]											Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source A/B	•	•		•	•	•	•		•			•												•	•	•					•	•		•
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute the Source A to power Source B and place the result in Destination.

If the Destination is floating point, the result shall always be rounded using the round to even rule.

XPY Instruction Operation

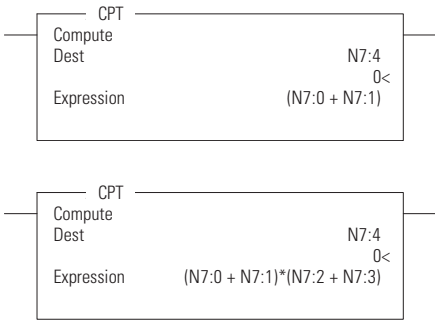
Inputs				Conditions	Output		
Source A Type	Source B Type	Destination Type	S:2/14		S:0/3-0 S, Z, V, C	S:5/0	Destination
W,DW, F	W,DW, F	F	X	Source A > 0 && (A)**(B) is +NRM	0,0,0,0	L	(A)**(B)
W,DW, F	W,DW, F	F	X	Source A > 0 && (A)**(B) is +DNRM	0,1,0,0	L	0
W,DW, F	W,DW, F	W	X	Source A > 0 && 0.5 <= (A)**(B) < 32767.5	0,0,0,0	L	Rnd((A)**(B))
W,DW, F	W,DW, F	DW	X	Source A > 0 && 0.5 <= (A)**(B) < 2147483647.5	0,0,0,0	L	Rnd((A)**(B))
W,DW, F	W,DW, F	W, DW	X	Source A > 0 && (A)**(B) < 0.5	0,1,0,0	L	0
W,DW, F	DNRM	W,DW, F	X	(A)**(B) is not DNRM	0,0,0,0	L	1

XPY Instruction Operation

Inputs				Conditions	Output		
Source A Type	**Source B Type**	**Destination Type**	**S:2/14**		**S:0/3-0 S, Z, V, C**	**S:5/0**	**Destination**
W,DW, F	W,DW, F	F	X	Source A < 0 && Source B = 1	1,0,0,0	L	Source A
DNRM	W,DW, F	W,DW, F	X	Source B > 0	0,1,0,0	L	0
W,DW, F	W,DW, F	F	X	Source A < 0 && Source B is even integer	0,0,0,0	L	(A)**(B)
W,DW, F	W,DW, F	F	X	Source A < 0 && Source B is odd integer	1,0,0,0	L	(A)**(B)
W,DW, F	W,DW, F	F	X	Source A > 0 && (A)**(B) floating point overflow	0,0,1,0	1	+INF
W,DW, F	W,DW, F	F	X	Source A < 0 && Source B is odd integer && (A)**(B) floating point overflow	1,0,1,0	1	-INF
W,DW, F	W,DW, F	W	X	Source A > 0 && (A)**(B) > 32767.5	0,0,1,0	1	32767
DNRM	DNRM	F	X	X	0,0,1,0	1	0x7FFFFFFF
DNRM	DNRM	W	X	X	0,0,1,0	1	32767
DNRM	DNRM	DW	X	X	0,0,1,0	1	2147483647
DNRM	W,DW, F	F	X	Source A < 0	0,0,1,0	1	0x7FFFFFFF
W,DW, F	W,DW, F	F	X	Source A < 0 && Source B is not integer value	1,0,1,0	1	Abs(Source A) ** (Source B)
W,DW, F	W,DW, F	F	X	Source A = NAN or Source B = NAN	0,0,1,0	1	0x7FFFFFFF
W,DW, F	W,DW, F	F	X	(A = INF && B >=0) or ((A<=-1 or A>1) and (B=+INF)) or (-1<=A<1 and B=-INF)	0,0,1,0	1	0x7FFFFFFF
W,DW, F	W,DW, F	F	X	(A = INF and (B < 0 or B = -INF)) or (abs(A)>1 and B = -INF) or (abs(A)<1 and B=+INF)	0,1,1,0	1	0

IMPORTANT The XPY instruction processes at the floating-point level, so the result causes the truncation error when it is used with the long data types.

CPT - Compute



Instruction Type: output

Execution Time for the CPT Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	4.8535 µs	0.6610 µs

The CPT instruction performs copy, arithmetic, logical, and conversion operations. You define the operation in the Expression and the result is written in the Destination. The CPT uses functions to operate on one or more values in the Expression to perform operations such as:

- converting from one number format to another.
- manipulating numbers.
- performing trigonometric functions.

TIP The execution time of a CPT instruction is longer than a single arithmetic operation and uses more instruction words.

Enter the following parameters when programming this instruction:

- Expression is zero or more lines, with up to 28 characters per line, up to 255 characters.
- Destination is a word address or the address of a floating-point data element.

CPT Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files(1)											DLS - Data Log(2)	Address Mode(3)						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O		Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Destination	•	•		•	•	•	•		•			•												•	•						•	•		•

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction shall compute the expression and place the result in Destination.

Note that it takes on the attributes of the instruction which it is computing. The following instructions can be included in the expression section of CPT instruction.

CPT Instruction Operation

Sub Instructions	Symbol
Addition	+ (ADD)
Subtraction	- (SUB)
Multiplication	* (MUL)
Division	/ (DIV)
Square Root	SQR
Negate	- (NEG)
Logical Not	NOT
Logical Exclusive Or	XOR
Logical Inclusive Or	OR
Logical And	AND
Covert to BCD	TOD
From BCD to Binary	FRD
Natural Log	LN
Base 10 Log	LOG
Tangent	TAN
Sine	SIN
Cosine	COS
Arc Tangent	ATN
Arc Sine	ASN
Arc Cosine	ACS
Absolute Value	ABS
Radians to Degrees	DEG
Degrees to Radians	RAD
X to the Power Y	* *(XPY)

MATH FLAGS EFFECTS

- Carry: shall be set or reset based on the result of the last instruction in the Expression.
- Overflow: shall be set any time an overflow occurs during the evaluation of the Expression. This bit never cleared in the CPT instruction.

- Zero: shall be set if the lower 31 bits of float result of Destination are all zero (handles negative zero), otherwise reset.
- Sign: shall be set if the most significant bit of Destination is set (bit 31 for float), otherwise reset.

The Math Overflow Trap Bit shall ONLY be set if the Overflow bit is set. Otherwise, it remains in last state.

Application Specific Instructions

This chapter contains general information about the application specific instructions and explains how they function in your application program. Each of the instructions includes information on:

- what the instruction symbol looks like.
- how to use the instruction.

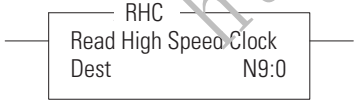
These instructions simplify your ladder program by allowing you to use a single instruction or pair of instructions to perform common complex operations.

In this chapter you will find a general overview preceding groups of instructions. Before you learn about the instructions in each of these groups, we suggest that you read the overview that precedes each section. This chapter contains the following overviews:

Application Specific Instructions

Instruction	Used To:	Page
RHC	Provide a high performance time-stamp for performance diagnostics and performing calculations such as velocity.	199
RPC	Copy the program checksum from processor memory or from the memory module into the data table.	201
TDF	Calculate the number of 9.92063492 µs "ticks" between any two time-stamps captured using the RHC instruction.	202

RHC - Read High Speed Clock



Instruction Type: Output

Execution Time for the RHC Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	2.5910 µs	0.2150 µs
	long word	3.1210 µs	0.1802 µs

The RHC instruction provides a high performance timestamp for diagnostics and calculation such as velocity. The controller maintains a 9.92063492 µs long integer free running clock/counter. This 32 bit value increments every 9.92063492 µs after power-up.

Enter the following parameters when programming this instruction:

- Destination is the address to store the current value of the 9.92063492 μs free running clock. It can be an integer address, long integer address or Float address.

RHC Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log[2]	Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Destination						•	•		•																								

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

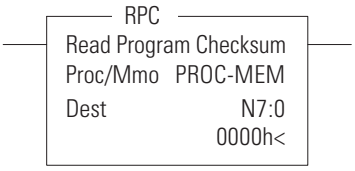
This instruction executes on a true rung. When the rung is true, this instruction moves the current value of the 9.92063492 μs free running clock into the Destination.

If the Destination is an integer address type, only 16 bits are moved into the address. If the Destination is floating point address, the long integer value is converted into a float and is moved the relative address.

After the free running clock reaches 0xFFFFFFFF (42608.8025 seconds) value, it wraps around to 0 and continues incrementing. The RESET signal or Power Cycle sets the free running clock to 0.

RPC - Read Program Checksum

Instruction Type: Output



Execution Time for the RPC Instruction

Controller	When Instructions Is:	
	True	False
MicroLogix 1400	4.2844 µs	0.2028 µs

The RPC instruction reads Program copies the checksum of the processor program from either the processor's RAM memory or from the installed memory module into the designated destination integer file location.

Enter the following parameters when programming this instruction:

- Proc/Mmod is an immediate value with a range from 0...1. Specify where the Program checksum is read, and what type of operation to be performed (Proc-Mem / Mem Mod).
- Destination is the address to store the result of Program Checksum from processor memory.

RPC Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]														Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log[2]	Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Source A						•																		•										
Destination						•																		•										

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

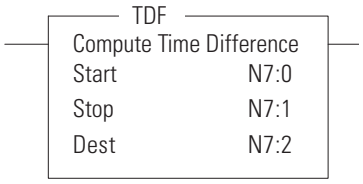
Instruction Operation

When the rung is true, this instruction shall read Program Checksum from processor memory or from memory module and places the result into Destination.

Address Levels for the operands involved in the RPC should be Word. The result is stored to the data type of Destination. The RPC instruction destination address supports Direct Addressing. It does not support Indirect addressing, Indexed addressing, or Indirect Indexed Addressing.

When the Source A is 1(read from Memory Module), the memory module should be installed in Non-Executing mode. If the Memory Module is installed during RUN mode, the Destination value will be 0.

TDF - Compute Time Difference



Instruction Type: Output

Execution Time for the RPC Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	5.9770 µs	0.2219 µs
	long word	7.2150 µs	0.2035 µs

The Compute Time Difference Instruction (TDF) is used to calculate the number of 9.92063492 µs ticks between any two time-stamps captured using the RHC instruction. This lets your program determine the time difference between any two events using a 9.92063492 µs timebase and places the result into the Destination.

(Stop) - (Start) -> Destination

Enter the following parameters when programming this instruction:

- Start is the address of the earliest value previously captured using the RHC instruction.
- Stop is the address of a later value captured using the RHC instruction.
- Destination is the address to store the result of the time difference calculation.

All of these parameters should be of the same data type (Nx:x, Lx:x or Fx:x). The data range for the Start and Stop timestamp is from -32768...32767 (Word) or -2,147,483,648...2,147,483,647 (Long Word), or any IEEE-754 32-bit value.

TDF Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]											DLS - Data Log[2]	Address Mode[3]						Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O		Direct	Indirect	Immediate	Int16	Immediate Int32	Immediate Float	Bit	Word	Double Word	Element	Float
Start						•	•		•															•										
Stop						•	•		•															•										
Destination						•	•		•															•										

(1) PTOX and PWMX files are only for use with MicroLogix 1400 BXB or BXBA unit.

(2) The Data Log Status file can only be used for the following math instructions: ADD, SUB, MUL, DIV, NEG, and SCP.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Instruction Operation

When the rung is true, this instruction calculates the number of 9.92063492 µs "ticks" that have elapsed from the Start value to the Stop value and places the result into the Destination.

Any TDF instruction with a floating point address computes the time difference between 2 timestamps captured within 10.4025 seconds of each other (1048575 9.92063492 µsec ticks). It calculates an invalid result if more than 10.4025 seconds have elapsed between the start and stop timestamps.

Any TDF instruction with an integer address computes the positive time difference between the START and END timestamps. It calculates an invalid result if more than 327.67 ms have elapsed between the start and stop timestamps.

Any TDF instruction with a double word address computes the time difference between 2 timestamps captured within 42608.8025 seconds of each other (4294967295 9.92063492 µsec ticks). It calculates an invalid result if more than 42608.8025 seconds have elapsed between the start and stop timestamps.

This instruction shall do nothing on a false rung.

Notes:

Conversion Instructions

The conversion instructions multiplex and de-multiplex data and perform conversions between binary and decimal values.

Instruction	Used To:	Page
DCD - Decode 4 to 1-of-16	Decodes a 4-bit value (0…15), turning on the corresponding bit in the 16-bit destination.	206
ENC - Encode 1-of-16 to 4	Encodes a 16-bit source to a 4-bit value. Searches the source from the lowest to the highest bit and looks for the first set bit. The corresponding bit position is written to the destination as an integer.	206
FRD - Convert From Binary Coded Decimal	Converts the BCD source value to an integer and stores it in the destination.	208
TOD - Convert to Binary Coded Decimal	Converts the integer source value to BCD format and stores it in the destination.	211
GCD - Gray Code	Converts Gray code data (Source) to an integer value and stores it in the destination.	213

Using Decode and Encode Instructions

Addressing Modes and File Types can be used as shown in the following table:

Conversion Instructions Valid Addressing Modes and File Types

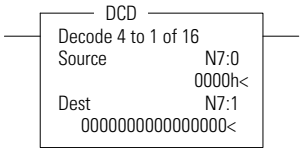
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files											Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•					•														•	•		•		
Destination	•	•		•	•	•					•														•	•		•		

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

DCD - Decode 4 to 1-of-16



Instruction Type: output

Execution Time for the DCD Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	4.6300 µs	0.2720 µs

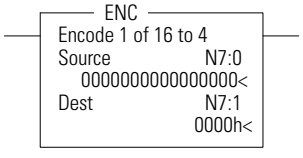
The DCD instruction uses the lower four bits of the source word to set one bit of the destination word. All other bits in the destination word are cleared. The DCD instruction converts the values as shown in the table below:

Decode 4 to 1-of-16

Source Bits					Destination Bits															
15 to 04	03	02	01	00	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
x	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
x	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
x	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
x	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
x	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
x	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
x	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
x	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
x	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
x	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
x	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
x	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
x	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
x	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x = not used

ENC - Encode 1-of-16 to 4



Instruction Type: output

Execution Time for the ENC Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	5.7230 µs	0.3660 µs

The ENC instruction searches the source from the lowest to the highest bit, looking for the first bit set. The corresponding bit position is written to the destination as an integer. The ENC instruction converts the values as shown in the table below:

Encode 1-of-16 to 4

Source Bits																Destination Bits				
15	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**	**15 to 04**	**03**	**02**	**01**	**00**
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	1
x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	1	0
x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	1
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	0	0
x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	1	0	1
x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	1	1	0
x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	1	1	1
x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	1	0	0	0
x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1
x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
x	x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

x = determines the state of the flag

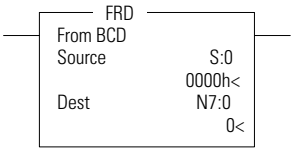
TIP If source is zero, the destination is zero and the math status is zero, the flag is set to 1.

Updates to Math Status Bits

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always resets
S:0/1	Overflow	sets if more than one bit in the source is set; otherwise resets. The math overflow bit (S:5/0) is not set.
S:0/2	Zero Bit	sets if result is zero, otherwise resets
S:0/3	Sign Bit	always resets

FRD - Convert from Binary Coded Decimal (BCD)



Instruction Type: output

Execution Time for the FRD Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	5.4790 µs	0.5151 µs

The FRD instruction is used to convert the Binary Coded Decimal (BCD) source value to an integer and place the result in the destination.

Addressing Modes and File Types can be used as shown in the following table:

FRD Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files										Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•	•	•	•	•					•														•	•		•		(2)
Destination	•	•		•	•	•					•														•	•		•		

(1) See Important note about indirect addressing.

(2) See FRD Instruction Source Operand on page 208.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

FRD Instruction Source Operand

The source can be either a word address or the math register. The maximum BCD source values permissible are:

- 9999 if the source is a word address (allowing only a 4-digit BCD value)
- 32768 if the source is the math register (allowing a 5-digit BCD value with the lower 4 digits stored in S:13 and the high order digit in S:14).

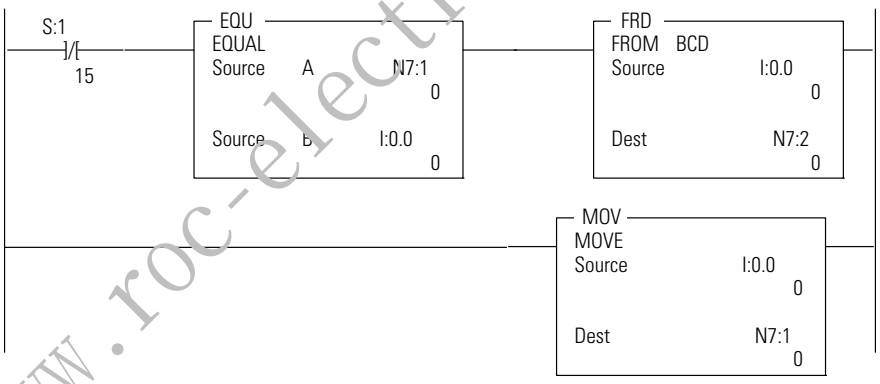
If the source is the math register, it must be directly addressed as S:13. S:13 is the only status file element that can be used.

Updates to Math Status Bits

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always resets
S:0/1	Overflow	sets if non-BCD value is contained at the source or the value to be converted is greater than 32,767; otherwise resets. On overflow, the minor error flag is also set.
S:0/2	Zero Bit	sets if result is zero, otherwise resets
S:0/3	Sign Bit	always resets

TIP Always provide ladder logic filtering of all BCD input devices prior to performing the FRD instruction. The slightest difference in point-to-point input filter delay can cause the FRD instruction to overflow due to the conversion of a non-BCD digit.

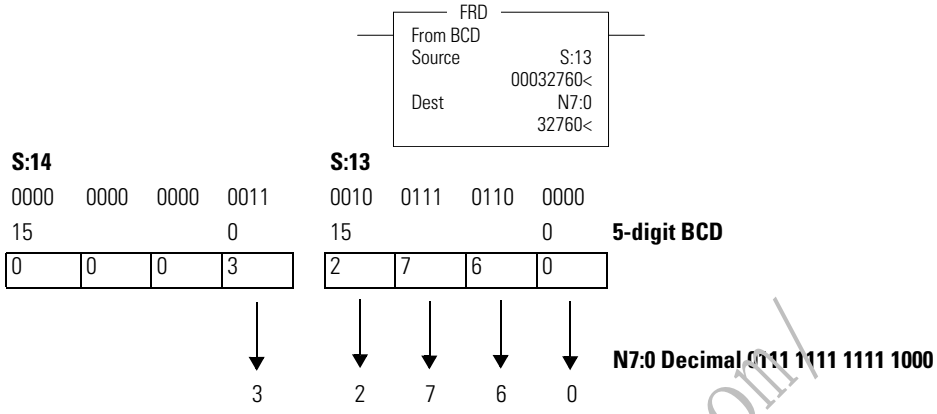


The two rungs shown cause the controller to verify that the value I:0 remains the same for two consecutive scans before it executes the FRD. This prevents the FRD from converting a non-BCD value during an input value change.

TIP To convert numbers larger than 9999 BCD, the source must be the Math Register (S:13). You must reset the Minor Error Bit (S:5.0) to prevent an error.

Example

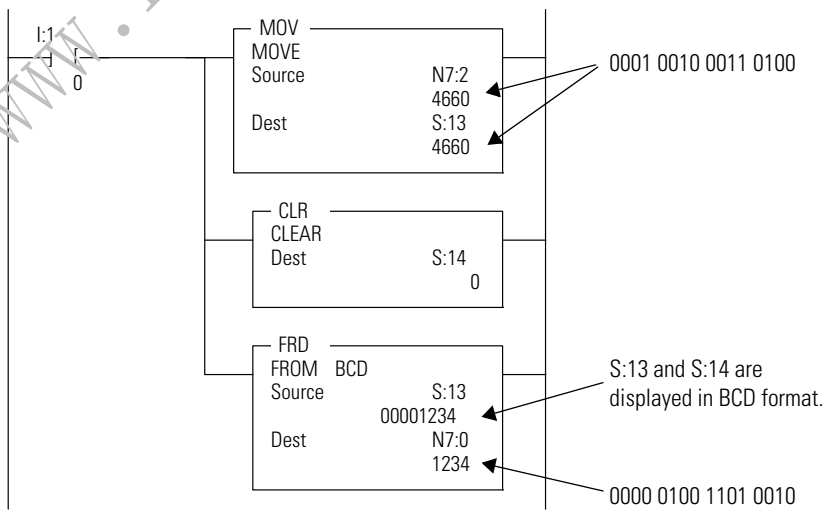
The BCD value 32,760 in the math register is converted and stored in N7:0. The maximum source value is 32767 (BCD).



You should convert BCD values to integer before you manipulate them in your ladder program. If you do not convert the values, the controller manipulates them as integers and their value may be lost.

TIP If the math register (S:13 and S:14) is used as the source for the FRD instruction and the BCD value does not exceed four digits, be sure to clear word S:14 before executing the FRD instruction. If S:14 is not cleared and a value is contained in this word from another math instruction located elsewhere in the program, an incorrect decimal value is placed in the destination word.

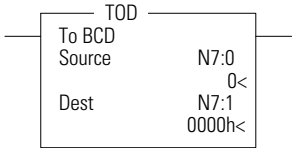
Clearing S:14 before executing the FRD instruction is shown below:



When the input condition I:0/1 is set (1), a BCD value (transferred from a 4-digit thumbwheel switch for example) is moved from word N7:2 into the math

register. Status word S:14 is then cleared to make certain that unwanted data is not present when the FRD instruction is executed.

TOD - Convert to Binary Coded Decimal (BCD)



Instruction Type: output

Execution Time for the TOD Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	5.9198 µs	0.3916 µs

The TOD instruction is used to convert the integer source value to BCD and place the result in the destination.

Addressing Modes and File Types can be used as shown in the following table:

TOD Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files											Function Files								CS0 - Comms	IOS - I/O	DLS - Data Log	Address Mode(1)			Address Level			
	O	I	S	B	T, C, R	N	F	L	MG, PD	Ri	PIS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•				•														•	•		•		
Destination	•	•	•	•	•	•				•														•	•		•		(2)

(1) See Important note about indirect addressing.

(2) See TOD Instruction Destination Operand below.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

TOD Instruction Destination Operand

The destination can be either a word address or math register.

The maximum values permissible once converted to BCD are:

- 9999 if the destination is a word address (allowing only a 4-digit BCD value)
- 32768 if the destination is the math register (allowing a 5-digit BCD value with the lower 4 digits stored in S:13 and the high order digit in S:14).

If the destination is the math register, it must be directly addressed as S:13. S:13 is the only status file element that can be used.

Updates to Math Status Bits

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always resets
S:0/1	Overflow	sets if BCD result is larger than 9999. On overflow, the minor error flag is also set.
S:0/2	Zero Bit	sets if result is zero, otherwise resets
S:0/3	Sign Bit	sets if the source word is negative, otherwise resets

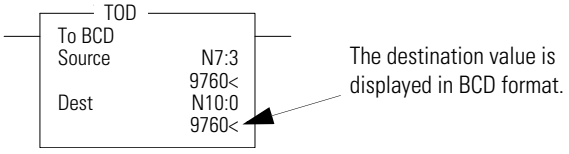
Changes to the Math Register

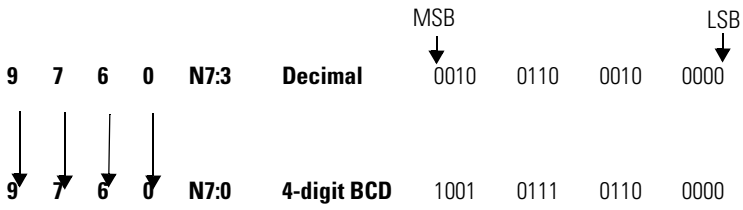
Contains the 5-digit BCD result of the conversion. This result is valid at overflow.

TIP To convert numbers larger than 9999 decimal, the destination must be the Math Register (S:13). You must reset the Minor Error Bit (S:5/0) to prevent an error.

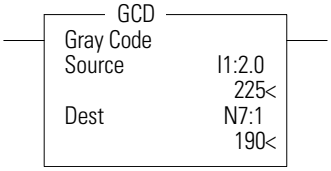
Example

The integer value 9760 stored at N7:3 is converted to BCD and the BCD equivalent is stored in N7:0. The maximum BCD value is 9999.





GCD - Gray Code



Instruction Type: output

Execution Time for the GCD Instructions

Controller	When Rung Is:	
	True	False
MicroLogix 1400	5.4970 µs	0.5618 µs

The GCD instruction converts Gray code data (Source) to an integer value (Destination). If the Gray code input is negative (high bit set), the Destination is set to 32767 and the overflow flag is set.

Addressing Modes and File Types are shown in the following table:

GCD Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 4-2.

Parameter	Data Files												Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•					•														•	•		•		
Destination	•	•		•	•	•					•														•	•		•		

Updates to Math Status Bits

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always reset
S:0/1	Overflow	set if the Gray code input is negative, otherwise is reset
S:0/2	Zero Bit	set if the destination is zero, otherwise reset
S:0/3	Sign Bit	always reset
S:5/0	Overflow Trap	set if the Overflow Bit is set, otherwise reset

Notes:

Logical Instructions

The logical instructions perform bit-wise logical operations on individual words.

Instruction	Used To:	Page
AND - Bit-Wise AND	Perform an AND operation	216
OR - Logical OR	Perform an inclusive OR operation	217
XOR - Exclusive OR	Perform an Exclusive Or operation	217
NOT - Logical NOT	Perform a NOT operation	218

Using Logical Instructions

When using logical instructions, observe the following:

- Source and Destination must be of the same data size (that is, all words or all long words).

IMPORTANT Do not use the High Speed Counter Accumulator (HSC.ACC) for the Destination parameter in the AND, OR, and XOR instructions.

- Source A and Source B can be a constant or an address, but both cannot be constants.
- Valid constants are -32768...32767 (word) and -2,147,483,648...2,147,483,647 (long word).

Addressing Modes and File Types can be used as shown in the following table:

Logical Instructions Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[2]								CS - Comms	IOS - I/O	DLS- Data Log	Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Source B[1]	•	•	•	•	•	•			•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•			•	•	•		•		•	•	•			•					•	•		•	•	

(1) Source B does not apply to the NOT instruction. The NOT instruction only has one source value.

(2) PTOX and PWMX files are valid for MicroLogix 1400 BXB or BXBA unit.

(3) See Important note about indirect addressing.

IMPORTANT	You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

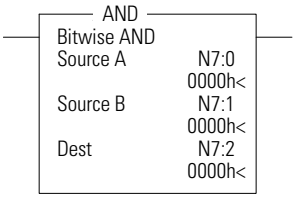
Updates to Math Status Bits

After a logical instruction is executed, the arithmetic status bits in the status file are updated. The arithmetic status bits are in word 0 bits 0-3 in the processor status file (S2).

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always resets
S:0/1	Overflow	always resets
S:0/2	Zero Bit	sets if result is zero, otherwise resets
S:0/3	Sign Bit	sets if result is negative (MSB is set), otherwise resets

AND - Bit-Wise AND



Instruction Type: output

Execution Time for the AND Instruction

Controller	Data Size	When Rung Is:	
		True	**False**
MicroLogix 1400	word	1.7894 µs	0.3781 µs
	long word	1.8185 µs	0.3967 µs

The AND instruction performs a bit-wise logical AND of two sources and places the result in the destination.

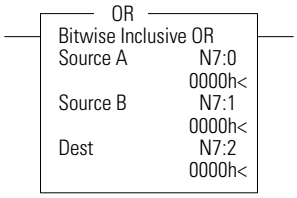
Truth Table for the AND Instruction

Destination = A AND B															
Source: A															
1	1	1	1	1	0	1	0	0	0	0	0	1	1	0	0
Source: B															
1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1
Destination:															
1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

IMPORTANT	Do not use the High Speed Counter Accumulator (HSC.ACC) for the Destination parameter in the AND, OR, and XOR instructions.

For more information, see Using Logical Instructions on page 215 and Updates to Math Status Bits on page 216.

OR - Logical OR



Instruction Type: output

Execution Time for the OR Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	1.8278 µs	0.3962 µs
	long word	1.8374 µs	0.3956 µs

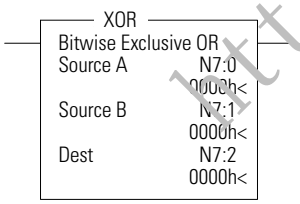
The OR instruction performs a logical OR of two sources and places the result in the destination.

Truth Table for the OR Instruction

Destination = A OR B															
Source: A															
1	1	1	1	1	0	1	0	0	0	0	0	1	1	0	0
Source: B															
1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1
Destination:															
1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1

IMPORTANT Do not use the High Speed Counter Accumulator (HSC.ACC) for the Destination parameter in the AND, OR, and XOR instructions.

XOR - Exclusive OR



Instruction Type: output

Execution Time for the XOR Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	4.9480 µs	0.3671 µs
	long word	4.8454 µs	0.3646 µs

The XOR instruction performs a logical exclusive OR of two sources and places the result in the destination.

Truth Table for the XOR Instruction

Destination = A XOR B															
Source: A															
1	1	1	1	1	0	1	0	0	0	0	0	1	1	0	0

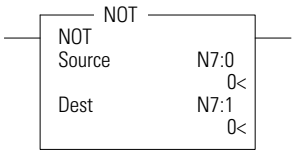
Truth Table for the XOR Instruction

Source: B																
1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1
Destination:																
0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	1

IMPORTANT Do not use the High Speed Counter Accumulator (HSC.ACC) for the Destination parameter in the AND, OR, and XOR instructions.

For more information, see Using Logical Instructions on page 215 and Updates to Math Status Bits on page 216.

NOT - Logical NOT



Instruction Type: output

Execution Time for the NOT Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	1.3682 µs	0.4074 µs
	long word	1.3620 µs	0.3900 µs

The NOT instruction is used to invert the source bit-by-bit (one's complement) and then place the result in the destination.

Truth Table for the NOT Instruction

Destination = A NOT B																
Source:																
1	1	1	1	1	0	1	0	0	0	0	0	1	1	0	0
Destination:																
0	0	0	0	0	1	0	1	1	1	1	1	0	0	1	1

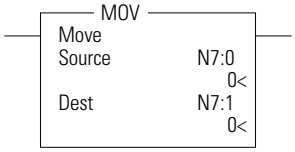
For more information, see Using Logical Instructions on page 215 and Updates to Math Status Bits on page 216.

Move Instructions

The move instructions modify and move words.

Instruction	Used to:	Page
MOV - Move	Move the source value to the destination.	219
MVM - Masked Move	Move data from a source location to a selected portion of the destination.	221

MOV - Move



Instruction Type: output

Execution Time for the MOV Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	1.4231 µs	0.3542 µs
	long word	1.4103 µs	0.3722 µs

The MOV instruction is used to move data from the source to the destination. As long as the rung remains true, the instruction moves the data each scan.

Using the MOV Instruction

When using the MOV instruction, observe the following:

- Source and Destination can be different data sizes. The source is converted to the destination size when the instruction executes. If the signed value of the Source does not fit in the Destination, the overflow is handled as follows:
  - If the Math Overflow Selection Bit is clear, a saturated result is stored in the Destination. If the Source is positive, the Destination is 32767 (word). If the result is negative, the Destination is -32768 (word).
  - If the Math Overflow Selection Bit is set, the unsigned truncated value of the Source is stored in the Destination.
- Source can be a constant or an address.
- Valid constants are -32,768...32,767 (word) and -2,147,483,648...2,147,483,647 (long word).

Addressing Modes and File Types can be used as shown in the following table:

MOV Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files[1]								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	
Destination	•	•	•	•	•	•	•	•	•	•	•			(2)	(2)	(2)	(2)								•	•		•	•	

(1) PTOX and PWMX files are valid for MicroLogix 1400 BXB or BXBA unit.

(2) Some elements can be written to. Consult the function file for details.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Updates to Math Status Bits

After a MOV instruction is executed, the arithmetic status bits in the status file are updated. The arithmetic status bits are in word 0, bits 0 to 3 in the processor status file (S2).

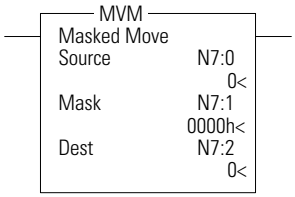
Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always resets
S:0/1	Overflow	sets when an overflow, infinity, or NAN (not a number) condition is detected, otherwise resets
S:0/2	Zero Bit	sets if result is zero, otherwise resets
S:0/3	Sign Bit	sets if result is negative (MSB is set), otherwise resets
S:5/0	Math Overflow Trap Bit[1]	sets Math Overflow Trap minor error if the Overflow bit is set, otherwise it remains in last state

(1) Control bit.

TIP If you want to move one word of data without affecting the math flags, use a copy (COP) instruction with a length of 1 word instead of the MOV instruction.

MVM - Masked Move



Instruction Type: output

Execution Time for the MVM Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	0.2210 µs	0.1750 µs
	long word	1.9050 µs	0.2180 µs

The MVM instruction is used to move data from the source to the destination, allowing portions of the destination to be masked. The mask bit functions as follows:

Mask Function for MVM Instruction

Source Bit	Mask Bit	Destination Bit
1	0	last state
0	0	last state
1	1	1
0	1	0

Mask data by setting bits in the mask to zero; pass data by setting bits in the mask to one. The mask can be a constant, or you can vary the mask by assigning a direct address. Bits in the Destination that correspond to zeros in the Mask are not altered.

Using the MVM Instruction

When using the MVM instruction, observe the following:

- Source, Mask, and Destination must be of the same data size (i.e. all words or all long words).

  To mask data, set the mask bit to zero; to pass data, set the mask bit to one. The mask can be a constant value, or you can vary the mask by assigning a direct address.

  TIP Bits in the destination that correspond to zeros in the mask are not altered as shown in the shaded areas in the following table.

Mask Example (Word Addressing Level)

Word	Value in Hexadecimal	Value in Binary															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value in Destination Before Move	FFFF	1	1	1	1	**1**	**1**	**1**	**1**	1	1	1	1	**1**	**1**	**1**	**1**
Source Value	5555	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Mask	F0F0	1	1	1	1	**0**	**0**	**0**	**0**	1	1	1	1	**0**	**0**	**0**	**0**
Value in Destination After Move	5F5F	0	1	0	1	**1**	**1**	**1**	**1**	0	1	0	1	**1**	**1**	**1**	**1**

- Valid constants for the mask are -32768...32767 (word) and -2,147,483,648...2,147,483,647 (long word). The mask is displayed as a hexadecimal unsigned value from 0000 0000...FFFF FFFF.

Addressing Modes and File Types can be used as shown in the following table:

MVM Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files											Function Files								Comms	IOS - I/O	DLS - Data Log	Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•		•	•		•														•	•		•	•	
Mask	•	•		•	•	•		•	•		•													•	•	•		•	•	
Destination	•	•		•	•	•		•	•		•														•	•		•	•	

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Updates to Math Status Bits

After a MVM instruction is executed, the arithmetic status bits in the status file are updated. The arithmetic status bits are in word 0 bits 0-3 in the processor status file (S2).

Math Status Bits

With this Bit:		The Controller:
S:0/0	Carry	always resets
S:0/1	Overflow	always resets
S:0/2	Zero Bit	sets if destination is zero, otherwise resets
S:0/3	Sign Bit	sets if the MSB of the destination is set, otherwise resets

Notes:

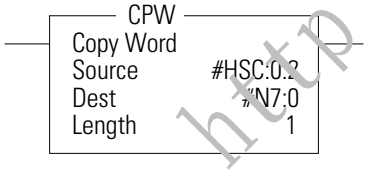
File Instructions

The file instructions perform operations on file data.

Table 1:

Instruction	Used To:	Page
CPW - Copy Word	Copy words of data from one location to another	225
COP - Copy File	Copy a range of data from one file location to another	226
FLL - Fill File	Load a file with a program constant or a value from an element address	228
BSL - Bit Shift Left	Load and unload data into a bit array one bit at a time	229
BSR - Bit Shift Right		231
FFL - First In, First Out (FIFO) Load	Load words into a file and unload them in the same order (first in, first out)	233
FFU - First In, First Out (FIFO) Unload		235
LFL - Last In, First Out (LIFO) Load	Load words into a file and unload them in reverse order (last in, first out)	237
LFU - Last In, First Out (LIFO) Unload		239
SWP - Swap	Swap low byte with high byte in a specified number of words	241

CPW - Copy Word



Instruction Type: output

Execution Time for the CPW Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	2.5630 µs	0.2034 µs

The CPW instruction copies words of data, in ascending order, from one location (Source) to another (Destination). Although similar to the File Copy (COP) instruction, the CPW instruction allows different source and destination parameters. Examples include:

- integer to long word
- long word to floating point
- long word to integer
- integer to PTOX function file

Observe the following restrictions when using the CPW instruction:

- The length of the data transferred cannot exceed 128 words.
- Function files can be used for Source or Destination, but not both.
- When referencing either a PLS file or a function file, addressing must be specified to the sub-element level.
- You can reference a sub-element of bits in a function file containing a combination of read-only and read/write bits.
- You cannot directly reference the high word of a long word as an operand in the CPW instruction.
- A Major fault (003F) is generated if the execution of the instruction exceeds the data table space.
- A Major fault (0044) is generated if a write attempt fails to the RTC function file. This only occurs when attempting to write invalid data to the RTC function file. Examples of invalid data are: setting the Day of Week to zero or setting the Date to February 30th.

Addressing Modes and File Types are shown in the following table:

CPW Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 4-2.

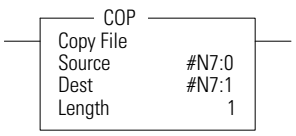
Parameter	Data Files													Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST[1]	A[1]	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•					•	•				•
Destination	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•	•								•	•				•
Length																									•						

(1) Valid for MicroLogix 1400 Series B only

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

COP - Copy File



Instruction Type: output

Execution Time for the COP Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	3.6020 µs	0.1853 µs

The COP instruction copies blocks of data from one location into another.

COP Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files													Function Files											Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•	•	•	•	•		•														•	•				•
Destination	•	•		•	•	•	•	•	•	•		•														•	•				•
Length																															

(1) See Important note about indirect addressing.

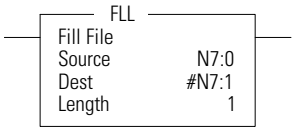
IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

The source and destination file types must be the same except bit (B) and integer (N); they can be interchanged. It is the address that determines the maximum length of the block to be copied, as shown in the following table:

Maximum Lengths for the COP Instruction

Source/Destination Data Type	Range of Length Operand
1 word elements (ie. word)	1…128
2 word elements (ie. long word)	1…64
3 word elements (ie. counter)	1…42
42 word elements (ie. string)	1…3

FLL - Fill File

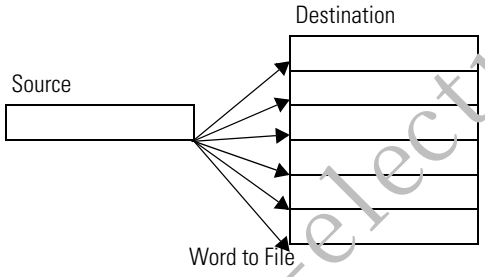


Instruction Type: output

Execution Time for the FLL Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	3.1531 µs	0.5290 µs
	long word	3.2470 µs	0.3918 µs

The FLL instruction loads elements of a file with either a constant or an address data value for a given length. The following figure shows how file instruction data is manipulated. The instruction fills the words of a file with a source value. It uses no status bits. If you need an enable bit, program a parallel output that uses a storage address.



This instruction uses the following operands:

- Source - The source operand is the address of the value or constant used to fill the destination. The data range for the source is from -32768...32767 (word) or -2,147,483,648...2,147,483,647 (long word), or any IEEE-754 32-bit value.

TIP A constant cannot be used as the source in a timer (T), counter (C), or control (R) file.

- Destination - The starting destination address where the data is written.
- Length - The length operand contains the number of elements. The length can range from 1...128 (word), 1...64 (long word), or 1...42 (3 word element such as counter).

TIP The source and destination operands must be of the same file type, unless they are bit (B) and integer (N).

Addressing Modes and File Types can be used as shown in the following table:

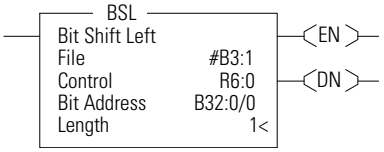
FLL Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files													Function Files											Address Mode[1]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•	•		•	•		•													•	•	•		•	•	•
Destination	•	•		•	•	•	•		•	•		•															•				•
Length																									•						

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

BSL - Bit Shift Left

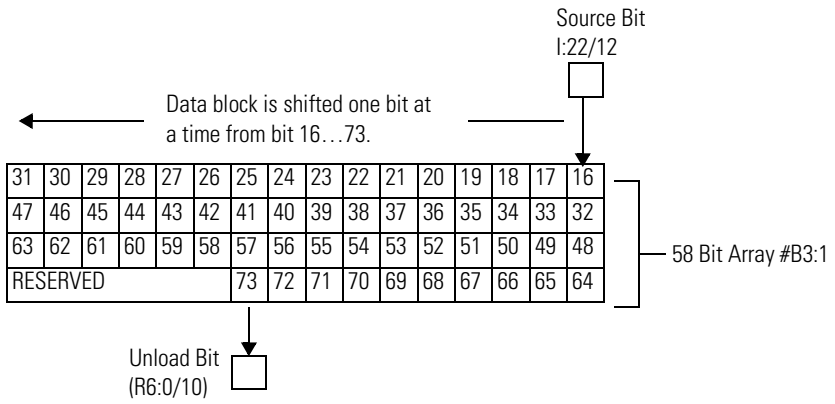


Instruction Type: output

Execution Time for the BSL Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	6.1018 µs	5.8258 µs

The BSL instruction loads data into a bit array on a false-to-true rung transition, one bit at a time. The data is shifted left through the array, then unloaded, one bit at a time. The following figure shows the operation of the BSL instruction.



If you wish to shift more than one bit per scan, you must create a loop in your application using the JMP, LBL, and CTU instructions.

This instruction uses the following operands:

- File - The file operand is the address of the bit array that is to be manipulated.
- Control - The control operand is the address of the BSL's control element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	--	ER(3)	UL(4)	not used									
Word 1	Size of bit array (number of bits).															
Word 2	not used															

(1) EN - Enable Bit is set on false-to-true transition of the rung and indicates the instruction is enabled.

(2) DN - Done Bit, when set, indicates that the bit array has shifted one position.

(3) ER - Error Bit, when set, indicates that the instruction detected an error such as entering a negative number for the length or source operand.

(4) UL - Unload Bit is the instruction's output. Avoid using the UL (unload) bit when the ER (error) bit is set.

- Bit Address - The source is the address of the bit to be transferred into the bit array at the first (lowest) bit position.
- Length - The length operand contains the length of the bit array in bits. The valid data range for length is from 0...2048.

Addressing Modes and File Types can be used as shown in the following table:

BSL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

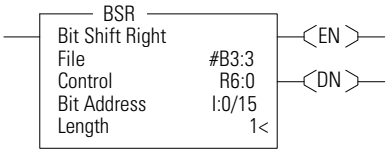
Parameter	Data Files												Function Files											Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
File		•		•		•			•		•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Source	•	•		•	•	•			•																	•	•			

(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

BSR - Bit Shift Right



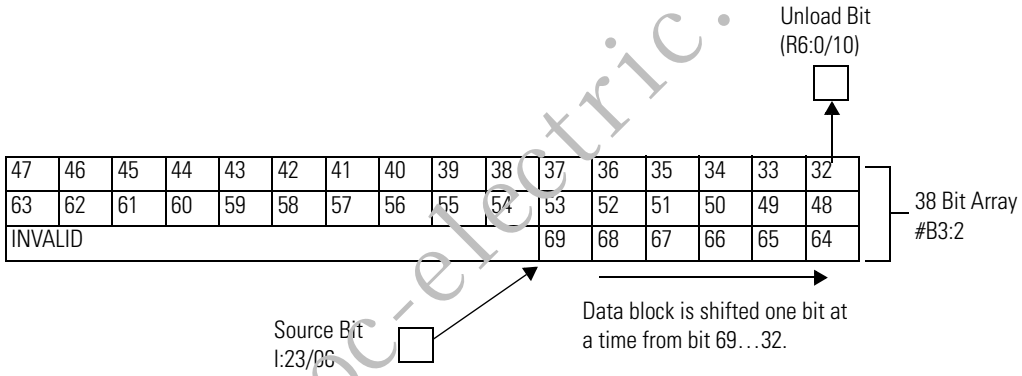
Instruction Type: output

Execution Time for the BSR Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	6.0790 µs	5.9942 µs

If you wish to shift more than one bit per scan, you must create a loop in your application using the JMP, LBL, and CTU instructions.

The BSR instruction loads data into a bit array on a false-to-true rung transition, one bit at a time. The data is shifted right through the array, then unloaded, one bit at a time. The following figure shows the operation of the BSR instruction.



This instruction uses the following operands:

- File - The file operand is the address of the bit array that is to be manipulated.
- Control - The control operand is the address of the BSR's control element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	--	ER(3)	UL(4)	not used									
Word 1	Size of bit array (number of bits).															
Word 2	not used															

(1) EN - Enable Bit is set on false-to-true transition of the rung and indicates the instruction is enabled.

(2) DN - Done Bit, when set, indicates that the bit array has shifted one position.

(3) ER - Error Bit, when set, indicates that the instruction detected an error such as entering a negative number for the length or source operand.

(4) UL - Unload Bit is the instruction's output. Avoid using the UL (unload) bit when the ER (error) bit is set.

- Bit Address - The source is the address of the bit to be transferred into the bit array at the last (highest) bit position.
- Length - The length operand contains the length of the bit array in bits. The data range for length is from 0...2048.

Addressing Modes and File Types can be used as shown in the following table:

BSR Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	L	ST	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
File	•	•		•		•		•			•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Source	•	•		•	•	•		•																	•	•	•			

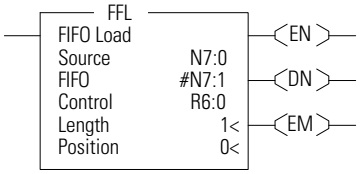
(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

FFL - First In, First Out (FIFO) Load

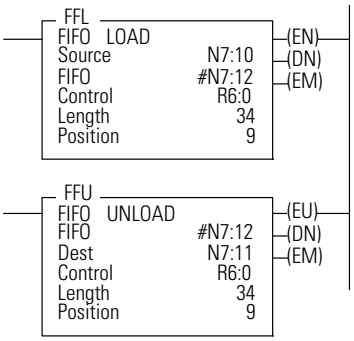
Instruction Type: output



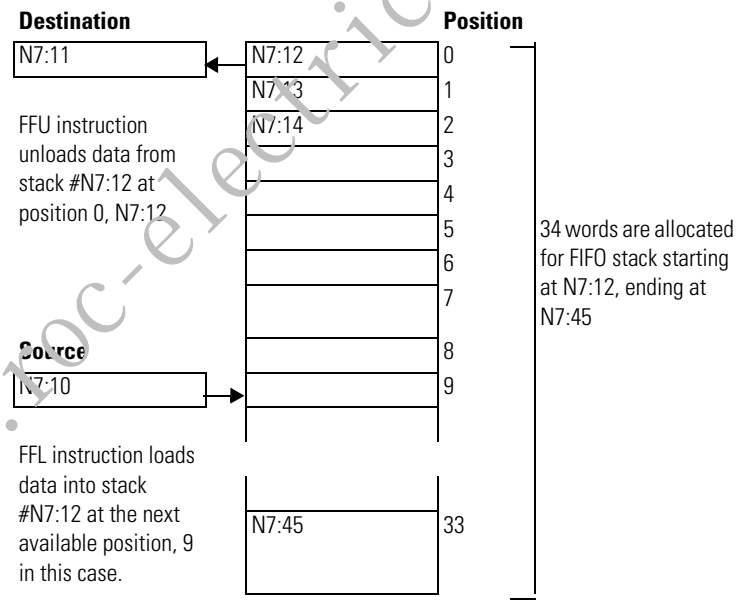
Execution Time for the FFL Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	8.2970 µs	6.1730 µs
	long word	9.0910 µs	7.4630 µs

On a false-to-true rung transition, the FFL instruction loads words or long words into a user-created file called a FIFO stack. This instruction's counterpart, FIFO unload (FFU), is paired with a given FFL instruction to remove elements from the FIFO stack. Instruction parameters have been programmed in the FFL - FFU instruction pair shown below.



FFL and FFU Instruction Pair



Loading and Unloading of Stack #N7:12

This instruction uses the following operands:

- Source - The source operand is a constant or address of the value used to fill the currently available position in the FIFO stack. The address level of the source must match the FIFO stack. If FIFO is a word size file, source must be a word value or constant. If FIFO is a long word size file, source must be a long word value or constant. The data range for the source is from -32768...32767 (word) or -2,147,483,648...2,147,483,647 (long word).
- FIFO - The FIFO operand is the starting address of the stack.
- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	EM(3)	not used											
Word 1	Length - maximum number of words or long words in the stack.															
Word 2	Position - the next available location where the instruction loads data.															

(1) EN - Enable Bit is set on false-to-true transition of the rung and indicates the instruction is enabled.

(2) DN - Done Bit, when set, indicates that the stack is full.

(3) EM - Empty Bit, when set, indicates FIFO is empty.

- Length - The length operand contains the number of elements in the FIFO stack to receive the value or constant found in the source. The length of the stack can range from 1...128 (word) or 1...64 (long word). The position is incremented after each load.
- Position - This is the current location pointed to in the FIFO stack. It determines the next location in the stack to receive the value or constant found in source. Position is a component of the control register. The position can range from 0...127 (word) or 0...63 (long word).

Addressing Modes and File Types can be used as shown in the following table:

FFL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

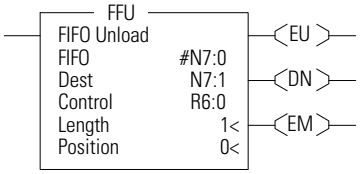
Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•			•		•													•	•	•		•	•	
FIFO	•	•		•		•			•		•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Position																								•				•		

(1) Control file only. Not valid for Timers or Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

FFU - First In, First Out (FIFO) Unload

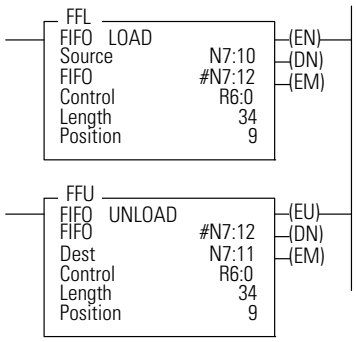


Instruction Type: output

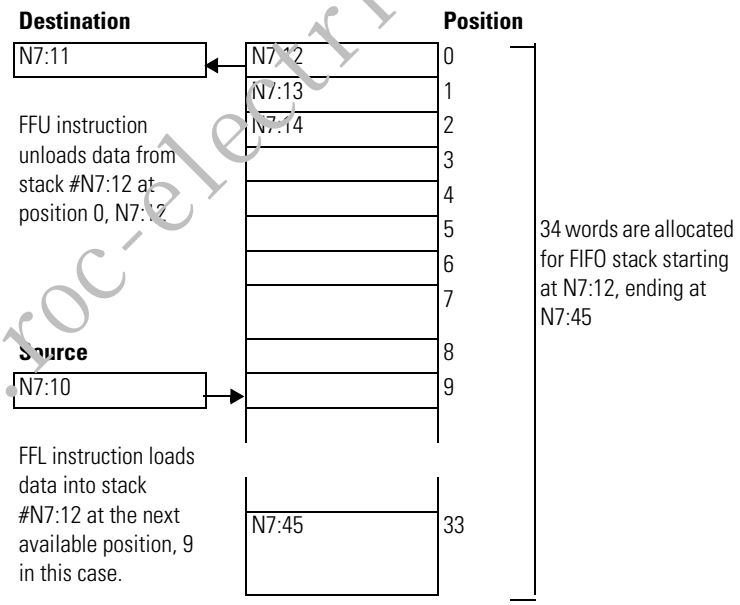
Execution Time for the FFU Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	8.7180 µs	6.6490 µs
	long word	9.8890 µs	7.2150 µs

On a false-to-true rung transition, the FFU instruction unloads words or long words from a user-created file called a FIFO stack. The data is unloaded using first-in, first-out order. After the unload completes, the data in the stack is shifted one element toward the top of the stack and the last element is zeroed out. Instruction parameters have been programmed in the FFL - FFU instruction pair shown below.



FFL and FFU Instruction Pair



Loading and Unloading of Stack #N7:12

This instruction uses the following operands:

- FIFO - The FIFO operand is the starting address of the stack.
- Destination - The destination operand is a word or long word address that stores the value which exits from the FIFO stack. The FFU instruction unloads this value from the first location on the FIFO stack and places it in the destination address. The address level of the destination must match the FIFO stack. If FIFO is a word size file, destination must be a word size file. If FIFO is a long word size file, destination must be a long word size file.

- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	--	EU(1)	DN(2)	EM(3)	not used											
Word 1	Length - maximum number of words or long words in the stack.															
Word 2	Position - the next available location where the instruction unloads data.															

(1) EU - Enable Unload Bit is set on false-to-true transition of the rung and indicates the instruction is enabled.

(2) DN - Done Bit, when set, indicates that the stack is full.

(3) EM - Empty Bit, when set, indicates FIFO is empty.

- Length - The length operand contains the number of elements in the FIFO stack. The length of the stack can range from 1...128 (word) or 1...64 (long word).
- Position - Position is a component of the control register. The position can range from 0...127 (word) or 0...63 (long word). The position is decremented after each unload. Data is unloaded at position zero.

Addressing Modes and File Types can be used as shown in the following table:

FFU Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files								Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
FIFO	•	•		•		•			•		•														•	•		•	•	
Destination	•	•		•	•	•			•		•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Position																								•				•		

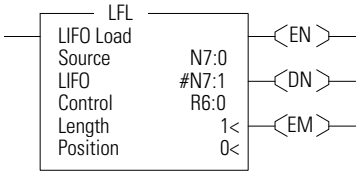
(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

LFL - Last In, First Out (LIFO) Load

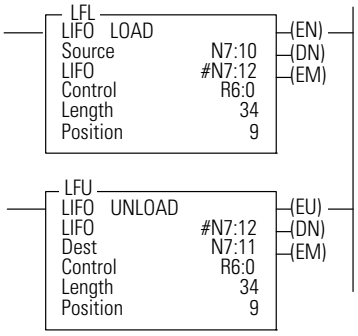
Instruction Type: output



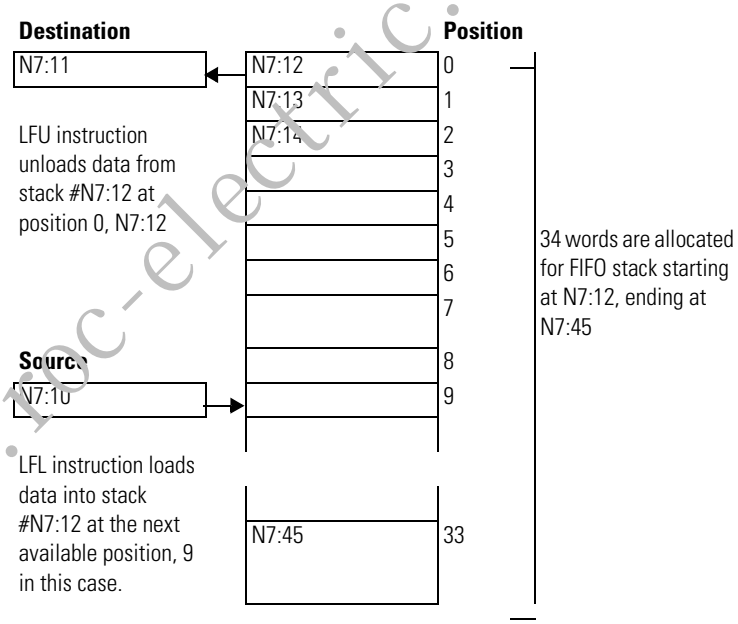
Execution Time for the LFL Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	6.4950 µs	6.5650 µs
	long word	7.3570 µs	7.0030 µs

On a false-to-true rung transition, the LFL instruction loads words or long words into a user-created file called a LIFO stack. This instruction's counterpart, LIFO unload (LFU), is paired with a given LFL instruction to remove elements from the LIFO stack. Instruction parameters have been programmed in the LFL - LFU instruction pair shown below.



LFL and LFU Instruction Pair



Loading and Unloading of Stack #N7:12

This instruction uses the following operands:

- Source - The source operand is a constant or address of the value used to fill the currently available position in the LIFO stack. The data size of the source must match the LIFO stack. If LIFO is a word size file, source must be a word value or constant. If LIFO is a long word size file, source must be a long word value or constant. The data range for the source is from -32768...32767 (word) or -2,147,483,648...2,147,483,647 (long word).
- LIFO - The LIFO operand is the starting address of the stack.
- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	EM(3)	not used											
Word 1	Length - maximum number of words or long words in the stack.															
Word 2	Position - the next available location where the instruction loads data.															

(1) EN - Enable Bit is set on false-to-true transition of the rung and indicates the instruction is enabled.

(2) DN - Done Bit, when set, indicates that the stack is full.

(3) EM - Empty Bit, when set, indicates that LIFO is empty.

- Length - The length operand contains the number of elements in the FIFO stack to receive the value or constant found in the source. The length of the stack can range from 1...128 (word) or 1...64 (long word). The position is incremented after each load.
- Position - This is the current location pointed to in the LIFO stack. It determines the next location in the stack to receive the value or constant found in source. Position is a component of the control register. The position can range from 0...127 (word) or 0...63 (long word).

Addressing Modes and File Types can be used as shown in the following table:

LFL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•			•		•													•	•	•		•	•	
LIFO	•	•		•		•			•		•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Position																								•				•		

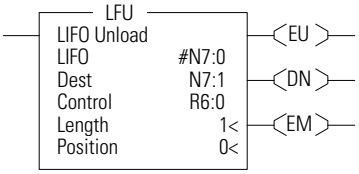
(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

LFU - Last In, First Out (LIFO) Unload

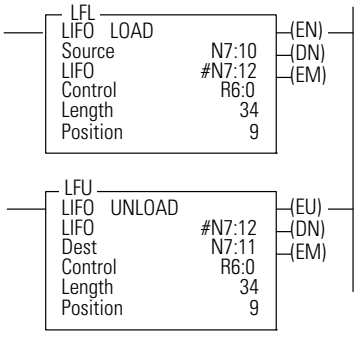
Instruction Type: output



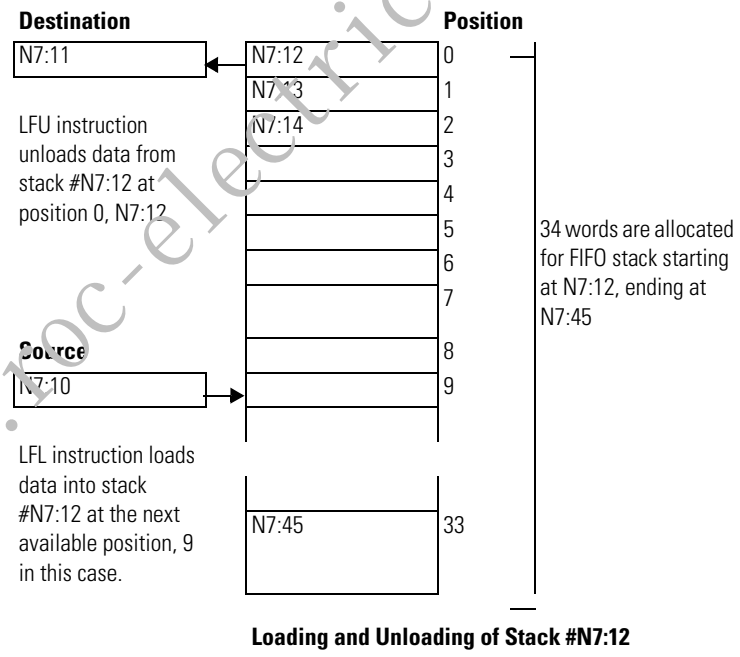
Execution Time for the LFU Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	6.8227 µs	6.5089 µs
	long word	7.6680 µs	7.2102 µs

On a false-to-true rung transition, the LFU instruction unloads words or long words from a user-created file called a LIFO stack. The data is unloaded using last-in, first-out order (the data in the stack is not cleared after unloading). Instruction parameters have been programmed in the LFL - LFU instruction pair shown below.



LFL and LFU Instruction Pair



Loading and Unloading of Stack #N7:12

This instruction uses the following operands:

- LIFO - The LIFO operand is the starting address of the stack.
- Destination - The destination operand is a word or long word address that stores the value which exits from the LIFO stack. The LFU instruction unloads this value from the last location on the LIFO stack and places it in the destination address. The address level of the destination must match the LIFO stack. If LIFO is a word size file, destination must be a word size file. If LIFO is a long word size file, destination must be a long word size file.
- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	--	EU(1)	DN(2)	EM(3)	not used											
Word 1	Length - maximum number of words or double words in the stack.															
Word 2	Position - the next available location where the instruction unloads data.															

(1) EU - Enable Unload Bit is set on false-to-true transition of the rung and indicates the instruction is enabled.

(2) DN - Done Bit, when set, indicates that the stack is full.

(3) EM - Empty Bit, when set, indicates LIFO is empty.

- Length - The length operand contains the number of elements in the LIFO stack. The length of the stack can range from 1...128 (word) or 1...64 (long word).
- Position - This is the next location in the LIFO stack where data will be unloaded. Position is a component of the control register. The position can range from 0...127 (word) or 0...63 (long word). The position is decremented after each unload.

LFU Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70

Parameter	Data Files												Function Files								Address Mode(2)			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
LIFO	•	•		•		•			•		•														•	•		•	•	
Destination	•	•		•	•	•			•		•														•	•		•	•	
Control					(1)																				•					•
Length																								•				•		
Position																								•				•		

(1) Control file only. Not valid for Timers and Counters.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

SWP - Swap

Instruction Type: output

Execution Time for the SWP Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	1.0728 µs	0.1963 µs

Use the SWP instruction to swap the low and high bytes of a specified number of words in a bit, integer, or string file. The SWP instruction has 2 operands:

- Source is the word address containing the words to be swapped.
- Length is the number of words to be swapped, regardless of the file type. The address is limited to integer constants. For bit and integer filetypes, the length range is 1...128. For the string filetype, the length range is 1...41. Note that this instruction is restricted to a single string element and cannot cross a string element boundary.

Addressing Modes and File Types can be used as shown in the following table:

SWP Instruction Valid Addressing Modes and File Types

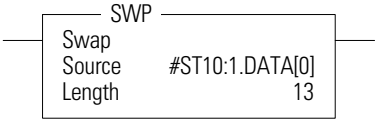
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files											Address[1] Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source				•		•		•	•			•															•		•		
Length																										•			•		

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

Example:



Source Value before executing SWP instruction: abcdefghijklmnopqrstuvwxyzabcdefg
Source Value before executing SWP instruction: badcfehgjilknmporqtsvuxwzyabcdefg

The underlined characters show the 13 words where the low byte was swapped with the high byte.

Notes:

Sequencer Instructions

Sequencer instructions are used to control automatic assembly machines or processes that have a consistent and repeatable operation. They are typically time based or event driven.

Instruction	Used To:	Page
SQC - Sequencer Compare	Compare 16-bit data with stored data	243
SQO - Sequencer Output	Transfer 16-bit data to word addresses	246
SQL - Sequencer Load	Load 16-bit data into a file	249

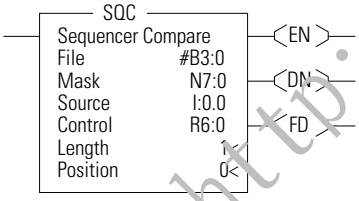
Use the sequencer compare instruction to detect when a step is complete; use the sequencer output instruction to set output conditions for each step. Use the sequencer load instruction to load data into the sequencer file.

The primary advantage of sequencer instructions is to conserve program memory. These instructions monitor and control 16 (word) or 32 (long word) discrete outputs at a time in a single rung.

You can use bit integer or double integer files with sequencer instructions.

SQC- Sequencer Compare

Instruction Type: output



Execution Time for the SQC Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	3.1762 µs	0.8505 µs
	long word	3.2480 µs	0.9823 µs

On a false-to-true rung transition, the SQC instruction is used to compare masked source words or long words with the masked value at a reference address (the sequencer file) for the control of sequential machine operations.

When the status of all non-masked bits in the source word match those of the corresponding reference word, the instruction sets the found bit (FD) in the control word. Otherwise, the found bit (FD) is cleared.

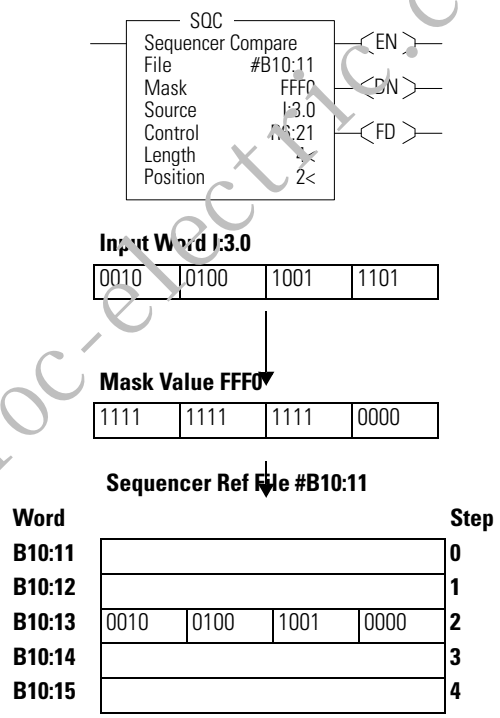
The bits mask data when reset (0) and pass data when set (1).

The mask can be fixed or variable. If you enter a hexadecimal code, it is fixed. If you enter an element address or a file address (direct or indirect) for changing the mask with each step, it is variable.

When the rung goes from false-to-true, the instruction increments to the next step (word) in the sequencer file. Data stored there is transferred through a mask and compared against the source for equality. While the rung remains true, the source is compared against the reference data for every scan. If equal, the FD bit is set in the SQCs control counter.

Applications of the SQC instruction include machine diagnostics.

The following figure explains how the SQC instruction works.



SQC FD bit is set when the instruction detects that an input word matches (through mask) its corresponding reference word.

The FD bit R6:21/FD is set in the example, since the input word matches the sequencer reference value using the mask value.

This instruction uses the following operands:

- File - This is the sequencer reference file. Its contents, on an element-by-element basis, are masked and compared to the masked value stored in source.

TIP If file type is word, then mask and source must be words. If file type is long word, mask and source must be long words.

- Mask - The mask operand contains the mask constant, word, or file which is applied to both file and source. When mask bits are set to 1, data is allowed to pass through for comparison. When mask bits are reset to 0, the data is masked (does not pass through to for comparison). The immediate data ranges for mask are from 0...0xFFFF or 0...0xFFFFFFFF.

  TIP If mask is direct or indirect, the position selects the location in the specified file.

- Source - This is the value that is compared to file.
- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	--	ER(3)	not used		FD(4)	not used							
Word 1	Length - contains the number of steps in the sequencer reference file.															
Word 2	Position - the current position in the sequence															

(1) EN - Enable Bit is set by a false-to-true rung transition and indicates that the instruction is enabled.

(2) DN - Done Bit is set after the instruction has operated on the last word in the sequencer file. It is reset on the next false-to-true rung transition after the rung goes false.

(3) ER - Error Bit is set when the controller detects a negative position value, or a negative or zero length value. When the ER bit is set, the minor error bit (S2:5/2) is also set.

(4) FD - Found bit is set when the status of all non-masked bits in the source address match those of the word in the sequencer reference file. This bit is assessed each time the SQC instruction is evaluated while the rung is true.

- Length - The length operand contains the number of steps in the sequencer file (as well as Mask and/or Source if they are file data types). The length of the sequencer can range from 1...256.
- Position - This is the current location or step in the sequencer file (as well as Mask and/or Source if they are file data types). It determines the next location in the stack to receive the current comparison data. Position is a component of the control register. The position can range from 0...255 for words and 0...127 for long words. The position is incremented on each false-to-true transition.

Addressing Modes and File Types can be used as shown in the following table:

SQC Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

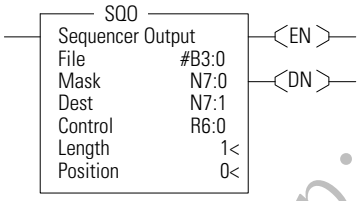
Parameter	Data Files												Function Files											Address Mode[2]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
File	•	•		•		•			•		•														•	•		•	•	
Mask	•	•		•	•	•			•		•														•	•		•	•	
Source	•	•		•	•	•			•		•														•	•		•	•	
Control					(1)																					•				•
Length																								•				•		
Position																								•				•		

(1) Control file only.

(2) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, LCD, and DLS files.

SQO- Sequencer Output



Instruction Type: output

Execution Time for the SQO Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	3.6105 µs	0.9480 µs
	Long word	3.1920 µs	1.1850 µs

On a false-to-true rung transition, the SQO instruction transfers masked source reference words or long words to the destination for the control of sequential machine operations. When the rung goes from false-to-true, the instruction increments to the next step (word) in the sequencer file. Data stored there is transferred through a mask to the destination address specified in the instruction. Data is written to the destination word every time the instruction is executed.

The done bit is set when the last word of the sequencer file is transferred. On the next false-to-true rung transition, the instruction resets the position to step one.

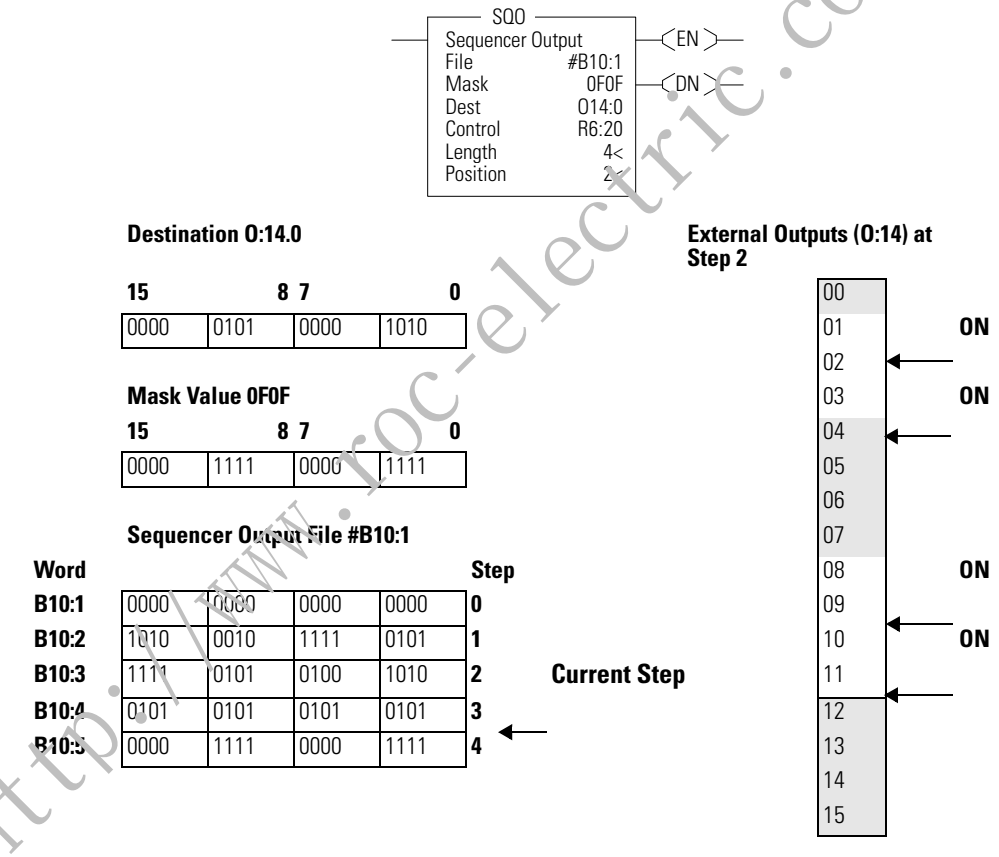
If the position is equal to zero at start-up, when you switch the controller from the program mode to the run mode, the instruction operation depends on whether the rung is true or false on the first scan.

- If the rung is true, the instruction transfers the value in step zero.
- If the rung is false, the instruction waits for the first rung transition from false-to-true and transfers the value in step one.

The bits mask data when reset (0) and pass data when set (1). The instruction will not change the value in the destination word unless you set mask bits.

The mask can be fixed or variable. It is fixed if you enter a hexadecimal code. It is variable if you enter an element address or a file address (direct or indirect) for changing the mask with each step.

The following figure indicates how the SQO instruction works.



This instruction uses the following operands:

- File - This is the sequencer reference file. Its contents, on an element-by-element, basis are masked and stored in the destination.

  TIP If file type is word, then mask and source must be words. If file type is long word, mask and source must be long words.

- Mask - The mask operand contains the mask value. When mask bits are set to 1, data is allowed to pass through to destination. When mask bits are reset to 0, the data is masked (does not pass through to destination). The immediate data ranges for mask are from 0...0xFFFF (word) or 0...0xFFFFFFFF (long word).

  TIP If mask is direct or indirect, the position selects the location in the specified file.

- Destination - The destination operand is the sequencer location or file.
- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	--	ER(3)	not used		FD	not used							
Word 1	Length - contains the index of the last element in the sequencer reference file															
Word 2	Position - the current position in the sequence															

(1) EN - Enable Bit is set by a false-to-true rung transition and indicates that the instruction is enabled.

(2) DN - Done Bit is set after the instruction has operated on the last word in the sequencer file. It is reset on the next false-to-true rung transition after the rung goes false.

(3) ER - Error Bit is set when the controller detects a negative position value, or a negative or zero length value. When the ER bit is set, the minor error bit (S2:5/2) is also set.

- Length - The length operand contains the number of steps in the sequencer file (as well as Mask and/or Destination if they are file data types). The length of the sequencer can range from 1...256.
- Position - This is the current location or step in the sequencer file (as well as Mask and/or Destination if they are file data types). It determines the next location in the stack to be masked and moved to the destination. Position is a component of the control register. The position can range from 0...255. Position is incremented on each false-to-true transition.

Addressing Modes and File Types can be used as shown in the following table:

SQO Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Element	Bit	Word	Long Word
File[1]	•	•		•		•			•		•														•	•			•	•
Mask[1]	•	•		•	•	•			•		•														•	•			•	•
Destination[1]	•	•		•	•	•			•		•														•	•			•	•
Control					[2]																				•		•			
Length																								•					•	
Position																								•					•	

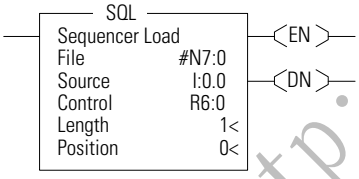
(1) File Direct and File Indirect addressing also applies.

(2) Control file only.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, LCD, and DLS files.

SQL - Sequencer Load



Instruction Type: output

Execution Time for the SQL Instruction

Controller	Data Size	When Rung Is:	
		True	False
MicroLogix 1400	word	2.7700 µs	1.1741 µs
	long word	2.8680 µs	1.2800 µs

On a false-to-true rung transition, the SQL instruction loads words or long words into a sequencer file at each step of a sequencer operation. This instruction uses the following operands:

- File - This is the sequencer reference file. Its contents are received on an element-by-element basis from the source.

TIP If file type is word, then mask and source must be words. If file type is long word, mask and source must be long words.

- Source - The source operand is a constant or address of the value used to fill the currently available position sequencer file. The address level of the source must match the sequencer file. If file is a word type, then source must be a word type. If file is a long word type, then source must be a long word type. The data range for the source is from -32,768...32,767 (word) or -2,147,483,648...2,147,483,647 (long word).
- Control - This is a control file address. The status bits, stack length, and the position value are stored in this element. The control element consists of 3 words:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word 0	EN(1)	--	DN(2)	--	ER(3)	not used		FD	not used							
Word 1	Length - contains the index of the last element in the sequencer reference file															
Word 2	Position - the current position in the sequence															

(1) EN - Enable Bit is set by a false-to-true rung transition and indicates that the instruction is enabled.

(2) DN - Done Bit is set after the instruction has operated on the last word in the sequencer file. It is reset on the next false-to-true rung transition after the rung goes false.

(3) ER - Error Bit is set when the controller detects a negative position value, or a negative or zero length value. When the ER bit is set, the minor error bit (S2:5/2) is also set.

- Length - The length operand contains the number of steps in the sequencer file (this is also the length of source if it is a file data type). The length of the sequencer can range from 1...256.
- Position - This is the current location or step in the sequencer file (as well as source if it is a file data type). It determines the next location in the stack to receive the value or constant found in source. Position is a component of the control register. The position can range from 0...255.

Addressing Modes and File Types can be used as shown in the following table:

SQL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode[3]			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
File[1]	•	•		•		•			•		•														•	•		•	•	
Source[1]	•	•		•		•			•		•														•	•		•	•	
Control					(2)																				•					•
Length																								•				•		
Position																								•				•		

(1) File Direct and File Indirect addressing also applies.

(2) Control file only.

(3) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, LCD, and DLS files.

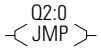
Notes:

Program Control Instructions

Use these instructions to change the order in which the processor scans a ladder program. Typically these instructions are used to minimize scan time, create a more efficient program, and troubleshoot a ladder program.

Instruction	Used To:	Page
JMP - Jump to Label	Jump forward/backward to a corresponding label instruction	253
LBL - Label		254
JSR - Jump to Subroutine	Jump to a designated subroutine and return	254
SBR - Subroutine Label		254
RET - Return from Subroutine		255
SUS - Suspend	Debug or diagnose your user program	255
TND - Temporary End	Abort current ladder scan	255
END - Program End	End a program or subroutine	256
MCR - Master Control Reset	Enable or inhibit a master control zone in your ladder program	256

JMP - Jump to Label



Instruction Type: output

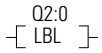
Execution Time for the JMP Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.3290 µs	0.2320 µs

The JMP instruction causes the controller to change the order of ladder execution. Jumps cause program execution to go to the rung marked LBL *label number*. Jumps can be forward or backward in ladder logic within the same program file. Multiple JMP instructions may cause execution to proceed to the same label.

The immediate data range for the label is from 0...999. The label is local to a program file.

LBL - Label



Instruction Type: input

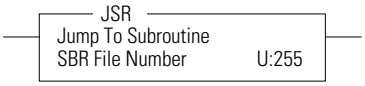
Execution Time for the LBL Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.2633 µs	

The LBL instruction is used in conjunction with a jump (JMP) instruction to change the order of ladder execution. Jumps cause program execution to go to the rung marked LBL *label number*.

The immediate data range for the label is from 0...999. The label is local to a program file.

JSR - Jump to Subroutine



Instruction Type: output

Execution Time for the JSR Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.4615 µs	0.2325 µs

The JSR instruction causes the controller to start executing a separate subroutine file within a ladder program. JSR moves program execution to the designated subroutine (SBR *file number*). After executing the SBR, control proceeds to the instruction following the JSR instruction.

The immediate data range for the JSR file is from 3...255.

SBR - Subroutine Label

Instruction Type: input

Execution Time for the SBR Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.2510 µs	0.2510 µs

The SBR instruction is a label which is not used by the processor. It is for user subroutine identification purposes as the first rung for that subroutine. This instruction is the first instruction on a rung and is always evaluated as true.

RET - Return from Subroutine

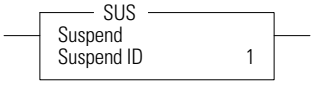
Instruction Type: output

Execution Time for the RET Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.3710 µs	0.2510 µs

The RET instruction marks the end of subroutine execution or the end of the subroutine file. It causes the controller to resume execution at the instruction following the JSR instruction, user interrupt, or user fault routine that caused this subroutine to execute.

SUS - Suspend



Instruction Type: output

The SUS instruction is used to trap and identify specific conditions for program debugging and system troubleshooting. This instruction causes the processor to enter the suspend idle mode, causing all outputs to be de-energized. The suspend ID and the suspend file (program file number or subroutine file number identifying where the suspend instruction resides) are placed in the status file (S:7 and S:8).

The immediate data range for the suspend ID is from -32768...32767.

TND - Temporary End

Instruction Type: output

Execution Time for the TND Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.3320 µs	0.2100 µs

The TND instruction is used to denote a premature end-of-ladder program execution. The TND instruction cannot be executed from a STI subroutine, HSC subroutine, EII subroutine, or a user fault subroutine. This instruction may appear more than once in a ladder program.

On a true rung within the main program (file 2), TND will stop the processor from scanning the rest of the main program and go directly to the end-of-scan aspects of the processor scan cycle. On a true rung within a subroutine program, TND will return from the subroutine and continue to scan the rest of the main program (file 2). If this instruction is executed in a nested subroutine, it terminates execution of all nested subroutines.

END - Program End

–(END)–

Instruction Type: output

Execution Time for the END Instruction

Controller	Instruction	When Rung Is:	
		True	False
MicroLogix 1400	END	1.2016 µs	1.2032 µs

The END instruction must appear at the end of every ladder program. For the main program file (file 2), this instruction ends the program scan. For a subroutine, interrupt, or user fault file, the END instruction causes a return from subroutine.

MCR - Master Control Reset

–(MCR)–

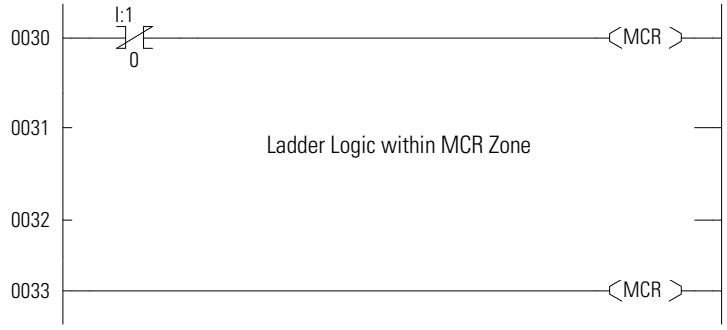
Instruction Type: output

Execution Time for the MCR Instructions

Controller	Instruction	When Rung Is:	
		True	False
MicroLogix 1400	MCR Start	1.0510 µs	1.1510 µs
	MCR End	0.4510 µs	0.5510 µs

The MCR instruction works in pairs to control the ladder logic found between those pairs. Rungs within the MCR zone are still scanned, but scan time is reduced due to the false state of non-retentive outputs. Non-retentive outputs are reset when the rung goes false.

This instruction defines the boundaries of an MCR Zone. An MCR Zone is the set of ladder logic instructions bounded by an MCR instruction pair. The start of an MCR zone is defined to be the rung that contains an MCR instruction preceded by conditional logic. The end of an MCR zone is defined to be the first rung containing just an MCR instruction following a start MCR zone rung as shown below.



While the rung state of the first MCR instruction is true, execution proceeds as if the zone were not present. When the rung state of the first MCR instruction is

false, the ladder logic within the MCR zone is executed as if the rung is false. All non-retentive outputs within the MCR zone are reset.

MCR zones let you enable or inhibit segments of your program, such as for recipe applications.

When you program MCR instructions, note that:

- You must end the zone with an unconditional MCR instruction.
- You cannot nest one MCR zone within another.
- Do not jump into an MCR zone. If the zone is false, jumping into it activates the zone.

TIP The MCR instruction is not a substitute for a hard-wired master control relay that provides emergency stop capability. You still must install a hard-wired master control relay to provide emergency I/O power shutdown.

ATTENTION: If you start instructions such as timers or counters in an MCR zone, instruction operation ceases when the zone is disabled. Re-program critical operations outside the zone if necessary.

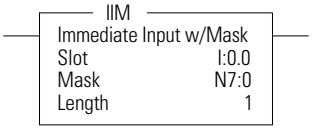
Notes:

Input and Output Instructions

The input and output instructions allow you to selectively update data without waiting for the input and output scans.

Instruction	Used To:	Page
IIM - Immediate Input with Mask	Update data prior to the normal input scan.	259
IOM - Immediate Output with Mask	Update outputs prior to the normal output scan.	260
REF - I/O Refresh	Interrupt the program scan to execute the I/O scan (write outputs, service communications, read inputs)	261

IIM - Immediate Input with Mask



Instruction Type: output

TIP This instruction is used for embedded I/O only. It is not designed to be used with expansion I/O.

Controller	When Rung Is:	
	True	False
MicroLogix 1400	10.9098 µs	0.2064 µs

The IIM instruction allows you to selectively update input data without waiting for the automatic input scan. This instruction uses the following operands:

- **Slot -** This operand defines the location where data is obtained for updating the input file. The location specifies the slot number and the word where data is to be obtained. For example, if slot = I:0, input data from slot 0 starting at word 0 is masked and placed in input data file I:0 starting at word 0 for the specified length. If slot = I:0.1, word 1 of slot 0 is used, and so on.

IMPORTANT Slot 0 is the only valid slot number that can be used with this instruction. IIM cannot be used with expansion I/O.

- **Mask -** The mask is a hex constant or register address containing the mask value to be applied to the slot. If a given bit position in the mask is a "1", the corresponding bit data from slot is passed to the input data file. A "0" prohibits corresponding bit data in slot from being passed to the input data file. The mask value can range from 0...0xFFFF.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Real Input	Input Word															
Mask	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Input Data File	Data is Not Updated								Updated to Match Input Word							

- **Length -** This is the number of masked words to transfer to the input data file.

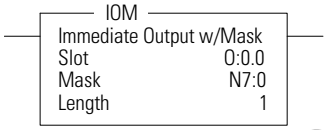
Addressing Modes and File Types can be used as shown below:

IIM Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 66.

Parameter	Data Files												Function Files									Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Slot		•									•														•			•		
Mask	•	•		•	•	•																		•	•	•		•		
Length																								•						

IOM - Immediate Output with Mask

Instruction Type: output



TIP This instruction is used for embedded I/O only. It is not designed to be used with expansion I/O.

Controller	When Rung Is:	
	True	False
MicroLogix 1400	10.4010 µs	0.3220 µs

The IOM instruction allows you to selectively update output data without waiting for the automatic output scan. This instruction uses the following operands:

- **Slot** - The slot is the physical location that is updated with data from the output file.

IMPORTANT Slot 0 is the only valid slot number that can be used with this instruction. IOM cannot be used with expansion I/O.

- **Mask -** The mask is a hex constant or register address containing the mask value to be applied. If a given bit position in the mask is a "1", the corresponding bit data is passed to the physical outputs. A "0" prohibits corresponding bit data from being passed to the outputs. The mask value can range from 0...0xFFFF.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output Data	Output Word															
Mask	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Real Outputs	Data is Not Updated								Updated to Match Output Word							

- **Length -** This is the number of masked words to transfer to the outputs.

Addressing Modes and File Types can be used as shown below:

IOM Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 66.

Parameter	Data Files												Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTO, PWM	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Slot	•										•														•			•		
Mask	•	•		•	•	•																		•	•	•		•		
Length																								•						

REF- I/O Refresh

Instruction Type: output

Execution Time for the REF Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	See MicroLogix 1400 Scan Time Calculation on page 501	0.1490 µs

The REF instruction is used to interrupt the program scan to execute the I/O scan and service communication portions of the operating cycle for all communication channels. This includes: write outputs, service communications (all communication channels, communications toggle functionality, and comms housekeeping), and read inputs.

The REF instruction has no programming parameters. When it is evaluated as true, the program scan is interrupted to execute the I/O scan and service communication portions of the operating cycle. The scan then resumes at the instruction following the REF instruction.

The REF instruction cannot be executed from an STI subroutine, HSC subroutine, EII subroutine, or a user fault subroutine.

TIP Using an REF instruction may result in input data changing in the middle of a program scan. This condition needs to be evaluated when using the REF instruction.

ATTENTION: The watchdog and scan timers are reset when executing the REF instruction. You must insure that the REF instruction is not placed inside a non-terminating program loop. Do not place the REF instruction inside a program loop unless the program is thoroughly analyzed.

Using Interrupts

Interrupts allow you to interrupt your program based on defined events. This chapter contains information about using interrupts, the interrupt instructions, and the interrupt function files. The chapter is arranged as follows:

- Information About Using Interrupts on page 263.
- User Interrupt Instructions on page 267.
- Using the Selectable Timed Interrupt (STI) Function File on page 272.
- Using the Event Input Interrupt (EII) Function File on page 276.

See also: Using the High-Speed Counter and Programmable Limit Switch on page 77.

Information About Using Interrupts

The purpose of this section is to explain some fundamental properties of the User Interrupts, including:

- What is an interrupt?
- When can the controller operation be interrupted?
- Priority of User Interrupts
- Interrupt Latency
- User Fault Routine

What is an Interrupt?

An interrupt is an event that causes the controller to suspend the task it is currently performing, perform a different task, and then return to the suspended task at the point where it suspended. The Micrologix 1400 supports the following User Interrupts:

- User Fault Routine
- Event Interrupts (8)
- High-Speed Counter Interrupts (6)
- Selectable Timed Interrupt

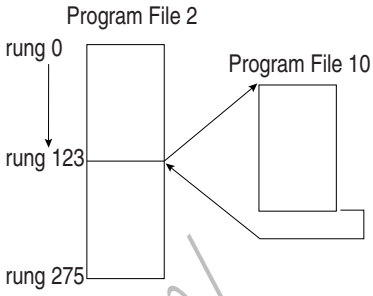
An interrupt must be configured and enabled to execute. When any one of the interrupts is configured (and enabled) and subsequently occurs, the user program:

1. suspends its execution

2. performs a defined task based upon which interrupt occurred
3. returns to the suspended operation.

Interrupt Operation Example
Program File 2 is the main control program.
Program File 10 is the interrupt routine.

- An Interrupt Event occurs at rung 123.
- Program File 10 is executed.
- Program File 2 execution resumes immediately after program file 10 is scanned.



Specifically, if the controller program is executing normally and an interrupt event occurs:

1. the controller stops its normal execution
2. determines which interrupt occurred
3. goes immediately to rung 0 of the subroutine specified for that User Interrupt
4. begins executing the User Interrupt subroutine (or set of subroutines if the specified subroutine calls a subsequent subroutine)
5. completes the subroutine(s)
6. resumes normal execution from the point where the controller program was interrupted

When Can the Controller Operation be Interrupted?

The Micrologix 1400 controllers only allow interrupts to be serviced during certain periods of a program scan. They are:

- At the start of a ladder rung
- Anytime during End of Scan

The interrupt is only serviced by the controller at these opportunities. If the interrupt is disabled, the pending bit is set at the next occurrence of one of the three occasions listed above.

ATTENTION: If you enable interrupts during the program scan via an OTL, OTE, or UIE, this instruction (OTL, OTE, or UIE) *must* be the *last* instruction executed on the rung (last instruction on last branch). It is recommended this be the only output instruction on the rung.

Priority of User Interrupts

When multiple interrupts occur, the interrupts are serviced based upon their individual priority.

When an interrupt occurs and another interrupt(s) has already occurred but has not been serviced, the new interrupt is scheduled for execution based on its priority relative to the other pending interrupts. At the next point in time when an interrupt can be serviced, all the interrupts are executed in the sequence of highest priority to lowest priority.

If an interrupt occurs while a lower priority interrupt is being serviced (executed), the currently executing interrupt routine is suspended, and the higher priority interrupt is serviced. Then the lower priority interrupt is allowed to complete before returning to normal processing.

If an interrupt occurs while a higher priority interrupt is being serviced (executed), and the pending bit has been set for the lower priority interrupt, the currently executing interrupt routine continues to completion. Then the lower priority interrupt runs before returning to normal processing.

The priorities from highest to lowest are:

User Fault Routine	**highest priority**
Event Interrupt0	
Event Interrupt1	
High-Speed Counter Interrupt0	
Event Interrupt2	
Event Interrupt3	
High-Speed Counter Interrupt1	
Selectable Timed Interrupt	
Event Interrupt4	
High-Speed Counter Interrupt2	
Event Interrupt5	
High-Speed Counter Interrupt3	
Event Interrupt6	
High-Speed Counter Interrupt4	
Event Interrupt7	
High-Speed Counter Interrupt5	**lowest priority**

User Fault Routine

The user fault routine gives you the option of preventing a controller shutdown when a specific user fault occurs. The fault routine is executed when any

recoverable or non-recoverable user fault occurs. The fault routine is not executed for non-user faults.

Faults are classified as recoverable, non-recoverable, and non-user faults. A complete list of faults is shown in Fault Messages and Error Codes on page 525. The basic types of faults are described below:

Fault Types

Recoverable	Non-Recoverable	Non-User Fault
Recoverable Faults are caused by the user and may be recovered from by executing logic in the user fault routine. The user can attempt to clear the Major Error Halted bit, S:1/13. **Note:** You may initiate a MSG instruction from the controller to another device to identify the fault condition of the controller.	Non-Recoverable Faults are caused by the user, and cannot be recovered from. The user fault routine executes when this type of fault occurs. However, the fault cannot be cleared. **Note:** You may initiate a MSG instruction to another device to identify the fault condition of the controller.	Non-User Faults are caused by various conditions that cease ladder program execution. The user fault routine does not execute when this type of fault occurs.

Status File Data Saved

The Arithmetic Flags (Status File word S:0) are saved on entry to the user fault subroutine and re-written upon exiting the subroutine.

Creating a User Fault Subroutine

To use the user fault subroutine:

1. Create a subroutine file. Program Files 3...255 can be used.
2. Enter the file number in word S:29 of the status file.

Controller Operation

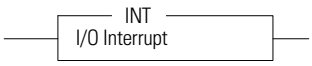
The occurrence of recoverable or non-recoverable faults causes the controller to read S:29 and execute the subroutine number identified by S:29. If the fault is recoverable, the routine can be used to correct the problem and clear the fault bit S:1/13. The controller then continues in its current executing mode. The routine does not execute for non-user faults.

User Interrupt Instructions

User Interrupt Instructions

Instruction	Used To:	Page
INT - Interrupt Subroutine	Use this instruction to identify a program file as an interrupt subroutine (INT label) versus a regular subroutine (SBR label). This should be the first instruction in your interrupt subroutine.	267
STS - Selectable Timed Start	Use the STS (Selectable Timed Interrupt Start) instruction to the start the STI timer from the control program, rather than starting automatically.	267
UID - User Interrupt Disable	Use the User Interrupt Disable (UID) and the User Interrupt Enable (UIE) instructions to create zones in which I/O interrupts cannot occur.	268
UIE - User Interrupt Enable		269
UIF - User Interrupt Flush	Use the UIF instruction to remove selected pending interrupts from the system.	270

INT - Interrupt Subroutine



Instruction Type: input

Execution Time for the INT Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	0.5460 µs	0.5460 µs

The INT instruction is used as a label to identify a user interrupt service routine (ISR). This instruction is placed as the first instruction on a rung and is always evaluated as true. Use of the INT instruction is optional.

STS - Selectable Timed Start

Instruction Type: output

Execution Time for the STS Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	20.8470 µs	0.2125 µs

The STS instruction can be used to start and stop the STI function or to change the time interval between STI user interrupts. The STI instruction has one operand:

- **Time -** This is the amount of time (in milliseconds) which must expire prior to executing the selectable timed user interrupt. A value of zero disables the STI function. The time range is from 0...65,535 milliseconds.

The STS instruction applies the specified set point to the STI function as follows:

- If a zero set point is specified, the STI is disabled and STI:0/TIE is cleared (0).
- If the STI is disabled (not timing) and a value greater than 0 is entered into the set point, the STI starts timing to the new set point and STI:0/TIE is set (1).

- If the STI is currently timing and the set point is changed, the new setting takes effect immediately and the STI continues to time until it reaches the new set point.

  Note that if the new setting is less than the current accumulated time, the STI times-out immediately. For example, if the STI has been timing for 15 microseconds, and the STI set point is changed from 20 microseconds to 10 microseconds, an STI user interrupt occurs at the next start-of-rung.

Addressing Modes and File Types can be used as shown below:

Parameter	Data Files												Function Files								Address(1) Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Element	Bit	Word	Long Word
Time	•	•		•	•	•					•													•	•	•			•	

(1) See Important note about indirect addressing.

IMPORTANT You cannot use indirect addressing with: S, MG, PD, RTC, HSC, PTOX, PWMX, STI, EII, BHI, MMI, CS, IOS, and DLS files.

UID - User Interrupt Disable

UID
User Interrupt Disable
Interrupt Types 5

Instruction Type: output

Execution Time for the UID Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	2.7470 µs	0.1859 µs

The UID instruction is used to disable selected user interrupts. The table below shows the types of interrupts with their corresponding disable bits:

Types of Interrupts Disabled by the UID Instruction

Interrupt	Element	Decimal Value	Corresponding Bit
EII - Event Input Interrupt	Event 4	16384	bit 14
HSC - High-Speed Counter	HSC2	8192	bit 13
EII - Event Input Interrupt	Event 5	4096	bit 12
HSC - High-Speed Counter	HSC3	2048	bit 11
EII - Event Input Interrupt	Event 6	1024	bit 10
HSC - High-Speed Counter	HSC4	512	bit 9
EII - Event Input Interrupt	Event 7	256	bit 8
HSC - High-Speed Counter	HSC5	128	bit 7
EII - Event Input Interrupt	Event 0	64	bit 6
EII - Event Input Interrupt	Event 1	32	bit 5

Types of Interrupts Disabled by the UID Instruction

Interrupt	Element	Decimal Value	Corresponding Bit
HSC - High-Speed Counter	HSC0	16	bit 4
EII - Event Input Interrupt	Event 2	8	bit 3
EII - Event Input Interrupt	Event 3	4	bit 2
HSC - High-Speed Counter	HSC1	2	bit 1
STI - Selectable Timed Interrupt	STI	1	bit 0
Note: Bit 15 must be set to zero			

To disable interrupt(s):

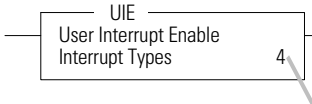
1. Select which interrupts you want to disable.
2. Find the Decimal Value for the interrupt(s) you selected.
3. Add the Decimal Values if you selected more than one type of interrupt.
4. Enter the sum into the UID instruction.

For example, to disable EII Event 1 and EII Event 3:

EII Event 1 = 32, EII Event 3 = 4

32 + 4 = 36 (enter this value)

UIE - User Interrupt Enable



Instruction Type: output

Execution Time for the UIE Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	3.4226 µs	0.1968 µs

The UIE instruction is used to enable selected user interrupts. The table below shows the types of interrupts with their corresponding enable bits:

Types of Interrupts Enabled by the UIE Instruction

Interrupt	Element	Decimal Value	Corresponding Bit
EII - Event Input Interrupt	Event 4	16384	bit 14
HSC - High-Speed Counter	HSC2	8192	bit 13
EII - Event Input Interrupt	Event 5	4096	bit 12
HSC - High-Speed Counter	HSC3	2048	bit 11
EII - Event Input Interrupt	Event 6	1024	bit 10
HSC - High-Speed Counter	HSC4	512	bit 9
EII - Event Input Interrupt	Event 7	256	bit 8

Types of Interrupts Enabled by the UIE Instruction

Interrupt	Element	Decimal Value	Corresponding Bit
HSC - High-Speed Counter	HSC5	128	bit 7
EII - Event Input Interrupt	Event 0	64	bit 6
EII - Event Input Interrupt	Event 1	32	bit 5
HSC - High-Speed Counter	HSC0	16	bit 4
EII - Event Input Interrupt	Event 2	8	bit 3
EII - Event Input Interrupt	Event 3	4	bit 2
HSC - High-Speed Counter	HSC1	2	bit 1
STI - Selectable Timed Interrupt	STI	1	bit 0
Note: Bit 15 must be set to zero			

To enable interrupt(s):

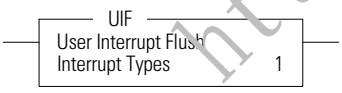
1. Select which interrupts you want to enable.
2. Find the Decimal Value for the interrupt(s) you selected.
3. Add the Decimal Values if you selected more than one type of interrupt.
4. Enter the sum into the UIE instruction.

For example, to enable EII Event 1 and EII Event 3:

EII Event 1 = 32, EII Event 3 = 4
32 + 4 = 36 (enter this value)

ATTENTION: If you enable interrupts during the program scan via an OTL, OTE, or UIE, this instruction *must* be the *last* instruction executed on the rung (last instruction on last branch). It is recommended this be the only output instruction on the rung.

UIF - User Interrupt Flush



Instruction Type: output

Execution Time for the UIF Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	2.7930 µs	0.1847 µs

The UIF instruction is used to flush (remove pending interrupts from the system) selected user interrupts. The table below shows the types of interrupts with their corresponding flush bits:

Types of Interrupts Disabled by the UIF Instruction

Interrupt	Element	Decimal Value	Corresponding Bit
EII - Event Input Interrupt	Event 4	16384	bit 14
HSC - High-Speed Counter	HSC2	8192	bit 13
EII - Event Input Interrupt	Event 5	4096	bit 12
HSC - High-Speed Counter	HSC3	2048	bit 11
EII - Event Input Interrupt	Event 6	1024	bit 10
HSC - High-Speed Counter	HSC4	512	bit 9
EII - Event Input Interrupt	Event 7	256	bit 8
HSC - High-Speed Counter	HSC5	128	bit 7
EII - Event Input Interrupt	Event 0	64	bit 6
EII - Event Input Interrupt	Event 1	32	bit 5
HSC - High-Speed Counter	HSC0	16	bit 4
EII - Event Input Interrupt	Event 2	8	bit 3
EII - Event Input Interrupt	Event 3	4	bit 2
HSC - High-Speed Counter	HSC1	2	bit 1
STI - Selectable Timed Interrupt	STI	1	bit 0
Note: Bit 15 must be set to zero			

To flush interrupt(s):

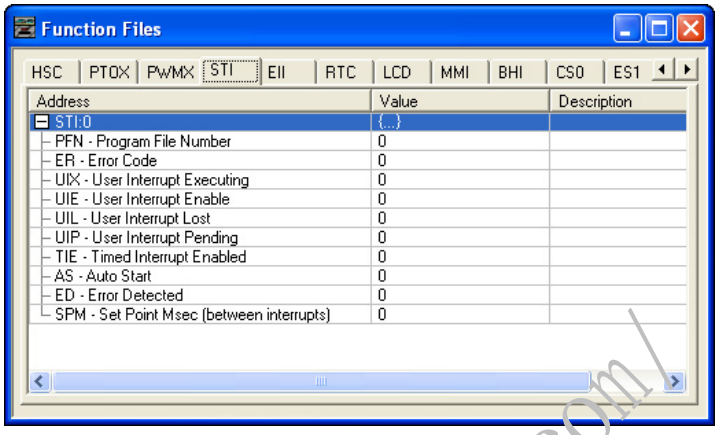
1. Select which interrupts you want to flush.
2. Find the Decimal Value for the interrupt(s) you selected.
3. Add the Decimal Values if you selected more than one type of interrupt.
4. Enter the sum into the UIF instruction.

For example, to disable EII Event 1 and EII Event 3:

EII Event 1 = 32, EII Event 3 = 4

32 + 4 = 36 (enter this value)

Using the Selectable Timed Interrupt (STI) Function File



The Selectable Timed Interrupt (STI) provides a mechanism to solve time critical control requirements. The STI is a trigger mechanism that allows you to scan or solve control program logic that is time sensitive.

Example of where you would use the STI are:

- PID type applications, where a calculation must be performed at a specific time interval.
- A motion application, where the motion instruction (PTO) needs to be scanned at a specific rate to guarantee a consistent acceleration/deceleration profile.
- A block of logic that needs to be scanned more often.

How an STI is used is typically driven by the demands/requirements of the application. It operates using the following sequence:

1. The user selects a time interval.
2. When a valid interval is set and the STI is properly configured, the controller monitors the STI value.
3. When the time period has elapsed, the controller's normal operation is interrupted.
4. The controller then scans the logic in the STI program file.
5. When the STI file scan is completed, the controller returns to where it was prior to the interrupt and continues normal operation.

Selectable Time Interrupt (STI) Function File Sub-Elements Summary

Selectable Timed Interrupt Function File (STI:0)

Sub-Element Description	Address	Data Format	Type	User Program Access	For More Information
PFN - Program File Number	STI:0.PFN	word (INT)	control	read only	273
ER - Error Code	STI:0.ER	word (INT)	status	read only	273
UIX - User Interrupt Executing	STI:0/UIX	binary (bit)	status	read only	274
UIE - User Interrupt Enable	STI:0/UIE	binary (bit)	control	read/write	274
UIL - User Interrupt Lost	STI:0/UIL	binary (bit)	status	read/write	274
UIP - User Interrupt Pending	STI:0/UIP	binary (bit)	status	read only	274
TIE - Timed Interrupt Enabled	STI:0/TIE	binary (bit)	control	read/write	275
AS - Auto Start	STI:0/AS	binary (bit)	control	read only	275
ED - Error Detected	STI:0/ED	binary (bit)	status	read only	275
SPM - Set Point Msec	STI:0.SPM	word (INT)	control	read/write	275

STI Function File Sub-Elements

STI Program File Number (PFN)

Sub-Element Description	Address	Data Format	Type	User Program Access
PFN - Program File Number	STI:0.PFN	word (INT)	control	read only

The PFN (Program File Number) variable defines which subroutine is called (executed) when the timed interrupt times out. A valid subroutine file is any program file (3...255).

The subroutine file identified in the PFN variable is not a special file within the controller; it is programmed and operates the same as any other program file. From the control program perspective it is unique, in that it is automatically scanned based on the STI set point.

STI Error Code (ER)

Sub-Element Description	Address	Data Format	Type	User Program Access
ER - Error Code	STI:0.ER	word (INT)	status	read only

Error codes detected by the STI sub-system are displayed in this register. The table below explains the error codes.

STI Error Code

Error Code	Recoverable Fault (Controller)	Description
1	Invalid Program File Number	Program file number is less than 3, greater than 255, or does not exist.

STI User Interrupt Executing (UIX)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIX - User Interrupt Executing	STI:0/UIX	binary (bit)	status	read only

The UIX (User Interrupt Executing) bit is set whenever the STI mechanism completes timing and the controller is scanning the STI PFN. The UIX bit is cleared when the controller completes processing the STI subroutine.

The STI UIX bit can be used in the control program as conditional logic to detect if an STI interrupt is executing.

STI User Interrupt Enable (UIE)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIE - User Interrupt Enable	STI:0/UIE	binary (bit)	control	read/write

The UIE (User Interrupt Enable) bit is used to enable or disable STI subroutine processing. This bit must be set if you want the controller to process the STI subroutine at the configured time interval.

If you need to restrict when the STI subroutine is processed, clear the UIE bit. An example of when this is important is if a series of math calculations need to be processed without interruption. Before the calculations take place, clear the UIE bit. After the calculations are complete, set the UIE bit and STI subroutine processing resumes.

STI User Interrupt Lost (UIL)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIL - User Interrupt Lost	STI:0/UIL	binary (bit)	status	read/write

The UIL (User Interrupt Lost) is a status flag that indicates an interrupt was lost. The controller can process 1 active and maintain up to 2 pending user interrupt conditions before it sets the lost bit.

This bit is set by the controller. It is up to the control program to utilize, track if necessary, and clear the lost condition.

STI User Interrupt Pending (UIP)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIP - User Interrupt Pending	STI:0/UIP	binary (bit)	status	read only

The UIP (User Interrupt Pending) is a status flag that represents an interrupt is pending. This status bit can be monitored or used for logic purposes in the control program if you need to determine when a subroutine cannot execute immediately.

This bit is automatically set and cleared by the controller. The controller can process 1 active and maintain up to 2 pending user interrupt conditions before it sets the lost bit.

STI Timed Interrupt Enabled (TIE)

Sub-Element Description	Address	Data Format	Type	User Program Access
TIE - Timed Interrupt Enabled	STI:0/TIE	binary (bit)	control	read/write

The TIE (Timed Interrupt Enabled) control bit is used to enable or disable the timed interrupt mechanism. When set (1), timing is enabled, when clear (0) timing is disabled. If this bit is cleared (disabled) while the timer is running, the accumulated value is cleared (0). If the bit is then set (1), timing starts.

This bit is controlled by the user program and retains its value through a power cycle.

STI Auto Start (AS)

Sub-Element Description	Address	Data Format	Type	User Program Access
AS - Auto Start	STI:0/AS	binary (bit)	control	read only

The AS (Auto Start) is a control bit that can be used in the control program. The auto start bit is configured with the programming device and stored as part of the user program. The auto start bit automatically sets the STI Timed Interrupt Enable (TIE) bit when the controller enters any executing mode.

STI Error Detected (ED)

Sub-Element Description	Address	Data Format	Type	User Program Access
ED - Error Detected	STI:0/ED	binary (bit)	status	read only

The ED (Error Detected) flag is a status bit that can be used by the control program to detect if an error is present in the STI sub-system. The most common type of error that this bit represents is a configuration error. When this bit is set, the user should look at the error code in parameter STI:0.ER

This bit is automatically set and cleared by the controller.

STI Set Point Milliseconds Between Interrupts (SPM)

Sub-Element Description	Address	Data Format	Range	Type	User Program Access
SPM - Set Point Msec	STI:0.SPM	word (INT)	0…65,535	control	read/write

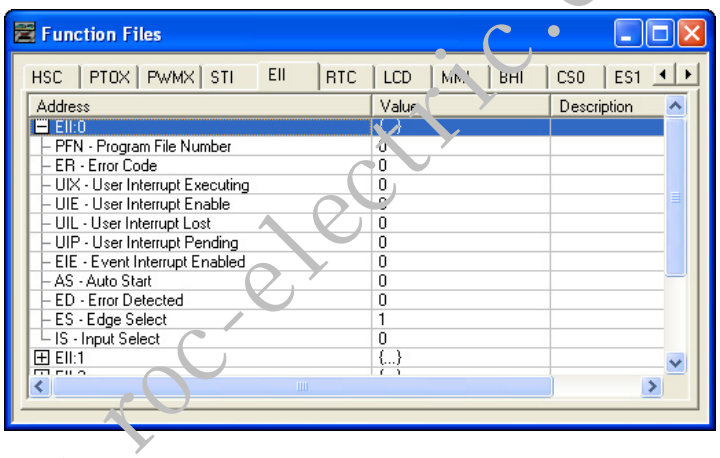
When the controller transitions to an executing mode, the SPM (set point in milliseconds) value is loaded into the STI. If the STI is configured correctly, and enabled, the program file identified in the STI variable PFN is scanned at this interval. This value can be changed from the control program by using the STS instruction.

TIP The minimum value cannot be less than the time required to scan the STI program file (STI:0.PFN) plus the Interrupt Latency.

Using the Event Input Interrupt (EII) Function File

The EII (event input interrupt) is a feature that allows the user to scan a specific program file (subroutine) when an input condition is detected from a field device.

Within the function file section of RSLogix 500/RSLogix Micro, the user sees an EII folder. Within the folder are eight EII elements. Each of these elements (EII:0, EII:1, EII:2, EII:3, EII:4, EII:5, EII6 and EII7) are identical; this explanation uses EII:0 as shown below.



Each EII can be configured to monitor any one of the first eight inputs (I1:0.0/0 to I1.0.0/7). Each EII can be configured to detect rising edge or falling edge input signals. When the configured input signal is detected at the input terminal, the controller immediately scans the configured subroutine.

Event Input Interrupt (EII) Function File Sub-Elements Summary

Event Input Interrupt Function File (EII:0)

Sub-Element Description	Address	Data Format	Type	User Program Access	For More Information
PFN - Program File Number	EII:0.PFN	word (INT)	control	read only	277
ER - Error Code	EII:0.ER	word (INT)	status	read only	277
UIX - User Interrupt Executing	EII:0/UIX	binary (bit)	status	read only	278
UIE - User Interrupt Enable	EII:0/UIE	binary (bit)	control	read/write	278
UIL - User Interrupt Lost	EII:0/UIL	binary (bit)	status	read/write	278
UIP - User Interrupt Pending	EII:0/UIP	binary (bit)	status	read only	278
EIE - Event Interrupt Enabled	EII:0/EIE	binary (bit)	control	read/write	279
AS - Auto Start	EII:0/AS	binary (bit)	control	read only	279

Event Input Interrupt Function File (EII:0)

Sub-Element Description	Address	Data Format	Type	User Program Access	For More Information
ED - Error Detected	EII:0/ED	binary (bit)	status	read only	279
ES - Edge Select	EII:0/ES	binary (bit)	control	read only	279
IS - Input Select	EII:0.IS	word (INT)	control	read only	280

EII Function File Sub-Elements

EII Program File Number (PFN)

Sub-Element Description	Address	Data Format	Type	User Program Access
PFN - Program File Number	EII:0.PFN	word (INT)	control	read only

PFN (Program File Number) defines which subroutine is called (executed) when the input terminal assigned to EII:0 detects a signal. A valid subroutine file is any program file (3...255).

The subroutine file identified in the PFN variable is not a special file within the controller. It is programmed and operated the same as any other program file. From the control program perspective it is unique, in that it is automatically scanned based on the configuration of the EII.

EII Error Code (ER)

Sub-Element Description	Address	Data Format	Type	User Program Access
ER - Error Code	EII:0.ER	word (INT)	status	read only

Any ER (Error Code) detected by the EII sub-system is displayed in this register. The table below explains the error codes.

EII Error Codes

Error Code	Recoverable Fault (Controller)	Description
1	Invalid Program File Number	Program file number is less than 3, greater than 255, or does not exist
2	Invalid Input Selection	Valid numbers must be 0, 1, 2, 3, 4, 5, 6, or 7.
3	Input Selection Overlap	EIIs cannot share inputs. Each EII must have a unique input.

EII User Interrupt Executing (UIX)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIX - User Interrupt Executing	EII:0/UIX	binary (bit)	status	read only

The UIX (User Interrupt Executing) bit is set whenever the EII mechanism detects a valid input and the controller is scanning the PFN. The EII mechanism clears the UIX bit when the controller completes its processing of the EII subroutine.

The EII UIX bit can be used in the control program as conditional logic to detect if an EII interrupt is executing.

EII User Interrupt Enable (UIE)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIE - User Interrupt Enable	EII:0/UIE	binary (bit)	control	read/write

The UIE (User Interrupt Enable) bit is used to enable or disable EII subroutine processing. This bit must be set if you want the controller to process the EII subroutine when an EII event occurs.

If you need to restrict when the EII subroutine is processed, clear the UIE bit. An example of when this is important is if a series of math calculations need to be processed without interruption. Before the calculations take place, clear the UIE bit. After the calculations are complete, set the UIE bit and EII subroutine processing resumes.

EII User Interrupt Lost (UIL)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIL - User Interrupt Lost	EII:0/UIL	binary (bit)	status	read/write

UIL (User Interrupt Lost) is a status flag that represents an interrupt has been lost. The controller can process 1 active and maintain up to 2 pending user interrupt conditions before it sets the lost bit.

This bit is set by the controller. It is up to the control program to utilize, track, and clear the lost condition.

EII User Interrupt Pending (UIP)

Sub-Element Description	Address	Data Format	Type	User Program Access
UIP - User Interrupt Pending	EII:0/UIP	binary (bit)	status	read only

UIP (User Interrupt Pending) is a status flag that represents an interrupt is pending. This status bit can be monitored, or used for logic purposes, in the control program if you need to determine when a subroutine cannot execute immediately.

This bit is automatically set and cleared by the controller. The controller can process 1 active and maintain up to 2 pending user interrupt conditions before it sets the pending bit.

EII Event Interrupt Enable (EIE)

Sub-Element Description	Address	Data Format	Type	User Program Access
EIE - Event Interrupt Enabled	EII:0/EIE	binary (bit)	control	read/write

EIE (Event Interrupt Enabled) allows the event interrupt function to be enabled or disabled from the control program. When set (1), the function is enabled, when cleared (0, default) the function is disabled.

This bit is controlled by the user program and retains its value through a power cycle.

EII Auto Start (AS)

Sub-Element Description	Address	Data Format	Type	User Program Access
AS - Auto Start	EII:0/AS	binary (bit)	control	read only

AS (Auto Start) is a control bit that can be used in the control program. The auto start bit is configured with the programming device and stored as part of the user program. The auto-start bit automatically sets the EII Event Interrupt Enable (EIE) bit when the controller enters any executing mode.

EII Error Detected (ED)

Sub-Element Description	Address	Data Format	Type	User Program Access
ED - Error Detected	EII:0/ED	binary (bit)	status	read only

The ED (Error Detected) flag is a status bit that can be used by the control program to detect if an error is present in the EII sub-system. The most common type of error that this bit represents is a configuration error. When this bit is set, look at the specific error code in parameter EII:0.ER

This bit is automatically set and cleared by the controller.

EII Edge Select (ES)

Sub-Element Description	Address	Data Format	Type	User Program Access
ES - Edge Select	EII:0/ES	binary (bit)	control	read only

The ES (Edge Select) bit selects the type of trigger that causes an Event Interrupt. This bit allows the EII to be configured for rising edge (off-to-on, 0-to-1) or falling edge (on-to-off, 1-to-0) signal detection. This selection is based on the type of field device that is connected to the controller.

The default condition is 1, which configures the EII for rising edge operation.

EII Input Select (IS)

Sub-Element Description	Address	Data Format	Type	User Program Access
IS - Input Select	EII:0.IS	word (INT)	control	read only

The IS (Input Select) parameter is used to configure each EII to a specific input on the controller. Valid inputs are 0...7, which correspond to I1:0.0/0...I1:0.0/7.

This parameter is configured with the programming device and cannot be changed from the control program.

Process Control Instruction

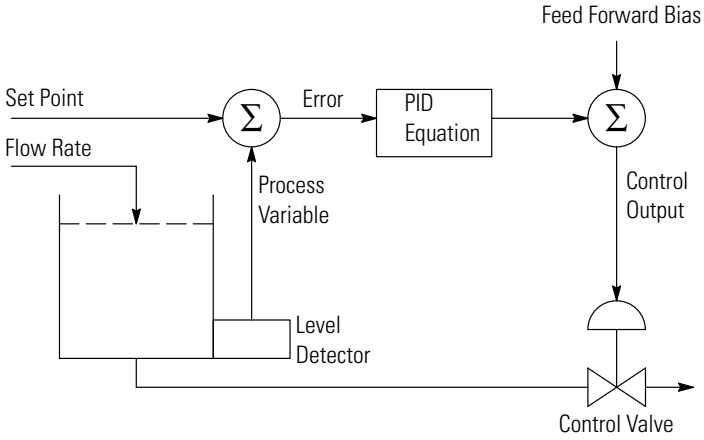
This chapter describes the MicroLogix 1400 Proportional Integral Derivative (PID) instruction. The PID instruction is an output instruction that controls physical properties such as temperature, pressure, liquid level, or flow rate using process loops.

The PID Concept

The PID instruction normally controls a closed loop using inputs from an analog input module and providing an output to an analog output module. For temperature control, you can convert the analog output to a time proportioning on/off output for driving a heater or cooling unit. An example appears on page 298.

The PID instruction can be operated in the timed mode or the Selectable Time Interrupt (STI mode). In the timed mode, the instruction updates its output periodically at a user-selectable rate. In the STI mode, the instruction should be placed in an STI interrupt subroutine. It then updates its output every time the STI subroutine is scanned. The STI time interval and the PID loop update rate must be the same in order for the equation to execute properly. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information on STI interrupts.

PID closed loop control holds a process variable at a desired set point. A flow rate/fluid level example is shown below.



The PID equation controls the process by sending an output signal to the control valve. The greater the error between the setpoint and process variable input, the greater the output signal. Alternately, the smaller the error, the smaller the output signal. An additional value (feed forward or bias) can be added to the control

output as an offset. The PID result (control variable) drives the process variable toward the set point.

The PID Equation

The PID instruction uses the following algorithm:

Standard equation with dependent gains:

$$Output = K_C\left[(E) + \frac{1}{T_I}\int(E)dt + T_D \cdot \frac{d(PV)}{dt}\right] + bias$$

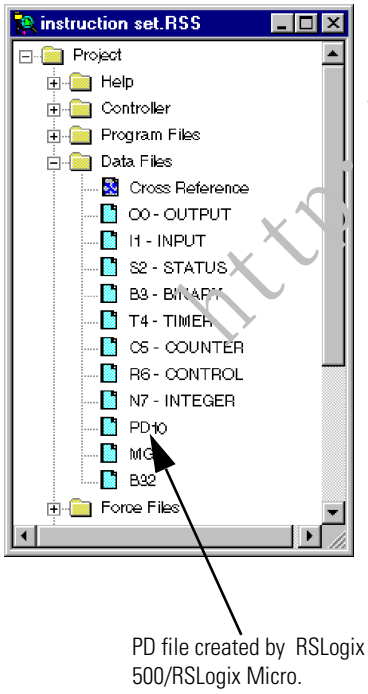
Standard Gains constants are:

Term	Range (Low to High)	Reference
Controller Gain K_C	0.01…327.67 (dimensionless)(1)	Proportional
Reset Term $1/T_I$	327.67…0.01 (minutes per repeat)(1)	Integral
Rate Term T_D	0.01…327.67 (minutes)(1)	Derivative

(1) Applies to MicroLogix 1400 PID range when Reset and Gain Range (RG) bit is set to 1. For more information on reset and gain, see PLC 5 Gain Range (RG) on page 293.

The derivative term (rate) provides smoothing by means of a low-pass filter. The cut-off frequency of the filter is 16 times greater than the corner frequency of the derivative term.

PD Data File



PD file created by RSLogix 500/RSLogix Micro.

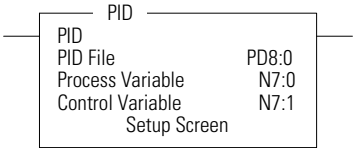
The PID instruction implemented by the MicroLogix 1400 controllers is virtually identical in function to the PID implementation used by the Allen-Bradley SLC 5/03 and higher processors. Minor differences primarily involve enhancements to terminology. The major difference is that the PID instruction now has its own data file. In the SLC family of processors, the PID instruction operated as a block of registers within an integer file. The Micrologix 1400 PID instruction utilizes a PD data file.

You can create a PD data file by creating a new data file and classifying it as a PD file type. RSLogix automatically creates a new PD file or a PD sub-element whenever a PID instruction is programmed on a rung. The PD file then appears in the list of Data Files as shown in the illustration.

Each PD data file has a maximum of 255 elements and each PID instruction requires a unique PD element. Each PD element is composed of 20 sub-elements,

which include bit, integer and long integer data. All of the examples in this chapter use PD file 10 sub-element 0.

PID - Proportional Integral Derivative



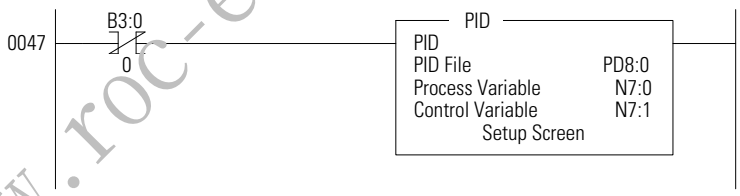
Instruction Type: output

Controller	When Rung Is:	
	True	False
MicroLogix 1400	7.1750 µs	7.0910 µs

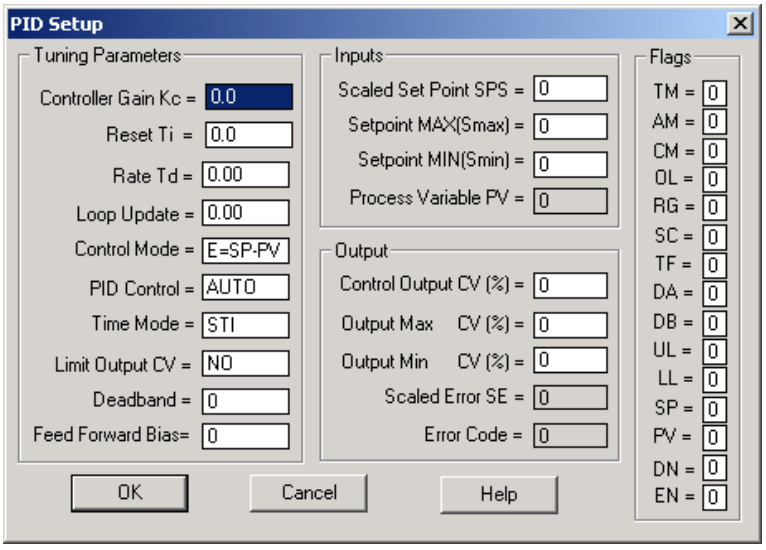
It is recommended that you place the PID instruction on a rung without any conditional logic. If conditional logic exists, the Control Variable output remains at its last value, and the CVP CV% term and integral term are both cleared when the rung is false.

TIP In order to stop and restart the PID instruction, you need to create a false-to-true rung transition.

The example below shows a PID instruction on a rung with RSLogix 500/ RSLogix Micro programming software.



When programming, the setup screen provides access to the PID instruction configuration parameters. The illustration below shows the RSLogix 500/ RSLogix Micro setup screen.



Input Parameters

The table below shows the input parameter addresses, data formats, and types of user program access. See the indicated pages for descriptions of each parameter.

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access	For More Information
SPS - Setpoint	PD10:0.SPS	word (INT)	0…16383[1]	control	read/write	284
PV - Process Variable	user defined	word (INT)	0…16383	control	read/write	284
MAXS - Setpoint Maximum	PD10:0.MAXS	word (INT)	-32,768…+32,767	control	read/write	285
MINS - Setpoint Minimum	PD10:0.MINS	word (INT)	-32,768…+32,767	control	read/write	285
OSP - Old Setpoint Value	PD10:0.OSP	word (INT)	-32,768…+32,767	status	read only	285
OL - Output Limit	PD10:0/OL	binary	1 = enabled 0 = disabled	control	read/write	285
CVH - Control Variable High Limit	PD10:0.CVH	word (INT)	0…100%	control	read/write	286
CVL - Control Variable Low Limit	PD10:0.CVL	word (INT)	0…100%	control	read/write	286

(1) The range listed in the table is for when scaling is not enabled. With scaling, the range is from minimum scaled (MINS) to maximum scaled (MAXS).

Setpoint (SPS)

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
SPS - Setpoint	PD10:0.SPS	word (INT)	0…16383[1]	control	read/write

(1) The range listed in the table is for when scaling is not enabled. With scaling, the range is from minimum scaled (MINS) to maximum scaled (MAXS).

The SPS (Setpoint) is the desired control point of the process variable.

Process Variable (PV)

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
PV - Process Variable	user defined	word (INT)	0…16383	control	read/write

The PV (Process Variable) is the analog input variable.

Setpoint MAX (MAXS)

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
MAXS - Setpoint Maximum	PD10:0.MAXS	word (INT)	-32,768…+32,767	control	read/write

If the SPV is read in engineering units, then the MAXS (Setpoint Maximum) parameter corresponds to the value of the setpoint in engineering units when the control input is at its maximum value.

Setpoint MIN (MINS)

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
MINS - Setpoint Minimum	PD10:0.MINS	word (INT)	-32,768…+32,767	control	read/write

If the SPV is read in engineering units, then the MINS (Setpoint Minimum) parameter corresponds to the value of the setpoint in engineering units when the control input is at its minimum value.

TIP *MinS - MaxS* scaling allows you to work in engineering units. The deadband, error, and SPV are also displayed in engineering units. The process variable, PV, must be within the range of 0…16383. Use of *MinS - MaxS* does not minimize PID PV resolution.

Scaled errors greater than +32767 or less than -32768 cannot be represented. If the scaled error is greater than +32767, it is represented as +32767. If the scaled error is less than -32768, it is represented as -32768.

Old Setpoint Value (OSP)

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
OSP - Old Setpoint Value	PD10:0.OSP	word (INT)	-32,768…+32,767	status	read only

The OSP (Old Setpoint Value) is substituted for the current setpoint, if the current setpoint goes out of range of the setpoint scaling (limiting) parameters.

Output Limit (OL)

Output Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
OL - Output Limit	PD10:0/OL	binary	1 = enabled 0 = disabled	control	read/write

An enabled (1) value enables output limiting to the values defined in PD10:0.CVH (Control Variable High) and PD10.0.CVL (Control Variable Low).

A disabled (0) value disables OL (Output Limiting).

Control Variable High Limit (CVH)

Output Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
CVH - Control Variable High Limit	PD10:0.CVH	word (INT)	0…100%	control	read/write

When the output limit bit (PD10:0/OL) is enabled (1), the CVH (Control Value High) you enter is the maximum output (in percent) that the control variable attains. If the calculated CV exceeds the CVH, the CV is set (overridden) to the CVH value you entered and the upper limit alarm bit (UL) is set.

When the output limit bit (PD10:0/OL) is disabled (0), the CVH value you enter determines when the upper limit alarm bit (UL) is set.

If CV exceeds the maximum value, the output is not overridden and the upper limit alarm bit (UL) is set.

Control Variable Low Limit (CVL)

Output Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
CVL - Control Variable Low Limit	PD10:0.CVL	word (INT)	0…100%	control	read/write

When the output limit bit (PD10:0/OL) is enabled (1), the CVL (Control Value Low) you enter is the minimum output (in percent) that the Control Variable attains. If the calculated CV is below the minimum value, the CV is set (overridden) to the CVL value you entered and the lower limit alarm bit (LL) is set.

When the output limit bit (PD10:0/OL) is disabled (0), the CVL value you enter determines when the lower limit alarm bit (LL) is set. If CV is below the minimum value, the output is not overridden and the lower limit alarm bit (LL) is set.

Output Parameters

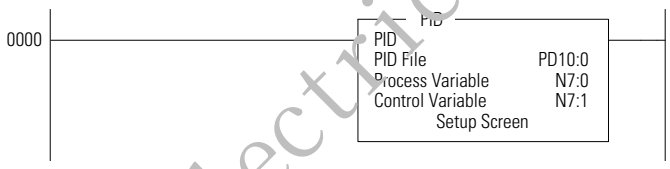
The table below shows the output parameter addresses, data formats, and types of user program access. See the indicated pages for descriptions of each parameter.

Output Parameter Descriptions	Address	Data Format	Range	Type	User Program Access	For More Information
CV - Control Variable	User-defined	word (INT)	0…16,383	control	read/write	287
CVP - Control Variable Percent	PD10:0.CVP	word (INT)	0…100	control	read/write	287
SPV - Scaled Process Variable	PD10:0.SPV	word (INT)	0…16383	status	read only	288

Control Variable (CV)

Output Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
CV - Control Variable	User-defined	word (INT)	0…16,383	control	read/write

The CV (Control Variable) is user-defined. See the ladder rung below.



Control Variable Percent (CVP)

Output Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
CVP - Control Variable Percent	PD10:0.CVP	word (INT)	0…100	control	status read

CVP (Control Variable Percent) displays the control variable as a percentage. The range is 0...100%.

If the PD10:0/AM bit is off (automatic mode), CVP tracks the control variable (CV) output being calculated by the PID equation.

If the PD10:0/AM bit is on (manual mode), CVP tracks the value that can be manipulated in the Control Variable (CV) data word.

The only way for a programmer to have control of the PID CV is to place the PID instruction in manual mode and write to the CV word via the control program or programming software. If no change is made to CV while in manual mode, the CVP will display the last value calculated by the PID equation.

Scaled Process Variable (SPV)

Input Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
SPV - Scaled Process Variable	PD10:0.SPV	word (INT)	0…16383	status	read only

The SPV (Scaled Process Variable) is the analog input variable. If scaling is enabled, the range is the minimum scaled value (MinS) to maximum scaled value (MaxS).

If the SPV is configured to be read in engineering units, then this parameter corresponds to the value of the process variable in engineering units. See Analog I/O Scaling on page 298 for more information on scaling.

Tuning Parameters

The table below shows the tuning parameter addresses, data formats, and types of user program access. See the indicated pages for descriptions of each parameter.

Tuning Parameter Descriptions

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access	For More Information
KC - Controller Gain - K_c	PD10:0.KC	word (INT)	0…32,767	control	read/write	289
TI - Reset Term - T_i	PD10:0.Ti	word (INT)	0…32,767	control	read/write	289
TD - Rate Term - T_d	PD 10:0.TD	word (INT)	0…32,767	control	read/write	290
TM - Time Mode	PD10:0.TM	binary	0 or 1	control	read/write	291
LUT - Loop Update Time	PD10:0.LUT	word (INT)	1…1024	control	read/write	291
ZCD - Zero Crossing Deadband	PD10:0.ZCD	word (INT)	0…32,767	control	read/write	292
FF - Feed Forward Bias	PD10:0.FF	word (INT)	-16,383…+16,383	control	read/write	292
SE - Scaled Error	PD10:0.SE	word (INT)	-32,768…+32,767	status	read only	292
AM - Automatic/Manual	PD10:0/AM	binary (bit)	0 or 1	control	read/write	292
CM - Control Mode	PD10:0/CM	binary (bit)	0 or 1	control	read/write	293
DB - PV in Deadband	PD10:0/DB	binary (bit)	0 or 1	status	read/write	293
RG - PLC 5 Gain Range	PD10:0/RG	binary (bit)	0 or 1	control	read/write	293
SC - Setpoint Scaling	PD10:0/SC	binary (bit)	0 or 1	control	read/write	294

Tuning Parameter Descriptions

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access	For More Information
TF - Loop Update Too Fast	PD10:0/TF	binary (bit)	0 or 1	status	read/write	294
DA - Derivative Action Bit	PD10:0/DA	binary (bit)	0 or 1	control	read/write	294
UL - CV Upper Limit Alarm	PD10:0/UL	binary (bit)	0 or 1	status	read/write	294
LL - CV Lower Limit Alarm	PD10:0/LL	binary (bit)	0 or 1	status	read/write	295
SP - Setpoint Out of Range	PD10:0/SP	binary (bit)	0 or 1	status	read/write	295
PV - PV Out of Range	PD10:0/PV	binary (bit)	0 or 1	status	read/write	295
DN - Done	PD10:0/DN	binary (bit)	0 or 1	status	read only	295
EN - Enable	PD10:0/EN	binary (bit)	0 or 1	status	read only	295
IS - Integral Sum	PD10:0.IS	long word (32-bit INT)	-2,147,483,648…2,147, 483,647	status	read/write	296
AD - Altered Derivative Term	PD10:0.19	long word (32-bit INT)	-2,147,483,648…2,147, 483,647	status	read only	296

Controller Gain (K_c)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
KC - Controller Gain - K_c	PD10:0.KC	word (INT)	0…32,767	control	read/write

Gain K_c (word 3) is the proportional gain, ranging from 0...3276.7 (when RG = 0), or 0...327.67 (when RG = 1). Set this gain to one-half the value needed to cause the output to oscillate when the reset and rate terms (below) are set to zero.

TIP Controller gain is affected by the reset and gain range (RG) bit. For information, see PLC 5 Gain Range (RG) on page 293.

Reset Term (T_i)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
TI - Reset Term - T_i	PD10:0.Ti	word (INT)	0…32,767	control	read/write

Reset T_i (word 4) is the Integral gain, ranging from 0...3276.7 (when RG = 0), or 327.67 (when RG = 1) minutes per repeat. Set the reset time equal to the natural period measured in the above gain calibration. A value of 1 adds the maximum integral term into the PID equation.

TIP Reset term is affected by the reset and gain range (RG) bit. For information, see PLC 5 Gain Range (RG) on page 293.

Rate Term (T_d)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
TD - Rate Term - T_d	PD 10:0.TD	word (INT)	0…32,767	control	read/write

Rate T_d (word 5) is the Derivative term. The adjustment range is 0 to 327.67 minutes. Set this value to 1/8 of the integral gain T_i.

TIP This word is not effected by the reset and gain range (RG) bit. For information, see PLC 5 Gain Range (RG) on page 293.

Time Mode (TM)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
TM - Time Mode	PD10:0.TM	binary	0 or 1	control	read/write

The time mode bit specifies when the PID is in timed mode (1) or STI mode (0). This bit can be set or cleared by instructions in your ladder program.

When set for timed mode, the PID updates the CV at the rate specified in the loop update parameter (PD10:0.LUT).

When set for STI mode, the PID updates the CV every time the PID instruction is scanned in the control program. When you select STI, program the PID instruction in the STI interrupt subroutine. The STI routine should have a time interval equal to the setting of the PID "loop update" parameter (PD10:0.LUT). Set the STI period in word STI:0.SPM. For example, if the loop update time contains the value 10 (for 100 ms), then the STI time interval must also equal 100 (for 100 ms).

TIP When using timed mode, your processor scan time should be at least ten times faster than the loop update time to prevent timing inaccuracies or disturbances.

Loop Update Time (LUT)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
LUT - Loop Update Time	PD10:0.LUT	word (INT)	1…1024	control	read/write

The loop update time (word 13) is the time interval between PID calculations. The entry is in 0.01 second intervals. Enter a loop update time five to ten times faster than the natural period of the load. The natural period of the load is determined by setting the reset and rate parameters to zero and then increasing the gain until the output begins to oscillate. When in STI mode, this value must equal the STI time interval value loaded in STI:0.SPM. The valid range is 0.01...10.24 seconds.

Zero Crossing Deadband (ZCD)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
ZCD - Zero Crossing Deadband	PD10:0.ZCD	word (INT)	0…32,767	control	read/write

The deadband extends above and below the setpoint by the value entered. The deadband is entered at the zero crossing of the process variable and the setpoint. This means that the deadband is in effect only after the process variable enters the deadband *and* passes through the setpoint.

The valid range is 0 to the scaled maximum, or 0...16,383 when no scaling exists.

Feed Forward Bias (FF)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
FF - Feed Forward Bias	PD10:0.FF	word (INT)	-16,383…+16,383	control	read/write

The feed forward bias is used to compensate for disturbances that may affect the CV output.

Scaled Error (SE)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
SE - Scaled Error	PD10:0.SE	word (INT)	-32,768…+32,767	status	read only

Scaled error is the difference between the process variable and the setpoint. The format of the difference (E = SP-PV or E = PV-SP) is determined by the control mode (CM) bit. See Control Mode (CM) on page 293.

Automatic / Manual (AM)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
AM - Automatic/Manual	PD10:0/AM	binary (bit)	0 or 1	control	read/write

The auto/manual bit can be set or cleared by instructions in your ladder program. When off (0), it specifies automatic operation. When on (1), it specifies manual operation. In automatic operation, the instruction controls the control variable (CV). In manual operation, the user/control program controls the CV. During tuning, set this bit to manual.

TIP Output limiting is also applied when in manual.

Control Mode (CM)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
CM - Control Mode	PD10:0/CM	binary (bit)	0 or 1	control	read/write

Control mode, or forward-/reverse-acting, toggles the values E=SP-PV and E=PV-SP.

Forward acting (E=PV-SP) causes the control variable to increase when the process variable is greater than the setpoint.

Reverse acting (E=SP-PV) causes the control variable to decrease when the process variable is greater than the setpoint.

PV in Deadband (DB)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
DB - PV in Deadband	PD10:0/DB	binary (bit)	0 or 1	status	read/write

This bit is set (1) when the process variable is within the zero-crossing deadband range.

PLC 5 Gain Range (RG)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
RG - PLC 5 Gain Range	PD10:0/RG	binary (bit)	0 or 1	control	read/write

When set (1), the reset (TI) and gain range enhancement bit (RG) causes the reset minute/repeat value and the gain multiplier (KC) to be divided by a factor of 10. That means a reset multiplier of 0.01 and a gain multiplier of 0.01.

When clear (0), this bit allows the reset minutes/repeat value and the gain multiplier value to be evaluated with a reset multiplier of 0.1 and a gain multiplier of 0.1.

Example with the RG bit set: The reset term (TI) of 1 indicates that the integral value of 0.01 minutes/repeat (0.6 seconds/repeat) is applied to the PID integral algorithm. The gain value (KC) of 1 indicates that the error is multiplied by 0.01 and applied to the PID algorithm.

Example with the RG bit clear: The reset term (TI) of 1 indicates that the integral value of 0.1 minutes/repeat (6.0 seconds/repeat) is applied to the PID integral algorithm. The gain value (KC) of 1 indicates that the error is multiplied by 0.1 and applied to the PID algorithm.

TIP The rate multiplier (TD) is not affected by this selection.

Setpoint Scaling (SC)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
SC - Setpoint Scaling	PD10:0/SC	binary (bit)	0 or 1	control	read/write

The SC bit is cleared when setpoint scaling values are specified.

Loop Update Too Fast (TF)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
TF - Loop Update Too Fast	PD10:0/TF	binary (bit)	0 or 1	status	read/write

The TF bit is set by the PID algorithm if the loop update time specified cannot be achieved by the controller due to scan time limitations.

If this bit is set, correct the problem by updating your PID loop at a slower rate or move the PID instruction to an STI interrupt routine. Reset and rate gains will be in error if the instruction operates with this bit set.

Derivative Action Bit (DA)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
DA - Derivative Action Bit	PD10:0/DA	binary (bit)	0 or 1	control	read/write

When set (1), the derivative (rate) action (DA) bit causes the derivative (rate) calculation to be evaluated on the error instead of the process variable (PV). When clear (0), this bit allows the derivative (rate) calculation to be evaluated where the derivative is performed on the PV.

CV Upper Limit Alarm (UL)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
UL - CV Upper Limit Alarm	PD10:0/UL	binary (bit)	0 or 1	status	read/write

The control variable upper limit alarm bit is set when the calculated CV output exceeds the upper CV limit.

CV Lower Limit Alarm (LL)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
LL - CV Lower Limit Alarm	PD10:0/LL	binary (bit)	0 or 1	status	read/write

The control variable lower limit alarm bit is set (1) when the calculated CV output is less than the lower CV limit.

Setpoint Out Of Range (SP)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
SP - Setpoint Out of Range	PD10:0/SP	binary (bit)	0 or 1	status	read/write

This bit is set (1) when the setpoint:

- exceeds the maximum scaled value, or
- is less than the minimum scaled value.

PV Out Of Range (PV)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
PV - PV Out of Range	PD10:0/PV	binary (bit)	0 or 1	status	read/write

The process variable out of range bit is set (1) when the unscaled process variable

- exceeds 16,383, or
- is less than zero.

Done (DN)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
DN - Done	PD10:0/DN	binary (bit)	0 or 1	status	read only

The PID done bit is set (1) for one scan when the PID algorithm is computed. It resets (0) whenever the instruction is scanned and the PID algorithm was not computed (applies to timed mode only).

PD10:0.19Enable (EN)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
EN - Enable	PD10:0/EN	binary (bit)	0 or 1	status	read only

The PID enabled bit is set (1) whenever the PID instruction is enabled. It follows the rung state.

Integral Sum (IS)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
IS - Integral Sum	PD10:0.IS	long word (32-bit INT)	-2,147,483,648…2,147,483,647	status	read/write

This is the result of the integration $\frac{K_C}{T_I}\int E(dt)$.

Altered Derivative Term (AD)

Tuning Parameter Descriptions	Address	Data Format	Range	Type	User Program Access
AD - Altered Derivative Term	PD10:0.19	long word (32-bit INT)	-2,147,483,648…2,147,483,647	status	read only

This long word is used internally to track the change in the process variable within the loop update time.

Runtime Errors

Error code 0036 appears in the status file when a PID instruction runtime error occurs. Code 0036 covers the following PID error conditions, each of which has been assigned a unique single byte code value that appears in the MSB of the second word of the control block. The error code is also displayed on the PID Setup Screen in RSLogix 500/RSLogix Micro.

Runtime Error Codes

Error Code	Description of Error Condition or Conditions		Corrective Action	
11H	1. Loop update time $D_t > 1024$		Change loop update time $0 < D_t \leq 1024$	
	2. Loop update time $D_t = 0$			
12H	Proportional gain $K_c < 0$		Change proportional gain K_c to $0 < K_c$	
13H	Integral gain (reset) $T_i < 0$		Change integral gain (reset) T_i to $0 \leq T_i$	
14H	Derivative gain (rate) $T_d < 0$		Change derivative gain (rate) T_d to $0 \leq T_d$	
15H	Feed Forward Bias (FF) is out-of-range.		Change FF so it is within the range -16383…+16383.	
23H	Scaled setpoint min MinS > Scaled setpoint max MaxS		Change scaled setpoint min MinS to $-32768 \leq MinS \leq MaxS \leq +32767$	
31H	If you are using setpoint scaling and MinS > setpoint SP > MaxS, or If you are not using setpoint scaling and 0 > setpoint SP > 16383, then during the initial execution of the PID loop, this error occurs and bit 11 of word 0 of the control block is set. However, during subsequent execution of the PID loop if an invalid loop setpoint is entered, the PID loop continues to execute using the old setpoint, and bit 11 of word 0 of the control block is set.		If you are using setpoint scaling, then change the setpoint SP to $MinS \leq SP \leq MaxS$, or If you are not using setpoint scaling, then change the setpoint SP to $0 \leq SP \leq 16383$.	
41H	**Scaling Selected**	**Scaling Deselected**	**Scaling Selected**	**Scaling Deselected**
	1. Deadband < 0, or	1. Deadband < 0, or	Change deadband to $0 \leq$ deadband $\leq$ $(MaxS - MinS) \leq 16383$	Change deadband to $0 \leq$ deadband ≤ 16383
	2. Deadband > (MaxS – MinS)	2. Deadband > 16383		
51H	1. Output high limit < 0, or 2. Output high limit > 100		Change output high limit to $0 \leq$ output high limit ≤ 100	
52H	1. Output low limit < 0, or 2. Output low limit > 100		Change output low limit to $0 \leq$ output low limit $\leq$ output high limit ≤ 100	
53H	Output low limit > output high limit		Change output low limit to $0 \leq$ output low limit $\leq$ output high limit ≤ 100	

Analog I/O Scaling

To configure an analog input for use in a PID instruction, the analog data must be scaled to match the PID instruction parameters. In the MicroLogix 1400, the process variable (PV) in the PID instruction is designed to work with a data range of 0...16,383. The 1762 expansion I/O analog modules ((1762-IF4, 1762-OF4, 1762-IF2OF2, 1762-IT4 and 1762-IR4)) are capable of on-board scaling. Scaling data is required to match the range of the analog input to the input range of the PID instruction. The ability to perform scaling in the I/O modules reduces the amount of programming required in the system and makes PID setup much easier.

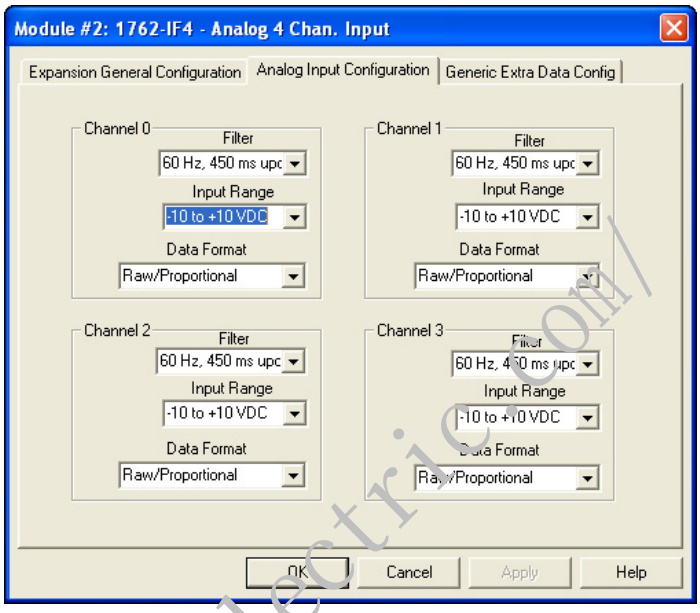
The example shows a 1762-IF4 module. The IF4 has 4 inputs, which are individually configurable. In this example, analog input 0 is configured for -10...10V and is scaled in engineering units. Channel 0 is not being used in a PID instruction. Input 1 (channel 1) is configured for 4. .20 mA operation with scaling configured for a PID instruction. This configures the analog data for the PID instruction.

PID Instruction Analog Data

Field Device Input Signal	Analog Register Scaled Data
> 20.0 mA	16,384…17,406
20.0 mA	16,383
4.0 mA	0
< 4.0 mA	-819…-1

The analog configuration screen is accessed from within RSLogix 500/RSLogix Micro. Double click the I/O configuration item in the Controller folder, and then double click on the specific I/O module.

The configuration for the analog output is virtually identical. Simply address the PID control variable (CV) to the analog output address and configure the analog output to "Scaled for PID" behavior.



Application Notes

The following paragraphs discuss:

- Input/Output Ranges
- Scaling to Engineering Units
- Zero-crossing Deadband
- Output Alarms
- Output Limiting with Anti-reset Windup
- The Manual Mode
- Feed Forward

ATTENTION: Do not alter the state of any PID control block value unless you fully understand its function and how it will affect your process. Unexpected operation could result with possible equipment damage and/or personal injury.

Input/Output Ranges

The input module measuring the process variable (PV) must have a full scale binary range of 0...16383. If this value is less than 0 (bit 15 set), then a value of zero is used for PV and the "Process var out of range" bit is set (bit 12 of word 0 in the control block). If the process variable is greater than 16383 (bit 14 set), then a value of 16383 is used for PV and the "Process var out of range" bit is set.

The Control Variable, calculated by the PID instruction, has the same range of 0...16383. The Control Output (word 16 of the control block) has the range of 0...100%. You can set lower and upper limits for the instruction's calculated output values (where an upper limit of 100% corresponds to a Control Variable limit of 16383).

Scaling to Engineering Units

Scaling lets you enter the setpoint and zero-crossing deadband values in engineering units, and display the process variable and error values in the same engineering units. Remember, the process variable PV must still be within the range 0...16383. The PV is displayed in engineering units, however.

Select scaling as follows:

1. Enter the maximum and minimum scaling values MaxS and MinS in the PID control block. The MinS value corresponds to an analog value of zero for the lowest reading of the process variable. MaxS corresponds to an analog value of 16383 for the highest reading. These values reflect the process limits. Setpoint scaling is selected by entering a non-zero value for one or both parameters. If you enter the same value for both parameters, setpoint scaling is disabled.

   For example, if measuring a full scale temperature range of -73°C (PV=0)...+1156°C (PV=16383), enter a value of -73 for MinS and 1156 for MaxS. Remember that inputs to the PID instruction must be 0...16383. Signal conversions could be as follows:

Signal Conversion example

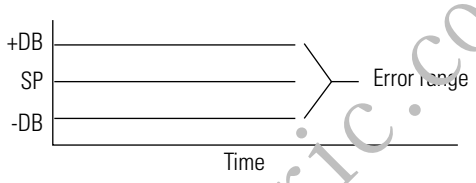
Example Values	
Process limits	-73…+1156°C
Transmitter output (if used)	+4…+20 mA
Output of analog input module	0…16383
PID instruction, MinS to MaxS	-73…+1156°C

2. Enter the setpoint (word 2) and deadband (word 9) in the same scaled engineering units. Read the scaled process variable and scaled error in these units as well. The control output percentage (word 16) is displayed as a percentage of the 0...16383 CV range. The actual value transferred to the CV output is always between 0 and 16383.

When you select scaling, the instruction scales the setpoint, deadband, process variable, and error. You must consider the effect on all these variables when you change scaling.

Zero-Crossing Deadband DB

The adjustable deadband lets you select an error range above and below the setpoint where the output does not change as long as the error remains within this range. This lets you control how closely the process variable matches the setpoint without changing the output.



Zero-crossing is deadband control that lets the instruction use the error for computational purposes as the process variable crosses into the deadband until it crosses the setpoint. Once it crosses the setpoint (error crosses zero and changes sign) and as long as it remains in the deadband, the instruction considers the error value zero for computational purposes.

Select deadband by entering a value in the deadband storage word (word 9) in the control block. The deadband extends above and below the setpoint by the value you enter. A value of zero inhibits this feature. The deadband has the same scaled units as the setpoint if you choose scaling.

Output Alarms

You may set an output alarm on the control variable at a selected value above and/or below a selected output percent. When the instruction detects that the control variable has exceeded either value, it sets an alarm bit (bit LL for lower limit, bit UL for upper limit) in the PID instruction. Alarm bits are reset by the instruction when the control variable comes back inside the limits. The instruction does not prevent the control variable from exceeding the alarm values unless you select output limiting.

Select upper and lower output alarms by entering a value for the upper alarm (CVH) and lower alarm (CVL). Alarm values are specified as a percentage of the output. If you do not want alarms, enter zero and 100% respectively for lower and upper alarm values and ignore the alarm bits.

Output Limiting with Anti-Reset Windup

You may set an output limit (percent of output) on the control variable. When the instruction detects that the control variable has exceeded a limit, it sets an alarm bit (bit LL for lower limit, bit UL for upper limit), and prevents the control variable from exceeding either limit value. The instruction limits the control variable to 0 and 100% if you choose not to limit.

Select upper and lower output limits by setting the limit enable bit (bit OL), and entering an upper limit (CVH) and lower limit (CVL). Limit values are a percentage (0...100%) of the control variable.

The difference between selecting output alarms and output limits is that you must select output limiting to enable limiting. Limit and alarm values are stored in the same words. Entering these values enables the alarms, but not limiting. Entering these values and setting the limit enable bit enables limiting and alarms.

Anti-reset windup is a feature that prevents the integral term from becoming excessive when the control variable reaches a limit. When the sum of the PID and bias terms in the control variable reaches the limit, the instruction stops calculating the integral sum until the control variable comes back in range. The integral sum is contained in element, IS.

The Manual Mode

In the MANUAL mode, the PID algorithm does not compute the value of the control variable. Rather, it uses the value as an input to adjust the integral sum (IS) so that a smooth transfer takes place upon re-entering the AUTO mode.

In the MANUAL mode, the programmer allows you to enter a new CV value from 0...100%. This value is converted into a number from 0...16383 and written to the Control Variable address. If your ladder program sets the manual output level, design your ladder program to write to the CV address when in the MANUAL mode. Remember that the new CV value is in the range of 0...16383, not 0...100. Writing to the CV percent (CVP) with your ladder program has no effect in the MANUAL mode.

PID Rung State

If the PID rung is false, the integral sum (IS) is cleared and CV remains in its last state.

Feed Forward or Bias

Applications involving transport lags may require that a bias be added to the CV output in anticipation of a disturbance. This bias can be accomplished using the processor by writing a value to the Feed Forward Bias element (word FF). (See page 292.) The value you write is added to the output, allowing a feed forward action to take place. You may add a bias by writing a value between -16383 and +16383 to word 6 with your programming terminal or ladder program.

Application Examples

PID Tuning

PID tuning requires a knowledge of process control. If you are inexperienced, it will be helpful if you obtain training on the process control theory and methods used by your company.

There are a number of techniques that can be used to tune a PID loop. The following PID tuning method is general and limited in terms of handling load disturbances. When tuning, we recommend that changes be made in the MANUAL mode, followed by a return to AUTO. Output limiting is applied in the MANUAL mode.

TIP

- This method requires that the PID instruction controls a non-critical application in terms of personal safety and equipment damage.
- The PID tuning procedure may not work for all cases. It is strongly recommended to use a PID Loop tuner package for the best result (i.e. RSTune, Rockwell Software catalog number 9323-1003D).

Procedure

1. Create your ladder program. Make certain that you have properly scaled your analog input to the range of the process variable PV and that you have properly scaled your control variable CV to your analog output.

2. Connect your process control equipment to your analog modules. Download your program to the processor. Leave the processor in the program mode.

ATTENTION: Ensure that all possibilities of machine motion have been considered with respect to personal safety and equipment damage. It is possible that your output CV may swing between 0 and 100% while tuning.

TIP If you want to verify the scaling of your continuous system and/or determine the initial loop update time of your system, go to the procedure on page 305.

3. Enter the following values: the initial setpoint SP value, a reset T_i of 0, a rate T_d of 0, a gain K_c of 1, and a loop update of 5.

   Set the PID mode to STI or Timed, per your ladder diagram. If STI is selected, ensure that the loop update time equals the STI time interval.

   Enter the optional settings that apply (output limiting, output alarm, MaxS - MinS scaling, feed forward).

4. Get prepared to chart the CV, PV, analog input, or analog output as it varies with time with respect to the setpoint SP value.
5. Place the PID instruction in the MANUAL mode, then place the processor in the RUN mode.
6. While monitoring the PID display, adjust the process manually by writing to the CO percent value.
7. When you feel that you have the process under control manually, place the PID instruction in the AUTO mode.
8. Adjust the gain while observing the relationship of the output to the setpoint over time.
9. When you notice that the process is oscillating above and below the setpoint in an even manner, record the time of 1 cycle. That is, obtain the natural period of the process.

   Natural Period ≅ 4x deadtime

   Record the gain value. Return to the MANUAL mode (stop the process if necessary).

10. Set the loop update time (and STI time interval if applicable) to a value of 5 to 10 times faster than the natural period.

For example, if the cycle time is 20 seconds, and you choose to set the loop update time to 10 times faster than the natural rate, set the loop update time to 200, which would result in a 2-second rate.

11. Set the gain K_c value to 1/2 the gain needed to obtain the natural period of the process. For example, if the gain value recorded in step 9 was 80, set the gain to 40.
12. Set the reset term T_i to approximate the natural period. If the natural period is 20 seconds, as in our example, you would set the reset term to 3 (0.3 minutes per repeat approximates 20 seconds).
13. Now set the rate T_d equal to a value 1/8 that of the reset term. For our example, the value 4 is used to provide a rate term of 0.04 minutes per repeat.
14. Place the process in the AUTO mode. If you have an ideal process, the PID tuning is complete.
15. To make adjustments from this point, place the PID instruction in the MANUAL mode, enter the adjustment, then place the PID instruction back in the AUTO mode.

    This technique of going to MANUAL, then back to AUTO, ensures that most of the "gain error" is removed at the time each adjustment is made. This allows you to see the effects of each adjustment immediately. Toggling the PID rung allows the PID instruction to restart itself, eliminating all of the integral buildup. You may want to toggle the PID rung false while tuning to eliminate the effects of previous tuning adjustments.

Verifying the Scaling of Your Continuous System

To ensure that your process is linear, and that your equipment is properly connected and scaled, do the following:

1. Place the PID instruction in MANUAL and enter the following parameters:
   - type: `0` for MinS
   - type: `100` for MaxS
   - type: `0` for CO%
2. Enter the REM RUN mode and verify that PV=0.
3. Type: `20` in CO%
4. Record the PV = _____
5. Type: `40` in CO%.
6. Record the PV = _____
7. Type: `60` in CO%.

8. Record the PV = _____
9. Type: `80` in CO%.
10. Record the PV = _____
11. The values you recorded should be offset from CO% by the same amount. This proves the linearity of your process. The following example shows an offset progression of fifteen.
    - CO 20% = PV 35%
    - CO 40% = PV 55%
    - CO 60% = PV 75%
    - CO 80% = PV 95%

If the values you recorded are not offset by the same amount:

- Either your scaling is incorrect, or
- the process is not linear, or
- your equipment is not properly connected and/or configured.

Make the necessary corrections and repeat steps 2-10.

Determining the Initial Loop Update Time

To determine the approximate loop update time that should be used for your process, perform the following:

1. Place the normal application values in MinS and MaxS.
2. Type: `50` in CO%.
3. Type: `60` in CO% and immediately start your stopwatch.
4. Watch the PV. When the PV starts to change, stop your stopwatch. Record this value. It is the deadtime.
5. Multiply the deadtime by 4. This value approximates the natural period. For example, if deadtime = 3 seconds,
   then 4 x 3 = 12 seconds (≅ natural period)
6. Divide the value obtained in step 5 by 10. Use this value as the loop updated time. For example, if:

   natural period = 12 seconds, then 12/10 = 1.2 seconds.

   Therefore, the value 120 would be entered as the loop update time.
   (120 x 10 ms = 1.2 seconds)
7. Enter the following values: the initial setpoint SP value, a reset T_i of 0, a rate T_d of 0, a gain K_c of 1, and the loop update time determined in step 17.

Set the PID mode to STI or Timed, per your ladder diagram. If STI is selected, ensure that the loop update time equals the STI time interval.

Enter the optional settings that apply (output limiting, output alarm, MaxS - MinS scaling, feed forward).

8. Return to page 304 and complete the tuning procedure starting with step 4.

Notes:

ASCII Instructions

This chapter contains general information about the ASCII instructions and explains how they function in your control program. This chapter is arranged into the following sections:

General Information

- Instruction Types and Operation on page 310
- Protocol Overview on page 311
- String (ST) Data File on page 312
- Control Data File on page 313

ASCII Instructions

The ASCII instructions are arranged so that the Write instructions precede the Read instructions.

Instruction	Function	Page
ACL - ASCII Clear Buffer	Clear the receive and/or transmit buffers.	314
AIC - Integer to String	Convert an integer value to a string.	316
AWA - ASCII Write with Append	Write a string with user-configured characters appended.	316
AWT - ASCII Write	Write a string.	319
ABL - Test Buffer for Line	Determine the number of characters in the buffer, up to and including the end-of-line character.	321
ACB - Number of Characters in Buffer	Determine the total number of characters in the buffer.	322
ACI - String to Integer	Convert a string to an integer value.	323
ACN - String Concatenate	Link two strings into one.	325
AEX - String Extract	Extract a portion of a string to create a new string.	326
AHL - ASCII Handshake Lines	Set or reset modem handshake lines.	327
ARD - ASCII Read Characters	Read characters from the input buffer and place them into a string.	328
ARL - ASCII Read Line	Read one line of characters from the input buffer and place them into a string.	330
ASC - String Search	Search a string.	331
ASR - ASCII String Compare	Compare two strings.	333

Instruction Types and Operation

There are two types of ASCII instructions, ASCII string control and ASCII port control. The string control instruction type is used for manipulating data and executes immediately. The port control instruction type is used for transmitting data and makes use of the ASCII queue. More details are provided below.

ASCII String Control

These instructions are used to manipulate string data. When a string control instruction is encountered in a ladder logic program, it executes immediately. It is never sent to the ASCII queue to wait for execution. The following table lists the ASCII string control instructions used by the MicroLogix 1400 controllers:

MicroLogix 1400	
ACI (String to Integer)	AIC (Integer to String)
ACN (String Concatenate)	ASC (String Search)
AEX (String Extract)	ASR (ASCII String Compare)

ASCII Port Control

These instructions use or alter the communication channel for receiving or transmitting data. All ASCII port control instructions support channel 2, as well as channel 0. The following table lists the ASCII port control instructions used by the MicroLogix 1400 controllers:

MicroLogix 1400	
ABL (Test Buffer for Line)	ARD (ASCII Read Characters)
ACB (Number of Characters in Buffer)	ARL (ASCII Read Line)
ACL (ASCII Clear Buffer)	AWA (ASCII Write with Append)
AHL (ASCII Handshake Lines)	AWT (ASCII Write)

When the ACL (ASCII Clear Buffer) instruction is encountered in a ladder logic program, it executes immediately and causes all instructions to be removed from the ASCII queue, including stopping execution of the ASCII instruction currently executing. The ER (error) bit is set for each instruction that is removed from the ASCII queue.

When any of the other port control instructions are encountered in a ladder logic program, it may or may not execute immediately depending on the contents of the ASCII queue. The ASCII queue is a FIFO (first-in, first-out) queue which can contain up to 16 instructions. The ASCII queue operates as follows:

- When the instruction is encountered on a rung and the ASCII queue is empty, the instruction executes immediately. It may take several program scans for the instruction to complete.

- When the instruction is encountered on a rung and there are from 1 to 15 instructions in the ASCII queue, the instruction is put into the ASCII queue and is executed when the preceding instructions are completed. If the ASCII queue is full, the instruction waits until the next program scan to determine if it can enter the ASCII queue. The controller continues executing other instructions while the ASCII port control instruction is waiting to enter the queue.

Programming ASCII Instructions

When programming ASCII output instructions, always precede the ASCII instruction with conditional logic that detects when new data needs to be sent or, send data on a time interval. If sent on a time interval, use an interval of 0.5 second or greater. Do not continuously generate streams of ASCII data out of a communications port.

IMPORTANT	If ASCII write instructions execute continuously, you may not be able to re-establish communications with RSLogix 500/RSLogix Micro when the controller is placed into the RUN mode.

Protocol Overview

Using the Full ASCII Instruction Set

To use the full ASCII instruction set, the serial communication channel must be configured for ASCII protocol, as described in ASCII Driver on page 566. Configuration of the two append characters for the AWA instruction can be found in the General tab of the Channel Configuration screen.

Using AWA and AWT Instructions with Other Serial Channel Drivers

The AWA and AWT instructions may still be used with the following serial channel drivers to send strings out of the serial port even though the channel is not configured for ASCII (which is useful for sending ASCII dial-up strings to an attached phone modem):

- DF1 Full-Duplex
- DF1 Half-Duplex Master
- Modbus RTU Master
- DNP3 Slave

The serial channel driver packets take precedence over ASCII strings, so if an AWA or AWT instruction is triggered while a driver packet is being transmitted, the ASCII instruction will error out with an error code of 5.

If the serial channel is configured for any of the following drivers, and an AWA or AWT instruction is triggered, the ASCII instruction will immediately error out with an error code of 9:

- DH-485
- DF1 Half-Duplex Slave
- DF1 Radio Modem
- Modbus RTU Slave

String (ST) Data File

File Description

The string data file is used by the ASCII instructions to store ASCII character data. The ASCII data can be accessed by the source and destination operands in the ASCII instructions. The string data file can also be used by the copy (COP) and move (MOV, MVM) instructions.

String files consist of 42-word elements. One string file element is shown below. You can have up to 256 of these elements in the string file.

String Data File Structure

	String Element															
Bit	**15**	**14**	**3**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
Word	***upper byte***								***lower byte***							
0	String Length - number of characters (range is from 0…82)															
1	character 0								character 1							
2	character 2ayy								character 3							
↓	↓								↓							
40	character 78								character 79							
41	character 80								character 81							

Addressing String Files

The addressing scheme for the string data file is shown below.

Addressing Scheme for String Files

Format	**Explanation**		
	ST	String file	
STf:e.s	**f**	File number	The valid file number range is from 3…255.
	:	Element delimiter	
	e	Element number	The valid element number range is from 0…255. Each element is 42 words in length as shown in .

Addressing Scheme for String Files

Format	Explanation		
	.	Subelement delimiter	
	s	Subelement number	The valid subelement number range is from 0…41. You can also specify .LEN for word 0 and .DATA[0] through .DATA[40] for words 1…41. The subelement represents a word address.
Examples:	**ST9:2** **ST17:1.LEN** **ST13:7.DATA[1]**		**String File 9, Element 2** **String File 17, Element 1, LEN Variable** **String File 13, Element 7, word 2 (characters 2 and 3)**

Control Data File

File Description

The control data element is used by ASCII instructions to store control information required to operate the instruction. The control data element for ASCII instructions includes status and control bits, an error code byte, and two character words as shown below:

ASCII Instructions Control Data File Elements

	Control Element															
Word	**15**	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
0	EN (1)	EU(2)	DN(3)	EM(4)	ER(5)	UL (6)	R N (7)	FD(8)	Error Code Byte							
1	Number of characters specified to be sent or received (LEN)															
2	Number of characters actually sent or received (POS)															

(1) EN = Enable Bit - indicates that an instruction is enabled due to a false-to-true transition. This bit remains set until the instruction completes execution or generates an error.

(2) EU = Queue Bit - when set, indicates that an ASCII instruction was placed in the ASCII queue. This action is delayed if the queue is already filled.

(3) DN = Asynchronous Done Bit - is set when an instruction successfully completes its operation.

(4) EM = Synchronous Done Bit - not used

(5) ER = Error Bit - when set, indicates that an error occurred while executing the instruction.

(6) UL = Unload Bit - when this bit is set by the user, the instruction does not execute. If the instruction is already executing, operation ceases. If this bit is set while an instruction is executing, any data already processed is sent to the destination and any remaining data is not processed. Setting this bit will not cause instructions to be removed from the ASCII queue. This bit is only examined when the instruction is ready to start executing.

(7) RN = Running Bit - when set, indicates that the queued instruction is executing. NOTE: The RN bit is not addressable via the Control (R) file.

(8) FD = Found Bit - when set, indicates that the instruction has found the end-of-line or termination character in the buffer. (only used by the ABL and ACB instructions)

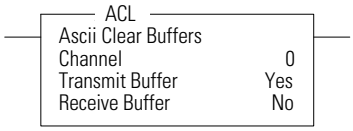
Addressing Control Files

The addressing scheme for the control data file is shown below.

Addressing Scheme for Control Files

Format	Explanation		
	R	Control file	
R:e.s/b	**f**	File number	The valid file number range is from 3…255.
	:	Element delimiter	
	e	Element number	The valid element number range is from 0…255. Each element is 3 words in length as shown in .
	.	Subelement delimiter	
	s	Subelement number	The valid subelement number range is from 0…2. You can also specify .LEN or .POS.
	/	Bit delimiter	
	b	Bit number	The valid bit number range is from 0…15. The bit number is the bit location within the string file element. Bit level addressing is not available for words 1 and 2 of the control element.
Examples:	**R6:2** **R6:2.0/13** **R18:1.LEN** **R18:1.POS**		**Element 2, control file 6** **Bit 13 in sub-element 0 of element 2, control file 6** **Specified string length of element 1, control file 18** **Actual string length of element 1, control file 18**

ACL - ASCII Clear Buffers



Instruction Type: output

Execution Time for the ACL Instruction

Controller	When Instruction Is:	
	True	**False**
MicroLogix 1400	clear buffers:	
both	26.5540 µs	0.4500 µs
receive	7.8820 µs	0.3848 µs
transmit	5.8590 µs	0.3706 µs

The ACL instruction clears the Receive and/or Transmit buffer(s). This instruction also removes instructions from ASCII queue.

TIP For MicroLogix 1400, the ACL instruction can also be used to clear the DF1 communication buffers when the channel is configured for any of the DF1 communication drivers.

Select 0 for the channel number that is configured for DF1 and Yes for both the Receive and Transmit Buffers. When the ACL instruction is executed, any pending outgoing DF1 replies, any pending incoming DF1 commands and any pending outgoing DF1 commands are flushed. Any MSG instructions in progress on that channel will error out with an error code of 0x0C. However, this functionality is not applied if DCOMM (Default communication setting) is selected.

This instruction executes immediately upon the rung transitioning to a true state. Any ASCII transmissions in progress are terminated when the ACL instruction executes.

TIP The ASCII queue may contain up to 16 instructions that are waiting to run.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Receive Buffer** clears the Receive buffer when set to "Yes" and removes the Receive ASCII port control instructions (ARL and ARD) from the ASCII queue.
- **Transmit Buffer** clears the Transmit buffer when set to "Yes" and removes the Transmit ASCII port control instructions (AWA and AWT) from the ASCII queue.

Addressing Modes and File Types can be used as shown below:

ACL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]														Function Files										CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD					Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																										•				•		
Receive Buffer																										•				•		
Transmit Buffer																										•				•		

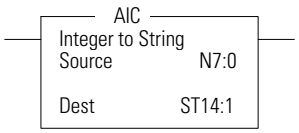
(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

When Clear Receive Buffer and Clear Transmit Buffer are both set to Yes, all Receive and Transmit instructions (ARL, ARD, AWA, and AWT) are removed from the ASCII queue.

When instructions are removed from the ASCII queue, the following bits are set: ER = 1, RN = 0, EU = 0, and ERR = 0x0E.

AIC - ASCII Integer to String



Instruction Type: output

Execution Time for the AIC Instruction

Controller	Data Size	When Instruction Is:	
		True	False
MicroLogix 1400	word	8.3032 µs	0.2591 µs
	long word	9.8913 µs	0.2155 µs

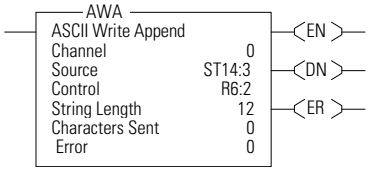
The AIC instruction converts an integer or long word value (source) to an ASCII string (destination). The source can be a constant or an address. The source data range is from -2,147,483,648...2,147,483,647.

Addressing Modes and File Types can be used as shown below:

AIC Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files													Function Files									Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•				•		•														•	•			•	•	
Destination								•																			•					•

AWA - ASCII Write with Append



Instruction Type: output

Execution Time for the AWA Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	10.7810 µs	9.0122 µs

Use the AWA instruction to write characters from a source string to an external device. This instruction adds the two appended characters that you configure on the Channel Configuration screen. The default is a carriage return and line feed appended to the end of the string.

TIP You configure append characters via the Channel Configuration screen. The default append characters are carriage return and line feed.

Programming AWA Instructions

When programming ASCII output instructions, always precede the ASCII instruction with conditional logic that detects when new data needs to be sent or, send data on a time interval. If sent on a time interval, use an interval of 0.5 second or greater. Do not continuously generate streams of ASCII data out of a communications port.

IMPORTANT If ASCII write instructions execute continuously, you may not be able to re-establish communications with RSLogix 500/RSLogix Micro when the controller is placed into the RUN mode.

This instruction will execute on either a false or true rung. However, if you want to repeat this instruction, the rung must go from false-to-true.

When using this instruction you can also perform in-line indirection. See page Using In-line Indirection on page 334 for more information.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Source** is the string element you want to write.
- **Control** is the control data file. See page 313.
- **String Length** (.LEN) is the number of characters you want to write from the source string (0...82). If you enter a 0, the entire string is written. This is word 1 in the control data file.
- **Characters Sent** (.POS) is the number of characters that the controller sends to an external device. This is word 2 in the control data file. Characters Sent (.POS) is updated after all characters have been transmitted.

  The valid range for .POS is from 0...84. The number of characters sent to the destination may be smaller or greater than the specified String Length (.LEN) as described below:

  - Characters Sent (.POS) may be smaller than String Length (.LEN) if the length of the string sent is less than what was specified in the String Length (.LEN) field.
  - Characters Sent (.POS) can be greater than the String Length (.LEN) if the appended characters or inserted values from in-line indirection are used. If the String Length (.LEN) is greater than 82, the string written to the destination is truncated to 82 characters plus the number of append characters (this number could be 82, 83, or 84 depending on how many append characters are used).
- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error code descriptions.

Addressing Modes and File Types can be used as shown below:

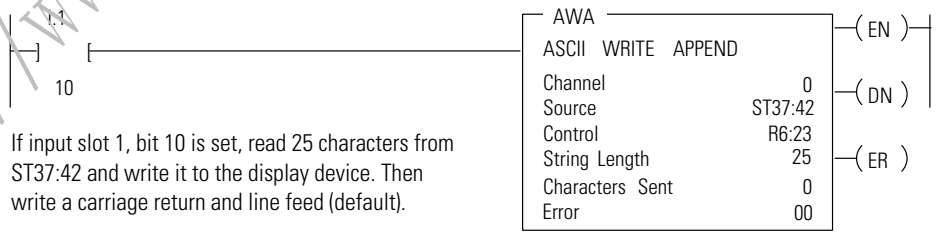
AWA Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	SFI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																									•				•		
Source								•																		•					•
Control					•																					•					•

(1) The Control data file is the only valid file type for the Control Element.

Example



If input slot 1, bit 10 is set, read 25 characters from ST37:42 and write it to the display device. Then write a carriage return and line feed (default).

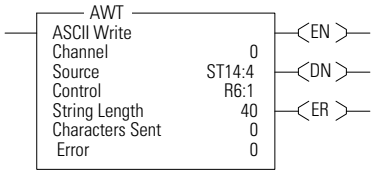
In this example, when the rung goes from false-to-true, the control element Enable (EN) bit is set. When the instruction is placed in the ASCII queue, the Queue bit (EU) is set. The Running bit (RN) is set when the instruction is executing**.** The DN bit is set on completion of the instruction.

The controller sends 25 characters from the start of string ST37:42 to the display device and then sends user-configured append characters. The Done bit (DN) is set and a value of 27 is present in .POS word of the ASCII control data file.

When an error is detected, the error code is written to the Error Code Byte and the Error Bit (ER) is set. See ASCII Instruction Error Codes on page 335 for a list of the error codes and recommended action to take.

TIP For information on the timing of this instruction, see the timing diagram on page 334.

AWT - ASCII Write



Instruction Type: output

Execution Time for the AWT Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	13.6110 µs	7.2706 µs

Use the AWT instruction to write characters from a source string to an external device.

Programming AWT Instructions

When programming ASCII output instructions, always precede the ASCII instruction with conditional logic that either detects when new data needs to be sent or, send data on a time interval. If sent on a time interval, use an interval of 0.5 second or greater.

IMPORTANT Do not continuously generate streams of ASCII data out of a communications port. If ASCII write instructions execute continuously, you may not be able to re-establish communications with RSLogix 500/RSLogix Micro when the controller is placed into the RUN mode.

This instruction executes on a true rung. Once started, if the rung goes false, the instruction continues to completion. If you want to repeat this instruction, the rung must transition from false-to-true.

When using this instruction you can also perform in-line indirection. See page 334 for more information.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Source** is the string element you want to write.
- **Control** is the control data file. See page 313.

- **String Length** (.LEN) is the number of characters you want to write from the source string (0...82). If you enter a 0, the entire string is written. This is word 1 in the control data file.
- **Characters Sent** (.POS) is the number of characters that the controller sends to an external device. This is word 2 in the control data file. Characters Sent (.POS) is updated after all characters have been transmitted.

  The valid range for .POS is from 0...82. The number of characters sent to the destination may be smaller or greater than the specified String Length (.LEN) as described below:
  - Characters Sent (.POS) may be smaller than String Length (.LEN) if the length of the string sent is less than what was specified in the String Length (.LEN) field.
  - Characters Sent (.POS) can be greater than the String Length (.LEN) if inserted values from in-line indirection are used. If the String Length (.LEN) is greater than 82, the string written to the destination is truncated to 82 characters.
- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error code descriptions.

Addressing Modes and File Types can be used as shown below:

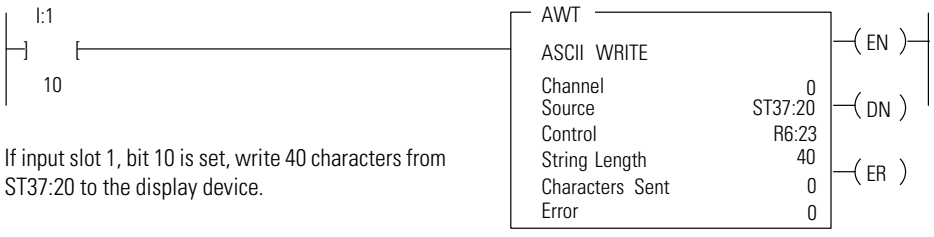
AWT Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																									•				•		
Source								•																		•					•
Control					•																					•					•

(1) The Control data file is the only valid file type for the Control Element.

Example



If input slot 1, bit 10 is set, write 40 characters from ST37:20 to the display device.

In this example, when the rung goes from false-to-true, the control element Enable (EN) bit is set. When the instruction is placed in the ASCII queue, the

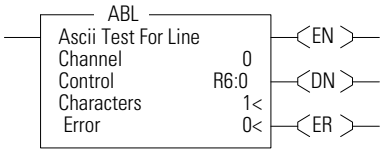
Queue bit (EU) is set. The Running bit (RN) is set when the instruction is executing. The DN bit is set on completion of the instruction.

Forty characters from string ST37:40 are sent through channel 0. The Done bit (DN) is set and a value of 40 is present in the POS word of the ASCII control data file.

When an error is detected, the error code is written to the Error Code Byte and the Error Bit (ER) is set. See ASCII Instruction Error Codes on page 335 for a list of the error codes and recommended action to take.

TIP For information on the timing of this instruction, see the timing diagram on page 334.

ABL - Test Buffer for Line



Instruction Type: output

Execution Time for the ABL Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	21.5621 µs	1.8710 µs

The ABL instruction is used to determine the number of characters in the receive buffer of the specified communication channel, up to and including the end-of-line characters (termination). This instruction looks for the two termination characters that you configure via the channel configuration screen. On a false-to-true transition, the controller reports the number of characters in the POS field of the control data file. The channel configuration must be set to ASCII.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Control** is the control data file. See page 313.
- **Characters** are the number of characters in the buffer that the controller finds (0...1024). This parameter is read-only and resides in word 2 of the control data file.
- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error code descriptions.

Addressing Modes and File Types can be used as shown below:

ABL Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files								Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																									•				•		
Control					•																					•					•

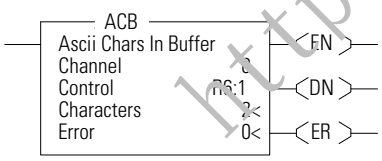
(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

When the rung goes from false-to-true, the Enable bit (EN) is set. The instruction is put in the ASCII instruction queue, the Queue bit (EU) is set, and program scan continues. The instruction is then executed outside of the program scan. However, if the queue is empty the instruction executes immediately. Upon execution, the Run bit (RN) is set.

The controller determines the number of characters (up to and including the termination characters) and puts this value in the POS field of the control data file. The Done bit (DN) is then set. If a zero appears in the POS field, no termination characters were found. The Found bit (FD) is set if the POS field is set to a non-zero value.

ACB - Number of Characters in Buffer



Instruction Type: output

Execution Time for the ACB Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	22.6154 µs	3.5250 µs

Use the ACB instruction to determine the number of characters in the buffer. On a false-to-true transition, the controller determines the total number of characters and records it in the POS field of the control data file. The channel configuration must be set to ASCII.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Control** is the control data file. See page 313.
- **Characters** are the number of characters in the buffer that the controller finds (0...1024). This parameter is read-only.
- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error descriptions.

Addressing Modes and File Types can be used as shown below:

ACB Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files								Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	HMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																									•				•		
Control					•																					•					•

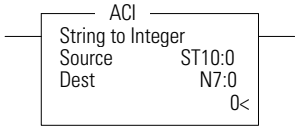
(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

When the rung goes from false-to-true, the Enable bit (EN) is set. When the instruction is placed in the ASCII queue, the Queue bit (EU) is set. The Running bit (RN) is set when the instruction is executing. The Done bit (DN) is set on completion of the instruction.

The controller determines the number of characters in the buffer and puts this value in the POS field of the control data file. The Done bit (DN) is then set. If a zero appears in the POS field, no characters were found. The Found bit (FD) is set when the POS filed is set to a non-zero value

ACI - String to Integer



Instruction Type: output

Execution Time for the ACI Instruction

Controller	Data Size	When Instruction Is:	
		True	False
MicroLogix 1400	word	6.5719 µs	0.2142 µs
	long word	7.1146 µs	0.1978 µs

Use the ACI instruction to convert a numeric ASCII string to an integer (word or long word) value.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source** - The contents of this location are converted to an integer value.
- **Destination** - This is the location which receives the result of the conversion. The data range is from -32,768...32,767 if the destination is a word and from -2,147,483,648...2,147,483,647 if the destination is a long word.

Addressing Modes and File Types can be used as shown below:

ACI Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MAI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source								•																			•					•
Destination	•	•		•	•				•	•		•															•			•	•	

(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

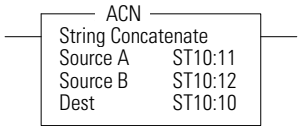
The controller searches the source (file type ST) for the first character between 0 and 9. All numeric characters are extracted until a non-numeric character or the end of the string is reached. Action is taken *only* if numeric characters are found. The string length is limited to 82 characters. Commas and signs (+, -) are allowed in the string. However, only the minus sign is displayed in the data table.

This instruction sets the following math flags in the controller status file:

Controller Status File Math Flags

Math Flag		Description
S:0/1	Overflow (V)	Flag is set if the result is outside of the valid range.
S:0/2	Zero (Z)	Flag is set if the result is zero.
S:0/3	Sign (S)	Flag is set if the result is negative.
S:5/0	Overflow Trap	Flag is set when the Overflow flag (S:0/1) is set.
S:5/15	ASCII String Manipulation Error	Flag is set if the Source string exceeds 82 characters. When S:5/15 is set, the Invalid String Length Error (1F39H) is written to the Major Error Fault Code (S:6).

ACN - String Concatenate



Instruction Type: output

Execution Time for the ACN Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	9.4852 µs	0.1982 µs

The ACN instruction combines two ASCII strings. The second string is appended to the first and the result stored in the destination.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source A** is the first string in the concatenation procedure.
- **Source B** is the second string in the concatenation procedure.
- **Destination** is where the result of Source A and B is stored.

Addressing Modes and File Types can be used as shown below:

ACN Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

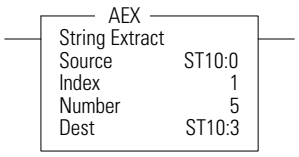
Parameter	Data Files[1]												Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A								•																	•					•
Source B								•																	•					•
Destination						•		•																	•					•

(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

This instruction executes on a false-to-true rung transition. Source B is appended to Source A and the result is put in the Destination. Only the first 82 characters (0...81) are written to the destination. If the string length of Source A, Source B, or Destination is greater than 82, the ASCII String Manipulation Error bit S:5/15 is set and the Invalid String Length Error (1F39H) is written to the Major Error Fault Code word (S:6).

AEX - String Extract



Instruction Type: output

Execution Time for the AEX Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	10.0290 µs	0.1850 µs

The AEX instruction creates a new string by taking a portion of an existing string and storing it in a new string.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source** is the existing string. The Source value is not affected by this instruction.
- **Index** is the starting position (from 1...82) of the string you want to extract. (An index of 1 indicates the left-most character of the string.)
- **Number** is the number of characters (from 1...82) you want to extract, starting at the indexed position. If the Index plus the Number is greater than the total characters in the source string, the Destination string will be the characters from the Index to the end of the Source string.
- **Destination** is the string element (ST) where you want the extracted string stored.

Addressing Modes and File Types can be used as shown below:

AEX Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files									Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source								•				•															•					•
Index	•	•		•	•	•			•			•														•				•		
Number	•	•		•	•	•			•																	•				•		
Destination								•																			•						•

(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

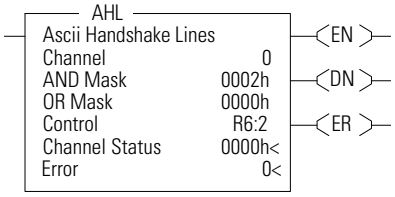
This instruction executes on a true rung.

The following conditions cause the controller to set the ASCII String Manipulation Error bit (S:5/15):

- Source string length is less than 1 or greater than 82
- Index value is less than 1 or greater than 82
- Number value is less than 1 or greater than 82
- Index value greater than the length of the Source string

The Destination string is not changed in any of the above error conditions. When the ASCII String Manipulation Error bit (S:5/15) is set, the Invalid String Length Error (1F39H) is written to the Major Error Fault Code word (S:6).

AHL - ASCII Handshake Lines



Instruction Type: output

Execution Time for the AHL Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	26.5267 µs	2.9480 µs

The AHL instruction is used to set or reset the RS-232 Request to Send (RTS) handshake control line for a modem. The controller uses the two masks to determine whether to set or reset the RTS control line, or leave it unchanged. The channel configuration must be set to ASCII.

TIP Make sure the automatic modem control used by the port does not conflict with this instruction.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **AND Mask** is the mask used to *reset* the RTS control line. Bit 1 corresponds to the RTS control line. A value of "2" in the AND mask resets the RTS control line; a value of "0" leaves the line unchanged.
- **OR Mask** is the mask used to *set* the RTS control line. Bit 1 corresponds to the RTS control line. A value of "2" in the OR mask sets the RTS control line; a value of "0" leaves the line unchanged.
- **Control** is the control data file. See page 313.
- **Channel Status** displays the current status (0000...001F) of the handshake lines for the specified channel. This status is read-only and resides in the .POS field in the control data file. The following shows how to determine the channel status value. In this example, the value is 001F.

Channel Status Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Handshake Control Line Setting	Reserved											--		--	RTS	CTS
	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Channel Status	0				0				1				F			
	Word 2 of the Control Element = 001F															

- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error code descriptions.

Addressing Modes and File Types can be used as shown below:

AHL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

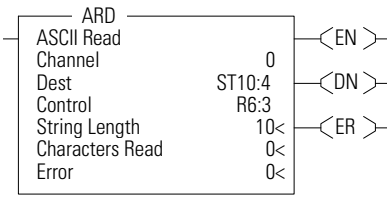
Parameter	Data Files[1]													Function Files										Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTO, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																									•				•		
AND Mask	•	•		•	•	•			•			•													•	•			•		
OR Mask	•	•		•	•	•			•			•													•	•			•		
Control					•																					•					•

(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

This instruction executes on either a false or true rung. However a false-to-true rung transition is required to set the EN bit to repeat the instruction.

ARD - ASCII Read Characters



Instruction Type: output

Execution Time for the ARD Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	9.3760 µs	7.7770 µs

Use the ARD instruction to read characters from the buffer and store them in a string. To repeat the operation, the rung must go from false-to-true.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Destination** is the string element where you want the characters stored.
- **Control** is the control data file. See page 313.
- **String Length** (LEN) is the number of characters you want to read from the buffer. The maximum is 82 characters. If you specify a length larger than 82, only the first 82 characters will be read. If you specify 0 characters, LEN defaults to 82. This is word 1 in the control data file.
- **Characters Read** (POS) is the number of characters that the controller moved from the buffer to the string (0...82). This field is updated during the execution of the instruction and is read-only. This is word 2 in the control data file.
- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error code descriptions.

Addressing Modes and File Types can be used as shown below:

ARD Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70

Parameter	Data Files[1]													Function Files											Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																										•				•		
Destination								•																			•					•
Control					•																						•					•

(1) The Control data file is the only valid file type for the Control Element.

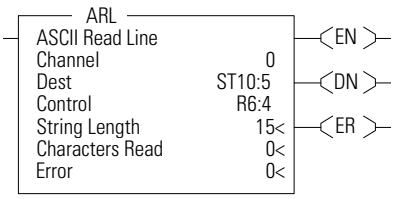
Instruction Operation

When the rung goes from false-to-true, the Enable bit (EN) is set. When the instruction is placed in the ASCII queue, the Queue bit (EU) is set. The Running bit (RN) is set when the instruction is executing. The DN bit is set on completion of the instruction.

Once the requested number of characters are in the buffer, the characters are moved to the destination string. The number of characters moved is put in the POS field of the control data file. The number in the POS field is continuously updated and the Done bit (DN) is not set until all of the characters are read.

TIP For information on the timing of this instruction, see the timing diagram on page 334.

ARL - ASCII Read Line



Instruction Type: output

Execution Time for the ARL Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	33.9910 µs	8.5690 µs

Use the ARL instruction to read characters from the buffer, up to and including the Termination characters, and store them in a string. The Termination characters are specified via the Channel Configuration screen.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Channel** is the number of the serial port being used, 0 or 2.
- **Destination** is the string element where you want the string stored.
- **Control** is the control data file. See page 313.
- **String Length** (LEN) is the number of characters you want to read from the buffer. The maximum is 82 characters. If you specify a length larger than 82, only the first 82 characters are read and moved to the destination. (A length of "0" defaults to 82.) This is word 1 in the control data file.
- **Characters Read** (POS) is the number of characters that the controller moved from the buffer to the string (0...82). This field is read-only and resides in word 2 of the control data file.
- **Error** displays the hexadecimal error code that indicates why the ER bit was set in the control data file. See page 335 for error code descriptions.

Addressing Modes and File Types can be used as shown below:

ARL Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Channel																									•				•		
Destination								•																		•					•
Control					•																					•					•

(1) The Control data file is the only valid file type for the Control Element.

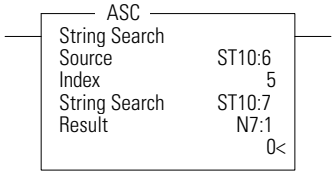
Instruction Operation

When the rung goes from false-to-true, the control element Enable (EN) bit is set. When the instruction is placed in the ASCII queue, the Queue bit (EU) is set. The Running bit (RN) is set when the instruction is executing. The DN bit is set on completion of the instruction.

Once the requested number of characters are in the buffer, all characters (including the Termination characters) are moved to the destination string. The number of characters moved is stored in the POS word of the control data file. The number in the Characters Read field is continuously updated and the Done bit (DN) is not set until all of the characters have been read. Exception: If the controller finds termination characters before done reading, the Done bit (DN) is set and the number of characters found is stored in the POS word of the control data file.

TIP For information on the timing of this instruction, see the timing diagram on page 334.

ASC - String Search



Instruction Type: output

Execution Time for the ASC Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400r	8.0844 µs	0.1984 µs

Use the ASC instruction to search an existing string for an occurrence of the source string. This instruction executes on a true rung.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source** is the address of the string you want to find.
- **Index** is the starting position (from 1...82) within the search string. (An index of 1 indicates the left-most character of the string.)
- **Search** is the address of the string you want to examine.
- **Result** is the location (from 1...82) that the controller uses to store the position in the Search string where the Source string begins. If no match is found, result is set equal to zero.

Addressing Modes and File Types can be used as shown below:

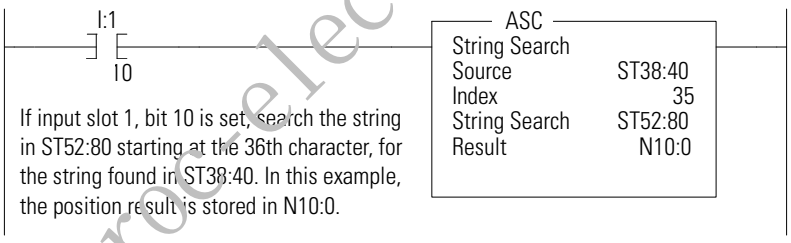
ASC Instruction Valid Addressing Modes and File Types

For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files[1]													Function Files									CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source	•	•		•	•	•																				•	•			•		
Index								•	•			•															•					•
Search								•																			•					•
Result	•	•		•	•	•			•			•															•			•		

(1) The Control data file is the only valid file type for the Control Element.

Example



If input slot 1, bit 10 is set, search the string in ST52:80 starting at the 36th character, for the string found in ST38:40. In this example, the position result is stored in N10:0.

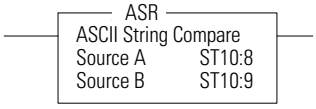
Error Conditions

The following conditions cause the controller to set the ASCII Error bit (S:5/15).

- Source string length is less than 1 or greater than 82.
- Index value is less than 1 or greater than 82.
- Index value is greater than Source string length.

The destination is not changed in any of the above conditions. When the ASCII String Manipulation Error bit (S:5/15) is set, the Invalid String Length Error (1F39H) is written to the Major Error Fault Code word (S:6).

ASR - ASCII String Compare



Instruction Type: input

Execution Time for the ASR Instruction

Controller	When Instruction Is:	
	True	False
MicroLogix 1400	4.8596 µs	0.2016 µs

Use the ASR instruction to compare two ASCII strings. The controller looks for a match in length and upper/lower case characters. If two strings are identical, the rung is true; if there are any differences, the rung is false.

Entering Parameters

Enter the following parameters when programming this instruction:

- **Source A** is the location of the first string used for comparison.
- **Source B** is the location of the second string used for comparison.

Addressing Modes and File Types can be used as shown below:

ASR Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70

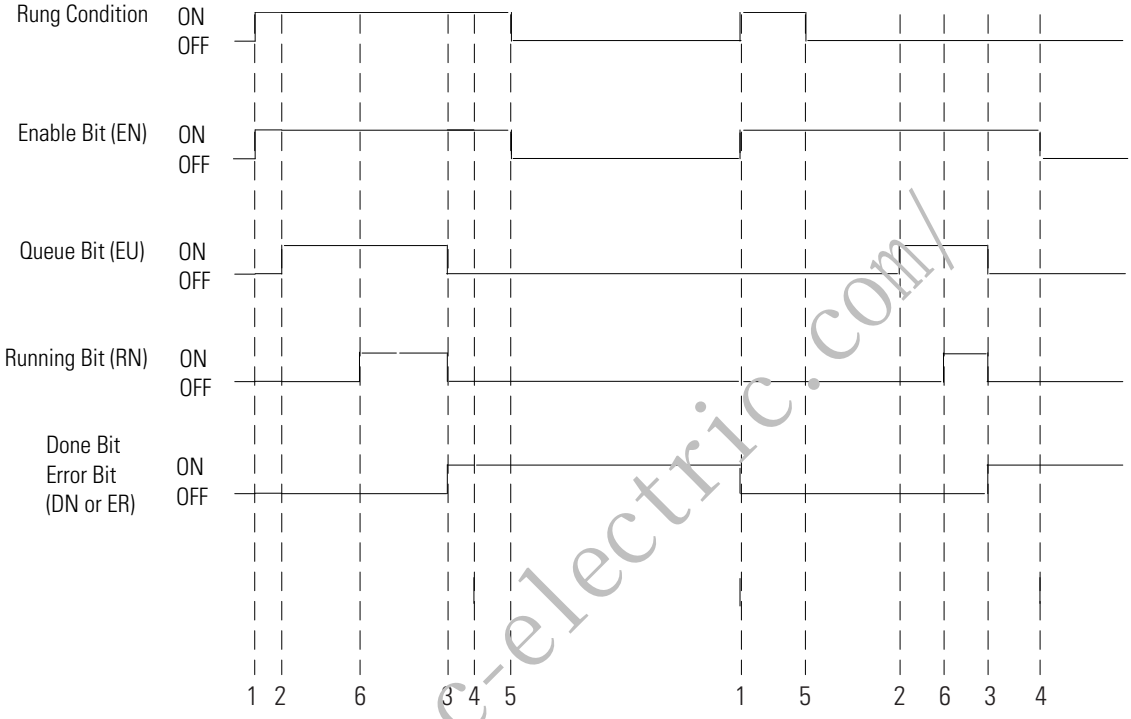
Parameter	Data Files[1]													Function Files												Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	A	L	MG, PD	RI/RIX	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Source A								•																			•					•
Source B								•																			•					•

(1) The Control data file is the only valid file type for the Control Element.

Instruction Operation

If the string length of Source A or Source B exceeds 82 characters, the ASCII String Manipulation Error bit (S:5/15) is set and the rung goes false.

Timing Diagram for ARD, ARL, AWA, and AWT Instructions



NOTE: The RN bit is not addressable via the Control (R) file.

1 - rung goes true
2 - instruction successfully queued
3 - instruction execution complete
4 - instruction scanned for the first time after execution is complete
5 - rung goes false
6 - instruction execution starts

Using In-line Indirection

This allows you to insert integer and long word values into ASCII strings. The Running bit (RN) must be set before the string value can be used.

The following conditions apply to performing in-line indirection:

- All valid integer (N) and long word (L) files can be used.
- File types are not case sensitive and can include either a colon (:) or semicolon (;)
- Positive value symbol (+) and leading zeros are not printed. Negative values (-) are printed with a leading minus sign. Commas are not inserted where they would normally appear in numbers greater than one thousand.

Examples

For the following examples:
N7:0 = 25

N7:1 = -37
L9:0 = 508000
L9:1 = 5

Valid in-line indirection:	
Input:	Flow rate is currently [N7:0] liters per minute and contains [L9:0] particles per liter contaminants.
Output:	Flow rate is currently 25 liters per minute and contains 508000 particles per liter contaminants.
Input:	Current position is [N7:1] at a speed of [L9:1] RPM.
Output:	Current position is -37 at a speed of 5 RPM.
Invalid in-line indirection:	
Input:	Current position is [N5:1] at a speed of [L9:1] RPM.
Output:	Current position is [N5:1] at a speed of 5 RPM.

TIP Truncation occurs in the output string if the indirection causes the output to exceed 82 characters. The appended characters are always applied to the output.

ASCII Instruction Error Codes

The following error codes indicate why the Error bit (ER) is set in the control data file.

Error Code		**Description**	**Recommended Action**
decimal	**hexadecimal**		
0	0x00	No error. The instruction completed successfully.	None Required.
3	0x03	The transmission cannot be completed because the CTS signal was lost.	Check the modem and modem connections.
5	0x05	While attempting to perform an ASCII transmission, a conflict with the configured communications protocol was detected.	Reconfigure the channel and retry operation.
7	0x07	The instruction cannot be executed because the communications channel has been shut down via the channel configuration menu.	Reconfigure the channel and retry operation.
8	0x08	The instruction cannot be executed because another ASCII transmission is already in progress.	Resend the transmission.
9	0x09	Type of ASCII communications operation requested is not supported by the current channel configuration.	Reconfigure the channel and retry operation.
10	0x0A	The unload bit (UL) is set, stopping instruction execution.	None required.
11	0x0B	The requested number of characters for the ASCII read was too large or negative.	Enter a valid string length and retry operation.
12	0x0C	The length of the Source string is invalid (either a negative number or a number greater than 82).	Enter a valid string length and retry operation.

Error Code		Description	Recommended Action
decimal	hexadecimal		
13	0x0D	The requested length in the Control field is invalid (either a negative number or a number greater than 82).	Enter a valid length and retry operation.
14	0x0E	Execution of an ACL instruction caused this instruction to abort.	None required.
15	0x0F	Communications channel configuration was changed while instruction was in progress.	None required.

ASCII Character Set

The table below lists the decimal, hexadecimal, octal, and ASCII conversions.

Standard ASCII Character Set

Column 1					Column 2				Column 3				Column 4			
Ctrl-	DEC	HEX	OCT	ASC	DEC	HEX	OCT	ASC	DEC	HEX	OCT	ASC	DEC	HEX	OCT	ASC
^@	00	00	000	NUL	32	20	040	SP	64	40	100	@	96	60	140	\
^A	01	01	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
^B	02	02	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
^C	03	03	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
^D	04	04	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
^E	05	05	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
^F	06	06	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
^G	07	07	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
^H	08	08	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
^I	09	09	011	HT	41	29	051	)	73	49	111	I	105	69	151	i
^J	10	0A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
^K	11	0B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
^L	12	0C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
^M	13	0D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
^N	14	0E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
^O	15	0F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
^P	16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
^Q	17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
^R	18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
^S	19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
^T	20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
^U	21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
^V	22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
^W	23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
^X	24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
^Y	25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
^Z	26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
^[	27	1B	033	ESC	59	3B	073	;	91	5B	133	[	123	7B	173	{
^\	28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	|
^]	29	1D	035	GS	61	3D	075	=	93	5D	135	]	125	7D	175	}
^^	30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
^_	31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

The standard ASCII character set includes values up to 127 decimal (7F hex). The MicroLogix 1400 Controller also supports an extended character set (decimal 128...255). However, the extended character set may display different characters depending on the platform you are using.

Decimal values 0 through 31 are also assigned Ctrl- codes.

Communications Instructions

This chapter contains information about the Message (MSG) and Service Communications (SVC) communication instructions. This chapter provides information on:

- Messaging Overview on page 337
- SVC - Service Communications on page 339
- MSG - Message on page 341
- The Message Element on page 342
- Timing Diagram for the MSG Instruction on page 348
- MSG Instruction Ladder Logic on page 353
- Local Messages on page 355
- Configuring a Local Message on page 357
- Local Messaging Examples on page 365
- Remote Messages on page 379
- Configuring a Remote Message on page 382
- Configuring a Multi-hop Remote Message on EtherNet/IP Communication Channel on page 385
- Configuring a MicroLogix 1400 CIP Generic Message via Ethernet on page 400
- MSG Instruction Error Codes on page 404
- Special Function with MSG instruction on page 407

The communication instructions read or write data to another station.

Instruction	Used To:	Page
SVC	Interrupt the program scan to execute the service communications part of the operating cycle. The scan then resumes at the instruction following the SVC instruction.	339
MSG	Transfer data from one device to another.	341

Messaging Overview

The communication architecture is comprised of three primary components:

- Ladder Scan
- Communications Buffers
- Communication Queue

These three components determine when a message is transmitted by the controller. For a message to transmit, it must be scanned on a true rung of logic.

When scanned, the message and the data defined within the message (if it is a write message) are placed in a communication buffer. The controller continues to scan the remaining user program. The message is processed and sent out of the controller via the communications port after the ladder logic completes, during the Service Communications part of the operating cycle, unless an SVC is executed.

If a second message instruction is processed before the first message completes, the second message and its data are placed in one of the three remaining communication buffers. This process repeats whenever a message instruction is processed, until all four buffers are in use.

When a buffer is available, the message and its associated data are placed in the buffer immediately. If all four buffers for the channel are full when the next (fifth) message is processed, the message request, not the data, is placed in the channel's communications queue. The queue is a message storage area that keeps track of messages that have not been allocated a buffer. The queue operates as a first-in first-out (FIFO) storage area. The first message request stored in the queue is the message that is allocated a buffer as soon as a buffer becomes available. The queue can accommodate all MSG instructions in a ladder program.

When a message request in a buffer is completed, the buffer is released back to the system. If a message is in the queue, that message is then allocated a buffer. At that time, the data associated with the message is read from within the controller.

TIP If a message instruction was in the queue, the data that is actually sent out of the controller may be different than what was present when the message instruction was first processed.

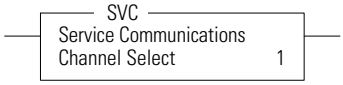
The buffer and queue mechanisms are completely automatic. Buffers are allocated and released as the need arises, and message queuing occurs if buffers are full.

The controller initiates read and write messages through available communication channels when configured for the following protocols:

- DH-485
- DF1 Full-Duplex
- DF1 Half-Duplex Master
- DF1 Half-Duplex Slave
- DF1 Radio Modem
- Modbus RTU Master
- Ethernet

For a description of valid communication protocols, see Knowledgebase Quick Starts on page 577.

SVC - Service Communications



Instruction Type: output

Execution Time for the SVC Instruction

Controller	When Rung Is:(1)	
	True	False
MicroLogix 1400	39.8260 µs (CH0) 5.9042 µs (CH1) 36.5800 µs (CH0 & CH1)	0.1933 µs (CH0) 0.1857 µs (CH1) 0.1774 µs (CH0 & CH1)

(1) This value for the SVC instruction is for when the communications servicing function is accessing a data file. The time increases when accessing a function file.

Under normal operation the controller processes communications once every time it scans the control program. If you require the communications port to be scanned more often, or if the ladder scan is long, you can add an SVC (Service Communications) instruction to your control program. The SVC instruction is used to improve communications performance/throughput, but also causes the ladder scan to be longer.

Simply place the SVC instruction on a rung within the control program. When the rung is scanned, the controller services any communications that need to take place. You can place the SVC instruction on a rung without any preceding logic, or you can condition the rung with a number of communications status bits. The table on page 340 shows the available status file bits.

TIP The amount of communications servicing performed is controlled by the Communication Servicing Selection Bits (CSS) and Message Servicing Selection Bits (MSS) in the Channel Communication Configuration File. Refer to Communication Servicing Selection and Message Servicing Selection on page 352 for more information.

For best results, place the SVC instruction in the middle of the control program. You may not place an SVC instruction in a Fault, DII, STI, or I/O Event subroutine.

Channel Select

When using the SVC instruction, you must select the channel to be serviced. The channel select variable is a one-word bit pattern that determines which channel is serviced. Each bit corresponds to a specific channel. For example, bit 0 equals channel 0. When any bit is set (1), the corresponding channel is serviced.

Channel Select Setting

Controller	Channel Select Setting	Channel(s) Serviced
MicroLogix 1400	1h	Channel 0
	2h	Channel 1
	4h	Channel 2
	7h	All Channels

Communication Status Bits

The following communication status bits allow you to customize or monitor communications servicing. See General Channel Status Block on page 45 for additional status information.

Communication Status Bits

Address		Description
Channel 0	Channel 1	
CS0:4/0	ES:4/0	ICP - Incoming Command Pending
CS0:4/1	ES:4/1	MRP - Incoming Message Reply Pending
CS0:4/2	ES:4/2	MCP - Outgoing Message Command Pending
CS0:4/4		CAB - Communications Active Bit

Application Example

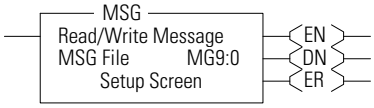
The SVC instruction is used when you want to execute a communication function, such as transmitting a message, prior to the normal service communication portion of the operating scan.

You can place this rung after a message write instruction. CS0:4/MCP is set when the message instruction is enabled and put in the communications queue. When CS0:4/MCP is set (1), the SVC instruction is evaluated as true and the program scan is interrupted to execute the service communication's portion of the operating scan. The scan then resumes at the instruction following the SVC instruction.

The example rung shows a conditional SVC, which is processed only when an outgoing message is in the communications queue.

TIP You may program the SVC instruction unconditionally across the rungs. This is the normal programming technique for the SVC instruction.

MSG - Message

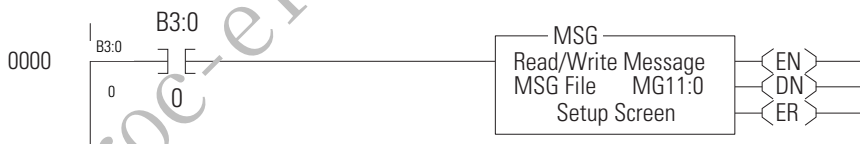


Instruction Type: output

Execution Time for the MSG Instruction

Controller	Rung Condition	When Rung Is:	
		True	False
MicroLogix 1400	Steady State True	2.5670 µs	0.7310 µs
	False-to-True Transition for Reads	48.1677 µs	0.8510 µs
	False-to-True Transition for Writes	58.8510 µs	0.9177 µs

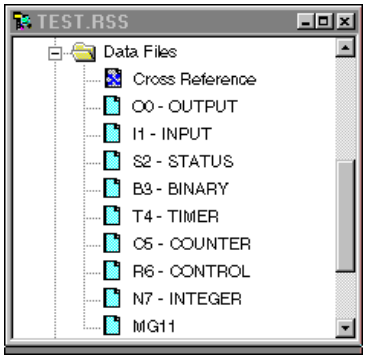
Any preceding logic on the message rung must be solved true before the message instruction can be processed. The example below shows a message instruction.



If B3:0 is on (1), the MSG rung is true, and MG11:0 is not already processing a message; then MG11:0 is processed. If one of the four buffers is available, the message and its associated data are processed immediately.

TIP How quickly the message is actually sent to the destination device depends on a number of issues, including the selected channel's communication protocol, the baud rate of the communications port, the number of retries needed (if any), and the destination device's readiness to receive the message.

The Message Element



The MSG instruction built into the controller uses a MG data file to process the message instruction. The MG data file, shown at left, is accessed using the MG prefix. Each message instruction utilizes an element within a MG data file. For example, MG11:0 is the first element in message data file 11.

Message File Sub-Elements

Each MSG instruction must use a unique Element in a MSG File. The MSG element for each MSG instruction holds all of the parameters and status information for that particular MSG instruction.

Each MSG File Element consists of Sub-Elements 0...24 as shown in the following table.

Message File Elements

Sub-Element	Name	Description	Parameter	Size	User Program Access[2]
0 to 1		Reserved		Word	read only
2		Messaging Type: 0 (for PCCC), 1 (for CIP), 2 (for Modbus Master)		Word	read only
3		for PCCC Messaging: bits 07-00 (CMD code), bits 15-08 (FNC code) for CIP: bits 07-00 (Service Code), bits 15-08 (Supplemental Object Path Data Count) for Modbus Master: bits 07-00 (Function Code), bits 15-08 (Reserved)	derived	Word	read only
4	MG11:0.IA	Internal Physical Address		Word	read only
5	MG11:0.RBL	PCCC: Remote Bridge Link ID Modbus Master: not used	Y	Word	read only
6	MG11:0.LBN	PCCC: Local Bridge Node Address Modbus Master: not used	Y	Word	read only
7	MG11:0.RBN	PCCC: Remote Bridge Node Address Modbus Master: not used	Y	Word	read only
8	MG11:0.CHN	Channel: bits 07-00 (0 for Channel 0,1 for Channel 1) Slot: bits 15-08 (Not used)	Y	Word	read/write
9	MG11:0.NOD	Target Node Number	Y	Word	read/write
10	MG11:0.MTO	Message timeout setting or preset in seconds	Y	Word	read/write
11	MG11:0.NB	PCCC: Number of bytes to read/write Modbus Master: Number of Modbus elements to read/write		Word	read only
12	MG11:0.TFT	Target Location information (See tables on page 343 for options)	Y	Word	read only
13	MG11:0.TFN		Y	Word	read/write
14	MG11:0.ELE		Y	Word	read/write
15			Y	Word	read only
16		Control bits (See Control Bits table on page 345 for details)	N	16-bits	read/write
17		Status bits and Range parameter (See table on page 345 for details)	Mixed	16-bits	read only
18	MG11:0.ERR	Error code (See Error Codes on page 404)	N	Word	read only
19		Time since message started in seconds	N	Word	read only
20		Reserved		Word	read only

Message File Elements

Sub-Element	Name	Description	Parameter	Size	User Program Access[2]
21		Internal message start time in seconds	N	Word	read only
22		bits 15-08: Internal ERR Error Code bits 07-00: Internal Fail Code Note: When CIP sub-system cannot send a message due to some reason or reply contains error code, error code is displayed via MSG instruction. When messaging through CIP communication and non-zero Status Code was received, the low byte is 0xE0 and the high byte of this sub-element contains detailed Status Code returned by CIP reply for MicroLogix 1400 Note: When SMTP sub-system cannot send an email due to some reason, error code is shown via MSG instruction. When messaging through SMTP communication and the low byte is 0xDD, the high byte of this sub-element contains detailed Fail Code returned by the SMTP sub-system for MicroLogix 1400.	N	Word	read only
23		Extended Status Error Code from expansion I/O communications module.			
24[1]		Supplemental Routing Path Data Address: bits 7…0: Starting Element, bits 15…8: File Number			

(1) Channel 1 only. Refer to the Routing Information File on page 375.

(2) User access refers to user program access (MSG File word or bit used as an operand for an instruction in a ladder program) or access via Comms while in any mode other than download (via Programming Software or Memory Module).

The Target file information contained in Sub-Elements 12...15 of the MSG File Element depend upon the message type, as shown in the tables below.

Message File Target Location Information

Target Device = 485 CIF

Sub-Element	Name	Description	Parameter	Size	User Program Access
12	MG11:0.TFT	Reserved	Y	Word	read only
13	MG11:0.TFN	Target File Number	Y	Word	read/write
14	MG11:0.ELE	Offset in elements into CIF	Y	Word	read/write
15		Reserved	Y	Word	read only

Message File Target Location Information

Target Device = 500CPU or PLC

Sub-Element	Address	Description	Parameter	Size	User Program Access
12	MG11:0.TFT	Target File Type	Y	Word	read only

Message File Target Location Information

Target Device = 500CPU or PLC

Sub-Element	Address	Description	Parameter	Size	User Program Access
13	MG11:0.TFN	Target File Numb(1)	Y	Word	read/write
14	MG11:0.ELE	Target File Element Number for B, S, N, F, T, C, R, L, ST and RTC files; or Target File Slot Number for O and I files.	Y	Word	read/write
15		Target File Element Number for O and I files. Set to zero for any file other than O or I.	Y	Word	read only

(1) The file number for RTC function files is set to 0 by the programming software.

Message File Target Location Information

Target Device = Modbus Device

Sub-Element	Name	Description	Parameter	Size	User Program Access
12	MG11:0.TFT	starting bit address for coils and inputs	Y	Word	read only
13	MG11:0.TFN	Modbus Target Data Address - 1	Y	Word	read/write
14	MG11:0.ELE	Reserved	Y	Word	read/write
15		Reserved	Y	Word	read only

Message File Target Location Information

Target Device = CIP Generic

Sub-Element	Name	Description	Parameter	Size	User Program Access
12	MG11:0.TFT	Target Class	Y	Word	read only
13	MG11:0.TFN	Target Instance	Y	Word	read/write
14	MG11:0.ELE	CIP Send Data Count	Y	Word	read/write
15		Internal Physical Address of CIP Send Data Table Address operand	Y	Word	read only

The Control Bits, Sub-Element 16, of the MSG File Element are defined below:

Message File Sub-Element 16 - Control Bits

Bit	Address	Description	Parameter	Size	User Program Access
15	MG11:0.0/EN	Enable 1=MSG enabled 0=MSG not enabled	N	bit	read/write
9…14		Reserved	N	bit	read/write
8	MG11:0.0/TO	Time Out 1=MSG time out by user 0=no user MSG time out	N	bit	read/write
1…7		Reserved	N	bit	read/write
1	ML11:0.0/UC	Unconnected Message For Channel 1, 1=Unconnected type 0=Connected type	N	bit	read/write
0	MG11:0.0/BK	Break Connection For Channel 1, 1=MSG Connection closed by user 0=MSG Connection not closed by user	N	bit	read/write

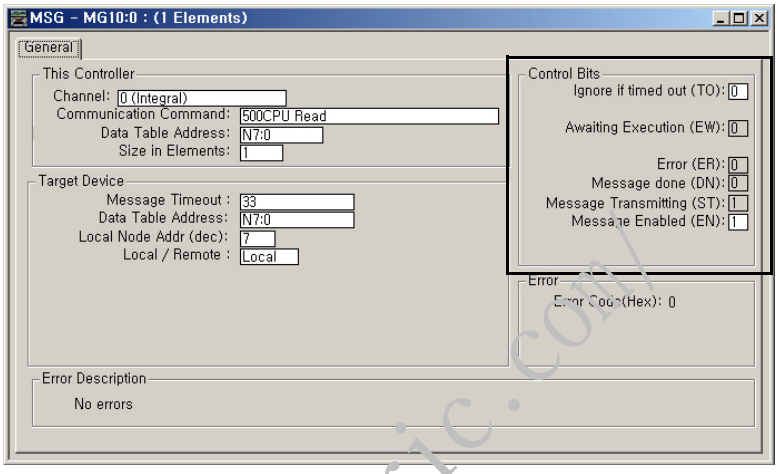
The Status Bits, Sub-Element 17, of the MSG File Element are defined below.

Message File Sub-Element 17 - Status Bits

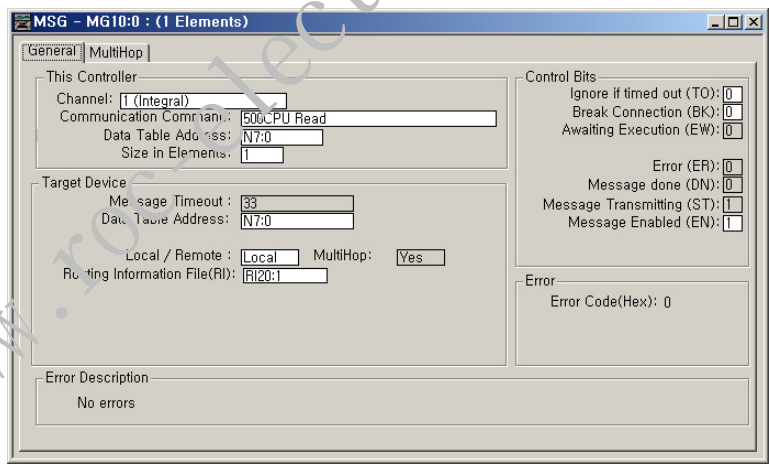
Bit	Address	Description	Parameter	Size	User Program Access
15		Reserved	N	bit	read only
14	MG11:0.0/ST	Start: 1 = MSG transmitted and acknowledged by target device 0 = MSG has not been received by target	N	bit	read only
13	MG11:0.0/DN	Done 1 = MSG completed successfully 0 = MSG not complete	N	bit	read only
12	MG11:0.0/ER	Error 1 = error detected 0 = no error detected	N	bit	read only
11		Reserved	N	bit	read only
10	MG11:0.0/EW	Enabled and Waiting: 1=MSG Enabled and Waiting 0=MSG not Enabled and Waiting	N	bit	read only
1…9		Reserved	N	bit	read only
0	MG11:0.0/R	For PCCC Messaging: Range (1 = Local, 0 = Remote) For Modbus Messaging: Range (1 = Local)	Y	bit	read only

"Control Bits" Parameters

Channel 0 Setup Screen



Channel 1 Setup Screen



Ignore if Timed Out (TO)

Address	Data Format	Range	Type	User Program Access
MG11:0/TO	Binary	On or Off	Control	Read / Write

The Timed Out Bit (TO) can be set in your application to remove an active message instruction from processor control. You can create your own timeout routine by monitoring the EW and ST bits to start a timer. When the timer times out, you can set the TO bit, which removes the message from the system. The controller resets the TO bit the next time the associated MSG rung goes from false to true.

An easier method is to use the message timeout variable described on page 363, because it simplifies the user program. This built-in timeout control is in effect whenever the message timeout is non-zero. It defaults to 5 seconds for channel 0, so unless you change it, the internal timeout control is automatically enabled.

When the internal timeout is used and communications are interrupted, the MSG instruction will timeout and error after the set period of time expires. This allows the control program to retry the same message or take other action, if desired.

To disable the internal timeout control, enter zero for the MSG instruction timeout parameter. If communications are interrupted, the processor waits indefinitely for a reply. If an acknowledge (ACK) is received, indicated by the ST bit being set, but the reply is not received, the MSG instruction appears to be locked up, although it is actually waiting for a reply from the target device.

Enable (EN)

Address	Data Format	Range	Type	User Program Access
MG11:0/EN	Binary	On or Off	Control	Read / Write

The Enable Bit (EN) is set when rung conditions go true and the MSG is enabled. The MSG is enabled when the command packet is built and put into one of the MSG buffers, or the request is put in the MSG queue. It remains set until the message transmission is completed and the rung goes false. You may clear this bit when either the ER or DN bit is set in order to re-trigger a MSG instruction with true rung conditions on the next scan.

IMPORTANT Do not set this bit from the control program.

Enabled and Waiting (EW)

Address	Data Format	Range	Type	User Program Access
MG11:0/EW	Binary	On or Off	Status	Read Only

The Enabled and Waiting Bit (EW) is set after the enable bit is set and the message is in the buffer (not in the queue) and waiting to be sent. The EW bit is cleared after the message has been sent and the processor receives acknowledgement (ACK) from the target device. This is before the target device has processed the message and sent a reply.

Error (ER)

Address	Data Format	Range	Type	User Program Access
MG11:0/ER	Binary	On or Off	Status	Read Only

The Error Bit (ER) is set when message transmission has failed. An error code is written to the MSG File. The ER bit and the error code are cleared the next time the associated rung goes from false to true.

Done (DN)

Address	Data Format	Range	Type	User Program Access
MG11:0/DN	Binary	On or Off	Status	Read Only

The Done Bit (DN) is set when the message is transmitted successfully. The DN bit is cleared the next time the associated rung goes from false to true.

Start (ST)

Address	Data Format	Range	Type	User Program Access
MG11:0/ST	Binary	On or Off	Status	Read Only

The Start Bit (ST) is set when the processor receives acknowledgment (ACK) from the target device. The ST bit is cleared when the DN, ER, or TO bit is set.

The DF1 Radio Modem and Modbus RTU Master protocols do not have acknowledgements. When the channel that the MSG instruction is being initiated on is configured for either of these two drivers, the Start Bit (ST) is set when the message has been successfully transmitted.

UnConnected(UC)

Address	Data Format	Range	Type	User Program Access
MG11:0/UC	Binary	On or Off	Control	Read / Write

When the UnConnected bit is set, an unconnected type message will be generated whenever that EtherNet/IP MSG instruction is triggered. When unconnected Ethernet/IP messaging is used, there is no establishment process of CIP 3 connection (Forward Open, Forward Close, ...). This is useful for slow networks to minimize traffic (for example, through cellular modems).

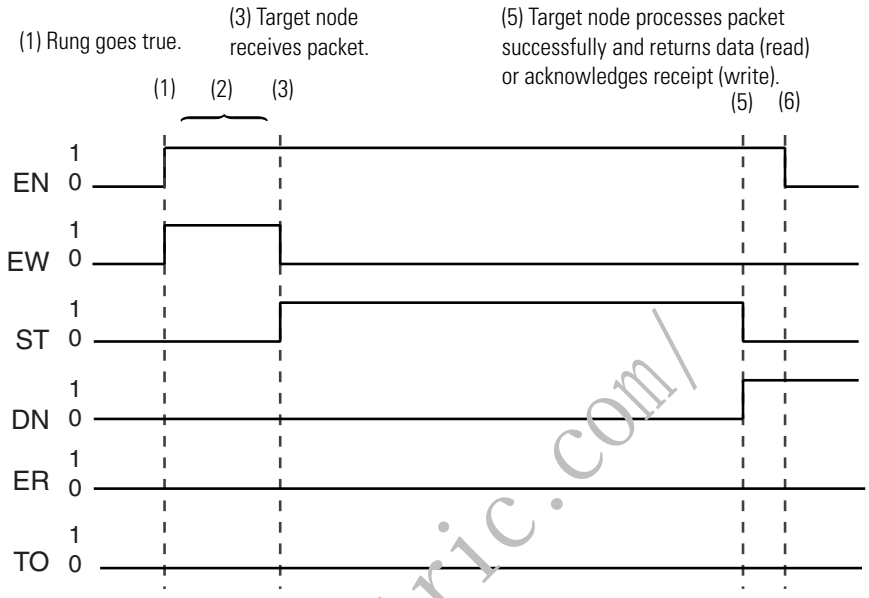
Break Connection (BK))

Address	Data Format	Range	Type	User Program Access
MG11:0/BK	Binary	On or Off	Control	Read / Write

When the BreaK bit is set, the Ethernet TCP connection will be closed after the MSG instruction has completed. If cleared to 0, the Ethernet TCP connection will remain connected after the MSG instruction has completed. If set to 0 value, the Ethernet/IP connection will remain even if the MSG instruction sent successfully.

Timing Diagram for the MSG Instruction

The following section describes the timing diagram for a message instruction.



1. If there is room in any of the four active message buffers when the MSG rung becomes true and the MSG is scanned, the EN and EW bits for this message are set. If this is a MSG write instruction, the source data is transferred to the message buffer at this time.

   (Not shown in the diagram.) If the four message buffers are in use, the message request is put in the message queue and only the EN bit is set. The message queue works on a first-in, first-out basis that allows the controller to remember the order in which the message instructions were enabled. When a buffer becomes available, the first message in the queue is placed into the buffer and the EW bit is set (1).

   TIP The control program does not have access to the message buffers or the communications queue.

   Once the EN bit is set (1), it remains set until the entire message process is complete and either the DN, ER, or TO bit is set (1). The MSG Timeout period begins timing when the EN bit is set (1). If the timeout period expires before the MSG instruction completes its function, the ER bit is set (1), and an error code (37H) is placed in the MG File to inform you of the timeout error.

2. At the next end of scan, REF, or SVC instruction, the controller determines if it should examine the communications queue for another instruction. The controller bases its decision on the state of the channel's Communication Servicing Selection (CSS) and Message Servicing Selection (MSS) bits, the network communication requests from other nodes, and whether previous message instructions are already in progress. If the controller determines that it should not access the queue, the message instruction remains as it was. Either the EN and EW bits remain set (1) or only the EN bit is set (1) until the next end of scan, REF, or SVC instruction.

   If the controller determines that it has an instruction in the queue, it unloads the communications queue entries into the message buffers until all four message buffers are full. If an invalid message is unloaded from the communications queue, the ER bit in the MG file is set (1), and a code is placed in the MG file to inform you of an error. When a valid message instruction is loaded into a message buffer, the EN and EW bits for this message are set (1).

   The controller then exits the end of scan, REF, or SVC portion of the scan. The controller's background communication function sends the messages to the target nodes specified in the message instruction. Depending on the state of the CSS and MSS bits, you can service up to four active message instructions per channel at any given time.

3. If the target node successfully receives the message, it sends back an acknowledge (ACK). The ACK causes the processor to clear (0) the EW bit and set (1) the ST bit. The target node has not yet examined the packet to see if it understands your request.

   Once the ST bit is set (1), the controller waits for a reply from the target node. The target node is not required to respond within any given time frame.

   TIP If the Target Node faults or power cycles during the message transaction, you will never receive a reply. This is why you should use a Message Timeout value in your MSG instruction.

4. Step 4 is not shown in the timing diagram. If you do not receive an ACK, step 3 does not occur. Instead, either no response or a negative acknowledge (NAK) is received. When this happens, the ST bit remains clear (0).

   No response may be caused by:
   - the target node is not there
   - the message became corrupted in transmission
   - the response was corrupted in response transmission

   A NAK may be caused by:
   - target node is busy
   - target node received a corrupt message
   - the message is too large

   When a NAK occurs, the EW bit is cleared (0), and the ER bit is set (1), indicating that the message instruction failed.

5. Following the successful receipt of the packet, the target node sends a reply packet. The reply packet contains one of the following responses:
   - successful write request.
   - successful read request with data
   - failure with error code

   At the next end of scan, REF, or SVC instruction, following the target node's reply, the controller examines the message from the target device. If the reply is successful, the DN bit is set (1), and the ST bit is cleared (0). If it is a successful read request, the data is written to the data table. The message instruction function is complete.

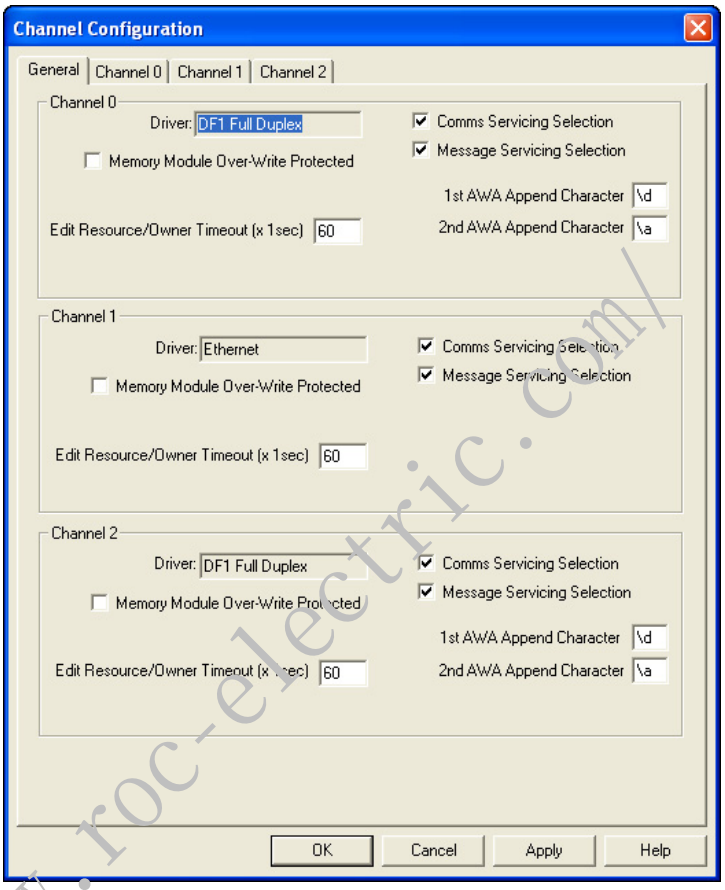
   If the reply is a failure with an error code, the ER bit is set (1), and the ST bit is cleared (0). The message instruction function is complete.

6. If the DN or ER bit is set (1) and the MSG rung is false, the EN bit is cleared (0) the next time the message instruction is scanned.

See MSG Instruction Ladder Logic on page 353 for examples using the message instruction.

Communication Servicing Selection and Message Servicing Selection

The following screen shows the channel configuration window for Communication Servicing Selection and Message Servicing Selection.



Communication Servicing Selection

Use the check box to enhance communication capability. If the check box is unchecked, communication throughput (and scan time) will increase.

When Communications Servicing Selection is checked, at the next execution of a Service Communications (SVC) instruction, I/O Refresh (REF) instruction, or when it performs Communications Servicing, whichever occurs first, commands/replies are processed as follows:

- One incoming channel command;
- One incoming channel message reply;
- One outgoing channel message on the overflow queue.

When the Communications Servicing Selection bit is unchecked, at the next execution of a Service Communications (SVC) instruction, I/O Refresh (REF) instruction, or when it performs Communications Servicing, whichever occurs first, commands/replies are processed as follows:

- One incoming channel command;

- (conditional) If the Message Servicing Selection is clear (not checked) first, all incoming channel message replies; then all outgoing channel messages on the overflow queue. If the Message Servicing Selection is set (checked), First the incoming channel message reply; then one outgoing channel message on the overflow queue.
- All remaining incoming channel commands.

Message Servicing Selection

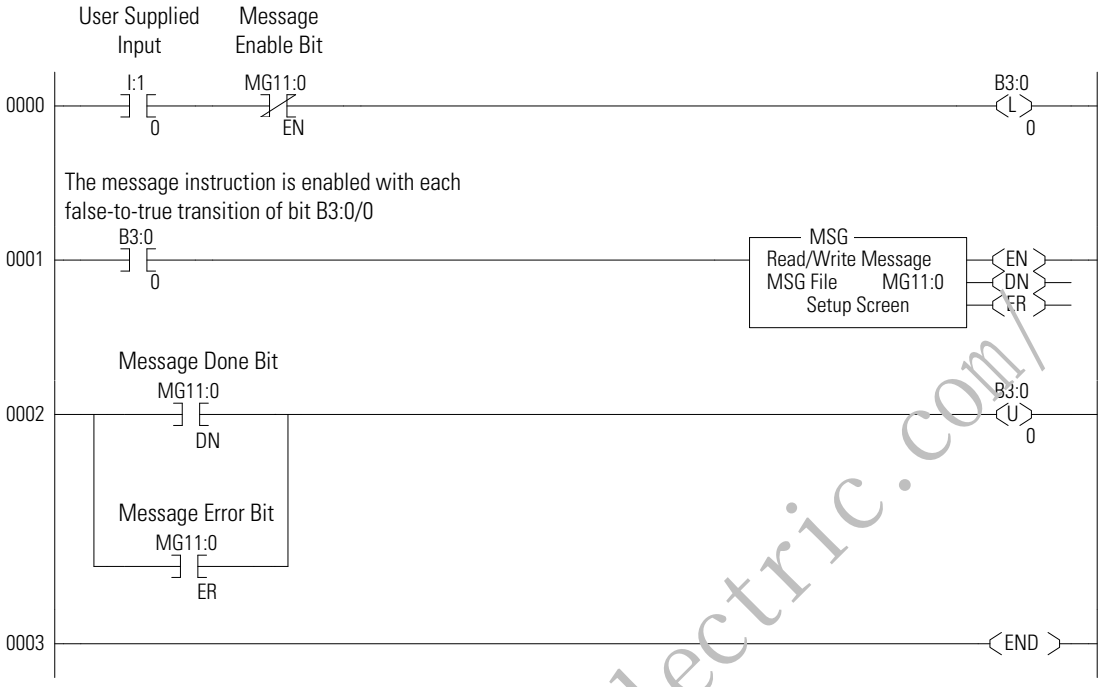
Use this check box to enhance communication capability. If this is checked and the Communication Servicing Selection check box is not checked the MSG functionality throughput (and scan time) will increase.

- When this check box is checked, the controller services one outgoing channel MSG instruction at the next Service Communications (SVC) instruction, I/O Refresh (REF) instruction, or when it performs Communications Servicing. When this check box is clear (unchecked), the controller services all outgoing channel MSG instructions at the next Service Communications (SVC) instruction, I/O Refresh (REF) instruction, or when it performs Communications Servicing.
- The Message Servicing Selection (channel) bit is applied by the controller when the Communications Servicing Selection (channel) bit is clear (unchecked).

MSG Instruction Ladder Logic

Enabling the MSG Instruction for Continuous Operation

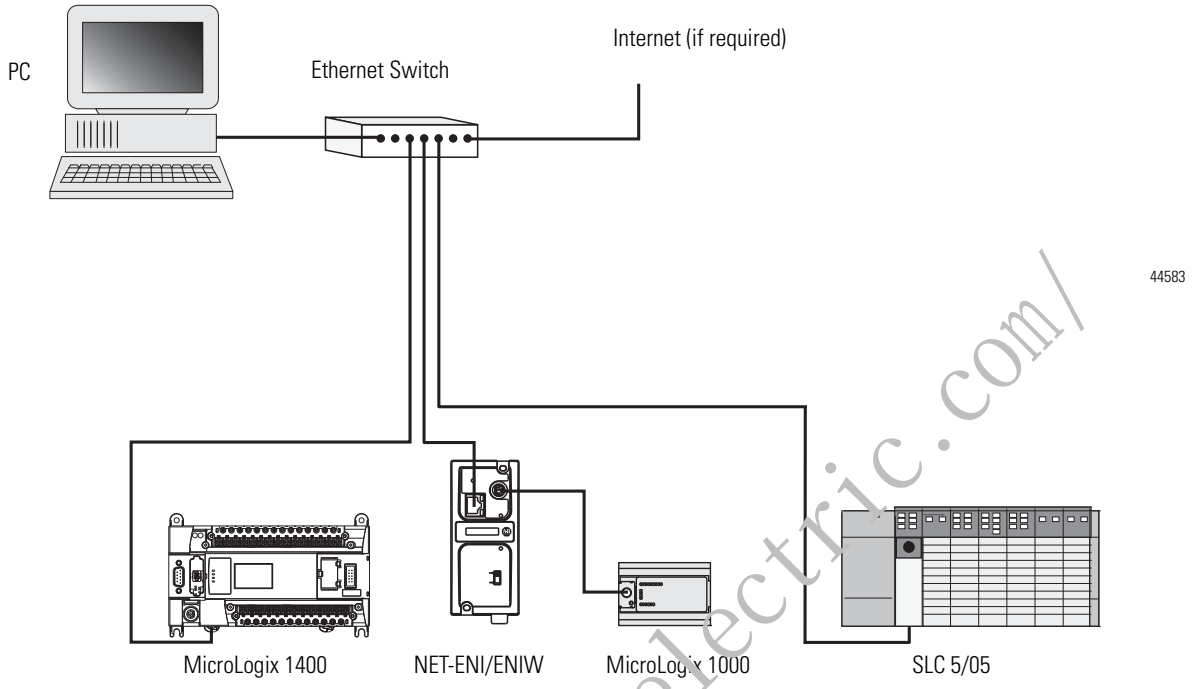
The message instruction is enabled during the initial processor program scan and each time the message completes. For example, when the DN or ER bit is set.



Enabling the MSG Instruction Via User Supplied Input

This is an example of controlling when the message instruction operates. Input I:1/0 could be any user-supplied bit to control when messages are sent. Whenever

I:1/0 is set and message MG11:0 is not enabled, the message instruction on rung 0001 is enabled.



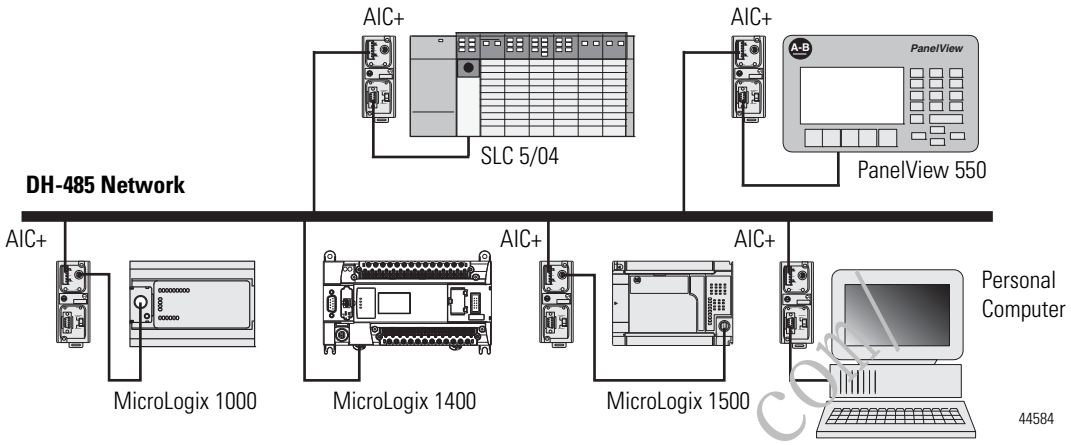
Local Messages

The controller is capable of communicating using local or remote messages. With a local message, all devices are accessible without a separate device acting as a bridge. Different types of electrical interfaces may be required to connect to the network, but the network is still classified as a local network. Remote messages use a remote network, where devices are accessible only by passing or routing through a device to another network. Remote networks are discussed on page 379.

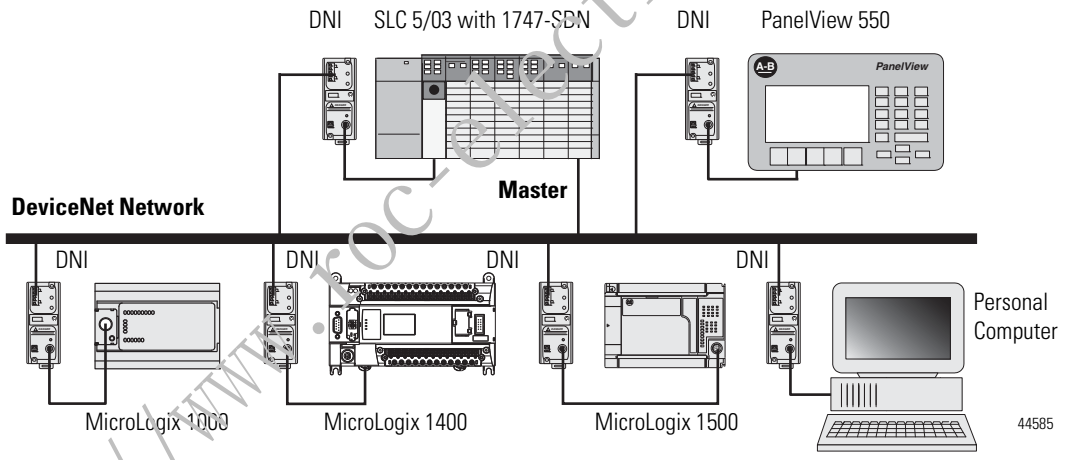
Local Networks

The following three examples represent different types of local networks.

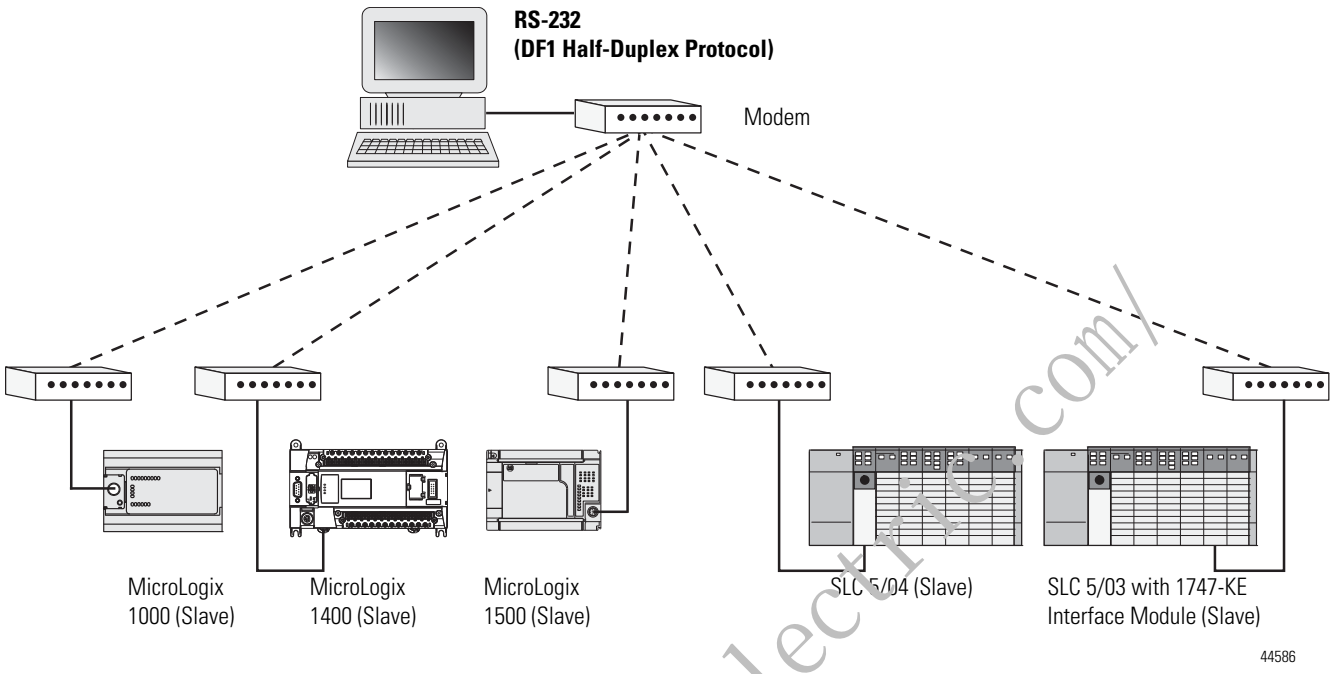
Example 1 - Local DH-485 Network



Example 2 - Local DeviceNet Network with DeviceNet Interface (1761-NET-DNI)



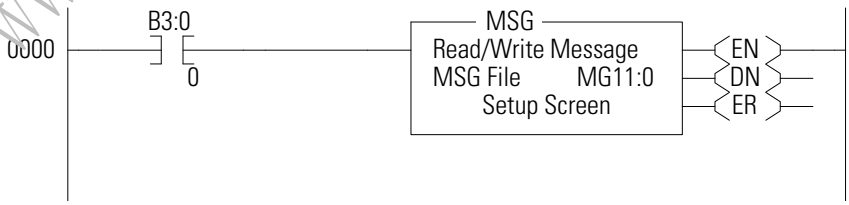
Example 3 - Local DF1 Half-Duplex Network



Configuring a Local Message

Message Setup Screen

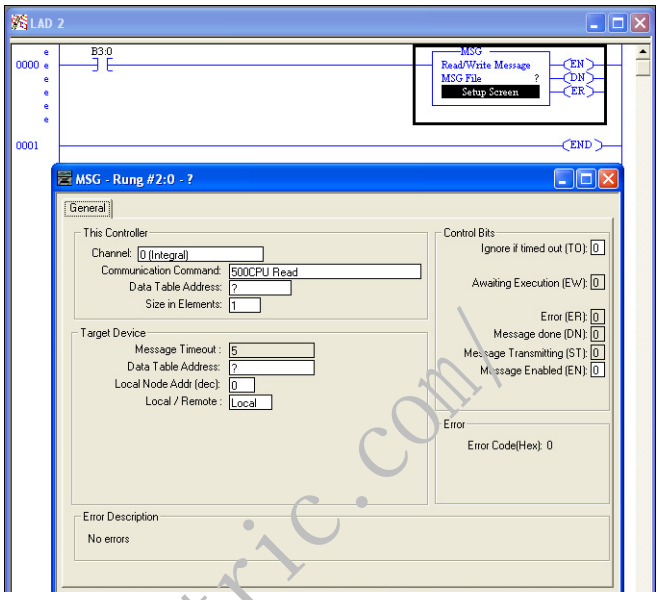
The rung below shows a MSG instruction preceded by conditional logic. Access the message setup screen by double-clicking Setup Screen.



The RSLogix Message Setup Screen is shown below. This screen is used to setup "This Controller", "Target Device", and "Control Bits". Descriptions of each of the elements follow.

1.The Edit mark should be deleted in rung 0 of the ladder program

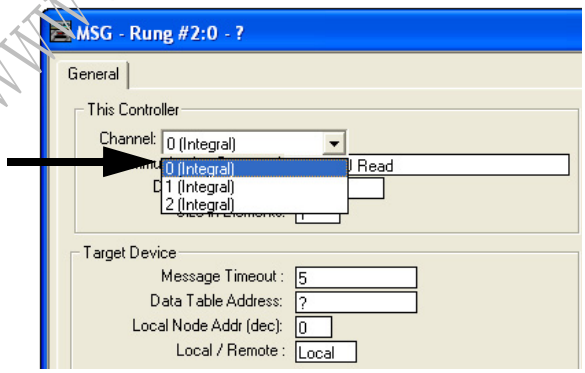
2. Use blue or gray for color of Window title.



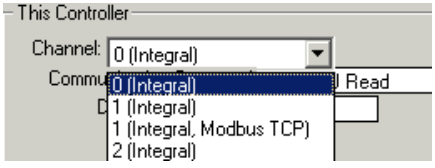
"This Controller" Parameters

Channel

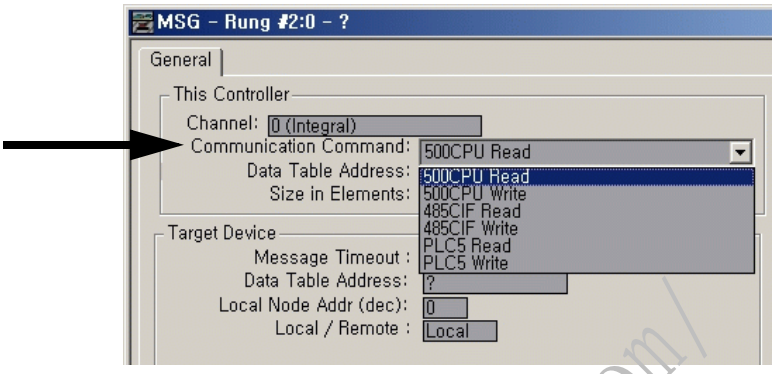
The MicroLogix 1400 supports messaging on all three channels. Channel 0 is the RS-232/RS-485 port, Channel 1 is the Ethernet port and Channel 2 is the RS-232 port.



If either Channel 0 (Integral) or Channel 2 (Integral) is selected and that channel is already configured for Modbus RTU Master, or if Channel 1 (Integral, Modbus TCP) is selected, then the next line with display 'Modbus Command.



Communication Command



The controller supports six different types of communications commands. If the target device supports any of these command types, the controller should be capable of exchanging data with the device. Supported commands include:

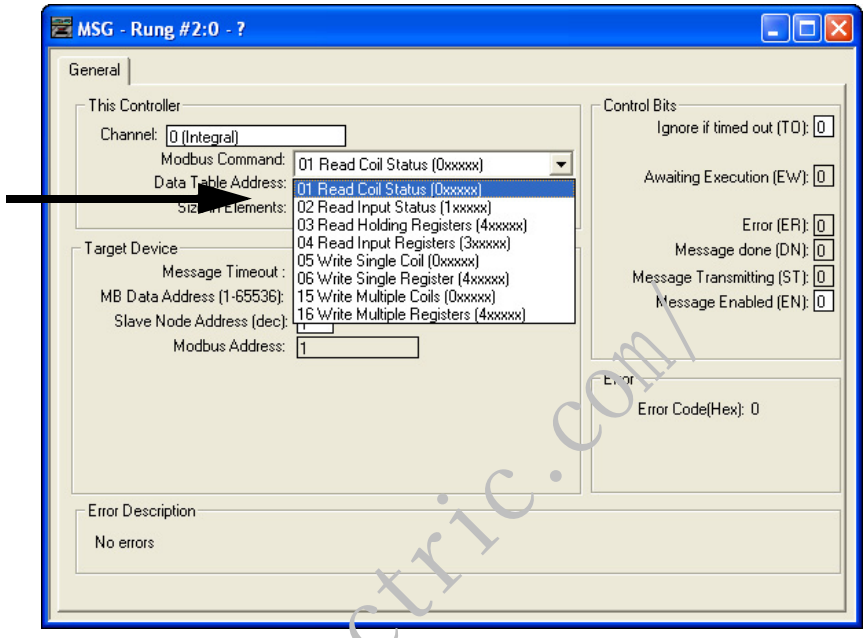
Communication Command Types

Communication Command	Description	Used For
500CPU Read	The target device is compatible with and supports the SLC 500 command set (all MicroLogix controllers).	reading data
500CPU Write	The target device is compatible with and supports the SLC 500 command set (all MicroLogix controllers).	sending data
485CIF Read(1)	The target device is compatible with and supports the 485CIF (PLC2).	reading data
485CIF Write(1)	The target device is compatible with and supports the 485CIF (PLC2).	sending data
PLC5 Read	The target device is compatible with and supports the PLC5 command set.	reading data
PLC5 Write	The target device is compatible with and supports the PLC5 command set.	sending data

(1) See Important note below.

IMPORTANT The Common Interface File (CIF) in the MicroLogix 1100, 1200, 1400, 1500, and SLC 500 processors is File 9. The CIF in the MicroLogix 1000 controller is Integer File 7.

Modbus Command



The controller supports eight Modbus commands. If the target device supports any of these Modbus command types, the controller should be capable of exchanging data with the device. Supported Modbus commands include:

Modbus Command Types

Modbus Command	Used For
01 Read Coil Status	reading bits
02 Read Input Status	reading bits
03 Read Holding Registers	reading words
04 Read Input Registers	reading words
05 Write Single Coil	writing 1 bit
06 Write Single Register	writing 1 word
15 Write Multiple Coil	writing multiple bits
16 Write Multiple Registers	writing multiple words

Data Table Address

This variable defines the starting address in the local controller. Valid file types for the Data Table Address are shown below:

Message Read	Message Write
Bit (B) Timer (T) Counter (C) Control (R) Integer (N) Floating Point (F)[1] Long Word (L) String (ST) ASCII (A)[2]	Output (O) Input (I) Bit (B) Timer (T) Counter (C) Control (R) Integer (N) Floating Point (F)[1] Long Word (L) String (ST) ASCII (A)[2] Real-Time Clock (RTC)[3]

(1) Message Type must be 500CPU or PLC5. The Local File Type and Target File Type must both be Floating Point.

(2) ASCII type not supported by MicroLogix 1400 Series A.

(3) 500CPU write RTC-to-Integer or RTC-to-RTC only.

TIP Only Bit (B) and Integer (N) file types are valid for Modbus Command messages. Modbus bit commands require a starting bit address for the Data Table Address.

Floating Point (F) and Long (L) file types are valid for Modbus Command messages for Holding Registers (commands 03, 06 and 16) when Data is configured for 32 bit.

Size in Elements

This variable defines the amount of data (in elements) to exchange with the target device.

The maximum amount of data that can be transferred via a MSG instruction is 103 words (120 words for Modbus commands) and is determined by the destination data type. The destination data type is defined by the type of message: read or write.

- For Read Messages: When a read message is used, the destination file is the data file in the local or originating processor.

  TIP Input, output, string, and RTC file types are not valid for read messages.

- For Write Messages: When a write message is used, the destination file is the data file in the target processor.

The maximum number of elements that can be transmitted or received are shown in the following table. You cannot cross file types when sending messages. For example, you cannot read a timer into an integer file and you cannot write counters to a timer file. The only exceptions to this rule are that:

- long integer data can be read from or written to bit or integer files, and

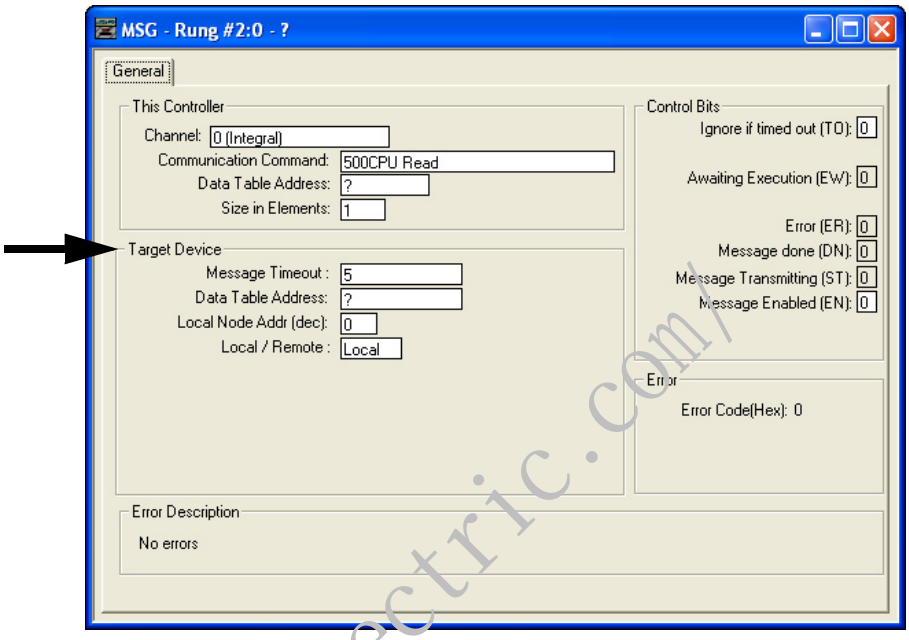
- RTC files can be written to integer files.

TIP The table below is not intended to illustrate file compatibility, only the maximum number of elements that can be exchanged in each case.

Message Type	File Type	Element Size	Maximum Number of Elements per Message
485CIF	O, I, B, N, A	1-word	103
	L	2-word	51
	T, C, R	3-word	34
	ST(1)	42-word	2 (write only)
500CPU	O, I, B, N, A	1-word	103
	F, L	2-word	51
	T, C, R	3-word	34
	RTC	8-word	1 (write only)
	ST	42-word	2
PLC5	O, I, B, N, A	1-word	103
	F(1), L	2-word	51
	T	5-word	20
	ST	42-word	1
Modbus Commands	B, N (command 5)	1-bit	1
	B, N (command 6)	1-word	1
	B, N (commands 1, 2, and 15)	1-bit	1920 Modbus bit elements (120 words) (Commands 1 and 2 are read only, 15 is write only.)
	B, N (commands 3, 4, and 16)	multi-register	120 Modbus register elements (120 words) (Commands 3 and 4 are read only, 16 is write only.)

(1) Message Type must be 500CPU or PLC5. The Local File Type and Target File Type must both be Floating Point.

"Target Device" Parameters



Message Timeout

This value defines how long, in seconds, the message instruction has to complete its operation once it has started. Timing begins when the false-to-true rung transition occurs, enabling the message. If the timeout period expires, the message errors out. The default value is 5 seconds (2 seconds for Modbus commands). The maximum timeout value is 255 seconds.

Message Timeout for any MicroLogix 1400 channel 1 MSG can not be modified in the Ethernet Message Setup dialog box. It is assigned by the processor and is determined by adding the Channel 1 MSG Connection Timeout to the MSG Reply Timeout, then adding 15 seconds. This value can be modified by changing one or both of the timeout values in the channel configuration screen for channel 1. The modified message timeout applies to all Ethernet MSG instructions.

> MSG timeout for channel 1 = MSG Connection Timeout + MSG Reply Timeout + 15 (seconds)

If the message timeout is set to zero, the message instruction will never timeout. Set the Time Out bit (TO = 1) to flush a message instruction from its buffer if the destination device does not respond to the communications request.

Data Table Address/Offset

This variable defines the starting address in the target controller. The data table address is used for a 500CPU and PLC5 type messages. A valid address is any valid, configured data file within the target device whose file type is recognized by the controller. Valid combinations are shown below:

Message Type	Local File Type	Target File Type
500CPU and PLC5	O, I, B, N, F(1), L	O, I, S, B, N, F(1), L
	T	T
	C	C
	R	R
	RTC(2)	N, RTC
500CPU, PLC5 and 485 CIF	ST	ST

(1) Message Type must be 500CPU or PLC5. The Local File Type and Target File Type must both be Floating Point.

(2) 500CPU write RTC-to-Integer or RTC-to-RTC only.

The data table offset is used for 485CIF type messages. A valid offset is any value in the range 0...255 and indicates the word or byte offset into the target's Common Interface File (CIF). The type of device determines whether it is a word or byte offset. MicroLogix controllers and SLC processors use word offset; PLC-5 and Logix processors use byte offset.

Modbus - MB Data Address (1...65536)

Modbus addressing is limited to 16 bits per memory group, each with a range of 1...65,536. There are four memory groups, one for each function:

- coils (generally addressed as **0**xxxx)
- contacts (**1**xxxx)
- input registers (**3**xxxx)
- holding registers (**4**xxxx)

Coils and contacts are addressed at the bit level. Coils are outputs and can be read and written. Contacts are inputs and are read-only.

Input registers and holding registers are addressed at the word level. Input registers are generally used for internally storing input values. They are read-only. Holding registers are general purpose and can be both read and written.

The most significant digit of the address is considered a prefix, and does not get entered into the MB Data Address field when configuring the message instruction.

When the message is sent, the address is decremented by 1 and converted into a 4-character hex number to be transmitted via the network (with a range of

0-FFFFh); the slave increments the address by 1, and selects the appropriate memory group based on the Modbus function.

TIP Modbus protocol may not be consistently implemented in all devices. The Modbus specification calls for the addressing range to start at 1; however, some devices start addressing at 0.

The Modbus Data Address in the Message Setup Screen may need to be incremented by one to properly access a Modbus slave's memory, depending on that slave's implementation of memory addressing.

Local/Slave Node Address

This is the destination device's node number if the devices are on a DH-485, DeviceNet (using 1761-NET-DNI), DF1, or Modbus network.

TIP To initiate a broadcast message on a DH-485, DF1 Half-Duplex, or DF1 Radio Modem network, set the local node address to -1.

To initiate a broadcast message on a Modbus network, set the slave node address to 0. Do not initiate more than one Modbus broadcast message at a time. When sequentially triggering multiple Modbus broadcast messages, insert at least 10 msec. delay in between each message.

Local/Remote

This variable defines the type of communications that is used. Always use local when you need point-to-point communications via DF1 Full-Duplex or network communications such as Ethernet/IP, DeviceNet (using 1761-NET-DNI), DF1 Half-Duplex, or DF1 Radio Modem. For DH-485, use local if the target node is on the same DH-485 network as this controller, or remote if the path to the target node goes through one or more communication bridges.

Local Messaging Examples

Five examples of local messaging are shown in this section:

- 500CPU message type
- 485CIF message type
- PLC5 message type
- Modbus RTU Message type
- EtherNet/IP Message type
- Write Message type with ST data file

A summary of the message instruction configuration parameters is shown in the following table.

Message Instruction Configuration Parameters

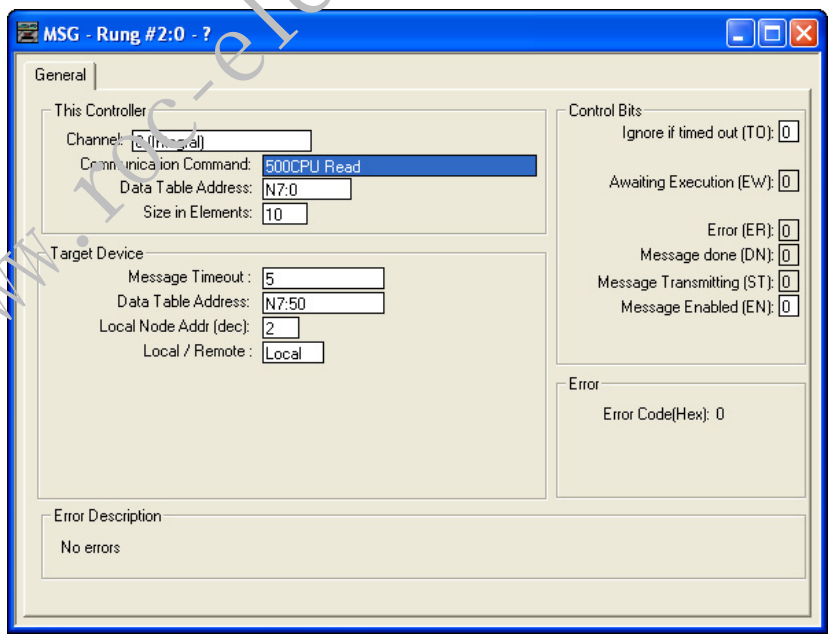
Parameter		Description
This Controller	Channel	Identifies the communication channel. Channel 0, Channel 1, Channel 1 Modbus TCP, or Channel 2
	Communication Command (500CPU, 485CIF, and PLC5 message types)	Specifies the type of message. Valid types are: • 500CPU Read • 500CPU Write • 485CIF Read • 485CIF Write • PLC5 Read • PLC5 Write
	Modbus Command	Specifies the type of message. Valid types are: • 01 Read Coil Status • 02 Read Input Status • 03 Read Holding Registers • 04 Read Input Registers • 05 Write Single Coil • 06 Write Single Register • 15 Write Multiple Coils • 16 Write Multiple Registers
	Data Table Address	For a Read, this is the starting address which receives data. Valid file types are B, T, C, R, N, F, L, A and ST (for Modbus command, B, N, F and L only).
		For a Write, this is the starting address which is sent to the target device. Valid file types are O, I, B, T, C, R, N, F, L, ST and RTC (for Modbus command, B, N, F and L only).
	Size in elements	Defines the length of the message in elements. • 1-word elements; valid size: 1…103. • 2-word elements; valid size: 1…51. • 8-word RTC elements; valid size: 1 • 42-word String elements; valid size 1…2 • Timer (500CPU and 485CIF), Counter, and Control elements; valid size: 1…34. • PLC-5 Timer elements; valid size: 1…20 • Modbus bit elements: 1…1920 • Modbus register elements: 1…120

Message Instruction Configuration Parameters

Parameter		Description
Target Device	Message Timeout	Defines the amount of time the controller waits for the reply before the message errors. A timeout of 0 seconds means that the controller waits indefinitely for a reply. Valid range is from 0…255 seconds.
	Data Table Address (500CPU and PLC5 message types)	For a Read, this is the address in the target processor which is to return data. Valid file types are O, I, S, B, T, C, R, N, F, L and ST.
		For a Write, this is the starting address in the target processor which receives data. Valid file types are O, I, S, B, T, C, R, N, F, L, ST and RTC
	Data Table Offset (485CIF message types)	This is the word offset value in the common interface file (byte offset for PLC device) in the target processor, which is to send the data.
	MB Data Address	Specifies the Modbus address in the target device. Valid range is from 1…65,536.
	Local Slave Node Address	Specifies the node number of the device that is receiving the message. Valid range is 0…31 for DH-485 protocol, 0…254 for DF1 protocol, 0…63 for DeviceNet, or 0…247 for Modbus.
	Local/Remote	Specifies whether the message is local or remote. (Modbus messages are local only.)

Example 1 - Local Read from a 500CPU

Message Instruction Setup



In this example, the controller reads 10 elements from the target's (Local Node 2) N7 file, starting at word N7:50. The 10 words are placed in the controller's integer file starting at word N7:0. If five seconds elapse before the message completes, error bit MG11:0/ER is set, indicating that the message timed out.

Valid File Type Combinations

Valid transfers between file types are shown below for MicroLogix messaging:

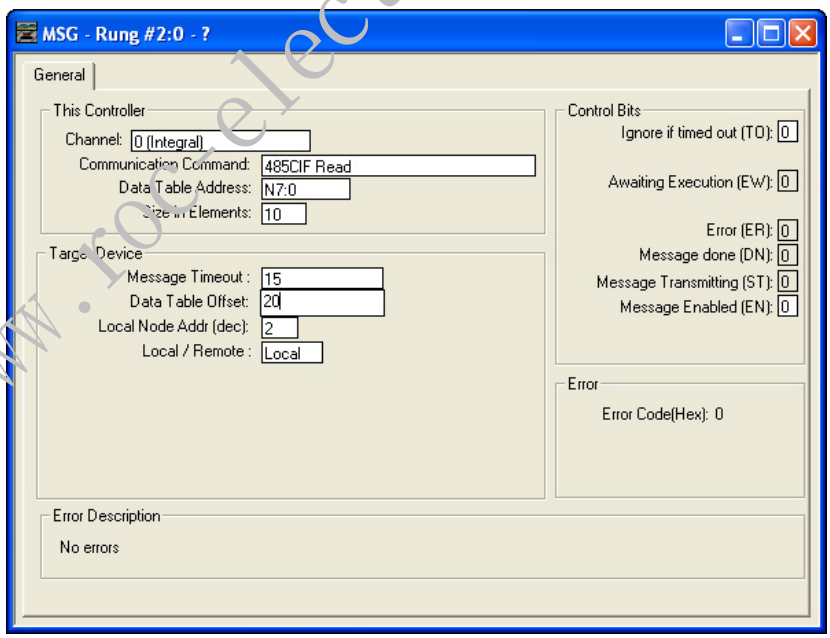
Local Data Types	Communication Type	Target Data Types
O[1], I[1], B, N, L	<---> read/write	O, I, S, B, N, L
T	<---> read/write	T
C	<---> read/write	C
R	<---> read/write	R
RTC[2]	---> write	N, RTC
ST	<---> read/write	ST
F	<---> read/write	F
F	<---> read/write	485CIF
F	<---> read/write	F

(1) Output and input data types are not valid local data types for read messages.

(2) 500CPU write RTC-to-Integer or RTC-to-RTC only.

Example 2 - Local Read from a 485CIF

Message Instruction Setup



In this example, the controller reads five elements (words) from the target device's (Local Node 2) CIF file, starting at word 20 (or byte 20 for non-SLC 500 devices). The five elements are placed in the controller's integer file starting at word N7:0. If 15 seconds elapse before the message completes, error bit MG11:0/ER is set, indicating that the message timed out.

Valid File Type Combinations

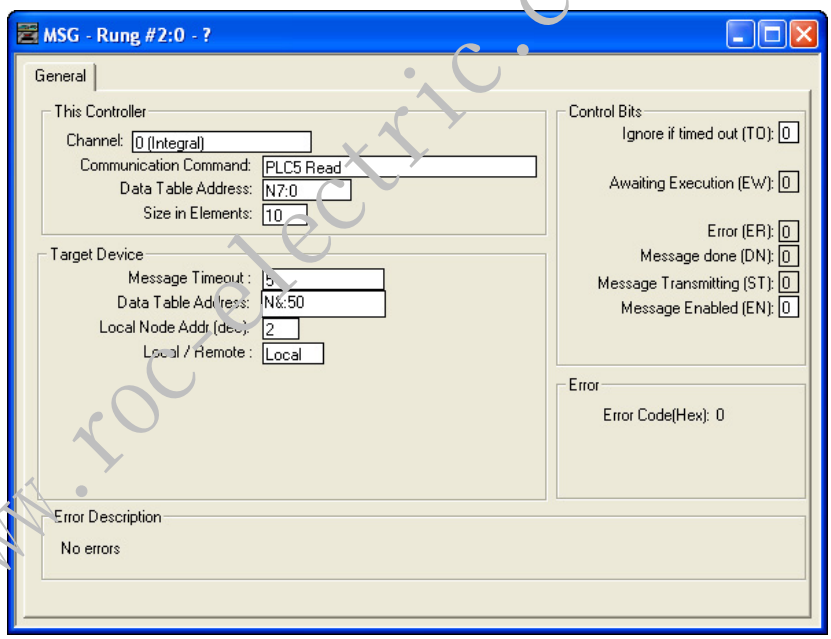
Valid transfers between file types are shown below for MicroLogix messaging:

Local Data Types	Communication Type	Target Data Types
O(1), I(1), B, N, L	<---> read/write	485CIF
T	<---> read/write	485CIF
C	<---> read/write	485CIF
R	<---> read/write	485CIF
ST	<---> read/write	485CIF

(1) Output and input data types are not valid local data types for read messages.

Example 3 - Local Read from a PLC-5

Message Instruction Setup



In this example, the controller reads 10 elements from the target device's (Local Node 2) N7 file, starting at word N7:50. The 10 words are placed in the controller's integer file starting at word N7:0. If five seconds elapse before the message completes, error bit MG11:0/ER is set, indicating that the message timed out.

Valid File Type Combinations

Valid transfers between file types are shown below for MicroLogix messaging:

Local Data Types	Communication Type	Target Data Types
O(1), I(1), B, N, L	<---> read/write	O, I, S, B, N, L
T	<---> read/write	T
C	<---> read/write	C
R	<---> read/write	R
ST	<---> read/write	ST

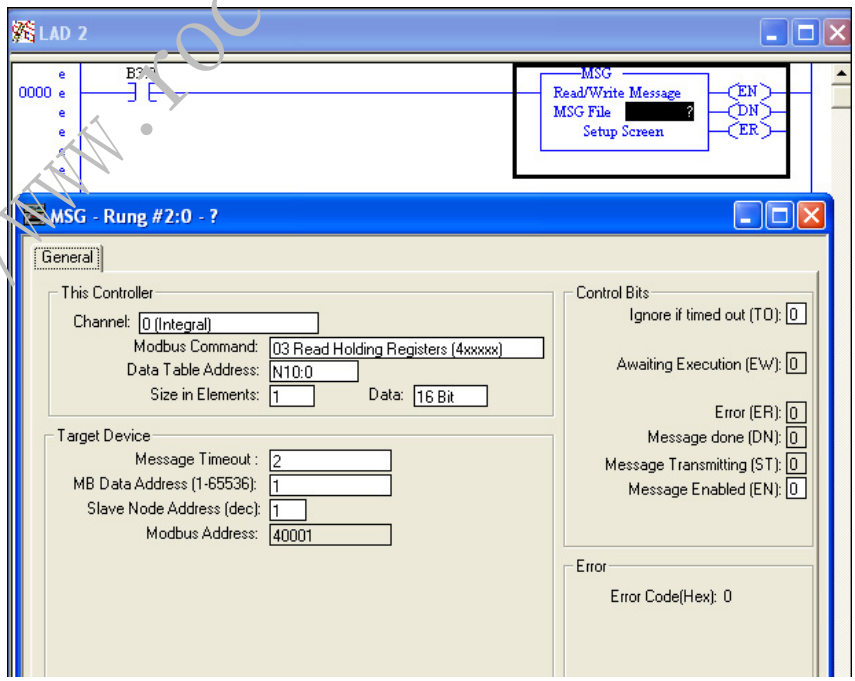
(1) Output and input data types are not valid local data types for read messages.

Example 4 - Configuring a Modbus Message for Channel 0 or Channel 2

This section describes how to configure a local message using the Modbus communication commands. Since configuration options are dependent on which channel is selected, the programming software has been designed to only show the options available for the selected channel.

Before configuring the MSG instruction, open the Channel Configuration screen and set the Driver to Modbus RTU Master. For more information on Channel Configuration, see Modbus RTU Master Configuration on page 556.

Message Setup Screen



Rung 0 shows a standard RSLogix 500/RSLogix Micro message (MSG) instruction preceded by conditional logic.

1. Access the message setup screen by double-clicking Setup Screen.

2. The RSLogix 500/RSLogix Micro Message Setup Screen appears. This screen is used to setup or monitor message parameters for “This Controller”, “Target Device”, and “Control Bits”. Descriptions of each of these sections follow.

"This Controller" Parameters

If a Channel configured for Modbus Master is selected in the Channel field of the Message Setup Screen, the following Modbus Command options will become available:

- 01 Read Coil Status (0xxxx)
- 02 Read Input Status (1xxxx)
- 03 Read Holding Registers (4xxxx)
- 04 Read Input Registers (3xxxx)
- 05 Write Single Coil (0xxxx)
- 06 Write Single Register (4xxxx)
- 15 Write Multiple Coils (0xxxx)
- 16 Write Multiple Registers (4xxxx)

Data Table Address

Local file types must be Binary (B) or Integer (N) for Modbus commands. Starting data table address for coil/input bit commands (1, 2, 5 and 15) require a bit address. Starting data table addresses for register commands (3, 4, 6 and 16) require a word address.

Size in Elements

Size in elements defaults to "1". For coil/input commands (1, 2, 5 and 15), elements are in bits. For register commands (3, 4, 6 and 10), elements are in words.

Target Device

Message Timeout

Message timeout is specified in seconds. If the target does not respond within this time period, the message instruction will generate a specific error (see MSG Instruction Error Codes on page 404). The amount of time that is acceptable should be based on application requirements and network capacity/loading. A 2-second message timeout is generally sufficient, as long as only one message is triggered at a time.

Modbus Data Address (decimal)

The default Modbus Data Address is 1. The Range is 1...65,536.

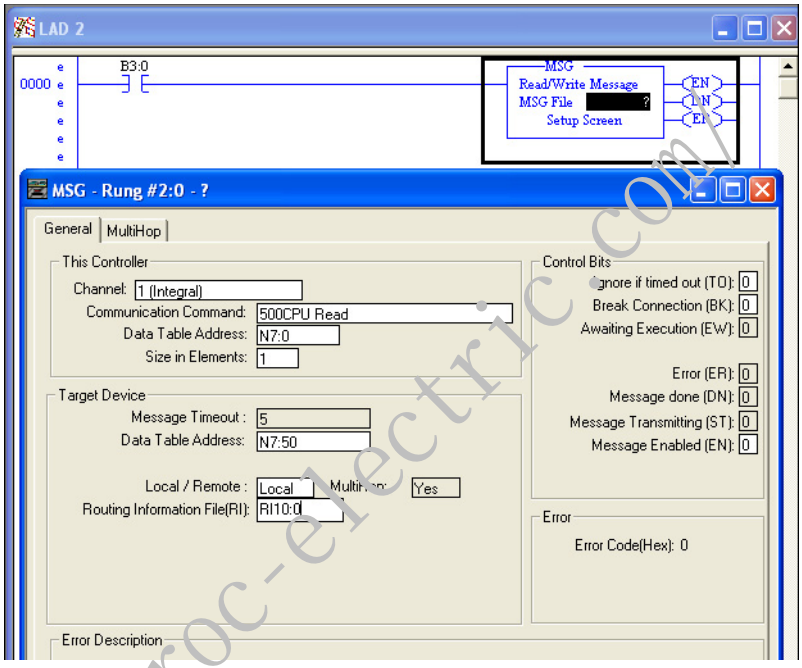
Slave Node Address (decimal)

The default Slave Node Address is 1. The Range is 0...247. Zero is the Modbus broadcast address and is only valid for Modbus write commands (5, 6, 15 and 16).

Example 5 - Configuring an Ethernet/IP Message

This section describes how to configure a local message when you are use Ethernet communication channel 1 of the MicroLogix 1400.

Message Setup Screen



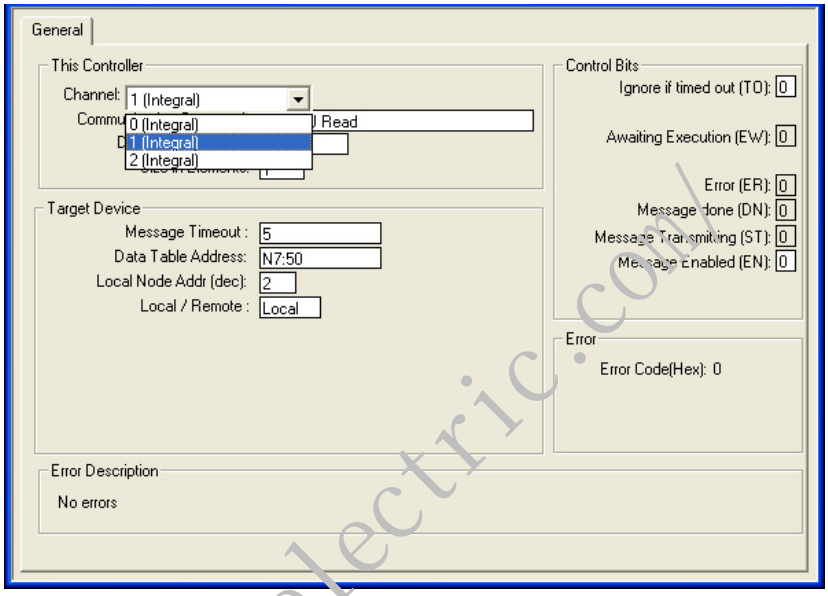
Rung 0 shows a standard RSLogix 500/RSLogix Micro message (MSG) instruction preceded by conditional logic.

1. Access the message setup screen by double-clicking Setup Screen.
2. The RSLogix 500/RSLogix Micro Message Setup Screen appears. This screen is used to setup or monitor message parameters for “This Controller”, “Target Device”, and “Control Bits”. Descriptions of each of these sections follow.

"This Controller" Parameters

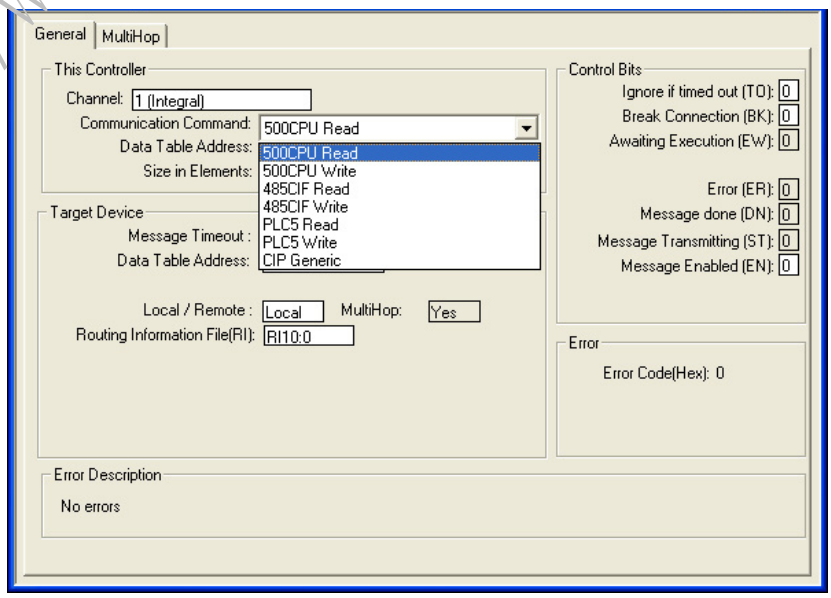
Channel

You must select Channel 1 (Integral) to use Ethernet pathways for messaging.



Communication Command

The controller supports seven different types of communication commands. If the target device supports any of these command types, the controller should be capable of exchange data with the device. You can use one of the seven kinds of message commands, 500CPU, 485CIF, PLC5 and CIP Generic[. Refer to the previous examples for the usage of each command.



"Target Device" Parameters

Message Timeout

Message Timeout for any MicroLogix 1400 channel 1 MSG cannot be modified in the Ethernet Message Setup dialog box. It is assigned by the processor and is determined by adding the Channel 1 MSG Connection Timeout to the MSG Reply Timeout, then adding 15 seconds. This value can be modified by changing one or both of the timeout values in the channel configuration screen for channel 1. The modified message timeout applies to all Ethernet MSG instructions.

Routing Information File

The Routing Information (RI) File stores the path for reaching the destination node. Each RI File Element consists of Sub-Elements 0 through 19 as shown in the following table.

To reach another MicroLogix 1400, an SLC 5/05, a PLC-5E or a controller connected to Ethernet via a 1761-NET-ENI, simply enter in the destination IP address.

Routing Information File Element

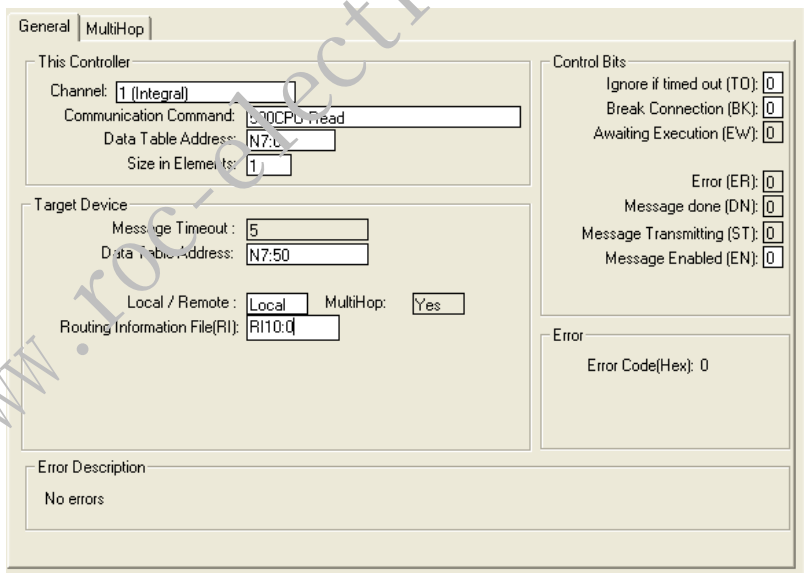
Sub-Element	Bit	Description
0	-	Subtype of Ethernet Message: • 16 (0x10) for normal Multi-Hop MSG • 17 (0x11) for Remote Multi-Hop MSG for a DH+ Network
1	-	High word of 32-bit target IP address(1)
2	-	Low word of 32-bit target IP address
3	8…15	Internal Object Identifier (IOI) size in words (1…5)
	0…7	ASA Service Code
4…8	-	ASA Internal Object Identifier (IOI)
9	-	ASA Connection Path Size in words (1…8)
10…17	-	ASA Connection Paths
18…19	-	Reserved for future use - always 0

(1) IP address is stored in network byte order (big-endian order). For example, IP address 10.121.30.11 will be stored as 0x0a791e0b. Then the IP address will be stored as described in the table below.

Sub-Element	Bit	Value
1	8…15	0x79 (decimal value: 121)
	0…7	0x0a (decimal value: 10)
2	8…15	0x0b (decimal value: 11)
	0…7	0x1e (decimal value: 30)

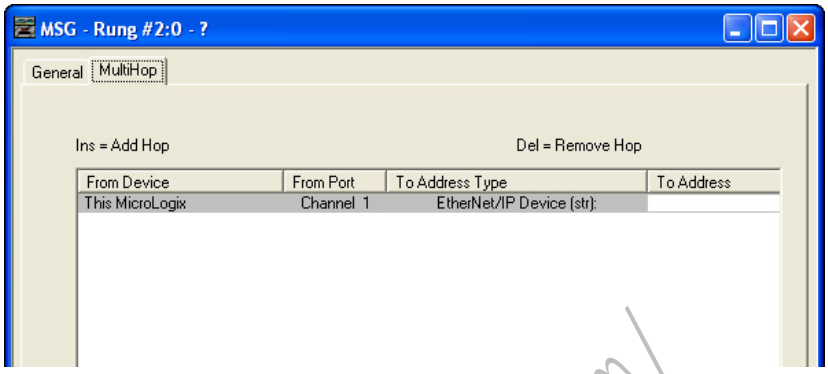
Channel 1 Ethernet

In each MSG instruction setup screen, enter in RIx:y for the Routing Information FIle, where x is an existing RI file number or an unused file number, and y is an unused RI element number. Each Channel 1 Ethernet MSG Instruction must have its own RIx:y. If the RIx:y entered in the MSG setup screen does not yet exist, then the programming software will automatically create it when the rung is verified.



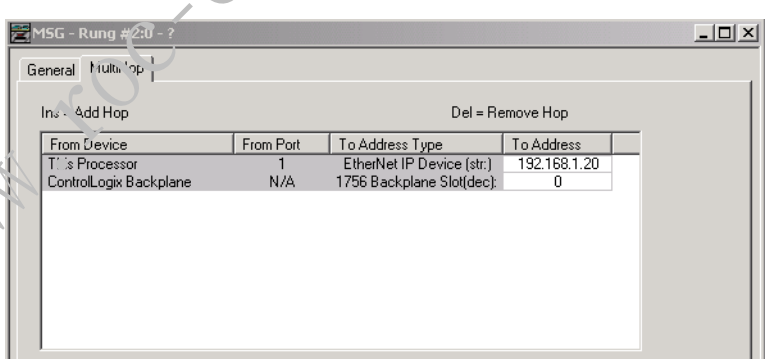
In this example, the controller reads 10 elements from the target's N7 file, starting at word N7:50 using 500CPU Read command. The 10 words are placed in the controller's integer file starting at word N7:0. If 33 seconds elapse before the message completes, error bit MG11:0/ER is set, indicating that the message timed out.

If the target device is another MicroLogix 1400, a SLC 5/05, a PLC-5E or a controller connected to Ethernet via a 1761-NET-ENI, then simply enter in the device's IP address in the "To Address" column as shown below under the MultiHop tab.

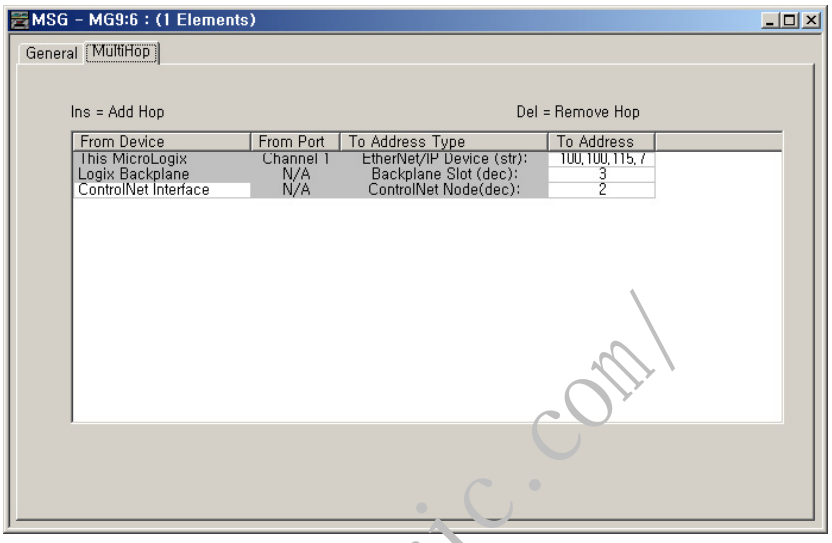


For more information on routing through a ControlLogix gateway, refer to Configuring a Multi-hop Remote Message on EtherNet/IP Communication Channel on page 385.

If the target device is a ControlLogix, FlexLogix or CompactLogix controller with an Ethernet interface, then enter in the interface's IP address in the first row of the "To Address" column, press the computer's Insert key to add a hop. Select ControlLogix Backplane and enter in the backplane slot number for the controller (always 0 for FlexLogix and CompactLogix, 0-16 for ControlLogix) in the second row of the "To Address" column as shown below under the MultiHop tab.



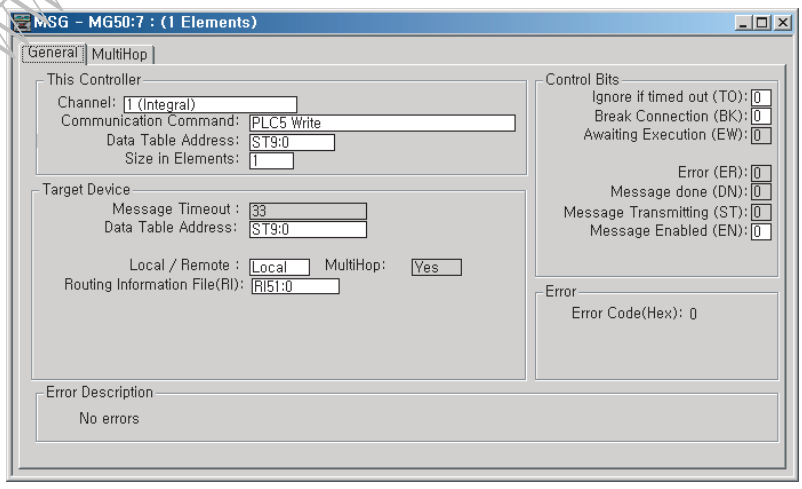
If the target device is a ControlLogix controller with an ControlNet interface, then enter in the interface's IP address of the 1756 ENET module in the first row of the "To Address" column, press the computer's Insert key to add a hop. Select ControlLogix Backplane and enter in the backplane slot number of the ControlNet Interface module (0-16 for ControlLogix) in the second row of the "To Address" column as shown below under the MultiHop tab. Double click on the From Device under the ControlLogix Backplane and select the 1756-ControlNet Interface. Enter the address of the ControlNet Interface using 1747-KFC15.

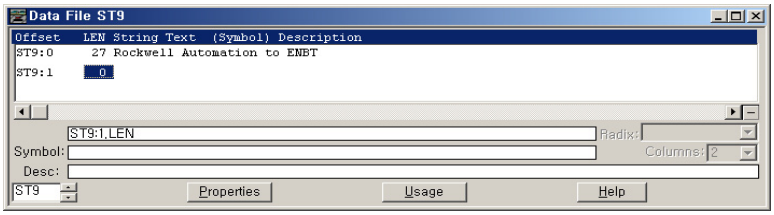


Example 6 - Configuring Local Write message with ST file

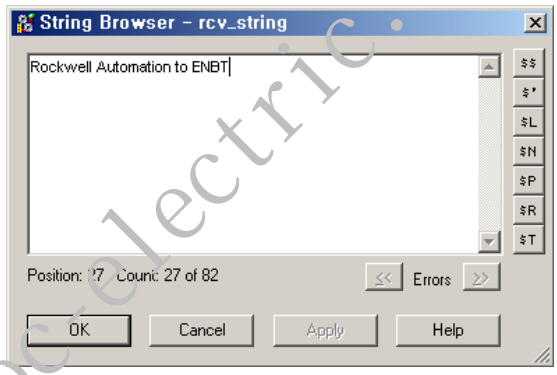
The MicroLogix 1400 can use a message instruction to transfer string file data to target device (SLC5/0x, 1756-L1)

The following message setup screen is used to send local PLC5 write to the 1756-L1 via 1756-ENBT module. A message read will also work.





The “rcv_string” tag on the ControlLogix controller is created as a string type and mapped PLC/SLC mapping table to allow the controller to accept those messages.



Remote Messages

The controller is also capable of remote or off-link messaging. Remote messaging is the ability to exchange information with a device that is not connected to the local network. This type of connection requires a device on the local network to act as a bridge or gateway to the other network.

Remote Networks

DH-485 and DH+ Networks

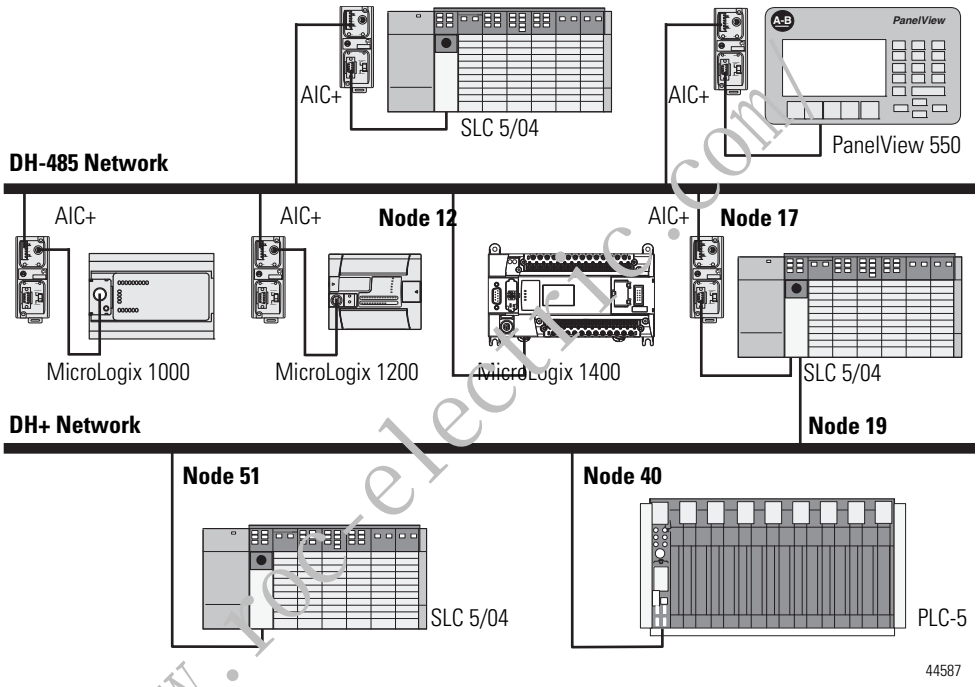
The illustration below shows two networks, a DH-485 and a DH+ network. The SLC 5/04 processor at DH-485 node 17 is configured for passthru operation. Devices that are capable of remote messaging and are connected on either network can initiate read or write data exchanges with devices on the other network, based on each device's capabilities. In this example, node 12 on DH-485 is a MicroLogix 1400. The MicroLogix 1400 can respond to remote message requests from nodes 40 or 51 on the DH+ network and it can initiate a message to any node on the DH+ network.

TIP The MicroLogix 1000 can respond to remote message requests, but it cannot initiate them.

TIP The MicroLogix 1400 capabilities are the same as the MicroLogix 1200 or MicroLogix 1500 in this example.

This functionality is also available on Ethernet by replacing the SLC 5/04 at DH-485 node 17 with an SLC 5/05 processor.

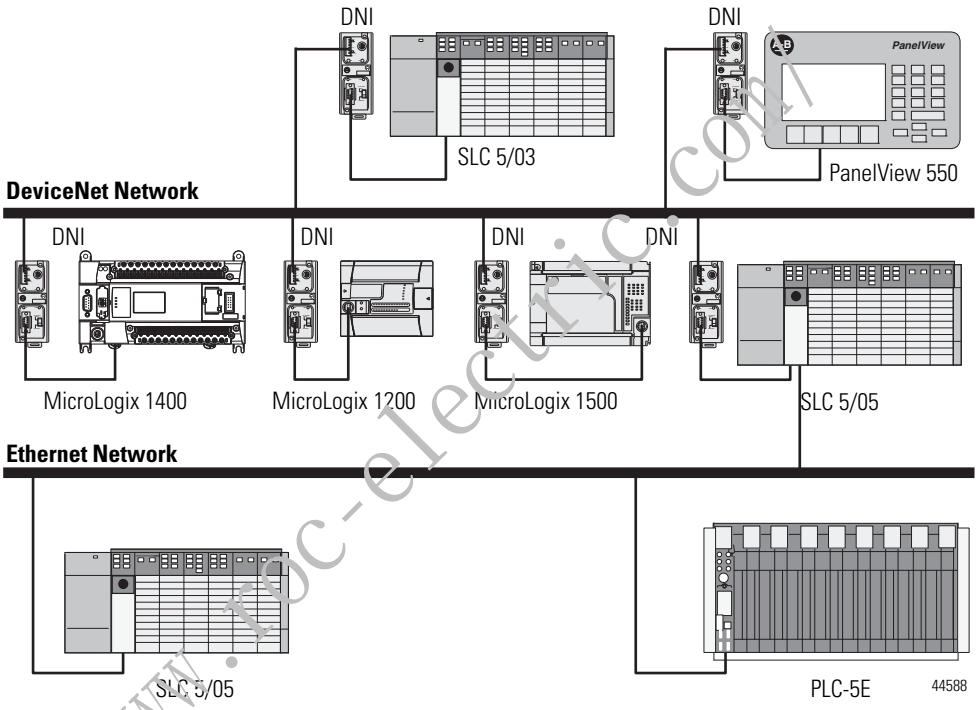
DH-485 and DH+ Networks



DeviceNet and Ethernet Networks

The illustration below shows a DeviceNet network using DeviceNet Interfaces (1761-NET-DNI) connected to an Ethernet network using an SLC 5/05. In this configuration, controllers on the DeviceNet network can reply to requests from devices on the Ethernet network, but cannot initiate communications to devices on Ethernet.

DeviceNet and Ethernet Networks



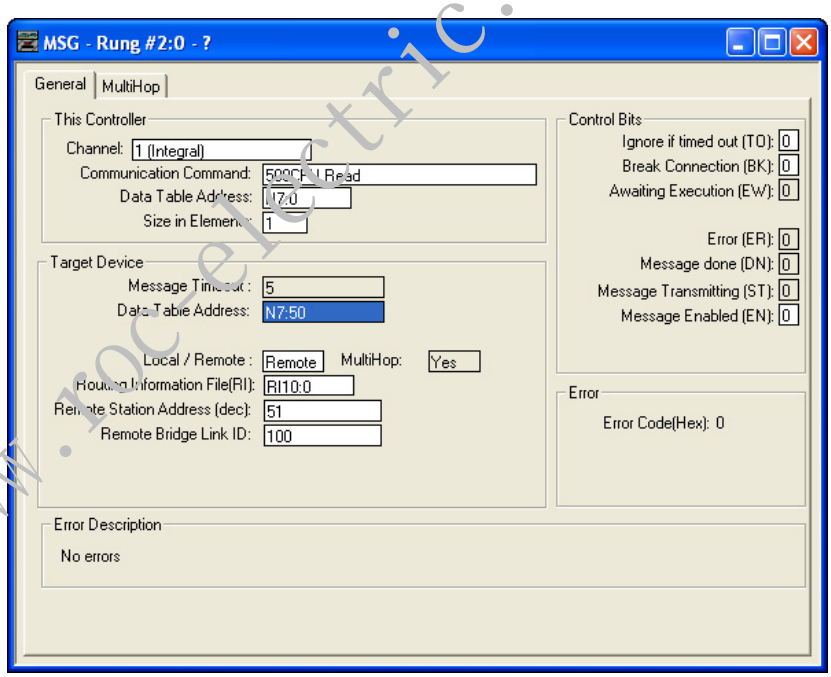
Configuring a Remote Message

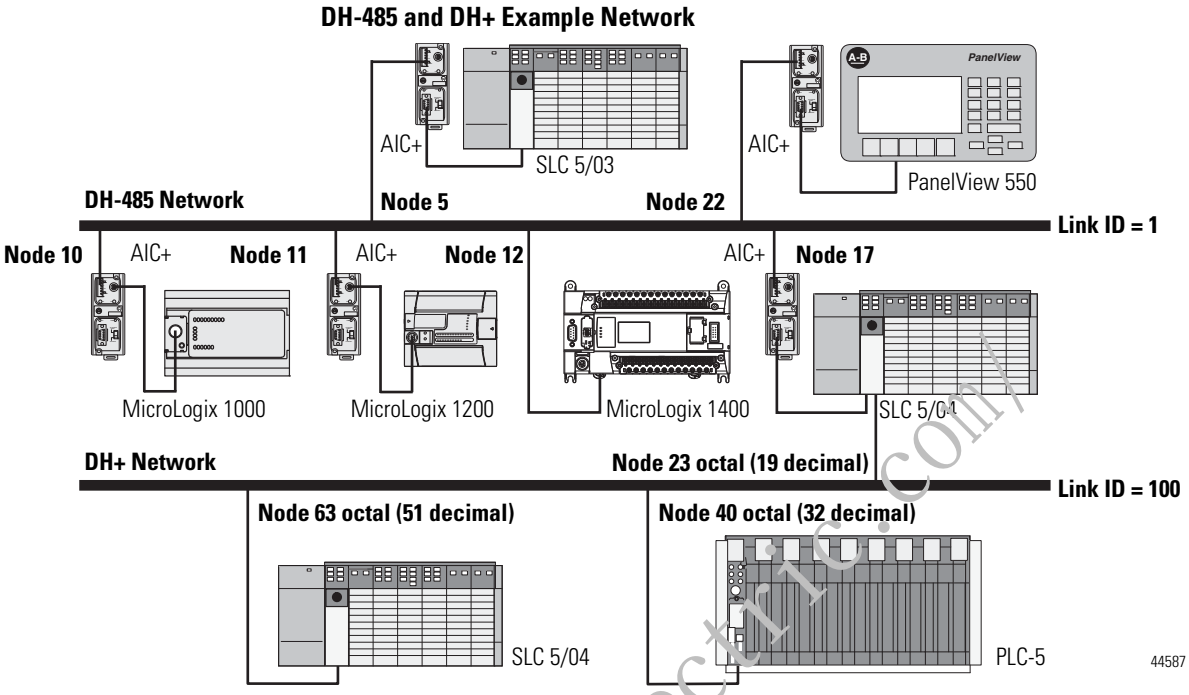
Remote capability is configured through the RSLogix 500/RSLogix Micro Message Setup screen.

Example Configuration Screen and Network

The message configuration shown below is for the MicroLogix 1400 at node 12 on the DH-485 network. This message reads five elements of data from the SLC 5/04 (node 51 on the DH+ network) starting at address N:50:0. The SLC 5/04 at Node 23 of the DH+ network is configured for passthru operation.

TIP The MicroLogix 1400 capabilities are the same as the MicroLogix 1200 or MicroLogix 1500 in this example.





"This Controller" Parameters

See "Target Device" Parameters on page 363.

"Control Bits" Parameters

See "Control Bits" Parameters on page 346.

"Target Device" Parameters

Message Timeout

See Message Timeout on page 363.

Data Table Address

See Data Table Address/Offset on page 363.

Local Bridge Address

This variable defines the bridge address on the local network. In the example, DH-485 node 12 (MicroLogix 1400 on Link ID 1) is writing data to node 51 (SLC 5/04 on Link ID 100). The SLC 5/04 at node 17 is the bridge device.

This variable sends the message to local node 17.

Remote Bridge Address

This variable defines the remote node address of the bridge device. In this example, the remote bridge address is set to zero, because the target device, SLC 5/04 at node 63 (octal) is a remote-capable device. If the target device is remote-capable, the remote bridge address is not required. If the target device is not remote-capable (SLC 500, SLC 5/01, SLC 5/02, and MicroLogix 1000 Series A, B and C), the remote bridge address is required.

Remote Station Address

This variable is the final destination address of the message instruction. In this example, integer file 50 elements 0...4 of the SLC 5/04 on Link ID 100 at node 63 (octal) receives data from the MicroLogix 1400 controller at node 12 on Link ID 1.

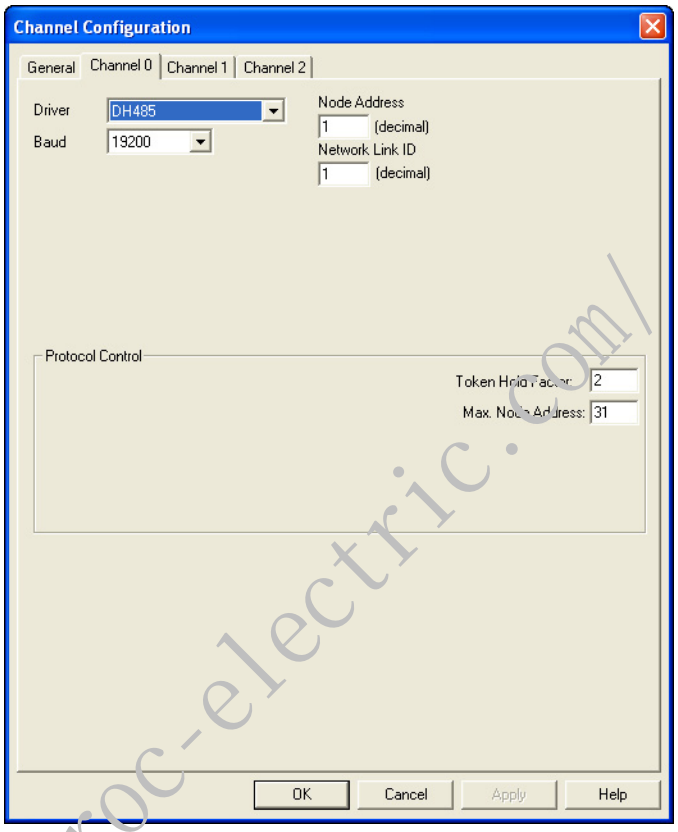
Remote Bridge Link ID

This variable is a user-assigned value that defines the remote network as a number. This number must be used by any device initiating remote messaging to that network. In the example, any controller on Link ID 1 sending data to a device on Link ID 100 must use the remote bridge link ID of the passthru device. In this example, the SLC 5/04 on Link ID1, node 17 is the passthru device.

Network Link ID

Set the Network Link ID in the General tab on the Channel Configuration screen. The Link ID value is a user-defined number between 1 and 65,535. All

devices that can initiate remote messages and are connected to the local network must have the same number for this variable.



Configuring a Multi-hop Remote Message on EtherNet/IP Communication Channel

A user can configure a multi-hop remote message in the RSLogix500 Message Setup screen.

Three examples of EtherNet/IP messaging are shown in this section:

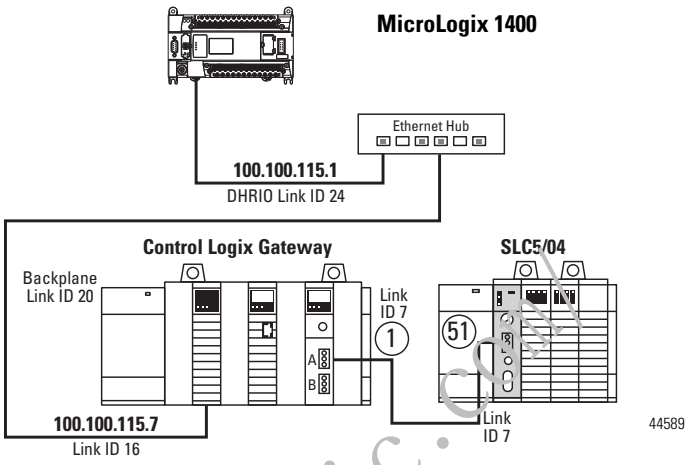
- MicroLogix 1400 Ethernet to SLC5/04 DH+ via ENET & DHRIO.
- MicroLogix 1400 Ethernet to SLC 5/03 DH485 via ENET, DHRIO and 1785-KA5 bridge device
- MicroLogix 1400 Unsolicited Write Message to RSLinx via Ethernet

Network Message Example 1:

MicroLogix 1400 Ethernet to SLC5/04 DH+ via ENET & DHRIO

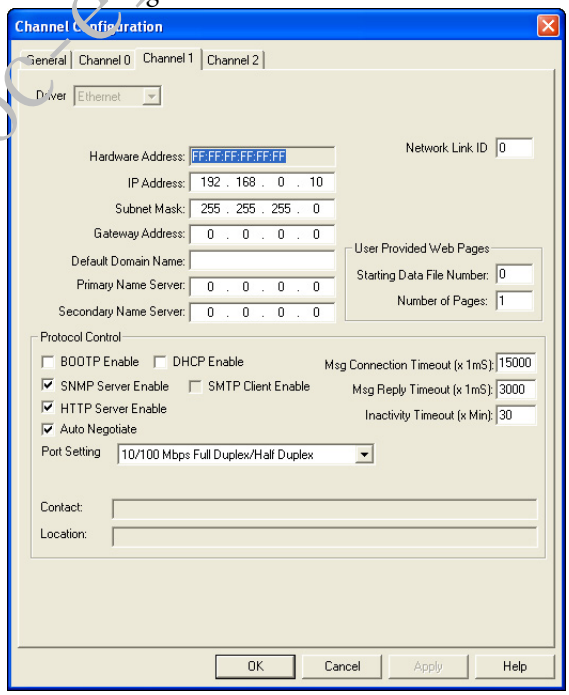
The following illustrates the MicroLogix 1400 (CH1 Ethernet) sending a remote message to a SLC5/04 processor (DH+ Node 51). The remote message will s an ENET module, a ControlLogix chassis (Gateway) and a DHRIO module. In

order for the message to pass through the network, a MultiHop MSG must be setup and a DHRIO Routing table must exist.



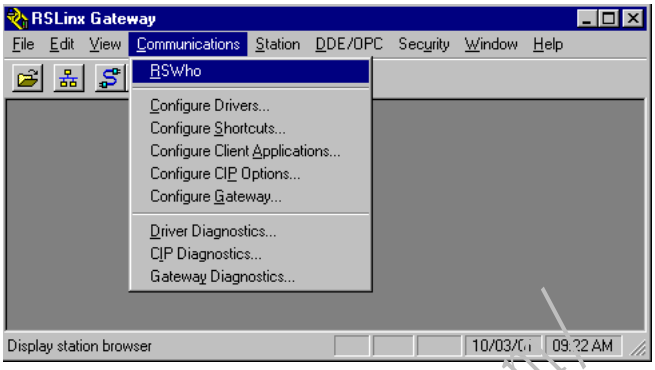
Belden 9463 "Blue Hose" cable is used to connect the DH+ devices on the network. Ethernet cable and an Ethernet hub are used to connect the ENET module and the MicroLogix 1400 CH1 Ethernet ports together.

MicroLogix 1400 CH1 Configuration

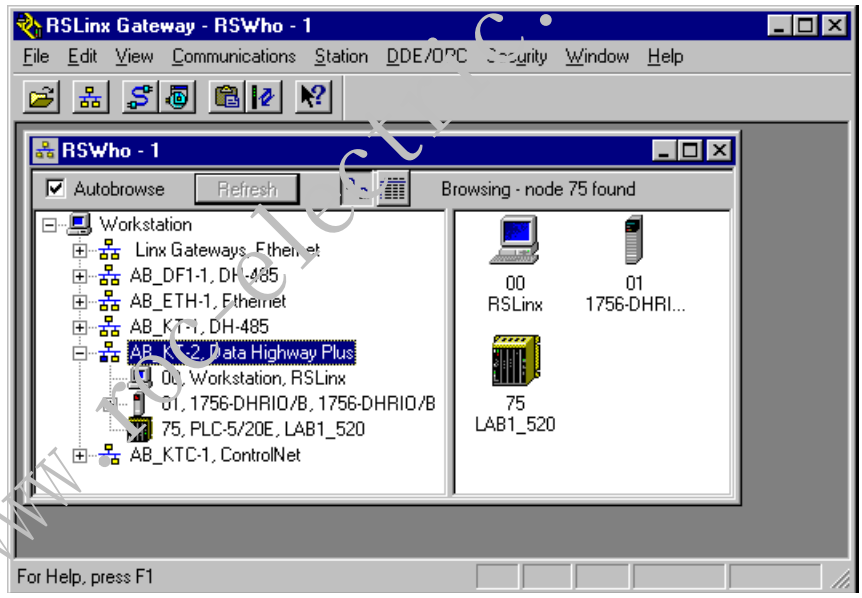


DHRIO Routing table creation

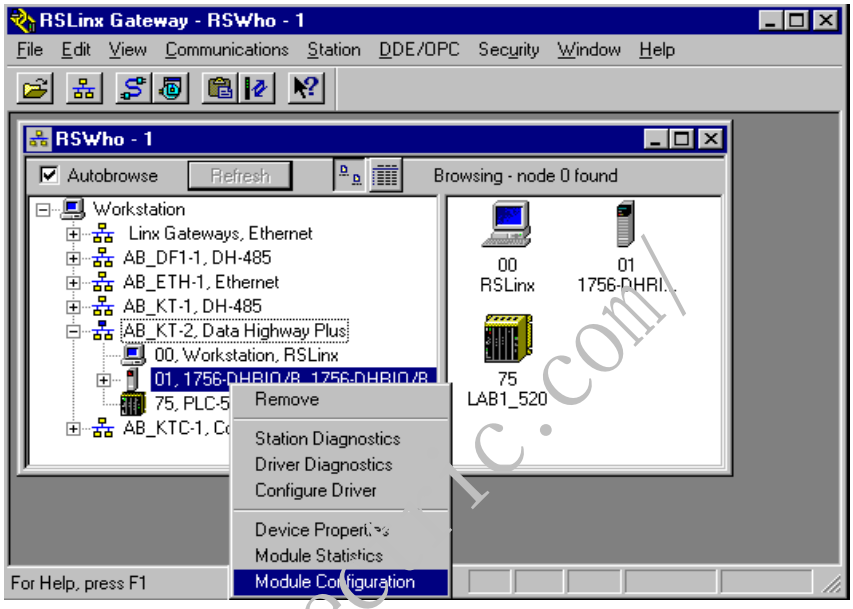
To create a DHRIO Routing table open up RSLinx and under Communication select RSWho.



Select a driver that will allow you to see and connect up to the DHRIO module.

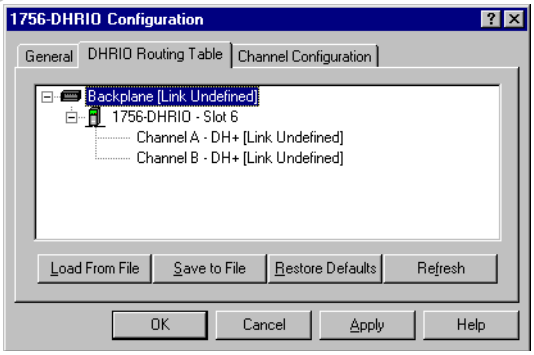


Right Click your mouse on top of the DHRIO module and a drop down box will open.



Select Module Configuration by clicking with the left mouse button.

Select DHRIO Routing Table tab. If no routing table has been created the following should appear.



Right click on the Backplane and left click on Edit Module. Make sure that the Back plane Link ID is set to 20.

Right click on the 1756-DHRIO module and left click on Edit Module. Make sure that CH A's Link ID is set for 7 and CH B's Link ID is set for 2. Select OK. Channel B is actually not necessary.

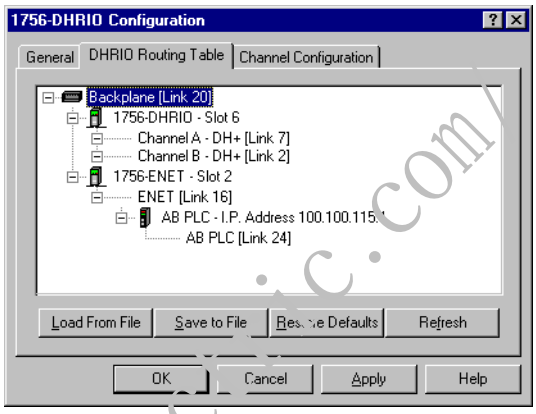
Right click on the Backplane and left click on Add Module. Left click on 1756-ENET.

Enter the correct slot number 2 and Link ID 16 for the ENET module.

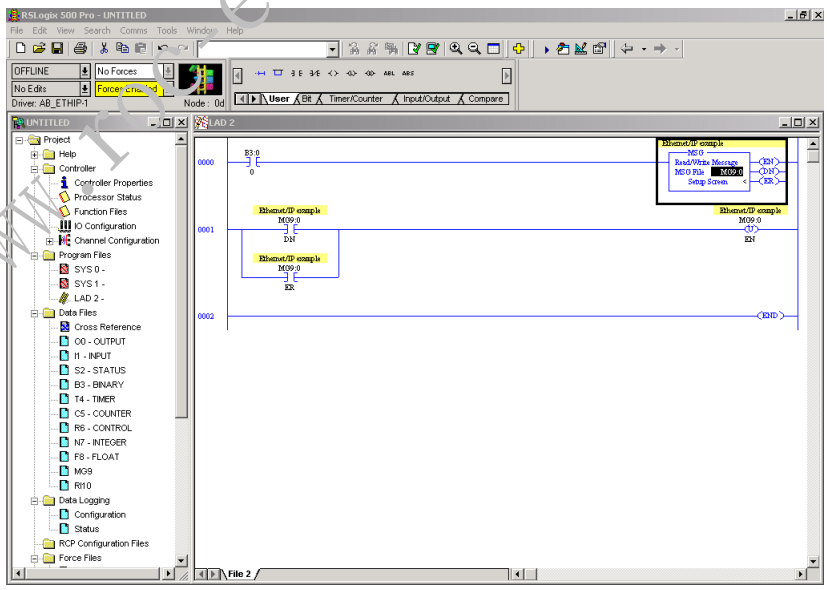
Right click on the 1756-ENET Link ID and left click on Add Module. Left click on AB PLC.

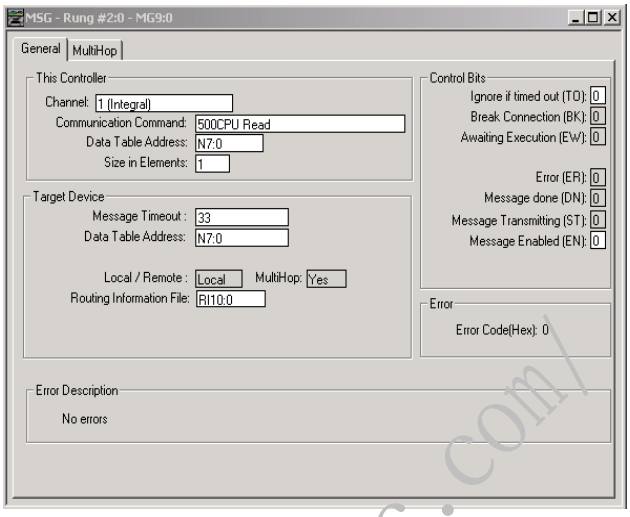
Enter the IP address (100.100.115.1) for the destination Ethernet processor and its Link ID (24).

The Configuration should now look like the following.



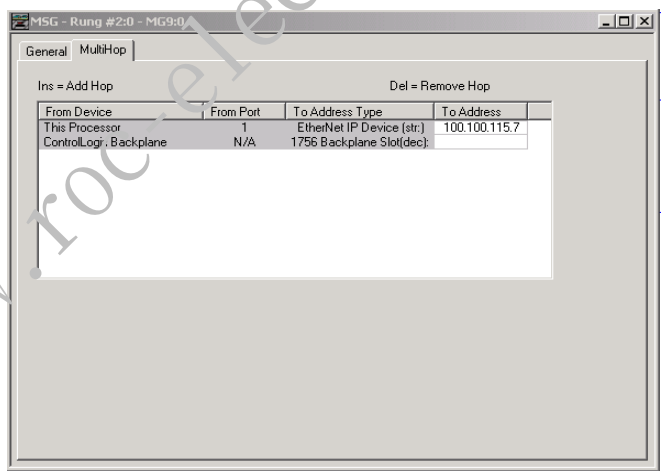
The following is the logic necessary for the MicroLogix 1400 processor.





A MSG route must be configured in the MultiHop tab of the MSG Setup Screen.

Click on the MultiHop tab.



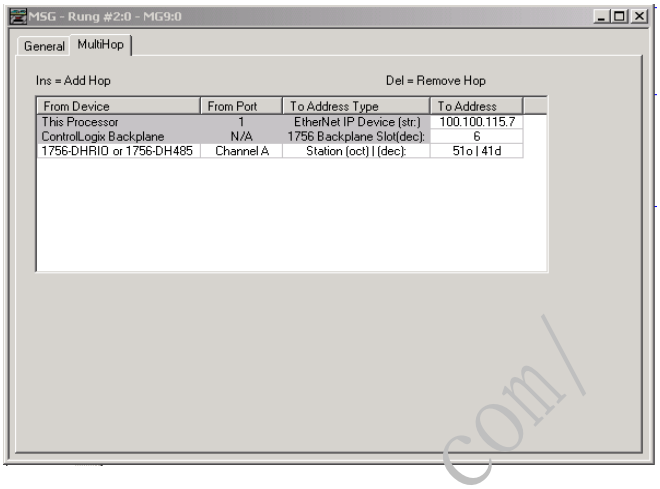
Enter in the IP address of the 1756 ENET module, select ControlLogix backplane, press the Insert key and enter in the backplane slot numbers of the DHRIO module (0-16) under the 'To Address' fields.

Click on the ControlLogix Backplane to highlight it and press the Insert key on your computer's keyboard to add another hop.

Double click on the From Device under the ControlLogix Backplane and select the 1756-DHRIO.

Make sure that the From Port for the DHRIO module is set for Channel A.

Enter in the destination node address (DH+ octal address of target processor) under the To Address.

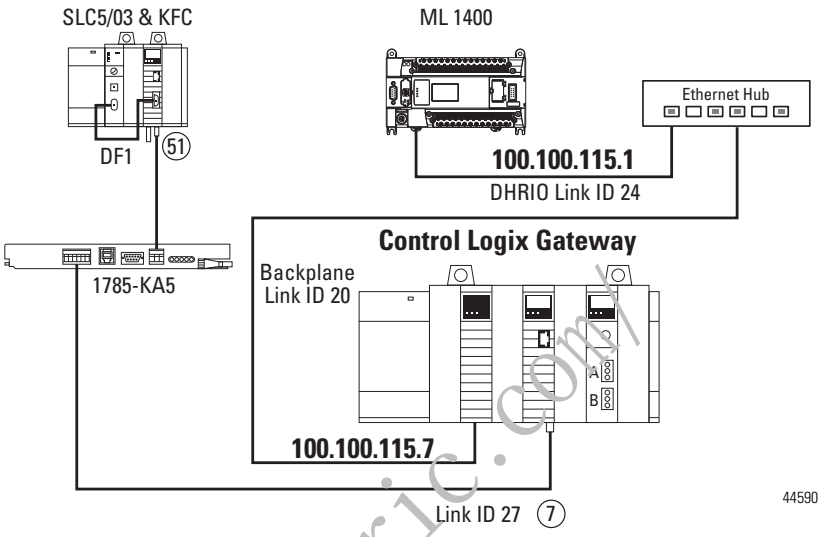


Note: Make sure that the Target Device Data Table Address exists in the target device.

Network Message Example 2:

MicroLogix 1400 Ethernet to SLC 5/03 DH485 via ENET, DHRIO and 1785-KA5 bridge device

The following illustrates the MicroLogix 1400 (CH1 Ethernet) sending a remote message to a SLC5/03 processor (DH+ Node 51). The remote message will passthru an ENET module, a ControlLogix chassis (Gateway), a DHRIO module and a 1785-KA5 bridge device. In order for the message to pass through the network, a multiHop MSG must be set up and a DHRIO Routing table must exist. It must route to a DHRIO module onto DH+ thru a 1785-KA5 bridge to DH485. Follow the example below for the configuration steps.

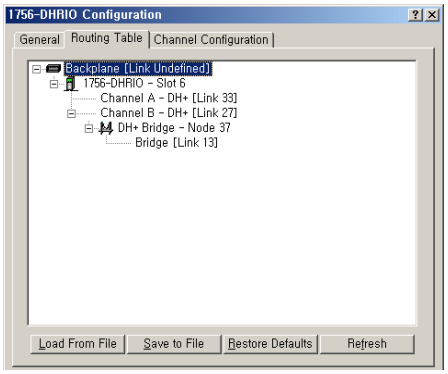


Adding 1785-KA5 bridge module

Routing to a DHRIO module on to DH+ thru a 1785KA5 bridge to DH485. In order for the RSLinx, RSWho window to browse the DH485 network you must configure the 1785KA5 bridge in the ControlLogix Gateway Configuration software (1756gtwy). Follow the example below for the configuration steps.

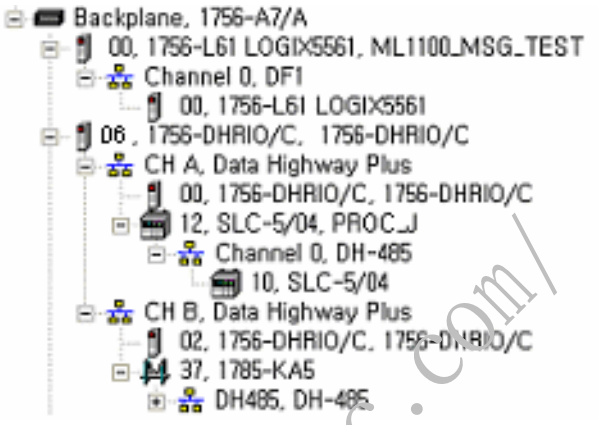
To go from a DHRIO module through a 1785-KA5 bridge device to a DH485 network, the DHRIO module must be configured using the ControlLogix Gateway Configuration tool. For example, if a 1785-KA5 bridge is on a DH+ network at node 37 and the DH485 LinkID is 13, complete the following:

DHRIO Routing table creation

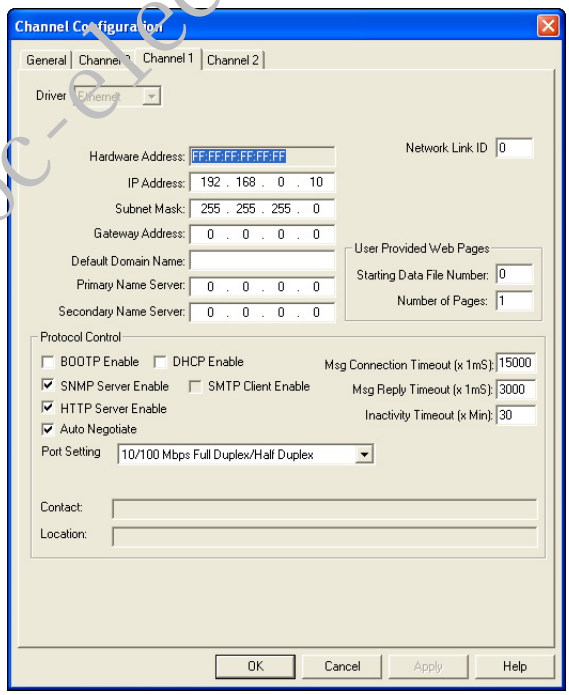


Go to the routing table configuration tab. Right-click on the DHRIO channel being used and select Add Module. Select the DH+ Bridge. Enter the DH+ node number of the KA5 (37 in this example), and the Link ID of the DH485 (13 in this example). Click Apply.

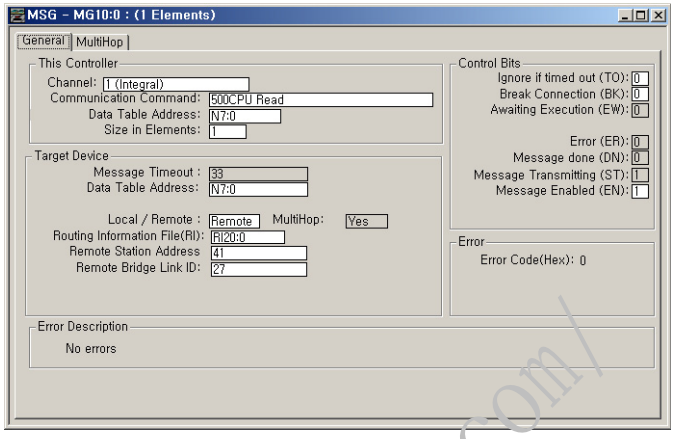
You can now browse through the KA5 module from RSWho.



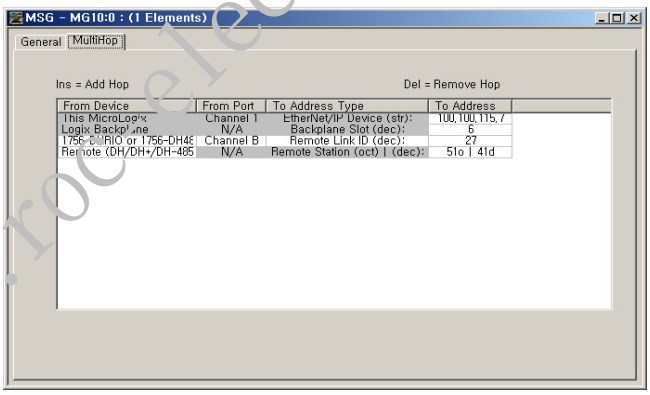
ML1400 Channel1 Configuration



The following is the message setup screen for the MicroLogix 1400 controller.



A MSG route must be configured in the Multi Hop tab of the MSG Setup Screen. Click on the MultiHop tab.



Enter in the IP address of the 1756 ENET module, select ControlLogix backplane, press the Insert key and enter in the backplane slot numbers of the DHRIO module (0-16) under the 'To Address' fields. Click on the ControlLogix backplane to highlight it and press the Insert key on your computer's keyboard to add another hop. Double click on the From Device under the ControlLogix backplane and select the 1756-DHRIO. Make sure that the From Port for the DHRIO module is set for Channel B.

Enter in the destination Link ID (DH+ address of target processor) under the To Address. Press the Insert key to add another hop. Double click on the From Device and select Remote(DH/DH+/DH-485). Enter in the destination node address (DH+ octal address of target processor) under the To Address.

Note: Make sure that the Target Device Data Table Address exists in the target device.

Network Message Example 3:

MicroLogix 1400 Unsolicited Write Message to RSLinx via Ethernet

Initiating an unsolicited write MSG to RSLinx via EtherNet/IP requires sending a remote format message that includes Source Link ID and address, as well as Destination Link ID and address.

The MicroLogix 1400 can send remote ethernet messages using EtherNet/IP messages. The local version of EtherNet/IP messages use the 0x4B Execute PCCC service code, whereas the remote version needs to use the 0x4C Execute DH+ service code.

The remote Ethernet message setup screen works the same as for a remote DH-485 message. Selecting remote in the selectable local/remote field shows two new fields: Remote Station Address and Remote Bridge Link ID

Also, following set up is needed in RSLinx side for MicroLogix 1400 unsolicited MSG communication with OPC client.

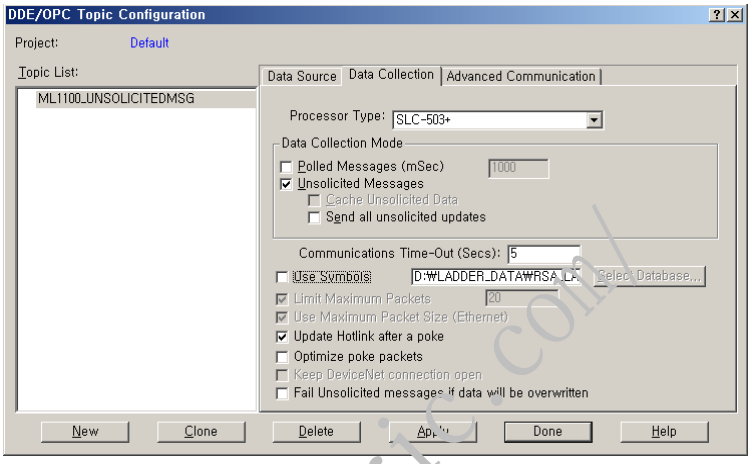
- Remote Bridge Link ID 15 (dec)
- Remote Station Address 63 (dec)
- Chan 1 Network Link ID 0
- Source Station Address 0 (always)

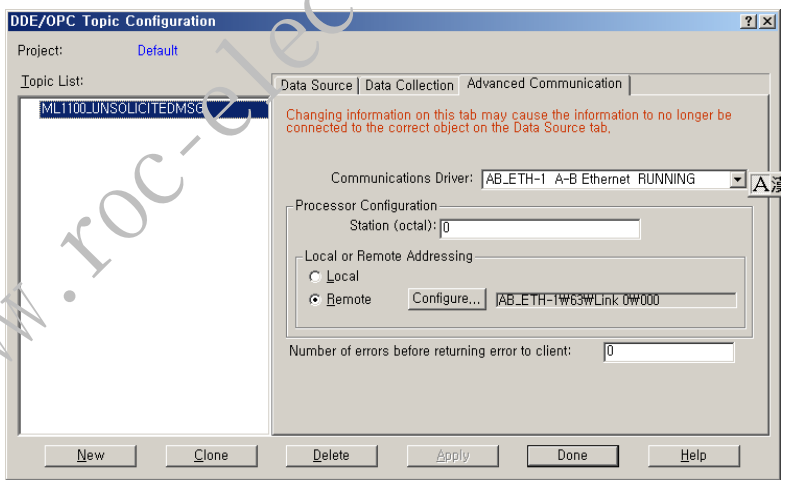
There are four steps required to send unsolicited message to RSLinx DDE/OPC client applications via EtherNet/IP

- Configure a new DDE/OPC topic in RSLinx for unsolicited data.
- Configure Remote Routing Configuration.
- Configure the DDE topic and Item in RSLinx.
- Configure the ML1400 MSG instruction

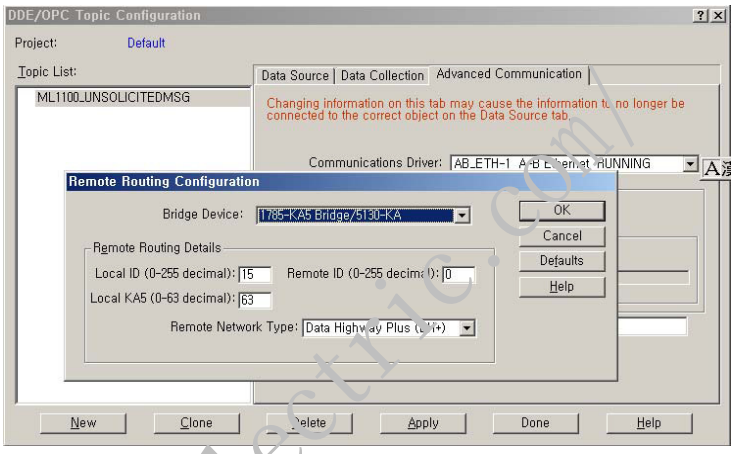
1. Configure a new DDE/OPC topic in RSLinx for unsolicited data

In RSLinx, select DDE/OPC menu, then select Topic Configuration. Click new, enter a topic name, and click OK. The DDE/OPC Topic Configuration dialog will appear.

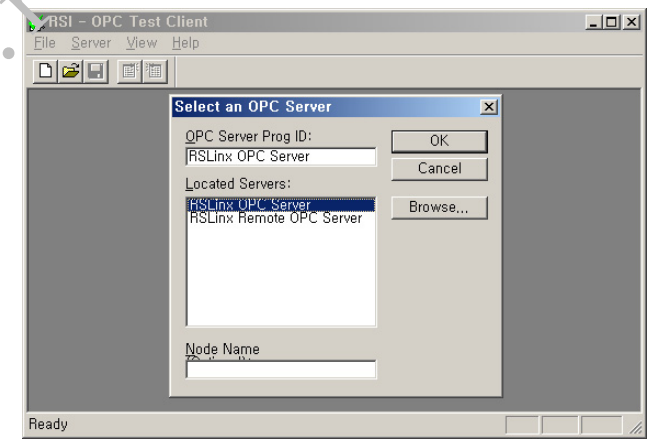




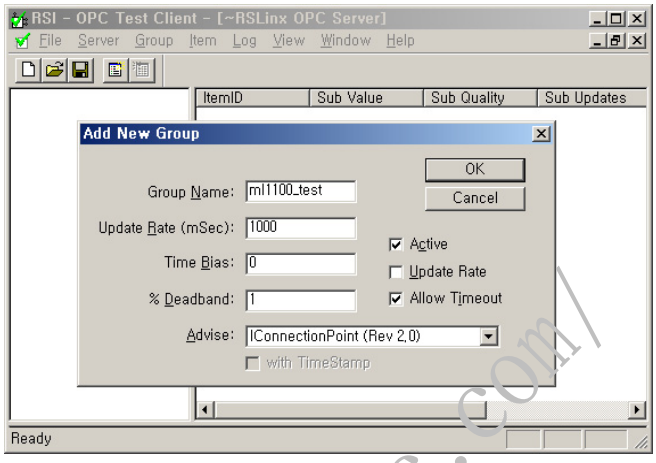
2. Configure Remote Routing Configuration After selecting Remote Addressing and clicking on the Configure button, select the 1785-KA5 Bridge/5130-KA for the Bridge Device. Select DH+ for the Remote Network Type and Local ID is set to 15(dec), Local KA5 is set to 63(dec). The Remote ID should match the Network Link ID of the initiating ML1400, and must be unique between controllers initiating unsolicited messages to RSLinx:



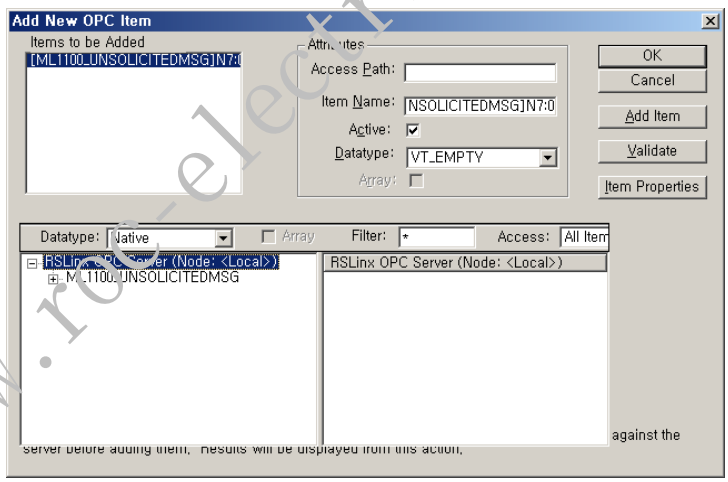
3. Configure the DDE topic and Item in RSLinx
   1. Connect to the RSLinx OPC Server:



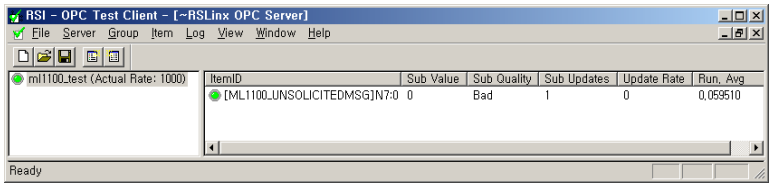
2. Add a new group:



3. Add a new item:

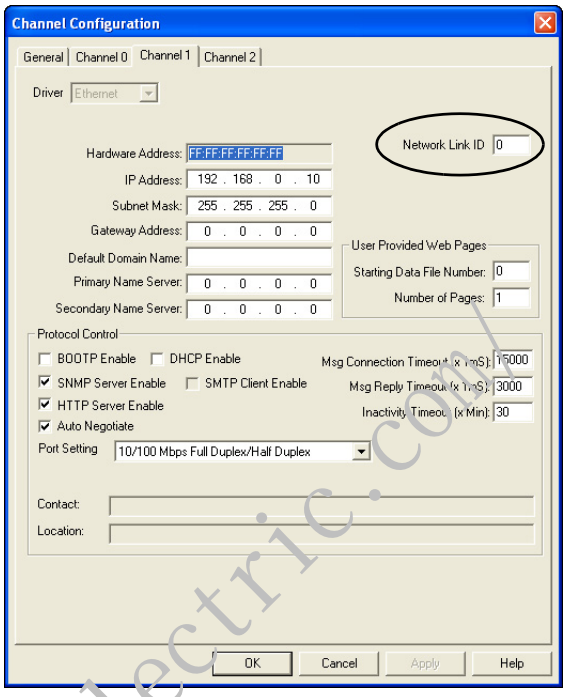


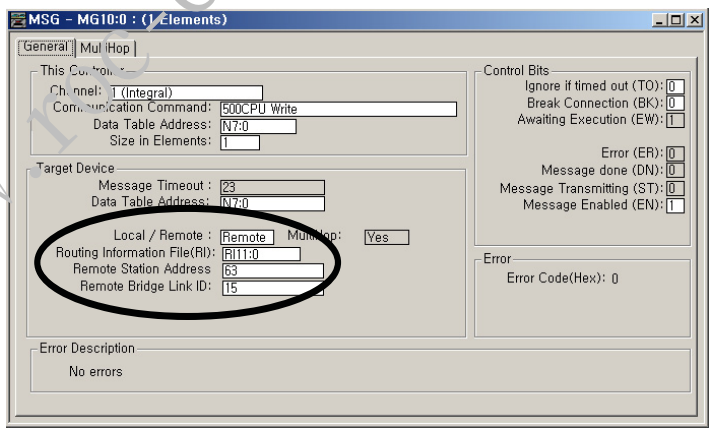
4. Note that the "Sub Quality" will be "Bad" until an unsolicited message is received:

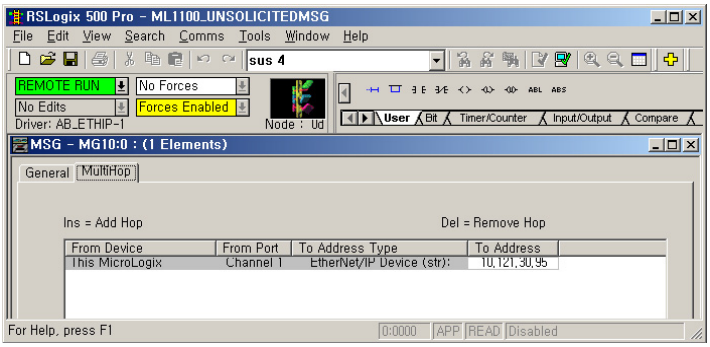


4. Configure the ML1400 MSG instruction

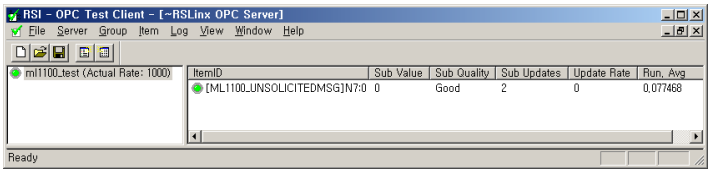
ML1400 Channel 1 Network Link ID must be matched with the DDE/OPC Topic Remote ID configured in step #2 (0, in this example). Also configure a Remote Ethernet MSG in the ML1400 to Remote Station Address 63 and Remote Link ID 15:





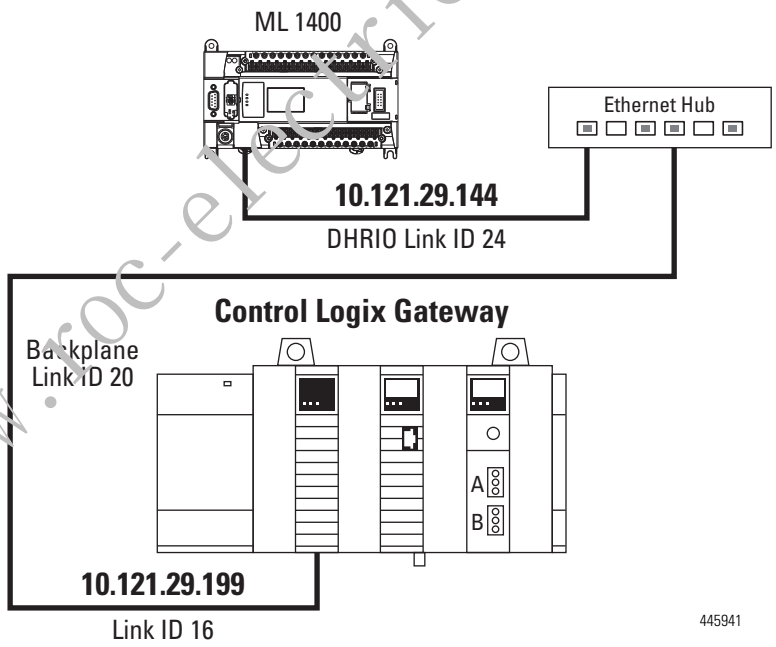


Trigger MSG instruction. It should complete done, and OPC Test Client should display the N7:0 data, as well as "Good" Sub Quality:

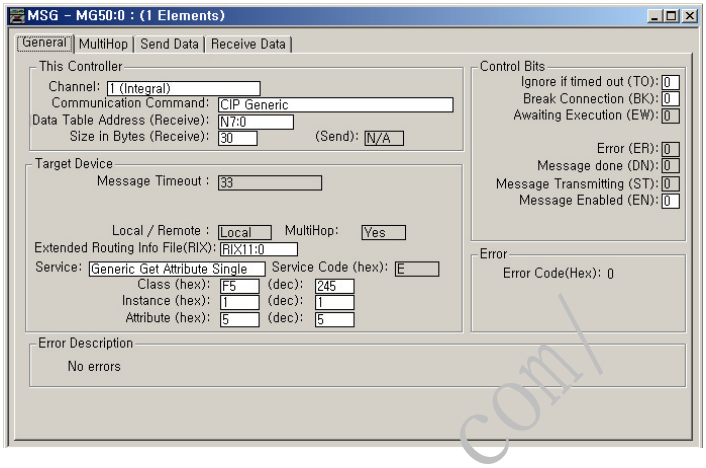


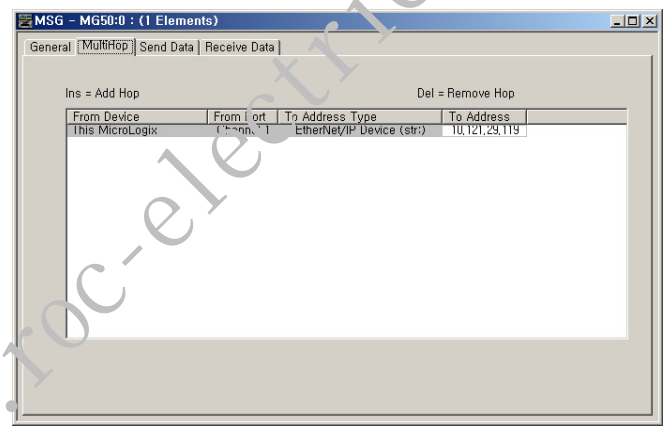
Configuring a MicroLogix 1400 CIP Generic Message via Ethernet

The MicroLogix 1400 supports CIP Generic messages over ethernet port. This section describes how to configure a CIP Generic message when you are use Ethernet communication channel 1 of the MicroLogix 1400. The Network Configuration is shown below.



The RSLogix Message Setup Screen is shown below. This screen is used to setup "This Controller", "Target Device", and "Control Bits". Descriptions of each of the elements follow.





"This Controller" Parameters

You must select CIP Generic in Communication Command field. Also the size of bytes(receive or send) is the length of service data to be sent or received in the message.

"Target Device" Parameters

Extended Routing Information File

The CIP generic communication command requires an extended routing information file type, RIX, to store the longer path information for CIP generic message types. Each RIX file Element consists of Sub-Elements 0 through 24 as shown in the following table. To reach another MicroLogix 1400, an SLC 5/05, a PLC-5E or a controller connected to Ethernet via 1761-ENI, simply enter in the destination IP address.

Extended Routing Information File Element

Sub-Element	Bit	Description
0	-	Subtype of Ethernet Message:
		19 (0x13) for CIP Generic MSG
1	-	High word of 32-bit target IP address
2	-	Low word of 32-bit target IP address
3	15 to 8	ASA Service
	7 to 0	Internal Object Identifier (IOI) size in words (1 to 5)
4 to 8	-	ASA Internal Object Identifier (IOI)
9	-	ASA Connection Path Size in words (1 to 15)
10 to 24	-	ASA Connection Path

Service Type and Service Code

The table below indicates the service (for example, Get Attribute Single or Set Attribute Single) that you want to perform. Available services depend on the class and instance that you are using. When the user clicks on the pull-down button on the right of the Service Type box, then a pull-down list window with Custom as the default will appear for the user to select one of these service types. Depending on which Service Type is selected, user must fill the Class, Instance, and Attribute field that is represented as '?' mark with an appropriate Hex value.

The Service Code is the code for the requested EtherNet/IP service. This value changes based on the Service type that has been selected. When user select a Service type other than Custom, this is a read-only box. If user select "Custom" in the Service type box, then user need to specify a service code in this box. Note that only the Service Code is filled in for the user. The Class, Instance, and Attribute must be filled in by the user just as the table below indicates with question marks in their corresponding columns.

Service Type Pull-Down List

	Auto-Fill fields			
Service	**Service Code**	**Class**	**Instance**	**Attribute**
Custom	?	?	?	?
Read Assembly	0x0E	0x04	?	3
Write Assembly	0x10	0x04	?	3
Read Output Point	0x0E	0x10	?	3
Write Output Point	0x10	0x09	?	3
Read Input point	0x0E	0x08	?	3

Service Type Pull-Down List

Read Parameter	0x0E	0x0F	?	1
Write Parameter	0x10	0x0F	?	1
Read Analog Input	0x0E	0x0A	?	3
Write Analog Output	0x10	0x0B	?	3
Generic Get Attribute Single	0x0E	?	?	?
Generic Set Attribute Single	0x10	?	?	?
Generic Get Member	0x18	?	?	?
Generic Set Member	0x19	?	?	?
Reset Identity Object	0x05	0x01	?	N/A

Note 1: Everywhere there is a question mark, this box is filled in by the user.

Note 2: Everywhere there is a value, that box also has user edits disabled.

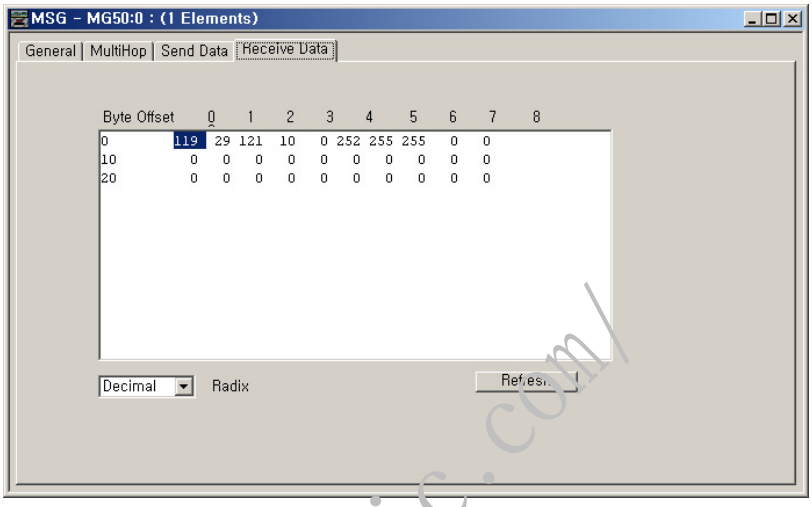
Note 3: All other fields not mentioned here are unaffected by the Service Type.

In this example, a Get Attribute Single message reads a single attribute value. The Class Code 0xF5 indicates TCP/IP Interface Object. The TCP/IP Interface Object provides an attribute that identifies the link-specific object for the associated physical communications interface. Each device shall support exactly one instance of the TCP/IP Interface Object for each TCP/IP capable communications interface on the module. A request to access instance 1 of the TCP/IP Interface Object shall always refer to the instance associated with the interface over which the request was received.

The attribute ID 5 identifies TCP/IP network interface configuration parameters (consist of IP address, network mask, gateway address, DNS name)

The controller reads 30 elements for a single attribute value from the ControlLogix controller. The 30 bytes are placed in the controller's integer file starting at word N7:0

When the message is replied successfully, a user can check configuration parameters in Receive Data tab.



CIP Generic Error Codes/Internal Fail Codes

When CIP Generic sub-system cannot send a message due to some reason or reply contains error code, error code is shown via MSG instruction. Error Code 0xE0 is stored in Word 13 of MG file. Internal Fail Code is stored in Word 22 of MG file. When messaging through CIP communication and the low byte is 0xE0, the high byte of this sub-element contains detailed Fail Code returned by the CIP sub-system.

Status Code Reference: CIP Common Specification Appendix B: Status Codes

MSG Instruction Error Codes

When the processor detects an error during the transfer of message data, the processor sets the ER bit and enters an error code that you can monitor from your programming software.

MSG Instruction Error Codes

Error Code	Description of Error Condition
02H	Target node is busy. NAK No Memory retries by link layer exhausted.
03H	Target node cannot respond because message is too large.
04H	Target node cannot respond because it does not understand the command parameters OR the control block may have been inadvertently modified.
05H	Local processor is off-line (possible duplicate node situation).
06H	Target node cannot respond because requested function is not available.
07H	Target node does not respond.
08H	Target node cannot respond.
09H	Local modem connection has been lost.
0BH	Target node does not accept this type of MSG instruction.

MSG Instruction Error Codes

Error Code	Description of Error Condition
0CH	Received a master link reset (one possible source is from the DF1 master).
0FH	DCOMM button was activated while an ASCII instruction was waiting to execute.
10H	Target node cannot respond because of incorrect command parameters or unsupported command.
12H	Local channel configuration protocol error exists.
13H	Local MSG configuration error in the Remote MSG parameters.
15H	Local channel configuration parameter error exists.
16H	Target or Local Bridge address is higher than the maximum node address.
17H	Local service is not supported.
18H	Broadcast is not supported.
20H	PCCC Description: Host has a problem and will not communicate.
21H	Bad MSG file parameter for building message.
30H	PCCC Description: Remote station host is not there, disconnected, or shutdown.
37H	Message timed out in local processor.
39H	Local communication channel reconfigured while MSG active.
3AH	STS in the reply from target is invalid.
40H	PCCC Description: Host could not complete function due to hardware fault.
45H	MSG reply cannot be processed. Either Insufficient data in MSG read reply or bad network address parameter.
50H	Target node is out of memory.
60H	Target node cannot respond because file is protected.
70H	PCCC Description: Processor is in Program Mode.
80H	PCCC Description: Compatibility mode file missing or communication zone problem.
81H	Modbus Error 1: Illegal Function
82H	Modbus Error 2: Illegal Data Address
83H	Modbus Error 3: Illegal Data Value
84H	Modbus Error 4: Slave Device Failure
85H	Modbus Error 5: Acknowledge
86H	Modbus Error 6: Slave Device Busy
87H	Modbus Error 7: Negative Acknowledge
88H	Modbus Error 8: Memory Parity Error
89H	Modbus Error: Non-standard reply. Actual code returned can be found in the upper byte of sub-element 22.
90H	PCCC Description: Remote station cannot buffer command.
B0H	PCCC Description: Remote station problem due to download.
C0H	PCCC Description: Cannot execute command due to active IPBs.
D0H	No IP address configured for the network, or Bad command - unsolicited message error, or Bad address - unsolicited message error, or No privilege - unsolicited message error, or Multihop messaging cannot route request
D1H	Maximum connections used - no connections available.
D2H	Invalid internet address or host name.

MSG Instruction Error Codes

Error Code	Description of Error Condition
D3H	No such host exists.
D4H	Cannot communicate with the name server.
D5H	Connection not completed before user–specified timeout.
D6H	Connection timed out by the network.
D7H	Connection refused by destination host.
D8H	Connection was broken.
D9H	Reply not received before user-specified timeout.
DAH	No network buffer space available.
DBH	Multi-hop messaging CIP message format error.
DCH	Class 3 CIP connections are duplicated for same IP address.
DDH	SMTP General Error Code. The error code returned can be found in the upper byte of sub-element 22.
DEH	CIP Object Specific General error code. The error code returned can be found in the upper byte of sub-element 22.
DFH	Multi-hop messaging has no IP address configured for network.
E0H	Expansion I/O Communication Module Error or CIP device response error code. The error code returned can be found in the upper byte of sub-element 22.
E1H	PCCC Description: Illegal Address Format, a field has an illegal value.
E2H	PCCC Description: Illegal Address format, not enough fields specified.
E3H	PCCC Description: Illegal Address format, too many fields specified.
E4H	PCCC Description: Illegal Address, symbol not found.
E5H	PCCC Description: Illegal Address Format, symbol is 0 or greater than the maximum number of characters support by this device.
E6H	PCCC Description: Illegal Address, address does not exist, or does not point to something usable by this command.
E7H	Target node cannot respond because length requested is too large.
E8H	PCCC Description: Cannot complete request, situation changed (file size, for example) during multi–packet operation.
E9H	PCCC Description: Data or file is too large. Memory unavailable.
EAH	PCCC Description: Request is too large; transaction size plus word address is too large.
EBH	Target node cannot respond because target node denies access.
ECH	Target node cannot respond because requested function is currently unavailable.
EDH	PCCC Description: Resource is already available; condition already exists.
EEH	PCCC Description: Command cannot be executed.
EFH	PCCC Description: Overflow; histogram overflow.
F0H	PCCC Description: No access.
F1H	Local processor detects illegal target file type.
F2H	PCCC Description: Invalid parameter; invalid data in search or command block.
F3H	PCCC Description: Address reference exists to deleted area.
F4H	PCCC Description: Command execution failure for unknown reason; PLC-3 histogram overflow.
F5H	PCCC Description: Data conversion error.
F6H	PCCC Description: The scanner is not able to communicate with a 1771 rack adapter. This could be due to the scanner not scanning, the selected adapter not being scanned, the adapter not responding, or an invalid request of a “DCM BT (block transfer)”.
F7H	PCCC Description: The adapter is not able to communicate with a module.

MSG Instruction Error Codes

Error Code	Description of Error Condition
F8H	PCCC Description: The 1771 module response was not valid size, checksum, etc.
F9H	PCCC Description: Duplicated Label.
FAH	Target node cannot respond because another node is file owner (has sole file access).
FBH	Target node cannot respond because another node is program owner (has sole access to all files).
FCH	PCCC Description: Disk file is write protected or otherwise inaccessible (off-line only).
FDH	PCCC Description: Disk file is being used by another application; update not performed (off-line only).
FFH	Local communication channel is shut down.

TIP For 1770-6.5.16 DF1 Protocol and Command Set Reference Manual users: The MSG error code reflects the STS field of the reply to your MSG instruction.

- Codes E0 to EF represent EXT STS codes 0 to F.
- Codes F0 to FC represent EXT STS codes 10 to 1C.

Special Function with MSG instruction

MicroLogix 1400 supports the configuration of IP Address, Subnet Mask, Gateway Address, Default Domain Name, Primary Name Server, and Secondary Name Server in the Ethernet Channel Configuration File via Ethernet MSG instruction.

MicroLogix 1400 also supports SMTP Email. Using the SMTP Configuration in the Channel Configuration and MSG instruction, the MicroLogix 1400 can send SMTP messages to an Email Server. MicroLogix 1400 Series B controllers support the configuration of the SMTP Configuration File via the Ethernet MSG instruction.

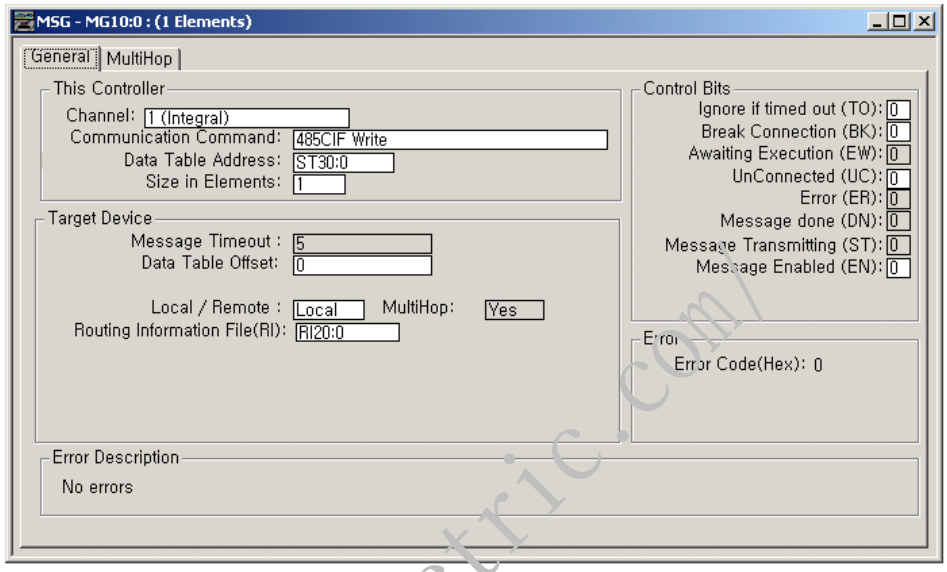
These features are enabled by sending the 485CIF write Write message to the local IP Address with ST data file type.

Ethernet Channel Configuration Change Functionality

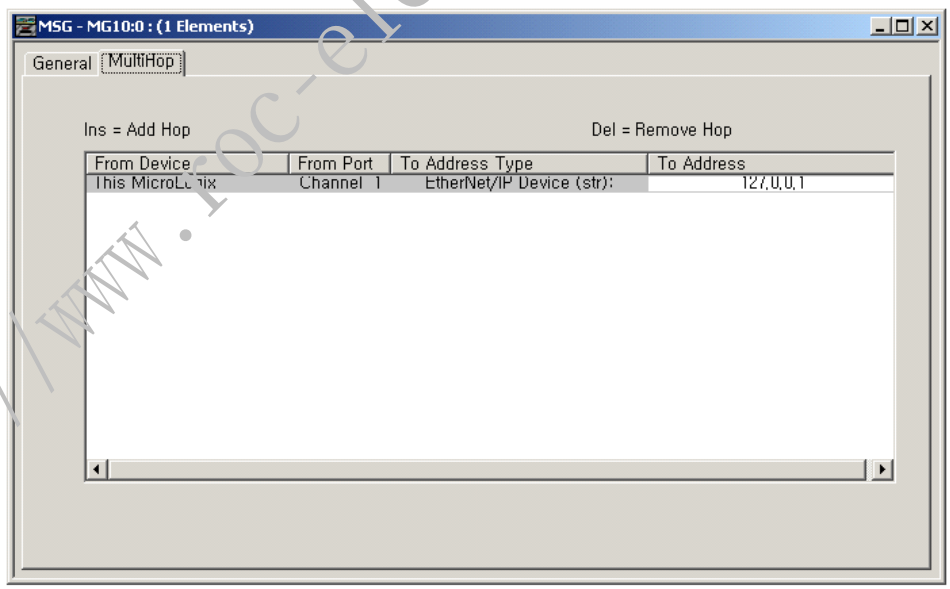
You can use a MSG instruction to change the Ethernet Channel Configuration.

Set up MSG for Changing Channel Configuration

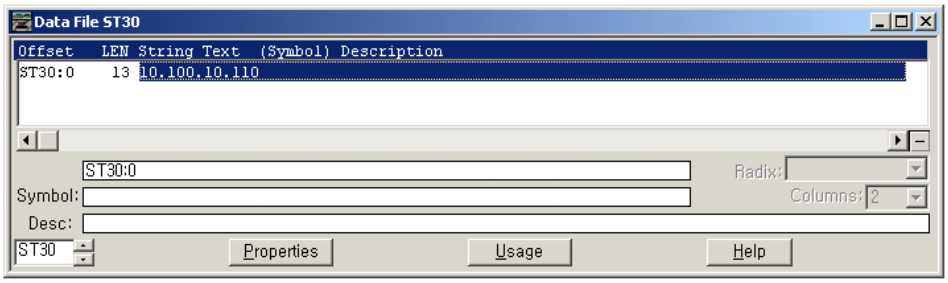
General MSG Setup Screen to change IP Address



MultiHop Setup Screen to change IP Address



Setup String Data File



Parameters for configuration:

- **Channel** : Channel must be "1 (Integral)".
- **Communication Command** : Communication Command must be "485 CIF Write".
- **Data Table Address** : Data Table Address must be a String file. To change Ethernet Channel Configuration, you must enter the characters for the configuration parameter.
- **Size in Elements** : Size in Element must be 1.
- **Message Timeout** : Not editable.
- **Data Table Offset** : Valid value is 0...5, 10, and 40 for Ethernet Channel Configuration. For Series B controllers, 50...53 and 60...69 for SMTP Configuration. Basically, "Data Table Offset" in MSG configuration setup screen is used to direct the "internal virtual offsets". You can configure the listed parameter by sending a String File Data to these offsets.

Data Table Offset Addressing to change Channel Configuration parameters

Data Table Offset	Affect to; Description	Comments
0	Ethernet IP Addres	Writable by an element of String File. Changes IP Address in Ethernet Communication File. Disables BOOTP/DHCP flags in Ethernet Communication File. Power cycle is required.
1	Ethernet Subnet Mask	Writable by an element of String File. Changes Subnet Mask in Ethernet Communication File. Disables BOOTP/DHCP flags in Ethernet Communication File. Power cycle is required.
2	Ethernet Gateway Address	Writable by an element of String File. Changes Default Gateway Address in Ethernet Communication File. Disables BOOTP/DHCP flags in Ethernet Communication File. Power cycle is required.
3	Ethernet Default Domain Name	Writable by an element of String File. Used for DNS naming in SMTP subsystem.
4	Ethernet Primary Name Server	Writable by an element of String File. Used for DNS naming in SMTP subsystem.
5	Ethernet Secondary Name Server	Writable by an element of String File. Used for DNS naming in SMTP subsystem.
-	-	-
10	Apply IP Address, Subnet Mask, and Default Gateway Address right away.	Applies IP Address, Subnet Mask, and Default Gateway Address configured by Offset 0, 1, and 2 right away. These parameters are updated to the Ethernet Status File also if they are applied. String File configured in MSG instruction will be ignored.
11	DNP3 Unsolicited Channel Switching	To change unsolicited channel, you must enter the characters 0, 1, or 2 (that is, 0 = Channel 0; 1 = Channel 1; 2 = Channel 2) in string data file.
-	-	-
40	Flush DNS Cache	Flushes all DNS names in DNS Cache before TTL (Time to Live) timeout. String File configured in MSG instruction will be ignored.
-	-	-
50	Email Server	Writable by an element of String File. Changes Email Server in the SMTP Configuration File.
51	From Address	Writable by an element of String File. Changes Email Server in the SMTP Configuration File.
-	Authentication	To set Enable or Disable state for Authentication, change the Password. For more details, see Password below.
52	User Name	Writable by an element of String File. Changes User Name in the SMTP Configuration File.

Data Table Offset Addressing to change Channel Configuration parameters

53	Password	Writable by an element of String File. Changes Password in the SMTP Configuration File. When the string size of Password is 0, then Authentication is disabled. When the string size of the Password is not 0, then Authentication is enabled.
-	-	-
60	TO Address [0]	Writable by an element of String File. Changes TO Address [0] in the SMTP Configuration File.
61	TO Address [1]	Writable by an element of String File. Changes TO Address [1] in the SMTP Configuration File.
62	TO Address [2]	Writable by an element of String File. Changes TO Address [2] in the SMTP Configuration File.
63	TO Address [3]	Writable by an element of String File. Changes TO Address [3] in the SMTP Configuration File.
64	TO Address [4]	Writable by an element of String File. Changes TO Address [4] in the SMTP Configuration File.
65	TO Address [5]	Writable by an element of String File. Changes TO Address [5] in the SMTP Configuration File.
66	TO Address [6]	Writable by an element of String File. Changes TO Address [6] in the SMTP Configuration File.
67	TO Address [7]	Writable by an element of String File. Changes TO Address [7] in the SMTP Configuration File.
68	TO Address [8]	Writable by an element of String File. Changes TO Address [8] in the SMTP Configuration File.
69	TO Address [9]	Writable by an element of String File. Changes TO Address [9] in the SMTP Configuration File.

- **Local / Remote** : "Local / Remote" has no impact on the operation.
- **MultiHop** : Cannot edit.
- **Routing Information File** : Routing Information File must be RI data.
- **Break Connection (BK)** : This bit has no impact on the operation.
- **IP Address of MultiHop** : IP Address of MultiHop must be local IP Address (127.0.0.1 or its own IP Address).

Considerations for Changing Ethernet Channel Configuration

TIP If you sent a message with Data Table Offset 0, 1, or 3 to change IP Address, Subnet Mask, or Gateway Address respectably, these addresses will be applied after power-cycle. If you want to change IP Address, Subnet Mask, and Gateway Address right away, you must send another Ethernet message to the local IP Address with Data Table Offset 10.

ATTENTION: If you sent an Ethernet message to the local IP Address with Data Table Offset 10 and the different IP Address has been configured, all the Ethernet connection is broken.

Considerations for Changing SMTP Configuration

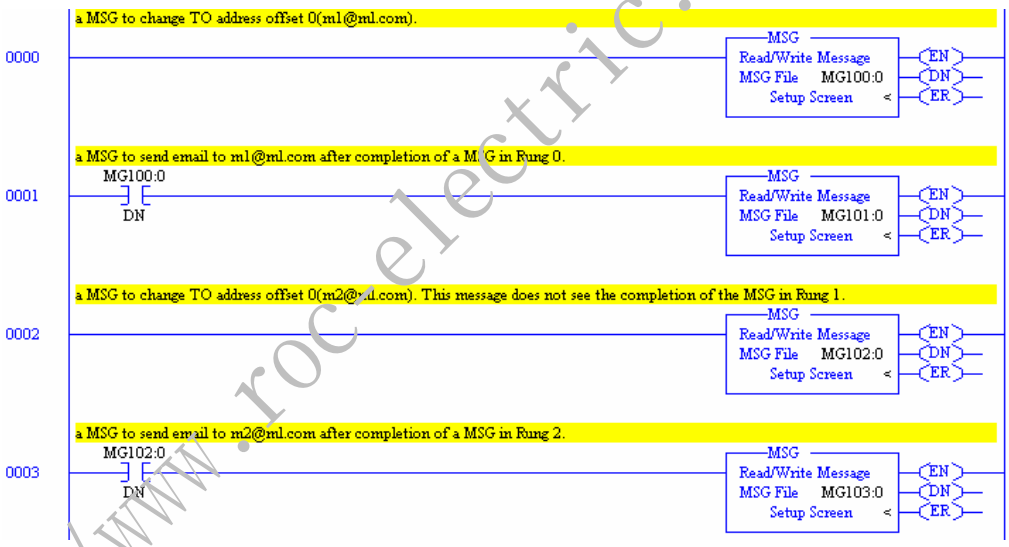
The maximum string size in SMTP configuration file is 62 bytes. MSG returns an error if the string size exceeds 62bytes.

For the change of the Email Server, once an email is triggered, a TCP connection between Email Server and the controller is kept. In this case, although Email

Server in the SMTP configuration file was changed by MSGs, SMTP subsystem does not disconnect the TCP connection. So, newly configured Email Server is not applied immediately. In this case, you can use BK bit in the MSG instruction that triggers email so that the TCP connection is closed immediately after the completion of the SMTP message.

If a parameter in the SMTP configuration file needs to be changed, every MSG instruction must be triggered serially. This means that the next MSG with a parameter modification must not be triggered before the previous MSG e-mail is complete. This is because SMTP protocol sends several packets to send a single email.

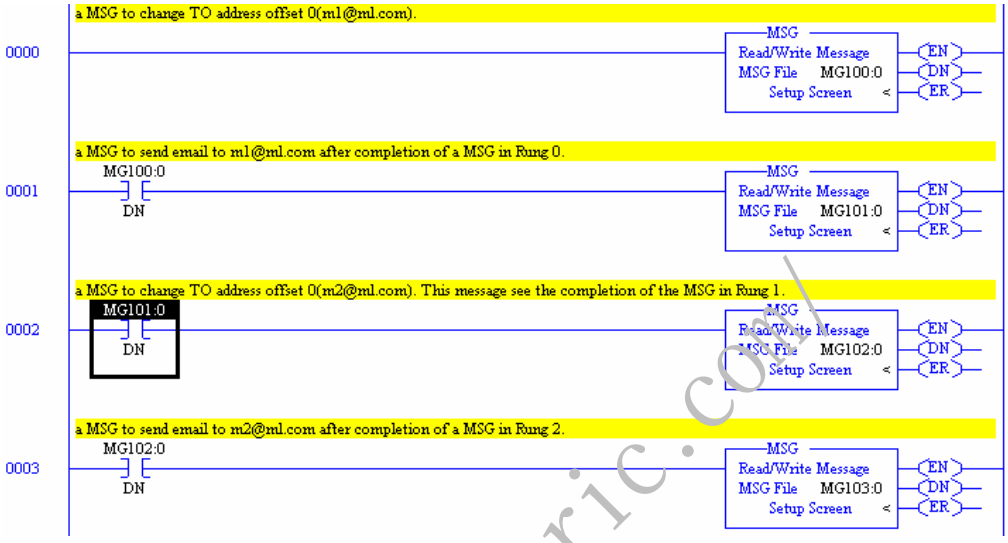
For example, if you need two consecutive MSG instructions for sending emails after each MSG for changes to TO address offset 0, you might write ladder programs as below.



In this case, rung 0 and 2 are executed almost at the same time. And rung 1 may not be started or completed, since it needs several transactions between the Email Server and the controller to completely send a single email.

The MSG in rung 1 may send the email to the second address m2@ml.com. To avoid this situation, the MSG instruction in rung 2 must not be executed before the completion of the MSG in rung 1. The MSG in rung 2 must see the

completion (DN bit set) of the MSG in rung 1. The following ladder program are more correct.



Email Functionality

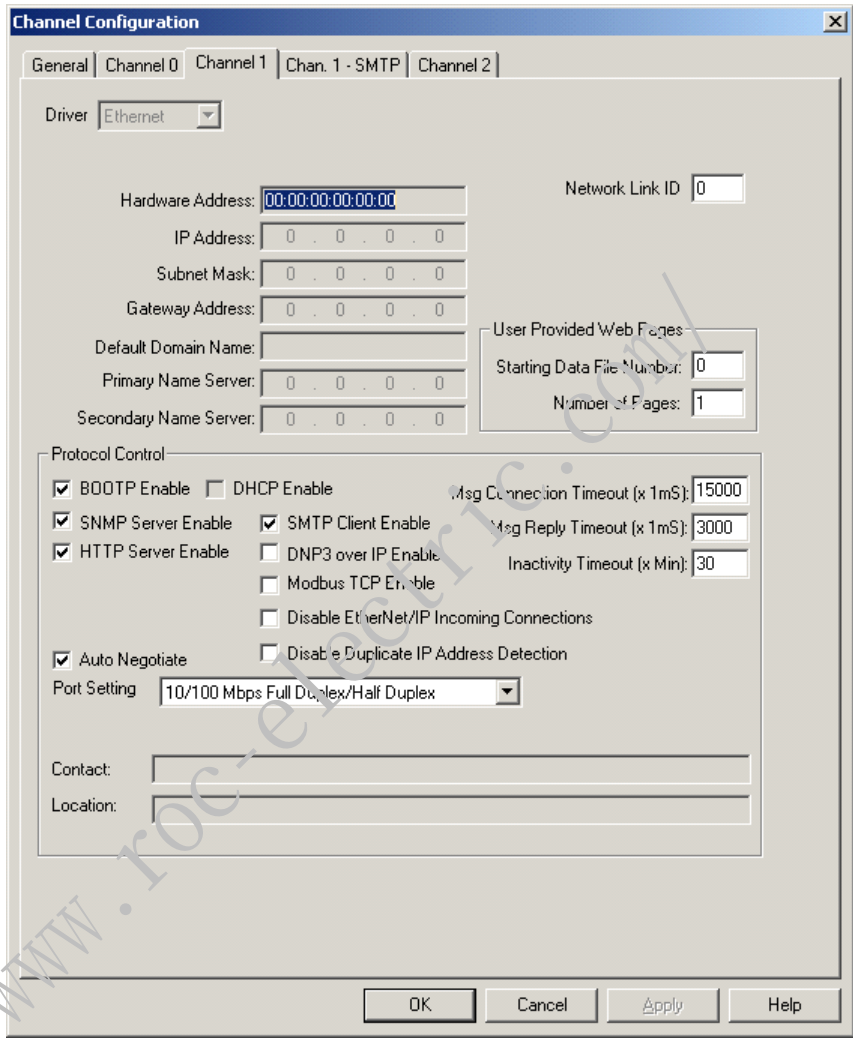
This section describes how to configure a SMTP email message when you use Ethernet communication channel 1 of the MicroLogix 1400.

Setup SMTP Configuration File

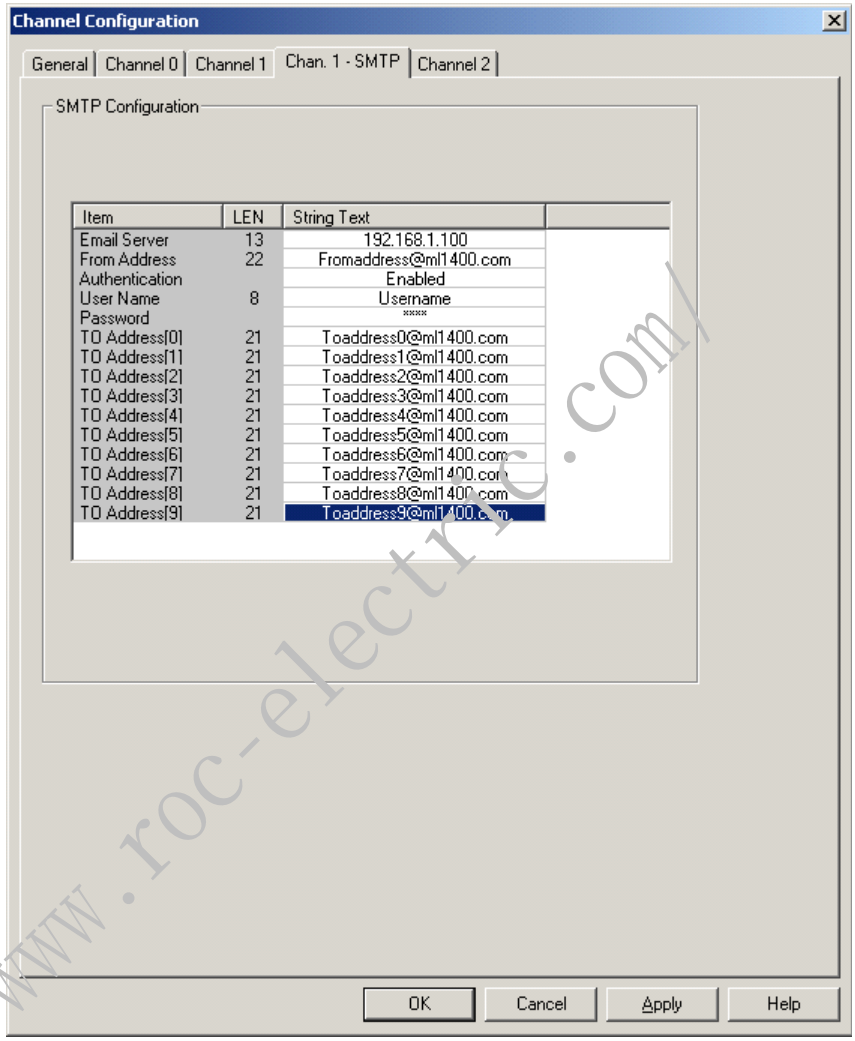
While the processor is selected to MicroLogix 1400, “SMTP Client Enable” check box is shown in the Ethernet Channel Configuration tab. If this check box is checked, SMTP configuration page appears. Otherwise, SMTP configuration page does not appear.

You must set up SMTP configuration before sending SMTP messages.

SMTP Client Enable Bit Setup Screen



SMTP Configuration Setup Screen



The following is an explanation of parameters to be configured:

- **Email Server** : Email Server IP Address or DNS name. For input of the DNS name, make sure that the Default Domain Name has been configured in Ethernet Channel Configuration, where the DNS Sub-system will query total DNS name to DNS server. For example, if “dnsname” is entered, and “default.com” is configured in the Default Domain Name of the Ethernet Channel Configuration, the DNS sub-system will query “dnsname.default.com” to the DNS server.
  In the MicroLogix 1400 Series B controllers, the configuration for alternate SMTP Port number is supported. The default SMTP Port number in the MicroLogix 1400 is 25 in TCP. To specify the target IP address and port number in the Email Server configuration, you can enter them in the following format.

  HOSTNAME>
  <IPADDR>

<HOSTNAME>?port=<xxx>
<IPADDR>?port=<xxx>

For example:

REMOTE_MOD: equivalent to REMOTE_MOD?port=25
192.168.1.100: equivalent to 192.168.1.100?port=25
REMOTE_MOD?port=28123
192.168.1.100?port=28123

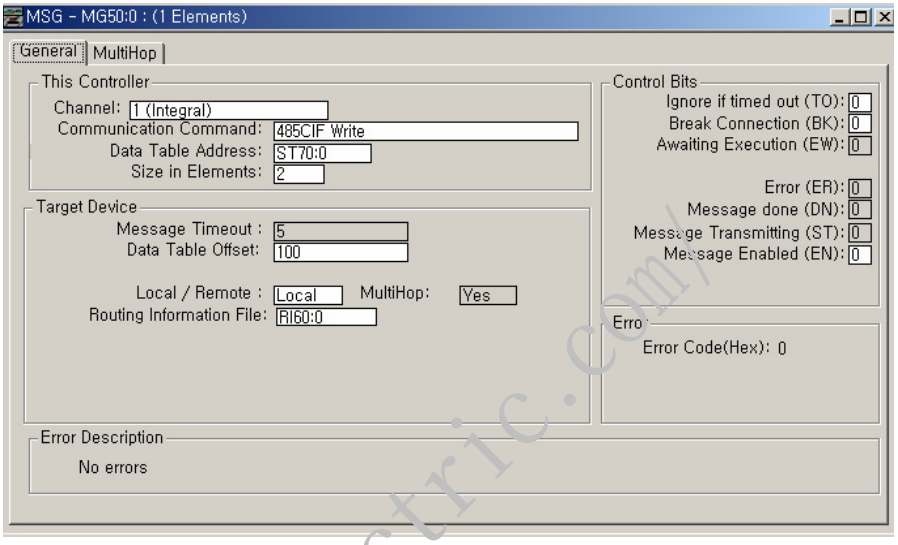
- **FROM Address**: Email From Address. From Address must be written as an email style.
- **Authentication**: Disable or Enable. Disable is a default value. If this flag is disabled, "User Name" and "Password" tab is changed to non-editable.
- **User Name**: User Name registered in the SMTP server. According to the SMTP server, Username must be written as an email style.
- **Password**: Password registered in the SMTP server.
- **TO Address [0]**: Email TO address [0]. To Address must be written as an email style.
- **TO Address [1]**: Email TO Address [1]. To Address must be written as an email style.
- **TO Address [2]**: Email TO Address [2]. To Address must be written as an email style.
- **TO Address [3]**: Email TO Address [3]. To Address must be written as an email style.
- **TO Address [4]**: Email TO Address [4]. To Address must be written as an email style.
- **TO Address [5]**: Email TO Address [5]. To Address must be written as an email style.
- **TO Address [6]**: Email TO Address [6]. To Address must be written as an email style.
- **TO Address [7]**: Email TO Address [7]. To Address must be written as an email style.
- **TO Address [8]**: Email TO Address [8]. To Address must be written as an email style.
- **TO Address [9]**: Email TO Address [9]. To Address must be written as an email style.

These parameters are non-editable in RUN mode. You can change them in offline or online PROGRAM mode. Maximum character length for the string parameters is 62 bytes. LEN (=Length) fields in the SMTP Configuration File are not editable. If String Text is entered, RSLogix500 updates the Length fields automatically.

Configure MSG Setup Screen to send SMTP message

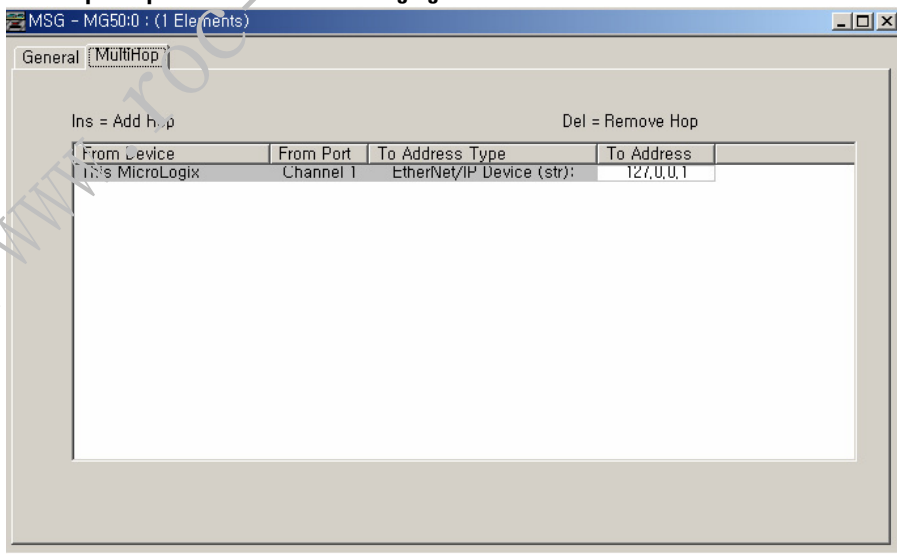
The picture below provides an example of how the MSG Instruction would be configured to be used to send SMTP message.

General MSG Setup Screen for SMTP messaging

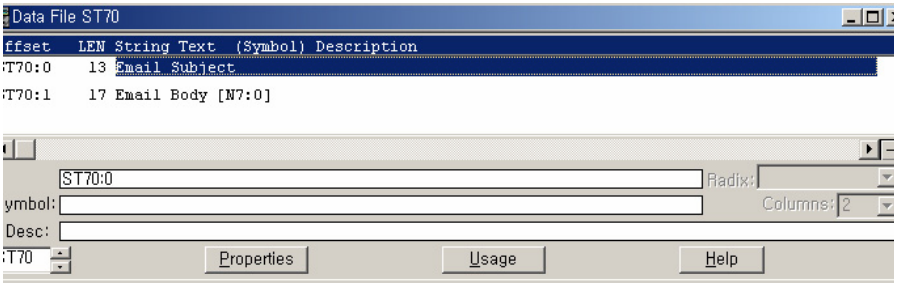


The picture below shows MultiHop setup screen and configured E-mail Subject and Body using an ST file.

MultiHop Setup Screen for SMTP messaging



Setup String Data File



An explanation of the parameters is provided here.:

- **Channel** : Channel must be "1 (Integral)".
- **Communication Command** : Communication Command must be "485 CIF Write".
- **Data Table Address** : Data Table Address must be a String file and the String file must contain Email Subject and Email Body. If Data Table Address is not a String File, the contents of email subject are filled to "No Subject" and that of email body are filled to "No Body".
- **Size in Elements** : Size in Element must be 1 or 2. To send SMTP messages, Size in Elements must be 1 or 2. The first string element (Offset 0 of String File) is a subject and the second string element (Offset 1 of String File) is an email body. If Size in Element is 1, the contents of email body are filled to "No Body".
- **Message Timeout** :Not editable.
- **Data Table Offset** : Valid value is 100...109 for SMTP messaging. Basically, "Data Table Offset" in MSG configuration setup screen is used to direct the "internal virtual offsets". You can configure the listed parameter setting by sending a String File Data to these offsets.

Data Table Offset Addressing for SMTP messaging

Data Table Offset	Affect to; Description	Comments
100	Send email to SMTP TO address [0] configured in SMTP configuration File.	Used to trigger the SMTP message
101	Send email to SMTP TO address [1] configured in SMTP configuration File.	Used to trigger the SMTP message
102	Send email to SMTP TO address [2] configured in SMTP configuration File.	Used to trigger the SMTP message
to	...	...
108	Send email to SMTP TO address [8] configured in SMTP configuration File.	Used to trigger the SMTP message
109	Send email to SMTP TO address [9] configured in SMTP configuration File.	Used to trigger the SMTP message

- **Local / Remote** : "Local / Remote" has no impact on the operation.

- **MultiHop** : Not editable.
- **Routing Information File** : Routing Information File must be RI data.
- **Break Connection (BK)** : If this bit is cleared for SMTP messaging, the connection with the SMTP Server is not closed after the SMTP message is sent out to the SMTP Server. If this bit is set, the connection is closed.
- **IP Address of MultiHop** : IP Address of MultiHop must be local IP Address (127.0.0.1 or its own IP Address).

SMTP Error Codes/Internal Fail Codes

When the SMTP subsystem cannot send an email, the error code is shown via MSG instruction. Error Code 0xDD is stored in Word 18 of the MG file (MGx:y.ERR). Internal Fail Code is stored in Word 22 of MG file (MGx:y.22). When messaging through SMTP communication and the low byte is 0xDD, the high byte of this sub-element contains detailed Fail Code returned by the SMTP subsystem.

The element values of Fail Codes are shown below.

Fail Codes in MG File

Fail Code in MG file (MGx:y.22	Comment
0x0000	Delivery successful to the mail relay server.
0x01DD	SMTP mail server IP Address not configured.
0x02DD	To (destination) Address not configured or invalid.
0x03DD	From (reply) Address not configured or invalid.
0x04DD	Unable to connect to SMTP mail server.
0x05DD	Communication error with SMTP server.
0x06DD	Authentication required.
0x07DD	Authentication failed.
0x10DD	SMTP Configuration File does not exist.

Inline Indirection in String File for Subject and Body

For E-mail subject and body, Inline Indirection functionality can be used. In the previous General MSG setup screen, E-mail subject is ST70:0 and body is ST70:1. If "SMTP BODY 0 [N7:0]" is written in the String File, "[N7:0]" is replaced by a String of the value N7:0.

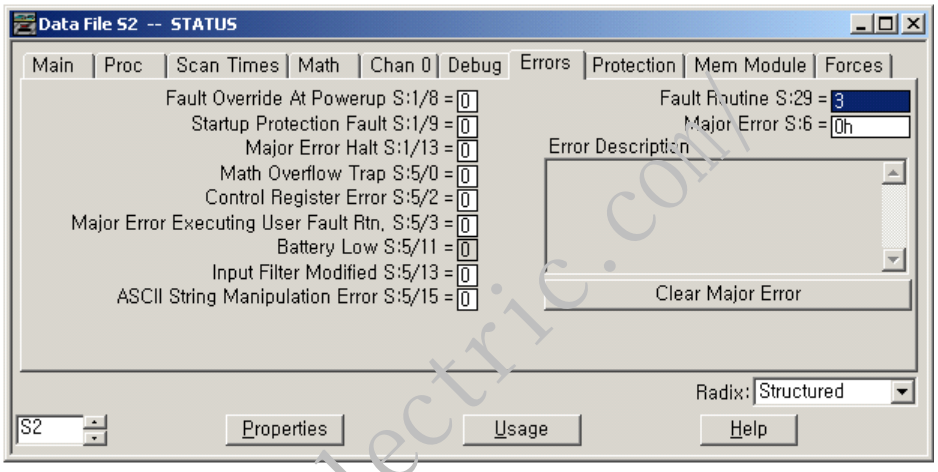
SMTP Authentication Encoding

If the SMTP Authentication Flag is enabled, MicroLogix 1400 sends encoded Username and Password. There are several Authentication methods; login, plain, cram-md5, and so on. However, MicroLogix 1400 supports only the login method. Base64 encoding is used to log into the SMTP server.

Sending email in User Fault Routine

When the controller mode is changed to User Fault mode, the user fault routine must be defined in the word 29 (Fault Routine S2:29) of System Status File before e-mails can be sent. The SMTP MSG instruction must be used in the configured Fault Routine as well.

Configuration for sending email in UFR



Notes:

Modbus TCP

With the Modbus TCP feature, you can use a MicroLogix 1400 controller to communicate via Modbus TCP protocol.

The Modbus TCP feature is supported in the MicroLogix 1400 Series B controllers only.

Modbus TCP Architecture

MicroLogix 1400 Series B controllers support both the TCP Server and Client features. Modbus TCP Client takes over Modbus Master features and Modbus TCP Server takes over Modbus Slave features on the Ethernet.

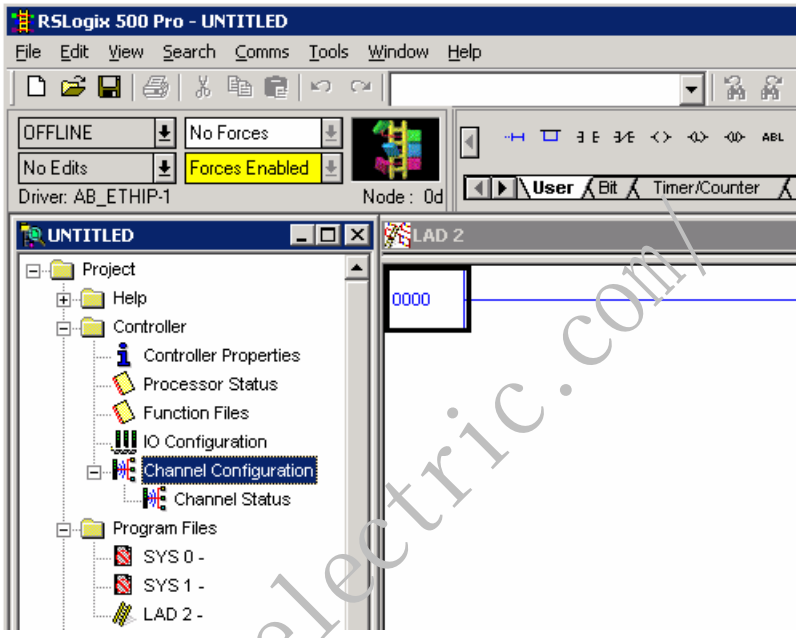
Both Modbus TCP Server and Client are enabled through Channel Configuration. For Modbus TCP Client, a MSG instruction is used to send or receive the Modbus TCP packets.

For Modbus TCP Server, the MicroLogix 1400 supports up to 16 connections at the same time. For Modbus TCP Client, 16 connections are supported.

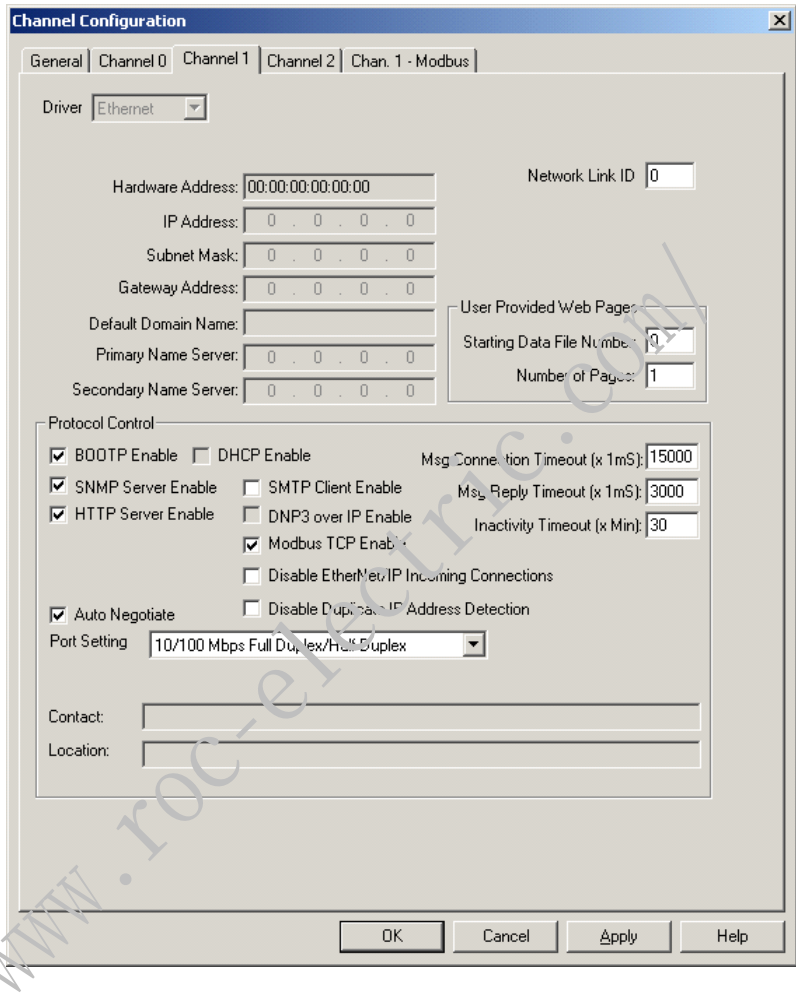
Channel Configuration for Modbus TCP

The default communication protocol for the Ethernet Channel 1 in the MicroLogix 1400 is Ethernet/IP. To communicate with Modbus TCP protocol in the MicroLogix 1400 Series B controller, the channel must be configured for Modbus TCP protocol.

Programming the MicroLogix 1400 Series B controller is done using RSLogix 500/RSLogix Micro, version 8.30.00 or later. In RSLogix 500/RSLogix Micro, click the MicroLogix 1400 Series B project tree.



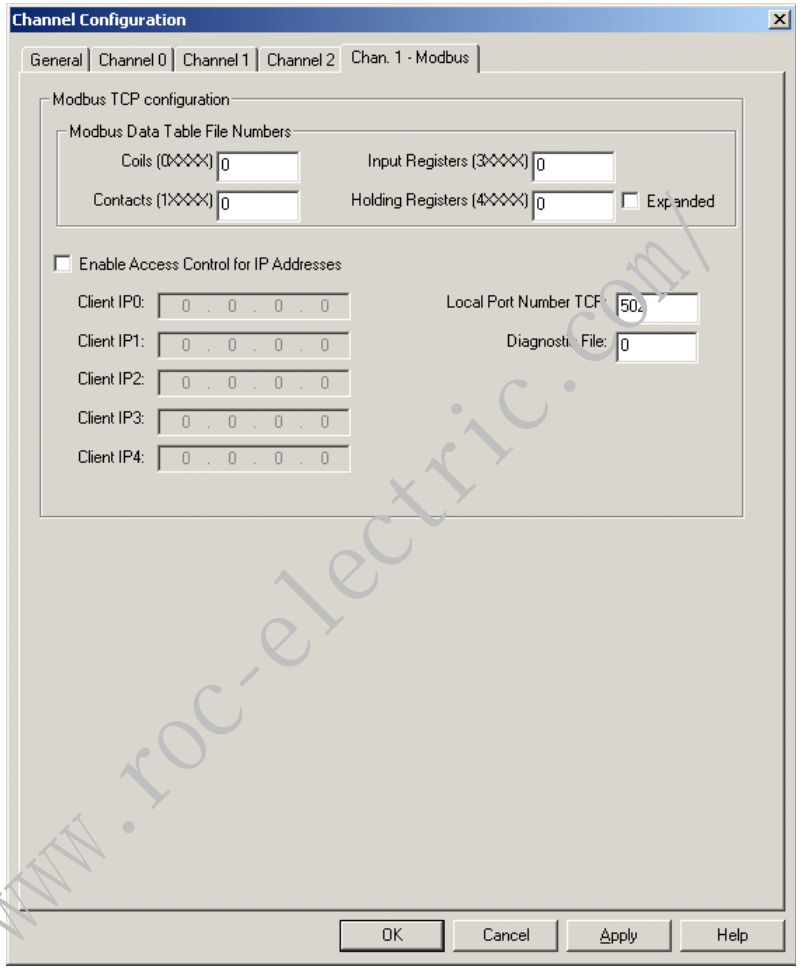
To enable Modbus TCP protocol, select Modbus TCP Enable in the Channel 1 configuration.



Unlike serial port configuration, you must cycle power to the controller after downloading the Ethernet port configuration to enable Modbus TCP.

Modbus TCP Server Configuration

Modbus TCP Server configuration can be done in the Chan. 1 - Modbus tab.



Modbus TCP Server Configuration Parameters

This configuration is for the Modbus TCP Server subsystem in MicroLogix 1400 controllers. The parameter Modbus TCP Enable is configured in the Channel 1 tab and other parameters are configured in the Chan. 1 - Modbus tab.

Modbus TCP Enable

The valid selections are Enabled (Checked) and Disabled (Unchecked). Default value is Disabled (Unchecked). A power cycle is required for changes to take effect.

Modbus Data Table File Numbers

The parameters File Numbers for Coils, Contacts, Input Registers and Holding Registers are configurable, with B or N files only. Valid range is 0, 3, 7, 9...255. Default value is 0.

See Modbus Slave Memory Map on page 561 for more details.

Expanded for Holding Registers

The valid selections are Enabled (Checked) and Disabled (Unchecked). Default value is Disabled (Unchecked).

When the selection is Disabled (Unchecked), a single file is allocated for the Holding Registers that are configured.

When the selection is Enabled(Checked), multiple files are allocated for the Holding Registers that are configured successively.

See Modbus RTU Slave Configuration on page 559 for more details.

Enable Access Control for IP Addresses

The valid selections are Enabled (Checked) and Disabled (Unchecked). Default value is Disabled (Unchecked).

When the selection is Disabled (Unchecked), MicroLogix 1400 accepts the requests from any Modbus TCP Client.

When the selection is Enabled (Checked), MicroLogix 1400 accepts the requests only from the Modbus TCP Client IP Address which is configured in the parameters Client IP Address0 to Client IP Address4. The maximum number of Client IP Address for Access Control is 5.

Client IP Address0...Client IP Address4

This value is used for validation of the Client IP address when the Enable Access Control for IP Addresses is Enabled (Checked). This value is only valid when the Enable Access Control for IP Addresses is Enabled (Checked).

The valid value is an IP address. Default value is 0.0.0.0.

Local TCP Port Number

This value is used to configure Local TCP Port Number, which is used for TCP socket listening.

The valid range is 0 to 65535. Default value is 502.

Diagnostic File Number

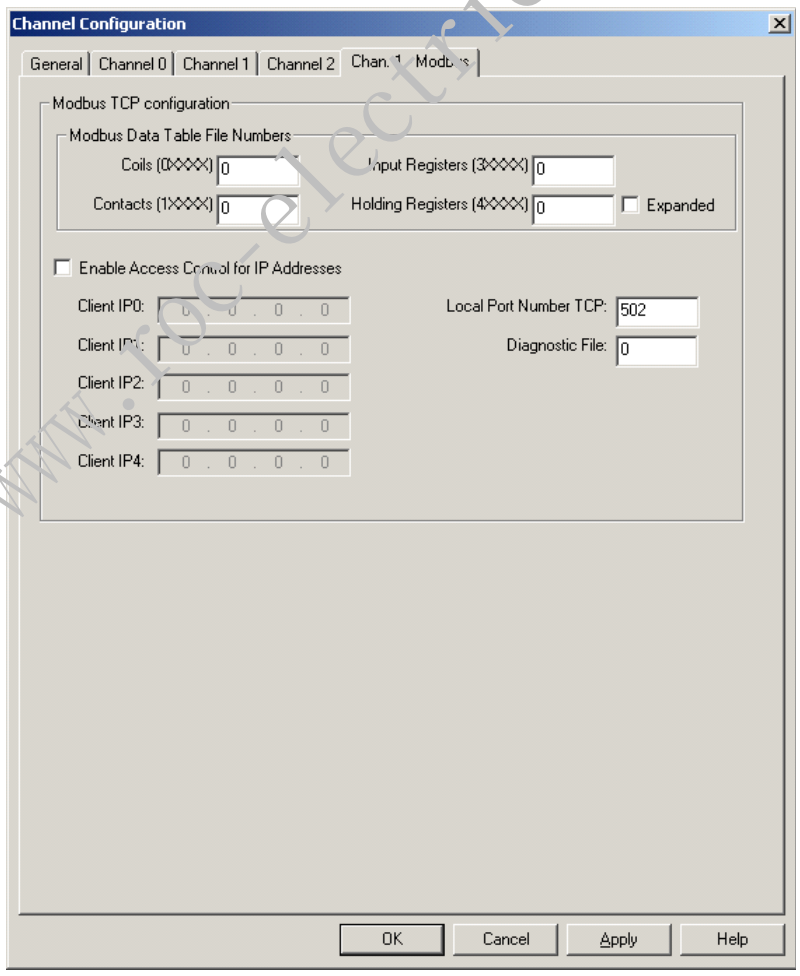
The diagnostic file number is used to store the diagnostics for troubleshooting the Modbus TCP Ethernet subsystem. The status of Modbus TCP Server and Client subsystem is stored in this data file.

The value of this parameter is N file only. Valid range is 0, 7, 9...255. Default value is 0.

See Diagnostics for Modbus TCP on page 430 for troubleshooting information.

Modbus TCP Client Configuration

Modbus TCP Client configuration can be done in the Chan. 1 - Modbus tab.



Modbus TCP Client Configuration Parameters

This configuration is for Modbus TCP Client subsystem in MicroLogix 1400 controllers. The parameter Modbus TCP Enable is configured in the Channel 1 tab and other parameters are configured in the Chan. 1 - Modbus tab. The only parameter in the Chan. 1 - Modbus tab for Modbus TCP Client is the parameter Diagnostic File Number.

Modbus TCP Enable

The valid selections are Enabled (Checked) and Disabled (Unchecked). Default value is Disabled (Unchecked). A power cycle is required for changes to take effect.

Diagnostic File Number

The diagnostic file number is used to store the diagnostics for the troubleshooting of Modbus TCP Ethernet subsystem. The status of Modbus TCP Server and Client subsystem is stores to this data file.

The value of this parameter is N file only. Valid range is 0, 7, 9...255. Default value is 0.

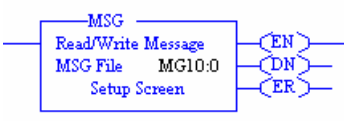
See Diagnostics for Modbus TCP on page 430 for troubleshooting information

Messaging for Modbus TCP Client

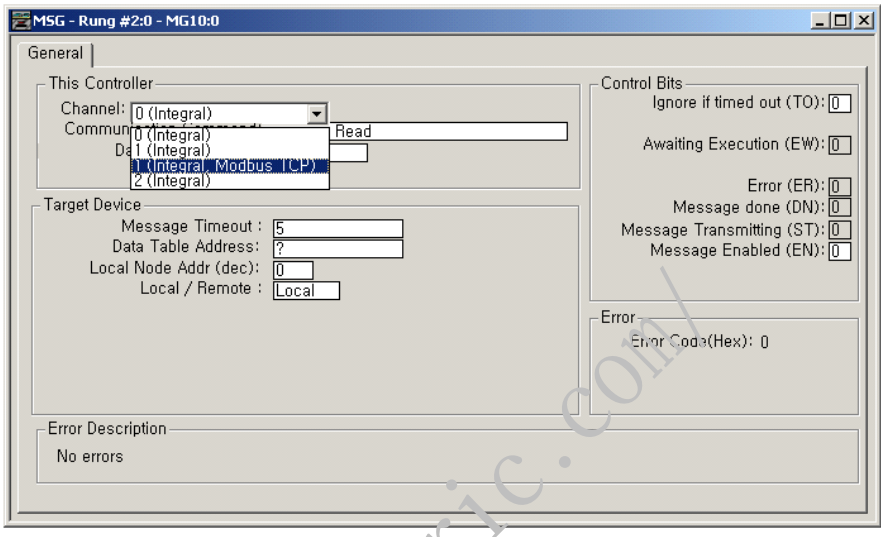
In MicroLogix 1400 controller ladder programs, use a MSG instruction to request or receive a Modbus TCP packet.

MSG Configuration Parameters

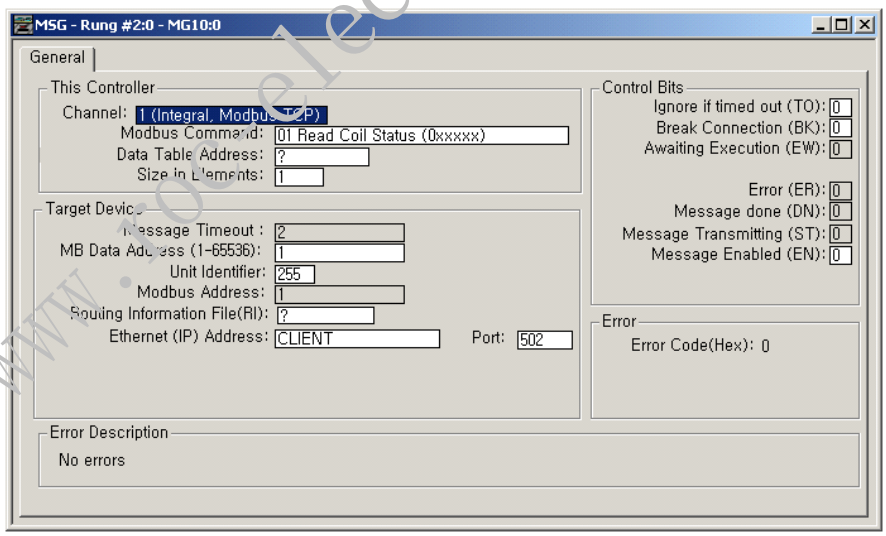
1. In a new MSG instruction, double-click Setup Screen in the MSG instruction.



2. Select Integral, Modbus TCP from the Channel drop-down list.



The following window appears:



You can see configurable parameters that are similar to the ones for Modbus Master MSG instruction over Serial communication. These parameters are described in the following table.

Modbus TCP MSG Parameter

Modbus TCP MSG Parameter	Description
Message Timeout	The default value is 2 seconds. However, once this MSG instruction is activated, it is changed to 33 seconds by the controller.
Unit Identifier	This is used in the identification of a remote slave that is connected on a serial line or on other buses. Default value is 255. Use this default value if the target device is not a gateway.
Routing Information File (RI)	Specify an RI file.
Ethernet (IP) Address	Enter the IP Address of the target device.
Port	Enter the TCP Port Number of the target device. Default value is 502.
Control Bits (TO, BK, EN)	TO bit: If this bit is set, the MSG instruction will cause message timeout and set error bit. BK bit: If this bit is set, the TCP connection is closed after the Modbus TCP message is sent out to the Target Device. EN bit: If this bit is set, the MSG instruction will be triggered.
Error Code (Hex)	Error Codes as described for Ethernet/IP messaging. See MSG Instruction Error Codes on page 404.
Error Description	Error Code descriptions as described for Ethernet/IP messaging. See MSG Instruction Error Codes on page 404.

For more information, see:

- The Message Element on page 342
- Modbus RTU Master on page 555

Message Instruction Timeouts

The default MSG instruction timeout is 33 seconds; the maximum MSG timeout is approximately 146 seconds. Specify the MSG instruction timeout by setting the appropriate configuration in Ethernet Channel Configuration:

- Msg Connection Timeout : up to 65500 milliseconds.
- Msg Reply Timeout : up to 65500 milliseconds.

The MSG timeout is determined by adding the Msg Connection Timeout, Msg Reply Timeout and Default addition time (15 seconds).

Change between Executing and Non-Executing Controller Modes

Executing modes include Run, Remote Run, Test Continuous Scan, and Test Single Scan modes. All others are Non-Executing modes.

If the MicroLogix controller transitions from Executing to Non-Executing mode while Modbus TCP MSG requests are active, all connections are forced closed by the controller.

In the RSLogix 500/RSLogix Micro ladder program, you can also set the MGx:y.TO bit for any outstanding Modbus TCP MSG instructions. This causes the MSG instruction to timeout and set the MGx:y.ER bit.

Diagnostics for Modbus TCP

Diagnostic Counters and Errors in the Modbus TCP subsystem for the Ethernet channel are logged in the N Data File. The data file is configured with the parameter Diagnostic File Number. The following table shows the 80 words of the data file for troubleshooting.

Data File Words for Troubleshooting

Word Offset	Description	Category
0	Counter for Commands Received	TCP Server - Link Layer Diagnostics for Modbus TCP Server.
1	Counter for Commands Received with Error	
2	Counter for Replies Sent	
3	Counter for Replies Sent with Error	
4	Reserved	
5	Reserved	
6	Error Count in sessions	
7	Error Code in sessions. See Range of Error codes for Modbus TCP Server and TCP Client on page 432.	
8	Incoming Message Connections	
9	Maximum Connections Allowed	
10…19	Reserved	

Data File Words for Troubleshooting

Word Offset	Description	Category
20	Presentation Layer Error Code	TCP Server – Presentation Layer Diagnostics for Modbus Slave.
21	Presentation Layer Error Count	
22	Function Code that caused the last error	
23	Last Transmitted Error Code	
24	Data file number of last error request	
25	Data element number of last error request	
26	FC 1 Counter	
27	FC 2 Counter	
28	FC 3 Counter	
29	FC 4 Counter	
30	FC 5 Counter	
31	FC 6 Counter	
32	FC 8 Counter	
33	FC 15 Counter	
34	FC 16 Counter	
35…39	Reserved	
40	Counter for Commands Sent	TCP Client – Link Layer Diagnostics for Modbus TCP Client.
41	Reserved	
42	Counter for Replies Received	
43	Counter for Replies Received with Error	
44	Counter for Replies Timed Out	
45	Reserved	
46	Error Count in sessions	
47	Error Code in sessions. See Range of Error codes for Modbus TCP Server and TCP Client on page 432.	
48	Outgoing Message Connections	
49	Maximum Connections Allowed	
50…56	Reserved	
57…59	Reserved, Firmware use only	

Data File Words for Troubleshooting

Word Offset	Description	Category
60	Error code 1 counter	TCP Client – Presentation Layer Diagnostics for Modbus Master
61	Last device reporting error code 1	
62	Error code 2 counter	
63	Last device reporting error code 2	
64	Error code 3 counter	
65	Last device reporting error code 3	
66	Error code 4 counter	
67	Error code 5 counter	
68	Error code 6 counter	
69	Error code 7 counter	
70	Error code 8 counter	
71	Non-Standard Response Counter	
72	Last device reporting error code 4-8 (Non Std Response)	
73…79	Reserved	

Range of Error codes for Modbus TCP Server and TCP Client

Word offsets 7 and 47 reflect the Error Codes in sessions for Modbus TCP Server and TCP Client respectively. The possible Error Codes are listed in the following table. Others that are not defined here are reserved.

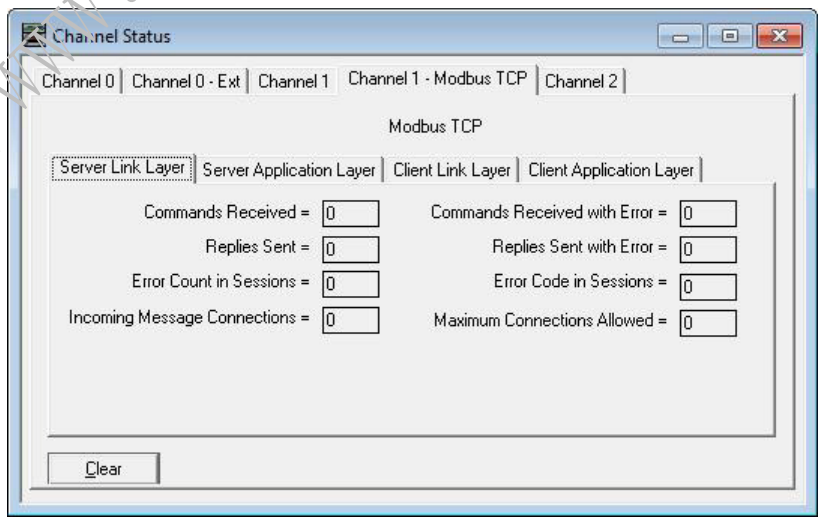
Error Codes for Modbus TCP Server and TCP Client

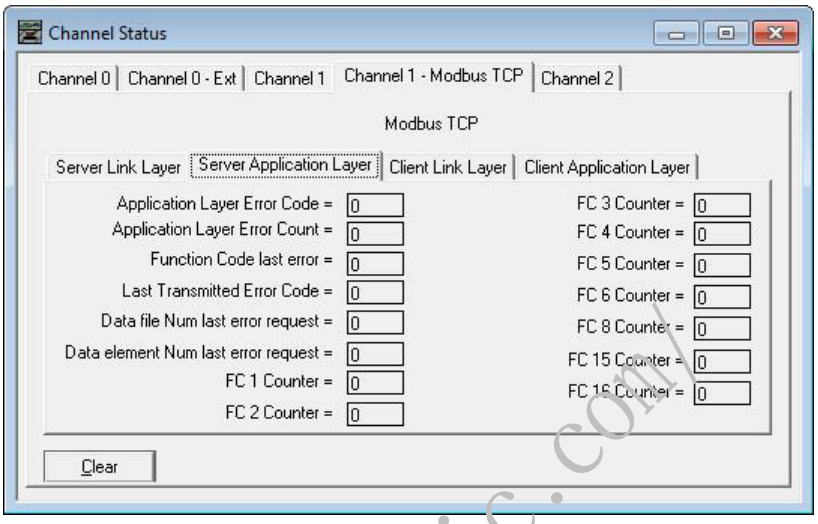
Value (DEC)	Error Code	Description
0	NO_ERROR	No error found
1	ERR_SOCKET_CREATE	Socket error during Create operation
2	ERR_SOCKET_LISTEN	Socket error during Listen operation
3	ERR_SOCKET_BIND	Socket error during Bind operation
4	ERR_SOCKET_ACCEPT	Socket error during Accept operation
5	ERR_SOCKET_CONNECT	Socket error during Connect operation
6	ERR_SOCKET_SEND	Socket error during Send operation
7	ERR_SOCKET_RECEIVE	Socket error during Receive operation
8	ERR_SOCKET_UNLISTEN	Socket error during Unlisten operation
9	ERR_SOCKET_UNBIND	Socket error during Unbind operation
10	ERR_SOCKET_UNACCEPT	Socket error during Unaccept operation
11	ERR_SOCKET_DISCONNECT	Socket error during Disconnect operation
12	ERR_SOCKET_DELETE	Socket error during Delete operation.
13…14	Reserved	–
15	ERR_QUE_FULL	Firmware use only

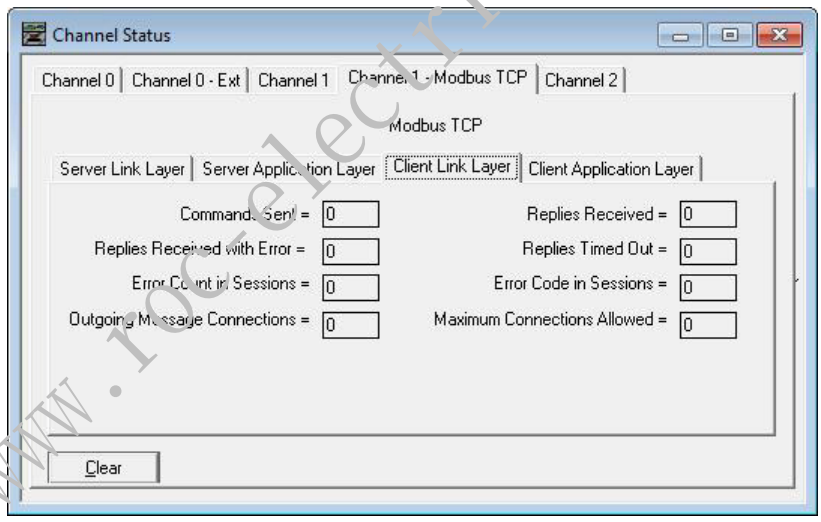
Error Codes for Modbus TCP Server and TCP Client

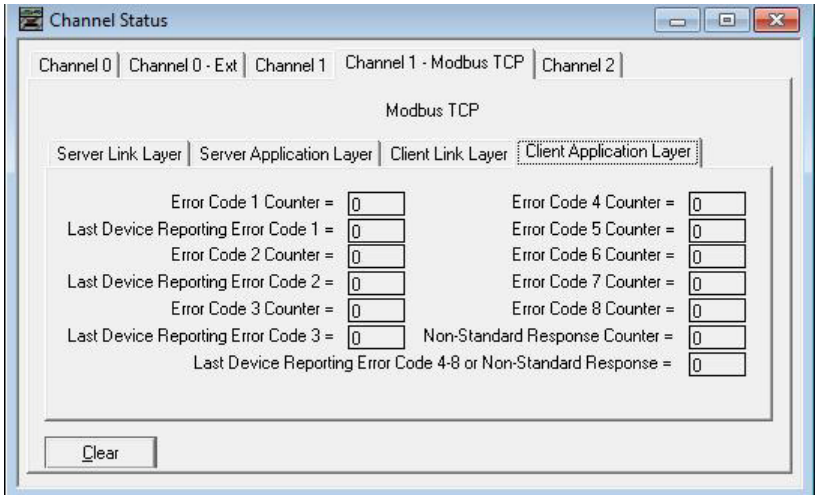
Value (DEC)	Error Code	Description
16	ERR_BUFFER_ALLOC	Firmware use only
17	ERR_PACKET_ALLOC	Firmware use only
18	ERR_PACKET_RELEASE	Firmware use only
19…29	Reserved	–
30	ERR_CONN_REJECTED	Incoming Connection is rejected by the IP address validation
31	ERR_INVALID_HEADER_CRC	Received packet header has invalid CRC
32	ERR_INVALID_HEADER	Received packet header has invalid packet format
33	ERR_INVALID_PACKET_CRC	Received packet has invalid CRC
34	ERR_BAD_PACKET_RECEIVED	Received packet is unknown
35	ERR_PACKET_REJECTED	Received packet is rejected
36	ERR_CONNECTION_BROKEN	The connection has been broken for an unspecified reason
37…49	Reserved	–
50	ERR_INVALID_IP_ADDRESS	Target IP Address is invalid.
51	ERR_INVALID_PORT	Target Port Number is invalid.
52	Reserved	–

The Channel 1 - Modbus TCP tab of the Channel Status dialog allows you to see status and errors for Server Link Layer, Server Application, Client Link Layer and Client Application Layer.









Socket Interface Using CIP Generic Messaging

Overview

The socket interface allows you use a MicroLogix controller to communicate, via the Ethernet Subsystem, with Ethernet devices that do not support the EtherNet/IP application protocol, such as bar code scanners, RFID readers, or other standard Ethernet devices.

Before you use the socket interface, you should be familiar with the following:

- basic TCP/IP, UDP, and socket programming concepts.
- how to write socket programs in a programming language, such as C or Visual Basic.
- how to use diagnostic tools, such as a network sniffer.
- the application protocols of the devices and applications with which the MicroLogix controller will communicate.
- how to write ladder logic for a MicroLogix controller.

Socket Interface Architecture

The socket interface is implemented via the Socket Object in the Ethernet Subsystem. MicroLogix controller programs communicate with the Socket Object via MSG instructions. MSG requests to the Socket Object are similar to socket API calls in most computer operating systems. The Socket Object services let you open connections, accept incoming connections, send data, and receive data.

To communicate with another device, you must understand the other device's application protocol. The Ethernet Subsystem has no application protocol knowledge — it simply makes the socket services available to programs in MicroLogix controllers.

Number and Type of Sockets

You can create as many as 8 socket instances. Each socket instance can be one of these socket types:

- UDP socket (to send/receive UDP datagrams)
- TCP client socket (RSLogix500 initiates the connection)
- TCP server socket (another device initiates the connection to RSLogix500)

You can partition the 8 available socket instances between UDP and TCP sockets by:

- Using all 8 instances for client TCP connections.
- Using all 8 instances to listen for incoming TCP connections and then accept 8 connections from other device.
- Performing both TCP client and server operations.
- Performing both TCP and UDP operations.

When you use the socket instance as a TCP server type, you don't need to make a listen socket. Even if you would like to listen for incoming TCP connections to the same port, you should create a new socket instance. No listen socket instance is supported.

Available Socket Services

Socket Service	Socket Instance
CreateSocket	Server or Client
OpenConnection	Client
AcceptConnection	Server
Read	Server or Client
Write	Server or Client
DeleteSocket	Server or Client
DeleteAllSockets	Server or Client

Once you open a connection on a client socket instance, you cannot use the same socket instance to accept incoming connections. Similarly, if you accept connections on a socket instance, you cannot then use the instance to open outgoing connections. This behavior is consistent with standard socket API behavior.

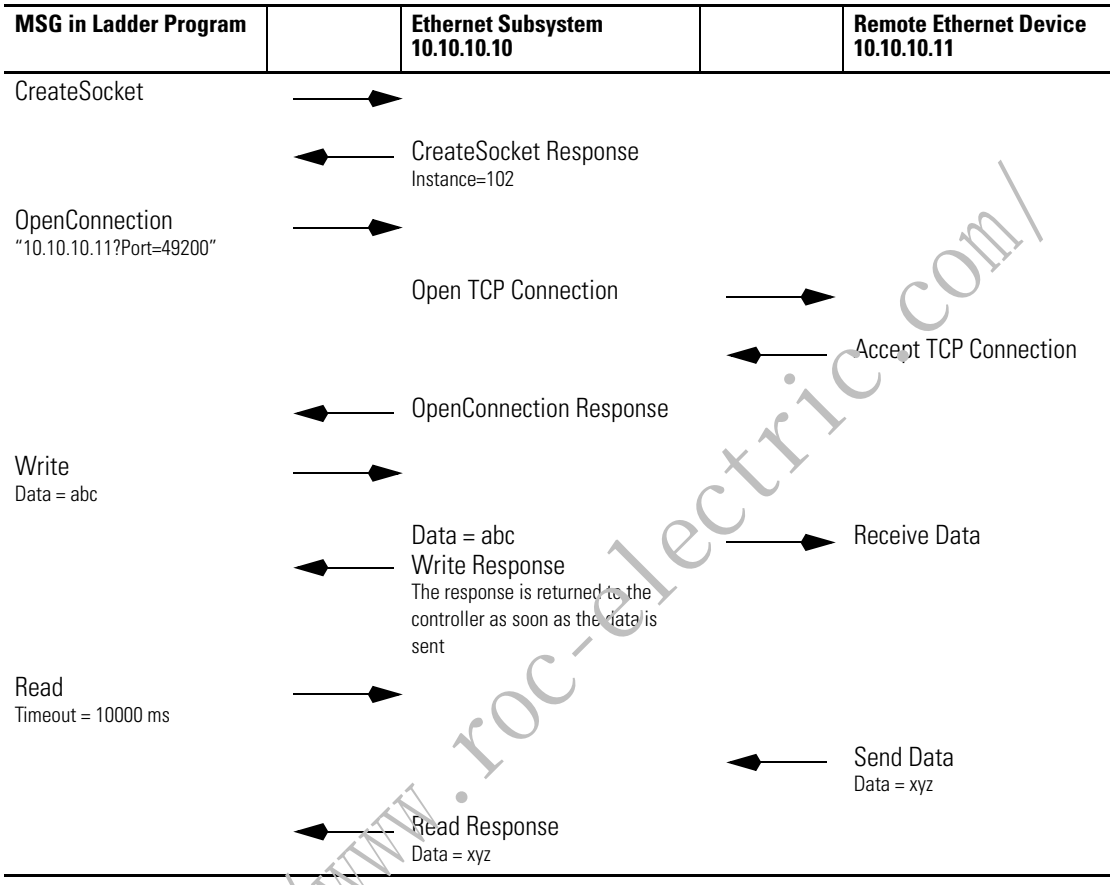
Typical Sequence of Transactions For a TCP Client

The following diagram shows a typical sequence of socket interface transactions with the MicroLogix controller acting as a TCP client.

Here, the MicroLogix controller sends data to a device and then the device sends a response. This is a typical sequence of transactions. Depending on the application protocol, the device could initiate sending data to the MicroLogix controller once the connection is open.

Additionally, each Write does not require an application response or acknowledgement. The application protocol determines the exact sequence of application transactions.

Typical Sequence of Transactions for a TCP Client

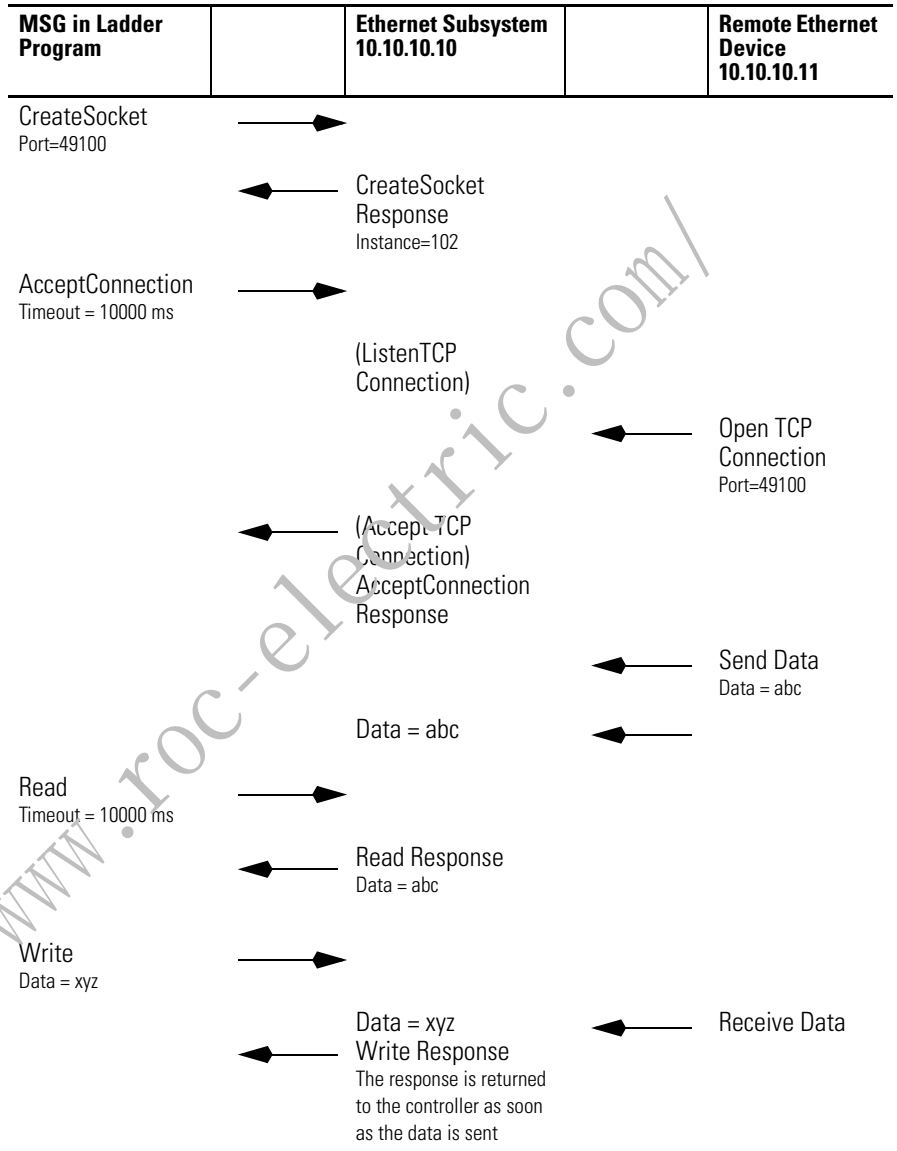


Typical Sequence of Transactions For a TCP Server

The following diagram shows a typical sequence of socket interface transactions with the MicroLogix controller as a TCP server.

The exact sequence of sending and receiving data depends on the application protocol.

Typical Sequence of Transactions for a TCP Server

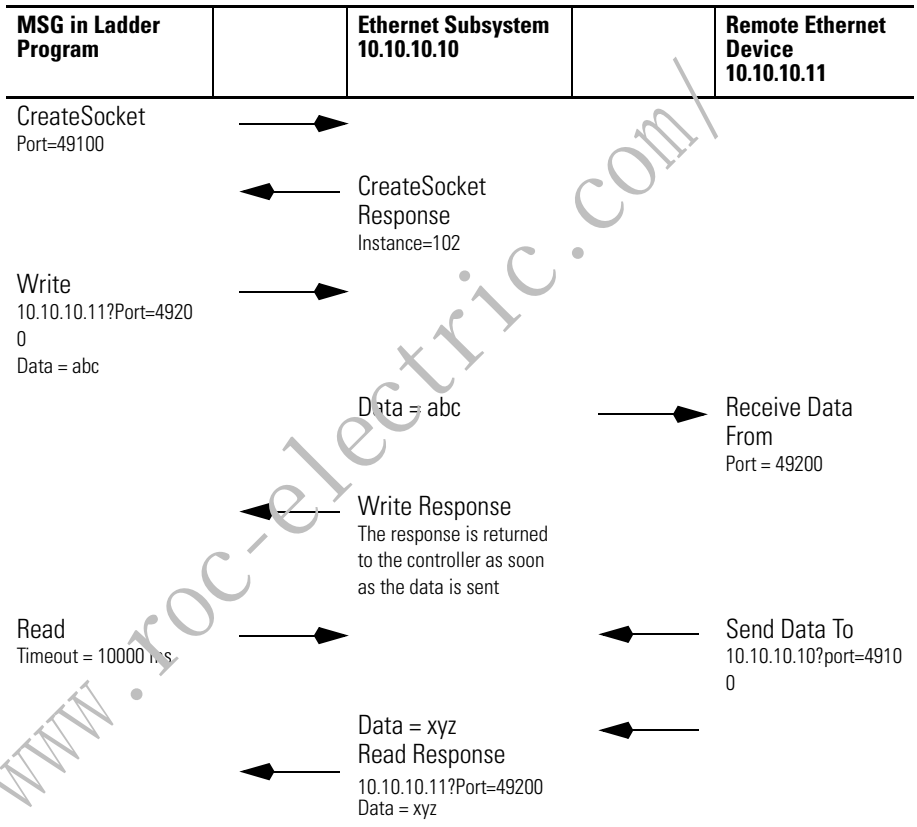


Typical Sequence of Transactions For UDP Without OpenConnection

The following diagram shows a typical sequence of socket interface transactions for UDP communications without using the OpenConnection service to specify the destination address. In this case, the MicroLogix controller specifies the destination for each datagram and receives the sender's address along with each datagram it receives.

The example below shows the MicroLogix controller sending data to a device and then the device sending a response. This is a typical sequence of transactions. Depending on the application protocol, the device could instead initiate sending data to the MicroLogix controller. Additionally, each Write does not require an application response or acknowledgement. The application protocol determines the exact sequence of application transactions.

Typical Sequence of Transactions for UDP Without OpenConnection

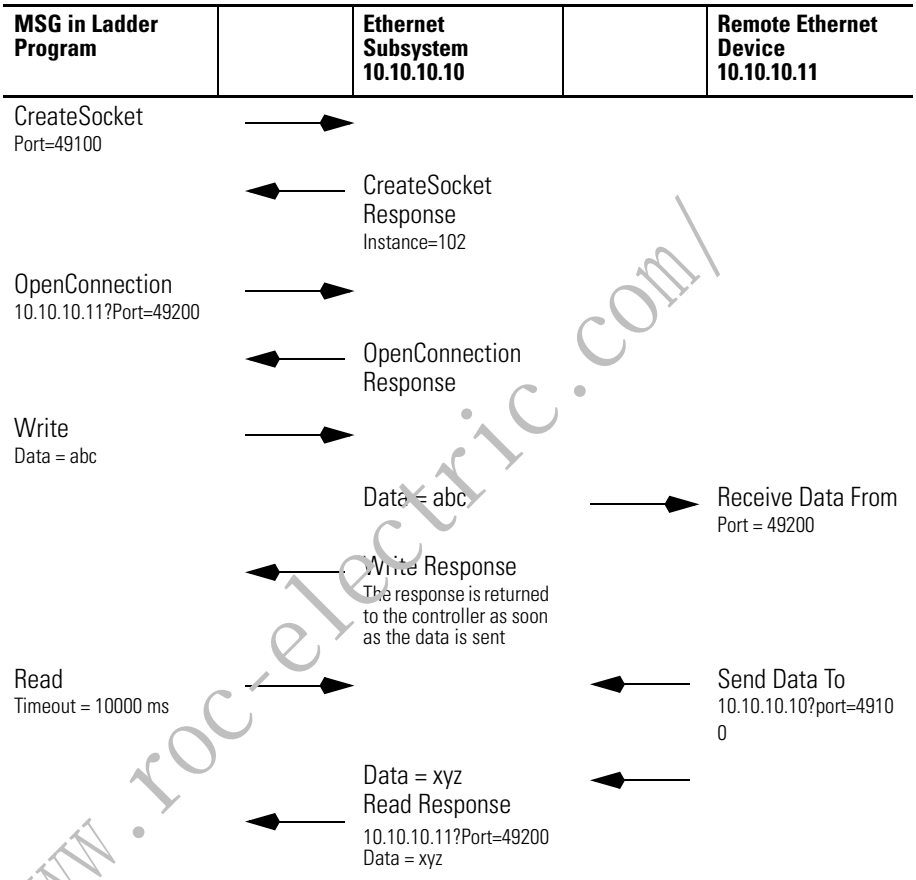


Typical Sequence of Transactions For UDP With OpenConnection

The following diagram shows a typical sequence of socket interface transactions for UDP communications when using the OpenConnection service to specify the destination address.

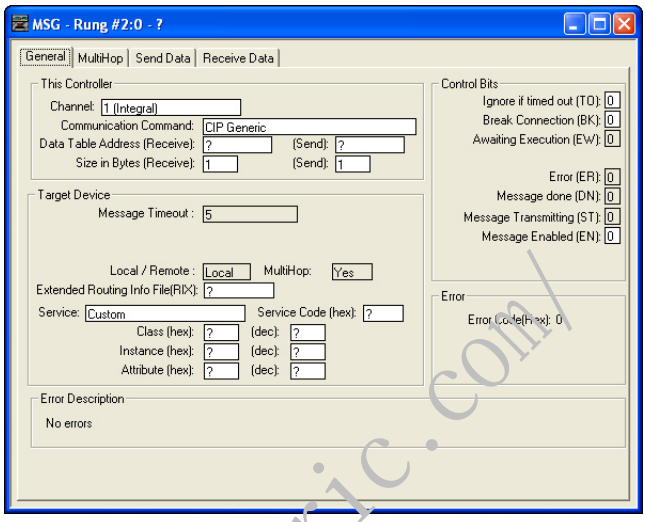
The exact sequence of sending and receiving data depends on the application protocol.

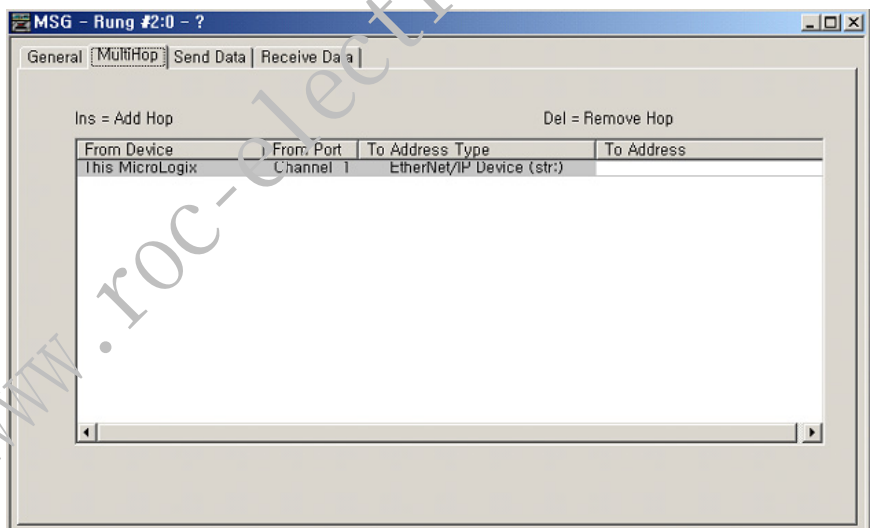
Typical Sequence of Transactions for UDP With OpenConnection



Communicate With the Socket Object Via a MSG Instruction

In MicroLogix controller programs, you can use a CIP Generic MSG instruction to request socket services.





Configure the MSG with these values.

Message Parameters

CIP Generic Msg Parameter	Description
Channel	Select 1 (Integral)
Communication Command	Select CIP Generic
Service	Select Custom

Message Parameters

CIP Generic Msg Parameter	Description
Service Code	Each socket service has a unique service code: •CreateSocket : 4B (hexadecimal) •OpenConnection : 4C (hexadecimal) •AcceptConnection : 50 (hexadecimal) •Read : 4D (hexadecimal) •Write : 4E (hexadecimal) •DeleteSocket : 4F (hexadecimal) •DeleteAllSockets : 51 (hexadecimal).
Class	Specify 342 (hexadecimal) for the Socket Object.
Instance	Specify: •0 for CreateSocket and DeleteAllSockets services •Instance number returned by CreateSocket for other services Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.
Attribute	Specify the attribute value only when getting or setting an attribute, not when using other services.
Data Table Address (Send)	Specify N file. Contains the request parameters for the socket service. Create a user defined type for the request parameters for each service.
Size in Bytes (Send)	Specify the length of the Send Element.
Data Table Address (Receive)	Specify N file. Contains the response data returned by the service. Create a user defined type for the response data for each service.
Size in Bytes (Receive)	Specify the length of the Receive Element.
Extended Routing Info File(RIX)	Specify RIX file.
Control Bits(TO, BK, EN)	TO bit: If this bit is set, the MSG instruction will cause message timeout and set error bit. BK bit: Specify 0. This bit is not used for this Socket Object and is ignored. EN bit: If this bit is set, the MSG instruction will be triggered.
Error Code(Hex)	See Possible Error Codes for Socket Services on page 463.
Error Description	See Possible Error Codes for Socket Services on page 463.

Message Parameters

CIP Generic Msg Parameter	Description
To Address in MultiHop tab	Always specify “127.0.0.1”. The "To Address" must point to the controller's Local IP Address.
Data in Send Data tab	This tab shows the Send Data table in CIP byte order. Some parameters in Socket Interface services require ASCII string format. Use this tab to enter the ASCII string with ASCII Radix. See OpenConnection on page 450.
Data in Receive Data tab	This tab shows the Receive Data table in CIP byte order.

For details on specific services, see Socket Object Services on page 447.

Message Transfer Sizes

The maximum amount of application data you can send or receive depends on whether you configure the MSG instruction as connected or unconnected. The size of the application data does not include the parameters in the Read and Write services.

Message Transfer Sizes

Service	TCP or UDP (Configured on the CreateSocket Service parameter)	Maximum Amount of Application Data (Configured on the Read or Write Service parameter)
Read	TCP Client/Server UDP with OpenConnection Service UDP without OpenConnection Service	240(=252-12) bytes
Write	TCP Client/Server UDP with OpenConnection Service UDP without OpenConnection Service	236(=252-16) bytes

For Read/Write service from/to TCP sockets, if the application data is larger than the maximum size, you can issue multiple Reads or Writes to receive or send the entire application message.

For Read service from UDP sockets, if the application data is larger than the maximum size, you can issue multiple Reads to receive the entire application message.

For Write service to UDP sockets, the size application data cannot exceed the maximums listed for Read and Write services.

Service Timeouts

You must specify a Timeout parameter (in milliseconds) for any service that might not complete immediately (OpenConnection, AcceptConnection, Read, and Write). The timeout tells the Socket Object the maximum amount of time it

should wait when attempting to complete the service. While waiting for the service to complete, the MSG instruction is enabled.

If the requested service does not complete before the Timeout period expires, the Socket Object returns a response to the service request. See the service descriptions for the content of the response.

IMPORTANT	Make the value of the service Timeout parameter shorter than the MSG instruction timeout, otherwise application data could be lost.

Message Instruction Timeouts

The default MSG instruction timeout is 33 seconds; the maximum MSG timeout is approximately 146 seconds. Specify the MSG instruction timeout by setting the appropriate configuration of the Ethernet Channel Configuration:

- Msg Connection Timeout : up to 65500 milliseconds.
- Msg Reply Timeout : up to 65500 milliseconds.

The MSG timeout is determined by adding the Msg Connection Timeout, Msg Reply Timeout and Default addition time (15 seconds).

Socket Interface Timeouts

Each socket instance has an Inactivity Timeout (default of 30 minutes). If a socket instance receives no service requests for the amount of time specified by the Inactivity Timeout, the socket instance is deleted. If you then try to use the socket instance, the MSG instruction may receive the error NO_SOCKET or FORCE_BUFFER_RETURN.

If you put the controller in Program mode before existing socket instances time out, the controller will disconnect all the connections and delete all the socket instances.

IMPORTANT	Make sure the Inactivity Timeout is longer than the longest interval between socket operations. If the Inactivity Timeout is too short, socket instances may time out, resulting in MSG instruction errors.

Programming Considerations

You should observe the following programming considerations.

TCP Connection Loss

Your application program may encounter conditions that result in TCP connection loss. For example, a network cable can be unplugged, or a target device can be turned off.

Your application program should detect the loss of TCP connections and handle those events appropriately. You can detect connection loss when a:

- Read service returns with an error
- Write service returns with an error. See Possible Error Codes for Socket Services on page 463.

Depending on the application, you might want to:

- fault the controller
- try to re-establish the connection (in the case of a client connection), or
- wait for another incoming connection to be established (in the case of a server connection).

If you want to re-establish communications with the other device, you must:

- delete the socket instance for the lost connection
- if the connection is a client connection, create a new socket instance and issue an OpenConnection service to the target device
- if the connection is a server connection, issue an AcceptConnection service to wait for another connection from the remote device.

Change Controller Mode Between Executing and Non-Executing

Executing mode includes Run, Remote Run, Test Continuous Scan, and Test Single Scan modes. Any others are Non-Executing modes.

If the MicroLogix controller transitions from Executing to Non-executing mode while socket requests are active, all connections are closed and all instances are deleted. This is forced by the controller. You can experience the error code NO_SOCKET from MSG instructions. See Possible Error Codes for Socket Services on page 463.

In RSLogix500, you can also set the MGx:y.TO bit for any outstanding socket-related MSG instruction. This causes the MSG instruction to timeout and set the MGx:y.ER bit.

Application Messages and TCP

A TCP connection is a byte stream between two application entities. The application protocol determines the message formats. Messages can be fixed size or variable size.

If an application sends variable size messages, a common strategy is to first send a fixed size 'header' containing the size of the message, followed by the message. The receiving device can first issue a Read of the fixed size header to determine the remaining size, and then issue a subsequent Read to receive the remaining data.

Partial Reads

It is possible for a Read service to return a BufLen that is less than the requested amount of data. For example, your program may request 100 bytes of data. Because TCP is a byte stream and not a datagram protocol, you can receive less than 100 bytes when the Read service returns.

Depending on the application protocol, you can issue additional Read requests to receive all the data. If the application protocol dictates that all messages are 100 bytes, then you must issue additional Reads until you receive 100 bytes. If the application protocol uses variable size messages, your program needs additional logic to handle variable message sizes as defined by the application protocol.

When issuing multiple Read services, be careful to adjust the destination data table that receives the data so that data is not overwritten. This fragment of Structured Text logic shows an example of handling a partial Read service.

```
/* copy the message we just read */
CPW ( ReadResponse.Buf[0], ReadBuf[CurrentLen], ReadResponse.BufLen );
CurrentLen := CurrentLen + ReadResponse.BufLen;

/* do we need to read more data get a complete message? */
if ( CurrentLen < ApplicationMsgLen ) then
/* issue another read */
ReadParams.BufLen := ApplicationMsgLen - CurrentLen;
MSG ( ReadMSG0 );
end_if;
```


IMPORTANT	If you do not issue consecutive Read services, the rest of the application data will be lost. For example, if the application data size is 100 bytes and you issued a Read service with 50 bytes, you should read the rest of data 50 bytes with consecutive Read service. If you issued a Write service without consecutive Read service, the rest of data 50 bytes will be lost.

Partial Writes

Your program may need to handle the situation, although uncommon, where the Write service is unable to send all the specified bytes. Such a situation can occur if the Write service is called multiple times before the target application can receive the data.

If the Write service is not able to send all of the requested data, your program should issue subsequent Writes to send the remaining data. Your program should also adjust the source data table, so that old data is not sent.

Socket Object Services

The Socket Object supports the following services.

Socket Service
CreateSocket
OpenConnection
AcceptConnection
Read
Write
DeleteSocket
DeleteAllSockets

The MicroLogix controller assumes that the outgoing data is in CIP byte order except the application data(data in Buf). The application data is sent out as it is.

The MicroLogix controller assumes that the incoming data is in CIP byte order except the application data. For example, if you issue a Write service with 2 bytes integer, that integer is sent over a TCP connection or in a UDP datagram in CIP byte order. If you issue a Read service and your destination data table (for the response) contains an integer, the MicroLogix controller assumes the incoming data is in CIP byte order.

Depending on the native byte order of the application you are communicating with, you may have to convert the byte order in RSLogix500 and/or in the application.

The following is an example for Write service:

N file for sending contains: 22 bytes

Nx:1	Nx:2	Nx:3	Nx:4, Nx:5	Nx:6	Nx:7	Nx:8 …
Timeout (4 bytes)	ToAddr Family (2 bytes)	ToAddr Port (2 bytes)	ToAddr IPAddress (4 bytes)	BufLen (2 bytes)	Reserved (2 bytes)	Buf (n bytes)
10000 msec	2 : family	100 : port	1.2.3.4	6	0	"ABCEDEF"
27 10 00 00	02 00	64 00	01 02 03 04	06 00	00 00	41 42 43 44 45 46

So, N file should contain the data in CIP byte order as shown below:
27 10 00 00 02 00 64 00 01 02 03 04 06 00 00 00 41 42 43 44 45 46

N file for receiving contains: 4 bytes

Nx:0	Nx:1
BufLen (2 bytes)	Reserved (2 bytes)
6	0
06 00	00 00

So, N file will contain the data in CIP byte order as shown below:
06 00 00 00

CreateSocket

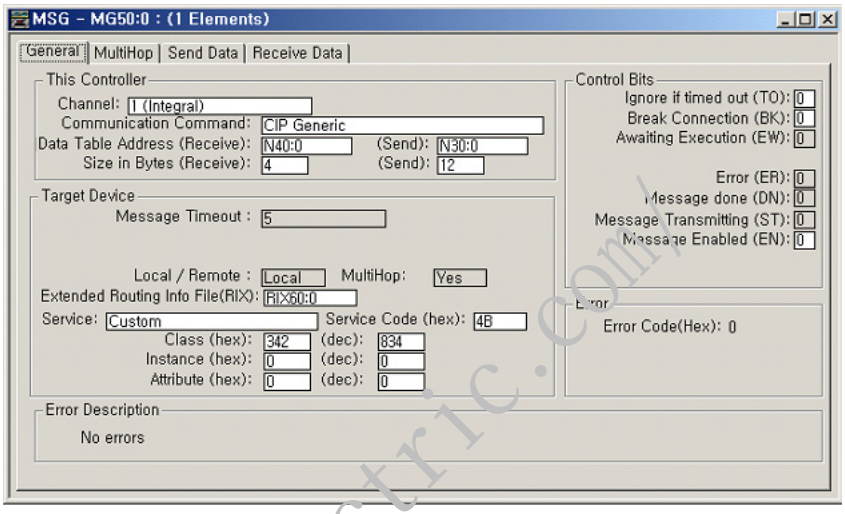
The CreateSocket service creates an instance of the Socket object. The service returns an instance number that you use in the subsequent socket operations.

Call the CreateSocket service with instance 0 (Socket object class).

Parameter	Value
Service Type	CreateSocket
Service Code	4B (hexadecimal)
Class	342 (hexadecimal)
Instance	0
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	4B (hexadecimal)
Class	342 (hexadecimal)
Instance	0
Attribute	0
Data Table Address (Send)	Specify N file
Size in Bytes (Send)	12 bytes
Data Table Address (Receive)	Specify N file
Size in Bytes (Receive)	4 bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1
Considerations : Use the instance returned by CreateSocket on subsequent service requests. Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.	

N file for sending contains: 12 bytes

Nx:0	Nx:1	Nx:2	Nx:3	Nx:4, Nx:5
SocketType (2 bytes)	Reserved (2 bytes)	SocketAddr Family (2 bytes)	SocketAddr Port (2 bytes)	SocketAddr IP Address (4 bytes)

- SocketType : Specify 1 for TCP, 2 for UDP. 0 or all others are reserved.
- Reserved : Specify 0.
- Structure of SocketAddr :
  - Family : Specify the address family. Must be 2.
  - Port : Specify a local port number, or set to 0 (the local port number will be chosen by Ethernet subsystem). For TCP client operations, specify 0 unless you want a specific local port number. For TCP server communications, specify the port number on which to accept incoming connection requests. For UDP, to receive datagrams on a specific port, you must specify a local port number.
  - IPAddr : Specify an IP address. Typically, set to 0 (any address).

N file for receiving contains: 4 bytes

Nx:0	Nx:1
SockInstance (2 bytes)	Reserved (2 bytes)

- SockInstance : Contains Instance number of the socket.

OpenConnection

The OpenConnection service does one of the following:

- For TCP, opens a TCP connection with the specified destination address.
- For UDP, associates a destination IP address and port number with the specified socket.

Open Connection Parameter

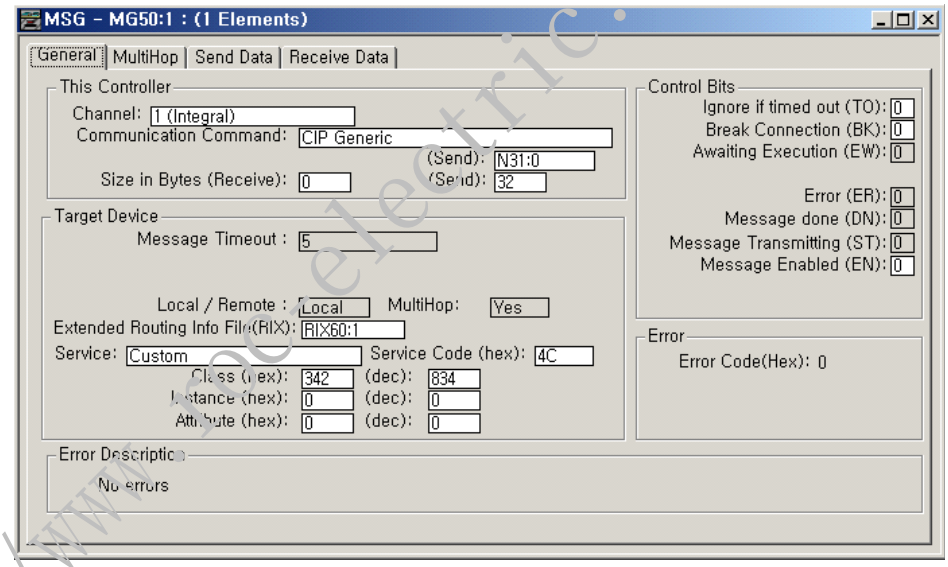
Parameter	Value
Service Type	OpenConnection
Service Code	4C (hexadecimal)

Open Connection Parameter

Parameter	Value
Class	342 (hexadecimal)
Instance	0. Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Message Parameters

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	4C (hexadecimal)
Class	342 (hexadecimal)
Instance	0 Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.
Attribute	0
Data Table Address (Send)	Specify N file
Size in Bytes (Send)	8 + n (number of characters in the destination address) bytes

CIP Generic Message Parameters

CIP Generic Msg Parameter	Description
Data Table Address (Receive)	Not used. The MSG instruction does not return any data.
Size in Bytes (Receive)	0 bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1

Considerations :

In some cases, the OpenConnection service can return before the timeout period without creating a TCP connection. For example, if the destination device is running, but is not listening for connections on the specified port number, OpenConnection returns with an error before the timeout period.

For UDP, if you use OpenConnection, you do not have to specify the IP address and port number each time you send data. If you do not specify an IP address and port number, you can only receive data from the previously specified IP address and port number in the OpenConnection service.

For UDP, if you do not use OpenConnection, you must specify the destination address each time you call the Write service to send data. When you call the Read service, in addition to the data, you receive the address of the sender. You can then use the address of the sender to send a response via the Write service.

N file for sending contains: 8 bytes + n bytes

Nx:0, Nx:1	Nx:2	Nx:3	Nx:4…
Timeout (4 bytes)	DestAddr Length (2 bytes)	DestAddr Reserved (2 bytes)	DestAddr Data (n bytes, max 64 bytes)

- Timeout : Specify the timeout in milliseconds.
- DestAddr Length : The length of the destination address.
- Reserved : Specify 0.
- DestAddr Data : Specify an array of characters (maximum of 64) to define the destination of the connection. Specify either of these:
  - Hostname?port=xxx
  - IPAddr?port=xxx
    For example, to specify an IP address, enter "10.88.81.10?port=2813".
    To specify a host name, enter "REMOTE_MOD?port=2823".

N file for receiving contains : 0 bytes
- NONE

AcceptConnection

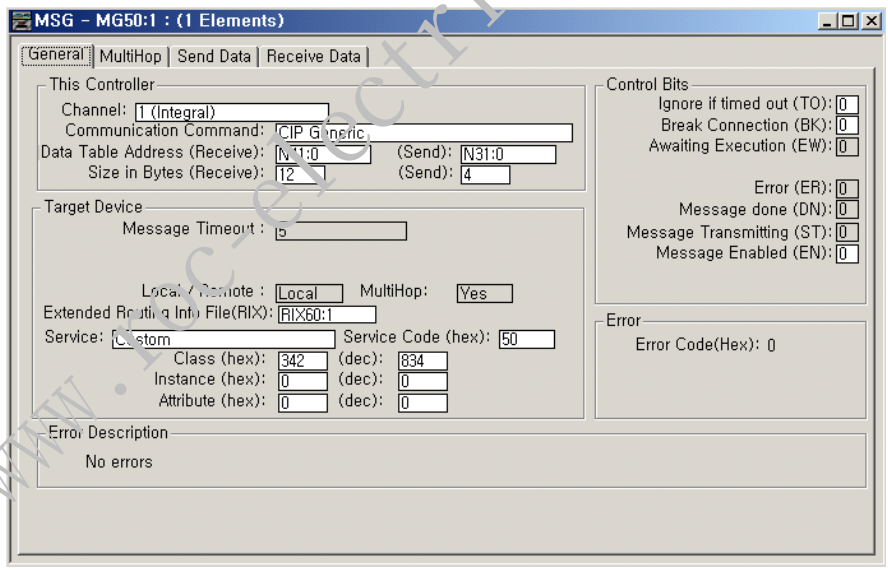
The AcceptConnection service accepts a TCP connection request from a remote destination. Before calling AcceptConnection, call CreateSocket and specify the local port number that will accept the connection. When AcceptConnection completes, it returns a socket instance that you use for sending and receiving data on the newly-created connection.

The AcceptConnection service is not valid for UDP sockets.

Parameter	Value
Service Type	AcceptConnection
Service Code	50 (hexadecimal)
Class	342 (hexadecimal)
Instance	0. Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Message Paraneters

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	50 (hexadecimal)
Class	342 (hexadecimal)
Instance	0 Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.
Attribute	0

CIP Generic Message Paraneters

CIP Generic Msg Parameter	Description
Data Table Address (Send)	Specify N file
Size in Bytes (Send)	4 bytes
Data Table Address (Receive)	Specify N file.
Size in Bytes (Receive)	12 bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1

Considerations :

Create a separate socket instance (CreateSocket) for each port number that will accept connections. After you create a socket instances, call AcceptConnection to wait for an incoming connection request. You can accept connections on the same port number. Each call to AcceptConnection returns a different instance number to use when subsequently reading and writing data.

If you use a local port number that is already in use by Ethernet Subsystem other than Socket Interface subsystem, you may receive an ADDR_IN_USE error (see Possible Error Codes for Socket Services on page 463). Ethernet Subsystem uses these port numbers:

. 80 - HTTP Server
. 161 - SNMP Server
. 2222 - EtherNet Server
. 44818 - EtherNet/IP Server

N file for sending contains: 4 bytes

Nx:0, Nx:1
Timeout (4 bytes)

- Timeout: Specify the timeout in milliseconds.

N file for receiving contains: 12 bytes

Nx:0	Nx:1	Nx:2	Nx:3	Nx:4, Nx:5
SockInstance (2 bytes)	Reserved (2 bytes)	FromAddr Family (2 bytes)	FromAddr Port (2 bytes)	FromAddr IP Address (4 bytes)

- SockInstance : Contains the instance for this service. Use this Instance on subsequent Read and Write services for this connection.
- Reserved : Specify 0.
- Structure of FromAddr :
  - Family : Contains the address family. Must be 2.
  - Port : Contains a remote port number.
  - Addr : Contains an remote IP address.

Read

The Read service reads data on a socket. You can specify a number of bytes to receive. The Read service returns the number of bytes received.

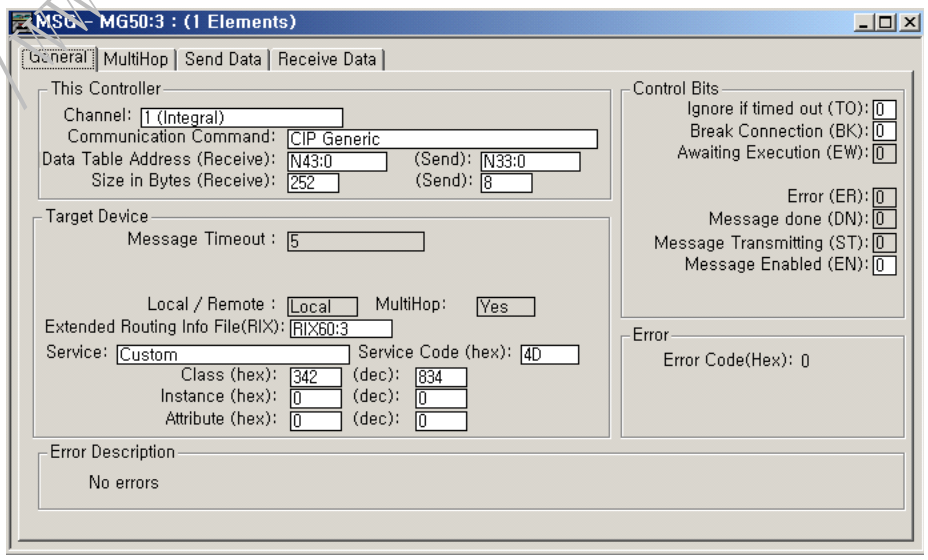
For TCP, the Read service returns when any data is received, up to the requested number of bytes. The Read service can return fewer bytes than were requested. Your application might need to issue multiple Read requests to receive an entire application message.

For UDP, the Read service completes when a datagram is available

Parameter	Value
Service Type	Read
Service Code	4D (hexadecimal)
Class	342 (hexadecimal)
Instance	0. Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	4D (hexadecimal)
Class	342 (hexadecimal)
Instance	0 Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.
Attribute	0
Data Table Address (Send)	Specify N file
Size in Bytes (Send)	8 bytes
Data Table Address (Receive)	Specify N file.
Size in Bytes (Receive)	12 bytes + n (number of bytes of data to receive) bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1
Considerations :	

N file for sending contains: 8 bytes

Nx:0, Nx:1	Nx:2	Nx:3
Timeout (4 bytes)	BufLen (2 bytes)	Reserved Family (2 bytes)

- Timeout : Specify the timeout in milliseconds.
- BufLen : Specify the number of bytes of data to receive. If configured BufLen is less than the actual received data length, the rest of the data can be read from the next read request service. If BufLen is larger than the actual received data length, all the received data will be stored to the Buf. For more detail, see Partial Reads on page 446.
- Reserved : Specify 0.

N file for receiving contains: 12 bytes + n bytes

Nx:0	Nx:1	Nx:2, Nx:3	Nx:4	Nx:5	Nx:6…
FromAddr Family (2 bytes)	FromAddr Port (2 bytes)	FromAddr IP Address (4 bytes)	BufLen (2 bytes)	Reserved (2 bytes)	Buf (n bytes)

- Structure of FromAddr :
  - Family : Contains the address family. Must be 2.

- Port : Contains a local port number.
- Addr : Contains an IP address.

- BufLen : Contains the number of bytes of data received.
- Buf : Contains the data.

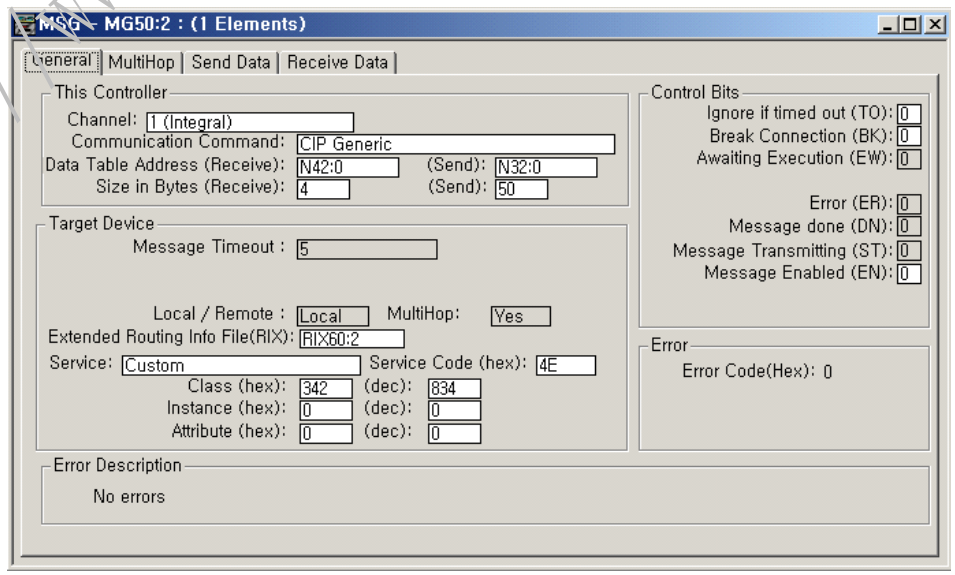
Write

The Write service sends data on a socket. You can specify the number of bytes to send. The Write service attempts to send the requested number of bytes and returns the number of bytes sent.

Parameter	Value
Service Type	Write
Service Code	4E (hexadecimal)
Class	342 (hexadecimal)
Instance	0. Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Message Parameters

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	4E (hexadecimal)
Class	342 (hexadecimal)
Instance	0 Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.
Attribute	0
Data Table Address (Send)	Specify N file. In this N file, Inline Indirection functionality is supported for ASCII communication. For more detail, see Inline indirection functionality on page 459.
Size in Bytes (Send)	16 bytes + n (number of bytes of data to send) bytes
Data Table Address (Receive)	Specify N file.
Size in Bytes (Receive)	4 bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1
Considerations : For TCP connections, ToAddr is ignored. Connectionless UDP messages may use the address stored in the open connection service.	

N file for sending contains: 16 bytes + n bytes

Nx:0, Nx:1	Nx:2	Nx:3	Nx:4, Nx:5	Nx:6	Nx:7	Nx:…
Timeout (4 bytes)	ToAddr Family (2 bytes)	ToAddr Port (2 bytes)	ToAddr IP Address (4 bytes)	BufLen (2 bytes)	Reserved (2 bytes)	Buf (n bytes)

- Timeout : Specify the timeout in milliseconds.
- Structure of ToAddr :
  - Family : Specify the address family. Must be 2.
  - Port : Specify a local port number, or set to 0 (the local port number will be chosen by the Ethernet subsystem). For TCP client operations, specify 0 unless you want a specific local port number. For TCP server communications, specify the port number on which to accept incoming connection requests. For UDP, to receive datagrams on a specific port, you must specify a local port number.
  - Addr : Specify an IP address. Typically, set to 0 (any address).
- BufLen : Specify the number of bytes of data to write.
- Reserved : Specify 0.

- Buf : Specify the data to write.

N file for receiving contains: 4 bytes

Nx:0	Nx:1
BufLen (2 bytes)	Reserved (2 bytes)

- BufLen : the number of bytes that were written.
- Reserved : Contains 0.

Inline Indirection functionality

You can use Inline Indirection functionality for N file in Write service.

If "SEND N7:2 = [N7:2]" is written in the Buf field of the N file, "[N7:2]" is replaced to a string of the value of N7:2. For example, if N7:2 contains a data 39, the string "SEND N7:2 = 39" is sent out. If "SEND L9:3 = [L9:3]" is written in the Buf field of the N file, "[L9:3]" is replaced to a string of the value of L9:3. For example, if L9:3 contains a data 3456789, the string "SEND L9:3 = 3456789" is sent out.

The number of replaced characters may be smaller or larger than the number of Inline Indirection characters. If the number of replaced characters is smaller than the number of Inline Indirection characters, NULL will be filled in at the end of the buffer, as many as the difference between the numbers. If the number of replaced characters is larger than the number of Inline Indirection characters, the transmitted data is trimmed by as many characters as the difference between the numbers.

Inline Indirection can be supported for N file or L file only.

DeleteSocket

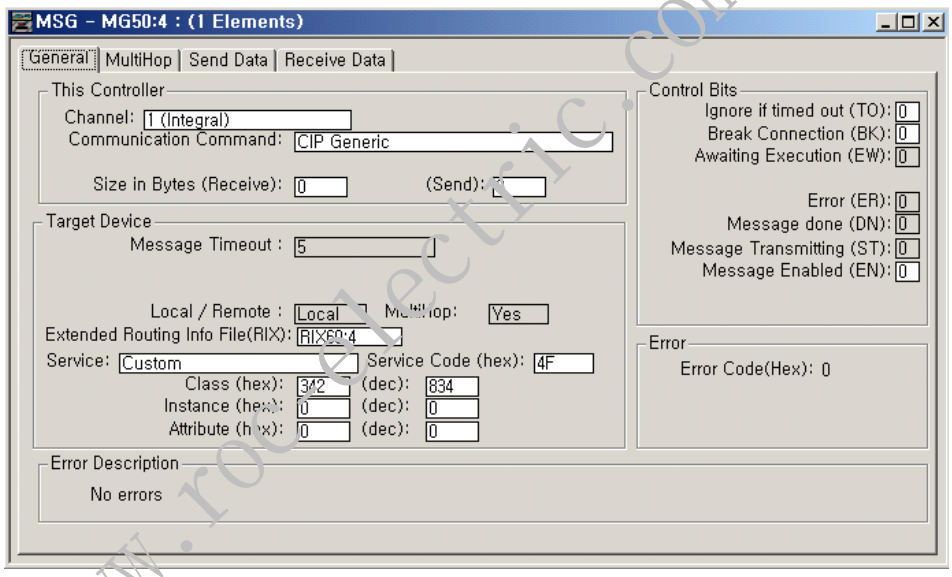
The DeleteSocket service deletes a socket instance. For a TCP connection, the DeleteSocket service also closes the connection prior to deleting the instance.

Parameter	Value
Service Type	Read
Service Code	4F (hexadecimal)

Parameter	Value
Class	342 (hexadecimal)
Instance	0. Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Message Parameters

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	4F (hexadecimal)
Class	342 (hexadecimal)
Instance	0 Use a relay ladder instruction to move the returned instance number from a CreateSocket service into the MGx:y.TFN element of a MSG instruction.
Attribute	0
Data Table Address (Send)	Not used.
Size in Bytes (Send)	0 bytes
Data Table Address (Receive)	Not used.

CIP Generic Message Parameters

CIP Generic Msg Parameter	Description
Size in Bytes (Receive)	0 bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1
Considerations : Delete a socket instance if it is no longer needed. If unused instances are not deleted and you continue to create additional instances, you can exceed the maximum number of instances.	

N file for sending contains : 0 bytes
- NONE

N file for receiving contains : 0 bytes
- NONE

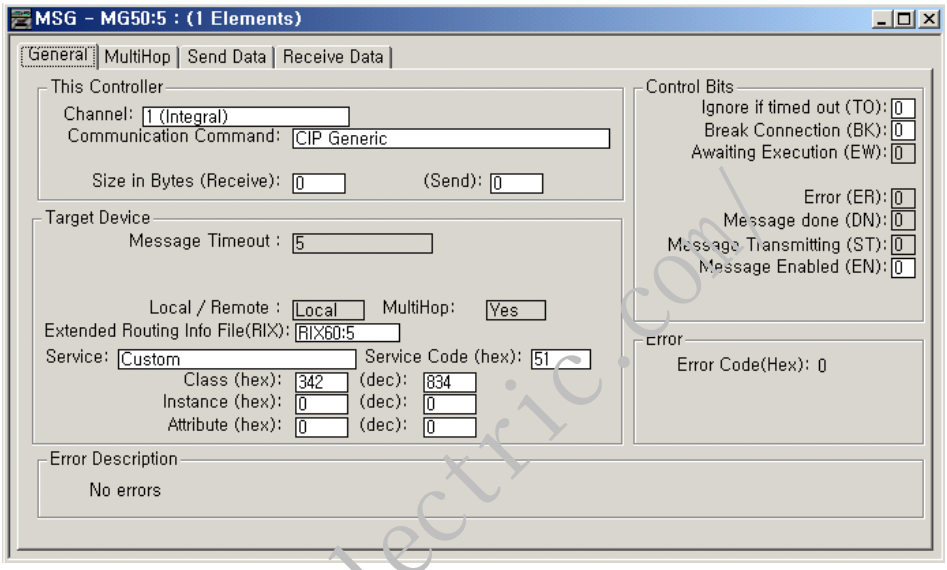
DeleteAllSockets

The DeleteAllSockets service deletes all currently created socket instances. For TCP, the DeleteAllSockets service also closes all connections prior to deleting the instances.

Parameter	Value
Service Type	DeleteAllSockets
Service Code	51 (hexadecimal)
Class	342 (hexadecimal)
Instance	0
Attribute	0

MSG Configuration Parameters

To call this service, click "Setup Screen" in the MSG instruction and enter the values displayed below.



Configure the MSG with these values:

CIP Generic Msg Parameters

CIP Generic Msg Parameter	Description
Channel	1 (Integral)
Communication Command	CIP Generic
Service	Custom
Service Code	51 (hexadecimal)
Class	342 (hexadecimal)
Instance	0
Attribute	0
Data Table Address (Send)	Not used.
Size in Bytes (Send)	0 bytes
Data Table Address (Receive)	Not used.
Size in Bytes (Receive)	0 bytes
Extended Routing Info File(RIX)	Specify RIX file
To Address in MultiHop tab	127.0.0.1
Considerations : Call the DeleteAllSockets service with instance 0. A typical use of DeleteAllSockets is when application development and debugging is complete.	

N file for sending contains : 0 bytes
- NONE

N file for receiving contains : 0 bytes
- NONE

Possible Error Codes for Socket Services

If the Socket Object encounters an error with a service request, or while processing the service request:

- The Socket Object returns an error code.
- The MSG instruction sets the MGx:y/ER bit.
- The MSG instruction sets error codes in Word 18(MGx:y.ERR) of MG file. Low byte is error code 0xDE (Object Specific General Error) and high byte is 0.
- The MSG instruction sets internal fail code in Word 22(MGx:y.22) of MG file. Low byte is error code 0xDE (Object Specific General Error) and high byte of this sub-element contains detailed extended error code.

The values of the extended error code are as the following.

Socket Services Error Codes

Value (Hex)	Error Code	Description
01	NO_SOCKET	No socket available with the same instance.
02	NO_BUFFER_AVAILABLE	No Ethernet buffer available.
03	MAX_CONNECTION	Reached to maximum connections.
04	ILLEGAL_SEQUENCE	Socket Interface Sequence is not valid.
05	CONNECTION_BROKEN	TCP connection has been broken when Read or Write operation.
06	ADDR_IN_USE	The port number is in use already.
07	DNS_ERROR	IP address could not be resolved from DNS server.
08	FORCE_BUFFER_RETURN	MSG buffer has been returned while DeleteSocket or DeleteAllSockets services are executed. Or while the Inactivity timer is expired.
	Reserved	-
10	INVALID_SEND_DATA_SIZE	Invalid send data size in the MSG instruction.
11	INVALID_SERVICE_CODE	Invalid service code in the parameter of the service.
12	INVALID_SOCKET_TYPE	Invalid socket type in the parameter of the service.
13	INVALID_SERVER_TYPE	Invalid server type in the parameter of the service.
14	INVALID_TYPE_CODE	Invalid type code in the parameter of the service.
15	INVALID_FAMILY	Invalid family in the parameter of the service.
16	INVALID_PORT	Invalid port in the parameter of the service.
17	INVALID_ADDRESS	Invalid IP address in the parameter of the service.
18	INVALID_ADDRESS_LENGTH	Invalid address structure length in the parameter of the service.
19	INVALID_DATA_LENGTH	Invalid data length in the parameter of the service.

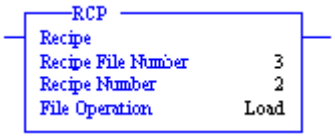
Socket Services Error Codes

Value (Hex)	Error Code	Description
1A	INVALID_TIMEOUT	Invalid timeout value in the parameter of the service. For more detail, see Service Timeouts on page 443.
	Reserved	-
20	SOCKET_ERROR_CREATE	Socket error during Create operation.
21	SOCKET_ERROR_LISTEN	Socket error during Listen operation.
22	SOCKET_ERROR_BIND	Socket error during Bind operation.
23	SOCKET_ERROR_ACCEPT	Socket error during Accept operation.
24	SOCKET_ERROR_CONNECT	Socket error during Connect operation.
25	SOCKET_ERROR_SEND	Socket error during Send operation.
26	SOCKET_ERROR_RECEIVE	Socket error during Receive operation.
27	SOCKET_ERROR_UNLISTEN	Socket error during Unlisten operation.
28	SOCKET_ERROR_UNBIND	Socket error during Unbind operation.
29	SOCKET_ERROR_UNACCEPT	Socket error during Unaccept operation.
2A	SOCKET_ERROR_DISCONNECT	Socket error during Disconnect operation.
2B	SOCKET_ERROR_DELETE	Socket error during Delete operation.
	Reserved	-
30	OPEN_CONN_TIMEOUT	Open connection operation timer was expired.
31	ACCEPT_CONN_TIMEOUT	Accept connection operation timer was expired.
32	READ_TIMEOUT	Read operation timer was expired.
33	WRITE_TIMEOUT	Write operation timer was expired.
	Reserved	-

Recipe and Data Logging

This chapter describes how to use the Recipe and Data Logging functions.

RCP - Recipe



Instruction Type: output

Execution Time for the RCP Instruction

Controller	Operation	When Rung is:	
		True	False
MicroLogix 1400	Load	14.5910 µs	0.5205 µs
	Store	14.8690 µs	0.4515 µs

The RCP file allows you to save custom lists of data associated with a recipe. Using these files along with the RCP instruction lets you transfer a data set between the recipe database and a set of user-specified locations in the controller file system.

The recipe data is stored in Data Log Queue memory.

This section contains the following topics:

- Recipe File and Programming Example on page 467
- Example Queue 0 on page 472
- Example Queue 5 on page 473
- Example Maximum Record of String Data on page 475
- Retrieval Tools on page 481
- Information for Creating Your Own Application on page 481

The following reasons may help you choose which type of memory to use:

- All the recipe data is stored into the controller's memory module. Because the recipe data is stored in Data Log Queue memory, it does not consume user program space.
- If you are not using the data logging function, it allows you more memory (up to 64K bytes) for RCP files. You can use the Data Log Queue for data logging and recipe data, but the total cannot exceed 128K bytes.

See step 2, "Create a RCP File" on page 467 for the recipe file procedure.

The RCP instruction uses the following parameters:

- Recipe File Number - this is the file number that identifies the custom list of addresses associated with a recipe.
- Recipe Number - specifies the number of the recipe to use. If the recipe number is invalid, a user fault (code 0042) is generated.
- File Operation - identifies whether the operation is a Load from the database or a Store to the database.

When executed on a True rung, the RCP instruction transfers data between the recipe database and the specified data locations.

Addressing Modes and File Types are shown in the following table:

RCP Instruction Valid Addressing Modes and File Types

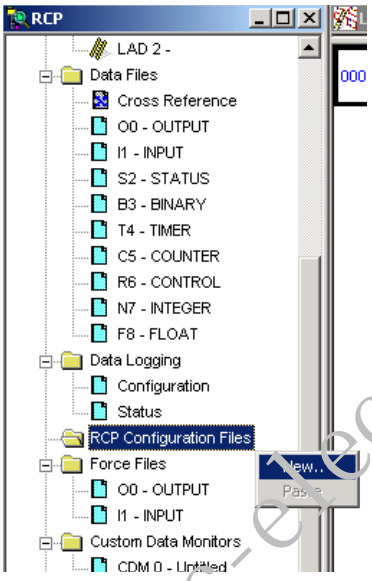
For definitions of the terms used in this table see Using the Instruction Descriptions on page 4-2.

Parameter	Data Files											Function Files								PLS - Programmable L/S	CS - Comms	IOS - I/O	Address Mode			Address Level				
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	RI/RIX	RTC	HSC	PTOX, PWMX	STI	EII	HHI	PMI	DAT	TPI				Immediate	Direct	Indirect	Bit	Word	Long Word	Floating Point	Element
Recipe Number																								•							
File	•	•		•	•	•	•		•				•	•										•	•			•			

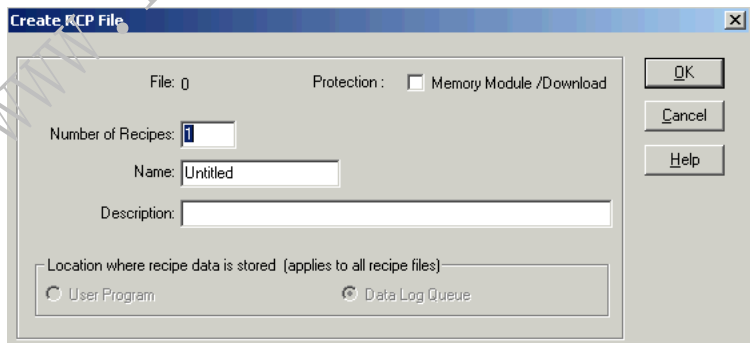
Recipe File and Programming Example

Configuring the RCP file

1. Using RSLogix 500/RSLogix Micro, locate and select *RCP Configuration Files*. Right-click and select *New*.



2. Create a RCP File.

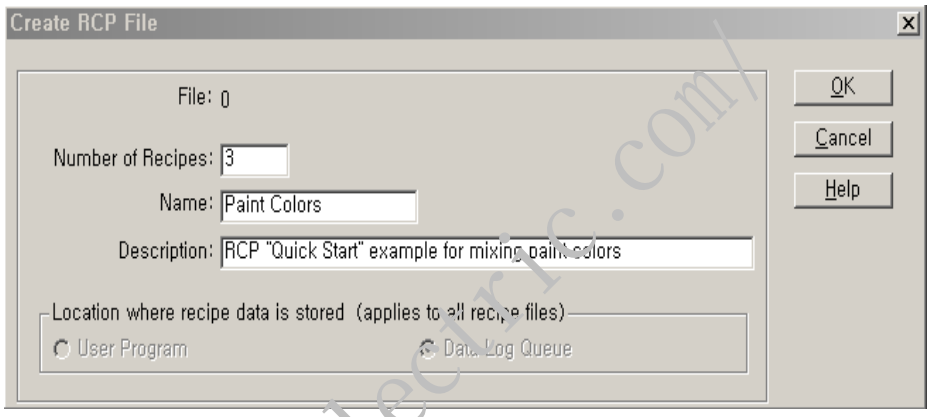


- File - This is the number identifying the RCP file. It is the Recipe File Number used in the RCP instruction in your ladder program and identifies the recipe database.
- Number of Recipes - This is the number of recipes contained in the RCP file. This can never be more than 256. This is the Recipe Number used in the RCP instruction in your ladder program.
- Name - This is a descriptive name for the RCP file. Do not exceed 20 characters.
- Description - This is the file description (optional).
- Location where recipe data is stored (applies to all recipe files) - This allows you to designate a memory location for your RCP files.

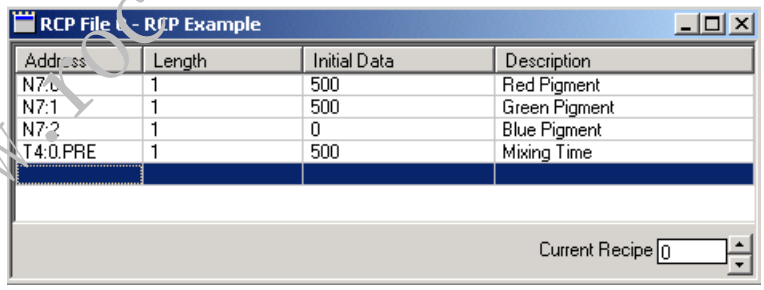
- User Program - Cannot be used.
- Data Log Queue - Recipe data is stored in the data log memory space (max. 64K bytes).

IMPORTANT	All the recipe data is stored into the controller's memory module (1766-MM1).

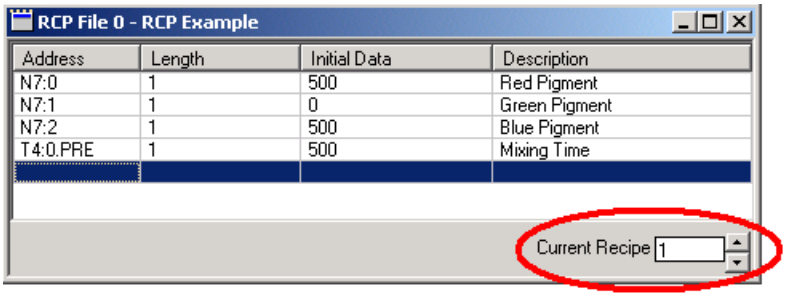
3. Enter the RCP file parameters as shown below. When finished click on OK.



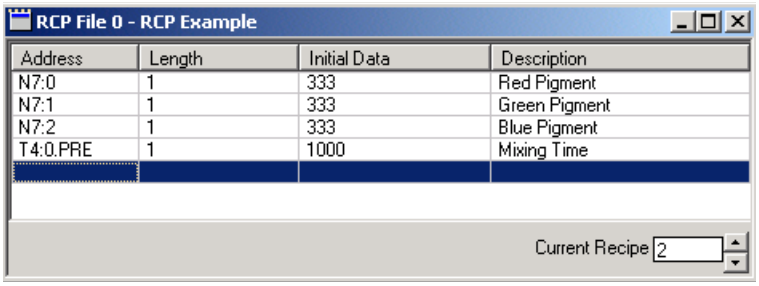
4. A new window will appear. In this window, enter the values as shown below.



5. Change the Current Recipe from 0 to 1. Notice the addresses were duplicated, but the data was not.
6. Enter the data for Recipe 1 as shown below.

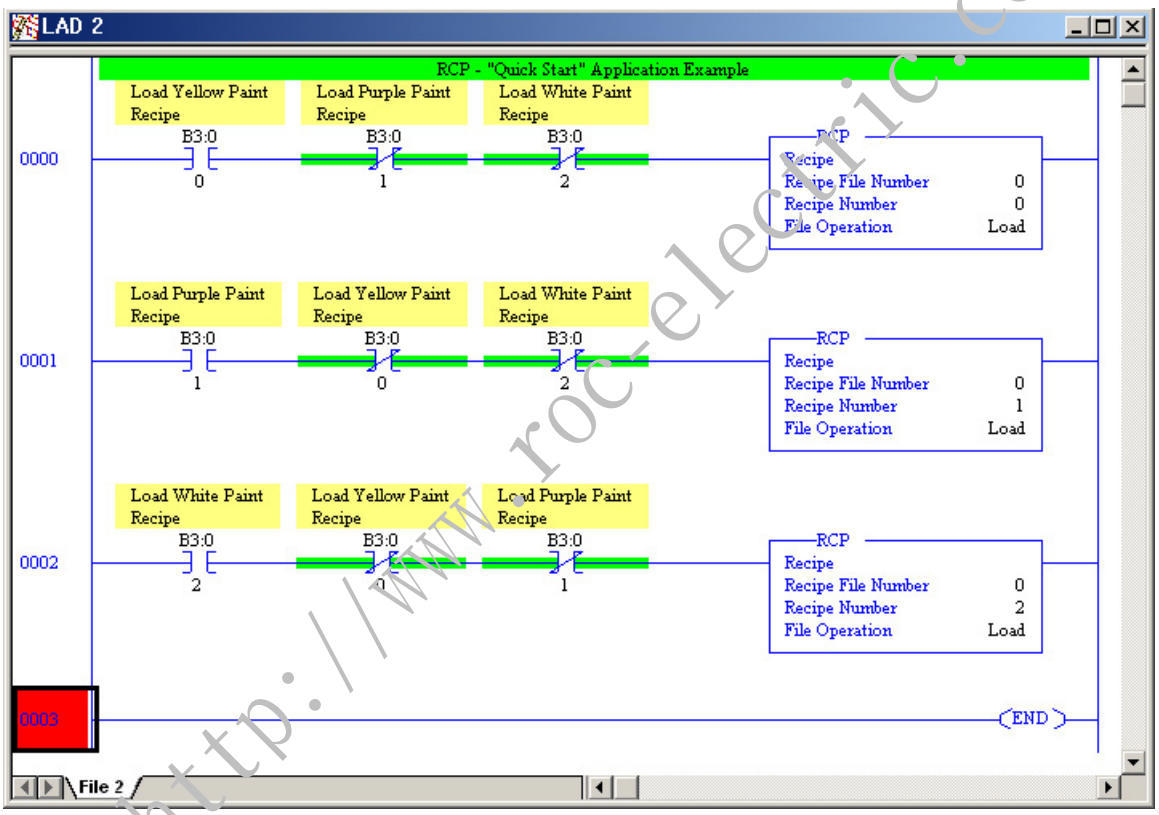


7. Change from Recipe 1 to Recipe 2 and enter the following data.



The Recipes are now configured.

8. Create the following ladder logic.



Application Explanation of Operation

When B3:0/0 is energized and B3:0/1 and B3:0/2 are de-energized, Recipe File 0:Recipe number 0 is executed loading the following values to create Yellow paint.

- N7:0 = 500
- N7:1 = 500
- N7:2 = 0

- T4:0.PRE = 500

When B3:0/1 is energized and B3:0/0 and B3:0/2 are de-energized, Recipe File 0:Recipe number 1 is executed loading the following values to create Purple paint.

- N7:0 = 500
- N7:1 = 0
- N7:2 = 500
- T4:0.PRE = 500

When B3:0/2 is energized and B3:0/0 and B3:0/1 are de-energized, Recipe File 0:Recipe number 2 is executed loading the following values to create White paint.

- N7:0 = 333
- N7:1 = 333
- N7:2 = 333
- T4:0.PRE = 1000

Monitor the N7 data file. Notice the values change after each bit is toggled.

This example describes *loading* values from a RCP file to data table addresses. However, note that by changing the RCP file operation from *Load* to *Store,* values can be loaded by ladder logic into the recipe database for each Recipe number.

Calculation of Consumed Memory

The consumed memory in this example can be calculated by the following equation.

Data Field	Memory Consumption
N7:0	2-byte
N7:1	2-byte
N7:2	2-byte
T4:0.PRE	2-byte
Integrity Check	2-byte
Total	10-byte

Consumed memory size for Recipe File 0

= Data Field per a recipe * Number of Recipes

= 10 * 3 (bytes)

= 30 bytes

Data Logging

Data Logging allows you to capture (store) application data as a record for retrieval at a later time. Each record is stored in a user-configured queue in battery backed memory (B-Ram). Records are retrieved from the MicroLogix 1400 processor via communications. This chapter explains how Data Logging is configured and used.

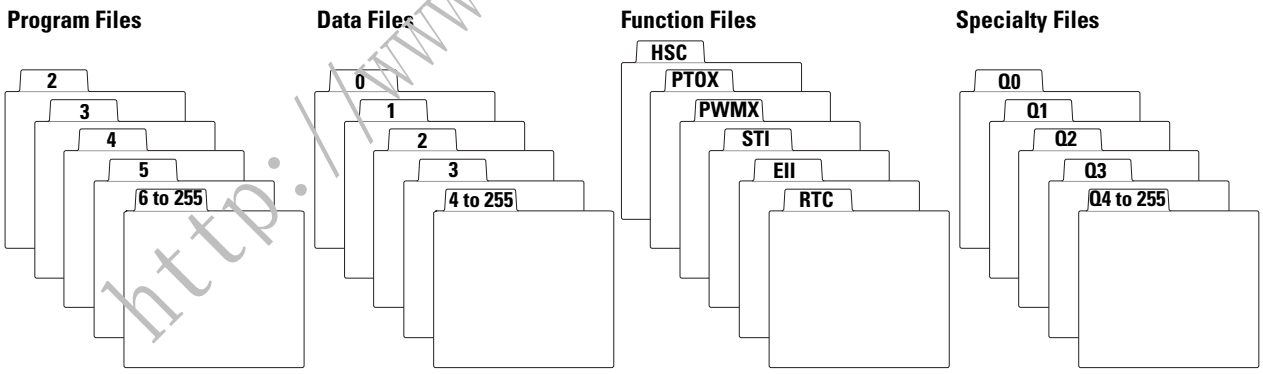
This section contains the following topics:

- Queues and Records on page 471
- Configuring Data Log Queues on page 476
- DLG - Data Log Instruction on page 478
- Data Log Status File on page 479
- Retrieving (Reading) Records on page 481

Queues and Records

The MicroLogix 1400 processor has 128K bytes (128 x 1024) of additional memory for data logging purposes. Within this memory, you can define up to 256 (0...255) data logging queues. Each queue is configurable by size (maximum number of records stored), and by length (each record is 1...109 characters). The length and the maximum number of records determine how much memory is used by the queue. You can choose to have one large queue or multiple small queues.

The memory used for data logging is independent of the rest of the processor memory and cannot be accessed by the User Program. Each record is stored as the instruction is executed and is non-volatile (battery-backed) to prevent loss during power-down.



Example Queue 0

This queue is used to show how to calculate the string length of each record and maximum number of records.

Queue 0 (Date = ✔, Time = ✔, Delimiter = ,)

	Date		Time		N7:11		L14:0		T4:5.ACC		I1:3.0		B3:2
Record 0	01/10/2000	,	20:00:00	,	2315	,	103457	,	200	,	8190	,	4465
Record 1	01/10/2000	,	20:30:00	,	2400	,	103456	,	250	,	8210	,	4375
Record 2	01/10/2000	,	21:00:00	,	2275	,	103455	,	225	,	8150	,	4335
Record 3	01/10/2000	,	21:30:00	,	2380	,	103455	,	223	,	8195	,	4360
Record 4	01/10/2000	,	22:00:00	,	2293	,	103456	,	218	,	8390	,	4375
Record 5	01/10/2000	,	22:30:00	,	2301	,	103455	,	231	,	8400	,	4405
Record 6	01/10/2000	,	23:00:00	,	2308	,	103456	,	215	,	8100	,	4395
Record 7	01/10/2000	,	23:30:00	,	2350	,	103457	,	208	,	8120	,	4415
Record 8	01/11/2000	,	00:00:00	,	2295	,	103457	,	209	,	8145	,	4505
Record 9	01/11/2000	,	00:30:00	,	2395	,	103456	,	211	,	8190	,	4305
Record 10	01/11/2000	,	01:00:00	,	2310	,	103455	,	224	,	8195	,	4455
Record 11	01/11/2000	,	01:30:00	,	2295	,	103456	,	233	,	8190	,	4495

String Length of Record

The size of a record is limited so that the length of the maximum formatted string does not exceed 109 characters. The following table can be used to determine the formatted string length.

Data	Memory Consumed	Formatted String Size
delimiter	0 bytes	1 character
word	2 bytes	6 characters
long word	4 bytes	11 characters
date	2 bytes	10 characters
time	2 bytes	8 characters
string	84 bytes	89 characters
float	4 bytes	13 characters

For queue 0, the formatted string length is 59 characters, as shown below:

Data	Date		Time		N7:11		L14:0		T4:5.ACC		I1:3.0		I1:2.1
Characters	10	1	8	1	6	1	11	1	6	1	6	1	6

= 10 + 1 + 8 + 1 + 6 + 1 + 11 + 1 + 6 + 1 + 6 + 1 + 6
= 59 characters

Number of Records

Using Queue 0 as an example, each record consumes:

Record Field	Memory Consumption
Date	2 bytes
Time	2 bytes
N7:11	2 bytes
L14:0	4 bytes
T4:5.ACC	2 bytes
I1:3.0	2 bytes
B3:2	2 bytes
Integrity Check	2 bytes
Total	**18 bytes**

In this example, each record consumes 18 bytes. So if one queue was configured, the maximum number of records that could be stored would be 7281. The maximum number of records is calculated by:

Maximum Number of Records = Data Log File Size/Record Size
= 128K bytes/18 bytes
= (128)(1024)/18
= 7281 records

Example Queue 5

Queue 5 (Time = ✔, Delimiter = TAB)

	Time		N7:11		I1:3.0		I1:2.1
Record 0	20:00:00	TAB	2315	TAB	8190	TAB	4465
Record 1	20:30:00	TAB	2400	TAB	8210	TAB	4375
Record 2	21:00:00	TAB	2275	TAB	8150	TAB	4335
Record 3	21:30:00	TAB	2380	TAB	8195	TAB	4360
Record 4	22:00:00	TAB	2293	TAB	8390	TAB	4375
Record 5	22:30:00	TAB	2301	TAB	8400	TAB	4405
Record 6	23:00:00	TAB	2308	TAB	8100	TAB	4395

String Length of Record

The size of a record is limited so that the length of the maximum formatted string does not exceed 109 characters. The following table can be used to determine the formatted string length.

String of Length Record

Data	Memory Consumed	Formatted String Size
delimiter	0 bytes	1 character
word	2 bytes	6 characters
long word	4 bytes	11 characters
date	2 bytes	10 characters
time	2 bytes	8 characters
string	84 bytes	89 characters
float	4 bytes	13 characters

For queue 5, the formatted string length is 29 characters, as shown below:

Data	Time		N7:11		I1:3.0		I1:2.1
Characters	8	1	6	1	6	1	6

= 8 + 1 + 6 + 1 + 6 + 1 + 6 = 29 characters

Number of Records

Using Queue 5 as an example, each record consumes:

Record Memory Consumption

Record Field	Memory Consumption
Time	2 bytes
N7:11	2 bytes
I1:3.0	2 bytes
I1:2.1	2 bytes
Integrity Check	2 bytes
Total	**10 bytes**

Each record consumes 10 bytes. So if only one queue was configured, the maximum number of records that could be stored would be 13107. The maximum number of records is calculated by:

Maximum Number of Records = Data Log File Size/Record Size
= 128K bytes/10 bytes
= (128)(1024)/10
= 13107 records

Example Maximum Record of String Data

String Length of Record

The size of a record is limited so that the length of the maximum formatted string does not exceed 109 characters. The following table can be used to determine the formatted string length.

String Length of Record

Data	Memory Consumed	Formatted String Size
delimiter	0 bytes	1 character
date	2 bytes	10 characters
time	2 bytes	8 characters
string element	84 bytes	89 characters

For queue 0, the formatted string length is 29 characters, as shown below:

Data	Date		Time		String
Characters	10	1	8	1	89

= 10 + 1 + 8 + 1 + 89 = 109 characters

Number of Records

Using Queue 0 as an example, each record consumes:

Record Memory Consumption

Record Field	Memory Consumption
Date	2 bytes
Time	2 bytes
ST10:0	84 bytes
Integrity Check	2 bytes
Total	**90 bytes**

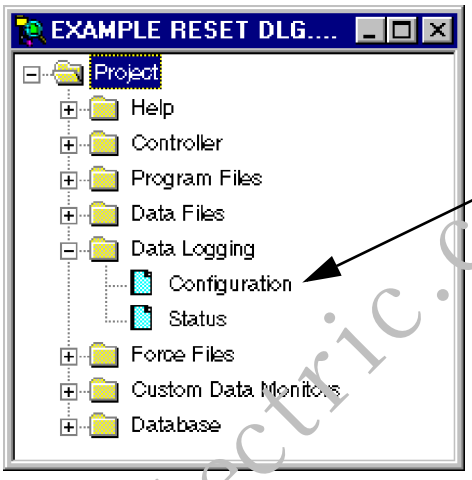
Each record consumes 90 bytes. So if only one queue was configured, the maximum number of records that could be stored would be 1456. The maximum number of records is calculated by:

Maximum Number of Records = Data Log File Size/Record Size
= 128K bytes/90 bytes
= (128)(1024)/90
= 1456 records

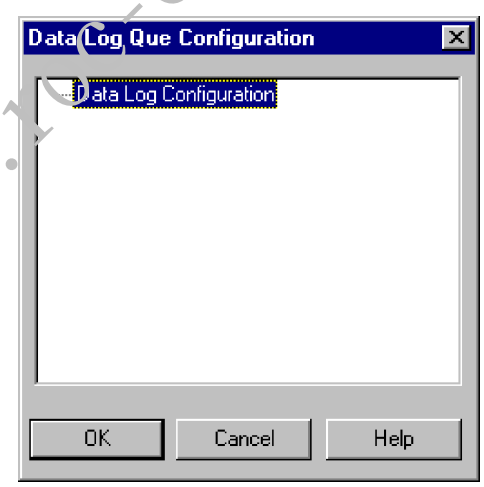
Configuring Data Log Queues

Data Logging is configured using RSLogix 500/RSLogix Micro programming software version V8.10.00 or later.

1. Open a MicroLogix 1400 application. The first step in using Data Logging is to configure the data log queue(s). Access to this function is provided via the RSLogix 500/RSLogix Micro Project tree:

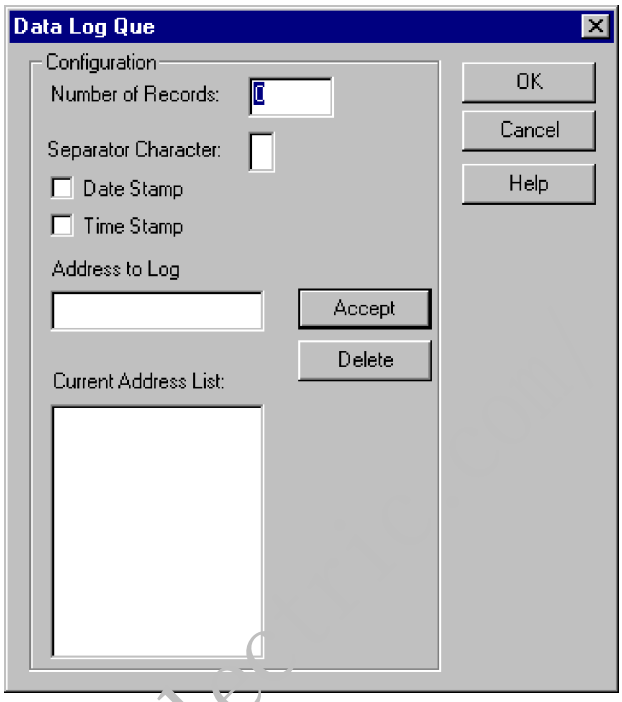


2. The Data Log Que window appears. Double-click on Data Log Configuration.



Appearance of Data Log Que Configuration window before creating a queue.

3. The Data Log Que dialog box appears as shown below. Use this dialog box to enter the queue information.



Enter the following information:

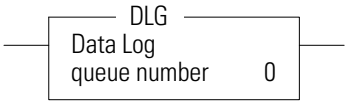
Data Log Queue Configuration Parameters

Data Log Queue Configuration Parameter	Description
Number of Records	Defines the number of records (data sets) in the queue.
Separator Character	Choose the character to act as the separator for the data in this queue (tab, comma, or space). The separator character may be the same or different for each queue configured.
Date Stamp	if selected, the date is recorded in mm/dd/yyyy format(1).
Time Stamp	if selected, the time is recorded in hh:mm:ss format(1).
Address to Log	Enter the address of an item to be recorded and click on *Accept* to add the address to the *Current Address List*. The address can be any 16 or 32-bit piece of data.
Current Address List	This is the list of items to be recorded. Record size can be up to 109 bytes. You can use the *Delete* button to remove items from this list. See page 472 for information on record size.
A record consists of configured Date Stamp, Time Stamp, Current Address List, and Separator Characters.	

(1) If the real-time clock is disabled and Date Stamp and Time Stamp are selected (enabled), the date is recorded as 00/00/0000 and the time as 00:00:00.

4. After entering all the information for the data log queue, click on OK. The queue is added to the Data Log Que window with a corresponding queue number. This is the queue number to use in the DLG instruction.

DLG - Data Log Instruction



Instruction Type: output

Execution Time for the DLG Instruction

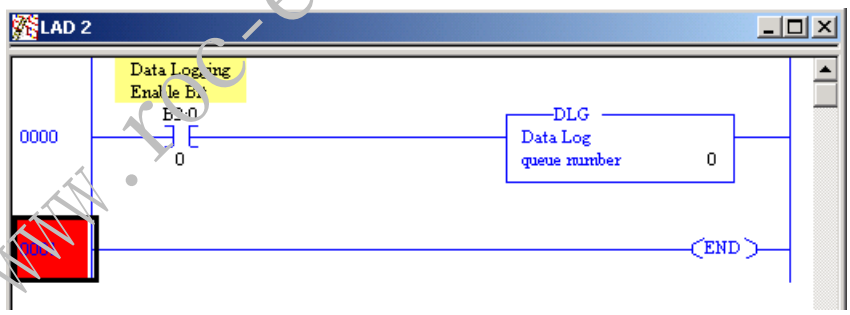
Controller	When Rung Is:	
	True	False
MicroLogix 1400	8.9910 µs	2.6050 µs

IMPORTANT You must configure a data log queue before programming a DLG instruction into your ladder program.

The DLG instruction triggers the saving of a record. The DLG instruction has one operand:

Queue Number - Specifies which data log queue captures a record.

The DLG instruction only captures data on a false-to-true rung transition. The DLG rung must be reset (scanned false) before it will capture data again. Never place the DLG instruction alone on a rung. It should always have preceding logic, as shown below:



Data Log Status File

There is a Data Log Status (DLS) file element for each Data Log Queue. The DLS file does not exist until a data log queue has been configured.

The Data Log Status file has 3-word elements. Word 0 is addressable by bit only through ladder logic. Words 1 and 2 are addressable by word and/or bit through ladder logic.

The number of DLS file elements depends upon the number of queues specified in the application. The status bits and words are described below.

Data Log Status (DLS) File Elements

	Control Element															
Word	**15**	**14**	**13**	**12**	**11**	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**
0	EN(1)	0	DN(2)	OV(3)	CQ(4)	0	0	0	0	0	0	0	0	0	0	0
1	FSZ = File Size (number of records allocated)															
2	RST = Records Stored (number of records recorded)															

(1) EN = Enable Bit

(2) DN = Done Bit

(3) OV = Overflow Bit

(4) CQ = ClearQueue bit

Data Logging Enable (EN)

When the DLG instruction rung is true, the Data Logging Enable (EN) is set (1) and the DLG instruction records the defined data set. To address this bit in ladder logic, use the format: DLS0:Q/EN, where Q is the queue number.

Data Logging Done (DN)

The Data Logging Done (DN) bit is used to indicate when the associated queue is full. This bit is set (1) by the DLG instruction when the queue becomes full. This bit is cleared when a record is retrieved from the queue. To address this bit in ladder logic, use the format: DLS0:Q/DN, were Q is the queue number.

Data Logging Overflow (OV)

The Data Logging Overflow (OV) bit is used to indicate when a record gets overwritten in the associated queue. This bit is set (1) by the DLG instruction when a record is overwritten. Once set, the OV bit remains set until you clear (0) it. To address this bit in ladder logic, use the format: DLS0:Q/OV, where Q is the queue number.

Data Logging ClearQueue (CQ)

The Data Logging ClearQueue(CQ) bit is used to clear the associated queue. This bit is set (1) by the user. This bit is cleared when all of the records in

associated queue are cleared from memory. To address this bit in ladder logic, use the format DLS0:Q/CQ, where Q is the queue number.

File Size (FSZ)

File Size (FSZ) shows the number of records that are allocated for this queue. The number of records is set when the data log queue is configured. FSZ can be used with RST to determine how full the queue is. To address this word in ladder logic, use the format: DLS0:Q.FSZ, where Q is the queue number.

Records Stored (RST)

Records Stored (RST) specifies how many data sets are in the queue. RST is decremented when a record is read from a communications device. To address this word in ladder logic, use the format: DLS0:Q.RST, where Q is the queue number.

TIP If a queue is full and another record is saved, the oldest record is over-written. Queue behavior is the same as a FIFO stack—first in, first out. If a queue is full and an additional record is saved, the "first" record is deleted.

DLS information can be used in the following types of instructions:

Instruction Types

Instruction Type	Operand
Relay (Bit)	Destination Output Bit
Compare	Source A
	Source B
	Low Limit (LIM instruction)
	Test (LIM instruction)
	High Limit (LIM instruction)
	Source (MEQ instruction)
	Mask (MEQ instruction)
	Compare (MEQ instruction)
Math	Source A
	Source B
	Input (SCP instruction)
Logical	Source A
	Source B
Move	Source

Retrieving (Reading) Records

Data is retrieved from a data logging queue by sending a logical read command that addresses the Data Log retrieval file. The oldest record is retrieved first and then, deleted. The record is deleted as soon as it is queued for transmission. If there is a power failure before the transmission is complete, the record is lost.

The data is retrieved as an ASCII string with the following format:

<date><UDS><time><UDS><1st Data><UDS><2nd Data><UDS>…<UDS><Last Data><NUL>

- where:
  <date> = mm/dd/yyyy - ASCII characters (date is optional)
  <time> = hh:mm:ss - ASCII characters (time is optional)
  <UDS> = User Defined Separator (TAB, COMMA, or SPACE)
  <X Data> = ASCII decimal representation of the value of the data
  <NUL> = record string is null terminated
- If the Real Time Clock module is disabled in the controller, <date> is formatted as 00/00/0000, and <time> is formatted as 00:00:00.
- The Communications Device determines the number of sets of data that have been recorded but not retrieved. See the Data Log Status File on page 479.
- The controller performs a the data integrity check for each record. If the data integrity check is invalid, a failure response is sent to the Communications Device. The data set is deleted as soon as the failure response is queued for transmission.

TIP For easy use with Microsoft Excel, use the TAB character as the separator character.

Accessing the Retrieval File

You can use a dedicated retrieval tool or create your own application.

Retrieval Tools

You can download a free Data Logging Tool for Windows-based PCs from http://www.ab.com/programmablecontrol/plc/micrologix/downloads.html.

Information for Creating Your Own Application

Controller Receives Communications Packet

Command Structure

DST	SRC	CMD 0f	STS	TNS	FNC A2	Byte Size	File No.	File Tpe	Ele. No.	S/Ele. No.

Application Fields

Field	Function	Description
DST	Destination Node	
SRC	Source Node	
CMD	Command Code	
STS	Status Code	Set to zero (0)
TNS	Transaction Number	Always 2 bytes
FNC	Function Code	

Application Fields

Field	Function	Description
Byte Size	Number of bytes to be read	Formatted string length (see equation below)
File Number		Always set to zero (0)
File Type		Must be A5 (hex)
Element Number	Queue number	Determines the queue to be read (0…255)
Sub/Element Number		Always set to zero (0)

Equation

Record Field 1	+ Record Field 2	+ Record Field 3	…	+ Record Field 7	=	Formatted String Length

Record Field Sizes

Data Type	Maximum Size
Word	7 bytes (characters)
Long Word	12 bytes (characters)
Date Field	11 bytes (characters)
Time Field	9 bytes (characters)
Float	14 bytes (characters)
String	90 bytes (characters)

TIP The formatted string length cannot exceed 109 bytes in length.

TIP The last byte will be a zero value representing the terminator character.

Controller Responds with Reply

Reply Structure

SRC	DST	CMD 4f	STS	TNS	DATA	EXT STS

Field	Function	Description
SRC	Source Node	
DST	Destination Node	
CMD	Command Code	
STS	Status Code	
TNS	Transaction Number	Always 2 bytes
DATA		Formatted string

If the data integrity check fails, the record is deleted and an error is sent with STS of 0xF0 and ext STS of 0x0E.

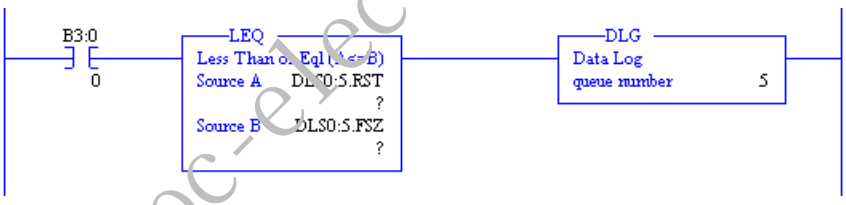
For more information on writing a DF1 protocol, refer to Allen-Bradley publication 1770-6.5.16, *DF1 Protocol and Command Set Reference Manual* (available from http://www.literature.rockwellautomation.com).

Conditions that Will Erase the Data Retrieval File

IMPORTANT The data in the retrieval file can only be read once. Then it is erased from the processor.

The following conditions will cause previously logged data to be lost:

- Program download from RSLogix 500/RSLogix Micro to controller.
- Memory Module transfer to controller except for Memory Module autoload of the same program.
- Full Queue - when a queue is full, new records are recorded over the existing records, starting at the beginning of the file. You can put the following rung in your ladder program to prevent this from happening:



Notes:

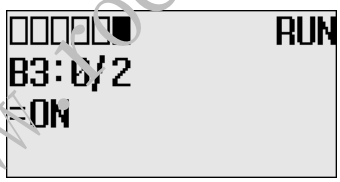
LCD - LCD Information

This chapter describes how to use the LCD functions.

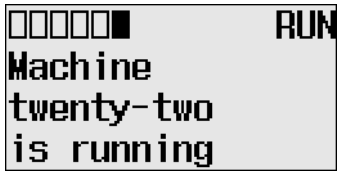
LCD Overview

Through the embedded LCD, your MicroLogix 1400 lets you monitor bit, integer and long integer data within the controller, and optionally modify that data, to interact with your control program. Similarly to the optional 1764-DAT for the MicroLogix 1500 controllers, the embedded MicroLogix 1400 LCD allows users access to 256 bits, 256 integers and 256 long integers, each of which can be individually protected. If you need to know the speed of a conveyor, the status of a remote sensor, or how close your process is running relative to its optimal temperature, you can just monitor your LCD.

You can manually start an operation, change a timing sequence, or make adjustments to a counter and use the LCD to simulate pushbuttons or numeric entry devices. By simply moving or copying data in and out of the bit and integer files, you now can monitor and modify the parameters that your controller uses.



Making use of the new MicroLogix 1400 “LCD Instruction”, your controller can directly interface with a local operator using your ladder logic. The LCD Instruction executes under two modes of operation, the first mode being ladder logic output to the display only (hereafter called “Display Only mode”). In this Display Only mode, up to three lines of data, with up to 16 characters per line, can be sent to the display from the ladder logic running in the controller. Think of this as messaging to the LCD. These lines can consist of combinations of Bits, Integers, Long Integers, Floating and String characters. So now the control program can send alert/alarm messages, I/O data values, simple text messages, or combinations of these messages to the operator. These messages can be triggered by events (input sensors, timer “done bits”, message from another controller, etc.), or based on a scheduled action (using the embedded real time clock, or free running timers).



The second mode of operation again allows for output from the ladder logic to the display, but adds input from the operator back to the controller (hereafter called "Display With Input mode"). Up to two lines of up to 16 characters each can still be sent to the LCD for display, but the third line, in this mode, is used to obtain numeric input from the user. Bit, integer, or long integer file types can be used to provide this input.

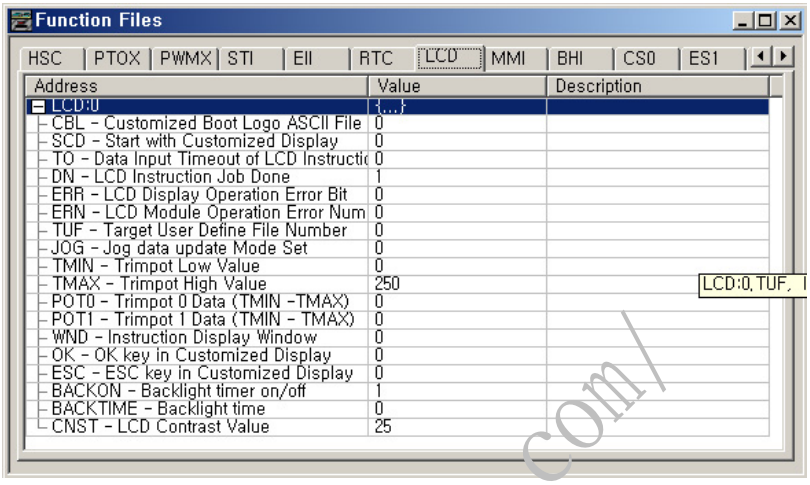
The user can select "User Display" from the LCD menu. The User Display screen will show the specified output data when the LCD Instruction is energized.

If "DISPLAY WITH INPUT" is set to "YES", the user can enter input using the LCD keypad to enter Bit, Integer, or Long Integer data.

LCD Function File

Within the RSLogix 500/RSLogix Micro Function File Folder, you see a LCD Function File. This file provides access to LCD and Trimpot configuration data, and also allows the control program access to all information pertaining to LCD screen, keypad, Trimpot.

TIP If the controller is in the run mode, TUF, TMIN, TMAX may not be changing, those values can be available for changing only when program is downloaded.



The LCD Function File contains status information and control configurations for LCD, Trimpot, and keypad, such as:

- Information about whether to use a customized display at power-up
- Keypad key-in mode and timeout settings
- Bit, Integer and Long Integer data files to monitor
- Current Trimpot values and Trimpot value range settings

LCD Function File Sub-Elements Summary

LCD function file is comprised of 15 sub-elements. These sub-elements are either bit, word structures that are used to provide control over LCD, Keypad, Trimpot. A summary of the sub-element is provided in the following table.

LCD Function File

Feature	Address	Data Format	Type	User Program Access
CBL - Customized Boot Logo ASCII file Address	LCD:0.CBL	word (INT)	control	read/write
SCD - Start with Customized Display	LCD:0/SCD	binary (bit)	control	read-only
TO - Data Input Timeout of LCD instruction	LCD:0.TO	word (INT)	control	read-only
DN - LCD Instruction Job Done	LCD:0/DN	binary (bit)	status	read-only
ERR - LCD Display Operation Error Bit	LCD:0/ERR	binary (bit)	status	read-only
ERN - LCD Module Operation Error Number	LCD:0.ERN	word (INT)	status	read-only
TUF- Target User Defined File Number	LCD:0.TUF	word (INT)	control	read-only
JOG - Jog data update Mode set	LCD:0/JOG	binary (bit)	control	read/write
TMIN - Trimpot low value	LCD:0.TMIN	word (INT)	control	read-only
TMAX - Trimpot high value	LCD:0.TMAX	word (INT)	control	read-only
POT0 - Trimpot 0 Data (TMIN – TMAX)	LCD:0.POT0	word (INT)	status	read-only
POT1 - Trimpot 1 Data (TMIN – TMAX)	LCD:0.POT1	word (INT)	status	read-only
WND - Instruction Display Window	LCD:0/WND	binary (bit)	status	read-only

LCD Function File

Feature	Address	Data Format	Type	User Program Access
OK - OK key in Customized Display	LCD:0/OK	binary (bit)	status/control	read/write
ESC - ESC key in Customized Display	LCD:0/ESC	binary (bit)	status/control	read/write
CNST - LCD contrast	LCD:0.CNST	word (INT)	control	read/write
BACKON - LCD backlight on/off	LCD:0.BACKON	binary (bit)	status/control	read/write
BACKTIME - LCD backlight timer	LCD:0.BACKTIME	binary (bit)	status/control	read/write

LCD Function File Sub-Elements

Customized Boot Logo ASCII File Address Offset (CBL)

Feature	Address	Data Format	Type	User Program Access
CBL - Customized Boot Logo ASCII File Address Offset	LCD:0.CBL	word (INT)	control	read/write

Customized Boot Logo ASCII File Address Offset (CBL) defines which ASCII file number is used for boot image. When the imported BMP file exists in an ASCII data file and a valid ASCII file is set, the controller checks the file type and file size in the BMP header. If there is a proper image in a ASCII file, the controller displays the BMP image in the whole window screen. If an ASCII file does not have enough space to display an image, or a data value is not valid, the controller displays the default logo image.

Start with Customized Display (SCD)

Feature	Address	Data Format	Type	User Program Access
SCD - Start with Customized Display	LCD:0/SCD	binary (bit)	control	read-only

Start with Customized Display (SCD) specifies whether to display a customized LCD screen instead the default I/O Status screen at power-up. When this bit is set (1), the controllers enters the Customized Display mode at power-up, instead entering the default mode and displaying the I/O Status screen, and interfaces with LCD instructions in the ladder program. By setting this bit to ON (1), you can let your controller display LCD instructions or get keypad inputs from the user at power-up, without additional operations.

Data Input Timeout of LCD instruction (TO)

Feature	Address	Data Format	Type	User Program Access
TO - Data Input Timeout of LCD instruction	LCD:0.TO	word (INT)	control	read-only

Data Input Timeout of LCD instruction (TO) specifies timeout period for data input when key inputs are obtained from the user using the LCD instruction in the ladder program. When this word is set to zero (0), it means no timeout is used. When set to a positive value, the LCD exits U-DISP mode and continues to the upper menu if there is no keypad input for more than the specified timeout period (in seconds).

LCD Instruction Job Done (DN)

Feature	Address	Data Format	Type	User Program Access
DN - LCD Instruction Job Done	LCD:0/DN	binary (bit)	status	read-only

LCD Instruction Job Done (DN) is set (1) when an LCD instruction is completed. If the Display With Input bit of the LCD instruction is clear (0, No), DN bit is set (1) immediately after its execution result is displayed on the LCD. If the Display With Input bit is set (0, Yes), DN bit is set (1) when the OK or ESC key is pressed.

LCD Display Operation Error Bit (ERR)

Feature	Address	Data Format	Type	User Program Access
ERR - LCD Display Operation Error Bit	LCD:0/ERR	binary (bit)	status	read-only

LCD Display Operation Error Bit (ERR) bit indicates whether there is a Trimpot range error at each program download. Whenever a program is downloaded, the controller compares the old Trimpot values (POT0 and POT1) with the new Trimpot range (TMIN to TMAX) and sets (1) ERR bit if an error is found and resets (0) if no error is found.

For example, if old POT0 =100, new TMIN=200 and TMAX=50, controller sets (1) ERR bit. This means a Trimpot range error has occurred at program download.

For more information about Trimpot functionality, refer to Using Trim Pots described in the *MicroLogix 1400 Programmable Controllers User Manual*, publication 1766-UM001.

LCD Module Operation Error Number (ERN)

Feature	Address	Data Format	Type	User Program Access
ERN - LCD Module Operation Error Number	LCD:0.ERN	word (INT)	status	read-only

LCD Module Operation Error Number (ERN) shows the error code when an error occurs in LCD configurations and operation.

LCD Error Codes

Error Code	Name	Description
0	None of Error	Normal condition
1	Trimpot Hardware fault	The retentive trim pot(s) value crashed by external failure as like battery fault error.
2	Trimpot Range Over	The Trimpot range of new program is range over.

Target User Defined File Number (TUF)

Feature	Address	Data Format	Type	User Program Access
TUF - Target User Defined File Number	LCD:0.TUF	word (INT)	control	read-only

Target User Defined File Number (TUF) specifies the data file number to monitor on the LCD. If the data type to display is not System Status, Bit, Integer, Long Integer or Floating, the controller displays the "Not S/B/N/L/F" message. For more information, refer to Using Trim Pots described in the *MicroLogix 1400 Programmable Controllers User Manual*, publication 1766-UM001.

Jog data update Mode set (JOG)

Feature	Address	Data Format	Type	User Program Access
JOG - Jog data update Mode set	LCD:0/JOG	binary (bit)	control	read/write

Jog data update Mode set (JOG) determines how the value changes are applied when you press the Up and Down keys to change the data value for a trim pot. When this bit is set (1), the changes are applied immediately whenever you press the Up and Down keys. When it is clear (0), the changes are applied only when you press the OK key after you have changed the value using the Up and Down keys.

There are three ways to change JOG bit:

- Editing the LCD Function File with your RSLogix 500/RSLogix Micro programming tool
- Manipulating this bit using a ladder program
- Using the Advance Set and KeyIn Mode menus on the LCD

Trimpot 0 Data (TMIN – TMAX) (POT0),
Trimpot 1 Data (TMIN – TMAX) (POT1)

Feature	Address	Data Format	Type	User Program Access
POT0 - Trimpot 0 Data (TMIN – TMAX)	LCD:0.POT0	word (INT)	status	read-only
POT1 - Trimpot 1 Data (TMIN – TMAX)	LCD:0.POT1	word (INT)	status	read-only

The Data resident in POT0 represents the position of trim pot 0. The Data resident in POT1 corresponds to the position of trim pot 1. Those valid data range for both is from TMIN to TMAX. POT0 and POT1 value is evaluated on valid value when a new program is downloaded. If the previous Trimpot value is out of the new Trimpot range, Trimpot value is changed to the nearest bound. For example, old POT0 =1000, new TMIN=0 and TMAX=250, controller changes the POT0 from 1000 to 250 after downloading program. You can check the ERR and ERN to see if the POT value is modified or not. The POT0 operation described above is identical to POT1.

Instruction Display Window (WND)

Feature	Address	Data Format	Type	User Program Access
WND - Instruction Display Window	LCD:0/WND	binary (bit)	status	read-only

WND is set when LCD menu is in U-DISP. The controller also notifies this status to LCD screen as U-DISP status.

OK key in Customized Display (OK)

Feature	Address	Data Format	Type	User Program Access
OK - OK key in Customized Display	LCD:0/OK	binary (bit)	status/ control	read/write

OK is set when OK key is pressed. This bit should be cleared so as to get the next key input. Because once OK key is pressed, this bit is set and latched until it is cleared by manually. This OK bit is very useful for LCD instruction. You can use this bit as any input of ladder logic when you program with several LCD instructions.

TIP OK bit is presented for handy interface to LCD instruction and keypad, so it is just updated in U-DISP screen.

ESC key in Customized Display (ESC)

Feature	Address	Data Format	Type	User Program Access
ESC - ESC key in Customized Display	LCD:0/ESC	binary (bit)	status/ control	read/write

ESC is set when ESC key is pressed. This bit should be cleared so as to get the next key input. Because once ESC key is pressed, this bit is set and latched until it is cleared by manually. This ESC bit is very useful for LCD instruction. You can use this bit as any input of ladder logic when you program with several LCD instruction.

TIP ESC bit is presented for handy interface to LCD instruction and keypad, so it is just updated in U-DISP screen.

LCD Backlight On/Off (BACKON)

Feature	Address	Data Format	Type	User Program Access
BACKON - LCD Backlight on/ off	LCD:0/BACKON	binary (bit)	control	read/write

LCD Backlight On/Off (BACKON) specifies whether to turn the backlight on or off. When this bit is set (1), the LCD backlight turns on. Otherwise, the backlight always off.

LCD Backlight Time (BACKTIME)

Feature	Address	Data Format	Type	User Program Access
BACKTIME - LCD Backlight timer	LCD:0/BACKTIME	binary (bit)	control	read/write

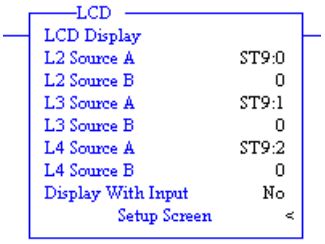
LCD Backlight Time (BACKTIME) specifies the backlight timer period. When this bit is set to zero (0), the default backlight timer period of 30 seconds applies. When set to (1), the backlight is always on.

LCD contrast (CNST)

Feature	Address	Data Format	Type	User Program Access
CNST - LCD Contrast	LCD:0/CNST	word (INT)	control	read/write

LCD contrast (CNST) sets the contrast of the LCD, with a range of 15...35. If the entered data is out of range, the contrast value is changed to the nearest bound value of range. When set to a positive value, the LCD continues to apply the specified contrast value.

LCD - LCD Instruction

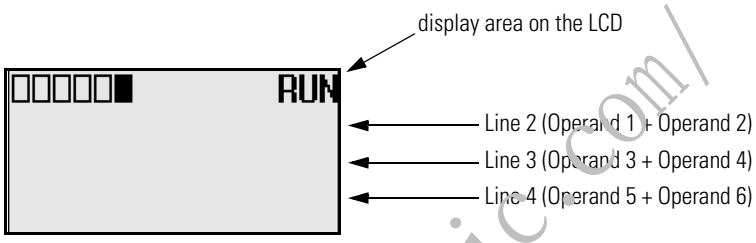


Instruction Type: output

Execution Time for the LCD Instruction

Controller	When Rung Is:	
	True	False
MicroLogix 1400	2.1233 µs	0.2191 µs

The LCD instruction is used to display string or number, get value with keypad.



Addressing Modes and File Types can be used as shown in the following table:

LCD Instruction Valid Addressing Modes and File Types
For definitions of the terms used in this table see Using the Instruction Descriptions on page 70.

Parameter	Data Files											Function Files								CS - Comms	IOS - I/O	DLS - Data Log	Address Mode			Address Level			
	O	I	S	B	T, C, R	N	F	ST	L	MG, PD	PLS	RTC	HSC	PTOX, PWMX	STI	EII	BHI	MMI	LCD				Immediate	Direct	Indirect	Bit	Word	Long Word	Element
Line 2 Source A				•		•	•	•															•	•	•	•	•		•
Line 2 Source B				•		•	•	•	•														•	•	•	•	•		•
Line 3 Source A				•		•	•	•	•														•	•	•	•	•		•
Line 3 Source B				•		•	•	•	•														•	•	•	•	•		•
Line 4 Source A(1)				•		•	•	•	•														•	•	•	•	•	•	•
Line 4 Source B(2)				•		•	•	•	•														•	•	•	•	•	•	•
Display With Input																							•						

(1) L4 Source A "B, N, L" Data File is only available when Display With Input is set to 1.

(2) L4 Source B operand is not available when Display With Input is set to 1.

If Display With Input is set to 0 and the address mode of L2 Source A, L2 Source B, L3 Source A, L3 Source B, L4 Source A, L4 Source B are immediate mode, these value shall be 0.

Default Values:

- L2 Source A, L2 Source B, L3 Source A, L3 Source B, L4 Source A, L4 Source B: 0 (Zero means Address not assigned.)
- Display With Input: 0 (Zero means Display Only mode.)

On a true rung status, the LCD instruction operation depends on how to set the Display With Input operand value. If Display With Input operand is clear (0), LCD instruction mode is set to String or (and) number display. If Display With Input operand is set (1), LCD instruction mode is set to String or (and) number display and value input. User can use two operands per line message to display the two different data. If Bit or Integer or Long Integer or Floating data file is assigned to any operands (except the Display With Input operand), the number converted to signed range string automatically. If the value is less than zero, minus sign ("-") is attached to the head of converted string.

If you need to get long range data (-2,147,483,648 ~ +2,147,483,647) from the keypad, use the L data file.

The largest string size of each line is 16 characters. If there are more than 16 characters in the string file, the remaining characters except the first sixteen are ignored. Special characters such as carriage return and new line are invalid and have no effect on the next line.

Getting Value with Keypad

Your application program can get value from user's keypad inputs if Display With Input bit is set (1) in LCD instructions. User inputs can be obtained with arrow, ESC, and OK keys. In this case, Line 4 is used for user input. L4 Source A is used to specify the target file or element to store user input and the L4 Source B is not used. Where, the data value range for different file types are as follows:

- Integer file (word): -32,768...+32,767
- Bit file: 0 or 1
- Long file (double word): -2,147,483,648...+2,147,483,647

When rung conditions go true, the LCD instruction displays Line 2 and Line 3 strings and positions the cursor at Line 4. Then, the user can input a desired value using the LCD keypad. User input is applied when the OK key is pressed. If the ESC key is pressed, user input is cancelled and no operation is followed. During user's input with the LCD keypad, Line 2 and Line 3 displays are not updated.

Displaying Special Characters

With the LCD instruction, the LCD screen can display the characters of A to Z, a to z, 0 to 9, and special characters in the range of ASCII 0x00...0x7F. If an Input character is in the range of ASCII 0x80...0xFF, a question mark (?) is displayed on the LCD.

Be careful that some special characters are substituted with the corresponding embedded characters in the LCD. The table below shows the available character set on the LCD. For information on how to manipulate a string file for display of special characters, refer to your RSLogix 500/RSLogix Micro Online Help.

LCD Character Set

Notes:

MicroLogix 1400 Memory Usage and Instruction Execution Time

This appendix contains a complete list of the MicroLogix 1400 programming instructions. The list shows the memory usage and instruction execution time for each instruction. Execution times using indirect addressing and a scan time worksheet are also provided.

Programming Instructions Memory usage and Execution Time

The tables below lists the execution times and memory usage for the programming instructions. These values depend on whether you are using *word* or *long word* as the data format.

MicroLogix 1400 Controllers -
Memory Usage and Instruction Execution Time for Programming Instructions

Programming Instruction	Instruction Mnemonic	Word			Long Word		
		Execution Time in µs		Memory Usage in Words	Execution Time in µs		Memory Usage in Words
		True	False		True	False	
ASCII Test Buffer for Line	ABL	21.5621	1.8710	1.14	None	None	None
Absolute Value	ABS	1.4410	0.3750	1.14	1.5390	0.3730	None
ASCII Number of Characters in Buffer	ACB	22.6154	3.5250	1.14	None	None	None
ASCII String to Integer	ACI	6.5719	0.2142	1.14	7.1146	0.1978	None
ASCII Clear Buffer	ACL (both)	26.5540	0.4500	1.57	None	None	None
ASCII Clear Buffer	ACL (receive)	7.8820	0.3848	1.57	None	None	None
ASCII Clear Buffer	ACL (transmit)	5.8590	0.3706	1.57	None	None	None
ASCII String Concatenate	ACN	9.4852	0.1982	1.57	None	None	None
Arc Cosine	ACS	18.0150	0.3750	1.14	18.3070	0.4150	None
Add	ADD	1.8868	0.3540	1.57	1.7807	0.3546	None
ASCII String Extract	AEX	10.0290	0.1850	2.00	None	None	None
ASCII Handshake Lines	AHL	26.5267	2.9480	2.00	None	None	None
ASCII Integer to String	AIC	8.3032	0.2591	1.14	9.8913	0.2155	None
And	AND	1.7894	0.3781	1.57	1.8185	0.3967	None
ASCII Read Characters	ARD	9.3760	7.7770	1.57	None	None	None
ASCII Read Line	ARL	33.9910	8.5690	1.57	None	None	None
ASCII String Search	ASC	8.0844	0.1984	2.00	None	None	None

MicroLogix 1400 Controllers - Memory Usage and Instruction Execution Time for Programming Instructions

Programming Instruction	Instruction Mnemonic	Word			Long Word		
		Execution Time in µs		Memory Usage in Words	Execution Time in µs		Memory Usage in Words
		True	False		True	False	
Arc Sine	**ASN**	42.4610	0.3870	1.14	43.1010	0.3790	None
ASCII String Compare	**ASR**	4.8596	0.2016	1.14	None	None	None
Arc Tangent	**ATN**	146.7510	0.3740	1.14	146.4885	0.4088	None
ASCII Write with Append	**AWA**	10.7810	9.0122	1.57	None	None	None
ASCII Write	**AWT**	13.6110	7.2706	1.57	None	None	None
Bit Shift Left	**BSL**	6.1018	5.8258	1.57	None	None	None
Bit Shift Right	**BSR**	6.0790	5.9942	1.57	None	None	None
Clear	**CLR**	2.0522	0.3714	0.71	2.0125	0.3691	None
File Copy	**COP**	2.5630	0.2034	1.57	None	None	None
Cosine	**COS**	112.7110	0.7686	1.14	19.8070	0.7694	None
Compute	**CPT**	4.8535	0.6610	0.14	None	None	None
Copy Word	**CPW**	2.5630	0.2034	1.57	None	None	None
Count Down	**CTD**	0.4350	0.3803	0.71	None	None	None
Count Up	**CTU**	0.4849	0.3812	0.71	None	None	None
Decode 4-to-1 of 16	**DCD**	4.6300	0.2720	1.14	None	None	None
Radians to Degrees	**DEG**	27.7310	0.4106	1.14	31.2470	0.4098	None
Divide	**DIV**	2.3124	0.3914	1.57	2.3636	0.3914	None
Data Log	**DLG**	8.9910	2.6050	0.71	None	None	None
Encode 1-of-16 to 4	**ENC**	5.7230	0.3660	1.14	None	None	None
End	**END**	1.2016	1.2032	0.14	None	None	None
Equal	**EQU**	1.0814	1.0854	1.29	1.0674	1.0828	None
FIFO Load	**FFL**	8.2970	6.1730	1.57	9.0910	7.4630	None
FIFO Unload	**FFU**	8.7180	6.6490	1.57	9.8890	7.2150	None
Fill File	**FLL**	3.1531	0.5290	1.57	3.2470	0.3918	None
Convert from BCD	**FRD**	5.4790	0.5151	0.71	None	None	None
Gray Code	**GCD**	5.4970	0.5618	1.14	None	None	None
Greater Than or Equal To	**GEQ**	1.0710	0.2228	1.29	1.0601	0.2242	None
Greater Than	**GRT**	1.0682	0.2414	1.29	1.0942	0.2212	None
High-Speed Load	**HSL**	18.8260	0.2910	2.43	18.6510	0.4690	None
Immediate Input with Mask	**IIM**	10.9098	0.2064	1.57	None	None	None
Interrupt Subroutine	**INT**	0.5460	0.5460	0.71	None	None	None
Immediate Output with Mask	**IOM**	10.4010	0.3220	1.57	None	None	None
Jump	**JMP**	0.3290	0.2320	0.71	None	None	None
Jump to Subroutine	**JSR**	0.4615	0.2325	0.71	None	None	None
Label	**LBL**	0.2633	none	0.71	None	None	None

MicroLogix 1400 Controllers -
Memory Usage and Instruction Execution Time for Programming Instructions

Programming Instruction	Instruction Mnemonic	Word			Long Word		
		Execution Time in µs		Memory Usage in Words	Execution Time in µs		Memory Usage in Words
		True	False		True	False	
Lcd Display	**LCD**	2.1233	0.2191	3.29	None	None	None
Less Than or Equal To	**LEQ**	1.0640	0.1847	1.29	1.0364	0.1851	None
Less Than	**LES**	1.0772	0.2106	1.29	1.0935	0.2137	None
LIFO Load	**LFL**	6.4950	6.5650	1.57	7.3570	7.0030	None
LIFO Unload	**LFU**	6.8227	6.5089	1.57	7.6680	7.2102	None
Limit	**LIM**	7.0970	0.2086	1.71	7.3803	0.2009	None
Natural Log	**LN**	127.3260	0.4094	1.14	130.3635	0.4094	None
Base 10 Logarithm	**LOG**	112.7110	0.7686	1.14	19.8070	0.7694	None
Master Control Reset	**MCR (End)**	0.4510	0.5510	0.14	None	None	None
Master Control Reset	**MCR (Start)**	1.0510	1.1510	0.14	None	None	None
Masked Comparison for Equal	**MEQ**	6.2730	0.1934	1.71	7.1602	0.1780	None
Move	**MOV**	1.4231	0.3542	1.14	1.4103	0.3722	None
Message, Steady State	**MSG**	2.5670	0.7310	1.14	None	None	None
Message, False-to-True Transition for Reads	**MSG**	48.1677	0.8510	1.14	None	None	None
Message, False-to-True Transition for Writes	**MSG**	58.8510	0.9177	1.14	None	None	None
Multiply	**MUL**	3.3260	0.3920	1.57	3.3476	0.3918	None
Masked Move	**MVM**	0.2210	0.1750	1.57	1.9050	0.2180	None
Negate	**NEG**	1.3570	0.3548	1.14	1.3660	0.3413	None
Not Equal	**NEQ**	1.5056	0.1880	1.29	1.3892	0.2070	None
Not	**NOT**	1.3682	0.4074	1.14	1.3620	0.3900	None
One Shot	**ONS**	0.2776	0.3110	0.71	None	None	None
Or	**OR**	1.8278	0.3962	1.57	1.8374	0.3956	None
One Shot Falling	**OSF**	1.3672	2.0952	1.14	None	None	None
One Shot Rising	**OSR**	1.3766	1.3724	1.14	None	None	None
Output Enable	**OTE**	0.2685	0.2629	0.57	None	None	None
Output Latch	**OTL**	0.2541	0.1882	0.57	None	None	None
Output Unlatch	**OTU**	0.2830	0.1732	0.57	None	None	None
Proportional Integral Derivative	**PID**	7.1750	7.0910	1.57	None	None	None
Pulse Train Output	**PTO**	11.0210	5.5115	0.71	None	None	None
Pulse Width Modulation	**PWM**	13.2160	7.1710	0.71	None	None	None
Reset Accumulator	**RAC**	8.3310	0.2030	1.14	None	None	None
Degrees to Radians	**RAD**	23.0610	0.4070	1.14	26.211	0.3790	None
Recipe	**RCP**	14.5910	0.5205		14.8690	0.4515	None
I/O Refresh	**REF**	None	0.1490	0.14	None	None	None

MicroLogix 1400 Controllers - Memory Usage and Instruction Execution Time for Programming Instructions

Programming Instruction	Instruction Mnemonic	Word			Long Word		
		Execution Time in µs		Memory Usage in Words	Execution Time in µs		Memory Usage in Words
		True	False		True	False	
Reset	**RES**	0.6320	0.4305	0.71	None	None	None
Return	**RET**	0.3710	0.2510	0.14	None	None	None
Read High Speed Clock	**RHC**	2.5910	0.2150	1.14	3.1210	0.1802	None
Read Program Checksum	**RPC**	4.2844	0.2028	1.14	None	None	None
Real Time Clock Adjust	**RTA**	999.8510	0.4090	0.14	None	None	None
Retentive Timer On	**RTO**	1.1710 (DN=0) 0.6100 (DN=1)	0.5480	0.71	None	None	None
Subroutine	**SBR**	0.2510	0.2510	0.14	None	None	None
Scale	**SCL**	10.9080	0.3608	2.00	None	None	None
Scale with Parameters	**SCP**	83.2977	0.3878	2.86	87.0493	0.2910	None
Sine	**SIN**	92.8635	0.4210	1.14	95.0760	0.4210	None
Sequencer Compare	**SQC**	3.1762	0.8505	2.00	3.2480	0.9823	None
Sequencer Load	**SQL**	2.7700	1.1741	1.57	2.8680	1.2800	None
Sequencer Output	**SQO**	3.6105	0.9480	2.00	3.1920	1.1850	None
Square Root	**SQR**	54.8140	0.3561	1.14	45.1450	0.3732	None
Selectable Timed Interrupt Start	**STS**	20.8470	0.2125	0.71	None	None	None
Subtract	**SUB**	1.8426	0.3767	1.57	1.7651	0.3758	None
Suspend	**SUS**	None	None	None	None	None	None
Service Communications	**SVC**	39.8260 (CH0) 5.9042 (CH1) 36.5800 (CH0 & CH1)	0.1933 (CH0) 0.1857 (CH1) 0.1774 (CH0 & CH1)	1.86	None	None	None
Swap	**SWP**	1.0728	0.1963	1.14	None	None	None
Tangent	**TAN**	122.6760	0.3915	1.14	126.9135	0.4234	None
Compute Time Difference	**TDF**	5.9770	0.2219	1.57	7.2150	0.2035	None
Temporary End	**TND**	0.3320	0.2100	0.14	None	None	None
Convert to BCD	**TOD**	5.9198	0.3916	0.71	None	None	None
Off-Delay Timer	**TOF**	0.5203	1.0962 (DN=0) 0.5322 (DN=1)	0.71	None	None	None
On-Delay Timer	**TON**	2.0338 (DN=0) 1.2608 (DN=1)	0.8608 (DN=0)	0.71	None	None	None
User Interrupt Disable	**UID**	2.7470	0.1859	0.71	None	None	None
User Interrupt Enable	**UIE**	3.4226	0.1968	0.71	None	None	None
User Interrupt Flush	**UIF**	2.7930	0.1847	0.71	None	None	None
Examine if Closed	**XIC**	0.2646	0.2512	0.71	None	None	None
Examine if Open	**XIO**	0.2513	0.2775	0.71	None	None	None
Exclusive Or	**XOR**	4.9480	0.3671	1.57	4.8454	0.3646	None
X Power Y	**XPY**	66.2050	0.3920	1.57	69.0550	0.3548	None

MicroLogix 1400 Indirect Addressing

The following sections describe how indirect addressing affects the execution time of instructions in the MicroLogix 1400 processor. The timing for an indirect address is affected by the form of the indirect address.

For the address forms in the following table, you can interchange the following file types:

- Input (I) and Output (O)
- Bit (B), Integer (N)
- Timer (T), Counter (C), and Control (R)
- String(ST)

MicroLogix 1400 Scan Time Calculation

The following is an example of how to calculate a typical scan time for a ladder program. In this example, a program scan time of 15.0 ms is assumed. The program scan time includes the system overhead time.

Communication Channels Inactive

Program scan time: 15 ms

System overhead: 3.35 ms (typically 20% of program scan time)

Ladder logic execution time:
program scan time - system overhead = 15 - 3.35 = 11.65 ms

One Or More Communication Channels Active

When any of the channels are active, the following typical overheads should be taken into account:

- Channel 0 overhead: 0.8 ms
- Channel 1 overhead: 0.4 ms
- Channel 2 overhead: 0.7 ms

If Channel 1 is active and the other two channels are inactive, total program scan time is:

> Program scan time (when no communication channels active) + Channel 1 overhead = 15 ms + 0.4 ms = 15.4 ms

If all Channels are active, total program scan time is:

> Program scan time (when no communication channels active) + Channel 0 overhead + Channel 1 overhead + Channel 2 overhead = 15 ms + 0.8 ms + 0.4 ms + 0.7 ms= 16.9 ms

Notes:

System Status File

The status file lets you monitor how your controller works and lets you direct how you want it to work. This is done by using the status file to set up control bits and monitor both hardware and programming device faults and other status information.

IMPORTANT	Do not write to reserved words in the status file. If you intend writing to status file data, it is imperative that you first understand the function fully.

Status File Overview

The status file (S:) contains the following words:

Status File Words

Address	Function	Page
S:0	Arithmetic Flags	504
S:1	Controller Mode	505
S:2	STI Mode	510
S:2/9	Memory Module Program Compare	511
S:2/15	Math Overflow Selection	511
S:3H	Watchdog Scan Time	512
S:4	Free Running Clock	512
S:5	Minor Error Bits	513
S:6	Major Error Code	515
S:7	Suspend Code	515
S:8	Suspend File	516
S:9	Active Nodes (Nodes 0 to 15)	516
S:10	Active Nodes (Nodes 16 to 31)	516
S:13, S:14	Math Register	516
S:15L	Node Address	517
S:15H	Baud Rate	517
S:22	Maximum Scan Time	517
S:29	User Fault Routine File Number	517
S:30	STI Set Point	518
S:31	STI File Number	518
S:33	Channel 0 Communications	518

Status File Words

Address	Function	Page
S:35	Last 100 µSec Scan Time	519
S:36/10	Data File Overwrite Protection Lost	520
S:37	RTC Year	520
S:38	RTC Month	520
S:39	RTC Day of Month	520
S:40	RTC Hours	521
S:41	RTC Minutes	521
S:42	RTC Seconds	521
S:53	RTC Day of Week	522
S:57	OS Catalog Number	522
S:58	OS Series	522
S:59	OS FRN	522
S:60	Processor Catalog Number	522
S:61	Processor Series	523
S:62	Processor Revision	523
S:63	User Program Functionality Type	523
S:64L	Compiler Revision - Build Number	523
S:64H	Compiler Revision - Release	523

Status File Details

Arithmetic Flags

The arithmetic flags are assessed by the processor following the execution of any math, logical, or move instruction. The state of these bits remains in effect until the next math, logical, or move instruction in the program is executed.

Carry Flag

Address	Data Format	Range	Type	User Program Access
S:0/0	binary	0 or 1	status	read/write

This bit is set (1) if a mathematical carry or borrow is generated. Otherwise the bit remains cleared (0). When a STI, High-Speed Counter, Event Interrupt, or User Fault Routine interrupts normal execution of your program, the original value of S:0/0 is restored when execution resumes.

OverFlow Flag

Address	Data Format	Range	Type	User Program Access
S:0/1	binary	0 or 1	status	read/write

This bit is set (1) when the result of a mathematical operation does not fit in the destination. Otherwise the bit remains cleared (0). Whenever this bit is set (1),

the overflow trap bit S:5/0 is also set (1). When an STI, High-Speed Counter, Event Interrupt, or User Fault Routine interrupts normal execution of your program, the original value of S:0/1 is restored when execution resumes.

Zero Flag

Address	Data Format	Range	Type	User Program Access
S:0/2	binary	0 or 1	status	read/write

This bit is set (1) when the result of a mathematical operation or data handling instruction is zero. Otherwise the bit remains cleared (0). When an STI, High-Speed Counter, Event Interrupt, or User Fault Routine interrupts normal execution of your program, the original value of S:0/2 is restored when execution resumes.

Sign Flag

Address	Data Format	Range	Type	User Program Access
S:0/3	binary	0 or 1	status	read/write

This bit is set (1) when the result of a mathematical operation or data handling instruction is negative. Otherwise the bit remains cleared (0). When a STI, High-Speed Counter, Event Interrupt, or User Fault Routine interrupts normal execution of your program, the original value of S:0/3 is restored when execution resumes.

Controller Mode

User Application Mode

Address	Data Format	Range	Type	User Program Access
S:1/0 to S:1/4	binary	0…1 1110	status	read only

Bits 0...4 function as follows:

S:1/0 to S:1/4					Mode ID	Controller Mode	Use by MicroLogix Controller[1]
S:1/4	S:1/3	S:1/2	S:1/1	S:1/0			1400
0	0	0	0	0	0	remote download in progress	•
0	0	0	0	1	1	remote program mode	•
0	0	0	1	1	3	remote suspend mode (operation halted by execution of the SUS instruction)	•
0	0	1	1	0	6	remote run mode	•
0	0	1	1	1	7	remote test continuous mode	•
0	1	0	0	0	8	remote test single scan mode	•
1	0	0	0	0	16	download in progress	•
1	0	0	0	1	17	program mode	•
1	1	0	1	1	27	suspend mode (operation halted by execution of the SUS instruction)	•
1	1	1	1	0	30	run mode	•

(1) Valid modes are indicated by the (•) symbol. N/A indicates an invalid mode for that controller.

Forces Enabled

Address	Data Format	Range	Type	User Program Access
S:1/5	binary	1	control	read only

When user sets (1) this bit, I/O forcing is enabled. When the bit is reset (0), I/O forcing is disabled. Enabling I/O force means user can force I/O by writing I/O force files. When I/O force is disabled, whatever is written in the I/O force file does not affect physical output or input data file.

TIP This bit is can only be modified in offline mode. User must set/reset this bit first and then download the program in order for the change to take effect.

Forces Installed

Address	Data Format	Range	Type	User Program Access
S:1/6	binary	0 or 1	status	read only

This bit is set (1) by the controller to indicate that 1 or more inputs or outputs are forced. When this bit is clear, a force condition is not present within the controller.

Fault Override At Power-Up

Address	Data Format	Range	Type	User Program Access
S:1/8	binary	0 or 1	control	read only

When set (1), causes the controller to clear the Major Error Halted bit (S:1/13) at power-up. The power-up mode is determined by the controller mode switch and the Power-Up Mode Behavior Selection bit (S:1/12).

See also:FO - Fault Override on page 43.

Startup Protection Fault

Address	Data Format	Range	Type	User Program Access
S:1/9	binary	0 or 1	control	read only

When set (1) and the controller powers up in the RUN or REM RUN mode, the controller executes the User Fault Routine prior to the execution of the first scan of your program. You have the option of clearing the Major Error Halted bit (S:1/13) to resume operation. If the User Fault Routine does not clear bit S:1/13, the controller faults and does not enter an executing mode. Program the User Fault Routine logic accordingly.

TIP When executing the startup protection fault routine, S:6 (major error fault code) contains the value 0016H.

Load Memory Module On Error Or Default Program

Address	Data Format	Range	Type	User Program Access
S:1/10	binary	0 or 1	control	read only

For this option to work, you must set (1) this bit in the control program before downloading the program to a memory module. When this bit it set in the memory module and power is applied, the controller downloads the memory module program when the control program is corrupt or a default program exists in the controller.

TIP If you clear the controller memory, the controller loads the default program.

The mode of the controller after the transfer takes place is determined by the controller mode switch and the Power-Up Mode Behavior Selection bit (S:1/12).

See also:LE - Load on Error on page 43.

Load Memory Module Always

Address	Data Format	Range	Type	User Program Access
S:1/11	binary	0 or 1	control	read only

For this option to work, you must set (1) this bit in the control program before downloading the program to a memory module. When this bit is set in the memory module and power is applied, the controller downloads the memory module program.

The mode of the controller after the transfer takes place is determined by the controller mode switch and the Power-Up Mode Behavior Selection bit (S:1/12).

See also:LA - Load Always on page 44.

Power-Up Mode Behavior

Address	Data Format	Range	Type	User Program Access
S:1/12	binary	0 or 1	control	read only

If Power-Up Mode Behavior is clear (0 = Last State), the mode at power-up is dependent upon the:

- position of the mode switch
- state of the Major Error Halted flag (S:1/13)
- mode at the previous power down

If Power Up Mode Behavior is set (1 = Run), the mode at power-up is dependent upon the:

- position of the mode switch
- state of the Major Error Halted flag (S:1/13)

IMPORTANT	If you want the controller to power-up and enter the Run mode, regardless of any previous fault conditions, you must also set the Fault Override bit (S:1/8) so that the Major Error Halted flag is cleared before determining the power up mode.

The following table shows the Power-Up Mode under various conditions

MicroLogix 1400 - Mode Switch Position at Power-Up	Major Error Halted	Power-Up Mode Behavior	Mode at Last Power-Down	Power-Up Mode
Program	False	Don't Care	Don't Care	Program
	True			Program w/Fault
Remote	False	Last State	REM Download, Download, REM Program, Program or Any Test mode	REM Program
			REM Suspend or Suspend	REM Suspend
			REM Run or Run	REM Run
		Run	Don't Care	REM Run
	True	Don't Care	Don't Care	REM Program w/Fault
Run	False	Last State	REM Suspend or Suspend	Suspend
			Any Mode except REM Suspend or Suspend	Run
		Run	Don't Care	Run
	True	Don't Care	Don't Care	Run w/Fault(1)

(1) Run w/Fault is a fault condition, just as if the controller were in the Program /w Fault mode (outputs are reset and the controller program is not being executed). However, the controller enters Run mode as soon as the Major Error Halted flag is cleared.

See also:MB - Mode Behavior on page 44.

Major Error Halted

Address	Data Format	Range	Type	User Program Access
S:1/13	binary	0 or 1	status	read/write

The controller sets (1) this bit when a major error is encountered. The controller enters a fault condition and word S:6 contains the Fault Code that can be used to diagnose the condition. Any time bit S:1/13 is set, the controller:

- turns all outputs off and flashes the FAULT LED,
- or, enters the User Fault Routine allowing the control program to attempt recovery from the fault condition. If the User Fault Routine is able to clear S:1/13 and the fault condition, the controller continues to execute the control program. If the fault cannot be cleared, the outputs are cleared and the controller exits its executing mode and the FAULT LED flashes.

ATTENTION: If you clear the Major Error Halted bit (S:1/13) when the controller mode switch is in the RUN position, the controller immediately enters the RUN mode.

Future Access (OEM Lock)

Address	Data Format	Range	Type	User Program Access
S:1/14	binary	0 or 1	status	read only

When this bit is set (1), it indicates that the programming device must have an exact copy of the controller program.

See Allow Future Access Setting (OEM Lock) on page 34 for more information.

First Scan Bit

Address	Data Format	Range	Type	User Program Access
S:1/15	binary	0 or 1	status	read/write

When the controller sets (1) this bit, it indicates that the first scan of the user program is in progress (following entry into an executing mode). The controller clears this bit after the first scan.

TIP The First Scan bit (S:1/15) is set during execution of the start-up protection fault routine. Refer to S:1/9 for more information.

STI Mode

STI Pending

Address(1)	Data Format	Range	Type	User Program Access
S:2/0	binary	0 or 1	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated at STI:0/UIP. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.

STI Enabled

Address(1)	Data Format	Range	Type	User Program Access
S:2/1	binary	0 or 1	control	read/write

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated at STI:0/TIE. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.

STI Executing

Address(1)	Data Format	Range	Type	User Program Access
S:2/2	binary	0 or 1	control	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated at STI:0/UIX. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.

Memory Module Program Compare

Address	Data Format	Range	Type	User Program Access
S:2/9	binary	0 or 1	control	read only

When this bit is set (1) in the controller, its user program and the memory module user program must match for the controller to enter an executing mode.

If the user program does not match the memory module program, or if the memory module is not present, the controller faults with error code 0017H on any attempt to enter an executing mode.

See also:LPC - Load Program Compare on page 43.

Math Overflow Selection

Address	Data Format	Range	Type	User Program Access
S:2/14	binary	0 or 1	control	read/write

Set (1) this bit when you intend to use 32-bit addition and subtraction. When S:2/14 is set, and the result of an ADD, SUB, MUL, or DIV instruction cannot be represented in the destination address (underflow or overflow),

- the overflow bit S:0/1 is set,
- the overflow trap bit S:5/0 is set,
- and the destination address contains the unsigned truncated least significant 16 or 32 bits of the result.

The default condition of S:2/14 is cleared (0). When S:2/14 is cleared (0), and the result of an ADD, SUB, MUL, or DIV instruction cannot be represented in the destination address (underflow or overflow),

- the overflow bit S:0/1 is set,
- the overflow trap bit S:5/0 is set,
- the destination address contains +32,767 (word) or +2,147,483,647 (long word) if the result is positive; or -32,768 (word) or -2,147,483,648 (long word) if the result is negative.

To provide protection from inadvertent alteration of your selection, program an unconditional OTL instruction at address S:2/14 to ensure the new math overflow operation. Program an unconditional OTU instruction at address S:2/14 to ensure the original math overflow operation.

Watchdog Scan Time

Address	Data Format	Range	Type	User Program Access
S:3H	Byte	2…255	control	read/write

This byte value contains the number of 10 ms intervals allowed to occur during a program cycle. The timing accuracy is from -10 ms to +0 ms. This means that a value of 2 results in a timeout between 10 and 20 ms.

If the program scan time value equals the watchdog value, a watchdog major error is generated (code 0022H).

Free Running Clock

Address	Data Format	Range	Type	User Program Access
S:4	binary	0…FFFF	status	read/write

This register contains a free running counter. This word is cleared (0) upon entering an executing mode.

Bits in status word 4 can be monitored by the user program. The bits turn on and off at a particular rate (cycle time). The On/Off times are identical, and are added together to determine the cycle time.

S:4 Free Running Clock Comparison for SLC 500 and MicroLogix Controllers

The Free Running Clocks in the SLC 500 and MicroLogix controllers function the same, but have different resolutions. The resolution of the Free Running Clock depends upon which controller you are using.

- SLC 500 and MicroLogix 1000: 10 ms/bit (0.010 seconds/bit)
- MicroLogix 1100, MicroLogix 1200, MicroLogix 1400 and MicroLogix 1500: 100 μs/bit (0.0001 seconds/bit)

The following table illustrates the differences.

Free Running Clock Cycle Times (all Times are in Seconds)

Bit	SLC 500 and MicroLogix 1000		MicroLogix 1100, MicroLogix 1200, MicroLogix 1400 and MicroLogix 1500	
	On/Off Time	Cycle Time	On/Off Time	Cycle Time
S:4/0	0.010	0.020	0.0001	0.0002
S:4/1	0.020	0.040	0.0002	0.0004
S:4/2	0.040	0.080	0.0004	0.0008
S:4/3	0.080	0.160	0.0008	0.0160
S:4/4	0.160	0.320	0.0016	0.0320
S:4/5	0.320	0.640	0.0032	0.0640
S:4/6	0.640	1.280	0.0064	0.1280

Free Running Clock Cycle Times (all Times are in Seconds)

Bit	SLC 500 and MicroLogix 1000		MicroLogix 1100, MicroLogix 1200, MicroLogix 1400 and MicroLogix 1500	
	On/Off Time	Cycle Time	On/Off Time	Cycle Time
S:4/7	1.280	2.560	0.0128	0.2560
S:4/8	2.560	5.120	0.0256	0.5120
S:4/9	5.120	10.240	0.0512	0.1024
S:4/10	10.240	20.480	0.1024	0.2048
S:4/11	20.480	40.960	0.2048	0.4096
S:4/12	40.960	81.92	0.4096	0.8192
S:4/13	81.92	163.84	0.8192	1.6384
S:4/14	163.84	327.68	1.6384	3.2768
S:4/15	327.68	655.36	3.2768	6.5536

For example, if bit S:4/7 is monitored in an SLC 500, then that bit will be on for 1.28 seconds and off for 1.28 seconds for a total cycle time of 2.56 seconds. If bit S:4/7 is monitored in a MicroLogix 1400, then that bit will be on for 0.0128 seconds and off for 0.0128 seconds for a total cycle time of 0.0256 seconds.

Minor Error Bits

Overflow Trap Bit

Address	Data Format	Range	Type	User Program Access
S:5/0	binary	0 or 1	status	read/write

If this bit is ever set (1) upon execution of the END or TND instruction, a major error (0020H) is generated. To avoid this type of major error from occurring, examine the state of this bit following a math instruction (ADD, SUB, MUL, DIV, NEG, SCL, TOD, or FRD), take appropriate action, and then clear bit S:5/0 using an OTU instruction with S:5/0.

Control Register Error

Address	Data Format	Range	Type	User Program Access
S:5/2	binary	0 or 1	status	read/write

The LFU, LFL, FFU, FFL, BSL, BSR, SQO, SQC, and SQL instructions are capable of generating this error. When bit S:5/2 is set (1), it indicates that the error bit of a control word used by the instruction has been set.

If this bit is ever set upon execution of the END or TND instruction, major error (0020H) is generated. To avoid this type of major error from occurring, examine the state of this bit following a control register instruction, take appropriate action, and then clear bit S:5/2 using an OTU instruction with S:5/2.

Major Error Detected in User Fault Routine

Address	Data Format	Range	Type	User Program Access
S:5/3	binary	0 or 1	status	read/write

When set (1), the major error code (S:6) represents the major error that occurred while processing the User Fault Routine due to another major error.

Memory Module Boot

Address	Data Format	Range	Type	User Program Access
S:5/8	binary	0 or 1	status	read/write

When this bit is set (1) by the controller, it indicates that a memory module program has been transferred due to S:1/10 (Load Memory Module on Error or Default Program) or S:1/11 (Load Memory Module Always) being set in an attached memory module user program. This bit is not cleared (0) by the controller.

Your program can examine the state of this bit on the first scan (using bit S:1/15) on entry into an Executing mode to determine if the memory module user program has been transferred after a power-up occurred. This information is useful when you have an application that contains retentive data and a memory module has bit S:1/10 or bit S:1/11 set.

Memory Module Password Mismatch

Address	Data Format	Range	Type	User Program Access
S:5/9	binary	0 or 1	status	read/write

At power-up, if Load Always is set, and the controller and memory module passwords do not match, the Memory Module Password Mismatch bit is set (1).

See Program Password Protection on page 30 for more information.

STI Lost

Address[1]	Data Format	Range	Type	User Program Access
S:5/10	binary	0 or 1	status	read/write

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated at STI:0/UIL. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.

Processor Battery Low

Address	Data Format	Range	Type	User Program Access
S:5/11	binary	0 or 1	status	read only

This bit is set (1) when the battery is low.

IMPORTANT Install a replacement battery immediately. See your hardware manual for more information.

See also: RTC Battery Operation on page 40.

Input Filter Selection Modified

Address	Data Format	Range	Type	User Program Access
S:5/13	binary	0 or 1	status	read/write

This bit is set (1) whenever the discrete input filter selection in the control program is not compatible with the hardware.

ASCII String Manipulation Error

Address	Data Format	Range	Type	User Program Access
S:5/15	binary	0 or 1	status	read

This bit is set (1) whenever an invalid string length occurs. When S:5/15 is set, the Invalid String Length Error (1F39H) is written to the Major Error Fault Code word (S:6).

Major Error Code

Address	Data Format	Range	Type	User Program Access
S:6	word	0…FFFF	status	read/write

This register displays a value which can be used to determine what caused a fault to occur. See Identifying Controller Faults on page 525 to learn more about troubleshooting faults.

Suspend Code

Address	Data Format	Range	Type	User Program Access
S:7	word	-32,768…+32,767	status	read/write

When the controller executes an Suspend (SUS) instruction, the SUS code is written to this location, S:7. This pinpoints the conditions in the application that caused the Suspend mode. The controller does not clear this value.

Use the SUS instruction with startup troubleshooting, or as runtime diagnostics for detection of system errors.

Suspend File

Address	Data Format	Range	Type	User Program Access
S:8	word	0…255	status	read/write

When the controller executes an Suspend (SUS) instruction, the SUS file is written to this location, S:8. This pinpoints the conditions in the application that caused the Suspend mode. The controller does not clear this value.

Use the SUS instruction with startup troubleshooting, or as runtime diagnostics for detection of system errors.

Active Nodes (Nodes 0 to 15)

Address[1]	Data Format	Range	Type	User Program Access
S:9	word	0…FFFF	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File (CSx:0.27). See Active Node Table Block on page 59 for more information.

Active Nodes (Nodes 16 to 31)

Address[1]	Data Format	Range	Type	User Program Access
S:10	word	0…FFFF	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File (CSx:0.28). See Active Node Table Block on page 59 for more information.

Math Register

Address	Data Format	Range	Type	User Program Access
S:13 (low byte)	word	-32,768…+32,767	status	read/write
S:14 (high byte)	word	-32,768…+32,767	status	read/write

These two words are used in conjunction with the MUL, DIV, FRD, and TOD math instructions. The math register value is assessed upon execution of the instruction and remains valid until the next MUL, DIV, FRD, or TOD instruction is executed in the user program.

Node Address

Address[1]	Data Format	Range	Type	User Program Access
S:15 (low byte)	byte	0…255	status	read only

(1) This byte can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File (CSx:0.5/0 through CSx:0.5/7). See General Channel Status Block on page 45 for more information.

Baud Rate

Address[1]	Data Format	Range	Type	User Program Access
S:15 (high byte)	byte	0…255	status	read only

(1) This byte can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File (CSx:0.5/8 through CSx:0.5/15). See General Channel Status Block on page 45 for more information.

Maximum Scan Time

Address	Data Format	Range	Type	User Program Access
S:22	word	0…32,767	status	read/write

This word indicates the maximum observed interval between consecutive program scans.

The controller compares each scan value to the value contained in S:22. If a scan value is larger than the previous, the larger value is stored in S:22.

This value indicates, in 100 us increments, the time elapsed in the longest program cycle of the controller. Resolution is -100 µs to +0 µs. For example, the value 9 indicates that 800 to 900 us was observed as the longest program cycle.

User Fault Routine File Number

Address	Data Format	Range	Type	User Program Access
S:29	word	0…255	status	read only

This register is used to control which subroutine executes when a User Fault is generated.

STI Set Point

Address[1]	Data Format	Range	Type	User Program Access
S:30	word	0…65535	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated at STI:0/SPM. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.

STI File Number

Address[1]	Data Format	Range	Type	User Program Access
S:31	word	0…65535	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated at STI:0/PFN. See Using the Selectable Timed Interrupt (STI) Function File on page 272 for more information.

Channel 0 Communications

Incoming Command Pending

Address[1]	Data Format	Range	Type	User Program Access
S:33/0	binary	0 or 1	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File at CS0:0.4/0. See General Channel Status Block on page 45 for more information.

Message Reply Pending

Address[1]	Data Format	Range	Type	User Program Access
S:33/1	binary	0 or 1	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File at CS0:0.4/1. See General Channel Status Block on page 45 for more information.

Outgoing Message Command Pending

Address(1)	Data Format	Range	Type	User Program Access
S:33/2	binary	0 or 1	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File at CS0:0.4/2. See General Channel Status Block on page 45 for more information.

Communications Mode Selection

Address(1)	Data Format	Range	Type	User Program Access
S:33/3	binary	0 or 1	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File at CS0:0.4/3. See General Channel Status Block on page 45 for more information.

Communications Active

Address(1)	Data Format	Range	Type	User Program Access
S:33/4	binary	0 or 1	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Communications Status File at CS0:0.4/4. See General Channel Status Block on page 45 for more information.

Scan Toggle Bit

Address	Data Format	Range	Type	User Program Access
S:33/9	binary	0 or 1	status	read/write

The controller changes the status of this bit at the end of each scan. It is reset upon entry into an executing mode.

Last 100 µSec Scan Time

Address	Data Format	Range	Type	User Program Access
S:35	word	0…32,767	status	read/write

This register indicates the elapsed time for the last program cycle of the controller (in 100 µs increments).

Data File Overwrite Protection Lost

Address	Data Format	Range	Type	User Program Access
S:36/10	binary	0 or 1	status	read/write

When clear (0), this bit indicates that at the time of the last program transfer to the controller, protected data files in the controller were not overwritten, or there were no protected data files in the program being downloaded.

When set (1), this bit indicates that data has been overwritten. See User Program Transfer Requirements on page 27 for more information.

See Setting Download File Protection on page 27 for more information.

RTC Year

Address[1]	Data Format	Range	Type	User Program Access
S:37	word	1998…2097	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.YR. See Real-Time Clock Function File on page 38 for more information. **Note:** This value will not update while viewing online in RSLogix 500/RSLogix Micro/ RSLogix Micro. Monitor address in function file to see online values.

RTC Month

Address[1]	Data Format	Range	Type	User Program Access
S:38	word	1…12	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.MON. See Real-Time Clock Function File on page 38 for more information. **Note:** This value will not update while viewing online in RSLogix 500/RSLogix Micro. Monitor address in function file to see online values.

RTC Day of Month

Address[1]	Data Format	Range	Type	User Program Access
S:39	word	1…31	status	read only

(1) This bit can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.DAY. See Real-Time Clock Function File on page 38 for more information. **Note:** This

value will not update while viewing online in RSLogix 500/RSLogix Micro. Monitor address in function file to see online values.

RTC Hours

Address(1)	Data Format	Range	Type	User Program Access
S:40	word	0…23	status	read only

(1) This word can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.HR. See Real-Time Clock Function File on page 38 for more information. **Note:** This value will not update while viewing online in RSLogix 500/RSLogix Micro. Monitor address in function file to see online values.

RTC Minutes

Address(1)	Data Format	Range	Type	User Program Access
S:41	word	0…59	status	read only

(1) This word can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.MIN. See Real-Time Clock Function File on page 38 for more information. **Note:** This value will not update while viewing online in RSLogix 500/RSLogix Micro. Monitor address in function file to see online values.

RTC Seconds

Address(1)	Data Format	Range	Type	User Program Access
S:42	word	0…59	status	read only

(1) This word can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.SEC. See Real-Time Clock Function File on page 38 for more information. **Note:** This value will not update while viewing online in RSLogix 500/RSLogix Micro. Monitor address in function file to see online values.

RTC Day of Week

Address[1]	Data Format	Range	Type	User Program Access
S:53	word	0…6	status	read only

(1) This word can only be accessed via ladder logic. It cannot be accessed via communications (such as a Message instruction from another device).

This address is duplicated in the Real-Time Clock Function File at RTC:0.DOW. See Real-Time Clock Function File on page 38 for more information. **Note:** This value will not update while viewing online in RSLogix 500/RSLogix Micro. Monitor address in function file to see online values.

OS Catalog Number

Address	Data Format	Range	Type	User Program Access
S:57	word	0…32,767	status	read only

This register identifies the Catalog Number for the Operating System in the controller.

OS Series

Address	Data Format	Range	Type	User Program Access
S:58	ASCII	A…Z	status	read only

This register identifies the Series letter for the Operating System in the controller.

OS FRN

Address	Data Format	Range	Type	User Program Access
S:59	word	0…32,767	status	read only

This register identifies the FRN of the Operating System in the controller.

Processor Catalog Number

Address	Data Format	Range	Type	User Program Access
S:60	ASCII	"A"…"ZZ"	status	read only

This register identifies the Catalog Number for the processor.

Processor Series

Address	Data Format	Range	Type	User Program Access
S:61	ASCII	A…Z	status	read only

This register identifies the Series of the processor.

Processor Revision

Address	Data Format	Range	Type	User Program Access
S:62	word	0…32,767	status	read only

This register identifies the revision (Boot FRN) of the processor.

User Program Functionality Type

Address	Data Format	Range	Type	User Program Access
S:63	word	0…32,767	status	read only

This register identifies the level of functionality of the user program in the controller.

Compiler Revision - Build Number

Address	Data Format	Range	Type	User Program Access
S:64 (low byte)	byte	0…255	status	read only

This register identifies the Build Number of the compiler which created the program in the controller.

Compiler Revision - Release

Address	Data Format	Range	Type	User Program Access
S:64 (high byte)	byte	0…255	status	read only

This register identifies the Release of the compiler which created the program in the controller.

Notes:

Fault Messages and Error Codes

This chapter describes how to troubleshoot your controller. Topics include:

- identifying controller faults
- contacting Rockwell Automation for assistance

Identifying Controller Faults

While a program is executing, a fault may occur within the operating system or your program. When a fault occurs, you have various options to determine what the fault is and how to correct it. This section describes how to clear faults and provides a list of possible advisory messages with recommended corrective actions.

Automatically Clearing Faults

You can automatically clear a fault by cycling power to the controller when the Fault Override at Power-Up bit (S:1/8) is set in the status file.

You can also configure the controller to clear faults and go to RUN every time the controller is power cycled. This is a feature that OEMs can build into their equipment to allow end users to reset the controller. If the controller faults, it can be reset by simply cycling power to the machine. To accomplish this, set the following bits in the status file:

- S2:1/8 - Fault Override at Power-up
- S2:1/12 - Mode Behavior

If the fault condition still exists after cycling power, the controller re-enters the fault mode. For more information on status bits, see System Status File on page 503.

TIP You can declare your own application-specific major fault by writing your own unique value to S:6 and then setting bit S:1/13 to prevent reusing system defined codes. The recommended values for user-defined faults are FF00 to FF0F.

Manually Clearing Faults Using the Fault Routine

The occurrence of recoverable or non-recoverable user faults can cause the user fault subroutine to be executed. If the fault is recoverable, the subroutine can be used to correct the problem and clear the fault bit S:1/13. The controller then continues in the Run or test mode.

The subroutine does not execute for non-user faults. See User Fault Routine on page 265 for information on creating a user fault subroutine.

Fault Messages

This section contains fault messages that can occur during operation of the MicroLogix 1400 programmable controller. Each table lists the error code description, the probable cause, and the recommended corrective action.

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
0001	NVRAM ERROR	The default program is loaded to the controller memory. This occurs: • if a power down occurred during program download or transfer from the memory module. • RAM integrity test failed.	Non-User	• Re-download or transfer the program. • Verify battery is connected. • Contact your local Rockwell Automation representative if the error persists.
0002	UNEXPECTED RESET	• The controller was unexpectedly reset due to a noisy environment or internal hardware failure. • The default program is loaded.	Non-User	• Refer to proper grounding guidelines and using surge suppressors in your controller's User Manual. • Verify battery is connected. • Contact your local Rockwell Automation representative if the error persists.
0003	MEMORY MODULE USER PROGRAM IS CORRUPT	Memory module memory error. This error can also occur when going to the Run mode.	Non-User	Re-program the memory module. If the error persists, replace the memory module.
0004	MEMORY INTEGRITY ERROR	While the controller was powered up, ROM or RAM became corrupt or when background integrity check failed.	Non-User	• Cycle power on your unit. Then, re-download your program and start up your system. • Refer to proper grounding guidelines and using surge suppressors in your controller's User Manual. • Contact your local Rockwell Automation representative if the error persists.
0005	Reserved	N/A	N/A	

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
0006	MEMORY MODULE HARDWARE FAULT	The memory module hardware faulted or the memory module is incompatible with OS.	Non-User	• Upgrade the OS to be compatible with memory module. • Obtain a new memory module.
0007	MEMORY MODULE TRANSFER ERROR	Failure during memory module transfer.	Non-User	Re-attempt the transfer. If the error persists, replace the memory module.
0008	FATAL INTERNAL SOFTWARE ERROR	An unexpected software error occurred.	Non-User	• Cycle power on your unit. Then, re-download your program and re-initialize any necessary data. • Start up your system. • Refer to proper grounding guidelines and using surge suppressors in your controller's User Manual. • Contact your local Rockwell Automation representative if the error persists.
0009	FATAL INTERNAL HARDWARE ERROR	An unexpected hardware error occurred.	Non-User	• Cycle power on your unit. Then, re-download your program and re-initialize any necessary data. • Start up your system. • Refer to proper grounding guidelines and using surge suppressors in your controller's User Manual. • Contact your local Rockwell Automation representative if the error persists.
000A	OS MISSING OR CORRUPT	The operating system required for the user program is corrupt or missing.	Non-User	• Download a new OS using ControlFlash. • Contact your local Rockwell Automation representative for more information about available operating systems your controller.
000B	BASE HARDWARE FAULT	The base hardware faulted or is incompatible with the OS.	Non-User	• Upgrade the OS using ControlFlash. • Replace the Controller *(MicroLogix 1100, MicroLogix 1200, MicroLogix 1400)*. • Replace the Base Unit *(MicroLogix 1500 only)*. • Contact your local Rockwell Automation representative for more information about available operating systems your controller.
0011	EXECUTABLE FILE 2 IS MISSING	Ladder File 2 is missing from the program.	Non-User	• Re-compile and reload the program.

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
0012	LADDER PROGRAM ERROR	The ladder program has a memory integrity problem.	Non-User	• Reload the program or re-compile and reload the program. If the error persists, be sure to use RSI programming software to develop and load the program. • Refer to proper grounding guidelines and using surge suppressors in your controller's User Manual.
0015	I/O CONFIGURATION FILE ERROR	The user program I/O configuration is invalid.	Non-User	Re-compile and reload the program, and enter the Run mode. If the error persists, be sure to use RSI programming software to develop and load the program.
0016	STARTUP PROTECTION FAULT	The user fault routine was executed at power-up, prior to the main ladder program. Bit S:1/13 (Major Error Halted) was not cleared at the end of the User Fault Routine. The User Fault Routine ran because bit S:1/9 was set at power-up.	Recoverable	• Either reset bit S:1/9 if this is consistent with the application requirements, and change the mode back to RUN, or • clear S:1/13, the Major Error Halted bit, before the end of the User Fault Routine.
0017	NVRAM/MEMORY MODULE USER PROGRAM MISMATCH	Bit S:2/9 is set in the controller and the memory module user program does not match the controller user program.	Non-Recoverable	Transfer the memory module program to the controller and then change to Run mode.
0018	MEMORY MODULE USER PROGRAM INCOMPATIBLE WITH OS	The user program in the memory module is incompatible with the OS.	Non-User	• Upgrade the OS using ControlFlash to be compatible with the memory module. • Obtain a new memory module. • Contact your local Rockwell Automation representative for more information about available operating systems your controller.
001A	USER PROGRAM INCOMPATIBLE WITH OS AT POWER-UP	The user program is incompatible with the OS.	Non-User	• Upgrade the OS using ControlFlash. • Contact your local Rockwell Automation representative for more information about available operating systems your controller.
0020	MINOR ERROR AT END-OF-SCAN DETECTED	A minor fault bit (bits 0-7) in S:5 was set at the end of scan.	Recoverable	• Correct the instruction logic causing the error. • Enter the status file display in your programming software and clear the fault. • Enter the Run mode.
0021	Reserved	N/A	N/A	
0022	WATCHDOG TIMER EXPIRED, SEE S:3	The program scan time exceeded the watchdog timeout value (S:3H).	Non-Recoverable	• Determine if the program is caught in a loop and correct the problem. • Increase the watchdog timeout value in the status file.
0023	STI ERROR	An error occurred in the STI configuration.	Recoverable	See the Error Code in the STI Function File for the specific error.

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
0028	INVALID OR NONEXISTENT USER FAULT ROUTINE VALUE	• A fault routine number was entered in the status file, number (S:29), but either the fault routine was not physically created, or • the fault routine number was less than 3 or greater than 255.	Non-User	• Either clear the fault routine file number (S:29) in the status file, or • create a fault routine for the file number reference in the status file (S:29). The file number must be greater than 2 and less than 256.
0029	INSTRUCTION INDIRECTION OUTSIDE OF DATA SPACE	An indirect address reference in the ladder program is outside of the entire data file space.	Recoverable	Correct the program to ensure that there are no indirect references outside data file space. Re-compile, reload the program and enter the Run mode.
002E	EII ERROR	An error occurred in the EII configuration.	Recoverable	See the Error Code in the EII Function File for the specific error.
0030	SUBROUTINE NESTING EXCEEDS LIMIT	The JSR instruction nesting level exceeded the controller memory space.	Non-User	Correct the user program to reduce the nesting levels used and to meet the restrictions for the JSR instruction. Then reload the program and Run.
0031	UNSUPPORTED INSTRUCTION DETECTED	The program contains an instruction(s) that is not supported by the controller.	Non-User	• Modify the program so that all instructions are supported by the controller. • Re-compile and reload the program and enter the Run mode.
0032	SQO/SQC/SQL OUTSIDE OF DATA FILE SPACE	A sequencer instruction length/position parameter references outside of the entire data file space.	Recoverable	• Correct the program to ensure that the length and position parameters do not point outside data file space. • Re-compile, reload the program and enter the Run mode.
0033	BSL/BSR/FFL/FFU/LFL/LFU CROSSED DATA FILE SPACE	The length/position parameter of a BSL, BSR, FFL, FFU, LFL, or LFU instruction references outside of the entire data file space.	Recoverable	• Correct the program to ensure that the length and position parameters do not point outside of the data space. • Re-compile, reload the program and enter the Run mode.
0034	NEGATIVE VALUE IN TIMER PRESET OR ACCUMULATOR	A negative value was loaded to a timer preset or accumulator.	Recoverable	• If the program is moving values to the accumulated or preset word of a timer, make certain these values are not negative. • Reload the program and enter the Run mode.
0035	ILLEGAL INSTRUCTION IN INTERRUPT FILE	The program contains a Temporary End (TND), Refresh (REF), or Service Communication instruction in an interrupt subroutine (STI, EII, HSC) or user fault routine.	Non-Recoverable	• Correct the program. • Re-compile, reload the program and enter the Run mode.
0036	INVALID PID PARAMETER	An invalid value is being used for a PID instruction parameter.	Recoverable	See page 281, Process Control Instruction for more information about the PID instruction.
0037	HSC ERROR	An error occurred in the HSC configuration.	Recoverable	See the Error Code in the HSC Function File for the specific error.
003B	PTOX ERROR	An error occurred in the PTOX instruction configuration.	Recoverable or Non-User	See the Error Code in the PTOX Function File for the specific error.

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
003C	PWMX ERROR	An error occurred in the PWMX instruction configuration.	Recoverable or Non-User	See the Error Code in the PWMX Function File for the specific error.
003D	INVALID SEQUENCER LENGTH/POSITION	A sequencer instruction (SQO, SQC, SQL) length/position parameter is greater than 255.	Recoverable	Correct the user program, then re-compile, reload the program and enter the Run mode.
003E	INVALID BIT SHIFT OR LIFO/FIFO PARAMETER	A BSR or BSL instruction length parameter is greater than 2048 or an FFU, FFL, LFU, LFL instruction length parameter is greater than 128 (word file) or greater than 64 (double word file)	Recoverable	Correct the user program or allocate more data file space using the memory map, then reload and Run.
003F	COP/CPW/FLL OUTSIDE OF DATA FILE SPACE	A COP, CPW or FLL instruction length parameter references outside of the entire data space.	Recoverable	• Correct the program to ensure that the length and parameter do not point outside of the data file space. • Re-compile, reload the program and enter the Run mode.
0042	INVALID RECIPE NUMBER	Number of Recipes specified is greater than 256.	Recoverable	• Correct the value for Number of Recipes. • Re-compile, reload the program and enter the Run mode.
0044	INVALID WRITE TO RTC FUNCTION FILE	Write attempt to RTC function file failed. This only occurs when attempting to write invalid data to the RTC function file. Examples of invalid data are: setting the Day of Week to zero, or setting the Date to February 30th.	Recoverable	• Correct the invalid data. • Re-compile, reload the program and enter the Run mode.
0050	CONTROLLER TYPE MISMATCH	A particular controller type was selected in the user program configuration, but did not match the actual controller type.	Non-User	• Connect to the hardware that is specified in the user program, or • Reconfigure the program to match the attached hardware.
0051	BASE TYPE MISMATCH	A particular hardware type (AWA, BWA, BXB, AWAA, BWAA, BXBA) was selected in the user program configuration, but did no match the actual base.	Non-User	• Connect to the hardware that is specified in the user program, or • Reconfigure the program to match the attached hardware.
0052	MINIMUM SERIES ERROR	The hardware minimum series selected in the user program configuration was greater than the series on the actual hardware.	Non-User	• Connect to the hardware that is specified in the user program, or • Reconfigure the program to match the attached hardware.
xx71[(1)]	EXPANSION I/O HARDWARE ERROR	The controller cannot communicate with an expansion I/O module.	Non-Recoverable	• Check connections. • Check for a noise problem and be sure proper grounding practices are used. • Replace the module. • Cycle power.

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
xx79[1]	EXPANSION I/O MODULE ERROR	An expansion I/O module generated an error.	Non-Recoverable	• Refer to the I/O Module Status (IOS) file. • Consult the documentation for your specific I/O module to determine possible causes of a module error.
xx81[1]	EXPANSION I/O HARDWARE ERROR	The controller cannot communicate with an expansion I/O module.	Non-User	• Check connections. • Check for a noise problem and be sure proper grounding practices are used. • Replace the module. • Cycle power.
0083	MAX I/O CABLES EXCEEDED	The maximum number of expansion I/O cables allowed was exceeded.	Non-User	• Reconfigure the expansion I/O system so that it has an allowable number of cables. • Cycle power.
0084	MAX I/O POWER SUPPLIES EXCEEDED	The maximum number of expansion I/O power supplies allowed was exceeded.	Non-User	• Reconfigure the expansion I/O system so that it has the correct number of power supplies.
0085	MAX I/O MODULES EXCEEDED	The maximum number of expansion I/O modules allowed was exceeded.	Non-User	• Reconfigure the expansion I/O system so that it has an allowable number of modules. • Cycle power.
xx86[1]	EXPANSION I/O MODULE BAUD RATE ERROR	An expansion I/O module could not communicate at the baud rate specified in the user program I/O configuration.	Non-User	• Change the baud rate in the user program I/O configuration, and • Re-compile, reload the program and enter the Run mode, or • Replace the module. • Cycle power.
xx87[1]	I/O CONFIGURATION MISMATCH	• The expansion I/O configuration in the user program did not match the actual configuration, or • The expansion I/O configuration in the user program specified a module, but one was not found, or • The expansion I/O module configuration data size for a module was greater than what the module is capable of holding.	Non-User	• Either correct the user program I/O configuration to match the actual configuration, or • With power off, correct the actual I/O configuration to match the user program configuration.
xx88[1]	EXPANSION I/O MODULE CONFIGURATION ERROR	The number of input or output image words configured in the user program exceeds the image size in the expansion I/O module.	Non-User	• Correct the user program I/O configuration to reduce the number of input or output words, and • Re-compile, reload the program and enter the Run mode.

Error Codes

Error Code (Hex)	Advisory Message	Description	Fault Classification	Recommended Action
xx89[1][2]	EXPANSION I/O MODULE ERROR	An expansion I/O module generated an error.	Non-User	• Refer to the I/O status file. • Consult the documentation for your specific I/O module to determine possible causes of a module error.
xx8A[1][2]	EXPANSION I/O CABLE CONFIGURATION MISMATCH ERROR	• Either an expansion I/O cable is configured in the user program, but no cable is present, or • an expansion I/O cable is configured in the user program and a cable is physically present, but the types do not match.	Non-User	• Correct the user program to eliminate a cable that is not present • Re-compile, reload the program and enter the Run mode, or • Add the missing cable. • Cycle power.
xx8B[1][2]	EXPANSION I/O POWER SUPPLY CONFIGURATION MISMATCH ERROR	• Either an expansion I/O power supply is configured in the user program, but no power supply is present, or • an expansion I/O power supply is configured in the user program and a power supply is physically present, but the types do not match.	Non-User	• Correct the user program to eliminate a power supply that is not present • Re-compile, reload the program and enter the Run mode, or • With power removed, add the missing power supply.
xx8C[1][2]	EXPANSION I/O OBJECT TYPE MISMATCH	An expansion I/O object (i.e. cable, power supply, or module) in the user program I/O configuration is not the same object type as is physically present.	Non-User	• Correct the user program I/O configuration so that the object types match the actual configuration, and • Re-compile, reload the program and enter the Run mode. Or • Correct the actual configuration to match the user program I/O configuration. • Cycle power.
0x1F39	INVALID STRING LENGTH	The first word of string data contains a negative, zero, or value greater than 82.	Recoverable	Check the first word of the string data element for invalid values and correct the data.

(1) xx indicates module number. If xx = 0, problem cannot be traced to a specific module.

(2) The xx in this error code means that the error occurs at the location of the last properly configured Expansion I/O module +1. You should use this information in conjunction with the specific error code to determine the source of the problem.

Contacting Rockwell Automation for Assistance

If you need to contact Rockwell Automation or local distributor for assistance, it is helpful to obtain the following information ready:

- controller type, series letter, and revision letter of the base unit
- series letter, revision letter, and firmware (FRN) number of the processor (on bottom side of processor unit)

TIP You can also check the FRN by looking at word S:59 (Operating System FRN) in the Status File.

- controller LED status
- controller error codes (found in S2:6 of status file) or LCD screen.

Rockwell Automation phone numbers are listed on the back cover of this manual.

To contact us via the Internet, go to http://www.rockwellautomation.com.

Notes:

Protocol Configuration

Use the information in this appendix for configuring communication protocols. The following protocols are supported from any RS-232 communication channel:

- DH-485
- DF1 Full-Duplex
- DF1 Half-Duplex
- DF1 Radio Modem
- Modbus RTU
- ASCII
- DNP3 Slave

This appendix is organized into the following sections:

- DH-485 Communication Protocol on page 536
- DF1 Full-Duplex Protocol on page 539
- DF1 Half-Duplex Protocol on page 540
- DF1 Radio Modem Protocol on page 549
- Modbus RTU Protocol on page 555
- ASCII Driver on page 566
- Ethernet Driver on page 567

See your controller's User Manual, 1766-UM001 for information about required network devices and accessories.

See Appendix F in your controller's User Manual, 1766-UM001 for more information about configuring serial channel(s) for DNP3 Slave.

DH-485 Communication Protocol

The information in this section describes the DH-485 network functions, network architecture, and performance characteristics. It also helps you plan and operate the controller on a DH-485 network.

DH-485 Network Description

The DH-485 protocol defines the communication between multiple devices that coexist on a single pair of wires. DH-485 protocol uses RS-485 Half-Duplex as its physical interface. (RS-485 is a definition of electrical characteristics; it is *not* a protocol.) RS-485 uses devices that are capable of co-existing on a common data circuit, thus allowing data to be easily shared between devices.

The DH-485 network offers:

- interconnection of 32 devices
- multi-master capability
- token passing access control
- the ability to add or remove nodes without disrupting the network
- maximum network length of 1219 m (4000 ft.)

The DH-485 protocol supports two classes of devices: initiators and responders. All initiators on the network get a chance to initiate message transfers. To determine which initiator has the right to transmit, a token passing algorithm is used.

The following section describes the protocol used to control message transfers on the DH-485 network.

DH-485 Token Rotation

A node holding the token can send a message onto the network. Each node is allowed a fixed number of transmissions (based on the Token Hold Factor) each time it receives the token. After a node sends a message, it passes the token to the next device.

The allowable range of node addresses is 0...31. There must be at least one initiator on the network (such as a MicroLogix controller, or an SLC 5/02 or higher processor).

DH-485 Broadcast Messages

A broadcast write command is sent as a DH-485 Send Data No Acknowledgement (SDN) packet. No acknowledgement or reply is returned.

DH-485 Configuration Parameters

When communications are configured for DH-485, the following parameters can be changed:

DH-485 Configuration Parameters

Parameter	Options	Programming Software Default
Baud Rate	9600, 19.2K	19.2K
Node Address	1…31 decimal	1
Token Hold Factor	1…4	2
Max Node Address	1…31	31

The major software issues you need to resolve before installing a network are discussed in the following sections.

Software Considerations

Software considerations include the configuration of the network and the parameters that can be set to the specific requirements of the network. The following are major configuration factors that have a significant effect on network performance:

- number of nodes on the network
- addresses of those nodes
- baud rate

The following sections explain network considerations and describe ways to select parameters for optimum network performance (speed). Refer to your programming software's documentation for more information.

Number of Nodes

The number of nodes on the network directly affects the data transfer time between nodes. Unnecessary nodes (such as a second programming terminal that is not being used) slow the data transfer rate. The maximum number of nodes on the network is 32.

Setting Node Addresses

The best network performance occurs when node addresses are assigned in sequential order. Initiators, such as personal computers, should be assigned the lowest numbered addresses to minimize the time required to initialize the network. The valid range for the MicroLogix controllers is 1...31 (controllers cannot be node 0). The default setting is 1. The node address is stored in the controller Communications Status file (CS0:5/0...CS0:5/7). Configure the node address via Channel Configuration using RSLogix 500/RSLogix Micro. Select the Channel 0 tab.

Setting Controller Baud Rate

The best network performance occurs at the highest baud rate, which is 19200. This is the default baud rate for a MicroLogix devices on the DH-485 network. All devices must be at the same baud rate. This rate is stored in the controller Communications Status file (CS0:5/8...CS0:5/15). Configure the baud rate via *Channel Configuration* using RSLogix 500/RSLogix Micro. Select the *Channel 0* tab.

Setting Maximum Node Address

Once you have an established network set up, and are confident that you will not be adding more devices, you may enhance performance by adjusting the maximum node address of your controllers. It should be set to the highest node address being used.

IMPORTANT All devices should be set to the same maximum node address.

MicroLogix 1400 Remote Packet Support

These controllers can respond and initiate with device's communications (or commands) that do not originate on the local DH-485 network. This is useful in installations where communication is needed between the DH-485 and DH+ networks.

DF1 Full-Duplex Protocol

DF1 Full-Duplex protocol provides a point-to-point connection between two devices. DF1 Full-Duplex protocol combines data transparency (American National Standards Institute ANSI - X3.28-1976 specification subcategory D1) and 2-way simultaneous transmission with embedded responses (subcategory F1).

The MicroLogix controllers support the DF1 Full-Duplex protocol via RS-232 connection to external devices, such as computers, or other controllers that support DF1 Full-Duplex.

DF1 is an open protocol. Refer to *DF1 Protocol and Command Set Reference Manual*, Allen-Bradley publication 1770-6.5.16, for more information.

DF1 Full-Duplex Operation

DF1 Full-Duplex protocol (also referred to as DF1 point-to-point protocol) is useful where RS-232 point-to-point communication is required. This type of protocol supports simultaneous transmissions between two devices in both directions. DF1 protocol controls message flow, detects and signals errors, and retries if errors are detected.

When the system driver is DF1 Full-Duplex, the following parameters can be changed:

Parameter	Options	Programming Software Default
Channel	MicroLogix 1400: Channel 0	0
Driver	DF1 Full Duplex	DF1 Full Duplex
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19.2K, 38.4K	19.2K
Parity	none, even	none
Source ID (Node Address)	0…254 decimal	1
Control Line	no handshaking, Full-Duplex modem	no handshaking
Error Detection	CRC, BCC	CRC
Embedded Responses	auto detect, enabled	auto detect
Duplicate Packet (Message) Detect	enabled, disabled	enabled
ACK Timeout (x20 ms)	1…65535 counts (20 ms increments)	50 counts
NAK retries	0…255	3 retries
ENQ retries	0…255	3 retries
Stop Bits	not a setting, always 1	1

DF1 Half-Duplex Protocol

DF1 Half-Duplex protocol provides a multi-drop single master/multiple slave network. In contrast to the DF1 Full-Duplex protocol, communication takes place in one direction at a time. You can use the RS-232 port on the MicroLogix controller as both a Half-Duplex programming port, and a Half-Duplex peer-to-peer messaging port.

MicroLogix 1400controller supports Half-Duplex modems using RTS/CTS hardware handshaking.

DF1 Half-Duplex supports up to 255 devices (addresses 0...254, with address 255 reserved for master broadcasts). *Note: When configuring a message instruction, set the target node address to -1 for broadcast messages.*

Broadcast messages are handled as follows:

DF1 Half-Duplex Master Driver Broadcast Messages

A broadcast write command initiated by the DF1 half-duplex master is received and executed by all DF1 half-duplex slaves. A broadcast write command received by the DF1 half-duplex master after polling a DF1 half-duplex slave is received, acknowledged and re-broadcast without being executed by the DF1 half-duplex master. It is treated like any other slave-to-slave command, except that no acknowledgement is expected after re-broadcast.

DF1 Half-Duplex Slave Driver Broadcast Messages

When a broadcast write command is initiated by a DF1 half-duplex slave, it is queued up just like any other MSG command until it receives a poll from the DF1 half-duplex master. After transmitting the broadcast write command, the DF1 half-duplex slave receives an acknowledgement that the DF1 half-duplex master received the packet without error. When the DF1 half-duplex master re-broadcasts the broadcast write command, the initiating DF1 half-duplex slave receives and executes the command along with all of the other slave nodes receiving the broadcast packet. No acknowledgement or reply is returned.

Choosing a Polling Mode for DF1 Half-Duplex Master

A master station can be configured to communicate with slave stations in either Message-based polling mode or Standard polling mode. The pros and cons of each polling mode are described below.

Message-Based Polling Mode

Message-based polling mode is best used in networks when communication with the slave stations is not time critical and where the user needs to be able to limit when and how often the master station communicates with each slave station. It is ***not*** recommended for larger systems that require time critical communication

between the master and all the slave stations, or for systems where slave station-initiated messages are going to be used.

With Message-Based polling mode, the only time a master station communicates with a slave station is when a message (MSG) instruction in ladder logic is triggered to that particular slave station's address. This polling mode gives the user complete control (through ladder logic) over when and how often to communicate with each slave station.

If multiple MSG instructions are triggered "simultaneously," they will be executed in order, one at a time, to completion (i.e., the first MSG queued up will be transmitted and completed to done or error before the next queued up MSG is transmitted). Any time a message is triggered to a slave station that cannot respond (for instance, if its modem fails), the message will go through retries and time-outs that will slow down the execution of all the other queued up messages. The minimum time to message to every responding slave station increases linearly with the number of slave stations that cannot respond.

If the Message-based selection is "*allow* slaves to initiate messages," a slave station can initiate a message to the master station *(polled report by exception messaging*) or to another slave station *(slave-to-slave messaging)*. The MSG command packet will remain in that slave station's transmit queue until the master station triggers its own MSG command packet to it (which could be seconds, minutes or hours later, depending on the master's ladder logic).

If the Message-based selection is "*don't allow* slaves to initiate messages," then even if a slave station triggers and queues up a MSG instruction in its ladder logic, the master station will not process it.

Standard Polling Mode

Standard polling mode is *strongly* recommended for larger systems that require time critical communication between the master and all the slave stations, or for any system where slave station-initiated messages are going to be used (this includes slave programming over the network, since this uses the same mechanism that slave-to-slave messaging uses). The Active Node Table "automatically" keeps track of which slaves are (and are not) communicating. Standard polling mode should *not* be used in cases where the user needs to be able to limit when and how often the master station communicates with each slave station.

Standard polling mode causes the master station to continuously send one or more 4-byte poll packets to each slave station address configured by the user in the poll list(s) in round robin fashion – as soon as the end of the polling list is reached, the master station immediately goes back and starts polling slave stations from the top of the polling list over again. This is independent and asynchronous to any MSG instructions that might be triggered in the master station ladder logic. In fact, this polling continues even while the master station is in program mode!

When a MSG instruction is triggered while the master station is in run mode, the master station will transmit the message packet just after it finishes polling the current slave station in the poll list and before it starts polling the next slave station in the poll list (no matter where it currently is in the poll list). If multiple MSG instructions have been triggered "simultaneously," at least four message packets may be sent out between two slave station polls. Each of these messages will have an opportunity to complete when the master polls the slave station that was addressed in the message packet as it comes to it in the poll list.

If each of the transmitted message packets is addressed to a different slave station, the order of completion will be based upon which slave station address comes up next in the poll list, not the order in which the MSG instructions were executed and transmitted.

When a slave station receives a poll packet from the master station, if it has one or more message packets queued up to transmit (either replies to a command received earlier or MSG commands triggered locally in ladder logic), the slave station will transmit the first message packet in the transmit queue.

If the standard mode selection is "*single* message per poll scan," then the master station will then go to the next station in the poll list. If the standard mode selection is "*multiple* messages per poll scan," the master station will continue to poll this slave station until its transmit queue is empty.

The master station "knows" the slave station has no message packets queued up to transmit when the slave station responds to the master poll packet with a 2-byte poll response.

Every time a slave station responds or fails to respond to its poll packet, the master station "automatically" updates its Active Node Table (again, even if it's in program mode). In this list, one bit is assigned to each possible slave station address (0...254). If a slave station does not respond when it is polled, its Active Node Table bit is cleared. If it does respond when it is polled, its Active Node Table bit is set. Besides being an excellent online troubleshooting tool, two common uses of the Active Node Table are to report good/bad communication status for all slave stations to an operator interface connected to the master station for monitoring, alarming and logging purposes, and to precondition MSG instructions to each particular slave.

This second use is based on the supposition that if a slave station did not respond the last time it was polled, it may not be able to receive and respond to a MSG instruction now, and so it would most likely process the maximum number of retries and time-outs before completing in error. This slows down both the poll scan and any other messaging going on. Using this technique, the minimum time to message to every responding slave station actually *decreases* as the number of slave stations that can't respond *increases*.

IMPORTANT	In order to remotely monitor and program the slave stations over the half-duplex network while the master station is configured for Standard polling mode, the programming computer DF1 slave driver (typically Rockwell Software RSLinx) station address must be included in the master station poll list.

About Polled Report-by-Exception

Polled report-by-exception lets a slave station initiate data transfer to its master station, freeing the master station from having to constantly read blocks of data from each slave station to determine if any slave input or data changes have occurred. Instead, through user programming, the slave station monitors its own inputs for a change of state or data, which triggers a block of data to be written to the master station when the master station polls the slave.

About Slave-to-Slave Messaging

If one slave station has a message to send to another, it simply includes the destination slave station's address in the message instruction's destination field in place of the master station's address when responding to a poll. The master station checks the destination station address in every packet header it receives from any slave station. If the address does not match the slave's own station address, the entire message is forwarded back onto the telemetry network to the appropriate slave station, without any further processing.

Addressing Tips

Each station on the network, including the master station, must have a unique address. The address range is 0...254, so you can have a maximum of 255 stations on a single telemetry network. Station address 255 is the broadcast address, which you cannot select as a station's individual address.

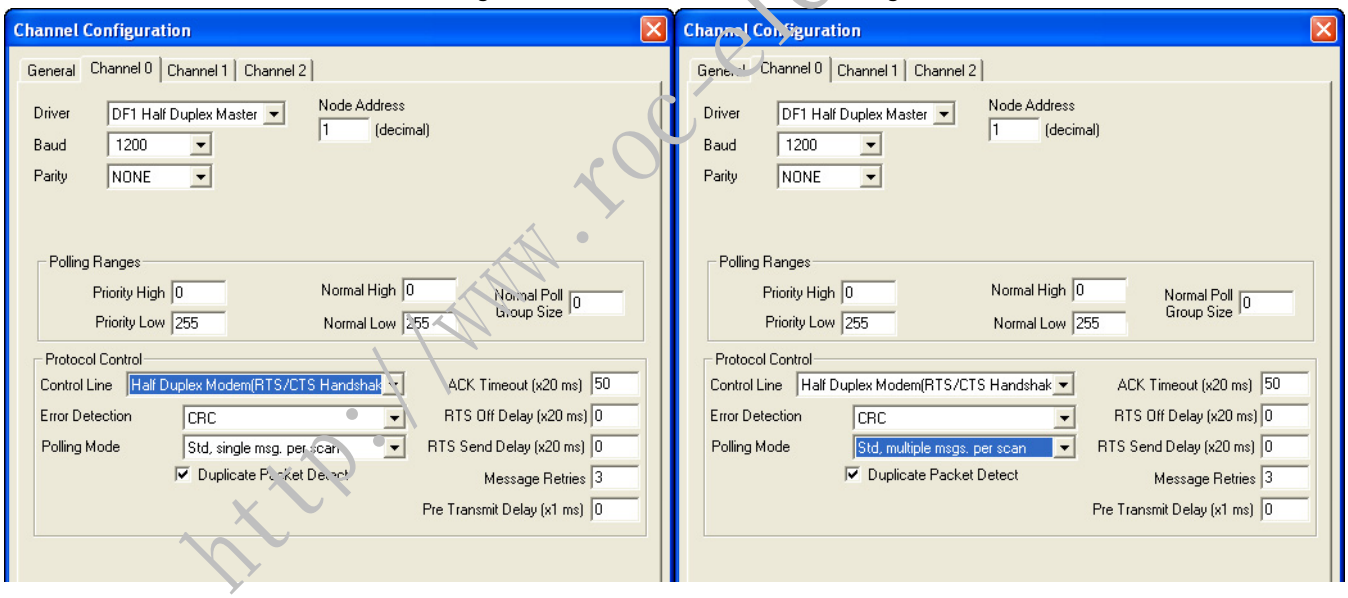
DF1 Half-Duplex Master Standard Polling Mode

With standard polling mode, the master device initiates all communication by polling each slave address configured in the priority and normal polling ranges. The slave device may only transmit message packets when it is polled by the master. Based on a slave's inclusion in the priority and/or normal poll ranges, the master polls each slave on a regular and sequential basis to allow slave devices an opportunity to communicate. During a polling sequence, the master polls a slave either repeatedly until the slave indicates that it has no more message packets to transmit ("standard polling mode, multiple messages per scan") or just one time per polling sequence ("standard polling mode, single message per scan"), depending on how the master is configured.

The polling algorithm polls all of the priority slave addresses each poll scan (priority low to priority high) and a subset of the normal slave address range. The number of normal slave addresses to poll each poll scan is determined by the Normal Poll Group Size configuration parameter. In order to poll all of the slave addresses each poll scan with equal priority, you may define the entire slave address range in either the Priority Poll Range or the Normal Poll Range, and leave the other range disabled. The Polling Range is disabled by defining the low address as 255.

An additional feature of the DF1 Half-Duplex protocol in Standard Polling Mode operation is that it is possible for a slave device to enable a MSG instruction in its ladder program to send or request data to/from the master or another slave. When the initiating slave is polled, the message command is sent to the master. If the message is addressed to the master, then the master replies to the message. If the master recognizes that the message is not intended for it, but for another slave, the master immediately re-broadcasts the message so that it can be received by the intended slave. This slave-to-slave transfer is a built-in function of the master device and can also be used by programming software to upload and download programs to processors on the DF1 Half-Duplex link.

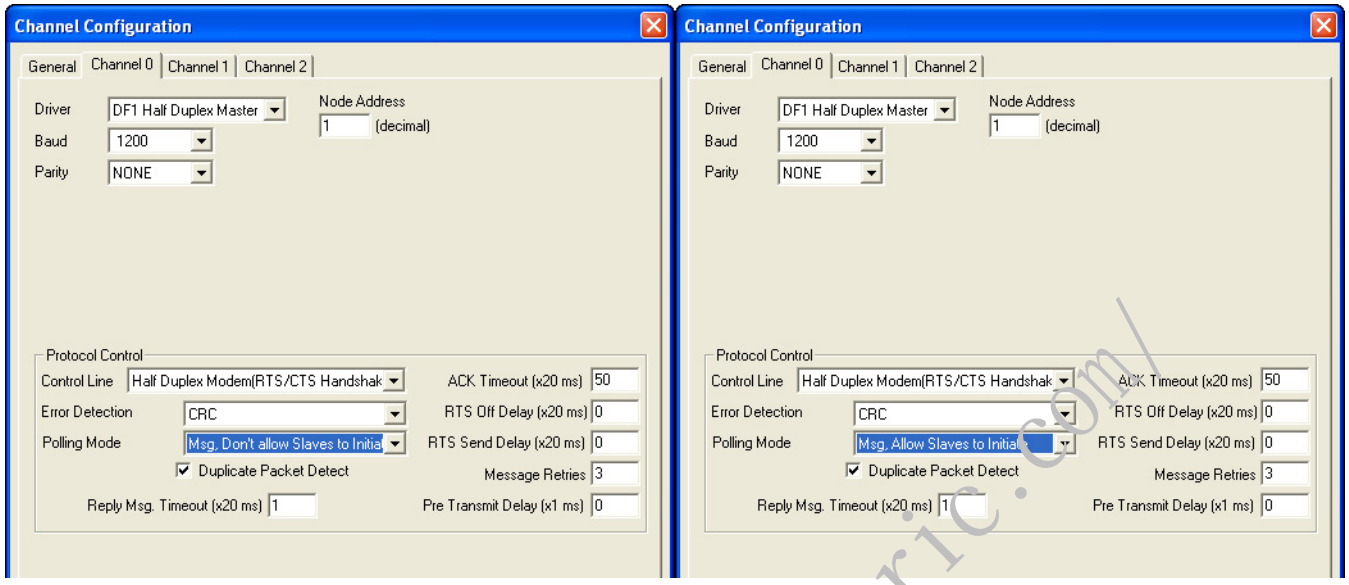
Figure 1 - Standard Mode Channel Configuration



DF1 Half-Duplex Master MSG-based Polling Mode Operation

With MSG-based Polling Mode, the master device only initiates communication with a slave when a MSG instruction to that slave is triggered in ladder logic. Once the read or write command has been transmitted, the master waits the Reply MSG Timeout period and then polls that slave for a reply to its command. The master can be configured either to ignore (“MSG-based Polling, don’t allow slaves to initiate”) or to accept (“MSG-based Polling, allow slaves to initiate”) MSGs that may have been triggered and queued up in the slave.

Figure 2 - Message-Based Polling Mode Channel Configuration



When the system driver is DF1 Half-Duplex Master, the following parameters can be changed:

DF1 Half-Duplex Master Configuration Parameters

Parameter	Options	Programming Software Default
Channel	MicroLogix 1400: Channel 0	0
Driver	DF1 Half Duplex Master	
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19.2K, 38.4K	19.2K
Parity	none, even	none
Node Address	0…254 decimal (255 is reserved for broadcast)	1
Control Line	No Handshaking, Half-Duplex Modem (RTS/CTS Handshaking), Full-Duplex Modem (RTS on) No Handshaking (485 Network)	No Handshaking
Error Detection	CRC, BCC	CRC
Duplicate Packet Detect	enabled, disabled Detects and eliminates duplicate responses to a message. Duplicate packets may be sent under noisy communication conditions if the sender's Message Retries are set greater than 0.	enabled
RTS Off Delay (x20 ms)	0…65535 (can be set in 20 ms increments) – only with control line set to "Half Duplex Modem (RTS/CTS Handshaking)" Specifies the delay time between when the last serial character is sent to the modem and when RTS is deactivated. Gives the modem extra time to transmit the last character of a packet.	0
RTS Send Delay (x20 ms)	0…65535 (can be set in 20 ms increments) – only with control line set to "Half Duplex Modem (RTS/CTS Handshaking)" Specifies the time delay between setting RTS until checking for the CTS response. For use with modems that are not ready to respond with CTS immediately upon receipt of RTS.	0
Message Retries	0…255 Specifies the number of times the master device attempts to re-send a message packet when it does not receive an ACK from the slave device. For use in noisy environments where acknowledgements may become corrupted in transmission.	3

DF1 Half-Duplex Master Configuration Parameters

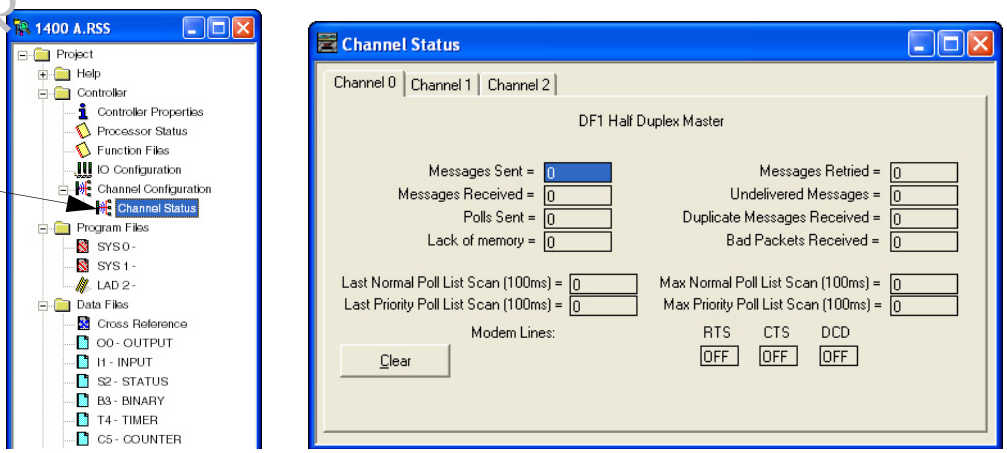
Parameter	Options	Programming Software Default
Pre Transmit Delay (x1 ms)	0…65535 (can be set in 1 ms increments) When the Control Line is set to "No Handshaking", this is the delay time before transmission. Required for 1761-NET-AIC physical Half-Duplex networks. The 1761-NET-AIC needs 2 ms of delay time to change from transmit to receive mode. When the Control Line is set to "Half-Duplex Modem (RTS/CTS Handshaking)", this is the minimum time delay between receiving the last character of a packet and the next RTS assertion.	0
ACK Timeout (x20 ms)	0…255 (can be set in 20 ms increments) Specifies the amount of time the master will wait for an acknowledgement to a message it has transmitted before it retries the message or errors out the message instruction. This timeout value is also used for the poll response timeout.	50
Reply MSG Timeout (x 20 ms)	0…255 (can be set in 20 ms increments) – only with MSG-based Polling Modes Specifies the amount of time the master will wait after receiving an ACK to a master-initiated MSG before polling the slave station for its reply.	1
Priority Polling Range – High	Select the last slave station address to priority poll – only with Standard Polling Modes.	0
Priority Polling Range – Low	Select the first slave station address to priority poll. Entering 255 disables priority polling – only with Standard Polling Modes.	255
Normal Polling Range – High	Select the last slave station address to normal poll – only with Standard Polling Modes.	0
Normal Polling Range – Low	Select the first slave station address to normal poll. Entering 255 disables normal polling – only with Standard Polling Modes.	255
Normal Poll Group Size	Enter the quantity of active stations located in the normal poll range that you want polled during a scan through the normal poll range before returning to the priority poll range. If no stations are configured in the Priority Polling Range, leave this parameter at 0.	0

DF1 Half-Duplex Master Channel Status

Channel Status data is stored in the Communication Status Function File.

Figure 3 - Viewing Channel Status Data for DF1 Half-Duplex Master

Double-click the Channel Status Icon Located beneath the Configuration icon to bring up the Channel Status screen.



Communication Status Function DF1 Half-Duplex Master Channel Status

Status Field	Status File Location[1]	Definition
Messages Sent	CSx:10	The total number of DF1 messages sent by the processor (including message retries)
Messages Received	CSx:11	The number of messages received with no errors
Polls Sent	CSx:15	The number of poll packets sent by the processor
Lack of Memory	CSx:17	The number of times the processor could not receive a message because it did not have available memory
Last Normal Poll List Scan	CSx:19	Time in 100 ms increments of last scan through Normal Poll List
Last Priority Poll List Scan	CSx:21	Time in 100 ms increments of last scan through Priority Poll List
Message Retry	CSx:13	The number of message retries sent by the processor
Undelivered Messages	CSx:12	The number of messages that were sent by the processor but not acknowledged by the destination device
Duplicate Messages Received	CSx:18	The number of times the processor received a message packet identical to the previous message packet
Bad Packets Received	CSx:16	The number of incorrect data packets received by the processor for which no ACK was returned
Max Normal Poll List Scan	CSx:20	Maximum time in 100 ms increments to scan the Normal Poll List
Max Priority Poll List Scan	CSx:22	Maximum time in 100 ms increments to scan the Priority Poll List
RTS (Request to Send)	CSx:9/1	The status of the RTS handshaking line (asserted by the processor)
CTS (Clear to Send)	CSx:9/0	The status of the CTS handshaking line (received by the processor)
DCD (Data Carrier Detect)	CSx:9/3	*Reserved*

(1) x equals the Channel number.

Monitor Active Stations

To see which slave stations are active when the channel is configured for Standard Polling Mode (either single or multiple message per scan), view the DF1 Half-Duplex Master Active Node Table. The table is stored in the Communications Status Function File, words CSx:27...CSx:42, where x is the channel number (x = 0 for MicroLogix 1400). Each bit in the table represents a station on the link, from 0...254, starting with CSx:27/0 for address 0 and CSx:42/14 for address 254. The bit for address 255 (CSx:42/15) is never set, since it is the broadcast address, which never gets polled.

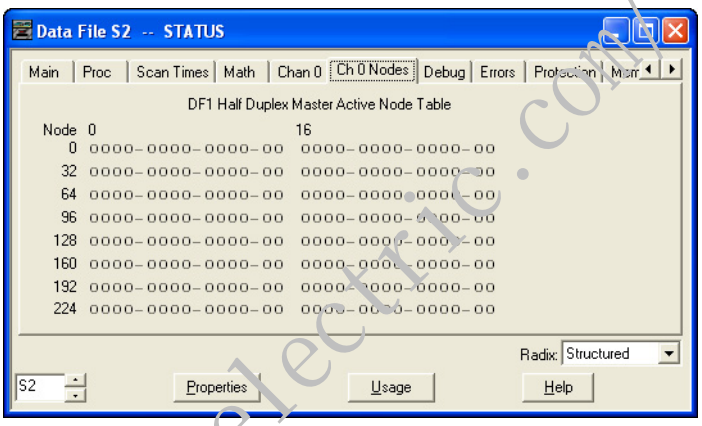
When valid Normal and/or Priority Poll Ranges are defined:

- if a slave responded the last time it was polled by the master, the bit corresponding to its address is set (1 = active).
- if a slave didn't respond the last time it was polled by the master, the bit corresponding to its address is cleared (0 = inactive).

TIP The bit corresponding to the address configured for the DF1 Master is always cleared because the master address never gets polled.

If you are using RSLogix 500/RSLogix Micro version 7.00.00 or higher, you can view the active node table by clicking on "Processor Status" and then selecting the tab for the DF1 Master channel.

Figure 4 - Example Active Node Table



At power-up or after reconfiguration, the master station assumes that all slave stations are inactive. A station is shown active only after it responds to a poll packet.

DF1 Half-Duplex Slave Configuration

When the system driver is DF1 Half-Duplex Slave, the following parameters can be changed:

DF1 Half-Duplex Slave Configuration Parameters

Parameter	Options	Programming Software Default
Channel	MicroLogix 1400: Channel 0	0
Driver	DF1 Half Duplex Slave	
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19.2K, 38.4K	19.2K
Parity	none, even	none
Node Address	0…254 decimal (255 is reserved for broadcast)	1
Control Line	No Handshaking, Half-Duplex Modem (RTS/CTS Handshaking) No Handshaking (485 Network)	No Handshaking
Error Detection	CRC, BCC	CRC

DF1 Half-Duplex Slave Configuration Parameters

Parameter	Options	Programming Software Default
EOT Suppression	enabled, disabled When EOT Suppression is enabled, the slave does not respond when polled if no message is queued. This saves modem transmission power when there is no message to transmit.	disabled
Duplicate Packet (Message) Detect	enabled, disabled Detects and eliminates duplicate responses to a message. Duplicate packets may be sent under noisy communication conditions if the sender's Message Retries are set greater than 0.	enabled
Poll Timeout (x20 ms)	0…65535 (can be set in 20 ms increments) Poll timeout only applies when a slave device initiates a MSG instruction. It is the amount of time that the slave device waits for a poll from the master device. If the slave device does not receive a poll within the Poll Timeout, a MSG instruction error is generated, and the ladder program needs to re-queue the MSG instruction. If you are using a MSG instruction, it is recommended that a Poll Timeout value of zero is not used. Poll Timeout is disabled when set to zero.	3000
RTS Off Delay (x20 ms)	0…65535 (can be set in 20 ms increments) – only with control line set to "Half Duplex Modem (RTS/CTS Handshaking)" Specifies the delay time between when the last serial character is sent to the modem and when RTS is deactivated. Gives the modem extra time to transmit the last character of a packet.	0
RTS Send Delay (x20 ms)	0…65535 (can be set in 20 ms increments) – only with control line set to "Half Duplex Modem (RTS/CTS Handshaking)" Specifies the time delay between setting RTS until checking for the CTS response. For use with modems that are not ready to respond with CTS immediately upon receipt of RTS.	0
Message Retries	0…255 Specifies the number of times the master device attempts to re-send a message packet when it does not receive an ACK from the slave device. For use in noisy environments where acknowledgements may become corrupted in transmission.	3
Pre Transmit Delay (x1 ms)	0…65535 (can be set in 1 ms increments) When the Control Line is set to "No Handshaking", this is the delay time before transmission. Required for 1761-NET-AIC physical Half-Duplex networks. The 1761-NET-AIC needs 2 ms of delay time to change from transmit to receive mode. When the Control Line is set to "Half-Duplex Modem (RTS/CTS Handshaking)", this is the minimum time delay between receiving the last character of a packet and the next RTS assertion.	0

DF1 Radio Modem Protocol

This driver implements a protocol, optimized for use with radio modem networks, that is a hybrid between DF1 Full-Duplex and DF1 Half-Duplex protocols and is not compatible with either protocol.

The primary advantage of using DF1 Radio Modem protocol for radio modem networks is in transmission efficiency. Each read/write transaction (command and reply) requires only one transmission by the initiator (to send the command) and one transmission by the responder (to return the reply). This minimizes the number of times the radios need to "key-up" to transmit, which maximizes radio life and minimizes radio power consumption. It also maximizes communication throughput. In contrast, DF1 Half-Duplex protocol requires five transmissions for the DF1 Master to complete a read/write transaction with a DF1 Slave – three by the master and two by the slave.

IMPORTANT	The DF1 Radio Modem driver should only be used among devices that support and are configured for the DF1 Radio Modem protocol. DF1 Radio Modem protocol is currently supported by SLC 5/03, 5/04 and 5/05 controllers; MicroLogix 1400, 1200 and 1500 controllers; and Logix controllers at Version 16.1 firmware or higher.

Like DF1 Full-Duplex protocol, DF1 Radio Modem allows any node to initiate to any other node at any time (if the radio modem network supports full-duplex data port buffering and radio transmission collision avoidance). Like DF1 Half-Duplex protocol, up to 255 devices are supported, with unique addresses from 0...254. A node ignores any packets received that have a destination address other than its own, with the exception of broadcast packets. A broadcast write command initiated by any DF1 radio modem node is executed by all of the other DF1 radio modem nodes that receive it. No acknowledgement or reply is returned.

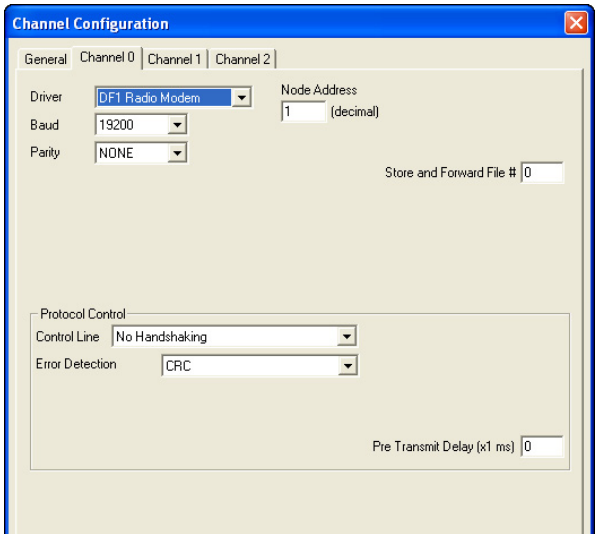
Unlike either DF1 Full-Duplex or DF1 Half-Duplex protocols, DF1 Radio Modem protocol does not include ACKs, NAKs, ENQs, or poll packets. Data integrity is ensured by the CRC checksum.

Using the DF1 Radio Modem

Using RSLogix 500/RSLogix Micro version 7.00.00 or higher, the DF1 Radio Modem driver can be configured as the system mode driver for Channel 0 in MicroLogix 1400.

Channel configuration appears as follows. Figure 5 shows Channel 0 configuration.

Figure 5 - DF1 Radio Modem Channel 0 Configuration (MicroLogix 1400)



When the system driver is DF1 Radio Modem, the following parameters can be changed for Channel 0.

DF1 Radio Modem Channel 0 Configuration Parameters

Parameter	Options	Programming Software Default
Channel	Channel 0	0
Driver	DF1 Radio Modem	
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19.2K, 38.4K	19.2K
Parity	none, even	none
Node Address	0…254 decimal (255 is reserved for broadcast)	1
Store and Forward File Number	Store and Forward allows messages between two out-of-radio-range nodes to be routed through one or more in-radio-range nodes. This is the data table file number used for the Store and Forward Table.	0
Control Line	No Handshaking, Half-Duplex Modem (RTS/CTS Handshaking)	No Handshaking
Error Detection	CRC, BCC	CRC
Pre Transmit Delay (x1 ms)	0…65535 (can be set in 1 ms increments) When the Control Line is set to "No Handshaking", this is the delay time before transmission. Required for 1761-NET-AIC physical Half-Duplex networks. The 1761-NET-AIC needs 2 ms of delay time to change from transmit to receive mode. When the Control Line is set to "Half-Duplex Modem (RTS/CTS Handshaking)", this is the minimum time delay between receiving the last character of a packet and the next RTS assertion.	1

The DF1 Radio Modem driver can be used in a "pseudo" Master/Slave mode with any radio modems, as long as the designated "Master" node is the only node initiating MSG instructions, and as long as only one MSG instruction is triggered at a time.

For modern serial radio modems that support full-duplex data port buffering and radio transmission collision avoidance, the DF1 Radio Modem driver can be used to set up a "Masterless" peer-to-peer radio network, where any node can initiate communications to any other node at any time, as long as all of the nodes are within radio range so that they receive each other's transmissions.

Using Store and Forward Capability

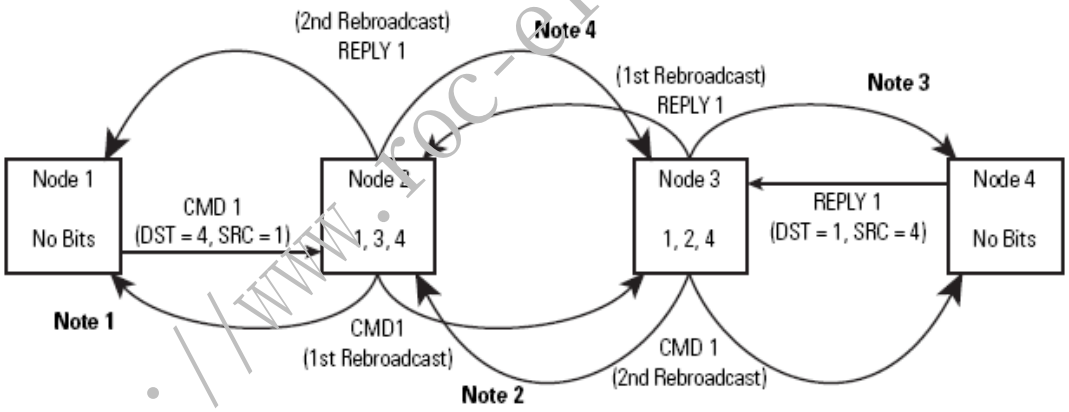
DF1 Radio Modem also supports Store and Forward capability in order to forward packets between nodes that are outside of radio range of each other. Each node that is enabled for Store and Forward has a user-configured Store and Forward Table to indicate which received packets it should re-broadcast, based on the packet's source and destination addresses.

Configuring the Store and Forward Table

The Store and Forward Table can be configured to use any valid binary data table file (B3, B9...B255) of length 16 words. Each bit in the file corresponds to a DF1 Radio Modem node address. In order to configure a MicroLogix to Store and Forward message packets between two other nodes, the bits corresponding to the addresses of those two other nodes must be set. For instance, if node 2 is used to Store and Forward message packets between nodes 1 and 3, then both bits Bx/1 and Bx/3 (where x is the configured data table file number) would have to be set in the Store and Forward Table file (see Figure 7). You can set bit 255 to enable Store and Forward of broadcast packets, as well.

IMPORTANT	Once Store and Forward is enabled, duplicate packet detection is also automatically enabled. Whenever Store and Forward is used within a radio modem network, every node should have a Store and Forward Table file configured, even if all of the bits in the file are cleared, so that duplicate packets will be ignored.

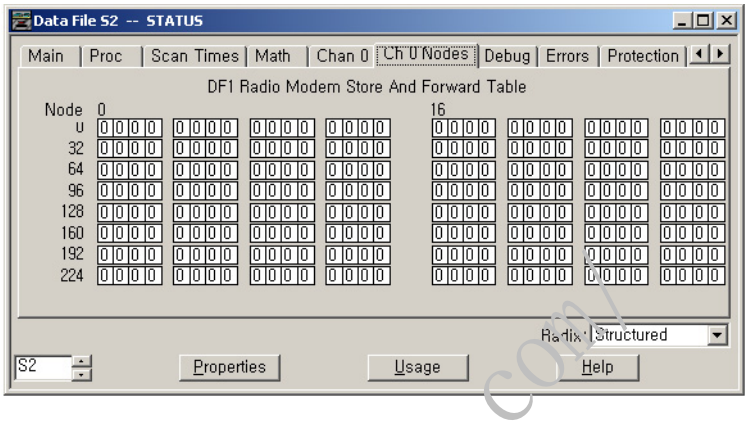
Figure 6 - Applying DF1 Radio Modem Protocol



Note 1 – The link layer of Node 1 blocks the re-transmission of a packet that is received with the SRC byte equal to the receiving node's station address. Packets received that originate from the receiving node should never be re-transmitted.
Note 2 – To prevent Node 2 from re-transmitting a duplicate packet, the link layer of Node 2 updates the duplicate packet table with the last 20 packets received.
Note 3 – The link layer of Node 4 blocks the re-transmission of a packet that is received with the SRC byte equal to the receiving node's station address. Packets received that originate from the receiving node should never be re-transmitted.
Note 4 – To prevent Node 3 from re-transmitting a duplicate packet, the link layer of Node 3 updates the duplicate packet table with the last 20 packets received.

If you are using RSLogix 500/RSLogix Micro version 7.00.00 or higher, you can view the store and forward table by clicking on "Processor Status" and then selecting the tab for the DF1 Master channel.

Figure 7 - Example Store and Forward Table

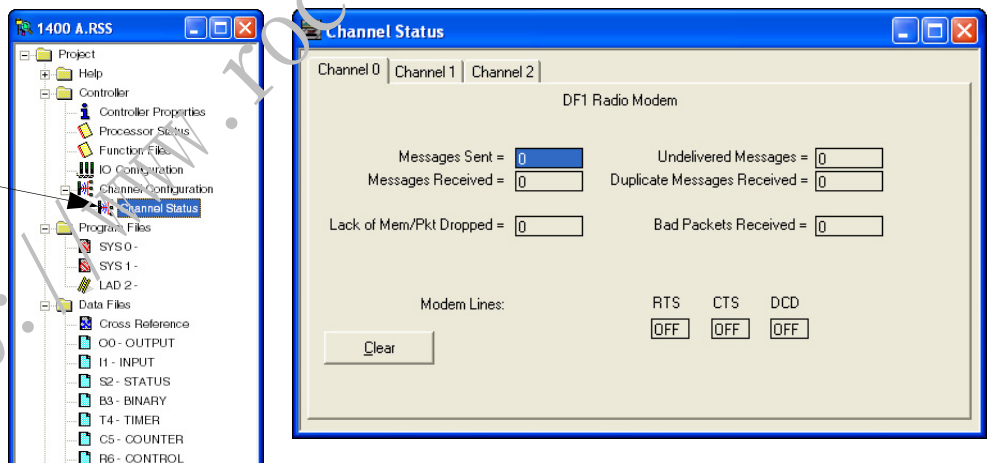


DF1 Radio Modem Channel Status

Channel Status data is stored in the Communication Status Function File.

Figure 8 - Viewing Channel Status for DF1 Radio Modem

Double-click the Channel Status Icon Located beneath the Configuration icon to bring up the Channel Status screen.



Communication Status Function DF1 Radio Modem Channel Status

Status Field	Diagnostic File Location[1]	Definition
Messages Sent	CSx:10	The total number of DF1 messages sent by the processor (including message retries)
Messages Received	CSx:11	The number of messages received with no errors
Lack of Memory	CSx:17	The number of times the processor could not receive a message because it did not have available memory
Undelivered Messages	CSx:12	The number of messages that could not be sent by the processor due to bad modem handshake signals

Communication Status Function DF1 Radio Modem Channel Status

Status Field	Diagnostic File Location[1]	Definition
Duplicate Messages Received	CSx:18	The number of times the processor received a message packet identical to the previous message packet
Bad Packet Received	CSx:16	The number of data packets received by the processor that had bad checksum or were truncated
RTS (Request to Send)	CSx:9/1	The status of the RTS handshaking line (asserted by the processor)
CTS (Clear to Send)	CSx:9/0	The status of the CTS handshaking line (received by the processor)
DCD (Data Carrier Detect)	CSx:9/3	*Reserved*

(1) x equals Channel number

DF1 Radio Modem System Limitations

The following questions need to be answered in order to determine if you can implement the new DF1 Radio Modem driver in your radio modem network:

1. Do all of the devices support DF1 Radio Modem protocol?

   In order to be configured with the DF1 Radio Modem driver, using RSLogix 6.0 or higher, MicroLogix 1200 controllers must be at FRN 7 or higher and MicroLogix 1500 controllers must be at FRN 8 or higher.

   SLC 5/03, 5/04 or 5/05 processors must all be at FRN C/6 or higher in order to be configured with the DF1 Radio Modem driver using RSLogix 500/RSLogix Micro version 5.50 or higher.

   All MicroLogix 1400 controllers support DF1 Radio Modem protocol. RSLogix 500/RSLogix Micro version 7.0 or higher is required to configure the MicroLogix 1400.

   Starting with Version 16.1, all Logix controllers can be configured for DF1 Radio Modem protocol.

2. Does each node receive the radio transmissions of every other node, being both within radio transmission/reception range and on a common receiving frequency (either via a “Simplex” radio mode or via a single, common, full-duplex repeater)?

   If so, then go to question #3 to see if you can use the DF1 Radio Modem driver to set up a peer-to-peer radio network. If not, then you may still be able to use the DF1 Radio Modem driver, by configuring intermediary nodes as Store and Forward nodes.

3. Do the radio modems handle full-duplex data port buffering and radio transmission collision avoidance?

If so, and the answer to #2 is yes as well, then you can take full advantage of the peer-to-peer message initiation capability in every node (i.e., the ladder logic in any node can trigger a MSG instruction to any other node at any time). If not, then you may still be able to use the DF1 Radio Modem driver, but only if you limit MSG instruction initiation to a single "master" node whose transmission can be received by every other node.

4. Can I take advantage of the SLC 5/03, 5/04 and 5/05 channel-to-channel passthru to remotely program the other SLC and MicroLogix nodes using RSLinx and RSLogix 500/RSLogix Micro running on a PC connected to a local SLC processor via DH+ or Ethernet?

   Yes, with certain limitations imposed based on the radio modem network. Refer to the SLC 500 Instruction Set Reference Manual, publication 1747-RM001, for more passthru details and limitations when using the DF1 Radio Modem driver.

Modbus RTU Protocol

This section shows the configuration parameters for Modbus RTU (*Remote Terminal Unit* transmission mode) protocol. For more information about the Modbus RTU protocol, see the Modbus Protocol Specification (available from http://www.modbus.org).

The driver can be configured as Modbus RTU Master or Modbus RTU Slave. The Modbus RTU Slave driver maps the four Modbus data types—coils, contacts, input registers, and holding registers—into four binary and/or integer data table files created by the user.

Modbus RTU Master

Message instructions are used to transfer information between the data files in the Modbus RTU Master and the Modbus RTU Slaves. Refer to Chapter 22 for detailed information about configuring a MSG instruction for Modbus Communications.

Modbus addressing is limited to 16 bits per memory group, each with a range of 1...65,536. There are four memory groups, one for each function:

- coils (generally addressed as **0**xxxx)
- contacts (**1**xxxx)
- input registers (**3**xxxx)
- holding registers (**4**xxxx)

Coils and contacts are addressed at the bit level. Coils are like outputs and can be read and written to. Contacts are like inputs and are read-only. Input registers and holding registers are addressed at the word level. Input registers are generally used for internally storing input values. They are read-only. Holding registers are general purpose and can be both read and written to.

The most significant digit of the address is considered a prefix, and does not get entered into the Modbus Data Address field when configuring the message instruction.

When the message is sent, the address is decremented by 1 and converted into a 4-character hex number to be transmitted via the network (with a range of 0-FFFFh); the slave increments the address by 1, and selects the appropriate memory group based on the Modbus function.

TIP Modbus protocol may not be consistently implemented in the field. The Modbus specification calls for the addressing range to start at 1; however, some devices start addressing at 0.

The Modbus Data Address in the Message Setup Screen may need to be incremented by one to properly access a Modbus slave's memory, depending on that slave's implementation of memory addressing.

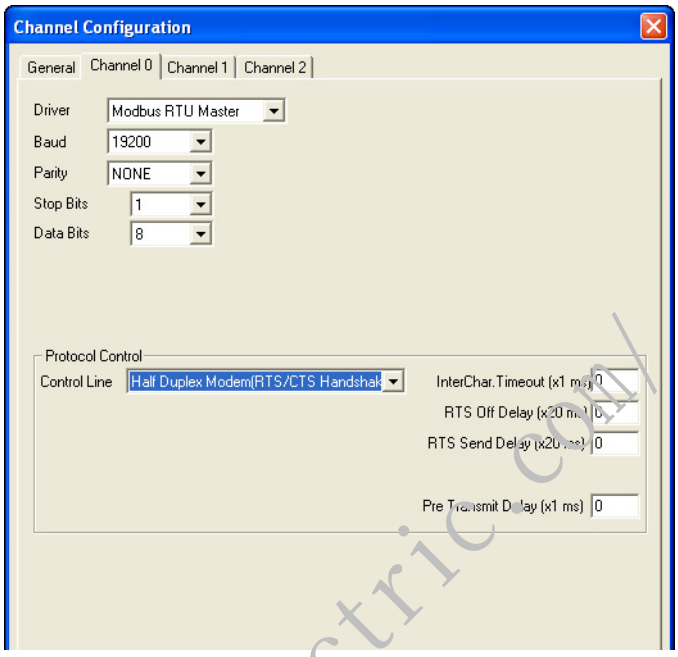
Modbus RTU Slave

The coil and contact files can contain up to 4096 coils or contacts in each register when the data table file is configured for a maximum size of 256 words. Each input register and holding register file can contain up to 256 registers when the data table file is configured for a maximum size of 256 words. With the "Expanded" box checked, the controllers can be specifically configured to use up to six 256-word data table files for a total of 1536 Modbus Holding registers.

TIP A request to access a group of holding registers that span across two files is permitted. Note that the maximum number of registers in a command does not allow for more than two files to be accessed during a single Modbus command.

Modbus RTU Master Configuration

Select the Modbus RTU Master from the Channel Configuration menu as shown below.



The Baud defaults to 19200.

The Control Line can be configured as:

- No Handshaking
- Full-Duplex Modem (RTS on)
- Half Duplex Modem (RTS/CTS handshaking).
- No Handshaking (485 Network)

TIP In order to connect directly to an RS-485 Modbus network, use a 1763-NC01 cable and configure the Control Line setting for No Handshaking (485 network).

The Protocol Control defaults are:

- No Handshaking
- InterChar. Timeout = 0
- Pre Transmit Delay = 0.

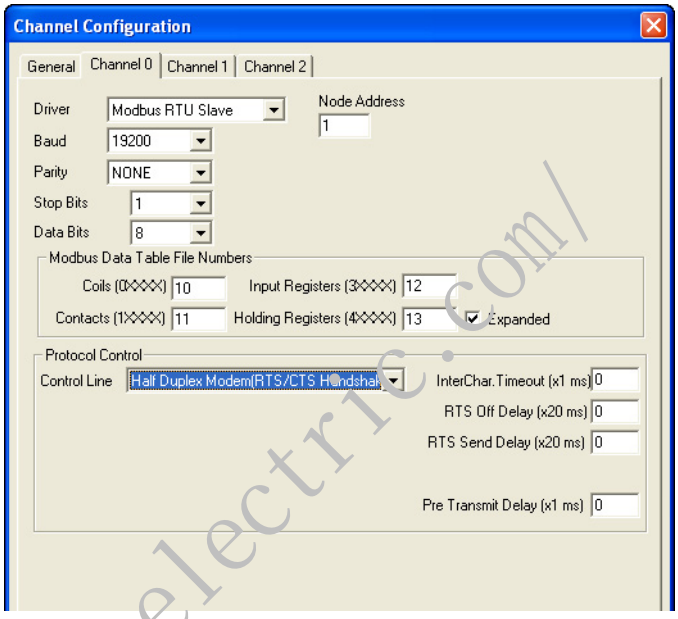
When the system driver is Modbus RTU Master, the following communication port parameters can be changed:

Port Parameters for the Modbus RTU Master

Parameter	Options	Programming Software Default
Channel	Channel 0	0
Driver	Modbus RTU Master	
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19.2K, 38.4K	19.2K
Parity	none, even, odd	none
Control Line	No Handshaking, Full-Duplex Modem (RTS on), Half-Duplex Modem (RTS/CTS handshaking), No Handshaking (485 Network)	No Handshaking
Inter-character Timeout (x1 ms)	0…65535 (can be set in 1 ms increments); 0 = 3.5 character times Specifies the minimum delay between characters that indicates the end of a message packet.	0
RTS Off Delay (x20 ms)	0…65535 (can be set in 20 ms increments) Specifies the delay time between when the last serial character is sent to the modem and when RTS is deactivated. Gives the modem extra time to transmit the last character of a packet.	0
RTS Send Delay (x20 ms)	0…65535 (can be set in 20 ms increments) Specifies the time delay between setting RTS until checking for the CTS response. For use with modems that are not ready to respond with CTS immediately upon receipt of RTS.	0
Pre Transmit Delay (x1 ms)	0…65535 (can be set in 1 ms increments) When the Control Line is set to *No Handshaking*, this is the delay time before transmission. Required for 1761-NET-AIC physical Half-Duplex networks. The 1761-NET-AIC needs 2 ms of delay time to change from receive to transmit mode. When the Control Line is set to *Half-Duplex Modem or Full-Duplex Modem*, this is the minimum time delay between receiving the last character of a packet and the RTS assertion.	0
Stop Bits	1,5,2	1
Data Bits	7	8

Modbus RTU Slave Configuration

The Modbus configuration screen and configuration procedure are shown below:



1. To set up Channel 0 and data files for Modbus communication, select the Channel 0 Configuration tab.
2. Choose "Modbus RTU Slave" driver and assign driver characteristics.
3. Enter Modbus Data Table File Numbers. Select the Expansion check box to utilize multiple holding register data files.

   TIP The controller default is one data file of 256 registers. The Expansion check box enables an additional five files and 1280 holding registers.

   The five additional tables do not need to be individually defined, but sequentially follow the first integer or bit file. For example, if the first file is N10 (or B10), then the additional five files will be N11 (or B11), N12 (or B12), N13 (or B13), N14 (or B14), and N15 (or B15).

4. Enter the data table size and type for each required file. The data table file(s) (not including the five additional tables if Expanded is checked) will be created automatically.

When the system driver is Modbus RTU Slave, the following communication port parameters can be changed:

Port Parameters for the Modbus RTU Slave

Parameter	Options	Programming Software Default
Channel	Channel 0	0
Driver	Modbus RTU Slave	
Baud Rate	300, 600, 1200, 2400, 4800, 9600, 19.2K, 38.4K	19.2K
Parity	none, even, odd	none
Node Address	1…247 decimal	1
Control Line	No Handshaking, Half-Duplex Modem (RTS/CTS Handshaking), No Handshaking (485 Network)	No Handshaking
Inter-character Timeout (x1 ms)	0…6553 (can be set in 1 ms increments); 0 = 3.5 character times Specifies the minimum delay between characters that indicates the end of a message packet.	0
Modbus Data Table File Number Assignment (Must be Binary or Integer file type)	Coils (Discrete outputs, Modbus addresses 0001…4096) range = 3…255, 0 = no file	0
	Contacts (Discrete inputs, Modbus addresses 10001…14096) range = 3…255, 0 = no file	0
	Input Registers (Read Only, Modbus addresses 30001…30256) range = 3…255, 0 = no file	0
	Holding Registers (Read/Write, Modbus addresses 40001…40256) range = 3…255, 0 = no file	0
RTS Off Delay (x20 ms)	0…65535 (can be set in 20 ms increments) Specifies the delay time between when the last serial character is sent to the modem and when RTS is deactivated. Gives the modem extra time to transmit the last character of a packet.	0
RTS Send Delay (x20 ms)	0…65535 (can be set in 20 ms increments) Specifies the time delay between setting RTS until checking for the CTS response. For use with modems that are not ready to respond with CTS immediately upon receipt of RTS.	0
Pre Transmit Delay (x1 ms)	0…65535 (can be set in 1 ms increments) When the Control Line is set to *No Handshaking*, this is the delay time before transmission. Required for 1761-NET-AIC physical Half-Duplex networks. The 1761-NET-AIC needs 2 ms of delay time to change from receive to transmit mode. When the Control Line is set to *Half-Duplex Modem*, this is the minimum time delay between receiving the last character of a packet and the RTS assertion.	0
Stop Bits	1,5,2	1
Data Bits	7	8

Modbus Slave Memory Map

The modbus Memory map is summarized in and detailed in :

Modbus to MicroLogix Memory Map - Summary

Modbus Addressing	Description	Valid MicroLogix Addressing		
		File Type	Data File Number	Address
0001…4096	Read/Write Modbus Coil Data space	Bit (B) or Integer (N)	3…255	bits 0…4095
10001…14096	Read-Only Modbus Contact Data space	Bit (B) or Integer (N)	3…255	bits 0…4095
30001…30256	Read-Only Modbus Input Register space	Bit (B) or Integer (N)	3…255	words 0…255
30501…30532	Modbus Communication Parameters	Communication Status File	-	-
31501…31566	Read-Only System Status File space	Status (S)	2	words 0…65
40001…40256	Read/Write Modbus Holding Register space	Bit (B) or Integer (N)	3…255	words 0…255
40257…41280[1]	Read/Write Modbus Holding Register space	Bit (B) or Integer (N)	3…255	words 0…255 of four Holding Register files
41501…41566	Read/Write System Status File space	Status (S)	2	words 0…65
41793…42048[1]	Read/Write Modbus Holding Register space	Bit (B) or Integer (N)	3…255	words 0…255 of the last Holding Register file

(1) These addresses only become active when specially configured for expanded holding registers.

Modbus Slave to MicroLogix Memory Map - Detail

Modbus Addressing	Modbus Address Reference	Modbus Function Code (decimal)
0001…4096	Read/Write Modbus Coil Data space	1, 5, 15
10001…14096	Read Only Modbus Contact Data space	2
30001…30256	Read Modbus Input Register space	4
30501	Modbus Data Table Coil File Number	4
30502	Modbus Data Table Contact File Number	4
30503	Modbus Data Table Input Register File Number	4
30504	Modbus Data Table Holding Register File Number	4
30506	Pre-Send Delay	4
30507	Modbus Slave Address	4
30508	Inter-character Timeout	4
30509	RTS Send Delay	4
30510	RTS Off Delay	4
30511	Parity	4
30512	Presentation Layer Error Code	4
30512	Presentation Layer Error Code	4
30513	Presentation Layer Error Count	4
30514	Executed Function Code Error	4
30515	Last Transmitted Exception Code	4

Modbus Slave to MicroLogix Memory Map - Detail

Modbus Addressing	Modbus Address Reference	Modbus Function Code (decimal)
30516	File Number of Error Request	4
30517	Element Number of Error Request	4
30518	Function Code 1 Message Counter - Read Single Output Coil	4
30519	Function Code 2 Message Counter - Read Discrete Input Image	4
30520	Function Code 3 Message Counter - Read Single Holding Register	4
30521	Function Code 4 Message Counter - Read Single Input Register	4
30522	Function Code 5 Message Counter - Set/Clear Single Output Coil	4
30523	Function Code 6 Message Counter - Read/Write Single Holding Register	4
30524	Function Code 8 Message Counter - Run Diagnostics	4
30525	Function Code 15 Message Counter - Set/Clear for Block of Output Coils	4
30526	Function Code 16 Message Counter - Read/Write for Block of Holding Registers	4
30527	Modem Status	4
30528	Total messages responded to by this slave	4
30529	Total messages to this Slave	4
30530	Total Messages Seen	4
30531	Link Layer Error Count	4
30532	Link Layer Error	4
31501…31566	Read Only System Status File	4
40001…40256	Read/Write Modbus Holding Register space (1st Holding Register file).	3, 6, 16
40257…40512	Read/Write Modbus Holding Register space (2nd Holding Register file).	3, 6, 16
40513…40768	Read/Write Modbus Holding Register space (3rd Holding Register file).	3, 6, 16
40769…41024	Read/Write Modbus Holding Register space (4th Holding Register file).	3, 6, 16
41025…41280	Read/Write Modbus Holding Register space (5th Holding Register file).	3, 6, 16
41501…41566	Read/Write System Status File	3, 6, 16
41793…42048	Read/Write Modbus Holding Register space (6th Holding Register file).	3, 6, 16

Modbus Commands

The controller configured for Modbus RTU Slave responds to the Modbus command function codes listed in below:

Supported Modbus Commands as a Modbus RTU Slave

Command	Function Code (decimal)	Subfunction Code (decimal)
Read Coil Status	1	-
Read Input Status	2	-
Read Holding Registers	3	-
Read Input Registers	4	-

Supported Modbus Commands as a Modbus RTU Slave

Command	Function Code (decimal)	Subfunction Code (decimal)
Write Single Coil(1)	5	-
Write Single Holding Register(1)	6	-
Echo Command Data	8	0
Clear Diagnostic Counters	8	10
Write Multiple Coils(1)	15	-
Write Multiple Holding Registers(1)	16	-

(1) Broadcast is supported for this command.

Supported Modbus Commands as a Modbus RTU Master

Command	Function Code (decimal)	Subfunction Code (decimal)
Read Coil Status	1	-
Read Input Status	2	-
Read Holding Registers	3	-
Read Input Registers	4	-
Write Single Coil(1)	5	-
Write Single Holding Register(1)	6	-
Write Multiple Coils(1)	15	-
Write Multiple Holding Registers(1)	16	-

(1) Broadcast is supported for this command.

Modbus Error Codes

Upon receiving a Modbus command that is not supported or improperly formatted, the controller configured for Modbus RTU Slave will respond with one of the exception codes listed in below:

Modbus Error Codes Returned by Modbus RTU Slave

Error Code	Error	Description	Transmitted Exception Code(2)
0	No error.		none
1	Function Code cannot Broadcast.	The function does not support Broadcast.	nothing transmitted
2	Function Code not supported.	The controller does not support this Modbus function or subfunction.	1
3	Bad Command Length.	The Modbus Command is the wrong size.	3
4	Bad Length.	The function attempted to read/write past the end of a data file.	3
5	Bad parameter	The function cannot be executed with these parameters.	1
6	Bad File Type	The file number being referenced is not the proper file type.	2
7	Bad File Number	The file number does not exist.	2
8	Bad Modbus Address	The function attempted to access an invalid Modbus address.(1)	3
9	Table Write protected	The function attempted to write to a read-only file.	3
10	File Access Denied	Access to this file is not granted.	2
11	File Already Owned	Data file is already owned by another process.	2

(1) See on page 561 for valid Modbus memory mapping.

(2) If Modbus Command is sent with a valid Broadcast address, then no exception reply will be sent for Error Codes 2…11.

The following table lists the possible error codes and error descriptions for the Modbus RTU Master MSG Instruction.

Modbus Error Codes in Modbus RTU Master MSG Instruction

Error Code	Error	Description	Received Exception Code
81	Illegal Function	The function code sent by the Master is not supported by the slave or has an incorrect parameter.	1
82	Illegal Data Address	The data address referenced in the Master command does not exist in the slave, or access to that address is not allowed.	2
83	Illegal Data Value	The data value being written is not allowed, either because it is out of range, or it is being written to a read-only address.	3
84	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action.	4
85	Acknowledge	The slave has accepted the request, but a long duration of time will be required to process the request.	5
86	Slave Device Busy	The slave is currently processing a long-duration command.	6

Modbus Error Codes in Modbus RTU Master MSG Instruction

Error Code	Error	Description	Received Exception Code
87	Negative Acknowledge	The slave cannot perform the program function received in the command.	7
88	Memory Parity Error	The slave attempted to read extended memory, but detected a parity error in the memory.	8
89	Non-standard Error Code	An error code greater than 8 was returned by the slave.	>8

When Channel 0 or Channel 2 is configured for Modbus RTU Master or Modbus RTU Slave, the associated Channel Status screen displays a Link Layer Error Count and a Link Layer Error Code. Use the table below to interpret the Link Layer Error Code being displayed.

Modbus RTU Link Layer Error Codes

Error Code	Description
0	No error
1	No receive buffer available for reply
2	Too short message received
3	Too long message received
4	UART error during reply reception
5	Bad CRC in reply packet
6	CTS one second timeout prior to transmission
7	CTS dropped in mid-packet transmission
9	Packet receive from unknown slave or bad slave
10	Function code mismatch
11	Function code not supported
13	Reply timeout

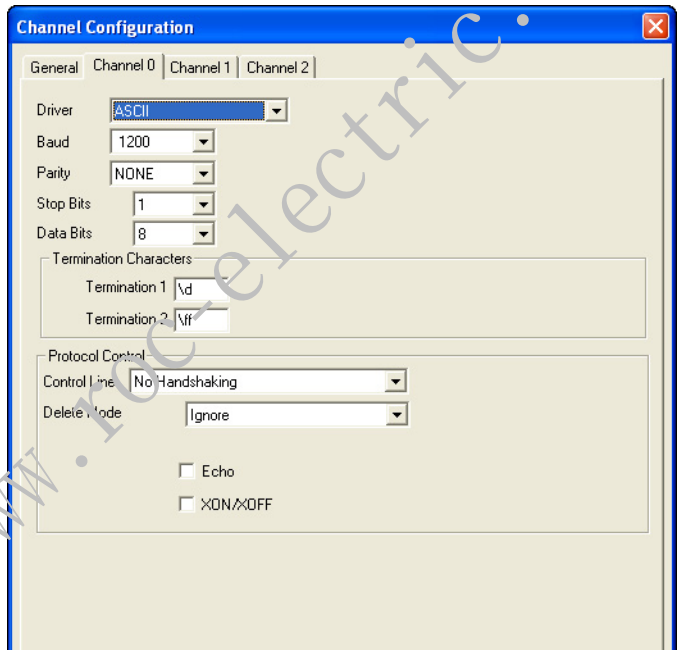
ASCII Driver

The ASCII driver provides connection to other ASCII devices, such as bar code readers, weigh scales, serial printers, and other intelligent devices.

You can use ASCII by configuring the RS-232 port, channel 0 for ASCII driver. When configured for ASCII, all received data is placed in a buffer. To access the data, use the ASCII instructions in your ladder program. See ASCII Instructions on page 309 for information on using the ASCII instructions. You can also send ASCII string data to most attached devices that accept ASCII data/characters.

TIP Only ASCII instructions can be used when a channel is configured for ASCII. If you use a Message (MSG) instruction that references the channel, an error occurs.

The channel configuration screen is shown below:



The controller updates changes to the channel configuration at the next execution of a Service Communications (SVC) instruction, I/O Refresh (REF) instruction, or when it performs Communications Servicing, whichever comes first.

When the driver is set to ASCII, the following parameters can be changed:

ASCII Channel Configuration Parameters

Parameter	Description	Programming Software Default
Channel	Channel 0	0
Driver	ASCII	
Baud Rate	Toggles between the communication rate of 300, 600, 1200, 2400, 4800, 9600, 19.2K, and 38.4K.	1200
Parity	Toggles between None, Odd, and Even.	None
Termination 1	Specifies the first termination character. The termination character defines the one or two character sequence used to specify the end of an ASCII line received. Setting the first ASCII termination character to undefined (\ff) indicates no ASCII receiver line termination is used.	\d
Termination 2	Specifies the second termination character. The termination character defines the one or two character sequence used to specify the end of an ASCII line received. Setting the second ASCII Termination character to undefined (\ff) and the first ASCII Termination character to a defined value (\d) indicates a single character termination sequence.	\ff
Control Line	Toggles between No Handshaking, Full-Duplex Modem (RTS On), Half-Duplex Modem (RTS/CTS handshaking), and No Handshaking (485 Network)	No Handshaking
Delete Mode	The Delete Mode allows you to select the mode of the "delete" character. Toggles between Ignore, CRT, and Printer. Delete Mode affects the characters echoed back to the remote device. When Delete Mode is enabled, the previous character is removed from the receive buffer. •In CRT mode, when a delete character is encountered, the controller echos three characters to the device: backspace, space, and backspace. This erases the previous character on the terminal. •In Printer Mode, when a delete character is encountered, the controller echos the slash character, then the deleted character. Enable the Echo parameter to use Delete Mode.	Ignore
Echo	When Echo Mode is enabled, all of the characters received are echoed back to the remote device. This allows you to view characters on a terminal connected to the controller. Toggles between Enabled and Disabled.	Disabled
XON/XOFF	Allows you to Enable or Disable XON/XOFF software handshaking. XON/XOFF software handshaking involves the XON and XOFF control characters in the ASCII character set. When the receiver receives the XOFF character, the transmitter stops transmitting until the receiver receives the XON character. If the receiver does not receive an XON character after 60 seconds, the transmitter automatically resumes sending characters. Also, when the receive buffer is more than 80% full, an XOFF character is sent to the remote device to pause the transmission. Then, when the receive buffer drops to less than 80% full, an XON character is sent to the remote device to resume the transmission.	Disabled
RTS Off Delay (x20 ms)	Allows you to select the delay between when a transmission is ended and when RTS is dropped. Specify the RTS Off Delay value in increments of 20 ms. Valid range is 0…65535.	0
RTS Send Delay (x20 ms)	Allows you to select the delay between when RTS is raised and the transmission is initiated. Specify the RTS Send Delay value in increments of 20 ms. Valid range is 0…65535.	0
Stop Bits	1,5,2	1
Data Bits	7	8

Ethernet Driver

The MicroLogix1400 supports Ethernet communication via the Ethernet communication channel 1. Ethernet is a local area network that provides communication between a variety of network devices at 10/100 Mbps.

TCP/IP is the mechanism used to transport Ethernet messages. The MicroLogix1400 processor uses TCP/IP to establish sessions and to send MSG commands. Connections can be initiated by either a client program (INTERCHANGE or RSLinx application) or a processor. Refer to Communications Instructions on page 337 for the MSG instruction operation to see how connections are established using the MSG instruction. The MicroLogix1400 Ethernet connector conforms to ISO/IEC 8802-3 STD 802.3 and utilizes 10/100Base-T media. Connections are made directly from the MicroLogix1400 to an Ethernet hub or switch. The network setup is simple and cost-effective.

There are two ways to configure the MicroLogix1400 Ethernet channel 1. The configuration can be done via a BOOTP or DHCP request at processor power up, or by manually setting the configuration parameters using RSLogix 500/ RSLogix Micro Programming Software (refer to Example 5 - Configuring an Ethernet/IP Message on page 373).

Ethernet Configuration Parameters

Parameter	Options	Programming Software Default
Driver	Ethernet	Ethernet
Hardware Address	The processor's Ethernet hardware address. This value cannot be changed.	Ethernet Hardware Address
IP Address	1…254 (zero and 255 are reserved for broadcast purposes.) The processor's internet address. You must specify the IP address to enable the processor to connect to the TCP/IP network. You can specify the address manually, or enable BOOTP or DHCP (located in the Protocol Control section of this dialog) to provide the address.	0.0.0.0
Subnet Mask	0…255 in each field. Used by the processor to interpret IP addresses when the Internet is divided into subnets. The subnet mask must be specified. You can do this either manually or by enabling BOOTP or DHCP. The processor compares and screens addresses using the mask to identify its own address to see if it should listen to corresponding messages. The comparison occurs in binary. Any address position for which the mask is set to a binary 1 will be compared, any address position for which the mask is set to a binary 0 will be ignored. For example, if the mask is 255.255.255.0 the processor will listen to all addresses whose first three segments match its own address regardless of the value in the last segment. (255 in decimal equals to 1111 1111 in binary.)	0.0.0.0
Gateway Address	1…254 in each field. The IP address of the gateway that provides a connection to another IP network. This field is required when you communicate with other network devices, not on a local subnet.	0.0.0.0
Default Domain Name	The default domain name can have the following formats: 'a.b.c', 'a.b' or 'a', where a, b, c must start with a letter, end with a letter or digit, and have as interior characters only letters, digits or hyphens. Maximum length is 63 characters.	NULL
Primary Name Server	This is the IP address of the computer acting as the local Ethernet network Primary Domain Name System (DNS) server.	0.0.0.0
Secondary Name Server	This is the IP address of the computer acting as the local Ethernet network Secondary Domain Name System (DNS) server.	0.0.0.0
Network Link ID	0…199. The Link ID assigned to the MicroLogix 1400 either by an RSLinx OPC topic or by the routing table in a 1756-DHRIO or 1756-DH485 module.	0

Ethernet Configuration Parameters

Parameter	Options	Programming Software Default
Bootp Enable	enabled, disabled Check this box to enable Bootp. If enabled, this causes the processor at power up to try to obtain its network-related parameters (IP address, subnet mask, gateway address, etc.) via BOOTP request. You can not manually change the IP address if BOOTP is enabled. If you disable BOOTP make sure that you have an IP address specified. If you change this field from enabled to disabled, the change will take effect only when the system is restarted. If BOOTP is enabled, DHCP will be automatically disabled. Note: If BOOTP is enabled, you must have the BOOTP server running at all times because the processor requests its address to the BOOTP server at any time during its power up.	1 (enabled)
DHCP Enable	enabled, disabled. DHCP (Dynamic Host Configuration Protocol) automatically assigns IP addresses to client stations logging on to a TCP/IP network. There is no need to manually assign permanent IP parameters. DHCP is only available when BOOTP is disabled.	0 (disabled)
SNMP Server Enable	enabled, disabled. Check this box to enable SNMP (Simple Network Management Protocol). Disable SNMP to guarantee better security since it prevents anyone from obtaining information about the processor or network using SNMP. Any change to this function does not take effect until the system is restarted. This function can be changed through online modification of the channel configuration or through offline modification followed by downloading it to the processor. Once changed, the function will be operational in the processor after the system is restarted.	1 (enabled)
SMTP Client Enable	The SMTP Client service enable switch. When SMTP is enabled, MicroLogix 1400 is capable of transmitting e-mail messages generated by a 485CIF write message with a string element. There must be a SMTP server on the network capable of processing e-mail service. This provides an extremely versatile mechanism to report alarms, status, and other data-related functions.	0 (disabled)
HTTP Server Enable	enabled, disabled. Check this box to enable HTTP (Hyper-Text Transfer Protocol). Disable HTTP to guarantee better security since it prevents access to the processor using a web browser. Note that disabling HTTP will prevent you from viewing the extended diagnostics available through a web browser. Any change to this function does not take effect until the system is restarted. This function can be changed through online modification of the channel configuration or through offline modification followed by downloading it to the processor. Once changed, the function will be operational in the processor after the system is restarted.	1 (enabled)
Auto Negotiate	enabled, disabled. Check this box to enable Auto Negotiation. Auto Negotiation allows the processor to negotiate with switches, routers, and modems for optimal performance. When Auto Negotiation is enabled, the port speed selections will list the available options. When two settings are shown for port speed, Auto Negotiation will choose the optimal setting. When one setting is shown, that setting will be used if possible. If the attached device does not support 100 Mbps full duplex, then the default setting will be 10 Mbps half duplex.	1 (enabled)
DNP3 over IP Enable	enabled, disabled. Check this box to enable DNP3 over IP subsystem. Any change to this function does not take effect until the system is restarted.This function can be changed through online modification of the channel configuration or through offline modification followed by downloading it to the processor. Once changed, the function will be operational in the processor after the system is restarted. You cannot enable both DNP3 over IP and Modbus TCP. This field is only available for MicroLogix 1400 OS Series B controllers	0 (disabled)
Modbus TCP Enable	enabled, disabled. Check this function to enable Modbus TCP client and server subsyste. Any change to this function does not take effect until the system is restarted. This function can be changed through online modification of the channel configuration or through offline modification followed by downloading it to the processor. Once changed, the function will be operational in the processor after the system is restarted. You cannot enable both DNP3 over IP and Modbus TCP. This field is only available for MicroLogix 1400 OS Series B controllers	0 (disabled)

Ethernet Configuration Parameters

Parameter	Options	Programming Software Default
Disable Ethernet/IP Incoming Connections	enabled, disabled. Select this function to disable the Ethernet/IP inbound subsystem to avoid a security issue when using DNP3 over IP. When this box is selected, the MicroLogix 1400 controller will not allow any inbound connections (MSG instruction uses outbound connections and will work.). This function can be changed through online modification of the channel configuration or through offline modification followed by downloading it to the processor. Once changed, the function will be operational in the processor after the system is restarted. This field is only available for MicroLogix 1400 OS Series B controllers.	0 (enabled)
Disable Duplicate IP Address Detection	enabled, disabled. By default the MicroLogix 1400 controller broadcasts probe packets to detect if there is another device on the network with the same IP address. Few packets are sent, but this can cause high traffic if many MicroLogix 1400 controllers are on the network or if the network is slow (such as when using Ethernet radio modems). Select this function to disable duplicate IP address detection. This field is only available for MicroLogix 1400 OS Series B controllers.	0 (enabled)
Port Setting	Auto Negotiate is enabled 10/100Mbps Full Duplex/Half Duplex, 100 Mbps Full or 100 Mbps Half Duplex, 100 Mbps Full Duplex or 10 Mbps Full Duplex, 100 Mbps Half Duplex or 10 Mbps Full Duplex, 100 Mbps Full Duplex, 100 Mbps Half Duplex, 10 Mbps Full Duplex, 10 Mbps Half Duplex Only Auto Negotiate is disabled 100 Mbps Full Duplex Forced, 100 Mbps Half Duplex Forced, 10 Mbps Full Duplex Forced, 10 Mbps Half Duplex Forced Select the port setting from the drop down list. The selections will vary depending on whether you are online or offline, and whether Auto Negotiation is enabled or disabled.	10/100Mbps Full Duplex/Half Duplex
Msg Connection Timeout (x 1ms)	250…65,500 ms. The number of milliseconds allowed for a MSG instruction to establish a connection with the destination node.	15000
Msg Reply Timeout (x ms)	250…65,500 ms. The number of milliseconds the Ethernet interface waits for a reply from a command it initiated (through a MSG instruction).	3000
Inactivity Timeout	The amount of time (in minutes) that a MSG connection may remain inactive before it is terminated. The Inactivity Timeout has a 1 minute resolution and a range from 1…65,500 minutes.	30 minutes.
Contact	The Contact string which is specified by the SNMP client. The maximum length is 63 characters.	Null
Location	The Location string which is specified by the SNMP client. The maximum length is 63 characters.	Null

SNMP MIB II Data Groups

Simple Network Management Protocol (SNMP) specifies the diagnostic data that a host computer must maintain for a network management software to access. Hosts typically keep:

- statistics on the status of their network interfaces.
- incoming and outgoing traffic.
- dropped datagrams.
- error messages.

Network management protocols let network management software access these statistics.

Management Information Base II is the SNMP standard for the management of network data. The following table lists the MIB II data items and their descriptions available from the MicroLogix 1400 controller when SNMP Server is enabled within the Channel 1 configuration. (MIBs sysContact and sysLocation are read/write – all other MIBs are read-only.)

MicroLogix 1400 supports SNMP version SNMPv2c.

Note: We recommend that you disable the SNMP server unless you need the functionality.

MIB Data and Descriptions

Group	MIB	Description
System	sysDescr	Description of device
	sysObjectID	Identity of agent software
	sysUpTime	How long ago the agent started
	sysContact	Device contact information
	sysName	Device name
	sysLocation	Device location
	sysServices	A value which indicates the set of services that this entity may potentially offer
Interfaces	ifNumber	The number of network interfaces (regardless of their current state) present on this system
	ifIndex	Interface number
	ifDescr	Description of the interface

MIB Data and Descriptions

Group	MIB	Description
	ifType	Type of interface
	ifMtu	MTU size
	ifSpeed	Transmission rate in bits/second
	ifPhysAddress	Media specific address
	ifAdminStatus	Desired interface state
	ifOperStatus	Current interface state
	ifLastChange	How long ago interface changes state
	ifInOctets	Total octets received from the data
	ifInUcastPkts	Unicast packets delivered above
	ifInNUcastPkts	Broadcast/multicast packets delivered above
	ifInDiscards	Packets discarded due to resource limitations
	ifInErrors	Packets discarded due to format
	ifInUnknownProtos	Packets destined for unknown protocols
	ifOutOctets	Total octets sent on the media
	ifOutUcastPkts	Unicast packets from above
	ifOutNUcastPkts	Broadcast/multicast packets from above
	ifOutDiscards	Packets discarded due to resource limitations
	ifOutErrors	Packets discarded due to errors
	ifOutQLen	Packet size of output queue
	ifSpecific	MIB-specific pointer
Address Translation	atIfIndex	The interface on which this entry's equivalence is effective
	atPhysAddress	The media-dependent physical address
	atNetAddress	The NetworkAddress (for example, the IP address) corresponding to the media-dependent physical address
IP	ipForwarding	Acting as a gateway or host
	ipDefaultTTL	Default TTL for IP packets
	ipInReceives	Total datagrams received
	ipInHdrErrors	Datagrams discarded due to format errors
	ipInAddrErrors	Datagrams discarded due to misdelivery
	ipForwDatagrams	Datagrams forwarded
	ipInUnknownProtos	Datagrams destined for unknown protocols
	ipInDiscards	Datagrams discarded due to resource limitations
	ipInDelivers	Datagrams delivered above
	ipOutRequests	Datagrams from above
	ipOutDiscards	Datagrams discarded from above
	ipOutNoRoutes	Datagrams discarded due to no route
	ipReasmTimeout	Timeout value for reassembly queue
	ipReasmReqds	Fragments received needing reassembly

MIB Data and Descriptions

Group	MIB	Description
	ipReasmOKs	Datagrams successfully reassembled
	ipReasmFails	Reassembly failure
	ipFragsOK	Datagrams successfully fragmented
	ipFragsFails	Datagrams fail fragmented
	ipFragsCreates	Fragments created
	ipAdEntAddr	The IP address of this entry
	ipAdEntIfIndex	Interface number
	ipAdEntNetMask	Subnet mask for IP address
	ipAdEntBcastAddr	LSB of IP broadcast address
	ipAdEntReasmMaxSize	The largest IP datagram able to be reassembled
	ipRouteDest	Destination IP address
	ipRouteIfIndex	Interface number
	ipRouteMetric1	Routing metric number 1
	ipRouteMetric2	Routing metric number 2
	ipRouteMetric3	Routing metric number 3
	ipRouteMetric4	Routing metric number 4
	ipRouteNextHop	Next hop (gateway IP address for indirect routing)
	ipRouteType	Type (direct, remote, valid, invalid)
	ipRouteProto	Mechanism used to determine route
	ipRouteAge	Age of route in seconds
	ipRouteMask	Subnet mask for route
	ipRouteMetric5	An alternate routing metric for this route
	ipRouteInfo	Route information
	ipNetToMediaIfIndex	Interface number
	ipNetToMediaPhysAddress	Media address of mapping
	ipNetToMediaNetAddress	IP address of mapping
	ipNetToMediaType	How mapping was detemined
	ipRoutingDiscards	Routing entries discarded
ICMP	icmpInMsgs	ICMP messages received
	icmpInErrors	Error ICMP messages received
	icmpInDestUnreachs	Destination unreachable ICMP messages received
	icmpInTimExcds	Time exceed ICMP messages received
	icmpInParmProbs	Param error ICMP messages received
	icmpInSrcQuenchs	Source quench ICMP messages received
	icmpInRedirects	Redirect ICMP messages received
	icmpInEchos	Echo request ICMP messages received
	icmpInEchoReps	Echo reply ICMP messages received
	icmpInTimestamps	Time stamp request ICMP messages received

MIB Data and Descriptions

Group	MIB	Description
	icmpInTimestampReps	Time stamp reply ICMP messages received
	icmpInAddrMasks	Address mask request ICMP messages received
	icmpInAddrMaskReps	Address mask reply ICMP messages received
	icmpOutMsgs	ICMP messages transmitted
	icmpOutErrors	Error ICMP messages transmitted
	icmpOutDestUnreachs	Destination unreachable ICMP messages transmitted
	icmpOutTimeExcds	Time exceed ICMP messages transmitted
	icmpOutParmProbs	Param error ICMP messages transmitted
	icmpOutSrcQuenchs	Source quench ICMP messages transmitted
	icmpOutRedirects	Redirect ICMP messages transmitted
	icmpOutEchos	Echo request ICMP messages transmitted
	icmpOutEchoReps	Echo reply ICMP messages transmitted
	icmpOutTimestamps	Time stamp request ICMP messages transmitted
	icmpOutTimestampReps	Time stamp reply ICMP messages transmitted
	icmpOutAddrMasks	Address mask request ICMP messages transmitted
	icmpOutAddrMaskReps	Address mask reply ICMP messages transmitted
TCP	tcpRtoAlgorithm	Identifies retransmission algorithm
	tcpRtoMin	Minimum retransmission timeout in ms
	tcpRtoMax	Maximum retransmission timeout in ms
	tcpMaxConn	Maximum of simultaneous TCP connections allowed
	tcpActiveOpens	Number of active opens
	tcpPassiveOpens	Number of passive opens
	tcpAttemptFails	Number of failed connection attempts
	tcpEstabResets	Number of connections reset
	tcpCurrEstab	Number of current connections
	tcpInSegs	Number of segments received
	tcpOutSegs	Number of segments sent
	tcpRetranSegs	Number of segments retransmitted
	tcpConnState	State of connection
	tcpConnLocalAddress	Local IP address
	tcpConnLocalPort	Local TCP port
	tcpConnRemoteAddress	Remote IP address
	tcpConnRemotePort	Remote TCP port
	tcpInErrors	Number of segments discarded due to format errors
	tcpOutRsts	Number of resets generated

MIB Data and Descriptions

Group	MIB	Description
UDP	udpInDatagrams	Datagrams delivered above
	udpNoPorts	Datagrams destined for unknown ports
	udpInErrors	Datagrams discarded due to format errors
	udpOutDatagrams	Datagrams sent from above
	udpLocalAddress	Local IP address
	udpLocalPort	Local UDP port
EGP	egpInMsgs	EGP message received
	egpInErrors	Error EGP message received
	egpOutMsgs	EGP message transmitted
	egpOutErrors	Error EGP message transmitted
	egpNeighState	EGP state of the local system with respect to this entry's EGP neighbor
	egpNeighAddr	IP address of this entry's EGP neighbor
	egpNeighAs	Autonomous system of this EGP peer
	egpNeighInMsgs	Number of EGP messages received without error from this EGP peer
	egpNeighInErrs	Number of EGP messages received with error from this EGP peer
	egpNeighOutMsgs	Number of EGP messages transmitted without error to this EGP peer
	egpNeighOutErrs	Number of EGP messages transmitted with error to this EGP peer
	egpNeighInErrMsgs	Number of EGP-defined error messages received from this EGP peer
	egpNeighOutErrMsgs	Number of EGP-defined error messages transmitted to this EGP peer
	egpNeighStateUps	Number of EGP state transitions to the UP state with this EGP peer
	egpNeighStateDowns	Number of EGP state transitions to the DOWN state with this EGP peer
	egpNeighIntervalHello	Interval between EGP Hello command
	egpNeighIntervalPoll	Interval between EGP Hello command retransmissions (in hundredths of a second)
	egpNeighMode	Interval between EGP Poll command retransmissions (in hundredths of a second)
	egpNeighEventTrigger	Polling mode of this EGP
	egpAs	Event to trigger operator initiated start and stop Autonomous system number of this EGP entity
SNMP	snmpInPkts	SNMP packets received

MIB Data and Descriptions

Group	MIB	Description
	snmpOutPkts	SNMP packets transmitted
	snmpInBadVersions	Bad version SNMP packets received
	snmpInBadCommunityNames	Bad community name SNMP packets received
	snmpInBadCommunityUsers	Bad community user SNMP packets received
	snmpInASNParseErrs	ASN parse error SNMP packets received
	snmpInTooBigs	Too big SNMP packets received
	snmpInNoSuchNames	No such name SNMP packets received
	snmpInBadValues	Bad value SNMP packets received
	snmpInReadOnlys	Read only SNMP packets received
	snmpInGenErrs	Generation error SNMP packets received
	snmpInTotalReqVars	The total number of MIB objects which have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs
	snmpInTotalSetVars	The total number of MIB objects which have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs
	snmpInGetRequests	Number of get request packets received
	snmpInGetNexts	Number of get next packet received
	snmpInSetRequests	Number of set request packet received
	snmpInGetResponses	Number of set response packet received
	snmpInTraps	Number of trap message received
	snmpOutTooBigs	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field was “tooBig”
	snmpOutNoSuchNames	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field was `”No Such Names”
	snmpOutBadValues	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field was “Bad Values”
	snmpOutGenErrs	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field was “Gen Err”
	snmpOutGetRequests	Number of get request packet transmitted
	snmpOutGetNexts	Number of get next packet transmitted
	snmpOutSetRequests	Number of set request packet transmitted
	snmpOutGetResponses	Number of set response packet transmitted
	snmpOutTraps	Number of trap message transmitted
	snmpEnableAuthenTraps	Indicates whether the SNMP entity is permitted to generate authenticationFailure traps

Knowledgebase Quick Starts

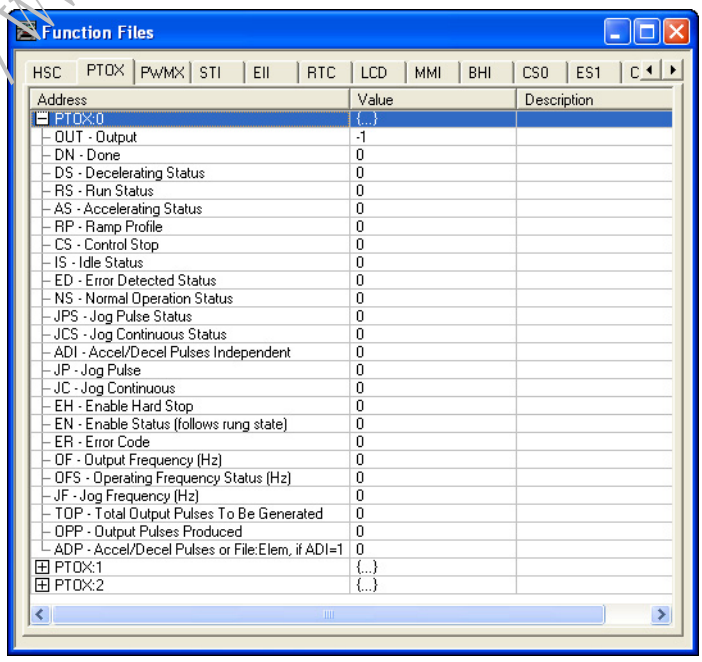
The following Quick Start topics are included:

- # 17444 Pulse Train Output (PTOX) Quick Start on page 577
- # 17446 Pulse Width Modulation (PWMX) Quick Start on page 580
- # 17447 High Speed Counter (HSC) Quick Start on page 582
- # 17465 Message (MSG) Quick Start on page 586
- # 17501 Selectable Timed Interrupt (STI) Quick Start on page 590
- # 17503 Real Time Clock (RTC) Quick Start on page 592
- # 17558 User Interrupt Disable (UID) Quick Start on page 593
- # 18465 RTC Synchronization Between Controllers Quick Start on page 594
- # 18498 Data Logging (DLG) Quick Start on page 597

17444 Pulse Train Output (PTOX) Quick Start

The PTOX and PWMX functions are only available when using the BXB or BXBA models of the MicroLogix 1400.

Locate the Function Files under Controller in RSLogix 500/RSLogix Micro v8.10.00 or later and select the PTOX tab, then click the [+] sign next to PTOX:0. See screencap below.

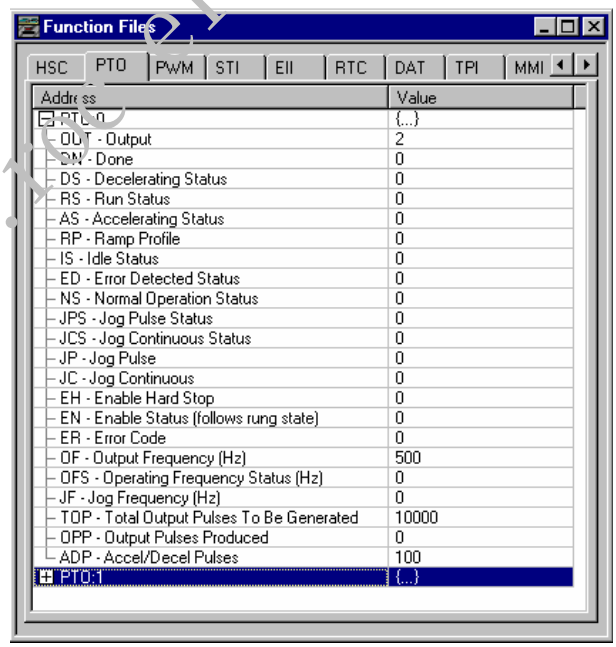


Enter the following parameters as the “Minimum Configuration” required for the PTOX to generate pulses.

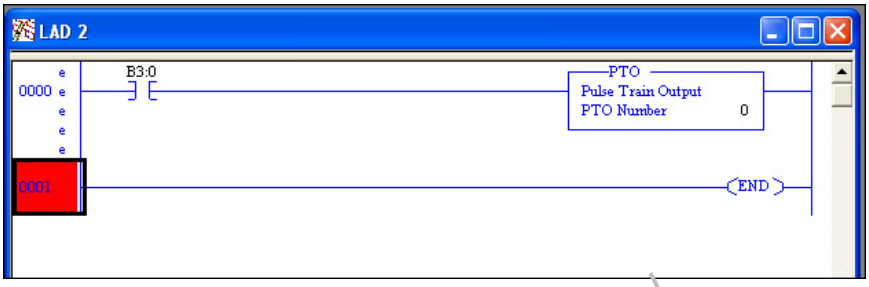
PTOX:0.OUT	Select Destination Output for pulses: Output O:0/2, O:0/3 or O:0/4
PTOX:0.OF	Output Frequency - Frequency of pulses: 0...100,000 Hz Data less then zero and greater then 100,000 generates a PTOX error
PTOX:0.TOP	Total Output Pulses - Determines total number of pulses to be generated by the controller
PTOX:0.ADP	Accel/Decel Pulses - How many of the total pulses will be used for the Accel/Decel component

Example

The following example will generate 10,000 pulses on Output O:0/2 at a frequency of 500Hz and 100 pulses will be used for Accelerating and 100 pulses will be used for decelerating.



The following ladder logic will need to be entered into File #2



By toggling Bit B3/0 the PTOX can be activated. On execution, the PTOX instruction generates the number of pulses entered into the PTOX:0.TOP long word and then stop. To restart, toggle B3/0.

General Information on the PTO

Once running, the PTOX instruction continuously generates pulses until all pulses have been generated or the PTOX:0/EH (Enable Hard Stop) bit has been activated.

Once the EH bit is set the instruction will generate a PTOX error of 1 (hard stop detected). In order to clear this error the PTOX instruction must be scanned on a false rung of logic and the EH bit must be off.

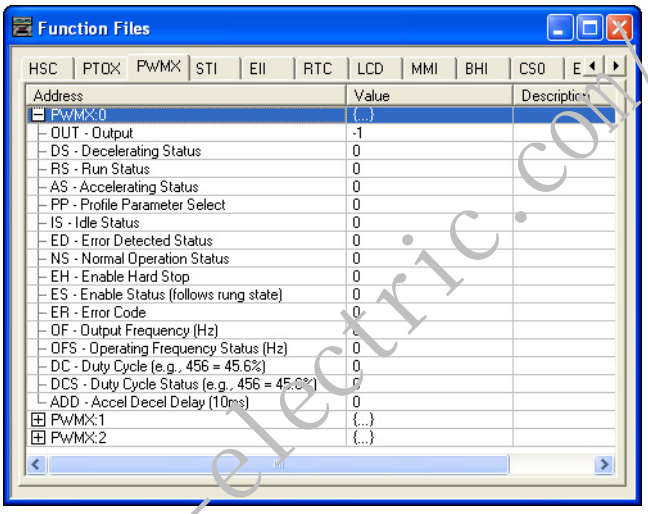
To change the Total Output Pulses Generated in a working program a new value can be moved into PTOX:0.TOP by using the MOV command.

IMPORTANT Once the PTOX has been initiated and is generating pulses a new TOP value will not take effect until the PTOX has either completed generating pulses and has been restarted or has been Hard Stopped using PTOX:0/EH bit and been restarted.

17446 Pulse Width Modulation (PWMX) Quick Start

IMPORTANT The PTOX and PWMX functions are only available when using the BXB or BXBA models of the MicroLogix 1400.

Locate the Function Files under Controller in RSLogix 500/RSLogix Micro v8.10.00 or later and select the PWMX tab, then click the [+] sign next to PWMX:0 as shown below.



Enter the following parameters as the "Minimum Configuration" required for the PWMX to generate a waveform at the specified frequency.

PWMX:0.OUT Select Destination Output for pulses: Output O:0/2, O:0/3 or O:0/4

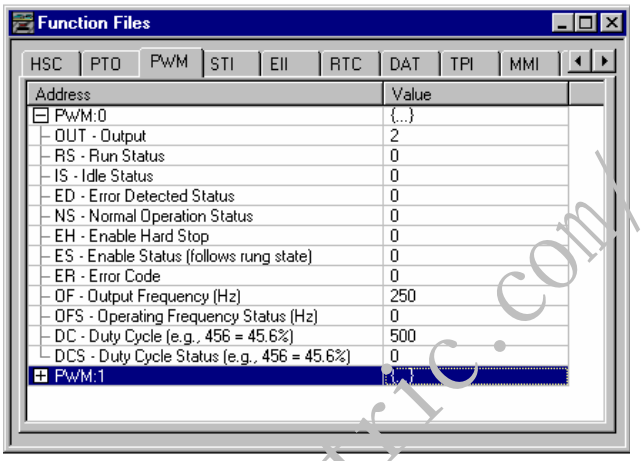
PWMX:0.OFS Output Frequency - Frequency of the PWM: 0...40,000 Hz

PWMX:0.DC PWMX Duty Cycle - Controls the output signal of the PWM: 1...1000

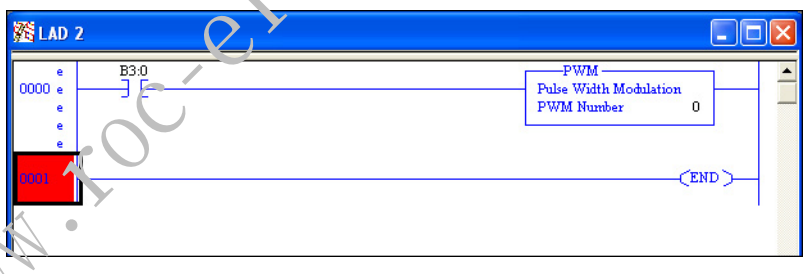
DC = 1000	100%	Output **ON**	(Constant no waveform)
DC = 0750	075%	Output **ON**	025% Output **OFF**
DC = 0500	050%	Output **ON**	050% Output **OFF**
DC = 0250	025%	Output **ON**	075% Output **OFF**
DC = 0000	000%	Output **OFF**	(Constant no waveform)

Example

The following example generates a waveform on Output O:0/2 at a frequency of 250 Hz and a 50% Duty Cycle.



The following ladder logic will need to be entered into file #2.



By toggling Bit B3/0 the PWMX instruction can be activated.

IMPORTANT	Once activated, the PWMX instruction continuously generates a waveform until B3/0 is toggled OFF or the PWMX:0/EH (Enable Hard Stop) bit has been activated.

17447 High Speed Counter (HSC) Quick Start

General Information

The MicroLogix 1400 has six 100KHz high-speed counters. There are three main high-speed counters (counters 0, 1, and 2) and three sub-high speed counters (counters 3, 4, and 5).

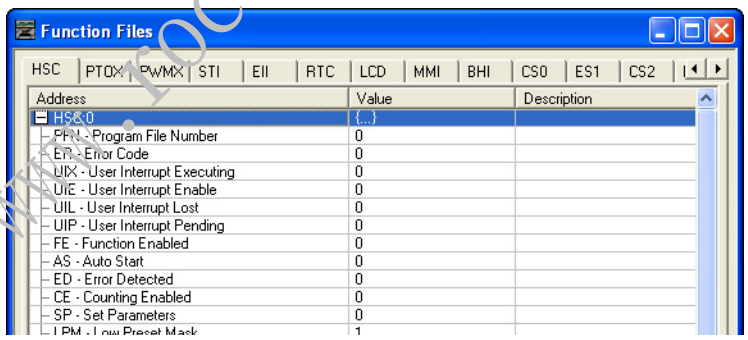
Each main high-speed counter has four dedicated inputs and each sub high-speed counter has two dedicated inputs.

HSC0 utilizes inputs 0...3, HSC1 utilizes inputs 4...7, HSC2 utilizes inputs 8...11, HSC3 utilizes inputs 2 and 3, HSC4 utilizes inputs 6 and 7 and HSC5 utilizes inputs 10 and 11.

In some cases, a sub-counter can be disabled by master counter mode. Refer to HSC Mode (MOD) on page 91. Input device connection depends on the counter mode selected. The MicroLogix 1400 uses a 32-bit signed integer for the HSC this allows for a count range of (+/-) 2,147,483,647.

Getting Started

Locate the Function Files under Controller in RSLogix 500/RSLogix Micro and select the HSC tab, then select the [+] sign next to HSC:0 as shown below.



Enter the following parameters for the "Minimum Configuration" required for the HSC to count pulses.

IMPORTANT There is no additional ladder logic required to enable the High Speed Counter. In other words there is no HSC instruction needed for the ladder logic program.

HSC:0.PFN Program File Number defines which subroutine is executed when the HSC:0 accumulated count equals the High or Low preset or passes through Overflow or Underflow. The Integer number entered must be a valid sub-routine program file (3...255).

HSC:0/AS Auto-Start defines if the HSC function will automatically start when the MicroLogix controller enters run or test.

HSC:0/CE Counting Enabled control bit is used to enable or disable the HSC.

HSC:0.HIP High Preset is the upper set point (in counts) that defines when the HSC will generate an interrupt and execute the PFN sub-routine.

Example

The following example uses the HSC in **Mode 0** - "Up Counter". The **"Up Counter"** clears the accumulated value (0) when it reaches the High Preset (HIP). This mode configures I1:0.0/0 (I:0/0) as the HSC:0 input.

IMPORTANT Each mode for the HSC will configure the inputs for different functionality.

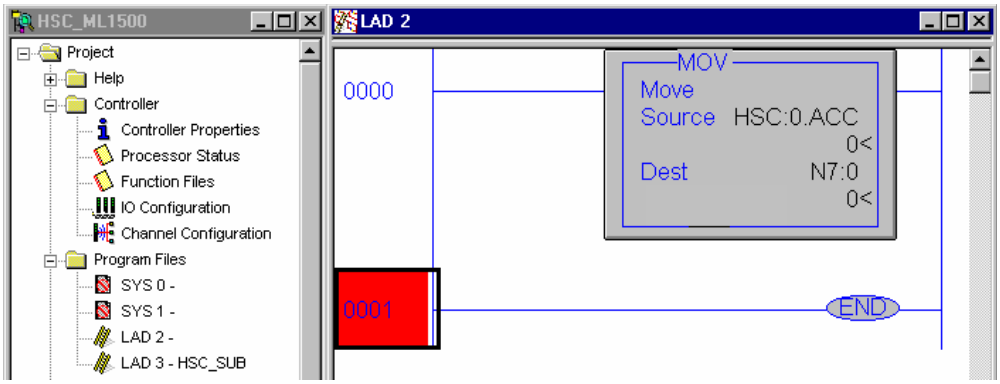
In this example, the HSC will count input pulses coming into I:0/0, when the total number of pulses counted equals the High Preset (HIP) the HSC will jump to subroutine file #3.

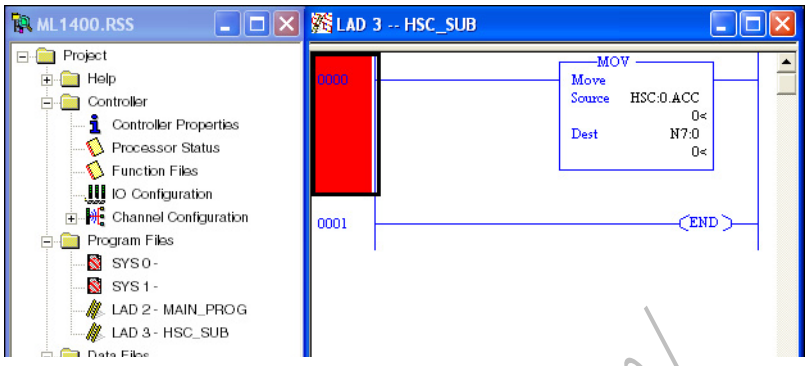
The HIP is set for 5000 pulses in this example. Also, once the HIP is reached the HSC will then reset HSC:0.ACC to zero (0) and start counting again.

IMPORTANT It is assumed that the user has connected a device to I:0/0 to generate pulses.

IMPORTANT The following ladder logic does not need to be entered into File #2, however this allows for easy viewing of the accumulated counts from the HSC:0.ACC.

IMPORTANT Ladder Logic Subroutine file #3 must be created in order for this example to work. If the subroutine is not created the CPU will fault due to an HSC Error Code 1 - Invalid File Number for PFN has been entered.

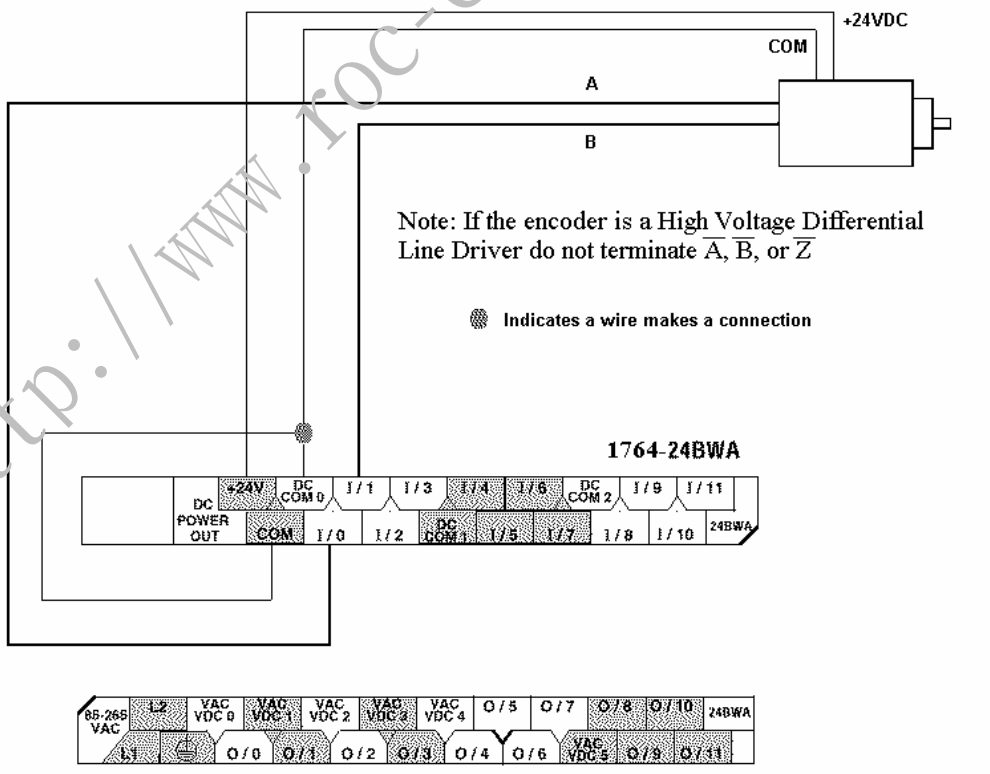




Proper wiring of a single ended encoder (Typical Allen-Bradley 845TK) when configuring HSC.MOD for Mode 6 (Quadrature Counter).

The following diagram shows an encoder connected to the MicroLogix 1400 controller.

The minimum configuration required for Mode 6 operation is to enter a file number for the PFN parameter, set the AS and CE bits to a (1) and enter a (6) for the MOD parameter.



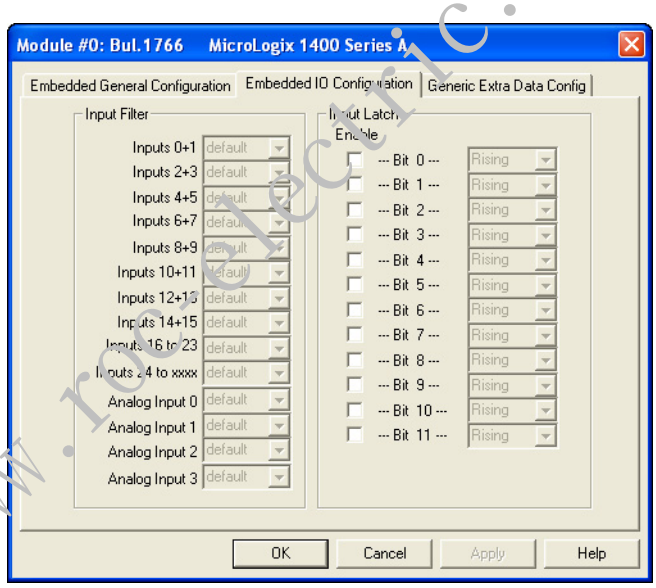
Troubleshooting

Problem #1: The input display on the MicroLogix LCD screen turns on and off, but no counts are seen in the HSC accumulator.

Solution: The input filter frequency may need to be adjusted in order to capture the input pulses.

Follow the steps below.

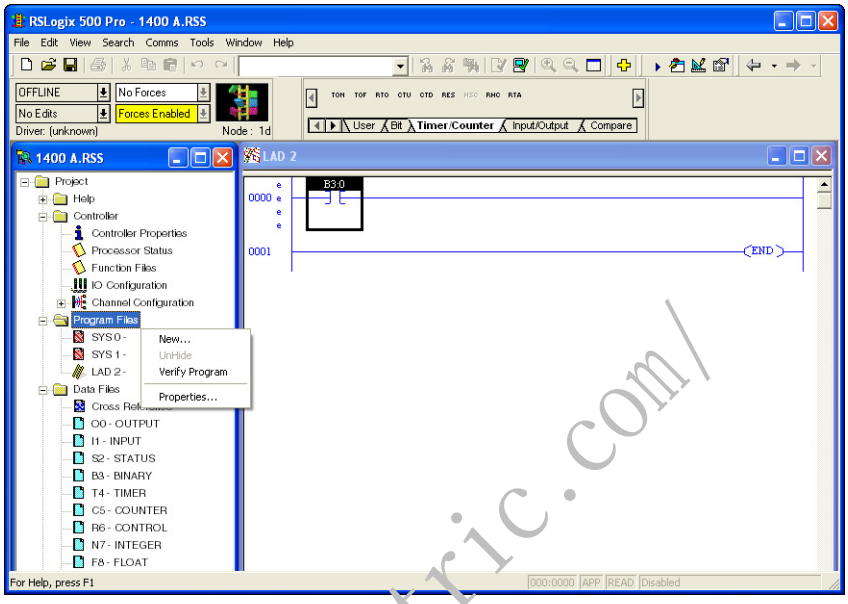
1. Select "I/O Configuration".
2. Highlight the "MicroLogix 1400".
3. Select "Adv Config".
4. Select the "Embedded I/O Configuration" tab.
5. Adjust Input filters as needed.



Problem #2: The HSC instruction does not accumulate counts and the Error Code (ER) shows a value of (1).

Solution: A file number was entered into (PFN) but the value entered was less than (3) or greater then (255) or the file number entered was correct, however the file does not exist.

Create the NEW program file. Right-click Program Files.



Problem #3: Some of my outputs does not turn On or Off when the ladder logic appears to indicate that they should.

Solution: **OMB (Output Mask Bits)** - Verify what the OMB has been configured for in the HSC function file. If an output has been assigned to the HSC for control, then the output will not be controlled anywhere else in the ladder program. Only the HSC will have control over these outputs.

17465 Message (MSG) Quick Start

Communications Specifications

The MicroLogix 1400 processors contain a total of (12) Message Buffers.

(8) Incoming	Any incoming MSG, communication, and/or responses to a command the ML1400 initiated.
(4) Outgoing	Any outgoing MSG, communication and/or responses to incoming request for data.

The Outgoing queue also supports unlimited queuing. This means that even if a buffer is not available the MSG will simply wait until one of the outgoing buffers becomes available and then transmit.

IMPORTANT	If a message has been waiting in the queue, at the moment of buffer availability, the most current data will be sent, not the data that was available at the time the message instruction was first scanned true.

How quickly a message is actually sent or received to/by a destination device depends on a number of factors, including the selected channels communication

protocol, baud rate of the communications port, number of retries, destination devices readiness to receive, ladder logic scan time, and so on.

Message (MSG) instruction

The message instruction (MSG) is an output instruction which, when configured correctly, allows data to be sent or received to other compatible devices.

The MSG instruction in the MicroLogix 1400 controller uses a Data File MG to process the message instruction. All message elements are accessed using the MG prefix (example: MSG done bit = MG11:0/DN).

Continuous Message Example

The following example illustrates how, by using the MSG Done (DN) and Error (ER) bits to unlatch the Enable (EN) bit the MSG instruction can be configured for continuous execution.

This example uses MG11:0 for the MSG file and will require two MicroLogix controllers one a ML1400 and the other either a ML1000 or ML1400. The ML1400 will need to be configured as Node 1 and the other processor as node 4.

The processor at node 1 will contain the ladder logic below and transfer data from it's N7:0 Integer file to the processor at node 4's N7:0 Integer file. Since N7:0 is the source file for this example, data must be entered into this register for node 1. For this example Locate N7:0 in the ML1400 (Node 1) and enter the value 63.

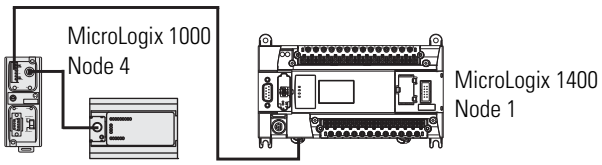


Figure 9 - MicroLogix 1400 (Node 1) Ladder Logic

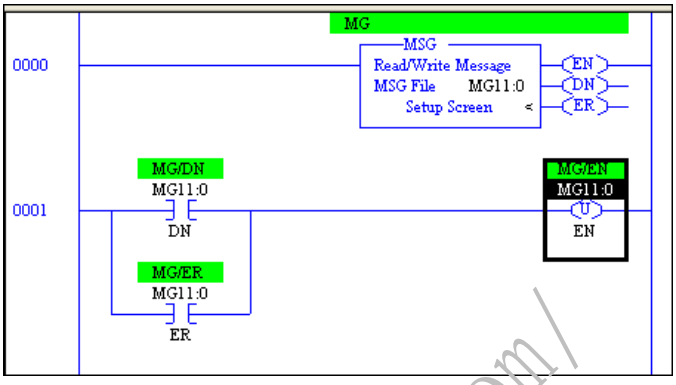
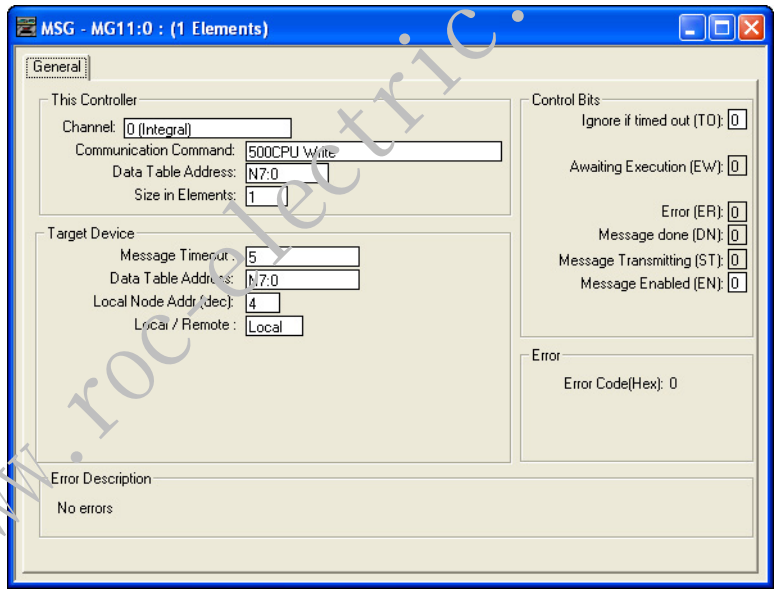


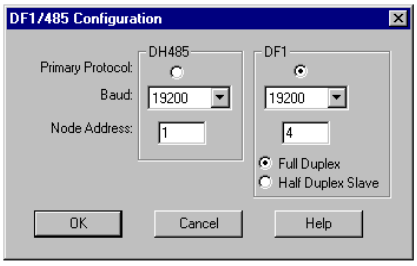
Figure 10 - MSG Setup Screen



Micrologix 1000 (Node 4) Ladder Logic

No ladder logic is required in the destination processor, however, the communications channel must be configured to match the source processor. Since the default settings for the ML1400 communications channel is DF1 protocol, 19,200 Kbaud the ML1000 must be configured to match.

Figure 11 - MicroLogix 1000 Channel Configuration



IMPORTANT Do not connect to MicroLogix 1000 directly using a 1761-CBL-AM00 cable.

IMPORTANT After the ladder logic has been entered into the MicroLogix 1400 and the MicroLogix 1000 channel configuration has been changed, leave it connected until the COMM 0 LED on the MicroLogix 1400 starts to blink. This ensures proper connection to the controllers using a 1761-CBL-PM02 cable

Verify Data Sending

To verify the data has been sent to node 4, disconnect the PM02 cable and connect the PC running RSLogix 500/RSLogix Micro to the ML1000 (Node 4). Go to N7:0 and view the data, this should match the data in N7:0 of node 1.

Another way to verify the data has been sent to node 4 is to replace the Target Device Data Table Address with an output modules address. In this example, the output module is a MicroLogix 1000, the address is O:0.0 This displays on the output LEDs, whatever number (in binary format) was entered into N7:0 of the ML1400.

IMPORTANT By addressing O:0.0 the outputs of the destination processor will be energized upon successful transmission of data. Verify that nothing is connected to the outputs to ensure safe operation of the controller.

If a 16-point MicroLogix 1000 is being used as the destination processor (Node 1), and the number 63 is entered into the above example, all the outputs will be energized. If the number entered is greater than 63, then a fault may occur with an error stating that the extended I/O bit (S:0/8) was not set. In this case, clear the fault, go offline, set bit (S:0/8) and re-download the ladder program.

The above example uses the DF1 Full Duplex protocol. This is a point-to-point or One Device to One Device protocol, using this protocol no other devices can be connected. To create a network of multiple processors or devices use the DH485 protocol and 1761-NET-AIC devices.

IMPORTANT This example was written using a MicroLogix 1400 communicating with a MicroLogix 1000. However any DF1 or DH485 device may substitute MicroLogix 1000 (such as MicroLogix 1200, MicroLogix1500, SLC 5/03, 5/04, 5/05, PLC-5, Bar Code Scanners, and so on).

17501 Selectable Timed Interrupt (STI) Quick Start

What is an Interrupt?

An interrupt is an event that causes the processor to suspend the task that it is currently performing, perform a different task, and then return to the suspended task at the point where it suspended.

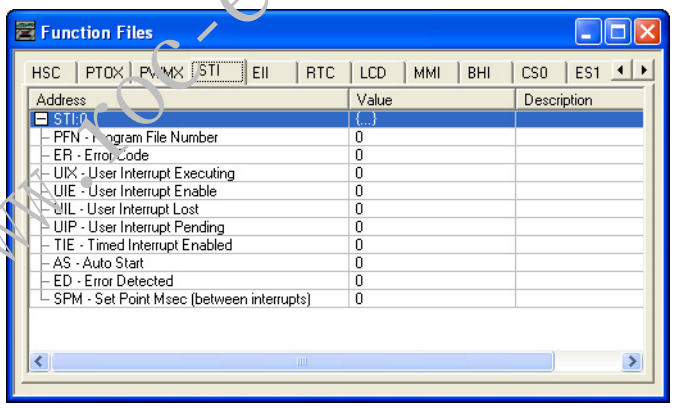
Selectable Timed Interrupt

STI provides a mechanism to solve time-critical control requirements. The STI is a trigger mechanism that allows you to scan or solve control program logic that is time sensitive.

Example of application: A block of logic that needs to be scanned more often than the rest of the ladder program.

Getting Started

Locate the Function Files under Controller in RSLogix 500/RSLogix Micro v8.10.00 or later and select the STI tab as shown below.



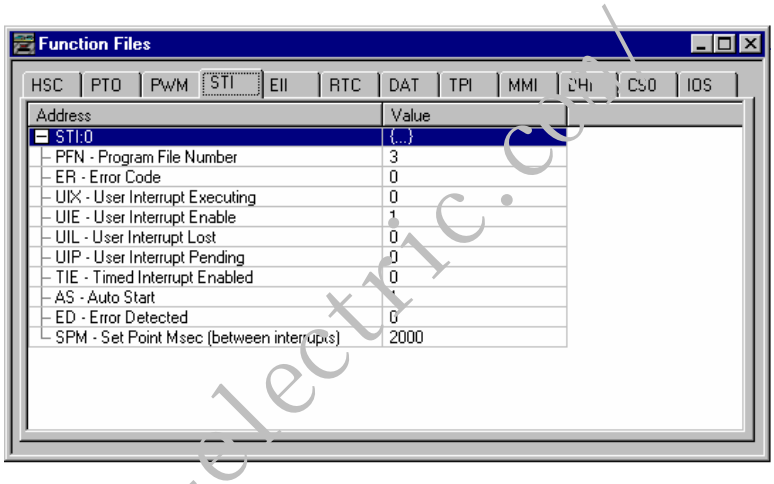
Enter the following parameters as the Minimum Configuration required for the STI.

STI:0.PFN	Program File Number defines which subroutine is executed when the SPM value has timed out. The Integer number entered must be a valid sub-routine program file (3...255).
STI:0/AS	Auto-start defines if the STI function will automatically start when the MicroLogix 1400 enters run or test.
STI:0/UIE	User Interrupt Enabled control bit is used to enable or disable the STI subroutine from processing.
STI:0.SPM	Setpoint (in milliseconds) defines the interval that the interrupt will scan the PFN sub-routine.

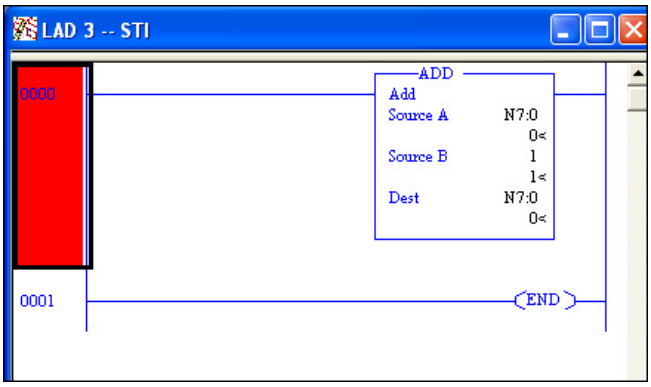
Example

The following example configures the STI to execute sub-routine file #3 (PFN=3) every 2 seconds (SPM=2000). In the subroutine file there is an ADD instruction that simply adds the value of 1 to N7:0 each time the sub-routine is scanned.

This example also sets the User Interrupt Enable bit and the Auto Start bit allowing the STI to execute.



IMPORTANT Ladder Logic Subroutine file #3 must be created in order for this example to work. If the subroutine is not created the CPU will fault due to a STI Error Code 1 - Invalid File Number for PFN has been entered.



Some Guidelines on Using Interrupt bits

If the Auto Start bit (AS) is set, this starts the interrupt upon power up and sets the Timed Interrupt Enabled bit (TIE) automatically, allowing the interrupt to execute as shown in the above example.

If the AS bit is not set, then the TIE bit must be set through the ladder logic in order for the interrupt to execute.

The User Interrupt Enable bit (UIE) determines if the interrupt executes or not.

17503 Real Time Clock (RTC) Quick Start

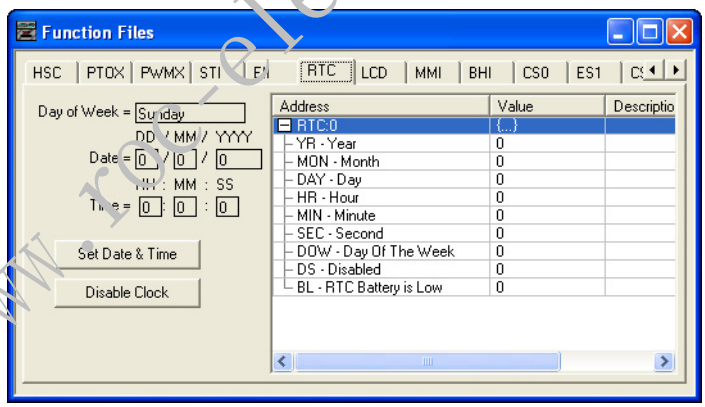
General Information

The RTC instruction provides Year, Month, Day, Day of Month, Day of Week, Hour, Minute, and Second information to the RTC Function file in the controller.

The MicroLogix 1400 controller has a built-in real time clock.

Getting Started

Locate the Function Files under Controller in RSLogix 500/RSLogix Micro v8.10.00 or later and select the RTC tab as shown below.

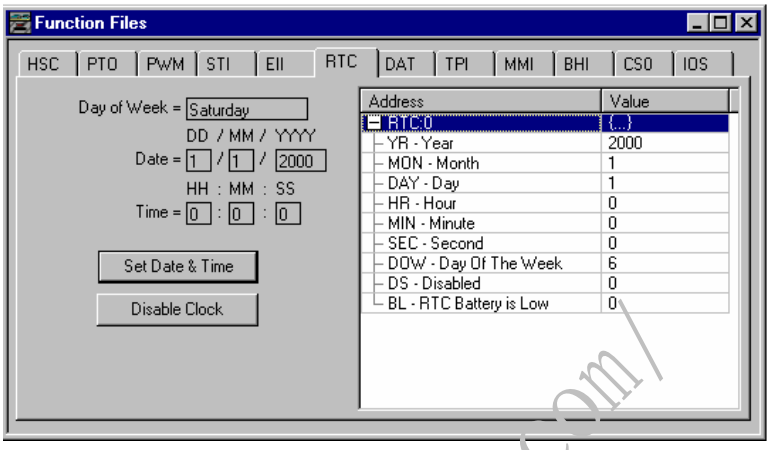


Values can be entered for the Year, Month, Day, Hour, Minute, and Seconds offline, once downloaded, the values will take effect immediately.

Note: The Day of the week is calculated by the RTC Online.

Press the 

button to set the MicroLogix 1400 clock to the same Date and Time as the computer connected online.



Press the 

button to disable the RTC from functioning and decrease the drain on the battery during storage.

RTC:0/ BL The Battery Low bit will be set (1) when the battery is low. This means that the battery will fail in less than 14 Days after which the RTC data may become invalid. At this time, replace battery. The RTC uses the same battery that the controller uses.

17558 User Interrupt Disable (UID) Quick Start

The UID instruction can be used as an output instruction to disable selected user interrupts.

Once a user interrupt is disabled, the User Interrupt Enable bit (UIE) for the selected interrupt is cleared or resets to zero (0). This stops the interrupt from executing.

To re-enable an interrupt, the UIE bit must be set to a one (1), or a UIE instruction must be used.

The following table indicates the types of interrupts disabled by the UID.

Types of Interrupt Disabled by the UID

Interrupt	Element	Decimal Value	Corresponding Bit
EII - Event Input Interrupt	Event 4	16384	bit 14
HSC - High-Speed Counter	HSC2	8192	bit 13
EII - Event Input Interrupt	Event 5	4096	bit 12
HSC - High-Speed Counter	HSC3	2048	bit 11
EII - Event Input Interrupt	Event 6	1024	bit 10
HSC - High-Speed Counter	HSC4	512	bit 9
EII - Event Input Interrupt	Event 7	256	bit 8

Types of Interrupt Disabled by the UID

Interrupt	Element	Decimal Value	Corresponding Bit
HSC - High-Speed Counter	HSC5	128	bit 7
EII - Event Input Interrupt	Event 0	64	bit 6
EII - Event Input Interrupt	Event 1	32	bit 5
HSC - High-Speed Counter	HSC0	16	bit 4
EII - Event Input Interrupt	Event 2	8	bit 3
EII - Event Input Interrupt	Event 3	4	bit 2
HSC - High-Speed Counter	HSC1	2	bit 1
STI - Selectable Timed Interrupt	STI	1	bit 0
Note: Bit 15 must be set to zero			

To disable interrupt(s) follow these steps.

1. Select which Interrupt(s) to disable from the above table.
2. Locate the decimal value for each Interrupt(s).
3. Add the decimal values together if more then one Interrupt was selected.
4. Enter the sum into the UID instruction.
   For example, to disable EII Event 1 and EII Event 3:
   EII Event 1 = 32 EII Event 3 = 04
   32 + 04 = 36 (Enter this value in the UID instruction)

Some Guidelines on Using Interrupt bits

The setting of the Auto Start bit (AS) starts the interrupt on power up and sets the Timed Interrupt Enabled bit (TIE) automatically, allowing the interrupt to execute.

If the AS bit is not set then the TIE bit must be set through the ladder logic in order for the interrupt to execute.

The User Interrupt Enable bit (UIE) determines if the interrupt executes or not.

18465 RTC Synchronization Between Controllers Quick Start

The following example illustrates a message write from an SLC 5/03 or higher processor to a Micrologix 1400 processor with RTC enabled.

This example can also be applied for messaging between MicroLogix 1100, 1200, 1400, and 1500 controllers. When messaging from a MicroLogix 1100/1200/1400/1500 controller to MicroLogix 1100/1200/1400/1500 controller it is recommended that RTC:0 be used as the source instead of (S:37 - S:42).

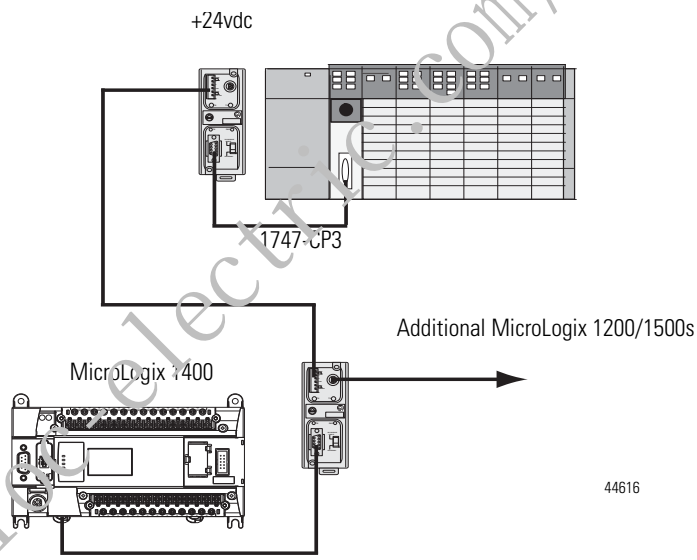
Minimum Hardware/Software Requirements

- All MicroLogix 1400

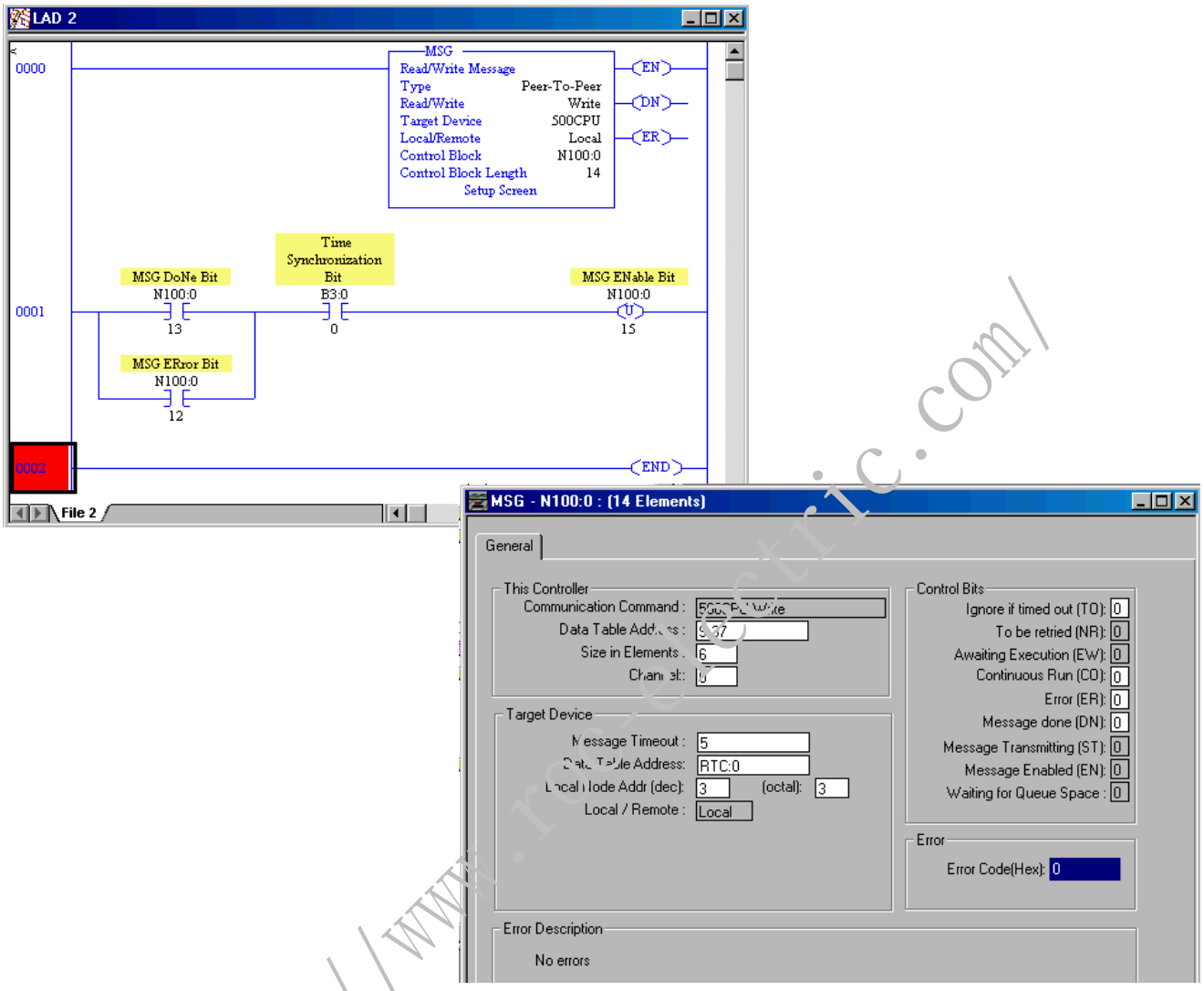
- MicroLogix 1200 Series B FRN 2
- MicroLogix 1500 Series B FRN 4
- RSLogix 500/RSLogix Micro v8.10.00
- All MicroLogix 1100

Example

The example shows network connections using DH-485, however DF1 Full or Half Duplex Ethernet/IP will also work.



1. Configure the SLC Channel 0 port for DH-485 protocol.
2. Enter the following ladder logic into the SLC processor.



The example above messages the SLC 500 Date and Time data (S:37 - S:42)to the MicroLogix 1400 RTC each time the SLC processor is powered up and placed into the RUN mode or each time the Time Synchronization Bit (B3:0/0) is enabled.

ATTENTION: Valid years for the MicroLogix 1400 begin with 1998. Any date/time/year values, prior to 1998, that are sent to a MicroLogix controller generates MSG Error Code 10h.

For each processor that requires its RTC to be synchronized, a MSG write will be required. This is done simply by duplicating the above ladder logic, referencing a different Control Block (that is, N100:0 = MSG1 | N100:20 = MSG2 | N100:40 = MSG3, and so on.) and specifying a different node address in the MSG set-up screen.

18498 Data Logging (DLG) Quick Start

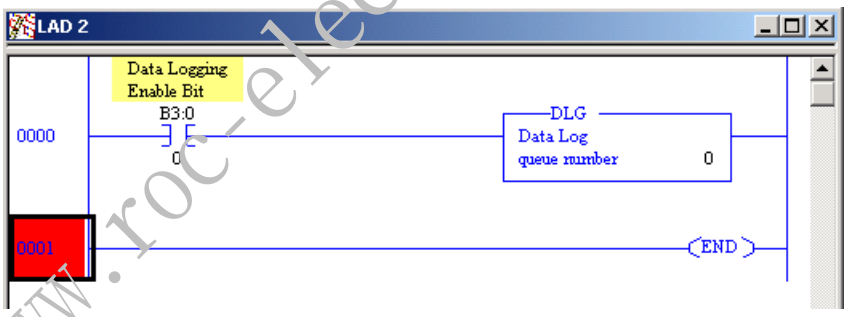
General Information

The Data Logging feature allows the creation of memory queues to capture or store application data as a record for later retrieval. Each record is stored in a user-configured battery backed queue. The size of memory where queues are stored is 128K bytes, this is independent of the rest of the processor memory.

The Data Logging feature allows the capture or storage of application data as a record for later retrieval. Each record is stored in a user-configured battery backed queue. The size of the queue is 128K bytes, independent of the rest of the processor memory.

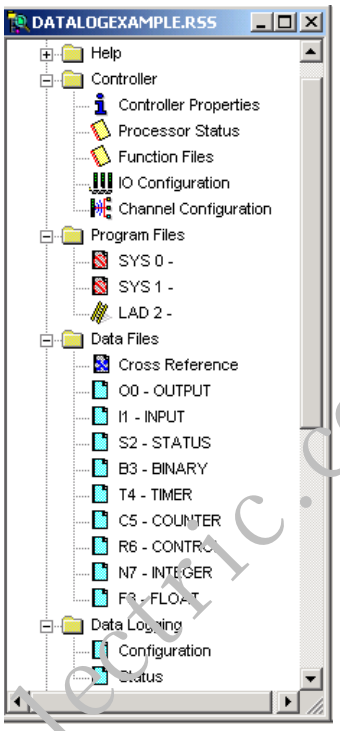
Configure the DLG Instruction in MicroLogix 1400

1. Create a new RSLogix 500/RSLogix Micro project for the MicroLogix 1400 controller.
2. Create a new rung of ladder logic in File 2 as shown below.



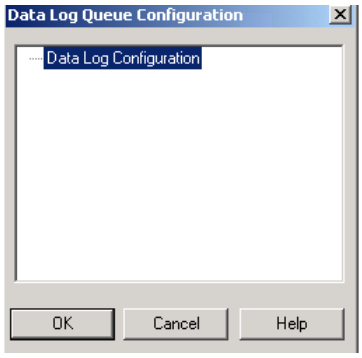
IMPORTANT The DLG instruction ONLY captures data on a false-to-true rung transition.

3. Double Click ***Data Logging*** - ***Configuration*** in the controller organizer to access the Data Log Queue Configuration window

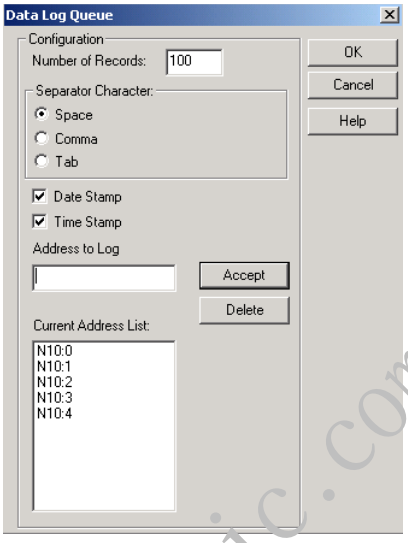


IMPORTANT Every time Configuration above is double-clicked a new queue is added. To delete queues, simply select the queue with the mouse and press the <delete> key on the keyboard.

4. Double-Click Data Log configuration to open the Configuration window.

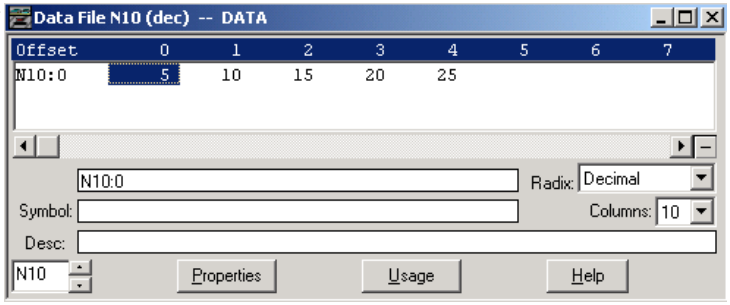


5. Complete the Data Log Queue as shown below. The number of records and addresses selected are arbitrary for this example.



IMPORTANT Integer file N10 must be created with a length of 5 or the software will not compile the ladder program. Also, an embedded RTC must be enabled and configured if the Date and Time stamp are to be used. If an RTC module is disabled the data for these fields will contain zeros.

6. Click OK when completed.
7. Click OK and accept the Data Log Queue window.
8. Once the N10 file has been created enter the following values for each.



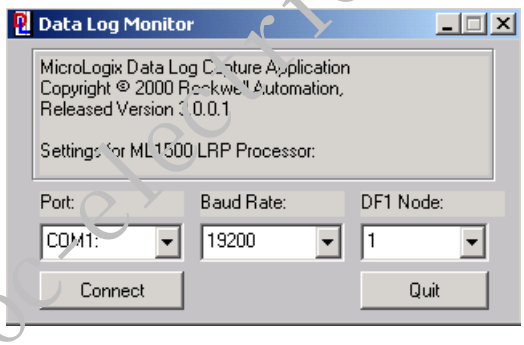
9. Download the program to your MicroLogix 1400 controller.
10. Go online.
11. Toggle the Data Logging Enable(B3:0/0) bit Off to On a total of 5 times.

Use the Data Logging Utility

ATTENTION: If any other software package, such as RSLinx has control of the computers communication port or if the wrong COM port is selected or a processor other then the MicroLogix 1400 is connected to the computer you will not be able to continue.

The Data Logging utility is the only supported method for retrieving data, that has been stored in the processor.

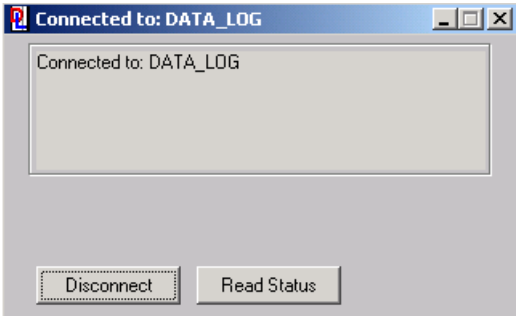
1. Install the DLOG utility (found at http://www.ab.com/plclogic/micrologix/)
2. Launch DLCA1764.EXE.
3. Configure Port, Baud Rate, and DF1 Node as shown below.



4. Click "Connect".

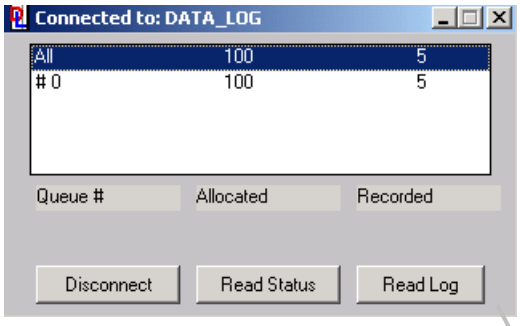
IMPORTANT By default the MicroLogix 1400 communications are configured for 19200 baud If using defaults select 19200 above, otherwise, select the baud rate configured in the MicroLogix Channel Configuration Screen.

If a correct configuration has been selected, the utility software will indicate that it has connected to the processor as shown below.



5. Click Read Status once a valid connection is established

The DLOG utility will now retrieve the status information from the MicroLogix 1400 controller.

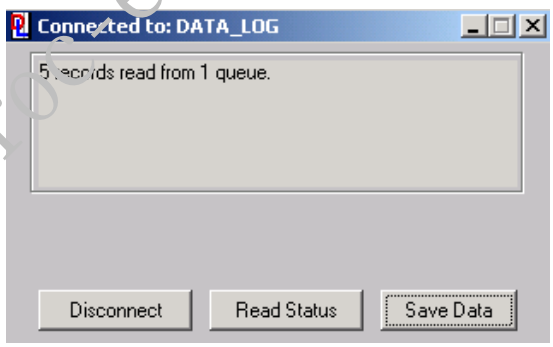


In this example you can see that Queue #0 has 100 records allocated and 5 recorded.

IMPORTANT If you do not see five records verify your Data Logging Enable bit was toggled 5 times causing the 5 entries to be recorded in the Queue.

6. Select Read Log. This retrieves the data from the MicroLogix 1400 controller.

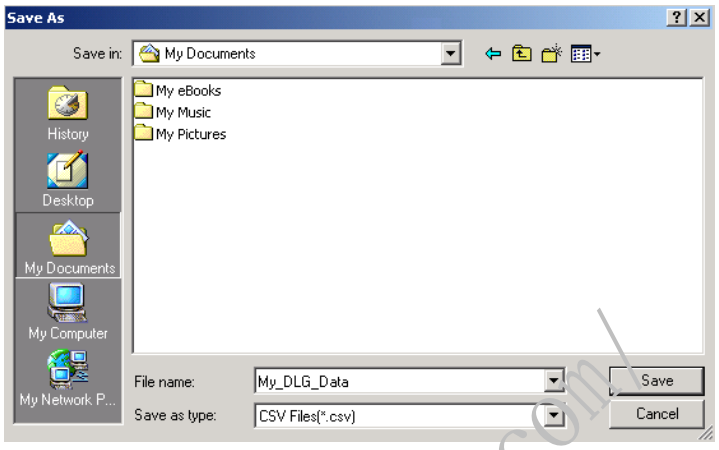
   Once the Read Log has completed the following screen appears confirming the number of records that have been read from the Queue(s)



IMPORTANT Remember that once the data records have been read from the MicroLogix controller, the queue is automatically cleared.

7. Click Save Data.
8. Enter a file name. In our example "My_DLG_Data" was used.

   Make note of the filename used and the directory where it was saved for future reference.



9. Open the data file that was created using Microsoft Excel.

IMPORTANT If you are unable to locate your file in Excel, remember "Files of type" must be changed to "Text Files" or "All files (*.*)" in order to locate your saved file.

The headings for each column are not stored in the data file these were added for readability.

QUE #	Date	Time	N10:0	N10:1	N10:2	N10:3	N10:4
Queue 0	5/1/2000	8:00:00	5	10	15	20	25
Queue 0	5/1/2000	8:00:02	5	10	15	20	25
Queue 0	5/1/2000	8:00:05	5	10	15	20	25
Queue 0	5/1/2000	8:00:07	5	10	15	20	25

Each time the DLG instruction receives a false-to-true transition, another entry is saved in the Data Logging queue. The above data reflects that the DLG instruction was executed 5 times. The above data also reflects that no data points had changed during each DLG execution.

Frequently Asked Questions

Question 1: Can I write my own software application to retrieve the data stored in the Data Logging queue?

Answer 1: Yes. In the MicroLogix 1400 Instruction Set Reference manual, under the Data Logging chapter, all the information necessary to create your own software application, for retrieving the data stored in the processors Data Logging queue, is shown. See Recipe and Data Logging on page 465.

Question 2: Can the MicroLogix 1400 controller automatically send the information stored in the Data Logging queue directly to a printer?

Answer 2: No. To retrieve data, use either the free Data Logging Utility or a custom application must be created by the user. If the data does not need to be stored in the processor, but sent directly to a printer, then use the ASCII instructions of the MicroLogix processor to send out the data.

Using the Datalog Utility

This utility allows retrieval of data remotely through a Remote Access Modem Kit (RAD).

For more information on Remote Access Modem Kits visit http://support.rockwellautomation.com/modem/modem_Main.asp

The following outlines the configuration and steps that can be used to read data log records from an MicroLogix 1400 controller remotely through a 1747CH0RAD (Remote Access Modem Kit).

This example assumes that the programmer has configured the DLG instruction in the ML1400 to log data and that HyperTerminal is installed, configured and the user is familiar with its use.

Establish Connections

1. Connect the modem to Channel 0 of the MicroLogix 1400.
2. Configure Channel 1 (9-Pin) for DF1 Full Duplex, 9600 baud, no parity, and full duplex modem handshaking. This setting is critical, as the system will not communicate if *full duplex modem* handshaking is not applied to the Comms Channel connected to the modem.
3. Configure HyperTerminal for direct connection to the PC COMM port the modem is connected to. Make sure the HyperTerminal connection is configured for 9600 baud.
4. Save configuration as “DataLog”.
5. Send the following dial-out string using HyperTerminal to dial the modem and establish the connection:

   AT&C1DT(Phone number of destination Modem) then press enter

   your modem will respond: **CONNECT 9600**

Once the connection is established, exit HyperTerminal by selecting File/Exit from the pull-down menu. When asked "Do you want to close connection" select Yes. This will only close the connection from HyperTerminal to the RS-232 port. The connection will remain active.

IMPORTANT It will appear as though HyperTerminal has disconnected. It has not; the connection is still established, only HyperTerminal is no longer running.

6. Open the Data Logging Utility.
7. Select in the DLG Utility the COMM port that the PC modem is configured for.
8. Click Connect.

Disconnect the Modem

1. Ensure the DLG Utility has been shut down.
2. Start HyperTerminal. Do not re-connect.
3. Open the previously configured "Datalog".
4. Type "**+++**" to place modem in command mode,

   Do not press the ENTER KEY!

   Your modem will respond: **OK**
5. Type "**ATH**".
6. Press Enter to send the Disconnect command to modem.

How to Use 40kHz PTOX/PWMX of MicroLogix 1400 Series B Controller

The PTOX and PWMX function files of MicroLogix 1400 Series B controller are changed to support 40kHz PTOX (Pulse Train Output) and PWMX (Pulse Width Modulation). In addition, a newer version of RSLogix 500 is released to support the changes. To guarantee seamless operation of MicroLogix 1400 Series B controller, special cares are required in handling some of PTOX and PWMX function file elements in user programs. In this section, detailed information regarding changes in PTOX and PWMX function files of MicroLogix 1400 Series B controller and how to handle PTOX and PWMX function file elements are described.

Basic requirements to use 40KHz PTOX and PWMX in MicroLogix Controller

MicroLogix 1400 Series A controller does not support 40kHz PTOX and PWMX. Only MicroLogix 1400 Series B controller supports 40kHz PTOX and PWMX with any version of RSLogix500.

IMPORTANT	When a user uses a prior version of RSLogix 500 (version 7.10 or lower) with MicroLogix 1400 Series B controller, maximum operating frequency of PTOX and PWMX is still 32767Hz. In order to use 40kHz PTOX and PWMX with a prior version of RSLogix 500, variable type change process is required (unsigned integer to signed integer).

PTOX and PWMX function file changes in Series B Controller

In a prior version of RSLogix500 (version 7.10 or lower), the frequency elements (PTOX.OF, PTOX.OFS, PTOX.JF, PWMX.OF, PWMX.OFS) of PTOX and PWMX function files are treated as signed 16-bit (-32768 ~ 32767) and MicroLogix 1400 Series A firmware didn't support values above 20000 (20kHz). However, these values are changed to unsigned 16-bit and 40000 (40kHz) respectively to implement 40kHz PTOX and PWMX functions in Series B controller.

Changes made in PTOX and PWMX function files of MicroLogix 1400 Series B controller and RSLogix 500 version 7.2

		RSLogix500 version 7.10 or lower		RSLogix500 version7.2 or higher	
		ML1400 Series A	**ML1400 Series B**	**ML1400 Series A**	**ML1400 Series B**
PTOX	OF	Signed 16-bit value : -32768~32767			Unsigned 16-bit value : 0~65535
	OFS	(F/W generates fault when it exceeds 20000)			(F/W generates fault when it exceeds 40000)
	JF				
PWMX	OF				
	OFS				

The newer version of RSLogix 500 and MicroLogix 1400 Series B handles PTOX and PWMX frequencies as unsigned 16-bit integer. Therefore, if an older version of RSLogix 500 is used with MicroLogix 1400 Series B controller or the newer version of RSLogix 500 is used with Series A controller, there occur compatibility issues.

- RSLogix 500 display issues : Since an older version of RSLogix 500 treats PTOX/PWMX frequencies as signed integers, if a user uploads a program from MicroLogix 1400 Series B controller into the software, a negative frequency value may be displayed. For example, 40kHz is 0x9C40 in hexadecimal. The 0x9C40 is 40000 in unsigned integer, but -25536 in signed integer.
- Instruction issues : All the instructions in MicroLogix controller treats integer as signed integer. Therefore, unsigned integers in PTOX/PWMX function file elements need to be carefully treated when they are used in the variables for instructions. An undesired result may be returned or math overflow may be detected during instruction executions. Workaround should be used such as copying PTOX/PWMX frequencies to long integer before manipulation.

RSLogix500 display issues

The difference of PTOX/PWMX between Series A and Series B is the maximum output frequency. The Series A supports up to 20kHz for PTOX/PWMX output frequency and the Series B supports up to 40kHz. In other words, Series A PTOX/PWMX output frequency range is in signed 16-bit range (-32768~32767) and Series B PTOX/PWMX output frequency range is in unsigned 16-bit range (0~65535). For this reason, there may be a display issue when displaying output frequency in RSLogix500. However, since an output frequency can not be a negative value, MicroLogix 1400 controller works properly although the displayed value looks weird.

The following table summarizes possible display issues for each different combination of a RSLogix 500 version and MicroLogix 1400 Series.

MicroLogix 1400 compatibility with RSLogix500

	ML1400 Project of RSLogix500 version 7.10 or lower	ML1400 Series A Project of RSLogix500 version 7.2 or higher	ML1400 Series B Project of RSLogix500 version 7.2 or higher
ML1400 Series A	✔	✔	(1)
ML1400 Series B	Display issues may occur.(2)	Display issues may occur.(2)	✔

(1) (During Donwload) If OF or JF value is set over 20000, then MicroLogix 1400 Series A will report an error before running the PTOX or PWMX instruction.

(2) (During Upload) To set the OF or JF value over 32767, a negative decimal value or hex value should be entered. For example, to set OF value as 40000, OF = -25536 (dec) or 9C40H (hex) should be entered. OFS value will be also displayed as a negative decimal value if an OFS value is over 32767. No problem with the operation although negative values are displayed.

New RSLogix500 compatibility with Old RSLogix500

	ML1400 Project of RSLogix500 version 7.10 or lower	ML1400 Series A Project of RSLogix500 version7.2 or higher	ML1400 Series B Project of RSLogix500 version7.2 or higher
Upload using RSLogix500 version 7.10 or lower	✔	✔	Display issues may occur.(1)
Upload using RSLogix500 version 7.2 or higher	✔	✔	✔

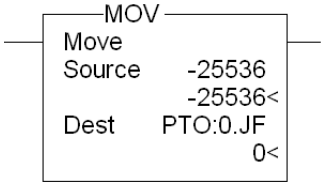
(1) If the values of OF, OFS, JF are greater than 32767, than these values will be displayed as negative value. For example : If the JF value is 40000 with Series B project of RSLogix500 version 7.2 or later, then -25536 will be displayed with RSLogix500 version 7.10 or lower. No problem with operation although negative values are displayed.

Instruction issues

There are some instruction issues to support maximum frequency of PTOX & PWMX up to 40kHz. Normally, MicroLogix and RsLogix500 treat data variables as signed value. When a user sets the frequency value that is greater than 32,767 using MOV, EQU, NEQ, LES, LEQ, GRT, GEQ, MEQ, LIM, ADD, SUB, MUL, DIV, NEG, ABS and SCP instructions, numerical issues may happen. 2's complement notation and hexadecimal values are useful to solve this issue. (See Number Systems on page 605.)

- MOV Instruction

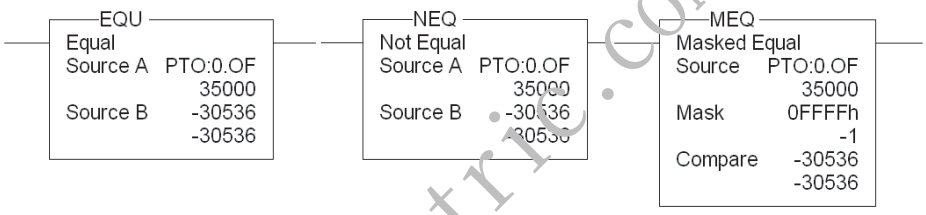
When setting JF, OF, OFS of the PTOX/PWMX function file using the MOV instruction, a user cannot enter a source operand value over 32,767 in decimal format because the operand format is 16-bit signed integer (-32,768...32,767) even though 16-bit unsigned integer (0...65,535) is functionally supported by PTOX/PWMX. To solve this issue, 2's complement notation or hexadecimal value should be used. For example, if a user wants to set the PTOX:0.JF to 40000, then put 2's complement of 40000 (-25536) or 9C40h (hexadecimal value) to source operand as shown below.



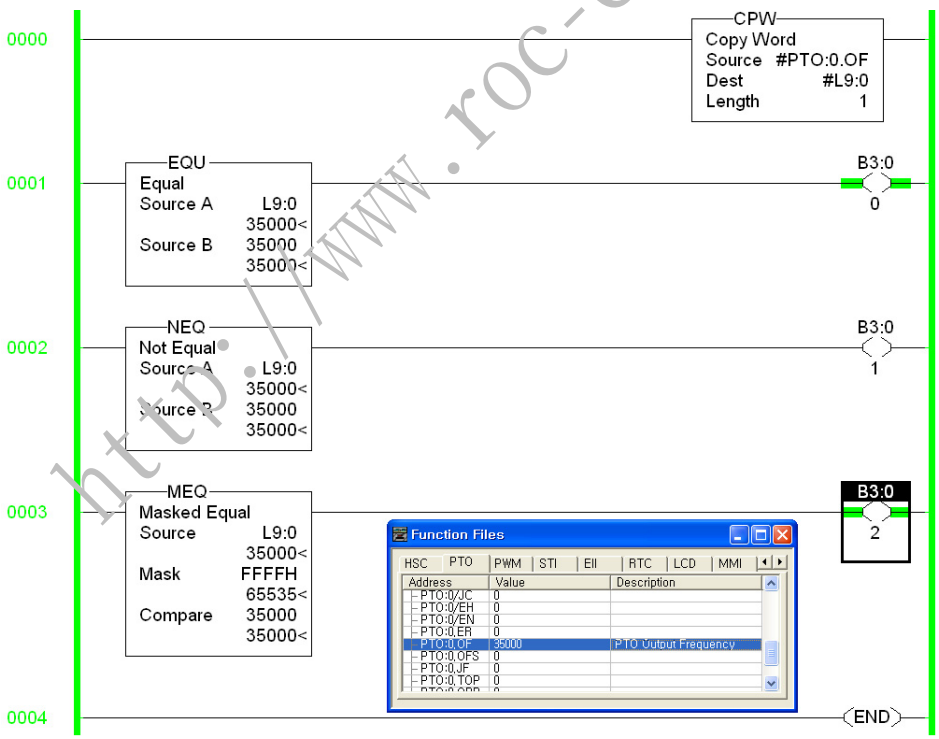
- EQU/NEQ/MEQ Instruction

When comparing the JF, OF, OFS of PTOX/PWMX with a specific value using the EQU, NEQ or MEQ instruction, a user can not put the specific value over 32767 because these variables are unsigned 16-bit value in this instruction. To solve this issue, 2's complement notation or hexadecimal value should be used.

For example, when a user wants to check if the PTOX:0.OF is equal(EQU) 35000 or not equal(NEQ), a user should put 2's complement of 35000 (-30536) or 88B8h (hexadecimal value) as a specific value as shown below.



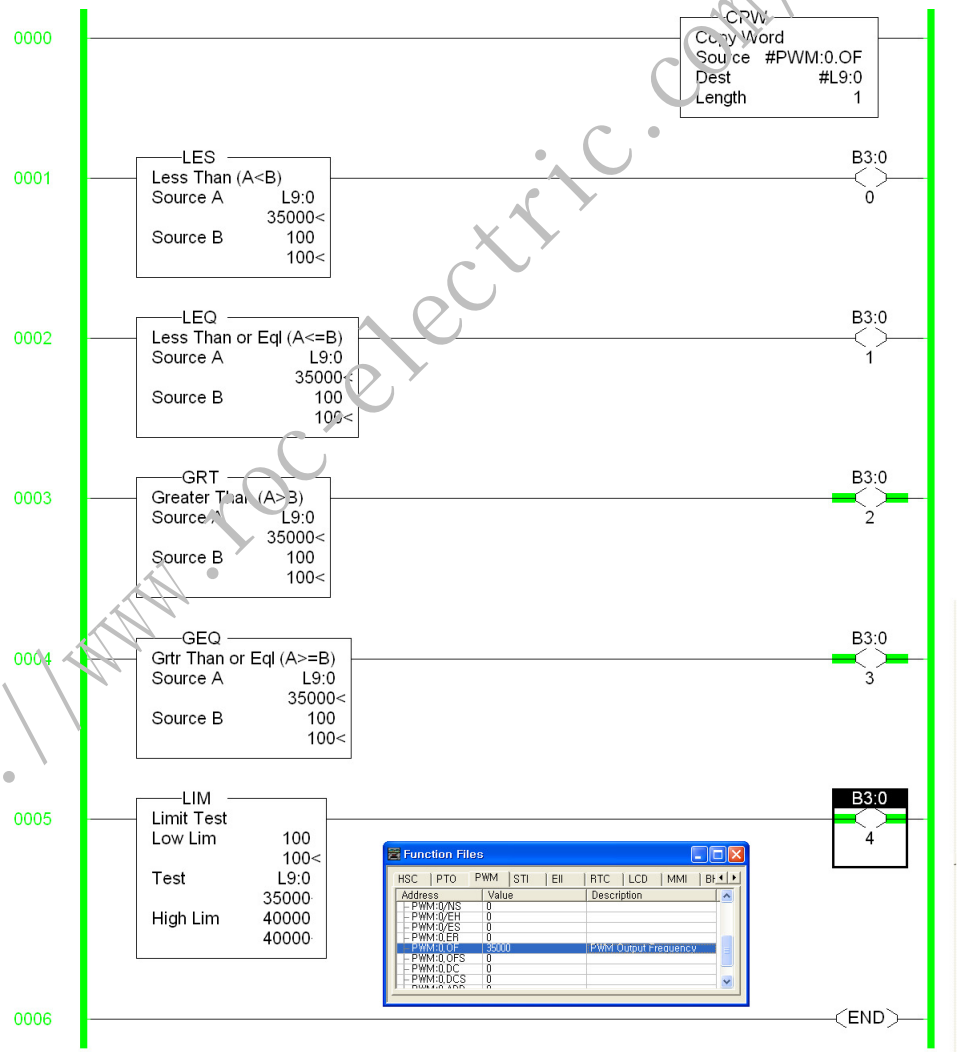
Otherwise, variable type should be changed to Long Type (32-bit) by CPW instruction before the execution of these instructions as shown below.



- LES/ LEQ/ GRT/ GEQ/ LIM Instruction

The operands of LES, LEQ, GRT, GEQ and LIM instruction work as signed value, but JF, OF, OFS of PTOX/PWMX are unsigned 16-bit variable at MicroLogix 1400 Series B controller. Therefore, these instructions could cause undesired results when JF, OF and OFS of PTOX/PWMX values are greater than 32767. For example, assume the PWMX:0.OF value is 35000 and the value to compare is 100. Actually, PWMX:0.OF is greater than 100, but the executed result of this instruction is just the opposite.

To solve this issue, variable type should be changed to Long Type (32-bit) by CPW instruction before the execution of these instructions as shown below.

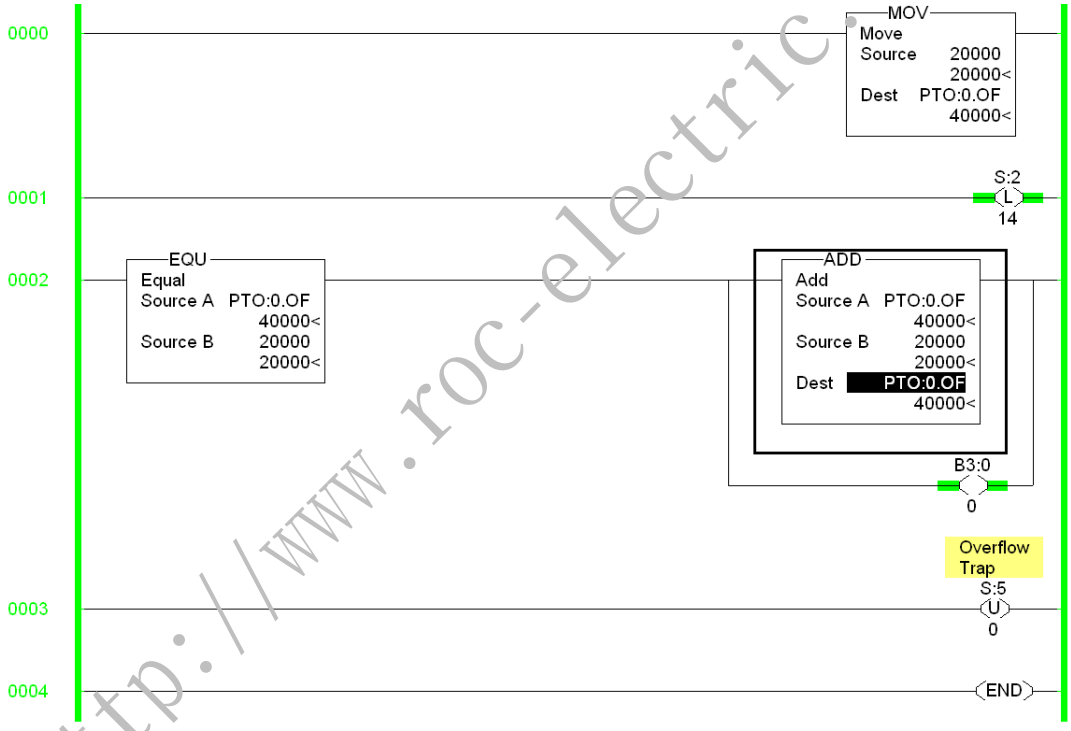


- ADD/ SUB/ MUL Instruction

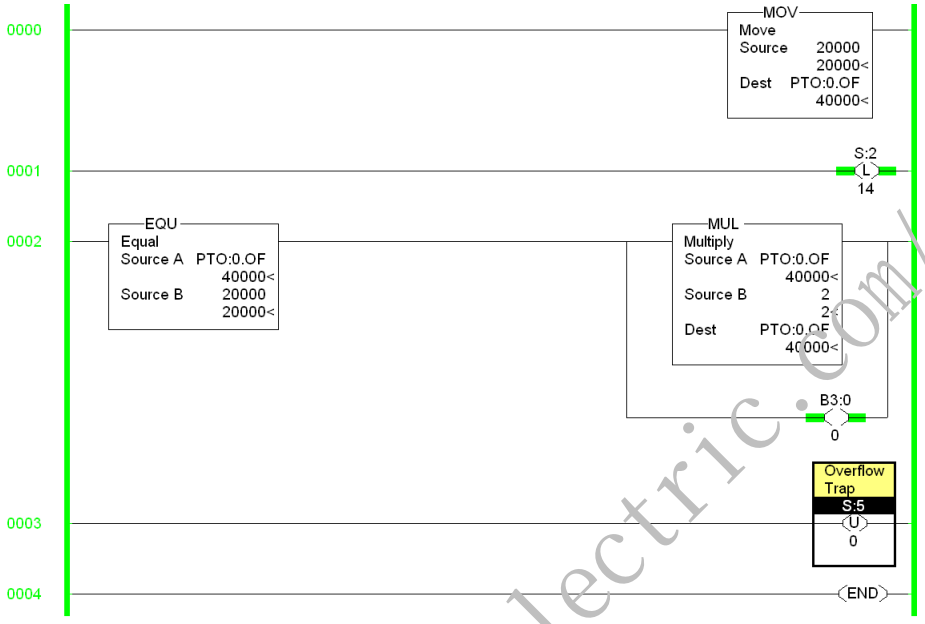
To get an unsigned result without Math Overflow Error from ADD/ SUB/ MUL instructions, S:2/14 (Math Overflow Selected) bit should be set and the S:5/0 (Math Overflow Trap) bit should be cleared after the execution of these instructions.

For example, assume the PTOX:0.OF value is 20000 and a user wants to set that value to x2 (twice) using the MUL instruction. If S2/14 is cleared and S:5/0 bit is not cleared after execution of the MUL instruction, then the result will be 32767 and Math Overflow Error will be reported at the End of Scan.

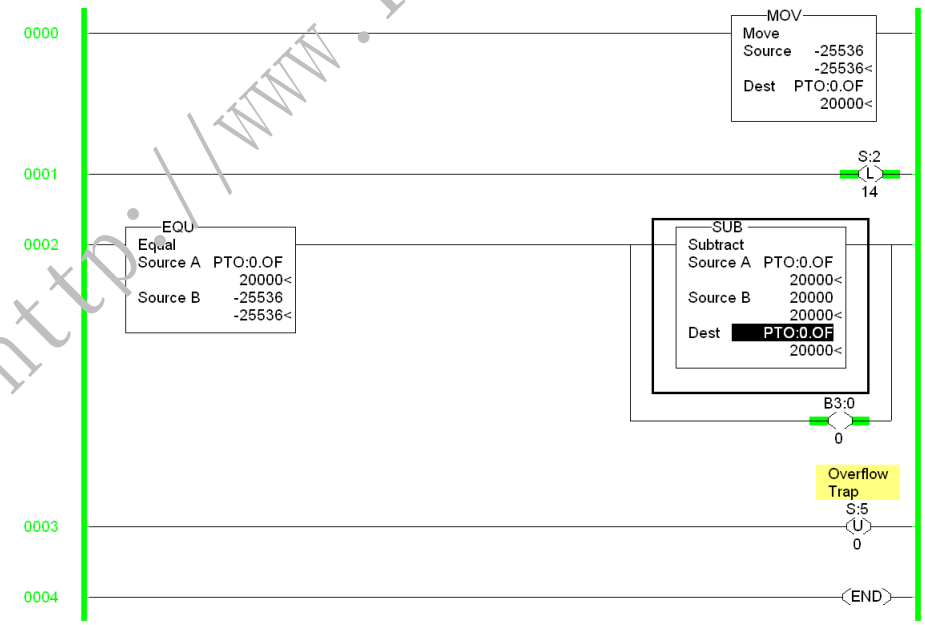
ADD Instruction Workaround Example: PTOX:0.OF(40000) = PTOX:0.OF(20000) + 20000



MUL Instruction Workaround Example: PTOX:0.OF(40000) = PTOX:0.OF(20000) X 2



SUB Instruction Workaround Example: PTOX:0.OF(20000) = PTOX:0.OF(40000) – 20000



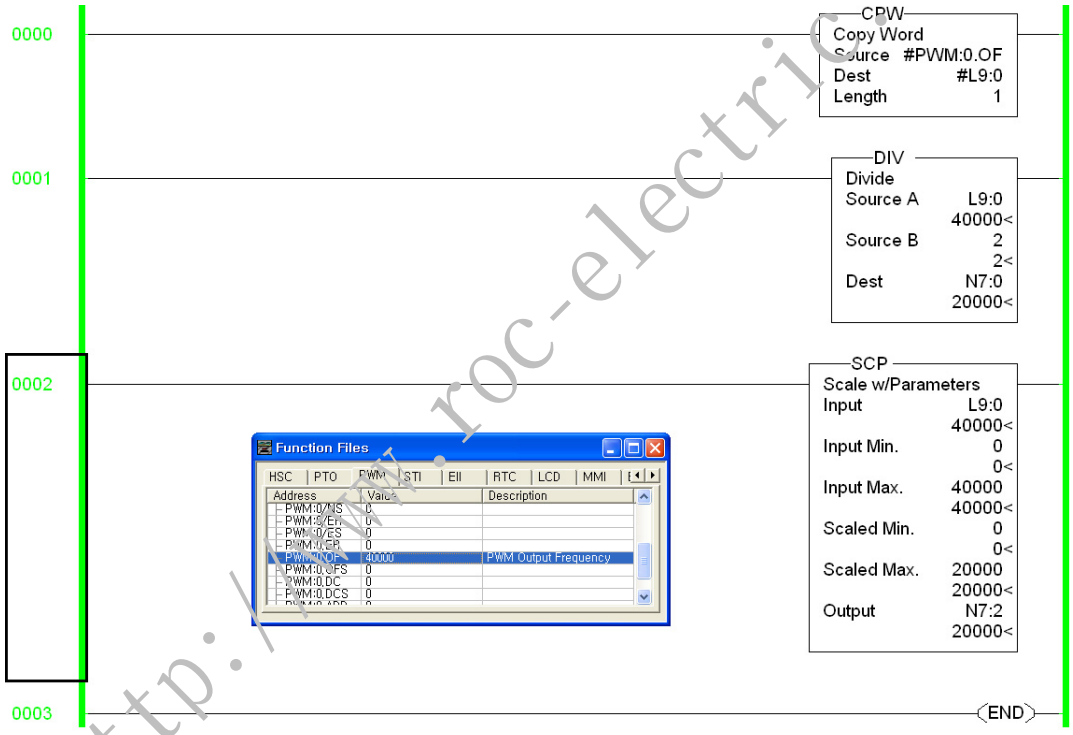
Signed 16-bit: 0x9C40 (Hexadecimal) = -25536,

Unsigned 16-bit: 0x9C40 (Hexadecimal) = 40000

- DIV/SCP Instruction

The operands of DIV and SCP instructions work as signed value, but JF, OF, OFS of PTOX/PWMX are unsigned 16-bit variable at MicroLogix 1400 Series B controller. Therefore, these instructions could cause undesired results when JF, OF and OFS of PTOX/PWMX values are greater than 32767. For example, assume the PWMX:0.OF value is 40000 (9C40h) and a user wants to divide it by 2. The expected result is 20000, but the actual return value is -12768 because DIV instruction recognize 9C40h as -25536.

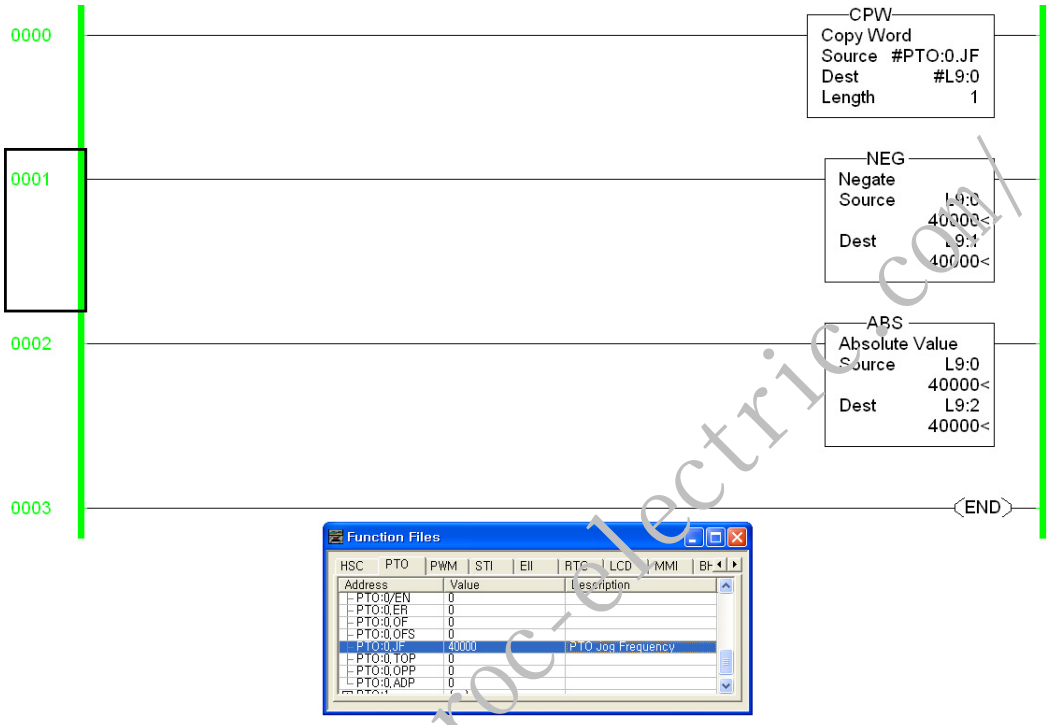
To solve this issue, JF, OF, OFS of PTOX/PWMX type should be changed to Long Type (32-bit) by CPW instruction before the execution of these instructions as shown below.



- NEG/ABS Instruction

The operands of NEG and ABS instructions work as signed value, but JF, OF, OFS of PTOX/PWMX are unsigned 16-bit variable at MicroLogix 1400 Series B. Therefore, these instructions could cause undesired results when JF, OF and OFS of PTOX/PWMX values are greater than 32767. If PTOX:0.JF is 40000, the expected NEG result is -40000 and the expected ABS result is 40000. However, the actual results are 25536 both because both NEG and ABS instruction recognize 9C40h as -25536.

To solve this issue, JF, OF, OFS of PTOX/PWMX type should be changed to Long Type (32-bit) by CPW instruction before the execution of these instructions as shown below.



Notes:

The following terms are used throughout this manual. Refer to the Allen-Bradley Industrial Automation Glossary, publication AG-7.1, for a complete guide to Allen-Bradley technical terms.

address

A character string that uniquely identifies a memory location. For example, I:1/0 is the memory address for data located in Input file word 1, bit 0.

AIC+ Advanced Interface Converter

A device that provides RS-232 isolation to an RS-485 Half-Duplex communication link. (Catalog Number 1761-NET-AIC.)

application

1) A machine or process monitored and controlled by a controller. 2) The use of computer- or processor-based routines for specific purposes.

ASCII

American Standard Code for Information Interchange. A standard for defining codes for information exchange between equipment produced by different manufacturers. The basis of character sets used in most microcomputers; a string of 7 binary digits represents each character.

baud rate

The speed of communication between devices. Baud rate is typically displayed in *K baud*. For example, 19.2K baud = 19,200 bits per second.

bit

The smallest unit of memory used in discrete or binary logic, where the value 1 represents ON and 0 represents OFF.

block diagrams

A method used to illustrate logic components or a sequence of events.

Boolean operators

Logical operators such as AND, OR, NAND, NOR, NOT, and Exclusive-OR that can be used singularly or in combination to form logic statements or circuits. Can have an output response of T or F.

branch

A parallel logic path within a rung of a ladder program. Its primary use is to build OR logic.

communication scan

A part of the controller's operating cycle. Communication with devices (such as other controllers and operator interface devices) takes place during this period.

control program

User logic (the application) that defines the controller's operation.

controller

A device, such as a programmable controller, used to control output devices.

controller overhead

A portion of the operating cycle used for housekeeping purposes (memory checks, tests, communications, etc.).

control profile

The means by which a controller determines which outputs turn on under what conditions.

counter

A device that counts the occurrence of some event.

CPU (Central Processing Unit)

The decision-making and data storage section of a programmable controller.

data table

The part of processor memory that contains I/O status and files where user data (such as bit, integer, timers, and counters) is monitored, manipulated, and changed for control purposes.

DIN rail

Manufactured according to Deutsche Industrie Normenausshus (DIN) standards, a metal railing designed to ease installation and mounting of your devices.

download

The transfer of program or data files to a device.

DTE

Data Terminal Equipment

EMI

Electromagnetic interference.

embedded I/O

Embedded I/O is the controller's on-board I/O. For MicroLogix controllers, embedded I/O is all I/O residing at slot 0.

expansion I/O

Expansion I/O is I/O that is connected to the controller via a bus or cable. MicroLogix 1100, 1200 and 1400 controllers use Bulletin 1762 expansion I/O. MicroLogix 1500 controllers use Bulletin 1769 expansion I/O. For MicroLogix controllers, embedded I/O is all I/O residing at slot 1 and higher.

encoder

A device that detects position, and transmits a signal representing that position.

executing mode

Any run or test mode.

false

The status of an instruction that does not provide a continuous logical path on a ladder rung.

FET

Field Effect Transistor. DC output capable of high-speed operation.

FIFO (First-In-First-Out)

The order that data is stored and retrieved from a file.

file

A collection of data or logic organized into groups.

full-duplex

A mode of communication where data may be transmitted and received simultaneously (contrast with half-duplex).

half-duplex

A mode of communication where data transmission is limited to one direction at a time.

hard disk

A storage device in a personal computer.

high byte

Bits 8 to 15 of a word.

housekeeping

The portion of the scan when the controller performs internal checks and services communications.

input device

A device, such as a push button or a switch, that supplies an electrical signal to the controller.

input scan

The controller reads all input devices connected to the input terminals.

inrush current

The temporary surge of current produced when a device or circuit is initially energized.

instruction

A mnemonic defining an operation to be performed by the processor. A rung in a program consists of a set of input and output instructions. The input instructions are evaluated by the controller as being true or false. In turn, the controller sets the output instructions to true or false.

instruction set

The set of instructions available within a controller.

I/O

Input and Output

jump

Changes the normal sequence of program execution. In ladder programs a JUMP (JMP) instruction causes execution to jump to a specific rung in the user program.

ladder logic

A graphical programming format resembling a ladder-like diagram. The ladder logic programing language is the most common programmable controller language.

least significant bit (LSB)

The element (or bit) in a binary word that carries the smallest value of weight.

LED (Light Emitting Diode)

Used as status indicator for processor functions and inputs and outputs.

LIFO (Last-In-First-Out)

The order that data is stored and retrieved from a file.

low byte

Bits 0 to 7 of a word.

logic

A general term for digital circuits or programmed instructions to perform required decision making and computational functions.

Master Control Relay (MCR)

A hard-wired relay that can be de-energized by any series-connected emergency stop switch.

mnemonic

A simple and easy to remember term that is used to represent a complex or lengthy set of information.

Modbus RTU Slave

A half-duplex serial communication protocol.

modem

Modulator/demodulator. Equipment that connects data terminal equipment to a communication line.

modes

Selected methods of operation. Example: run, test, or program.

negative logic

The use of binary logic in such a way that "0" represents the desired voltage level.

network

A series of stations (nodes) connected by some type of communication medium. A network may be made up of a single link or multiple links.

nominal input current

The typical amount of current seen at nominal input voltage.

normally closed

Contacts on a relay or switch that are closed when the relay is de-energized or deactivated. They are open when the relay is energized or the switch is activated.

normally open

Contacts on a relay or switch that are open when the relay is de-energized or the switch is deactivated. They are closed when the relay is energized or the switch is activated.

off-delay time

The OFF delay time is a measure of the time required for the controller logic to recognize that a signal has been removed from the input terminal of the controller. The time is determined by circuit component delays and by any applied filter.

offline

When a device is not scanning/controlling or when a programming device is not communicating with the controller.

offset

A continuous deviation of a controlled variable from a fixed point.

off-state leakage current

When a mechanical switch is opened (off-state), no current flows through the switch. Semiconductor switches and transient suppression components which are sometimes used to protect switches, have a small current flow when they are in the off state. This current is referred to as the off-state leakage current. To ensure reliable operation, the off-state leakage current rating must be less than the minimum operating current rating of the device that is connected.

on-delay time

The ON delay time is a measure of the time required for the controller logic to recognize that a signal has been presented at the input terminal of the controller.

one shot

A programming technique that sets a bit ON or OFF for one program scan.

online

When a device is scanning/controlling or when a programming device is communicating with the controller.

operating voltage

For inputs, the voltage range needed for the input to be in the On state. For outputs, the allowable range of user-supplied voltage.

output device

A device, such as a pilot light or a motor starter coil, that receives a signal or command from the controller.

output scan

The controller turns on, off, or modifies the devices connected to the output terminals.

PCCC

Programmable Controller Communications Commands

processor

A Central Processing Unit. (See CPU.)

processor files

The set of program and data files resident in the controller.

program file

Areas within a processor that contain the logic programs. MicroLogix controllers support multiple program files.

program mode

When the controller is not scanning the control program.

program scan

A part of the controller's operating cycle. During the program scan, the logic program is processed and the Output Image is updated.

programming device

Programming package used to develop ladder logic diagrams.

protocol

The rules of data exchange via communications.

read

To acquire data. For example, the processor reads information from other devices via a read message.

relay

An electrically operated device that mechanically switches electrical circuits.

relay logic

A representation of binary or discrete logic.

restore

To transfer a program from a device to a controller.

reserved bit

A location reserved for internal use.

retentive data

Information (data) that is preserved through power cycles.

RS-232

An EIA standard that specifies electrical, mechanical, and functional characteristics for serial binary communication circuits.

run mode

An executing mode during which the controller scans or executes the logic program.

rung

A rung contains input and output instructions. During Run mode, the inputs on a rung are evaluated to be true or false. If a path of true logic exists, the outputs are made true (energized). If all paths are false, the outputs are made false (de-energized).

RTU

Remote Terminal Unit

save

To save a program to a computer hard disk.

scan

The scan is made up of four elements: input scan, program scan, output scan, and housekeeping.

scan time

The time required for the controller to complete one scan.

sinking

A term used to describe current flow between two devices. A sinking device provides a direct path to ground.

sourcing

A term used to describe current flow between two devices. A sourcing device or circuit provides a power.

status

The condition of a circuit or system.

terminal

A point on an I/O module that external devices, such as a push button or pilot light, are wired to.

throughput

The time between when an input turns on and a corresponding output turns on or off. Throughput consists of input delays, program scan, output delays, and overhead.

true

The status of an instruction that provides a continuous logical path on a ladder rung.

upload

Data is transferred from the controller to a programming or storage device.

watchdog timer

A timer that monitors a cyclical process and is cleared at the conclusion of each cycle. If the watchdog runs past its programmed time period, it causes a fault.

write

To send data to another device. For example, the processor writes data to another device with a message write instruction.

A

ABL instruction *321*
ABS instruction *169*
absolute value instruction *169*
ACB instruction *322*
accuracy, timer *146*
ACI instruction *323*
ACL instruction *314*
ACN instruction *325*
active nodes status *516*
ADD instruction *167*
address *615*
Addressing
  considerations *543*
addressing
  direct addressing *71*
  immediate addressing *71*
  indirect addressing *72*
  indirect addressing of a bit *73*
  indirect addressing of a file *72*
  indirect addressing of a word *72*
  modes *71*
  using in-line indirection *334*
AEX instruction *326*
AHL instruction *327*
AIC instruction *316*
AIC+ Advanced Interface Converter *615*
Allen-Bradley
  contacting for assistance *532*
allow future access setting *34*
AND instruction *216*
application *615*
ARD instruction *328*
arithmetic flags *504*
ARL instruction *330*
ASC instruction *331*
ASCII
  definition *615*
ASCII character set *336*
ASCII clear buffers instruction *314*
ASCII control data file *313*
ASCII file *312*
ASCII handshake lines instruction *327*
ASCII instruction error codes *335*
ASCII instructions *309*
  error codes *335*
  status bits *312*, *313*
  timing diagram *334*
ASCII integer to string instruction *316*
ASCII number of characters in buffer instruction *322*
ASCII protocol parameters *311*
ASCII read characters instruction *328*
ASCII read line instruction *330*
ASCII string compare instruction *333*
ASCII string concatenate *325*
ASCII string extract *326*
ascii string manipulation error *515*
ASCII string search instruction *331*
ASCII string to integer instruction *323*
ASCII test buffer for line instruction *321*
ASCII timing diagram *334*
ASCII write instruction *319*
ASCII write with append instruction *316*
ASR instruction *333*
AWA and AWT timing diagram *334*
AWA instruction *316*
AWT instruction *319*

B

base hardware information file *44*
battery
  operation *40*
battery low status bit *514*
baud rate *615*
baud rate status *517*
BHI Function File *44*
bit *615*
bit instructions *137*
bit shift left instruction *229*
bit shift right instruction *231*
bit-wise AND instruction *216*
block diagrams *615*
Boolean operators *615*
branch *616*
BSL instruction *229*
BSR instruction *231*

C

carry flag *504*
catalog number status *522*
channel 0
  communications status *518*
  CS0 communications status file *45*
channel configuration
  DF1 full-duplex parameters *539*
  DF1 half-duplex parameters *545*, *548*

DF1 radio modem parameters *551*
DH485 parameters *537*
Modbus RTU Slave parameters *559*
clear instruction *168*
clearing
controller faults *525*
controller memory *33*
clock, free running *512*
CLR instruction *168*
common techniques used in this manual *xxix*
communication instructions *337*
communication protocols
DF1 full-duplex *539*
DF1 half-duplex *540*
DH485 *536*
Modbus Slave RTU *555*
communication scan *616*
communications
active status bit *519*
channel 0 status *518*
mode selection status bit *519*
status file *45*, *60*
compare instructions *155*
compiler revision
build number status *523*
release status *523*
contacting Rockwell Automation for assistance *532*
control profile *616*
control program *616*
control register error status bit *513*
controller
definition *616*
fault messages *526*
mode *509*
mode status *505*
overhead *616*
status file *503*
controller properties *25*
conversion instructions *205*
convert from binary coded decimal (BCD) instruction *208*
convert to binary coded decimal (BCD) instruction *211*
COP instruction *226*
copy file instruction *226*
copy word instruction *225*
count down instruction *153*
count up instruction *153*
counters
counter file *151*
counter file and status bits *152*
definition *616*
how counters work *151*
CPU (central processing unit), definition *616*
CPW instruction *225*
CS function file *45*, *60*
CTD instruction *153*
CTU instruction *153*

D

data file download protection *26*
data file overwrite protection lost status bit *520*
data files *21*, *26*
bit (B) *26*
control (R) *26*
counter (C) *151*
floating point (F) *26*, *164*
I/O images for expansion modules (MicroLogix 1200) *3*
input (I) *26*
input and output addressing examples *12*
integer (N) *26*
long word (L) *26*
message (MG) file *342*
organization and addressing *312*
output (O) *26*
PID (PD) *282*
programmable limit switch (PLS) *105*
protecting data files *26*
status (S) file *503*
string (ST) file *312*
timer (T) *145*
data logging *471*, *479*
Quick Start example *597*
data table *616*
DCD instruction *206*
decode 4 to 1-of-16 instruction *206*
DF1 full-duplex protocol *539*
configuration parameters *539*
description *539*
DF1 half-duplex protocol *540*
configuration parameters *545*, *548*
description *540*
DH485 communication protocol *536*
configuration parameters *537*
DH485 network
configuration parameters *537*
description *536*
protocol *536*
token rotation *536*
DIN rail *616*

DIV instruction *168*
divide instruction *168*
DLG
  Quick Start example *597*
DLG Instruction *478*
download *617*
DTE, definition *617*

E

EII function file *276*
embedded I/O *2*
EMI *617*
ENC instruction *206*
encode 1-of-16 to 4 instruction *206*
encoder
  definition *617*
  quadrature *96*
END instruction *256*
EQU instruction *156*
equal instruction *156*
error codes *525*, *526*
  ASCII instruction error codes *335*
  fault messages and error codes *525*
  HSC error codes *81*
  major error code status *515*
  math overflow trap bit *164*
  math status bits *163*
  MSG instruction error codes *404*
  PID runtime errors *297*
  PTOX error codes *127*
  PWMX error codes *135*
  STI error code *273*
  troubleshooting guide *526*
errors, identifying *525*
Ethernet
  Configuring an EtherNet/IP Message *373*
  DeviceNet and Ethernet Networks *381*
  Driver *567*
  Multi-hop Remote Message *385*
event input interrupt (EII) function file *276*
examine if closed instruction *137*
examine if open instruction *137*
example
  active station file *59*, *548*
  DLG Quick Start *597*
  HSC Quick Start *582*
  MSG Quick Start *586*
  PTOX Quick Start *577*
  PWMX Quick Start *580*
  RTC Quick Start *592*
  RTC Synchronization Quick Start *594*
  STI Quick Start *590*
  user interrupt disable (UID) Quick Start *593*
exclusive OR instruction *217*
executing mode *617*
execution time
  MicroLogix 1500 instructions *497*
expansion I/O *2*
  analog I/O configuration *6*
  discrete I/O configuration *3*

F

false *617*
fault messages *525*, *526*
fault override at power-up bit *506*
fault recovery procedure *526*
fault routine
  description of operation *265*
  file number status *517*
  manually clearing faults *526*
  operation in relation to main control program *263*
  priority of interrupts *265*
faults
  automatically clearing *525*
  identifying *525*
  manually clearing using the fault routine *526*
  recoverable and non-recoverable *265*
FET *617*
FFL instruction *233*
FFU instruction *235*
FIFO (First-In-First-Out) *617*
FIFO load instruction *233*
FIFO unload instruction *235*
file *617*
file instructions *225*
fill file instruction *228*
filtering, inputs *13*
first scan status bit *510*
FLL instruction *228*
forces enabled status bit *506*
forces installed status bit *506*
forcing, inputs and outputs *12*
FRD
  example *209*
  instruction *208*
free running clock *512*
free running clock status *512*
full-duplex *618*

function files *37*
base hardware information (BHI) *44*
communications status (CS) file *45*, *60*
event input interrupt (EII) *276*
high-speed counter (HSC) *78*
input/output status file (IOS) *67*
memory module information (MMI) *42*
pulse train output (PTO) *115*
pulse width modulation (PWM) *129*
real-time clock (RTC) *38*
selectable timed interrupt (STI) *272*
future access status bit *509*

G

GCD instruction *213*
GEQ instruction *157*
Gray code instruction *213*
greater than instruction *157*
greater than or equal to instruction *157*
GRT instruction *157*

H

half-duplex *543*, *618*
hard disk *618*
high byte *618*
high-speed counter
Quick Start example *582*
high-speed counter function file *78*
high-speed counter load instruction *103*
high-speed outputs *111*
housekeeping *618*
HSC
Quick Start example *582*
HSC function file *78*
HSL instruction *103*

I

I/O *619*
I/O configuration *1*
I/O forcing *12*
I/O refresh instruction *261*
identifying controller faults *525*
IIM instruction *259*
immediate input with mask instruction *259*
immediate output with mask instruction *260*
in-line indirection *334*
input and output instructions *259*
input device *618*
input filter selection modified status bit *515*
input filtering *13*
input scan *618*
input/output status file *67*
inrush current *618*
instruction *618*
instruction execution time *497*
instruction set
definition *618*
MicroLogix 1500 execution times *497*
overview *69*
INT instruction *267*
interrupt subroutine instruction *267*
interrupts
interrupt instructions *267*
interrupt subroutine (INT) instruction *267*
overview *263*
selectable timed start (STS) instruction *267*
user fault routine *265*
user interrupt disable (UID) instruction *268*
user interrupt enable (UIE) instruction *269*
user interrupt flush (UIF) instruction *270*
IOM instruction *260*
IOS function file *67*

J

JMP instruction *253*
JSR instruction *254*
jump *619*
jump to label instruction *253*
jump to subroutine instruction *254*

L

label instruction *254*
ladder logic *619*
last 100 µSec scan time status *519*
latching inputs *15*
LBL instruction *254*
LCD Function File *485*, *486*
BACKON *492*
BACKTIME *492*
CBL *488*
CNST *492*
DN *489*
ERN *489*
ERR *489*
ESC *492*
JOG *490*

OK *491*
POT0 *491*
POT1 *491*
SCD *488*
Sub-Elements *487*, *488*
TIF *490*
TO *488*
WND *491*
LCD Instruction *493*
Addressing Modes and File Types *493*
Default Values *493*
Displaying Special Characters *494*
Getting Value with Keypad *494*
Use *493*
LCD Overview *485*
least significant bit (LSB) *619*
LED (light emitting diode) *619*
LEQ instruction *157*
LES instruction *157*
less than instruction *157*
less than or equal to instruction *157*
LFL instruction *237*
LFU instruction *239*
LIFO (Last-In-First-Out) *619*
LIFO load instruction *237*
LIFO unload instruction *239*
LIM instruction *159*
limit instruction *159*
LN Insrtuction *189*
load memory module always bit *507*
load memory module on error or default program bit *507*
local messages *355*
LOG Insrtuction *190*
logic *619*
logical instructions *215*
logical NOT instruction *218*
logical OR instruction *217*
low byte *619*

M

major error code status *515*
major error detected in user fault routine status bit *514*
major error halted status bit *509*
manuals, related *xxx*
mask compare for equal instruction *158*
masked move instruction *221*
master control relay (MCR) *619*
master control reset instruction *256*
math instructions *161*, *199*
math overflow selection bit *511*
math register status *516*
maximum scan time status *517*
MCR instruction *256*
memory *21*
clearing controller memory *33*
memory mapping
MicroLogix 1200 I/O *3*
memory module boot status bit *514*
memory module compare bit *511*
memory module information function file *42*
fault override *43*
functionality type *42*
load always *44*
load on error *43*
mode behavior *44*
module present *42*
program compare *43*
write protect *43*
memory module password mismatch status bit *514*
memory usage
checking controller memory usage *25*
MicroLogix 1500 instructions *497*
MEQ *158*
MEQ instruction *158*
message
Quick Start example *586*
message (MG) file *342*
message errors *404*
message instruction *341*
message reply pending status bit *518*
messages
local *355*
local messaging examples *365*
remote *379*
messaging
remote station-to-remote station *543*
messaging overview *337*
MicroLogix 1400 scan time example *501*
minor error bits *513*
MMI function file *42*
mnemonic *619*
Modbus definition *620*
Modbus RTU protocol *555*
Modbus TCP *421*
Modbus to MicroLogix memory map *561*, *562*, *563*, *564*
mode behavior *509*
mode status *505*
modem *620*

modes *620*
monitoring controller operation, fault recovery procedure *526*
MOV instruction *219*
move instructions *219*
MSG
  Quick Start example *586*
MSG instruction *341*
  error codes *404*
  ladder logic *353*
  local messaing examples *365*
  timing diagram *348*
MUL instruction *168*
multiply instruction *168*
MVM instruction *221*

N

NEG instruction *168*
negate instruction *168*
negative logic *620*
NEQ instruction *156*
network *620*
node address status *517*
nominal input current *620*
normally closed *620*
normally open *620*
not equal instruction *156*
NOT instruction *218*

O

OEM lock *34*
OEM lock status bit *509*
offline *620*
offset *621*
off-state leakage current *621*
one shot *621*
one shot falling instruction *142*
one shot instruction *141*
one shot rising instruction *142*
online *621*
ONS instruction *141*
operating system
  catalog number status *522*
  FRN status *522*
  series letter status *522*
operating voltage *621*
OR instruction *217*
OSF instruction *142*
OSR instruction *142*
OTE instruction *139*
OTL instruction *140*
OTU instruction *140*
outgoing message command pending status bit *519*
output device *621*
output instruction *139*
output latch instruction *140*
output scan *621*
output unlatch instruction *140*
overflow flag *504*
overflow trap status bit *513*

P

PCCC *621*
PD data file *282*
PID
  analog I/O scaling *298*
  application examples *303*
  application notes *299*
  errors *297*
  PID concept *281*
  PID equation *282*
  PID instruction *283*
  tuning parameters *288*
PLS file *105*
Polled report-by-exception, defined *543*
power-up mode behavior bit *507*
process control instruction *281*
processor *621*
processor battery low status bit *514*
processor catalog number status *522*
processor files *622*
processor revision status *523*
processor series status *523*
program control instructions *253*
program end instruction *256*
program file
  definition *622*
program mode *622*
program scan
  definition *622*
  MicroLogix 1500 scan time worksheet *501*
programmable limit switch *77*, *105*
programmable limit switch file *105*
programming device *622*
programming instructions *69*
proportional integral derivative
  application notes *299*
  PID instruction *283*

PID tuning *303*
runtime errors *297*
the PID concept *281*
the PID equation *282*
protocol *622*
DF1 full-duplex *539*
DF1 half-duplex *540*
DF1 radio modem *549*
DH485 communication *536*
Modbus RTU *555*
protocol configuration *535*, *577*, *605*
PTO
function file *115*
instruction *111*
Quick Start example *577*
publications, related *xxx*
pulse train output
function file *115*
instruction *111*
Quick Start example *577*
pulse width modulation
function file *129*
instruction *128*
Quick Start example *580*
Purpose of this Manual *xxix*
PWM
function file *129*
instruction *128*
Quick Start example *580*

Q

quadrature encoder *96*
queue *465*

R

RAC instruction *104*
RAD instruction *187*
RCP instruction *465*
read *622*
real time clock
accuracy *40*
battery low indicator bit *40*
disabling *39*
function file *38*
real-time clock
Quick Start example *592*
real-time clock adjust instruction *42*
recipe *465*
recipe instruction *465*
REF instruction *261*
refresh instruction *261*
related publications *xxx*
relay *622*
relay logic *622*
relay-type instructions *137*
remote messages *379*
remote packet support *538*
RES instruction *154*
reserved bit *622*
reset accumulated value instruction *104*
reset instruction *154*
restore *622*
RET instruction *255*
retentive data *623*
retentive data lost status bit *514*
retentive timer on delay instruction *150*
return from subroutine instruction *255*
RS-232, definition *623*
RTA instruction *40*
RTC
day of month status *520*
day of week status *522*
function file *38*
hours status *521*
minutes status *521*
month status *520*
Quick Start example *592*
seconds status *521*
year status *520*
RTC Synchronization
Quick Start example *594*
RTC synchronization
Quick Start example *594*
RTO instruction *150*
RTU, definition *623*
run mode *623*
rung *623*

S

save *623*
SBR instruction *254*
scale instruction *170*
scale with parameters instruction *171*
scan *623*
scan time *623*
last 100 µSec scan time status *519*
maximum scan time status *517*

scan toggle status bit *519*
SCL instruction *170*
SCP instruction *171*
selectable timed interrupt
  Quick Start example *590*
selectable timed interrupt (STI) function file *272*
selectable timed start instruction *267*
sequencer compare instruction *243*
sequencer instructions *243*
sequencer load instruction *249*
sequencer output instruction *246*
service communications instruction *339*
sign flag *505*
SIN instruction *173*
sinking *623*
SLC 5/03,5/04, and 5/05
  Active stations, monitoring *547*
  Channel Status *546*
sourcing *623*
SQC instruction *243*
SQL instruction *249*
SQO instruction *246*
SQR instruction *173*
square root instruction *173*
startup protection fault bit *506*
static file protection *28*
Station addresses
  defining *543*
Station list
  viewing *547*
status *624*
status file *503*
STI
  enabled bit *510*
  executing bit *510*
  file number status *518*
  function file *272*
  lost status bit *514*
  mode status *510*
  pending status bit *510*
  Quick Start example *590*
  set point status *518*
string data file *312*
STS instruction *267*
SUB instruction *167*
subroutine label instruction *254*
subtract instruction *167*
SUS instruction *255*
suspend code status *515*
suspend file status *516*
suspend instruction *255*
SVC instruction *339*
swap instruction *241*
SWP instruction *241*

T

TAN instruction *177*
temporary end instruction *255*
terminal *624*
throughput *624*
timer accuracy *146*
timer and counter instructions *145*
timer files *145*
timer off-delay instruction *149*
timer on-delay instruction *148*
timing diagrams
  ASCII *334*
  AWA and AWT instructions *334*
  latching inputs *15*
  MSG instruction *348*
  PTOX relative timing *113*
  quadrature encoder *96*
TND instruction *255*
TOD instruction *211*
  changes to the math register *212*
  example *212*
TOF instruction *149*
TON instruction *148*
troubleshooting *526*, *532*
  automatically clearing faults *525*
  contacting Allen-Bradley for assistance *532*
  identifying controller faults *525*
  manually clearing faults *526*
  using the fault routine *526*
true *624*

U

UID
  Quick Start example *593*
UID instruction *268*
UIE instruction *269*
UIF instruction *270*
upload *624*
user application mode status *505*
user fault routine
  creating a user fault routine *266*
  file number status *517*

major error detected status bit *514*
recoverable and non-recoverable faults *265*
user interrupt disable instruction *268*
user interrupt enable instruction *269*
user interrupt flush instruction *270*
user memory *23*
user program functionality type status *523*

W

watchdog scan time *512*
write *624*

X

XIC instruction *137*
XIO instruction *137*
XOR instruction *217*
XPY instruction *193*

Z

zero flag *505*

Instruction List

Instruction- Description	Page
ABL - Test Buffer for Line	321
ABS - Absolute Value	169
ACB - Number of Characters in Buffer	322
ACI - String to Integer	323
ACL - ASCII Clear Buffers	314
ACN - String Concatenate	325
ACS - Arc Cosine	181
ADD - Add	167
AEX - String Extract	326
AHL - ASCII Handshake Lines	327
AIC - ASCII Integer to String	316
AND - Bit-Wise AND	216
ARD - ASCII Read Characters	328
ARL - ASCII Read Line	330
ASC - String Search	331
ASN - Arc Sine	179
ASR - ASCII String Compare	333
ATN - Arc Tangent	183
AWA - ASCII Write with Append	316
AWT - ASCII Write	319
BSL - Bit Shift Left	229
BSR - Bit Shift Right	231
CLR - Clear	168
COP - Copy File	226
COS - Cosine	175
CPT - Compute	196
CPW - Copy Word	225
CTD - Count Down	153
CTU - Count Up	153
DCD - Decode 4 to 1-of-16	206
DEG - Radians to Degrees	185
DIV - Divide	168
DLG - Data Log Instruction	478
ENC - Encode 1-of-16 to 4	206
END - Program End	256
EQU - Equal	156
FFL - First In, First Out (FIFO) Load	233
FFU - First In, First Out (FIFO) Unload	235
FLL - Fill File	228
FRD - Convert from Binary Coded Decimal (BCD)	208
GCD - Gray Code	213
GEQ - Greater Than or Equal To	157
GRT - Greater Than	157
HSL - High-Speed Counter Load	103
IIM - Immediate Input with Mask	259
INT - Interrupt Subroutine	267
IOM - Immediate Output with Mask	260
JMP - Jump to Label	253
JSR - Jump to Subroutine	254

Instruction List

Instruction- Description	Page
LBL - Label	254
LCD - LCD Information	485
LEQ - Less Than or Equal To	157
LES - Less Than	157
LFL - Last In, First Out (LIFO) Load	237
LFU - Last In, First Out (LIFO) Unload	239
LIM - Limit Test	159
LN - Natural Log	189
LOG - Base 10 Logarithm	190
Memory Module Information Function File	42
MCR - Master Control Reset	256
MEQ - Mask Compare for Equal	158
MOV - Move	219
MSG - Message	341
MVM - Masked Move	221
MUL - Multiply	168
NEG - Negate	168
NEQ - Not Equal	156
NOT - Logical NOT	218
ONS - One Shot	141
OR - Logical OR	217
OSF - One Shot Falling	142
OSR - One Shot Rising	142
OTE - Output Energize	139
OTL - Output Latch	140
OTU - Output Unlatch	140
PID - Proportional Integral Derivative	283
PTO - Pulse Train Output	111
PWM – Pulse Width Modulation	128
RAC - Reset Accumulated Value	104
RAD - Degrees to Radians	189
RCP - Recipe	465
REF- I/O Refresh	261
RES - Reset	154
RET - Return from Subroutine	255
RHC - Read High Speed Clock	199
RPC - Read Program Checksum	201
RTA - Real Time Clock Adjust Instruction	40
RTO - Retentive Timer, On-Delay	150
SBR - Subroutine Label	254
SCL - Scale	170
SCP - Scale with Parameters	171
SIN - Sine	173
SQC- Sequencer Compare	243
SQL - Sequencer Load	249
SQO- Sequencer Output	249
SQR - Square Root	173
STS - Selectable Timed Start	267
SUB - Subtract	167
SUS - Suspend	255

Instruction List

Instruction- Description	Page
SVC - Service Communications	339
SWP - Swap	241
TAN - Tangent	177
TDF - Compute Time Difference	202
TND - Temporary End	255
TOD - Convert to Binary Coded Decimal (BCD)	211
TOF - Timer, Off-Delay	149
TON - Timer, On-Delay	148
UID - User Interrupt Disable	268
UIE - User Interrupt Enable	269
UIF - User Interrupt Flush	270
XIC - Examine if Closed	137
XIO - Examine if Open	137
XOR - Exclusive OR	217
XPY - X Power Y	193

Function File List

Function File- Description	Page
BHI - Base Hardware Information	44
CS - Communications Status	45
EII - Event Input Interrupt	276
ES - Ethernet Status	60
HSC - High Speed Counter	78
IOS - I/O Status	67
LCD-LCD Information	485
MMI - Memory Module Information	42
PTOX - Pulse Train Output	115
PWMX - Pulse Width Modulation	129
RTC - Real Time Clock	38
STI - Selectable Timed Interrupt	272

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products.
At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1766-RM001G-EN-P - March 2017
Supersedes Publication 1766-RM001F-EN-P - May 2014