SIEMENS

SIMATIC

S7-300
S7-300 CPU Data: CPU 315T-2 DP

Manual

12/2005
A5E00427933-02

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

Danger

indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device 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. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 48 48
90437 NÜRNBERG
GERMANY

Order No.: A5E00427933-02
Edition 12/2005

Table of contents

Tables

Preface

1

Purpose of this manual

This manual contains all the necessary information for the installation, the communication functions, the memory concept, the cycle and response times as well as the technical specifications of the Technology CPU. You will then learn the points to consider when upgrading to the CPU discussed in this manual.

Required basic knowledge

Knowledge required to understand this manual:

- General knowledge of automation
- Knowledge of motion control
- Knowledge of the STEP 7 basic software.
  For further information, refer to the *Programming with STEP 7 V5.3* manual.

Scope of this manual

This manual is valid for the following CPU with the following firmware and hardware versions:

Table 1-1 Scope of this manual

CPU	Order number	As of version	
		Firmware	Hardware
CPU 315T-2 DP	6ES7 315-6TG10-0AB0	V2.4/V3.2	02

This manual describes the properties and differences of the Technology CPU compared with the installation manual, *S7-300 Automation Systems: Hardware and Installation: CPU 31xC and CPU 31x.*

Note

This manual contains the descriptions of all current modules.

For new modules, or modules of a more recent version, we reserve the right to include a Product Information containing latest information.

Approvals

The SIMATIC S7-300 product series has the following approvals:

- Underwriters Laboratories, Inc.: UL 508 (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 No. 142 (Process Control Equipment)
- Factory Mutual Research: Approval Standard Class Number 3611

CE marking

The SIMATIC S7-300 product series satisfies the requirements and safety specifications of the following EC directives:

- EC Directive 73/23/EEC "Low-voltage directive"
- EC Directive 89/336/EEC "EMC Directive"

C-tick mark

The SIMATIC S7-300 product series is compliant with AS/NZS 2064 (Australia and New Zealand).

Standards

The SIMATIC S7-300 product series is compliant with IEC 61131-2.

The Technology CPU is oriented to the planned standards for PLCopen V2.0 and PROFIdrive V3.0.

Documentation classification

This manual is part of the Technology CPU documentation package.

All of these manuals are available as electronic manuals on the CD-ROM of the *S7-Technology* option package.

Table 1-2 Documentation for the Technology CPU

Title		Contents
Getting Started		
🗍	*CPU 317T-2 DP: Controlling a SINAMICS S120* *CPU 317T-2 DP: Controlling a physical axis* *CPU 317T-2 DP: Controlling a virtual axis*	The example used in these Getting Started manuals guides you through the various steps in commissioning required to obtain a fully functional application.
CPU Data Reference Manual		
🗍	→ *CPU data: CPU 315T-2 DP* (you are reading this manual) *CPU data:CPU 317T-2 DP*	Description of the operation, functions and technical specifications of the CPU 315T-2 DP and CPU 317T-2 DP
Manual		
🗍	*S7-Technology*	Description of the individual technological functions: • Application and benefits • Fundamentals and configuration • Loading, testing and diagnostics • PLCopen functions

Title		Contents
Software Installation Manual		
	S7-300 Automation Systems: *Installation: CPU 31xC and CPU 31x*	Description of the configuration, installation, wiring, networking and commissioning of an S7-300
Module Data Reference Manual		
	S7-300 Automation Systems:Module Data	Technical data and descriptions of the functions of signal modules, power supply modules and interface modules
Instruction List		
	CPU 31xC, CPU 31x IM 151-7 CPU, BM 147-1 CPU, BM 147-2 CPU	List of the CPU operation set and their execution times. List of the executable blocks (OBs/SFCs/SFBs) and their execution times

In addition to this documentation package, you require the following manual:

Table 1-3 Additional documentation for the Technology CPU

System Software for S7-300/400, System and Standard Functions reference manual		
	The reference manual is part of the STEP 7 documentation package	Description of the SFCs, SFBs and OBs of the CPUs. You can also find the description in the online help for STEP 7.

Further support

Do you have more questions about using the products described in the manual? Then contact the Siemens representative or office nearest you.

http://www.siemens.com/automation/partner

Training center

SIEMENS offers a range of courses to help you to get started with your S7-300 automation system. Please contact your regional Training Center, or the central Training Center in D-90327 Nuremberg.

Telephone:+49 (911) 895-3200

http://www.sitrain.com

SIMATIC documentation on the Internet:

You can find the documentation free of charge on the Internet at:

http://www.ad.siemens.de/support

Use the Knowledge Manager provided there to quickly find the required documentation. You can enter any questions or suggestions for the documentation in the forum. You will receive a quick reply.

Technical support

You can reach the technical support team for all A&D projects

- Using the support request web form:
  http://www.siemens.de/automation/support-request
- Telephone: + 49 180 5050 222
- Fax: + 49 180 5050 223

Further information about SIEMENS technical support is available on the Internet at http://www.siemens.com/automation/service

Service & Support on the Internet

In addition to our documentation, we offer a comprehensive knowledge base online on the Internet at:

http://www.siemens.com/automation/service&support

There you will find:

- The latest product information, FAQs (Frequently Asked Questions), downloads, tips and tricks.
- Our newsletter, providing you with the latest information about your products.
- A Knowledge Manager to find the right documents for you.
- Our bulletin board, where users and specialists share their knowledge worldwide.
- Your local contact partner for Automation & Drives in our Partner Database.
- Information about field service, repairs, spare parts and lots more under "Services."

Product Overview

2

Introduction

The current trend in the field of automation is toward PLC-integrated solutions. This also applies to technology and motion control applications.

Integrated technology of Technology CPU

With the Technology CPU, technology and motion control functions are integrated in one SIMATIC CPU.

The Technology CPU incorporates:

- SIMATIC CPU 31x-2 DP
- PLCopen-compliant motion control functions
- Technological configurations (technology objects, axis configurations, tools)

The Technology CPU is completely integrated in the SIMATIC family and thus in the TIA environment.

Field of application

The Technology CPU is especially suited to solving the following control tasks:

- Control tasks and technology requirements primarily relating to motion control in the SIMATIC S7-300
- Motion tasks for up to eight coupled axes or single axes
- Technological tasks, e.g. gearing and camming, position-controlled positioning (operating modes: absolute, relative, additive and superimposed), travel to fixed stop, probe-based print mark correction, position- or time-dependent cam control).

The Technology CPU is designed for use with flow machines, processing/assembly lines, flying shears, labeling equipment, drum feeds or simple gantries (without interpolation).

Interfaces

The Technology CPU has two interfaces:

- One MPI/DP interface, parameterizable as an MPI or DP interface (master or slave).
- one DP (DRIVE) interface for connection of drive systems

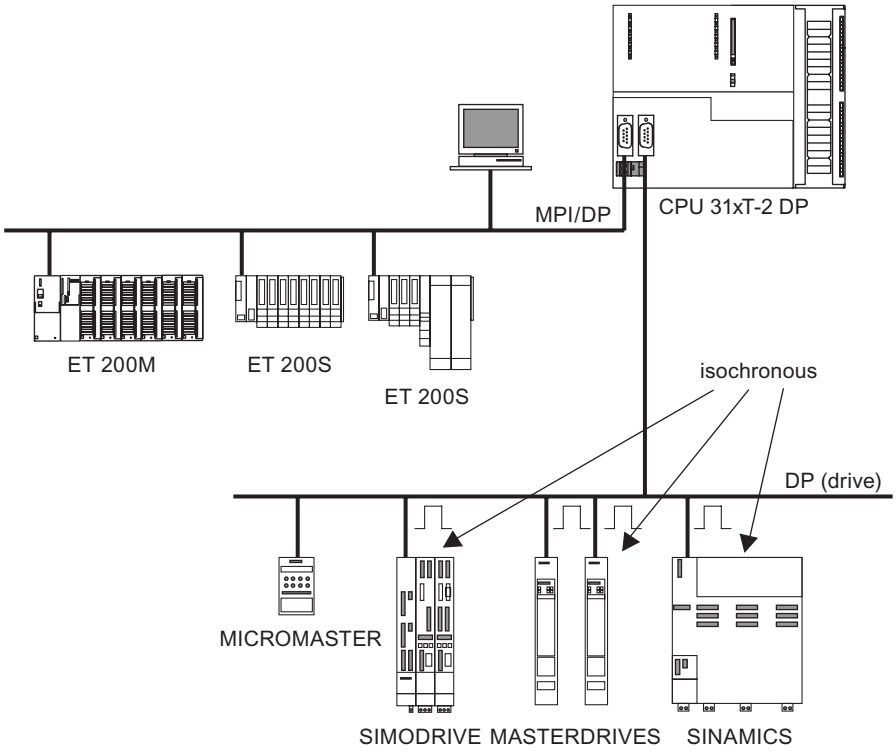


Figure 2-1 Typical configuration with the Technology CPU

MPI/DP interface

The MPI/DP interface is used to connect additional SIMATIC components, such as PG, OP, S7 controllers and distributed I/O. Extensive networks can be built up in DP interface mode.

DP(DRIVE) interface

The DP(DRIVE) interface is optimized for the connection of drives. It supports all the major SIEMENS drive types:

- MICROMASTER 420/430/440 and COMBIMASTER 411
- SIMODRIVE 611 universal
- SIMODRIVE POSMO CD/SI/CA
- MASTERDRIVES MC/VC
- ET 200M with IM 153-2 (isochronous!) and SM 322 for additional cam output
- ET 200S with IM 151-1 high feature
- SINAMICS S120 (optional with TM15 or TM17 high feature for high-speed cams)
- Analog drive interface ADI4

- Isochronous PROFIBUS encoder "SIMODRIVE sensor isochronous"

The components configured in HW Config are displayed in the "Hardware Catalog" window in HW Config. To show the screen, select profile "SIMATIC Technology CPU" in HW Config.

To ensure that the profile's selection list is complete, you must have installed the most recent version of S7-Technology.

The isochronous capability means that even high-speed processes can be controlled with excellent precision.

Integrated inputs and outputs for integrated technology

The Technology CPU has 4 digital inputs and 8 digital outputs integrated. You use these inputs and outputs for technology functions, e.g. reference point acquisition (reference cams) or high-speed output cam switching signals. The inputs and outputs can also be used with technology functions in the *STEP 7* user program.

Configuring and programming

The Technology CPU is configured and programmed completely in *STEP 7* (Version 5.3 SP3 and later) and optional package *S7-Technology V3.0* (the *S7-Technology* optional package is integrated after installation in *STEP 7*).

STEP 7 *HW Config* is the tool used to configure all hardware components of the system (e.g. to create subnets on the two interfaces MPI/DP and DP(DRIVE)) including the drive equipment.

You will need optional package *S7-Technology* to parameterize the so-called "technology objects", e.g. axes, cams, output cams and probes.

The objects are parameterized in specially provided screens. The technology object data are stored in data blocks for use by the *STEP 7* user program.

S7-Technology also includes a library containing PLCopen-compliant standard function blocks which are used to program the motion control tasks themselves. You call these standard FBs in your *STEP 7* user program.

STEP 7 languages LAD, CSF and STL and all the required engineering tools, e.g. S7-SCL or S7-GRAPH are provided to enable you to create the *STEP 7* user program (incl. motion control tasks).

Single-tier configuration

The Technology CPU supports only single-tier configurations.

Operator controls and indicators

3

Operator controls and indicators of the CPU

The following diagram shows the operator controls and indicators on the Technology CPU.

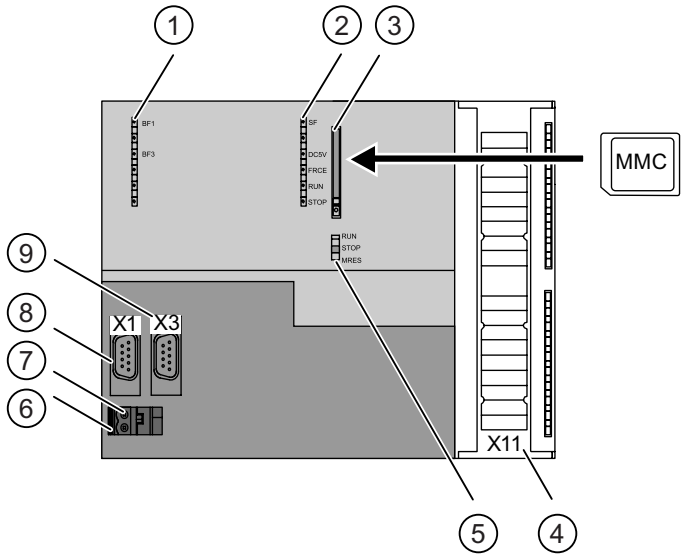


Figure 3-1 Operator controls and indicators of the Technology CPU

Table 3-1 Operator controls and indicators of the Technology CPU

The number in the diagram	points to the following element on the Technology CPU
1	Bus error indicators
2	Status and error displays
3	Slot for the Micro Memory Card (MMC), incl. the ejector
4	Connection of the integrated I/Os
5	Mode selector switch
6	Power supply connection
7	Grounding slide
8	Interface X1 MPI/DP
9	Interface X3 DP(DRIVE)

Integrated inputs and outputs for integrated technology

You can use the integrated technology inputs and outputs for technology functions and configure them using *S7T Config* (included in the optional package *S7-Technology*).

The digital outputs are provided for high-speed camming functions. They can be programmed with technology functions in the STEP 7 user program. Digital inputs can be used with technology functions such as reference point acquisition (reference cam) as well as with technology functions in the STEP 7 user program.

You use the integrated inputs and outputs for applications in which rapid technological processing is of prime importance.

If you wish to evaluate other inputs and outputs in the STEP 7 user program, they can be interconnected in the usual way by means of supplementary input/output modules.

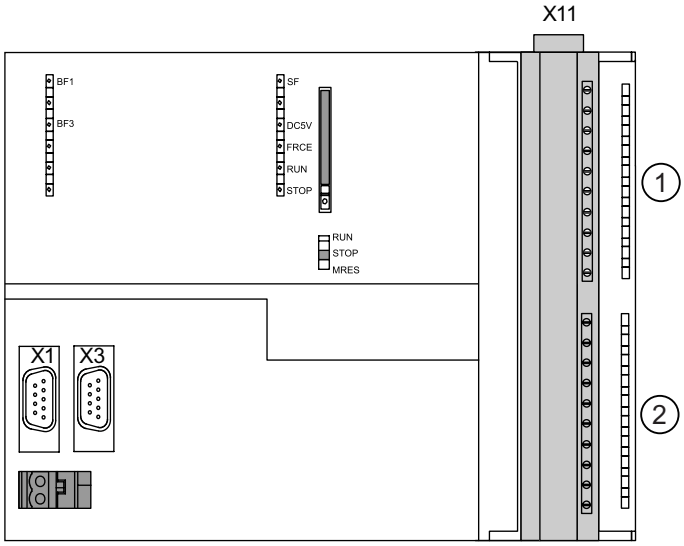


Figure 3-2 Integrated technology inputs and outputs on the Technology CPU with open front door

Table 3-2 Integrated technology inputs and outputs on the Technology CPU

The number in the diagram	points to the following integrated I/Os
1	4 digital inputs
2	8 digital outputs

Slot for the SIMATIC Micro Memory Card (MMC)

A SIMATIC Micro Memory Card (MMC) is used as memory module. The MMC is used as load memory and as a transportable data medium.

Note

Because the Technology CPU does not have an integrated load memory, you must plug in an MMC to operate it.

You require an MMC, 4 MB and larger; for an operating system update you require an MMC with 8 MB.

Mode selector switch

The mode selector switch is used to set the current CPU operating mode.

Table 3-3 Switch positions of the mode selector switch

Position	Meaning	Description
RUN	RUN mode	The CPU executes the user program.
STOP	STOP mode	The CPU does not execute a user program.
MRES	CPU memory reset	Mode selector switch position with pushbutton function for CPU memory reset. A general CPU reset by means of a mode selector switch requires a specific sequence of operation.

Power supply connection

Each CPU is equipped with a double-pole power supply socket. The connector with screw terminals is inserted into this socket when the CPU is delivered.

Status and error displays

The CPU is equipped with the following LEDs:

Table 3-4 Status and error displays of the CPU

LED	Color	Meaning
SF	red	Hardware or software error
BF1	red	Bus errors (MPI/DP)
BF3	red	Bus error on DP(DRIVE)
5 VDC	green	5V power supply for the CPU and the S7-300 bus
FRCE	yellow	Active force job
RUN	green	CPU in RUN The LED flashes during STARTUP at a rate of 2 Hz, and in HOLD state at 0.5 Hz.
STOP	yellow	CPU in STOP, or HOLD, or STARTUP The LED flashes • At 0.5 Hz on general reset request • At 2 Hz during general reset • At 2 Hz during shutdown (LED RUN lit).

Shutdown

What happens during shutdown?

1. The control of the Technology CPU is already in STOP mode during "shutdown". The outputs of the centralized and distributed I/Os on the MPI/DP are deactivated. The "STOP" LED flashes at 2 Hz. The "RUN" LED lights up.
2. The integrated inputs/outputs for integrated technology and the distributed I/Os on DP(DRIVE) are still active during shutdown.
3. The integrated technology of the Technology CPU shuts down the drives on PROFIBUS DP(DRIVE) in a controlled manner.
4. The integrated technology then also switches to STOP. The integrated inputs/outputs for integrated technology and the distributed I/Os on DP(DRIVE) are deactivated. The "STOP" LED lights up.

The maximum duration of shutdown depends on your configuration in S7T Config.

Caution

The distributed I/Os on DP(DRIVE) cannot be controlled from the user program during "shutdown". The outputs which can be controlled with technology function "MC_WritePeripherie" retain their last current setting.

Reference

Further information

- about CPU operating modes can be found in the *STEP 7 Online Help*.
- about operating the mode selector switch for a general CPU reset can be found in the installation manual, chapter *Commissioning*.
- about understanding LED displays in event of faults/trouble shooting can be found in the installation manual, chapter *Test Functions, Diagnostics and Fault Correction*.
- about operating MMCs and the memory concept can be found in chapter *Memory Concept.*

Setting up an S7-300 with a Technology CPU

4

4.1 Overview

This section

contains the information which differs from the content of installation manual *S7-300 Programmable Controller, Assembly: CPU 31xc and CPU 31x* or useful supplementary information.

4.2 S7-300 components

Which components do you need to build an S7-300 with Technology CPU?

The following diagram shows one possible configuration:

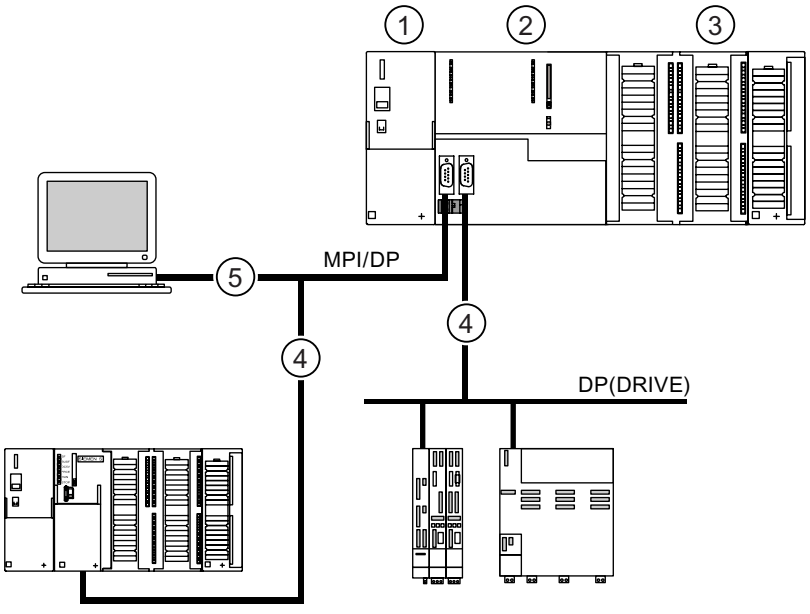


Figure 4-1 S7-300 components

Table 4-1 S7-300 components

The number in the diagram	points for the following component of an S7-300 system
(1)	Power supply (PS) module
(2)	Central processor unit (CPU)
(3)	Signal module (SM)
(4)	PROFIBUS cable
(5)	Cable for connecting a programming device (PG) or for networking with other SIMATIC controls

You use a programming device (PG) to program the S7300. You connect the PG to the CPU by means of a PG cable.

Using the PROFIBUS bus cable, you connect the CPU

- to other SIMATIC controls via the MPI/DP interface
- to the drives via the DP(DRIVE) interface.

No PG/OP on DP(DRIVE)

We do not recommend that you connect a PG/OP to DP(DRIVE).

Reason: If you connect a PG/OP to DP(DRIVE), the properties of DP(DRIVE) change (e.g. isochronism), and the synchronism between drives may be lost as a result. Always therefore connect a PG/OP to the MPI/DP interface and access the DP(DRIVE) via the "Routing" function.

4.3 Configuring

Single-tier configuration

The Technology CPU supports only single-tier configurations.

4.4 Subnets

4.4.1 Expanding and Networking Subnets

Overview: Subnets with the Technology CPU

The Technology CPU provides the following subnets:

- Multi Point Interface (MPI) or PROFIBUS DP
- DP(DRIVE): Optimized for drives

Transmission rate

Maximum transmission rates:

- MPI/PROFIBUS DP: 12 Mbaud

  We recommend that you set 12 Mbaud for the Technology CPU
- DP(DRIVE): 12 Mbaud

Note

Before you transfer projects to the Technology CPU via the MPI/DP interface, you should increase the baud rate to at least 1.5 Mbaud or else the data transmission can take a very long time (up to 15 minutes at 187.5 kbaud).

Number of nodes

Maximum number of nodes per subnet:

Table 4-2 Subnet nodes

Parameters	MPI	PROFIBUS DP	PROFIBUS DP (DRIVE)
Number	127	126	33
Addresses	0 to 126	0 to 125	1 to 125
Comments	Default: 32 addresses Reserved addresses: • Address 0 for PG • Address 1 for OP	of which: 1 master (reserved) 1 PG connection (address 0 reserved) 124 slaves or other masters	of which: • 1 master (reserved) and 32 slaves or drives

4.4.2 Interfaces

MPI/DP interface

You can reconfigure this interface in STEP 7 as a PROFIBUS DP interface.

The MPI (Multi-Point Interface) represents the CPU interface for PG/OP connections, or for communication on an MPI subnet.

The PROFIBUS DP interface is mainly used to connect distributed I/O. PROFIBUS DP allows you to create large subnets, for example.

Interfaces

Table 4-3 Possible operating modes of interfaces on the Technology CPU

MPI/DP interface (X1)	DP(DRIVE) interface (X3)
• MPI • DP master • DP slave	• DP master for DP(DRIVE)

Which devices can you connect to which interface?

Table 4-4 Connectable devices

MPI	PROFIBUS DP	DP (drive)
• PG/PC • OP/TD • S7-300/400 with MPI interface • S7-200 (with 19.2 kbaud only)	• PG/PC • OP/TD • DP slaves • DP master • Actuators/sensors • S7-300/400 with PROFIBUS DP interface	• MICROMASTER 420/430/440 and COMBIMASTER 411 • SIMODRIVE 611 universal • SIMODRIVE POSMO CD/SI/CA • MASTERDRIVES MC/VC • ET 200M with IM 153-2 (isochronous!) • ET 200S with IM 151-1 • SINAMICS S120 (optional with TM15 or TM17 high feature for high-speed cams) • SIMODRIVE sensor isochronous • Analog drive interface ADI4 Tip: You will find a list of connectable devices in STEP 7 under the "SIMATIC Technology CPU" profile in the hardware catalog.

4.5 Addressing

Slots of the S7-300 and associated module start addresses

The Technology CPU is assigned to two slot numbers: 2 and 3.

The input and output addresses for I/O modules begin at the same module start address.

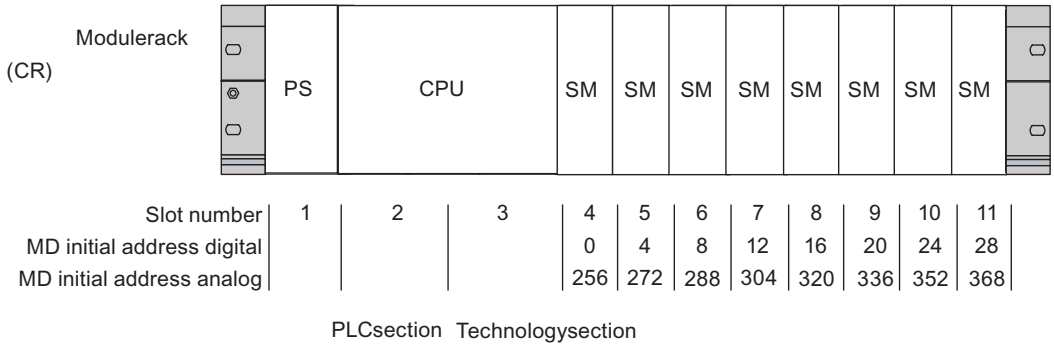


Figure 4-2 Slots of the S7-300 with Technology CPU and associated module start addresses

Integrated inputs and outputs for integrated technology

The Technology CPU has 4 integrated digital iputs and 8 integrated digital outputs for integrated technology. You use these integrated inputs and outputs for technology functions, e.g. reference point acquisition (reference cams) or high-speed output cam switching signals.

The integrated inputs and outputs can also be used with technology function in the STEP 7 user program.

You use the integrated inputs and outputs for applications in which rapid technological processing is of prime importance.

DP address areas

The CPU 315T-2 DP has the following address areas:

- For inputs and outputs respectively: 2048 bytes
- Of which in the process image, for inputs and outputs respectively: Bytes 0 to 127

DP (DRIVE) address areas

The CPU 315T-2 DP has the following DP(DRIVE) address areas:

- For inputs and outputs respectively: 1024 bytes
- Of which in the I/O image, for inputs and outputs respectively: Bytes 0 to 63

4.6 Commissioning

Conditions

If you wish to utilize the full scope of CPU functions, you will require

- STEP 7 as of V 5.3 + SP 3 and option package *S7-Technology V3.0*
- S7-300 is installed
- S7-300 is wired
- For a networked S7-300:
  - MPI/PROFIBUS addresses are set
  - Terminating resistors on the segments are enabled

4.7 Operating system

Technology CPU operating system

To meet the requirements of the integrated technology, technology functions have been added to the standard CPU operating system to obtain the technology operating system.

The technology operating system is included in the project and the configuration. In other words, if you load a project created with S7-Technology to the Technology CPU, the technology operating system is automatically transferred at the same time.

Updating the operating system

You can order the latest operating system versions from your Siemens contact or download it from the Internet at (Siemens Homepage; Industrial Automation, Customer Support).

4.8 Status and error displays of the Technology CPU

Status and error displays of the Technology CPU

Table 4-5 Status and error displays of the Technology CPU

LED					Meaning
SF	5 VDC	FRCE	RUN	STOP	
Off	Off	Off	Off	Off	CPU power supply missing. Remedy: Check whether the power supply module is connected to mains and switched on. Check whether the CPU is connected to the power supply and switched on.

LED					Meaning
SF	5 VDC	FRCE	RUN	STOP	
Off	On	X (see the description)	Off	On	The CPU is in STOP mode. Remedy: Start the CPU.
On	On	X	Off	On	The CPU is in STOP mode as a result of error. Remedy: refer to the tables below, evaluation of the SF LED
X	On	X	Off	Flashes (0.5 Hz)	The CPU requests memory reset.
X	On	X	Off	Flashes (2 Hz)	The CPU executes memory reset.
X	On	X	Flashes (2 Hz)	On	The CPU is in startup mode.
X	On	X	Flashes (0.5 Hz)	On	The CPU was halted by a programmed break point. For details, refer to the Programming Manual *Programming with STEP 7*.
On	On	X	X	X	Hardware or software error Remedy: refer to the tables below, evaluation of the SF LED
X	X	On	X	X	You enabled the Force function For details refer to the Programming Manual *Programming with STEP 7*.
X	X	X	On	Flashes (2 Hz)	STOP/shutdown What happens during shutdown? The control of the Technology CPU is already in STOP mode during "shutdown". The outputs of the centralized and distributed I/Os are deactivated. The integrated inputs/outputs for integrated technology and the ET 200M on the DP(DRIVE) are still active during shutdown. The integrated technology of the Technology CPU shuts down the drives on PROFIBUS DP(DRIVE) in a controlled manner. The integrated technology of the CPU then also goes into STOP. The integrated inputs/outputs for integrated technology and the ET 200M on the DP(DRIVE) are deactivated. The maximum duration of shutdown depends on your configuration in S7TConfig.
X	X	X	Flashes (0.5 Hz)	Flashes (2 Hz)	HOLD/shutdown
Flashes	Flashes	Flashes	Flashes	Flashes	Internal errors in Technology CPU. Contact your local SIEMENS partner.

Status and error displays for DP or DP(DRIVE)

Table 4-6 LEDs BF1 and BF3

LED		Meaning
BF1	**BF3**	
On/ flashes	X	Error on the PROFIBUS DP interface of the Technology CPU. Remedy: See table LED BF1 illuminated
X	On/ flashes	Error on the DP(DRIVE) interface Remedy: See table *LED BF1 flashing*

Description of status X:
The LED can assume the On or Off state. This status, however, is irrelevant for the current CPU function. For example, the states Force On or Off do not influence the CPU STOP status

Table 4-7 LED BF1 illuminated

Possible Errors	CPU reaction	Possible Remedies
• Bus fault (physical fault) • DP interface error • Different transmission rates in multiple DP master mode. • If the DP slave / master interface is active: Short-circuit on the bus. • With passive DP slave interface: transmission rate search, i.e. there are no other active nodes on the bus (a master, for example)	Call of OB 86 (when CPU is in RUN mode). CPU switches to STOP if OB 86 is not loaded.	• Check the bus cable for short-circuit or breaks. • Evaluate the diagnostics. Reconfigure or correct the configuration.

Table 4-8 ED BF1 flashes

Possible Errors	CPU reaction	Possible Remedies
The CPU is DP master / active slave: • Failure of a connected station • At least one of the configured slaves cannot be accessed. • Incorrect configuration	Call of OB 86 (when CPU is in RUN mode). CPU switches to STOP if OB 86 is not loaded.	Verify that the bus cable is connected to the CPU, or that the bus is not interrupted. Wait until the CPU has completed its startup. If the LED does not stop flashing, check the DP slaves or analyze the diagnostic data of the DP slaves.
The CPU is a DP slave The CPU parameters are incorrectly set. Possible causes: • The response monitoring period has elapsed. • PROFIBUS DP communication is down. • Wrong PROFIBUS address. • Incorrect configuration	Call of OB 86 (if CPU is in RUN mode). CPU switches to STOP if OB 86 is not loaded.	• Check the CPU. • Verify that the bus connector is properly seated. • Check whether the bus cable to the DP master has been disconnected. • Check the configuration and parameter assignment.

Table 4-9 LED BF3 illuminated

Possible Errors	CPU reaction	Possible Remedies
• Bus fault (physical fault) • DP interface error	Error message in the technology DB configured by you.	Check for short-circuit or interruption in the bus cable.

Table 4-10 LED BF3 flashes

Possible Errors	CPU reaction	Possible Remedies
• Failure of a connected station • At least one of the configured slaves cannot be accessed. • Incorrect configuration	Error message in the technology DB configured by you.	Verify that the bus cable is connected to the CPU, or that the bus is not interrupted. Wait until the CPU has completed its startup. If the LED does not stop flashing, check the DP slaves or analyze the diagnostic data of the DP slaves.

Communication with the S7-300

5

5.1 Interfaces

5.1.1 Overview

Overview

The Technology CPU has two interfaces:

- MPI/DP interface (X1)
- PROFIBUS DP(DRIVE) interface (X3)

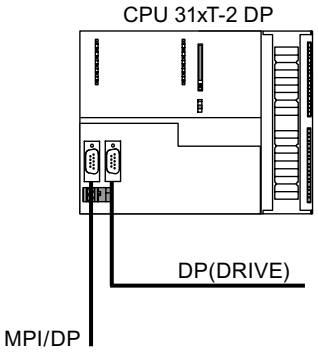


Figure 5-1 Technology CPU interfaces

5.1.2 MPI/DP interface (X1)

Availability

The Technology CPU features an MPI/DP interface (X1). A CPU with MPI/DP interface is supplied with default MPI parameter settings. Depending on your requirements, you may need to reconfigure the interface as a DP interface in STEP 7.

MPI properties

The MPI (Multi-Point Interface) represents the CPU interface for PG/OP connections, or for communication on an MPI subnet.

The typical (default) transmission rate of all CPUs is 187.5 kbaud. You can also set 19.2 kbps for communication with an S7-200. You can set baud rates of up to 12 Mbaud.

The CPU automatically broadcasts its bus configuration via the MPI interface (the transmission rate, for example). A PG, for example, can thus receive the correct parameters and automatically connect to a MPI subnet.

Note

You may only connect PGs to an MPI subnet which is in RUN.
Other stations (for example, OP, TD, ...) should not be connected to the MPI subnet while the system is in RUN. Otherwise, transferred data might be corrupted as a result interference, or global data packages may be lost.

Note

Before you transfer data to the CPU via the MPI interface, you should increase the baud rate to 1.5 Mbaud or else the data transmission can take a very long time (up to 15 minutes at 187.5 kbaud)!

Operating modes of the MPI/DP interface

Operating modes for MPI/DP interface (X1):

- MPI
- DP master
- DP slave
- I slave

Devices capable of MPI communication

- PG/PC
- OP/TD
- S7-300 / S7-400 with MPI interface
- S7-200 (with 19.2 kbaud only)

Properties of PROFIBUS DP

The PROFIBUS DP interface is mainly used to connect distributed I/O. PROFIBUS DP allows you to create extensive subnets, for example.

The PROFIBUS DP interface can be set for operation in master or slave mode, and supports transmission rates up to 12 Mbaud.

The CPU broadcasts its bus parameters (transmission rate, for example) via the PROFIBUS DP interface when master mode is set. A PG, for example, can thus receive the correct parameters and automatically connect to a PROFIBUS subnet. In your configuration you can specify to disable bus parameter broadcasting.

Devices capable of PROFIBUS DP communication

- PG/PC
- OP/TD
- DP slaves
- DP master
- Actuators/sensors
- S7-300/S7-400 with PROFIBUS DP interface

5.1.3 PROFIBUS DP(DRIVE) interface (X3)

Properties

The PROFIBUS DP(DRIVE) interface is used to connect to drive systems. You can connect drive systems in accordance with the PROFIdrive V3.0 profile.

The PROFIBUS DP(DRIVE) interface is configured as a master and supports transmission rates up to 12 Mbaud.

The PROFIBUS DP(DRIVE) interface support isochronous mode.

The CPU sends its bus parameter settings (e.g. baud rate) via the PROFIBUS DP(DRIVE) interface. In your configuration you can specify to disable bus parameter broadcasting.

Using the "Routing" function, you can access the drive parameters of the slaves in the DP(DRIVE) line for the purposes of commissionng and diagnostics. However, diagnostics cannot be performed via PROFIBUS DP(DRIVE) from the *STEP 7* user program.

Note

If you deselect "Startup with different target / actual configurations" in the Technology CPU properties in STEP 7, then the Technology CPU will boot even if the stations configured on DP-DRIVE are missing.

Connectable devices

You can connect drives to PROFIBUS DP(DRIVE), e.g.:

- MICROMASTER 420/430/440 and COMBIMASTER 411
- SIMODRIVE 611 universal
- SIMODRIVE POSMO CD/SI/CA
- MASTERDRIVES MC/VC
- ET 200M with IM 153-2 (isochronous!) and SM 322 for additional cam output
- ET 200S with IM 151-1 high feature
- SINAMICS S120 (optional with TM15 or TM17 high feature for high-speed cams)
- ADI4 (analog drive interface)
- Isochronous PROFIBUS encoder "SIMODRIVE sensor isochronous"

The components configured in HW Config are displayed in the "Hardware Catalog" window in HW Config. To show the screen, select profile "SIMATIC Technology CPU" in HW Config.

To ensure that the profile's selection list is complete, you must have installed the most recent version of S7-Technology.

Non-connectable devices

We would not recommend operating active PROFIBUS stations (PGs, PCs, OPs, TDs, etc.) on PROFIBUS DP(DRIVE). The DP cycle will be burdened by additional access times if you do operate these PROFIBUS stations on DP(DRIVE). In this case, isochronous processing of drive information cannot be guaranteed.

5.2 DPV1 (X1 only as PROFIBUS DP interface)

Acyclical access with DPV1

New automation and process engineering tasks require the range of functions performed by the existing DP protocol to be extended. In addition to cyclical communication functions, acyclical access to non-S7 field devices is another important requirement of our customers, and was implemented in the standard EN 50170. In the past, acyclical access was only possible with S7 slaves.

Definition DPV1

The term DPV1 is defined as a functional extension of the acyclical services (to include new interrupts, for example) provided by the DP protocol. Enhancement of the distributed I/O EN 50170 standard. All the changes concerning new DPV1 functions are included in IEC 61158/ EN 50170, volume 2, PROFIBUS.

Extended functions of DPV1

- Use of any DPV1 slaves from external vendors (in addition to the existing DPV0 and S7 slaves, of course).
- Selective handling of DPV1-specific interrupt events by new interrupt blocks.
- Reading/writing Sibs that conform to standards to the data record (although this can only be used for centralized modules).
- User-friendly SFB for reading diagnostics.

Availability

The Technology CPU, serving as DP master at the MPI/DP interface, features the enhanced DPV1 functionality.

Note

If you want to use the CPU as an intelligent DP slave, remember that it does not have DPV1 functionality.

Requirement for using the DPV1 functionality with DP slaves

For DPV1 slaves from other vendors, you will need a GSD file conforming to EN 50170, revision 3 or later.

Interrupt blocks with DPV1 functionality

Table 5-1 Interrupt blocks with DPV1 functionality

OB	Functionality
OB 40	Process interrupt
OB 55	DPV1: Status interrupt
OB 56	DPV1: Update interrupt
OB 57	DPV1: Vendor-specific interrupt
OB 82	Diagnostic interrupt

Note

You can now also use organizational blocks OB40 and OB82 for DPV1 interrupts.

System blocks with DPV1 functionality

Table 5-2 System function blocks with DPV1 functionality

SFB	Functionality
SFB 52	Read data record from DP slave or centralized module
SFB 53	Write data record to DP slave or centralized module
SFB 54	Read additional alarm information from a DP slave or a centralized module in the relevant OB.
SFB 75	Set any interrupts for intelligent slaves

Note

You can also use SFB 52 to SFB 54 for centralized input/output modules.

Reference

Further information

- about migrating to the Technology CPU can be found in chapter *Information about migrating to a Technology CPU.*
- about the blocks mentioned above can be found in the reference manual *System Software for S7-300/400: System and Standard Software*, or directly to the *STEP 7 Online Help*.

5.3 Communication services on the MPI/DP interface (X1)

5.3.1 Overview of communication services

Selecting the communication service

You need to decide on a communication service, based on functionality requirements. Your choice of communication service will influence

- the functionality available,
- whether an S7 connection is required or not, and
- the connection timing.

The user interface can vary considerably (SFC, SFB, ...), and is also determined by the hardware used (SIMATIC CPU, PC, ...).

Overview of communication services

The table below provides an overview of communication services offered by the CPU.

Table 5-3 CPU communication services

Communication service	Functionality	Time at which the S7 connection is established ...	via MPI	via DP	to DP(DRIVE)
PG communication	Commissioning, test, diagnostics	from the PG, starting when the service is being used	X	X	-
OP communication	Operator control and process monitoring	from the OP at Power ON	X	X	-
S7-based communication	Data exchange	is programmed at the blocks (SFC parameters)	X	-	-
S7 communication	Data exchange	as server only, communication link is set up by the communication partner	X	X	-
Global data communication	Cyclic data exchange (for example, flag bits)	does not require an S7 connection	X	-	-
Routing PG functions*	for example testing, diagnostics on other networks also	from the PG, starting when the service is being used	X	X	X

* Functions can be routed only to DP(DRIVE), but not on DP(DRIVE)!

5.3.2 PG communication

Properties

PG communication is used to exchange data between engineering stations (PG, PC, for example) and SIMATIC modules which are capable of communication. This service is available for MPI, PROFIBUS and Industrial Ethernet subnets. Transition between subnets is also supported.

PG communication provides the functions needed to download / upload programs and configuration data, to run tests and to evaluate diagnostic information. These functions are integrated in the operating system of SIMATIC S7 modules.

A CPU can maintain several simultaneous online connections to one or multiple PGs.

Reference

Further information

- on SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* or to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

5.3.3 OP communication

Properties

OP communication is used to exchange data between operator stations (OP, TD, for example) and SIMATIC modules which are capable of communication. This service is available for MPI, PROFIBUS and Industrial Ethernet subnets.

OP communication provides functions you require for monitoring and modifying. These functions are integrated in the operating system of SIMATIC S7 modules.

A CPU can maintain several simultaneous connections to one or several OPs.

Reference

Further information

- on SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* or to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

5.3.4 S7 basic communication

Properties

S7 basic communication is used to exchange data between S7 CPUs and the communication-capable SIMATIC modules within an S7 station (acknowledged data exchange). Data are exchanged across non-configured S7 connections. The service is available via MPI subnet, or within the station to function modules (FM).

S7-based communication provides the functions you require for data exchange. These functions are integrated into the CPU operating system.

You can utilize this service by means of "System function" (SFC) user interface.

Reference

Further information

- on SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* or to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

5.3.5 S7 communication

Properties

The CPUs act as servers in the S7 communication system. In this case, the connection is always established by the communication partner. This service is available for MPI, PROFIBUS and Industrial Ethernet subnets.

Services are performed by the operating system without an explicit user interface.

Note

S7 communication as a client can be implemented by means of CPs and loadable FBs.

Reference

Further information

- on SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* or to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

5.3.6 Global data communication

Global data communication

Global data communication is used for cyclic exchange of global data (for example, I, Q, M) between SIMATIC S7 CPUs (data exchange without acknowledgement). One CPU broadcasts its data to all other CPUs on the MPI subnet. This function is integrated in the CPU operating system.

Send and receive conditions

Conditions which should be satisfied for GD communication:

- The transmitter of a GD packet must meet the following requirement:
  Reduction ratio$_{transmitter}$ x cycle time$_{transmitter}$ ≥ 60 ms
- The receiver of a GD packet must meet the following requirement:
  Reduction factor$_{receiver}$ x cycle time$_{receiver}$
  < reduction factor$_{transmitter}$ x cycle time$_{transmitter}$

A GD packet may be lost if there requirements are not met. The reasons being:

- the performance of the "smallest" CPU in the GD circuit
- asynchronous transmitting / receiving of global data at the stations

When setting in *STEP 7*: “Transmit after each CPU cycle“, and the CPU has a short scan cycle time (< 60 ms), the operating system might overwrite a GD packet of the CPU before it is transmitted. The loss of global data is indicated in the status box of a GD circuit, if you set this function in your *STEP 7* configuration.

Reduction factor

The reduction ratio specifies the cyclic intervals for GD communication. You can set the reduction ratio when you configure global data communication in *STEP 7*. For example, if you set a reduction ratio of 7, global data are transferred only with every 7th cycle. This reduces CPU load.

GD resources of the CPU

Table 5-4 GD resources of the CPU

Parameters	Technology-CPU
Number of GD circuits per CPU	Max. 8
GD packets transmitted per GD circuit	Max. 1
GD packets transmitted by all GD circuits	Max. 8
GD packets received per GD circuit	Max. 1
GD packets received by all GD circuits	Max. 8
Data length per GD packet	Max. 22 bytes
Consistency	Max. 22 bytes
Min. reduction ratio (default)	1 (8)

Reference

Further information

- about SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* and to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

5.3.7 Routing

Definition

The routing function enables you to attach a PG/PC at any point in the network and establish a connection to all drives accessible via gateways.

Accessing drives in a DP(DRIVE) subnet from a PG/PC

Test, diagnostic and parameterizing functions can be routed via the MPI/DP interface (X1) to the DP(DRIVE) subnet with the Technology CPU.

The Technology CPU provides a certain number of connection resources for routing. These connections are available in addition to the S7 connection resources.

The number of routing connections can be found in the Technical Specifications.

Network gateway

Gateways between subnets are routed in a SIMATIC station that is equipped with interfaces to the respective subnets. The figure below shows the Technology CPU (DP master) acting as router for subnets 1 and 2.

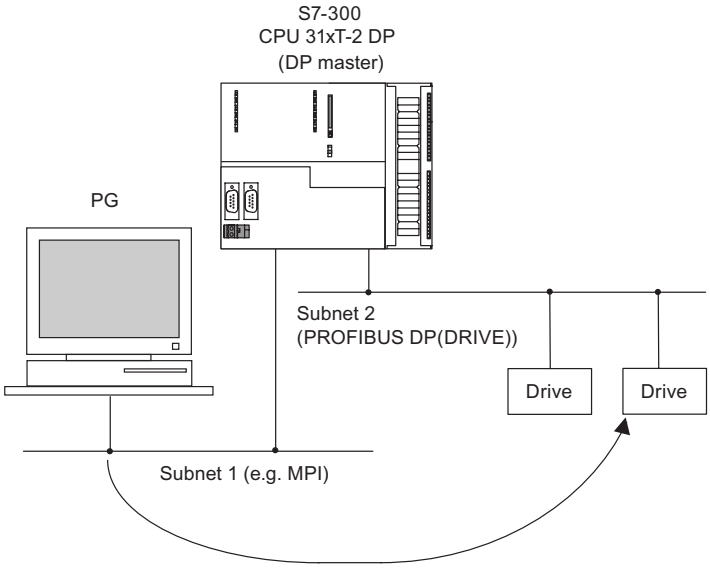


Figure 5-2 Routing - network gateway

Requirements for routing

- The station modules are "capable of routing" (CPUs or CPs).
- The network configuration does not exceed project limits.
- The modules have loaded the configuration data containing the latest "knowledge" of the entire network configuration of the project.

  Reason: All modules participating in the network transition must receive the routing information defining the paths to other subnets.
- In your network configuration, the PG/PC you want to use to establish a connection via network node must be assigned to the network it is physically connected to.

Example of an application: TeleService

The figure below shows the example of an application for remote maintenance of an S7 station using a PG. The connection to other subnets is here established via modem connection.

The lower section of the figure shows how to configure this in *STEP 7*.

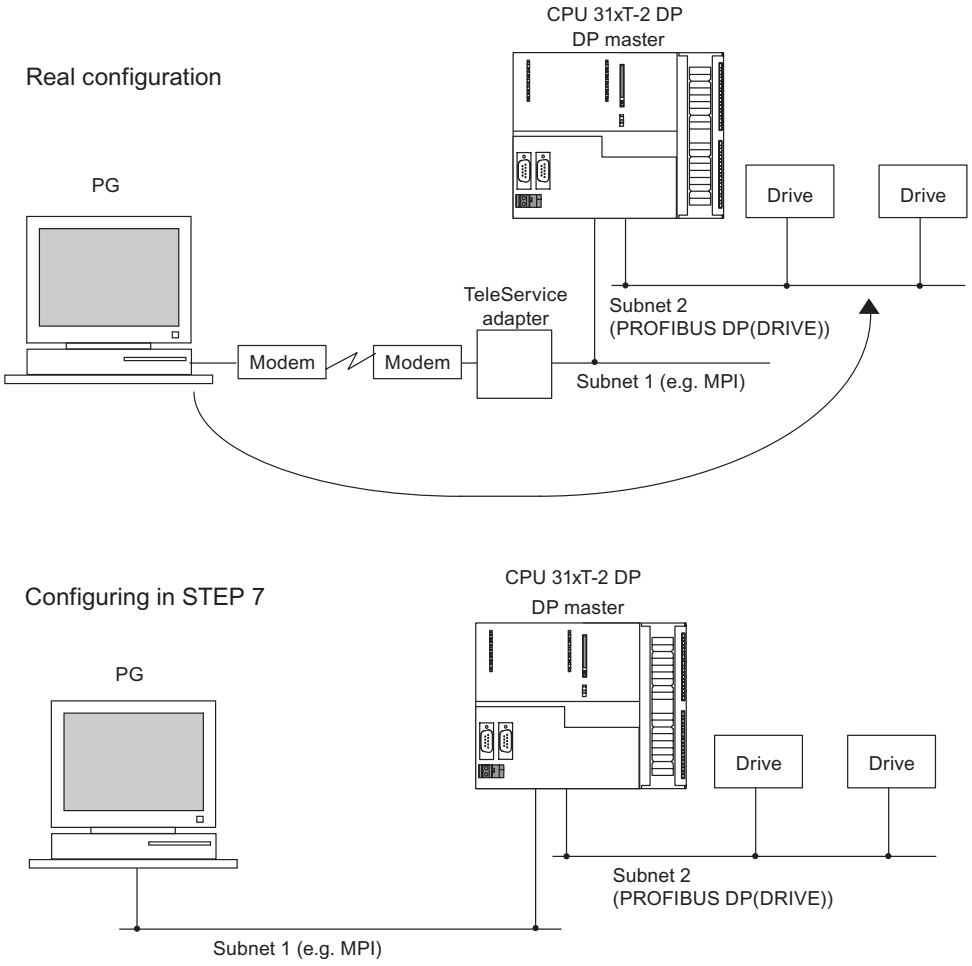


Figure 5-3 Routing - example of TeleService application

Reference

Further information

- about setting the PG/PC interface for routing can be found in Getting Started *CPU 317T-2 DP: Controlling a SINAMICS S120* in chapter *Configuring the PG/PC Interface*.
- about routing can be found in the *Programming with STEP 7* manual, or directly to the *STEP 7* online help.
- about configuring with *STEP 7* can be found in the *Configuring Hardware and Connections in STEP 7* manual
- fundamentals can be found in the *Communication with SIMATIC* Manual.
- about TeleService adapters can be found at Internet website http://www.ad.siemens.de/support with article ID 14053309 (documentation).
- on SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* or to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

5.3.8 Data consistency

Properties

A data area is consistent if it can be read or written to from the operating system as a consistent block. Data exchanged collectively between the stations should belong together and originate from a single processing cycle, that is, be consistent.

If the user program contains a programmed communication function, for example, access to shared data with XSEND/ XRCV, access to that data area can be coordinated by means of the "BUSY" parameter itself.

With PUT/GET functions

For S7 communication functions, such as PUT/GET or write / read via OP communication, which do not require a block in the user program on the CPU (operating in server mode), allowances must be made in the program for the extent of the data consistency.

The PUT/GET functions for S7 communication, or for reading/writing variables via OP communication, are executed at the CPU's scan cycle checkpoint.

In order to ensure a defined process interrupt reaction time, communication variables are copied consistently in blocks of maximum 160 bytes to/from the user memory at the scan cycle check point of the operating system. Data consistency is not guaranteed for larger data areas.

Note

Where defined data consistency is required, the length of communication variables in the CPU's user program may not exceed 160 bytes.

5.4 S7 communication structure

5.4.1 Communication path of an S7 connection

An S7 connection is established when S7 modules communicate with one another. This connection is the communication path.

Note

Global data communication and point-to-point links do not require S7 connections.

Every communication link requires S7 connection resources on the CPU for the entire duration of this connection.

Thus, every S7 CPU provides a specific number of S7 connection resources. These are used by various communication services (PG/OP communication, S7 communication or S7 basic communication).

Connection points

An S7 connection between modules with communication capability is established between connection points. The S7 connection always has two connection points: The active and passive connection points:

- The active connection point is assigned to the module that establishes the S7 connection.
- The passive connection point is assigned to the module that accepts the S7 connection.

Any module that is capable of communication can thus act as an S7 connection point. At the connection point, the established communication link always uses **one** S7 connection of the module concerned.

Transition point

If you use the routing functionality, the S7 connection between two modules capable of communication is established across a number of subnets. These subnets are interconnected via a network transition. The module that implements this network transition is known as a router. The router is thus the point through which an S7 connection passes.

Any CPU with a DP interface can be the router for an S7 connection. You can establish a certain maximum number of routing connections. This does not limit the data volume of the S7 connections.

The number of routing connections can be found in the Technical Specifications.

5.4.2 Assignment of S7 connections

There are several ways to allocate S7 connections on a communication-capable module:

- Reservation during configuration
- Assigning connections in the program
- Allocating connections during commissioning, testing and diagnostics routines
- Allocating connection resources to OCMS services

Reservation during configuration

- If a CPU is inserted when the hardware is configured with *STEP 7*, two S7 connections, one for PG and one for OP communication, are automatically reserved on the CPU.
- It is possible to reserve S7 connections in *STEP 7* for PG / OP communication and S7 basic communication.

Assigning connections in the program

In the case of S7 basic communication, the connection is set up by the user program. The CPU operating system initiates connection setup and the relevant S7 connections are assigned.

Using connections for commissioning, testing and diagnostics

An active online function on the engineering station (PG/PC with *STEP 7*) occupies S7 connections for PG communication:

- An S7 connection resource for PG communication which was reserved in your CPU hardware configuration is assigned to the engineering station, that is, it only needs to be allocated.
- If all reserved S7 connection resources for PG communication are allocated, the operating system assigns a free S7 connection resource which has not yet been reserved. If no more connection resources are available, the engineering station cannot go online to the CPU.

Allocating connection resources to OCMS services

An online function of the OCM station (OP/TD/... with *ProTool*) allocates S7 connection resources for OP communication:

- An S7 connection resource for OP communication you have reserved in your CPU hardware configuration is therefore assigned to the OCM station engineering station, that is, it only needs to be allocated.
- If all reserved S7 connection resources for OP communication are allocated, the operating system assigns a free S7 connection resource which has not yet been reserved. If no more connection resources are available, the OCM station cannot go online to the CPU.

Time sequence for allocation of S7 connection resources

When you program your project in *STEP 7*, the system generates parameter assignment blocks which are read by the modules in the startup phase. This allows the module's operating system to reserve or allocate the relevant S7 connection resources. That is, for instance, OPs cannot access a reserved S7 connection resource for PG communication.

The module's (CPU) S7 connection resources which were not reserved can be used freely. These S7 connection resources are allocated in the order they are requested.

Example

If there is only one free S7 connection left on the CPU, you can still connect a PG to the bus. The PG can then communicate with the CPU. The S7 connection is only used, however, when the PG is communicating with the CPU.

If you connect an OP to the bus while the PG is not communicating, the OP can establish a connection to the CPU. Since an OP maintains its communication link at all times, in contrast to the PG, you cannot subsequently establish another connection via the PG.

5.4.3 Distribution and availability of S7 connection resources

Distribution of the S7 connections

The distribution of the S7 connections of the CPUs can be found in the following table:

Table 5-5 Distribution of the S7 connections

Communication service	Distribution
PG communication OP communication S7 basic communication	In order to avoid allocation of the S7 connections being dependent only on the chronological sequence in which various communication services are requested, S7 connections can be reserved for these services. At least one S7 connection is reserved by default for the PG and OP communication respectively. In the table below, and in the technical data of the CPUs, you can find the configurable S7 connections and the default settings for each CPU. You "redistribute“ the S7 connections by setting the relevant CPU parameters in *STEP 7*.
S7 communication Other communication connections (e.g. via CP 343-1, with a data lengths of > 240 bytes)	Here you allocate the S7 connections which are still available and not reserved for a specific service (PG/OP communication, S7 basic communication).

Communication service	Distribution
Routing PG functions	The CPU provides a certain number of connection resources for routing. These connections are available in addition to the S7 connection resources. The number of routing connections can be found in the Technical Specifications.
Global data communication Point-to-point connection	These communication services do **not** use S7 connections.

Availability of the S7 connections

The following table shows the available S7 connections.

Table 5-6 Availability of the S7 connections for the CPU 315T-2 DP

CPU	Total number of S7 connections	Reserved for			Free S7 connections
		PG communication	OP communication	S7 basic communication	
315T-2 DP	16	1 to 15, default 1	1 to 15, default 1	0 to 12, default 0	Displays all non-reserved S7 connections as free connections.

Example of a CPU 315T-2 DP

The CPU 315T-2 DP provides 16 S7 connections:

- Reserve two S7 connections for PG communication.
- Reserve three S7 connections for OP communication.
- Reserve one S7 connection for S7 basic communication.

This leaves 10 S7 connections available for arbitrary communication services, e.g. S7 communication, OP communication, etc.

Reference

Further information

- on SFCs can be found in the *Instruction list*, for more details refer to the *STEP 7 online help* or to the *System and Standard Functions* reference manual.
- on communication can be found in the *Communication with SIMATIC* manual.

Memory concept

6

6.1 Memory areas and retentive address areas

6.1.1 Technology CPU memory areas

Introduction

The memory of the Technology CPU is divided into three areas:

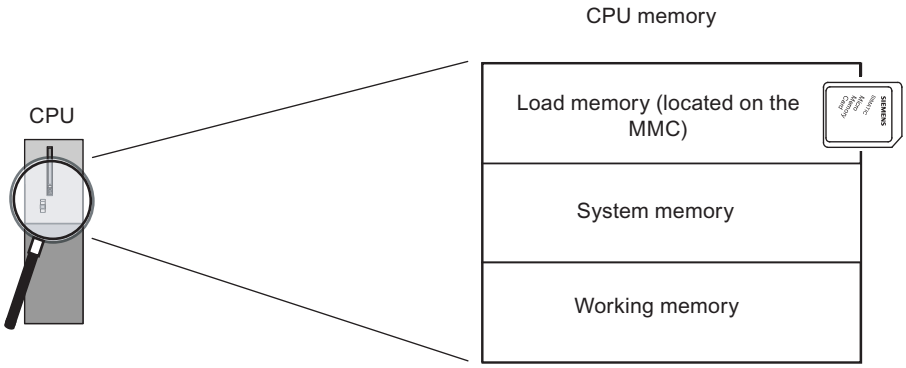


Figure 6-1 Technology CPU memory areas

Load memory

The load memory is located on a micro memory card (MMC). It is used to store code blocks, data blocks and system data (configuration, connections, module parameters, technology system data etc.).

For the Technology CPU, the size of the load memory corresponds to the size of the MMC minus approx. 3 MB. The 3 MB are required for the integrated technology and are therefore not available to the user.

Blocks that are identified as not relevant for the execution are stored exclusively in the load memory.

You can also store all the configuration data for your project on the MMC.

Caution

User programs can only be downloaded and thus the CPU can only be used if the MMC is inserted.

If you pull out the MMC while the CPU is in RUN mode, the CPU then goes into STOP mode and the drives are shut down in accordance with your programming in the *STEP 7* user program.

Therefore, only pull out the MMC while the CPU is in STOP mode.

Working memory

The working memory is integrated in the CPU and cannot be extended. It is used to run the code and process user program data. Programs only run in the working memory and system memory. The working memory is always retentive.

System memory

The system memory is integrated in the CPU and cannot be expanded.

It contains

- the address areas for flags, timers and counters
- the process image of the inputs and outputs
- the local data

6.1.2 Retentive address areas of the load memory, system memory and technology system data

Introduction

Your CPU has retentive memory. The retentive address areas are implemented via the MMC. Data is also retained in the retentive memory throughout POWER OFF and restart (warm restart).

Load memory

Your program in the load memory is always retentive: It is stored on the MMC, where it is protected against power failure or CPU memory reset.

System memory

In your configuration (Properties of CPU, Retentive address areas tab), specify which parts of the tags, timers and counters should be retentive and which of them are to be initialized with "0" on restart (warm restart).

The diagnostics buffer, MPI address (and transmission rate) and operating hour counter data are generally written to the retentive memory area on the CPU. The retentive address areas of the MPI address and transmission rate ensures that your CPU can continue to communicate, even after a power loss, memory reset or loss of communication parameters (e.g. due to removal of the MMC or deletion of communication parameters).

Working memory

The retentive data in the working memory are saved to a non-volatile memory on the CPU during POWER OFF. Therefore, contents of retentive DBs are always retentive.

With the Technology CPU, non-retentive DBs are also supported (the non-retentive DBs are initialized at restart and POWER OFF-ON with their initial values from the load memory).

Technology system data

The technology system data are always stored as retentive data in the load memory of the CPU.

6.1.3 Retentive behavior of the memory objects

Retentive behavior of the memory objects

The table below shows the retentive behavior of memory objects during specific operating state transitions.

Table 6-1 Retentive behavior of the memory objects

Memory object	Operating state transition		
	POWER ON / POWER OFF	STOP → RUN	CPU memory reset
User program/data (load memory)	X	X	X
• Retentive behavior of the DBs (without technology DB)	Can be set in the Properties of the DBs.		-
• Retentive behavior of the technology DBs	-	-	-
Flags, timers and counters configured as retentive data	X	X	-
Diagnostics buffer, operating hour counters	X	X	X
MPI/DP address, transmission rate (or also DP address, transmission rate of the Technology CPU MPI/DPinterface, if this is parameterized as DP node).	X	X	X
Technological parameters • Changed with FB "MC_WriteParameter" • Changed with *S7TConfig*	- X	X X	- X

x = retentive; - = non-retentive

Retentive behavior of a DB

For the Technology CPU, you can specify in *STEP 7* or via SFC 82 "CREA_DBL" (parameter ATTRIB -> NON_RETAIN bit), whether a DB at POWER ON/OFF or RUN-STOP

- retains the actual values (retentive DB), or
- accepts the initial values from load memory (non-retentive DB)

Table 6-2 Retentive behavior of the DBs for the Technology CPU

At POWER ON/OFF or restart of the CPU, the DB should ...	
... receive the initial values (non-retentive DB)	**... retain the last actual values (retentive DB)**
Reason: At POWER ON/OFF and restart (STOP-RUN) of the CPU, the actual values of the DB are non-retentive. The DB receives the initial values from the load memory.	Reason: At POWER OFF/ON and restart (STOP-RUN) of the CPU, the actual values of the DB are retained.
Requirement in STEP 7: • The "Non-retain" check box must be activated in the block properties of the DB, or • a non-retentive DB was generated with SFC 82 "CREA_DBL" and the associated block attribute (ATTRIB -> NON_RETAIN bit).	Requirement in STEP 7: • The "Non-retain" check box must be deactivated in the block properties of the DB, or • a retentive DB was generated with SFC 82.

Retentive behavior of a technology data block

Technology data blocks are non-retentive.

6.1.4 Address areas of system memory

Overview

The system memory of the S7 CPUs is organized in address areas (refer to the table below). In a corresponding operation of your user program, you address data directly in the relevant address area.

Table 6-3 Address areas of system memory

Address areas	Description
Process image of inputs	At every start of an OB1 cycle, the CPU reads the values at the input of the input modules and saves them the process image of inputs.
Process image of outputs	During its cycle, the program calculates the values for the outputs and writes these to the process image of outputs. At the end of the OB1 cycle, the CPU writes the calculated output values to the output modules.

Address areas	Description
Flags	This area provides memory for saving the intermediate results of a program calculation.
Timers	Timers are available in this area.
Counters	Counters are available in this area.
Local data	Temporary data in a code block (OB, FB, FC) is saved to this memory area while the block is being edited.
Data blocks	See *recipes*, *measurement value logs* and *technology data blocks.*

Reference

The address areas of your CPU are listed in the *Instruction list for CPUs 31xC and 31x*.

I/O process image

When the user program addresses the input (I) and output (O) address areas, it does not query the signal states of digital signal modules. Instead, it rather accesses a memory area in CPU system memory. This particular memory area is the process image.

The process image is organized in two sections: The process image of inputs, and the process image of outputs.

Advantages of the process image

Process image access, compared to direct I/O access, offers the advantage that a consistent image of process signals is made available to the CPU during cyclic program execution. When the signal status at an input module changes during program execution, the signal status in the process image is maintained until the image is updated in the next cycle. Moreover, since the process image is stored in CPU system memory, access is significantly faster than direct access to the signal modules.

Process image update

The operating system updates the process image periodically. The figure below shows the sequence of this operation within a cycle.

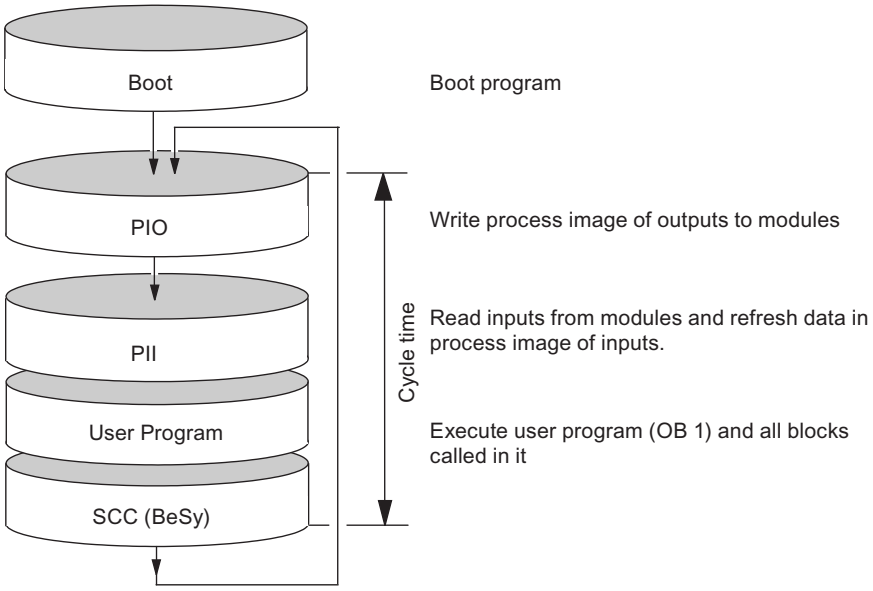


Figure 6-2 Sequence of operation within a cycle

Local data

Local data store:

- the temporary variables of code blocks
- the start information of the OBs
- transfer parameters
- intermediate results

Temporary Variables

When you create blocks, you can declare temporary variables (TEMP) which are only available during block execution and then overwritten again. These local data have fixed length in each OB. Local data must be initialized prior to the first read access. Each OB also requires 20 bytes of local data for its start information. Local data access is faster compared to access to data in DBs.

The CPU is equipped with memory for storing temporary variables (local data) of currently executed blocks. The size of this memory area depends on the CPU. It is distributed in partitions of equal size to the priority classes. Each priority class has its own local data area.

Caution

All temporary variables (TEMP) of an OB and its nested blocks are stored in local data. When using complex nesting levels for block processing, you may cause an overflow in the local data area.
The CPUs will change to STOP mode if you exceed the permissible length of local data for a priority class.
Make allowances for local data space required for synchronous error OBs. This is assigned to the respective triggering priority class.

6.1.5 Properties of the Micro Memory Card (MMC)

Plug-in SIMATIC Micro Memory Cards

Memory modules available:

Table 6-4 Plug-in MMCs

Type	Order number	Comments
MMC 4M	6ES7 953-8LM00-0AA0	-
MMC 8M	6ES7 953-8LP10-0AA0	Required for operating system update

The MMC as memory module for the CPU

The memory module used on your CPU is a SIMATIC Micro Memory Card (MMC.) You can use MMCs as load memory or as a portable storage medium.

Note

The CPU requires the MMC for operation.

Data stored on the MMC:

- User programs (all blocks)
- Recipes and archives
- Configuration data (STEP 7 projects)
- Data for operating system update and backup

Note

You can either store user and configuration data or the operating system on the MMC.

Properties of an MMC

The MMC ensures maintenance-free and retentive operation of these CPUs.

Caution

Data on a SIMATIC Micro Memory Card can be corrupted if you remove the card while it is being accessed by a write operation. In this case, you may have to delete the MMC on your PG, or format the card in the CPU.

Never remove an MMC in RUN mode. Always remove it when power is off, or when the CPU is in STOP state, and when the PG is not a writing to the card. When the CPU is in STOP mode and you cannot not determine whether or not a PG is writing to the card (e.g. load/delete block), disconnect the communication lines.

MMC copy protection

Your MMC has an internal serial number that provides copy protection at user level. You can read this serial number from the SSL partial list $011C_H$ index 8 using SFC 51 "RDSYSST."

You can then program a STOP command, for example, in a copy-protected block if the expected and actual serial numbers of your MCC do not tally.

Reference

Further information

- can be found in the *SSL partial list in the instruction list,* or
- in the *System and Standard Functions* manual.

Useful life of an MMC

The useful life of an MMC is determined in particular by the following factors:

- The number of delete or programming operations,
- external influences such as the ambient temperature.

At ambient temperatures up to 60 °C, a maximum of 100,000 delete/write operations can be performed on an MMC.

Caution

in order to prevent data losses, do not exceed this maximum of delete/write operations.

6.1.6 Saving/retrieving whole projects to/from the Micro Memory Card

Function principle

Using the **Save to Memory Card** and **Retrieve from Memory Card** functions, you can save all project data to a SIMATIC Micro Memory Card, and retrieve these at a later time. For this operation, the SIMATIC Micro Memory Card can be located in a CPU or in the MMC adapter of a PG or PC.

Project data are compressed before they are saved to a SIMATIC Micro Memory Card, and uncompressed when retrieved.

The volume of project data to be saved corresponds with the size of the project's archive file.

Note

In addition to project data, you may also have to store your user data on the MMC. You must therefore make sure beforehand that you select an MMC with sufficient memory capacity.

A message warns you if the memory capacity on your MMC is insufficient.

Even though you can load technology configuration data from the Micro Memory Card, you cannot edit the data.

For technical reasons, you can only transfer the entire contents (user program and project data) using the **Save to memory card** action.

Handling the functions

How you use the **Save to memory card** / **Retrieve from memory card** functions depends on the location of the SIMATIC micro memory card:

- If the micro memory card is inserted in the MMC slot, select a project level that is uniquely assigned to the CPU from the SIMATIC Manager project window (e.g. CPU, program, source or blocks). Select the **PLC > Save to memory card** or **PLC > Retrieve from memory card** menu command. Now the complete project data is written to / retrieved from the Micro Memory Card.
- If project data are not available on the currently used programming device (PG/PC), you can select the source CPU via "Available nodes" window. Select menu command **PLC > Display accessible nodes** to open the "Available nodes" window. Select the connection/CPU that contains your project data on Micro Memory Card. Now select menu command **Retrieve from Memory Card**.
- If the micro memory card is located in the MMC programming device of a PG or PC, open the "S7 memory card window using the **File > S7 Memory Card window > Open** menu command. Select the **PLC > Save to memory card** or **PLC > Retrieve from memory card** menu command. to open a dialog in which you can select the source or target project.

Note

Project data can generate high data traffic. Especially in RUN mode with read/write access to the CPU, this can lead to waiting periods of several minutes.

Sample application

When you assign more than one member of your service and maintenance department to perform maintenance tasks on a SIMATIC PLC, it may prove difficult to provide quick access to current configuration data to each staff member.

However, CPU configuration data available locally on any CPU that is to be serviced can be accessed by any member of the service department. They can edit these data and then release the updated version to all other personnel.

6.2 Memory functions

6.2.1 Downloading the user program

Introduction

Memory functions are used to generate, modify or delete entire user programs or specific blocks. You can also ensure that your project data are retained by archiving these.

General: Download a user program from PG/PC

All user program data are downloaded from your PG/PC to the CPU via MMC.

Blocks use the load memory area as specified under "Load memory requirements" on the "Blocks" tab under "Properties - block container offline".

You cannot run the program until all the blocks are downloaded.

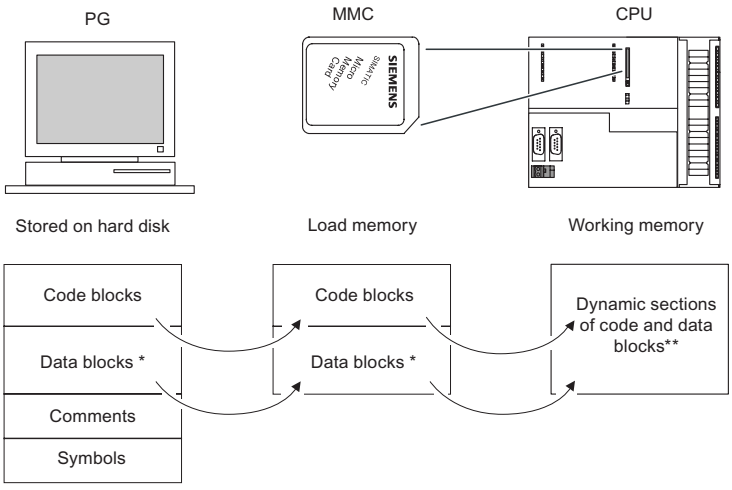


Figure 6-3 Load and working memory of the Technology CPU

* The technology system data are a component of the data blocks.
** If not all the working memory area is retentive, its retentive area is indicated in the STEP 7 module status as retentive memory.

Note

The function "Download a user program from PG/PC" may be used only when the CPU is in the STOP state.

Load memory is cleared if the load operation could not be completed due to power loss or illegal block data.

Before you transfer projects to the Technology CPU via the MPI/DP interface, you should increase the baud rate to at least 1.5 Mbaud or else the data transmission can take a very long time (up to 15 minutes at 187.5 kbaud).

6.2.2 Downloading a user program (enhanced handling)

Download user program from PG/PC to MMC

You can choose one of three methods to download user data to the MMC:

- Download a new user program: You have written a new user program. You download all the program data from PG/PC to the MMC.
- Download of blocks: You already written a user program and downloaded it to the MMC. You then want to add new blocks to the user program. In this case, you do not need to reload the entire user program to the MMC. Rather, you can download only the new blocks to the MMC (this reduces download times for highly complex programs).
- Download an existing user program: In this case, you only download the deltas in the blocks of your user program. In the next step, perform a delta download of the user program, or only of changed blocks to the MMC, using the PG/MC.

Caution

The delta download of blocks / user programs overwrites all data stored under the same name on the MMC.

The data of dynamic blocks are transferred to RAM and activated after the block is downloaded.

Upload

Other than "download" operations, an upload operation is the transfer of specific blocks or a user program **from the CPU to the PG/PC**. The block content is here identical with that of the last download to the MMC. Dynamic DBs form the exception, because their actual values are transferred.

An upload of blocks or of the user program from the CPU in *STEP 7* does not influence CPU memory.

Deleting blocks

When you delete a block, it is deleted from load memory. In *STEP 7*, you can also delete blocks with the user program (DBs also with SFC 23 "DEL_DB").

RAM used by this block is released.

Caution

If you delete a technology data block, the associated drive stops.

Remedy: Switch the CPU to STOP before you delete the technology data block.

Compression

When data are compressed, gaps which have developed between memory objects in load memory/RAM as a result of load/delete operations will be eliminated. This releases free memory in a continuous block.

Data compression is possible when the CPU is in RUN or in STOP.

Promming

When writing the RAM content to ROM, the actual values of the DBs are transferred from RAM to load memory to form the start values for the DBs.

Note

This function is only permitted when the CPU is in STOP mode.

Load memory is cleared if the function could not be completed due to power loss.

6.2.3 CPU memory reset and restart

CPU memory reset

After the insertion/removal of a Micro Memory Card, a CPU memory reset restores defined conditions for CPU restart (warm start).

Technology CPU:

- The CPU's memory management is rebuilt.
- Blocks in load memory are retained.
- All runtime-related blocks are transferred from the load memory to the working memory again.
- The data blocks in the working memory are initialized (are thus reset to their initial values).

Integrated technology of Technology CPU:

The CPU waits until the integrated technology has reached the STOP state.

- The integrated technology is parameterized again.
- The retentive memory of the integrated technology is rebuilt.
- Any distributed I/Os connected to DP(DRIVE) are parameterized again.
- The integrated technology is initialized again.

A description of the CPU memory reset procedure and its special features can be found in chapter *CPU Memory Reset* of the S7-300 installation manual.

Warm Restart

What happens during a warm restart:

- All retentive DBs retain their current values.
- Non-retentive DBs are reset to their initial values.
- The values of all retentive M, C, T are retained.
- All non-retentive user data are initialized:
  - M, C, T, I, O with "0"
- All run levels are initialized.
- The process images are deleted.

6.3 Recipes

Introduction

A recipe represents a collection of user data.

You can implement a simple recipe concept using static DBs. In this case, the recipes should have the same structure (length). One DB should exist per recipe.

Processing sequence

Recipe is written to load memory:

- The various data records of recipes are created as static DBs in *STEP 7* and then downloaded to the CPU. Therefore, recipes only use load memory, rather than RAM.

Working with recipe data:

- SFC83 "READ_DBL" is called in the user program to copy the data record of a current recipe from the DB in load memory to a static DB that is located in work memory. As a result, the RAM only has to accommodate the data of one record.

  The user program can now access data of the current recipe.

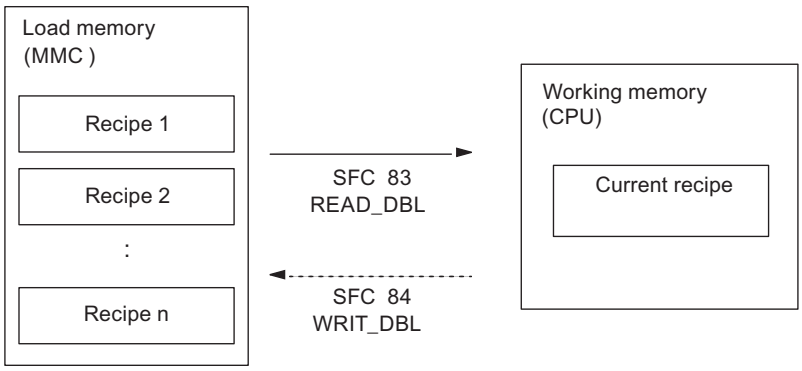


Figure 6-4 Handling recipes

Saving a modified recipe:

- The data of new or modified recipe data records generated during program execution can be written to load memory. To do this, call SFC 84 "WRIT_DBL" in the user program.

  The data written to load memory are portable and retentive on CPU memory reset.

  You can backup modified records (recipes) by uploading and saving these in a single block to the PG/PC.

Note

The active system functions SFC 82 to 84 (current access to the MMC) have a distinct influence on PG functions (block status, variable status, load block, upload, open, for example).

This typically reduces performance (compared to passive system functions) by the factor 10.

in order to prevent data losses, do not exceed this maximum of delete/write operations. Also refer to the SIMATIC Micro Memory Card (MMC) section in the "Structure and Communication Connections of a CPU" chapter.

Caution

Data on a SIMATIC Micro Memory Card can be corrupted if you remove the card while it is being accessed by a write operation. In this case, you may have to delete the MMC on your PG, or format the card in the CPU.

Never remove an MMC in RUN mode. Always remove it when power is off, or when the CPU is in STOP state, and when the PG is not a writing to the card. When the CPU is in STOP mode and you cannot not determine whether or not a PG is writing to the card (e.g. load/delete block), disconnect the communication lines.

6.4 Measured value log files

Introduction

Measured values are generated when the CPU executes the user program. These values are to be logged and analyzed.

Processing sequence

Acquisition of measured values:

The CPU writes all measured values to a DB (for alternating backup mode in several DBs) which is located in RAM.

Measured value logging:

- Before the data volume can exceed work memory capacity, you should call SFC 84 "WRIT_DBL" in the user program to swap measured values from the DB to load memory.

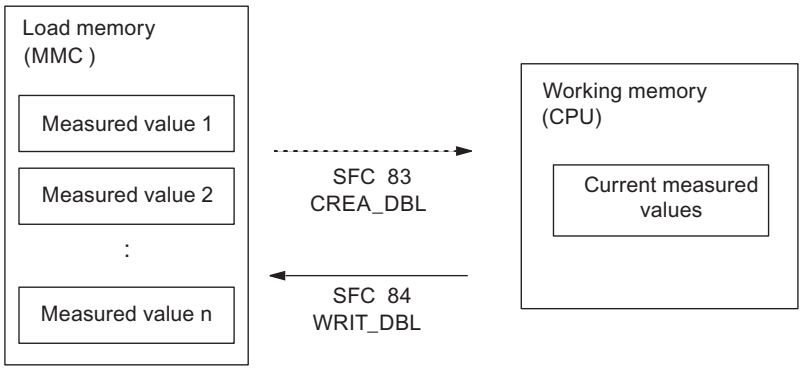


Figure 6-5 Handling measured value log files

- You can call SFC 82 "CREA_DBL" in the user program to generate new (additional) static DBs in load memory which do not require RAM space.

The data written to load memory are portable and retentive on CPU memory reset.

Note

SFC 82 is terminated and an error message is generated if a DB already exists under the same number in load memory and/or RAM.

Evaluation of measured values:

- Measured value DBs saved to load memory can be uploaded and evaluated by other communication partners (PG, PC, for example).

Note

The active system functions SFC 82 to 84 (current access to the MMC) have a distinct influence on PG functions (block status, variable status, load block, upload, open, for example).

This typically reduces performance (compared to passive system functions) by the factor 10.

With the **Technology CPU**, you can also generate non-retentive DBs using SFC 82 (parameter ATTRIB -> NON_RETAIN bit.)

in order to prevent data losses, do not exceed this maximum of delete/write operations. Also refer to the SIMATIC Micro Memory Card (MMC) section in the "Structure and Communication Functions of a CPU 31xC" chapter.

Caution

Data on a SIMATIC Micro Memory Card can be corrupted if you remove the card while it is being accessed by a write operation. In this case, you may have to delete the MMC on your PG, or format the card in the CPU.

Never remove an MMC in RUN mode. Always remove it when power is off, or when the CPU is in STOP state, and when the PG is not a writing to the card. When the CPU is in STOP mode and you cannot not determine whether or not a PG is writing to the card (e.g. load/delete block), disconnect the communication lines.

Reference

For detailed information on SFC 82, refer to the *System Software for S7-300/400, System and Standard Functions* Reference Manual, or directly to the *STEP 7 Online Help*.

6.5 Technology data blocks

Introduction

The integrated technology of the Technology CPU provides current information on the status and on the values of the technology objects via the technology data blocks. To achieve especially short response times, the technology data blocks can be evaluated in OB 65.

Processing sequence

When technology objects are configured, *S7-Technology* creates data blocks in the block folder.

If you initiate jobs to drives using a technology function, you read the states and values in the associated technology data block.

Reference

For further information, refer to the *S7-Technology* manual.

6.6 Memory of the integrated technology of the CPU

Memory utilization

The following table contains typical values for the memory utilization in the integrated technology. The values refer to a CPU 315T-2 DP with firmware version V3.2 of the integrated technology:

	Technology CPU (6ES7 315-6TG10-0AB0) with hardware version 02
Base load of the integrated technology	25 %
Drive axis	1,25 %
Position axis	1,5 %
Following axis (with one following object)	2,5 %
Following axis (with two following objects)	3,5 %
External encoder	0,6 %
Cams	0,25 %
Measuring input	0,25 %
Cam (empty)	0,25 %
Cam interpolation points*	0,0046 %
Cam interpolation points to be interpolated*	0,0027 %
Maximum memory utilization	
Recommended	80 %

* Further explanations can be found in the following calculation example.

Note

With insufficient memory, the Technology CPU goes into STOP mode. Note that these listed values are only typical values and some commands may temporarily require more memory during runtime.
If the memory utilization is excessive, online monitoring from S7T Config may no longer be possible.

Therefore, the recommended calculated maximum memory utilization should not be exceeded.

Calculation example

The table shows the memory utilization for a sample configuration with a CPU 315T-2 DP with hardware version 02. The maximum memory utilization is 77% and is therefore less than the recommended maximum memory utilization.

Quantity	Description	Memory utilization	Memory utilization (total)
1	Base load of the integrated technology	25 %	25 %
6	Following axis (with one following object)	2,5 %	15 %
2	External encoder	0,6 %	1,2 %
6	Output cam	0,25 %	1,5 %
2	Measuring input	0,25 %	0,5 %
14	Cam (empty)	0,25 %	3,5 %
6000*	Cam interpolation points	0,0046 %	27,6 %
1000**	Cam interpolation points to be interpolated	0,0027 %	2,7 %
	Total		**77 %**

* The maximum possible number of cam interpolation points must be taken into account as value in the Technology CPU. Example:

10 cams each with 300 cam interpolation points

2 cams each with 500 cam interpolation points

2 cams each with 1000 cam interpolation points

This is a total of 6000 cam interpolation points (10x300 + 2x500 + 2x1000).

** Additional memory is used during the interpolation of a cam. As only one cam can be interpolated at one time, in this case the cam with the largest number of cam interpolation points must be taken into account (this is 1000 cam interpolation points in the calculation example).

Reference

More detailed information on determining the actual memory assignment in the integrated technology can be found in the *S7-Technology* manual.

Retentive address areas of the memory of the CPU integrated technology

The values for the absolute encoder calibration are stored in a non-volatile memory in the integrated technology of the CPU.

With the technology function "MC_ReadSysParameter", you can read out the absolute encoder calibration values and store them retentively in a data block in the load memory on the MMC. If a CPU is replaced, you can write these stored values back to the integrated technology via the FB "MC_WriteParameter".

Address areas of the memory of the CPU integrated technology

I/O image (DRIVE)

Part of the address areas of the DP (DRIVE) is listed in the integrated technology as I/O image DP (DRIVE). You can read this area in the user program with the technology function "MC_ReadPeriphery" and write with the technology function "MC_WritePeriphery".

The update of the I/O image DP (DRIVE) is described in the *S7-Technology* manual at the technology functions "MC_ReadPeriphery" and "MC_WritePeriphery".

Cycle and response times

7

7.1 Overview

Overview

This section contains detailed information about the following topics:

- Cycle time
- Response time
- Interrupt response time
- Sample calculations

Reference: Cycle time

You can view the cycle time of your user program on the PG.

Reference: Execution time

You can find information in the *S7-300 Instruction List for CPUs 31xC and 31x*. This tabular list contains the execution times for all

- *STEP 7* instructions the relevant CPUs can execute,
- the SFCs / SFBs integrated in the CPUs,
- the IEC functions which can be called in *STEP 7*.

Reference: Motion Control runtimes

You can find information about runtimes on PROFIBUS DP(DRIVE) in the *S7-Technology manual.*

7.2 Cycle time

7.2.1 Overview

Introduction

This chapter explains what we mean by the term "cycle time", what it consists of, and how you can calculate it.

Meaning of the term cycle time

The cycle time represents the time that an operating system needs to execute a program, that is, one OB 1 cycle, including all program sections and system activities interrupting this cycle.

This time is monitored.

Time-Sharing Model

Cyclic program processing, and therefore user program execution, is based on time shares. To clarify these processes, let us assume that every time share has a length of precisely 1 ms.

Process image

During cyclic program processing, the CPU requires a consistent image of the process signals. To ensure this, the process signals are read/written prior to program execution. Subsequently, the CPU does not address input (I) and output (Q) address areas directly at the signal modules, but rather accesses the system memory area containing the I/O process image.

Sequence of cyclic program processing

The table and figure below show the phases in cyclic program processing.

Table 7-1 Cyclic program processing

Step	Operational sequence
1	The operating system initiates cycle time monitoring.
2	The CPU copies the values of the process image of outputs to the output modules.
3	The CPU reads the status at the inputs of the input modules and then updates the process image of inputs.
4	The CPU processes the user program in time shares and executes program instructions.
5	At the end of a cycle, the operating system executes pending tasks, such as the loading and clearing of blocks.
6	The CPU then returns to the start of the cycle, and restarts cycle time monitoring.

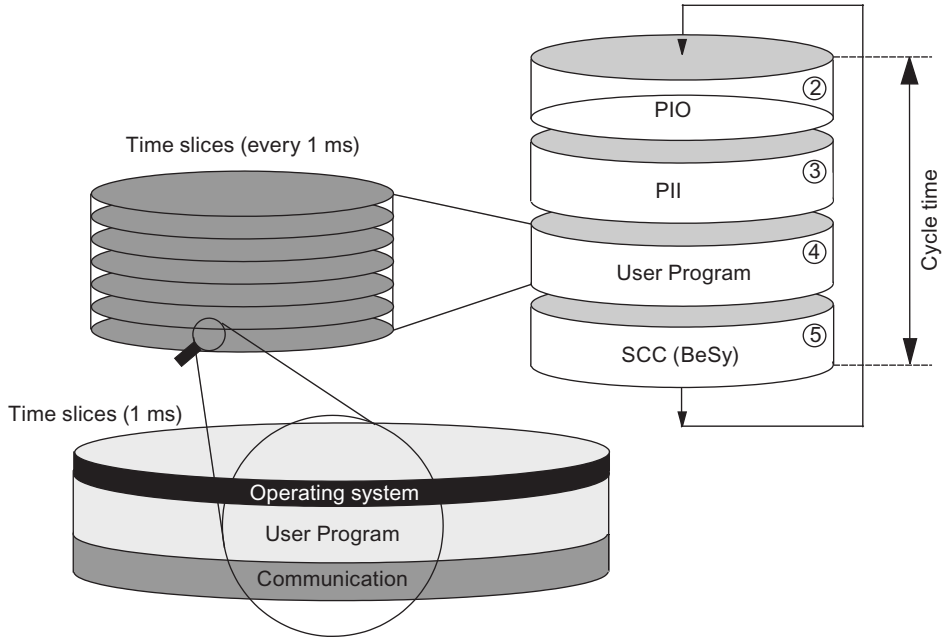


Figure 7-1 Time-Sharing Model

PIO: Process image of outputs
PII: Process image of inputs
SCC: Scan cycle check point
BeSy Operating system

In contrast to S7-400 CPUs (and the CPU 318-2 DP), the S7-300 CPUs data only allow data access from an OP / TD (monitor and modify functions) at the scan cycle check point (Data consistency, see the Technical Specifications). Processing of the user program is not interrupted by the monitor and modify functions.

Increasing the Cycle Time

Always make allowances for the extension of the cycle time of a user program due to:

- Time-based interrupt processing
- Process interrupt processing
- Diagnostics and error handling
- Communication with PGs, Operator Panels (OPs) and connected CPs (for example, Ethernet, PROFIBUS DP)
- Testing and commissioning such as, e.g. status/controlling of variables or block status functions.
- Transfer and deletion of blocks, compressing user program memory
- Write/read access to the MMC, using SFC 82 to 84 in the user program

7.2.2 Calculating the cycle time

Introduction

The cycle time is derived from the sum of the following influencing factors.

Process image update

The table below shows the CPU times for the process image update (process image transfer time). The times specified might be prolonged as a result of interrupts or CPU communication.

The transfer time for the process image update is calculated as follows

Base load K
+ number of bytes in the PI in the rack 0 × (A)
+ number of bytes in the PI via DP × (D)
= transfer time for the process image

Table 7-2 Data for calculating the process image (PI) transfer time

Constant	Components	CPU 315T-2 DP
K	Base load	100 µs
A	Per byte in the rack 0	37 µs
D (DP only)	Per WORD in the DP area for the integrated DP interface	1 µs

Extending the user program processing time

In addition to actually working through the user program, your CPU's operating system also runs a number of processes in parallel, such as timer management for the core operating system. These processes extend the processing time of the user program.

Multiply the processing time of your user program with the **factor 1.10** (CPU 315T-2 DP).

Operating system processing time at the scan cycle checkpoint

The operating system processing time at the scan cycle checkpoint is **500 µs**. This time is valid without:

- Testing and commissioning routines, e.g. status/controlling of variables or block status functions
- Transfer and deletion of blocks, compressing user program memory
- Communication
- Read/write access to the MMC, using SFC82 to 84

Extension of the cycle time due to nested interrupts and due to errors (CPU 315T-2 DP)

Enabled interrupts also extend cycle time. Details are found in the table below.

Table 7-3 Increase in cycle time by nesting interrupts

CPU	Process interrupt	Diagnostic interrupt	Time-of-day interrupt	Delay interrupt	Cyclic interrupt
CPU 315T-2 DP	500 µs	600 µs	400 µs	300 µs	150 µs

You have to add the program execution time at the interrupt level to this increase.

Table 7-4 Cycle time extension as a result of errors

CPU	Programming error	I/O access errors
CPU 315T-2 DP	400 µs	400 µs

You have to add the program execution time of the interrupt OB to this increase.

The times required for multiple nested interrupt/error OBs are added accordingly.

7.2.3 Different cycle times

Overview

The length of the cycle time (T_{cyc}) is not identical in each cycle. The following figure shows different cycle times, T_{cyc1} and T_{cyc2}. T_{cyc2} is longer than T_{cyc1}, because the cyclically scanned OB 1 is interrupted by a time-of-day interrupt OB (here, OB10).

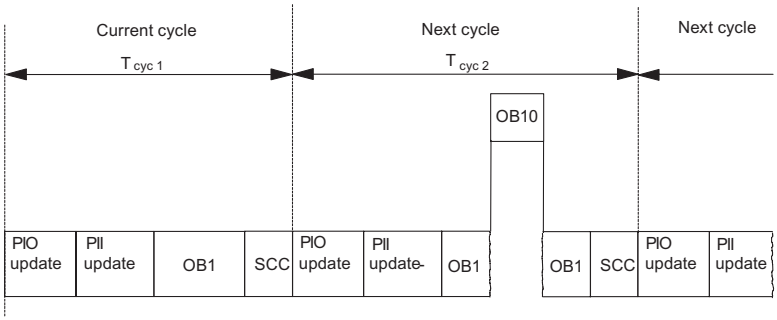


Figure 7-2 Different cycle times

Block processing times may fluctuate

Fluctuation of the block processing time (e.g. OB 1) may also be a factor causing cycle time fluctuation, due to:

- Conditional instructions,
- Conditional block calls,
- Different program paths,
- Loops, etc.

Maximum Cycle Time

In *STEP 7* you can modify the default maximum cycle time. If this time has expired, OB 80 is called, and in it you can define how you want the CPU to respond to the time error.

The CPU switches to STOP mode if OB80 does not exist in its memory.

7.2.4 Communication Load

Configured communication load (PG/OP communication)

The CPU operating system continuously provides a specified percentage of total CPU processing performance (time-sharing technology) for communication tasks. Processing performance not required for communication is made available to other processes.

In the hardware configuration, you can set the load due to communications between 5% and 50%. Default value is 20%.

You can use the following formula for calculating the cycle time extension factor:

100 / (100 - configured communication load in %)

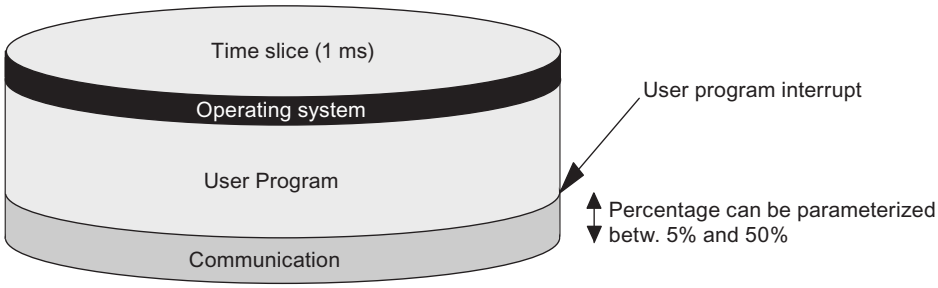


Figure 7-3 Breakdown of a time slice

Example: 20% communication load

You have configured a communication load of 20% in the hardware configuration.

The calculated cycle time is 10 ms.

Using the above formula, the cycle time is extended by the factor 1.25.

Example: 50 % communication load

You have configured a communication load of 50 % in the hardware configuration.

The calculated cycle time is 10 ms.

Using the above formula, the cycle time is extended by the factor 2.

Physical cycle time depending on communication load

The figure below describes the non-linear dependency of the physical cycle time on communication load. As an example, we have chosen a cycle time of 10 ms.

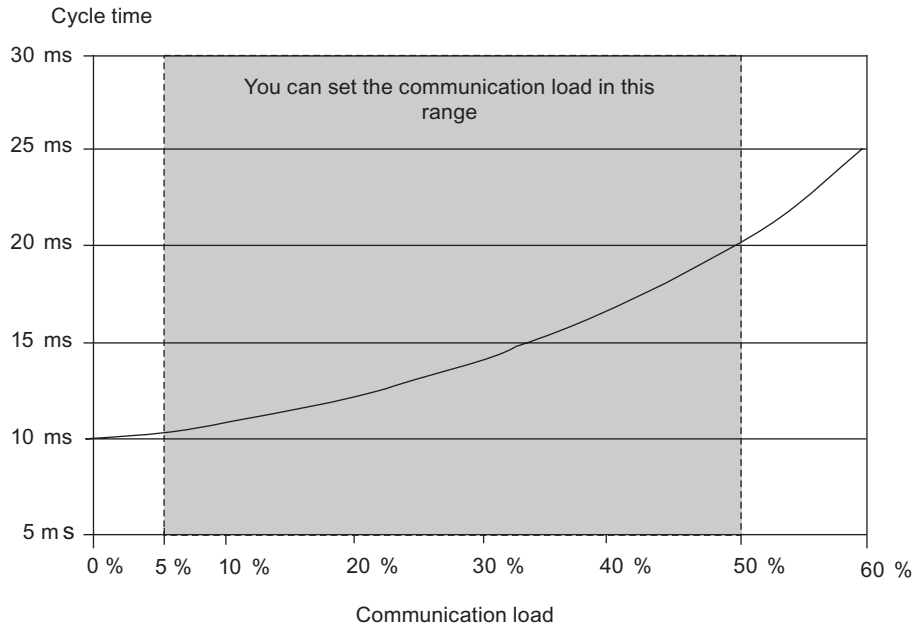


Figure 7-4 Dependency of the Cycle Time on the Communication Load

Influence on the physical cycle time

Due to the increase in the cycle time as a result of the communications component, even more asynchronous events occur, from a statistical point of view, within an OB 1 cycle than, say, interrupts. This further extends the OB1 cycle. This extension depends on the number of events that occur per OB1 cycle and the time required to process these events.

Note

Change the value of the "communication load" parameter to check the effects on the cycle time at system runtime. You must consider the communication load when you set the maximum cycle time, otherwise timing errors may occur.

Tips

- If possible, apply the default value.
- Increase this value only if the CPU is used primarily for communications and if the user program is not time critical.
- In all other situations you should only reduce this value.

7.2.5 Cycle time extension as a result of testing and commissioning functions

Technology CPU runtimes

The runtimes of the testing and commissioning functions are operating system runtimes, so they are the same for every CPU. Initially, there is no difference between process mode and testing mode.

How the cycle time is extended as a result of active testing and commissioning functions is shown in the table below.

Table 7-5 Cycle time extension as a result of testing and commissioning functions

Function	Technology-CPU
Status variable	50 µs for each variable
Control variable	50 µs for each variable
Block status	200 µs for each monitored line

Configuration during parameter assignment

For **Process Operation**, the maximum permissible cycle load by communication is not specified in "Cycle load by communication", but rather in "Maximum permitted increase of cycle time as a result of testing functions during process operation". Thus, the configured time is monitored absolutely in process mode and data acquisition is stopped if a timeout occurs. This is how *STEP 7* stops data requests in loops before a loop ends, for example.

When running in **Testing Operation**, the complete loop is executed in every cycle. This can significantly increase cycle time.

7.3 Response time

7.3.1 Overview

Definition of Response Time

The response time is the time from an input signal being detected to changing an output signal linked to it.

Variation

The actual response time is lies somewhere between a minimum and maximum response time. For configuring your system, you must always reckon with the longest reaction time.

The shortest and longest response times are analyzed below so that you can gain an impression of the variation of the reaction time.

Factors

The reaction time depends on the cycle time and on the following factors:

- Delay of the inputs and outputs of signal modules or integrated I/O.
- Additional DP cycle times on the PROFIBUS DP network
- Execution of the user program

Reference

Further information

- about delay times can be found in the Technical Specifications of the signal modules (*Module data* Reference Manual).
- about delay times for the integrated inputs and outputs can be found in chapter *Technical Specifications of Integrated Technology Inputs and Outputs*

DP Cycle Times on the PROFIBUS-DP Network

If you have configured your PROFIBUS DP network with *STEP 7*, then *STEP 7* will calculate the typical DP cycle time that must be expected. You can then view the DP cycle time of your configuration on the PG.

The figure below gives you an overview of the DP cycle time. We assume in this example that each DP slave has 4 bytes of data on average.

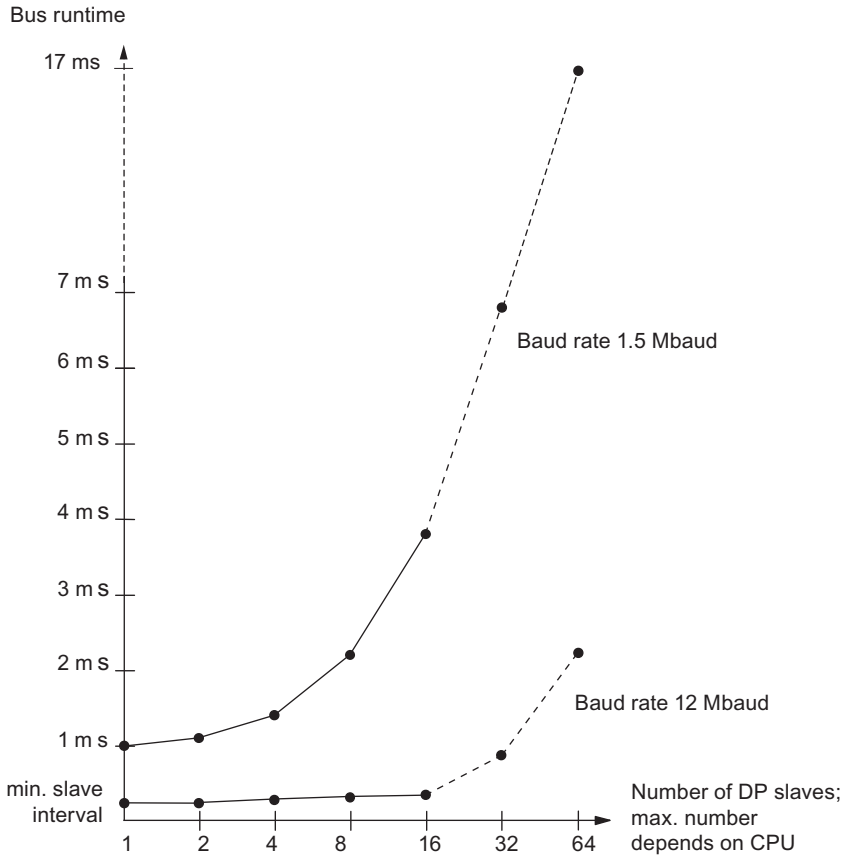


Figure 7-5 DP Cycle Times on the PROFIBUS-DP Network

With multi-master operation on a PROFIBUS-DP network, you must make allowances for the DP cycle time at each master. That is, you will have to calculate the times for each master separately and then add up the results.

7.3.2 Shortest response time

Conditions for the shortest response time

The following figure illustrates the conditions under which the shortest response time can be achieved.

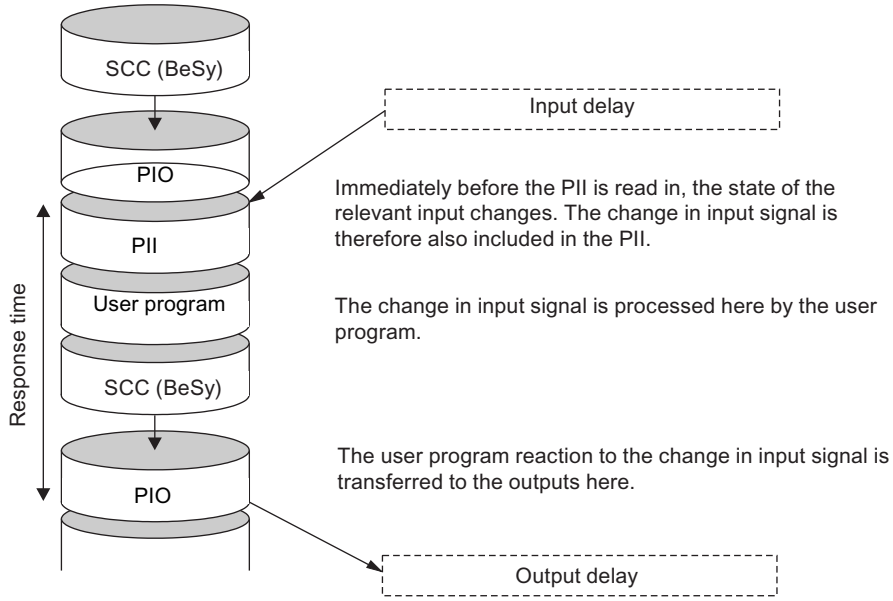


Figure 7-6 Shortest response time

Calculation

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

- 1 x process image transfer time for the inputs +
- 1 x process image transfer time for the outputs +
- 1 x program processing time +
- 1 x operating system processing time at the SCC +
- Delay in the inputs and outputs

The result is equivalent to the sum of the cycle time plus the I/O delay times.

7.3.3 Longest response time

Conditions for the longest response time

The following figure shows you how the longest response time results.

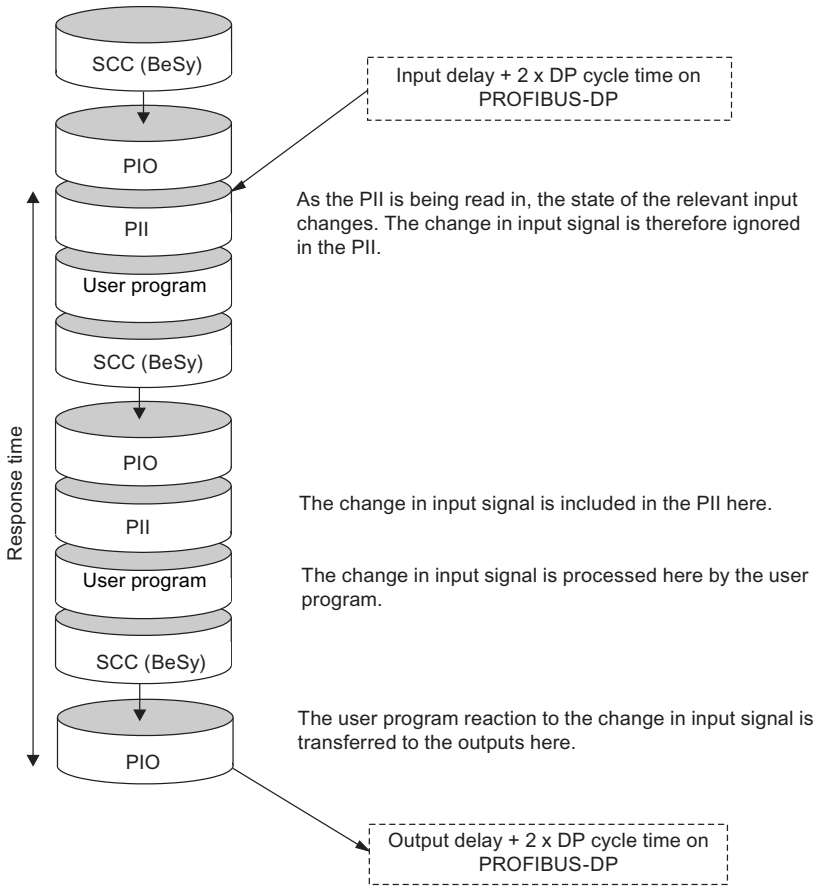


Figure 7-7 Longest response time

Calculation

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

- 2 x process image transfer time for the inputs +
- 2 x process image transfer time for the outputs +
- 2 x program processing time +
- 2 x operating system processing time at the SCC +
- 4 x runtime of the DP slave frame (including processing in the DP master) +
- Delay in the inputs and outputs

This is equivalent to the sum of twice the cycle time and the delay in the inputs and outputs plus four times the DP cycle time.

7.3.4 Reducing the response time with direct I/O access

Reducing the Response Time

You can reach faster response times with direct access to the I/O in your user program, e.g. with

- L PIB or
- T PQW

you can partially avoid the response times described above.

Note

You can also achieve fast response times by using process interrupts (see next chapter).

7.4 Calculating method for calculating the cycle/response time

Introduction

This section gives you an overview of how to calculate the cycle/response time.

Cycle time

1. Determine the user program runtime with the help of the instruction list.
2. Multiply the calculated value by the CPU-specific factor from the table *Extension of user program processing time*.
3. Calculate and add the process image transfer time. Corresponding guide values are found in table *Data for calculating process image transfer time*.
4. Add the processing time at the scan cycle checkpoint. Corresponding guide values are found in the table *Operating system processing time at the scan cycle checkpoint*.
5. Include the extensions as a result of testing and commissioning functions. These values are found in the table *Cycle time extension due to testing and commissioning functions. The result you achieve is the*
6. **cycle time**.

Extension of the cycle time as a result of interrupts and communication load

100 / (100 - configured communication load in %)

1. Multiply the cycle time by the factor as in the formula above.
2. Using the Instruction List, calculate the runtime of the program sections that hardware interrupts. Add the corresponding value from section *Calculating the cycle time*, table *Cycle time extension by interrupt nesting.*

3. Multiply both values by the CPU-specific extension factor of the user program processing time (see table "CPU communication services").
4. Add the value of the interrupt-processing program sequences to the theoretical cycle time, multiplied by the number of triggering (or expected) interrupt events within the cycle time.

   The result you obtain is approximately the **actual cycle time**. Make a note of the result.

Response time

Table 7-6 Calculating the response time

Shortest response time	Longest response time
-	Multiply the actual cycle time by a factor of 2.
Now add I/O delay.	Then, calculate the delays in the inputs and outputs and the DP cycle times on the PROFIBUS DP network.
The result is the shortest response time.	The result is the longest response time.

7.5 Interrupt response time

7.5.1 Overview

Definition of the interrupt response time

The interrupt response time is the time that expires between the first occurrence of an interrupt signal and the call of the first interrupt OB statement. Generally, the following is valid: Interrupts having a higher priority take precedence. This means that the interrupt response time is increased by the program processing time of the higher-priority interrupt OBs and the interrupt OBs of equal priority which have not yet been executed (queued).

Process/diagnostic interrupt response times of the CPU

Table 7-7 Process/diagnostic interrupt response times

CPU	Process interrupt response times			Diagnostic interrupt response times	
	External min.	External max.	Integrated inputs/outputs for integrated technology*	min.	Max.
CPU 315T-2 DP	0.4 ms	0.7 ms	-	0.4 ms	1.0 ms

* The integrated inputs/outputs for integrated technology do not have interrupt capability.

Calculation

The formulae below show how you can calculate the minimum and maximum interrupt response times.

Table 7-8 Process/diagnostic interrupt response times

Calculation of the minimum and maximum interrupt reaction time	
Minimum interrupt reaction time of the CPU + Minimum interrupt reaction time of the signal modules + DP cycle time on the PROFIBUS DP = Quickest interrupt reaction time	Maximum interrupt reaction time of the CPU + Maximum interrupt reaction time of the signal modules +2 x DP cycle time on the PROFIBUS DP The maximum interrupt reaction time is longer when the communication functions are active. The extra time is calculated using the following formula: **tv: 200 µs + 1000 µs × n%** Significant extension possible with n = cycle time extension through communication

Signal modules

The **process interrupt response time** of signal modules is determined by the following factors:

- Digital input modules

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

  You will find the times in the data sheet of the digital input module concerned.

- Analog input modules

  Process interrupt response time = internal interrupt preparation time + conversion time

  The internal interrupt processing time of the analog input modules is negligible. The conversion times can be taken from the data sheet of the analog input module concerned.

The **diagnostic interrupt response time** of signal modules is equivalent to the period that expires between the time a signal module detects a diagnostic event and the time this signal module triggers the diagnostic interrupt. This time is so small that it can be ignored.

Process interrupt processing

When the process interrupt OB 40 is called, the process interrupt is processed. Interrupts with higher priority interrupt process interrupt processing and direct access to the I/O is made when the statement is executed. When process interrupt processing is completed, either cyclic program processing is continued or other interrupt OBs with the same or a lower priority are called and processed.

7.5.2 Reproducibility of Time-Delay and Watchdog Interrupts

Definition of "Reproducibility"

Delay interrupt:

The deviation with time from the first instruction of the interrupt OB being called to the programmed interrupt time.

Watchdog interrupt:

The variation in the time interval between two successive calls, measures between the first instruction of the interrupt OB in each case.

Reproducibility

The following times apply for the CPUs described in this manual:

- Delay interrupt: +/- 200 µs
- Watchdog interrupt: +/- 200 µs

These times only apply if the interrupt can actually be executed at this time and if not interrupted, for example, by higher-priority interrupts or queued interrupts of equal priority.

7.6 Sample calculations

7.6.1 Calculation example for the cycle time of the CPU 315T-2 DP

Structure

You have configured an S7300 and equipped it with following modules in rack "0":

- 1 CPU 315T-2 DP
- 2 digital input modules SM 321; DI 32 x 24 VDC (4 bytes each in the PI)
- 2 digital output modules SM 322; DO 32 x 24 VDC / 0.5 A
  (4 bytes each in the PI)

User program

According to the Instruction List, your user program has a runtime of 5 ms. There is no active communication.

Calculation of the cycle time for the CPU 315T-2 DP

The cycle time for the example results from the following times:

- User program execution time:

  approx. 5 ms x CPU-specific factor 1.10 = approx. 5.5 ms
- Process image transfer time

  Process image of inputs: 100 µs +8 bytes x 37 µs = approx. 0.396 ms

  Process image of outputs: 100 µs +8 bytes x 37 µs = approx. 0.396 ms
- Operating system runtime at scan cycle checkpoint:

  approx. 0.5 ms

Cycle time = 5.5 ms +0.396 ms +0.396 ms + 0.5 ms = 6.792 ms.

Calculation of the actual cycle time

- There is no active communication.
- There is no interrupt handling.

Therefore, the **actual cycle time** is also 6.792 ms.

Calculation of the longest response time

Longest response time:

6.792 ms x 2 = 13.584 ms.

- The delay in the inputs and outputs is negligible.
- All the components are plugged into rack 0; DP cycle times do not therefore have to be taken into account.
- There is no interrupt handling.

7.6.2 Calculation example for the response time of the CPU 315T-2 DP

Structure

You have configured an S7300 and equipped it with the following modules on a rack:

- One CPU 315T-2 DP

  Configuration of the cycle load as a result of communication: 40 %
- 2 digital input modules SM 321; DI 32 x 24 VDC (4 bytes each in the PI)
- 3 digital output modules SM 322; DO 16 x 24 VDC/0.5 A (2 bytes each in the PI)
- 1 analog input module SM 331; AI 8 x 12-bit (not in the PI)
- 1 analog output module SM 332; AO 4 x 12-bit (not in the PI)

User program

According to the Instruction List, the user program has a runtime of 10.0 ms.

Cycle time calculation

The cycle time for the example results from the following times:

- User program execution time:

  approx. 10 ms x CPU-specific factor 1.10 = approx. 11.0 ms
- Process image transfer time

  Process image of inputs: 100 µs +8 bytes x 37 µs = approx. 0.396 ms

  Process image of outputs: 100 µs + 6 bytes x 37 µs = approx. 0.322 ms
- Operating system runtime at scan cycle checkpoint:

  approx. 0.5 ms

The cycle time for the example results from the sum of the times listed:

Cycle time = 11.0 ms +0.396 ms +0.322 ms + 0.5 ms = 12.218 ms.

Calculation of the actual cycle time

Allowance for communication load:

12.218 ms * 100 / (100-40) = 20.36 ms.

Thus, under consideration of time slots, the **actual cycle time** is approx. **20 ms**.

Calculation of the longest response time

- Longest response time = 20 ms * 2 = 40 ms.
- Delays in the inputs and outputs
  - The maximum input delay of the digital input module SM 321; DI 32 x 24 VDC is **4.8 ms** per channel.
  - The output delay of the digital output module SM 322; DO 16 x 24 VDC / 0.5 A **can be neglected**.
  - The analog input module SM 331; AI 8 x 12 bit was configured for an interference suppression at 50 Hz. This results in a conversion time of 22 ms per channel. Since 8 channels are active, a cycle time of **176 ms** results for the analog input module.
  - The analog output module SM 332; AO 4 x 12-bit was programmed for the measuring range of 0 ...10 Hz. This gives a conversion time of 0.8 ms per channel. Since 4 channels are active, a cycle time of 3.2 ms results. The settling time for the resistive load of 0.1 ms must still be added. The result is a response time of **3.3 ms** for an analog output.
- All the components are plugged into the central rack; DP cycle times do not therefore have to be taken into account.

- Response times plus I/O delay:
  - **Case 1:** An output channel of the digital output module is set when a signal is received at the digital input. This produces a response time of:

    Response time = 40 ms + 4.8 ms = 44.8 ms.
  - **Case 2:** An analog value is fetched, and an analog value is output. This produces a response time of:

    Longest response time = 40 ms + 176 ms + 3.3 ms = 219.3 ms.

7.6.3 Calculation example for the interrupt response time of the CPU 315T-2 DP

Structure

You have assembled an S7-300, consisting of one CPU 315T-2 DP and four digital modules in the central rack. One of the digital input modules is an SM 321; DI 16 x 24 VDC; with process/diagnostic interrupt function.

In the parameter assignment of the CPU and the SM, you have only enabled the process interrupt. You do not require time-driven processing, diagnostics and error handling. You have set a cycle load caused by communication of 20%.

You have set an input delay of 0.5 ms for the digital input module.

No activities are necessary at the cycle checkpoint.

Calculation

The process interrupt response time for the example results from the following times:

- Process interrupt response time of CPU 315T-2 DP: approx. 0.7 ms
- Extension by communication according to the formula:

  200 µs + 1000 µs x 20 % = 400 µs = 0.4 ms
- Process interrupt response time of SM 321; DI 16 x 24 VDC:
  - Internal interrupt preparation time: 0.25 ms
  - Input delay: 0.5 ms
- Since the signal modules are plugged into the central rack, the DP cycle time on the PROFIBUS DP is not relevant.

The process interrupt response time results from the sum of the listed times:

Process interrupt response time = 0.7 ms + 0.4 ms + 0.25 ms + 0.5 ms = **approx. 1.85 ms**.

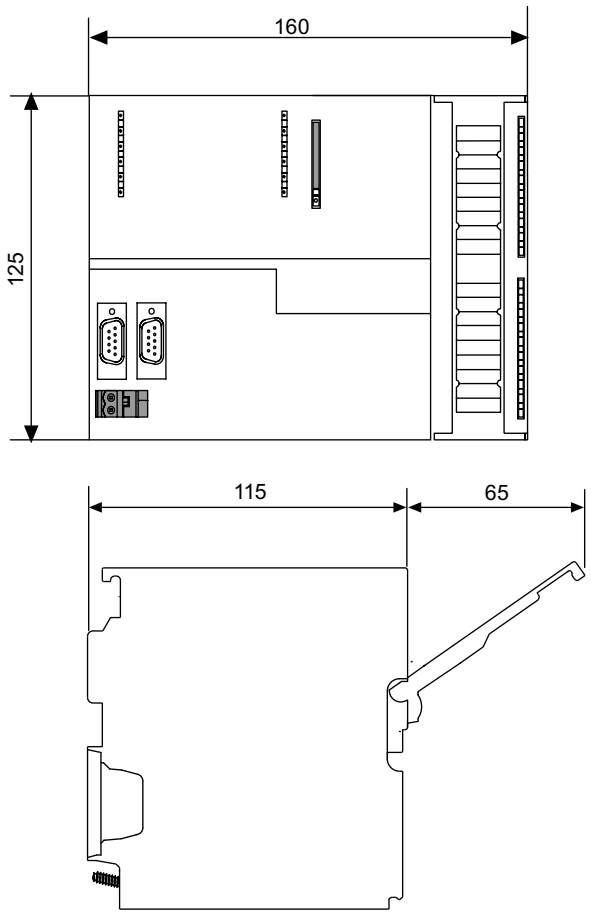
This calculated process interrupt response time is the time from a signal being applied across the digital input to the first statement in OB 40.

Technical data

8

8.1 General technical data

8.1.1 Dimension drawing



Dimension drawing of Technology CPU

8.1.2 Technical specifications of the Micro Memory Card (MMC)

Plug-in SIMATIC Micro Memory Cards

Memory modules available:

Table 8-1 Available MMCs

Type	Order number	Comments
MMC 4M	6ES7 953-8LM00-0AA0	-
MMC 8M	6ES7 953-8LP10-0AA0	Required for an operating system update

8.1.3 Clock

Features and functions

The following table contains the features and functions of the clock.

Table 8-2 Features and functions of the clock

Features	Technology-CPU
Type	Hardware clock
Factory setting	DT#1994-01-01-00:00:00
Buffering	By means of integrated capacitor
Buffered period	Typically 6 weeks (at an ambient temperature of 40 °C)
Behavior of the realtime clock after POWER ON	The clock continues running after POWER OFF.
Behavior of the clock on expiration of the buffered period	The clock keeps running, continuing at the time-of-day it had when power was switched off.

Information on..

- Synchronization and correction factor:

  The CPU parameterization functions in *STEP 7* enable you to set functions such as synchronization via MPI interface and correction factor. For further information, please refer to the *STEP 7 online help*.

- Set, read and program the clock:

  You can set and read the clock with the PG (see *Programming with STEP 7* manual). You can also program the clock in the user program with SFCs (see reference manual *System and Standard Functions*).

8.2 CPU 315T-2 DP

Technical data

Table 8-3 Technical data for the CPU 315T-2 DP

Technical data	
CPU and version	
Order no. [MLFB]	6ES7 315-6TG10-0AB0
• Hardware version	02
• Firmware version (CPU)	V 2.4
• Firmware version (integrated technology)	V 3.2
• Associated programming package	STEP 7 as of V 5.3 + SP 3 and option package *S7-Technology V3.0*
Technology objects	
• Total	32 (axes, cams, output cams, measuring inputs, external encoders)
• Axes	8 axes (virtual or real axes)
• Output cam	16 output cams 8 output cams can be output as "high-speed output cams" on the integrated outputs of the Technology CPU. A further 8 output cams can be implemented via the distributed I/Os (e.g. on the ET 200M or ET 200S). These can be implemented as "high-speed output cams" on the TM15 and TM17 High Feature.
• Cams	16 cams
• Measuring input	8 measuring inputs
• External encoders	8 external encoders
Memory	
Working memory	
• Integrated	128 KB
• Expandable	No
Capacity of the retentive memory for retentive data blocks	Max. 128 KB
Load memory	Plugged in with MMC (max. 8 MB)
Buffering	Buffering guaranteed by MMC (maintenance-free)
Data storage life on the MMC (following final programming)	At least 10 years
Execution times	
Processing times of	
• Bit operations	Typ. 0.1 µs
• Word instructions	Typ. 0.2 µs
• Integer maths	Typ. 2.0 µs
• Floating-point maths	Typ. 3.0 µs

Technical data	
Timers/counters and their retentive address areas	
S7 counters	256
• Retentive address areas	Configurable
• Default	From C0 to C7
• Counting range	0 to 999
IEC Counters	Yes
• Type	SFB
• Quantity	Unlimited (limited only working memory)
S7 timers	256
• Retentive address areas	Configurable
• Default	Not retentive
• Timer range	10 ms to 9990 s
IEC timers	Yes
• Type	SFB
• Quantity	Unlimited (limited only by working memory)
Data areas and their retentive address areas	
Flag	2048 bytes
• Retentive address areas	Configurable
• Preset retentive address areas	MB0 to MB15
Clock flags	8 (1 flag byte)
Data blocks	
• Quantity	1023 (DB 1 to DB 1023)
• Size	16 KB
• Non-retain support (configurable retentive address areas)	Yes
Local data per priority class	Max. 1024 bytes
Blocks	
Total	1024 (DBs, FCs, FBs) The maximum number of blocks that can be loaded may be reduced if you are using another MMC.
OBs	See the Instruction List
• Size	16 KB
Nesting depth	
• Per priority class	8
• Additional within an error OB	4
FBs	See the Instruction List
• Quantity	2048 (FB 0 to FB 2047)
• Size	16 KB

Technical data	
Technological functions	
• Maximum number of simultaneously active jobs	210
• Maximum number of simultaneously assigned job data compartments	100 The following technology functions each occupy (as long as they are active) one job data compartment: • "MC_ReadPeriphery" • "MC_WritePeriphery" • "MC_ReadRecord" • "MC_WriteRecord" • "MC_ReadDriveParameter" • "MC_WriteDriveParameter" • "MC_CamSectorAdd"
FCs	See the Instruction List
• Quantity	2048 (FC 0 to FC 2047)
• Size	16 KB
Address areas (I/O)	
Total I/O address area	Max. 2048 bytes / 2048 bytes (can be freely addressed)
Distributed	Max. 2048 bytes
I/O process image	128 bytes / 128 bytes
Digital channels	16348/16348
Of those central	Max. 256
Analog channels	1024/1024
Of those central	64 / 64
Address areas (I/O) of the integrated technology	
Total I/O address area	Max. 1024 bytes / 1024 bytes (can be freely addressed)
I/O image DP (DRIVE)	64/64
Removal	
Racks	1
Modules per rack	8
Number of DP masters	
• Integrated	1
• Via CP	2
Operable function modules and communication processors	
• FM	Max. 8
• CP (PtP)	Max. 8
• CP (LAN)	Max. 10
Time of day	
Clock	Yes (hardware clock)
• Buffered	Yes

Technical data	
• Buffered period	Typically 6 weeks (at an ambient temperature of 104 °F)
• Accuracy	Deviation per day: < 10 s
Operating hours counter	1
• Number	0
• Value range	2 31 hours (if SFC 101 is used)
• Granularity	1 hour
• Retentive	Yes; must be manually restarted after every restart
Time synchronization	Yes
• In the PLC	Master/slave
• On MPI	Master/slave
S7 message functions	
Number of stations that can be logged on for signaling functions	16 (depends on the number of connections configured for PG / OP and S7 basic communication)
Process diagnostics messages	Yes
• Simultaneously enabled interrupt S blocks	40
Test and startup functions	
Status/control variables	Yes
• Variables	Inputs, outputs, flags, DBs, timers, counters
• Number of variables	30
Of those as status variable	Max. 30
Of those as control variable	Max. 14
Force	
• Variables	Inputs/outputs
• Number of variables	Max. 10
Block status	Yes
Single-step	Yes
Breakpoints	2
Diagnostic buffer	Yes
• Number of entries (not configurable)	Max. 100
Communication functions	
PG/OP communication	Yes
Global data communication	Yes
• Number of GD circuits	8
• Number of GD packets	Max. 8
Transmitters	Max. 8
Receivers	Max. 8
• Size of GD packets	Max. 22 bytes
Consistent data	22 bytes
S7 basic communication	Yes

Technical data	
• User data per job	Max. 76 bytes
Consistent data	76 bytes (for X_SEND or X_RCV) 76 bytes (for X_PUT or X_GET as the server)
S7 communication	Yes
• As server	Yes
• As client	Yes (via CP and loadable FBs)
• User data per job	Max. 180 bytes (with PUT/GET)
Consistent data	64 byte (as the server)
S5compatible communication	Yes (via CP and loadable FCs)
Number of connections	16
Can be used for	
• PG communication	
Reserved (default)	1
Configurable	1 to 15
• OP communication	
Reserved (default)	1
Configurable	1 to 15
• S7-based communication	Yes
Reserved (default)	12
Configurable	0 to 12
Routing	Yes (depending on the version of the firmware product (CPU)): Version 2.3 and earlier: 4 Version 2.4 and later: 8
Interfaces	
1st interface (X1)	
Type of interface	Integrated RS485 interface
Physics	RS 485
Isolated	Yes
Interface power supply (15 to 30 VDC)	Max. 200 mA
• **Functionality**	
• MPI	Yes
• PROFIBUS DP	Yes
• PROFIBUS DP (DRIVE)	No
• Point-to-point connection	No
MPI	
Services	
• PG/OP communication	Yes
• Routing	Yes
• Global data communication	Yes
• S7 basic communication	Yes
• S7 communication	Yes

Technical data	
As server	Yes
As client	Yes (via CP and loadable FBs)
• Transmission rates	Max. 12 Mbps
DP master	
Services	
• PG/OP communication	Yes
• Routing	Yes
• Global data communication	No
• S7 basic communication	No
• S7 communication	No
• Constant bus cycle time	Yes
• SYNC/FREEZE	Yes
• DPV1	Yes
Transmission speed	Up to 12 Mbps
Number of DP slaves	124
Address area per DP slave	Max. 244 bytes
DP slave	
Services	
• Routing	Yes
• Global data communication	No
• S7 basic communication	No
• S7 communication	No
• Direct data exchange	Yes
• Transmission rates	Up to 12 Mbps
• Automatic baud rate search	No
• Intermediate memory	244 bytes I / 244 bytes O
• Address areas	Max. 32 with max. 32 bytes each
• DPV1	No
2nd interface (X3)	
Type of interface	Integrated RS485 interface
Physics	RS 485
Isolated	Yes
Type of interface	Integrated RS485 interface
Interface power supply (15 to 30 VDC)	Max. 200 mA
Functionality	
MPI	No
PROFIBUS DP	No
PROFIBUS DP (DRIVE)	Yes
Point-to-point connection	No
DP (DRIVE) master	
Services	
• PG/OP communication	No

Technical data	
• Routing	No
• Global data communication	No
• S7 basic communication	No
• S7 communication	No
• Constant bus cycle time	Yes
• SYNC/FREEZE	No
• DPV1	No
Transmission speed	Up to 12 Mbps
Number of DP slaves	32
Address area per station	Max. 244 bytes
DP slave	**No**
Programming	
Programming language	LAD/FBD/STL
Instruction set	See the Instruction List
Nesting levels	8
System functions (SFC)	See the Instruction List
System function blocks (SFB)	See the Instruction List
User program protection	Yes
Dimensions	
Mounting dimensions W × H × D (mm)	160 × 125 × 130
Weight	750 g
Voltages, currents	
Power supply (rated value)	24 VDC
• Permissible range	20.4 V to 28.8 V
Current consumption (no-load operation)	Typ. 200 mA
Making current	Typically 2.5 A
I2t	1 A2s
External fusing of power supply lines (recommended)	Min. 2 A
Power loss	Typically 6 W

8.3 Integrated Inputs/Outputs for Technology

8.3.1 Arrangement of integrated inputs/outputs for integrated technology

Introduction

The Technology CPU has 4 digital inputs and 8 digital outputs integrated. You use these inputs and outputs for technology functions, e.g. reference point acquisition (reference cams) or high-speed output cam switching signals.

The digital inputs and outputs can also be used in the STEP 7 user program.

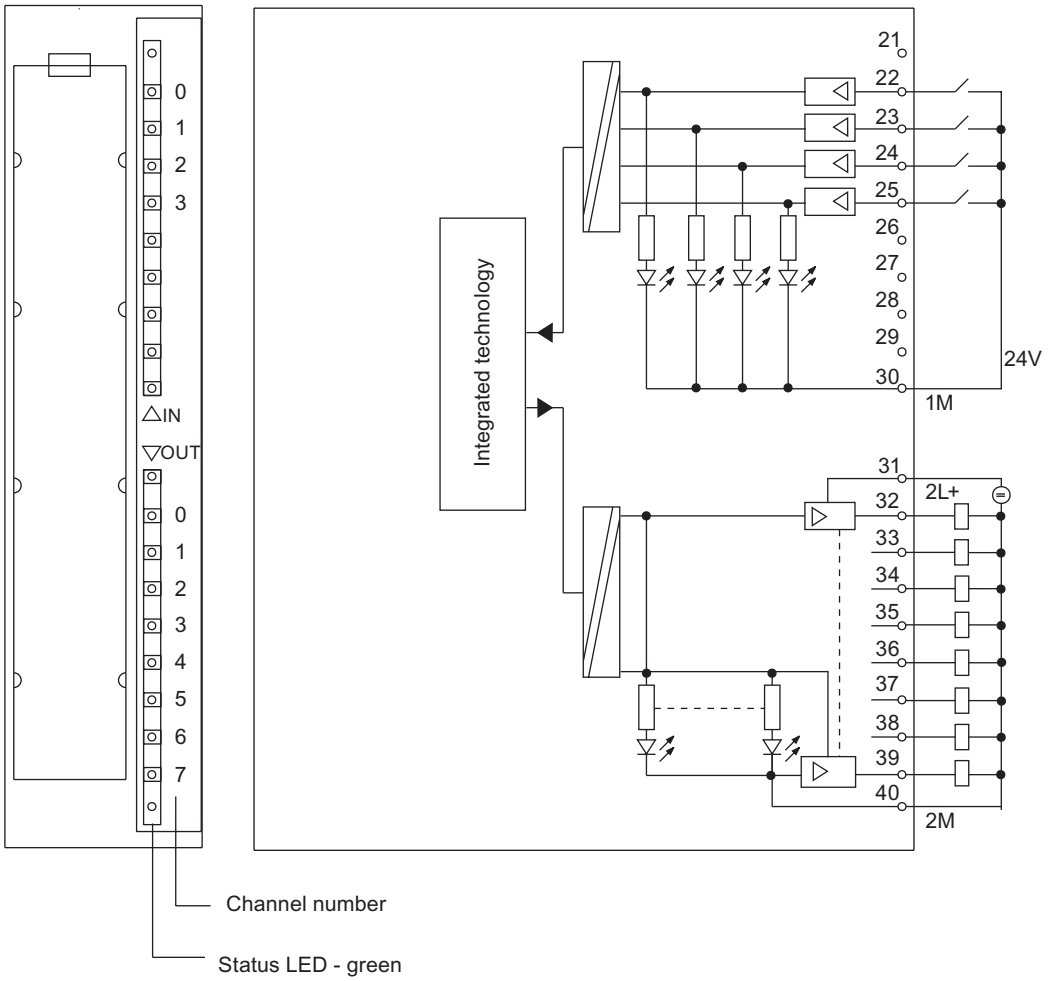


Figure 8-1 Block diagram of integrated inputs/outputs for integrated technology

8.3.2 Technical specifications of digital inputs

Technical specifications

The digital inputs are provided for technology functions such as reference point acquisition (reference cam). They can also be utilized for the STEP7 user program with FB "MC_ReadPeriphery".

Table 8-4 Technical specifications of the integrated inputs for integrated technology

Technical specifications	
Data for specific modules	**Digital inputs**
Number of inputs	4
• Number of these inputs which can be used for technological functions	4
Length of cable	
• Unshielded	600 m
• Shielded	1000 m
Voltage, currents, potentials	
Rated load voltage L+	24 V DC
• Reverse battery protection	No
Number of inputs that can be triggered simultaneously	
• Horizontal assembly	
Up to 40° C	4
up to 60° C	4
• Vertical assembly	
Up to 40° C	4
Galvanic isolation	
• Between channels and backplane bus	Yes
Permissible potential difference	
• Between the different circuits	75 V DC / 60 V AC
Insulation test voltage	500 V DC
Current consumption	
• From the load voltage L+ (no load)	0 mA
Status, interrupts, diagnostics	
Status display	green LED per channel
Interrupts	No
Diagnostic functions	No
Data for selecting a sensor	
Input voltage	
• Nominal value	24 V DC
• For signal "1"	15 V to 30 V
• For signal "0"	-3 V to 5 V
Input current	

Technical specifications	
• with signal "1"	Typ. 7 mA
Input delay	
• at "0" to "1"	Typ. 10 µs
• with "1" to "0"	Typ. 10 µs
Input characteristic curve	To IEC 1131, Type 1
Connection of 2-wire reference cam	No

8.3.3 Technical specifications of digital outputs

Technical specifications

The digital outputs are provided for high-speed camming functions. They can also be utilized for the STEP 7 user program with FB "MC_WritePeriphery".

Table 8-5 Technical specifications of the integrated outputs for integrated technology

Technical specifications	
Data for specific modules	**Digital outputs**
Number of outputs	8
Length of cable	
• Unshielded	max. 600 m
• Shielded	max. 1000 m
Voltage, currents, potentials	
Rated load voltage L+	24 V DC
• Reverse battery protection	No
Cumulative current of outputs (per group)	
• Horizontal assembly	
Up to 40° C	max. 4,0 A
up to 60° C	max. 3,0 A
• Vertical assembly	
Up to 40° C	max. 3,0 A
Galvanic isolation	
• Between channels and backplane bus	Yes
Permissible potential difference	
• Between the different circuits	75 V DC / 60 V AC
Insulation test voltage	500 V DC
Current consumption	
• From the load voltage L+ (no load)	max. 100 mA
Status, interrupts, diagnostics	
Status display	green LED per channel
Interrupts	No

Technical specifications	
Diagnostic functions	No
Data for the selection of an actuator for standard DI	
Output voltage	
• with signal "0"	max. 3 V
• with signal "1"	min. (2 L+) - 2.5 V
Output current	
• with signal "1"	
Nominal value	0,5 A
Permissible range	5 mA to 0,6 A
• With signal "0" (leakage current)	max. 0,3 mA
Load resistor range	48 Ω to 4 kΩ
Lamp load	max. 5 W
Connecting two outputs in parallel	
• For redundant triggering of a load	not supported
• To increase performance	not supported
Triggering a digital input	not supported
Switch rate	
• For resistive load	Max. 100 Hz
• For inductive load to IEC 947-5, DC13	Max. 0,2 Hz
• For lamp load	Max. 100 Hz
Limit (internal) of the inductive circuit interruption voltage to	typ. (2 L+) - 48 V
Short-circuit protection of the output	Yes, electronic
• Threshold on	typ. 1 A
Fast output cam	
• Switching accuracy	+/- 70 µs

Information for the Changeover to the Technology CPU 9

9.1 Scope

Who should read this chapter?

Are you already using a CPU from the SIMATIC S7-300 series and now want to upgrade to a Technology CPU?

Please note that problems may occur when you download your user program to the "new" CPU.

If you have used one of the following CPUs in the past ...

Table 9-1 CPUs previously used

CPU	Order number	As of version	
		Firmware	Hardware
CPU 312 IFM	6ES7 312-5AC02-0AB0 6ES7 312-5AC82-0AB0	1.0.0	01
CPU 313	6ES7 313-1AD03-0AB0	1.0.0	01
CPU 314	6ES7 314-1AE04-0AB0 6ES7 314-1AE84-0AB0	1.0.0	01
CPU 314 IFM	6ES7 314-5AE03-0AB0	1.0.0	01
CPU 314 IFM	6ES7 314-5AE83-0AB0	1.0.0	01
CPU 315	6ES7 315-1AF03-0AB0	1.0.0	01
CPU 315-2 DP	6ES7 315-2AF03-0AB0 6ES7 315-2AF83-0AB0	1.0.0	01
CPU 316-2 DP	6ES7 316-2AG00-0AB0	1.0.0	01
CPU 318-2 DP	6ES7 318-2AJ00-0AB0	V3.0.0	03

... then please read the following information regarding migration to the Technology CPU

9.2 Changed behavior of certain SFCs

SFC 13, SFC 56 and SFC 57 which work asynchronously...

Some of the SFCs that work asynchronously, when used on CPUs 312 IFM to 318-2 DP, were always, or under certain conditions, processed after the first call ("quasi-synchronous").

On the Technology CPU, these SFCs actually run asynchronously. Asynchronous processing may cover multiple OB1 cycles. As a result, a wait loop may turn into an endless loop within an OB.

The following are affected:

- SFC 13 "DPNRM_DG"

  On CPUs 312 IFM to 318-2 DP, this SFC always works "quasi synchronously" when it is called in OB 82. On the Technology CPU, it generally works asynchronously.

Note

In the user program, the job should merely be started in OB 82. The data should be evaluated in the cyclical program, taking account of the BUSY bits and the value returned in RET_VAL.

Tip

If you are using a Technology CPU, we recommend that you use SFB 54, rather than SFC 13 "DPNRM_DG".

- SFC 56 "WR_DPARM"; SFC 57 "PARM_MOD"

  On CPUs 312 IFM to 318-2 DP, these SFCs always work "quasi-synchronously" during communication with centralized I/O modules and always work synchronously during communication with distributed I/O modules.

Note

If you are using SFC 56 "WR_DPARM" or SFC 57 "PARM_MOD", you should always evaluate the SFCs' BUSY bit.

SFC 20 "BLKMOV"

In the past, this SFC could be used with CPUs 312 IFM to 318-2 DP to copy data from a non runtime-related DB.

SFC 20 no longer has this functionality with the **Technology CPU** SFC 83 "READ_DBL" must now be used instead.

SFC 54 "RD_DPARM"

This SFC is no longer available on the **Technology CPU**. Use SFC 102 "RD_DPARA" instead, which works asynchronously.

SFCs that may return other results

You can ignore the following points if you only use logical addressing in your user program.

When using address conversion in your user program (SFC 5 "GADR_LGC", SFC 49 "LGC_GADR"), you must check the assignment of the slot and logical start address for your DP slaves.

- In the past, the diagnostic address of a DP slave was assigned to the slave's virtual slot 2. Since DPV1 was standardized, this diagnostic address has been assigned to virtual slot 0 (station proxy) for the **Technology CPU**
- If the slave has modeled a separate slot for the interface module (e.g. **Technology CPU** as an intelligent slave or IM 153), then its address is assigned to slot 2.

Activating / deactivating DP slaves via SFC 12...

With the **Technology CPU**, slaves that were deactivated via SFC 12 are no longer automatically activated at the RUN to STOP transition. Now they are not activated until they are restarted (STOP to RUN transition).

9.3 Interrupt events from distributed I/Os while the CPU status is in STOP

Interrupt events from distributed I/Os while the CPU status is in STOP

With the new DPV1 functionality (IEC 61158/ EN 50170, volume 2, PROFIBUS), the handling of incoming interrupt events from the distributed I/Os while the CPU status is in STOP has also changed.

Previous response by the CPU in the STOP state:

With CPUs 312 IFM to 318-2 DP, an interrupt event was initially noted while the CPU was in STOP mode. When the CPU status subsequently returned to RUN, the interrupt was then fetched by an appropriate OB (e.g. OB 82).

New response by the CPU:

With the **Technology CPU**, an interrupt event (process or diagnostic interrupt, new DPV1 interrupts) is acknowledged by the distributed I/O while the CPU is still in STOP status, and is entered in the diagnostic buffer if necessary (diagnostic interrupts only). When the CPU status subsequently returns to RUN, the interrupt is no longer fetched by the OB. Possible slave faults can be read using suitable SSL queries (e.g. read SSL 0x692 via SFC51).

9.4 Runtimes that change while the program is running

Runtimes that change while the program is running

If you have created a user program that has been fine-tuned in relation to certain processing times, please note the following points if you are using a **Technology CPU**:

- The program will run much faster on the **Technology CPU**.
- Functions that require MMC access (e.g. system start-up time, program download in RUN, return of DP station, etc), may sometimes run slower on the **Technology CPU**.

9.5 Converting the diagnostic addresses of DP slaves

Converting the diagnostic addresses of DP slaves

If you are using a **Technology CPU** as the master, please note that you may have to reassign the diagnostic addresses for the slaves since the changes to the DPV1 standard sometimes require two diagnostic addresses per slave.

- The virtual slot 0 has its own address (diagnostic address of the station proxy). The module status data for this slot (read SSL 0xD91 with SFC 51 "RDSYSST") contains IDs that relate to the entire slave/station, e.g. the "station error ID". Failure and restoration of the station are also signaled in OB 86 on the master via the diagnostic address of the virtual slot 0.

- The interface module is also modeled as a separate, virtual slot for some slaves (e.g. CPU as intelligent slave or IM 153) in which case it is assigned to virtual slot 2 with a separate address.
  When the **Technology CPU** is used as an intelligent slave, for example, the operating status change is signaled in diagnostic interrupt OB 82 of the master via this address.

Note

Reading diagnostics data with SFC 13 "DPNRM_DG":
The originally assigned diagnostics address still works. Internally, *STEP 7* assigns this address to slot 0.

When using SFC51 "RDSYSST", for example, to read module status information or module rack/station status information, you must also consider the change in slot significance as well as the additional slot 0.

9.6 Reusing existing hardware configurations

Reusing existing hardware configurations

If you reuse the configuration of a CPU 312 IFM to 318-2 DP for a **Technology CPU**, the CPU may not run correctly.

If this is the case, you will have to replace the CPU in the STEP 7 hardware configuration editor. When you replace the CPU, STEP 7 will automatically accept all the settings (if appropriate and possible).

9.7 Replacement of a Technology CPU

Replacement of a Technology CPU

The **Technology CPU** is supplied with a connector inserted in the power supply port.

You do not need to disconnect the cables of the CPU when you replace a **Technology CPU**. Insert a screwdriver with 3.5 mm blade into the right side of the connector to open the interlock mechanism, then unplug it from the CPU. Once you have replaced the CPU, simply plug the connecting plug back into the power supply connector.

9.8 Using consistent data areas in the process image for DP slaves

Using consistent data areas in the process image for DP slaves

A data area is consistent if it can be read or written to from the operating system as a consistent block. Data exchanged collectively between the stations should belong together and originate from a single processing cycle, that is, be consistent.

If the user program contains a programmed communication function, for example, access to shared data with XSEND/ XRCV, access to that data area can be coordinated by means of the "BUSY" parameter itself.

9.9 Load memory design for the Technology CPU

Load memory design for the Technology CPU

On CPUs 312 IFM to 318-2 DP, the load memory is integrated into the CPU and may be extended with a memory card,

The load memory of the **Technology CPU** is located on the Micro Memory Card (MMC). and is retentive. When blocks are downloaded to the CPU, they are stored on the MMC and cannot be lost even in the event of a power failure or memory reset.

Note

User programs can only be downloaded and thus the CPU can only be used if the MMC is inserted.

9.10 Altered retentive response with a Technology CPU

Altered retentive response with a Technology CPU

With data blocks for the **Technology CPU**

- you can set the retentive response in the block properties of the DB.
- Using SFC 82 "CREA_DBL" -> Parameter ATTRIB, NON_RETAIN bit, you can specify if the actual values of a DB should be maintained at POWER OFF/ON or STOP-RUN (retentive DB) or if the start values should be read from the load memory (non-retentive DB).

9.11 FMs/CPs with their own MPI address in a Technology CPU rack

FMs/CPs with their own MPI address in a Technology CPU rack

Table 9-2 Behavior of FMs/CPs with their own MPI address

All CPUs except for CPU 317-2 DP, CPU 315T-2 DP, CPU 317T-2 DP and CPU 318-2 DP	CPU 317-2 DP, CPU 315T-2 DP, CPU 317T-2 DP and CPU 318-2 DP
If there are FMs/CPs with their own MPI address in the central rack of an S7-300, then they are in the exact same CPU subnet as the CPU MPI stations.	If there are FMs/CPs with their own MPI address in the central rack of an S7-300, then the CPU forms its own communication bus via the backplane bus with these FMs/CPs, which are separated from the other subnets. The MPI address of such an FM/CP is no longer relevant for the stations on other subnets. The communication to the FM/CP is made via the MPI address of the CPU.

When you replace your existing CPU with a Technology CPU, therefore,

- you must replace the existing CPU by the Technology CPU in the STEP 7 project,
- Reconfigure the OPs. you must re-assign the control and the destination address (= the MPI address of the Technology CPU and the slot of the respective FM)
- Reconfigure the project data for FM/CP to be loaded to the CPU.

This is necessary to ensure that the FM/CP can still be "addressed" by the OP/PG in this rack.

9.12 Information about interface X3 DP(DRIVE)

Interface X3 assigned to DP(DRIVE)

Note that the 2nd interface on the Technology CPU is assigned to PROFIBUS DP(DRIVE), i.e. it cannot be used as the second interface for PROFIBUS DP.

No PG/OP on DP(DRIVE)

We do not recommend that you connect a PG/OP to DP(DRIVE).

Reason: If you connect a PG/OP to DP(DRIVE), the properties of DP(DRIVE) change (e.g. isochronism), and the synchronism between drvies may be lost as a result. Always therefore connect a PG/OP to the MPI/DP interface and access the DP(DRIVE) via the "Routing" function.

No diagnosis on DP(DRIVE)

Remember that you cannot evaluate diagnostic data from DP(DRIVE) in your STEP 7 user program if you are using the Technology CPU.

However, with your PC/PG connected to PROFIBUS DP, you can use the "routing" function to access drive parameters in the DP(DRIVE) line (in conjunction with appropriate drive tools) for commissioning and diagnostic purposes.

Appendix A

A

A.1 List of abbreviations

List of abbreviations

Table A-1 List of abbreviations

Abbreviation	Explanation
IA	Automation system
O	Output
HMI	Operator control and process monitoring
DB	Data block
DP	Distributed I/Os
DP (drive)	Distributed I/Os for drives
E	Input
EU	Expansion Unit
FB	Function block
FC	Function
GD	Global data
HMI	Human Machine Interface
IM	Interface module
M	Flags
MC	Motion Control
MMC	Micro Memory Card
MPI	Multipoint Interface
OB	Operation Block
OP	Operation Panel
PIO	Process image of outputs
PII	Process image of inputs
PG	Programming device
PS	Power Supply
SFC	System Function Call
T	Timer
TD	Text Display
C	Counter
CR	Central rack

Abbreviation	Explanation
SCC	Scan cycle check point

Glossary

Activation/deactivation of slaves

You activate a DP slave with the SFC 12 "D_ACT_DP" and therefore include it in the cyclic processing. Deactivating a slave removes the DP slave from the cyclic processing.

Address

An address represents the identifier of a specific address or address range. Examples: input I 12.1; flag word MW 25; data block DB 3.

Analog module

Analog modules convert analog process values (for example, temperature) into digital values that can be processed by the CPU, or convert digital values into analog manipulated variables.

Back-up memory

The backup memory ensures buffering of the memory areas of a CPU without backup battery. It backs up a configurable number of timers, counters, flags and data bytes, the retentive timers, counters, flags and data bytes.

Clock flag

Flag which can be used to generate clock pulses in the user program (1 flag byte).

Note

When operating with S7300 CPUs, make sure that the clock flag byte is not overwritten in the user program!

Code block

A SIMATIC S7 code block contains elements of the STEP 7 user program. (In contrast to a data block: This contains only data.)

Cold restart

On CPU startup (e.g. after it is switched from STOP to RUN via the mode selector or with POWER ON), OB100 (restart) is initially executed, prior to cyclic program execution (OB1). On restart, the input process image is read in and the STEP 7 user program is executed, starting at the first command in OB1.

Configuration

Assignment of modules to racks/slots and (for example with signal modules) addresses.

Consistent data

Data that belongs together in terms of its content and must not be separated, is designated as consistent data.

As an example, the values of analog modules must always be treated as consistent data; i.e. the value of an analog module must not be corrupted by reading it out at two different times.

Counters

Counters are part of CPU system memory. The content of "counter cells" can be modified by STEP 7 statements (for example, up/down count.)

Cycle time

The cycle time represents the time a CPU requires for one execution of the user program.

Data block

Data blocks (DB) are data areas in the user program and contain the user data. There are global data blocks which can be accessed by all code blocks, and instance data blocks which are assigned to a specific FB call.

Data, static

Static data are data that are used only within a function block. These data are stored in an instance data block belonging to the function block. Data stored in an instance data block are retained until the next function block call.

Data, temporary

Temporary data represent local data of a block. They are stored in the L-stack when the block is executed. After the block has been processed, these data are no longer available.

Diagnostic interrupt

Modules capable of diagnostics report detected system errors to the CPU by means of diagnostic interrupts.

Diagnostics

System diagnostics

Diagnostics buffer

The diagnostics buffer is a buffered memory area in the CPU. It stores diagnostic events in the order of their occurrence.

DP (drive)

PROFIBUS interface that is controlled by the integrated technology of the CPU isochronously (and therefore also equidistant).

DP master

A master which behaves in accordance with EN 50170, Part 3 is known as a DP master.

DP slave

A slave operated on the PROFIBUS with PROFIBUS DP protocol and which behaves in accordance with EN 50170, Part 3 is referred to as DP slave.

DPV1

The designation DPV1 means extension of the functionality of the acyclical services (to include new interrupts, for example) provided by the DP protocol. The DPV1 functionality is integrated in IEC 61158 / EN 50170, Volume 2, PROFIBUS.

Engineering station

PC workplace, at which configuration work is performed on a control system.

Error display

One of the possible reactions of the operating system to a runtime error is to output an error message. The other possible reactions include: Error reaction in the user program, CPU in STOP.

Error handling via OB

After the operating system has detected a specific error (e.g. an access error with STEP 7), it calls a dedicated organization block (error OB) in which the subsequent behavior of the CPU can be defined.

Error reaction

Reaction to a runtime error. The operating system can react in the following ways: It sets the automation system to STOP, indicates the error, or calls an OB in which the user can program a reaction.

Execution level

Execution levels form the interface between the operating system of the CPU and the user program. The sequence for executing the blocks of the user program is specified in the execution levels.

Flag

Flags are part of the CPU's system memory. They store intermediate results of calculations. They can be accessed in bit, byte, word or double word operations.

Force

The force function can be used to assign fixed values to specific variables of a user program or CPU (also: inputs and outputs).

In this context, please note the limitations listed in the *Overview of the test functions section in the Test Functions, Diagnostics and Troubleshooting chapter in the S7-300 Installation manual*.

Function

According to IEC 1131-3, a function (FC) is a code block without static data. A function allows transfer of parameters in a user program. Functions are therefore suitable for programming frequently occurring complex functions, e.g. calculations.

Function block

According to IEC 1131-3, a function block (FB) is a code block with static data. An FB enables the transfer of parameters in the user program. Function blocks are therefore suitable for programming frequently occurring complex functions, e.g. controls, mode selections.

GD circuit

A GD circuit comprises a number of CPUs sharing data by means of global data communication, and is used as follows:

- A CPU sends a GD packet to the other CPUs.
- A CPU sends and receives a GD packet from another CPU.

A GD circuit is identified by a GD circuit number.

GD element

A GD element is generated by assigning shared global data. It is identified by a unique global data ID in the global data table.

GD packet

A GD packet can consist of one or more GD elements transmitted in a single message frame.

Global data

Global data can be addressed from any code block (FC, FB, OB). These include flag F, inputs I, outputs Q, timers, counters and data blocks DB. Global data can be accessed either absolutely or symbolically.

Global data communication

Global data communication is a method of transferring global data between CPUs (without CFBs).

GSD file

The device data file (GSD file) contains all slave-specific properties. The format of GSD files is specified in EN 50170, Volume 2, PROFIBUS.

Instance data block

The STEP 7 user program assigns an automatically generated DB to every call of a function block. The instance data block stores the values of input, output and in/out parameters, as well as local block data.

Integrated inputs/outputs for integrated technology

The Technology CPU has 4 digital inputs and 8 digital outputs integrated. You use these inputs and outputs for technology functions, e.g. reference point acquisition (BERO) or high-speed output cam switching signals. The integrated inputs and outputs can also be used with technology functions in the *STEP 7* user program.

Integrated technology

In addition to the standard PLC functions, the Technology CPU has been expanded with integrated technology functions. The operating system of the Technology CPU has also been expanded in order to guarantee fast processing times for these technology functions.

Intelligent DP slave

Signal preprocessing field device. One of its features is that the input/output range provided for the DP master does not correspond to an actually existing I/O, but an input/output range that is mapped by a preprocessing CPU.

Interrupt

The CPU's operating system knows different priority classes for controlling user program execution. These priority classes include interrupts, e.g. process interrupts. When an interrupt is triggered, the operating system automatically calls an assigned OB. In this OB the user can program the desired response (e.g. in an FB).

Load memory

Load memory is part of the CPU. It contains objects generated by the programming device. It can be implemented either as a plug-in memory card or as integrated memory.

Local data

Data, temporary

Master

When a master is in possession of the token, it can send data to other nodes and request data from other nodes (= active node).

Micro memory card (MMC)

Micro memory cards are memory media for CPUs and CPs. The difference to the memory card is the smaller size.

MPI

The multi-point interface (MPI) is the SIMATIC S7 interface for programming devices. It allows simultaneous operation of several nodes (programming devices, text displays, operator panels) on one or more CPUs. Each node is identified by a unique address (MPI address).

Nesting depth

Using block calls, one block can be called from within another. Nesting depth is referred to as the number of simultaneously called code blocks.

OB priority

The CPU operating system distinguishes between different priority classes, for example, cyclic program execution, process interrupt controlled program processing. Each priority class is assigned organization blocks (OBs) in which the S7 user can program a response. Per default, OBs have different priorities and they are processed and interrupt each other according to their priority classes.

Operating mode

SIMATIC S7 automation systems know the following operating modes: STOP, STARTUP, RUN.

Operating system of the CPU

The operating system of the CPU organizes all the functions and processes of the CPU that are not connected to a special control task.

Organization block

Organization blocks (OBs) form the interface between CPU operating system and the user program. The sequence for executing the user program is specified in the organization blocks.

Parameter

1. Variable of a STEP 7 code block
2. Variable for setting the response of a module (one or several per module). As delivered, each module has an appropriate default setting that can be changed via configuration in STEP 7. The parameters are divided into static and dynamic parameters.

Parameters, dynamic

In contrast to static parameters, the dynamic parameters of modules can be changed during operation by calling an SFC in the user program, for example limit values of an analog signal input module.

Parameters, static

Unlike dynamic parameters, static parameters of modules cannot be changed by the user program. You can only modify these parameters by editing your configuration in STEP 7, for example, modification of the input delay parameters of a digital signal input module.

Priority class

The S7 CPU operating system provides up to 26 priority classes (or "Program execution levels"). Specific OBs are assigned to these classes. The priority classes determine which OBs interrupt other OBs. Multiple OBs of the same priority class do not interrupt each other. In this case, they are executed sequentially.

Process image

The process image is a component of the system memory of the CPU. At the start of the cyclic program the signal states of the input modules are transferred to the process image of the inputs. At the end of the cyclic program the process image of the outputs is transferred as a signal state to the output modules.

PROFIBUS DP

Digital, analog and intelligent modules, as well as a wide range of field devices in accordance with EN 50170, Part 3, such as drives or valve terminals are moved to the local process by the automation system across distances of up to 23 km.

The modules and field devices are interconnected with the automation system via the PROFIBUS DP field bus and are addressed in the same way as the centralized I/O.

Programming device

Programming devices are essentially personal computers suitable for industry that have a compact design and can be transported. Their distinguishing feature is the special hardware and software for SIMATIC programmable logic controllers.

Reduction factor

The reduction factor determines the send/receive frequency for GD packets on the basis of the CPU cycle.

Retentive address areas

A memory area is considered retentive if its contents are retained even after a power loss and transitions from STOP to RUN. The non-retentive area of the flags, timers and counters is reset following a power failure and a transition from the STOP mode to the RUN mode.

Retentive can be the:

- Flags
- S7 timers
- S7 counters
- Data areas

Revision level

Differentiates products of the same order number. The product version is incremented when forward-compatible functions are enhanced, after production-related modifications (use of new parts/components) and for bug fixes.

Runtime error

Errors that occur during execution of the user program on the automation system (in other words, not in the process).

S7T Config

With S7T Config you configure the technology objects required to implement your motion control task. The STARTER for the drives from the MICROMASTER and SINAMICS families is integrated in S7T Config.

Segment

→ Bus segment

Shutdown

What happens during shutdown?

1. The control of the Technology CPU is already in STOP mode during shutdown. The outputs of the centralized and distributed I/Os are deactivated.
2. The integrated inputs/outputs for integrated technology and the ET 200M on the DP (DRIVE) are still active during shutdown.
3. The integrated technology of the Technology CPU shuts down the drives on the PROFIBUS DP (DRIVE) in a controlled manner.
4. The integrated technology of the CPU then also goes into STOP. The integrated inputs/outputs for integrated technology and the ET 200M on the DP (DRIVE) are deactivated.

The maximum duration of shutdown depends on your configuration in S7TConfig.

Signal module

Signal modules (SMs) form the interface between the process and the automation system. There are digital input and output modules (input/output module, digital) and analog input and output modules (input/output module, analog).

Signal status list

The system status list contains data that describes the current status of an S7-300. You can always use this list to obtain an overview of:

- The configuration of the S7-300
- The current CPU configuration and configurable signal modules
- The current states and processes in the CPU and in configurable signal modules.

Slave

A slave can only exchange data with a master after being requested to do so by the master.

STARTUP

The STARTUP mode is executed at the transition from STOP to RUN mode. It can be triggered by means of the mode selector switch, or after power on, or by an operator action on the programming device. An S7-300 performs a restart.

STEP 7

Programming software for the creation of user programs for SIMATIC S7 controllers.

Substitute value

Substitute values are parameterizable values which output modules transfer to the process when the CPU switches to STOP mode.

In the event of an I/O access error, a substitute value can be written to the accumulator instead of the input value which could not be read (SFC 44).

System diagnostics

System diagnostics is the detection, evaluation, and indication of faults/errors which occur within the automation system. Examples of such error/faults are program errors or failures on modules. System errors can be indicated by LEDs or in STEP 7.

System function

A system function (SFC) is a function integrated in the operating system of the CPU that can be called when necessary in the STEP 7 user program.

System function block

A system function block (SFB) is a function block integrated in the operating system of the CPU that can be called when necessary in the STEP 7 user program.

System memory

System memory is an integrated RAM memory in the CPU. System memory contains the address areas (e.g. timers, counters, flags) and data areas that are required internally by the operating system (for example, communication buffers).

Technology configuration data

The configuration that you have created with *STEP 7* is stored in the technology configuration data.

Technology data block

The integrated technology provides current information on the status and on the values of the technology objects via the technology data blocks.

Technology objects

Technology objects are the logical representation of axes, output cams, measuring inputs, cams and external encoders, with which the drive components can be controlled. The technology objects configured with the *SIMATIC S7-Technology* option package contain definitions for the physical properties of the mechanical system, for limits, monitoring and control.

Technology system data

Technology system data are the data of the technology objects, such as drive axis, output cam, ...

Time-of-day interrupt

Interrupt, time-of-day

Timer

Times

Times

Times are part of CPU system memory. The content of "timer cells" is automatically updated by the operating system, asynchronously to the user program. STEP 7 statements are used to define the precise function of the timer cell (for example, ON delay) and to initiate their execution (for example, start).

User memory

The user memory contains the code blocks and data blocks of the user program. User memory can be integrated in the CPU, or stored on plug-in memory cards or memory modules. However, the user program is principally processed from the working memory of the CPU.

User program

In SIMATIC, a distinction is made between the operating system of the CPU and user programs. The latter are created by the STEP 7 programming software in the available programming languages (ladder logic and statement list) and are stored in code blocks. Data is stored in data blocks.

Working memory

The working memory is a RAM in the CPU that the processor accesses when processing the user program.

Index

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X