BALLUFF

sensors worldwide

BMC_AOI_PROC_BISM045
AOI_BISM_32Byte_IOL_040

Add-On Instruction User Guide

May 18, 2016

Balluff Inc.
8125 Holton Drive
Florence, KY 41042
1-800-543-8390

Table of Contents

1.0 Scope

1.1 This User Guide describes the installation, use and maintenance of the Add-On Instruction (AOI) software module for the BIS M-xxx-045-xxx-07-S4 and BIS M-XXX-072-XXX-07-S4. This software module is designed for use with RSLogix5000, version 18.01 or later.

2.0 Products

2.1 This guide was developed for use with the software module AOI_BISM_32Byte_IOL_040.L5X and BMC_AOI_PROC_BISM045.L5X. These software modules are available for download at the web site www.Balluff.com/AOI.

2.2 This software module was developed for use with Balluff IO-Link devices BIS M-XXX-045-XXX-07-S4 and BIS M-XXX-072-XXX-07-S4.

2.3 For the development of this software module and hardware configuration, the following system components were used and validated:

2.3.1 RSLogix5000, version 18.01 (or later)

2.3.2 Rockwell 1756-L63 controller

3.0 Instructions

3.1 This section describes the process for installing and using the AOI module. This is a three step process that must be followed sequentially: hardware configuration; import AOI module along with all associated UDTs; and create new ladder logic with AOI modules.

3.2 Hardware Configuration

3.2.1 The BIS M-XXX-045 or -072-XXX-07-S4 does not have any configuration setting in the RSLogix5000 I/O Configuration Tree. All of its data must be passed through an IO-Link master that communicates with the PLC using EtherNet/IP. To understand how to configure an IO-Link master, please refer to the AOI_BNI004A_40_27_041 User Guide (for use with the Balluff BNI EIP-502-105-Z015).

3.2.2 When the IO-Link master has established a connection with the PLC using EtherNet/IP, the IO-Link port connected to the BIS M-XXX-045 or -072-XXX-07-S4 must be configured for IO-Link mode as described in the AOI_BNI004A_40_27_041 User Guide.

3.2.3 The BIS00xx is a configurable IO-Link device, therefore, certain parameters can be changed from the default setting. This is done by using an Explicit message to the IO-Link Master that will write to the parameters as defined in the BIS manual. In order to properly operate this AOI, at least one parameter must be changed. The parameter that must be changed from the default setting is "Action if tag present" (SPDU Index = 40h, Subindex = 3h). This section (3.2.3) details the four steps necessary to complete this process for changing the "Action if tag present" parameter:

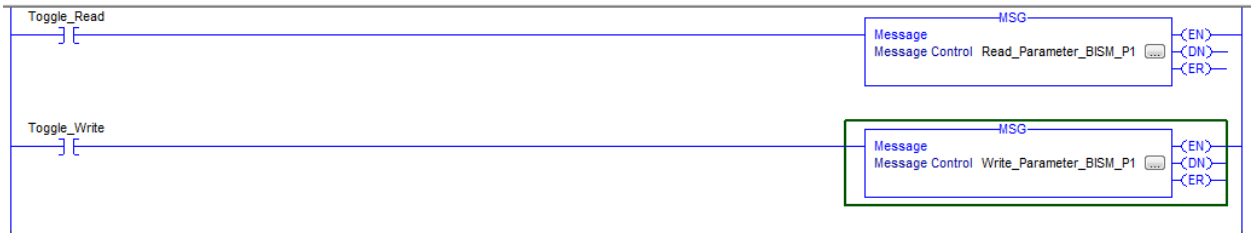
1) Understand the configuration environment (Section 3.2.3.2)
2) Create an Explicit Read instruction (Section 3.2.3.3)
3) Create an Explicit Write instruction (Section 3.2.3.4)
4) Using the Explicit instructions (Section 3.2.3.5)

3.2.3.1 Either logic must be included in the RSLogix project outside of the AOI that will ensure a codetag is present prior to sending a Read or Write command OR the BIS must be changed to Dynamic Mode using an Explicit instruction. If this, or any other parameters are changed based on the needs of the application, the AOI logic may need to be modified to ensure proper operation. In order to change any parameters from the factory default setting, an explicit message must be sent to the BIS.

3.2.3.2 The following diagram from the BIS manual shows the values required in this explicit message to configure RFID processor for automatic reading of 8 bytes:

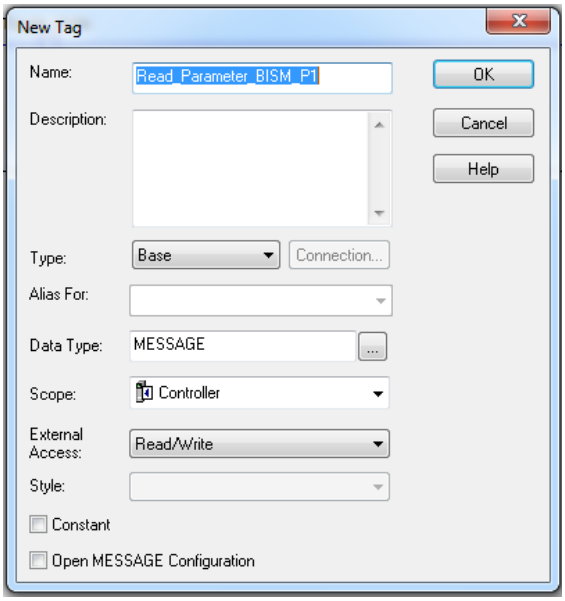
	Access SPDU		Description	Data width	Value range	Factory setting
	Index	Subindex				
terization data	40hex	1hex	CRC yes/no	1 byte	0 = without CRC 1 = with CRC	0
	40hex	2hex	Dynamic mode - yes/no	1 byte	0 = no 1 = yes	0
	40hex	3hex	Action if tag present	1 byte	0 = no action 1 = serial number and tag type 7 = automatically read 8 bytes of data beginning at a set start address after subindex 4 and 5	1

i. To execute an explicit message (MSG) instruction, the rung must go from a low to a high transition. Insert XIC instructions preceding each message instruction to accomplish this. These tags could be tied to momentary push buttons on an HMI as an example.

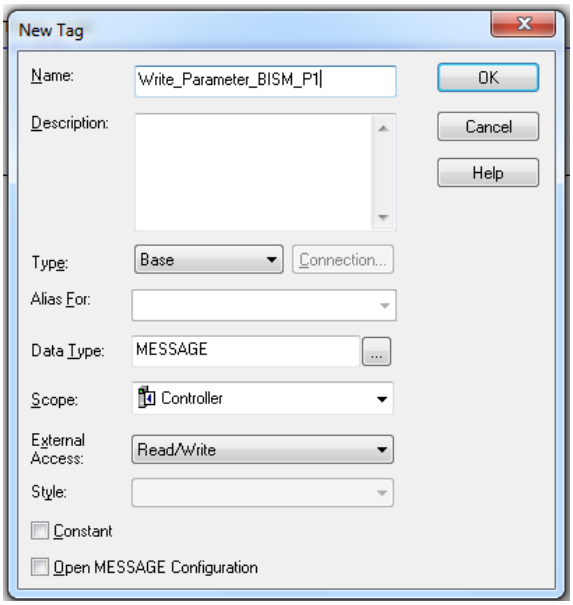


Note that in this example, two new tags have been created, both of data type “Message”:

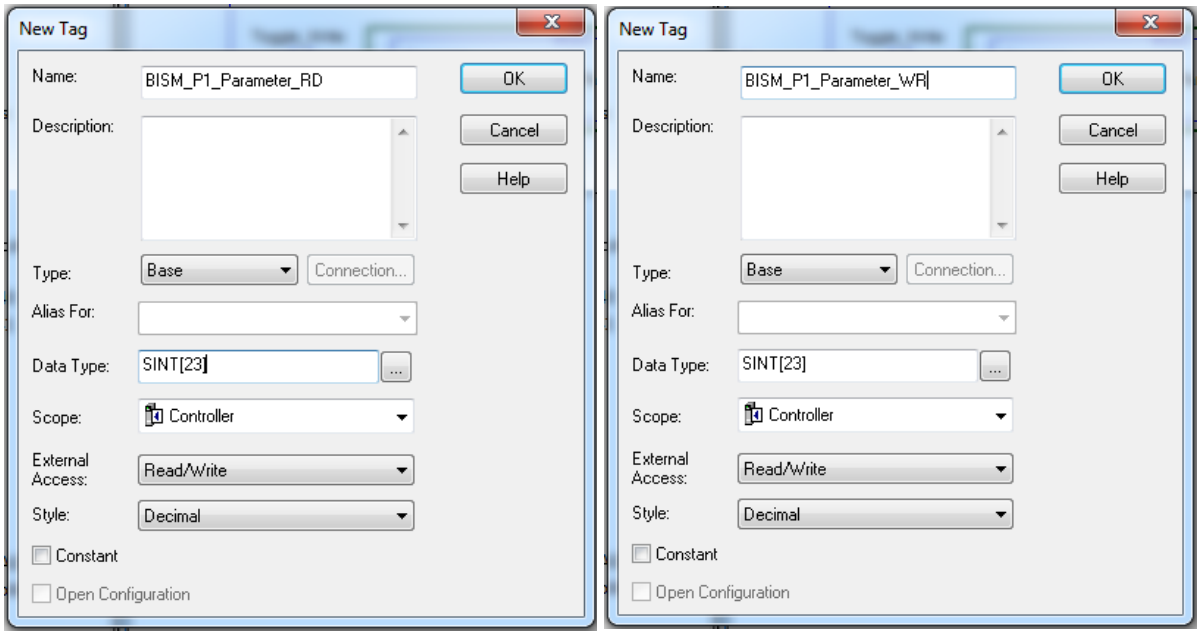
Read_Parameter_BISM_P1



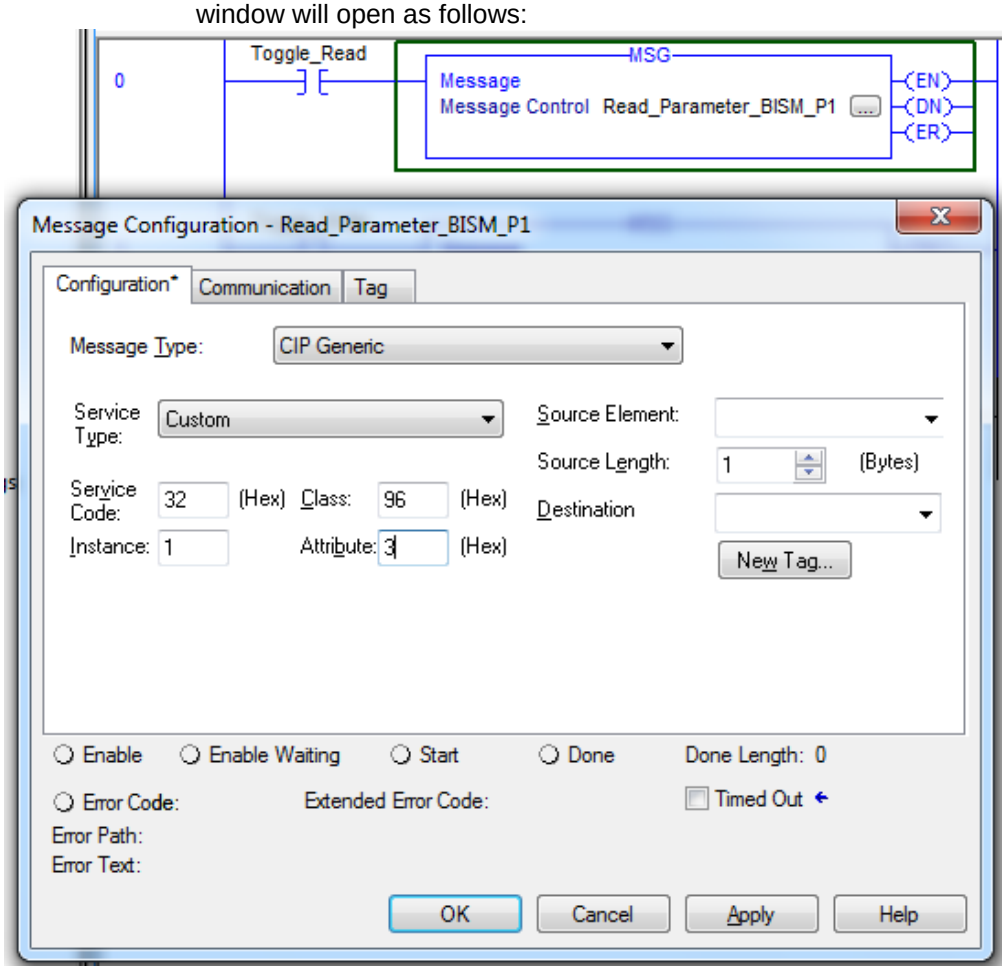
Write_Parameter_BISM_P1



3.2.3.3 To configure the Read_Parameter_BISM_P1 MSG instruction, two new tags must be created as follows:



i. First, click on the radio button on the MSG instruction. A Message Configuration dialog window will open as follows:



Message Type: CIP Generic

Service Type: Custom

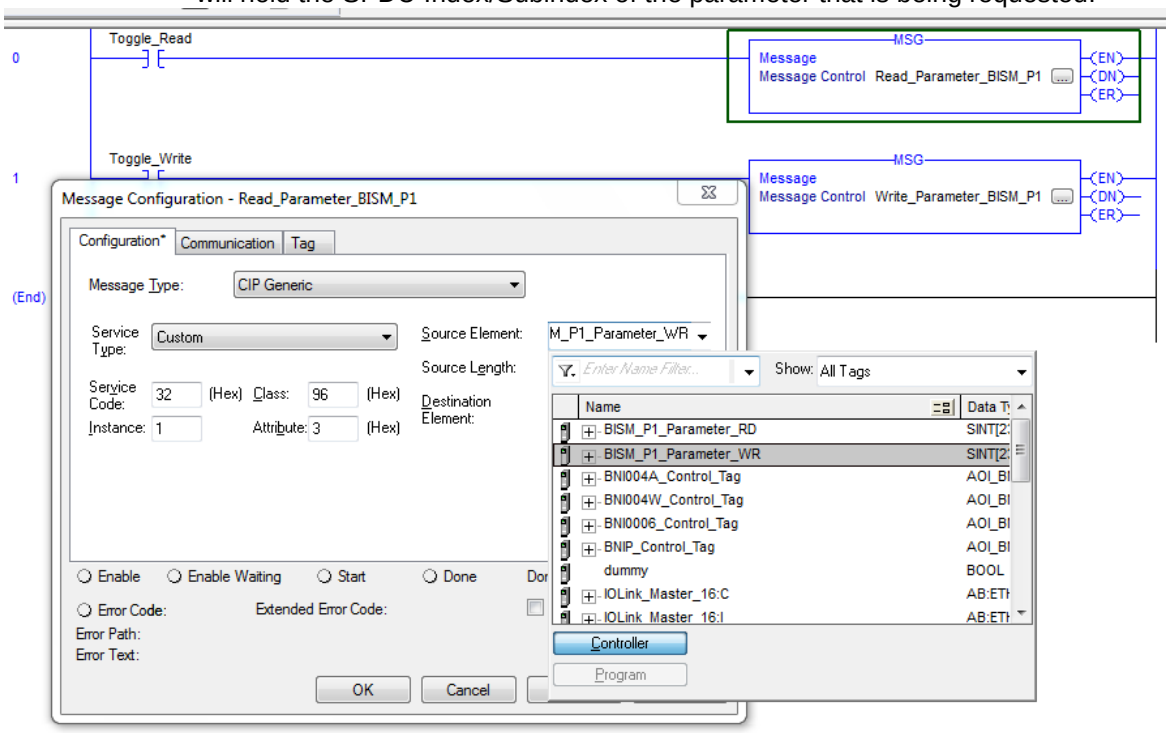
Service Code: 32 (50 Decimal)

Class: 96 (150 Decimal)

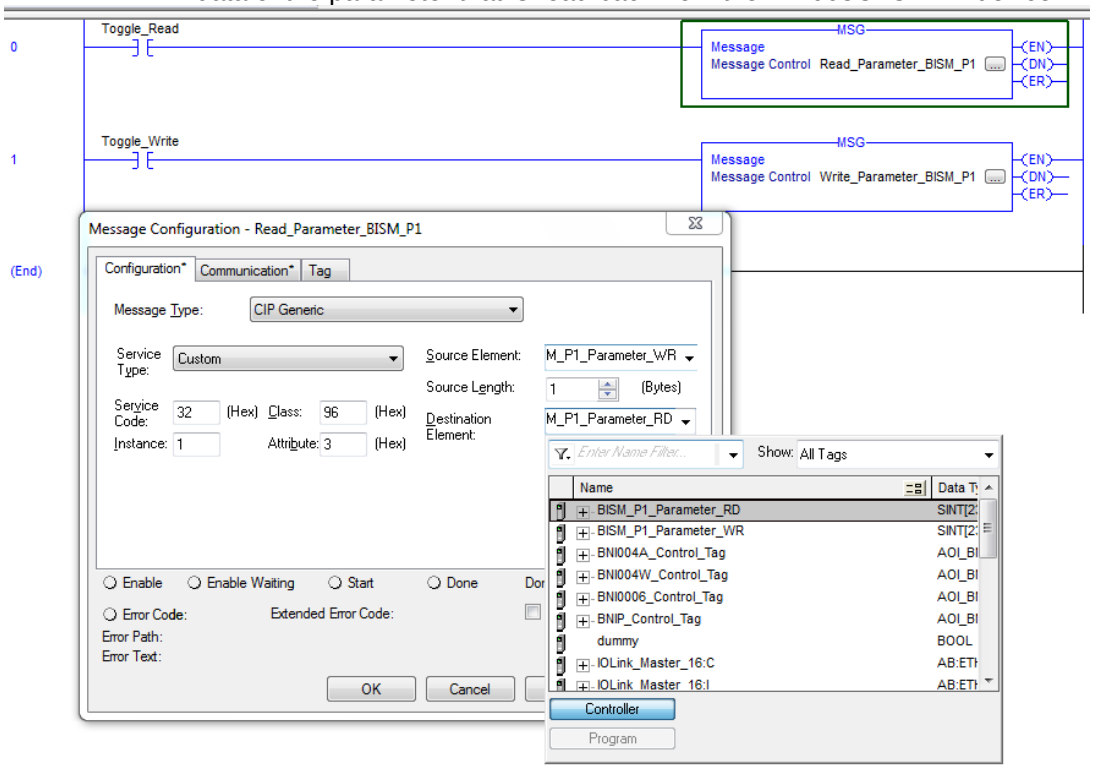
Instance: IO-Link Port Number device is connected to (Port 1 for this example)

Attribute: 3 for read, 2 for write (we are writing parameters with this MSG instruction)

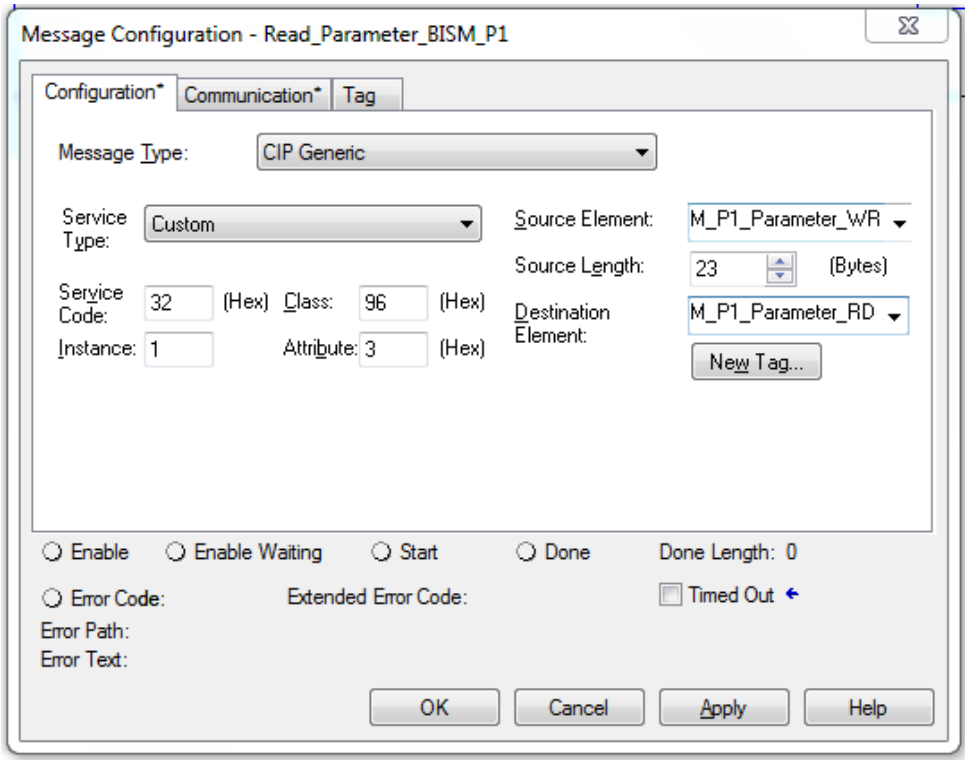
ii. For the Source Element, tag BNI003C_P1_Parameter_WR must be selected. This tag will hold the SPDU Index/Subindex of the parameter that is being requested.



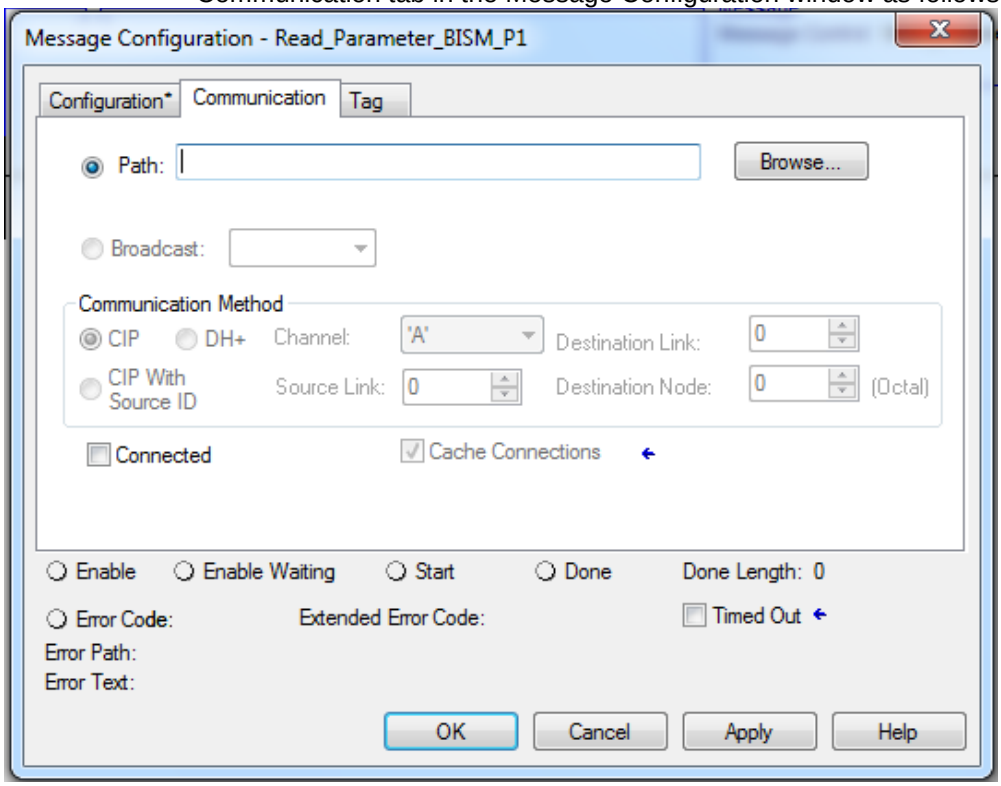
iii. For the Destination, tag BNI003C_P1_Parameter_RD must be selected. This tag will hold data of the parameter that is read back from the BNI003C IO-Link device.



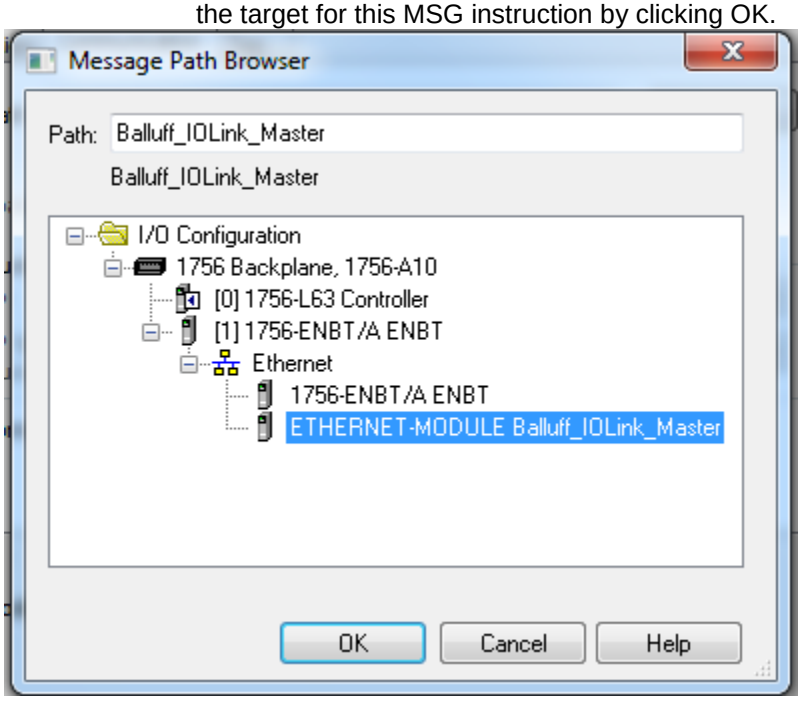
iv. The Source length will be the maximum length of the parameter that you expect to read. In this example, it is 23 bytes.



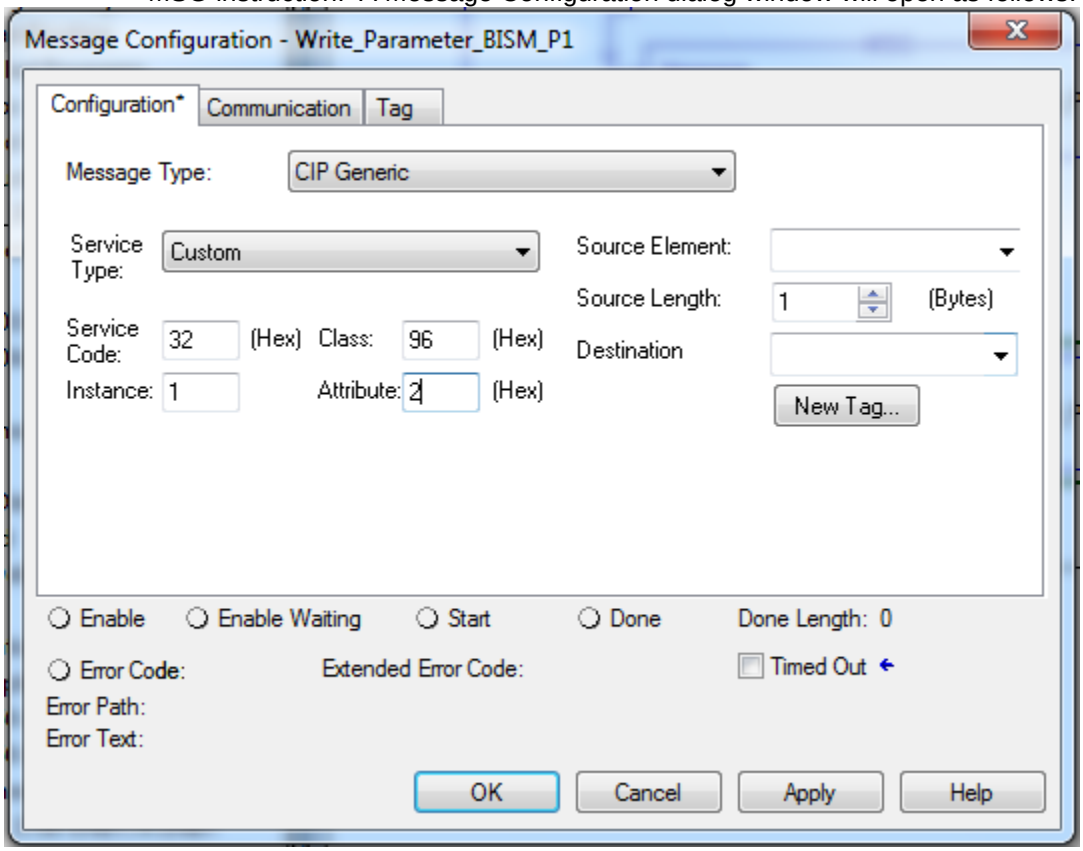
v. Set the communication path for the MSG instruction. This is done by clicking on the Communication tab in the Message Configuration window as follows:



vi. Next click on Browse and select the BNI EIP-502-105-Z015 IO-Link master block that is the target for this MSG instruction by clicking OK.



3.2.3.4 To configure the Write_Parameter_BISM_P1 MSG instruction, click on the radio button on the MSG instruction. A Message Configuration dialog window will open as follows:



Message Type: CIP Generic

Service Type: Custom

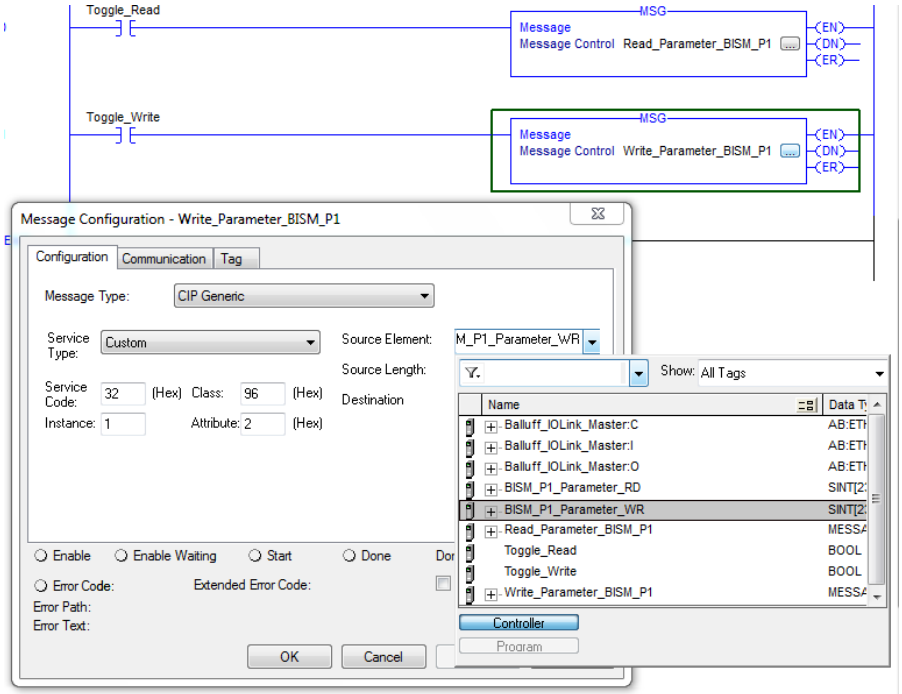
Service Code: 32 (50 Decimal)

Class: 96 (150 Decimal)

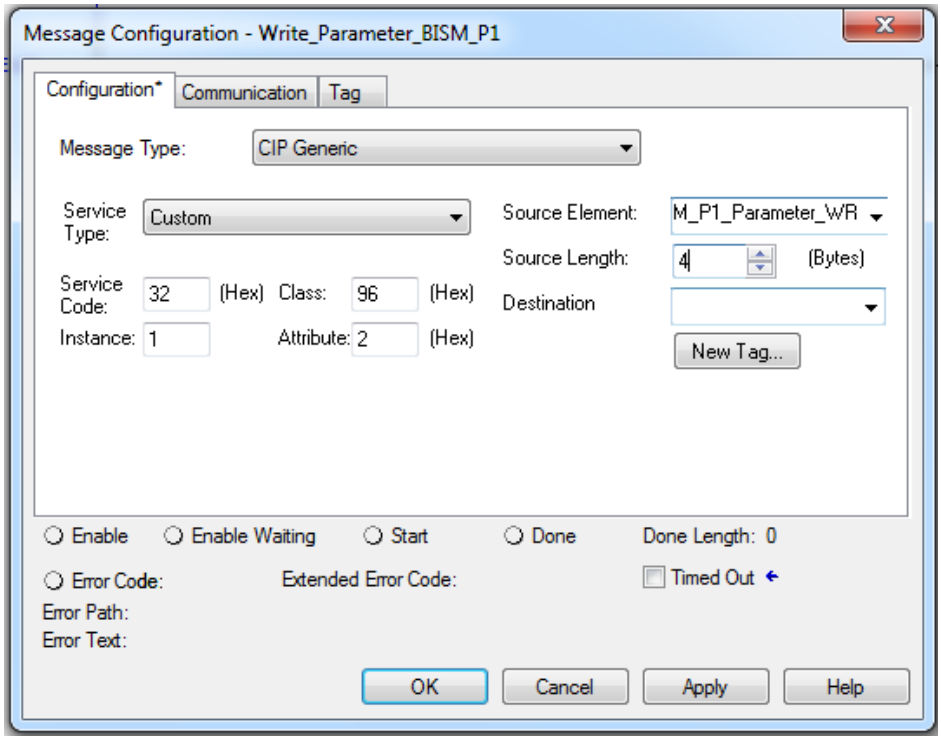
Instance: IO-Link Port Number device is connected to (Port 1 for this example)

Attribute: 3 for read, 2 for write (we are writing parameters with this MSG instruction)

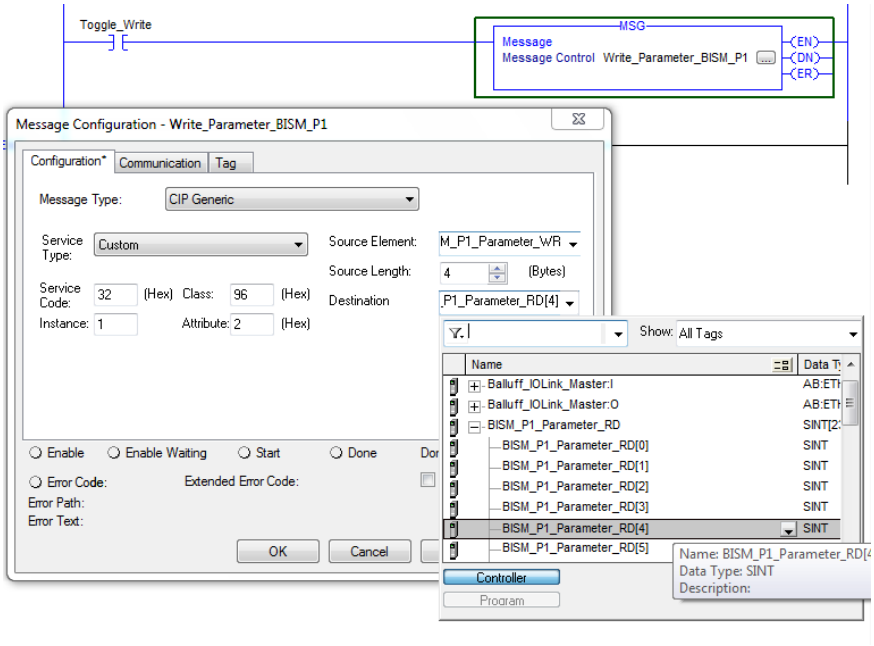
i. For the Source Element, tag BISM_P1_Parameter_WR must be selected. This tag will hold the SPDU Index/Subindex and values of the parameter that is being set.



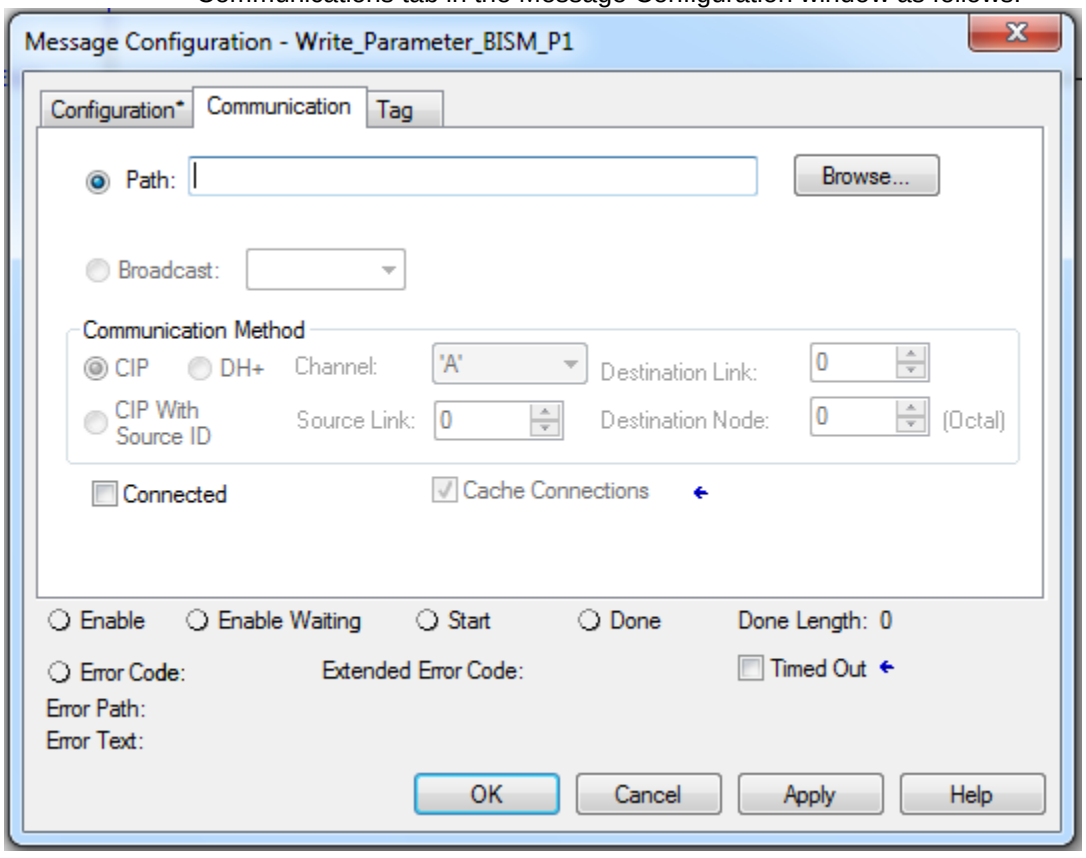
ii. The Source length will depend on the maximum length of the parameter value that is to be set. 3 Bytes will always be required for the SPDU Index Low Byte, Index High Byte and Sub-Index. Since we are configuring the BIS using the “Action if tag present” parameter, a value of 1 byte will be required as well. Therefore, a total of 4 bytes will be entered for the Source Length:



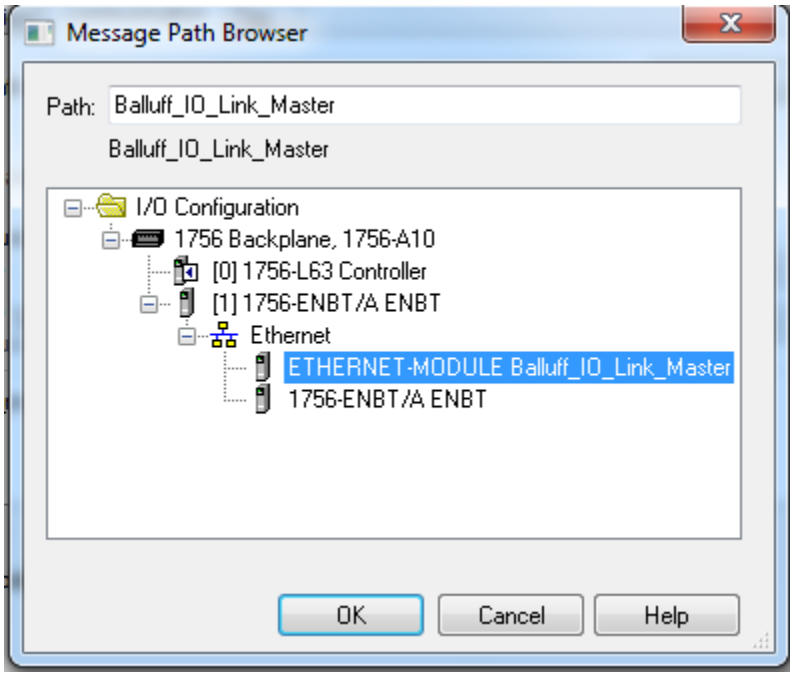
iii. For the Destination, tag BISM_P1_Parameter_WR must be selected. Since we are using 4 bytes of this tag as specified in the Source Length above, the tag selected will be BISM_P1_Parameter_WR[4].



iv. Set the communication path for the MSG instruction. This is done by clicking on the Communications tab in the Message Configuration window as follows:



v. Next, click on Browse and select the BNI EIP-502-105-Z015 that is the target for this MSG instruction by clicking OK:



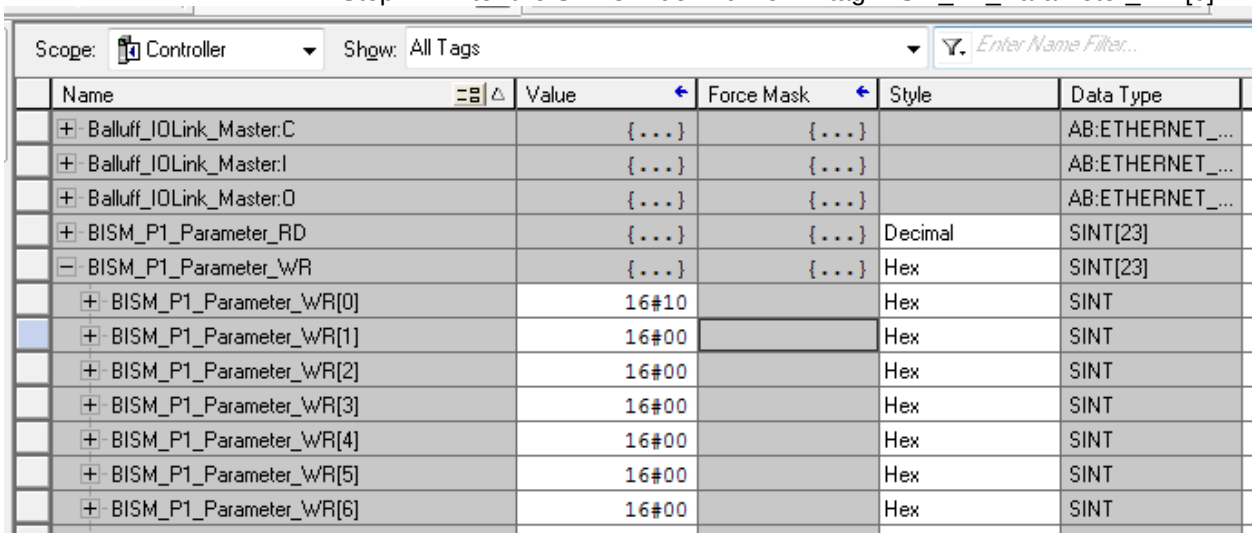
3.2.3.5 Using the MSG instructions: before this part is carried out, the project must be downloaded and the controller must be in run mode.

i. Reading Parameters from an IO-Link device: This example will demonstrate how to read a parameter from the BIS M IO-Link device. For this example, the Vendor Name will be read from the BIS M processor. The Parameter Data for the BIS M processor shows the following:

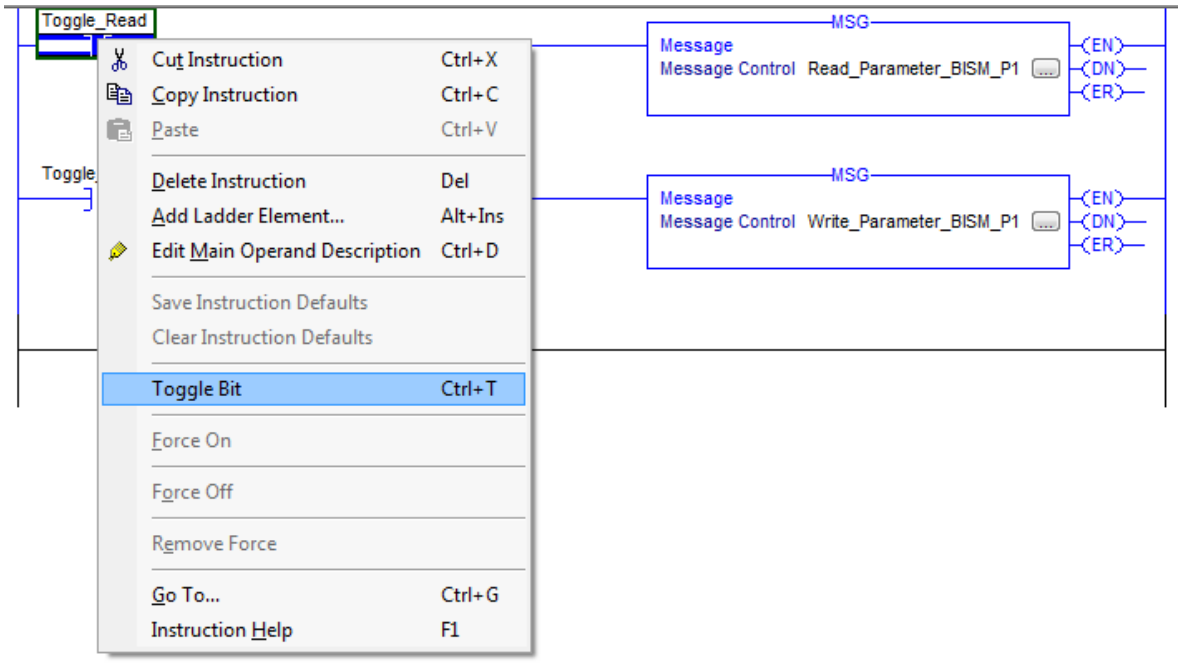
SPDU Index: 0x10 (10 Hex) – This is the Low Byte of the Index, there is no High Byte.
SPDU Sub-Index: 0

Using these values, the controller should read back the value “BALLUFF” and should be located in tag BISM_P1_Parameter_RD.

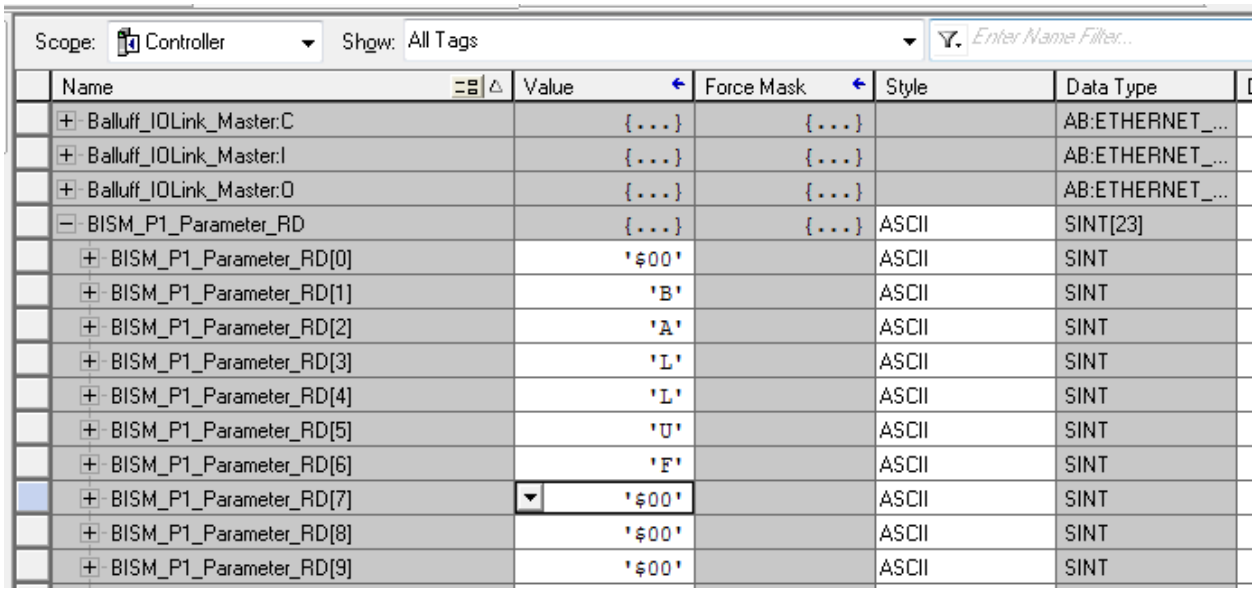
ii. Step 1: Enter the SPDU Index 10 Hex in tag BISM_P1_Parameter_WR[0]:



iii. Step 2: Toggle MSG Instruction Read_Parameters_BISM_P1



iv. Step 3: Read the parameter data in tag BISM_P1_Parameter_RD starting at Byte 0.



Scope: Controller Show: All Tags Enter Name Filter...

Name	Value	Force Mask	Style	Data Type
Balluff_IOLink_Master:C	{...}	{...}		AB:ETHERNET_...
Balluff_IOLink_Master:I	{...}	{...}		AB:ETHERNET_...
Balluff_IOLink_Master:O	{...}	{...}		AB:ETHERNET_...
BISM_P1_Parameter_RD	{...}	{...}	ASCII	SINT[23]
BISM_P1_Parameter_RD[0]	'$00'		ASCII	SINT
BISM_P1_Parameter_RD[1]	'B'		ASCII	SINT
BISM_P1_Parameter_RD[2]	'A'		ASCII	SINT
BISM_P1_Parameter_RD[3]	'L'		ASCII	SINT
BISM_P1_Parameter_RD[4]	'L'		ASCII	SINT
BISM_P1_Parameter_RD[5]	'U'		ASCII	SINT
BISM_P1_Parameter_RD[6]	'F'		ASCII	SINT
BISM_P1_Parameter_RD[7]	'$00'		ASCII	SINT
BISM_P1_Parameter_RD[8]	'$00'		ASCII	SINT
BISM_P1_Parameter_RD[9]	'$00'		ASCII	SINT

Note that BISM_P1_Parameter_RD[0] is a feedback byte and should be set equal to 0 if the Read command is successful. This shifts the data that is read by one byte.

v. Writing parameters to an IO-Link device. Next we will demonstrate changing "Action if tag present" from the Factory setting to "automatically read 8 bytes of data beginning at a set start address after subindex 4 and 5". Here are the parameter settings for this parameter in the BIS M:

	Access SPDU		Description	Data width	Value range	Factory setting
	Index	Subindex				
terization data	40hex	1hex	CRC yes/no	1 byte	0 = without CRC 1 = with CRC	0
	40hex	2hex	Dynamic mode - yes/no	1 byte	0 = no 1 = yes	0
	40hex	3hex	Action if tag present	1 byte	0 = no action 1 = serial number and tag type 7 = automatically read 8 bytes of data beginning at a set start address after subindex 4 and 5	1

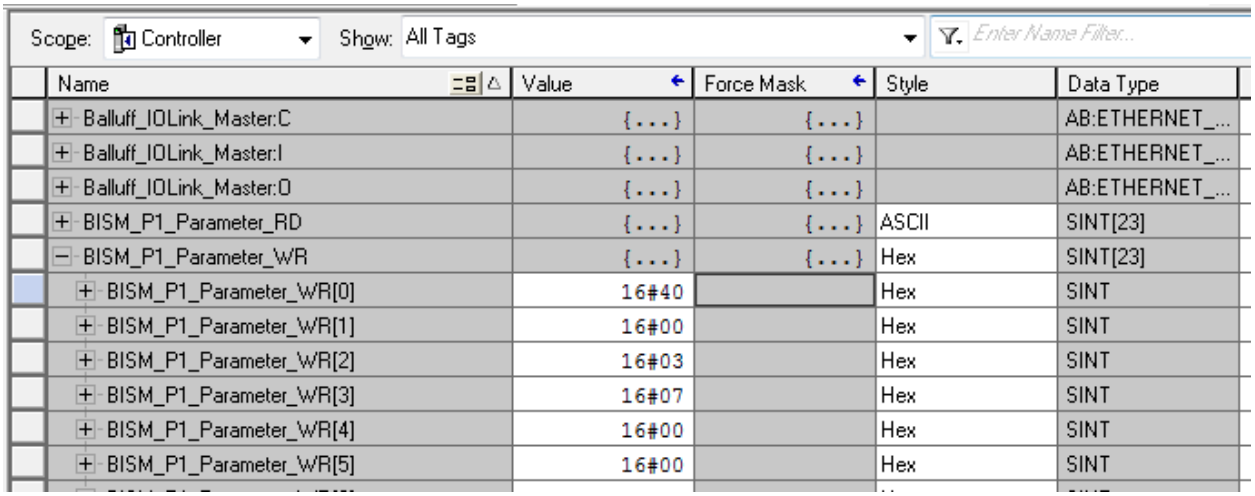
To change this feature, the parameter value should be set to 7hex.

As presented earlier, the value to set this parameter is as follows:

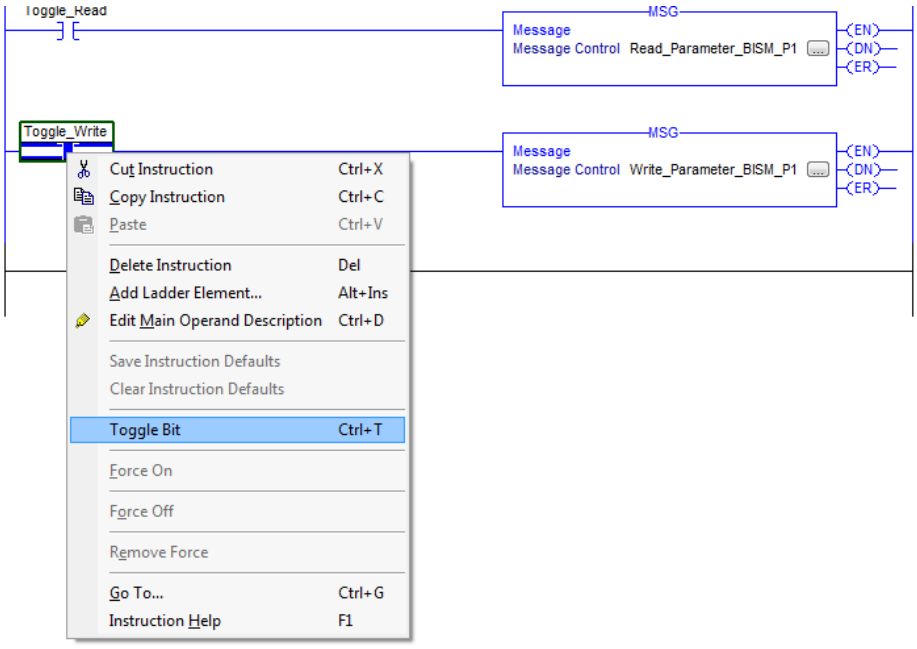
SPDU Index: 0x40 (40 Hex) – this is the Low Byte of the Index, there is no High Byte
SPDU Sub-Index: 3hex

vi. Step 4:

Enter the SPDU Index 40 Hex in tag BISM_P1_Parameter_WR[0]
Enter the SPDU Index 0 Hex in tag BISM_P1_Parameter_WR[1]
Enter the SPDU Sub-Index 3 Hex in tag BISM_P1_Parameter_WR[2]
Enter the Parameter value of 7 Hex in tag BISM_P1_Parameter_WR[3]



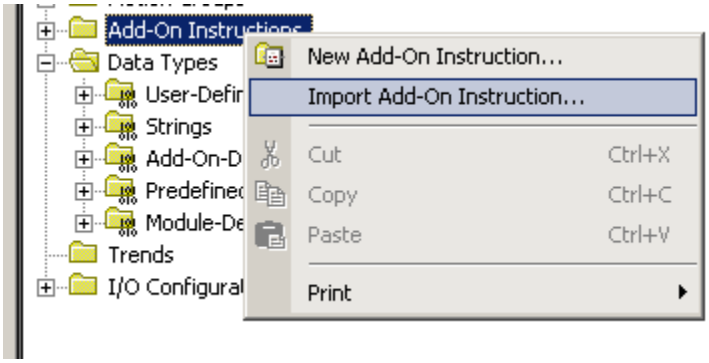
vii. Step 5: Toggle MSG Instruction Write_Parameters_BISM_P1.



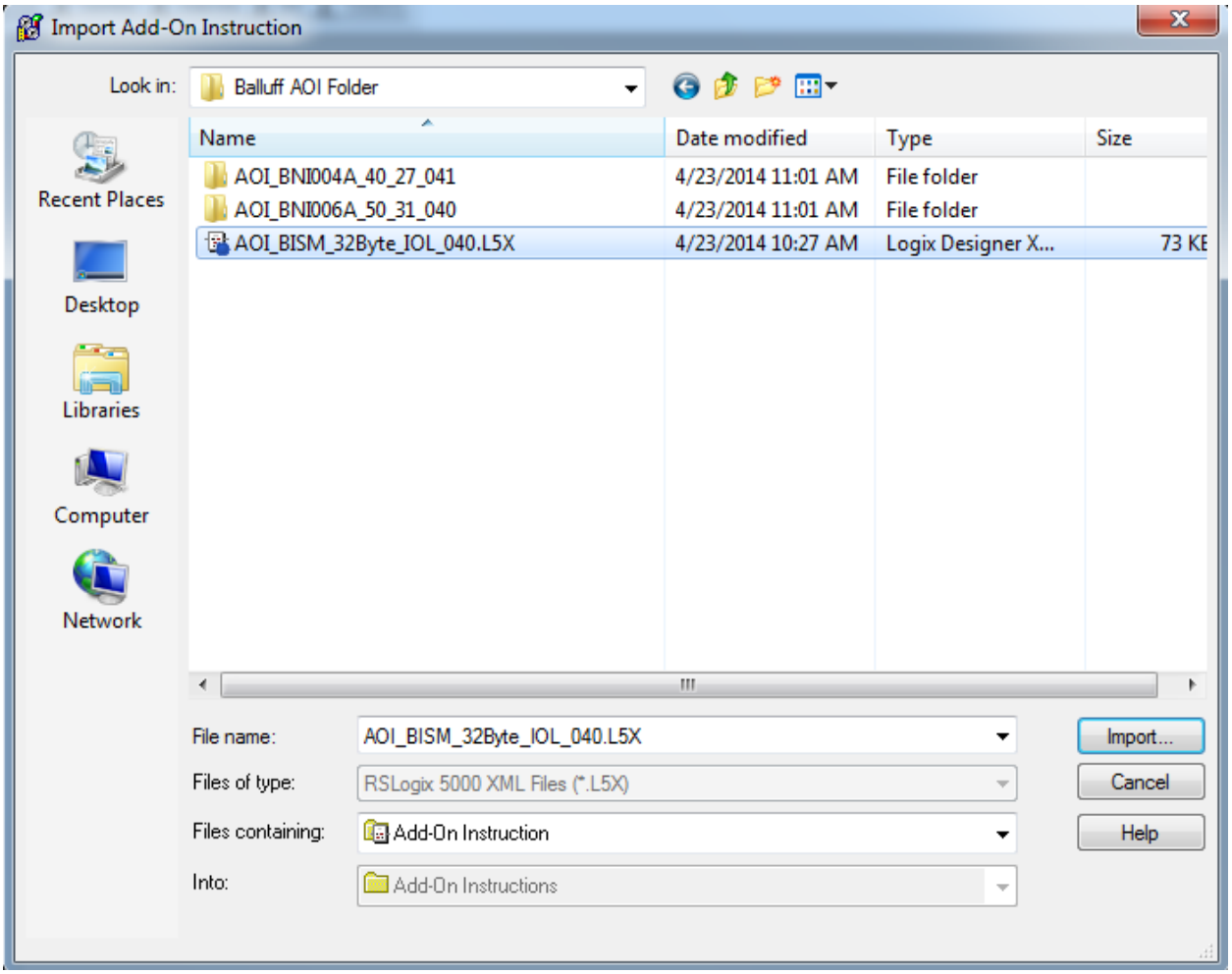
viii. Step 6: Observe the new configuration of the BIS M. When a codetag/data carrier is moved within range of the RFID processor, 8 bytes of data will automatically be passed to the PLC.

3.3 Import AOI

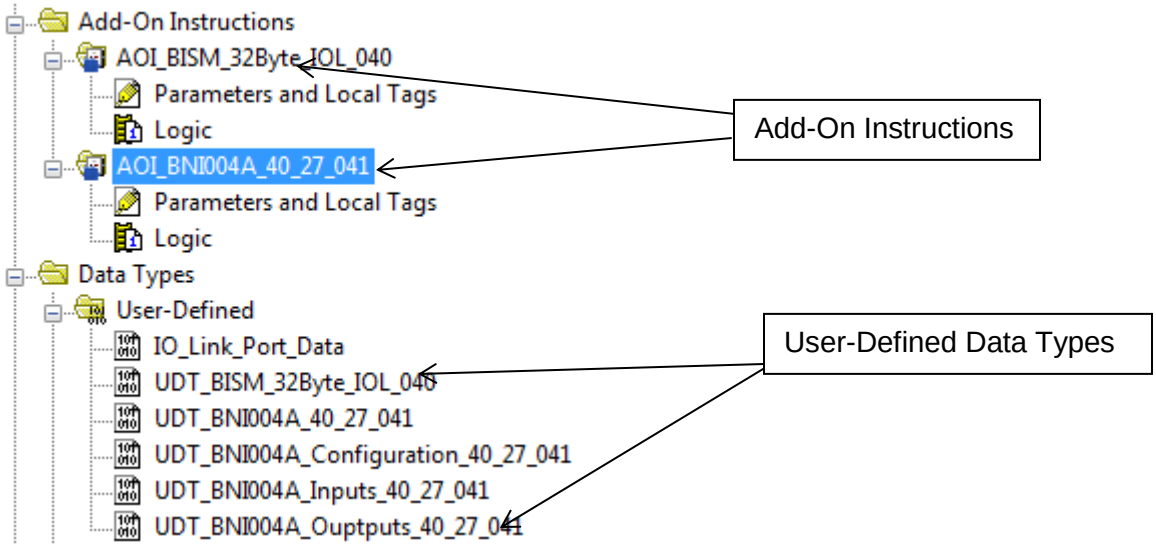
3.3.1 Right click on the Add-On Instructions element of the Controller Organizer to view a menu that includes “Import Add-On Instruction…” as shown below:



3.3.2 Locate your folder containing the downloaded Balluff AOI modules, highlight BMC_AOI_PROC_BISM045.L5X or AOI_BISM_32Byte_IOL_040.L5X and click *Import…* as shown below:



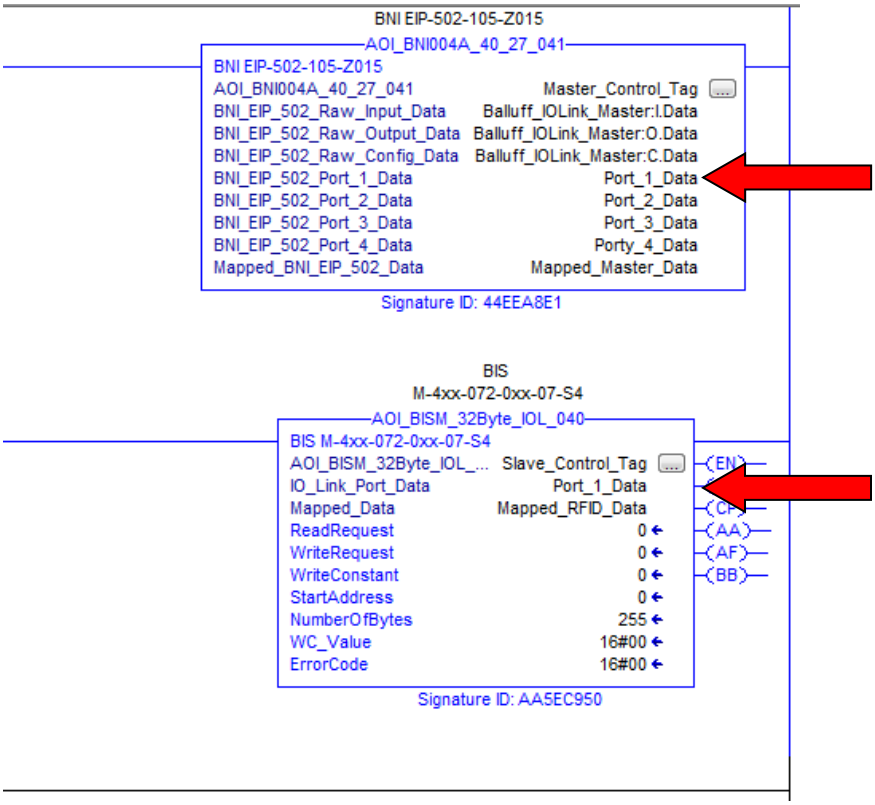
3.3.3 If it has not already been done, also import the AOI for the IO-Link master in the same manner. If the operation is successful, all UDTs associated with the AOI module will be imported into the project along with the AOI. This can be verified by checking the User-Defined element under Data Types in the Controller Organizer.



3.4 Create logic using RSLogix5000

3.4.1 Once an AOI has been imported, it can be used in the same manner as other ladder instructions. A new Element Group tab will be displayed in RSLogix5000's Language Element toolbar. This is shown here:

3.4.2 The AOI module can now be added to a rung of ladder:

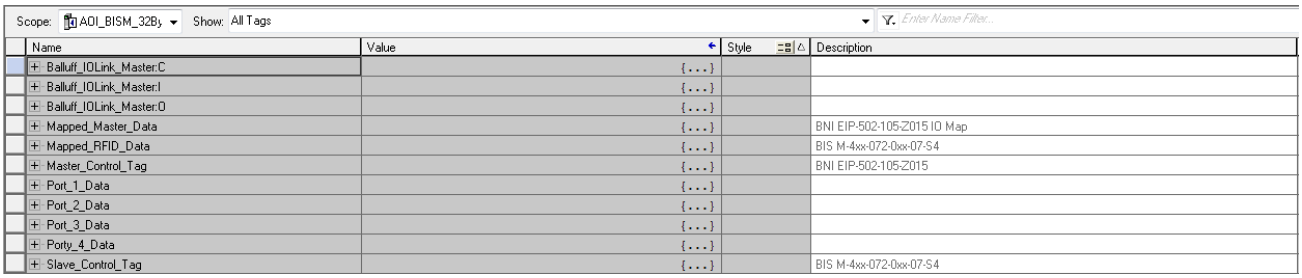


Note that in order for the AOI for the BIS M to function, it must be tied to the corresponding AOI for the Balluff IO-Link master. This is accomplished as shown above by using a “Drag and Drop” procedure to copy the BNI_EIP_502_Port_1_Data tag from the master AOI to the IO_Link_Port_Data tag in the slave. In the above example, the tag name that is copied this way is “Port1_Data” which is of the data type IO_Link_Port_Data. The tag chosen for this “Drag and Drop” is dependent on which port of the IO-Link master device that the slave is connected to.

3.4.3 All parameters must be defined with tags of valid data type as defined by the AOI module. The user should create each Tag Name relevant to the user's project.

Parameter Name	Description	Data Type	Tag Name (User specific)
AOI_BISM_32Byte_IOL…	Unique Control Tag for AOI Module	AOI_BISM_32Byte_IOL_040	Slave_Control_Tag
IO_Link_Port_Data	All I/O Data Associated with IO-Link Port 1	IO_Link_Port_Data	Port_1_Data
Mapped_BISMIOL_Data	All Unique data associated with RFID module	UDT_BISM_32Byte_IOL_040	Mapped_RFID_Data

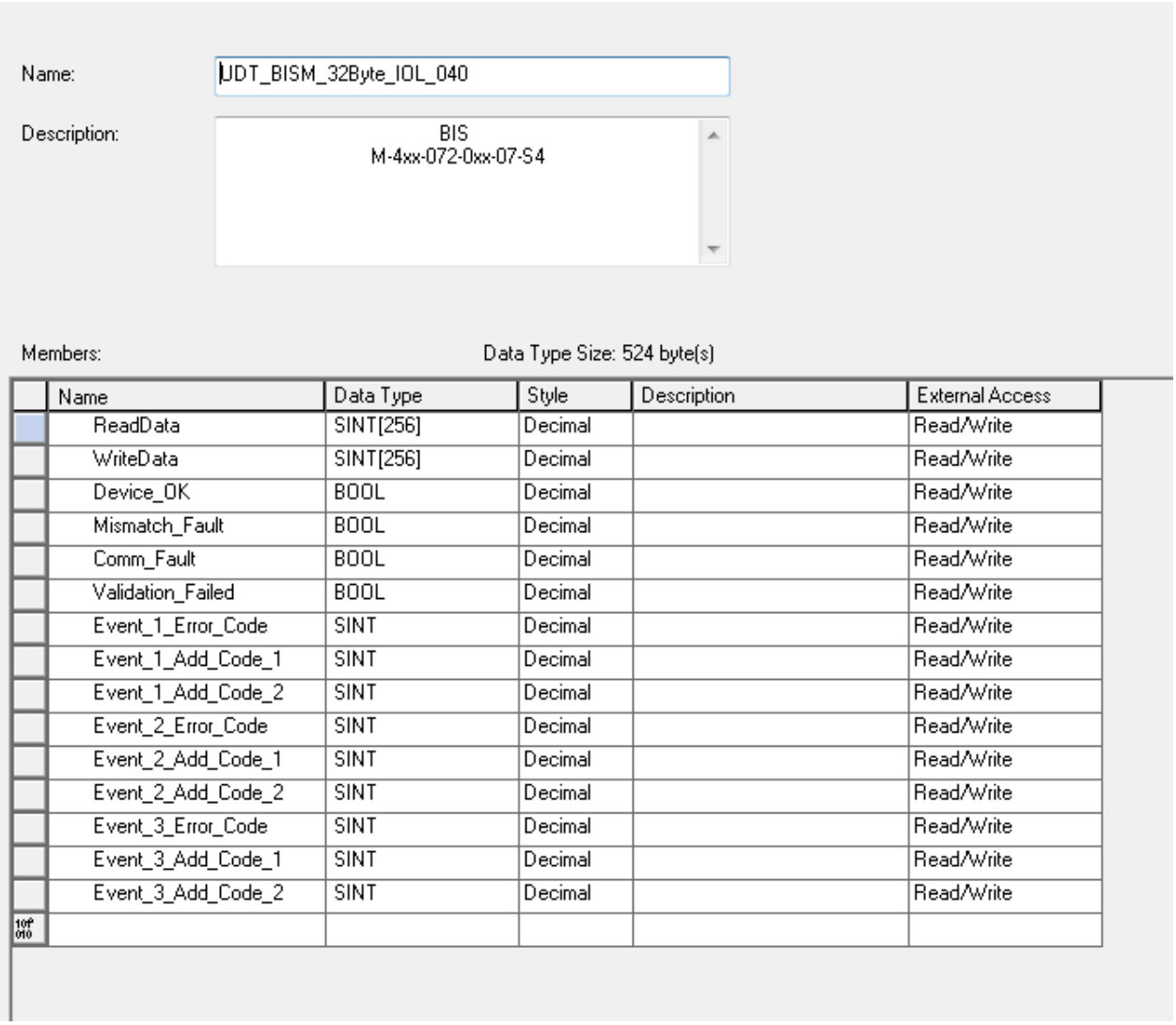
3.4.4 It is also important to verify that the following Local Tags have been created, along with the AOI Parameters, when the AOI file is imported. Because they are local tags, multiple instances of the AOI will not cause any conflicts. Note that the associated UDTs will be imported to the project along with the AOI_BISM_32Byte_IOL_040.L5X or BMC_AOI_PROC_BISM045.L5X files.



4.0 User-Defined Data Type (UDT)

4.1 The User-Defined Data Type for the Add On Instruction defines the interface for the AOI and the user's project. For AOI_BISM_IOL_040, only one UDT, UDT_BISM_IOL, is used for passing data between the AOI and the master project. Three additional UDTs (Header_Byte_I, Header_Byte_O, and M_IOL_BUFFER) are used internally within the AOI. One additional UDT is defined, IO_Link_Port_Data, for integrating data through configured IO-Link ports. Our Example Project also includes an optional UDT (Read_write_parameters) that facilitates passing of RFID command parameters to the AOI.

The main UDT (UDT_BISM_IOL) consists of all values associated with the inputs and outputs of the BIS M device. A description of this UDT and its functions is included here:



Name: UDT_BISM_32Byte_IOL_040

Description: BIS M-4xx-072-0xx-07-S4

Members: Data Type Size: 524 byte(s)

Name	Data Type	Style	Description	External Access
ReadData	SINT[256]	Decimal		Read/Write
WriteData	SINT[256]	Decimal		Read/Write
Device_OK	BOOL	Decimal		Read/Write
Mismatch_Fault	BOOL	Decimal		Read/Write
Comm_Fault	BOOL	Decimal		Read/Write
Validation_Failed	BOOL	Decimal		Read/Write
Event_1_Error_Code	SINT	Decimal		Read/Write
Event_1_Add_Code_1	SINT	Decimal		Read/Write
Event_1_Add_Code_2	SINT	Decimal		Read/Write
Event_2_Error_Code	SINT	Decimal		Read/Write
Event_2_Add_Code_1	SINT	Decimal		Read/Write
Event_2_Add_Code_2	SINT	Decimal		Read/Write
Event_3_Error_Code	SINT	Decimal		Read/Write
Event_3_Add_Code_1	SINT	Decimal		Read/Write
Event_3_Add_Code_2	SINT	Decimal		Read/Write

4.1.1 ReadData- This is the area populated when you Read data from the Data Carrier.

4.1.2 WriteData- This is the area you put the data you want to Write to the Data Carrier.

4.1.3 Device_OK-this Boolean value indicates that the IO-Link master has established communication with an IO-Link device at the expected port. This could mean that an IO-Link device is not connected or that there is a defective/broken cable between the IO-Link master and slave device.

4.1.4 Mismatch_Fault – this Boolean value indicates that the IO-Link master has established

communication with an IO-Link device at this port, but not the BIS M device.

4.1.5 Comm_Fault – this Boolean value indicates that the IO-Link master has not established communication with an IO-Link device at this port. This could mean that an IO-Link device is not connected or that there is a defective/broken cable between the IO-Link master and slave devices.

4.1.6 Validation Failed – If Validation is used this bit indicated the Vendor ID or Device ID did not match.

4.1.7 Event 1 Error Code – this SINT byte will contain the 1st of 3 bytes containing the IO-Link Error Code generated from the RFID processor.

4.1.8 Event_01_Add_Code_1 – this SINT byte will contain the 2nd of 3 bytes containing the IO-Link Error Code generated from the RFID processor.

4.1.9 Event_01_Add_Code_2 – this SINT byte will contain the 3rd of 3 bytes containing the IO-Link Error Code generated from the RFID processor.

4.1.10 Event_2_Error_Code – When there is a second or 3rd Event, the Error code cycles from 1 to 2 to 3 in a first In First out format.

4.1.11 Event_02_Add_Code_1- When there is a second or 3rd Event, the Error code cycles from 1 to 2 to 3 in a first In First out format.

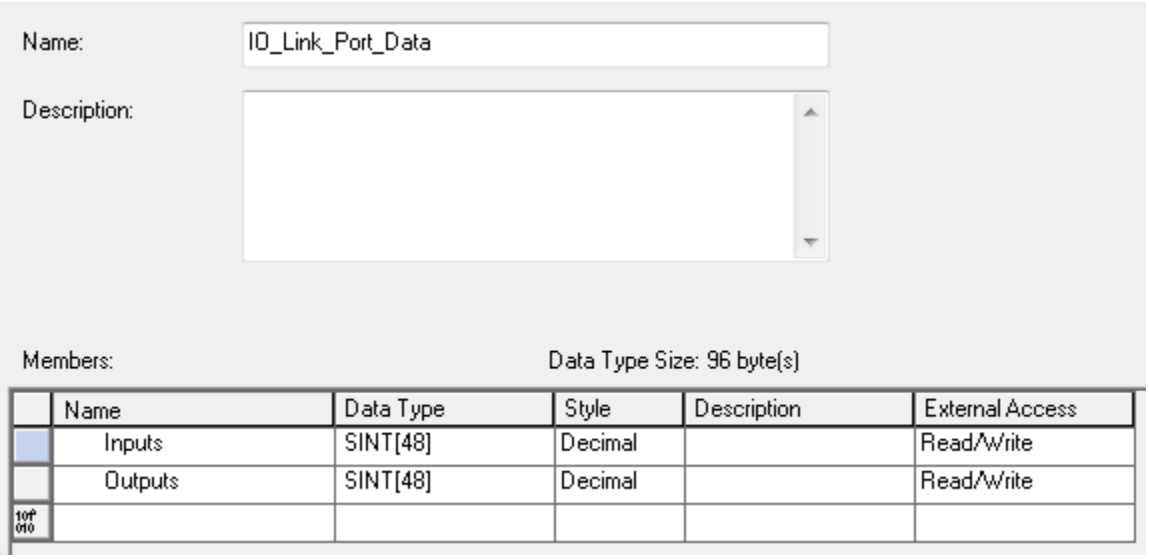
4.1.12 Event_02_Add_Code_2– When there is a second or 3rd Event, the Error code cycles from 1 to 2 to 3 in a first In First out format.

4.1.13 Event_3_ Error_Code - When there is a second or 3rd Event, the Error code cycles from 1 to 2 to 3 in a first In First out format.

4.1.14 Event_03_Add_Code_1-When there is a second or 3rd Event, the Error code cycles from 1 to 2 to 3 in a first In First out format.

4.1.15 Event_03_Add_Code_2–When there is a second or 3rd Event, the Error code cycles from 1 to 2 to 3 in a first In First out format.

4.2 The IO_Link_Port_Data UDT consists of the 48 input bytes (SINTs) and 48 output bytes (SINTs) that the IO-Link master allocates to each port. Under normal operating conditions, the user will not be aware of any data being passed through this UDT. The AOI automates the parsing of the data for each port and allows the user to access the data directly without mapping each of the four port's data into the master's single data buffer. This UDT is used for both the IO-Link master AOI as well as each IO-Link slave AOI. The master and slave must share the same tag name for the corresponding port.



4.2.1 Inputs – this 48-byte SINT array contains all raw input data values being passed from the IO-Link slave device input array to the master device input array.

4.2.2 Outputs – this 48-byte SINT array contain all raw output data values being passed from the IO-Link slave device output array to the master device output array.

4.3

Slave_Control_Tag	{...}		BIS M-4xx-072-0xx-07-S4
Slave_Control_Tag.EnableIn	1	Decimal	BIS M-4xx-072-0xx-07-S4 Enable Input - System Defined Parameter
Slave_Control_Tag.EnableOut	0	Decimal	BIS M-4xx-072-0xx-07-S4 Enable Output - System Defined Parameter
Slave_Control_Tag.InputHeader	2#0000_0000	Binary	BIS M-4xx-072-0xx-07-S4 Input Header
Slave_Control_Tag.OutputHeader	2#0000_0000	Binary	BIS M-4xx-072-0xx-07-S4 Output Header
Slave_Control_Tag.ReadRequest	0	Decimal	BIS M-4xx-072-0xx-07-S4 Read Request Bit
Slave_Control_Tag.WriteRequest	0	Decimal	BIS M-4xx-072-0xx-07-S4 Write Request Bit
Slave_Control_Tag.InitCRC_16	0	Decimal	BIS M-4xx-072-0xx-07-S4 Initialize CRC_16 Data Bit
Slave_Control_Tag.WriteConstant	0	Decimal	BIS M-4xx-072-0xx-07-S4 Write Constant Request Bit
Slave_Control_Tag.GroundStateRequest	0	Decimal	BIS M-4xx-072-0xx-07-S4 Ground State Request Bit
Slave_Control_Tag.ShutdownRequest	0	Decimal	BIS M-4xx-072-0xx-07-S4 Shutdown Request Bit
Slave_Control_Tag.StartAddress	0	Decimal	BIS M-4xx-072-0xx-07-S4 Start Address of Job
Slave_Control_Tag.NumberOfBytes	255	Decimal	BIS M-4xx-072-0xx-07-S4 Number of Bytes for Job (Max 255)
Slave_Control_Tag.WC_Value	16#00	Hex	BIS M-4xx-072-0xx-07-S4 Write Constant Value
Slave_Control_Tag.ErrorCode	16#00	Hex	BIS M-4xx-072-0xx-07-S4 Error Code from Device
Slave_Control_Tag.EN	0	Decimal	BIS M-4xx-072-0xx-07-S4 AOI Enabled/Device Correct
Slave_Control_Tag.ER	0	Decimal	BIS M-4xx-072-0xx-07-S4 AOI Disabled/Device Disconnected or Incorrect
Slave_Control_Tag.CP	0	Decimal	BIS M-4xx-072-0xx-07-S4 Codetag Present
Slave_Control_Tag.AA	0	Decimal	BIS M-4xx-072-0xx-07-S4 Job Accepted
Slave_Control_Tag.AE	0	Decimal	BIS M-4xx-072-0xx-07-S4 Job End
Slave_Control_Tag.AF	0	Decimal	BIS M-4xx-072-0xx-07-S4 Job Error
Slave_Control_Tag.T_O	0	Decimal	BIS M-4xx-072-0xx-07-S4 TO Toggle
Slave_Control_Tag.HF	0	Decimal	BIS M-4xx-072-0xx-07-S4 Head Failure
Slave_Control_Tag.BB	0	Decimal	BIS M-4xx-072-0xx-07-S4 Device Ready
Slave_Control_Tag.AV	0	Decimal	BIS M-4xx-072-0xx-07-S4 Job Pending
Slave_Control_Tag.GR	0	Decimal	BIS M-4xx-072-0xx-07-S4 Ground Reset
Slave_Control_Tag.KA	0	Decimal	BIS M-4xx-072-0xx-07-S4 Head Shutdown
Slave_Control_Tag.T_I	0	Decimal	BIS M-4xx-072-0xx-07-S4 TI Toggle

4.3.1 Slave-Control_Tag_Input Header – this is a view of the Input Bit Header.

4.3.2 Slave-Control_Tag_Output Header – this is a view of the Output Bit Header.

4.3.3 Slave-Control_Tag_ReadRequest – this BOOL value indicates that a Read command is being triggered. This bit should be maintained but it must not pass the trigger to the AOI unless the Codetag Present (CP) bit has been confirmed from the RFID processor. .

4.3.4 Slave-Control_Tag_WriteRequest – this BOOL value indicates that a Write command is being triggered. This bit should be maintained but it must not pass the trigger to the AOI unless the Codetag Present (CP) bit has been confirmed from the RFID processor.

4.3.5 Slave-Control_Tag_WriteConstant– this BOOL value indicates that a Write Constant command is being triggered. This bit should be maintained but it must not pass the trigger to the AOI unless the Codetag Present (CP) bit has been confirmed from the RFID processor.

4.3.6 Slave-Control_Tag_GroundStateRequest – this BOOL value indicates that a Ground State command is being triggered. This bit should be a momentary bit.

4.3.7 Slave-Control_Tag_ShutdownRequest – this BOOL value indicates that a Head Shutdown command is being triggered. This bit should be a momentary bit.

4.3.8 Slave-Control_Tag_StartAddress – This is the start address on the Data Carrier to Read or Write.

4.3.9 Slave-Control_Tag_NumberOfBytes – This is the number of bytes to be read from or written to the Data Carrier. If you need to read or write more than 255 bytes, multiple commands must be used.

4.3.10 Slave-Control_Tag_WC_Value – This is the constant value you want to write to memory locations on the data carrier.

4.3.11 Slave-Control_Tag_Error Code – If the AF bit goes high, the RFID Error is displayed here. It will reset to zero on the next rising edge of the AV bit (new request).

4.3.12 Slave-Control_Tag.EN - This BOOL value indicates that the AOI is enabled and the correct type

of RFID reader has been indentified.

4.3.13 Slave-Control_Tag.ER - This BOOL value indicates that the AOI is disabled or the RFID reader has been disconnected or is an incorrect type

4.3.14 Slave-Control_Tag.CP - This BOOL value indicates a Data Carrier (RFID Tag) is detected by the RFID reader.

4.3.15 Slave-Control_Tag.AA - This BOOL value indicates that the RFID reader has received a read/write request and is in process of executing the command. This bit should be monitored to ensure that no more than one request is being sent to the RFID reader at any given time.

4.3.16 Slave-Control_Tag.AE - This BOOL value indicates that the RFID reader has finished processing the request. Note: In the case of a ReadRequest, this bit will be HIGH before the job is actually complete (meaning before the control system has collected all data from the RFID tag). Refer to AOI Implementation, Section 5.0 for details regarding how to monitor for job completion.

4.3.17 Slave-Control_Tag.AF - This BOOL value indicates that the RFID reader has encountered an error, either upon acceptance of the read/write request, or it has encountered an error during processing of the command.

4.3.18 Slave-Control_Tag.T_O - This BOOL is used within the RFID reader's input bit header and is for data handshaking purposes only. User does not need to monitor for this bit, as it is handled within the AOI logic. This bit is for display purposes only.

4.3.19 Slave-Control_Tag.HF - This BOOL value indicates that the RFID reader has a fault (reader has been disabled, or is faulty). This bit will be HIGH if a ShutdownRequest is sent to the RFID reader.

4.3.20 Slave-Control_Tag.BB - This BOOL value indicates that the RFID reader is ready on the system.

4.3.21 Slave-Control_Tag.AV - This BOOL is used within the RFID reader's output bit header for job start. User does not need to monitor for this bit, as it is handled within the AOI logic. This bit is for display purposes only.

4.3.22 Slave-Control_Tag.GR - This BOOL is used within the RFID reader's output bit header for ground reset control. User does not need to control this bit. This bit is only for display purposes. User should use .GroundStateRequest bit to issue a reset to the controller.

4.3.23 Slave-Control_Tag.KA - This BOOL is used within the RFID reader's output bit header for head shutdown control. User does not need to control this bit. This bit is only for display purposes. User should use .ShutdownRequest bit to issue a shutdown of the RFID reader.

4.3.24 Slave-Control_Tag.T_I - This BOOL is used within the RFID reader's output bit header and is for data handshaking purposes only. User does not need to control this bit, as it is handled within the AOI logic. This bit is for display purposes only.

4.3.25

Slave_Control_Tag.EN	0	Decimal	BIS M-4xx-072-0xx-07-S4	AOI Enabled/Device Correct
Slave_Control_Tag.ER	0	Decimal	BIS M-4xx-072-0xx-07-S4	AOI Disabled/Device Disconnected or Incorrect
Slave_Control_Tag.CP	0	Decimal	BIS M-4xx-072-0xx-07-S4	Codetag Present
Slave_Control_Tag.AA	0	Decimal	BIS M-4xx-072-0xx-07-S4	Job Accepted
Slave_Control_Tag.AE	0	Decimal	BIS M-4xx-072-0xx-07-S4	Job End
Slave_Control_Tag.AF	0	Decimal	BIS M-4xx-072-0xx-07-S4	Job Error
Slave_Control_Tag.T_O	0	Decimal	BIS M-4xx-072-0xx-07-S4	TO Toggle
Slave_Control_Tag.HF	0	Decimal	BIS M-4xx-072-0xx-07-S4	Head Failure
Slave_Control_Tag.BB	0	Decimal	BIS M-4xx-072-0xx-07-S4	Device Ready
Slave_Control_Tag.AV	0	Decimal	BIS M-4xx-072-0xx-07-S4	Job Pending
Slave_Control_Tag.GR	0	Decimal	BIS M-4xx-072-0xx-07-S4	Ground Reset
Slave_Control_Tag.KA	0	Decimal	BIS M-4xx-072-0xx-07-S4	Head Shutdown
Slave_Control_Tag.T_I	0	Decimal	BIS M-4xx-072-0xx-07-S4	TI Toggle

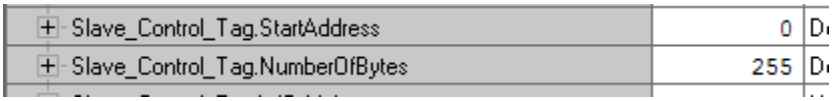
5.0 AOI Implementation

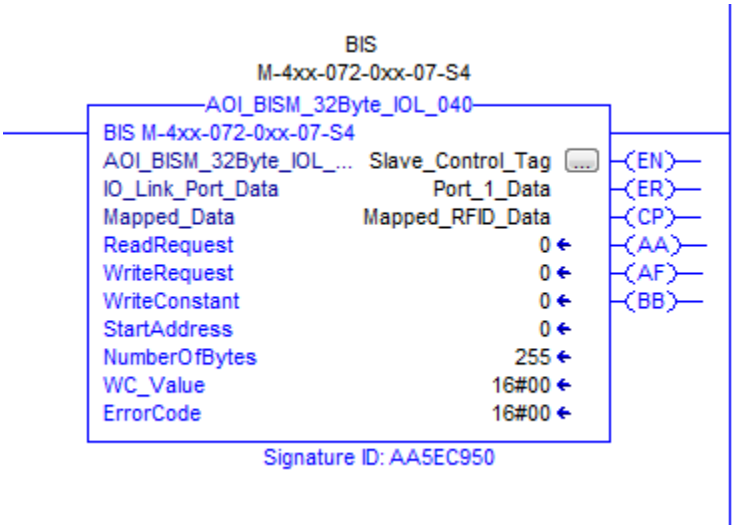
5.1 In order for the AOI to become fully functional, it is imperative that a non-zero value is entered in the Number of Bytes field for either a Read or Write command.

5.2 Unless the BIS M has been configured for Dynamic Mode using an Explicit Message (see Section 3.2.3), the user must ensure that the BIS M can detect a codetag prior to sending a new command.

5.3 Set the Start Address in the AOI Control Tag ".StartAddress". The Start Address is the memory location on the data carrier where the data will begin.

5.4 Set the Number of Bytes to Read or Write in the AOI Control Tag ".NumberOfBytes". NumberOfBytes should not exceed 256. If larger amounts of data are required, user should issue multiple commands to handle the job.





5.5 Ensure the Device Ready bit (BB) and Codetag Present bit (CP) are true and toggle the ReadRequest bit to 1. User should use One-Shot Latch logic for ReadRequest.

ReadRequest bit is self-resetting.

5.6 Now you can see the data you read from the tag in the "Mapped_RFID_Data.ReadData". Regardless of the starting memory address on the tag, the data read from the RFID data carrier will populate at ReadData[0] to ReadData[n].

For example, if the command had (StartAddress = 20) and (NumberOfBytes = 50), the data will populate at ReadData[0] thru ReadData[49].

5.7 To Write to the data carrier, populate the desired write data in Mapped_RFID_Data.WriteData[0] thru [n]. Regardless of the starting memory address on the tag, the data written to the RFID data carrier should be populated at WriteData[0] to WriteData[n].

For example, if the command had (StartAddress = 20) and (NumberOfBytes = 50), the data to write

should be populated at WriteData[0] thru WriteData[49].

5.8 Ensure the Device Ready bit (BB) and Codetag Present bit (CP) are true and toggle the WriteRequest bit to 1. User should use One-Shot Latch logic for WriteRequest.

WriteRequest bit is self-resetting.

5.9 To view the data you just wrote you must repeat the ReadRequest.

5.10 To monitor for job complete, user should monitor the falling transition of the Job End bit (AE).

If the command was carried out successfully, the Job End bit (AE) will be reset by the AOI, the Job Error bit (AF) should be LOW, and ErrorCode should equal zero.

If there was an error during the job, the Job End bit (AE) will also be reset by the AOI, and the Job Error bit (AF) will be HIGH, with the Error Code populated into .ErrorCode. The Job Error bit and ErrorCode will be set to zero automatically at the start of the next job.

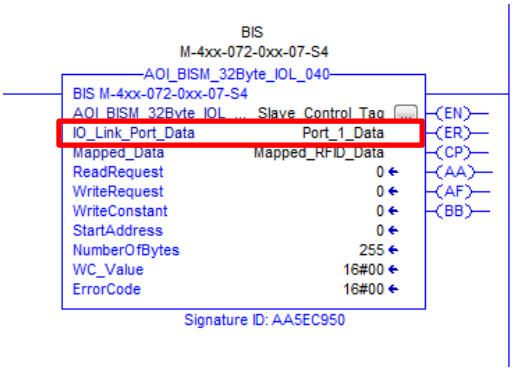
5.11 In most error conditions, the RFID reader is recoverable and the job can be repeated again. In the event the job cannot be completed successfully (after the user has followed normal troubleshooting practices), the RFID reader may require a Ground Reset.

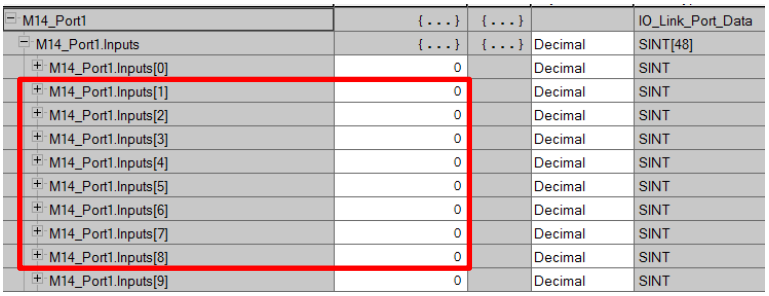
User should toggle the GroundStateRequest bit to 1. This bit is a momentary bit. If the RFID processor has acknowledged the request, the Device Ready bit (BB) will be set to zero by the RFID reader. Once this occurs, the user should reset the GroundStateRequest bit to zero and resume operation. Note: The RFID reader also resets all other input status bits (CP, AA, AE, AF, T_O) to zero during a Ground State.

5.12 To disable the RFID reader, the user should toggle the ShutdownRequest bit to 1. This bit is a momentary bit. If the processor has acknowledged the request, the Head Fault bit (HF) will be set to HIGH. User should reset the ShutdownRequest bit to zero when operation should resume.

5.13 AUTO READ - In the case where Auto-Read is used, it is important to note that the data collected from the RFID tag will NOT be passed through the AOI. If Auto-Read is utilized, the user must retrieve the data from the IO_Link_Port_Data Input array. Upon a set CP bit, the user can be assured that the (8) bytes of auto-read data are current. Below shows the location where the automatically read data will reside.

User should collect the data from IO_Link_Port_Data.Inputs[1] thru [8].





M14_Port1	{ . . . }	{ . . . }		IO_Link_Port_Data
M14_Port1.Inputs	{ . . . }	{ . . . }	Decimal	SINT[48]
M14_Port1.Inputs[0]	0		Decimal	SINT
M14_Port1.Inputs[1]	0		Decimal	SINT
M14_Port1.Inputs[2]	0		Decimal	SINT
M14_Port1.Inputs[3]	0		Decimal	SINT
M14_Port1.Inputs[4]	0		Decimal	SINT
M14_Port1.Inputs[5]	0		Decimal	SINT
M14_Port1.Inputs[6]	0		Decimal	SINT
M14_Port1.Inputs[7]	0		Decimal	SINT
M14_Port1.Inputs[8]	0		Decimal	SINT
M14_Port1.Inputs[9]	0		Decimal	SINT

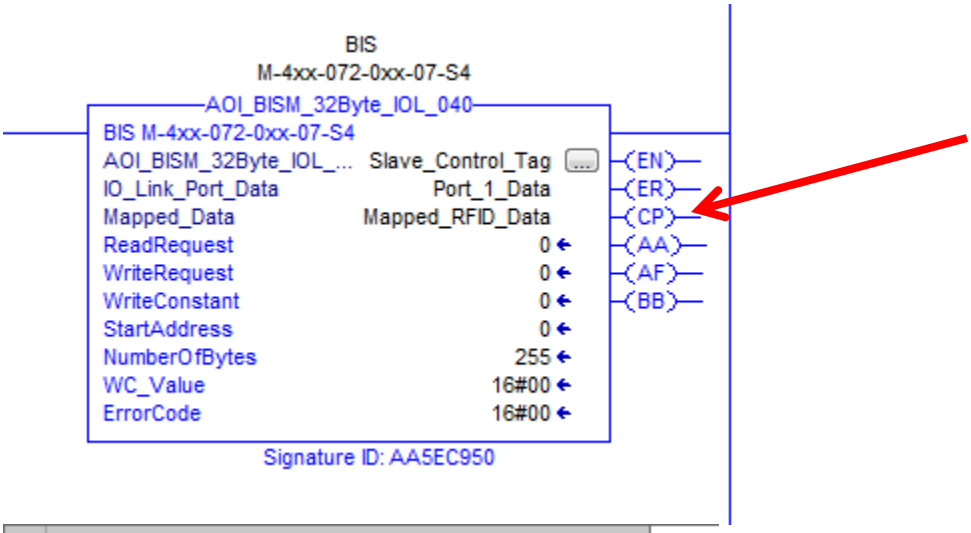
5.14 For complete functionality of the RFID Processor please refer to the user's manual for the processor itself.

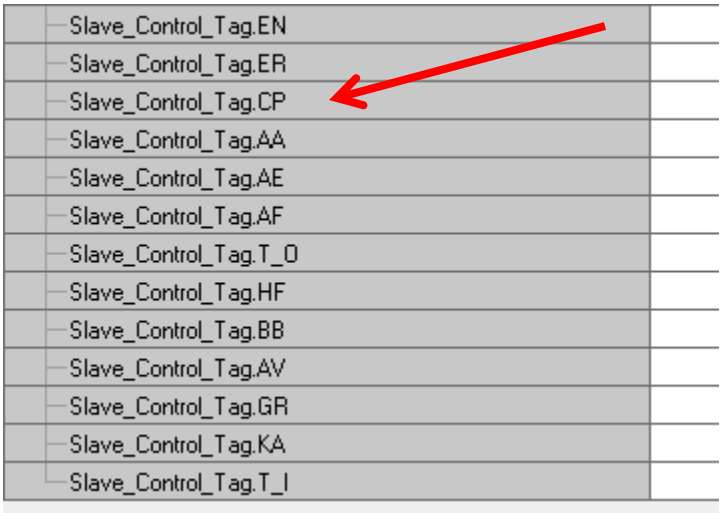
6.0 Software Validation

6.1 If the preceding instructions and implementation (3.0 Instructions and 5.0 AOI Implementation) have been successfully completed, the software module will be integrated into the RSLogix5000 project and it will be ready to use. The user will be able to verify proper operation of the AOI by monitoring the transition of a controller-scoped tag and its corresponding real world device. This section describes a sampling technique that will verify the proper operation of one input bit.

6.2 This example describes how to validate the transition of a controller-scoped input tag.

6.2.1 This example simply uses the “Data Carrier Present” (CP) bit to verify operation of the AOI. While monitoring the controller-scoped tags or the AOI, move a valid codetag/data carrier in range of the Read/Write head. This should cause the Tag Present bit to transition to ON or “high”. This bit can be viewed in the Controller scope tags as shown below or at the AOI:





Note that when the AOI module was used in the logic, the parameter Mapped_Data was assigned to a tag named “Mapped_RFID_Data”.

6.2.2 While this bit (Slave_Control_Tag.CP) is ON or “high”, the corresponding LED on the RFID processor should be lit. If this is not the case, please refer to Section 7.0 Troubleshooting.

7.0 Troubleshooting Tips

7.1 If the user does not obtain the results described in Section 6.0 above, please ensure that the following match those described in this User Guide:

7.1.1 Hardware System layout – system must consist of the following:

7.1.1.1 Rockwell Automation Logix processor

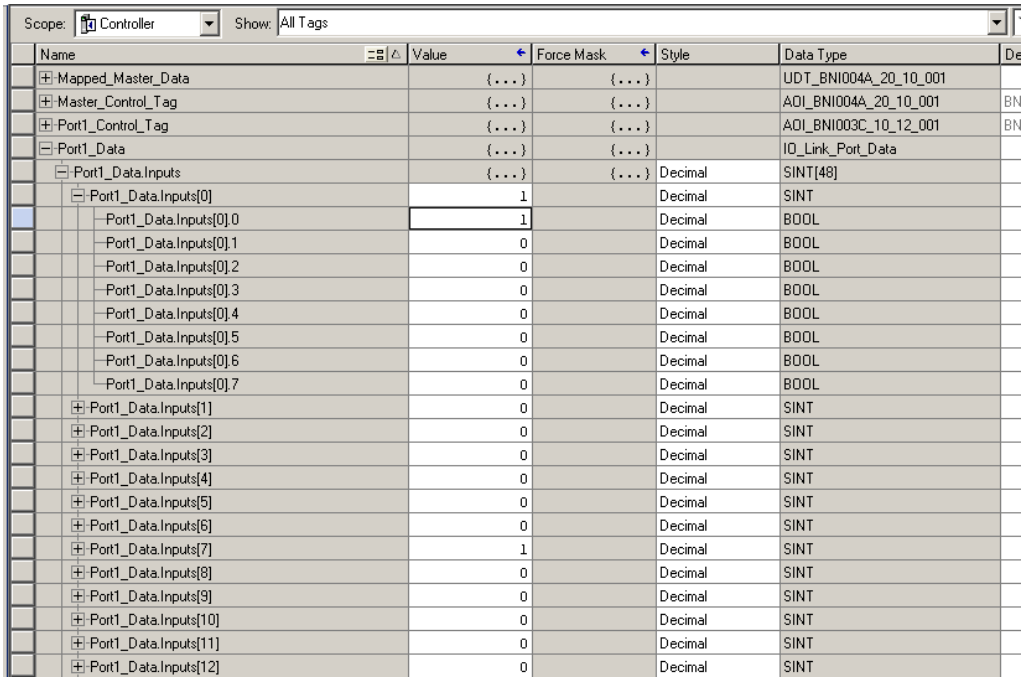
7.1.1.2 Rockwell Logix EtherNet/IP network bridge

7.1.1.3 ODVA-compliant EtherNet/IP physical media

7.1.2 Software configurations – RSLogix5000 version 18.01 or higher must be used with this Add-On Instruction

7.2 After confirming the revision levels of all system components, the next step should be to observe the raw data from the EtherNet/IP master in RSLogix5000. This can be accomplished by observing each step through the process of an input bit through the raw data into the AOI data buffers.

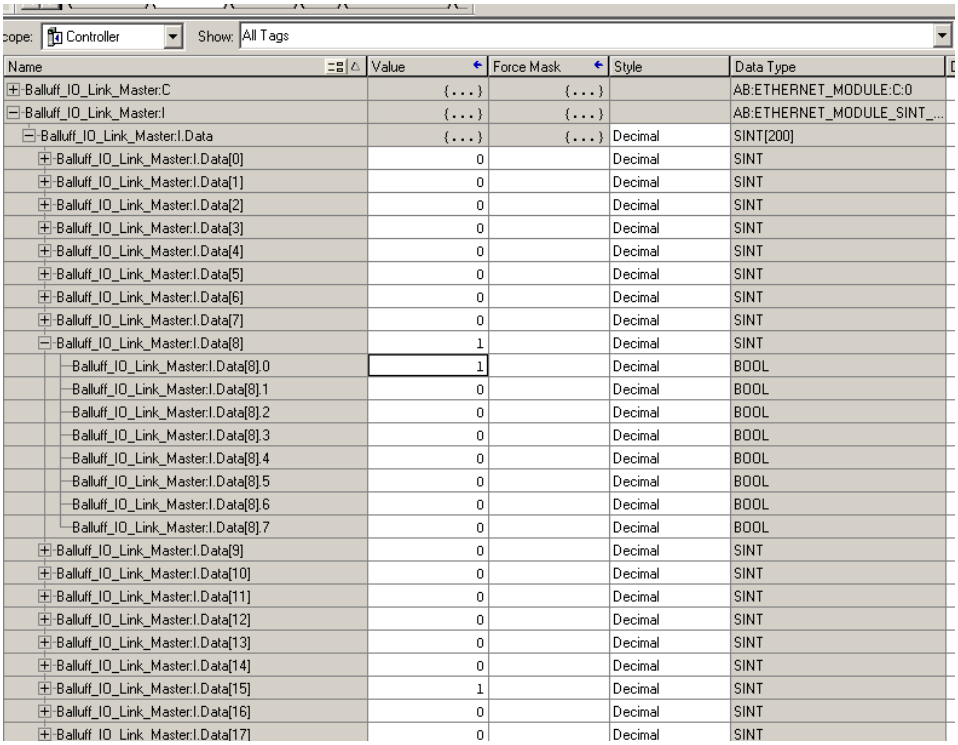
7.2.1 The initial step in confirming the proper operation of the controller-scoped input tag described in Section 6.2 (Tag Present bit of the BIS M IO-Link RFID slave) is to observe the IO_Link_Port_Data for the BIS M while the input bit has transitioned to “high” or ON. If working properly, your results will be as shown here:



Name	Value	Force Mask	Style	Data Type	De
Mapped_Master_Data	{...}	{...}		UDT_BNI004A_20_10_001	
Master_Control_Tag	{...}	{...}		AOI_BNI004A_20_10_001	BN
Port1_Control_Tag	{...}	{...}		AOI_BNI003C_10_12_001	BN
Port1_Data	{...}	{...}		IO_Link_Port_Data	
Port1_Data.Inputs	{...}	{...}	Decimal	SINT[48]	
Port1_Data.Inputs[0]	1		Decimal	SINT	
Port1_Data.Inputs[0].0	1		Decimal	BOOL	
Port1_Data.Inputs[0].1	0		Decimal	BOOL	
Port1_Data.Inputs[0].2	0		Decimal	BOOL	
Port1_Data.Inputs[0].3	0		Decimal	BOOL	
Port1_Data.Inputs[0].4	0		Decimal	BOOL	
Port1_Data.Inputs[0].5	0		Decimal	BOOL	
Port1_Data.Inputs[0].6	0		Decimal	BOOL	
Port1_Data.Inputs[0].7	0		Decimal	BOOL	
Port1_Data.Inputs[1]	0		Decimal	SINT	
Port1_Data.Inputs[2]	0		Decimal	SINT	
Port1_Data.Inputs[3]	0		Decimal	SINT	
Port1_Data.Inputs[4]	0		Decimal	SINT	
Port1_Data.Inputs[5]	0		Decimal	SINT	
Port1_Data.Inputs[6]	0		Decimal	SINT	
Port1_Data.Inputs[7]	1		Decimal	SINT	
Port1_Data.Inputs[8]	0		Decimal	SINT	
Port1_Data.Inputs[9]	0		Decimal	SINT	
Port1_Data.Inputs[10]	0		Decimal	SINT	
Port1_Data.Inputs[11]	0		Decimal	SINT	
Port1_Data.Inputs[12]	0		Decimal	SINT	

Please note that in this example, the BIS M RFID slave is connected to Port 1 of the BNI004A master.

7.2.2 The next step to confirm the raw data for the controller-scoped input tag described in Section 6.2 (Tag Present), is to observe the raw data returned from the IO-Link master as shown here:



Scope: Controller Show: All Tags

Name	Value	Force Mask	Style	Data Type
Balluff_IO_Link_Master:C	{...}	{...}		AB:ETHERNET_MODULE:C:0
Balluff_IO_Link_Master:I	{...}	{...}		AB:ETHERNET_MODULE_SINT_...
Balluff_IO_Link_Master:I.Data	{...}	{...}	Decimal	SINT[200]
Balluff_IO_Link_Master:I.Data[0]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[1]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[2]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[3]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[4]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[5]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[6]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[7]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[8]	1		Decimal	SINT
Balluff_IO_Link_Master:I.Data[8].0	1		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].1	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].2	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].3	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].4	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].5	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].6	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[8].7	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[10]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[11]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[12]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[13]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[14]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[15]	1		Decimal	SINT
Balluff_IO_Link_Master:I.Data[16]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[17]	0		Decimal	SINT

Note that because the BIS M is connected to Port 1 of the IO-Link master, the indicated bit will be byte 8, bit 0. If the slave is connected to a different port, the bit will be indicated in byte 56 (Port 2), byte 104 (Port 3), or byte 152 (Port 4).

7.3 If all system components are consistent with those described in Section 7.1 above and the raw data is being received properly in RSLogix5000 but the user does not view AOI data as described in Section 6.1 above, additional steps can be taken to verify the proper operation of both the software configuration and the hardware system and components.

7.3.1 To validate the software configuration:

7.3.1.1 Ensure that the Balluff hardware is communicating properly with the Logix processor. A communication error is easily identified when RSLogix5000 is online with the processor as the network hardware will be identified with a yellow warning triangle when the communication fails.

7.3.1.2 Start a new RSLogix5000 project using only the Balluff hardware and AOI module. This should eliminate any potential software conflicts.

7.3.2 To validate the hardware system and components:

7.3.2.1 Remove all other hardware nodes from the Ethernet network so that only the Balluff module and the Logix processor are connected.

7.3.2.2 Verify that any switches used in the network layout are operating properly and have solid connections.

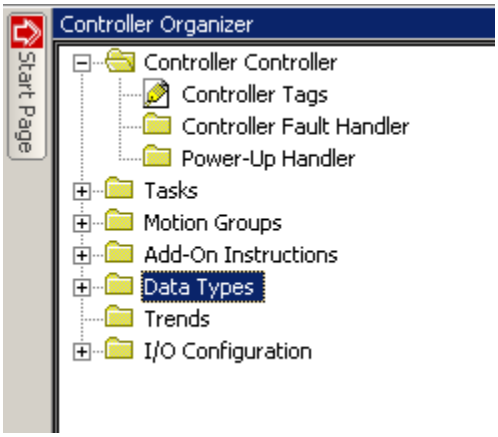
7.3.2.3 Identify and remove any potential sources of electrical noise or interference that might impede network communication.

7.4 If the system still does not respond properly, please contact the Balluff Technical Support Group at technicalsupport@balluff.com.

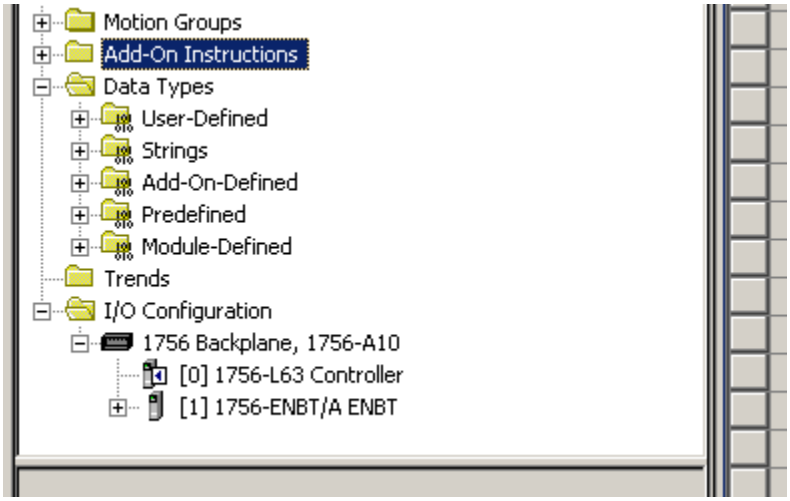
8.0 Definitions

8.1 The following terms are used in this guide with these definitions:

8.1.1 Controller Organizer – the area of RSLogix5000 where all project components can be accessed. If this toolbar is not visible, it can be opened by pressing Alt+0.



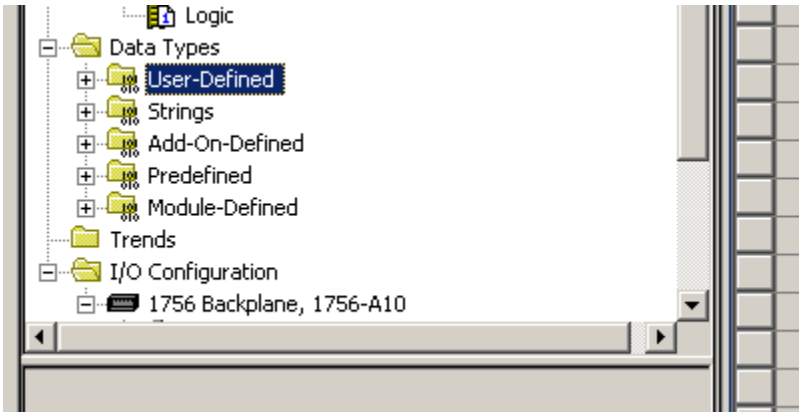
8.1.2 AOI – Add-On Instruction – reusable software module created with RSLogix5000. Balluff's AOI modules are composed of UDTs and associated logic to implement control algorithms. This module is hardware specific. All AOI modules contained in a project will be located in the Controller Organizer as shown below:



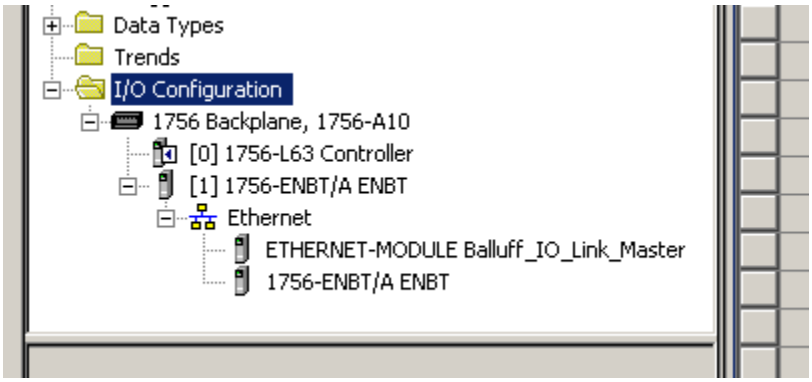
8.1.3 Controller scoped tag – a tag of any valid data type that is available to all programs within a project and must be used uniformly throughout the entire project.

8.1.4 AOI Parameter – a tag of any valid data type that is created during the definition of the AOI module and is isolated to that AOI. Its scope is limited to each discrete use of the AOI function.

8.1.5 UDT – User-Defined Data Type – reusable software module created with RSLogix5000. Balluff's UDTs define the input and output tag names to be used when implementing our AOI modules. A UDT can be composed of any combination of standard Data Types defined by RSLogix5000 or other UDTs. All UDTs contained in a project will be located in the Controller Organizer as shown below:



8.1.6 I/O Configuration Tree – the area of RSLogix5000 where all physical, or "real world", Inputs and Outputs associated with the controller are defined. One example of an I/O Configuration Tree is shown below:



8.1.7 EDS – Electronic Data Sheet – software configuration file that defines how a hardware product will communicate with the network master using EtherNet/IP. The format and content of an EDS is defined by ODVA, the controlling organization for EtherNet/IP specifications.

9.0 Related Documents

9.1 The following Balluff documents support the release of this User Guide and its related software module. Each of these documents can be obtained by contacting Balluff Technical Support at technicalsupport@balluff.com or at http://www.balluff.com/balluff/MUS/en/home.jsp:

9.1.1 AOI_BNI004A_40_27_041 User Guide

9.1.2 BNI EIP-502-105-Z015 User's Guide

9.1.3 BIS M-4xx IO-Link Device User's Guide

10.0 References

10.1 None