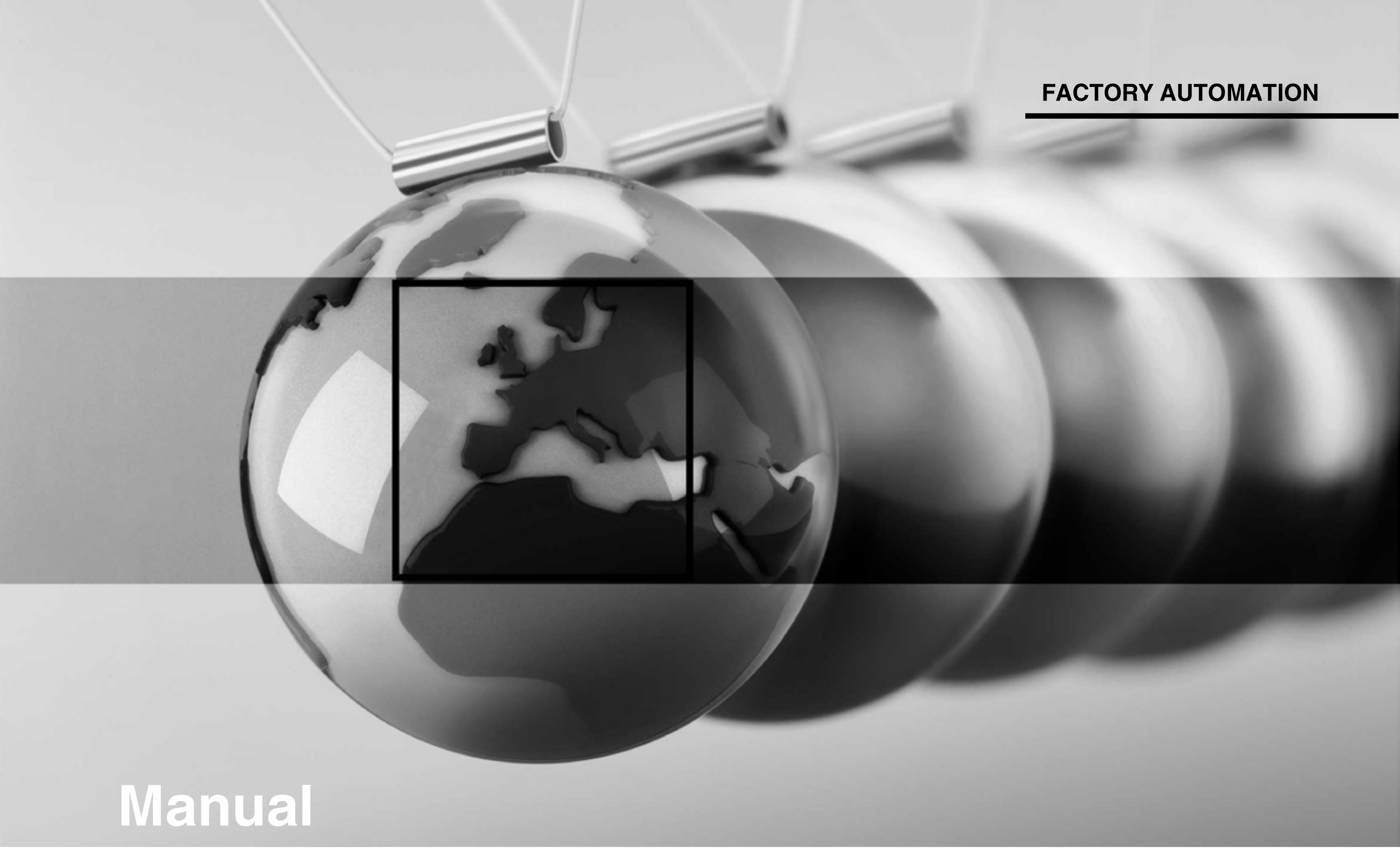


Manual

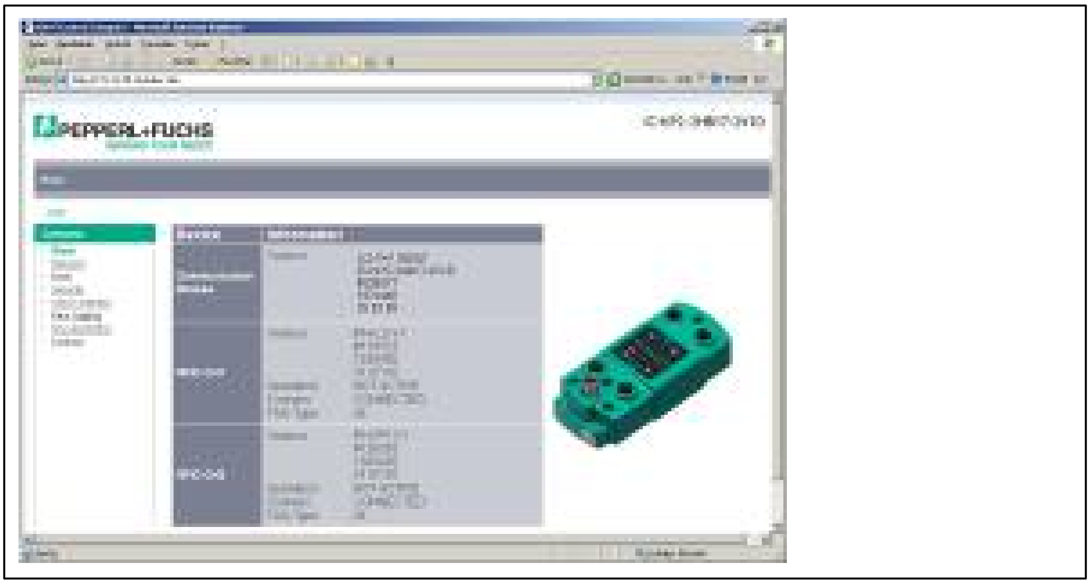
COMMISSIONING IDENTCONTROL COMPACT IC-KP2-2HB17-2V1D WITH PROFINET PROTOCOL TO SIEMENS S7 CONTROLLER

PEPPERL+FUCHS
SENSING YOUR NEEDS

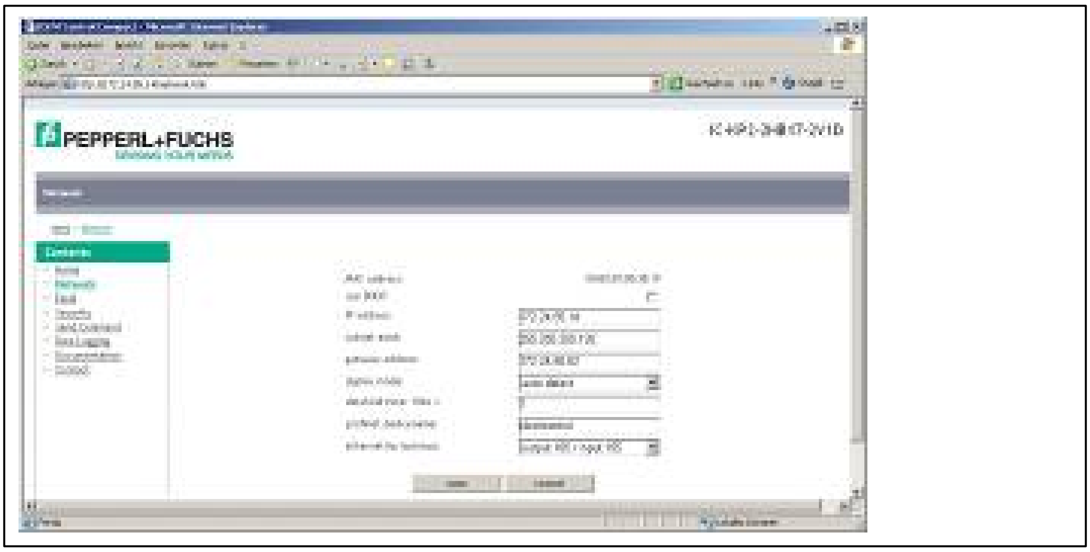
With regard to the supply of products, the current issue of the following document is applicable: The General Terms of Delivery for Products and Services of the Electrical Industry, published by the Central Association of the Electrical Industry (Zentralverband Elektrotechnik und Elektroindustrie (ZVEI)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

1 Setting Communication Parameters 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.

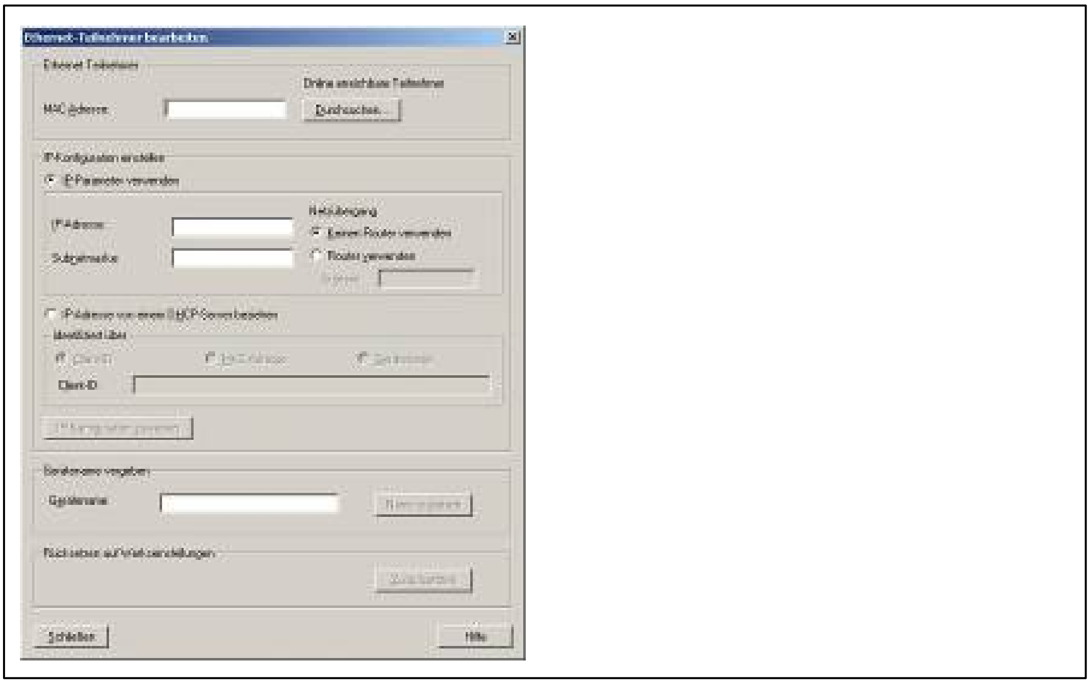


The communication parameters and the PROFINET name can be parameterized by selecting "Network".



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.

2014-02

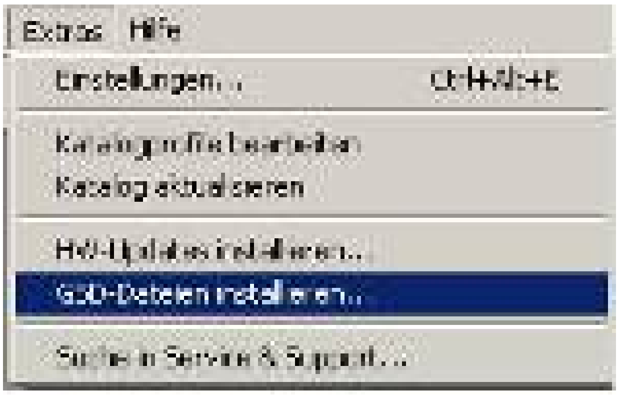
2 Installing the GSDML file

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.

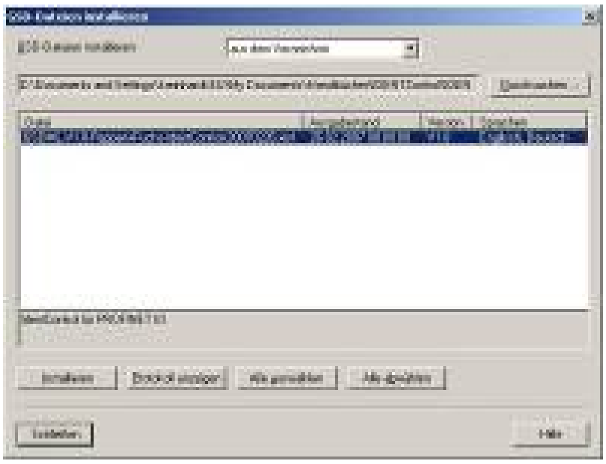
www.pepperl-fuchs.com

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

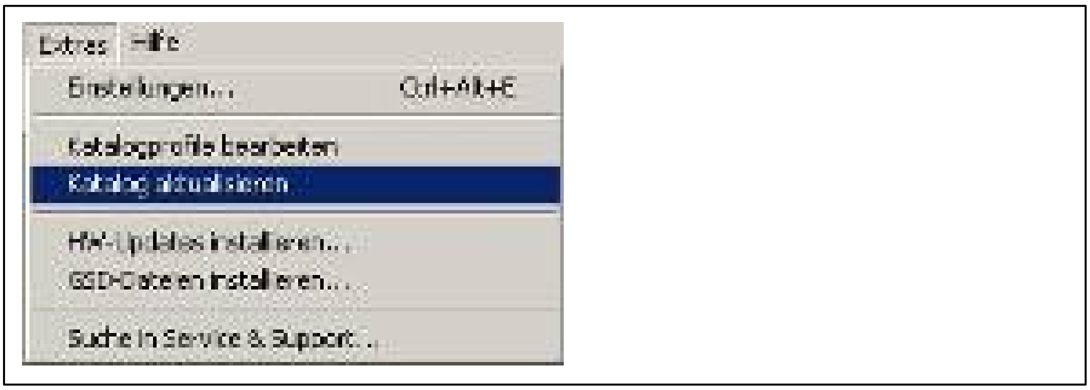
To install the file, select menu item "Options" → "Install GSD File..." from the Simatic hardware configuration menu.


Extras
Hilfe
Einstellungen...
Ctrl+Alt+E
Katalogprofile bearbeiten
Katalog aktualisieren
HW-Updates installieren...
GSD-Dateien installieren...
Suche in Service & Support...

Then select the GSDML file from the source directory.

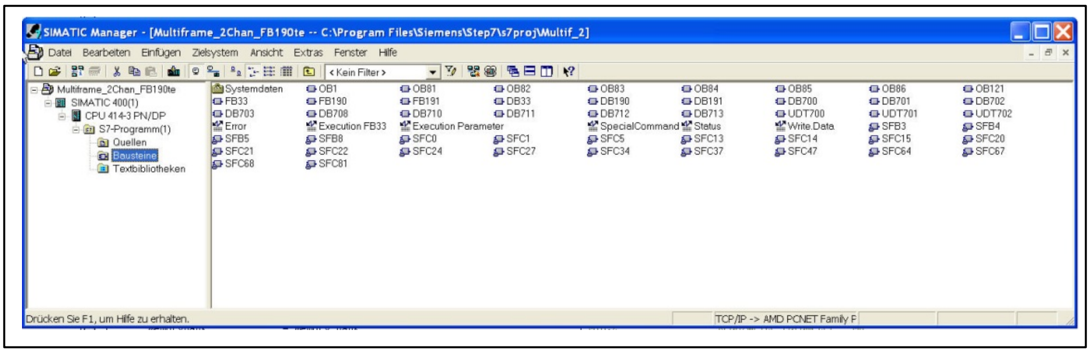


The GSDML file is added to the hardware catalog via the menu item "Options" → "Update Catalog".


Extras
Hilfe
Einstellungen... Ctrl+Alt+E
Katalogprofile bearbeiten
Katalog aktualisieren
HW-Updates installieren...
GSD-Dateien installieren...
Suche in Service & Support...

3 Installing the User Program

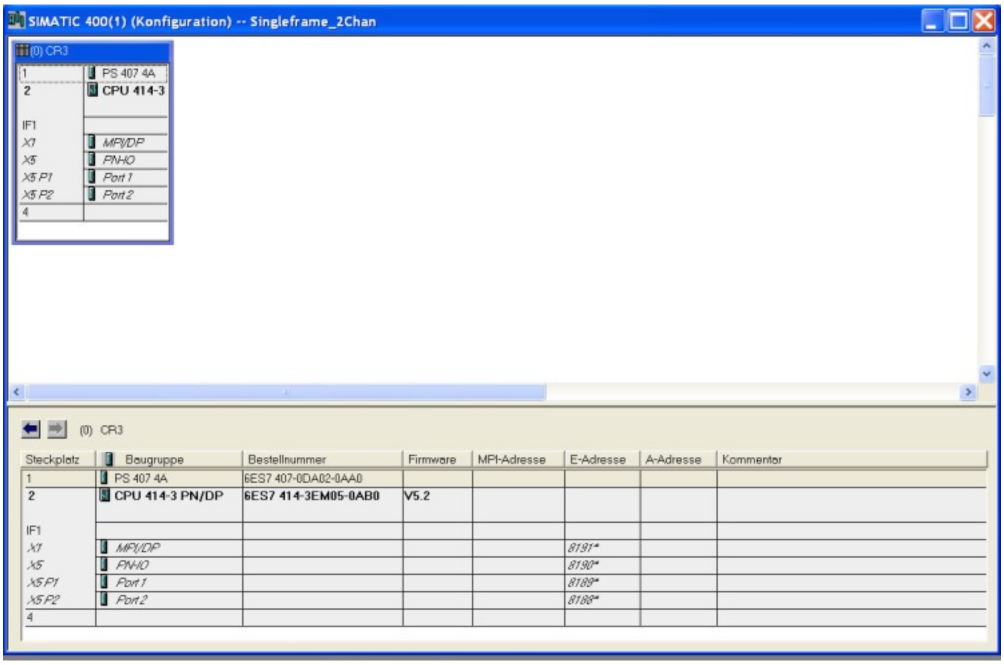
Before the user program can be installed, the file "IC-KP2-2HB17.zip" must first be unzipped. Unzip the file by selecting the menu item "File" → "Retrieve..." in SIMATIC Manager. Highlight the file, confirm with "Open", and select the relevant file path. Following successful installation, the user program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



Systemdaten	OB1	OB81	OB82	OB83	OB84	OB85	OB86	OB121
FB33	FB190	FB191	DB33	DB190	DB191	DB700	DB701	DB702
DB703	DB708	DB710	DB711	DB712	DB713	UDT700	UDT701	UDT702
Error	Execution FB33	Execution Parameter		SpecialCommand	Status	Write.Data	SFB3	SFB4
SFB5	SFB8	SFC0	SFC1	SFC5	SFC13	SFC14	SFC15	SFC20
SFC21	SFC22	SFC24	SFC27	SFC34	SFC37	SFC47	SFC64	SFC67
SFC68	SFC81							

4 Hardware Configuration

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



SIMATIC 400(1) (Konfiguration) -- Singleframe_2Chan

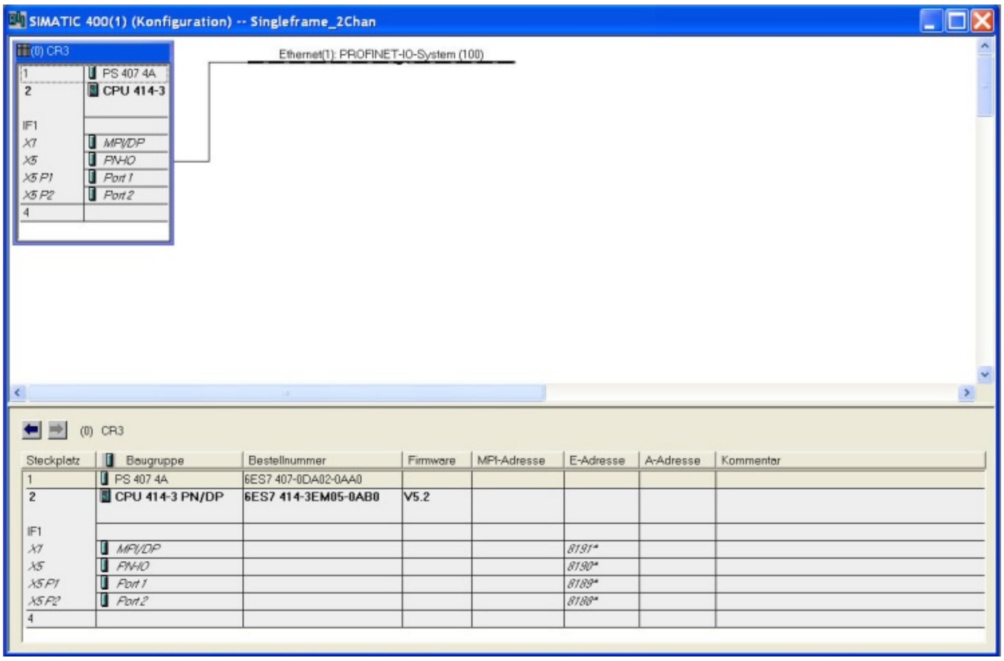
(0) CR3

1	PS 407 4A
2	CPU 414-3
IF1	
X1	MPI/DP
X5	PN-IO
X5 P1	Port 1
X5 P2	Port 2
4	

(0) CR3

Steckplatz	Baugruppe	Bestellnummer	Firmware	MPI-Adresse	E-Adresse	A-Adresse	Kommentar
1	PS 407 4A	6ES7 407-0DA02-0AA0					
2	CPU 414-3 PN/DP	6ES7 414-3EM05-0AB0	V5.2				
IF1							
X1	MPI/DP				8191*		
X5	PN-IO				8190*		
X5 P1	Port 1				8189*		
X5 P2	Port 2				8188*		
4							

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



SIMATIC 400(1) (Konfiguration) -- Singleframe_2Chan

(0) CR3

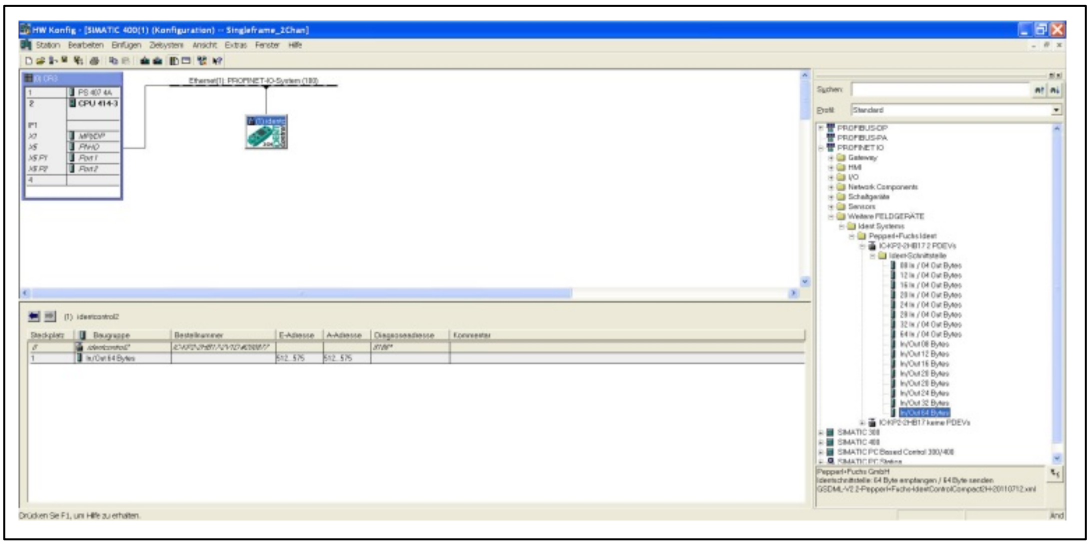
1	PS 407 4A
2	CPU 414-3
IF1	
X1	MPI/DP
X5	PN-IO
X5 P1	Port 1
X5 P2	Port 2
4	

Ethernet(1): PROFINET-IO-System (100)

(0) CR3

Steckplatz	Baugruppe	Bestellnummer	Firmware	MPI-Adresse	E-Adresse	A-Adresse	Kommentar
1	PS 407 4A	6ES7 407-0DA02-0AA0					
2	CPU 414-3 PN/DP	6ES7 414-3EM05-0AB0	V5.2				
IF1							
X1	MPI/DP				8191*		
X5	PN-IO				8190*		
X5 P1	Port 1				8189*		
X5 P2	Port 2				8188*		
4							

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.

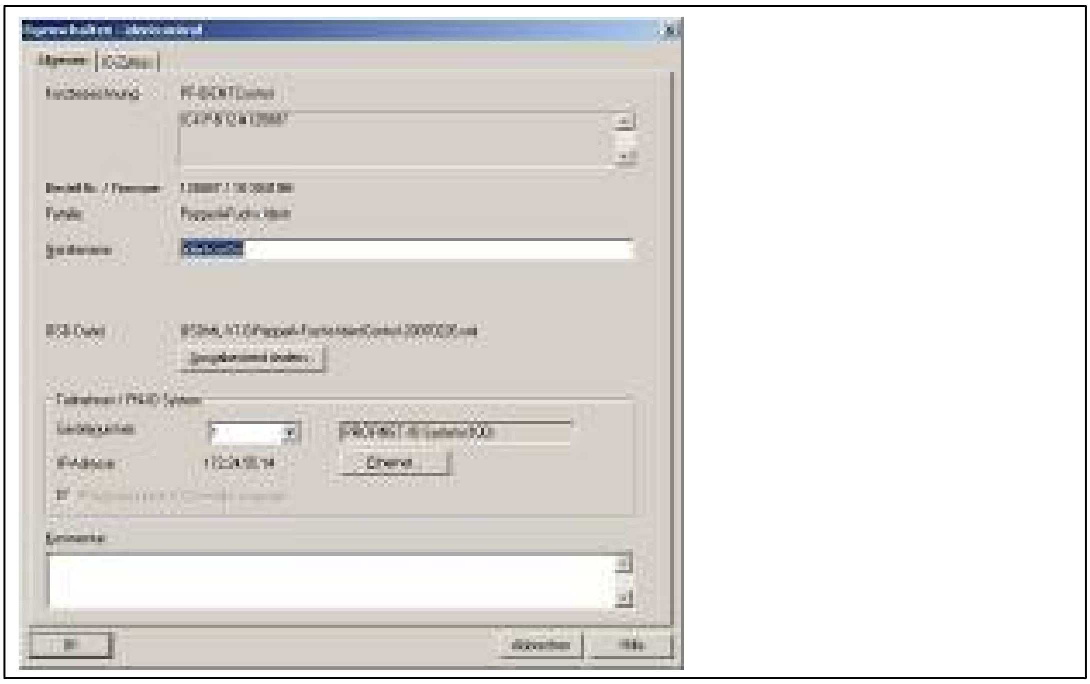


(1) identcontrol2

Steckplatz	Baugruppe	Bestellnummer	E-Adresse	A-Adresse	Diagnoseadresse	Kommentar
0	identcontrol2	IC-KP2-2HB17-2V1D #[illegible]			8188*	
1	In/Out 64 Bytes		512...575	512...575		

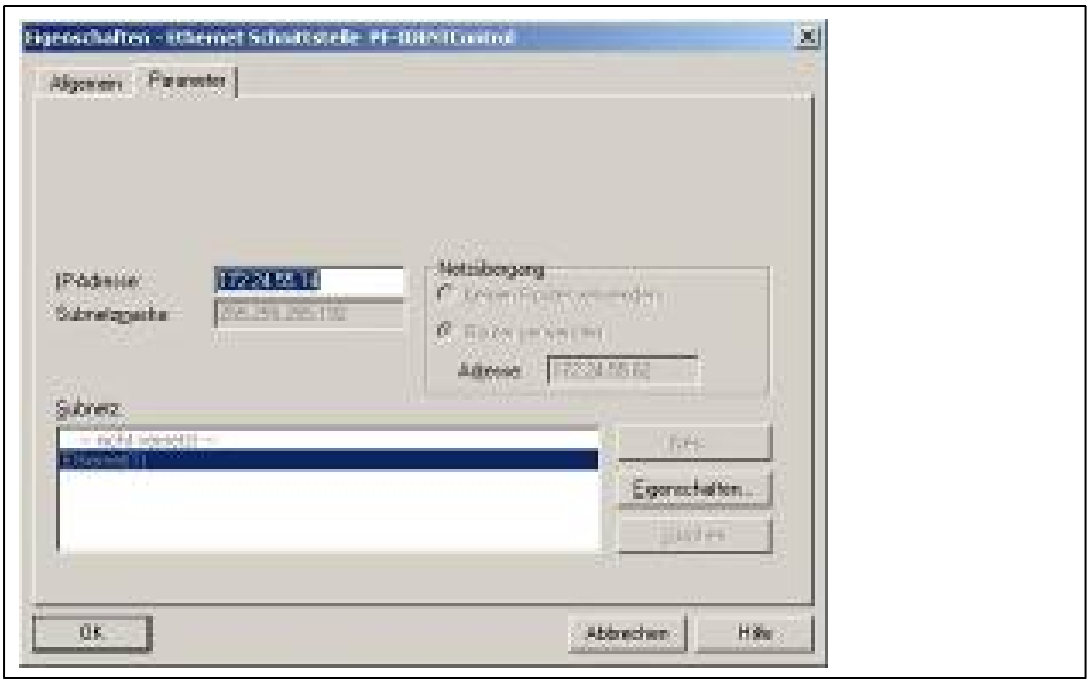
Pepperl+Fuchs GmbH
Identschnittstelle: 64 Byte empfangen / 64 Byte senden
GSDML-V2.2-Pepperl+Fuchs-IdentControlCompact2H-20110712.xml

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.

2014-02



Eigenschaften - Ethernet Schnittstelle PF-IDENTControl

IP-Adresse: 172.24.58.11

Subnetzmaske: 255.255.255.192

Netzübergang

○ Keinen Router verwenden

● Router verwenden

Adresse: 172.24.58.62

Subnetz:

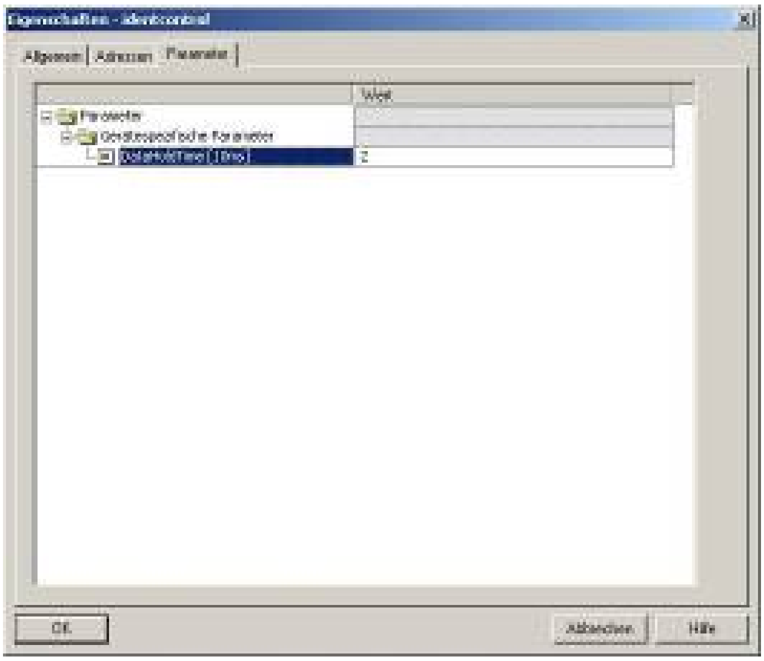
--- nicht vernetzt ---

Ethernet(1)

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.



	Wert
Parameter	
Gerätespezifische Parameter	
DataHoldTime (10ms)	2

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)

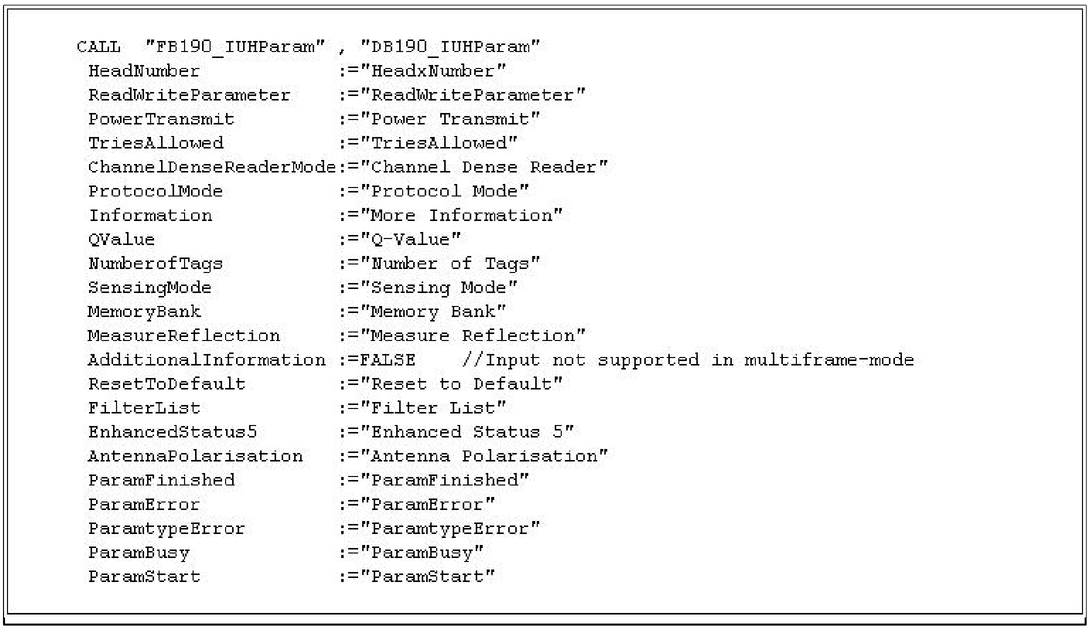
or

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.



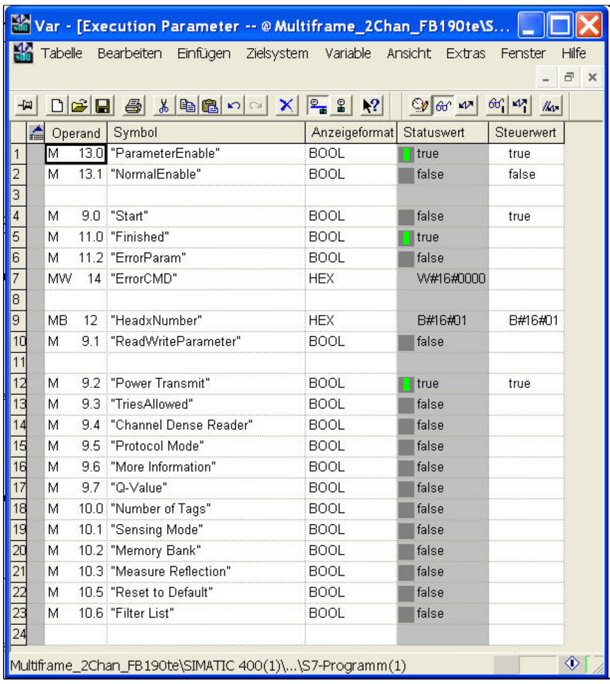
```
CALL  "FB190_IUHParam" , "DB190_IUHParam"
 HeadNumber               :="HeadxNumber"
 ReadWriteParameter       :="ReadWriteParameter"
 PowerTransmit            :="Power Transmit"
 TriesAllowed             :="TriesAllowed"
 ChannelDenseReaderMode:="Channel Dense Reader"
 ProtocolMode             :="Protocol Mode"
 Information              :="More Information"
 QValue                   :="Q-Value"
 NumberofTags             :="Number of Tags"
 SensingMode              :="Sensing Mode"
 MemoryBank               :="Memory Bank"
 MeasureReflection        :="Measure Reflection"
 AdditionalInformation :=FALSE    //Input not supported in multiframe-mode
 ResetToDefault           :="Reset to Default"
 FilterList               :="Filter List"
 EnhancedStatus5          :="Enhanced Status 5"
 AntennaPolarisation      :="Antenna Polarisation"
 ParamFinished            :="ParamFinished"
 ParamError               :="ParamError"
 ParamtypeError           :="ParamtypeError"
 ParamBusy                :="ParamBusy"
 ParamStart               :="ParamStart"
```

Name	Decla- ration	Data type	Description
HeadNumber	Input	BYTE	Head/channel number on which parameters are to be read/written
ReadWriteParameter	Input	BOOL	Execution of 0:= read, 1:= write parameters
PowerTransmit	Input	BOOL	Access to reader transmission power
TriesAllowed	Input	BOOL	Access to read/write attempts
ChannelDenseReaderMode	Input	BOOL	Access to the sequence of transmission channels permitted in dense reader mode (DRM)
ProtocolMode	Input	BOOL	Access to output protocol (singleframe and multiframe)
Information	Input	BOOL	Access to output of additional information, provided that the reading was successful and the multiframe protocol is set

QValue	Input	BOOL	Access to Q-value used to define the time slots used (2^Q) for anti-collision (Slotted Aloha)
NumberofTags	Input	BOOL	Access to the number of tags in the detection range for which the read/write head is searching
SensingMode	Input	BOOL	Access to pause time in ms, after all runs are complete
MemoryBank	Input	BOOL	Access to the bank that the read/write commands SR, ER, SW, and EW are accessing
MeasureReflection	Input	BOOL	Access to measured reflected transmission power (read only)
AdditionalInformation	Input	BOOL	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!
Reset to Default	Input	BOOL	Returns all settings of the read/write head to the default configuration
Filterlist	Input	BOOL	Writes the filter mask defined in OB1 for access to the tags in the detection range
Antenna Polarisation	Input	BOOL	Switches the polarization to linear or circular
Enhanced Status 5	Input	BOOL	Sets the number of unsuccessful write/read attempts before a status 5 is output for an enhanced command
ParamStart	Input	BOOL	Execution of parameter command (positive edge)
ParamBusy	Output	BOOL	Individual parameter command is currently being executed
ParamFinished	Output	BOOL	Parameter access terminated
ParamError	Output	BOOL	An error occurred
ParamtypeError	Output	WORD	Return of the incorrect parameter type (HEX)

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.


Var - [Execution Parameter -- @Multiframe_2Chan_FB190te\S...
Tabelle Bearbeiten Einfügen Zielsystem Variable Ansicht Extras Fenster Hilfe

	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	M 13.0	"ParameterEnable"	BOOL	🟩 true	true
2	M 13.1	"NormalEnable"	BOOL	▨ false	false
3					
4	M 9.0	"Start"	BOOL	▨ false	true
5	M 11.0	"Finished"	BOOL	🟩 true	
6	M 11.2	"ErrorParam"	BOOL	▨ false	
7	MW 14	"ErrorCMD"	HEX	W#16#0000	
8					
9	MB 12	"HeadxNumber"	HEX	B#16#01	B#16#01
10	M 9.1	"ReadWriteParameter"	BOOL	▨ false	
11					
12	M 9.2	"Power Transmit"	BOOL	🟩 true	true
13	M 9.3	"TriesAllowed"	BOOL	▨ false	
14	M 9.4	"Channel Dense Reader"	BOOL	▨ false	
15	M 9.5	"Protocol Mode"	BOOL	▨ false	
16	M 9.6	"More Information"	BOOL	▨ false	
17	M 9.7	"Q-Value"	BOOL	▨ false	
18	M 10.0	"Number of Tags"	BOOL	▨ false	
19	M 10.1	"Sensing Mode"	BOOL	▨ false	
20	M 10.2	"Memory Bank"	BOOL	▨ false	
21	M 10.3	"Measure Reflection"	BOOL	▨ false	
22	M 10.5	"Reset to Default"	BOOL	▨ false	
23	M 10.6	"Filter List"	BOOL	▨ false	
24					

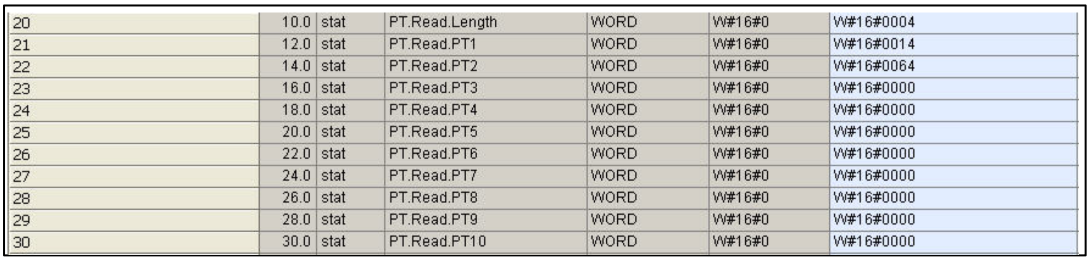
Multiframe_2Chan_FB190te\SIMATIC 400(1)\...\S7-Programm(1)

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)



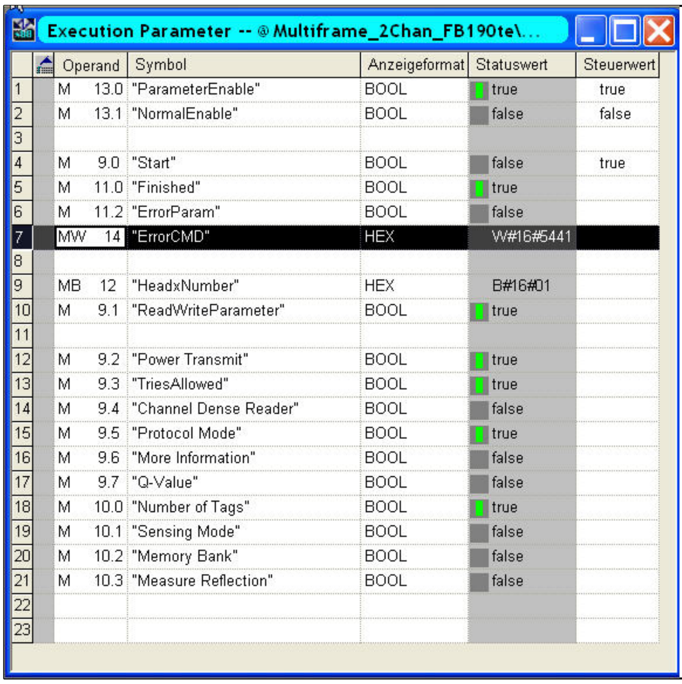
20	10.0	stat	PT.Read.Length	WORD	W#16#0	W#16#0004
21	12.0	stat	PT.Read.PT1	WORD	W#16#0	W#16#0014
22	14.0	stat	PT.Read.PT2	WORD	W#16#0	W#16#0064
23	16.0	stat	PT.Read.PT3	WORD	W#16#0	W#16#0000
24	18.0	stat	PT.Read.PT4	WORD	W#16#0	W#16#0000
25	20.0	stat	PT.Read.PT5	WORD	W#16#0	W#16#0000
26	22.0	stat	PT.Read.PT6	WORD	W#16#0	W#16#0000
27	24.0	stat	PT.Read.PT7	WORD	W#16#0	W#16#0000
28	26.0	stat	PT.Read.PT8	WORD	W#16#0	W#16#0000
29	28.0	stat	PT.Read.PT9	WORD	W#16#0	W#16#0000
30	30.0	stat	PT.Read.PT10	WORD	W#16#0	W#16#0000

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.



	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	M 13.0	"ParameterEnable"	BOOL	🟩 true	true
2	M 13.1	"NormalEnable"	BOOL	▨ false	false
3					
4	M 9.0	"Start"	BOOL	▨ false	true
5	M 11.0	"Finished"	BOOL	🟩 true	
6	M 11.2	"ErrorParam"	BOOL	▨ false	
7	MW 14	"ErrorCMD"	HEX	W#16#5441	
8					
9	MB 12	"HeadxNumber"	HEX	B#16#01	
10	M 9.1	"ReadWriteParameter"	BOOL	🟩 true	
11					
12	M 9.2	"Power Transmit"	BOOL	🟩 true	
13	M 9.3	"TriesAllowed"	BOOL	🟩 true	
14	M 9.4	"Channel Dense Reader"	BOOL	▨ false	
15	M 9.5	"Protocol Mode"	BOOL	🟩 true	
16	M 9.6	"More Information"	BOOL	▨ false	
17	M 9.7	"Q-Value"	BOOL	▨ false	
18	M 10.0	"Number of Tags"	BOOL	🟩 true	
19	M 10.1	"Sensing Mode"	BOOL	▨ false	
20	M 10.2	"Memory Bank"	BOOL	▨ false	
21	M 10.3	"Measure Reflection"	BOOL	▨ false	
22					
23					

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 (5441_h = TA_{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.

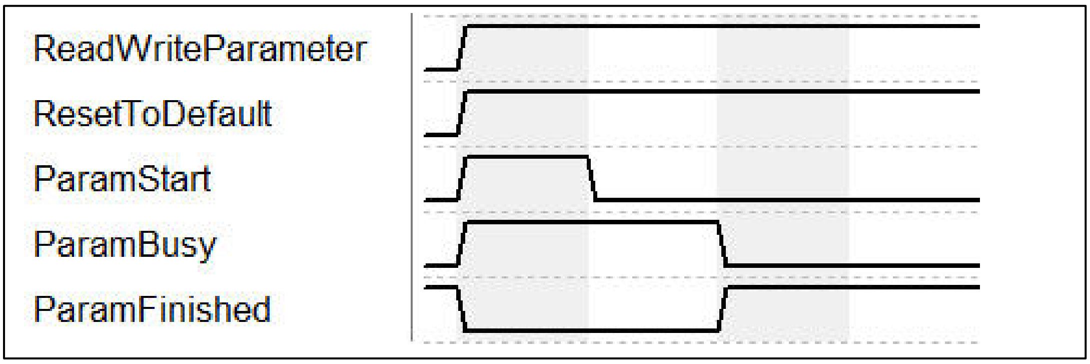
Examples command execution:

6.1 Reset to Default: Channel 1

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:

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

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

Processing complete:
ParamBusy := 0
Finish := 1
Usable data: 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 := 1
ChannelDenseReaderMode := 1
"DB190_IUHParam".CD.Write.Length := B#16#2
"DB190_IUHParam".CD.Write.ChannelA := B#16#D
"DB190_IUHParam".CD.Write.ChannelB := B#16#7

Start execution:
Start := 1 (positive edge)

Processing complete:
Finish := 1

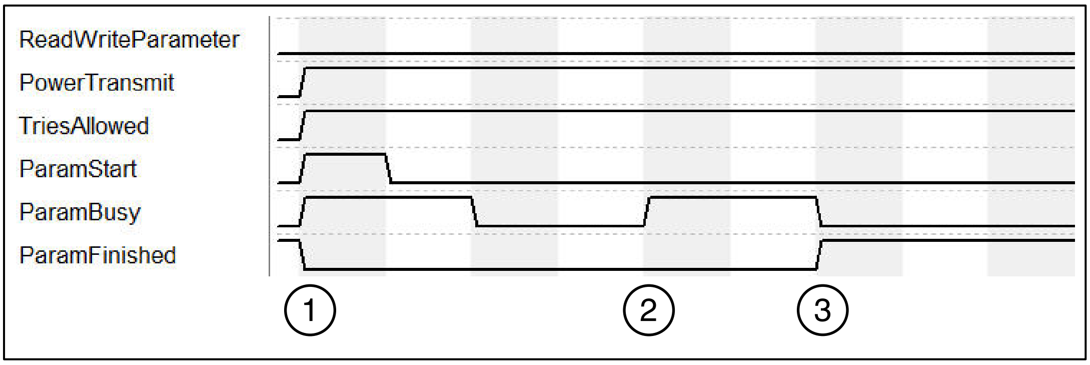
6.4 Read/Write Multiple Parameters: Channel 1

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



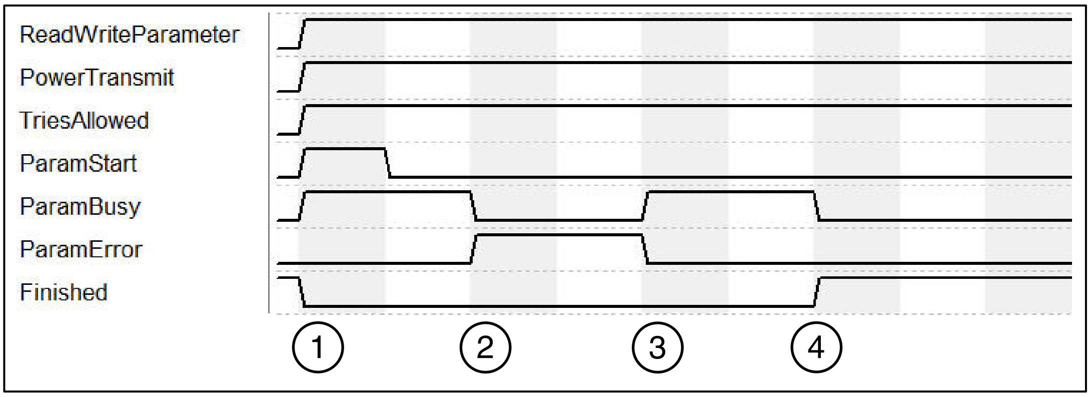
1. Command execution (read) is started with "ParamStart". "ParamFinished" switches to FALSE.
2. "ParamBusy" indicates command processing of "Power Transmit" and "Tries Allowed".
3. Once both parameters are complete, "ParamFinished" returns to TRUE.

Writing:

Define parameters:
HeadNumber := B#16#01
ReadWriteParameter := 1
PowerTransmit := 1
ChannelDenseReaderMode := 1
NumberofTags := 1
"DB190_IUHParam".PT.Write.Length := W#16#2
"DB190_IUHParam".PT.Write.PT1 := W#16#30
"DB190_IUHParam".CD.Write.Length := B#16#2
"DB190_IUHParam".CD.Write.ChannelA := B#16#D
"DB190_IUHParam".CD.Write.ChannelB := B#16#7
"DB190_IUHParam".NT.Write.NT := B#16#FF

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

Processing complete:
ParamBusy := 0
Finish := 1



1. Command execution (write) is started with "ParamStart". "ParamFinished" switches to FALSE.
2. "ParamBusy" indicates command processing of "Power Transmit" and "Tries Allowed". "PowerTransmit" parameter configured incorrectly. "ParamError" switches to TRUE.
3. "TriesAllowed' is executed. "ParamError" is reset. "ParamtypeError" output indicates incorrect parameters
4. All commands processed. "ParamFinished" returns to TRUE.

7 FB32 „Multiframe“ Function Block

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

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

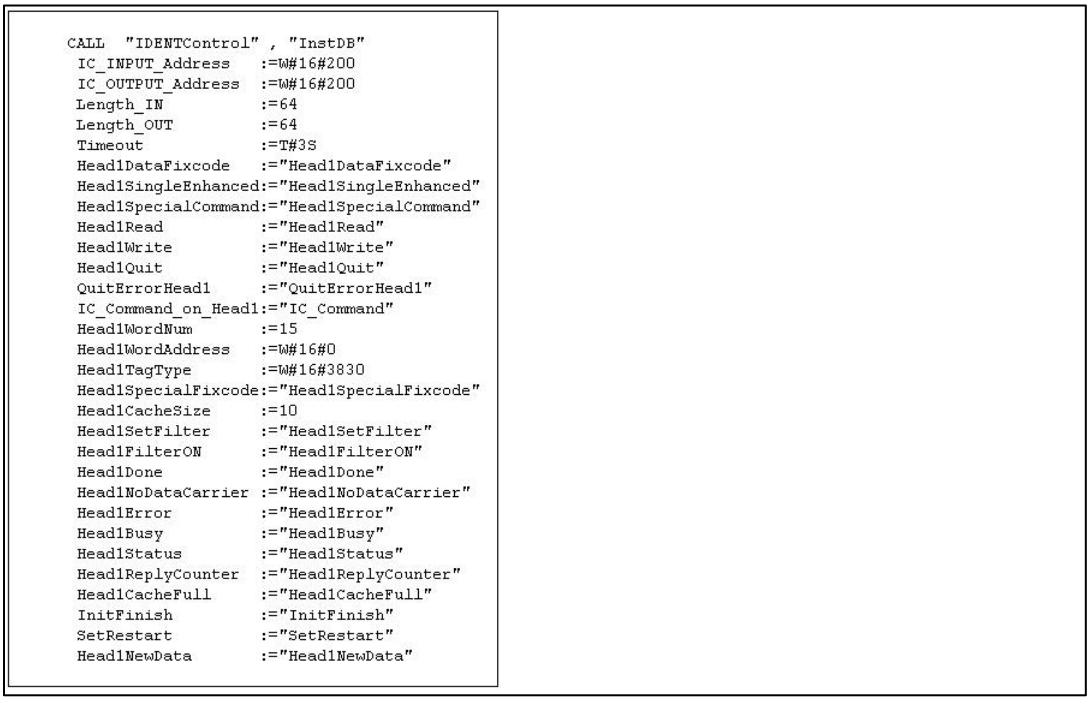
or

Call FB32, DB32

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

This block supports control interfaces with up to two channels.

The following image shows the function and the variables requiring parameterization being accessed



```
CALL  "IDENTControl" , "InstDB"
 IC_INPUT_Address   :=W#16#200
 IC_OUTPUT_Address  :=W#16#200
 Length_IN          :=64
 Length_OUT         :=64
 Timeout            :=T#3S
 Head1DataFixcode   :="Head1DataFixcode"
 Head1SingleEnhanced:="Head1SingleEnhanced"
 Head1SpecialCommand:="Head1SpecialCommand"
 Head1Read          :="Head1Read"
 Head1Write         :="Head1Write"
 Head1Quit          :="Head1Quit"
 QuitErrorHead1     :="QuitErrorHead1"
 IC_Command_on_Head1:="IC_Command"
 Head1WordNum       :=15
 Head1WordAddress   :=W#16#0
 Head1TagType       :=W#16#3830
 Head1SpecialFixcode:="Head1SpecialFixcode"
 Head1CacheSize     :=10
 Head1SetFilter     :="Head1SetFilter"
 Head1FilterON      :="Head1FilterON"
 Head1Done          :="Head1Done"
 Head1NoDataCarrier :="Head1NoDataCarrier"
 Head1Error         :="Head1Error"
 Head1Busy          :="Head1Busy"
 Head1Status        :="Head1Status"
 Head1ReplyCounter  :="Head1ReplyCounter"
 Head1CacheFull     :="Head1CacheFull"
 InitFinish         :="InitFinish"
 SetRestart         :="SetRestart"
 Head1NewData       :="Head1NewData"
```

Name	Declaration	Data type	Description
IC_INPUT_Address	Input	WORD	Start address of the control interface in the process image of the inputs (I address)
IC_OUTPUT_Address	Input	WORD	Start address of the control interface in the process image of the outputs (O address)
Length_IN	Input	INT	Length of the input image (length of the PROFIBUS telegram received)
Length_OUT	Input	INT	Length of the output image (length of the PROFIBUS telegram sent)
Timeout	Input	TIME	Timer for monitoring the response time of the control interface
Head1DataFixcode	Input	BOOL	Head 1 access to 0:=Fixcode 1:=data range
Head2DataFixcode	Input	BOOL	Head 2 access to 0:=Fixcode 1:=data range
Head1SingleEnhanced	Input	BOOL	Execution on head 1 from 0:=single 1:=enhanced command

Head2SingleEnhanced	Input	BOOL	Execution on head 2 from 0:=single 1:=enhanced command
Head1SpecialCommand	Input	BOOL	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
Head2SpecialCommand	Input	BOOL	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
Head1Read	Input	BOOL	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
Head2Read	Input	BOOL	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
Head1Write	Input	BOOL	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
Head2Write	Input	BOOL	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
Head1Quit	Input	BOOL	Execution of quit command on head 1 to cancel an enhanced command (positive edge)
Head2Quit	Input	BOOL	Execution of quit command on head 2 to cancel an enhanced command (positive edge)
QuitErrorHead1	Input	BOOL	Execution of error routine on head 1 (positive edge)
QuitErrorHead2	Input	BOOL	Execution of error routine on head 2 (positive edge)
IC_Command_Head1	Input	BOOL	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
Head1WordNum	Input	INT	Number of data blocks activated on head 1
Head2WordNum	Input	INT	Number of data blocks activated on head 2
Head1WordAddress	Input	WORD	Start address of the memory activated on channel 1 on read/write tag
Head2WordAddress	Input	WORD	Start address of the memory activated on channel 2 on read/write tag
Head1TagType	Input	WORD	Tag type channel 1 (table of tag types)
Head2TagType	Input	WORD	Tag type channel 2 (table of tag types)
Head1SpecialFixcode	Input	BOOL	Head 1 access to 1:=SpecialFixcode (EPC)
Head2SpecialFixcode	Input	BOOL	Head 2 access to 1:=SpecialFixcode (EPC)
Head1CacheSize	Input	INT	Number of telegrams in intermediate memory on head 1

Head2CacheSize	Input	INT	Number of telegrams in intermediate memory on head 2
Head1Filtersettings	Input	BOOL	Initializes the filter parameters defined in OB1 on head 1 → Head1Write
Head2Filtersettings	Input	BOOL	Initializes the filter parameters defined in OB1 on head 2 → Head2Write
Head1FilterON	Input	BOOL	Activates the filter(s) set on head 1 (positive edge) Deactivates the filter(s) set on head 1 (negative edge)
Head2FilterON	Input	BOOL	Activates the filter(s) set on head 2 (positive edge) Deactivates the filter(s) set on head 2 (negative edge)
Head1Done	Output	BOOL	New data present (enhanced) or command complete (single) on channel 1 (positive edge)
Head2Done	Output	BOOL	New data present (enhanced) or command complete (single) on channel 2 (positive edge)
Head1NoDataCarrier	Output	BOOL	No read/write tag for execution time of a command on channel 1 present
Head2NoDataCarrier	Output	BOOL	No read/write tag for execution time of a command on channel 2 present
Head1Error	Output	BOOL	Error occurred on channel 1 (positive edge)
Head2Error	Output	BOOL	Error occurred on channel 2 (positive edge)
Head1Busy	Output	BOOL	Command is processed on channel 1
Head2Busy	Output	BOOL	Command is processed on channel 2
Head1Status	Output	BYTE	Channel 1 status value
Head2Status	Output	BYTE	Channel 2 status value
Head1ReplyCounter	Output	BYTE	Channel 1 reply counter value
Head2ReplyCounter	Output	BYTE	Channel 2 reply counter value
Head1CacheFull	Output	BOOL	Telegram memory (DB) for head 1 is full
Head2CacheFull	Output	BOOL	Telegram memory (DB) for head 2 is full
InitFinish	In/ Output	BOOL	Initialization complete (positive edge)
SetRestart	In/ Output	BOOL	Execution of control interface initialization (positive edge)
Head1NewData	In/ Output	BOOL	New data present in telegram memory (DB)
Head2NewData	In/ Output	BOOL	New data present in telegram memory (DB)
Head1SetFilter	In/ Output	BOOL	Initializes the filter parameters defined in OB1 on head 1 → Head1Write
Head2SetFilter	In/ Output	BOOL	Initializes the filter parameters defined in OB1 on head 2 → Head2Write
Head1FilterON	In/ Output	BOOL	Activates the filter(s) set on head 1 (positive edge) Deactivates the filter(s) set on head 1 (negative edge)
Head2FilterON	In/ Output	BOOL	Activates the filter(s) set on head 2 (positive edge) Deactivates the filter(s) set on head 2 (negative edge)

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:

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.

Important:

To save the incoming telegrams in multiframe mode, the project has four additional data blocks per channel. These data blocks are composed of UDTs 700 – 702.

Below are details of the data blocks and their function:

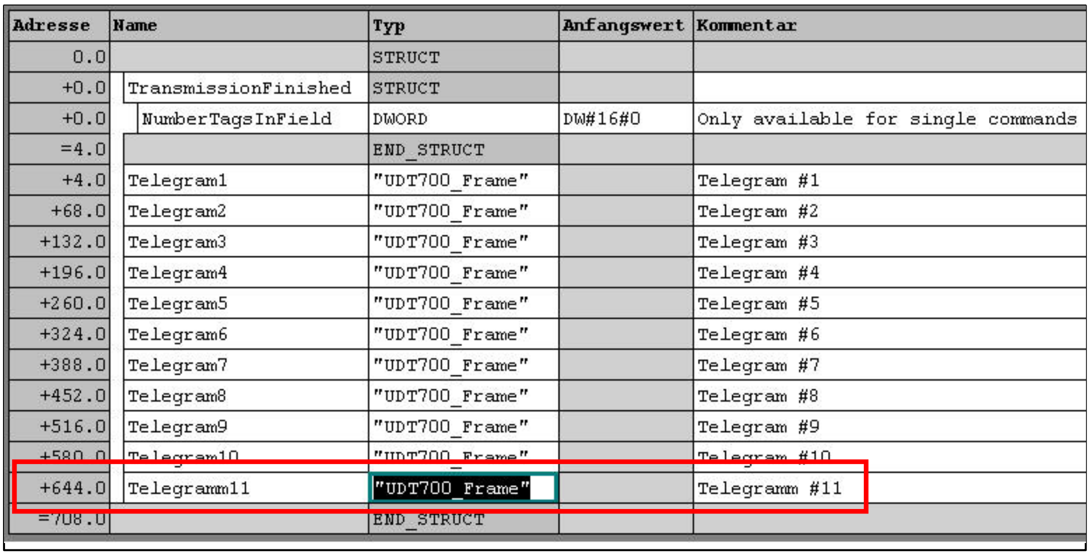
- DB700: Saves read Words (user data), Fixcode (TID), and Special Fixcode (EPC). Displays the number of read tags after a single command
- DB701: If the IF parameter is activated, the additional information for each telegram is displayed here
- DB703: During the Enhanced SpecialFixcode command, displays tags with associated EPC that have left the detection range

If data is present in a data block, the "HeadXNewData" bit is set.

Each data block acts as a buffer and saves up to ten telegrams in the intermediate memory. This buffer can be expanded or reduced as required using copy and paste. It is recommended that all four data blocks be expanded/reduced to ensure that the data remains consistent.

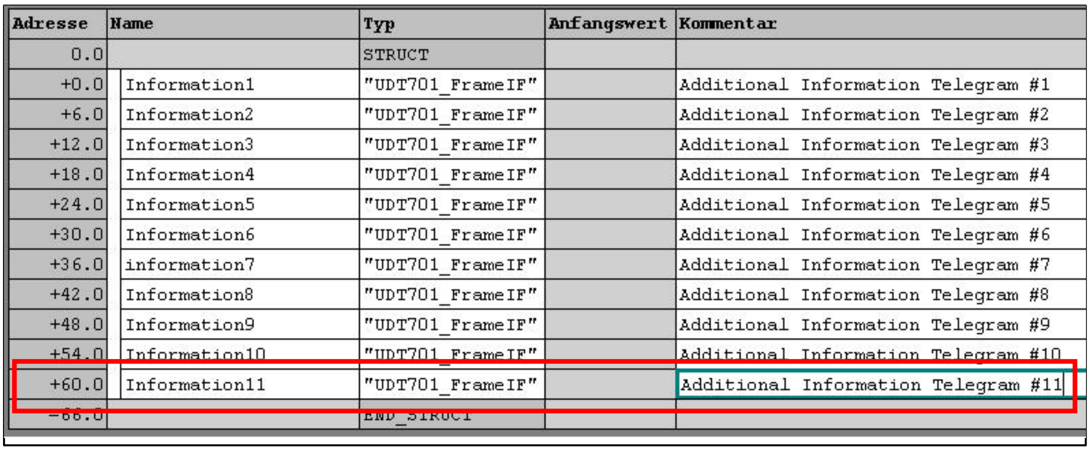
Expansion:

DB700:



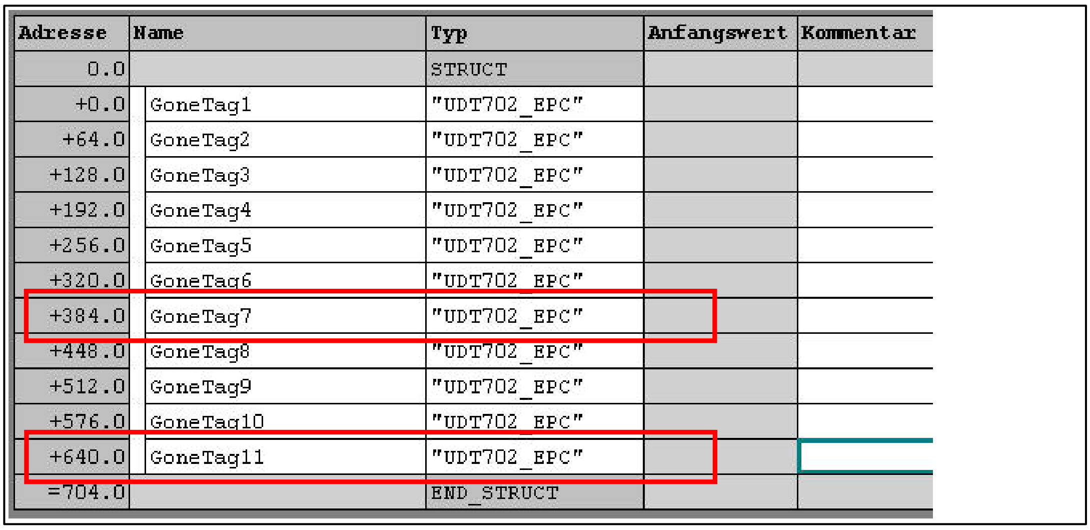
Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	TransmissionFinished	STRUCT		
+0.0	NumberTagsInField	DWORD	DW#16#0	Only available for single commands
=4.0		END_STRUCT		
+4.0	Telegram1	"UDT700_Frame"		Telegram #1
+68.0	Telegram2	"UDT700_Frame"		Telegram #2
+132.0	Telegram3	"UDT700_Frame"		Telegram #3
+196.0	Telegram4	"UDT700_Frame"		Telegram #4
+260.0	Telegram5	"UDT700_Frame"		Telegram #5
+324.0	Telegram6	"UDT700_Frame"		Telegram #6
+388.0	Telegram7	"UDT700_Frame"		Telegram #7
+452.0	Telegram8	"UDT700_Frame"		Telegram #8
+516.0	Telegram9	"UDT700_Frame"		Telegram #9
+580.0	Telegram10	"UDT700_Frame"		Telegram #10
+644.0	Telegramm11	"UDT700_Frame"		Telegramm #11
=708.0		END_STRUCT		

DB701:



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Information1	"UDT701_FrameIF"		Additional Information Telegram #1
+6.0	Information2	"UDT701_FrameIF"		Additional Information Telegram #2
+12.0	Information3	"UDT701_FrameIF"		Additional Information Telegram #3
+18.0	Information4	"UDT701_FrameIF"		Additional Information Telegram #4
+24.0	Information5	"UDT701_FrameIF"		Additional Information Telegram #5
+30.0	Information6	"UDT701_FrameIF"		Additional Information Telegram #6
+36.0	information7	"UDT701_FrameIF"		Additional Information Telegram #7
+42.0	Information8	"UDT701_FrameIF"		Additional Information Telegram #8
+48.0	Information9	"UDT701_FrameIF"		Additional Information Telegram #9
+54.0	Information10	"UDT701_FrameIF"		Additional Information Telegram #10
+60.0	Information11	"UDT701_FrameIF"		Additional Information Telegram #11
=66.0		END_STRUCT		

DB703:



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	GoneTag1	"UDT702_EPC"		
+64.0	GoneTag2	"UDT702_EPC"		
+128.0	GoneTag3	"UDT702_EPC"		
+192.0	GoneTag4	"UDT702_EPC"		
+256.0	GoneTag5	"UDT702_EPC"		
+320.0	GoneTag6	"UDT702_EPC"		
+384.0	GoneTag7	"UDT702_EPC"		
+448.0	GoneTag8	"UDT702_EPC"		
+512.0	GoneTag9	"UDT702_EPC"		
+576.0	GoneTag10	"UDT702_EPC"		
+640.0	GoneTag11	"UDT702_EPC"		
=704.0		END_STRUCT		

To ensure the output "HeadXCacheFull" switches correctly, the number of elements for saving the telegrams must be transferred to the block.

In this case:

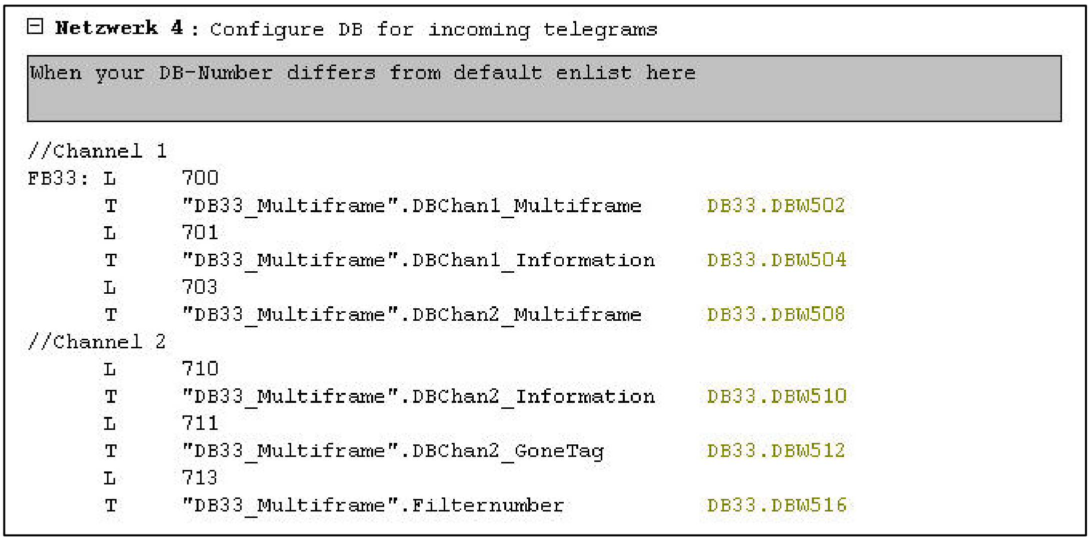
HeadXCache := 11

"HeadXCacheFull" indicates that the data block has reached its maximum memory capacity and subsequent telegrams will not be saved. To clear the buffer and reset the CacheFull status bit, "HeadXNewData" must be acknowledged with "0". This deletes and resets the buffer. To avoid data loss, make sure that the required data is copied from the data block in good time, before resetting "HeadXNewData".

"SetRestart" also deletes all content from the data blocks.

For channel 2, DB710 to DB713 work in the same way.

If a different numbering system is required for the data blocks, this system must be identified in OB1 in network 4.



```
⊟ Netzwerk 4 : Configure DB for incoming telegrams
When your DB-Number differs from default enlist here

//Channel 1
FB33: L      700
      T      "DB33_Multiframe".DBChan1_Multiframe      DB33.DBW502
      L      701
      T      "DB33_Multiframe".DBChan1_Information     DB33.DBW504
      L      703
      T      "DB33_Multiframe".DBChan2_Multiframe      DB33.DBW508
//Channel 2
      L      710
      T      "DB33_Multiframe".DBChan2_Information     DB33.DBW510
      L      711
      T      "DB33_Multiframe".DBChan2_GoneTag         DB33.DBW512
      L      713
      T      "DB33_Multiframe".Filternumber            DB33.DBW516
```

Example for channel 1:

OB1—network 4:

L 1200

T "DB32_Multiframe".DBChan1_Multiframe

L 1201

T "DB32_Multiframe".DBChan1_Information

L 1203

T "DB32_Multiframe".DBChan1_GoneTag

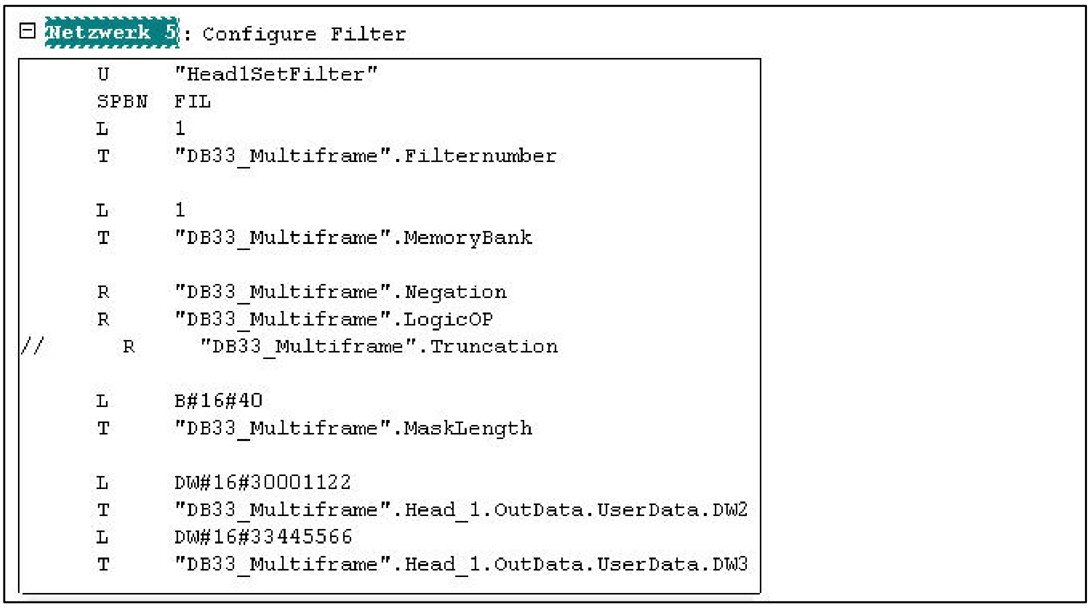
- "DB32_Multiframe".DBChan1_Multiframe saves Words (usable data), Fixcode, or Special Fixcode
- "DB32_Multiframe".DBChan1_Information saves additional information if the IF parameter is activated
- In enhanced SpecialFixcode mode, "DB32_Multiframe".DBChan1_GoneTag displays the EPC of the tags that have left the detection range during reading

7.1 Filter Configuration

Function block FB32 enables filter masks to be created and set so that specific read/write tags in the detection range can be accessed. To configure the filter, the required settings must be set in the OB1 (network 5: configure filter). Up to three filters can be created (filter numbers: 1 – 3).

Value ranges:

Filter number	1, 2, 3
Memory Bank	1 (EPC/UII), 2 (TID), 3 (User Memory)
Negation	0 (not negated), 1 (negated)
LogicOP	0 (OR link), 1 (AND link)
Truncation	Value is always 0
MaskLength	Mask length, 40_h = 64_{dec} (64 bit or 8 bytes)
MaskData (UserData.DW2)	DW#16#30001122 (example)
MaskData (UserData.DW3)	DW#16#33445566 (example)



```
Netzwerk 5: Configure Filter
      U     "Head1SetFilter"
      SPBN  FIL
      L     1
      T     "DB33_Multiframe".Filternumber

      L     1
      T     "DB33_Multiframe".MemoryBank

      R     "DB33_Multiframe".Negation
      R     "DB33_Multiframe".LogicOP
//      R     "DB33_Multiframe".Truncation

      L     B#16#40
      T     "DB33_Multiframe".MaskLength

      L     DW#16#30001122
      T     "DB33_Multiframe".Head_1.OutData.UserData.DW2
      L     DW#16#33445566
      T     "DB33_Multiframe".Head_1.OutData.UserData.DW3
```

The figure shows a filter configuration for channel 1. Configuration of the filter number with the value 1; Allocation memory bank 1 for filtering the EPC code. There is no negation on the mask and an OR link of the filters (only effective when multiple filters are active). The mask length is 64 bits. The filter mask consists of the following EPC, which is 8 bytes in length: 30.00.11.22.33.44.55.66

Example: Filter parameterization (filter no. 2)

The filter is applied to read/write tags, the data from which in the "User Memory" does not match the 5-byte-long mask (AA.BB.CC.DD.EE).

Caution!

Make sure the mask data notation is correct.

Define filter number:
"DB32_Multiframe".Filternumber := 2
Define memory segment:
"DB32_Multiframe".MemoryBank := 3 (User Memory)

Further configuration:
Negation := 1
LogicOP := 0
MaskLength := B#16#28 (5 bytes)
MaskData := DW#16#AABBCCDD
MaskData :=DW#16#EE000000 (note the notation!)

Set filter:
Head1SetFilter := 1
Head1Write := 1

Activate filter:
Head1FilterON := 1

Deactivate filter:
Head1FilterON := 0

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.

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

9 Examples Command Execution

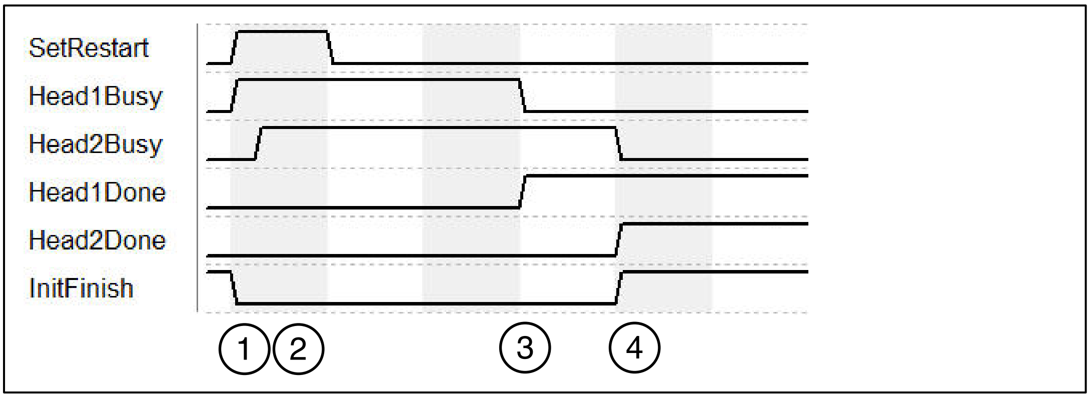
9.1 Initialiszation: (with tag type IUC72)

Define tag type:
Head1TagType := W#16#3038
Head2TagType := W#16#3038

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)

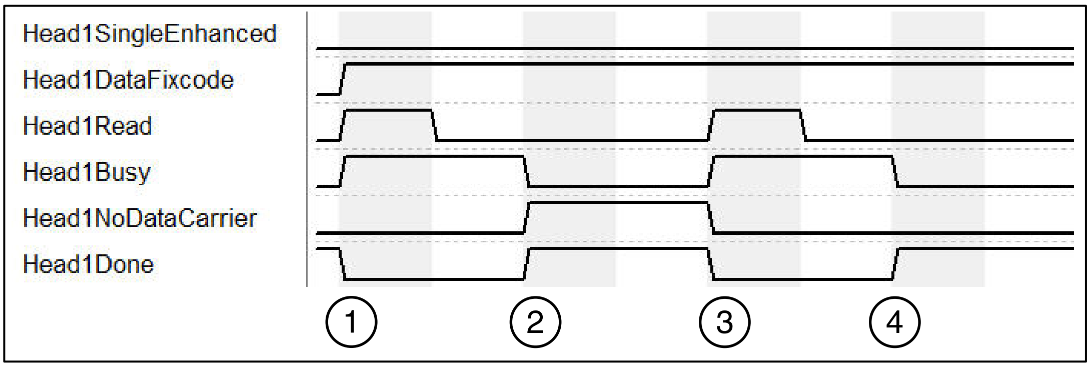
Define command parameters:
Head1SingleEnhanced := 0
Head1DataFixcode := 1

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)
Head1NewData = 1
Usable data:
DB700

Command complete -> No read/write tag present:
Head1Busy := 0
Head1NoDataCarrier := 1 (positive edge)
Head1Done := 1 (positive edge)
Head1NewData = 0
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
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Head1NewData = 1
Usable data:
DB700

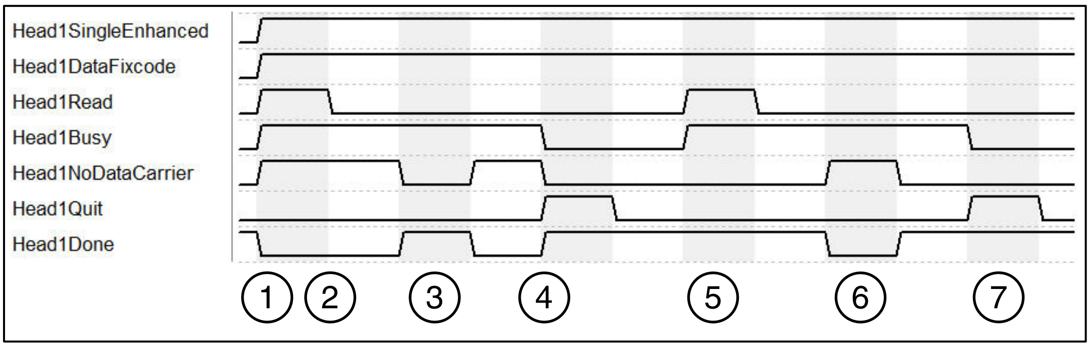
No read/write tag present and command active:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Head1NewData = 0
Usable data: none

Read/write tag moving out of the read range and no read/write tag present:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Head1NewData = 1
Usable data: DB703

Read/write tag moving into the read range and data read in:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Head1NewData = 1
Usable data:
DB700

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)

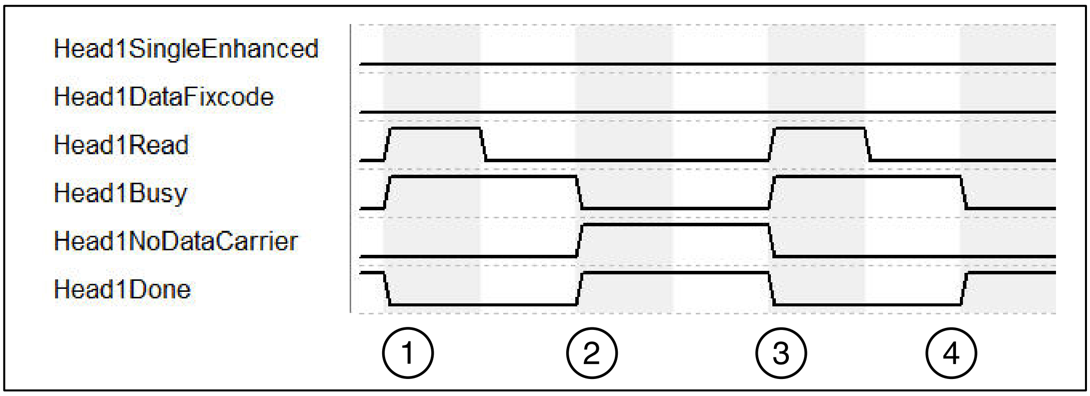
Define command parameters:
Head1WordNum := 15 (usable data count)
Head1WordAddress := 0 (memory address)
Head1SingleEnhanced := 0
Head1DataFixcode := 0

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

Command is processing:
Head1Busy := 1

Data read in:	No read/write tag present:

Head1Busy := 0
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Head1NewData :=1
Usable data:
DB700
	Head1Busy := 0
	Head1NoDataCarrier := 1 (positive edge)
	Head1Done := 1 (positive edge)
	Head1NewData := 0
	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)
Head1NewData =: 1
Usable data:
DB700

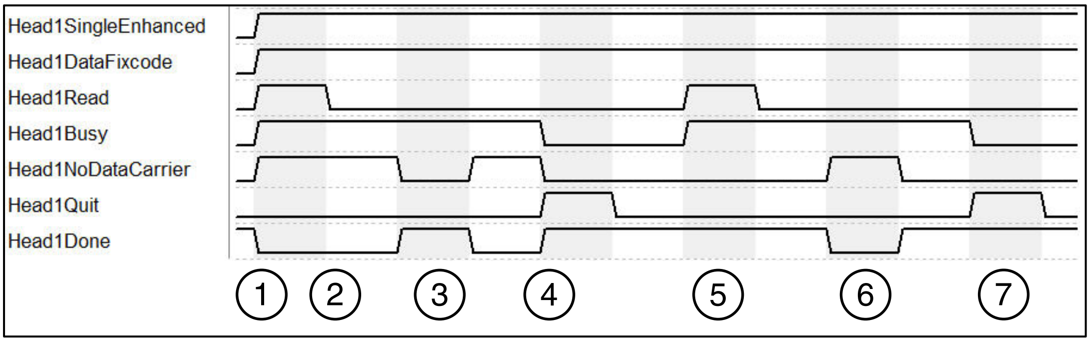
No read/write tag present and command active:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Head1NewData =: 0
Usable data: none

Read/write tag moving out of the read range and no read/write tag present:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Head1NewData =: 1
Usable data: DB703

Read/write tag moving into the read range and data read in:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Head1NewData =: 1
Usable data:
DB700

End enhanced command:
Head1Quit := 1

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



1. 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)

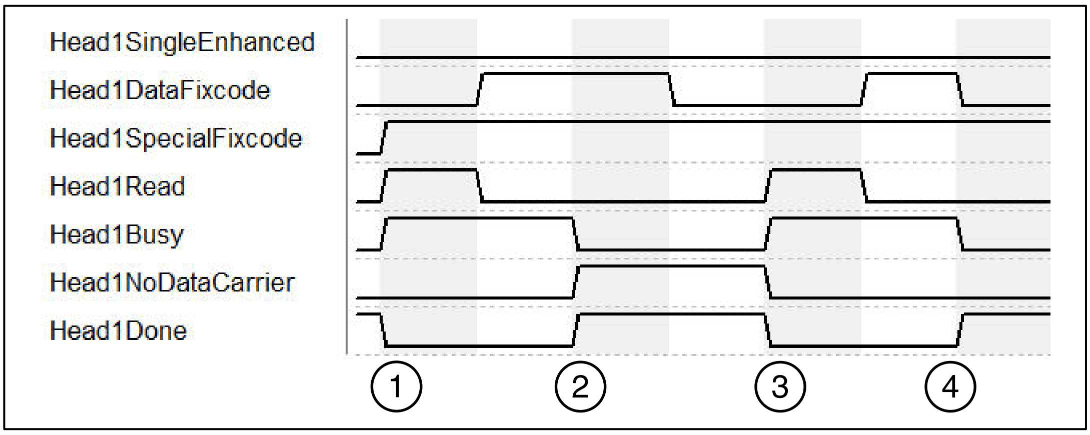
Define command parameters:
Head1SingleEnhanced := 0
Head1SpecialFixcode := 1

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)
Head1NewData =: 1
Usable data:
DB700

Command complete -> No read/write tag present:
Head1Busy := 0
Head1NoDataCarrier := 1 (positive edge)
Head1Done := 1 (positive edge)
Head1NewData =: 0
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)

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

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)
Head1NewData =: 1
Usable data:
DB700

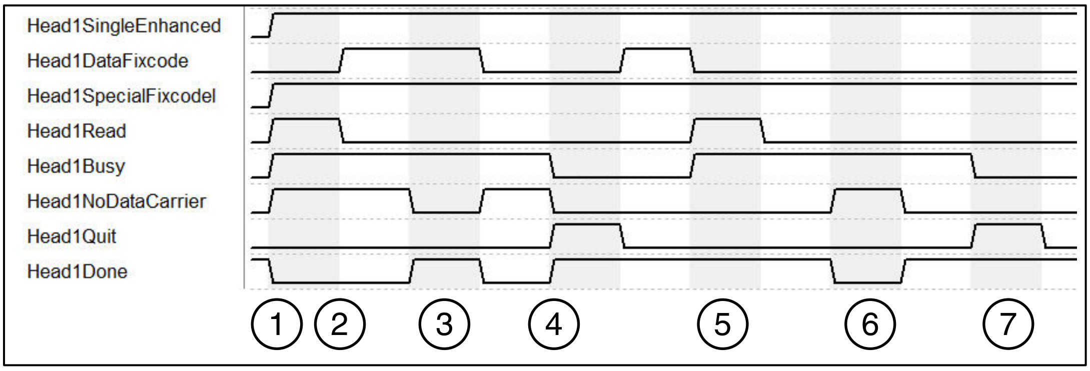
No read/write tag present and command active:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Head1NewData =: 0
Usable data: none

Read/write tag moving out of the read range and no read/write tag present:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0 (positive edge)
Head1NewData =: 1
Usable data: DB703

Read/write tag moving into the read range and data read in:
Head1Busy := 1
Head1NoDataCarrier := 0
Head1Done := 1 (positive edge)
Head1NewData =: 1
Usable data:
DB700

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 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

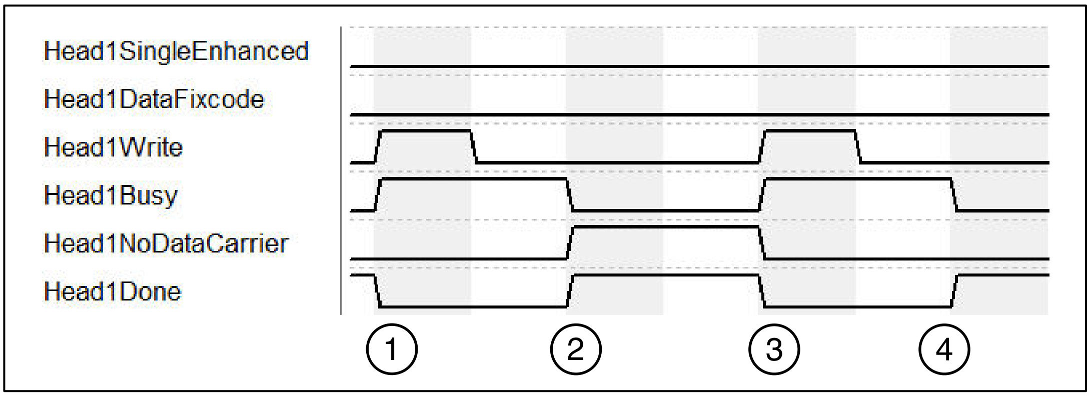
Define usable data in the output data field:
Head_1.OutData.UserData := XXXX

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. "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

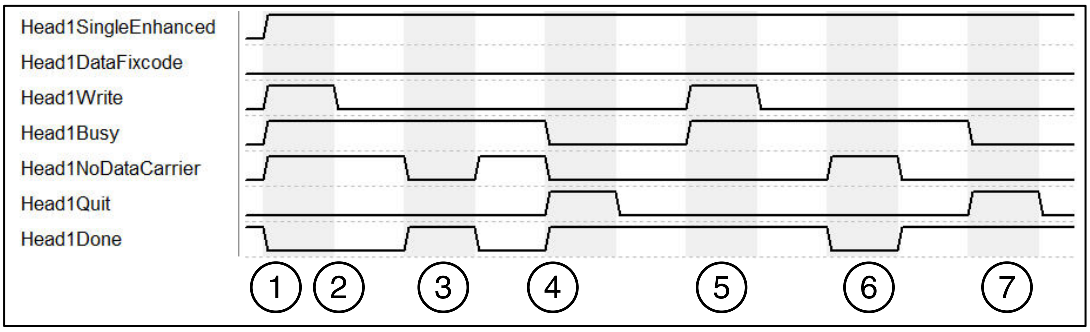
No read/write tag present and command active:
Head1Busy := 1
Head1NoDataCarrier := 1
Head1Done := 0

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“

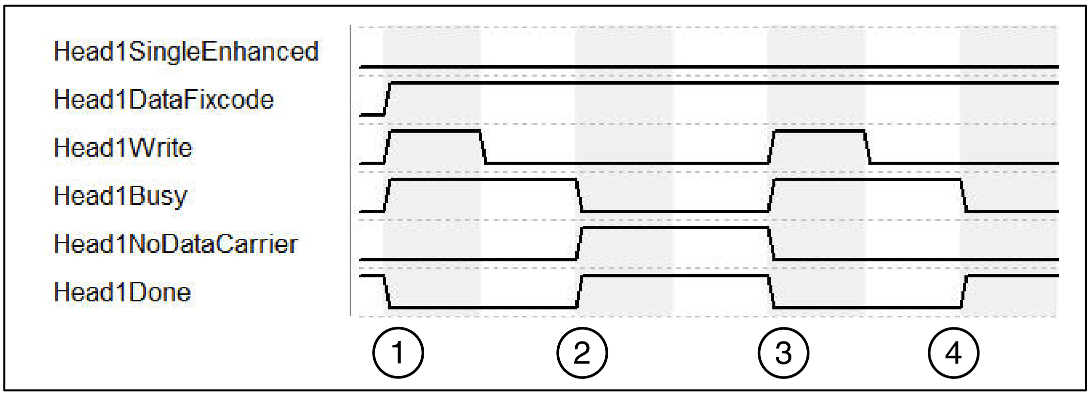
9.10 Single Write Fixcode: (Head 1; IPC11)

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

2014-02

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: (Head 1)

Define command parameters:
Head1SpecialFixcode := 1
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

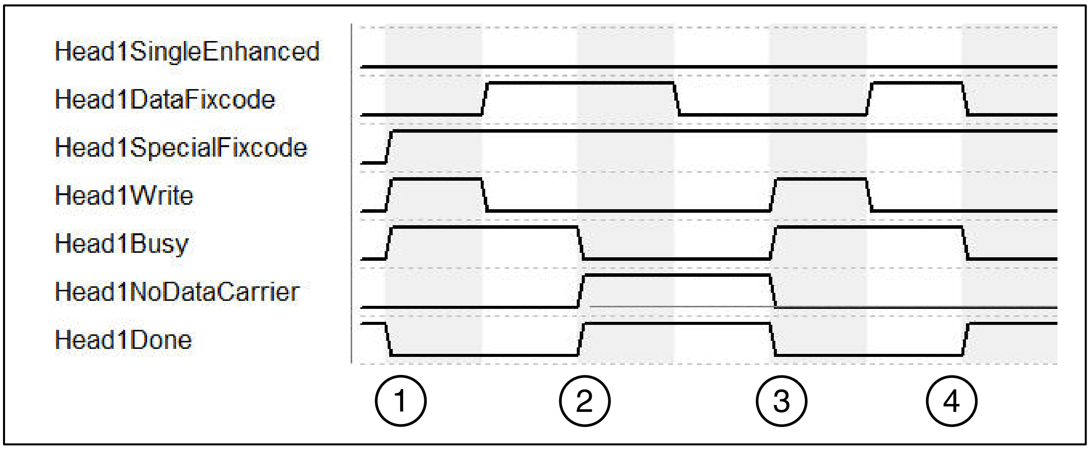
Start command execution:
Head1Write := 1 (positive edge)

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!

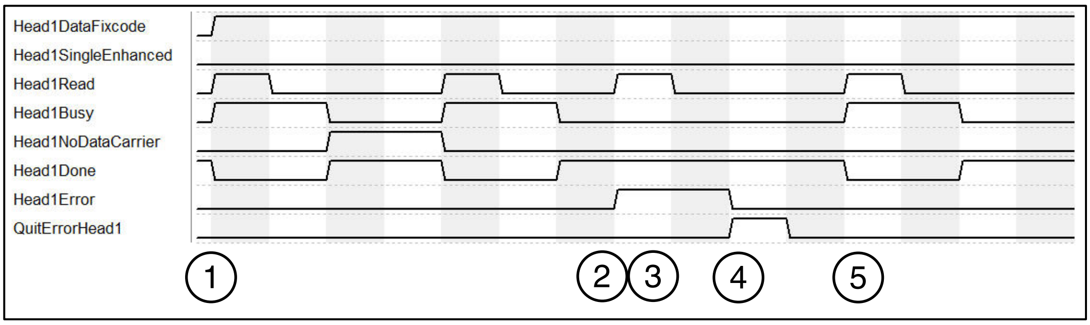


1. "Head1SingleEnhanced" is zero-active in order to initialize a single command. "Head1DataFixcode" is irrelevant as "Head1SpecialFixcode" has higher priority than

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. 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.

Öpen 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.

9.14 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.

10 Table of Read/Write Tags

Name	TagType	Access	Fixcode	Data	WordAddress	Frequency
IPC02	W#16#3032	Read Fixcode	5 Byte	-	-	125 kHz
IPC03	W#16#3033	Read Fixcode Read Words Write Words	4 Byte	116 Byte	0000…001C	125 kHz
IPC11	W#16#3131	Read Fixcode Write Fixcode	5 Byte	-	-	125 kHz
IPC12	W#16#3132	Read Fixcode Read Words Write Words	4 Byte	8192 Byte	0000…07FF	125 kHz
IDC	W#16#3530	Read Words Write Words Read Fixcode Write Fixcode Read Special Fixcode Write Special Fixcode	7 Byte Fixcode 6 Byte Special Fixcode	128 Byte	0000…001F	250 kHz
ICC	W#16#3532	Read Fixcode	7 Byte	-	-	250 kHz
IUC72	W#16#3830	Read Fixcode Write Fixcode Read Word Write Words Read Special Fixcode Write Special Fixcode	8 Byte	64 Byte		865 … 868 MHz
IUC73	W#16#3830	Read Fixcode Read Special Fixcode	8 Byte	-	-	868 MHz
IUC74	W#16#3830	Read Fixcode Write Fixcode Read Words Write Words Read Special Fixcode Write Special Fixcode	8 Byte	28 Byte		865 ... 871 MHz
IUC75	W#16#3830		8 Byte	-	-	865 ... 928 MHz
IQC21	W#16#3231	Read Fixcode Read Words Write Words	8 Byte	112 Byte	0000…001B	13,56 MHz
IQC22	W#16#3232	Read Fixcode Read Words Write Words	8 Byte	256 Byte	0000…003F	13,56 MHz
IQC23	W#16#3233	Read Fixcode Read Words Write Word	8 Byte	224 Byte	0000…0037	13,56 MHz
IQC24	W#16#3234	Read Fixcode Read Words Write Words	8 Byte	928 Byte	0000…00E7	13,56 MHz
IQC31	W#16#3331	Read Fixcode Read Words Write Words	8 Byte	32 Byte	0000…0007	13,56 MHz
IQC33	W#16#3333	Read Fixcode Read Words Write Words	8 Byte	2000 Byte	0000…00F9	13,56 MHz

IQC35	W#16#3335	Read Fixcode Read Words Write Words	8 Byte	256 Byte	0000…003F	13,56 MHz
MVC	W#16#3630	Read Fixcode Read Words Write Words	8 Byte	7552 Byte	0000…075F	2,45 GHz

11 Table of Status Values

HeadX Status	Description	Action
0x00	Command has been executed without error	None; a new command has been sent to this channel
0xFF	Processing command	Command is being processed on this channel; a command can be sent to another channel (not the same channel)
0x01	Low battery (only applies to MVC read/write tags)	Usable data is included in the same telegram; it is recommended that the battery or the whole read/write tag be replaced
0x02	Switch-on message; reset command was executed	Control interface is ready for command execution
0x04	Incorrect or incomplete command; command parameter not in the valid range	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?)
0x05	There is no read/write tag in the detection range.	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?)
0x06	Hardware error; no read/write head connected to this channel; read/write head is faulty	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)
0x07	Internal device error	Internal memory overflow (reduce the data hold time)
0x09	Parameterized tag type is not compatible with the connected reader	Check the defined tag type
0x0A	There are multiple tags with the same EPC in the detection range	Make sure that each tag has a unique EPC
0x0B	Additional telegram with IF parameter switched on	None; telegram with additional information was sent
0x0F	Last telegram in multiframe mode	None; after a single command, number of tags read in the detection range in ASCII
0x20	Switch-on message; reset command was executed	Error was sent by the bus interface; control interface is ready for command execution
0x40	Incorrect or incomplete command; command parameter not in the valid range	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?)
0x60	Hardware error; no read/write head connected to this channel; read/write head is faulty	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)
0x70	Internal device error	Internal memory overflow (reduce the data hold time)

12 Table Version History

Version	Date	Change to Function Block	Change to Documentation
2.0	11/18/ 2008	Adjustment from 4-channel version to 2-channel version Replacing the IN variable "IDENTControlAddress" with the IN variables INPUT/OUTPUT_Address and INPUT/OUTPUT_Length Output data length of 4 bytes possible through implementation of SFC81 Introduction of OUT variables Done; NoDataCarrier; Busy; and Error	First version
2.1	02/28/ 2009	Variable Memory.Error_SFC_14 is connected to the variables Head1/2Error Change to check of variables Head_1/2.ExistTC and Head_1/2.NotExist when parameterizing the commands -> trigger mode Introduction of the status check on the value 0x0F -> last telegram of the command list Change to the consistency check of the input and output data fields -> When executing the command list, the input and output data field are not equal Resetting the variable Head_1/2.QuitError in the Restart and QuitError routine Introduction of new symbol names for the usable data fields of the input and output data fields -> Head_1/2.InData/OutData.UserData Introduction of a value check of the parameter Head1/2WordNum -> Value > 15 is Head1/2Error set Output parameters Head1/2Status and Head1/2ReplyCounter introduced -> battery status and command list	New image introduced for the overview of the function block variables (page 4) New variables introduced in the function description table (page 4 – 5) Information on the maximum telegram length for different CPU series (page 6) Change to the variable name of the status value (page 6 – 7)
2.2	04/24/ 2009	Text library inserted for allocation of status messages	None
2.8	08/02/ 2013	Implementation of multiframe mode Parameterization of IUH readers now possible SpecialFixcode (EPC) readable/writeable Data blocks for processing the different telegrams on channel 1/2 inserted (DB70X & DB71X) SetRestart deletes content of the data blocks In/Out HeadXNewData displays new data in data blocks HeadXCacheFull indicates data block overflow	Parameterization options with FB190 explained (page 6) New multiframe block FB32 explained (page 10) Update to various status values
2.9	09/04/ 2013	Filter functions added to FB32 FB190 displays incorrectly configured parameters	Filter functions explained (page 17) Fault analysis for parameter commands (page 8)
3.0	10/25/ 2013	A DB for saving all data telegrams (user/Fixcode/Special Fixcode) Behavior of HeadXDone for single and enhanced commands adapted Write commands no longer cause HeadXCacheFull to be displayed Handling of status 0x0A adapted to multiframe FB190: Internal processes optimized, parameter error and parameter type error are now displayed correctly, parameters E5 and AP added	Status diagrams for all commands added Identification of inputs and outputs on the block Manual adapted to changelog

FABRIKAUTOMATION – SENSING YOUR NEEDS

Zentrale weltweit
Pepperl+Fuchs GmbH
68307 Mannheim · Deutschland
Tel. +49 621 776-0
E-Mail: info@de.pepperl-fuchs.com

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

Zentrale Asien
Pepperl+Fuchs Pte Ltd.
Singapur 139942
Tel. +65 67799091
E-Mail: sales@sg.pepperl-fuchs.com

www.pepperl-fuchs.com

PEPPERL+FUCHS
SENSING YOUR NEEDS

TDOCT3598__ENG
02/2014