Profile : Process controller

Profile No. : 81

Date : November 22, 1994

Published by : INTERBUS-S CLUB e.V.
Geschäftsstelle
Postfach 23 47, D-67611 Kaiserslautern
Telephone: * 49 - 631 - 7 94 24,
Fax: * 49 - 631 - 9 76 58

Order No. : 81

Contents

Page

1. Preface ... 3

2. Introduction ... 3

3. Scope of Application ... 4

4. References ... 4

5. Definitions ... 5

6. Symbols and Abbreviations ... 7

7. Device Characterization ... 8
7.1. Device Data ... 10
7.2. System Data ... 11

8. Application and Device Characteristics ... 12
8.1. Device Control ... 13
8.1.1. Device Control State Machine ... 14
8.1.2. Control Word ... 18
8.1.3. Status Word ... 20
8.1.4. Malfunction Function ... 22
8.1.5. Warning Function ... 26
8.2. Controller Function ... 28
8.2.1. Setpoint Value / Actual Value Scaling ... 29
8.2.2. Sensor Error Function ... 45
8.2.3. Deviation Function ... 46
8.2.4. Learn Function ... 48
8.3. Sensor/Actuator Functions ... 50
8.3.1. Communication Function ... 50
8.3.2. Device Information ... 50

9. Data Structures ... 51

10. Device Relationships ... 55

11. Operating Phases of the Application ... 55
11.1. Start-up/Abort ... 56
11.2. Operation ... 56
11.3. Start-up and Projecting Phases ... 56

12. Communication Profile ... 57
12.1. Layer 1 ... 57
12.2. Layer 2 ... 58
12.3. Layer 7 ... 59

1. Preface

Within the framework of factory automation, increasingly powerful and flexible systems are needed in the field of industrial sensors and actuators. Process controllers can meet these requirements. However, open and standardized communication capabilities are needed to enable their complete integration into complex production sequences.

The basic concept of open systems is to enable an exchange of information between application functions implemented on hardware from a diversity of manufacturers.

These functions include defined application functions, a standard user interface for communications and a standard transmission medium.

To be able to define the device functions of the process controller independent of the communication medium, an internationally recognized and standardized user interface (DIN 19 245 Part 2) was used for communications. This created compatibility with MMS.

The INTERBUS-S system, which meets the requirements of sensors and actuators with regard to real-time response and a standardized user interface, was chosen as the communication medium.

As standardization work is continuing, additions are to be expected.

2. Introduction

This profile defines application functions of process controllers. The application functions are subdivided into process controller functions, communication functions, and control functions. In addition, free areas for the manufacturer-specific functions are defined (see Figure 1).

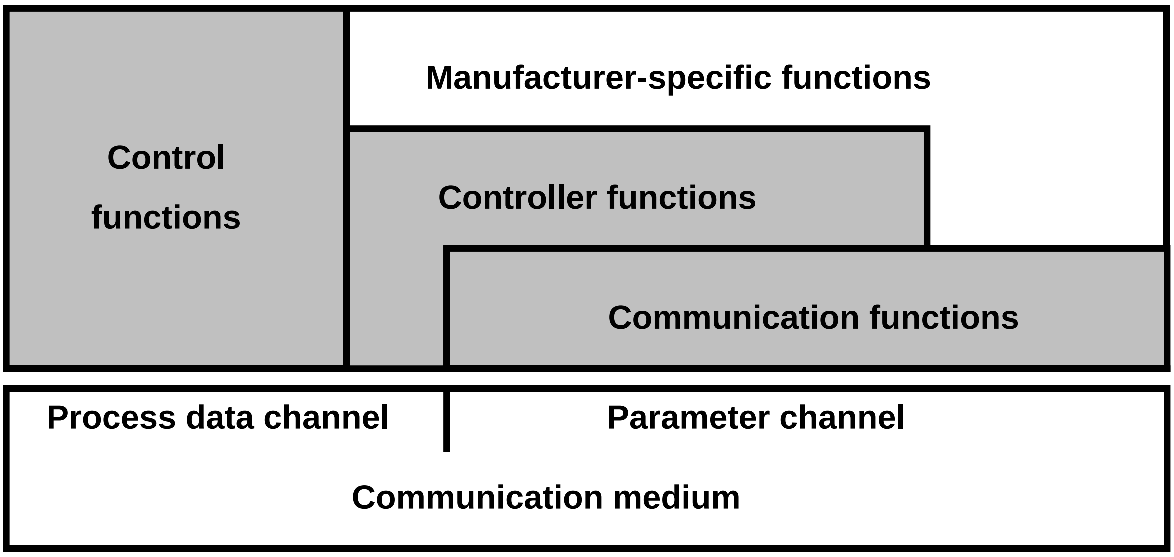

Figure 1: Application functions of process controllers

Each application function is described with the aid of a function block. The device response is described with the help of a state machine in the control functions. This profile takes into account the fact that there may be separate hardware for communication and for tasks related to communication.

The process controller functions comprise the standardized process controller functions, which communicate with the communication medium through the defined communication functions.

The control functions serve to coordinate the function areas; a further subdivision into standardized and manufacturer-specific functions is possible. In this profile, the device control is described as a standardized control function.

The freely definable manufacturer-specific functions may utilize the standardized functions of all other function areas.

3. Scope of Application

The definitions in this profile are intended for the user and device manufacturer of process controllers which are to be operated on the sensor/actuator bus.

For the user, this profile definition is a useful supplement to the standardized communication and provides a universal convention on data contents and device behavior. It causes devices of different device manufacturers to respond identically in connection with the communication medium.

4. References

The structure of this profile conforms to the design guidelines of the INTERBUS-S CLUB e.V. for INTERBUS-S profiles.

The application protocol and the data structures conform to DIN 19 245, Part 2.

The application interface for communication via the INTERBUS-S parameter channel also conforms to DIN 19 245, Part 2.

The definitions for data transfer through the process data channel are based on the INTERBUS-S specification.

Authors:

Mr. Ammon	Mannesmann Rexroth Pneumatic, Hanover
Mr. Dörflinger	Mercedes-Benz AG, Sindelfingen
Mr. Ehrlich	Mercedes-Benz AG, Sindelfingen
Mr. Grigoleit	Behr Industrieanlagen, Bietigheim
Mr. Jessel	Behr Industrieanlagen, Bietigheim
Mr. Krumsiek	Phoenix Contact, Blomberg
Mr. Loos	LacTec, Rodgau
Mr. Roser	SIG-Berger Lahr, Lahr
Mr. Schuhmacher	ABB Fläkt Ransburg, Heusenstamm

5. Definitions

Device Profile

The device profile defines the application functions that are visible through communication. The application functions are mapped onto the communication by the following definitions:

- by the communication profile,
- by interaction between the application functions, insofar as they are executed through the communication system, and
- by the communication services (utilities) used and the communication objects that can be manipulated with them.

The result of this mapping is the visible response of the application. The definitions contained in an application profile enable interoperability in a field of application if permitted by the device characteristics used.

Characteristics of devices significant to the user are also defined.

A distinction is made between mandatory functions, optional and manufacturer-specific device functions, and parameters.

If users restrict themselves to the mandatory functions or parameters, interchangeability of devices is possible if this is permitted by the device characteristics and settings used. With respect to communication, and regardless of the function, devices are always interchangeable if use is made of the same parameters.

Communication Profile

In relation to the specific application or hardware group, the communication profile limits or classifies the degrees of freedom contained in the specification of the data transfer medium. The communication profile defines communication services and parameters that are identified in the specification as being optional.

All optional functions and parameters that are not stated in the communication profile remain optional. Mandatory services and parameters are binding, even if not stated in the profile.

The profile also limits or defines value ranges of attributes and parameters.

The communication medium is INTERBUS-S.

Sensor/Actuator

This profile contains the basic functions that every sensor and actuator must provide to a user. These are mainly the communication functions and the device information.

Communication Interface

The communication interface is composed of a channel for services compatible with DIN 19 245, Part 2 (Peripherals Communication Protocol channel, PCP channel), and a channel for process data.

All communication objects can be accessed through the PCP channel. The process data channel serves the purpose of swift transfer of specific communication objects. The PCP services allow acknowledged access to communication objects, i.e. access to a communication object is confirmed by the process controller.

Below, the data transferred via the process data channel are referred to as process data.

Through the process data channel, data is transferred in unacknowledged and equidistant form. It has a width of up to 16 bytes. Each byte can be read and written.

The direction specified for the process data is viewed from the bus, i.e.,

- Process output data is data read by the process controller out of the process data channel.

- Process input data is data written by the process controller into the process data channel.

VFD Object

The Virtual Field Device (VFD) is an abstract model for describing the data and the behavior of a programmable controller from the point of view of its communication partner. The basis of the VFD model is the VFD object. The VFD object contains all objects and object descriptions that can be used by a communication partner through services. The object descriptions are contained in an object dictionary. There is precisely one object dictionary for each VFD.

Communication Reference

Each communication relationship between two devices is projected, independent of when it will be used. The projecting is stored in each bus station in a communication relationship list (KBL). An application process identifies the communication relationship via a local communication reference. Thus, the communication reference is used for addressing the communication partner.

Error Message

The error message is returned when a service could not be executed.

Index, Subindex

The index is used to address a parameter (communication object). The subindex addresses a subparameter (element of a communication object) within a parameter created as a structure.

Device Parameter

This profile contains default values for all device parameters.

Substitute Values

When the optional communication objects are not implemented, the devices respond according to the substitute value defined for this parameter.

Mandatory Range

The mandatory area is the range of values where a parameter, if implemented, can be parameterized in any case.

State Machine

Some functions are described in this profile with the aid of a state machine. A state represents a specific internal and external response. It can only be terminated by means of defined events. Corresponding state transitions are assigned to events. Actions can be executed at a transition. The response of the state is changed at the transition. When the transition is ended, the current state is followed by the new state.

Definitions Specific to Process Controllers

Single Controller

A single controller is a device with a bus connection of its own.

Multiple Controller

A multiple controller is a device combining between 1 and 12 controllers with a common bus connection.

6. Symbols and Abbreviations

AF	Air flow controller
DP	Dosing pump
HV	High-voltage controller
PQ	Paint quantity controller
PR	Pressure controller
SP	Speed controller

Network-Specific Abbreviations

KBL	Communication relationship list

7. Device Characterization

The process controller constitutes the link between the automation equipment (PLC, host computer) and the process. In simple terms: the process controller converts setpoint values from the application program into physical variables for the process.

The market of general-purpose process controllers requires a wide range of different devices with regard to functions and prices. Owing to their open structure, the process controllers cover the whole variety of functions.

The device functions and parameters are classified into mandatory, optional and manufacturer-specific ones. If users restrict themselves to the mandatory functions or mandatory parameters, interchangeability of devices is possible.

With respect to communication, and regardless of the function, the devices are always interchangeable if use is made of the same parameters.

The process controllers are InterBus-S stations; they may take between 1 and 12 controllers of different types (see Figure 2).

The bus interface includes a process data channel with a length of 2 words for single controllers, and 8 or 24 words for multiple controllers. In addition, the parameter channel is defined as having a length of 1 or 2 words. According to the type of controller used, the controller parameters are generated in the object dictionary of the station as corresponding communication objects.

A control and a status word, as well as a setpoint value and an actual value are mapped onto the process data channel as default.

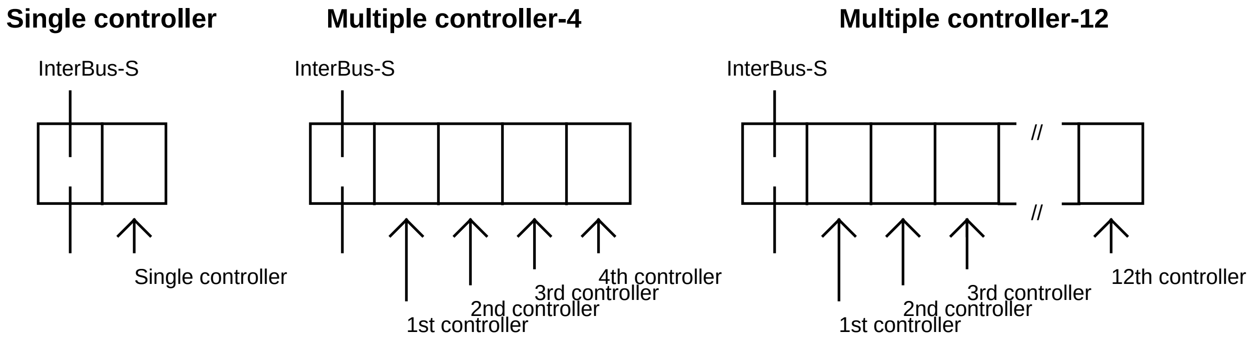

Figure 2: Different types of process controllers

Single Controllers

Single controllers use one word for control and status information and one word for setpoint values and for actual values in the process data channel.

The bus interfacing is shown in Figure 3.

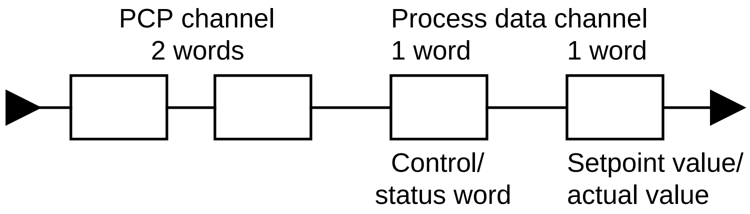

Figure 3: Bus interfacing

NOTE: A PCP channel with 2 words (SUPI2-internal: 4 words) currently does not require an external register extension.

Multiple Controllers

Multiple controllers use one word for control and status information and one word for setpoint and actual values per controller. The length of the process data channel is fixed (8 or 24 words).

The length of the process data channel must not change during operation (e.g. when a controller fails).

Bus interfacing of a 4-fold multiple controller(see Figure 4):

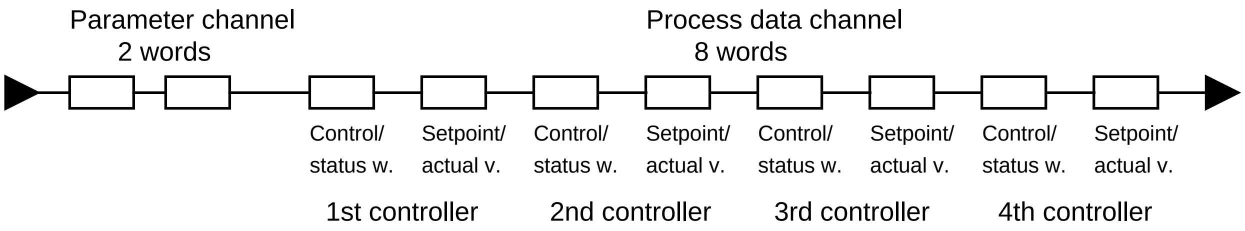

Figure 4: Bus interfacing of a 4-fold multiple controller

Bus interfacing of a 12-fold multiple controller (see Figure 6):

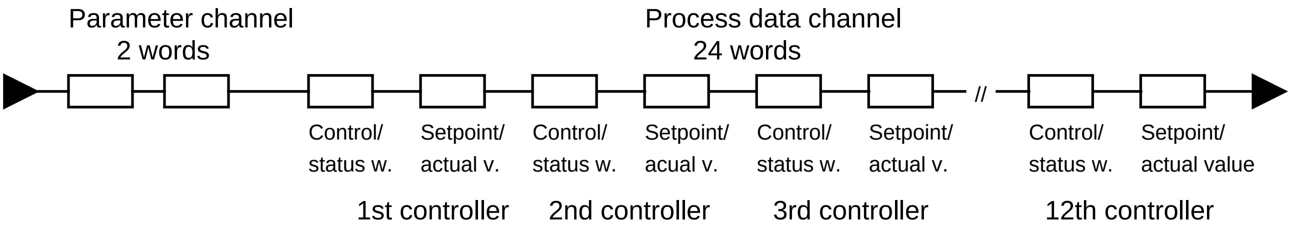

Figure 5: Bus interfacing of a 12-fold multiple controller

EXAMPLE: 4-fold multiple controller with 2 controllers in use

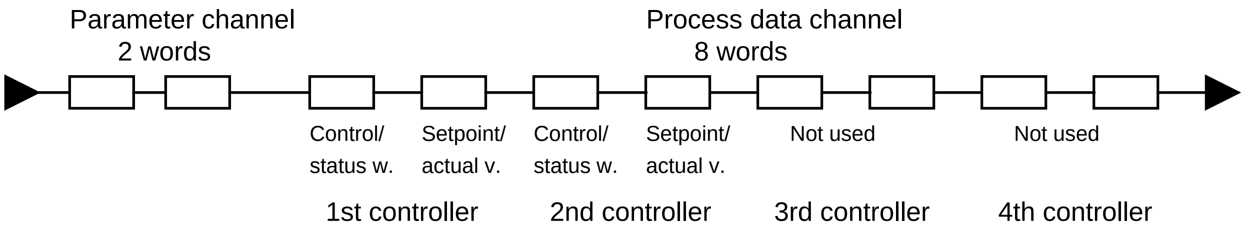

Figure 6: Bus interfacing of a 4-fold multiple controller

NOTE: In this case the two controllers use the first 4 words of the 8-word process data channel.

7.1. Device Data

Controllers are characterized by the fact that supplied setpoint values effect changes in the process variable. The optimum adaptation of controllers to the process requires additional parameters.

An analysis of the parameters in a control process leads to the following classification of data:

- Data to be transferred cyclically;
- Setpoint values to be transferred acyclically;
- Setting parameters;
- Information parameters.

Data To Be Transferred Cyclically

This data class is composed of control and status signals (e.g. paint pin status) and setpoint values/actual values that are to be transferred very fast (few ms) and cyclically.

In this case the signals are transferred via the process data channel.

Assignment of the control and status signals and of the setpoint values/actual values to the process data words:

EXAMPLE: for a single controller (see Figure 7):

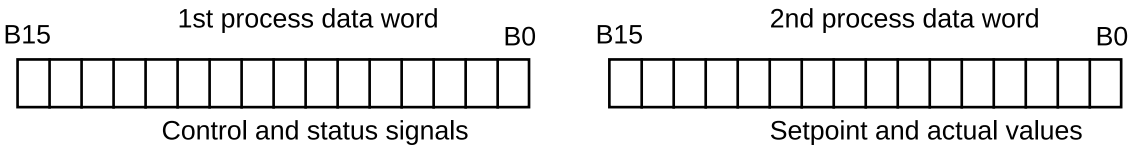

Figure 7: Example of the assignment of single-controller control and status signals and setpoint values/actual values to the process data words

Setpoint Values To Be Transferred Acyclically

This data class is composed of setpoint values to be transferred at low frequencies (once every few seconds). These parameters are transferred via the PCP channel.

Setting Parameters

This data class is composed of preset controller-specific initialization data that may be changed when required and is stored in non-volatile memory. These parameters are transferred via the PCP channel.

Information Parameters

This data class is composed data that is read out only for information (e.g. in the commissioning stage). These parameters are transferred via the PCP channel.

7.2. System Data

Typical system configurations are shown in Figure 7.

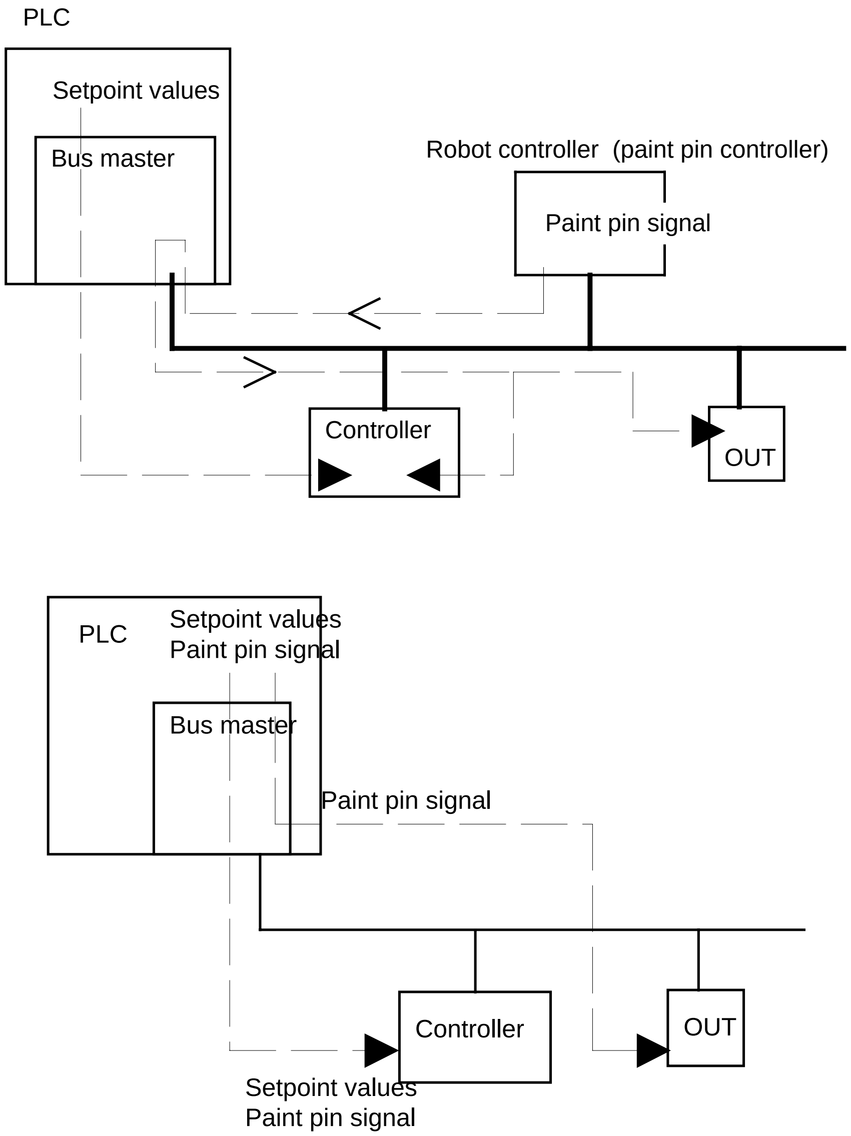

Figure 8: Description of typical system configurations

Typical Communication Time Periods:

- Setpoint values from the PLC to the controller in 30 - 200 ms
- Paint pin signals synchronously from the paint pin control unit to controller and OUT module within 3 bus cycles (interfacing via process data transport)
- Paint pin signals from the PLC to controller and OUT module within one bus cycle.

Typical Bus Scan Times:

2 - 20 ms

8. Application and Device Characteristics

This chapter describes the complete application from the point of view of communication. As shown in Figure 8, the application is divided into the following function blocks:

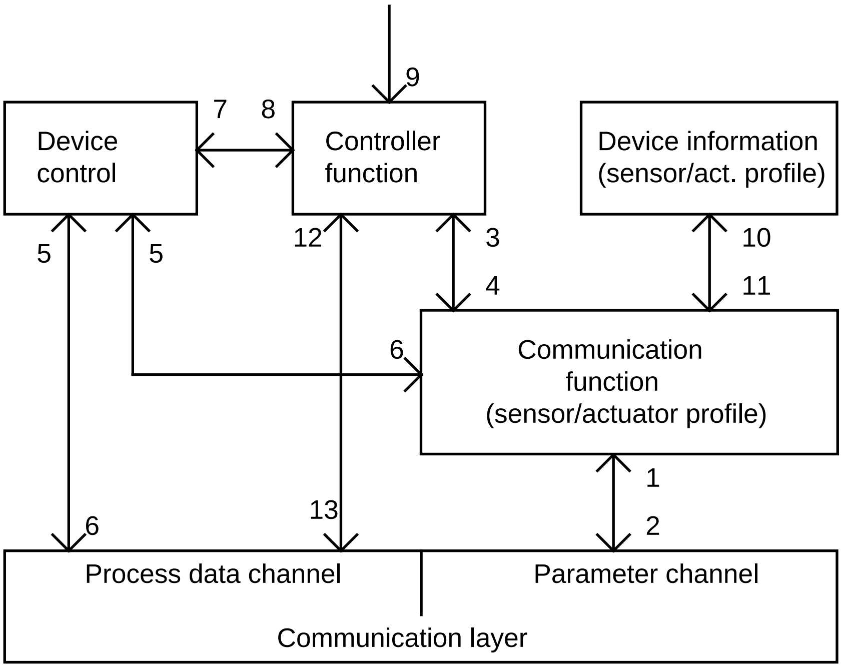

Figure 9: Function blocks of an application

Communication Function

The communication function executes all communication-specific functions via the parameter channel.

Device Control

The device control function block controls the complete device function (controller function).

Controller Function

The controller function executes all controller-specific functions.

Device Information

The device information keeps information on the device in non-volatile memory.

Communication Layer

The communication layer contains a layer 7 according to DIN 19245 Part 2 and a layer 2 according to the INTERBUS-S specification (see Chapters 5.4 - 5.5)

Interaction Between the Function Blocks

1 Data from the bus system

2 Data to the bus system

3 Supply of the controller parameters

4 Reading out the controller parameters

5 Commands to the device control (control word)

6 Device control state

7 Status of the controller function, fault

8 Control of the controller function

9 Actual value from the sensor

10 Storage of device information

11 Reading out the device information

12 Setpoint values to the controller

13 Actual values from the controller

8.1. Device Control

The device control function block controls the complete device operation (see Figure 9). The control sequence is described by a state machine. The device control block is influenced by the control word and the control byte, by internal signals and by malfunctions. The device control affects the controller functions. The status word is generated from the device state and internal signals and can be read out via the bus.

NOTE: Unlike a fault, a warning does not cause a state change in the device control.

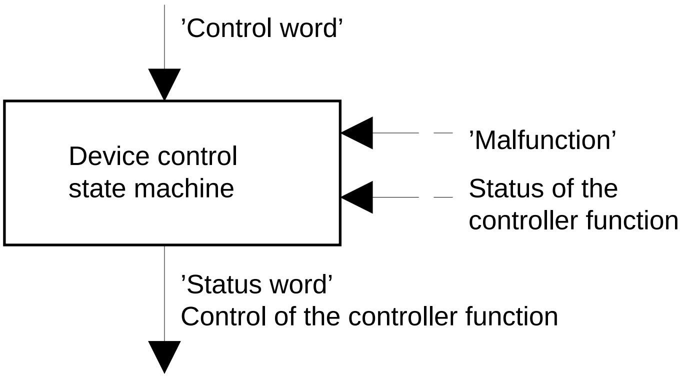

Figure 10: Device control function block

’Control Word’

The controller can be controlled by the bits in the ’control word’.

’Status Word’

The ’status word’ parameter gives information on the controller state and messages.

8.1.1. Device Control State Machine

The state machine (see Figure 10) describes the device states and the possible control sequence of the controller. A state represents a particular internal and external response. Using device control commands and internal events, the state can be changed to a execute a control sequence. The current state can be read out by way of the status word.

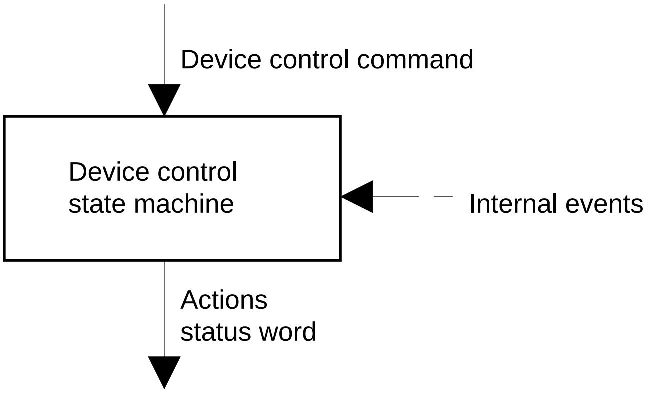

Figure 11: Device control state machine

State Diagram (see Figure 12)

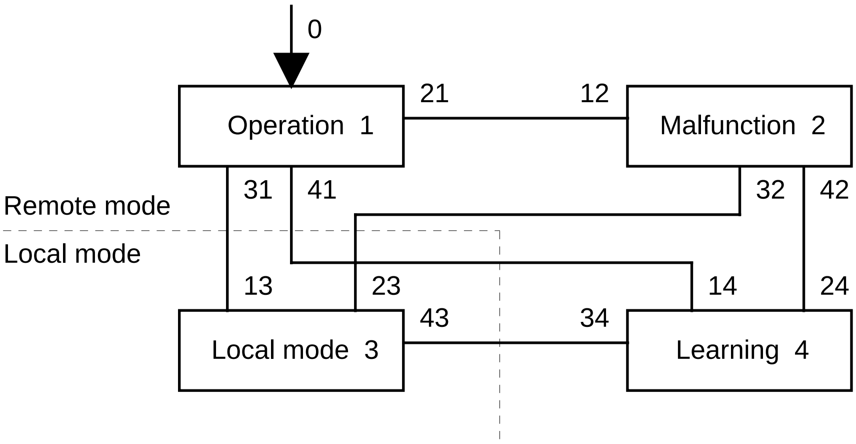

Figure 12: Device control state diagram

Device Control States

MALFUNCTION

In the malfunction state the controller sets all setpoint values to zero.

OPERATION

In the operation state all setpoint values are set. The following modes of operation are available:

- The emergency operation uses the secondary controlled variable from the list of secondary controlled variables;
- Ejecting keeps the manipulated variable constant;
- Purging uses the ’purge setpoint value’.

NOTE: If the actual value is not available (e.g. sensor error), the device can work without this actual value in the emergency mode.

LOCAL MODE

In the local mode all write accesses via the bus are disabled. All control signals, setpoint values and parameters are supplied locally. Exception: Writing to this bit is enabled unless the local mode was set locally on the device. All write accesses are carried out.

LEARNING

In this state the ’list of secondary controlled variables’ is initialized by the controller.

Device Control State Changes

Priorities:

Highest: Local mode
Fault
Learning
Lowest: Operation

0 Power Up -> OPERATION
Event: Power On
Action: None

12 OPERATION -> MALFUNCTION
Event: Device-internal malfunction
or External malfunction (e.g. defective sensor)
Action: None

21 MALFUNCTION -> OPERATION

Event: Reset malfunction = 0 1

Condition: Malfunction has been cleared

and Local mode = 0

and Learning = 0

Action: None

13 OPERATION -> LOCAL MODE

Event: Local mode = 1

Condition: No malfunction

Action: Adjustment to the setpoint values = 0

31 LOCAL MODE -> OPERATION

Event: Local mode = 1 0

Condition: Learning = 0

and No malfunction

Action: Adjustment to the current values of the ’setpoint value-x’ parameters

14 OPERATION -> LEARNING

Event: Lernen = 1

Condition: Local mode = 0

And No malfunction

Action: None

41 LEARNING -> OPERATION

Event: Learning = 0

or Learning completed (result shown in ’learning result’ parameter)

Condition: Local mode = 0

and No malfunction

Action: Adjustment to the current values of the ’setpoint values-x’

23 MALFUNCTION -> LOCAL MODE

Event: The malfunction is acknowledged locally

Condition: The malfunction has been cleared

and Local mode = 1

Action: None

32 LOCAL MODE -> MALFUNCTION

Event: Device-internal malfunction

or External malfunction (e.g. defective sensor)

Action: None

24 MALFUNCTION -> LEARNING

Event: Reset malfunction = 0 1

Condition: The malfunction has been cleared

and Local mode = 0

and Learning = 1

Action: None

42 LEARNING -> MALFUNCTION

Event: Device-internal malfunction

or External malfunction (e.g. defective sensor)

Action: None

34 LOCAL MODE -> LEARNING

Event: Local mode = 1 0

Condition: Learning = 1

and No malfunction

Action: None

43 LEARNING -> LOCAL MODE

Event: Local mode = 1

Condition: No malfunction

Action: Adjustment to the setpoint values = 0

The state change takes place only when the actions have been completely carried out. The order of actions corresponds to their order of processing when a state change takes place. After the actions have been completely executed the next state is reached and new commands are accepted.

8.1.2. Control Word

By means of logical operations, the ’control word’ and the internal signals generate the device control commands which act on the device control state machine. This initiates functions and defines operating states of the device. The device control words consists of 16 bits. The meanings of these bits are shown in Table 1:

Table 1: Meanings of the control word bits

	Controller					
Bit	PQ	DP	AF	SP	HV	PR
0	Run	Run	Run	Run	Run	Run
1	Paint pin	Reserved	Paint pin	Paint pin	Reserved	Reserved
2	Local mode	Local mode	Local mode	Local mode	Local mode	Local mode
3	Emergency operation	Reserved	Emergency operation	Emergency operation	Emergency operation	Emergency operation
4	Warning acknowl.	Warning acknowl.	Warning acknowl.	Warning acknowl.	Warning acknowl.	Warning acknowl.
5	Learning	Reserved	Learning	Reserved	I/U constant	Reserved
6	Reset qty. counter	Reset qty. counter	Reserved	Reserved	Dynamic threshold	Reserved
7	Malfunction acknowl.	Malfunction acknowl.	Malfunction acknowl.	Malfunction acknowl.	Malfunction acknowl.	Malfunction acknowl.
8	Ejection	Reserved	Reserved	Reserved	Reserved	Reserved
9	Pressure mode	Reserved	Pressure mode	Reserved	Reserved	Reserved
10	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
. . .	. . .	. . .	. . .	. . .	. . .	. . .
15	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

PQ paint quantity controller	SP speed controller
DP dosing pump	HV high-voltage controller
AF air flow controller	PR pressure controller

Run

1 - The controller adjusts to the current setpoint values.

0 - The controller adjusts to all setpoint values = 0.

Paint Pin

1 - Indicates that the paint pin valve is open.

0 - Indicates that the paint pin valve is closed.

Local Mode

In the local mode all write accesses via the bus are disabled. All control signals, setpoint values and parameters can be locally supplied. Exception: Writing to this bit is enabled unless the local mode was set locally on the device.

> Note:
> When the PLC or host computer sets this bit, the PLC or host computer can switch back to the remote mode by resetting the bit. When the local mode is switched on locally on the device, the PLC or host computer is unable to switch back to the remote mode.

1 - Write access disabled

0 - Write access enabled

Malfunction Acknowledgment

0->1 - State change into the operation state

Emergency Operation

1 - The controller uses for its control operations its ’list of secondary controlled variables’ (e.g. the quantity/pressure ratio with the paint quantity controller).

0 - Normal mode

Warning Acknowledgment

0->1 - This edge resets the warning message.

Learning

1 - The controller is in the learning mode

0 - The controller is not in the learning mode

Ejecting

1 - The manipulated variable is kept constant

0 - The controller is not in this mode

Pressure Mode

1 - The setpoint value_2 parameter is used for the control operation

0 - The controller is not in this mode

Reset Quantity Counter

1 - The ’quantity counter’ parameter is set to ’0’ and held.

0 - The quantity is sensed with the ’quantity counter’ parameter.

I/U Constant

1 - The setpoint value-1 is interpreted as a current setpoint value

0 - The setpoint value-1 is interpreted as a voltage setpoint value

Dynamic Threshold

1 - The setpoint value-3 is active

0 - The setpoint value-3 is inactive.

8.1.3. Status Word

The status word provides information on the device state as well as messages (see Table 2).

Table 2: Meaning of the status word bits

	Controller					
Bit	PQ	DP	AF	SP	HV	PR
0	Ready	Ready	Ready	Ready	Ready	Ready
1	Local mode	Local mode	Local mode	Local mode	Local mode	Local mode
2	Emergency operation	Reserved	Emergency operation	Emergency operation	Emergency operation	Emergency operation
3	Malfunction	Malfunction	Malfunction	Malfunction	Malfunction	Malfunction
4	Learning	Reserved	Learning	Speed OK	High voltage OK	Reserved
5	Actual < setpt.	Reserved	Actual < setpt.	Actual < setpt.	Reserved	Actual < setpt.
6	Actual > setpt.	Reserved	Actual > setpt.	Actual > setpt.	Reserved	Actual > setpt.
7	Warning	Warning	Warning	Warning	Warning	Warning
8	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
. . .						
15	Reserved	Reserved	reserved	Reserved	Reserved	Reserved

PQ paint quantity controller

DP dosing pump

AF air flow controller

SP speed controller

HV high-voltage controller

PR pressure controller

Device States

The device states are indicated in the status word by the bit combinations stated in Table 3:

Table 3: Device states according to the bit combinations in the status word

State	Ready Bit 0	Learning Bit 4	Mal-function Bit 3	Local mode Bit 1
LOCAL MODE	x	x	x	1
MALFUNCTION	x	x	1	0
LEARNING	x	1	0	0
OPERATION	1	0	0	0

NOTE: The high-voltage controller does not support the learning state.

Learning

The device is in the ’learning’ state

High Voltage OK

1 - The voltage/current actual value is within the tolerance range (+/- 10%)

0 - Actual value outside tolerance or malfunction

Speed OK

1 - Minimum speed for paint application reached

0 - Actual value below minimum speed or malfunction

Warning

Collective indication of manufacturer-specific or standard warnings. Bit = 1 indicates a warning.

Note:
The device manufacturer defines whether an event is indicated as a warning, or whether a transition to the malfunction state is to take place.

Actual<Setpoint

The actual value is below the supplied setpoint value.

Actual>Setpoint

The actual value is higher than the supplied setpoint value.

Reserved

These status bits are reserved for profile extensions. The bits must be set to "0" as long as they are not assigned to a defined status.

8.1.4. Malfunction Function

This function manages the ’malfunction code’ parameter (see Figure 12). As the result of a drive controller malfunction, the ’malfunction code’ parameter is set to the corresponding value (see malfunction list). The parameter is reset to the value 0 by the malfunction reset action of the device control.

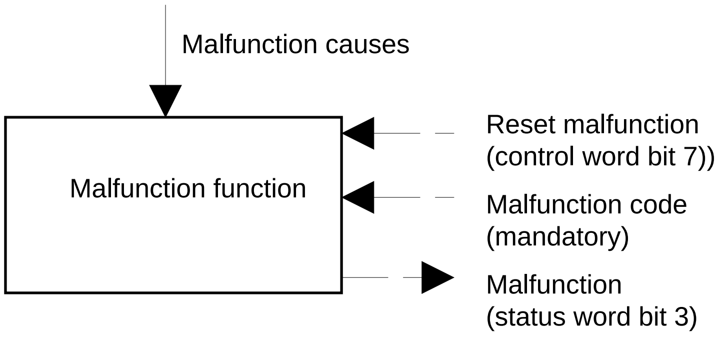

Figure 13: Malfunction function

’Malfunction Code’

The ’malfunction code’ is represented as an octet string with a length of 2 bytes. It is coded hierarchically, ranging from a coarse distinction to one that becomes increasingly finer.

Bit	Grouping
15 . . . 12	Main groups
11 . . . 8	Subgroups
7 . . . 0	Details

The parameter is assigned a value unequal to zero when the controller is in the malfunction state. The parameter is assigned the value 0 when the controller is not in the malfunction state (see Table 4).

When there is precisely one cause of a malfunction, the value assigned to this cause in the ’malfunction code’ parameter can be read out unchanged until the malfunction state no longer applies. This is the case whenever the cause of the malfunction has been remedied and the malfunction reset command has been issued.

When there are several simultaneous causes of a malfunction, one of them is indicated in the "malfunction code" parameter. When only the indicated malfunction cause is remedied and the reset malfunction command is issued, the malfunction state is not terminated because the other malfunction causes still apply. One of these malfunction causes is then indicated in the "malfunction code" object.

Object class:	Mandatory
Access:	Read only
Process data mapping:	Not possible
Unit:	None
Value range:	0 to 65535
Mandatory range:	-
Substitute value:	-

Table 4: Malfunction codes and malfunction causes

Code hex	Meaning
0000	**No malfunction**
1000	**General malfunction**
1800	**General malfunction (manufacturer-specific)**
...	**General malfunction (manufacturer-specific)**
1FFF	**General malfunction (manufacturer-specific)**
2000	**Current**
2100	**Current, device input side**
2110	Short circuit/short to ground
2120	Short to ground
2130	Short circuit
2200	**Current, device-internal**
2211	Current, device internal no. 1
2212	Current, device-internal no. 2
2220	Continuous overcurrent
2230	Short circuit/short to ground
2240	Short to ground
2250	Short circuit
2300	**Current, device output side**
2310	Continuous overcurrent
2311	Continuous overcurrent no. 1
2312	Continuous overcurrent no. 2
2320	Short circuit/short to ground
2330	Short to ground
2340	Short circuit
3000	**Voltage**
3100	**AC supply voltage**
3110	AC supply overvoltage
3120	AC supply undervoltage
3130	Phase failure
3140	AC supply frequency
3200	**Voltage, device-internal**
3210	Overvoltage, device-internal
3220	Undervoltage, device-internal
3230	Charging error

(continued on next page)

Table 4: Malfunction codes and malfunction causes (continued from previous page)

Code hex.	Meaning
3300	**Output voltage**
3310	Output overvoltage
3320	Output undervoltage
4000	**Temperature**
4100	**Ambient temperature**
4110	Ambient overtemperature
4120	Ambient undertemperature
4200	**Device temperature**
4210	Device overtemperature
4220	Device undertemperature
4300	**Drive temperature**
4400	**Supply temperature**
4410	Supply overtemperature
4420	Supply undertemperature
5000	**Device hardware (only inside the controller housing)**
5100	**Supply**
5110	Low-voltage supply
5111	+/- 15 V supply
5112	+ 24 V supply
5113	+ 5 V supply
5120	Air supply
5130	Lacquer supply
5200	**Control**
5210	Measurement circuits
5220	Computing circuits
5300	**Operator control unit**
5400	**Power section**
5500	**Communication with additional module**
5510	Interface no. 1
5520	Interface no. 2
6000	**Device software**
6010	Software reset (watchdog)
6100	**Internal software**
6200	**User software**
6300	**Data record**
6310	Parameter loss
6320	Parameter error
7000	**Defective additional module**
7100	**Power**
7200	**Measurement circuit**
7300	**Sensor**
7301	Defective sensor 1
730F	Defective sensor 15
7400	**Computing circuit**
7500	**Cascade error**
7600	**Data memory**
7700	**Cable error**
7701	Defective cable 1
770F	Defective cable 15

continued on next page

Table 4: Malfunction Codes and Malfunction Causes (end)

Code hex	Meaning
8000	**Monitoring**
8100	**Communication**
8110	Process data monitoring
8120	Host monitoring
8200	**Closed-loop control**
8210	Deviation Setpoint value > actual value; the deviation lasts longer than a given time (manufacturer-specific)
8211	Max. manipulated value reached
8220	Deviation Setpoint value< actual value; the deviation lasts longer than a given time (manufacturer-specific)
8221	Max. manipulated value reached
...	Reserved for profile-specific control errors
827f	Reserved for profile-specific control errors
8280	Manufacturer-specific control errors
...	Manufacturer-specific control errors
82FF	Manufacturer-specific control errors
9000	**External malfunction**
F000	**Additional functions**

Codes that are not listed are reserved.

NOTE: Every device manufacturer must precisely define the malfunction codes for the respective device.

Error Message

Yes, see read or write function.

Mapping the Device Function onto Communication

’Malfunction code’ object description (see Table 5)

Table 5: Object-description: ’malfunction code’

Object attribute	Value hex	Meaning
Index	603F	Malfunction code
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	0A	Octet string
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0001	Read all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

8.1.5. Warning Function

The warning function manages the ’warning code’ parameter (see Figure 13). The ’warning code’ parameter is set to the corresponding value by a controller warning. When the warning is no longer applied, the parameter is set to 0.

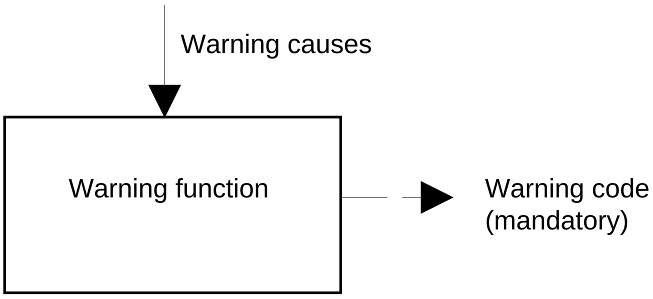

Figure 14: Warning function

’Warning Code’

The ’warning code’ is represented as an octet string with a length of 2 bytes. It is coded hierarchically, ranging from a coarse distinction to one that becomes increasingly finer.

Bit	Grouping
15 . . . 12	Main groups
11 . . . 8	Subgroups
7 . . . 0	Details

The parameter is assigned a value that is not equal to zero when the controller is in the warning state. The parameter is assigned the value 0 when the controller is not in the warning state.

When there is precisely one cause of a fault, the value assigned to this cause in the ’malfunction code’ can be read out unchanged until the warning state no longer applies. This is the case when the cause of the warning has been remedied.

When there are several simultaneous causes of a malfunction, one of them is indicated in the ’warning code’ parameter. One of these warning causes is then indicated in the ’warning cause’ object.

Object class:	Mandatory
Access:	Read only
Process data mapping:	Not possible
Unit:	No
Value range:	0 to 65535
Mandatory range:	-
Substitute value:	-

NOTE: The warning codes are identical with the malfunction codes, i.e. all malfunctions can also be indicated as warnings.

Mapping the Device Function onto Communication

Object description: ’Warning code’ (see Table 6)

Table 6: Object description: ’warning code’

Object attribute	Value hex	Meaning
Index	603E	Warning code
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	0A	Octet string
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0001	Read all
Local address	xxxx	Manufacturer-specific
Extension	-	Not existent

8.2. Controller Function

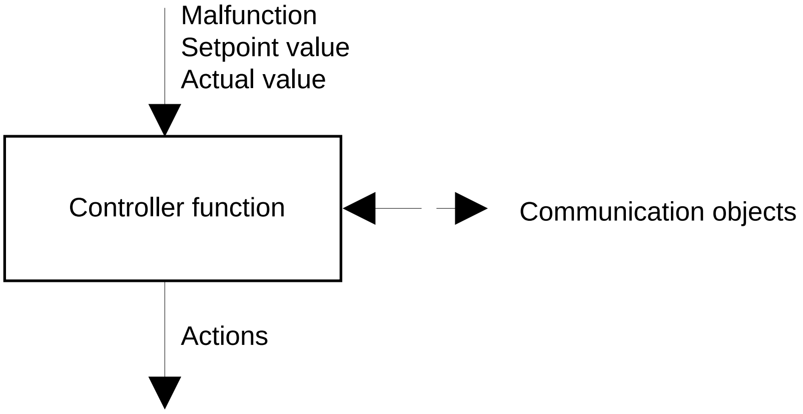

Figure 15: Controller function

The controller function (see Fig. 14) consists of functions describing the control of physical process variables.

The controller function is composed of the following subfunctions:

- Setpoint value/actual value scaling
- Sensor error function
- Controller deviation function
- Learn function

The controller functions can be parameterized with the following parameters:

- Resolution (physical value/numerical value);
- Setpoint value min./max.;
- Offset.

The controller function supplies the following output parameter:

- Actual value.

8.2.1. Setpoint Value/Actual Value Scaling

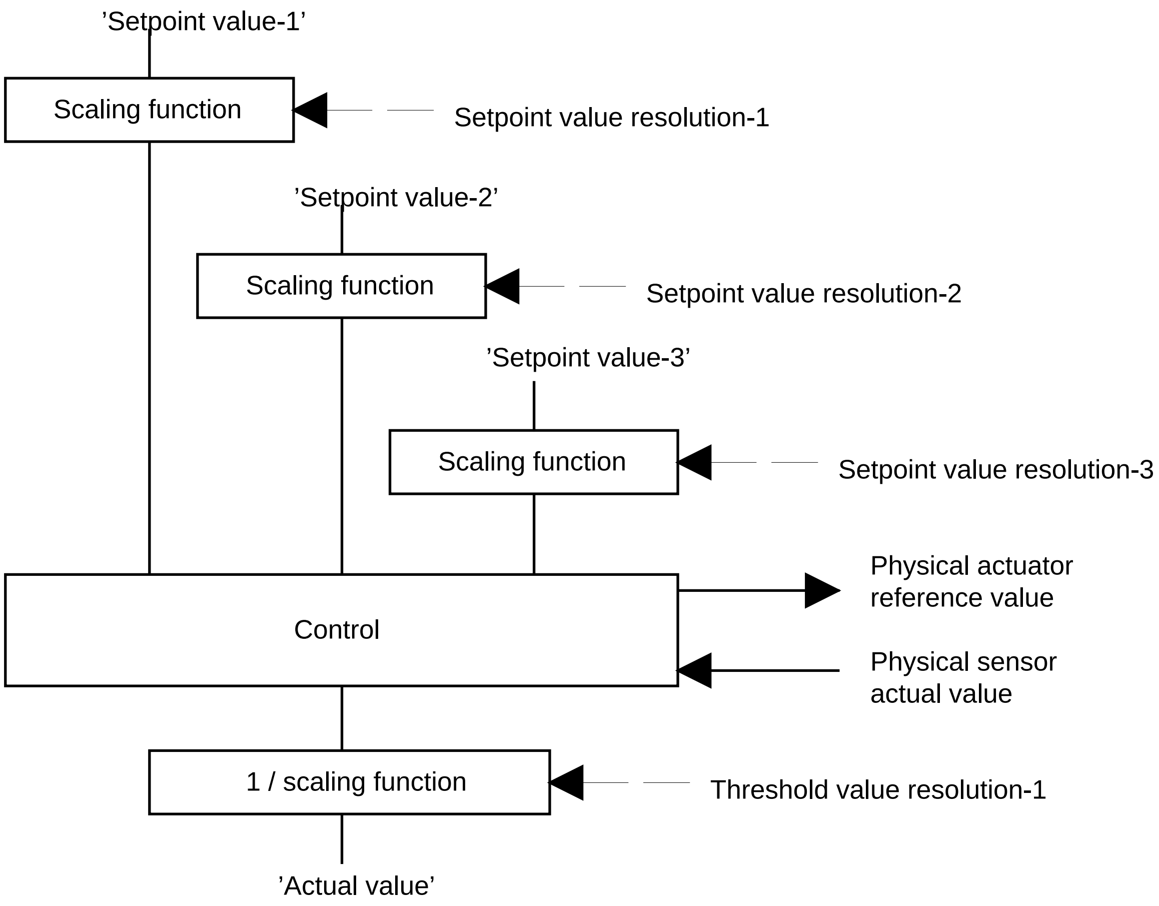
’Setpoint value-1’
Scaling function
Setpoint value resolution-1
’Setpoint value-2’
Scaling function
Setpoint value resolution-2
’Setpoint value-3’
Scaling function
Setpoint value resolution-3
Physical actuator reference value
Control
Physical sensor actual value
1 / scaling function
Threshold value resolution-1
’Actual value’

Figure 16: Factor function

’Setpoint Value_1’

The ’setpoint value_1’ parameter is the 1st setpoint value for the controller. According to the controller type a paint quantity, pressure, rotational speed etc. is set.

Object class:	mandatory
Access:	Read and write
Process data mapping:	Possible
Unit:	See ’setpoint value description data’
Value range:	-32768 to 32767
Parameter description:	See setpoint value_1 description
Mandatory range:	-
Substitute value:	-0

Controller type	Meaning	Default setting for	
		Unit	Resolution
Paint quantity controller	Quantity of paint	ml/min	1
Pressure controller	Pressure	mbar	1
Speed controller	Rotational speed	1/min	10
High-voltage controller	Voltage	kV	1
Current controller	Current	µA	1
Air flow controller	Air flow rate	l/min	1
Dosing pump	Quantity of paint	ml/min	1

Setpoint Value_2

The ’setpoint value_2’ parameter is the 2nd setpoint value for the controller. According to the controller type a paint flow, pressure, rotational speed etc. is set.

Object class:	Mandatory
Access:	Read and write
Process data mapping:	possible
Unit:	See ’setpoint value description data’
Value range:	-32768 to 32767
Mandatory range:	-
Substitute value:	- 0

Controller type	Meaning	Default setting for	
		Unit	Resolution
Paint quantity controller	Paint pressure	mbar	1
Pressure controller	Reserved		
Speed controller	Reserved		
High-voltage controller	Current threshold static	µA	1
Current controller	Min. voltage	KV	1
Air flow controller	Air pressure	mbar	1
Dosing pump	Reserved		

Setpoint Value_3

The ’setpoint value_3’ parameter is the 3rd setpoint value for the controller. According to the controller type a paint quantity, pressure, rotational speed etc. is set.

Object class:	Mandatory
Access:	Read and write
Process data mapping:	Possible
Unit:	-See ’setpoint value description data’
Value range: Paint quantity controller: High-voltage controller	 1 to 230 0 to 32767
Mandatory range:	-
Substitute value:	-0

Controller type	Meaning	Default setting for	
		Unit	Resolution
Paint quantity controller	Paint number		
Pressure controller	Reserved		
Speed controller	Reserved		
High-voltage controller	Current threshold dynamic	µA/sec	1
Current controller	Reserved		
Air flow controller	Reserved		
Dosing pump	Reserved		

Actual Value_1

Object class:	Mandatory
Access:	Read only
Process data mapping:	Possible
Unit:	See ’setpoint value description data’
Value range:	-32768 to 32767
Mandatory range:	-
Substitute value:	-0

Controller type	Meaning	Default setting for	
		Unit	Resolution
Paint quantity controller	Paint quantity	ml/min	1
Pressure controller	Pressure	mbar	1
Speed controller	Rotational speed	1/min	10
High-voltage controller	Voltage	kV	1
Current controller	Current	µA	1
Air flow controller	Air flow	l/min	1
Dosing pump	Paint quantity	ml/min	1

Actual Value_2

Object class:	Optional
Access:	Read only
Process data mapping:	Possible
Unit:	See ’setpoint value description data’
Value range:	-32768 to 32767
Mandatory range:	-
Substitute value:	-0

Controller type	Meaning	Default setting for	
		Unit	Resolution
Paint quantity controller	Pressure actual value	mbar	1
Pressure controller	Reserved		
Speed controller	Turbine pressure	mbar	1
High-voltage controller	Current threshold, static	µA	1
Current controller	Min. voltage	KV	1
Air flow controller	Pressure actual value	mbar	
Dosing pump	Reserved		

Actual Value_3

Object class:	Optional
Access:	Read only
Process data mapping:	Possible
Unit:	See ’setpoint value description data’
Value range:	-32768 to 32767
Mandatory range:	-
Substitute value:	-0

Controller type	Meaning	Default setting for	
		Unit	Resolution
Paint quantity controller	Total quantity	ml	1
Pressure controller	Reserved		
Speed controller	Reserved		
High-voltage controller	Reserved		
Current controller	Reserved		
Air flow controller	Reserved		
Dosing pump	Total quantity	ml	1

Parameter Description

The parameters setpoint value_1 to 3 and actual value_1 to 3 are described by the following parameters:

- Value type_1 to value type_3
- Variable index_1 to variable index_3
- Unit index_1 to unit index_3

Value type_1 to 3

This parameter defines how the controller parameters ’setpoint value_1’, ’setpoint value_2’ and ’setpoint value_3’ of the controller are to be interpreted.

	Meaning
-128 to -1	Manufacturer-specific
0	Reserved
1	Pressure
2	Rotational speed
3	Paint quantity
4	Air flow
5	Dosing pump
6	High voltage
7-127	Reserved

Setpoint Value Variable Index_1 to 3

The setpoint value variable index is a parameter that contains in encoded form information on the physical variable of the actual value 1 to 3. Refer to the sensor/actuator profile for the coding.

Object class:	Optional
Access:	Read only
Process data mapping:	Not possible
Unit:	-
Value range:	-128 to 127
Mandatory range:	-
Substitute value:	-0

Actual Value Variable Index_1 to 3

The actual value variable index_1 to 3 is a parameter that contains in encoded form information on the physical variable of the actual value_1 to 3. Refer to the sensor/actuator profile for the coding.

Object class:	Optional
Access:	Read only
Process data mapping:	Not possible
Unit:	-
Value range:	-128 to 127
Mandatory range:	-
Substitute value:	-0

Setpoint Value Unit Index_1 to 3

The setpoint unit index_1 to 3 is a parameter that contains in encoded form information on the unit of measurement of the setpoint value_1 to 3. Refer to the sensor/actuator profile for the coding.

Object class:	Optional
Access:	Read only
Process data mapping:	Not possible
Unit:	-
Value range:	-128 to 127
Mandatory range:	-
Substitute value:	-0

Actual Value Unit Index_1 to 3

The actual value unit index_1 to 3 is a parameter that contains in encoded form information on the unit of measurement of the actual value_1 to 3. Refer to the sensor/actuator profile for the coding.

Object class:	Optional
Access:	Read only
Process data mapping:	Not possible
Unit:	-
Value range:	-128 to 127
Mandatory range:	-
Substitute value:	-0

Setpoint Value Resolution_1 to 3

This parameter indicates the resolution of the setpoint value_1 to 3 parameter.

Object class:	Optional
Access:	Read, optionally write
Process data mapping:	Not possible
Unit:	-
Value range:	0 to 65535
Mandatory range:	-
Substitute value:	1

Actual Value Resolution_1 to 3

This parameter indicates the resolution of the actual value_1 to 3 parameter.

Object class:	Optional
Access:	Read, optionally write
Process data mapping:	Not possible
Unit:	-
Value range:	0 to 65535
Mandatory range:	-
Substitute value:	1

The resolution parameter is represented as follows:

$$R = \frac{PM}{N}$$

where

R is the resolution;

PV is the physical variable;

N is the numerical value.

Structure of the parameters:

the physical variable
the numerical value

The physical setpoint value is calculated according to the following formula

$$PV = S * R$$

where

PS is the physical setpoint value;

S is the setpoint value;

R is the resolution.

EXAMPLE 1:

$$R = \frac{16 \text{ ml}}{65\ 565}$$

$$PS = 32\ \ 782 \times \frac{16 \text{ ml}}{65\ 565} = 8\text{ml}$$

EXAMPLE 2:

$$R = \frac{1\ \mathrm{ml}}{1}$$

$$PS = 333 \times \frac{1\ \mathrm{ml}}{1} = 333\ \mathrm{ml}$$

EXAMPLE 3:

$$R = \frac{10\ \mathrm{rpm}}{1}$$

$$PS = 1\ 000 \times \frac{10\ \mathrm{rpm}}{1} = 10\ \ 000\ \mathrm{rpm}$$

Mapping the Device Function onto Communication

Object description: ’setpoint value_1’ (see Table 7)

Table 7: Object description: ’setpoint value_1’

Object attribute	Value hex	Meaning
Index	6050	Setpoint value_1
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	03	Integer16
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’setpoint value_2’ (see Table 8)

Table 8: Object description: ’setpoint value_2’

Object attribute	Value hex	Meaning
Index	6058	Setpoint value_2
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	03	Integer16
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’setpoint value_3’ (see Table 9)

Table 9: Object description: ’setpoint value_3’

Object attribute	Value hex	Meaning
Index	6060	Setpoint value_3
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	03	Integer16
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’actual value_1’ (see Table 10)

Table 10: Object description: ’actual value_1’

Object attribute	Value hex	Meaning
Index	6068	Actual value_1
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	03	Integer16
Length	02	2 bytes
Password	00	No password
Access- groups	00	No access groups
Access rights	0003	Read all
Local-Address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’actual value_2’ (see Table 11)

Table 11: Object description: ’actual value_2’

Object attribute	Value hex	Meaning
Index	6070	Actual value_2
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	03	Integer16
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’actual value_3’ (see Table 12)

Table 12: Object description: ’actual value_3’

Object attribute	Value hex	Meaning
Index	6078	Actual value-3
Variable name	-	Not existent
Object code	07	Simple variable
Data type index	03	Integer16
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all
Local address	xxxx	Manufacturer-specific
Extension	-	Not existent

Object description: variable index_1 to 3’ (see Table 13)

Table 13: Object description: variable index_1 to 3’

Object attribute	Value hex	Meaning
Index	6052	Setpoint value - variable index_1
	605A	Setpoint value - variable index_2
	6062	Setpoint value - variable index_3
		Actual value - variable index_1
		Actual value - variable index_2
		Actual value - variable index_3
Variable name	-	Variable index
Object code	07	Simple variable
Data type index	02	Integer8
Length	01	1 byte
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: unit_index-1 to 3’ (see Table 14)

Table 14: Object description: unit_index_1 to 3

Object attribute	Value hex	Meaning
Index	6053	Setpoint value - unit ind._1
	605B	Setpoint value - unit ind._2
	6063	Setpoint value - unit ind._3
		Actual value - unit index_1
		Actual value - unit index_2
		Actual value - unit index_3
Variable name	-	Unit index
Object code	07	Simple variable
Data type index	02	Integer8
Length	01	1 byte
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Not existent

Object description: ’setpoint value resolution’ (see Table 15)

Table 15: Object description: setpoint value - resolution’

Object attribute	Value hex	Meaning
Index	6055	Resolution-1
	605D	Resolution-2
	6065	Resolution-3
Variable name	-	Resolution
Object code	08	Array
Data type index	03	Integer16
Length	2	2 bytes
Number of elements	2	
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Not existent

8.2.2. Sensor Error Function

When a sensor error is present for more than the sensor monitoring time, the sensor error function (see Figure 16) indicates a sensor error with the ’malfunction’ device control command.

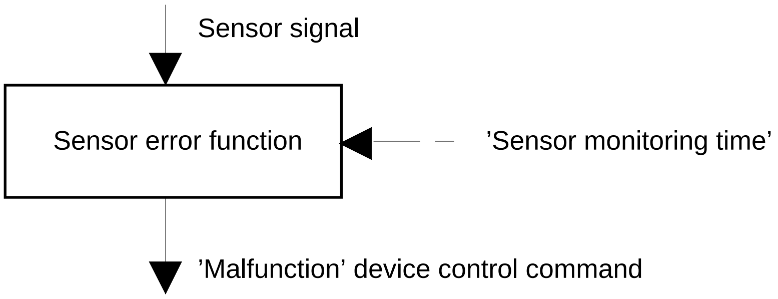

Figure 17: Sensor error function

’Sensor Monitoring Time’

The ’sensor monitoring time’ is the length of time an error has to be present until it is indicated by the status bit. This time period is specified in ms. There may be manufacturer-specific restrictions to the value range. The sensor error function is disabled with the value 65535.

Object class:	Optional
Access:	Read and write
Process data mapping:	Not possible
Value range:	0 to 65535
Mandatory range:	Manufacturer-specific
Substitute value:	65535 (disabled)

Mapping the Device Function to Communication

Object description: ’sensor monitoring time’ (see Table 16)

Table 16: Object description: ’sensor monitoring time’

Object attribute	Value hex	Meaning
Index	6083	Sensor monitoring time
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	06	Unsigned16
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

8.2.3. Deviation Function

This function (see Figure 17) compares the current setpoint value with the current actual value. If the setpoint value is greater than the actual value + tolerance, the ’actual value < setpoint value’ status bit is set. If the actual value is greater than the actual value + tolerance , the ’actual value > setpoint value’ status bit is set.

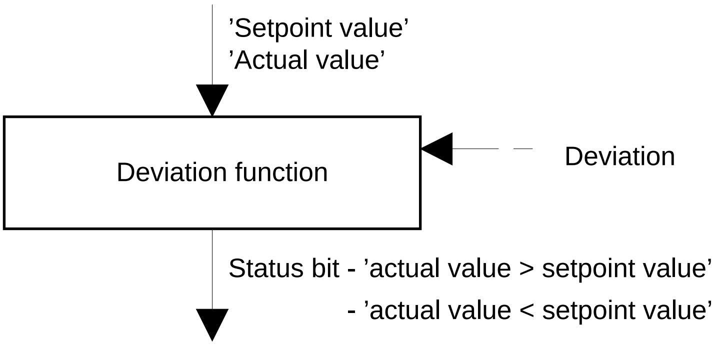
’Setpoint value’
’Actual value’
Deviation function
Deviation
Status bit - ’actual value > setpoint value’
- ’actual value < setpoint value’

Figure 18: Deviation function

’Deviation’

The tolerance is relative to the variable of the setpoint value.

$$T = \frac{S \times D}{100}$$

where

T is the tolerance;

S is the setpoint value;

D is the deviation;

Default value = 5%,

During start-up, the value that is stored in non-volatile memory is inititalized. It may differ from the default value (5%) if it was changed by the user.

Object class:	Optional
Access:	Read only
Process data mapping:	Not possible
Unit:	Per cent
Default value:	5 or the value that has been stored in non-volatile memory
Value range:	0 to 255
Mandatory range:	0 to 255
Substitute value:	5

Mapping the Device Function onto Communication

Object description: ’deviation’ (see Table 17)

Table 17: Object description: ’deviation’

Object attribute	Value hex	Meaning
Index	6081	
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	05	Unsigned8
Length	02	2 bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read all, write all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

8.2.4. Learn Function

The learn function (see Figure 19) initializes the 'list of secondary controlled variables' parameter for the paint color defined by the paint color number. Internally there may be a list of secondary controlled variables for each paint color. The 'learn paint color' parameter defines which one of the internal lists of secondary controlled variables is accessed.

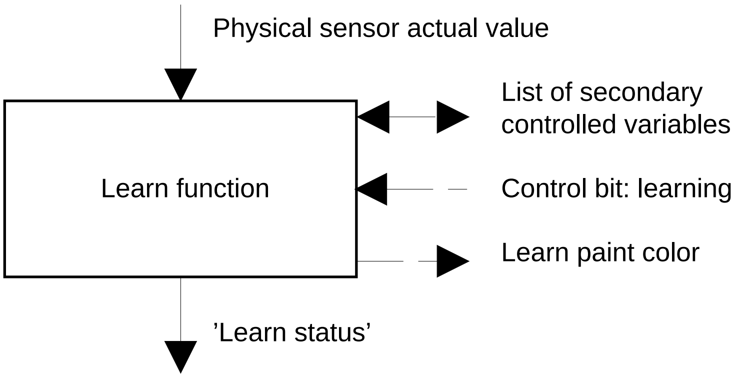
Physical sensor actual value
List of secondary controlled variables
Learn function
Control bit: learning
Learn paint color
'Learn status'

Figure 19: Learn function

'List of secondary controlled variables'

The elements contained in this parameter are setpoint value-1 and the related secondary controlled variable. These two elements are the main points of the function: secondary controlled variable = f(setpoint value-1).

E.g. paint pressure = f(paint flow)

Object class:	Optional
Access:	Read and write
Process data mapping:	Not possible
Value range:	Integer16
Unit:	-
Mandatory range:	Integer16
Substitute value:	-

Learn status
Setpoint value-1
Secondary controlled variable
Setpoint value-1
Secondary controlled variable
. . .
Setpoint value-1
Secondary controlled variable

’Learn Paint Color’

This parameter defines which one of the internal lists of controlled variables is accessed.

Object class:	Optional
Access:	Read and write
Process data mapping:	Not possible
Unit:	-
Value range:	Unsigned8
Mandatory range:	0 to 255
Substitute value:	-

’Learn Status

This parameter defines the status in the learn mode. The parameter is an array. The learn status is stored in the elements of the array for the corresponding paint colors. The subindex for addressing the elements corresponds to the paint color number.

> Note:
> Arrays can only be transferred up to a size of 250 bytes.

Subindex =

1	Learn status for paint color number = 1
2	Learn status for paint color number = 2
n	Learn status for paint color number = n

Learn status	Meaning
-128 to - 1	Learning unsuccessfully completed (manuf.-specific)
0	Learn mode never activated before
1	Learning successfully completed
2	Learn mode is active

Object class:	Optional
Access:	Read only
Process data mapping:	Not possible
Unit:	-
Value range:	Unsigned8
Mandatory range:	0 to 255
Substitute value:	-

8.3. Sensor/Actuator Functions

8.3.1. Communication Function

See sensor/actuator profile 12, Chapter ’Communication Functions’.

> NOTE: The Initiate service is a request to establish a connection. The user of the connection establishment must set the ’profile number’ service parameter to the value 0081 hex.

8.3.2. Device Information

See sensor/actuator profile 12, Chapter ’Device Information’ .

9. Data Structures

This chapter lists the data structures of all user data.

The parameters of a process controller are stored in an object dictionary the structure of which is specified in Table 18. This object dictionary is used for describing the parameters. It contains data on the index, data type, object type, the access rights, etc. The index is used for addressing the parameter when writing or reading is to take place. This object dictionary can be read out with the ’read communication object list’ function.

Table 18: Structure of the Object Dictionary

Index	Object dictionary
0000	OV object dictionary
0001 001F	(DIN 19245/Part 2) Static type dictionary
0020 003F	(Profiles) Static type dictionary
0040 005F	(Free for manufacturer) Static type dictionary
. . .	. . .
2000 5FFF	(Free for manufacturer) Static object dictionary
6000 603F	(Devices according to server profile) Static object dictionary
6040 9FFF	(Profiles) Static object dictionary
A000 BFFF	(Free for manufacturer) Dynamic variable list dictionary
C000 DFFF	(Profiles) Dynamic variable list dictionary
E000 EFFF	(Free for manufacturer) Dynamic program invocation dictionary
F000 FFFF	(Profiles) Dynamic program invocation dictionary

Object Description of the Zero Objects

This object description is for the indices to which no object is assigned (e.g. optional objects that are not supported).

Mapping the Device Function onto Communication (see Table 19)

Table 19: Object description: ’zero object’

Object attribute	Value hex	Meaning
Index	xxxx	Zero object
Object code	00	Zero object

Table 20 contains the list of all parameters that can be accessed via communication.

Table 20: List of all parameters that can be accessed via communication

Index	Type	Object	Name	m/o
6000	PDB structure	Record	PI data description	o
6001	PDB structure	Record	PO data description	o
6002	Boolean	Var	PO data enable	o
6003	Unsigned16	Var	PD monitoring time	o
6004	Integer16	Var	PD monitoring selection code	o
6005	Unsigned16	Var	K monitoring time	o
6006	Integer16	Var	K monitoring selection code	o
6007	Integer16	Var	Connection abort selection code	o
6008	Unsigned16	Var	ID number	o
6009	Unsigned8	Var	Parameter record ID	o
600A	Unsigned32	Var	Serial number	o
600B	Date	Var	Calibration date	o
600C	Visible string	Var	Device description	o
600D	Visible string	Var	Model description	o
600E	Date	Var	Parameterization date	o
603E	Octet string	Var	Warning code	o
603F	Octet string	Var	Malfunction code	m
6040	Octet string	Var	Control word	m
6041	Octet string	Var	Status word	m

continued on next page

Table 20: List of all parameters that can be accessed via communication

Index	Type	Object	Name	m/o
6050	Integer16	Var	Setpoint value-1	m
6051	Integer8	Var	Value type-1	
6052	Integer8	Var	Variable index	o
6053	Integer8	Var	Unit index	o
6054			Reserved	
6055	Integer16	Array	Resolution	o
6056			Reserved	
6057			Reserved	
6058	Integer16	Var	Setpoint value-2	m
6059	Integer8	Var	Value type-2	
605A	Integer8	Var	Variable index	o
605B	Integer8	Var	Unit index	o
605C			Reserved	
605D	Integer16	Array	Resolution	o
605E			Reserved	
605F			Reserved	
6060	Integer16	Var	Setpoint value-3	m
6061	Interger8	Var	Value type-3	o
6062	Interger8	Var	Variable index	o
6063	Integer8	Var	Unit index	o
6064			Reserved	
6065	Integer16	Array	Resolution	o
6066			Reserved	
6067			Reserved	
6068	Integer16	Var	Actual value-1	m
6069	Interger8	Var	Value type	
606A	Integer8	Var	Variable index	o
606B	Integer8	Var	Unit index	o
606C			Reserved	
606D	Integer16	Array	Resolution	o
606E			Reserved	
606F			Reserved	
6070	Integer16	Var	Actual value-2	o

continued on next page

Table 20: List of all parameters that can be accessed via communication (end)

Index	Type	Object	Name	m/o
6071	Integer8	Var	Value type	
6072	Integer8	Var	Actual value variable index-2	o
6073	Integer8	Var	Actual value unit index-2	o
6074			Reserved	
6075	Integer16	Array	Actual value resolution-2	o
6076			Reserved	
6077			Reserved	
6078	Integer16	Var	Actual value-3	o
6079	Interger8	Var	Value type	
607A	Integer8	Var	Variable index	o
607B	Integer8	Var	Unit index	o
607C			Reserved	
607D	Integer16	Array	Resolution	o
607E			Reserved	
607F			Reserved	
6080	Unsigned8	Var	Deviation	o
6081	Unsigned16	Var	Sensor monitoring time	o
6082	Integer16	Array	List of secondary controlled variables	o
6083	Unsigned8	Var	Learn paint color	o
6084	Unsigned8	Var	Learn status	o

Extension of the object dictionaries for multiple controllers (see Table 21):

The order of communication objects is retained. The communication objects for the controller at the 2nd slot start at index 6100, etc.

Table 20: Extension of the object dictionaries for multiple controllers, etc.

Index	Type	Object	Name	m/o
5000		Var	Manufacturer-specific parameter for	o
50--		Var	the controller at slot 0	o
50FF		Var		o
5100		Var	Manufacturer-specific parameter for	o
51--		Var	the controller at slot 1	o
51FF		Var		o
5200		Var	Manufacturer-specific parameter for	o
52--		Var	the controller at slot 2	o
52FF		Var		o
6000		Var	Parameter for slot 0	o
60--		Var		o
60FF		Var		o
6100		Var	Parameter for slot 1	o
61--		Var		o
61FF		Var		o
6200		Var	Parameter for slot 2	o
62--		Var		o
62FF		Var		o

Slot number 1 - C

m = mandatory o = optional

10. Device Relationships

Not defined

11. Operating Phases of the Application

This chapter describes the possible operating phases of the device. The chapter is divided into:

- Start-up/abort
- Operation
- Start-up phase and projecting phase

11.1. Start-up/Abort

Start-up

The start-up operation of a controller begins after power on or reset of the device.

- Configuration of the process input and output data

The assignment of the process input and output data depends on the configuration that has been stored (for the default setting see the function block of the communication function).

- Process data initialization

The process input and output data registers get default zeroes.

Table 23 specifies the communication objects which the device parameterizes during start-up with the corresponding stored values or - if there are no such values - with the substitute values.

Table 21: Communication objects and their parameterization during start-up

Communication object	Value	Substitute value
Process data monitoring time	FFFF	Disabled
Process data monitoring selection code	0	No response
Communication monitoring time	FFFF	Disabled
Communication monitoring selection code	0	No response
Communication abort selection code	0	No response

Abort

The following steps are carried out:

- Process data reset.

When the communication unit and the controller unit are decoupled, the process input data is set to zero in the event of a controller unit failure.

11.2. Operation

The following functions are active in the ’Operation’ phase:

- Device control;
- Controller function;
- Sensor/actuator functions.

11.3. Start-up and Projecting Phases

This chapter lists start-up and projecting steps, which are carried out via the INTERBUS-S interface of the devices.

Not yet defined.

12. Communication Profile

12.1. Layer 1

This chapter specifies all definitions concerning layer 1.

Remote Bus Interface

- D-Sub 9-position (male) to the controller
- D-Sub 9-position (female) to the end of the bus
- 2-wire ring
- Diagnostic LEDs
  - Remote bus control (RC) green
  - Remote bus disable (Rbd) red
  - Bus active (BA) green
  - Transmit (TR) green

Supplying the Device with Voltage

SUPI and PCP processor are supplied together with the same voltage.

The bus interface (SUPI and PCP processor) of the device can be supplied via the bus.

The appliation processor can be supplied via the bus.

Error Messages

A module error is indicated when the bus interface (communication processor) has an error.

If the communication processor is working, no module error is indicated, but all errors are indicated in the status word with the malfunction.

12.2. Layer 2

This chapter specifies all definitions concerning layer 2.

INTERBUS-S Registers

The data register arrangement of an INTERBUS-S station and, therefore, the addressing on the I/O level, is defined in the following.

Configuration of the INTERBUS-S registers:

Single controller (see Figure 20):

Communication channel	:	2 words
Process data channel	:	2 words

Four-fold multiple controller-4:

Communication channel	:	2 words
Process data channel	:	8 words

12-fold multiple controller-12:

Communication channel	:	2 words
Prozess data channel	:	24 words

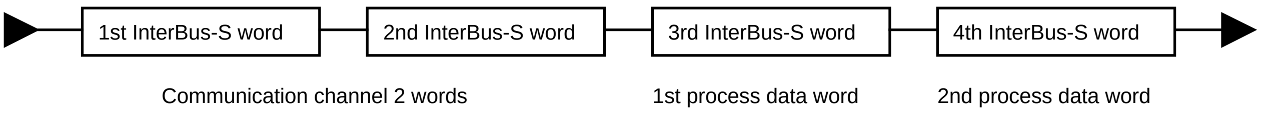

Figure 20: Configuration of the INTERBUS-S registers for a single controller

Addressing the process data (see Figure 21):

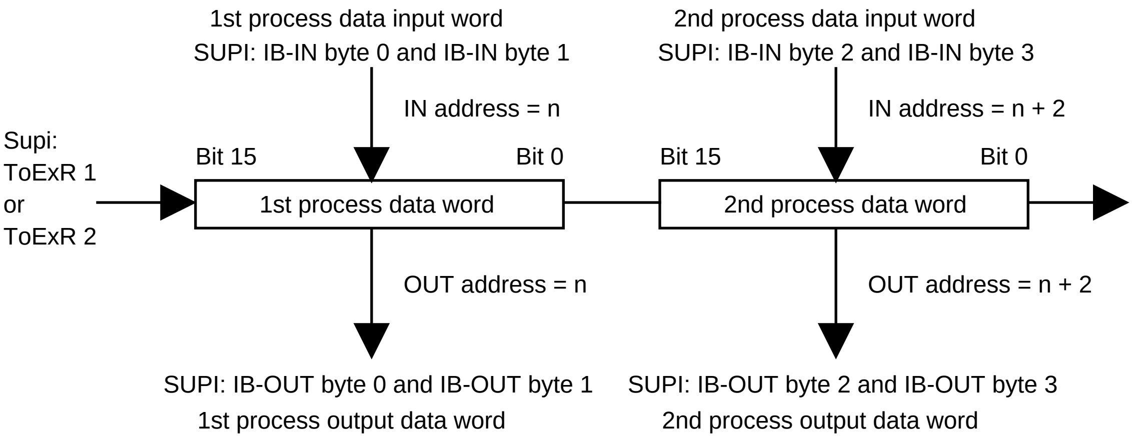
1st process data input word
SUPI: IB-IN byte 0 and IB-IN byte 1
2nd process data input word
SUPI: IB-IN byte 2 and IB-IN byte 3
IN address = n
IN address = n + 2
Supi:
ToExR 1
or
ToExR 2
Bit 15
Bit 0
Bit 15
Bit 0
1st process data word
2nd process data word
OUT address = n
OUT address = n + 2
SUPI: IB-OUT byte 0 and IB-OUT byte 1
1st process output data word
SUPI: IB-OUT byte 2 and IB-OUT byte 3
2nd process output data word

Figure 21: Addressing the process data

Process data direction:

Process input data is transferred from the controller to the bus system.

Process output data is transferred from the bus system to the controller.

Identification of the INTERBUS-S Communication Stations (see Table 16)

Table 16: Identification of the INTERBUS-S Communication Stations

Controller type	Number of words	INTERBUS-S station	ID code
Single controller	4 IN	0000 0100 1111 0000	04E4 hex
4-fold multiple controller	10 IN	0001 0101 1111 0000	15E4 hex
12-fold multiple controller	26 IN / 26 OUT	0001 0001 1111 0000	11E4 hex

12.3. Layer 7

Supported optional PCP services:

- Read;
- Write.

NOTE: The mandatory PCP services are not listed.