TURCK

Industrial
Automation

USER MANUAL RFID SYSTEM

SET-UP WITH THE PROXY IDENT FUNCTION BLOCK

BL ident

S1579/01

About this manual

Safety Notes!

Before starting the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Earth and short circuit.
- Cover or enclose neighboring units that are live.
- Follow the engineering instructions (AWA) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50 110-1/-2 (VDE 0 105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalization. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference do not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60 364-4-41 (VDE 0 100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.
- Emergency stop devices complying with IEC/EN 60 204-1 must be effective in all operating modes of the automation devices. Unlatching the emergency-stop devices must not cause restart.

- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.
- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.
- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).
- The electrical installation must be carried out in accordance with the relevant regulations (e. g. with regard to cable cross sections, fuses, PE).
- All work relating to transport, installation, commissioning and maintenance must only be carried out by qualified personnel. (IEC 60 364 and HD 384 and national work safety regulations).
- All shrouds and doors must be kept closed during operation.

About this manual

Documentation concept

The first chapter of this manual will provide you with an overview of the TURCK *BLident* system.

The second chapter contains instructions for the commissioning of a *BLident* system using the standard function block „Proxy Ident Function Block“. The commissioning example is conducted on a SIMATIC S7/-300 station (Siemens). The SIMATIC basis software STEP 7 is deployed.

The third chapter contains an excerpt from the „Proxy Ident Function Block“ specification.

Attention

This section needs to be read carefully as the safety considerations when dealing with electrical devices cannot be left to chance.

This manual contains the information required for the commissioning of the TURCK *BLident* system
It has been designed for qualified personnel with the necessary specialist knowledge.

Prescribed use

Warning

The devices described in this manual are only to be used for the applications prescribed here or in the respective technical descriptions and only in connection with certified third party devices and components.

The appropriate transport, storage, installation and assembly as well as careful operation and maintenance are the preconditions for the efficient and safe operation of the devices.

Notes concerning configuration/installation of this product

Warning

The applicable regulations and accident prevention guidelines must be complied with for the respective deployment.

Explanation of the symbols used

Warning

This symbol appears next to all warning notices that indicate a source of danger. This can refer to both injury to persons or damage to the system (hard and software).

For the operator this symbol means: work with extreme caution.

Attention

This symbol appears next to all warning notices that indicate a potential source of danger.

This can refer to both possible injury to persons, damage to the system (hard and software) or the installation.

Note

This symbol appears next to general notices providing important information concerning one or more operating steps.

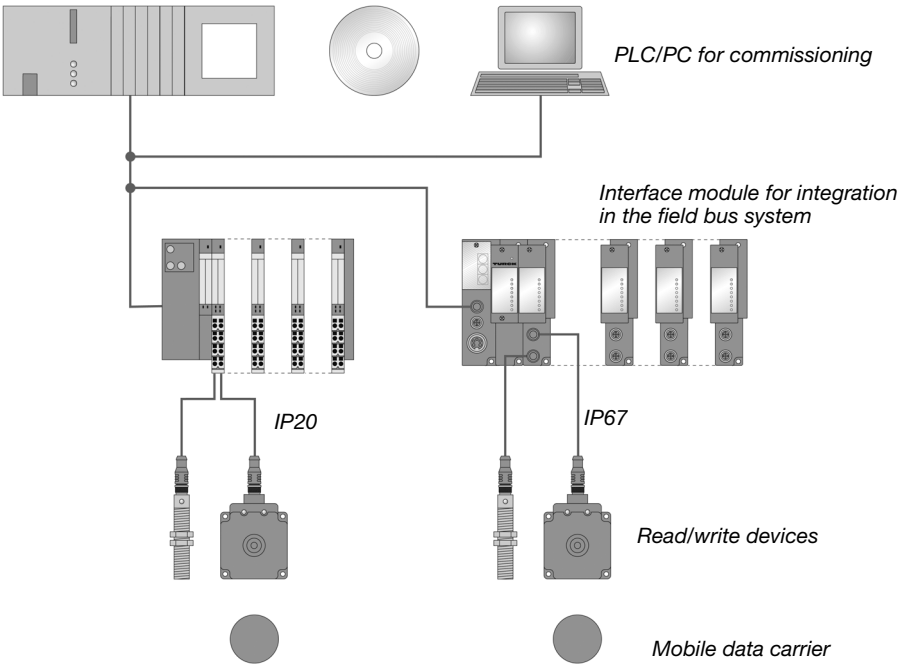
The relevant information may contribute to making the operation easier and assist e.g. in preventing additional work due to an incorrect procedure.

1 The TURCK *BLident* system

Schematic presentation of the identification system *BLident*

The TURCK *BLident* system is composed of a number of levels. Each level offers a range of possibilities. An application tailored to the total system is possible.

Illustration: 1
System overview



Support for *BLident* projects

The following software and documents provide support for configuration, installation and commissioning:

- For simulation and optimisation of an application a "*BLident* configurator" is available under www.turck.com, free of charge.
- D101582 - this manual lists the technical details of the available TURCK data carriers and TURCK read/write devices.

- D101581 - this manual describes the correct operation of *BLident* Interface modules.
- D101606 - this manual contains a description of the software for a so called "Handheld" (programming device) which can be used for mobile data read out.
- D101585 - this manual contains a description of the hardware of a so called "Handheld" (programming device) which can be used for mobile data read out.

Networking with *BLident* systems

BLident systems can be integrated into (existing) bus systems, enabling the networking of a number of *BLident* systems.

The guidelines for the maximum extension of the respective bus system apply.

A PROFIBUS system without repeaters can be composed of e.g. a maximum of 31 stations and a master.

Identification systems with radio frequency technology (RFID)

RFID is the abbreviation for Radio Frequency Identification

An RFID system is composed of a data storage medium, a device for reading and writing the data on the storage medium as well as further devices for transmitting and processing the data.

The transmission of the data from the storage medium to the read/write unit is carried out by means of electromagnetic waves. This form of transmission is conducted magnetically and without visual contact and is thus unaffected by dirt and temperature fluctuations.

The data storage medium can be attached directly to a product. For this reason the term "mobile data storage" i.e. MDS is employed. Further terms for the data storage medium are TAG or transponder. The data can be composed of production and processing data. Of special importance is the data that identifies the product. This is where the term "Identification System" originates.

A whole range of possibilities are opened up by the ability to change the data written on the data storage medium. This enables production/manufacturing processes to be tracked. Logistics/distribution can be optimised.

The "Identification Systems" can be integrated into (existing) field bus automation systems (e.g. PROFIBUS). The connection to the respective field bus system is established using the appropriate interface module.

Standardised software blocks (e.g. the Proxy Ident Function Block for PROFIBUS) allow for simple system integration and commissioning with a range of different control systems.

Performance and areas of application of the *BLident* system

In order to meet the requirements of a range of different application areas, the TURCK *BLident* system provides a variety of combination options for data carriers and read/write units as well as interface modules for connecting to automation systems (e.g. PROFIBUS-DP). Software blocks facilitate easy integration and commissioning.

Below is a list of the performance characteristics of the TURCK *BLident* system:

Degree of protection

All data carriers as well as the corresponding read/write heads have a high degree of mechanical protection (e.g. **IP67**) and thus can be employed under the roughest industrial conditions.

The connection to a field bus system is established using suitable TURCK interface modules. The interface modules are available in the degrees of protection IP20 and IP67. TURCK connection cables with the appropriate degree of protection complete the identification system.

Lifetime

The lifetime is determined by the possible read/write operations carried out on the data carrier.

FRAM data carriers can guarantee an **almost unlimited** number (10^{10}) of read and write operations.

EEPROM memories can guarantee an **unlimited** number of read operations and a limited number of write operations.

External batteries for the data carriers are not required.

Transmission frequency

The TURCK *BLident* system operates at a transmission frequency of **13.56 MHz** between the data carrier and the read/ write devices. This transmission frequency is also employed by the data carriers of so called Smart Labels, offering good performance at an attractive price. Systems operating at this transmission frequency are largely unaffected by electromagnetic interference. The transmission

frequency of 13.56 MHz has established itself as the standard in a large number of RFID applications.

Size

TURCK supplies round data carriers with diameters of 16, 20, 30 and 50 mm.

The read/write units are available in a range of structural shapes, from M18 and M30, CK40, Q80 to S32XL.

Memory

The memory capacity of the data carrier is **64 bytes** with EEPROM memory and **2 kbyte** with a FRAM memory.

FRAM: (Ferroelectric Random Access Memory), non-volatile, longer lifetime due to the higher number of read/write operations (10^{10}-10^{11})

EEPROM: (Electrically erasable programmable read only memory), non-volatile

Read/ write time

The read and write times are identical with FRAM data carriers and are approx. 0.5ms/byte. However a minimum of 8 bytes must be read or written at any one time. The read time with EEPROM data carriers is approx. 2ms/byte and the write time approx. 4ms/byte. Here at least 4 byte blocks must be read, i.e. written at any one time.

Read/write range

The attainable read/ write distances are dependent on the respective combination of data carrier and read/write head and are between 15 and 200 mm. With the *BLident* configurator the application parameters speed, range and data volume can be varied in order to select the optimal combination for the respective application. The configurator is available online under http:// www.turck.com/

Compatibility

All technical data refers to the *BLident*system. i.e. to the combination of *BLident* data carriers, read/write heads and interface modules. Completely different values may apply for data carriers from other manufacturers. They are only to be deployed following authorisation from TURCK.

Areas of application (examples):

The performance characteristics listed in the previous chapter enable the TURCK *BLident* system to be deployed in the following branches:

- Car industry
- Transport and handling
- Machine engineering
- Food and drinks industry
- Chemical industry
- Pharmaceuticals and petrochemicals

Deployment in the following areas:

- Assembly lines
- Conveying and handling systems
- Industrial manufacture
- Warehousing
- Logistics
- Distribution
- Order picking
- Transport logistics

is possible.

Examples:

Two extreme applications illustrate the flexibility and potential of *BLident*. For example VW in Saxony employs the *BLident* system to control the electric monorail overhead conveyors. The decisive factors in favour of deploying this system were the ranges,

transmission speeds as well the possibility of mounting the read-write heads very close to one another.

At Ford in Genk the high temperature data carriers have been deployed in the paint shop. Here the operating temperatures of up to 210°C were decisive in choosing to deploy *BLident* data carriers.

2 Commissioning a TURCK *BLident* system

Commissioning example using STEP 7 and PIB

In the following the commissioning of a *BLident* system using the SIMATIC basis software Step 7 and the standard software block „Proxy Ident Function Block“ (PIB) is described.

In order to carry out a first commissioning, simply and without any programming knowledge, TURCK has provided an example project to download.

Hardware description of the example project

For the following commissioning example the following hardware components were employed:

- S7 control „CPU 315-2DP“
- *BLident* interface module „TI-BL67-DP1-2“
- *BLident* read/write head „TN-CK-H1147“
- Data carrier „TW-R50-B128“

In the case of uncertainty concerning the correct connection i.e. wiring of your *BLident* system, the manuals D101580 (interface module) and D101582 (read/write heads and the data carriers) are available for download from the TURCK website.

Memory requirements for a *BLident* commissioning

Basic memory requirements

The basic memory requirements for the commissioning of the *BLident* system with the Proxy Ident Block are:
14KByte

Memory requirement per PIB instance (channel)

For each channel an instance of the Proxy Ident Block is generated.

Each channel requires, in addition to the basic memory requirement, **0.6 kbyte**.

Memory requirements for the read and write data

The Proxy Ident Function Block (PIB) occupies a memory area as send and receive buffer. The size of this memory area must be set in accordance with the data volume generated when reading and writing.

The TURCK *BLident* system provides PIB variants in order to cater for the different data volumes generated when reading and writing:

- PIB-1KB
- PIB_16K
- PIB_32K

The following calculation rule shows how you can calculate the memory requirements for reading and writing. It is assumed that the buffer is used by a number of channels/instances. If each instance was to be allocated its own buffer the memory requirements would be considerably higher. Following the completion of the calculation you can select the function block suitable for your application. The total data volume results from the following sum:

- The volume of data that is read via all active channels and saved to a "read only memory area".
- The volume of data that is written via all active channels and saved to a "write only memory area".
- The data volumes that can re-transmit both read and write data. The memory area is used alternatively as write memory area and read memory area. The required memory area can be reduced by a half.

Memory requirements for the hardware example

The „Hardware description of the example project” Page 2-2 provides 2 channels which can each be used for the connection of a read/write unit. The read memory area and the write memory area should be arranged separately.

The data carrier employed can store a maximum of 128 bytes. For the calculation a maximum of 200 bytes read memory and 200 bytes write memory have been allowed for. Each channel occupies 400 bytes. Both channels **800 bytes**.

For the **total memory requirement** the „Basic memory requirements” Page 2-2 and two times the „Memory requirement per PIB instance (channel)” Page 2-2 are added to the 800 bytes:

Total memory requirement
= 14 kbyte + (2 x 0.6 kbyte) + 800 bytes = 16 kbyte

Downloading the example project and the current device master file

The example project provided by TURCK enables you to easily reconstruct a first commissioning operation.

The current device master file is required in order to carry out the configuration of the *BLident*interface module.

The example project can be found under the following link:

http://www.turck.de/en....

(RFID > *BLident* Product Presentation > Sample Project S7...)

The example project is available as a ZIP file. Keep it in zipped form and note the location where you have stored the file.

The correct device master file is available at:

http://www.turck.de/en....

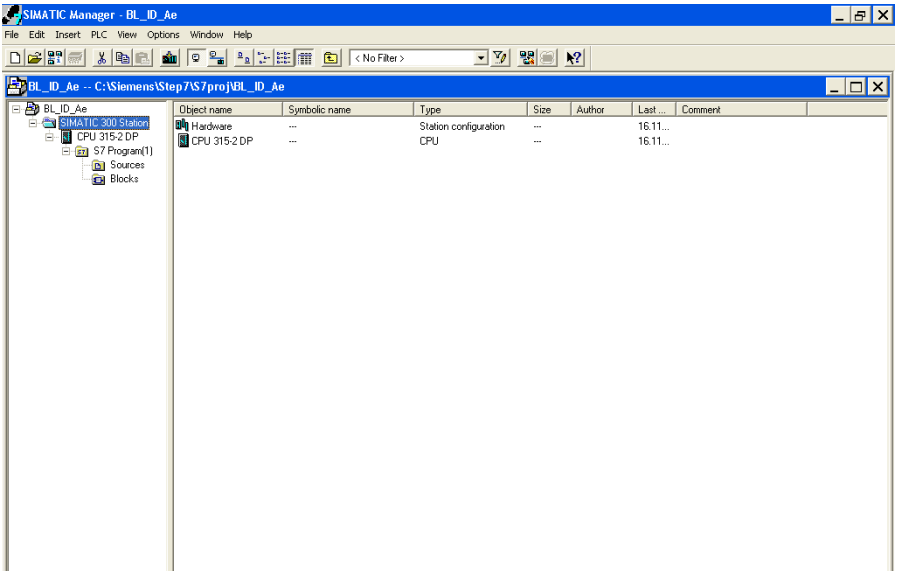
(Download > configuration > GSD PROFIBUS)

The procedural instructions provided in this document will enable you to commission a range of different applications.

Starting the S7 software and loading the example project

Update the device master file if required (before or after Start). Start the „SIMATIC Basis software Step 7“. Following Start the window "SIMATIC manager" is activated.

Illustration: 1
After the start of the SIMATIC Manager



The example project is opened with:
File > dearchive

Select the TURCK example file from your directory:

„BL_ID_A.zip“

The SIMATIC manager suggests a save location (target directory) for your *BLident* test project. You can confirm or alter this. Confirm "Open the file".

Hardware configuration and I/O addresses

Double clicking on "SIMATIC" in the directory tree on the left side of the window calls up, amongst others, "Hardware" on the right side of the window. Here you can make any necessary adjustments to the configuration due to a different hardware setup to that of the example project. Double clicking on the PROFIBUS station (here BL67) enables you to view the hardware configuration.

You can change the I/O addresses suggested by the SIMATIC manager. It is of advantage if you maintain the suggested I and O addresses "2..5", for this example.

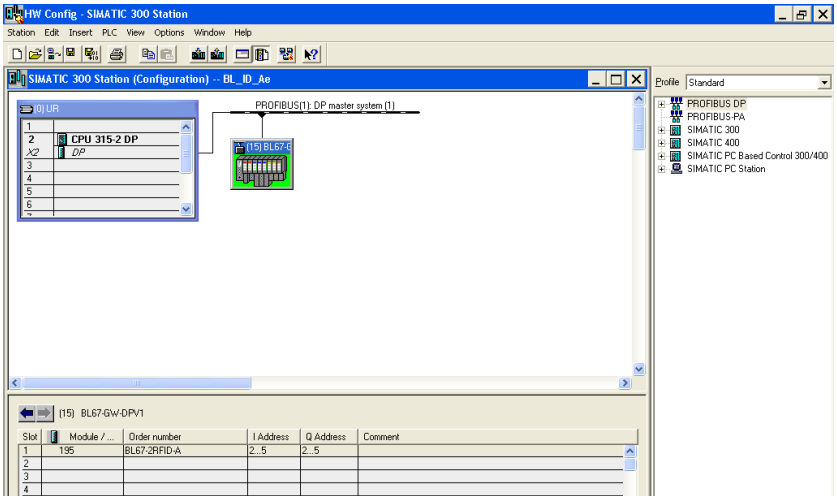
Note

When using the function block PIB it is necessary to choose the same value for Input and Output addresses.

Transfer the configuration data to the automation system (Target system > load.)

Confirm the request for the module to be restarted.

Illustration: 2
Hardware configurator



Setting up the PIB function block

The main settings in the example project have already been made.

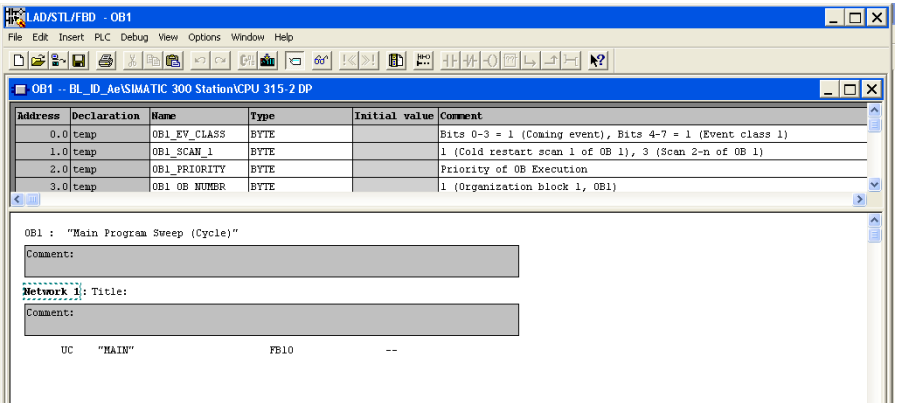
The following explanations serves to improve general understanding so that you will also be able to commission applications which differ from that of this example project.

Close the hardware configurator if it is still open.

Open the folder "Blocks" in the project tree in the left window (last point in the project tree). The block OB1 represents the top program level which is cyclically processed by the CPU.

By double clicking on OB1 you can view the program structure.

Illustration: 3



The main program OB1 calls up in principle the FB10. Close OB1 and double click in the block folder on FB10.

PIB variable table with the FB10

The FB 10 assigns the variables according to specification (formal parameters) to the variables for the PIB instance of a channel (actual parameter).

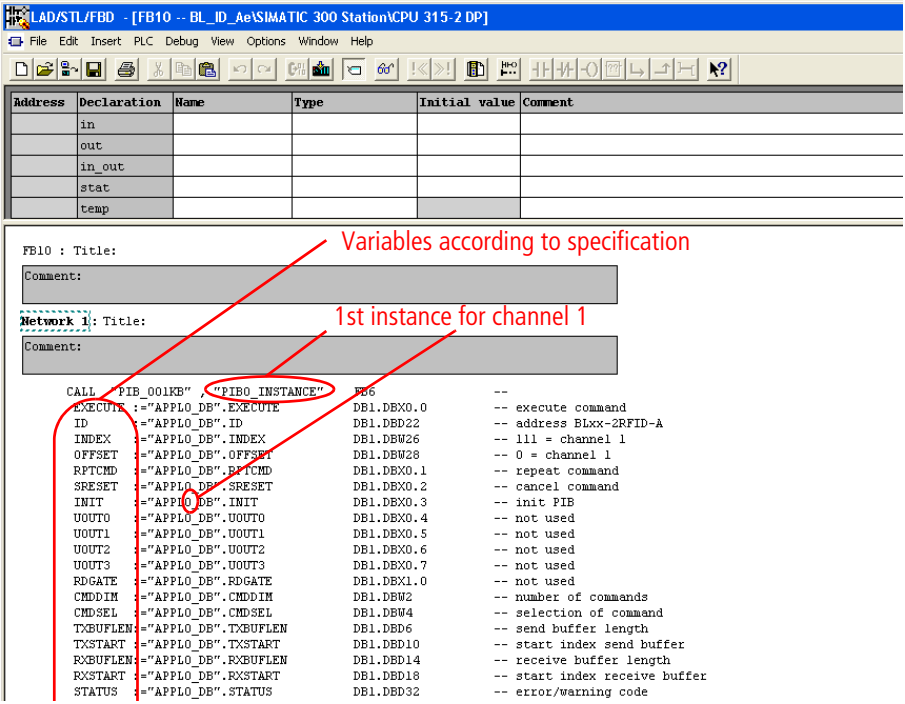
An explanation of all the variables carried out in this block can be found in „3 Proxy Ident Block (PIB) Definition” Page 3-7

As 2 channels for a *BLident*communication are available in the TURCK example project, two "instances" of the Proxy Ident Block are generated.

The PIB instance for the 1st channel is labelled "0". All the variables for the 1st instance also contain a "0".

The 2nd channel is accordingly "1" etc.

Illustration: 4
Variables for the
1st instance



Monitoring and controlling with the variable table vartable_pibX

Close the FB10 and open via the block folder the variable table vartable_pib0. This table belongs to the 1st instance of the PIB and thus to channel 1.

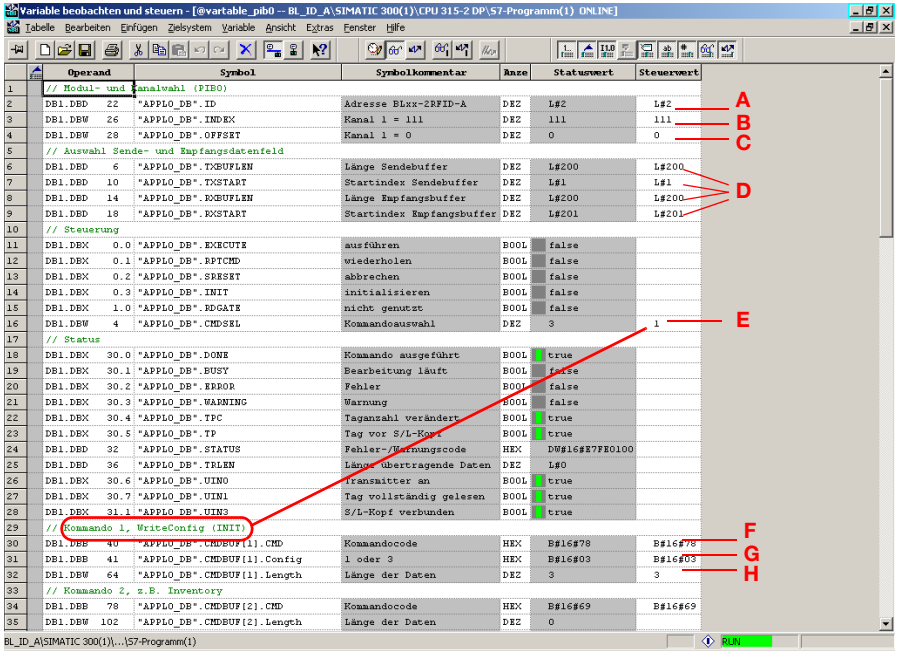
In order to read the status values and load the control values activate the online connection to your control (Target system > connect to directly attached CPU). The mode "RUN" indicates green in the right hand corner of the window.

Adjust the values indicated in the points A to D of the explanatory text in the column control values, should your application differ from that of the example project.

Note

Load the values into your control (variable control) and check using the column status values (variable monitor) that the control has accepted the values!

Illustration: 5



A *This is the start address of the BLidentprocess data for both channels. The address area for each individual channel is fixed using the offset (explanatory text point* **C***).*

B *The index "111" indicates that the next execution of a data transfer (also parameter data) is to channel 1. The index "112" relates to channel 2 etc. Index "118" addresses channel 8. These indexes are employed universally (including parameter data transfer). The indexes 101 to 108, which according to the specification are to be selected for parameter data transfer, are no longer employed.*

C *This offset is added to the start address (***A***). The address calculated refers to the process data of a channel. Here the offset is "0" as vartable_pib0 belongs to the first channel. The process data for a BLident channel makes up 2 bytes. The corresponding offset in vartable_pib1, which belongs to the 2nd channel is "2".*

D *Here the buffer area for the read and write data for the 1st instance (1st channel) is given. The „Memory requirements for the read and write data" Page 2-2 is, with the selection of a PIB-1KB, limited to a total of 1 kbyte. Here it is indicated that the send buffer for the 1st channel occupies the area 1 to 200. The receive buffer occupies the area 201 to 400.*
For the 2nd channel the areas 401 to 600 and 600 to 800 are occupied accordingly (vartable_pib1).

E *Here 1, 2 or 3 can be entered, when only one from 3 possible commands (see vartable_pibX: command 1, WriteConfig (INIT), command 2, e.g. Inventory...) is to be executed. As only the Write-Config command („Write-Config" Page 3-38) is to be executed, "1" has already been entered here.*

F *The hexadecimal coding for the command "Write-Config" is* 78_{hex}.

G *Changes to the configuration data can be made according to the specification („Config" Page 3-39)*
by means of a reset (01_{hex}),
by the writing of the new data (02_{hex})
with a combination of reset and new configuration data (03_{hex}) *(as in the example).*

H *The number of configuration data that is to be written.*
(Here there are 3 configuration data elements for the data carrier, which will be described in more detail in the next section.)

Configuration data of the data carrier

Entering the values in E to H (Picture 6) completes the preparations for the sending of 3 configuration values.

The 3 configuration values describe the type of data carrier from which the data is to be read from and written to.

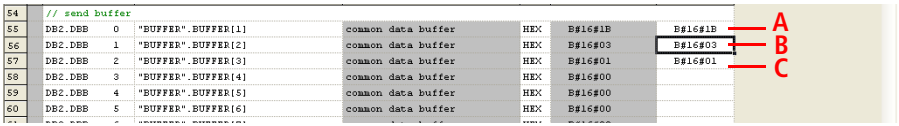
In our example it is the „TW-R50-B128.“ On this data carrier the user data is arranged in 28 blocks of 4 bytes.

The 3 configuration values for the data carrier in the TURCK example project are already in the "transmit data field" of vartable_pib0. The transmit data field is beneath the command section of vartable_pibX (here X=0).

Note

Load all the values into your control (variable control) and check using the column status values (variable monitor) that the control has accepted the values!

Illustration: 6
Transmit data field
of vartable_pib0



A *This hexadecimal value $1B_{hex}$* stands for the *number of 28 blocks on the data carrier. Note that the number cannot be simply converted to a hexadecimal value. The allocation is as follows:*
00_{hex} = > 1block
...
$1B_{hex}$ => 28 blocks etc.
Subtract "1" from the number of blocks and then convert this to the hexadecimal value!

B *This hexadecimal value 03_{hex}* stands for the *number of 4 bytes per block on the data carrier. Note that the number cannot be simply converted to a hexadecimal value. The allocation is as follows:*
00_{hex} = > 1Byte
...
03_{hex} => 4 bytes etc.

Subtract "1" from the number of bytes per block and then convert this to the hexadecimal value!

C *With the value 01_{hex} the transmitter (aerial) of the read/write head is turned on. With the value 00_{hex} you can turn off the transmitter.*

Initialisation of the 1st channel

In the section „Setting up the PIB function block” Page 2-8 you have been familiarised with the relevant settings (control values) for an initialisation. If your *BLident* project differs from that of the example project then you have adjusted the settings.

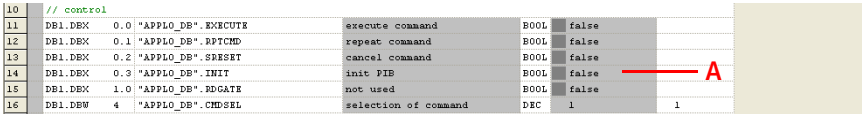
Note

Load all the values that have been described in the previous sections into your control (variable control)! Check via the status values column (variable monitor), that the control has accepted the values!

Now conduct the initialisation. Make sure that the online connection to your control is active. The mode "RUN" indicates green in the right hand corner of the window.

With a "rising signal edge" of the control variable “APPL0_DB“.INIT the command "Initialisation" is executed. You can generate the rising signal edge by changing the variable from "false" to "true". Enter either "1" or "true" as the control value.

Illustration: 7
Control field of vartable_pib0



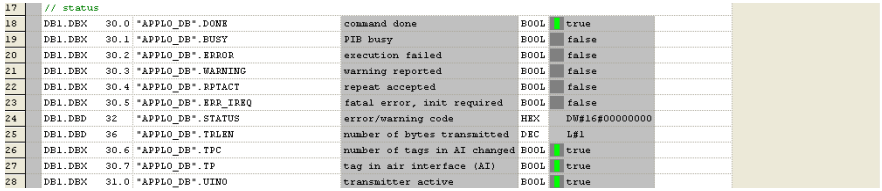
A *The initialisation is carried out with the rising signal edge (change from false-> true or 0->1)*

With:
Variable > control or

the command "Initialisation" is executed.

You can view the execution of the command in the status field of vartable_pib0.

Illustration: 8
Status field of vartable_pib0



17		// status				
18	DB1.DBX	30.0	"APPL0_DB".DONE	command done	BOOL	true
19	DB1.DBX	30.1	"APPL0_DB".BUSY	PIB busy	BOOL	false
20	DB1.DBX	30.2	"APPL0_DB".ERROR	execution failed	BOOL	false
21	DB1.DBX	30.3	"APPL0_DB".WARNING	warning reported	BOOL	false
22	DB1.DBX	30.4	"APPL0_DB".RPTACT	repeat accepted	BOOL	false
23	DB1.DBX	30.5	"APPL0_DB".ERR_IREQ	fatal error, init required	BOOL	false
24	DB1.DBD	32	"APPL0_DB".STATUS	error/warning code	HEX	DW#16#00000000
25	DB1.DBD	36	"APPL0_DB".TBLEN	number of bytes transmitted	DEC	L#1
26	DB1.DBX	30.6	"APPL0_DB".TPC	number of tags in AI changed	BOOL	true
27	DB1.DBX	30.7	"APPL0_DB".TP	tag in air interface (AI)	BOOL	true
28	DB1.DBX	31.0	"APPL0_DB".UINO	transmitter active	BOOL	true

The status variable “APPL0_DB“.DONE changes momentarily to the condition "Busy" and then returns to indicating "Command executed"= "true". The error free execution is confirmed by “APPL0_DB“.ERROR = false

You can find explanations of a number of the error codes for the status variables „APPL0_DE“.STATUS, in particular those for the *BLident* specific errors, in „Warnings and error messages” Page 2-36

A complete description of the status data can be found in „3.1.2 Error and Warning Concept” Page 3-18.

Reset the variable “APPL0_DB“.INIT to „false“, when the initialisation has been successfully completed.

With:
Variable > control or

"false" becomes the status value again.

Reading the UID from the data carrier/ channel

Each RFID data carrier receives a „UDP" Page 4-10 (unique identifier) during manufacture. The UID provides a unique global TAG identification number consisting of 8 bytes.

The reading of the UID is executed with the command "inventory". The instruction code 69_{hex} for the inventory is already entered into the field "Command 2" of the vartable_pib0 in the TURCK example project. A complete description of this instruction code can be found in „Inventory" Page 3-44.

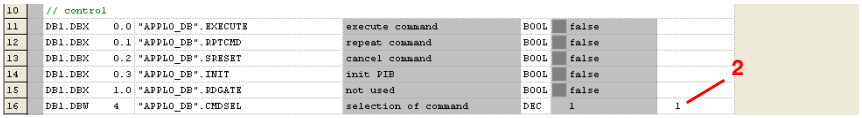
Illustration: 9
Command 2 of the vartable_pib0

Make sure that the online connection to your control is active. The mode "RUN" indicates green in the right hand corner of the window.

The control value "APPL0_DP".CMDSEL of the vartable_pib0 is most likely to still have the value "1" as you have just carried out the "initialisation" with the command 1.

Enter "2" for this control value in order to select command 2.

Illustration: 10
Control field of vartable_pib0



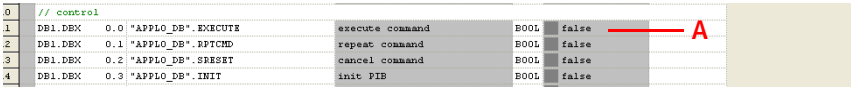
Note

Load **all** the values into your control (variable control) and check using the column status values (variable monitor) that the control has accepted the values!

With a "rising signal edge" of the control variable "APPL0_DB".EXECUTE the command "Inventory" is shown. You

can generate the rising signal edge by changing the variable from "false" to "true". Enter either "1" or "true" as the control value.

Illustration: 11
Control field of vartable_pib0



A *The "Inventory"proceeds with the rising signal edge (change from false-> true or 0->1)*

With:
Variable > control or

the command "Inventory" is executed.

You can view the execution of the command in the status field of vartable_pib0.

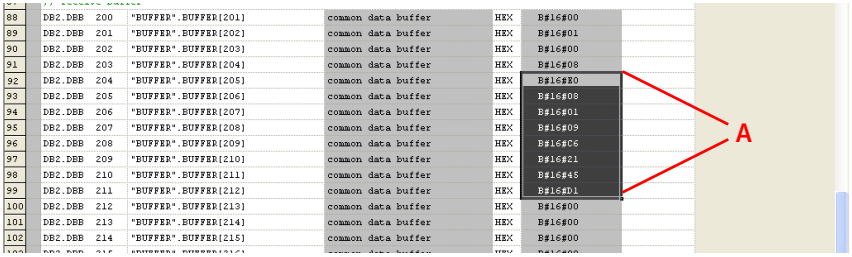
The status variable “APPL0_DB“.DONE changes momentarily to the condition "Busy" and then returns to indicating "Command executed"= "true". The error free execution is confirmed by “APPL0_DB“.ERROR = false

A total of 8 UID data bytes are received with "Inventory". The size of the data transmitted here is “APPL0_DB“.TRLEN =„12“.

A complete description of the status data can be found in „3.1.2 Error and Warning Concept” Page 3-18.

You can now read off the "Unique Identifier / UID" from the received data field in vartable_pib0.

Illustration: 12
The UID in the received data field



88	DB2.DBB	200	"BUFFER".BUFFER[201]	common data buffer	HEX	B#16#00
89	DB2.DBB	201	"BUFFER".BUFFER[202]	common data buffer	HEX	B#16#01
90	DB2.DBB	202	"BUFFER".BUFFER[203]	common data buffer	HEX	B#16#00
91	DB2.DBB	203	"BUFFER".BUFFER[204]	common data buffer	HEX	B#16#08
92	DB2.DBB	204	"BUFFER".BUFFER[205]	common data buffer	HEX	B#16#E0
93	DB2.DBB	205	"BUFFER".BUFFER[206]	common data buffer	HEX	B#16#08
94	DB2.DBB	206	"BUFFER".BUFFER[207]	common data buffer	HEX	B#16#01
95	DB2.DBB	207	"BUFFER".BUFFER[208]	common data buffer	HEX	B#16#09
96	DB2.DBB	208	"BUFFER".BUFFER[209]	common data buffer	HEX	B#16#C6
97	DB2.DBB	209	"BUFFER".BUFFER[210]	common data buffer	HEX	B#16#21
98	DB2.DBB	210	"BUFFER".BUFFER[211]	common data buffer	HEX	B#16#45
99	DB2.DBB	211	"BUFFER".BUFFER[212]	common data buffer	HEX	B#16#D1
100	DB2.DBB	212	"BUFFER".BUFFER[213]	common data buffer	HEX	B#16#00
101	DB2.DBB	213	"BUFFER".BUFFER[214]	common data buffer	HEX	B#16#00
102	DB2.DBB	214	"BUFFER".BUFFER[215]	common data buffer	HEX	B#16#00

A *The 8 byte UID is located from byte 5 onwards in the received data buffer*

Reset the variable “APPL0_DB“.EXECUTE to „false“, when the UID has been successfully read.

With:
Variable > control or

"false" becomes the status value again.

Writing on the data carrier / channel 1

This section explains the procedure for writing 32 byte data with any content onto your RFID data carrier.

It is possible to write on the data carrier of the 1st channel after you have carried out the „Initialisation of the 1st channel” Page 2-16.

In this example we have selected data that are easy to recognise again in the subsequent „Reading from the data carrier / channel 1” Page 2-25.

The writing of the data is carried out with the command "PhysicalWrite". Enter the command code 71_{hex} for the PhysicalWrite in the field "command 3“ of vartable_pib0. A complete description of this instruction code can be found in „Physical-Write” Page 3-36.

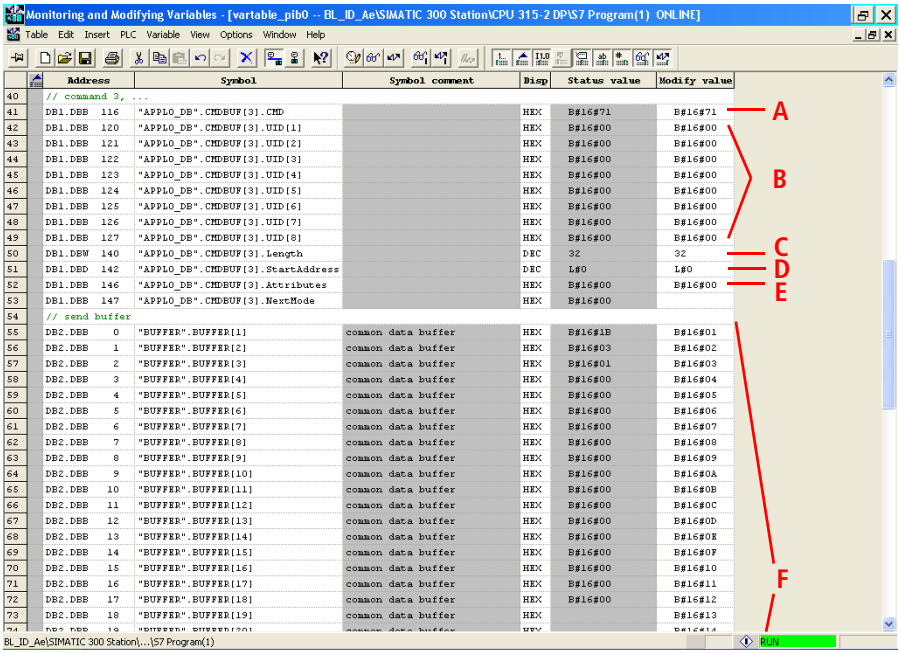
Make sure that the online connection to your control is active. The mode "RUN" indicates green in the right hand corner of the window.

Enter "3" for the control value “APPL0_DP“.CMDSEL of vartable_pib0 in order to select the command 3.

In the transmit data field enter the 32 byte as a hexadecimal number. Following the writing, we will explain the reading. The following numerical sequence: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B... 20, which is easy to remember, is entered.

You can enter the hexadecimal numbers either single or double digit and without any further format details (B#16#..). The SIMATIC manager converts them to the correct format.

Illustration: 13
Preparing the vartable_pib0 for writing



A *Here the code for the command which is to be carried out next is entered, in this case it is command 3. An overview of all the possible commands can be found in „3.1.3 Commands" Page 3-33.*
The command code 71_{hex} stands for the order to write on a physically available TAG „PhysicalWrite".

B *This 8 byte data field can contain a UID. This UID is always compared to the TAG's UID when values ≠00 are entered here.*
Make sure that all 8 bytes have the value "00" if you don't want to carry out the UID compare function.

C *Here you enter the number of bytes that should be transmitted from the transmit data field. The number of possible bytes is dependent on the size of the transmit data field (Page 2-13) and the memory capacity of the TAG employed. In this TURCK example 32 bytes are written to the TAG (112 byte). The size of the transmit data field here is 200 bytes.*

D *The memory areas on the TAG can be explicitly addressed using this address. In this example the start address L#0 has been selected. Note*

that the start address can only be the first byte of a block. TAGs with 4 byte blocks are addressed with L#0, L#3,...

E *These "attributes" enable a number of commands to be specified in more detail. This value is not taken into account with the read and write command.*
In „3.1.3 Commands" Page 3-33 you will find an overview of all possible commands and the options available for specifying these in more detail via "attributes". Please also take note of the following „BLident definitions on the command and diagnosis level" Page 2-29.

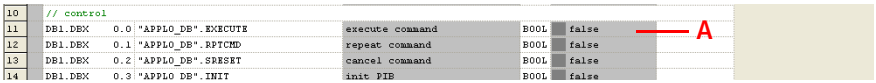
F *In this transmit data field you enter the data you want to write on your TAG.*

Note

Load all the values in the fields "control", "command 3" and "transmit data field" in your control (Variable control) and check via the status value column (Variable monitor), that the control has accepted the values!

With a "rising signal edge" of the control variable "APPL0_DB".EXECUTE the command „PhysicalWrite" is executed. You can generate the rising signal edge by changing the variable from "false" to "true". Enter either "1" or "true" as the control value.

Illustration: 14
Control field of vartable_pib0



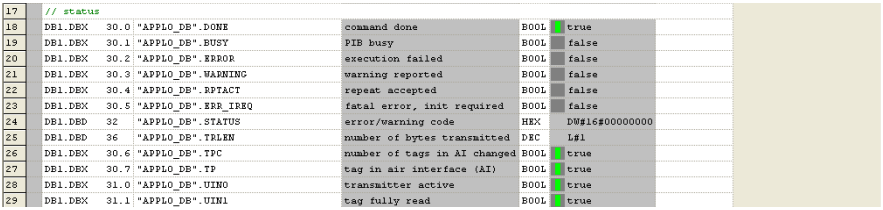
A *"PhysicalWrite" proceeds with the rising signal edge (change from false-> true or 0->1)*

With:
Variable > control or

the command "PhysicalWrite" is executed.

You can view the execution of the command in the status field of vartable_pib0.

Illustration: 15
Status field of
vartable_pib0



17	// status					
18	DB1.DBX	30.0	"APPL0_DB".DONE	command done	BOOL	true
19	DB1.DBX	30.1	"APPL0_DB".BUSY	PIB busy	BOOL	false
20	DB1.DBX	30.2	"APPL0_DB".ERROR	execution failed	BOOL	false
21	DB1.DBX	30.3	"APPL0_DB".WARNING	warning reported	BOOL	false
22	DB1.DBX	30.4	"APPL0_DB".RPTACT	repeat accepted	BOOL	false
23	DB1.DBX	30.5	"APPL0_DB".ERR_IREQ	fatal error, init required	BOOL	false
24	DB1.DBD	32	"APPL0_DB".STATUS	error/warning code	HEX	DW#16#00000000
25	DB1.DBD	36	"APPL0_DB".TPLEN	number of bytes transmitted	DEC	L#1
26	DB1.DBX	30.6	"APPL0_DB".TPC	number of tags in AI changed	BOOL	true
27	DB1.DBX	30.7	"APPL0_DB".TP	tag in air interface (AI)	BOOL	true
28	DB1.DBX	31.0	"APPL0_DB".UIN0	transmitter active	BOOL	true
29	DB1.DBX	31.1	"APPL0_DB".UIN1	tag fully read	BOOL	true

The status variable “APPL0_DB“.DONE changes momentarily to the condition "Busy" and then returns to indicating "Command executed"= "true". The error free execution is confirmed by “APPL0_DB“.ERROR = false

A complete description of the status data can be found in „3.1.2 Error and Warning Concept” Page 3-18.

Reset the variable “APPL0_DB“.EXECUTE to „false“, when the writing has been successfully completed.

With:
Variable > control or

"false" becomes the status value again.

Reading from the data carrier / channel 1

This section explains the procedure for reading 32 byte data with any content from your RFID data carrier.

It is possible to read from the data carrier of the 1st channel after you have carried out the „Initialisation of the 1st channel” Page 2-16.

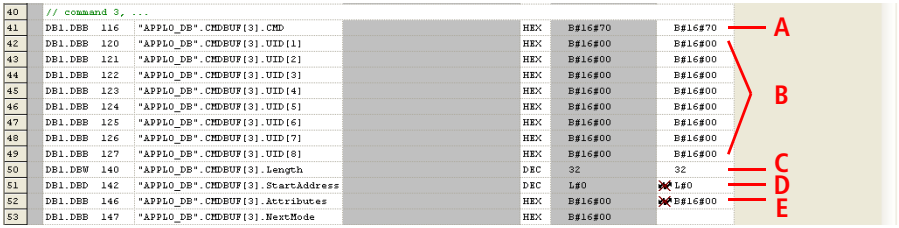
In the previous section you have written data (arbitrary) to the data carrier using the command "PhysicalWrite". In this section the same data will now be read from the data carrier using the command "PhysicalRead".

Enter the command code 70_{hex} for the PhysicalRead in the field "command 3“ of vartable_pib0. A complete description of this instruction code can be found in „Physical-Read” Page 3-35.

Make sure that the online connection to your control is active. The mode "RUN" indicates green in the right hand corner of the window.

Enter "3" for the control value “APPL0_DP“.CMDSEL of vartable_pib0 in order to select the command 3.

Illustration: 16
Preparing the vartable_pib0 for writing



A *Here the code for the command which is to be carried out next is entered, in this case it is command 3. An overview of all the possible commands can be found in „3.1.3 Commands" Page 3-33.*
The command code 70_{hex} stands for the order to read from a physically available TAG „PhysicalRead".

B *This 8 byte data field can contain a UID. This UID is always compared to the TAG's UID when values ≠00 are entered here.*
Make sure that all 8 bytes have the value "00" if you don't want to carry out the UID compare function.

C *Here you enter the number of bytes that should be transmitted to the received data field. The number of possible bytes is dependent on the size of the received data field (Page 2-13) and the memory capacity of the TAG employed. In this TURCK example 32 byte are read from the TAG (112 byte). The size of the received data field here is 200 byte.*

D *The memory areas on the TAG can be explicitly addressed using this address. In this example the start address L#0 has been selected. Note that the start address can only be the first byte of a block. TAGs with 4 byte blocks are addressed with L#0, L#3,...*

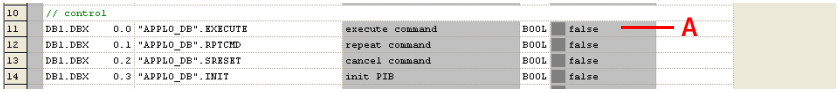
E *These "attributes" enable a number of commands to be specified in more detail. This value is not taken into account with the read and write command.*
In „3.1.3 Commands" Page 3-33 you will find an overview of all possible commands and the options available for specifying these in more detail via "attributes". Please also take note of the following „BLident definitions on the command and diagnosis level" Page 2-29.

Note

Load all the values in the fields "control", "command 3" in your control (Variable control) and check via the status value column (Variable monitor), that the control has accepted the values!

With a "rising signal edge" of the control variable “APPL0_DB“.EXECUTE the command „PhysicalRead“ is executed. You can generate the rising signal edge by changing the variable from "false" to "true". Enter either "1" or "true" as the control value.

Illustration: 17
Control field of vartable_pib0



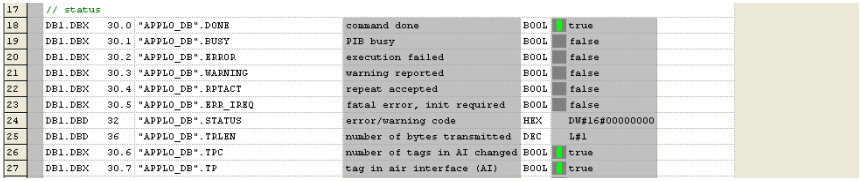
A *"PhysicalRead" proceeds with the rising signal edge (change from false-> true or 0->1)*

With:
Variable > control or

the command "PhysicalRead" is executed.

You can view the execution of the command in the status field of vartable_pib0.

Illustration: 18
Status field of vartable_pib0



The status variable “APPL0_DB“.DONE changes momentarily to the condition "Busy" and then returns to indicating "Command executed"= "true". The error free execution is confirmed by “APPL0_DB“.ERROR = false

The size of the received data here is “APPL0_DB“.TRLEN =„32“.

A complete description of the status data can be found in „3.1.2 Error and Warning Concept” Page 3-18.

In the received data field of vartable_pib0 you will find the data read from the TAG:

Illustration: 19 Received data field of vartable_pib0 following successful reading

87		// receive buffer					
88	DB2.DBB	200	"BUFFER".BUFFER[201]	common data buffer	HEX	B#16#01	
89	DB2.DBB	201	"BUFFER".BUFFER[202]	common data buffer	HEX	B#16#02	
90	DB2.DBB	202	"BUFFER".BUFFER[203]	common data buffer	HEX	B#16#03	
91	DB2.DBB	203	"BUFFER".BUFFER[204]	common data buffer	HEX	B#16#04	
92	DB2.DBB	204	"BUFFER".BUFFER[205]	common data buffer	HEX	B#16#05	
93	DB2.DBB	205	"BUFFER".BUFFER[206]	common data buffer	HEX	B#16#06	
94	DB2.DBB	206	"BUFFER".BUFFER[207]	common data buffer	HEX	B#16#07	
95	DB2.DBB	207	"BUFFER".BUFFER[208]	common data buffer	HEX	B#16#08	
96	DB2.DBB	208	"BUFFER".BUFFER[209]	common data buffer	HEX	B#16#09	
97	DB2.DBB	209	"BUFFER".BUFFER[210]	common data buffer	HEX	B#16#0A	
98	DB2.DBB	210	"BUFFER".BUFFER[211]	common data buffer	HEX	B#16#0B	
99	DB2.DBB	211	"BUFFER".BUFFER[212]	common data buffer	HEX	B#16#0C	
100	DB2.DBB	212	"BUFFER".BUFFER[213]	common data buffer	HEX	B#16#0D	
101	DB2.DBB	213	"BUFFER".BUFFER[214]	common data buffer	HEX	B#16#0E	
102	DB2.DBB	214	"BUFFER".BUFFER[215]	common data buffer	HEX	B#16#0F	
103	DB2.DBB	215	"BUFFER".BUFFER[216]	common data buffer	HEX	B#16#10	
104	DB2.DBB	216	"BUFFER".BUFFER[217]	common data buffer	HEX	B#16#11	
105	DB2.DBB	217	"BUFFER".BUFFER[218]	common data buffer	HEX	B#16#12	
106	DB2.DBB	218	"BUFFER".BUFFER[219]	common data buffer	HEX	B#16#13	
107	DB2.DBB	219	"BUFFER".BUFFER[220]	common data buffer	HEX	B#16#14	
108	DB2.DBB	220	"BUFFER".BUFFER[221]	common data buffer	HEX	B#16#15	
109	DB2.DBB	221	"BUFFER".BUFFER[222]	common data buffer	HEX	B#16#16	
110	DB2.DBB	222	"BUFFER".BUFFER[223]	common data buffer	HEX	B#16#17	

Reset the variable “APPL0_DB“.EXECUTE to „false“, when the reading has been successfully completed.

With:
Variable > control or

"false" becomes the status value again.

BLident definitions on the command and diagnosis level

The commissioning of the TURCK *BLident* system with the „Proxy Ident Function Block“ differs in a number of respects from the „Excerpt from the Specification” Page 3-1 (PROFIBUS Specification) . The differences relate to the command and diagnosis levels.

The following conformity table indicates which command and status, i.e. diagnostic messages from the complete specification are supported by *BLident* :

Table 1: Conformity table

Name	Type	Additional information to the TURCK specific instance	Conform? (Y/N)
To point 3.1.4 of the PROFIBUS specification			
Read	Command		N
Get	Command		Y
Physical-Read	Command		Y
Write	Command		N
Put	Command		N
Physical-Write	Command		Y
Format	Command		N
Create	Command		N
Delete	Command		N
Clear	Command		N
Update	Command		N
Next	Command		Y
Get-Directory	Command		N
Set-Attribute	Command		N

Table 1: Conformity table

Name	Type	Additional information to the TURCK specific instance	Conform? (Y/N)
Get-Attribute	Command		N
Write-Config	Command		Y
Read-Config	Command		Y
Mem-Status	Command		Y
Dev-Status	Command		Y
Inventory	Command		Y
Read-Bar-Code	Command		N
To point 4.2.1 of the PROFIBUS specification			
Reading-Gate	Control-Bit		N
Repeat-Command	Control-Bit		N
Soft-Reset	Control-Bit		Y
To point 4.2.2 of the PROFIBUS specification			
Target_Presence_Changed	Status Bit		Y
Target_Present	Status Bit		Y
Soft_Reset_Active	Status Bit		Y
Repeat_Command_Active	Status Bit		N
Busy	Status Bit		Y
Error	Status Bit		Y
UIN3	Status Bit	R/W head is connected	Y

Table 1: Conformity table

Name	Type	Additional information to the TURCK specific instance	Conform? (Y/N)
UIN2	Status Bit		N
UIN1	Status Bit	TAG data completely read in the R/W head	Y
UIN0	Status Bit	Transmitter activated (see also WriteConfig)	Y

The following commands differ from the PROFIBUS specification:

Write-Config

This command is activated in the PIB by INIT and supports Config=1 (reset only) and Config=3 (write data and reset). Config data 3 bytes in size can be written. The Config data includes the option to turn the transmitter of the R/W head on and off. The turning on and off of the transmitter can be used to prevent the mutual interference of R/W heads situated close to one another.

Table 2: Configuration data

Byte	Bit	Meaning
0	0..7	O must be reserved
1	0..7	O must be reserved
2	0	1= Transmitter On / 0= Transmitter Off, (default = 1)
	1..7	O must be reserved

Example of configuration data

- Tag with 28 blocks each with 4 byte

„0x1B, 0x03, 0x01“

Read-Config

This command reads the Config data written under Write-Config

Possible command specific STATUS values following faulty execution:

Table 3:

STATUS	Description of the error
DW#16#E**4**FE**82**xx	There is no transceiver connected.

Inventory

Only the parameter Attribute = 0 is supported. („Inventory” Page 3-44)

Physical-Read

The parameters StartAddress and Length (+StartAddress) must be within the tag's address range. („Physical-Read” Page 3-35)

Physical-Write

The parameters StartAddress and Length (+StartAddress) must be within the tag's address range. („Physical-Write” Page 3-36)

Mem-Status

The command Mem-Status supports the attribute 0x40 (physical status information).
The response of the tag to a GET_SYSTEM_INFORMATION command is transmitted to ISO/IEC15693-3:

- Byte 0 = Serial number (UID), LSB
- .
- .
- Byte 7 = Serial number (UID), MSB

- Byte 8 = Data carrier format (DSFID)
- Byte 9 = Application identification (AFI)
- Byte 10 = Memory size: Number of blocks-1
- Byte 11 = Memory size: Bytes/block-1
- Byte 12 = IC identification

Dev-Status

Only the parameter Attribute = 0x04 is supported. A data record according to the I&M specification I&M0 is returned. The connected transceiver is described. („Dev-Status" Page 3-42)

Example:

Table 4:

From byte	to byte	Field	Content
0	9	Manufacturer specific header	0, 0, 0, 0, 0 ,0 ,0 ,0 ,0 ,0
10	11	MANUFACTURER_ID	0x0B12 (299 = TURCK)
12	31	ORDER_ID	‚TN-CK40-H1147', 0x00, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20 , 0x20
32	47	SERIAL_NUMBER	(not supported)
48	49	HARDWARE _REVISION	0x0003
50	53	SOFTWARE_REVISION	‚V' (0x56), 0x01, 0x03, 0x00 (= V1.3.0)
54	55	REVISION_COUNTER	(not supported)

Table 4:

From byte	to byte	Field	Content
56	57	PROFILE_ID	0x5B00 (identification system, PIB profile)
58	59	PROFILE_SPECIFIC_TYPE	0x0000
60	61	IM_VERSION	0x01, 0x01 (= I&M V1.1)
62	63	IM_SUPPORTED	0x01, I&M0 supported

Next

Only the parameter NextMode = 0 or 1 is supported. („Next” Page 3-37)

Get

This command enables you to set the write protection for a tag's block.

In order to do this it is necessary to know the memory organisation of the tag employed (number and size of the blocks).

Attention

Once the write protection has been set for a block it is no longer possible to undo this!

Table 5: Transmit data field for the command "Get"

Byte in transmit data field	Content
0	0x02

Table 5: Transmit data field for the command "Get"

Byte in transmit data field	Content
1	UID of the tag, UID = 0 -> arbitrary
2	UID of the tag
3	UID of the tag
4	UID of the tag
5	UID of the tag
6	UID of the tag
7	UID of the tag
8	UID of the tag
9	0x67
10	Block number of the block to be switched to write protected (0x00 = 1. Block, 0xFF = 256. Block)

If successful the following data is returned:

Table 6:

Byte in received data field	Content
0	0x02
1	0x67
2	Command index

If an error occurs this will be displayed accordingly in STATUS.

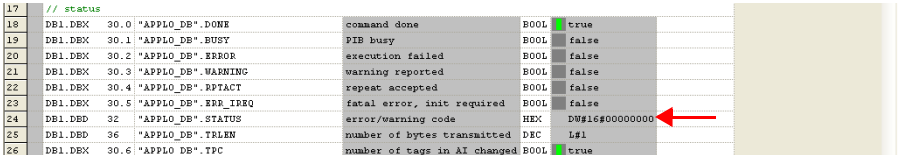
Further commands

An overview of further commands can be found in „3.1.3 Commands” Page 3-33. To execute the command proceed as with reading and writing.

Warnings and error messages

With the variable „APPLO_DB“.STATUS an error and warning code is transmitted.

Illustration: 20



17		// status				
18	DB1.DBX	30.0	"APPLO_DB".DONE	command done	BOOL	true
19	DB1.DBX	30.1	"APPLO_DB".BUSY	PIB busy	BOOL	false
20	DB1.DBX	30.2	"APPLO_DB".ERROR	execution failed	BOOL	false
21	DB1.DBX	30.3	"APPLO_DB".WARNING	warning reported	BOOL	false
22	DB1.DBX	30.4	"APPLO_DB".RPTACT	repeat accepted	BOOL	false
23	DB1.DBX	30.5	"APPLO_DB".ERR_IREQ	fatal error, init required	BOOL	false
24	DB1.DBD	32	"APPLO_DB".STATUS	error/warning code	HEX	DW#16#00000000
25	DB1.DBD	36	"APPLO_DB".TRLEN	number of bytes transmitted	DEC	L#1
26	DB1.DBX	30.6	"APPLO DB".TPC	number of tags in AI changed	BOOL	true

The following table provides information on the meaning of a number of STATUS values:

Table: 7

Status value of „APPLO_DB“.STATUS	Meaning of the error code
DW#16#E4FE01xx	The output current of the channel is too high (e.g. short circuit). The channel is turned off.
DW#16#E4FE82xx	There is no read/write head connected.
DW#16#E4FE03xx	The transmitter (aerial) of the read/write head is turned off. With the „Configuration data of the data carrier” Page 2-14 the activation of the transmitter can be requested. The execution is carried out with a subsequent initialisation (Page 2-16) using the "WriteConfig" command.
DW#16#E4FE04xx	The command is not accepted. Cause: Following an initial Next command („Next” Page 3-37), NextMode defines an event. Only when this event has occurred is the command accepted and executed.
***BLident* specific error codes**	

Table: 7

Status value of „APPLO_DB“.STATUS	Meaning of the error code
DW#16#E4FE89xx	The R/W head continuously registers a CRC error in the RS485 connection. Check whether there is an EMC problem!
DW#16#E4FE8Axx	The Ident unit continuously registers a CRC error in the RS485 connection. Check whether there is an EMC problem!
DW#16#E4FE81xx	The R/W head is defect.
DW#16#E4FE88xx	The R/W head is not receiving enough power.
DW#16#E1FE04xx	The data carrier (tag) is defect.
DW#16#E4FE82xx	No R/W head is connected.

3 Excerpt from the Specification

1 General

TURCK publishes this excerpt of the PROFIBUS Specification „Profile for Identification Systems, Proxy Ident Function Block“ (Version 1.02, December 2005) by courtesy of the PNO (Profibus Nutzer Organisation).

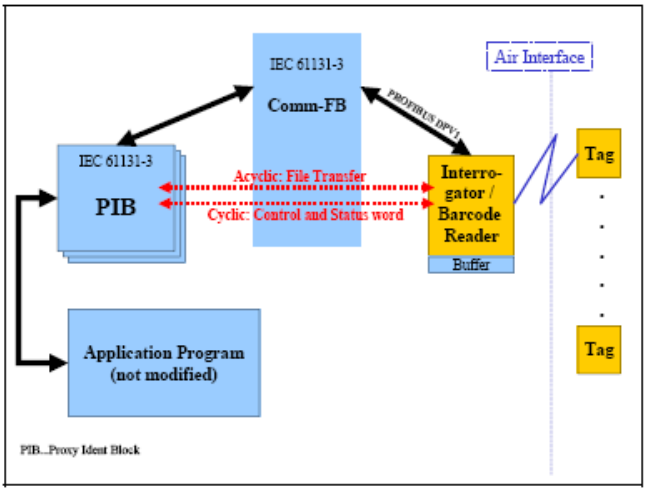
Please see also „BLident definitions on the command and diagnosis level” Seite 2-29.

1.6 Functional Requirements

This subclause defines the functional requirements coming from the application of RFID and BR systems. They define the restrictions/ limitations for the design of the PIB and the corresponding functions to be realized within the field device. As there are different functions regarding RFID and BR they are described in parallel.

1.6.1 General Requirements

The basic idea is to adapt existing RFID and BR systems to Profibus technology enabling integration into existing systems and to ease the use in new systems (refer to figure below).



As existing proprietary solutions have to overcome there are certain restrictions to be considered in addition to the conformance to Profibus technology.

It is required that:

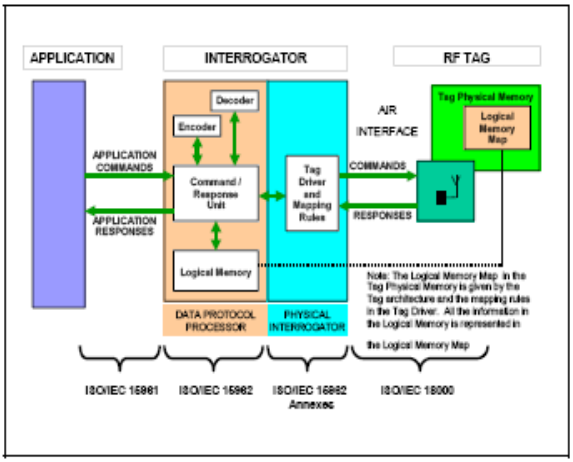
- "Integration of RFID and BR systems do interfere with existing Profibus systems to a minimum extend
  - Control flow is done with cyclic communication
  - data transfer is done with acyclic communication
  - application programs are kept independent from introducing PIB
  - fragmentation, de-fragmentation of data packages is kept inside the PIB

1.6.2 Requirements coming from the use of Radio Frequency Identification Systems

Regarding RFID systems actually standardization activities are under progress. These activities are targeted on defining the air interface (ISO/IEC 18000) as well as the file format and handling of files (ISO/IEC 15962) as illustrated below.

This profile specification explicitly focuses on data transfer via industrial networks based on Profibus as well as on the integration into programmable controller systems.

Figure 21: Related Standardization Activities

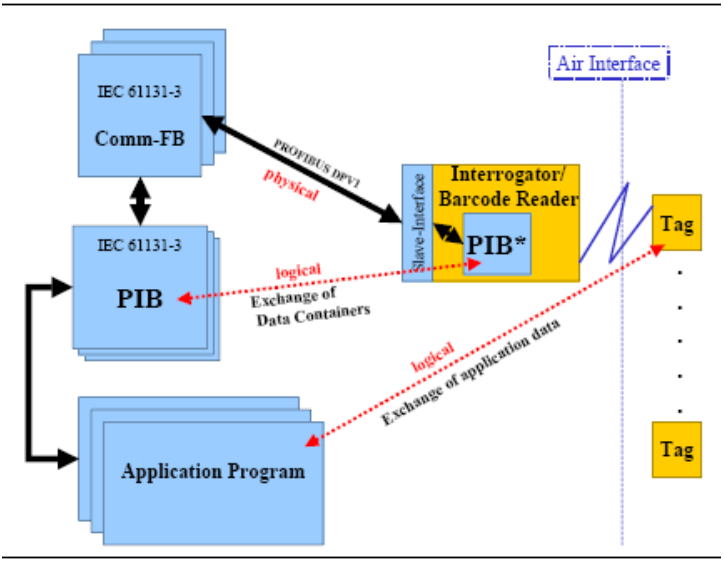


Because of these ongoing processes it seems not to be needed to consider these subjects within this document. Later on, if the standardization within IEC is fin-ished the related documents can be referenced. Consequently this document will not consider:

- Air interface,
- File format,
- File handler,
- User data coding.

Currently it is usual to consider data as a packet that is interpreted by both appli-cations the one in the Ident Unit and the one within the PLC using the PIB (see figure below):

Figure 22: Transfer of data



2.1 Principles of Modelling

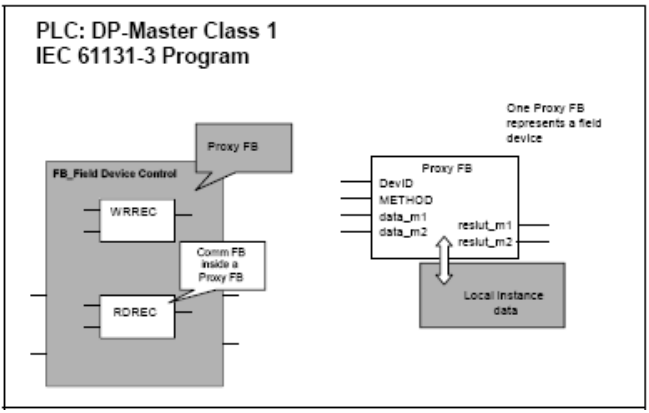
The following principles of modelling for the Proxy Ident Block (PIB) have to be met:

- to fit into the existing PLC systems, e.g. using the existent addressing concept
- to be efficient and without overhead; that means the model shall be performance oriented
- to enable an easy application program portation between different PLC systems
- to use directly the existing Comm FBs
- to apply good programming style is to avoid dependencies of the hardware configuration data such as addressing in the application program

2.2 General PIB Model

The PIB is modeled as a Proxy FB representing a complete Ident Unit. It follows the basic concepts of Proxy FB modeling as described in [4].

Figure 23: Usage of Comm FB and Proxy FB for PIB modeling



2.3 Representation

The representation of the interface of function block types is given in graphical and textual form according IEC 61131-3. The behavior of the function blocks is presented as a graphical state diagram with tables for the transitions and the actions.

3 Proxy Ident Block (PIB) Definition

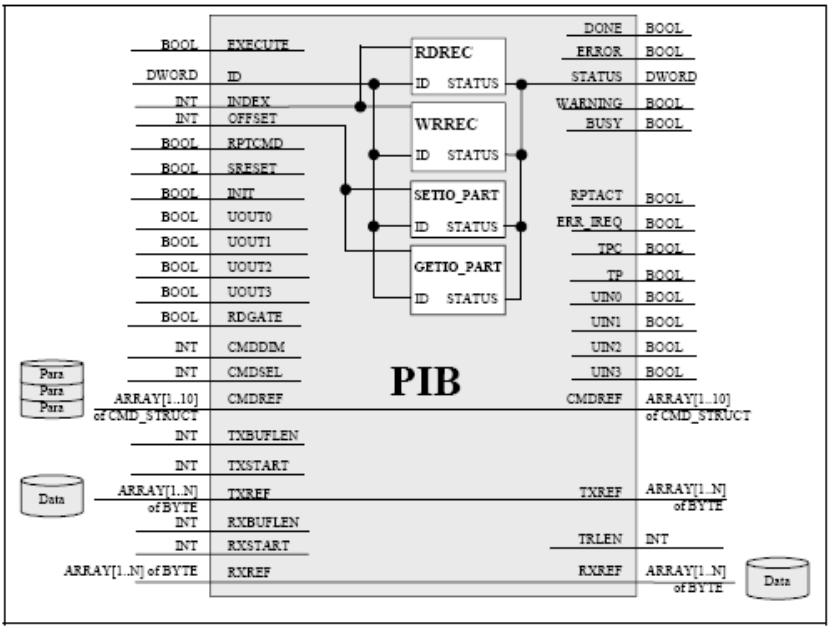
This chapter provides the specification of the Proxy Ident Block following the guidelines as stated within [4].

3.1 Functional Description

NOTE: In case several PIB instances are operated by an application program in parallel it must be guarantied that individual instances do not block each other.Block Definition

The following figure provides a graphical presentation for the interface of the PIB:

Figure 24:
Graphical Presentation of the PIB Interface



3.1.1 Parameters

Tabelle: 8 PIB parameter description

Name	Description
EXECUTE	The user (application program) can start the operation of a block instance by setting this input parameter of type BOOL to TRUE. Before starting the block operation the command and its associated parameters have to be put to the memory associated to the CMDREF parameter. Also the CMDSEL parameter has to be set properly. This parameter is activated with the rising edge.
ID*	The value of this input parameter is used as a unique identifier for addressing a single device or a slot within a device. The detailed description is given in [4]. In combination with the parameter "INDEX" it addresses an individual Ident Unit.
INDEX*	The value of this input parameter is used to identify a single ident channel within a slot of a device. The use of the parameter corresponds to the definition of the address parameter "Index" within [1] and [2].
OFFSET*	The value of this input parameter is used to identify the channel related I/O data as transferred cyclically as sub-part of the data associated to a slot (module).
RPTCMD*	This parameter forces the Ident Unit to repeat the command currently executed or next to be executed as long as the parameter is set to "1". The parameter is mapped to the "Repeat_Command" bit of the cyclic control word (see chapter 4.2.1)
SRESET*	If this input parameter is set to "1" the command currently processed within the Ident Unit will become cancelled. The parameter is mapped to the "Soft_Reset" bit of the cyclic control word (see chapter 4.2.1). This parameter is activated with the rising edge.

*Tabelle: 8**(Forts.)** PIB parameter description*

Name	Description
INIT*	This input parameter forces, if set from "0" to "1", the Ident Unit to restart its operation except the communication interface. Local Ident Unit operations done within this procedure are manufacturer specific. The parameter is mapped to the "Init" bit of the cyclic control word (see chapter 4.2.1). After the sequence "Init_Active=0 ' Init_Active=1 Init_Active=0" (cyclic status word) has been completed the PIB has to send automatically a Write-Config command to the Ident Unit. This parameter is activated with the rising edge.
UOUT0*	This parameter of type BOOL represents the user specific bit 0 as defined to be transferred within the cyclic control word (see chapter 4.2.1). The parameter is mapped to bit 0 of the cyclic control word.
UOUT1*	This parameter of type BOOL represents the user specific bit 1 as defined to be transferred within the cyclic control word (see chapter 4.2.1). The parameter is mapped to bit 1 of the cyclic control word.
UOUT2*	This parameter of type BOOL represents the user specific bit 2 as defined to be transferred within the cyclic control word (see chapter 4.2.1). The parameter is mapped to bit 2 of the cyclic control word.
UOUT3*	This parameter of type BOOL represents the user specific bit 3 as defined to be transferred within the cyclic control word (see chapter 4.2.1). The parameter is mapped to bit 3 of the cyclic control word.
RDGATE*	This parameter of type BOOL represents the optional bit 8 as defined to be transferred within the cyclic control word (see chapter 4.2.1). If set to "1" means start operation.

*Tabelle: 8**(Forts.)** PIB parameter description*

Name	Description
CMDDIM*	In order to write a more efficient user application several Commands can be stored within the memory. The number of commands stored influences the range of memory to be instantiated for this PIB instance. The input parame-ter CMDDIM defines the number of command parameter structures "CMD_STRUCT"
CMDSEL*	As there may be a certain number of commands stored in parallel there must be a selector for a single command to be executed. The input parameter CMDSEL is used for this purpose. It references one of the predefined commands. The first CMD_STRUCT is reserved for parameters associated to the command "Write_Config".
CMDREF*	This In_Out parameter references a global memory range. This memory range is used to store commands and their associated parameters. The max. number of commands stored in association with a single PIB instance shall not exceed 10. Chapter 3.1.3 defines the commands supported by the profile version.
TXBUFLEN*	This input parameter specifies the number of bytes used by this instance of the PIB for storing data to be send. The number is calculated starting from the relative position within the memory range defined by the TXSTART pa-rameter. For consistency reasons it is recommended not to modify this parameter after the PIB is instantiated.
TXSTART*	The input parameter TXSTART indicates the relative position of the "TXBUF" within the global memory area referenced by the parameter "TXREF". This is the starting point of memory assigned to the PIB instance. For consistency reasons it is recommended not to modify this parameter after the PIB is instantiated.

*Tabelle: 8**(Forts.)** PIB parameter description*

Name	Description
TXREF*	This In_Out parameter is a reference to a global memory area used by several blocks. The PIB instance may share the memory with several other blocks.
RXBUFLEN*	This input parameter specifies the number of bytes used by this instance of the PIB for storing data received. The number is calculated starting from the relative position within the memory range defined by the RXSTART parameter. For consistency reasons it is recommended not to modify this parameter after the PIB is instantiated.
RXSTART*	The input parameter RXSTART indicates the relative position of the "RXBUF" within the global memory area referenced by the parameter "RXREF". This is the starting point of memory assigned to the PIB instance. For consistency reasons it is recommended not to modify this parameter af-ter the PIB is instantiated.
RXREF*	This In_Out parameter is a reference to a global memory area used by several blocks. The PIB instance may share the memory with several other blocks.
TRLEN	This output parameter indicates the number of user bytes transmitted (sent or received - depending on the command executed) after the command succeeded. It counts the number of data transferred starting at the position referenced by (TXSTART + OffsetBuffer) or (RXSTART + OffsetBuffer).
DONE	This output parameter indicates, if set to "TRUE", that a command has been executed with a positive result. The application program may check this flag before calculating the data transferred while executing the command.

*Tabelle: 8**(Forts.)** PIB parameter description*

Name	Description
ERROR	This output parameter is used to indicate, if set to "TRUE" that an error has been recognized. The error may be recognized local (within the host) or remote (within the Ident Unit). Detailed error information is provided by the "STATUS" parameter. The flag is set by the PIB internally depending on the Error bit within the acknowledgement frame (Bit 0 of CI). The flag is reset to "FALSE" by the PIB instance after a new call of a command.
WARNING	This output parameter is used to indicate, if set to "TRUE" that a warning has been recognized. The warning may be recognized local (within the host) or remote (within the Ident Unit). Detailed warning information is provided by the "STATUS" parameter. The flag is set by the PIB internally depending on the warning bits within the acknowledgement frame (Bit 1..7 of CI - one or more bits are set to 1). The flag is reset to "FALSE" by the PIB instance after a new call of a command. Note: In case of WARNING all user data associated to a command may be transferred correctly (if ERROR is not set). In this case the data buffer contains valid values.
STATUS	The output parameter "STATUS" provides detailed error information or/and warning information for the last command that has been executed with a negative result or warning indication. The value changes not before the next appearance of a warning or error. Coding depends on the source of the warning or error which may be reported from the embedded CommFBs, functions to access cyclic data, the Ident Unit, tag or from the PIB internal functions.
BUSY	This output parameter indicates, if set to "TRUE", that the PIB is "busy" and a command will no be accepted to become started (except "INIT" and "SRESET"). The parameter is intended to indicate the user that the PIB is still operating.

*Tabelle: 8**(Forts.)** PIB parameter description*

Name	Description
RPTACT	This output parameter indicates, if set to "TRUE", that the PIB* accepted the request for repeating the current command. It is mapped from the "Repeat_Command_Active" bit of the cyclic input data. The output parameter remains active as long as the bit within the cyclic telegram is set. The PIB* provides data following a command execution as long as this output parameter is set - the user shall read command result.
ERR_REQ	This output parameter indicates, if set to "TRUE", that the PIB* noticed a fatal error. It is mapped from the Error_Flag of the cyclic input data. The output parameter remains active as long as the bit within the cyclic telegram is set. The PIB remains in the current state (state machine). After noticing that this parameter is "TRUE" the user has to set the INIT input parameter of the PIB or send a Dev-Status command (if possible).
UINO	This output parameter of type BOOL represents the user specific bit 0 as defined to be transferred within the cyclic status word (see chapter 4.2.2).
UIN1	This output parameter of type BOOL represents the user specific bit 1 as defined to be transferred within the cyclic status word (see chapter 4.2.2).
UIN2	This output parameter of type BOOL represents the user specific bit 2 as defined to be transferred within the cyclic status word (see chapter 4.2.2).
UIN3	This output parameter of type BOOL represents the user specific bit 3 as defined to be transferred within the cyclic status word (see chapter 4.2.2).

*Tabelle: 8**(Forts.)** PIB parameter description*

Name	Description
TP	This output parameter indicates, if set to "1", that a target is in the range of the Ident Unit. The parameter corresponds to the "Target_Present" bit of the cyclic status word (see chapter 4.2.2). It becomes reset in case there is no target in the range of the Ident Unit. In case an Ident Unit does not support this feature the parameter is set to "0". This parameter is not used for barcode reader
TPC	This output parameter indicates, if set to "1", that a new target is in the range of the Ident Unit. The parameter corresponds to the "Target_Presence_Changed" bit of the cyclic status word (see chapter 4.2.2). The parameter is reset to "0" after the next "Inventory" command succeeded. In case an Ident Unit does not support this feature the parameter is set to "0". This parameter is not used for barcode reader.

*The application is responsible for resetting/changing all the input parameters.

Table 9: PIB parameter characteristics

Name	Data Type	Range	Conditions for use m=mandatory, o=optional
EXECUTE	BOOL	If DONE≠0 or ERROR≠0 a change from "0" to "1" of the parameter EXECUTE forces the function block to execute a command. (if DONE=0 and ERROR=0 the PIB has not been initialized or a command is in process).	m

*Table 9:**(Forts.)** PIB parameter characteristics*

Name	Data Type	Range	Conditions for use m=mandatory, o=optional
ID	DWORD	0x00000000 .... 0xFFFFFFFF (Refer to [4])	m
INDEX	INT	101...108, 111...118,201...254	m
OFFSET	INT	0...244	m
RPTCMD	BOOL	0 = no Repeat_Command 1 = Repeat_Command	o
SRESET	BOOL	Change from "0" to "1" forces the function block to execute the SRESET command (termination of the last actual command).	m
INIT	BOOL	Change from "0" to "1" forces the function block to execute the initialization procedure.	m
UOUT0	BOOL	Bit 0 = 0/1	o
UOUT1	BOOL	Bit 0 = 0/1	o
UOUT2	BOOL	Bit 0 = 0/1	o
UOUT3	BOOL	Bit 0 = 0/1	o
RDGATE	BOOL	0 = reading gate is not active 1 = reading gate is active	o
CMDDIM	INT	2 ... 10	m

*Table 9:**(Forts.)** PIB parameter characteristics*

Name	Data Type	Range	Conditions for use m=mandatory, o=optional
CMDSEL	INT	1 ...10	m
CMDREF	ARRAY[2.. CMDDIM] von CMD_STRUCT	CMDDIM x 42	m
TXBUFLEN	INT	0 ... 32768	m
TXSTART	INT	0 ... 32768	m
TXREF	ARRAY [1..N]of BYTE		m
RXBUFLEN	INT	0 ... 32768	m
RXSTART	INT	0 ... 32768	m
RXREF	ARRAY [1..N]of BYTE	N	m
TRLEN	INT	0 ... 32768	m
DONE	BOOL	0 = command not completed 1 = command completed	z
ERROR	BOOL	0 = last command finished without error 1 = last command finished with error	z

*Table 9:**(Forts.)** PIB parameter characteristics*

Name	Data Type	Range	Conditions for use m=mandatory, o=optional
WARNING	BOOL	0 = no warning information available 1 = warning information available	z
STATUS	DWORD	Refer to chapter 3.1.2	z
BUSY	BOOL	0 = PIB currently does not execute a command 1 = command currently executed by the PIB	z
RPTACT	BOOL	0 = no repeat command function active at the PIB* 1 = repeat command function active at the PIB*	o
ERR_REQ	BOOL	0 = no error indicated from the PIB* 1 = error indicated by the PIB*	m
UINO	BOOL	Bit 0 = 0/1	o
UIN1	BOOL	Bit 0 = 0/1	o
UIN2	BOOL	Bit 0 = 0/1	o
UIN3	BOOL	Bit 0 = 0/1	o
TP	BOOL	0 = no target present 1 = target present	o (not used for barcode reader)

*Table 9:**(Forts.)** PIB parameter characteristics*

Name	Data Type	Range	Conditions for use m=mandatory, o=optional
TPC	BOOL	0 = no target changed 1 = target changed	o (not used for barcode reader)

To ease the access to individual elements of the command structure a common structure has been defined to be used for every command independent of the use of single parameters. The use of a parameter depends on the definition of the command itself. The parameters are defined in chapter 3.1.3

Abbildung: 25 Data Type definition for the PIB command:

```
TYPE
    CMD STRUCT
        STRUCT
            CMD : BYTE;
            Config : BYTE ;
            OffsetBuffer : INT;
            UID : ARRAY[1..8] OF BYTE;
            FileName : ARRAY[1..8] OF BYTE;
            Offset : DINT;
            Length : INT;
            StartAddress : DINT;
            Attributes : BYTE;
            NextMode : BYTE;
            Timeout : INT;
            ObjectNumber : INT;
            FileType : WORD;
      END STRUCT;
END TYPE
```

3.1.2 Error and Warning Concept

The PIB indicates if the requested command was performed successfully or not. The error indication is typically used for two purposes:

1 To change the reaction to the process i.e. to implement a substitute reaction e.g. to repeat the request at another time or another place or to abort the process task.

2 To issue an alarm message to a HMI system by the application program or by the PLC system automatically.

Note

In case 1 only very few different reactions dependent on the indicated error are typical. Detailed error information is hardly used..

As the PIB encapsulates other FB's (Communication FB's) and functions, the STATUS parameter represents status information generated by several sources.

Also the parameter STATUS can be used to provide warning information.

Optionally to the use of the STATUS parameter it is possible to transmit error and warning information within diagnostic data (refer to chapter 4.4).

The STATUS output has the data type DWORD which is interpreted as a packed array of four bytes as described in the following table.

Table 10: Structure of the Output STATUS

Byte	Name	Definition	Date type
	Function_Num	Refer to table 11	Byte
1	Error_Decode	Refer to table 12	Byte
2	Error_Code_1	Refer to table 13	Byte
3	Error_Code_2	warnings or vendor specific error	Byte

The Function_Num byte definition is based on (IEC 61158_6, Part 6; 1999) and extends the additions made in (PROFIBUS Comunication and Proxy Function Blocks acc. to IEC 61131-3 Vers. 1,2). It is used for grouping of failures and warnings.

Table 11: Function_Num values

Frame Selector (Bit 7), deCimal	PDU_ Identifier (Bit 5 to 6), decimal	Error_Code_2 used for warning indication (Bit 4)	Function_ Code / Error_ Code (Bit 0 to 3) decimal	Description acc. To this profile
0	0 . . . 3	0/1	0 ... 15	No failurer
1	0, 1	0/1	0 . . . 15	Error not related to DP protocol and not defined for this profile
1	2	0/1	0 . . . 15	Error message regarding DP protocol, refer to / 2/ and [4]
1	3	0/1	0	vendor specific coding of Error_Code_1 and Error_Code_2
1	3	0/1	1	Error_Code_1 provides error information related to Tag/ Transponder/ Barcode vendor specific coding of Error_Code_2

*Table 11:**(Forts.)** Function_Num values*

Frame Selector (Bit 7), deCimal	PDU_ Identifier (Bit 5 to 6), decimal	Error_Code_2 used for warning indication (Bit 4)	Function_ Code / Error_ Code (Bit 0 to 3) decimal	Description acc. To this profile
1	3	0/1	2	Error_Code_1 provides error information related to air interface vendor specific coding of Error_Code_2
1	3	0/1	3	Error_ Code _1 provides error informa-tion related to file system vendor specific coding of Error_Code_2
1	3	0/1	4	Error_Code_1 provides error information related to Ident Unit (Interroga-tor/Barcode Reader) vendor specific coding of Error_ Code _2

*Table 11:**(Forts.)** Function_Num values*

Frame Selector (Bit 7), deCimal	PDU_ Identifier (Bit 5 to 6), decimal	Error_Code_2 used for warning indication (Bit 4)	Function_ Code / Error_ Code (Bit 0 to 3) decimal	Description acc. To this profile
1	3	0/1	5	Error_Code_1 provides error information regarding communication between PIB and Ident Unit (except DP error) vendor specific coding of Error_Code_2
1	3	0/1	6	Error_ Code _1 provides command specific error information vendor specific coding of Error_Code_2
1	3	0/1	7	Error_Code_1 provides error information generated internally by PIB vendor specific coding of Error_Code_2
1	3	0/1	8 ... 15	Not defined here

* Bit 4 = 0: Error_Code_2 contains vendor specific information, Bit 4 = 1: Error_Code_2 contains warning information (Byte 5 (CI) of acknowledge frame)

The Error_Decode byte is used to define the meaning of Function_Num, Er-ror_Code_1 and Error_Code_2.

Table 12: Error_Decode values

Error_Decode	Quelle	Bedeutung
	SPS	No error, no warning
	SPS	Warning (not used for this profile) verwendet)
	DP V1	Error reported according to IEC 61158 6 /2/ *111* gemeldet
	SPS	0x8x reports an error according the x-th parameter of the call of the Comm FB as defined in [4] Comm-FB, wie in [4] definiert
	Profile (PIB, Ident Unit)	profile-specific error r
	Profile (PIB, Ident Unit)	reserved for future use

The Error_Code_1 provides a number detailing the error or warning. Within the following table Error_Decode is fixed to 0xFE.

Table 13: Error_Code_1 values

Function _Code/ Error_ Code*	Error_ Code_1 (decimal)	Indi-cated by	Meaning	manda tory / option al
1	1	IG	Tag memory error (e.g. CRC error)	o

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function_Code/ Error_Code*	Error_Code_1 (decimal)	Indicated by	Meaning	mandatory / optional
1	2	IG	Presence error (indicated by Ident Unit), tag has left the transmission window	o
1	3	IG	Address or command does not fit the tag characteristics (memory size) (indicated by Ident Unit)	o
1	4	IG	Tag is defective. (replace tag or battery)	o
1	5	IG	Tag memory overflow.	o
1	6	IG	Unformatted tag.	o
1	7	IG	Inconsistent tag data structure. Reformat tag.	o
1	8	IG	tag within the transmission window does not have the expected UID (indicated by Ident Unit)	o
1	9	IG	Command not supported by the Tag	o
1	10	IG	Access violation (e.g. block locked) refer to ISO18000-x	o

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function _Code/ Error_ Code*	**Error_ Code_1 (decimal)**	**Indi- cated by**	**Meaning**	**manda tory / option al**
1	11..127	IG	Reserved for future profile use	o
1	128..255	IG	Vendor specific	o
2	1	IG	Communication timeout at air inter- face (indicated by Ident Unit)	o
2	2	IG	Communication timeout at air inter- face (indicated by Ident Unit)	o
2	3..127	IG	Reserved for future profile use	o
2	128..255	IG	Vendor specific	o
3	1	IG	Uncorrect file name (indicated by Ident Unit)	o
3	2	IG	File does not exist (indicated by Ident Unit)	o
3	3	IG	The Tag type is incorrect or unsuit- able for the selected mode of operation. No file system available on tag. (indicated by Ident Unit)	o

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function_Code/ Error_ Code*	**Error_ Code_1 (decimal)**	**Indi-cated by**	**Meaning**	**manda tory / option al**
3	4	IG	Create command; no more directory entries available. (indicated by Ident Unit)	o
3	5	IG	Create command; file already exists in direc-tory. (indi-cated by Ident Unit)	o
3	6	IG	Access right viola-tion. (indicated by Ident Unit)	o
3	7	IG	File length overflow (indicated by Ident Unit)	o
3	8	IG	File not accessible (corrupted) (indi-cated by Ident Unit)	o
3	9..127	IG	Reserved for future profile use	o
3	128..255	IG	Vendor specific	o
4	1	IG	Power supply failure (indicated by Ident Unit)	o
4	2	IG	Hardware failure within Ident Unit (indicated by Ident Unit)	o

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function_Code/ Error_ Code*	Error_ Code_1 (decimal)	Indi-cated by	Meaning	manda tory / option al
4	3	IG	Hardware failure within Ident Unit (indicated by Ident Unit)	o
4	4	IG	Command buffer overflow within Ident Unit (indi-cated by Ident Unit)	o
4	5	IG	data buffer over-flow within Ident Unit (indi-cated by Ident Unit)	o
4	6	IG	Command in this mode not supported by Ident Unit. (indicated by Ident Unit)	o
4	7	IG	IIdent Unit reports an unspecific error indicated by the cyclic Status Word (e.g. antenna does not work, …) This error is not related to a specific command..	o
4	8..127	IG	Reserved for future profile use	o
4	128..255	IG	Vendor specific	o
5	1	IG	Wrong sequence number (SN) indi-cated by Ident Unit	m

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function_Code/ Error_ Code*	Error_ Code_1 (decimal)	Indi-cated by	Meaning	manda tory / option al
5	2	PIB	Wrong sequence number (SN) indi-cated by PIB	m
5	4	IG	Invalid data block number DBN indi-cated by Ident Unit.	m
5	5	PIB	Invalid data block number DBN indi-cated by PIB.	z
5	6	IG	Invalid data block length DBL indi-cated by Ident Unit.	o
5	7	PIB	Invalid data block length DBL indi-cated by PIB.	z
5	8	IG	Command from another user being proc-essed (indi-cated by Ident Unit)	o
5	9	PIB	The Ident Unit proceeds a hard-ware reset (Init_Active set to "1"), Init (Bit 15 within the cyclic control word) is expected from the PIB.	z

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function _Code/ Error_ Code*	Error_ Code_1 (decimal)	Indi- cated by	Meaning	manda tory / option al
5	10	PIB	The command code "CMD" and the respective acknowledgement do not corre- spond. This is a software or synchronization error that cannot occur in normal operation.	z
5	11	PIB	Wrong sequence of acknowledge tele- grams (TDB/DBN)	z
5	12	PIB	Synchroization error (increment of AC_H/AC_L and CC_H/CC_L in cyclic control word is wrong), INIT needed to be performed	z
5	13..127	IG	Reserved for future profile use (must not be used)	o
5	128..255	IG	Vendor specific	o
6	1	IG	Invalid CMD	z
6	2	IG	Invalid command index CI indicated by Ident Unit.	z

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function_Code/ Error_Code*	Error_Code_1 (decimal)	Indicated by	Meaning	mandatory / optional
6	3	IG	Invalid command parameter (e.g. data range) indicated by Ident Unit.	o
6	4	IG	Wrong synchronization between application program and Tag. A command expected is missing. (Object detect error)	o
6	5	IG	Only Write-Config command permissible in this state. (indicated by Ident Unit)	z
6	6..127	IG	Reserved for future profile use	o
6	128..255	IG	Vendor specific	o
7	1	PIB	Only INIT command permissible in this state. (indicated by PIB)	z
7	2	PIB	Command code "CMD" not permissible	z

Table 13: ***(Forts.)*** *Error_Code_1 values*

Function_Code/ Error_Code*	Error_Code_1 (decimal)	Indicated by	Meaning	mandatory / optional
7	3	PIB	Parameter "Length" of the command too long, does not fit the global data reserved within the TXBUF. (indicated by PIB)	z
7	4	PIB	RXBUF overflow (more data received than memory in RXBUF available)	z
7	5	PIB	This indication tells the user that only a "INIT" is permitted as the next command. All other commands will be rejected.	z
7	6	PIB	Wrong Index (out of range: 101 .. 108)	z
7	7	PIB	Ident Unit does not respond to INIT (Init_Active expected in cyclic status message)	z
7	8	PIB	Timeout during Init (60 seconds as defined by TC3WG9)	z
7	9	PIB	Command repetition not supported by the PIB*	o

*Table 13: **(Forts.)** Error_Code_1 values*

Function_Code/ Error_Code*	Error_Code_1 (decimal)	Indicated by	Meaning	mandatory / optional
7	10	PIB	Error during packet size adoption within PIB.	m
7	11..127	PIB	Reserved for future profile use	o
7	128..255	PIB	Vendor specific	o

*) Bit 0 to 3 (decimal coded) of Function_Num IU … Ident Unit

The Error_Code_2 provides either warning information (if Bit 4 of Function_Num is set "1") or optional vendor specific number detailing the error (if Bit 4 of Function_Num is reset "0"). The warnings are mapped from the acyclic acknowledge telegram Byte 5 (CI)

Table 14: coding of warnings

Error_Decode	Error_Code_2 (Bit position)	Source	Meaning
	0	Ident Unit	Bit is fixed to "0"
	1	Ident Unit	Vendor specific
	2	Ident Unit	Vendor specific in case of Ident Unit being a Bar-code Reader Low Battery in case of Ident Unit being a RFID
	3	Ident Unit	Vendor specific
	4	Ident Unit	Vendor specific
	5	Ident Unit	Vendor specific
	6	Ident Unit	Vendor specific
	7	Ident Unit	Vendor specific

3.1.3 Commands

This chapter describes the commands supported by the PIB and their associated parameters. The following restrictions apply to the use of commands:

- "Cyclic control flow is of higher priority in relation to acyclic transmitted commands
- ""INIT" and "SRESET" are aborting the execution of a command within the Ident Unit
- "after sending a cyclic control message (INIT, SRESET) the consecutive change of parameter "DONE" is related to the cyclic control message and not to the command interrupted by INIT or SRESET
- "INIT" resets communication (cyclic control and status flow, acyclic commands) between PIB and Ident Unit. For this procedure first "Init" is sent within cyclic control word. After the "Init_Active" bit within the status word is reset a "Write-Config" command is generated and sent to the Ident Unit. This requires that the user is providing "Write-Config" parameters within the command area before requesting "INIT". (see chapter 4.2.3)
- "Write-Config" resets all functions within the Ident Unit except communication
- "SRESET" terminates the last command

Table 15: PIB command description

Name	Description
Get	This is the command to read data from a file (in case of RFID system) out of the Ident Unit. Data are stored in the RXBUF after the command succeeded. The parameter TRLEN of the PIB indicates the number of bytes received.
	VAR CMD : BYTE := 0x62; (* b *) END_VAR
Parameters	**Description**

*Table 15:**(Forts.)** PIB command description*

Name	Description	
	Offset-Buffer	This parameter specifies the relative offset within the TXBUF. It specifies the first address within the memory area where the first byte of the parameter data to be send is stored. Other parameter data follow consecutively.
	Length	This parameter specifies the number of bytes to be sent to the Ident Unit starting at the address indicated by the parameter Off-setBuffer. The range is: 0 … 226.

*Table 15:**(Forts.)** PIB command description*

Name	Description	
Physical-Read	This is the command to read data from a transponder using a physical start address and the length of data to be read. The parameter TRLEN of the PIB indicates the number of bytes received.	
	VAR CMD : BYTE := 0x70; (* p *) END_VAR	
	Parameters	**Description**
	Offset-Buffer	This parameter specifies the relative offset within the RXBUF. It specifies the first address within the memory area where the first byte of data received has to be stored. All following bytes have to be stored at incremented addresses.
	UID	This parameter identifies a single transponder. UID = 0: Any (no specific) The tag currently present is read.
	Length	This parameter specifies the number of bytes to be read from transponder starting at the address indicated by the parameter StartAddress.
	Start Address	This parameter specifies a physical address within the trans-ponder memory.

*Table 15:**(Forts.)** PIB command description*

Name	**Description**	
Physical-Write	This is the command to write data to a transponder using a physical start address and the length of data to be written.	
	VAR CMD : BYTE := 0x71; (* q *) END VAR	
	Parameters	**Description**
	Offset-Buffer	This parameter specifies the relative offset within the TXBUF. It specifies the first address within the memory area where the first byte of data to be send is stored.
	UID	This parameter identifies a single transponder. UID = 0: Any (no specific) The tag currently present is read.
	Length	This parameter specifies the number of bytes to be send to the transponder starting at the address indicated by the parameter StartAddress.
	Start Address	This parameter specifies a physical address within the trans-ponder memory.

*Table 15:**(Forts.)** PIB command description*

Name	**Description**	
Next	This is the command to finish operations regarding one transponder. The next command will be executed when the next transponder is recognized/ indicated.	
	VAR CMD : BYTE := 0x6E; (* n *) END VAR	
	Parameters	**Description**
	UID	This parameter identifies a single transponder. UID = 0: Any (no specific) The tag currently present is read.
	NextMode	valid values: NextMode = 0 (The next command can be applied to each (another or the same) tag) NextMode = 1 (only a different tag will become handled)

*Table 15:**(Forts.)** PIB command description*

Name	**Description**
Write-Config	This service is used to modify operation of the Ident Unit except interruption of the communication. It is possible to send new parameters to the Ident Unit (ConfigData). Also a reset can be initiated requesting the Ident Unit to restart operation. The TXBUF is used as manufacturer specific area for config data. Config data are manufacturer specific. Normally Write-Config is executed automatically by the PIB during INIT. Optionally Write-Config may be started with EXECUTE. Mandatory operation to be supported: Config = 1 Optional to be supported: Config = 2 Or Config = 3
	VAR CMD : BYTE := 0x78; (* x *) END VAR

Parameters	**Description**
Offset-Buffer	This parameter specifies the relative offset within the TXBUF. It specifies the first address within the memory area where the first byte of data to be sent is to be stored.
Length	This parameter specifies the number of config data byte to be written to the Ident Unit.

*Table 15:**(Forts.)** PIB command description*

Name	Description	
	Config	Config = 0 … not allowed Config = 1 … Reset, no ConfigData Config = 2 … no Reset, ConfigData to be send Config = 3 … Reset, ConfigData to be send Config > 3 … reserved
	Definition of sub-parameters delivered within the response:	
	MaxPack-etSize	This parameter is transmitted from the PIB* to the PIB and provides information about the max. length of the Ident PDU (Ident header + data) the slave device is able to receive or send. The PIB checks the PDU length supported by the PIB* dynamically within the initialization phase (INIT). PIB adapst the internal algorithm for packaging data and align the PDU size. 00 = Default (corresponds to 240 Byte) 64…240 = max. permitted PDU size of within the PIB* 01…63 = reserved 241…255 = reserved

*Table 15:**(Forts.)** PIB command description*

Name	**Description**
Read-Config	This service is used to read config data out of the Ident Unit. The RXBUF is used as manufacturer specific area for config data. Config data are manufacturer specific. The parameter TRLEN of the PIB indicates the number of bytes received.
	VAR CMD : BYTE := 0x61; (* a *) END VAR
Parameters	**Description**
Offset-Buffer	This parameter specifies the relative offset within the RXBUF. It specifies the first address within the memory area where the first byte of data read is to be stored.

*Table 15:**(Forts.)** PIB command description*

Name	Description	
Mem-Status	This service is used to read the status of a tag (battery condition, memory size, type of tag, available capacity, …). The RXBUF is used as manufacturer specific area for status data. Status data are manufacturer specific. The parameter TRLEN of the PIB indicates the number of bytes received.	
	VAR CMD : BYTE := 0x73; (* s *) END VAR	
	Parameters	**Description**
	UID	This parameter identifies a single transponder. UID = 0: Any (no specific) The tag currently present is read.
	Attributes	This parameter is used to specify the class of information to be read. Valid values are: 0x00…reserved 0x01…warning detail 0x02…reserved 0x03…reserved 0x04…physical status information (man. Spec. details) 0x05…file system related status information (man. Spec. details) 0x06 - 0x7F reserved 0x80 - 0xFF manufacturer specific
	Offset-Buffer	This parameter specifies the relative offset within the RXBUF. It specifies the first address within the memory area where the first byte of data read is to be stored.

*Table 15:**(Forts.)** PIB command description*

Name	**Description**
Dev-Status	This service is used to read the status of an Ident Unit. The RXBUF is used as manufacturer specific area for status data. Status data are manufacturer specific. The pa-rameter TRLEN of the PIB indicates the number of bytes received.
	VAR CMD : BYTE := 0x74; (* t *) END VAR
	Parameters / **Description**
	Attributes / This parameter is used to specify the class of information to be read. Valid values are: 0x00… reserved 0x01… warning detail (man. Spec. details) 0x02… error history (man. Spec. details) 0x03… command history (man. Spec. details) 0x04… channel related I&M information (data record I&M0) 0x05… channel related I&M information (data record I&M1) 0x06… channel related I&M information (data record I&M2) 0x07… channel related I&M information (data record I&M3) 0x08… channel related I&M information (data record I&M4) 0x09 - 0x7F reserved 0x80 - 0xFF manufacturer specific

*Table 15:**(Forts.)** PIB command description*

Name	Description	
	Offset-Buffer	This parameter specifies the relative offset within the RXBUF. It specifies the first address within the memory area where the first byte of data read is to be stored.

*Table 15:**(Forts.)** PIB command description*

Name	Description
Inventory	This command is used to request a list of all UID's currently accessible within the an-tenna lobe. Vendor specific it is possible to deliver additional information. The RXBUF is structured as follows. Following an example is given to illustrate the construction of the data transferred. It is not intended to be used as Structured Text code within the PLC program. The exam-ple represents the transmission of 5 objects (ObjectNumber = 5) and ObjectLength = 16. VAR CONSTANT ObjectNumber : INT := 5; ObjectLength : INT := 16; END_VAR TYPE UID_STRUCT STRUCT UID : ARRAY[1..8] OF BYTE; Data : ARRAY [1.. (ObjectLength-8)] OF BYTE; END STRUCT; END_TYPE TYPE UidList: ARRAY[1..ObjectNumber] OF UID _STRUCT; END_TYPE VAR CMD : BYTE := 0x69; (* i *) END VAR

*Table 15:**(Forts.)** PIB command description*

Name	Description	
	Parameters	**Description**
	Attributes	This parameter is used to specify the information to be read. Valid values are: 0x00…all UID's are read (without additional information) 0x01 - 0x7F reserved 0x80 - 0xFF manufacturer specific
	Offset-Buffer	This parameter specifies the relative offset within the RXBUF. It specifies the first address within the memory area where the first byte of data read is to be stored.
	Definition of sub-parameters delivered within the response:	
	Parameters	**Description**
	Object Number	This parameter specifies the number of UID's delivered within the acknowledge.
	Object Length	This parameter specifies the number of bytes associated to a single UID (length of UID + additional data). In case of Attributes =0x00 the following applies: ObjectLength = 8.
	UidList	This optional parameter contains a list of manufacturer spe-cific information associated to the UID's currently accessible within the antenna lobe.

Table 16:

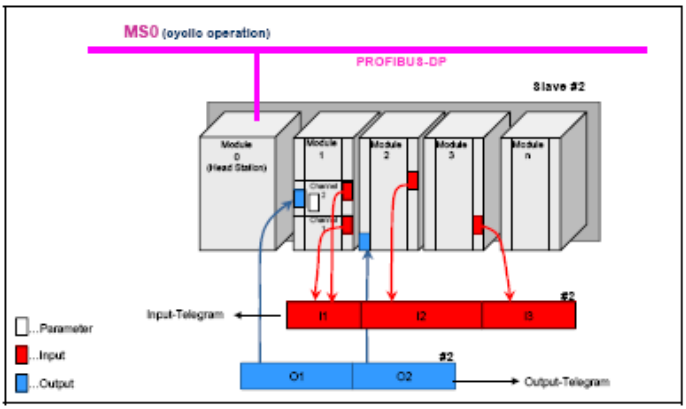
Octet	Coding / Data	Meaning	Notes / Remedy
	Bit 7...0: Channel Number	Channel number within the Module the Status_Message is related to.	
	Bit 7...0: Function_Num	For definition and coding refer to table 11.	Ident profile specific extension
	Bit 7...0: Error_Decode	For definition and coding refer to table 12.	Ident profile specific extension
	Bit 7...0: Error_Code_1	For definition and coding refer to table 13.	Ident profile specific extension
		For definition and coding refer to table 13.	Ident profile specific extension

4.5 Data Access within Field Device

4.5.1 General Device Model

A Profibus device is whether a compact or a modular device (see figure below). In case it is a modular device it is constructed of a head station and a number of modules. It may optionally have binary inputs and outputs. The granularity of devices and modules may vary.

Figure 26: General Device Model



Head station

The head station contains the interface to Profibus and the according parameters. The head station can be redundant. In this case, the redundant head station may occupy a backplane slot different from 0.

Module

A module usually contains a logical or physical combination of channels. The modules are inserted into the slots, or are otherwise stacked. A module may con-tain several channels. Modules are counted continously, starting with the module next to the head station. Modules can exist, which do not contain channels (e.g. power supply). A module is logically identified by a slot number as shown in the figure above. The slot number is used for addressing purpose.

Channel

A channel contains all hard- and software components used to produce a measurement value (input channel) or to create a physical output signal (output chan-nel). Thus, the ident channels represent the Ident Unit. Channels are counted separately for each module, starting with first channel 1. A module may contain up to 8 ident channels.

4.5.2 Block Mapping for Cyclic PROFIBUS-DP Data Transfer

The output values from master to the slave and the input values from the con-nected sensors are transferred as cyclic data. Cyclic data will be transferred via the Data_Exchange service ("...cyclic exchange of the I/O data with the DP-Master (Class 1)..."). The input and output values may be mapped into the cyclic data stream. This mapping is device-global. The principle is shown in figure 30.

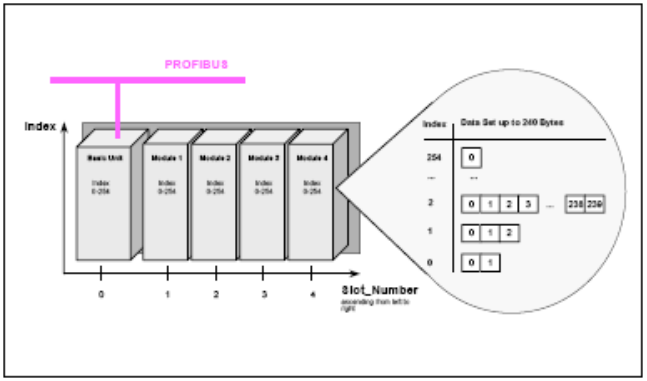
The mapping relies on the modular structure of the GSD [7]. The modules in a device are defined using the Module/EndModule keywords. The assignment to a slot is controlled by the values inside the SlotDefinition/EndSlotDefinition keywords. The assignment of modules to physical slots is done by the configuration device.

4.5.3 Block Mapping for Acyclic PROFIBUS-DP Data Transfer

The acyclic Profibus read and write services are based on slot - index address combinations. Within this profile a mapping is defined between the device structure and the slots. The device block is always assigned to slot number 0 (zero). Every module of a device is mapped to one slot.

Each parameter of a module has to be mapped to an index in order to be accessible by MS1/MS2 services.

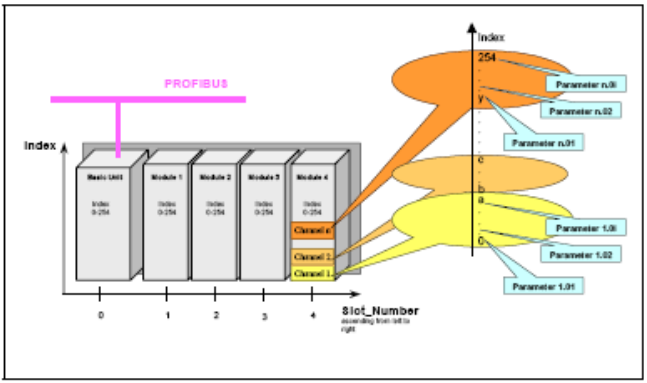
Figure 27: Mapping of input and output values to cyclic communication



In case several channels are assigned to one module, they will share the addressing space of the related slot. All parameters, independent from the channel they are assigned to, have to become addressed by an individual index as illustrated below.

Figure 28:1

Correspondance between parameters and addresses



4.5.4 Ident Channel Definition

This chapter describes the mapping of ident channels to modules. Basically it is possible that up to 8 ident channels may be assigned to an individual module. Module 0 is not used to support an ident channel. For each of the ident channels two indices are used for acyclic communication:

- "Index 10x ("x" representing the channel number)

Indices 101 to 108 are used to parameterize one ident channel. Only the Write_Configcommand is allowed to be transferred through these indices. After the module starts up, a Write-Config must be sent to each ident channel. The channel is not ready for operation until this is done. A Write-Config is also accepted during normal operation. Write-Config interrupts a running command. The user receives no further acknowledgment for the interrupted command.

- Index 11x ("x" representing the channel number)

Indices 111 to 118 are used for sending the actual commands and related acknowledgments (all commands except Write_Config).

The indices of a module are used as follows:

Table 17:

Index	**Usage**
0 ... 100	reserved
101	Parameterization channel 1 (CHL=1)
102	Parameterization channel 2(CHL=2)
103	Parameterization channel 3(CHL=3)
104	Parameterization channel 4(CHL=4)
105	Parameterization channel 5(CHL=5)
106	Parameterization channel 6(CHL=6)
107	Parameterization channel 7(CHL=7)
108	Parameterization channel 8(CHL=8)
109, 110	reserved
111	Data transfer channel 1 (CHL=1)
112	Data transfer channel 2(CHL=2)
113	Data transfer channel 3(CHL=3)
114	Data transfer channel 4(CHL=4)
115	Data transfer channel 5(CHL=5)

Table 17:

Index	Usage
116	Data transfer channel 6(CHL=6)
117	Data transfer channel 7(CHL=7)
118	Data transfer channel 8(CHL=8)
119 ... 200	reserved
201 ... 254	Manufacturer specific
255	I&M functions

5 Identification & Maintenance Functions

The main purpose of the I&M functions is to support the end user during various scenarios of a device's life cycle be it configuration, commissioning, parameterization, diagnosis, repair, firmware update, asset management, audit trailing, and alike. Well-defined uniform parameters and rules should enable the manufacturers to offer devices that behave in a uniform manner and that should enable the end user to act quick and straight. Information provided by the I&M function may be related to the complete device, to an individual module within a device ar a channel associated to a module. Data structures and access mechanisms for device and module related I&M functions are defined in "PROFIBUS Profile Guideline, Part 1, Identification and Maintenance Functions" [5].

Addressing of channel related I&M functions is currently under discussion within TC3 of PNO. The next version of the "PROFIBUS Profile Guideline, Part 1, Identification and Maintenance Functions" will define how to accees these information. As it is the intention of TC3 WG9 not to define a propriatory version of channel related I&M functions this new version of "PROFIBUS Profile Guideline, Part 1, Identification and Maintenance Functions" shall become applicable here.

5.1 PROFILE_ID

The Profile ID's used for the profile "PROFIBUS Proxy Ident Function Blocks acc. to IEC 61131-3" are:

- 5B00H for RFID systems and
- 5B10H for Barcode Reader.

Assignment of Profile ID's is managed by PNO.

5.2 Channel related Information

For each channel it is possible to provide channel related I&M functions. Thus, a device may contain as many channel related information, as channels may be implemented in the individual modules. The scope of this information is restricted to an individual channel. The semantic meaning of the channel related I&M information is the same as of those related to a complete device or module. They are represented by the data types described in [5].

Annex A - Compliance Table

The following table lists the permitted "Implementation dependant features" of the PIB. A manufacturer which claims to be compliant with this PNO specification shall provide a list in the format of this table and shall identify all compliant features of the PIB supported.

Tabelle: 18 Implementation dependant features for the PIB

Clause	Feature	Implementation chosen
3	Maximum memory size supported for the TXBUF	
3	Maximum memory size supported for the RXBUF	
3	Maximum number of commands to be stored within the CMDREF	

The following table lists the communication related features of the Ident Unit relevant to this profile specification. A manufacturer of an Ident Unit which claims to be able to communicate with an instance of a PIB shall provide a list in the format of this table and shall identify all compliant features of the Ident Unit supported.

Tabelle: 19 Compliance table for Ident Units

Clause	Feature	Explanation	implementation specific additional information	compliant (Y/N
3.1.3	Read	Command		
3.1.3	Get	Command		
3.1.3	Physical-Read	Command		
3.1.3	Write	Command		
3.1.3	Put	Command		
3.1.3	Physical-Write	Command		
3.1.3	Format	Command		
3.1.3	Create	Command		
3.1.3	Delete	Command		
3.1.3	Clear	Command		
3.1.3	Update	Command		
3.1.3	Next	Command		
3.1.3	Get-Directory	Command		
3.1.3	Set-Attribute	Command		
3.1.3	Get-Attribute	Command		
3.1.3	Write-Config	Command		
3.1.3	Read-Config	Command		
3.1.3	Mem-Status	Command		
3.1.3	Dev-Status	Command		
3.1.3	Inventory	Command		
3.1.3	Read-BarCode	Command		

Tabelle: 19 Compliance table for Ident Units

Clause	Feature	Explanation	implementation specific additional information	compliant (Y/N
4.2.1	Reading_Gate	Control Bit		
4.2.1	Command repetition	Control Bit		
4.2.2	Target_Presence_Changed	Status Bit		
4.2.2	Target_Present	Status Bit		
	MaxPacketSize	Max. Ident PDU size supported		

Anhang B - Elementary Data Types used within this Specification

This informative annex provides the data type definitions used for this specific profile specification.

Table 20: Data Types

Name	Definition	Reference
BOOL	Boolean (The possible values of variables of this data type shall be 0 and 1, corresponding to the keywords FALSE and TRUE, respectively.)	IEC 61131-3
DWORD	Bit string of length 32	IEC 61131-3
WORD	Bit string of length 16	IEC 61131-3
INT	Integer (The range of values for variables of this data type is from - (216-1) to (216-1)-1.)	IEC 61131-3
ANY		IEC 61131-3
DINT	Double integer (The range of values for variables of this data type is from - (232-1) to (232-1)-1.)	IEC 61131-3
BYTE	Bit string of length 8	IEC 61131-3
ARRAY[1..x] of Data Type		IEC 61131-3

*Table 20:**(Forts.)** Data Types*

Name	Definition	Reference
CMD_ STRUCT	TYPE CMD STRUCT STRUCT CMD : BYTE; Config : BYTE; OffsetBuffer : INT; UID : ARRAY[1..8] OF BYTE; FileName : ARRAY[1..8] OF BYTE; Offset : DINT; Length : INT; StartAddress : DINT; Attributes : BYTE; NextMode : BYTE; Timeout : INT; ObjectNumber : INT; FileType : Word; END STRUCT; END TYPE	
DIRELEMENTS _STRUCT	TYPE DIRELEMENTS_STRUCT STRUCT FileName : ARRAY[1..8] OF BYTE; UsedLength : DINT; Attributes : BYTE; FileLength : DINT; FileType : WORD; END STRUCT; END_TYPE	

*Table 20:**(Forts.)** Data Types*

Name	Definition	Reference
DIRLIST_ STRUCT	Whereas this structure is defined in ST notation it is not intended to become copied into a ST program as it contains an array of dynamic length, witch is not conformant to IEC 61131-3 (ST). St has only been chosen for consistency reason within the document. TYPE DIRLIST_ STRUCT STRUCT UID1 : ARRAY[1..8] OF BYTE; TagName : ARRAY[1..8] OF BYTE; FreeUserMem : DINT; Checksum : WORD; FileCount : INT; FileList : ARRAY[1..FileCount] of DIRELEMENTS_STRUCT; END STRUCT; END TYPE	

*Table 20:**(Forts.)** Data Types*

Name	Definition	Reference
UID_STRUCT	Whereas this structure is defined in ST notation it is not intended to become copied into a ST program as it contains an array of dynamic length, witch is not conformant to IEC 61131-3 (ST). St has only been chosen for consistency reason within the document. TYPE ObjectLength : INT; END_TYPE TYPE UID STRUCT STRUCT UID : ARRAY[1..8] OF BYTE; Data : ARRAY[1..(ObjectLength-8)] OF BYTE; END_STRUCT; END_TYPE	
UidList	Whereas this structure is defined in ST notation it is not intended to become copied into a ST program as it contains an array of dynamic length, witch is not conformant to IEC 61131-3 (ST). St has only been chosen for consistency reason within the document. TYPE ObjectNumber : INT; END TYPE TYPE UldList: ARRAY[1..ObjectNumber] OF UID_STRUCT; END TYPE	

The encoding of UID is defined within Technical Report ISO/IEC /TR 15963, Automatic identification - Radio Frequency Identification for

item management - Unique identification for RF tags, Annex A as follows:

AC (Allocation Class)	UID issuer Registration Number	Serial number
	Size defined by AC value	Size defined by AC value

Table 22:

Class	**UID issuer identifier size**	**Serial Number size.**	**Registration authority (of "UID issuer Registration Number")**
7816-6	8 Bit	48 Bit	APACS (ISO/ IEC 7816-6 registration authority)
14816	perNEN	perNEN	NEN (ISO 14816 registration authority)
EAN.UCC	per EAN.UCC	per EAN.UCC	EAN.UCC
INCITS 256	per ANS INCITS 256	per ANS INCITS 256	ANSI ASC INCITS T6
RFU	N/A	N/A	Reserved for future use by ISO

4 Glossary

A

Acknowledge

Acknowledgement of a signal received.

Active metal component

Conductor or conductive component that is under voltage during operation.

Address

Number for identifying e.g. a memory position, a system or a module within a network.

Addressing

Allocation or setting of an address, e.g. for a module within a network.

Analogue

Continuously variable value, e.g. of a voltage. The value of an analogue signal can take on any value within a specific range.

ARP

Worldwide protocol regulating the assignment of hardware addresses (MAC IDs) to the IP addresses of network stations from internal tables.

Automation device

A device with inputs and outputs for controlling a connected technical process. Programmable logic controllers (PLC) are a special group of automation devices.

B

Baud rate

See "Baud"

Bi-directional

Working in both directions.

Bus

Group signal line for data exchange, e.g. between CPU, memory and I/O levels. A bus can be composed of a number of parallel cables for data transmission, addressing, control and power supply.

Bus cycle time

Time interval required for a master to serve all slaves or stations in a bus system, i.e. reading inputs and writing outputs.

Bus line

The smallest unit connected to a bus, composed of a PLC, a coupling element for modules on the bus and a module.

Bus system

The totality of all units that communicate with one another via a bus.

C

Capacitive coupling

A capacitive (electrical) coupling occurs between cables with different potentials. Typical sources of interference are e.g. parallel routed signal cables, contactors and static discharges.

Coding element

Two part element for the unambiguous assignment of electronic and base module.

Command capable module

Command capable modules are modules with an internal memory set that are capable of carrying out specific commands (e.g. the output of substitute values).

Configuration

Systematic arrangement of the I/O modules of a station.

CPU

English acronym for "Central Processing Unit". Central unit for data processing, the core of a computer.

CSA

(Chip Sharing Approach): DOT provides different companies within the supply chain the possibility of using the chip for storing their data. It is possible to allocate read and right access for the chip.

D

DHCP

A client-server protocol that simplifies the allocation of IP addresses and other parameters. Used for dynamic and automatic end device configuration.

Digital

A value, e.g. a voltage, that can only take on a certain status within a finite set, generally defined as either 0 or 1.

DIN

German acronym for German Industrial Standard.

DOT

(Data on Tag): a range of data stored on a chip enabling independent decisions to be made at a local level without having to refer to a central data base.

DP master class 1

The automation system (PLC) which is mainly responsible for the cyclical data processing. The „DPV1“ functions can be employed additionally / optionally. (also DPM1/DPC1) .

DP master class 2

Exclusive transmission of acyclical demand data. This data transmission can e.g. be carried out using an engineering tool (PC application program).

DPV1

Functional extension to the Profibus-DP. In addition to the cyclical process data, demand data can be transmitted via acyclical communication functions. The acyclical services are conducted temporally parallel and in addition to the cyclical process data transmission with lower priority.

E

EIA

English acronym for "Electronic Industries Association". An association of companies from the electronics industry in the USA.

Electrical components

All objects that are employed for the production, conversion, transmission and utilization of electrical power, e.g. conductors, cables, machines and control devices.

EMC

Acronym for "Electromagnetic Compatibility". Refers to the ability of an electrical component to operate error-free in a particular environment and without having a negative influence on that environment.

EPC

(Electronic Product Code)
The actual data on an article/product is stored in one or a number of data bases maintained by the manufacturer, shipper, retailer etc.

Equipotential bonding

The equalisation of the electrical levels of the bodies of electrical components and external, conductive bodies by means of an electrical connection.

ESD

English acronym for „Electro Static Discharge“.

F

Field bus

Data network on the sensor/actuator level. A field bus connects the devices at the field level. Characteristic of a field bus is its high transmission security and real-time behaviour.

Field power supply

Power supply for field devices as well as the signal voltage.

Force Mode

Software mode enabling an installation to be set to a specific condition by the forced setting of certain variables on the input and output modules.

Function Code

Is incorporated into the data frame by the Modbus. Contains amongst other things commands for the reading and writing of input and output data.

G

Galvanic coupling

A galvanic coupling generally occurs when two electrical circuits have a common conductor. Typical sources of interference are e.g. starting motors, static discharges, clocked devices and a difference in potential between the component housings and the common power supply.

GND

English acronym for "GROUND" (potential 0)

Ground

The term employed in electrical engineering to describe an area whose electrical potential at any point is equal to zero. The electrical potential in the area of grounding devices may not necessarily be zero, in such cases one refers to ground reference.

Grounding

The connection of an electrically conductive component to the ground using a grounding setup.

Grounding strip

Flexible conductor, generally plaited, that connects the inactive parts of a component e.g. the door of a switch cabinet to the body of the switch cabinet.

Ground connection

One or more components that have direct and good contact to the earth.

Ground reference

Potential of the ground in the area of grounding devices. Unlike "earth" which always has a potential of zero it can have a potential other than that of zero.

H

Hexadecimal

Numerical system with a base of 16. The sequence begins with 0 to 9 and continues with the letters A, B, C, D, E and F.

Hysteresis

A sensor can stall at a particular point and then "oscillate" around this position. This condition leads to a fluctuation of the counter value around a particular value. If a reference value is within this fluctuation range, the respective output is switched on and off in rhythm with the signal fluctuation.

I

Impedance

Impedance that a component or a circuit composed of a number of components displays for an alternating current with a specific frequency.

Inactive metal components

Unexposed conductive element that is electrically isolated from the active metal components. However can carry a voltage in the event of a fault.

Inductive coupling

An inductive (magnetic coupling) occurs between two current carrying cables. The magnetic effect generated by the electrical currents induces an interference voltage. Typical sources of interference are e.g. motors, parallel routed power cables, and HF signal cables.

I/O

English acronym for "Input/Output".

IP protocol

Acronym for Internet Protocol, a protocol for the packet oriented and connectionless transport of data packets from a sender across a number of networks to a receiver.

L

Lightening protection

All measures that serve to protect a system from damage due to overvoltages caused by lightening strikes.

Low impedance connection

Connection with a low AC impedance.

LSB

English abbreviation for „Least Significant Bit“. Bit with the lowest significance.

M

MAC-ID

A manufacturer specific ID for the unambiguous identification of a node in a network, allocated according to a specific code.

Mass

Totality of all connected inactive parts of a component that do not have a touch voltage, even in the event of a fault.

Modbus TCP

The Modbus protocol as part of the TCP/IP protocol.

Modbus communicates by means of function codes which are integrated into the data frame. Modbus TCP employs the Transport Control Protocol (TCP) for the transmission of the Modbus application protocol in Ethernet TCP/IP networks.

Mode

English term for type of operation.

Module bus

The module bus is the internal bus of a BL-67 station. The BL-67 modules communicate with the gateway via this bus. It is independent of the field bus.

MSB

English abbreviation for „Most Significant Bit“. Bit with the highest significance.

Multitag capability

Simultaneous, unambiguous recognition of multiple tags directed past an aerial.

O

Overhead

The system administration time required for each transmission cycle.

P

Parameterisation

The setting of parameters for the individual bus stations, i.e. their modules, within the configuration software of the DP master.

PIB

Abbreviation for „Proxy Ident Block“. This function block represents an Ident system in the control. This constitutes a uniform program interface for the actual application.

Ping

Implementation of an echo protocol, used in order to test the accessibility of target stations.

PLC

Acronym for Programmable Logic Controller.

Potential free

Galvanic isolation of the reference potentials of control and load current circuits in I/O modules.

Potential linked

Galvanic connection of the reference potentials of control and load current circuits in I/O modules.

Protective conductor

A conductor used to protect against dangerous shock currents, represented by the term PE (English acronym for "Protective Earth").

R

Reaction time

The time interval within a bus system between transmitting a read request and receiving an answer. The time interval within an input module between the change in the signal at the input and the output of the same signal to the bus system.

Reference potential

Potential from which the voltages of all connected circuits are viewed and/or measured.

Repeater

Amplifier for the signals transmitted via a bus.

RFID

Radio frequency identification

RFID technology

This technology enables a contactless transmission of data. The data is transmitted by means of radio frequency technology. A „Transponder” Page 4-9 is employed as a data storage medium.

RS 485

Serial interface according to the EIA standard for the rapid data transmission via multiple transmitters.

S

Serial

Term for a type of information transmission where the data is transmitted sequentially - bit for bit - via a cable.

Shield

Term describing the conductive casing of cables, housings and cabinets.

Shielding

Totality of the measures and components employed to connect the installation elements to the shield.

Short circuit proof

A characteristic of electronic components. A short circuit proof component is able to withstand the thermic and dynamic loads that can occur at the point of installation due to a short circuit.

Station

A functional unit or I/O component composed of a number of elements.

T

Terminating resistance

Resistor at both ends of a bus cable which serves to prevent interference from signal reflections and provide bus cable matching. Terminating resistors must always be the last units at the end of a bus segment.

TCP

English acronym for "Transmission Control Protocol", connection oriented transport protocol built onto the Internet protocol. Specific error recognition mechanisms (e.g. acknowledgement of frames, time monitoring of frames) guarantee a secure and error free data transport.

Topology

Geometrical structure of a network or the arrangement of circuits.

Transponder

(**Trans**mitter + Res**ponder**)
Transponder. A transponder consists of a microchip (with an unambiguous identification number), a transmission / receiving aerial and a housing. Data is transmitted between a reader device and the transponder via electromagnetic waves.

Transponder technology

(also „RFID technology” Page 4-8)

U

UDP

English acronym for "User Datagram Protocol". UDP is a transport protocol for the connectionless data exchange between Ethernet stations.

UID

English acronym for "Unique Identifier" A UID is an unambiguous serial number for a transponder. As an address it indicates the data belonging to the transponder i.e. the tagged product. This data can e.g. be stored in a data base.

TURCK

Industrial
Automation

TURCK WORLD-WIDE HEADQUARTERS

GERMANY
Hans Turck GmbH & Co. KG
Witzlebenstraße 7, Germany
45472 Mülheim an der Ruhr
P. O. Box 45466 Mülheim an der Ruhr
Phone +49 (0) 208 4952-0
Fax +49 (0) 208 4952-2 64
E-Mail turckmh@turck.com

www.turck.com

D101579 0406