User manual

IQT1-…-IO-V1 Easy Mode Function block Beckhoff TwinCAT 3 Controller with EL6224 IO-Link Master

IO-Link RFID-head IQT1-…-IO-V1

Project Name:	**IO-Link RFID-head ISO15693 13,56MHz**
Date:	**25.10.2018**
Creator:	**Karsten Reinhardt**

Version overview

Version	Release Datum	Comments
A	**25.10.2018**	Initial Version

Table of contents

1. Input and Output data fields for process data

The RFID head IQT1-...-IO-V1 operates according to IO-Link Standard V1.1. This RFID head therefore has a data width of 32 bytes for the process input and output data.

The input and output data fields for the process data are defined in the Global Variable List (GVL). The following figure shows the definition of the process data for inputs and outputs for 4 RFID heads IQT1-...-IO-V1 that can be connected to the IO-Link master.

GVL × IQT1-

```
{attribute 'qualified_only'}
VAR_GLOBAL
    (*Note:
     The assignment of the data fields to the ports is carried out via the port linkage
     *)

     ProcessDataIn   AT %I*: ARRAY[0..31] OF BYTE;              //Global Process input data field for Port 1
     ProcessDataOut  AT %Q*: ARRAY[0..31] OF BYTE;              //Global Process output data field for Port 1

     ProcessInputData_Port_2   AT %I*: ARRAY[0..31] OF BYTE;    //Global Process input data field for Port 2
     ProcessOutputData_Port_2  AT %Q*: ARRAY[0..31] OF BYTE;    //Global Process output data field for Port 2

     ProcessInputData_Port_3   AT %I*: ARRAY[0..31] OF BYTE;    //Global Process input data field for Port 3
     ProcessOutputData_Port_3  AT %Q*: ARRAY[0..31] OF BYTE;    //Global Process output data field for Port 3

     ProcessInputData_Port_4   AT %I*: ARRAY[0..31] OF BYTE;    //Global Process input data field for Port 4
     ProcessOutputData_Port_4  AT %Q*: ARRAY[0..31] OF BYTE;    //Global Process output data field for Port 4
//VAR_GLOBAL CONSTANT
END_VAR
```

These are data fields of the type "Array of Byte". The length is always 32 bytes per data field.

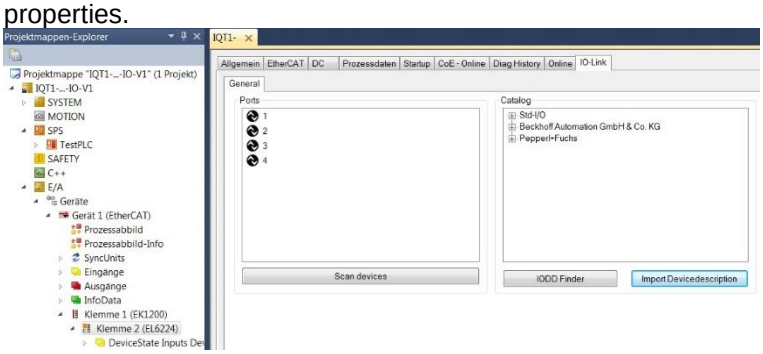
Input data field head 1/Port 1:	ProcessDataIn	AT %I*: ARRAY[0..31] OF BYTE;
Output Data field head 1/Port 1:	ProcessDataOut	AT %Q*: ARRAY[0..31] OF BYTE;
Input data field head 2/Port 2:	ProcessInputData_Port_2	AT %I*: ARRAY[0..31] OF BYTE;
Output data field head 2/Port 2:	ProcessOutputData_Port_2	AT %Q*: ARRAY[0..31] OF BYTE;
Input data field head 3/Port 3:	ProcessInputData_Port_3	AT %I*: ARRAY[0..31] OF BYTE;
Output data field head 3/Port 3:	ProcessOutputData_Port_3	AT %Q*: ARRAY[0..31] OF BYTE;
Input data field head 4/Port 4:	ProcessInputData_Port_4	AT %I*: ARRAY[0..31] OF BYTE;
Output data field head 4/Port 4:	ProcessOutputData_Port_4	AT %Q*: ARRAY[0..31] OF BYTE;

The input and output data fields are parameterized to the function block "IQT1_EasyMode" for the variables "ADR_IN", "ADR_OUT", "SIZE_IN" and "SIZE_OUT". The following figure shows this for head 1 at port 1.

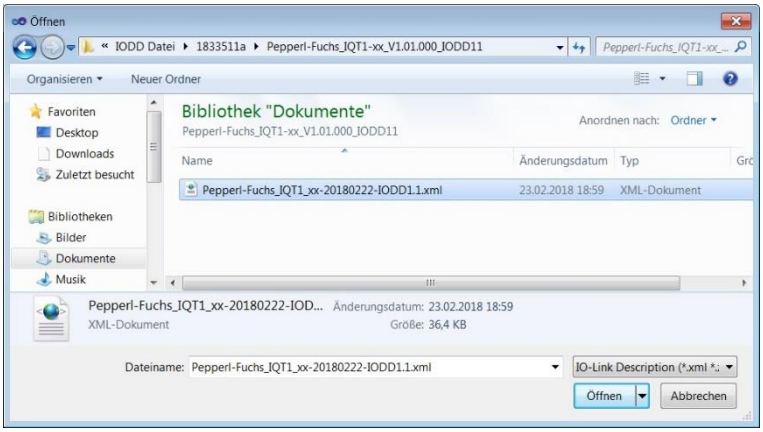
```
'Commissioning for IQT1-...-IO-V1 EasyMode connected to Port 1
   IQT1_EasyMode(
   ADR_IN                    := ADR(GVL.ProcessDataIn),      //Address of process input data
   i_I_SIZE_IN               := SIZEOF(GVL.ProcessDataIn),   //length of input data
   ADR_OUT                   := ADR(GVL.ProcessDataOut),     //Address pf process output data
   i_I_SIZE_OUT              := SIZEOF(GVL.ProcessDataOut),  //length of output data
```

2. Installation of the IODD file

When commissioning an IO-Link capable device for the first time, the IODD (IO Device Description) must first be installed. This file contains information on IO-Link device parameters and communication properties.

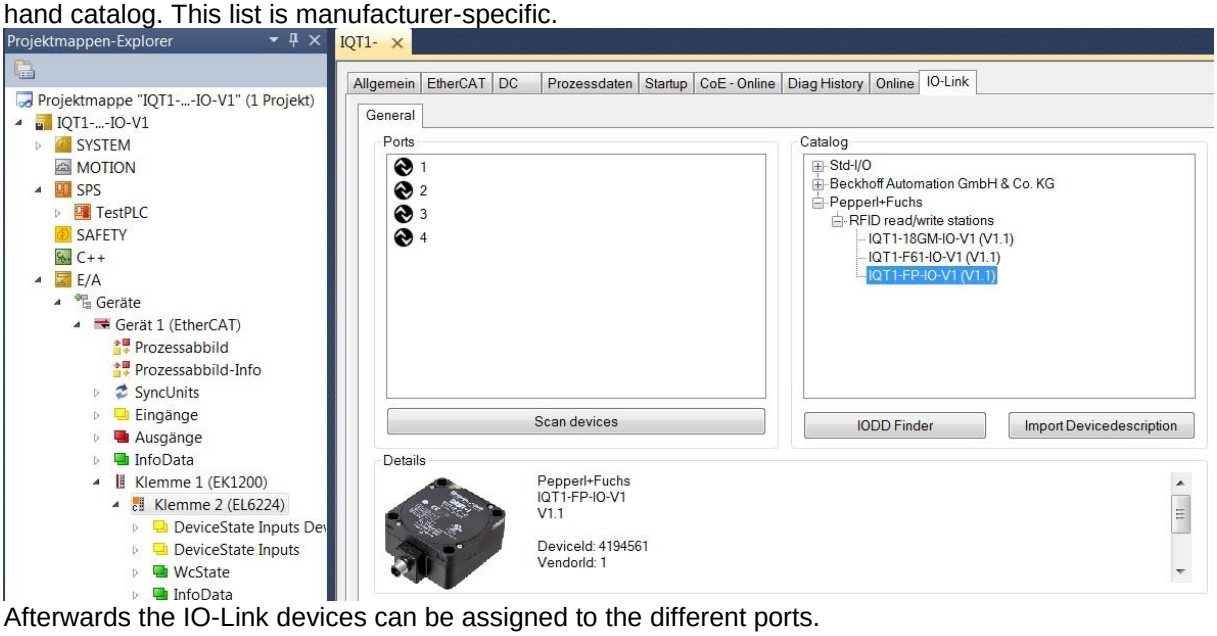


A double-click on "Terminal 2 (EL6224)" opens an overview window of the device properties of the IO-Link master EL6224. The parameterization of the IO-Link devices to be connected is carried out in the "IO-Link" submenu.



The selection "Import Device description" opens a window for selecting the IODD file. The IODD is a file in .xml format.

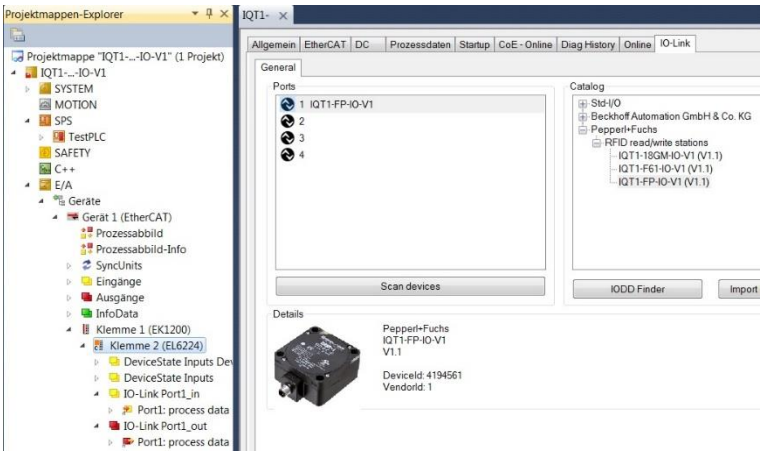
After successful installation of the IODD, an overview of the various RFID heads appears in the right-hand catalog. This list is manufacturer-specific.



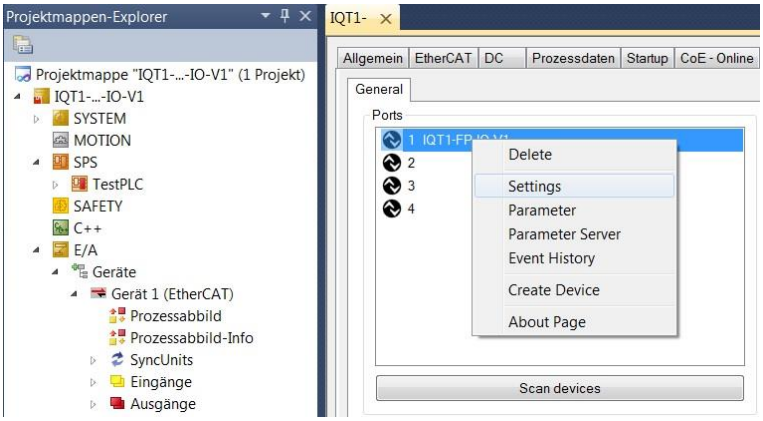
Afterwards the IO-Link devices can be assigned to the different ports.

3. Setting IO-Link Ports

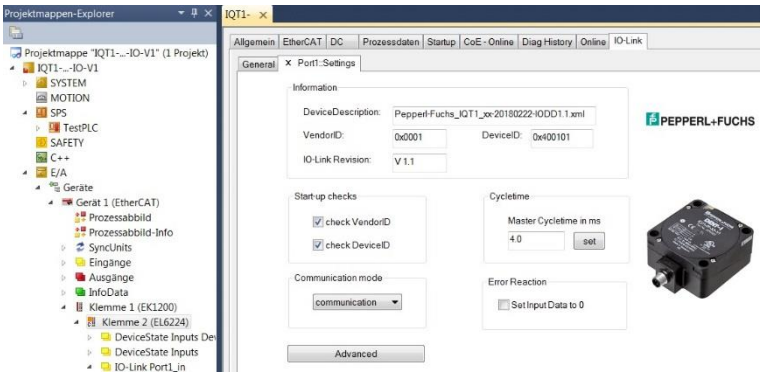
For the assignment of the IO-Link device to the IO-Link port of the master, the symbolic device designation from the catalog must be drawn into the placeholder for port 1.



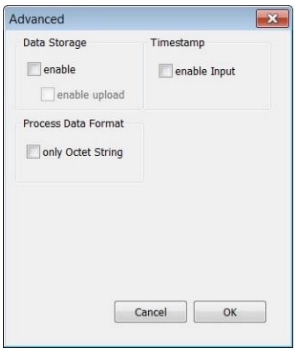
Assignment of IO-Link device "IQT1-FP-IO-V1" to port 1 of the IO-Link master EL6224.



A right click on the symbol designation "IQT1-FP-IO-V1" opens the "Settings" submenu for setting the IO-Link communication.



A click on "Advanced" activates the setting for saving the IO-Link parameters. The storage of the parameters is deactivated to allow a dynamic setting of the IO-Link parameters by the function block.



Deactivation of the "Data Storage" function. This allows the IO-Link parameters to be changed during the runtime of the PLC. A function block is available for this purpose.

4. Easy-Mode – General information

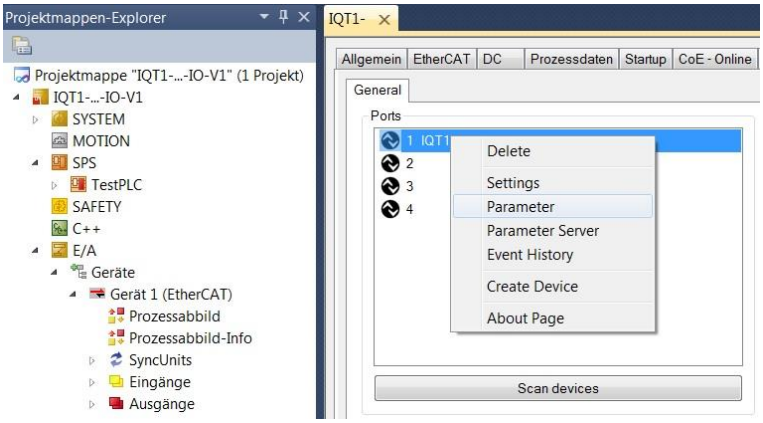
The IQT1-...-IO-V1 RFID head communicates via an "Easy Mode" when the device is delivered. With the "Easy-Mode", a limited amount of data can be read or written from a data carrier without a handshake procedure. The RFID head can be connected to the controller without using an additional function block. This simplifies the effort for device integration and data processing. For simple applications with limited performance requirements, the use of the "Easy-Mode" is therefore recommended.

Within the "Easy-Mode", an "Autostart" function is activated at the factory status. This function automatically starts read access to the user data area of the data carrier by the RFID head as soon as the head is supplied with power. In this case, no control values need to be sent to the device, only data is received in the controller.

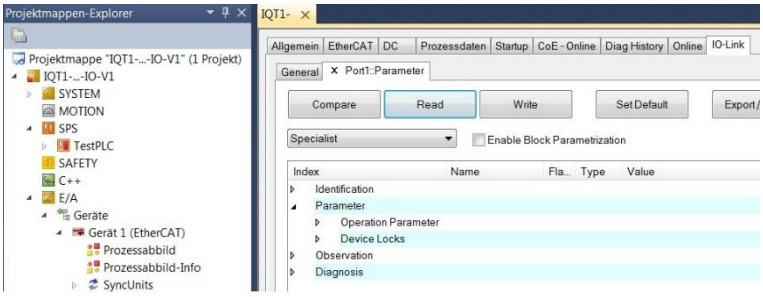
The setting of the IQT1-...-IO-V1 RFID head must be carried out via the configuration of IO-Link device parameters. In the delivery state of the head, the parameters are preset with their initial values, but they can be changed depending on the application.

5. Access IO-Link Parameter IQT1-…-IO-V1

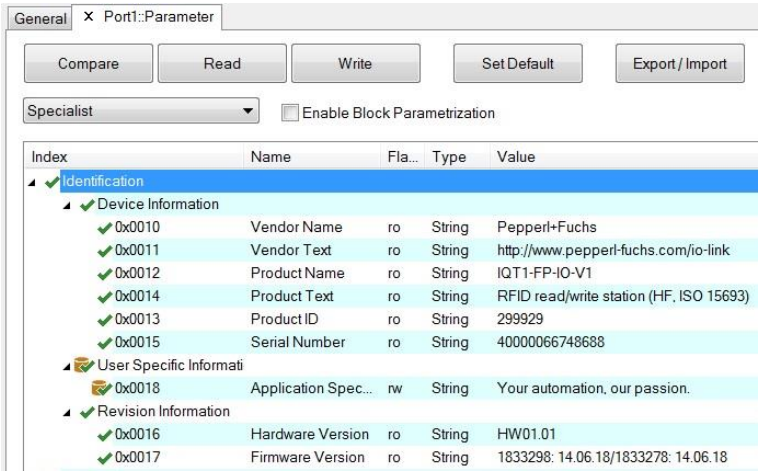
The IO-Link RFID head IQT1-...-IO-V1 has various IO-Link parameters. These parameters are divided into general and device-specific parameters. Each IO-Link capable device has the general parameters. The device-specific parameters only apply to the associated IO-Link device.



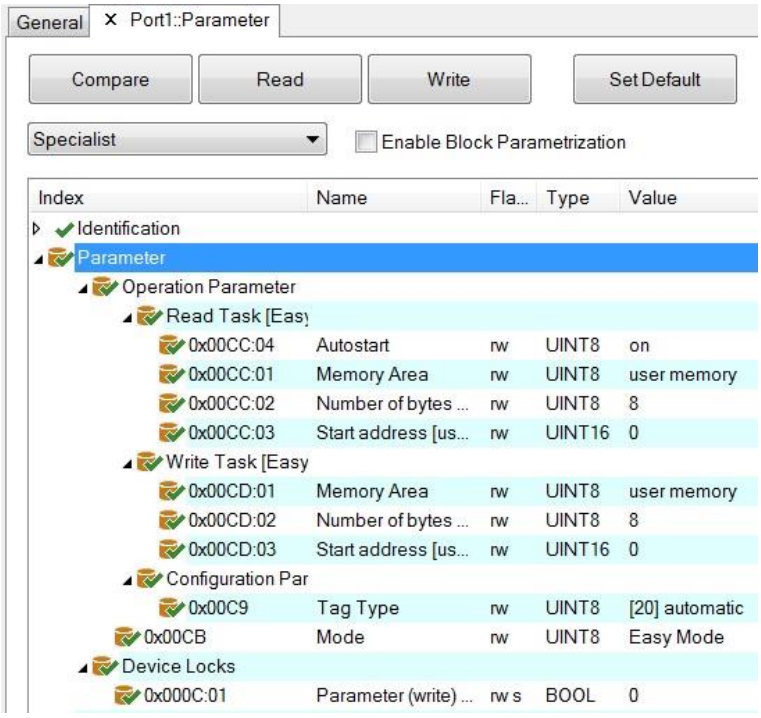
A right click on the symbol designation "IQT1-FP-IO-V1" opens the "Parameters" submenu for setting the IO-Link parameters.



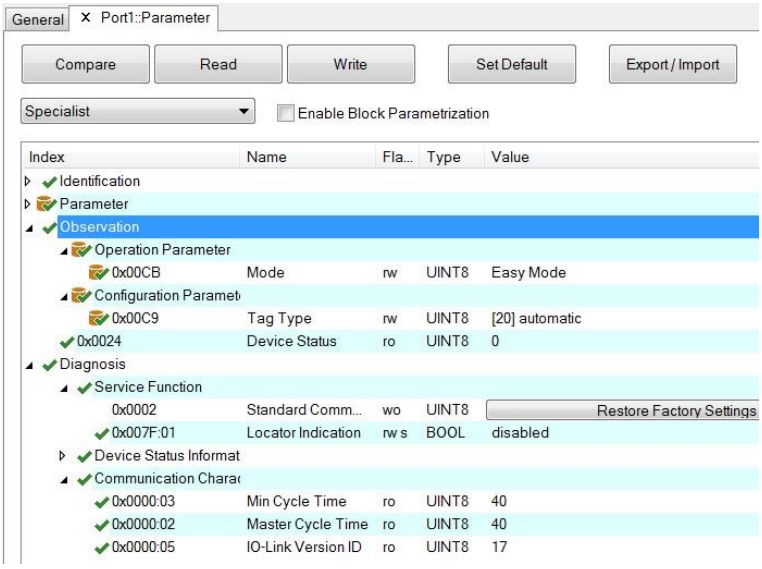
The "Read" field allows all IO-Link parameters to be read out together within one read operation.



General manufacturer information about the IO-Link device can be displayed within the "Identification" area. This is, for example, the product name (IQT1-FP-IO-V1) or the article number (#299929). In addition, information on the hardware and firmware version can be viewed.



The device-specific parameters are displayed in the "Parameters" area. Here, changes can be made to the behavior of the RFID head.



In the "Observation" and "Diagnosis" areas, for example, the device status can be read out. It is also possible to reset the device to factory settings.

6. Easy-Mode – Process data structure

The process data is transferred between the IQT1-...-IO-V1 RFID head and the controller via the process data fields. There is a process data field for input data, i.e. from the direction of the head to the controller, and a process data field for output data, i.e. from the direction of the controller to the RFID head. Both process data fields have a fixed length of 32 bytes. This length is constant and always amounts to 32 bytes. A deviating length parameterization is not possible.
The following table shows the structure of the process data field for the output data:

Byte	Content							
0	0	0	0	0	0	0	Start Write	Start Read
1	unused							
2	unused							
3	unused							
4	Write Data							
5	Write Data							
6	Write Data							
….	Write Data							
31	Write Data							

If the "Autostart" function is activated, no output data must be sent. The header performs a permanent read access to user data (factory setting, 8 byte length) or fixed code. When using the "Autostart" function, the bits "Start Read" and "Start Write" have no relevance.

The "Autostart" function can be switched off via parameter 204 "Read Task". If the function is switched off, a read task or a write task can be started via the "Start Read" or "Start Write" bit.

A read task or a write task is executed as long as the corresponding start bit is set. The task can only be aborted by resetting the corresponding start bit.

The parameters required for data carrier access such as "memory area", "number of bytes" and "start address" can be set for the read task via parameter 204 and for the write task via parameter 205.

The following table shows the structure of the process data field for the input data:

Byte	Content							
0	0	0	0	0	Error	Active	Write Valid	Read Valid
1	Length Data							
2	unused							
3	unused							
4	Read Data							
5	Read Data							
6	Read Data							
…	Read Data							
31	Read Data							

As soon as a read or write task is started and executed, this is indicated by the "Active" bit. This bit remains set for the complete period of the task execution. Only when the read or write task is aborted does the "Active" bit reset itself.
If a read task is active, the "Read Valid" bit is set if the data carrier is within the detection range and the data has been read. The bit remains set for the duration of the data carrier's stay in the detection zone. This bit is only reset again when the data carrier leaves the detection zone.

The bit "Write Valid" behaves identically. It is set if the data carrier is in the detection zone and the data was successfully written to the data carrier. The reset takes place as soon as the data carrier leaves the detection zone again.

The byte "Length Data" contains the length specification of the read data in bytes. The length depends on the number of bytes set via parameter 204. When accessing the fixcode, the length is 8 bytes and when accessing the user data, the length is a multiple of 4 bytes (or 8 bytes when using an IQC33 transponder).

An error may occur during the execution of a read or write task. The error state is indicated by the "Error" bit. If there is an error state, additional error information is transmitted via the process input data field. This information contains an error code as well as an error description in plain text (ASCII character). A check of the error description provides an indication of the cause of the error condition.

The following table shows the structure of the process data field of the input data in error state:

Byte	**Content**							
0	0	0	0	0	Error	Active	Write Valid	Read Valid
1	Length Data							
2	Unused							
3	Unused							
4	Error Code (HEX)							
5	Error String							
6	Error String							
…	Error String							
31	Error String							

7. Function block „IQT1_EasyMode“

The function block "IQT1_EasyMode" allows the execution of write and read tasks by controlling binary inputs of the block. The number of bytes to be accessed and the start address are determined separately by the IO-Link parameters. The binary outputs of the function block signal the execution status of a task. In addition, there are counters which indicate the number of read and write operations within a task. The read-in user data and the data to be written are provided in separate data fields.

The data fields of the function block are stored in the application-specific data type (DUT; Data Unit Type) "IQT1-EasyMode_Data". A distinction is made between the input and output data fields with a length of 32 bytes and the read and write data fields with a length of 28 bytes.

The user program can be tested with the support of a visualization. In the visualization "FB_EasyMode_IO_Port1" the function block for head 1 can be controlled at port 1.

7.1 Call function block „IQT1_EasyMode“

The function block is called in the "MAIN (PRG)". The following figure shows this for head 1, which is connected to port 1.

```
(*Commissioning for IQT1-...-IO-V1 EasyMode connected to Port 1*)
   IQT1_EasyMode(
   ADR_IN                  := ADR(GVL.ProcessDataIn),      //Address of process input data
   i_I_SIZE_IN             := SIZEOF(GVL.ProcessDataIn),   //length of input data
   ADR_OUT                 := ADR(GVL.ProcessDataOut),     //Address pf process output data
   i_I_SIZE_OUT            := SIZEOF(GVL.ProcessDataOut),  //length of output data
   b_I_StartRead           := StartRead,                   //True = Start Read Task; False = Stop Read Task
   b_I_StartWrite          := StartWrite,                  //True = Start Write Task; False = Stop Write Task
   Data                    := Data_RFID_Head,              //Structure for data fields
   b_O_ReadValid           => ReadValid,                   //True = Read successful; Tag in detection zone
   b_O_WriteValid          => WriteValid,                  //True = Write successful; Tag in detection zone
   b_O_Active              => Active,                      //True = Read or Write Tasl active
   b_O_Error               => Error,                       //True = Error occure; see input data field
   B_O_LengthData          => LengthData,                  //Length of the read data
   b_O_NoTag               => NoTag,                       //True = no tag inside detection zone
   w_O_ReadCounter         => ReadCounter,                 //number of read access during read task
   w_O_WriteCounter        => WriteCounter);               //number of write access during write task
```

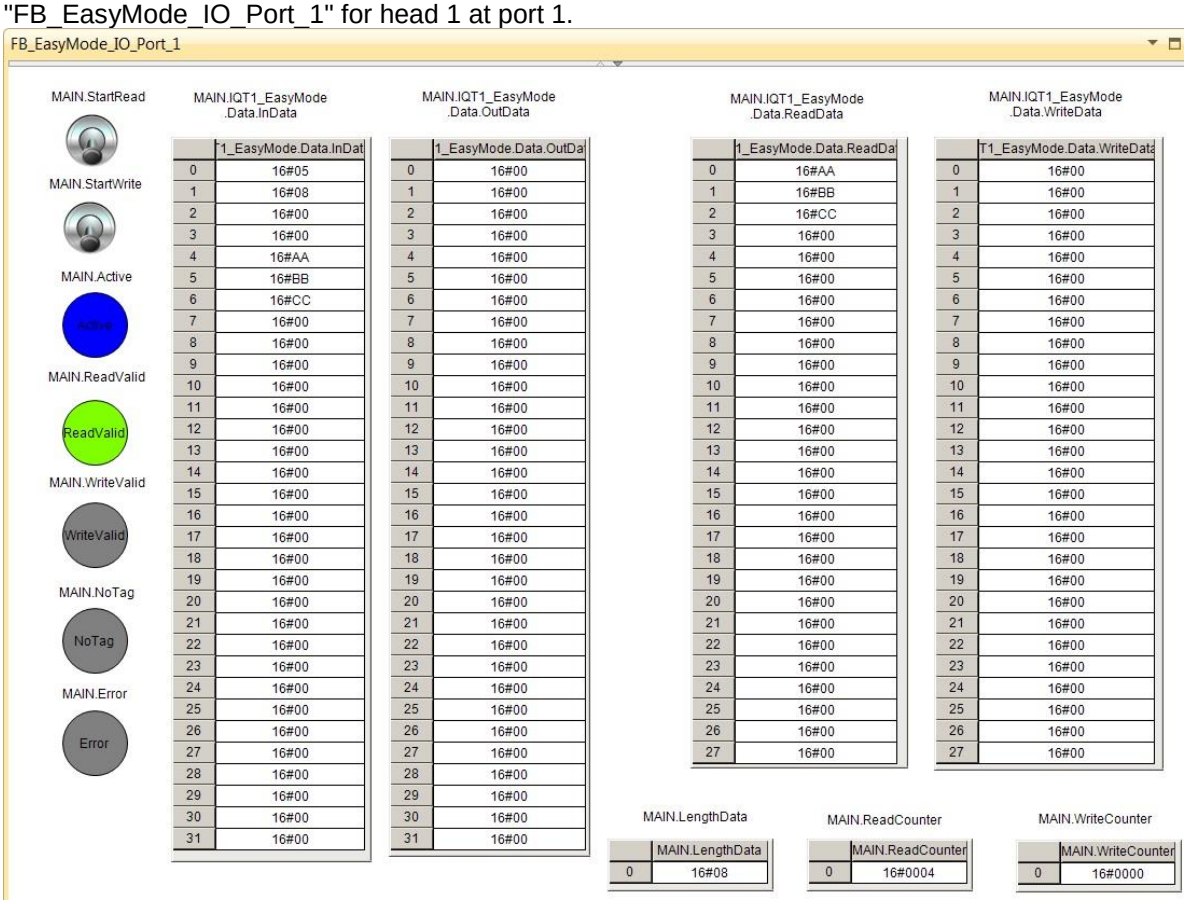
The inputs and outputs have the following meaning:

Variable	Type	meaning
ADR_IN	pvoid	Address Input data field from parameterization in the GVL; ADR (GVL.ProcessDataIn)
I_I_SIZE_IN	UINT	Length of the input data field; SIZEOF(ProcessDataIn)
ADR_OUT	pvoid	Address Output data field from parameterization in the GVL; ADR (GVL.ProcessDataOut)
I_I_SIZE_OUT	UINT	Length of the input data field; SIZEOF(ProcessDataOut)
b_I_StartRead	BOOL	Input for start read task: TRUE = Start reading task FALSE = End reading task; no reading task active If the Autostart function is activated, this input has no relevance. In Auto-start mode, the read task is started automatically.
b_I_StartWrite	BOOL	Input for start write task: TRUE = Start write task; define write data beforehand in the WriteData structure; FALSE = end of write task; no write task active If the Autostart function is activated, this input has no relevance. The Au-tostart function does not support a write task.
Data	IQT1_EasyMode_Data	Structure with the read and write data read in data = ReadData field Write data = WriteData field
b_O_ReadValid	BOOL	TRUE = Read access successful; read data within the ReadData struc-ture; data carrier within the detection zone; ReadCounter + 1; FALSE = no data carrier within the detection zone

b_O_WriteValid	BOOL	TRUE = write access successful; data carrier within detection zone; WriteCounter + 1; FALSE = no data carrier within the detection zone;
b_O_Active	BOOL	TRUE = Read or write task active; FALSE = no task active; autostart function off
b_O_Error	BOOL	TRUE = error state; error code within data structure ReadData; FALSE = no error
B_O_LengthData	Byte	Number of bytes read in; length of data read in
b_O_NoTag	BOOL	TRUE = no tag in the detection zone; FALE = tag in the detection zone if Active = TRUE;
w_O_ReadCounter	Word	Counter for the number of read accesses per read task; incremented per successful read access
w_O_WriteCounter	Word	Counter for the number of write accesses per write task; incremented per successful write access

7.2 Visualization of the “IQT1_EasyMode” function block

Within the sample project, there are ready-made visualizations. This allows the function block to be controlled and tested via a graphical user interface. The following figure shows the visualization "FB_EasyMode_IO_Port_1" for head 1 at port 1.



8. Function block „FB_IOLinkParameter“

The function block "FB_IOLinkParameter" allows dynamic access to the IO-Link parameters during the runtime of the control program. Thus, all IO-Link parameters can be read out and changed at any time.

A distinction is made between "general parameters" and "device-specific parameters".
The general parameters contain manufacturer information on the IO-Link device. This is, for example, the device name or the serial number. These parameters can be read out.
Device-specific parameters have an influence on the mode of operation of the IO-Link RFID head. These parameters can be read but also changed. For example, the number of bytes to be read out can be changed in this way.

8.1 Call of the function block „FB_IOLinkParameter“

The function block "FB_IOLinkParameter" is called in the "MAIN (PRG)". The following figure shows this for head 1, which is connected to port 1.

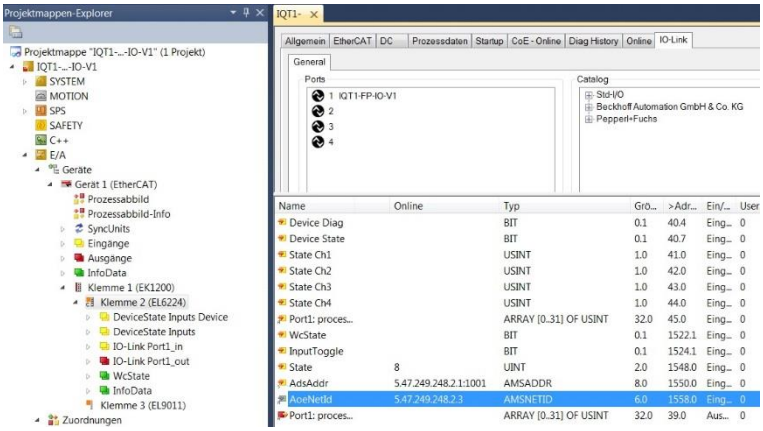
```
//Function block for configuration of IO-Link Parameter on IQT1-...-IO-V1
//Call of function block for Port 1
FB_IOLinkParameter(
    str_I_NETID                 :='5.47.249.248.2.3' ,          //AoE NetID of the EL6224
    w_I_PORT                    :=16#1000 ,                     //Number of the port on which the device is co
    dw_I_IDXGRP_IO_LINK         :=16#F302 ,                     //Index group; always 16#F302
    b_I_ReadConfiguration       := b_I_ReadConfiguration,       //Start Read configuration
    b_I_WriteConfiguration      := b_I_WriteConfiguration,      //Start Write configuration
    b_O_ReadConfiguration_finish    => b_O_ReadConfiguration_finish,    //Reading of configuration finished
    b_O_ReadConfiguration_busy      => b_O_ReadConfiguration_busy,      //Reading of configuration active
    b_O_WriteConfiguration_finish   => b_O_WriteConfiguration_finish,   //Writing of configuration finished
    b_O_WriteConfiguration_busy     => b_O_WriteConfiguration_busy);    //Writing of configuration active

//Reset Input for Start Reading IO-Link Parameters; positive edge triggering
IF b_I_ReadConfiguration
    THEN b_I_ReadConfiguration := FALSE;
END_IF
//Reset Input for Start Writing IO-Link Parameters; positive edge triggering
IF b_I_WriteConfiguration
    THEN b_I_WriteConfiguration := FALSE;
END_IF
```

The inputs and outputs have the following meaning:

Variable	Type	Meaning
str_NETID	String	AoENetId of the EL6224 IO-Link Master; visible in the "General" properties of the Master
w_I_Port	Word	Port number to which the RFID head is connected; the following numbering applies: Port 1 = 16#1000 Port 2 = 16#1001 Port 3 = 16#1002 Port 4 = 16#1003
dw_I_IDXGRP_IO_LINK	DWord	Index group of the IO-Link master EL6224; always 16#F302
b_I_ReadConfiguration	BOOL	Input for starting the read access to the IO-Link parameters: TRUE (positive edge) = Start read out of the IO-Link parameters; all parameters are read out; read in parameter values are located in the In-Data variables.
b_I_WriteConfiguration	BOOL	Input for starting the write access to the IO-Link parameters: TRUE (positive edge) = Start write the IO-Link parameters; only the device-specific parameters are written; the parameter values must be defined beforehand in the OutData variables.
b_O_ReadConfiguration_finish	BOOL	TRUE = End of readout process of the IO-Link parameters
B_O_ReadConfiguration_busy	BOOL	TRUE = Readout process of the IO-Link parameters active FALSE = Readout process completed
B_O_WriteConfiguration_finish	BOOL	TRUE = End of write operation of the IO-Link parameters
B_O_WriteConfiguration_busy	BOOL	TRUE = write operation of the IO-Link parameters active FALSE = write operation completed

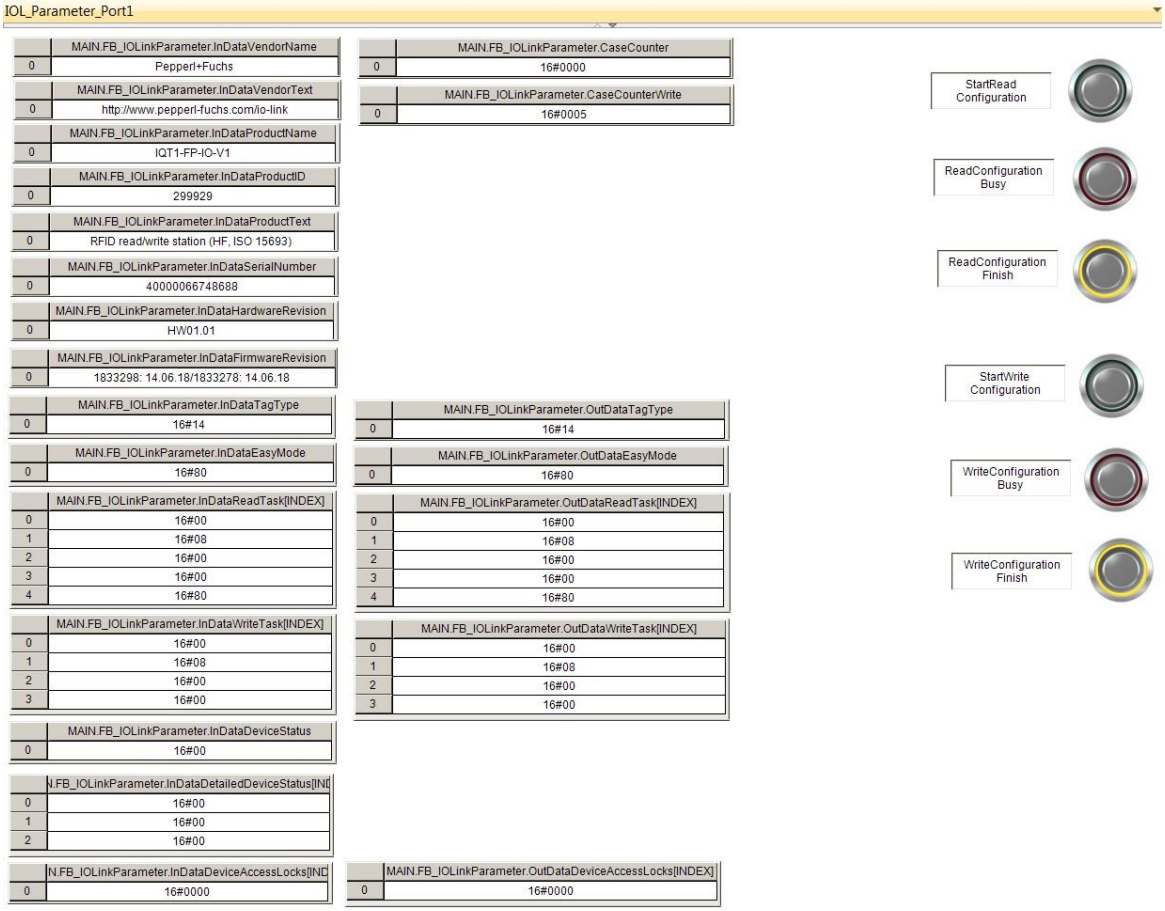
The inputs "ReadConfiguration" and "WriteConfiguration" react to a positive edge. For this reason, these inputs are reset again after the function block has been called.



The AoENetID can be taken from the properties of the IO-Link master EL6224.

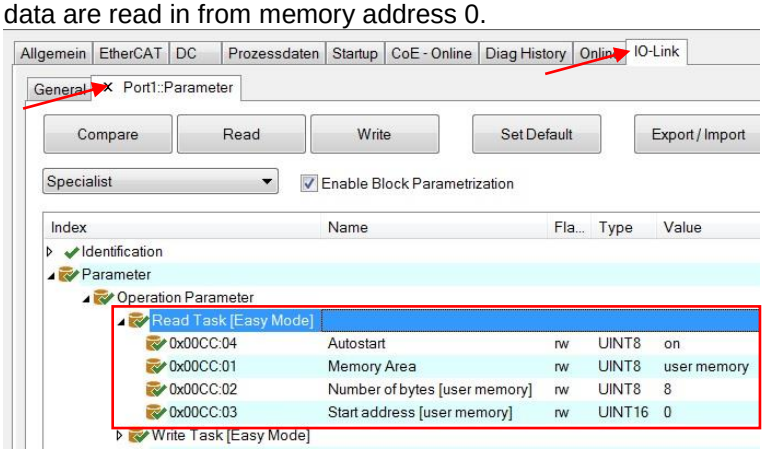
8.2 Visualization function block „FB_IOLinkParameter“

Within the sample project, there are ready-made visualizations. This allows the function block to be controlled and tested via a graphical user interface. The following figure shows the visualization "IOL_Parameter_Port1" for head 1 at port 1.



9. Example: Reading user data with Autostart function

In the delivery state of the IQT1-...-IO-V1, the Autostart function is activated and 8 bytes of the user data are read in from memory address 0.



Default setting Parameter 204 (0x00CC) „Read Task [Easy Mode]
Autostart := ON
Memory Area := User Memory
Number of Bytes := 8
Start address := 0

	MAIN.FB_IOLinkParameter.InDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.InDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.InDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#80

	MAIN.FB_IOLinkParameter.InDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

	MAIN.FB_IOLinkParameter.OutDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.OutDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.OutDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#80

	MAIN.FB_IOLinkParameter.OutDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

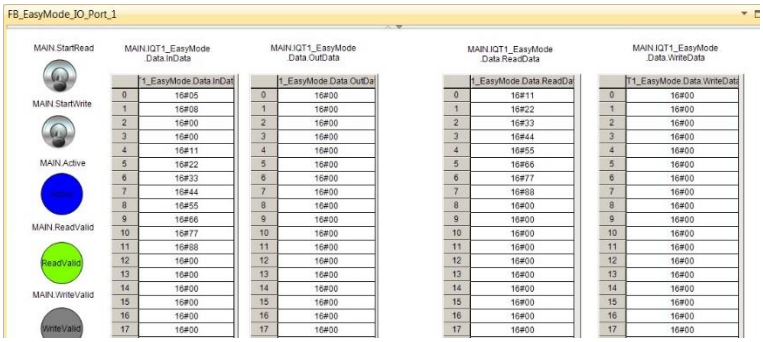
ReadTask[0] := 16#00 = User Memory

ReadTask[1] := 16#08 = Number of Bytes (8)

ReadTask[2][3] := 16#0000 = Start address (0000)

ReadTask[4] := 16#80 = Autostart ON

Data carrier inside detection zone:



	MAIN.IQT1_EasyMode.Data.InData	MAIN.IQT1_EasyMode.Data.OutData	MAIN.IQT1_EasyMode.Data.ReadData	MAIN.IQT1_EasyMode.Data.WriteData
0	16#05	16#00	16#11	16#00
1	16#08	16#00	16#22	16#00
2	16#00	16#00	16#33	16#00
3	16#00	16#00	16#44	16#00
4	16#11	16#00	16#55	16#00
5	16#22	16#00	16#66	16#00
6	16#33	16#00	16#77	16#00
7	16#44	16#00	16#88	16#00
8	16#55	16#00	16#00	16#00
9	16#66	16#00	16#00	16#00
10	16#77	16#00	16#00	16#00
11	16#88	16#00	16#00	16#00
12	16#00	16#00	16#00	16#00
13	16#00	16#00	16#00	16#00
14	16#00	16#00	16#00	16#00
15	16#00	16#00	16#00	16#00
16	16#00	16#00	16#00	16#00
17	16#00	16#00	16#00	16#00

StartRead := False
StartWrite := False
Active = True
ReadValid = True
WriteValid = False
NoTag = False
Error = False
LengthData = 16#08
ReadCounter = > 16#00
WriteCounter = 16#00

MAIN.IQT1_EasyMode .Data.ReadData

	1_EasyMode.Data.ReadDa
0	16#11
1	16#22
2	16#33
3	16#44
4	16#55
5	16#66
6	16#77
7	16#88
8	16#00

The read-in data are located in the data structure "ReadData[0]....[7]".

MAIN.LengthData

	MAIN.LengthData
0	16#08

MAIN.ReadCounter

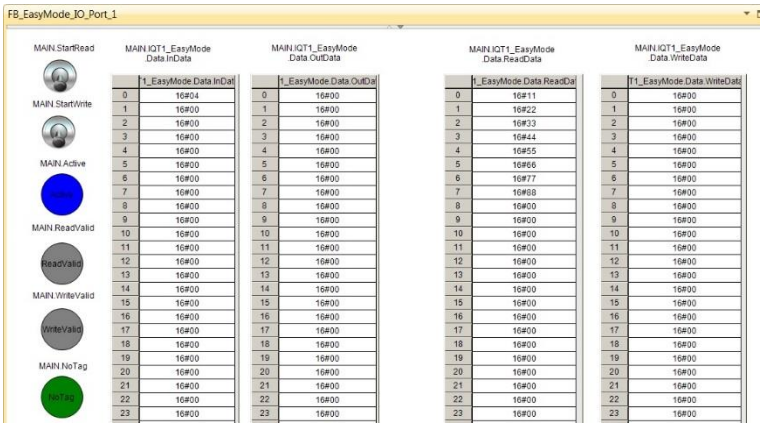
	MAIN.ReadCounter
0	16#0002

MAIN.WriteCounter

	MAIN.WriteCounter
0	16#0000

The output "LengthData" has the value 16#08, since 8 bytes of user data were read in. The output "ReadCounter" is incremented per successful read access.

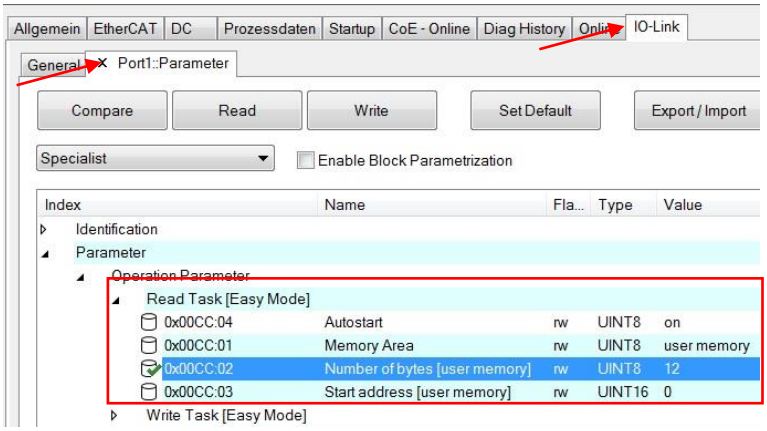
Data carrier outside detection zone:



StartRead := False
StartWrite := False
Active = True
ReadValid = False
WriteValid = False
NoTag = True
Error = False
LengthData = 16#00
ReadCounter = unchanged; no change to the previous read access
WriteCounter = 16#00

The data structure "ReadData" contains the read data of the last successful read access.

The number of user data to be read in can be increased by the IO-Link parameter 204 (0x00CC) "Read Task ".



Change parameter "Number of Bytes" to 12. 12 bytes of user data are thus accessed from address 0.

	MAIN.FB_IOLinkParameter.InDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.InDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.InDataReadTask[INDEX]
0	16#00
1	16#0C
2	16#00
3	16#00
4	16#80

	MAIN.FB_IOLinkParameter.InDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

	MAIN.FB_IOLinkParameter.OutDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.OutDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.OutDataReadTask[INDEX]
0	16#00
1	16#0C
2	16#00
3	16#00
4	16#80

	MAIN.FB_IOLinkParameter.OutDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

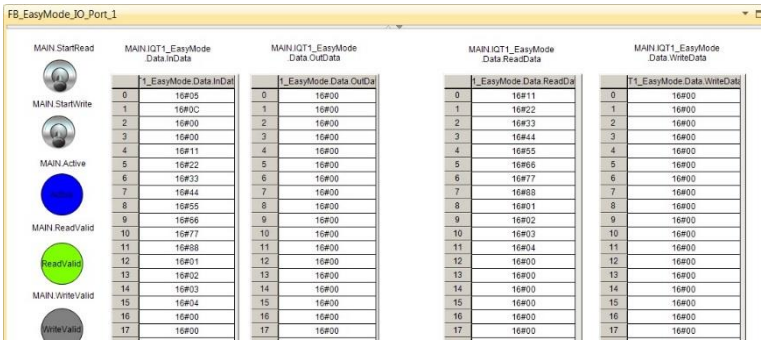
ReadTask[0] := 16#00 = User Memory

ReadTask[1] := 16#0C = Number of Bytes (12)

ReadTask[2][3] := 16#0000 = Start address (0000)

ReadTask[4] := 16#80 = Autostart ON

Data carrier inside detection zone:



StartRead := False
StartWrite := False
Active = True
ReadValid = True
WriteValid = False
NoTag = False
Error = False
LengthData = 16#0C
ReadCounter = > 16#00
WriteCounter = 16#00

MAIN.IQT1_EasyMode .Data.ReadData

The read-in data are located in the data structure "ReadData[0]....[11]".

	1_EasyMode.Data.ReadDa
0	16#11
1	16#22
2	16#33
3	16#44
4	16#55
5	16#66
6	16#77
7	16#88
8	16#01
9	16#02
10	16#03
11	16#04
12	16#00

MAIN.LengthData

	MAIN.LengthData
0	16#0C

MAIN.ReadCounter

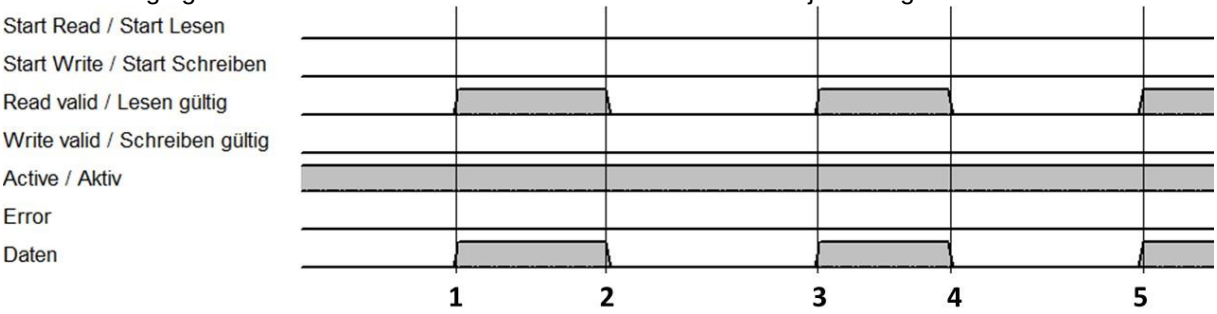
	MAIN.ReadCounter
0	16#0001

MAIN.WriteCounter

	MAIN.WriteCounter
0	16#0000

The output "LengthData" has the value 16#0C, since 12 bytes of user data were read in. The output "ReadCounter" is incremented per successful read access.

The following figure shows the flow chart for the execution of a read job using the Autostart function.



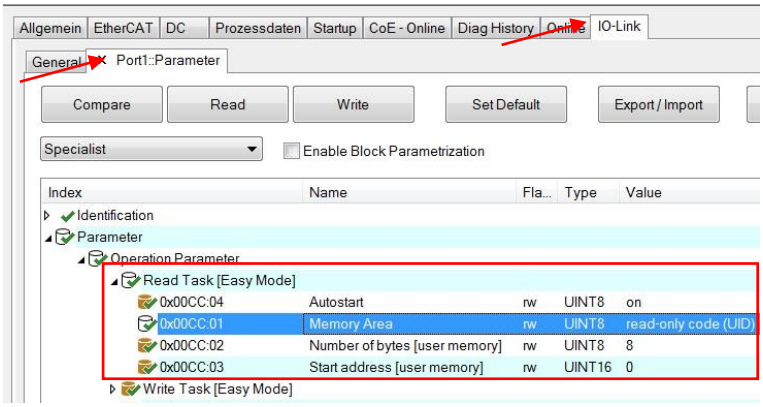
When a read task is executed by the Autostart function, the read task is automatically started by the RFID head. The execution of the reading job is indicated by the bit "Active".

Event	Meaning
1	Data carrier A enters the detection zone of the RFID head and the data are read in; the "Read valid" bit signals that the read-in data are available in the process input data field; for the time period of the data carrier's presence in the detection zone, the "Read valid" bit remains set and the data in the process input data field remains valid
2	Data carrier A leaves the detection zone of the RFID head; the "Read valid" bit is reset, as there is no data carrier within the detection zone; the area with the previously read data is overwritten with 0x00.

3	Data carrier B enters the detection zone of the RFID head and the data are read in; the "Read valid" bit signals that the read-in data are available in the process input data field; for the time period of the data carrier's presence in the detection zone, the "Read valid" bit remains set and the data in the process input data field remains valid
4	Data carrier B leaves the detection zone of the RFID head; the "Read valid" bit is reset as there is no data carrier within the detection area; the area with the previously read data is overwritten with 0x00.
5	Analogous to events 1 and 3

10. Example: Read Fixcode with Autostart function

In addition to the user data (user memory; customer-specific data), it is also possible to read the Fixcode of the data carrier. The fixcode is an 8 byte long unique number for every ISO15693 compliant 13.56MHz data carrier. To read out the Fixcode, the access to the Fixcode must be changed within the parameter 204 (0x00CC) "Read Task [Easy Mode]". The autostart function remains switched on.



Change parameter "Memory Area" to "read only code (UID)". This accesses the unique 8 byte long Fixcode (UID) of the data carrier.

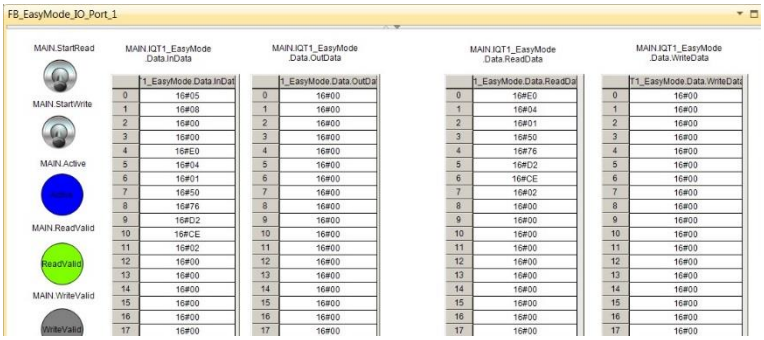
ReadTask[0] := 16#80 = User Memory

ReadTask[1] := 16#08 = Number of Bytes (8); nicht relevant

ReadTask[2][3] := 16#0000 = Start address (0000); nicht relevant

ReadTask[4] := 16#80 = Autostart ON

Data carrier inside detection zone:



StartRead := False
StartWrite := False
Active = True
ReadValid = True
WriteValid = False
NoTag = False
Error = False
LengthData = 16#08
ReadCounter = > 16#00
WriteCounter = 16#00

MAIN.IQT1_EasyMode .Data.ReadData

	1_EasyMode.Data.ReadDa
0	16#E0
1	16#04
2	16#01
3	16#50
4	16#76
5	16#D2
6	16#CE
7	16#02
8	16#00

The read-in Fixcode (UID) is located in the data structure "ReadData[0].....[7]".

MAIN.LengthData

	MAIN.LengthData
0	16#08

MAIN.ReadCounter

	MAIN.ReadCounter
0	16#0001

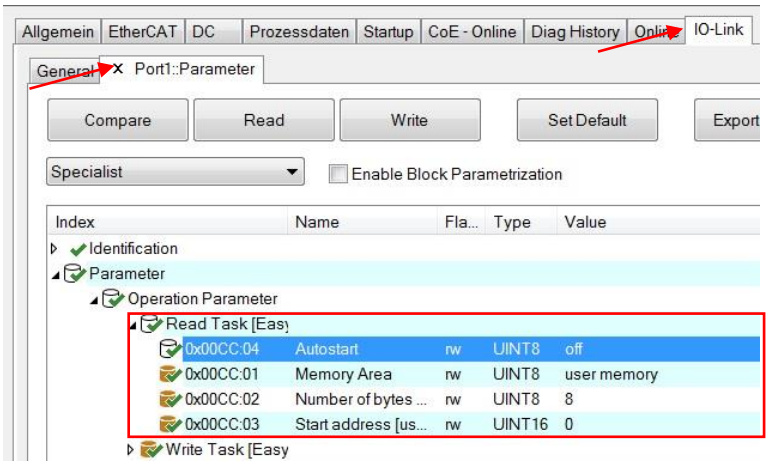
MAIN.WriteCounter

	MAIN.WriteCounter
0	16#0000

The output "LengthData" has the value 16#08, since the Fixcode (UID) has a length of 8 bytes. The output "ReadCounter" is incremented per successful read access.

11. Example: Read user data without Autostart function

In the delivery state of the IQT1-...-IO-V1 the autostart function is activated and 8 bytes of the user data are read in starting from memory address 0. The IO-Link parameter 204 (0x00CC) "Read Task [Easy Mode]" deactivates the autostart function.



Change parameter "Autostart" to "off" (switched off). This means that no read task is started automatically. The read task must therefore be started via the output process data field.

Autostart := OFF
Memory Area := User Memory
Number of Bytes := 8
Start address := 0

	MAIN.FB_IOLinkParameter.InDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.InDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.InDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#00

	MAIN.FB_IOLinkParameter.InDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

	MAIN.FB_IOLinkParameter.OutDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.OutDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.OutDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#00

	MAIN.FB_IOLinkParameter.OutDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

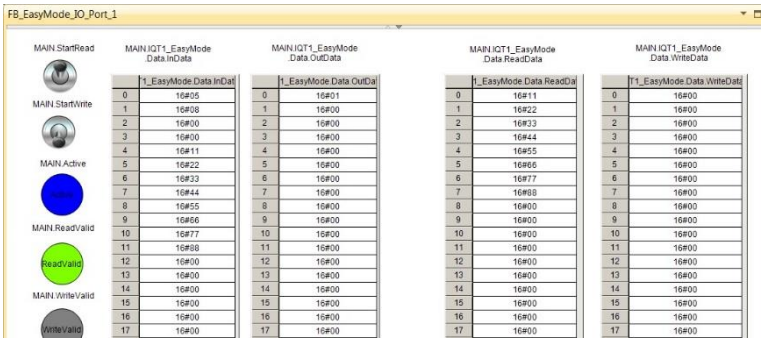
ReadTask[0] := 16#00 = User Memory

ReadTask[1] := 16#08 = Number of Bytes (8)

ReadTask[2][3] := 16#0000 = Start address (0000)

ReadTask[4] := 16#00 = Autostart OFF

Data carrier inside detection zone:



StartRead := True
StartWrite := False
Active = True
ReadValid = True
WriteValid = False
NoTag = False
Error = False
LengthData = 16#08
ReadCounter = > 16#00
WriteCounter = 16#00

MAIN.IQT1_EasyMode .Data.ReadData

The read-in data are located in the data structure "ReadData[0]....[7]".

	1_EasyMode.Data.ReadDa
0	16#11
1	16#22
2	16#33
3	16#44
4	16#55
5	16#66
6	16#77
7	16#88
8	16#00

MAIN.LengthData

	MAIN.LengthData
0	16#08

MAIN.ReadCounter

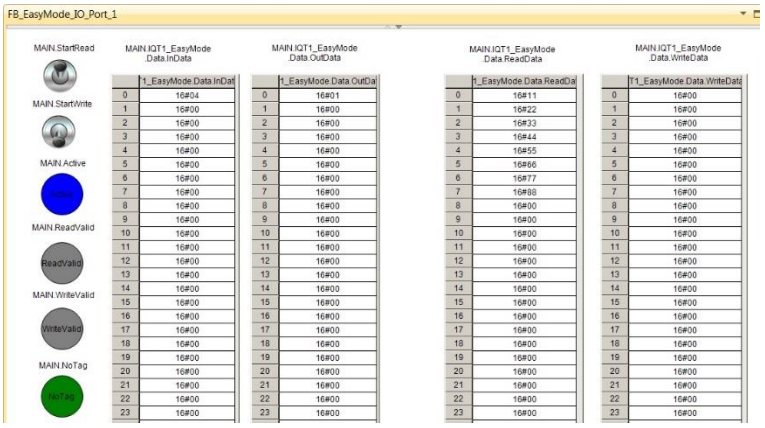
	MAIN.ReadCounter
0	16#0002

MAIN.WriteCounter

	MAIN.WriteCounter
0	16#0000

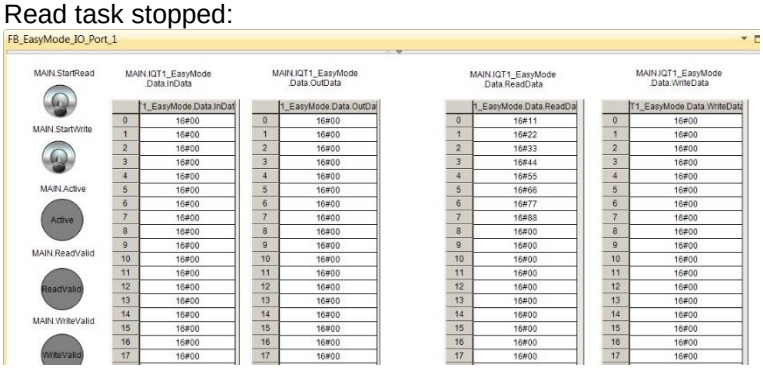
The output "LengthData" has the value 16#08, since 8 bytes of user data were read in. The output "ReadCounter" is incremented per successful read access.

Data carrier inside detection zone:



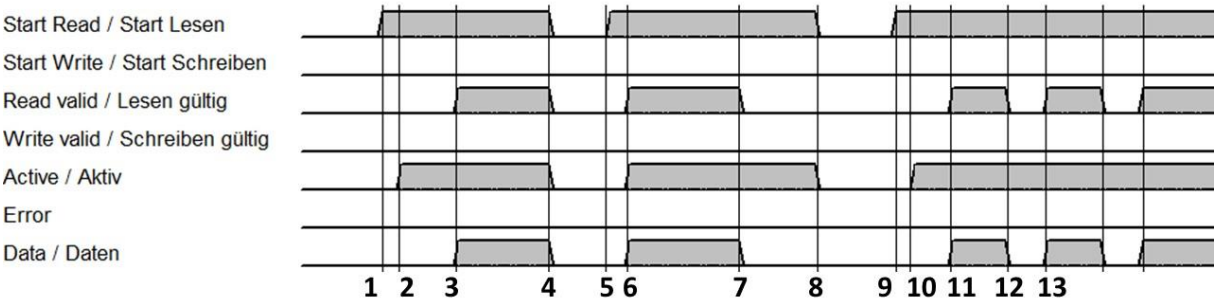
StartRead := True
StartWrite := False
Active = True
ReadValid = False
WriteValid = False
NoTag = True
Error = False
LengthData = 16#00
ReadCounter = unverändert
WriteCounter = 16#00

Read task stopped:



StartRead := False
StartWrite := False
Active = False
ReadValid = False
WriteValid = False
NoTag = False
Error = False
LengthData = 16#00
ReadCounter = 16#00
WriteCounter = 16#00

The following figure shows the flow chart for the execution of a read task without autostart function.

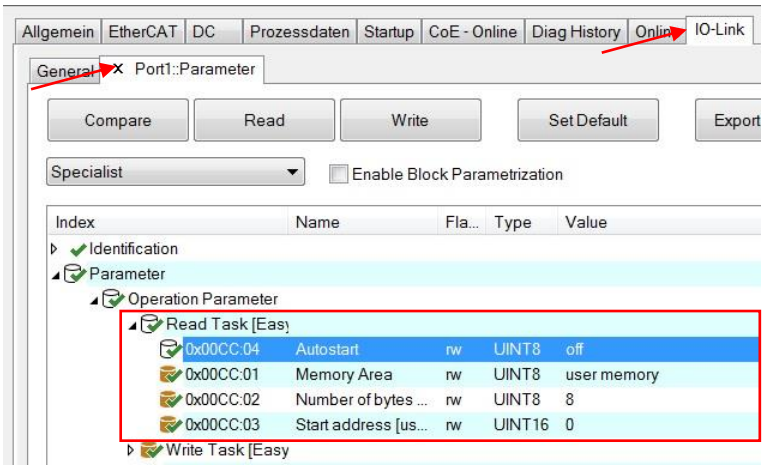


The execution of a read task must be controlled by the bit "Start Read" in the process output data. Automatic control does not take place because the autostart function has been deactivated.

Event	Meaning
1	Start of the execution of a read task by the "Start Read" bit in the process output data; there is currently no data carrier in the detection zone
2	The read request is executed, which is signaled by the bit "Active".
3	A data carrier A enters the detection zone; the "Read valid" bit displays the validity of the imported data in the process input field
4	The read task is stopped by resetting the "Start Read" bit; bits "Read Valid" and "Active" are reset and the read-in data is overwritten with 0x00.
5	Start of the execution of a read task by the "Start Read" bit in the process output data; there is currently a data carrier B in the detection zone.
6	The bit "Active" indicates the execution of the read task; the bit "Read valid" changes to "1", because at the start time of the read task a data carrier was already in the detection zone; the read data are valid.
7	Data carrier B leaves the detection zone which is indicated by a signal change of the bit "Read valid" from "1" to "0"; the read task remains active (bit "Active" = "1"); the data field with the previously read data is overwritten with 0x00.
8	The read task is terminated (bit "Start Read" = "0"); the bit "Active" changes from "1" to "0" to signal the end of the task
9	Start of a read task by changing the signal of the "Start Read" bit from "0" to "1"; there is no data carrier in the detection zone
10	Read task is executed; the bit "Active" is "1".
11	Data carrier C enters the detection zone; the data is read in ("Read valid" = "1") and can be found in the process input data.
12	Data carrier C leaves the detection zone; read task still active ("Active" = "1"); the data field with the previously read data is overwritten with 0x00.
13	Data carrier D enters the detection zone; the data is read in ("Read valid" = "1") and can be found in the process input data.

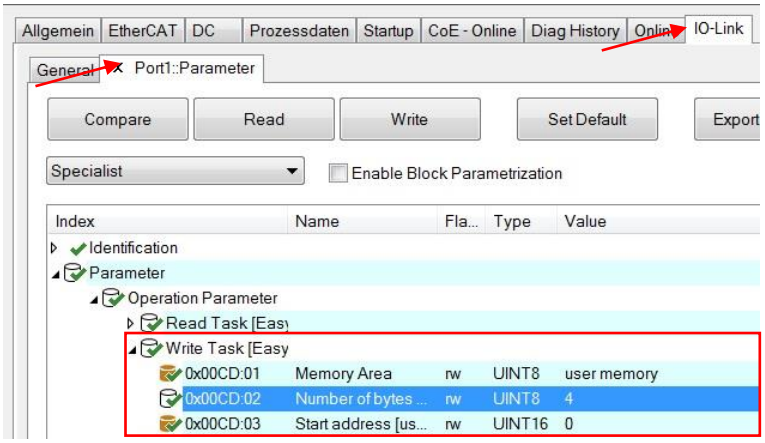
12. Example: Write user data

In the delivery state of the IQT1-...-IO-V1, the Autostart function is activated and 8 bytes of the user data are read in starting from memory address 0. A write operation on a data carrier is not possible when the Autostart function is active. The IO-Link parameter 204 (0x00CC) "Read Task [Easy Mode]" deactivates the autostart function.



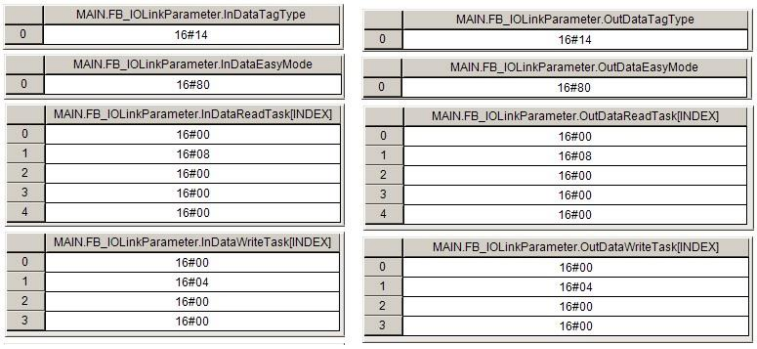
Change parameter "Autostart" to "off" (switched off). This means that no read task is started automatically. A write task can now be started via the process output data field.
Autostart := OFF

The setting of the configuration for writing is carried out via the IO-Link parameter 205 (0x00CD) "Write Task [Easy Mode]". The parameters in the "Write Task" are used to define the memory area to which data is to be written, and the number of bytes with which data is to be written, starting from an address (Start address).



Change parameter "Number of Bytes" to the value 4 (0x04). Thus 4 bytes can be written. The write access to the user data is executed starting at address 0. The write task must be started via the output process data field.

Memory Area := User Memory
Number of Bytes := 4
Start address := 0x0000



	MAIN.FB_IOLinkParameter.InDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.InDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.InDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#00

	MAIN.FB_IOLinkParameter.InDataWriteTask[INDEX]
0	16#00
1	16#04
2	16#00
3	16#00

	MAIN.FB_IOLinkParameter.OutDataTagType
0	16#14

	MAIN.FB_IOLinkParameter.OutDataEasyMode
0	16#80

	MAIN.FB_IOLinkParameter.OutDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#00

	MAIN.FB_IOLinkParameter.OutDataWriteTask[INDEX]
0	16#00
1	16#04
2	16#00
3	16#00

WriteTask[0] := 16#00 = User Memory

WriteTask[1] := 16#04 = Number of Bytes (4)

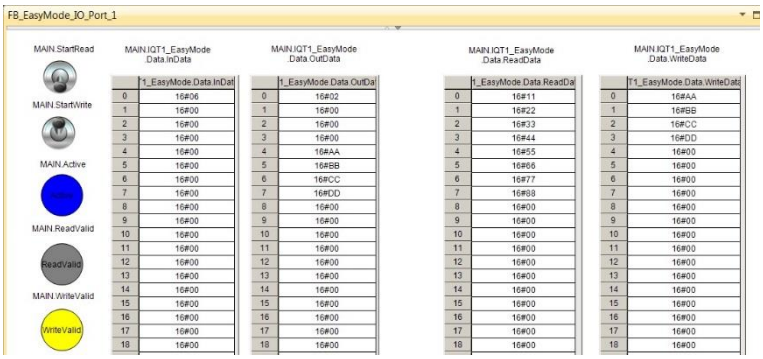
WriteTask[2][3] := 16#0000 = Start address (0000)

MAIN.IQT1_EasyMode .Data.WriteData

	T1_EasyMode.Data.WriteData
0	16#AA
1	16#BB
2	16#CC
3	16#DD
4	16#00

The write data must first be parameterized in the data structure "WriteData[0]....[3]".

Data carrier inside detection zone:



StartRead := False
StartWrite := True
Active = True
ReadValid = False
WriteValid = True
NoTag = False
Error = False
LengthData = 16#00
ReadCounter = 16#00
WriteCounter = >16#00

MAIN.LengthData

	MAIN.LengthData
0	16#00

MAIN.ReadCounter

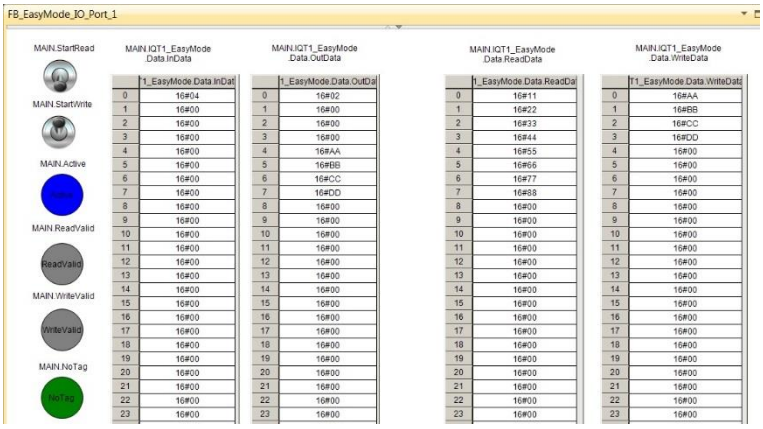
	MAIN.ReadCounter
0	16#0000

MAIN.WriteCounter

	MAIN.WriteCounter
0	16#0001

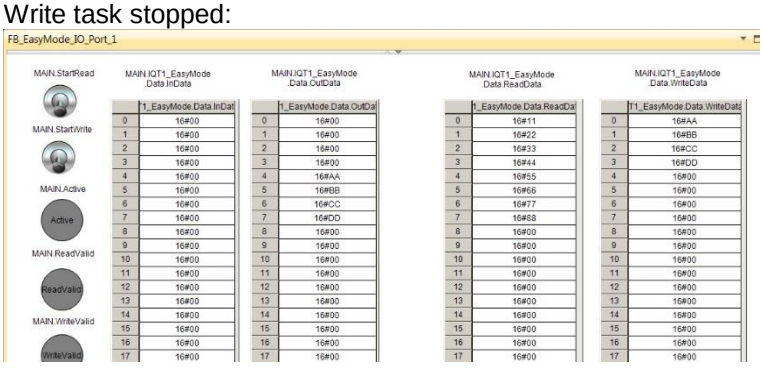
The output "LengthData" has the value 16#00, because no user data was read in. The output "WriteCounter" is incremented per successful write access.

Data carrier outside detection zone:



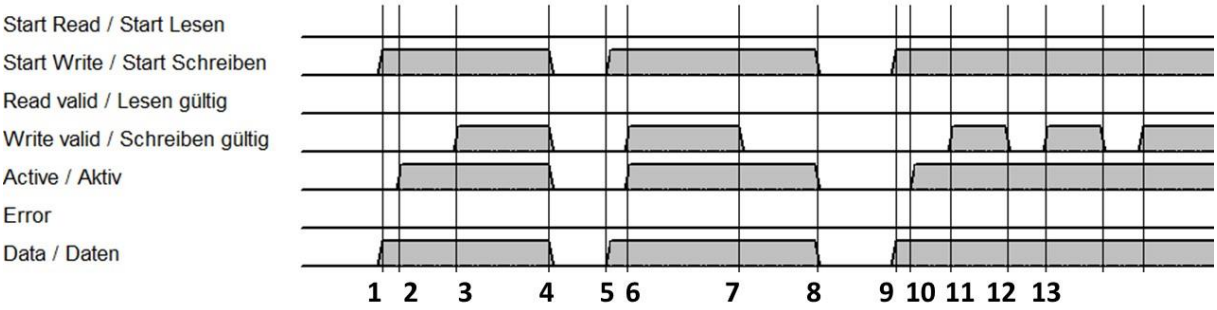
StartRead := False
StartWrite := True
Active = True
ReadValid = False
WriteValid = False
NoTag = True
Error = False
LengthData = 16#00
ReadCounter = 16#00
WriteCounter = unchanged

Write task stopped:



StartRead := False
StartWrite := False
Active = False
ReadValid = False
WriteValid = False
NoTag = False
Error = False
LengthData = 16#00
ReadCounter = 16#00
WriteCounter = 16#00

The following figure shows the flow chart for executing a write task.



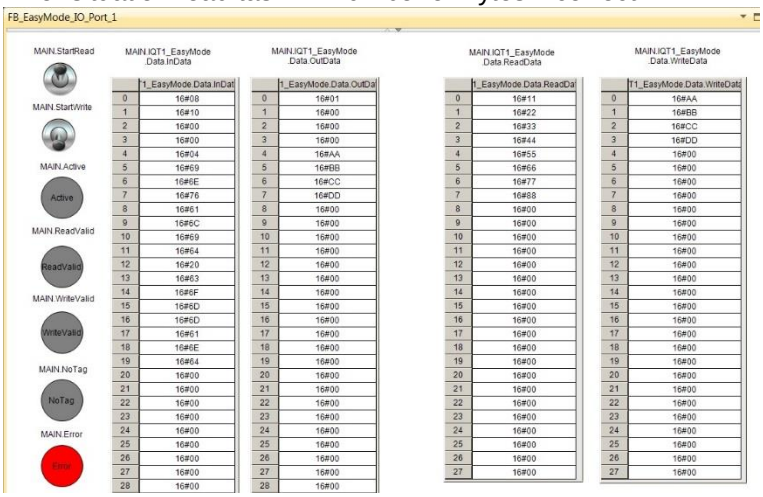
Event	Meaning
1	Start of the execution of a write task by the "Start Write" bit in the process output data; simultaneous transfer of the write data to the process output data field; there is currently no data carrier in the detection zone.
2	The write task is executed, which is indicated by the bit "Active".
3	A data carrier A enters the detection zone; the "Write valid" bit indicates that the data has been successfully written to the data carrier.
4	The write task is stopped by resetting the "Start Write" bit; bits "Write valid" and "Active" are reset and the write data is reset to 0x00.
5	Start of the execution of a write task by the "Start Write" bit in the process output data; there is currently a data carrier B in the detection zone.
6	The bit "Active" indicates the execution of the write task; the bit "Write valid" changes to "1", because at the start time of the write task a data carrier was already in the detection zone; the data have been written successfully.
7	Data carrier B leaves the detection zone which is indicated by a signal change of the bit "Write valid" from "1" to "0"; the write task remains active (bit "Active" = "1");
8	The write task is terminated (bit "Start Write" = "0"); the bit "Active" changes from "1" to "0" to signal the end of the job
9	Start of a write task by changing the signal of bit "Start Write" from "0" to "1"; there is no data carrier in the detection zone
10	Write task is executed; the bit "Active" is "1".
11	Data carrier C enters the detection zone; the data is successfully written to the data carrier ("Write valid" = "1").
12	Data carrier C leaves the detection zone; write task still active ("Active" = "1")
13	Data carrier D enters the detection zone; the data is successfully written to the data carrier ("Write valid" = "1").

13. Example: Error message via process data field

The IQT1-...-IO-V1 RFID head sends an error message to the controller via the process data field when a read or write task is executed. If an error condition occurs, an error code and a short error description are transmitted in plain text (ASCII).

The following figure shows the error message when executing a read task for access to 4 bytes of user data. The error message was generated when a data carrier IQC33 entered the detection zone. This data carrier has a block size of 8 bytes and if these data carriers are used, the number of bytes to be read must be a multiple of 8 bytes.

Error situation read task → Number of Bytes incorrect:



StartRead := True
StartWrite := False
Active = False
ReadValid = False
WriteValid = False
NoTag = False
Error = True
LengthData = 16#10
ReadCounter = 16#00
WriteCounter = 16#00

Process input data field with error message:

MAIN.IQT1_EasyMode .Data.InData

	1_EasyMode.Data.InDat
0	16#08
1	16#10
2	16#00
3	16#00
4	16#04
5	16#69
6	16#6E
7	16#76
8	16#61
9	16#6C
10	16#69
11	16#64
12	16#20
13	16#63
14	16#6F
15	16#6D
16	16#6D
17	16#61
18	16#6E
19	16#64
20	16#00

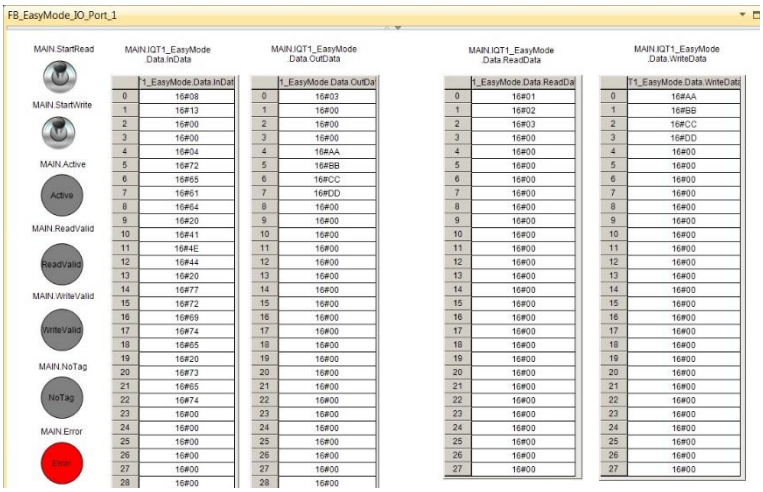
The error code is located within the byte "InData[4]" and has the value 16#04. This signals an error in the command parameters.

The length of the transmitted error message is within the byte "InData[1]". The value within the byte is 16#10. Thus the error message has a length of 16 bytes.

The transmitted error description is "invalid command". This indicates that the command parameter (Number of Bytes) does not match the data carrier used (IQC33).

InData[1] = Length error message
InData[4] = Error code
InData[5]…[19] = Error description

Error state → Write and Read task active at the same time:



StartRead := True
StartWrite := True
Active = False
ReadValid = False
WriteValid = False
NoTag = False
Error = True
LengthData = 16#13
ReadCounter = 16#00
WriteCounter = 16#00

Process input data field with error message:

MAIN.IQT1_EasyMode .Data.InData

	T1_EasyMode.Data.InDat
0	16#08
1	16#13
2	16#00
3	16#00
4	16#04
5	16#72
6	16#65
7	16#61
8	16#64
9	16#20
10	16#41
11	16#4E
12	16#44
13	16#20
14	16#77
15	16#72
16	16#69
17	16#74
18	16#65
19	16#20
20	16#73
21	16#65
22	16#74
23	16#00

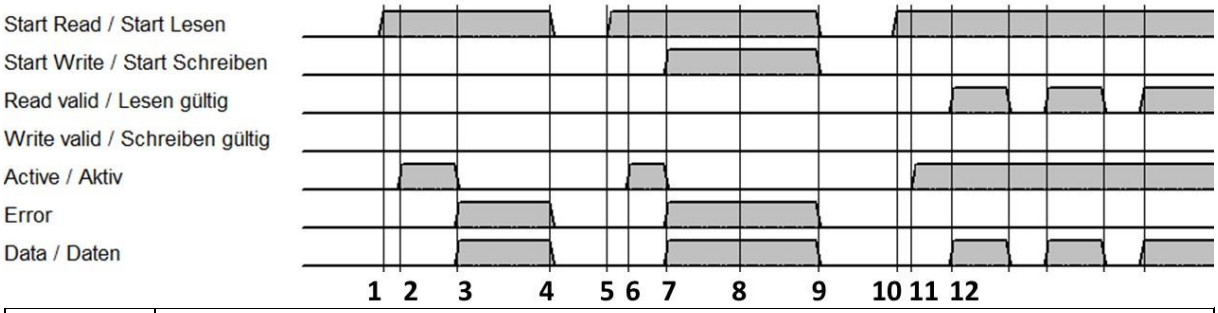
The error code is located within the byte "InData[4]" and has the value 16#04. This signals an error in the command parameters.

The length of the transmitted error message is within the byte "InData[1]". The value within the byte is 16#13, so the error message has a length of 19 bytes.

The transmitted error description is "read and write". This indicates that a read and write job has been triggered at the same time.

InData[1] = Length error message
InData[4] = Error code
InData[5]…[22] = Error description

The following figure shows the flowchart when an error message occurs:



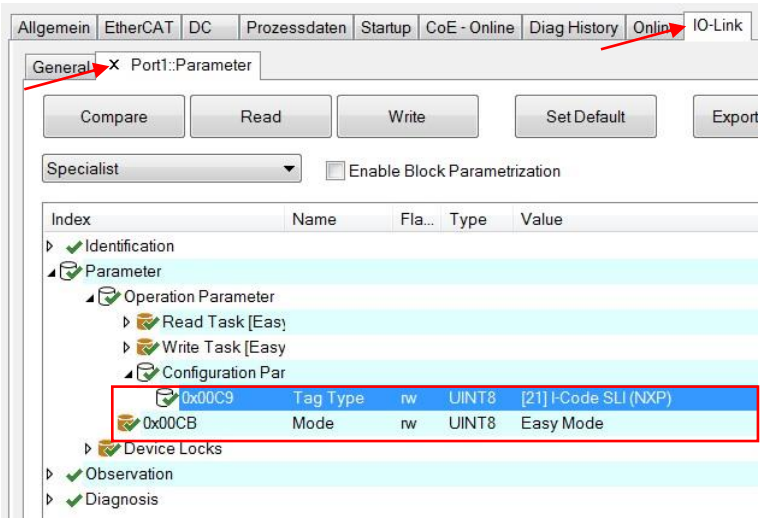
Event	Meaning
1	Start of the execution of a read task by the "Start Read" bit in the process output data; there is currently no data carrier in the detection zone.
2	The read task is executed, which is signaled by the bit "Active".
3	A data carrier A enters the detection zone; the bit "Error" is set and an error message is entered within the data area; the cause of the error is, for example, a number of bytes that is not suitable for the data carrier; the bit "Active" is reset.
4	The read task is stopped by resetting the "Start Read" bit; the "Error" bit is reset and the error message is overwritten with 0x00.
5	Start of the execution of a read task by the "Start Read" bit in the process output data; there is currently no data carrier within the detection zone.
6	The bit "Active" indicates the execution of the read task
7	The "Start Write" bit is set; this causes a read and write task to be active at the same time; this is inadmissible and is signaled by the "Error" bit; an error message is entered in the data area.
8	The bits "Start Read" and "Start Write" are still set and the error message remains unchanged.
9	The bits "Start Read" and "Start Write" are reset; the bit "Error" is reset and at the same time, the error message 0x00 is overwritten.
10	Start read task by "Start Read" = 1
11	Read task is active; signalization by "Active" = 1
12	Data carrier enters the detection zone and is read out successfully; „Read valid" is set

14. Easy-Mode – IO-Link Parameter 201 (0x00C9) „TagType“

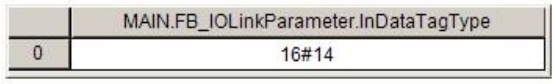
The "TagType" parameter is used to set the chip type of the data carrier used. The RFID head IQT1-...-IO-V1 supports various chip types. Chip type "20" is set in the delivery state. Thus the fixcode of all ISO15693 compatible data carriers can be read out. The following table shows the supported data carrier types.

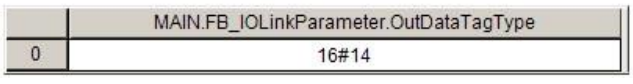
Name	Tag Type	Value (HEX)	Access	Fixcode	Data	Block size	Chip	Frequency
	20	0x14	Read Fixcode	8 Byte	-	-	Each ISO15693	13,56MHz
IQC21	21	0x15	Read Fixcode Read / Write Data	8 Byte	112 Byte	4	I-Code SLI(X)	13,56MHz
IQC22	22	0x16	Read Fixcode Read / Write Data	8 Byte	256 Byte	4	Tag-It HF-I Plus	13,56MHz
IQC23	23	0x17	Read Fixcode Read / Write Data	8 Byte	224 Byte	4	My-d SRF55V02P	13,56MHz
IQC24	24	0x18	Read Fixcode Read / Write Data	8 Byte	928 Byte	4	My-d SRF55V10P	13,56MHz
IQC27	27	0x1B	Read Fixcode Read / Write Data	8 Byte	288 Byte	4	EM4135	13,56MHz
IQC31	31	0x1F	Read Fixcode Read / Write Data	8 Byte	32 Byte	4	Tag-It HF-I Stand-ard	13,56MHz
IQC32	32	0x20	Read Fixcode Read / Write Data	8 Byte	32 Byte	4	Tag-It HF-I Pro	13,56MHz
IQC33	33	0x21	Read Fixcode Read / Write Data	8 Byte	2000 Byte	8	MB89R118	13,56MHz
IQC34	34	0x22	Read Fixcode Read / Write Data	8 Byte	232 Byte	4	MB89R119	13,56MHz
IQC35	35	0x23	Read Fixcode Read / Write Data	8 Byte	256 Byte	4	I-Code SLI-S	13,56MHz
IQC36	36	0x24	Read Fixcode Read / Write Data	8 Byte	32 Byte	4	I-Code SLI-L	13,56MHz

The parameter "TagType" has the index value 201 (0x00C9). The supported tag types are specified by a list.



Change of the IO-Link parameter "TagType" to the data carrier type "21". This corresponds to the chip "I-Code SLI" of the manufacturer NXP.

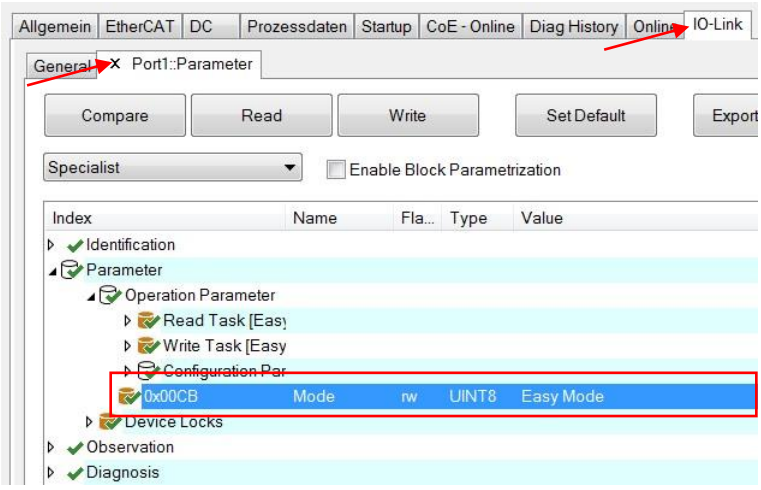




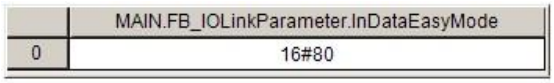
15. Easy-Mode – IO-Link Parameter 203 (0x00CB) „Mode“

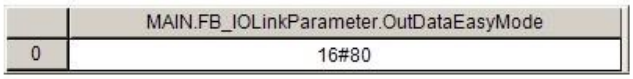
The "Mode" parameter can be used to switch between Easy- and Expert-Mode. The “Easy-Mode” is preset at the factory status and allows simplified data access to the data carrier. This means that no additional function block is required for data transmission.
The "Expert-Mode" allows access to large amounts of data using a handshake procedure. This requires the use of a function block to transfer the data.

The parameter "Mode" has the index value 203 (0x00CB). It can be switched between the values "Easy-Mode" and "Expert-Mode".



IO-Link parameter "Mode" is set to the selection "Easy-Mode". This is the factory setting.





Index	Sub index	length	value (HEX)	Access	Significance
203	0	1 Byte	0x80	Read / Write	Easy mode active; factory setting; allows simplified data access to 28 bytes of user data or fixed code
203	0	1 Byte	0x00	Read / Write	Expert mode active; setting for transferring large amounts of data using handshake methods; use of a function block required

16. Easy-Mode – IO-Link Parameter 204 (0x00CC) „Read Task“

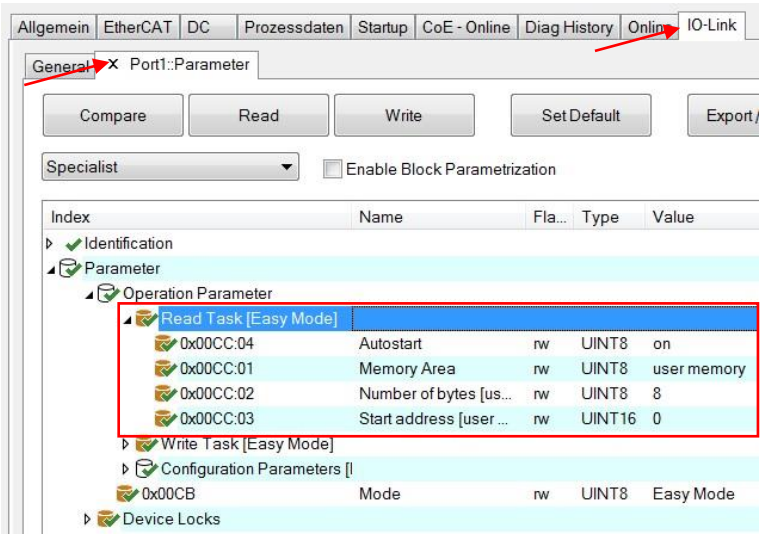
The parameter "Read Task [Easy Mode]" is used to configure the read access to the data carrier. This includes the setting whether the fixcode or the user data are read out. In addition, the number of bytes to be read and the start address are defined. Furthermore it is possible to activate an autostart function. Thus a permanent read command is executed automatically without additional control.

The following table shows the structure of the parameter "Read Task".

Index	Sub index	length	value(HEX)	Access	Significance
204	1	1 Byte	0x00 (user memory)	Read / Write	Access read execution to user data (user memory; customer specific data)
204	1	1 Byte	0x80 (read-only code (UID))	Read / Write	Access reading execution to Fixcode
204	2	1 Byte	0x00 … 0x1C	Read / Write	Number of bytes of user data to be read in; value must be a multiple of 4; when using the IQC33 data carrier, a multiple of 8 must be set
204	3	2 Byte	0x0000 …. 0xFFFF	Read / Write	Start address on data carrier when accessing user data (user data); value must be a multiple

					of 4; when using the IQC33 data carrier a multiple of 8 must be set
204	4	1 Byte	0x80 (on)	Read / Write	Autostart function active; the Autostart function can be used to activate a permanent reading execution; additional control is then no longer required
204	4	1 Byte	0x00 (off)	Read / Write	Autostart function deactivated; read or write must be started by triggering the "Read" or "Write" bit in the output data field

The parameter "Read Task [Easy-Mode]" has the index value 204 (0x00CC).



IO-Link parameter "Read Task [Easy-Mode] in factory setting.
Autostart := on
Memory Area := user memory
Number of Bytes := 8
Start address := 0

	MAIN.FB_IOLinkParameter.InDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#80

	MAIN.FB_IOLinkParameter.OutDataReadTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00
4	16#80

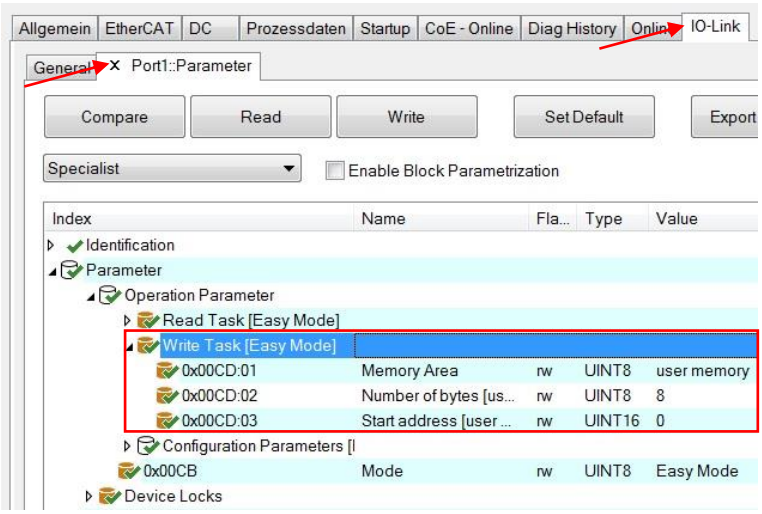
17. Easy-Mode – IO-Link Parameter 205 (0x00CD) „Write Task“

The parameter "Write Task [Easy-Mode]" is used to configure write access to the data carrier. Only the user data can be accessed by writing. In addition, the number of bytes to be written and the start address are set. The configuration of the autostart function is not possible for the write job. The write job is activated via the "Write" bit in the process output data field.

The following table shows the structure of the parameter "Write task".

Index	Sub index	length	Value (HEX)	Access	Significance
205	1	1 Byte	0x00	Read / Write	Access Write request to user data (user memory; customer specific data)
205	2	1 Byte	0x00 … 0x1C	Read / Write	Number of bytes of user data to be added; value must be a multiple of 4; when using the IQC33 data carrier, a multiple of 8 must be set
205	3	2 Byte	0x0000 …. 0xFFFF	Read / Write	Start address on data carrier when accessing user data (user memory; customer-specific data); value must be a multiple of 4; when using the IQC33 data carrier a multiple of 8 must be set

The parameter "Write request" has the index value 205 (0x00CD).



IO-Link Parameter „Write Task [Easy-Mode] in factory setting.
Memory Area := user memory
Number of Bytes := 8
Start address := 0

	MAIN.FB_IOLinkParameter.InDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

	MAIN.FB_IOLinkParameter.OutDataWriteTask[INDEX]
0	16#00
1	16#08
2	16#00
3	16#00

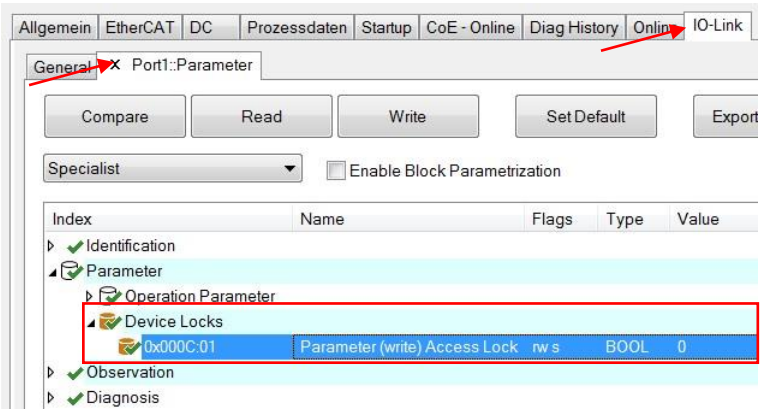
18. Easy-Mode – IO-Link Parameter 12 (0x000C) „Device Access Locks“

The parameter "Device Access Locks" offers the possibility to activate write protection for the device parameters. This means that the IO-Link device parameters can no longer be changed. In addition, the data storage of the device can be switched off.

The following table shows the structure of the "Device Access Locks" parameter.

Index	Sub index	length	value (HEX)	Access	Significance
12	0	2 Byte	0x0000	Read / Write	Not locked, parameters can be changed
12	0	2 Byte	0x0001	Read / Write	Lock for changing parameters
12	0	2 Byte	0x0002	Read / Write	Data storage lock
12	0	2 Byte	0x0003	Read / Write	Lock for changing parameters and data storage

The parameter "Device Access Locks" has the index value 12 (0x000C).



IO-Link parameter "Device Access Lock" in the factory setting. A lock for parameter access is not activated. The parameters can be changed.

	N.FB_IOLinkParameter.InDataDeviceAccessLocks[IND
0	16#0000

	MAIN.FB_IOLinkParameter.OutDataDeviceAccessLocks[INDEX]
0	16#0000

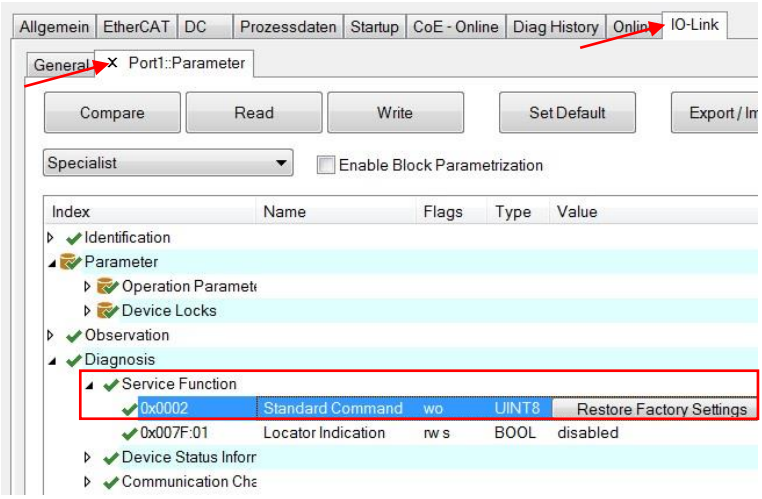
19. Easy-Mode – IO-Link Parameter 2 (0x0002) „System Command“

The parameter "System Command" offers the possibility to reset the IO-Link parameters to the factory settings. Please note that access to the IO-Link parameters is enabled (Device Access Locks not activated). The factory setting is only active after a manual power interruption.

The following table shows the structure of the "System Command" parameter.

Index	Sub index	length	value (HEX)	Access	Significance
2	0	1 Byte	0x82	Write	Reset to factory setting

The parameter "System Command" has the index value 2 (0x0002).



IO-Link "System Command" parameter for resetting to the factory setting.