Programmable Safety Systems
PSS-Range

FS System Description
Item No. 18 645

The spirit of safety.

Version VII, 01/2001

Contents

Contents

Introduction

This System Description forms part of the PSS system manuals. It explains how the failsafe section of the PSS-range of programmable safety systems function and operate. The 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 processes and the changes which can be made by the operator.

7 Start-up Procedure
Explains the procedure during 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, may 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 have 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.

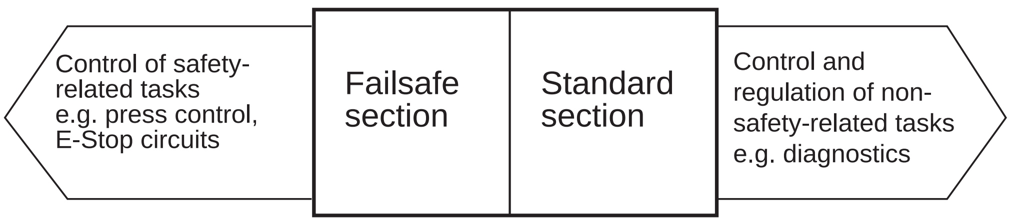


Fig. 2-1: Functions fo the failsafe and standard sections

Hardware

On modular systems different input and output modules can be installed on the 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 systems, the power supply, CPU, bus and periphery modules are incorporated within a single housing.

INFORMATION
The bus for the standard section on the PSS 3000 is available as an option.

Overview

Example of a modular PSS

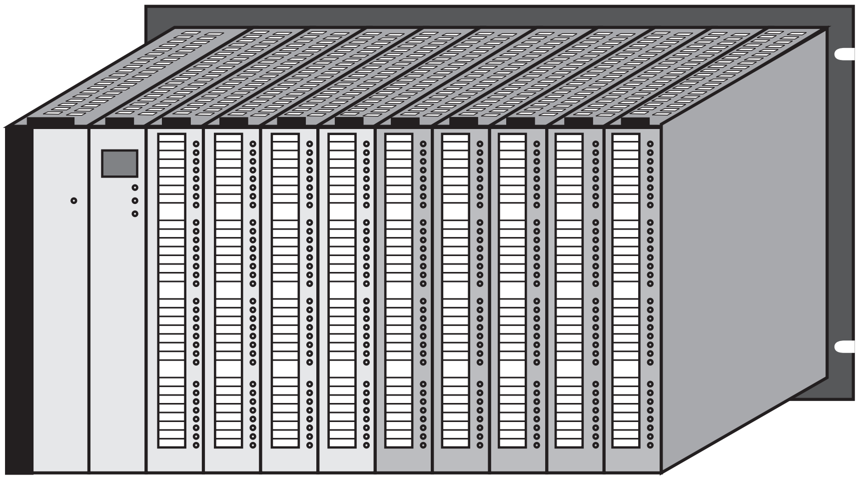


Fig. 2-2: Example of a PSS 3000 layout (left to right): Power supply, CPU, 4 failsafe modules and 5 standard modules

Example of a compact PSS

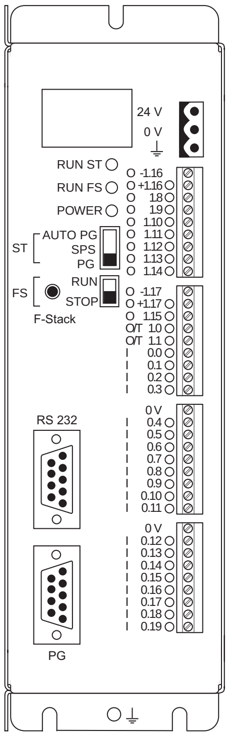


Fig. 2-3: PSS 3032 with integral power supply, CPU and peripheries

Programming

Application programs are created on a PC, the PG programming device, using a special software 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.
Once the program has been installed, parameters need to be set for on-line mode.
The final step is to link up all the blocks into a whole program, which will then be checked and sent 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 engineer who is familiar with the contents of this manual and 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-boards, e.g. exchanging components or additional soldering carried out by the user.
- The safety regulations and EMC-measures described in the “Installation Manual” must be observed at all times.

Safety requirements

Each machine or plant must undergo a risk assessment in accordance with EN 1050. This may be conducted by the manufacturer or operator. The aim is to reduce the risk to below a justifiable risk limit by carrying out a series of relevant measures, which are as follows:

- Measurement and control protection measures
  These prevent personal injury as a priority, plus major damage to plant, machinery or product.
- Non measurement and control protection measures
  These are measures such as closing off dangerous areas, putting up warning signs or covering moving parts. If such measures are taken and the risk is still above the justifiable risk limit, measurement and control protection measures must be taken.

EN 954-1, 11/94 offers another option for risk assessment
The focus of this legislation is to categorise the safety requirements of the safety systems into five categories of control, regardless of the area of technology. These categories include both simple and more complex requirements, such as failsafe operation, redundancy, diversity and/or self-monitoring. The following pages give a summary of some of the sections from the standard.

Intended Application

Overview of categories

Category	Requirement	Risk Assessment	Safety
B	Safety-related sections of the controller must be selected in accordance with the latest technology, and must be able to withstand the anticipated environmental conditions.	An error can lead to the loss of the safety function. Some errors remain undetected.	Through the selection of components and safety principles
1	In addition to category B: Only components and principles whose safety function is guaranteed must be used.	As category B, but with higher reliability of the safety function.	
2	In addition to category B and 1: Safety functions must be tested by the controller at suitable intervals.	An error can lead to the loss of the safety function between checks. The error will be detected through checks.	Through design
3	In addition to category B and 1: Controllers must be designed in such a way that: - a single error does not lead to the loss of the safety function (failsafe) and - a single error will be detected using suitable methods (in accordance with the latest technology).	The safety function is always maintained even if one single error occurs. Some but not all errors are detected. An accumulation of undetected errors can lead to the loss of the safety function.	
4	In addition to category B and 1: Controllers must be designed in such a way that: - a single error in the controller does not lead to the loss of the safety function (failsafe) and - if possible, a single error will be detected at or before the next demand on the safety function or - if possible, an accumulation of errors should not lead to a loss of the safety function	Safety functions are maintained even if one single error occurs. Errors are detected in time to prevent a loss of the safety function.	

Please refer to EN 954-1 for more detailed information.

The following criteria must always be applied to the whole of a plant’s control section and not just to its component parts. A component part cannot itself meet categories 3 and 4, but if correctly applied in conjunction with other components, category 3 and 4 can be achieved.

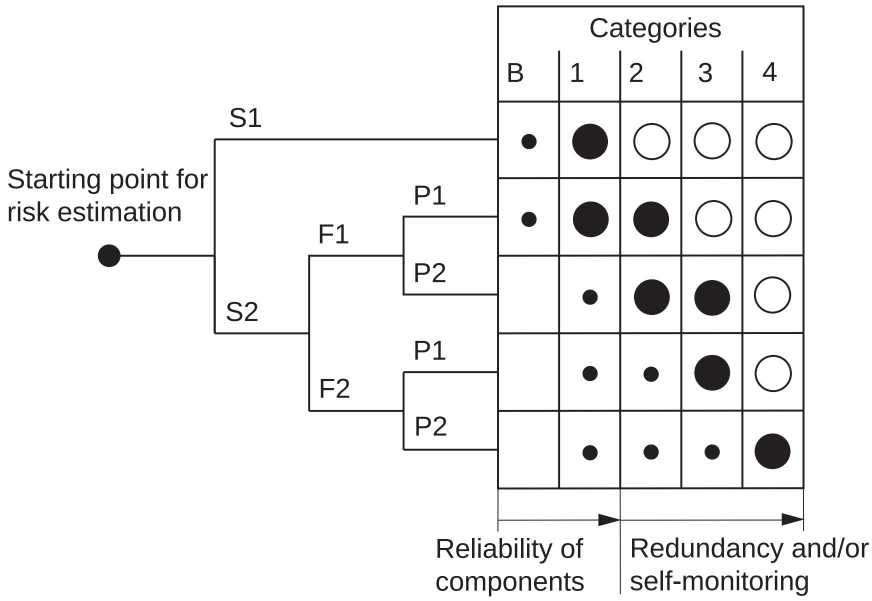
The following table shows the category into which individual product groups should be classified:

Category B	1	2	3	4	Product group
		x		x	Electrosensitive protection equipment
x	x	x	x	x	Electrical equipment to EN 60 204
	x		x	x	Locking devices
	x		x	x	Two-hand circuits
x	x	x	x	x	Pressure mats
	x		x	x	Emergency stop devices

The fact that a category is technically possible does not necessarily mean that it is permitted for a particular application.
For example: only categories 2 and 4 are permitted for electrosensitive protection equipment (e.g. light barriers).

Intended Application

Risk assessments must be carried out individually for each application. The diagram below should be of help when carrying out risk assessments: (refer to EN 954-1):



S Severity of injury
S1 Slight (normally reversible) injury
S2 Serious (normally irreversible) injury including death

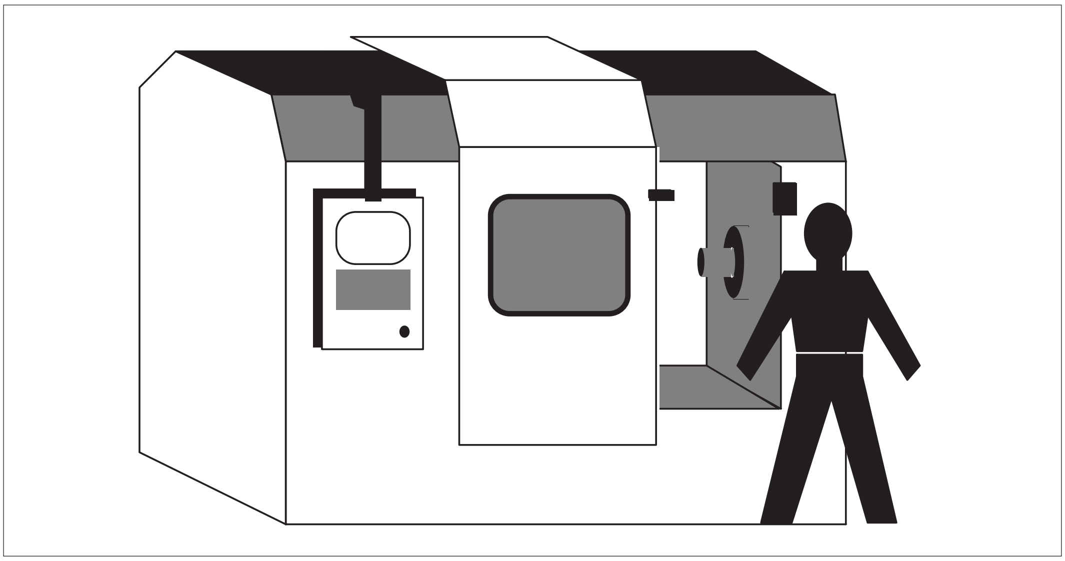
F Frequency and/or exposure time to hazard
F1 Seldom to quite often and/or the exposure time is short
F2 Frequent or continuous and/or the exposure time is long

P Possibility of avoiding the hazard
(generally these relate to the speed and frequency with which the hazardous part moves and the distance from the hazardous part)
P1 Possible under specific conditions
P2 Scarcely possible

B, 1 to 4 Categories for safety-related parts of control systems

● Preferred categories for reference points
• Possible categories which can require additional measures
○ Measures which can be over dimensioned for the relevant risk

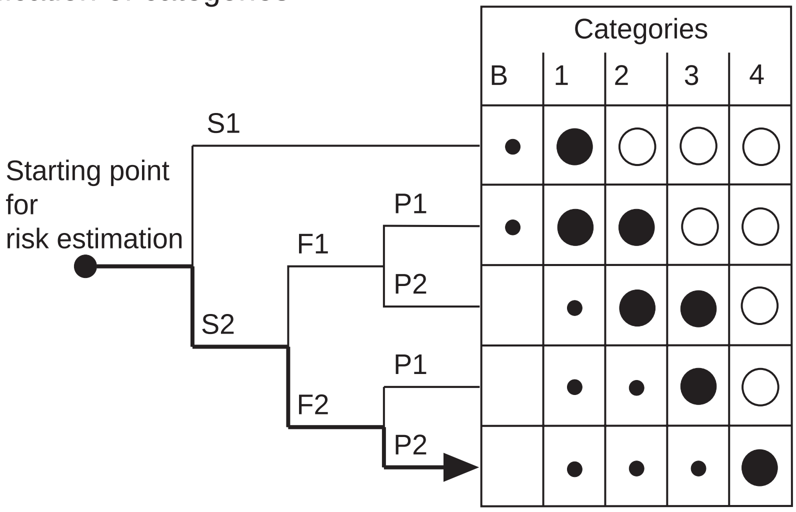
Example:
Cyclically operated safety gates on a lathe.
A CNC or PLC-driven lathe is fed manually with a work-piece into the chuck, the operation is started up and, once complete, the work-piece is again removed by hand.



If no safety devices are in place there is an extremely high risk of an accident, as careless mistakes can occur during repetitive and monotonous tasks.

Risk assessment:
S2: As the work-piece is inserted manually, severe and irreversible injuries can be caused by the rotating part of the lathe.
F2: As the work-piece is manually inserted into the hazardous areas as part of each cycle, the hazard occurs frequently.
P2: The hazard can only be avoided by using a monitored safety gate, as the work procedure is routine.

Classification of categories



The risk assessment results in category 4.
A fixed guard would not be suitable in this case as access is required every cycle. A sliding or hinged guard, preferably with position monitoring and at best with a shot bolt, would be the most suitable safety measure in this case, together with safety-related control measures to achieve category 4.

Design (Hardware)

Compact system

A compact system is composed of the following parts:

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

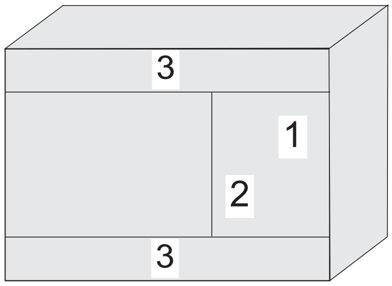


Fig. 4-1: Example of a PSS 3056 compact system

Modular system

The 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)

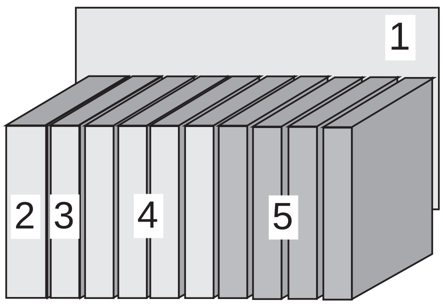


Fig. 4-2: Example of a layout for PSS 3000

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.

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.
A 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 CPU consists of three microcomputers which operate independently. Each microcomputer has a processor module, a bus system and a bus interface. Each computer processes the application program independently. The computers monitor one another in order to achieve the high safety level that is required.

- Input signals are compared. The application program will only be processed if the signals mach. If the signals do not match the safety system will switch to a safe condition.
- Output signals are only emitted if all three channels have calculated the same value. If this is not the case the safety system switches to a safe condition.

  The different processing times of the three processors need synchronising when reading in and outputting data. The operating system carries out this synchronisation automatically.

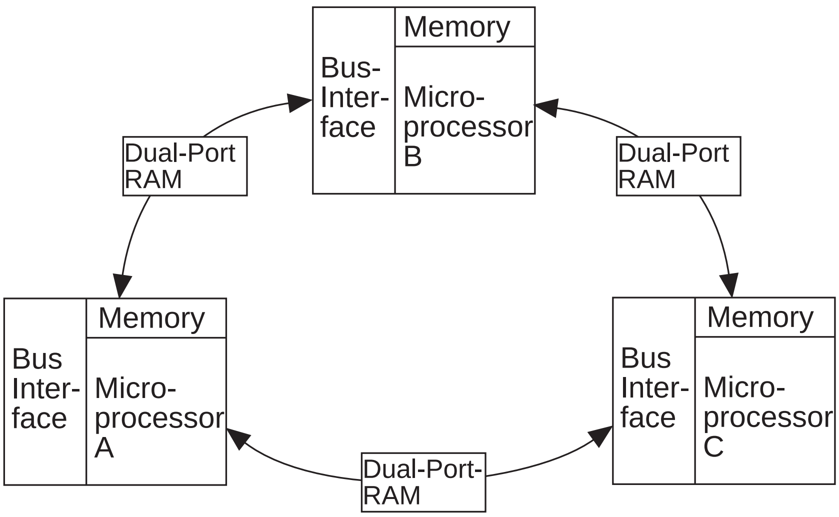


Fig. 4-3: Processing via three microcomputers

Memory

The CPU has various memories:

- Program memory
- Data memory
- Dual Port RAM

Program memory

Each microcomputer has a Flash-EPROM memory, which is used as the program memory for the application programs and data blocks.
It can store between 4,000 and 5,000 user commands, depending on the

type of command and the number of data blocks. CRC check sums monitor the program memory.

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 1,024 words. A data block consists of a data block header and user data. The data block header contains information on the data block. There are two types of data blocks:

- Read only data blocks
  can only be read by the application program
- Read/write data blocks
  can be both read 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 16 kB. The memory sector is non-retentive, so the behaviour of the system on start-up always remains the same.
  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.

INFORMATION
The values in the copy of the read/write data block are also used if the safety system switches to a STOP-condition due to an error and is then restarted.

Dual Port RAM:
The Dual Port-RAMs (DPR) are used for communication between the three computers. They store the data while the values are matched.

PG (programming) 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 it can be configured. The standard settings are:

- Baud rate: 9600 Baud
- Parity: even
- Stop bit: 1
- Data bit: 8
- Handshake: on

The user interface can be activated via the standard function block SB 255 (see chapter 6).

Timers

The CPU has 64 timers, controlled by a central timer.
The timers have a switch-on delay function and each one 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
Timer value:	Any

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

Timer addresses 64 ... 127 are available for the failsafe section.

Counters

The CPU has 64 counters, each consisting of one counter word and one counter bit. Counter words can accept values between -32.768 and 32.767. If the value of the counter word is $\geq$ 0, the counter bit assumes the value 1. The failsafe section uses counters 64 ... 127.

Display

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

Format of error messages:

F-xx	Error in failsafe section, xx: error class 01h ... FFh
*xxx +xxx	Fatal errors , xxx: error number 000h ... FFFh

Format when displaying error messages from the error stack :

Fyxx	Error in failsafe section y: Entry number in error stack 0h ... Fh x: Error class 01h ... FFh

Error messages starting with “S” refer to the standard section.

Selector switch

The programmable safety system has 2 selector switches:

- 3-position switch for the standard section (see ST System Description)
- 2-position switch for failsafe section

The selector switch for the failsafe section has three functions:

- To determine the start-up procedure when voltage is re-applied

Program starts up automatically

Program stopped

- Start/Stop switch

Start program

Stop program

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

The following tests are carried out to enable the unit to be used in safety-related plants:

- Output module
  Each cycle, outputs which are on are switched off to detect short circuits and wire breaks in the wiring
- Input module
  After the optocoupler the inputs are designed to be 3-channel. A digital test checks the digital part of the module. On some modules the optocoupler is also tested.
- A test for short circuits and wire breaks is carried out when the transmitter is connected via test signal outputs. The input filter and optocoupler are also tested.

WARNING
When wiring the inputs and outputs, the instructions given in the "Installation Manual" must be observed at all times! If wired incorrectly, subsequent faults in a machine's cycle could lead to **severe injury or even death.**

Programming

Creating programs

Parameters must be set for the failsafe section of the CPU before blocks are edited. The parameters can be set using the menu options within the configurator of the programming device, for example:

- Block run times: max. processing time for the failsafe and standard program
- Min. scan time: min. time for a cycle time to run
- Number of match attempts: max. number of attempts for matching between the three internal computers
- Set layout configuration: current configuration with inputs and outputs
- Match algorithm: Match minimum, mean or maximum value

Application programs are created on a PC (programming device) using a specially designed software package which provides various editors. The software is menu-driven and simple to use.

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

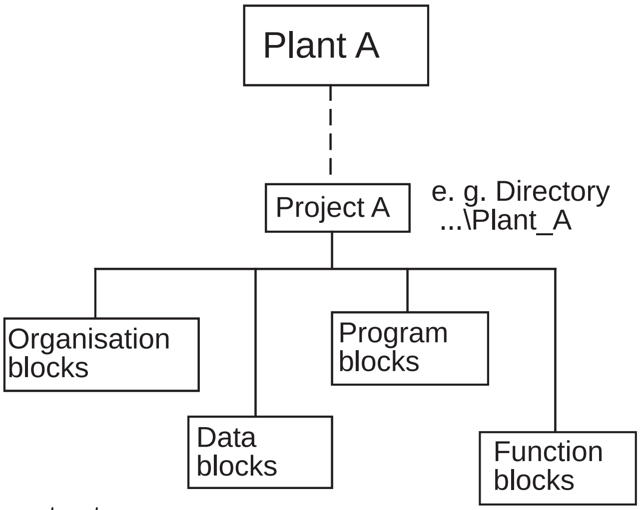


Fig. 5-1: Program structure

When all the parameters and blocks have been entered they are linked into a whole program. This process allows many program checks to be carried out, even offline. As the link is made a file is created which contains all the blocks and parameters in the program. This file can be transferred to the safety system.

NOTICE
You must link the program each time an amendment is made, otherwise the amendment will not be made online.

Transferring programs

The linked program is transferred to the safety system by activating a menu item in the programming software. Communication is only possible if the programming device containing both the software and the program is connected to the safety system. Depending on the type of PG interface used on the safety system, you may require an additional PAP interface adapter (see chapter 4, section "PG (programming) interface").

The program is transferred block by block. The transfer starts with a signal telegram which contains the name of the project. The first block is transferred once an error-free response to the telegram has been received from the safety system. The three computers in the CPU compare the block and form a check sum before sending a reply to the programming

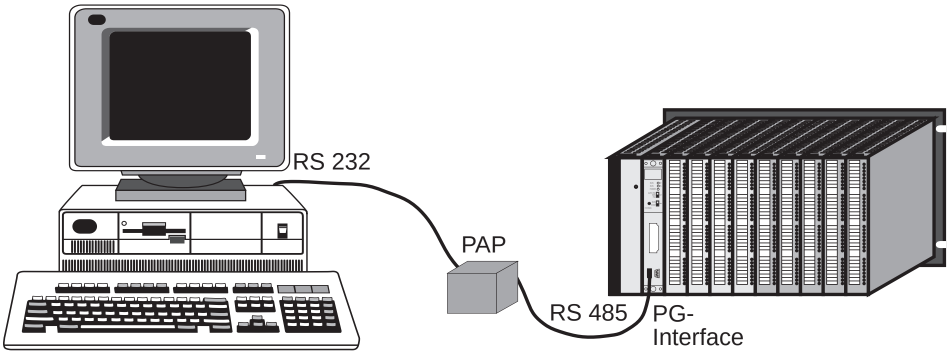


Fig. 5-2: Program transfer using PSS 3000 as an example

device. If all three computers calculate a different result an error message will occur. If they are all the same, this shows the transfer data was successful and the block is translated. The remaining blocks are processed in a similar fashion until the whole program, including parameters, is transferred.
This process ensures a high level of security. To prevent manipulations, the program can only be read back in its linked form and amendments can only be made to the source blocks.

Program cycle

An executable program is divided into the following blocks:

- Read block
- Application program
- Output block
- Test block
- Scan time block

The processing times on all these blocks is monitored by the operating system. A program cycle ends when all blocks have been run through once. The times for the program cycle are described in the chapter “Operation”.

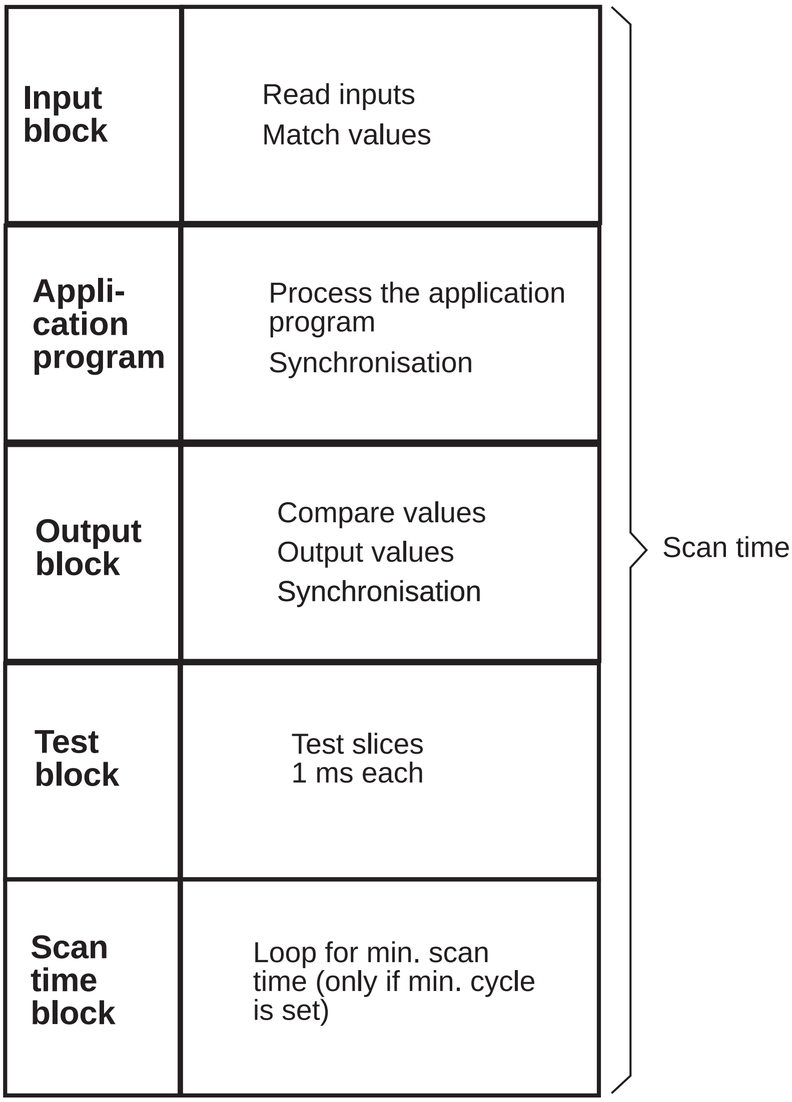


Fig. 5-3: Program cycle

Addressing

Modules are addressed through the slots. Each slot is divided into two sub-slots.
On modular systems the first sub-slot corresponds to the first two connectors and the second sub-slot to the 3rd and 4th connectors on the module. For example, on the DIO Z module there are 16 digital inputs available on the first two connectors and 8 dual-pole outputs on 3 and 4. The 16 inputs are assigned to the first sub-slot and the 8 dual-pole outputs to the second sub-slot.

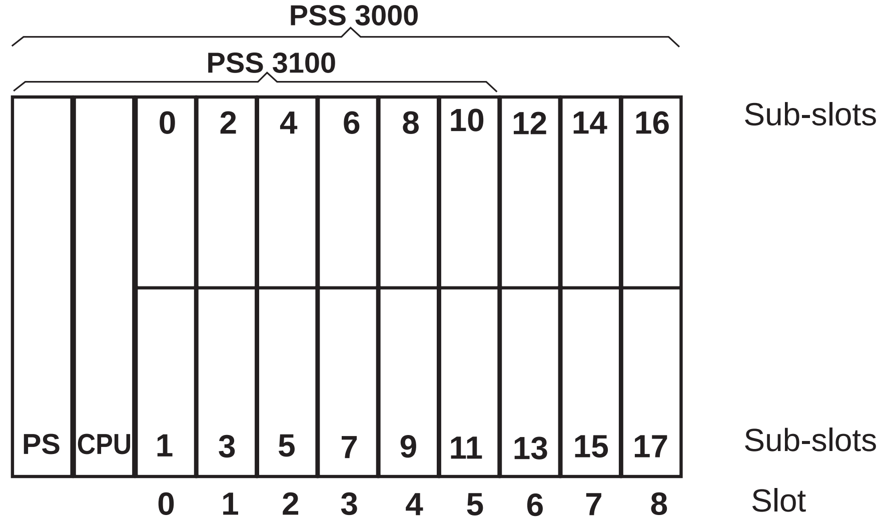


Fig. 5-4: Addressing on the PSS 3000 and PSS 3100

The way in which the slots and sub-slots are arranged on compact systems is described in the overview for the particular system.

Each slot is allocated a slot number and the digital inputs and outputs are addressed through the 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. Subsequents slots are addressed consecutively beginning from 0.

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

Address: 3.08

For word modules which have more than 32 bits a range of 8 words is reserved on each slot. Access to the addressed modules is achieved via the process I/O register (XW).

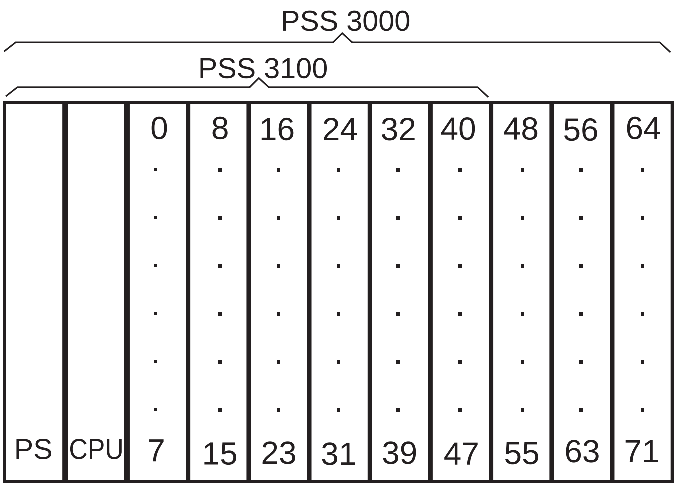


Fig. 5-5: Addressing the words on word modules

Organisation block

The various functions of the organisation blocks are set by the operating system. Each application program must contain organisation block OB 101, which among other things, manages the program cycle. Application programs are called up from within OB 101. All other organisation blocks are reserved for specific applications and it is not always necessary for these to be installed. The functions of the organisation blocks are described in the “Programming Manual” and also 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. Standard function blocks are divided into two sectors:

- SB 002 ... SB 199 (exceptions: see next point) are available to the user for any function.
- SB 001, SB 003, SB 007, SB 011, SB 015, SB 041 and SB 200 ... SB 255 are predefined and supplied by Pilz.

Standard function blocks which are predefined are tested and approved by the relevant approval bodies (e.g. BG, TÜV). Once they have been approved they are sealed to prevent any changes being made at a later date. This means they can be called up (CALL-command), but cannot be edited.
The functions of the predefined standard function blocks are described in the “Programming Manual”.

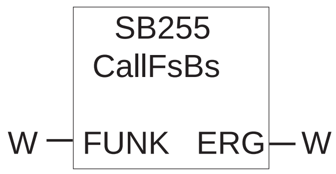
INFORMATION
Do not use the numbers of predefined standard function blocks for newly created ones.
Any standard function block saved under this description will overwrite the original block. This will then be no longer available and you will be unable to carry out the functions described within it.

Standard function block SB 255

Standard function block SB 255 is used for communication between the failsafe section and the operating system. It provides the following functions:

- Sets the test bit slices for the self-check
- Changes the match algorithm (Option)
- Sets the read segments which are to be updated (Option)
- Sets the write segments which are to be output (Option)
- Configures user interface and send/receive
- For additional functions see the corresponding system manual (e.g Selective Shutdown).

SB 255 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 entered in DB 003. The output parameter "ERG" indicates whether the function has been carried out correctly. If an error message is received, data block DB 001 will contain the cause of the error.

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

- Enter the parameters in DB 003
- Call up SB 255
- Enter the function code (input parameter "FUNK")
- Allocate an operand to the output parameter "ERG" e.g. flag
- Interrogate the contents of the flag. If it contains the code for an error message, call up data block DB 001, where the error description can be found.

Communication with peripheries

The CPU can communicate in two ways with the peripheries:

- Direct
  The signal status of the inputs is read directly or the outputs are set directly. In addition the available commands are, for example, L PB, L PW, T PB, T PW for bit modules and SB 255 with FUNK = 29 or FUNK = 25 for word modules
- Via process images
  The status of the inputs and outputs is stored as a process image. Commands for reading from and writing to the process image are e.g. L E, L EB, L EW, =A, TAB 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.

Process images

Communication is normally carried out via the process image. The status of the inputs are read at the beginning of the program cycle and compared by the 3 microprocessors. If all three processors reach the same result, this is stored in the process image of the inputs (PII). The application program is then called up and processed with the process image values. The PIO (process image of the outputs) is created during the application program and presented to the outputs at the end of the program.

The advantage of communicating with the peripheries via the process image is that the status of the inputs and outputs will remain the same during a program cycle. In addition the access time to the process image is less than the time taken for direct access.

Operation

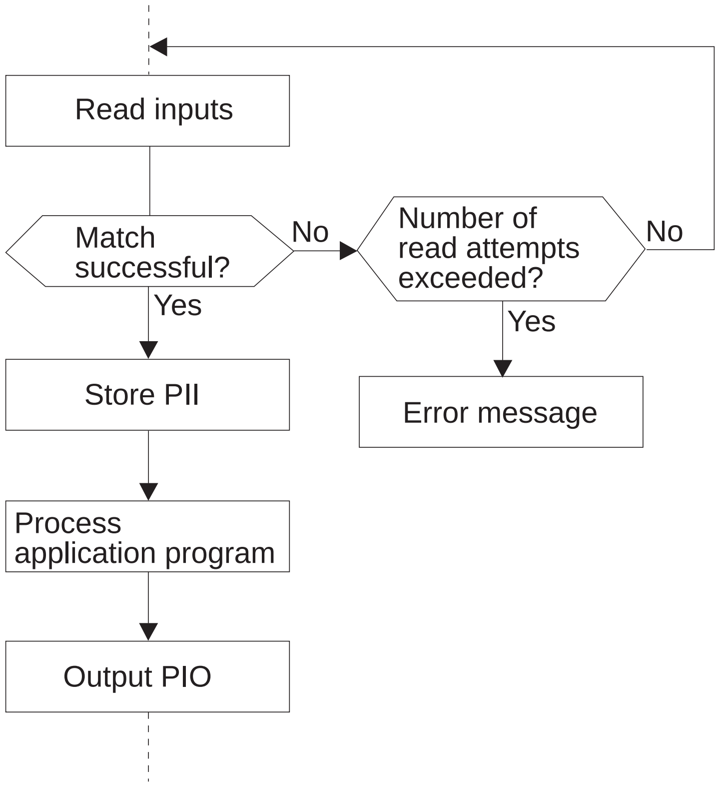


Fig. 6-1: Communication via PII and PIO

Process image of inputs on bit modules

The three microcomputers of the CPU read the digital values and compare them. If differences are found, the read operation will be carried out again and the values compared once more.

The number of permitted read and compare operations is set in the programming device’s configurator. A maximum of 50 attempts can be initiated and the length of each operation is approx 0.5 ms.
An error message will be displayed if the number of read and compare operations has been exceeded (see chapter 8).

Process image of inputs on word modules

The three processors in the CPU-module read in the values of the word modules (e. g. analogue inputs, counters) and compare them. The values that are read in are usually not identical, due to the different processor speeds and to temperature variations. The match-algorithm establishes the permitted variance, and also how the compared value should be determined. The following match-algorithms are available (set in the configurator of the programming device):

- Compare: the values read in by all 3 processors must be identical
- Minimum value without algebraic prefix: the match value is the least of the values read in
- Arithmetic mean without algebraic prefix: the match value is the arithmetic mean of the three values read in
- Maximum value without algebraic prefix: the match value is the greatest of the values read in
- Minimum value with algebraic prefix: the match value is the least of the values read in; algebraic prefixes are taken into account
- Arithmetic mean with algebraic prefix: the match value is the arithmetic mean of the values read in; algebraic prefixes are taken into account
- Maximum value with algebraic prefix: the match value is the greatest of the values read in; algebraic prefixes are taken into account.

Operation

The tolerance window indicates the maximum differential for the values read in. If “compare” has been selected as the match algorithm, no tolerance window is required.

INFORMATION
We recommend you select a low value for the tolerance window if the values read in change little in relation to time, and a high value if there are large variations.

The match algorithm is set in the configurator of the programming device. Read segments are also defined which consist of one or two data words. Read segments from double words must always start with an even address.
Example:

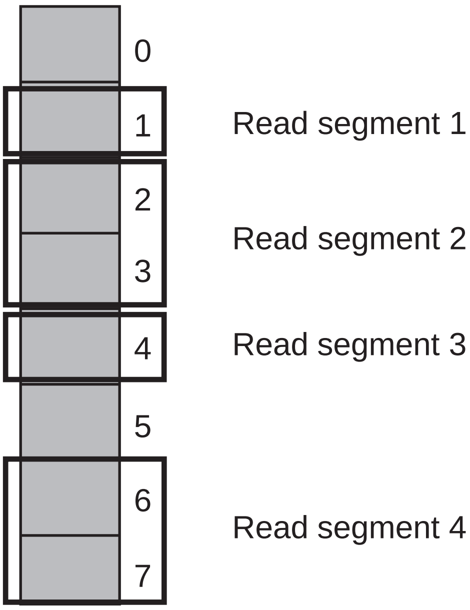


Fig. 6-2: Example of read segments

The read segments can be updated at each cycle change (set in the configurator) or after the operating system has been called up. SB 255 with the function code 25 is available for this purpose (see page 6-6). The standard function block accesses DB 003, which states which read segments are to be updated. Only the read segments in DB 003 will be processed and not all the segments on the slot, as happens when the update occurs at each cycle change.

Changing the match algorithm using the application program

The match algorithms set in the configurator can be changed using the application program by calling up SB 255 with function 21. To do this you must enter the parameters for the read segments in DB 003:

- Slot number
  The number of the slot on which the word module is located
- Read segment number
  The number of the read segment, whose match algorithm is to be changed; if you are changing several segments, the segment numbers must be located consecutively in DB 003.
- Match algorithm
  0: Compare
  1: Minimum value without algebraic prefix
  2: Arithmetic mean without algebraic prefix
  3: Maximum value without algebraic prefix
  4: Minimum value with algebraic prefix
  5: Arithmetic mean with algebraic prefix
  6: Maximum value with algebraic prefix

The end of the table must be marked by entering the value FFFF instead of the match segment number.

	Input	Output	Key
SB 255	FUNK = 21		Function code for "Change match algorithm"
		ERG = 1	No error
			All errors detected result in a STOP-condition
DB 003	DW200 = 0 ... 8		Slot number
	DW201 = 0 ... 7		Read segment number
	DW202 = 0 ... 6		Match algorithm
	DW203 DW204		Read segment number and match algorithm for second segment
	. . .		

	Input	Output	Key
	DW 215		Read segment number for last read segment
	DW 216		Match algorithm for last read segment

If the update is error-free, SB 255 will show the result as 1; if there is an error in DB 003's parameter table, the PSS changes to a STOP-condition.

Update read segment on request

Function 25 of SB 255 defines the read segments which are to be updated. The following information is required and must be entered in DB 003:

- Slot number
  Slot number on which the analogue word module is located
- Read segment number
  Number of the read segment to be updated; the numbers must be located consecutively in DB 003.

The end of the table must be marked by entering the value FFFF instead of the read segment.

	Input	Output	Key
SB 255	FUNK = 25		Function code for "Update read segment"
		ERG = 1	No error
			All errors detected result in a STOP-condition
DB 003	DW 200 = 0 ... 8		Slot number
	DW 201 = 0 ... 7		First read segment number
	.		.
	.		.
	.		.
	DW 209 = 0 .. 7		Last read segment number

If the update is error-free, SB 255 will show the result as 1; if there is an error in DB 003's parameter table, the PSS changes to a STOP-condition.

Process image of outputs on bit modules

The calculated process image from the 3 microprocessors is exchanged between the computers and compared. If no error is found the process image of the outputs is outputted. If an error occurs the PSS switches to a STOP-condition.

Process image of outputs on word modules

The 3 processors in the CPU-module calculate the output values and store them as a process output register. The processors compare the values before they are output to the word modules. Output will occur if the values are identical, otherwise the safety system will go into a STOP-condition and switch off all the outputs.
The output is set in the configurator of the programming device. Write segments are also defined which consist of one or two data words. Write segments from double words must always start with an even address (see read segments).
The write segments can be updated at each cycle change (set in the configurator) or after the operating system has been called up. SB 255 with function 29 is available for this purpose. The standard function block accesses DB 003, which states which write segments are to be updated. Only the write segments in DB 003 will be processed and not all the segments on the slot as happens when the update occurs at each cycle change.

Update write segment on request

Function 29 on SB 255 defines which output segments are to be updated. The following information is required and must be entered in DB 003:

- Slot number
  Slot number on which the analogue word module is located
- Write segment number
  Number of the write segment which is to be output
  The numbers must be located consecutively in DB 003.

The end of the table must be marked by entering the value FFFF instead of the write segment number.

Setting parameters:

	Input	Output	Key
SB 255	FUNK = 29		Function code for “Update write segment”
		ERG = 1	No error
			Each error lead to a STOP-condition
DB 003	DW 200 = 0 ... 8		Slot number
	DW 201 = 0 ... 7		Write segment number
	.		.
	.		.
	.		.
	DW 209 = 0 ... 7		Last write segment number

If the update is error-free, SB 255 will show the result as 1; if there is an error in DB 003’s parameter table, the PSS changes to a STOP-condition.

Times

The time required by the system to process a program once, including all sub-programs, is known as the “scan time”. The other run times are run through once only, e.g. when the system is switched on.

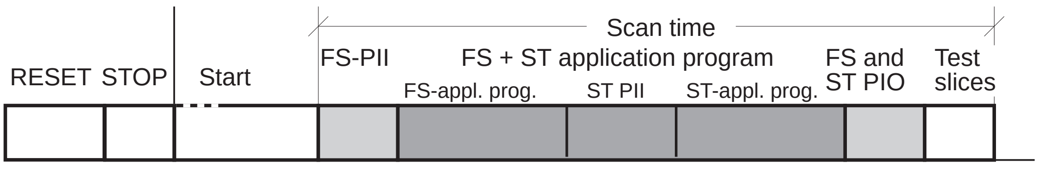


Fig. 6-3: Program cycle

Reset block
The reset block is run through once only when the safety system is switched on. The display shows "****".
When the system is switched on a self check is carried out to test the hardware and the software. After the test has ended the three microcomputers are initialised and synchronised. When this has ended, the safety system is in the STOP-condition and the display shows "0000".
Duration: approx. 30 s

Start block
The start block is run through once when there is change to the STOP-condition. A test is carried out on the application program and the structure of the internal administration tables. The configuration test is then carried out and the start-OB called up.
Duration: approx. 2 s

Read block for the process image of the inputs (PII)
Reading the process images of the inputs is explained in detail in the section “Communication with peripheries” on page 6-1.
Duration: a few ms

FS and ST application program
The application program will be started up once the input values have been successfully matched. For reasons of safety, all three computers process the application program in the failsafe section, while the program in the standard section is only processed by one of the computers. At the end of the program the three computers are again synchronised.

In addition the PII for the ST section will be read.
Duration: max. 100 ms for the combined length of the FS and ST application program, depending on the program.

Output block for the process image of the outputs (PIO)
The output values are available in the PIO once the application program has been processed. The three computers compare the values and forward them to the outputs. Again, all three computers are responsible for the output to the failsafe section, but only one is responsible for the output to the standard section. If required, one processor can take care of communication with the interface and the programming device. The processors are then synchronised.
Duration: a few ms

Test slices
A test block is processed at the end of each cycle. All the tests on the system are divided into test slices of 1 ms duration. The operating system automatically carries out a test slice in each test block.
The operator can influence the number of test slices by calling up the operating system. The test frequency can then be adapted to the status of the process being controlled.

Minimum scan time
For the scan time to be held constant, a loop can be written into the program together with the scan time. While the program is in a loop, the operating sysem carries out a series of self-checks:

  Block run time FS
+ Block run time ST
+ Operating system (read, output, test)
+ Test slices (loop)

Minimum scan time

The minimum scan time must be less than 100 ms.

Defining the scan time and block run time

The scan time is made up of constant and variable values:

- Creating the PIO and PII is just one of the tasks of the operating system. The operating system requires approx. 10 ms for this task and all other tasks (i.e. testing the inputs), which must be fulfilled during the cycle.

INFORMATION
There are safety systems, e.g. selective shutdown, which require other times for these tasks. Please refer to the respective manuals.

- The length of an application programs is variable. An application program comprises mostly of different program sections. In each cycle many different program sections can be carried out. The combined length of the FS and ST application program may be a maximum of 100 ms.
- Test slices can be carried out in each cycle. The length of this process is variable and depends on the number of test slices to be carried out. Each test slice is a minimum of 1 ms. The number of test slices can be defined via calling up the operating system (see the section on “Self-check”).
- By setting a minimum scan time an additional waiting time may be necessary, which is taken up in whole or part by the test slice process.

Defining the block run times and scan time:

- Default values are given in the configurator for the block run time of the aaplication program and the minimum scan time. Use these values for the first program test!
- Leave the application program to run through several cycles. If the information given for the minimum scan time is too small, an error message “F-20” appears on the display. Increase the value for the minimum scan time in the configurator and start a new test.
- The current, maximum and minimum times are entered in the data block DB 0, DW 007 ... DW 012 . The values can be called up online using the function “display variables”(see the “Programming Device” manual).
  DB 000

DW 007	current scan time in ms
DW 008	max. scan time in ms
DW 009	current block run time of the FS application progam
DW 010	max. block run time of the FS application program
DW 011	current block run time of the ST application program
DW012	max. block run time of the ST application program

- If the times in the configurator are suitable, enter the minimum cycle time if you require a constant scan time.
  The minimum cycle time must be
  - Less than the FS block run time + ST block run time + operating system or 100 ms and
  - Greater than the current max. FS block run time (DW 010) + current max. ST block run time (DW 012) + operating system.

  Delete the value for the minimum scan time if the cycles are to be variable.

INFORMATION

The minimum cycle time must be less than the maximum cycle time and a max. 100 ms.
The block run time for the FS application program may be a maximum of 100 ms.

Self check

A regular self check must be carried out on the safety system to ensure that undetected errors do not accumulate. The self check tests the parts of the safety system in which a failure may not be detected immediately under certain conditions of normal operation:

- Processors
- Memory
- Periphery devices

The self check is divided into test bit slices. Each bit slice carries out part of the self check. Approximately 40,000 bit slices are required to test the entire system.

Self checks are carried out in various situations:

- When voltage is restored
  Complete self check of the whole system.
  The test lasts approximately 30 seconds, and all the segments on the display are switched on.
- When changing from STOP to RUN condition (warm start)
  The length of the test is negligible; the configuration is checked along wit the module functions.
- Self check during cycle
  During operation the self check is carried out in bit slices at each cycle change. One bit slice requires approximately 1 ms of test time. One bit slice is carried out automatically each cycle. The number of bit slices per cycle can be influenced by the following:
  - Defining a minimum scan time
    There is a waiting time at the end of each cycle, depending on the length of the application program. If possible test slices are carried out during this period.
  - Calling up the operating system with SB 255
    The number of test bit slices is set by the user.

Procedure:

- Start self check
  The number of test bit slices (range 1 ... 99) must be entered in data word DW 200 in DB 003.
  In the application program, call up SB 255 using FUNK = 1.
- Status poll
  - Call up SB 255 using parameter FUNK = 1
  - Poll value of ERG parameter:
    ERG = 0 signifies that installation has been error-free
    ERG = 16 indicates an error in the SB 255 parameters.

The set number of bit slices will be processed at the next cycle change. The setting is valid for one cycle only. Once the test bit slices have been processed, another test bit slice will be automatically set.

	Input	Output	Key
SB 255	FUNK = 1	 ERG = 1 ERG = 16	Function code for "Set test slices" No error Error present
DB 003		DW200 =	1 ... 99 Number of test slices
DB 001		if ERG = 16: DW200 = 0 DW200 = 1	 No error Too many test slices

If the self check finds a major error, the safety system will immediately switch to a safe condition. If the standard section is not affected by the error it will continue to be processed.

Process interrupts

Processing process interrupts enables external events to be managed quickly. Program processing is interrupted if an alarm occurs. Process interrupts can only be triggered using special alarm inputs.

If an alarm occurs the CPU exits from the application program and returns to the operating system. The CPU determines the module and the input triggering the alarm. An alarm-OB is assigned to the input marked by the alarm. The CPU returns to the application program and continues to process the current block. At the next block change or network command the highlighted OB is processed.
If several alarms occur simultaneously, they are stored in a queue. A maximum of 32 entries can be stored at any one time and are processed in sequence. All entries are processed in sequence at the next cycle change. Any more than 32 alarms cannot be processed.
The alarm function is locked during the reading and matching of the PII and the reading and outputting of the PIO. The alarms will be assigned to the queue.

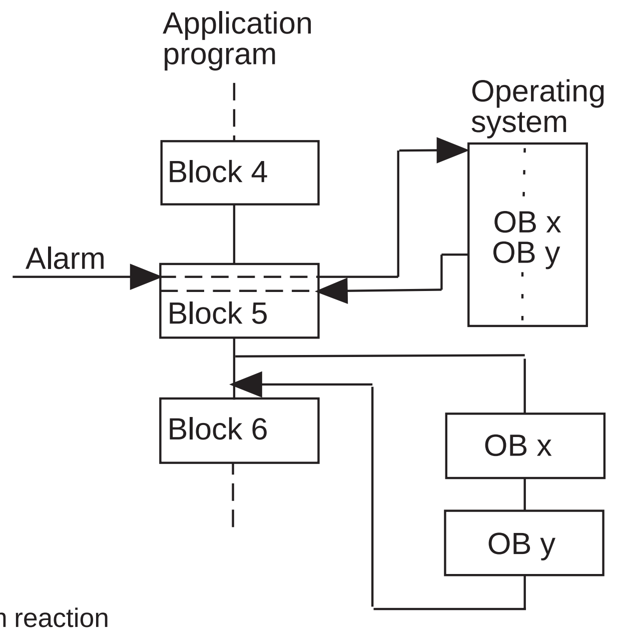


Fig. 6-4: Alarm reaction

Operation

Alarms which occur during the ST application program interrupt the application program immediately, and process the FS Alarm-OB. The ST application program then continues from the point where the program was interrupted.

The alarm reaction time comprises:

- Max. run time difference of the CPUs
- Max. time difference between the block change or two network commands.
- run time of the alarm OBs.

If, due to a system error, the CPU switches to a STOP condition, all alarms are cleared.

Alarm inputs

Special alarm interrupts are available for alarm processing, e.g. PSS DIF module. These inputs react quicker to a signal change than “normal” inputs.
On modular systems:
If several alarms occur simultaneously the slot determines the processing sequence. Slot 0 has the highest priority and the order of priority decreases as the slot number increases.

An alarm can be triggered via a

- Rising edge
- Falling edge
- Each edge

Alarm organisation blocks

If a process interrupt has occurred due to an alarm input, the alarm calls up the Alarm-OB. This OB will contain the reaction to the process interrupt. OB 140 ... OB 171 are reserved for alarm processing. The allocation of alarm input and alarm -OB, together with the edge selection is entered in the configurator.

NOTICE

- Alarm inputs on a sub-slot must be allocated consecutively to an Alarm-OB. An alarm input need not be configured
- DIF-inputs which are not configured as alarm inputs can be used as normal inputs with very quick response times
- Alarm inputs used for safety functions must operate to normally energised mode. Only alarms configured with a falling edge are safety-relevant.
- Check safety alarm inputs with test pulses!

When managing process interrupts, please note:

- Alarms are managed in the sequence in which they occur
- The reaction time of sporadic process interrupts depends on the configuration, the application program, the use of test signals and the number of alarms which occur simultaneously
- If process interrupts occur regularly, the CPU has little time to process administrative tasks and the application program. If the application program and process interrupt management procedures have been incorrectly designed, the CPU may be permanently occupied with handling alarms: while one alarm is being processed the next one is triggered, and the program locks up. Signal monitoring is energised.
- With dynamic program display of alarm organisation blocks, block execution time is extended during diagnostics. This means the program could lock up if regular process interrupts are being handled and the program design is critical.

If an alarm is present, it will only be detected if the following conditions are met:

- Both the "0"-Signal and the "1"-Signal must be present for at least 1.5 ms (alarm detection time) for an alarm to be detected if a pulse edge changes
  Example: An alarm is to be triggered if a falling pulse edge occurs. The "1"-Signal must be present for at least 1.5 ms and then the "0"-Signal for at least 1.5 ms for an alarm to be triggered for a change from "1" to "0".
  A time delay of 3 ms must occur between two alarms at one input, which is configured for a rising or falling edge.
  If these conditions are not met, the alarm that is triggered cannot be traced, error F-20 will be displayed and will be entered in the error stack.

- If the time delay between alarms at various alarm inputs is important, these alarms must be separated by at least 1.5 ms. If the time delay is not kept alarms will not be processed according to time sequence, but in the order of the alarm input number (e.g. first of all the alarm at E0.0, then the alarm at E0.1, then E0.2 etc.).
  A shaft encoder, which supplies two in quadrature square signals to determine speed and direction, may have a maximum frequency 166 Hz. 166 Hz corresponds to a 6 ms time period and a 1.5 ms delay from edge to edge.
- A maximum of two PSS(1) DIF modules should be installed in one system. A maximum of 32 alarm inputs are permitted per system.
  If more than two PSS(1) DIF modules are installed, the alarm detection time will be increased. The CPU will have to interrogate more modules for alarms that may have been triggered.
- When an Alarm-OB is called up, the result of the logic operation (RLO) is set at the value of the input at the time of the alarm.
- In order to reduce reaction time, use at least 4 to 5 block segment commands per block, depending on the length of the block.
  Alarms are only triggered at segment commands or when changing blocks. Long blocks without segment commands will increase the alarm reaction time just as much as time-intensive commands (timer operations, access from periphery devices).
- Time-intensive alarm-OBs increase alarm reaction time.
  All alarms have equal priority. When an alarm is triggered, the relevant alarm-OB is called up and processed. Reaction to the next alarm will only be possible once the Alarm-OB has been processed.
- Alarms are stored in a queuing stack. If a lot of alarms occur in quick succession, the alarm reaction time to the last alarm will be increased by the time taken to process the alarms which entered the stack first.
- Alarm reaction time is not dependent on the number of alarms configured, but rather on the number of alarms which occur (almost) simultaneously.
- For quicker alarms, do not use pulsed alarm inputs. Alarm reaction time is longer with pulsed inputs, as no alarms are triggered during the function test on the inputs.

Communication with the standard section

Two types of flag 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 failsafe section has read access only.
  - Status flags FS and ST provide information on the status of the system. The failsafe section has 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	FS 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
M111.01	= 1, when Overflow-Flag is set via arithmetic operation	read
M111.02	= 1, when Zero-Flag is set via arithmetic operation	read
M111.03	= 1, when Sign-Flag is set via arithmetic operation	read

Flag type		Key	FS access
Status flags ST			
	M 112.00	= 0, when ST is in STOP-condition = 1, when ST is in RUN-condition	read
	M 112.01	= 1, when error in ST	read
	M 112.02	= 1, when ST is in STOP-condition following a command	read
	M 112.03	= 1 after transfer from STOP to RUN (for one cycle)	read
	M 112.04	= 1 after transfer from "voltage off" to RUN (for one cycle)	read
	M 112.05	= 1, when reset in the ST was carried out (for one cycle)	read
Status flags FS			
	M 113.00	= 0, when FS is in STOP-condition = 1, when FS 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			
MW114.00 ... MW114.16		Indicator for indirect Addressing	read/write

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

User interface for the failsafe section

The user interface is an RS 232 interface which can be used with either the failsafe or the standard section for communicating with other devices.

NOTICE
The user is responsible for the accuracy of the data transfer. In the application program check whether the received data is feasible. we also recommend you use the CRC-calculation, supplied as standard function block SB 001 (see “CRC-calculation” on page 6-27).

The following functions in SB 255 enable the failsafe section to access the user interface:

FUNK	Key
200	Poll status: Configuration
201	Configure
202	Acknowledge configuration error
204	Poll status: Send
205	Send
206	Acknowledge send error
208	Poll status: Receive
210	Acknowledge receive error
211	Acknowledge receipt

Configuration of the user interface

Block	Input	Output	Key
SB 255	FUNK = 200		**Poll status: Configuration**
		ERG = 1	Interface ready for operation
		ERG = 2	Interface not yet configured
		ERG = 16	Configuration error
SB 255	FUNK = 201		**Configure**
		ERG = 1	Interface ready for operation
		ERG = 16	Configuration error
DB 003	DW 201		Error coding on configuration error
	DW 202		Data block number configuration
	DW 203		Data block number receive buffer
	DW 204		Data block number send buffer
SB255	FUNK = 202		**Acknowledge configuration error**
		ERG = 1	Interface ready for operation
		ERG = 2	Interface is being configured

In DB 003, numbers must be established for 3 data blocks:

- DW 202: Configuration block
- DW 203: Receive block
- DW 204: Send block

Structure of the configuration block:

DW 002: Baud rate, default setting: 9600

Value	0	1	2	3	4	5	6	7
Baud	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

INFORMATION
The user interface is configured with the base settings every time there is a transfer from STOP to RUN.
The user interface can be configured **either** for the failsafe section **or** for the standard section!
If the user interface is configured for the failsafe section, it can only be used by the standard section if:

- A general reset is carried out in the standard section and
- The PSS operating voltage is switched off and then on again.

Sending data via the user interface

	Input	Output	Key
SB 255	FUNK = 204		**Poll status: Send**
		ERG = 1	Interface ready for operation
		ERG = 2	Telegram is sent
		ERG = 16	Send error
SB 255	FUNK = 205		**Send**
		ERG = 1	Interface ready for operation
		ERG = 2	Telegram is sent
		ERG = 16	Send error
Send block	DW000		Number of bytes to be sent
		DW001	Error coding on send error
	DW002 .. DWxxx		Send data
SB 255	FUNK = 206		**Acknowledge data error**
		ERG = 1	Interface ready for operation
		ERG = 2	Telegram is sent
		ERG = 16	Acknowledge send error

Send block: Enter the number for the DB in DB003, DW204
Send sequence: DR2, DL2, DR3, DL3, DR4, DL4 ... DRx, DLx

Receive via the user interface

	Input	Output	Key
SB 255	FUNK = 208		**Poll status: Receive**
		ERG = 1	No data received
		ERG = 2	Data received is ready to retrieve
		ERG = 16	Receive error
Receive block		DW 000	Number of bytes received
		DW 001	Error coding on receive error
		DW 002 .. DWxxx	Receive data
SB 255	FUNK = 210		**Acknowledge receive error**
		ERG = 1	No data received
		ERG = 2	Received data ready to be retrieved
		ERG = 16	Receive error
SB 255	FUNK = 211		**Acknowledge receipt**
		ERG = 1	Acknowledgement successful
		ERG = 16	Acknowledgement error
DB 003	DW 200		Number of bytes to be ackowledged

The data received is transferred to the receive block. Enter the number for the DB in DB 003, DW 203.

Example for communication via the user interface

- Configuring the interface
  Baud rate: 9600
  Parity: even
  Stop bit: 1
  Data bit: 8
  Handshake: on

  Configuration block: DB 010
  Send block: DB 011
  Receive block: DB 012

  Cycle:
  call up SB 255
  Poll status with FUNK = 200
  if ERG = 1: Store configuration data in DB 003:
  DW 202 010
  DW 203 012
  DW 204 011
  and enter configuration data in DB 010
  DW 002 6
  DW 003 2
  DW 004 1
  DW 005 3
  DW 005 1
  Configure interface with FUNK = 201
  if ERG = 1: Configuration successful

  ERG = 2: Interface is being configured
  ERG = 16: Configuration error; in DW 201 of DB 003 states the number of the data word which contains the fault;
  Remove error and acknowledge with FUNK = 202

- Send
  Call up SB 255
  Poll status with FUNK = 204:
    if ERG = 1: Store send data in DB 011 from DW 002 onwards,
    in DW 000 stores the number of sent bytes to be sent
  Send data with FUNK = 205
    if ERG = 1: Sent successfully

  ERG = 2: Data is being sent
  ERG = 16 Send error; error code appears in DB 011 DW 001;
  Remove error and acknowledge with FUNK = 206.

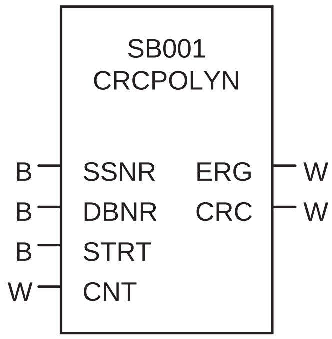
- Receive
  Poll status with FUNK = 208:
    If ERG = 2: received data stored in DB012 from DW 002 onwards,
    the number of received bytes appears in DW 000
    With the next call up of FUNK = 208 , any additionally received data will be written in DB 012 behind the data already retrieved.
    The number of received data is updated in DW 000.
    Acknowledge receipt with FUNK = 211
    After receipt the data must be retrieved from DB 012 and then the receipt acknowledged; the operating system sets the number of bytes in DB 012, DW 000 to 0;
    The number of acknowledged bytes appears in DB 003, DW 200
    Poll status with FUNK = 208
    All data not yet acknowledged will be written in DB 012 and then the new data.

  if ERG = 1: No data received
  if ERG = 16: Send error; check whether the number of acknowledged bytes in DB 003, DW 200 is correct and acknowledge the error with FUNK = 211.

CRC-calculation

Standard function block SB 001 is integrated in the FS section using data and flag ranges. Up to five CRC-calculations can be carried out by this block in any one cycle. The maximum run time per call up is 2 ms. The generator polynomial is 1021h and the start value is FFFFh.

Block header:



Input parameters

- SSNR   Call up number,
  Value range : 0 ... 255
- DBNR   Data block number of the data range to be used to calculate the CRC check sum; only when the parameter
  STRT = DL or DR (or STRT = MB: DBNR no meaning)
  Value range: 10 ... 255
- STRT   Start address of the data range to be used to calculate the CRC check sum
  Value ranges: MB64.00 ... MB99.24, DL0 ... DL1023,
  DR0 ... DR1023
- CNT   Number of bytes to be used to calculate the CRC check sum
  Value range: > 0

Output parameters

• ERG	Result:	2	CRC-calculation not completed , needs to be called up again.
		4	CRC-calculation completed CRC-check sum is in parameter CRC
		16	Error occurred during CRC-calculation.
• CRC	CRC check sum, if ERG = 4		

Test pulses for signal inputs with infrequent operation

In order to carry out a function test on signal inputs with infrequent operation (i.e. inputs which do not change status often), the signal inputs must be switched off momentarily. Modules with test pulse outputs are available for this purpose, e.g. PSS DIO T. Allocating test pulse outputs and those inputs which are to be tested is carried out through the configurator on the programming device.

INFORMATION

- Always use test pulses in sequence
  Test pulses which are not being used must be tested in exactly the same way as those that are used, as the test time would be extended unnecessarily if the signals were not used in sequence.
- Adjacent inputs must be tested with different test pulses.
  If two adjacent inputs were tested using the same test signal, any short circuit between the two inputs would not be detected. Design measures should exclude the possibility of shorts between non-adjacent inputs.
- Connect the PSS (1) DI-module and the PSS(1) DIF-module to different test pulses.
  The response time of the PSS (1) DIF-modules is shorter than that for the PSS(1) DI-module. The pulse adapts automatically to the response time to keep the test time short. The response time of the PSS(1) DIF-module is approximately 0.5 ms, and that for the PSS(1) DI module approximately 1 ms or 3 ms. If the pulse is connected to both the PSS(1) DIF and the PSS(1) DI-module, the test time cannot be reduced according to the response time of the PSS(1) DIF-module.

Output to display

Hexadecimal figures can be output to the CPU display. The characters must be located in data block **DB 003** DW 200 . Each time SB 255 is called up the display is updated with the function FUNK = 32. Hexadecimal figures can be output to the CPU display.

INFORMATION
System error messages are displayed as priority.

Setting the parameters:

	Input	Output	Key
SB 255	FUNK = 32		Function code for "Output to display"
		ERG = 1	No error
		ERG = 16	Error during output
DB 003	DW200 = xxxx		Show xxxx characters on the display
	DW200 = FFFF		Clear display

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.

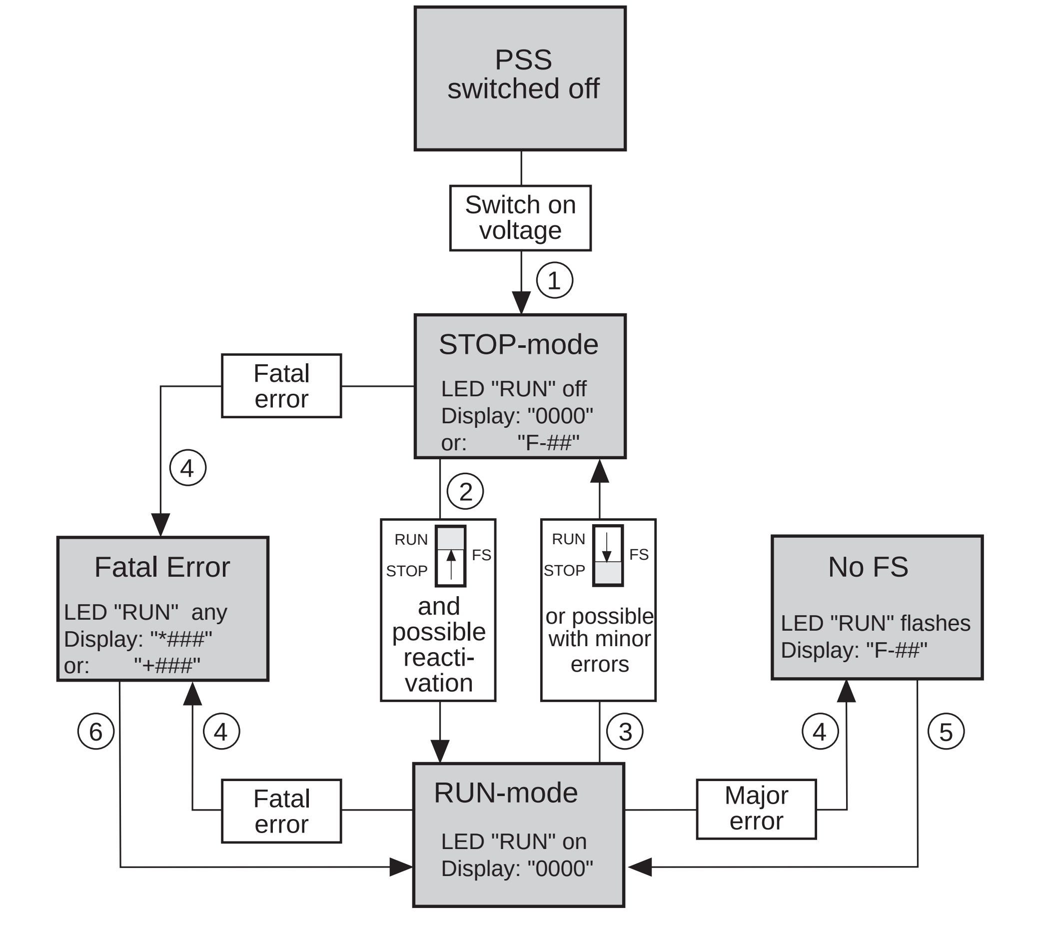


Fig. 6-5: PSS operating status

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 PII and PIO of the FS section
- 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 PII and PIO of the FS section
- All programming device functions are available
  (Exceptions: Load program and clear program).

"No FS"

After a major error the PSS goes into a "No FS" condition
- The FS application program is not processed
- The ST application program continues and can read the PII and PIO of the FS section
- The programming device functions are limited to read access only.

"Fatal Error"

After a fatal error the PSS goes to a "Fatal Error" condition:
- FS and ST section are not operating
- Communication is not possible with the programming device

If the PSS goes into this condition, you can only do the following:
- Establish the conditions under which the errors occurred
- Write down the error message displayed.
- Contact Pilz.

Change in the PSS operating condition

Switch on voltage ①

When the voltage is switched on, the PSS is in a STOP condition and carries out a self test.
Display: all segments are illuminated
Status flag M 113.04 = 1
Duration: approx. 30 s

Change from STOP to RUN (start-up) ②

By setting the FS selector switch from STOP to RUN the PSS reacts in the following way:

- Checks the application program
- Runs start-up-OB
- Carries out a configuration test and module test
- PIO and PII written with 0
- Sets flags to 0
- Initialises DB (with PG values)
- Stops timers
- Resets counters
- Status flags M 113.03 = 1 and M 113.00 ... M 113.02 = 0

Change from RUN to STOP ③

If the FS mode selector switch is set to STOP or due to an error reaction the PSS goes into a STOP condition:

- All outputs are switched off
- PIO written with 0
- Status flags M 113.00 = 0 or, if STP-command: M113.02 = 1 or if an error occurred : M 113.01 = 1

Change from RUN to "No FS" 4

When a major error occurs the PSS goes into a “No FS” condition. All outputs of the FS modules are switched off and the FS application program is stopped.

Change from "No FS" to RUN 5

A direct switch from "No FS" to RUN is not possible. If you wish to exit from a “No FS” condition, the following method should be used:

- Read from the error stack using the programming device
- Switch off the PSS and set the FS mode selector switch to STOP
- Rectify the error.
- Switch on the PSS and set the FS mode selector switch to RUN.

Change from RUN to"Fatal Error" 4

When a fatal error occurs the PSS goes into "Fatal Error". All output (FS and ST) will be switched off. Both the FS and ST application progams will be halted.

Change from "Fatal Error" to RUN 6

see "Change from "No FS" to RUN

Change from STOP to "Fatal Error" 4

Any fatal errors are detected by the PSS even in a STOP condition and will cause the PSS to go into a "Fatal Error".

Start-up Procedure

Initial start-up

Hardware requirements:

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

Software requirements:

- Data entered in the configurator on the programming device (slot configuration, block run time etc.) is correct
- Executable application program is available in a linked form (see "Link" in the “Programming Manual”).

Initial start-up procedure:

- Set "FS" selector switch to STOP
- Switch on power supply (Position "I")
  Reation: LED "Power" on the power supply and CPU module are illuminated
  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" from the programming software
  - We recommend you clear the program memory before transferring the program
  - Transfer program
- Set “FS” selector switch to “RUN”
  Reaction: The program is run.
  LED "RUN" lights up.

Error messages are described in chapter 8.

Start-up Procedure

Reset

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 error messages) and the LED "RUN " will either flash or go out:

- LED "RUN" flashes: a major error has occurred. The number on the display provides information on the error.

  Reset:
  - Rectify the error, if necessary use the programming 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" goes out and the display shows “F-xx”: an error has occurred.

  Reset:
  - Rectify the error. If necessary use the programming device to locate the error in the error stack
  - Set "FS" switch to "STOP" and then to "RUN". The program will run again
  - LED "RUN" goes out and the display shows error number "+xxx" or *xxx": a fatal error has occurred. The system is defective.

Changing the configuration of the application program

Changing the configuration, e.g. adding an extra module or exchanging one must be entered in the configurator:

- Enter the changes using the configurator in the programming software (see "Programming Device" manual).
- Link program (see "Programming Device" manual).
- Proceed as by “Initial start-up”

After a change to the application program the program must be re-linked using the programming software. Switching on the safety system, transferring the program and the start of the program correspond to the initial start-up procedure.

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 PSS goes immediately into a "Fatal Error"
- In event of a major error the PSS goes into "No FS".
- In the event of minor errors the PSS switches to a STOP condition. With some errors an error OB will be called up before the change to STOP
- Message errors do not influence the program cycle; an error message appears only.

Error stack

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

DW	Configuration
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

INFORMATION
DB 000 can only be read in the standard section. Access to the error stack from the failsafe section is not possible.

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

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.

Minor errors/message errors

Minor errors apply to errors in the application program which are detected during the program cycle or from the operating system. Program errors include:

- Feasibility errors, i.e. errors in the periphery modules and the external wiring
- Address range exceeded
- Attempting to write a write-protected block
- Module errors

The reaction to a minor error is described in an error organisation block, OB 125 or OB 127. 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.

Allocation between error-OB and error classes:

- OB125 has error classes F-21 and F-22 assigned.
- OB127 has error classes F-23 and F-24 assigned.

Major/fatal errors

Major errors or fatal errors:

- Errors during the self check
- Block run time exceeded

Error diagnostics and correction

- Errors which prevent correct program processing
- Differences in PIO
- Power supply error

The safety system switches immediately to one of the two conditions, depending on the type of error:
- No FS
- Fatal Error

Reaction to an error

Fig. 8-1 shows the correlation between errors and the PSS operating status.

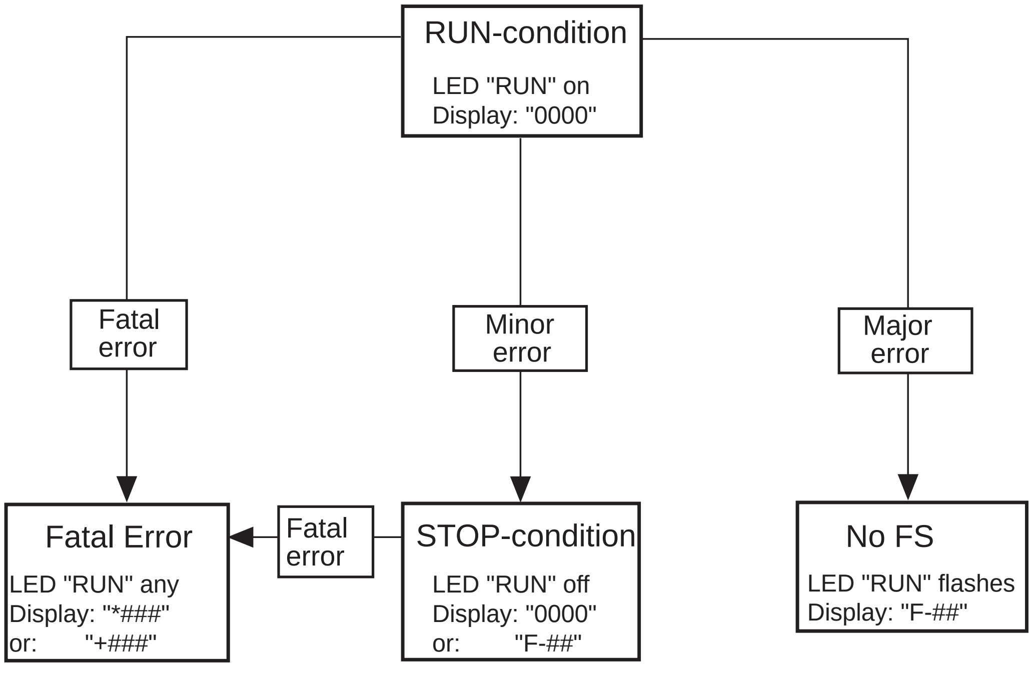


Fig. 8-1: Change in operating status due to an error or a change in position of the FS selector switch

Error types:

- Minor errors
  The system switches to a STOP condition, the LED "RUN" is off and the error message "**F-##**" appears on the display. The standard application program continues to run and all functions on the programming device are available.
  - Possible causes:
    Errors in the application program or a minor module error
  - Remedy:
    - Read from the error stack using the programming device
    - Set a trigger condition for error search with the dynamic program display
    - Correct the error
    - Restart the failsafe application program by operating the FS selector switch: move the selector switch from RUN to STOP and back to RUN.
- Major errors
  The system switches to a "No FS", the LED "RUN" flashes and an error message "**F-##**" appears on the display. The standard section continues to run and the functions on the programming device are restricted to read-only functions.
  - Possible causes:
    Major module errors or irregularities between the three channels
  - 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
  The system switches to "Fatal Error", the LED "RUN" assumes any status and an error message "***###**" or "**+###**" appears on the display. Both the failsafe section and the standard section are 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
- Write down the error message displayed
- Contact Pilz

Display contents of the 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: F012

  Error in the failsafe section
  Entry number
  Error class

- 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:12 for error class 12
  - Error number, e.g. N:23 for error number 23
    and where possible
  - Error parameter
  - Error location

Alternatively the contents of the error stack can be displayed using the programming device, providing that the programming device is connected to the safety system. The “ONLINE - show 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 description of the “error messages”
- P
  Processor which detected the error (e.g. A, B or C; with selective shutdown 1 ... 7)
- Error description
  Information on the error number, error description, module and slot with module error

• Error location
  Error in the application program: Information on faulty blocks,
  Errors in modules or wiring errors : Information on the slot and bit address (hex).

Display module errors

Module errors will be identified by the slot on which they occur and displayed via the slot number and the letter "U" or "O". "U" means the lower sub-slot and "O" the upper sub-slot.
The bit address for a wiring error or module error is displayed as a hexadecimal code: Bit = 1 means error-free, Bit = 0 means an error.

• Allocation: Bit number - code word:
  Each bit of code has a bit of the module assigned.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit address for Bit 0 ... 15 (upper sub-slot)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit address for Bit 16 ... 31 (lower sub-slot)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

• Conversion of the bit coding to the hexadecimal code:
  4 Bit of a word give a hexadecimal digit: Bit 12 ... 15 results in the 1st digit, Bit 8 ... 11 the 2nd digit, Bit 4 ... 7 the 3rd digit Bit 0 .. 3 the 4th.

• Conversion into hexadecimal numbers:

Binary	Hexadecimal	Binary	Hexadecimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Table of examples for the faulty bits:

Bit numbers on the upper sub-slots "O"

Code		
Hexadecimal	Binary	Addresses
FFFE	1111 1111 1111 1110	X.00
FFFD	1111 1111 1111 1101	X.01
FFFB	1111 1111 1111 1011	X.02
FFF7	1111 1111 1111 0111	X.03
FFEF	1111 1111 1110 1111	X.04
FFDF	1111 1111 1101 1111	X.05
FFBF	1111 1111 1011 1111	X.06
FF7F	1111 1111 0111 1111	X.07
FEFF	1111 1110 1111 1111	X.08
FDFF	1111 1101 1111 1111	X.09
FBFF	1111 1011 1111 1111	X.10
F7FF	1111 0111 1111 1111	X.11
EFFF	1110 1111 1111 1111	X.12
DFFF	1101 1111 1111 1111	X.13
BFFF	1011 1111 1111 1111	X.14
7FFF	0111 1111 1111 1111	X.15

Table 1: Coding of the bit addresses of the upper slots

Bit numbers on the lower slots "U"

Code		Addresses	
Hexadecimal	Binary	Single-pole	Dual-pole
FFFE	1111 1111 1111 1110	X.16	X.16+
FFFD	1111 1111 1111 1101	X.17	X.17+
FFFB	1111 1111 1111 1011	X.18	X.18+
FFF7	1111 1111 1111 0111	X.19	X.19+
FFEF	1111 1111 1110 1111	X.20	X.20+
FFDF	1111 1111 1101 1111	X.21	X.21+
FFBF	1111 1111 1011 1111	X.22	X.22+
FF7F	1111 1111 0111 1111	X.23	X.23+
FEFF	1111 1110 1111 1111	X.24	X.16-
FDFF	1111 1101 1111 1111	X.25	X.17-
FBFF	1111 1011 1111 1111	X.26	X.18-
F7FF	1111 0111 1111 1111	X.27	X.19-

Code		Addresses	
Hexadecimal	Binary	single-pole	dual-pole
EFFF	1110 1111 1111 1111	X.28	X.20-
DFFF	1101 1111 1111 1111	X.29	X.21-
BFFF	1011 1111 1111 1111	X.30	X.22-
7FFF	0111 1111 1111 1111	X.31	X.23

Table 2: Coding for the bit addresses on the lower sub-slots.

If several bits are faulty, a 0 appears at the corresponding position, e.g.: FAFB corresponds to 1111 1010 1111 1011 (see the conversion on page 8-7), i.e. the bits 10, 8 and 2 are defective.

Examples

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

Class	No.	P	Error stack of PSS
F-06	07	1	Wiring error or error on the DIOZ at slot 3.21

"F-06": error class 06 means a module error (see description for error messages)
"1": Processor A has detected the error
"07": Error number 07 means: The PIO cannot be read back/ short circuit to 24 V on dual-pole outputs or in the wiring

Error evalution on display
The display shows the following error message:

F-06

Press and hold down the error stack button.
The following messages appear on the display :

F020 - C=20 - N=03

F020: Failsafe error message in position 0 of the error stack with the error class 20
C=20: Error class 20,
N=03: error number 03 i.e. PSS is in a STOP-condition

Release the error stack button and press again keeping the button held down. The next entry in the error stack will be displayed:

F106	-	C=06	-	N=07	-	AT	-	FFFE	-	PARA	-	0201

F106: Failsafe error message in position 1 of the error stack with error class 06
C=06: Error class 06 means module error (see descripton for error messages)
N=07: Error class number 07 means: The PIO cannot be read back/short circuit to 24 V on dual-pole outputs or in the wiring
AT: means: error location follows
FFFE: according to table 2 the fault bit is Bit x.16
PARA: means: the error parameter follows
0201: lower sub-slot, slot 1
(the first two digits indicate the sub-slot: 01 = upper
02 = lower, and the last two digits indicate the slot number)

Result: output 1.16 is defective or a short circuit has occurred.

Diagnostics

There are various tools available for error diagnostics:

- Variable display: The current status of variables is displayed as a table (e.g. inputs and outputs)
- Dynamic program display: The current contents of the operands in a program section are displayed.

The use of the diagnostic tools is described in the Programming Manual.

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 in the non-volatile memory.

Dynamic program display

Dynamic program display shows the content 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 programming 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 a new cycle the search for the program segment begins again, and the information is 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 in 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.

Appendix

System data blocks

Data blocks are available which enable communication between the failsafe application program or the programming device and the operating system. The following table provides a configuration overview.

DB 000

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

INFORMATION
DB 000 can only be read from the standard section!

Data word	Coding	Configuration
000	KY	Current year 00,00 ... 00,99
001	KY	Current month 01 ... 12, current day 01 ... 31
002	KY	Current hour 00 ... 23, current minute 00 ... 59
003	KY	Current seconds 00 ... 59, free
004	KH	Operating system verison number
005	KH	Hardware version number
006	KH	Reserved
007	KH	Current scan time in ms
008	KH	Max. scan time since the last start of the failsafe section in ms
009	KH	Current block run time of the failsafe application program in ms
010	KH	Max. block run time of the failsafe application program since the last program start in ms
011	KH	Current block run time of the standard application program in ms
012	KH	Max. block run time of the standard application program since the last program start in ms
013	KH	Length of self check in ms
014	KH	Display: 0 = DW15 contains user data 1 = "F-xx" Failsafe error 3 = "S-xx" Standard error
015	KH	Display: hex. number, meaning see DW 014

DB 000

Data word	Coding	Configuration
016 ... 019	KH	Reserved
020 ... 043	KH	True layout configuration of the ST section
044 ... 083	KH	Reserved
084	KH	Indicator for the error stack
085 ... 148	KH	Contents of the 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 ST section
156	KH	Internal hardware version of the ST section
157 ... 163	KH	Operating system CRC check sum
164	KH	FS application program CRC check sum
165 ... 168	KC	FS application program project name
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	Transfer parameter for operating system call-ups in the ST section

DB 001

DB 001 contains messages for the operating system call ups and information for selective shutdown (this can only be read by the PSS with selective shutdown configured).

Data word	Coding	Configuration
000	KH	Number of read repeats, Value range: 1 ... 50
001	KH	Version ID: Bit 0 = 1: PSS with selective shutdown
002 ... 199		Free
200 ... 220	KH	Enable parameter for operating system call ups, see. "Standard function blocks - operating system-call ups"
221 ... 249		Free
250 ... 397	KH	Reserved for selective shutdown

DB 002

DB 002 contains the data which was set in the configurator. It cannot be amended by the application program.

Data word	Coding	Configuation
000	KF	Block run time FS program in ms, range 1 ... 100
001	KF	Block run time ST program in ms, range 0.5 .. 99
002	KF	Minimum scan time in ms, range 1 ... 100 No minimum scan time: FFFF
003	KF	Number of match attempts, range: 1 ... 50 Length of attempt: approx. 0.5 ms
004	KF	Function test time of the DI module in ms, range: 1 ... 5
005	KF	Max. alarm reaction time in ms, range: 2 ... 100 No max. alarm reaction time: FFFF
006	KH	Operating system version
007 ... 014	KY	DB-ID for operating system version
015 ... 018	-	Free
019	KF	PG version

DB 002

Data word	Coding	Configuration
020 ...037	KH	Set layout configuration, DW 020: Code for sub-slot 0 DW 021: Code for sub-slot 1 . . . DW 037: Code for sub-slot 17
038 ... 083	-	Reserved
084 ... 099	-	Free
100 ... 387	 KY KY KF KF KF KF	Parameters for match algorithms DR 100: 1 = Update on each cycle change 2 = Update on request DL 100: 0 ... 8 = Slot number DR 101 1 ... 2 = Length of read segment in words DL 101 0 ... 7 = Start address for read segment DW 102 0 = compare 1 = Min. value with prefix 2 = Arithmetic mean without prefix 3 = Max. value without prefix 4 = Min. value with prefix 5 = Arithmetic mean with prefix 6 = Max. value with prefix DW 103 0 ... 65535 = Tolerance window DW 104 ... DW 107 Data for second read segment . . . DW 384 ... DW 387 Data for last read segment Marking table end: Slot number = 255
388 ... 399	-	Free

DB 002

Data word	Coding	Configuration
400 ... 543	 KY KY KY KY	Parameters for write segments DR 400 1 = Output at cycle change 2 = Output on request DL 400 0 ... 8 = Slot number DR 401 1 ... 2 = Length of write segment in words 0 ... 7 = Start address of write segment DW 402 ... DW 403 Data for second write segment . . . DW 542 ... DW 543 Data for last write segment Marking for table end : Slot number = 255
544 ... 549	-	Free
550 ... 838	 KY KM	Test pulse allocation DR 550 0 ... 15 = Number of the last pulsed output used; no test pulse used: Enter 255 DL 550 0/1 = sub slot on which the module with the pulsed output is located DW 551 Test pulse allocation sub-slot 0, test pulse 1 . . . DW 838 Test pulse allocation sub-slot 17, test pulse 15

DB 002

Data word	Coding	Configuration
840 ... 871		Configuration of the process alarms
	KY	DR 840 allocation to OB 140
		0 ... 17 = sub-slot number of the alarm module
		Dl 840
		0 = No alarm
		1 = Alarm with rising edge
		2 = Alarm with falling edge
		3 = Alarm with rising and falling edge
	KY	DR 841 allocation to OB 141
		0 ... 17 = sub-slot number
		DL 841
		0 = No alarm
		1 = rising edge
		2 = falling edge
		3 = rising and falling edge
		.
		.
		.
	KY	DR 871 allocation to OB 171
		0 ... 17 = sub-slot number
		DR 871
		0 = no alarm
		1 = rising edge
		2 = falling edge
		3 = rising and falling edge
		Assign data words not used with FFFF

DB 003

DB 003 contain various parameters for the different functions of SB 255.

Data word	Coding	Configuration
000 ... 199		Free
200 ... 220		Transfer parameter failsafe - operating system-routine

Standard function blocks - operating system call ups

The following table describes the functions of the standard function block SB 255, which trigger an operating system call up:

Function code	Function
1	Sets the test bit slices
21	Sets the match algorithm
25	Updates the read segments
29	Updates the write segments
200 ... 211	Operating the user interface

Test slices for self check: FUNK = 1

Block	Input	Output	Key
SB 255	FUNK = 1	 ERG = 1	Sets test slices No error Any error leads to a STOP condition
DB 003	DW 200 = 1 ... 99		Number of test slices
DB 001		DW 200 = 1	Too many test slices

Change match algorithm: FUNK = 21

Block	Input	Output	Key
SB 255	FUNK = 21		Change match algorithm
		ERG = 1	No errors
			Any errors detected lead to a STOP-condition
DB 003	DW 200 = 0 ... 8		Slot number
	DW 201 = 0 ... 7		Read segment number
	DW 202 = 0 ... 6		Match algorithm
	DW 203		Read segment and
	DW 204		Match algorithm for second segment
	. . .		
	DW 215		Read segment number for last segment
	DW 216		Match algorithm for last read segment

Update read segments: FUNK = 25

Block	Input	Output	Key
SB 255	FUNK = 25		Update read segments
		ERG = 1	No errors
			Any errors detected lead to STOP-condition
DB 003	DW 200 = 0 ... 8		Slot number
	DW 201 = 0 ... 7		Read segment number
	. . .		
	DW 209 = 0 .. 7		Last read segment number

Update write segments: FUNK = 29

Block	Input	Output	Key
SB 255	FUNK = 29	 ERG = 1	Send write segment No errors Each error leads to a STOP-condition
DB 003	DW 200 = 0 ... 8 DW 201 = 0 ... 7 . . . DW 209 = 0 ... 7		Slot number Write segment number . . . Last write segment number

Output to CPU display: FUNK = 32

Block	Input	Output	Key
SB 255	FUNK = 32	 ERG = 1 ERG = 16	Function code for "Output to display" No error Error during output
DB 003	DW200 = xxxx DW200 = FFFF		Show xxxx characters on the display Clear display

Define user interface:

Configuring the user interface: FUNK = 200/201/202

Block	Input	Output	Key
SB 255	FUNK = 200		**Poll status: Configuration**
		ERG = 1	Interface ready for operation
		ERG = 2	Interface is being configured
		ERG = 16	Configuration error
SB 255	FUNK = 201		**Configure**
		ERG = 1	Interface not ready
		ERG = 16	Configuration error
DB 003	DW 202		Data block number: Configuration
	DW 203		Data block number: Receive buffer
	DW 204		Data block number: Send buffer
SB255	FUNK = 202		**Acknowledge configuration error**
		ERG = 1 ERG = 2	Interface ready for operation Interface is being configured

Send via the user interface: FUNK = 204/205/206

Block	Input	Output	Key
SB 255	FUNK = 204		**Poll status: Send**
		ERG = 1	Interface ready for operation
		ERG = 2	Telegram is being sent
		ERG = 16	Send error
SB 255	FUNK = 205		**Send**
		ERG = 1	Interface ready for operation
		ERG = 2 ERG = 16	Telegram is being sent Send error
Send block	DW000		Number of bytes to be sent
		DW001	Error coding on send error
	DW002 .. DWxxx		Send data
SB 255	FUNK = 206		**Acknowledge send error**
		ERG = 1	Interface ready for operation
		ERG = 2	Telegram is being sent
		ERG = 16	Send acknowledgement error

Receive via user interface: FUNK = 208/210/211

Block	Input	Output	Key
SB 255	FUNK = 208		**Poll status: Receive**
		ERG = 1	No data received
		ERG = 2	Received data ready to be retrieved
		ERG = 16	Receive error
Receive block		DW000	Nmber of received bytes
		DW001	Error coding on receive error
		DW002 .. DWxxx	Receive data
SB 255	FUNK = 210		**Acknowledge receive error**
		ERG = 1	No data received
		ERG = 2	Received data ready to be retrieved
		ERG = 16	Receive error
SB 255	FUNK = 211		**Acknowledge receipt**
		ERG = 1	Receive acknowledge successful
		ERG = 16	Receive acknowledge error
DB 003	DW200		Number of bytes to be acknowledged

Organisation blocks

The application progam determines the call up conditions for the organisation blocks. The following table provides a configuration overview.

Organisation block	Call-ups	Application
OB 101	Cycle start	Administration of program cycles, which must be contained in every application program
OB 120	Transfer from STOP to RUN	Initialisation
OB 125, OB 127	With errors (see chapter 8)	Data transfer to the standard section for diagnostics
OB 128	On a manual change to a STOP-condition (selector switch "RUN - STOP")	Data transfer to the standard section
OB 140...OB 171	With process alarms	Reaction to process alarms

Amendment list

Changes from Version V to Version VI

Old page	New page	Change
4-5 5-2	4-5 5-2	Type of PG interface used (RS 232 or RS 485) depends on the safety system. A PAP interface adapter is not required for an RS 232 interface.
9-3	9-3	Data word configuration of DB 002
9-7	9-7	Data word configuration of DB 003

Changes from Version VI to Version VII

Old page	New page	Change
8-9	8-9	"Error evaluation using the programming device" has been amended

Index

Index

Index

Notes

A

Pilz Ges.m.b.H.
Modecenterstraße 14
1030 Wien
Austria
Telephone (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
Australia
Telephone (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
Belgium
Telephone (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
Brazil
Telephone (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
Switzerland
Telephone (0 62) 8 89 79 30
Telefax (0 62) 8 89 79 40
E-Mail: pilz@bluewin.ch

D

Headquarters:
Pilz GmbH & Co.
Felix-Wankel-Straße 2
73760 Ostfildern
Germany
Telephone (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
Denmark
Telephone 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
Spain
Telephone (93) 8 49 74 33
Telefax (93) 8 49 75 44
E-Mail: central@pilzspain.com

F

Pilz France Electronic
1, rue Jacob Mayer
BP 12
67037 Strasbourg Cedex
France
Telephone 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
Finland
Telephone (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
United Kingdom
Telephone (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)
Italy
Telephone (0 31) 78 95 11
Telefax (0 31) 78 95 55
E-Mail: pilzitalia@pilz.it

IRL

Pilz Ireland Industrial Automation
Cork Business and Technology Park
Model Farm Road
Cork
Ireland
Telephone (0 21) 4 34 65 35
Telefax (0 21) 4 80 49 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
Telephone (0 45) 4 71-22 81
Telefax (0 45) 4 71-22 83
E-Mail: pilz@pilz.co.jp

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
Mexico
Telephone (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
Netherlands
Telephone (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
Telephone (21) 9 28 91 09
Telefax (21) 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
Telephone (0 20) 87 37 16 18
Telefax (0 20) 87 37 35 55
E-Mail: pilzchn@public.guangzhou.gd.cn

S

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

SGP

Pilz Industrial Automation Pte Ltd.
61, Kaki Bukit Ave 1, #05-01
Shun Li Industrial Park
Singapore 417943
Singapore
Telephone 8 44 44 40
Telefax 8 44 44 41
E-Mail: sales@pilz.com.sg

USA

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

...

In many countries we are represented by sales partners. Please refer to our Homepage for further details or contact our headquarters.

www

www.pilz.com

pilz

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

CERTIFIED DIN EN ISO 9001 DQS QUALITY SYSTEM
REG. NO. 2462

18 645-01/01 Printed in Germany