SIEMENS

SIMATIC

C7-621, C7-621 AS-i Control Systems

Volume 2 Working with C7

Manual

C79000-G7076-C622-01

Contents

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Danger

indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.

Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this manual.

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:

Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

SIMATIC® and SINEC® are registered trademarks of SIEMENS AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Siemens AG
Bereich Automatisierungssystem
Industrial Automation Systems
Postfach 4848, D-90327 Nuernberg

Contents

Introduction

1

In this Chapter

This chapter explains what you require to program the C7 and how you can use the C7 as an operator control and monitoring device.

Note

The C7 consists of two independent units.
- C7 CPU with C7 AS-i CP (in the C7-621 AS-i) and
- C7 OP

Where necessary, these components are dealt with separately.

What You Require to Operate the C7

You require the following equipment and tools:

- A programming device or PC with an MPI interface and a programming device cable
- STEP 7, Version 5.1 or higher and the relevant documentation
- The ProTool or ProTool/Lite configuration tool, Version 2.51 or higher and the relevant documentation
- C7 connector set for I/Os and power supply

Chapter Overview

1.1 Controlling with the C7

Overview

The user program runs on the C7 CPU. This controls the process to be visualized by the operator control and monitoring components of the C7.

C7 CPU

The way in which the C7 CPU operates is determined by the following components:

Program Memory
This contains the user program

Processor
The processor executes the program cyclically:

- At the beginning of the cycle, the processor reads the signal states of all inputs and creates the process image input table (PII)
- The program is executed step-by-step while referencing internal counters, memory bits and timers.
- The processor stores the calculated signal states in the process image output table (PIQ). The signal states are then transferred to the outputs.

The C7 CPU is independent of the C7 OP. The C7 CPU has its own MPI address and is connected to the C7 OP via the MPI interface.

Programming Languages

At present, there are three standard programming languages available for programming the C7 CPU:

- STL: A statement list consists of a series of statements. Each statement in your program contains instructions in the form of mnemonics which represent a function for the C7 CPU to perform.
- LAD: A ladder diagram is a graphical program language that resembles electrical circuit diagrams.
- FBD: A function block diagram is a graphical programming language representing the logic in logic boxes familiar from Boolean algebra.

Other optional programming languages include SCL and HiGraph.

Programming Software

The tool you use to create user programs is STEP 7. The user manual **/231/** contains all the instructions for programming. When programming in a specific language, use the manual listed in the preface of /231/.

Devices

STEP 7 runs on a programming device or PC. You can operate these devices independent of the C7. You generally only connect the programming device or PC to the C7 via the MPI interface when you want to download your user program to the C7 CPU.

1.2 C7 AS-i CP in the C7-621 AS-i

Overview

The AS interface or actuator-sensor interface is a communication system for binary sensors and actuators at the lowest field level. Process signals generated in the field are normally transferred to the controller using a large amount of parallel wiring and input/output modules. This means that every sensor or actuator in the field is connected to the input/output modules with its own separate cable.

With the AS interface, this cable harness can be replaced by a simple common 2-wire cable for all sensors and actuators.

The technical data and ordering information for AS-i actuators and sensors is listed in the Siemens Catalog ST PI.

How the C7 AS-i CP Functions

The main characteristics of the AS-i system are as follows:

- Master-slave access technique
- Electronic address setting
- Operating reliability and flexibility
- 2-wire cable (unshielded) for data and power supply
- Tree configuration with cable lengths up to 100 m
- Direct integration
- Increased functionality, greater use for the customer
- Additional power supply when more power is required

Numeric Data

- Cycle time maximum 5 ms with 31 slaves
- Maximum of 31 slaves
- Maximum of 248 binary I/Os (124 inputs, 124 outputs)

C7 AS-i CP Modes

The AS-i CP supports two types of operation:

Standard Operation

Standard operation allows particularly simple installation and programming of the C7 AS-i CP.

In standard operation, the C7 AS-i CP behaves like an I/O module. It occupies 16 input and 16 output bytes in the analog area of the controller. In this operation, the slaves are assigned the default value for the parameters (F_H) saved on the CP. Parameters and commands cannot be transferred in standard operation.

Extended Operation

In extended operation, you can use the complete range of functions as described in the AS-i master specification. This type of operation is supported by a function (FC) that is supplied on a diskette with this manual. In extended operation using the FC, you can also implement master calls in the user program.

1.3 Operator Control and Monitoring with C7

Overview

Using the C7, you can visualize operating states and current process values. You can also enter information at the C7 that is written to the C7 CPU. You can also configure functions for machine diagnostics on the C7.

The C7 provides a series of standard functions. You can adapt the displays and the operation of the C7 to the particular requirements of your process.

C7 OP

The C7 OP processes the operator control and monitoring functions of the C7. It is independent of the C7 CPU and continues to operate, for example, if the C7 CPU changes to the STOP mode. The C7 OP has its own MPI address and is connected to the C7 CPU via the MPI interface. This MPI interface also provides the link between the C7 OP and a configuration computer (programming device/PC).

Operator Control and Monitoring Functions

The basic functions of a C7 are to display process states and control the process. The operator control and monitoring is flexible and is selected by the user and downloaded to the C7. The following display and operator functions can be configured for the C7:

- Screens
- Event messages
- Multi-language operator menus

Screens

Logically related process data from the control system can be displayed together in one screen and modified individually as required. A screen consists of several screen entries since, for example, the description of a machine state generally requires more related data than can be represented in one section of the display. This allows, for example, data about the operating temperature, tank level, rotational speed and run time to indicate the current machine state.

The C7-621 and C7-621 AS-i systems have line-oriented displays. A screen consists of text elements made up of static text and current (dynamic) status values.

Event Messages

Event messages provide information and instructions for the operator relating to current machine or process states during normal operation. Event messages can contain process values. The representation of the process values can be either numeric or symbolic.

System Messages

System messages display internal states of the C7. They indicate, for example, operator errors or problems in communication. This type of message has the highest display priority. If a problem occurs on the C7, the currently displayed event message is cleared and a system message is displayed.

Languages

Message texts, screens, information texts and system messages can be displayed in several languages. A maximum of three of the languages listed below can be loaded simultaneously and selected by the operator online:

- German
- English
- French
- Italian
- Spanish

Configuration / Process Management

Before a C7 can be put into operation, it must be prepared so that it can visualize data from the C7 CPU. This preparation is known as configuration.

You configure the C7 OP using a computer (programming device/PG) with the ProTool/Lite configuration software.

Configuration can be divided into three parts:

- Control data
- Assignment of parameters for the interface
- Language selection

Once this configuration has been created, it is downloaded to the C7 OP. For this step, the computer must be connected to the C7 via an MPI interface.

Once the configuration has been downloaded to the C7 OP, the process management phase can be started after running a restart on the OP. The C7 OP then reacts to process signals from the C7 CPU or to operator input based on the configuration that has been loaded on it.

For information about configuring the C7 OP, refer to the user manuals for *ProTool* or *ProTool/Lite*.

1.4 C7 Overview

The SIMATIC C7-621/C7-621 AS-i units consist of several components that interact with each other:

- A CPU of the SIMATIC S7-300 class (C7 CPU)
- A line-oriented SIMATIC OP (C7 OP)
- Integrated digital and analog I/Os (C7-621 I/Os)
- A P bus attachment for expanding the C7-621 with S7-300 modules via the IM 621
- An MPI interface for communication with the programming device/PC and other S7-CPUs, C7 control systems and OPs
- An AS-i interface (AS-i) for connecting sensors and actuators (with the C7-621 AS-i version, see Figure 1-2)

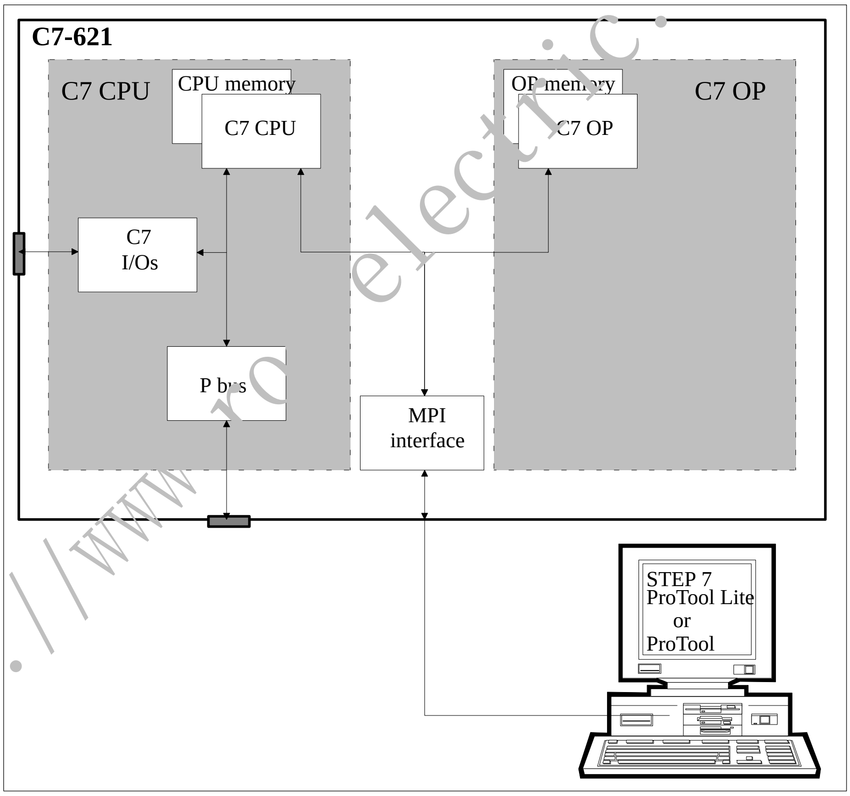


Figure 1-1 Components of the C7-621

The individual components integrated in the SIMATIC C7 control system correspond to the components that can also be used in modular configurations consisting of an S7-300 CPU, OP etc. The I/O expansion via the P bus interface (IM 621) allows the connection of a maximum of 4 SIMATIC S7-300 modules. The AS-i interface allows the connection of sensors and actuators to the C7-621 AS-i (Figure 1-2).

The general functions also correspond to a configuration with standard modules from the PLC and OP ranges. The individual components operate independently of each other and each processor has its own memory.

The C7 CPU is programmed with STEP 7 and the C7 OP is configured with ProTool/Lite. The software runs under Windows 95 on a programming device or PC.

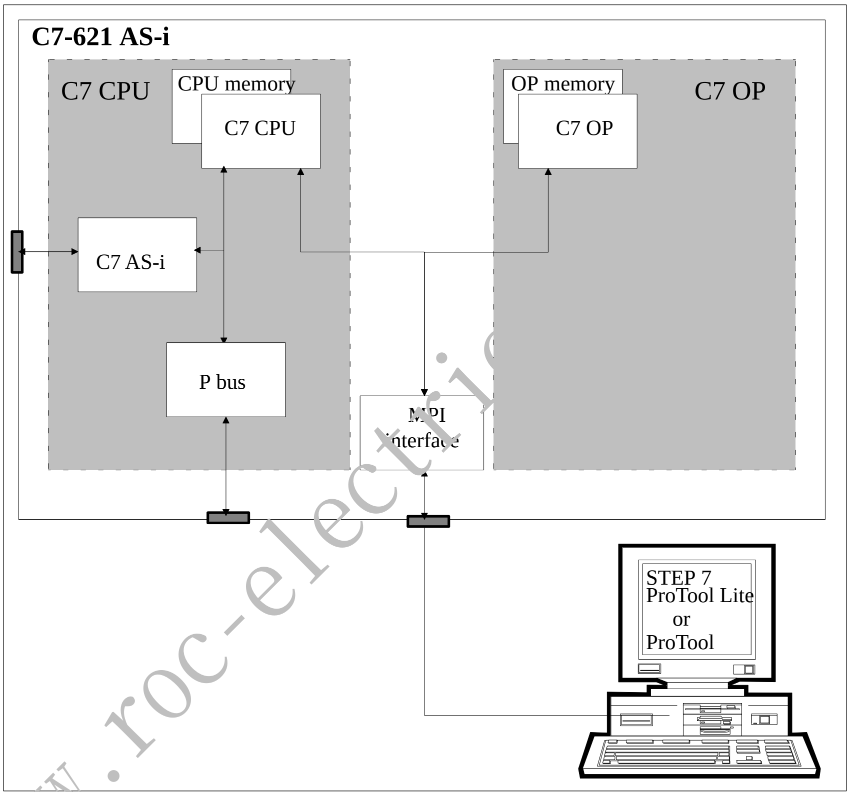


Figure 1-2 Components of the C7-621 AS-i

Startup

2

In this Chapter

This chapter explains the following:

- How the C7 starts up
- What you need to do if no configuration has been downloaded and when a configuration has been downloaded
- How to activate the C7 CPU modes RUNP, RUN, STOP and MRES
- How to activate the DI/DO status display
- How to reset the memory on the C7

Chapter Overview

2.1 Starting Up

Startup

After connecting the power supply, the C7 performs a self test. During this test, it checks the functionality of the most important device components and indicates the test results with the status LEDs and on the display. The startup procedure is as follows:

1. The C7 runs a self test after POWER ON.
2. The C7 runs an operating system test for both sections (C7 CPU and C7 OP). If there is already a user program in the flash memory, the C7 CPU loads this program into the work memory.
3. During startup (1. and 2.), the C7 CPU remains in the STOP mode.

   After startup, the C7 OP is in the message mode.
   The following standby message is then displayed:

```
SIMATIC C7     Vx.xx
C7-621
```

Figure 2-1 C7-621 Standby Message

4. After starting up, the last selected C7 CPU mode is valid.

   How you proceed from here depends on the situation on the C7:

   - **No control program** loaded. The control program must be loaded.
   - A **control program is loaded**.
   - **No configuration** loaded.
   - A **configuration is loaded**.

   The following sections explain what to do in each of the situations.

Load Control Program

To be able to control and monitor the process, the C7 OP must be able to access data on the C7 CPU. This means that you must first load a user program if no user program is present. To load a user program, follow the steps below:

1. Set the C7 CPU to the STOP mode (see Section 2.2).
2. Activate the transfer of the user program and data blocks on your programming device/PC using STEP 7.
3. Start the copy function on the programming device/PC.

No OP Configuration Loaded

If no configuration has been loaded, the screens in the firmware are activated in English. In this case you select the **C7 OP mode** in which you can transfer a configuration. To do this, follow the steps outlined below:

Step	Activity	Result
1.	Select the configuration permanently stored on the C7 by pressing ENTER	`Screens System` `| | >>`
2.	Select `System` with F4	`OpMode Language` `| | >>`
3.	Select `OpMode` with F2	`Operat. mode Online`

4.	Select the `Transfer` function by pressing the two keys simultaneously SHIFT ▽
	
	 Start the transfer with ENTER
		`Operat. mode MPIDown`
		
		 `Ready for transfer`
		
		 When you transfer the configuration, it is loaded via the MPI interface.

5.	Now activate the transfer of the configuration to the C7 on your programming device/PC using "ProTool/Lite".	A soon as the configuration is loaded, the message mode is set and the configured standby message is displayed.
6.	You can now change to the following modes • C7 system function menu by pressing the two keys simultaneously SHIFT 0 • Screens by pressing ENTER	`SIMATIC C7 Vx.xx` `C7-621`

OP Configuration Loaded

If a configuration has already been loaded, you can start this by pressing the ENTER key and the configured "Start Screen" is displayed.

Reloading an OP Configuration

If a configuration is already loaded, you can download a new configuration following the same procedure as explained in “No Configuration Loaded”.

If the standard screen or the function mentioned above is not available in the currently loaded configuration, you must remove the existing configuration using the memory reset function before you can load a new configuration. For detailed information about the memory reset function, refer to Section 2.4.

To reset the memory on the OP:

1. Turn off the C7.
2. Press and hold the keys [▽] [▷] [ESC] simultaneously.
3. Turn on the power for the C7.
4. The remaining steps are analogous to “No Configuration Loaded”.

2.2 Selecting the C7 CPU Mode

System Functions Menu

You can select the system function menu in any mode on the C7-621 and C7-621 AS-i. When this menu is displayed, you can select the following menus or functions:

- C7 CPU mode selection F1
  - RUN-P
  - RUN
  - STOP
  - MRES
- DI/DO status display (C7-621) F5
- AS-i (C7-621 AS-i) F5

Selecting the System Function Menu

You select this menu by pressing the SHIFT 0 keys simultaneously.

The following menu is displayed depending on the version of the unit you are using:

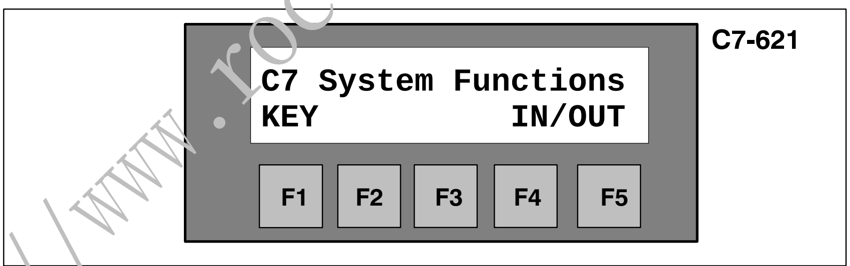


Figure 2-2 System Function Menu and Function Keys on a C7-621

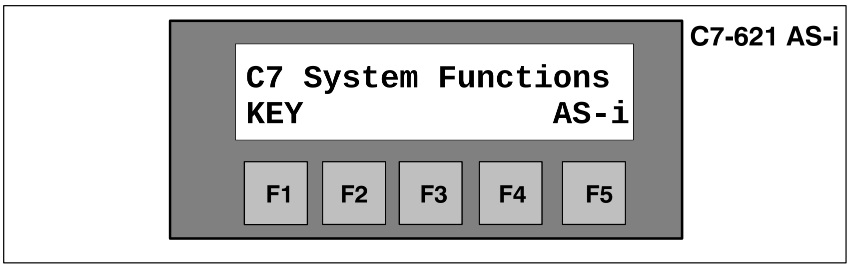


Figure 2-3 System Function Menu and Function Keys on a C7-621 AS-i

Selecting the Mode Menu

You select the **C7 CPU modes** menu in the system function menu by pressing the F1 key.

The following menu is displayed (the MODE: STOP is an example):

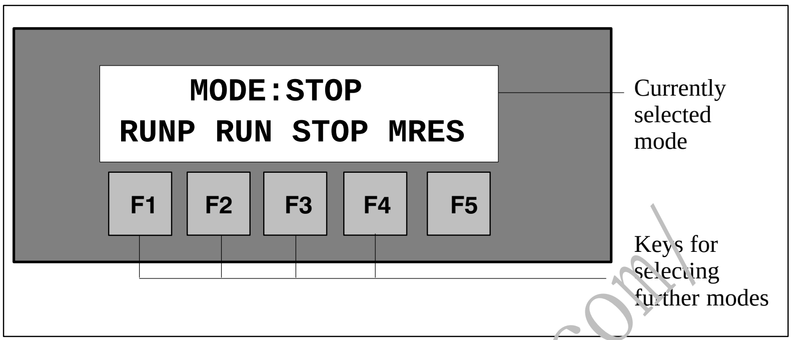


Figure 2-4 "C7 CPU Modes" Menu and Function Keys

Selecting the C7 CPU Modes

You can select one of the C7 CPU modes as follows:

Table 2-1 Selecting a C7 CPU Mode

Mode	Key	Explanation
RUNP	F1	The C7 CPU executes the user program. Programs and data • can be read out from the C7 CPU using a PG (C7 → PG) • transferred to the C7 CPU (PG → C7).
RUN	F2	The C7 CPU executes the user program. Program and data in the C7 CPU can be read out with the programming device (C7 → PG). The user program cannot be downloaded or modified.
STOP	F3	The C7 CPU does not execute the user program. Programs • can be read out from the C7 CPU with a programming device (C7 → PG) • can be transferred to the C7 CPU (PG → C7). **Note:** The STOP mode applies only to the C7 CPU. It does not apply to the C7 OP. You can still continue working with the C7 OP.
MRES	F4	**Memory Reset** The memory reset on the C7 CPU (clear memory, reload user program from flash memory) requires a special operating sequence in the modes STOP and MRES (see Section 2.4). If data required by the configuration are deleted during memory reset, an error message is displayed by the C7 OP. **Note:** The MRES setting is not a momentary contact state and the MRES mode remains set. For the C7 CPU, the MRES mode is only a control mode. If this mode is set permanently, the C7 CPU cannot operate correctly, in other words, you must always reset the mode before exiting the menu by setting STOP, RUN or RUN-P.

Protecting Mode Changes with a Password

To prevent accidental or unauthorized mode changes on the C7 CPU, password protection is activated when a configuration is loaded. The procedure is as follows:

1. If you attempt to change the C7 CPU mode, the active password level is checked (password level >= 8 is necessary).
2. If the password level is not high enough, the LOGIN screen is displayed in which the password can be entered (see Section 7.7).
3. Enter the password
   - You can only change the C7 CPU mode with a valid password.
   - If no key is activated within a time specified in the configuration (0 to 60 minutes), an automatic logoff is started (the current password level is reset to 0, the lowest level).
   - If no password for level 8 has been allocated, you can only modify the mode of the C7 CPU using the configured superuser password (default **100**).
4. If the password is recognized as being valid, you can change the C7 CPU mode.

Exiting the Mode Menu

You can exit the **C7 CPU modes** menu by pressing the ESC key.

Exiting the System Function Menu

You also exit the **system function menu** by pressing the ESC key.

2.3 DI/DO Status Display (C7-621)

Selecting the DI/DO Status Display

You select the DI/DO status display screen in the system function menu by pressing the F5 key.

The state of the digital I/Os can be displayed with a C7 system function. The displayed values are read as direct process values (PIW) of the DI and as the process image (QW) of the DO C7 I/Os and displayed in the BIN format. It is not possible to modify the display.

In the STOP mode, the real process state is DO = 0. The displayed process image can, however, deviate from this (the values are those last set by the control program).

Selecting the DI/DO Status Display

To display the DI/DO status display, you do not require an operator password level. You select the DI/DO status display screen in the system function menu by pressing the F5 key.

The following data are displayed:

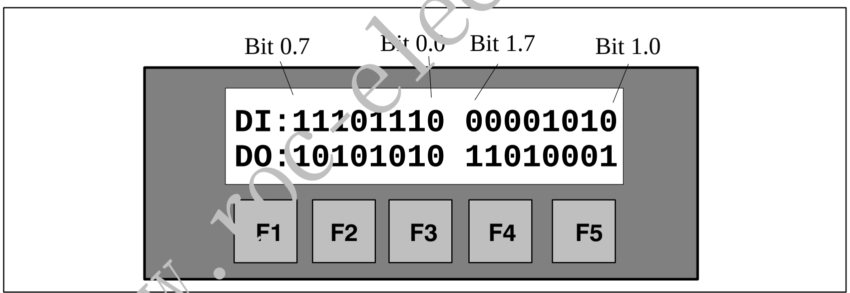


Figure 2-5 Example of a Status Display on a C7-621

Table 2-2 Explanation of the DI/DO Status Display in Figure 2-5

Display	Explanation
1	DI/DO set
0	DI/DO reset

Note

The values of the DI/DO are read in and displayed at one second intervals. Any signal state changes between the scan points are not displayed.

Exiting the DI/DO Display

You exit the DI/DO status display by pressing the ESC key.

Exiting the System Function Menu

You also exit the **System Functions menu** by pressing the ESC key.

2.4 Memory Reset on the C7

Overview

If you want to set the C7 to a “neutral” state, you must reset the memory on the C7 CPU and if required also on the C7 OP.

C7 OP Memory Reset

The C7 must be turned off. To reset the memory on the C7 OP, follow the steps outlined below:

1. Press and hold the [▽] [▷] [ESC] keys simultaneously.
2. Turn on the power for the C7.
3. Hold down the keys until the first line is displayed under `Flash`.

The memory of the C7 OP has now been reset and the C7 OP starts up with the English default configuration.

What Remains After a Memory Reset on the C7 OP...

After resetting the memory on the C7 OP, only the following remains:

- C7 system function menu
- C7 OP default configuration

Memory Reset on the C7 CPU

There are two ways of resetting the memory on the C7 CPU:

- Memory reset with the **C7 CPU Mode** system function
- Memory reset with the PG functions (refer to the STEP 7 manual **/231/**)

A memory reset using the PG function is only possible when the C7 CPU is in the STOP mode.

How to Reset the C7 CPU Memory

The sequence of steps shown below explains how to reset the C7 CPU memory using the **C7 CPU Mode** system function.

1. Turn on the power for the C7 and wait until the startup tests are completed.
   The standby message is displayed.
2. Select the system function menu by pressing the key combination
   [SHIFT] [0]

The system functions menu is displayed:

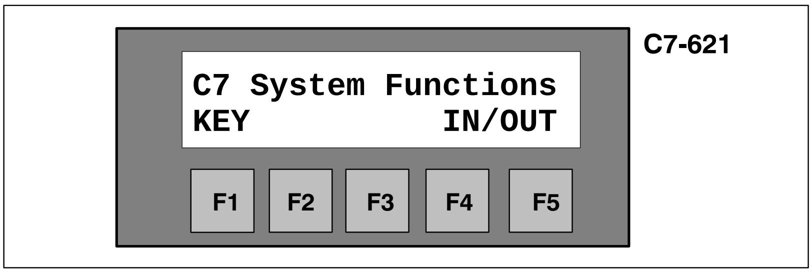


Figure 2-6 System Functions Menu of the C7-621 and Function Keys

or for the C7-621 AS-i

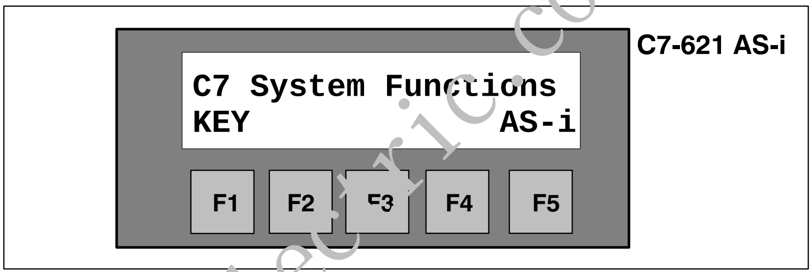


Figure 2-7 System Functions Menu of the C7-621 AS-i and Function Keys

3. Select the **C7 CPU Mode** menu by pressing the F1 key.

   If no password has been entered, this is requested.

4. Enter the password (entering a password is described in Section 7.7).

The following menu is displayed:

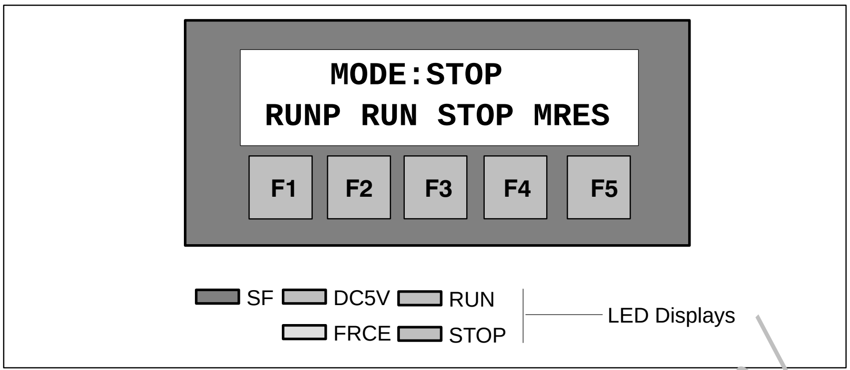


Figure 2-8 “C7 CPU Modes” Menu with Function Keys

5. Select the STOP function by pressing the F3 key.
   The STOP LED is lit.
6. Select the MRES function (memory reset) by pressing the F4 key and wait until the STOP LED goes off twice.
7. Immediately after the STOP LED lights up again:
   Select STOP and immediately afterwards MRES (within one second). If you wait longer than one second between STOP and MRES, no memory reset takes place.

   Result:

   – If the STOP LED flashes quickly for approximately 3 seconds and then lights up again; the memory of the C7 CPU has been reset.

   – If the STOP LED of the C7 lights up without first flashing, repeat steps 4 and 5.
8. After resetting the memory on the C7 CPU, you must change the mode explicitly to STOP or RUN/RUNP otherwise the C7 CPU remains set to MRES.

Note

The content of the C7 CPU flash memory is not affected by a memory reset.

Erasing the Flash Memory

How you erase the contents of the flash memory is described in Section 3.8.

Sequence in the C7 CPU During a Memory Reset

When the memory of the C7 CPU is reset, the STOP display flashes and the following procedure is run through:

1. The C7 CPU erases the entire user program in the work memory and in the load memory.
2. The C7 CPU tests the hardware.
3. If there is a program stored in the integrated flash memory of the C7 CPU, its content is copied automatically to the load memory and compiled into the work memory (how to erase the flash memory is described in Section 3.8).

   If there is no program in the flash memory and work memory, the load memory remains empty and the C7 CPU has memory content "0".

What Remains After Resetting the C7 CPU Memory...

After you have reset the C7 CPU memory, the following remains:

- The content of the diagnostic buffer

  The content can be read out with a programming device.
- The system diagnostics parameters
- A user program with newly initialized data if this has been loaded from the flash memory.
- The last selected MPI parameter settings.

Controlling with the C7 CPU

3

In this Chapter

The C7 CPU is described in this chapter.

The chapter describes the languages you can use and the tools you require to program the C7 CPU.

You can configure characteristics of the C7 CPU by assigning appropriate parameters. The characteristics that you can configure are described in Section 3.6.

Chapter Overview

3.1 C7 CPU : Overview

Characteristics of the C7 CPU for the C7-621 and C7-621 AS-i

The C7 CPU has the following features:

- 12 Kbytes of work memory
- 20 Kbytes of integrated load memory (RAM)
- 20 Kbytes of integrated flash memory
- P bus for connecting external I/Os
- Speed: approximately 0.3 ms per 1000 binary instructions
- Maximum 128 digital inputs/outputs can be connected
- Maximum 32 analog inputs/outputs can be connected

3.2 Programming the C7 CPU

Overview

The user program that controls the process to be visualized by the C7 OP runs on the C7 CPU.

Required Tools

To develop a user program, you require the following tools:

- Programming device/PC with MPI interface and appropriate cable
- STEP 7 and the relevant documentation
- C7

Programming Languages

At present, there are three standard programming languages available for programming the C7 CPU:

- STL: A statement list consists of a series of statements. Each statement in your program contains instructions in the form of mnemonics which represent a function for the C7 CPU to perform.
- LAD: A ladder diagram is a graphical program language that resembles electrical circuit diagrams.
- FBD: A function block diagram is a graphical programming language representing the logic in logic boxes familiar from Boolean algebra.

3.3 Performance of the C7 CPU

Introduction

Table 3-1 contains the most important performance characteristics of the C7 CPU.

Table 3-1 Performance Characteristics of the C7 CPU

Characteristic	C7 CPU	
Work memory (integrated)	12 Kbytes	
Load memory (integrated)	• 20 Kbytes RAM • 20 Kbytes for flash memory. FEPROM	
Speed ms/1000 binary instructions	Approx. 0.3 ms	
Digital inputs	128 + 16 onboard	
Digital outputs	128 + 16 on board	
Analog inputs	32 + 4 on board	
Analog outputs	32 + 1 on board	
Process image • On board Inputs Outputs	124 to 125 I 124.0 to I 125.7 Q 124.0 to Q 125.7	
• External Inputs Outputs	for C7-621 I 0.0 to I 15.7 Q 0.0 to Q 15.7	for C7-621 AS-i I 4.0 to I 19.7 Q 4.0 to Q 19.7
Bit memory	2048	
Counters	64	
Timers	128	
Retentive data area	2 DBs maximum 144 data bytes in total retentive	
Maximum total of all retentive data	144 bytes	
Clock memory	Memory bits that can be used for the clock signal in the user program. Number: 8 (1 memory byte); freely selectable address of a memory byte	
Local data	1536 bytes in total; 256 bytes per priority class	
Nesting depth	8 for each priority class;	
Blocks: OBs FBs FCs DBs SFCs	 12 128 128 127 32	

Table 3-1 Performance Characteristics of the C7 CPU

Characteristic	C7 CPU
MPI interface	
• Transmission rate	187.5 Kbps
• Maximum number of nodes	32 (126 with repeater)
• Distance:	
Without repeaters	50 m
With 2 repeaters	1100 m
With 10 repeaters in series	9100 m
Communication via MPI	
• Maximum number of connections	4
• Global data circles	4
• Transmitted packets	1 per GD circle[1]
• Received packets	1 per GD circle[1]
• Data per packet	max. 22 bytes
• Consistent data per packet	8 bytes
Expansion C7-621	max. 4 signal modules

1 If there are more than two nodes in a GD circle, only one transmitted or received packet.

3.4 Test and Reference Data Functions of the C7 CPU

Overview

The C7 CPU provides test and reference data functions that allow correct working and ensure a defined reaction to errors or faults. The C7 CPUs also provide a series of test and reference data functions with which the status of the C7 CPU and the signal modules assigned to it can be queried. This means that you can obtain information about the following on the C7 CPU and the signal modules assigned to it:

- The current configuration (degree of expansion) of the C7
- The current parameter assignments
- The current states
- The current sequences

You can also modify process variables independent of the user program.

Monitoring Functions

The hardware and the operating system of the C7 CPU provide various monitoring functions.

Any errors that occur are displayed with the SF LED and the cause of the problem is entered in the diagnostic buffer. The C7 CPU either changes to the STOP mode or you can react to the problem using error or interrupt OBs in the user program. The OBs you can program for the C7 CPU are described in Section 3.5.

Note

Please note that despite comprehensive monitoring functions and error reaction functions, the system is not a high-safety or fail-safe system.

List of Reference Data Functions

Table 3-2 contains the reference data function of the C7 CPU. For a detailed description of the reference data functions, refer to the user manual.

Table 3-2 Reference Data Functions on the C7 CPU

Reference Data Functions	Application
User memory	Displays how much memory is currently being used, as follows: • Load memory on the C7-621 integrated flash memory (FEPROM) • Load memory of the C7 CPU (RAM) • Work memory of the C7 CPU (RAM)
Blocks	Displays all the available blocks and possible priority classes: • SFCs (system functions) • OBs (organization blocks) • All blocks
Stacks	Reads out the contents of the following: • B stack (block stack) • I stack (interrupt stack) • L stack (local data stack)
Communication	Displays the following: • Total number of connections • Frame sizes • Transmission rate via the MPI • Reserved OP connections • Reserved PG connections • Free connections
Time system	Displays the following values: • C7 time • C7 data • Time system • Correction factor • Cycle of the synchronization frames
Cycle times	Displays the cycle times of the user program: • Monitoring time • Length of the longest cycle • Length of the shortest cycle • Length of the last cycle

Table 3-2 Reference Data Functions on the C7 CPU, continued

Reference Data Functions	Application
Diagnostic buffer	Reads out the content of the diagnostic buffer: • Date and time at which a diagnostic event occurred • Aim of the diagnostic event • Information describing the diagnostic event in greater detail, for example error OB call with access errors
C7 CPU data	Displays the following information about a C7: • C7 type and version of the C7 CPU • Size of the work and load memory of the C7 CPU • Configuration of the load memory • Number and range of the inputs, outputs, timers, counters and bit memory • Area of the local data • C7 system response

List of Test Functions

Table 3-3 contains the test functions of the C7 CPU.

Table 3-3 Test Functions of the C7

Test Functions	Application
Status variable	Selected process variables (inputs, outputs, bit memory, timers, counters, data) can be monitored at a selected point in the user program.
Force variable	Selected process variables (inputs, outputs, bit memory, timers, counters, data) can be assigned a value at a particular point in the user program (start of the cycle, end of the cycle, transition from RUN → STOP) allowing operator intervention in the user program.
Block status	Monitoring a block in terms of the program sequence as an aid during installation and for troubleshooting. The block status function allows you to monitor the contents of certain registers, for example accumulators, address register, status register, DB register while instructions are being executed.

3.4.1 Clock and Run-Time Meter

Introduction

The C7 CPU has a software clock.

The C7 CPU also provides a run time meter that allows you to count the operating hours of the C7 CPU or a connected resource.

You set and read the clock using the programming device (refer to the *STEP 7* user manual) or you can program the clock in the user program with SFCs refer to the reference manual *System and Standard Functions* and Appendix B).

You program the run time meter in the user program once again using SFCs (see reference manual *System and Standard Functions* and Appendix B).

Characteristics

Table 3-4 lists the characteristics and functions of the C7 CPU clock.

When you assign parameters for the C7 CPU in *STEP 7*, you can also set functions such as synchronization and a correction factor. For more information refer to Section 3.6.3 and the online help of *STEP 7*.

Table 3-4 Characteristics of the C7 CPU Clock

Characteristics	C7 CPU
Type	Software clock
Default when shipped	DT#1994-01-01-00:00:00
Backup	Not possible
Run-time meter Range of values	1 0 to 32767 hours

Power Off

When the power is turned on, the C7 CPU clock continues at the time at which the power was turned off. Since there is no backup for the C7 CPU, the clock does not continue to run when the power is turned off.

3.5 Blocks on the C7 CPU

Overview

This chapter provides an overview of the blocks that can be executed on your C7 CPU.

The operating system of the C7 CPU is designed for event-driven user program execution. The following tables indicate which organization blocks (OBs) the operating system calls automatically for which event.

Further Information

The programming manual **/234/** contains a detailed description of event-driven user program execution. The OBs and the start events listed here are described in detail in the reference manual **/235/**. For an overview of the *STEP 7*documentation, refer to Appendix D.

C7 CPU Blocks

Table 3-5 shows all the blocks that the C7 CPU can execute.

Table 3-5 Overview: C7 CPU Blocks

Block	Number	Range	Maximum Size	Comment
OB	12	-	8 Kbytes[1]	There is a list of all possible OBs following this table.
FB	128	0 - 127	8 Kbytes[1]	-
FC	128	0 - 127	8 Kbytes[1]	-
DB	127	1 - 127	8 Kbytes[1]	0 is reserved
SFC	32	-	-	Appendix A contains a list of all SFCs for the C7 CPU. For a detailed description, refer to the reference manual **/235/**.

1 Section of the block relevant to program execution

OBs for the Scan Cycle and Startup

Table 3-6 lists the OBs that determine the response of the C7 CPU during the cycle and during startup.

Table 3-6 List of OBs for the Scan Cycle and Startup

Scan Cycle and Startup	Called OB	Possible Start Events	Default Priority of the OBs
Scan cycle	OB 1	1101_H, 1103_H	Lowest priority
Startup (STOP-RUN transition)	OB 100	1381_H, 1382_H	—

OBs for Internal and External Interrupts

Table 3-7 lists the OBs that determine the response of the C7 CPU to interrupts.

The priority of the OBs cannot be modified.

OB 35

You can set intervals of 1 ms and higher for OB 35 (cyclic interrupt OB). If you set intervals less than 5 ms, then despite short program run times for the OB 35 program, cyclic interrupt errors can occur.

Table 3-7 OBs for Internal and External Interrupts

Interrupts (Internal and External)	Called OB	Possible Start Events	Priority of the OB	Priority
Time-delay interrupt Range: 1 ms to 60000 ms (can be set in 1 ms steps)	OB 20	1121_H	3	↓
Cyclic interrupt Range: 1 ms to 60000 ms (can be set in 1 ms steps; we recommend a setting > 5 ms)	OB 35	1136_H	12	
Hardware interrupt	OB 40	1141_H	16	
Diagnostic interrupt	OB 82	3842_H, 3942_H	26	High

Response of the C7 CPU if there is no Interrupt OB

If you do not program an interrupt OB, the C7 CPU responds as follows:

C7 CPU Changes to STOP Without ..	C7 CPU Remains in RUN Without ..
OB 20 (time delay interrupt) OB 40 (hardware interrupt) OB 82 (diagnostic interrupt)	OB 35 (cyclic interrupt)

OBs for Reactions to Errors

Table 3-8 lists the OBs that influence the response of the C7 CPU to errors/faults.

Table 3-8 OBs for Error/Fault Reactions

Error/Fault	Called OB	Possible Start Events	Default Priority of the OBs
Time error (for example triggered by watchdog monitoring)	OB 80	3501_H, 3502_H, 3505_H, 3507_H	26
Power supply fault	OB 81	3822_H, 3922_H	26
One of the following program errors occurred: • Event to start an OB (for example time delay interrupt) occurred but the corresponding OB cannot be executed • Error in process image updating (module does not exist or defective) • Error caused by the operating system accessing a non-existing block (for example the DB for an integrated function was deleted)	OB 85	$35A1_H$, $35A3_H$, $39B1_H$, $39B2_H$,	26
Communication errors • Wrong frame ID when receiving global data • The data block for the status of the global data does not exist or is too short	OB 87	$35E1_H$, $35E2_H$, $35E6_H$	26
Programming error (for example addressed timer does not exist)	OB 121	2521_H, 2522_H, 2523_H, 2524_H, 2525_H, 2526_H, 2527_H, 2528_H, 2529_H, 2530_H, 2531_H, 2532_H, 2533_H, 2534_H, 2535_H, $253A_H$; $253C_H$, $253E_H$	The same priority as the OB in which the error occurred
Error in direct I/O access (module defective or does not exist)	OB 122	2944_H, 2945_H	The same priority as the OB in which the error occurred

OB 121 and 122

Note the following response of the C7 CPU to OBs 121 and 122:

Note

Note the following special reaction to OBs 121 and 122:

The CPU enters the value “0” in the following temporary variables of the variable declaration table in the local data of the OBs:

- **Byte No. 4:** OB121_BLK_TYPE or OB122_BLK_TYPE
  (Type of block in which the error occurred)
- **Bytes No. 8 and 9:** OB121_BLK_NUM or OB122_BLK_NUM
  (Number of the block in which the error occurred)
- **Bytes No. 10 and 11:** OB121_PRG_ADDR or OB122_PRG_ADDR
  (Address of the block in which the error occurred)

Response of the CPU if there is no Error OB

If you do not program an error OB, the C7 CPU reacts as follows:

C7 CPU Changes to STOP without ...		C7 CPU Remains in RUN without ...	
OB 80	(Time error)	OB 81	(Power supply fault)
OB 85	(Program sequence error)		
OB 87	(Communication error)		
OB 121	(Programming error)		
OB 122	(Direct I/O access error)		

3.6 C7 CPU Parameters

Selectable Characteristics of the C7 CPU

You can assign parameters to specify the characteristics and response of the C7 CPU. You assign parameters in STEP 7 in various tab pages (refer to the *STEP 7 documentation* and online help of*STEP 7*).

Parameter fields of the C7 CPU:

- Cycle/Clock Memory
- Startup
- Retentive memory
- Interrupts
- Diagnostics
- MPI node address

Tool for Parameter Assignment

You assign parameters to the C7 CPU using STEP 7 *hardware configuration*. *Hardware configuration* is described in manual **/231/**.

When does the C7 CPU "Adopt" the Parameters?

The C7 CPU adopts the selected parameters (configuration data)

- After POWER ON
- After correct transfer of the parameters to the C7 CPU online in the STOP mode
- After a memory reset on the C7 CPU (see Section 2.4):
  If the integrated flash memory of the C7 CPU contains configuration data, these are loaded with the exception of the MPI parameters.

Caution

After saving the program in the flash memory of the C7 (STEP 7: "Copy RAM to ROM"), parameter settings with the exception of the MPI parameters are lost the next time memory is reset.

3.6.1 “Cycle/Clock Memory” Tab

Parameter Field “Cycle”

Table 3-9 lists the parameters of the “Cycle” parameter field from the “Cycle/Clock Memory” tab.

Table 3-9 “Cycle”Parameter Field

Parameter	Explanation	Range of Values	Default Setting
Cycle load from communication (via the MPI) (in %)	To limit the slowing down of program execution by communication processes, the maximum percentage cycle load can be specified. Limiting the cycle load can slow down communication between the C7 CPU and a programming device or between communicating C7 CPUs. The operating system services such as collection and preparation of data for communication are not affected. Functions that demand that data are read without interruptions slow down program execution regardless of the value set for this parameter. Example: Block status, reading out system data.	From 10 to 50	20
Scan Cycle Monitoring (in ms)	If the scan cycle exceeds the “Scan Cycle Monitoring”, the C7 CPU changes to the STOP mode. The “Scan Cycle Monitoring” can be exceeded, for example due to the following: • Communication processes • Large numbers of interrupts • Errors in the user program (for example endless loops).	From 1 to 6000	150
Cycle load from self-test (in µs)	If “Self Test ≠ 0” is set, the C7 CPU tests its internal RAM during the “Program” cycle. This self test takes up additional cycle time. You can enter the time by which the (program) cycle can be extended as a multiple of 10 µs (“0” = no cyclic self test).	0 **cannot be modified**	

Definition of Clock Memory

Clock bits are memory bits that change their bit state periodically with a pulse-pause ratio of 1: 1. Eight fixed frequencies are defined for the C7 that can be entered in any memory byte. The period is shown in Figure 3-1.

Clock Pulse Period

Figure 3-1 shows the period and the corresponding frequency of clock pulses generated by the “Clock Pulse Byte”.

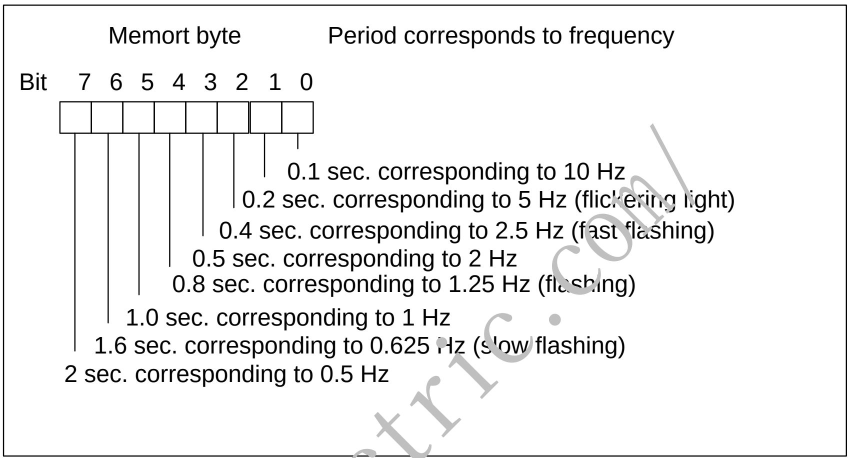


Figure 3-1 Period of the Clock Pulses in the “Clock Memory Byte”

“Clock Memory” Parameter Field

Table 3-10 lists the parameters of the “Clock Memory” parameter field of the “Cycle/Clock Memory” tab.

Table 3-10 “Clock Memory” Parameter Block

Parameter	Explanation	Range of Values C7 CPU	Default Setting
Clock memory	If “Clock Memory = yes” is set, a memory bit must be specified	yes/no	no
Memory byte	Memory byte to be used as the “Clock Memory Byte”.	from 0 to 255	0

3.6.2 “Startup” Tab

“Startup”

Table 3-11 lists the parameters of the “Startup” tab.

Table 3-11 Parameter Field in the “Startup” Tab Page

Parameter	Explanation	Range of Values	Default Setting
Hardware test on complete restart	If “hardware test during complete restart = yes” is set, the CPU tests its internal RAM each time the power supply is turned on.	yes/no	no
Startup after Power On	Only a complete restart can be set for the C7 CPU.	Complete restart	Complete restart
Monitoring time for ...			
• Transfer of parameters to modules (in ms)	Maximum time for “distributing” the parameters to the modules within the rack.	from 1 to 10000 (= 100ms to 100s) (time base = 100 ms)	100 (= 10 s)
• Ready message from modules (in ms)	Maximum time for the ready message from all modules after power on. If the modules do not send a ready message during this time to the C7 CPU, the C7 CPU remains in the STOP mode.	from 1 to 65000	65000

Tip

Select the largest values for the parameters “Monitoring Time For ...” if you are unsure of the times required in the C7 CPU.

3.6.3 “Diagnostics/Clock” Tab

Diagnostics

In the “System Diagnostics” parameter field of the “Diagnostics/Clock” tab, you decide the range within in which the C7 CPU provides you with messages from system diagnostics.

Definition: System Diagnostics

System diagnostics involves the acquisition, evaluation and signaling of an error within the programmable controller. System diagnostics is also responsible for monitoring the wiring to the process so that, for example, a wire break can be recognized by the system diagnostics.

Example

Examples of errors that can be identified, evaluated and signaled by system diagnostics are as follows:

- Errors in the user program
- Module failures

“System Diagnostics” Field

Table 3-12 lists the parameters of the “ System Diagnostics” parameter field.

Table 3-12 System Diagnostics” Parameter Field

Parameter	Explanation	Range of Values	Default Setting
Extended functional scope	If “Expanded Diagnostic Entries = yes” is set, the C7 CPU not only enters error events but also other events in the diagnostic buffer, for example OB calls.	—	—
Display cause of STOP	If “Display Cause of STOP = yes” is set, the C7 CPU automatically sends the cause of a STOP via the MPI to a node logged on for this purpose (PG, OP). This diagnostic message is the “newest” entry in the diagnostic buffer.	yes/no	yes

Undetected Errors

Errors that occur in the process, in other words outside the programmable controller, are not detected by system diagnostics. Such errors for example include “Motor Defective”. Errors of this type belong to process error diagnostics.

3.6.4 “Retentive Memory” Tab

Definition: Retentive Memory

A memory area is retentive if its contents are retained even after a power failure and a transition from STOP to RUN. The non-retentive area for bit memory, timers and counters is reset following a power outage and the transition from STOP to RUN.

The following can be retentive:

- Bit memory
- S7 timers
- S7 counters
- Data areas (only with an integrated EPROM)

What is Retained?

The areas that you specify in the “Retentive Memory” parameter field are retained following a power outage and the transition from STOP to RUN (without battery backup).

Retention of Data Blocks

Note the following points about data blocks:

- All DBs (retentive, non-retentive) are transferred from the integrated EPROM to the work memory during startup.
- Data blocks or data areas that you create with SFC 22 “CREAT_DB” are non-retentive.
- Retentive data areas are retained after a power outage. **Note:** These data areas are stored on the C7 CPU. The non-retentive data areas contain the values programmed in the EPROM.

“Retentive Memory”

Table 3-13 lists the parameters of the “Retentive Memory” tab page. The total retentive area for all areas (bit memory, timers, counters and data bytes) must not be larger than the sum of all the parameters shown in Table 3-13.

Table 3-13 “Retentive Memory” Parameter Field

Parameter	Explanation	Range of Values C7	Default Setting
Number of memory bytes from MB0	The parameter value entered is the number of retentive memory bytes starting at memory byte 0	0 to 143	16
Number of S7 timers from T0	The entered parameter value is the number of retentive S7 times starting at timer 0 (space required: 2 bytes/timer)	0 to 71	0
Number of S7 counters from C0	The entered parameter value is the number of S7 counters starting at counter 0 (space required: 2 bytes/counter)	0 to 64	8

Table 3-13 “Retentive Memory” Parameter Field

Parameter	Explanation	Range of Values C7	Default Setting
DB No. • Data block number	If “Data Block = yes” is set, the data block and the required “number of data bytes” from “data byte address” must be specified. 2 data blocks can be retentive with a total of 144 bytes. The start address of the data area plus the number of data bytes must not exceed 8192.	yes/no • from 1 to 127	• 1
• Number of bytes		• from 0 to 144	• 0
• Byte address (start address of the length of the data area)		• from 0 to 8191	• 0
Sum of all the retentive data		144 bytes	

3.6.5 “Interrupts” Tab Page

“Interrupts”

Table 3-14 lists the parameters of the “Interrupts” tab page.

The priority of the time delay interrupt OB (OB20) and the hardware interrupt OB (OB40) cannot be modified.

Table 3-14 Parameters in the “Hardware Interrupts” Tab Page

Parameter	Explanation	Range of Values	Default Setting
Time delay interrupts Priority OB 20	The priority of OB 20 cannot be modified.	3	3
Hardware interrupts Priority OB 40	The priority of OB 40 cannot be modified.	16	16

3.6.6 “Cyclic Interrupt” Tab Page

Definition: Cyclic Interrupt

A cyclic interrupt is a periodic signal that the C7 CPU generates internally and that causes a “Cyclic Interrupt OB” (OB 35) to be called.

Priority

The priority of OB 35 is fixed at 12. You cannot modify this value.

“Cyclic Interrupts”

Table 3-15 lists the parameters of the “Cyclic Interrupts” tab page.

Table 3-15 Parameters in the “Cyclic Interrupts” Tab Page

Parameter	Explanation	Range of Values	Default Setting
Priority of OB 35	You cannot modify the priority of OB 35.	12	12
Execution (in ms)	Call interval of OB 35	from 1 to 60000	100

Periodicity > 5 ms

For cyclic interrupts, you should set the execution to a value greater than 5 ms. If you set a value less than 5 ms, this increases the risk of cyclic interrupt errors occurring due to the following:

- The program run time of the OB 35 program
- The frequency and program run times of higher priority classes
- PG functions.

3.6.7 “Nodes – MPI” Parameter Field in the “General” Tab Page

Multipoint Interface (MPI)

The properties of the multipoint interface (MPI) of the C7 CPU can be selected in the “MPI Addresses” parameter field. You only need to change values in this parameter field if you have networked more than one C7 or S7-300 via the multipoint interface (MPI).

Values Following Memory Reset

The parameters of the “MPI Addresses” parameter field have one unusual feature: The parameter values are retained even following a memory reset. The reason for this is that a C7 CPU must remain capable of communication even after a memory reset. The default parameters when shipped are as follows:

- Highest MPI address: 15

MPI Addresses of the C7

The C7 occupies two MPI addresses:

- One for the C7 CPU (default address 2)
- One for the C7 OP (default address 1)

“MPI Addresses” Parameter Field

Table 3-16 lists the parameters of the “MPI Addresses” parameter field.

Table 3-16 “MPI Addresses” Parameter Field

Parameter	Explanation	Range of Values	Default Setting
C7 CPU MPI address	Each node that is networked via the MPI must have an address. The address assigned must only occur once in the network. The C7 OP has its own MPI address (Default = 1)	from 2 to 126	2
Highest MPI address	The highest MPI address in the network must be specified for the following reasons: • So that each node is addressable • The communication process runs effectively. Tip: Only assign as many MPI addresses as necessary. This reduces the time necessary for communication. The “Highest MPI Address” parameter must be the same on all MPI nodes!	15 31 63 126	31
Transmission rate	You cannot change the transmission rate of the MPI network	187,5 Kbps	187.5 Kbps

Note

The MPI interface is the only communication interface to the C7. Take extreme care if you decide to change these parameters.

The default settings for the programming device are MPI address 0 and highest MPI address 15. If you set a different value, for example 31, 63 or 126, you must also adapt the highest address on the programming device/PC.

3.7 Calculating the Cycle and Reaction Time of the C7 CPU

In this Section

This section explains how the cycle and reaction times of the C7 CPU are made up.

The cycle time of your user program on the C7 CPU can be read out on the programming device (refer to the programming manual **/280/**).

An example illustrates how to calculate the cycle time.

When considering a process, the reaction time is more important. Calculating the reaction time is explained in detail in this section. The AS-i and the OP times are not taken into account in this section.

Definition of the Cycle Time

The cycle time is the time required for a full program scan cycle.

Parts of the Cycle Time

The cycle time consists of the following:

Factors	Comment
Operating system run time	See Section 3.7.1
Process image transfer time (PII and PIQ)	
User program execution time	... Calculated from the execution times of the individual instructions (refer to the *Instruction List of the S7-300*) and a CPU-specific factor.
S7 timers	See Section 3.7.1
Communication via the MPI	You select the maximum permitted load in the cycle caused by communication as a percentage of the cycle using *STEP 7* .
Delays due to interrupts	See Sections 3.7.3 and 3.7.4

Figure 3-2 shows the components of the cycle time

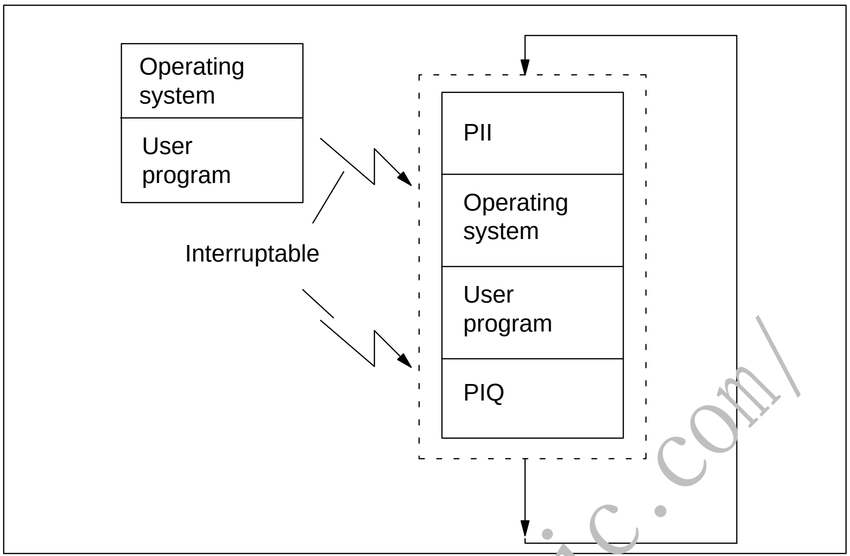


Figure 3-2 Components of the Cycle Time

Extension of the Cycle Time

You should always keep in mind that the cycle time of a user program can be extended by the following:

- Time-driven interrupt servicing
- Hardware interrupt servicing (see also Section 3.7.3)
- Diagnostics and error processing (see also Section 3.7.3)
- Communication via the MPI

3.7.1 Reaction Time

Definition: Reaction Time

The reaction time is the time between detecting an input signal to modifying the output signal associated with it.

Factors

The reaction time depends on the following factors:

Factors	Comment
Delay of the inputs and outputs	The delay times are listed in the technical data • Of the signal modules in reference manual *Module Data*. • Of the integrated inputs/outputs of the C7-621 CPU

Range of Fluctuation

The actual reaction time is between a shortest and a longest reaction time. When you configure your system, you must always assume the longest reaction time.

The next section looks at the shortest and longest reaction time so that you can get a general picture of the range by which the reaction time can vary.

Shortest Reaction Time

Figure 3-3 shows the conditions under which the shortest reaction time is achieved.

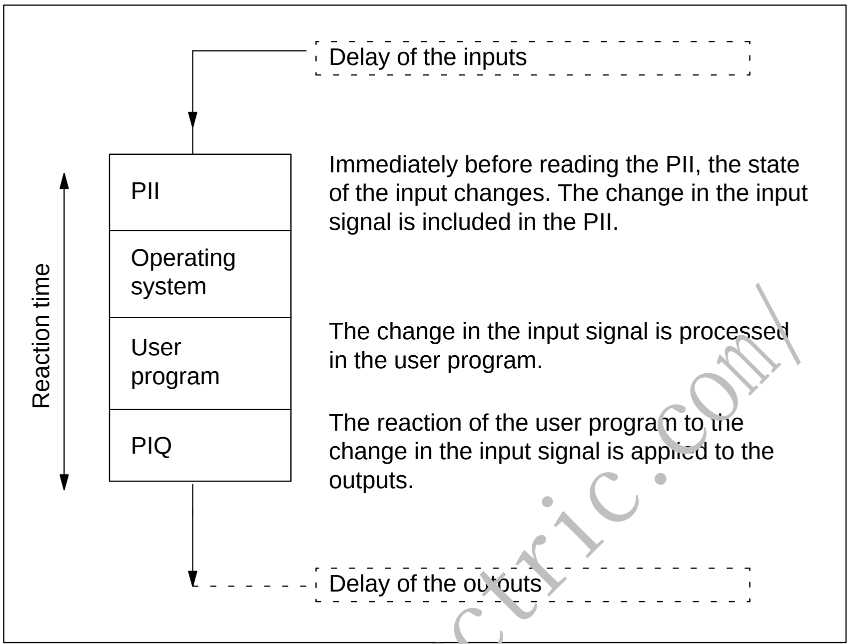


Figure 3-3 Shortest Reaction Time

Calculation

The (shortest) reaction time is made up as follows:

- 1 × process input image transfer time +
- 1 × operating system execution time +
- 1 × program execution time +
- 1 × process output image transfer time +
- Processing time for the S7 timers +
- Delay of the inputs and outputs

This corresponds to the total made up of the cycle time and the delay of the inputs and outputs.

Longest Reaction Time

Figure 3-4 shows how the longest reaction time occurs.

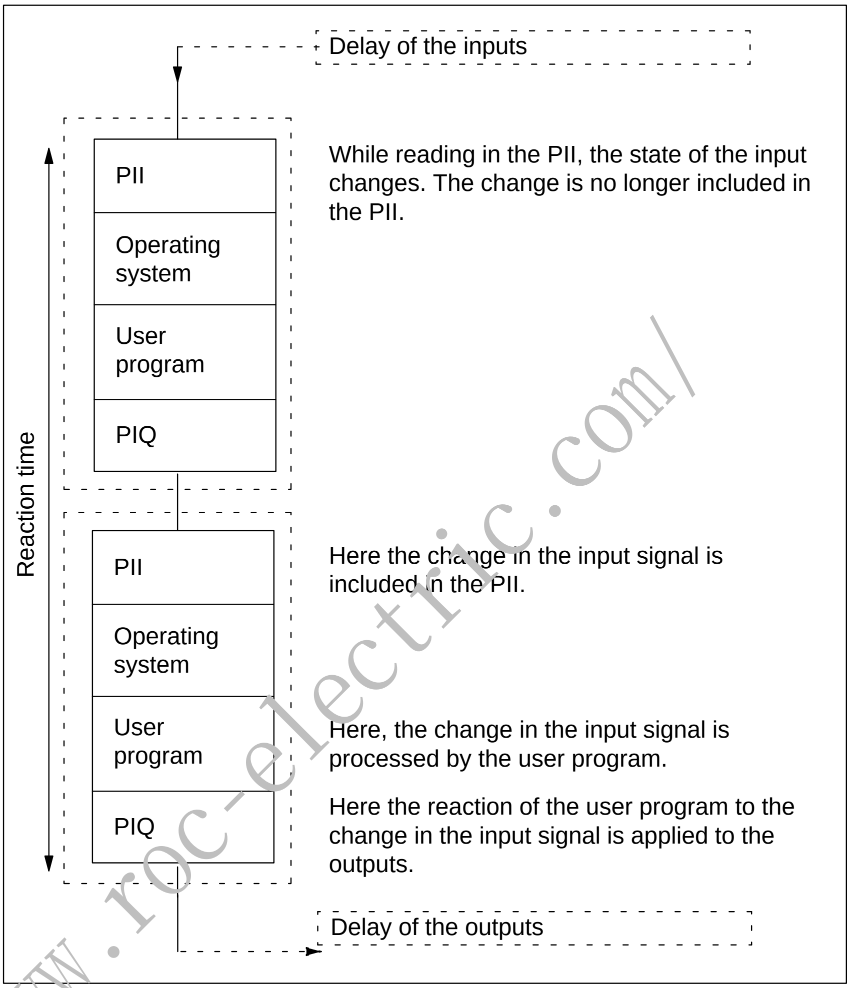


Figure 3-4 Longest Reaction Time

Calculation

The (longest) reaction time is made up as follows:

- 2 × process input image transfer time +
- 2 × process output image transfer time +
- 2 × operating system execution time +
- 2 × program execution time +
- Time for processing the S7 timers +
- Delay of the inputs and outputs

This corresponds to the total of twice the cycle time and delay of the inputs and outputs including twice the bus run time.

Operating System Execution Time

Table 3-17 contains the times you require to calculate the operating system execution times of the C7 CPU.

The specified times apply without

- Test functions, for example status, force
- Functions for loading, deleting and compressing blocks
- Communication.

Table 3-17 Operating System Execution Times of the C7 CPU

Activity	C7 CPU
Cycle control	770 to 1340 μs

Process Image Updating

Table 3-18 contains the C7 CPU times for process image updating (process image transfer time). The specified times are “Ideal Values” that can be extended by interrupts or CPU communication.
(Process image = PI)

The C7 CPU time for process image updating is calculated as follows:

K + number of bytes in the PI in the rack “0” × A
+ number of bytes in the PI in the racks “1 to 3“ × B
+ number of bytes in the PI over DP × D
= C7 CPU time

Table 3-18 Process Image Updating on the C7 CPU

	Components	C7 CPU
K	Basic load	147 μs

User Program Execution Time

The user program execution time consists of the sum of execution times of the instructions and the called SFB/SFCs. You will find these execution times in the instruction list. You must also multiply the user program execution time with a factor for the specific C7 CPU.

The factor for the C7 CPU is 1.2.

S7 Timers

The S7 timers are updated every 10 ms.

Table 3-19 Updating the S7 Timers

Activity	C7 CPU
Updating the S7 timers (every 10 ms)	Number of simultaneously active S7 timers × 5 μs

Delay of the Inputs and Outputs

Depending on the modules, you must take into account the following delay times:

- For digital inputs: the input delay time
- For digital outputs: insignificant delay times
- For relay outputs: typical delay times of 10 ms to 20 ms. The delay of the relay outputs is also dependent on temperature and voltage.
- For analog inputs: cycle time of the analog input
- For analog outputs: response time of the analog output

Reducing the Reaction Time

You can achieve faster reaction times by direct access to the I/Os in the user program, for example with L PIB or T PQW. In this way you can avoid the reaction times as shown in Figure 3-4.

The execution times for direct access by the CPUs to I/O modules are listed in the instruction list.

Extending the Cycle by Including Interrupts

Table 3-20 shows how the cycle time is extended by interrupts. The program run time in the interrupt priority class must also be added to this extension. If several interrupts occur, the times of each interrupt must be added.

Table 3-20 Extending the Cycle with Interrupts

C7 CPU	Hardware Interrupt	Diagnostic Interrupt	Time-of-Day Interrupt	Time Delay Interrupt	Cyclic Interrupt	Programming/ Access Error
C7-621	Approx. 730 µs	Approx. 1000 µs	Approx. 700 µs	Approx. 560 µs	Approx. 380 µs	Approx. 760 µs

3.7.2 Example of Calculating the Cycle and Reaction Time

Components of the Cycle Time

To recap: The cycle time consists of the following:

- The process image transfer time +
- The operating system execution time +
- The user program execution time +
- The time for processing S7 timers

Example Configuration 1

You have configured the C7-621 with the following modules in a rack:

- 1 x CPU 314
- 2 x digital input modules SM 321; DI 32 × DC 24 V (each 4 bytes in the PI)
- 2 x digital output modules SM 322; DO 32 × DC 24 V/0.5A (each 4 bytes in the PI)

According to the instruction list, the user program has a run time of 1.5 ms. There is no communication.

Calculation

With this example, the cycle time consists of the following times:

- Process image transfer time

  Process input image: 147 µs + 8 bytes × 15.6 µs = approx. **0.272 ms**

  Process output image: 147 µs + 8 bytes × 15.6 µs = approx. **0.272 ms**

- Operating system run time

  Cycle control: approx. **1 ms**

- User program execution time:

  approx. 1.5 ms × CPU specific factor 1.2 = **1.8 ms**

- Time for processing S7 timers

  Assumption: 30 S7 timers are running.

  For 30 S7 timers, one update takes the following time:

  30 × 5 µs = 150 µs.

  Adding the process image transfer time, the operating system run time and the user program execution time, results in the following time interval:

  0.272 ms + 0.272 ms + 1 ms + 1.8ms = 3.34 ms.

  Since the S7 timers are called every 10 ms, only one call can possibly take place in this time interval, in other words the cycle time can only be extended by a maximum of 150 µs by the S7 timers.

The cycle time is the sum of the listed times:
Cycle time = 0.272 ms + 0.272 ms + 1 ms + 1.8 ms + 0.015 ms **= 3.35 ms.**

Components of the Reaction Time

To recap: The reaction time is the sum of the following:

- 2 × process input image transfer time +
- 2 × process output image transfer time+
- 2 × operating system execution time +
- 2 × program execution time +
- Time for processing S7 timers +
- Delay times of the inputs and outputs

Tip: To simplify the calculation: Calculated cycle time × 2 + delay times. For the configuration in Example 1, this means 3.34 ms × 2 + delay times of the input/output modules.

Sample Configuration 2

You have configured a C7-621 with the following modules in the rack:

- 1 x CPU 314
- 1 x digital input module SM 321; DI 32 × DC 24 V (4 bytes in the process image)
- 1 x digital output module SM 322; DO 16 × DC 24 V/0.5A (2 bytes in the process image)
- 1 analog input module SM 331; AI 8 × 12 bits (not in the process image)
- 1 analog output module SM 332; AO 4 × 12 bits (not in the process image)

User Program

According to the instruction list, the user program has a run time of 2.0 ms. Taking into account the CPU-specific factor of 1.19, the run time is approximately 2.4 ms. The C7 CPU has 56 active S7 timers. No activity is necessary at the cycle checkpoint.

Calculation

For this example, the reaction time is as follows:

- Process image transfer time

  Process image of the inputs: 147 μs + 4 bytes × 15.6 μs = approx. **0.21 ms**

  Process image of the outputs: 147 μs + 2 bytes × 15.6 μs = approx. **0.18 ms**

- Operating system execution time

  Cycle control: approx. **1 ms**

- User program execution time: **2.4 ms**

- **First subtotal:** The sum of all the previously listed times used as a time base for calculating the processing time of the S7 timers:

2×0.21 ms	(Process input image transfer time)
+ 2×0.18 ms	(Process output image transfer time)
+ 2×1 ms	(Operating system execution time)
+ 2×2.4 ms	(User program execution time)
$\approx$ **7.6 ms**.	

- Time for processing the S7 timers

  A single update of 56 S7 timers takes
  $56 \times 5\ \mu s = 280\ \mu s \approx 0.3$ ms.

  Since the S7 timers are called every 10 ms, the cycle time can include a maximum of one call, in other words the cycle time can be extended by a maximum of 0.3 ms by the S7 timers.

- **Second subtotal:** The reaction time **without** delay times of the inputs and outputs is calculated from the sum of the following:

7.6 ms	(Result of the first subtotal)
+ 0.3 ms	(Time for processing the S7 timers)
=7.9 ms.	

- Delay times of the inputs and outputs
  - The digital input module SM 321; DI $32 \times$ DC 24 V has an input delay of a maximum **4.8 ms** per channel
  - The output delay of the digital output module SM 322; DO $16 \times$ DC 24 V/0.5A can be ignored.
  - The analog input module SM 331; AI 8×12 bit was assigned parameters for noise frequency suppression of 50 Hz. This results in a conversion time of 22 ms per channel. Since 8 channels are active, this results in a cycle time for the analog input module of **176 ms**.
  - The analog output module SM 332; AO 4×12 bits was assigned parameters for the measuring range 0 to 10 V. The conversion time is **0.8 ms** per channel. Since 4 channels are active, this results in a cycle time of 3.2 ms. The settling time for an ohmic load of 0.1 ms must be added. This produces a response time for an analog output of **3.3 ms**.

- Reaction times with delay times of the inputs and outputs:
- **Case 1:** When a digital input signal is read in, the output channel of the digital output module is set. This results in a reaction time of:

  Reaction time = 4.8 ms + 7.9 ms **= 12.7 ms**.

- **Case 2:** An analog value is read in and an analog value is output. This results in a reaction time:

  Reaction time = 176 ms + 7.9 ms + 3.3 ms **= 187.2 ms**.

3.7.3 Interrupt Reaction Time

Definition: Interrupt Reaction Time

The interrupt reaction time is the time from the first occurrence of an interrupt signal until the first statement in the interrupt OB is started.

Higher priority interrupts have precedence. This means that the interrupt reaction time is extended by the program execution time of the higher priority interrupt OBs and the same priority interrupt OBs that have not yet been executed.

Calculation

The interrupt reaction time is calculated as follows:

Shortest interrupt reaction time =
minimum interrupt reaction time of the C7 CPU +
minimum interrupt reaction time of the signal modules

Longest interrupt reaction time =
maximum interrupt reaction time of the C7 CPU +
maximum interrupt reaction time of the signal module

Hardware Interrupt Reaction Times of the C7 CPU

Table 3-21 shows the hardware interrupt reaction times of the C7 CPU (without communication).

Table 3-21 Hardware Interrupt Reaction Times of the CPUs

C7 CPU	min.	max.
C7 CPU	0.5 ms	1.1 ms

Diagnostic Interrupt Reaction Times of the C7 CPU

Table 3-22 contains the diagnostic interrupt reaction times of the C7 CPU (without communication).

Table 3-22 Diagnostic Interrupt Reaction Times of the C7 CPU

C7 CPU	min.	max.
C7 CPU	0.7 ms	1.3 ms

Signal Modules

The hardware interrupt reaction time of the signal modules is made up as follows:

- Digital input modules

  Hardware interrupt reaction time = internal interrupt preparation time + input delay

  The times are listed in the data sheet of the particular digital input module.

- Analog input modules

  Hardware interrupt reaction time = internal interrupt preparation time + conversion time

  The internal interrupt preparation time of the analog input modules is insignificant. The conversion times can be found in the data sheet of the particular analog input module.

The diagnostic interrupt reaction time of the signal modules is the time from the detection of the diagnostic event by the signal module to the triggering of the diagnostic interrupt by the signal module. This time is insignificant and can be ignored.

Hardware Interrupt Servicing

A hardware interrupt is serviced by calling the hardware interrupt OB (OB 40). Higher priority interrupts can interrupt process interrupt servicing. Direct access to the I/Os is made when the instruction is executed. When the hardware interrupt has been serviced, either the cyclic program is resumed or other interrupt OBs with the same or lower priority are called and executed.

3.7.4 Example of Calculating the Interrupt Reaction Time

Components of the Interrupt Reaction Time

To recap: The hardware interrupt reaction time is made up of the following:

- Hardware interrupt reaction time of the CPU and
- Hardware interrupt reaction time of the signal module.

Example: You have a C7-621 consisting of a C7 CPU and 4 digital modules. One digital input module is the SM 321; DI 16 × DC 24V; with hardware and diagnostic interrupt. In the parameter assignment of the CPU and the SM you have only enabled the hardware interrupt. You do not require time-driven execution, diagnostics and error processing. You have selected an input delay of 0.5 ms for the digital input module. No activities are necessary at the cycle checkpoint. There is no communication via MPI.

Calculation

In the example, the hardware interrupt reaction time is calculated as follows:

- Hardware interrupt reaction time of the C7 CPU: approx. 1.1 ms
- Hardware interrupt reaction time of the SM 321; DI 16 × DC 24V:
  - Internal interrupt preparation time:
    0.25 ms
  - Input delay:
    0.5 ms

The hardware interrupt reaction time is the sum of the listed times:

Hardware interrupt reaction time = 1.1 ms + 0.25 ms + 0.5 ms = **approx. 1.85 ms**.

The calculated hardware interrupt reaction time is the time between the occurrence of a signal at the digital input until the first statement in OB 40.

3.7.5 Reproducibility of Time-Delay and Cyclic Interrupts

Definition: "Reproducibility"

Time Delay Interrupt:

The difference in time between calling the first instruction of the task and the programmed interrupt point.

Cyclic Interrupt:

The range of variation in the time interval between two consecutive calls, measured between the first statement of the task.

Reproducibility

Table 3-23 shows the reproducibility of time delay and cyclic interrupts on the C7 CPU.

Table 3-23 Reproducibility of Time Delay and Cyclic Interrupts on the CPU

C7 CPU	Reproducibility	
	Time Delay Interrupt	Cyclic Interrupt
C7 CPU	approx. -1/+0.4 ms	approx. ± 0.2 ms

3.8 Loading/Erasing the C7 CPU Flash Memory

Overview

When it is transferred to the C7 CPU, a user program is only transferred to the load memory of the C7 CPU and not automatically to the C7 CPU flash memory.

The content of the C7 CPU flash memory is not erased when the C7 CPU memory is reset.

If you want to erase the C7 CPU flash memory, you must do this explicitly using the PG function.

Downloading User Programs to the Flash Memory

After you have transferred the user program to the load memory of the C7 CPU with “PLC –> Download”, you must also copy the content of the load memory to the flash memory using the STEP 7 function

“Accesible Nodes –> MPI=xx –> PLC –> Copy RAM to ROM”.

Erasing the Flash Memory

The content of the flash memory can only be modified using the STEP 7 function “Copy RAM to ROM”. To erase the flash memory, follow the steps outlined below:

1. Using STEP 7, select all OBs, FBs, FCs, DBs on the C7 CPU.
2. Using STEP 7, delete the object selected on the C7 CPU. This deletes the load memory.
3. Now copy the empty load memory to the flash memory using the “Copy RAM to ROM” STEP 7 function.

Addressing, Parameter Assignment, and Functions of the C7 I/Os

4

Chapter Overview

4.1 Slot-Oriented Address Assignment for Signal Modules

Overview

With slot-oriented addressing (the default type of addressing) each slot number is assigned a module start address. Depending on the type of module, this is a digital or an analog address. This chapter explains which module start address is assigned to which slot number. You require this information to determine the module start addresses of the modules you are using.

Maximum Configuration of the C7-621

Figure 4-1 shows a possible configuration with a C7-CPU with additional S7-300 modules connected.

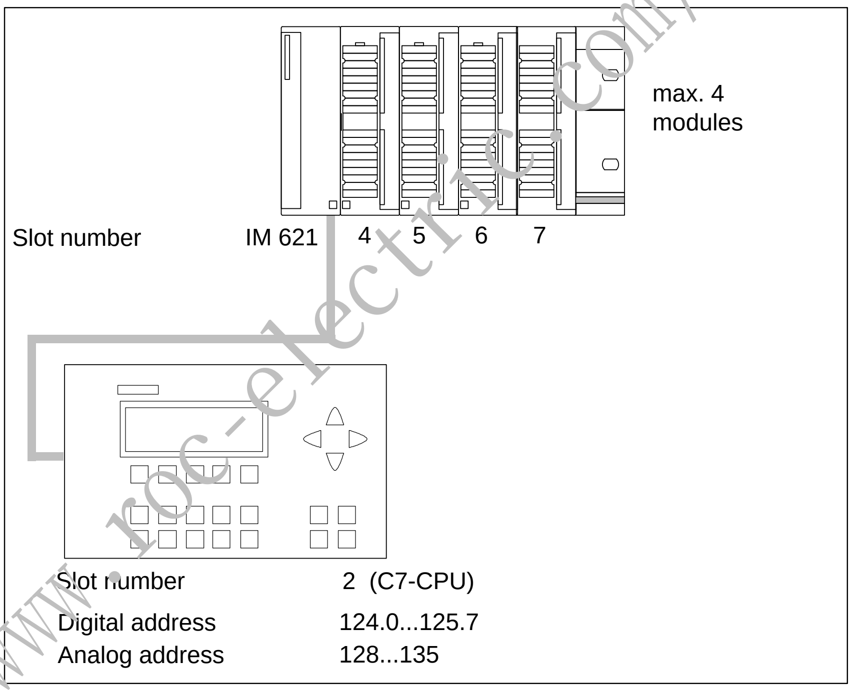


Figure 4-1 Maximum Configuration of a C7-621

C7-621 CPU

The C7-621 CPU occupies slot number 2.

IM 621

The IM 621 does not occupy a logical slot.

Additional S7-300 Racks

You can connect a maximum of one S7-300 rack with a maximum of 4 modules to the C7-621. How to connect a rack is explained in manual **/70/**.

Module Start Addresses

The table below lists the module start addresses related to the slots.

Table 4-1 Module Start Address Related to Slot Number

Module Start Address	Slot Number					
	2	3	4	5	6	7	Integrated I/Os
Digital			0	4	8	12	124
Analog	C7-CPU		256	272	288	304	128

Maximum Configuration of the C7-621 AS-i

Figure 4-2 shows a possible configuration of a C7-621 AS-i with the connection of additional S7-300 modules and AS-i modules.

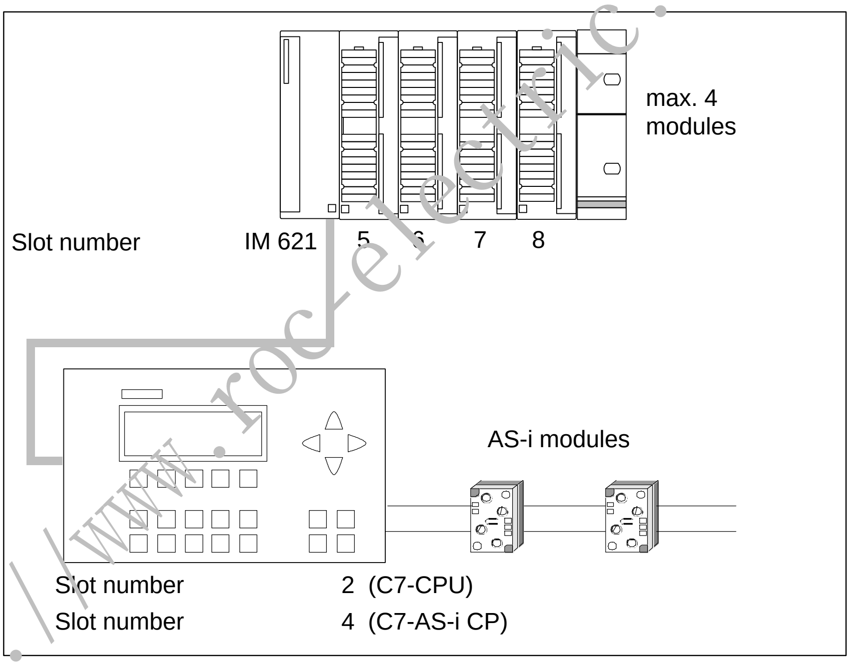


Figure 4-2 Maximum Configuration of a C7-621 AS-i

C7-621 CPU

The C7 CPU occupies slot number 2.

C7-AS-i CP

The C7 AS-i CP occupies slot number 4.

Additional S7-300 Racks

You can attach a maximum of one S7-300 rack with a maximum of 4 modules to the C7-621 AS-i. How this connection is made, is explained in manual **/70/**.

Additional AS-i Modules

You can connect 31 AS-i slaves to the C7-621 AS-i. This is explained in Chapter 5.

Module Start Addresses

The table below lists the module start addresses related to the slots.

Table 4-2 Module Start Address Related to the Slot Numbers

Module Start Address	Slot Number						
	2	3	C7-AS-i CP 4	5	6	7	8
Digital			0	4	8	12	16
Analog	C7-CPU		256	272	288	304	320

4.2 Addressing the C7 Digital I/Os

Overview

The following section describes the addressing of the integrated digital I/Os. You require this information to be able to address the channels of the digital inputs and outputs in the user program.

Address Range

The addresses of the digital I/Os for inputs and outputs are between 124.0 and 125.7.

Figure 4-3 shows the scheme for addressing the individual channels of the digital I/Os.

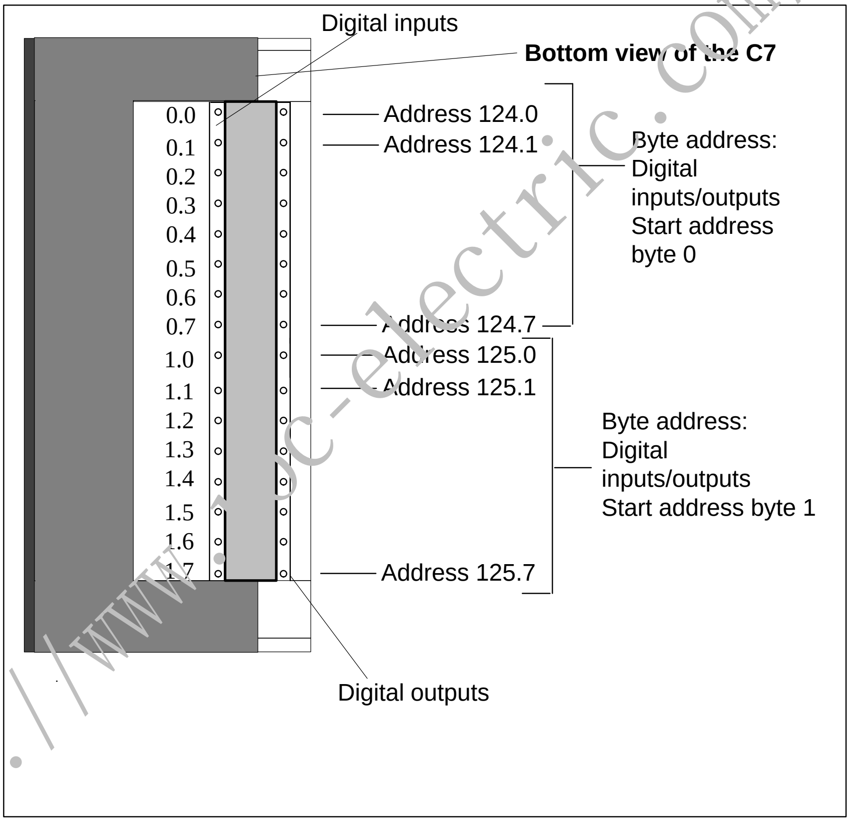


Figure 4-3 Addresses of the Digital Inputs/Outputs

4.3 Use and Functions of the C7 Analog I/Os

In this Section

This section explains the following:

- Basic terminology for analog value processing
- How to address analog I/Os
- The behavior of the individual analog input channels and the analog output channel

4.3.1 Addressing Analog I/Os

Analog Function Addresses

The address of an analog input or output channel is always a word address. The channel address is based on the module start address.

The analog I/Os have the same start addresses for the analog input and analog output channels.

Figure 4-4 shows which channel addresses result. You can see that for the analog I/Os, the analog input channel and the analog output channel are addressed beginning with the same start address.

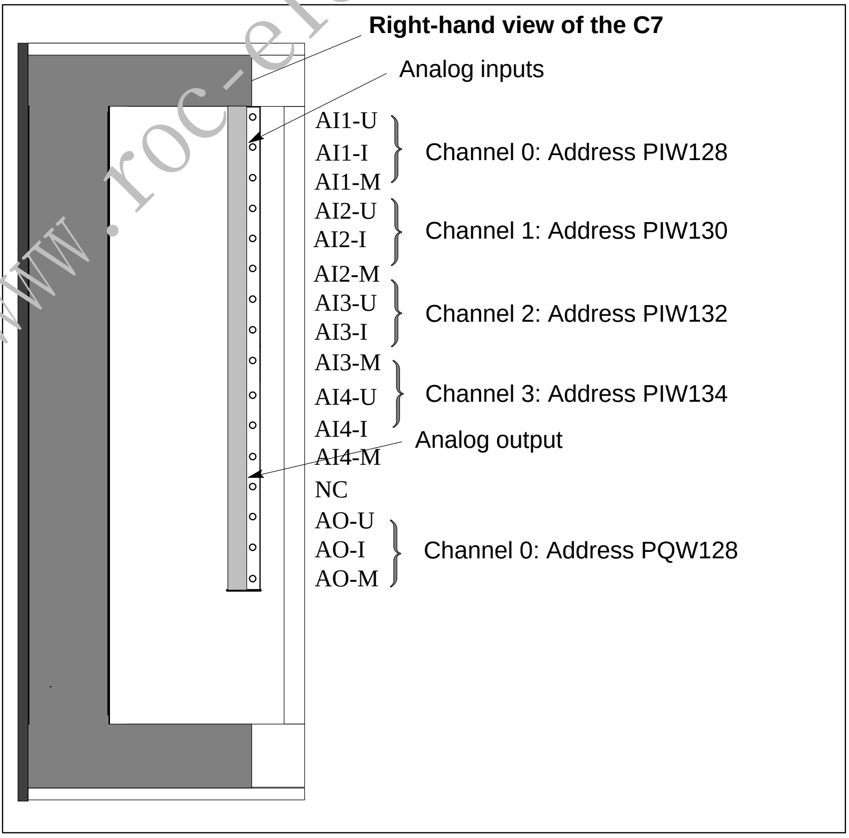


Figure 4-4 Addresses of the Analog Inputs and Outputs

4.3.2 Analog Value Representation

Overview

The analog value representation (an analog value in binary form) is the same in all C7 analog inputs and in the analog output.

This section shows the analog values for **all** measuring ranges or output ranges that can be used with the C7 analog I/Os.

Analog Value Representation

The digitalized analog value is the same for input and output values with the same nominal range.

Analog values are represented as two’s compliment.

Table 4-3 shows the analog value representation of the analog I/Os:

Table 4-3 Analog Value Representation

Resolution	Analog Value															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value of the bits	S	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

Sign

The sign (S) of the analog value is always in bit number 15:

- “0” → +
- “1” → –

12-Bit Resolution

The resolution is 12 bits. The analog value is entered left-justified in the ACCU. The unoccupied less significant bits are written with “**0**”.

Table 4-4 shows a bit pattern illustrating how the unused places are padded with “**0**” with the 12-bit resolution.

Table 4-4 Bit Pattern of a 12-Bit Analog Value (Example)

Resolution	Analog Value															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
12-bit analog value (including sign)	0	1	0	0	0	1	1	0	0	1	1	1	**0**	**0**	**0**	**0**

4.3.3 Analog Value Representation for the Measuring Ranges of the Analog Inputs

Overview

The tables in this chapter contain the digitalized analog values for the measuring ranges of the analog inputs.

Table 4-5 shows the representation of the binary analog values and the corresponding decimal or hexadecimal representation of the units of the analog values.

How to Read the Measured Value Tables

Table 4-6 contains digitalized analog values for various measuring ranges.

Since the binary representation of the analog values is always the same, this table contains only the comparison of the measuring ranges and the units.

This makes the table clearer and easier to read. The corresponding binary representation of the measured values can be seen in Table 4-5.

Measured Value Resolution

With the 12-bit resolution, the bits marked with “x” are irrelevant.

Table 4-5 Possible Resolutions of Analog Values

Resolution in Bits (Including Sign)	Units		Analog Value	
	Decimal	Hexadecimal	High Byte	Low Byte
12	16	10_H	S 0 0 0 0 0 0 0	0 0 0 1 x x x x

Voltage and Current Measuring Ranges

Table 4-6 contains the representation of the digitalized voltage measuring range for ± 10 V and the digitalized current ranges ± 20 mA.

Table 4-6 Representation of the Digitalized Measuring Value of the Analog Inputs (Voltage and Current Measuring Ranges)

Measuring Range ± 10 V	Measuring Range ± 20 mA	Units		Range
		Decimal	Hexadecimal	
≥ 11.759	≥ 23.516	32767	$7FF_H$	Overflow
11.7589	23.515	32511	$7EFF_H$	
:	:	:	:	Overload range
10.0004	20.0007	27649	$6C01_H$	
10.00	20.000	27648	$6C00_H$	
7.50	14.998	20736	5100_H	
:	:	:	:	Nominal range
- 7.50	- 14.998	-20736	$AF00_H$	
- 10.00	- 20.000	-27648	9400_H	
- 10.0004	- 20.0007	-27649	$93FF_H$	
:	:	:	:	Underload range
- 11.759	- 23.516	-32512	8100_H	
<- 11.759	<- 23.516	-32768	8000_H	Underflow

4.3.4 Analog Value Representation for the Output Range of the Analog Outputs

Table for Output Ranges

Table 4-7 contains the analog output range of the analog output.

Voltage/Current Output Ranges

Table 4-7 contains the representation of the voltage output range ± 10 V and the current output range ± 20 mA.

Table 4-7 Representation of the Analog Output Range of the Analog Output (Voltage/Current Output Ranges)

Output Range ± 10 V	Output Range ± 20 mA	Units		Range
		Decimal	Hexadecimal	
0	0	> 32511	> $7EFF_H$	Overflow
11.7589	23.515	32511	$7EFF_H$	
:	:	:	:	Overload range
10.0004	20.0007	27649	$6C01_H$	
10.0000	20.000	27648	$6C00_H$	
:	:	:	:	
0	0	0	0_H	
0		:	:	Nominal range
:	:	- 6912	$E500_H$	
		- 6913	$E4FF_H$	
		:	:	
-10.0000	- 20.000	- 27648	9400_H	
10.0004		- 27649	$93FF_H$	Underload range
	:	:	:	
-11.7589	23.515	- 32512	8100_H	
0	0	< - 32513	< 8100_H	Underflow

4.4 Examples of Programming the Analog I/Os

Overview

The following examples of programming the analog I/Os will help to familiarize you with programming the C7 I/Os.

The following examples are used:

- Block for scaling analog output values
- Block for scaling analog input values

4.4.1 Block for Scaling Analog Output Values

Block Function

This block (FC127) converts the setpoint specified as a floating-point number in a memory double word to the corresponding hexadecimal pattern (=analog value) to be output to a peripheral output word. A simple calculation is programmed.

1. First the setpoint is related to the total range (RANGE_DEC) resulting from the difference (upper limit – lower limit).

   The result is a percentage of the absolute setpoint. This is the same in the floating-point representation and in the hexadecimal representation.
2. Finally, depending on whether the measuring range is monopolar or bipolar, the entire range (RANGE_HEX) is calculated from the difference (UL – LL).
3. The previously calculated percentage (PERCENT) is now related to the entire hexadecimal range (RANGE_HEX).

   The result is the absolute value to be output.
4. Finally, the lower limit (LL) is added to this value as an offset.
5. The resulting bit pattern is then output.

Summary of the Formulas

PERCENT = (setpoint – lower limit) / (upper limit – lower limit)
RANGE_DEC = upper limit – lower limit
RANGE_HEX = UL – LL
Channel = PERCENT * RANGE_HEX + LL

The Statements in FC127

The program in FC127 contains the following statements:

```
FUNCTION FC 127 : void

var_input
                lowerlimit : DWORD
                upperlimit : DWORD
                setpoint : DWORD;
end_var

var_temp
                LL : DWORD;
                UL : DWORD;
                RANGE_DEC : DWORD;
                RANGE_HEX : DWORD;
                PERCENT : DWORD;
end_var

BEGIN

//*** Determine between monopolar or bipolar range ***
L lowerlimit;                   // lowerlimit negative?
L 0.0;                          // YES => bipolar range
<R;
JC bipo;

L DW#16#0000_0000;              // Monopolar range lower limit
T LL;
JU rech;

bipo: NOP 0;
L W#16#9400;                    // Bipolar range lower limit
ITD;
T LL;

//*** Calculation of the range (hexadecimal) ***
rech: NOP 0;

L W#16#6C00;                    // upper limit for monopolar and bipolar
                                //range same
ITD;
L LL;
-D;
T RANGE_HEX;                    // Buffer difference

//*** Relate setpoint to entire range ***
L upperlimit;                   // Calculate range
L lowerlimit;
-R;
T RANGE_DEC;

L setpoint;                     // Relate setpoint to entire range
L lowerlimit;
-R;
L RANGE_DEC;
/R;
T PERCENT;
```

```
//*** calculate hex pattern to output ***
L RANGE_HEX;                    // Relate hex value to entire range
DTR;
L PERCENT;
*R;
L LL;                           // Add offset
DTR;
+R;
RND;                            //Convert floating-point
                                // to 32 bit integer
          T Channel;            //Output result
```

Calling FC127 in OB1

The FC127 call is explained below in the form of an example.

Before the call, the range limits and the setpoint must be routed to memory double words. This is necessary so that it is possible to work with variable values. Generally “upperlimit” and “lowerlimit” are fixed; the “setpoint” is variable.

This can be achieved by setting the “upperlimit” and “lowerlimit” to “REAL” in the declaration section of FC127. To remain more flexible for test purposes, this variant was not used.

Sequence of statements in OB1

```
ORGANIZATION_BLOCK OB1
var_temp
          start_info : array [0..19] of byte;
end_var
BEGIN

L -10.0;
T MD0;

L 10.0;
T MD4;

L 2.2;
T MD8;

CALL FC 127 (
                    lowerlimit := MD0,
                    upperlimit := MD4,
                    setpoint := MD8,
                    Channel := PQW272
          );

END_ORGANIZATION_BLOCK
```

4.4.2 Block for Scaling Analog Input Values

Block Function

This block (FC126) converts the process value available as a hexadecimal number in a peripheral input word to a corresponding floating point number that can be entered in a memory double word. A simple calculation is programmed.

1. The process value is first related to the total range (RANGE_HEX) resulting from the difference (UL – LL).

   This produces the absolute process value as a percentage. This is the same in the floating point representation and in the hexadecimal representation.

2. Finally depending on whether the measuring range is monopolar or bipolar, the total range is calculated in floating point representation from the difference (UL – LL).

3. The percentage calculated above (PERCENT) is now related to the entire floating point range.

   The result is the value read in.

4. Finally, the lower limit (LL) is added to this value as an offset.

5. The resulting floating point number is output.

Summary of the Formulas

PERCENT = (channel – LL) / (UL – LL)
RANGE_HEX = UL – LL
Process value = PERCENT * (upperlimit – lower limit) + lower limit

Sequence of Statements in FC126

The program in FC126 contains the following statements:

```
FUNCTION FC 126 : void
var_input
            lowerlimit : DWORD;
            upperlimit : DWORD;
            Channel : WORD;
end_var
var_output
            Actvalue : DWORD;
end_var
var_temp
            LL : DWORD;
            RANGE_HEX : DWORD;
            PERCENT : DWORD;
end_var
BEGIN
```

```
// *** Determine between monopolar or bipolar range ***
L lowerlimit;                   // lowerlimit negative?
L 0.0;                          // YES => bipolar range
<R;
JC bipo;

L DW#16#000_00000;              // Monopolar range lower limit
T LL;
JU rech;

bipo: NOP 0;
L W#16#9400;                    // Bipolarer range lower limit
ITD;
T LL;

// *** Calculation of the range (hexadecimal) ***
rech: NOP 0;

L W#16#6C00;                    // Upper limit for monpolar and bipolar
                                // range same
ITD;
L LL;
-D;
T RANGE_HEX;                    // Buffer difference

// *** Relate actual value to entire range ***
L Channel;                      // Relate input value to entire range
ITD;
L LL;
-D;
DTR;
L RANGE_HEX;
DTR;
/R;
T PERCENT;

// *** Calculate floating-point number ***
L upperlimit;                   // Calculate floating-point range
L lowerlimit;
-R;
L PERCENT;
*R;
L lowerlimit;
+R;
T Actvalue;

END_FUNCTION
```

Calling FC126 in OB1

The FC126 call is explained below in the form of an example.

Before the call, the range limits must be routed to memory double words. This is necessary so that it is possible to work with variable values. Generally “upperlimit” and “lowerlimit” are fixed.

This can be achieved by setting the “upperlimit” and “lowerlimit” to “REAL” in the declaration section of FC126. To remain more flexible for test purposes, this variant was not used.

Sequence of statements in OB1

```
ORGANIZATION_BLOCK OB1
var_temp
        start_info : array [0..19] of byte;
end_var
BEGIN

L 10.0;
T MD4;

L -10.0;
T MD0;

CALL FC 126 (
                lowerlimit := MD0,
                upperlimit := MD4,
                channel := PEW272,
                Actvalue := MD8
                );
END_ORGANIZATION_BLOCK
```

Introduction to AS-i 5

Overview

Chapters 5 and 6 describe the system concept, the functions and programming of the C7-621 AS-i.

To familiarize yourself with the topic of AS-i, we recommend the following reading:

- Read the AS-i brochure

  Actuator-Sensor Interface

  Order number E20001-P285-A497-V2-7600

that can be ordered from all Siemens offices and service centers.

- For a general overview, read this chapter.
- The functions, programming, and operation of the C7-621 AS-i are explained in Chapter 6.

Chapter Overview

Note

The As-i attachment is implemented with the C7-621 AS-i with an integrated AS-i master module. In this chapter, the C7-AS-i CP is called the AS-i master module in the C7-621 AS-i.

5.1 Introduction

Overview

The actuator/sensor interface, abbreviated to **AS-i**, is a connection system for the lowest process level in automation systems. The cable harnesses previously found at this level are replaced by a single electrical cable, the AS-i cable. Using the AS-i cable and the AS-Interface master, the simplest binary sensors and actuators can be connected to the control devices at the field level via AS-i modules.

AS-Interface is the product name within the SIMATIC system for this AS-i technology which has been proposed for international standardization.

The following diagram illustrates the position of AS-Interface and AS-i within the automation system.

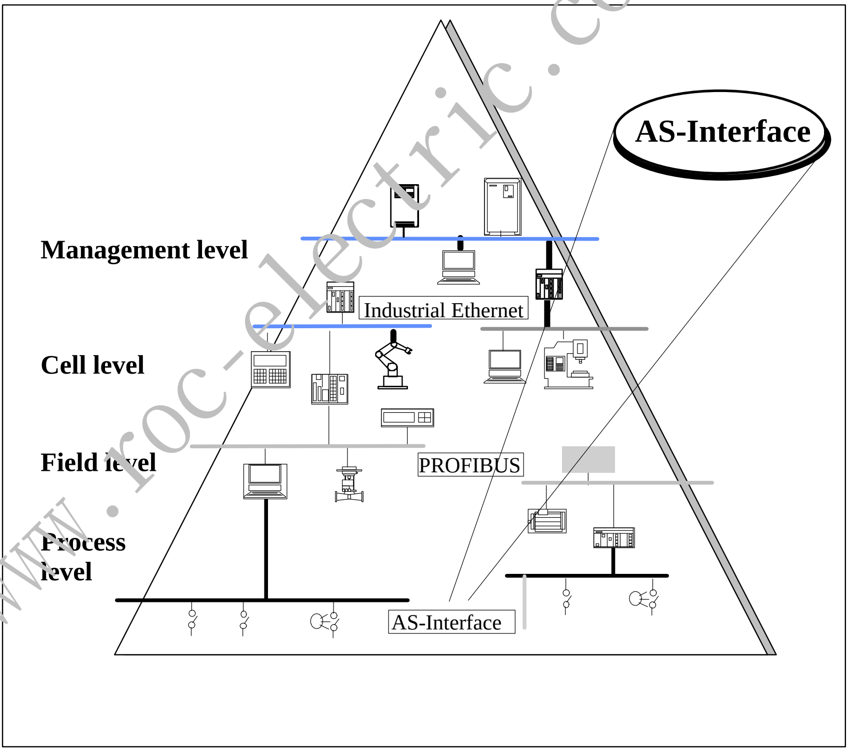


Figure 5-1 Position of AS-i in the Automation Hierarchy

Main Characteristics of AS-i

AS-Interface is distinguished by several main characteristics:

- AS-Interface is optimized for connecting binary sensors and actuators. The AS-i cable is used both for data exchange between the sensors/actuators and the master as well as for power supply to the sensors and actuators.
- Simple and cost-effective wiring: simple installation with the “penetration” technique, high flexibility with tree-like wiring.
- Fast cycle times: the AS-Interface master requires a maximum of 5 ms for cyclic data exchange with up to 31 stations.
- Stations on the AS-i cable can either be sensors/actuators with an integrated AS-i connector or AS-i modules to which up to four conventional binary sensors/actuators can be connected.
- With AS-i modules, up to 124 actuators and 124 sensors can be operated on the AS-i cable

5.2 AS-i System Concept

System Components

The following system components make up an AS-i network:

Table 5-1 System Components for an AS-i Network

AS-i	Explanation
C7-AS-i CP	C7-621 AS-i
AS-i Module	In AS-i, a module concept is defined so that there is a block-like interconnection of AS-i stations (sensors and actuators) via so-called AS-i modules. The following distinction is made between user modules: • The **active** AS-i module with an integrated AS-i chip; using this, up to four conventional sensors and actuators can be connected. • The **passive** AS-i module; this functions as a distributor and provides a connection for up to four sensors and actuators with an integrated AS-i chip.
AS-i Cable	The AS-i cable, designed as an unshielded 2-wire cable, transfers signals and the power supply for the sensors and actuators connected using AS-i modules.
AS-i Power Supply Unit	The AS-i power supply unit supplies power for the AS-interface/AS-i stations connected to the AS-i cable. For actuators with particularly high power requirements, the AS-i power supply unit can, if necessary, be supplemented by a separate auxiliary power supply unit (for example using a special user module).
Sensors with an Integrated AS-i Chip	These sensors can be connected directly to the AS-i cable. As slaves in the AS-i network, they have four bits of data available with which additional information over and above the basic switching information can be transmitted (e.g. coil failure).

Example of an AS-i Network

The following diagram illustrates how the described components can be interconnected. The tree structure is particularly clear. Networking is not restricted to one type of cable. If necessary, appropriate modules or “T pieces” can be used to change to a simple 2-wire cable.

Note

The technical requirements such as the conductor cross-section, the voltage drop and the length of cables must be met in compliance with the AS-i specification (see /1/ and /2/).

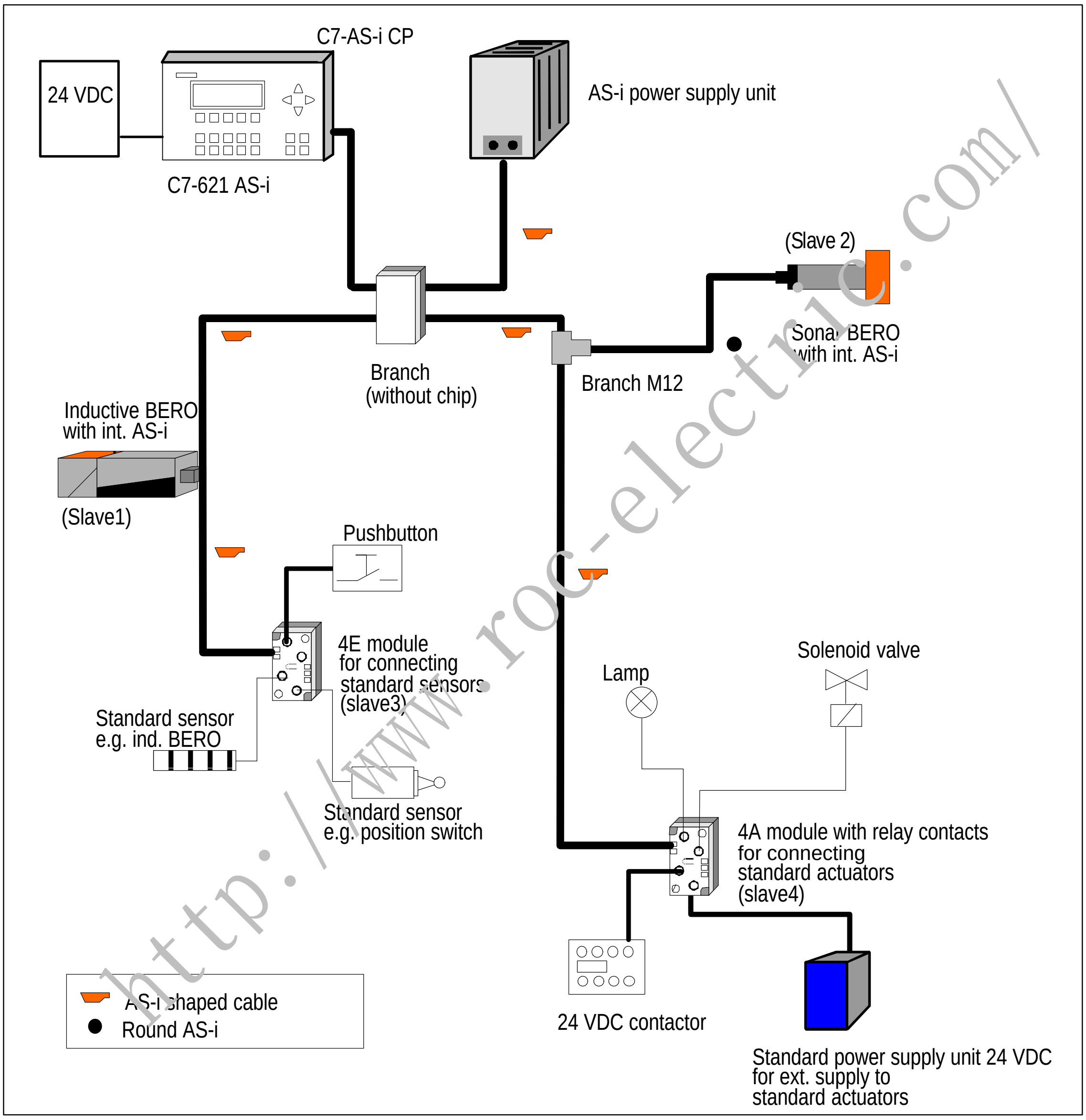


Figure 5-2 Example of the Configuration of an AS-i Network with Nodes

5.2.1 System Characteristics and Important Data

How AS-i Functions

The AS-interface/AS-i system operates as outlined below:

- Master-slave access technique

  The AS-interface is a “single master system”, this means that there is only one master per line section which controls the data exchange. It polls all the slaves one after the other and waits for a reply.

- Electronic address setting

  The address of the slave is its identifier. This only occurs once within a AS-interface/AS-i system. The setting can either be made using the integrated C7-621 addressing function (see section. 6.3.2) or using a programming and diagnostic device. This address is then stored permanently on the slave.

  When shipped, the slaves always have the address “0”.

- Operating reliability and flexibility

  The transmission technique used (current modulation) guarantees high operating reliability. The master monitors the voltage on the cable and the transferred data. It detects transmission errors and the failure of slaves and sends a message to the PLC. The user can then react to this message.

  Exchange or addition of slaves during normal operation does not impair communication with the other stations.

Physical Characteristics of AS-i

The most important physical characteristics of the AS-interface/AS-i system and its components:

- 2-wire cable for data and power supply

  A simple 2-wire cable with a cross-section of 2 x 1.5 mm^2 can be used. Shielding or twisting is not necessary. Both the data and the power supply are transferred on this cable. The power available depends on the AS-i power supply unit used.

  For optimum wiring, the mechanically coded AS-i cable is available preventing the connections being reversed and making simple contact with the AS-i user modules using the penetration technique.

- Tree structure network with a cable length up to 100 m

  The “tree structure” of the AS interface allows any point on a cable section to be used as the start of a new branch. Loops are not permitted. The total length of all subsections can be up to 100 m.

- Direct integration

  Practically all the electronics required for a slave has been integrated on a special IC. This allows the AS-i connector to be integrated directly in binary actuators or sensors. All the required components can be installed within a space of approximately 2 cm^3.

- More functions, better customer utilization

  Direct integration allows devices to be equipped with a wide range of functions. Four data and four parameter lines are available. Such “intelligent” actuators/sensors increase the possibilities, for example monitoring, parameter assignment, wear or pollution checks etc.

- Additional voltage supply when more power is required

  An external source of power can be provided for slaves with a higher power requirement (see /1/ in Appendix D).

Limits

- Cycle time max. 5ms

  The AS-interface uses constant message lengths. Complicated procedures for controlling transmission and identifying message lengths or data formats are not required. This makes it possible for a master to poll all connected slaves within a maximum of 5ms and to update the data both on the master and slave.

- Maximum of 31 Slaves

  Slaves are the input and output channels of AS-interface/AS-i system. They are only active when called by the master. They trigger actions or transmit reactions to the master when commanded. Each slave is identified by its own address (1 to 31).

- Maximum of 124 binary stations

  Each slave can transmit 4 bits of data. Special modules allow each of these bits to be used for a binary actuator or a binary sensor. This means that there can be a maximum of 124 binary nodes on one AS-i cable. All typical actuators or sensors can be connected to the AS-interface/AS interface in this way. The modules are used as distributed inputs/outputs.

5.3 The C7-AS-i CP for C7-621 AS-i

Overview

The C7-AS-i CP is integrated in controllers of the C7-300 series.

The C7-AS-i CP is designed for two types of operation:

Standard operation

Standard operation is intended for particularly simple installation and programming of AS-i.

In this type of operation, the AS-i CP operates like an I/O module. It occupies 16 input and 16 output bytes in the analog area of the controller. In this operation, the slaves are assigned the default value of the parameter (F_H) stored on the interface module.

Parameters and commands cannot be transferred in standard operation.

Extended operation

In extended operation, the full range of functions according to the AS-i master specification is available (see /2/). For this mode, an additional function block is available for the PLC which is supplied on diskette along with this manual. By using this function block, master calls can also be executed by the control program in addition to standard operation.

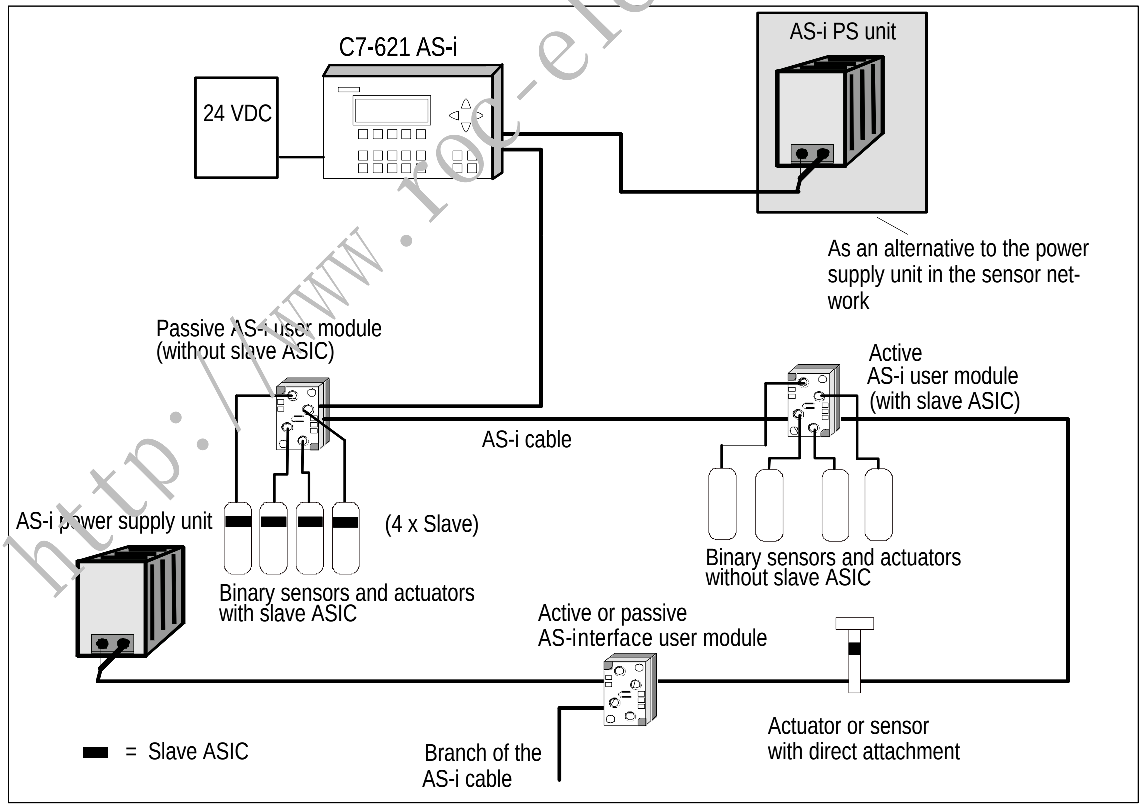


Figure 5-3 AS-i Configuration

5.4 Further AS-i System Components

Overview

As well as the AS-interface masters described in this manual, the components of the AS-i transmission system and the AS-i slaves are also required. The following sections provide an overview of the basic characteristics and interaction of these components.

Due to the continuing development of new AS-i system components, a complete presentation of all the currently available components is not possible. Refer to the available system catalogs /3/ and ask your Siemens office for more information.

5.4.1 The AS-i Cable

Overview

The AS-i cable (shaped cable) allows simple and fast installation of a AS-interface system. The AS-i cable is a rubberized 2-wire cable (2 x 1.5 mm^2). The profile cross section prevents stations being connected with incorrect polarity.

Attaching to the AS-i Cable

The AS-i cable is contacted using the penetration technique. Contact blades penetrate the rubber jacket and make contact with the two wires. This guarantees low contact resistance and ensures a reliable data connection. The cable does not need to be cut, have its insulation removed or be screwed. For this type of connection, there are coupling modules designed for the penetration technique.

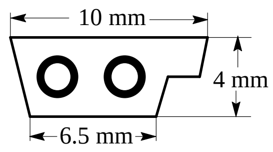


Figure 5-4 Cross Section of the Cable

The jacket of the AS-i cable is rubber. If modules need to be moved after they have been connected to the AS-i cable this is possible without causing any problems. The AS-i cable is “self-healing”. This means that the holes made by the contact blades in the rubber jacket of the cable close themselves and revert to the type of protection IP67. When the cable is installed in an AS-i module, the cable seals the openings. The type of protection IP67 is therefore achieved.

Using Other 2-Wire Cables

Apart from the special AS-i cable, any 2-wire cable with a cross-section of 2 x 1.5 mm^2 can be used. Shielding or twisting is not necessary. For the transition from the special AS-i cable to a different cable (e.g. a standard round cable) there is a special module available without integrated electronics (transition from the AS-i cable to four M12 connectors and transition from the AS-i cable to one M12 connector).

5.4.2 AS-i Modules

Overview

Within the AS-i system, the AS-i modules can be compared with input and output modules. They connect actuators or sensors to the C7-AS-i CP.

Attaching AS-i Modules

The actuators/sensors are connected via M12 connectors. The pin assignment corresponds to DIN IEC 947 5-2. The modules with dimensions of approximately 45 x 45 x 80 mm are used locally on the machine itself. They are connected via the AS-i cable and have the degree of protection IP67.

The following modules must be distinguished:

- The **active** AS-i module with integrated AS-i chip; using this, conventional sensors and actuators can be connected. Every normal actuator or sensor can therefore be networked via AS-interface.
- The **passive** AS-i module: This does not contain its own electronics and allows the connection of AS-i sensors and actuators with integrated AS-i chips.

The modules are designed so that a uniform electro-mechanical interface to the AS-i cable can be created. This is achieved with the uniform lower section of the module, which is therefore also known as a coupling module.

5.4.3 AS-i Repeater / Extender

Overview

The AS-Interface repeater and extender is designed for use in an actuator-sensor interface environment.

This device is used to extend the possible span of the actuator-sensor interface beyond the maximum of 100 m. An existing 100 m segment can be extended by up to two further 100 m segments. The possible configurations are described in the installation instructions.

Repeater

The AS-Interface repeater is used when slaves are operated on all cable segments. Each AS-Interface segment (before and after the repeater) then requires its own AS-Interface power supply unit. The repeater has the following features:

- Extension of the cable length to a maximum of 300 m
- Slaves can be used on both sides of the repeater
- A separate AS-Interface power supply unit is required for each segment
- The two cable sections are electrically isolated
- Separate display of the correct voltage for each side
- Installed in a standard user module housing

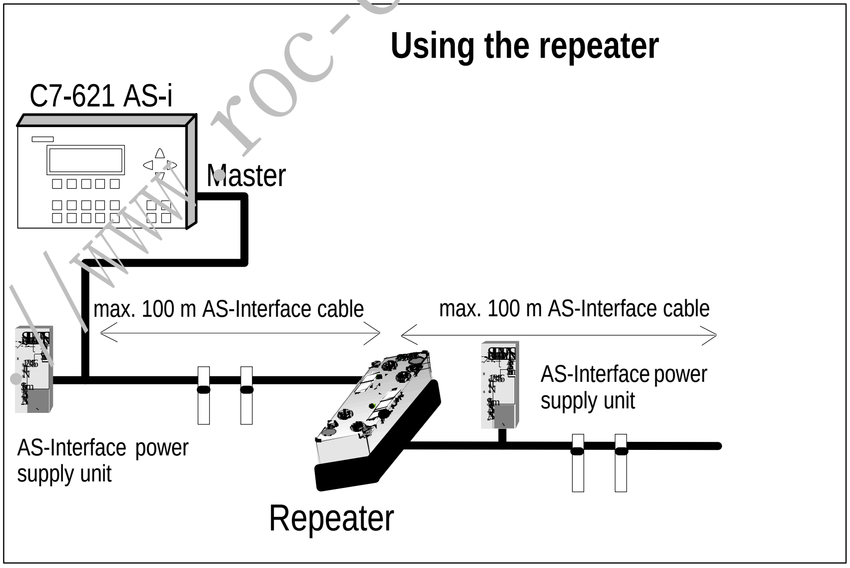


Figure 5-5 Using the Repeater

Extender

The AS-Interface extender is used when the master is installed at a greater distance from the actual AS-Interface installation:

- The master can be up to 100 m from the AS-Interface segment
- Slaves are only attached on the side away from the master
- Power supply is only required on the side away from the master
- No electrical isolation of the two cable sections
- Display of the correct voltage
- Installed in a standard user module housing
- The FK-E coupling module is used as the base

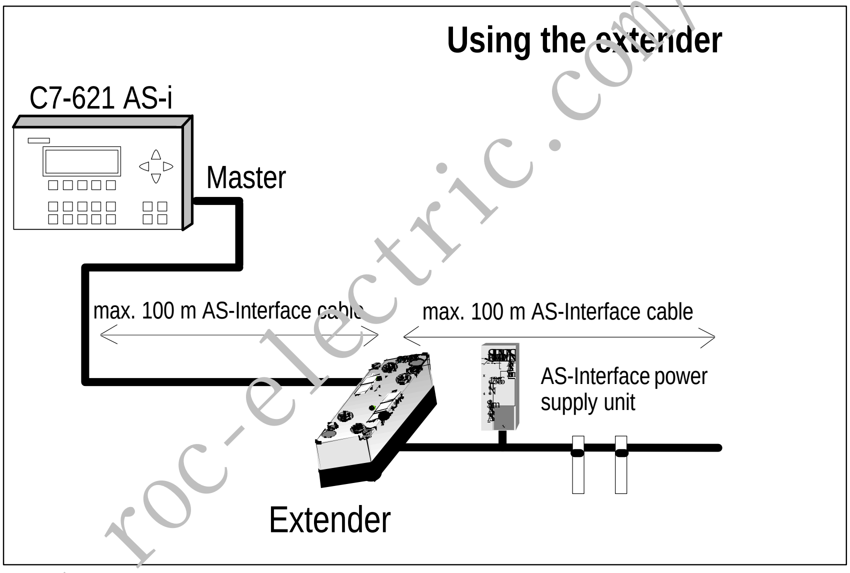


Figure 5-6 Using the Extender

5.4.4 Sensors/Actuators with an Integrated AS-i Connection

Overview

The AS-i chip is particularly important in the AS-i system. It permits the use of sensors and actuators with an integrated AS-i connection.

2cm^3 Slave Electronics

The electronics required by the AS-i chip requires very little space.

The AS-i chip provides the sensor with four data outputs and inputs as well as four parameter outputs. With these additional parameter outputs, it is possible to assign parameters to intelligent sensors via the AS-i cable (for example setting various switching ranges with a sonar BERO).

Sensors with an integrated AS-i connection are available from Siemens and other manufacturers. For further information refer to the catalogs.

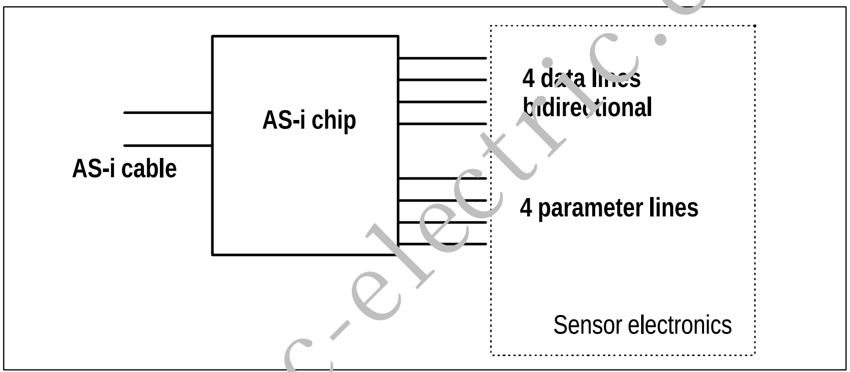


Figure 5-7 Block Diagram of an AS-i Slave

5.5 The Master Mode - Commands, Sequence, Programming

Overview

The tasks and functions of a C7-AS-i CP are described below.

This section is important for understanding the functions, modes and interfaces available with the AS-interface.

How the Master/Slave Principle Functions

The AS-interface-AS interface operates on the master-slave principle. This means that the C7-AS-i CP connected to the AS-i cable controls the data exchange with up to 31 slaves via the interface to the AS-i cable.

The following diagram illustrates the two interfaces of the AS-interface master CP and the AS-i CP or AS-i slave. The process data and parameter assignment commands are transferred via these data paths between the C7 CPU and the master CP.

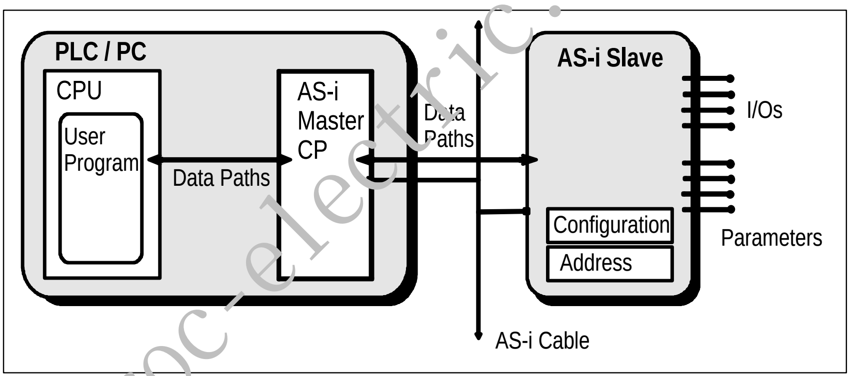


Figure 5-8 Master-Slave Structure

AS-i Specification

The AS-i master specification distinguishes masters with different ranges of functions known as a “profile”.

With the C7-621 AS-i, two different master classes are distinguished (M0, M1). The AS-i specification stipulates which functions a master in a particular class must be able to perform.

The profiles have the following practical significance:

- Master profile M0:

  The master can exchange I/O data with the individual stations. The master is configured by using the station configuration found on the cable known as the “desired configuration”.

- Master profile M1:

  This profile covers all the functions according to the AS-i master specification.

In keeping with this graded concept, different function modes are available.

These profiles are also used to match the handling of the C7-AS-i CP to the particular requirements especially keeping handling simple for the normal mode:

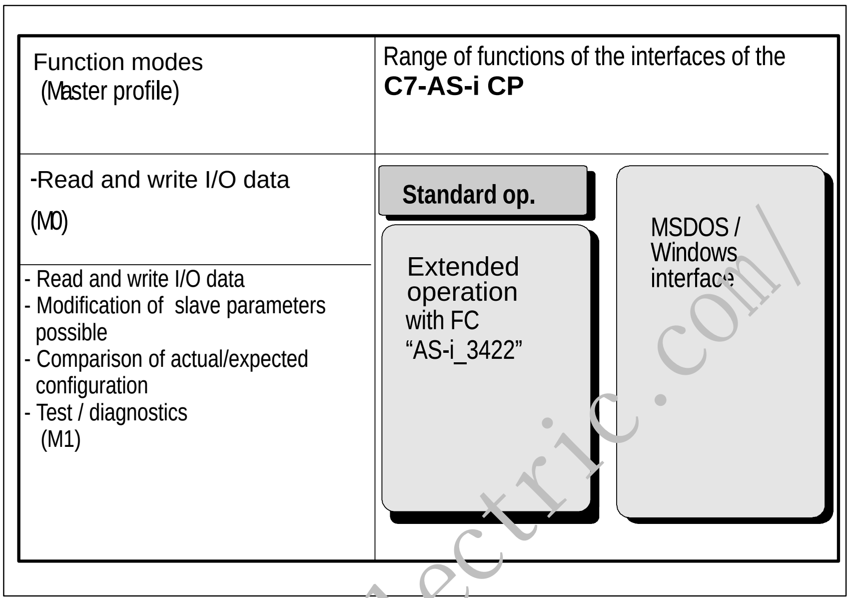


Figure 5-9 Range of Functions of the Interfaces and Operating Modes

Standard Operation with the C7-AS-i CP

The C7-AS-i CP can control the connected AS-i slaves in standard operation without requiring extra instructions in the PLC user program.

The operating and display elements of the C7-621 AS-i are adequate to initialize the slaves and to trigger and monitor cyclic operation. Standard operation is adequate for many tasks. This meets the requirements of master profile M0.

Note

If you decide to use standard operation, you can skip the remaining sections in this chapter. Continue reading in Section 6.2 to find out the steps required for installation and operating the C7-AS-i CP in standard operation.

Extended Operation with the AS-i CP

In extended operation, the complete range of possibilities stipulated in the AS-i specification for controlling the AS-i slaves with the user program is available.

To use these functions, you require additional software.

The AS-i Slave

The AS-i slave contains an integrated circuit (ASIC) which allows coupling of an AS-i device (sensor/actuator) to the common bus cable to the AS-i master. The integrated circuit contains the following components:

- 4 configurable data inputs and outputs
- 4 parameter outputs

The operating parameters, configuration data with I/O assignment, identification code and slave address are stored in additional memory (EEPROM).

- **I/O data**

  The user data for the automation components that were transferred from the C7-AS-i CP to the AS-i slave are available at the data outputs. The values at the data inputs are made available to the C7-AS-i CP when the AS-i slave is polled.

- **Parameters**

  Using the parameter outputs of the AS-i slave, the C7-AS-i CP can transfer values that are not interpreted as simple data. These parameter values can be used to control and switch over between internal operating modes of the sensors or actuators. It could, for example, be possible to update a calibration value during various operating phases. This function is possible with slaves with an integrated AS-i connection providing they support the function in question.

- **Configuration**

  The input/output configuration (I/O configuration) indicates which data lines of the AS-i slave are used as inputs, outputs or as bi-directional outputs. The I/O configuration (4 bits) is stipulated by the slave manufacturer and can be found in the description of the AS-i slave (an overview of codings can be found in /1/).

  In addition to the I/O configuration, the type of the AS-i slave is described by an identification code. The identification code for each AS-i slave is coded in 4 bits. This is also specified by the manufacturer and can be found in the manufacturer’s description of the AS-i slaves.

5.5.1 Operating Phases and Functions

Information / Data Structure

Before introducing you to the operating phases and the functions during these operating phases, a brief outline of the information structure of the AS-interface master/slave system is necessary.

In the following diagram, the data fields and lists of the system are configured in the system structure diagram you saw in the previous section.

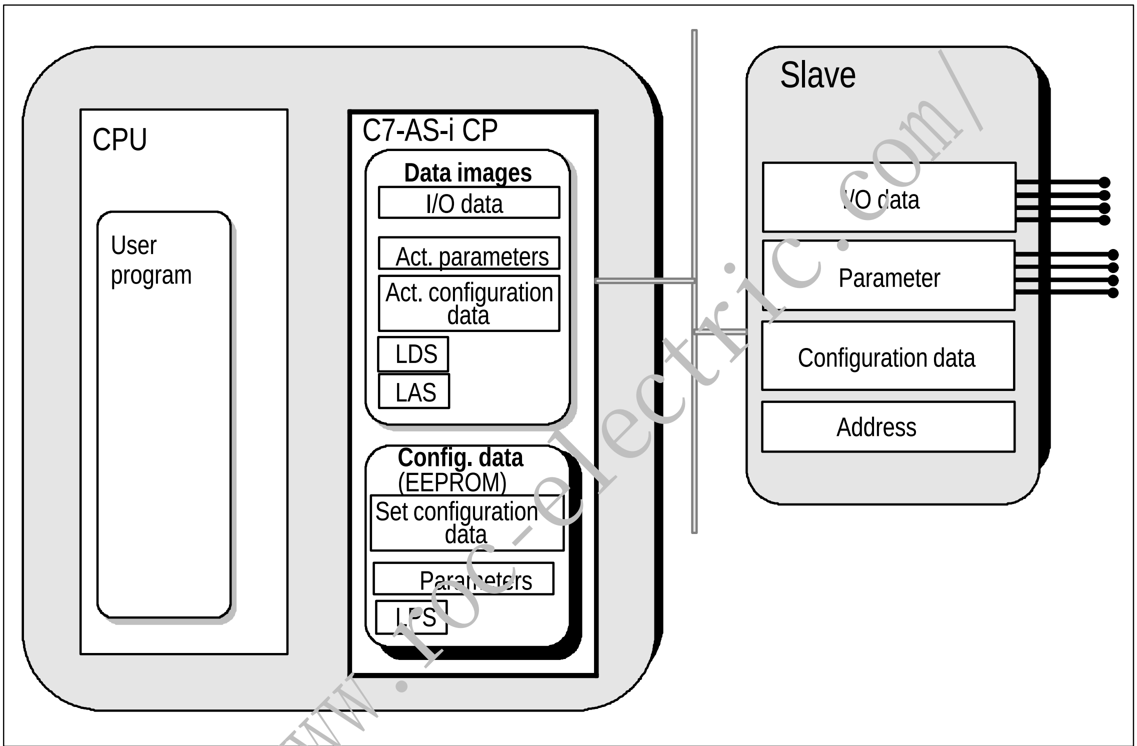


Figure 5-10 Information / Data Structure on the C7-AS-i CP and AS-i-Slave

The following structures are found on the C7-AS-i CP:

- **Data images**
  These contain temporarily stored information:
  - I/O data
    The process input and output data
  - Actual parameters
    The actual parameters are an image of the parameters currently on the slave.
  - Actual configuration data
    The actual configuration data field contains the I/O configurations and ID codes of all connected AS-i slaves once these data have been read from the AS-i slaves.
  - List of detected slaves (LDS)
    The LDS specifies which slaves were detected on the AS-i bus.
  - List of activated slaves (LAS)
    The LAS specifies which slaves were activated by the AS-i master. I/O data are only exchanged with activated slaves.
- **Configuration data**
  These are non-volatile data (e.g. stored in an EEPROM), which are available unchanged even following a power failure.
  - Desired configuration data
    These are selectable comparison values which allow the configuration data of the detected slaves to be checked.
  - Parameters
  - List of permanent slaves (LPS)
    This list specifies the AS-i slaves expected on the AS-i cable by the C7-AS-i CP. The C7-AS-i CP checks continuously whether all the stations specified in the LPS exist and whether their configuration data match the desired configuration data.

The AS-i slave has the following structures:

- **I/O data**
- **Parameters**
- **Configuration data**
  The configuration data contain the I/O configuration and the ID code of the slave.
- **Address**
  The slaves have address “0” when installed. To allow a data exchange, the slaves must be programmed with addresses other than “0”.

  The address “0” is reserved for special functions.

The Operating Phases

The following diagram illustrates the individual operating phases.

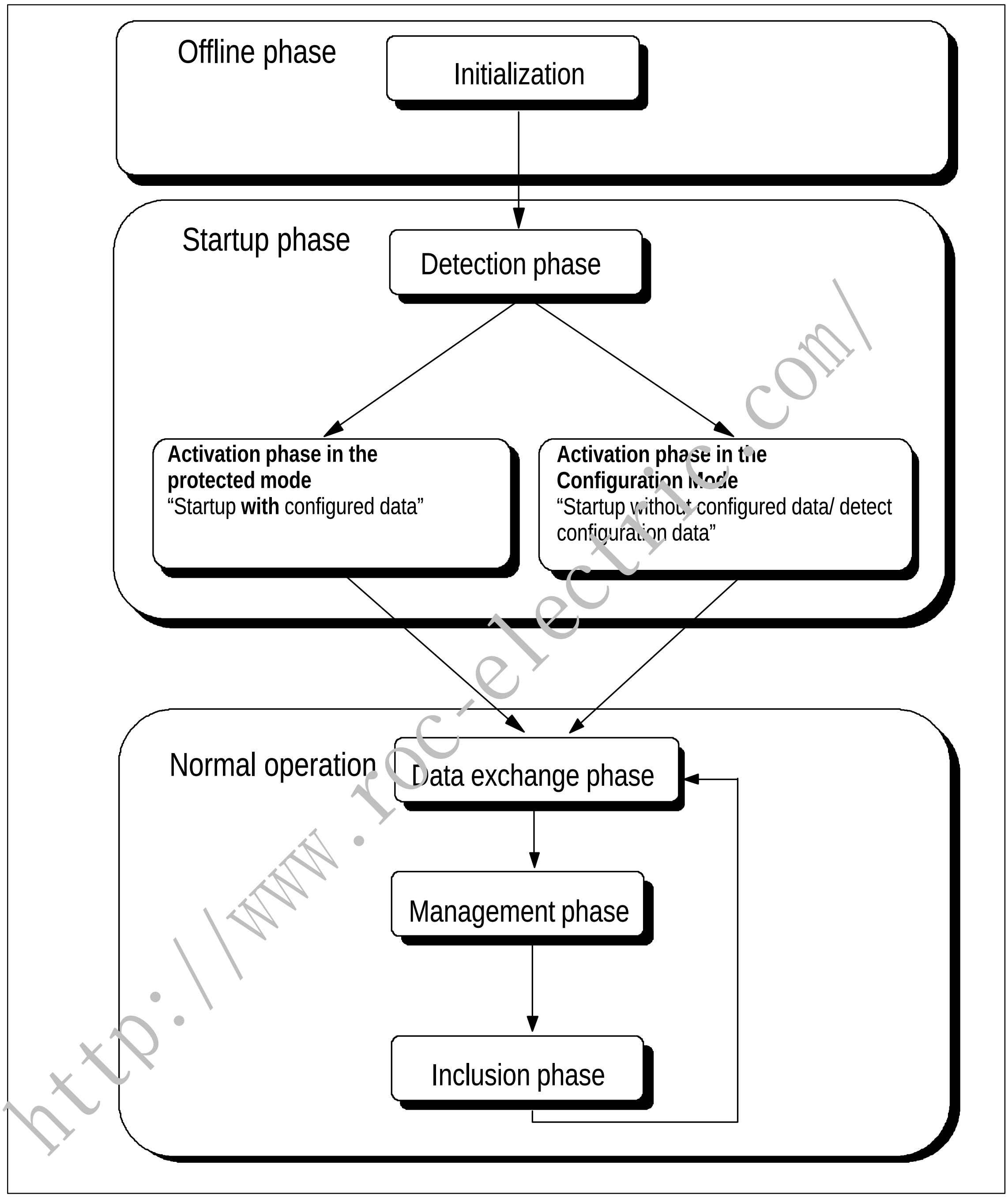


Figure 5-11 The Operating Phases

Offline / Initialization Phase

The initialization which is also known as the offline phase, establishes the basic status of the master. The module is initialized after switching on the power supply or following a cold restart during operation. During the initialization, the images of all the slave inputs and the output data from the point of view of the application are set to the value “0” (inactive).

After switching on the power supply, the configured parameters are copied to the parameters field so that the subsequent activation is with the preset parameters. If the C7-AS-i CP is re-initialized during operation, the values from the parameters field which may have changed in the meantime are retained.

The Startup Phase

Startup includes the following phases:

- **Detection phase: detecting the slaves during the startup phase**

  During start-up or after a reset, the C7-AS-i CP runs through a startup phase during which it detects which of the maximum 31 possible stations are connected to the AS-i cable and what type these slaves are. The type of station is stipulated by a configuration byte permanently stored on the slave that can be scanned by the master. This byte identifies the I/O assignment of a slave and the slave type (ID code).

  The master enters detected slaves in the list of detected slaves (LDS).

- **Activation phase: activating slaves**

  After detecting the stations, they are activated by the master sending a special call. When activating individual stations, a distinction is made between two modes on the AS-interface master:

  – Master in the configuration mode:

    All detected slaves (with the exception of the slave with address “0”) are activated. In this mode, it is possible to read actual values and to store them for a configuration.

  – Master in the protected mode:

    Only the stations corresponding to the desired configuration stored on the C7-AS-i CP are activated. If the actual configuration found on the AS-i cable differs from this desired configuration, this is indicated by the C7-AS-i CP.

  The master enters activated slaves in the list of activated slaves (LAS).

Normal Mode

On completion of the startup phase, the C7 AS-i CP switches to the normal mode.

- Data exchange phase

  In the normal mode, the master sends cyclic data (output data) to the individual stations and receives their acknowledgment messages (input data). If an error is detected during the transmission, the master repeats the appropriate poll. All the slaves connected to the AS-i cable are polled within 5 ms.

- Management phase

  During this phase, all existing jobs of the control application are processed and sent. Possible jobs are, for example, as follows:

  – Parameter transmission. Four parameter bits are transferred to a station, that can, for example, be used to set a threshold value.

  – Modification of slave addresses. This function allows the addresses of slaves to be changed by the master if the slave supports this particular function.

- Inclusion phase

  During the inclusion phase, new slaves added in the meantime are included in the list of detected slaves and providing the configuration mode is selected also entered in the list of active slaves (except for slaves with address 0). If the master is in the protected mode, only slaves in the desired configuration stored on the C7 AS-i CP are activated. Slaves which were temporarily out of service are also included again.

AS-i Application and Functions 6

Chapter Overview

Note

The AS-i attachment is implemented with the C7-621 AS-i with an integrated AS-i master module. In this chapter, the C7 AS-i CP means the AS-i master module in the C7-621 AS-i control system.

6.1 AS-i Application and Functions

Overview

This chapter explains the following:

- The functions of the C7 AS-i CP.
- The uses of the AS-interface.
- How to address and assign parameters to the AS-interface.
- How to replace a slave in an AS-i chain.
- How to operate AS-i and display messages from the AS-i I/Os.

Operating AS-i

With AS-i, there are two types of operation possible:

- **Standard operation**. In this type of operation, the AS-i behaves like a conventional I/O module. Four input and four output bits are reserved in the C7 CPU for each slave connected to the AS-i cable.

  In the standard mode, no commands or special parameters can be transferred to the slaves on the AS-i cable.
- **Extended operation (with FC "AS-i_3422")**: Here, the PLC programmer has the full range functions of the AS-i system available.

  The FC "AS-i_3422" is required for extended operation.

6.2 Standard Operation

Overview

Standard operation represents the most common and at the same time simplest use of the C7 AS-i CP. It allows direct access to the inputs and outputs of the slave (e.g. bus modules) just like analog I/O modules of the SIMATIC PLC. This type of operation is available immediately after turning on the C7-621 AS-i; no function (FC) is required.

6.2.1 Configuring the C7 AS-i CP for Standard Operation

The C7 AS-i CP is capable of 2 operating modes:

- **Configuration Mode**:
  The configuration mode is used to install and start up an AS-i network. In the configuration mode, the C7 AS-i CP can exchange data with every slave connected to the AS-i cable. Newly added slaves are detected by the master immediately and included in cyclic data exchange. After testing the PLC program, the slaves can be entered as the desired configuration (see Section 6.3.1 **set conf**) and the C7 AS-i CP changed to the protected mode using a control on the C7-621 (see Section 6.3.1 Config Mode from Enabled –> Disabled).
- **Protected Mode:**
  If the C7 AS-i CP is in the protected mode, it only exchanges data with slaves that are “configured”. In this sense, “configured” means that the slave address saved on the CP and the configuration data stored on the CP match the values of a slave.

Setting this mode and its displays are described in Section 6.3.

The configuration mode is only possible when the C7 CPU is in the STOP mode.

Configuration of the C7-621 AS-i CP during Installation and Startup of the AS-i Network

The following situation is assumed:

- The connected AS-i slaves are supplied with addresses. Use the addressing functions of the C7-621 AS-i (see Section 6.3.2)
- The AS-i bus is complete, in other words, all the slaves are connected via the AS-i cable.

To configure the C7 AS-i CP in standard operation while installing the AS-i network, follow the steps outlined below:

1. Switch the C7 CPU to STOP.
2. Change the C7 AS-i CP to the configuration mode. If the CP is already in the configuration mode (as shipped), this step can be omitted.
3. Switch the C7 to RUN and test your program.

   Note:
   In the configuration mode, you can also add or remove slaves from the AS-i cable. Newly added slaves are activated immediately by the C7 AS-i CP.
4. On completion of the installation of the AS-i slaves, switch the C7 CPU to the STOP mode.
5. Now change the C7 AS-i CP to the protected mode (see Section 6.3.3). The C7 AS-i CP adopts the configuration displayed as the active slaves as the desired configuration and switches to the protected mode.
6. Switch the C7 to RUN. The installation of the C7 AS-i CP is then completed.

Simplified Configuration

Once you are certain that all the slaves on the AS-i cable are functioning correctly (e.g. when a C7 AS-i CP is replaced), the C7 AS-i CP can be started up as follows:

1. Switch the C7 CPU to the STOP mode.
2. Change the C7 AS-i CP to the configuration mode (see Section 6.3.1). If the AS-i CP is already in the configuration mode (as shipped), this step can be omitted.
3. The CP then adopts the actual configuration as the desired configuration and switches to the protected mode.
4. Switch the C7 CPU to RUN. The AS-i CP is then started up.

6.2.2 Addressing the C7 AS-i CP with a Program

Overview

The C7 AS-i CP occupies 16 bytes in the address area (analog area) of the C7 CPU:

- 16 input bytes
- 16 output bytes

Address Area

Of the 16 byte address area of the C7 AS-i CP, 31 x 4 bits are used for the AS-interface slave data. The remaining 4 bits are reserved for later applications.

The start address of this area is set to 256 on the C7-621 AS-i.

How the C7 CPU Addresses the Slaves

Each station (slave) on the AS-i cable is assigned 4 bits (a nibble) by the C7 AS-i CP. The C7 CPU can write (slave output data) and read (slave input data) this nibble. This allows bidirectional slaves to be addressed.

Note

The first four input bits (first nibble) are reserved for the use of FC “AS-i_3422”. If you do not use the FC, the first four input bits change between the values 8_H and E_H approximately every 2.5 seconds. The first four output bits have no significance for the C7 AS-i CP.

Table 6-1 illustrates the assignment of the AS-interface. The table shows the assignment of the slave I/O bits to the I/O bytes of the C7 CPU.

Table 6-1 Assignment of the C7 AS-i CP Interface

I/O Byte Number	Bit 7-4	Bit 3-0
256	reserved for FC AS-i_3422	Slave 1 Bit 3 | Bit 2 | Bit 1 | Bit 0
257	Slave 2	Slave 3
258	Slave 4	Slave 5
259	Slave 6	Slave 7
260	Slave 8	Slave 9
261	Slave10	Slave 11
262	Slave 12	Slave 13
263	Slave 14	Slave 15
264	Slave 16	Slave 17
265	Slave 18	Slave 19

Table 6-1 Assignment of the C7 AS-i CP Interface

I/O Byte Number	Bit 7-4	Bit 3-0
266	Slave 20	Slave 21
267	Slave 22	Slave 23
268	Slave 24	Slave 25
269	Slave 26	Slave 27
270	Slave 28	Slave 29
271	Slave 30	Slave 31
	Bit 3 | Bit 2 | Bit 1 | Bit 0	Bit 3 | Bit 2 | Bit 1 | Bit 0

Example of the Assignment of Connections

If AS-i modules are used as slaves on the AS-i cable, each of the connections to the AS-i module corresponds to exactly 1 bit in the C7 CPU. The following example illustrates the assignment of two AS-i modules with addresses 2 and 3.

Table 6-2 Assignment of the Connections on the AS-i Module

	Slave 2				Slave 3			
C7 CPU I/O bits	7	6	5	4	3	2	1	0
Slave bits Connection to AS-interface module.	4	3	2	1	4	3	2	1

Explanation:

- Slave 2 corresponds, in this case, to the AS-i module with address 2.
- Slave 3 corresponds to the AS-i module with address 3.

Example of Addressing a Slave

The inputs and outputs of the AS-i slaves can be accessed just like the analog I/Os of the C7. The following example illustrates this procedure:

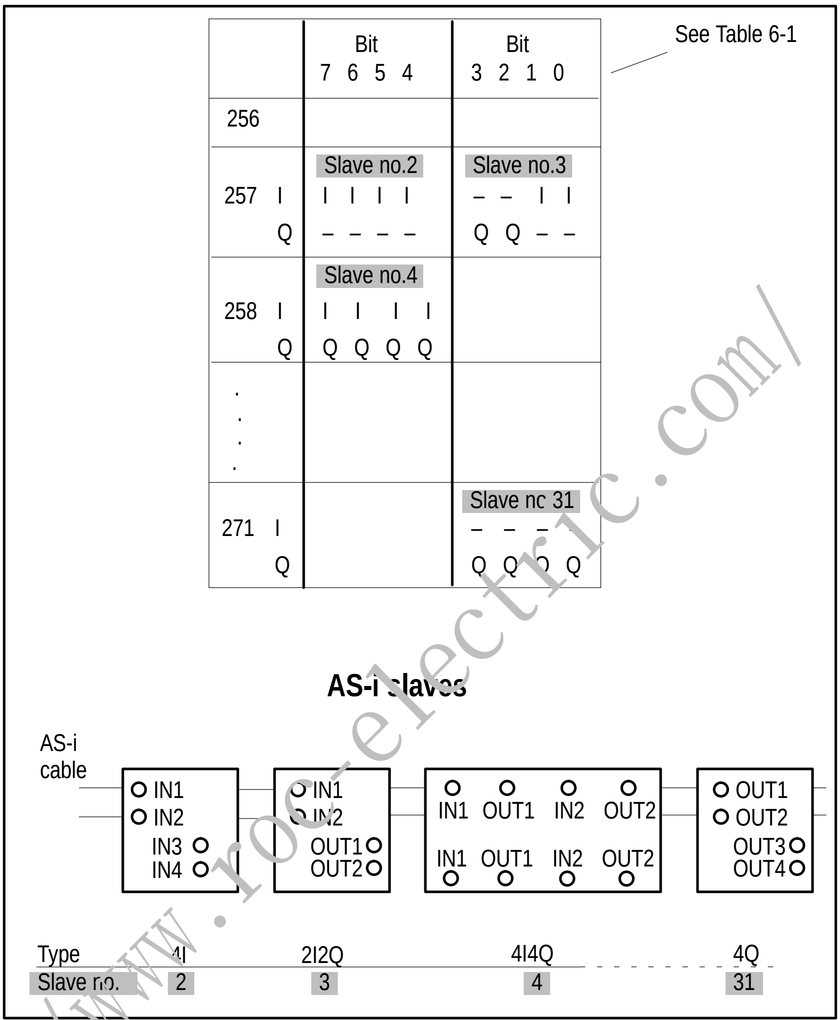


Figure 6-1 Mapping of the Slave Addresses on the I/O Addresses of the C7-621 AS-i

Access to the AS-i User Data

The bits of the AS-i slaves are accessed using the following S7 load and transfer commands:

L PIW X

L PID X

T PQW X

T PQD X

X stands for the byte address on the C7 AS-i CP.

Note

The system allows only word-oriented or double word-oriented access to even byte addresses. The byte transfer commands L PIB X or T PQB X are not permitted with the C7 AS-i CP.

Example:

Correct:	`L PIW 260`
Wrong:	`T PQB 260`
Wrong:	`L PIW 257`

Programming Example

If you want to access bits of the individual slave input and output data, you can use a program similar to that shown below for an AS-i-CP with start address 256:

```
OPN DB 20      //Open a data block
//
//             1.) At start of program:  – Read in “pseudo PII”
//                                          (copy input data of the
//                                          C7 AS-i CP to a
//                                          data block)
L PID 256
T DBD 0
L PID 260
T DBD 4
L PID 264
T DBD 8
L PID 268
T DBD 12
  .
  .
//
//             2.) In the program:     –Evaluate single “input bits”
//                                    – Set/reset single
//                                      “output bits”
A DBX 5.4
S DBX 22.3
R DBX 28.0
  .
```

```
  .
  .
//               3.) At end of program:  –   Output "pseudo PIQ"
//                                           (Copy data words to the
//                                           output data of the
//                                           C7 AS-i CPL DBD 16)
T PQD 256
L DBD 20
T PQD 260
L DBD 24
T PQD 264
L DBD 28
T PQD 268
```

6.3 Control and Display Elements of the C7-621 AS-i

Overview

The C7-621 AS-i allows you to make settings for the connected slaves and to display certain statuses of the slaves. The following control and display elements are available:

Controls

- Setting the addresses of the slaves Section 6.3.2
- Setting the desired configuration from the current actual configuration Section 6.3.5
- Setting the Configuration mode Section 6.3.3
- Setting the Protected mode Section 6.3.3
- Setting the AUTOPROG mode Section 6.3.3.

Displays

- The firmware version of the C7 AS-i CP
- The current status of the connected slaves Section 6.3.6
- The status and error messages of the master Section 6.3.7
- A warning that the destination address might be overwritten by the AUTOPROG mode.

Selecting an AS-i Function

You can only select the individual AS-i functions after you have selected the system function AS-i. You select the AS-i functions as follows:

1. The basic menu must be displayed.

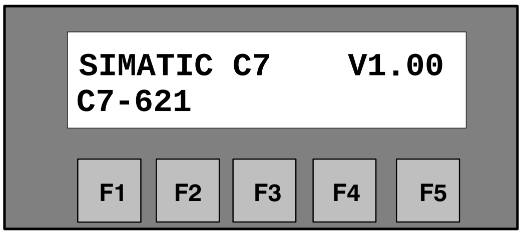


Figure 6-2 Basic Menu

2. Select the system functions menu by pressing SHIFT 0 at the same time

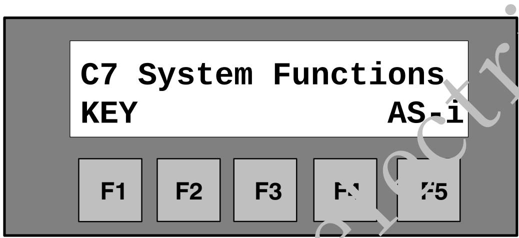


Figure 6-3 System Functions Menu with Function Keys

3. Select the initial menu for the AS-i functions by pressing F5

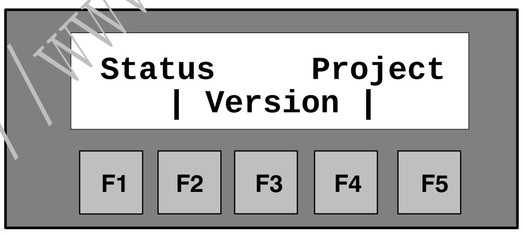


Figure 6-4 Initial Menu

Quitting the Selected Menu

You quit a selected menu by pressing the ESC key. The previous menu is then displayed. If a different operation is necessary, a message is displayed to indicate the required procedure.

Messages

The following messages can be displayed:

- No error occurred

```
AS–I DS Write state
NO_ERROR   confirm
```

- Error(s) occurred

```
AS–I DS Write state
ERROR      confirm
```

- If a slave with address 0 exists and you attempt to change from the configuration mode to the protected mode, the following message is displayed

```
AS–I DS Write state
NOT ALLOWED  confirm
```

- A slave with address 0 already exists

```
AS–I DS Write state
SD0_ERROR   confirm
```

- A slave with the NEW address already exists

```
AS–I DS Write state
SD2_ERROR   confirm
```

Messages must be acknowledged with the **F5** key (confirm).

Displaying the Firmware Version of the C7 AS-i CP

You can display the firmware version of the C7 AS-i CP as follows:

1. Display the initial menu
2. Select Version with **F3**

   The version of the firmware is displayed

```
AS-i Master Version
****** Vx.xx ******
```

AS-i Menu Structure

Figure 6-5 shows an overview of the most important menus and how they depend on each other. How to call the menus is explained in the following sections.

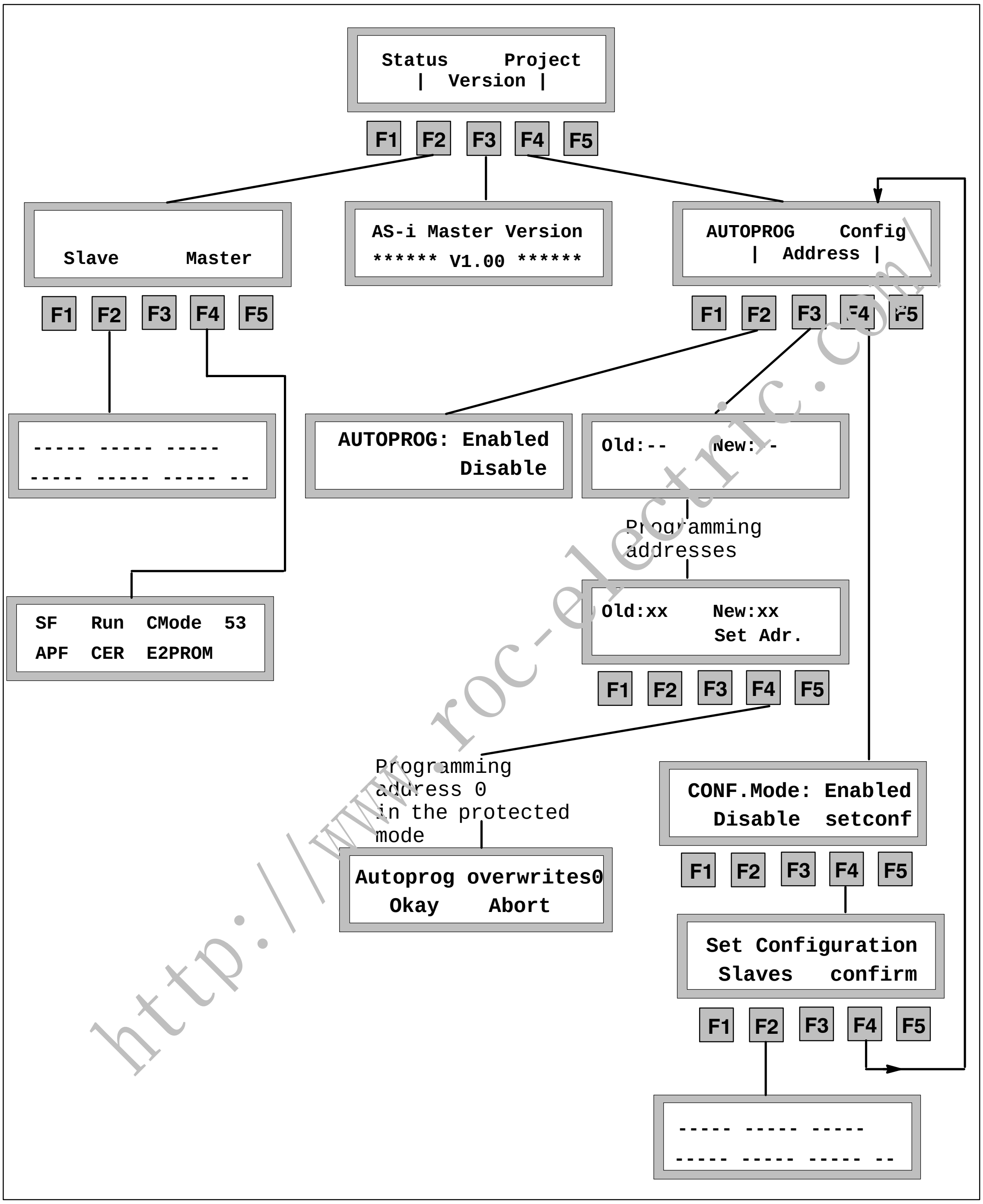


Figure 6-5 Menu Structure

6.3.1 Setting the Configuration or Protected Mode

Overview

Before you can set addresses or adopt the actual configuration, certain modes must be set.

- Turn off the configuration mode ≙ protected mode.
- Turn on the configuration mode.

How to do this is explained below.

Condition

Before you can set the **configuration mode**:

- The C7 must be in the STOP mode (see Section).

  If the RUN mode is set, the following message is displayed

  ```
  Wrong PLC Mode
  Please check
  ```

  and the configuration mode is not set.
- The initial menu must be selected (see page 6-11)
- The highest password level must be set.

Procedure with CONFIG

If the configuration mode is enabled, the protected mode is disabled and vice versa. You can toggle the configuration mode between Enabled and Disabled. To activate or deactivate the configuration mode or protected mode, follow the steps below:

Configuration Mode from Enabled –> Disabled ≙ Protected Mode

Step	Activity	Result
1.	Select the menu for **Project** with **F4**	**AUTOPROG Config** **| Address |**
2.	Select the menu for **Config** with **F4**	Display when enabled **Conf.Mode: Enabled** **Disabled setconf**
3.	**Select Enable –> Disabled** Select **Disabled** with **F4**	Menu displayed **Conf.Mode: Disabled** **Enable** Input is accepted when enabled is set. Messages are displayed (see page 6-12)

Configuration Mode from Disabled –> Enable

Step	Activity	Result
1.	Select the menu for **Project** with F4	AUTOPROG Config | Address |
2.	Select the menu for **Config** with F4	Display when Disabled. Conf.Mode: Disabled Enable
3.	**Select Disabled –> Enable** Select **Enabled** with F2	Menu displayed Conf.Mode: Disabled Enable Input is accepted when Disabled is set. Messages are displayed (see page 6-12) Display indicates that you selected Enabled. Conf.Mode: Enabled Disabled setconf

6.3.2 Setting the Address of a Slave

Overview

You can modify the current address of a slave using the C7. How to do this is described below.

Condition

Before you can enter the address for a slave:

- The C7 must be in the STOP mode.

  If the C7 is in the RUN mode, the following message is displayed

  ```
  Wrong SPS-Mode
  Please check
  ```

  and your input is rejected.
- The initial menu must be selected (see page 6-11)
- The highest password level must be set.

Procedure

To enter the address of a slave, follow the steps below:

Step	Activity	Result
1.	Select the menu for **Project** with F4	`AUTOPROG      Config` `  |  Address |`
2.	Select the menu for **Address** with F3	`Old:■-  New:--`

3.	Enter the old and new address with the keys 0 to 9.
	
	Move the cursor with ◁ ▷
		Once you have entered the old and new addresses `Set Adr` is displayed.
		`Old:--  New:-■`
		`     Set Adr.`

4.1 4.2	Confirm the address change with F4 Confirm the new address with F5	The required address is adopted. A message as described on page 6-12 is displayed. 1. **No error**–> you can enter a further address (Step 3). 2. **Error** –>repeat address input (Step 3).
5.	Quit with ESC	The previous menu is displayed (Menu step 1).

6.3.3 Setting the AUTOPROG Mode

Overview

The AUTOPROG mode is used to set the address of **one** slave automatically (for example when a defective slave is replaced). You can address any number of slaves automatically one after the other.

How to use the AUTOPROG function is described below.

Condition

To set the **AUTOPROG** mode:

- The C7 must be in the STOP mode.

  If the RUN mode is set, the following message is displayed

  ```
  Wrong SPS-Mode
  Please check
  ```

  and the call is rejected.
- A desired configuration must exist
- The initial menu must be selected (see page 6-11)
- The highest password level must be set.
- AUTOPROG operates only in the protected mode (Config.Mode Disable).

Procedure for AUTOPROG

To start or stop the AUTOPROG mode, follow the steps below:

Step	Activity	Result
1.	Select the menu for **Project** with **F4**	`AUTOPROG Config` `| Address |`

2.	Select the menu for **AUTOPROG** with **F2**	**Either**
		 `AUTOPROG: Enabled`
		 `Disable`
		 or
		 `AUTOPROG: Disabled`
		 `Enable`

Step	Activity	Result

3.	Activate **AUTOPROG : Enabled**
	
	Select **Enabled** with F2
	
	No error: acknowledge with F5
	
	Error: acknowledge with F5
	
	Quit with ESC
		Menu displayed
		`AUTOPROG: Disabled`
		`Enable`
		The input is accepted if Disabled is set.
		Messages are displayed (see page 6-12)
		The display indicates that you selected Enabled.
		Input can be repeated.
		Menu as in Step 3 is displayed again.
		The input menu (Step 1) is displayed again.

3.1	Deactivate **AUTOPROG : Disabled**
	
	Select **Disabled** with F4
	
	Further steps as for Enabled
		Menu displayed
		`AUTOPROG: Enabled`
		`Disable`
		The input is accepted if Enabled is set.
		Messages are displayed (see page 6-12)

6.3.4 Replacing a Defective Slave and Automatic Address Programming (AUTOPROG)

Overview

If SF is set the status display in the AS-i menu of the master and CER in the protected mode of the C7 AS-i CP, this means the following:

- Exactly **one** slave has failed.
- Automatic address programming (AUTOPROG) by the C7 AS-i CP is possible.

Replacing a Defective Slave

You can now replace the defective slave as follows:

1. Remove the failed slave from the AS-i cable.

You now have two options:

2. If you have an identical slave with address 0 (as shipped) simply replace the slave.

   The C7 AS-i CP then programs this slave with the address of the original station that had failed.

or

3. If you have an identical slave with an address other than 0 (the address must not already be in use in your AS-i configuration), you can reprogram this to address 0 (New)

```
Old:xx  New:00
     Set Adr.
```

When you confirm the new address with **F4** the following message is displayed:

```
Autoprog overwrites0
Okay          Abort
```

OKAY: The C7 AS-i CP accepts address 0 and now programs the slave automatically to the address of the original failed node.

The following message is displayed:

```
AS–I DS Write state
NO_ERROR   confirm
```

Acknowledge with **F5**

Abort: The reprogramming to address 0 followed by AUTOPROG is abandoned. You can enter a new address.

The SF and CER display is cleared (see Section 6.3.7). The C7 AS-i CP displays the new slave in the status display of the slaves.

Note

Note that “automatic address programming” is only possible when the C7 AS-i CP is in the protected mode and when only one slave has failed.

6.3.5 Setting the Desired Configuration from the Actual Configuration

Overview

You can accept the detected actual configuration of the slaves as the desired configuration.

Condition

To set the desired configuration:

- The C7 must be in the STOP mode.

  If RUN is set, the following message is displayed

  ```
  Wrong SPS-Mode
  Please check
  ```

  and the call is rejected.
- The initial menu must be selected.
- The highest password level (9) must be set.
- The configuration mode must be activated (Enabled).

Procedure

To set a desired configuration, follow the steps below:

Step	Activity	Result
1.	Select the menu for **Project** with **F4**	`AUTOPROG Config` `| Address |`

2.	Select the menu for **CONFIG** with **F4**
	
	 In this case activate `Enabled` with **F2** **F5**
	
	 (see Section 6.3.3)
		If the C7 AS-i CP **is** in the configuration mode and Enabled, the display indicates that the desired configuration can be set ①
		 `Conf.Mode: Enabled`
		 `Disable setconf` ①
		
		 If the C7 AS-i CP is **not** in the configuration mode, the following menu is displayed.
		 `Conf.Mode: Disabled`
		 `Enable`
		 You cannot adopt an actual configuration.
		 `Conf.Mode: Enabled`
		 `Disable setconf`

Step	Activity	Result
3.	If enabled, select the menu to set the desired configuration F4	`Set Configuration` `Slaves   confirm`

4.	If you wish, select the display of the current actual configuration (`Slaves`) with F2	`Example of display`
		`----- ----- -----`
		`----- ----- ----- --`

	Quit display with ESC	The previous menu (Step 3) is displayed again.
5.	Set the desired configuration with F4	The actual configuration is set as the desired configuration. Messages are displayed (see page 6-12).
	No error: acknowledge with F5	Menü as in Step 1 is displayed again
	Error: acknowledge with F5	You can repeat Step 3
	Quit with ESC	The initial menu is displayed again (Step 1)

6.3.6 Status of the Slaves

Overview

Using the `Status` menu of the C7-621 AS-i, you can display the existing and activated slaves.

Condition

To display the status:

- The initial menu must be selected.

Procedure

To display the status, follow the steps below:

Step	Activity	Result
1.	Select the menu for **Status** with F2	`Slaves    Master`

2.	Select the **Status Display** for `slaves` with F2	Address 0
		`----- ----- -*---`
		`----- ----- ----- --`
		Address 31
		For an explanation of the display see Table 6-3

3.	Quit the display with ESC	The menu (Step 1) is displayed again.

Explanation of the Status Display

The display is graphical. Table 6-3 shows the meaning of the displayed characters.

Table 6-3 Explanation of the Status Display

Character Displayed	Explanation	Slave in List of Active Slaves	Slave in Delta List
Underscore	Slave does not exist	no	no
Solid rectangle	Slave exists, OK	yes	no
Asterisk (*)	• Slave missing, or • Slave too many, or • Slave has wrong configuration.	no	yes

How the Slave Display is Counted

The slaves are displayed starting from address 0 at the top left. The slave with the highest address is shown at the bottom right.

6.3.7 Status and Error Messages of the Master

Overview

A menu on the C7 621 AS-i allows you to display the status of the master.

Condition

To display the status of the master:

- The initial menu must be selected.

Procedure

To display the status, follow the steps below:

Step	Activity	Result
1.	Select the menu for **Status** with F2	`Slaves    Master`

2.	Select the **Status Display** for `Master` with F4	`SF  Run  CMODE`
		`APF  CER  E2PROM`
		For an explanation of the display, see Table 6-4

3.	Quit the display with ESC	The menu (Step 1) is displayed again.

Explanation of the Display

The status and error messages are explained in the following table.

Table 6-4

Text Display	Status
SF	Group error
APF	AS-i Power Fail
CER	Config Error
RUN or STOP	Run or STOP (on the C7 CPU)
CMODE or PMODE	ConfigMode (enable) Protected Mode (ConfigMode disable)
E2PROM	E2PROM FAIL

6.4 Error Indicators on the C7 AS-i CP / How to Remedy Errors

Overview

This section lists possible causes of problems during operation of the C7 AS-i CP and explains how to deal with the problem.

You can display the error messages in the status display for the master (see Section 6.3.7).

Error	Possible Cause	Remedy
APF	The AS-i power supply unit is not connected or is defective.	Check the connection of the AS-i power supply unit and if necessary replace it.
	AS-i slaves require too much current.	Check the current requirements of the AS-i slaves. If necessary, arrange for a separate power supply for the slaves.
SF	The C7 AS-i CP is in the protected mode, and there is an AS-i configuration error (for example slave failure).	Eliminate the configuration error.
	The C7 AS-i CP is defective. Internal EEPROM error -> see signaling S7-300 system diagnostic buffer.	Replace the C7-621 CP.
CER	The C7 AS-i CP is not yet configured.	Configure the C7 AS-i CP with the OP user interface as explained in Section 6.3
	A configured slave has failed (Evaluate the "ACTIVE SLAVES" display).	Replace the defective slave or reconfigure the C7 AS-i CP, if the slave is not required.
	A non-configured slave was connected to the AS-i cable.	Remove the slave or reconfigure the C7 AS-i CP.
	A slave was connected whose configuration data (I/O configuration, ID code) do not match the values of the configured slave.	Check whether the wrong slave was connected. If necessary, reconfigure the C7 AS-i CP.
CER display changes sporadically	Bad contact	Check the connections of the AS-i slaves.
	Noise on the AS-i cable.	Check the grounding of the PLC and the AS-i cable. **Check whether the shield of the AS-i power supply unit is correctly connected**.
C7 AS-i CP does not change from the configuration mode to the protected mode.	The C7 is in the "RUN" mode.	Switch the C7 to "STOP".
	A slave with address 0 is connected to the AS-i cable. The C7 AS-i CP cannot change to the protected mode as long as this slave exists.	Remove the slave with address 0.
C7 AS-i CP does not change from the protected mode to the configuration mode.	The C7 is in the "RUN" mode	Switch the C7 to "STOP".

No automatic address programming although AUTOPROG is set.	The configuration data (I/O configuration, ID code) of the replaced slave do not match the values of the original slave.	Check whether the correct replacement slave was used. Compare the vendor information about the configuration data. If the original slave is being replaced with another type, assign the address with the appropriate menu (see Section 6.3.2) and reconfigure the C7 AS-i CP.
	Replaced slave does not have the address “zero”.	Set the address of the replaced slave with the appropriate menu (see Section 6.3.2) and reconfigure the C7 AS-i CP.
	Replaced slave is not correctly connected or defective.	Check the connections of the slave; if necessary replace the slave.

6.4.1 Diagnostics and Response to Interrupts of the C7 AS-i CP

Overview

If the C7 AS-i CP recognizes an external or internal error during operation (AS-i slave failure, EEPROM error on the CP, ...), it signals this using a diagnostic interrupt (diagnostic interrupt) to the S7 I/O bus.

Reaction to Errors

The C7 CPU then interrupts the cyclic user program (OB1), enters the event as a “module fault” message in the system diagnostic buffer and then reacts as follows:

- If the user has programmed OB82, this is started by the C7 CPU operating system. The local data of OB82 already contain some diagnostic information for the user (which module triggered the interrupt? what type of error has occurred? ...).

  You can obtain more detailed information (which slave has failed? ...) in the user program by reading the diagnostic data record DR 1 using SFC59 (“RD_REC”). When OB82 has been executed, the C7 CPU continues the cyclic program (OB1) from the point at which it was interrupted.

Note

If you do not program an interrupt OB (OB82), the C7 CPU changes to STOP!

Interrupt Events

The C7 AS-i CP distinguishes between events entering and leaving the state. If an interrupt event occurs that leads to an error-free status (AS-i-CONFIG_OK=TRUE and there is no internal CP error), a diagnostic interrupt leaving state is triggered (in OB82: bit OB82_MDL_DEFECT = 0). All other interrupt events lead to a diagnostic interrupt entering state (bit OB82_MDL_DEFECT = 1).

The events that can lead to the C7 AS-i CP triggering a diagnostic interrupt are listed below:

Interrupt events external to AS-i:

- All changes to the AS-i slave configuration in the protected mode
- AS-i power fail in the protected mode

Internal AS-i interrupt events:

- EEPROM error

Note

AS-i internal interrupt events are always events entering the state. After an AS-i internal interrupt event, the group error bit remains set to TRUE. The bit is only reset after a complete restart on the C7.

Servicing Diagnostic Interrupts

If the C7 AS-i CP detects an interrupt event, it triggers a diagnostic interrupt. The C7 CPU checks the source of the interrupt and reads data record 0 from the CP. It then interrupts the cyclic user program and reacts as follows:

- If OB82 is not programmed, the C7 CPU changes to STOP.
- If the organization block OB 82 is programmed, it is started. Local data bytes 8 to 11 of the local data area of OB82 contain the information from data record 0. Data record 1 containing the so-called delta list can (but does not need to be) read in OB82 using an SFC call (SFC59 “RD_REC”).
- When it has been executed, OB82 acknowledges the diagnostic interrupt to the C7 CPU in the C7 AS-i CP.

If interrupt events occur in a status in which they cannot be signaled by triggering a diagnostic interrupt (for example when the PLC is in the STOP mode or when an old diagnostic interrupt has not yet been acknowledged), the C7 AS-i CP reacts as follows:

- When it becomes possible again to trigger a diagnostic interrupt and when at this time the current total AS-i configuration (in other words, the AS-i slave configuration and AS-i internal status relevant to the interrupt) is not the same as the configuration signaled earlier with the diagnostic interrupt, a diagnostic interrupt is triggered with the current configuration information.
- When it becomes possible again to trigger a diagnostic interrupt and when at this time the current total AS-i configuration is the same as the configuration signaled earlier with the diagnostic interrupt, no diagnostic interrupt is triggered. This means that brief slave failures (for example a bad contact) may not be signaled.

Reaction to Interrupts in Different AS-i Modes

The C7 AS-i CP generates diagnostic interrupts triggered by external interrupt events only in the protected mode and not in the configuration mode.

Note

When the C7 CPU changes to STOP, the external and internal interrupt history is reset, this means that the bit OB82_MDL_DEFECT and all other error bits in data record 0 are reset.

During a change from the protected mode to the configuration mode, the external interrupt history is reset.

A change from the configuration mode to the protected mode when there is no configuration error is signaled by a diagnostic interrupt.

If it is temporarily not possible to output a diagnostic interrupt (for example because the C7 is in the STOP mode), a diagnostic interrupt is only generated at the next possible opportunity if the error still exists.

Local Data of the Diagnostic Organization Block (OB82)

The following table lists the date record DR 0 available in the local data of OB82 (local byte 8 to local byte 11). The meaning of the remaining local data of OB82 is explained in the descriptions of STEP 7.

Table 6-5 Data Record 0 in Local Data OB82 (from Byte 8 –> Byte 11)

Byte	Bit	Variable Name	Data Type	Meaning
8	2^0	OB82_MDL_DEFECT	BOOL	Group error bit (0: interrupt leaving state, 1: interrupt entering state)
8	2^1	OB82_INT_FAULT	BOOL	Internal C7 AS-i CP error/fault (e.g. EEPROM defective)
8	2^2	OB82_EXT_FAULT	BOOL	External C7 AS-i CP error/fault (e.g. slave failed or APF)
8	2^3	OB82_PNT_INFO	BOOL	At least one slave different from set configuration
8	2^4	OB82_EXT_VOLTAGE	BOOL	Voltage too low on AS-interface (APF)
8	2^5	OB82_FLD_CONNCTR	BOOL	For the C7 AS-i CP always 0
8	2^6	OB82_NO_CONFIG	BOOL	For the C7 AS-i CP always 0
8	2^7	OB82_CONFIG_ERR	BOOL	For the C7 AS-i CP always 0
9		OB82_MDL_TYPE	BOOL	Module class (for C7 AS-i CP: 1C H)
10	2^0	OB82_SUB_NDL_ERR	BOOL	At least one slave different from set configuration
10	2^1	OB82_COMM_FAULT	BOOL	For the C7 AS-i CP always 0
10	2^2	OB82_MDL_STOP	BOOL	0: C7 AS-i CP in normal status 1: C7 AS-i CP in offline status
10	2^3	OB82_WTCH_DOG_FLT	BOOL	Hardware fault on the CPs (internal watchdog)
10	2^4	OB82_INT_PS_FLT	BOOL	For the C7 AS-i CP always 0
10	2^5	OB82_PRIM_BATT_FLT	BOOL	For the C7 AS-i CP always 0
10	2^6	OB82_BCKUP_BATT_FLT	BOOL	For the C7 AS-i CP always 0
10	2^7	OB82_RESERVED_2	BOOL	For the C7 AS-i CP always 0
11	2^0	OB82_RACK_FLT	BOOL	For the C7 AS-i CP always 0
11	2^1	OB82_PROC_FLT	BOOL	For the C7 AS-i CP always 0
11	2^2	OB82_EPROM_FLT	BOOL	EEPROM of the C7 AS-i CP defective
11	2^3	OB82_RAM_FLT	BOOL	For the C7 AS-i CP always 0
11	2^4	OB82_ADU_FLT	BOOL	For the C7 AS-i CP always 0
11	2^5	OB82_FUSE_FLT	BOOL	For the C7 AS-i CP always 0
11	2^6	OB82_HW_INTR_FLT	BOOL	For the C7 AS-i CP always 0
11	2^7	OB82_RESERVED_3	BOOL	For the C7 AS-i CP always 0

Note

The four bytes described above are entered in the system diagnostic buffer by the C7 CPU operating system when an interrupt occurs.

Reading the Diagnostic Data Record 1

The C7 AS-i CP continuously updates a so-called delta list that contains all existing slaves that differ from the configuration, in other words, missing, incorrect or unconfigured slaves. (Each slave is assigned a bit in the delta list: 0 = no error; 1 = error).

This delta list is part of the diagnostic data record 1 that you can read out by calling SFC59 (“RD_REC”) both in the interrupt OB (OB82) and at any time in the cyclic program (OB1).

With the C7 AS-i CP, data record 1 always has a length of 11 bytes and is structured as follows:

Table 6-6 Structure of Data Record 1

Byte	Explanation
0 to 3	These four bytes contain data record 0 and correspond to the local data bytes bytes 8 to 11 in OB82 (see Section 3.4.5)
4 to 6	Fixed value: 60 00 20_H
7 to 10	Delta list Bit 2^0 in byte 7 corresponds to slave 0 Bit 2^7 in byte 10 corresponds to slave 31

Following the sample program, you will find an example showing how to evaluate the delta list.

Sample Program

The following sample program is an example of how you can use OB82 to read data record 1 and react to a diagnostic interrupt from the C7 AS-i CP.

It is assumed that slaves 1 and 12 were configured on the C7 AS-i CP using the AS-i menus of the C7-621 AS-i and that the C7 AS-i CP is in the protected mode (setting the mode, see Section 6.3.3).

- If slave 7 fails, the C7 AS-i CP generates a diagnostic interrupt. The C7 CPU operating system then enters the message “module fault” in the system diagnostic buffer and starts OB82 (program shown below). On completion of OB82, the delta list contains the following information:

  MB 107 80_H
  MB 108 00_H
  MB 109 00_H
  MB 110 00_H

- If the unconfigured slave 15 is then connected to the AS-interface, the C7 AS-i CP once again generates a diagnostic interrupt. The system diagnostic buffer once again has the message “module fault”. The delta list changes to the following value:

  MB 107 80_H
  MB 108 80_H
  MB 109 00_H
  MB 110 00_H

- After connecting slave 7 to the AS-interface again, there is still a problem (slave 15). The system diagnostic buffer contains the message “module fault” and the delta list has the following value:

  MB 107 00_H
  MB 108 80_H
  MB 109 00_H
  MB 110 00_H

- After disconnecting slave 15, there is no further error. The C7 AS-i CP signals this in a diagnostic interrupt. The message “module OK” appears in the system diagnostic diagnostic buffer and the delta list is empty:

  MB 107 00_H
  MB 108 00_H
  MB 109 00_H
  MB 110 00_H

```
ORGANIZATION_BLOCK "OB82"
TITLE =
VERSION : 0.0
VAR_TEMP
 OB82_EV_CLASS : BYTE                //16#39, Event class 3,entering event state
 OB82_FLT_ID : Byte;                 //16#xx fault identification code
 OB82_PRIORITY : BYTE ;              //26/28 (priority 1 is lowest)
 OB82_OB_NUMBR : BYTE ;              //82 (organization block 82, OB82)
 OB82_RESERVED_1 : BYTE ;            //Reserved for system
 OB82_IO_FLAG : BYTE ;               //Input (01010100), output (01010101)
 OB82_MDL_ADDR : INT ;               //Base address of module with fault
 OB82_MDL_DEFECT : BOOL ;            //Module defective
 OB82_INT_FAULT : BOOL ;             //Internal fault
 OB82_EXT_FAULT : BOOL ;             //External fault
 OB82_PNT_INFO : BOOL ;              //Point information
 OB82_EXT_VOLTAGE : BOOL ;           //External voltage low
 OB82_FLD_CONNCTR : BOOL ;           //Field wiring connector missing
 OB82_NO_CONFIG : BOOL ;             //Module has no configuration data
 OB82_CONFIG_ERR : BOOL ;            //Module has configuration error
 OB82_MDL_TYPE : BYTE ;              //Type of module
 OB82_SUB_NDL_ERR : BOOL ;           //Sub-Module is missing or has error
 OB82_COMM_FAULT : BOOL ;            //Communication fault
 OB82_MDL_STOP : BOOL ;              //Module is stopped
 OB82_WTCH_DOG_FLT : BOOL ;          //Watchdog timer stopped module
 OB82_INT_PS_FLT : BOOL ;            //Internal power supply fault
 OB82_PRIM_BATT_FLT : BOOL ;         //Primary battery fault
 OB82_BCKUP_BATT_FLT : BOOL ;        //Backup battery fault
 OB82_RESERVED_2 : BOOL ;            //Reserved for system
 OB82_RACK_FLT : BOOL ;              //Rack fault, only for bus interface module
 OB82_PROC_FLT : BOOL ;              //Processor fault
 OB82_EPROM_FLT : BOOL ;             //EPROM fault
 OB82_RAM_FLT : BOOL ;               //RAM fault
 OB82_ADU_FLT : BOOL ;               //ADU fault
 OB82_FUSE_FLT : BOOL ;              //Fuse fault
 OB82_HW_INTR_FLT : BOOL ;           //Hardware interrupt input fault
 OB82_RESERVED_3 : BOOL ;            //Reserved for system
 OB82_DATE_TIME : DATE_AND_TIME ;    //Date and time OB82 started
 t_req : BOOL ;                      //Trigger for RD_REC
 t_busy : BOOL ;                     //Busy from RD_REC
 t_return : INT ;                    //Return value from RD_REC
 t_laddr : WORD ;                    //Module address
END_VAR


BEGIN
NETWORK
TITLE =
        L       #OB82_MDL_ADDR;                  //Address of interrupting
                                                 //module
        T       #t_laddr;                        //Save
        SET     ;
        =       #t_request;                      //Trigger bit for RD_REC=1
loop:   NOP     0;
        CALL SFC  59 (
```

```
        REQ                 := #t_req,              //If 1: trigger read
        IOID                := B#16#54,             //Always C7 AS-i CP
        LADDR               := #t_laddr,            //Module address
        RECNUM              := B#16#1               //Data record number = 1
        RET_VAL             := #t_return            //Return for error
                                                    //or status info
        BUSY                := #t_busy              //Read job still active
        RECORD              := P#M 100.0 BYTE 11);  //11 read bytes are
                                                    //transferred starting at
                                                    //memory byte 100
                                                    //To simplify the
                                                    //program there is no
                                                    //evaluation of the return
                                                    //value included here
END_ORGANIZATION_BLOCK
```

6.5 Extended Operation with FC AS-i_3422

Overview

For extended operation, “FC AS-i_3422” is required on the C7-621 AS-i. This section explains which extra functions are available compared with standard operation when you operate the C7 AS-i CP with “AS-i_3422”.

Extended operation allows complete control of the behavior of the master by the user program. Access to the inputs/outputs is the same as in standard operation with the C7 AS-i CP. In the extended mode, a special FC in the user program is used for communication with the C7 AS-i CP.

Function

Commands are sent to the C7 AS-i CP by the user program using FC “AS-i_3422”. The user specifies the command call in a send buffer and starts the job. FC “AS-i_3422” transfers the command call to the C7 AS-i CP. On completion of the job, the job status is transferred to the user and any reply data are entered in a receive buffer.

Calling the Function

The function must be called cyclically by the user for every C7 AS-i CP. At any one time, only one job can be processed per C7 AS-i CP. An active job cannot be interrupted by the user and is not time monitored by the function.

STL representation

```
CALL  AS-i_3422        (ACT:=
                        STARTUP:=
                        LADDR:=
                        SEND:=
                        RECV:=
                        DONE:=
                        ERROR:=
                        STATUS:=);
```

LAD representation

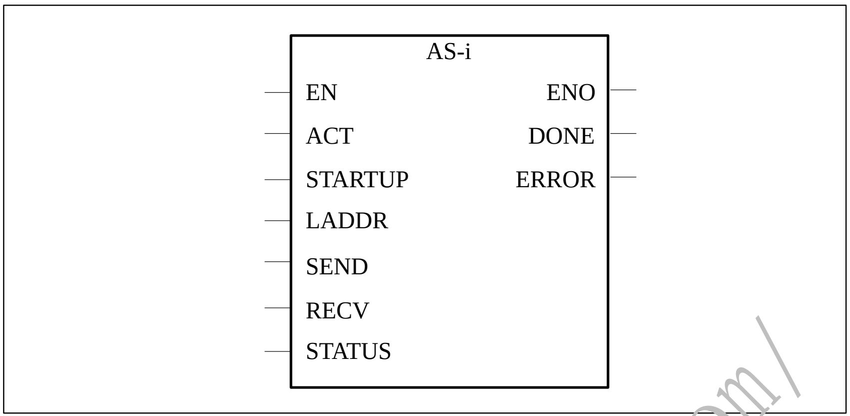


Figure 6-6 LAD Representation of the Call Function

Formal Parameters

The following table explains the formal parameters of the function:

Table 6-7 Formal Parameters of the Functions

Name	I/O	Type	Memory Area	Explanation
ACT	I	BOOL	I,Q,M,D,L, constant	Command processing by the function is level-triggered, in other words, as long as ACT = 1, execution of a command is started, as long as no other job is being executed.
STARTUP	I	BOOL	I,Q,M,D,L, constant	A C7 CPU startup is indicated to the function by STARTUP = 1. After the function is run for the first time, STARTUP must be reset by the user.
LADDR	I	WORD	I,Q,M,D,L, constant	Module start address The module start address can be queried within the framework of slot-oriented address assignment for signal modules (see Section 6.2.2).
SEND	I	ANY	I,Q,M,D,L	Send buffer This parameter indicates a memory area in which the command will be specified by the user. e.g.: P#DB20.DBX 20.0 byte 10

Table 6-7 Formal Parameters of the Functions, continued

Name	I/O	Type	Memory Area	Explanation
RECV	I	ANY	I,Q,M,D,L	Receive buffer This buffer is only relevant for commands that return data. The parameter indicates a memory area in which the reply to the command is entered. The length of the data area selected here is irrelevant. e.g.: P#DB30.DBX 20.0 byte 1
DONE	Q	BOOL	Q,M,D,L	DONE = 1 indicates 'Job complete no error'.
ERROR	Q	BOOL	Q,M,D,L	ERROR = 1 indicates 'Job complete with error'.
STATUS	I/Q	DWORD	M,D	1st word: job status / error code (see following table) If 'Job complete with error' is set, an error code is generated to describe the error in greater detail. 2nd word: required by the FC for internal purposes and must **not** be modified.

Error During Execution

If an error occurs during execution of the function, the BR bit has the value "0" in addition to the information in ERROR and STATUS. The BR bit is queried differently in LAD and STL user programs:

- LAD: query using the output parameter ENO
- STL: direct query of the BR bit

Status and Error Information

Table 6-8 lists the possible contents of the first word of STATUS depending on DONE and ERROR.

Table 6-8 Possible Contents of the First Word of STATUS

DONE	ERROR	STATUS	Meaning
0	0	8181_H	Job active
1	0	0000_H	Job complete no error
0	1	8090_H	Address in LADDR invalid
0	1	$80A0_H$	Negative acknowledgment when reading from the module
0	1	$80A1_H$	Negative acknowledgment when writing to the module
0	1	$80B0_H$	Module does not recognize data record
0	1	$80B1_H$	Specified data record length is wrong
0	1	$80C0_H$	Data record cannot be read
0	1	$80C1_H$	The specified data record is being processed
0	1	$80C2_H$	Job bottleneck
0	1	$80C3_H$	Resources (memory) being used
0	1	$80C4_H$	Communication error
0	1	8182_H	ID after complete restart (STARTUP=TRUE)
0	1	8184_H	Data type of the formal parameter RECV invalid
0	1	8381_H	Slave address wrong
0	1	8382_H	Slave not activated (not in LAS)
0	1	8383_H	Error on S7 interface
0	1	8384_H	Command not permitted (in current CP status)
0	1	8385_H	Slave 0 exists
0	1	$83A1_H$	Slave with address to be modified not found on S7 interface
0	1	$83A2_H$	Slave 0 exists
0	1	$83A3_H$	Slave with new address already exists on the S7 interface
0	1	$83A4_H$	Slave address cannot be deleted
0	1	$83A5_H$	Slave address cannot be set
0	1	$83A6_H$	Slave address cannot be permanently stored
0	1	$83F8_H$	Job number unknown
0	1	$83F9_H$	EEPROM error

Table 6-8 Possible Contents of the First Word of STATUS

DONE	ERROR	STATUS	Meaning
0	1	8F22H 8F23H	Area length error when reading a parameter Area length error when writing a parameter This error code indicates that a parameter is completely or partly outside the address range or that the length of a bit field with an ANY parameter cannot be divided by 8.
0	1	8F24H 8F25H	Area error when reading a parameter Area error when writing a parameter This error code indicates that a parameter is in an area that is illegal for a system function.
0	1	8F28H 8F29H	Alignment error when reading a parameter Alignment error when writing a parameter This error code indicates that the reference to a parameter is an address whose bit address is not 0.
0	1	8F30H 8F31H	Parameter is in write-protected shared DB Parameter is in write-protected instance DB This error code indicates that a parameter is in a write-protected data block.
0	1	8F32H	DB number in parameter too high
0	1	8F3AH	Parameter contains number of a DB that is not loaded
0	1	8F42H 8F43H	Access error occurred while the system attempted to read a parameter from the input area. Access error occurred while the system attempted to write a parameter to the output area.
0	1	8F44H 8F45H	This error code indicates that read access to a parameter was denied This error code indicates that write access to a parameter was denied
0	1	8F7FH	Internal error

Signal States of the Formal Parameters

A command call is started by ACT = 1. During the execution of a job, the first word of STATUS has the value 8181_H. This indicates that a job is being executed. On completion of the job, the user is informed of the result in the parameters DONE or ERROR.

- If no error occurred, DONE is set. If the job returns data from the C7 AS-i CP, the data are entered in the receive buffer identified by RECV. In this case, the value 0000_H is entered in the first word of STATUS.
- If an error occurred, ERROR is set. In this case, jobs involving return data from the C7 AS-i CP do not provide data. An error code is entered in the first word of STATUS to describe the error in greater detail.

The parameters DONE, ERROR, and STATUS remain unchanged until the next job is executed.

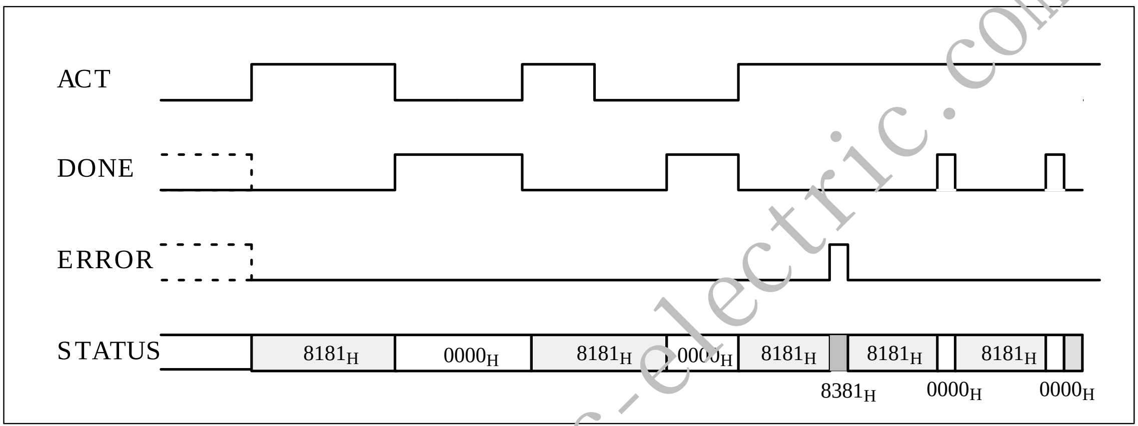


Figure 6-7 Signal States of the Formal Parameters

Block Data

The block length of the MC7 code and the number of local data bytes used depend on the version of the function. The current data can be found in the ‘Block Properties’ dialog in the STEP 7 Program Editor.

6.5.1 Interface for C7 AS-i CP Commands

Overview

This section describes the C7 AS-i CP-command calls sent by the C7 CPU to the C7 AS-i CP. With these command calls, the C7 AS-i CP makes the entire functionality of the master profile M1 of the AS-i master specification available. The C7 AS-i CP can also be configured completely by the C7 CPU using command calls.

Commands Supported by the C7 AS-i CP

How you use the jobs is explained in the descriptions of the jobs in Chapter 3 and in Appendix B.2 PICS and in the detailed descriptions in /1/ and /2/.

The commands that can be executed are listed in Table 6-9.

Table 6-9 Overview of the Command Calls

Name / Section	Parameter	Returned Data	Coding
Set_Permanent_Parameter	Slave address, Parameter		00_H
Get_Permanent_Parameter	Slave address	Parameter	01_H
Write_Parameter	Slave address, Parameter	Parameter echo (optional)	02_H
Read_Parameter	Slave address	Parameter value	03_H
Store_Actual_Parameter	None		04_H
Set_Permanent_Configuration	Slave address, configuration		05_H
Get_Permanent_Configuration	Slave address	Set configuration data	06_H
Store_Actual_Configuration	None		07_H
Read Actual Configuration	Slave address	Actual configuration data	08_H
Set_LPS	LPS		09_H
Set_Offline_Mode	Mode		$0A_H$
Set_Autoprogramming	Mode		$0B_H$
Set_Operation_Mode	Mode		$0C_H$
Change_Slave_Address	Address1, address2		$0D_H$
Read Slave Status	Slave address	Error record of slave	$0F_H$
Get Lists and Flags	None	LES,LAS,LPS,Flags	10_H
Read Total Configuration		Actual configuration data Current parameters LAS, Flags	19_H
Configure Total System	Total configuration		$1A_H$
Write Parameter List	List of parameters		$1C_H$
Read Parameter Echo List	None	Parameter echo list	13_H
Read Version ID	None	Version string	14_H
Read and Delete Slave Status	Slave address	Error record of the slaves	16_H
Read Slave ID	Slave address	ID code	17_H
Read Slave I/O	Slave address	I/O configuration	18_H

Set_Permanent_ Parameter

With this call, a parameter value for the specified slave is transferred to the C7 AS-i CP. The value is entered as a configured value in the configuration data.

The parameter is **not** transferred immediately to the slave by the C7 AS-i CP but only after a restart on the C7 CPU. The parameter value is transferred when the slave is activated.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	0 H
Byte 1	Slave address	
Byte 2	0	Parameter

Get_Permanent_Pa rameter

With this call, a parameter value for a specific slave in the EEPROM of the C7 AS-i CP is read.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	1 H
Byte 1	Slave address	

Structure of the return data in the receive buffer

	Bit 7	Bit 0
Byte 0	0 H	Parameter

Write_Parameter

With this call, a parameter value is transferred and is sent directly via the AS-i bus to the addressed slave. The parameter is stored in temporary storage on the C7 AS-i CP.

In the response, the slave returns the current parameter values that can deviate from the currently written values according to the AS-i master specification (/2/). These data are entered in the parameter echo field.

The RECEIVE job to fetch the parameter echo is optional. The parameter echo is not normally evaluated.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	2 H
Byte 1	Slave address	
Byte 2	0	Parameter

Structure of the return data in the receive buffer

	Bit 7	Bit 0
Byte 0	0	Parameter echo

Read_Parameter

With this call, the current parameter value (actual parameter) of a slave is returned.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	3 H
Byte 1	Slave address	

Structure of the return data in the receive buffer

	Bit 7	Bit 0
Byte 0	0	Parameter echo

Store_Actual_ Parameters

With this call, the configured parameters are overwritten by the actual parameters, in other words, the parameters of all slaves are configured.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	4 H

Set_Permanent_ Configuration

With this call, the I/O configuration data and the ID code for the addressed AS-i slave are configured. The data are stored permanently on the C7 AS-i CP.

Note

When executing this command, the C7 AS-i CP changes to the offline phase and then switches to the normal mode (complete restart on the CP). In the protected mode, this call is not executed.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	5 H
Byte 1	Slave address	
Byte 2	ID code	I/O configuration

Get_Permanent_ Configuration

With this call, the desired configuration data of an addressed slave stored permanently in the EEPROM (configuration data and the ID codes) are returned.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	6 H

Structure of the return data in the receive buffer

	Bit 7	Bit 0
Byte 0	ID code	I/O configuration

Store_Actual_ Configuration

With this call, the actual I/O configuration data and actual ID codes detected on the S7 interface are stored permanently in the EEPROM as the desired configuration data. The list of active slaves (LAS) is also entered in the list of configured slaves.

Note

When executing this command, the C7 AS-i CP changes to the offline phase and then switches to the normal mode (complete restart on the CP).

In the protected mode, this call is **not** executed.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	7 H

Read Actual Configuration

With this call, the actual I/O configuration data and actual ID codes of an addressed slave detected on the C7 AS-i CP are returned.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	8 H
Byte 1	Slave address	

Structure of the return data in the receive buffer

	Bit 7	Bit 0
Byte 0	ID code	I/O configuration

Set_LPS

With this call, the list of configured slaves is transferred for permanent storage in the EEPROM of the C7 AS-i CP.

Note

When executing this command, the C7 AS-i CP changes to the offline phase and then switches to the normal mode (complete restart on the C7 AS-i CP).

In the protected mode, this call is **not** executed.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	9 H
Byte 1	0 H	0 H
Byte 2	LPS slave 0..3 0 | 1 | 2 | 3	LPS slave 4..7 4 | 5 | 6 | 7
Byte 3	LPS slave 8..11 8 | 9 | 10 | 11	LPS slave 12..15 12 | 13 | 14 | 15
Byte 4	LPS slave 16..19 16 | 17 | 18 | 19	LPS slave 20..23 20 | 21 | 22 | 23
Byte 5	LPS slave 24..27 24 | 25 | 26 | 27	LPS slave 28..31 28 | 29 | 30 | 31

In the LPS data: 0 = slave not configured
1= slave configured

Set_Offline_Mode

This call switches between the online and offline mode.

The offline bit is **not** permanently stored, in other words, during a complete restart/restart the bit is set to online again.

In the offline mode, the CP only processes jobs from the user. There is no cyclic data exchange.

The **online mode** is the normal situation with the C7 AS-i CP. Here, the following jobs are processed cyclically:

- During the data exchange phase, the fields of the output data are transferred to the slave outputs for all slaves in the LAS. The addressed slaves transfer the values of the slave inputs to the master when the transfer was free of errors.
- This is followed by the inclusion phase in which there is a search for the slaves connected to the C7 AS-i CP and newly added slaves are entered in the LDS or LAS.
- In the management phase, jobs from the user such as writing parameters are executed.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	A H
Byte 1	0 H	Mode 0=Online 1=Offline

Set Autoprogramming

With this call, the automatic address programming function can be enabled or disabled.

The **AUTO_ADDR_ENABLE** bit is not set permanently on the master.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	B H
Byte 1	0 H	Mode 1=Autoprog enabled 0=Autoprog disabled

Set_Operation_ Mode

With this call, you can select between the configuration mode and the protected mode.

In the **protected mode**, only slaves marked in the LPS and whose desired and actual configuration match are activated, in other words, if the I/O configuration and the identification codes of the slaves in the LDS are identical to those of the configured values.

In the **configuration mode**, all detected slaves (except slave address “0”) are activated. This also applies to slaveseven if there is a difference between the desired and actual configuration. The OPERATION MODE bit is saved permanently in the EEPROM, in other words, it is retained following a complete restart/restart.

When changing from the configuration mode to the protected mode, the CP is restarted (transition to the offline phase followed by switchover to the online mode).

If the address 0 is entered in the LDS for a slave, the CP cannot switch from the configuration mode to the protected mode.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	C H
Byte 1	0 H	Mode 0=Protected mode 1=Configuration mode

Change_Slave_ Address

With this call, the slave address can be changed.

This call is mainly used to add a new AS-i slave with the default address 0 to the AS-i system. In this case the address is changed from the old slave address (0) to the new slave address.

This change can only be made when the following conditions are fulfilled.

- A slave with the old address exists.
- If the old slave address is not equal to 0, then a slave with address 0 cannot be connected at the same time.
- The new slave address must have a valid value.
- A slave with the new slave address must not already exist.

Note

When changing the slave address, the slave is not reset, in other words, data are retained until new data arrive for the new address.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	D H
Byte 1	Old slave address	
Byte 2	New slave address	

Read Slave Status

With this call, the status register of the addressed slave can be read out.

The flags of the status register have the following significance:

S0 “Address volatile”
This flag is set
- when the internal slave routine for permanent storage of the slave address is active. This can take up to 15 ms and must not be interrupted by a further addressing call.
- when the internal slave address comparison recognizes that the stored address is not the same as the entry in the address register.

S1 “Parity error detected”
This flag is set when the slave has recognized a parity error in a received frame since the last “read and delete status” job.

S2 “End bit error detected”
This flag is set when the slave has recognized an end bit error in a
received frame since the last “read and delete status” job.

S3 “Read error non-volatile memory”
This flag is set when a read error has occurred when reading the non-volatile memory.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	0 H	F H
Byte 1	Slave address	

Structure of the return data in the receive buffer

<table>
<tr><td></td><td>Bit 7</td><td>Bit 0</td></tr>
<tr><td>Byte 0</td><td>0</td><td>Slave status
S3 | S2 | S1 | S0</td></tr>
</table>

Get Lists and Flags

Get_LPS, Get_LAS, Get_LDS, Get_Flags: With this call, the following entries are read out of the C7 AS-i CP:

- the list of configured (permanent) slaves LPS,
- the list of active slaves LAS,
- the list of detected slaves LDS,
- the flags according to the AS-i specification.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	1 H	0 H

Structure of the return data in the receive buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	LAS slave 0..3	LAS slave 4..7
Byte 1	LAS slave 8..11	LAS slave 12..15
Byte 2	LAS slave 16..19	LAS slave 20..23
Byte 3	LAS slave 24..27	LAS slave 28..31
Byte 4	LDS slave 0..3	LDS slave 4..7
Byte 5	LDS slave 8..11	LDS slave 12..15
Byte 6	LDS slave 16..19	LDS slave 20..23
Byte 7	LDS slave 24..27	LDS slave 28..31
Byte 8	LPS slave 0..3	LPS slave 4..7
Byte 9	LPS slave 8..11	LPS slave 12..15
Byte 10	LPS slave 16..19	LPS slave 20..23
Byte 11	LPS slave 24..27	LPS slave 28..31
Byte 12	Flag 1	
Byte 13	Flag 2	

Flag 1

Name	Bit number
OFFLINE_READY	8
APF	9
NORMAL_MODE	10
CONFIG_MODE	11
AUTO_ADDR_AVAIL	12
AUTO_ADDR_ASSIGN	13
LES_0	14
CONFIG_OK	15

Flag 2

Name	Bit number
OFFLINE	0
RESERVED	1
EEPROM_OK	2
AUTO_ADDRESS_ENABLE	3
RESERVED	4
RESERVED	5
RESERVED	6
RESERVED	7

Table 6-10

Name of the Flag	Meaning of the Flag
OFFLINE_READY	This flag is set when the offline phase is active.
APF	This flag is set when the voltage on the AS-i cable is too low.
NORMAL_MODE	This flag is set when the C7 AS-i CP is in the normal mode.
CONFIG_MODE	This flag is set in the configuration mode and reset in the protected mode.
AUTO_ADDR_ AVAIL	This flag is set when automatic address programming is enabled. (This means that exactly **one** slave has failed).
AUTO_ADDR_ ASSIGN	This flag is set when automatic address programming is possible. (This means AUTO_ADDRESS_ENABLE = 1 **and** no 'bad' slave is / was attached to the C7 AS-i CP).
LES_0	This flag is set when a slave exists with address 0.
CONFIG_OK	This flag is set when the desired (configured) and actual configuration match.
OFFLINE	This flag is set when the CP is to change to the OFFLINE mode or is already in this mode.
EEPROM_OK	This flag is set when the test of the internal EEPROM did not detect any errors.
AUTO_ADDRESS_ ENABLE	This flag indicates whether autprogramming is disabled or enabled by the user (0=disabled; 1=enabled).

Read Total Configuration

This command reads the following data from the C7 AS-i CP:

- the list of active slaves (LAS). This shows which connected slaves are active,
- the current configuration data of the connected slaves (I/O configuration and ID code),
- the current parameters of the slaves (actual parameters),
- the current flags.

The command can be used, for example, to find out the configuration of the slaves connected to the AS-i cable after installation and startup. The configuration data that are read in can, if required, be modified and saved on the C7 AS-i CP as the desired configuration with the command 'Configure Total System'.

Structure of the job data in the send buffer

	Bit 7	Bit 0
Byte 0	1 H	9 H

Structure of the return data in the receive buffer

	Bit 7 Bit 4	Bit 3 Bit 0
Byte 0		
Byte 1		
Byte 2	LAS slave 0..3	LAS slave 4..7
Byte 3	LAS slave 8..11	LAS slave 12..15
Byte 4	LAS slave 16..19	LAS slave 20..23
Byte 5	LAS slave 24..27	LAS slave 28..31
Byte 6	ID code slave 0	I/O config. slave 0
Byte 7	ID code slave 1	I/O config. slave 1
Byte 8	ID code slave 2	I/O config. slave 2
Byte 9	ID code slave 3	I/O config. slave 3
Byte 10	ID code slave 4	I/O config. slave 4
Byte 11	ID code slave 5	I/O config. slave 5
Byte 12	ID code slave 6	I/O config. slave 6
Byte 13	ID code slave 7	I/O config. slave 7
Byte 14	ID code slave 8	I/O config. slave 8
Byte 15	ID code slave 9	I/O config. slave 9
Byte 16	ID code slave 10	I/O config. slave 10
Byte 17	ID code slave 11	I/O config. slave 11
Byte 18	ID code slave 12	I/O config. slave 12
Byte 19	ID code slave 13	I/O config. slave 13
Byte 20	ID code slave 14	I/O config. slave 14
Byte 21	ID code slave 15	I/O config. slave 15
Byte 22	ID code slave 16	I/O config. slave 16
Byte 23	ID code slave 17	I/O config. slave 17
Byte 24	ID code slave 18	I/O config. slave 18
Byte 25	ID code slave 19	I/O config. slave 19

Byte 26	ID code slave 20	I/O config. slave 20
Byte 27	ID code slave 21	I/O config. slave 21
Byte 28	ID code slave 22	I/O config. slave 22
Byte 29	ID code slave 23	I/O config. slave 23
Byte 30	ID code slave 24	I/O config. slave 24
Byte 31	ID code slave 25	I/O config. slave 25
Byte 32	ID code slave 26	I/O config. slave 26
Byte 33	ID code slave 27	I/O config. slave 27
Byte 34	ID code slave 28	I/O config. slave 28
Byte 35	ID code slave 29	I/O config. slave 29
Byte 36	ID code slave 30	I/O config. slave 30
Byte 37	ID code slave 31	I/O config. slave 31
Byte 38		Parameters slave 1
Byte 39	Parameters slave 2	Parameters slave 3
Byte 40	Parameters slave 4	Parameters slave 5
Byte 41	Parameters slave 6	Parameters slave 7
Byte 42	Parameters slave 8	Parameters slave 9
Byte 43	Parameters slave 10	Parameters slave 11
Byte 44	Parameters slave 12	Parameters slave 13
Byte 45	Parameters slave 14	Parameters slave 15
Byte 46	Parameters slave 16	Parameters slave 17
Byte 47	Parameters slave 18	Parameters slave 19
Byte 48	Parameters slave 20	Parameters slave 21
Byte 49	Parameters slave 22	Parameters slave 23
Byte 50	Parameters slave 24	Parameters slave 25
Byte 51	Parameters slave 26	Parameters slave 27
Byte 52	Parameters slave 28	Parameters slave 29
Byte 53	Parameters slave 30	Parameters slave 31
Byte 54	Flag 1	
Byte 55	Flag 2	

Flag 1

Name	Bit number
OFFLINE_READY	0
APF	1
NORMAL_MODE	2
CONFIG_MODE	3
AUTO_ADDR_AVAIL	4
AUTO_ADDR_ASSIGN	5
LES_0	6
CONFIG_OK	7

Flag 2

Name	Bit number
OFFLINE	0
RESERVED	1
EEPROM_OK	2
AUTO_ADDRESS_ENABLE	3
RESERVED	4
RESERVED	5
RESERVED	6
RESERVED	7

The meaning of the flags is the same as for the Get Lists and Flags job (Get_LPS, Get_LAS, Get_LDS, Get_Flags)”.

Configure Total system

This call transfers the desired total configuration to the C7 AS-i CP where it is stored as the desired configuration. This configures the C7 AS-i CP.

The following data are transferred:

- The list of configured slaves that indicates which slaves can be activated by the C7 AS-i CP in the protected mode.
- The list of configuration data that specifies which ID code and which I/O configuration the connected slaves must have.
- The list of permanent parameters on the C7 AS-i CP. These are transferred to the slaves when the C7 AS-i CP starts up.
- The flags that determine the mode of the C7 AS-i CP after startup.

Note

In the protected mode, this call is not executed.

Structure of the job data in the send buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	1 H	A H
Byte 1		
Byte 2	LPS slave 0..3	LPS slave 4..7
Byte 3	LPS slave 8..11	LPS slave 12..15
Byte 4	LPS slave 16..19	LPS slave 20..23
Byte 5	LPS slave 24..27	LPS slave 28..31
Byte 6	ID code slave 0	I/O config. slave 0
Byte 7	ID code slave 1	I/O config. slave 1
Byte 8	ID code slave 2	I/O config. slave 2
Byte 9	ID code slave 3	I/O config. slave 3
Byte 10	ID code slave 4	I/O config. slave 4
Byte 11	ID code slave 5	I/O config. slave 5
Byte 12	ID code slave 6	I/O config. slave 6
Byte 13	ID code slave 7	I/O config. slave 7
Byte 14	ID code slave 8	I/O config. slave 8
Byte 15	ID code slave 9	I/O config. slave 9
Byte 16	ID code slave 10	I/O config. slave 10
Byte 17	ID code slave 11	I/O config. slave 11
Byte 18	ID code slave 12	I/O config. slave 12
Byte 19	ID code slave 13	I/O config. slave 13
Byte 20	ID code slave 14	I/O config. slave 14
Byte 21	ID code slave 15	I/O config. slave 15
Byte 22	ID code slave 16	I/O config. slave 16
Byte 23	ID code slave 17	I/O config. slave 17
Byte 24	ID code slave 18	I/O config. slave 18
Byte 25	ID code slave 19	I/O config. slave 19
Byte 26	ID code slave 20	I/O config. slave 20
Byte 27	ID code slave 21	I/O config. slave 21
Byte 28	ID code slave 22	I/O config. slave 22
Byte 29	ID code slave 23	I/O config. slave 23
Byte 30	ID code slave 24	I/O config. slave 24

Byte 31	ID code slave 25	I/O config. slave 25
Byte 32	ID code slave 26	I/O config. slave 26
Byte 33	ID code slave 27	I/O config. slave 27
Byte 34	ID code slave 28	I/O config. slave 28
Byte 35	ID code slave 29	I/O config. slave 29
Byte 36	ID code slave 30	I/O config. slave 30
Byte 37	ID code slave 31	I/O config. slave 31
Byte 38		Parameters slave 1
Byte 39	Parameters slave 2	Parameters slave 3
Byte 40	Parameters slave 4	Parameters slave 5
Byte 41	Parameters slave 6	Parameters slave 7
Byte 42	Parameters slave 8	Parameters slave 9
Byte 43	Parameters slave 10	Parameters slave 11
Byte 44	Parameters slave 12	Parameters slave 13
Byte 45	Parameters slave 14	Parameters slave 15
Byte 46	Parameters slave 16	Parameters slave 17
Byte 47	Parameters slave 18	Parameters slave 19
Byte 48	Parameters slave 20	Parameters slave 21
Byte 49	Parameters slave 22	Parameters slave 23
Byte 50	Parameters slave 24	Parameters slave 25
Byte 51	Parameters slave 26	Parameters slave 27
Byte 52	Parameters slave 28	Parameters slave 29
Byte 53	Parameters slave 30	Parameters slave 31
Byte 54	Flag 1	
Byte 55	Flag 2	

Flag 1

Name	Bit number
OFFLINE_READY	0
APF	1
NORMAL_MODE	2
CONFIG_MODE	3
AUTO_ADDR_AVAIL	4
AUTO_ADDR_ASSIGN	5
LES_0	6
CONFIG_OK	7

Flag 2

Name	Bit number
OFFLINE	0
RESERVED	1
EEPROM_OK	2
AUTO_ADDRESS_ENABLE	3
RESERVED	4
RESERVED	5
RESERVED	6
RESERVED	7

Modifiable flags whose values change the mode of the C7 AS-i CP are shown on a shaded background:

CONFIG_MODE
: 0 = The C7 AS-i CP starts up in the protected mode after synchronization,
: 1 = The C7 AS-i CP starts up in the config mode.

AUTO_ADDRESS_ENABLE
: 0 = Automatic address programming disabled
: 1 = Automatic address programming enabled.

The values of the other flags are irrelevant for the ‘Configure Total System’ command and they cannot be modified.

Write Parameter List

With this command, the parameters for all slaves are transferred to the C7 AS-i CP. The C7 AS-i CP transfers **only** the parameters, **that have changed, in other words, differ from the current actual parameters**, to the slaves.

Structure of the job data in the send buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	1 H	C H
Byte 1		
Byte 2		Parameters slave 1
Byte 3	Parameters slave 2	Parameters slave 3
Byte 4	Parameters slave 4	Parameters slave 5
Byte 5	Parameters slave 6	Parameters slave 7
Byte 6	Parameters slave 8	Parameters slave 9
Byte 7	Parameters slave 10	Parameters slave 11
Byte 8	Parameters slave 12	Parameters slave 13
Byte 9	Parameters slave 14	Parameters slave 15
Byte 10	Parameters slave 16	Parameters slave 17
Byte 11	Parameters slave 18	Parameters slave 19
Byte 12	Parameters slave 20	Parameters slave 21
Byte 13	Parameters slave 22	Parameters slave 23
Byte 14	Parameters slave 24	Parameters slave 25
Byte 15	Parameters slave 26	Parameters slave 27
Byte 16	Parameters slave 28	Parameters slave 29
Byte 17	Parameters slave 30	Parameters slave 31

Read Parameter Echo List

When parameters are transferred to the slaves, the slaves return echo values in response. The Read Parameter Echo List call outputs the echo values of all the slaves. The echo values originate from the last parameter call sent to the slaves.

Structure of the job data in the send buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	1 H	3 H

Structure of the return data in the receive buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0		Par. echo slave 1
Byte 1	Par. echo slave 2	Par. echo slave 3
Byte 2	Par. echo slave 4	Par. echo slave 5
Byte 3	Par. echo slave 6	Par. echo slave 7
Byte 4	Par. echo slave 8	Par. echo slave 9
Byte 5	Par. echo slave 10	Par. echo slave 11
Byte 6	Par. echo slave 12	Par. echo slave 13
Byte 7	Par. echo slave 14	Par. echo slave 15
Byte 8	Par. echo slave 16	Par. echo slave 17
Byte 9	Par. echo slave 18	Par. echo slave 19
Byte 10	Par. echo slave 20	Par. echo slave 21

Byte 11	Par. echo slave 22	Par. echo slave 23
Byte 12	Par. echo slave 24	Par. echo slave 25
Byte 13	Par. echo slave 26	Par. echo slave 27
Byte 14	Par. echo slave 28	Par. echo slave 29
Byte 15	Par. echo slave 30	Par. echo slave 31

Read Version ID

This call outputs the version ID of the AS-i master software.

Structure of the job data in the send buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	1 H	4 H

The response of the C7 AS-i CP includes the name and the firmware version number of the C7 AS-i CP as shown below:

Structure of the return data in the receive buffer

	Bit 7 … Bit 0
Byte 0	S
Byte 1	i
Byte 2	e
Byte 3	m
Byte 4	e
Byte 5	n
Byte 6	s
Byte 7	
Byte 8	A
Byte 9	G
Byte 10	
Byte 11	
Byte 12	
Byte 13	C
Byte 14	7
Byte 15	6
Byte 16	2
Byte 17	1
Byte 18	
Byte 19	
Byte 20	A
Byte 21	S
Byte 22	I
Byte 23	
Byte 24	V
Byte 25	
Byte 26	x
Byte 27	.
Byte 28	y
Byte 29	y

x.yy stands for the current version number.

Read and Delete Slave Status

This call reads out the status of a slave and at the same time clears the status register of the slave.

The flags of the status register have the following meaning:

S0 “Address volatile”
This flag is set
– when the internal slave routine for permanent storage of the slave address is active. This can take up to 15 ms and must not be interrupted by a further addressing call.
– when the internal slave address comparison recognizes that the stored address is not the same as the entry in the address register.

S1 “Parity error detected”
This flag is set when the slave has recognized a parity error in a received frame since the last “read and delete status” job.

S2 “End bit error detected”
This flag is set when the slave has recognized an end bit error in a received frame since the last “read and delete status” job.

S3 “Read error non-volatile memory”
This flag is set when a read error has occurred when reading the non-volatile memory.

Structure of the job data in the send buffer

	Bit 7	Bit 4	Bit 3	Bit 0
Byte 0	1 H		6 H	
Byte 1	Slave address			

Structure of the return data in the receive buffer

	Bit 7	Bit 4	Bit 3	Bit 0
Byte 0	0 H		Slave status	

Read Slave ID

With this call, the ID code of a slave can be read out directly over the AS-i cable. The call is intended for diagnostic purposes and is not required in the normal master mode.

Structure of the job data in the send buffer

Byte 0	1 H	7 H
	Slave address	

Structure of the return data in the receive buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	0 H	Slave-ID

Read Slave I/O

With this call, the I/O configuration of a slave can be read out directly over the AS-i cable. The call is intended for diagnostic purposes and is not required in the normal master mode.

Structure of the job data in the send buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	1 H	8 H
	Slave address	

Structure of the return data in the receive buffer

	Bit 7 … Bit 4	Bit 3 … Bit 0
Byte 0	0 H	Slave I/O

6.6 AS-i Cycle Time and Number of Connected Slaves

Relationship

The following table shows the relationship between the AS-i cycle time and the number of connected slaves.

Table 6-11 AS-i Cycle Time and Number of Connected Slaves

Number of Slaves	Maximum Cycle Time in ms	
	typical	worst case
1...5	1.092	1.404
6...10	1.715	2.028
11...15	2.496	2.808
16...20	3.276	3.588
21...25	4.046	4.368
25...31	4.992	5.304

The “typical” times assume that there are no repeated messages, no management calls and that all slaves are synchronized.

If messages are repeated, the cycle time is extended by 0.156 ms per repetition. If a management phase occurs in the cycle, the cycle time is also extended by 0.156 ms.

If the AS-i network is correctly installed, it is assumed that a maximum of one message repetition occurs per cycle. If it is further assumed that a management call occurs in this cycle, the worst case cycle time is (typical time + 0.312 ms).

Operating the C7

7

Chapter Overview

Note

The explanations in this chapter relate to “standard screens” shipped in a configuration with *ProTool* or *ProTool/Lite*. Using these standard screens, you can call special screens. The standard screens can be completely redesigned to produce a customized user interface. The special screens, on the other hand, are part of the firmware of the C7 and cannot be modified.

7.1 Keyboard

Keyboard Design

The keyboard of the C7 consists of three functional blocks of keys (see Figure 7-1):

- System keys
- Softkeys (F keys)
- Numeric keypad

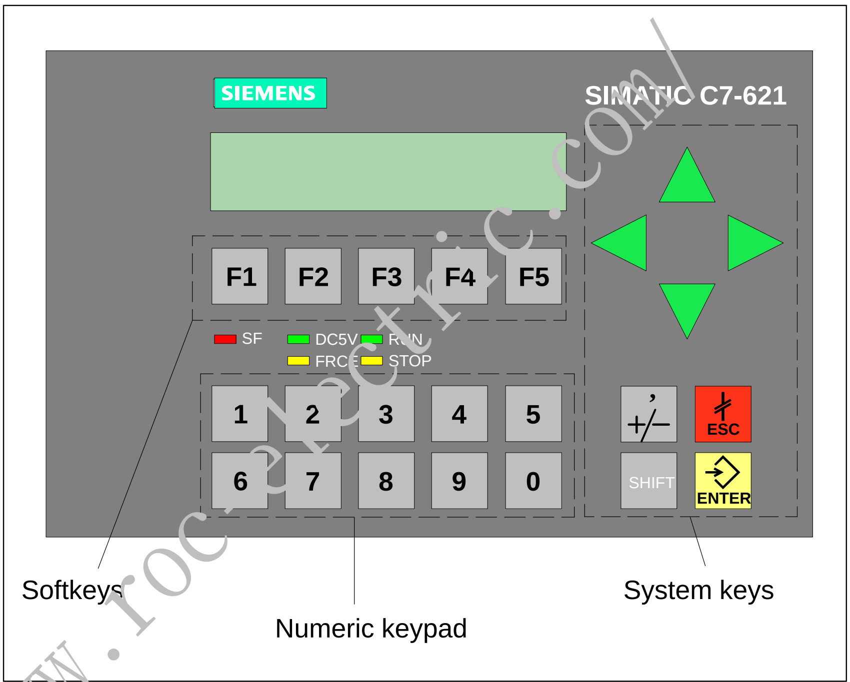


Figure 7-1 C7-621 with Keyboard and Display.

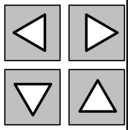
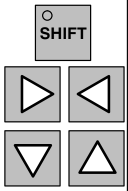
System Keys

Figure 7-1 shows the key block with the system keys. The function of the keys is explained in Table 7-1.

Table 7-1 Key Functions

Key	Function	Explanation
SHIFT	Shift key	Switches to the second function of dual-function keys. Press the SHIFT key at the same time as the other key.
0 to 9	Numeric keys	Input keys for numeric characters (0 to 9) The numeric keys have a repeat function. If you hold down the keys, the input is repeated at brief intervals until you release the key again.
+/−	Sign key	Changes the sign from “Plus” to “Minus” and vice versa.
SHIFT, +/−	Decimal key	Inputs a decimal point. Press the SHIFT key at the same time as the required key.
ENTER	ENTER	With this key, you confirm and complete your input. This key is also used, for example, to change from the message level to the screen level.
ESC	ESCAPE	Cancel, go back, change operating level **Undo** You can undo entries you have made provided you have not already entered them with with the ENTER key. **Branch back** You can branch back from a screen to a configured destination (as default, this is the last position you called). By branching further back, you finally return to the start screen and then to the message level. **Reset when scrolling messages** Cancels scrolling in the pending messages and returns the display to the current pending message. **Clear a system message from the display** A displayed system message is cleared.

Table 7-1 Key Functions, continued

Key	Function	Explanation
	Arrow keys	These keys move the cursor. Depending on the current situation, the cursor is moved character by character, field by field or from display to display to the right, left, up, or down. These keys also have a repeat function. if you hold down the key, the input is repeated at brief intervals until you release the key.
	Page	Next /previous position in the input field Page through the character set or stored text.

Note

Pressing more than one key at the same time can lead to incorrect input.

Softkeys

The softkeys F1 to F5 below the display can be assigned different functions depending on the current display.

Note

By pressing several keys in quick succession, it is possible to “lose” a keystroke. Input not accepted by the C7 OP is indicated by an acoustic signal.

7.2 Operating Levels

Overview

When working with the C7, there are two distinct operating levels and you can switch from one to the other:

- **Message level**

  At the message level, current messages are displayed.
- **Screen level**

  At the screen level, you select, define and execute functions.

Message Level

The message level is the highest level on the C7. At the message level, pending event messages and system messages are displayed. After the C7 starts up, it changes to the message level and displays the standby message.

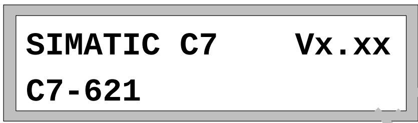


Figure 7-2 C7-621 Standby Message (System Message No. 0)

Screen Level

After you have changed to the screen level, the first screen to be called is the start screen. From the start screen you branch to other screens depending on the configuration. On the screens, you view the actual process values, and you can enter values and initiate functions by means of softkeys.

The linking of individual screens is referred to as a screen hierarchy (see Section 9.2). If you have worked your way into the screen hierarchy, you return stage by stage back to the start screen by pressing the ESC key repeatedly (see Section 7.1).

Changing the Operating Level

You change the operating level as follows:

- from message level to screen level by pressing ENTER
- from screen level to message level by pressing ESC

You cannot branch further back from the message level by pressing ESCAPE. At this level. the key is only used to clear the display of a system message.

Figure 7-3 shows how you switch from one operating level to the other.

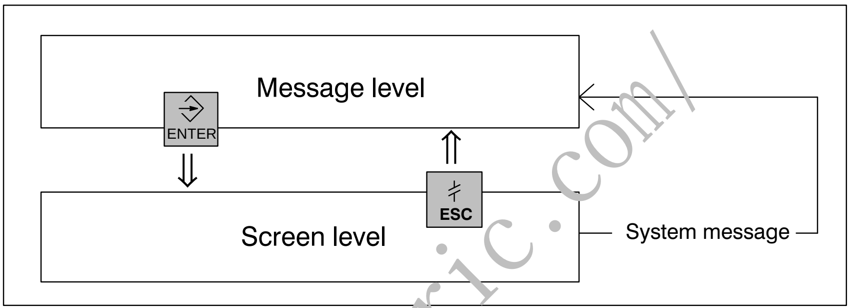


Figure 7-3 Changing between the Message and Screen Level

7.3 Working with Standard Functions

Overview

A configuration is shipped with the configuration software “ProTool” or “ProTool/Lite” that contains standard screens.

Using these standard screens, you can activate all the functions necessary for operation. The description of the individual functions in this manual is based on the standard screens. If you want to work with the standard screens, load the configuration shipped with the software. This configuration is in the directory ProTool\Standard\C7 621.pdb.

If you have not loaded a configuration, the English standard screens are active after turning on the power. These are loaded from the firmware of the C7-OP (see Section 2.1).

Note

The supplied standard screens call special screens that are loaded in the firmware of the C7. You can also call these screens in your own configuration with the “Special screen selection” function.

The standard screens contain functions that are fundamental to the basic operation of C7, such as Display Screens, Modify Password and Set C7 OP Operating Mode. Process-specific implementations, such as event messages or screens for the process, are not included.

Standard Screens

Standard screens are called from a basic screen by pressing a softkey. From the basic screen, you branch to the following screens:

- Screens
  The screen directory is called to display screens. All the screens which were assigned the “directory” attribute are listed here. If you have still not created any screens of your own, the directory will contain only two standard screens, “Counter” and “Timer” (see Section 8.3).
- System settings
  You can modify settings in the online mode. This includes, for example, selecting the C7 mode, changing languages, or setting the date and time.

- Status Variable

  The PG function STATUS VAR is called; you can use this to display PLC addresses.

- Force Variable

  The PG function FORCE VAR is called; you can use it to display and modify PLC addresses.

- Password processing

  The superuser assigns the passwords for the different password levels. The logout function is also included here.

Screen Hierarchy of the Standard Screens

Standard configurations, with ready-to-use standard screens, are supplied for the C7-621 with the ProTool configuration software. Figure 7-4 shows the screen hierarchy for these standard screens. You will find detailed information about the functions and how to use standard screens in the corresponding sections of this manual.

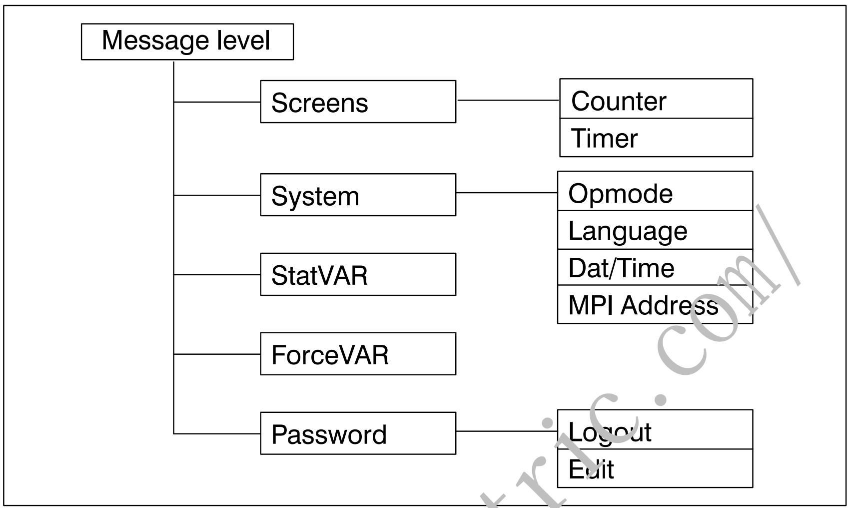


Figure 7-4 Screen hierarchy of the Shipped Standard Screens

Internal Standard Screens

If there is no configuration available on the C7-621, for example when you first start up, the English standard screens (Figure 7-5) are loaded from the memory of the C7-621.

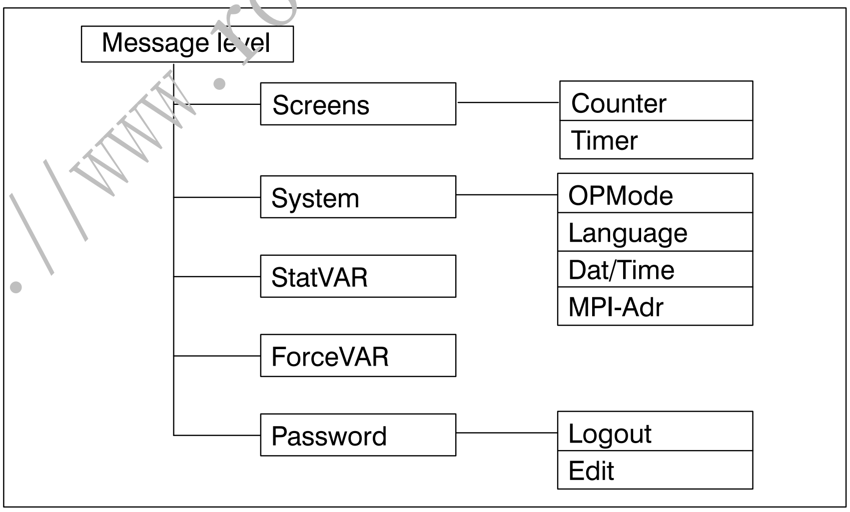


Figure 7-5 The Internal Standard Screens of the C7 OP

Branching in Standard Screens

With the ENTER key, you change from the message level to the screen level. At the screen level, you can operate and monitor the process or system using the appropriate screens and standard screens and make system settings.

Example

Taking standard screens as an example, the description below explains how to branch from one screen to another within the screen hierarchy.

Step	Activity
1.	Call the standard **basic screen** in your configured screen hierarchy.
2.	Using the softkeys beneath the symbols « and », you can move the displayed screen segment of the active screen (scroll screen function).
3.	You can branch to the next screen by pressing the softkey beneath the screen text. A vertical line designates the assigned softkey.

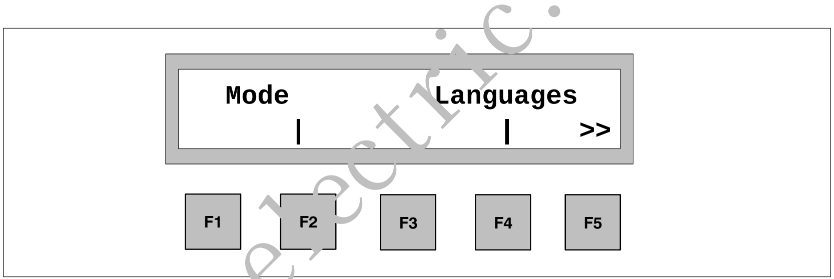


Figure 7-6 Branching at the Screen Level

Selecting a Screen

You select a screen by pressing the softkey assigned to it during configuration.

If either of the symbols << or >> is displayed at the beginning or the end of the second line on the display, you can use the screen scroll function for further selections with F1 or F5, if the entry you require is not within the visible display area.

Calling a Function

Functions are called by means of the softkeys assigned to them during configuration.

To protect against unauthorized use, a password with a suitable password level must be entered first for some functions (see Section 7.7).

Figure 7-6 illustrates branching to different functions.

7.3.1 Setting C7 OP Modes

Overview

You can set the Online, Offline, and Transfer modes on the C7 OP using standard screens.. Table 7-2 explains these modes.

Table 7-2 C7 OP Modes

Mode	Explanation
Online	In the online mode, there is a logical link between the C7 OP and the C7 CPU, or the C7 attempts to establish a link.
Offline	In the offline mode, there is no logical link between the C7 OP and the C7 CPU. The C7 OP does not attempt to establish a link and variables are not updated.
Transfer	In the transfer mode, data can be downloaded from your PG or PC to the C7 OP. You cannot operate the C7 OP while the download operation is in progress.

Configuration not Loaded

If the standard screen for changing the **C7 OP mode** is not loaded, you cannot load a configuration. You must then run a memory reset on the C7-OP (see also Section 2.1).

Transfer to the C7-OP

Configuration data are always transferred from the programming device or PC to the C7 OP. Downloading to the destination device is described in the manuals for *ProTool* or *ProTool/Lite*. In these manuals, simply substitute C7-621 for OP.

Setting Modes

To set the C7 OP modes, follow the steps below:

Step	Activity	Result
1.	Select the standard screen **System** → **Mode**. by pressing F4 and then F2	`Mode Online` (Example)
2.	Shift + △ or Shift + ▽ to select one of the modes listed in Table 7-2 (Online, Offline or Transfer).	`Mode MPITrns` (Example)
3.	Confirm your selection by pressing ENTER	The required mode is set and the following message is displayed: • If you select Online mode: `$ 110 Normal_Mode` • If you select Offline mode: `$ 135 Of-fline_Mode` • If you select Transfer mode: `Ready for Transfer` Data (configurations) can only be transferred from the PG/PC to the OP after changing to the “Transfer” mode. If you change the mode, a restart is run on the OP.
4.	Quit the standard screen with ESC	The previous screen is displayed again.

7.4 Modifying the Address in the MPI Network Configuration

Setting in the Standard Screen

You can set and modify the address of the C7 OP in the MPI network configuration in a standard screen by following the steps below:

Procedure

1. Select the standard screen *System → MPI Address.*
2. Change the C7 OP address and, if necessary, the transmission rate.
3. After you confirm with ENTER, the C7 OP restarts.

7.5 Entering Values

Overview

You enter values in the input fields of screens on the C7 that will be transferred to the C7 CPU. There are different types of values and different ways of entering values.

- Numeric values
- Alphanumeric values
- Symbolic values

Entering Values in General

The general steps for entering values are as follows:

Step	Activity	Result
1.	First branch to the required screen as described on page 7-10 and then to the required screen entry.	The required screen is displayed.
2.	Using the cursor keys, select the input field in the screen entry.	Cursor is located in the input field.
3.	Enter the required value. Depending on how the field was configured, you can enter one of the value types listed above in “Overview” (see pages following).	The input field flashes.
4.	Confirm your input with ENTER Clear incorrect entries with ESC	The value is entered and the field no longer flashes. The original value of the field is automatically restored.
5.	If required, position the cursor on another input field and make the next entry as described above. With the cursor keys, you can also move the cursor left to the previous input field and enter a different value.	
6.	Close the screen with ESC	The display is reset and you change back to the previous screen.

Numeric Values

In fields that allow you to enter a numeric value, you enter the numeric value character by character using the numeric keypad.

If there is a value in the field already, it is cleared completely from the field when the first character is entered.

Once input has started, you cannot exit the input field until the input has been entered or canceled.

Entry of Numeric Values

In numeric fields, input is usually right-justified. Digits that have already been entered are moved to the left.

Exception: Input fields for setpoints in BIN format (for example, when calling the PG functions STATUS/FORCE VAR) are changed to left-justified. When you start to input the value, the old value does not disappear from the display completely but its bit pattern is overwritten one character at a time. You move the cursor in this type of field by simultaneously pressing the SHIFT key and an arrow key [◁] and [▷]

Decimal Point

You enter a decimal point by pressing the following keys simultaneously

[SHIFT] [. +/–]

Limit Values

You can configure **limit values** for numeric input fields. In this type of field, a limit value check is made. Entered values are accepted only if they are within the configured limits. If a value outside these limits is entered, a system message is displayed and, after it has been canceled, the old value is displayed again.

Field with Decimal Places

If a numeric field has been configured with a certain number of **decimal places** and if, after you confirm your input, too many decimal places have been entered, the extra ones are ignored; if too few have been entered, the field is padded with zeros.

Alphanumeric Values

When you input alphanumeric values, digits and letters are mixed.

For the numerical components of the input, follow the steps as described in "numeric values".

Entering Alphanumeric Values

If you want to enter a letter at the current cursor position, you must activate the alphanumeric character set.

To enter the string 18OCT61, for example, proceed as follows:

Step	Activity	Result
1.	You must first configure the input field with the variable. As an example, you can select a string with the format CHR. Enter 1 and 8 by using the numeric keypad.	`Variable:18......`
2.	Press and hold down SHIFT	The extended character set becomes available.
3.	Page up with △ in the permitted character set. You obtain other special characters with ▽	`Variable:18......`
4.	Select **O** and move the cursor one place to the right.	The selected character is entered when you move the cursor.
5.	Select **C** and move the cursor one place to the right.	as above
6.	Select **T** and move the cursor one place to the right.	as above
7.	Release SHIFT	The extended character set is deactivated.
8.	Enter the remaining figures 6 and 1 using the numeric keypad and confirm with ENTER	`Variable:18OCT61.` Your entry is accepted and display of the entry is reset.

Symbolic Values

If a field requires symbolic values, you select the symbol from a list.

To enter a symbolic value, follow the steps below:

Step	Activity	Result
1.	Press and hold down SHIFT while in the input field	The list with the configured symbolic values is displayed.
2.	Select the required value with the cursor key.	
3.	Release the SHIFT key.	
3.	Confirm the selected value with ENTER	

7.6 Setting C7 System Parameters and Language

Overview

Once you have loaded a configuration, you can modify the following system settings for the C7 OP resulting from the configuration using standard screens:

- Language
- Date and time

Setting the Language

Messages and screens can be displayed in several languages. Up to three of the languages listed below can be loaded simultaneously on the C7 and can be selected in the online mode:

- German
- English
- French
- Italian
- Spanish

To select another language, follow the steps below:

Step	Activity	Result
1.	Change to the screen level, if you are not already at this level.	The basic screen is displayed.
2.	Select the standard screen **System → Language** by pressing F4 and then F4	`Language   ENGLISH`
3.	• Select the required language by pressing Shift + △ or Shift + ▽ The list only contains languages that were loaded on the C7. • Complete your selection with ENTER	The C7 OP now restarts and displays all language-dependent texts in the new language.

7.6.1 Setting the Date and Time

Date and Time Setting

You can adjust the current date and time on the C7 OP. The day of the week is calculated internally. Any change you make will affect all messages and screens with which a date or time variable is displayed. The display format for date and time is defined in your configuration and cannot be modified on the C7.

To set the date and time, follow the steps below:

Step	Activity	Result
1.	Select the screen level “System Settings” from the basic screen with **System**. Then select the standard screen **Dat/Time**.	Date: Fr 01.01.95 Time: 01:30:40
2.	Set the day of the week holding down SHIFT and pressing △ or ▽	
3.	Confirm the setting with ENTER	
4.	Press SHIFT ◁ or ▷ to move the cursor right and left in the date field.	The cursor moves forwards or backwards in the field.
5.	Enter the data as a numeric value. remember to press the shift key first.	
6.	You can change between the date and time fields with the cursor keys.	The cursor is in the field for the time of day.
7.	Enter the time as in Step 4.	
5.	Confirm with ENTER	The new setting is entered.
6.	Quit the standard screen with ESC	You return to the previously set screen level.

Note

The C7 OP does not have a hardware clock. Since the date and time are generated by software, this information must be updated every time the C7 OP starts up.

7.7 Password Protection

Overview

To prevent unauthorized operation of the C7, it is possible to control access by means of passwords and password levels that are required to call certain functions and make certain settings.

When you assign a password to an operator, you grant permission to execute functions at a specific password level.

A password level is specified during configuration.

If password protection is active you must log in and out on the C7.

7.7.1 Password Level and Access Rights

Password Level

During configuration, hierarchical password levels from 0 to 9 are assigned for softkeys and input fields. The password levels of the standard screens are listed in Appendix C.2.

When you assign a password to an operator, you grant permission to execute functions at a specific password level.

If an operator logs in on the C7 with a password for a certain level, he/she is authorized to execute functions with that password level and at lower levels.

Table 7-3 Password Levels

Password Level	Explanation
0	At this level, the lowest in the hierarchy, the functions available have little or no effect on the execution of the process; these are normally calls for functions that do not involve input. You do not need to enter a password to call password level 0 functions. If you call a function assigned to a higher password level than 0, you will be prompted to enter a suitable password.
1-8	Levels 1 to 8 are assigned to functions of increasing importance. The superuser assigns the password level to a password using password management functions.
9	Permission to execute functions of password level 9 is reserved for the superuser, who has access to all C7 functions. The password for level 9 functions is entered during configuration.

7.7.2 Logging In and Out on the C7 OP

Logging In

If a function is called on the C7 for which the current password level is too low, you are automatically prompted on the display to enter the required password.

You complete password input by pressing the ENTER key.

Logging Out

Select the standard screen **Password** → **Logout** to log out from the C7 OP. The C7 then changes from the current password level to 0, the lowest password level, and then branches to the message level.

7.7.3 Password Management

Overview

Only the superuser (password level 9) is authorized to call the password management functions (**Password → Edit)**.

When shipped the password for the C7 is **100**. This can be changed during configuration.

A maximum of 20 passwords can be assigned. A password must have at least 3 digits and can be up to 8 digits long. Leading zeros and letters are not permitted. The password for level 9 functions is assigned during configuration (default: **100**).

Displaying the Password List

Select the standard screen **Password → Edit.**

The password list is displayed. It can contain a maximum of 50 passwords.

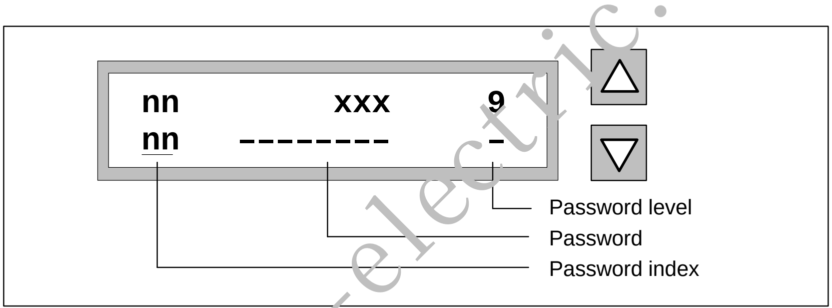


Figure 7-7 Example of a Password List

Table 7-4 Explanation of Figure 7-7

Label in Figure	Explanation
Password index	Passwords are numbered consecutively with a two-digit password index. The fields for the password and its assigned password level are on the right of the password index. Only the superuser entry is contained in the fields when the password list is called for the first time.
Password	To the right is the field for the password. You can scroll the list with the cursor keys.
Password level	The first time the list is displayed, it contains only the superuser entry.

Assigning a Password and Password Level

To assign a password and a password level, follow the steps below:

Step	Activity	Result
1.	Select the line for the password entry in the password list.	The cursor is located on the first character of the field for password input.
2.	Enter a new password and confirm it by pressing ENTER. Leading zeros are not permitted.	
3.	Move the cursor with the right arrow key to the field for the password level.	
4.	Enter a password level of 1 to 8 for the password and confirm it by pressing ENTER.	
5.	Quit the standard screen with ESC	

Changing a Password and Password Level

To change a password, call the password entry in the same way as you would to assign a password and enter the new password by overwriting the old one.

If you just want to modify the password level and not the password, skip the field containing the password entry by pressing ENTER. Then move the cursor with the RIGHT arrow key to the field for the password level and enter the new level.

Deleting a Password

To delete a password, call the password entry in the same way as you would to assign and change a password but overwrite the first character of the password with a zero. Then confirm the deletion by pressing ENTER.

7.8 Hardware Test

Overview

Apart from a brief startup test (”eprom test”, “ram test”, “flash test”) that is run through at every complete restart on the C7, you can also run a hardware test to check all the important components of the device.

Regardless of the selected language, the hardware test is always displayed in English.

Starting the Hardware Test

The hardware test is not started at the screen level but when the power is turned on. To start the test, press the three cursor keys at the same time

◁ ▽ ▷

Test Sequence

You can select individual components in the hardware test menu:

- Select components with the “up” and “down” cursor keys.
- Start the test with ENTER.
- The symbols >> and << at the right and left edges of the display indicate that further menu items are available outside the visible section of the display.

While a test is running the message “active” is displayed. If the test was completed without finding errors or faults, the message “OK” is displayed for approximately 2 seconds.

End of the Test

If you do not press a key for 5 seconds, the test is stopped and the menu is displayed again.

Components that Can Be Tested

The table lists the components that can be tested:

Table 7-5 Tested Components

Component	Explanation
CPU TEST	The internal registers, the timers and the interrupt controller of the processor are tested.
RAM TEST	The entire static RAM is tested by writing and then reading it. Its previous contents are overwritten.
EPROM TEST	The checksums of the memories are formed.
FLASH TEST	During the FLASH TEST, the size and status of the FLASH memory is displayed (for example 128 K, “empty” or “prg.” for programmed).
KEYBOARD TEST	When you press one of the system keys, the name of the key, for example “ENTER”, is displayed. The value of numeric keys is displayed.
DISPLAY TEST	The following test screens are displayed one after the other: 1. Display dark 2. Display lit 3. The cursor runs through all the display locations in both display lines from left to right and back.
END OF TEST	A hardware reset is triggered followed by a complete restart.

Standard Operator Control and Monitoring Functions

8

Chapter Overview

8.1 Messages

Overview

Events and states in the control process are displayed on the C7 in message form. A message always consists of static text and may also contain variables.

The following types of message are displayed on the C7:

- Event messages
- System messages

Selecting the Message Level

To change to the message level, press ESC

Quitting the Message Level

To quit the message level, press ENTER

Event Messages

Event messages are initiated by the PLC. They are configured and contain process-related information.

System Messages

System messages are initiated by the C7 OP. They are not configured. They provide information about operating states of the C7 OP or incorrect operations and communication problems.

8.1.1 Event Messages

Overview

Event messages contain process-related information (for example, messages relating to states or processes) such as

`Temperature reached` or
`Motor running`

Apart from status messages, notices to operators can also be configured as event messages. If, for example, a machine operator wants to start a bottling system but has forgotten to open the water inlet valve on the mixer, a message such as `Open water intake valve` can prompt him to take appropriate action.

Presentation

Event messages can be configured so that particular text components flash to highlight them compared with the rest of the message text.

Messages can contain static text and variable fields. The variable fields display, for example, current values of the C7 CPU in numeric or symbolic form. The date and time can also be output in messages.

Message Bits

If there is a condition present in the current process that causes a message to be generated (for example, a setpoint has been reached), a bit is set by the application program in the data area for event messages. The C7 reads the data area after a configured polling time. In this way, a message is detected as “entering the state”. The bit is reset by the C7 CPU when the condition that caused the message no longer exists. The message is then ”leaving state”.

Event Message Area

You must define an *event message area* for event messages in your configuration. In ProTool or ProTool/Lite, you set the event message area by selecting *System* → *Area Pointer* from the menu.

You can configure a single event message for every bit that has been configured in the event message area. The event message area (up to 64 bytes) can be divided into a maximum of 4 address areas. The address areas do not need to be contiguous.

Figure 8-1 shows the assignment of bit numbers to message numbers for data bytes. Bit numbers are assigned automatically to message numbers on the C7 OP.

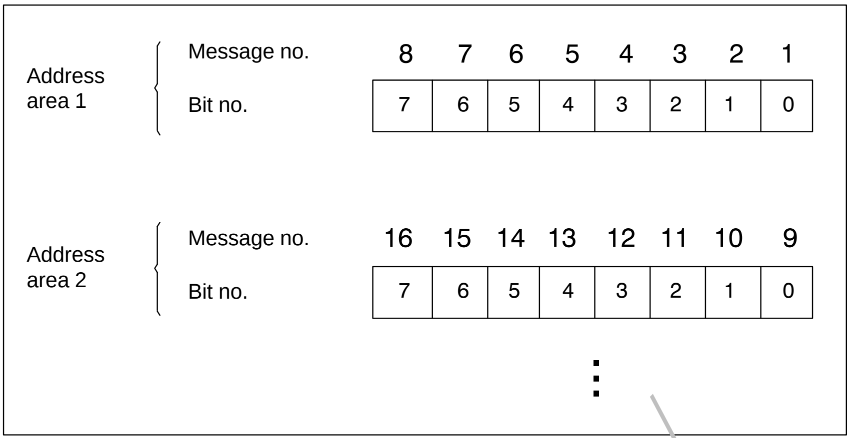


Figure 8-1 Event Message Area and Message Numbers

Updating Messages

When the C7 detects a message for an event entering the state, it reads the value to be displayed for the message variables supplied by the C7 CPU and displays them. The fields defined in the messages are updated periodically at the configured polling intervals.

If the event leaves the state while the message is displayed, the display is updated and the next message is displayed automatically.

Standby Message

The standby message is event message No. 0. It appears on the display when the C7 is working at the message level and there are no event messages or system messages pending. The standby message is stored in the firmware and as default contains the release and the device type:

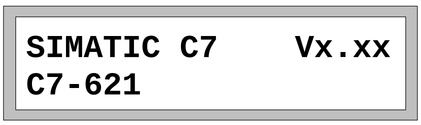


Figure 8-2 Standby Message of the C7-621 and C7-621 AS-i

Depending on the configuration, the standby message can be represented by other text. It can contain the date and time but not variables.

8.1.2 System Messages

Overview

System messages indicate operating states within the C7 OP. For example, indicate incorrect operations or communication problems.

Displaying System Messages

This message type has top display priority. If a fault occurs on the C7, the active event message is removed from the display and a system message is displayed in its place.

After the system message has been cleared from the screen, the C7 returns to the point from which it branched.

Fatal/Non-Fatal System Messages

System messages are classified as fatal and non-fatal system messages.

- A **fatal** system message results from an error that can be rectified only by a complete restart or restart of the C7.
- **Non-fatal** system message: All other errors generate a non-fatal system message, for example, when a particular screen cannot currently be selected. The display of a non-fatal system message can be canceled by pressing ESCAPE or it can also be canceled automatically after a configured time.

  A list of possible system messages and their explanations can be found in Appendix B.

Disabling System Messages

The display of system messages (except for internal errors 7xx) can be disabled during configuration. This setting on the C7 OP cannot be changed later.

8.1.3 Displaying Messages

Overview

Event messages are always output to the display at the message level on the C7 and are displayed according to display and message priorities. Messages are displayed one at a time on the C7, even if they have been configured as single-line messages.

Selecting the Message Level

You change to the message level by pressing ESC

Priorities

The messages have different display priorities .

Table 8-1 Explanation of the Priorities

Priority	Explanation
Display priority	System messages always have top display priority. Event messages are displayed according to their message priorities.
Message priority	During configuration, you can set message priorities for event messages from 1 (low) to 4 (high) according to their importance.

If several messages having the same display and message priorities exist simultaneously, the most recent message is shown first.

Example

Table 8-2 Order of Arrival and Display of Messages

Order of Arrival	Order of Display
1) Event message A (priority 2) 2) Event message B (priority 3) 3) Event message C (priority 2) 4) System message A 5) Event message D (priority 4)	1) System message A 2) Event message D (priority 4) 3) Event message B (priority 3) 4) Event message C (newer with priority 2) 5) Event message A (older with priority 2)

Message Buffer

The C7 OP message buffer stores the fifty latest messages in the order in which they arrive. When the message buffer is full, the oldest message is overwritten.

Message “Shower”

If there are more than fifty messages at any one time (message shower), only the fifty current messages contained in the buffer will be displayed. Any other messages that may be waiting cannot be displayed even when messages leave the state. When it reads the event message area, the C7 detects only a status change of the bits. Since the bit status of waiting messages that have yet been entered in the buffer has not changed, the C7 does not then detect these messages as having “entered state”.

Paging Pending Event Messages

If there is currently no system message, you can scroll at the message level through the messages that have not yet left the state. Event messages are sorted according to priority groups and are displayed in their order of arrival.

To page through the messages, use the cursor keys at the message level.

Before you can scroll through waiting messages starting from the message being currently displayed, you must first change to the scroll mode using the ↓ or ↑ keys:

Table 8-3

	Display of the next older (or next lower priority) message. Following the oldest message in a priority group, the most recent message of the priority group with lower priority is displayed. The end of the message area is marked by ”↓↓↓”. You cannot scroll beyond this end mark.
	Display of previous (or next higher priority) message. Following the most recent message in a priority group, the oldest message of the group with the next higher priority is displayed. The beginning of the message area is marked by ”↑↑↑”. You cannot scroll beyond this start mark.

Redisplaying a Message

The currently pending message is displayed again if you press ESCAPE or if you make no input to the C7 OP for one minute.

8.2 Screens

Overview

On the C7, the process (for example, a bottling plant or a mixing unit) is displayed and controlled in screens. These screens are customized for the particular application.

In screens, logically associated process values are acquired and provide an overview of a process or system. In addition to this alphanumeric “visualization” of the process, screens allow you to enter new process values and so control the process. Up to 40 screens can be configured on the C7.

Process values in a screen can be freely assigned to subject-related groups.

Example:

```
Temperature tank 1: 80 C
Temperature tank 2: 78 C
Level tank 1: 1200 l
Level tank 2: 3000 l
Pressure valve 1: normal
Pressure valve 2: high
```

Components of a Screen

A screen consists of the following components:

- Title
- Screen entries (maximum of 40).

Screen Directory

Screens can be grouped during configuration in a screen directory, which is used to display them on the screen and also to edit them. A screen can be found in the screen directory by its screen number and its screen title, if configured.

Selecting a Screen

You can select a screen using either:

- Softkeys
- Screen directory

Table 8-4 explains the ways of selecting a screen.

Table 8-4 Ways of Selecting a Screen

Selection	Explanation
Selection with softkey	With soft keys, you can branch from one screen to another The branch is defined in the configuration.
Selection from screen directory	Call the standard screen **Screens**. The screen directory is then displayed on the screen. It contains only the screens which were included in it during configuration. Enter the number of the screen you require or "scroll" in the screen directory using the arrow keys. In either case, press ENTER to display the screen.

Exiting the Screen Level

To exit the screen level, press ESC

Editing Screens

You can enter values in screens. To edit a screen, follow the steps below:

Step	Activity	Result
1.	Select the screen you want to edit as described in “Selecting a Screen”.	The screen is displayed. The cursor jumps to the first input field.
2.	Move the cursor with the keys ◁ ▷ to the required field.	The cursor is located in the input field.
3.	Make the required modifications as described in Section 7.5.	The cursor is located at the end of your input.
4.	After confirming your input, reposition the cursor to make any further modifications you require.	The cursor is located at the end of your input.
5.	Quit editing with ESC	You return to the previous level.

Screen Entries

Screens consist of one or more entries. Each screen can have up to 40 entries. On the C7 OP, only one entry is displayed per display page. Lines which have not been fully configured are displayed as blank lines.

An example of a screen entry might be:

```
Temp. tank 1: 80 °C
Temp. tank 2: 78 °C
```

Components of a Screen Entry

A screen entry consists of the following components:

- Text
  The static text contains explanations for the operator. It may also include information on how softkeys have been assigned.
- Fields for
  - Displaying
    - Date
    - Time,
    - C7 CPU actual values
  - Input of C7 CPU setpoints that are transferred immediately after to the C7 CPU
  - Combined input/output of C7 CPU setpoint and actual values.
- Softkeys
  Soft keys are assigned variable, screen dependent functions.

Updating Values in Screen Entries

The configuration defines the intervals at which C7 CPU values are updated, in other words the intervals at which they are read from the C7 CPU and displayed. The lowest configured polling time applies to the whole screen entry.

To optimize performance, you should do the following:

- Configure the polling times for updating as high as possible
- Configure short polling times only for those entries which really do need to be updated quickly.

Input and Output Fields

Input and output fields have the following characteristics:

- Input fields define setpoints in numeric or symbolic form.
- In input fields, the flashing cursor is visible.
- Output fields display actual values of the C7 CPU in numerical or symbolic form.
- For symbolic I/O fields, you can configure up to *256* text elements, which you can call on the C7 using a selection field. The value you select is entered.
- When entering numeric values, configured number formats apply or limit values restricting the number of places before and after the decimal point.

8.3 Timers and Counters

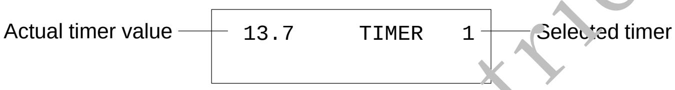
Overview

With the C7 OP, you can access the timers and counters of the C7 CPU. Examples of this can be found in the C7 standard screens. The following description of how to access timers and counters refers to the ”Timer” and ”Counter” standard screens. You can select these standard screens from the screen directory on the C7 OP.

8.3.1 Timer

Displaying Timer Actual Values

On the C7 OP, you can display the current actual value of every timer configured and enabled on the C7 CPU by calling the *Screens -> Timers* standard screen. The following display appears (example):



To exit the standard screen, press ESC.

Editing Timer Settings

You do not enter timer setting on the C7 OP. The values for the timers are supplied from the C7 CPU accumulator when the timer is called.

Time Base

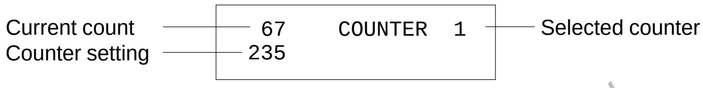
The common time base for the timers can be configured (10 ms, 100 ms, 1 s or 10 s). The C7 OP detects the selected time base and scales the displayed value to seconds.

8.3.2 Counters

Displaying/Editing Counter Values

On the C7 OP, you can display the current count and, if configured, you can change the counter setting for every counter configured and activated on the C7 CPU. Follow the steps below:

1. Call the *Screens -> Counters* standard screen. The following display appears (example):



2. If you do not wish to modify the setting, exit the standard screen by pressing ESC.
3. Use the arrow keys to select a counter (for example counter 3). The cursor is located in the “Counter setting” field.
4. Using the keypad, modify the counter setting.
5. Confirm your entry by pressing ENTER.
6. If necessary, repeat steps 3. to 5. for other counters.
7. Exit the screen by pressing ESC.

8.4 STATUS VAR and FORCE VAR with the C7 OP

Overview

By configuring special standard screen similar to the PG functions STATUS VAR and FORCE VAR, the C7 can be used to display and modify address values. In the online mode, this means that C7 CPU addresses can be manipulated directly on the C7 without having to connect a programming device or a PC to the C7.

STATUS VAR

STATUS VAR can be used only to display addresses of a C7 CPU.

FORCE VAR

FORCE VAR is used to display the addresses of a C7 CPU and to modify their variable values and to transfer them back to the C7 CPU. The selected addresses are retained in the static RAM and also apply after the C7 restarts.

Calling FORCE VAR

You call the FORCE VAR function from the **Force VAR** standard screen. After the call, the address list is displayed.

Figure 8-3 illustrates the SIMATIC S7 representation.

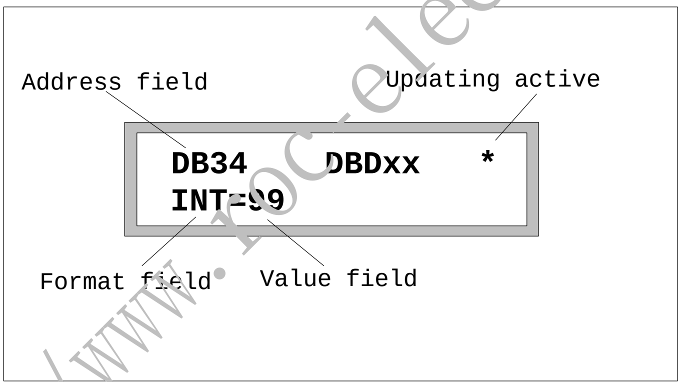


Figure 8-3 Example of the Display of an Address

Addresses

You can scroll up and down through the addresses. Holding down the SHIFT key, you can select the data type to be displayed using the cursor keys. Press ENTER to set the corresponding data format in the format field.

PLC Address

The PLC address is the MPI node address of the selected CPU. You can set this address.

Using FORCE VAR

Table 8-5 shows you how to work with FORCE VAR.

Table 8-5 Operator Controls

Activity	Key
General: Confirm each entry in a field with ENTER:	ENTER
Move to the numeric field with:	▷
Move the cursor in a line with:	◁ ▷
Within lines and the field with a value, you can move the cursor horizontally. A total of 10 lines can be assigned. The values of the addresses you select are displayed in the value field in the specified format.	▽ △
Move up and down in the column with the **addresses** with:	▽ △
If the cursor is in the column for **addresses** you can select the data type to be displayed (DB, MW, IW, QW, C, T) with SHIFT and:	SHIFT ▽ △
If the cursor is in the column for the **format**, you can set the data formats HEX, DEC, BIN, CHR, T, C by pressing the above keys	
Enter the number of the address you want to display or modify using the numeric keypad.	0....9

Updating Values

When you have finished editing the address list, the values on the C7 CPU must be updated. This is not done immediately after an individual value has been confirmed. The new values are not transferred to the C7 CPU until you press the ENTER key again after confirming the final value. During updating, a flashing asterisk * is displayed in the top right corner of the display. If the asterisk does not flash, this means that no logical link has been established to the C7 CPU.

Canceling Updating

Inputs cannot be made while updating is in progress. Updating can be canceled by pressing ESC.

Permitted Data Types

The table shows the data types permitted for a SIMATIC S7-300.

Address	Data Type
SIMATIC S7-300	
DB, M	CHAR BYTE INT WORD DINT DWORD REAL BOOL STRING TIMER COUNTER
I, PI, Q, PQ	CHAR BYTE INT WORD DINT DWORD REAL BOOL STRING
T	TIMER
C	COUNTER

Extended Operator Control and Monitoring Functions

9

Chapter Overview

9.1 Process-Dependent Operator Control and Monitoring

Overview

The actions that are required or permitted vary as the process situation changes. To meet changing requirements during process control, you can configure the following to support the operator:

- Screen-dependent softkeys
- User-defined screen hierarchies

Calling Functions with Softkeys

You can configure function calls on the C7 OP using softkeys. Softkeys are special function keys to which different function calls are assigned for different screen entries while you edit a screen. This allows the operator to select functions as and when required by the situation. The keys that can be assigned as softkeys on the C7 OP are F1 to F5.

The functions that can be assigned to softkeys include:

- Display screen level or message level
- Select screen
- Display screen directory
- Display special screen
- Logout

On the C7 OP, a bit can be configured in a variable for every softkey. This means that a bit is set on the C7 CPU when a softkey is pressed.

9.2 Self-Defined Screen Hierarchy

Overview

The screen hierarchy can be adapted to system-specific requirements and can be modified either in part or in whole. Screens can be removed or added.

Any combination of screens can be linked together. The configuration, sequence of the screen linking, inclusion in the screen directory and the relevant return jump destinations are defined during configuration with ProTool/Lite.

Branching with Softkeys and Returning

You branch between the different screens using softkeys and configured return jump destinations. You can branch to the same screen from entries in different screens as illustrated in Figure 9-1. Return jumps are not restricted to screen level allowing you to branch to the message level.

Definition of the Start Screen

During configuration, you can also decide on the layout of the screen you want displayed on the C7 OP as your start screen.

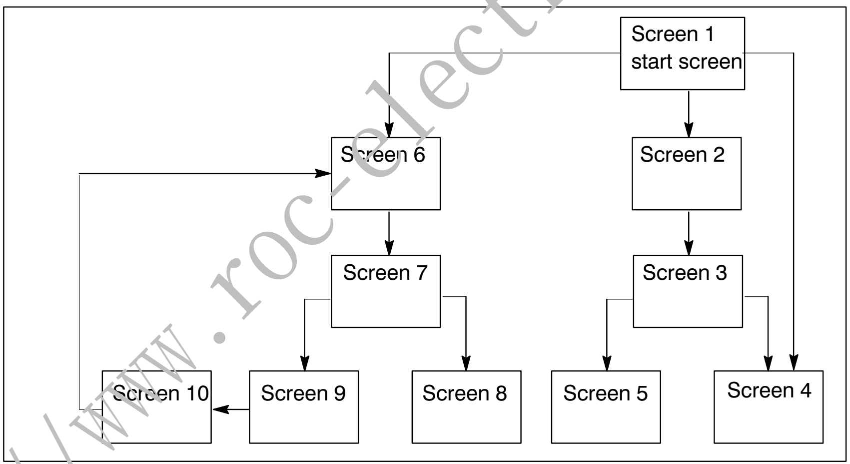


Figure 9-1 How the Screen Hierarchy Works

Example of a Screen Hierarchy

The example below illustrates the structure of a screen hierarchy. For more detailed information, refer to the *ProTool* or *ProTool//Lite* User's Guide.

Example:

The C7 OP is used to operate and monitor a system for producing and bottling different fruit juices. The system consists basically of a mixing unit and a bottling machine.

Mixing unit The ingredients for the fruit juices are contained in three tanks. Depending on the juice that you wish to manufacture, ingredients are mixed in certain ratios.

Bottling Unit After it has been mixed, the fruit juice flows into the bottling tank after a valve has been opened and it is then bottled in the correct quantities. The bottles are conveyed on a belt. Before being filled, they are checked for breakages. After they have been filled, the bottles are capped, labeled and transferred to pallets.

Start Screen

The configured start screen could, for example, appear as shown in Figure 9-2. This screen consists of static text only.

```
  Mix      Bottle
<<   |          |   >>
```

Figure 9-2 Start Screen for the Plant (Example)

The screen segment on the display can be selected with the softkeys beneath the display. The display can be moved horizontally with the symbols « and ».

Selecting “Mix”

Pressing the softkey beneath the ”Mix” entry allows you to view the entry shown in Figure 9-3. It also consists of only static text that identifies further screens (“Tank2”, “Tank3” and “Mixer”).

```
 Tank2  Tank3  Mixer
<< |     |       |
```

Figure 9-3 Static Screen Text (Example)

Selecting “Tank2”

If you press the “Tank2” softkey, the entry shown in Figure 9-4 appears. This entry contains static text and an output field (Contents) and an input field (Valve). The position of the tank valve can be set in the input field by entering a symbolic value (for example, OPEN or CLOSED).

```
Contents:  371 Liter
Valve: Open       >>
```

Figure 9-4 Screen with Input and Output Fields (Example)

9.3 Evaluation of the Screen Number

Overview

The screen number area is located on the C7 CPU. The C7 OP writes the number of the current screen to this area. If the C7 CPU writes a screen number to the screen number area, the screen is opened on the C7 OP. This allows you to configure operator support.

Configuring the Screen Number Area

If you want to use the screen number area, it must be specified during configuration as an area pointer and created on the C7 CPU. Figure 9-5 shows the structure of the screen number area.

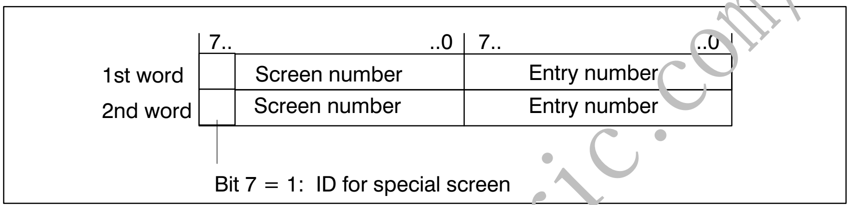


Figure 9-5 Structure of the Screen Number Area on the C7 CPU

The screen number area consists of two consecutive data words. The first data word is used by the C7 OP to store information about the display contents (screen number and entry number).

If the C7 CPU stores a screen number and entry number in the second data word, the display of a specific screen or a specific screen entry is initiated on the C7 OP.

The hexadecimal value FFFF in the first or the second data word indicates the message level; a value of 0 in the second data word indicates enabling of C7 OP operation.

Special Screens

If the most significant bit is set in the data word (=1), the screen number refers to a special screen. If the most significant bit is not been set (=0), the screen is a user-defined screen.

The screen numbers of the special screens are listed in the table below. An offset of 128 (most significant bit = 1) must be added to these screen numbers.

Table 9-1 Screen Numbers

Screen Number	Screen
0	Screen directory
25	Status Variable
26	Force Variable
30	Language Selection
31	Changing the Operating Mode
35	Set Time/Date
36	MPI Address/Baud Rate
55	Password Login
56	Password Edit

Screen Selection by the C7 CPU

The following diagram illustrates how the C7 CPU selects screen 5:

1. A screen is open on the C7 OP.

	Screen number	Entry number
1st word	x	x
2nd word	x	x

2. Before the user program enters the value 5 in the 2nd word of the screen number field, it must briefly set the screen number to 0....

	Screen number	Entry number
1st word	x	x
2nd word	0	x

   ... and enter the value 5 at the earliest after one polling cycle (1 second).

	Screen number	Entry number
1st word	x	x
2nd word	5	x

3. The C7 OP recognizes the change from 0 to 5 and opens screen 5.

	Screen number	Entry number
1st word	5	
2nd word	5	

9.4 Image of the System Keyboard

Overview

Each key of the system keyboard (except for the cursor keys) is assigned a bit in the data area for system keyboard bits. As long as the key is pressed, the bit remains set. When the key is released, the bit is reset.

By evaluating this data area, it is possible, for example, to generate an error message to indicate incorrect operator input.

Configuring the System Keyboard Image

Before the data area for system keyboard bits can be used, it must be specified during configuration as an area pointer and created on the C7 CPU. The system keyboard assignment is a data area with a fixed length of two data words. Figure 9-6 shows its configuration for a C7 OP.

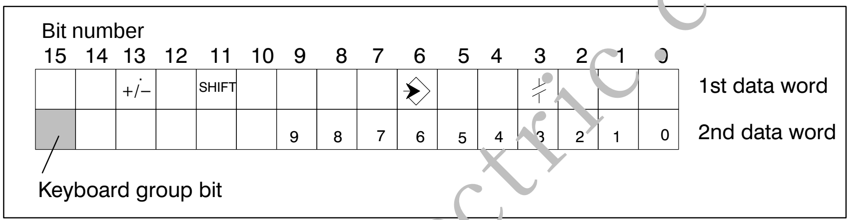


Figure 9-6 Keyboard Assignment for the C7 OP

The keyboard image is transferred spontaneously to the C7 CPU whenever a modification on the C7 OP is registered. There is therefore no need to configure a polling time.

The keyboard group bit is used as a check bit. It is set to 1 every time the keyboard image is transferred from the C7 OP to the C7 CPU and should be reset by the user program after the data area has been evaluated. Regular reading of the group bit makes it possible for the user program to check whether the system keyboard image has been transferred again.

9.5 Communication

Types of Connection

The C7 OP can be connected to SIMATIC S7 controllers via a network configuration. The following type of connection is possible:

- SIMATIC S7-300 Multipoint interface connection (MPI)

The type of connection affects the configuration and the mode of addressing used.

User Data Areas

The C7 OP and the SIMATIC S7 controller communicate via user data areas on the programmable controller. The user data areas you create on the S7 PLC depend on the configuration. You create user data areas suitable for the objects contained in the configuration and the data to be exchanged.

For some user data areas, you must create an interface area to handle synchronization of the C7 OP and the S7 PLC, if the functions are to be used by the S7. Some user data areas are even located in this interface area.

For the C7 OP, the following user data areas are possible:

- Event message area (see Section 8.1.1)
- Interface area for the connection ID, date and time
- Screen number area (see Section 9.3)
- Image of the system keyboard (see Section 9.4).

Note

The following applies to data areas:

- The system keyboard image and screen number area must only be created once.
- The interface area can be created only once for each CPU.
- All other user data areas can be created more than once on different CPUs

---

9.5.1 Connection to the S7-300 via MPI

Connection

When connecting a C7 OP to an S7-300, the C7 OP is connected to the MPI interface of the S7 CPU. You can connect up to two S7 CPUs to a C7 OP. Up to four C7 OPs can communicate with an S7 CPU simultaneously. The CPU determines the maximum number of connections. A maximum of 32 nodes can communicate in an MPI network configuration.

Network Configuration

Figure 9-7 illustrates a possible network configuration. The numbers `1, 2`, etc. are examples of the addresses. The addresses on the S7 CPU are specified with *S7 Configuration*.

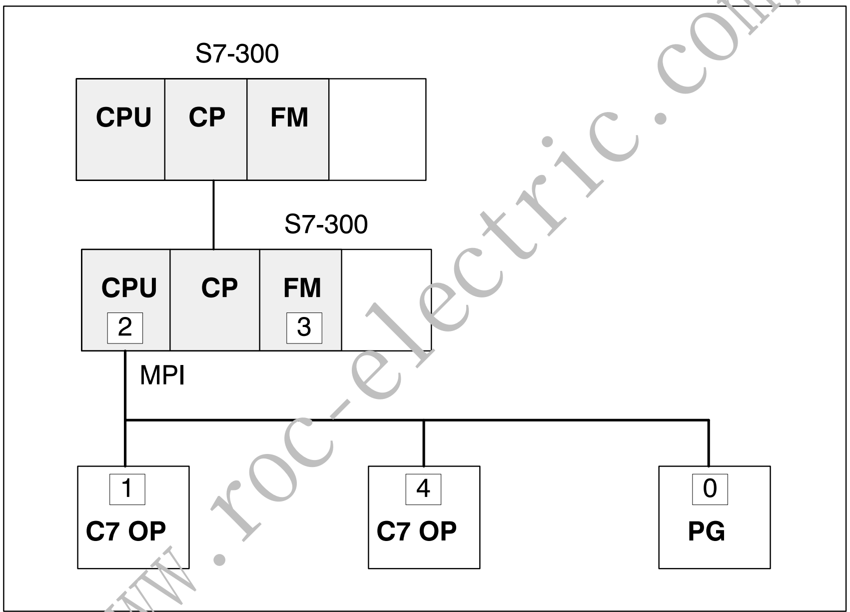


Figure 9-7 Connecting the C7 OP to the SIMATIC S7-300

Parameters

The following parameters must be configured in the configuration software for a connection via the MPI:

Address of the communication partner	MPI address of the S7 module to which the C7 OP is connected. The default address is 2.
Slot	The number of the slot containing the S7 module with which the C7 OP exchanges data.
Rack	The number of the rack containing the S7 module with which the C7 OP exchanges data.
C7 OP address	The MPI address of the C7 OP in the network configuration. Any address can be assigned. It must be unique in the network configuration and may not occur more than once. The default address is 1.
HSA	Highest station address. The address must be identical in the whole network configuration.
Interface	The interface on the C7 OP through which it is connected to the MPI network. The default is `IF 1A`.
Profile	The protocol profile that is used in the network configuration. Set MPI here.
Baud rate	The transmission rate at which communication takes place in the network configuration.
Interface area	If data user areas are used that are located in the interface area, you must create an interface area. You must configure a separate interface area for each S7 CPU connected.

Settings in ProTool or ProTool/Lite

With ProTool or ProTool/Lite, all settings with the exception of the interface area must be performed by selecting *System –> C7 CPU* from the menu. You configure the interface areas by choosing *System –> Area Pointer* from the menu.

9.5.2 Interface Area in SIMATIC S7

Purpose

The interface area is required only if the following functions are used or evaluated by the SIMATIC S7:

- Synchronize the date and time of the S7 and the C7 OP
- Evaluate connection ID
- Detect C7 OP startup in the S7 program.

Structure

Figure 9-8 shows the structure of the interface area. You can create the interface area in a data block or in a bit memory area on the SIMATIC S7-300.

The address of the interface area must be specified in the configuration. This is necessary so that the C7 OP knows where the data are located.

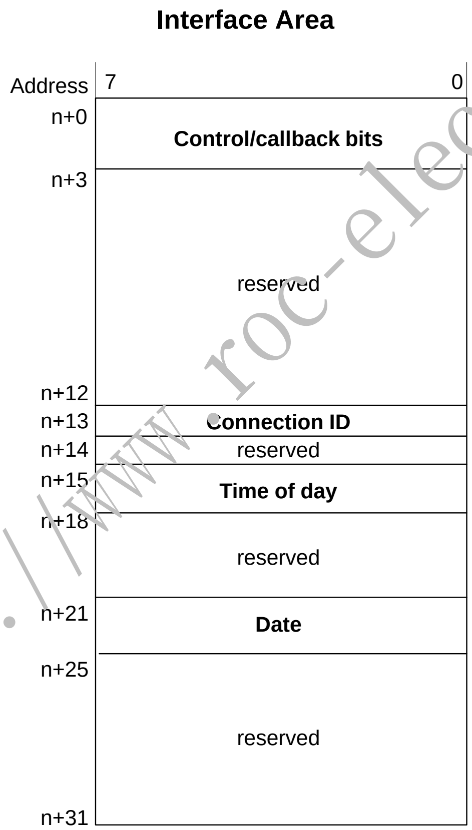


Figure 9-8 Structure of the Interface area on a SIMATIC S7 CPU

9.5.3 Control and Callback Bits

Purpose

Three bytes are present in the interface area for the control and callback bytes. Bytes n+0 and n+1 are used to synchronize the C7 OP and the S7 CPU. Byte n+3 is irrelevant for the C7 OP.

Byte n+0:
Request Date/Time

Byte n+0 is used by the C7 OP to request the current time and the date from the S7 CPU. The structure of the data byte is shown below.

7	6	5	4	3	2	1	0

Bit 2: Request date/time

Bit 2 Request date/time
`1` = C7-OP requests the date and time
`0` = The S7 program has updated the data and time in the interface area

Byte n+1:
Detect OP Startup

Byte n+1 is used by the S7 CPU to detect the C7 OP restarting. The structure of the data byte is shown below.

7	6	5	4	3	2	1	0

Bit 0: C7 OP startup

Bit 0 `1` = C7-OP has started up

Bit 0 in data byte n + 1 is set by the C7 OP on completion of startup. You can reset the bit in the S7 program and detect C7 OP restart.

9.5.4 Connection ID

Byte n+13

The C7 OP enters the connection ID (MPI) in byte n+13.
This allows the connection ID to be evaluated by the S7 CPU.
The structure of the data byte is shown below.

7 … 0
Connection ID

`0` = Connection via MPI

9.5.5 Time and Date

Purpose

The current time and date are stored by the S7 program in bytes n+15 to n+17 and n+21 to n+24. This allows the C7 OP to synchronize the time and date with the S7 CPU.

Bytes n+15 to n+17: Time

Bytes n+15 to n+17 contain the current time of the S7 in BCD. The structure of the data byte is shown below.

Address	7 … 0
n+15	Hour (0...23)
n+16	Minute (0...59)
n+17	Second (0...59)

Bytes n+21 to n+24: Date

Bytes n+21 to n+24 contain the current date of the S7 in BCD. The structure of the data byte is shown below.

Address	7 … 0
n+21	Day of week (1...7)
n+22	Day (1...31)
n+23	Month (1...12)
n+24	Year (0...99)

Synchronization with the S7 CPU

Synchronization of the C7 OP and the SIMATIC S7 CPU is performed in three steps:

1. Every hour, the C7 OP sets bit 2 in data byte n+0 to 1.
2. As soon as you reset bit 2, the C7 OP detects that the S7 program has entered up-to-date values for the time and date in the interface area.
3. The C7 OP reads the up-to-date data from data bytes n+15 to n+17 and n+21 to n+24 of the interface area.

SFCs, SFBs, and IEC Functions on the C7 CPU

A

Chapter Overview

A.1 SFCs and SFBs

Overview

The C7 CPU provides you with various system functions, for example for program execution and diagnostics. You call these system functions in your user program using the number of the SFC or SFB.

For a detailed description of all the system functions, refer to the Reference manual **/235/**.

Clock Functions

For clock functions, the C7 CPU provides the following integrated functions:

SFC	No.	Name	Description	Execution Time
SFC	0	SET_CLK	Set clock If the clock to be set is a master clock, time-of-day synchronization is started at the same time. If the clock to be set is a slave clock, only the clock is set.	120 µs
SFC	1	READ_CLK	Read clock	190 µs
SFC	2	SET_RTM	Set run-time meter You can set one run-time meter on the C7 CPU.	65 µs
SFC	3	CTRL_RTM	Start or stop run-time meter	55 µs
SFC	4	READ_RTM	Read run-time meter	90 µs
SFC	64	TIME_TICK	Read system time You can read the system time with an accuracy of ms	45 µs

Block Functions

The following table lists system functions for copying and setting the variables of a field.

SFC	No.	Name	Description	Execution Time
SFC	20	BLKMOV	Copy variables of any type	90 µs+ 2 µs/byte
SFC	21	FILL	Preassign the value in a field	90 µs+ 3.2 µs/byte

Creating a Data Block

You create a data block with SFC22 “CREAT_DB”.

SFC	No.	Name	Description	Execution Time
SFC	22	CREAT_DB	Creates a data block with a specified length in a selected area.	110 µs+ 3.5 µs per DB in the specified area

Time-of-Day Interrupts

You can use the time-of-day interrupts for internal time-driven program execution controlled internally by the C7 CPU.

SFC	No.	Name	Description	Execution Time
SFC	28	SET_TINT	Set time-of-day interrupt	190 µs
SFC	29	CAN_TINT	Cancel time-of-day interrupt	50 µs
SFC	30	ACT_TINT	Activate time-of-day interrupt	50 µs
SFC	31	QRY_TINT	Query time-of-day interrupt	85 µs

Time-Delay Interrupt

The operating system starts time-delay interrupts after a specified time has elapsed.

SFC	No.	Name	Description	Execution Time
SFC	32	SRT_DINT	Start time-delay interrupt	85 µs
SFC	33	CAN_DINT	Cancel time-delay interrupt	50 µs
SFC	34	QRY_DINT	Query started time-delay interrupts	80 µs

Interrupt and Error Handling

For reactions to interrupts and errors, the C7 CPU provides the following system functions.

SFC	No.	Name	Description	Execution Time
SFC	36	MSK_FLT	Mask synchronous errors	150 µs
SFC	37	DMSK_FLT	Unmask synchronous errors	160 µs
SFC	38	READ_ERR	Query and delete programming and access errors	160 µs
SFC	39	DIS_IRT	Disable processing of new interrupts	215 µs
SFC	40	EN_IRT	Enable processing of new interrupts	305 µs
SFC	41	DIS_AIRT	Delay processing of new interrupts and asynchronus errors	35 µs
SFC	42	EN_AIRT	Enable processing of new interrupts and asynchronus errors	35 µs
SFC	43	RE_TRIGR	Retrigger watchdog (cycle time monitoring)	30 µs
SFC	44	REPL_VAL	Copy substitute value to accumulator 1 of the priority class that caused the error	45 µs

Mode Changes

With the following system functions, you can control mode changes.

SFC	No.	Name	Description	Execution Time
SFC	46	STP	Change to C7 CPU to the STOP mode	-
SFC	47	WAIT	Implement wait times	200 µs

Address Conversion

To assign the free address of a module to the corresponding rack and slot, you can use the following SFCs.

SFC	No.	Name	Description	Execution Time
SFC	5	GADR_LGC	Query the logical address of a channel *x* of signal module in slot *y*.	–
SFC	49	LGC_GADR	Query the module slot and rack belonging to the logical address of a module.	140 µs
SFC	50	RD_LGADR	Query all logical addresses of a module.	190 µs

Diagnostic Functions

To read and write diagnostic information, you can use the following system functions.

SFC	No.	Name	Description	Execution Time
SFC	51	RDSYSST	Read information from the system status list	280 µs + 200 µs/data record
SFC	52	WR_USMSG	Write selectable diagnostic information to the diagnostic buffer	110 µs

Module Parameter Assignment Functions

To write and read parameters of a module, the C7 CPU provides the following system functions.

SFC	No.	Name	Description	Execution Time
SFC	55	WR_PARM	Write dynamic parameters to a module	1.6 ms
SFC	56	WR_DPARM	Write predefined dynamic parameters to a module	1.75 ms
SFC	57	PARM_MOD	Assign parameters to a module	2.2 ms
SFC	58	WR_REC	Write a module-specific data record	1.4 ms + 32 µs/byte
SFC	59	RD_REC	Read a module-specific data record	0.49 ms

A.2 IEC Functions

DATE_AND_TIME

For instructions with the data formats DATE, TIME_OF_DAY and DATE_AND_TIME, STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
3	D_TOD_DT	Combine the data formats DATE and TIME_OF_DAY (TOD) and convert to the data format DATE_AND_TIME.	approx. 680 µs
6	DT_DATE	Extract the data format DATE from the data format DATE_AND_TIME.	approx. 230 µs
7	DT_DAY	Extract the day of the week from the data format DATE_AND_TIME.	approx. 230 µs
8	DT_TOD	Extract the data format TIME_OF_DAY from the data format DATE_AND_TIME.	approx. 200 µs

Time Formats

To convert the time formats S5 Time and Time, STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
33	S5TI_TIM	Convert data format S5 TIME to data format TIME	approx. 80 µs
40	TIM_S5TI	Convert data format TIME to data format S5 TIME	approx. 160 µs

Times (Duration)

For instructions with times, STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
1	AD_DT_TM	Add a time (duration) in the TIME format to a time (point) in the DT format. The result is a time in the DT format.	0.75 ms
35	SB_DT_TM	Subtract a time (duration) in the TIME format from a time (point) in the DT format. The result is a new time in the DT format.	0.75 ms
34	SB_DT_DT	Subtract two times in the DT format. The result is a time (duration) in the TIME format.	0.7 ms

Compare DATE_AND_TIME

To compare the contents of variables in the data format DATE_AND_TIME, STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
9	EQ_DT	Compare the content of two variables in the DATE_AND_TIME format for equality.	190 µs
12	GE_DT	Compare the content of two variables in the DATE_AND_TIME format for greater than or equal.	190 µs
14	GT_DT	Compare the content of two variables in the DATE_AND_TIME format for greater than.	190 µs
18	LE_DT	Compare the content of two variables in the DATE_AND_TIME format for less than or equal.	190 µs
23	LT_DT	Compare the content of two variables in the DATE_AND_TIME format for less than.	190 µs
28	NE_DT	Compare the content of two variables in the DATE_AND_TIME format for unequality.	190 µs

Compare STRING

To compare the contents of variables in the data format STRING, STEP 7 provides the following IEC functions.

FC-No.	Name	Description	Execution Time
10	EQ_STRNG	Compare the content of two variables in the STRING format for equality.	150 µs + (n × 32)
13	GE_STRNG	Compare the content of two variables in the STRING format for greater than or equal.	150 µs + (n × 32)
15	GT_STRNG	Compare the content of two variables in the STRING format for greater than.	140 µs + (n × 38)
19	LE_STRNG	Compare the content of two variables in the STRING format for less than or equal.	150 µs + (n × 32)
24	LT_STRNG	Compare the content of two variables in the STRING format for less than.	140 µs + (n × 38)
29	NE_STRNG	Compare the content of two variables in the STRING format for unequality.	150 µs + (n × 32)

n = number of characters

Processing STRING Variables

For instructions affecting the contents of STRING variables, STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
21	LEN	Read the current length of a STRING variable.	90 µs
20	LEFT	Read the first L characters of a STRING variable.	150 µs + (L × 26)
32	RIGHT	Read the last L characters of a STRING variable.	150 µs + (L × 26)
26	MID	Read the middle L characters of a STRING variable. (from the specified character).	150 µs + (L × 26)
2	CONCAT	Combine two STRING variables to one STRING variable.	180 µs + (n × 28)
17	INSERT	Insert one STRING variable in another STRING variable at a specified point.	250 µs + (n × 26)
4	DELETE	Delete L characters of a STRING variable.	300 µs + ((L + P) × 27)
31	REPLACE	Replace L characters of a STRING variable with a second STRING variable.	300 µs + ((L + P) × 27)
11	FIND	Specify the position of the second STRING variable within the first STRING variable.	k × 50 µs

L, P = block parameter (if l + P = 0, then execution time L + P = 254 µs
n = number of characters
k = number of characters in parameter IN1

Format Conversion with STRING

To convert variables to a STRING or from a STRING, STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
16	I_STRNG	Convert a variable in the INTEGER format to the STRING format.	1.11 ms
5	DI_STRNG	Convert a variable in the INTEGER (32-bit) format to the STRING format.	1.5 ms
30	R_STRNG	Convert a variable in the REAL format to the STRING format.	1.72 ms
38	STRNG_I	Convert a variable in the in the STRING format to the INTEGER format.	0.5 ms
37	STRNG_DI	Convert a variable in the in the STRING format to the INTEGER (32-bit) format.	0.84 ms
39	STRNG_R	Convert a variable in the in the STRING format to the REAL format.	2.0 ms

Processing Numeric Values

As selection functions STEP 7 provides the following IEC functions.

FC No.	Name	Description	Execution Time
22	LIMIT	Limit a numeric value to selectable limits.	0.45 ms
25	MAX	Select the highest of three numeric variable values.	0.43 ms
27	MIN	Select the lowest of three numeric variable values.	0.43 ms
36	SEL	Select one of two variable values.	0.32 ms

System Status List on the C7 CPU and AS-i PICS

B

Chapter Overview

B.1 System Status List

Definition

The system status list contains data describing the current status of a C7 CPU. This provides you at any time with an overview of:

- The current parameter assignment of the C7 CPU and configurable signal modules
- The current statuses and sequences on the C7 CPU and configurable signal modules.

For a detailed description of the structure of the system status list and all possible entries, refer to the reference manual *STEP 7 Standard and System Functions*.

Reading the System Status List

You can read the entries in the system status list with SFC51 “RDSYSST” from within the user program (see reference manual **/235/**).

Partial Lists

The system status list is divided into partial lists. This allows querying of specific information from the system status list.

Structure of the Partial Lists

Each partial list contains the following:

- Header information 4 data words long
- A certain number of data records containing event information.

Header Information

The header information of a partial list is 4 data words long. Figure B-1 shows the content of the header information of a partial list.

SZL-ID (identification of the partial list)
Index (specifies an object type/object number)
Length of a data record contained in the partial list
Number of data records in the partial list

Figure B-1 Header Information of a Partial List of the System Status List

SZL-ID

Each partial list has an identifier, the “SZL-ID”. It is also possible to read only an extract from the partial list. The ID of this extract from the partial list is also contained in the “SZL-ID”. Figure B-2 shows the structure of the SZL-ID” for the CPUs.

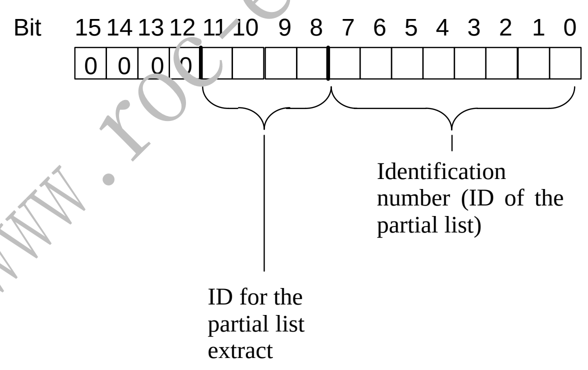


Figure B-2 structure of the ID of the Partial List “SZL-ID”

ID for the Partial list Extract

With the ID for the partial list extract, you can select the extent of the partial list to be output:

- **0_H**: the complete partial list is output
- **1_H to E_H**: a special partial list is output
- **F_H**: only the header information is output

Index

You must specify an index if you only want to read **one** particular data record from the partial list.

Length of the Following Data Records

This data word indicates how much information (in bytes) a data record of the partial list contains.

Number of Data Records

This data word indicates how many data records the transferred partial list contains.

List of Partial Lists

The table below shows the individual partial lists of the system status list with the entries relevant to the C7 CPU.

Table B-1 Partial Lists in the System Status List of the C7 CPU

SZL_ID	Partial List	Index (= ID of the individual data records of the partial list)	Data Record Contents (partial list extract)
	C7 CPU identification	–	C7 CPU type and version number
0011_H	all data records of the partial list		
0111_H	one data record of the partial list		
	C7 CPU characteristics		
0012_H	all data records of the partial list		
0112_H	only the data records of a group of characteristics	0000_H	STEP 7 processing
		0100_H	Time system on the C7 CPU
		0300_H	STEP 7 instruction set
0013_H	**Memory areas**	01_H	Work memory
		02_H	Integrated load memory
		05_H	Size of the backup memory

Table B-1 Partial Lists in the System Status List of the C7 CPU, continued

SZL_ID	Partial List	Index (= ID of the individual data records of the partial list)	Data Record Contents (partial list extract)
0014_H	**System areas**	0001_H	Process input image (size in bytes)
		0002_H	Process output image (size in bytes)
		0003_H	Number of memory bits
		0004_H	Number of timers
		0005_H	Number of counters
		0006_H	Size of the address area for I/Os
		0007_H	Total local data area of the C7 CPU (in bytes)
	Block types		
0015_H	all data records of the partial list		
0115_H	one data record selected with index	0800_H	OBs (number and size)
		$0A00_H$	DBs (number and size)
		$0B00_H$	SDBs (number and size)
		$0C00_H$	FCs (number and size)
		$0E00_H$	FBs (number and size)
0017_H 0117_H	**Permitted SDBs**	SDB number	-
	Rack information		
0018_H	all data records of the partial list		
0118_H	one data record selected with index	0000_H	Rack 0
		0001_H	Rack 1
		0002_H	Rack 2
		0003_H	Rack 3
	Assignment of interrupts/errors using the number of the corresponding OB	-	-
0021_H	data records of all possible interrupts		
$0A21_H$	data records all used interrupts		
	Interrupt status;		
0222_H	data record for the specified interrupt	0001_H	Interrupt class free cycle
		5050_H	Interrupt class asynchronous interrupts
	Priority class		
0023_H	Data records of all priority classes	0000_H	Priority of possible OBs
	only partial list header information		

Table B-1 Partial Lists in the System Status List of the C7 CPU, continued

SZL_ID	Partial List	Index (= ID of the individual data records of the partial list)	Data Record Contents (partial list extract)
	Operating mode of the C7 CPU		
0024_H	Information about all saved mode changes		
0124_H	Information about last mode change		
	Information about current mode		
0424_H	Information about specified mode		
0524_H		5000_H	Mode STOP
		5010_H	Mode STARTUP
		5020_H	Mode RUN
0131_H	**Communication capability parameters** for the specified communication type	0001_H	Number of connections, transm. rates
		0002_H	Test and startup parameters
		0003_H	Operator interface (parameters)
		0005_H	Diagnostic functions and diagnostic entries
		0007_H	Communication using global data (parameters)
		0008_H	Operator interface (time info)
0132_H	**Communication status information** for the specified communication type	0001_H	Number and type of connections
		0002_H	Number of test jobs
		0003_H	Number of current cyclic operator interface jobs
		0004_H	Protection levels of the C7 CPU
		0005_H	Diagnostic status data
		0007_H	Communication via global data
		0008_H	Cycle time, correction factor, run-time meter, date/time
		0009_H	Transmission rate set on MPI
$0D91_H$	**Module status information** for all modules in the rack		Properties/parameters of the inserted module
		0000_H	Rack 0
		0001_H	Rack 1
		0002_H	Rack 2
		0003_H	Rack 3

Table B-1 Partial Lists in the System Status List of the C7 CPU, continued

SZL_ID	Partial List	Index (= ID of the individual data records of the partial list)	Data Record Contents (partial list extract)
	Diagnostic buffer	x	Event information
$00A0_H$	all entered event information		The information depends on the event.
$01A0_H$	the x most recently entered event messages		
	Module Diagnostics	Rack + slot number	Module-dependent diagnostic information
$00B2_H$	complete module-dependent data record with module diagnostic information		

B.2 AS-Interface Protocol Implementation Conformance Statement (PICS)

Vendor	Siemens AG
Product Name	C7-621 ASi Control System
Order number	
Version	
Master Profile	M1 / M0
Date	

List of Available Master Functions with FC "ASI_3422"

The table explains the master functions of master class M1 on the host interface.

No.	Function or call on the host interface (symbolic representation)	M1	Note / Implementation of the function by/ Section
1	Image, Status = Read_IDI()	X	By controller access to the I/O interface of the ASi CP
2	Status = Write_ODI(Image)	X	By controller access to the I/O interface of the ASi CP
3	Status = Set_Permanent_Parameter(Addr, Param)	X	
4	Param, Status = Get_Permanent_Parameter(Addr)	X	
5	Status, GParam = Write_Parameter(Addr, Param)	X	
6	Status, Param = Read_Parameter(Addr)	X	
7	Status = Store_Actual_Parameters()	X	
8	Status = Set_Permanent_Configuration(Addr, Config)	X	
9	Status, Config = Get_Permanent_Configuration(Addr)	X	
10	Status = Store_Actual_Configuration()	X	With this command, there is also a complete restart on the CP.
11	Status, Config = Read_Actual_Configuration(Addr)	X	
12	Status = Set_LPS(List31)	X	
13	Status, List31 = Get_LPS()	X	Read lists and flags
14	Status, List31 = Get_LAS()	X	Read lists and flags

15	Status, List32 = Get_LDS()	X	Read lists and flags
16.0	Status = Get_Flags()	X	Read lists and flags
16.1	Status, Flag = Get_Flag_Config_OK()	X	DS0 error bit, Read lists and flags
16.2	Status, Flag = Get_Flag_LDS.0()	X	Read lists and flags
16.3	Status, Flag = Get_Flag_Auto_Address_Assign()	X	Read lists and flags
16.4	Status, Flag = Get_Flag_Auto_Prog_Available()	X	Read lists and flags
16.5	Status, Flag = Get_Flag_Configuration_Active()	X	Read lists and flags
16.6	Status, Flag = Get_Flag_Normal_Operation_Active()	X	Read lists and flags
16.7	Status, Flag = Get_Flag_APF()	X	DS0 error bit, Read lists and flags
16.8	Status, Flag = Get_Flag_Offline_Ready()	X	Read lists and flags
17	Status = Set_Operation_Mode(Mode)	X	
18	Status = Set_Offline_Mode(Mode)	X	
19	Status = Activate_Data_Exchange(Mode)	–	not implemented
20	Status = Change_Slave_Address(Addr1, Addr2)	X	
21	Status = Set_Auto_Address_Enable	X	
22	Status = Get_Auto_Address_Enable	X	Read lists and flags
23.1	Status, Resp = Cmd_Reset_ASi_Slave(Addr, RESET)	–	not implemented
23.2	Status, Resp = Cmd_Read_IO_Configuration(Addr, CONF)	X	
23.3	Status, Resp = Cmd_Read_Identification_Code(Addr, IDCOD)	X	
23.4	Status, Resp = Cmd_Read_Status(Addr, STAT)	X	
23.5	Status, Resp = Cmd_Read_Reset_Status(Addr, STATRES)	X	

List of Available Master Functions without FC "ASI_3422"

The table explains the master functions of master class M0 on the host interface.

No.	Function or call on the host interface (symbolic representation)	M0	Note / Implementation of the function by/ Section
1	Image, Status = Read_IDI()	X	By C7 CPU access to the I/O interface of the ASi CP
2	Status = Write_ODI(Image)	X	By C7 CPU access to the I/O interface of the ASi CP
3	Status = Set_Permanent_Parameter(Addr, Param)	–	not implemented
4	Param, Status = Get_Permanent_Parameter(Addr)	–	not implemented
5	Status, GParam = Write_Parameter(Addr, Param)	–	not implemented
6	Status, Param = Read_Parameter(Addr)	–	not implemented
7	Status = Store_Actual_Parameters()	–	not implemented
8	Status = Set_Permanent_Configuration(Addr, Config)	–	not implemented
9	Status, Config = Get_Permanent_Configuration(Addr)	–	not implemented
10	Status = Store_Actual_Configuration()	X	By menu to activate Config (Section 6.3.3)
11	Status, Config = Read_Actual_Configuration(Addr)	–	not implemented
12	Status = Set_LPS(List31)	–	not implemented
13	Status, List31 = Get_LPS()	–	not implemented
14	Status, List31 = Get_LAS()	–	not implemented
15	Status, List32 = Get_LDS()	–	not implemented
16.0	Status = Get_Flags()	–	not implemented
16.1	Status, Flag = Get_Flag_Config_OK()	X	By screen with status of the master (Section 6.3.7)
16.2	Status, Flag = Get_Flag_LDS.0()	–	not implemented
16.3	Status, Flag = Get_Flag_Auto_Address_Assign()	–	not implemented
16.4	Status, Flag = Get_Flag_Auto_Prog_Available()	–	not implemented
16.5	Status, Flag = Get_Flag_Configuration_Active()	–	not implemented
16.6	Status, Flag = Get_Flag_Normal_Operation_Active()	–	not implemented
16.7	Status, Flag = Get_Flag_APF()	X	By screen with status of the master (Section 6.3.7)
16.8	Status, Flag = Get_Flag_Offline_Ready()	–	not implemented

17	Status = Set_Operation_Mode(Mode)	X	By menu to activate Config (Section 6.3.3)
18	Status = Set_Offline_Mode(Mode)	–	not implemented
19	Status = Activate_Data_Exchange(Mode)	–	not implemented
20	Status = Change_Slave_Address(Addr1, Addr2)	–	implemented
21	Status = Set_Auto_Address_Enable	–	By menu with AUTOPROG selection (Section 6.3.3)
22	Status = Get_Auto_Address_Enable	–	not implemented
23.1	Status, Resp = Cmd_Reset_ASI_Slave(Addr, RESET)	–	not implemented
23.2	Status, Resp = Cmd_Read_IO_Configuration(Addr, CONF)	–	not implemented
23.3	Status, Resp = Cmd_Read_Identification_Code(Addr, IDCOD)	–	not implemented
23.4	Status, Resp = Cmd_Read_Status(Addr, STAT)	–	not implemented
23.5	Status, Resp = Cmd_Read_Reset_Status(Addr, STATRES)	–	not implemented

Key to Column 3

Char	Meaning
X	Function exists
–	Function does not exist

C7 OP Functionality / Standard Screens / System Messages

C

Chapter Overview

C.1 C7 OP Functionality

Table with Range of Functions

The following table provides an overview of the functions of the C7-621 and C7-621 AS-i.

Functions of the C7 OP	
Display – Contrast control	 with potentiometer
Event messages – Maximum number – Maximum length (characters) – Paging through pending messages	 499 40 50
Values entered in screens – Numbers or letters – Using symbolic variables	 x x
Actual value display (numeric and symbolic)	x
Combined actual value display/value input	x
Limit value check of operator input	x
Password protection – Password level – Passwords	 0–9 20
Screens – Maximum number – Entries per screen – Maximum number of fields per screen – Maximum number of fields per screen entry – Display	 40 20 300 32 x
Diagnostic functions (STATUS/FORCE VAR)	x
Configurable OP languages	German, English, French, Italian, Spanish
Online languages (available in user interface)	3
Communication via SIMATIC S7 – MPI	 x
Number of connectable controllers	4

C.2 Brief Description of the Standard Screens

Introduction

The following overview lists all the standard screens for the C7-621 and C7-621 AS-i. Along with a brief description of the function, the required password level is also shown. Under the column “1st Level” you can see the screens that can be selected in the basic screen. From these screens, you can branch further to the screens shown in the column “2nd Level”.

The hierarchy shown here relates to the sample configuration shipped with ProTool/Lite (see Section 7.3).

Table C-1 Overview of the Standard Screens

1st Level	2nd Level	Function	Password Level
Screens		• Display screen directory • Display screens	0
System	Mode	Set C7 OP mode: Online, Offline, Transfer	8
System	Languages	Select language	2
System	Dat/time	Set date and time	4
System	MPI Addr	Set address in the MPI network configuration	6
StatVAR		Display S7 addresses	0
ForceVAR		Display and modify S7 addresses	8
Password	Logout	User logout and return to the message level	0
Password	Edit	• Display password list • Assign and modify passwords and password levels • Delete passwords	9

C.3 System Messages

Introduction

This section lists the most important system messages, explains when they occur and how to eliminate the cause of the error.

Message Number

The system messages of C7 can be divided into various categories.

The message number indicates which category a system message belongs to:

Message number

□□□ Message text

0 Driver error
1 Startup message
2 Warning
3 Notification
4 Operator error
5 Other message
6 Configuration error
7 Internal error

Message Category

Using the message category, you can narrow down the cause of a system message.

Below, you will find a selection of the most important system messages with an explanation of when they occur and how you might eliminate them. Self explanatory system messages have been omitted.

Note

If the C7 has not had configuration data loaded on it, messages are displayed in English.

How to Handle "Internal Errors"

If you receive system messages that relate to "Internal Errors", please follow the steps outlined below:

1. Turn off the C7 and restart it.
2. During startup, change the C7 to the transfer mode (see Section 2.1), transfer the configuration again and start up the C7 again.
3. If the error is repeated, please contact your local Siemens office. Note down the error number that has occurred and any variables contained in the message.

Messages

The tables list the most important messages, their causes and possible remedies:

Message	Cause	Remedy
Please wait	A mode change is taking place	
Ready for transfer	Waiting for data from the programming device/PC	
Data transfer	Data transfer active between the programming device/PC and OP	
Firmware not compatible	The firmware cannot be used for the current configuration.	
EPROM memory failure	Memory chip defective Internal hardware error	Send in the device for repair indicating the error that has occurred.
RAM memory failure		
Flash memory failure	Memory chip defective or transfer error	Retransfer the configuration or send in the device for repair.

Message	Cause	Remedy
$ 005	Internal error	
$ 006	Internal error in data transfer during the transfer mode (message with 1 variable) 1 Internal error 4 The connection to ProTool/Lite broke down 5 Flash error (when writing) 6 Flash is full (configuration too large) 7 Flash error (when deleting) 8 Bad object number 9 Bad object length 10 Bad field number 11 Bad field length 12 Undefined job 13 Unexpected job 14 Unexpected mail type	Check the connection, transfer again
$ 040	Controller not responding Cable defective or not plugged in	Check the physical connection
$ 041	Temporary driver error	– Reboot PC – Retransfer the configuration
$ 043	Transfer error remote driver	
$ 044	Transfer error MPI	
$ 045	No connection to PLC number x (message with 1 variable)	

Message	Cause	Remedy
$ 100	Invalid RAM content	
$ 104	Transfer mode was aborted by keystroke	
$ 119	Automatic startup on the C7 (password list is not automatically deleted)	

$ 202	Error reading date	Re-enter date (send in C7)
$ 203	Error reading the time	Re-enter the time (send in C7)
$ 204	Error reading the day of the week	Re-enter day of week (send in C7)
$ 224	Event message buffer full; buffer was partly deleted and a printout was started	

$ 311	Memory bit x does not exist in controller	Change configuration (variable)
$ 316 $ 317	Current password level too low for required operation	Log in with a higher password level
$ 318	Log in attempted with invalid password	
$ 319	An existing password was entered when editing passwords	
$ 320 $ 321		First enter the password and then specify the level
$ 322	Password too short	Enter a password with at least three digits
$ 324	Selected screen or entry number does not exist.	
$ 340	Operator input to the C7 is not possible when the status function is active on the programming device	

Message	Cause	Remedy
$ 401	Selected value does not match the representation format	
$ 402	Operator error in the STATUS VAR or FORCE VAR screen; (after pressing INS when the 10th variable line is already completed)	
$ 403	Bad time input	
$ 404	Bad date input	
$ 409	Lower input limit value violated	Enter a value greater than or equal to *Var*
$ 410	Upper input limit value violated	Enter a value less than or equal to *Var*

$ 500 $ 501 $ 502 $ 503 $ 504	Transfer to the C7 CPU not currently possible C7 CPU lack of resources Standard FB not called for longer than 1.5 seconds	Check user program
$ 520	Too many return jumps saved	Branch to the message level (if necessary with the ESC key)
$ 522	Screen cannot be selected, not enough memory. Leads to a restart with memory optimization.	1. Delete unused field from the configuration 2. Configure a smaller screen (with less fields) or divide into more than one screen
$ 541	Peripheral I/O x does not exist	
$ 542	Input x does not exist	
$ 543	Output x does not exist	
$ 544	Memory bit x does not exist	
$ 545 $ 546	DB number x does not exist	
$ 549	Counter x does not exist	
$ 550	Timer x does not exist	

Message	Cause	Remedy
$ 600	Incorrect parameter transferred in the transfer mode (overflow warning)	Set the required value using a standard screen or by the controller
$ 601	Incorrect parameter transferred in the transfer mode (message log)	Set the required value in a standard screen or by the controller
$ 604	No message is configured for a set message bit	Configure and transfer messages
$ 606 $ 607 $ 609 $ 610 $ 611	Incorrect configuration	See internal error
$ 613	Data block does not exist or too short	Create a data block with the required length on the PLC
$ 616 $ 617	Incorrect configuration	See internal error
$ 619	Error in transfer mode (data structure for default setpoints)	Restart the transfer mode, Retransfer the configuration
$ 620	Incorrect parameter transferred in the transfer mode (function keys)	Retransfer the configuration
$ 621	Incorrect parameter transferred in the transfer mode (message type)	Set the required value in a standard screen or by the controller
$ 623		See internal error
$ 627	Incorrect configuration	See internal error
$ 631	(Message with 1 variable) 5, 6 Triggered event message not configured 25 Illegal field type 60 Event message range has polling time 0 8...20 Internal error	Complete the configuration and transfer it again
$ 632	(Message with 1 variable) 12 Screen does not contain entries 3, 6, 7, 8, 11, 13 Internal error	Complete the configuration and transfer it again
$ 634	(Message with 1 variable) 18 Screen title not configured 0 ... 8, 34 Internal error	Complete the configuration and transfer it again

$ 635	(Message with 1 variable)**x**. 6 Message or entry text not configured for current language 18 Screen title not configured 25 Illegal data format for symbolic field 33 Illegal data format for setpoint 48 Too many fields in the process screen 50 Variable for softkeys does not exist 55 Softkey specified in the entry does not exist 60 Loadable symbol record longer than 8 characters 61 Configured field length too short 63 Configured display format illegal 64 Configured data type illegal 7...9, 19, 28, 41...43 Internal errors	Complete the configuration or modify it and transfer it again
$ 636 $ 637	Triggered event message (number x) not configured	Complete the configuration and transfer it again
$ 645 $ 649	Internal errors	
$ 650	Area pointer for function used not configured	Configure area pointer
$ 651	Internal error	
$ 668	MPI configuration error	

Message	Cause	Remedy
$ 702	**Internal error** (actual value error)	
$ 703	**Internal error** (job incorrect)	
$ 704	Flash full	Reduce size of configuration
$ 705	**Internal error** (unknown message acknowledged)	
$ 7xx	**Internal error**	

C.3.1 Internal Errors

The error numbers from 700 onwards and some of the errors listed in the previous sections describe internal errors of the C7 units or of the ProTool configuration tool.

Procedure

If an internal error occurs, follow the steps outlined below:

- Change the C7 CPU to the *STOP*-mode. Turn off the C7 and then start it up again.
- During startup change the C7 OP to the transfer mode. Transfer the configuration again and restart the C7.
- If the error persists, please contact your local Siemens office. Note down the error number and any variables that may be contained in the message.

Possible Messages

- **005** Error no.: #Var1, #Var2, #Var3, #Var4
- **6xx** Error in the configuration file
- **701** Internal actual value error
- **702** Job incorrect (bad job number or job parameter)
- **703** Flash full (restrict the size of the configuration)
- **704** Error in the controller
- **705** Acknowledgment of an unknown message
- **706** Request already active
- **7xx** Internal errors

References

D

Manuals

/70/ Manual: *S7-300 Programmable Controller,* Hardware and Installation

/71/ Reference Manual: *S7-300 and M7-300 Programmable Controllers,* Module Specifications

/72/ Instruction List: *S7-300 Programmable Controller*

/231/ User Manual: *Standard Software for S7 and M7,* STEP 7

/232/ Manual: *Statement List (STL) for S7-300 and S7-400,* Programming

/233/ Manual: *Ladder Logic (LAD) for S7-300 and S7-400,* Programming

/234/ Programming Manual: *System Software for S7-300 and S7-400,* Program Design

/235/ Reference Manual: *System Software for S7-300 and S7-400,* System and Standard Functions

/236/ Manual: *Function Block Diagram (FBD) for S7-300 and S7-400,* Programming

/280/ Programming Manual: *System Software for M7-300 and M7-400,* Program Design

References for AS-i

/1/ AS-Interface. Das Aktuator-Sensor-Interface für die Automation

Werner Kriesel, O.W. Madelung, Carl Hanser Verlag München Wien 1994

/2/ AS-Interface Complete Specification

can be ordered from the AS-International Association e.V.

Address: AS-International Association e.V.
Geschäftsführung: Dr. Otto W. Madelung
Auf den Broich 4A
D - 51519 Odenthal
Germany

Tel.: +49 - 2174 - 40756
Fax.: +49 - 2174 - 41571

(AS-i technology is promoted by the AS-International Association e.V.)

/3/ SIMATIC NET Industrial Communications Networks

Catalog IK 10 1997

The catalog can be ordered from any SIEMENS office.

Siemens Worldwide

E

Overview

This appendix contains the following list:

- The towns and cities in the Federal Republic of Germany with Siemens sales offices
- All European and non-European Siemens subsidiaries and distributors

Who to contact about SIMATIC in Germany

ZN Zweigniederlassung (Regional Office)

ZN 52066 Aachen
AUT P 13, Hr. Georgens
Kurbrunnenstr. 22
☎ (02 41) 4 51-2 52
Fax (02 41) 4 51-3 98

ZN 86159 Augsburg
AUT S11, Hr. Hirth
Werner-von-Siemens Str. 6
☎ (08 21) 25 95-4 50
Fax (08 21) 25 95-4 08

ZN 95448 Bayreuth
AUT P/S 11, Fr. Hösl
Weiherstr. 25
☎ (09 21) 2 81-3 41
Fax (09 21) 2 81-4 44

ZN 10587 Berlin
AUT P 1, Hr. Liebner
Salzufer 6-8
☎ (0 30) 39 93-23 97
Fax (0 30) 39 93-23 02

ZN 33605 Bielefeld
AUT P 12, Fr. Schlüpmann
Schweriner Str. 1
☎ (05 21) 2 91-5 21
Fax (05 21) 2 91-5 90

ZN 38126 Braunschweig
AUT P 11, Hr. Pelka
Ackerstr. 20
☎ (05 31) 27 12-3 05
Fax (05 31) 27 12-4 16

ZN 28195 Bremen
AUT P 12, Fr. Ulbrich
Contrescarpe 72
☎ (04 21) 3 64-24 27
Fax (04 21) 3 64-28 42

ZN 09114 Chemnitz
AUT P 11, Fr. Aurich
Bornaer Str. 205
☎ (03 71) 4 75-35 10
Fax (03 71) 4 75-35 25

ZN 01189 Dresden
AUT 1, Hr. Lehmann
Karlsruher Str. 111
☎ (03 51) 40 22-2 77
Fax (03 51) 40 22-2 74

ZN 40219 Düsseldorf
AUT P 15, Hr. Becker
Lahnweg 10
☎ (02 11) 3 99-16 64
Fax (02 11) 3 99-18 48

ZN 99097 Erfurt
AUT P 22, Hr. Skudelny
Haarbergstr. 47
☎ (03 61) 4 25-23 51
Fax (03 61) 4 25-23 50

ZN 45128 Essen
AUT P 14, Hr. Klein
Kruppstr. 16
☎ (02 01) 8 16-24 28
Fax (02 01) 8 16-23 31

ZN 60329 Frankfurt
AUT P 25, Hr. W. Müller
Rödelheimer Landstr. 1-3
☎ (0 69) 7 97-34 18
Fax (0 69) 7 97-34 42

ZN 79104 Freiburg
AUT P, Hr. Thoma
Habsburgerstr. 132
☎ (07 61) 27 12-2 38
Fax (07 61) 27 12-4 46

ZN 20099 Hamburg
AUT 1, Hr. Rohde
Lindenplatz 2
☎ (0 40) 28 89-30 03
Fax (0 40) 28 89-32 09

ZN 30519 Laatzen (Hannover)
AUT P 10, Fr. Hoffmann
Hildesheimer Str. 7
☎ (05 11) 8 77-23 19
Fax (05 11) 8 77-27 39

ZN 74076 Heilbronn
AUT P/S, Hr. Gaul
Neckarsulmer Str. 59
☎ (0 71 31) 1 83-2 03
Fax (0 71 31) 1 83-3 20

ZN 76185 Karlsruhe
AUT 14 P, Hr. Boltz
Bannwaldallee 48
☎ (07 21) 9 92-24 13
Fax (07 21) 9 92-25 85

ZN 34117 Kassel
AUT P 13, Hr. Uhlig
Bürgermeister-Brunner-Str.15
☎ (05 61) 78 86-3 32
Fax (05 61) 78 86-4 48

ZN 87439 Kempten
AUT P, Hr. Fink
Lindauer Str. 112
☎ (08 31) 58 18-2 25
Fax (08 31) 58 18-2 40

ZN 24109 Kiel
AUT 1, Fr. Drews
Wittland 2-4
☎ (04 31) 58 60-3 26
Fax (04 31) 58 60-2 48

ZN 56068 Koblenz
AUT P 11, Hr. Ricke
Frankenstr. 21
☎ (02 61) 1 32-2 44
Fax (02 61) 1 32-2 55

ZN 50823 Köln
AUT P 14, Hr. Prescher
Franz-Geuer-Str. 10
☎ (02 21) 5 76-27 62
Fax (02 21) 5 76-27 95

ZN 78416 Konstanz
AUT P, Fr. Wiest
Fritz-Arnold-Str. 16
☎ (075 31) 988-2 02
Fax (075 31) 988-1 40

ZN 04105 Leipzig
AUT P 2, Fr. Kiesewetter
Springerstr. 15
☎ (03 41) 2 10-30 07
Fax (03 41) 2 10-30 63

ZN 39106 Magdeburg
AUT VG 33, Hr. Ganschinietz
Sieverstorstr. 32-33
☎ (03 91) 5 88-17 21
Fax (03 91) 5 88-17 22

ZN 68165 Mannheim
AUT 16 P, Hr. Sulzbacher
Dynamostr. 4
☎ (06 21) 4 56-28 43
Fax (06 21) 4 56-25 45

ZN 81679 München
AUT P 14, Hr. Schäfer
Richard-Strauss-Str. 76
☎ (0 89) 92 21-30 64
Fax (0 89) 92 21-43 99

ZN 48153 Münster
AUT S 13, Hr. Schlieckmann
Siemensstr. 55
☎ (02 51) 76 05-4 25
Fax (02 51) 76 05-3 36

ZN 90439 Nürnberg
AUT P 11, Hr. Glas
Von-der-Tann-Str. 30
☎ (09 11) 6 54-35 87
Fax (09 11) 6 54-73 84

ZN 49090 Osnabrück
AUT S 13, Hr. Pöhler
Eversburger Str. 32
☎ (05 41) 12 13-2 73
Fax (05 41) 12 13-3 50

ZN 93053 Regensburg
AUT P/S 12, Hr. Rewitzer
Hornstr. 10
☎ (09 41) 40 07-1 97
Fax (09 41) 40 07-2 36

ZN 18069 Rostock
AUT, Fr. Langhammer
Industriestr. 15
☎ (03 81) 78-21 71
Fax (03 81) 78-21 75

ZN 66111 Saarbrücken
AUT, Hr. Müller
Martin-Luther-Str. 25
☎ (06 81) 3 86-22 89
Fax (06 81) 3 86-21 11

ZN 57072 Siegen
AUT P 11, Hr. Patz
Sandstr. 42-48
☎ (02 71) 23 02-2 40
Fax (02 71) 23 02-2 38

ZN 70499 Stuttgart
AUT P 11, Hr. Müller
Weissacherstr. 11
☎ (07 11) 1 37-26 44
Fax (07 11) 1 37-29 46

ZN 54292 Trier
AUT VG 14 P, Hr. Baldauf
Löbstr. 15
☎ (06 51) 20 09-23
Fax (06 51) 20 09-24

ZN 89079 Ulm
AUT ZR, Hr. Birk
Nikolaus-Otto-Str. 4
☎ (07 31) 94 50-3 28
Fax (07 31) 94 50-3 34

ZN 97084 Würzburg
AUT PIS 13, Hr. Vogt
Andreas-Grieser-Str. 30
☎ (09 31) 61 01-4 59
Fax (09 31) 61 01-5 42

ZN 42103 Wuppertal
siehe ZN 45128 Essen
AUT P 14, Hr. Klein
Kruppstr. 16
☎ (02 01) 8 16-24 28
Fax (02 01) 8 16-23 31

Who to contact about SIMATIC in Europe (except Germany)

Austria
6901 Bregenz
Siemens AG, AUT,
Mr Madlener Jun.,
Josef-Huter-Straße 6,
P.O. Box 347
☎ 00 43 (55 74) 41 92 72
Fax 00 43 (55 74) 41 92 88

8054 Graz
Siemens AG, AUT
Mr Jammernegg
Strassganger Straße 315
P.O. Box 39
☎ 00 43 (3 16) 2 80 42 80
Fax 00 43 (3 16) 2 80 42 85

6040 Innsbruck/Neu-Rum
Siemens AG, AUT
Mr Mayr
Siemensstraße 24, P.O. Box 9 04
☎ 00 43 (5 12) 23 12 60
Fax 00 43 (5 12) 23 15 30

9020 Klagenfurt
Siemens AG, AUT
Mr Weber
Werner von Siemens Park 1
☎ 00 43 (4 63) 3 88 32 43
Fax 00 43 (4 63) 3 88 34 49

4020 Linz
Siemens AG, AUT, Mr Schmidt
Wolfgang-Pauli-Straße 2, P.O. Box 563
☎ 00 43 (7 32) 3 33 02 95
Fax 00 43 (7 32) 3 33 04 93

5020 Salzburg
Siemens AG, AUT, Mr Mariacher
Innsbrucker Bundesstraße 35
P.O. Box 3
☎ 00 43 (6 62) 4 48 83 35
Fax 00 43 (6 62) 4 48 83 09

1211 Wien
Siemens AG, AUT 1
Mr Strasser
Siemensstraße 88-92,
P.O. Box 83
☎ 00 43 (1) 25 01 37 88
Fax 00 43 (1) 25 01 39 40

Belgium
1060 Brussels
Siemens S.A., VP4, Mr Gmuer
Chaussee de Chaleroi 116
☎ 00 32 (2) 5 36 25 33
Fax 00 32 (2) 5 36 23 87

Bulgaria
1113 Sofia
Siemens AG, Mrs Kirova
Blvd. Dragan Zankov Nr. 36
☎ 0 03 59 (2) 70 85 21
Fax 0 03 59 (2) 68 50 51

Croatia
41000 Zagreb
Siemens d.o.o., Mr Culjak
Trg Drazena Petrovica 3
("Cibona")
☎ 0 03 85 (41) 33 88 95
Fax 0 03 85 (41) 32 66 95

Czech Republic
60200 Brno
Siemens AG, Kancelar Brno,
Hr. Tucek, Vinarska 6
☎ 00 42 (5) 43 21 17 49
Fax 00 42 (5) 43 21 19 86

14000 Praha 4
Siemens AG, Zastoupeni v CR,
Mr Skop, Na strzi 40
☎ 00 42 (2) 61 21 50 33 6
Fax 00 42 (2) 61 21 51 46

Denmark
2750 Ballerup
Siemens A/S, IP, Mr Hansen
Borupvang 3
☎ 00 45 (44) 77 42 90
Fax 00 45 (44)77 40 16

Finland
02601 Espoo
Siemens Osakeyhtioe,
OEM/AUT 1, Mr Saarelainen
Majurinkatu, P.O. Box 60
☎ 0 03 58 (0) 51 05 36 70
Fax 0 03 58 (0) 51 05 36 56

France
69300 Caluire
Siemens S.A., AUT 1
9-11, chemin des petites brosses, BP 39
☎ 00 33/ 78 98 60 08
Fax 00 33/ 78 98 60 18

59812 Lesquin, Cedex/Lille
Siemens S.A., AUT 1
78, rue de Gustave Delroy BP 239
☎ 00 33/ 20 95 71 91
Fax 00 33/ 20 95 71 86

33694 Merignac/Bordeaux
Siemens S.A., AUT 1
Parc Cadera Sud
36, Avenue Ariane, BP 351
☎ 00 33/ 56 13 32 66
Fax 00 33/ 56 55 99 59

44300 Nantes
Siemens S.A., AUT 1
Zac du Perray
9, rue du petit chatelier
☎ 00 33/ 40 18 68 30
Fax 00 33/ 40 93 04 83

93527 Saint Denis, Cedex 2 Paris
Siemens S.A., AUT 1, Mr Granger
39/47, Bd Ornano
☎ 00 33 (1) 49 22 33 18
Fax 00 33 (1) 49 22 32 05

67016 Strasbourg, Cedex
Siemens S.A., AUT 1
2, rue du Rhin-Napoleon BP 48
☎ 00 33/ 88 45 98 22
Fax 00 33/ 88 60 08 40

31106 Toulouse
Siemens S.A., AUT 1, Mr Huguet
ZAC de Basso Cambo
avenue du mirail, BP 1304
☎ 00 33/ 62 11 20 15
Fax 00 33/ 61 43 02 20

Great Britain
Manchester M20 2UR
Siemens PLC, Control Systems,
Mr Hardern
Sir William Siemens House,
Princess Road
☎ 00 44 (61) 4 46 52 33
Fax 00 44 (61) 4 46 52 32

Greece
54110 Thessaloniki
Siemens A.E., VB 3 AUT
Mr Passalidis
Georgikis Scholis 89, P.O. Box 10290
☎ 00 30 (31) 47 92 12
Fax 00 30 (31) 47 92 65

15110 Amaroussio/Athen
Siemens A.E., HB 3 AUT
Mr Antoniou; Paradissou & Artemidos, P.O. Box 6 10 11
☎ 00 30 (1) 68 64-5 15
Fax 00 30 (1) 68 64-5 56

Hungary
1036 Budapest
Siemens GmbH, AUT 1
Mr Turi, Lajos utca 103
☎ 00 36 (1) 2 69 74 55
Fax 00 36 (1) 2 69 74 54

Iceland
121 Reykjavik
Smith & Norland H/F,
Mr Kjartansson,
Noatuni 4, P.O. Box 519
☎ 0 03 54 (1) 62 83 00
Fax 0 03 54 (1) 62 83 40

Ireland
Dublin 11
Siemens Ltd., Power & Automation Division,Mr Mulligan
8-11 Slaney Road
Dublin Industrial Estate
☎ 0 03 53 (1) 8 30 28 55
Fax 0 03 53 (1) 8 30 31 51

Italy
40127 Bologna
Siemens S.p.A., AUT R10A
Mr Tosatti, Via Casciarolo, 8
☎ 00 39 (51) 6 38 45 09
Fax 00 39 (51) 24 32 13

25128 Brescia
Siemens S.p.A., AUT R10A
Mr Gaspari
Via della Volta, 92
☎ 00 39 (30) 3 53 05 26
Fax 00 39 (30) 34 66 20

20124 Milano
Siemens S.p.A., AUT R10A
Mr Berti
Via Lazzaroni, 3
☎ 00 39 (2) 66 76 28 36
Fax 00 39 (2) 66 76 28 20

35129 Padova
Siemens S.p.A., AUT R10A
Mr Millevoi
Viale dell'Industria, 19
☎ 00 39 (49) 8 29 13 11
Fax 00 39 (49) 8 07 00 09

00142 Roma
Siemens S.p.A., AUT R10A
Mr Vessio
Via Laurentina, 455
☎ 00 39 (6) 5 00 95-1
Fax 00 39 (6) 5 00 95 20

10127 Torino
Siemens S.p.A., AUT R10A
Mr Montoli, Via Pio VII, 127
☎ 00 39 (11) 6 17 3-1
Fax 00 39 (11) 61 61 35

Luxembourg
1017 Luxemburg-Hamm
Siemens S.A., AUT, Mr Nockels
20, Rue des Peupliers, B.P. 1701
☎ 0 03 52/ 4 38 43-4 21
Fax 0 03 52/ 4 38 43-4 15

Netherlands
S2595 AL Den Haag
Siemens Nederland N.V. IPS/APS,
Mr Penris
Prinses Beatrixlaan 26
☎ 00 31 (70) 3 33 32 74
Fax 00 31 (70) 3 33 34 96

Norway
5033 Fyllingsdalen
Siemens A/S Bergen
Mr Troan, Bratsbergveien 5
P.O. Box 36 60
☎ 00 47 (55) 17 67 41
Fax 00 47 (55) 16 44 70

0518 Oslo 5
Siemens A/S, AUT Produkter
Mr Eggen, Ostre Aker vei 90
Postboks 10, Veitvet
☎ 00 47 (22) 63 34 09
Fax 00 47 (22) 63 33 90

7004 Trondheim
Siemens A/S Trondheim,
Mr Thorsen, Spelaugen 22
☎ 00 47 (73) 95 96 69
Fax 00 47 (73) 95 95 04

Poland
40-931 Katowice
Siemens Sp. z.o.o.
Katowice Branch Office, Mr Krzak
Ul. Kosciuszki 30
☎ 00 48 (3) 157 32 66
Fax 00 48 (3) 157 30 75

60-815 Poznan
Siemens Sp. z.o.o.
Poznan Branch Office
Mr Weiss, Ul. Gajowa 6
☎ 00 48 (61) 47 08 86
Fax 00 48 (61) 47 08 89

03-821 Warszawa
Siemens Sp. z.o.o.
Mr Cieslak
Ul. zupnicza 11
☎ 00 48 (2) 6 70 91 47
Fax 00 48 (2) 6 70 91 49

53-332 Wroclaw
Siemens Sp. z.o.o.
Wroclaw Branch Office, Mr Wojniak
Ul. Powstanców Slaskich 95
☎ 00 48 (71) 60 59 97
Fax 00 48 (71) 60 55 88

Portugal
2700 Amadora
Siemens S.A., Dep. Energia e Industria, Mr C. Pelicano, Eng.
Estrada Nacional 117 ao km 2,6
Alfragide, Apartado 60300
☎ 0 03 51 (1) 4 17 85 03
Fax 0 03 51 (1) 4 17 80 71

4450 Matosinhos-Porto
Siemens S.A., Dep. Energia e Industria, Mr A. Amaral, Eng.
Estrada Nacional 107,
No. 3570 Freixieiro, Apartado 5145
☎ 0 03 51 (2) 9 99 21 11
Fax 0 03 51 (2) 9 99 20 01

Romania
76640 Bucuresti
Siemens, Birou de consultatii tehnice, Mr Fritsch
Str. Zarii No. 12, sector 5
☎ 00 40 (1) 2 23 47 95
Fax 00 40 (1) 2 23 45 69

Russia
113043 Moskau
Siemens AG, Mr Engelhard/
Mr Michailow, Ul. Dubininskaja 98
☎ 0 07 (0 95) 2 36 75 00
Fax 0 07 (0 95) 2 36 62 00

Slovakia
81261 Bratislava, Siemens AG
Mr Sykorcin,
Tovarenska 11
☎ 00 42 (7) 31 21 74
Fax 00 42 (7) 31 63 32

Slovenia
61000 Ljubljana
Siemens Slovenija
Mr Lavric
Dunajska C47
☎ 0 03 86 (61) 1 32 60 68
Fax 0 03 86 (61) 1 32 42 81

Spain
48011 Bilbao
Siemens S.A., AUT 1
Mr Tapia Maximo Aguirre, 18
☎ 00 34 (4) 4 27 64 33
Fax 00 34 (4) 4 27 82 39

08940 Cornella de Llobregat/ Barcelona
Siemens S.A., AUT 1, Mr Ortiz
Joan Fernandez Vallhonrat, 1
☎ 00 34 (3) 4 74 22 12
Fax 00 34 (3) 4 74 42 34

33206 Gijon
Siemens S.A., AUT 1
Mr Huchet
Corrida, 1
☎ 00 34 (85) 35 08 00
Fax 00 34 (85) 34 93 10

15005 La Coruna
Siemens S.A., AUT 1
Mr Pereira
Linares Rivas, 12-14
☎ 00 34 (81) 12 07 51
Fax 00 34 (81) 12 03 60

30008 Murcia
Siemens S.A., AUT 1
Mr Martinez
Marques de los Velez, 13
☎ 00 34 (68) 23 36 62
Fax 00 34 (68) 23 52 36

41092 Sevilla
Siemens S.A., AUT 1 ,Mr de la Fuente, ISLA DE LA CARTUJA
Paseo de la Acacias, s/n
(Edificio Siemens)
☎ 00 34 (5) 4 46 30 00
Fax 00 34 (5) 4 46 30 46

28760 Tres Cantos (Madrid)
Siemens S.A., AUT 1
Mr Olaguibel
Ronda de Europa, 5
☎ 00 34 (1) 8 03 12 00
Fax 00 34 (1) 8 03 22 71

46021 Valencia
Siemens S.A., AUT 1
Mr Albors
Avda. Aragon, 30 (Ed. Europa)
☎ 00 34 (6) 3 69 94 00
Fax 00 34 (6) 3 62 61 19

36204 Vigo
Siemens S.A., AUT 1, Mr Garrido Pizarro, 29
☎ 00 34 (86) 41 60 33
Fax 00 34 (86) 41 84 64

50012 Zaragoza
Siemens S.A., AUT 1, Mr Aliaga
Avda. Alcalde Gomez Laguna, 9
☎ 00 34 (76) 35 61 50
Fax 00 34 (76) 56 68 86

Sweden
40020 Göteborg
Siemens AB, ASP, Mr Ohlsson
Östergardsgatan 2-4
P.O. Box 1 41 53
☎ 00 46 (31) 7 76 86 53
Fax 00 46 (31) 7 76 86 76

55111 Jönköping
Siemens AB, ASP, Mr Jonsson
Klubbhusgatan 15
P.O. Box 10 07
☎ 00 46 (36) 15 29 00
Fax 00 46 (36) 16 51 91

20123 Malmö
Siemens AB, ASP
Mr Jämtgren
Grimsbygatan 24, P.O. Box 326
☎ 00 46 (40) 17 46 14
Fax 00 46 (40) 17 46 17

85122 Sundsvall
Siemens AB, ASP
Mr Sjöberg
Lagergatan 14, P.O. Box 766
☎ 00 46 (60) 18 56 00
Fax 00 46 (60) 61 93 44

19487 Upplands Väsby/Stockholm
Siemens AB, ASP-A1
Mr Persson
Johanneslandsvägen 12-14
☎ 00 46 (8) 7 28 14 64
Fax 00 46 (8) 7 28 18 00

Switzerland
1020 Renens/Lausanne
Siemens-Albis SA, Systemes d'automation, VHRL, Mrs Thevenaz
5, Av. des Baumettes, BP 1 53
☎ 00 41 (21) 6 31 83 09
Fax 00 41 (21) 6 31 84 48

8047 Zürich
Siemens-Albis AG, VHR 3,
Mr Engel, Freilagerstraße 28-40
☎ 00 41 (1) 4 95 58 82
Fax 00 41 (1) 4 95 31 85

Turkey
06680 Ankara-Kavaklidere
SIMKO-ANKARA, Mr Ensert,
Atatürk Bulvari No. 169/6
☎ 00 90 (312) 4 18 22 05

80040 Findikli-Istanbul
SIMKO TIC. ve SAN. A. S.,
AUT 1, Mrs Yargic
Meclisi Mebusan Cad. No 125
☎ 00 90 (212) 2 51 17 06
Fax 00 90 (212) 2 52 39 16

Ukraine
252054 Kiew 54
Siemens Representative, AUT,
Mr Liebschner,
Ul. Worowskowo 27
☎ 0 07 (044) 2 16 02 22
Fax 0 07 (044) 2 16 94 92

Who to contact about SIMATIC outside Europe

Africa

Algeria

16035 Hydra/Alger
Siemens, Bureau d'Alger, Division Energie, Mr Bennour,
44, rue Abri Areski, P.O. Box 112
☎ +213 (2) 60 40 88
Fax +213 (2) 60 65 98

Egypt

Zamalik/EGY-Cairo
ELETECH, AUT, Mr W. Y. Graiss
6 Zarkaria Rizk Street,
P.O. Box 90
☎ +20 (2) 3 42 03 71
Fax +20 (2) 3 42 03 76

Ivory Coast

Abidjan 15/R. C. I.
Siemens AG, SEMEN, Mr. Hellal,
16 P.O. Box 1062
☎ +2 25 (37) 46 57
Fax +2 25 (27) 10 21

Libya

Tripoli/Libya S.P.L.A.J.
Siemens AG, Branch Libya,
Mr Wahab, Zat-EL-Imad-
Building Tower No. 5, Floor No. 9
P.O. Box 91 531
☎ +218 (21) 4 15 34
Fax +218 (21) 4 79 40

Morocco

Casablanca 05
SETEL S.A., AUT, Mr El Bachiri,
Immeuble Siemens,
km 1, Route de Rabat,
Ain Sebaa
☎ +212 (2) 35 10 25
Fax +212 (2) 34 01 51

Namibia

Windhoek 9000
Siemens (Pty) Ltd., Mr Jürgen Hoff
9 Albert Wessels Street
Industries North, P.O.B. 23125
☎ +2 64 (61) 6 13 58/59
Fax +2 64 (61) 6 13 77

South Africa

RSA-2001 Braamfontein
Siemens Ltd., AUT, Mr E. Hillermann
Siemens House SH 401
Corner Wolmarans & Biccard
Streets, P.O. Box 4583
2000 Johannesburg
☎ +27 (11) 4 07 41 11
☎ +27 (11) 4 07 48 15
Fax +27 (11) 4 07 46 82

Tunisia

TN-2062 Romana-Le Bardo
FAZE Sarl Electrotechnique,
Mr Fantar, Immeuble Cham
☎ +2 16 (1) 51 90 91
Fax +2 16 (1) 50 49 32

TN-2035 Charguia II Tunis
SITELEC S.A. Mr Mouelhi
16, Rue de l'Usine
Zone industrielle (Aéroport),
BP 115, 1050 Tunis Cedex
☎ +2 16 (1) 70 00 99
Fax +2 16 (1) 71 70 10

Zimbabwe

Electro Technologies Corp. (Pvt.) Ltd./
Siemens Zimbabwe,
Mr Ron Claassens, Savoy House
cnr. Inez Terrace/J. Moyo Ave
P.O. Box 46 80
☎ +263 (4) 79 18 66
Fax +263 (4) 75 44 06

America

Argentina

8000 Bahia Blanca,
Prov. de Buenos Aires
Siemens S.A., Mr S.Duran,
Rudriguez 159
☎ +54 (91) 55-61 41
Fax +54 (91) 55-61 71

(1650) San Martin,
Prov. de Buenos Aires
Siemens S.A., PEI-AUT,
Mr Rudriguez Juis/Mr Roland Herron,
Gral, Roca 1865, Ruta 8, km 18 C.C.
☎ +54 (1) 7 38 71 92/7 15
☎ +54 (1) 7 38 71 85
Fax +54 (1) 7 38 71 71

5000 Cordoba, Prov. de Cordoba
Siemens S.A., Mr S. Garcia,
Campillo 70
☎ +54 (51) 73-9940/994
Fax +54 (51) 72-97 14

5539 Las Heras, Prov. de Mendoza
Siemens S.A., Mr S. Suarez,
Acceso Norte 379
☎ +54 (61) 30-00 22/0 37
Fax +54 (61) 30-00 22/0 37

2000 Rosario, Prov. de Santa Fe
Siemens S.A., Mr R. Stiza,
Ricchieri 750
☎ +54 (1) 41 37-03 21/0
Fax +54 (1) 41 37-07 87

Bolivia

La Paz
Sociedad Comercial e Industrial
Hansa Ltda., E & A, Mr Beckmann
Calle Mercado esq. Yanacocha
C. P. 10 800
☎ +591 (2) 35 44 45
Fax +591 (2) 37 03 97

Brazil

05110-900 Sao Paulo, SP, Pinituba
MAXITEC S.A., AUT-PA, Mr F. Rocco,
Avenida Mutinga, 3650
☎ +55 (11) 8 36 29 99
Fax +55 (11) 8 36 29 50

Canada

Mississauga, ON L5N 7A6
Siemens Electric Ltd., Dept. SL 20,
Mr Fred Leon, 2185 Derry Road
West
☎ +1 (905) 7 92 81 95 82
Fax +1 (905) 58 19 58 12

Point Claire, QUE H9R-4R6
Siemens Electric Ltd., Mr D. Goulet
7300 Trans Canada Highway
☎ +1 (514) 4 26 60 99
Fax +1 (514) 4 26 61 44

Burnaby, B. C. V5J 5J1
Siemens Electic Ltd., Hr. A. Mazurek
Marine Way Business Park
8875 Northbrook Court
☎ +1 (604) 4 35 08 80
Fax +1 (604) 4 35 10 23

Chile

Santiago de Chile
INGELSAC,Div. Energia, Hr. Browne
Avda. Holanda 64, Cas. 242-V
☎ +56 (2) 2 31 00 00
Fax +56 (2) 2 32 66 88

Costa Rica

San Jose 1000
Siemens S.A. San Jose, Division
Energia y Automatizacion, VAT,
Hr. Ferraro,
La Uruca, Apartado 100 22
☎ +5 06 87 50 50
Fax +5 06 21 50 50

Ecuador

Quito
Siemens S.A., Dept. DEA,
Mr J. Guerra
Calle Manuel Zambrano y
Panamericana Norte km 2 1/2
Casilla de Correos 17-01-3580
☎ +5 93 (2) 47 40 60
Fax +5 93 (2) 40 77 38

El Salvador

San Salvador
Siemens S.A., E/A, Mr M. Dubon
43, Calle Siemens
Parque Industrial Sta. Elena
Apartado 1525
☎ +5 03 78 33 33
Fax +5 03 78 33 34

Guatemala

Ciudad de Guatemala
Siemens S.A., EA/AUT, Mr Godoy
2a Calle 6-76
Zona 10, Apartado 1959
☎ +5 02 (2) 32 44 44
Fax +5 02 (2) 34 36 70

Columbia

Baranquilla
Siemens S.A., EA, Mr C. Perez,
Carrera 58 No. 709-40
☎ +57 (958) 56 11 48
Fax +57 (958) 56 11 48

Bogota 6
Siemens S.A., Division Energia,
Mr M. Jaramillo
Carrera 65, No. 11-83
Apartado 80150
☎ +57 (1) 2 94 22 66
Fax +57 (1) 2 94 24 98

Cali
Siemens S.A., Barranquilla,
Mr Guido Hernandez
Carrera 40, No. 13-05
☎ +57 (92) 66-4 44 00
Fax +57 (92)66-5 30 56

Cali
Siemens S.A. Cali Mr C. A. Naranjo
Carrera 48 A, 15 Sur 92
☎ +57 (94) 2 66-30 66
Fax +57 (94) 2 68-25 57

Mexico

02300 Mexico, D.F.
Siemens S.A. de C.V., EI-AUT,
Mr Gregorio Sanchez
Delegacion Azcapotzalco
Poniente 116, No. 590
Colonia Industrial Vallejo
Apartado Postal
15-064, 02600 mexico
☎ +52 (5) 3 28 20 00
Fax +52 (5) 3 28 21 92
Fax +52 (5) 3 28 21 93

Peru

Lima 13
ESIM S.A., Dept. AUT,
Mr Paz-Soldan
Avda, N. Arriola 385 4to Piso
☎ +51 (14) 71 46 61
Fax +51 (14) 71 09 93

USA

Alpharetta, GA 30202
SIA Inc., Regional Sales Manager
Southeast, Mr Mich Gunyon,
Technology Drive
☎ +1 (4 04) 7 40 36 60
Fax +1 (4 04) 7 40 36 96

Andover, MA 01810
SIA Inc., North East Region,
Mr Mark Fondl, One Tech Drive,
Suite 310
☎ +1 (5 08) 6 85 60 77
Fax +1 (5 08) 6 86 88 72

Houston, TX 77040
SIA Inc., SouthWest Region,
Mr Wade Bradford
13100 Northwest Freeway, Suite 210
☎ +1 (713) 6 90 03 33
Fax +1 (713) 4 60 44 50

Mason, OH 45040-9011
SIA Inc., Central Region,
Mr Luther Crouthamel,
4770 Duke Drive suite 381
☎ +1 (5 13) 3 98 96 91
Fax +1 (5 13) 3 98 98 39

Mukilteo, WA 98275
SIA Inc., MidwWest Region,
Mr Earl Haas, 8412 54th Avenue
West
☎ +1 (7 14) 9 79 66 00
Fax +1 (7 14) 5 57 90 61

Plymouth, MN 55442
SIA Inc., MidWest Region,
Mr Greg Jaster,
13235 45th Avenue No.
☎ +1 (7 08) 6 40 15 95
Fax +1 (7 08) 6 40 80 26

Venezuela

1071 Caracas
Siemens S.A., AUT-ASI,
Mr Jesus Cavada
Avda. Don Diego Cisneros
Urbanizacion Los Ruices,
Ap. 3616, Caracas 1010 A
☎ +58 (2) 2 39 07 33
Fax +58 (2) 2 03 82 00

Asia

China

510064 Guangzhou
Siemens Ltd. China, Guangzhou
Office, Mr Peter Chen,
Room 1134-1157 GARDEN Hotel
Garden Tower,
368 Huanshi Dong Lu
☎ +86 (20) 3 85 46 88
Fax +86 (20) 3 34 74 54

100015 Beijing
Siemens Ltd. China, Beijing Office,
Mr Wolfgang Söllner
7, Wangjing Zhonghuan Nan Lu
Chaoyang District
P.O. Box 8543
☎ +86 (10) 4 36 18 88
Fax +86 (10) 4 36 32 13

200090 Shanghai
Siemens Ltd. China, Shanghai
Office, Mr William Cui,
450, Lin Quing Lu
☎ +86 (21) 5 39 54 10
Fax +86 (21) 5 39 54 21

110001 Shenyang
Siemens Ltd. China, Shenyang
Office, Mr Ren Qi, Sakei Torch
Building 23rd Fl. 262A Shifu Da Lu
Shen He District
☎ +86 (24) 2 79 02 87
Fax +86 (24) 2 79 02 86

Who to contact about SIMATIC outside Europe

Hong Kong

Hong Kong
Siemens Ltd. Hang Kong A. R. O.,
Automation System ,
Division , Mr Keiren Lake,
7th Floor, Regency Centre,
39 Wong Chuk Hang Road
☎ +85 (2) 28 70 76 11
Fax +85 (2) 25 18 04 11

India

Bangalore 560 001
Siemens Ltd., BAN/AUT–MAP,
Mr B. Sunderram
Jyoti Mahal, 3rd Floor
49, St. Marks Road, P.O. Box 5212
☎ +91 (80) 2 21 21 01
Fax +91 (80) 2 21 24 18

Bombay 400 018
Siemens Ltd., AUT/M–AP,
Mr S. Mistry
Head Office B Building
130, Ganpat Jahav Marg. Worli
☎ +91 (22) 4 93 13 50/60
Fax +91 (22) 4 95 08 22

Calcutta 700 071
Siemens Ltd., CAL/AUT–MAP,
Mr D. K. Ganguli
6, Little Russel Street, P.O. Box 715
☎ +91 (33) 2 47 83 74/–80
Fax +91 (33) 2 47 47 83

New Delhi 110 002
Siemens Ltd., DEL/AUT–MAP,
Mr R. Narayanan
4A, Ring Road, I.P. Estate,
P.O. Box 7036
☎ +91 (11) 3 31 81 44
Fax +91 (11) 3 31 41 78

Indonesia

Jakarta 12870
Dian Graha Elektrika, Jakarta, Power
Eng. & Autom. Div., Mr M. Zafrullah
Jl. Gatot Subroto Kov. 74–75,
Mustika centre Building Floor 2a.,
P.O. Box 4267
☎ +62 (21) 8 30 65 74
Fax +62 (21) 8 30 74 02

Iran

15914 Teheran
Siemens S.S.K., Mr Din–Payuh
Khiabane Ayatollah Taleghani 32
Siemenshouse, P.O. B.ox15875–4773,
15 Teheran
☎ +98 (21) 61 41
Fax +98 (21) 6 40 23 89

Japan

Tokyo 141–00
Siemens K.K., ATT, Mr Nakamichi
Siemens Fujikara Building, 8F
11-20, Nishi-Gotanda 2-chome
Shinagawa-ku
☎ +81 (3) 54 90 44 37
Fax +81 (3) 54 95 97 92

Pakistan

Karachi - 74400
Siemens Pakistan Eng. Co. Ltd.,
Power Division, Mr Ilyas
ILACO House
Abdullah Haroon Road
P.O. Box 7158
☎ +92 (21) 51 60 61
Fax +92 (21) 5 68 46 79

Philippines

Metro Manila
Siemens Inc., Mr B. Bonifacio
2nd & 4th Fl., Sterling Centre Bldg.
Esteban cor. de la Rosa
Legaspi Village
☎ +63 (2) 8 18 48 18
Fax +63 (2) 8 18 48 22

Saudi Arabia

Jeddah - 21412
Arabia Electric Ltd. Service Center,
Mr Kobeissi, P.O. Box 4621
☎ +9 66 (2) 6 65 84 20
Fax +9 66 (2) 6 65 84 90

Singapore

Singapore 1334
Siemens (Pte) Ltd. Singapore, AUT,
Mr Ulf Bexell,
2 Kallang Sector
☎ +65 8 41 35 28
Fax +65 8 41 35 29

South Korea

Seoul
Siemens Ltd., E+A, Mr Kang W. S.
Asia Tower Building, 9th Floor
726 Yeoksam-dong, Kang–nam–ku,
C.P.O. Box 3001
☎ +82 (2) 5 27 77 62
Fax +82 (2) 5 27 77 19

Taiwan

Taipei 106
Siemens Ltd., AUT 1, Mr Gulden
6th Fl., Cathy Life Insurance Bldg.
296, Jen Ai Road, Sec. 4
☎ +886 (2) 3 25 48 88
Fax +886 (2) 7 05 49 75

Thailand

Bangkok 10110
Berli Jucker Co. Ltd., Mr Narong
Berli Jucker House
99, Soi Rubia, Sukhumvit 42 Road
P.O. Box 173 BMC, Bangkok 1000
☎ +66 (2) 3 67 11 11
Fax +66 (2) 3 67 10 00

Vietnam

Hanoi
Siemens AG, Representation Office
Mr Nguyen Huang Giang
18, Phan Boi Chau Street
☎ +84 (4) 25 60 61
Fax +84 (4) 26 62 27

Australia

Australia

Adelaide
Siemens Ltd. Adelaide Office, CS/I.A.,
Mr J. Weiss, 315 Glen Osmond Road
Glenunga, S.A. 5064
☎ +61 (8) 3 79 66 66
Fax +61 (8) 3 79 08 99

Melbourne
Siemens Ltd., CS/I.A., Mr N. Gilholm,
544 Church Street
Richmond, Victoria 3121
☎ +61 (3) 4 20 75 20
Fax +61 (3) 4 20 75 00

Perth
Siemens Ltd., CS/I.A., Mr A. Lostrom
153, Burswood Road
Victoria Park, W.A. 6100
☎ +61 (9) 3 62 01 42
Fax +61 (9) 3 62 01 47

Sydney, N.S.W. 2064
Siemens Ltd. Sidney, Industrial
Automation, Mr Stephen Coop,
383 Pacific Highway, Artamon
☎ +61 (2) 4 36 78 04
Fax +61 (2) 4 36 86 24

New Zealand

Greenlane, Auckland 5
Siemens Ltd. Auckland Office,
CS/I.A., Mr A. Richmond
300 Great South Road
P.O. Box 17-122
☎ +64 (9) 5 20 30 33
Fax +64 (9) 5 20 15 56

Glossary

A

ACCU

Accumulators are registers in the → C7 CPU and are used as interim memory for load, transfer, compare, math and convert instructions.

Address

The address indicates the physical memory location and allows direct access to the value stored at the address.

Analog Input/Output

Analog inputs/outputs convert analog process values (for example temperature) to digital values that can be processed by the C7 CPU or convert digital values to analog manipulated values.

APF

ASi-Power-Fail. Signal that indicates that the power supply on the AS-i cable is too low or has failed (for example AS-i power supply failure).

Area Pointer

This is required to allow a data exchange between the operator monitoring and control section and the C7 controller. This contains information about the position and size of data areas in the controller.

AS-i

Actuator Sensor Interface

AS-i Driver

Driver that makes the services provided by the AS-i CP available to user programs.

AS-i Library

Library, with which user programs can communicate with the AS-i driver.

Authorization Input

External access to the super user password level.

B

Backup Memory

The backup memory buffers memory areas of the C7 that do not have a backup battery. A selectable number of timers, counters, memory bits and data bytes are backed up, these are known as → retentive timers, counters, memory bits and data bytes.

Bit Memory

Bit memory is part of the → system memory of the CPU for saving interim results. This can be accessed in units of bits, bytes, words or double-words.

C

C7-620

The C7-620 consists of an S7-300 CPU, COROS-OP, I/Os and IM360 interface module integrated in one device.

C7 CPU

The C7 CPU (central processing unit) is a central module of the C7 with control and arithmetic unit, memory, operating systems and interfaces for programming devices. The C7 CPU is independent of the → C7 OP. The C7 CPU has its own MPI address and is connected to the C7 OP via the MPI interface.

C7 I/Os

The C7 I/Os (→ signal modules) form the interface between the process and programmable logic controller. The I/Os provide digital inputs and outputs as well as analog inputs and outputs. The integrated universal inputs on the C7 have special functions (interrupt/counter inputs).

C7 OP

The C7 OP of the C7 processes the OP functions of the C7. It is independent of the → C7 CPU and continues to operate when the C7 CPU changes to the STOP mode. The C7 OP has its own MPI address and is connected to the C7 CPU via the MPI interface. The C7 OP is connected to a configuration computer (programming device/PC) via this MPI interface.

Communications Processor

Communications processors are modules for point-to-point and bus connections.

Complete Restart

When a C7 CPU starts up (for example after selecting one of the **C7 CPU modes** in the system functions menu or when the power is turned on), the organization block OB100 (complete restart) is executed prior to cyclic program execution (OB1). During a complete restart, the process image input table is read in and the STEP 7 user program is executed starting at the first instruction in OB1.

Compress

With the “Compress” PG online function, all the valid blocks in the RAM of the C7 CPU are shifted to the beginning of the user memory. This eliminates and gaps caused by deleting or correcting blocks.

Configuration

Configuration

1. Assignment of modules to racks/slots and (for example with signal modules) addresses.
2. Specification of system-specific basic settings, messages and screens using the configuration software ProTool/Lite.

Configuration Memory

The configuration memory is a flash memory integrated on the C7 OP in which the configuration data are stored.

Control Job

Triggering a function by the C7. Handling control jobs is described in the manual.

Counter

Counters are part of the → C7 CPU. The content of the “counter cells” can be modified by STEP 7 instructions (for example count up/count down).

CP

Communication processors (CP) are intelligent modules with their own processor. They form an important group within the components of a programmable logic controller. Depending on their particular task, we distinguish between various types of communications processors, for example CPs for signaling and logging, for point-to-point links, for operator control and monitoring (COROS), for bus connections (SINEC) for diagnostic and mass memory applications.

Cyclic Interrupt

A cyclic interrupt is generated at selectable intervals by the C7 CPU. A corresponding organization block is then executed.

D

Data Block

Data blocks (DBs) are data areas in the user program containing user data. There are shared data blocks that can be accessed by all logic blocks and instance data blocks that are assigned to a particular FB call.

Data, Static

Static data are data that are used only within a function block. These data are stored in an instance data block belonging to the function block. The data stored in the instance data block are retained until the next function block call.

Data, Temporary

Temporary data are local data of a block that are entered in the L stack while the block is being executed and are no longer available after execution of the block is completed.

Default Setting

The default setting is a functional basic setting that is always used when no other value is entered.

Diagnostics

→ Diagnostic functions → system diagnostics

Diagnostic Interrupt

Modules with diagnostic capability signal detected system errors to the → C7 CPU using diagnostic interrupts.

Diagnostic Events

Diagnostic events are, for example errors in a digital function on the C7, system errors on the C7 that were caused for example by a program error or transitions between operating modes.

Diagnostic Functions

The diagnostic functions cover the entire system diagnostics and include the detection, evaluation and signaling of errors within the C7.

Display Function

Function that leads to a change in the content of the display, for example display message level, display error message buffer, display screen.

Display Time

The time between the arrival and departure of an event message.

E

Entering State (Message)

Time at which a message is triggered by the C7.

Error Display

The error display is one of the possible reactions of the operating system to a → run-time error. The other possible reactions are: → error reaction in the user program, STOP state on the 7-CPU.

Error Handling with OBs

If the operating system recognizes a particular error (for example access error with STEP 7), it calls an organization block programmed to deal with the situation (error OB). The organization block then determines the response of the C7 CPU.

Error Reaction

Reaction to a → run-time error. The operating system can react in the following ways: change the C7 CPU to the STOP mode, call an organization block in which the user can program a reaction or display the error.

Event Message

Indicates certain operating states on the machine or plant connected to the C7.

F

FB

Function block

FC

Function

Fields

Reserved areas in the configured or fixed texts for output and/or input of values.

Flash EPROM

FEPROMs correspond to electrically erasable EEPROMS in that they retain data if there is a power outage. FEPROMs are, however, erased much more quickly (FEPROM = Flash Erasable Programmable Read Only Memory).

The following data can be stored, safe from power outages in the flash memory:

- The → user program
- The → parameters that determine the response of the → C7 CPU and the I/O functions of the C7.

Flash Memory

→ Flash EPROM

Forced Printout

Automatic printout of fault and event messages deleted due to a buffer overflow.

Function

According to IEC 1131-3 a function (FC) is a → logic block without→ static data. A function allows the transfer of parameters in the user program. Functions are suitable for programming recurring complex functions, for example calculations.

Function Block

According to IEC 1131-3 a function block (FB) is a → logic block with → static data. An FB allows parameters to be transferred in the user program. Function blocks are suitable for programming recurring complex functions for example closed-loop controls, mode selection.

H

Hard Copy

Output of the contents of the display on a connected printer.

Hardware Interrupt

A hardware interrupt is triggered by interrupt triggering modules due to a particular event in the process. The hardware interrupt is signaled to the C7 CPU. Depending on the priority of this interrupt, the assigned → organization block is executed.

I

Information Function

The STEP 7 information functions allow you to display status information on the connected C7 during the various phases of startup and operation of a programmable logic controller.

IM

Interface module: module for expanding the S7-300 system

Information Text

Configurable additional information about message, screens, screen entries and selection fields.

Instance Data Block

Each function block call in the STEP 7 user program is assigned a data block that is generated automatically. The instance data block contains values for input, output and in/out parameters as well as the local block data.

Interface, Multipoint

→ MPI

Interrupt

The → operating system of the C7 CPU recognizes ten different priority classes that control the execution of the user program. These priority classes include among other things interrupts, for example hardware interrupts. If an interrupt occurs, the operating system automatically calls an assigned organization block in which the user can program a reaction to the interrupt (for example in an FB).

L

LAS

List of active slaves

Leaving State (Message)

Point in time at which a message is cleared by the controller.

LDS List of detected slaves

Load Memory The load memory is part of the C7 CPU. It contains objects (load objects) created by the programming device. This is a permanently integrated memory.

Logic Block In SIMATIC S7, a logic block is a block that contains part of the STEP 7 user program. (In contrast to a → data block: this contains only data).

LPS List of permanent (configured) slaves

M

Memory Reset During a memory reset on the → **C7 CPU**, the following memory is cleared:

- The → work memory
- The write/read area of the → load memory
- The → system memory
- The → backup memory

and the user program is reloaded from the flash memory.

When a memory reset is performed on the → **C7 OP**, the following memory is cleared:

- The → work memory
- The configuration memory

This means that there is no longer a user configuration loaded.

Message Level Operating level of the C7 in which messages are displayed.

Message Logging Parallel to the display, fault and event messages are printed out.

MPI The multipoint interface (MPI) is the programming device interface of SIMATIC S7. It allows simultaneous operation of several nodes (programming devices, text displays operator panels) connected to one or more CPUs. The nodes on the MPI are connected to each other using a bus system. Each node is identified by a unique address (the MPI address).

N

Nesting Depth

A block can be called from within another block by block calls. The nesting depth is the number of simultaneously called → logic blocks.

Network

A network is a connection of several C7 and/or S7-300 and other devices, for example a PG via a → connecting cable. Data is exchanged between the connected devices via the network.

Normal Mode

C7 mode in which messages are displayed and input can be made in screens.

O

OB

→ Organization block

OB Priority

The → operating system of the C7 CPU distinguishes between different priority classes, for example cyclic program execution, hardware interrupt driven program execution. Each priority class is assigned → organization blocks (OBs) in which the S7 user can program a reaction. As standard, the OBs have various priorities that decide the order in which they are executed if they are called simultaneously. The priorities also decide which blocks can interrupt other blocks.

Operating System of the C7 CPU

The operating system of the C7 CPU organizes all the functions and sequences of the C7 that are not associated with a special control task.

Organization Blocks

Organization blocks (OBs) perform the interface between the operating system of the C7 CPU and the user program. The order of execution of the user program is specified in the organization blocks.

Output Field

Field for displaying an actual value.

P

Parameter

1. Variable of a STEP 7 logic block.
2. Variable for setting the response of a module (one or more per module). When shipped, each module has a functioning basic setting that can be modified using the STEP 7 tool *S7 Configuration* .
There are → static parameters and → dynamic parameters.

Parameter, Dynamic
Dynamic parameters of modules, in contrast to static parameters, can be modified during operation by calling an SFC in the user program, for example limit values of an analog signal input module.

Parameter, Static
Static parameters of modules, in contrast to dynamic parameters, cannot be modified by the user program but only using the software tool *S7 Configuration* , for example input delay of a digital signal input module.

Parameter Assignment
Parameter assignment involves the setting of parameter values to specify the response of a module.

Password
Password Level
A protected function can only be used after a password has been entered that belongs to a password level high enough for the function. The password level indicates the right of the operator to use the function. The password level is specified during configuration and can range from 0 (lowest level) to 9 (highest level).

PG
Programming device

PLC
→ Programmable logic controller

Process Image
The process image is part of the → system memory of the C7 CPU. At the start of the cyclic program, the signal states of the input modules are transferred to the process image input table. At the end of the cyclic program, the process image output table is transferred as a signal state to the output modules.

Programmable Logic Controller
Programmable logic controllers (PLCs) are electronic controllers whose function is stored as a program on the control unit. The design and wiring of the device therefore do not depend on the function of the controller. The programmable logic controller has the structure of a computer; it consists of a → C7 CPU with memory, inputs/outputs and an internal bus system. The I/Os and the programming language are designed to meet the requirements of control engineering.

Programming Device
Programming devices are essentially personal computers that are designed for use in the industrial sector and are particularly compact and portable. They have a hardware and software especially designed for SIMATIC programmable logic controllers.

R

RAM

A RAM (Random Access Memory) is a memory in which each memory cell can be addressed and modified singly. RAM memory is used to store data and programs.

Reference Data

The reference data are used to check a C7 CPU program and include the cross-reference list, assignment list, program structure and list of unused addresses. The STEP 7 user manual explains how these data can be read out.

Replacement Value

Replacement values are values that are output to the process when signal output modules are defective or that are used in the user program instead of a process value when signal input modules are defective. The replacement values can be selected by the user (for example the old value is retained).

Retentiveness

Data areas in data blocks and timers, counters and bit memory are retentive when their content is not lost during a complete restart or power outage.

S

Screen

Representation of logically associated process data that are displayed together on the C7 and can be modified individually.

Screen Entry

Element of a screen consisting of the entry number, texts and variables.

Screen Level

Operating level on the C7 on which screens can be monitored and input made to them.

Selection Field

A field for setting the value of a parameter (a value can be selected from a range of possible values).

SFB

→ System function block

SFC

→ System function

Signal Module

Signal modules (C7 I/Os) form the interface between the process and the C7. There are digital inputs and outputs as well as analog inputs and outputs.

Softkeys Keys with a variable function assignment (depending on the displayed screen entry).

STARTUP The STARTUP mode is run through when the CPU changes from STOP to RUN.

Startup Test The status of the CPU and memory is checked when the power is turned on.

STEP 7 Programming software for creating user programs for SIMATIC S7 controls.

STEP 7 Tool A STEP 7 tool is a tool belonging to → STEP 7 designed for a specific task.

System Function A system function (SFC) is is a → function integrated in the operating system of the C7 CPU. When required, system functions can be called in the STEP 7-user program.

System Function Block A system function block (SFB) a → function block integrated in the operating system of the C7 CPU. When required, system function blocks can be called in the STEP 7-user program.

System Memory The system memory is integrated on the CPU as RAM memory. The system memory contains address areas (for example timers, counters, bit memory) and data areas required internally by the → operating system (for example buffers for communication).

System Message Indicates internal states on the C7 and the controller.

T

Time-Delay Interrupt The time-delay interrupt belongs to one of the priority classes in the program execution of SIMATIC S7. It is generated when a time started in the user program expires. A corresponding organization block is then executed.

Time-of-Day Interrupt The time-of-day interrupt belongs to one of the priority classes for the execution of the C7 CPU program. It is generated on a certain date (or daily) and at a certain time (for example 9:50 or every hour, every minute). A corresponding organization block is then executed.

Timer

Timers are components of the → C7 CPU. The operating system updates the content of the “timer cells”automatically asynchronous to the user program. The exact function of the timer (for example on delay) is specified with STEP 7 instructions and their execution is triggered (for example start).

Transfer Mode

C7 mode in which the data are transferred from the programming device to the C7 or vice versa. A distinction is made between a transfer (S7 transfer) via the MPI interface and via the printer interface (transfer).

Transmission Rate

Speed of data transmission (bps).

U

User Memory

The user memory contains → logic and → data blocks of the user program. The user memory is integrated in the C7 CPU as a flash memory. The user program is, however, always processed in the → work memory of the C7 CPU.

User Program

The user program contains all the → instructions and declarations as well as data for signal processing with which a plant or process can be controlled. The program is assigned to a programmable module (for example C7 CPU, FM) and can be structured in smaller units (blocks).

W

Work Memory

The work memory is a RAM on the C7 which the processor accesses during execution of the user program.

Index

A

B

C

D

G

H

I

K

L

M

N

O

S

Siemens AG
AUT E 146

Östliche Rheinbrückenstr. 50
D-76181 Karlsruhe
Federal Republic of Germany

From:
Your Name: _
Your Title: _
Company Name: _
Street: _
City, Zip Code _
Country: _
Phone: _

Please check any industry that applies to you:

- ❐ Automotive
- ❐ Chemical
- ❐ Electrical Machinery
- ❐ Food
- ❐ Instrument and Control
- ❐ Nonelectrical Machinery
- ❐ Petrochemical
- ❐ Pharmaceutical
- ❐ Plastic
- ❐ Pulp and Paper
- ❐ Textiles
- ❐ Transportation
- ❐ Other _ _ _ _ _ _ _ _ _ _ _

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _