

SICK **RFH5xx Add-On Instruction**

SICK RFH5xx IO-Link Add-On Instruction
for Rockwell ControlLogix / CompactLogix PLCs
(Studio5000 V16 or higher)



Version history

Block Version	Date	Remark
V1.0	14.10.2021	Initial version
V1.1	15.11.2021	<ul style="list-style-type: none"> • Read/Write UMem process updated • Data size for read/write can be adjusted

Table of content

1 About this document.....	3
1.1 Function of this document	3
1.2 Target group	3
2 General information	4
3 Hardware Configuration.....	5
3.1 Supported PLCs.....	5
3.2 Studio5000 Configuration.....	5
4 Add-On Instruction.....	6
4.1 Block specifications.....	6
4.2 Process data handover to the AOI	7
4.3 Operation of the AOI	7
4.4 Data type description	8
4.4.1 AntennaField	8
4.4.2 ReadUID	8
4.4.3 ReadUMem	9
4.4.4 WriteUMem	9
4.1 Adjust maximum data length	10
4.2 Behavior when error occurs	10
5 Parameter.....	11
6 Error description	13
7 Example	14
7.1 Implementation	14
7.1.1 Initialization.....	14
7.1.2 Read tag parameterization	15
7.1.3 Write tag parameterization.....	15
7.1.4 Start read / write request	15
7.1.5 Handle response data	15
7.1.6 Add-On Instruction call up	16
7.2 Writing user data.....	17
7.3 Reading user data.....	17

1 About this document

Please read this chapter carefully before you start working with this technical information and the SICK_RFH5xx_IOL Add-On Instruction Add-On Instruction.

1.1 Function of this document

This technical information describes how to use the SICK_RFH5xx_IOL Add-On Instruction. 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 Add-On Instruction “SICK_RFH5xx_IOL” simplifies the use of RFH5xx RFID interrogators on Rockwell ControlLogix / CompactLogix PLCs. The device has to be embedded into the IO-Link surrounding of the PLC-Controller.

The Add-On Instruction enables reading and writing tag data as well as controlling the RFH5xx device via the cyclic IO-Link process data channel.

Functionalities:

- Read UID
- Write user memory up to 512 bytes
- Read user memory up to 512 bytes
- Switch RF-Field power on/off
- Tag present indication
- Antenna state
- Information about RSSI value

Figure 1 shows the concept behind the RFH5xx IO-Link PLC integration.

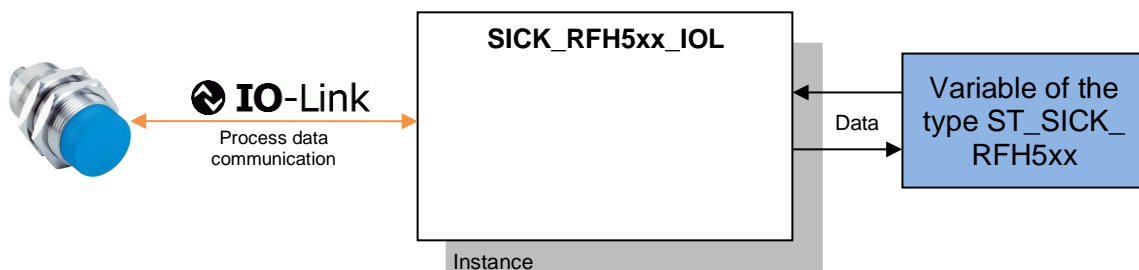


Figure 1: Concept behind the RFH5xx IO-Link Add-On Instruction

3 Hardware Configuration

3.1 Supported PLCs

The Add-On Instruction can be used with all Rockwell / Allen Bradley PLCs from type CompactLogix / ControlLogix which can be programmed with Studio5000 / RSLogix5000 V16 or higher.



Please note!

The Add-On Instruction is IO-Link Master independent and can be used for all available Masters.

3.2 Studio5000 Configuration

Before the Add-On Instruction can be used, an IO-Link Master device must be configured in the hardware configuration. The IO-Link port used for communication with the RFH5xx must support a process data length of 32byte In/Out.

Figure 2 shows an example projecting of a SICK SIG200 IO-Link Master (EtherNet/IP). The RFH5xx IO-Link device is connected to the first master port.

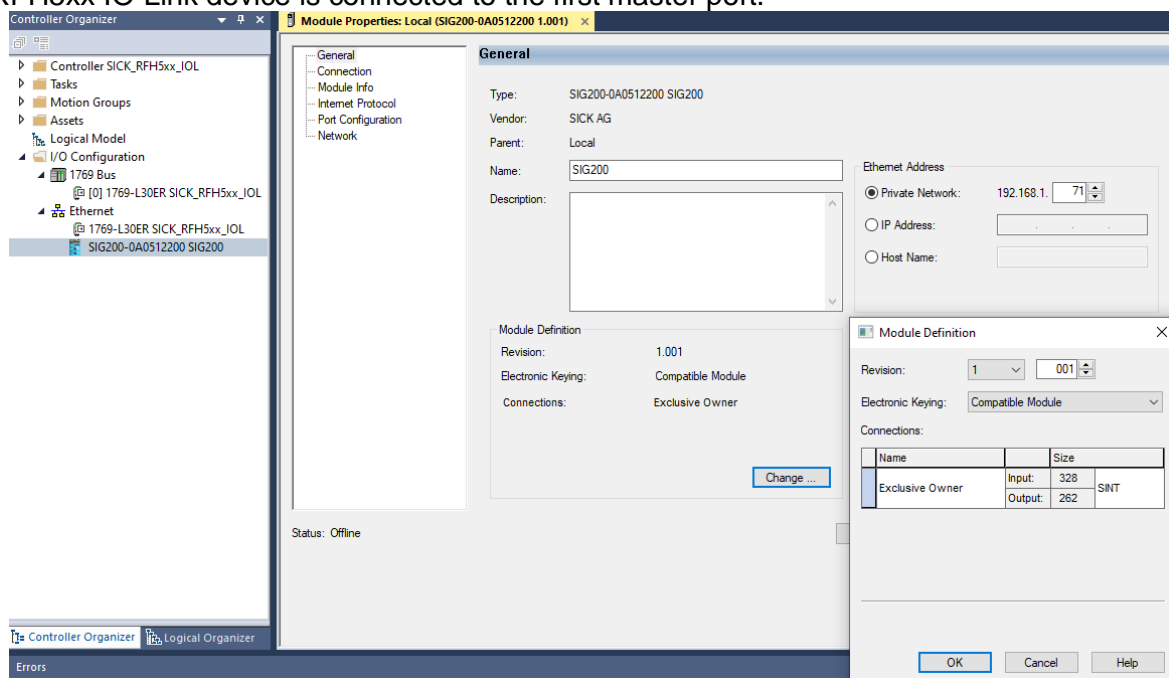


Figure 2: Example hardware configuration

Name	Value	Force Mask	Style	Data Type
▲ SIG200:O	{...}	{...}		_0328:SIG200_0A0512...
▶ SIG200:O.Data	{...}	{...}	Decimal	SINT[262]
▲ SIG200:I	{...}	{...}		_0328:SIG200_0A0512...
SIG200:I.ConnectionFaulted	0		Decimal	BOOL
▶ SIG200:I.Data	{...}	{...}	Decimal	SINT[328]
▶ SIG200:C	{...}	{...}		_0328:SIG200_0A0512...

Figure 3: SIG200 controller tags

4 Add-On Instruction

This Add-On Instruction (AOI) simplifies the usage of a SICK RFH5xx RFID interrogator in combination with a Rockwell ControlLogix / CompactLogix PLC. The AOI uses only the IO-Link process data communication channel for reading or writing RFID transponder data.

The Add-On Instruction works asynchronously, that is, processing requires several block calls. Therefore, it is necessary that the AOI is called cyclically in the user program.

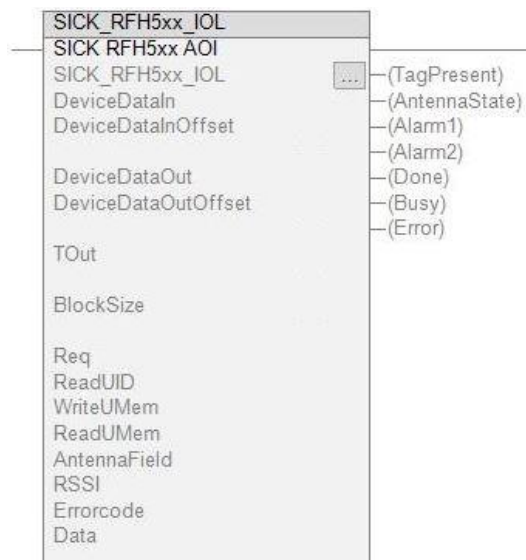


Figure 4: SICK_RFH5xx_IOL Add-On Instruction

4.1 Block specifications

Block name:	SICK_RFH5xx_IOL
Version:	1.1
Used PLC data types:	ST_SICK_RFH5xx └ T_SICK_RFH5xx_ReadUMem └ T_SICK_RFH5xx_WriteUMem └ T_SICK_RFH5xx_CommandQueue ST_SICK_RFH5xx_Error
Block call up:	Cyclically
Used flags:	No
Language:	Structured Text (ST)
Developed with:	Studio5000 V33.00.00

4.2 Process data handover to the AOI

The function block reads or writes the user data of a RFID transponder using the process data. Because the process data structure depends on the selected IO-Link Master, the corresponding array containing the process data of the IO-Link device must be passed to the “DeviceDataIn” / “DeviceDataOut” parameters of the AOI. The parameters “DeviceDataInOffset” / “DeviceDataOutOffset” can be used to specify the byte position in the data array at which the process data of the connected IO-Link device begins (starting with 0).

In this example, the input process data of the RFH51xx device connected to port 1, starts from byte 8. The outgoing process data from byte 6.

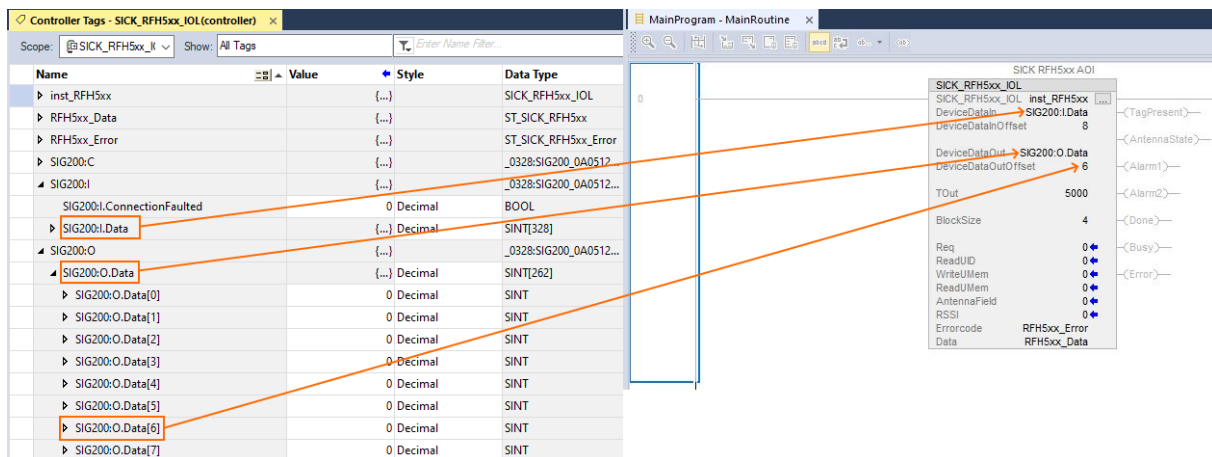


Figure 5: Process data mapping

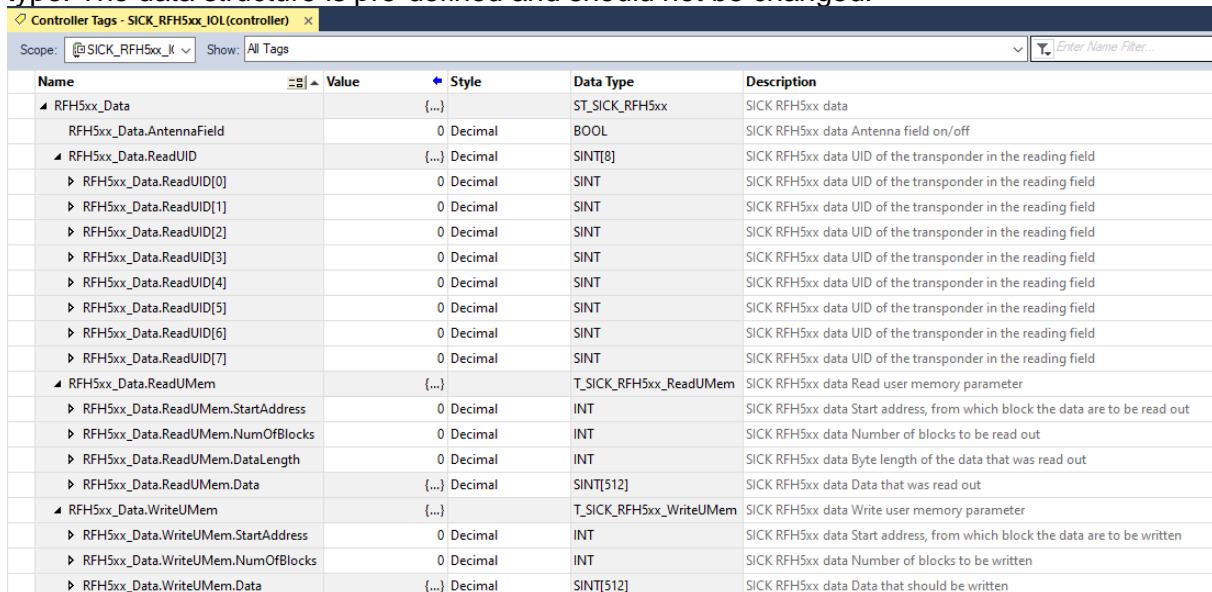
4.3 Operation of the AOI

Each block action (“ReadUMem”, “WriteUMem” etc.) can be parameterized via the data type “ST_SICK_RFH5xx” (Data). In order to execute an AOI action, the desired action has to be selected first. It is also possible to select more than one action. In order to execute the selected action, the parameter “Req” has to be triggered with a positive edge (signal change from a logical zero to one). As long as no valid device answer has to be received, this is signaled via the parameter “Busy”.

If the Add-On Instruction signals “Done = TRUE” at the output parameter, the action has been done successfully. If, for this action (e.g. “ReadUMem”) data has been requested from the device, it will be copied into the respective data area (“Data”).

4.4 Data type description

The data type “ST_SICK_RFH5xx” contains input and output parameters for all supported Add-On Instruction actions. The Add-On Instruction used an instance (variable) of this data type. The data structure is pre-defined and should not be changed.



Name	Value	Style	Data Type	Description
RFH5xx_Data		{...}	ST_SICK_RFH5xx	SICK RFH5xx data
RFH5xx_Data.AntennaField	0	Decimal	BOOL	SICK RFH5xx data Antenna field on/off
RFH5xx_Data.ReadUID		{...}	SINT[8]	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[0]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[1]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[2]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[3]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[4]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[5]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[6]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUID[7]	0	Decimal	SINT	SICK RFH5xx data UID of the transponder in the reading field
RFH5xx_Data.ReadUMem		{...}	T_SICK_RFH5xx_ReadUMem	SICK RFH5xx data Read user memory parameter
RFH5xx_Data.ReadUMem.StartAddress	0	Decimal	INT	SICK RFH5xx data Start address, from which block the data are to be read out
RFH5xx_Data.ReadUMem.NumOfBlocks	0	Decimal	INT	SICK RFH5xx data Number of blocks to be read out
RFH5xx_Data.ReadUMem.DataLength	0	Decimal	INT	SICK RFH5xx data Byte length of the data that was read out
RFH5xx_Data.ReadUMem.Data		{...} Decimal	SINT[512]	SICK RFH5xx data Data that was read out
RFH5xx_Data.WriteUMem		{...}	T_SICK_RFH5xx_WriteUMem	SICK RFH5xx data Write user memory parameter
RFH5xx_Data.WriteUMem.StartAddress	0	Decimal	INT	SICK RFH5xx data Start address, from which block the data are to be written
RFH5xx_Data.WriteUMem.NumOfBlocks	0	Decimal	INT	SICK RFH5xx data Number of blocks to be written
RFH5xx_Data.WriteUMem.Data		{...} Decimal	SINT[512]	SICK RFH5xx data Data that should be written

Figure 6: ST_SICK_RFH5xx data type

4.4.1 AntennaField

This function can be used to switch the RF-Field of the Antenna on or off.

Parameter	Declaration	Data type	Description
AntennaField	Input	BOOL	Antenna power on/off True = On False = Off

Table 1: Parameter of the AntennaField function

4.4.2 ReadUID

When the ReadUID function is executed, the following variable was filled with the transponder identifier (UID).

Parameter	Declaration	Data type	Description
ReadUID	Output	ARRAY [0..7] OF SINT	Transponder UID in a hexadecimal representation.

Table 2: Parameter of the ReadUID function

4.4.3 ReadUMem

Here you can define which area of the RFID tag should be read out.

Parameter	Declaration	Data type	Description
StartAddress	Input	INT	Block number at which the reading should be started. <u>Valid range:</u> [0..255]
NumOfBlocks	Input	INT	Number of blocks that should be read. The valid range depends on the predefined block size (AOI input parameter). <u>Valid range:</u> (BlockSize x NumOfBlocks) <= 512
DataLength	Output	INT	Byte length of the data that was read out
Data	Output	ARRAY [0..511] OF SINT	Data that was read out.

Table 3: Parameter of the ReadUMem function

4.4.4 WriteUMem

Here you can define which area of the RFID tag should be written.

Parameter	Declaration	Data type	Description
StartAddress	Input	INT	Block number at which the writing should be started. <u>Valid range:</u> [0..255]
NumOfBlocks	Input	INT	Number of blocks that should be written. The valid range depends on the predefined block size (AOI input parameter). <u>Valid range:</u> (BlockSize x NumOfBlocks) <= 512
Data	Input	ARRAY [0..511] OF SINT	Data that should be written.

Table 4: Parameter of the WriteUMem function

4.1 Adjust maximum data length

By default, this AOI can read or write a maximum number of 512 bytes of user memory. For applications that require more data, the AOI can be adapted accordingly.

Necessary adaption in the data types:

For the data allocation the UMem "Data" arrays of the data types

"T_SICK_RFH5xx_ReadUMem" and "T_SICK_RFH5xx_WriteUMem" must be adapted accordingly. **Make sure that the value is divisible by 4.**

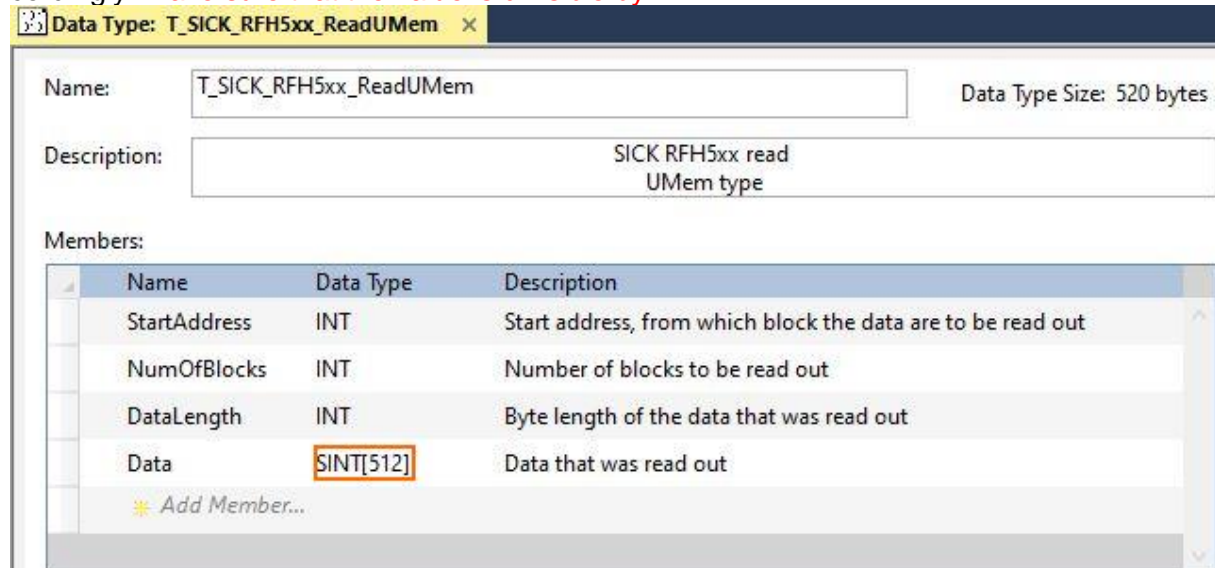


Figure 7: Adaptation in the data type (ReadUMem)

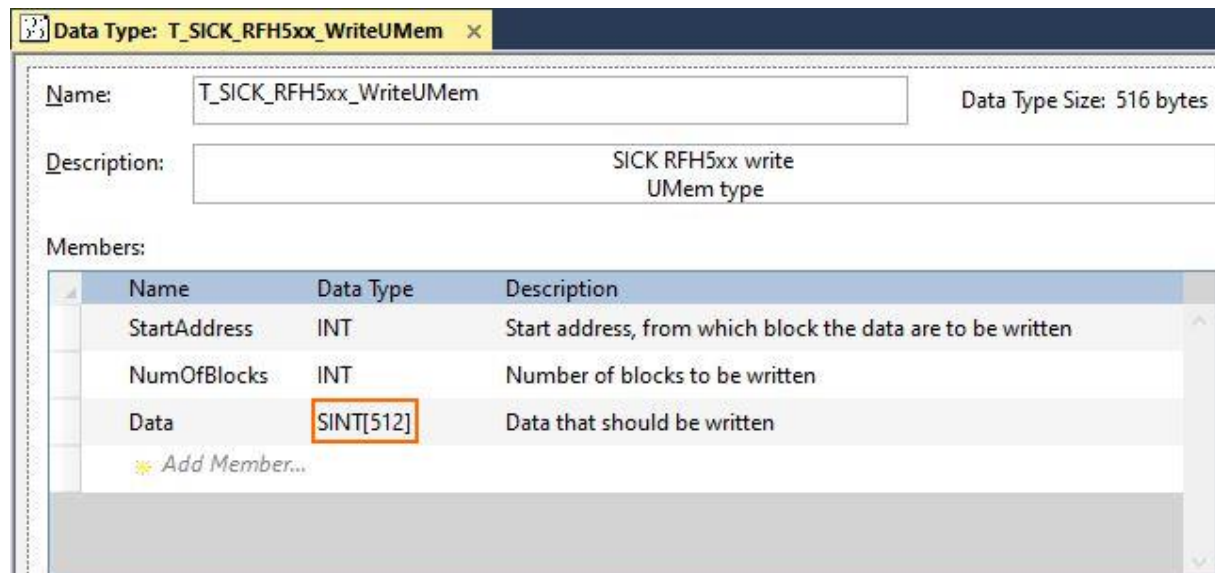


Figure 8: Adaptation in the data type (WriteUMem)

4.2 Behavior when error occurs

If there is a wrong input value of the Add-On Instruction, 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.

5 Parameter

Parameter	Declaration	Data type	Description
DeviceDataIn	In/Out	SINT[1]	Process data input link to the RFH device. See chapter 4.2
DeviceDataIn Offset	Input	INT	Byte offset, from where the port specific process data input starts, beginning with 0. See chapter 4.2
DeviceDataOut	In/Out	SINT[1]	Process data output link of the RFH device. See chapter 4.2
DeviceDataOut Offset	input	INT	Byte offset, from where the port specific process data output starts, beginning with 0. See chapter 4.2
TOut	Input	DINT	Time in [ms] after a timeout error occurs.
BlockSize	Input	SINT	Block size of the transponder you want to use in the application. <u>Valid range:</u> [4,8]
Req	Input	BOOL	A rising edge executes the selected actions.
ReadUID	Input	BOOL	Action: Reading the UID of the transponder in the RF-Field.
WriteUMem	Input	BOOL	Action: Writing data blocks into the user memory of the transponder. Please use the corresponding data structure to define which blocks of the user data should be written.
ReadUMem	Input	BOOL	Action: Reading data blocks from the user memory of the transponder. Please use the corresponding data structure to define which blocks of the user data should be read.
AntennaField	Input	BOOL	Action: Switching the RF-Field of the Antenna on or off. Please use the corresponding data structure to define whether the RF field should be switched on or off.
TagPresent	Output	BOOL	Indicates if there is a tag in the RF field of the RFH5xx. This flag is updated cyclically.
AntennaState	Output	BOOL	Indicates the state of the antenna power. This flag is updated cyclically.
RSSI	Output	SINT	RSSI signal level coming from the transponder. This value is updated cyclically
Alarm1	Output	BOOL	Status of RFH Alarm1. This flag is updated cyclically.

Parameter	Declaration	Data type	Description
Alarm2	Output	BOOL	Status of RFH Alarm2. This flag is updated cyclically.
Done	Output	BOOL	Indicates that the selected Add-On Instruction action has been performed without errors.
Busy	Output	BOOL	Request in process FALSE: Request is terminated TRUE: Request is being processed
Error	Output	BOOL	Error occurred. FALSE: No error TRUE: Error detected
Errorcode	In/Out	ST_SICK_ RFH5xx_ Error	Error information (see error code description)
Data	In/Out	ST_SICK_ RFH5xx	Contains input and output parameters for all supported Add-On Instruction actions.

Table 5: Add-On Instruction parameters

6 Error description

The parameter "Errorcode" contains the following error information:

- Block specific error code
- Device error code

Block Errorcode	Description																									
16#0000	No error																									
16#0001	Timeout error occurs. The processing of the actions takes longer than the time set at the "TOut" parameter.																									
16#0002	No AOI action selected. A request (Req) was executed without selecting an action.																									
16#0003	The defined block size is not supported by the Add-On Instruction. Valid values: [4, 8]																									
16#0004	It's not possible to read or write more than 512 byte or less than 0 bytes of the user memory. Please adjust the number of blocks to be read or written.																									
16#0005	No transponder present in the RF-Field of RFH5xx.																									
16#0006	The start address of the requested command does not match with the response.																									
16#0007	Invalid value for parameter StartAddress (read or write UMem) (255 < StartAddress < 0)																									
16#0008 - 16#000B	Reserved																									
16#000C	Invalid value for the parameter DeviceDataOutOffset (255 < value < 0)																									
16#000D	Invalid value for the parameter DeviceDataInOffset (value < 0)																									
16#000E	The array, which is specified by the pointer DeviceDataOut and the offset DeviceDataOutOffset, must be at least 32 bytes long.																									
16#000F	The array, which is specified by the pointer DeviceDataIn and the offset DeviceDataInOffset, must be at least 32 bytes long.																									
16#0010	Device error detected The variable "DeviceErrorcode" contains the device error code. <table><tr><th>Error Code</th><th>Name</th><th>Description</th></tr><tr><td>1</td><td>CommandNotSupported</td><td rowspan="7">Error code values replied by the transponder to the RWM interrogation. Depend of ISO15693 command set supported by the different transponder IC of the market. These are error code values defined by the IOS15693 standard.</td></tr><tr><td>2</td><td>FormatError</td></tr><tr><td>3</td><td>OptionNotSupported</td></tr><tr><td>5</td><td>CommandProblem</td></tr><tr><td>6</td><td>CommTagError</td></tr><tr><td>15</td><td>TagError</td></tr><tr><td>16</td><td>NoMemoryBlock</td></tr><tr><td>18</td><td>BlockProtected</td><td rowspan="3">Indicates a transponder communication error (e.g. more than 1 transponder detected or transponder reply not understood)</td></tr><tr><td>30</td><td>TAGCommError</td></tr><tr><td>255</td><td>AppGeneralError</td></tr></table>	Error Code	Name	Description	1	CommandNotSupported	Error code values replied by the transponder to the RWM interrogation. Depend of ISO15693 command set supported by the different transponder IC of the market. These are error code values defined by the IOS15693 standard.	2	FormatError	3	OptionNotSupported	5	CommandProblem	6	CommTagError	15	TagError	16	NoMemoryBlock	18	BlockProtected	Indicates a transponder communication error (e.g. more than 1 transponder detected or transponder reply not understood)	30	TAGCommError	255	AppGeneralError
Error Code	Name	Description																								
1	CommandNotSupported	Error code values replied by the transponder to the RWM interrogation. Depend of ISO15693 command set supported by the different transponder IC of the market. These are error code values defined by the IOS15693 standard.																								
2	FormatError																									
3	OptionNotSupported																									
5	CommandProblem																									
6	CommTagError																									
15	TagError																									
16	NoMemoryBlock																									
18	BlockProtected	Indicates a transponder communication error (e.g. more than 1 transponder detected or transponder reply not understood)																								
30	TAGCommError																									
255	AppGeneralError																									

Table 6: Add-On Instruction error codes

7 Example

The example enables you to write the current time into an accessible RFID transponder using the SICK RFH5xx Add-On Instruction. The RFH5xx RFID interrogator is connected to a SIG200 IO-Link Master. The RFID tag will be a NXP with 112Byte user data and a block size of 4 byte.

7.1 Implementation

Please use the Controller Tag to control this program.

7.1.1 Initialization

First, all selection bits are reset.



Figure 9: Initialization

7.1.2 Read tag parameterization

Definition of the tag data that should be read.

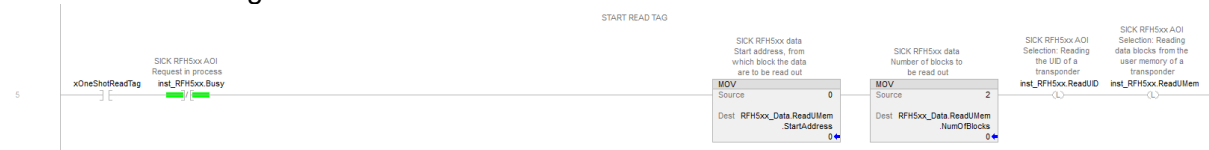


Figure 10: Start read tag

7.1.3 Write tag parameterization

Definition of the tag data that should be written. Here the actual system time is read out and converted into a string.

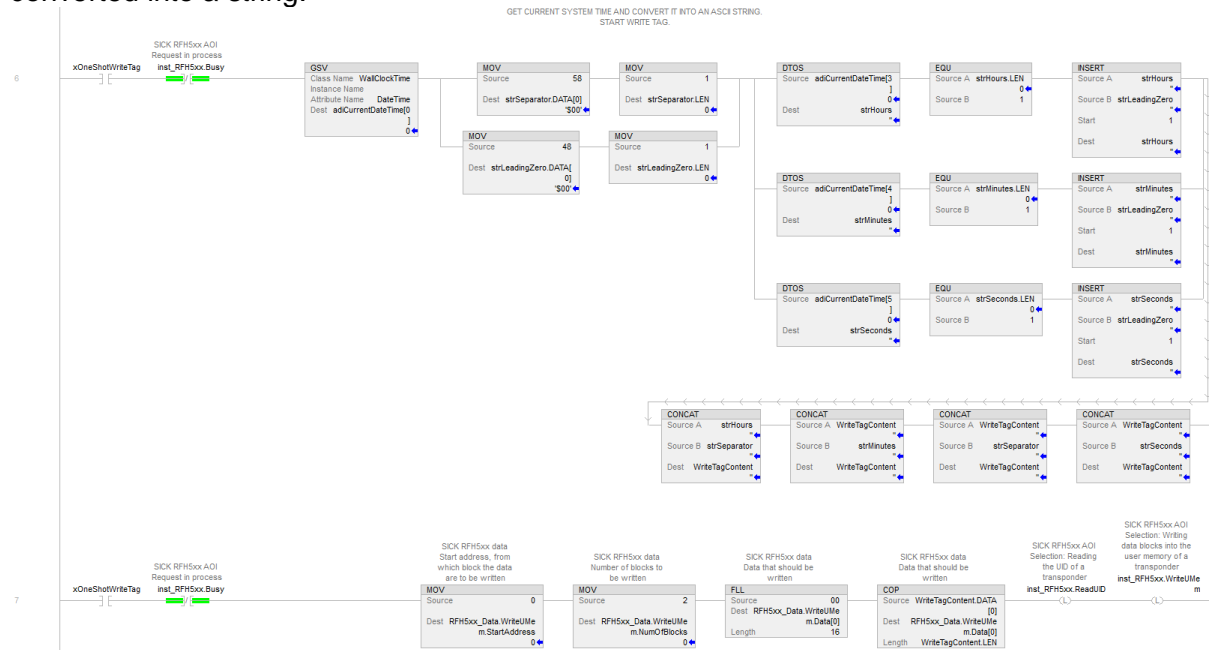


Figure 11: Start write tag

7.1.4 Start read / write request



Figure 12: Start read / write request

7.1.5 Handle response data

If data has been read successfully, it is converted into a string variable.



Figure 13: Handle response data

7.1.6 Add-On Instruction call up

Call up of the RFH5xx Add-On Instruction.



Figure 14: Call RFH AOI

7.2 Writing user data

Set the "WriteTag" variable to write the actual PLC time into the current transponder. The "WriteDone" flag indicates a successful execution of the function.

Name	Scope	Value	
WriteTag	MainProgram	1	Triggers the writing process
ReadTag	MainProgram	0	
WriteTagDone	MainProgram	1	Indicates that the write process has been completed successfully
ReadTagDone	MainProgram	0	
ReadTagContent	MainProgram	"	
WriteTagContent	MainProgram	'08:44:10'	Data written to the user memory
RFH5xx_Data.ReadUID	MainProgram	{...}	
RFH5xx_Data.ReadUID[0]	MainProgram	16#e0	
RFH5xx_Data.ReadUID[1]	MainProgram	16#04	
RFH5xx_Data.ReadUID[2]	MainProgram	16#01	
RFH5xx_Data.ReadUID[3]	MainProgram	16#00	
RFH5xx_Data.ReadUID[4]	MainProgram	16#08	
RFH5xx_Data.ReadUID[5]	MainProgram	16#16	
RFH5xx_Data.ReadUID[6]	MainProgram	16#57	
RFH5xx_Data.ReadUID[7]	MainProgram	16#6e	UID of the used transponder
inst_RFH5xx.AntennaState	MainProgram	1	
inst_RFH5xx.TagPresent	MainProgram	1	
inst_RFH5xx.RSSI	MainProgram	6	
inst_RFH5xx.Alarm1	MainProgram	0	
inst_RFH5xx.Alarm2	MainProgram	0	

Figure 15: Trigger write process

7.3 Reading user data

Set the "xReadTag" variable to read out the first 32Byte of the current transponder. The "xReadDone" flag indicates a successful execution of the function. The "sReadTagContent" string contains the tag data.

Name	Scope	Value	
WriteTag	MainProgram	0	
ReadTag	MainProgram	1	Triggers the reading process
WriteTagDone	MainProgram	0	
ReadTagDone	MainProgram	1	Indicates that the read process has been completed successfully
ReadTagContent	MainProgram	'08:44:10'	Read tag content
WriteTagContent	MainProgram	'08:44:10'	
RFH5xx_Data.ReadUID	MainProgram	{...}	
RFH5xx_Data.ReadUID[0]	MainProgram	16#e0	
RFH5xx_Data.ReadUID[1]	MainProgram	16#04	
RFH5xx_Data.ReadUID[2]	MainProgram	16#01	
RFH5xx_Data.ReadUID[3]	MainProgram	16#00	
RFH5xx_Data.ReadUID[4]	MainProgram	16#08	
RFH5xx_Data.ReadUID[5]	MainProgram	16#16	
RFH5xx_Data.ReadUID[6]	MainProgram	16#57	
RFH5xx_Data.ReadUID[7]	MainProgram	16#6e	UID of the used transponder
inst_RFH5xx.AntennaState	MainProgram	1	
inst_RFH5xx.TagPresent	MainProgram	1	
inst_RFH5xx.RSSI	MainProgram	6	
inst_RFH5xx.Alarm1	MainProgram	0	
inst_RFH5xx.Alarm2	MainProgram	0	

Figure 16: Trigger read process