

# **SICK** **RFU610 IO-Link Function Block**

SICK RFU610 IO-Link function block  
for Siemens S7-1200 / S7-1500 PLCs  
(TIA-Portal V14 or higher)



## Version history

Block Version	Date	Remark
V1.0	08.02.2022	Initial version
V1.1	02.01.2023	Updated documentation
V1.2	28.02.2023	Added remark for mode change

## Table of content

<b>1 About this document</b>	<b>3</b>
1.1 Function of this document	3
1.2 Target group	3
<b>2 General information</b>	<b>4</b>
<b>3 Hardware Configuration</b>	<b>5</b>
3.1 Supported PLCs	5
3.2 TIA-Portal Configuration	5
<b>4 Function block</b>	<b>6</b>
4.1 Block specifications	6
4.2 Operation of the function block	7
4.2.1 Mode	7
4.2.2 Trigger type	7
4.3 Value transfer	8
4.3.1 WriteTag	8
4.3.2 ReadTag	9
4.3.3 SetTagAccessPwd	10
4.3.4 SetTagKillPwd	10
4.3.5 LockKill	10
4.3.6 ParamAccess	10
4.4 Behavior when error occurs	13
<b>5 Parameter</b>	<b>14</b>
<b>6 Error description</b>	<b>16</b>
<b>7 Example</b>	<b>19</b>
7.1 Implementation	19
7.2 Write user memory	20
7.3 Read user memory	21
7.4 Change Operation Mode	22
7.5 Change Trigger Type	23

## **1 About this document**

Please read this chapter carefully before you start working with this technical information and the FB\_SICK\_RFU610\_IOL function block.

### **1.1 Function of this document**

This technical information describes how to use the FB\_SICK\_RFU610\_IOL function block. It is used for guiding technical personnel working for the machine manufacturer / operator in project planning and commissioning.

### **1.2 Target group**

This technical information is aimed for specialists, such as technicians and engineers.

## 2 General information

The function block “FB\_SICK\_RFU610\_IOL” simplifies the use of RFU610 RFID interrogators on Siemens S7-1200 / S7-1500 PLCs. The device must be embedded into the IO-Link environment of the PLC-Controller.

The function block enables reading and writing tag data as well as controlling the RFU610 device via the cyclic IO-Link process data channel.

### Functionalities:

- Write up to 14 words tag data
- Read up to 14 words tag data (in Read/Write Mode up to 13 words)
- Set tag access password<sup>i</sup>
- Set tag kill password<sup>i</sup>
- Lock / Kill tag<sup>i</sup>
- Read/Write parameters
  - Operation Mode
  - Read Power
  - Absolute RSSI Filter
  - Trigger Source
  - Stop Condition
  - Reading gate length
  - Start of object trigger delay

Figure 1 shows the concept behind the RFU610 IO-Link PLC integration.

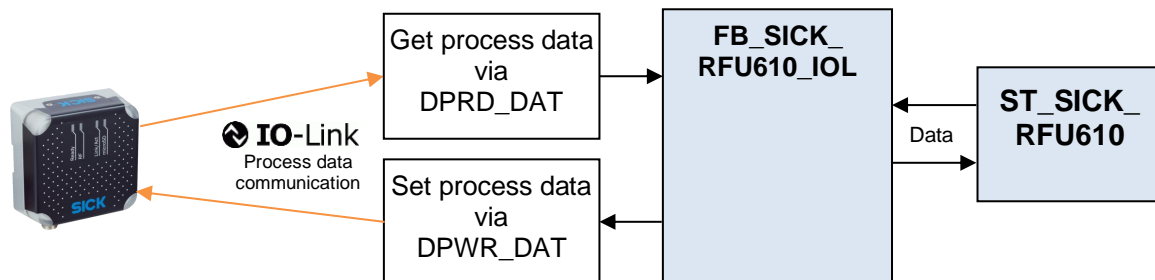


Figure 1: Concept behind the RFU610 IO-Link function block

<sup>i</sup> IO-Link service data units (ISDUs) are used to parameterize these functions.

## 3 Hardware Configuration

### 3.1 Supported PLCs

The function block can only be used with S7-1200 (FW V4.0 or higher) / S7-1500 PLCs which are programmed with TIA-Portal V14 (or higher).



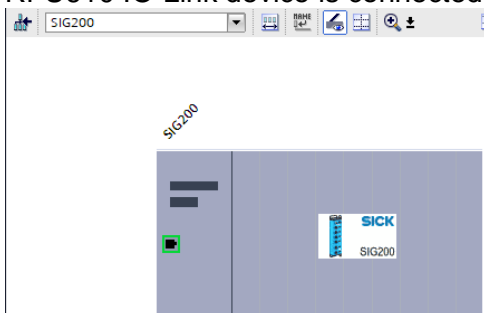
#### Please note!

The function block is IO-Link Master independent and can be used for all available Masters.

### 3.2 TIA-Portal Configuration

Before the function block can be used, an IO-Link Master device must be configured in the hardware configuration. The IO-Link port used for communication with the RFU610 must support a process data length of 32byte In/Out.

Figure 2 shows an example projecting of a SICK SIG200 IO-Link master (PROFINET). The RFU610 IO-Link device is connected to the first master port.



Module	Rack	Slot	I address	Q address	Type
▼ SIG200	0	0			SIG200
▶ Interface	0	0 X1			SIG200
▼ SIG200 Global Functions_1	0	1			SIG200 Global Functions
Global configuration	0	1 1			Global configuration
Logic editor 128I / 128O	0	1 2	256...383	256...383	Logic editor 128I / 128O
▼ SIG200 IO-Link Master_1	0	2			SIG200 IO-Link Master
SIG200 IO-Link Master	0	2 1	0		SIG200 IO-Link Master
IO-Link 32I / 32O + PQI	0	2 2	1...33	0...31	IO-Link 32I / 32O + PQI
Digital input	0	2 3	34		Digital input
Digital input_1	0	2 4	35		Digital input
Digital input_2	0	2 5	36		Digital input

Figure 2: Example hardware configuration

## 4 Function block

This function block (FB) simplifies the usage of a SICK RFU610 RFID interrogator in combination with a Siemens S7-1200 / S7-1500 PLC. The FB uses only the IO-Link process data communication channel to interact with the device. For device parameter access, the device uses the acyclic IO-Link service data communication channel.

The function block works asynchronously, which means the processing requires several call ups. Therefore, it is necessary that the function block is called cyclically in the user program.

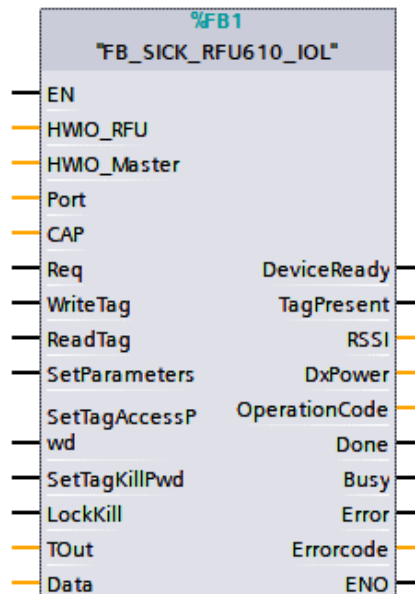


Figure 3: FB\_SICK\_RFU610\_IOL function block

### 4.1 Block specifications

Block name:	FB_SICK_RFU610_IOL
Used PLC data types:	ST_SICK_RFU610 ST_SICK_RFU610_Error
Function block call up:	Cyclically
Optimized block access:	Yes
Used flags:	No
Language:	Structured Language (SCL)
Developed with:	STEP7 Professional V14

## 4.2 Operation of the function block

Each block action (#ReadTag, #WriteTag etc.) can be parameterized via the data type "ST\_SICK\_RFU610" (Data). To execute a function block action, the desired action must be selected first. It is not possible to select more than one action at time. To execute the selected action, the parameter #Req or an external connected digital signal must be triggered with a positive edge (signal change from a logical zero to one). The #Busy signal is set to TRUE as long as no valid device answer has been received. As soon as a falling edge is detected on the #Req input, the active command (e.g., #ReadTag) is cancelled. This behavior prevents ISDU Parameter access during an active read or write access.

If the function block signals #Done = TRUE at the output parameter, the action has been done successfully. If, for this action (e.g. "ReadTag") data has been requested from the device, it will be copied into the respective data area (#Data).

### 4.2.1 Mode

The RFU can be operated in two different modes, which can be set via ISDU (index 10#120, 16#78) manually or via the ParamAccess function, see Chapter 4.3.6 ParamAccess. Each mode has its own process data format. The detected mode is shown on the function block output #OperationMode. The mode can only be changed when the device is not busy. So make sure, there is no RFID-Tag on the Reader when changing from mode 0 to 1 and there is no function being executed when changing from mode 1 to 0.

Value	Mode	Description
0	ReadUII	Cyclic readout of the UII without trigger.
1	ReadWrite	Allows reading / writing of transponder contents of the different memory banks.

Table 1: Device mode



#### Please note!

If the ReadUII Mode (#OperationMode = 0) is active, the inputs #WriteTag or #ReadTag will be functionless. If the trigger shall be set manually, set the OperationMode to 1 (ReadWrite). For further information see Chapter 7.5 Change Trigger Type

### 4.2.2 Trigger type

Depending on the trigger mode, the trigger window is opened/closed via the function block (Software-Trigger) or via a signal connected directly to the RFU (Hardware-Trigger). The trigger type can be changed via an ISDU (index 310) or via SOPAS-ET. The detected mode is shown on the function block output #TriggerType.

Value	Type	Description
0	Software-Trigger	The trigger window is opened on a read/write request (#Req) and is closed according to the stop condition (index 312). If the stop condition is set to "Trigger input", the trigger window is closed by resetting the #Req flag.

1	Hardware-Trigger	In this case the #Req flag has no effect. The selected function is executed as soon as the external digital signal (e.g. light switch) opens / closes a trigger window.
---	------------------	---

Table 2: Trigger type

### 4.3 Value transfer

The data type ST\_SICK\_RFU610 contains all input and output parameter values of the supported function block actions. The function block uses an instance (variable) of this data type called #Data. The data structure is pre-defined and should not be changed.

ST_SICK_RFU610									
	Name	Datentyp	Defaultwert	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Kommentar	
1	WriteTag	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Write tag parameters	
2	Bank	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Memory bank to be written. 0=Reserved, 1=UII/EPC, 2=TID, 3=UMEM	
3	StartWord	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	First word to be written beginning with 0	
4	WordCount	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Number of words to be written	
5	Data	Array[0..27] of Byte		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data to be written to the selected tag area	
6	ReadTag	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read tag parameters	
7	Bank	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Memory bank to be read out. 0=Reserved, 1=UII/EPC, 2=TID, 3=UMEM	
8	StartWord	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	First word to be read beginning with 0	
9	WordCount	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Number of words to be read out	
10	DataLengthReceived	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Received data length in byte	
11	Data	Array[0..25] of Byte		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Data that were read out	
12	TagPasswords	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tag passwords	
13	TagAccessPwd	Array[0..3] of Byte		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tag access password to be written to the tag. Each octet stands for one character	
14	KillTagPwd	Array[0..3] of Byte		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Kill password to be written to the tag. Each octet stands for a one character	
15	Parameters	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
16	Values	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFU Parameter Values - multiselection is possible	
17	Mode	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Process Data Select [0..1] - Index 16#78, 10#120	
18	ReadPower	UDInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read Power - [0..140]e-1 dBm - Index 16#190, 10#400	
19	RSSIFilter	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Absolute RSSI Filter - Index 16#19A, 10#410	
20	On/Off	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RSSI Filter On/Off, Subindex 1	
21	LowerThreshold	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RSSI Filter Threshold [-100..0]. Subindex 2	
22	TriggerType	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trigger Source, [0..1] Index 16#136, 10#310	
23	TriggerStop	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Stop Condition, [0..1] Index 16#138, 10#312	
24	ReadingGateLength	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Reading gate length, [0..1] Index 16#13A, 10#314	
25	TriggerDelay	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Start of object trigger delay, [0..10000] ms Index 16#137, 10#311	
26	Selection	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFU Parameter Selection	
27	Mode	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
28	ReadPower	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
29	RSSIFilter	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
30	TriggerType	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
31	TriggerStop	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
32	ReadingGateLength	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
33	TriggerDelay	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #SetParameters Request	
34	RW	Bool	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFU Read or Write Parameter, FALSE = READ, TRUE = WRITE	

Figure 4: ST\_SICK\_RFU610 data type

#### 4.3.1 WriteTag

The WriteTag function is used to write values to a defined area of a TAG.

The following parameters must be set in the #Data.WriteTag structure before executing this block function.

Parameter	Declaration	Data type	Description
WriteTag. Bank	Input	USINT	Memory bank to be written. 0= Reserved 1= UII/EPC 2=TID 3=UMEM  <u>Valid range:</u> [0..3]
WriteTag. StartWord	Input	UINT	First word (16 bit) to be written beginning with 0.



Parameter	Declaration	Data type	Description
WriteTag. WordCount	Input	UINT	Number of words to be written.  <u>Valid range:</u> [1..14]
WriteTag. Data	Input	ARRAY [0..27] OF BYTE	Data to be written to the selected tag area. The length is defined by the WordCount parameter.  A maximum of 14 words (28 byte) can be written.

Table 3: Parameter of the WriteTag function

### 4.3.2 ReadTag

The ReadTag function is used to read a defined area of a TAG.

The following parameters must be set in the #Data.ReadTag structure before executing this block function.

Parameter	Declaration	Data type	Description
ReadTag. Bank	Input	USINT	Memory bank to be read. 0= Reserved 1= UII/EPC 2=TID 3=UMEM  <u>Valid range:</u> [0..3]
ReadTag. StartWord	Input	UINT	First word (16 bit) to be read beginning with 0.
ReadTag. WordCount	Input	UINT	Number of words to be read out.  <u>Valid range:</u> [1..13]
ReadTag. DataLengthRe- ceived	Output	USINT	Received data length in byte
Data	Output	ARRAY [0..25] OF BYTE	Data that were read out

Table 4: Parameter of the ReadTag function

### 4.3.3 SetTagAccessPwd

This function is used to set a password for TAG-Access on the TAG. Only if the password on the RFU and the TAG match, a TAG access is possible.

The following parameters must be set in the #Data.TagPasswords structure before executing this block function.

Parameter	Declaration	Data type	Description
TagPasswords. TagAccessPwd	Input	ARRAY [0..3] OF BYTE	Tag access password to be written to the tag. Each octet stands for one character.  <u>Example:</u> Password = 1234ABCD  [0]= 0x12 [1]= 0x34 [2]= 0xAB [3]= 0xCD

Table 5: Parameter of the SetTagAccessPwd function

### 4.3.4 SetTagKillPwd

This function is used to set a password for transponder kill on the TAG. Only if the password on the RFU and the TAG match, a TAG kill is possible.

The following parameters must be set in the #Data.TagPasswords structure before executing this block function.

Parameter	Declaration	Data type	Description
TagPasswords. KillTagPwd	Input	ARRAY [0..3] OF BYTE	Kill password to be written to the tag. Each octet stands for a one character  <u>Example:</u> Password = 1234ABCD  [0]= 0x12 [1]= 0x34 [2]= 0xAB [3]= 0xCD

Table 6: Parameter of the SetKillTagPwd function

### 4.3.5 LockKill

This function is used to lock or kill a tag. Whether a tag will be locked or killed depends on the value in the ISDU on Index 0x1A8 (Subindex 1) "Command". If the ISDU is '0', the tag will be locked, if the ISDU is '1', the tag will be killed. If a tag is password protected, the password needs to be defined on Index 0x1A6 when locking or Index 0x1A7 when killing a tag. These parameters can be accessed via the ParamAccess function, see chapter 4.3.6 ParamAccess.

### 4.3.6 ParamAccess

This function is used to set device parameters also known as ISDUs (Indexed **S**ervice **D**ata **U**nits). There are several parameters, that may be written to. The parameters which shall be

written to the device, must be selected via the #Data.ParamAccess.Selection structure. The value of the selected parameters must be defined via the #Data.ParamAccess.Values structure. There may be more than one parameter selected per command request, as selected parameters are queued and processed in serial. The #Data.ParamAccess.RW bit is used to decide whether a value should be read or written to.

Values	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFU Parameter Values - multiselection is possible
Mode	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Process Data Select [0..1] - Index 16#78, 10#120
ReadPower	UDInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read Power - [0..140]e-1 dBm - Index 16#190, 10#400
RSSIFilter	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Absolute RSSI Filter - Index 16#19A, 10#410
On/Off	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RSSI Filter On/Off, Subindex 1
LowerThreshold	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RSSI Filter Threshold [-100..0], Subindex 2
TriggerType	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Trigger Source, [0..1] Index 16#136, 10#310
TriggerStop	USInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Stop Condition, [0..1] Index 16#138, 10#312
ReadingGateLength	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Reading gate length, [0..1] Index 16#13A, 10#314
TriggerDelay	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Start of object trigger delay, [0..10000] ms Index 16#137, 10#311
DeviceAccessPassword	Array[1..8] of Char		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Device Access password, Index 16#1A6, 10#422
DeviceKillPassword	Array[1..8] of Char		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Device Kill password, Index 16#1A7, 10#423
LockKillSelector	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Lock/Kill Selector, Index 16#1A8, 10#424
Command	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Defines the command Lock(0) or Kill(1), [0..1] Subindex 1
BankUserMemory	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bank UserMemory. Preserve(0), Unlock(1), Lock(2), Unlock permanently(3), Lock p...
BankEPC	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bank EPC. Preserve(0), Unlock(1), Lock(2), Unlock permanently(3), Lock permanent...
BankTID	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bank TID. Preserve(0), Unlock(1), Lock(2), Unlock permanently(3), Lock permanent...
BankTagAccess	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bank Tag Access. Preserve(0), Unlock(1), Lock(2), Unlock permanently(3), Lock per...
BankTagKill	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bank Tag Kill. Preserve(0), Unlock(1), Lock(2), Unlock permanently(3), Lock perman...
Selection	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFU Parameter Selection
Mode	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
ReadPower	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
RSSIFilter	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
TriggerType	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
TriggerStop	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
ReadingGateLength	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
TriggerDelay	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
DeviceAccessPassword	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
DeviceKillPassword	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
LockKillSelector	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	If true, parameter will be written to RFU on the next #ParamAccess Request
RW	Bool	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	RFU Read or Write Parameter, FALSE = READ, TRUE = WRITE

Figure 5: Parameters structure

The following table shows the supported parameters.

Parameter (ISDU)		Data type	Description
Mode		UInt	Process Data Select [0..1] Index: 0x78
ReadPower		UDInt	Read Power [0..140] e-1 dBm Index: 0x190
RSSIFilter		Struct	RSSI Filter Index: 0x19A
L	"On/Off"	Bool	Absolute RSSI Filter – RSSI Filter On/Off Subindex: 1
L	LowerThreshold	Int	Absolute RSSI Filter – RSSI Filter Threshold [-100..0] Subindex: 2
TriggerType		USInt	Trigger Source [0..1] Index: 0x136

TriggerStop		USInt	Trigger Stop Condition [0..1] Index: 0x138
ReadingGateLength		UInt	Reading Gate Length [0..1] Index: 0x13A
TriggerDelay		UInt	Start of object trigger delay [0..10000] ms Index: 0x137
DeviceAccessPassword		Array[1..8] of Char	Device access password UTF8 formatted Index: 0x1A6
DeviceKillPassword		Array[1..8] of Char	Device kill password UTF8 formatted Index: 0x1A7
LockKillSelector		Struct	Lock/Kill Selector Index: 0x1A8
L	Command	UInt	Defines the command Lock(0) or Kill(1) [0..1] Subindex: 1
L	BankUserMemory	UInt	Defines the Bank User Memory [0..4] Subindex: 2
L	BankEPC	UInt	Defines the Bank User Memory [0..4] Subindex: 3
L	BankTID	UInt	Defines the Bank User Memory [0..4] Subindex: 4
L	BankTagAccess	UInt	Defines the Bank User Memory [0..4] Subindex: 5
L	BankTagKill	UInt	Defines the Bank User Memory [0..4] Subindex: 6

**Please note!**

The command will only be executed if there is no tag is being accessed simultaneously (Read or Write). The ParamAccess Request is latched, so the command will be executed as soon as the tag access is over.

---

For further informations see the device IODD: <https://ioddfinder.io-link.com/productvariants/search/36951>

#### 4.4 Behavior when error occurs

If there is a wrong input value of the function block, an error bit (`#Error`) is set, and an error code (`#Errorcode`) will be given out. In this case, there is no further processing. The diagnosis parameter (`#Error` and `#Errorcode`) of the routine maintains their value until a new request has been started. Two types of error sources can lead to the error bit (`#Error`) being set. If an error occurs during the processing of a command or the device signals that an error has occurred. For diagnostic purposes, the `#ErrorCode` structure can be evaluated.

## 5 Parameter

Parameter	Declaration	Data type	Description
EN	Input	BOOL	Enable input (only in the FBD view)
TOut	Input	TIME	Time after a timeout error occurs.
HWIO_RFU	Input	HW_SUBMODULE	HW-Identifier of the RFU610 IO-Link device.
HWIO_Master	Input	HW_IO	HW-Identifier of the IO-Link Master.
Port	Input	Int	Port number, to which the IO-Link device is connected to on the Master.
CAP	Input	DINT	<b>CAP (Client Access Point)</b>  Known values: <ul style="list-style-type: none"> <li>- Siemens IOL-M: 227</li> <li>- SICK SIG200: 46080</li> <li>- Others: 255</li> </ul>
Req	Input	BOOL	A rising edge executes the selected actions.
TriggerType	Input	USINT	Currently set trigger mode. The trigger type can be changed via an ISDU or via SOPAS-ET. 0= Software-Trigger 1= Hardware-Trigger
WriteTag	Input	BOOL	Write tag data.  Data to be written needs to be defined in the corresponding structure (#Data).
ReadTag	Input	BOOL	Read tag data.  Data to be read needs to be defined in the corresponding structure (#Data).
ParamAccess	Input	BOOL	Access device Parameters. Grants read and write access to selected ISDUs.  Parameter to be read or written to needs to be defined in the corresponding structure (#Data).
SetTag AccessPwd	Input	BOOL	Write tag access password to the transponder.
SetTag KillPwd	Input	BOOL	Write kill tag password to the transponder.
LockKill	Input	BOOL	Locks / kills the tag area defined via an ISDU parameter.
Data	In/Out	ST_SICK_RFU610	Contains input and output parameters for all supported function block actions.
DeviceReady	Output	BOOL	Indicates if the RFU is ready. <b>This flag is updated cyclically.</b>
TagPresent	Output	BOOL	Indicates if there is a tag in the RF field of the RFU610.
RSSI	Output	SINT	RSSI signal level coming from the transponder.

Parameter	Declaration	Data type	Description
TxPower	Output	USINT	Used transmission power [dBm]
OperationMode	Output	UINT	Indicates the current operation mode (0 = ReadEPC, 1 = Read/Write)
TriggerType	Output	UINT	Indicates the current trigger type (0 = Software-trigger, 1 = Hardware-trigger)
Done	Output	BOOL	Indicates that the selected function block action has been performed without errors.
Busy	Output	BOOL	Request in process  FALSE: Request is terminated TRUE: Request is being processed
ReqDone	Output	BOOL	Indicates that the selected acyclic request has been performed without errors. (#SetParameters)
ReqBusy	Output	BOOL	Indicates that an acyclic request is being process.
Error	Output	BOOL	An error occurred.  FALSE: No error TRUE: Error detected
Errorcode	Output	ST_SICK_RFU610_Error	Error information (see error code description)
ENO	Output	BOOL	Enable output (only in the FBD view)

Table 7: Function block parameters

## 6 Error description

The parameter "Errorcode" contains the following error information:

- Block specific error code
- Device error code
- Extended error code

Block Errorcode	Description
16#0000	No error
16#0001	Timeout error occurred. The processing of the actions takes longer than the time set at the #Tout parameter.  The timeout is only active when "Software-Trigger" is used and the #Tout parameter is >0s.
16#0002	A request was executed without selecting a parameter, or with more than one parameter selected.
16#0003	Unsupported memory bank (>3)
16#0004	Reserved
16#0005	No transponder present in the RF-Field of the RFU
16#0006	Write tag: Wordcount >14 not usable in mode 1
16#0007	Read tag: Wordcount >13 not usable in mode 1
16#0008	Unsupported device mode (>1)
16#0009	Reserved
16#000E	An error occurred while writing to the output process image.  The variable ExtendedErrorcode contains the status of the MOVE_BLK_VARIANT function. Please use the TIA-Portal help system to get more information.
16#000F	An error occurred while reading the input process image.  The variable "ExtendedErrorcode" contains the status of the MOVE_BLK_VARIANT function. Please use the TIA-Portal help system to get more information.



Block Errorcode	Description																																																
16#0010	<p>Device error detected</p> <p>The variable "DeviceErrorcode" contains the device error code.</p> <table> <tr> <th>Error-code</th><th>Description</th></tr> <tr> <td>0x0001</td><td>Device not ready</td></tr> <tr> <td>0x0002</td><td>Request execution failed</td></tr> <tr> <td>0x0003</td><td>Invalid number of requested data</td></tr> <tr> <td>0x0004</td><td>Invalid telegram length</td></tr> <tr> <td>0x0005</td><td>More than one tag detected</td></tr> <tr> <td>0x0006</td><td>Static tag in field</td></tr> <tr> <td>0x0007</td><td>Received ULL is too long</td></tr> <tr> <td>0x0008</td><td>Requested write operation failed</td></tr> <tr> <td>0x0009</td><td>Timeout occurred during confirmed messaging protocol was processing</td></tr> <tr> <td>0x000A</td><td>Wrong data standard was detected (other coding scheme expected)</td></tr> <tr> <td>0x000B</td><td>Lock command failed</td></tr> <tr> <td>0x000C</td><td>Kill command failed</td></tr> <tr> <td>0x000D</td><td>Wrong trigger input active (HW trigger selected by ISDU, but master sends trigger requests via PDO)</td></tr> <tr> <td>0x000E</td><td>Logging active, but no SD card inserted</td></tr> <tr> <td>0x000F</td><td>Logging active, but SD card failed</td></tr> <tr> <td>0x1000</td><td>Net interface logging: write error on SD-card</td></tr> <tr> <td>0x1003</td><td>Net interface logging: no free space on SD-card</td></tr> <tr> <td>0x200A</td><td>Reverse power too high</td></tr> <tr> <td>0x200F</td><td>Timeout</td></tr> <tr> <td>0x2010</td><td>Commands not allowed at the time (e.g., outside temperature too low)</td></tr> <tr> <td>0x2015</td><td>Power amplifier temperature threshold exceeded</td></tr> <tr> <td>0x2018</td><td>MAC: Communication Error</td></tr> <tr> <td>0x20FF</td><td>MAC: Generic Error</td></tr> </table>	Error-code	Description	0x0001	Device not ready	0x0002	Request execution failed	0x0003	Invalid number of requested data	0x0004	Invalid telegram length	0x0005	More than one tag detected	0x0006	Static tag in field	0x0007	Received ULL is too long	0x0008	Requested write operation failed	0x0009	Timeout occurred during confirmed messaging protocol was processing	0x000A	Wrong data standard was detected (other coding scheme expected)	0x000B	Lock command failed	0x000C	Kill command failed	0x000D	Wrong trigger input active (HW trigger selected by ISDU, but master sends trigger requests via PDO)	0x000E	Logging active, but no SD card inserted	0x000F	Logging active, but SD card failed	0x1000	Net interface logging: write error on SD-card	0x1003	Net interface logging: no free space on SD-card	0x200A	Reverse power too high	0x200F	Timeout	0x2010	Commands not allowed at the time (e.g., outside temperature too low)	0x2015	Power amplifier temperature threshold exceeded	0x2018	MAC: Communication Error	0x20FF	MAC: Generic Error
Error-code	Description																																																
0x0001	Device not ready																																																
0x0002	Request execution failed																																																
0x0003	Invalid number of requested data																																																
0x0004	Invalid telegram length																																																
0x0005	More than one tag detected																																																
0x0006	Static tag in field																																																
0x0007	Received ULL is too long																																																
0x0008	Requested write operation failed																																																
0x0009	Timeout occurred during confirmed messaging protocol was processing																																																
0x000A	Wrong data standard was detected (other coding scheme expected)																																																
0x000B	Lock command failed																																																
0x000C	Kill command failed																																																
0x000D	Wrong trigger input active (HW trigger selected by ISDU, but master sends trigger requests via PDO)																																																
0x000E	Logging active, but no SD card inserted																																																
0x000F	Logging active, but SD card failed																																																
0x1000	Net interface logging: write error on SD-card																																																
0x1003	Net interface logging: no free space on SD-card																																																
0x200A	Reverse power too high																																																
0x200F	Timeout																																																
0x2010	Commands not allowed at the time (e.g., outside temperature too low)																																																
0x2015	Power amplifier temperature threshold exceeded																																																
0x2018	MAC: Communication Error																																																
0x20FF	MAC: Generic Error																																																
16#0011	Selected command is not supported in ReadEPC Mode (Mode = 0)																																																
16#0200	There are no parameters selected in #Data.ParamAccess.Selection. Select at least on parameter and re-trigger the command.																																																
16#0210	One or more values that shall be written to the parameters are not supported. Make sure the values are in the supported range and re-trigger the command.																																																
16#0300	<p>An error occurred during the initialization of the function block. Could not read the length and address of the provided In-/Output HW_Submodule.</p> <p>For further informations see the #ExtendedErrorcode and the TIA documentation for the RD_ADDR function.</p>																																																

Block Errorcode	Description
16#0301	<p>An error occurred during the initialization of the function block. The detected length of the provided In-/Output HW_Submodule #HWIO_RFU does not fulfill the requirements.</p> <p>Make sure the provided HW_Submodule has a width of 32 Byte In- and 32 Byte Output.</p>

*Table 8: Function block error codes*

## 7 Example

This example shows how to use the function block to interact with the device. In the example project, the RFU device is connected on port 2 of the SICK IOLG2 IO-Link Master.

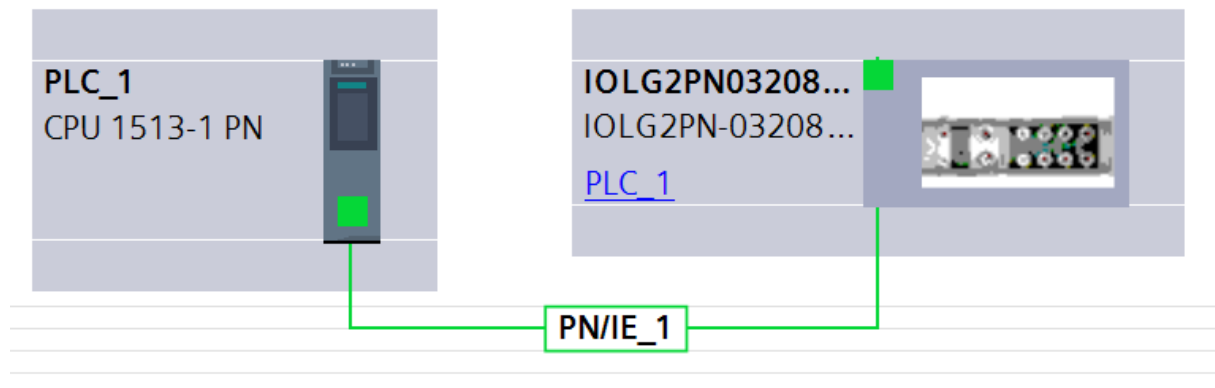
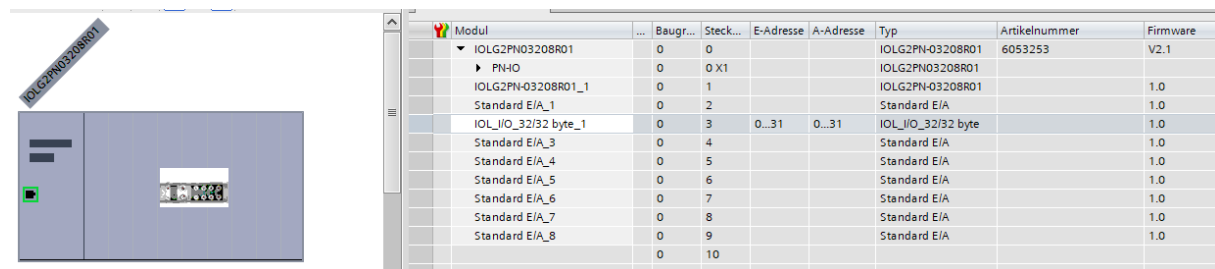


Figure 6: Hardware configuration, device view

The RFU610 IO-Link has a process data width of 32 Bytes on the In- and Output process data image. Depending on whether a PQI is provided by the IOL Master, the Input process data image can be 33 Bytes wide.



Modul	Baugr...	Steck...	E-Adresse	A-Adresse	Typ	Artikelnummer	Firmware
IOLG2PN03208R01	0	0			IOLG2PN-03208R01	6053253	V2.1
PNHO	0	0 X1			IOLG2PN03208R01		
IOLG2PN-03208R01_1	0	1			IOLG2PN-03208R01		1.0
Standard E/A_1	0	2			Standard E/A		1.0
IOL_IIO_32/32 byte_1	0	3	0...31	0...31	IOL_IIO_32/32 byte		1.0
Standard E/A_3	0	4			Standard E/A		1.0
Standard E/A_4	0	5			Standard E/A		1.0
Standard E/A_5	0	6			Standard E/A		1.0
Standard E/A_6	0	7			Standard E/A		1.0
Standard E/A_7	0	8			Standard E/A		1.0
Standard E/A_8	0	9			Standard E/A		1.0
	0	10					

Figure 7: Hardware configuration, module view

### 7.1 Implementation

The following code is inserted in the main routine. IDB\_RFU610 is the instance-DB of the FB\_SICK\_RFU610\_IOL. Please use the watch table "IOLCall" and "Example" to control this program.

```

(*===== RFU =====*)
IDB_RFU610 (HWIO_RFU:=258,
            HWIO_Master:=283,
            CAP:=255,
            TOut:=T#5s,
            Data := "DB_Data".stRFUData);

```

Figure 8: Code example

## 7.2 Write user memory

In this example, a part of the user memory is written using the function block.

Prerequisites:

- Operation mode = 1 (ReadWrite)
- Trigger mode = 0 (Software-Trigger)

First, the data to be written to the user memory must be defined.

"DB_Data".stRFUData.WriteTag.Bank	DEC	3	User memory
"DB_Data".stRFUData.WriteTag.StartWord	DEC	0	Start word
"DB_Data".stRFUData.WriteTag.WordCount	DEC	3	Number of words to be written
"DB_Data".stRFUData.WriteTag.Data[0]	Character	'H'	Content
"DB_Data".stRFUData.WriteTag.Data[1]	Character	'e'	
"DB_Data".stRFUData.WriteTag.Data[2]	Character	'I'	
"DB_Data".stRFUData.WriteTag.Data[3]	Character	'I'	
"DB_Data".stRFUData.WriteTag.Data[4]	Character	'o'	
"DB_Data".stRFUData.WriteTag.Data[5]	Character	' '	

The writing function (#WriteTag) is executed as soon as the bit #Req is triggered with a rising edge.

"IDB_RFU610".TOUT	Time	T# 5S	
"IDB_RFU610".Req	Bool	<input checked="" type="checkbox"/> TRUE	Rising edge: Start write process
"IDB_RFU610".Mode	DEC	1	
"IDB_RFU610".TriggerType	DEC	0	
"IDB_RFU610".WriteTag	Bool	<input checked="" type="checkbox"/> TRUE	Select: Write tag
"IDB_RFU610".ReadTag	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".SetTagAccessPwd	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".SetTagKillPwd	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".LockKill	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".DeviceReady	Bool	<input checked="" type="checkbox"/> TRUE	
"IDB_RFU610".TagPresent	Bool	<input checked="" type="checkbox"/> TRUE	
"IDB_RFU610".RSSI	DEC+/-	-43	
"IDB_RFU610".DxPower	DEC	13	
"IDB_RFU610".Done	Bool	<input checked="" type="checkbox"/> TRUE	Write process performed successfully
"IDB_RFU610".Busy	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".Error	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".Errorcode.BlockErrorcode	Hex	16#0000	
"IDB_RFU610".Errorcode.ExtendedErrorcode	Hex	16#0000	
"IDB_RFU610".Errorcode.DeviceErrorcode	Hex	16#0000	

The writing function is completed as soon as bit #Done indicates TRUE.

### 7.3 Read user memory

In this example, the first 3 words of the user-memory is read out using the function block.

Prerequisites:

- Device mode 1 (ReadWrite)
- Trigger mode 0 (Software-Trigger)

First, the data to be read out must be define in the data structure.

"DB_Data".stRFUData.ReadTag.Bank	DEC	3	User memory
"DB_Data".stRFUData.ReadTag.StartWord	DEC	0	Start word
"DB_Data".stRFUData.ReadTag.WordCount	DEC	3	Number of words to be read

The reading function (#ReadTag) is executed as soon as the bit #Req is triggered with a rising edge.

"IDB_RFU610".TOUT	Time	T#5S	
"IDB_RFU610".Req	Bool	<input checked="" type="checkbox"/> TRUE	Rising edge: Start read process
"IDB_RFU610".Mode	DEC	1	
"IDB_RFU610".TriggerType	DEC	0	
"IDB_RFU610".WriteTag	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".ReadTag	Bool	<input checked="" type="checkbox"/> TRUE	Select: Read tag
"IDB_RFU610".SetTagAccessPwd	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".SetTagKillPwd	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".LockKill	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".DeviceReady	Bool	<input checked="" type="checkbox"/> TRUE	
"IDB_RFU610".TagPresent	Bool	<input checked="" type="checkbox"/> TRUE	
"IDB_RFU610".RSSI	DEC+/-	-43	
"IDB_RFU610".DxPower	DEC	13	
"IDB_RFU610".Done	Bool	<input checked="" type="checkbox"/> TRUE	Read process performed successfully
"IDB_RFU610".Busy	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".Error	Bool	<input type="checkbox"/> FALSE	
"IDB_RFU610".Errorcode.BlockErrorcode	Hex	16#0000	
"IDB_RFU610".Errorcode.ExtendedErrorcode	Hex	16#0000	
"IDB_RFU610".Errorcode.DeviceErrorcode	Hex	16#0000	

The reading function is completed as soon as bit #Done indicates TRUE. Afterwards, the content is available in the #Data array.

"DB_Data".stRFUData.ReadTag.Bank	DEC	3	
"DB_Data".stRFUData.ReadTag.StartWord	DEC	0	
"DB_Data".stRFUData.ReadTag.WordCount	DEC	3	
"DB_Data".stRFUData.ReadTag.DataLengthReceived	DEC	6	Received data length (byte)
"DB_Data".stRFUData.ReadTag.Data[0]	Character	'H'	Readout content
"DB_Data".stRFUData.ReadTag.Data[1]	Character	'e'	
"DB_Data".stRFUData.ReadTag.Data[2]	Character	'I'	
"DB_Data".stRFUData.ReadTag.Data[3]	Character	'I'	
"DB_Data".stRFUData.ReadTag.Data[4]	Character	'o'	
"DB_Data".stRFUData.ReadTag.Data[5]	Character	' '	
"DB_Data".stRFUData.ReadTag.Data[6]	Character	'\$00'	
"DB_Data".stRFUData.ReadTag.Data[7]	Character	'\$00'	

## 7.4 Change Operation Mode

To change the operation mode of the RFU610-IOL, the ParamAccess function is used. Therefore, the parameters of parameters must be set. Relevant parameters for the change of the operation mode are #ParamAccess.RW, #ParamAccess.Values.Mode and #ParamAccess.Selection.Mode.

"DB_Data".strFUDData.ParamAccess.RW	BOOL	<input checked="" type="checkbox"/> TRUE
"DB_Data".strFUDData.ParamAccess.Selection.Mode	BOOL	<input checked="" type="checkbox"/> TRUE
"DB_Data".strFUDData.ParamAccess.Values.Mode	DEZ	0

Figure 9: Necessary parameters for the change of operation mode

At last the action has to be triggered. Therefore select #ParamAccess and put a rising edge on the #Req input.

"IDB_RFU610".Req	BOOL	<input checked="" type="checkbox"/> TRUE
"IDB_RFU610".WriteTag	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".ReadTag	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".SetTagAccessPwd	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".SetTagKillPwd	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".LockKill	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".ParamAccess	BOOL	<input checked="" type="checkbox"/> TRUE

Figure 10: FB inputs for the ParamAccess Function

Once the action has been triggered, the #ReqBusy output will be set to busy until the action is finished. Once the action is finished, the #ReqDone output will be set to true.

"IDB_RFU610".ReqBusy	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".ReqDone	BOOL	<input checked="" type="checkbox"/> TRUE

Figure 11: State outputs

The function block output #OperationMode outputs the value '0', the operation has successfully changed the operation mode.

"IDB_RFU610".DeviceReady	BOOL	<input checked="" type="checkbox"/> TRUE
"IDB_RFU610".TagPresent	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".RSSI	DEZ+/-	0
"IDB_RFU610".TxPower	DEZ	10
"IDB_RFU610".OperationMode	DEZ	0

Figure 12: Active operation mode



### Please note!

The operation mode can only be changed when the device is idle. Therefore, when changing modes from 0 to 1 that there is no tag present.



## 7.5 Change Trigger Type

To change the operation mode of the RFU610-IOL, the ParamAccess function is used. Therefore, the parameters of parameters must be set. Relevant parameters for the change of the operation mode are #ParamAccess.RW, #ParamAccess.Values.TriggerType and #ParamAccess.Selection.TriggerType. In this example, the Trigger Type will be changed from 0 to 1, from software to hardware trigger.

"DB_Data".stRFUData.ParamAccess.RW	BOOL	<input checked="" type="checkbox"/> TRUE
"DB_Data".stRFUData.ParamAccess.Selection.TriggerType	BOOL	<input checked="" type="checkbox"/> TRUE
"DB_Data".stRFUData.ParamAccess.Values.TriggerType	DEZ	1

Figure 13: Necessary parameters for the change of trigger type

At last the action has to be triggered. Therefore select #ParamAccess and put a rising edge on the #Req input.

"IDB_RFU610".Req	BOOL	<input checked="" type="checkbox"/> TRUE
"IDB_RFU610".WriteTag	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".ReadTag	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".SetTagAccessPwd	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".SetTagKillPwd	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".LockKill	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".ParamAccess	BOOL	<input checked="" type="checkbox"/> TRUE

Figure 14: FB inputs for the ParamAccess Function

Once the action has been triggered, the #ReqBusy output will be set to busy until the action is finished. Once the action is finished, the #ReqDone output will be set to true.

"IDB_RFU610".ReqBusy	BOOL	<input type="checkbox"/> FALSE
"IDB_RFU610".ReqDone	BOOL	<input checked="" type="checkbox"/> TRUE

Figure 15: State outputs

The function block output #TriggerType outputs the value '1', the operation has successfully changed the trigger type.

"IDB_RFU610".TriggerType	DEZ	1
--------------------------	-----	---

Figure 16: Active operation mode