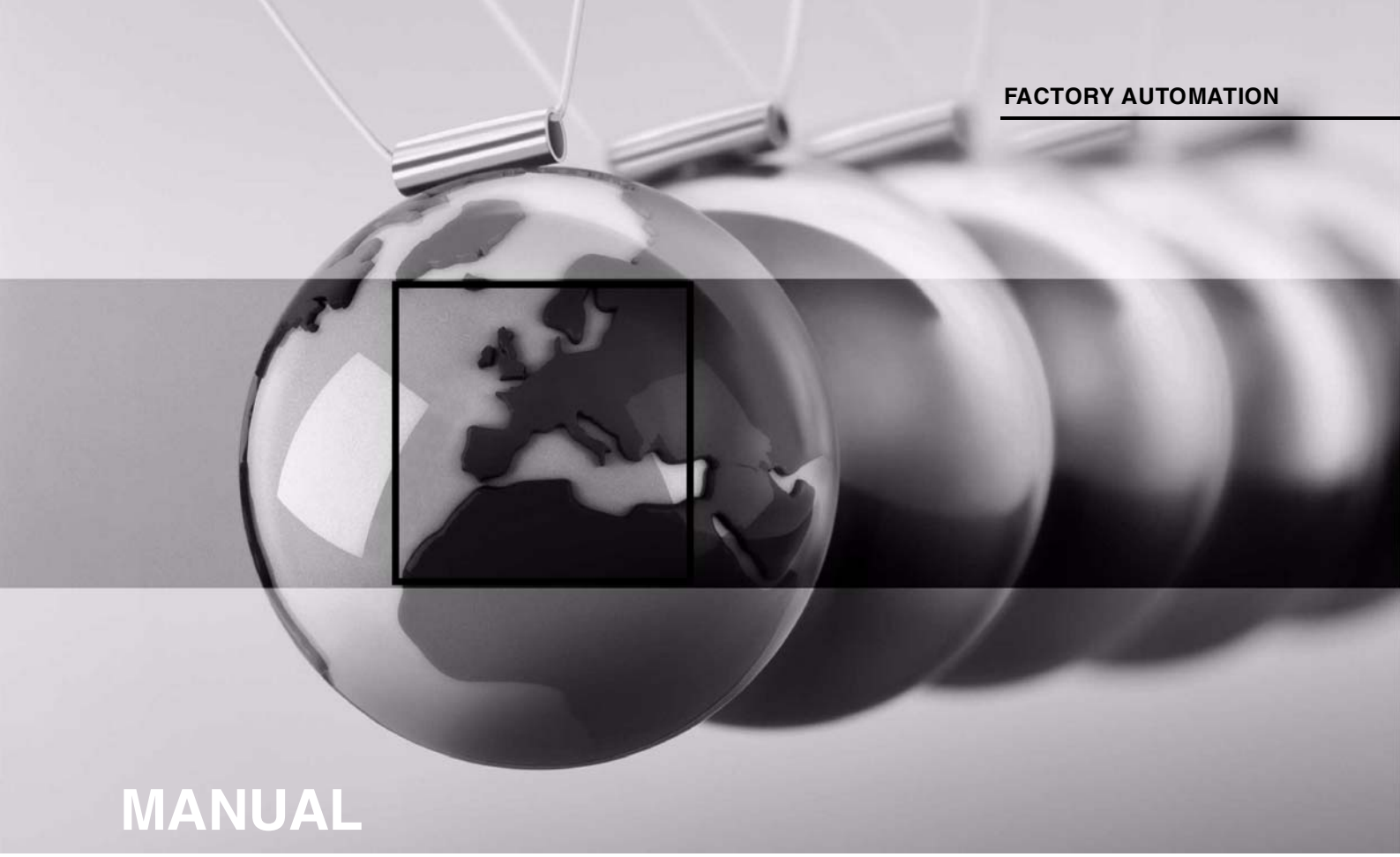


MANUAL

IC-KP2-2HB17-2V1D

IDENTControl Compact unit with Ethernet interface

PEPPERL+FUCHS
SENSING YOUR NEEDS

With regard to the supply of products, the current issue of the following document is applicable: The General Terms of Delivery for Products and Services of the Electrical Industry, published by the Central Association of the Electrical Industry (Zentralverband Elektrotechnik und Elektroindustrie (ZVEI) e.V.) in its most recent version as well as the supplementary clause: "Expanded reservation of proprietorship"

1 Introduction

Congratulations

You have chosen a device manufactured by Pepperl+Fuchs. Pepperl+Fuchs develops, produces and distributes electronic sensors and interface modules for the market of automation technology on a worldwide scale.

Before you install this device and put it into operation, please read the operating instructions thoroughly. The instructions and notes contained in this operating manual will guide you step-by-step through the installation and commissioning procedures to ensure trouble-free use of this product. By doing so, you:

- guarantee safe operation of the device
- can utilize the entire range of device functions
- avoid faulty operation and the associated errors
- reduce costs from downtime and incidental repairs
- increase the effectiveness and operating efficiency of your plant.

Store this operating manual somewhere safe in order to have it available for future work on the device.

After opening the packaging, please ensure that the device is intact and that the package is complete.

Symbols used

The following symbols are used in this manual:

Note!

This symbol draws your attention to important information.

Handling instructions

You will find handling instructions beside this symbol

Contact

If you have any questions about the device, its functions, or accessories, please contact us at:

Pepperl+Fuchs GmbH
Lilienthalstraße 200
68307 Mannheim
Telephone: +49 621 776-4411
Fax: +49 621 776-274411
E-Mail: fa-info@pepperl-fuchs.com

2014-02

2 Declaration of conformity

2.1 CE conformity

This product was developed and manufactured under observance of the applicable European standards and guidelines.

Note!

A declaration of conformity can be requested from the manufacturer.

3 Safety

3.1 Symbols relevant to safety

Danger!

This symbol indicates an imminent danger.

Non-observance will result in personal injury or death.

Warning!

This symbol indicates a possible fault or danger.

Non-observance may cause personal injury or serious property damage.

Caution!

This symbol indicates a possible fault.

Non-observance could interrupt devices and any connected facilities or systems, or result in their complete failure.

3.2 Intended use

The IDENTControl Compact IC-KP2-2HB17-2V1D is a control unit with integral Ethernet interface designed for identification systems. The IDENTControl Compact can be used as a control cabinet module or for field applications. You can connect suitable inductive read/write heads, microwave antennas, or trigger sensors to the IDENTControl Compact. However, wiring suitable for the system design must always be used.

Read through these instructions thoroughly. Familiarize yourself with the device before installing, mounting, or operating.

Always operate the device as described in these instructions to ensure that the device and connected systems function correctly. The protection of operating personnel and plant is only guaranteed if the device is operated in accordance with its intended use.

3.3 General notes on safety

Only instructed specialist staff may operate the device in accordance with the operating manual.

User modification and or repair are dangerous and will void the warranty and exclude the manufacturer from any liability. If serious faults occur, stop using the device. Secure the device against inadvertent operation. In the event of repairs, return the device to your local Pepperl+Fuchs representative or sales office.

The connection of the device and maintenance work when live may only be carried out by a qualified electrical specialist.

The operating company bears responsibility for observing locally applicable safety regulations.

Store the not used device in the original packaging. This offers the device optimal protection against impact and moisture.

Ensure that the ambient conditions comply with regulations.

Note!

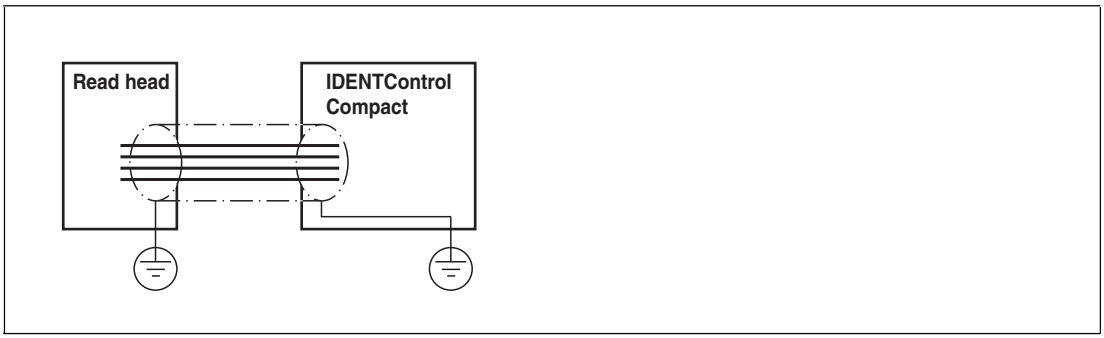
Disposal

Electronic waste is hazardous waste. When disposing of the equipment, observe the current statutory requirements in the respective country of use, as well as local regulations.

2014-02

3.4 Contact protection

Our housings are manufactured using components made partly or completely from metal to improve noise immunity.



Danger!

Electric shock

The metallic housing components are connected to ground to protect against dangerous voltages that may occur in the event of a fault in the SELV power supply!

See chapter 5.4.3

4 Product Description

4.1 Product family

The brand name, IDENTControl, represents a complete identification system. The system consists of an IDENTControl Compact unit with bus interface, inductive R/W heads (125 kHz and 13.56 MHz), R/W heads with electromagnetic coupling (UHF with 868 MHz) and accompanying code, and read/write tag in many different designs. The IDENTControl Compact can be connected to other identification systems.

The system is equally well suited for use in the switching cabinet and for field use in IP67. The interface to the controlling fieldbus is integrated into the enclosure and all connections are implemented as plugs. This enables simple installation and quick, correct replacement in case of device failure. The consistent EMC design (metal enclosure, grounding, shielded wires) offers a high degree of noise immunity.

4.1.1 R/W heads

There are different R/W heads available for the IDENTControl Compact in different designs. You can connect inductive R/W heads (125 kHz and 13.56 MHz) or R/W heads with electromagnetic coupling (UHF with 868 MHz) depending on your particular application.

4.1.2 Read Only Tags/Read/Write Tags

Read only / read/write tag 125 kHz (inductive)

A wide range of read only and read/write tag designs are available for this frequency range, from a 3 mm thin glass tube to a transponder 50 mm in diameter. Read/write tags are available for temperatures up to 300 °C (max. 5 min) in chemical-resistant housings for installation in metal and in degree of protection IP68/IP69K. IPC02-... read only tags offer 40-bit read only codes. IPC03-... read/write tags have a 928-bit freely programmable memory bank and an unmodifiable 32-bit read only code. You can define 40-bit read only codes with IPC11-... read only tags. You can use these as permanent read only codes or continually redefine them.

Read/write tag 13.56 MHz (inductive)

Read/write tags in this frequency range save larger quantities of data and offer a considerably higher reading speed than read/write tags of the 125 kHz system. IQH-* and IQH1-* read/write heads from Pepperl+Fuchs are compatible with most existing read/write tags that comply with standard ISO 15693. With the IQH2-* read/write heads you can use read/write tags that comply with standard ISO 14443A.

The 13.56 MHz technology even allows smart labels (read/write tags in the form of adhesive labels with printed barcode). Currently available read/write tags have a memory capacity of 64 bits of read only code and a maximum 2 KB of programmable memory.

Data carrier 868 MHz (UHF)

Data carriers in this frequency range can be passive as well as active (with battery) and use a specially-shaped rod antenna as the resonance element. The passive transponders can be produced very cheaply and have a range of several meters.

As material handling and the automotive sector requires ranges of 1 to 5 meters, this system represents a low-cost alternative to microwave systems, particularly because of its low transponder costs. The high carrier frequency supports large data volumes and extremely short read times.

4.1.3 Handhelds

There are various handheld read/write devices available for controlling processes (write/read functions, initialization of data carriers).

Figure 4.1

Handheld	Frequency range
IPT-HH20	125 kHz
IST-HH20	250 kHz
IQT1-HH20	13.56 MHz
IC-HH20-V1	depending on the read/write head

4.2 Connection accessories

4.2.1 Connection cable for R/W heads and trigger sensors

Compatible connection cables with shielding are available for connecting the R/W heads and trigger sensors.

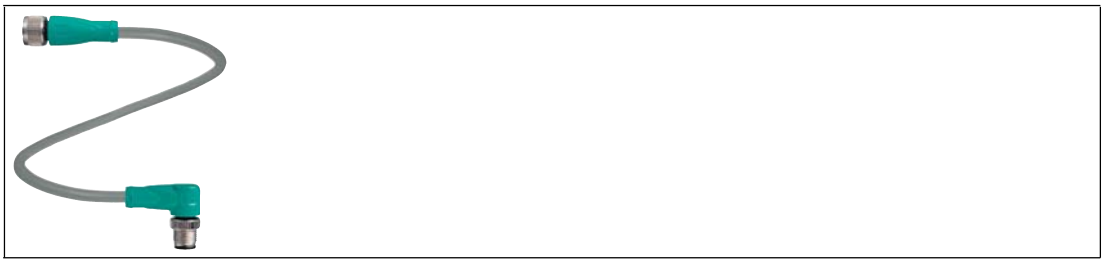


Figure 4.2

Accessories	Description
2 m long (straight female, angled male)	V1-G-2M-PUR-ABG-V1-W
5 m long (straight female, angled male)	V1-G-5M-PUR-ABG-V1-W
10 m long (straight female, angled male)	V1-G-10M-PUR-ABG-V1-W
20 m long (straight female, angled male)	V1-G-20M-PUR-ABG-V1-W
Field attachable female connector, straight, shielded	V1-G-ABG-PG9
Field attachable male connector, straight, shielded	V1S-G-ABG-PG9
Field attachable female connector, angled, shielded	V1-W-ABG-PG9
Field attachable male connector, angled, shielded	V1S-W-ABG-PG9
Dummy plug M12x1	VAZ-V1-B3

2014-02

4.2.2 Cable connectors for the power supply

Compatible M12 sockets with an open cable end for connecting the IDENTControl Compact to a power supply are available in different lengths.

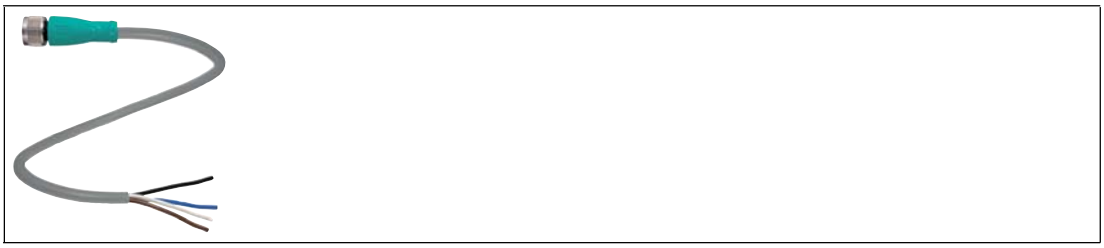


Figure 4.3

Accessories	Model number
Length 2 m (straight socket)	V1-G-2M-PUR
Length 5 m (straight socket)	V1-G-5M-PUR
Length 10 m (straight socket)	V1-G-10M-PUR

4.2.3 Connection cable to the Ethernet interface

The IDENTControl Compact has a D-coded, M12 socket and is connected to the network using a suitable cable.

Figure 4.4

Accessories	Designation
5 m connection cable	V1SD-G-5M-PUR-ABG-V45-G

4.2.4 Mounting aid

An aid for mounting the IDENTControl Compact to a DIN mounting rail is available.

Accessories	Model number
Mounting aid	ICZ-MH05-SACB-8

4.3 Delivery package

The delivery package contains:

- 1 IDENTControl Compact unit
- 1 Quick start guide
- 2 grounding screws
- 2 serrated lock washers
- 2 crimp connectors

4.4 Range of application

The system is suited for the following applications:

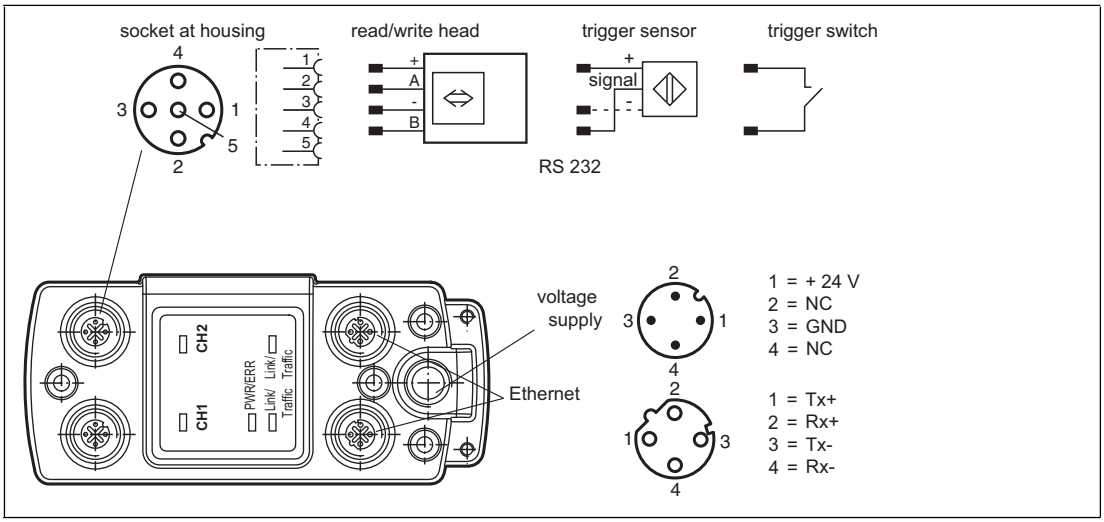
- Automation
- Material flow control in production
- Acquisition of operating data
- Access control
- Identification of storage vessels, pallets, work piece carriers, refuse containers, tanks, containers, etc.

4.5 Device characteristics

- Up to 2 R/W heads can be connected
- Alternatively, 1 R/W head and 1 trigger sensor can be connected
- LED status indicators for bus communication and R/W heads

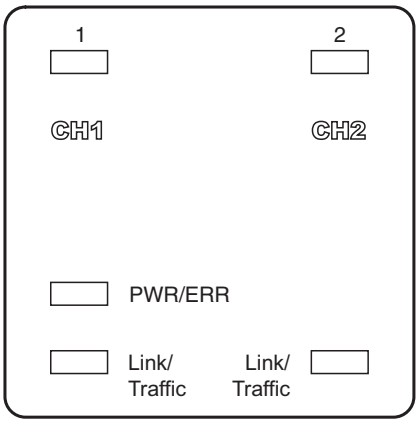
4.6 Interfaces and connections

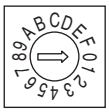
The control unit IC-KP2-2HB17-2V1D has the following interfaces and connections:



PEPPERL+FUCHS

4.7 Displays and Controls





Address configuration

Address state

LEDs

Description	Function	Status description
1 2	Status display for the read/write heads	LED illuminates **green** when there is an active command on the read/write head. LED illuminates **yellow** for approx. 1 second when a command is executed successfully.
CH1 CH2	Displays the read/write heads connected (channel)	LED illuminates **green** when a read/write head is connected to channel 1 or channel 2. LED illuminates **red** if a configuration error occurs.
PWR/ERR	Status display for the IDENTControl Compact	LED illuminates **green** when the IDENTControl Compact is connected to a power supply and the interface is ready for operation. LED illuminates **red** if there is a hardware error or a PROFINET name has been assigned and no PROFINET connection has been established. LED illuminates **yellow** while the IDENTControl Compact starts up. The start-up process lasts approximately 20 seconds LED flashes **green** if a signal is sent to the IDENTControl Compact via the "Flashing" PROFINET function or if there is an internal data overflow.
Link/Traffic	Connection/network activity for the relevant channel (1/2)	LED is off until initial communication is made via Ethernet. LED illuminates **green** when a connection to the network is established. LED flashes **yellow** at the same speed as the data being sent.
ADDR STATE	Status display for address setting	LED flashes **green** when the IDENTControl Compact is ready for the address to be entered using the rotary switch. LED illuminates **green** when one digit of the address is entered successfully using the rotary switch.

Operating controls

Description	Function	Configuration option
Rotary switch	Device address setting	0 ... F

5 Installation

5.1 Unpacking

Check the product for damage while unpacking. In the event of damage to the product, inform the post office or parcel service and notify the supplier.

Check the package contents with your purchase order and the shipping documents for:

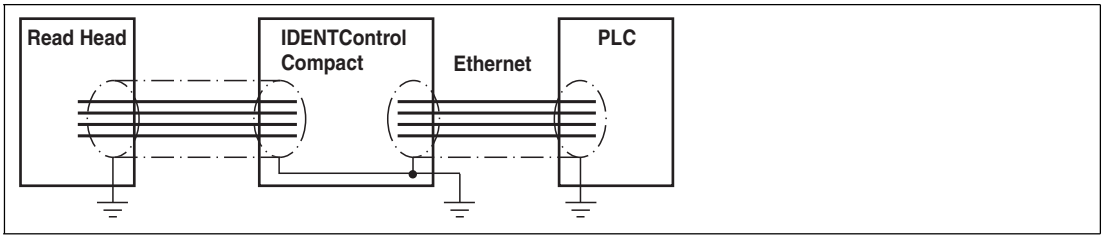
- Delivery quantity
- Device type and version in accordance with the type plate
- Accessories
- Quick start guide

Retain the original packaging in case you have to store or ship the device again at a later date.

Should you have any questions, please contact Pepperl+Fuchs.

5.2 EMC concept

The outstanding noise immunity of the IDENTControl Compact against emission and immission is based on its consistent shielding design which uses the principle of the Faraday cage. Interference is caught in the shield and safely diverted via the ground connections.



The cable shielding is used to discharge electromagnetic interference. When shielding a cable, you must connect both sides of the shield to ground with low resistance and low inductance.

Note!

If cables with double shields are used, e.g. wire mesh and metalized foil, the both shields must be connected together, with low resistance, at the ends when making up the cable.

Power supply cables are the source of much interference, e.g. from the supply lines of 3-phase electric motors. For this reason, the parallel laying of power supply cables with data and signal cables should be avoided, particularly in the same cable duct.

The metal enclosure of the IDENTControl Compact and the metal enclosure of the R/W heads complete the continuous shielding concept.

You must establish a low resistance and low inductance connection between the shields and ground so that the shielding is not interrupted through the metal enclosure. The complete electronics system and all routed cables are located within a Faraday cage.

Caution!

Electromagnetic interference

Device malfunction caused by EMC effects.

Use cables with continuous shield.

5.3 Installation

If you would like to the mount the IDENTControl Compact to a DIN rail, we recommend using mounting accessory ICZ-MH05-SACB-8.

Note!

The rotary switch used to select the device address is located on the underside of the IDENTControl Compact. This rotary switch is no longer accessible once the IDENTControl Compact is installed.

Set the rotary switch before mounting the IDENTControl Compact (see chapter 6.4).

5.4 Device connection

Electrical connection using plug connectors makes installation simple.

5.4.1 Power Supply

Connect the power supply using an M12 connector. A plug with the following pin assignment is located on the enclosure:

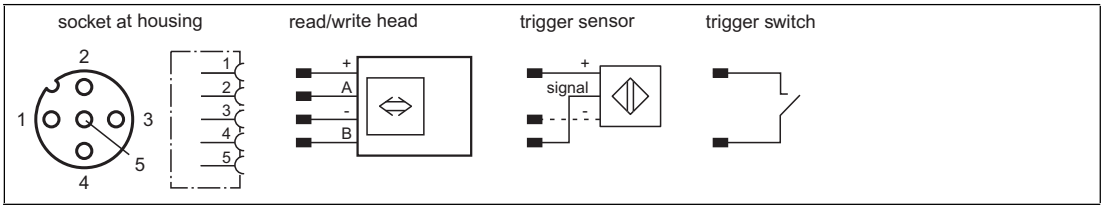
1 + 24 V
2 NC
3 GND
4 NC

Compatible connecting cable see chapter 4.2.2.

5.4.2 Read/Write Head and Trigger Sensors

You can connect a maximum of two read/write heads (125 kHz or 13.56 MHz) or read/write heads with electromagnetic coupling (UHF with 868 MHz) to the IDENTControl Compact.

You can connect a trigger sensor, instead of a read/write head, to sockets 1 and 2. You can assign the trigger sensor to a read/write head. The trigger sensor must be PNP.



For details of compatible read/write heads, see chapter 4.1.1 and of compatible connecting cables, see chapter 4.2.1.

Connecting read/write heads

Connect the read/write heads or the trigger sensor with compatible connecting cable to the top of the housing via the M12 connector.

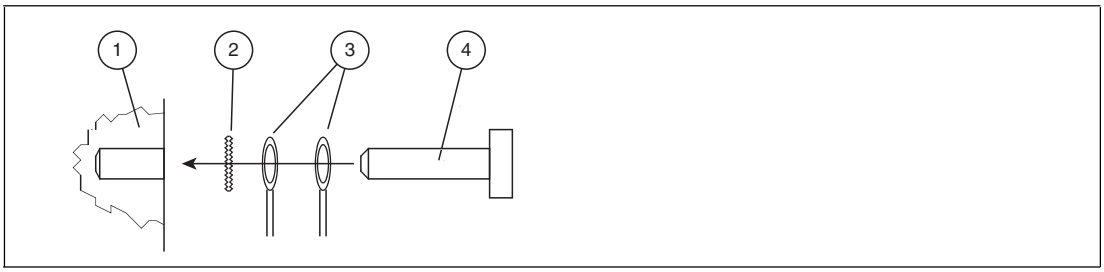
5.4.3 Ground connection

Connect the IDENTControl Compact unit to ground via a screw on the right under the housing.

Note!

In order to guarantee safe grounding, mount the serrated washer between the crimp connector and the housing.

Use a ground conductor lead with a cross-section of at least 4 mm^2.



1 Housing
2 Serrated lock washer
3 Crimp connector
4 Lock screw

Connecting the IDENTControl Compact to ground

Screw the ground conductor to the housing with a crimp connector.

5.4.4 Ethernet connection guide

Use the D-coded M12 socket and the V1SD-G-5M-PUR-ABG-V45-G cable to connect the IDENTControl Compact unit to an Ethernet network.

1 Tx+
2 Rx+
3 Tx-
4 Rx-

Caution!

Electromagnetic interference

Device malfunction caused by EMC effects.

The M12 socket is connected galvanically to the grounded housing. The EtherNet/IP specification recommends not using cables with a shield connected to the plug at both ends. However, we recommend using cables with a continuous shield to avoid malfunctions.

2014-02

6 Commissioning

6.1 Preliminary considerations

Caution!
Uncontrolled triggered processes

Before commissioning the device, make sure that all processes are running smoothly; otherwise damage may occur in the plant.

This manual contains important information required to operate the IDENTControl Compact unit with Ethernet interface. Due to the wide variety of programming options in an Ethernet network, we are unable to include examples relating to commissioning in this manual.

One important aspect of the operation of an extended identification system on the Ethernet is the time response of the overall system. The answer to the question "How long after the positioning of a tag in front of a read/write head will the read data be available in the computer or PLC?" depends on many different factors.

The most important factors that determine the response time are:

- Nature of the higher-level host system, e.g. PLC or PC
- Communication between the client and server.
- Network utilization.
- Number and nature of connected read/write heads.
- Code / tag types used.
- Nature of access to the communication objects of the read/write head.
- Nature of the commands to the read/write head.
- Structure of the user program.

If you are planning larger projects or gaining basic experience in programming an Ethernet-based system, we recommend constructing a model of your application before installing the system in the plant. Use this model to test the process of data transfer to the identification system.

6.2 Connection

Warning!
Incorrect electrical connection

Incorrect connections may damage the system.

Before commissioning, familiarize yourself with the system of communication between the Ethernet controller and the read/write station. Check all connections before commissioning.

When the supply voltage is connected and the device is initialized, the "PWR/ERR" LED lights up green. The LED flashes green during initialization. The LED lights up red if a device fault occurs.a Profinet name is not assigned or there is no Profinet connection

6.3 Device Settings

Warning!
Device not configured or configured incorrectly

Configure the device prior to commissioning. A device that has not been configured or configured incorrectly may lead to faults in the plant.

You must set the various parameters prior to commissioning.

The parameters are volatile and non-volatile parameters. Volatile parameters are reset to their default setting when the system is switched off and on again.

Non-volatile parameters

Parameter	Default setting	Value range
General		
Multiplex mode	Off	On / off
Read/write head		
Trigger mode	Off	On / off
Tag type	99	00 ... 99
Ethernet interface		
MAC address	00:0D:81:xx:xx:xx	00:0D:81:xx:xx:xx
DHCP	Off	On / off
IP address	169.254.10.12	yyy.yyy.yyy.yyy
Standard gateway	169.254.254.1	yyy.yyy.yyy.yyy
Subnet mask	255.255.0.0	yyy.yyy.yyy.yyy
Assembly inst. Out	100d	100d ... 112d
Data hold time	50d x 10 ms	0d ... 255d x 10 ms

Volatile parameters

Parameter	Default setting	Value range
R/W head		
Password mode	Off	on / off
Password	00000000	00000000 ... FFFFFFFF

Configure the R/W system using the system commands described (see chapter 7.11). "99" is preset as the tag type.

6.4 Setting the IP address

The IP address of the IDENTControl Compact is preset to 169.254.10.12. The way in which the IP address is modified depends on if you are using a DHCP server.

If you are **not using a DHCP server**, the IP address is manually assigned: Either select the IP address using the rotary switch **or** start the device with the preset IP address and specify the new IP address on the device website.

If you are **using a DHCP server**, the server assigns the IP address to the IDENTControl Compact.

Note!

We recommend using a fixed preset IP address in order to avoid system malfunctions.

Note!

The rotary switch used to select the device address is located on the underside of the IDENTControl Compact. This rotary switch is no longer accessible once the IDENTControl Compact is installed.

2014-02

Setting the IP address using the rotary switch

1. Select an IP address. Insert leading zeros so that all the numbers have three digits and then remove the periods. Example: Write the IP address 172.16.10.15 like this: 172016010015.
2. Set the rotary switch to position "F".
3. Restart the IDENTControl Compact by resetting the power supply.

   ↳ The "ADDR STATE" LED will flash green.
4. Select the first digit of the IP address using the rotary switch and wait until the "ADDR STATE" LED permanently lights up.
5. Select the next digit using the rotary switch and wait until the "ADDR STATE" LED permanently lights up.
6. Repeat this procedure until you have entered all the digits of the IP address.
7. Set the rotary switch to position "0".
8. Configure other network settings in the **Network** window on the device website (see chapter 7.5.2).

Entering the IP address on the website

The IP address of the PC connected to the IDENTControl Compact must come from the address area 169.254.X.X.

1. Set the rotary switch to position "A".
2. Connect the IDENTControl Compact to a PC via the Ethernet interface.
3. Restart the IDENTControl Compact by resetting the power supply.
4. Open the IDENTControl Compact website by entering the preset IP address (169.254.10.12) in a browser.
5. Open the **Network** window. Enter the user name and password (default on delivery: "identcontrol"). Enter the new IP address and configure other settings as required.
6. Set the rotary switch to position "0".
7. Click **save**.

   ↳ The IDENTControl Compact restarts with the new IP address.

For a description of other web functions see chapter 7.5.

Obtaining an IP address from the DHCP server

Set the rotary switch to "D" and start the device.

↳ The IDENTControl Compact retrieves the network settings from the DHCP server during startup: IP address, subnet mask and gateway address.

7 Commands

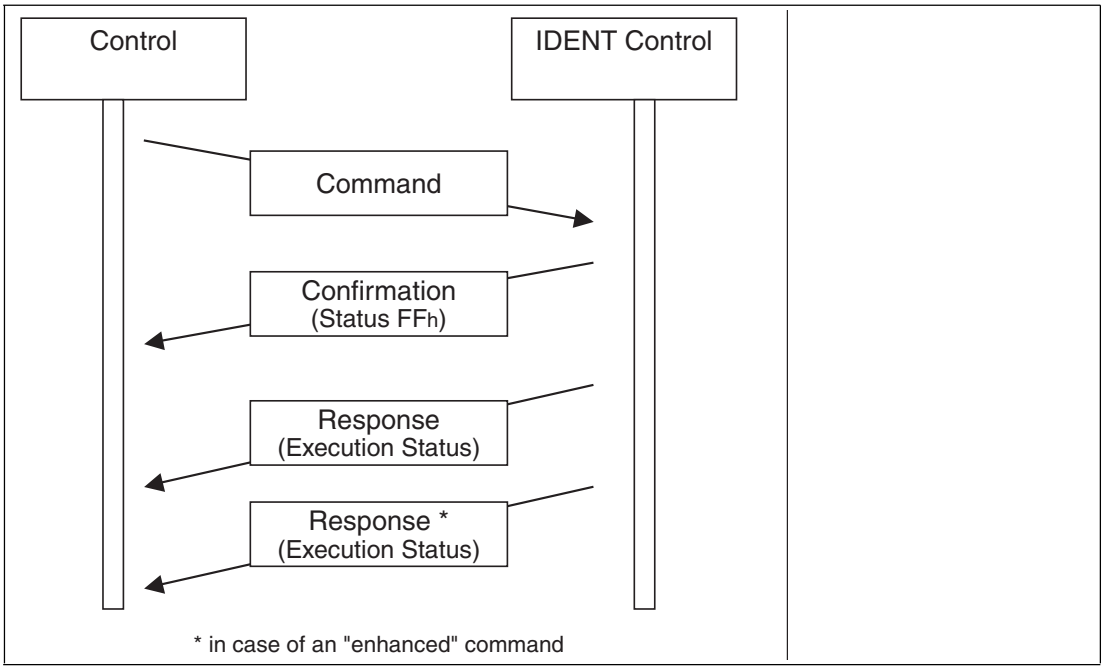
7.1 Communication of the IDENTControl Compact

The IDENTControl Compact is suitable for use in 10 Base-T and 100 Base-TX networks and adapts automatically to the speed of the relevant network. The IDENTControl Compact is fully duplex-compatible.

The IDENTControl Compact communicates via the protocols TCP/IP, MODBUS TCP/IP, EtherNet/IP or PROFINET IO. To change the protocol that the IDENTControl Compact uses to communicate, reset the power supply.

7.2 Data exchange

The transferred data is composed of command, confirmation and response telegrams.



The control software (client) sends a command to the IDENTControl Compact (server). The IDENTControl Compact then sends confirmation of receipt (not with MODBUS TCP/IP). The IDENTControl Compact sends the response after the command is executed.

The IDENTControl Compact can send multiple responses for enhanced and triggered mode commands. Only a single confirmation is sent, however.

A command consists of the telegram length (TCP/IP und MODBUS TCP/IP only), the command code, the channel associated with the read/write head (ident channel), a specified number of parameters, and data relating to the command.

The confirmation consists of the telegram length (TCP/IP and MODBUS TCP/IP only), the echo of the command code, the echo of the ident channel, the status FFh, and the reply counter.

The response consists of the telegram length (TCP/IP and MODBUS TCP/IP only), the echo of the command code, the ident channel, the status, the reply counter, and the requested data.

7.3 Command Execution

The controller initiates an identification command. If the data has changed since the last read-out, the control interface executes the new command If the control interface is scheduled to execute a command a number of times, the toggle bit must be inverted. Only then does the device detect that the command has to be executed again.

2014-02

If the control interface detects a new command, it sets the status in the input field to FFh. The reply counter value increases by 1. The status is displayed (see chapter 7.17) after the control interface has executed the commands.

The toggle bit of the response is the same as the toggle bit of the command.

When new data becomes available, the previous data is overwritten. The reply counter value increases by 1. In the event of an overflow, the reply counter is reset to its start value (01h).

New commands may only be sent to an output field after the response from the previous command is read.

For an overview of supported commands, see chapter 7.10.

Command:

Byte 0*	Telegram length, high byte
Byte 1*	Telegram length, low byte
Byte 2	Command code
Byte 3	Channel/toggle bit = 0
Byte 4	Parameters
Byte 5	Parameters
Byte 6	Data to be written
...	...
Byte N	Data to be written

Table 7.1 * This byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

Confirmation (with MODBUS/TCP, no confirmation is sent)

Byte 0*	Telegram length, high byte
Byte 1*	Telegram length, low byte
Byte 2	Command code (echo)
Byte 3	Channel/toggle bit (echo) = 0
Byte 4	Status FFh
Byte 5	Reply counter
...	00h
Byte N	00h

Table 7.2 * This byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

Response:

Byte 0*	Telegram length, high byte
Byte 1*	Telegram length, low byte
Byte 2	Command code (echo)
Byte 3	Channel/toggle bit (echo) = 0
Byte 4	Status
Byte 5	Reply counter
Byte 6	Read data
...	...
Byte N	Read data

Table 7.3 * This byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

7.4 Command types

When using commands, a distinction is always made between the two command types **single mode** and **enhanced mode**.

Single mode

The command is executed once. A response is issued immediately.

Enhanced mode

The command remains permanently active until it is interrupted by the user or by an error message. A response is issued immediately.

The command remains active after the response is issued. Data is only transferred if read/write tags change. Read/write tags are not read twice. If a read/write tag leaves the read range, the status '5' is output.

7.5 Web function

7.5.1 Website of the IDENTControl Compact

You can configure and operate the IDENTControl Compact using an integral website, which also provides information about the device.

The website is displayed in a standard web browser.

To address the IDENTControl Compact over the web, connect the IDENTControl Compact to a PC using a network cable.

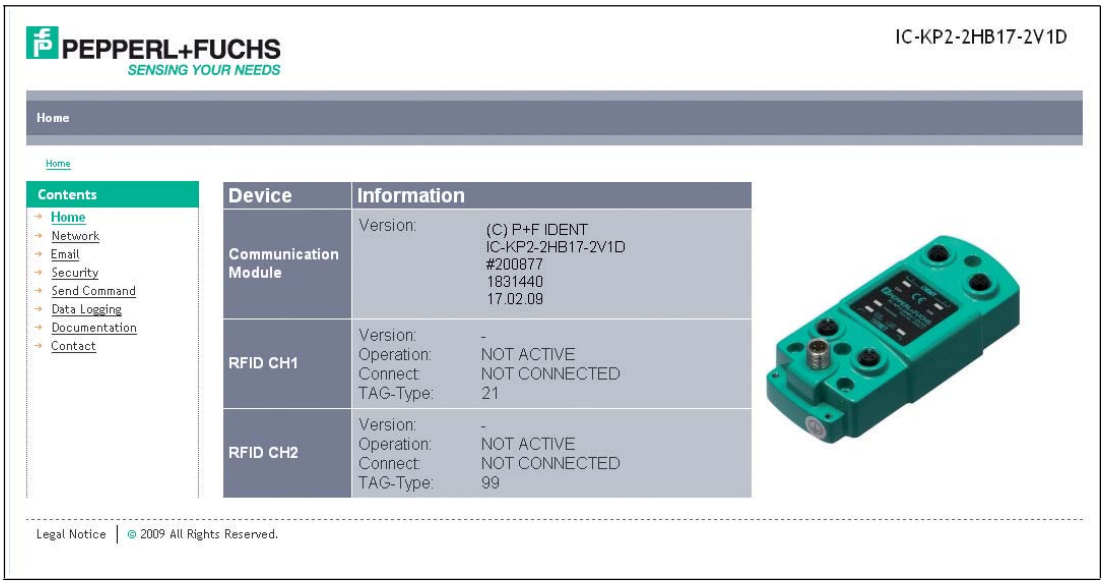
Note!

If you have not changed the IP address of the IDENTControl Compact, the IP address of the PC must fall within the area 169.254.X.X.

Accessing the website

Enter the IP address of the IDENTControl Compact (preset to **169.254.10.12**) in the input field of a web browser and press return to confirm.

↳ The homepage opens.



Select one of the following windows on the homepage by clicking the link in the Contents box:

- Network settings (**Network**)
- Email function settings (**Email**)
- Security settings (**Security**)
- Send command (**Send command**)
- Display command log (**Data logging**)
- Display data sheet (**Documentation**)
- Display contact data (**Contact**)

Entering user name and password

If you select a function by clicking, a window containing boxes for your user name and password appear. The default user name is: "identcontrol" and the default password is: "identcontrol".

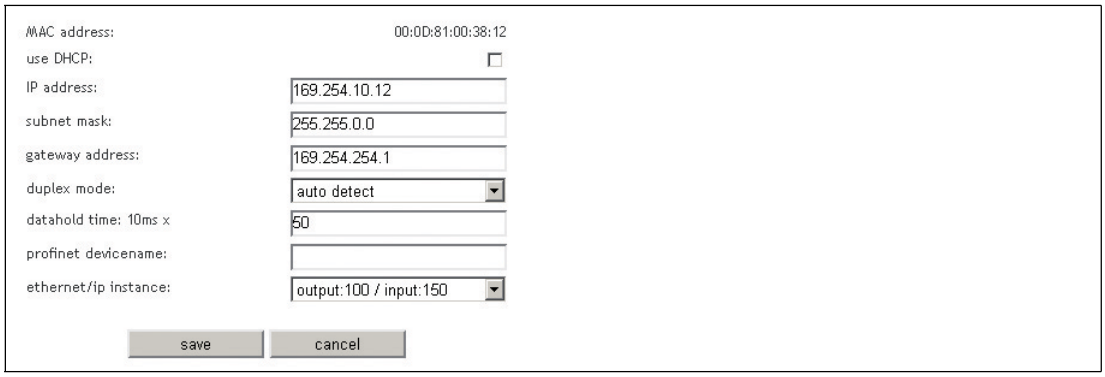
1. Enter "identcontrol" in the **User name** box.
2. Enter "identcontrol" in the **Password** box and press return to confirm.
3. If you have modified the user name and password, enter the new user name and password in the boxes.

You can modify the user name and password in the **Security** window.

7.5.2 Network settings

Settings that you adjust in this window:

- Use/Do not use DHCP server
- Enter IP address
- Enter subnet mask 255.255.255.0
- Enter gateway address
- Select duplex mode
- Set data hold time
- Allocate device names for PROFINET
- Select settings for EtherNet/IP instances



The device performs a restart when you click **save**. The network settings become active after the restart.

Note!

Duplex mode

We recommend the setting **auto detect** for duplex mode. If you do not select **auto detect**, make sure that the same duplex mode is preset for the device communication partner (e.g. the switch).

7.5.3 Email function

In the **Email** window, enter the error codes that will prompt the IDENTControl Compact to send an email. Every response that the IDENTControl Compact sends is allocated a status. The IDENTControl Compact sends an email when the status and error code of a channel are the same.

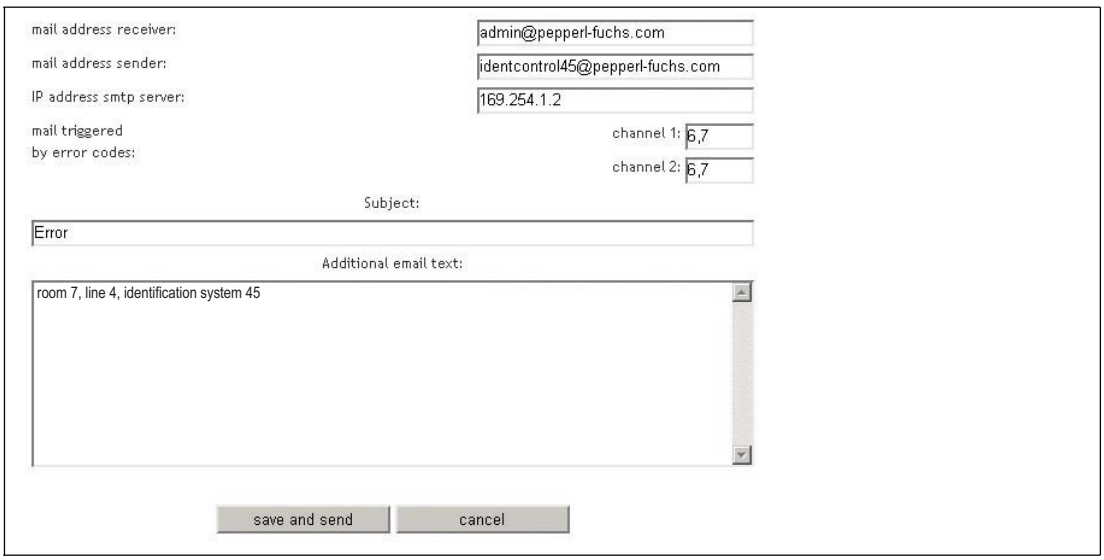
Do not enter an error code for channels that are not connected to a read/write head or are connected to a trigger sensor. If you intend to enter several error codes, separate them with commas. Enter a maximum of five error codes per channel.

We recommend the following entries for error codes:

- 6: Hardware error
- 7: Internal device error

Note!

Your network will require an SMTP server for you to use the email function.



Setting the email function

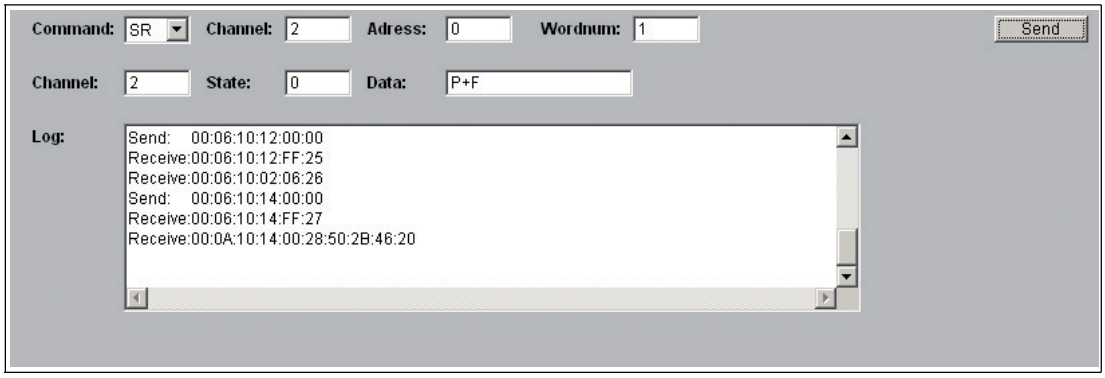
1. Enter the recipient address in the **mail address receiver** box.
2. Enter the email address allocated to the device (sender) in the **mail address sender** box. The email server may have to recognize the address, depending on the SMTP server.
3. Enter the IP address of the SMTP server in the **IP address smtp server** box.
4. Enter the error codes for channel 1 or channel 2 in the **mail triggered by error codes** boxes. Separate the digits with commas.
5. Enter the subject in the **Subject** box.
6. Enter any text that you wish to include in the email such as a location or series information in the **Additional email text** box. The device adds the channel, the type designation of the read/write head, and the cause of the error that has occurred to this text.

2014-02

7.5.4 Sending commands via the website

For commissioning purposes, you can activate an applet in this window, which can be used to execute identification commands that appear in the drop-down menu.

Communication with the identification system is only possible if there is no connection between a controller and the device.



Command:

Drop-down menu with the selection of identification commands. More input boxes may be available depending on the command. An entry is suggested for each command. For an explanation of the commands .

Channel (in the second line):

Channel number of the response telegram.

State:

Status of the response telegram.

Data:

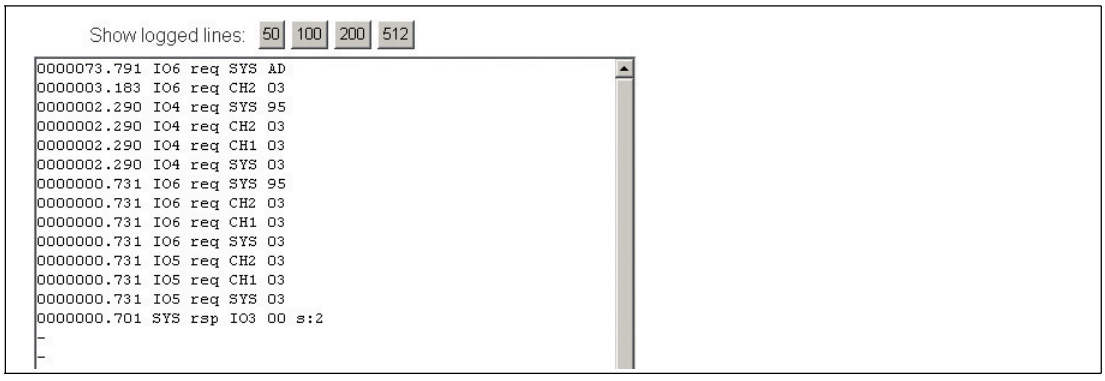
Data of the response telegram (if available).

Reply:

Received response telegrams.

7.5.5 Data logging

You can track commands activated in the IDENTControl Compact in the **Data logging** window. Select between 50, 100, 200, or 512 displayed lines. The current time appears at the top of the window.



2014-02

Example of logged lines

The following appears in the window:

0000029.987 CH1 rsp BUS 01 s:0 l:0005 64.03.03.03.03
0000029.845 BUS req CH1 01

Meaning of logged lines

Time stamp	Sender	Message type	Receiver	Command	Status	Data length	Data
0000029.987	CH1	rsp	BUS	01	s:0	l:0005	64.03.03.03.03
0000029.845	BUS	req	CH1	01			

When the time is 0000029.845, the bus sends the command 01 (=SF) to channel 1 of the IDENTControl Compact. When the time is 0000029.987, the IDENTControl Compact sends the response with the data packet to the bus.

7.6 Communication via TCP/IP

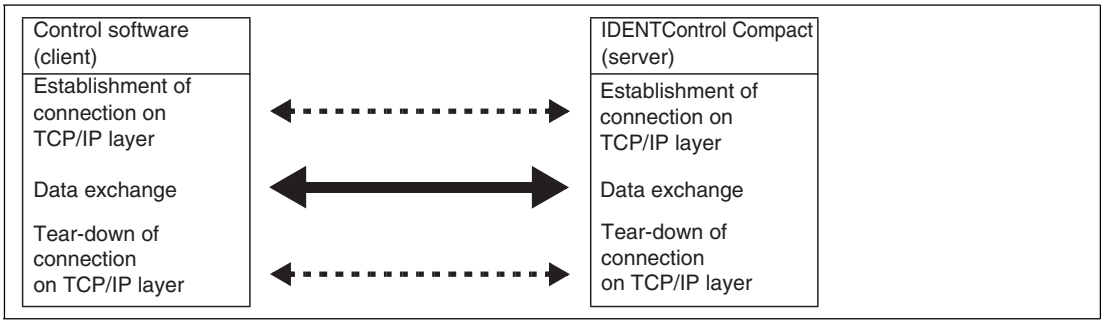
7.6.1 Data exchange via TCP/IP

TCP/IP is a widely used communications protocol. TCP/IP is not classed as an industrial Ethernet protocol, but is used in industrial environments. The advantage of the TCP/IP is that you can use most programming languages to operate the control unit without requiring additional protocol stacks.

The IDENTControl Compact is a TCP/IP **server**. Each function must be actuated by a command from the **client** (control software).

Communication occurs via TCP port 10000: The IDENTControl Compact waits for a connection to be established to port 10000 and responds to incoming commands. Knowledge of TCP/IP sockets is necessary in order to program the control software.

The illustration shows the basic communication flow:



Command

Byte 0	Telegram length, high byte [(N+1) div 256]
Byte 1	Telegram length, low byte [(N+1) mod 256]
Byte 2	Command code
Byte 3	Channel / Toggle bit = 0
Byte 4	Parameter
Byte 5	Parameter
Byte 6	Write data
...	...
Byte N	Write data

2014-02

Confirmation

Byte 0	Telegram length, high byte 0
Byte 1	Telegram length, low byte 6
Byte 2	Command code (Echo)
Byte 3	Channel / Toggle bit (echo) = 0
Byte 4	Status FFh
Byte 5	Reply counter

Response

Byte 0	Telegram length, high byte [(N+1) div 256]
Byte 1	Telegram length, low byte [(N+1) mod 256]
Byte 2	Command code (Echo)
Byte 3	Channel / Toggle bit = 0
Byte 4	Status
Byte 5	Reply counter
Byte 6	Read data
...	...
Byte N	Read data

Note!

The toggle bit is not required for communication via TCP/IP.

7.6.2 Command examples TCP/IP

1st example: Preset data carrier type using the change tag command (see "Change tag (CT)" on page 58)

Command: Change data carrier type on channel 1 to IPC03.

00:06:04:02:30:33 (hexadecimal format)	
00:06	Telegram length (6 bytes)
04	Command code (CT)
02	Reserved/Channel (I), toggle bit (0)
30:33	Data carrier type (IPC03)

Confirmation

00:06:04:02:FF:01	
00:06	Telegram length (5 bytes)
04	Repeat command code (CT)
02	Reserved/Channel (I), toggle bit (0)
FF	Status FFh (processing command)
01	Reply counter

Response: There is a type IPH-... R/W head on channel 1.

00:06:04:02:00:02	
00:06	Telegram length (6 bytes)
04	Repeat command code (CT)
02	Reserved/Channel (I), toggle bit (0)
00	Status 0 (command was executed without error)
02	Reply counter

Alternative response: There is no R/W head on channel 1.

00:06:04:02:06:02	
00:06	Telegram length (6 bytes)
04	Repeat command code (CT)
02	Reserved/Channel (I), toggle bit (0)
06	Status 6 (hardware error)
02	Reply counter

2. Example: Read data carrier using the single read command (see "Single read words (SR)" on page 66)

For this command example, it is assumed that

- the data carrier type IPC03 is set.
- one type IPH-… read head is connected to channel 1.

Command: Read two words from address 0 on channel 1.

00:06:10:22:00:00	
00:06	Telegram length (6 bytes)
10	Command code (SR)
22	Word number (2) / Channel (1), toggle bit (0)
00:00	Word address (0000)

Confirmation

00:06:10:22:FF:01	
00:06	Telegram length (6 bytes)
10	Repeat command code (SR)
22	Word number (2) / Channel (1), toggle bit (0)
FF	Status FFh (processing command)
01	Reply counter

Response: A type IPC03 data carrier is located in front of the read head. The highlighted part is the content of the data carrier.

00:0E:10:22:00:02:**31:32:33:34:35:36:37:38**	
00:0E	Telegram length (14 bytes)
10	Repeat command code (SR)
22	Word number (2) / Channel (1), toggle bit (0)
00	Status 0 (command was executed without error)
02	Reply counter
31:32:33:34:35:36:37:38	**Data**

Alternative response: No data carrier in front of the read head.

00:06:10:02:05:02	
00:06	Telegram length (6 bytes)
10	Repeat command code (SR)
02	Word number (2) / Channel (1), toggle bit (0)
05	Status 5 (no data carrier in the detection range)
02	Reply counter

7.7 Communication via MODBUS TCP/IP

7.7.1 Data exchange via MODBUS TCP/IP

The MODBUS protocol is a simple communication protocol developed in 1979 by Modicon and used to establish master/slave connections between intelligent devices. The MODBUS protocol is an open “de facto standard” and a widely accepted network protocol in industrial environments.

A MODBUS**master** (controller) and a MODBUS **slave** (IDENTControl Compact) exchange data by reading and writing registers. The slave contains read and write registers. The master initiates the data exchange and transfers a command to the write register. The master can then retrieve the response from the read registers. The terms "input register" and "output register" are defined from a PLC perspective.

The commands **read holding registers**, **write multiple registers** and **read/write multiple registers** are available on the IDENTControl Compact for data exchange.

MODBUS communication occurs via port 502. The identification commands correspond to the commands for communication via TCP/IP.

Visit the website of the Modbus Group at www.modbus.org for information on MODBUS TCP/IP.

7.7.2 Registers

Multimaster capability

The IDENTControl Compact can communicate with several masters. Each ident channel can be addressed by two masters. The first master is the controlling master and addresses the device using device ID 1. This master possesses both write and read permissions. The second master is a protocolling master that is able to read the data. The protocolling master addresses the device using device ID 2. The IDENTControl Compact can be addressed once on each channel using device ID 1 and once using device ID 2. If another master attempts to address the device on the same channel, the IDENTControl Compact denies access to this channel. A maximum of six masters can communicate with the device at any one time.

MODBUS commands

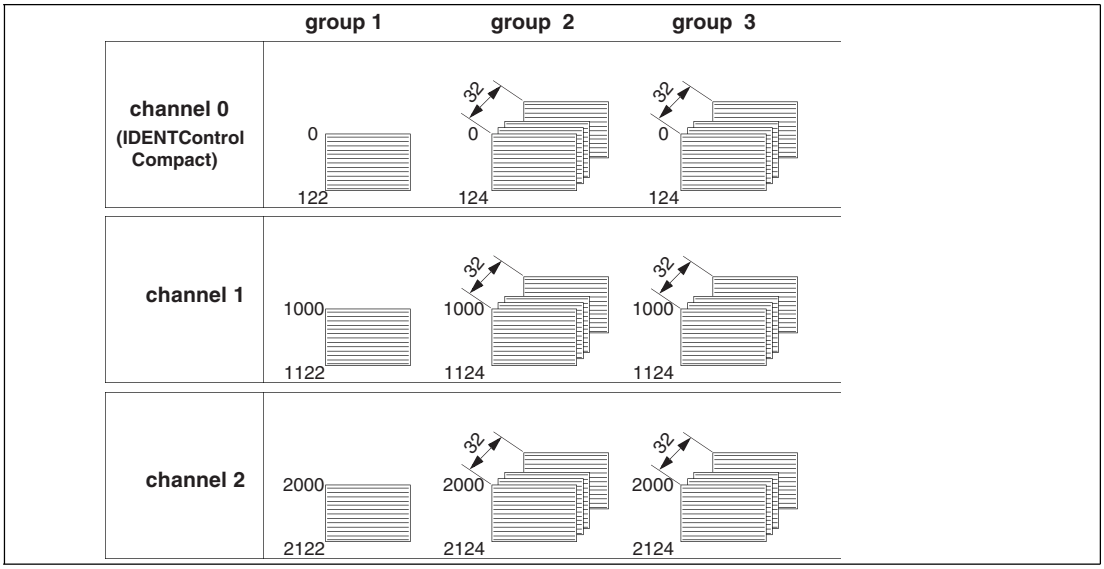
The device supports the commands **read holding registers**, **write multiple registers** and **read/write multiple registers**.

Division of the register

The IDENTControl Compact has two ident channels and a configuration channel. Each channel is assigned a separate register area so that a different master can address each channel. All channels can also be addressed by the same master.

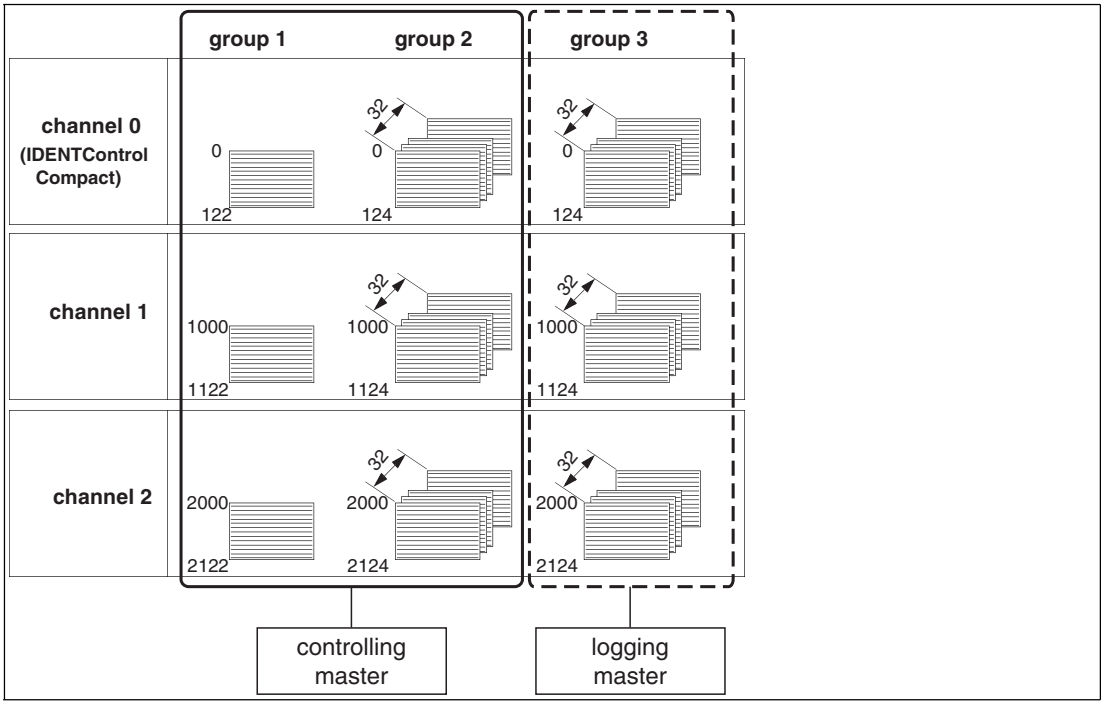
Each channel is assigned a register area containing three register groups:

- Group 1: Output register (device ID 1)
- Group 2: FIFO input register (device ID 1)
- Group 3: FIFO monitor register (device ID 2)



Example: A master for all channels

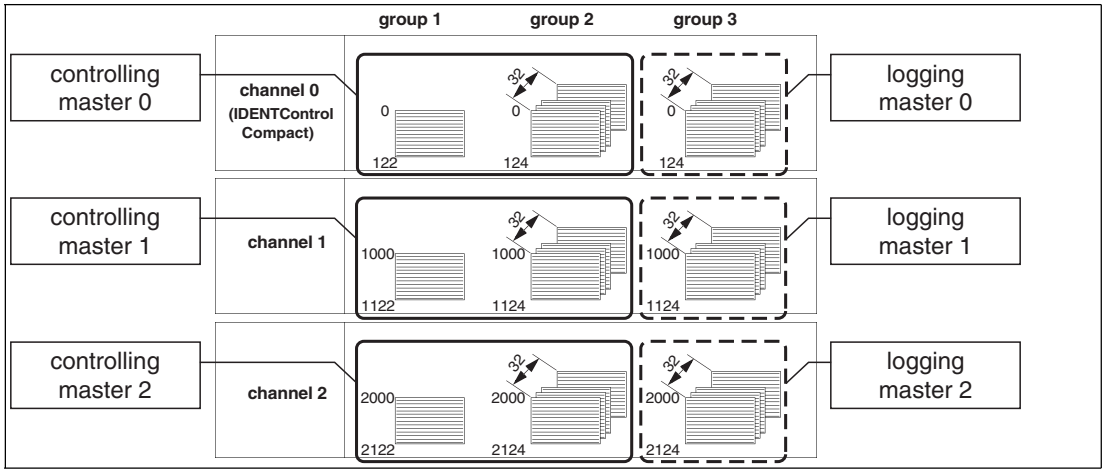
A controlling master communicates with all channels. Optionally, an additional master can be used to log the exchange of data between controlling masters and the IDENTControl Compact.



Example: One controlling and one protocolling master per channel

Each channel has a separate controlling master and a protocolling master. Alternatively, a protocolling master can access channels. Access of two masters to a register group within one channel is not permitted.

2014-02



Group 1: Output Register

Breakdown of the Areas in the Output Register

Address (0-based, decimal)	Byte number of the identification telegram	Use
0 + K	-	Reserved
	-	Reserved/deletion bit (LSB)
1 + K	Byte 0	Telegram length, high byte [(N+1) div 256]
	Byte 1	Telegram length, low byte [(N+1) mod 256]
2 + K	Byte 2	Command code
	Byte 3	Reserved/toggle bit
I + K	Byte N-1	Parameters
	Byte N	Parameters

Table 7.4 K = 0, 1000, 2000, 3000, 4000
I = 3, 4, 5 etc.

If the master sets the deletion bit, all data from the relevant channel (defined by K) that exists in the FIFO register is deleted. The deletion bit must change from 0 to 1.

Group 2 and 3: FIFO Input Register and FIFO Monitor Input Register

Breakdown of the Areas in the Input Register

Address (0-based, decimal)	Byte number of the identification telegram	Use
0 + K	-	Reserved
	-	Utilization A
1 + K	Byte 0	Telegram length, high byte [(N+1) div 256]
	Byte 1	Telegram length, low byte [(N+1) mod 256]
2 + K	Byte 2	Command code (Echo)
	Byte 3	Reserved/channel/toggle bit
3 + K	Byte 4	Status
	Byte 5	Reply counter
I + K	Byte N-1	Data
	Byte N	Data

Table 7.5 K = 0, 1000, 2000, 3000, 4000
I = 4, 5, 6 etc.

FIFO memory

The input data of each channel is stored in two FIFO memories with identical structures. Each FIFO memory is composed of 32 elements.

The utilization of a FIFO memory is indicated in the first register of the relevant channel. When registers are read from the memory, the channel content in the memory is shifted along one memory element. A master can therefore read data from the memory only once.

A protocolling master addresses the device using device ID 2. The protocolling master only has read permissions for the third group.

Utilization of the FIFO Memory

Utilization indicates the percentage of the FIFO memory occupied by data that has not been retrieved. If the response data of the device is retrieved at a lower rate than new data is generated, the FIFO memory overflows and data is lost. In order to avoid data loss, ensure that utilization remains below 15% by setting a high polling rate.

Utilization	Meaning
0	No data available
1–100	Data available
101	Data available, but data lost due to FIFO memory overflowing

7.7.3 MODBUS commands

The IDENTControl Compact supports the MODBUS commands **read holding registers**, **write multiple registers** and **read/write multiple registers**.

The MODBUS master issues a command by sending a request. The slave (IDENTControl Compact) sends a response that corresponds to the MODBUS specifications. If an error occurs, the response contains an exception code.

Command: write multiple registers (10h)

This MODBUS command (request) starts system commands and read/write commands. You must know the following parameters to make the request:

Start address (depending on channel)	Channel 0	0d
	Channel 1	1000d
	Channel 2	2000d
Number of registers to be written:	Maximum 123d	

The IDENTControl Compact sends a response and executes a read/write command from the first address of the relevant channel. If the IDENTControl Compact is scheduled to execute the command a number of times in succession, change the first four bytes of the telegram. Remember to set the toggle bit.

Example

In this example, the tag type IPC03 is set on channel 2 and a read/write command is executed.

Prerequisites

- One type IPH-XX read head is connected to channel 2.
- The IP address of the IDENTControl Compact is preset.
- A network connection is established.

Settings in the MODBUS master

Slave IP address:	IP address of the IDENTControl Compact
Timeout:	1000 ms
Device ID:	1
Start address:	2000d
Number of registers to be written.	4d

The command **write multiple registers** is executed. The tag type is set using the **change tag** (CT) command.

Address (0-based)	Register division Byte number of the identification telegram	Use	Type	Meaning
2000d	High byte	Reserved	00h	-
	Low byte	Reserved/Deletion bit (LSB)	00h	No delete operation
2001d	High byte Byte 0	Length of the identification telegram from this byte onwards	00h	-
	Low byte Byte 1	Length of the identification telegram	06h	6 bytes long
2002d	High byte Byte 2	Command code	04h	**Change tag**
	Low byte Byte 3	Reserved/Channel/Toggle bit	00h	No channel specification required
2003d	High byte Byte 4	Tag type High byte	30h	IPC03
	Low byte Byte 5	Tag type Low byte	33h	IPC03

The IDENTControl Compact confirms the successful transaction. If the IDENTControl Compact does not confirm the transaction, the master signals an error.

The command **write multiple registers** is executed. The read/write command is initiated using the command **enhanced read** (ER).

Address (0-based)	Register division Byte number of the identification telegram	Use	Type	Meaning
2000d	High byte	Reserved	00h	-
	Low byte	Reserved/Deletion bit (LSB)	00h	No delete operation
2001d	High byte Byte 0	Length of the identification telegram from this byte onwards	00h	-
	Low byte Byte 1	Length of the identification telegram	06h	6 bytes long
2002d	High byte Byte 2	Command code	19h	**Enhanced read**
	Low byte Byte 3	Word number/Channel/Toggle bit	40h	Read 4 words, no channel specification required
2003d	High byte Byte 4	Word address High byte	00h	Read from tag address 0
	Low byte Byte 5	Word address Low byte	00h	Same as previous byte

If the master is scheduled to transfer the read/write command to the IDENTControl Compact a second time, the toggle bit must be inverted. This enables cyclic data communication with the PLC.

Command: read holding registers (03h)

This MODBUS command (request) reads the input register. When data becomes available, the IDENTControl Compact writes it to the input register. If the data is not retrieved, up to 32 data sets can be stored in the cache. If the cache is full, the data is lost. If the register is empty of data, the value of the register is 0.

You must know the following parameters to make the request:

Start address (depending on channel)	Channel 0	0d
	Channel 1	1000d
	Channel 2	2000d
Number of registers to be read:	Maximum 125d	

The IDENTControl Compact sends a response that includes the content of the requested register. A read/write command is executed from the first address of the relevant channel.

Example

In this example, the data generated in the previous example is read out.

Prerequisite: The command **write multiple registers** was executed successfully.

Settings in the MODBUS master

Slave IP address:	IP address of the IDENTControl Compact
Timeout:	1000 ms
Device ID:	1
Start address:	2000d
Number of registers to be read:	12d

2014-02

The command **read holding registers** is executed. The register contains the response to the executed command **change tag** (CT).

Address (0-based)	Register division Byte number of the identification telegram	Use	Type	Meaning
2000d	High byte	Reserved	00h	-
	Low byte	Utilization register	06h	6 % of the FIFO memory is utilized
2001d	High byte Byte 0	Length of the identification telegram from this byte onwards	00h	-
	Low byte Byte 1	Length of the identification telegram	06h	6 bytes long
2002d	High byte Byte 2	Command code	04h	**Change tag** command
	Low byte Byte 3	Reserved/Channel/Toggle bit	04h	4 corresponds to channel 2. Channel number shifted 1 bit to the left.
2003d	High byte Byte 4	Status	00h	00h = command executed (meaning of the identification statuses see chapter 7.17)
	Low byte Byte 5	Reply counter	01h	Increases by 1 after each additional response.
2004d - 2011d	High byte	-	00h	-
	Low byte	-	00h	-

The command **read holding registers** is executed. The register contains the response to the executed command **enhanced read** (ER).

Address (0-based)	Register division Byte number of the identification telegram	Use	Type	Meaning
2000d	High byte	Reserved	00h	-
	Low byte	Utilization register	03h	3 % of the FIFO memory is utilized
2001d	High byte Byte 0	Length of the identification telegram from this byte onwards	00h	-
	Low byte Byte 1	Length of the identification telegram	06h	6 bytes long
2002d	High byte Byte 2	Command code	19h	**Enhanced read** command
	Low byte Byte 3	Word number/Channel/Toggle bit	04h	Word count = 0. 4 corresponds to channel 2. Channel number shifted 1 bit to the left.

Address (0-based)	Register division Byte number of the identification telegram	Use	Type	Meaning
2003d	High byte Byte 4	Status	05h	05h = identification read error (meaning of the identification statuses see chapter 7.17)
	Low byte Byte 5	Reply counter	02h	Increases by 1 after each additional response.
2004d - 2011d	High byte	Data	00h	No data read because no tag in front of the read head.
	Low byte	Data	00h	No data read because no tag in front of the read head.

If you hold a type IPC03 tag in front of the read/write head, you can read the data by executing the command **read holding registers** again.

Command: read/write multiple registers (17h)

This MODBUS command combines the functions of the commands **write multiple registers** and **read holding registers**. You should use this command if you wish to systematically exchange data via the PLC.

If you use this command to execute an identification command (e.g. a read command), you will not receive the answer in the response associated with this request. The command requires a certain amount of time to finish processing, after which the data will be available.

You must know the following parameters to make the request:

Writing:

Start address (depending on channel)	Channel 0	0d
	Channel 1	1000d
	Channel 2	2000d
Number of registers to be written:	Maximum 121d	

Reading:

Start address	Start address
Number of registers to be read:	Maximum 125d

The IDENTControl Compact sends a response. If the IDENTControl Compact is scheduled to execute the command a number of times in succession, change the first four bytes of the telegram. Remember to set the toggle bit.

7.7.4 Notes for creating the control program

TCP connection:

Many MODBUS masters enable the use of transactions accompanied by a TCP link connection and disconnection. As outlined in the MODBUS specification, we recommend maintaining as opposed to terminating the connection following a transaction.

Cyclic reading and writing of registers:

Writing:

During a cyclic data exchange, the master can transfer the content of a register a number of times without having to execute the identification command again. If you intend to execute the unmodified identification command again, change the first four telegram bytes. Use the toggle bit for this.
Each identification command contains the number of a channel (ident channel). This number is ignored when MODBUS TCP/IP is used. The channel is defined using the register address. The ident channel is included in the telegram when the register is read.

Reading:

The IDENTControl Compact transfers the response to an identification command once. Note that the controller evaluates each telegram transferred via the bus. The CPU cycle of the controller is faster than the bus cycle. When using **enhanced** commands, we recommend employing the following procedure for distinguishing between old and new data:

1. If a telegram is available: Telegram length > 0?
2. Is the reply counter for this telegram different to the previous one?
3. If the status set to 0?

If you have answered yes to all three questions, new data is available.

Timeout:

The timeout time depends on the load on the network you are using. Adopt 1000 ms as a guide.

Cycle time:

The master must retrieve the data for each channel more frequently than new data is generated to avoid data loss. Select a cycle time that is suitably short. Monitor the FIFO load register to determine whether the refresh rate at which the master retrieves the data is sufficiently high.

Note!

Visit http://www.pepperl-fuchs.com to view an example for a PLC program.

7.7.5 MODBUS exception codes

The device issues a response for each MODBUS transaction. The following table contains a list of possible exception codes:

Code	Name	Description
01	Illegal function	The function code is not 03h, 16h, 17h.
02	Illegal data address	The registers to be written or read are outside of the defined range.
03	Illegal data value	The number of data sets to be read or written is invalid.
04	Slave device failure	Internal error
06	Slave device busy	A MODBUS master attempts to access a channel that is already being used by another MODBUS master.
0A	Gateway path unavailable	The device ID is not 1 or 2.

7.8 Communication via Ethernet/IP

7.8.1 Data exchange via EtherNet/IP

EtherNet/IP is an open fieldbus standard, which enables data exchange between PLCs, PCs, control systems, monitoring systems, sensors, and actuators.
Visit the ODVA website for information on the EtherNet/IP: www.odva.org.

Performance spectrum

- Implicit message
- Explicit message
- PCCC

PLC settings for implicit communication

Configure the following parameters in addition to the IP address:

	Assembly instance	Size (32 bits)
Input	150 - 162 (output + 50d)	2-15*
Output	100 - 112	2-15*
Configuration	112	0

Table 7.6 * see chapter 7.8.5

The lower limit of the RPI is 10 ms.

Data/Command transmission

The data is exchanged by commands that the IDENTControl Compact recognizes. Unlike commands that are used for TCP/IP and MODBUS protocols, EtherNet/IP commands do not have a command length parameter.

The commands are transmitted via Ethernet/IP objects, i.e. objects from classes 04h, 64h, and 65h. There are two different modes: "Mixed mode" and "Separated mode".

7.8.2 Mixed mode

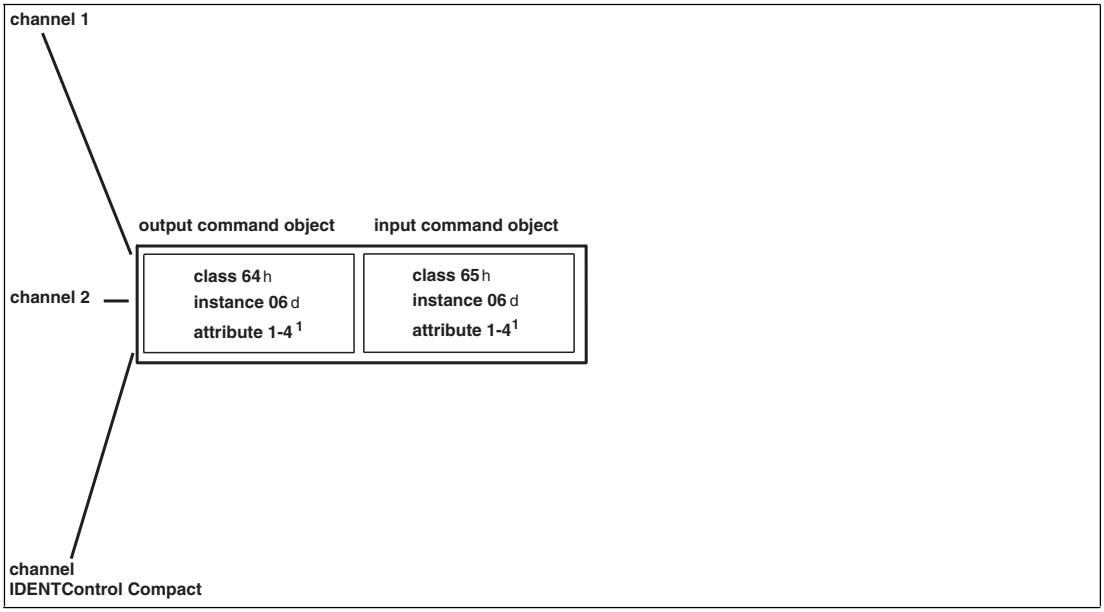


Figure 7.1 1 see chapter 7.8.4

2014-02

PEPPERL+FUCHS

All three ident channels (two read/write head channels, one configuration channel) are addressed using an input and an output instance. The PLC requires less storage space as a result.

The channels are distinguished by the different parameters of the ident telegram.

Refer to appendix A to view an example for mixed mode.

7.8.3 Separated mode

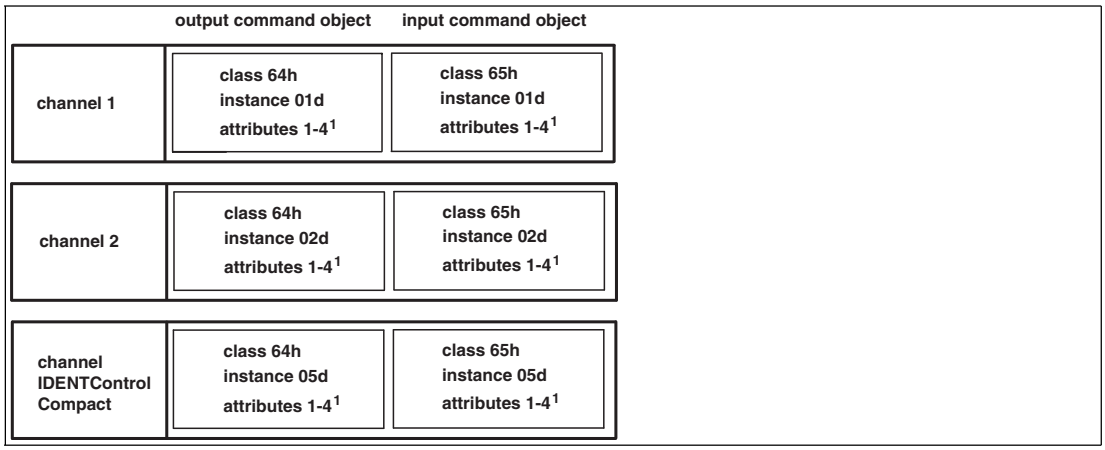


Figure 7.2 1 see chapter 7.8.4

Each ident channel is addressed using a separate input and separate output instance, which simplifies data processing because different ident channels process the data in different areas of the memory.

Refer to appendix A to view an example for separated mode.

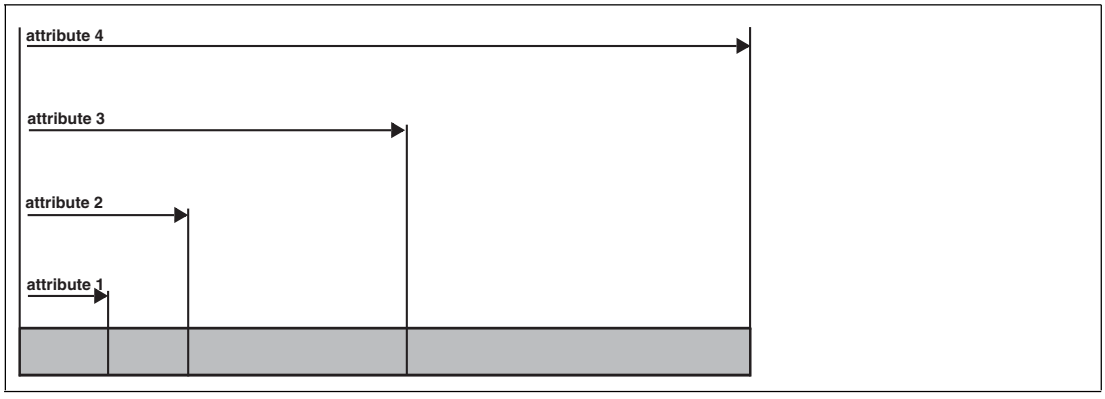
7.8.4 Data length

Depending on the data length required, select one of four attributes with different lengths per input/output instance.

Class 64h and 65h, instance 1, 2, 5, 6

Attribute ID	Data length	Maximum number of double words (4 bytes) that can be read/written at any one time
1	8	1
2	12	2
3	32	7
4	60	14

Attributes 1-3 require fewer data bytes than attribute 4.



7.8.5 Assembly attributes

The attributes of the output object (class 64h) and the input objects (class 65h) appear in different combinations in the assembly object. 26 assembly instances can be used in pairs for implicit communication, therefore there are 13 possible combinations. You can select these combinations either via attribute 100 from instance 0 of the assembly object or by specifying the combination on the PLC (Forward Open).

	Output instance	Input instance	Channel size					Required/ Generated size Output/Input byte
			1	2	Reser - ved	Reser - ved	5 (conf.)	
Mixed mode	100d	150d	8*					8 / 8
	101d	151d	12*					12 / 12
	102d	152d	32*					32 / 32
	103d	153d	60*					60 / 60
Sepa- rated mode	104d	154d	8	8	8	8		32 / 32
	105d	155d	12	12	12	12		48 / 48
	106d	156d	32	32	32	32		128 / 128
	107d	157d	60	60	60	60		240 / 240
	108d	158d	8	8	8	8	8	40 / 40
	109d	159d	12	12	12	12	8	56 / 56
	110d	160d	32	32	32	32	8	136 / 136
	111d	161d	60	60	60	60	8	248 / 248
	112d	162d	0 / 10					0 / 10

Table 7.7 * access to mixed mode instance for input/output command object

The following comparison shows the relationship between input and output instances: Input instance = output instance + 50d.

A combination of the "heartbeat" and the ident status form the output instance 112 and input instance 162. Refer to appendix B for a more detailed view of the object model.

7.8.6 Access administration

The assembly object is a collection of attributes from classes 64h and 65h (input and output). Implicit or explicit access to these objects is possible (via the assembly object). Simultaneous access is regulated to prevent the attributes from overwriting one another.

	Implicit data exchange		Explicit access Instances from input/output commands			
	Output instance	Input instance	1	2	5 (conf.)	6
Mixed mode	100d	150d	x	x	x	x
	101d	151d	x	x	x	x
	102d	152d	x	x	x	x
	103d	153d	x	x	x	x

2014-02

	Implicit data exchange		Explicit access Instances from input/output commands			
	Output instance	Input instance	1	2	5 (conf.)	6
Sepa-rated mode	104d	154d	x	x	✓	x
	105d	155d	x	x	✓	x
	106d	156d	x	x	✓	x
	107d	157d	x	x	✓	x
	108d	158d	x	x	x	x
	109d	159d	x	x	x	x
	110d	160d	x	x	x	x
	111d	161d	x	x	x	x
	112d	162d	✓	✓	✓	x

7.8.7 Heartbeat and ident status

If you select instance 112d and 162d of the assembly object, the size of the output field is 0 bytes and the size of the input field is 10 bytes. The status and the reply counter are included.

Byte status	Description
00	Status of the IDENTControl
01	Reply counter of the IDENTControl
02	Status of channel 1
03	Reply counter from channel 1
04	Status of channel 2
05	Reply counter from channel 2
06	Reserved
07	Reserved
08	Reserved
09	Reserved

The advantage of this procedure is that only a few bytes are transferred via the bus. Identification commands are transferred to the control unit as explicit commands. A new ident response can be read as soon as the value on the reply counter changes.

7.8.8 Data hold time

The data hold time indicates how long the responses from the control unit remain in the input field. Set the data hold time via the boot-up object or on the device website. Select values between 0 and 2.55 seconds in 10 ms increments. The default value is 100 ms. The PLC must read the response before a new response is written so that no responses are lost. Ensure that the data hold time is greater than the maximum cycle time of the whole system (including data storage to the PLC). Do not select a data hold time with a value excessive to requirements because

- the reaction time of the device will increase if several responses arrive in quick succession.
- The maximum possible number of response telegrams per time unit decreases when the data hold time is increased. A memory overflow may occur. The PWR/ERR LED flashes green.

7.8.9 PCCC

PLC5 and SLC500 PLCs use PCCC messaging. The IDENTControl Compact supports PCCC. The data telegrams receive the output and input data from the selected assembly object. Observe the sequence of the bytes (see example).

Supported PCCC commands:

- Write PLC5 type point-to-point
- Read PLC5 type point-to-point

The data is written as an integer data set starting with N14:0. The data is read as an integer data set starting with N7:0. The assembly instance determines the length of the data.

Example: Assembly instance 104, single read command

Configure output instance 104d in the **Network** window on the IDENTControl Compact website.

	102d	152d	32*					32 / 32
	103d	153d	60*					60 / 60
Sepa-	104d	154d	8	8	8	8		32 / 32
rated	105d	155d	12	12	12	12		48 / 48

Write PLC5 type:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N14:0	<WordNum>							<T>	<Command>							
N14:1	<WordAddr> (low byte)								<WordAddr> (high byte)							
N14:2	Write data DW 1 (if not used, please set to 0)															
N14:3	Write data DW 1 (if not used, please set to 0)															
N14:4	<WordNum>							<T>	<Command>							
N14:5	<WordAddr> (low byte)								<WordAddr> (high byte)							
N14:6	Write data DW 1 (if not used, please set to 0)															
N14:7	Write data DW 1 (if not used, please set to 0)															
N14:8	<WordNum>							<T>	<Command>							
N14:9	<WordAddr> (low byte)								<WordAddr> (high byte)							
N14:10	Write data DW 1 (if not used, please set to 0)															
N14:11	Write data DW 1 (if not used, please set to 0)															
N14:12	<WordNum>							<T>	<Command>							
N14:13	<WordAddr> (low byte)								<WordAddr> (high byte)							
N14:14	Write data DW 1 (if not used, please set to 0)															
N14:15	Write data DW 1 (if not used, please set to 0)															

Read PLC5 type:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N7:0	<WordNum>				0	0	1	<T>	<Command>							
N7:1	<ReplyCounter>								<Status>							
N7:2	Read data DW 1															
N7:3	Read data DW 1															
N7:4	<WordNum>				0	1	0	<T>	<Command>							
N7:5	<ReplyCounter>								<Status>							
N7:6	Read data DW 1															
N7:7	Read data DW 1															
N7:8	<WordNum>				0	1	1	<T>	<Command>							
N7:9	<ReplyCounter>								<Status>							
N7:10	Read data DW 1															
N7:11	Read data DW 1															
N7:12	<WordNum>				1	0	0	<T>	<Command>							
N7:13	<ReplyCounter>								<Status>							
N7:14	Read data DW 1															
N7:15	Read data DW 1															

Example: Read/writ operations involving more than 114 registers

Restrictions apply to assembly instances 107d/157d and 111d/161d:

The IDENTControl Compact does not support fragmentation protocols. It is therefore not possible to send or receive more than 114 words simultaneously. The data set must be divided into two separate PCCC commands because instances 107d/157d and 111d/161d use more than 114 registers.

Output instance	Input instance	Channel size					Size needed / generated
		1	2	reserv.	reserv.	5 (conf.)	Output / Input bytes
111d	111d	60	60	60	60	8	248 / 248

1. Command:

Message type	Start register	Number of registers
Write PLC5 type	N14:0	60

2. Command:

Message type	Start register	Number of registers
Write PLC5 type	N14:60	64 (128 bytes / 2)

3. Command:

Message type	Start register	Number of registers
Read PLC5 type	N7:0	60

4. Command:

Message type	Start register	Number of registers
Read PLC5 type	N7:60	64 (128 bytes / 2)

7.9 Communication via PROFINET

7.9.1 What is PROFINET?

PROFINET is an open standard for industrial automation based on industrial Ethernet. PROFINET integrates information technology with established standards such as TCP/IP and XML into automation technology.

The communication concept for setting up decentralized applications within PROFINET is PROFINET IO, i.e. decentralized field devices are installed by PROFINET IO. The familiar IO view of PROFIBUS DP is used where the usable data of the field devices is cyclically transferred to the PLC process image. PROFINET IO is a device model consisting of slots and channels, which is based on the main features of PROFIBUS DP. Descriptions of the field device properties are stored in a **G**eneral **S**tation **D**escription file in XML format ("GSD file" or "GSDML file"). PROFINET IO is engineered in the same way as system integrators from PROFIBUS DP. The decentralized field devices are assigned in the settings of a PLC.

PROFINET IO draws a distinction between three device types: IO controller, IO device, and IO supervisor.

- IO controller: Controller that executes the automation program (PLC).
- IO device: Decentrally assigned field device that is assigned to an IO controller.
- IO supervisor: Programming unit/PC with commissioning and diagnostic function.

In the network, the control unit is an IO device that communicates cyclically with the IO controller during operation.

7.9.2 Project planning using device description (GSD)

As with PROFIBUS DP, the PROFINET IO device is integrated in the configuration tool by way of a device description. The properties of an IO device are stored in the GSD file. The GSD file contains the data of a field device (technical features and information for communication) required to operate the device in a PROFINET network.

You must import the GSD file of the IO device into the configuration tool. Peripheral addresses are assigned to the individual IO channels of the field devices. The peripheral input addresses incorporate the received data. The user program evaluates and processes this data. The user program generates the peripheral output values and sends them to the control unit.

Once planning is complete, the IO controller receives the planning and configuration data. The IO controller programs and configures the IO devices automatically.

The GSD file can be downloaded from our internet site (http://www.pepperl-fuchs.com) by searching for the device name and clicking **Technical documents**.

The following data field sizes (modules) are predefined in the GSD file:

For read/write operation:

"In/Out 8 bytes"	Corresponds to	1 word (32 bits)	Input and output data
"In/Out 12 bytes"	"	2 words	"
"In/Out 16 bytes"	"	3 words	"
"In/Out 20 bytes"	"	4 words	"
"In/Out 24 bytes"	"	5 words	"
"In/Out 28 bytes"	"	6 words	"
"In/Out 32 bytes"	"	7 words	"
"In/Out 64 bytes"	"	15 words	"

For read only operation:

"8 In/4 Out bytes"	Corresponds to	1 word (32 bits)	Input data
"12 In/4 Out bytes"	"	2 words	"
"16 In/4 Out bytes"	"	3 words	"
"20 In/4 Out bytes"	"	4 words	"
"24 In/4 Out bytes"	"	5 words	"
"28 In/4 Out bytes"	"	6 words	"
"32 In/4 Out bytes"	"	7 words	"
"64 In/4 Out bytes"	"	15 words	"

Select one of the predefined modules. In doing so, make sure that the data field size for the read/write commands used is sufficient, depending on the parameter word count.

Note!

The data hold time is stored in the GSD file. This value can be changed via the properties of the IO device.

The data hold time is the time after which the control unit may overwrite the input data field. Select a time that is longer than the cycle time of the PLC. If two data carriers are read directly after one other, the code of the carrier read first remains in the input data field for the specified time, then the next code is entered.

7.9.3 Start-up: Assignment of device name, LED flashes

The IO devices within a PROFINET IO system must have a unique device name. The IO devices are identified in the network by their device name and IP address.

You assign the device name using the configuration tool ("Device creator"). Search for the MAC address of the IO device and select from the device list. Select a unique device name and assign this name to the IO device.

An LED on the device can be made to flash using the configuration tools. If you have several identical devices on the network, this function will enable to you to uniquely identify each device. Select the IO device with reference to the MAC address and activate the flashing function. The "PWR/ERR" LED on the relevant IO device will start to flash.

Please refer to the documentation for the configuration tool you are using for information on the exact procedure for allocating device names and activating the flashing function.

Once you have assigned the device name, load the modified configuration into the IO controller. The IO controller then automatically assigns an IP address to the IO device.

7.9.4 Data Transfer Statistics

Using the example of the Step7 software from Siemens, the graphic below shows an engineering tool that provides data transfer statistics.

You can open a window using the "Module Information" functions in the HW Config hardware configuration. Select the "Statistics" tab to retrieve statistical data regarding the telegram transmission.

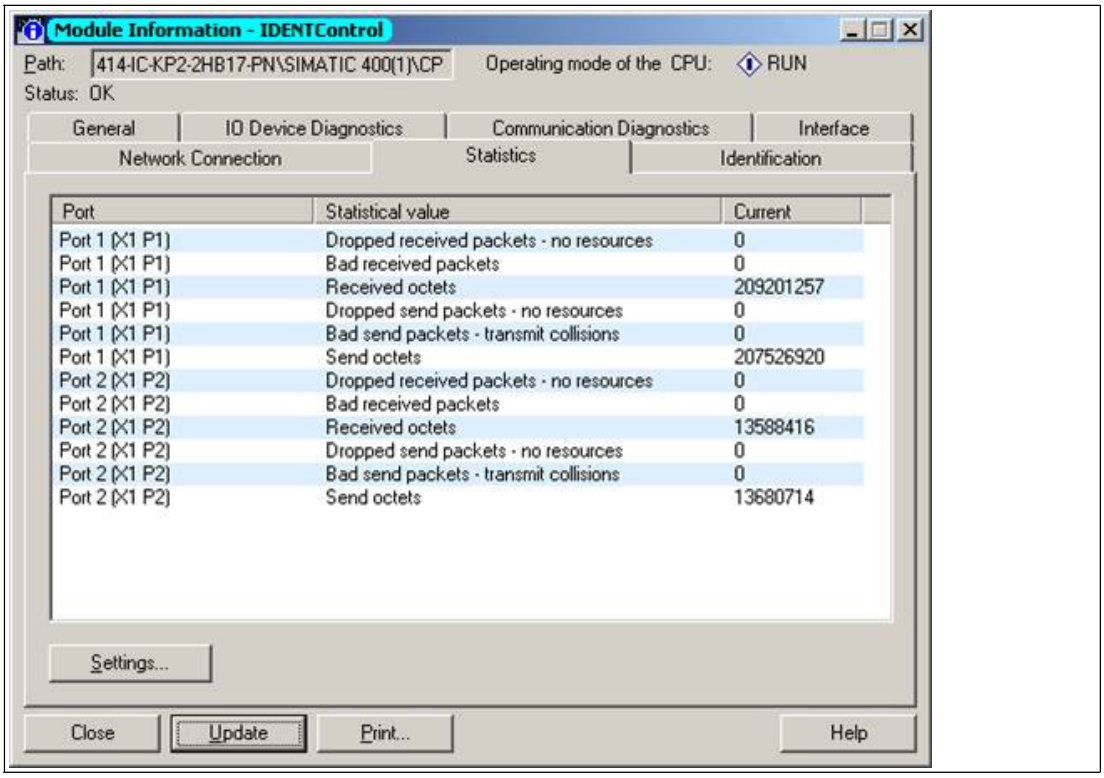


Figure 7.3

7.9.5 Topology detection

To avoid extra effort during commissioning and diagnosis, PROFINET defines a process for topology detection, which displays the system topology graphically in the higher-level controller. The data for representing the topology is stored in the **P**hysical **Dev**ice (PDEV) of the IDENTControl device.

Using the example of the Step7 software from Siemens, you see an engineering tool that offers you options for topology detection using a topology editor.

You can display the interconnections of the PROFINET devices connected to the network in the hardware configuration HW Config. The connected devices must be integrated with the second **D**evice **A**ccess **P**oint (DAP2). The following diagram shows an IC-KP2-2HB17-2V1D with 2 PDEV.

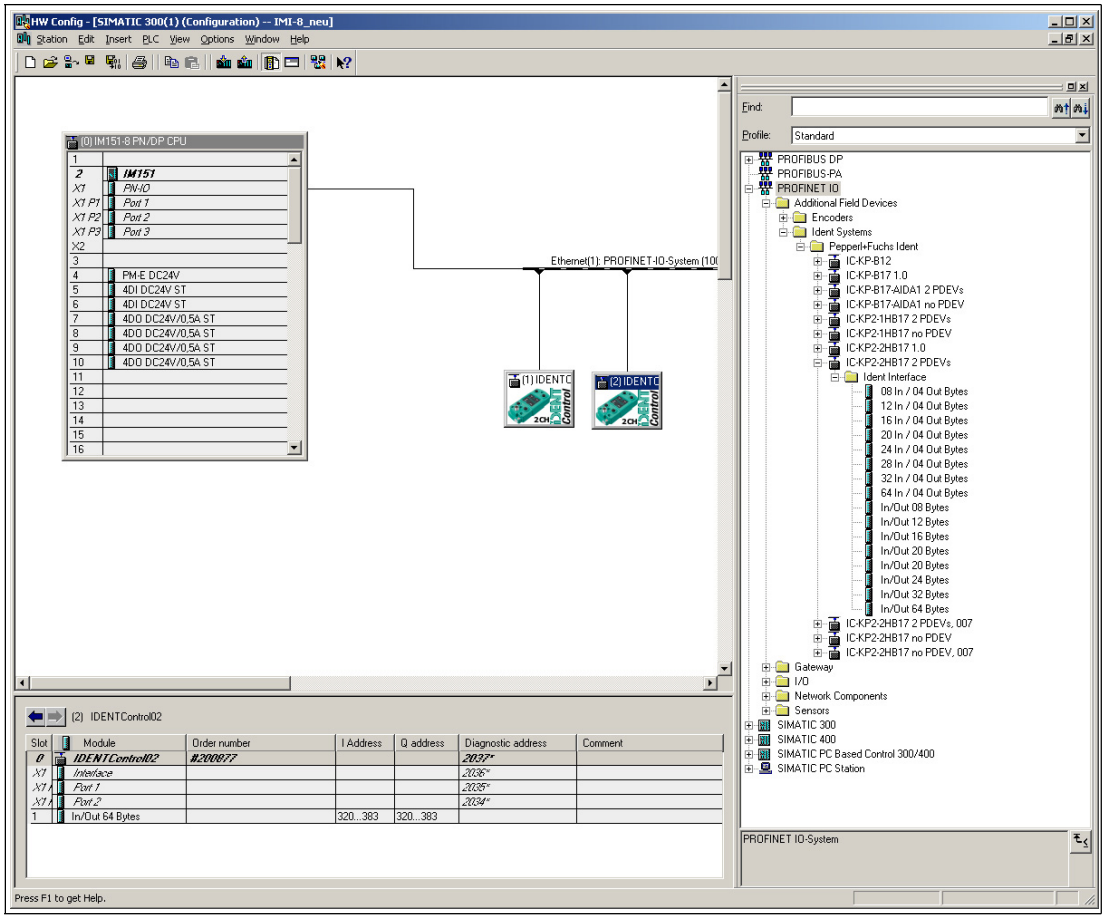


Figure 7.4 HW Konfig / Siemens Step7

Opening the topology editor:

1. Open the topology editor by right-clicking the Ethernet connection and then clicking the "PROFINET IO Topology ..." menu item.

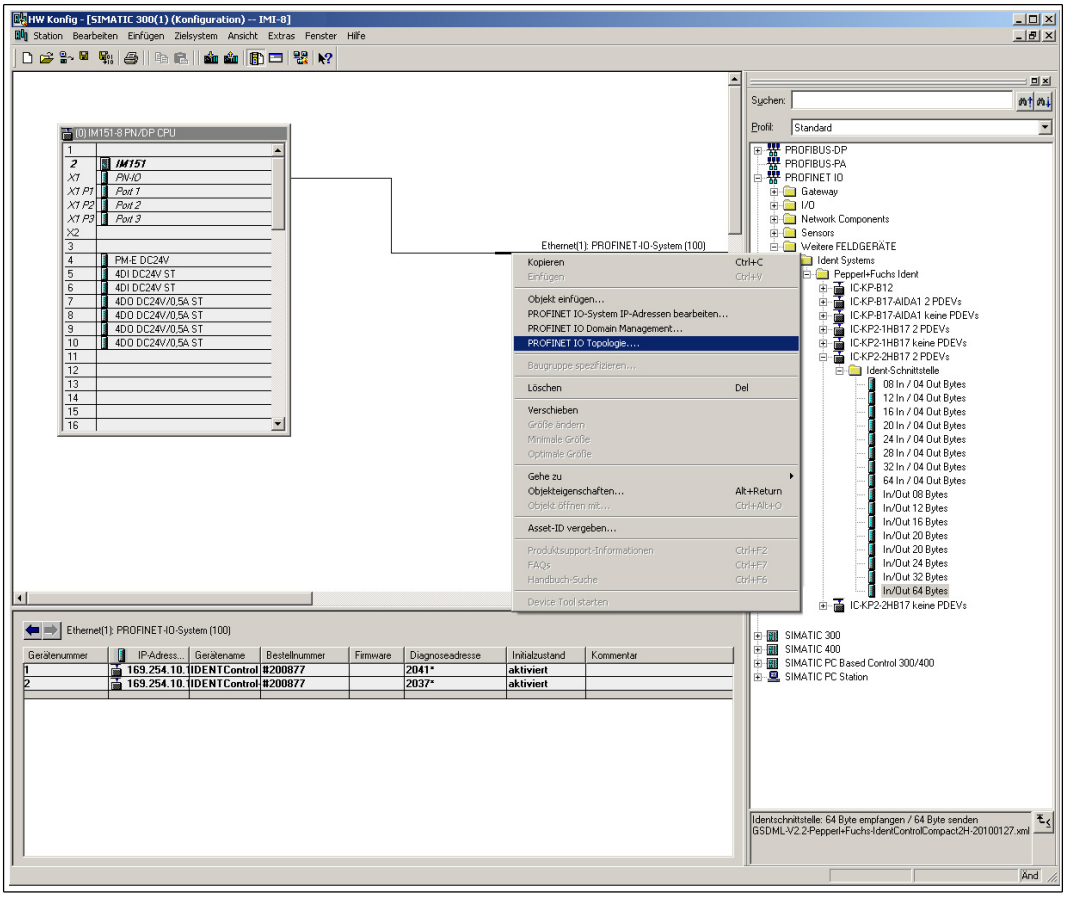


Figure 7.5 Opening the topology editor

2. The topology editor opens.

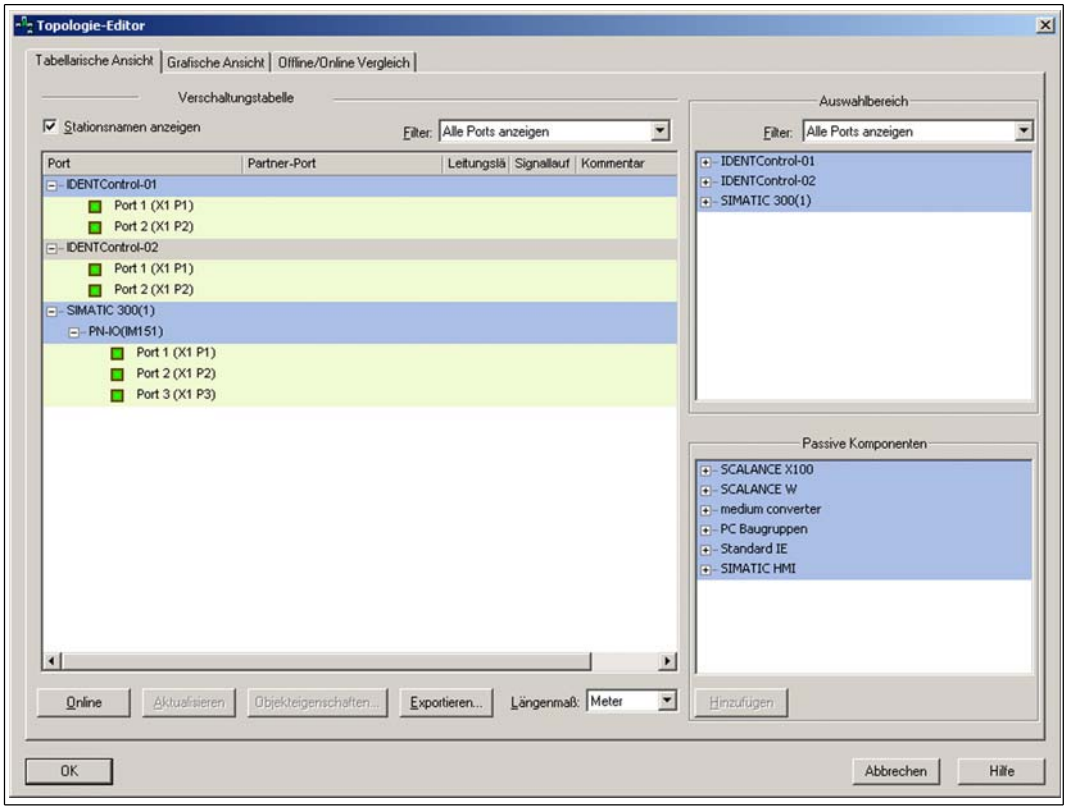


Figure 7.6 Table view

↳ The interconnection table lists 3 devices.

3. Open the offline/online comparison by clicking the **Offline/online comparison** tab.

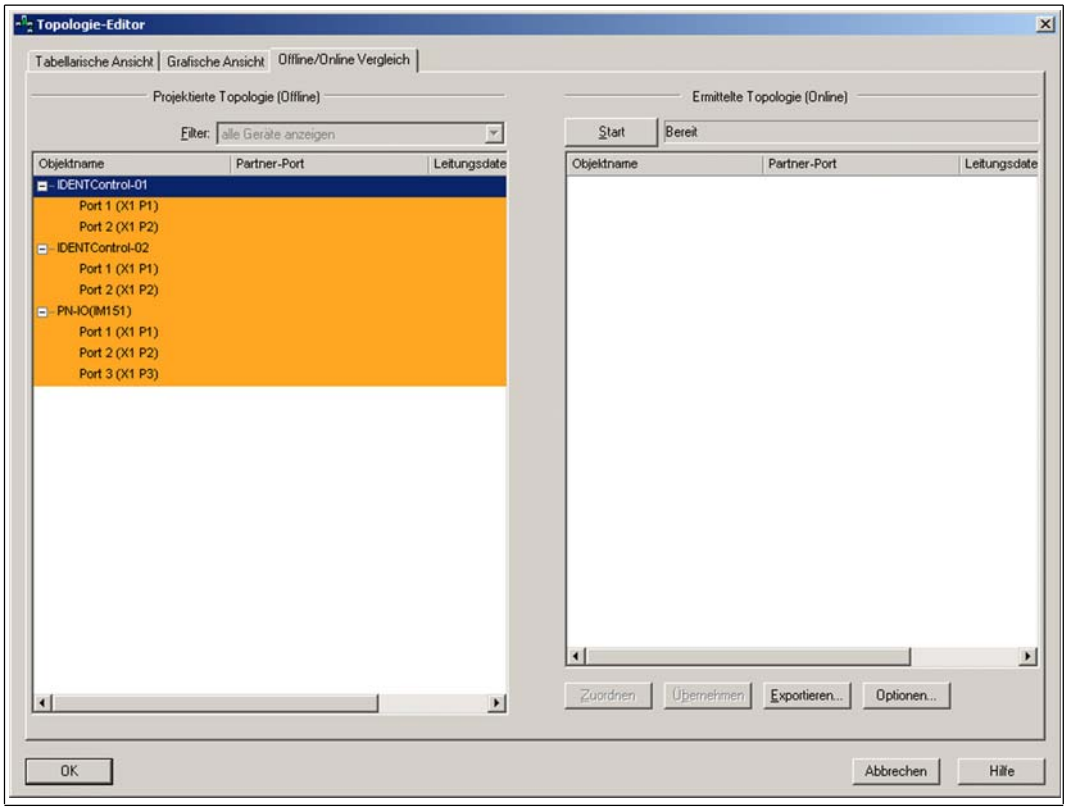


Figure 7.7 Offline/online comparison

↳ The configured topology (offline) is shown in the left-hand window. The configured topology shows the topology stored in the higher-level controller.

4. Press the "Start" button above the right-hand window to identify the actual topology. The detected topology (online) is shown in the right-hand window.
5. Confirm the detected topology by accepting the ports. To do this, select the object name of the device and click the "Apply" button.

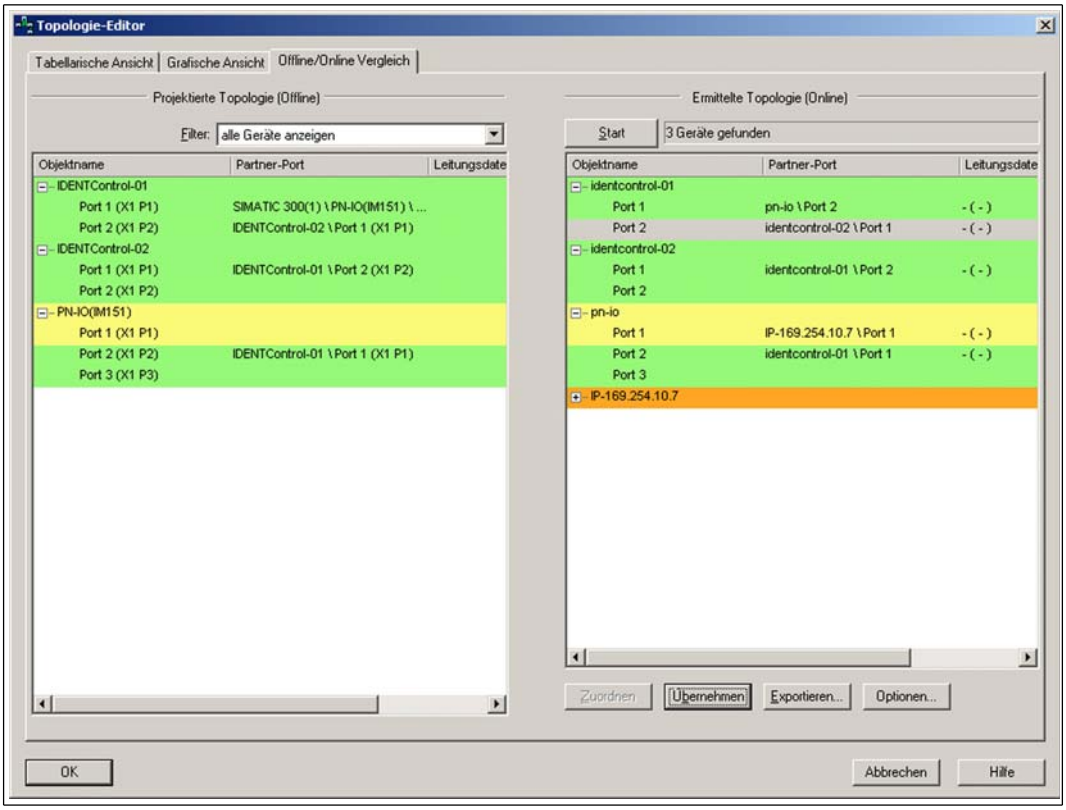


Figure 7.8 Offline/Online Vergleich

6. Open a graphical representation of the topology by clicking the **Graphic view** tab.

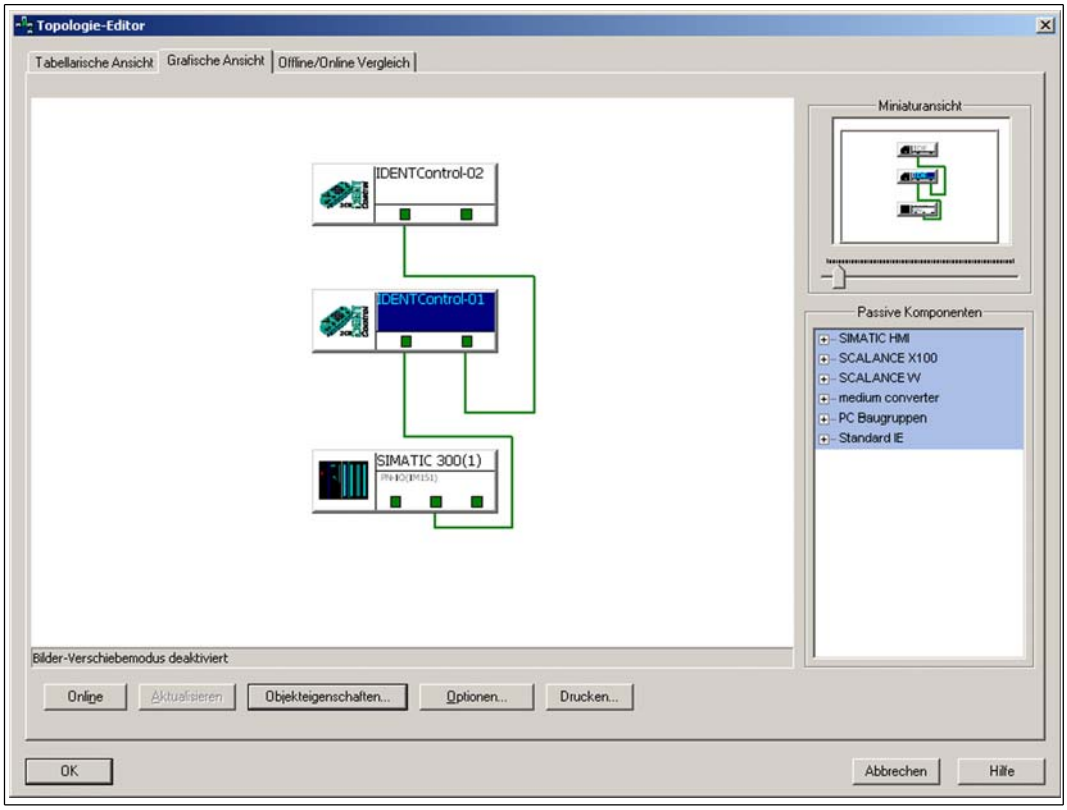


Figure 7.9 Graphic view

↳ The port interconnections for the connected devices are graphically represented.

7. Confirm the configuration by clicking the "OK" button.

↳ The configuration is stored in the higher-level controller as a configured topology.

Tip

Replacing devices without a removable medium

Once the topology has been stored in the controller, each device within the topology can be replaced with another device that is identical in construction. The PLC assigns this device the device name and IP address of the replaced device.

7.9.6 Identification & Maintenance Data

Identification and maintenance data (I&M data) is information stored in a device. I&M data uniquely identifies a device within a system. The identification data (I data) includes information about the device, for example the item number and device name. Identification data cannot be changed.

Maintenance data (M data) includes information regarding the device within the system, for example the installation location and installation date. Maintenance data is initially stored in the device during installation; this data can be changed.

Data Input

The Step7 software from Siemens can be used to display and change the I&M data. You can read and write I&M data 1, 2 and 3 in the "Target System" menu of the HW Config hardware configuration using the functions "Download Module Identification" and "Download Module Identification to PG". → see Figure 7.10 on page 54.

I&M Data

I&M data 1	=	system ID location ID
I&M data 2	=	installation date
I&M data 3	=	additional information

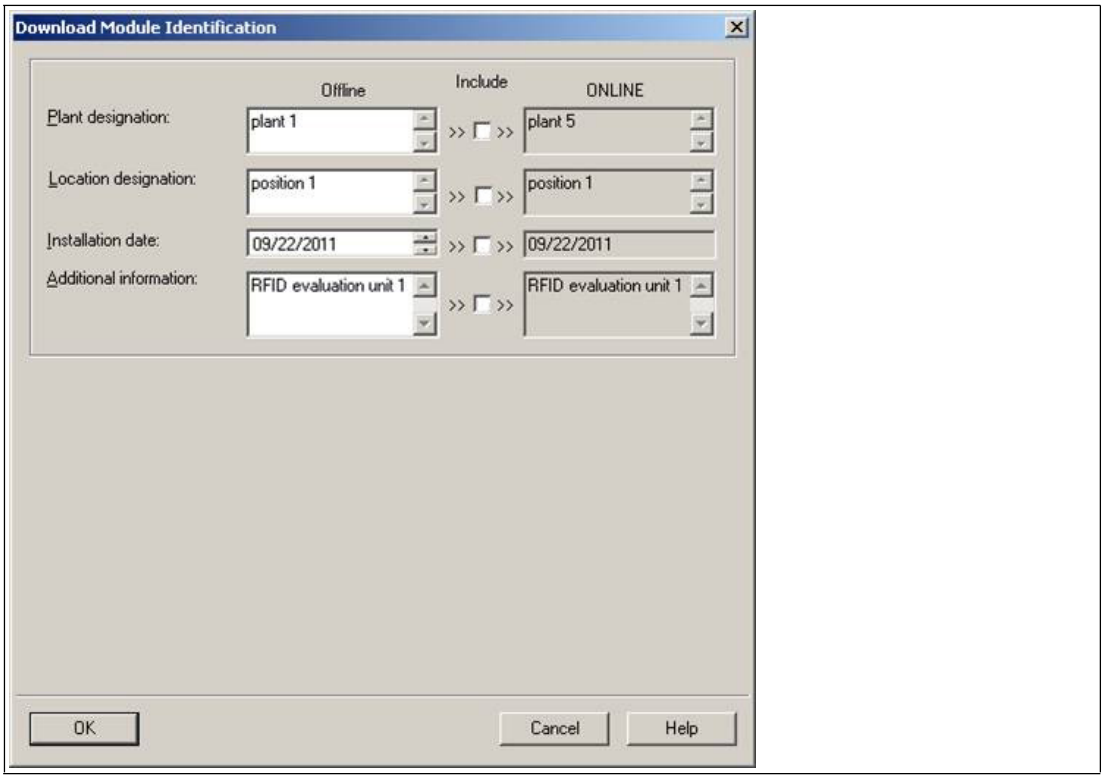


Figure 7.10

7.9.7 Command examples

Example: Define tag type

Command: Change tag type on channel 1 to IPC03

04:02:30:33 (hexadecimal format)	
04	Command code **CT** (change tag)
02	Reserved/Channel (1), toggle bit 0b
30:33	tag type (IPC03)

Confirmation

04:02:FF:01	
04	Repeat command code **CT** (change tag)
02	Reserved/Channel (1), toggle bit 0b
FF	Status FFh (processing command)
01	Reply counter

2014-02

Response: Type IPH-... read/write head on channel 1

04:02:00:02	
04	Repeat command code **CT** (change tag)
02	Reserved/Channel (1), toggle bit 0b
00	Status 0 (command executed without error)
02	Reply counter

Alternative response: No read/write head on channel 1

04:02:06:02	
04	Repeat command code **CT** (change tag)
02	Reserved/Channel (1), toggle bit 0b
06	Status 6 (hardware error)
02	Reply counter

Example: Read tag

Prerequisites:

- The tag type IPC03 is set.
- One type IPH-… read/write head is connected to channel 1.

Command: Read two double words starting from address 0 on channel 1

10:22:00:00	
10	Command code **SR** (single read words)
22	Number of double words (2) / channel (1), toggle bit
00:00	Address of double words (0000)

Confirmation

10:22:FF:01	
10	Repeat command code **SR** (single read words)
22	Number of double words (2) / channel (1), toggle bit
FF	Status FFh (processing command)
01	Reply counter

Response: Type IPC03 tag is located in front of the read/write head, the highlighted part depends on the content of the tag

10:22:00:02:**31:32:33:34:35:36:37:38**	
10	Repeat command code **SR** (single read words)
22	Number of double words (2) / channel (1), toggle bit
00	Status 0 (command executed without error)
02	Reply counter
31:32:33:34:35:36:37:38	**Data**

Alternative response: No tag in front of read/write head

10:02:05:02	
10	Repeat command code **SR** (single read words)
02	Number of double words (0) / channel (1), toggle bit
05	Status 5 (no tag in the detection range)
02	Reply counter

7.10 Command overview

The commands in the list are described in detail on the following pages.

System commands

Command code		Command description	Abbre-viation
4d	04h	See "Change tag (CT)" on page 58	**CT**
2d	02h	See "Quit (QU)" on page 60	**QU**
23d	17h	See "Configuration store (CS)" on page 61	**CS**
22d	16h	See "Reset (RS)" on page 62	**RS**
155d	9Bh	See "Set multiplexed mode (MM)" on page 62	**MM**
156d	9Ch	See "Set trigger mode (TM)" on page 63	**TM**

Standard read/write commands

Fixcode

Command code		Command description	Abbre-viation
1d	01h	See "Single read fixcode (SF)" on page 64	**SF**
29d	1Dh	See "Enhanced buffered fixcode (EF)" on page 65	**EF**

Read data

Command code		Command description	Abbre-viation
16d	10h	See "Single read words (SR)" on page 66	**SR**
25d	19h	See "Enhanced buffered read words (ER)" on page 67	**ER**

Write data

Command code		Command description	Abbre-viation
64d	40h	See "Single write words (SW)" on page 68	**SW**
26d	1Ah	See "Enhanced buffered write words (EW)" on page 69	**EW**

Special command modes

Password mode with IPC03

Command code		Command description	Abbre-viation
24d	18h	See "Set password mode (PM)" on page 72	**PM**
65d	41h	See "Change password (PC)" on page 73	**PC**
66d	42h	See "Set password (PS)" on page 74	**PS**

IPC03 configuration

Command code		Command description	Abbre-viation
97d	61h	See "Single get configuration (SG)" on page 76	**SG**
104d	68h	See "Enhanced buffered get configuration (EG):" on page 77	**EG**

2014-02

Command code		Command description	Abbre-viation
18d	12h	See "Single write configuration (SC)" on page 78	**SC**
102d	66h	See "Enhanced buffered write configuration (EC)" on page 79	**EC**

Writing fixcode IPC11 and IDC-...-1K

Command code		Command description	Abbre-viation
31d	1Fh	See "Single write fixcode (SX)" on page 80	**SX**
36d	24h	See "Enhanced buffered write fixcode (EX)" on page 82	**EX**

Extended commands for tag type IDC-...-1K and IUC...

Command code		Command description	Abbre-viation
10d	0Ah	See "Single read special fixcode (SS)" on page 84	**SS**
113d	71h	See "Enhanced read special fixcode (ES)" on page 85	**ES**
13d	0Dh	See "Single program special fixcode (SP)" on page 86	**SP**
117d	75h	See "Enhanced program special fixcode (EP)" on page 87	**EP**
107d	6Bh	See "Initialize tag (SI)" on page 88	**SI**

Extended commands for tag type IDC-...-1K and IQC...

Command code		Command description	Abbre-viation
71d	47h	See "Single write words with lock (SL)" on page 89	**SL**
72d	48h	See "Enhanced write words with lock (EL)" on page 90	**EL**

Extended commands for IQH2-... and IUH-... read/write heads

With the commands **WriteParam WP** and **ReadParam RD** you can configure the IUH-F117-V1 read/write head using different parameters. The parameters are described in the manual for the read/write head.

Command code		Command description	Abbre-viation
190d	BEh	See "read param (RP)" on page 91	**RP**
191d	BFh	See "write param (WP)" on page 92	**WP**

7.11 System Commands

Change tag (CT)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (04h)	0	0	0	0	0	1	0	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Tag type in ASCII	<TagType> (high byte)							
Byte 5	Tag type in ASCII	<TagType> (low byte)							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (04h)	0	0	0	0	0	1	0	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.8 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

This command tells the read/write head on the relevant channel which tag type to communicate with. This setting is stored in the non-volatile memory on the unit.

Supported Tag Types

Tag type		P+F designation	Chip type	Access	Writable memory [bytes]	Read only code length [byte]	Frequency range
High byte	Low byte						
'0'	'2'	IPC02	Unique, EM4102 (EM microelectronic)	Read only code	5	5	125 kHz
'0'	'3'	IPC03	EM4450 (EM microelectronic), Titan	Read/write read only code	116	4	125 kHz
'1'	'1'	IPC11	Q5 (Sokymat)	Read/write	5	-	125 kHz
'1'	'2'	IPC12	P+F FRAM	Read/write read only code	8k	4	125 kHz
'2'	'0'	IQC20[1)]	All ISO 15693 compliant read/write tags	Read/write read only code	8	8	13.56 MHz
'2'	'1'	IQC21	I-Code SLI (NXP)	Read/write read only code	112	8	13.56 MHz
'2'	'2'	IQC22	Tag-it HF-I Plus (Texas Instruments)	Read/write read only code	250	8	13.56 MHz
'2'	'3'	IQC23	my-D SRF55V02P (Infinion)	Read/write read only code	224	8	13.56 MHz

2014-02

Tag type		P+F designation	Chip type	Access	Writable memory [bytes]	Read only code length [byte]	Frequency range
High byte	Low byte						
'2'	'4'	IQC24	my-D SRF55V10P (Infinion)	Read/write read only code	928	8	13.56 MHz
'3'	'1'	IQC31	Tag-it HF-I Standard (Texas Instruments)	Read/write read only code	32	8	13.56 MHz
'3'	'3'	IQC33[2)]	FRAM MB89R118 (Fujitsu)	Read/write read only code	2k	8	13.56 MHz
'3'	'4'	IQC34	FRAM MB89R119 (Fujitsu)	Read/write read only code	29	8	13.56 MHz
'3'	'5'	IQC35	I-Code SLI-S (NXP)	Read/write read only code	160	8	13.56 MHz
'4'	'0'	IQC40	All ISO 14443A compliant read/write tags	Read only code	-	4/7[6)]	13.56 MHz
'4'	'1'	IQC41	Mifare UltraLight MF0 IC U1 (NXP)	Read/write read only code	48	7	13.56 MHz
'4'	'2'	IQC42[3)]	Mifare Classic MF1 IC S50 (NXP)	Read/write read only code	752	4/7[6)]	13.56 MHz
'4'	'3'	IQC43[3)]	Mifare Classic MF1 IC S70 (NXP)	Read/write read only code	3440	4/7[6)]	13.56 MHz
'5'	'0'	IDC-...-1K	P+F	Read/write read only code	125	4	250 kHz
'5'	'2'	ICC-...	P+F	Read only code	28	7	250 kHz
'7'	'2'	IUC72[4)]	UCode-EPC-G2XM (NXP)	Read/write read only code	64	8	868 MHz
'7'	'3'	IUC73[4)]	Higgs-2 (Alien)	Read only code	-	96	868 MHz
'7'	'4'	IUC74[4)]	UCode-EPC-G2 (NXP)	Read/write read only code	28	96	868 MHz
'7'	'5'	IUC75[4)]	Monza 2.0 (Impinj)	Read only code	-	96	868 MHz
'7'	'6'	IUC76[4)]	Higgs-3 (Alien)	Read/write read only code	56	240	868 MHz
'8'	'0'	-	All Class 1 Gen 2 compliant read/write tags	-	-	Max. 96	868 MHz
'9'	'9'	Depends on the reader [5)]	-	-	-	-	-

1) IQC20 is not an actual tag type as such, but is used to read the UID (read only code) of all ISO 15693 compliant read/write tags.

2) Read/write tag IQC33 can only be used in combination with a IQH1-... read/write head. The memory is divided into 8-byte blocks (instead of 4-byte blocks). You must enter a continuous initial address for write commands SR, ER, SW and EW.
<WordNum> specifies the number of 8-byte blocks (here, max. 7) and must be an even number.

3) Read/write tags IQC40–IQC43 can only be used in combination with a IQH2-... read/write head.
<WordNum> specifies the number of 16-byte blocks and must be a multiple of 4.
The memory can be encrypted for each sector (1 sector = 4 blocks of 16 bytes).
The default key in the tag and reader is FF FF FF FF FF FF_{ASCII}. The key in the reader can be read using the `Read param` command and written using the `Write param` command (see System Commands). The key is only changed in the reader during this process and not in the tag!
The key in the reader is stored in the non-volatile memory.

2014-02

4) IUC7* type read/write tags can only be used with read/write head IUH-F117-V1 in combination with certain control interfaces.

5) The tag type configured in the read/write head as the default is selected.

6) Read/write tags can have 4-byte (older versions) or 7-byte UIDs. IQC42 and IQC43 type read/write tags from Pepperl+Fuchs generally have 7-byte UIDs.

Note!

In a plant where only one tag type is used, it is advantageous to permanently configure that tag type so that the read/write head detects the tag quicker.

Quit (QU)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	0
Byte 2	Command code (02h)	0	0	0	0	0	0	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (02h)	0	0	0	0	0	0	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.9 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The command running on this channel is interrupted.

Configuration store (CS)

Command:

Byte	Contents	Bit no.							
		7	6	5	4	3	2	1	0
Byte 2	Command code (17h)	0	0	0	1	0	1	1	1
Byte 3	Reserved/Ident channel/Toggle bit	-	-	-	-	<Channel>			<T>
Byte 4	Mode	0	0	0	0	0	0	0	<Mode>
Byte 5	not used	-	-	-	-	-	-	-	-
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Response:

Byte	Contents	Bit no.							
		7	6	5	4	3	2	1	0
Byte 2	Command code (17h)	0	0	0	1	0	1	1	1
Byte 3	Reserved/Ident channel/Toggle bit	-	-	-	-	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

The configuration store (CS) command allows you to store the last command sent to the R/W head in the non-volatile memory of the IDENTControl Compact. The R/W head executes the command automatically again if the power supply is interrupted or the IDENTControl Compact is reset.

<Mode>='1' activates the mode.
<Mode>='0' deactivates the mode.

Configuration store is deactivated by default.

Reset (RS)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	0
Byte 2	Command code (16h)	0	0	0	1	0	1	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	0	0	0	<T>

Table 7.10 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

This command terminates all active commands. The device settings are reloaded from the non-volatile memory.

This confirmation is issued for this command (status FFh) instead of a response. The device resets the hardware and then restarts.

Set multiplexed mode (MM)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (9Bh)	1	0	0	1	1	0	1	1
Byte 3	Reserved/Toggle bit	0	0	0	0	0	0	0	<T>
Byte 4	Multiplex mode	0	0	0	0	0	0	0	<T>

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (9Bh)	1	0	0	1	1	0	1	1
Byte 3	Reserved/Toggle bit	0	0	0	0	0	0	0	<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.11 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

This command switches multiplex mode on and off. In multiplex mode, the R/W heads are controlled according to the time multiplex process, i.e. only one R/W head is active. The procedure minimizes mutual interference between R/W heads, allowing two R/W heads to be mounted side by side.

Each IDENT channel sends a response in reply to an MM command.

Multiplex mode <F>='0': Mode off
<F>='1': Mode on

If a R/W head is not connected to a channel, the response telegram receives the status "06h" (hardware fault) from this channel.

2014-02

Set trigger mode (TM)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (9Ch)	1	0	0	1	1	1	0	0
Byte 3	Ident channel/sensor channel/ toggle bit	0	<Ident channel>			<Sensor channel>			<T>
Byte 4	Trigger mode	<Trigger mode>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (9Ch)	1	0	0	1	1	1	0	0
Byte 3	Reserved/sensor channel/ toggle bit	0	0	0	0	<Sensor channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.12 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

Permitted parameters:

<Sensor channel>	1 (0001b), 2 (0010b)
<Ident channel>	0, 1 (0001b), 2 (0010b) (but not <Sensor channel>)
<Trigger mode>	0 (00000000b): Trigger mode off 1 (00000001b): Trigger mode on 2 (00000010b): Trigger mode inverted

Activating trigger mode interrupts a command running on the <Ident channel>.

If trigger mode is activated with <Trigger mode>=1 (=2), dampening the trigger sensor generates the status 0 (5) and after changing to undamped state, generates the status 5 (0) as a response to the <Sensor channel>. Activating trigger mode generates a response that includes the current status of the sensor on the <Sensor channel>.

If a read/write command is sent to the triggered channel <Ident channel> when trigger mode is active, this command is always activated if the <Sensor channel> transmits status 0. <Ident channel> transmits status 0 to confirm receipt of this command.

If you set <Ident channel> 0, the signal is transferred without influencing a reading head.

The command activated by the <Sensor channel> initiates execution as if it had just been restarted by the host.

The command is deactivated again if the status of the <Sensor channel> changes to 5 or trigger mode is deactivated.

If the <Sensor channel> requests a version message, the response contains the status 0 and no other information.

<Ident channel>=0 allows you to assign the trigger signal to channel '0' so that the trigger signal is transmitted to the controller and not to a read head.

This function can be used to monitor functions via the PLC if trigger signals and reading of data cannot occur simultaneously for application related reasons. Correlation must take place in the PLC.

7.12 Standard read/write commands

Single read fixcode (SF)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	0
Byte 2	Command code (01h)	0	0	0	0	0	0	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	1[1)]
Byte 2	Command code (01h)	0	0	0	0	0	0	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	ID code 00h ... FFh	<ID code>							
Byte 7	ID code 00h ... FFh	<ID code>							
...	ID code 00h ... FFh	<ID code>							
Byte N[2)]	ID code 00h ... FFh	<ID code>							

Table 7.13 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) The telegram length depends on length of the read only code from the tag.
2) N = <FixLen> + 6; Ethernet/IP: N = <FixLen> + 4

The R/W head makes only one attempt to read a read only code.

The length of the read only code that is output depends on the tag type. See table "Supported Tag Types" on page 58.

Enhanced buffered fixcode (EF)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	0
Byte 2	Command code (1Dh)	0	0	0	1	1	1	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	1[1]
Byte 2	Command code (1Dh)	0	0	0	1	1	1	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	ID code 00h ... FFh	<ID code>							
Byte 7	ID code 00h ... FFh	<ID code>							
...	ID code 00h ... FFh	<ID code>							
Byte N[2)]	ID code 00h ... FFh	<ID code>							

Table 7.14 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

1) The telegram length depends on length of the read only code from the tag.

2) N = <FixLen> + 6; Ethernet/IP: N = <FixLen> + 4

The R/W head makes attempts until successful to read a read only code. Only data that changes is transferred via the interface, i.e. the R/W head transfers data whenever it reads a new read/write tag or whenever it reads a read/write tag where there was previously no read/write head within the detection range.

The status '05h' (read command) is output whenever a read/write tag leaves the detection range.

The length of the read only code that is output depends on the tag type. See table "Supported Tag Types" on page 58.

Single read words (SR)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (10h)	0	0	0	1	0	0	0	0
Byte 3	Word number/Channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Word address	<WordAddr> (high byte)							
Byte 5	Word address	<WordAddr> (low byte)							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (10h)	0	0	0	1	0	0	0	0
Byte 3	Word number/Channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
Byte 8	Data 00h ... FFh	<Data>							
Byte 9	Data 00h ... FFh	<Data>							
...	Data 00h ... FFh	<Data>							
Byte N[1)]	Data 00h ... FFh	<Data>							

Table 7.15 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) N = 4 x <WordNum> + 5; Ethernet/IP: N = 4 x <WordNum> + 3

The R/W head makes one attempt to read <WordNum> 32-bit words from the address<WordAddr>.

2014-02

Enhanced buffered read words (ER)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (19h)	0	0	0	1	1	0	0	1
Byte 3	Word number/Channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Word address	<WordAddr> (high byte)							
Byte 5	Word address	<WordAddr> (low byte)							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (19h)	0	0	0	1	1	0	0	1
Byte 3	Word number/Channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
Byte 8	Data 00h ... FFh	<Data>							
Byte 9	Data 00h ... FFh	<Data>							
...	Data 00h ... FFh	<Data>							
Byte N[1)]	Data 00h ... FFh	<Data>							

Table 7.16 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) N = 4 x <WordNum> + 5; Ethernet/IP: N = 4 x <WordNum> + 3

The R/W head makes attempts until successful, to read <WordNum> 32-bit words from the address <WordAddr>. Only modified data is transferred via the interface.

When a read/write tag leaves the detection range, the status '05h' (read command) is output.

Single write words (SW)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (40h)	0	1	0	0	0	0	0	0
Byte 3	Word number/Channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Word address	<WordAddr> (high byte)							
Byte 5	Word address	<WordAddr> (low byte)							
Byte 6	Data 00h ... FFh	<Data>							
...	Data 00h ... FFh	<Data>							
Byte N[1)]	Data 00h ... FFh	<Data>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (40h)	0	1	0	0	0	0	0	0
Byte 3	Word number/Channel/Toggle bit	0				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.17 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) N = 4 x <WordNum> + 5; Ethernet/IP: N = 4 x <WordNum> + 3

The R/W head makes one attempt to write <WordNum> 32-bit words from the address<WordAddr>.

Within the reply from the read/write head, <WordNum> is always 0, because the answer does not contain any user data.

Enhanced buffered write words (EW)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (1Ah)	0	0	0	1	1	0	1	0
Byte 3	Word number/Channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Word address	<WordAddr> (high byte)							
Byte 5	Word address	<WordAddr> (low byte)							
Byte 6	Data 00h ... FFh	<Data>							
...	Data 00h ... FFh	<Data>							
Byte N[1)]	Data 00h ... FFh	<Data>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (1Ah)	0	0	0	1	1	0	1	0
Byte 3	Word number/Channel/Toggle bit	0				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.18 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) N = 4 x <WordNum> + 5; Ethernet/IP: N = 4 x <WordNum> + 3

Within the reply from the read/write head, <WordNum> is always 0, because the answer does not contain any user data.

The read/write head repeatedly attempts to write <WordNum> 32-bit words from the address <WordAddr> until successful. After each successful write, the head sends the response and then switches to continuous read. The read/write head then reads the same tag until the tag has left the detection range or a new tag appears within the detection range. At this point, the read/write head again starts write attempts.

The status '05h' is only output when a tag leaves the detection range or is not yet within the detection range.

If two tags enter the read range one immediately after the other, the status '05h' is not issued between the two readings.

7.13 Special commands for type IPC03 tags.

Note!

You can only use the commands in this section for the data carrier type '03' (IPC03).

IPC03 Configuration

The storage of a data carrier IPC03 is organized by word. A data word is defined with a length of 32 bits. For the normal data range, 29 words from addresses 3 through 31 (<WordAddr> = 00h ... 1Ch) are available.

Address	Meaning	<WordAddr>	<ConfAddr>	Note
Word 0	Password	-	-	Write only
Word 1	Protection word	-	1	Read/write
Word 2	Control word	-	2	Read/write
Word 3 ...31	Data range	00h ... 1Ch	-	Read/write
Word 32	Device Serial Number	1Dh	-	Read only
Word 33	Device identification	1Eh	-	Read only

Word 0 contains the password. The password can only be written.

With word 1, the "Protection Word", you can define a read-protected and a write-protected range. The "Protection Word" can only be read and written with the correct password.

With word 2, the "Control Word", you can set various operating modes and the read range for the operating mode "Default Read". The "Control Word" can only be read and written with the correct password.

If you would like to use the "Protection Word" and the "Control Word", you must first activate the password mode.

The individual bits have the following meanings:

Protection word		
Bit	Meaning	Byte
0 ... 7	First read-protected word	0
8 ... 15	Last read-protected word	1
16 ... 23	First write-protected word	2
24 ... 31	Last write-protected word	3

Control word		
Bit	Meaning	Byte
0 ... 7	Read range start	0
8 ... 15	Read range end	1
16	Password mode on/off	2
17	"Read after write" operating mode on/off	
18 ... 23	Open	
24 ... 31	Open	3

2014-02

IPC03 password mode

If the password mode in the data carrier is activated, the data range of the data carrier is read and write-protected and can only be read or written if the R/W head sends the correct password to the data carrier.

If the password mode in the data carrier is deactivated, every data word on the data carrier can be read or written.

The default password of the R/W heads and the data carrier is 00000000h. In the R/W head, the password is stored in the volatile memory and in the data carrier, the password is stored in the non-volatile memory.

To read or write the "Protection Word" and the "Control Word", you must first enter the password in the password mode (see the commands **SC** or **EC**).

You can also limit access to the data carriers by defining the start and end of a read-protected and a write-protected range in the Protection Word.

Setting the password

1. Enter the correct password once with the command **PS** (set password).
2. Activate the password mode with the command **PM** (set password mode).

Changing the password

To change the password in the R/W head and on the read/write tag, use the command **PC**.

Set password mode (PM)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (18h)	0	0	0	1	1	0	0	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Password mode	0	0	0	0	0	0	0	<P>

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (18h)	0	0	0	1	1	0	0	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							

Table 7.19 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The command **PM** activates and deactivates the password mode of the relevant channel. In the password mode, the password is transferred to the data carrier before each read/write access. If a data carrier is addressed with the wrong password, then even the other data areas on the data carrier can no longer be accessed.

Password mode "off":	<P>=0 (0b)	(deactivated)
Password mode "on":	<P>=1 (1b)	(activated)

Change password (PC)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	1	0	0
Byte 2	Command code (41h)	0	1	0	0	0	0	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Old password 00h ... FFh	<PSW> (byte 3)							
Byte 5	Old password 00h ... FFh	<PSW> (byte 2)							
Byte 6	Old password 00h ... FFh	<PSW> (byte 1)							
Byte 7	Old password 00h ... FFh	<PSW> (byte 0)							
Byte 8	New password 00h ... FFh	<PSW> (byte 3)							
Byte 9	New password 00h ... FFh	<PSW> (byte 2)							
Byte 10	New password 00h ... FFh	<PSW> (byte 1)							
byte 11	New password 00h ... FFh	<PSW> (byte 0)							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (41h)	0	1	0	0	0	0	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							

Table 7.20 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The command **PC** changes the password in a tag. Enter the old and then the new password <PSW> here. If the password has been successfully written, the password in the read/write head also changes and the **set password** command is no longer required. The password of the IPC03 can also be changed if the password mode is deactivated.

Set password (PS)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (42h)	0	1	0	0	0	0	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Reserved	0	0	0	0	0	0	0	0
Byte 6	Password 00h ... FFh	<PSW> (byte 3)							
Byte 7	Password 00h ... FFh	<PSW> (byte 2)							
Byte 8	Password 00h ... FFh	<PSW> (byte 1)							
Byte 9	Password 00h ... FFh	<PSW> (byte 0)							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (42h)	0	1	0	0	0	0	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							

Table 7.21 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The command **PS**sets the password, which the R/W head communicates to the data carrier in password mode.

Operating mode “Default Read”

In "default read" operating mode, 1 or 2 words are read extremely quickly. The area of memory earmarked for reading is already specified on the tag. The R/W head does not have to identify the memory area for the tag.

The start and end of the read range are stored in the bytes 0 and 1 of the control word. As soon as power is supplied to the tag, it sends data from the data range defined by the start and end of the read range. The data range between read range start and end is read with the read commands **SR** (Single read words) and **ER** (enhanced buffered read words) when <WordAddr> is set to 0000h and <WordNum> to 00h.

The advantage of "default read" operating mode is the readout speed. The readout of one data word (4 bytes) is twice as fast in this mode as the other modes. The readout of two words takes approx. 1/3 less time. No more time advantages can be gained after three data words because "default read" mode is designed to read a maximum of two words (= 8 bytes). Reading larger data ranges can lead to error messages if the reading head does not respond within the planned reaction time.

Note!

The addresses for the start and end of the read range are based on the absolute word address of the read/write tag, not on <WordAddr>.
Example: With the setting read range start 03h and read range end 03h, the R/W head only reads the first data word in the read/write tag.

Setting "Default Read"

1. Activate the password mode.
2. Write the read range start and end into the "Control Word".
3. Deactivate the password mode.
4. Read the data range with address designation 0000h and word count 0h.

Single get configuration (SG)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (61h)	0	1	1	0	0	0	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Address in the configuration range	<ConfAddr>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (61h)	0	1	1	0	0	0	0	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
Byte 8	Data 00h ... FFh	<Data>							
Byte 9	Data 00h ... FFh	<Data>							

Table 7.22 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head makes exactly one attempt to read a word in the configuration range ("Protection Word" or "Control Word") from the address <ConfAddr>.

Enhanced buffered get configuration (EG):

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	1
Byte 2	Command code (68h)	0	1	1	0	1	0	0	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Address in the configuration range	<ConfAddr>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (68h)	0	1	1	0	1	0	0	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
Byte 8	Data 00h ... FFh	<Data>							
Byte 9	Data 00h ... FFh	<Data>							

Table 7.23 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head attempts to read a word in the configuration range from the address <ConfAddr> until successful. Only data that changes is transferred via the interface, i.e. the R/W head transfers data whenever it reads a new data carrier or whenever it reads a data carrier where there was previously no R/W head within the detection range.

The status '05h' (read/write command) is output when the data carrier leaves the detection range or if the data carrier is not yet within the detection range when the command is executed.

If two data carriers enter the read range one immediately after the other, the status '05h' is not issued between the two readings.

Single write configuration (SC)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (12h)	0	0	0	1	0	0	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Address in the configuration range	<ConfAddr>							
Byte 6	Data 00h ... FFh	<Data byte 3>							
Byte 7	Data 00h ... FFh	<Data byte 2>							
Byte 8	Data 00h ... FFh	<Data byte 1>							
Byte 9	Data 00h ... FFh	<Data byte 0>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (12h)	0	0	0	1	0	0	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.24 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head makes exactly one attempt to write a word to the configuration range ("Protection Word" or "Control Word") from the address <ConfAddr>.

The password mode must be active so that the R/W head can write to the configuration range.

If the password mode is deactivated, every data word outside of the write-protected range can be written to. If you would like to modify the write-protected range, you must modify the "Protection Word" accordingly.

Enhanced buffered write configuration (EC)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (66h)	0	1	1	0	0	1	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Address in the configuration range	<ConfAddr>							
Byte 6	Data 00h ... FFh	<Data byte 3>							
Byte 7	Data 00h ... FFh	<Data byte 2>							
Byte 8	Data 00h ... FFh	<Data byte 1>							
Byte 9	Data 00h ... FFh	<Data byte 0>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (66h)	0	1	1	0	0	1	1	0
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.25 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head attempts to write a word in the configuration range to the address <ConfAddr> until successful. After each write, the status is evaluated and the system waits until a new data carrier is within the detection range. The command then starts again from the beginning. In order to write in the configuration range, the password mode must be active.

The status '05h' (read/write command) is only output when a data carrier leaves the detection range or is not yet within the detection range when the command is executed.

If two data carriers enter the read range one immediately after the other, the status '05h' is not issued between the two readings.

Write read only code IPC11 and IDC-..-1K

"Read-after-write" operating mode is not used.

7.14 "Write Read Only Code" Command for Type IPC11 and IDC-...-1K Read/Write Tags

Tags IPC11 can be programmed to behave like the IPC02 read only tag. To do this, use the commands **SX** and **EX**. The code is read when tag type '02' or '11' is set with the commands **SF** and **EF**.

Tags IDC-...- 1K can be programmed to behave like the ICC read only tag. This programming occupies the first 8 bytes in the tag and occurs when the tag type '50' is set with the commands **SX** or **EX**.
This code is read when tag type '52' is set with the commands **SF** or **EF**. If you use the command **SF** or **EF** when tag type '50' is selected, the 4-byte read only code of the tag is issued.

Single write fixcode (SX)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	1
Byte 2	Command code (1Fh)	0	0	0	1	1	1	1	1
Byte 3	FixLen/Channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	FixType	<FixType> (high byte)							
Byte 5	FixType	<FixType> (low byte)							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
Byte 8	Data 00h ... FFh	<Data>							
Byte 9	Data 00h ... FFh	<Data>							
Byte 10	Data 00h ... FFh	<Data>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (1Fh)	0	0	0	1	1	1	1	1
Byte 3	FixLen/Channel/Toggle bit	0	1	0	1	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.26 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head makes only one attempt to write a read only code.

IPC11: <FixLen> = 5

<FixType> = '02' ASCII (30h 32h), the read only code cannot be changed
'11' ASCII (31h 31h), the read only code can be overwritten

2014-02

IDC-...-1K:	<FixLen>	=	7 The first 3 bytes are hexadecimal (0h ... Fh), the last 4 bytes are decimal (0d ... 9d).
	<FixType>	=	'52' ASCII (35h 32h), the read only code can be overwritten
	<Data>	=	(Byte 1 to 3): 0x30 ... 0x39; 0x41...0x46 (Byte 4 to 7): 0x30...0x39

Type IDC-...-1K tags can be programmed in such a way that they are compatible with the type ICC-... read only carriers. This programming occupies the first 8 bytes in the tag. The read/write commands can be used to access the remaining memory.

You must set the tag type '50' in order to program type IDC-...-1K tags. To do this, transmit the command **SX** or **EX**.

The value range contains 7 characters:

- the first 3 characters contain the values 0 ... F (hexadecimal code)
- the last 4 characters contain the values 0 ... 9 (decimal code)

You must select the tag type '50' (ICC-...) beforehand in order to read out this code. If a "read only code" command is executed when the tag type '50' (IDC-...-1K) is set, the 4-byte read only code for this tag is issued.

Enhanced buffered write fixcode (EX)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	1
Byte 2	Command code (24h)	0	0	1	0	0	1	0	0
Byte 3	FixLen/Channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	FixType	<FixType> (high byte)							
Byte 5	FixType	<FixType> (low byte)							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
Byte 8	Data 00h ... FFh	<Data>							
Byte 9	Data 00h ... FFh	<Data>							
Byte 10	Data 00h ... FFh	<Data>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	1	0
Byte 2	Command code (24h)	0	0	1	0	0	1	0	0
Byte 3	FixLen/Channel/Toggle bit	0	1	0	1	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							

Table 7.27 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head constantly attempts to write a read only code. After each successful write, the response is sent and the system waits until a new tag is within the detection range. The command then starts again from the beginning.

IPC11:

<FixLen> = 5

<FixType> = '02' ASCII (30h 32h), the read only code cannot be changed
'11' ASCII (31h 31h), the read only code can be overwritten

IDC-...-1K:

<FixLen> = 7
The first 3 bytes are hexadecimal (0h ... Fh), the last 4 bytes are decimal (0d ... 9d).

<FixType> = '52' ASCII (35h 32h), the read only code can be overwritten

<Data> = (Byte 1 to 3): 0x30 ... 0x39; 0x41...0x46
(Byte 4 to 7): 0x30...0x39

Type IDC-...-1K tags can be programmed in such a way that they are compatible with the type ICC-... read only carriers. This programming occupies the first 8 bytes in the tag. The read/write commands can be used to access the remaining memory.

You must set the tag type '50' in order to program type IDC-...-1K tags. To do this, transmit the command **SX** or **EX**.

2014-02

The value range contains 7 characters:

- the first 3 characters contain the values 0 ... F (hexadecimal code)
- the last 4 characters contain the values 0 ... 9 (decimal code)

You must select the tag type '50' (ICC-...) beforehand in order to read out this code. If a "read only code" command is executed when the tag type '50' (IDC-...-1K) is set, the 4-byte read only code for this tag is issued.

7.15 Extended Command Modes

Extended commands for type IDC-... and IUC… tags

Type IDC-...-1K tags can be programmed to read 24-bit information (so-called **special read only code)** very quickly. This is useful for detecting containers in automated warehouses.

Length of the **special read only code**:

- Tag of the type IDC-...-1K: 48 bit
- Tag of the type IUC: 96 ... 240 bit

To write the **special read only code** use the commands **SP** and **EP**; to read it out, use the commands **SS** and **ES**.

If **SP** or **EP** is used to write to an IDC-...-1K tag, the tag is then locked. If you wish to write to the tag again using standard commands, unlock it using the command **SI**.

Single read special fixcode (SS)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (0Ah)	0	0	0	0	1	0	1	0
Byte 3	FixLen/Ident channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	not used	-	-	-	-	-	-	-	-
Byte 5	not used	-	-	-	-	-	-	-	-
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	1	0	0
Byte 2	Command code (0Ah)	0	0	0	0	1	0	1	0
Byte 3	Reserved/Ident channel/Toggle bit	-	-	-	-	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	ID code 00h ... FFh	<ID code>							
Byte 7	ID code 00h ... FFh	<ID code>							
Byte 8	ID code 00h ... FFh	<ID code>							
Byte 9	ID code 00h ... FFh	<ID code>							
Byte 10	ID code 00h ... FFh	<ID code>							
byte 11	ID code 00h ... FFh	<ID code>							

Table 7.28 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head makes only one attempt to read a **special read only code**.

Note!

The <FixLen> of IDC-...-1K read/write tags is always 6 bytes.

2014-02

Enhanced read special fixcode (ES)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (71h)	0	1	1	1	0	0	0	1
Byte 3	WordNum/Ident channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	not used	-	-	-	-	-	-	-	-
Byte 5	not used	-	-	-	-	-	-	-	-
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	1
Byte 2	Command code (71h)	0	1	1	1	0	0	0	1
Byte 3	Reserved/Ident channel/Toggle bit	-	-	-	-	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	ID code 00h ... FFh	<ID code>							
Byte 7	ID code 00h ... FFh	<ID code>							
Byte 8	ID code 00h ... FFh	<ID code>							
Byte 9	ID code 00h ... FFh	<ID code>							
Byte 10	ID code 00h ... FFh	<ID code>							
byte 11	ID code 00h ... FFh	<ID code>							

Table 7.29 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head attempts to read a **special read only code** until successful. Only data that changes is transferred via the interface, i.e. the R/W head transfers data whenever it reads a new data carrier or whenever it reads a data carrier where there was previously no R/W head within the detection range.

The status '05h' (read command) is output whenever a data carrier leaves the detection range.

Note!

The <FixLen> of IDC-...-1K read/write tags is always 6 bytes.

Single program special fixcode (SP)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (0Dh)	0	0	0	0	1	1	0	1
Byte 3	Word number/Ident channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Reserved	0	0	0	0	0	0	0	0
Byte 6	ID code 00h ... FFh	<ID code>							
Byte 7	ID code 00h ... FFh	<ID code>							
Byte 8	ID code 00h ... FFh	<ID code>							
Byte 9	ID code 00h ... FFh	<ID code>							
Byte 10	ID code 00h ... FFh	<ID code>							
byte 11	ID code 00h ... FFh	<ID code>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (0Dh)	0	0	0	0	1	1	0	1
Byte 3	FixLen/Ident channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Table 7.30 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head makes only one attempt to write a **special read only code**.

Note!

The <FixLen> of IDC-...-1K read/write tags is always 6 bytes.

Enhanced program special fixcode (EP)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (75h)	0	1	1	1	0	1	0	1
Byte 3	FixLen/Ident channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	Reserved	0	0	0	0	0	0	0	0
Byte 5	Reserved	0	0	0	0	0	0	0	0
Byte 6	ID code 00h ... FFh	<ID code>							
Byte 7	ID code 00h ... FFh	<ID code>							
Byte 8	ID code 00h ... FFh	<ID code>							
Byte 9	ID code 00h ... FFh	<ID code>							
Byte 10	ID code 00h ... FFh	<ID code>							
byte 11	ID code 00h ... FFh	<ID code>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	0
Byte 2	Command code (75h)	0	1	1	1	0	1	0	1
Byte 3	FixLen/Ident channel/Toggle bit	<FixLen>				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Table 7.31 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

The R/W head attempts to write a **special read only code** until successful. After each successful write, the head sends the response and then switches to continuous reading. Then the R/W head reads the same data carrier until it has left the detection range or a new data carrier appears within the detection range. The command then starts again with write attempts.

The status '05h' (read/write command) is output when the data carrier leaves the detection range or if the data carrier is not yet within the detection range when the command is executed.

If two data carriers enter the read range one immediately after the other, the status '05h' is not issued between the two readings.

Note!

The <FixLen> of IDC-...-1K read/write tags is always 6 bytes.

Initialize tag (SI)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	0	1	0	0
Byte 2	Command code (6Bh)	0	1	1	0	1	0	1	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	0	0	0	0	0	0	0	0
Byte 1*	Telegram length, low byte	0	0	0	0	1	0	1	1
Byte 2	Command code (6Bh)	0	1	1	0	1	0	1	1
Byte 3	Reserved/Channel/Toggle bit	0	0	0	0	<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-
Byte 10	not used	-	-	-	-	-	-	-	-

Table 7.32 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

This command allows conventional reading and writing of IDC-...-1K read/write tags that were configured using the **EP** or **SP** commands.

Extended Commands for Type IDC-...-1K and IQC-... Read/Write Tags

Single write words with lock (SL)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (47h)	0	1	0	0	0	1	1	1
Byte 3	Word number/Ident channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Word address	<WordAddr> (high byte)							
Byte 5	Word address	<WordAddr> (low byte)							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
...	Data 00h ... FFh	<Data>							
Byte N[1)]	Data 00h ... FFh	<Data>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0	Telegram length, high byte	<TelegramLenH>							
Byte 1	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (47h)	0	1	0	0	0	1	1	1
Byte 3	Word number/Ident channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Table 7.33 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

1) N = 4 x <WordNum> + 3

This command is the same as a normal write command. The data is write-protected at the end of the write process, provided the tags offer this function. This applies for 13.56 MHz tags of the type 21, 22, 24, and 33 as well as for LF tags IDC-...-1K. Write protection is only activated for memory blocks involved in the write process. Data can continue to be written to all other memory blocks.

The read/write head makes one attempt to write <WordNum> 32-bit words from the address <WordAddr>.

Enhanced write words with lock (EL)

Command:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (48h)	0	1	0	0	1	0	0	0
Byte 3	Word number/Ident channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Word address	<WordAddr> (high byte)							
Byte 5	Word address	<WordAddr> (low byte)							
Byte 6	Data 00h ... FFh	<Data>							
Byte 7	Data 00h ... FFh	<Data>							
...	Data 00h ... FFh	<Data>							
Byte N[1)]	Data 00h ... FFh	<Data>							

Response:

Byte	Type	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length, high byte	<TelegramLenH>							
Byte 1*	Telegram length, low byte	<TelegramLenL>							
Byte 2	Command code (48h)	0	1	0	0	1	0	0	0
Byte 3	Word number/Ident channel/Toggle bit	<WordNum>				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
Byte 6	not used	-	-	-	-	-	-	-	-
Byte 7	not used	-	-	-	-	-	-	-	-
Byte 8	not used	-	-	-	-	-	-	-	-
Byte 9	not used	-	-	-	-	-	-	-	-

Table 7.34 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) 4 x <WordNum> + 6

This command is the same as a normal write command. The data is write-protected at the end of the write process, provided the tags offer this function. This applies for 13.56 MHz tags of the type 21, 22, 24, and 33 as well as for LF tags IDC-...-1K. Write protection is only activated for memory blocks involved in the write process. Data can continue to be written to all other memory blocks.

The read/write head repeatedly attempts to write <WordNum> 32-bit words from the address <WordAddr> until successful. After each successful write, the head sends the response and then switches to continuous reading. Then the read/write head reads the same tag until it has left the detection range or a new tag appears within the detection range. The command then starts again with write attempts.

The status '05h' is only output when a tag leaves the detection range or is not yet within the detection range. If two tags enter the read range one immediately after the other, the status '05' is not issued between the two readings.

2014-02

Extended commands for IQH2-... and IUH-... read/write heads

Note!

A detailed description and further commands can be found in the manual for the read/write head.

read param (RP)

Command:

Byte	Contents	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length 00 ... FFh	<TelegramLen> (high byte)							
Byte 1*	Telegram length 00 ... FFh	<TelegramLen> (low byte)							
Byte 2	Command code (BEh)	1	0	1	1	1	1	1	0
Byte 3	Reserved/Ident channel/Toggle bit	-				<Channel>			<T>
Byte 4	System code	<SystemCode>							
Byte 5	Parameter type	<ParamTyp> (high byte)							
Byte 6	Parameter type	<ParamTyp> (low byte)							
Byte 7	not used	-							
Byte 8	not used	-							

Table 7.35 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

Response:

Byte	Contents	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length 00 ... FFh	<TelegramLen> (high byte)							
Byte 1*	Telegram length 00 ... FFh	<TelegramLen> (low byte)							
Byte 2	Command code (BEh)	1	0	1	1	1	1	1	0
Byte 3	Reserved/Ident channel/Toggle bit	0				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
...	Data 00 ... FFh	<Data>							
...	Data 00 ... FFh	<Data>							
...	Data 00 ... FFh	<Data>							
Byte N[1)]	Data 00 ... FFh	<Data>							

Table 7.36 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

write param (WP)

Command:

Byte	Contents	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0	Telegram length 00 ... FFh	<TelegramLen> (high byte)							
Byte 1	Telegram length 00 ... FFh	<TelegramLen> (low byte)							
Byte 2	Command code (BFh)	1	0	1	1	1	1	1	1
Byte 3	Reserved/Ident channel/Toggle bit	-				<Channel>			<T>
Byte 4	reserved	0							
Byte 5	System code	<SystemCode>							
Byte 6	Parameter type	<ParamTyp> (high byte)							
Byte 7	Parameter type	<ParamTyp> (low byte)							
Byte 8	Length 00 … FFh	<DataLength> (high byte)							
Byte 9	Length 00 … FFh	<DataLength> (low byte)							
...	Data 00 ... FFh	<Data>							
...	Data 00 ... FFh	<Data>							
Byte N[1)]	Data 00 ... FFh	<Data>							

Table 7.37 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.
1) N = <DataLength> + 8

Response:

Byte	Contents	Bit no.							
		7	6	5	4	3	2	1	0
Byte 0*	Telegram length 00 ... FFh	<TelegramLen> (high byte)							
Byte 1*	Telegram length 00 ... FFh	<TelegramLen> (low byte)							
Byte 2	Command code (BFh)	1	0	1	1	1	1	1	1
Byte 3	Reserved/Ident channel/Toggle bit	0				<Channel>			<T>
Byte 4	Status	<Status>							
Byte 5	Reply counter	<ReplyCounter>							
...	not used	-	-	-	-	-	-	-	-
...	not used	-	-	-	-	-	-	-	-
...	not used	-	-	-	-	-	-	-	-
Byte N[1)]	not used	-	-	-	-	-	-	-	-

Table 7.38 * this byte is only used with the TCP/IP and MODBUS TCP/IP protocol.

Note!

Toggle bit

If you send two commands with the same SystemCode and same ParamTyp in succession on the bus interface, you must change the toggle bit in the second command in order for the node to detect the command.

2014-02

7.16 Legend

<BatteryConditon 1>	:	1 byte, first digit of battery status (percentage, decimal, ASCII encoded). 30h, 31h
<BatteryConditon 2>	:	1 byte, second digit of battery status (percentage, decimal, ASCII encoded). 30h, 39h
<BatteryConditon 3>	:	1 byte, third digit of battery status (percentage, decimal, ASCII encoded). 30h, 39h
<ByteNum>	:	4 bits, length of <IDCode>; System MV: 4 characters (04h) System IQ: 8 characters (08h)
<Channel>	:	3 bits, channel of the read/write head Channel 1 (001b), channel 2 (010b), all channels (111b)
<ConfAddr>	:	1 ASCII character, word starting address in configuration range of data carrier. The following applies for IPC03: 01h = Protection Word 02h = Control Word
<Data>	:	<WordNum> times 4 bytes. When communicating a word, the highest value byte is transferred first and the lowest value byte, or bytes, last.
<F>	:	1 bit, multiplex mode, 0 (0b): Mode off, 1 (1b): Mode on
<Fill Sign>	:	1 ASCII character
<FixLen>	:	4 bits, length of the read only code in bytes,
<FixType>	:	2 ASCII characters, for example: '02' for IPC02
<IDCode>	:	4 bytes, 6 bytes or 8 bytes (depending on the data carrier type)
<Identchannel>	:	3 bits, channel of the read/write head 0(0000b), 1 (001b), 2 (010b), all channels (111b) (but not <Sensorchannel> in trigger mode)
<Length>	:	2 $ASCII_{hex}$ characters = number of data bytes When writing the UII segment + 1: Range "03", "05", "07" ... (read) "00" (write)
<Month>	:	2 bytes ASCII, hexadecimal encoding, 01 ... 0C (01=January, 0C=December)
<P>	:	1 bit, password mode, 0 (0b): Mode off, 1 (1b): Mode on
<ParamTyp>	:	Parameter type, 2 ASCII characters
<PSW>	:	4 bytes HEX, password
<ReplyCounter>	:	1 byte, increases by 1 after each response and confirmation. The reply counter starts from 0 after the system is switched on. When the maximum value is reached, the counter skips the value 0 (from 255 to 1).
<Sensorchannel>	:	3 bits, channel 1 (001b) or 2 (010b)
<Status>	:	1 byte (see chapter 7.17)
<SystemCode>	:	= "U"
<T>	:	1 bit, toggle bit
<TagType>	:	2 ASCII characters, for example: '02' for IPC02
<TelegramLenH>	:	1 byte, high byte of the 16-bit telegram = (N+1) div 256
<TelegramLenL>	:	1 byte, low byte of the 16-bit telegram = (N+1) mod 256

<Triggermode> : 8 bits
0 (00000000b): Trigger mode off
1 (00000001b): Trigger mode on
2 (00000010b): Trigger mode inverted

<WordAddr> : 2 bytes, word start address in the data carrier, range from 0000h to FFFFh, depending on data carrier type.

<WordNum> : 4 bits, number of words to be read or written, range from 0h to Fh, depending on data carrier type.
The following applies for IPC03: The word count 0h is used with the word address 0000h to read the preset data range on the data carrier ("Default Read").
The following applies for IQC33: The word count parameter must be even-numbered because of the block size of 8 bytes. The word address then indicates the offset in 8-byte increments.

<Year> : 2 bytes ASCII, hexadecimal encoding, 00h ... 63h

7.17 Fault/Status messages

Status	Meaning
00h	The command has been executed without error.
FFh	The command is processing.

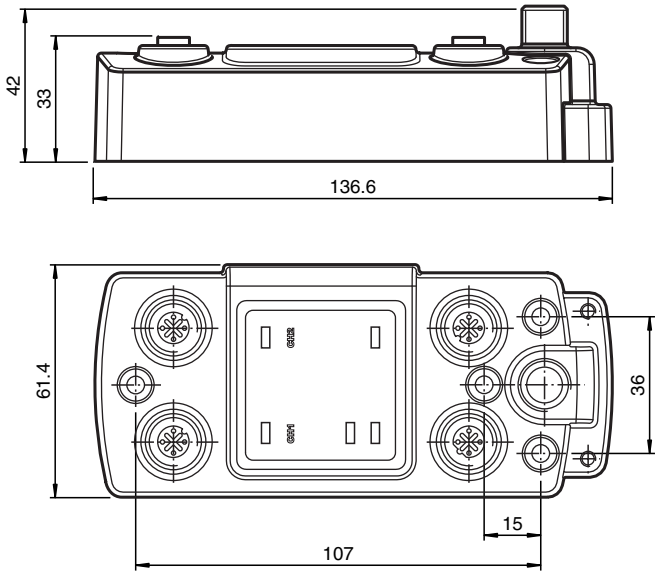
Error messages which triggered the identification system

Status	Meaning
01h	The battery of the read/write tag is weak.
02h	Reserved
03h	Reserved
04h	Incorrect or incomplete command or parameter not in the valid range.
05h	No data carrier in the detection range.
06h	Hardware error, e.g. error during self-test or R/W head defect.
07h	Internal device error.
08h	Reserved
09h	The parameterized tag type is not compatible with the connected reading head.
0Ah	Several tags in the detection range (UHF).
0Bh	Reserved
0Ch	Reserved
0Dh	Reserved
0Eh	Internal buffer overflow.
0Fh	Reserved

2014-02

8 Technical Specifications

8.1 Dimensions



8.2 Technical Data

General Data

Number of read/write heads	Max. 2 alternatively 1 read/write head and 1 trigger sensor
UL file number	E87056

Displays/Controls

Link/Traffic LED	Green: connection to the network Yellow: flashes at the same speed as the data being sent
LEDs 1, 2	Status display for read/write heads Green: Command to read/write head active Yellow: approx. 1 second if command is executed successfully
LEDs CH1, CH2	Green: reader identified Red: configuration error
PWR/ERR LED	Green: power on Yellow: system starting Red: PROFINET bus error
Rotary switch	Address setting

Electrical Data

Rated operational voltage	20–30 V DC, PELV
Ripple	≤ 10 % at 30 V DC
Current consumption	≤ 4 A incl. read/write heads
Power consumption	3.5 W without read/write heads
Galvanic isolation	Basic insulation in accordance with DIN EN 50178, rated insulation voltage 50 V_{eff}

Interface 1

Physical	Ethernet
Protocol	SMTP HTTP TCP/IP (port 10000) MODBUS/TCP EtherNet/IP PROFINET IO
Transmission rate	10 Mbit/s or 100 Mbit/s

Interface 2

Physical	Ethernet
Protocol	SMTP HTTP TCP/IP (port 10000) MODBUS/TCP EtherNet/IP PROFINET IO
Transmission rate	10 Mbit/s or 100 Mbit/s

Conformity with Standards and Directives

Directive conformity	
EMC Directive 2004/108/EC	EN 61000-6-2:2006, EN 61000-6-4:2007
Standard conformity	
Degree of protection	IEC 60529:2001

Ambient Conditions

Ambient temperature	-25–70 °C (-13–158 °F)
Storage temperature	-40–85 °C (-40–185 °F)
Climatic conditions	Max. humidity 96% Salt spray resistant in accordance with EN 60068-2-52
Shock and impact resistance	Oscillation (sine): 5 g, 10–1000 Hz in accordance with EN 60068-2-6 Shock (half-sine): 30 g, 11 ms in accordance with EN 60068-2-27

Mechanical Data

Degree of protection	IP67
Connection	Read/write heads: Shielded, 4-pin, M12 connector Power supply: M12 connector Protective ground: M4 grounding screw Ethernet: M12 connector
Enclosure material	Zinc, powder-coated
Mounting	Screw mounting
Weight	Approx. 500 g

2014-02

9 Troubleshooting

Source of fault	Possible cause	Remedy
The PWR/ERR LED does not illuminate.	Power supply not guaranteed.	Ensure the power supply using a 24 V DC source.
The PWR/ERR LED illuminates yellow.	The device is still booting up.	Wait until the booting process is complete.
The PWR/ERR LED flashes green. (For Ethernet/IP communication only)	Memory overflow.	Reduce the data hold time (see chapter 7.8.8).
The PWR/ERR LED illuminates red. (For PROFINET communication only)	No physical connection to the master, e.g. due to lead breakage.	Check the lead and repair if necessary.
	The assigned device name is incorrect	Configure the same device names in the master and in the device.
The Link/Act LED does not illuminate.	No physical network connection, e.g. due to lead breakage.	Check the lead and repair if necessary.
The CHx display does not appear even though the read/write head is connected to port 1.	The lead is faulty or not connected correctly.	Check the lead and repair if necessary.
	The read/write head is faulty.	Check the read/write head and repair if necessary.
The LEDs in the reader and the CHx display on the IDENTControl display are flashing.	The reader connected does not support the tag type set.	Select a tag type that is supported by the reader.
A read command (e.g. **SR ...**) gives the status 4 even though the syntax is correct.	An incorrect tag type (e.g. IPC02) is set for the relevant channel. The read commands only function with read/write tags and not with read only tags.	Set the correct tag type (e.g. IPC03) or "Autodetect" using the **CT...** command.
The **SG** or **EG** (get configuration) command gives the status 4 even though the syntax is correct.	IPC03 is not set for the relevant channel. The configuration commands only function if the read/write tag IPC03 is set and not in Autodetect mode.	Set the tag type IPC03 using the **CT...** command.
The website cannot be displayed.	The IP address is incorrect.	**With the DHCP server**: Make sure that the IP address entered in the browser corresponds with the address assigned to the device. **Without DHCP server**: Check the IP address entered in the browser to see whether it corresponds with the fixed IP address set in the device. Check the subnet mask in both cases.

Table 9.1 This table will be updated as necessary. For the latest manual, visit: www.pepperl-fuchs.com

10 ASCII table

hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII	hex	dec	ASCII
00	0	NUL	**20**	**32**	**Space**	40	64	@	**60**	**96**	'
01	1	SOH	**21**	**33**	**!**	41	65	A	**61**	**97**	**a**
02	2	STX	**22**	**34**	**"**	42	66	B	**62**	**98**	**b**
03	3	ETX	**23**	**35**	**#**	43	67	C	**63**	**99**	**c**
04	4	EOT	**24**	**36**	**$**	44	68	D	**64**	**100**	**d**
05	5	ENQ	**25**	**37**	**%**	45	69	E	**65**	**101**	**e**
06	6	ACK	**26**	**38**	**&**	46	70	F	**66**	**102**	**f**
07	7	BEL	**27**	**39**	**'**	47	71	G	**67**	**103**	**g**
08	8	BS	**28**	**40**	**(**	48	72	H	**68**	**104**	**h**
09	9	HT	**29**	**41**	**)**	49	73	I	**69**	**105**	**I**
0A	10	LF	**2A**	**42**	*****	4A	74	J	**6A**	**106**	**j**
0B	11	VT	**2B**	**43**	**+**	4B	75	K	**6B**	**107**	**k**
0C	12	FF	**2C**	**44**	**,**	4C	76	L	**6C**	**108**	**l**
0D	13	CR	**2D**	**45**	**-**	4D	77	M	**6D**	**109**	**m**
0E	14	SO	**2E**	**46**	**.**	4E	78	N	**6E**	**110**	**n**
0F	15	SI	**2F**	**47**	**/**	4F	79	O	**6F**	**111**	**o**
10	16	DLE	**30**	**48**	**0**	50	80	P	**70**	**112**	**p**
11	17	DC1	**31**	**49**	**1**	51	81	Q	**71**	**113**	**q**
12	18	DC2	**32**	**50**	**2**	52	82	R	**72**	**114**	**r**
13	19	DC3	**33**	**51**	**3**	53	83	S	**73**	**115**	**s**
14	20	DC4	**34**	**52**	**4**	54	84	T	**74**	**116**	**t**
15	21	NAK	**35**	**53**	**5**	55	85	U	**75**	**117**	**u**
16	22	SYN	**36**	**54**	**6**	56	86	V	**76**	**118**	**v**
17	23	ETB	**37**	**55**	**7**	57	87	W	**77**	**119**	**w**
18	24	CAN	**38**	**56**	**8**	58	88	X	**78**	**120**	**x**
19	25	EM	**39**	**57**	**9**	59	89	Y	**79**	**121**	**y**
1A	26	SUB	**3A**	**58**	**:**	5A	90	Z	**7A**	**122**	**z**
1B	27	ESC	**3B**	**59**	**;**	5B	91	[	**7B**	**123**	**{**
1C	28	FS	**3C**	**60**	**<**	5C	92	\	**7C**	**124**	**|**
1D	29	GS	**3D**	**61**	**=**	5D	93	]	**7D**	**125**	**}**
1E	30	RS	**3E**	**62**	**>**	5E	94	^	**7E**	**126**	**~**
1F	31	US	**3F**	**63**	**?**	5F	95	_	**7F**	**127**	**DEL**

2014-02

11 Appendix A

11.1 Example 1

Assembly objects 101d/151d (mixed mode) are used in the example. The following functions are activated:

- Set tag type IPC02 on channel 1 and channel 2.
- Read read only code from IPC02 tag.
- Implicit communication.

Prerequisites:

- One type IPH-XX read/write head is connected to channel 1 and channel 2 on the IDENTControl Compact.
- One type IPC02 tag is located in front of each read/write head.
- The IP address of the IDENTControl Compact is set to a free address.
- The device is connected to the network.

Setting connection parameters

These parameters are configured on the PLC:

Assembly instance	Size (32 bits)
Input: 151	3
Output: 101	3
Configuration: 112 (this value is used for all input/output instances)	0 (this value is used for all input/output instances)

Setting tag type IPC02 on channel 1 and channel 2

Send the command change tag to channel 1 as an implicit command:

Byte no.	Use	Type	Description
Byte 0	Command code	04h	Command **CT** (change tag)
Byte 1	Channel/Toggle bit	02h	Channel = 1 Toggle bit = 0
Byte 2	Tag type (high byte)	30h	IPC 02
Byte 3	Tag type (low byte)	32h	IPC 02
Byte 4 … Byte 11		00h	

Confirmation:

Byte no.	Use	Type	Description
Byte 0	Command code	04h	Command **CT** (change tag type)
Byte 1	Number of double words/ Channel/Toggle bit	02h	Channel = 1 Toggle bit = 0
Byte 2	Status	FFh	Processing command.
Byte 3	Reply counter	01h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4 … Byte 11		00h	

Response:

Byte no.	Use	Type	Description
Byte 0	Command code	04h	Command **CT** (change tag)
Byte 1	Number of double words/ Channel/Toggle bit	02h	Channel = 1 Toggle bit = 0
Byte 2	Status	00h	Command executed.
Byte 3	Reply counter	02h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4 ... Byte 11		00h	

Send the command change tag to channel 2 as an implicit command:

Byte no.	Use	Type	Description
Byte 0	Command code	04h	Command **CT** (change tag)
Byte 1	Channel/Toggle bit	04h	Channel = 2 Toggle bit = 0
Byte 2	Tag type (high byte)	30h	IPC 02
Byte 3	Tag type (low byte)	32h	IPC 02
Byte 4 ... Byte 11		00h	

Confirmation:

Byte no.	Use	Type	Description
Byte 0	Command code	04h	Command **CT** (change tag)
Byte 1	Number of double words/ Channel/Toggle bit	04h	Channel = 2 Toggle bit = 0
Byte 2	Status	FFh	Processing command.
Byte 3	Reply counter	03h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4 ... Byte 11		00h	

Response:

Byte no.	Use	Type	Description
Byte 0	Command code	04h	Command **CT** (change tag)
Byte 1	Number of double words/ Channel/Toggle bit	04h	Channel = 2 Toggle bit = 0
Byte 2	Status	00h	Command executed.
Byte 3	Reply counter	04h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4 ... Byte 11		00h	

2014-02

Note!

Alternatively select the tag type by configuring the relevant setting in the boot-up object.

Reading tags of the read/write heads on channel 1 and channel 2.

Send the command single read to channel 1 as an implicit command:

Byte no.	Use	Type	Description
Byte 0	Command code	01h	Command **SF** (single read fixcode)
Byte 1	Channel/Toggle bit	02h	Channel = 1 Toggle bit = 0
Byte 2		00h	
Byte 3		00h	
Byte 4		00h	
Byte 5		00h	
Byte 6		00h	
Byte 7		00h	
Byte 8		00h	
Byte 9		00h	
Byte 10		00h	
byte 11		00h	

Confirmation

Byte no.	Use	Type	Description
Byte 0	Command code	01h	Command **SF** (single read fixcode)
Byte 1	Channel/Toggle bit	02h	Channel = 1 Toggle bit = 0
Byte 2	Status	FFh	Processing command.
Byte 3	Reply counter	05h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4		00h	
Byte 5		00h	
Byte 6		00h	
Byte 7		00h	
Byte 8		00h	
Byte 9		00h	
Byte 10		00h	
byte 11		00h	

Response

Byte no.	Use	Type	Description
Byte 0	Command code	01h	Command **SF** (single read fixcode)
Byte 1	Channel/Toggle bit	02h	Channel = 1 Toggle bit = 0
Byte 2	Status	00h	Command executed.

Byte no.	Use	Type	Description
Byte 3	Reply counter	06h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4	ID code 00h ... FFh	-	<ID code>
Byte 5	ID code 00h ... FFh	-	<ID code>
Byte 6	ID code 00h ... FFh	-	<ID code>
Byte 7	ID code 00h ... FFh	-	<ID code>
Byte 8	ID code 00h ... FFh	-	<ID code>[1)]
Byte 9		00h	
Byte 10		00h	
byte 11		00h	

Table 11.1 1) only IPC02 and IPC11

Send the command single read to channel 2 as an implicit command:

Byte no.	Use	Type	Description
Byte 0	Command code	01h	Command **SF** (single read fixcode)
Byte 1	Channel/Toggle bit	04h	Channel = 2 Toggle bit = 0
Byte 2		00h	
Byte 3		00h	
Byte 4		00h	
Byte 5		00h	
Byte 6		00h	
Byte 7		00h	
Byte 8		00h	
Byte 9		00h	
Byte 10		00h	
byte 11		00h	

Confirmation:

Byte no.	Use	Type	Description
Byte 0	Command code	01h	Command **SF** (single read fixcode)
Byte 1	Channel/Toggle bit	04h	Channel = 2 Toggle bit = 0
Byte 2	Status	FFh	Processing command.
Byte 3	Reply counter	07h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4		00h	
Byte 5		00h	
Byte 6		00h	
Byte 7		00h	
Byte 8		00h	
Byte 9		00h	

2014-02

Byte no.	Use	Type	Description
Byte 10		00h	
byte 11		00h	

Response:

Byte no.	Use	Type	Description
Byte 0	Command code	01h	Command **SF** (single read fixcode)
Byte 1	Channel/Toggle bit	04h	Channel = 2 Toggle bit = 0
Byte 2	Status	00h	Command executed.
Byte 3	Reply counter	08h	For every ident telegram, the value on the reply counter increases by 1.
Byte 4	ID code 00h ... FFh	-	<ID code>
Byte 5	ID code 00h ... FFh	-	<ID code>
Byte 6	ID code 00h ... FFh	-	<ID code>
Byte 7	ID code 00h ... FFh	-	<ID code>
Byte 8	ID code 00h ... FFh	-	<ID code>[1)]
Byte 9		00h	
Byte 10		00h	
byte 11		00h	

Table 11.2 1) only IPC02 and IPC11

11.2 Example 2

Assembly objects 104d/154d (separated mode) are used in the example. The following functions are activated:

- Set tag type IPC03 on channel 1 and channel 2.
- Write data to a IPC03 tag.
- Read data from a IPC03 tag.

Prerequisites:

- One type IPH-XX read/write head is connected to channel 1 and channel 2 on the IDENTControl Compact.
- One type IPC03 tag is located in front of each read/write head.
- The IP address of the IDENTControl is set to a free address.
- The device is connected to the network.

Setting connection parameters

These parameters are configured on the PLC:

Assembly instance	Size (32 bits)
Input: 154	8
Output: 104	8
Configuration: 112 (this value is used for all input/output instances)	0 (this value is used for all input/output instances)

Division of the input and output instance from the assembly object:

Output instance 104d - 32 bytes

Bytes	Class, instance, attribute	Description
0 - 7	64h, 01d, 01h	Channel 1 [8]
8 - 15	64h, 02d, 01h	Channel 2 [8]
16 - 23	64h, 03d, 01h	Reserved
24 - 31	64h, 04d, 01h	Reserved

Input instance 154d - 32 bytes

Bytes	Class, instance, attribute	Description
0 - 7	65h, 01d, 01h	Channel 1 [8]
8 - 15	65h, 02d, 01h	Channel 2 [8]
16 - 23	65h, 03d, 01h	Reserved
24 - 31	65h, 04d, 01h	Reserved

The IDENTControl Compact requires 32 bytes of input data and sends back 32 bytes of output data.

Setting tag type IPC03 on channels 1 and 2

Send the command change tag as an implicit command:

Element	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	04h	Command **CT** (change tag)
	Byte 1	Channel/Toggle bit	00h	The element defines the channel. Toggle bit = 0
	Byte 2	Tag type (high byte)	30h	IPC 03
	Byte 3	Tag type (low byte)	33h	IPC 03
	Byte 4		00h	Not used
	Byte 5		00h	Not used
	Byte 6		00h	Not used
	Byte 7		00h	Not used
2	Byte 8	Command code	04h	Command **CT** (change tag)
	Byte 9	Channel/Toggle bit	00h	The element defined the channel. Toggle bit = 0
	Byte 10	Tag type (high byte)	30h	IPC 03
	byte 11	Tag type (low byte)	33h	IPC 03
	byte 12		00h	Not used
	Byte 13		00h	Not used
	Byte 14		00h	Not used
	Byte 15		00h	Not used
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Confirmation:

Element	Implicit telegram	Use	Type	Description
1	Byte 0	Command code (Echo)	04h	Command **CT** (change tag)
	Byte 1	Channel/Toggle bit	02h	Channel = element = 1 Toggle bit = 0
	Byte 2	Status	FFh	Processing command.
	Byte 3	Reply counter	01h	For every ident telegram, the value on the reply counter increases by 1.
	Byte 4		00h	
	Byte 5		00h	
	Byte 6		00h	
	Byte 7		00h	

Ele-ment	Implicit telegram	Use	Type	Description
2	Byte 8	Command code (Echo)	04h	Command **CT** (change tag)
	Byte 9	Channel/Toggle bit	04h	Channel = element = 2 Toggle bit = 0
	Byte 10	Status	FFh	Processing command.
	byte 11	Reply counter	01h	For every ident telegram, the value on the reply counter increases by 1.
	byte 12		00h	Not used
	Byte 13		00h	Not used
	Byte 14		00h	Not used
	Byte 15		00h	Not used
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Response:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	04h	Command **CT** (change tag)
	Byte 1	Number of double words/ Channel/Toggle bit	02h	Channel = element = 1 Toggle bit = 0
	Byte 2	Status	00h	Command executed.
	Byte 3	Reply counter	02h	For every ident telegram, the value on the reply counter increases by 1.
	Byte 4		00h	
	Byte 5		00h	
	Byte 6		00h	
	Byte 7		00h	
2	Byte 8	Command code	04h	Command **CT** (change tag)
	Byte 9	Number of double words/ Channel/Toggle bit	04h	Channel = element = 2 Toggle bit = 0
	Byte 10	Status	00h	Processing command.
	byte 11	Reply counter	02h	For every ident telegram, the value on the reply counter increases by 1.
	byte 12		00h	Not used
	Byte 13		00h	Not used
	Byte 14		00h	Not used
	Byte 15		00h	Not used

2014-02

Ele-ment	Implicit telegram	Use	Type	Description
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Sending the write command as an implicit command

Send the single write words command:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	40h	Command **SW** (single write words)
	Byte 1	Number of double words/ Channel/Toggle bit	10h	1 double word = 4 bytes Element defines the channel. Toggle bit = 0
	Byte 2	Address of double word (high byte)	00h	Start address = 0
	Byte 3	Address of double word (low byte)	00h	Start address = 0
	Byte 4	Data	00h	1. Byte data
	Byte 5	Data	01h	2. Byte data
	Byte 6	Data	02h	3. Byte data
	Byte 7	Data	03h	4. Byte data
2	Byte 8	Command code	40h	Command **SW** (single write words)
	Byte 9	Number of double words/ Channel/Toggle bit	10h	1 double word = 4 bytes Element defines the channel. Toggle bit = 0
	Byte 10	Address of double word (high byte)	00h	Start address = 0
	byte 11	Address of double word (low byte)	00h	Start address = 0
	byte 12	Data	10h	1. Byte data
	Byte 13	Data	11h	2. Byte data
	Byte 14	Data	12h	3. Byte data
	Byte 15	Data	13h	4. Byte data
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Confirmation:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	40h	Command **SW** (single write words)
	Byte 1	Number of double words/ Channel/Toggle bit	12h	1 double word = 4 bytes Channel = element = 1 Toggle bit = 0
	Byte 2	Status	FFh	Processing command.
	Byte 3	Reply counter	03h	For every ident telegram, the value on the reply counter increases by 1.
	Byte 4	-	00h	
	Byte 5	-	00h	
	Byte 6	-	00h	
	Byte 7	-	00h	
2	Byte 8	Command code	40h	Command **SW** (single write words)
	Byte 9	Number of double words/ Channel/Toggle bit	14h	1 double word = 4 bytes Channel = element = 1 Toggle bit = 0
	Byte 10	Status	FFh	Processing command.
	byte 11	Reply counter	03h	For every ident telegram, the value on the reply counter increases by 1.
	byte 12	-	00h	
	Byte 13	-	00h	
	Byte 14	-	00h	
	Byte 15	-	00h	
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

2014-02

Response:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	40h	Command **SW** (single write words)
	Byte 1	Number of double words/ Channel/Toggle bit	02h	Element defines channel. Toggle bit = 0
	Byte 2	Status	00h	Command executed.
	Byte 3	Reply counter	04h	For every ident telegram, the value on the reply counter increases by 1.
	Byte 4	-	00h	
	Byte 5	-	00h	
	Byte 6	-	00h	
	Byte 7	-	00h	
2	Byte 8	Command code	40h	Command **SW** (single write words)
	Byte 9	Number of double words/ Channel/Toggle bit	04h	Element defines channel. Toggle bit = 0
	Byte 10	Status	00h	Command executed.
	byte 11	Reply counter	04h	For every ident telegram, the value on the reply counter increases by 1.
	byte 12	-	00h	
	Byte 13	-	00h	
	Byte 14	-	00h	
	Byte 15	-	00h	
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Sending the read command as an implicit command

Send the single read words command:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	10h	Command **SR** (single read words)
	Byte 1	Number of double words/ Channel/Toggle bit	10h	1 double word = 4 bytes Element defines the channel. Toggle bit = 0
	Byte 2	Address of double word (high byte)	00h	Start address = 0
	Byte 3	Address of double word (low byte)	00h	Start address = 0
	Byte 4		00h	
	Byte 5		00h	
	Byte 6		00h	
	Byte 7		00h	
2	Byte 8	Command code	10h	Command **SR** (single read words)
	Byte 9	Number of double words/ Channel/Toggle bit	10h	1 double word = 4 bytes Element defines the channel. Toggle bit = 0
	Byte 10	Address of double word (high byte)	00h	Start address = 0
	byte 11	Address of double word (low byte)	00h	Start address = 0
	byte 12		00h	
	Byte 13		00h	
	Byte 14		00h	
	Byte 15		00h	
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Confirmation:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	10h	Command **SR** (single read words)
	Byte 1	Number of double words/ Channel/Toggle bit	12h	1 double word = 4 bytes Channel = element = 1 Toggle bit = 0
	Byte 2	Status	FFh	Processing command.
	Byte 3	Reply counter	05h	For every ident telegram, the value on the reply counter increases by 1.
	Byte 4	-	00h	
	Byte 5	-	00h	
	Byte 6	-	00h	
	Byte 7	-	00h	
2	Byte 8	Command code	10h	Command **SR** (single read words)
	Byte 9	Number of double words/ Channel/Toggle bit	14h	1 double word = 4 bytes Channel = element = 1 Toggle bit = 0
	Byte 10	Status	FFh	Processing command.
	byte 11	Reply counter	05h	For every ident telegram, the value on the reply counter increases by 1.
	byte 12	-	00h	
	Byte 13	-	00h	
	Byte 14	-	00h	
	Byte 15	-	00h	
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

Response:

Ele-ment	Implicit telegram	Use	Type	Description
1	Byte 0	Command code	10h	Command **SR** (single read words)
	Byte 1	Number of double words/ Channel/Toggle bit	12h	1 double word = 4 bytes Channel = element = 1 Toggle bit = 0
	Byte 2	Status	00h	Command executed.
	Byte 3	Reply counter	06h	For every ident telegram, the value on the reply counter increases by 1.
	Byte 4	Data	00h	1. Byte data
	Byte 5	Data	01h	2. Byte data
	Byte 6	Data	02h	3. Byte data
	Byte 7	Data	03h	4. Byte data
2	Byte 8	Command code	10h	Command **SR** (single read words)
	Byte 9	Number of double words/ Channel/Toggle bit	14h	1 double word = 4 bytes Channel = element = 2 Toggle bit = 0
	Byte 10	Status	00h	Command executed.
	byte 11	Reply counter	06h	For every ident telegram, the value on the reply counter increases by 1.
	byte 12	Data	10h	1. Byte data
	Byte 13	Data	11h	2. Byte data
	Byte 14	Data	12h	3. Byte data
	Byte 15	Data	13h	4. Byte data
3	Byte 16 ... Byte 23		00h	Reserved
4	Byte 24 ... Byte 31		00h	Reserved

12 Appendix B

12.1 Object model in the EtherNet/IP protocol

Class	Object name	Number of instances
01h	Identity	1
04h	Assembly	25
64h	Output command for channel 1 and 2, IDENTControl Compact and mixed mode	13
65h	Input command for channel 1 and 2, IDENTControl Compact and mixed mode	13
66h	Boot-up parameters	4
67h	Diagnostics	4

12.2 Identity object (01h)

Class attribute (instance 0)

Attribute ID	Name	Data type	Data content	Access authorization
1	Revision	UINT	1	Get

Instance attributes (instance 1)

Attribute ID	Name	Data type	Data content	Access authorization
1	Vendor number	UINT	57d	Get
2	Device type	UINT	00h	Get
3	Product code number	UINT	2	Get
4	Product major revision Product minor revision	USINT USINT	01 25	Get
5	Status	WORD	see below	Get
6	Serial number	UDINT	Unique 32-bit value	Get
7	Product number	String from USINT	IC-KP2-2HB17-2V1D #200877	Get

Shared services

Service code	integrated in		service designation
	Class level	Instance level	
0Eh	Yes	Yes	Get attribute single
05h	No	Yes	Reset

12.3 Assembly object (04h)

These instances are taken from classes 64h and 65h.

Class attributes (instance 0)

Attribute ID	Name	Data type	Data content	Access authorization
1	Revision	UINT	2	Get
2	Max. instance	UINT	162	Get
100	I/O output instance	USINT	100	Get / Set
101	I/O output produce length	UINT	8	Get
102	I/O input instance[1)]	USINT	150	Get
103	I/O input consume length	UINT	8	Get

Table 12.1 1) I/O input instance = I/O output instance + 50d

Output instance attributes (instances 100d-112d)

Attribute ID	Name	Data type	Data content	Access authorization
3	Output data	USINT [0-248]	0	Get

Output instance 100d - 8 bytes

Bytes	Class, instance, attribute	Description
0 - 7	64h, 06d, 01h	Mixed mode [8]

Output instance 101d - 12 bytes

Bytes	Class, instance, attribute	Description
0 - 11	64h, 06d, 02h	Mixed mode [12]

Output instance 102d - 32 bytes

Bytes	Class, instance, attribute	Description
0 - 31	64h, 06d, 03h	Mixed mode [32]

Output instance 103d - 60 bytes

Bytes	Class, instance, attribute	Description
0 - 59	64h, 06d, 04h	Mixed mode [60]

Output instance 104d - 32 bytes

Bytes	Class, instance, attribute	Description
0 - 7	64h, 01d, 01h	Channel 1 [8]
8 - 15	64h, 02d, 01h	Channel 2 [8]
16 - 23	Reserved	Reserved
24 - 31	Reserved	Reserved

Output instance 105d - 48 bytes

Bytes	Class, instance, attribute	Description
0 - 11	64h, 01d, 02h	Channel 1 [12]
12 - 23	64h, 02d, 02h	Channel 2 [12]

2014-02

Bytes	Class, instance, attribute	Description
24 - 35	Reserved	Reserved
36 - 47	Reserved	Reserved

Output instance 106d - 128 bytes

Bytes	Class, instance, attribute	Description
0 - 31	64h, 01d, 03h	Channel 1 [32]
32 - 63	64h, 02d, 03h	Channel 2 [32]
64 - 95	Reserved	Reserved
96 - 127	Reserved	Reserved

Output instance 107d - 240 bytes

Bytes	Class, instance, attribute	Description
0 - 59	64h, 01d, 04h	Channel 1 [60]
60 - 119	64h, 02d, 04h	Channel 2 [60]
120 - 179	Reserved	Reserved
180 - 239	Reserved	Reserved

Output instance 108d - 40 bytes

Bytes	Class, instance, attribute	Description
0 - 7	64h, 01d, 01h	Channel 1 [8]
8 - 15	64h, 02d, 01h	Channel 2 [8]
16 - 23	Reserved	Reserved
24 - 31	Reserved	Reserved
32 - 39	64h, 05d, 01h	IDENTControl Compact

Output instance 109d - 56 bytes

Bytes	Class, instance, attribute	Description
0 - 11	64h, 01d, 02h	Channel 1 [12]
12 - 23	64h, 02d, 02h	Channel 2 [12]
24 - 35	Reserved	Reserved
36 - 47	Reserved	Reserved
48 - 55	64h, 05d, 01h	IDENTControl Compact

Output instance 110d - 136 bytes

Bytes	Class, instance, attribute	Description
0 - 31	64h, 01d, 03h	Channel 1 [32]
32 - 63	64h, 02d, 03h	Channel 2 [32]
64 - 95	Reserved	Reserved
96 - 127	Reserved	Reserved
128 - 135	64h, 05d, 01h	IDENTControl Compact

Output instance 111d - 248 bytes

Bytes	Class, instance, attribute	Description
0 - 59	64h, 01d, 04h	Channel 1 [60]
60 - 119	64h, 02d, 04h	Channel 2 [60]
120 - 179	Reserved	Reserved
180 - 239	Reserved	Reserved
240 - 299	64h, 05d, 01h	IDENTControl Compact

Output instance 112d - 0 bytes

Bytes	Class, instance, attribute	Description
NONE	N/A	Heartbeat

Input instance attributes (instances 150d-162d)

Attribute ID	Name	Data type	Data content	Access authorization
3	Input data	USINT [8-248]	0	Get

Input instance 150d - 8 bytes

Bytes	Class, instance, attribute	Description
0 - 7	65h, 06d, 01h	Mixed mode [8]

Input instance 151d - 12 bytes

Bytes	Class, instance, attribute	Description
0 - 11	65h, 06d, 02h	Mixed mode [12]

Input instance 152d - 32 bytes

Bytes	Class, instance, attribute	Description
0 - 31	65h, 06d, 03h	Mixed mode [32]

Input instance 153d - 60 bytes

Bytes	Class, instance, attribute	Description
0 - 59	65h, 06d, 04h	Mixed mode [60]

Input instance 154d - 32 bytes

Bytes	Class, instance, attribute	Description
0 - 7	65h, 01d, 01h	Channel 1 [8]
8 - 15	65h, 02d, 01h	Channel 2 [8]
16 - 23	Reserved	Reserved
24 - 31	Reserved	Reserved

Input instance 155d - 48 bytes

Bytes	Class, instance, attribute	Description
0 - 11	65h, 01d, 02h	Channel 1 [12]
12 - 23	65h, 02d, 02h	Channel 2 [12]

2014-02

Bytes	Class, instance, attribute	Description
24 - 35	Reserved	Reserved
36 - 47	Reserved	Reserved

Input instance 156d - 128 bytes

Bytes	Class, instance, attribute	Description
0 - 31	65h, 01d, 03h	Channel 1 [32]
32 - 63	65h, 02d, 03h	Channel 2 [32]
64 - 95	Reserved	Reserved
96 - 127	Reserved	Reserved

Input instance 157d - 240 bytes

Bytes	Class, instance, attribute	Description
0 - 63	65h, 01d, 04h	Channel 1 [60]
64 - 127	65h, 02d, 04h	Channel 2 [60]
128 - 191	Reserved	Reserved
192 - 255	Reserved	Reserved

Input instance 158d - 40 bytes

Bytes	Class, instance, attribute	Description
0 - 7	65h, 01d, 01h	Channel 1 [8]
8 - 15	65h, 02d, 01h	Channel 2 [8]
16 - 23	Reserved	Reserved
24 - 31	Reserved	Reserved
32 - 39	65h, 05d, 01h	IDENTControl Compact

Input instance 159d - 56 bytes

Bytes	Class, instance, attribute	Description
0 - 11	65h, 01d, 02h	Channel 1 [12]
12 - 23	65h, 02d, 02h	Channel 2 [12]
24 - 35	Reserved	Reserved
36 - 47	Reserved	Reserved
48 - 55	65h, 05d, 01h	IDENTControl Compact

Input instance 160d - 136 bytes

Bytes	Class, instance, attribute	Description
0 - 31	65h, 01d, 03h	Channel 1 [32]
32 - 63	65h, 02d, 03h	Channel 2 [32]
64 - 95	Reserved	Reserved
96 - 127	Reserved	Reserved
128 - 135	65h, 05d, 01h	IDENTControl Compact

Input instance 161d - 248 bytes

Bytes	Class, instance, attribute	Description
0 - 59	65h, 01d, 04h	Channel 1 [60]
60 - 119	65h, 02d, 04h	Channel 2 [60]
120 - 179	Reserved	Reserved
180 - 239	Reserved	Reserved
240 - 299	65h, 05d, 01h	IDENTControl Compact

Input instance 162d - 10 bytes

Bytes	Class, instance, attribute	Description
0 - 9	65h, 00d, 64h	Status

Shared services

Service code	integrated in		service designation
	Class level	Instance level	
0Eh	Yes	Yes	Get attribute single
10h	Yes	Yes	Set attribute single

12.4 Output command object (instances 64h - 6)

Class attributes (instance 0)

Attribute ID	Name	Data type	Data content	Access authorization
1	Revision	UINT	1	Get

Instance attributes (instances 1-6)

- Instances 1 and 2: Read/write head 1 and 2
- Instance 5 - Ident Control
- Instance 6 - Mixed mode

Attribute ID	Name	Data type	Data content	Access authorization
1	Output data image (first 8 bytes)	USINT[8]	0	Get / Set
2	Output data image (first 12 bytes)	USINT[12]	0	Get / Set
3	Output data image (first 32 bytes)	USINT[32]	0	Get / Set
4	Output data image (first 60 bytes)	USINT[60]	0	Get / Set

Shared services

Service code	integrated in		service designation
	Class level	Instance level	
0Eh	Yes	Yes	Get attribute single
10h	No	Yes	Set attribute single

2014-02

12.5 Input command object (instances 65h - 6)

Class attributes (instance 0)

Attribute ID	Name	Data type	Data content	Access authorization
1	Revision	UINT	1	Get
100	Explicit status	USINT[10]	0	Get

Instance attributes (instances 1-6)

- Instances 1 and 2: Read/write head 1 and 2
- Instance 5 - Ident Control
- Instance 6 - Mixed mode

Attribute ID	Name	Data type	Data content	Access authorization
1	Input data image (first 8 bytes)	USINT[8]	0	Get
2	Input data image (first 12 bytes)	USINT[12]	0	Get
3	Input data image (first 32 bytes)	USINT[32]	0	Get
4	Input data image (first 60 bytes)	USINT[60]	0	Get

Shared services

Service code	integrated in		service designation
	Class level	Instance level	
0Eh	Yes	Yes	Get attribute single

12.6 Boot-up parameter object (instances 66h - 4)

Class attributes (instance 0)

Attribute ID	Name	Data type	Data content	Access authorization
1	Revision	UINT	1	Get
100	Multiplex mode	BOOL	0	Get / Set
101	Data hold time	USINT	0	Get / Set

Instance attributes (instances 1 and 2, channels 1 and 2)

Attribute ID	Name	Data type	Data content	Access authorization
1	Tag type	USINT	3	Get / Set

Shared services

Service code	integrated in		service designation
	Class level	Instance level	
0Eh	Yes	Yes	Get attribute single
10h	Yes	Yes	Get attribute single

12.7 Diagnostics object (instances 67h - 5)

Class attributes (instance 0)

Attribute ID	Name	Data type	Data content	Access authorization
1	Revision	UINT	1	Get
100	Refresh all	BOOL	0	Get / Set
101	Multiplex mode	BOOL	0	Get
102	Trigger condition 3	USINT	0	Get
103	Trigger condition 4	USINT	0	Get
104	Version gateway	Bytes [52]	0	Get

Instance attributes (instances 1 and 2, channels 1 and 2)

Attribute ID	Name	Data type	Data content	Access authorization
1	Tag type	USINT	0	Get
2	Version	Bytes [52]	0	Get

Instance attributes (instance 5, IdentControl)

Attribute ID	Name	Data type	Data content	Access authorization
1	Unused			
2	Version	Bytes [52]	0	Get

Shared services

Service code	integrated in		service designation
	Class level	Instance level	
0Eh	Yes	Yes	Get attribute single
10h	Yes	No	Set attribute single

2014-02

FACTORY AUTOMATION – SENSING YOUR NEEDS

Worldwide Headquarters
Pepperl+Fuchs GmbH
68307 Mannheim · Germany
Tel. +49 621 776-0
E-mail: info@de.pepperl-fuchs.com

USA Headquarters
Pepperl+Fuchs Inc.
Twinsburg, Ohio 44087 · USA
Tel. +1 330 4253555
E-mail: sales@us.pepperl-fuchs.com

Asia Pacific Headquarters
Pepperl+Fuchs Pte Ltd.
Company Registration No. 199003130E
Singapore 139942
Tel. +65 67799091
E-mail: sales@sg.pepperl-fuchs.com

www.pepperl-fuchs.com

PEPPERL+FUCHS
SENSING YOUR NEEDS

/ TDOCT1748F_ENG
02/2014