INTERBUS-S CLUB

PROFILE

MMI Devices D1

Profile : Operator interfaces and displays

Profile No. : D1

Date : January 25, 1996

Published by : InterBus-S Club e.V.
Geschäftsstelle
Postfach 1108, D-32817 Blomberg
Telephone: *49-5235-34 21 00
Fax: *49-5235-34 12 34

Order No. : D1

Contents **Page**

Preface

"MMI-COM" stands for "**M**an **M**achine **I**nterface **Com**munication".

Within the framework of factory automation, increasingly powerful and flexible systems are needed in the field of industrial sensors and actuators. Operator interfaces and displays 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 operator interfaces and displays independent of the communication medium, an internationally recognized and standardized user interface (DIN 19245 Part 2) was used for communications. This created compatibility with the Manufacturing Message Specification (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.

The profile for operator interfaces and displays is oriented to the user and manufacturer of robot controllers to be operated on the sensor/actuator bus.

For the user, this profile definition is a useful addition to standardized communication and represents a generally valid convention concerning the contents of data and the response of devices. These function definitions standardize a few essential device parameters of a robot controller. Consequently, hardware from different manufacturers exhibits the same response in the communication medium when these standard parameters are used.

The MMI-COM working group was founded in March 1993 by 9 companies on the suggestion of "Verfahrenstechnik Elektrotechnik (VEE)" of Mercedes Benz AG, Untertürkheim (Germany). Its objective is to communicate all working results to all interested parties and to launch MMI-COM products onto the international markets.

This profile consists of a set of standardized functions. When the profile was worked out, the principle was followed that future additions to the specification can be incorporated without effects on the standardized functions. In addition, allowance was made for the use of manufacturer-specific functions.

Authors:

Mr. Tural	Consulting Group Quarré, Stuttgart
Mr. Haugg	Mercedes Benz, Sindelfingen
Mr. Quickert	Mercedes Benz, Sindelfingen
Mr. Maier	Mercedes Benz, Untertürkheim, VEE
Mr. Kunze	Kunze, Sindelfingen
Mr. Bischof	Lauer, Unterensingen
Mr. Krumsiek	Phoenix Contact, Blomberg
Mr. Müller	Phoenix Contact, Blomberg
Mr. Wank	Pilz GmbH, Stuttgart-Ostfildern
Mr. Schuster	SAE-Elektronik, Cologne
Mr. Frees	SMA, Kassel
Mr. Schmitt-Walter	SWAC, Munich
Mr. Kriebitzsch	Wöhrle GmbH, Steinenbronn

(all in Germany)

Introduction

A basis for this profile for operator interfaces and displays is the assumption that, within an automation system, a distinction should be made between control functions on the one hand, and display and operating functions on the other hand.

This approach toward modularization and decentralization of system functions and system components reduces the complexity of automation solutions and, therefore, their susceptibility to errors; it increases their availability, facilitates maintenance and, therefore, contributes to the improvement of their quality and the reduction of costs.

However, considering operator interfaces and displays as separate components in a complete system can only bring about real advantages when certain conditions are met. In this connection, the definition of profiles plays a particular part, as it provides standards between users and suppliers, thus facilitating the interchangeability of software as well as of hardware.

However, profiles and standards only stand a chance of being accepted when they equally provide continuity, progress, and future-proof technologies. The conversion of existing solutions to the standard must be possible easily and efficiently; the MMI-COM standard functions must be efficient and suited to practical use, and there must be sufficient free areas for technical progress and user-specific and manufacturer-specific functions.

To meet all the above requirements specified for operator interfaces and displays, MMI-COM is based on the following premises:

All MMI-COM devices are to be accessed at the same time, i.e. they are to use the same protocol.

An operator interface or display is to relieve the host controller of as much load as possible. Therefore, MMI-COM defines basic functions, which are sufficiently comprehensive and flexible for covering all standard tasks.

The MMI-COM standard functions guarantee the interchangeability of operator interfaces and displays, as it leads to identical results on devices with identical parameterizations (see Figure 1).

With regard to this profile, the InterBus-S master (host controller) and MMI-COM devices as InterBus-S slaves are considered to have the same rights, i.e. each can function as a source as well as a receiver of MMI-COM function calls.

To put these premises into practice, a function-oriented view should be adopted. The question what a device can do is always in the foreground; the question how it does it is only of secondary importance.

A controller (PLC, PC) which sends data to an MMI-COM uses a particular function of this device, without knowing (and without being interested in) how the device implements the function. Such a function may be, for example, "control of an indicator field"; the implementation on the device may be LEDs that light up red and green, or as well circles on a color monitor that are filled in red and green..

We consider the function-oriented model as a real chance for the users and manufacturers for developing portable and reusable software. For the manufacturers, it opens up new ways of reacting quickly and at low costs to the changing requirements of the market.

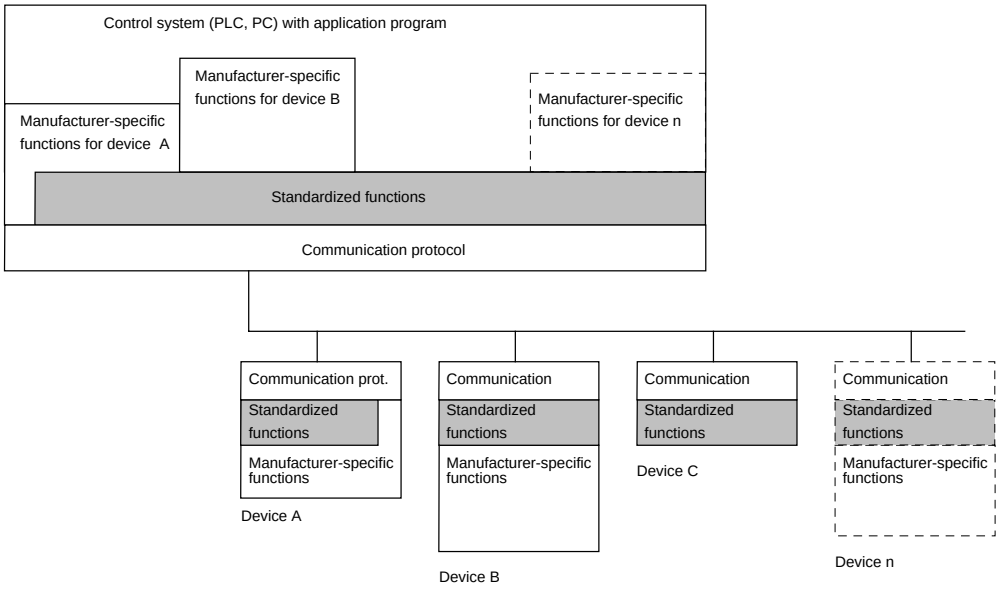


Figure 1: Device function architecture

Economic Advantages for Users and Manufacturers Due to the MMI-COM Profile

Carrying out an automation project involves various costs.

- Quantitative, i.e. measurable costs,

  acquisition costs for hardware and software

  - falling hardware prices due to standardized device functions
  - Reduced programming, maintenance and training expenses due to standardized function components

- Project-phase-oriented qualitative costs (which cannot be directly measured)

  The economic cost advantages due to the use of the MMI-COM profile affect to different degrees all project phases, from the study and concept phases to the phase of use. These advantages, especially in connection with the interface handling, are, for example:

  - Reduced training expenses when devices from different manufacturers are used
  - Reusability of software modules for using the functions defined in the profile
  - Use of devices with standard functions in mixed operation with devices which have additional manufacturer-specific functions required in the system.

1. Scope

The definitions in this profile are oriented towards the use of operator interfaces and displays in a sensor/actuator network on the sensor/actuator level. The sensor/actuator network used is InterBus-S.

2. References

The application interface for communication via the InterBus-S parameter channel conforms to the InterBus-S Club Guideline.

The definitions for data transmission via the process data channel are based on the InterBus-S Club Guideline and to the draft standard DIN 19258.

This profile is based on the definitions of the Sensor/Actuator Profile 12 (InterBus-S Club e.V.)

3. Terms

This chapter defines general terms on which the communication and device model of MMI-COM is based, and special terms in connection with operator interfaces and displays.

3.1. General Terms

The essential concept behind the communication model of MMI-COM is that data is sent to objects or received from objects. The objects are addressed by a number (index). As a rule, it becomes apparent only in connection with the index how the data is to be interpreted, as, for example, the bit pattern 01000001 may stand for the binary-coded number 65, for the BCD-coded number 41, for the ASCII character A, for two keys that are currently being pressed, for a text stored under this ID, or for the request to activate the buzzer.

Telegram

A telegram is a data packet that is transferred in an InterBus-S cycle. Its maximum size is, therefore, the process data channel length. A telegram is not transferred in every bus cycle, but only when this is indicated in the communication protocol. Various communication profiles are available for the structure of telegrams.

Object

An object is a hardware or software item which can carry out particular functions. These functions may be of any complexity, e.g. activating an indicator, drawing a graphic chart, or sending a key code.

The function of an object need not always be visible "outside"; it may, for example, also consist of storing a value. The execution of the object functions may be dependent on other objects (e.g. parameters).

Remote/Local

The terms "remote" and "local" are always used from the point of view of the receiver and refer to the interpretation of the index data. The interpretation of the telegram on the MMI device always results from the local index. Therefore, the MMI device must convert remote indices into local indices.

According to the performance and load of the communication partner, the conversion job can be assigned to the source side (local index), or the receiver side (remote index). In the simplest case, standardized indices are used throughout the complete system.

3.2. Terms Specific to Operator Interfaces and Displays

This profile defines appropriate standard objects for all display and operating functions listed in the following. The precise function results from the description of these standard objects.

Key Field

Each key of a key field is mapped directly to a bit of the telegram. The bit has the value 1 when the key is pressed. When several keys are pressed, the corresponding number of bits is set.

Indicator Field

Each indicator of a lamp field is mapped directly to a bit of the telegram. The indicator is lit when the bit has the value 1.

Key Encoding

Each key of a key field is encoded, and this encoded value is transferred. The transfer of several key codes in one telegram is possible.

Table 1: Key encoding example

Key no.	Meaning of key	Encoded value, e.g. 8 bits (bit pattern)
1.	0	0011 0000
2.	1	0011 0001
	2	0011 0010
	..	00xx 0001

Indicator Encoding

Each indicator of an indicator field is encoded, and this encoded value is transferred. In addition to the indicator code, it may be possible to transfer further attributes.

Text Display

The text to be output is stored in the MMI device. The telegram transfers a text code.

Text Monitor

A text monitor can output ASCII characters. The output position results from the current cursor position. The output is either transparent or with the interpretation of control characters (e.g. VT100 terminal). According to the circumstances, one telegram may transfer several characters.

Variable

A variable may store a numerical value. As standard, it is connected with a variable description, from which further attributes (conversion factors, unit, output position) result. To output a variable on the MMI device, the variable description contains the index of the relevant input or output function.

Output Function

An output function may display a variable on a display device. The representation results from the description of the variable (position) and the specific characteristics of the output function. The type and the scope of output functions are not defined in this profile.

Input Function

An input function can automatically handle the input procedure of a value on the MMI device. The resulting numerical value is only transferred after the completion of the input (as a rule with the enter or return key).

Character

A character is a displayable, readable character. It is transmitted as a bit pattern with the appropriate ASCII code.

Character String

Several keys pressed in succession are combined into a sequence. The individual elements of the sequence are key codes. The sequence is completed by the enter key. The character sequence can be transferred via the PD channel or the PCP channel. The transfer is carried out in a single operation, via the bus. The number of elements is limited.

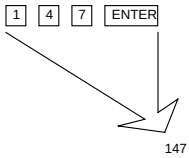


Figure 2: Example of a character string

Table 2: Example of a character string

Character string	Encoded value, e.g. 16 bits (bit pattern)
147	0000 0001 0100 0111
xxx	0000 0001 0011 1100

3.3. Communication-Specific Terms

InterBus-S

The InterBus-S sensor/actuator network is a digital, serial communication system for communication between a controller (e.g. programmable logic controllers, PLCs) and devices for the complete area of industrial sensors/actuators. This includes devices ranging from very simple limit switch and valve to sensors, transducers, actuators and even highly complex high-tech controls such as controlled drives, controllers for automatic wrenching, process controllers, welding controllers, etc.

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 interactions between the application functions, as far as they are executed via the communication system,

and

- by the communication services 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.

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

The communication medium is InterBus-S.

Sensor/Actuator Profile

The Sensor/Actuator Profile is the basis for all devices with a server functionality. 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. All InterBus-S profiles such as the profiles Motion Control, Encoder and Process Controller are based on the Sensor/Actuator Profile.

Process Data Channel

The process data channel is used for a quick transfer of process data. The process data transfers data in an unacknowledged and equidistant form. Process data can be both read and written.

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

- Process output data is data transferred from the controller system to the device. The device reads this data from the process data channel and outputs it to the process.

- Process input data is data transferred from the device to the controller system. The device writes this data to the process data channel, thus transferring it to the controller system.

Parameter Channel

The parameter channel services allow an acknowledged access to all device parameters, i.e. the access to a device parameter is acknowledged by the device.

Index, Subindex

The index is used for addressing a parameter (communication object). The subindex addresses a subparameter (element of a communication object) within a parameter established as a structure.

Mandatory Range

The mandatory range is the value range to which a parameter can be parameterized in any case, provided that it has been implemented.

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.

4. Symbols and Abbreviations

MMI-COM Abbreviations

MF keyboard	Multi-functional keyboard
MMI	Man-machine interface

Network-Specific Abbreviations

PD channel	Process data channel
iPD channel	Indirect process data channel
PA channel	Parameter channel
ID code	Identification code

5. Device Characterization

An operator interface is the link between the controller system (PLC, process control computer) and the user. Operator interfaces may also be active or passive device for the use of the parameter channel on the bus.

The market for operator interfaces requires a broad range of different devices with regard to function and price. Due to the open structure of the Operator Interface and Display Profile, the different elementary functions are covered.

A distinction is made between mandatory, optional and manufacturer-specific device functions and parameters. If the user uses only the mandatory functions or parameters, the operator interfaces and displays are interchangeable.

Device Types

The functions defined in this profile can be implemented in devices of highly different types. The device types are defined mainly by possible hardware and software implementations (see example list).

- Key field/indicator field
- Text monitor
- Plain-text display
- Operator terminal
- Industrial PC

5.1. Structure of Functions Within an MMI Device

This profile defines application functions of operator interfaces. The application functions are divided into display functions, operating functions, global functions, communication functions and control functions. In addition, free areas for the manufacturer-specific functions are defined.

- Display functions
  Operations aiming at the MMI device (e.g. display of texts on the device's display panel)
- Operating functions
  Operations aiming at the control system (e.g. entry of a setpoint by the operator)
- Global functions
  Combination of different functions and directions of operations (e.g. a setpoint entered by the operator is enabled by the controller system).

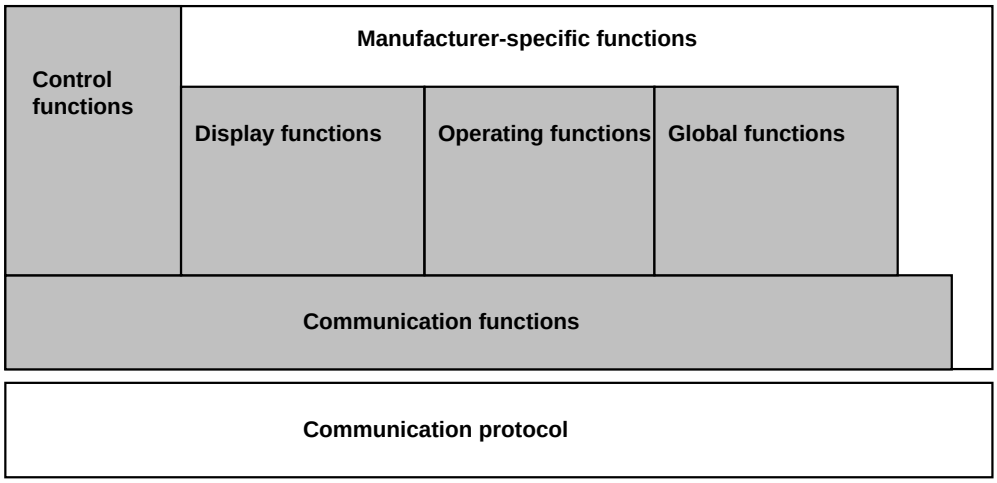


Figure 2: Structure of functions of an operator interface or display

Documentation of the Display Functions, Operating and Global Functions

The device manufacturer must specify on the device data sheet and type label the function group that is supported by the device.

Function group	Functions	Control word/ status word	Via PD channel	Via PA channel	Function group ID (hex) (S/A profile)
A	**Display functions**				
A1	Bit-based indicator functions	o	m		1
A2	Encoded indicator functions	m	m	o	2
A3	Text monitor (ASCII characters)	m	m	o	3
A4	Encoded text display	m	m	o	4
A5	Variable display	m	m	o	5
B	**Operating functions**				
B1	Bit-based key functions	o	m		81
B2	Encoded key functions	m	m	o	82
B3	Variable input	m	m	o	83
G	**Global functions**				
G1	Variable request	m	m	o	FF

m = mandatory, o = optional

The table defines which functions are implemented within the function groups as mandatory and optional functions on the respective communication channel. The control/status word column defines whether the control/status word is mapped to the PD channel. The various function groups are encoded with the function group ID and can be read out via the parameter channel (see the Sensor/Actuator Profile, Chapter "Function Group Description").

Data on the Device Type Label

Example for a text display:
A text display with the functions of ASCII character output, encoded text output and variable request with variable display must provide the following data on the type label.

Example for a typical display device

ID code:	181
iPD channel length	6 bytes
PD channel length	8 bytes
Control and status word	yes
Byte swap	no
Function groups A	3, 4, 5
Function groups B	-
Function groups G	1

Example for a very simple key input field

ID code:	181
iPD channel length	6 bytes
PD channel length	8 bytes
Control and status word	no
Byte swap	yes
Function groups A	
Function groups B	1
Function groups G	

Note: See Chapter "Direct Process Data" for the byte swap

5.2. Communication Functions

The communication functions define how the data transport via the process data channel and the parameter channel is carried out. The communication functions define a device behavior by mapping events to communication functions. The trigger events may be, e.g., the operation of a key, the execution of a macro, or device operations. The trigger events are defined by the device configurations or can be configured. The device functions which can be mapped to the process data channel can also be mapped to the parameter channel. The communication functions provide three types of data transmission.

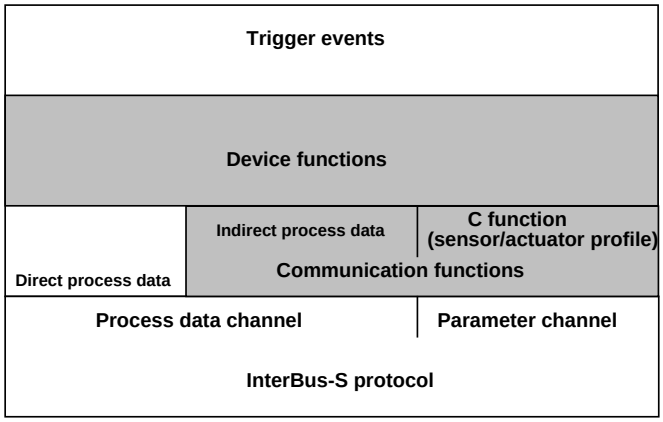


Figure 4: Communication functions

- "Direct process data"
  Unacknowledged process data transmission on the process data channel
  Static mapping of the data onto the process data channel
- "Indirect process data"
  Acknowledged data transmission on the process data channel
  Dynamic mapping of the data onto the process data channel
  The current assignment of the process data channel is defined and indicated via the control status word and an index.
  The data transfer takes place with the Send Data function.
  (See Chapter "Send Data")

- Parameter
  Acknowledged parameter transfer via the parameter channel
  The data transfer takes place with the write or read function
  (see Sensor/Actuator Profile)

5.2.1. Assignment of the Process Data Channel

The division of the process data channel is so that the indirect process data is mapped as the first field, and the direct process is mapped optionally as the second field.

The "indirect process data" field has during operation a fixed length of at least 6 bytes.

Indirect process data		**Direct process data**
Control/status word	**Data field**	

Figure 5: Assignment of the process data channel

Note:
The application on the control system has to ensure the data consistency of the entire process data channel or at least the indirect PD channel. If necessary a controller board (IBS master) can ensure the consistency of the entire process data channel of an MMI-COM device.

5.2.1.1. Indirect Process Data

The area for indirect process data is composed of the control/status word with an integrated PD index and a data field. The current structure of the data field is defined by the PD index in the control/status word. The length of the indirect PD channel is fixed and is specified in the device description. It contains the following fields.

Control/status word	**Data field**
Control/status bits, PD index	

The indirect process data is structured as follows:

- Standard indirect data transfer
- Indicator field (data direction: from the controller system to the MMI device)
- Key field (Data direction: from the MMI device to the controller system)
- Manufacturer-specific data transfer (for manufacturer-specific functions implemented in addition to the mandatory functions)

Control/Status Word

Some bits in the control/status word define the structure of the indirect process data channel (Refer to the "Control/Status Word" Chapter for the assignment of the bits).

PD-Index (in the Control/Status Word)

The PD index defines the meaning of the indirect process data's data field. If the standard bit (S bit) in the control/status word has been set, the values of the PD index have a fixed meaning. If the standard bit has not been set, the meaning is manufacturer-specific

Data Field

In the data field, the useful data defined by the PD index is transmitted. The data field contains, dependent on the PD index,

a **length specification** of the data contained in the following bytes
or
a **subindex** for further addressing of the data

Table 3: Possible Assignment of the Process Data Channel

Control/status word Bytes 1 and 2	**Data field byte 3**	**Data field byte 4**	**Data field byte 5**	**Data field Byte 6**	**Data field byte 7 (optional)**	**Data field byte 8 (optional)**
(S bit =1, I bit =0)	Key field / indicator field					
(S bit =1, I bit =1), PD index	Standard data					
(S bit =0, I bit =1), index	Manufacturer-specific					
(S bit =0, I bit =0)	Manufacturer-specific					

5.2.1.2. Direct Process Data

The "direct process data" is mapped to the process data channel following the "indirect process data". The position is constant for the operating phase of the device, i.e. it cannot be dynamically changed. The direct process data is used for the transfer of bit-based indicator fields and bit-based key fields. The device may be equipped with a switch for switching the assignment of the input and output data to the process data. In the normal position, "Standard", the mapping is carried out as described in the profile. When the switch is in the "byte swap" position (see device type label: Byte Swap) bytes 1 and 2, 3 and 4, 5 and 6, etc. of the direct process data channel are swapped.

5.2.2. Parameter Channel

The transfer of the parameters via the parameter channel is implemented by means of the communication functions defined in the sensor/actuator profile. The addressing of the parameters via the parameter channel takes place by means of an index.

5.2.3. Correlation Between PD Index and Parameter Channel Index

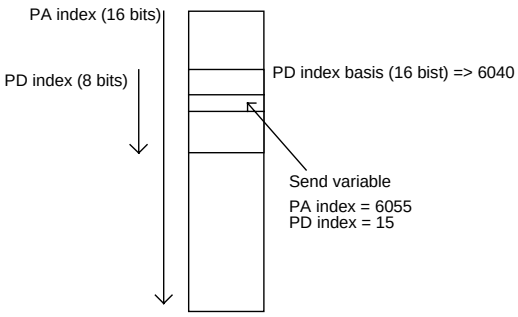
The PD index is used for addressing device parameters via the process data channel. The PA index is used for addressing the device parameters via the parameter channel.

The device parameters are addressed via an unambiguous 16-bit index (PA index). The addressing of these parameters via the indirect process data channel is carried out with a PD index. This PD index has a length of 8 bits and, added to a basis, it amounts to the value of the PA index. This basic value is the PD index basis.

Die PD index basis is a device parameter with the default value 603F.

PA index = PD index + PD index basis

Example:



5.3. Device Data

An analysis of the parameters for an operator interface or display leads to the following data classes:

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

Data to Be Transferred Cyclically

This data class contains control and status signals that have to be transferred very fast (few ms) and cyclically. These signals are transferred via the process data channel.

Data to Be Transferred Acyclically

This data class contains e.g. data blocks which need to be transferred relatively rarely (at intervals of several seconds). These parameters are transferred via the parameter channel without influencing the process data transfer. These parameters can also be transferred via the process data channel; however, depending on the device configuration it may be the case that this channel cannot be used for process data transfer.

Setting Parameters

This data class contains preset device-specific initialization data which can be changed as required and is stored in non-volatile form. These parameters are transferred on the parameter channel.

Information Parameters

This data class contains data that is read out only for information (e.g. on startup). These parameters are transferred via the parameter channel.

6. Application and Device Characteristics

This chapter describes the complete application from the point of view of the communication layer.

6.1. Device Control

The device control is influenced by the control word, by internal signals and by malfunctions. Device control affects the operating and control functions. The status word is generated from the device state and internal signals, and can be read out via the bus.

6.1.1. Control Word and Status Word

The control word and the status word have the same structure and the same function for a communication direction. The control word is the first word of the data from the master to the slave, and the status word the first word of the data from the slave to the master.

The control and status word bits and the internal signals result from logical operations involving the device control instructions affecting the state machine of the respective device. This enables functions and determines device states.

The control word and the status word are composed of the following 16 bits:

Byte	Bit	Name	mandatory
2	0	PD index (bit 0)	o
	1	PD index (bit 1)	o
	2	PD index (bit 2)	o
	3	PD index (bit 3)	o
	4	PD index (bit 4)	o
	5	PD index (bit 5)	o
	6	PD index (bit 6)	o
	7	PD index (bit 7)	o
1	0	Reserved	m
	1	Reserved	m
	2	Index	m
	3	Standard	m
	4	Handshake sending	m
	5	Handshake receiving	m
	6	Malfunction	m
	7	Online	m

The following explanations refer equally to the two communication directions. "Device" means, therefore, according to the direction, either the InterBus-S master or the MMI device.

Online (Bit 7)

This bit indicates that the device is ready to receive. No telegrams may be sent to a device that does not report online. Independent of that, such a device may send data itself. Online and malfunction do not rule out each other.

Malfunction (Bit 6)

This bit indicates a malfunction on the device. If the device is online, a malfunction code can be inquired.

Handshake Receiving (Bit 5)

This bit is used to acknowledge the reception of a telegram. This bit is inverted whenever the remote station indicates an new telegram by inverting its handshake receiving bit (bit 4) and this new telegram has been read out by the receiving station.

This handshake signal only indicates on the protocol level that the transmission channel involved is free again, not that the application has already processed the telegram (e.g. receive buffer). Therefore, the application may still classify the telegram as faulty at a later moment.

Handshake Sending (Bit 4)

This bit signals a new telegram. This bit may be inverted only if it matches the Handshake Receiving bit (bit 5) of the remote station and the remote station is online (bit 7). The transmission channel then remains enabled until the remote station causes its Handshake Receiving bit to match the new value of its own handshake sending bit.

Standard Bit (Bit 3)

This bit defines the assignment of the data fields for process data transfer. In this case the meaning of the PD index is standardized. If this bit is not set, the meaning of the subsequent field is defined in a manufacturer-specific way. If, while the standard bit is set, the index bit is not set, the subsequent field is a bit-by-bit assignment to a key or indicator field.

Index Bit (Bit 2)

This bit defines the existence of an index field in byte 2 of the indirect process data channel's data field. If the standard bit is set, the PD index is transferred in the index field, byte 2. If the standard bit is not set, this indicates a manufacturer-specific index (see the following table)

Standard	Index	Meaning
0	0	Manufacturer-specific data
0	1	Manufacturer-specific object
1	0	Key field / indicator field function
1	1	Standard object

Mapping of the Device Function onto Communication

Mapping onto the PD Channel

Control/ status word Byte 1 Byte 2 PD index	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**

b7 1st byte b0 b7 2nd byte b0

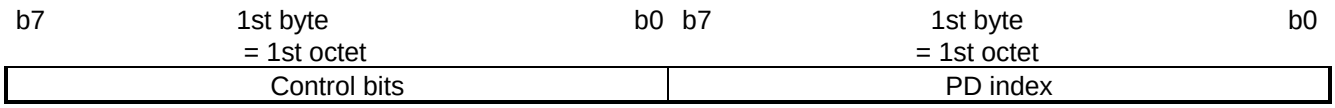
Control/status bits	PD index

Mapping onto the Parameter Channel

Object description: ’Control word’

Object attribute	Value	Meaning
Index	6040	Control word
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	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

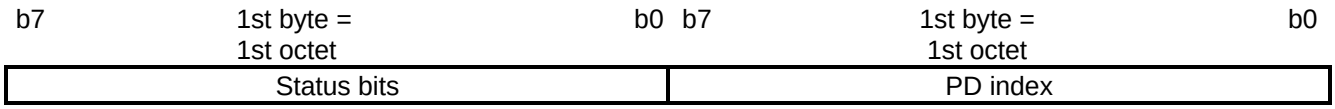
Mapping of the Control Word onto the Octet String:



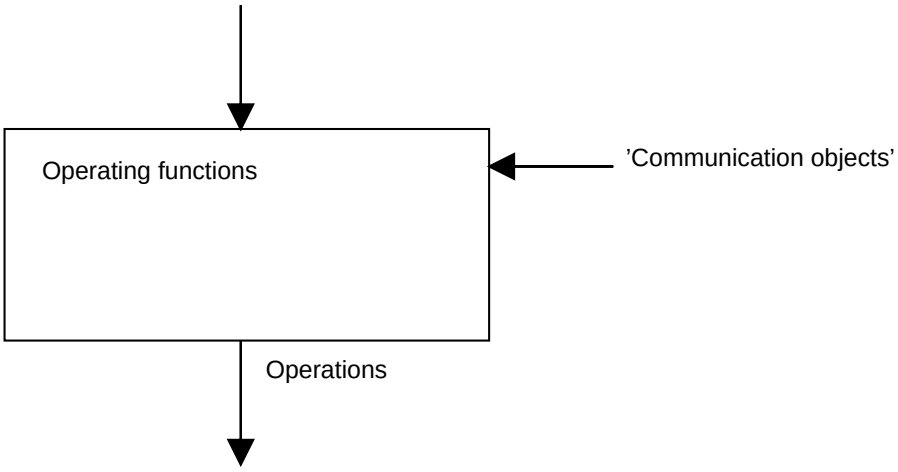
Object description: ’Status word’

Object attribute	Value	Meaning
Index	6041	Status word
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

Mapping of the Status Word onto the Octet String:



6.2. Operating Functions



The operating functions consist of the following subfunctions:

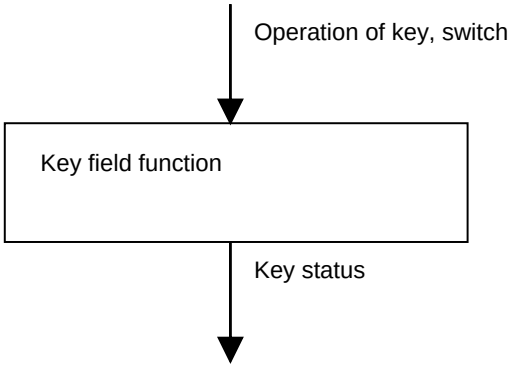
- Key field function
- Key field encoding function
- Variable input function
- PD index basis functions

6.2.1. Key Field Function

This function is implemented as a mandatory function in a class B1 device and as an optional function in all other classes. A key field contains one or more mechanical elements by means of which the operator transfers a state transition to the controller system. This function is mapped directly to the process data channel or to the Send Data communication function.

Possible mechanical elements:

	Parameter value = 0	**Parameter value = 1**
Key	not pressed	pressed
Switch	off	on



PD Index

This parameter contains the value

- Send key status
- Key group

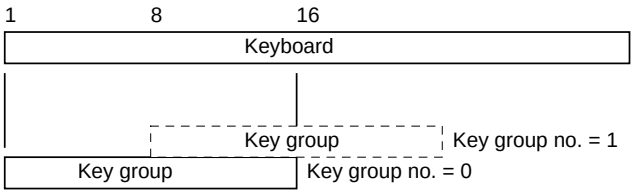
if the function is mapped onto the Send Data communication function.

Key Status

This parameter contains the status of all keys. The parameter has a length of n * 8 bits. This parameter is mapped onto the process input data.

Key Group (n Bytes)

This parameter contains the status of the operated keys in the key group. The parameter has a length of n * 8 bits. This parameter is mapped onto the process input data.

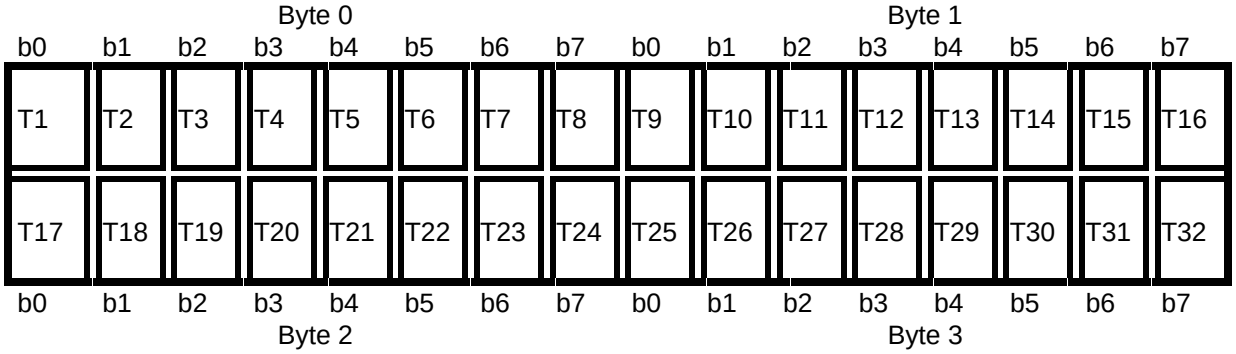


Key Group No.

This parameter defines from which key onwards the 'key group' parameter is mapped onto the key field.

Proposal for the assignment of the keys to the bits in the process data channel

Example: Key field 2 x 16 keys



Mapping of the Device Function onto Communication

Mapping onto the PD Channel

Direct Mapping onto the PD channel

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7 (optional)**	**Byte 8 (optional)**
Key status	Key status	Key status	Key status	Key status	Key status	...	...

Indirect Mapping onto the PD channel

Control/ status word Byte 1	**Byte 2 PD index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
S bit = 1, I bit = 1	10: Send key status	Key status	Key status	Key status	Key status	Key status	Key status
S bit = 1, I bit = 1	11: Key group	Number of keys (in bytes)	Key group no.	Key group		...	...

Mapping onto the Parameter Channel

Object Description: 'Key status'

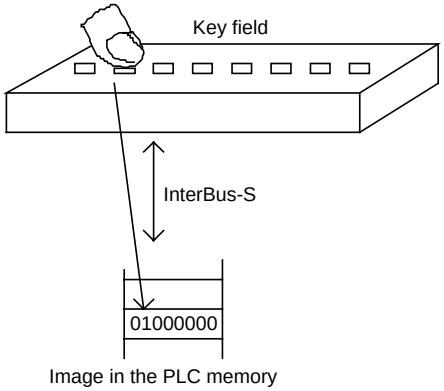
Object attribute	Value	Meaning
Index	6050	Key status
Variable name	-	Key status
Object code	0B	String variable
Data type index	0A	Octet-String n
Length	0n	n bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object Description: 'Key group'

Object attribute	Value	Meaning
Index	6051	Key group
Variable name	-	
Object code	08	Array
Number of elements	256	256 key groups
Data type index	10	Octet string
Length	0n	n 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

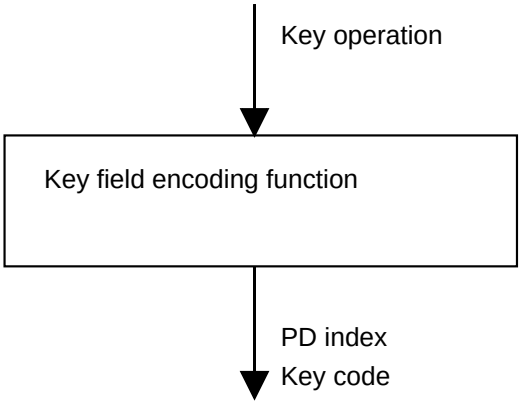
Application Example

The operation of a key is to be transferred as bit information to the process data image (memory) of the PLC.



6.2.2. Key Field Encoding Function

This function is contained as a mandatory function in a class B2 device. A key field contains one or more mechanical elements, by means of which the operator can transfer a change of information to the controller system. This function can be mapped onto the Send Data communication function or to the parameter channel.



PD Index

The PD index defines the data field assignment.

- Send 1-byte key code
- Send variable-length keycode

Key Code

This parameter contains the code of a state change of a key (e.g. pressing or releasing). The parameter length is 1 byte.

Key Code-V

This parameter contains the code of a state change of a key (e.g. pressing or releasing). The 'length' field defines the parameter length.

Length

This parameter defines the length of the key This parameter defines the length of the key code-V parameter.

Mapping the Device Function onto Communication

Mapping onto the PD Channel

Indirect process data

Control/ status word Byte 1	**Byte 2 PD index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	12: key code (1 byte)	Length = 1	Key code				
	13: key code (variable)	Length		Key code-V		...	...

Mapping onto the Parameter Channel

Object description: ' Keycode'

Object attribute	Value	Meaning
Index	6052	Key code (1 byte)
Variable name	-	Key code
Object code	07	Simple variable
Data type index		Octet string
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: ’Key code-V’

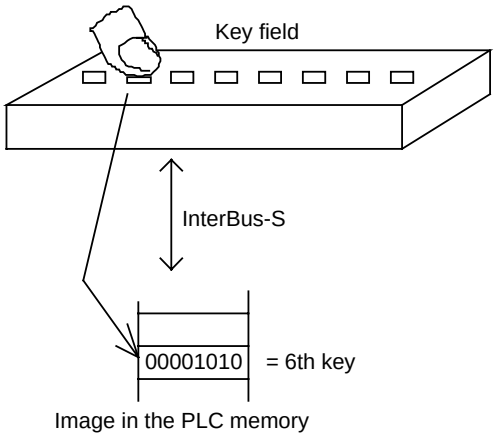
Object attribute	Value	Meaning
Index	6053	Key code-V
Variable name	-	Key code-V
Object code	07	Simple variable
Data type index		Octet string
Length	0n	n Byte
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Key Code-V Mapping onto the Octet String:

b7 1st byte = 1st octet b0	b7 1st byte = 1st octet b0
Key code (high byte)	Key code (low byte)

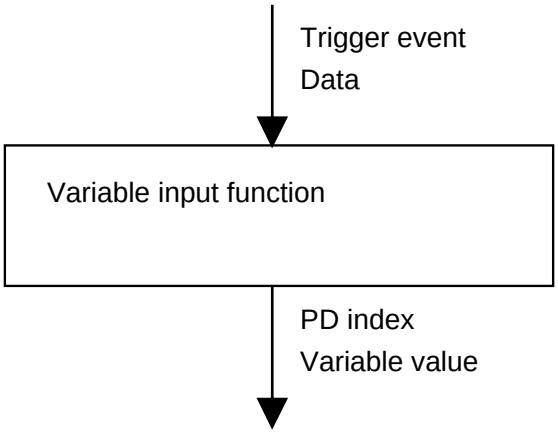
Application Example

The operation of a key is to be transferred as an encoded number to the process data image (memory) of the PLC.



6.2.3. Variable-Input Functions

The input of a variable is a trigger event, which activates a data transfer of a variable to the controller system. This operating function is mapped onto the Send Data communication function. The variable is selected with the variable number; the value of the variable is mapped onto the data field. The encoding of the variable's value with the various data types is defined in the Sensor/Actuator Profile.



PD Index

The PD index defines the type of variable to be transferred. The following PD indices have been defined:

- Send variable (1 byte)
- Send variable (2 bytes)
- Send variable (4 bytes)
- Send floating point (IEEE format: see sensor/actuator profile)
- Send bytes
- Send data (BCD)
- Send time (BCD)
- Send date (binary)
- Send time (binary)
- Send memory area

Note:
A variable of the floating point data type is identified in a special way. All other variables are only distinguished according to their length (1,2,4 bytes). All other specific data types can be transmitted with "Send bytes".

Variable Value

This parameter contains the contents of a variable.

Length

This parameter defines the length of the variable value parameter. If the length exceeds the length of the data field available, the actually transferred variable is split into several transfers. The transfer is completed when the value entered in the length parameter is identical with the length of the data field.

Variable No.

This parameter is used to address the variable. This parameter has a length of 16 bit or 8 bits, depending on whether a length has to be specified for the variable value. If more than 256 variables are to be addressed, this is possible only for a 16-bit variable or via the process data channel.

Current Date

A device sends its current date with time. The transfer of the date is always followed the corresponding time.

Current Time

A device sends it current time.

Send Memory Area

This function writes to a memory area in the destination system. The 'destination address' parameter defines the destination address where the data is to be written. The length of the data field is 16 bits.

Mapping of the Device Function onto the Communication

Mapping onto the PD Channel

Indirect process data

Control/ status word Byte 1	**Byte 2 PD-Index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	14: Send 8 bits	b15 b0 Variable no.		b7 b0 Var. value			
	15: Send 16 bits	b15 b0 Variable no.		b15 b0 Variable value			
	16: Send 32 bits	b15 b0 Variable no.		b31 b0 Variable value			
	17: Send floating point	b15 b0 Variable no.		1st byte	2nd byte	3rd byte	4th byte
	18: Send bytes	Length	Variable no.	1st byte	2nd byte	...	...
	19: Send date BCD-coded	Year	Month	Day of the month	Day of the week		
	1A: Send time BCD-coded	Hour	Minute	Second	1/100 second		
	1B: Send date binary coded	Year (0...99)	Month (1...12)	Day of the month (1...31)	Day of the week (1...7)		
	1C: Send time binary-coded	Hour (0...23)	Minute (0...59)	Second (0...59)	1/100 second (0...99)		
	1D: Send mem. area	Destination address		Data			

Mapping onto the Parameter Channel

Object description: ’1 byte variable’

Subindex (Var.-No.)	**Octet 1**
1	Variable 1
2	Variable 2
3	Variable 3
...	...
255	Variable 255

Object attribute	Value	Meaning
Index	6054	1 byte variable
Variable name	-	1 byte variable
Object code	08	Array
Number of Elements	n	n elements
Data type index	3	Octet string
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: ’2 byte variable’

Subindex (Var. no.)	**Octet 1 (high octet)**	**Octet 2 (low octet)**
1	Variable 1	
2	Variable 2	
3	Variable 3	
...	...	
255	Variable 255	

Object attribute	Value	Meaning
Index	6055	2 byte variable
Variable name	-	2 byte variable
Object code	08	Array
Number of elements	n	n elements
Data type index	3	Octet string
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: ’4 byte variable’

Subindex (Var. no.)	**Octet 1 (high octet)**	**Octet 2**	**Octet 3**	**Octet 4 (low octet)**
1	Variable 1			
2	Variable 2			
3	Variable 3			
...	...			
255	Variable 255			

Object attribute	Value	Meaning
Index	6056	4 byte variable
Variable name	-	4 byte variable
Object code	08	Array
Number of elements	n	n elements
Data type index	3	Octet string
Length	04	4 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: ’Send floating point’

Object attribute	Value	Meaning
Index	6057	Send floating point
Variable name	-	Send floating point
Object code	08	Array
Number of elements	n	n elements
Data type index	08	Floating point
Length	04	4 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: 'Send bytes'

Subindex (Var.-No.)	Octet 1 (high octet)	Octet 2	Octet 3	...	Octet n (low octet)
1	Variable 1				
2	Variable 2				
3	Variable 3				
...	...				
255	Variable 255				

Object attribute	Value	Meaning
Index	6058	Send bytes
Variable name	-	Send bytes
Object code	08	Array
Number of elements	n	n elements
Data-type index	10	Octet string
Length	0n	n 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: 'Send data BCD and binary'

Octet	Meaning
1	Year
2	Month
3	Day of the week
4	Day
5	Hour
6	Minute
7	Second
8	1/100 second

Object attribute	Value	Meaning
Index	6059	Send current date/time BCD
Variable name	-	Date
Object code	07	Simple variable
Data type index	0A	Octet string
Length	08	8 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 attribute	Value	Meaning
Index	605B	Send current date/time binary
Variable name	-	Send date
Object code	07	Simple variable
Data type index	0A	Octet string
Length	08	8 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: ’Send memory area’

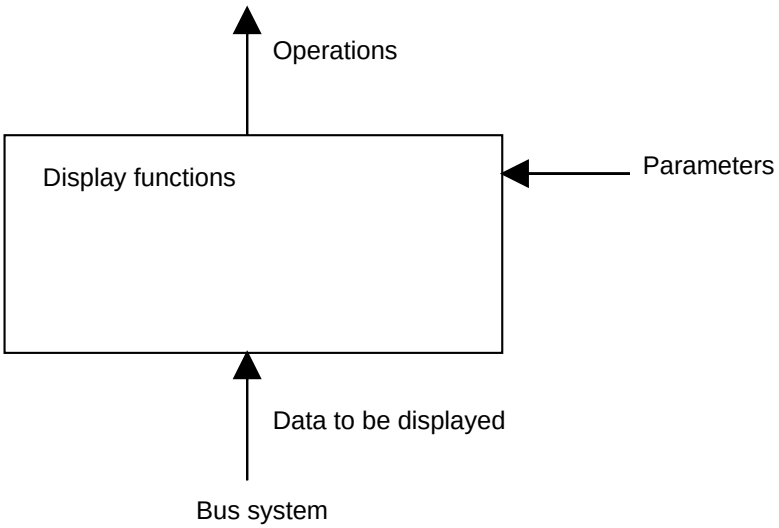
Example: Data = length of 16 bits

Octet 1 (higher octet)	**Octet 2 (lower octet)**	**Octet 3 (higher octet)**	**Octet 4 (lower octet)**
Destination address		Data	

Object attribute	Value	Meaning
Index	605D	Send memory area
Variable name	-	Send date
Object code		Simple variable
Data type index	0A	Octet string
Length	04	4 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

6.3. Display Functions

Display functions are functions that display states, values or texts to the operator of the device.



The display functions consist of the following partial functions:

- Indicator field functions
- Indicator field encoding function
- Text display function
- Text monitor function

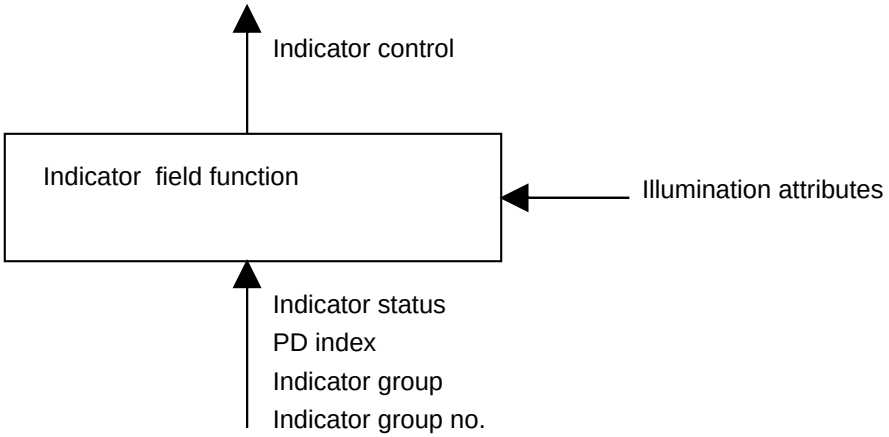
- Combined parameter Mapping onto the PD channel

6.3.1. Indicator Field Function

This function is contained as a mandatory function in a class A1 device, and as an optional function in all other A classes. An indicator field contains one or more elements that indicate states.

Possible elements:

	Parameter value 0	**Parameter value 1**
Indicator	Is not lit	Is lit
Display	Background color with borders	Considerably brighter than the background color



PD Index

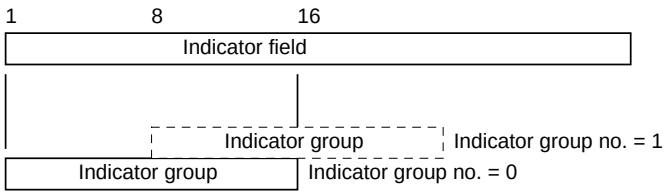
- Indicator status
- Indicator group
- Indicator parameterization

Indicator Status (n Bytes)

This parameter contains the status of all indicators. The parameter has a length of n * 8 bits. This parameter is mapped onto the process input data.

Indicator Group (n Bytes)

This parameter contains the status of the controlled indicators of an indicator group. The parameter has a length of n * 8 bits. This parameter is mapped onto the process output data.

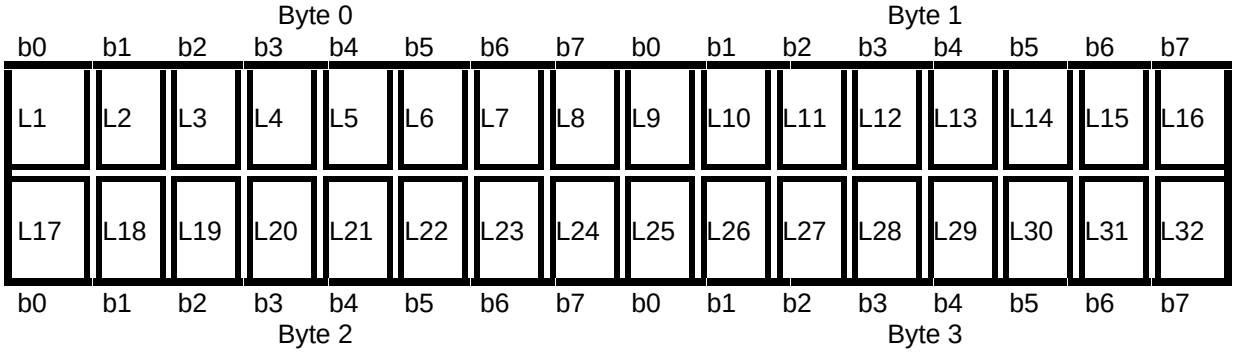


Indicator Group No.

This parameter defines from which indicator onwards the indicator group parameter is mapped onto the indicator field.

Proposal for the assignment of the indicators to the process data channel bits:

Example: Indicator field 2 x 16 indicators



Mapping of the Device Function onto the Communication

Mapping onto the PD Channel

Direct Mapping onto the PD channel

Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7 (optional)**	**Byte 8 (optional)**
Indicator status	Indicator status	Indicator status	Indicator status	Indicator status	Indicator status	...	...

Indirect process data

Control/ status word Byte 1	**Byte 2 PD-Index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	20: Indicator status	Indicator status	Indicator status	Indicator status	Indicator status	Indicator status	Indicator status
	21: Indicator group	No. of indicators (in bytes)	Indicator group no.	Indicator group		...	...
	22: Indicator parameter-ization	Number of indicators (in bytes)	Switching attribute color of illumination	Indicator group		...	...

Mapping onto the Parameter Channel

Object description: ’Indicator status’

Object attribute	Value	Meaning
Index	6060	Indicator status
Variable name	-	Non-existent
Object code	0B	String variable
Data type index	0A	Octet string
Max. length	0n	n 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: ’Indicator group’

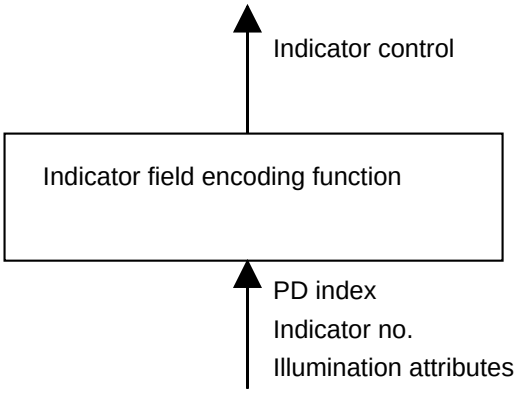
Object attribute	Value	Meaning
Index	6061	Indicator group (bit by bit)
Variable name	-	
Object code	08	Array
Number of Elements	256	256 (Subindex = indicator group)
Data type index	10	Octet string
Length	0n	n bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’Indicator parameterization’

object attribute	Value	Meaning
Index	6062	Indicator parameterization
Variable name	-	
Object code	08	Array
Number of elements	256	256 elements (subindex = indicator no.)
Data type index	10	Octet string
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

6.3.2. Indicator Field Encoding Function

This function is contained as a mandatory function in a class A2 device, and as an optional function in the A-classes 3 and 4. An indicator field contains one or more elements for indicating states. This function is used for controlling the indicators and for defining the illumination attribute.



PD Index

- Indicator no.

Indicator No.

This parameter contains the number of the indicator to be controlled.

Indicator Attribute

b7 b6 b0

Switching attribute	Illumination attribute

Switching attribute

This parameter determines the type of indicator control operation. E.g., the indicator defined by the indicator number is turned on with the bit 7 = 1.

b7 Switching attribute	**Meaning**
1	Switch indicator ON
0	Switch indicator OFF

Illumination attribute (optional)

b5	b4	b3 b0	Meaning
0	0		No flashing
0	1		Flashing interval = slow
1	0		Flashing interval = medium
1	1		Flashing interval = quick
		0	Black on
		1	Red on
		2	Green on
		3	Yellow on
		4	White on
		5	Blue on
		6	Reserved
		7	Reserved
		8	Manufacturer-specific
		.....	Manufacturer-specific
		15	Manufacturer-specific

Mapping of the Device Function onto Communication

Mapping onto the PD Channel

Indirect process data

Control/status word Byte 1	Byte 2 PD-Index	Data field Byte 3	Data field Byte 4	Data field Byte 5	Data field Byte 6	Data field Byte 7 (optional)	Data field Byte 8 (optional)
	23: Indicator no.	Illumination attributes	Indicator no.	Blank	Blank	Blank	Blank

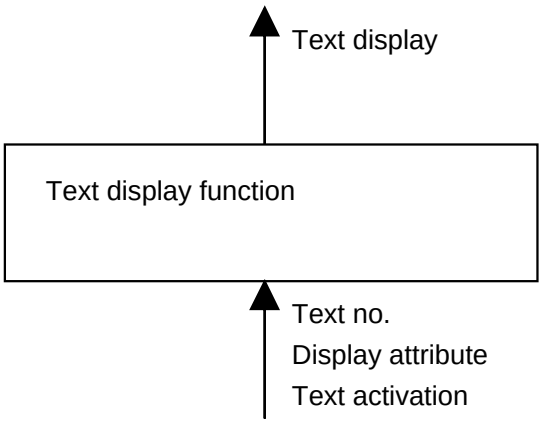
Mapping onto the Parameter Channel

Object description: 'Illumination attribute'

Object attribute	Value	Meaning
Index	6063	Illumination attribute
Variable name	-	
Object code	08	Array
Number of elements	256	256 elements (indicator no.)
Data type index	10	Octet string
Length	01	1 byte (illumination attribute)
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

6.3.3. Text Display Function

The texts displayed with this function are stored in the device. This function is used to select and display a text.



PD Index

- Display text no. (BCD)
- Display text no. (binary)
- Text control (1 of n)

Text No. BCD-coded

This parameter defines which one of the stored texts is to be displayed. The value is BCD-coded.

Text No. Binary-coded

This parameter defines which one of the stored texts is to be displayed. The value is binary-coded.

Display Attribute

The display attribute defines whether the text beside the display is also to be processed.

Display attribute	**Meaning**	mandatory/ optional	
xxxx xxx1	Displaying	m	
xxxx xx1x	Printing	o	
xxxx x1xx	Archive entry	o	e.g. statistics memory

Text Control

This parameter controls a text. The assignment is bit-based; a text is assigned to each bit.

Length

This parameter defines the length of the 'text control' parameter.

Mapping of the Device Function onto Communication

Mapping onto the PD Channel

Indirect process data

Control/status word Byte 1	**Byte 2 PD-Index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	24: Display of text no. (BCD)		Display attribute	Text no. BCD			
	25: Display of text no. (binary)		Display attribute	Text no. binary-coded			
	26: Text control (1 of n)	Length	Display attribute	Text control 8 bits	Text control 8 bits	...	...

Mapping onto the Parameter Channel

Object description: 'Display of text no. (BCD)'

Object attribute	Value	Meaning
Index	6064	Display of text no. (BCD)
Variable name	-	
Object code	07	Simple variable
Data type index	10	Octet string
Length	2	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: 'Display text no.(binary)'

object attribute	Value	Meaning
Index	6065	Display of text no. (binary)
Variable name	-	
Object code	07	Simple variable
Data type index	6	Unsigned16
Length	2	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: ’Text control (1 of n)’

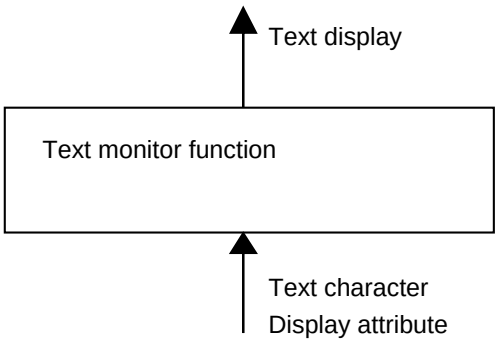
Object attribute	Value	Meaning
Index	6066	Text control (1 of n)
Variable name	-	
Object code	07	Simple variable
Data type index	10	Octet string
Length	n	n bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: ’Display attribute’

Object attribute	Value	Meaning
Index	6067	Display attribute
Variable name	-	
Object code	07	Simple variable
Data type index	10	Octet string
Length	1	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

6.3.4. Text Monitor Function

The transferred text characters are displayed. The position results from the order of transfer.



Text Character

This parameter contains one or more transferred characters.

Length

This parameter defines the number of transferred characters in the data field.

Display Attribute

This parameter defines how the transferred characters are to be interpreted and displayed.

Display attribute	Meaning	mandatory /optional
xxxx xxx1	Displaying	m
xxxx xx1x	Printing	o
xxxx x1xx	Archive entry	o
0001 xxxx	ASCII, the control characters CR, LF and FF are interpreted	m
0010 xxxx	ASCII, control characters are interpreted according to ANSI	o
0011 xxxx	ASCII, control characters are interpreted according to Televideo	o
0100 xxxx	ASCII, control characters are not interpreted (transparent mode)	o

Text Cursor Position

This parameter defines the x,y position of the text cursor.

	x=1	x=2	x=3	x=4	x=5	x=6	x=7
y=1	**H**	**a**	**l**	**l**	**o**		
y=2	**M**	**M**	**I**	**-**	**C**	**O**	**M**
y=3							
y=n							

x position	y position

Mapping of the Device Function to Communication

Mapping onto the PD Channel

Indirect process data

Control/Status word Byte 1	**Byte 2 PD Index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	28: Text monitor	Number of characters	Display attribute	Text attribute	...	...	...
	29: Text cursor	Text cursor x position	Text cursor y position	Blank	Blank		

Mapping onto the Parameter Channel

Object description: 'Text character'

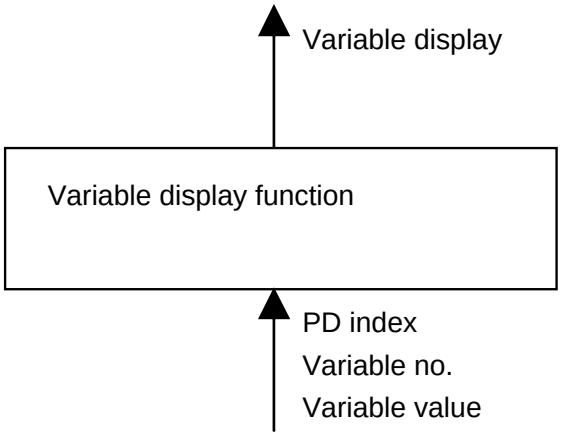
Object attribute	Value	Meaning
Index	6068	Text character
Variable name	-	Non-existent
Object code	0B	String variable
Data type index	0A	Octet string
Max. length	0n	n bytes
Password	00	No password
Access groups	00	No access groups
Access rights	0003	Read-all, write-all
Local address	xxxx	Manufacturer-specific
Extension	-	Non-existent

Object description: 'Text cursor'

Object attribute	Value	Meaning
Index	6069	Text cursor
Variable name	-	Non-existent
Object code	07	Simple variable
Data type index	0A	Octet string 2
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

6.3.5. Variable Display Functions

The variable display is used for displaying or processing variables transferred from the controller system to the display device. This display function is mapped onto the Send Data communication function. The controller system addresses the variable within the display device by the variable number. The value of the variable is transferred in the data field. The encoding of the variable values with the various data types is defined in the Sensor/Actuator Profile.



PD Index

The PD index defines the type of variable to be transferred. The following PD indices have been defined:

- Display variable (1 byte)
- Display variable (2 bytes)
- Display variable (4 bytes)
- Display floating point (see the Sensor/Actuator Profile for the format)
- Display bytes (BCD)
- Display time (BCD)
- Display date (binary)
- Display time (binary)
- Display memory area

Variable Value

This parameter contains the contents of a variable.

Length

This parameter defines the length of the 'variable value' parameter. If the length exceeds the length of the data field available, the actually transferred is distributed among several transfers. The transfer is completed when the value entered for 'length' corresponds to the length of the data field.

Variable No.

This parameter addresses the variable. This parameter has a length of 16 bits or 8 bits, depending on whether a length has to be specified for the variable value. If more than 256 variables are to be addressed, this is only possible for variables with a length of 16 bits, or via the process data channel.

Current Date

The controller system sends its current date with the current time. The transfer of the data is always followed by the time.

Current Time

The control system sends it current time.

Send Memory Area

This function sends a memory area to the device. The 'destination address' parameter defines the destination address to which the data is to be written. The length of the data field is defined in the status word.

Mapping of the Device Function onto Communication

Mapping onto the PD Channel

Indirect process data

Device/status word Byte 1	**Byte 2 PD index**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	2A: Display 8 bits	b15 b0 Variable no.		b7 b0 Var. value			
	2B: Display 16 bits	b15 b0 Variable no.		b15 b0 Variable value			
	2C: Display 32 bits	b15 b0 Variable no.		b31 b0 Variable value			
	2D: Display floating point	b15 b0 Variable no.		1st byte	2nd byte	3rd byte	4th byte
	2E: Display bytes	Length	Variable no.	1st byte	2nd byte	...	...
	2F: Display date BCD-coded	Year	Month	Day of the month	Day of the week		
	30: Display time BCD-coded	Hours	Minute	Second	1/100 second		
	31: Display date binary-coded	Year (0...99)	Month (1...12)	Day of the month (1...31)	Day of the week (1...7)		
	32: Display time binary-coded	Hour (0...23)	Minute (0...59)	Second (0...59)	1/100 second (0...99)		
	33: Display memory area	Destination address		Data			

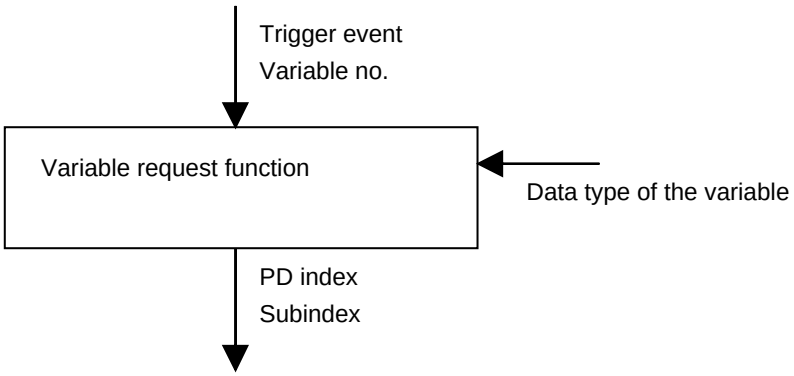
Mapping onto the Parameter Channel

To display variables via the parameter channel, the variables are sent to the device with the write service. See 'send variable' function for a diagram..

6.4. Global Functions

6.4.1. Variable Request Functions

The variable request is a trigger event activating a data transfer of a variable from the controller system to the MMI device in the controller system. This operating function is mapped onto the Send Data communication function. The variable is selected with the variable no.



PD Index

The PD index defines the type of variable to be transferred. The following PD indices have been defined:

- Request variable (1 byte)
- Request variable (2 bytes)
- Request variable (4 bytes)
- Request floating point (see the Sensor/Actuator Profile for the format)
- Request bytes
- Request date (BCD)
- Request time (BCD)
- Request date (binary)
- Request time (binary)
- Request memory area

Variable No.

This parameter is used to address the variable to be sent by the controller system.

Request Date

This function is used to request the transfer of a date from the communication partner. The communication partner responds with the Current Date function, followed by the Current Time function, which goes with the date transferred before.

Request Time

This function is used to request the current time.

Request Memory Area

This function is used to read out a memory area in the destination system. The ’destination address’ parameter contains the address of the memory area to be sent by the controller.

Length

The ’length’ parameter defines the length of the memory area to be sent by the controller.

Mapping of the Device Function onto the Communication

Control/status word Byte 1	Byte 2 PD index	Data field Byte 3	Data field Byte 4	Data field Byte 5	Data field Byte 6	Data field Byte 7 (optional)	Data field Byte 8 (optional)
	40: Request 8 bits	b15 Variable no.	b0				
	41: Request 16 bits	b15 Variable no.	b0				
	42: Request 32 bits	b15 Variable no.	b0				
	43: Request floating point	b15 Variable no.	b0				
	44: Request bytes	Length	Variable no.				
	45: Request date BCD-coded	-	-	-	-	-	-
	46: Request time BCD-coded	-	-	-	-	-	-
	47: Request date binary-coded	-	-	-	-	-	-
	48: Request time binary-coded	-	-	-	-	-	-
	49: Request memory area	Destination address		Length			

Mapping onto the Parameter Channel

To request variables via the parameter channel, the variables are read from the controller system with the read service. See the Send Variable function for a diagram.

Application Example

MMI device configuring:

Variable no.: 50 = Unsigned 16, nominal rotational speed; can be edited by the operator

Variable no.: 67 = Unsigned 16, actual rotational speed, display function

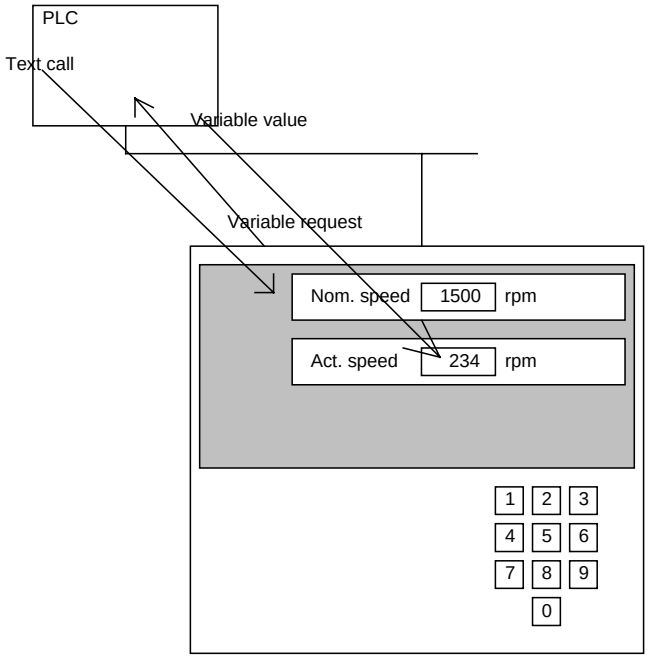
Text no. 100 = "nominal value = [50:xxxx] rpm
actual value = [67:xxxx] rpm"

PLC configuring:

Variable no: 50 = Unsigned 16, specified nominal rotational speed corresponds to e.g. data word 50 in the PLC

Variable no: 67 = Unsigned 16, current actual rotational speed corresponds to e.g. data word 67 in the PLC

Transfer of Variables from the Device to the Controller System

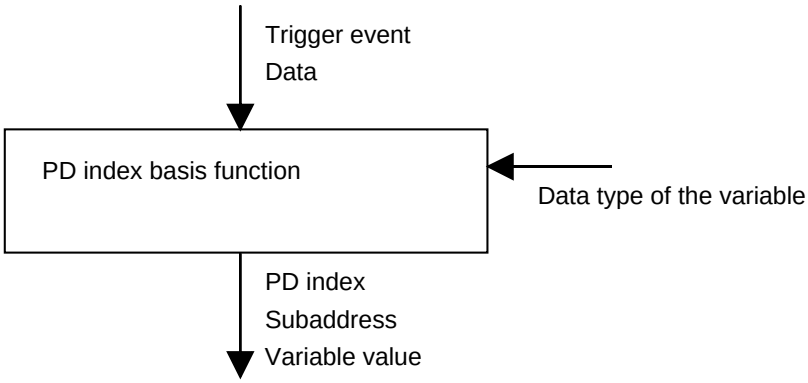


Procedure:

PLC	MMI device
8C 00, 00, 00, 00, 00, 00, 00	8C 00, 00, 00, 00, 00, 00, 00
The PLC requests a text display (text no. 100) Text display function: PD index = text no. (binary) Text no. = 100	
9C 25, 00, 00, 00, 64, 00, 00 (sending)	
	AC 00, 00, 00, 00, 00, 00, 00 (acknowledgment)
9C 25, 00, 00, 00, 64, 00, 00	
	The text is displayed on the text display
	Variable request with variable no. = 50 (old setpoint value) Send Data: PD index = var. request (2 bytes) Variable no. = 50 (hex)
	BC 41, 00, 50, 00, 00, 00, 00 (sending)
9C 25, 00, 00, 00, 64, 00, 00	Waiting
	BC 41, 00, 50, 00, 00, 00, 00
BC 25, 00, 00, 00, 64, 00, 00 (acknowledgment)	
Send variable: Variable no. 50, variable value = 3000 Send data: PD index = send variable (2 bytes) Variable no. = 50(hex) Variable value = 3000	Waiting for variable
AC 15, 00, 50, 0B, B8, 00, 00 (sending)	
Waiting	BC 41, 00, 50, 00, 00, 00, 00
Waiting	BC 41, 00, 50, 00, 00, 00, 00
	9C 41, 00, 50, 00, 00, 00, 00 (acknowledgment)
	Mapping of value 3000 into the text The data of the Send Data function is evaluated and a variable display is performed.
	Variable request with variable no. = 67 (current actual value) Send Data: PD index = var. request (Unsigned 22) Variables no. = 67(hex)
	8C 15, 00, 67, 00, 00, 00, 00 (sending)
8C 15, 00, 50, 0B, B8, 00, 00 (acknowledgment)	
Send variable: Variable no. 67, variable value = 2997 Send Data: PD index = send var. (2 bytes) Variable no. = 67(hex) Variable value = 2997	Waiting for variable
9C 15, 00, 67, 0B, B5, 00, 00 (sending)	
	AC 15, 00, 67, 00, 00, 00, 00 (acknowledgment)
	Mapping of the value2997 to the text The data of the Send Data function is evaluated and a variable display is carried out.
	Operator input : 1500 Use of the variable input function: Send Data: PD index = Send Variable (2 bytes) Variable no. = 50 Variable value = 1500
	BC 15, 00, 50, 05, DC, 00, 00 (sending)
BC 15, 00, 67, 0B, B5, 00, 00 (acknowledged)	
The PLC uses the variable value = 1500 as setpoint specification (variable no.: 50 is the setpoint value)	

6.4.2. PD Index Basis Functions

This function is used to parameterize and read out the PD index basis.



PD Index Basis

This parameter is the offset of the calculation of the PA index.

Mapping of the Device Function onto Communication

Mapping onto the PD Channel

Indirect process data

Control/ status word Byte 1	**PD index Byte 2**	**Data field Byte 3**	**Data field Byte 4**	**Data field Byte 5**	**Data field Byte 6**	**Data field Byte 7 (optional)**	**Data field Byte 8 (optional)**
	4: Read out PD index basis	Blank	Blank	PD index basis			
	5: Set PD index basis	Blank	Blank	PD index basis			

Mapping onto the Parameter Channel

Object description: 'PD index basis'

Object attribute	Value	Meaning
Index	6042	PD index basis
Variable name	-	
Object code	07	Simple variable
Data type index	6	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

6.5. MMI-COM Communication Functions

This chapter contains all functions defines all functions defined for both transfer directions, from the controller system to the MMI device and vice versa.

6.5.1. Send Data

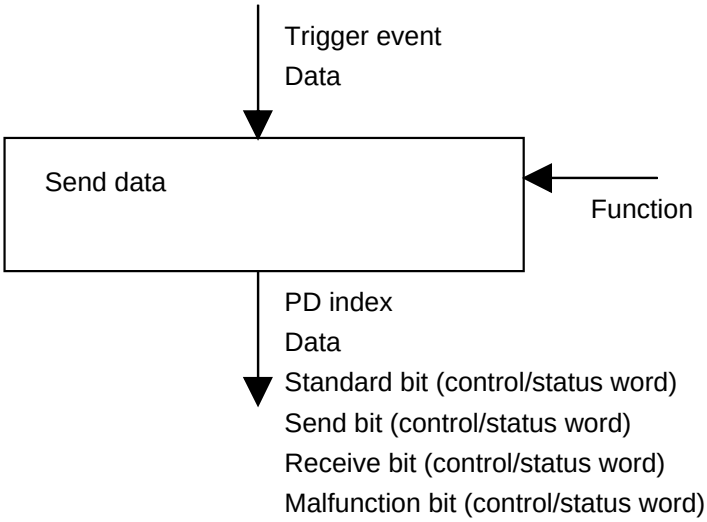
This function is used for data transfer on the process data channel, from the device to the controller system and vice versa.

Applications:

Specification of value, digit sequence, status and control of variables, etc.

Send Data Sequence from the MMI Device to the Controller System:

The device checks whether the receive bit in the control word has the same status as the send bit in the status word. In this case the MMI device can send data. The appropriate data and the PD index are loaded into the process data channel together. At the same time the send bit in the status word can be inverted to indicate sending. The controller system handshaking is monitored. If the controller system does not respond within 1 sec., the send bit is toggled.



PD Index

This parameter defines which data items are to be sent.

Data

This parameter contains the data to be sent.

6.6. Sensor/Actuator Functions

6.6.1. Communication function

See Sensor/Actuator Profile 12, Chapter 'Communication Functions'.

Note:

The Initiate service requests a connection setup. The initiator of the connection establishment must set the 'profile number' service parameter to the value 00D1 hex.

6.6.2. Device Information

See Sensor/Actuator Profile 12, Chapter 'Device Information' .

6.6.3. Malfunction Function

The malfunction function (see Figure 6) manages the 'malfunction code' parameter. As a result of a device 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.

Note:
The malfunction function is not mandatory. However if the device detects a malfunction this malfunction must be allocated to a malfunction cause given in the table and marked with the corresponding code.

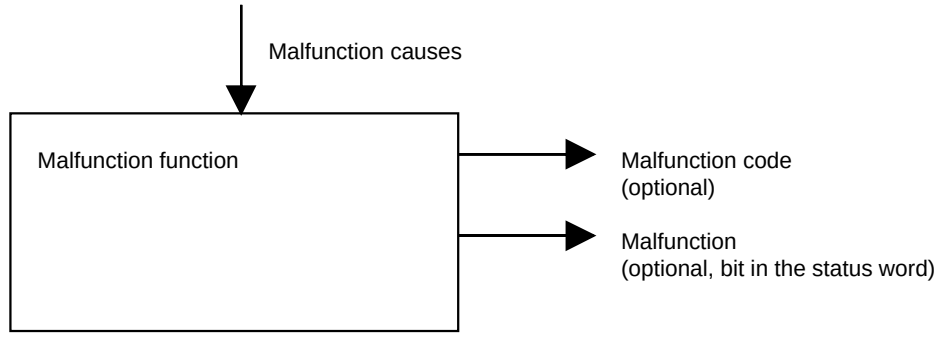


Figure 6: Malfunction function

'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 (see Table 4).

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

When the device is in the malfunction state, the parameter is assigned a value unequal to 0. The parameter is assigned the value 0 when the device is not in the malfunction state.

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 is quit. 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.

Table 4: Malfunction codes and malfunction causes

Code hex	Meaning
0000	**No malfunction**
5000	**Device hardware (only inside the device housing)**
5300	Operating and display unit
6000	**Device software**
6100	Internal software
6200	User software
6210	PD index not available
6211	Variable no. not available
6300	Data set not okay
7000	**Additional modules**
7600	Data memory
8000	**Monitoring**
8100	Communication
8110	Process data monitoring
8120	Host monitoring
8121	iPD channel handshake timeout

Codes that are not listed are reserved.

7. Data Structures

This chapter lists the data structures of all user data.

7.1. PD Index

Table 5: Meaning of the PD-Index

PA index	PD index	Meaning (contents of the data field)	B1	B2	B3	A1	A2	A3	A4	A5	G1
6040	0	Reserved for control word	o	m	m	o	m	m	m	m	m
6041	1	Reserved for status word	o	m	m	o	m	m	m	m	m
6042	2	Read out PD index basis	o	o	o	o	o	o	o	o	o
	3	Set PD index base	o	o	o	o	o	o	o	o	o
	4	Request malfunction code (Sensor/Actuator Profile)	o	o	o	o	o	o	o	o	o
	5	Send malfunction code (Sensor/Actuator Profile)	o	o	o	o	o	o	o	o	o

PA index	PD index	**Operating functions** Meaning (contents of the data field)	B1	B2	B3
6050	10	Send key status	m	o	o
6051	11	Send key group	o	o	o
6052	12	Key code (1 byte)	-	m	o
6053	13	Key code (variable)	-	m	o
6054	14	Send 1 byte variable	-	-	m
6055	15	Send 2 byte variable	-	-	m
6056	16	Send 4 byte variable	-	-	o
6057	17	Send floating point	-	-	o
6058	18	Send bytes	-	-	o
6059	19	Send current date (BCD)	-	-	o
	1A	Send current time (BCD)	-	-	o
605B	1B	Send current date (binary)	-	-	o
	1C	Send current time (binary)	-	-	o
605D	1D	Send memory area	-	-	

PA Index	PD Index	**Display functions** Meaning (contents of the data field)	A1	A2	A3	A4	A5
6060	20	Indicator status	m	o	o	o	o
6061	21	Indicator group	m	o	o	o	o
6062	22	Parameterize indicator (bit by bit), value = illumination attribute	-	m	o	o	o
6063	23	Send indicator no., value = illumination attribute	-	m	o	o	o
6064	24	Display text no. (BCD)	-	-	-	m	o
6065	25	Display text no. (binary)	-	-	-	m	o
6066	26	Text control (1 of n)	-	-	-	m	o
6067	--	Display attribute	-	-	m	m	o
6068	28	Text monitor	-	-	m	-	o
6069	29	Text cursor positioning	-	-	m	m	o
	2A	Display variable 1 byte	-	-	-	-	m
	2B	Display variable 2 bytes	-	-	-	-	m
	2C	Display variable 4 bytes	-	-	-	-	o
	2D	Display floating point	-	-	-	-	o
	2E	Display bytes	-	-	-	-	o
	2F	Display current date (BCD)	-	-	-	-	o
	30	Display current time (BCD)	-	-	-	-	o
	31	Display current date (binary)	-	-	-	-	o
	32	Display current time (binary)	-	-	-	-	o
	33	Display memory area	-	-	-	-	o

PA Index	PD Index	**Global functions** Meaning (contents of the data field)	G1
	40	Request variable 1 byte	m
	41	Request variable 2 bytes	m
	42	Request variable 4 bytes	o
	43	Request floating point	o
	44	Request bytes	o
	45	Request current date (BCD)	o
	46	Request current time (BCD)	o
	47	Request current date (binary)	o
	48	Request current time (binary)	o
	49	Request memory area	o

7.2. Object Dictionary Structure

See Sensor/Actuator Profile 12

8. Operating Phases of the Application

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

- Initialization/abort
- Operation
- Startup phase and configuring phase.

8.1. Initialization/Abort

Startup

The device initialization procedure begins after power-on of the device, or after it has been reset.

The following operations are carried out by the device:

- Configuration of the process input and output data
- Process data initialization

The process input and output data registers get zeros as default.

The device parameterizes the following communication objects with the appropriate stored values or - if not - with the substitute values during startup.

Communication object	Value	Meaning
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
Connection abort selection code	0	no response

Abort

The following operations are carried out:

- Process data reset

If the communication device and the MMI-Com device are decoupled, the process input data is set to zero when the device fails.

8.2. Operation

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

- Device control
- Operating functions and/or display functions
- Sensor/actuator functions

9. Communication Profile

9.1. Layer 1

This chapter contains all definitions concerning Layer 1.

Interfacing with InterBus-S

- Installation remote bus interface CONINVERS connector (IP 65)

or

- Installation remote bus interface 9-position subminiature D connector (IP 20)

(up to 1 A)

or

- Remote bus interface
- 9-position subminiature D connector (male) to the controller
- 9-position subminiature D (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 (only when the parameter channel has been implemented)

9.2. Layer 2

This chapter contains all definitions concerning layer 2.

9.2.1. Configuration of the InterBus-S Registers

The arrangement of an InterBus-S device's data register and, therefore, the addressing at the I/O level, is defined in the following. Example for an InterBus-S device with a 2-byte parameter channel:

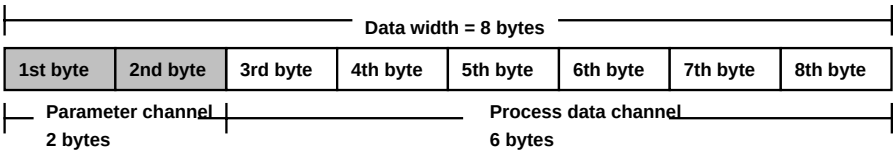


Figure 7: Relations between data width, process data channel and parameter channel

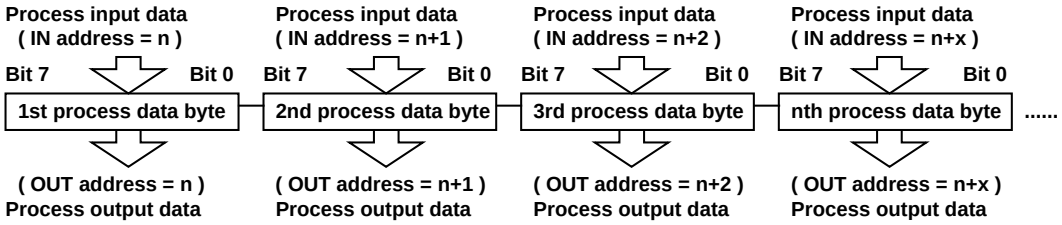


Figure 8: Process data addressing

Process Data Direction:

- Process input data is transmitted from the device to the bus system.
- Process output data is transmitted from the bus system to the device.

9.2.2. Identification of the InterBus-S Devices

The ID code is composed as follows:

b15 b13	b12 b8	b7 b0
Message	**Data width**	**ID code**

Message

This bit in the ID code is used to transfer messages to the controller board.

Table 6: Messages

b 15	b 14	b13	Meaning
1	x	x	Device message
x	1	x	CRC error
x	x	1	Reserved

Device message
This message is generated when the device has detected a malfunction of the periphery. A malfunction in the periphery can be defined in detail in further profiles.

CRC error
This message is generated when transmission errors have been detected (by the protocol chip).

Data Width

The data width indicates how many bits the device uses on the bus. If a device has, for example, 16 input bits and 32 output bits, it occupies 32 bits (4 bytes) in the ring (the higher value is decisive). The length of the parameter channel is defined in the ID code.

Table 7: Data width

Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Data width
0	0	0	1	1	6 bytes
0	0	1	0	0	8 bytes
0	0	1	0	1	10 bytes
0	1	1	1	0	12 bytes
0	1	1	1	1	14 bytes
0	0	1	1	0	16 bytes
0	0	1	1	1	18 bytes
1	0	1	0	1	20 bytes
1	0	1	1	0	24 bytes
1	0	1	1	1	28 bytes
1	0	0	1	0	32 bytes
1	0	0	1	1	48 bytes
1	0	0	0	1	52 bytes
1	0	1	0	0	64 bytes
1	0	0	0	0	reserved
1	1	x	x	x	reserved

x = "don't care"

ID code

Description of the device function		**ID code (dec)**	**ID code (hex)**
Remote bus device, digital			
Profile-compatible digital devices with output addresses	PROFILE DO	13	0D
Profile-compatible digital devices with input addresses	PROFILE DI	14	0E
Profile-compatible digital devices with input and output addresses	PROFILE DIO	47	2F
Remote bus devices with parameter channel			
Profile-compatible devices (2 PCP words)	PROFILE PA chan.	228	E4
Profile-compatible devices (4 PCP words)	PROFILE PA chan.	229	E5
Profile-compatible devices (1 PCP words)	PROFILE PA chan.	231	E7
Local bus devices, digital			
Profile-compatible digital devices with output addresses	PROFILE DO	181	B5
Profile-compatible digital devices with input addresses	PROFILE DI	182	B6
Profile-compatible digital devices with input and output addresses	PROFILE DIO	183	B7
Local bus devices with parameter channel			
Profile-compatible devices (2 PCP words)	PROFILE PA chan.	216	D8
Profile-compatible devices (4 PCP words)	PROFILE PA chan.	217	D9
Profile-compatible devices (1 PCP word)	PROFILE PA chan.	219	DB

9.3. Layer 7

The parameter channel is used in accordance with the Sensor/Actuator Profile 12.