SIEMENS

SIMATIC

C7-623, C7-624 Control Systems

Volume 2
Working with C7

Manual

C79000-G7076-C624-01

Table of 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
Automation Group
Industrial Automation Systems
Postfach 4848, D-90327 Nürnberg

Contents

Introduction

1

In this Chapter

This chapter will inform you about the prerequisites for programming the C7 and its functionality as an operator interface device

---

Note

The C7 consists of two independent units, each with its own processor:
- C7-CPU and
- C7-OP

Where necessary, these components are explicitly named in Section 2 of the manual.

What Do you Require to Operate the C7

You require the following equipment and tools:

- A programming device or PC with MPI interface, and a programming device cable
- STEP 7 appplications from version 2, including documentation
- The ProTool or ProTool/Lite configuring tool from version 2, incl. documentation
- C7 connector set for I/O and power supply

1.1 Control with the C7

Overview

The user program that controls the process, that is to be visualized by the operator interface part of the C7, runs in the C7-CPU.

C7 CPU

The operation of the C7 CPU is determined by the following functional elements.

Program memory

It 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 generates the process image input table (PII)
- The program is processed step-by-step, taking into account all counter and timer values and bit memories
- The processor stores the signal states resulting from program execution in the process image output table (PIQ). The states are then transferred to the outputs.

The C7 CPU is independent of the C7 CPU. It has a separate MPI address via which it is connected to the C7 OP.

Programming Languages

At present, the C7-CPU generally supports two programming languages:

- STL: The statement list consists of a series of statements. Each statement in your program contains instructions which mnemonically represent a function of the C7-CPU.
- LAD: A ladder diagram is a graphical programming language that resembles electrical circuit diagrams.

Program with What

The tool with which you prepare user programs is STEP 7. You can find the operation notes necessary for programming in the *User Manual /**231**/*. You can use the manuals listed in the foreword for the individual languages.

With Which Devices

The STEP 7 runs on a programming device or PC. You can operate these devices independently from the C7. You must connect the programming device/PC to the C7 via the MPI only when you wish to load your user program into the C7.

1.2 Operator Control and Monitoring with C7

Overview

Operational states, current process values and faults can be visualized with the C7. In addition, you can perform inputs to the C7 that will be written to the C7-CPU. Machine diagnostics functions can also be configured in the C7.

The C7 offers a series of standard functions. The screens and the operation of the C7 can be optimally tailored to the individual requirements of the process during the configuration.

C7 OP

The C7 OP processes the O/I functions configured for the C7. It is independent of the C7 CPU and is still operable, for example, if the C7 CPU enters the STOP state. The C7 OP is assigned a separate MPI address and connected to the C7 CPU via the MPI interface. This interface forms the link between the C7 OP and the computer used for configuring (programming device/PC).

Operator Interface Functions

The basic functions of a C7 consist of the visualization of process states and the operation of the process. The sequence of operator prompting is controlled by a configuration to be planned and loaded by the user. The following display and operator functions can be configured for C7:

- Screens
- Event messages
- Fault messages
- Recipes
- Information texts
- Logging
- Multi-language operator prompts

Screens

Logically related process data from the control system can be displayed together in one screen and changed individually as appropriate. A screen comprises several screen entries, since, for example, the description of a machine state normally requires more related data than can be represented in one display section. In this manner, data on operating temperature, liquid level, rotational speed and run time can illustrate the current machine state.

The C7-623 and C7-624 have line-oriented displays. A screen is thus compiled from text elements, that can be combined with static text and current status values.

Event Messages

Event messages are information and operating tips regarding the current machine or process states during normal production operations. Event messages can contain process values. The representation of the process values can be either numeric, for example motor running at 3000 revolutions per second, or symbolic.

Recipes

When a product is made up of various ingredients in certain ratios, this information is stored in a so–called recipe.

Take orange juice for example. Its production and bottling requires that orange concentrate and water be mixed in a certain ratio. These values are stored as variables in a recipe.

Each recipe can be made up of several data records containing different values for the recipe.

The data can be edited in the C7–OP, transferred to the C7–CPU and read out from the C7–CPU.

In the C7–OP, the recipes are combined by recipe number and title in a recipe directory.

Fault Messages

Contrary to operational messages, fault messages indicate critical machine states during the production sequence. Due to their importance, they must be acknowledged before further actions are possible.

Fault messages may contain process values. The representation of the process values is either numeric for example motor running at 4500 revolutions per second, or symbolic.

Information Texts

Information texts contain supplementary information which makes reference to the current display (operational messages, fault messages and screens). In this manner, additional information can, for example, be displayed suggesting methods for the removal of a fault after receipt of a fault message.

Logging

Messages can be logged on-line by means of the printer connected to the operator panel. In addition, it is possible to print out the operating and fault messages that have accumulated in the respective buffer.

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 simultaneously loaded into a C7 and offered to the operator for online selection:

- German
- English
- French
- Italian
- Spanish

Displays can be configured for the C7-624 in Russian, that is Cyrillic characters.

Configuration/ Process Management

Before a C7 can be commissioned, it must be prepared, that is configured, for the task of visualizing data from the C7-CPU.

You configure the C7-OP using a computer (PC/programing device) with the configuration software ProTool or ProTool/Lite. After completion of the configuration, it is transferred to the C7-OP. To do this, you must connect the computer to the C7 via an MPI interface.

The project control phase commences once the configuration has been loaded in the C7-OP. The C7-OP now reacts to the process signals from the C7-CPU or operator actions in accordance with the configured instructions.

Information regarding the configuration of the C7-OP can be found in the *ProTool* or *ProTool/ Lite* user manual.

1.3 C7 Overview

The SIMATIC C7-623/C7-624 control systems consist of several components that are optimally tuned to one another:

- A CPU of the SIMATIC S7-300 PLC range (C7 CPU)
- A line-oriented COROS OP (C7 OP) with printer port
- Integral digital and analog I/O (C7 I/O)
- An integrated IM 360 module (C7 IM) for expanding the C7-623 or C7-624 with S7-300 modules
- An interface for communicating with programming devices/PCs and further S7 CPUs, C7 control systems and OPs.

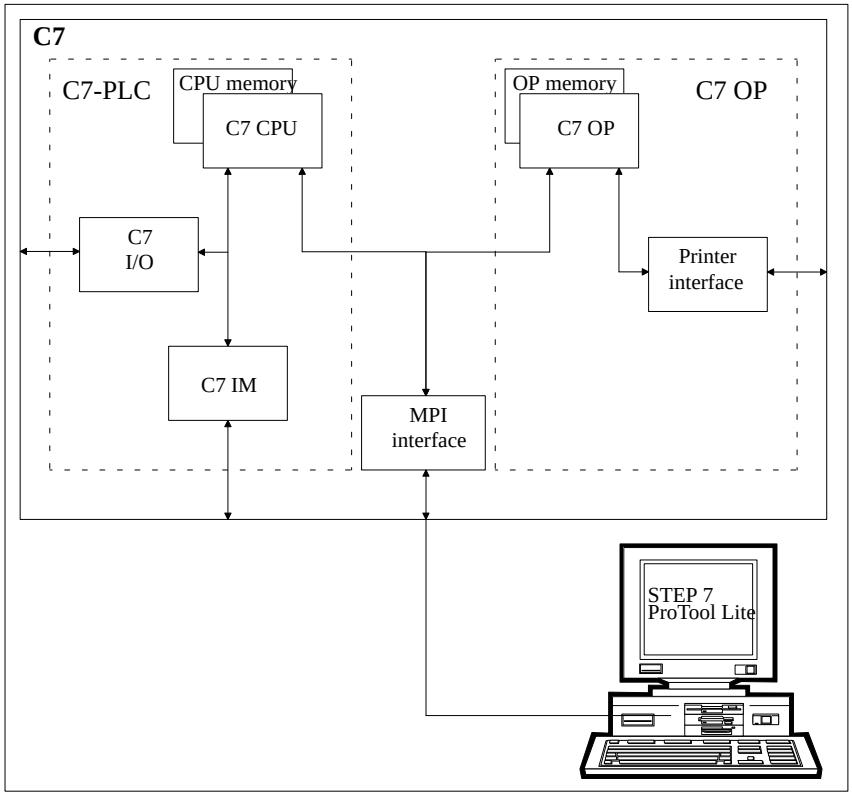


Figure 1-1 Components of the C7

The individual functional units integrated in the SIMATIC C7 correspond to the modules and devices that can be used in modular configurations comprising S7-300 CPUs, COROS OPs, etc. The C7's I/O expansion capability via its IM interface permits the connection of SIMATIC S7-300 modules distributed over three racks with a maximum of 24 S7-300 modules.

The general functionality of the C7 also corresponds to a configuration with standard modules of the PLC and OP ranges. The individual components operate independently of one another and each processor has a separate memory.

STEP 7 is used for programming the C7 CPU whereas the C7 OP is programmed with ProTool or ProTool/Lite. Both tools run under Windows either on programming devices or PCs.

Commissioning (Start-Up)

2

In this Chapter

In this chapter, you will learn:

- How the C7 behaves on start-up
- What you must do if the configuration has/already not been loaded
- How you can activate the C7-CPU operational modes RUNP, RUN, STOP and MRES
- How you can activate the DI/DO status indication
- How you perform a memory reset of the C7 controller

2.1 Start-Up Characteristics and Commissioning

Start-Up

After connecting the power supplies, the C7 performs a self-test. During this test, it checks the functionality of the most important device components and shows the test results on the status LED and display. The following start-up procedure is followed:

1. The C7 performs a self-test after POWER ON.
2. If the configuration has not been loaded, a contrast setting is requested.
   You must confirm this with +/- when the display is easily readable.
3. The C7 performs an operating system test.
4. During the start-up phase (1. and 3.), the C7-CPU remains in the STOP mode. After start-up, the C7 is in the message mode. The following standby messages is displayed:

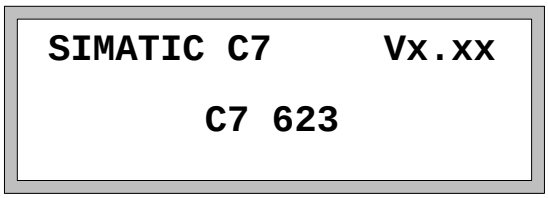


Figure 2-1 C7-623 Standby Message

5. After start-up of the C7, the previously selected C7-CPU mode is valid.

   The following states can occur in the C7. These determine your further actions.

   - **No control program** loaded.
   - A **control program is loaded**
   - **No configuration** loaded
   - A **configuration is loaded**

   How you should proceed in these cases is explained in the following sections.

Load Control Program

The C7-OP must access data in the C7-CPU in order to permit the operation and monitoring of the process. Therefore the control program must be loaded first of all, if this has not already been done. You load the application program in the following way:

1. Activate the transmission of the application program and the data blocks on your programming device/PC by means of STEP 7
2. Set the C7-CPU in the STOP mode (see Chapter 2.2).
3. Start the copy process from the programming device/PC.

No Configuration Loaded

You now select the C7-OP mode in which you can initiate the transfer of a configuration. To do this you press the ENTER key.

The following menu is displayed with the menu item “off-line”:

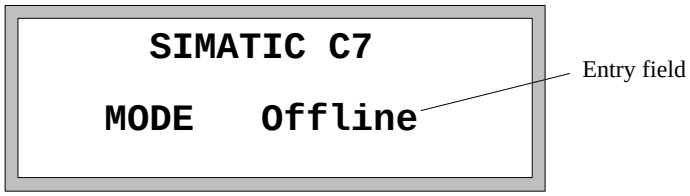


Figure 2-2 Menu for Selection of a C7-OP Mode

1. Select the **Transfer** function with SHIFT ▽
   Confirm the selection with ENTER

   The configuration will be loaded via the MPI interface using **Transfer.**
2. Activate the transmission of the configuration to C7 using your programing device/PC with “ProTool or ProTool/ Lite”.
3. As soon as the configuration has been loaded, the message mode is set and the configured standby message is displayed.
4. You can now change to the following levels.

- C7 system function menu SHIFT HELP
- Screens (the configuration start screen) ENTER

Configuration Loaded

After a configuration has been loaded, this can be started by pressing the ENTER key and the configured “Start” screen will be displayed. After this, operator actions are only possible in the sequence defined in the configuration.

Reload Configuration

If a configuration has been loaded, you can reload a new configuration only after activation of the standard screen **Load Mode** or by means of the configured function **Change Mode**. The modes On-line, Off-line, Transfer can be selected from the standard screen (see Section KEIN MERKER). The function **Change Mode** immediately activates the corresponding transfer mode.

If the standard screen or the above function is not available in the currently loaded configuration, you must remove the loaded configuration via a memory reset before reloading. Detailed descriptions of the Memory reset function can be found in Section 2.3.

Memory reset procedure:

1. Switch off C7.
2. Simultaneously depress and hold the keys 

3. Switch on the C7.
4. Further actions as described under “No configuration loaded”.

2.2 C7-CPU Mode Selection and DI/DO Status Display

System Function Menu

You can select the system function menu from all operational modes. From this menu, the following additional menus can be selected:

- C7-CPU mode selection
  - RUN-P
  - RUN
  - STOP
  - MRES
- DI/DO status display

Selection of the System Function Menu

You can select the menu by pressing the keys SHIFT HELP The Shift LED that was lit after pressing Shift goes dark.

The following menu is displayed:

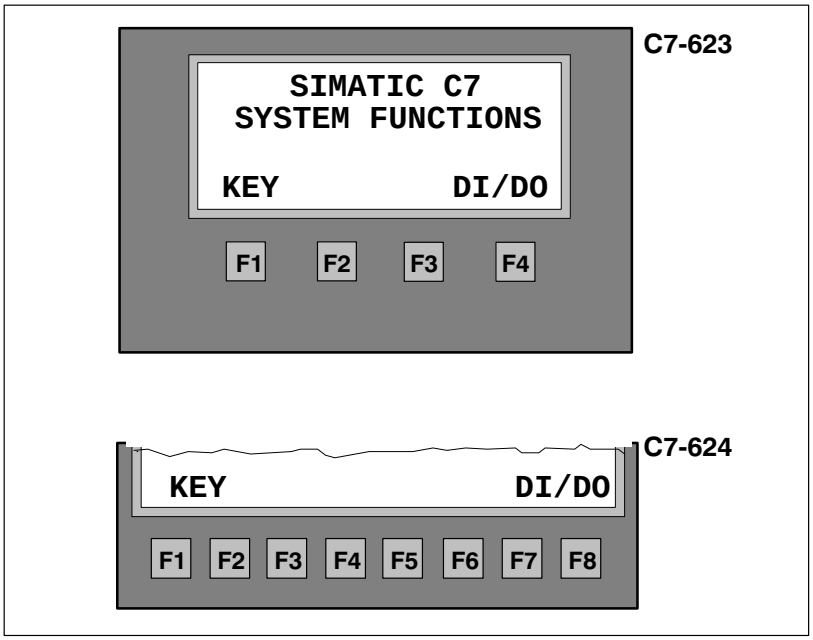


Figure 2-3 System Function Menu with Associated Function Keys

Select Mode Menu

You select the **C7-CPU Modes menu** from the system function menu in the following manner:

- C7-623: by depressing the key F1 or F2
- C7-624: by depressing one of the keys F1 .... F4

The following menu will be displayed (MODE: STOP is used as an example):

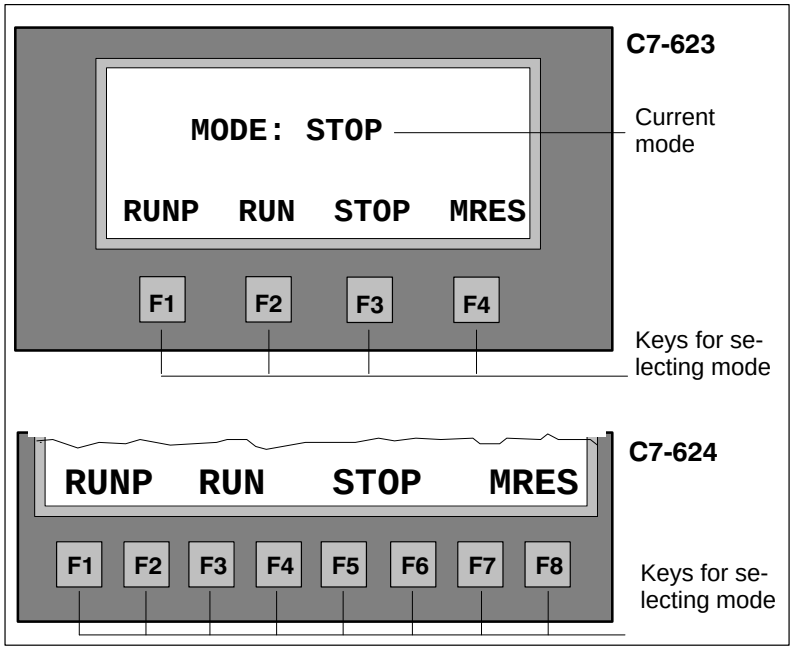


Figure 2-4 “C7-CPU Modes” Menu with Associated Function Keys

Select C7-CPU Modes

You select the individual C7-CPU modes in the following manner:

Table 2-1 Select C7-CPU Mode

Mode	Key C7-623	Keys C7-624	Explanation
RUNP	F1	F1 or F2	The C7-CPU processes the user program. Programs and data can: • Be read out from the C7-CPU with programming device (C7 → programming device) • Be transmitted to the C7-CPU (programming device → C7; exception: OB1)
RUN	F2	F3 or F4	The C7-CPU processes the user program. Programs and data in the C7-CPU can be read out using the programming device (C7 → programming device). The program cannot be loaded or altered. Data cannot be loaded or altered.
STOP	F3	F5 or F6	The C7-CPU is not processing an user program. Programs can: • Be read from the C7-CPU using the programming device (C7 → programming device) • Be transmitted to the C7-CPU (programming device → C7) **Note:** The operating mode STOP is valid only for the C7-CPU. It is not valid for the C7-OP. Further processing with the C7-OP is possible.
MRES	F4	F7 or F8	**Memory reset** The memory reset of the C7-CPU (erase memory, reload user program from flash memory) requires a special operating sequence of the modes STOP and MRES (see Chapter 2.3). If data that is required in the configuration was destroyed during the memory reset, then a corresponding error message is issued by the C7-OP. **Note:** The MRES position is not a momentary-contact state which means that the MRES status persists. For the C7 CPU, the MRES status is only a control mode. When this mode is set permanently, the C7 CPU does not function properly. This mode **must** therefore always be reset prior to exiting the menu with STOP; RUN or RUN-P.

Mode Change Protected by Password

If a configuration is not loaded into the C7-OP, then you can change the C7-CPU mode at any time without password protection.

When a configuration has however been loaded' password protection is activated to prevent uncontrolled C7-CPU mode change during process control operations. The procedure is as follows:

1. If the C7-CPU mode is to be changed, the active password level is checked (password level >=8 is necessary).
2. If the password level is not sufficient, the LOGIN screen for password entry is automatically displayed (see Section 5.8).
3. Enter password
   - You can change the C7-CPU mode only with a valid password.
   - If no keys are operated within a time specified during the configuration, an automatic log-off is initiated (reset of the current password level to 0 = lowest level).
   - If a password for Level = 8 has not yet been allocated, you can only change the C7-CPU mode by means of the configured superuser password **(default 100)**.
4. If the password has been recognized as valid, you can now change the C7-CPU mode.

Exit the Mode Menu

You can exit the **C7-CPU mode** menu by pressing the key ESC

Select DI/DO Status Display

The displayed values are read as a direct **process image** of the DI and internal process image of the DO of the digital C7 I/O and are displayed in BIN format. It is not possible to alter the display.

It is to be noted that in the STOP state of the C7-CPU, the real process state is DO = 0, however the last value set by the program will be displayed.

The selection of the DI/DO status display does not require a password level. The function can be selected from the System Functions menu:

- C7-623: by pressing the F3 or F4 keys
- C7-624: by pressing one of the keys F7 or F8

The following data will be displayed:

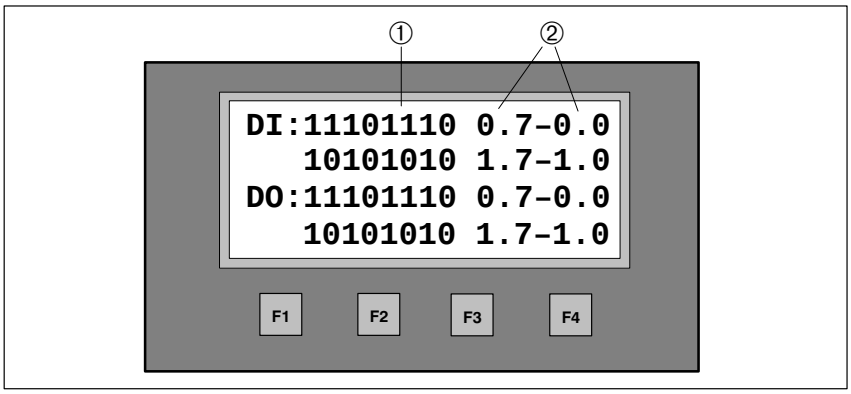


Figure 2-5 DI/DO Status Display on a C7 623

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

Position	Explanation
①	Signal state of the DI/DO • 1 DI/DO set • 0 DI/DO reset
②	Pin numbers from - to

Note

The values of the DI/DO are read and displayed every 400 ms. Any signal changes occurring between two scan points are not indicated.

Exit the DI/DO Display

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

Exit the System Functions Menu

You exit the System Functions Menu by pressing the ESC key.

2.3 Memory Reset of the C7

Overview

If you wish to achieve a neutral state of the C7, you must completely erase the C7-CPU and possibly also the C7-OP.

C7-OP Memory Reset

The C7 must be switched off. You follow the following procedure to completely erase the C7-OP:

1. Depress and hold the following keys simultaneously 

2. Switch on the C7.

After this operation, the C7-OP is cleared.

What Remains after a Memory Reset of the C7-OP ...

After clearing the C7-OP, the following items remain:

- C7 system functions menu
- C7-OP mode selector MODE for loading the configuration

Memory Reset of the C7-CPU

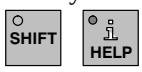
There are two methods of erasing the C7-CPU:

- Erase with the system function **C7-CPU Modes**
- Erase with programming function (see programming device manual)

The complete erasure using the programming device function is only possible when the C7-CPU is in STOP mode.

How to Clear the C7-CPU

How you clear the C7-CPU with the help of the system function **C7-CPU Modes** is described below.

1. Switch on the C7 and wait until the start-up tests have been completed. The standby message will be displayed.
2. Select the system function menu by pressing the keys 

   The system function menu will be displayed:
3. Select the **C7-CPU Modes menu** in the following manner:
   - C7-623: by pressing the key 

or 

   - C7-624: by pressing one of the keys 

.... 

   The menu with the **C7-CPU Mode** RUNP, RUN, STOP, MRES will be displayed.

4. Select the STOP by pressing the corresponding function keys. The STOP lights up.
5. Select the MRES function (memory reset) and wait (approx. 3 seconds) until the STOP LED lights up again.
6. Immediately after the STOP LED lights up again: select STOP with the corresponding function keys and then MRES a second time.

   Result:

   – If the STOP LED flashes for approx. 3 seconds and then lights up again: everything is O.K.; C7 has been completely cleared.

   – If the STOP LED does not flash, or other indicators light up or flash (exception: BATF LED): repeat steps 4 and 5; perhaps evaluate the diagnostic backup of the C7 using the programming device.

   – If the BATF and the SF LEDs on the C7 light up, the buffer battery is absent. If a buffer battery is indeed fitted, then you must search the diagnostic backup of the C7-CPU for additional error entries.
7. In order to be able to continue working, you must set the C7-CPU to STOP or RUN / RUNP.

Procedure in the C7-CPU During Memory Reset

The STOP LED flashes during the memory reset of the C7-CPU and the following process is performed:

1. The C7-CPU erases the entire application program in the work memory and in the load memory.
2. The C7-CPU erases the backup memory.
3. The C7-CPU tests the hardware.
4. If an application is stored in the integrated flash memory of the C7-CPU, its contents will be automatically copied into the load memory (see Section 3.8, Load/Erase).

   If no application is stored then the load memory remains empty and the C7-CPU has the memory content “0”.

What Remains after Clearing the C7-CPU ...

After clearing the C7-CPU, the following items remain:

- The contents of the diagnostic backup

  The contents can be read out using the programming device.
- System diagnostics parameters
- Perhaps an user program loaded from the flash memory with newly initialized data
- The MPI parameters last set.

2.4 Load/Erase C7-CPU Flash Memory

Overview

During transmission of an application program to the C7-CPU, it is transmitted only to the load memory and not automatically into the C7-CPU flash memory also.

The contents of a C7-CPU flash memory are not automatically reset during a complete erasure of the C7-CPU.

You must explicitly initiate these actions using a PG function.

Load Application Program into Flash Memory

You must explicitly load the C7 flash memory using the STEP 7 function “copy RAM to ROM”. You do this after you have copied the program and data into the C7-CPU load memory. In this way, the whole contents of the load memory is transmitted to the flash memory.

Erase Flash Memory

After the complete erasure, the load memory is reinitialized with the contents of the flash memory. However, since you can change the contents of the flash memory only using the STEP 7 function “copy RAM to ROM”, you must proceed in the following manner to erase the flash memory:

1. Erase the entire contents of the load memory using the appropriate STEP 7 function.
2. Activate the STEP 7 function “copy RAM to ROM”. In this way, the empty load memory will be ”loaded” into the flash memory. Afterwards, this is also empty.

Control with the C7-CPU

3

In this Chapter

The C7-CPU is described in this chapter.

You will learn which languages can be used to program the C7-CPU and which tools are necessary.

The characteristics of the C7-CPU can be configured using parameters. The characteristics that can be configured are described in Chapter 3.5.

3.1 C7-CPU: Overview

C7-CPU Characteristics

C7-CPU possesses the following characteristics:

- 24 Kbytes work memory
- 40 Kbytes integrated load memory RAM
- 40 Kbytes integrated flash memory
- Integrated IM 360
- Speed: approx. 0.3 ms per 1000 statements
- Connection of max. 512 digital inputs/outputs
- Connection of max. 128 analog inputs/outputs
- Buffer battery

3.2 Programming the C7-CPU

Overview

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

Required Tools

You require the following tools to develop the user program:

- Programming device/PC plus cable
- STEP 7 with the appropriate manuals
- C7

Programming Languages

Two programming languages are currently relevant to the C7-CPU:

- STL: The statement list consists of several statements. Each statement in your program contains operations, which represent a function of the C7-CPU as a mnemonic.
- LAD: A ladder diagram is a graphical programming language which resembles an electrical circuit diagram.

3.3 Performance Characteristics of the C7-CPU

Introduction

Table 3-1 lists the most significant performance characteristics of the C7-CPU.

Table 3-1 Performance Characteristics of the C7-CPU

Performance Characteristic	C7-CPU
Load memory	• Integrated RAM 40 Kbytes • Integrated flash memory 40 Kbytes
Work memory	Integrated RAM (24 Kbytes)
Speed	Approx. 0.3 ms per 1000 statements
Digital inputs/outputs	16/16
Analog inputs/outputs	4/1
Universal inputs	4
Bit memories	2048 From M 0.0 to 255.7 Selectable retentivity; Preset: 16 retentive memory bytes (from 0 to 15)
Counters	64 From C 0 to C 63 Selectable retentivity; Preset: 8 retentive counters (from 0 to 7)
Times	128 From T 0 to T 127 Selectable retentivity; Preset: no retentive times
Retentive data area 0	Max. eight data areas from one or more data blocks Maximum of 4096 retentive data bytes
Maximum total of all retentive data areas	4544 bytes
Clock memory	Memory that can be used in the user program to obtain a clock beat. Number: 8 (1 memory byte); freely selectable address of a memory byte
Local data	Total 1536 bytes 256 bytes for each priority class
Process image area	From 0 to 127 Digital inputs: from E 0.0 to E 127.7 Digital outputs: from A 0.0 to A 127.7
Nesting depth	8 for each priority class 4 additional within an error OB

Table 3-1 Performance Characteristics of the C7-CPU

Performance Characteristic	C7-CPU
Blocks	
OBs	13
FBs	128
FCs	128
DBs	127
SDBs	9
SFCs	37
SFBs	-
Clock	Hardware clock
Run-time meter	1

3.4 C7-CPU Blocks

Overview

Table 3-2 lists all blocks that the C7-CPU can process.

Table 3-2 Overview: C7-CPU Blocks

Block	Number	Range	Maximum Size	Comment
OB	13	-	8 Kbytes	A complete list of all possible OBs can be found at the end of this table.
FB	128	0 - 127	8 Kbytes	-
FC	128	0 - 127	8 Kbytes	-
DB	127	1 - 127	8 Kbytes	0 is reserved
SFC	34	-	-	A complete list of all SFCs in the C7-CPU can be found in Appendix A. A detailed description can be found in the STEP 7 documentation.

Organization Block (OB)

The operating system of the C7-CPU is based on event controlled user program processing. The following table shows which organization blocks (OBs) are automatically called up by the operating system for which event.

Description of the OB

A detailed description of the various OB and their user can be found in the manual **/280/.**

Size of an OB

An OB can have a maximum size of 8 Kbytes.

OB for Scan Cycle and Start-Up

Table 3-3 lists the OBs which determine the behavior of the C7-CPU during the scan cycle and start-up.

Table 3-3 List of OB for Scan Cycle and Start-up

Scan Cycle and Start-up	Activated OB	Possible Start Events	Preset Priority of the OB
Scan cycle	OB 1	1101_H, 1103_H	Lowest priority
Start-up (STOP-RUN transition)	OB 100	1381_H, 1382_H	-

OB for Internal and External Alarms

The Table 3-4 lists OBs which determine the behavior of the C7-CPU after alarm events.

The priority of the OB cannot be changed.

Table 3-4 List with C7-CPU Alarm Events

Alarms (Internal and External)	Activated OB	Possible Start Events	Priority of the OB	Priority
Time-of-day alarm	OB 10	1111_H	2	Low
Delay alarm Range: 1 ms to 60000 ms (adjustable in 1 ms increments)	OB 20	1121_H	3	↓
Watchdog alarm Range: 1 ms to 60000 ms (adjustable in 1 ms increments)	OB 35	1136_H	12	
Process alarm	OB 40	1141_H	16	
Diagnostics alarm	OB 82	3842_H, 3942_H	26	High

Behavior of the C7-CPU with Absent OB

The C7-CPU reverts to STOP if a

- Time-of-day alarm
- Delay alarm
- Process alarm
- Diagnostics alarm

occurs, but the corresponding OB has not been programmed.

The C7-CPU does not revert to STOP if a watchdog alarm occurs and OB 35 has not been programmed.

OB for Error Reactions

The Table 3-5 lists the OBs which determine the behavior of C7-CPU in an error situation.

The C7-CPU reverts to STOP if an error occurs, but the corresponding OB has not been programmed.

Table 3-5 List of OBs

Error	Activated OB	Possible Start Events	Preset Priority of the OB
Time error (e.g. initiated by the cycle time monitor)	OB 80	3501_H, 3502_H, 3505_H, 3507_H	26
Power supply error (e.g. also absence of backup battery)	OB 81	3822_H, 3922_H	26

Table 3-5 List of OBs

Error	Activated OB	Possible Start Events	Preset Priority of the OB
One of the following errors has occurred: • Event that triggers OB start (e.g. delay alarm) has occurred but the relevant OB cannot be executed • Error during update of the process image	OB 85	$35A1_H$, $39B1_H$, $39B2_H$	26
Communications error • Wrong message frame identifier when receiving global data • The data block for the global data status is not available or too short	OB 87	$35E1_H$, $35E2_H$, $35E6_H$	26
Programming error (e.g. the 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 has occurred
Error during direct peripheral access (defective or absent module)	OB 122	2944_H, 2945_H	The same priority as the OB in which the error has occurred

OB 121 and OB 122

Please note the following peculiarity of the C7 CPU in connection with OBs 121 and 122:

Note

Please note the following peculiarity for OBs 121 and 122:

The C7 CPU enters the value ”0” in the following temporary variables of the variable declaration table in the local data of the OBs:

- **Byte no. 3**: OB121_BLK_TYPE or OB122_BLK_TYPE (type of block where error occurred)
- Byte nos. 8 and 9: OB121_BLK_NUM or OB122_BLK_NUM (number of block where error occurred)
- **Byte nos. 10 and 11**: OB121_PRG_ADDR or OB122_PRG_ADDR (address in block where error occurred)

3.5 C7-CPU Parameters

Configurable Characteristics of the C7-CPU

The characteristics and behavior of the C7-CPU can be configured.

Parameter blocks of the C7-CPU:

- Clock memory
- Start-up characteristics
- System diagnostics
- Retentive areas
- Process alarm
- Real time clock
- Time-of-day alarm
- Watchdog alarm
- Cycle behavior
- MPI station addresses

Tool for Parameter Assignment

The tool that you use to assign the parameters to the C7-CPU is the STEP 7 Function *Hardware configuration*. Working with *Hardware configuration* is described in manual **/100/**.

When Does the C7-CPU "Accept" the Parameters

The C7-CPU accepts the selected parameters

- After POWER ON
- After the parameters have been transmitted online and error-free to the C7-CPU in STOP mode.
- After erasing the C7-CPU (see Chapter 2.3)
  If a SDB0 is available on the integrated flash memory, then the stored parameters will be loaded with the exception of the MPI parameter. If no SDB0 is present in the flash memory, then the standard parameters of SDB2 will be set.

Caution

After the last storage of the program in the flash memory of the C7-CPU (STEP 7: "Copy RAM to ROM"), the set parameters will be lost after the next erase, with exception of the MPI parameters.

3.5.1 Parameter Block “Clock Memory”

Definition: Clock Memory

Clock memories are memories which periodically change their binary states at fixed frequency in a pulse-pause ratio 1:1. Eight fixed frequencies are defined for C7, which can be allocated to any arbitrary memory byte. The period times can be found in Table 3-1.

Clock Period Times

Figure 3-1 shows the period times and the corresponding clock frequencies generated by the “clock memory byte”.

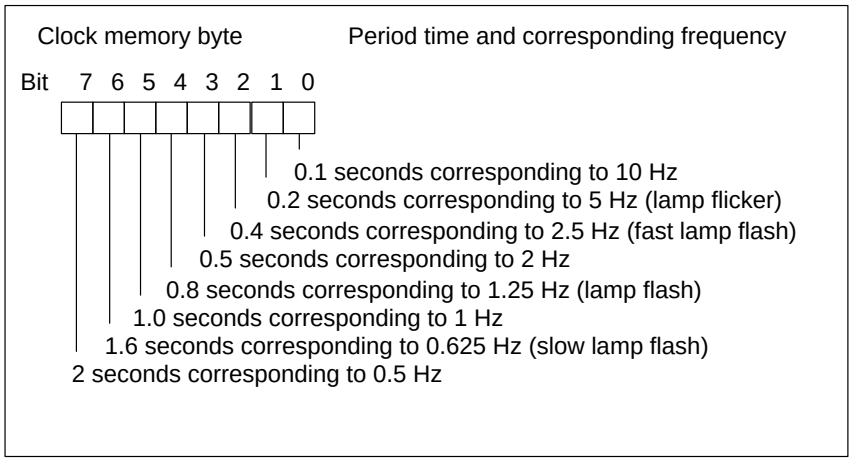


Figure 3-1 Clock Period Times in “Clock Memory Byte”

Parameter Block “Clock Memory”

Table 3-6 lists the parameters of the parameter block “Clock memory”.

Table 3-6 Parameter Block “Clock Memory”

Parameter	Explanation	Value Range C7-CPU	Default Setting
Clock memory	For “clock memory = yes”, a memory byte must be defined	Yes/No	No
Memory byte	Memory byte that should be used for the “clock memory byte”	From 0 to 255	-

3.5.2 Parameter Block “Start-Up Characteristics”

Parameter Block “Start-Up Characteristics”

The Table 3-7 lists the parameters of the parameter block “Start-up characteristics”

Table 3-7 Parameter Block “Start-up Characteristics”

Parameter	Explanation	Value Range	Default Setting
Self-test after POWER ON and erase	For “Self-test at cold restart = yes”, the C7-CPU tests its internal RAM after every POWER ON	Yes/no	Yes
Start-up • Manual	Only Restart can be set for the C7-CPU.	Restart	Restart
Restart • Automatic			
Module time limit- • Parameter assignment during start-up (in ms)	Maximum time for the “distribution” of the parameters to the modules within the module mounting sub-rack	from 1 to 10,000	100
Module time limit • Ready message after POWER ON (in ms)	Maximum time for the ready message of all modules after POWER ON If the modules do not transmit a ready message to the C7-CPU within this time, then the C7-CPU reverts to STOP.	from 1 to 65.000	65000

Note: You should assign the highest values to the parameters for the “Module time limits” if you are not sure of the required times in the C7.

3.5.3 Parameter Block “System Diagnostics”

Definition: System Diagnostics

System diagnostics perform the acquisition, evaluation and reporting of an error within the automation device. The cabling to the process is also included in the system diagnostics so that, for example, “wire breakage” can be recognized by the system diagnostics.

Example

Examples for errors that can be identified, evaluated and reported by the system diagnostics are:

- Errors in the user program
- Failure of hardware modules
- Breaks in wiring to transducers

Parameter Block “System Diagnostics”

Table 3-8 lists the parameters of the parameter block “System Diagnostics”

Table 3-8 Parameter Block “System Diagnostics”

Parameter	Explanation	Value Range	Default Setting
Extended diagnostics buffer entries	For “Extended diagnostics buffer entries = yes”, then the C7-CPU enters not only the error events into the diagnostics buffer but also all OB calls.	Yes/No	No
Transmission of diagnostics messages after reversion to the STOP mode	For “Transmission of diagnostics messages.... = yes”, then the C7-CPU transmits the cause of STOP via the multipoint (MPI) interface to the display system (programming device, OP). This diagnostic message is the “youngest” entry in the diagnostics buffer.	Yes/No	Yes

Undetected Errors

Errors that occur in the process, that is outside the automation system, are not detected by the system diagnostics. Such errors are for example “motor failure”. These errors fall within the area of the process error diagnostics.

3.5.4 Parameter Block “Retentive Areas”’

Definition: Retentivity

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

The following can be retentive:

- Flags
- S7 timers
- S7 counters
- Data areas

Retentivity Without Buffer Battery

Areas that you declare to be “retentive areas” in the parameter block will be preserved without a buffer battery after a power failure and after a transition from STOP to RUN. The boundary defined for the retentive and non-retentive areas is not influenced by the use of a buffer battery in the C7.

Retentivity of Data Blocks

You must take note of the following for the retentivity of data areas in data blocks:

- C7 with buffer battery: all data blocks are retentive. This means that a setting for the retentivity of DBs is not effective so long as the buffer battery voltage is sufficient.
- C7 without buffer battery: the user program is located in the flash memory:
  - The data blocks for which a retentive area has been set must be programmed in the flash memory.

    Data blocks or data areas that you generate with the SFC 22 “CREATE_DB” are not retentive.
  - The retentive data areas are preserved after a power failure. The contents programmed in the flash memory are held in the non-retentive data areas.

Parameter Block "Retentive Areas"

Table 3-9 lists the parameters of the parameter block "Retentive areas". The retentive area for all areas (flags, timers, counters and data bytes) may not be larger than for the sum of all the parameters listed in Table 3-9.

Table 3-9 Parameter Block "Retentive Areas"

Parameter	Explanation	Value Range C7	Default Setting
Memory bytes	The parameter value entry is the number of retentive memory bytes from memory byte 0	0 to 256	16
S7 timers	The parameter value entry is the number of retentive S7 timers from timer 0 (space req.: 2 bytes/timer)	0 to 128	0
S7 counters	The parameter value entry is the number of retentive S7 counters from counter 0 (space req.: 2 bytes/timer)	0 to 64	8
Data areas • Data block number	Max. 8 data areas may be retentive with a maximum of 4096 bytes. The start address of the data area + the number of data bytes may not exceed 8191.	Yes/No • From 1 to 127	No • 1
• Number of bytes		• From 0 to 4096	• 0
• Byte address (start address of the data area)		• From 0 to 8191	• 0
Sum of all retentive data		4736 bytes	

3.5.5 Parameter Block "Process Alarms"

Parameter Block "Process Alarms"

Table 3-10 lists the parameters of the parameter block "Process alarms".

The priority of the process alarm OB40 cannot be altered.

Table 3-10 Parameter Block "Process Alarms"

Parameter	Explanation	Value Range	Default Setting
Priority OB40	The priority of the OB40 cannot be altered.	16	16

3.5.6 Parameter Block “Real Time Clock”

Set Clock

You set the C7-CPU clock using STEP 7 or via the SFC 0 “SET_CLK” in the user program (see Appendix A and Reference Manual **/235/**.

Parameter Block “Real Time Clock”

Table 3-11 lists the parameters of the parameter block “Real time clock”

Table 3-11 Parameter Block “Real-Time Clock”

Parameter	Explanation	Value Range	Default Setting
Synchronization: on K bus	The synchronization of the real-time clock is performed via the K bus	None as master	None
Synchronization: on the MPI	Not possible	None	None
Synchronization: interval	Interval during which the real-time clock is synchronized.	Default Seconds 10 seconds Minute 10 minutes Hour 12 hours 24 hours	No synchronization
Correction factor	A deviation of the real-time clock is compensated with the correction factor within 24 hours. Example: if the real-time clock is slow by 4 ms after 24 hours, then you must set a correction factor of “+4 ms”. Example: If the real–time clock is two seconds slow after seven days, the correction factor is to be calculated as follows: 2 seconds : 7 days = 286 ms/day;consequently, you have to set a correction factor of +286.	From - 10000 to + 10000	0

3.5.7 Parameter Block “Time-of-Day Alarms”

Overview

The C7-CPU can trigger Time-of-Day alarms which you can activate and assign parameters via the parameter block “Time-of-Day alarms”.

Priority

The priority of the OB 10 is fixed at the value 2. You cannot change this value.

Parameter Block “Time-of-Day Alarms”

Table 3-12 lists the parameters of the parameter block “Time-of-Day alarms”

Table 3-12 Parameter Block “Time-of-Day Alarms”

Parameter	Explanation	Value Range	Default Setting
Active OB 10	Activation of OB 10	Yes/No	No
Execution OB 10	Here you set the execution intervals in which the time-of the day alarm should be triggered. The execution interval refers to the start date and the start time setting.	None Once only Every minute Every hour Every day Every week Every month Every year	None
Start date OB 10	The start date on which the time-of-day alarm should be triggered.	-	1994-01-01
Start time OB 10	The start time at which the time-of-day alarm should be triggered. The start time can only be entered in hours and minutes.	-	00:00:00

3.5.8 Parameter Block “Watchdog Alarms”

Overview

A watchdog alarm is a periodic signal that the C7-CPU generates internally and leads to the automatic call-up of a “Watchdog alarm OB” (OB 35).

Priority

The priority of the OB 35 has a fixed setting of 12. You cannot change this value.

Parameter Block “Watchdog Alarms”

Table 3-13 lists the parameters of the parameter block “Watchdog alarms”.

Table 3-13 Parameter Block “Watchdog Alarms”

Parameter	Explanation	Value Range	Default Setting
Periodicity of the OB 35 (in ms)	Call-up interval of the OB 35	from 1 to 60000	100

3.5.9 Parameter Block “Cycle Behavior”

Parameter Block “Cycle Behavior”

Table 3-14 list the parameters of the parameter block “Cycle behavior”

Table 3-14 Parameter Block “Cycle Behavior”

Parameter	Explanation	Value Range	Default Setting
Cycle loading due to communication (via the MPI) (in %)	To limit the extent of “slowing down” program processing due to communications processes, you can define the maximum percentage of cycle loading. The communication between C7-CPU and PG or between communicating C7-CPUs can slow down due to the limitation of the cycle loading. Operating system services such as the collection and provision of data for the communication will not be influenced. Functions that require uninterruptible reading of data ”slow down” program execution irrespectively of the value set for this parameter. Example: status block, reading of system data (*STEP 7*).	From 5 to 50	20
Maximum cycle time (in ms)	If the cycle time exceeds the “maximum cycle time”, then the C7-CPU reverts to the STOP mode if no 0B80 is provided for error handling. The maximum cycle time can be exceeded due to e.g.: • communications processes • accumulation of error events • errors in the user program (e.g. “continuous loops”)	From 1 to 6000	150
Cycle loading due to self-test (in s)	For “Cyclic self-test ≠ 0”, then the C7-CPU tests its internal RAM during the program cycle. This self-test consumes cycle time. You can define the time by which the program cycle may be lengthened in multiples of 10 µs (“0” = no cyclic self-test).	From 0 to 65000	0

3.5.10 Parameter Block “MPI Addresses”

Multipoint Interface MPI

The characteristics of the multipoint interface MPI of the C7-CPU can be assigned parameters with the parameter block “MPI addresses”. You need only process this parameter block if several C7 or S7-300 are networked via the multipoint interface MPI.

Values after Memory Reset

The parameters of the parameter block “MPI addresses” have a specialty: the parameter values are preserved even after a memory reset! Reason: the communications ability of an “erased” C7-CPU to the external world must be preserved even after a memory reset.

MPI Addresses C7

The C7 occupies two MPI addresses:

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

Parameter Block “MPI Addresses”

Table 3-15 lists the parameters of the parameter block “MPI addresses”

Table 3-15 Parameter Block “MPI Addresses”

Parameter	Explanation	Value Range	Default Setting
Highest MPI address	The definition of the highest MPI address in the network is necessary since: • Every (network) station is addressable • The communications process operates effectively Note: issue only as many MPI addresses as are necessary. You will then reduce the communications times. The “highest MPI address” must be the same for all network stations.	15 31 63 126	15
C7-CPU MPI address	Each station that is networked via the MPI must possess an address. The specified address may only be allocated once in the network. The C7-OP possesses its own MPI address (default = 1).	From 2 to 126	2 1

Note

The MPI is the only communications interface of the C7. Parameters should only be modified with utmost care.

3.6 Calculation of the Scan Cycle Time and Response Time of the C7-CPU

In this Section

This section explains how the cycle and response times of an user program are structured.

Use the programming device to read the cycle time of the user program on the C7 CPU (see programming manual **/280/**).

We will show you how to calculate the cycle time using an example.

The response time is of greater interest, however. This chapter includes a detailed description of how to calculate the response time.

Definition: Cycle Time

The cycle time is the time that passes during a program cycle.

Elements of the Cycle Time

The cycle time comprises the following elements:

- Process image transfer time (PII and PIQ)
- Operating system run time
- User program processing time

Diagram 3-2 shows the elements of the cycle time

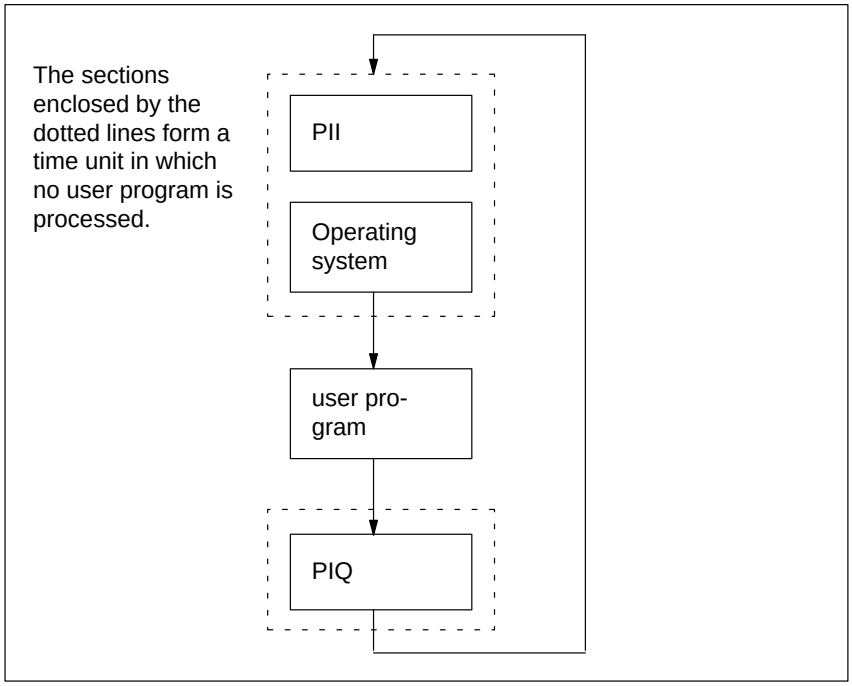


Figure 3-2 Elements of the Cycle Time

Definition: Response Time

The response time is the time that elapses between the recognition of an input signal and the change to the associated output signal.

Factors

The response time is dependent upon the following factors:

- Process image transfer time
- Operating system run time
- user program processing time
- Communications via the MPI
- Inherent delays in the inputs and outputs

Variation Range

The actual response time lies between a shortest and a longest response time. During the configuration of your system, you must always assume the longest response time.

The shortest and longest response times will be considered below in order that you can form a picture of the variation range of the response time.

Shortest Response Time

Figure 3-3 illustrates the conditions for the shortest response time.

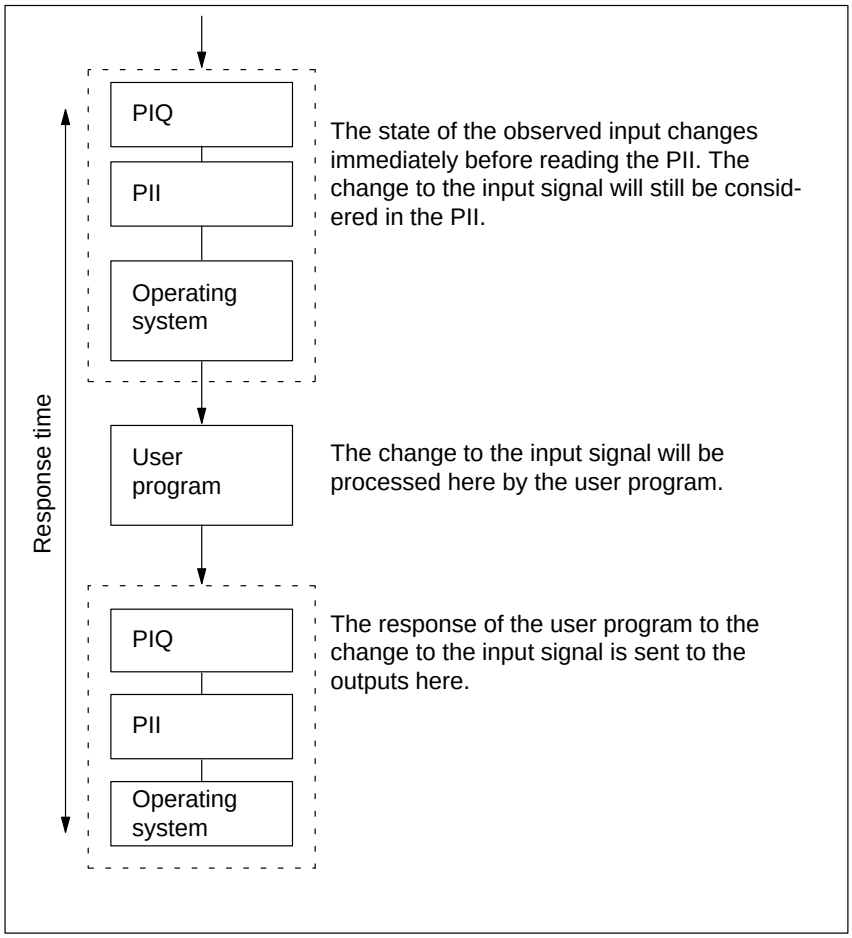


Figure 3-3 Shortest Response Time

Calculation

The (shortest) response time comprises the following:

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

The input delays are not marked in the diagram. Depending on the module, however, you have to take into account the following delay times:

- For digital inputs: input delay times
- For digital outputs: negligible delay times
- For analog inputs: cycle time of the analog input
- For analog outputs: response time of the analog output

Longest Response Time

Figure 3-4 describes the cirumstances under which the longest possible response time has to be taken into account:

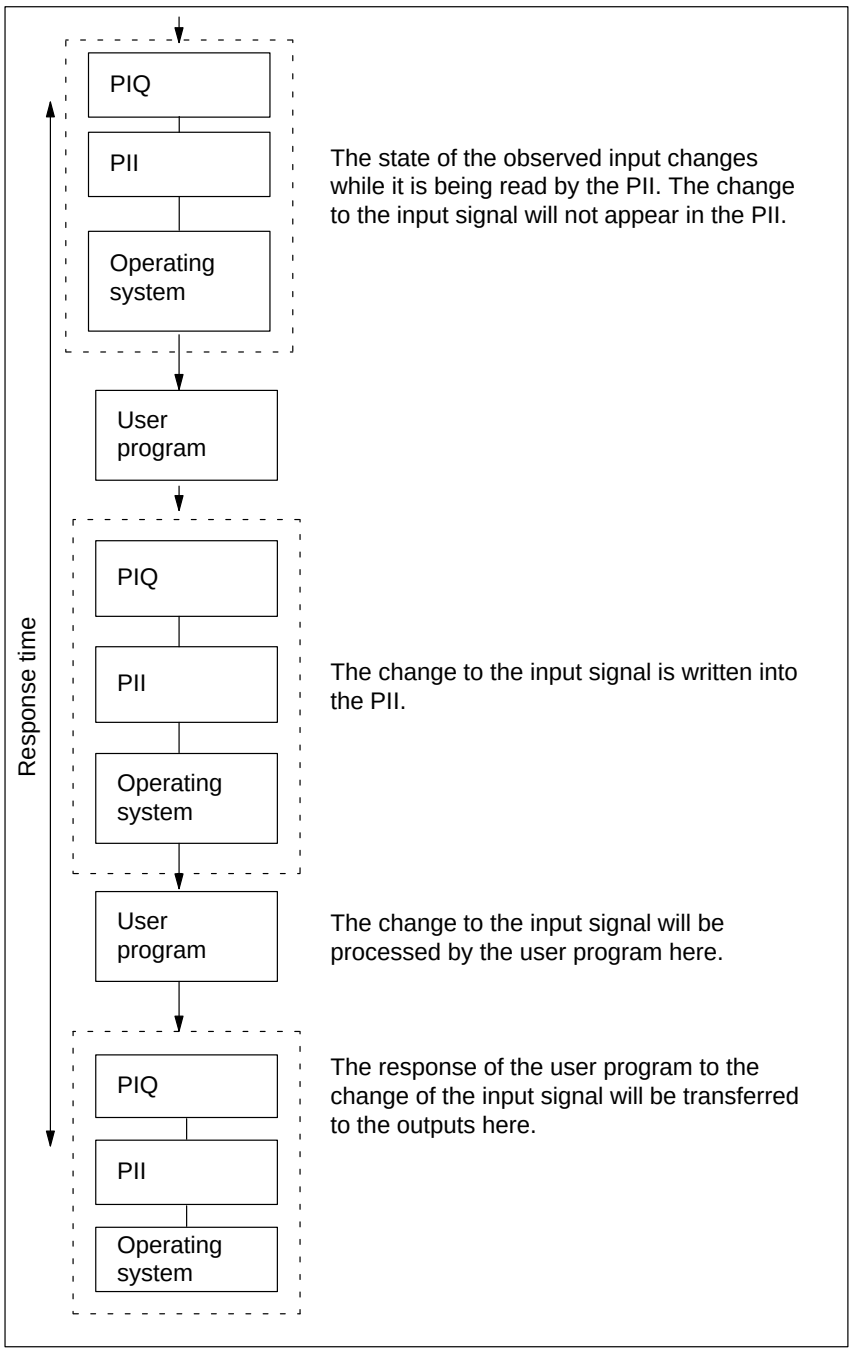


Figure 3-4 Longest Response Time

Calculation

The (longest) possible response time can be calculated as follows:

- 2 x process input image transfer time +
- 2 x process output image transfer time +
- 2 x operating system run time +
- 2 x program processing time +
- Processing time for S7 timers +
- Input and output delays

The input delays are not marked in the diagram. Depending on the module, however, you have to take into account the following delay times:

- For digital inputs: input delay times
- For digital outputs: negligible delay times
- For analog inputs: cycle time of the analog input
- For analog outputs: response time of the analog output

Extension of the Cycle Time:

As a basic principle, you must consider that the cycle time of an user program will be increased by:

- Time controlled processing
- Alarm processing
- Diagnostics and error processing
- Communications via MPI

Reduction of the Cycle Time

You will achieve faster response times by means of direct accesses to the I/Os by the user program, for example, with L PIB or T PQW. This method helps you avoid prolonged cycle times, as described in Figure 3-4.

Operating System Run Time

The operating system run time is caused by various sequences in the C7-CPU.

System Run Times

Table 3-16 lists all times relevant for determining the operating system run times of the C7-CPU.

Table 3-16 Operating System Run Times of the C7-CPU

Task	C7-CPU
Cycle control	Approx. 870 µs
Communications via the multipoint interface	Block functions (load/delete/copy) extend the cycle by max. 10 %. Communications via the MPI may load the cycle by up to 50 %, depending on the parameterization in STEP 7 (see Section 3.5.9).
Updating of the S7 timers	7 µs x number of timers currently running
C7-CPU self-test	From 0 to 65,000 µs; depending on parameterization (see Section 3.5.9)
Rack monitoring	Per rack: 50 µs

Process Image Update

Table 3-17 contains the C7-CPU times that are applicable to the process image update. The quoted figures are “ideal values”, which can be increased by alarms or communications via the MPI of the C7-CPU.

Table 3-17 Process image update of the C7-CPU

I/O Configuration	Process Image Update Times
1 C7 (16 bytes)	PII: approx. 200 µs PIQ: approx. 150 µs
C7 + 3 racks (64 bytes)	PII: approx. 850 µs PIQ: approx. 600 µs
C7 + 3 racks (128 bytes = max. configuration)	PII: approx.1480 µs PIQ: approx.1020 µs

3.6.1 Calculation Example for the Cycle Time

Component Parts of the Cycle Time

To recap: The cycle time consists of the following:

- Process image transfer time
- Operating system execution time
- User program execution time
- Processing time for S7 timers

Example

The user program in the C7 CPU has an execution time of 1.5 ms. Your user program uses four S7 timers.

In the C7 CPU, you have done without the C7 CPU self–test and, since the C7 CPU is a stand–alone unit, the cycle is not loaded with communication tasks.

Calculation

In the example, the cycle time is calculated from the following times:

- Process image transfer time
  - Process image input table: approx. 0.2 ms
  - Process image output table: approx 0.15 ms
- Operating system run time
  - Cycle control: approx. 0.87 ms
  - Rack monitoring: approx. 0.05 ms
- User program execution time: approx. 1.5 ms
- Processing time for S7 timers

  For four S7 timers, the single update takes

  4 x 7 us = 28 us = 0.03 ms. Adding the process image transfer time, the operating system run time and the user program execution time yields the time interval:

  0.2 ms + 0.15 ms + 0.87 ms + 0.05 ms + 1.5 ms = 2.77 ms. This time interval prolongs the execution time of the S7 timers:

$$Execution\ of\ S7\text{-}timer = \left(1 + \frac{2.77\ ms}{10\ ms}\right) \times 0.03\ ms\ =\ 0,04\ ms$$

The cycle time is the sum of the times listed:

Cycle time = 0.2 ms + 0.15 ms + 0.87 ms + 0.04 ms + 1.5 ms $\approx$ **2.8 ms.**

3.6.2 Calculation Example for the Response Time

Response Time

To recap, the response time is a total of :

- 2 x process input image transfer time +
- 2 x process output image transfer time +
- 2 x operating system run time +
- 2 x program execution time +
- Processing time for S7 timers +
- Input and output delay times

Example Configuration

It is based on a C7 control system including I/O.

Parameters of the CPU

It is based on the data from the calculation example for the cycle time.

Calculation

The response time for the example is as follows:

- Process image transfer time
  - Process input image (PII): = 2*0.2ms = **0.4 ms**
  - Process output image (PIQ): = 2*0.15ms = **0.3 ms**
- Operating system run time
  - Cycle control: = 2*0.87ms = **1.74 ms**
  - Rack monitoring: = 2*0.05ms =ca. **0.1 ms**
- User program execution time: **1.5 ms**
- **1st subtotal:** As a time basis for calculating the
  - Processing time for the timers

    the sum of all the times listed below is valid:

0.4 ms	(Process input image transfer time)	
+ 0.3 ms	(Process output image transfer time)	+
1.74	(Operating system run time)	+
0.1 ms	(Rack monitoring)	
+ 1.5ms	(User program execution time)	
≈**4.04 ms**.		

- Processing time for the S7 timers

  For four timers, the single update takes 4 x 7 µs ≈ 0.03 ms.
  The execution time for the timers is calculated as follows:

$$Processing\ time\ for\ S7\text{-}timer = \left(1 + \frac{4,04\ ms}{10\ ms}\right) \times 0,03\ ms\ \approx\ 0,04\ ms$$

- **2nd subtotal:** The response time **not including** the delay times of the inputs and outputs is calculated from the total of:

  4.04 ms (Result of the first subtotal)
  + 0.03 ms (Processing time for S7 timers)
  = 4.07 ≈ **5 ms** .

- Delay times for the inputs and outputs
  - The digital input of the C7 has an input delay of max. **4.8 ms** per channel
  - The delay time of the digital output of the C7 can be neglected.
  - The analog input of the C7 has a resolution of 12 bits. The conversion time per channel is approximately 0.5 ms. All four channels are supposed to be active. Calibration measurement must be enabled. The cycle time is thus **2.5 ms**.
  - The analog output of the C7 has a conversion time of **0.8 ms** per channel. A settling time for the ohmic load of 0.1 ms must be added to this. This yields a response time for the analog output of **0.9 ms**.
- Response times with delay times of the inputs and outputs:
- **Case 1:** An output channel is enabled when a digital input signal is read in. The response time is:

  Response time = 4.8 ms + 5 ms = 9.8 ms ≈ **10 ms**.

- **Case 2:** An analog value is read in and an analog value is output. The response time is:

  Response time = 2.5 ms + 5 ms + 0.9 ms = 3.9 ms ≈ **4 ms**.

3.6.3 Process Interrupt Response Time

Process Interrupt Response Time

The process interrupt response time is the time that elapses between the first occurrence of a process interrupt signal and the calling of the first instruction in the process interrupt OB (OB 40).

In general, higher–priority interrupts take precedence. That means the process interrupt response time is increased by the program execution time of the higher–priority process interrupt OBs and process interrupt OBs of the same priority, which have not yet been processed.

Calculation

The process interrupt response time is calculated as follows:

Process interrupt response time = Process interrupt response time of the C7 CPU + process interrupt response time of the signal module

C7 CPU

The process interrupt response time of the C7 CPU is 1.1 ms.

Signal Modules

The process interrupt response time of the signal modules is calculated as follows:

- Digital input modules

  Process interrupt response time = Internal interrupt preparation time + input delay

  The times are listed in the data sheet for the respective digital input module

- Analog input modules

  Process interrupt response time = Internal interrupt preparation tme + conversion time

  The internal interrupt preparation time of the analog input modules can be neglected. Please refer to the data sheet of the respective analog input module for conversion times.

Process Interrupt Handling

Process interrupt handling begins when the process interrupt OB (OB 40) is called. Higher–priority interrupts cause the process interrupt handling routine to be interrupted. Direct accesses to the I/O are made at the execution time of the instruction. When the process interrupt handling routine has finished, either cyclic program execution continues or further interrupt OBs of higher priority or the same priority are called up and executed.

Calculation Example

To recap, the process interrupt response time is composed of:

- The process interrupt response time of the CPU
- The process interrupt response time of the signal module

Example: In the example, the C7 I/O is to be used exclusively.

Calculation

For the example, the process interrupt response time is calculated from the following times:

- Process interrupt response time of the C7 CPU: approx. 1.1 ms
- Process interrupt response time of the module:
  - Internal interrupt preparation time: 0.6 ms
  - Input delay: 0.01 ms

The process interrupt response time is calculated from the sum of the times below:

Process interrupt response time = (Zahlen wie dt. , Dez.punkte)

This process interrupt response time elapses from the time a signal is present at the digital input until the first instruction in OB 40 is processed.

3.6.4 Diagnostics Interrupt Response Time

Diagnostics Interrupt Response Time

The diagnostics interrupt response time is the time that elapses between the first occurrence of a diagnostics interrupt signal and the calling of the first instruction in the diagnostics interrupt OB (OB 82).

In general, higher–priority interrupts take precedence. That means the diagnostics interrupt response time is increased by the program execution time of the higher–priority interrupt OBs and interrupt OBs of the same priority, which have not yet been processed.

Calculation

The diagnostics interrupt response time is calculated as follows:

Diagnostics interrupt response time = Diagnostics interrupt response time of the C7 CPU + diagnostics interrupt response time of the signal module.

For C7: 0.6 ms

C7 CPU

The diagnostics interrupt response time of the C7 CPU is approximately 1.3 ms.

Calculation Example

1.3ms + 0.6ms = 1.9ms ≈**2 ms**

3.7 Test and Reference Data Functions of the C7-CPU

Overview

The C7-CPU offers the possibility to scan the status of the C7-CPU and its associated signal modules by means of a series of test and reference data functions. By this method, you can obtain information on the following:

- The current layout of the C7
- The current configuration
- The current states
- The current sequences

in the C7-CPU and the associated signal modules.

You can also change process variables independently from the user program

Description of the Test and Reference Data Functions

The test and reference data functions can only be executed using the STEP 7 software. The description of the test and reference data functions can be found in the appropriate chapters of the manual **/230/**.

List of Test and Reference Data Functions

Table 3-18 contains the reference data functions of the C7-CPU.

Table 3-18 Reference data functions of the C7-CPU

Reference Data Function	Application
User memory	Display of the current loading of the: • Constants memory (EEPROM), integrated into the C7 • Load memory of the C7-CPU (RAM) • Work memory of the C7-CPU (RAM)
Blocks	Display of all available blocks and the possible priority levels • SFCs • SFBs • OBs • All blocks
Stacks	Read out the contents of • B stack • I stack • L stack

Table 3-18 Reference data functions of the C7-CPU, Continued

Reference Data Function	Application
Communication	Display of • Number of links • Message frame length • Transmission rate via the MPI • Reserved OP links • Reserved programming device links • Free links
Time system	Display of following values • C7 time • C7 date • Time system • Correction factor • Cycle of the synchronization frames
Cycle times	Display of the cycle times of the user program • Monitoring time • Length of the longest cycle • Length of the shortest cycle • Length of the last cycle
Read diagnostics buffer	Display the contents of the diagnostics buffer • Date and time of the day of occurrence of an error event • Designation of the error event • Information describing the event more exactly; e.g. error OB call for access errors
C7-CPU data	Display of the following information for a C7 • C7 type and version of the C7-CPU • Size of the work and load memories in the C7-CPU • Layout of the load memory • Number and area for the inputs, outputs, times, counters and flags • Area for local data • C7 system behavior

List of Test Functions

Table 3-19 lists the test functions of the C7-CPU.

Table 3-19 Test Functions of the C7

Test functions	user
Status variable	Observe selected process variables (inputs, outputs, memory bits, times, counters, data) at a specified position in the user program.
Monitor variable	Assign a value (cycle start, cycle end, transition RUN → STOP) to selected process variables (inputs, outputs, memory bits, times, counters, data) at a specified position and thus control the user program directly.
Block status	Observe a block with regard to the program sequence to aid commissioning and fault finding. Block status offers the possibility of observing certain register contents during the execution of statements , e.g. AKKUs, address register, status register, DB register.

3.8 Load/Erase C7-CPU Flash Memory

Overview

During transmission of an user program to the C7-CPU, it is transmitted only to the load memory and not automatically into the C7-CPU flash memory also.

The contents of a C7-CPU flash memory are not automatically reset during a memory reset of the C7-CPU**.**

You must explicitly initiate these actions.

Load user Program into Flash Memory

You must explicitly load the C7 flash memory using the STEP 7 function “copy RAM to ROM”. You do this after you have copied the program and data into the load memory. In this way, the whole contents of the load memory is transmitted to the flash memory.

Erase Flash Memory

After the memory reset, the load memory is reinitialized with the contents of the flash memory. However, since you can change the contents of the flash memory only using the STEP 7 function “copy RAM to ROM”, you must proceed in the following manner to erase the flash memory:

1. Select all OBs, FBs, FCs and DBs on the C7-CPU.

2. Use STEP 7 to erase the load memory of the CP–CPU; remove/delete files. **The function ”Direct PLC: delete” may not be used in this case since a memory a reset must be carried out.**

3. Activate the STEP 7 function “copy RAM to ROM”. In this way, the empty load memory will be ”loaded” into the flash memory. Afterwards, this is also empty.

Addressing, Parameter Assignment and Function of the C7 Input/Output

4

In this Chapter

4.1 Slot-Oriented Address Assignment for Signal Modules

Overview

The relationship between card slot and address assignment is described below. You require this information to determine the start addresses of the employed C7 modules.

C7 I/O and the customer specific C7 modules always occupy the rack Number 0.

C7 Slots

The logical slots 2 and 3 are reserved for the C7-CPU and the integrated IM360.

The I/O incorporated in the C7 occupies the logical slots 4 (digital I/O) and 5 (analog I/O and universal inputs).

Slots for Customer-Specific Module

The logical slots 6 to 11 can be occupied by a customer-specific module.

C7 Layout

Figure 4-1 shows an example of a C7 with a customer-specific module and the connection of an additional S7-300.

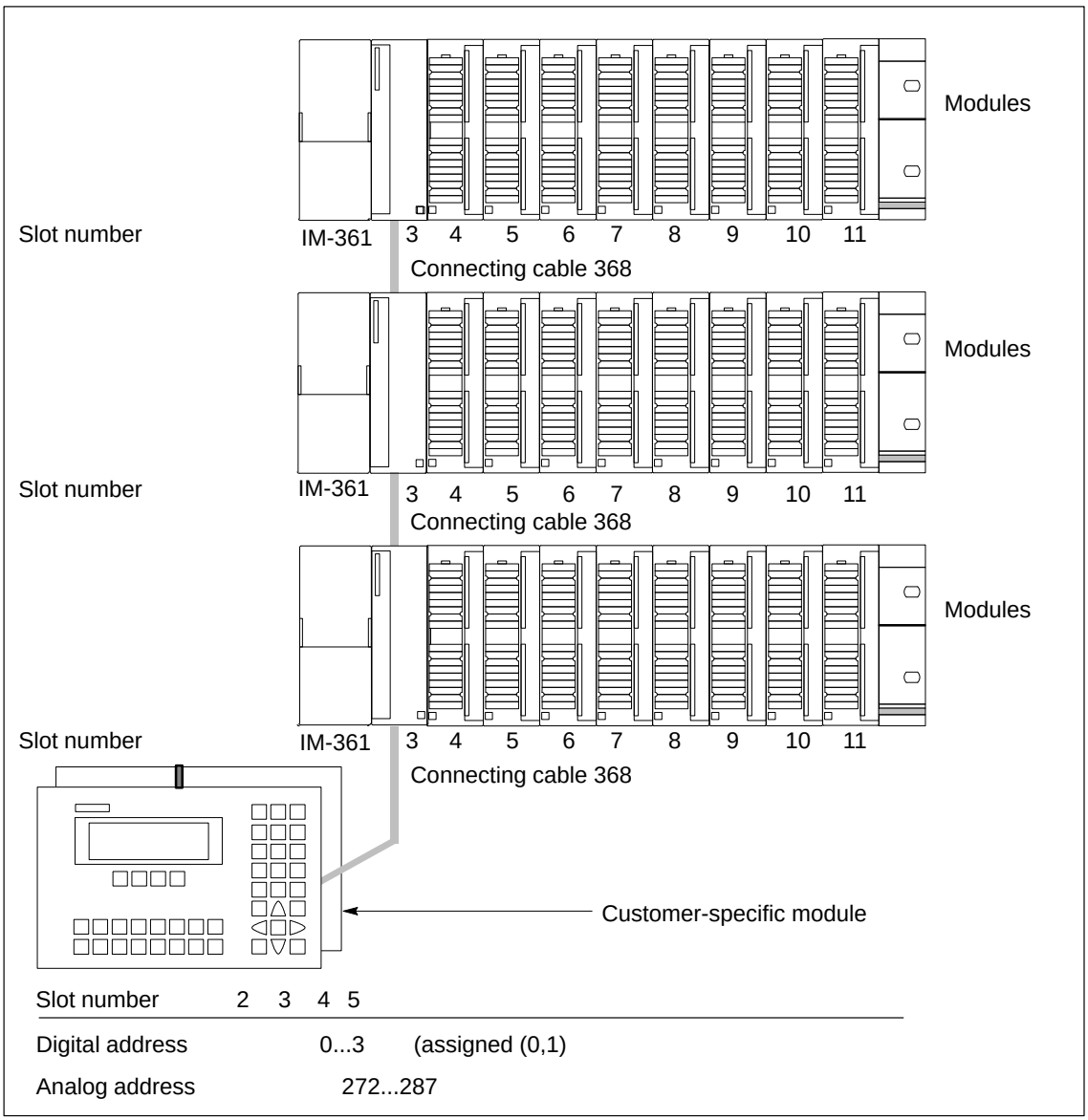


Figure 4-1 Example of Slots in C7

Additional S7-300 Racks

You can connect max. three S7-300 racks to a C7. How these are connected is described in the S7-300 Installation and Hardware Manual.

4.2 Addressing the C7 I/O

Overview

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

Figure 4-2 shows the scheme for addressing the individual channels of the digital I/O.

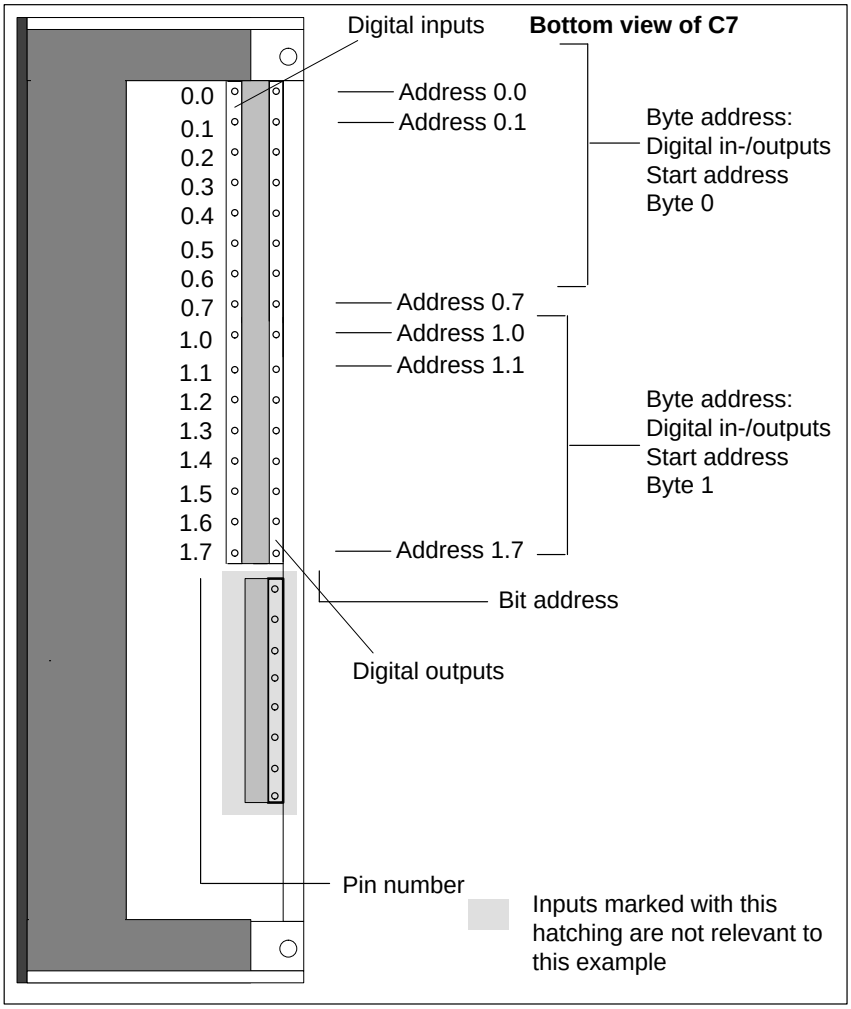


Figure 4-2 Digital Input/Output Addresses

4.3 Use and Function of C7 Analog I/O

In this Section

This section contains:

- Descriptions of the basic terminology for analog value processing.
- How to address and assign parameters to analog I/O.
- How you allocate measurement ranges to analog input channels.
- The behavior of the individual analog input channels and the analog output channel.

4.3.1 Addressing Analog I/O

Analog Function Addresses

The address of an analog channel is always a word address.

An analog input/output has the same start address for the analog input and output channels.

Figure 4-3 shows which channel addresses result. You can see that for the analog I/O, the analog input channel and the analog output channel are addressed starting from the same address.

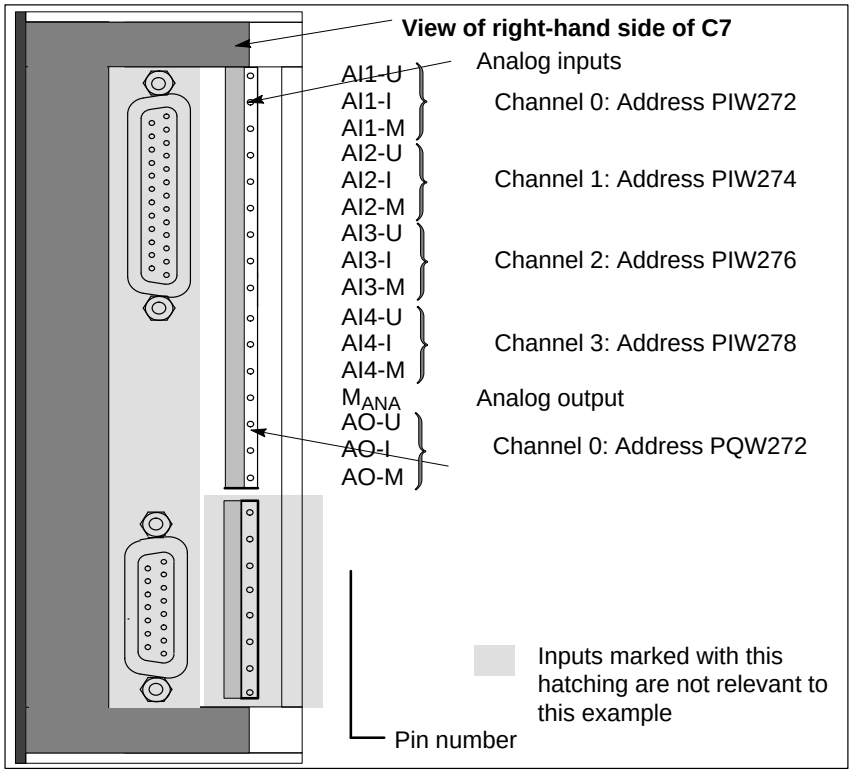


Figure 4-3 Analog Input/Output Addresses

4.3.2 Parameter Assignment for Analog I/O

Overview

This chapter contains an overview of the analog I/O and their parameters.

Parameter Assignment

You set the parameters for the analog I/O using the S7 *Hardware configuration*. A parameter block is generated that contains all selected peripheral parameters. After loading the SDB parameter, the parameters are not immediately transferred to the analog I/O. The C7-CPU then transfers the parameters to the respective analog I/O after every mode change from STOP→RUN.

Alternatively, you can also change some parameters in the application program with the SFC 55 to 57 (see Reference Manual **/235/**).

We subdivide the parameters for the two configuration alternatives into:

- Static parameters and
- Dynamic parameters

The following table explains when the static and dynamic parameters are adopted.

Table 4-1 Time of Transfer of the Parameters from the C7-CPU to the Analog I/O

Parameter	Set with	Time of Parameter Transfer
Static	*Hardware configuration*	STOP -> RUN
Dynamic	*Hardware configuration*	STOP -> RUN
	SFC 55 to 57	RUN

Assignment of Parameters for Characteristics

The following parameter blocks permit the assignment of parameters in S7 *Hardware configuration* to define the following characteristics of the analog I/O:

- For inputs
  - Basic setting
  - Diagnostics
  - Measurement
  - Alarm cycle
- For outputs
  - Basic setting
  - Diagnostics
  - Substitute values
  - Output range

Analog Input Parameters

Table 4-2 provides an overview of the analog input parameters.

Table 4-2 Analog Input Parameters

Parameter	**Analog Output**	
	Value Range	**Preset Value**
Basic setting		
• Enable diagnostic alarm	Yes/No	No
Diagnostics		
• Enable	Yes/No	No
– Configuration/parameter error		
– Wire break (only 4...20 mA)		
– Range undershoot		
– Range overshoot	Yes/No	No
• Wire break test (only for measurement range 4...20 mA)		
Measurement		
• Type of measurement	Deactivated Voltage Current	Voltage
• Measurement range	±10V ±20mA 4..20mA	±10V
Alarm cycle		
• Alarm	Yes/No	No
• Alarm time	Unsolicited, 3ms, 3.5ms, 4ms, 4.5ms....16ms	16ms

Alarm Cycle

If parameters are assigned to this mode of analog inputs, then:

- A measurement cycle will be processed (Channel 1, 2, 3, 4)
- The measured value of each measured channel will be made available to the application program for collection
- After expiry of the alarm time in the C7-CPU, a process alarm will be triggered and a new measurement cycle initiated.

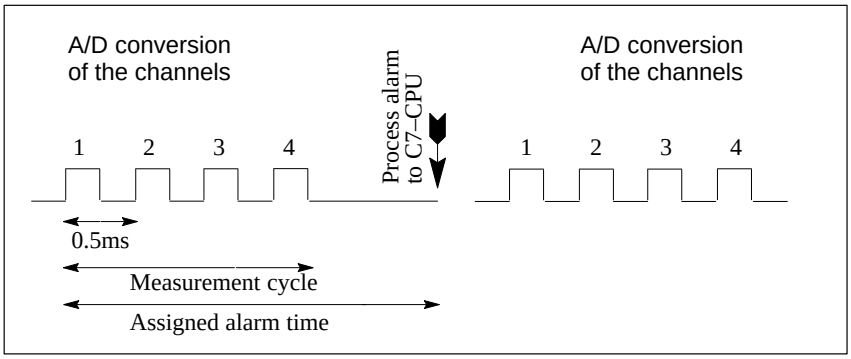


Figure 4-4 Sequence of the Measurement of all 4 Channels with an Assigned Measurement cycle.

If individual measurement channels are deactivated, then the measurement cycle will be correspondingly shorter.

Parameter Characteristics of Analog Inputs

Table 4-3 shows which parameters

- Are static or dynamic
- Can be set for all or individual analog inputs.

Table 4-3 Parameter Characteristics of Analog Inputs

Parameter	Static/Dynamic	Effective Range
Enable diagnostic alarm	Static	Analog inputs/ Analog output/ Universal inputs
Enable diagnostics	Static	Channel
Wire break test	Static	Channel
Type of measurement	Dynamic	Channel
Measurement range	Dynamic	Channel
Alarm cycle	Dynamic	Analog inputs

Analog Output Parameters

Table 4-4 provides an overview of the analog output parameters.

Table 4-4 Analog Output Parameters

Parameter	Analog Output	
	Value Range	Preset Value
Basic setting • Enable diagnostic alarm	 Yes/No	 No
Diagnostics • Enable – Configuration/parameter error – Substitute value switched on	 Yes/No	 No
Substitute value • Retain last value • Value	 Yes/No 9400_H...$6C00_H$	 No 0
Output range • Type of output • Output range	 Deactivated Voltage Current ±10V ±20mA 4..20mA	 Voltage ±10V

Parameter Characteristics of Analog Outputs

Table 4-5 shows which parameters

- Are static or dynamic
- Can be set.

Table 4-5 Parameter Characteristics of Analog Outputs

Parameter	Static/Dynamic	Effective Range
Enable diagnostic alarm	Static	Analog inputs / Analog output/ Universal inputs
Enable diagnostics	Static	Output
Substitute value • Maintain last value • Value	 Dynamic Dynamic	 Output Output
Type of output	Dynamic	Output
Output range	Dynamic	Output

4.3.3 Representation of Analog Values

Overview

The representation of analog values, or an analog value in binary form, is the same for all C7 analog inputs and analog outputs.

This chapter describes the analog values for **all** measurement ranges or output ranges that can be used with the C7 analog I/O.

Representation of Analog Values

The digitized analog value is the same for input and output values of the same nominal range.

The representation of analog values is performed as a two's complement

Table 4-6 illustrates the representation of the analog I/O:

Table 4-6 Representation of Analog Values

Resolution	Analog Value															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Weighting of the bits	VZ	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 Conventions

The sign of the analog value is always contained in bit number 15:

- "0" → +
- "1" → –

Resolution 12 Bit

The resolution is 12 bit. The analog value is entered into the ACCU left justified. The unoccupied low significance places are written with **"0"**.

Table 4-7 contains an example of a bit pattern showing how the unoccupied places for 12 bit resolution are filled with **"**0**"**.

Table 4-7 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 (incl. sign)	0	1	0	0	0	1	1	0	0	1	1	1	**0**	**0**	**0**	**0**

4.3.4 Representation of Analog Values for the Measurement Ranges of the Analog Inputs

Overview

The tables in this chapter contain the digitized analog values for the measurement ranges of the analog inputs.

In Table 4-8, you will find the representation of the binary analog values and the associated decimal or hexadecimal representation of the units of the analog values.

How the Measured Value Tables Should be Read

The tables 4-9 contain the digitized analog values for the various measurement ranges.

Since the binary representation of the analog values is always the same, the tables contain only the comparison of the measurement ranges to the units.

These tables are therefore clearer and easier to read. The corresponding binary representation of the measured values can be referred to in Table 4-8.

Measured Value Resolution

The bits identified with ”x” are not relevant to a resolution of 12 bits.

Table 4-8 Possible Resolutions of Analog Values

Resolution in Bit (incl. Sign)	Units		Analog Value	
	Decimal	Hexadecimal	High Byte	Low Byte
12	8	8_H	VZ 0 0 0 0 0 0 0	0 0 0 1 x x x x

Voltage and Current Measurement Ranges

Table 4-9 contains the representation of the digitized voltage measurement ranges for ± 10V and the digitized current measurement ranges ± 20 mA, 4...20 mA.

Table 4-9 Representation of the Digitized Measured Value of the Analog Inputs (Voltage and Current Measurement Ranges)

Measuring Range ± 10 V	Measuring Range ± 20 mA	Measuring Range 4..20mA	Units		Range
			Decimal	Hexadecimal	
≥ 11.759	≥ 23.516	0	≥ 32512	$7F00_H$	Overflow
11.7589	23.515	22.81	32511	$7EFF_H$	Upper range
:	:	:	:	:	
10.0004	20.0007	20.005	27649	$6C01_H$	
10	20.000	20.000	27648	$6C00_H$	Nominal range
7.500	14.998	:	20736	5100_H	
:0	:0	4.000	:0	$:0_H$	
– 7.500	- 14.998	3.9995	–20736	$AF00_H$	
– 10	– 20.000	0	–27648	9400_H	
– 10.0004	– 20.0007	Underflow range	–27649	$93FF_H$	Lower range
:	:		:	:	
– 11.759	– 23.516		–32512	8100_H	
≤ – 11.76	≤ – 23.517		≤ –32513	≤ $80FF_H$	Underflow

4.3.5 Representation of Analog Values for the Output Range of the Analog Outputs

Table for Output Ranges

Table 4-10 contains the analog output ranges of the analog output.

Voltage /Current Output Ranges

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

Table 4-10 Representation of the Analog Output Range of the Analog Outputs (Voltage/Current Output Ranges)

Output Range ± 10 V	Output Range 4 ..20 mA	Output Range ± 20 mA	Units		Range
			Decimal	Hexadecimal	
0	0	0	≥ 32512	≥ $7F00_H$	Overflow
11.7589	22.81	23.515	32511	$7EFF_H$	Upper range
:	:	:	:	:	
10.0004	20.005	20.0007	27649	$6C01_H$	
10.0000	20.000	20.000	27648	$6C00_H$	Nominal range
:	:	:	:	:	
0	4.000	0	0	0_H	
0	3.9995		:	:	
	0				
:	0	:	– 6912	$E500_H$	
			– 6913	$E4FF_H$	
			:	:	
–10.0000		– 20.000	– 27648	9400_H	
10.0004			– 27649	$93FF_H$	Lower range
		:	:	:	
–11.7589		23.515	– 32512	8100_H	
0		0	≤ – 32513	≤ $80FF_H$	Underflow

4.3.6 Conversion and Cycle Time of Analog I/O

Introduction

You can find the definitions and relationships between conversion time and cycle time for the analog inputs in this chapter.

Conversion Time

The conversion time consists of the basic conversion time and an additional conversion time necessary for the input calibration.

Cycle Time

The analog-digital conversion and the transfer of the digitized measured value to the C7-CPU is performed sequentially, i.e. the analog input channels are converted one after the other. The cycle time, i.e. the elapsed time before an analog input value is converted again, is the sum of all conversion times (0.5ms/channel) of all activated analog input channels plus one calibration measurement. Unused analog input channels should be deactivated in ”S7 Configuration” in order to reduce the cycle time.

Figure 4-5 illustrates in overview, how the cycle time for a 4-channel analog input is composed.

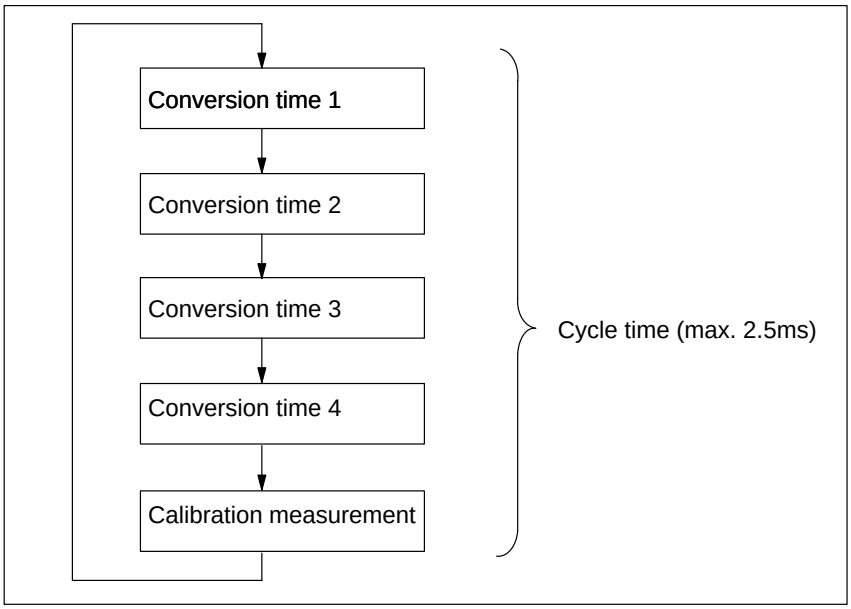


Figure 4-5 Cycle time of an Analog Input

Alarm Cycle

If the alarm cycle mode is parameterized, the new measuring cycle is not started unless the timed interrupt is initiated (see Section 4.3.2).

4.3.7 Conversion, Cycle, Settling and Response Times of Analog Outputs

Introduction

This chapter contains the definitions and relationships between relevant times for the analog output.

Conversion Time

The conversion time includes the acceptance of the digitized output value from the internal memory and the digital-analog conversion.

Cycle Time

The cycle time, i.e. the elapsed time before an analog output value is converted the next time is equal to the time for the conversion of the one analog output.

Settling Time

The settling time (t_2 to t_3), that is the elapsed time between the creation of the converted value and the attainment of the specific value at the analog output, is load-dependent. One must differentiate between resistive, capacitive and inductive load.

Response Time

The response time, that is the interval between providing the digital output value and attaining the specified value at the analog output, is between 100 µs and 2ms.

4.3.8 Behaviour of Analog I/O

Overview

This section decribes:

- The relationship of the analog input and output values to the supply voltages of the analog I/O and the operating modes of the C7.
- The behaviour of the analog I/O in relationship to the position of the analog values in the respective value range.
- The influence of errors on the analog I/O.

Influence of the Supply Voltage and the Operating Mode

The input and output values of the analog I/O are dependent upon the supply voltage of the analog I/O and the operating mode of the C7.

The triggering of a diagnostic alarm is dependent upon the parameter assignment.

Table 4-11 gives an impression of these inter-relationships.

Table 4-11 Dependencies Between Analog Input and Output Values upon the Operating State of the C7 and upon the Supply Voltage L+

Operating state of C7		Input value of the analog input	Output value of the analog output
POWER ON	RUN	Process value	C7 value
	STOP	Process value	Substitute value or retain last value (configurable)
POWER OFF	STOP	–	0 signal

Influence of the Value Range on the Input

The behaviour of the analog input is dependent upon the area of the value range in which the input value lies. Table 4-12 illustrates these dependencies for the analog input values.

Table 4-12 Behaviour of the Analog Input in Relation to the Position of the Analog Input Value in the Value Range

Process Value lies in	Input Value	Diagnostics	Alarm
Nominal range	Process value	–	–
Upper range/lower range	Process value	–	–
Overflow/underflow	$7FFF_H$	Message [1]	Diagnostic alarm[1]

KEIN MERKER According to parameter assignment.

Influence of the Value Range for the Output

The behaviour of the analog output is dependent upon the position of the output value within the value range. Table 4-13 illustrates this dependency for the analog output values.

Table 4-13 Behaviour of the Analog Input in Relation to the Position of the Analog Input Value in the Value Range

Output Value lies in	Output Value	Diagnostics	Alarm
Nominal range	C7 value	–	–
Upper range/lower range	C7 value	–	–
Overflow/underflow	0-signal	–	–

Influence of Errors

Errors lead to a diagnostics message and diagnostics alarm if diagnostics parameters have been assigned (see Section 2 of the Manual, Chapters 4.3.1 and 5).

4.3.9 Time Alarm/Alarm Cycle

Alarm Cycle

If the alarm cycle mode is parameterized, the new measuring cycle is not started unless the time alarm is initiated (see Section 4.3.2).

Parameterizable Events

Use the STEP 7 function *Hardware configuration* for parameter assignment.

Process Alarm OB

If a process alarm is transferred from the I/O to the C7 CPU, the process alarm OB (OB 40) is called in the C7 CPU. The event which called OB 40 is stored in the start information (declaration section) of OB 40. You have to evaluate the additional information Z1 to Z3 in the start information.

Declaration Section of OB 40

The entries in the declaration section of OB 40 are listed in Table 4–14. The bytes relevant to the user are hatched in the Table.

Table 4-14 Declaration Section of OB 40

Byte	Mearning		Mearning	Byte
0	Class	Identifier	Event number	1
2	Execution level		Current OB number	3
4	Data identifiers-Z2/3		Data identifier-Z1	5
6	Additional information Z1			7
8	Additional information Z2			9
10	Additional information Z3			11
12	Time tag of event			13
14				15
16				17
18				19

Additional Information Z1

Additional information Z1 contains the initial address of the C7 I/O module (bytes 6/7).

Address: 272 or 0110_H

Additional Information Z2

Bit 4 of byte 8 = 1 in the case of an end–of–cycle alarm.

Additional Information Z3

Additional information Z3 is not used and assigned the value 0000H.

Evaluation in the User Program

The evaluation of process alarms in the user program is described in the Manual /**234**/.

4.4 Use and Function of the Universal Inputs

In this Section

In this chapter you can find:

- Basic terminology about the function of the universal inputs
- How you can use the universal inputs
- How you can address and assign parameters to the universal inputs

4.4.1 Addressing Universal Inputs

Overview

You can select the following functions by assigning the appropriate parameters to the universal inputs:

- Digital input
- Alarm input
- Counter
- Frequency counter
- Period time counter

Universal Input Addresses

The addresses for the universal inputs are default addresses which cannot be changed. According to the application of the universal inputs, the results occupy differing addresses.

For the address allocation, differentiation is made in:

- Input range PIW280 ... PIB287 for count values or signal state of the digital inputs
- Output/control range PQW 274...PAB282 for counters

Input Range

The 4 universal inputs of the input range (see Figure 4-7) have the following addresses and weightings:

Table 4-15 Input Address of the Universal Inputs

Address	Designation	
PIW280	**CI1**: Counter input	
PIW282	**CI2**: Counter input	
PEB284	**CI3**: Counter input	Frequency/period time counter
PEB285		
PEB286	—	
PEB287: Bit 0	Current state of universal input 1	
Bit 1	Current state of universal input 2	
Bit 2	Current state of universal input 3	
Bit 3	Current state of universal input 4	
Bit 4	States of the count inputs see Table 4-15	
Bit 5		
Bit 6		
Bit 7	—	

States of the Inputs

The state of the individual inputs is stored as a bit pattern in PEB287:

Table 4-16 State of the Inputs

Address PEB287	State Indication of Universal (universal) Inputs
Bit 0	Bit = 1: universal input 1 set. Bit = 0: universal input 1 reset.
Bit 1	Bit = 1: universal input 2 set. Bit = 0: universal input 2 reset
Bit 2	Bit = 1: universal input 3 set. Bit = 0: universal input 3 reset.
Bit 3	Bit = 1: universal input 4 set. Bit = 0: universal input 4 reset.
Bit 4*)	Bit = 1: Counter1 counting Bit = 0: Counter1 stopped
Bit 5*)	Bit = 1: Counter2 counting Bit = 0: Counter2 stopped
Bit 6*)	Bit = 1: Counter3 counting Bit = 0: Counter3 stopped

*) Only relevant when universal inputs assigned as count input

Output Range

If the universal inputs are used as counters, then the behaviour of the counters is controlled via the output range.

Table 4-17 Addresses and Weighting of the Output Range of the Count Inputs

Address		Control of Counter 1..3
PQW274		Start/comparison value counter 1*
PAB276:	Bit 0	0 = Counter1 disable 1 = Counter1 enable
	Bit 1	0 = New start/comparison value not valid
		1 = Set new start/comparison value
PQW277		Start-/comparison value counter 2*
PAB279:	Bit 0	0 = Counter2 disable 1 = Counter 2 enable
	Bit 1	0 = New start/comparison value not valid
		1 = Set new start/comparison value
PQW280		Start-/comparison value counter 3*
PAB282:	Bit 0	0 = Counter3 disable 1 = Counter 3 enable
	Bit 1	0 = New start/comparison value not valid
		1 = Set new start/comparison value

*) Initial value for down counter, comparison value for up counter

Diagram with Universal Inputs

The pins and associated addresses are shown in the diagram.

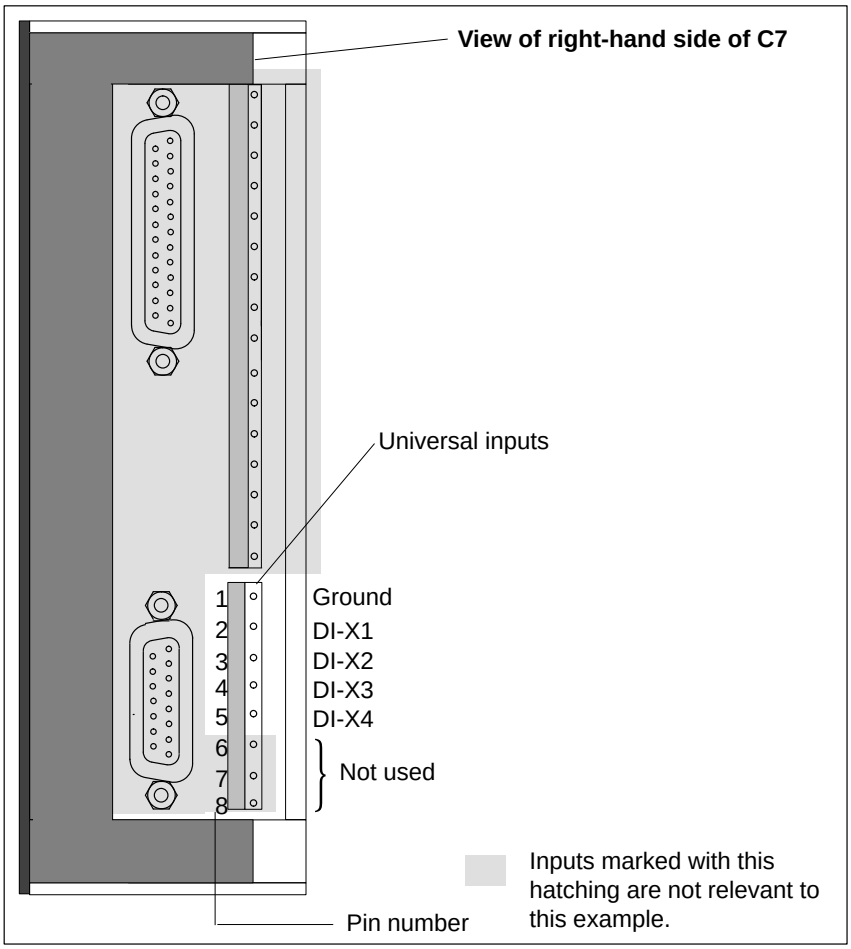


Figure 4-6 Pins of Universal Inputs

4.4.2 Parameter Assignment of Universal Inputs

Parameter Block Universal Inputs

In parameter block ”universal inputs” you set the parameters for:

- The alarm inputs
- The counters
- The frequency meter/period time counter
- Digital input (always if alarm or count input = deactivated)

Parameter Assignment

You set the parameters for the universal inputs using the STEP 7 function *Hardware configuration*. A parameter block is generated which contains all currently selected parameters of the universal inputs. After loading this parameter block, the C7-CPU then transfers the parameters to the appropriate universal inputs at every transition from STOP → RUN.

Alarm Inputs

If the universal inputs are used as alarm inputs, a process alarm will be triggered at the C7-CPU for the assigned rising or falling edge at the input. The default is the rising edge.

Counter Inputs

The universal inputs 1...3 can be assigned as:

- Count input
- Period time counter (input 3 only)
- Frequency meter (input 3 only)

The counter values are made available to the application program as 16 bit values and the frequency and period time counter as 24 bit values.

Table 4-18 lists the parameters for the above mentioned functions:

Table 4-18 Parameter Block of the Count Inputs

Parameter	Explanation	Value Range	Default Setting
Count input 1	Define the count direction	Forwards Backwards	Forwards
	Select count edge to be used for counting	Rising Negative	Rising
	Counter can trigger a process alarm after reaching the comparison value (when connting up) or zero transition (when counting down)	Yes No	No
Count input 2	Define the count direction	Forwards Backwards	Forwards
	Select count edge to be used for counting	Rising Negative	Rising
	Counter can trigger a process alarm after reaching the comparison value (when connting up) or zero transition (when counting down)	Yes No	No
Count input 3	Activate the count input and assignment of the counting type	Deactivated Counter Frequency counter Period time counter	Deactivated
	If counter activated then define the count	Forwards Backwards	Forwards
	If counter activated then define the edge to be used for counting	Rising Falling	Rising
	If counter activated, then the counter can trigger a process alarm after reaching the comparison value (when connting up) or a zero transition (when counting down)	Yes No	No
	If frequency counter selected, then select the gate time for the frequency counting	0.1s 1s 10s	1s
	No further parameters for period time counter	–	–

Digital Inputs

If the universal inputs are deactivated in the parameter block (default setting), then the inputs react as digital inputs. However, no automatically updated process image will be made available to the application program for these inputs. The current state of the input can only be read by means of a direct peripheral access. (See Table 4-14 or 4-15 for address).

4.4.3 Alarm Inputs

Introduction

If universal inputs are used as alarm inputs, then a process alarm will be generated every time the corresponding (assigned) edge at one of the inputs.

Assignable Events

The parameter assignment can be performed with STEP 7 Function *Hardware configuration.*

Process Alarm OB

If a process alarm is sent from the I/O to the C7-CPU, then the process alarm OB (OB 40) will be called by the C7-CPU. The event that called the OB 40 is stored in the start information (declaration section) of the OB 40. You must evaluate the additional information C1 to C3 in the start information.

Declaration Section of OB 40

The entries in the declaration section of OB 40 can be found in Table 4-19. The bytes that are relevant to the user are hatched in the table.

Table 4-19 Declaration section of OB 40

Byte	Meaning		Meaning	Byte
0	Class	Identifier	Event number	1
2	Priority level		Current OB number	3
4	Data identifier- C2/3		Data identifier- C1	5
6	Additional information C1			7
8	Additional information C2			9
10	Additional information C3			11
12	Time stamp of event			13
14				15
16				17
18				19

Additional Information Z1

The start address of the C7 peripheral module is contained in the additional information Z1 (byte 6/7).
Address: 272 or 0110_H

Additional Information Z2

The serial number of the universal inputs that triggered the process alarm can be found in byte 8 of the additional information Z2. Byte 9 is irrelevant.

You can find the additional information broken down into bits in Figure 4-7

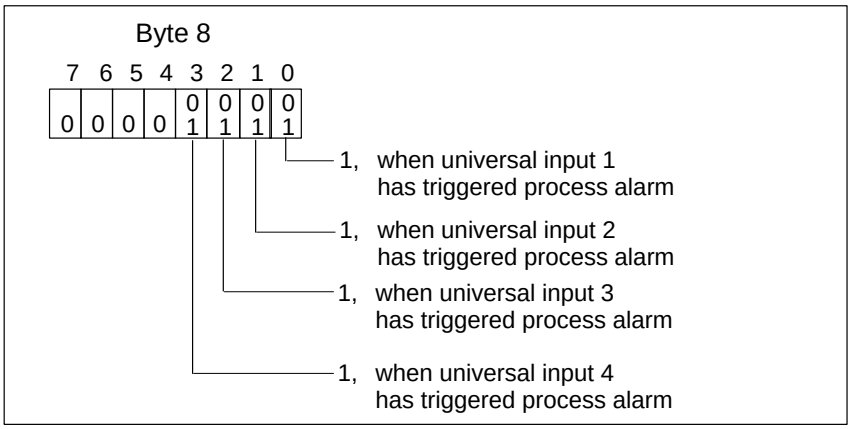


Figure 4-7 Addidtional Information Z2

Additional Information Z3

Additional information Z3 is not used and is set to 0110_H.

Evaluation in Application Program

The evaluation of process alarm in the application program is described in the Manual **/120/**.

4.4.4 Counters

Counter

The counter calculates the value of the count from the count pulses (forwards or backwards).

You can assign parameters with the STEP 7 Function *Hardware configuration* for:

- Count pulse is triggered by a rising or falling edge at the corresponding universal input
- Whether counting should be forwards or backwards

Actual Value of Counter

The counter calculates the actual value according to the following formula:

Actual value (forward counter) = number of edges

or

Actual value (backward counter) = start value ÷ number of edges

Forward Counting

Forward counting starts at zero or continued from the last counter value and until the selected comparison value is reached. The start value after resetting the counter is always zero. The comparision value is set by the user program.

Backward Counting

Backward counting starts counting backwards from the selected start value or is continued from the last counter value until the value zero has been reached. Start values are set by the application program.

Exceeding the Threshold Frequency

The universal input counter counts count pulses up to a maximum frequency of 10 kHz.

A frequency filter is fitted to the inputs.

Warning

If the actual frequency exceeds the threshold frequency of 10kHz, then the correct function of the universal inputs can no longer be guaranteed, since count pulses will be lost.

Stop and Start Counters

The universal counter inputs are controlled by the application program.

The control possibilities you have in the application program to influence the counters are explained in Table 4-20.

Table 4-20 Control of Counters by the Application Program

Aim	Procedure
Start counter	• Enter a valid start value (if backward counting) or a valid comparison value (PQW274, PQW277, PQW280). • Activate the new start/comparison value (Bit 1 of PQB276, PQB279, PQB282 rising edge ‘0‘→‘1‘) • Start the counter by selecting the start bit (Bit 0 of PQB276, PQB279, PQB282 falling edge ‘0‘→‘1‘
Stop counter	• Reset the start bits (Bit 0 of PQB276, PQB279,PQB282 falling edge ‘0‘→‘1‘)
Restart counter with counter initialization (reset)	• If necessary, enter a new start value or retain old start value (for backward counting) or comparison value (for forward counting) (PQW274, PQW277, PQW280). • Activate the new start/comparison value (Bit 1 of PQB276, PQB279, PQB282 rising edge ‘0‘→‘1‘). • Set the start bit (Bit 1 of PQB276, PQB279, PQB282 rising edge ‘0‘→‘1‘).
Restart counter without counter initialization (counter continues counting without a reset)	• No setting of the new start/comparison value • Start the counter by setting the start bit (Bit 0 of PQB276, PQB279, PQB282 rising edge ’0’→’1’).
Select new start/comparison value	• Enter new start/comparison value (PQW274, PQW277, PQW280). • Set value (Bit 1 of PQB276, PQB279, PQB282 rising edge ‘0‘→‘1‘. – New start/comparison value will be activated with the next rising edge at the count input – If forward counting running: new comparison value will be accepted – If backward counting running: new start value will be accepted, current count value will be corrected by the difference.

Table 4-20 Control of Counters by the Application Program (Continued)

Aim	Procedure
Initialization of the counter (start of a new counting process) always occurs:	• At zero transition (backward counting) or comparison value reached/exceeded (forward counting) • After setting the enable bit in the data area (Bit 0 of PQB276, PQB279, PQB282 rising edge '0'→'1'), if the bit "set new start/comparison value" is set simultaneously (Bit 1 of PQB276, PQB279, PQB282).
Generation of process alarm and resetting of counter	• Precondition is process alarm = yes of the counter has been assigned • When counting direction is forwards, if count value = comparison value • When counting direction is backwards, if count value = zero

4.4.5 Counter Alarms

Introduction

The universal inputs counters can be assigned parameters for process alarm. In this case, a forward counter triggers a process alarm when it reaches the comparison value, and a backward counter when it passes through zero.

Assignable Events

The parameter assignment is performed with the STEP 7 Function *Hardware configuration.*

Process Alarm OB

If a process alarm from the counter is sent to the C7-CPU, then the process alarm OB (OB 40) to the C7-CPU will be called. The event that called the OB 40 is stored in the start information (declaration section) of the OB 40. You must evaluate the additional information C1 to C3 in the start information.

Declaration Section of OB 40

The entries in the declaration section of OB 40 can be found in Table 4-21: the bytes that are relevant to the user are hatched in the table.

Table 4-21 Declaration Section of OB 40

Byte	Meaning		Meaning	Byte
0	Class	Identifier	Event number	1
2	Priority level		Current OB number	3
4	Data identifier- Z2/3		Data identifier- Z1	5
6	Additional information Z1			7
8	Additional information Z2			9
10	Additional information Z3			11
12	Time stamp of event			13
14				15
16				17
18				19

Additional Information Z1

The start address of the C7 peripheral module is contained in the additional information Z1 (byte 6/7).
Address: 272 or 0110_H

Additional Information Z2

The serial number of the universal inputs that triggered the process alarm can be found in byte 8 of the additional information Z2. Byte 9 is irrelevant.

You can find the additional information broken down into bits in Figure 4-8.

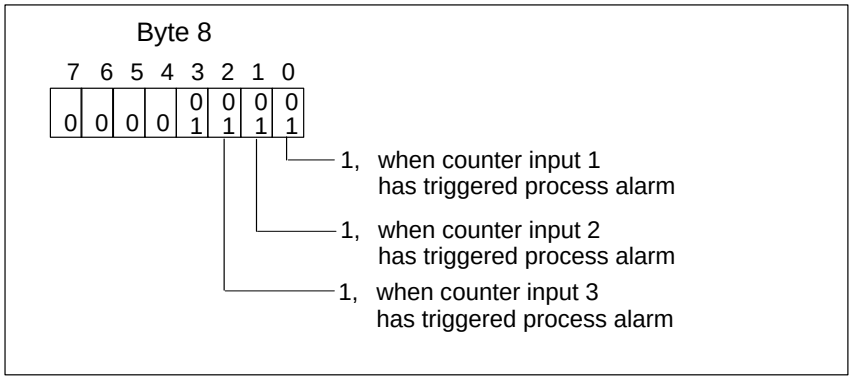


Figure 4-8 Layout of Additional Information Z3 in Declaration Section of OB 40

Additional Information Z3

Additional information Z3 is not used and is set to 0110_H.

Evaluation in Application Program

The evaluation of process alarms in the application program is described in the Manual **/280/**.

4.4.6 Frequency Counters

Overview

The universal input 3 (assigned as frequency meter) provides you with the possibility to continuously count identical edges within a selected time period for a frequency $\leq$ 10 kHz.

Application

Calculation of high frequencies.

Frequency Calculation

The frequency meter calculates the frequency value from the measured value and the measurement period.

The signal to be measured is to be connected to the universal input 3 (see Chapter 4.4.1 and 4.4.2) of the C7. The frequency counter counts the rising edges of the signal to be measured within a time period that can be assigned using parameters.

From this, the application program can derive the actual frequency using the following formula:

$$\text{Frequency} = \frac{\text{Number of positive edges}}{\text{Measurement period}}$$

Measurement Period

The measurement period can be selected using the STEP 7 Function *Hardware configuration.* You can choose between the measurement periods 0.1s, 1s or 10s. The measurement process is immediately restarted after the measurement period has elapsed, so that an updated frequency counter value is always available.

Example of Frequency Calculation

The measurement period is 1 s. During a measurement period, 6500 rising edges of the signal to be measured are counted. The value 6500 is made available to the application program.

$$\text{Frequency} = \frac{6500}{1\ \text{s}} = 6500\ \text{Hz}$$

Frequency During First Measurement Period

After starting up the C7, OB 1 is processed and the universal input frequency meter is automatically started.

The 1st valid frequency is calculated after the 1st measurement period. Before the end of the 1st measurement period, the frequency counter value $FFFFFF_H$ is available in the C7 CPU.

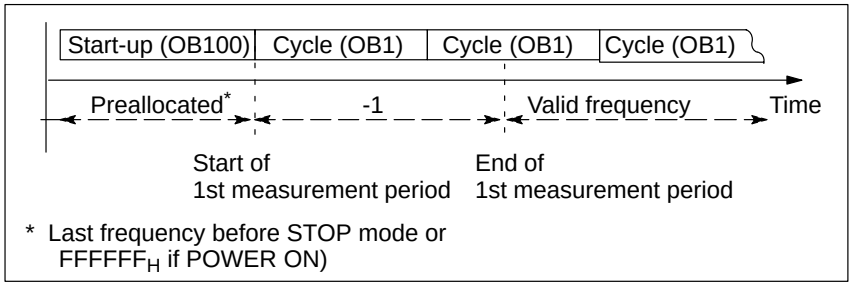


Figure 4-9 Frequency During First Measurement Period

Exceeding the Threshold Frequency

The universal inputs frequency meter is designed for a maximum frequency of 10 kHz.

A frequency filter is fitted to the input.

Warning

If the actual frequency exceeds the threshold frequency of 10 kHz, then the correct function of the universal inputs can no longer be guaranteed, since count pulses will be lost.

Resolution of Measurement

With relatively constant frequencies, the resolution of the measurement is higher if you set a longer measurement period. Table 4-22 displays the resulution of the measurement according to the configured measurement period.

Table 4-22 Resolution of the Measurement

Measurement Period	Resolution	Example of Count Value During Measurement	Frequency (Calculated)
0,1 s	Frequency can be calculated in 10 Hz increments	900	9000 Hz
		901	9010 Hz
1 s	Frequency can be calculated in 1 Hz increments	900	900 Hz
		901	901 Hz
10 s	Frequency can be calculated in 0.1 Hz increments	900	90 Hz
		901	90.1 Hz

Disadvantage of Long Measurement Periods

The frequency meter calculates the frequency in longer intervals. This means that with long measurement periods, an updated frequency value is more seldom available. If the frequency continuously changes, then only average values are available.

Disadvantage of Short Frequency

Due to the principle of measurement, the measurement error increases with a reduction in the measured frequency.

4.4.7 Period Time Measurement

Overview

The universal inputs 3 can be assigned as period time counter. This universal input reads pulses from a transducer. The transducer could, for example, be fitted to the barrel extruder of an injection moulding machine.

Application

Calculation of low frequencies and rotation speeds.

Principle

The period time counter counts the number of increments (fixed time intervals) of t_{zi} = 0.5µs between two rising edges. The first period starts at the first transition from ”0” to ”1” (rising edge). It ends at the next rising edge. This is also the start of the next period.

From this, a period time can be calculated:

t_p = number of counted increments * 0.5µs

In addition, for every rising edge, a counter is started that increases its value by 1 every 0.5µs until the next positive edge occurs.

The period time counter can be defined with a resolution of 0.5µs.

Explanation of Principle Based upon a Simple Transducer

Diagram 4-10 illustrates a simple transducer. The transducer delivers ”1” when the light passes through one of the slots in the disc. If the discs rotates, then the transducer delivers the signal shown in the diagram.

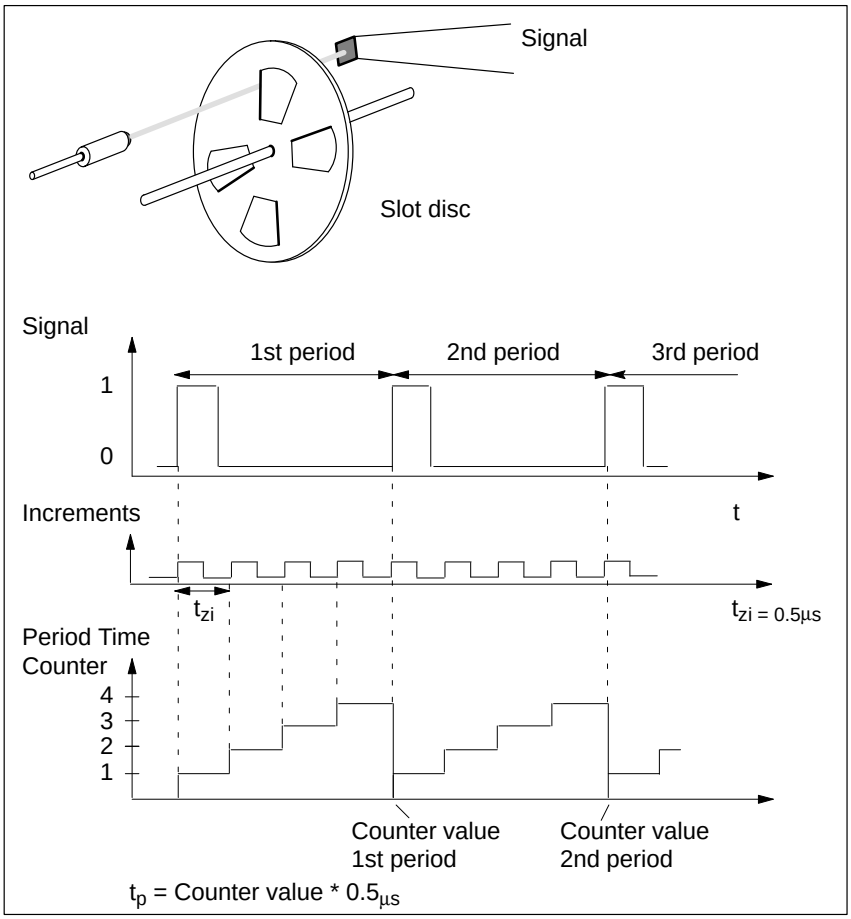


Figure 4-10 Simple Transducer, for example, a Slot Disc on a Shaft

If you know the number of pulses that are delivered by the transducer for each revolution of the barrel extruder, then you can calculate the speed with which the barrel extruder is rotating. An example follows:

N = 16 pulses are generated per revolution of the barrel extruder (N is also known as the slot number of the transducer). The interval between 2 pulses is 50000 increments (fixed time interval). The rotational speed of the barrel extruder is calculated as follows:

$$v = \frac{1}{N \text{ x } ti} = \frac{1}{16 \text{ x } 50,000 \cdot \text{x } 0.5\ \mu s} = 2.5\frac{1}{s} = 150\frac{rev}{min}$$

Lower Threshold

The period time counter generates a 24-bit counter value. These 3 bytes can represent values up to FF FF FF_H (16777214 decimal). From this, the lower threshold frequency for N = 1 is (when taking into account the period time stated below (t_p = 8.39 s)):

$$fu = \frac{1}{tp} \; ; \qquad tp = 16777214 * 0.5\mu s = 8.39s$$

$$fu = 0.119Hz$$

And for N = 1, the lower threshold rotation speed

$$v = \frac{1}{N \text{ x } ti} = \frac{1}{1 \text{ x } 8.39s} = 0.119\frac{1}{s} = 7.14\frac{rev}{min}$$

Upper Threshold

The upper threshold frequency results from the condition that the universal inputs are designed for a maximum frequency of 10kHz. The minimum period time of 0.1 ms follows. Therefore the upper threshold frequency is 10 kHz (corresponding to 600,000 rev/min).

If this frequency is exceeded, then the input values will be erroneous, since individual pulses will be suppressed by the input filter (of 10 kHz).

The relative measuring discrepancy gets smaller as the period time increases.

Thresholds

These thresholds are applicable for a transducer that generates one pulse per revolution. If you use transducers that generate several pulses per revolution, then you must reconsider the threshold frequencies.

Counter Overflow

The counter value FF FF FF_H indicates a shortfall of the lower threshold. A diagnostic report will not be generated in this case.

Parameter Assignment

In order to use the universal input 3 as a period time counter, this must also be assigned as such (parameters). This is performed with the STEP 7 Function *Hardware configuration*.

4.5 Data Set Description for Parameter Block of C7 Analog I/O and Universal Inputs

Overview

If a reassignment is to be performed during normal process operations, then the validity and inter-relationships between the individual parameters must be examined by the application program.

Incorrect value ranges of the parameters can result in incorrect behaviour of the I/O. Table 4-23 lists the layout of the parameter data sets.

Table 4-23 Table with Data Set Descriptions Parameter Block

DS	Byte	Bit	Time Value	What Can Be Assigned	Meaning of the Respective Bits
0	00	0	0	Enable diagnostics AI1	0=No 1=Yes
		1	0	Enable diagnostics AI2	0=No 1=Yes
		2	0	Enable diagnostics AI3	0=No 1=Yes
		3	0	Enable diagnostics AI4	0=No 1=Yes
		4	0	Enable diagnostics AQ1	0=No 1=Yes
		5..7	0	—	
	01	0	0	Enable diagn. wirebreak AI1	0=No 1=Yes (only if measurement range 4..20mA)
		1	0	Enable diagn. wirebreak AI2	0=No 1=Yes (only if measurement range 4..20mA)
		2	0	Enable diagn. wirebreak AI3	0=No 1=Yes (only if measurement range 4..20mA)
		3	0	Enable diagn. wirebreak AI4	0=No 1=Yes (only if measurement range 4..20mA)
		4..7	0	—	
	02	0..7	00_H	Reserved	
	03	0	0	Enable diagnostics alarm for mod.	0=No 1=Yes
		1..7	0	–	
1	00	0..2	0	I1 Use	0=Disable (normal DI), 1=Alarm-DI, 2=CI
		3	0	Process alarm	0=No, 1=Yes (always with alarm-DI) (selectable if use = 2)
		4	0	Edge	0=Rising edge, 1=Falling edge (only if not deactivated)
		5	0	Direction	0=Forwards, 1=Backwards (only if CI)
		6..7	0		
	01	0..2	0	I2 Use	0=Disable (normal DI), 1=Alarm-DI, 2=CI
		3	0	Process alarm	0=No, 1=Yes (always with alarm-DI) (selectable if use = 2)

Table 4-23 Table with Data Set Descriptions Parameter Block (Continued)

DS	Byte	Bit	Time Value	What Can Be Assigned		Meaning of the Respective Bits
		4	0		Edge	0=Rising. edge, 1=Falling. edge (only if not deactivated)
		5	0		Direction	0=Forwards, 1=Backwards (only if CI)
		6..7	0			
	02	0..2	0	I3	Use	0=Disable (normal DI), 1=Alarm-DI, 2=CI, 3=FC, 4=Period time counter
		3	0		Process alarm	0=No, 1=Yes (always with alarm-DI) (selectable if use = 2)
		4	0		edge	0=Rising. edge, 1=Falling edge (only if use = 1 or = 2)
		5	0		Direction	0=Forwards, 1=Backwards (if use = 2)
		6..7	0		Gate time	0=0.1s, 1=s, 2=10s (if use = 3)
	03	0..2	0	I3	Use	0=Disable (normal DI), 1=Alarm-DI
		3	0		Process alarm	0=No, (always when deactivated), 1=Yes (always with alarm-DI) (selectable if use = 2)
		4	0		edge	0=Rising edge, 1=Falling. edge (only if not deactivated)
		5..7	00	—		
	04		0	Cycle time		0=16ms, 1=continuous (ca.2.5ms), 6=3ms, 7=3.5ms, 8=4ms..(0.5ms increments).
	05	0..1	1	AI1	Measurement type	0=Deactivated, 1=Voltage=, 2=Current
		2	0		Cycle-end alarm*)	0=No, 1=Yes (only if Byte 4 <>1)
		3	0		—	
		4..7	9		Measurement range	0=Deactivated, 3=4..20mA, 4=±20mA (if measurement type=current) 9=±10V (if measurement type=voltage)
	06	0..1	1	AI2	Measurement type	0=Deactivated, 1=Voltage, 2=Current
		2	0		Cycle-end alarm*)	0=No, 1=Yes (only if Byte 4 <>1)
		3	0		—	
		4..7	9		Measurement range	0=Deactivated, 3=4..20mA, 4=±20mA (if measurement type=current) 9=±10V (if measurement type=voltage)
	07	0..1	1	AI3	Measurement type	0=Deactivated, 1=Voltage, 2=Current
		2	0		Cycle-end alarm*)	0=No, 1=Yes (only if Byte 4 <>1)
		3	0		—	

Table 4-23 Table with Data Set Descriptions Parameter Block (Continued)

DS	Byte	Bit	Time Value	What Can Be Assigned		Meaning of the Respective Bits
		4..7	9		Measurement range	0=Deactivated, 3=4..20mA, 4=±20mA (if measurement type=current) 9=±10V (if measurement type=voltage)
	08	0..1	1	AI4	Measurement type	0=Deactivated, 1=Voltage, 2=Current
		2	0		Cycle-end alarm*)	0=No, 1=Yes (only if Byte 4 <>1)
		3	0		—	
	09	4..7	9		Measurement range	0=Deactivated, 3=4..20mA, 4=±20mA (if measurement type=current) 9=±10V (if measurement type=voltage)

*)These 4 bits are ORed during evaluation as only one analog measuring cycle is available.

4.6 Examples for Programming the Analog I/O and the Universal Inputs

Overview

The following examples for programming the analog I/O and the universal inputs of counters will help you to get familiar with the principles for programming the C7 I/O.

The three following examples are contained in this section:

- Block for scaling analog output values
- Block for scaling analog input values
- Block for programming the counters

4.6.1 Block for Scaling Analog Output Values

Function of Block

The FC127 block is used to convert the setpoint to be specified in a memory double word as floating point number to the corresponding hexadecimal pattern (=analog value) which must be output to a peripheral output word. For this purpose, a simple calculation using the rule of three 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 value. This is identical in the floating point number and in the hexadecimal representation.

2. Then the total range (RANGE_DEC), resulting from the difference (UL – LL) is calculated in hexadecimal representation, depending on the measuring range being unipolar or bipolar.

3. Now the percentage (PERCENT) calculated before is related to the total 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 offset.

5. The resulting bit pattern is output.

Summary of Formulae

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

FC127 Sequence of Statements

The FC127 function block contains the following statement lines:

```
FUNCTION FC 127: void

var_input
                lower limit: DWORD
                upper limit: DWORD
                setpoint: DWORD
end_var

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

BEGIN

//***Case: unipolar or bipolar measuring range?***
L lower limit;                  // lower limit negative?
L 0.0;                          // yes => bipolar measuring range
<R;
JC bipo;

L DW#16#0000_0000;              //unipolar range lower limit
T LL
JU comp;

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

//***Calculating the range /hexadecimal)***
rech: NOP 0;

L W#16#6C00;                    // upper limit for unipolar and bipolar
                                // range identical
ITD;
L LL;
–D;
T RANGE_HEX;                    // buffer difference

//*** Relating setpoint to total measuring range***
L upper limit;                  // compute range
L lower limit
–R;
T RANGE_DEC;
```

```
L setpoint;                          // relate setpoint to total
                                     // range
L lower limit;
–R;
L RANGE_DEC;
/R;
T PERCENT;

//***Computing hex pattern to be output***
L RANGE_HEX;                         // relate hex value to total range
DTR;
L PERCENT;
*R;
L LL;                                // add offset
DTR;
+R;
RND;                                 // convert floating point number to
                                     // 32–bit integer
T channel;                           // output result
```

Calling the FC127 in OB1

An example for calling the FC127 is given in the following:

Before calling the function block, the range limits and the setpoint must be reassigned to memory double words. This is necessary to enable using variable values. Normally, ”upper limit” and ”lower limit” are fixed values; the ”setpoint” is variable.

This can be achieved by setting the ”upper limit” and ”lower limit” parameters in the declaration section of the FC127 to ”REAL”. To enhance flexibility in a test environment, this variant has been omitted.

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     (
                                lower limit:=MD0,
                                upper limit:=MD4,
                                setpoint:=MD8,
                                channel:=PQW272
                );

END_ORGANIZATION_BLOCK
```


4.6.2 Block for Scaling Analog Input Values

Function of Block

The FC126 block is used to convert the actual value entered as hexadecimal number in a peripheral input word into a corresponding floating point number (=analog value) to be output to a peripheral output word. For this purpose, a simple calculation using the rule of three is programmed.

1. First, the actual value is related to the total range (RANGE_HEX) resulting from the difference (UL – LL).

   The result is a percentage of the absolute actual value. This is identical in the floating point number and in the hexadecimal representation.

2. Then the total range (RANGE_DEC), resulting from the difference (UL – LL) is calculated in floating point number representation, depending on the measuring range being unipolar or bipolar.

3. Now the percentage (PERCENT) calculated before is related to the total floating point number range.

   The result is the absolute read–in value.

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

5. The resulting floating point number is output.

Summary of Formulae

PERCENT=(channel – LL) / (UL – LL)
RANGE_HEX=UL – LL
actual value=PERCENT*(upper limit – lower limit) + lower limit

FC126 Sequence of Statements

The FC126 function block contains the following statement lines:

```
FUNCTION FC 126: void

var_input
                lower limit: DWORD
                upper limit: DWORD
                channel: DWORD
end_var

var_output
                actual value : DWORD
end_var

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

BEGIN

//***Case: unipolar or bipolar measuring range?***
L lower limit;                  // lower limit negative?
L 0.0;                          // yes=> bipolar measuring range
<R;
JC bipo;

L DW#16#000_00000;              // unipolar range lower limit
T LL
JU comp;
bipo: NOP 0;

L W#16#9400;                    // bipolar range lower limit
ITD;
T LL;

//***Computing the range (hexadecimal)***
comp:NOP 0;

L W#16#6C00;                    // upper limit for unipolar and bipolar
range
                                // identical
ITD;
L LL;
–D;
T RANGE_HEX;                    // buffer difference
```

```
//*** Relating actual value to total measuring range***
L channel;                    // relate input value to total
                              // range
ITD;
L LL;
–D;
DTR;
L RANGE_HEX;
DTR;
/R;
T PERCENT;

//***Computing floating point number***
L upper limit;                // calculate floating point number range
L lower limit;
–R;
L PERCENT;
*R;
L lower limit;
+R;
T actual value;

END_FUNCTION
```

Calling the FC126 in OB1

An example for calling the FC126 is given in the following:

Before calling the function block, the range limits must be reassigned to memory double words. This is necessary to enable using variable values. Normally, ”upper limit” and ”lower limit” are fixed values.

This can be achieved by setting the ”upper limit” and ”lower limit” parameters in the declaration section of the FC126 to ”REAL”. To enhance flexibility in a test environment, this variant has been omitted.

Sequence of Statements in OB1

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

L10.0;
T MD4;

L –10.0;
T MD0;

CALL FC 126     (
                                lower limit:=MD0,
                                upper limit:=MD4,
                                channel:=PIW272
                                actual value:=MD8
                );
END_ORGANIZATION_BLOCK
```


4.6.3 Example for Programming the Counters

Function of Block

The program is intended to implement a simple function which shows the principle of addressing the counter inputs by the STEP7 program.

The counters are implemented to count up until the comparison value is reached. They are reset when the comparison value is reached and counting is restarted, beginning with zero. Due to the immediate reset, the specified comparison value never can be read out.

In the following program example, the universal inputs are parameterized as follows:

UI1	counter C1
UI2	counter C2
UI3	counter C3
UI4	standard digital input; not used in the example

The three counters are parameterized as follows:

alarm:	yes
counting direction:	forward
edge:	rising

Execution of block:

1. First, all three counters are stopped at start–up.

   This is necessary so that the counter will start counting at zero after a complete restart. If this is not required, that is if the counter must continue after a restart with its ”old” value, the counters must not be stopped.

2. After a waiting time of about 10 ms, a comparison value is written for each counter.

   This waiting time is required so that the STOP command for the counters can become effective on the C7 module. In the complete restart OB (OB100), the times are not critical since the cycles are not monitored.

3. Immediately after the comparison value has been written, the comparison values are declared valid and the counters are started.

4. **OB1**
   The counter values can be read cyclically in the OB1. The counter status bits are evaluated to ensure that the counters are active. OB1 is ended if not all counters are activated.

   If all counters are active, the read counter values are reassigned. This is an optional feature which can be useful for specific applications. If the same value must always be used within an OB1 cycle, re–assignment is recommended (for example in the case of high counting frequency and relatively long cycles > more than one accesses in OB1 might supply different values).

5. **OB40**
   This block is used for alarm evaluation. A jump is executed by evaluating the information of the alarm vector register from the start information of OB40 (LB 8). A memory byte is incremented as a function of the counter which has triggered the alarm. OB40 is programmed to recognize even several quasi–parallel alarms.

6. **OB35**
   OB35 is used to generate the count pulses. The following wiring is required to execute the example:

   Connect digital output 1.2 with DI–X1
   Connect digital output 1.3 with DI–X2
   Connect digital output 1.4 with DI–X3

   In OB35, the output bits of the C7 digital outputs are toggled, and the effect is a period time of 200 ms at each output, correponding to a frequency of 5 Hz. This value results from the 100 ms default alarm time of OB35. This means that each output is set to logic ”1” for 100 ms and then, also for 100 ms, reset to logic ”0”.

S7 Status

With the ”S7 status” function of the STEP 7, the following can be monitored:

PIW280	current counter value C1
MW20	counter image C1
PIW282	current counter value C2
MW22	counter image C2
PIW284	current counter value C3
MW24	counter image C3
MB40	number of alarms triggered by C1
MB41	number of alarms triggered by C2
MB42	number of alarms triggered by C3
PIB287	status of alarms

OB100 Statement Sequence

The OB100 complete restart block contains the following statements:

```
ORGANIZATION_BLOCK OB100
var_temp
                start_info : array(0..19) of byte;
end_var
BEGIN

//***Resetting the counters***
L 0;            // explicit stopping of all counters
T PQB276;       // C1
T PQB279;       // C2
T PQB282;       // C3

CALL SFC 47     (
                    WT:=10000 // wait so that STOP becomes effective
                 );

//***Setting comparison values***
L 10;           // setting comparison value C1
T PQW274;
L 20;           // setting comparison value C2
T PQW277;
L 40;           // setting comparison value C3
T PQW280;

//***Declaring comparison values valid and starting counter***
L 3;            // declaring comparison value valid and starting
T PQB276;       // C1
T PQB279;       // C2
T PQB282;       // C3

END_ORGANIZATION_BLOCK
```

OB1 Statement Sequence

OB1 contains the following statements:

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

//***Evaluation if all counters are active***
L PIB287;       // scanning status bits
T status;

A L20.4;        // C1 signalled active
A L20.5;        // C2 signalled active
A L20.6;        // C3 signalled active
JC run;
BEU;

//***Determining counter image (optional)***
run: NOP 0;
L PIW280;       // C1
T MW20;
L PIW282;       // C2
T MW22;
L PIW284;       // C3
T MW24;

END_ORGANIZATION_BLOCK
```

OB35 Statement Sequence

OB35 contains the following statements:

```
ORGANIZATION_BLOCK OB35
var_temp
                start_info : array(0..19) of byte;
end_var
BEGIN

AN Q1.2;        // assigned to C1
=Q1.2;

AN Q1.3;        // assigned to C2
=Q1.3;

AN Q1.4;        // assigned to C3
=Q1.4;

L QW0;          // transfer QW0 immediately
T PQW0;

END_ORGANIZATION_BLOCK
```

OB40 Statement Sequence

OB40 contains the following statements:

```
ORGANIZATION_BLOCK OB40
var_temp
                start_info : array[0..19] of byte;
end_var
BEGIN

//***Determining which input has triggered alarm***

AN L8.0;        // alarm from C1?
JC c2;

L MB40;         // counts number of alarms from C1 (up to 255)
INC 1;
T MB40;

c2:NOP 0;
AN L8.1;        // alarm from C2?
JC c3;

L MB41;         // counts number of alarms from C2 (up to 255)
INC 1;
T MB41;

c3:NOP 0;
AN L8.2;        // alarm from C3?
BEB;

L MB42;         // counts number of alarms from C3 (up to 255)
INC 1;
T MB42;

END_ORGANIZATION_BLOCK
```

Diagnostics

5

In this Chapter

In this section, you will learn which diagnostics messages you can select and the layout of the diagnostics buffer.

The diagnostics of the C7 analog I/O are described.

This chapter contains a statement of how you can correct the reported error for the most important diagnostics messages of C7 analog I/O with universal inputs.

The term “module” refers here to a unit consisting of the analog I/O and the universal inputs.

5.1 Diagnostics Messages

Overview

The C7-CPU possesses a diagnostics buffer in which detailed information is provided for all diagnostics events are recorded in the sequence of their occurrence. The contents of the diagnostics buffer is preserved even after a complete erasure of the C7-CPU. The diagnostics entries in the diagnostics buffer can be read and interpreted by the applications program.

Advantages

Errors in the system can be evaluated after a long time delay using the diagnostics buffer in order be able to identify the cause of e. g. a STOP or to trace and sort the occurrence of individual diagnostics events.

Diagnostics Events

Diagnostics events are e.g.

- Errors in an **I/O** (module)
- System errors in the C7-CPU
- Change of operating modes (e.g. from RUN to STOP)
- Program errors in the CPU program

C7-CPU I/O Diagnostics

The I/O diagnostics are divided into two groups:

- Standard diagnostics (general malfunction of the C7 analog I/O module and universal inputs)
- Module-specific diagnostics

The standard diagnostics are always entered into the diagnostics buffers of the C7-CPU after the occurrence of a diagnostics alarm. The precondition is an assigned module diagnostic.

The module-specific diagnostics provide detailed information regarding the type and possible cause of the error. This information can be called up by the application program by means of special system calls. The precondition is the assignment of a diagnostics enable (default setting is always ”no” in this case).

Assign I/O Diagnostics Parameters

You can select whether the analog input/output diagnostics messages should be generated using STEP 7.

Using the STEP 7 Tool “*S7 Configuration*)”, you can also assign parameters to the diagnostics behavior of the analog input/output, i.e. you select whether the analog I/O diagnostics messages should be sent to the C7-CPU on request. Furthermore, you can assign parameters to define whether the module should trigger a diagnostics alarm in the C7-CPU after the occurrence of an error.

Diagnostics Information (I/O)

In the diagnostics information, we differentiate between permanent and temporary diagnostics errors.

- Permanent diagnostics errors cannot be influenced by the application program and can only be removed by resetting the C7-CPU (complete erase + restart) or equipment exchange (after a fault).
- Temporary diagnostics errors disappear automatically after a renewed measurement (ADC error, overrange or underrange error), can be removed by the application program (perhaps by parameter assignment during the process operations via SFC55) or by means of a manual entry at the connectors (correction of the wiring).

Read Diagnostics Messages

Diagnostics messages will be entered into the diagnostics buffer only if the diagnostics alarm OB (OB82) occurs. The precondition is that the parameter “diagnostics alarm enable = yes” was assigned. Then you can read out the detailed diagnostics messages in addition to the standard diagnostics information using the STEP 7 Tool “S7 Info” (see *Manual* /**231**/). No entry is made to the diagnostics buffer of the C7-CPU in all other cases. The diagnostics message cannot therefore be read out.

5.2 Diagnostics Data of the C7 Analog I/O and Universal Inputs

Overview

This section describes the C7 analog I/O and universal inputs with regard to their module-specific diagnostics messages.

Analog Input Diagnostics

Table 5-1 provides an overview of the channel-specific diagnostics messages of the analog input.

The diagnostics information is allocated to the individual channels.

Table 5-1 Diagnostics Message of the Analog Input

Diagnostics message	Analog Input
Parameter assignment error	Yes
Synchronism error	No
P short circuit	No
M short circuit	No
Wire-break (only for 4-20mA by software)	Yes
Reference error	No
Underrange (underflow)	Yes
Overrange (overflow)	Yes

Analog Output Diagnostics

Only one collective error exists for the analog output.
Possible causes of the collective error could be:

- Parameter error
- Substitute value is connected

Layout of Diagnostics Area of the Module

The diagnostics area consists of:

- Data set 0: the standard diagnostics bytes (0...3)
- Data set 1: the channel specific diagnostics bytes (for enabled diagnostics).
  - Byte 4..7 and byte 8..11 - channel and individual information analog input (AI) diagnostics
  - Byte 12..15 - channel information- analog output(AO) diagnostics

Table 5-2 illustrates the layout of the diagnostics area and the meaning of the individual entries.

Table 5-2 Layout of the Diagnostics Area

Byte	Bit	Meaning	Explanation	Value Range
00	0	Module fault	1 = error occurrence, 0 = everything OK	0 | 1
	1	Internal error	1 = watchdog, EPROM, ADC error	0 | 1
	2	External error	1 = error at AI or AO	0 | 1
	3	Channel error	1 = with byte 0/bit 2 and channel-specific diagnostic byte byte 4 ...	0 | 1
	4	External auxiliary voltage absent	—	0
	5	Front plug absent	—	0
	6	Module not assigned parameters	Base condition (standard parameter set) byte 0/bit 0=0 ****)	0 | 1
	7	Incorrect parameters	1 with bit 0 of byte 8,9,10,11 or 15 (Standard parameter for channel set	0 | 1
01	0	Module class	SM type class	
	1	Module class	SM type class	5
	2	Module class	SM type class	
	3	Module class	SM type class	
	4	System oriented channel-specific diagnostics info.	Yes	1
02	0		—	0
	1		—	0
	2	Operating mode RUN/STOP		0
	3	Watchdog activated	with bit 1 of byte 0 *) **) ***)	0 | 1
03	0	Rack failure	—	0
	1		—	0
	2	EEPROM error	Serial calibration EEPROM for analog measured values calibration, set with bit 1 of byte 0 *) **)	0 | 1
	3		—	0
	4	ADC error	with bit 1 of byte 0	0
	5		—	0
	6	Process alarm lost	—	0
	7		—	0

Table 5-2 Layout of the Diagnostics Area (Continued)

Byte	Bit	Meaning	Explanation	Value Range
Channel–Specific Diagnostics Entry				
04	0..7	Channel type AI of the following channel–specific diagnostics information		71h
05	0..7	Number of analog input channels		4
06	0..7	Number of diagnostics bits per channel		8
07		Channel Vector Channel Group AI		
	0	Alteration diagnostics entry AI1	0 = no, 1 = yes	0 | 1
	1	Alteration diagnostics entry AI2	0 = no, 1 = yes	0 | 1
	2	Alteration diagnostics entry AI3	0 = no, 1 = yes	0 | 1
	3	Alteration diagnostics entry AI4	0 = no, 1 = yes	0 | 1
	4..7	—		0000
08		Channel–Specific Diagnostics Byte AI1		
	0	Parameter error in parameters for channel	0 = no, 1 = yes *)	0 | 1
	1..3	—		0
	4	Wire-breack in software	0 = no, 1 = yes (only for 4..20mA)	0 | 1
	5	—		0
	6	Measurement underrange	0 = no, 1 = yes (underflow)	0 | 1
	7	Measurement overrange	0 = no, 1 = yes (overflow)	0 | 1
09		Channel–Specific Diagnostics Byte AI2		
	0	Parameter error in parameters for channel	0 = no, 1 = yes *)	0 | 1
	1..3	—		0
	4	Wire-break in software	0 = no, 1 = yes (only for 4..20mA)	0 | 1
	5	—		0
	6	Measurement underrange	0 = no, 1 = yes (underflow)	0 | 1
	7	Measurement overrange	0 = no, 1 = yes (overflow)	0 | 1
10		Channel–Specific Diagnostics Byte AI3		
	0	Parameter error in parameters for channel	0 = no, 1 = yes *)	0 | 1
	1..3	—		0
	4	Wire-breack in software	0 = no, 1 = yes (only for 4..20mA)	0 | 1
	5	—		0
	6	Measurement underrange	0 = no, 1 = yes (underflow)	0 | 1
	7	Measurement overrange	0 = no, 1 = yes (overflow)	0 | 1
11		Channel–Specific Diagnostics Byte AI4		

Table 5-2 Layout of the Diagnostics Area (Continued)

Byte	Bit	Meaning	Explanation	Value Range
	0	Parameter error in parameters for channel	0 = no, 1 = yes *)	0 | 1
	1..3	—		0
	4	Wire-breack in software	0 = no, 1 = yes (only for 4..20mA)	0 | 1
	5	—		0
	6	Measurement underrange	0 = no, 1 = yes (underflow)	0 | 1
	7	Measurement overrange	0 = no, 1 = yes (overflow)	0 | 1
12	0..7	Channel type AQ of the following channel–specific diagnostics information		73h
13	0..7	Number of analog output channels		1
14	0..7	Number of diagnostics bits per channel		0
15		Channel Vector for Channel Group AQ		
	0	Collective error in AQ1	0 = no, 1 = yes	0 | 1
	1..7			0000000

*) Analog inputs will be reset until the channel functions again. (Exception: parameter assignment for wire-break check for setting <>4...20mA) AI=7FFFH

**) Analog output will be reset until channel functions again AQ=0V 0mA

***) Counters will be reset until channel functions again CI=FFFFH, FC/IC= FFFFFFH

****) No process alarm, no diagnostics alarm, no disturbance on the analog I/O bit 0 of byte 0 =0.

5.3 Constraints and Interactions of the Diagnostics Evaluation

Overview

The diagnostics entries are interdependent. For example, the message on the error “wire-break” can only become effective if the diagnostics entries “external error” and ”channel error” are set simultaneously.

Constraints for Error Entry

These interactions are illustrated in Table 5-3

Table 5-3 Constrants and Interactions of the Error Entries

Byte0 / Bit 0 = 1 BG-error			
	Byte0 / Bit 1 = 1 Internal error		
		Byte2 / Bit 3 = 1 Watchdog	(R)
		Byte3 / Bit 2 = 1 EEPROM-error	(R)
		Byte3 / Bit 4 = 1 ADC-error (M/R)	(R)
	Byte0 / Bit 2 = 1 External error		
		Byte0 / Bit 3 = 1 Channelerror	
		Byte7 Evaluate channel vector if req.	
		Wire-break Byte 8, 9, 10, 11: Bit 4 = 1 channel-specific diagnostics bytes AI (E/P)	
		Underrange Byte 8, 9, 10, 11: Bit 6 = 1 channel-specific diagnostics bytes AI	(E/P/M)
		Overrange Byte 8, 9, 10, 11: Bit 7 = 1 channel-specific diagnostics bytes AI	(E/P/M)
	Byte 0/ Bit 7 = 1 Incorrect parameter		
		Byte 8,9,10,11 / Bit 0 = 1 channel-specific diagnostics bytes AI (P)	
		Byte15/ Bit 0 = 1 Collective error AO (only parameter error possible) (P)	
Byte 0 / Bit 6 = 1 Module not assigned parameters			

Legend:

E = temporary, correction by alteration at connector
P = permanent, reset by use of correct parameter
R = permanent, remove by RESET (complete erasure and restart of C7-CPU) or exchange of equipment
M = temporary, disappears after new measurement

Reaction to Diagnostics Messages

The diagnostics messages listed in Table 5-4 refer to Table 5-3.

Table 5-4 lists the diagnostics messages and also possible reactions of the user.

Table 5-4 Diagnostics Messages and Reaction Possibilities

Reason for the Diagnostics Message	Source of Error	Reaction of the BG	Possible Elimination
Module not assigned parameters	During the start-up of the module, if no parameters were assigned to the C7-CPU. The “module error” bit is not set if no further error exists.	Report to C7-CPU that the module is working with default parameters (no channel-specific module diagnostics, no process and diagnostics alarms).	Assign parameters to module
Module fault	Collective error (except no module parameters) of all set diagnostics bits	The error is set/reset with the subordinate diagnostics bits. If the diagnostics alarm has been assigned, one will be generated.	See error under the grouping “module error” (Table 5-3)
Internal error	The error bit is set simultaneously with the error bits “Watchdog”, “EEPROM error”, or “ADC error”. In addition, the watchdog is activated with “EEPROM error”.		See error under the grouping “Internal error” (Table. KEIN MERKER)
Watchdog	The watchdog error is identified after an internal reset of the module. The watchdog error can arise as the result of an EPROM or general module error.	With watchdog, the module adopts a safe state. 0 V is output, the measured values become $7FFF_h$ and the counter values become $FFFF_h/FFFFFF_h$.	The error cannot be corrected by the user. The module can only be restarted after a reset on the bus (Restart C7-CPU).
EEPROM error	The error is identified after resetting the module during the reading of the calibration values for the compensation of the offset error of the analog I/O from the serial EEPROM.	The module adopts a safe state. 0 V is output, the measured values become $7FFF_h$ and the counter values become $FFFF_h/FFFFFF_h$.	The error cannot be corrected by the user. The module can only be restarted after a reset on the bus (Restart C7-CPU) or the analog I/O must be recalibrated at the manufacturer’s factory (equipment exchange).
External error	The error bit is set when channel-specific errors of the analog inputs or outputs occur.	Refer to the grouping “External error” in Table 5-3.	Refer to the grouping “External error” in Table 5-3.
Channel error	A channel causes an error. PLEASE CHECK	Refer to the grouping “External error” in Table 5-3.	Refer to the grouping “Channel” in Table 5-3.

Table 5-4 Diagnostics Messages and Reaction Possibilities (Continued)

Reason for the Diagnostics Message	Source of Error	Reaction of the BG	Possible Elimination
Wire-break	Precondition: The measurement range 0...20mA has been set for the channel. If a wirebreak check has been assigned, then the error is identified by evaluation of the input current of the analog input channel (<1.6mA).	An error counter will be incremented. If the error counter reaches a fixed value of 3, then the error ”wire-break” is reported.	Check the connection of the appropriate measurement channel.
Overflow	The error is identified after the comparison of the measured value (incl. correction calculation). Measurement >=overflow range.	The bit is set and reset again when the measurement decreases.	Check the connection of the appropriate input channel or transducer.
Underflow	The error is identified after the comparison of the measured value (incl. correction calculation). Measurement >= neg. overflow range. That is <0mA for 4...20mA.	The bit is set and reset again when the measurement becomes valid.	Check the connection of the appropriate input channel or transducer.
Incorrect Parameter	The error is identified after checking the parameter after reading and processing the parameter area.	The measurement $7FFF_h$ is set in the incorrectly configured measurement channel and the corresponding diagnostics bit set or, for an output channel, 0 V /0 mA is output and the corresponding bit set new line. If the module had not been assigned a parameter and the parameter is corrected (Please check!), then the bit ”incorrect parameter” is reset and (if diagnostics alarm = yes) a diagnostics alarm reported to the C7-CPU.	Assign correct parameters.

Operating the C7 (General)

6

In this Section

Note

The explanations in this chapter relate to the so-called "Standard screens" which are supplied in an standard configuration with the configuration tools *ProTool* and *ProTool/Lite*. The special screens are called up in the standard screens. The standard screen may be redesigned for a customized operator interface. However, the special screens are stored permanently in the C7 firmware and can therefore not be modified.

6.1 Keyboard

Keyboard Layout

The C7-623 and C7-624 key layout is ergonomically tailored to the various requirements of the device operating modes, by means of its arrangement and colored layout.

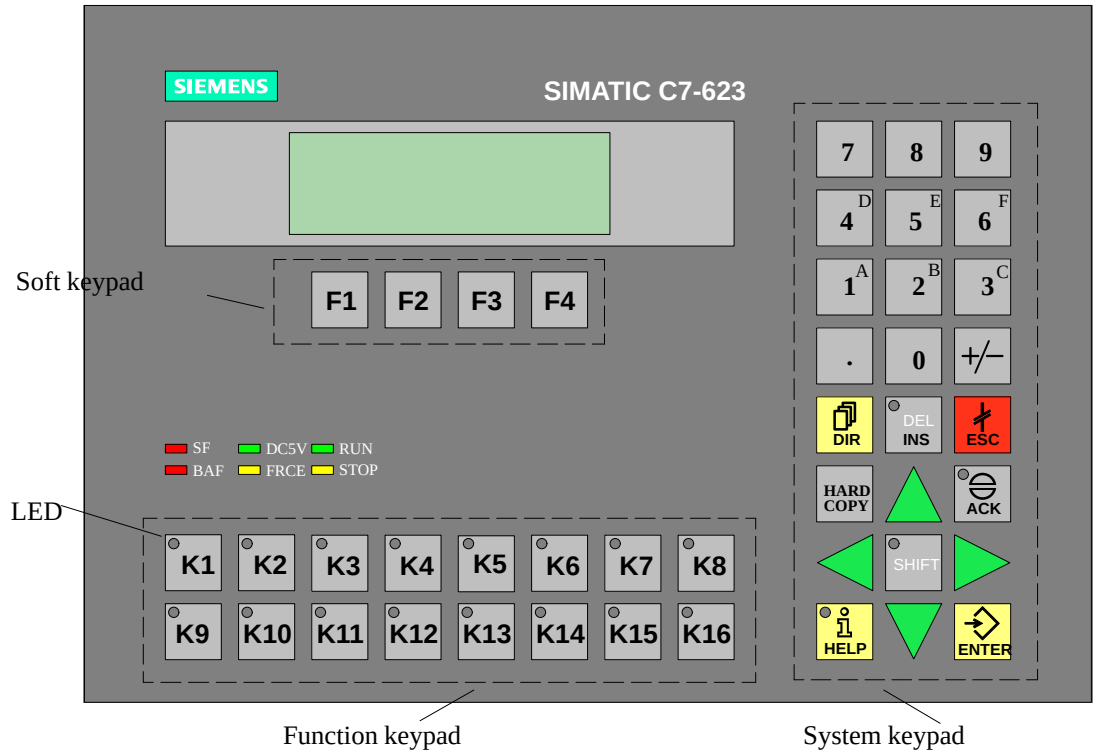


Figure 6-1 C7-623 with Keyboard and Display

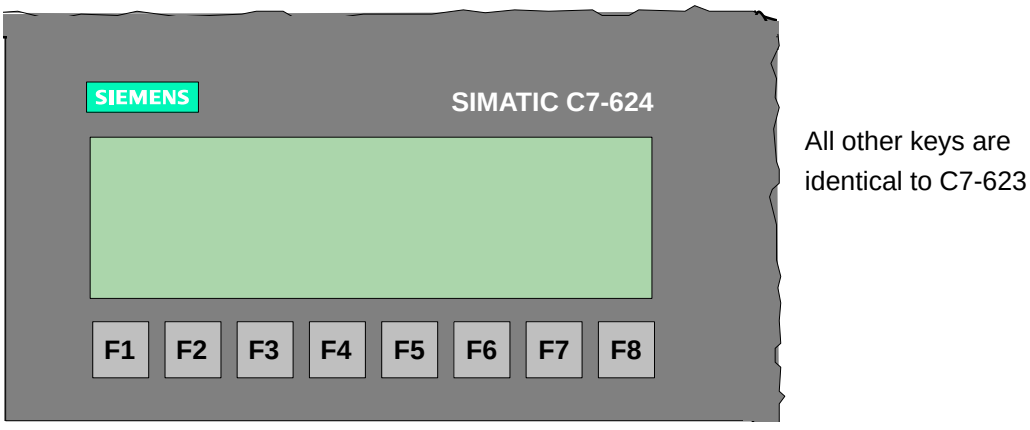


Figure 6-2 C7-624 with Keyboard and Display

Key Labeling and Key Functions

The C7 is operated by means of a keyboard. The keyboard is categorized into

three functional blocks (see Figure 6-1):

- System keypad
- Function keypad
- Soft keypad

System Keypad

In Figure 6-1, the key block with the system keys is marked. The functionality of the individual keys is explained in table 6-1.

Table 6-1 Key Function

Key	Function	Explanation
SHIFT	**Shift key**	By means of the shift key, you can activate the second function of dual assignment keys - for example, the shifting of numeric keys 1 - 6 for inserting characters A - F. The enabled SHIFT mode is indicated by a SHIFT LED which lights up after you press the SHIFT Key. The secondary function of a dual assignment key can then be called.
0 to 9	**Numeric keys**	Input keys for numerical characters (0 - 9); In shift mode of keys 1 - 6: Input keys for alphanumeric characters (A - F). The keys have a repeat function. When the keys are pressed and held down, after a short time delay, the input entry is repeated for as long as the key is pressed down.
+/-	**Sign key**	Changes sign from “plus” to “minus”
DIR	**Directory key**	Displays screen directory (see Section 7.5). In Shift Lock mode: Recipe directory.
ACK	**Acknowledge key**	Acknowledges alarm messages The LED of the key is on when at least one unacknowledged alarm message is present.
DEL INS	**Insert key**	Pressing the insert key inserts a blank at the current cursor position, when alphanumeric values are being entered. All subsequent characters are moved one position to the right.
SHIFT + DEL INS	**Delete key**	The delete key deletes the character at the current cursor position, when alpha-numeric values are being entered. All the characters thereafter are moved one position to the left.
ENTER	**Enter key**	With this key you enter and confirm your input. This means, that among other things, you also change from message level to screen level.
ESC	**Escape key**	Escape, branch backwards, change operating levels; (see page 6-5).

Table 6-1 Key Function (Continued)

Key	Function	Explanation
HARD COPY	**Hardcopy key (Print)**	Outputs the current display to the printer.
i HELP	**Help key (displays information text)**	When the assigned LED is on, information text relating to the current display can be viewed on the screen (see Section 6.5).
◁ ▷ ▽ △	**Arrow keys**	Move the cursor. Depending on the operating situation, the cursor is moved one character, field, entry or display to the left, right, up or down. The keys have a repeat function. When the keys are pressed and held down, after a short time delay, the input entry is repeated for as long as the key is pressed down.
SHIFT ◁ ▷ ▽ △	**Browse the symbol list**	Browses through the list of symbols and calls up extended character sets.
SHIFT +/-	**Display contrast adjustment**	Adjusts the display's contrast.
SHIFT i HELP	**Select system functions menu**	The menu for choosing the C7 CPU operating modes (RUNP, RUN, STOP and MRES) is displayed. The DI/DO condition is also displayed. The menu can be selected from every level.

Note

Simultaneous pressing of more than one key may result in erroneous input.

Function Keypad

You can call up freely configurable functions up with the function keys K1...K16. Every key has its own LED (see Figure 6-1) which can be controlled from the C7-CPU.

Soft Keypad

The soft keys F1...F4 (C7-623) or F1...F8 (C7-624) underneath the display can be configured with different functions depending on the display.

ESC Function

The ESC key has various functions, depending on the settings of the function level. This is explained in Table 6-2.

Table 6-2 Function of the ESC Key

Function	Explanation
Cancel	Pressing ESC cancels field inputs provided that they have not been confirmed with ENTER.
Branch backwards	If the ESC key is pressed within a screen, you branch back to the configured cross-jump destination (by default, the last position called) and finally, from the start screen to the message level. You cannot branch back further than the message level.
Hide system message	At message level, the ESC key is used to hide a system message
Cancel display of information text	To cancel the display of information text, press ESC to return to the previous display.
Reset while scrolling through messages	To cancel scrolling through messages at the message level, press ESC to reset the display to the current message.

6.2 Operating Levels

Overview

When operating the C7, you have to distinguish between two distinct operating levels, between which you can switch:

- **Message level**

  At the message level, current messages are displayed.

- **Screen level**

  At the screen level, functions are chosen, serviced and executed.

Message Level

The message level is the highest level on the C7. At message level, waiting event messages, alarm messages and system messages are displayed. After the C7 is up and running, it changes to message level and displays the "standby message".

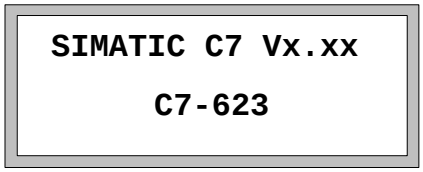


Figure 6-3 C7-623 Standby Message (Operating Message No. 0)

Screen Level

The screen level is reached by pressing the ENTER key. The first screen to be called is the "start" screen. From the start screen, you can branch to other screens, depending on the configuration. On the screens, you can view current process values, enter values and initiate functions by means of soft keys.

The linking of individual screens is referred to as a "screen hierarchy". If you go to the bottom of the screen hierarchy, you go back one stage by pressing the ESC key. You can step right back to the start screen if you wish. From here you can return to the message level by pressing the ESC key. You can also return directly to the message level from a screen, depending on the configuration.

Changing Operating Levels

You can change the operating level:

- From screen level to message level by pressing the ESC key
- From message level to screen level by pressing the ENTER key

You cannot branch backward from the message level by pressing ESCAPE. This key is designed to terminate the display of a system message at this level.

Forced Change to Message Level

You are automatically exited from the screen level whenever a system message or an alarm message is waiting to be displayed. The C7-OP then changes to message level to display the message. You cannot exit from the message level while a system message or unacknowledged alarm message is being displayed. The display of an unacknowledged alarm message is indicated on the C7-OP by the:

- Alarm message flashing and
- Illuminated ACK LED 

Use the ACK key for acknowledging an alarm message. Press the ESCAPE key to fade out a system message.

Acknowledging of the alarm message or fading out the system message takes the C7-OP back to the point where it changed to the message level.

6.3 Operating with the Standard Functions

Overview

A standard configuration, which contains standard screens, is supplied with the configuration software “ProTool”. All functions required for the operation can be selected using these standard displays. The individual functions are described in this manual with the help of standard screens. If you want to work with these standard screens, you must load the supplied configuration.

The standard screens contain functions that are fundamental to C7 operation. They include for instance, calling and printing message buffers, editing passwords and modifying parameters on-line. Process-specific implementations such as event messages or screens for the process, are not included.

Note

The standard screens, supplied with the standard configuration, call up the so-called special screens, which are loaded into the C7’s firmware. You can call up these screens in your own configuration by using the function “Select Special Screen”.

Standard Screens

Standard screens are called from a basic screen by means of a soft key. From the basic screen, a branch can be made to the following screens:

- Event messages

  At this point, the event buffer is called, printed or deleted (see also Section 7.1.1).

- Alarm messages

  At this point, the alarm buffer is called, printed or deleted (see also Section 7.1.1).

- Screens

  At this point, the screen contents directory is called to edit or print screens. All the screens which were given the “directory” attribute, are listed here. If you still have not created any screens of your own, the directory will be empty (see also Section 7.5).

- Data records
  At this point you can set up and edit data records. You can also transfer data records from the C7–OP to the C7–CPU and back.

- System settings (System)

  At this point you can modify settings in on-line mode. This includes for example, printer parameters, interface parameters, operating mode and language changing.

- Status variable (StatVar)

  At this point, the PG function STATUS VAR is called; you can use it to display C7-CPU addresses.

- Modify variable

  At this point, the PG function MODIFY VAR is called; you can use it to display and modify C7-CPU addresses.

- Edit password (password)

  At this point, the super-user assigns passwords for the different password levels. Furthermore, login and logout procedures are included here.

Standard Screen Hierarchy

Figure 6-4 shows an overview of the screen hierarchy of standard screens. You will find detailed information on functions and handling standard screens in the relevant chapters of this manual.

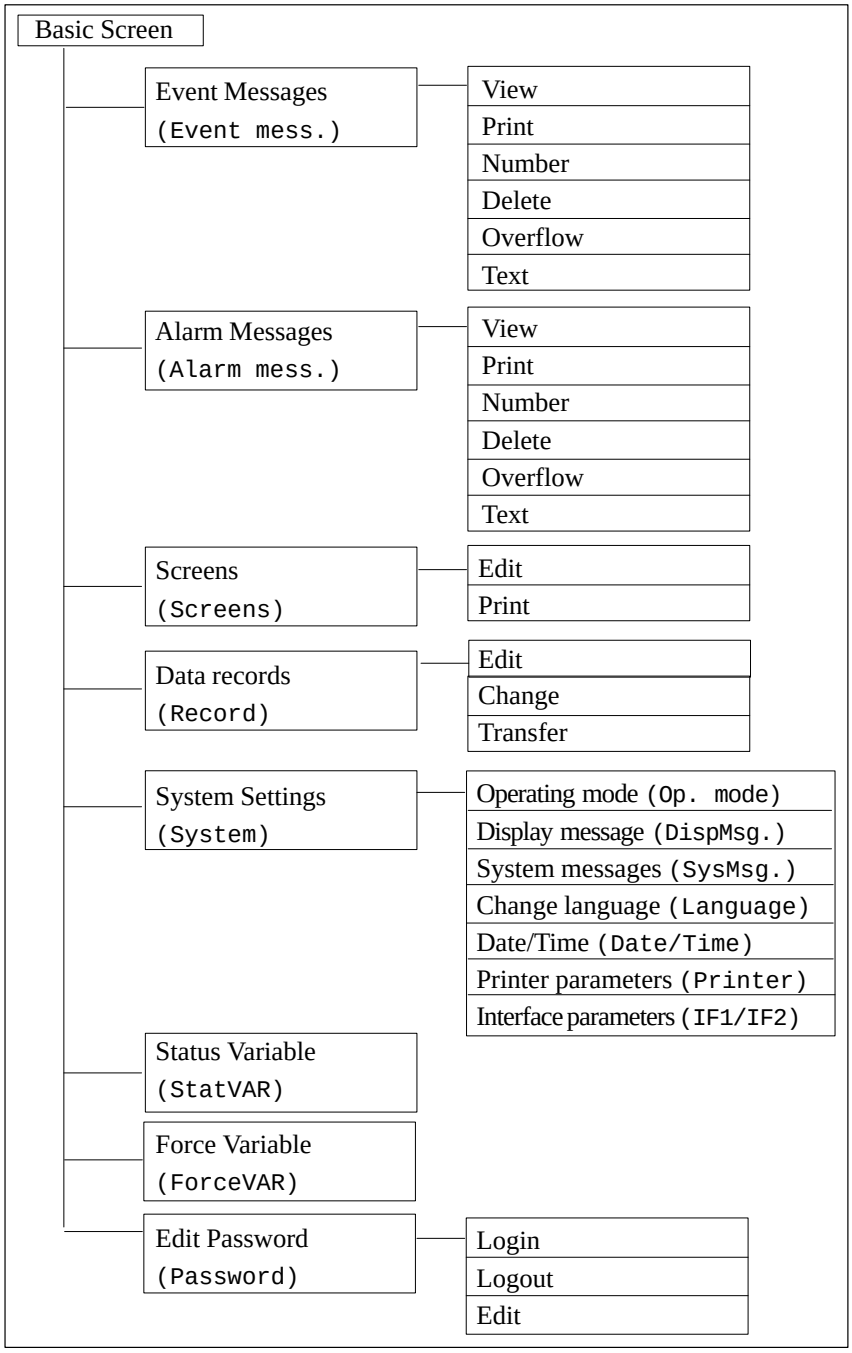


Figure 6-4 Hierarchy of Standard Screens in the Example Configuration. The Labels in ( ) are the Actual Messages Which Appear on the Display.

Branching in Standard Screens

You change from message level to screen level with the ENTER key. Here you can control and monitor the process or installation by means of suitable screens as well as perform system settings.

Example

With reference to the standard screens, a description of the procedure for branching between individual screens within the screen hierarchy is provided below.

Step	Action	Result
1.	Call up the **basic screen** from the supplied standard screens.	Its full name is displayed in the first line. Depending on the C7 version and display size concerned, the next lines include a different number of text elements of the current entry. These text elements indicate the name of other standard screens to which you can branch by using the assigned soft keys.
2.	Using the soft keys beneath the symbols << and >>, you can move the displayed screen segment of the current screen.	Screen segments outside the viewable display area are displayed.
3.	You branch to the next screen by pressing the soft key beneath the appropriate text indicating the name of the screen to be called.	

Figure 6-5 shows the branch principle with reference to the 8-line display of the C7-624.

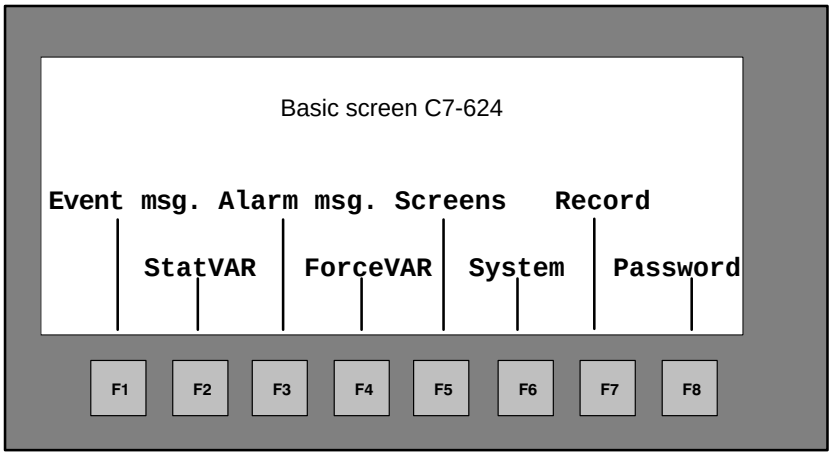


Figure 6-5 Branching at Screen Level (Example for C7-624, 8x40 Character Display)

Figure 6-6 shows the branch principle of the C7-623 4-line display.

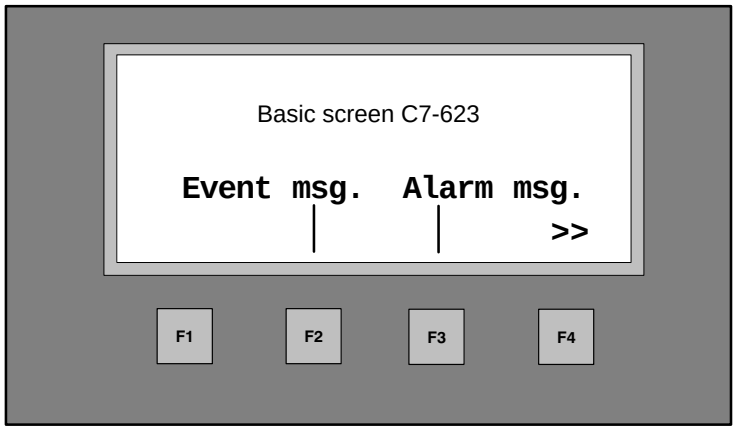


Figure 6-6 Branching at Screen Level (Example for C7-623)

6.3.1 Setting C7-OP Operation Modes

Overview

A standard screen can be used to set the C7-OP modes On-line, Off-line and Download. You can set the C7-OP modes via the basic screen, as long as a configuration which contains this screen is loaded. Table 6-3 provides a description of these modes.

Table 6-3 C7 Operating Modes

Operating Mode	Explanation
On-line	In On-line mode, there is either a logical link between the C7-OP and the C7-CPU already, or the C7-OP attempts to establish a link.
Off-line	In Off-line mode, a logical link does not exist between the C7-OP and C7-CPU. The C7-OP does not even attempt to establish a link.
Download	In transfer mode, data is transferred from the programming device/PC to the C7-OP. The transfer is achieved over the serial interface. The C7 is not serviceable during the transfer.

Configuration Not Loaded

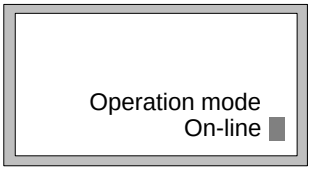
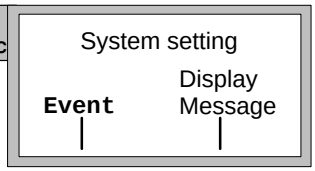
If either the configuration, or the basic screen for changing the **C7-OP mode** is not loaded, you can load the configuration afterwards. You must then reset C7-OP (see also section 2.1).

Transfer in the C7-OP

The transfer of the configuration data from the PU/PC is always carried out in the C7-OP. The transfer to the target device is described in the *ProTool* or *ProTool/Lite* manual. The OP is also situated there to stand in for the C7-623 and C7-624 as a target device.

Setting Operating Modes

The adjustment of the C7-OP operating modes is described below:

Step	Action	Result
1.	Select the standard screen **System → Operation**	
2.	Press the SHIFT key and, if necessary, by pressing the keys ▲ or ▼ several times, set the required operating mode shown in Table 6-3.	
3.	Confirm your choice by pressing the ENTER key	The required mode is now set.
4.	Leave the standard screen by pressing ESC	

6.4 Entering Values

Overview

You enter the values to the C7 by using input fields. There are various types of values which you enter in various ways:

- Numeric values
- Alphanumeric values
- Symbolic values

Input of General Values

In general follow the procedure outlined below:

Step	Action	Result
1.	As described on page 6-11, branch initially to the screen you require and then to the corresponding screen entry.	The required screen is displayed.
2.	Using the cursor keys, select the input field you require within the screen entry.	Cursor is positioned on the input field.
3.	Enter the required value. Depending on the field configuration, you can input the values listed under ”Overview”.	The input field is flashing.
4.	Confirm your input with ENTER Cancel incorrect input with ESC	The value is accepted and the input display stops flashing. The original value is automatically re-entered into the field.
5.	Position the cursor with the cursor keys to the next field and follow the same procedure. You can also use the cursor keys to move the cursor to the left or right of the previous input field to insert a different value into it.	dto.
6.	Close the screen by pressing ESC	The display is reset and changed back to the previous screen.

Numerical Values

In fields that allow the operator to enter a purely numerical value, you enter the numerical value character by character using the keypad on the system keyboard.

Entering Numerical Values

In numerical fields, the inputs are normally right-justified. Digits that have already been entered are moved to the left.

If there is a value in the field already, it is cleared completely from the field when the first character is entered.

Exception: Input fields for set-points in BIN format (for instance, when calling the PG functions STATUS/FORCE VAR) are changed to left-justified. When input begins, 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 pressing the arrow key  and  with the Shift key activated.

As soon as you have started entering numbers, the relevant field cannot be left until the entry is either confirmed or aborted.

Limit Values

You can configure **limit values** for numerical input fields. In this type of field, a limit value check takes place, this means the entered values are applied only if they lie 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 restored in the field.

Field with Decimal Places

If a numerical field has been configured with a certain number of **decimal places** and too many have been entered, they are ignored; if too few have been entered, the field is padded with zeroes.

Alphanumeric Values

Alphanumeric inputs contain a mix of digits and letters.

For the numerical part of the input, please proceed as described in “Numerical values”.

Entering Alphanumeric Values

If you wish to enter a letter at the current cursor position, you must enable the corresponding alphanumeric character set.

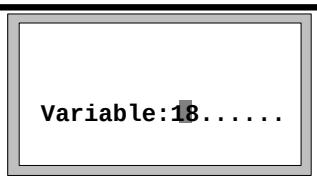
Entering Characters A - F

Characters A to F can be entered directly with the numeric keys 1 to 6 in combination with the shift key. To enter an alphanumeric string for instance 7FEB53, follow the procedure outlined below:

Step	Action	Result
1.	Enter the number 7 as normal, via the system keyboard.	Digit appears in the input field
2.	Press the SHIFT key.	The activated shift key is indicated by the illuminated LED
3.	Enter the letters F (on the 6 key), E (on the 5 key) and B (on the 2 key).	The letters appear in the input field.
4.	Press the SHIFT key again	The Shift key is deactivated. The LED goes out.
5.	Enter the rest of the digits (5 and 3) using the keyboard and confirm your entry with ENTER	Your input is accepted and the displayed entry is reset.

Entering the Complete Character Set of the C7

If the characters available on the keypad are not sufficient for the alphanumeric input, you can select additional letters and special characters from the extended character set. The extended character set is available for all fields with the format STRING. To enter the string 18 OCT 61, for example, please proceed as follows:

Step	Action	Result
1.	You must first configure the input field with the variables. As an example, you can select a string with the format CHR. Enter the digits 1 and 8 as normal, using the numeric keys on the system keyboard.	
2.	Press the SHIFT key	The extended character set is available in the current input field.

Step	Action	Result
3.	Scroll with the △ or ▽ keys through the valid extended character set.	
4.	Select **O**, and using the arrow key, move one position to the right.	The selected character is accepted by moving the cursor key
5.	Select **C**, and using the arrow key, move one position to the right.	ditto
6.	Select **T**, and using the arrow key, move one position to the right.	ditto
7.	Press the ○ SHIFT key	The Shift key is deactivated. The LED goes out.
8.	Enter the remaining digits 6 and 1 as before using keys 6 and 1 with de-activated Shift key. Then confirm with ENTER	 The input is accepted and the input display is reset.

Symbolic Values

If a field has to be filled by entering symbolic values, take the value from a list of options.

For entering a symbolic value, proceed as follows:

Step	Action	Result
1.	Press ○ SHIFT in the input field.	The activated Shift key is indicated by the illuminated LED. The selection list with the configured symbolic inputs is activated.
2.	Using the cursor key, select the required value.	
3.	Confirm the selected value with ENTER	The Shift key is automatically de-activated.

6.5 Setting C7 System Parameters and Language

Overview

After you have loaded a configuration, you can change the C7-OP system settings via standard screens:

- Language (German, English, French, Italian and Spanish)
- Parameters
  - Change C7-OP interface parameters
  - Change printer parameters
  - Set message logging
  - Set message display (first/last message)
  - Set date and time
- Adjust contrast

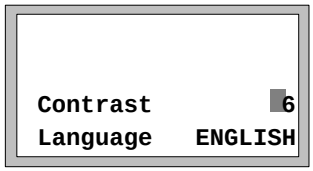
Setting the Language

Messages, screens and information text can be displayed in several languages. Up to three of the languages listed below can be loaded simultaneously on the same C7 and presented to the operator for selection on-line:

- German
- English
- French
- Italian
- Spanish

Displays can also be configured in Russian (Cyrillic characters) on the C7-624.

To choose another language, proceed as follows:

Step	Action	Result
1.	Go to the screen level if it has not yet been set.	The basic screen is displayed.
2.	Select the standard screen **System → Language.**	
3.	a. Position the cursor to the selection field ENGLISH b. Select the required language via the symbolic inputs. Only languages that have been loaded into the C7 are listed. c. Confirm the input with 	The C7 starts again and shows all language-dependent text in the new language.

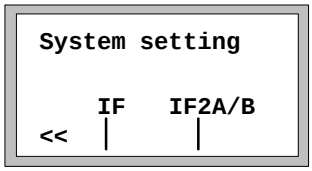
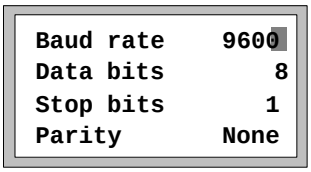
6.5.1 Changing Parameters On-line

Overview

You can change the parameters which were set by the configuration, on-line, using the C7-OP.

Changing the Interface Parameters

You can change the parameters for the C7-OP interfaces via the standard screens. The procedure is outlined below:

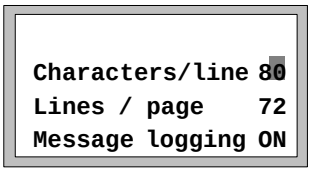
Step	Action	Result
		The result is dependent on the individual C7 and the chosen configuration.
1.	Depending on which C7 system you are using, choose the screen level "system settings" from the basic menu. Shown here is one of the standard screens with **SYSTEM → IF**.	
2.	Using the arrow keys, scroll through the parameter list on the display to the required parameter. Some of the parameters displayed cannot be modified.	
3.	Choose the value for the parameter by using a symbolic value (to be configured in advance).	New parameter is set.
4.	Accept the changed parameter with ENTER	If a password has been set, the required password is requested by the system.
5.	Enter the password and acknowledge with ENTER Quit the special screen with ESC	The parameter list is displayed.
6.	Leave the standard screen with ESC	You are changed to the previous level.

Changing Printer Parameters

The parameters, set as default parameters for the C7-OP interfaces during configuration can be modified by means of the standard screen. You can set the following:

- Characters per line (parameter values: 20, 40 or 80)
- Lines per page (parameter values: 60 to 72)
- Message logging (ON, OFF)

To set the printer parameters, follow the procedure outlined below:

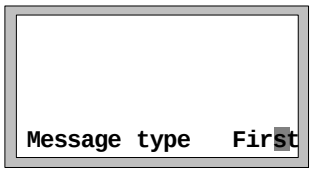
Step	Action	Result
1.	From the basic screen choose **System** and the screen level “system settings”. Select the standard screen **Printer**.	
2.	Using the arrow keys, scroll through the parameter list on the display to the required parameter.	
3.	Choose the value for the parameter by using a symbolic value (to be configured in advance)	New parameter is set.
4.	Confirm the changed parameter with ENTER	If a password has been set, the required password is requested by the system.
5.	Enter the password and acknowledge with ENTER Quit the special screen with ESC	The parameter list is displayed. again
6.	Leave the standard screen with ESC	You change back to the previous screen level.

Activating / Deachivating the Message Logging

When the message logging function is active (ON), all alarm and event messages which occur on-line, are printed out. This is also so for alarm messages upon acknowledgement. You can switch off this logging process by setting the parameter OUT. The procedure for setting this parameter is similar to “changing printer parameters”.

Setting First/Last Message Display

To define whether the oldest or the most recent alarm messages (see section 7.2 for first/last message) is displayed when several are waiting to be displayed, proceed as follows:

Step	Action	Result
1.	From the basic screen choose **System** and the screen level “system settings”. Select the standard screen **DispMsg**. - check	
2.	Select the corresponding parameter **first** or **last** with SHIFT and ▲ or ▼	Password is requested
3.	Enter password	
4.	Repeat operation as described in Step 3 and confirm.	
5.	Exit the standard screen with ESC	Basic screen reappears.

Setting the Date and Time

You can adjust the current date and time on the C7-OP to compensate for summer or winter time for example. Any change you make will affect all messages and screens for which a date or time variable has been defined. The display format for date and time is defined in your configuration and cannot be changed by means of the menu system on the C7-OP.

To change the date and time, proceed as follows:

Step	Action	Result
1.	From the basic screen choose **System** and the screen level “system settings”. Select the standard screen **Date/Time**.	

Step	Action	Result
2.	Set the weekday with the keys SHIFT and ▲ or ▼	
3.	Confirm input with ENTER	
4.	With the keys SHIFT ◀ or ▶ position the cursor to the right or to the left in the date field.	Cursor jumps to the date field or back.
5.	Enter the date with a numerical input. First press the Shift key.	
6.	Change between `date` and `time` with the cursor keys.	Cursor is positioned on the time field.
7.	Carry out the procedure described in Step 4.	
5.	Confirm with ENTER	New parameter is set.
6.	Leave the standard screen with ESC	Previous screen setting appears.

6.5.2 Contrast Adjustment

Overview

You can adjust the contrast of the C7's LCD. The display contrast can be adjusted either by:

- Direct adjustment or via a
- Standard screen

Direct Contrast Adjustment

The display contrast of the LCD can be adjusted using the system keyboard.

Step	Action	Result
1.	Press the key ○ SHIFT	The SHIFT LED lights up.
2.	Adjust the contrast +/- to the required value by pressing the key (either a single key press or continually)	The contrast changes for as long as the key is pressed.
3.	Press the key ○ SHIFT	The SHIFT LED goes out.

Contrast Adjustment via Standard Screen

You can also adjust the contrast of the LCD display using the standard screen. To do this, proceed as follows:

Step	Action	Result
1.	Select the standard screen **System → Language**.	Cursor is positioned on the input field for contrast.
2.	Select the required contrast value using the symbolic inputs (1 for the lowest value, 16 for the highest value).	
3.	Leave the standard screen with ESC	You change back to the previous screen level.

6.6 Password Protection

Overview

To prevent unauthorized operation of the C7, it is possible to protect access to certain functions and inputs, by means of passwords and password levels.

When an operator is issued a password s/he is simultaneously granted permission to execute functions at a specific password level.

A password and password level are defined during the configuration.

If password protection is implemented you must first log into the system (LOGIN) and then log out (LOGOUT) when finished.

Logging into the C7

You can log into the C7 in various ways:

- Via the standard screen **Password → Login**
- By calling a function for which the current password level is too low. In this case the request for a password automatically appears on the display.

Logging in Using the Standard Screen

To log in using the standard screen, proceed as follows:

Step	Action	Result
1.	Select the standard screen **Password → Login**.	The password input mask appears on the display. This mask appears automatically on the display every time a function is called for which the current password level is too low. The cursor is positioned on the first input field.
2.	Enter the password using the system keyboard and then press 	After entering each character, the cursor automatically jumps to the next field. Every character input is shown as asterisk (*). For valid passwords, the corresponding password level is indicated.
3.	Close the screen with 	You change back to the previous password level.

Logging out of the C7

Select the standard screen **Password → Logout**. Upon selection of the standard screen, the C7 switches from the current password level to the lowest password level, level 0, and branches to the message level.

Automatic Call

If a function requires a higher password level than the current one, you are automatically requested to enter a suitable password when the function is called.

After entering the password, you return to the calling function by pressing

6.6.1 Password Level and Access Rights

Password Level

During configuration, you assign hierarchically ascending password levels from 0 to 9 for functions called by function keys, soft keys and input fields. The password levels assigned to the standard screens are listed in Appendix C.I.

When an operator is issued a password s/he is simultaneously granted permission to execute functions at a specific password level.

Table 6-4 Password Level

Password level	Explanation
0	At this level, the lowest in the hierarchy, functions are assigned which, when executed, have no or little effect on the course of the process; these are normally calls for functions which have no input options - for example, viewing message buffers. A password is not required to call functions with password level 0. If an operator calls a function of a level other than 0, he is prompted to enter a suitable password.
1-8	Levels 0 to 8 are assigned to functions with ascending priority. A password level is assigned to a password by the super-user as part of the password management responsibilities.
9	Permission to execute functions of password level 9 is solely reserved for the super-user, who has access to all of the C7 functions. Only a super-user is allowed to perform password management. The password for level 9 functions is created at the time of configuration.

6.6.2 Password Management

Overview

Only the super-user (password level 9) is authorized to call the functions for password management (standard screen **Password → Edit**). Upon delivery of the C7, the default setting of the password is **Password 100**. This can be changed during configuration.

Displaying the Password List

Select the standard screen **Password → Edit**.

The password list is displayed. A maximum of 50 passwords is possible.

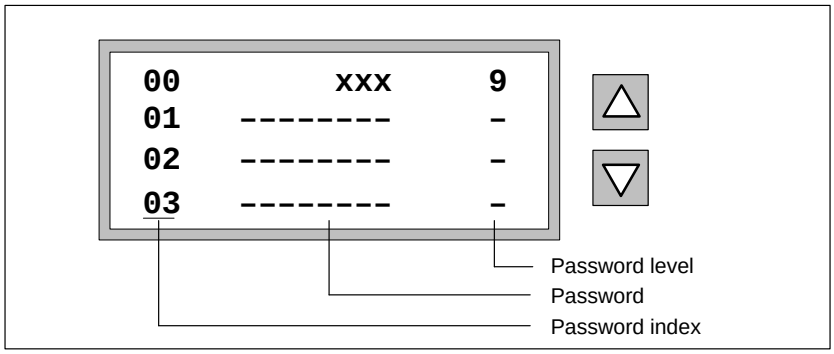


Figure 6-7 Example of a Password List

Table 6-5 Explanation of Terms in Figure 6-7

Legend in Diagram	Explanation
Password index	The two-digit number at the start of the password entry is the password index. If no password is entered for a defined password index, the fields for the passwords are represented by dotted lines.
Password	To the right of the password index is the field for the actual password. The password must be a minimum of 3 and a maximum of 8 characters long. It can be made up of digits and the letters A-F.
Password level	When the password list is first called up, it contains only the entry of the super-user.

Issue of Password and Password Level

A password and password level are issued according to the following procedure:

Step	Action	Result
1.	Select the line for password entry in the password list.	The cursor is positioned on the first character of the field for the password input.
2.	Enter a password that does not yet exist and confirm it. A leading zero in the password is not valid.	
3.	Position the cursor to the right in the field for password level.	
4.	Enter a password level 1-8 for the password and confirm it.	
5.	Leave the standard screen with ESC	

Changing Password and Password Levels

To change a password, call the password entry in the same way as you do for allocating a password and enter the new password by overwriting the old one.

If you just want to change 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 Password

To delete a password, call the password entry in the same way as you do for allocating and changing a password but overwrite the first character of the password with a zero.

Standard O/I Functions

7

In this Section

7.1 Messages

Overview

Events and states in the control process are shown on the C7 display by way of messages. A message consists of static text as a minimum. If required, variables are integrated into a message.

The following types of message are displayed on the C7:

- Event messages
- Alarm messages
- System messages

Selecting Message Level

You get into the message level by pressing 

Leaving the Message Level

You leave the message level by pressing 

Event and Alarm Messages

Event and alarm messages are initiated by the C7-CPU. They are configured and they contain process-related information. Alarm messages have to be acknowledged.

System Messages

System messages are initiated by the C7-OP. They are not configured. They provide information on operating states of the C7-OP or faults and breakdowns in communication.

7.1.1 Event and Alarm Messages

Overview

The process states indicated by event or alarm messages are defined during configuration.

- Messages concerning malfunctions related to processes or states, (that is 'motor temperature too high' or 'valve will not open' should be categorized as **alarm messages**.
- Messages that provide information concerning regular processes or states such as 'temperature reached' or 'motor running' should be categorized as **event messages**.

In addition to status messages, information relating to an operation can also be configured as event or alarm messages. If that is, a machine operator wants to start the bottling process but has forgotten to open the water intake valve on the mixer, a message such as '`open water intake valve`' is displayed as a prompt to rectify the oversight.

Event and alarm messages can be configured so that any of their text components flash to distinguish them from normal message text. Unacknowledged alarm messages flash permanently.

Messages may contain static text and variable fields. The variable fields display, that is, current actual values of the C7-CPU in numerical or symbolic form. In addition, the date and time can also be output in messages.

Output event and alarm messages are stored in dedicated message buffers. The messages available in the buffers can be displayed on the screen and logged with a printer.

Message Bit Procedure for Event and Alarm Messages

If there is a condition present in the current process for a message to be issued, that is a setpoint has been reached, the application program sets a bit in the data area for event and alarm messages. The C7 reads the data area after a set polling time. In this way, a message is detected as having "arrived". The C7 resets the bit when the condition for issuing the message no longer exists. The message is then regarded as having "departed".

Further information about the data area can be found in section 9.3.

Alarm and Event Message buffers

Alarm and event messages are written to the C7's alarm and event message buffers respectively upon arrival. The following events are entered into the buffer in chronological order:

- Time of the event
- Arrival and departure of the event
- Acknowledgment of alarm messages
- Message number
- Values of variables at the time of arrival or departure.

The C7's message buffer can store up to 256 events. Events are:

- The arrival of a message
- The acknowledgment of a message (alarm messages only)
- The departure of a message

Size of Residual Buffer

During configuration you can define a residual buffer area. When this residual buffer space has been reached, an automatic overflow warning is issued, that is `EM residual buffer`. This overflow warning is a system message. Messages continue to be inserted into the buffer even after the residual buffer space has been reached.

Standby Message

A sub-category of the event message is the standby message. The standby message is the event message number 0. It appears on the display when the C7 is operating at the message level and event or alarm messages are not waiting. The standby message is stored in the firmware and contains the version and the device type:

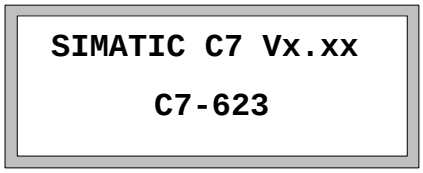


Figure 7-1 Standby Message C7-623

Depending on the configuration, the standby message can be represented by other text. It can contain the date and time but not variables.

7.1.2 Acknowledging and Suppressing Alarm Messages

Acknowledging Alarm Messages

Alarm messages have to be acknowledged because of their urgency. This can be done automatically by the C7-CPU or by the operator by means of the acknowledge key. The C7 displays unacknowledged messages in flashing mode. The ACK LED assigned to the acknowledge key also lights up.

- If several alarm messages are waiting to be displayed, after one has been acknowledged, the next one is shown on the display, which similarly has to be acknowledged.
- If no more alarm messages are waiting to be displayed, the C7 reverts to the operating level from which it originally branched out of.

Acknowledgement Groups

During configuration, you can combine several alarm messages to form an "acknowledgement group". In this way you can acknowledge all other alarm messages (consequential faults) in that acknowledgement group by acknowledging the first alarm message (that is, the cause of the fault) without all of them having to be displayed one after the other. You can configure up to four acknowledgement groups.

If alarm messages are not assigned to an acknowledgement group, only the one that happens to be on the display is acknowledged if more are waiting. See also section 9.3.

Suppressing Alarm Messages

The possibility exists to suppress the display of frequently incoming alarm messages for the duration which the C7 takes to boot up as a result of a warm or cold restart.

Password level 8 is the minimum level required to suppress the display.

Activating the Suppress Function

To suppress an alarm message, press the ENTER and ESC keys simultaneously. This message will not be displayed until the next time the C7 runs up.

After alarm messages have been suppressed, they are no longer stored in the alarm message buffer. They are stored in the alarm message buffer only from the time that they first occur until the time that the display is suppressed.

Note

The alarm message suppression function should be used only during the commissioning phase of the C7.

7.1.3 System Messages

Overview

System messages indicate operating states within the C7, that is, they draw your attention to malfunctions or a breakdown in communication.

Display of System Messages

This type of message has top display priority. If a fault occurs with the C7, the active event message or alarm message is removed from the display and a system message is issued in its place.

Serious and Non-serious System Messages

System messages are classified as either serious or non-serious system messages.

- **Serious** system messages. This is based on a fault which can only be rectified by a cold or warm restart of the C7.
- **Non-serious** system messages. All other errors generate non-serious system messages, that is when a print job has been initiated but a printer is not connected to the C7. Display of a non-serious system message can also be canceled automatically when a specific, configurable display time has expired.

  A list of possible system messages and their explanations can be found in Appendix C.

Inhibiting System Messages

You can inhibit the display of system messages (with the exception of internal errors 7xx) at the configuration stage. System messages whose display has been suppressed continue to be inserted into the system message buffer and can thus be viewed on the display later (by selecting the menu '**System settings → System messages**').

System Message Buffer

The system message buffer can store up to 100 messages. System messages are inserted into it with their message number and their arrival; their departure is not recorded. Some minor errors and operator errors are not logged in the system message buffer. Messages from the system message buffer are displayed in the same order as they arrive, this means first in first out, last in last out.

7.1.4 Displaying Messages

Overview

Event and alarm messages are always output to the display at message level and are displayed according to display and message priorities.

Select Message Level

You enter the message level by pressing ESC

Display Sequences

Event and alarm messages can be configured with up to 80 characters and displayed either separately or together. The differences are explained in Table 7-1 below:

Table 7-1 Explanation of Display Sequences

Display Sequence	Explanation
Separate	With this display variant only one message is shown on the display. **Exception:** On the C7-624, four messages are shown for an 8 x 40 display. If several messages are present simultaneously, the messages are displayed in succession and output according the following priorities: 1. System message - the most recent, if there are more than one. 2. Unacknowledged alarm message 3. Event message yet to depart 4. Acknowledged alarm message yet to depart 5. Standby message

Table 7-1 , FortsetzungExplanation of Display Sequences

Display Sequence	Explanation
Mixed	If there is no system message, the alarm and event messages having the highest priority are displayed simultaneously one on top of the other. For messages which extend beyond the limit of the display line, the full-length message text is repeated automatically across the display. **Exception:** On the C7-624, two alarm messages and two event messages are shown for an 8 x 40 display. If several messages are present simultaneously for the combined output, the messages are output according to the following priorities: 1. System message - the most recent, if there is more than one. 2. Unacknowledged alarm message - the second line simultaneously displays the most recent, highest priority event message yet to depart 3. Acknowledged alarm message yet to depart - the second line simultaneously displays the most recent, highest priority event message 4. Standby message

Priorities

The messages have various display priorities.

Table 7-2 Explanation of Priorities

Priority	Explanation
Display priority	System messages always have the highest display priority and unacknowledged alarm messages always have the second highest priority. Otherwise, messages are displayed in different sequences depending on whether separate or mixed output of event and alarm messages has been configured.
Message priority	You can set message priorities from 1 (low) to 4 (high) for alarm and event messages, according to their importance. If several messages of the same display priority exist simultaneously, they are displayed according to their message priorities. Those with the highest message priority are shown first and those with the lowest are shown last.

Displaying First/Last Message

If several messages having identical display and message priorities exist simultaneously, the most recent is displayed in all cases, for both event and alarm messages.

Depending on the value preset during configuration, the oldest message or the most recent message is shown for alarm messages. You can use the standard screen to modify the setting for viewing the oldest (first) or most recent (last) alarm message when there are several of them waiting. To do this, choose **System → DispMsg** and set the corresponding parameter.

Paging through Messages in the Message Level

If there are neither system messages nor unacknowledged alarm messages, you can page through the messages that have not yet departed, at message level.

To page through the messages, use the following cursor keys:

Table 7-3 Keys for Paging through Messages

Message	Key
Next alarm message	
Previous alarm message	
Next event message	
Previous event message	

The current message will be displayed again if you press ESC or there is no operator input for one minute.

Viewing Event and Alarm Messages

All configured event and alarm message text can be shown on the display, irrespective of the current event message buffer or alarm message buffer contents. To do this, please proceed as follows:

Select the standard screen **Alarms → Text** (to display alarm messages) or **Events → Text** (to display event messages).

By using the cursor keys, you can now scroll through the list of configured message texts, either upwards or downwards.

7.1.5 Viewing Message Buffers

Overview

Messages displayed on the C7 are written to the corresponding message buffer. The message buffers can be called in order to view message history by means of standard screens.

The times of occurrence, acknowledgment and departure are displayed in chronological order for all messages.

Viewing the alarm message buffer

Choose the standard screen **Alarms → View.**

The messages contained in the alarm message buffer are shown on the display in the alarm message list, together with their message number, the date and time of occurrence and the message status (arrived, departed or acknowledged) and their acknowledgement group:

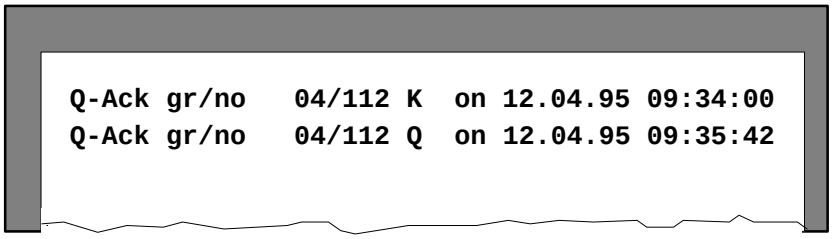


Figure 7-2 Example Display of alarm message buffer (for C7-524)

Scroll through the message text using the keys ▽ and △

Table 7-4 Explanation of Example in Figure 7-2

Part of Message	Explanation
Q-Ack gr/no	Details relating to next field: acknowledgement group and alarm message number.
04/112	The displayed alarm message belongs to acknowledgement group 04. The alarm message number is 112.
K	Message has arrived
Q	Message has been acknowledged
G	Message has departed
On Date Time	Date and time of arrival/acknowledgement/departure of alarm message.

Displaying Alarm Message Text

Display message text for marked alarm message: 

Return to alarm message list: 

Viewing Total Numbers of Messages in event message buffer

Select the standard screen **Alarms → Number**.
Here you receive an overview of all alarm messages in the buffer and the number of the alarm messages still existing.

Displaying Event Message Buffer

Select the standard screen **Events → View**.
The messages received in the event message buffer are now displayed in the event message list. They are categorized with message number, date and time, occurrence and message status (arrived or departed):

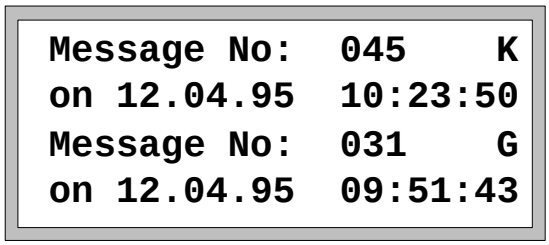


Figure 7-3 Example Display of the Event Message Buffer (C7-623)

Table 7-5 Explanation of Figure 7-3

Part of Message	Explanation
Message No.	Details relating to next field: Message number of event message
045	The displayed event message is number 045.
K	Message has arrived
G	Message has departed
On Date Time	Date and time of arrival/departure of event message.

Displaying Event Message Text

Display message text relating to selected event message: 

Return to event message list: 

Viewing the Total Number of Messages in Event Message Buffer

Select the standard screen **Event → Number**.
Here you receive an overview of all event messages in the buffer and the number of the alarm messages still existing.

Viewing the System Message Buffer

Select the standard screen **System → SysMsg**.

The messages contained in the system message buffer are now shown on the display together with details of the message number, and the date and time of their occurrence:

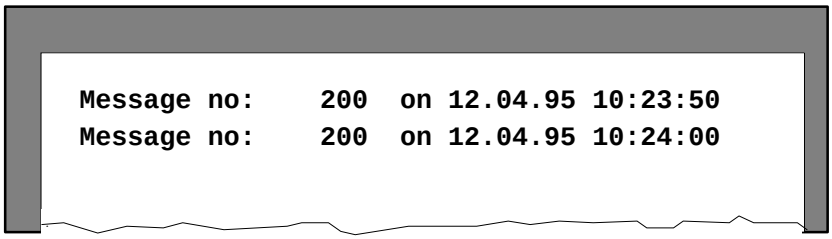


Figure 7-4 Example Display of System Message Buffer (for C7-624)

Table 7-6 Explanation of Example in Figure 7-4

Part of Message	Explanation
Message No.	Details relating to the next field: message number of the system message.
200	The displayed message number is number 200.
On Date Time	Date and time occurrence of the system message.

7

Display Message Text

Display message text of selected system message: 

Return to system message: 

7.1.6 Deleting Messages

Overview

Event and alarm messages are stored automatically in the event and alarm message buffers, respectively. Each of these buffers can contain as many as 256 events. To prevent a buffer from overflowing, you should delete event and alarm messages from the buffer.

Messages from the event and alarm message buffers are deleted either:

- By means of a standard screen or
- Automatically upon buffer overflow.

System messages are deleted automatically only when the buffer overflows.

Deleting Event and Alarm Messages in Standard Screens

Standard screens can be used to delete:

- All (not individual) acknowledged and departed alarm messages
- All (not individual) arrived and departed event messages

To delete event and alarm messages, proceed as follows:

1. Depending on the type of message you want to delete, choose one of the two standard screens, either **Alarms → Delete** (for alarm messages) or **Events → Delete** (for event messages).
2. Delete buffer: press 

   Do not delete buffer: press 

Automatic Deletion of the Alarm Message Buffer upon Buffer Overflow

By default, an overflow warning is output to the display when the alarm message buffer reaches the configured residual buffer size. Output of the overflow warning can be enabled and disabled on- line by means of the standard screen
Alarms → Overflow. If the alarm message buffer cannot accept any more messages, those which have been acknowledged and have departed are the first to be deleted automatically until the configured residual buffer size is reached.

If the residual buffer is still not free, further messages are deleted. The deleting process occurs in the following sequence:

- Alarm message acknowledged - not yet departed
- Alarm message not acknowledged - already departed
- Alarm message not acknowledged - not departed

If “overflow logging” has been configured, and an operational printer is connected to the C7, a compulsory print-out of all the deleted messages is forced.

Automatic Deletion of the Event Buffer upon Buffer Overflow

By default, an overflow warning is output to the display when the event message buffer reaches the configured residual buffer size. Output of the overflow warning can be enabled and disabled on- line by means of the standard screen **Events → Overflow**. If the event message buffer cannot accept any more messages, those which have been acknowledged and have departed are the first to be deleted automatically until such time as the configured residual buffer size is reached.

If this does not free up the residual buffer, as many of the oldest event messages are deleted as necessary to reestablish the configured residual buffer size. This deleting process is independent of the message priorities.

If the residual buffer is still not available, as many of the older quened event messages are deleted as necessary to restore the residual buffer size.

If “overflow logging” has been configured, and an operational printer is connected to the C7, a compulsory print-out of all the deleted messages is forced.

Automatic Deletion of the System Message Buffer upon Buffer Overflow

If the system message buffer is completely full, the oldest message is deleted automatically from the buffer when another system message occurs. There is no overflow warning or forced print-out of the deleted message, as is the case of event and alarm messages.

7.1.7 Printing Messages

Overview

You can print alarm and event messages by one of the following methods:

- Direct message logging
- Full print-out of the message buffer
- Forced print-out upon buffer overflow (see page 7-13 “Automatic Deletion”)

Direct Message Logging

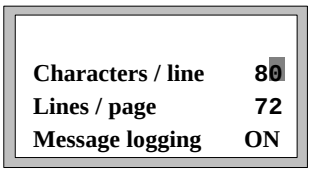
Event and alarm messages can be printed upon arrival and departure (alarm messages also upon acknowledgement), if this was set during the message configuration. System messages are not logged directly.

If direct message logging follows another print function, the header of the direct message log is re-printed.

If the printer is disabled, not ready or assigned to another print function, up to 16 messages are stored temporarily. Temporary storage of the messages for the message log is independent of the message buffer concerned.

Switching On/Off Message Logging

Direct message printing can be enabled and disabled on-line by means of a standard screen:

Step	Action	Result
1.	From the basic screen select **System** then “System Settings” Select the standard screen **Printer**.	
2.	Position the cursor on **‘Message logging’.**	Cursor is positioned on message logging**.**
3.	Select the corresponding parameter **ON** or **OFF** with SHIFT and △ or ▽	Password is requested
4.	Enter password	
5.	Please repeat operation as in Step 3 and acknowledge.	
6.	Confirm with ENTER	
7.	Leave the standard screen with ESC	Basic screen is displayed again.

Printing the Message Buffer

You can print messages from the alarm and event message buffers, but not from the system buffer:

- **Chronologically**: this means all messages contained in the buffer of the message type concerned are printed in the order in which they are situated in the buffer.
- As a **summary**: this means all messages contained in the buffer of the type of message concerned are printed in ascending message number order. If a message occurs more than once, it is slotted in according to time-related criteria.

The default setting is **chronological**. The default setting can be modified during configuration. An on-line choice of either **Chronological** or **Summary** is only possible when it has been correspondingly configured.

Activate Printer

Select the standard screen **Alarms → Print** to print out the alarm messages or **Events → Print** to print out the event messages.

7.2 Screens

Overview

The C7's process is displayed and administered on the screens (that is a processing machine or mixing station). The responsible person configures the screens for the relevant user-specific applications.

Logically associated process values are acquired and displayed on screens to provide an overview of a process or system. In addition to this alphanumeric "imaging" of the processes, the screens provide a means of entering process values and consequently a means for controlling the process. You can configure up to 99 screens for the C7.

Example: Process values on a screen can be randomly assigned to subject related groups. An example of this is shown in Figure 7-5.

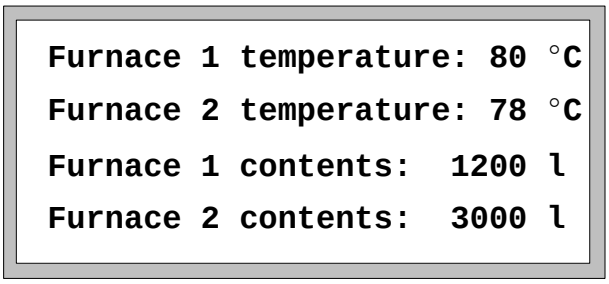


Figure 7-5 Example for Thematically Grouped Process Values (C7-623)

Screen Structure

A screen consists of the following components:

- A title (optional - max. character count: display width less 3)
- Screen entries (max. 99)
- A return destination, reached by pressing the ESC key. Possible return destinations are:
  - The message level
  - Another screen
  - A screen directory
  - The point at which the current screen was chosen

Screen Directory

During configuration, screens can be grouped in a screen directory which is used to display them on the screen or print and edit them. A screen can be retrieved from its screen directory by its screen number and its screen title (if configured).

Selecting Screens

The C7 screens can be viewed, serviced (edited) and printed. Before this is possible, they have to be selected.

A screen can be easily selected by the following means:

- Soft key
- Function key
- Directory contents
- Control job

The various selections are explained in Table 7-7.

Table 7-7 Selection Possibilities for a Screen

Selection	Explanation
Selecting with the soft key	You can branch from one screen to another using soft keys. The branch is defined in the configuration.
Selecting with the function key	The configured assignment of the function keys is valid globally, unlike the soft keys. In this way, the screen that has been permanently defined during configuration can be selected in any operating situation.
Selecting by directory	Call the standard screen via **Screens → Edit**. The screen directory is shown in the display. It contains only screens which were included in the directory during configuration. You can scroll through the screen directory using the arrow keys. A screen is chosen by pressing RETURN. As an alternative to the standard screen **Screens → Edit** you can call the screen directory using the DIR key on the system keyboard. A screen chosen with ENTER can be edited but not printed.
Selecting with a control job	For a guided operation, you can select a screen from the C7-CPU by means of a control job. In this instance, the cursor is already positioned in a specified entry or in an input field, in which the operator can perform input (see also section 9.6)

Exiting Screen Level

You leave the screen level by pressing 

.

Editing Screens

Screens can be manipulated, this means edited, by means of entries in input fields and combined input/output fields. To edit a screen proceed as follows:

Step	Action	Result
1.	Select the screen you wish to edit as described under “Select Screen”	The screen is displayed. The cursor jumps to the first input field.
2.	Move to the required field using the keys  and 	The cursor is positioned on the input field.
3.	Make the required changes as described in Section 6.4.	The cursor is positioned at the end of the input field.
4.	After confirming your input, position the cursor to make further changes if applicable.	The cursor is positioned at the end of the input field.
5.	End the editing procedure with 	You change back to the previous level.

Print Screen

To print a screen to a connected printer, please proceed as follows:

Step	Action	Result
1.	Select the screen to be printed via the standard screen **SCREENS → PRINT**.	The screen contents directory is displayed (but not when choice is the function key or a control job; in these cases, the corresponding screens are printed immediately).
2.	Scroll with the keys  or  to the required screen number, or, if configured, to the screen title.	The required screen title is displayed.
3.	Send the screen to the printer by pressing 	The selected screen is sent to the printer.
4.	Finish the action with 	You change back to the previous level.

Screen Entries

Screens consist of entries. Precisely one entry is displayed per page on the C7, irrespective of the configured number of lines. Lines which may have not been fully configured are displayed as blank lines.

An example of a screen entry is the two lines in Figure 7-5 on page 7-17.

```
Furnace 1 temperature: 80°C
Furnace 2 temperature: 78°C
```

Components of a Screen Entry

A screen entry consists of the following components:

- Static text

  The static text contains explanations for the user. You can also receive information about the soft key layout.

  – Fields for the output of:

    - Date
    - Time
    - C7-CPU setpoints

  – Fields for the input of C7-CPU setpoints, which are immediately transferred to the C7-CPU after being entered.

  – Fields for combined input/output of C7-CPU setpoints and actual values.

- Soft keys

  Different function calls for the different screens are assigned to the soft keys.

The configuration defines the intervals at which C7-CPU values are updated, this means read again from the C7-CPU and shown on the display.

Updating Values for Large Screens

With large screens containing more than 159 actual values or setpoint values. “Partial screen updating” is automatically enabled. To enhance performance, only the values are updated which are currently present on the display. This may mean that an older value is briefly displayed when scrolling through screen entries. Updating of the visible actual values is, however, faster as a result.

Input and Output Fields

Input and output fields have the following properties:

- Input fields define the setpoints in numeric or symbolic form.
- The blinking cursor can be seen in input fields.
- Output fields display actual values of the C7–CPU in numerical or symbolic form.
- Up to 256 individual texts can be configured for symbolic input and output fields. These can be selected via the C7's selection field. The selected value is taken.
- Configured number formats or limit values with respect to the number of decimal places are valid for numerical input values.

7.3 Information Text

Overview

Information text contains information about the current display contents. It is created during configuration with ProTool/Lite and provides additional information in the language set on the C7. Information text can be configured for:

- Event messages and alarm messages
- Screens
- Screen entries
- Input fields and combined I/O fields

Information text can be used to convey information to the user - for example on the current assignment of the soft keys. Information text relating to an alarm message may contain such things as additional hints about the possible cause of the alarm and how to remedy it.

Select Information Text

Configured information text is displayed by pressing the HELP key. The availability of information text is recognized by an illuminated LED.

Display Information Text

Depending on the cursor position (that is input field, screen, etc.), pressing the HELP key will react in various ways. These are described in Table 7-8.

Table 7-8 Explanation

Where is the Cursor	Explanation
Input or combined Input/Output field	If information text exists (illuminated LED), it is displayed by pressing the HELP key. Pressing the HELP key a second time displays information text associated with the screen entry (if configured).
Field without information text	If the HELP key is pressed in a field for which information text is not available, information text associated with the screen entry (if configured) is displayed.
Screen	You receive the configured information text for a screen either; • By pressing the HELP key in the directory contents when the cursor is situated on the corresponding screen, • Or within a screen, when the cursor is positioned on the screen title.

Browse/Leave Information Text

You can browse through the information text by means of the vertical cursor keys  and  .

You leave the information text screen by pressing 

. The original screen is then displayed.

7.4 Recipes

Overview

Recipes are combinations of variables for a certain purpose. The purpose of recipes is to transfer data together to the C7–CPU.

During configuration, the recipe is used to specify the data structure. This structure is then assigned with data in the C7–OP. Since multiple assignment of the data structure is possible, we will now speak of data records. Data records are stored on the C7–OP which saves memory space on the C7–CPU.

Example of a recipe

Let’s assume that orange nectar, orange drink and orange juice are to be produced using the same bottling station in a fruit juice plant. The mixing ratios are different for each drink but the ingredients are the same. The information required for production can be configured as the recipe ”Mixture”.

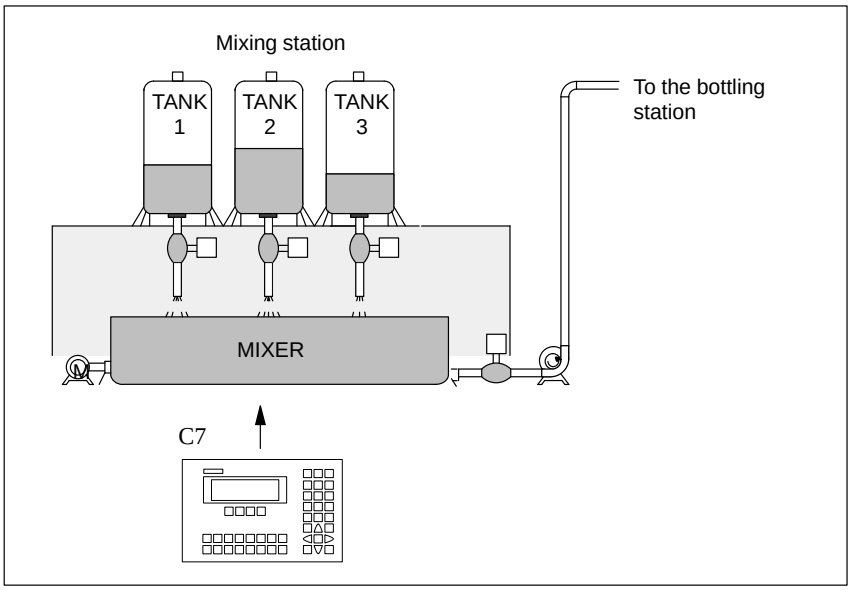
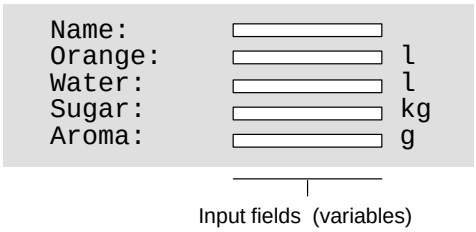


Figure 7-6 Example of a fruit juice plant

Components of a recipe

A recipe consists of a series of recipe entries. Each entry contains a maximum of one input field (i.e., variable). Depending on the configuration, either direct or symbolic value input can be used in an input field.

The recipe called *Mixture* could be set up from the following entries.



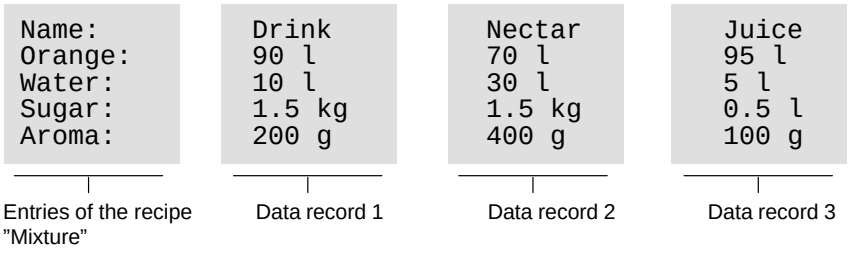
In contrast to screens, several entries are displayed on the OP simultaneously for recipes.

Data records on the OP

On the OP, the variables allocated to the input fields are assigned with values and stored on the OP.

Together, these values form one data record of the recipe.

Several data records can be set up for one recipe. This permits a fruit juice plant to operate with different types of drinks, for example, by using a different data record for each type of drink.



All data records are stored on the OP. Only the data record which is active at the moment is stored on the PLC. This saves memory space in the PLC.

Identification of recipes and data records

A recipe is identified by a recipe number and a recipe title specified during configuration.

A data record is identified by a data record number and a data record name assigned in the OP.

The clear and unmistakable identification factor for a data record is the data record number and not the data record name.

This means that, although several data records with the same name may exist, several data records with the same number cannot exist. If no name is assigned to a data record, it is assigned the name ”data record”.

Standard screens for data records

Standard screens with the following functions are available for data records.

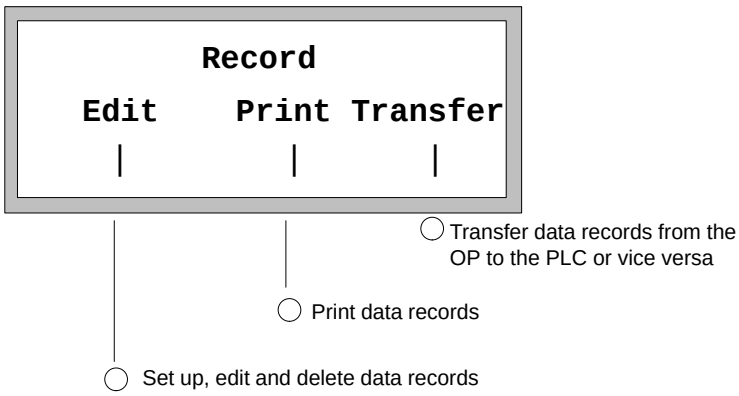


Figure 7-7 Standard screens for data records

Directory for recipes

Recipes are combined into a recipe directory which can be used to indicate them on the display, and to print and edit them. A recipe is located in the recipe directory under its recipe number and its recipe title.

Transferring data records

The C7–OP offers a transfer screen for transferring data records from the C7–OP to the C7–CPU and back. Transfer a data record to the C7–CPU when required by the C7–CPU.

The data record in the C7–CPU can be overwritten by another data record stored on the C7–OP.

The active data record can also be transferred from the C7–CPU to the C7–OP and stored on the C7–OP.

Note

Only one recipe data record at a time can be active on the C7–CPU.

Transmission of data records between C7–OP and C7–CPU is synchronized to prevent uncontrolled overwriting.

Direct/indirect transfer

Transmission of data records between C7–OP and C7–CPU can be performed ”directly” or ”indirectly”.

Direct transmission:

The variables of the data record are written directly to the defined addresses of the PLC.

Indirect transmission:

The variables of the data record are stored in an intermediate storage location on the C7–CPU.

See chapter 9 for detailed information on the internal procedures.

7.4.1 Setting Up and Editing Data Records

The OP automatically sets up a data record with the number 1 and the name ”data record” for each recipe. All values of the data records are preset with 0.

You can edit this data record. To set up more data records, either copy this data record (see chapter 7.4.2), or edit the first data record and save it under a different name.

Editing data records

Proceed as follows to change existing recipe, data record values.

1. Select standard screen *Records → Edit*
2. Select the desired recipe in the recipe directory.
3. Select the data record to be edited in the data record directory.

Note

You can have the recipe title displayed by paging up in the first entry.

---

4. Position cursor on the value to be changed.
5. Enter numeric or symbolic value.
6. Confirm the value. If you want to change more values, position the cursor in the input field of the next entry and change the value there.
7. To conclude editing data records after the last modified value has been accepted, press *ESCAPE*. You will then be asked whether you want to save the data record.
8. Save: Position cursor on the Yes field and confirm with ENTER.

   Do not save: Press ESCAPE.

   Since the data record number already exists, you will then be asked whether you want to overwrite the data record.
9. Overwrite: Press ENTER.

   Do not overwrite: Press ESCAPE.
10. Exit standard screen with ESCAPE.

As long as you have not confirmed an input value, you can terminate editing with ESCAPE. The old value is then displayed again.

Storing a changed data record as a new data record

To save a changed data record as a new data record, do not overwrite the called data record when exiting the input screen. Instead, enter a new data record number and, if necessary, a new data record name before saving the data record.

1. Position cursor in the field for the data record number, enter the number, and confirm.
2. Position cursor in the field for the data record name, enter the name, and confirm. The data record name may not exceed 12 characters.

The next time you call the data record directory, you will find the new data record listed under the new data record number.

7.4.2 Copying Data Records

Overview

There are two ways to copy a data record.

- You can call an existing data record for editing, and save it unchanged under a new number and, if necessary, under a new name. Then make your changes.
- Save an existing record via data record transfer under a new number and, if necessary, under a new name. Then make your changes in the copy.

Copying a data record

Proceed as follows to obtain a copy of a data record whose values you want to change later.

1. Call the data record to be copied via the standard screen *Records → Edit*.
2. Instead of changing a value in the data record, exit the displayed data record again immediately by pressing ENTER.

   When the standard screen is exited, you will be asked whether you want to save the data record regardless of whether values have been changed or not.
3. Position cursor in the field for the data record number, enter the number, and confirm.
4. Position cursor in the field for the data record name, enter the name, and confirm.

The next time you call the data record directory, you will find the new data record in the list.

7.4.3 Transferring Data Records

Overview

A transfer screen offering the following functions is available for transferring data records.

- C7–OP → C7–CPU

  Before a changed or newly set up data record in a recipe can become effective in the PLC, it must first be transferred via data record transfer to the C7–CPU.

- C7–CPU → C7–OP

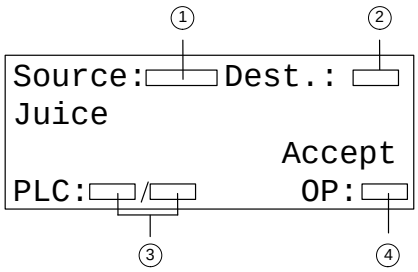
  Transfer the data record which is active in the C7–CPU to the C7–OP under a new number and a new name.

- C7–OP → C7–OP

  Copy an already existing data record on the C7–OP.

Transfer screen

The transfer screen is called via *Records → Transfer.* It is shown below.



① No. of the source data record

② No. of the destination data record

③ The recipe/data record active in the PLC

④ Recipe number selected on the OP

Figure 7-8 Transfer screen

Specifying source and destination

The data record numbers for the source data record and the destination data record must be entered in the transfer screen.

- The C7–CPU data record is always ”0” for both directions of transfer.
- The C7–OP data record must always be specified as a number > 0.

Example of entering source and destination:

C7–OP → C7–CPU Transfer		
Source	Destination	Description
11	0	Source data record **11** is an existing data record on the C7–OP and is to become the active data record on the C7–CPU (destination).

C7–CPU → C7–OP Transfer		
Source	Destination	Description
0	22	The source data record is the active data record on the C7–CPU which is to appear on the C7–OP (destination) after the transfer under data record number **22**. If data record number **22** already exists, it can be overwritten or not if requested.

C7–OP → C7–OP Transfer (i.e., copying)		
Source	Destination	Description
33	44	Source data record **33** is an already existing data record on the C7–OP which is to be copied as data record **44**. If data record number **44** already exists on the C7–OP, it can be overwritten or not if requested.

Note

Symbolic value entries in the input fields source and destination:

The data record numbers from data records already stored on the C7–OP can be taken from a data record selection list. ”C7–CPU” stands for data record 0 in the selection list.

Using the transfer screen

Proceed as follows to transfer a data record.

1. Select the standard screen *Records → Transfer.*
2. Select the recipe in the recipe directory.
3. Position the cursor on the `Source` field in the transfer screen.

   Depending on the direction of transfer desired, enter the appropriate source number (see examples), and confirm.
4. Position the cursor on the `Destination` field.

   Depending on the direction of transfer desired, enter the appropriate destination number (see examples), and confirm.
5. Position the cursor on the `Accept` field, and confirm transfer.

   During the data transfer, an asterisk (”*”) is indicated in the acceptance field.
6. Exit the standard screen with `ESCAPE`.

7.4.4 Deleting Data Records

Data records of a recipe can be deleted from the data record directory. These data records are then no longer available for the recipe.

When a data record which is active in the C7–CPU is deleted from the directory of the C7–OP, this data record still continues to be active in the C7–CPU.

How to proceed

Proceed as follows to delete data records.

1. Select *Records → Edit* in the standard screen.
2. Select the desired recipe from the recipe directory.
3. Using the cursor keys, select the data record to be deleted from the data record directory.
4. Press SHIFT + DELETE.
5. Delete: Press ENTER.

   Do not delete: Press ESCAPE.
6. Repeat steps 3 and 4 as necessary.

   If only one single data record exists, this record cannot be deleted.
7. Exit the standard screen with ESCAPE.

7.5 STATUS VAR and MOD VAR with the C7 OP

Overview

The C7 offers the possibility to display and to modify address values. This is possible by means of the configuration of special screens, similar to the PU functions STATUS VAR and FORCE VAR. Thus, during on-line operation the control addresses of the C7-CPU can be processed directly at the C7-OP.

STATUS VAR

The addresses can be displayed using STATUS VAR.

FORCE VAR

With FORCE VAR, the addresses can be displayed and their values modified and transferred back to the C7-CPU. The chosen addresses remain in the static RAM and are valid after restarting the C7.

Calling up FORCE VAR

You call the function FORCE VAR via the standard screen **ForceVAR**. After calling, the address list is displayed. Depending on the display line length a one or two line address is presented.

Figure 10-1 shows an example display for the SIMATIC S7.

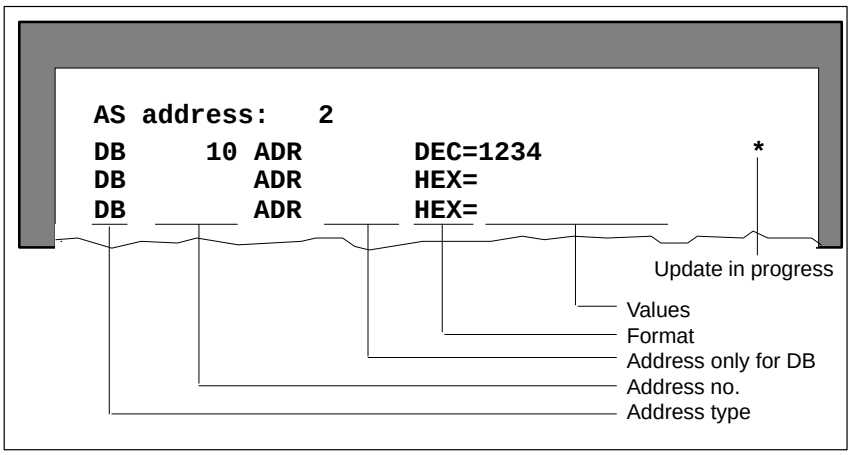


Figure 7-9 Representation of Address for the SIMATIC S7 (Example: C7-624 2x40)

AS Address

The AS address is the MPI node address of the chosen CPU. This can be adjusted.

Working with FORCE VAR

Table 10-1 summarizes the operational possibilities during the display of FORCE VAR.

Table 7-9 Operational Possibilities

Procedure	Key
General: You acknowledge every input field by field by pressing	ENTER
Position the cursor in the line using the keys	◁ ▷
In the column with the **addresses**, scroll up and down using the keys	▽ △
If the cursor is positioned in the **address** column, you can select the data type to be displayed (DB, MW, EW, AW, Z, T) by pressing the keys described above.	SHIFT
If the cursor is positioned in the **Format** column, you can set the data format to HEX, DEC, BIN, CHR, T or C, using the above mentioned cursor keys.	
Enter here the number of the address to be displayed or modified and its address, using the numeric keypad.	0....9
The cursor can be horizontally offset within the lines and the fields with **values**. 10 lines can be occupied in total. The values of the selected addresses are displayed in the value fields in a pre-defined format.	▽ △
Insert a new variable line before the current line. The cursor must be positioned on the first field. No input or updating can be commenced. **Delete** the current line. If the cursor is positioned in a value field (not BIN), and there is at least one character, the character at the cursor position is deleted.	DEL INS DEL INS

Update Values

After editing the address list, the values in the C7-CPU must be updated. This does not happen directly after confirmation of an individual value. First when you press the ENTER key after confirmation of the last entry, will the new values be transferred to the C7-CPU. During the update, a flashing asterisc appears in the top right-hand corner of the display. If the asterisc does not flash, then no logical connection to the C7-CPU has been established.

Terminate Updating Process

During the update, no inputs can be entered. The updating process can be broken with ESC.

Extended O/I Functions

8

In this Section

In this section you will discover what possibilities you have to adjust the C7-OP so that the C7-CPU process can trigger situation-dependent reactions of the C7-OP.

Overview of Section:

8.1 Process-Dependent Operator Guidance

Overview

Different action is normally required or allowed in different operating situations. To support changing requirements during process control, you can configure the following aspects, which provide the operator with purposeful, situation-specific help:

- Screen-dependent softkeys
- Global function keys
- User-defined screen hierarchies

Branching by Means of Softkeys and Function Keys

You can assign function calls to the function keys on the C7-623 and C7-624. You can distinguish here between local and global assignment.

- **Global** means that the assignment applies to the entire configuration.
- **Local** means that the assignment applies to one screen entry only.

This makes it possible for the operator to initiate functions as and when required by the situation. When the assignment of function keys vary from screen entry to screen entry, the keys are referred to as softkeys.

The following keys can be assigned as softkeys on the different devices:

Table 8-1 Keys which can be assigned as softkeys

Device	Keys	Explanation
C7-623	F1...F4	These function keys can be configured as softkeys.
	K1 ... K16	These function keys can be globally assigned.
C7-624	F1...F8	These keys (underneath the display) can be configured as softkeys.
	K1...K16	These function keys can be globally assigned.

Note

While configuring softkeys you must make sure that function calls which are required to be permanently available, this means also during screen editing, are not assigned to keys which are intended to be used as softkeys.

Key Assignment

You can assign the following functions to function keys and softkeys:

- Branch to screen level
- Select screen
- Start print job
- Display directory contents
- Modify parameters on-line

A function key or a softkey can be assigned a password level. This means that certain functions for example ”Modify parameters on-line” are only accessible to authorized persons.

Special Features with Function Keys

An LED integrated in a function key can prompt the operator if he must use this key for initiating a function in a given situation. The LED is activated by the C7-CPU.

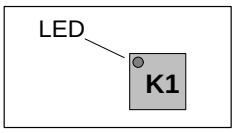


Figure 8-1 Function Key with LED

Special Features with softkeys

A bit can be configured in a variable for every softkey. This ensures that a bit is set in the C7-CPU when a softkey is pressed. In this instance, the C7-OP always transfers the whole variable to the C7-CPU instead of the individual bits and overwrites the information it contains.

The bit is reset when the key is released, or upon a change of display which modifies softkey assignment.

8.1.1 Self-defined Screen Hierarchy

Overview

The screen hierarchy can be adapted to system-specific requirements and be modified either in part or in whole. Screens can be removed or added.

Screens can be linked together in random order. The design, sequence of the link, inclusion in the screen directory and the relevant cross-jump destinations are defined during configuration with ProTool or ProTool/Lite.

You branch between the different screens by means of softkeys and configured cross-jump destinations. Branching from entries of different screens to one and the same screen is also possible (see Figure 8-2). Cross-jumps are not restricted to screen level, but can branch to the message level too.

Another feature that can be configured is the choice of picture that you want to display on the C7 as your start screen.

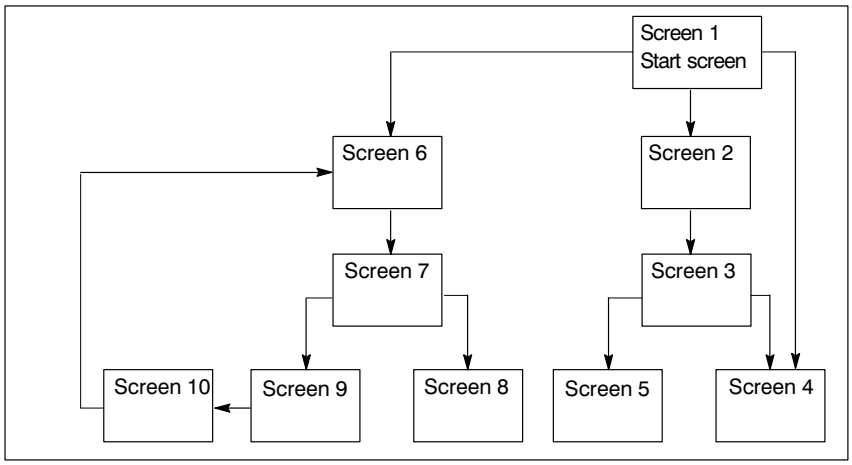


Figure 8-2 Principle of Screen Hierarchy

Example of Screen Hierarchy Design

In the following section you will receive, by way of example, an overview of a screen hierarchy design. For exact details with respect to the configuration, please refer to the ProTool or ProTool/Lite user manual.

Example:

The C7 is used to operate and monitor a system for producing and bottling different fruit juices. The system basically consists of a mixing unit and a bottling machine.

Mixing unit The ingredients of the fruit juices are contained in three tanks.

Depending on the juice that you wish to manufacture, ingredi-

ents are mixed in certain proportions.

Bottling machine After the fruit juice has been mixed, it flows into the bottling tank after a valve has been opened, and then bottled in the correct quantities. The bottles are transported on a conveyor belt. Before being filled, they are checked for damage. After they have been filled, the bottles are capped, labeled, and transferred to pallets.

Basic Screen

The C7-623's configured basic screen could look like the one shown in Figure 8-3. The basic screen consists of static text only.

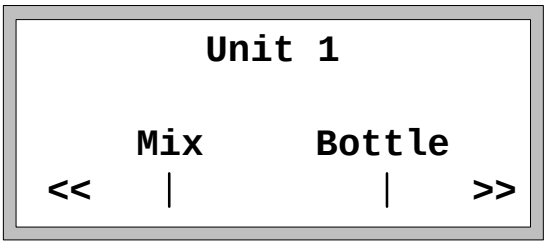


Figure 8-3 Start Screen for the Mixing and Bottling System (C7-623)

The screen segment on the display can be selected with the softkeys below the symbols. You can shift the screen horizontally using the softkeys << and >>.

Select "Mix"

Pressing the softkey beneath the "`Mix`" entry allows you to view the entry shown in Figure 8-4. It similarly consists of static text which refers to other screens ("`Tank2`", "`Tank3`" and "`Mixer`").

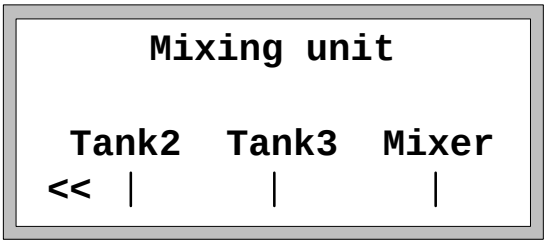


Figure 8-4 Screen with Static Text (Example)

Select "Tank2"

If you press the "Tank2" softkey, the entry displayed in Figure 8-5 appears. The entry contains the static text and two output fields (tank contents and set temperature) as well as an input field (valve position). The position of the tank valve can be set in the input field by means of a symbolic value input - for example, OPEN or CLOSED.

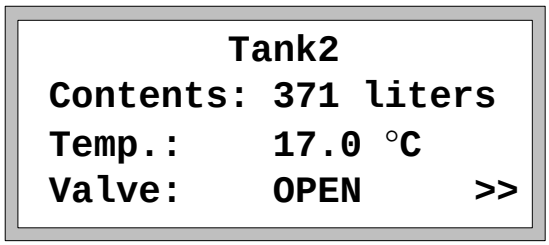


Figure 8-5 Screen with Input and Output Fields (Example)

8.2 Controlling the C7-OP from the C7-CPU

Overview

The C7-OP and the application program communicate with each other by alternately reading and writing to the data areas (memory area pointers). Thus the C7 can bring about different actions through evaluation of these user data areas.

The user data areas are described in detail in Section 9.2.

Various Actions

The table below explains which actions can be controlled.

Table 8-2 Control Functions

Actions	Explanation
Control jobs	Control jobs are functions that are initiated on the C7-CPU, for example selecting screens, printing screens, transferring the date and time (from the C7-OP to the C7-CPU). Jobs are submitted with a job number and certain parameters. The C7-CPU can call up a screen, for example, to draw the operators attention to a particular condition or to the fact that inputs are required. The required interface is described in Section 9.6. You will find a complete list of control jobs with their parameters and a description of all the user data areas that have to be configured on the C7-CPU in Appendix C.3. **Note** Jobs sent from the C7-CPU to the C7-OP can be initiated only if the C7-OP is in online mode.
LED control	The LEDs on the keys of the C7 can be driven by the C7-CPU. This way it is possible to signal to the user that a certain key press makes sense in a given situation. For LED control, it is necessary to create an LED image on the C7-CPU. In the configuration, two bits are assigned to every LED in the LED image area. These two bits can be used to implement four different states: off, steady light, low-speed flashing and high-speed flashing. The structure of the LED image is described in Section 9.4.3.

Table 8-2 Control Functions (continued)

Actions	Explanation
Evaluating the screen number	The C7 stores information about the object type called in its screen number area. Thus it is possible for information about the current display contents of the C7-OP to be transferred to the application program on the C7-CPU, and from there, for certain reactions to be triggered - for example calling another screen. The structure of the screen number area is described in Section 9.5.
Function keyboard image	By pressing a function key on the C7, you can set a bit in the function keyboard image in the C7-CPU. This bit can be evaluated via the application program. In order to achieve this, you must first create the necessary data area on the C7-CPU for the keyboard image, and specify the appropriate area pointer in the configuration for assigning the key to a bit. A maximum of two pressed keys can be transferred. The make-up of the function keyboard image is described in Section 9.4.2.
System keyboard image	A bit is permanently assigned to every key on the system keyboard (apart from the cursor keys and the directory key) in the data area for system keyboard bits. The bit remains set for as long as the corresponding key is pressed. The bit is reset when you release the key. Due to the evaluation of this data area, the operator's attention can be drawn to the wrong operation of a key - for example, by means of an error message. The make-up of the system keyboard image is described in Section 9.4.1.
Date and time	Transfer of date and time can be initiated from the C7-OP by means of a control job in order to synchronize the C7-OP and the C7-CPU.
Acknowledgement area C7-CPU→ C7-OP	By configuring a suitable data area (area pointer), alarm messages can also be acknowledged by the C7-CPU (application program) instead of from the I/O part. (See also Section 9.3).
Acknowledgment area C7-OP → C7-CPU	By configuring a suitable data area (area pointer), the C7-CPU (application program) can be told that an alarm message has been acknowledged (see Section 9.3).

8.3 Schedulers

Overview

Depending on the configuration, you can specify so–called schedulers for the C7–624. A scheduler is a regularly recurring point in time (i.e., hourly, daily, weekly, annually) at which a certain function is to be executed. Up to 48 schedulers and their corresponding functions can be configured.

A scheduler is contained in a screen entry where it can be entered, changed or deactivated when the screen is displayed.

When a scheduler time is reached, the configured bit is set in the interface area of the PLC.

See chapter 9 for detailed information on the interface area of the C7–CPU.

Functions

The following functions can be called via schedulers.

- Print alarm message buffer (chronologically or together)
- Print event message buffer (chronologically or together)
- Select screen
- Print screen
- Print data record

Changing schedulers

The schedulers have already been preset via configuration. A scheduler can be changed or deactivated via input fields in a screen entry.

The scheduler is activated as long as a time is displayed in the input field. The LED of the DELETE key goes on when a scheduler is activated. Activated means that the configured function will be executed when the scheduler time is reached. If the input field contains ???, the scheduler is deactivated.

Proceed as follows to set a scheduler time.

1. Select input field.

   You can use the cursor keys to switch between scheduler type and day of the week field, and date field and time field.

2. Enter scheduler time.

   – Date and time are entered via the system keyboard.

     Activate the SHIFT–LOCK combination to control the cursor within the input field for date or time.

   – The day of the week is set via a symbolic entry.

3. Confirm entry with ENTER.

   The scheduler is now activated.

Deactivating schedulers

You can delete the scheduler time in a screen entry by pressing the DELETE key. The scheduler is then deactivated (i.e., the configured function will not be executed).

Data Areas for Communication between the C7-OP and the C7-CPU

9

In this Chapter

In this chapter, you receive information concerning configuration parameters that are necessary for the communication between the C7-OP and the C7-CPU.

The communication is achieved using two data areas:

- The user data area

  and/or

- The interface area.

The functions, construction and special features of the various user data areas and the interface areas are described in this chapter.

In this Section

9.1 Configured Communications Parameters

Parameters

In the configuration software, the following parameters are to be set for communication via the MPI:

Note

The following parameters are already assigned generally applicable default values and need not be modified unless the C7 is used in connection with any other S7, C7 or OP units in a network.

Table 9-1

Parameter	Explanation
CPU type	Control CPU. The S7-300 is to be set for the C7-CPU. If further CPUs are connected, they must be set with S7-300 or S7-400.
CPU address	MPI address of the C7-CPU in the network configuration. The address has a default setting but can also be assigned by the user. It must be unique in a network.
Slot/subrack	Here you must set the slot and subrack. For the C7, the valid setting is: Slot 2 Subrack 0
C7-OP address	MPI address of the C7-OP in the network configuration. The address can be freely issued. It must be unique in the network configuration. The default value is address 2.
Interface	Here, you determine which interface of the OP the S7 is connected to.
Baud rate	The data transfer rate between the C7-OP and the C7-CPU is fixed at 187.5 kbaud.

Configuration Tool

All settings can be made with ProTool and ProTool/Lite, under the menu **Target System → Control**.

C7 Speciality

The first control in the list **must** always be the C7-CPU since the C7 system function “DI/DO status display” always accesses the digital I/O of the first control in the list.

9.2 Overview of User Data Areas

User Data Areas

User data areas are used to exchange data between the C7-CPU and the C7-OP. It is by means of these data areas that the C7-CPU and the C7-OP communicate.

The communication process consists of writing and reading information into and out of the data areas. Upon evaluation of the data, the C7-CPU and the C7-OP are triggered into the various actions.

The user data areas can reside in any required memory area in the C7-CPU.

Functionality

The following user data areas are possible:

- Event messages
- Alarm messages
- Recipes
- Control jobs
- System keyboard image
- Function keyboard image
- LED image
- Date and time
- Screen number area
- User version

9.3 Event and Alarm Messages

Message Triggering

Messages are triggered by setting a bit in one of the message areas in the C7-CPU. The situation of the message area is defined by the configuration tool. The corresponding area is also to be established in the C7-CPU.

As soon as the bit in the event or alarm message area of the C7-CPU is set and transferred to the C7-OP, the message is recognized as having “arrived”.

Conversely, after resetting the same bit in the C7-CPU, the message is registered in the C7-OP as having ”departed”.

Message Areas

Table KEIN MERKER represents the number of message areas for event and alarm messages and alarm acknowledgement areas, as well as the total length of all areas, for both the C7-623 and the C7-624.

Table 9-2 Message Areas of the C7-OP

Device	**Event message bit area**		**Alarm message area and alarm message acknowledgement area**	
	Number	**Length** (words)	**Number of each type**	**Total length of each type** (words)
C7-623	4	32	4	32
C7-624	4	64	4	64

Message Bit and Message Number Assignment

A message can be configured for every bit in the configured message area. The bits are assigned to the message numbers in ascending sequence.

Example:

The following event message area is configured for the C7-CPU:

`DB 60 Address 42 Length 5` (in words)

Figure 9-1 shows the assignment of all 80 (5 x 16) message numbers to the individual bit numbers in the control event message area.

The assignment follows automatically in the C7-OP.

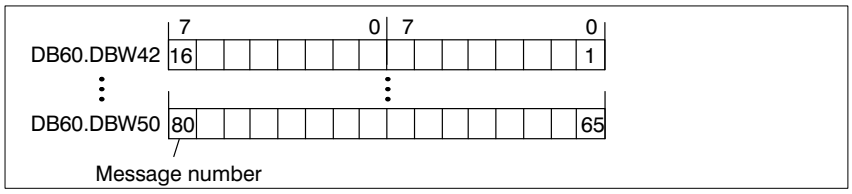


Figure 9-1 Assignment of Message Bit and Message Number

Acknowledgement

As alarm messages indicate faulty behavior of some sort, these must be acknowledged.
Acknowledgement follows either by

- Appropriate action on the C7 or
- By setting a bit in the acknowledgement area of the C7-CPU.

Acknowledgement Area

If the C7-CPU is to be informed about an acknowledgement of an alarm message at the C7-OP it self, or if the acknowledgement should be given by the C7-CPU, the corresponding acknowledgement areas are to be configured in the C7-CPU:

- **Acknowledgement area C7-OP → C7-CPU**:
  The control is informed when an alarm message is acknowledged by an operation at the OP.
- **Acknowledgement area C7-CPU → C7-OP**:
  The alarm message is acknowledged via the C7-CPU.

These acknowledgement areas are to be allocated in the configuration; - when using ProTool and ProTool/Lite, under “area pointers”.

Figure 9-2 shows schematically the individual alarm message and acknowledgement areas. The acknowledgement procedure is detailed in Table 9-4.

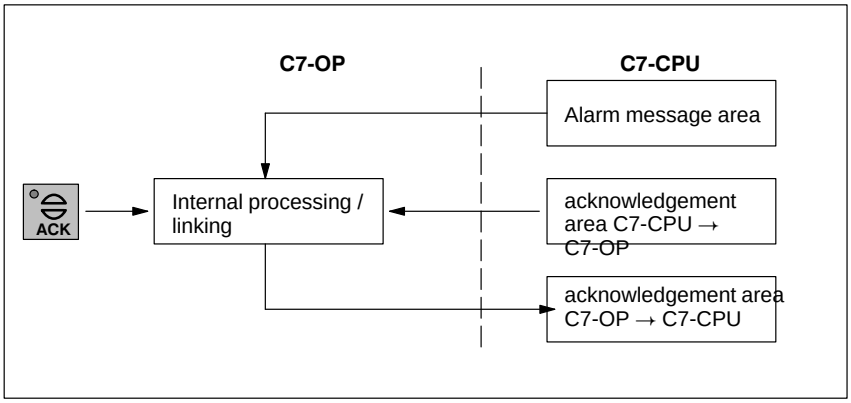


Figure 9-2 Alarm Message and Acknowledgement Areas

Table 9-3 Alarm Message Acknowledgement Procedure

Action	Reaction	Meaning
Set alarm message bit in C7-CPU	Appropriate acknowledge bit C7-OP → C7-CPU is reset	Alarm message has arrived and is unacknowledged
Set acknowledge bit in C7-CPU [1] or Acknowledgement via a C7-OP operation.	Acknowledge bit C7-OP → C7-CPU is set	Alarm message has been acknowledged
Reset alarm message bit in C7-CPU		Alarm message has departed (independent of Acknowledgement condition)

[1] If Acknowledgement is initiated via the C7-CPU, the acknowledge bit C7-CPU → C7-OP must be reset using the application program by the time a renewed alarm message comes.

Assignment of Acknowledgement Bit to Message Number

Every alarm message has a message number. To this message number, the same bit of the alarm message area and the bit multiplying of the acknowledgement area are assigned. This is also valid for more than one Acknowledgement area, if the length of the previous Acknowledgement area does not encompass the entire length of the alarm message area.

Figure 9-3 clarifies this situation.

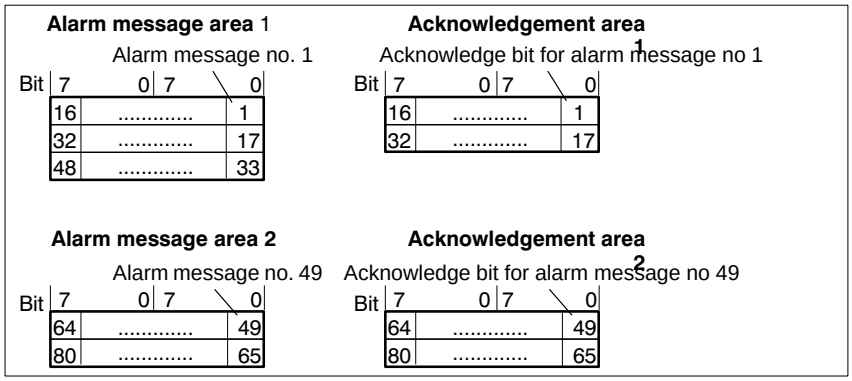


Figure 9-3 Assignment of Acknowledge Bit and Message Number

Acknowledgement Area C7-CPU → C7-OP

One of the bits set in the C7-CPU area, causes the acknowledgement of the corresponding alarm message at the C7-OP.

The Acknowledgement area C7-CPU → C7-OP

- Must be immediately connected to the relevant alarm message area
- Must have exactly the same polling time and
- Can have the same maximum length as the corresponding alarm message area.

Acknowledgement Area C7-OP → C7-CPU

If an alarm message is acknowledged at the C7-OP, the corresponding bit is set in the acknowledgement area C7-OP → C7-CPU. Thus the S7 can recognize that the alarm message has been acknowledged.

The acknowledgement area C7-OP → C7-CPU can have the same maximum length as the corresponding alarm message area.

Size of the Acknowledgement AreasC7-CPU → C7-OP and C7-OP → C7-CPU.

The acknowledgement areas may not be larger than the corresponding alarm message area. It can, however, be configured to be smaller, if it is not necessary to acknowledge every alarm message. Figure 9-4 clarifies this case.

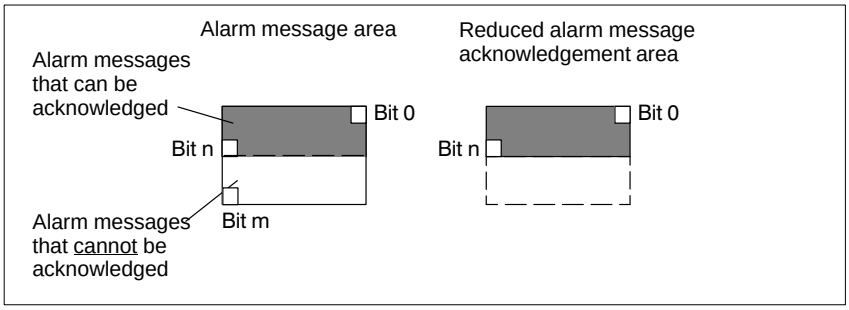


Figure 9-4 Reduced Acknowledgement Area

Note

Allocate important alarm messages, whose acknowledgement is to be signalled to the C7-CPU in the alarm message area from bit 0 in an ascending order!

9.4 Keyboard and LED Image

Application

Key presses on the C7-OP can be transferred to the C7-CPU and evaluated. This way control of an action (for instance switching on a motor) can be triggered.

The LEDs in the C7's function keys can be controlled. It is therefore possible to signal the user via a lit up LED, which key she should press, depending on the situation.

Prerequisite

In order to enable this possibility, you must configure the appropriate data areas (so-called images) in the C7-CPU, and declare them as "area pointers" during the configuration.

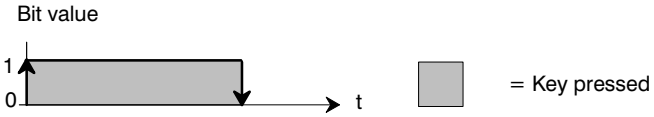
Transfer

The keyboard image is spontaneously transferred. That means the transfer always happens if a change is registered at the C7-OP. In this case you do not need to configure a polling time. A maximum of two simultaneous key presses can be transferred at one time.

Value Allocation

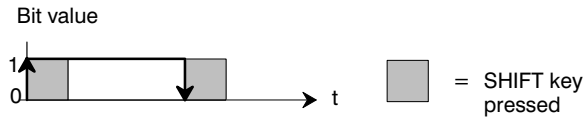
- **All keys (except SHIFT key)**

  As long as the corresponding key is pressed, the assigned bit in the keyboard image has the value 1, otherwise it has the value 0.



- **SHIFT key**

  When the SHIFT key is first pressed, the assigned bit in the keyboard image receives the value 1. This state remains also after letting the key go until the SHIFT key is pressed again.



9.4.1 System Keyboard Image

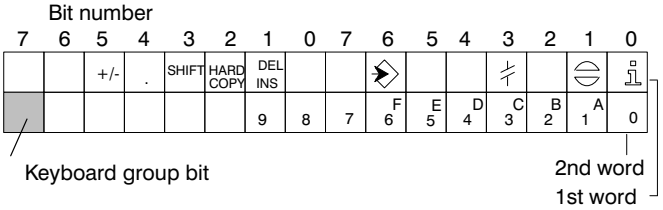
Layout

The system keyboard image is a data area with a fixed length of **two** data words.

Every key of the system keyboard is assigned exactly one bit in the system keyboard image, with the exception of the DIR key and the cursor keys.

The system keyboard image must also be declared as “area pointer, type: system keyboard” in the configuration. This image can be assigned **once** only and in **one** CPU.

Keyboard image:



Note

Unused bits may not be overwritten by the application program.

Keyboard Group Bit

The keyboard group bit serves as the control bit. It is set to the value `1` for each transfer of the keyboard image from C7-OP to C7-CPU. After evaluation of the data area by the application program, it should be reset.

By regular reading of the group bits, you can determine in the application program, whether the image of the system keyboard has been newly transferred.

9.4.2 Function Keyboard Image

Data Areas

The image of the function keyboard can be categorized into separate data areas:

- Maximum number of data areas - 4
- Total length of all data areas (words) - 4

Key Assignment

The assignment of the individual keys to the data area bits is defined during the configuration of the function keys. The number within the image area is declared for every key.

The function keyboard image must also be declared in the configuration, under “area pointer, type: function keyboard”.

Keyboard Group Bit

The most significant bit in the last data word of **every** data area is the keyboard group bit. It serves as a control bit. This bit is set to 1 for every transfer of the keyboard image. After evaluation of the data area by the application program, the keyboard group bit should be reset.

You can determine whether a block is newly transferred, by regularly reading the group bits using the application program.

9.4.3 LED Image

Data Areas

The LED image can be categorized into separate data areas.

- Maximum number of data areas: 4 (e. g. 4 different data areas in varions CPUs)
- Total length of all data areas (words):

LED Assignment

The assignment of the individual LEDs to the data area bits is defined during the configuration of the function keys. The bit number within the image area is declared for every LED.

The bit number (n) denotes the first of two consequential bits, which can control four different LED states in total:

Bit n + 1	Bit n	LED Function
0	0	Off
0	1	Flashing at about 2 Hz
1	0	Flashing at about 0.5 Hz
1	1	Continually on

9.5 Screen Number Area

Application

The C7-OP stores information concerning the C7-OP's called up screen in the screen area number.

It is thus possible to transfer information about the current display contents from the C7-OP to the C7-CPU, and from there, to trigger certain reactions, like calling up another screen for example.

Prerequisite Requirement

If the screen number area is to be used, it must be declared as an "area pointer" in the configuration. It can only be placed **once** and in **one** C7-CPU.

The screen number area is spontaneously transferred. That means the transfer always happens if a change is registered at the C7-OP. In this case you do not need to configure a polling time.

Layout

The screen number area is a data area with a fixed length of 2 data words.

The layout of the screen number area for the C7-OP in the control memory is represented below.

	7 0	7 0
1st word	Current screen type	Current screen number
2nd word	Current Input number	Current Input field number

Input	Assignment
Current screen type	1: Screen
Current screen no.	1 to 99
Current input number	1 to 99
Current input field number	0 to 8 0: Input field number

All bytes of the screen number area are assigned with FF_H in the message level and for the display of a contents directory.

For **special screens** (see Section C. 3) the screen number area is assigned as follows:

	7 0	7 0
1st word	3	Special screen number
2nd word	FF_H	Current input field number

9.6 Interface Area

Overview

The interface area is only necessary for the C7-CPU if its functions are to be used or evaluated by the C7-CPU.

The interface area must be configured if you want to use the following functions:

- Send control jobs to the C7-OP
- Synchronize data and time between the C7-CPU and the C7-OP
- Evaluate coupling identifier
- Recognize C7-OP startup in C7-CPU program
- Evaluate C7-OP operation mode in C7-CPU program
- Evaluate C7-OP's ready bit in C7-CPU program

Layout of the Interface Area

Figure 9-5 shows the layout of the interface area. You can define the interface area in a data block or a marker area. The address of the interface area is to be declared in the configuration. This is necessary so that the OP knows where to put the data.

The interface area is to be determined once per CPU.

Interface area:

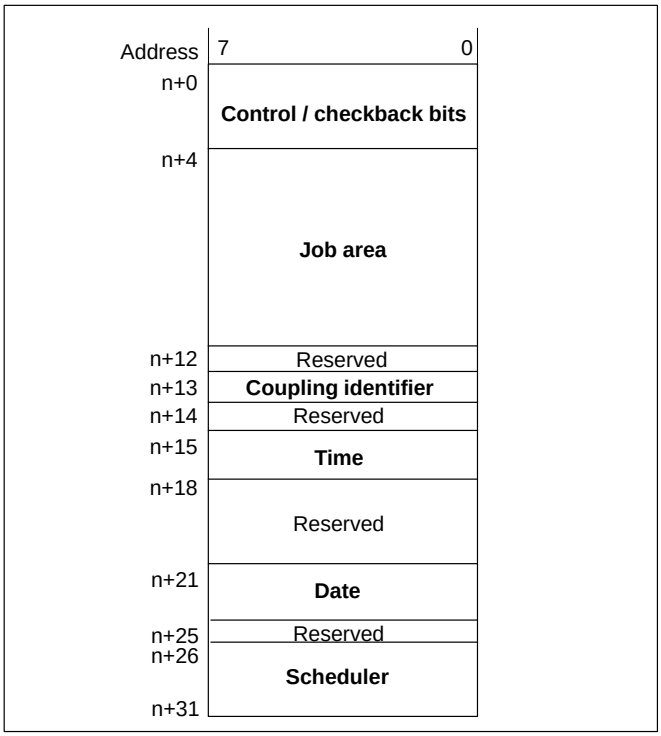


Figure 9-5 Layout of the Interface Area for the C7-CPU

9.6.1 Control and Checkback Bits

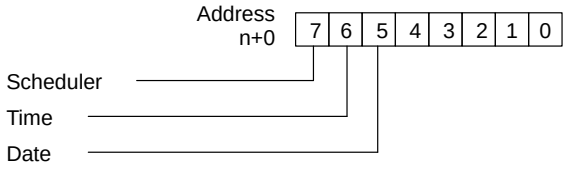
Introduction

Three bytes are available in the interface area for the control and checkback bits. Bytes n+0 and n+1 are used to coordinate between the C7-OP and the C7-CPU. Byte n+3 is required for the transfer of data records and indirect variables.

Byte n+1 and n+2 are described below:

Description of byte n+0

The structure of byte n+0 is shown below. The diagram is followed by a description of the individual bits.



Bits 5–6 Date/time 1 = New

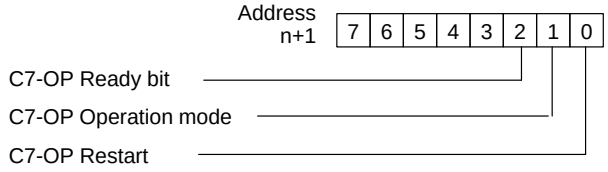
The transfer of the date and time from the C7–OP to the C7–CPU can be ini tiated by means of C7–CPU job 41. These bits are set by the C7–OP if a new date or a new time is transferred. The bits must be reset in the S7 program after the date or time has been evaluated.

Bit 7 Scheduler bits: 1 = New

Scheduler bits are possible only for the C7–624.
If the C7–624 has set a new scheduler bit in the interface area, it also sets the corresponding bit in the control and acknowledge bits. You therefore need only to poll this bit to detect modification of the scheduler bits. Following evaluation, the bit has to be reset in the C7–CPU.

Description of byte n+1

The following diagram shows the layout of byte n+1. Afterwards follows the description of the individual bits.



Bit 0 C7-OP restart
1 = C7-OP has been started

Bit 0 is reset upon completion of the restart. You can reset the bit in the C7-CPU program and thus recognize when the C7-OP is being restarted.

Bit 1 C7-OP operation mode
1 = C7-OP in off-line mode
0 = C7-OP in normal mode

Bit 1 is set if the user switches the C7-OP to off-line. The bit has a value 0 in an on-line condition.

Bit 2 C7-OP ready bit
The C7-OP inverts the ready bit within the interval of 1 second. In the C7-CPU program, you can recognize whether a connection to the C7-OP exists.

Description of byte n+3

Byte n+3 serves to synchronize the transfer of data records and indirect vari ables. The meanings of the individual bits are described below. Exactly how the transfer works is described in Chapter 9.7.3.

Bit 0 `1` = Data mailbox is disabled (is set only by the C7–OP)
`0` = Data mailbox is enabled

Bit 1 `1` = Data record/variable is errored

Bit 2 `1` = Data record/variable is correct

Bit 3 `1` = Data transfer terminated successfully

Bit 4 `1` = Request data record/variable

Bit 5 `1` = C7–OP should read the data mailbox

Bit 6 `1` = Request data mailbox disable

9.6.2 Data Areas in the Interface Area

Overview

In this section, the layout and usage of the interface data areas is described.

The C7-CPU initiates an action to be carried out on the C7-OP by means of the ?job area?. All other bytes are areas in which the C7-OP writes data. These areas can be evaluated by the C7-CPU program. The bytes are individually described below.

Job Area

Byte n+4 to n+11:
The C7-OP is passed on the control jobs via the job area. The actions on the C7-OP are initiated in this way.

The job area consists of four words. In the first word of the job area is the job number. In further words, the job parameters are to be entered (maximum 3).

Address	7 0	7 0
n+4	Job number.	
	Parameter 1	
	Parameter 2	
n+10	Parameter 3	

If the first word of the job area is not equal to zero, the C7-OP evaluates the control job. After that, the C7-OP sets this data word to zero. For this reason, the parameters must first be entered in the job area and only then the job number.

The possible control jobs are listed with job numbers and parameters in Appendix 2.

Coupling identifier

Byte n+13:
The C7-OP enters the coupling identifier in byte 13.

Date and Time

Time = byte n+15 to n+17
Date = byte n+21 to n+24
The date and time can be transferred from the C7-OP to the C7-CPU via control job 41.

The following screens show the layout of the data area. All inputs are coded in Binary Coded Decimal (BCD).

Time:

Address	7	0
n+15	Hours (0...23)	
n+16	Minutes (0...59)	
n+17	Seconds (0...59)	

Date:

Address	7	0
n+21	Weekday (1...7)	
n+22	Day (1...31)	
n+23	Month (1...12)	
n+24	Year (0...99)	

Scheduler bits (C7–624 only)

Byte n+26 to n+31:
A scheduler time is a periodically recurring point in time (hourly, daily, weekly, annually) at which a predefined function is executed – for example:

- print message buffer or screen
- select screen.

When a scheduler time is reached on the C7–OP, the corresponding bit is set in this area:

Address	7	0	7	0
n + 26	16			1
	32			17
n + 30	48			33

Scheduler No.

9

9.6.3 Example for activating a control job

Control Job Procedure

Procedure for the activation of a control job:

1. You must configure the interface module (which contains the interface area) in the C7-CPU (example DB52).
2. You enter the interface module (52) into the “area pointer” during the configuration. The interface module is thus made known to the C7-OP.
3. The job is entered in the interface module via the C7-CPU program (see fig 9-6).
4. The C7-OP reads the interface module and carries out the job.

Program for Screen Selection

AUF DB	52	Call interface module.
L	screen no.	Enter first parameter (screen no.) into
TDBW		6 job area of the interface module.
L	Input number.	Enter 2nd parameter into
TDBW		8 job area of the interface module.
L	Field no.	Enter 3rd parameter (field no.) into
TDBW		10 job area of the interface module.
L	51	Enter the job no. into the job area of the interface module and activate the job.

Figure 9-6 Example for a C7-CPU Program

9.7 Recipes

Definition

A recipe is a group of variables for a fixed data structure. You set this structure in your configuration and assign data to it on the C7–OP. You cannot modify the structure on the C7–OP later.

Since the data structure can be assigned several times, we refer to data records. These data records are stored (created), loaded, deleted and modified on the C7–OP. The data are stored on the C7–OP, thus saving memory on the C7–CPU.

The use of recipes insures that, when a data record is transferred to the C7–CPU, several items of data are transferred to the S7 **together** and in a **synchronized** fashion.

Transferring data records

Data records can be transferred from the C7–OP to the C7–CPU or from the C7–CPU to the C7–OP. You transfer data records from the C7–OP to the C7–CPU to set specific values on the C7–CPU – for example, to produce orange juice. It is similarly possible to fetch data from the C7–CPU and to store them on the C7–OP as a data record to save, say, a favorable assignment of values.

Synchronization

To insure a coordinated procedure for transferring data records and to prevent any uncontrolled overwriting of data, bits are set in control and response byte 3 of the interface area.

9.7.1 Transferring Data Records

Übersicht

When a data record is written, the variables in the data record are written directly to the defined addresses concerned. With direct reading, the variables are read into the C7–OP from the system memories of the C7–CPU.

With ProTool, the variables must have a direct link to the C7–CPU and the `write directly` attribute for direct transfer. Variables not having an assigned address on the C7–CPU are not transferred.

9.7.2 Addressing Recipes and Data Records, and the Requisite Data Areas

Adressing/ Data areas

During configuration, the recipe is given a name and a number. Both the recipe name and the recipe number can be seen on the C7–OP.

The data records you create on the C7–OP are similarly given a name and a number.

When a data record transfer is initiated from the C7–OP to the C7–CPU, the recipe name and the data record number are transferred to the C7–CPU together with the data. For this, you have to create a data mailbox on the C7–CPU. Use the same specifications as were set in the configuration under *Area Pointer*. The values of the data record are written directly to the addresses on the C7–CPU.

Data mailbox:

1st word	Recipe Number
2nd word	Reserved
3rd word	Reserved
4th word	Data record Number
5th word	Reserved

9.7.3 Synchronization during Transfer – Normal Case

Transferring data records

The control and response bits in the interface area synchronize data record transfer. A transfer is normally initiated by an operator input on the C7–OP.

Transfer C7–OP → C7–CPU (initiated on C7–OP)

The following description shows the procedure by which the C7–OP sets synchronizing bits and the manner in which the C7–CPU program has to react to them.

Table 9-4 Procedure for Transmission

Step	Erklärung
1	Bit 0 is checked by the C7–OP. If bit 0 is set to `1` (= data mailbox disabled), the transfer is terminated with a system error message. If bit 0 is set to `0`, the C7–OP sets the bit to `1`.
2	The C7–OP enters the identifications in the data mailbox. With an indirectly transferred data record, the data record values are also written to the data mailbox. With a directly transferred data record, the values of the variables are written to the configured address.
3	The OP sets bit 3 to `1` (= data transfer terminated).
4	The data record or the variable can be evaluated in the C7–CPU program. You then have to acknowledge in the C7–CPU program whether the transfer was error-free or erroneous. Error-free: bit 2 is set to `1` Erroneous: bit 1 is set to `1`
5	Reset bit 0 in the C7–CPU program.
6	The C7–OP resets the bits set in step 3 and step 4.

9.7.4 Synchronization during Transfer – Special Cases

Transfer C7–OP → C7–CPU (initiated by C7–CPU)

Make sure with this type of transfer that the current variables on the C7–OP are transferred. The values are not read directly from the data medium.

Table 9-5 Procedure for Transmission

Step	Erklärung
1	In the S7 program, request the data mailbox lockout by setting bit 6 to `1`.
2	If lockout is possible, the C7–OP sets bit 0 to `1` and simultaneously resets bit 6 to `0`.
3	In the S7 program, inform the C7–OP through the data mailbox which data record it should transfer. To do this, enter the identifications of the recipe in the data mailbox.
4	Set bit 4 to `1` (= request data through data mailbox) in the S7 program.
5	The C7–OP reads the data mailbox.
6	The C7–OP resets bit 4 and transfers the data record or the variable as described for case 1 from step 2 onwards.

9.8 Notes on Optimization

Decisive Factors

The construction of the user data area described in section 9.3 and the configured polling times in the **area pointers** are significant factors for the **actual** reachable update times. The update time is the polling time plus the transfer time plus the processing time.

To reach an optimal update time, the following points should be observed during the configuration:

- Configure the individual data areas as large as necessary, but as small as possible.
- Define the data areas which belong together contiguously. The actual update time improves if you configure **one** large area as opposed to several small areas.
- The total performance is unnecessarily compromised if the polling times are too small. Set the polling time according to the changing speed of the process values. The change in temperature of an oven for example, is significantly more sluggish than the change of rotary speed in an electrical drive.
- The approximate value for the polling time is around 1 second.
- Avoid cyclic transfer of the user data area (Polling time 0). Instead of this use the control jobs to transfer the user data areas spontaneously.
- Place the variables of a message or a screen without gaps in one data area.
- In order that changes in the C7-CPU are recognized by the C7-OP, they must at least be impending during the actual polling time.

SFCs and SFBs in the C7-CPU

A

Introduction

The C7–CPU provides you with various system functions, for example, for program handling and diagnostics. You invoke these system functions in your user program with the number of the SFC or SFB.

You will find detailed descriptions of all system functions in the Reference Manual **/235/**.

Real-Time Clock Function

For the clock functions, the CPU offers you the following system functions.

SFC	No.	Name	Description	Execution Time
SFC	0	SET_CLK	Setting the clock time If the clock to be set is a master clock, the clock time synchronization is triggered simultaneously. If the clock to be set is a slave clock, only the clock time is set.	120 µs
SFC	1	READ_CLK	Reading the clock time	190 µs
SFC	2	SET_RTM	Setting the operating hours counters In the C7–CPU, you can set 1 operating hours counter.	65 µs
SFC	3	CTRL_RTM	Starting and stopping the operating hours counter	55 µs
SFC	4	READ_RTM	Reading the operating hours counter	90 µs
SFC	64	TIME_TICK	Reading out the system time You can read out the system time with an exactness in the ms range.	45 µs

Block Functions

The following table contains system functions for copying and setting array default variables.

SFC	No.	Name	Description	Execution Time
SFC	20	BLKMOV	Copying variable of random type	90 µs + 2 µs/byte
SFC	21	FILL	Setting array default variables	90 µs + 3.2 µs/byte

Creating a Data Block

You create a data block using SFC 22 “CREAT_DB”.

SFC	No.	Name	Description	Execution Time
SFC	22	CREAT_DB	Generate a data block of specified length in a specified area	110 µs+ 3.5 µs per DB in the specified area

Time-of-Day Interrupt Functions

You can use the time-of-day interrupts for program processing controlled by the C7–CPU-internal real-time clock.

SFC	No.	Name	Description	Execution Time
SFC	28	SET_TINT	Setting the times for a time-of-day interrupt	190 µs
SFC	29	CAN_TINT	Canceling the times for a time-of-day interrupt	50 µs
SFC	30	ACT_TINT	Activating a time-of-day interrupt	50 µs
SFC	31	QRY_TINT	Querying the status of a time-of-day interrupt	85 µs

Delay Interrupts

Delay interrupts start the operating system at the end of a specified time.

SFC	No.	Name	Description	Execution Time
SFC	32	SRT_DINT	Start a delay interrupt	85 µs
SFC	33	CAN_DINT	Cancel a delay interrupt	50 µs
SFC	34	QRY_DINT	Query started delay interrupts	80 µs

Interrupt and Error/Fault Handling

The C7–CPU provides you with the following system functions for responding to interrupts and errors/faults:

SFC	No.	Name	Description	Execution Time
SFC	36	MSK_FLT	Masking sync faults	150 µs
SFC	37	DMSK_FLT	Enabling sync faults	160 µs
SFC	38	READ_ERR	Reading and erasing programming and access errors that have occurred or have been disabled	160 µs
SFC	39	DIS_IRT	Disabling the handling of new interrupts	215 µs
SFC	40	EN_IRT	Enabling the handling of new interrupts	305 µs
SFC	41	DIS_AIRT	Delaying the handling of interrupts	35 µs
SFC	42	EN_AIRT	Enabling the handling of interrupts	35 µs
SFC	43	RE_TRIGR	Re-triggering the scan time monitor	30 µs
SFC	44	REPL_VAL	Copying a substitute value into accumulator 1 of the level causing the error	45 µs

Status Changes

You can influence the CPU status with the following system functions:

SFC	No.	Name	Description	Execution Time
SFC	46	STP	Forcing the CPU into the STOP mode	–
SFC	47	WAIT	Implementing waiting times	200 µs

Address Allocation

For allocating the free address of a module to the associated rack and slot, you can use the following SFCs.

SFC	No.	Name	Description	Execution Time
SFC	5	GADR_LGC	Determining the free address of channel x of the signal module in slot y.	–
SFC	49	LGC_GADR	Converting a free address into the associated slot and rack number of a module	140 µs
SFC	50	RD_LGADR	Calculating all the predefined free addresses of a module	190 µs

Diagnostics Functions

You can use the following system functions to read and write diagnostics information:

SFC	No.	Name	Description	Execution Time
SFC	51	RDSYSST	Read the information out of the system status list	280 µs + 200 µs/data set
SFC	52	WR_USMSG	Write specific diagnostics information into the diagnostics buffer	110 µs

Module Initialization Functions

The C7–CPU places the following system functions at your disposal for writing and reading initial module parameters.

SFC	No.	Name	Description	Execution Time
SFC	55	WR_PARM	Writing dynamic parameters to a module	1.6 ms
SFC	56	WR_DPARM	Writing predefined parameters to a module	1.75 ms
SFC	57	PARM_MOD	Assigning a module's parameters	2.2 ms
SFC	58	WR_REC	Writing a module-specific data record	1.4 ms + 32 µs/byte
SFC	59	RD_REC	Reading a module-specific data record	0.49 ms

System Status List in the C7-CPU

B

Introduction

The C7-CPU is able to provide you with certain information. The C7-CPU stores this information in the “system status list”.

This appendix contains the sublists of the system status list provided by the C7-CPU.

Definition

The system status list contains data describing the current status of an C7-CPU. You can use it to gain an overview at any time of the following:

- The current parameterization of the CPU and the parameterizable signal modules
- The current statuses and sequences in the CPU and the parameterizable signal modules.

See the *STEP 7 Standard and System Functions* Reference Manual for a detailed description of the structure of the system status list and all possible entries.

Reading the System Status list

You can use SFC 51 “RDSYSST” from the user program to read out the entries in the system status list (see the Reference Manual **/235/**).

Sublists

The system status list is divided into sublists. This makes it possible to target specific information in the system status list.

Structure of the Sublists

Each sublist contains:

- Header information of 4 data words
- A specific number of records containing the event information

Header Information

The header information of a sublist is 4 data words long. Figure B-1 shows the contents of the header information of a sublist.

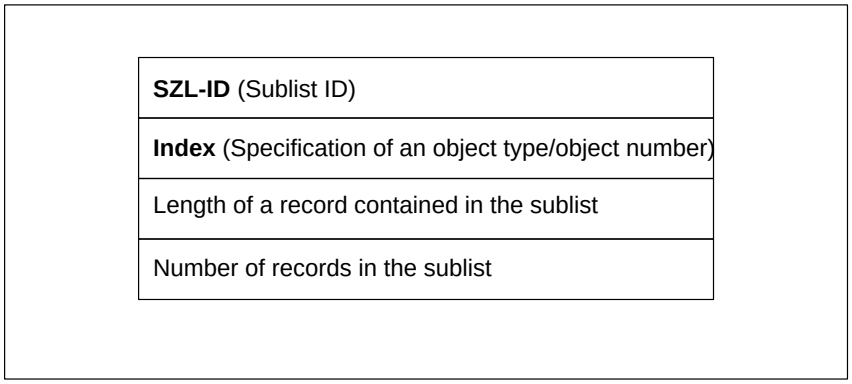


Figure B-1 Header Information of a Sublist of the system status list

SZL ID

Each sublist has an “SZL ID”. In addition, it is possible to read only one excerpt from a sublist. The ID of this excerpt from the sublist is also contained in the “SZL ID”. Figure B-2 shows the structure of the “SZL ID” for the CPUs.

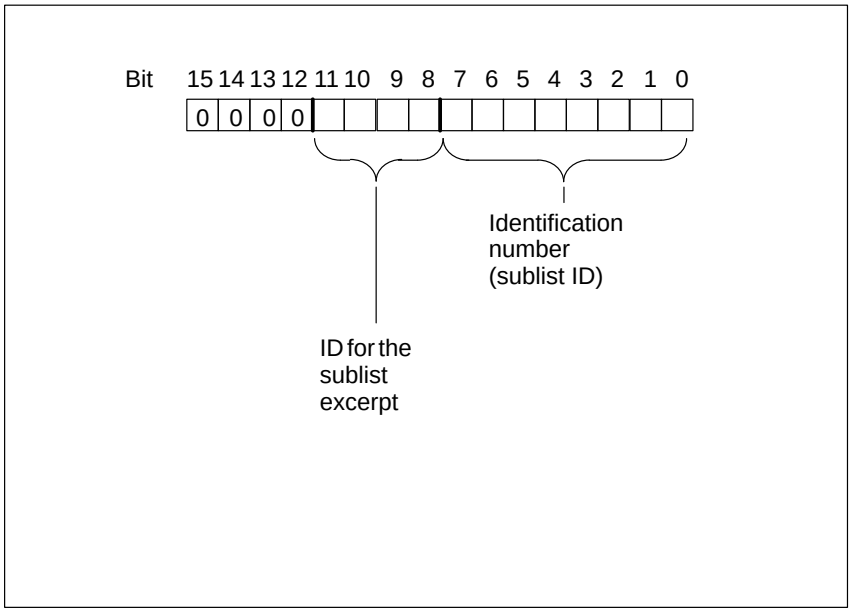


Figure B-2 Structure of the “SZL ID” of the Sublist

B

ID for the Sublist Excerpt

You use the ID for the sublist excerpt to select the extent of the sublist to be output.

- **0_H**: The complete sublist is output
- **1_H to E_H**: A special sublist is output
- **F_H** only header information is output

Index

You must assign an index if you want to read out only **one** specific record from the sublist.

Length of the Subsequent Records

This data word shows how much information (in bytes) a record of the sublist contains.

Number of Records

This data word shows how many records the transferred sublist contains.

Table of Sublists

Table B-1 below shows the individual sublists of the system status list with the entries relevant for the individual C7-CPUs.

Table B-1 Sublists of the System Status list of the C7-CPU

SZL_ID	Sublist	Index (= ID of the Individual Records of the Sublist)	Record Contents (Sublist Excerpt)
	C7-CPU identification	-	CPU type and version number
0011_H	All records of the sublist		
0111_H	One record of the sublist		
	C7-CPU features		
0012_H	All records of the sublist		
0112_H	Only those records of a group of features	0000_H	STEP 7 processing
		0100_H	Time system in the C7-CPU
		0300_H	STEP 7 instruction set
0013_H	**User memory areas**	01_H	Working memory
		02_H	Integrated load memory
		05_H	Size of backup memory

Table B-1 Sublists of the System Status list of the C7-CPU (Continued)

SZL_ID	Sublist	Index (= ID of the Individual Records of the Sublist)	Record Contents (Sublist Excerpt)
0014_H	**Operating system areas**	0001_H	Process image of the inputs (number in bytes)
		0002_H	Process image of the outputs (number in bytes)
		0003_H	Number of flags
		0004_H	Number of timers
		0005_H	Number of counters
		0006_H	Size of the I/O address area
		0007_H	Entire local data area of the C7-CPU (in bytes)
	Block types		
0015_H	All records of the sublist		
0115_H	One record depending on the 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)
	Loadable SDBs	SDB Number	-
0017_H			
0117_H			
	Rack information		
0018_H	All records of the sublist		
0118_H	One record depending on the index	0000_H	Rack 0
		0001_H	Rack 1
		0002_H	Rack 2
		0003_H	Rack 3
	Interrupt/error assignment via number of assigned OBs	-	-
0021_H	Records of all possible interrupts		
$0A21_H$	Records of all assigned interrupts		
	Interrupt status		
0222_H	Record for the specified interrupt	0001_H	Interrupt class of free cycle
		5050_H	Interrupt class of asynchronous interrupts
	Priority class		
0023_H	Records for all priority classes	0000_H	Priority of possible OBs
	Only sublist header info		

B

Table B-1 Sublists of the System Status list of the C7-CPU (Continued)

SZL_ID	Sublist	Index (= ID of the Individual Records of the Sublist)	Record Contents (Sublist Excerpt)
	Operating statuses of the C7-CPU		
0024_H	Information on all stored operating status transitions		
0124_H	Information on the last executed operating status transition		
0424_H	Information on the current operating status		
0524_H	Information on the operating status specified	5000_H	STOP status
		5010_H	STARTUP status
		5020_H	RUN status
0131_H	**Communication performance parameters** on the communications type specified	0001_H	Number of connections, baud rates
		0002_H	Test and startup parameters
		0003_H	Operator interface (parameters)
		0005_H	Diagnostics functions and diagnostics entries
		0007_H	Communications via global data (parameters)
		0008_H	Operator interface (time specifications)
0132_H	**Communications status information** on the communications type specified	0001_H	Number and type of connections
		0002_H	Number of test jobs set up
		0003_H	Number of current cyclic operator interface tasks
		0004_H	Protection levels of the C7-CPU
		0005_H	Diagnostics status data
		0007_H	Communications via global data
		0008_H	Cycle time, correction factor, operating hours counter, date/time of day
		0009_H	Set baud rate via the MPI
$0D91_H$	**Module status information** of all modules in the rack specified		Features/parameters of the module plugged in
		0000_H	Rack 0
		0001_H	Rack 1
		0002_H	Rack 2
		0003_H	Rack 3
	Diagnostics buffer	x	Event information
$00A0_H$	All entered event information		The information in each case depends on the event
$01A0_H$	The x latest information entries		
	Module diagnostics	Module rack and slot number	Module-dependent diagnostics information
$00B2_H$	Complete module-dependent record of the module diagnostics information		

C7-OP Functionality/Standard Screens/ Control Jobs/System Messages

C

In this Appendix In this appendix, you will obtain information concerning:

C.1 C7-OP Functionality

Table with List of Functions

The table below provides an overview of the functions of C7-623 and C7-624 with their different versions.

Table C-1 Functions of C7-623 and C7-624

	C7-623	C7-624
Display		
– Design	LCD	A, C: LCD A-VF: VF
– Lines x characters per line/character height (mm)	4x20/5	4x20/8 or 8x40/4.5
– Contrast control	x	x
Event Messages		
– Maximum number	499	999
– Maximum length (characters)	80	80
– Display	x	x
– Print	x	x
– View event message text	x	x
– Maximum number of entries in the event buffer	256	256
– View event buffer	x	x
– Print event buffer	x	x
– Delete event buffer	x	x
Alarm Messages		
– Maximum number	499	999
– Maximum length (characters)	80	80
– Display	x	x
– Print	x	x
– View alarm message text	x	x
– Maximum number of entries in alarm buffer	256	256
– View alarm buffer	x	x
– Print alarm buffer	x	x
– Delete alarm buffer	x	x
Message Acquisition		
In buffer with date, time, state	x	x

Table C-1 Functions of C7-623 and C7-624

	C7-623	C7-624
Setpoint Input		
– Number or letters	x	x
– Symbolic variables	x	x
Display of Actual Value (numerical and symbolic)	x	x
Combined Actual Value Display and Setpoint Input	x	x
Limit Value Check of Operator Input	x	x
Password Protection (also by means of authorization input)	x	x
Screens		
– Maximum number	99	99
– Display	x	x
– Print	x	x
– Screen entries per screen	99	99
– Maximum number of fields per screen	300	300
– Maximum number of fields per screen entry	32	32
Recipes		
– Maximum number	99	99
– View	x	x
– Print	x	x
– Recipe entries per recipe	99	99
– Recipe memory size (in Kbytes)	4	20
– Data records per recipe (maximum)	99	99
– Save/fetch data record in/from the C7–OP	x	x
Information Text		
– Maximum length (characters)	320	320
Scheduler times	-	48
Function Keys		
– Number	6	16
– integrated LEDs	–	x
Softkeys (number)	4	8
Logs	x	x
Diagnostic Function (STATUS/FORCE VAR)	x	x
Configurable C7-OP Languages		
German, English, French, Italian, Spanish, Russian	x	x
	–	C
Changing Languages in Online Mode	3	3

C

C.2 Brief Description of Standard Screens

Introduction

The table below presents an overview of all the standards screens for the units C7-623 and C7-624. As well as a brief comment on each function, mention is made of the required password level. The “Level 1” column lists the screens that you can choose from the basic screen. These screens allow you to make different calls, which are listed under “Level 2.”

The hierarchy detailed here relates to the supplied example configuration (see section 6.3)

Table C-2 Overview of Standard Screens

Level 1	Level 2	Function	Password Level
Alarm Msg	View	• Display alarm messages in alarm message buffer • Display message text for a message selected in the alarm message buffer	0
Alarm Msg	Print	Print alarm messages chronologically, that is all message events (arriving, departing, acknowledged) in the order of their occurrence.	2
Alarm Msg	Number	Display messages in the buffer, differentiating between ’All Messages’ and ’Pending Messages’	0
Alarm Msg	Delete	Delete all acknowledged and departed messages in the alarm message buffer	6
Alarm Msg	Overflow	Switch on/off system messages at overflow of alarm message buffer	4
Alarm Msg	Texts	Display all alrm messages texts	0
Event Msg	View	• Display event messages in event message buffer • Display message text for a message selected in the event message buffer	0
Event Msg	Print	Print event messages chronologically, that is all events (arriving, departing, acknowledged) in the order of their occurrence	2
Event Msg	Number	Display messages in the buffer, differentiating between ’All Messages’ and Pending Messages’	0
Event Msg	Delete	Delete all acknowledged and departed messages in the event message buffer	6
Event Msg	Overflow	Switch on/off system messages at overflow of event message buffer	4
Event Msg	Texts	Display all event messages	0
System	Operattion	Set C7-OP operation modes: on-line, off-line, transfer	8
System	Msg Display	Define whether the oldest (first) or most recent (last) message is to be displayed when several alarm messages are pending	4
System	System Msg	Display system message buffer	0
System	Language	Select language	2
System	Date/time	Set date and time	4
System	Printer	Set printer parameters	2

Table C-2 Overview of Standard Screens (Continued)

Level 1	Level 2	Function Level	Password
System	IF1 IF2	Set interface parameters (V.24) Set interface parameters (MPI)	6
STATUS VAR		Display S7-operands	0
MODIFY VAR		Display and modify S7-operands	8
Password	Login	User login via password input	0
Password	Logout	User logout and branch back to the message level	0
Password	Edit	• Display password list • Assign and change passwords and password levels • Delete passwords	9

C

C.3 Control Jobs and Their Parameters

Overview

Control jobs can be used to trigger functions from the user program in the C7 OP. Examples of these functions are as follows:

- Display screen
- Set date and time
- Modify general settings

A control job consists of four data words. The first data word contains the job number. Depending on the function, up to three parameters are transferred in data words 2 to 4. Figure C–1 shows the general structure of a control job.

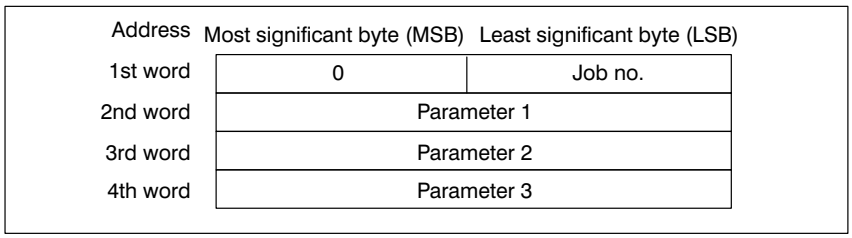


Figure C-1 Structure of a Control Job

Note

Please refer to Section 9.6 for information on any provisions to be made in the interface area.

Listing

Table C-2 lists all the control jobs and their parameters which are possible with the C7 (No. = Job number of the control job):

Table C-1 Control Jobs with Parameters

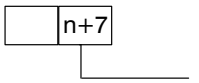
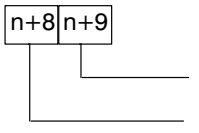
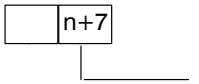
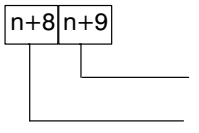
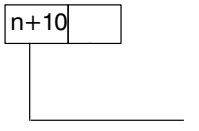
No.	Function
1	**Select C7 operating mode** Parameter 1 — 1: Off-line Parameters 2, 3 - In general, jobs can only be initiated when the C7 is already in on-line mode. Note: That means that the C7 can not be switched on-line by a control job.
3	**Hardcopy** Parameter 1, 2, 3 -
5	**Select contents directory** Parameter 1 — 1: Directory: Display screens 2: Directory: recipes, display 4: Directory: print screen 5: Directory: print recipes 7: Directory: recipes, data record transfer Parameters 2, 3 -
7	**Print out all process screens** Parameters 1, 2, 3 -
10	**Print recipe with all data records** Parameters 2 Recipe number (1..99) Parameters 2, 3

C

Table C-2 Control Jobs with Parameters (Continued)

No.	Function
11	**Select special screens** The following screens integrated in the firmware can be selected via their (fixed) object number. Parameter 1 n+6: Cursor lock (0=off/1=on) n+7: Special screen number **Alarm message buffer** 1 Buffer output 2 Output message number 3 Overflow warning on/off 4 Delete buffer yes/no **Event message buffer** 1 Buffer output 2 Output message number 3 Overflow warning on/off 4 Delete buffer yes/no **PG functions** 25 Status VAR 26 Control VAR **Special functions** 30 Language, brightness (contrast) 31 Change operating mode **Settings** 35 Adjust date/time 36 V.24 interface 38 Printer parameters 40 Message type **Message text** 45 Display alarm message 46 Display event message **System messages** 50 Output system message buffer **Password** 55 Login 56 Password input **C7 system functions** 97 C7 system function menu 98 C7-CPU mode selection 99 DI/DO status display Parameters 2, 3 -
12	**Message logging on/off** Parameter 1 0: Off 1: On Parameters s2, 3 -

Table C-2 Control Jobs with Parameters (Continued)

No.	Function
13	**Switching language** Parameter 1 0: 1st language 1: 2nd language 2: 3rd language Parameters 2, 3 - -
14	**Set time** **(BCD-coded)** Parameter 1  Parameter 2  Parameter 3 - -
15	**Set date** **(BCD-coded)** Parameter 1  Parameter 2  Parameter 3 

C

Table C-2 Control Jobs with Parameters (Continued)

No.	Function
16	**Parameters for V.24 interface** Parameter 1 Value for parameter 2 **Baud rate**[3] 0: 300 baud 1: 600 baud 2: 1200 baud 3: 2400 baud 4: 4800 baud 5: 9600 baud **Data bits** 0: 7 data bits 1: 8 data bits **Stop bits** 0: 1 stop bit 1: 2 stop bits **Parity** 0: even 1: odd 2: none **TD/OP address:** 1...30 [2] Parameter 2 Interface parameters to be set 0: Baud rate 1: Data bits 2: Stop bits 3: Parity Parameter 3 -
19	**Printer parameters** Parameter 1 Value for parameter 2 **Number of characters per line** 0: 20 characters/line 1: 40 characters/line 2: 80 characters/line **Number of lines per page** 0: 60 lines/page 1: 61 lines/page ... 12: 72 lines/page Parameter 2 Printer parameters to be set 0: Number of characters per line 1: Number of lines per page
21	**Display mode for alarm messages** Parameter 1 0: First value (oldest message) 1: Last value (newest message)
22	**Set display contrast** Parameter 1 0..15 Parameters 2, 3
23	**Set password level** Parameter 1 1..9 (1 = lowest password level, 9 = highest password level) Parameters 2, 3 -
24	**Password logout** (branching in message level) Parameters 1, 2, 3 -

Table C-2 Control Jobs with Parameters (Continued)

No.	Function
25	**Print shift log** Parameters 1, 2, 3 -
29	**Print nproduction report** Parameters 1, 2, –
31	**Print alarm message buffer** Parameter 1 0: Print chronologically 1: Print together Parameters 2, 3 -
32	**Print event message buffer** Parameter 1 0: Print chronologically 1: Print together Parameters 2, 3 -
37	**Overflow warning for event messages on/off** Parameter 1 0: Off 1: On Parameters 2, 3 -
38	**Overflow warning for alarm messages on/off** Parameter 1 0: Off 1: On Parameters 2, 3 -
41	**Transfer date/time to C7-CPU** If this job is submitted too often, the C7 can get overloaded, as two transfers per job are necessary.
42	**Fetch LED area from C7-CPU** Parameter 1 Block numbers Parameters 2, 3 -
43	**Fetch C7-CPU event message bit area** Parameter 1 Block numbers Parameters 2, 3 -
44	**Fetch C7-CPU alarm message bit area** Parameter 1 Block numbers Parameters 2, 3 -
45	**Fetch C7-CPU acknowledgement area** Parameter 1 Block numbers Parameters 2, 3 -

C

Table C-2 Control Jobs with Parameters (Continued)

No.	Function
47	**Fetch LED area from C7-CPU** Parameter 1: LED-image Area number (1...4) Parameter 2: LED image word 0 Parameter 3: LED image word 1 LED image → LED image word 0 / LED image word 1 Area pointer
	Note: The difference between this job and job 42 (fetch LED area from C7-CPU) is as follows: With job 47, the LED image is also transferred resulting in a faster control of the LEDs. The declared LED area may not be configured larger than 2DW!
48	**Menu selection** Parameter 1: Menu no in the standard menu 1 Message level 2 Main menu 3 Alarm messages 12 Print alarm messages 4 Event messages 14 Print event messages 5 Screens 6 Recipes 7 Statistical functions 18 Alarm statistics 19 Event statistics 8 PU functions 9 Special functions 24 System messages 23 Message texts 22 Settings 10 Password Parameter 2: Menu item number 0: first menu item 1...20 Other menu items Parameter 3: -
49	**Delete event message buffer**
50	**Delete alarm message buffer**
51	**Screen selection** Parameter 1: n+6: Cursor lock (0=off/1=on) n+7: Screen number (1...99) Parameter 2: Entry number[1)] (0...99) Parameter 3: Field number[2)] (0...8)

Table C-2 Control Jobs with Parameters (Continued)

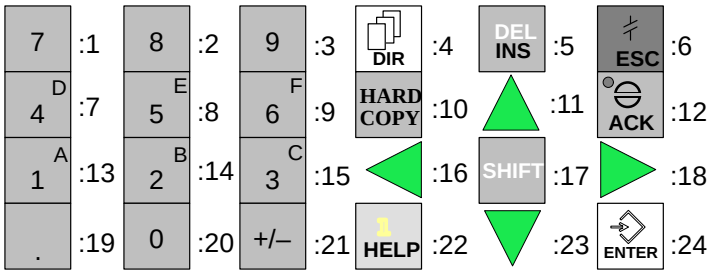
No.	Function
52	**Print screen** Parameter 1 Screen number (1..99) format: Byte Parameters 2,3 -
53	**Select recipe** Parameter 1 LB: Cursor inhibit (0: off; 1: on) RB: Recipe number (1..99) Parameters 2 Data record number (1..99) Parameters 3 LB:Entry number (0..99) (0:Cursor positioned to first entry) RB: Field number (0/1) The input fields of an entry are numbered consecutively: 0 Entry number field 1 First input field : nLast input field The numbering of the input fields starts at 1 again for each entry. Output fields are ignored by the serial numbering system!
54	**Print recipe** Parameter 1 Recipe number (1..99) Parameters 2 Data record number (1..99) Parameters 3 -
69	**Transfer recipe data record to C7–OP** Parameter 1 Recipe number: 1..99 Parameters 2 Data record number 1..99 Parameters 3 0,1 0: Data record is not overwritten. 1: Data record is overwritten.
70	**Transfer recipe data record from C7–OP to C7–CPU** Parameter 1 Recipe number: 1..99 Parameters 2 Data record number 1..99 Parameters 3 –.
71	**Partial screen updating** Parameter 1 0: Off 1: On Parameters 2,3 - This job may not be initiated if no screen is selected!
72	**Cursor positioning** in current process picture Parameter 1 Entry number (0..99) Parameter 2 Field number (0..8) Parameter 3 Cursor lock (0=off/1=on)

C

Table C-2 Control Jobs with Parameters (Continued)

<table>
<tr><th>No.</th><th>Function</th></tr>
<tr><td>73</td><td>Cursor positioning in current special screen
Parameter 1 Field number
Parameter 2 Cursor lock (0=off/1=on)
Parameter 3 -</td></tr>
<tr><td>74</td><td>Keyboard emulation
Parameter 1 n+6: Internal function number 1
System keyboard number 2

n+7: Password level
0: Password level will be evaluated
1: Password level will not be evaluated
Parameter 2 n+8: First key code
n+9: Second key code

Function keys
F1. 1 ... F16:16
System keyboard C7
 

Parameter 3
For keyboard emulation via a control job, please pay attention to the transfer time from the C7-CPU to the C7-OP.
For example – The acknowledgement of an alarm message by keyboard emulation on the C7-CPU can lead to an undesired result: –
– if the alarm message in question has already been acknowledged by the C7 or
– if a new alarm message or system message arrives before the job has been evaluated.</td></tr>
</table>

Control Job with Cursor Lock

If the cursor lock parameter is not equal to zero in one of the jobs 11, 51, 53, 72 and 73, the selected input field cannot be left using the cursor keys or the BREAK key. The cursor lock will first be released;

- upon repeating the job with cursor lock = 0
- or upon execution of another job which causes a change of the display.

If you try and leave the input field when the cursor is locked, you will receive the system message ”`$400 Invalid input`”.

C.4 System Messages

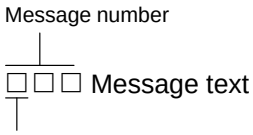
Introduction

This chapter provides on overview of the most important system message with their causes and the action required to remedy the errors.

Message Number

System messages on the OP can be placed in different categories.

Information on the category to which a system message belongs is contained in the message number:



0 Driver error
1 Startup message
2 Warning
3 Note
4 Operating error
5 Other message
6 Configuration error
7 Internal error

Message Category

The message category gives you a rough idea of the cause of a system message.

A few important system messages are shown below together with their causes and the action that has to be taken. Self explanatory system messages are not included.

Note

Messages are displayed in English until configuration data have been downloaded to the C7.

Procedure for ”Internal Errors”

Proceed as follows for all system messages that relate to ”internal errors”:

1. Switch off the C7 then restart it.
2. Put the C7 in Download mode during start-up (refer to Section 2.1), download the configuration again and restart the C7.
3. If the error continues to occur, please contact the nearest Siemens branch office. Report the number of the error that has occurred and any variable that may be included in the message.

C

Messages

The following tables list the most important messages, their causes and possible action to take.

Message	Cause	Action
Please wait	Mode in process of being changed	
Ready for transfer	Waiting for data from PG/PC	
Data transfer	Data being transferred between PG/PC and C7	
Firmware not compatible	Firmware cannot be used for current configuration	
EPROM memory failure	Memory submodule defective, internal hardware error	Return device for repair with details of error
RAM memory failure		
Flash memory failure	Memory submodule defective or transfer failure	Repeat download configuration or return device for repair

Message	Cause	Action
$ 005	Internal error	
$ 006	Error during data transfer in Download mode (message with two variables) VAR. 1 Status display VAR .2 1 **Internal error** 3 Time-out error 5 Parity error 6 Framing error 7 Overrun error 8 Line interruption 9 Receive buffer overflow 10 Wrong control character 11 **Internal error**	Check connection, repeat download
$ 040	No response from controller – Cable defective or not plugged in	 – Check physical connection
$ 041	Temporary driver error	– Restart PC – Download configuration
$ 100	Invalid RAM contents	
$ 104	Download mode canceled by pressing a key	
$ 106	Serious error eliminated and key pressed	
$ 108	Mode change	
$ 110	Mode change	
$ 114	Controller cold restart	
$ 115	Establishment of logical link	
$ 117	Connection to Controller OK again, following a fault	
$ 119	Automatic start of C7 (password list is not deleted)	
$ 125	Language changed by standard screen or control job	
$ 131	Mode change	
$ 133	Mode change	
$ 135	Mode change	
$ 136	No response from controller. Check program execution in controller or physical connection	
$ 138	Data block no. x not available in controller memory. Create % memory space.	

C

Message	Cause	Action
$ 200	Backup battery voltage has fallen below minimum value or: wrong battery type inserted	Replace battery
$ 201	Error during clock chip write (hardware error)	Return device for repair
$ 202	Error when reading date	Re-enter date (return C7 for repair)
$ 203	Error when reading time	Re-enter time (return C7 for repair)
$ 204	Error when reading day	Re-enter day (return C7 for repair)
$ 205	Printer not ready and internal storage of print jobs is no longer possible (capactity exceeded)	Ready printer or disable message log
$ 206	Printer not ready, print job will be stored temporarily	Ready printer
$ 207	Print job was aborted	Check printer, cable and connector
$ 210	Internal error	Refer to Action for internal errors
$ 212	Internal error	Refer to Action for internal errors
$ 213	Off-line mode not possible at present	Try mode change again later
$ 214	The job number configured by the C7-CPU or in a screen, is too large	Check user program and configured procress screen
$ 217 $ 218	The addresses of two variables overlap	Modify configuration (variable)
$ 220 $ 221	Printer buffer overflow, messages have been lost	
$ 222	Event buffer full to remaining buffer space	Delete buffer or configure smaller remaining buffer space
$ 224	Event buffer full; buffer partially deleted and forced printout initiated	
$ 225	Alarm buffer full to remaining buffer space	Delete buffer or configure smaller remaining buffer space
$ 227	Alarm buffer full; buffer partially deleted and forced printout initiated	
$ 229	Keyboard connector faulty or loose (hardware fault)	Return device for repair

Message	Cause	Action
$ 250	Changeover to required OP mode not possible via job.	This message may occur, for example, when trying to change over to loop–through mode during communications via FAP.
$ 252	Recipe functions of the TD/OP cannot be carried out simultaneously. Two examples for situations where message 252 may be issued: – The selected function is already active (e.g. in the background, used by the controller) – The operator tries to transfer or delete a data record after selecting it.	
$ 256	Insufficient memory space for executing the selected function. Select the function again. If the message is still output, ”simplify” the screen layout, i.e. configure fewer screen elements/functions. Examples: – Shift the function causing the message to another screen – Do not use any curves in connection with this function	
$ 257	A data record was not stored in connection with the currently loaded recipe version. If the data records concerned are still to be used, the old version must be entered in the recipe configuration. The assignment of the values of a data record is defined in the recipe structure. If the structure was modified in a new configuration, an ”old” data set may be interpreted incorrectly.	
$ 259	A data record must be transferred to the controller within a specified period of time. This time was exceeded. Possible causes: – Reception of the data record was not acknowledged by the controller (user program) – The data record is very large. The data record is transferred completely despite the timeout.	
$ 260	The controller mode (e.g. manual mode, automatic mode, STOP mode) does not correspond to the configuration.	
$ 261	A data set can no longer be used since the data is not consistent.	

Message	Cause	Action
$ 303	controller did not invert life flag. Data have not been requested or are no longer vaild	Check controller status
$ 304	Invalid job number or job parameters	Modify job on controller
$ 305	Data block number *x* missing	Create the data block that is missing
$ 306	Temporary driver error	
$ 307	Counter *x* not present on controller	Modify configuration (variable)
$ 308	Timer *x* not present on controller	Modify configuration (variable)
$ 309	Input *x* not present on controller	Modify configuration (variable)
$ 310	Output *x* not present on controller	Modify configuration (variable)
$ 311	Flag *x* not present on controller	Modify configuration (variable)
$ 312	Print job rejected because similar job now being executed	Wait until previous job terminated; initiate again
$ 313	Print job will be processed later because printer now busy	
$ 315	Information text has not been configured for highlighted object (that is message, setpoint)	
$ 316 $ 317	Current password level too low for required operator input	Log in at higher password level
$ 318	Login attempted with invalid password	
$ 319	You entered and tried to edit an existing password	
$ 320 $ 321		Enter password first, and then specify level
$ 322		Password must contain at least 3 characters
$ 323	In a buffer mask, you pressed → (message text), though an entry does not exist for the current message.	
$ 324	The screen number or entry number in your input does not exist.	
$ 335	Confirmation of alarm message inhibit	
$ 336 $ 337 $ 338	Printer cannot be addressed	Check printer and connection to C7
$ 339	Communication with controller resumed	
$ 340	You cannot operate the C7 with status function running on programming device	
$ 341	Internal error	

$ 342	This message indicates a data block error. Variables **x** and **y** are used to identify the cause of the error (variable **x**) and the number of the receive block (variable **y**). Variable x: 0 Wrong block length entered in receive block no. **y** 1 Wrong number entered in receive block no. **y** Correct the required block length or block number or send the correct data block.	
$ 385 $ 386	Recipe transfer in progress. During this time, the operator panel is not ready for operator inputs. Possible cause: The PLC has not reset the corresponding control/checkback bit which cancels the recipe inhibit state in DB–TDOP.	

C

Message	Cause	Action
$ 400	Invalid key pressed	
$ 401	Entred value does not match display format	
$ 402	Operating error in STATUS VAR or MODIFY VAR screen; (after pressing INS when 10th variable line already assigned).	
$ 403	Incorrect time input	
$ 404	Incorrect time input	
$ 406	Operating error in STATUS VAR or FORCE VAR screen	Abort updating (ESCAPE key)
$ 407		
$ 409	Lower limit for input ignored	Enter a value greater than or equal to *Var*
$ 410	Upper limit for input ignored	Enter a value smaller than or equal to *Var*
$ 411	Selection of special screens is not permissible in this case. If necessary, modify the configured interface parameters.	
$ 442	Thie message indicates a data block error. The **x** and **y** variables identify the cause of the error (x variable) and the number of the affected receive block (y variable). **x variable**: 0 Invalid block size entered in receive block no. **y**. 1 Invalid block number entered in receive block no. **y**.	Either correct the block size or the block number or send the correct data block.

Message	Cause	Action
\$ 500 \$ 501 \$ 502 \$ 503 \$ 504	Download to C7-CPU not possible at pressend – C7-CPU overloaded – Standard FB not called for more than 1.5 s	– Check user program
\$ 506	Too many message blocks having the same block number in transit (overloading)	Error occurs when C7-CPU sends too many jobs within a certain time with “Fetch message bit area”
\$ 509	Firmware version different from standard FB version	Load new standard FB onto C7-CPU
\$ 510	This message is output if: – The data block for the variable does not exist in the recipe, or – The recipe data contains errors.	Either set up the data block or change the configuration.
\$ 511	The data record number in the PLC job or the function key is invalid.	
\$ 512	This message indicates a PLC error. The variable which is transferred with the message identifies the number of a data block that is too short.	Correct the configuration if necessary.
\$ 520	Too many returns stored	Go to message level (if necessary, by pressing ESCAPE key)
\$ 522	Screen cannot be selected due to inadequate storage space. Results in cold restart with memory optimization	1. Delete unused fields from configuration 2. Configure smaller screen (with fewer fields) or partition screen
\$ 526	Loop-through mode is set on C7	Change to Normal mode
\$ 536	Disturbance on link between OP and diskette drive.	Check the physical connection.
\$ 538	Job and operator access data record simultaneously.	Repeat your input if it was not executed.
\$ 539	The data records stored in RAM for recipe no. x contain errors; they have been deleted. Any data records stored in flash memory are still valid.	
\$ 540	The maximum permissible number of data records in memory has already been reached.	
\$ 541 to 550	The specified variable cannot be found in the controller.	Check the configuration.
§ 551	Controller address cannot be found.	

Message	Cause	Action
$ 600	Wrong parameter transferred by ProTool/Lite (overflow warning)	Set required value by means of standard screen or controller
$ 601	Wrong parameter transferred by ProTool/Lite (message log)	Set required value by means of standard screen or controller
$ 602	Wrong parameter transferred by ProTool/Lite (remaining buffer size)	Re-configure and repeat download of required value
$ 603	Recipe setpoint is only set up symbolically.	
$ 604	Message not configured for a set message bit	Configure messages and repeat download
$ 605	Process connection only configured symbolically.	
$ 606	Too many message variables are configured.	
$ 607	Configured data type does not exist.	
$ 608	Process connection only configured symbolically.	
$ 609	Screen number does not exist.	
$ 610	Control operator object for header or footer does not exist or is not allowed.	
$ 611	Special operator object for buffer printout does not exist.	
$ 613	Data block does not exist or is too short	Create DB or required length in controller
$ 614	The layout of the listing had not been configured when the print job was issued.	
$ 615	Line to be output is larger than the reserved print memory or the number of control sequences is too large	Check configuration for log
$ 616 $ 617		See Internal Errors
$ 618	Wrong value entered: Bit no. is actual control value.	
$ 619	ProTool/Lite error (data structure for presetting of setpoint)	Re-load ProTool/Lite, repeat download of configuration
$ 620	Wrong parameter transferred by ProTool/Lite (function keyboard)	Repeat download of configuration
$ 621	Wrong parameter transferred by ProTool/Lite (message type)	Set required value by means of standard screen or controller
$ 622	Configured recipe does not fit in the C7–CPU recipe mailbox (more than 256 data words)	
$ 623		See Internal Errors
$ 624	No recipe entries available.	
$ 625	Invalid recipe number in control job or function area.	

$ 626	No setpoints configured.	
$ 627		See Internal Errors
$ 628	The recipe does not fit into the boxes.	
$ 629	LED image area too small	Enlarge LED image area according to configured bit offsets
$ 630	Keyboard image area too small	Enlarge image area according to configured bit offsets
$ 631	Message configuration **x** is incomplete or incorrect. **x variable**: 1, 2 The triggered alarm is not configured 3 The process connection is only set up symboli–cally 4 The actual value field is only set up symbolically 5, 6 The triggered event message is not configured 7 The symbolic actual value field is only set up symbolically 21...24 Field texts do not exist for the symbolic actual value 25 Field type not permitted 8...20 Internal error	Add to configuration and repeat download
$ 632	(Message with one variable) 1, 4 Information text not available 2 Information text identifier not available for messages 12 Process screen does not contain entries 3, 6, 7, 8, 11, 13 Internal errors	Add to configuration and repeat download

C

$ 634	(Message with one variable) 18 Screen title not configured 0 to 8, 34 Internal errors	Add to configuration and repeat download
$ 635	Configuration error in **x** **.x variable** : 1 The screen or recipe entry is only set up symbolically 3 The field is only set up symbolically 6 The message, entry or information text is not configured for the current language 7...9, 19, 28, 41...43 Internal errors 18 The screen or recipe title is not configured 20 The process connection is only set up symbolically 21 The information text is only set up symbolically 22 The symbolic field is only set up symbolically 23 Fewer than 2 field texts are configured for a symbolic field 24 The current field text is not configured for a symbolic field 25 Invalid data format for a symbolic field (only KF and KY are permitted) 26 Recipe setpoint configured with Char data format 33 Invalid data format for setpoint field 35 Data format for the scheduler is too short 36 Invalid data format for actual control value 44 For fixed cross-jump to menu: menu item does not exist 45 For fixed cross-jump to screen: entry or field number does not exist 46 Too many current control values on screen (maximum of 200 permitted) 48 Too many fields on process screen 50 Process connection for soft keys does not exist 51 Soft key number too high 53 Information text for soft key not configured or not in all languages 55 Soft key specified in entry does not exist	Add to or modify configuration and repeat download

Message	Cause	Action
$ 636 $ 637	Initiated event messages (No. x) not configured	Add to configuration and repeat download
$ 638 $ 639	The actual value field for event message no. x is available in symbolic form only.	
$ 640 $ 641	Initiated alarm messages (No. x) not configured	Add to configuration and repeat download
$ 642 $ 643	The actual value field for alarm message no. x is available in symbolic form only.	
$ 645 $ 649	Internal errors	
$ 650	Area pointer for function you used not configured	Configured area pointer
$ 651	Internal error	
$ 653	Configured user version number does not agree with that stored in C7-CPU	Adjust user version number
$ 655	controller acknowledgement area not physically beyond alarm message bit area (serious error, no startup)	Re-configure C7-CPU → C7 O/I acknowledgement areas and repeat download
$ 657	Configured controller driver not supported by version of device you are using (serious error)	Change protocol for version of device you are using and repeat download configuration
$ 659	Invalid variable in recipe no. x (BIN format).	
$ 660	Invalid destination configured for cross-jump in screen	Add to configuration and repeat download
$ 662	Invalid destination configured for cross-jump in screen	Add to configuration and repeat download

Message	Cause	Acton
$ 667	Incorrect configuration **x**. **Variable x:** 1 Data type does not correspond to DB 2 DB number greater than 15 3 DB length greater than 1024 4 DW is in data block header 5 Actual value is not in send block 6 Setpoint is not in receive block 7 Setpoint/actual value is not in receive block 8 Initial value is not in send block 9 Data type does not correspond to DB 10 DB number greater than 15 11 DB length greater than 1024 12 DW is not in data block header 13 Area located in wrong DB 14 Sum of data blocks too high	x = 1..8: Change configuration of process link and transfer again x = 9..13: Change configuration of area pointer and transfer again x = 14: Restrict configuration and transfer again
$ 670	Too many variables were requested simultaneously. Remedy: Configure – a longer basic clock, – fewer variables in the screen.	
$ 681	The link between OP and controller is disturbed. The interface parameters are possibly set incorrectly.	
$ 682	Configure fewer process links for the screen currently displayed.	

Message	Cause	Action
$ 702	**Internal error** (actual value error)	
$ 703	**Internal error** (job faulty)	Limit configuration
$ 704	Flash memory full	
$ 705	**Internal error** (S5 error)	
$ 706	**Internal error** (unknown message acknowledged)	
$ 7xx	**Internal errors**	

C.4.1 Internal Errors

The field numbers from 700 onwards and some errors defined in previous sections are used to describe internal errors of the C7 control systems or the ProTool configuring tools.

Procedure

Please proceed step by step as described below if an internal error occurs:

- Set the C7 CPU to *STOP* mode. Switch off the C7 and then restart it.
- Set the C7 OP to transfer mode during startup. Transfer the configuration again and restart the C7.
- Should the error occur again, contact the Siemens regional office in your vicinity. Specify the error number including any variables coming up in the message.

Possible Causes

- **005** Error no.: #Var1, ...(wie dt.
- **6xx** Error in configuration file
- **701** Internal actual value errors
- **702** Invalid job (illegal job number or job parameter)
- **703** Flash full (restrict your configuration)
- **704** Controller error
- **705** Acknowledgement for unknown message
- **706** Recipe request already active
- **7xx** Internal errors

SIMATIC C7 and S7 Reference

D

Supplementary Literature

/70/ S7–300 Programmable Controller
Hardware and Installation Manual

/71/ *S7–300/M7–300 Programmable Controllers,*
*Module Specification*s
Reference Manual

/72/ *S7–300 Programmable Controller, CPU 312/CPU 314*
Instruction List

/230/ *Standard Software for S7, Converting STEP 5 Programs*
Manual

/232/ *Statement List (STL) for S7–300 and S7–400,* Programming Manual

/233/ Ladder Logic (LAD) for S7–300 and S7–400, Programming Manual

/235/ *System Software for S7–300 and S7–400, System and Standard Functions*
Reference Manual

/280/ *System Software for M7–300 and M7–400, Program Design*
Programming Manual

Siemens Worldwide

E

In this Appendix

In this appendix you will find a list of:

- All cities in the Federal Republic of Germany with Siemens Sales Offices and
- All European and non-European Siemens Companies and Representatives

SIMATIC Contact in the Federal Republic of Germany

ZN Zweigniederlassung

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

SIMATIC Contact in Europe

Belgien

1060 Bruessel
Siemens S.A., VP4, Hr. Gmuer
Chaussee de Chaleroi 116
00 32 (2) 5 36 25 33
Fax 00 32 (2) 5 36 23 87

Bulgarien

1113 Sofia
Siemens AG, Fr. Kirova
Blvd. Dragan Zankov Nr. 36
0 03 59 (2) 70 85 21
Fax 0 03 59 (2) 68 50 51

Dänemark

2750 Ballerup
Siemens A/S, IP, Hr. Hansen
Borupvang 3
00 45 (44) 77 42 90
Fax 00 45 (44)77 40 16

Finnland

02601 Espoo
Siemens Osakeyhtioe,
OEM/AUT 1, Hr. Saarelainen
Majurinkatu, P.O.B. 60
0 03 58 (0) 51 05 36 70
Fax 0 03 58 (0) 51 05 36 56

Frankreich

69641 Caluire-et-Cuire/Lyon
Siemens S.A., AUT 1, Leitstelle
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, Leitstelle
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,
Leitstelle, 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,
Leitstelle, 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, Hr. 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, Leistelle
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, Hr. Huguet
ZAC de Basso Cambo
Avenue du Mirail, BP 1304
00 33/ 62 11 20 15
Fax 00 33/ 61 43 02 20

Griechenland

15110 Amaroussio/Athen
Siemens A.E., HB 3 AUT,
Hr. Antoniou; Paradissou &
Artemidos, P.O.B. 6 10 11
00 30 (1) 68 64-5 15
Fax 00 30 (1) 68 64-5 56

54110 Thessaloniki
Siemens A.E., VB 3 AUT,
Hr. Passalidis
Georgikis Scholis 89, P.O.B. 10290
00 30 (31) 47 92 12
Fax 00 30 (31) 47 92 65

Grossbritannien

Manchester M20 2UR
Siemens PLC, Control Systems,
Hr. Hardern
Sir William Siemens House,
Princess Road
00 44 (61) 4 46 52 33
Fax 00 44 (61) 4 46 52 32

Irland

Dublin 11
Siemens Ltd., Power & Automa-
tion Division, Hr. 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

Island

121 Reykjavik
Smith & Norland H/F,
Hr. Kjartansson,
Noatuni 4, P.O.B. 519
0 03 54 (1) 62 83 00
Fax 0 03 54 (1) 62 83 40

Italien

40127 Bologna
Siemens S.p.A., AUT R10A,
Hr. 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,
Hr. 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,
Hr. 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,
Hr. 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,
Hr. 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,
Hr. Montoli, Via Pio VII, 127
00 39 (11) 6 17 3-1
Fax 00 39 (11) 61 61 35

Kroatien

41000 Zagreb
Siemens d.o.o., Hr. Culjak
Trg Drazena Petrovica 3 ("Cibona")
0 03 85 (41) 33 88 95
Fax 0 03 85 (41) 32 66 95

Luxemburg

1017 Luxemburg-Hamm
Siemens S.A., AUT, Hr. 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

Niederlande

2595 AL Den Haag
Siemens Nederland N.V., IPS/APS,
Hr. Penris, Prinses Beatrixlaan 26
00 31 (70) 3 33 32 74
Fax 00 31 (70) 3 33 34 96

Norwegen

5033 Fyllingsdalen
Siemens A/S Bergen,
Hr. Troan, Bratsbergveien 5
Postboks 36 60
00 47 (55) 17 67 41
Fax 00 47 (55) 16 44 70

0518 Oslo 5
Siemens A/S, AUT Produkter,
Hr. 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,
Hr. Thorsen, Spelaugen 22
00 47 (73) 95 96 69
Fax 00 47 (73) 95 95 04

Österreich

6901 Bregenz
Siemens AG, AUT, Hr. Madlener
Josef-Huter-Straße 6,
Postfach 347
00 43 (55 74) 41 92 72
Fax 00 43 (55 74) 41 92 88

8054 Graz
Siemens AG, AUT, Hr. Jammernegg
Strassganger Straße 315
Postfach 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, Hr. Mayr
Siemensstraße 24, Postf. 9 04
00 43 (5 12) 23 12 60
Fax 00 43 (5 12) 23 15 30

9020 Klagenfurt
Siemens AG, AUT, Hr. 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, Hr. Schmidt
Wolfgang-Pauli-Straße 2
Postfach 563
00 43 (7 32) 3 33 02 95
Fax 00 43 (7 32) 3 33 04 93

5020 Salzburg
Siemens AG, AUT, Hr. Mariacher Jun.
Innsbrucker Bundesstraße 35
Postfach 3
00 43 (6 62) 4 48 83 35
Fax 00 43 (6 62) 4 48 83 09

1211 Wien
Siemens AG, AUT 1, Hr. Strasser,
Siemensstraße 88-92,
Postfach 83
00 43 (1) 25 01 37 88
Fax 00 43 (1) 25 01 39 40

Polen

40-931 Katowice
Siemens Sp. z.o.o., Niederlassung
Katowice, Hr. 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., Niederlassung
Poznan, Hr. 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., Hr. 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., Niederlassung
Wroclaw, Hr. 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, Hr. Eng. C. Pelicano
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, Hr. Eng. A. Amaral,
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

Rumänien

76640 Bucuresti
Siemens, Birou de consultatii
tehnice, Hr. Fritsch
Str. Zarii No. 12, sector 5
00 40 (1) 2 23 47 95
Fax 00 40 (1) 2 23 45 69

Russland

113043 Moskau
Siemens AG, Hr. Engelhard/
Hr. Michailow, Ul. Dubininskaja 98
0 07 (0 95) 2 36 75 00
Fax 0 07 (0 95) 2 36 62 00

Schweden

40020 Göteborg
Siemens AB, ASP, Hr. Ohlsson
Ostergardsgatan 2-4
Box 1 41 53
00 46 (31) 7 76 86 53
Fax 00 46 (31) 7 76 86 76

55111 Jönköping
Siemens AB, ASP, Hr. Jonsson
Klubbhusgatan 15, Box 10 07
00 46 (36) 15 29 00
Fax 00 46 (36) 16 51 91

20123 Malmö
Siemens AB, ASP, Hr. Jämtgren
Grimsbygatan 24, Box 326
00 46 (40) 17 46 14
Fax 00 46 (40) 17 46 17

85122 Sundsvall
Siemens AB, ASP, Hr. Sjöberg
Lagergatan 14, Box 766
00 46 (60) 18 56 00
Fax 00 46 (60) 61 93 44

19487 Upplands Väsby/Stockholm
Siemens AB, ASP-A1, Hr. Persson
Johanneslandsvägen 12-14
00 46 (8) 7 28 14 64
Fax 00 46 (8) 7 28 18 00

Schweiz

1020 Renens/Lausanne
Siemens-Albis SA, Systemes
d'automation, VHRL, Fr. Thevenaz
5, Av. des Baumettes
Case postale 1 53
00 41 (21) 6 31 83 09
Fax 00 41 (21) 6 31 84 48

8047 Zürich
Siemens-Albis AG, VHR 3,
Hr. Engel, Freilagerstraße 28-40
00 41 (1) 4 95 58 82
Fax 00 41 (1) 4 95 31 85

Slowakische Republik

81261 Bratislava
Siemens AG, Hr. Sykorcin,
Tovarenska 11
00 42 (7) 31 21 74
Fax 00 42 (7) 31 63 32

Slowenien

61000 Ljubljana
Siemens Slovenija, Hr. Lavric
Dunajska C47
0 03 86 (61) 1 32 60 68
Fax 0 03 86 (61) 1 32 42 81

Spanien

48011 Bilbao
Siemens S.A., AUT 1, Hr. 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, Hr. 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, Hr. Huchet
Corrida, 1
00 34 (85) 35 08 00
Fax 00 34 (85) 34 93 10

15005 La Coruna
Siemens S.A., AUT 1, Hr. 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, Hr. 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,
Hr. 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,
Hr. 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, Hr. 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, Hr. Garrido
Pizarro, 29
00 34 (86) 41 60 33
Fax 00 34 (86) 41 84 64

50012 Zaragoza
Siemens S.A., AUT 1, Hr. Aliaga
Avda. Alcalde Gomez Laguna, 9
00 34 (76) 35 61 50
Fax 00 34 (76) 56 68 86

Tschechien

60200 Brno
Siemens AG, Kancelar Brno,
Hr. Tucek, Vinarská 6
00 42 (5) 43 21 17 49
Fax 00 42 (5) 43 21 19 86

14000 Praha 4
Siemens AG, Zastoupeni v CR,
Hr. Skop, Na strzi 40
00 42 (2) 61 21 50 33 6
Fax 00 42 (2) 61 21 51 46

80040 Findliki-Istanbul
SIMKO A.S., AUT ASI 1, Fr. Yargic
Meclisi Mebusan Cad. 125
00 90 (1) 25 10 90 01 706
Fax 00 90 (1) 25 10 90 07 09

Türkei

06680 Ankara-Kavaklidere
SIMKO-ANKARA, Hr. Ensert,
Atatürk Bulvari No. 169/6
00 90 (312) 4 18 22 05

80040 Findikli-Istanbul
SIMKO TIC. ve SAN. A. S.,
AUT 1, Fr. 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-Vertretung, AUT,
Hr. Liebschner,
Ul. Worowskowo 27
0 07 (044) 2 16 02 22
Fax 0 07 (044) 2 16 94 92

Ungarn

1036 Budapest
Siemens GmbH, AUT 1, Hr. Turi
Lajos utca 103
00 36 (1) 2 69 74 55
Fax 00 36 (1) 2 69 74 54

SIMATIC Contact Outside Europe

Afrika

Ägypten

Zamalik/EGY-Cairo
ELETECH, AUT, Hr. W. Y. Graiss
6 Zarkaria Rizk Street,
P.O.B. 90
✆ +20 (2) 3 42 03 71
Fax +20 (2) 3 42 03 76

Algerien

16035 Hydra/Alger
Siemens, Bureau d'Alger, Division Energie, Hr. Bennour,
44, rue Abri Areski , B.P. 112
✆ +213 (2) 60 40 88
Fax +213 (2) 60 65 98

Elfenbeinküste

Abidjan 15/R. C. I.
Siemens AG, SEMEN, Mr. Hellal,
16 B.P. 1062
✆ +2 25 (37) 46 57
Fax +2 25 (27) 10 21

Libyen

Tripoli/Libya S.P.L.A.J.
Siemens AG, Branch Libya,
Hr. Wahab, Zat-EL-Imad-
Building Tower No. 5, Floor No. 9
P.O.B. 91 531
✆ +218 (21) 4 15 34
Fax +218 (21) 4 79 40

Marokko

Casablanca 05
SETEL S.A., AUT, Hr. 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., Hr. 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

Südafrika

RSA-2001 Braamfontein
Siemens Ltd., AUT, Hr. 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

Tunesien

TN-2062 Romana-Le Bardo
FAZE Sarl Electrotechnique,
Hr. Fantar, Immeuble Cham
✆ +2 16 (1) 51 90 91
Fax +2 16 (1) 50 19 32

TN-2035 Charguia II Tunis
SITELEC S.A. Hr. 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,
Hr. 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

Amerika

Argentinien

8000 Bahia Blanca,
Prov. de Buenos Aires
Siemens S.A., Hr. 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,
Hr. Rudriguez Juis/Hr. 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., Hr. S. Garcia,
Campillo 70
✆ +54 (51) 73-9940/994
Fax +54 (51) 72-97 14

5539 Las Heras, Prov. de Mendoza
Siemens S.A., Hr. 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., Hr. R. Stiza,
Ricchieri 750
✆ +54 (1) 41 37-03 21/0
Fax +54 (1) 41 37-07 87

Bolivien

La Paz
Sociedad Comercial e Industrial
Hansa Ltda., E & A, Hr. Beckmann
Calle Mercado esq. Yanacocha
C. P. 10 800
✆ +591 (2) 35 44 45
Fax +591 (2) 37 03 97

Brasilien

05110-900 Sao Paulo, SP, Pinituba
MAXITEC S.A., AUT-PA, Hr. F. Rocco,
Avenida Mutinga, 3650
✆ +55 (11) 8 36 29 99
Fax +55 (11) 8 36 29 50

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,
Hr. 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, Hr. 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, Hr. Godoy
2a Calle 6-76
Zona 10, Apartado 1959
✆ +5 02 (2) 32 44 44
Fax +5 02 (2) 34 36 70

Kanada

Mississauga, ON L5N 7AG
Siemens Electric Ltd., Dept. SL 20,
Hr. 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., Hr. 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

Kolumbien

Baranquilla
Siemens S.A., EA, Hr. 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,
Hr. 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,
Hr. 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 Hr. C. A. Naranjo
Carrera 48 A, 15 Sur 92
✆ +57 (94) 2 66-30 66
Fax +57 (94) 2 68-25 57

Mexiko

02300 Mexico, D.F.
Siemens S.A. de C.V., EI-AUT,
Hr. 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,
Hr. 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
Sautheast, Hr. 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,
Hr. 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,
Hr. 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,
Hr. 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,
Hr. 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,
Hr. 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,
Hr. 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

Asien

China

510064 Guangzhou
Siemens Ltd. China, Guangzhou
Office, Hr. 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,
Hr. 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, Hr. 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, Hr. 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

E

SIMATIC Contact Outside Europe

Hongkong

Hong Kong
Siemens Ltd. Hang Kong A. R. O.,
Automation System ,
Division , Hr. Keiren Lake,
7th Floor, Regency Centre,
39 Wong Chuk Hang Road
✆ +85 (2) 28 70 76 11
Fax +85 (2) 25 18 04 11

Indien

Bangalore 560 001
Siemens Ltd., BAN/AUT-MAP,
Hr. B. Sunderram
Jyoti Mahal, 3rd Floor
49, St. Marks Road, P.O.B. 5212
✆ +91 (80) 2 21 21 01
Fax +91 (80) 2 21 24 18

Bombay 400 018
Siemens Ltd., AUT/M-AP,
Hr. 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,
Hr. D. K. Ganguli
6, Little Russel Street, P.O.B. 715
✆ +91 (33) 2 47 83 74/-80
Fax +91 (33) 2 47 47 83

New Delhi 110 002
Siemens Ltd., DEL/AUT-MAP,
Hr. R. Narayanan
4A, Ring Road, I.P. Estate,
P.O.B. 7036
✆ +91 (11) 3 31 81 44
Fax +91 (11) 3 31 41 78

Indonesien

Jakarta 12870
Dian Graha Elektrika, Jakarta, Power
Eng. & Autom. Div., Hr. 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., Hr. Din-Payuh
Khiabane Ayatollah Taleghani 32
Siemenshouse, P.O.B. 15875-4773, 15
Teheran
✆ +98 (21) 61 41
Fax +98 (21) 6 40 23 89

Japan

Tokyo 141-00
Siemens K.K., ATT, Hr. Nakamichi
Siemens Fujikara Building, 8F
11-20, Nishi-Gotanda 2-chome
Shinagawa-ku
✆ +81 (3) 34 90 44 37
Fax +81 (3) 34 95 97 92

Pakistan

Karachi - 74400
Siemens Pakistan Eng. Co. Ltd.,
Power Division, Hr. Ilyas
ILACO House
Abdullah Haroon Road
P.O. Box 7158
✆ +92 (21) 51 60 61
Fax +92 (21) 5 68 46 79

Philippinen

Metro Manila
Siemens Inc., Hr. 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 Arabien

Jeddah - 21412
Arabia Electric Ltd. Service Center,
Hr. Kobeissi, P.O.B. 4621
✆ +9 66 (2) 6 65 84 20
Fax +9 66 (2) 6 65 84 90

Singapur

Singapore 1334
Siemens (Pte) Ltd. Singapore, AUT,
Hr. Ulf Bexell,
2 Kallang Sector
✆ +65 8 41 35 28
Fax +65 8 41 35 29

Südkorea

Seoul
Siemens Ltd., E+A, Hr. 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, Hr. Gulden
6th Fl., Cathy Life Insurance Bldg.
296, Jen Ai Road, Sec. 4
✆ +8 86 (2) 3 25 48 88
Fax +8 86 (2) 7 05 49 75

Thailand

Bangkok 10110
Berli Jucker Co. Ltd., Hr. 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
Hr. Nguyen Huang Giang
18, Phan Boi Chau Street
✆ +84 (4) 25 60 61
Fax +84 (4) 26 62 27

Australien

Australien

Adelaide
Siemens Ltd. Adelaide Office, CS/I.A.,
Hr. 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., Hr. 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., Hr. 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, Hr. Stephen Coop,
383 Pacific Highway, Artamon
✆ +61 (2) 4 36 78 04
Fax +61 (2) 4 36 86 24

Neuseeland

Greenlane, Auckland 5
Siemens Ltd. Auckland Office,
CS/I.A., Hr. A. Richmond
300 Great South Road
P.O.B 17-122
✆ +64 (9) 5 20 30 33
Fax +64 (9) 5 20 15 56

Glossary

A

ACCU

The accumulators are registers in the → C7 CPU and are a clipboard for load, transfer and comparison, math and conversion operations.

Address

An address is the identifier for a specific operand or operand area. Examples: I 12.1; memory word MW 24; data block DB 3.

Analog I/O

The analog I/O converts analog process values (e.g. temperature) into digital values that can be processed by the C7 CPU or converts digital values into analog manipulated variables.

Area Pointer

The area pointer is necessary for exchanging data between the operator panel section and the controller section of the C7. It includes information on the location and size of data areas in the controller.

Authorization Input

External access point to superuser password level.

B

Back-to-Normal

Point of time at which the controller revokes a message.

Backup Buffer

The backup buffer backs up memory areas of the C7 without the need for a backup battery. The memory backs up a programmable number of timers, counters, memory marker bits and data bytes, the → retentive timers, counters, memory marker bits and data bytes.

Baud Rate

Data transmission rate (bit/s).

C

C7-620

The C7-620 control system integrates an S7-300 CPU, a COROS OP, I/O and an IM 360 interface module in one single unit.

C7 CPU

The C7 CPU is a central processing unit of the C7 range, complete with processor, arithmetic unit, memory, operating system 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 (multipoint interface).

C7 OP

The C7 OP handles the OP functions of the C7 control system. It is independent of the → C7 CPU and continues to operate, for example, if the C7 CPU enters the STOP mode. The C7 OP has its own MPI address and is connected to the C7 CPU via this interface. It is via the MPI that the C7 OP is connected to a configuring computer (programming device/PC).

C7 I/O

The C7 I/O (→ signal modules) forms the interface between the process and the programmable controller. There are digital input and outputs as well as analog inputs and outputs. The integral universal inputs have special functions in the C7 (interrupt/counter inputs).

Code Block

With SIMATIC S7, a code block is a block containing part of the STEP 7 user program. (In contrast to a → Data Block, which contains only data).

Coming (Message)

Point of time at which the C7 triggers a message output.

Communications Processor

Communication processors are modules for point-to-point and bus connections.

Complete Restart

When a central processing unit starts up (for example, after selecting a C7 CPU mode in the system functions menu or on power ON), organization block OB 100 (complete restart) is executed prior to the cyclic program processing (OB 1). On a complete restart, the process image input table is read in and the STEP 7 user program is executed starting with the first instruction in OB 1.

Compress

The PG online function ”Compress” is used to shift all valid blocks in the RAM of the C7 CPU evenly and without any gaps to the beginning of the user memory. All gaps caused by deleting or correcting of blocks are thus removed.

Configuration

Defining plant-specific defaults, messages and displays using the ProTool/Lite configuring software.

Configuration Memory

The configuration memory is a flash memory integrated in the C7 OP. It is used for storing the configuring data.

Control Job — It is used for triggering a function in the C7.

Counters — Counters are component parts of the → C7 CPU. The contents of the ”counter cells” can be changed by means of STEP 7 instructions (for example, upward/downward counting).

D

Data Block — Data blocks (DB) are data areas in the user program, which contain user data. Global data blocks can be accessed by all code blocks and instance data blocks are assigned to a specific FB call.

Data, Static — Static data are data which 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 stored in the L stack during execution of the block and that are not retained after the execution.

Default Setting — The default setting comprises generally applicable basic settings which are used whenever no other values are specified.

Diagnosis — → Diagnostic Functions, → System Diagnosis

Diagnostic Events — Modules with diagnostics capability signal any system fault to the → C7 CPU via diagnostic interrupts.

Diagnostic Functions — Diagnostic functions comprise the complete scope of system diagnostics and the detection, evaluation and signalling of faults within the C7.

Diagnostic Interrupt — Modules with diagnostics capability signal system errors to the → C7 CPU by means of diagnostic interrupts.

Display — Representation of logically related process data for display and modification in the C7 either in groups or individually.

Display Duration — Interval between the arrival of an event message and reaching of the back-to-normal state.

Display Entry Element of a display. It consists of the entry number, text and variables.

Display Function Function that causes a change of the display contents, for example, display of message level or display of fault message buffer.

Display Level Processing level of the C7 at which displays can be monitored and operated.

E

Event Message It draws the operator's attention to certain operating states in the machine or plant to which the C7 is connected.

F

Fault Message It points out particularly critical operating states. A fault message always requires an acknowledgement.

Fault Time Interval between the arrival of a fault message and reaching of the back-to-normal state.

FB → Function Block

FC → Function

Flash EPROM With regard to their characteristic to retain data in the case of a power failure, FEPROMs correspond to the electrically erasable EEPROMs, but can be erased much faster (FEPROM = Flash Erasable Programmable Read Only Memory).

The following data can be held in flash memory without being affected by a power failure:

- The → User Program.
- The → Parameters that determine the characteristics of the → C7 CPU and the I/O functionality of the C7.

Forced Printout Automatic printout of fault and event messages that can be deleted in the event of a buffer overflow.

Function

In accordance with IEC 1131-3, a function (FC) is a → Code Block without → Static Data. With a function, you can transfer parameters in the user program. Functions are therefore suitable for programming frequently recurring complex functions, for example, calculations.

Function Block

In accordance with IEC 1131-3, a function block (FB) is a → Code Block with → Static Data. With a function block, you can transfer parameters in the user program. Function blocks are therefore suitable for programming frequently recurring complex functions, for example, closed-loop controls, operating mode selection.

H

Hardcopy

Output of the display contents to a printer.

I

Information Text

Supplementary, user-configurable information on messages, displays, display entries and selection fields.

Inquiry Function

The inquiry functions of STEP 7 permit the display of status information on the programming device via one or more C7 systems during the various startup phases and during operation of a programmable controller.

Instance Data Block

Each call of a function block in the STEP 7 user program is assigned a data block which is generated automatically. In the instance data block, the values of the input, output and in/out parameters, as well as the module local data are stored.

Interface, Multipoint

→ MPI

Interrupt

The → Operating System of the C7 CPU has 10 different priority classes which control execution of the user program. These priority classes include interrupts, as for example, process interrupts. When an interrupt occurs, the operating system automatically calls a corresponding organization block where the user can program the reaction desired (for example, in an FB).

L

LAN

A local area network is a bus-type transmission medium which interconnects several nodes. Data transmission can be serial or parallel over electrical conductors or fiber optic cables.

Load Memory

The load memory is part of the C7 CPU. It contains objects generated by the programming device. It is available as a permanently integrated memory.

M

Memory Marker Bits

Memory marker bits are a component part of the → System Memory of the CPU, which are used to store intermediate results. They can be accessed bit by bit, byte by byte, word by word or double word by double word.

Memory Reset

During a memory reset of the → **C7 CPU**, the following memories are cleared:

- The → Work Memory
- The write/read area of the → Load Memory
- The → System Memory
- The → Backup Buffer

The user program is reloaded from the flash memory.

The following memories are cleared in a memory reset of the → **C7 OP**.

- The → Work Memory
- The → Configuration Memory.

After this procedure the C7 OP no longer contains any user-specific data.

Message Level

Operating level of the C7 at which any messages triggered in the system are displayed.

Message Logging

Printout of fault and event messages parallel to display output.

Module Parameters

Module parameters are used to set the module reactions. A difference is made between static and dynamic module parameters.

MPI

The multipoint interface (MPI) is the SIMATIC S7 interface for programming devices. It enables several nodes (programming devices, text displays, operator panels) to be operated simultaneously at one or several central processing units. The nodes with MPI interfaces are networked. Each node is identified by an address (MPI address).

N

Nesting Depth

A block can be called from another block by means of block calls. The nesting depth is the number of simultaneously called → Code Blocks.

Network

A network is a number of C7 systems and/or S7-300 systems and further terminals, such as programming devices via connecting cables for the purpose of data communication.

Normal Operation

Operating mode of the C7; in this mode, messages are displayed and entries can be made in displays.

O

OB

→ Organization Block

OB Priority

The → Operating System of the C7 differentiates between various priority classes, for example, cyclic program processing, process-interrupt-controlled program processing. Each priority class is assigned → Organization Blocks (OBs), where the S7 user can program a reaction. As a standard, the OBs have different priorities to which they are processed when they occur simultaneously or when they interrupt each other.

Operating System of the C7 CPU

The operating system of the C7 CPU organizes all functions and sequences of the C7, which are not connected to a specific control task.

Organization Block

Organization blocks (OBs) represent the interface between the operating system of the C7 CPU and the user program. The sequence of user program processing is defined in the organization blocks.

Output Field

Field for the display of an actual value.

P

Parameterization Parameterization is the act of defining the module characteristics.

Parameters 1. Variable of a STEP 7 code block
2. Variable for setting the module reaction (one or several per module). Each module is supplied with a basic setting that can be changed by means of the STEP 7 tool *S7 Configuration*.
Parameters can be → Dynamic or → Static.

Parameters, Dynamic In contrast to static parameters, dynamic parameters of modules can be changed during running operation by calling an SFC in the user program, for example, limit values of an analog signal input module.

Parameters, Static In contrast to dynamic parameters, static parameters of modules cannot be changed by means of the user program, but only via the STEP 7 tool *S7 Configuration*, for example, input delay of a digital signal input module.

Password Password Level A password with a defined password level is required for accessing a protected function. The password level corresponds to the authorization level allocated to the operator. At the configuring stage, the necessary password level can be preset in the range from 0 (lowest level) to 9 (highest level).

PG → Programming Device

PLC → Programmable Controller

Process Image The process image is a component part of the → System Memory of the C7 CPU. At the beginning 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 to the output modules as signal state.

Process Interrupt A process interrupt is triggered by interrupt-generating modules as a result of a certain event in the process. The process interrupt is signalled to the C7 CPU. Depending on the priority of the interrupt, the associated → Organization Block is then processed.

Programmable Controller

Programmable controllers (PLCs) are electronic control devices whose functions are stored in the controller in the form of a program. The configuration and wiring of a PLC therefore do not depend on the actual functions of the control. Programmable controllers and computers have similar structures: they consist of a CPU (central processing unit) with memory, input/output modules and an internal bus system. The I/O and programming language are tailored to the requirements of open-loop control technology.

Programming Device

Programming devices are principally personal computers that are industry-standard, compact and transportable. They are characterized by a special hardware and software for SIMATIC programmable controllers.

R

RAM

A RAM (Random Access Memory) is a read/write memory medium. Its memory cells can be addressed directly and their contents changed individually. RAMs are used as data and program memories.

Reference Data

Reference data are used for verifying program execution in the C7 CPU. They comprise cross-reference lists, pin assignment lists, program structure, list of unassigned addresses and list of missing identifiers. The *STEP 7 User Manual* describes how these data can be read.

Retentivity

Retentive data areas and retentive timers, counters and memory marker bits retain their contents in the case of a complete restart or power off.

S

Selection Field

Field provided for assigning values to a parameter (values can be selected from a list of permissible values).

SFB

→ System Function Block

SFC

→ System Function

Signal Module

Signal modules (C7 I/O) are the interface between the process and the C7. Signal modules comprise digital input and output modules and analog input and output modules.

Softkeys Keys with variable assignments (depending on the current display entry).

START-UP The START-UP mode is active during the transition from the STOP to the RUN state.

Startup Test Verification of the CPU status and the memory status each time the C7 is powered up.

STEP 7 Programming language for generating user programs for SIMATIC S7 programmable controllers.

STEP 7 Tool A STEP 7 tool is an application of → STEP 7, which is tailored to a specific task.

Substitute Value Substitute values are values which are output to the process in the case of faulty signal output modules or which are used in the user program instead of a process variable in the case of faulty signal input modules. The substitute values can be specified in advance by the user (for example, maintain old value).

System Diagnosis System diagnosis comprises the recognition, evaluation and signalling of errors which occur within the programmable controller. Examples of such errors include: Program errors or module failures. System errors can be indicated via LEDs or via the STEP 7 tool *S7 Information*.

System Function A system function (SFC) is a → Function integrated in the operating system of the CPU, which can be called in the STEP 7 user program if required.

System Function Block A system function block (SFB) is a → Function Block integrated in the operating system of the CPU, which can be called in the STEP 7 user program if required.

System Memory The system memory is a RAM integrated in the central processing unit. The address areas (for example, timers, counters, memory marker bits) and data areas required internally by the → Operating System (for example, communication buffer) are stored in the system memory.

System Message It reports internal states in the C7 and in the controller.

T

Time-of-Day Interrupt

→ Interrupt, Time-of-Day

Timers

Timers are component parts of the → C7 CPU. Asynchronous to the user program, the contents of the ”timer cells” are automatically updated by the operating system. By means of STEP 7 instructions the exact function of the timer cell (for example, on delay) is determined and processing (for example, starting) initiated.

Tool

→ STEP 7 Tool

Transfer Mode

Operating mode of the C7 used for transferring data from the programming device to the C7 and vice versa. A distinction is made between a transfer (S7 trans) via the MPI and via the printer port (trans).

U

User Memory

The user memory contains → Code Blocks and → Data Blocks of the user program. The user memory is integrated in the C7 CPU as flash memory. In general, however, the user program is executed from the → Work Memory of the C7 CPU.

User Program

The user program contains all instructions, declarations and data for signal processing that are used for controlling a plant or process. It is assigned to a programmable module (e.g. C7 CPU, FM) and can be split up into smaller units (blocks).

W

Work Memory (RAM)

The work memory is a RAM in the → C7 where the processor accesses the user program during program processing.

Index

A

B

C

D

E

F

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

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: ☐
6. ☐
7. ☐
8. ☐

Additional comments:

_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _
_ _