BALLUFF

sensors worldwide

AOI_BNI0048_10_11_040 User Guide

October 7, 2011

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

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 BNI IOL-302-S01-Z013. This software module is designed for use with RSLogix5000.

2.0 Products

2.1 This guide was developed for use with the software module AOI_BNI0048_10_11_040.L5X. This software module is available for download at the web site www.Balluff.com/AOI.

2.2 This software module was developed for use with the Balluff EtherNet/IP Connectivity block BNI IOL-302-S01-Z013-C01(hardware version 1.0, firmware version 1.2).

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

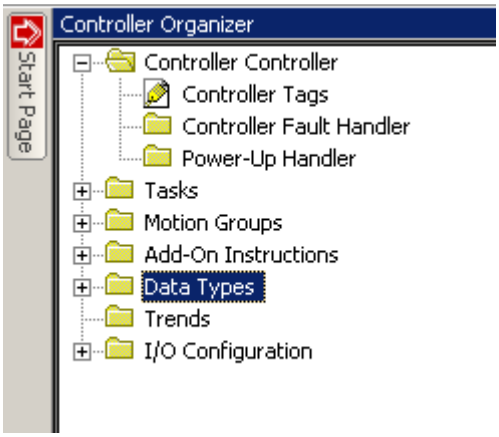
2.3.1 RSLogix5000, version 18.01 (or later)

2.3.2 Rockwell 1756-L63 controller

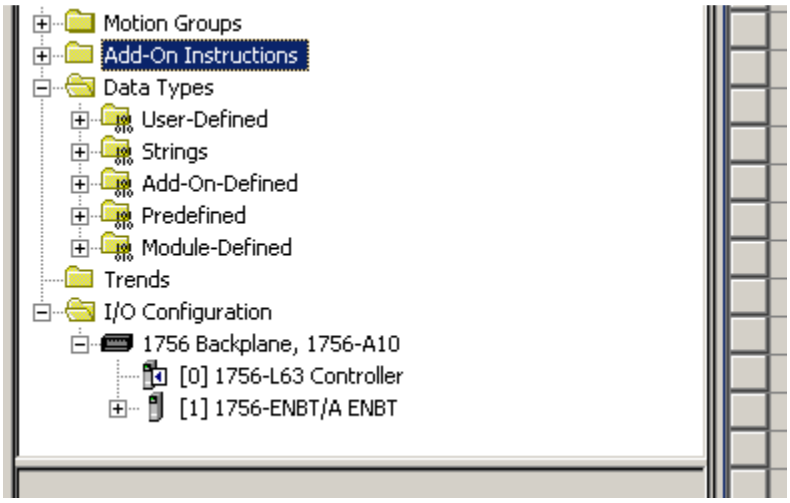
3.0 Definitions

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

3.1.1 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.



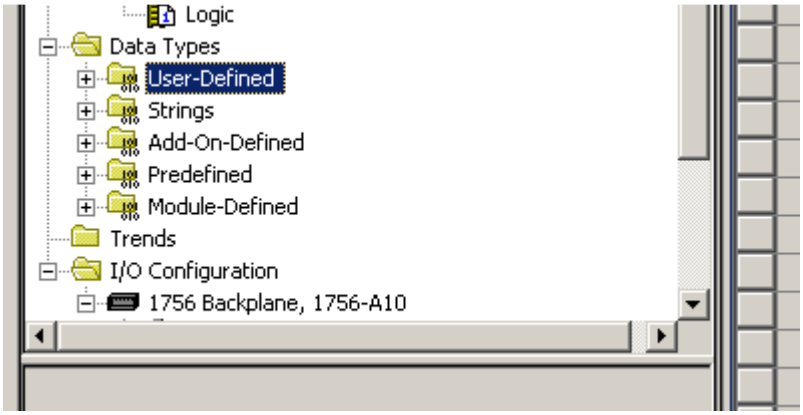
3.1.2 AOI – Add On Instruction – reusable software module created with RSLogix5000. Balluff's AOI modules are composed of UDTs and associated logic to implement control algorithms. This module is hardware specific. All AOI modules contained in a project will be located in the Controller Organizer as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



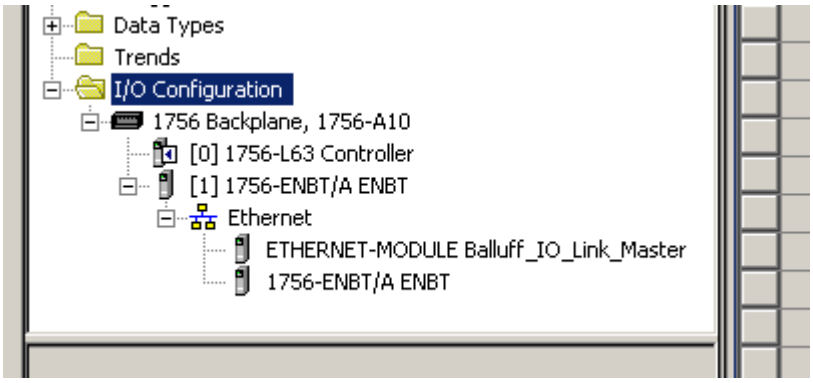
3.1.3 Controller scoped tag – a tag of any valid data type that is available to all programs within a project and must be used uniformly throughout the entire project.

3.1.4 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 the each discrete use of AOI function.

3.1.5 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:



3.1.6 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:

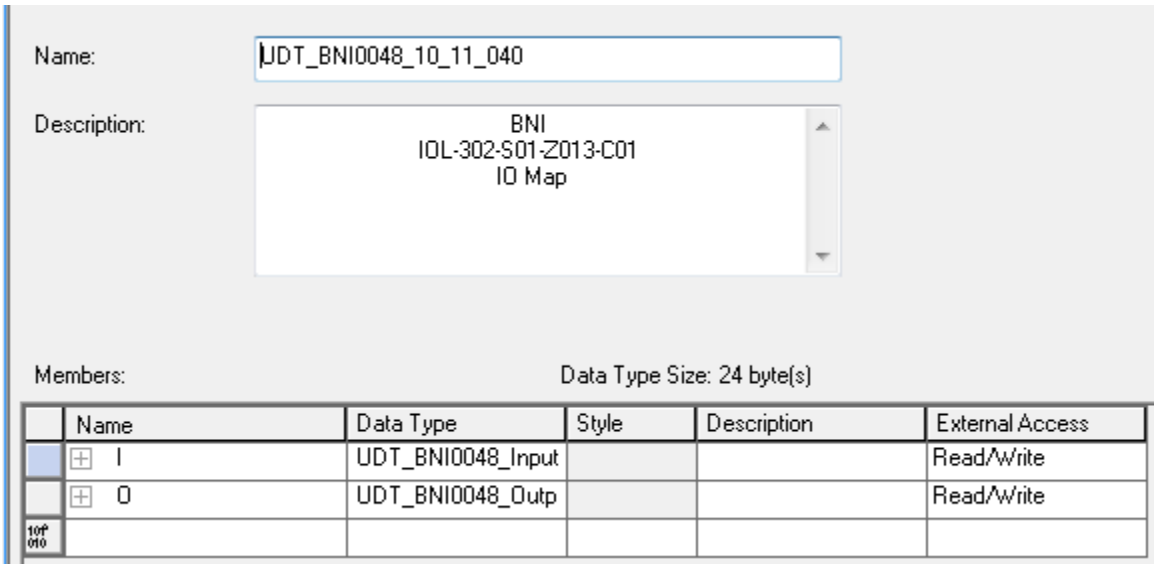


3.1.7 EDS – Electronic Data Sheet – software configuration file that defines how a hardware product will communicate with the network master using EtherNet/IP. The format and content of an EDS is defined by ODVA, the controlling organization for EtherNet/IP specficiations.

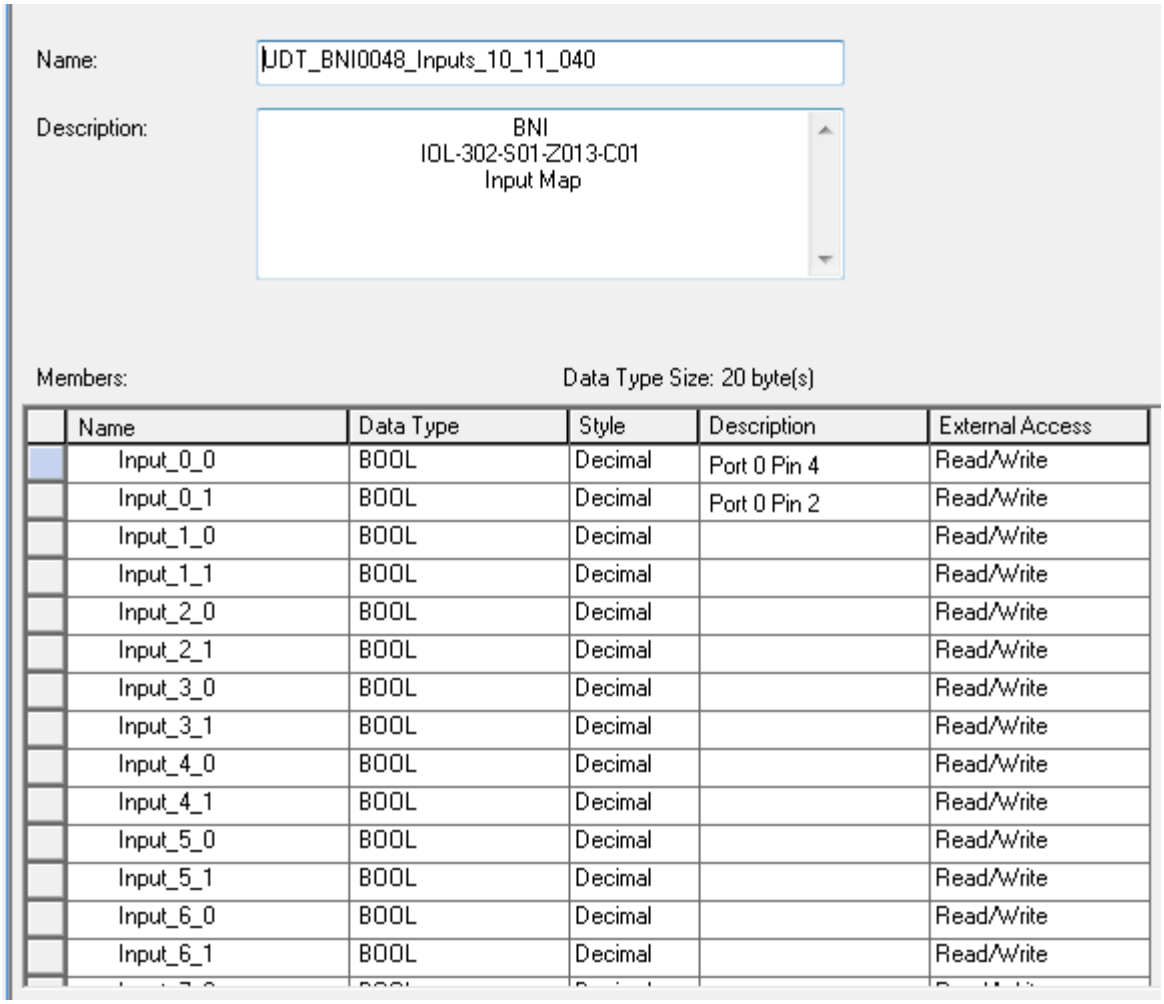
4.0 User-Defined Data Type (UDT)

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_BNI0048_10_11_040, the highest level UDT. UDT_BNI0048_10_11_040, is comprised of two sub-UDT components: Input (I) and Output (O). One additional UDT is defined, IO_Link_Port_Data, for integrating data through configured IO-Link ports.

4.2 The highest level UDT (UDT_BNI0048_10_11_040) consists of two sub-UDT components described in Sections 4.3 and 4.4 below.



4.3 The Input UDT (UDT_BNI0048_Inputs_10_11_040) consists of all inputs associated with the BNI004A block. A description of this UDT and its functions is included here:



Name: UDT_BNI0048_Inputs_10_11_040

Description: BNI
IOL-302-S01-Z013-C01
Input Map

Members: Data Type Size: 20 byte(s)

Name	Data Type	Style	Description	External Access
Input_0_0	BOOL	Decimal	Port 0 Pin 4	Read/Write
Input_0_1	BOOL	Decimal	Port 0 Pin 2	Read/Write
Input_1_0	BOOL	Decimal		Read/Write
Input_1_1	BOOL	Decimal		Read/Write
Input_2_0	BOOL	Decimal		Read/Write
Input_2_1	BOOL	Decimal		Read/Write
Input_3_0	BOOL	Decimal		Read/Write
Input_3_1	BOOL	Decimal		Read/Write
Input_4_0	BOOL	Decimal		Read/Write
Input_4_1	BOOL	Decimal		Read/Write
Input_5_0	BOOL	Decimal		Read/Write
Input_5_1	BOOL	Decimal		Read/Write
Input_6_0	BOOL	Decimal		Read/Write
Input_6_1	BOOL	Decimal		Read/Write

4.3.1 Input_a_b – (a = 0 through 7 indicating Port number, b = 0 or 1 indicating Pin number 0=Pin4 / 1=Pin2) these 16 boolean registers each contain a discrete input bit corresponding to a real world input at the defined point.

4.3.2 Short_Port_x – (x = 0 through 7 indicating Port number) these 8 boolean registers each contain a discrete bit that indicates a short circuit condition between pins 1 and 3 on the corresponding port.

4.3.3 UA_Fault – this Boolean value indicates whether auxiliary power is currently applied to the BNI004A device's output points. This bit is typically used to indicate an Emergency Stop has occurred.

4.3.4 US1_Fault – this Boolean value indicates whether sensor power from the IO-Link communication cable is currently applied to the BNI0048 device.

4.3.5 US2_Fault – this Boolean value indicates whether sensor power from the auxiliary power cable is currently applied to the BNI0048 device.

4.3.6 Overload_c_d – (c = 0 through 7 indicating Port number, d = 0 or 1 indicating Pin number 0=Pin4 / 1=Pin2) these 16 boolean registers each contains a discrete overload bit corresponding to an I/O point. These bits indicate an overload has occurred on the indicated port and pin number. Note that these bits will only function if the port is configured as an output.

4.3.7 Warning_e_f – (e = 0 through 7 indicating Port number, f = 0 or 1 indicating Pin number 0=Pin4 / 1=Pin2) these 16 boolean registers each contains a discrete warning bit corresponding to an I/O point. These bits indicate that a short has occurred between +24VDC (pin 1) and the indicated port and pin number.

4.3.8 Identification_Byte_w – (w = 0 or 1 indicating either low byte (0) or high byte (1)) these 2 Hexadecimal SINT registers hold an identification value, similar to a serial number except that it is assigned by the user. In order to change this value from its factory default (00 00), an explicit message must be sent to the BNI0048 hub with the desired value.

4.3.9 Device_OK – this Boolean value indicates that the IO-Link master has established communication with the expected device at the expected port.

4.3.10 Mismatch_Fault – this Boolean value indicates that the IO-Link master has established communication with an operational device at this port, but not the expected device.

4.3.11 Comm_Fault – this Boolean value indicates that the IO-Link master has not established communication with any operational device at this port.

4.3.12 Validation_Failed – this Boolean value indicates that the device connected to the assigned IO-Link port failed the IO-Link master block's configured Validation test. This means it did not match the Vendor ID, Device ID and/or Serial Number, depending on configuration.

4.3.13 Event_y_Error_Code – (y = 1 through 3 indicating the event number) these SINT registers hold the error codes for the three most recent error events in the BNI0039. These error codes are defined in the hardware User's Guide.

4.3.14 Event_y_Additional_Code_z – (y = 1 through 3 indicating the event number; z = 1 or 2 indicating which Additional Code) these SINT registers hold additional information associated with certain error codes. This additional information is defined in the hardware User's Guide.

4.4 The Output UDT (UDT_BNI0048_Outputs_10_11_040) consists of all outputs associated with the BNI0048 block. A description of this UDT and its function is included here:

Name: UDT_BNI0048_Outputs_10_11_040

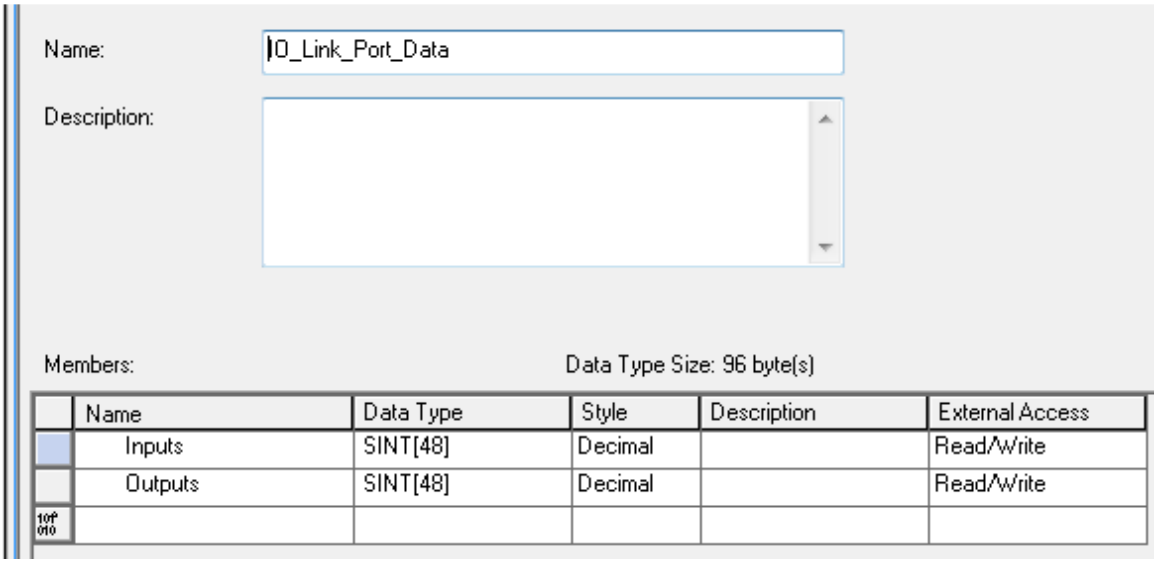
Description: BNI
IOL-302-S01-Z013-C01
Output Map

Members: Data Type Size: 4 byte(s)

Name	Data Type	Style	Description	External Access
Output_0_0	BOOL	Decimal	Port 0 Pin 4	Read/Write
Output_0_1	BOOL	Decimal	Port 0 Pin 2	Read/Write
Output_1_0	BOOL	Decimal		Read/Write
Output_1_1	BOOL	Decimal		Read/Write
Output_2_0	BOOL	Decimal		Read/Write
Output_2_1	BOOL	Decimal		Read/Write
Output_3_0	BOOL	Decimal		Read/Write
Output_3_1	BOOL	Decimal		Read/Write
Output_4_0	BOOL	Decimal		Read/Write
Output_4_1	BOOL	Decimal		Read/Write
Output_5_0	BOOL	Decimal		Read/Write
Output_5_1	BOOL	Decimal		Read/Write
Output_6_0	BOOL	Decimal		Read/Write
Output_6_1	BOOL	Decimal		Read/Write

4.4.1 Output_a_b – (a = 0 through 7 indicating Port number, b = 0 or 1 indicating Pin number 0=Pin4 / 1=Pin2) these 16 boolean registers each contain a discrete output bit corresponding to a real world output at the defined point.

4.5 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.



	Name	Data Type	Style	Description	External Access
	Inputs	SINT[48]	Decimal		Read/Write
	Outputs	SINT[48]	Decimal		Read/Write

4.5.1 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.

4.5.2 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.

5.0 Instructions

5.1 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.

5.2 Hardware Configuration

5.2.1 The BNI IOL-302-S01-Z013-C01 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_20_10_040 User Guide (for use with the Balluff BNI EIP-502-105-Z015).

5.2.2 When the IO-Link master has established a connection with the PLC using EtherNet/IP, the IO-Link port connected to the BNI IOL-302-S01-Z013-C01 must be configured for IO-Link mode.

5.2.3 The BNI0048 is a configurable IO-Link hub, therefore, each point can be configured either as an input or an output. This is done by writing to the parameter “Port Direction.” By default, the ports are configured as inputs (0x0000 in parameter data above). In order to change an I/O point to an output, an explicit message must be sent to the BNI0048. This section (5.2.3) details the four steps necessary to complete this process:

1) Understand the configuration environment (Section 5.2.3.1)
2) Create an Explicit Read instruction (Section 5.2.3.2)
3) Create an Explicit Write instruction (Section 5.2.3.3)
4) Using the Explicit instructions (Section 5.2.3.4)

5.2.3.1 The following shows the values required in this explicit message to configure the ports as outputs:

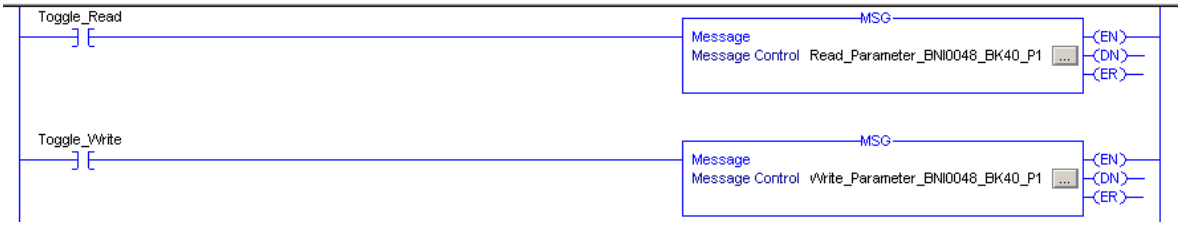
Port direction

Byte 0								Byte 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Direction 7.0	Direction 6.0	Direction 5.0	Direction 4.0	Direction 3.0	Direction 2.0	Direction 1.0	Direction 0.0	Direction 7.1	Direction 6.1	Direction 5.1	Direction 4.1	Direction 3.1	Direction 2.1	Direction 1.1	Direction 0.1

Direction port (x)
x.0: Pin 4,
x.1: Pin 2

Direction port (x)
0: Input
1: Output

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”:

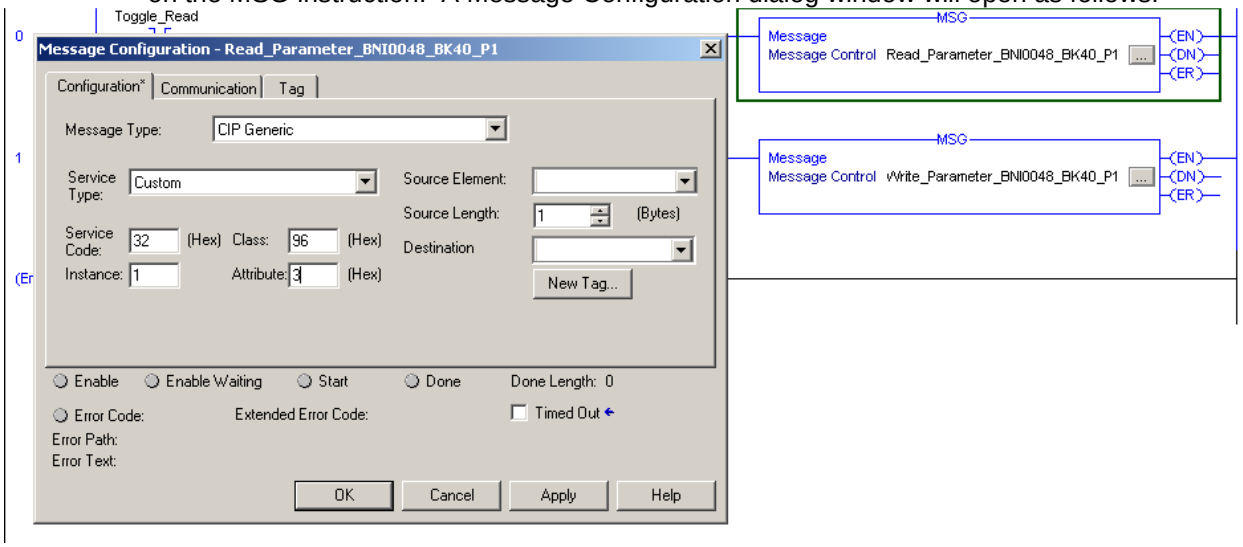
Read_Parameter_BNI0048_P1

Write_Parameter_BNI0048_P1

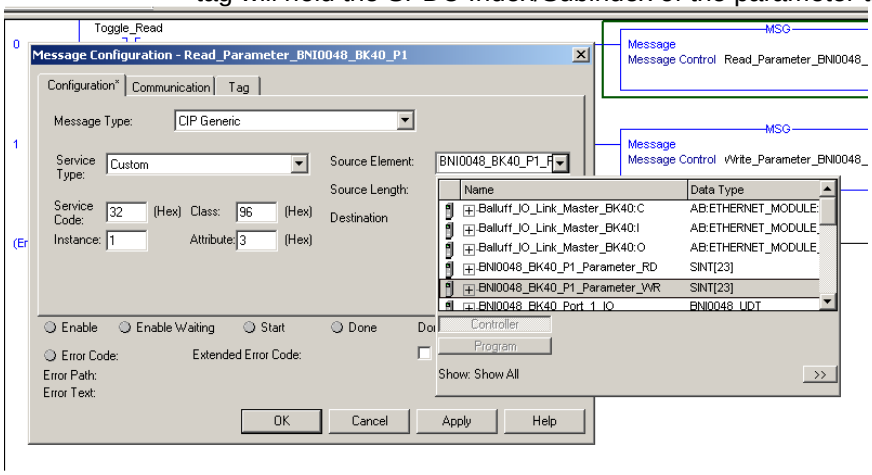
ii. Below are all the properties for the BNI EIP-502-105-Z015, which is Balluff’s IO-Link master block. The values in the table are decimal and must be entered in the RSLogix5000 Message Configuration window in Hex where specified:

Description	Request		
CIP Service Code	50	SPDU	
Reserved padding	0	Reserved	
CIP Class ID	150	Object	
	0		
CIP Instance ID	0	Port	1-4
	0		0
CIP Attribute	0	Read/Write	3: Read 2: Write
	0		
Service Data		Index	0 … 65535
		Subindex	0 … 255
		Data 0	
		…	
		Data n	

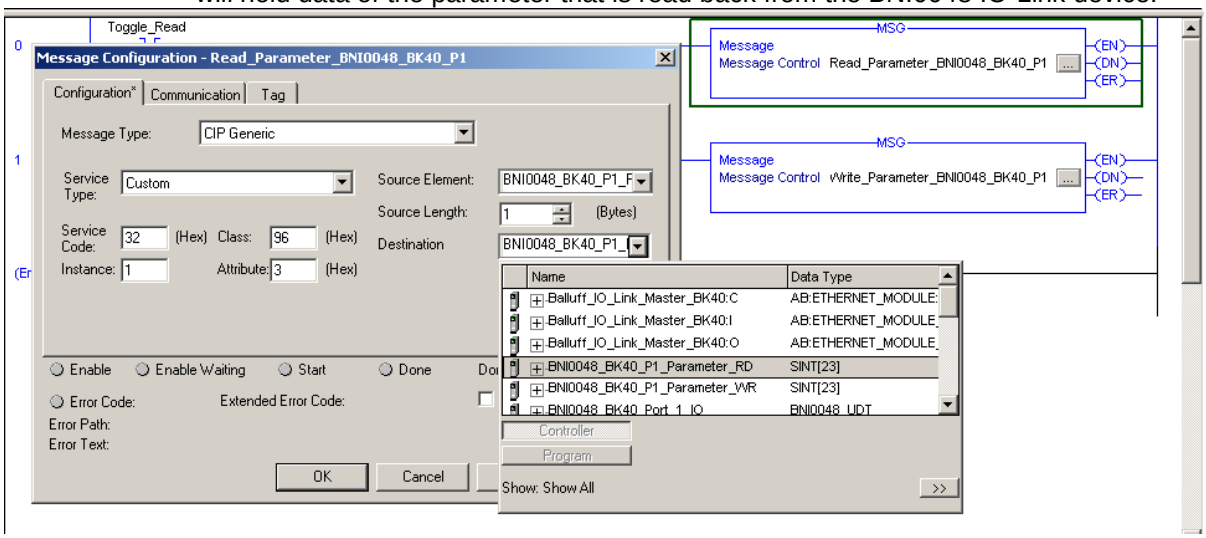
5.2.3.2 To configure the Read_Parameter_BNI0048_BK40_P1 MSG instruction, click on the radio button on the MSG instruction. A Message Configuration dialog window will open as follows:



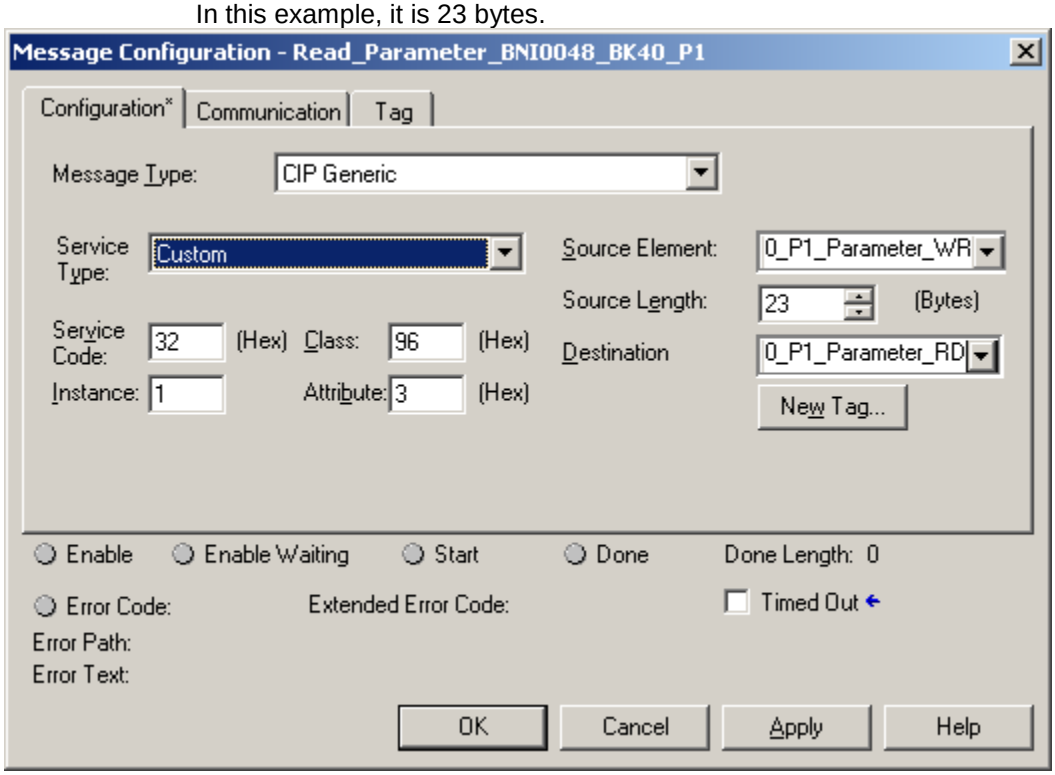
i. For the Source Element, tag BNI0048_BK40_P1_Parameter_WR must be selected. This tag will hold the SPDU Index/Subindex of the parameter that is being requested.



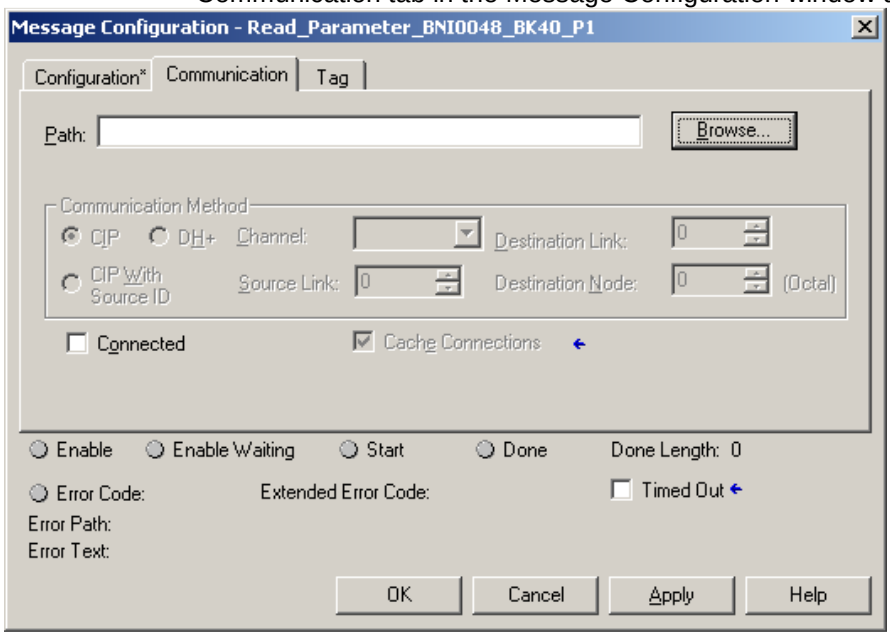
ii. For the Destination, tag BNI0048_BK40_P1_Parameter_RD must be selected. This tag will hold data of the parameter that is read back from the BNI0048 IO-Link device.



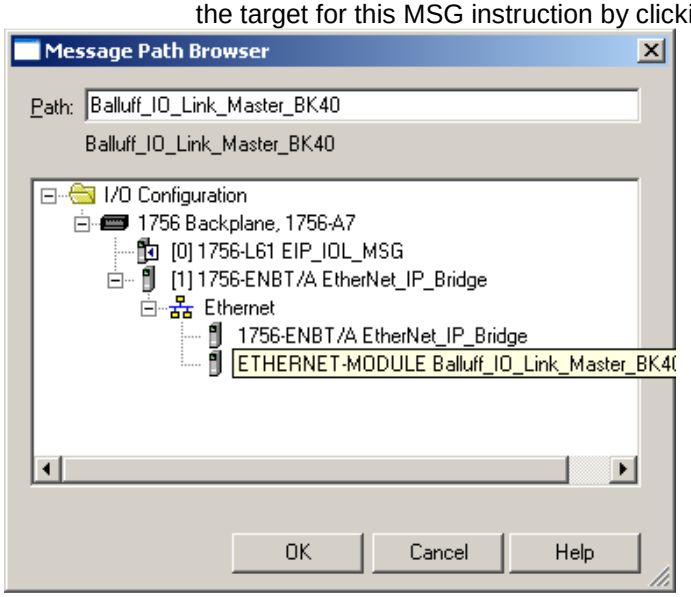
iii. The Source length will be the maximum length of the parameter that you expect to read. In this example, it is 23 bytes.



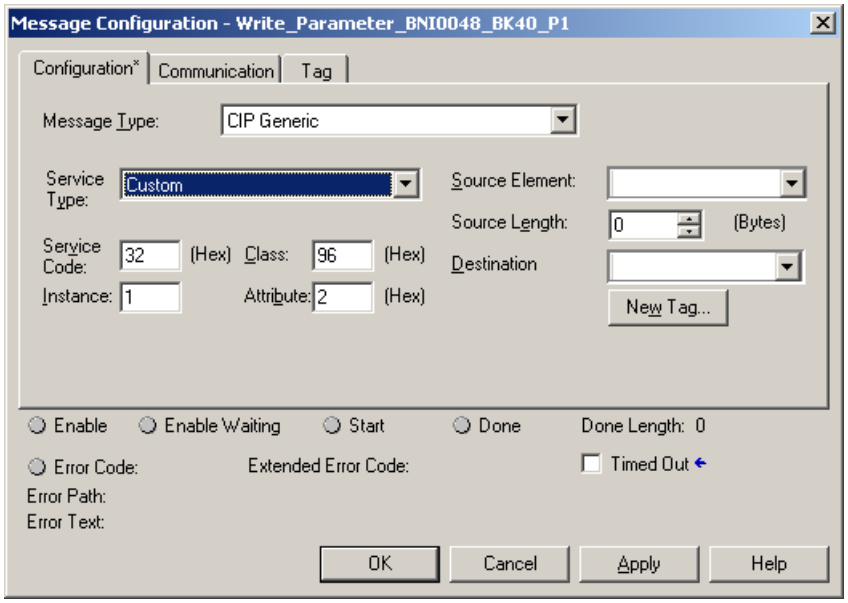
iv. Set the communication path for the MSG instruction. This is done by clicking on the Communication tab in the Message Configuration window as follows:



v. 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.



5.2.3.3 To configure the Write_Parameter_BNI0048_BK40_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

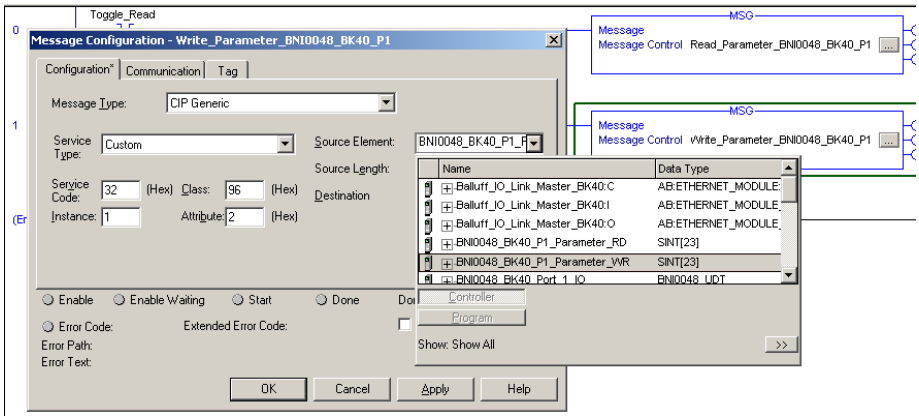
Service Code: 32 (50 Decimal)

Class: 96 (150 Decimal)

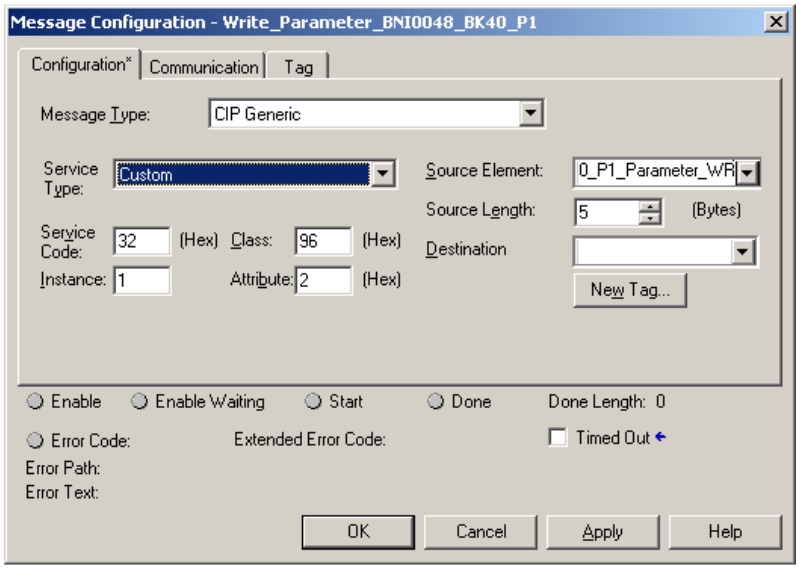
Instance: IO-Link Port Number device is connected to (Port 1 for this example)

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

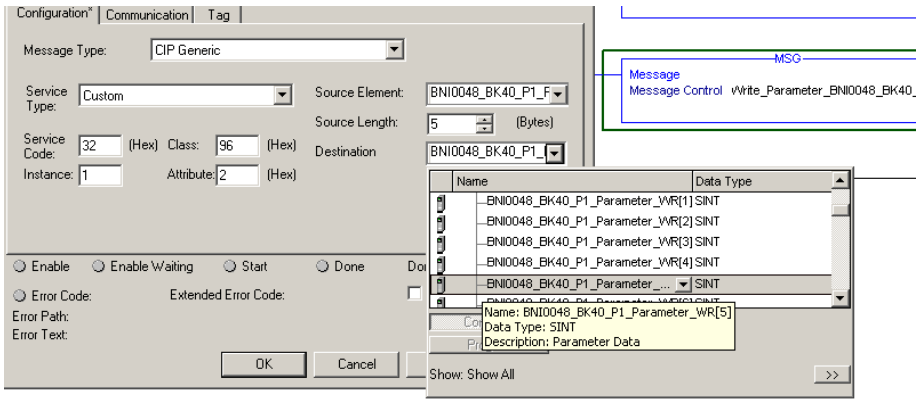
i. For the Source Element, tag BNI0048_BK40_P1_Parameter_WR must be selected. This tag will hold the SPDU Index/Subindex and values of the parameter that is being set.



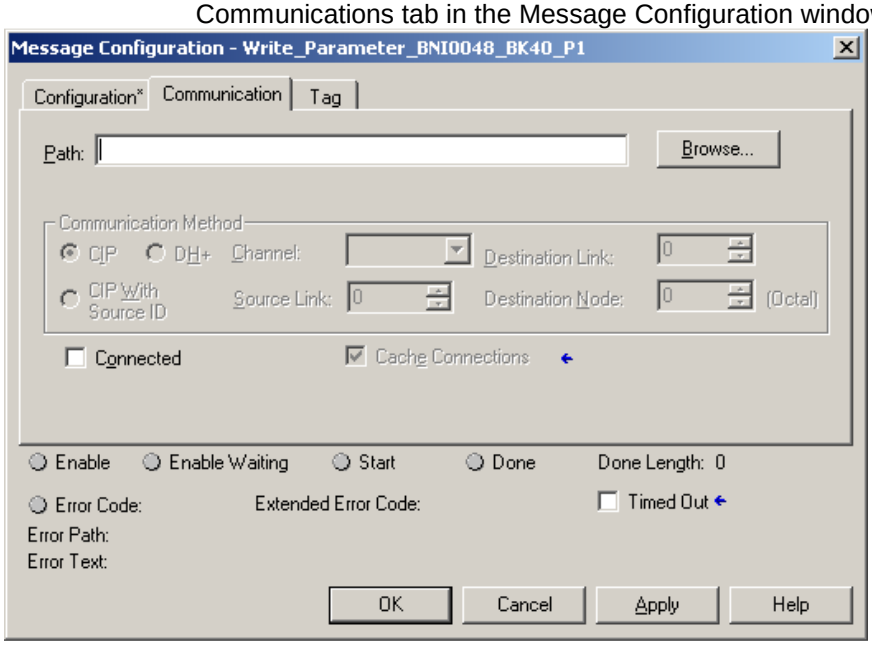
ii. The Source length will depend on the maximum length of the parameter value that is to be set. 3 Bytes will always be required for the SPDU Index Low Byte, Index High Byte and Sub-Index. Since we are configuring the BNI0048's port as inputs or outputs using the "Port Direction" parameter, a value of 2 bytes will be required as well (see Parameter Data for the BNI0048). Therefore, a total of 5 bytes will be entered for the Source Length:



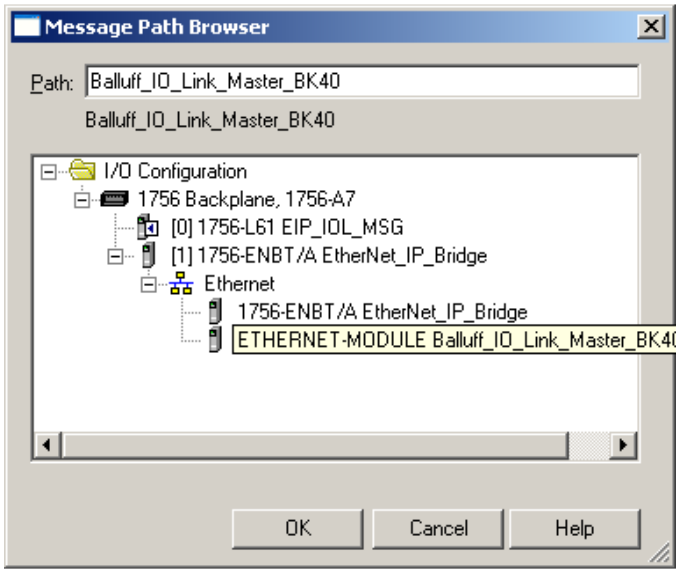
iii. For the Destination, tag BNI0048_BK40_P1_Parameter_WR must be selected. Since we are using 5 bytes of this tag as specified in the Source Length above, the tag selected will be BNI0048_BK40_P1_Parameter_WR[5].



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:



5.2.3.4 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 BNI0048. For this example, the Vendor Name will be read from the BNI0048. The Parameter Data for the BNI0048 shows the following:

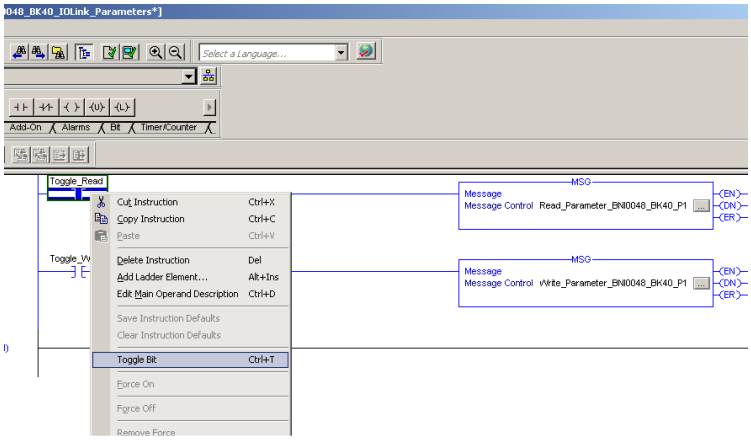
SPDU Index: 0x10 (10 Hex) – This is the Low Byte of the Index, there is no High Byte.
SPDU Sub-Index: 0

Using these values, the controller should read back the value “BALLUFF” and should be located in tag BNI_BK40_P1_Parameter_RD.

ii. Step 1: Enter the SPDU Index 10 Hex in tag BNI0048_BK40_P1_Parameter_WR[0]:

EIP_IOL_Mast_BK40_Port_3_IO_Data	{...}	{...}		IO_Link_Port_Data	
EIP_IOL_Mast_BK40_Port_4_IO_Data	{...}	{...}		IO_Link_Port_Data	
EIP_IOL_Mast_BK40_IO_Data	{...}	{...}		BNI004A_UDT_T...	
EIP_IOL_Mast_BK40_Port_1_Control	{...}	{...}		BNI0048_Rev_0_0	BNI IOL-302-S01-Z013-C01
BNI0048_BK40_Port_1_IO	{...}	{...}		BNI0048_UDT	
Read_Parameter_BNI0048_BK40_P1	{...}	{...}		MESSAGE	
Write_Parameter_BNI0048_BK40_P1	{...}	{...}		MESSAGE	
Toggle_Read	0		Decimal	BOOL	
Toggle_Write	0		Decimal	BOOL	
BNI0048_BK40_P1_Parameter_WR	{...}	{...}	Decimal	SINT[23]	
BNI0048_BK40_P1_Parameter_WR[0]	16#10		Hex	SINT	SPDU Index Low Byte
BNI0048_BK40_P1_Parameter_WR[1]	0		Decimal	SINT	SPDU Index High Byte
BNI0048_BK40_P1_Parameter_WR[2]	0		Decimal	SINT	SPDU Sub-Index
BNI0048_BK40_P1_Parameter_WR[3]	0		Decimal	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_WR[4]	0		Decimal	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_WR[5]	0		Decimal	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_WR[6]	0		Decimal	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_WR[7]	0		Decimal	SINT	Parameter Data

iii. Step 2: Toggle MSG Instruction Read_Parameters_BNI0048_BK40_P1



iv. Step 3: Read the parameter data in tag BNI0048_BK40_P1_Parameter_RD starting at Byte 0.

EIP_IOL_Mast_BK40_IO_Data	{...}	{...}		BNI004A_UDT_T...	
EIP_IOL_Mast_BK40_Port_1_Control	{...}	{...}		BNI0048_Rev_0_0	BNI IOL-302-S01-ZC
BNI0048_BK40_Port_1_IO	{...}	{...}		BNI0048_UDT	
Read_Parameter_BNI0048_BK40_P1	{...}	{...}		MESSAGE	
Write_Parameter_BNI0048_BK40_P1	{...}	{...}		MESSAGE	
Toggle_Read	0		Decimal	BOOL	
Toggle_Write	0		Decimal	BOOL	
BNI0048_BK40_P1_Parameter_WR	{...}	{...}	Decimal	SINT[23]	
BNI0048_BK40_P1_Parameter_RD	{...}	{...}	ASCII	SINT[23]	Parameter Data
BNI0048_BK40_P1_Parameter_RD[0]	'B'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[1]	'A'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[2]	'L'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[3]	'L'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[4]	'U'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[5]	'F'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[6]	'F'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[7]	'$00'		ASCII	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_RD[8]	'$00'		ASCII	SINT	Parameter Data

v. Writing parameters to an IO-Link device. Next we will demonstrate setting ports 4, 5, 6, and 7 of the BNI0048 to outputs and all other ports as inputs. Here are the parameter settings for Port Direction for the BNI0048:

Port direction

Byte 0								Byte 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Direction 7.0	Direction 6.0	Direction 5.0	Direction 4.0	Direction 3.0	Direction 2.0	Direction 1.0	Direction 0.0	Direction 7.1	Direction 6.1	Direction 5.1	Direction 4.1	Direction 3.1	Direction 2.1	Direction 1.1	Direction 0.1

Direction port (x)
x.0: Pin 4,
x.1: Pin 2

Direction port (x)
0: Input
1: Output

To set ports 4, 5, 6 and 7 to outputs (=1), the parameter values should be as follows:

Byte 0: 11110000 (F0 Hex)
Byte 1: 11110000 (F0 Hex)

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

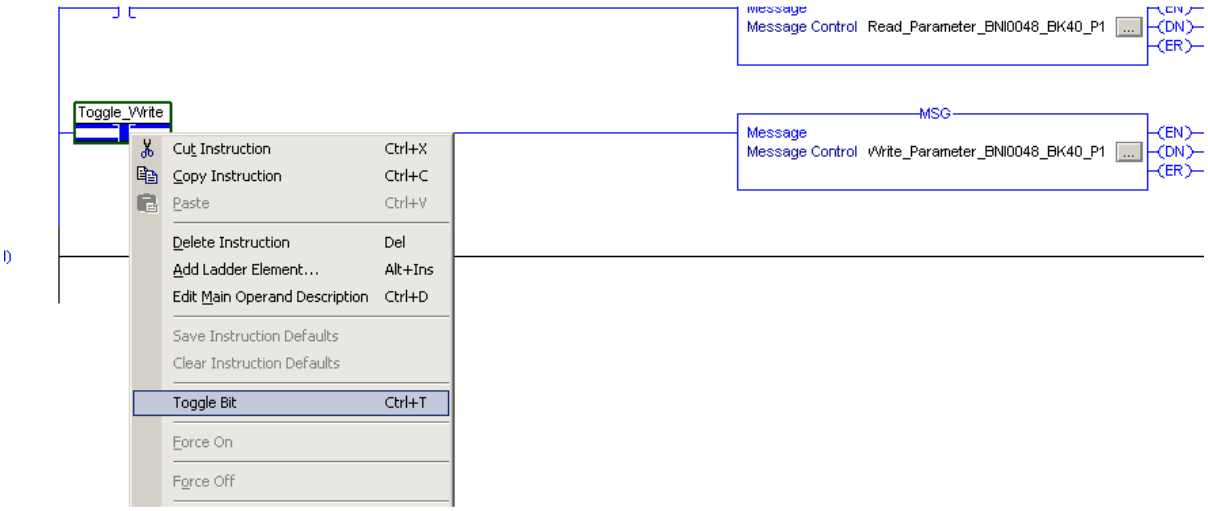
SPDU Index: 0x41 (41 Hex) – this is the Low Byte of the Index, there is no High Byte
SPDU Sub-Index: 0

vi. Step 4:

Enter the SPDU Index 41 Hex in tag BNI0048_BK40_P1_Parameter_WR[0]
Enter the SPDU Index 0 Hex in tag BNI0048_BK40_P1_Parameter_WR[1]
Enter the SPDU Sub-Index 0 Hex in tag BNI0048_BK40_P1_Parameter_WR[2]
Enter the Parameter value of F0 Hex in tag BNI0048_BK40_P1_Parameter_WR[3]
Enter the Parameter value of F0 Hex in tag BNI0048_BK40_P1_Parameter_WR[4]

Read_Parameter_BNI0048_BK40_P1	{...}	{...}		MESSAGE	
Write_Parameter_BNI0048_BK40_P1	{...}	{...}		MESSAGE	
Toggle_Read	0		Decimal	BOOL	
Toggle_Write	0		Decimal	BOOL	
BNI0048_BK40_P1_Parameter_WR	{...}	{...}	Hex	SINT[23]	
BNI0048_BK40_P1_Parameter_WR[0]	16#41		Hex	SINT	SPDU Index Low Byte
BNI0048_BK40_P1_Parameter_WR[1]	16#00		Hex	SINT	SPDU Index High Byte
BNI0048_BK40_P1_Parameter_WR[2]	16#00		Hex	SINT	SPDU Sub-Index
BNI0048_BK40_P1_Parameter_WR[3]	16#F0		Hex	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_WR[4]	16#F0		Hex	SINT	Parameter Data
BNI0048_BK40_P1_Parameter_WR[5]	16#00		Hex	SINT	Parameter Data

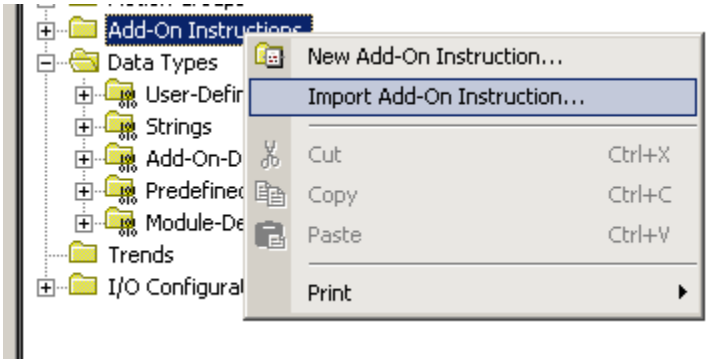
vii. Step 5: Toggle MSG Instruction Write_Parameters_BNI0048_BK40_P1.



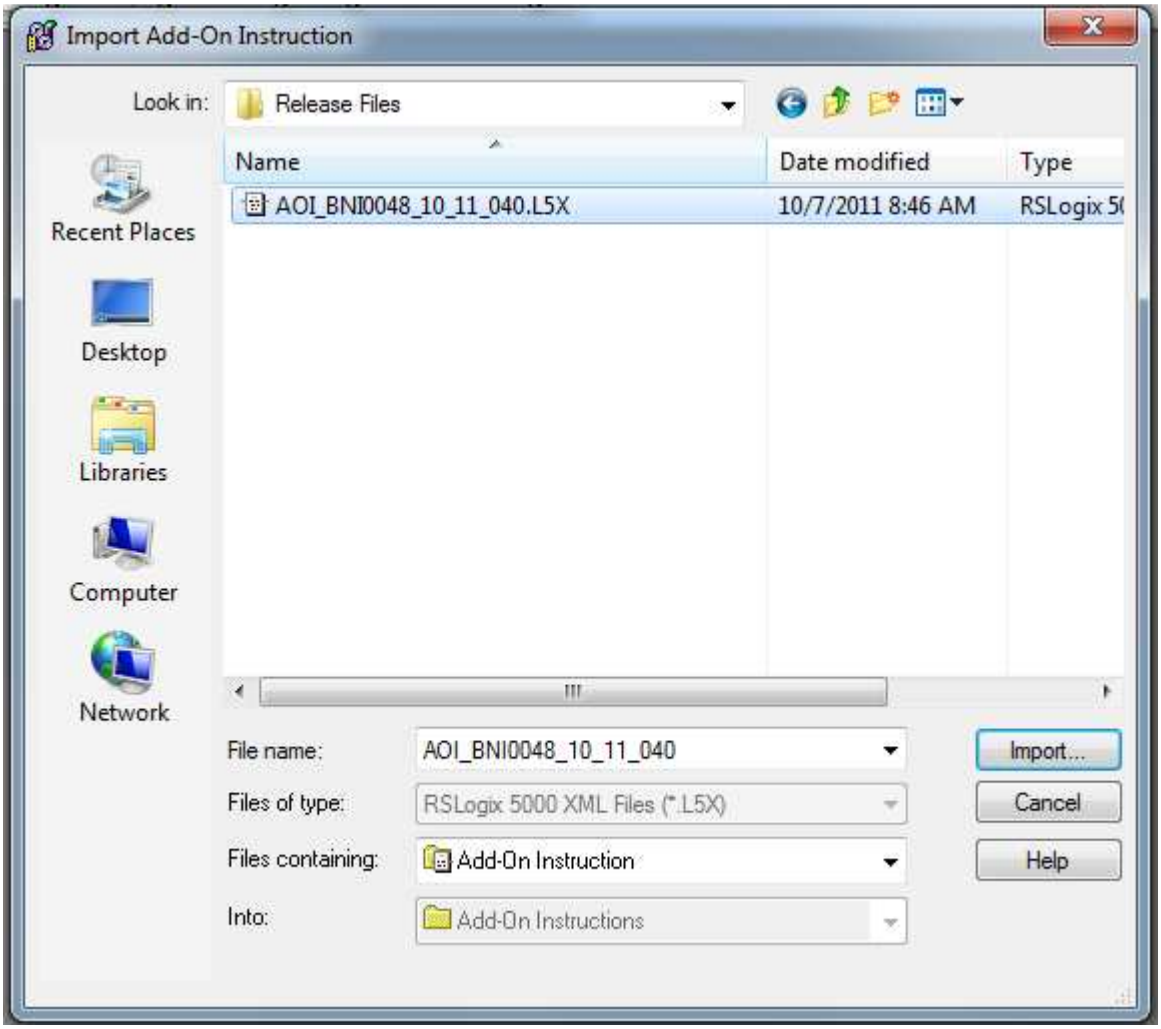
viii. Step 6: Observe the new configuration of the BNI0048. Ports 0, 1, 2, and 3 will function as inputs while ports 4, 5, 6 and 7 will function as outputs.

5.3 Import AOI

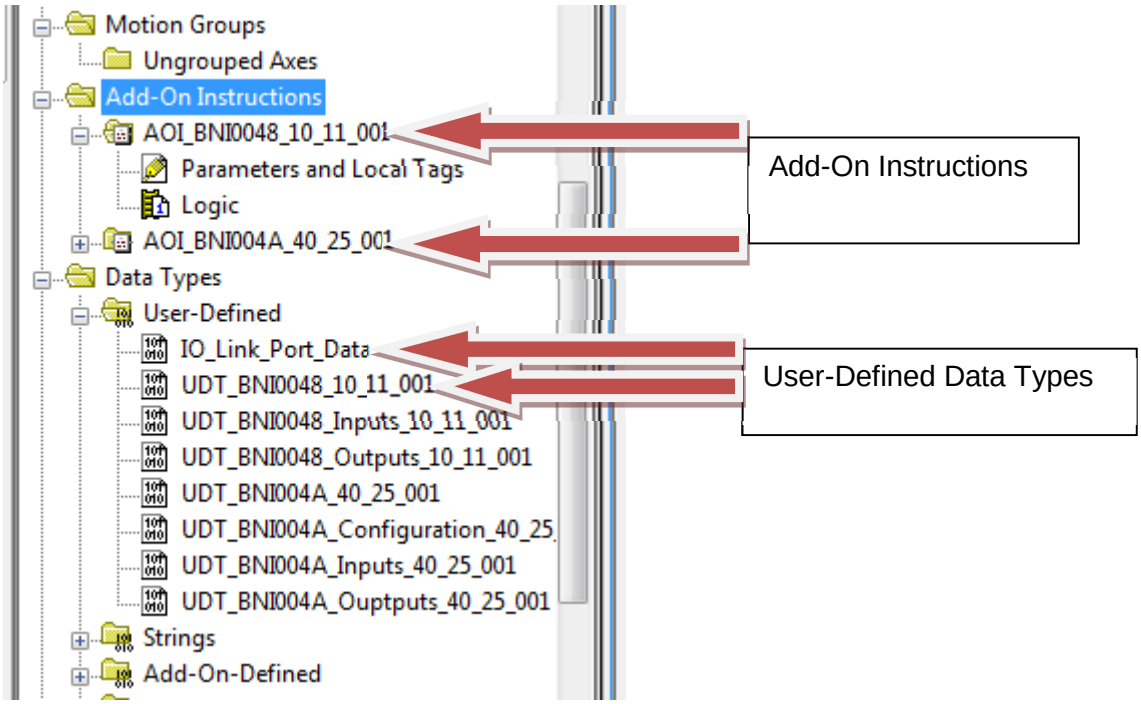
5.3.1 Right click on the Add-On Instructions element of the Controller Organizer to view a menu that includes “Import Add-On Instruction…” as shown below:



5.3.2 Locate your folder containing the downloaded Balluff AOI modules, highlight AOI_BNI0048_10_11_040.L5X and click *Import…* as shown below:

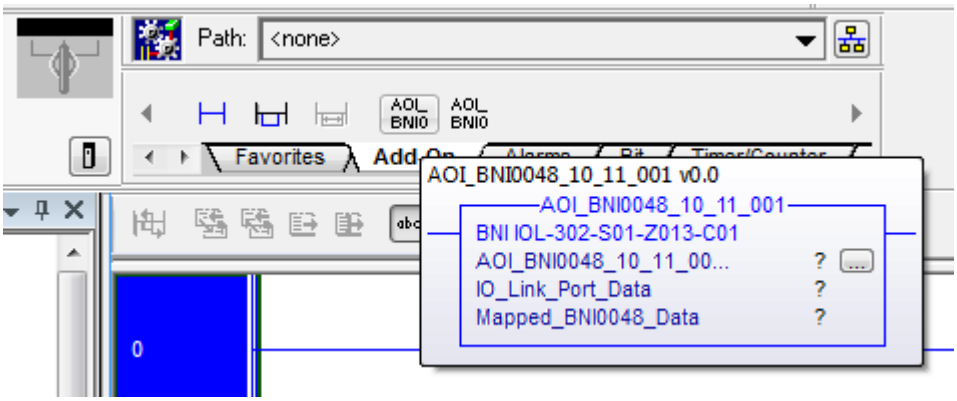


5.3.3 If the operation is successful, all UDTs associated with the AOI module will be imported into the project along with the AOI. This can be verified by checking the User-Defined element under Data Types in the Controller Organizer.

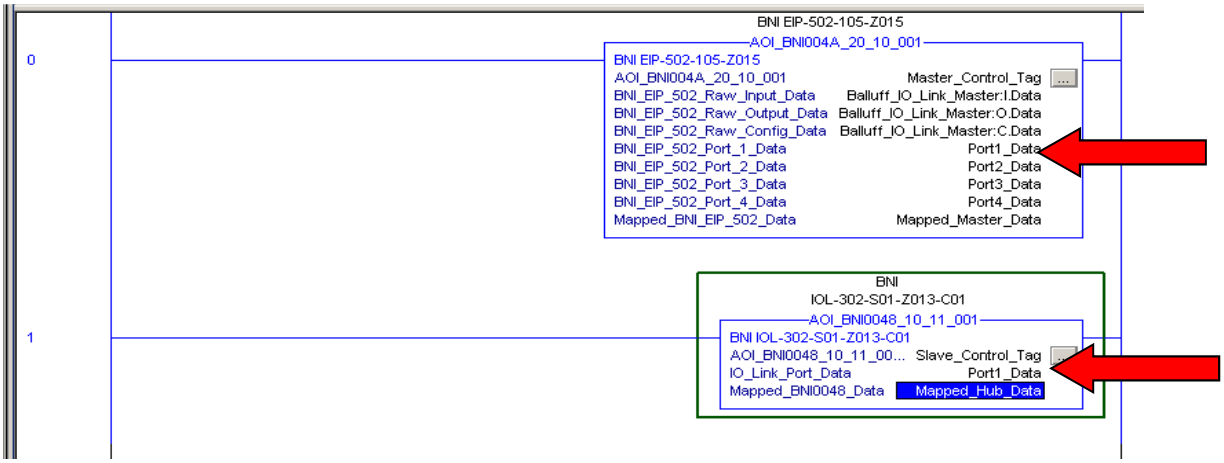


5.4 Create logic using RSLogix5000

5.4.1 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 with the hover window information displayed:



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



Note that in order for the AOI for the slave hub 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.

5.4.3 All parameters must be defined with tags of valid data type as defined by the AOI module.

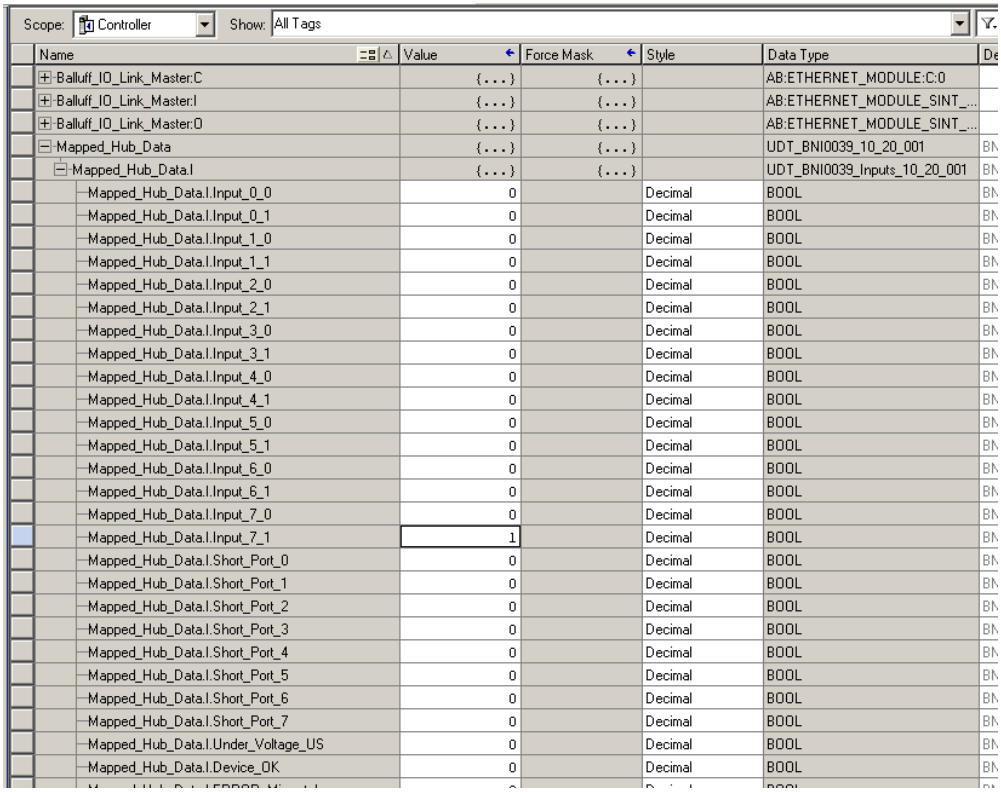
Parameter Name	Description	Data Type	Tag Name (User specific)
AOI_BNI0048_10_11_040	Unique Control Tag for AOI Module	UDT_BNI0048_10_11_040	Slave_Control_Tag
IO_Link_Port_Data	All I/O Data Associated with IO-Link Port 1	IO_Link_Port_Data	Port1_Data
Mapped_BNI0048_Data	All Unique data associated with BNI0048 module	UDT_BNI0048_10_11_040	Mapped_Hub_Data

6.0 Software Validation

6.1 If the preceding instructions (5.0 Instructions) 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 uses Port 7, Pin 2 although any input bit will suffice. The key point of this example is to ensure that the AOI bit corresponding to the selected input bit follows it. Place a device (discrete sensor or jumper wire) between pins 1 and 2 on Port 7 to cause this Input bit to transition to ON or “high”. This bit can be viewed in the Controller scope tags as shown below:



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:O	{...}	{...}		AB:ETHERNET_MODULE_SINT_...
Mapped_Hub_Data	{...}	{...}		UDT_BNI0039_10_20_001
Mapped_Hub_Data.I	{...}	{...}		UDT_BNI0039_Inputs_10_20_001
Mapped_Hub_Data.I.Input_0_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_0_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_1_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_1_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_2_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_2_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_3_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_3_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_4_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_4_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_5_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_5_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_6_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_6_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_7_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Input_7_1	1		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_0	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_1	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_2	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_3	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_4	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_5	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_6	0		Decimal	BOOL
Mapped_Hub_Data.I.Short_Port_7	0		Decimal	BOOL
Mapped_Hub_Data.I.Under_Voltage_US	0		Decimal	BOOL
Mapped_Hub_Data.I.Device_OK	0		Decimal	BOOL

Note that when the AOI module was used in the logic, the parameter Mapped_BNI0048_Data was assigned to a tag named “Mapped_Hub_Data”.

6.2.2 While this bit (Mapped_Hub_Data:I.Input_7_1) is ON or “high”, the corresponding LED on the I/O block should be lit. If this is not the case, please refer to Section 7.0 Troubleshooting.

6.3 All released Balluff AOI modules have been validated according to Balluff QF 4.4.12. A copy of this validation report can be obtained by contacting Balluff Technical Support at technicalsupport@balluff.com.

7.0 Troubleshooting Tips

7.1 If the user does not obtain the results described in Section 5.0 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

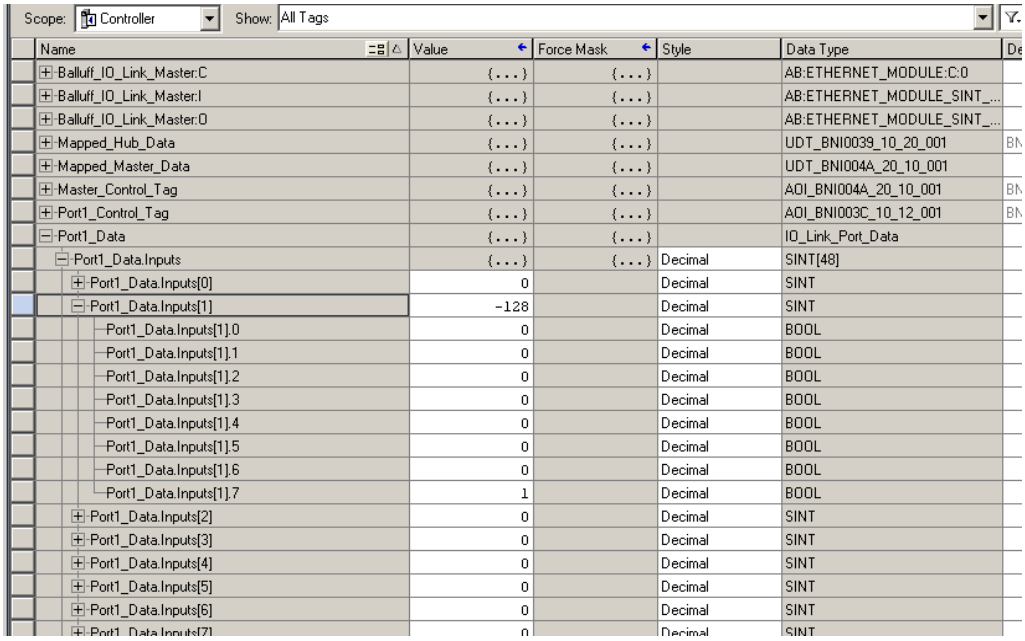
7.1.1.3 ODVA-compliant EtherNet/IP physical media

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

7.1.3 Revision levels

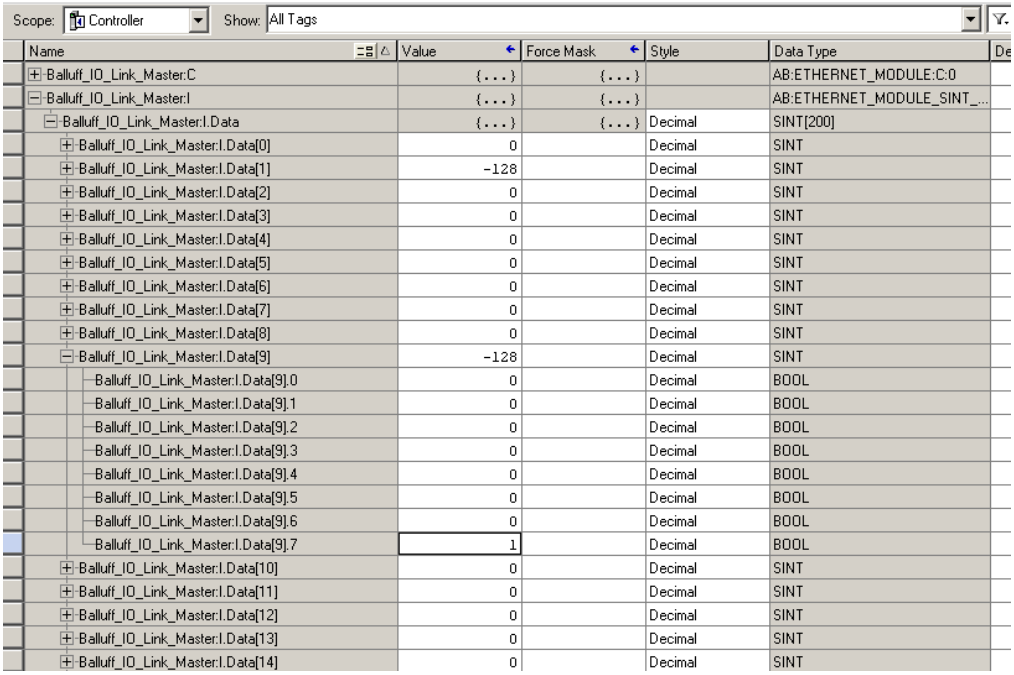
7.2 After confirming the revision levels of all system components, the next step should be to observe the raw data from the EtherNet/IP master in RSLogix5000. This can be accomplished by observing each step through the process of an input bit through the raw data into the AOI data buffers.

7.2.1 The initial step in confirming the proper operation of the controller-scoped input tag described in Section 6.2 (Port 7, Pin 2 of the BNI0048 IO-Link hub slave) is to observe the IO_Link_Port_Data for the BNI0048 while the input has transitioned to “high” or ON. If working properly, your results will be as shown here:



Please note that in this example, the BNI0048 hub 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 (Port 7, Pin 2), 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]	-128		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]	0		Decimal	SINT
Balluff_IO_Link_Master:I.Data[9]	-128		Decimal	SINT
Balluff_IO_Link_Master:I.Data[9].0	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].1	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].2	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].3	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].4	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].5	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].6	0		Decimal	BOOL
Balluff_IO_Link_Master:I.Data[9].7	1		Decimal	BOOL
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

Note that because the BNI0048 is connected to Port 1 of the IO-Link master, the indicate bit will be byte 9, bit7. If the slave is connected to a different port, the bit will be indicated in byte 57 (Port 2), byte 105 (Port 3), or byte 153 (Port 4).

7.3 If all system components are consistent with those described in Section 6.1 above and the raw data is being received properly in RSLogix5000 but the user does not view AOI data as described in Section 5.1 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 or hubs 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 Related Documents

8.1 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:

8.1.1 QF 4.4.11 – AOI Release Checklist

8.1.2 QF 4.4.12 – AOI Validation Report

8.1.3 WI 4.4.10 – AOI Development Revision Process

8.2 Other documents – Each Balluff IO-Link device that may be connected to this IO-Link Master device will have its own User Guides, both for hardware and for the AOI module software. These AOI modules and User Guides can be downloaded from Balluff's AOI website: www.Balluff.com/AOI.

9.0 References

9.1 None