pilz

Programmable Safety Systems
PSS-Range

ST System Description
Item No. 18 587

The spirit of safety.

Version IV, Dezember 1999

Contents

Contents

Contents

Introduction

This System Description forms part of the PSS System manuals. It explains how the standard section of the PSS-range of programmable safety systems function and operate. This manual is divided into the following chapters:

1 Introduction

2 Overview
Contains information on the most important features of the safety systems.

3 Intended application
Explains the purpose of the system and the conditions under which it can be applied. Also gives information on important safety regulations.

4 Design
Explains the structure of the hardware and the functions of the individual units.

5 Programming
Describes the programming and the program cycle as well as the addressing for the safety systems.

6 Operation
Explains the PSS system cycles and changes which can be made by the operator.

7 Start-up Procedure
Explains the procedure during the initial start-up and after a reset, e.g. after a fault.

8 Error diagnostics and correction
Points out possible faults, shows how errors can be detected and removed using a diagnostics program.

9 Appendix
Contains an overview list of the system data blocks and organisation blocks.

10 Index

Introduction

Definition of symbols

Information in this manual which is of particular importance can be identified as follows:

DANGER!

This warning must be heeded! It warns of a **hazardous situation which poses an immediate threat of serious injury and death,** and indicates preventive measures which may be taken.

WARNING!

This warning must be heeded! It warns of a **hazardous situation which could lead to serious injury or death,** and indicates preventive measures which may be taken.

CAUTION!

This refers to a hazard which can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures which may be taken.

NOTICE
This describes a situation in which the unit(s) could be damaged and also provides information on preventive measures which may be taken.

INFORMATION
This gives advice on applications and provides information on special features, as well as highlighting areas within the text which are of particular importance.

Overview

The PSS-range comprises modular and compact programmable safety systems for use in safety circuits in plant and machinery. They incorporate a failsafe section (FS-section) and a standard section (ST-section) into a single unit.
The failsafe section processes all the safety-related functions and is designed with three-channel diversity. The application program is processed separately by each channel. If the three channels are not identical the system will immediately switch to a safe condition and switch off all the outputs.
The application program is created once only and, once installed, can be approved by a test house such as BG or TÜV, or by the company's internal test/quality control department.
The failsafe section communicates independently from the standard section, i.e. without feedback. Errors in the standard section's application program will nave no effect on the failsafe section.
The standard section is single-channel and operates like a normal PLC (e.g. a P10). It has its own bus system with separate bus interface.

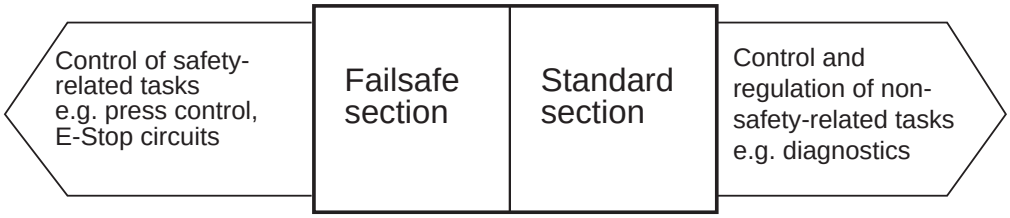


Fig. 2-1: Functions of the failsafe and standard section

Hardware

On the modular systems different input and output modules can be installed on to a module rack. The FS and ST sections communicate with the CPU via separate buses. The basic system comprises a module rack, power supply and a CPU.
On the compact PSS, the power supply, CPU, bus and periphery modules are incorporated within a single housing.

INFORMATION
The bus for the ST section on the PSS 3000 is an option.

Overview

Example of a modular PSS

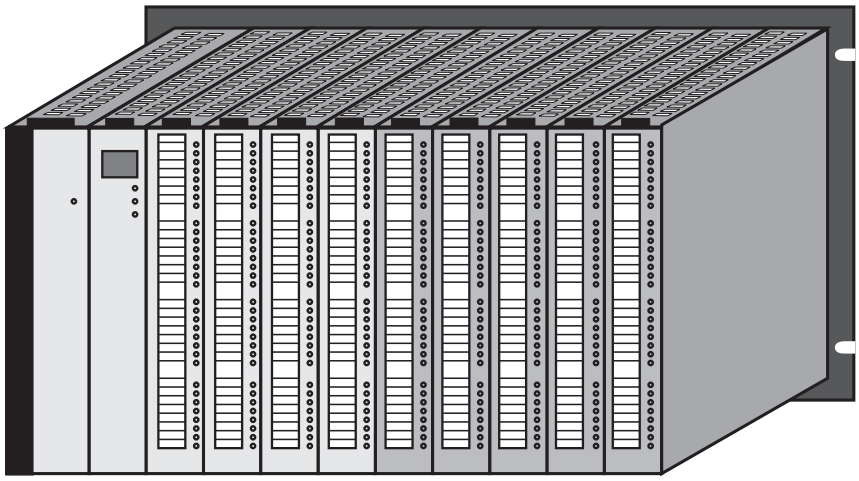


Fig. 2-2: Example of a PSS 3000 layout (from left to right): power supply, CPU, 4 FS modules and 5 standard modules

Example of a compact PSS

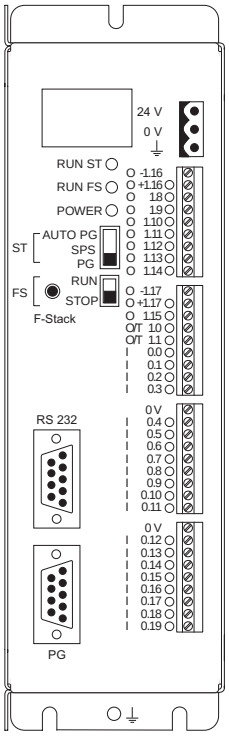
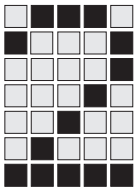


Fig. 2-3: PSS 3032 with power supply, CPU and peripheries

Programming

Application programs are created on a PC, the PG programming device, using a special software package which provides various editors for inputting programs.
Application programs for different plants are managed in “projects”. Each project corresponds to a directory in the MS-DOS operating system.
Projects are divided into blocks. For this reason programs are divided into individual functions (similar to sub-programs). Each function corresponds to one block.
The final step is to send the program to the programmable safety system via the serial interface.



Notes

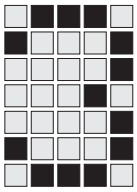
Intended Application

Safety guidelines

- The unit should only be installed and commissioned by a competent, qualified engineer who is familiar with the contents of this manual and the installation guidelines and has a good knowledge of the current regulations regarding health and safety at work and accident prevention.
- Both the guarantee and the approval on this unit will be rendered invalid if the housing is opened or any changes are made to the PCB-boads, e.g. exchanging components or additional soldering carried out by the user.
- The safety regulations and the EMC-measures described in the “Installation Manual” must be observed at all times.

WARNING!

The ST section of the safety system must **not** be used for safety-related applications/functions.



Notes

Design (Hardware)

Compact system

A compact system is composed of the following parts:

- Bus
- CPU (1)
- Power supply (2)
- Input and output modules (3)

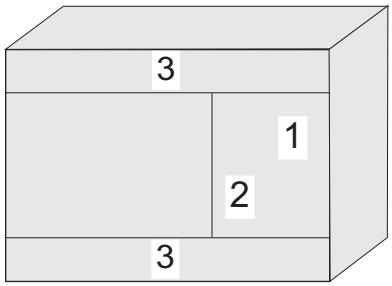


Fig. 4-1: Example of a compact system: PSS 3056 design

Modular system

Modular systems are composed of the following parts:

- Module rack (1)
- Power section(2)
- CPU module (3)
- Input and output modules on the FS section (4) and ST section (5)

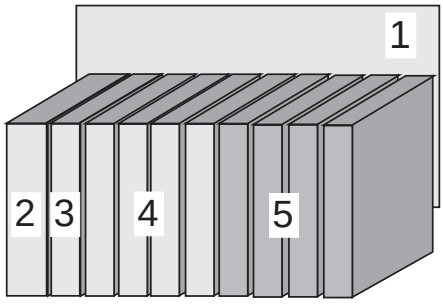


Fig. 4-2: Example for the structure of a PSS 3000

Design (Hardware)

The base unit consists of a module rack, power supply and CPU. Input and output modules are required to input and output data. Modules for failsafe applications can be installed on the module rack and if a standard bus is integrated, standard modules can be installed.

Further information on the module rack can be found in the “Installation Manual” and the module rack description.

INFORMATION
Modules for standard applications may only be used on the PSS 3000 with a standard bus (PSS BMP 8, order no. 301 000 and PSS BMP4/2, order no. 301 006).

Expansion racks for modular safety systems

A maximum of two expansion module racks can be connected to the base rack for standard modules, giving a maximum of 24 slots. The expansion modules are connected via expansion units, which are called EPBM for the base rack and EPEM for the expansion rack.

Power supply

The power supply provides the internal voltage to the CPU and bus. Power supplies are available for different voltages, e.g 230 VAC and 24 VDC.
The battery acts as a buffer for the CPU memory module.
The power supply on modular systems must always occupy the first slot on the rack.

CPU

The CPU module is the central processing unit for the safety system. It controls the input and output modules, and processes and stores the application program as well as the variable data. It has different operating

elements and interfaces, e.g.

- 4-digit display
- LED for operating mode and mains voltage
- 3-position switch for selecting the standard section (ST) operating mode
- Scroll error stack button
- 2-position switch for selecting the failsafe section (FS) operating mode
- Slot for standard section program memory cartridge
- Programming interface RS 485
- User interface RS 232

The standard section is designed as single-channel and the failsafe section consists of three microprocessors which operate independently. One of these processors, which has a separate bus system and a bus interface, also processes the standard section.

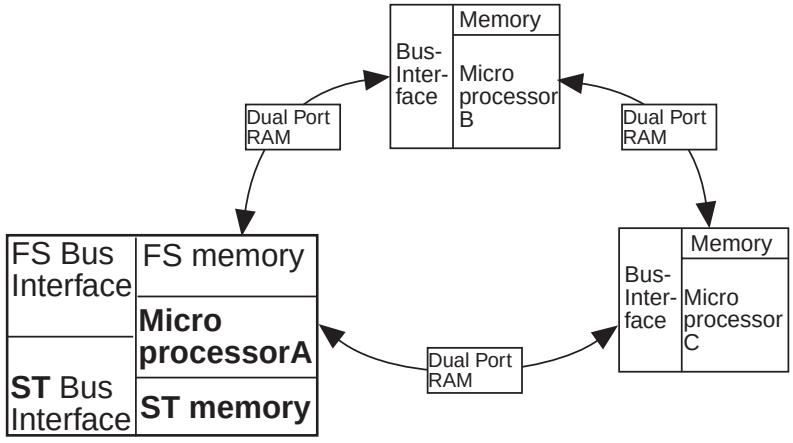


Fig. 4-3: Processing via three microprocessor

Memory

The CPU provides two memories for the standard section:

- Program memory
- Data memory

Program memory

The program memory is a plug-in cartridge and has either a RAM or Flash-EPROM memory. The Flash-EPROM memory can be programmed once it is inserted and as such a special programming device is not required.

There are various memory capacities available:

- RAM: 64, 128, 256 or 512 kB
- Flash-EPROM: 64, 256 or 512 kB

Data memory

Variable values such as set data, error messages and system data are stored in the data memory, which is divided into data blocks, each with a maximum of 1024 words.
A data block consists of a data block header and user data. The header contains information on the data block. There are two types of data block:

- Read only data blocks
  can only be read by the application program
- Read/write data blocks
  can be both read from and written to.

The data memory is divided in a similar manner:

- Read only data memory
  Stored in the program memory and contains the read only data blocks (data block header and user data)
- Read/write data memory
  Contains the user data of the read/write data blocks and has a memory capacity of 170 kB. The memory sector is non-volatile.
  A compressed copy of each read/write data block is stored in the program memory. This contains the values entered during editing. When an application program is started-up, reset or a cartridge is changed the read/write data blocks are automatically given the values edited in this copy.

Progamming (PG) interface

The PG interface (PG: programming device) enables communication between the programming device and the safety system.
The PG interface will either be either an RS 232 or RS 485 interface, depending on the type of safety system you have (see the “Installation Manual” for the respective safety system). If the PG interface is an RS 485, it is connected to the programming device using a PAP interface adapter. If the PG interface is an RS 232, an interface adapter is not required.

User interface

The user interface operates to the RS 232C Standard and can be configured. It can, for example, be used to communicate with a supervisory host PLC. The standard settings are:

- Baud rate: 9600 Baud
- Parity: even
- Stop bit: 1
- Data bit: 8
- Handshake: on
- Timeout, receive: 15 ms
- Timeout, send: 5,000 ms

The configuration is set and then stored in DB 006. The data block can be edited and the user interface can be configured using other data. DB 007 and DB 008 are used as send/receive buffers. More information on configuring the user interface via DB 006 can be found on page 6-13.

Timers

The CPU has 64 timers, controlled via a central timer.
The timers have a switch-on delay function and each timer is calculated from two parameters:

Time = time base x time value

Time base:	0 corresponds to 50 ms
	1 corresponds to 100 ms
	2 corresponds to 1 s
	3 corresponds to 10 s
	4 corresponds to 1 min
Time value:	1 ... 32.767

Example: Time is to be 8 s
Time base: 2, time value: 8
Time = 1 s x 8 = 8 s

Timers 0 ... 63 are available for the standard section. Timers 64 ... 127 of the failsafe section are read only.

Counters

The CPU has 64 counters, each consisting of one counter word and one counter bit. Counter values can accept values between -32.768 and 32.767. If the value of the counter word is > 0, the counter bit assumes the value 1. The standard section uses the counters 0... 63. Counters 64 ... 127 of the failsafe section are read only.

Display

The 4-digit hexadecimal display is used to output error messages and/or error numbers.

Format of error messages:
S-xx Error in the standard section, xx: error class 01h ... FFh

Format when displaying error messages from the error stack:
Syxx Error in the standard section
y: Entry number in the error stack 0h ... Fh
x: Error class 01h ... FFh

Error messages starting with “F”, “*” or “+” refer to the failsafe section.

Selector switch

The programmable safety system has two selector switches:

- 3-position selector switch for the standard section
- 2-position selector switch for the failsafe section (see failsafe section description)

The selector switch for the standard section has three functions:

The program starts up automatically, the programming device has read/write access (e.g. overwriting blocks during the program cycle).

Program starts up automatically, the programming device has read access only.

Program stops, the programming device has read/write access

Error stack button

Error messages are stored in an error stack. The current error meassage is always displayed. To display the message previous to the current one, the scroll error stack button must be pressed. As long as the button remains pressed, the error stack scrolls through the following: error class (C), error number (N), error location (AT) and error parameter (PARA).
(see chapter 8). When the button is released and pressed again the message before it will be displayed, etc.

Input and output modules

There are a wide variety of input and output modules available to communicate between the programmable safety system and the plant or machine, for example:

- Digital input module with 32 inputs
- Output module with 32 single-pole 2 A outputs
- Input and output modules with 16 inputs and 16 outputs

Creating programs

Application programs for different plants must be allocated to different “Projects”. A project corresponds to a directory, e.g. on a computer’s hard drive. Each project is divided into blocks, as follows:

- Organisations blocks (OB), which form the interface between the application program and the operating system
- Program blocks (PB), which contain fundamental operating functions as well as functions specific to your plant
- Function blocks (FB), which are made up of programming instructions for specific individual tasks
- Standard function blocks (SB), which carry out standard functions
- Data blocks (DB), which contain fixed or variable data

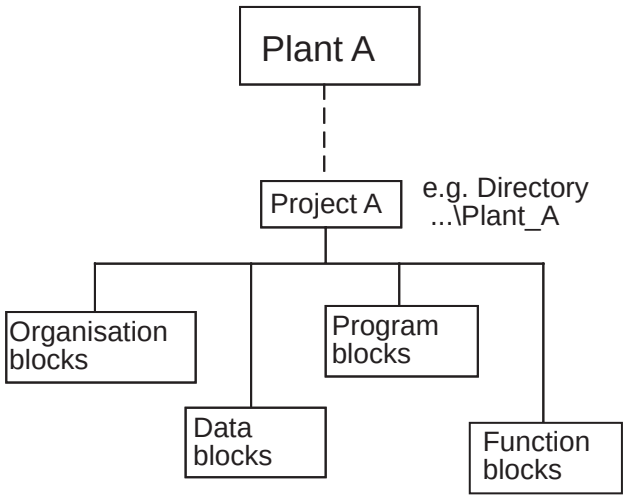


Fig. 5-1: Program structure

INFORMATION
The failsafe and standard programs must be stored in different project directories.

Programming

Program transfer

Individual blocks or all blocks within a program can be transferred to the safety system. Communication is only possible if the programming device is connected to the safety system. Depending on the type of PG interface on the safety system, you may also require a PAP interface (see chapter 4, section "PG (programming) interface".

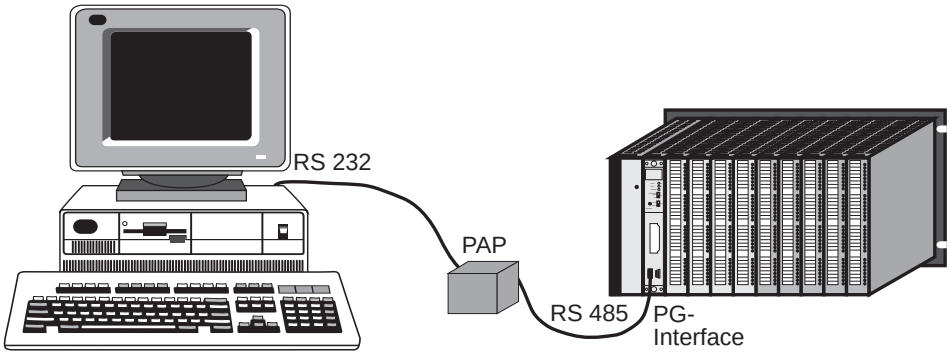


Fig. 5-2: PSS 3000 program transfer

Program cycle

An executable program is divided into the following blocks:

- Start block
- Read block FS
- Application program FS
- Read block ST
- Application program ST
- Output block ST
- Output block FS
- Test block FS
- Scan time block

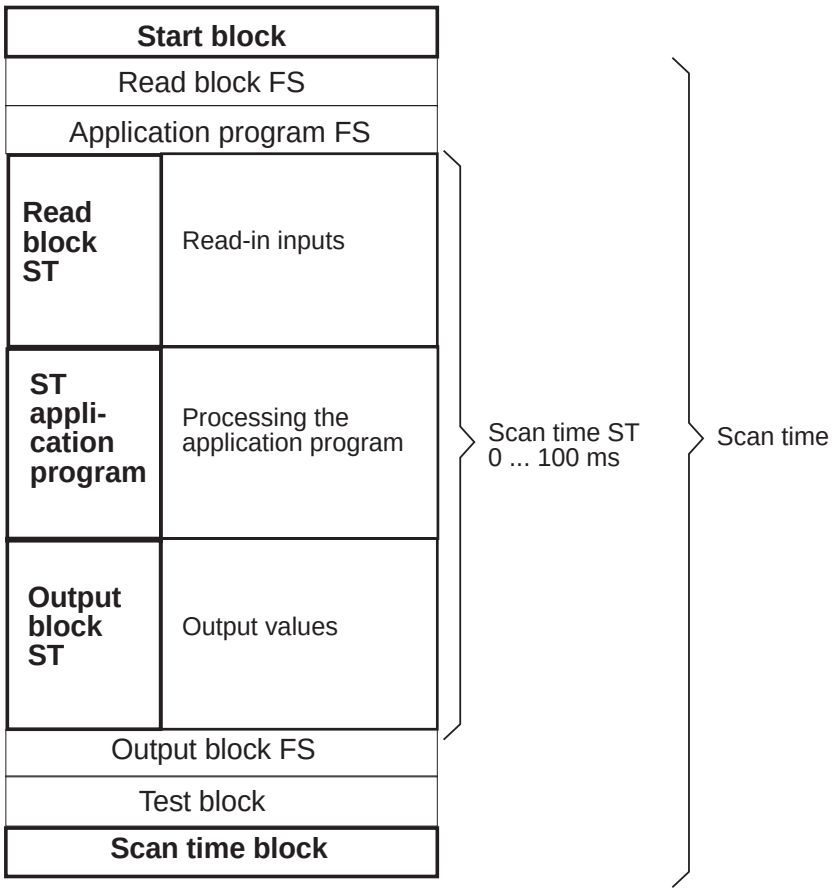


Fig. 5-3: Program cycle

A program cycle ends when all the blocks have been run through once. Process times are described in the chapter “Operation” on page 6-2.

Programming

Addressing

The modules are addressed through the slots. Each slot is allocated a slot number. The digital inputs and outputs are addressed with this slot number and a bit number. The two entries are separated by a full stop.
On modular systems the first two slots are occupied by the power supply and the CPU. Subsequent slots are addressed consecutively beginning with 0.

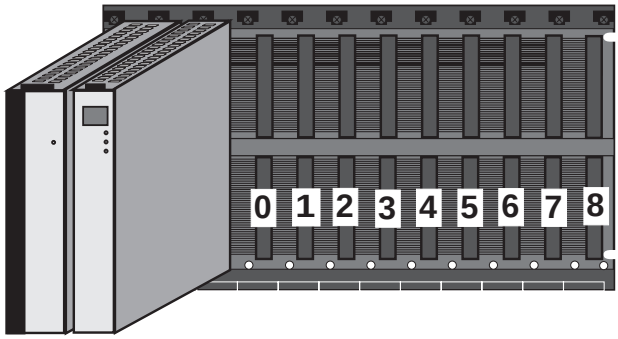


Fig. 5-4: Addressing the slots on the PSS 3000 and PSS 3100

Example: Bit 8 is to be addressed from the module in slot 3
Address: 3.08

Word modules which have more than 32 bits can be freely addressed. The address can be selected from the range 0 ... 16383 and is entered in data block DB 005.

Organisation blocks

The various functions of the organisation blocks are set by the operating system. Each application program must contain the cycle organisation block OB 001. It manages, amongst other things, the program cycle. The application program blocks are called up in the cycle-OB. All other organisation blocks are reserved for a particular application and must not be used. Functions of the organisation blocks are described in the section “Programming” in the “Programming Manual” or in this manual in the Appendix.

INFORMATION
Only use the OBs listed in the Appendix.

Standard function blocks

Standard function blocks contain functions which are identical on several plants or machines. The blocks are divided into two sections:

- SB 002 ... SB 199 are available to the user and can be used for any function (see below for exceptions).
- SB 001, SB 003, SB 007, SB 011, SB 015, SB 041 and SB 200 ... SB255 are predefined and supplied by Pilz.

The predefined standard function blocks have the following functions:

- SB 003 ... SB041: 32-Bit-Arithmetic
- SB 200 ... SB253: reserved
- SB 254: Communication between the standard section and the operating system.

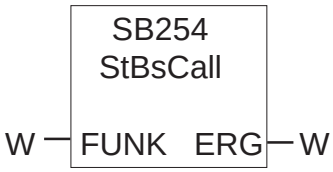
INFORMATION
Do not assign newly created blocks with the numbers of predefined function blocks.
Any standard function block saved under this description will overwrite the original block. This will then be no longer available and you will be unable to carry out the functions described within it.

Standard function block SB 254

Standard function block SB 254 is used for communication between the standard section and the operating system. It provides the following functions:

- Setting the real-time clock
- Operating the display
- Selecting failsafe data blocks
- Configuring the user interface
- Sending data via the user interface
- Receiving data via the user interface

Standard function block SB 254 can be called up via a "CALL"-command in the application program. It has the following structure:



The block's function is set through the input parameter "FUNK". Additional parameters will be required, depending on the function, and these parameters are stated in DB 004, DW 200 ... DW 220. The output parameter "ERG" indicates whether the function has been carried out correctly. If an error messages is received, data blocks DB 000, DW 200 ... DW220 will contain the cause of the error.

Parameters for the standard function block should be set as follows:

- Enter the parameters in the block (DB 004, DB 006 or DB007, depending on the application)
- Enter the function code (input parameter "FUNK")
- Allocate an operand to the output parameter "ERG", e.g. flag word
- Call up SB 254
- Interrogate the contents of the flag word. If it contains the code for an error message, call up the data block (DB 000, DB 006, DB 007 or DB 008) where the error descriptions can be found.

Communication with peripheries

The CPU can communicate in two ways with peripheries:

- Direct
  The signal status of the inputs is read directly or outputs are set directly. In addition the available commands are, for example, L PB, L PW, T PB, T PW etc.
- Via process images
  The status of the inputs and outputs is stored as a process image. Commands for reading and writing to the process image are e.g. L E, L EB, L EW, =A, T AB etc.

Direct periphery access

The direct access has the advantage that signals shorter than the scan time can be processed. The application program can scan the inputs and outputs several times during the program cycle and always contains the current status.

INFORMATION
There is always direct access to word modules.

Process images

Communication is normally carried out via the process image. The status of the inputs is read at the beginning of the program cycle and stored in the process image of the inputs (PII). Following this the application program is called up and processed with the process image values. Once the application program has finished, the process image of the outputs (PIO) is written to the outputs.
The advantage of communicating with the peripheries via the process image is that the status of the inputs and outputs remains the same during a program cycle.

Operation

Times

The time required by the system to process a program, including all sub-programs is known as the scan time. It consists of the following times:

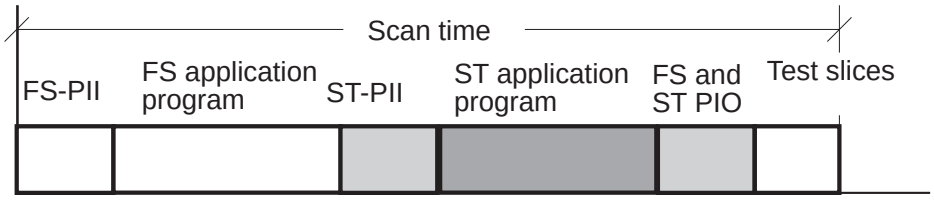


Fig. 6-1: Program structure

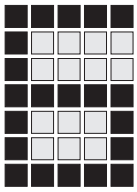
FS process image of inputs (PII)
see FS System Description

FS application program FS
see FS System Description

ST process image of inputs (PII)
The computer reads the inputs and stores the PII.
Time taken: a few ms

ST application program
The application program is started after the inputs values have been read successfully. The standard section enables read access to the failsafe inputs and outputs, flags, counters, counter words and timers.
Time taken for FS- and ST-application program together: max. 100 ms

ST process image of outputs (PIO)
After the application program has been processed, the output values are stored as a PIO and passed to the outputs.
Processing the FS-outputs, see FS System Description.
Time taken: a few ms.



Test bit slices
A test block is processed at the end of each cycle. All system tests are divided into slices. Each test bit slice takes 1 ms . One test bit slice is carried out automatically in every test block.
The number of test bit slices can be influenced by the user via an operating system call up (SB 254) in the failsafe section.

Scan time

The maximum and minimum scan times are set using the failsafe program. If the failsafe program is not loaded, the maximum scan time is 100 ms, and a minimum scan time cannot be set. The standard section is not called up if 0 is entered for the maximum ST block run time in the failsafe section. If the maximum ST block run time is exceeded, the standard section goes into a STOP-condition.
Failsafe process alarms can be interrupted by the standard application program. The alarm processing times belong to the FS block run time. If there is a process alarm error, the standard section is interrupted and only continues in the next cycle, i.e. there is no output of the PIO in this cycle, and in the following cycle the PII is not read in the standard section.

Self check

The self check is carried out in the FS section, refer to the FS System Description Manual.

Operation

Set layout configuration

The system detects all bit modules automatically. It must be told, however, on which slots the word modules are located. As word modules can be addressed, a start address must be given for each word module.

Enter set layout configuration

The start addresses for the word modules are entered in **DB 005**.

Data word	Code	Meaning
0	KF	Start address for word module at slot 0
1	KF	Start address for word module at slot 1
2	KF	Start address for word module at slot 2
.		
.		
.		
23	KF	Start address for word module at slot 23

Value ranges for start addresses: 0 ... 16 383
KF: fixed point number

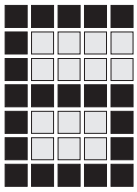
The parameters are entered in DB 005 using the programming device software, or if blocks OB 020 or OB 022 are used the parameters may be entered here.

Inputting in DB005:

- Call up data block DB 005; a table appears for entering the values
- Enter parameters
- Save data block

Inputting in OB 020 or OB 022 (OB 020 and OB 022 are initialisation blocks for a new start, refer to chapter 7):

- Select data block A DB 005
- Load start address for the first word module as a fixed point number in the accumulator
  e.g. L KF 1000



- Write start address for word module at 1 in DW
  e.g. T DW 1

.

.

.

- Load start address for last word module in the accumulator
- Write start address for last word module in DW

True layout configuration

The CPU detects automatically the true layout configuration of the modules. The result of the configuration test can be read in DB 000. The true layout configuration indicates on which slot the standard module is located. The slots are numbered from 0 ... 8 on the base module rack. If one or two expansion racks are connected the slots are numbered 0 ... 7 on the base module rack, 8 ... 15 on the first expansion rack and 16 ... 23 on the second one.

INFORMATION
The true layout configuration only indicates the standard modules and not failsafe modules!

Each module is assigned a code (see the description of the modules). This code is stored in data words 20 ... 43 in **DB 000**:

Data word	Code	Meaning
20	KH	Code for module at slot 0
21	KH	Code for module at slot 1
22	KH	Code for module at slot 2
.		
.		
.		
43	KH	Code for module at slot 23

KH: Hexadecimal number

Switch on configuration test

During the configuration test the true layout configuraton is compared to the set layout configuration (DB 004 DW 020 ... 043). If a slot does not correspond with the set layout configuration, the error-OB, OB 023 is called up. If OB 023 is not present, the standard section goes into a STOP condition and indicates a configuration error “S-05”. Otherwise OB 023 is processed.

A configuration test can be carried out after start-up (STOP - RUN) or in cycles. Enter when the configuration test should start in data word DW 018 or DW 019 in data word **DB 004**.

Data word	Code	Meaning
18	KH	= 0: Configuration test switched on in cycles = 1: Configuration test switched off in cycles*
19	KH	= 0: Configuration test switched on on start-up* = 1: Configuration test switched off on start-up

KH: Hexadecimal number
* Preset

Output to display

Hexadecimal figures can be output to the CPU display. The characters must be located in data block **DB 004** DW 200 . The display is updated when SB 254 is called up with function FUNK = 32.
System error messages and FS error messages are displayed as a priority.

Setting the parameters:

	Input	Output	Meaning
SB 254	FUNK = 32	 ERG = 1	Function code for “Output to display” No error
DB 004	DW200 = xxxx DW200 = FFFF		Show xxxx characters on the display Clear display*

* Failsafe errors **cannot** be cleared.

The current text shown on the display is stored in data block DB 000, DW 015.
DW 014 describes the type of message, but in a coded form.

Setting the real-time clock

The times are entered in **DB 004**, DW 200 ... DW 203 to set the real-time clock. Each time SB 254 is called up with the function FUNK=12 the real-time clock is re-initialised, set and the current time copied in the data block DB 000, DW 000 ... DW 003.

Setting the parameters:

	Input	Output	Meaning
SB 254	FUNK = 12	 ERG = 1	Function code for “Set real-time clock” No error
DB 004	 DW 201 DW 202 DW 203	DW200	Year 0 ... 0099 DL: Month 1 ... 12 DR: Day 1 ... 31 DL: Hour 0 ... 23 DR: Minute 0 ... 59 DL: Second 0 ... 59 DR: 0

If entries do not lie within the range stated in DB 004, the value 0 will be entered instead.

Reading failsafe data blocks

The function FUNK = 36 enables read access to the failsafe data blocks.

	Input	Output	Meaning
SB 254 DB 004	FUNK = 36 DW200		 Number of the failsafe data blocks to be read

If the failsafe data block does not exist, the standard section switches to a STOP condition.

INFORMATION
Failsafe blocks can only be selected if the FS section is in a RUN condition (FS status flags “Communication with the failsafe section”).

Control / force

Forcing means that an input or an output is fixed at a predefined condition regardless of the application program.
Flags, data bytes, data words, timers and counters on the other hand, are set only once at the predefined value, i.e. controlled unlike the inputs and the outputs, these can be amended by the application program.

Permitted for control/force:

Operand	Addresses	Force	Control
Inputs	E 00.00 ... E 23.31 EB 00.00 ... EB 23.24 EW 00.00 ... EW 23.16	X	
Outputs	A 00.00 ... A 23.31 AB 00.00 ... AB 23.24 AW 00.00 ... AW 23.16	X	
Data bytes/ Data words (DB 10..DB 255)	DW 0 ... DW 1023 DL 0 ... DL 1023 DR 0 ... DR 1023		X
Flags	M 00.00 ... M 63.31		X
Timers	T 00 ... T 63		X
Counters	ZW 00 ... ZW 63		X
Addressable modules	XW0 ... XW16383		X

User interface

The user interface is an RS 232 interface with handshake lines. It is available to the user for non-safety data transfer within the failsafe or standard section.

INFORMATION
The user interface can be used by **either** the standard section **or** the failsafe section.
If the interface is configured for the failsafe section, it can only be used for the standard section if:

- The standard section is cleared and
- The PSS operating voltage is switched off and on again.

Using a PAP interface adapter (RS 232 -> RS 485) and the ISI Protocol (see next page) a network can be created with a maximum 32 subscribers (1 Master, 31 Slaves) and a cable length of maximum of 1,200 m.

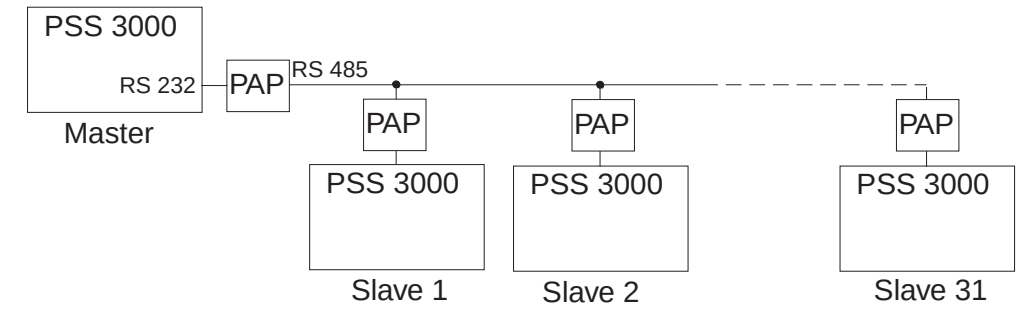


Fig. 6-2: Network with 32 subscribers

Handshake

The interface has the following handshake signals:

- RTS
- CTS
- DTR
- DSR

If data transfer occurs without **handshake,** input signals DSR and CTS are not considered. Output signals DTR and RTS are used. The connection cable for data transfer without handshake is shown in the Appendix.

If data transfer occurs **with handshake,** the output signals are used to control the partner and the input signals CTS and DTS to synchronise with the partner. The signals have the following meanings:

- Output signal DTR:
  The output signal indicates to the partner if the user interface is ready to receive data:
  - DTR = 1: User interface ready to receive data
  - DTR = 0: Not ready to receive as the receive buffer is full or the receive data is being processed (refer to layout of DB 006, DW 012).
- Output signal RTS:
  The output signal indicates the user interface is ready to send:
  - RTS = 0: Rest position
  - RTS = 1: Immediately before the beginning of a send cycle and during a send operation; when the send is complete the RTS is set to 0

  The output signal usually switches the interface adapter to send mode.
- Input signal DSR:
  The input signal indicates when the partner is ready to receive data. The safety system polls the inputs' status before a send cycle starts:
  - DSR =1: Start send process
  - DSR = 0: Partner not ready to receive, postpone send operation until DSR = 1.
- Input signal CTS:
  An interface adapter usually uses this input signal to indicate if it is switched to send mode. The safety system interrogates/polls this input status before the send cycle begins:
  - CTS = 1: Start send process
  - CTS = 0: Partner not ready to receive data, postpone send operation until CTS = 1.

The connection cable for data transfer with handshake is shown in the Appendix.

Tansferring without protocol

If data is transferred without protocol, it will be sent unamended from the send buffer via the user interface.
The received data will be written to the receive buffer block by block and unamended. When the timeout period elapses this will signal the end of the block.
The timeout period for receive is designed as an aid to detect the end of a data block when transferring without protocol. A timer is started when a character is received. Each additional character received will trigger the timer again. If the timer elapses, the interface interprets this as the end of a data block and writes the received data to the receive buffer. The timeout period can be configured (refer to the Layout of DB 006, DW 010).
When a data block end is detected the output signal DTR is set to 0.

Transferring with ISI Protocol

If transfer occurs with ISI-Protocol, the data from the send buffer will be given a protocol structure:

LF (0A hex.)	Telegram start	Tele-gram header
Slave-Adresse ID	Slave address 1 ... 31, ID-Field 0 ...3	
Data length low	Number of data bytes, Low-Byte	
Data length high	Number of data bytes , High-Byte	
1st piece of data Last piece of data (DL)	Max. 2042 Byte	Data
CRC	Check sum	Telegram end
CR (0D hex.)	End of telegram	

Any protocols received are searched for protocol errors. Error-free telegrams are written to the receive buffer after the protocol structure has been removed.

Operation

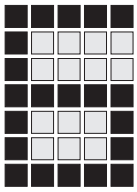
The ISI-Protocol operates according to the Master-Slave-Principle. Only one subscriber may be defined as the master. The addressed slave must respond after every send operation from the master within the specified timeout period (see the layout of DB 006, DW 009). A slave only sends data when a request has been made.

Calculating the check sum CRC:
CRC = 0 - (ID-Byte of Slave-Address
+ Data length low
+ Data length high
+ Sum of all data bytes)
First data byte is in DR 3.

Please note when sending with ISI-Protocol:

- The value “3” must be entered in data word DW 005, DB 006 (8 data bits, see “configuration”).
- If the safety system is the slave:
  - A telegram cannot be sent if the master is not ready to receive, the master will request the response again (no send error).
  - Data word DW 003 from data block DB 007 is the first piece of send data, DW 002 is reserved for the Slave Address (see “Send data”).
  - Only change the send buffer (DB 007) after a telegram has been received from the master. If the master requests the last telegram again because of a receive error, the safety system automatically repeats the telegram stored in the send buffer.
- If the safety system is the master:

- The slave address must be entered in data word DW 002 from data block Db 007, data word DW 003 contains the first piece of send data (see “Send data”).
- A send cycle is only complete when the addressed slave has responded and the receipt of the response or a receive error has been acknowledged. If the slave response is not acknowledged, the poll status results in FUNK = 4 ERG = 2 (see “Send data”). The receive status must be polled to detect whether the slave has responded.



- If the slave does not respond error-free within the permitted timeout period, the safety system will repeat the telegram 3 times. If the response is still not error-free, a receive error is given.

ID-Management

If the safety system is the master:
After configuring the interface a telegram with ID=0 is sent. After an error-free repsonse from the slave, the ID-byte for this slave is increased by 1. If ID = 3 and the response to the telegram is error-free, the ID-byte for this slave is reset to 1.
If the slave response contains an error, the safety system repeats the telegram 3 times using the identical ID-byte. If there is still no response, the ID-byte for the slave is immediately set to 0. The next telegram sent to this slave will have the ID = 0.

If the safety system is the slave:
An addressed slave always responds with the ID-byte received. If the ID-byte remains the same between two telegrams received, this means it was a repeat prompt for the last telegram sent. The repeat prompt is automatically answered by the safety system. If the ID-byte = 0, there will be a request to initialise the slave.

Operating the user interface

Configuration

The interface must be configured before data transfer takes place. The functions 0/1/2 of SB 254 are available.

	Input	Output	Meaning
SB 254	FUNK = 0 FUNK = 1 FUNK = 2		Poll status: configuration Configure Acknowledge configuration error
		ERG = 1 ERG = 2 ERG = 16 ERG = 32	Interface ready for operation Interface being configured Configuration error Error acknowledge being processed

Data block DB 006 is used as the configuration data block. Enter the configuration data for the interface in DW 002 .... DW 012.

Layout of the configuration data block:

DW 000: Reserved
DW 001: Error code with configuration error ERG = 16; contains the number of the data word having the invalid value. If the contents of DW 001 = "FFF0", the user interface is assigned to the FS section. DW 001 has read only access.

DW 002: Baud rate, default setting: 9600

Value	0	1	2	3	4	5	6	7
Baud/s	150	300	600	1200	2400	4800	9600	19200

DW 003: Parity bit, default setting: 2

Value	0	1	2
Parity bit	none	odd	even

DW 004: Stop bit, default setting: 0

Value	0	1	2
Stop bit	1	1.5	2

DW 005: Data bit, default setting: 3

Value	0	1	2	3
Data bit	5	6	7	8

DW 006: Handshake, default setting: 1

Value	0	1
Handshake	No	Yes

DW 007: ISI-Protocol, default setting: 0

Value	0	1
Protocol	No	Yes

DW 008: Transfer with ISI-protocol only
CPU: Master or slave; default setting: 0

Value	0	1 ... 31
Master/Slave	CPU: Master	Slave-Address

DW 009: Transfer with ISI protocol only
Timeout period for slave response; default setting: 0

Value	0	1 ... 65535
Timeout	None	Timeout in ms

DW 010: Timeout period during receive in ms; default setting: 15
Value range: 1 ... 65535

DW 011: Transfer with handshake only
Timeout period during send, default setting: 5000

Value	0	1 ... 65535
Timeout	Infinite	Timeout in ms

DW 012: DTR control
The receive buffer has space for 2044 characters. To avoid losing the characters, the send operation can be stopped before the receive buffer is full. The control line is set to 0. The number of characters before 2044 at which the send operation should be stopped is entered in DW 012 many characters before 2044 should the send operation be stopped is entered in DW 012.
Value range: 0 ... 2044, default setting: 3
Example: DW 012 = 5, i.e. after 2039 characters DTR is set to 0 and the partner is prompted to end the send operation.

Sequence of a configuration request:

- Poll status with FUNK = 0,
  A new configuration may only be set in DB 006 if the result is ERG = 1 or ERG = 16.
- Configure by calling up with FUNK = 1, if the configuration data in DB 006 is complete; during configuration the poll status gives FUNK = 0 gives a result ERG = 2; after an error-free configuration the result is ERG=1.

If an error occurs during configuration, the result of the status poll is ERG = 16. Data block DB 006, in DW 001 states the setting where the error has occurred. Rectify the error and acknowledge it using FUNK = 2 or a new configuration command FUNK = 1. While the acknowledgement is processed the result of the status poll is ERG = 32. If processing has ended then ERG = 1.

INFORMATION
With each STOP-RUN process the user interface is configured using the default values and DB 006 initialised with the same values. To configure the interface using user-specific values it must be re-initialised after every STOP-RUN process (e.g. via OB 020 or OB 022).
Configuration resets the interface and the send and receive buffers are cleared.

Send data

	Input	Output	Meaning
SB 254	FUNK = 4 FUNK = 5 FUNK = 6		Poll status: send Send Acknowledge send error
		ERG = 1 ERG = 2 ERG = 16 ERG = 32	Interface ready for operation Telegram is being sent Send error Error acknowledge is being processed
DB 007	DW 000 DW 002 DW 003-DW 1023	 DW 001 	Number of bytes to be sent Error code on send error Data word / Slave-Address* Send data

* DW 002 contains either the 1st piece of send data (without protocol) or a slave address (with protocol).

Layout of send block DB 007:

DW 000: Enter the number of bytes to be sent

DW 001: Contains the error code if ERG = 16

- = 0: Timeout period exceeded during send (partner not ready to receive)
- = 16: Number of bytes to be sent is too big
- = 17: Send command at the moment not possible e.g. as a telegram is being sent
- = 18: Interface not yet configured
- = FFF0: User interface is assigned to the FS section
- = FFFF: No error found

DW 002: with ISI protocol, if the safety system is the master:
enter the slave address
without ISI-protocol: 1st send data word

DW 003 ... DW 1023: Transfer the send data to these data words
Send sequence without ISI protocol: DR2, DL2, DR3, DL3, ...
Send sequence with ISI protocol: DR3, DL3, DR4, DL4, ...

Send request sequence:

- Poll status with FUNK = 4,
  Only when the result is ERG = 1 or ERG = 16 can the send data be stored in DB 007, DW 002 ... DW 1023 and the number of bytes to be sent stored in DW 000.
- Call up FUNK = 5, if the send data in DB 007 is complete; during the send operation a poll status with FUNK = 4 gives a result ERG = 2; after a telegram is sent error-free the result is ERG = 1.

If an error occurs during the send operation, the result of the status poll is ERG = 16. In data block DB 007, DW 001 states why the error occurred. In order to send the next telegram, rectify the error and acknowledge using the function FUNK = 6 or FUNK = 4. During acknowledgement processing, the poll status is given as result ERG = 32. If processing has ended, ERG = 1.

Receive data

	Input	Output	Configuration
SB254	FUNK = 8 FUNK = 10 FUNK = 11		Poll status: Receive Acknowledge receive error Acknowledge receipt
		ERG = 1 ERG = 2 ERG = 4 ERG = 8 ERG = 16 ERG = 32	No data received Data being received Receive telegram complete Process receive acknowledgement Receive error Error acknowledge being processed
DB008		DW 000 DW 001 DW 002 DW 003 ... DW 1023	Number of received bytes Error coding on receive error Data word/not used (ISI Protocol) Receive data

Layout of receive block DB 008:
DW000: contains the number of received bytes
DW001: contains the error code, if ERG = 16

= 0: Timeout period is exceeded during send operation *
= 1: Parity error
= 2: "Break" received
= 3: Stop bit error
= 8: Overrun error; characters have gone missing, e.g. because the last received telegram was not acknowledged in time
= 9: Overrun of the receive buffer
= 10: Slave does not respond in time *
= 11: Number of received bytes not correct *
= 12: CRC error*
= 13: ID error*
= 14: Incorrect slave responses*
= 15: End ID is missing or telegram end contains too few characters *
= 17: Operation not permitted at this time
= 18: Interface not yet configured
= 19: Telegram header contains too few characters *
= FFF0: User interface is assigned to the FS section
= FFFF: No error

* only with ISI Protocol

DW 002: contains first received data word if transfer is without ISI Protocol; when transferring with ISI-Protocol, DW 002 has no significance

DW 003 ... DW 1023: Contains the receive data

Receive sequence without ISI Protocol: DR2, DL2, DR3, DL3, ...

Receive sequence with ISI Protocol: DR3, DL3, DR4, DL4, ...

Sequence of a receive prompt:

- Poll status with FUNK = 8,
  Result ERG = 1 means that the interface received no data, ERG = 2 means that data is being received. The data is stored in DB 008.
- Poll status with FUNK = 8,
  Only if all data is received complete, the status poll gives a result ERG = 4; only then can the received data be processed in DB 008. A telegram is complete if a character is no longer received within the timeout period. This will be detected during the next cycle change or next stauts poll.
- Acknowledging receipt with FUNK = 11 if the receive data processing is finished; during error acknowledgement processing the status poll gives a result as ERG = 8; after an error-free acknowledgement the result is ERG = 1.

If an error occurs during a receive operation, the result of a status poll is ERG = 16. In data block DB 008, DW 001 indicates which error has occurred. Rectify the error and acknowlege this using function FUNK = 10 or FUNK = 11. During error acknowledgement processing the status poll gives a result ERG = 32. If processing is finished, then ERG = 1.
So that the application program detects whether the data was received via the interface, the receive status must be checked regularly.
The first received byte is in DR 3 with any transfer using ISI Protocol.
DW 002 is not occupied.

If the safety system is Master and the timeout period for sending is exceeded, this will be given as a receive error. The receive status must be polled while waiting for the response from the slave. The next telegram

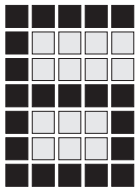
can only be sent if receipt or the receive error was ackowledged.
If the safety system is the slave, faulty telegams or telegrams without an addressed slave are discarded by the operating system without a message. Repeat requests (ID-Byte remains unchanged) are answered automatically by the safety system.

Communication with the failsafe (FS) section

Two types of flags are available for communicating between the failsafe and the standard section:

- Communication flags M 100.00 ... 104.31
  These can both be written and read by the standard application program. The failsafe section only has read access to these flags. The flags are freely available to the user.
- Status flags
  These have special functions:
  - Fixed flags influence the result of logic operation (RLO). The failsafe and standard sections have read access only.
  - Arithmetic flags set the operating system during an arithmetic operation. The standard section has read access only.
  - Status flags FS and ST provide information on the status of the system. The failsafe and standard section have read access only.
  - Status flags for indirect addressing are used as indicators during indirect addressing. The failsafe and standard section have both read and write access.

Flag type	Key	ST access
Fixed flags		
M110.00	= 0	read
M110.01	= 1	read
Arithmetic flags		
M111.00	=1, when Carry-Flag is set via arithmetic operation	read
M 111.01	= 1, when Overflow-Flag is set via arithmetic operation	read
M 111.02	= 1, when Zero-Flag is set via arithmetic operation	read
M 111.03	= 1, when Sign-Flag is set via arithmetic operation	read



Flag type	Key	ST access
ST status flags		
M112.00	= 0, when ST is in STOP-condition = 1, when ST is in RUN-condition	read
M112.01	= 1, when error in ST	read
M112.02	= 1, when ST is in STOP-condition following a command	read
M112.03	= 1 after transfer from STOP to RUN (for one cycle)	read
M112.04	= 1 after transfer from "Voltage off" to RUN (for one cycle)	read
M112.05	= 1, when a reset was carried out in ST (for one cycle)	read
FS status flags		
M 113.00	=0, when FS is in STOP-condition = 1, when FS is in RUN-condition	read
M 113.01	= 1, when error in FS	read
M 113.02	= 1, when FS is in STOP-condition following a command	read
M 113.03	= 1, after transfer from STOP to RUN (for one cycle)	read
M 113.04	= 1, after transfer from voltage off to RUN (for one cycle)	read
Status flags indirect addressing		
MW 114.00 ... MW 114.16	Indicator for indirect addressing	read/write

INFORMATION
The failsafe section **cannot** access the operands in the standard section. Communicaton is only possible via the flags described above.
The standard section has read access to the PIO, PII, flags, data blocks, timers and counters in the failsafe section.

Operating status and changes to operation

This chapter describes the various stages of operation on a PSS system. It also describes the changes in conditions that occur, what happens during these changes and how this can be triggered.

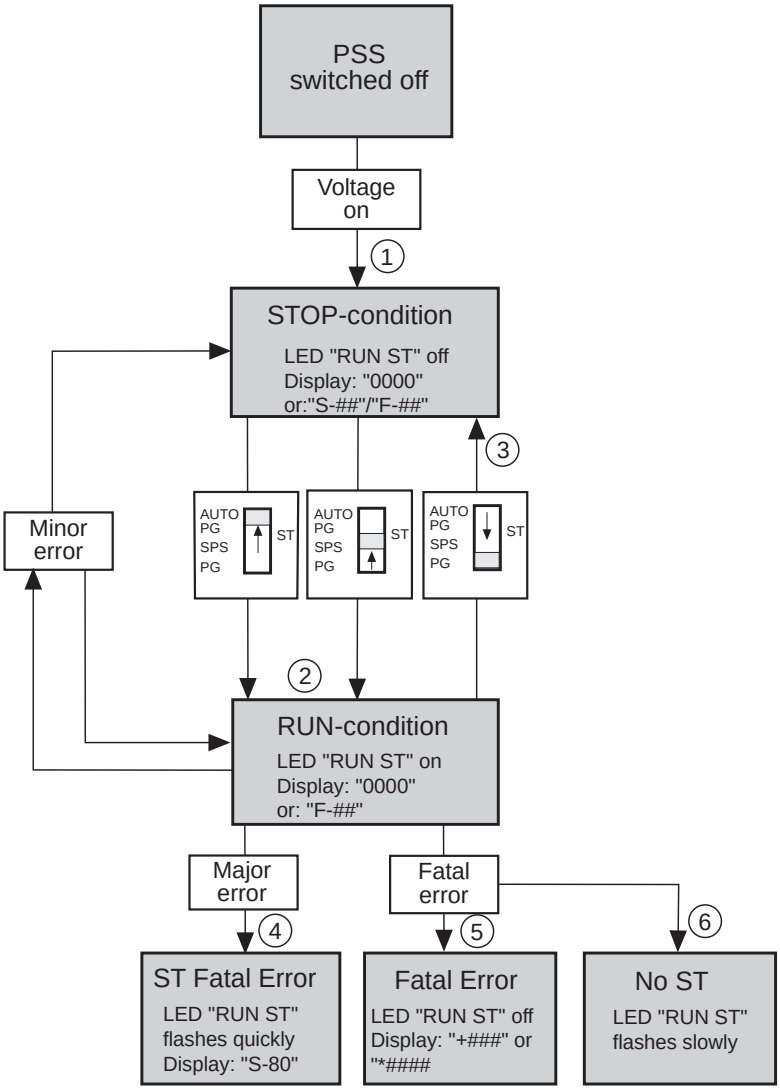


Fig. 6-3: PSS operating status

The numbers indicate a change in status, all of which are described on the following pages.

Operation

STOP

In a STOP-condition:

- The FS application program is not processed
- The ST application program continues and can read the FS-PII and FS-PIO
- All programming device functions are available.

RUN

In a RUN-condition:

- The FS application program is processed
- The ST application program continues and can read the FS-PII and FS-PIO
- All programming device functions are available (Exceptions: Load program and clear program).

"No ST"

"No ST" means:

- The standard section is defective
- The standard application program does not run
- The failsafe application program continues to be processed unamended
- The programming device functions are restricted to read access only.

"Fatal Error"

After a fatal error the PSS goes into a "Fatal Error" condition:

- FS and ST section are inoperable
- Communication is not possible with the programming device

If the PSS goes into this condition:

- Establish the conditions under which the errors occurred
- Write down the error message displayed.
- Contact Pilz.

"ST Fatal Error"

If a major error occurs:

- The standard application program is not processed
- The failsafe application program continues to be processed unamended
- The standard section, including the ST operating system, is not available.

Change in the PSS operating condition

Switch on voltage (1)

Once the voltage is switched on the CPU is re-initialised. Data must not be able to be lost. To ensure data is not lost, the system carries out the following tasks:

- Digital output modules are switched off
- Checks the program cartridge and creates a contents directory
- Reads from the clock and enters the current time in DB 000
- Initialises the ST status flags and sets the start flag M 112.04 = 1
- Checks whether the non-volatile RAM-memory when switched off was amended; if yes it carries out a reset (see "Start-up procedure")
- Sets the ST communication flags to = 0
- Writes the process images with 0
- Switches off the dynamic program display
- Switches off the variable display
- Transfers to a RUN-condition when the selector switch is in the position "SPS" or "AUTO PG" or to a STOP-condition,when the selector switch is in the position "PG".

Change from STOP to RUN ②

A change from a STOP to RUN is triggered by:

- Changing the selector switch from "PG" to "SPS" or "AUTO PG" or
- Re-applying the voltage if the selector switch is in the position "SPS" or "AUTO PG" or
- Start command from the PG

The CPU carries out the following actions:

- Switches on the LED "RUN ST"
- Sets the ST start-up flag M112.03 = 1
- Sets the ST error flag M112.01 = 0
- Sets the ST-STOP-command flag M112.02 = 0
- Writes the process images with 0
- Initialises the ST counters with 0
- Stops the ST timers
- Initialises the user interface with the base settings
- If expansion modules are connected and the mains voltage is switched on to all expansion modules, then the CPU:
  - Initialises the expansion modules, if an error occurs during initialisation the system switches to a STOP condition
  - Enters the true configuration in DB 000
  - checks the application program, if an error occurs the system switches to a STOP condition
  - If OB 022 is loaded or the ST-RAM-memory is corrupted or the PLC is started with the selector switch while the error stack button is held down, the CPU carries out a reset:
  - Runs OB 022 or OB 020
  - If a flag or the configuration test is set, carries out a configuration test; if an error occurs the system switches to a STOP condition
  - Initialises the word modules; if an error occurs the system switches to a STOP condition.
- If there is no mains voltage applied to the expansion modules after 3 s, the system switches to a STOP condition

INFORMATION

After OB 001 has been run through the reset flag (M 112.05), start-up flag (M 112.03), new start flag (M 112.04) are reset and the ST-RUN-flag (M112.00) is set.

Change from RUN to STOP (3)

If the ST selector switch is set to PG or because of an error reaction the system switches to a STOP-condition:

- All outputs are switched off
- PIO written with 0
- Status flags M112.00 = 0 or, if STP-command: M112.02 = 1 of if an error: M112.01 = 1

Change from RUN to "ST Fatal Error" (4)

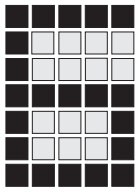
The standard section, including the ST operating system, is not available

Change from RUN to "Fatal Error" (5)

All outputs (FS and ST) are switched off. The FS and the ST application program are stopped.

Change from RUN to "No ST" (6)

A switch to "No ST" will stop the ST application program.



Notes

Start-up Procedure

Initial start-up

Hardware requirements:

- Power supply: supply voltage connected (see PS operating manual or the PSS Overview)
- Input and output modules: supply voltage connected (24 VDC)
- Correct module rack configuration: first slot occupied by the power supply and the second by CPU module.

Software requirement:

- Executable application program must be available
- Set configuration entered in DB 004
- Word modules: enter addresses in DB 005

Initial set-up:

- Set “ST” selector switch to “PG”
- Switch on power supply (Position "I")
  Reaction: LED "Power" on the power supply and CPU module lights up.
  CPU carries out a self-check, display shows: ****
- If the self check is successful the display will show: 0000
- Transfer program (see “Programming Device” manual)
  - Connect the serial interface on the computer (programming device) to the PG-interface of the CPU module
  - Activate the “Online-Menu”on the programming software
  - We recommend you clear the program memory before transferring the program
  - Transfer the program; with a Flash-EPROM program memory the FS section must be in a STOP condition
- Set the selector switch "ST" to "SPS" or "AUTO PG" or start the program using the programming device.
  Reaction: The program is run.
  LED "RUN ST" lights up.
  If a configuration error occurs, this could be due to the set layout configuration in the program used during the in-house function test (Pilz).
  A general reset can be used to remedy the error (set selector switch

"ST" to "PG", and then press the "error stack" button at the same time as setting the selector switch from "PG" to "SPS").

Error messages are described in Chapter 8.

Reset after a fault

If a hardware or software error occurs, the safety system will immediately switch to a safe condition. The fault will be shown on the display (see Chapter 8 for details of the error messages) and the LED "RUN ST" will either flash or go out:

- LED “RUN ST” flashes: a fatal error has occurred and the standard section is defective (the FS section continues unaffected). Contact Pilz.
- LED "RUN ST" flashes slowly: a major error has occurred. The number on the display provides information on the error.
  Reset:
  - Rectify the error, if necessary use the porgramming device to locate the error in the error stack
  - Switch off the power supply and switch on again
    If the error is removed, the CPU goes to RUN after the self-check, the program will run again.
- LED "RUN ST" goes out and the display shows “S-xx”: an error has occurred.

  Reset:
  - Rectify the error. If necessary use the programming device to locate the error in the error stack
  - Set “ST” selector switch to “PG” and then to "SPS" or "AUTO PG, or start using the programming device. The program will run again.
- LED "RUN ST" goes out and the display shows error number "+xxx" or ∗xxx": a fatal error has occurred. The system is defective. Contact Pilz.

Reset

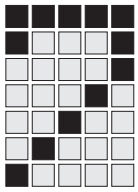
The CPU uses battery-buffered RAM chips as the system memory. These contain information on the current operating status. If the battery is removed while mains voltage is switched off, the information will be lost. The CPU re-initialises the system memory during the start-up procedure. This initialisation procedure is known as reset.

Reset can be carried out either manually or automatically:

- Manual: Keep the error stack button pressed while switching the operating mode selector switch from "PG" to "SPS"
- Automatic: When the mains voltage is switched off the CPU creates a check sum of the contents of the system memory. When the mains voltage is applied, the check sum is recalculated and compared with the one stored. If the check sums are different, a reset is triggered.
- Automatic: If OB 022 is loaded the CPU carries out a reset with every change from STOP to RUN. OB 022 can contain user-specific initialisation which can be carried out in addition to the initialisation through the reset.

The CPU carries out the following steps during reset:

- Enters the reset in the error stack
- Sets the reset flag M112.05
- Stops and sets the timers and counters to 0
- Sets the ST flags 00.00 ... 63.31 to 0
- Sets the ST communication flags 100.0 ... 104.31 to 0
- Sets the ST address indicator flags 114.00 ... 114.31 to 0
- Writes the process image with 0
- Switches off the supply to the digital output modules
- Initialises the real-time clock (time: 0:0 h)
- Sets the ST scan time in DB 000 = 0
- Enters the true layout configuration in DB 000 and DB 004
- Switches off the configuration test within the cycle



- Switches on the configuration test within the cycle
- Writes the parameters used when calling up SB 254 in DB 004 with 0
- Generates the data blocks loaded by the user and assigns the default values edited within the programming device to the blocks, provided the valid application program is loaded.
- Switches off the variable and the dynamic program display and initialises the tables with 0.

INFORMATION

A reset does not clear the program cartridge. RAM cartridges must be cleared using the programming device.

Error Diagnostics and Correction

Managing errors

The safety system constantly checks the hardware and software during the program cycle. Any errors detected will trigger the following:

- The error is classified into an error class. Each error class is assigned an error code.
- The error is displayed on the CPU-display
- The error is entered in the error stack, together with its parameters.

The reaction of the safety system to an error depends on the error class:

- In the event of a fatal error the standard application program is interrupted or not activated, and communication with the programming device is no longer possible
- In the event of major errors the standard application program is interrupted or not activated, a re-start is only possible by switching off the power supply and switching on again
- Minor errors are errors in the application program which are detected during the program cycle or by the operating system. Minor errors include:
  - Exceeding address range
  - Attempting to write a write-protected data block

  The reaction to a minor error is described in an error organisation block. If an error-OB is present in the program it will be processed. If one is not present the system will go into a STOP-condition immediately. Possible reactions are: switch to emergency mode, start a time-controlled switch off procedure, error messages sent to display.
- Message errors do not influence the program cycle, e.g. depleted batteries.

The FS and ST sections operate independently. Even if errors occur in the FS section the ST section will continue to run. Only if a fatal error occurs in the FS section and also in the ST section will communication with the programming device be interrupted.

Error Diagnostics and Correction

The error stack can contain a maximum of 16 errors. It occupies data words DW 085 ... DW 148 in DB 000 system data block. Each error occupies 4 words:

DW	Configuration
084	Indicates current error
085	Error class of 1st error
086	Error number of 1st error
087	Location of 1st error
088	Error parameters for 1st error
089 ... 092	Description of 2nd error
093 ... 096	Description of 3rd error
097 ... 100	Description of 4th error
101 ... 104	Description of 5th error
105 ... 108	Description of 6th error
109 ... 112	Description of 7th error
113 ... 116	Description of 8th error
117 ... 120	Description of 9th error
121 ... 124	Description of 10th error
125 ... 128	Description of 11th error
129 ... 132	Description of 12th error
133 ... 136	Description of 13th error
137 ... 140	Description of 14th error
141 ... 144	Description of 15th error
145 ... 148	Description of 16th error

As the error stack is organised as a loop buffer, data words are accessed via the indicator in DW 084. The indicator always highlights the data word containing the error class of the current error entry.

What the entries mean:

- Error class describes the error in coded form, which is displayed
- Error location describes where the error occurred
- Error number and error parameter contain additional information about the error.

If more than 16 errors occur the first entry will be overwritten.

The error stack contains errors from both the failsafe and standard sections. Errors from the failsafe section are shown on the display starting with the letter “F”. Errors from the standard section start with the letter “S”.

Error display with text messages:

- With the programming device
  Connect the programming device and activate the function “Read error stack”. The current error is displayed as text, together with the error parameters and location.
- With a text display
  Text display, e.g. PX-display. If an error has caused the failsafe section to switch to a STOP condition, the standard section can start up a function block which reads the error stack or the contents of DB 000, DW 085 ... DW 148 and then sends the data to the display.

Error Diagnostics and Correction

Error reaction

The correlation between errors and the PSS operating status is shown in Fig. 8-1.

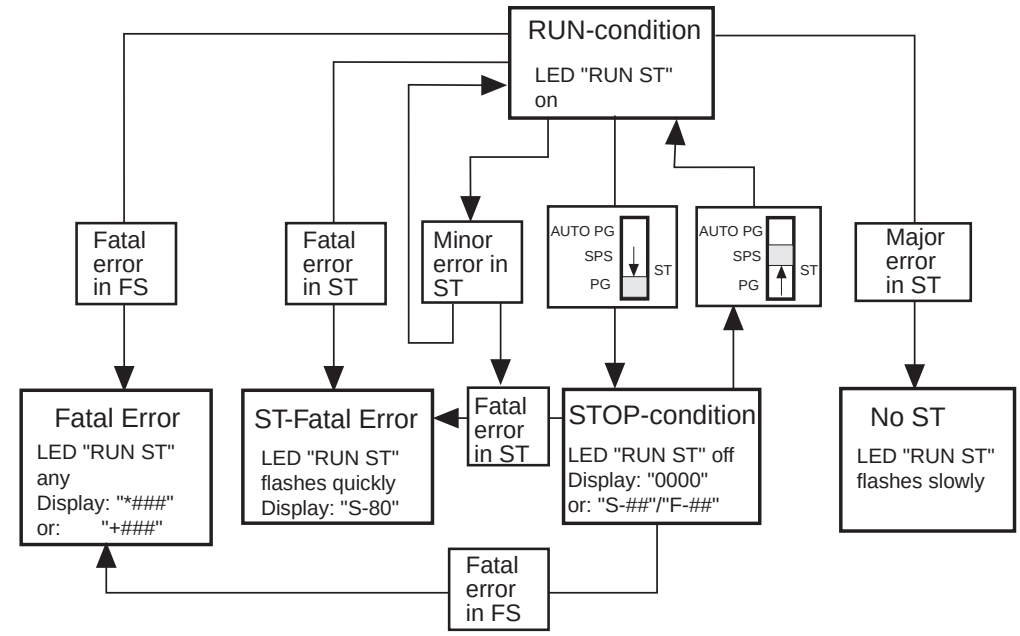
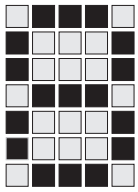


Fig. 8-1: Changes to operating status due to an error or by changing the FS selector switch

If no errors are present in the FS section and the system is in the RUN-condition, "0000" is shown on the display. The system can also be started or stopped via a programming device command.

Error types:

- Minor errors
  The system switches to a STOP-condition (if no other instruction is stated in the error OB) the LED "RUN" is off and the error message "S-##" appears on the display. The failsafe application program continues to run and all functions on the programming device are available.
  - Possible causes:
    Errors in the application program
  - Remedy:
    · Use the programming device to read the error stack
    · Set a trigger condition for error search with the dynamic program display



- Correct the error
- Restart the standard application program by operating the ST selector switch: move the selector switch from “PG” to “SPS” or use the programming device.

• Major errors
The system switches to “No ST”, the “RUN ST” LED flashes (approx. 1 flash per second) and an error message “S-##” appears on the display. The failsafe application program continues to run and the functions on the programming device are restricted to read-only functions.

- Possible causes:
  Irregularities in the system
- Remedy:
  - Use the programming device to read the error stack
  - Switch off the safety system
  - Correct the error
  - Switch on the safety system

• Fatal errors in the ST section
The system switches to “ST- Fatal Error”, the "RUN ST" LED flashes (approx. 2 per second) and an error message “S-80” appears on the display. The standard section is inoperable and it is impossible to communicate with the programming device.

- Possible causes:
  Major system defect
- Remedy:
  It is not normally possible for the user to correct this error
  - Establish the conditions under which the error occurred
  - Read the error stack using the FS programming device and write down the error message shown, or look at the error stack using the “error stack button” and note the error message
  - Contact Pilz

• Fatal errors (see FS System Descritpion)
It is not normally possible for the user to correct this error

- Establish the conditions under which the error occurred
- Write down the error message shown on the display
- Contact Pilz.

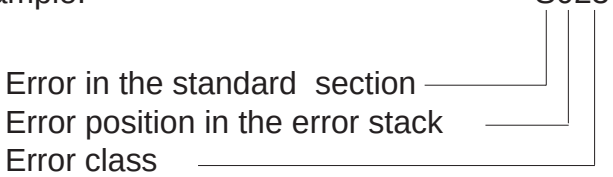
Error Diagnostics and Correction

Display contents of error stack

The errors stored in the error stack can be displayed by pressing the “error stack” button:

- Browse through the error messages: Each time the button is pressed the previous error message is displayed.

Example: S023



- Display error information: by holding down the button when the error is displayed, additional information can be viewed in the following order:
  - Error class, e.g. C = 23 for error class 23
  - Error number, e.g. N = 12 for error number 12
    and where possible
  - Error parameter
  - Error location

Alternatively the contents of the error stack can be displayed using the programmming device, providing the programming device is connected to the safety system. The "ONLINE- display contents of the PLC error stack" menu of the programming device displays the error stack as a list with the following entries:

- Class
  Error class code, see also the FS manual “Error messages”
- P
  Processor which had detected the error (e.g. A, B, C)
- Error description
  Information about he error number
- Error location/error parameter
  Error in the application program: Information about the faulty block

Examples

Error evaluation in the programming device

Error evaluation is carried out online using the programming device. An entry in the error stack is:

Class	P	Error no.	Error description	Location
S-07	A	001:	Program cartridge contains the faulty block	yy xxx

"S-07": CRC error
"A": Processor A had detected the error
"001" : Error number: Program cartridge contains the faulty block
"yy xxx": yy: block type, xxx: block number

Error evaluation on the display

The display shows the following error message:

S-07

Hold down the error stack button. The following messages appear on the display:

S020 - C=20 - N=02

S020:
C=20: error class 20, i.e. error message
N=02: error number 02, i.e. standard section switches to STOP.

Release the error stack button and press again and keep pressed down. The next entry is displayed:

S107 - C=07 - N=01 - AT - yyxxx

S107:
C=07: error class 07, CRC error
N=01: error number 01, i.e. the program cartridge contains the faulty block
AT means: the error location follows
yyxxx error location: block type yy and block number xxx

Error Diagnostics and Correction

Diagnostics

A number of diffferent tools are available for diagnosing errors:

- Variable display: the current condition of the variables (e.g. inputs and outputs) is displayed in tabular form
- Dynamic program display: the current contents of the operands of a program section are displayed.

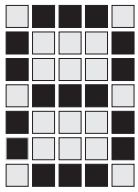
The Programming Manual contains a description of how to use these diagnostic tools.

Display variables

The display of variables enables the status of inputs, outputs, flags, timers, counters and data words to be shown. To achieve this the safety system must be connected to the programming device. The programming device can be used to create a table to establish which variables are to be displayed. The time at which the status of the variables is to be read in is defined by the “trigger” point within the program.
The programming device requests the current data in on-line mode. The moment the “trigger” point is reached, the variables are read in and stored in the table. At the start of the next cycle the search for the trigger point in the program starts again and variables are once more read in and stored. This means that the table is overwritten with the current variable status at each cycle. Data acquisition can be halted via the programming device.

The “trigger” point in the program provides a number of options for you to define precisely the point at which the data is acquired. The following points in the program may be selected:

- At a cycle change (also in STOP condition)
- At the start of a block, without program structure information
  The point in the program is defined by entering the block type and block number.
- At the start of a block with program structure information
  If a block is called up more than once, you will also need to define on which call-up the data is to be acquired. For this reason a program location is flagged within the program structure information.



Both the table and the “trigger” program are stored within non-volatile memory.

Dynamic program display

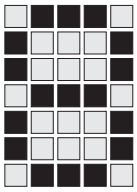
Dynamic program display shows the contents of the indirect addresses, word operands, the accumulator and auxiliary accumulator, plus the status of the contacts and the result of logic operations. The safety system must be connected to a programming device. The programming segment which is to be analysed is displayed on the programing device, which creates a table containing the data to be displayed. The point at which the data is to be read in is defined by the program segment displayed and the block selected.
The programming device requests the current data in on-line mode. Once the displayed program is reached, the programming device reads in the current data and shows it on screen, next to the commands. At the start of the new cycle the search for the program segment begins again, and the information updated.

The ability to select the recording point enables you to define precisely the point at which the data is acquired. The following points in the program may be selected:

- Program segment, without program structure information
  The point in the program is defined by entering the block type and block number
- Program segment with program structure information
  If a block is called up more than once in the application program, you will also need to define on which particular call-up the data is to be acquired. For this reason a program location is flagged within the program structure information.

Both the table and the recording point are stored within the non-volatile memory.

In on-line mode it is possible to switch between variable display and dynamic program display. The display will always return to the point at which it was interrupted.



Notes

Appendix

System data blocks

Data blocks DB 000 and DB 004 ... DB 009 enable communication between the standard application program or the programming device and the operating system. The following tables provide a configuration overview.

DB 000

DB 000 contains general program data. It cannot be amended directly by the application program.

Data word	Coding	Configuration
000	KY	Current year 00,00 00,99
001	KY	Current month 01 ... 12, current data 01 ... 31
002	KY	Hours 00 ... 23, minutes 00 ... 59
003	KY	Seconds 00 ... 59, free
004	KH	Operating system version
005	KH	Hardware version
006	KH	Reserved
007	KH	Current scan time in ms
008	KH	Max. scan time since the start of the FS section in ms
009	KH	Block run time FS application progam
010	KH	Max. block run time FS application program since the last program start in ms
011	KH	Block run time of the ST application program since the last program start in ms
012	KH	Max. block run time ST application program in ms
013	KH	Max. length of self-check in ms
014	KH	Display: 0 = DW 15 contains user data 1 = "F-xx" Failsafe errors 3 = "S-xx" Standard errors
015	KH	Hex. number, the meaning of the number is stated in DW 014
016 ... 019	KH	Reserved
020 ... 043	KH	True configuration layout standard section

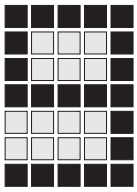
Appendix

Data word	Coding	Key
044 ... 083	KH	Reserved
084	KF	Error stack indicator
085 ... 148	KH	Contents of error stack
149	KH	Internal software version CPU68k
150	KH	Internal software version CPU186
151	KH	Internal software version CPU165
152	KH	Internal hardware version CPU68k
153	KH	Internal hardware version CPU186
154	KH	Internal hardware version CPU165
155	KH	Internal software version of the standard section
156	KH	Internal hardware version of the standard section
157 ... 163	KH	CRC-check sum operating system
164	KH	CRC-check sum FS application program
165 ... 168	KC	Project name FS application program
169	KH	Link data FS application program (year)
170	KY	Link data FS application program (month, day)
171	KY	Link data FS application program (hr, min.)
172 ... 199	KH	Reserved
200 ... 220	KH	Parameters for operating system call-ups in the standard section

DB 004

DB004 contains general program data for the standard section

Data word	Coding	Configuration
000 ... 017		Reserved
018	KH	ST section configuration test within the cycle: 0 = test on within the cycle, 1 = test off
019	KH	ST configuration test at start-up : 0 = test on at start-up, 1 = test off
020 ... 043	KH	Set layout configuration, standard section
044 ... 199	KH	Reserved
200 ... 220	KH	Transfer parameter, standard operating system routine



DB 005

DB 005 configures the word modules.

Data word	Coding	Key
000 ... 023	KF	Start address for word modules on slots 0 ... 23, range 0 ... 16 383
024 ... 063	KH	Reserved

DB 006

DB006 contains the configuration data for the user interface.

Data word	Coding	Key
000	KF	Reserved
001	KF	Error coding: configuration error on the user interface
002	KF	Baud rate: user interface
003	KF	Parity bit: user interface
004	KF	Number of stop bits: user interface
005	KF	Number of data bits: user interface
006	KF	Handshake: user interface
007	KF	ISI Protocol
008	KF	Module address with ISI Protocol
009	KF	Timeout period for Slave-response with ISI-Protocoll (in ms)
010	KF	Timeout period for receive (telegram end) (ms)
011	KF	Timeout period for send (ms)
012	KF	DTR = 0 system receiving

DB 007

DB007 contains the send data of the user interface.

Data word	Coding	Key
000	KF	Number of bytes to be sent
001	KF	Error coding on send error
002	KF	Data word/Slave Address with ISI-Protocol
003 ... 1023	KF	Send data

Appendix

DB 008

DB 008 contains the receive data for the user interface.

Data word	Coding	Key
0	KF	Number of received bytes
1	KF	Error coding on send data
2	KF	Data word/not used with ISI Protocol
3 ... 1023	KF	Receive data

DB 009

DB 009 can be used as remanent memory in the application program. The contents of the data block remains even after reset, it is not intialised.

Data word	Coding	Key
0 ... 63	KF	Not initialised

Organisation blocks

The operating system determines the call-up conditions of organisation block, whose functions are defined by the user. The following tables provide a configuration overview.

Organisation block	Call-up	Application
OB 001	Start of a cycle	Manages the program cycle; must be included in every application program.
OB 019	Error message	Calling up a block which is not available in the program memory
OB 020	At changeover from STOP to RUN	Initialisation with cold start
OB 022	At changeover from STOP to RUN	Initialisation with cold start and reset
OB 023	Configuration error	Incorrect configuration discovered during test
OB 025	Sector error	Sector has been exceeded by indirect addressing, access to data word outside the selected DB, division result does not fit into the accumulator, not permitted operand with a DEF-/ DUF-command
OB 027	Program error (except addressing errors)	Attempts to write to a read-only DB, invalid actual parameter, actual and formal parameters do not agree or an error at the start of a timer
OB 028	At a manual change to a STOP-condition (selector switch "AutoPG/SPS - PG")	Transfers data to the standard section
OB 029	When the battery is depleted	Battery error, battery in the power supply is depleted

Appendix

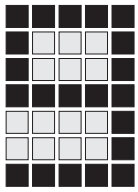
Standard function blocks - operating system call-ups

The following tables contain the function codes for SB 254 which trigger an operating system call-up:

Function code	Function
0 ... 11	Operating the user interface
12	Set real-time clock
32	Output to display
36	Select failsafe data blocks

Configuring the user interface: FUNK = 0/1/2

	Input	Output	Key
SB 254	FUNK = 0 FUNK = 1 FUNK = 2		Poll status: Configuration Configure Acknowledge configuration error
		ERG = 1 ERG = 2 ERG = 16 ERG = 32	Interface ready for operation Interface being configured Configuration error Error acknowledgement being processed
DB 006	DW 000 DW 001 DW 002 DW 003 DW 004 DW 005 DW 006 DW 007 DW 008 DW 009 DW 010 DW 011 DW 012		Reserved Error coding at ERG = 16 Baud rate Parity bit Number of stop bits Number of data bits With/without handshake With/without ISI-Protocol CPU as Master/Slave Timeout for slave Timeout during receive Timeout during send with handshake DTR-control



Send via user interface: FUNK = 4/5/6

	Input	Output	Meaning
SB254	FUNK = 4 FUNK = 5 FUNK = 6		Poll status: Send Send Acknowledge send error
		ERG = 1 ERG = 2 ERG = 16 ERG = 32	Interface ready Telegram being sent Send error Error reset being processed
DB007	DW000 DW002 DW003 ... DW1023	 DW001 	Number of bytes to be sent Error coding on send error Data word/slave address (ISI-Protocol) Send data

Receive via user interface: FUNK = 8/10/11

	Input	Output	Key
SB254	FUNK = 8 FUNK = 10 FUNK = 11		Poll status: Receive Acknowledge receive error Acknowledge receipt
		ERG = 1 ERG = 2 ERG = 4 ERG = 8 ERG = 16 ERG = 32	No data received Data being received Receive telegram complete Process receive acknowledgement Receive error Error acknowledgement being processed
DB008		DW 000 DW 001 DW 002 DW 003 ... DW1023	Number of received bytes Error coding on receive error Data word/not used (ISI Protocol) Receive data

Appendix

Setting the real-time clock: FUNK = 12

	Input	Output	Key
SB254	FUNK = 12	 ERG = 1	Set function code for real-time clock No error
DB004	DW 200 DW 201 DW 202 DW 203		Year 0 ... 0099 DL: Month 1 ... 12 DR: Day 1 ... 31 DL: Hour 0 ... 23 DR: Minutes 0 ... 59 DL: Seconds 0 ... 59 DR: 0

Output to display: FUNK = 32

	Input	Output	Key
SB254	FUNK = 32	 ERG = 1	Function code for “Output to Display” No error
DB004	DW 200 = xxxx DW 200 = FFFF		Show xxxx on the display Clear display

Select failsafe data blocks: FUNK = 36

	Input	Output	Key
SB254 DB004	FUNK = 36 DW200		 Number of failsafe data blocks to be read

User interface connection cable

Fig. 1 and Fig. 2 show the connection cable between 2 user interfaces (RS 232 interfaces) when transferring with or without handshake.

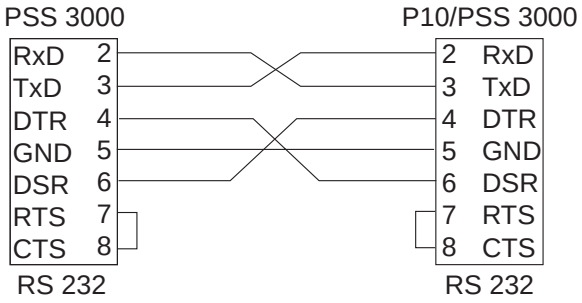


Fig. 1: With handshake

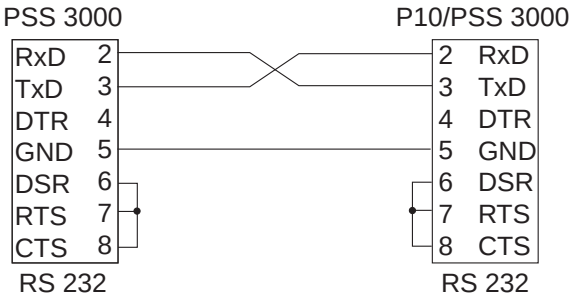


Fig. 2: Without handshake

Appendix

Amendment list

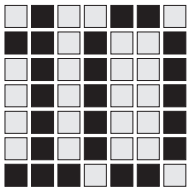
Changes from Version III to Version IV

Old page	New page	Amendment
4-4 5-2	4-5 5-2	Type of PG interface (RS 232 or RS 485) depends on the safety system used. A PAP adapter is not required with an RS 232 interface.
6-14 6-17 6-18	6-14 6-17 6-18	If the contents of data word DW 001 = "FFF0", the user interface is assigned to the FS section.
7-1	7-1	If a configuration error occurs during initial start-up, a general reset may remedy the error.

Index

Index

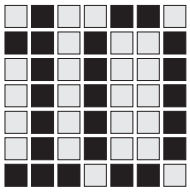
Index



S

T

W



Notes

A

Pilz Ges.m.b.H.
Modecenterstraße 14
1030 Wien
Österreich
Telefon (01) 7 98 62 63-0
Telefax (01) 7 98 62 64
E-Mail: pilz@eunet.at

AUS

Pilz Australia
Industrial Automation LLP.
9/475 Blackburn Road
Mt. Waverley, Melbourne VIC 3149
Australien
Telefon (03) 95 44 63 00
Telefax (03) 95 44 63 11
E-Mail: safety@pilz.com.au

B
L

Pilz Belgium
BC Building
Industriezone III
Industrielaan 4
9320 Erembodegem
Belgien
Telefon (0 53) 83 66 70
Telefax (0 53) 83 89 58
E-Mail: info@pilz.be

BR

Pilz do Brasil Sistemas Eletrônicos
Industriais Ltda.
Rua Ártico, 123 - Jd. do Mar
09726-300
São Bernardo do Campo - SP
Brasilien
Telefon (11) 43 37-12 41
Telefax (11) 43 37-12 42
E-Mail: pilz@pilzbr.com.br

CH

Pilz Industrieelektronik GmbH
Gewerbepark Hintermättli
Postfach 6
5506 Mägenwil
Schweiz
Telefon (0 62) 8 89 79 30
Telefax (0 62) 8 89 79 40
E-Mail: pilz@bluewin.ch

D

Pilz GmbH & Co.
Felix-Wankel-Straße 2
73760 Ostfildern
Deutschland
Telefon (07 11) 34 09-0
Telefax (07 11) 34 09-1 33
E-Mail: pilz.gmbh@pilz.de

DK

Pilz Skandinavien KS
Ellegaardvej 25 L
6400 Sonderborg
Dänemark
Telefon 74 43 63 32
Telefax 74 43 63 42
E-Mail: pilz@pilz.dk

E

Pilz Industrieelektronik S.L.
Edificio Tilma
Avda. Sant Julià 1
08400 Granollers
Spanien
Telefon (93) 8 49 74 33
Telefax (93) 8 49 75 44
E-Mail: pilz@camerdata.es

F

Pilz France Electronic
1, rue Jacob Mayer
BP 12
67037 Strasbourg Cedex
Frankreich
Telefon 03 88 10 40 00
Telefax 03 88 10 80 00
E-Mail : siege@pilz-france.fr

FIN

Pilz Skandinavien KS
Pakilantie 61
00660 Helsinki
Finnland
Telefon (09) 27 09 37 00
Telefax (09) 27 09 37 09
E-Mail: pilz.sk@kolumbus.fi

GB

Pilz Automation Technology
Willow House
Medlicott Close
Oakley Hay Business Park
Corby
Northants NN18 9NF
Großbritannien
Telefon (0 15 36) 46 07 66
Telefax (0 15 36) 46 08 66
E-Mail: sales@pilz.co.uk

I

Pilz Italia srl
Via Meda 2/A
22060 Novedrate (CO)
Italien
Telefon (0 31) 78 95 11
Telefax (0 31) 78 95 55
E-Mail: pilzitalia@tin.it

IRL

Pilz Ireland Industrial Automation
11 Emmet Place
Cork
Irland
Telefon (0 21) 27 48 83
Telefax (0 21) 27 48 94
E-Mail: sales@pilz.ie

J

Pilz Japan Co., Ltd.
Three One Building 701
3-20-5 Shin-Yokohama
Kohoku-ku
Yokohama 222-0033
Japan
Telefon (0 45) 4 71-22 81
Telefax (0 45) 4 71-22 83
E-Mail: pilz@pilz.co.jp

www

www.pilz.com

MEX

Pilz de Mexico S. de R.L. de C.V.
Av. San Ignacio 1079
Col. Jardines de San Ignacio
C.P. 45000
Guadalajara, Jalisco
Mexiko
Telefon (0 13) 1 22 16 81
Telefax (0 13) 6 47 81 85
E-Mail: pilz_msolis@infosel.net.mx

NL

Pilz Nederland
Postbus 186
4130 ED Vianen
Niederlande
Telefon (03 47) 32 04 77
Telefax (03 47) 32 04 85
E-Mail: info@pilz.nl

P

Pilz Industrieelektronik S.L.
Apartado 2028
2706-909 Colares
Portugal
Telefon (01) 9 28 91 09
Telefax (01) 9 28 91 13
E-Mail: pilz@esoterica.pt

PRC

Pilz China Representative Office
Flat F9/F Huijing Building
134 Siyou Xin Malu
Dongshan District
Guangzhou 510600
China
Telefon (0 20) 87 37 16 18
Telefax (0 20) 87 37 35 55
E-Mail: pilzch@public.guangzhou.gd.cn

S

Pilz Skandinavien KS
Energigatan 10 B
43437 Kungsbacka
Schweden
Telefon (03 00) 1 39 90
Telefax (03 00) 3 07 40
E-Mail: pilz@tripnet.se

SGP

Pilz Industrial Automation Pte Ltd.
25 International Business Park
#04-57 German Centre
Singapore 609916
Telefon 5 62 94 84
Telefax 5 62 94 85
E-Mail: pilzsg.com@pacific.net.sg

USA

Pilz LP
24850 Drake Road
Farmington Hills
Michigan 48335
Vereinigte Staaten von Amerika
Telefon (2 48) 4 73-11 33
Telefax (2 48) 4 73-39 97
E-Mail: info@pilzusa.com

pilz

Pilz GmbH & Co.
Felix-Wankel-Straße 2, 73760 Ostfildern, Deutschland
Telefon +49 (7 11) 34 09-0, Telefax +49 (7 11) 34 09-1 33

18 587-12/99 Printed in Germany