Manual
Commissioning
IDENTControl Compact
IC-KP2-2HB17-2V1D
with PROFINET Protocol
to Siemens S7 Controller
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  Setting Communication Parameter via Web Interface .......... 5
2  Installing the GSDML File ................................................. 7
3  Installing the User Program ............................................... 9
4  Hardware Configuration .................................................... 10
5  Setting Device Parameters ............................................... 13
6  Function Block „FB190_IUHParm“ ..................................... 14
   6.1  Reset to Default: Kanal 1 ............................................. 17
   6.2  PowerTransmit - PT: Channel 1 .................................... 19
   6.3  ChannelDenseReaderMode - CD: Channel 1 .................... 19
   6.4  Read/Write Multiple Parameters: Channel 1 ................. 20
7  Function Block FB32 „Singleframe“ ................................ 22
8  Error or Device Diagnostics ............................................. 26
9  Example Command Execution ......................................... 27
   9.1  Initialization: (with Tag Type IPC03) ......................... 27
   9.2  Single Read Fixcode: (Head 1) .................................. 28
   9.3  Enhanced Read Fixcode: (Head 1) .............................. 29
   9.4  Single Read Words: (Head 1; 15 Blocks from Address 0) ..... 30
   9.5  Enhanced Read Words: (Head 1; 15 Blocks from Address 0) .... 31
   9.6  Single Read SpecialFixcode: (Head 1) ......................... 32
   9.7  Enhanced Read SpecialFixcode: (Head 1) ................. 33
   9.8  Single Write Words: (Head 1; 15 Blocks from Address 0) ..... 34
   9.9  Enhanced Write Words: (Head 1; 15 Blocks from Address 0) .... 35
   9.10 Single Write Fixcode: (Head 1; IPC11) ...................... 36
   9.11 Single Write SpecialFixcode: (Kopf 1) ....................... 37
   9.12 Error Handling: (Head 1) ........................................ 38
9.13 Command List (Prefetch): (Head 1) ......................................................... 38

10 Table of Read/Write Tags ........................................................................ 42

11 Table Status Values ................................................................................. 44

12 Table Version History ............................................................................ 45
1 Setting Communication Parameter via Web Interface

In the default state of IDENTControl, no PROFINET IO name is assigned. The name of the node can be assigned using the device website. The default IP address is 169.254.10.12.

The communication parameters and the PROFINET name can be parameterized by selecting "Network".

The parameters are applied to the device by selecting "Save". The device will then restart automatically. The new configuration will be activated as soon as the rotary switch on the back of the device is set to the "0" position.
Alternatively, the PROFINET name can be assigned via the hardware configuration of the controller. To assign a name, select the following options in the hardware configuration menu: PLC → Ethernet → Edit Ethernet Node

Search for connected nodes in the PROFINET IO system by selecting "Browse...". The IP address and the PROFINET name can then be reassigned.
2 Installing the GSDML File

Before commissioning the IDENTControl identification system for the first time, the GSDML file must first be installed. The GSDML file can be found on the "Identification Systems" CD packaged with the product. The file can also be downloaded from the Internet.

www.pepperl-fuchs.com

(Product search → IC-KP2-2HB17-2V1D → 18XXXXXX.zip)

To install the file, select menu item "Options" → "Install GSD File..." from the Simatic hardware configuration menu.

Then select the GSDML file from the source directory.

The GSDML file is added to the hardware catalog via the menu item "Options" → "Update Catalog".

2014-02

PEPPERL+FUCHS

7
Commissioning IDENTControl Compact IC-KP2-2HB17-2V1D with PROFINET Protocol to Siemens S7 Controller
3 Installing the User Program

Before the user program can be installed, the file "IC-KP2-2HB17.zip" must first be unzipped. Unzip the file by selecting the menu item "File" → "Retrieve..." in SIMATIC Manager. Highlight the file, confirm with "Open", and select the relevant file path. Following successful installation, the user program will appear within SIMATIC Manager.
4 Hardware Configuration

The existing modules must be parameterized within the hardware configuration menu.

Add a PROFINET IO system by right-clicking on the PROFINET port.

IDENTControl is then connected to the PROFINET IO system In the hardware catalog, select the following and drag to the master system: Additional Field Devices → Ident Systems → Pepperl+Fuchs Ident → "IC-KP2-2HB17 2 PDEV’s". The telegram length is then defined by the communication blocks (e.g., In/Out 64 bytes). The module must be assigned to a slot.
Double-click the IDENTControl icon to open a window containing the properties of the PROFINET node. The name of the PROFINET node (in this case: identcontrol) can be changed in this window if required.

Selecting "Ethernet" allows you to change the IP address of the node.
If the function block is transferred to another application program, it is recommended that the symbol table also be copied to the new program. The symbol information will support working processes.
5 Setting Device Parameters

The device parameters can be accessed by right-clicking slot 0 of the I/O table.

The device-specific parameter "DataHoldTime" (DHT) describes the data hold time within the output data field of the IDENTControl. The DHT should be twice the value of the cycle time of the controller.
6 Function Block „FB190_IUHParam“

The FB190 is used to parameterize the IUH-F190 and IUH-F117 readers. The block and the associated instance data block is opened as follows:

Call "FB190_IUHParam", "DB190_IUHParam" (symbolic representation)

or

Call FB190, DB190

To activate the function block, "ParameterEnable" must be active. If "NormalEnable" is active, then this must first be reset.

This block supports control interfaces with up to two channels.

The following figure shows the function being accessed, as well as the variables requiring parameterization.

<table>
<thead>
<tr>
<th>Name</th>
<th>Declaration</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeadNumber</td>
<td>Input</td>
<td>BYTE</td>
<td>Head/channel number on which parameters are to be read/written</td>
</tr>
<tr>
<td>ReadWriteParameter</td>
<td>Input</td>
<td>BOOL</td>
<td>Execution of 0:= read, 1:= write parameters</td>
</tr>
<tr>
<td>PowerTransmit</td>
<td>Input</td>
<td>BOOL</td>
<td>Access to reader transmission power</td>
</tr>
<tr>
<td>TriesAllowed</td>
<td>Input</td>
<td>BOOL</td>
<td>Access to read/write attempts</td>
</tr>
<tr>
<td>ChannelDenseReaderMode</td>
<td>Input</td>
<td>BOOL</td>
<td>Access to the sequence of transmission channels permitted in dense reader mode (DRM)</td>
</tr>
<tr>
<td>ProtocolMode</td>
<td>Input</td>
<td>BOOL</td>
<td>Access to output protocol (singleframe and multiframe)</td>
</tr>
<tr>
<td>QValue</td>
<td>Input</td>
<td>BOOL</td>
<td>Access to Q-value used to define the time slots used (2(^{2})) for anti-collision (Slotted Aloha)</td>
</tr>
<tr>
<td>Function</td>
<td>Input/Output</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Number of Tags</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Access to the number of tags in the detection range for which the read/write head is searching</td>
</tr>
<tr>
<td>Sensing Mode</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Access to pause time in ms, after all runs are complete</td>
</tr>
<tr>
<td>Memory Bank</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Access to the bank that the read/write commands SR, ER, SW, and EW are accessing</td>
</tr>
<tr>
<td>Measure Reflection</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Access to measured reflected transmission power (read only)</td>
</tr>
<tr>
<td>Additional Information</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Access to output of additional information, provided that the reading was successful and the singleframe protocol has been set (read only)—not in multiframe operation!</td>
</tr>
<tr>
<td>Reset to Default</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Returns all settings of the read/write head to the default configuration</td>
</tr>
<tr>
<td>Enhanced Status 5</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Sets the number of unsuccessful write/read attempts before a status 5 is output for an advanced command</td>
</tr>
<tr>
<td>Antenna Polarisation</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Switches the polarization to linear or circular</td>
</tr>
<tr>
<td>Param Start</td>
<td>Input/Output</td>
<td>BOOL</td>
<td>Execution of parameter command (positive edge)</td>
</tr>
<tr>
<td>Param Finished</td>
<td>Output/Input</td>
<td>BOOL</td>
<td>Parameter access terminated</td>
</tr>
<tr>
<td>Param Error</td>
<td>Output/Input</td>
<td>BOOL</td>
<td>An error occurred</td>
</tr>
<tr>
<td>Param Type Error</td>
<td>Output/Input</td>
<td>WORD</td>
<td>Return of the incorrect parameter type (HEX)</td>
</tr>
<tr>
<td>Param Busy</td>
<td>Output/Input</td>
<td>BOOL</td>
<td>Individual parameter command is currently being executed</td>
</tr>
</tbody>
</table>

**Note!**

In the default state of the read/write head (IUH-F190), multiframe mode is activated. A write command must first be carried out on the protocol mode "ProtocolMode" in order to switch to singleframe mode!

It is possible to read or write individual parameters or multiple/all parameters at once.

The parameters that are read out are saved in the instance data block DB190 in the structures provided. The "Power Transmit" (PT) parameter illustrates this point.
Data read out in DB190:

PT.Read.Length := W#16#4

PT.Read.PT1 := W#16#14 (transmission power 1)

PT.Read.PT2 := W#16#64 (transmission power 2)

The structure is as follows: abbreviation of the parameter type + read/write + parameter

OB1 includes network 2 for writing the parameters. The required values must be written into the code in network 2. Depending on the inputs set, only the selected parameters are written.

The correct notation and appropriate value ranges can be found in the attached network comment (network 2: init parameters).

If a parameter is configured incorrectly, its parameter type is displayed after the transfer.
The figure shows the following parameters being written to channel 1: "Power Transmit" (PT), "Tries Allowed" (TA), "Protocol Mode" (QV), and "Number of Tags" (NT). The "Tries Allowed" parameter has been configured incorrectly, as identified by the return value of "ErrorCMD" := W#16#5441. The output contains the hexadecimal parameter type of the incorrect parameter (5441h = TA\text{ASCII}).

If parameters are to be written, and only parameter types to be read are set on the block (see manual F190), the function block ignores these inputs. It follows the same process when reading parameters.

In the block's default state, all parameters in OB1 are assigned default values.

**Example command executions:**

6.1 Reset to Default: Kanal 1

**Note!**

In the default state of the read/write head (IUH-F190), multiframe mode is activated. "Reset to Default" loads the default parameters, meaning the operating mode is switched back to multiframe mode!
Define parameters:
HeadNumber := B#16#01
ReadWriteParameter := 1
ResetToDefault := 1

Start execution:
Start := 1 (positive edge)
ParamBusy := 1

Processing complete:
ParamBusy := 0
Finish := 1

Once the command has been initialized with "ParamStart", "ParamBusy" indicates that command processing is in progress. During this time, "ParamFinished" switches to FALSE. Once the command is complete, "ParamFinished" returns to TRUE.
6.2 PowerTransmit - PT: Channel 1

Reading:

Festlegung der Parameter:
HeadNumber := B#16#01
ReadWriteParameter := 0
PowerTransmit := 1

Start der Ausführung:
Start := 1 (positive Flanke)
ParamBusy := 1

Bearbeitung beendet:
ParamBusy := 0
Finish := 1
Nutzdaten: DB190

Write: Parameterize three transmission powers

Define parameters:
HeadNumber := B#16#01
ReadWriteParameter := 1
PowerTransmit := 1
"DB190_IUHParam".PT.Write.Length := W#16#6
"DB190_IUHParam".PT.Write.PT1 := W#16#30
"DB190_IUHParam".PT.Write.PT2 := W#16#35
"DB190_IUHParam".PT.Write.PT3 := W#16#40

Start der Ausführung:
Start := 1 (positive Flanke)
ParamBusy := 1

Processing complete:
ParamBusy := 0
Finish := 1

6.3 ChannelDenseReaderMode - CD: Channel 1

Reading:

Define parameters:
HeadNumber := B#16#01
ReadWriteParameter := 0
ChannelDenseReaderMode := 1

Start execution:
Start := 1 (positive edge)
ParamBusy := 1

Processing complete:
ParamBusy := 0
Finish := 1
Usable data: DB190
Write: Channel A & channel B

Define parameters:
HeadNumber := B#16#01
ReadWriteParameter := 0
ChannelDenseReaderMode := 1

Start execution:
Start := 1 (positive edge)
ParamBusy := 1

Processing complete:
ParamBusy := 0
Finish := 1
Usable data: DB190

6.4 Read/Write Multiple Parameters: Channel 1

Reading:

Define parameters:
HeadNumber := B#16#01
ReadWriteParameter := 0
PowerTransmit := 1
TriesAllowed := 1

Start execution:
Start := 1 (positive edge)
ParamBusy := 1

Processing complete:
Finish := 1
ParamBusy := 0
Usable data: DB190

1. Command execution (read) is started with "ParamStart". "ParamFinished" switches to FALSE.
2. "ParamBusy" indicates command processing of "Power Transmit" and "Tries Allowed".
3. Once both parameters are complete, "ParamFinished" returns to TRUE.
Writing:

1. Command execution (write) is started with "ParamStart". "ParamFinished" switches to FALSE.
4. All commands processed. "ParamFinished" returns to TRUE.
Function Block FB32 „Singleframe“

The function block and corresponding instance data block are accessed as follows:

Call "IDENTControl", "InstDB" (symbolic representation)

or

Call FB32, DB32

For activation to be successful, "NormalEnable" must be active.

This block supports control interfaces with up to two channels.

The following figure shows the function being accessed, as well as the variables requiring parameterization.
### Commissioning IDENTControl Compact IC-KP2-2HB17-2V1D with PROFINET Protocol to Siemens S7 Controller

<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC_INPUT_Address</td>
<td>WORD</td>
<td>Start address of the control interface in the process image of the inputs (I address)</td>
</tr>
<tr>
<td>IC_OUTPUT_Address</td>
<td>WORD</td>
<td>Start address of the control interface in the process image of the outputs (O address)</td>
</tr>
<tr>
<td>Length_IN</td>
<td>INT</td>
<td>Length of the input image (length of the PROFIBUS telegram received)</td>
</tr>
<tr>
<td>Length_OUT</td>
<td>INT</td>
<td>Length of the output image (length of the PROFIBUS telegram sent)</td>
</tr>
<tr>
<td>Timeout</td>
<td>TIME</td>
<td>Timer for monitoring the response time of the control interface</td>
</tr>
<tr>
<td>Head1DataFixcode</td>
<td>BOOL</td>
<td>Head 1 access to 0:=Fixcode 1:=data range</td>
</tr>
<tr>
<td>Head2DataFixcode</td>
<td>BOOL</td>
<td>Head 2 access to 0:=Fixcode 1:=data range</td>
</tr>
<tr>
<td>Head1SingleEnhanced</td>
<td>BOOL</td>
<td>Execution on head 1 from 0:=single 1:=enhanced command</td>
</tr>
<tr>
<td>Head2SingleEnhanced</td>
<td>BOOL</td>
<td>Execution on head 2 from 0:=single 1:=enhanced command</td>
</tr>
<tr>
<td>Head1SpecialCommand</td>
<td>BOOL</td>
<td>Execution of a special command on head 1 (positive edge); define command parameters in advance in structure Head_1.SpecialCommand; usable data received is located in the structure Head_1.InData</td>
</tr>
<tr>
<td>Head2SpecialCommand</td>
<td>BOOL</td>
<td>Execution of a special command on head 2 (positive edge); define command parameters in advance in structure Head_2.SpecialCommand; usable data received is located in the structure Head_2.InData</td>
</tr>
<tr>
<td>Head1Read</td>
<td>BOOL</td>
<td>Execution of read command on head 1 (positive edge); define command parameters Head1WordNum and Head1Wordaddress; usable data received is located in the structure Head_1.InData</td>
</tr>
<tr>
<td>Head2Read</td>
<td>BOOL</td>
<td>Execution of read command on head 2 (positive edge); define command parameters Head2WordNum and Head2Wordaddress; usable data received is located in the structure Head_2.InData</td>
</tr>
<tr>
<td>Head1Write</td>
<td>BOOL</td>
<td>Execution of write command on head 1 (positive edge); define command parameters Head1WordNum and Head1Wordaddress; define data to be written in advance in the structure Head_1.OutData.DW1...15</td>
</tr>
<tr>
<td>Head2Write</td>
<td>BOOL</td>
<td>Execution of write command on head 2 (positive edge); define command parameters Head2WordNum and Head2Wordaddress; define data to be written in advance in the structure Head_2.OutData.DW1...15</td>
</tr>
<tr>
<td>Head1Quit</td>
<td>BOOL</td>
<td>Execution of quit command on head 1 to cancel an enhanced command (positive edge)</td>
</tr>
<tr>
<td>Head2Quit</td>
<td>BOOL</td>
<td>Execution of quit command on head 2 to cancel an enhanced command (positive edge)</td>
</tr>
<tr>
<td>QuitErrorHead1</td>
<td>BOOL</td>
<td>Execution of error routine on head 1 (positive edge)</td>
</tr>
<tr>
<td>QuitErrorHead2</td>
<td>BOOL</td>
<td>Execution of error routine on head 2 (positive edge)</td>
</tr>
<tr>
<td>IC_Command_Head1</td>
<td>BOOL</td>
<td>Execution of a special command on the control interface (positive edge); command is sent to channel 0; define command parameters in advance in the structure Head_1.SpecialCommand; usable data received is located in the structure Head_1.InData</td>
</tr>
<tr>
<td>Head1WordNum</td>
<td>INT</td>
<td>Number of data blocks activated on head 1</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Head2WordNum</td>
<td>INT</td>
<td>Number of data blocks activated on head 2</td>
</tr>
<tr>
<td>Head1WordAddress</td>
<td>WORD</td>
<td>Start address of the memory activated on channel 1 on read/write tag</td>
</tr>
<tr>
<td>Head2WordAddress</td>
<td>WORD</td>
<td>Start address of the memory activated on channel 2 on read/write tag</td>
</tr>
<tr>
<td>Head1TagType</td>
<td>WORD</td>
<td>Tag type channel 1 (table of tag types)</td>
</tr>
<tr>
<td>Head2TagType</td>
<td>WORD</td>
<td>Tag type channel 2 (table of tag types)</td>
</tr>
<tr>
<td>Head1SpecialFixcode</td>
<td>BOOL</td>
<td>Head 1 access to 1:=SpecialFixcode (EPC)</td>
</tr>
<tr>
<td>Head2SpecialFixcode</td>
<td>BOOL</td>
<td>Head 2 access to 1:=SpecialFixcode (EPC)</td>
</tr>
<tr>
<td>Head1Done</td>
<td>BOOL</td>
<td>New data present (enhanced) or command complete (single) on channel 1</td>
</tr>
<tr>
<td>Head2Done</td>
<td>BOOL</td>
<td>New data present (enhanced) or command complete (single) on channel 2</td>
</tr>
<tr>
<td>Head1NoDataCarrier</td>
<td>BOOL</td>
<td>No read/write tag for execution time of a command on channel 1 present</td>
</tr>
<tr>
<td>Head2NoDataCarrier</td>
<td>BOOL</td>
<td>No read/write tag for execution time of a command on channel 2 present</td>
</tr>
<tr>
<td>Head1Error</td>
<td>BOOL</td>
<td>Error occurred on channel 1 (positive edge)</td>
</tr>
<tr>
<td>Head2Error</td>
<td>BOOL</td>
<td>Error occurred on channel 2 (positive edge)</td>
</tr>
<tr>
<td>Head1Busy</td>
<td>BOOL</td>
<td>Command is processed on channel 1</td>
</tr>
<tr>
<td>Head2Busy</td>
<td>BOOL</td>
<td>Command is processed on channel 2</td>
</tr>
<tr>
<td>Head1Status</td>
<td>BYTE</td>
<td>Channel 1 status value</td>
</tr>
<tr>
<td>Head2Status</td>
<td>BYTE</td>
<td>Channel 2 status value</td>
</tr>
<tr>
<td>Head1ReplyCounter</td>
<td>BYTE</td>
<td>Channel 1 reply counter value</td>
</tr>
<tr>
<td>Head2ReplyCounter</td>
<td>BYTE</td>
<td>Channel 2 reply counter value</td>
</tr>
<tr>
<td>InitFinish</td>
<td>BOOL</td>
<td>Initialization complete (positive edge)</td>
</tr>
<tr>
<td>SetRestart</td>
<td>BOOL</td>
<td>Execution of control interface initialization (positive edge)</td>
</tr>
</tbody>
</table>

Example of setting communication parameters:

The communication module "64 In / 64 Out Bytes" has been selected in the hardware configuration. The process image of the inputs (I address) starts at address 512, ends after address 575, and has a length of 64 bytes. The process image of the outputs (O address) starts at address 512 and ends after address 575. This results in the following parameterization for the function block:

```plaintext
IC_INPUT_Address := W#16#200
IC_OUTPUT_Address := W#16#200
Length_IN := 64
Length_OUT := 64
```
Note:

When selecting a communication module for the pure read operation (e.g., "64 In / 4 Out Bytes"), the source address ("IC_OUTPUT_Address") must be completely within the process image. For CPUs in the S7-300 Series, the O range has a length of 0 to 256. The S7-400 Series has a process image for the outputs with a length of 0 to 512.

Please note the maximum telegram length for the CPU used. CPU S7-315-2DP is designed for a maximum telegram length of 32 bytes. CPUs in the S7-400 Series are designed for a telegram length of 64 bytes.
### 8 Error or Device Diagnostics

Most errors occur during commissioning of the block (initialization). If the InitFinish bit is not automatically set after the start of initialization (positive edge on SetRestart), an error has occurred during initialization. The primary cause is different parameterization of the hardware configuration within the Simatic hardware manager and the parameterization of the I/O range or the telegram length on the function block. The set tag type (Head1(2)TagType) must also be checked.

There is a further potential for errors when executing commands. If an error occurs in these cases, then the Head1(2)Error bit is set. A diagnosis can then be carried out using the error table below. The cause of the errors may be incorrect command parameters (particularly in the case of a special command) or an incorrectly set tag type. An incorrect command parameter is indicated by the status value (Head1(2)Status) 0x04.

<table>
<thead>
<tr>
<th>Name</th>
<th>Data type</th>
<th>Description</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1(2)Error</td>
<td>BOOL</td>
<td>Error on head 1(2)</td>
<td>Check the Data Hold Time. Check the I/O configuration.</td>
</tr>
<tr>
<td>Head_1(2).InvalidResponse</td>
<td>BOOL</td>
<td>Sent and received command telegram are not identical</td>
<td>Increase the variable timeout for long cycle times.</td>
</tr>
<tr>
<td>Head_1(2).TimeoutOccured</td>
<td>BOOL</td>
<td>Timer for communication monitoring has expired. Slave has not sent a response in the time frame.</td>
<td>Check the variables Head_1(2).Ret_Val_SFC14. Check the parameterized I/O address and telegram length.</td>
</tr>
<tr>
<td>Head_1(2).Error_SFC_14</td>
<td>BOOL</td>
<td>Error reading in the data from the process image</td>
<td></td>
</tr>
<tr>
<td>Head_1(2).Error_SFC_15</td>
<td>BOOL</td>
<td>Error writing the data to the process image</td>
<td>Check the variables Head_1(2).Ret_Val_SFC15. Check the parameterized I/O address and telegram length.</td>
</tr>
<tr>
<td>Head_1(2).Ret_Val_SFC14</td>
<td>WORD</td>
<td>Error code during execution of SFC14</td>
<td>W#16#8090</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>W#16#80B1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check the parameterized I/O address and telegram length (additional information on SFC14 system help)</td>
</tr>
<tr>
<td>Head_1(2).Ret_Val_SFC15</td>
<td>INT</td>
<td>Error code during execution of SFC15 (conversion to HEX format required)</td>
<td>W#16#8090</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>W#16#80B1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check the parameterized I/O address and telegram length (additional information on SFC15 system help)</td>
</tr>
<tr>
<td>Head1(2)Status</td>
<td>BYTE</td>
<td>Status of the data last read in on channel 1(2)</td>
<td>See table of status values</td>
</tr>
<tr>
<td>Memory.InData.Status</td>
<td>BYTE</td>
<td>Status of the data telegram last read in</td>
<td>See table of status values</td>
</tr>
</tbody>
</table>
9 Example Command Execution

9.1 Initialization: (with Tag Type IPC03)

Define tag type:
Head1TagType := W#16#3830
Head2TagType := W#16#3830

Start initialization:
SetRestart := 1 (positive edge)

Initialization is processing:
Head1Busy := 1
Head2Busy := 1

Initialization complete:
InitFinish := 1 (positive edge)
Head1Done := 1
Head2Done := 1

1. "SetRestart" triggers initialization routine. Initialization is completed on head 1 ("Head1Busy").
2. Head 2 is initialized ("Head2Busy") immediately after head 1 has sent confirmation that it must be initialized.
3. Initialization on head 1 complete ("Head1Done").
4. "Head1Done" and "Head2Done" switch to TRUE. Initialization complete; "InitFinish" switches to TRUE.
9.2 Single Read Fixcode: (Head 1)

Define command parameters:
Head1SingleEnhanced := 0
Head1DataFixcode := 1

Start command execution:
Head1Read := 1 (positive edge)

Command is processing:
Head1Busy := 1

Command complete -> data read in:
Head1Busy := 0
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Usable data: Head_1.InData.UserData

Command complete -> No read/write tag present:
Head1Busy := 0
Head1NoDataCarrier := 1 (positive edge)
Head1Done := 1 (positive edge)
Usable data: none

1. "Head1SingleEnhanced" is zero-active in order to initialize a single command. "Head1DataFixcode" is TRUE in order to access the Fixcode/TID. Single Read Fixcode is started by "Head1Read" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the reading, there was no read/write tag in the detection range and "Head1NoDataCarrier" switches to TRUE.

3. Single Read Fixcode is started by "Head1Read" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE. "Head1NoDataCarrier" switches to FALSE.

4. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the reading, there was a read/write tag in the detection range and "Head1NoDataCarrier" remains zero-active.
## 9.3 Enhanced Read Fixcode: (Head 1)

### Define Command Parameters:
- Head1SingleEnhanced := 1
- Head1DataFixcode := 1

### Start Command Execution:
- Head1Read := 1 (positive edge)

### Command is Processing:
- Head1Busy := 1

#### Data Read In and Command Active:
- Head1Busy := 1
- Head1DataFixcode := 0
- Head1Done := 1 (positive edge)
- Usable data: Head_1.InData.UserData

#### No Read/Write Tag Present and Command Active:
- Head1Busy := 1
- Head1DataCarrier := 1
- Head1Done := 0 (positive edge)
- Usable data: none

#### Read/Write Tag Moving Out of the Read Range and Data Read In:
- Head1Busy := 1
- Head1DataCarrier := 0
- Head1Done := 1 (positive edge)
- Usable data: Head_1.InData.UserData

#### End Enhanced Command:
- Head1Quit := 1

#### Command Complete:
- Head1Done := 1 (positive edge)
- Head1Busy := 0

---

1. "Head1SingleEnhanced" is TRUE in order to initialize an enhanced command. "Head1DataFixcode" is TRUE in order to access the Fixcode/TID. Enhanced Read Fixcode is started, "Head1Read" is TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. "Head1NoDataCarrier" is TRUE as there is no read/write tag in range at this time.

3. Read/write tag enters detection range. "Head1NoDataCarrier" switches to FALSE, "Head1Done" switches to TRUE. The read/write tag exits the detection range shortly afterward.

4. "Head1Quit" cancels continuous reading. "Head1Busy" switches to FALSE, "Head1Done" is now TRUE.
5. New Enhanced Read Fixcode command started. Read/write tag detected; "Head1NoDataCarrier" is FALSE, "Head1Done" is active.

6. "Head1Done" is FALSE as there is no read/write tag in the detection range; "Head1NoDataCarrier" is TRUE. Read/write tag enters the detection range again shortly afterward.

7. Cancel command with "Head1Quit"

9.4 Single Read Words: (Head 1; 15 Blocks from Address 0)

Define command parameters:
- Head1WordNum := 15 (usable data count)
- Head1WordAddress := 0 (memory address)
- Head1SingleEnhanced := 0
- Head1DataFixcode := 0

Start command execution:
- Head1Read := 1 (positive edge)

Command is processing:
- Head1Busy := 1

Data read in:
- Head1Busy := 0
- Head1NoDataCarrier := 0
- Head1Done := 1 (positive edge)

Usable data:
- Head_1.InData.UserData

No read/write tag present:
- Head1Busy := 0
- Head1NoDataCarrier := 1 (positive edge)
- Head1Done := 1 (positive edge)

Usable data: none

1. "Head1SingleEnhanced" is zero-active in order to initialize a single command. "Head1DataFixcode" is FALSE in order to access words/user data. Single Read Words is started by "Head1Read" switching to TRUE, "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the reading, there was no read/write tag in the detection range and "Head1NoDataCarrier" switches to TRUE.

3. Single Read Words is started by "Head1Read" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE. "Head1NoDataCarrier" switches to FALSE.

4. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the reading, there was a read/write tag in the detection range and "Head1NoDataCarrier" remains zero-active.
9.5 Enhanced Read Words: (Head 1; 15 Blocks from Address 0)

Define command parameters:
Head1WordNum := 15 (usable data count)
Head1WordAddress := 0 (memory address)
Head1SingleEnhanced := 1
Head1DataFixcode := 0

Start command execution:
Head1Read := 1 (positive edge)

Command is processing:
Head1Busy := 1

Data read in and command active:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Usable data:
Head_1.InData.UserData

No read/write tag present and command active:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Usable data: none

Read/write tag moving out of the read range and data read in:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 0 (positive edge)
Usable data: none

Read/write tag moving into the read range and data read in:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Usable data:
Head_1.InData.UserData

End enhanced command:
Head1Quit := 1

Command complete:
Head1Done := 1 (positive edge)
Head1Busy := 0

1. "Head1SingleEnhanced" is TRUE in order to initialize an enhanced command. "Head1DataFixcode" is FALSE in order to access words/user data. Enhanced Read Words is started, "Head1Read" is TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. "Head1NoDataCarrier" is TRUE as there is no read/write tag in range at this time.

3. Read/write tag enters detection range. "Head1NoDataCarrier" switches to FALSE, "Head1Done" switches to TRUE. The read/write tag exits the detection range shortly afterward.
4. "Head1Quit" cancels continuous reading. "Head1Busy" switches to FALSE, "Head1Done" is now TRUE.

5. New Enhanced Read Words command started. Read/write tag detected; "Head1NoDataCarrier" is FALSE, "Head1Done" is active.

6. "Head1Done" is FALSE as there is no read/write tag in the detection range; "Head1NoDataCarrier" is TRUE. Read/write tag enters the detection range again shortly afterward.

7. Cancel command with "Head1Quit"

9.6 Single Read SpecialFixcode: (Head 1)

Define command parameters:
Head1SingleEnhanced := 0
Head1SpecialFixcode := 1

Start command execution:
Head1Read := 1 (positive edge)

Command is processing:
Head1Busy := 1

Command complete -> data read in:
Head1Busy := 0
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Usable data: Head_1.InData.UserData

Command complete -> No read/write tag present:
Head1Busy := 0
Head1NoDataCarrier := 1 (positive edge)
Head1Done := 1 (positive edge)
Usable data: none

1. "Head1SingleEnhanced" is zero-active in order to initialize a single command. "Head1DataFixcode" is irrelevant as "Head1SpecialFixcode" has higher priority than Words or Fixcode (i.e., as soon as "Head1SpecialFixcode" is TRUE, "Head1DataFixcode" becomes invalid). "Head1DataFixcode" is TRUE in order to access the SpecialFixcode/EPC. Single Read SpecialFixcode is started by "Head1Read" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the reading, there was no read/write tag in the detection range and "Head1NoDataCarrier" switches to TRUE.
3. Single Read SpecialFixcode is started by "Head1Read" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE. "Head1NoDataCarrier" switches to FALSE.

4. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the reading, there was a read/write tag in the detection range and "Head1NoDataCarrier" remains zero-active.

9.7 Enhanced Read SpecialFixcode: (Head 1)

<table>
<thead>
<tr>
<th>Define command parameters:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1SingleEnhanced := 1</td>
</tr>
<tr>
<td>Head1SpecialFixcode := 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Start command execution:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Read := 1 (positive edge)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command is processing:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Busy := 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data read in and command active:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Busy := 1</td>
</tr>
<tr>
<td>Head1NoDataCarrier := 0</td>
</tr>
<tr>
<td>Head1Done := 1 (positive edge)</td>
</tr>
<tr>
<td>Usable data:</td>
</tr>
<tr>
<td>Head_1.InData.UserData</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No read/write tag present and command active:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Busy := 1</td>
</tr>
<tr>
<td>Head1NoDataCarrier := 1</td>
</tr>
<tr>
<td>Head1Done := 0 (positive edge)</td>
</tr>
<tr>
<td>Usable data:</td>
</tr>
<tr>
<td>none</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Read/write tag moving out of the read range and no read/write tag present:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Busy := 1</td>
</tr>
<tr>
<td>Head1NoDataCarrier := 1</td>
</tr>
<tr>
<td>Head1Done := 0 (positive edge)</td>
</tr>
<tr>
<td>Usable data: none</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Read/write tag moving into the read range and data read in:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Busy := 1</td>
</tr>
<tr>
<td>Head1NoDataCarrier := 1</td>
</tr>
<tr>
<td>Head1Done := 0 (positive edge)</td>
</tr>
<tr>
<td>Usable data: Head_1.InData.UserData</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>End enhanced command:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Quit := 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command complete:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head1Done := 1 (positive edge)</td>
</tr>
<tr>
<td>Head1Busy := 0</td>
</tr>
</tbody>
</table>

1. "Head1SingleEnhanced" is TRUE in order to initialize an enhanced command. "Head1DataFixcode" is irrelevant as "Head1SpecialFixcode" has higher priority than...
Words or Fixcode (i.e., as soon as "Head1SpecialFixcode" is TRUE, "Head1DataFixcode" becomes invalid). "Head1SpecialFixcode" is TRUE in order to access the SpecialFixcode/EPC. Enhanced Read SpecialFixcode is started, "Head1Read" is TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. "Head1NoDataCarrier" is TRUE as there is no read/write tag in range at this time.

3. Read/write tag enters detection range. "Head1NoDataCarrier" switches to FALSE, "Head1Done" switches to TRUE. The read/write tag exits the detection range shortly afterward.

4. "Head1Quit" cancels continuous reading. "Head1Busy" switches to FALSE, "Head1Done" is now TRUE.

5. New Enhanced Read SpecialFixcode command started. Read/write tag detected; "Head1NoDataCarrier" is FALSE, "Head1Done" is active.

6. "Head1Done" is FALSE as there is no read/write tag in the detection range; "Head1NoDataCarrier" is TRUE. Read/write tag enters the detection range again shortly afterward.

7. Cancel command with "Head1Quit"

9.8 Single Write Words: (Head 1; 15 Blocks from Address 0)

---

**Define command parameters:**

- Head1WordNum := 15 (usable data count)
- Head1WordAddress := 0 (memory address)
- Head1SingleEnhanced := 0
- Head1DataFixcode := 0

**Define usable data in the output data field:**

- Head_1.OutData.UserData := XXXX

**Start command execution:**

- Head1Write := 1 (positive edge)

**Command is processing:**

- Head1Busy := 1

**Data written:**

- Head1Busy := 0
- Head1NoDataCarrier := 0
- Head1Done := 1 (positive edge)

**No read/write tag present:**

- Head1Busy := 0
- Head1NoDataCarrier := 1
- Head1Done := 1 (positive edge)

---

1. "Head1SingleEnhanced" is zero-active in order to initialize a single command.
2. "Head1DataFixcode" is FALSE in order to access Words/User Data. Single Write
Words is started by "Head1Write" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the writing, there was no read/write tag in the detection range and "Head1NoDataCarrier" switches to TRUE.

3. Single Write Words is started by "Head1Write" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE. "Head1NoDataCarrier" switches to FALSE.

4. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of writing, there was a read/write tag in the detection range and "Head1NoDataCarrier" remains zero-active.

9.9 Enhanced Write Words: (Head 1; 15 Blocks from Address 0)

Define command parameters:
Head1WordNum := 15 (usable data count)
Head1WordAddress := 0 (memory address)
Head1SingleEnhanced := 1
Head1DataFixcode := 0

Define usable data in the output data field:
Head_1.OutData.UserData:= XXXX

Start command execution:
Head1Write := 1 (positive edge)
Command is processing:
Head1Busy := 1

Data written and command active:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1

No read/write tag present and command active:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0

Read/write tag moving out of range and no read/write tag present:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)

Read/write tag moving into range and data written:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)

End enhanced command:
Head1Quit := 1
Command complete:
Head1Done := 1 (positive edge)
Head1Busy := 0
1. "Head1SingleEnhanced" is TRUE in order to initialize an enhanced command. "Head1DataFixcode" is FALSE in order to access Words/User Data. Enhanced Write Words is started, "Head1Write" is TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. "Head1NoDataCarrier" is TRUE as there is no read/write tag in range at this time.

3. Read/write tag enters detection range. "Head1NoDataCarrier" switches to FALSE, "Head1Done" switches to TRUE. The read/write tag exits the detection range shortly afterward.

4. "Head1Quit" cancels continuous writing. "Head1Busy" switches to FALSE, "Head1Done" is now TRUE.

5. New Enhanced Write Words command started. Read/write tag detected; "Head1NoDataCarrier" is FALSE, "Head1Done" is active.

6. "Head1Done" is FALSE as there is no read/write tag in the detection range; "Head1NoDataCarrier" is TRUE. Read/write tag enters the detection range again shortly afterward.

7. Cancel command with "Head1Quit"

9.10 Single Write Fixcode: (Head 1; IPC11)

Define command parameters:
- Head_1.SpecialCommand.CommandCode := 0x1F
- Head_1.SpecialCommand.Channel := 0x50 (Fixcode length)
- Head_1.SpecialCommand.Parameter1 := FixType (high byte)
- Head_1.SpecialCommand.Parameter2 := FixType (low byte)
- Head_1.SpecialCommand.Parameter3 := Fixcode byte 1
- Head_1.SpecialCommand.Parameter4 := Fixcode byte 2
- Head_1.SpecialCommand.Parameter5 := Fixcode byte 3
- Head_1.SpecialCommand.Parameter6 := Fixcode byte 4
- Head_1.SpecialCommand.Parameter7 := Fixcode byte 5

Start command execution:
- Head1SpecialCommand := 1 (positive edge)

Command is processing:
- Head1Busy := 1

Command complete:
- Head1Done := 1 (positive edge)
- Head1Busy := 0
1. "Head1SingleEnhanced" is zero-active in order to initialize a single command. "Head1DataFixcode" is TRUE in order to access the Fixcode/TID. Single Write Fixcode is started by "Head1Write" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the writing, there was no read/write tag in the detection range and "Head1NoDataCarrier" switches to TRUE.

3. Single Write Fixcode is started by "Head1Write" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE. "Head1NoDataCarrier" switches to FALSE.

4. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of writing, there was a read/write tag in the detection range and "Head1NoDataCarrier" remains zero-active.

9.11 Single Write SpecialFixcode: (Kopf 1)

Define command parameters:

```
Head1SpecialFixcode := 1
Head1SingleEnhanced := 0
Head1WordNum := 14
Head_1.OutData.Wortadr_High := B#16#00 (reserved)
Head_1.OutData.Wortadr_Low := B#16#00 (reserved)
Head_1.OutData.UserData.DW1 := DW#16#30001122
Head_1.OutData.UserData.DW2 := DW#16#33445566
Head_1.OutData.UserData.DW3 := DW#16#77889900
Head_1.OutData.UserData.DW4 := DW#16#AABB0000
```

Start command execution:

```
Head1Write := 1 (positive edge)
```

Command is processing:

```
Head1Busy := 1
```

Command complete:

```
Head1Done := 1 (positive edge)
Head1Busy := 0
```

⚠️ **Caution!**

Note the correct notation of the PC (in accordance with EPC Gen 2 [ISO/IEC 18000-63])! The read/write tag could be rendered unusable if the configuration is incorrect!
1. "Head1SingleEnhanced" is zero-active in order to initialize a single command. "Head1DataFixcode" is irrelevant as "Head1SpecialFixcode" has higher priority than Words or Fixcode (i.e., as soon as "Head1SpecialFixcode" is TRUE, "Head1DataFixcode" becomes invalid). "Head1DataFixcode" is TRUE in order to access the SpecialFixcode/EPC. Single Write SpecialFixcode is started by "Head1Write" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE.

2. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of the writing, there was no read/write tag in the detection range and "Head1NoDataCarrier" switches to TRUE.

3. Single Write SpecialFixcode is started by "Head1Write" switching to TRUE. "Head1Busy" switches to TRUE; at the same time, "Head1Done" switches to FALSE. "Head1NoDataCarrier" switches to FALSE.

4. Once the command is executed, "Head1Busy" is zero-active again and "Head1Done" returns to TRUE. At the time of writing, there was a read/write tag in the detection range and "Head1NoDataCarrier" remains zero-active.

9.12 Error Handling: (Head 1)

1. Random command execution started.

2. An error occurred after the second read command; "Head1Error" is TRUE.

3. Errors disable the entire block. The block ignores subsequent commands. "Head1Read" has no effect while "Head1Error" is TRUE.

4. "QuitErrorHead1" acknowledges the error and enables the block.

5. Commands can once again be executed normally.

9.13 Command List (Prefetch): (Head 1)

Multiple commands can be processed in sequence using the command list. The command list must first be opened. The commands to be executed will then be transferred to the list and,
when completed, the command list will be closed. The list is processed as soon as it is activated in single or enhanced mode. The command lists are stored in a volatile state. The special command is used to open, close, and activate the command lists, as well as transfer the commands to be executed.

Open the command list:

Define command parameters: Open the command list
Head_1.SpecialCommand.CommandCode := 0xAC
Head_1.SpecialCommand.Channel := 0x00
Head_1.SpecialCommand.Parameter1 := 0x00 (ListNo)
Head_1.SpecialCommand.Parameter2 := 0x01 (ListModus)

Start command execution: Command transfer
Head1SpecialCommand := 1 (positive edge)

Command processing: Transfer in progress
Head1Busy := 1

Command complete: Transfer complete
Head1Done := 1 (positive edge)
Head1Busy := 0

After opening the command list, the commands to be executed are transferred to the control interface by the special command. The Enhanced Read Fixcode command is then parameterized, followed by the Enhanced Read Words command.

Command 1: Enhanced Read Fixcode head 1

Define command parameters: Enhanced Read Fixcode
Head_1.SpecialCommand.CommandCode := 0x01
Head_1.SpecialCommand.Channel := 0x00

Start command execution: Command transfer
Head1SpecialCommand := 1 (positive edge)

Command processing: Transfer in progress
Head1Busy := 1

Command complete: Transfer complete
Head1Done := 1 (positive edge)
Head1Busy := 0
**Command 2: Enhanced Read Words**

Define command parameters: Enhanced Read Words

- Head_1.SpecialCommand.CommandCode := 0x19
- Head_1.SpecialCommand.Channel := 0xF0 (number of data blocks)
- Head_1.SpecialCommand.Parameter1 := 0x00 (WordAddress High Byte)
- Head_1.SpecialCommand.Parameter2 := 0x00 (WordAddress Low Byte)

Start command execution: Command transfer

- Head1SpecialCommand := 1 (positive edge)

Command processing: Transfer in progress

- Head1Busy := 1

Command complete: Transfer complete

- Head1Done := 1 (positive edge)
- Head1Busy := 0

The number of parameterized commands can be increased to a maximum of 10. Parameterization is completed by closing the command list.

Close the command list:

Define command parameters: Close the command list

- Head_1.SpecialCommand.CommandCode := 0xAC
- Head_1.SpecialCommand.Channel := 0x00
- Head_1.SpecialCommand.Parameter1 := 0x00 (ListNo)
- Head_1.SpecialCommand.Parameter2 := 0x00 (ListModus)

Start command execution: Command transfer

- Head1SpecialCommand := 1 (positive edge)

Command processing: Transfer in progress

- Head1Busy := 1

Command complete: Transfer complete

- Head1Done := 1 (positive edge)
- Head1Busy := 0

The command list is executed as soon as it is activated. The command list can be activated in single or enhanced mode.
Activation in single mode:

Define command parameters: Activate single mode

- Head_1.SpecialCommand.CommandCode := 0xAC
- Head_1.SpecialCommand.Channel := 0x00
- Head_1.SpecialCommand.Parameter1 := 0x00 (ListNo)
- Head_1.SpecialCommand.Parameter2 := 0x02 (ListModus)

Start command execution: Command transfer

- Head1SpecialCommand := 1 (positive edge)

Command processing: Transfer in progress

- Head1Busy := 1

Command complete: Transfer complete

- Head1Done := 1 (positive edge)
- Head1Busy := 0

Once the command list is activated, the commands are processed in sequence. The usable data that has been read in is located in the data structure Head_1.InData.UserData. If there is a read/write tag in the detection range, the status (Head1Status) of all response telegrams has the value 0x00. The execution counter (Head1ReplyCounter) can be used to distinguish between the various successive response telegrams. When a new response telegram is received, the value of this variable in relation to the previous telegram has changed. Once a run through the command list is completed, a response telegram is received with the status (Head1Status) 0x0F. When the command list is activated in single mode, one run through the command list is completed. Following receipt of the last telegram (Head1Status = 0x0F), execution is stopped automatically. When enhanced mode is activated, the execution of the command list starts again from the beginning following receipt of the last telegram.
### Table of Read/Write Tags

<table>
<thead>
<tr>
<th>Name</th>
<th>TagType</th>
<th>Access</th>
<th>Fixcode</th>
<th>Data</th>
<th>Word Address</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPC02</td>
<td>W#16#3032</td>
<td>Read Fixcode</td>
<td>5 Byte</td>
<td>-</td>
<td>-</td>
<td>125kHz</td>
</tr>
<tr>
<td>IPC03</td>
<td>W#16#3033</td>
<td>Read Fixcode Read Words Write Words</td>
<td>4 Byte</td>
<td>116 Byte</td>
<td>0000…001C</td>
<td>125kHz</td>
</tr>
<tr>
<td>IPC11</td>
<td>W#16#3131</td>
<td>Read Fixcode Write Fixcode</td>
<td>5 Byte</td>
<td>-</td>
<td>-</td>
<td>125kHz</td>
</tr>
<tr>
<td>IPC12</td>
<td>W#16#3132</td>
<td>Read Fixcode Read Words Write Words</td>
<td>4 Byte</td>
<td>8192 Byte</td>
<td>0000…07FF</td>
<td>125kHz</td>
</tr>
<tr>
<td>IUC72</td>
<td>W#16#3830</td>
<td>Read Fixcode Write Fixcode Read Word Write Words Read Special Fixcode Write Special Fixcode</td>
<td>8 Byte</td>
<td>64 Byte</td>
<td></td>
<td>865 … 868 MHz</td>
</tr>
<tr>
<td>IUC73</td>
<td>W#16#3830</td>
<td>Read Fixcode Read Special Fixcode</td>
<td>8 Byte</td>
<td>-</td>
<td>-</td>
<td>868 MHz</td>
</tr>
<tr>
<td>IUC74</td>
<td>W#16#3830</td>
<td>Read Fixcode Write Fixcode Read Words Write Words Read Special Fixcode Write Special Fixcode</td>
<td>8 Byte</td>
<td>28 Byte</td>
<td></td>
<td>865 … 871 MHz</td>
</tr>
<tr>
<td>IUC75</td>
<td>W#16#3830</td>
<td>Read Special Fixcode</td>
<td>96 Byte</td>
<td>-</td>
<td>-</td>
<td>868 MHz</td>
</tr>
<tr>
<td>IUC80</td>
<td>W#16#3830</td>
<td>Read Fixcode Write Fixcode Read Words Write Words Read Special Fixcode Write Special Fixcode</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>868 MHz</td>
</tr>
<tr>
<td>IQC21</td>
<td>W#16#3231</td>
<td>Read Fixcode Read Words Write Words</td>
<td>8 Byte</td>
<td>112 Byte</td>
<td>0000…001B</td>
<td>13,56MHz</td>
</tr>
<tr>
<td>IQC22</td>
<td>W#16#3232</td>
<td>Read Fixcode Read Words Write Words</td>
<td>8 Byte</td>
<td>256 Byte</td>
<td>0000…003F</td>
<td>13,56MHz</td>
</tr>
<tr>
<td>IQC23</td>
<td>W#16#3233</td>
<td>Read Fixcode Read Words Write Words</td>
<td>8 Byte</td>
<td>224 Byte</td>
<td>0000…0037</td>
<td>13,56MHz</td>
</tr>
<tr>
<td>IQC24</td>
<td>W#16#3234</td>
<td>Read Fixcode Read Words Write Words</td>
<td>8 Byte</td>
<td>928 Byte</td>
<td>0000…00E7</td>
<td>13,56MHz</td>
</tr>
<tr>
<td>IQC31</td>
<td>W#16#3331</td>
<td>Read Fixcode Read Words Write Words</td>
<td>8 Byte</td>
<td>32 Byte</td>
<td>0000…0007</td>
<td>13,56MHz</td>
</tr>
<tr>
<td>IQC33</td>
<td>W#16#3333</td>
<td>Read Fixcode Read Words Write Words</td>
<td>8 Byte</td>
<td>2000 Byte</td>
<td>0000…00F9</td>
<td>13,56MHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>------------------------------</td>
<td>---</td>
<td>---</td>
<td>-----------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
| IQC35 | W#16#3335 | Read Fixcode  
Read Words  
Write Words  |  8 Byte  | 256 Byte  | 0000…003F | 13,56MHz |
| MVC | W#16#3630 | Read Fixcode  
Read Words  
Write Words  |  8 Byte  | 7552 Byte  | 0000…075F | 2,45GHz |
## 11 Table Status Values

<table>
<thead>
<tr>
<th>Head1(2) Status</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Command has been executed without error</td>
<td>None; a new command has been sent to this channel</td>
</tr>
<tr>
<td>0xFF</td>
<td>Processing command</td>
<td>Command is being processed on this channel; a command can be sent to another channel (not the same channel)</td>
</tr>
<tr>
<td>0x01</td>
<td>Low battery (only applies to MVC read/write tags)</td>
<td>Usable data is included in the same telegram; it is recommended that the battery or the whole read/write tag be replaced</td>
</tr>
<tr>
<td>0x02</td>
<td>Switch-on message; reset command was executed</td>
<td>Control interface is ready for command execution</td>
</tr>
<tr>
<td>0x04</td>
<td>Incorrect or incomplete command; command parameter not in the valid range</td>
<td>Check the command parameters and the tag type (IQC33 has only even-numbered multiples of the WordNum); check the installation condition of the head (Is the head earthed? Has shielded cable been used?)</td>
</tr>
<tr>
<td>0x05</td>
<td>There is no read/write tag in the detection range.</td>
<td>Check the distance between the read/write tag and head; check the installation condition of the head (Is the head earthed? Has shielded cable been used?)</td>
</tr>
<tr>
<td>0x06</td>
<td>Hardware error; no read/write head connected to this channel; read/write head is faulty</td>
<td>Check the reader cable (shielded cable V1-G-XM-PUR ABG-V1-W); check the LED of the head (off: head faulty; flashing: initialization required; constant: head is OK)</td>
</tr>
<tr>
<td>0x07</td>
<td>Internal device error</td>
<td>Internal memory overflow (reduce the data hold time)</td>
</tr>
<tr>
<td>0x09</td>
<td>Parameterized tag type is not compatible with the connected reader</td>
<td>Check the defined tag type</td>
</tr>
<tr>
<td>0x0A</td>
<td>There are multiple tags in the detection range.</td>
<td>Make sure that only one tag is in the detection range during command execution</td>
</tr>
<tr>
<td>0x0B</td>
<td>Additional telegram with IF parameter switched on</td>
<td>None; telegram with additional information was sent</td>
</tr>
<tr>
<td>0x0F</td>
<td>Last telegram in multiframe mode</td>
<td>None; after a single command, number of tags read in the detection range in ASCII</td>
</tr>
<tr>
<td>0x20</td>
<td>Switch-on message; reset command was executed</td>
<td>Error was sent by the bus interface; control interface is ready for command execution</td>
</tr>
<tr>
<td>0x40</td>
<td>Incorrect or incomplete command; command parameter not in the valid range</td>
<td>Error was sent by the bus interface; check the command parameters and the tag type (IQC33 has only even-numbered multiples of the WordNum); check the installation condition of the head (Is the head earthed? Has shielded cable been used?)</td>
</tr>
<tr>
<td>0x60</td>
<td>Hardware error; no read/write head connected to this channel; read/write head is faulty</td>
<td>Error was sent by the bus interface; check the reader cable (shielded cable V1-G-XM-PUR ABG-V1-W); check the LED of the head (off: head faulty; flashing: initialization required; constant: head is OK)</td>
</tr>
<tr>
<td>0x70</td>
<td>Internal device error</td>
<td>Internal memory overflow (reduce the data hold time)</td>
</tr>
</tbody>
</table>
## Table Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change to Function Block</th>
<th>Change to Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>11/18/2008</td>
<td>Adjustment from 4-channel version to 2-channel version</td>
<td>First version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Replacing the IN variable &quot;IDENTControlAddress&quot; with the IN variables INPUT/OUTPUT_Address and INPUT/OUTPUT_Length</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output data length of 4 bytes possible through implementation of SFC81</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction of OUT variables Done; NoDataCarrier; Busy; and Error</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>02/28/2009</td>
<td>Variable Memory.Error_SFC_14 is connected to the variables Head1/2Error</td>
<td>New image introduced for the overview of the function block variables (page 4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change to check of variables Head_1/2.ExistTC and Head_1/2.NotExist when parameterizing the commands -&gt; trigger mode</td>
<td>New variables introduced in the function description table (page 4 – 5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction of the status check on the value 0x0F -&gt; last telegram of the command list</td>
<td>Information on the maximum telegram length for different CPU series (page 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change to the consistency check of the input and output data fields -&gt; When executing the command list, the input and output data field are not equal</td>
<td>Change to the variable name of the status value (page 6 – 7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resetting the variable Head_1/2.QuitError in the Restart and QuitError routine</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction of new symbol names for the usable data fields of the input and output data fields -&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Head1/2.InData/OutData.UserData</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Introduction of a value check of the parameter Head1/2.WordNum -&gt; Value &gt; 15 is Head1/2.Error set</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output parameters Head1/2.Status and Head1/2.ReplyCounter introduced -&gt; battery status and command list</td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td>04/24/2009</td>
<td>Text library inserted for allocation of status messages</td>
<td>None</td>
</tr>
<tr>
<td>2.8</td>
<td>08/02/2013</td>
<td>Implementation of multiframe mode</td>
<td>Parameterization options with FB190 explained (page 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameterization of IUH readers now possible</td>
<td>New multiframe block FB33 explained (page 10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SpecialFixcode (EPC) readable/writeable</td>
<td>Update to various status values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data blocks for processing the different telegrams on channel 1/2 inserted (DB70X &amp; DB71X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SetRestart deletes content of the data blocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>In/Out HeadXNewData displays new data in data blocks</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HeadXCacheFull indicates data block overflow</td>
<td></td>
</tr>
<tr>
<td>2.9</td>
<td>09/04/2013</td>
<td>Filter functions added to FB33</td>
<td>Filter functions explained (page 17)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FB190 displays incorrectly configured parameters</td>
<td>Fault analysis for parameter commands (page 8)</td>
</tr>
<tr>
<td>3.0</td>
<td>10/25/2013</td>
<td>FB190: Internal processes optimized, parameter error and parameter type error are now displayed correctly, parameters E5 and AP added</td>
<td>Status diagrams for all commands added Identification of inputs and outputs on the block Manual adapted to changelog</td>
</tr>
</tbody>
</table>