MicroHAWK ID-40

Connectivity for MicroHAWK ID-40 and Rockwell CompactLogix Platform

Version 2.4
12/09/2016

1 Contents

1 Introduction

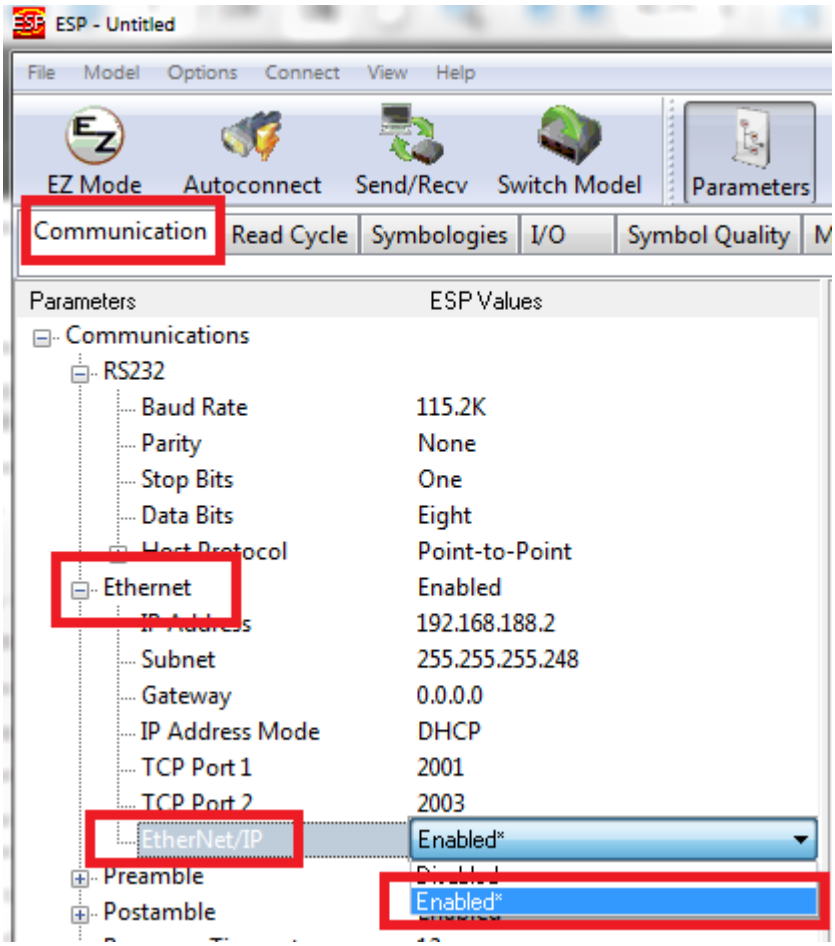
This guide explains how to setup the Microscan MicroHAWK Auto-ID product with the Rockwell CompactLogix platform. All files required for setup can be found on your MicroHAWK unit in the Microscan Connectivity under Rockwell CompactLogix. The files are also downloadable at: http://www.microscan.com/en-us/support/download-center

2 Protocol Switching in ESP and Weblink

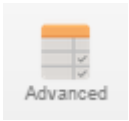
This section describes how to enable EtherNet/IP in ESP and Weblink.

2.1 ESP

Go to the communications tab in ESP and under Ethernet there will be a node called EtherNet/IP. To the right click the dropdown box and select **Enabled**.



2.2 Weblink

Go to the Application Settings button  in the upper right hand corner and select the Advanced settings button 

. In advanced settings select the communications tab under the Ethernet sections select Enabled in the dropdown box.

3 RSLogix Version Table

The table below illustrates how to setup the MicroHAWK with various version of RSLogix versions. If the block is red than the version of RSLogix does not support the setup process. If the block is green than the version supports the setup process. These setup processes are described in section 5 and 6 of this document.

3.1 RSLogix Version Table

MicroHAWK Input Assembly	RSLogix Version Table											
	RS Logix 5000 v16		RS Logix 5000 v17		RS Logix 5000 v18		RS Logix 5000 v19		RS Logix 5000 v20		RS Logix 5000 v21 and higher	
Small IO	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File
Big IO	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File
MXLSLC	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File
Input 1 Decode	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File
Input 4 Decode	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File
Input N Decode	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File	Generic Ethernet Module	EDS File

4 Using EtherNet/IP

This section provides information necessary for using the MicroHAWK in an EtherNet/IP environment.

Note:

- The units communication protocol must be enabled and set to EtherNet/IP enabled for the unit to begin using the EtherNet/IP protocol. Please follow the steps in Chapter 2 Using Protocol Switching in ESP and Weblink.

4.1 Overview

The EtherNet/IP interface will be identified as a Generic Device (0x00) for firmware releases under 1.2 and as a Generic Device keyable (0x2B) for firmware releases from 1.2 and beyond. The interface is designed to support implicit and explicit messaging using the EtherNet/IP protocol.

4.2 Necessary Tools

The following tools are helpful for configuring/debugging EtherNet/IP

- EtherNet/IP Messaging Tool—This can be a PLC or Software Tool but must be capable of sending and receiving Class3 explicit messages and establishing Class 1 connections. **EIPScan** is an example of such a tool.
- Terminal emulation or serial communication tool that can connect to a TCP socket, such as HyperTerminal.
- ESP – Microscan's Easy Setup Program. This tool has the ability to find Microscan products on the network, configure their IP address, then configure all application parameters.

4.3 EtherNet/IP Terms of Use

EtherNet/IP Technology is governed by the Open DeviceNet Vendor Association, Inc. (ODVA). Any person or entity that makes and sells products that implement EtherNet/IP Technology must agree to the Terms of Usage Agreement issued by ODVA. See http://www.odva.org for details.

4.4 Device Type

FIRMWARE VERSION	DEVICE TYPE
RELEASE < 1.1	0x00, Generic Device
RELEASE ≥ 1.2	0x2B, Generic Device (keyable)

4.5 EDS Files

MICROHAWK FIRMWARE VERSION	EDS FILE
RELEASE < 1.1	MicroHAWK_(35-9000034-10).eds
RELEASE ≥ 1.2	MicroHAWK_(35-9000034-11).eds

****Please note that the Device Type will change from 0x00 to 0x2B after FW Release 1.2**

4.6 Vendor ID

Microscan's Vendor ID is 1095

4.7 Product code

The Product code for the MicroHAWK is 3410

4.8 MicroHAWK EtherNet/IP Object Model

The MicroHAWK uses Class 1 connected messaging or Class 3 to communicate most data in eight different IO assemblies. The user chooses one of six input assemblies, and one of two output assemblies, for these connections

4.8.1 Connection Properties Supported

Class: 1 or 3
Trigger Mode: Cyclic and Change of State
Cyclic Rate: Greater than 20 ms recommended. 5 ms minimum.
Size: Fixed
Type: Point-to-Point (PLC OUT, O->T), Point-to-Point and Multicast (PLC IN, T->O)
Priority: Low, High, and Scheduled

4.9 Data Types

Microscan DataType	AB PLC	ODVA CIP EDS	Description
U8	SINT	USINT	Unsigned, 8 bit
U16	INT	UINT	Unsigned, 16 bit
U32	DINT	UDINT	Unsigned, 32 bit
STRING32	STRING	UDINT and BYTE[]	A 32 bit length field, followed by 8 bit ASCII characters

4.10 MicroHAWK Small Input Assembly (Instance Decimal: 100 Hex: 0x64) IN = MicroHAWK→PLC

This is a small, lightweight input assembly. Designed to hold 64 bytes of information in the decode data tag with minimal read cycle and device data. Below is a table showing the memory allocation for the assembly

NOTE: This input block must use the Output Legacy to function correctly

4.10.1 Table 3.11.1 Input Small Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
USER-DEFINED TAG ECHO	4
COMMAND ECHO	4
OUTPUT CONTROL ECHO	4
READ CYCLE SEQUENCE COUNTER	4
DECODE DATA LENGTH	4
DECODE DATA STRING	64

Total Size: 84 Bytes

4.10.2 User-Defined Tag Echo

These are a direct echo of the equivalent fields in the Output Legacy data block. They provide the PLC programmer with a method of verifying that the OUT data has been received by the MicroHAWK or any method the programmer wishes to use these 4 bytes of data.

4.10.3 Command Echo

These are a direct echo of the equivalent fields in the command field located in Output Legacy data block. This provides the PLC programmer with a method of verifying that the command data has been acknowledged by the MicroHAWK.

4.10.4 Output Control Status

Provides the PLC programmer with the current status of the external physical outputs for the MicroHAWK.

4.10.5 Read Cycle Sequence Counter

When this value changes, it indicates a new read cycle report is present. Read cycle report data is only valid when Sequence is not 0. Read cycle reports are only output during normal read cycles: continuous, serial, and triggered. Read cycle reports are not output during bar code configuration, read rate, auto-calibration, or ESP “Setup” mode.

4.10.6 Decode Length

The number of characters found in the decode string

4.10.7 Decode Data

Outputted decode data from the unit with one difference. Preamble and post amble symbols are not added.

4.10.8 Assembly Member Location

The following table displays the location of the members for the Input Small Assembly.

4.10.8.1 Memory Map Table 3.9.7.1.1

	Member	Data Type	Target	Bit Number	Style	Data Length	Byte Offset
	User Defined Tag Echo	**DINT**				**4 Bytes**	**0**
32 Bit Boundary	UserTag_1		User Defined Tag	0	BOOL	1 Bit	
	UserTag_2		User Defined Tag	1	BOOL	1 Bit	
	UserTag_3		User Defined Tag	2	BOOL	1 Bit	
	UserTag_4		User Defined Tag	3	BOOL	1 Bit	
	UserTag_5		User Defined Tag	4	BOOL	1 Bit	
	UserTag_6		User Defined Tag	5	BOOL	1 Bit	
	UserTag_7		User Defined Tag	6	BOOL	1 Bit	
	UserTag_8		User Defined Tag	7	BOOL	1 Bit	
	UserTag_9		User Defined Tag	8	BOOL	1 Bit	
	UserTag_10		User Defined Tag	9	BOOL	1 Bit	
	UserTag_11		User Defined Tag	10	BOOL	1 Bit	
	UserTag_12		User Defined Tag	11	BOOL	1 Bit	
	UserTag_13		User Defined Tag	12	BOOL	1 Bit	
	UserTag_14		User Defined Tag	13	BOOL	1 Bit	
	UserTag_15		User Defined Tag	14	BOOL	1 Bit	
	UserTag_16		User Defined Tag	15	BOOL	1 Bit	
	UserTag_17		User Defined Tag	16	BOOL	1 Bit	
	UserTag_18		User Defined Tag	17	BOOL	1 Bit	
	UserTag_19		User Defined Tag	18	BOOL	1 Bit	
	UserTag_20		User Defined Tag	19	BOOL	1 Bit	
	UserTag_21		User Defined Tag	20	BOOL	1 Bit	
	UserTag_22		User Defined Tag	21	BOOL	1 Bit	
	UserTag_23		User Defined Tag	22	BOOL	1 Bit	
	UserTag_24		User Defined Tag	23	BOOL	1 Bit	
	UserTag_25		User Defined Tag	24	BOOL	1 Bit	
	UserTag_26		User Defined Tag	25	BOOL	1 Bit	
	UserTag_27		User Defined Tag	26	BOOL	1 Bit	
	UserTag_28		User Defined Tag	27	BOOL	1 Bit	
	UserTag_29		User Defined Tag	28	BOOL	1 Bit	
	UserTag_30		User Defined Tag	29	BOOL	1 Bit	
	UserTag_31		User Defined Tag	30	BOOL	1 Bit	
	UserTag_32		User Defined Tag	31	BOOL	1 Bit	
	Command Echo	**DINT**				**4 Bytes**	**4**
	Trigger_Echo		Command Echo	0	BOOL	1 Bit	

32 Bit Boundary	New Master Echo		Command Echo	1	BOOL	1 Bit	
	Reserved for future use		Command Echo	2 - 7	BOOL	6 Bits	
	Disable Scanning Echo		Command Echo	8	BOOL	1 Bit	
	Reserved for future use		Command Echo	9 - 15	BOOL	7 Bits	
	Clear Read Cycle Report and Counters Echo		Command Echo	16	BOOL	1 Bit	
	Unlatch Outputs Echo		Command Echo	17	BOOL	1 Bit	
	Reserved for future use		Command Echo	18 - 31	BOOL	14 Bits	
	Output Control Echo	**DINT**				**4 Bytes**	**8**
32 Bit Boundary	Out1 Echo		External Output	0	BOOL	1 Bit	
	Out2 Echo		External Output	1	BOOL	1 Bit	
	Out3 Echo		External Output	2	BOOL	1 Bit	
	Reserved for future use		External Output	3 - 31	BOOL	29 Bits	
32 Bit	**Read Cycle Sequence count**	**DINT**	**Read Cycle Count**	**0-31**	**Decimal**	**4 Bytes**	**12**
32 Bit	**Decode Data Length**	**DINT**	**Decode Data Length**	**0 - 31**	**Decimal**	**4 Bytes**	**16**
32 Bit	**DecodeData**	**SINT[64]**		**0 - 512**	**ASCII**	**64 Bytes**	**20**

4.11 MicroHAWK Big Input Assembly (Instance Decimal: 101 Hex: 0x65) IN = MicroHAWK→PLC

The Big Input Assembly contains more device status information, and a longer bar code string, than the "Small IN Assembly 0x64". This assembly is designed to hold 128 bytes of information in the decode data tag and some additional ready cycle information.

NOTE: This input block must use the Output Legacy to function correctly

4.11.1 Table 3.11.1 Input Big Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
USER-DEFINED TAG ECHO	4
COMMAND ECHO	4
OUTPUT CONTROL ECHO	4
EXTERNAL INPUT STATUS	4
EXTERNAL OUTPUT STATUS	4
DEVICE STATUS	4
READ CYCLE SEQUENCE COUNTER	4
TRIGGER COUNT	4
DECODE/MATCH COUNT	4
MISMATCH COUNT	4
NOREAD COUNT	4
DECODE DATA LENGTH	4
DECODE DATA STRING	128

Total Size: 176 Bytes

4.11.2 User-Defined Tag Echo

These are a direct echo of the equivalent fields in the Output (Legacy) assembly (0xC6). They provide the PLC programmer with a method of verifying that the OUT data has been received by the MicroHAWK.

4.11.3 Command Echo

These are a direct echo of the equivalent fields in the Output (Legacy) assembly (0xC6). They provide the PLC programmer with a method of verifying that the OUT data has been received by the MicroHAWK.

4.11.4 Output Control Echo

These are a direct echo of the equivalent fields in the Output (Legacy) assembly (0xC6). They provide the PLC programmer with a method of verifying that the OUT data has been received by the MicroHAWK.

4.11.5 External Input Status

The current status of the physical input pins on the unit

4.11.5.1 External Input Status Bit Field

BIT	PIN NAME
0	Trigger
1	New Master
2-31	Reserved for future use

0 = No current sensed on input

1 = Current sensed on input

4.11.6 External Output Status

The current status of the physical output pins on the unit

BIT	PIN NAME
0	Output 1
1	Output 2
2	Output 3
3-31	Reserved for future use

0 = Output contact is open

1 = Output contact is closed

4.11.7 Device Status

Provides the current status of the unit. Below is the bit field table that defines each bit and the relationship to the unit's status

BIT	PIN NAME
0	Reserved
1	New Master Requested
2-7	Reserved for future use
8	Scanning Disabled
9-15	Reserved for future use
16	In read cycle
17	Actively Scanning

4.11.8 Read Cycle Sequence Counter

When this value changes, it indicates a new read cycle report is present. Read cycle report data is only valid when Sequence is not 0. Read cycle reports are only output during normal read cycles: continuous, serial, and triggered. Read cycle reports are not output during bar code configuration, read rate, auto-calibration, or ESP "Setup" mode.

4.11.9 Trigger Counter

The message displays the total number of triggers that have occurred since power-on or the last Trigger Counter Reset command

4.11.10 Decode/MatchCode Counter

The message displays either (1) the total number of good reads that match the master label or (2) the total number of good reads, or decodes. The count begins from the last power-on or Match Code/Good Read Counter Reset command. To count the good reads that match the master label, enable Match Code; to count good reads only, disable Match Code

4.11.11 Mismatch Counter

The message displays the total number of symbols successfully read that do not match the master label since power-on or the last Mismatch Counter command

4.11.12 NoRead Counter

The message displays the total number of noreads that have occurred since power-on or the last Noread Counter Reset command

4.11.13 Decode Length

The number of characters found in the decode string

4.11.14 Decode Data

Outputted decode data from the unit with one difference. Preamble and post amble symbols are not added.

4.11.15 Assembly Member Location

The following table displays the location of the members for the Input Big Assembly.

4.11.15.1 Memory Map Table 3.10.15.1

	Member	DataType	Target	Bit Number	Style	Data Length	Byte Offset
	User Defined Tag Echo	**DINT**				**4 Bytes**	**0**
32 Bit Boundary	UserTag_1		User Defined Tag	0	BOOL	1 Bit	
	UserTag_2		User Defined Tag	1	BOOL	1 Bit	
	UserTag_3		User Defined Tag	2	BOOL	1 Bit	
	UserTag_4		User Defined Tag	3	BOOL	1 Bit	
	UserTag_5		User Defined Tag	4	BOOL	1 Bit	
	UserTag_6		User Defined Tag	5	BOOL	1 Bit	
	UserTag_7		User Defined Tag	6	BOOL	1 Bit	
	UserTag_8		User Defined Tag	7	BOOL	1 Bit	
	UserTag_9		User Defined Tag	8	BOOL	1 Bit	
	UserTag_10		User Defined Tag	9	BOOL	1 Bit	
	UserTag_11		User Defined Tag	10	BOOL	1 Bit	
	UserTag_12		User Defined Tag	11	BOOL	1 Bit	
	UserTag_13		User Defined Tag	12	BOOL	1 Bit	
	UserTag_14		User Defined Tag	13	BOOL	1 Bit	
	UserTag_15		User Defined Tag	14	BOOL	1 Bit	
	UserTag_16		User Defined Tag	15	BOOL	1 Bit	
	UserTag_17		User Defined Tag	16	BOOL	1 Bit	
	UserTag_18		User Defined Tag	17	BOOL	1 Bit	
	UserTag_19		User Defined Tag	18	BOOL	1 Bit	
	UserTag_20		User Defined Tag	19	BOOL	1 Bit	
	UserTag_21		User Defined Tag	20	BOOL	1 Bit	
	UserTag_22		User Defined Tag	21	BOOL	1 Bit	
	UserTag_23		User Defined Tag	22	BOOL	1 Bit	
	UserTag_24		User Defined Tag	23	BOOL	1 Bit	

	UserTag_25		User Defined Tag	24	BOOL	1 Bit	
	UserTag_26		User Defined Tag	25	BOOL	1 Bit	
	UserTag_27		User Defined Tag	26	BOOL	1 Bit	
	UserTag_28		User Defined Tag	27	BOOL	1 Bit	
	UserTag_29		User Defined Tag	28	BOOL	1 Bit	
	UserTag_30		User Defined Tag	29	BOOL	1 Bit	
	UserTag_31		User Defined Tag	30	BOOL	1 Bit	
	UserTag_32		User Defined Tag	31	BOOL	1 Bit	
	Command Echo	**DINT**				**4 Bytes**	**4**
32 Bit Boundary	Trigger_Echo		Command Echo	0	BOOL	1 Bit	
	New Master Echo		Command Echo	1	BOOL	1 Bit	
	Reserved for future use		Command Echo	2 - 7	BOOL	6 Bits	
	Disable Scanning Echo		Command Echo	8	BOOL	1 Bit	
	Reserved for future use		Command Echo	9 - 15	BOOL	7 Bits	
	Clear Read Cycle Report and Counters Echo		Command Echo	16	BOOL	1 Bit	
	Unlatch Outputs Echo		Command Echo	17	BOOL	1 Bit	
	Reserved for future use		Command Echo	18 - 31	BOOL	14 Bits	
	Output Control Echo	**DINT**				**4 Bytes**	**8**
32 Bit Boundary	Out1 Echo		External Output	0	BOOL	1 Bit	
	Out2 Echo		External Output	1	BOOL	1 Bit	
	Out3 Echo		External Output	2	BOOL	1 Bit	
	Reserved for future use		External Output	3 - 31	BOOL	29 Bits	
	External Input Status (Physical Pint State)	**DINT**				**4 Bytes**	**12**
32 Bit Boundary	Trigger		External Input Status	0	BOOL	1 Bit	
	New Master		External Input Status	1	BOOL	1 Bit	
	Reserved for future use		External Input Status	2 - 31	BOOL	30 Bits	
	External Output Status (Physical Pint State)	**DINT**				**4 Bytes**	**16**
32 Bit Boundary	Out1		External Output Status	0	BOOL	1 Bit	
	Out2		External Output Status	1	BOOL	1 Bit	
	Out3		External Output Status	2	BOOL	1 Bit	
	Reserved for future use		External Output Status	3 - 31	BOOL	29 Bits	
	Device Status	**DINT**				**4 Bytes**	**20**
32 Bit Boundary	Reserved for future use		Device Status	0	BOOL	1 Bit	
	New Master Requested		Device Status	1	BOOL	1 Bit	
	Reserved for future use		Device Status	2 - 7	BOOL	6 Bits	
	Scanning Disabled		Device Status	8	BOOL	1 Bit	

	Reserved for future use		Device Status	9 - 15	BOOL	7 Bits	
	In Read Cycle		Device Status	16	BOOL	1 Bit	
	Actively Scanning		Device Status	17	BOOL	1 Bit	
	Reserved for future use		Device Status	18 - 31	BOOL	14 Bits	
32 Bit	**Read Cycle Sequence Counter**	**DINT**	**Read Cycle Sequence Counter**	**0 - 31**	**Decimal**	**4 Bytes**	**24**
32 Bit	**Trigger Count**	**DINT**	**Trigger Count**	**0 - 31**	**Decimal**	**4 Bytes**	**28**
32 Bit	**Decode/Match Count**	**DINT**	**Decode/Match Count**	**0 - 31**	**Decimal**	**4 Bytes**	**32**
32 Bit	**Mismatch Count**	**DINT**	**Mismatch Count**	**0 - 31**	**Decimal**	**4 Bytes**	**36**
32 Bit	**Mismatch Count**	**DINT**	**Mismatch Count**	**0 - 31**	**Decimal**	**4 Bytes**	**40**
32 Bit	**Decode Data Length**	**DINT**	**Decode Data Length**	**0 - 31**	**Decimal**	**4 Bytes**	**44**
32 Bit	**DecodeData**	**SINT[128]**		**0 - 1024**	**ASCII**	**128 Bytes**	**48**

4.12 MicroHAWK Input MXL/SLC Assembly (Instance Decimal: 102 Hex: 0x66) IN = MicroHAWK→PLC

This assembly is designed for controllers that cannot handle 500 bytes of input data. Designed to hold 184 bytes of information in the decode data tag, this data can be for 1 decoded string or a delimited number of decoded strings. In the case of a delimited number, the programmer shall parse the decoded data by reading the delimiter in ESP and/or issuing the K Command <K222?> to the command processor. This input data block also contains a Read Cycle Report and a Decode Cycle Report after an inspection. Details of these reports are described in detail later in the section.

NOTE: This input block must use the Output Assembly Premier to function correctly

4.12.1 Table 3.11.1 Input MXL/SLC Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
INFO BITS	1
DIAGNOSTIC SEQUENCE COUNT	1
CONFIGURATION SEQ. COUNT	1
RESERVED	1
DEVICE STATUS	4
FAULT	4
COUNTERS	24
READ CYCLE REPORT	8
DECODE CYCLE REPORT	16
DECODE LENGTH	4
DECODE DATA	184

Total Size: 248 Bytes

4.12.2 Input Assembly Description

This sub section will describe the tag and each field related for the Input Assembly.

4.12.3 Input Assembly Module Header

The following header is used at the beginning of the input (produced) assembly. Definitions for the members are included below.

4.12.3.1 Info Bits

Bit field of the input module status

INFO BIT FIELD	
BIT RUNMODE	0
BIT CONNECTIONFAULTED	1
BIT DIAGNOSTICACTIVE	2
RESERVED	3-7

4.12.3.1.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.12.3.1.2 Connection Faulted

Connection to the target is 0 = up and working, 1 = not connected. The module always returns a zero in this member. The controller overwrites the zero with a one when the connection is not up.

4.12.3.1.3 Diagnostic Active

0 = No diagnostics active, 1 = One or more diagnostic or prognostics thresholds reached

Note: “Diagnostic” means a detected condition that prevents the primary signal from propagating from a sensor to the controller, or from the controller to an actuator.

4.12.3.2 Diagnostic Sequence Count

SHORT NAME	SIZE
DIAGNOSTIC SEQUENCE COUNT	SINT

Increments for each time a distinct diagnostic condition is detected, and also each time a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.

4.12.3.3 Configuration Change Detection

When a change in the working set has been detected by the device this bit will be set to 1. This means that the configuration in the project no longer matches the configuration in the device.
Any forward open sets this value back to 0.

4.12.4 Device Status

This tag describes the current state of the device. In table 1.2.1 the bit field is mapped to allow the user to know what state the device is in.

4.12.4.1 Table 3.11.2 Device Status Bit Field

DEVICE STATUS

BIT FIELD	Status
0	Online
1	Trigger Acknowledge
2	Exposure Done
3	Decoding
4	Data Is Ready
5	Read Cycle Pass
6	Read Cycle Fail
7	General Fault
8	New match code acknowledged
9	Match Code Enabled
10	Image Sensor Calibrating
11	Image Sensor Calibration Complete
12	Training
13	Training Complete
14	Optimizing
15	Optimization Complete
16	AutoImage Photometry Enabled
17	AutoImage Photometry Complete
18	Output1 Status
19	Output2 Status
20	Output3 Status
21	Buffer Overflow
22-31	Reserved

4.12.4.2 Online

The units Current Read Cycle State

state

0 = Read cycle is disabled thus the unit is offline but the unit can receive commands. There is no data produced in the Input assembly and no data is consumed in the Output assembly when in this state.

1 = Read Cycle is enabled and the unit can be triggered and data is available for consumption and the unit will consume output data.

4.12.4.3 Trigger Acknowledged

This bit will go high when the unit has accepted the Trigger command in the Control tag. The user must lower the Trigger bit in the control tag in order for this bit to go back 0.

4.12.4.4 Exposure Done

When the image sensor exposure is complete this bit will go high and the user can move the object in the Field of view for the next image to be taken.

4.12.4.5 Decoding

When the unit is processing the image, this bit will be high. When the unit has completed the image process this bit will go low.

4.12.4.6 Data is Ready

The Read Cycle and Data Cycle Reports are ready for consumption when this bit goes high.

4.12.4.7 Read Cycle Pass

If the read cycle has passed all criteria, this bit will go high. It will go low when the ready begins to process the next image.

4.12.4.8 Ready Cycle Fail

If the read cycle has failed any of the criteria that was programmed, this bit will go high. It will go low when the ready begins to process the next image.

4.12.4.9 General Fault

When a fault occurs in the unit, this bit will go high. The user can reference the Fault Code tag for the error code and must remedy the problem. After the problem has been resolved the user can reset the fault in the Control tag in the Output assembly.

4.12.4.10 New Match Code Acknowledge

When active the unit has accepted the data read on the last trigger as the new match code. User shall set the Learn New Match Code bit in the Control tag to zero when this bit goes high.

4.12.4.11 Match Code Enabled

When this bit is 1 the unit will use the Match Code function to determine the Inspection Results.

4.12.4.12 Image Sensor Calibrating

The unit is undergoing a calibration on one or all of the following:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

When the unit has completed calibration this bit will be set to zero.

4.12.4.13 Image Sensor Calibration Complete

The unit has completed calibrating the image sensor for one or all of the following items:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

The user shall set the Control bit Calibration Image Sensor to zero if they have not done so already.

4.12.4.14 Training

When the unit is in the training process, this bit will be set to one. After the training process has completed, this bit will be set to zero.

4.12.4.15 Training Complete

After the unit has completed the training process, this bit will be set to one. If the user has set the Train Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

4.12.4.16 Optimizing

When the unit is optimizing this bit will be set to one. After optimization has completed, this bit will be set to zero.

4.12.4.17 Optimization Complete

After the unit has completed the optimization process, this bit will be set to one. If the user has set the Optimize Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

4.12.4.18 AutoImage Photometry Enabled

The unit will use AutoImage Photometry when trying to decode the symbol. Disabling this will mean the unit is using fixed values for Exposure, Gain and (if applicable) focal distance.

4.12.4.19 AutoImage Photometry Complete

This value will be set to one after the unit has completed an AutoImage Photometry calibration.

4.12.4.20 Output 1 Status

Current status of the physical output 1 signal

4.12.4.21 Output 2 Status

Current status of the physical output 2 signal

4.12.4.22 Output 3 Status

Current status of the physical output 3 signal

4.12.4.23 Buffer Overflow

When the data in the input buffer exceeds the buffer size (172 bytes) then this bit will go high alerting the user that the data is an incomplete segment.

4.12.5 Fault Code

This tag shall display the fault codes when the unit has faulted for any commands sent to it. When the user issues the Reset Fault in the Control Tag, this value will be set to zero.

4.12.6 Counters

Displays the counters stored in the unit upon power up or after a configuration change. These counters can be reset via the output command tag.

4.12.6.1 Table 3.11.3 Counters Input Array

COUNTERS	
NOREAD READCYCLE COUNTER	DINT
MISMATCH PER READCYCLE COUNTER	DINT
NOREAD COUNTER	DINT
TRIGGER COUNTER	DINT
MATCH CODE COUNTER	DINT
MISMATCH COUNTER	DINT

NOTE: Time starts over with power on but not with a <A> or <Z> type reset.

4.12.6.2 NoRead Cycle Counter

The message displays the total number of noread read cycles that have occurred since power-on or the last Noread Read cycle Counter Reset command

4.12.6.3 MisMatch Per ReadCycle Counter

The message displays the total number of mismatched code pre readcycle that have occurred since power-on or the last Mismatch per Readcycle Counter Reset command

4.12.6.4 NoRead Counter

The message displays the total number of noreads that have occurred since power-on or the last Noread Counter Reset command

4.12.6.5 Trigger Counter

The message displays the total number of triggers that have occurred since power-on or the last Trigger Counter Reset command

4.12.6.6 MatchCode Counter

The message displays either (1) the total number of good reads that match the master label or (2) the total number of good reads, or decodes. The count begins from the last power-on or Match Code/Good Read Counter Reset command. To count the good reads that match the master label, enable Match Code; to count good reads only, disable Match Code

4.12.6.7 Mismatch Counter

The message displays the total number of symbols successfully read that do not match the master label since power-on or the last Mismatch Counter command

4.12.7 Read Cycle Report

Information regarding the read cycle. Decode Data is referenced in the Decode Cycle Report

4.12.7.1 Table 3.11.4 Read Cycle Report Data

SHORT DESCRIPTION	SIZE
CAPTURE TIME	INT
TOTAL DECODE TIME	INT
TOTAL READCYCLE TIME	INT
RESERVED	INT

4.12.7.2 Capture Time

Total time it tool to capture the image

4.12.7.3 Total Decode Time

Total time spent decoding the symbol(s)

4.12.7.4 Total ReadCycle Time

Total Time Spent decoding the symbol which is the sum of the Capture, Decode and Overhead time.

4.12.8 Decode Cycle Report

Information on the decoded symbol

4.12.8.1 Table 3.11.5 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL

4.12.8.2 Decode Location Top

Defines the row position of the upper-left starting point of the image window.

4.12.8.3 Decode Location Left

Defines the column position of the upper-left starting point of the image window.

4.12.8.4 Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.12.8.5 Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.12.8.6 Code Type

Bit field of the symbol in that was decoded for this report

4.12.8.6.1 Table 3.11.5.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
COMPOSITE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

4.12.8.7 Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.12.8.8 Decode Length

The number of characters found in the decode string

4.12.9 Decode Length

The total number of characters contained in the Decode Data SINT array

4.12.10 Decode Data

Outputted decode data from the unit in ASCII with one difference. Preamble and post amble symbols are not added.

4.12.11 Assembly Member location

The following table is the Member location in the Input MXL/SLC assembly.

4.12.11.1 Memory Map Table 3.11.6

	Member	DataType	Target	BitNumber	Style	Data Length
	InfoBits	**SINT**				**1 Byte**
32 Bit	BIT RunMode	BIT	InfoBits	0	NA	1 Bit
	BIT ConnectionFaulted	BIT	InfoBits	1	NA	1 Bit
	BIT DiagnosticActive	BIT	InfoBits	2	NA	1 Bit
	Reserved	BIT	InfoBits	3 - 7	NA	5 Bits
	DiagnosticSequenceCount	**SINT**			**Decimal**	**1Byte**
	ConfigurationChangeDetect	**SINT**				**1 Byte**
	ConfigChangeDetect	BIT	ConfigurationChangeDetect	0	BOOL	1 Bit
	Reserved	BIT	ConfigurationChangeDetect	1 - 7	NA	7 Bits
	Reserved	**-**			**NA**	**1 Byte**
	DeviceStatus	**DINT**				**4 Bytes**
32 Bit Boundary	Online		DeviceStatus	0	BOOL	1 Bit
	TriggerAcknowledge		DeviceStatus	1	BOOL	1 Bit
	ExposureDone		DeviceStatus	2	BOOL	1 Bit
	Decoding		DeviceStatus	3	BOOL	1 Bit
	DataIsReady		DeviceStatus	4	BOOL	1 Bit
	ReadCyclePass		DeviceStatus	5	BOOL	1 Bit
	ReadCycleFail		DeviceStatus	6	BOOL	1 Bit
	GeneralFault		DeviceStatus	7	BOOL	1 Bit
	NewMatchCodeAcknowledged		DeviceStatus	8	BOOL	1 Bit
	MatchCodeEnabled		DeviceStatus	9	BOOL	1 Bit
	ImageSensorCalibrating		DeviceStatus	10	BOOL	1 Bit
	ImageSensorCalibrationComplete		DeviceStatus	11	BOOL	1 Bit
	Training		DeviceStatus	12	BOOL	1 Bit
	TrainingComplete		DeviceStatus	13	BOOL	1 Bit
	Optimizing		DeviceStatus	14	BOOL	1 Bit

	OptimizingComplete		DeviceStatus	15	BOOL	1 Bit
	AutoImagePhotometryEnabled		DeviceStatus	16	BOOL	1 Bit
	AutoImagePhotometryComplete		DeviceStatus	17	BOOL	1 Bit
	Output1Status		DeviceStatus	18	BOOL	1 Bit
	Output2Status		DeviceStatus	19	BOOL	1 Bit
	BufferOverflow		DeviceStatus	20	BOOL	1 Bit
	Reserved	-	DeviceStatus	21-31	NA	11 Bits
	Fault Code	**DINT**				**4 Bytes**
32 Bit Boundary	CommandErrorDetected		FaultCode	0	BOOL	1 Bit
	CommunicationError		FaultCode	1	BOOL	1 Bit
	FlashSectorUnprotectedFailure		FaultCode	2	BOOL	1 Bit
	HostPortBufferOverflow		FaultCode	3	BOOL	1 Bit
	Reserved		FaultCode	4 - 31	NA	28 Bits
	Counters	**DINT[6]**				**24 Bytes**
32 Bit	NoReadReadCycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes
32 Bit	MismatchPerReadcycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes
32 Bit	NoreadCounter	DINT	Counters	0 - 31	Decimal	4 Bytes
32 Bit	TriggerCounter	DINT	Counters	0 - 31	Decimal	4 Bytes
32 Bit	MatchCodeCounter	DINT	Counters	0 - 31	Decimal	4 Bytes
32 Bit	MismatchCounter	DINT	Counters	0 - 31	Decimal	4 Bytes
	ReadCycleReport	**INT[4]**				**8 Bytes**
32 Bit Boundary	CaptureTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes
	TotalDecodeTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes
32 Bit Boundary	TotalReadCycleTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes
	Reserved	INT	ReadCycleReport	0 - 15	NA	2 Bytes
	DecodeCycleReport					**16 Bytes**
32 Bit Boundary	DecodeLocationTop	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes
	DecodeLocationLeft	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes
32 Bit Boundary	DecodeLocationHeight	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes
	DecodeLocationWidth	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes
	CodeType (Subset)	DINT	DecodeCycleReport			4 Bytes
32 Bit Boundary	AztecCode		CodeType	0	BOOL	1 Bit
	MicroQRCode		CodeType	1	BOOL	1 Bit
	PostalCode		CodeType	2	BOOL	1 Bit
	Code39		CodeType	3	BOOL	1 Bit
	Codeabar		CodeType	4	BOOL	1 Bit
	Interleaved2of5		CodeType	5	BOOL	1 Bit
	UPCEAN		CodeType	6	BOOL	1 Bit
	Code128EAN128		CodeType	7	BOOL	1 Bit
	Code93		CodeType	8	BOOL	1 Bit

	PD417		CodeType	9	BOOL	1 Bit
	PharmaCode		CodeType	10	BOOL	1 Bit
	DataMatrix		CodeType	11	BOOL	1 Bit
	QRCode		CodeType	12	BOOL	1 Bit
	BC412		CodeType	13	BOOL	1 Bit
	RSS14		CodeType	14	BOOL	1 Bit
	RSS14LTD		CodeType	15	BOOL	1 Bit
	RSS14EXP		CodeType	16	BOOL	1 Bit
	MicroPDF		CodeType	17	BOOL	1 Bit
	PostalCode		CodeType	18	BOOL	1 Bit
	DotCode		CodeType	19	BOOL	1 Bit
	Reserved for future use		CodeType	20 - 31	BOOL	12 Bits
32 Bit	PixelsPerElement	REAL	DecodeCycleReport	0 - 31	Decimal	4 Bytes
32 Bit	**DecodeLength**	**DINT**		**0 - 31**	**Decimal**	**4 Bytes**
32 Bit	**DecodeData**	**SINT[184]**		**0 - 1472**	**ASCII**	**184 Bytes**

4.13 MicroHAWK Input 1 Decode Assembly (Instance Decimal: 103 Hex: 0x67) IN = MicroHAWK→PLC

Designed to hold 436 bytes of information in the decode data tag. This data can be for 1 decoded string or a delimited number of decoded strings. In the case of a delimited number, the programmer shall parse the decoded data by reading the delimiter in ESP and/or issuing the K Command <K222?> to the command processor.

This input data block also contains a Read Cycle Report and a Decode Cycle Report after an inspection. Details of these reports are described in detail later in the section.

NOTE: This input block must use the Output Premier to function correctly

4.13.1 Table 3.12.1 Input Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
INFO BITS	1
DIAGNOSTIC SEQUENCE COUNT	1
CONFIGURATION SEQ. COUNT	1
RESERVED	1
DEVICE STATUS	4
FAULT	4
COUNTERS	24
READ CYCLE REPORT	8
DECODE CYCLE REPORT	16
DECODE LENGTH	4
DECODE DATA	436

Total Size: 500 Bytes

4.13.2 Input Assembly Description

This sub section will describe the tag and each field related for the Input Assembly.

4.13.2.1 Input Assembly Module Header

The following header is used at the beginning of the input (produced) assembly. Definitions for the members are included below.

4.13.2.1.1 Info Bits

Bit field of the input module status

INFO BIT FIELD	
BIT RUNMODE	0
BIT CONNECTIONFAULTED	1
BIT DIAGNOSTICACTIVE	2
RESERVED	3-7

4.13.2.1.1.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.13.2.1.1.2 Connection Faulted

Connection to the target is 0 = up and working, 1 = not connected. The module always returns a zero in this member. The controller overwrites the zero with a one when the connection is not up.

4.13.2.1.1.3 Diagnostic Active

0 = No diagnostics active, 1 = One or more diagnostic or prognostics thresholds reached

Note: “Diagnostic” means a detected condition that prevents the primary signal from propagating from a sensor to the controller, or from the controller to an actuator.

4.13.2.1.2 Diagnostic Sequence Count

SHORT NAME	SIZE
DIAGNOSTIC SEQUENCE COUNT	SINT

Increments for each time a distinct diagnostic condition is detected, and also each time a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.

4.13.2.1.3 Configuration Change Detection

When a change in the working set has been detected by the device this bit will be set to 1. This means that the configuration in the project no longer matches the configuration in the device.
Any forward open sets this value back to 0.

4.13.2.2 Device Status

This tag describes the current state of the device. In table 1.2.1 the bit field is mapped to allow the user to know what state the device is in.

4.13.2.2.1 Table 3.12.2 Device Status Bit Field

DEVICE STATUS

BIT FIELD	Status
0	Online
1	Trigger Acknowledge
2	Exposure Done
3	Decoding
4	Data Is Ready
5	Read Cycle Pass
6	Read Cycle Fail
7	General Fault
8	New match code acknowledged
9	Match Code Enabled
10	Image Sensor Calibrating
11	Image Sensor Calibration Complete
12	Training
13	Training Complete
14	Optimizing
15	Optimization Complete
16	AutoImage Photometry Enabled
17	AutoImage Photometry Complete
18	Output1 Status
19	Output2 Status
20	Output3 Status
21	Buffer Overflow
22-31	Reserved

4.13.2.2.2 Online

The units Current Read Cycle State

state

0 = Read cycle is disabled thus the unit is offline but the unit can receive commands. There is no data produced in the Input assembly and no data is consumed in the Output assembly when in this state.

1 = Read Cycle is enabled and the unit can be triggered and data is available for consumption and the unit will consume output data.

4.13.2.2.3 Trigger Acknowledged

This bit will go high when the unit has accepted the Trigger command in the Control tag. The user must lower the Trigger bit in the control tag in order for this bit to go back 0.

4.13.2.2.4 Exposure Done

When the image sensor exposure is complete this bit will go high and the user can move the object in the Field of view for the next image to be taken.

4.13.2.2.5 Decoding

When the unit is processing the image, this bit will be high. When the unit has completed the image process this bit will go low.

4.13.2.2.6 Data is Ready

The Read Cycle and Data Cycle Reports are ready for consumption when this bit goes high.

4.13.2.2.7 Read Cycle Pass

If the read cycle has passed all criteria, this bit will go high. It will go low when the ready begins to process the next image.

4.13.2.2.8 Ready Cycle Fail

If the read cycle has failed any of the criteria that was programmed, this bit will go high. It will go low when the ready begins to process the next image.

4.13.2.2.9 General Fault

When a fault occurs in the unit, this bit will go high. The user can reference the Fault Code tag for the error code and must remedy the problem. After the problem has been resolved the user can reset the fault in the Control tag in the Output assembly.

*4.13.2.2.10*New Match Code Acknowledge

When active the unit has accepted the data read on the last trigger as the new match code. User shall set the Learn New Match Code bit in the Control tag to zero when this bit goes high.

*4.13.2.2.11*Match Code Enabled

When this bit is 1 the unit will use the Match Code function to determine the Inspection Results.

*4.13.2.2.12*Image Sensor Calibrating

The unit is undergoing a calibration on one or all of the following:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

When the unit has completed calibration this bit will be set to zero.

*4.13.2.2.13*Image Sensor Calibration Complete

The unit has completed calibrating the image sensor for one or all of the following items:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

The user shall set the Control bit Calibration Image Sensor to zero if they have not done so already.

*4.13.2.2.14*Training

When the unit is in the training process, this bit will be set to one. After the training process has completed, this bit will be set to zero.

*4.13.2.2.15*Training Complete

After the unit has completed the training process, this bit will be set to one. If the user has set the Train Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

*4.13.2.2.16*Optimizing

When the unit is optimizing this bit will be set to one. After optimization has completed, this bit will be set to zero.

*4.13.2.2.17*Optimization Complete

After the unit has completed the optimization process, this bit will be set to one. If the user has set the Optimize Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

*4.13.2.2.18*AutoImage Photometry Enabled

The unit will use AutoImage Photometry when trying to decode the symbol. Disabling this will mean the unit is using fixed values for Exposure, Gain and (if applicable) focal distance.

*4.13.2.2.19*AutoImage Photometry Complete

This value will be set to one after the unit has completed an AutoImage Photometry calibration.

*4.13.2.2.20*Output 1 Status

Current status of the physical output 1 signal

*4.13.2.2.21*Output 2 Status

Current status of the physical output 2 signal

*4.13.2.2.22*Output 3 Status

Current status of the physical output 3 signal

*4.13.2.2.23*Buffer Overflow

When the data in the input buffer exceeds the buffer size (444 bytes) then this bit will go high alerting the user that the data is an incomplete segment.

4.13.2.3 Fault Code

This tag shall display the fault codes when the unit has faulted for any commands sent to it. When the user issues the Reset Fault in the Control Tag, this value will be set to zero.

4.13.2.3.1 Table 3.12.3 Bit Field Layout

COUNTERS	
COMMAND ERROR DETECTED	0
COMMUNICATION ERROR	1
FLASH SECTOR UNPROTECTED FAILURE	2
HOST PORT BUFFER OVERFLOW	3
RESERVED	4-31

4.13.2.4 Counters

Displays the counters stored in the unit upon power up or after a configuration change. These counters can be reset via the output command tag.

4.13.2.4.1 Table 3.12.4 Counters Input Array

COUNTERS	
NOREAD READCYCLE COUNTER	DINT
MISMATCH PER READCYCLE COUNTER	DINT
NOREAD COUNTER	DINT
TRIGGER COUNTER	DINT
MATCH CODE COUNTER	DINT
MISMATCH COUNTER	DINT

NOTE: Time starts over with power on but not with a <A> or <Z> type reset.

4.13.2.4.2 NoRead Cycle Counter

The message displays the total number of noread read cycles that have occurred since power-on or the last Noread Read cycle Counter Reset command

4.13.2.4.3 MisMatch Per ReadCycle Counter

The message displays the total number of mismatched code pre readcycle that have occurred since power-on or the last Mismatch per Readcycle Counter Reset command

4.13.2.4.4 NoRead Counter

The message displays the total number of noreads that have occurred since power-on or the last Noread Counter Reset command

4.13.2.4.5 Trigger Counter

The message displays the total number of triggers that have occurred since power-on or the last Trigger Counter Reset command

4.13.2.4.6 MatchCode Counter

The message displays either (1) the total number of good reads that match the master label or (2) the total number of good reads, or decodes. The count begins from the last power-on or Match Code/Good

Read Counter Reset command. To count the good reads that match the master label, enable Match Code; to count good reads only, disable Match Code

4.13.2.4.7 Mismatch Counter

The message displays the total number of symbols successfully read that do not match the master label since power-on or the last Mismatch Counter command

4.13.2.5 Read Cycle Report

Information regarding the read cycle. Decode Data is referenced in the Decode Cycle Report

4.13.2.5.1 Table 3.12.5 Read Cycle Report Data

SHORT DESCRIPTION	SIZE
CAPTURE TIME	INT
TOTAL DECODE TIME	INT
TOTAL READCYCLE TIME	INT
RESERVED	INT

4.13.2.5.2 Capture Time

Total time it tool to capture the image

4.13.2.5.3 Total Decode Time

Total time spent decoding the symbol(s)

4.13.2.5.4 Total ReadCycle Time

Total Time Spent decoding the symbol which is the sum of the Capture, Decode and Overhead time.

4.13.2.6 Decode Cycle Report

Information on the decoded symbol

4.13.2.6.1 Table 3.12.6 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL

4.13.2.6.2 Decode Location Top

Defines the row position of the upper-left starting point of the image window.

4.13.2.6.3 Decode Location Left

Defines the column position of the upper-left starting point of the image window.

4.13.2.6.4 Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.13.2.6.5 Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.13.2.6.6 Code Type

Bit field of the symbol in that was decoded for this report

4.13.2.6.6.1 Table 3.12.6.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
POSTAL CODE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

4.13.2.6.7 Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.13.2.7 Decode Length

The number of characters found in the decode string

4.13.2.8 Decode Data

Outputted decode data from the unit in ASCII with one difference. Preamble and post amble symbols are not added.

4.13.3 Assembly Member Location

The following table is the Member location in the Input 1 Decode assembly.

	Member	Data Type	Target	Bit Number	Style	Data Length	Byte Offset
	InfoBits	**SINT**				**1 Byte**	**0**
32 Bit	BIT RunMode	BIT	InfoBits	0	NA	1 Bit	
	BIT ConnectionFaulted	BIT	InfoBits	1	NA	1 Bit	
	BIT DiagnosticActive	BIT	InfoBits	2	NA	1 Bit	
	Reserved	BIT	InfoBits	3 - 7	NA	5 Bits	
	DiagnosticSequenceCount	**SINT**			**Decimal**	**1Byte**	**1**
	ConfigurationChangeDetect	**SINT**				**1 Byte**	
	ConfigChangeDetect	BIT	ConfigurationChangeDetect	0	BOOL	1 Bit	
	Reserved	BIT	ConfigurationChangeDetect	1 - 7	NA	7 Bits	
	Reserved	**-**			**NA**	**1 Byte**	**3**
	DeviceStatus	**DINT**				**4 Bytes**	**4**
32 Bit Boundary	Online		DeviceStatus	0	BOOL	1 Bit	
	TriggerAcknowledge		DeviceStatus	1	BOOL	1 Bit	
	ExposureDone		DeviceStatus	2	BOOL	1 Bit	
	Decoding		DeviceStatus	3	BOOL	1 Bit	
	DataIsReady		DeviceStatus	4	BOOL	1 Bit	
	ReadCyclePass		DeviceStatus	5	BOOL	1 Bit	
	ReadCycleFail		DeviceStatus	6	BOOL	1 Bit	
	GeneralFault		DeviceStatus	7	BOOL	1 Bit	
	NewMatchCodeAcknowledged		DeviceStatus	8	BOOL	1 Bit	
	MatchCodeEnabled		DeviceStatus	9	BOOL	1 Bit	
	ImageSensorCalibrating		DeviceStatus	10	BOOL	1 Bit	
	ImageSensorCalibrationComplete		DeviceStatus	11	BOOL	1 Bit	
	Training		DeviceStatus	12	BOOL	1 Bit	
	TrainingComplete		DeviceStatus	13	BOOL	1 Bit	
	Optimizing		DeviceStatus	14	BOOL	1 Bit	
	OptimizingComplete		DeviceStatus	15	BOOL	1 Bit	

	AutoImagePhotometryEnabled		DeviceStatus	16	BOOL	1 Bit	
	AutoImagePhotometryComplete		DeviceStatus	17	BOOL	1 Bit	
	Output1Status		DeviceStatus	18	BOOL	1 Bit	
	Output2Status		DeviceStatus	19	BOOL	1 Bit	
	BufferOverflow		DeviceStatus	20	BOOL	1 Bit	
	Reserved	-	DeviceStatus	21-31	NA	11 Bits	
	Fault Code	**DINT**				**4 Bytes**	
32 Bit Boundary	CommandErrorDetected		FaultCode	0	BOOL	1 Bit	**8**
	CommunicationError		FaultCode	1	BOOL	1 Bit	
	FlashSectorUnprotectedFailure		FaultCode	2	BOOL	1 Bit	
	HostPortBufferOverflow		FaultCode	3	BOOL	1 Bit	
	Reserved		FaultCode	4 - 31	NA	28 Bits	
	Counters	**DINT[6]**				**24 Bytes**	
32 Bit	NoReadReadCycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**12**
32 Bit	MismatchPerReadcycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**16**
32 Bit	NoreadCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**20**
32 Bit	TriggerCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**24**
32 Bit	MatchCodeCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**28**
32 Bit	MismatchCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**32**
	ReadCycleReport	**INT[4]**				**8 Bytes**	
32 Bit Boundary	CaptureTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	**36**
	TotalDecodeTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	**38**
32 Bit Boundary	TotalReadCycleTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	**40**
	Reserved	INT	ReadCycleReport	0 - 15	NA	2 Bytes	**42**
	DecodeCycleReport					**16 Bytes**	
32 Bit Boundary	DecodeLocationTop	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**44**
	DecodeLocationLeft	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**46**
32 Bit Boundary	DecodeLocationHeight	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**48**
	DecodeLocationWidth	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**50**
	CodeType (Subset)	DINT	DecodeCycleReport			4 Bytes	
	AztecCode		CodeType	0	BOOL	1 Bit	**52**
	MicroQRCode		CodeType	1	BOOL	1 Bit	

32 Bit Boundary	PostalCode		CodeType	2	BOOL	1 Bit	
	Code39		CodeType	3	BOOL	1 Bit	
	Codeabar		CodeType	4	BOOL	1 Bit	
	Interleaved2of5		CodeType	5	BOOL	1 Bit	
	UPCEAN		CodeType	6	BOOL	1 Bit	
	Code128EAN128		CodeType	7	BOOL	1 Bit	
	Code93		CodeType	8	BOOL	1 Bit	
	PD417		CodeType	9	BOOL	1 Bit	
	PharmaCode		CodeType	10	BOOL	1 Bit	
	DataMatrix		CodeType	11	BOOL	1 Bit	
	QRCode		CodeType	12	BOOL	1 Bit	
	BC412		CodeType	13	BOOL	1 Bit	
	RSS14		CodeType	14	BOOL	1 Bit	
	RSS14LTD		CodeType	15	BOOL	1 Bit	
	RSS14EXP		CodeType	16	BOOL	1 Bit	
	MicroPDF		CodeType	17	BOOL	1 Bit	
	PostalCode		CodeType	18	BOOL	1 Bit	
	DotCode		CodeType	19	BOOL	1 Bit	
	Reserved for future use		CodeType	20 - 31	BOOL	12 Bits	
32 Bit	PixelsPerElement	REAL	DecodeCycleReport	0 - 31	Decimal	4 Bytes	**56**
32 Bit	**DecodeLength**	**DINT**		**0 - 31**	**Decimal**	**4 Bytes**	**60**
32 Bit	**DecodeData**	**SINT[184]**		**0 - 1472**	**ASCII**	**184 Bytes**	**64**

4.14 MicroHAWK Input 4 Decode Assembly (Instance Decimal: 104 Hex: 0x68) IN = MicroHAWK→PLC

Designed to hold 4 decoded symbols with decode cycle reports. The read cycle report contains data for the entire inspection while the decode # cycle report will contain data regarding the individual decoded symbols. Please note that decode symbol 1 is 160 bytes long while 2 through 4 are 72 bytes long. The unit will automatically place the largest decode symbol into Decode 1 Cycle Report and Decode 1 Data. The remaining will be placed in the remaining tags. If no data is found than the decode cycle report and the decode data will be null.

***NOTE[1]: The inspection will not need to have 4 decode symbols to use this input assembly. ***

NOTE[2]: This input block must use the Output Premier to function correctly

4.14.1 Table 3.13.1 Input Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
INFO BITS	1
DIAGNOSTIC SEQUENCE COUNT	1
CONFIGURATION SEQ. COUNT	1
RESERVED	1
DEVICE STATUS	4
FAULT	4
COUNTERS	24
READ CYCLE REPORT	8
DECODE 1 CYCLE REPORT	16
DECODE 1 LENGTH	4
DECODE 1 DATA	160
DECODE 2 CYCLE REPORT	16
DECODE 2 LENGTH	4
DECODE 2 DATA	72
DECODE 3 CYCLE REPORT	16
DECODE 3 LENGTH	4
DECODE 3 DATA	72
DECODE 4 CYCLE REPORT	16
DECODE 4 LENGTH	4
DECODE 4 DATA	72

Total Size: 500 Bytes

4.14.2 Input Assembly Description

This sub section will describe the tag and each field related for the Input Assembly.

4.14.2.1 Input Assembly Module Header

The following header is used at the beginning of the input (produced) assembly. Definitions for the members are included below.

4.14.2.1.1 Info Bits

Bit field of the input module status

INFO BIT FIELD	
BIT RUNMODE	0
BIT CONNECTIONFAULTED	1
BIT DIAGNOSTICACTIVE	2
RESERVED	3-7

4.14.2.1.1.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.14.2.1.1.2 Connection Faulted

Connection to the target is 0 = up and working, 1 = not connected. The module always returns a zero in this member. The controller overwrites the zero with a one when the connection is not up.

4.14.2.1.1.3 Diagnostic Active

0 = No diagnostics active, 1 = One or more diagnostic or prognostics thresholds reached

Note: “Diagnostic” means a detected condition that prevents the primary signal from propagating from a sensor to the controller, or from the controller to an actuator.

4.14.2.1.2 Diagnostic Sequence Count

SHORT NAME	SIZE
DIAGNOSTIC SEQUENCE COUNT	SINT

Increments for each time a distinct diagnostic condition is detected, and also each time a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.

4.14.2.1.3 Configuration Change Detection

When a change in the working set has been detected by the device this bit will be set to 1. This means that the configuration in the project no longer matches the configuration in the device.
Any forward open sets this value back to 0.

4.14.2.2 Device Status

This tag describes the current state of the device. In table 1.2.1 the bit field is mapped to allow the user to know what state the device is in.

4.14.2.2.1 Table 3.13.2 Device Status Bit Field

DEVICE STATUS

BIT FIELD	Status
0	Online
1	Trigger Acknowledge
2	Exposure Done
3	Decoding
4	Data Is Ready
5	Read Cycle Pass
6	Read Cycle Fail
7	General Fault
8	New match code acknowledged
9	Match Code Enabled
10	Image Sensor Calibrating
11	Image Sensor Calibration Complete
12	Training
13	Training Complete
14	Optimizing
15	Optimization Complete
16	AutoImage Photometry Enabled
17	AutoImage Photometry Complete
18	Output1 Status
19	Output2 Status
20	Output3 Status
21	Buffer Overflow
22-31	Reserved

4.14.2.2.2 Online

The units Current Read Cycle State

state

0 = Read cycle is disabled thus the unit is offline but the unit can receive commands. There is no data produced in the Input assembly and no data is consumed in the Output assembly when in this state.

1 = Read Cycle is enabled and the unit can be triggered and data is available for consumption and the unit will consume output data.

4.14.2.2.3 Trigger Acknowledged

This bit will go high when the unit has accepted the Trigger command in the Control tag. The user must lower the Trigger bit in the control tag in order for this bit to go back 0.

4.14.2.2.4 Exposure Done

When the image sensor exposure is complete this bit will go high and the user can move the object in the Field of view for the next image to be taken.

4.14.2.2.5 Decoding

When the unit is processing the image, this bit will be high. When the unit has completed the image process this bit will go low.

4.14.2.2.6 Data is Ready

The Read Cycle and Data Cycle Reports are ready for consumption when this bit goes high.

4.14.2.2.7 Read Cycle Pass

If the read cycle has passed all criteria, this bit will go high. It will go low when the ready begins to process the next image.

4.14.2.2.8 Ready Cycle Fail

If the read cycle has failed any of the criteria that was programmed, this bit will go high. It will go low when the ready begins to process the next image.

4.14.2.2.9 General Fault

When a fault occurs in the unit, this bit will go high. The user can reference the Fault Code tag for the error code and must remedy the problem. After the problem has been resolved the user can reset the fault in the Control tag in the Output assembly.

*4.14.2.2.10*New Match Code Acknowledge

When active the unit has accepted the data read on the last trigger as the new match code. User shall set the Learn New Match Code bit in the Control tag to zero when this bit goes high.

*4.14.2.2.11*Match Code Enabled

When this bit is 1 the unit will use the Match Code function to determine the Inspection Results.

*4.14.2.2.12*Image Sensor Calibrating

The unit is undergoing a calibration on one or all of the following:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

When the unit has completed calibration this bit will be set to zero.

*4.14.2.2.13*Image Sensor Calibration Complete

The unit has completed calibrating the image sensor for one or all of the following items:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

The user shall set the Control bit Calibration Image Sensor to zero if they have not done so already.

*4.14.2.2.14*Training

When the unit is in the training process, this bit will be set to one. After the training process has completed, this bit will be set to zero.

*4.14.2.2.15*Training Complete

After the unit has completed the training process, this bit will be set to one. If the user has set the Train Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

*4.14.2.2.16*Optimizing

When the unit is optimizing this bit will be set to one. After optimization has completed, this bit will be set to zero.

*4.14.2.2.17*Optimization Complete

After the unit has completed the optimization process, this bit will be set to one. If the user has set the Optimize Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

*4.14.2.2.18*AutoImage Photometry Enabled

The unit will use AutoImage Photometry when trying to decode the symbol. Disabling this will mean the unit is using fixed values for Exposure, Gain and (if applicable) focal distance.

*4.14.2.2.19*AutoImage Photometry Complete

This value will be set to one after the unit has completed an AutoImage Photometry calibration.

*4.14.2.2.20*Output 1 Status

Current status of the physical output 1 signal

*4.14.2.2.21*Output 2 Status

Current status of the physical output 2 signal

*4.14.2.2.22*Output 3 Status

Current status of the physical output 3 signal

*4.14.2.2.23*Buffer Overflow

When the data in the input buffer exceeds the buffer size (444 bytes) then this bit will go high alerting the user that the data is an incomplete segment.

4.14.2.3 Fault Code

This tag shall display the fault codes when the unit has faulted for any commands sent to it. When the user issues the Reset Fault in the Control Tag, this value will be set to zero.

4.14.2.3.1 Table 3.13.3 Bit Field Layout

COUNTERS	
COMMAND ERROR DETECTED	0
COMMUNICATION ERROR	1
FLASH SECTOR UNPROTECTED FAILURE	2
HOST PORT BUFFER OVERFLOW	3
RESERVED	4-31

4.14.2.4 Counters

Displays the counters stored in the unit upon power up or after a configuration change. These counters can be reset via the output command tag.

4.14.2.4.1 Table 3.13.4 Counters Input Array

COUNTERS	
NOREAD READCYCLE COUNTER	DINT
MISMATCH PER READCYCLE COUNTER	DINT
NOREAD COUNTER	DINT
TRIGGER COUNTER	DINT
MATCH CODE COUNTER	DINT
MISMATCH COUNTER	DINT

NOTE: Time starts over with power on but not with a <A> or <Z> type reset.

4.14.2.4.2 NoRead Cycle Counter

The message displays the total number of noread read cycles that have occurred since power-on or the last Noread Read cycle Counter Reset command

4.14.2.4.3 MisMatch Per ReadCycle Counter

The message displays the total number of mismatched code pre readcycle that have occurred since power-on or the last Mismatch per Readcycle Counter Reset command

4.14.2.4.4 NoRead Counter

The message displays the total number of noreads that have occurred since power-on or the last Noread Counter Reset command

4.14.2.4.5 Trigger Counter

The message displays the total number of triggers that have occurred since power-on or the last Trigger Counter Reset command

4.14.2.4.6 MatchCode Counter

The message displays either (1) the total number of good reads that match the master label or (2) the total number of good reads, or decodes. The count begins from the last power-on or Match Code/Good Read Counter Reset command. To count the good reads that match the master label, enable Match Code; to count good reads only, disable Match Code

4.14.2.4.7 Mismatch Counter

The message displays the total number of symbols successfully read that do not match the master label since power-on or the last Mismatch Counter command

4.14.2.5 Read Cycle Report

Information regarding the read cycle. Decode Data is referenced in the Decode Cycle Report

4.14.2.5.1 Table 3.13.5 Read Cycle Report Data

SHORT DESCRIPTION	SIZE
CAPTURE TIME	INT
TOTAL DECODE TIME	INT
TOTAL READCYCLE TIME	INT
RESERVED	INT

4.14.2.5.2 Capture Time

Total time it tool to capture the image

4.14.2.5.3 Total Decode Time

Total time spent decoding the symbol(s)

4.14.2.5.4 Total ReadCycle Time

Total Time Spent decoding the symbol which is the sum of the Capture, Decode and Overhead time.

4.14.2.6 Decode 1 Cycle Report

Information on the decoded symbol

4.14.2.6.1 Table 3.13.6 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL

4.14.2.6.2 Decode Location Top

Defines the row position of the upper-left starting point of the image window.

4.14.2.6.3 Decode Location Left

Defines the column position of the upper-left starting point of the image window.

4.14.2.6.4 Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.14.2.6.5 Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.14.2.6.6 Code Type

Bit field of the symbol in that was decoded for this report

4.14.2.6.6.1 Table 3.13.6.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
POSTAL CODE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

4.14.2.6.7 Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.14.2.7 Decode 1 Length

The total number of characters contained in the Decode Data SINT array

4.14.2.8 Decode 1 Data

Outputted decode 1 symbol data from the unit with one difference. Preamble and post amble symbols are not added. Maximum characters allowed is 160.

4.14.2.9 Decode 2 Cycle Report

Information on the decoded symbol

4.14.2.9.1 Table 3.13.7 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL

4.14.2.9.2 Decode Location Top

Defines the row position of the upper-left starting point of the image window.

4.14.2.9.3 Decode Location Left

Defines the column position of the upper-left starting point of the image window.

4.14.2.9.4 Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.14.2.9.5 Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.14.2.9.6 Code Type

Bit field of the symbol in that was decoded for this report

4.14.2.9.6.1 Table 3.13.7.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
POSTAL CODE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

4.14.2.9.7 Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.14.2.10 Decode 2 Length

The total number of characters contained in the Decode Data SINT array

4.14.2.11 Decode 2 Data

Outputted decode 2 symbol data from the unit with one difference. Preamble and post amble symbols are not added. Maximum characters allowed is 72.

4.14.2.12 Decode 3 Cycle Report

Information on the decoded symbol

*4.14.2.12.1*Table 3.13.8 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL
DECODE 3 LENGTH **DINT** **DECODE DATA** **SINT[72]**	

*4.14.2.12.2*Decode Location Top

Defines the row position of the upper-left starting point of the image window.

*4.14.2.12.3*Decode Location Left

Defines the column position of the upper-left starting point of the image window.

*4.14.2.12.4*Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

*4.14.2.12.5*Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

*4.14.2.12.6*Code Type

Bit field of the symbol in that was decoded for this report

4.14.2.12.6.1 Table 3.13.8.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
POSTAL CODE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

*4.14.2.12.7*Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.14.2.13 Decode 3 Length

The total number of characters contained in the Decode Data SINT array

4.14.2.14 Decode 3 Data

Outputted decode 3 symbol data from the unit with one difference. Preamble and post amble symbols are not added. Maximum characters allowed is 72.

4.14.2.15 Decode 4 Cycle Report

Information on the decoded symbol

*4.14.2.15.1*Table 3.13.9 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL

*4.14.2.15.2*Decode Location Top

Defines the row position of the upper-left starting point of the image window.

*4.14.2.15.3*Decode Location Left

Defines the column position of the upper-left starting point of the image window.

*4.14.2.15.4*Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

*4.14.2.15.5*Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

*4.14.2.15.6*Code Type

Bit field of the symbol in that was decoded for this report

4.14.2.15.6.1 Table 3.13.9.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
POSTAL CODE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

*4.14.2.15.7*Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.14.2.16 Decode 4 Length

The total number of characters contained in the Decode Data SINT array

4.14.2.17 Decode 4 Data

Outputted decode 4 symbol data from the unit with one difference. Preamble and post amble symbols are not added. Maximum characters allowed is 72.

4.14.3 Assembly Member Location

The following table is where members for the Input 4 Decode assembly are located.

	Member	Data Type	Target	Bit Number	Style	Data Length	Byte Offset
	InfoBits	**SINT**				**1 Byte**	**0**
32 Bit	BIT RunMode	BIT	InfoBits	0	NA	1 Bit	
	BIT ConnectionFaulted	BIT	InfoBits	1	NA	1 Bit	
	BIT DiagnosticActive	BIT	InfoBits	2	NA	1 Bit	
	Reserved	BIT	InfoBits	3 - 7	NA	5 Bits	
	DiagnosticSequenceCount	**SINT**			**Decimal**	**1Byte**	**1**
	ConfigurationChangeDetect	**SINT**				**1 Byte**	
	ConfigChangeDetect	BIT	ConfigurationChangeDetect	0	BOOL	1 Bit	
	Reserved	BIT	ConfigurationChangeDetect	1 - 7	NA	7 Bits	
	Reserved	**-**			**NA**	**1 Byte**	**3**
	DeviceStatus	**DINT**				**4 Bytes**	**4**
32 Bit Boundary	Online		DeviceStatus	0	BOOL	1 Bit	
	TriggerAcknowledge		DeviceStatus	1	BOOL	1 Bit	
	ExposureDone		DeviceStatus	2	BOOL	1 Bit	
	Decoding		DeviceStatus	3	BOOL	1 Bit	
	DataIsReady		DeviceStatus	4	BOOL	1 Bit	
	ReadCyclePass		DeviceStatus	5	BOOL	1 Bit	
	ReadCycleFail		DeviceStatus	6	BOOL	1 Bit	
	GeneralFault		DeviceStatus	7	BOOL	1 Bit	
	NewMatchCodeAcknowledged		DeviceStatus	8	BOOL	1 Bit	
	MatchCodeEnabled		DeviceStatus	9	BOOL	1 Bit	
	ImageSensorCalibrating		DeviceStatus	10	BOOL	1 Bit	
	ImageSensorCalibrationComplete		DeviceStatus	11	BOOL	1 Bit	
	Training		DeviceStatus	12	BOOL	1 Bit	
	TrainingComplete		DeviceStatus	13	BOOL	1 Bit	
	Optimizing		DeviceStatus	14	BOOL	1 Bit	
	OptimizingComplete		DeviceStatus	15	BOOL	1 Bit	
	AutoImagePhotometryEnabled		DeviceStatus	16	BOOL	1 Bit	
	AutoImagePhotometryComplete		DeviceStatus	17	BOOL	1 Bit	
	Output1Status		DeviceStatus	18	BOOL	1 Bit	
	Output2Status		DeviceStatus	19	BOOL	1 Bit	
	BufferOverflow		DeviceStatus	20	BOOL	1 Bit	
	Reserved	-	DeviceStatus	21-31	NA	11 Bits	

	Fault Code	**DINT**				**4 Bytes**	
32 Bit Boundary	CommandErrorDetected		FaultCode	0	BOOL	1 Bit	**8**
	CommunicationError		FaultCode	1	BOOL	1 Bit	
	FlashSectorUnprotectedFailure		FaultCode	2	BOOL	1 Bit	
	HostPortBufferOverflow		FaultCode	3	BOOL	1 Bit	
	Reserved		FaultCode	4 - 31	NA	28 Bits	
	Counters	**DINT[6]**				**24 Bytes**	
32 Bit	NoReadReadCycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**12**
32 Bit	MismatchPerReadcycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**16**
32 Bit	NoreadCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**20**
32 Bit	TriggerCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**24**
32 Bit	MatchCodeCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**28**
32 Bit	MismatchCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	**32**
	ReadCycleReport	**INT[4]**				**8 Bytes**	
32 Bit Boundary	CaptureTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	**36**
	TotalDecodeTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	**38**
32 Bit Boundary	TotalReadCycleTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	**40**
	Reserved	INT	ReadCycleReport	0 - 15	NA	2 Bytes	**42**
	DecodeCycleReport					**16 Bytes**	
32 Bit Boundary	DecodeLocationTop	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**44**
	DecodeLocationLeft	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**46**
32 Bit Boundary	DecodeLocationHeight	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**48**
	DecodeLocationWidth	INT	DecodeCycleReport	0 - 15	Decimal	2 Bytes	**50**
	CodeType (Subset)	DINT	DecodeCycleReport			4 Bytes	
32 Bit Boundary	AztecCode		CodeType	0	BOOL	1 Bit	**52**
	MicroQRCode		CodeType	1	BOOL	1 Bit	
	PostalCode		CodeType	2	BOOL	1 Bit	
	Code39		CodeType	3	BOOL	1 Bit	
	Codeabar		CodeType	4	BOOL	1 Bit	
	Interleaved2of5		CodeType	5	BOOL	1 Bit	
	UPCEAN		CodeType	6	BOOL	1 Bit	
	Code128EAN128		CodeType	7	BOOL	1 Bit	
	Code93		CodeType	8	BOOL	1 Bit	
	PD417		CodeType	9	BOOL	1 Bit	
	PharmaCode		CodeType	10	BOOL	1 Bit	
	DataMatrix		CodeType	11	BOOL	1 Bit	

	QRCode		CodeType	12	BOOL	1 Bit	
	BC412		CodeType	13	BOOL	1 Bit	
	RSS14		CodeType	14	BOOL	1 Bit	
	RSS14LTD		CodeType	15	BOOL	1 Bit	
	RSS14EXP		CodeType	16	BOOL	1 Bit	
	MicroPDF		CodeType	17	BOOL	1 Bit	
	PostalCode		CodeType	18	BOOL	1 Bit	
	DotCode		CodeType	19	BOOL	1 Bit	
	Reserved for future use		CodeType	20 - 31	BOOL	12 Bits	
32 Bit	PixelsPerElement	REAL	DecodeCycleReport	0 - 31	Decimal	4 Bytes	**56**
32 Bit	**DecodeLength**	**DINT**		**0 - 31**	**Decimal**	**4 Bytes**	**60**
32 Bit	**DecodeData**	**SINT[436]**		**0 - 3488**	**ASCII**	**436 Bytes**	**64**

4.15 MicroHAWK Input N Decode Assembly (Instance Decimal: 105 Hex: 0x69) IN = MicroHAWK→PLC

Designed to include any number of decode symbols, this assembly offers the most flexibility. In the read cycle data will contains the Read Cycle Report and the Decode Cycle Report with the decode data. Where this differs from the Input (1 Decode String) and Input (4 Decode String) is that in the Read cycle report will contain how many decode symbols where found, how many decode cycle reports are contained in the read cycle data and the offset for each decode cycle report. The user then go to the offset and read the decode cycle report and the decode data at that specific location and perform any action that needs to be done. (For instance move the data to a structure for the PLC program to use).

NOTE: This input block must use the Output Premier to function correctly

4.15.1 Table 3.14.1 Input Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
INFO BITS	1
DIAGNOSTIC SEQUENCE COUNT	1
CONFIGURATION SEQ. COUNT	1
RESERVED	1
DEVICE STATUS	4
FAULT	4
COUNTERS	24
READ CYCLE REPORT STATIC MEMBERS	8
RAW INPUT DATA	456

Total Size: 500 Bytes

4.15.2 Input Assembly Description

This sub section will describe the tag and each field related for the Input Assembly.

4.15.2.1 Input Assembly Module Header

The following header is used at the beginning of the input (produced) assembly. Definitions for the members are included below.

4.15.2.1.1 Info Bits

Bit field of the input module status

INFO BIT FIELD	
BIT RUNMODE	0
BIT CONNECTIONFAULTED	1
BIT DIAGNOSTICACTIVE	2
RESERVED	3-7

4.15.2.1.1.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.15.2.1.1.2 Connection Faulted

Connection to the target is 0 = up and working, 1 = not connected. The module always returns a zero in this member. The controller overwrites the zero with a one when the connection is not up.

4.15.2.1.1.3 Diagnostic Active

0 = No diagnostics active, 1 = One or more diagnostic or prognostics thresholds reached

Note: “Diagnostic” means a detected condition that prevents the primary signal from propagating from a sensor to the controller, or from the controller to an actuator.

4.15.2.1.2 Diagnostic Sequence Count

SHORT NAME	SIZE
DIAGNOSTIC SEQUENCE COUNT	SINT

Increments for each time a distinct diagnostic condition is detected, and also each time a distinct diagnostic condition transitions from detected to not detected. Set to zero by product reset or power cycle. Wraps from 255 (-1) to 1 skipping zero.

4.15.2.1.3 Configuration Change Detection

When a change in the working set has been detected by the device this bit will be set to 1. This means that the configuration in the project no longer matches the configuration in the device.
Any forward open sets this value back to 0.

4.15.2.2 Device Status

This tag describes the current state of the device. In table 1.2.1 the bit field is mapped to allow the user to know what state the device is in.

4.15.2.2.1 Table 3.14.1.2 Device Status Bit Field

DEVICE STATUS

BIT FIELD	Status
0	Online
1	Trigger Acknowledge
2	Exposure Done
3	Decoding
4	Data Is Ready
5	Read Cycle Pass
6	Read Cycle Fail
7	General Fault
8	New match code acknowledged
9	Match Code Enabled
10	Image Sensor Calibrating
11	Image Sensor Calibration Complete
12	Training
13	Training Complete
14	Optimizing
15	Optimization Complete
16	AutoImage Photometry Enabled
17	AutoImage Photometry Complete
18	Output1 Status
19	Output2 Status
20	Buffer Overflow
21-32	Reserved

4.15.2.2.2 Online

The units Current Read Cycle State

state

0 = Read cycle is disabled thus the unit is offline but the unit can receive commands. There is no data produced in the Input assembly and no data is consumed in the Output assembly when in this state.

1 = Read Cycle is enabled and the unit can be triggered and data is available for consumption and the unit will consume output data.

4.15.2.2.3 Trigger Acknowledged

This bit will go high when the unit has accepted the Trigger command in the Control tag. The user must lower the Trigger bit in the control tag in order for this bit to go back 0.

4.15.2.2.4 Exposure Done

When the image sensor exposure is complete this bit will go high and the user can move the object in the Field of view for the next image to be taken.

4.15.2.2.5 Decoding

When the unit is processing the image, this bit will be high. When the unit has completed the image process this bit will go low.

4.15.2.2.6 Data is Ready

The Read Cycle and Data Cycle Reports are ready for consumption when this bit goes high.

4.15.2.2.7 Read Cycle Pass

If the read cycle has passed all criteria, this bit will go high. It will go low when the ready begins to process the next image.

4.15.2.2.8 Ready Cycle Fail

If the read cycle has failed any of the criteria that was programmed, this bit will go high. It will go low when the ready begins to process the next image.

4.15.2.2.9 General Fault

When a fault occurs in the unit, this bit will go high. The user can reference the Fault Code tag for the error code and must remedy the problem. After the problem has been resolved the user can reset the fault in the Control tag in the Output assembly.

*4.15.2.2.10*New Match Code Acknowledge

When active the unit has accepted the data read on the last trigger as the new match code. User shall set the Learn New Match Code bit in the Control tag to zero when this bit goes high.

*4.15.2.2.11*Match Code Enabled

When this bit is 1 the unit will use the Match Code function to determine the Inspection Results.

*4.15.2.2.12*Image Sensor Calibrating

The unit is undergoing a calibration on one or all of the following:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

When the unit has completed calibration this bit will be set to zero.

*4.15.2.2.13*Image Sensor Calibration Complete

The unit has completed calibrating the image sensor for one or all of the following items:

- Exposure
- Gain
- Focus (If the unit has Auto focus capabilities)

The user shall set the Control bit Calibration Image Sensor to zero if they have not done so already.

*4.15.2.2.14*Training

When the unit is in the training process, this bit will be set to one. After the training process has completed, this bit will be set to zero.

*4.15.2.2.15*Training Complete

After the unit has completed the training process, this bit will be set to one. If the user has set the Train Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

*4.15.2.2.16*Optimizing

When the unit is optimizing this bit will be set to one. After optimization has completed, this bit will be set to zero.

*4.15.2.2.17*Optimization Complete

After the unit has completed the optimization process, this bit will be set to one. If the user has set the Optimize Unit bit in the Control Tag, they shall set it back to zero. If an error has occurred, the Fault Code Tag will display the error.

*4.15.2.2.18*AutoImage Photometry Enabled

The unit will use AutoImage Photometry when trying to decode the symbol. Disabling this will mean the unit is using fixed values for Exposure, Gain and (if applicable) focal distance.

*4.15.2.2.19*AutoImage Photometry Complete

This value will be set to one after the unit has completed an AutoImage Photometry calibration.

*4.15.2.2.20*Output 1 Status

Current status of the physical output 1 signal

*4.15.2.2.21*Output 2 Status

Current status of the physical output 2 signal

*4.15.2.2.22*Output 3 Status

Current status of the physical output 3 signal

*4.15.2.2.23*Buffer Overflow

When the data in the input buffer exceeds the buffer size (456 bytes) then this bit will go high alerting the user that the data is an incomplete segment.

4.15.2.3 Fault Code

This tag shall display the fault codes when the unit has faulted for any commands sent to it. When the user issues the Reset Fault in the Control Tag, this value will be set to zero.

4.15.2.3.1 Table 3.14.3 Bit Field Layout

COUNTERS	
COMMAND ERROR DETECTED	0
COMMUNICATION ERROR	1
FLASH SECTOR UNPROTECTED FAILURE	2
HOST PORT BUFFER OVERFLOW	3
RESERVED FOR FUTURE EXPANSION	4-31

4.15.2.4 Counters

Displays the counters stored in the unit upon power up or after a configuration change. These counters can be reset via the output command tag.

4.15.2.4.1 Table 3.14.4 Counters Input Array

COUNTERS	
NOREAD READCYCLE COUNTER	DINT
MISMATCH PER READCYCLE COUNTER	DINT
NOREAD COUNTER	DINT
TRIGGER COUNTER	DINT
MATCH CODE COUNTER	DINT
MISMATCH COUNTER	DINT

NOTE: Time starts over with power on but not with a <A> or <Z> type reset.

4.15.2.4.2 NoRead Cycle Counter

The message displays the total number of noread read cycles that have occurred since power-on or the last Noread Read cycle Counter Reset command

4.15.2.4.3 MisMatch Per ReadCycle Counter

The message displays the total number of mismatched code pre readcycle that have occurred since power-on or the last Mismatch per Readcycle Counter Reset command

4.15.2.4.4 NoRead Counter

The message displays the total number of noreads that have occurred since power-on or the last Noread Counter Reset command

4.15.2.4.5 Trigger Counter

The message displays the total number of triggers that have occurred since power-on or the last Trigger Counter Reset command

4.15.2.4.6 MatchCode Counter

The message displays either (1) the total number of good reads that match the master label or (2) the total number of good reads, or decodes. The count begins from the last power-on or Match Code/Good Read Counter Reset command. To count the good reads that match the master label, enable Match Code; to count good reads only, disable Match Code

4.15.2.4.7 Mismatch Counter

The message displays the total number of symbols successfully read that do not match the master label since power-on or the last Mismatch Counter command

4.15.2.5 Read Cycle Data

Due to there being more than 1 Decode Report the read cycle data is grouped into a 460 byte data field. The user shall use the Read Cycle Report to determine how many Decodes were found and the Decode Report offsets for each decoded symbol found in the Read Cycle.

4.15.2.5.1 Read Cycle Report

Information regarding the read cycle. The difference in this Input assembly is the variable length in this field. The user can reference the Offset of each report found, which is indicted in the tag Number of decode reports. Each Decode Cycle Report will have an offset to indicate to the user where to unpack the data in the assembly. The decode report is the same as in the Input Assembly for 1 decode symbol.

4.15.2.5.1.1 Table 3.14.5 Read Cycle Report Data

SHORT DESCRIPTION	SIZE
CAPTURE TIME	INT
TOTAL DECODE TIME	INT
TOTAL READCYCLE TIME	INT
NUMBER OF DECODES IN READ CYCLE	SINT
NUMBER OF DECODE REPORTS	SINT
OFFSET OF REPORT 1	DINT
OFFSET OF REPORT 2	DINT
…	…
OFFSET OF REPORT N	DINT

4.15.2.5.1.2 Capture Time

Total time it tool to capture the image

4.15.2.5.1.3 Total Decode Time

Total time spent decoding the symbol(s)

4.15.2.5.1.4 Total ReadCycle Time

Total Time Spent decoding the symbol which is the sum of the Capture, Decode and Overhead time.

4.15.2.5.1.5 Number of Decodes in Read Cycle

The total number of decoded symbols found during the read cycle

4.15.2.5.1.6 Number of Decode Reports

The total number of reports associated with the decode symbols. This will match the total number of symbols found in the ready cycle.

4.15.2.5.1.7 Offset of Report (n)

The offset value in bytes, where the Decode Cycle Report is located in the Read Cycle Data array. The offset of report 1 will always be 8, meaning that the user always read byte 8 in the Read Cycle Report to locate the Decode Cycle Report 1. (See figure below)

Byte	0	1	2	3	4	5	6	7	8	…	n
Item	Capture Time		Total Decode Time		Total Read Cycle Time		Number of Decodes in Read Cycle	Number of Decode Cycle Report	Offset of Report 1	…	Offset of Report n

4.15.2.6 Decode Cycle Report

Information on the decoded symbol

4.15.2.6.1 Table 3.14.6 Decode Cycle Report

DESCRIPTOIN	SIZE
DECODE LOCATION TOP	INT
DECODE LOCATION LEFT	INT
DECODE LOCATION HEIGHT	INT
DECODE LOCATION WIDTH	INT
CODE TYPE	DINT
PIXELS PER ELEMENT	REAL

4.15.2.6.2 Decode Location Top

Defines the row position of the upper-left starting point of the image window.

4.15.2.6.3 Decode Location Left

Defines the column position of the upper-left starting point of the image window.

4.15.2.6.4 Decode Location Height

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.15.2.6.5 Decode Location Width

Defines the size, in rows, of the image window. Maximum value is defined as the Maximum row size of Image sensor, minus the row pointer value.

4.15.2.6.6 Code Type

Bit field of the symbol in that was decoded for this report

4.15.2.6.6.1 Table 3.14.6.1 Code Type Bit Map

SYMBOLOGY	
AZTEC CODE	0
MICROQRCODE	1
POSTAL CODE	2
CODE 39	3
CODEABAR	4
INTERLEAVED 2 OF 5	5
UPC/EAN	6
CODE 128/EAN 128	7
CODE 93	8
PD417	9
PHARMACODE	10
DATAMATRIX	11
QRCODE	12
BC412	13
RSS-14	14
RSS-14 LTD	15
RSS-14 EXP	16
MICROPDF	17
POSTAL CODE	18
DOTCODE	19
RESERVED FOR FUTURE USE	20
RESERVED FOR FUTURE USE	21
RESERVED FOR FUTURE USE	22
RESERVED FOR FUTURE USE	23
RESERVED FOR FUTURE USE	24
RESERVED FOR FUTURE USE	25
RESERVED FOR FUTURE USE	26
RESERVED FOR FUTURE USE	27
RESERVED FOR FUTURE USE	28
RESERVED FOR FUTURE USE	29
RESERVED FOR FUTURE USE	30
RESERVED FOR FUTURE USE	31

4.15.2.6.7 Pixels Per Element

The number of pixels for each element, either dark or light for both x and y directions

4.15.2.6.8 Decode Length

The number of characters found in the decode string

4.15.2.6.9 Decode Data

Outputted decode data from the unit with one difference. Preamble and post amble symbols are not added.

4.15.2.7 Assembly Member Location

The following table is where members for the Input 4 Decode assembly are located.

	Member	Data Type	Target	Bit Number	Style	Data Length	Byte Offset
	InfoBits	**SINT**				**1 Byte**	0
32 Bit	BIT RunMode	BIT	InfoBits	0	NA	1 Bit	
	BIT ConnectionFaulted	BIT	InfoBits	1	NA	1 Bit	
	BIT DiagnosticActive	BIT	InfoBits	2	NA	1 Bit	
	Reserved	BIT	InfoBits	3 - 7	NA	5 Bits	0
	DiagnosticSequenceCount	**SINT**			**Decimal**	**1Byte**	1
	ConfigurationChangeDetect	**SINT**				**1 Byte**	2
	ConfigChangeDetect	BIT	ConfigurationChangeDetect	0	BOOL	1 Bit	
	Reserved	BIT	ConfigurationChangeDetect	1 - 7	NA	7 Bits	
	Reserved	**-**			**NA**	**1 Byte**	3
	DeviceStatus	**DINT**				**4 Bytes**	4
32 Bit Boundary	Online		DeviceStatus	0	BOOL	1 Bit	
	TriggerAcknowledge		DeviceStatus	1	BOOL	1 Bit	
	ExposureDone		DeviceStatus	2	BOOL	1 Bit	
	Decoding		DeviceStatus	3	BOOL	1 Bit	
	DataIsReady		DeviceStatus	4	BOOL	1 Bit	
	ReadCyclePass		DeviceStatus	5	BOOL	1 Bit	
	ReadCycleFail		DeviceStatus	6	BOOL	1 Bit	
	GeneralFault		DeviceStatus	7	BOOL	1 Bit	
	NewMatchCodeAcknowledged		DeviceStatus	8	BOOL	1 Bit	
	MatchCodeEnabled		DeviceStatus	9	BOOL	1 Bit	

	ImageSensorCalibrating		DeviceStatus	10	BOOL	1 Bit	
	ImageSensorCalibrationComplete		DeviceStatus	11	BOOL	1 Bit	
	Training		DeviceStatus	12	BOOL	1 Bit	
	TrainingComplete		DeviceStatus	13	BOOL	1 Bit	
	Optimizing		DeviceStatus	14	BOOL	1 Bit	
	OptimizingComplete		DeviceStatus	15	BOOL	1 Bit	
	AutoImagePhotometryEnabled		DeviceStatus	16	BOOL	1 Bit	
	AutoImagePhotometryComplete		DeviceStatus	17	BOOL	1 Bit	
	Output1Status		DeviceStatus	18	BOOL	1 Bit	
	Output2Status		DeviceStatus	19	BOOL	1 Bit	
	BufferOverflow		DeviceStatus	20	BOOL	1 Bit	
	Reserved	-	DeviceStatus	21-31	NA	11 Bits	
	Fault Code	**DINT**				**4 Bytes**	8
32 Bit Boundary	CommandErrorDetected		FaultCode	0	BOOL	1 Bit	
	CommunicationError		FaultCode	1	BOOL	1 Bit	
	FlashSectorUnprotectedFailure		FaultCode	2	BOOL	1 Bit	
	HostPortBufferOverflow		FaultCode	3	BOOL	1 Bit	
	Reserved		FaultCode	4 - 31	NA	28 Bits	
	Counters	**DINT[6]**				**24 Bytes**	8
32 Bit	NoReadReadCycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	12
32 Bit	MismatchPerReadcycleCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	16
32 Bit	NoreadCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	20
32 Bit	TriggerCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	24
32 Bit	MatchCodeCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	28
32 Bit	MismatchCounter	DINT	Counters	0 - 31	Decimal	4 Bytes	32
	ReadCycleReport	**INT[4]**				**8 Bytes**	
	CaptureTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	36
32 Bit Boundary	TotalDecodeTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	38
32 Bit Boundary	TotalReadCycleTime	INT	ReadCycleReport	0 - 15	Decimal	2 Bytes	40

	NumberofDecodesInReadCycle	SINT	ReadCycleReport	0 - 7		1 Byte	41
	NumberofDecodeReports	SINT	ReadCycleReport	0 - 7		1 Byte	42
	RAWInputData	**SINT[456]**				**16 Bytes**	44
	RawData	SINT	RAWInputData	0 - 3647	HEX	456 Bytes	

4.16 MicroHAWK Output Legacy Assembly (Instance Decimal: 198 Hex: 0xC6) OUT = PLC→ MicroHAWK

The section describes the output assembly for the Ethernet/IP Communications for the MicroHAWK. All output commands will issue the targeted event in the unit and will be echoed in the input assembly when the unit has responded and issued the event successfully. All parameter changes made on the next read cycle.

4.16.1 Table 3.16.1 Output Assembly Table

SHORT DESCRIPTION	PLC DATA TYPE	SIZE (BYTES)
USER DEFINED TAGS	DINT	4
COMMANDS	DINT	4
EXTERNAL OUTPUT	DINT	4

Total Size: 12 Bytes

4.16.2 Output Assembly Description

This sub section will describe the tag and each field related to the Output Assembly.

4.16.2.1 User Defined Tags

This provides the PLC programmer a method of uniquely identifying multiple readers in the system. This field serves no functional purpose in the MicroHAWK. The value sent by the PLC for this field is echoed back to the input assemblies.

4.16.2.2 Commands

The section describes the commands that can be outputted to the unit. The unit will respond to a successful acknowledgment and execution in the input assembly.

4.16.2.2.1 Table 3.2.1 Command Bit Field

BIT FIELD	COMMAND
0	Trigger
1	New Master
2	Buffer Overflow
3-7	Reserved
8	Disable Scanning
9-15	Reserved
16	Clear Read Cycle Report and Counters
17	Unlatch Outputs
18-31	Reserved

4.16.2.2.2 Trigger

Edge event-driven. Takes effect when read mode is Serial, Edge, or Level. A transition from 0 to 1 is a rising edge trigger event. A transition from 1 to 0 is a falling edge trigger event. The following sources all induce trigger events in the reader, including:

- A serial command from a serial com port
- EZ button
- External Trigger input signal on connector A
- Command: Trigger bit in the OUT assembly

If the reader is to be exclusively triggered by the PLC, then all other trigger sources must be kept idle

4.16.2.2.3 New Master

Edge-event driven. A transition from 0 to 1 is a command to the unit similar to sending the <G> serial command, or activating the New Master input on connector A. When activated, the New Master function instructs the reader to store the next decode in the master symbol database.

4.16.2.2.4 Disable Scanning

Operates the same as the <H> and <I> commands. A transition from 0 to 1 is the same as sending an <I> command, which issues a “disable” event. A transition from 1 to 0 is the same thing as sending an <H> command, which issues an “enable” event. Note that the most recent command, either <H> or <I> serial commands or the Camera Action:DisableScanning command will always override the previous “scanning disable” state. To verify scanning status, observe the DeviceStatus field in asm 0x65.

4.16.2.2.5 Clear Read Cycle Report and Counters

Trigger, Decode/Match, Mismatch, Noread, Decoded Data string, and Sequence. A transition from 0 to 1 is similar to sending the commands <U><W><Y><O>, which clear the historical read cycle counters. Also, the Sequence counter and Decoded Data string will go to 0. Note that if this command is received while a read cycle is active, execution of the command will be delayed until the read cycle has ended, and the read cycle’s information will probably be lost.

4.16.2.2.6 Unlatch Outputs

If any outputs are configured for “Unlatch on Input1”, a transition from 0 to 1 will unlatch the output. See configuration commands K810-812. It is not necessary for Input 1 to be enabled.

4.16.2.3 External Output

This sub section details the External Output bit field for the Output Assembly

BIT FIELD	PIN NAME
0	Out1
1	Out2
2	Out3
3-31	Reserved

0 = open the output contact
1 = close the output contact
Note: Not operational at this time.

4.16.3 Assembly Member Location

The following table is where members for the Output assembly are located.

	Member	Data Type	Target	Bit Number	Style	Data Length
	User Defined Tag	**DINT**				**4 Bytes**
32 Bit Boundary	UserTag_1		User Defined Tag	0	BOOL	1 Bit
	UserTag_2		User Defined Tag	1	BOOL	1 Bit
	UserTag_3		User Defined Tag	2	BOOL	1 Bit
	UserTag_4		User Defined Tag	3	BOOL	1 Bit
	UserTag_5		User Defined Tag	4	BOOL	1 Bit
	UserTag_6		User Defined Tag	5	BOOL	1 Bit
	UserTag_7		User Defined Tag	6	BOOL	1 Bit
	UserTag_8		User Defined Tag	7	BOOL	1 Bit
	UserTag_9		User Defined Tag	8	BOOL	1 Bit
	UserTag_10		User Defined Tag	9	BOOL	1 Bit
	UserTag_11		User Defined Tag	10	BOOL	1 Bit
	UserTag_12		User Defined Tag	11	BOOL	1 Bit
	UserTag_13		User Defined Tag	12	BOOL	1 Bit
	UserTag_14		User Defined Tag	13	BOOL	1 Bit
	UserTag_15		User Defined Tag	14	BOOL	1 Bit
	UserTag_16		User Defined Tag	15	BOOL	1 Bit
	UserTag_17		User Defined Tag	16	BOOL	1 Bit
	UserTag_18		User Defined Tag	17	BOOL	1 Bit
	UserTag_19		User Defined Tag	18	BOOL	1 Bit
	UserTag_20		User Defined Tag	19	BOOL	1 Bit
	UserTag_21		User Defined Tag	20	BOOL	1 Bit
	UserTag_22		User Defined Tag	21	BOOL	1 Bit
	UserTag_23		User Defined Tag	22	BOOL	1 Bit
	UserTag_24		User Defined Tag	23	BOOL	1 Bit
	UserTag_25		User Defined Tag	24	BOOL	1 Bit
	UserTag_26		User Defined Tag	25	BOOL	1 Bit
	UserTag_27		User Defined Tag	26	BOOL	1 Bit
	UserTag_28		User Defined Tag	27	BOOL	1 Bit
	UserTag_29		User Defined Tag	28	BOOL	1 Bit
	UserTag_30		User Defined Tag	29	BOOL	1 Bit
	UserTag_31		User Defined Tag	30	BOOL	1 Bit
	UserTag_32		User Defined Tag	31	BOOL	1 Bit
	Commands	**DINT**				**4 Bytes**
32 Bit Boundary	Trigger		Commands	0	BOOL	1 Bit
	New Master		Commands	1	BOOL	1 Bit
	Reserved for future use		Commands	2 - 7	BOOL	6 Bits
	Disable Scanning		Commands	8	BOOL	1 Bit
	Reserved for future use		Commands	9 - 15	BOOL	7 Bits
	Clear Read Cycle Report and Counters		Commands	16	BOOL	1 Bit
	Unlatch Outputs		Commands	17	BOOL	1 Bit
	Reserved for future use		Commands	18 - 31	BOOL	14 Bits
	External Output	**DINT**				**4 Bytes**
32 Bit Boundary	Out1		External Output	0	BOOL	1 Bit
	Out2		External Output	1	BOOL	1 Bit
	Out3		External Output	2	BOOL	1 Bit
	Reserved for future use		External Output	3 - 31	BOOL	29 Bits

4.17 MicroHAWK Output Assembly (Instance Decimal: 197 Hex: 0xC5) OUT = PLC→ MicroHAWK

The section describes the output assemblies for the Ethernet/IP Communications for the MicroHAWK. All output commands will issue the targeted event in the unit and will be echoed in the input assembly when the unit has responded and issued the event successfully. All parameter changes made on the next read cycle.

4.17.1 Table 3.16.1 Output Assembly Table

SHORT DESCRIPTION	PLC DATA TYPE	SIZE (BYTES)
COMMANDS	DINT	4

Total Size: 4 Bytes

4.17.2 Output Assembly Description

This sub section will describe the tag and each field related to the Output Assembly.

4.17.2.1 Commands

The section describes the commands that can be outputted to the unit. The unit will respond to a successful acknowledgment and execution in the input assembly.

4.17.2.1.1 Table 3.2.1 Command Bit Field

BIT FIELD	COMMAND
0	Run Mode
1	Trigger
2	Enable MatchCode
3	Reset General Fault
4	Clear No Read ReadCycle Count
5	Clear MisMatch ReadCycle Count
6	Clear No Read Count
7	Clear Trigger Count
8	Clear Matchcode Count
9	Clear Mismatch Count
10	Output_1
11	Output_2
12	Output_3
13-31	Reserved for future use

4.17.2.1.2 Online

Ends the current read cycle, and will not allow the imager to enter another read cycle until re-enabled by changing the state from 1 to 0. This feature is useful during extended periods of time when no symbols are being decoded, or the imager is being configured. Disabling the imager will not affect any commands that have already been downloaded.

4.17.2.1.3 Trigger

Edge event-driven. Takes effect when read mode is Serial, Edge, or Level. A transition from 0 to 1 is a rising edge trigger event. A transition from 1 to 0 is a falling edge trigger event. The following sources all induce trigger events in the reader, including:

- A serial command from a serial com port
- EZ button
- External Trigger input signal on connector A
- Command: Trigger bit in the OUT assembly

If the reader is to be exclusively triggered by the PLC, then all other trigger sources must be kept idle

4.17.2.1.4 Enable MatchCode

When the option is set to anything other than disabled the scanner will compare symbols read in the read cycle to master symbols in a database. The results of this comparison can be used to specify the output of the read cycle such as whether to output symbol data or change the state of the programmable outputs. Matchcode is only functional in triggered modes. Multi-symbol matchcode is supported but only with the matchcode type option set to true.

4.17.2.1.5 Reset General Fault

When a Fault occurs in the system, the user shall use this bit to try to reset the fault after they have remedied the problem (if applicable).

4.17.2.1.6 Clear Noread Readcycle Counter

Resets the total number of noread readcycles that have occurred since power-on or the last Noread Readcycle Counter Reset command to 000000000.

4.17.2.1.7 Clear Mismatch Readcycle Counter

Resets the total number of mismatched code pre readcycle that have occurred since power-on or the last Mismatch per Readcycle Counter Reset command to 000000000.

4.17.2.1.8 Clear Noread Counter

Resets the total number of noreads that have occurred since power-on or the last Noread Counter Reset command to 000000000.

4.17.2.1.9 Clear Trigger Counter

Resets the total number of triggers that have occurred since power-on or the last Trigger Counter Reset command to 000000000.

4.17.2.1.10 Clear Match Code Counter

Resets the Match Code/Good Read Counter to 000000000.

4.17.2.1.11 Clear Mismatch Counter

Resets the total number of symbols successfully read that do not match the master label since power-on or the last Mismatch Counter command to 000000000.

4.17.2.1.12 Output 1

Raises Output 1 if set to 1 and set's output 1 to 0 when this value is 0. This value can be read in the input assembly in bit 19 under the device status tag.

4.17.2.1.13 Output 2

Raises Output 2 if set to 1 and set's output 1 to 0 when this value is 0. This value can be read in the input assembly in bit 20 under the device status tag.

4.17.2.1.14 Output 3

Raises Output 2 if set to 1 and set's output 1 to 0 when this value is 0. This value can be read in the input assembly in bit 20 under the device status tag.

4.17.3 Assembly Member Location

The following table is where members for the Output assembly are located.

	Member	**DataType**	**Target**	**Bit Number**	**Style**	**Data Length**
	Commands	**DINT**				**4 Bytes**
32 Bit Boundary	RunMode		Commands	0	BOOL	1 Bit
	Trigger		Commands	1	BOOL	1 Bit
	EnableMatchCode		Commands	2	BOOL	1 Bit
	ResetGeneralFault		Commands	3	BOOL	1 Bit
	ClearNoReadReadCycleCount		Commands	4	BOOL	1 Bit
	ClearMisMatchReadCycleCount		Commands	5	BOOL	1 Bit
	ClearNoReadCount		Commands	6	BOOL	1 Bit
	ClearTriggerCount		Commands	7	BOOL	1 Bit
	ClearMatchcodeCount		Commands	8	BOOL	1 Bit
	ClearMismatchCount		Commands	9	BOOL	1 Bit
	Output_1		Commands	10	BOOL	1 Bit
	Output_2		Commands	11	BOOL	1 Bit
	Reserved for future use		Commands	12 - 31	BOOL	20 Bits

4.18 MicroHAWK Serial Command Assembly (Instance Decimal: 69 Hex: 0x45) IN/OUT = PLC←→ MicroHAWK

Microscan has created an assembly to send and receive K Commands to the MicroHAWK. To be able to receive K Command request the unit's firmware much be greater than version 1.1. Version under 1.1 will be able to send K Commands but not receive any response back. This elevates the need for the programmer to create a TCP Socket in the PLC program and to perform the same task.

This assembly is accessible only through explicit messaging, as to allow the programmer to use the implicit Input/Output assemblies in conjunction with this assembly (IE. The controller can be using the Input/Output Assemblies and send an explicit message through the MSG instruction as the same time).

4.18.1 Assembly Information

To use this assembly please use the following information in the MSG Instruction

Service Code	**45**
Class	68
Instance	1
Attribute	1

4.18.2 Serial Command Assembly Table

Description of the Serial Command Assembly. This mimics a STRING in the PLC program of 256 bytes long.

SHORT DESCRIPTION	PLC DATA TYPE	SIZE (BYTES)
COMMAND LENGTH	DINT	4
COMMAND STRING	SINT[256]	256

Total: 260 bytes

4.18.3 Command Length

Total number of characters in the Command string

4.18.4 Command String

The ASCII character array of the command that will be sent from the controller to the device.

4.18.5 Assembly Member Description

The following table is where members for the Serial Command Assembly are located.

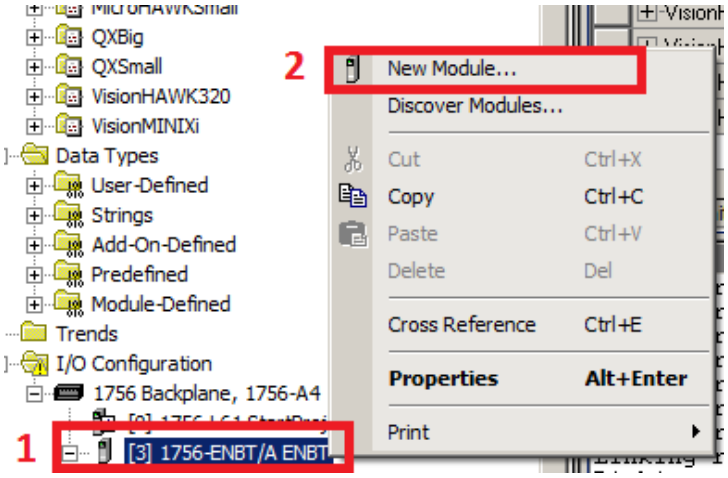
	Member	Data Type	Target	Bit Number	Style	Data Length	Byte Offset
32 Bit Boundary	**Command Length**	**DINT**			**Decimal**	**4 Bytes**	**0**
32 Bit Boundary	**Command String**	**SINT[256]**			**ASCII**	**256 Bytes**	**4**

5 Setup using Generic Ethernet Module

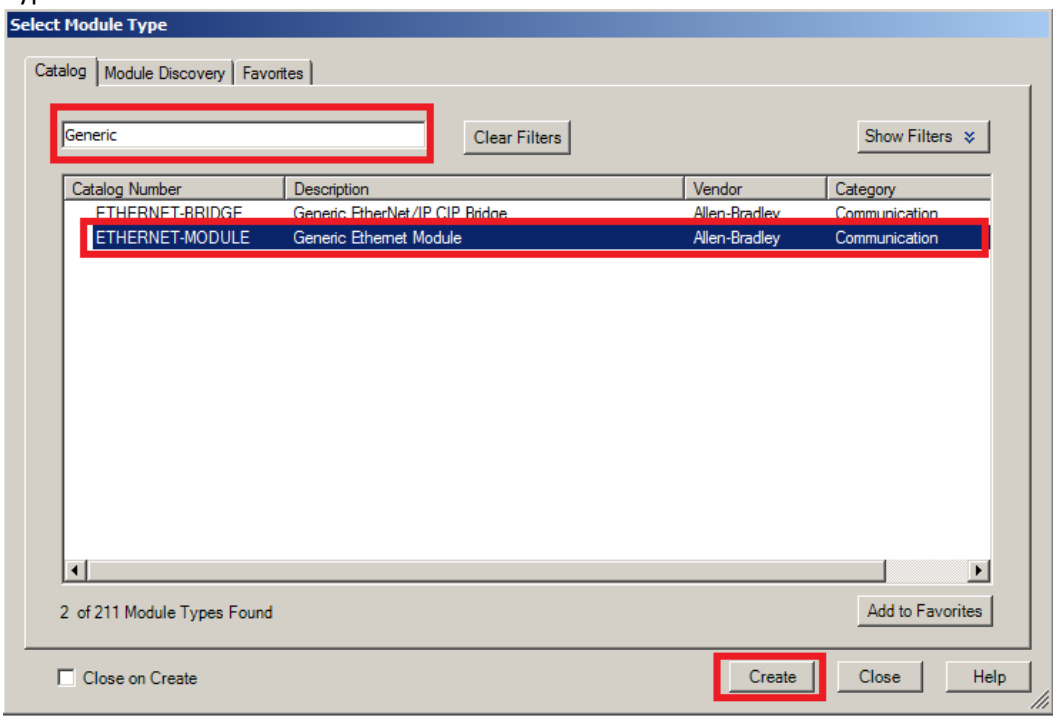
The section will go over how to add the MicroHAWK using the Generic Ethernet Module supplied in the RSLogix softare. This section can also be used if the controller's software does not support EDS files, or if programmer simply wants to use the Generic Ethernet Module in lieu of the EDS file. Make sure you can connect to the unit and communicate with it. A simple ping test should suffice for this test.

5.1 Adding the Generic Ethernet Module

1) Right Click the Ethernet Controller (1) and select **New Module** (2)



2) Type in the Search Field **Generic Ethernet Module**. Select the item and click **Create**



5.2 Configuration of Generic Ethernet Module

Configure the module as described below. Microscan offers many Input/Output Assemblies and created the table below to help with the pairing of the Input and Output Assemblies.

a. **Name:** A useful name to remember the unit
b. **IP Address:** The IP Address of MicroHAWK unit
c. **Assembly Input Instance** (Choose from the following)

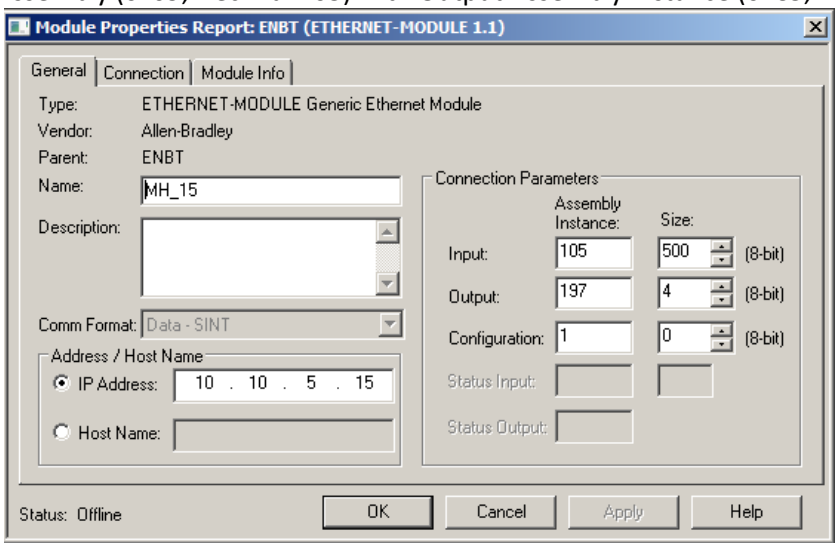
5.2.1 Input/Output Configuration Table

COMM FORMAT	INPUT INSTANCE	INPUT SIZE	OUTPUT INSTANCE	OUTPUT SIZE	CONFIGURATION INSTANCE	CONFIGURATION SIZE
DATA-DINT	100	21	198	3	1	0
DATA-DINT	101	44	198	3	1	0
DATA-SINT	102	248	197	4	1	0
DATA-SINT	103	500	197	4	1	0
DATA-SINT	104	500	197	4	1	0
DATA-SINT	105	500	197	4	1	0

Please note there is a difference in the COMM FORMAT!!!

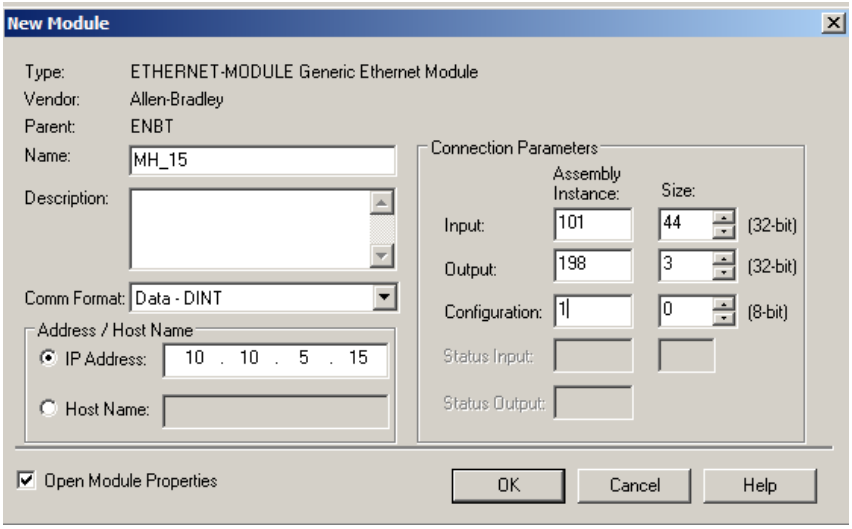
5.2.1.1 Configuration Example 1 using new Input/Output Assemblies

Input N Decode Assembly (0x69, Decimal 105) with Output Assembly Instance (0xC5, Decimal 197)

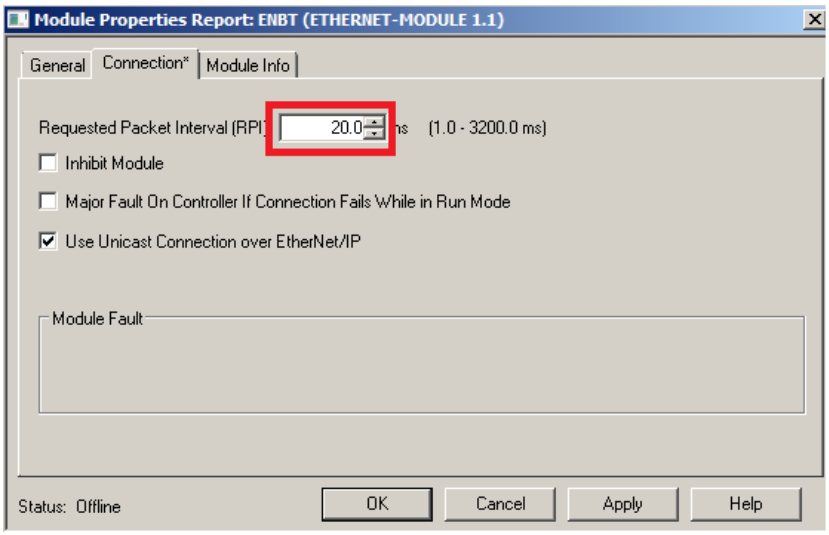


5.2.1.2 Configuration Example 2 using legacy Input/Output Assemblies

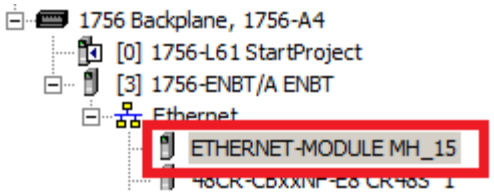
Input Big Assembly (0x65, Decimal 101) with Output (Legacy) Instance (0xC6, Decimal 198)



3) Click the **Connections** Tab and set the **Requested Packet Interval (RPI)** to the desired interval rate (Default is 20 ms).
   a. NOTE[1]: Try to make this value roughly the same as the units capture time to avoid heavy network bandwidth usage.
   b. NOTE[2]: The MicroHAWK does not currently support Multicast as of firmware release 1.1. Make sure **Use Unicast Connection over EtherNet/IP** is **checked**
   c.



4) Click **OK** to Add the Ethernet Module to the Ethernet Node Tree



   a.
5) It is safe to download to the controller now.
6) This completes the Addition of the Generic Ethernet Module. Please go to Section 7 to import the sample routines

6 Setup using the MicroHAWK EDS File

This section will go over the necessary steps needed to setup the MicroHAWK unit in RSLogix. Please refer to Section 4 to verify if the version of RSLogix supports the EDS file.

NOTE: To use the following input assemblies for RSLogix version 20 and lower

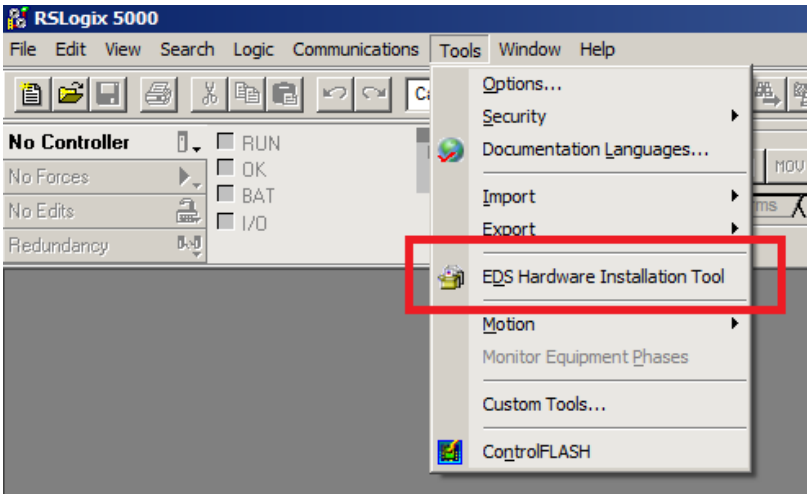
- **Input 1 Decode**
- **Input 4 Decode**
- **Input N Decode**

RSLogix version 20 will report a fault "Invalid Input Data Size" if adding the device using the EDS method. This is because the Input Assemblies are 500 bytes and do not have a status header. This issue has been resolve in RSLogix 5000 version 21.

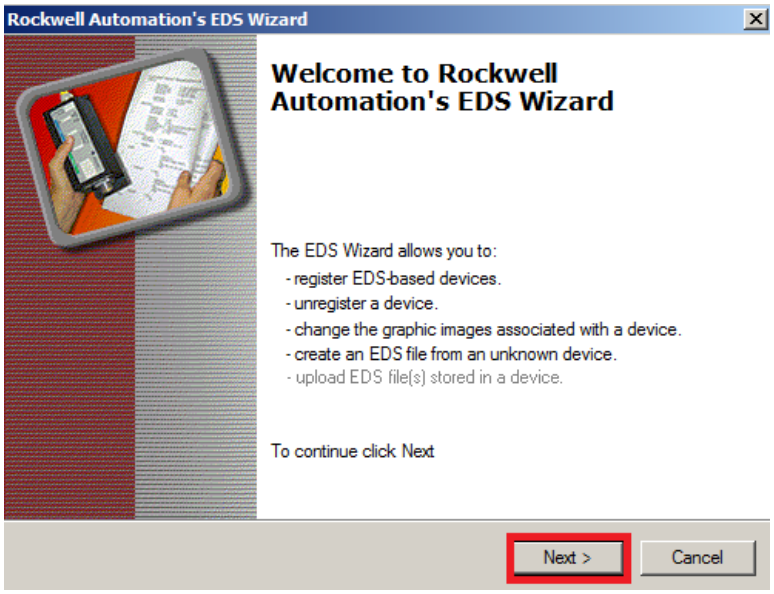
NOTE: Depending on what Firmware version the unit has, will determine what EDS file to use during the installation process. Below is a table to reference on which EDS file to install in RSLogix/Studio 5000

6.1 Installing the EDS file

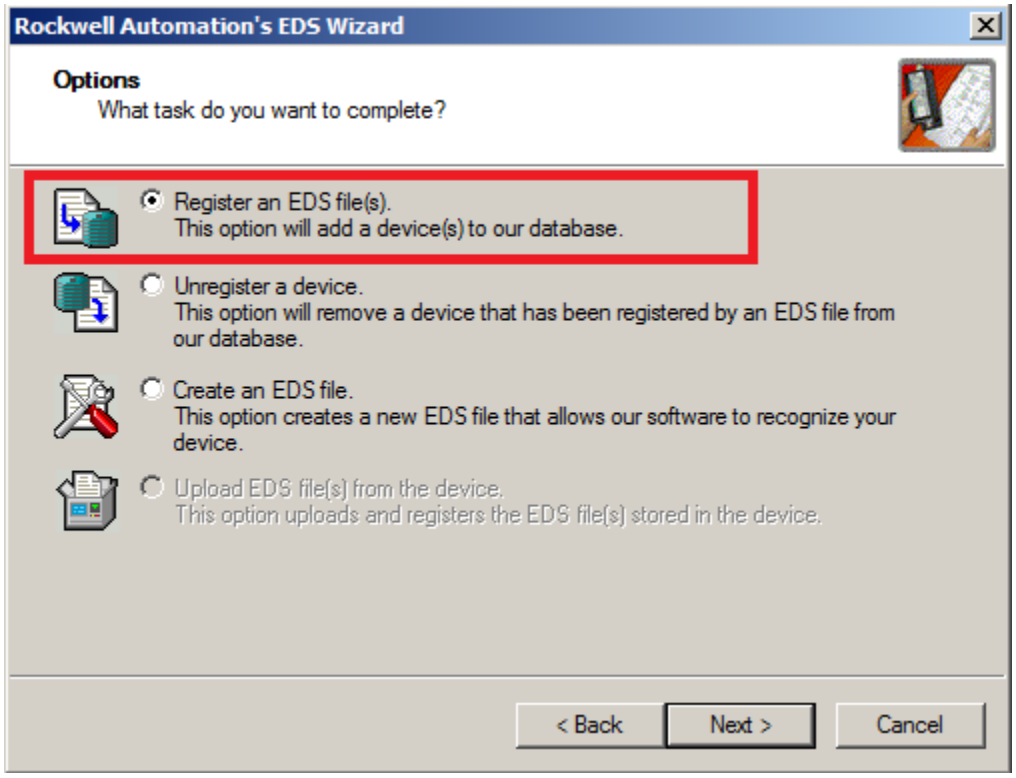
1) In RS Logix5000, import the EDS file by going to Tools→**EDS Hardware Installation Tool**



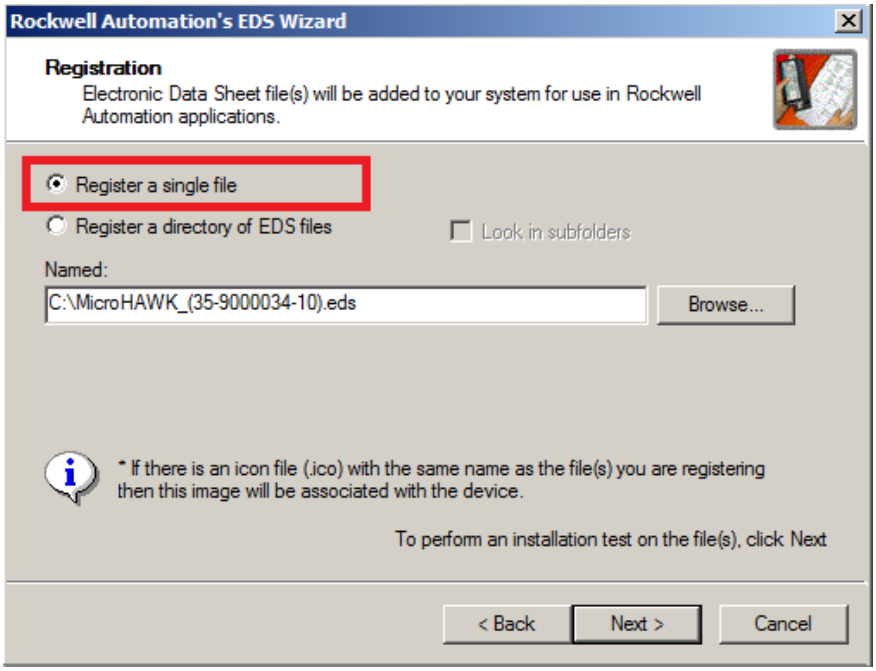
2) Click on **Next** when prompted



3) Select **Register an EDS file(s)**



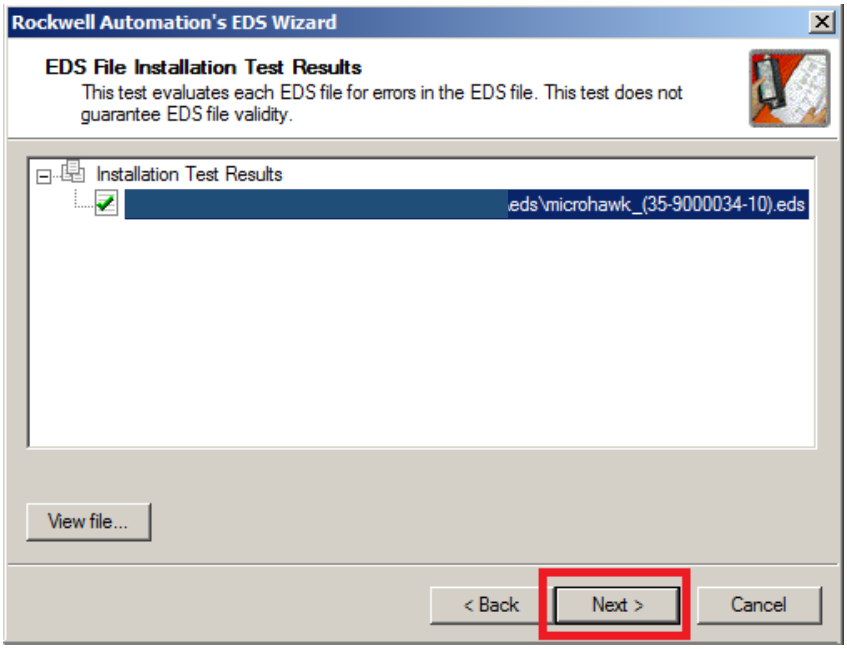
4) Select **Register a single file** and navigate to where you downloaded the .eds file and select next



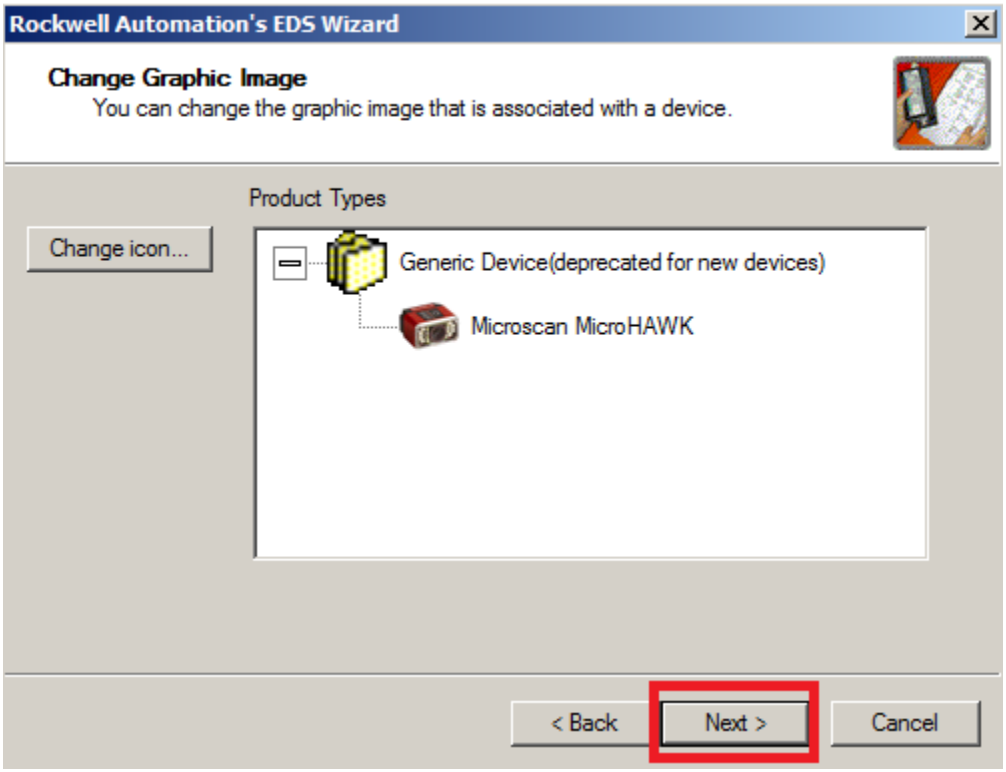
6.1.1 EDS File Table

MICROHAWK FIRMWARE VERSION	EDS FILE
RELEASE < 1.1	MicroHAWK_(35-9000034-10).eds
RELEASE ≥ 1.2	MicroHAWK_(35-9000034-11).eds

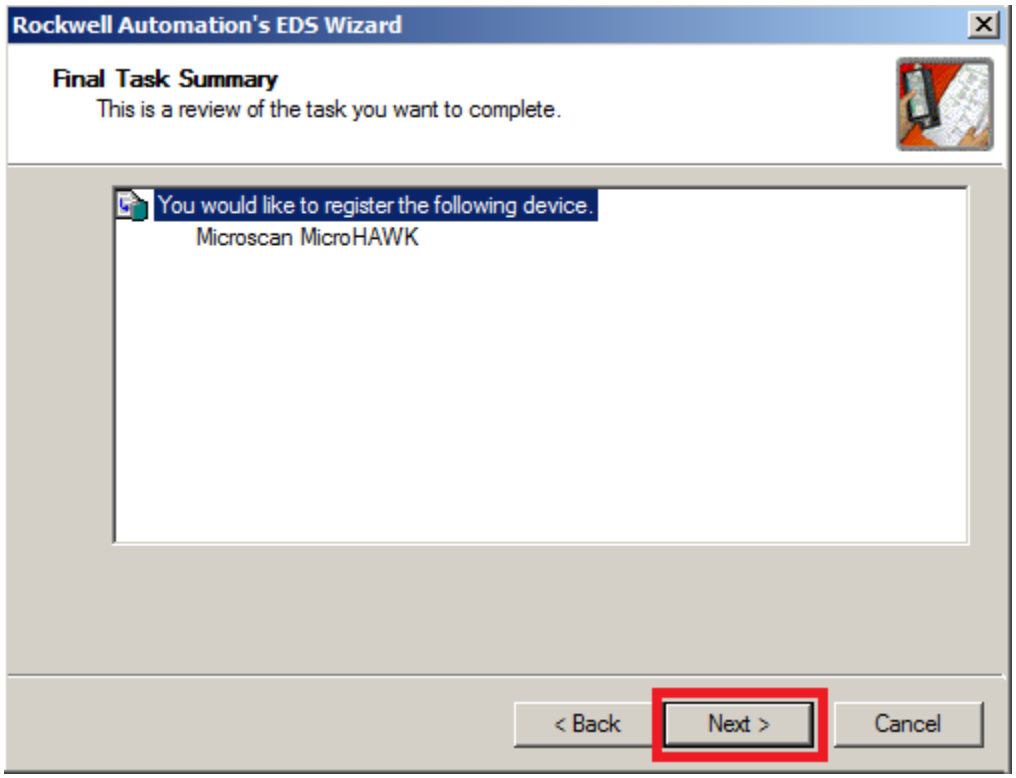
5) Select **Next** if the installation succeeds



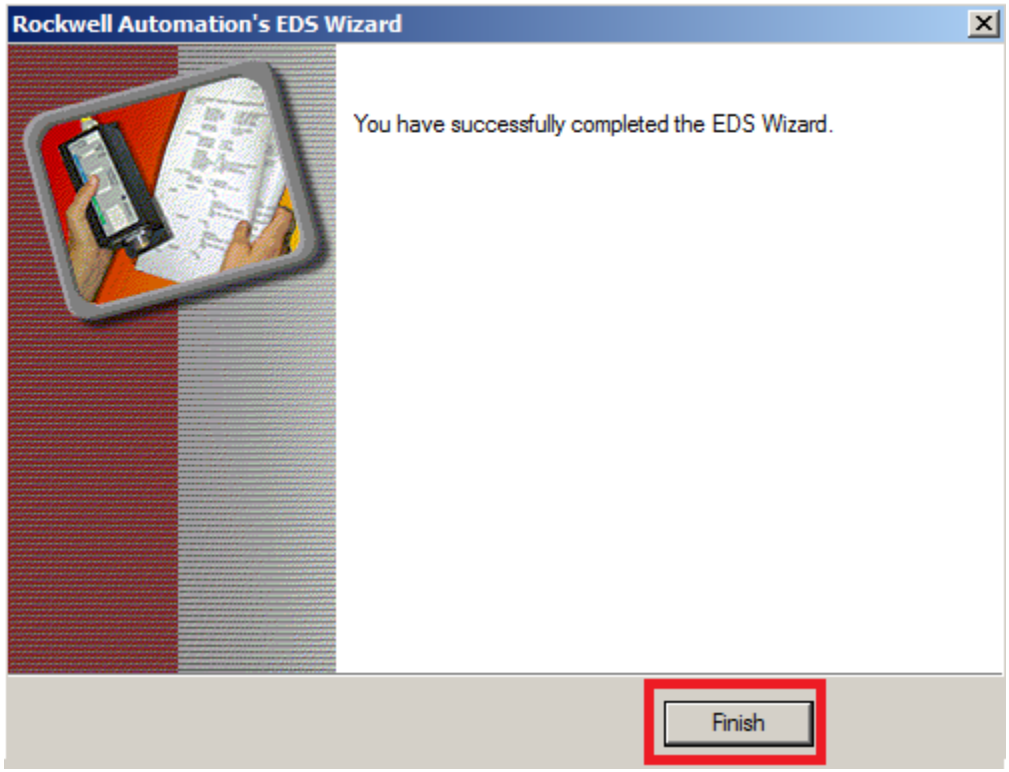
6) Change the icon if needed, click **Next** when completed



7) Select **Next** to finalize the EDS installation process



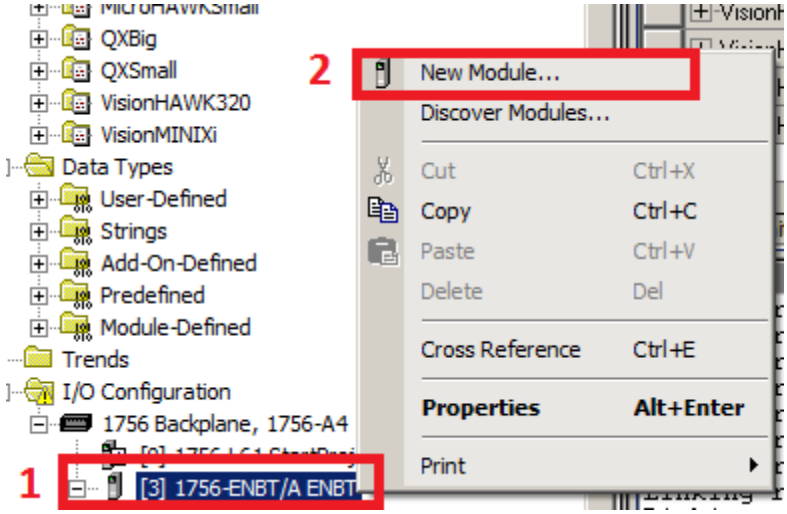
8) Select **Next** to Finish the EDS installation process



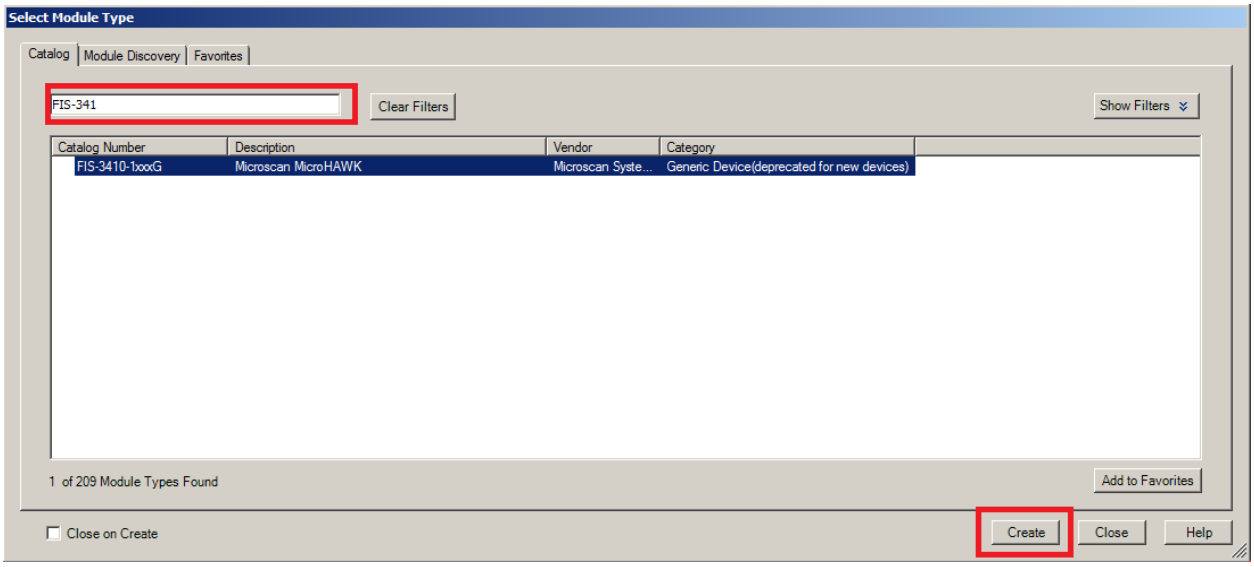
6.2 Adding the MicroHAWK Ethernet Module

This section will cover, adding the MicroHAWK Ethernet Module into your logic program. Please install the EDS file in step 5.1 before proceeding.

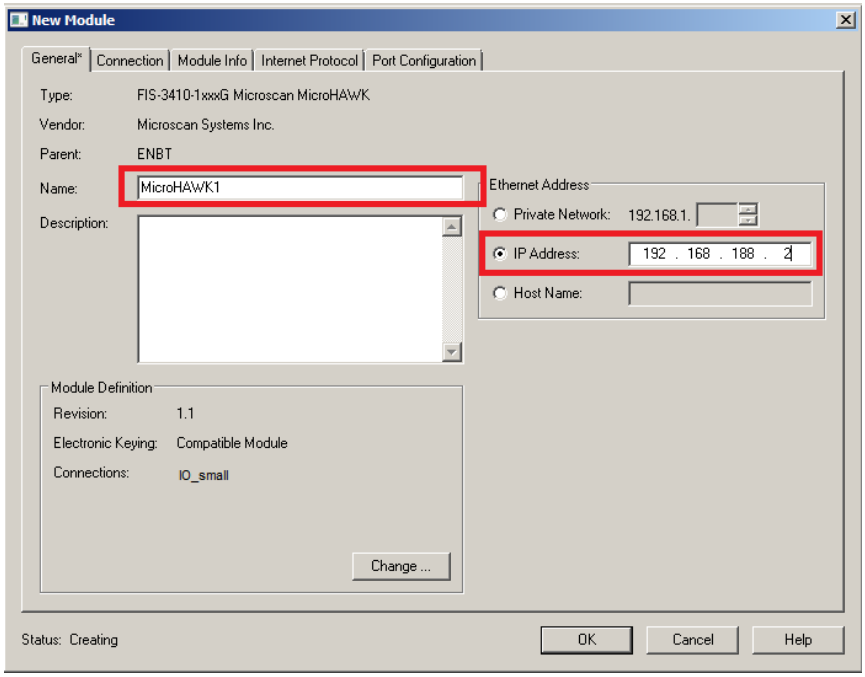
1) Right Click the Ethernet Controller (1) and select New Module (2)



2) In the Search bar type in FIS-3410-1xxxG until you find the MicroHAWK device then click Create

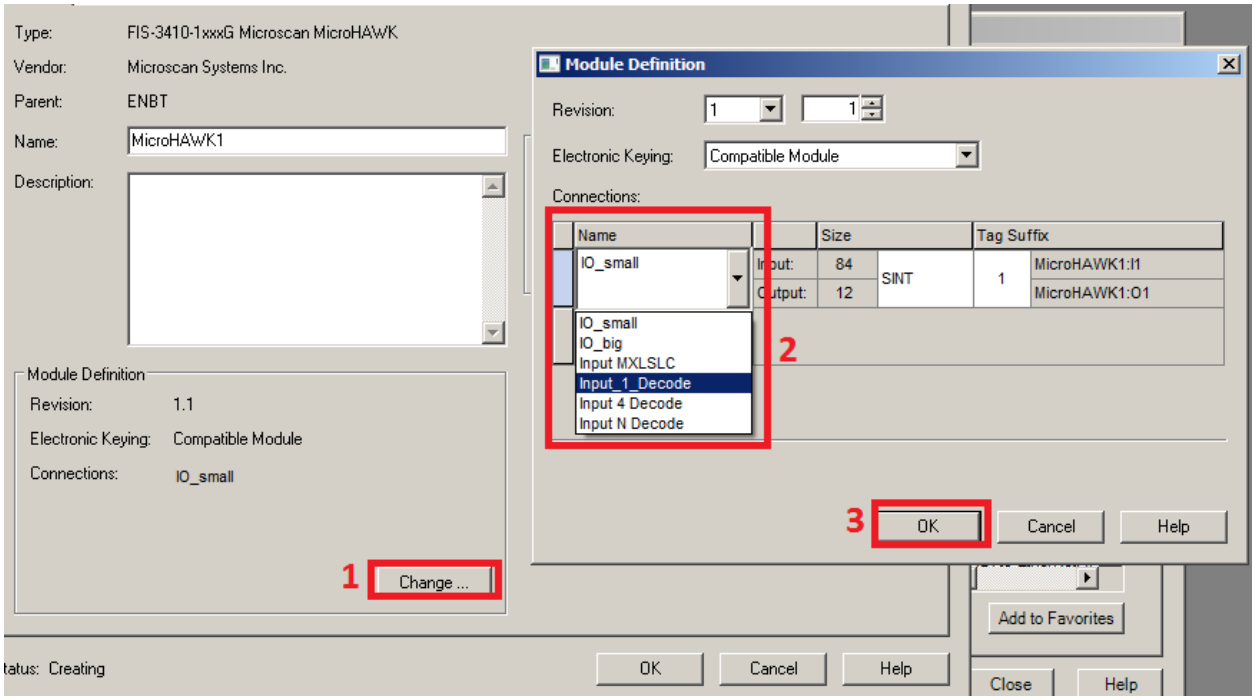


3) Type in a unique name for the unit and the units IP address (Default IP address is 192.168.188.2)

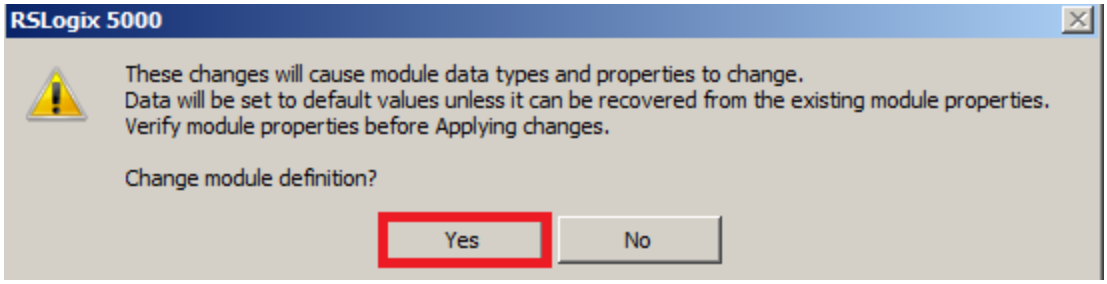


4) Change the Input Assembly by performing the following

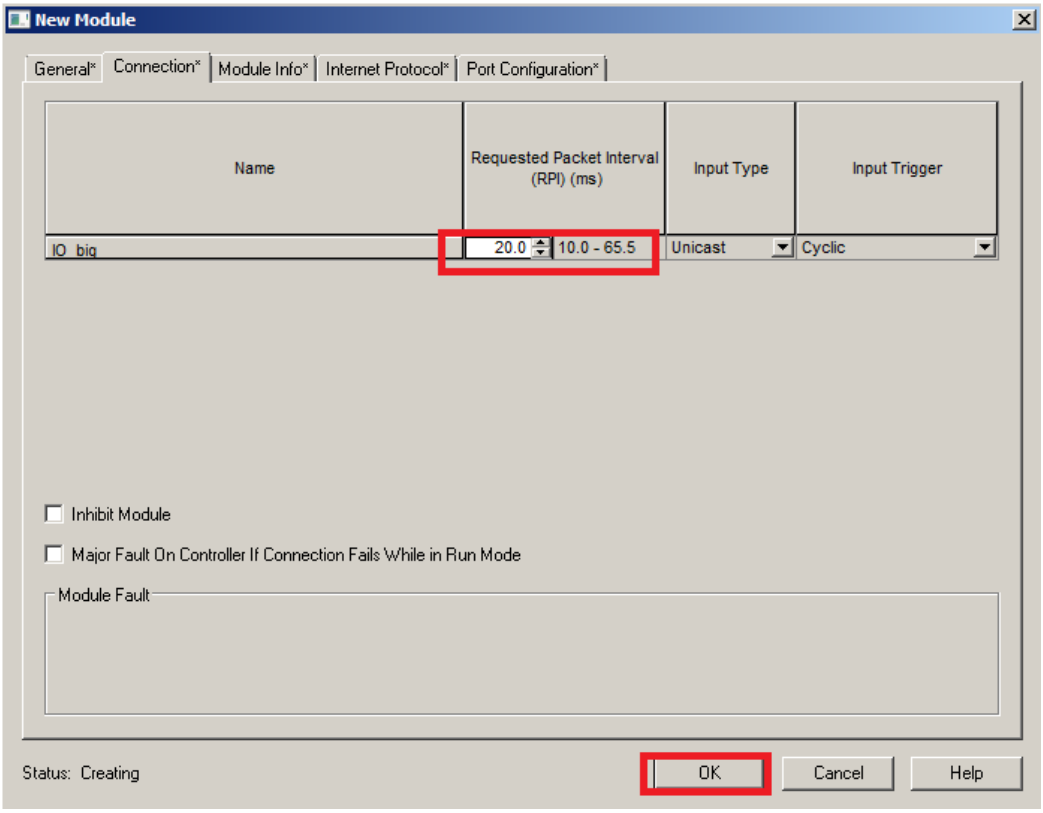
a. Click Change(1) and select the desired Connection Type(2) in the dropdown box. For simplicity, the connections where named the same as the Input Assemblies. This also correctly assigns the output assembly to the correct input assembly so the programmer does not need to worry about this. When done click OK.



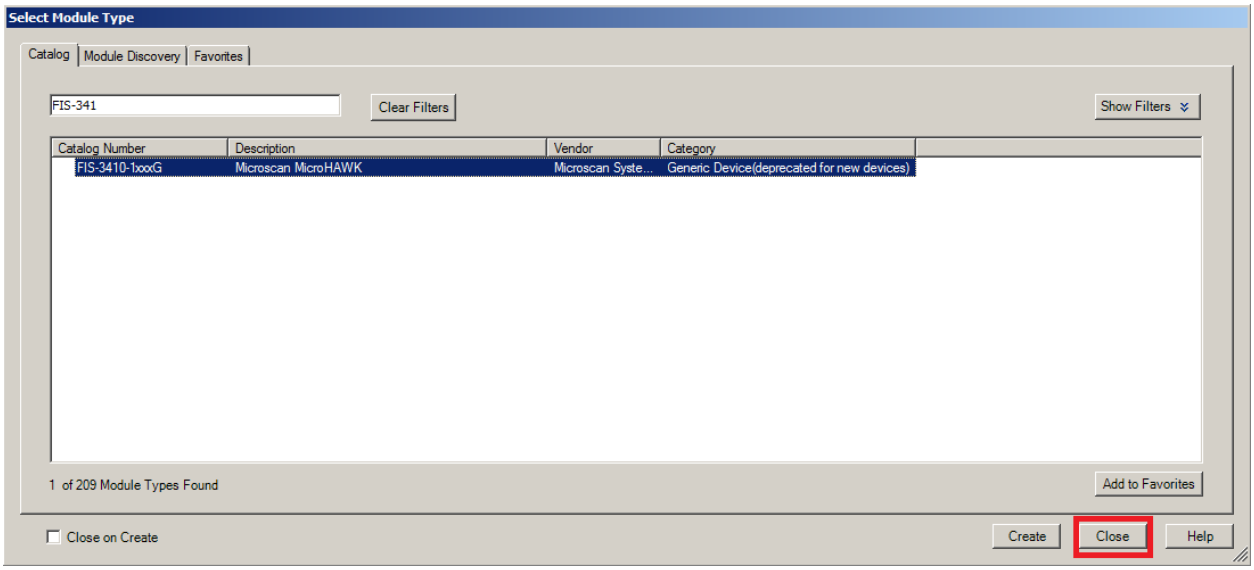
b. Select YES to change module definitions to complete.



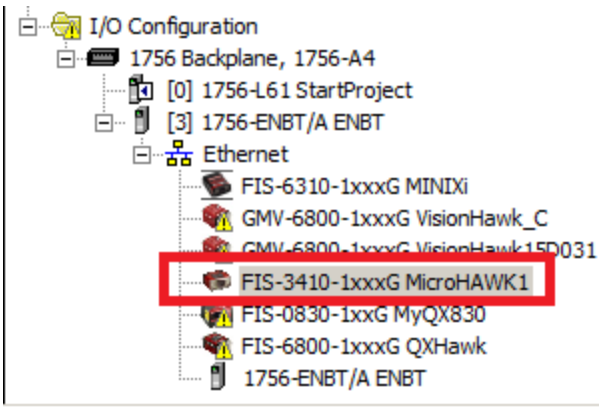
5) In the Connections tab select the RPI that the PLC will use. Default is 20 milliseconds. Click OK to complete the creation of this module
   a. NOTE[1]: Try to make this value roughly the same as the units capture time to avoid heavy network bandwidth usage.



6) Close the Select Module Type window



7) Verify that your unit was created by browsing the Ethernet module tree. In this example, you will see FIS-3410-1xxxG MicroHAWK1 was added to the tree. If you have not downloaded to your PLC, you may do so now.



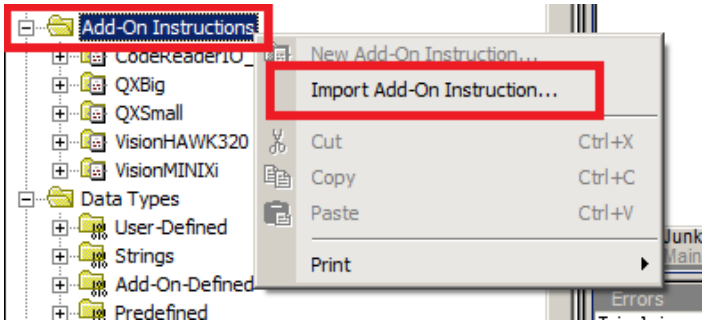
8) This completes adding the MicroHAWK Ethernet Module. Please proceed to Section 6.3 Add-On Instruction Import.

6.3 Add-On Instruction Import

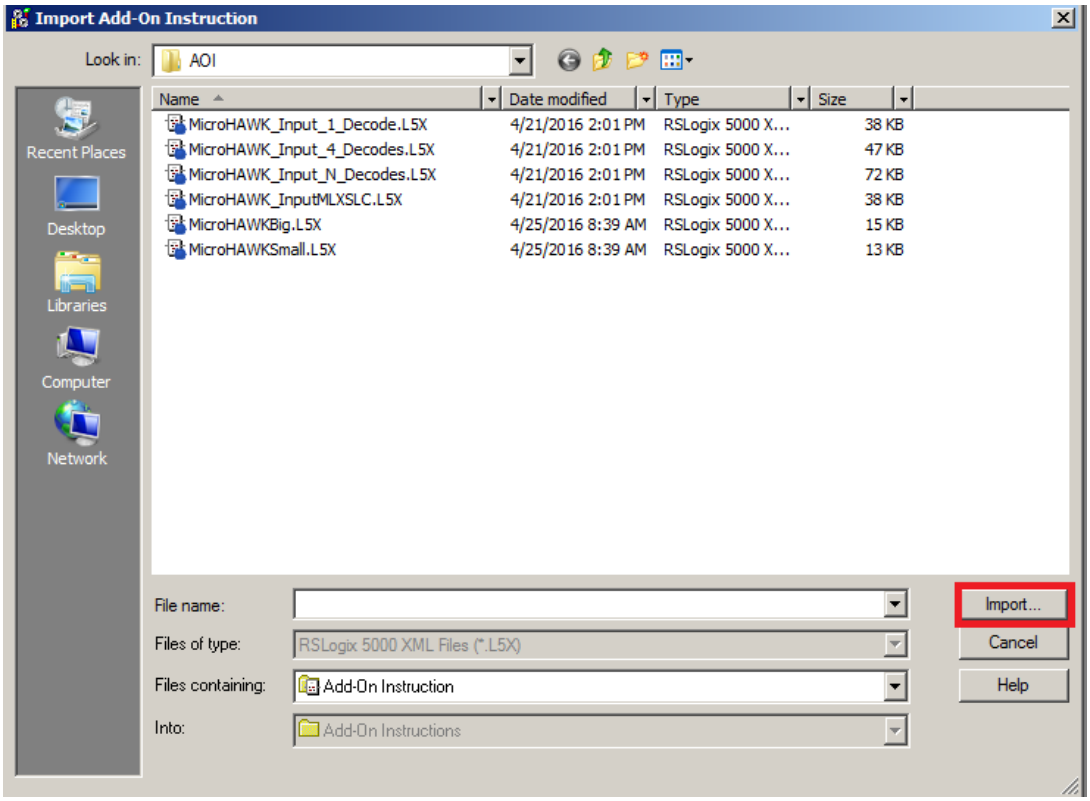
This section describes the importing of the Add-on instruction into the Rockwell Compactlogix platform. Microscan offers several types of Add-on instructions depending on the connection type selected in Section 5 or Section 6. Below are the AOI's the programmer will need to import based on the connection type.

CONNECTION TYPE	AOI FILE NAME
IO_SMALL	MicroHAWKSmall.L5X
IO_BIG	MicroHAWKBig.L5X
INPUT MXLSLC	MicroHAWK_InputMLXSLC.L5X
INPUT 1 DECODE	MicroHAWK_Input_1_Decode.L5X
INPUT 4 DECODE	MicroHAWK_Input_4_Decode.L5X
INPUT N DECODE	MicroHAWK_Input_N_Decode.L5X

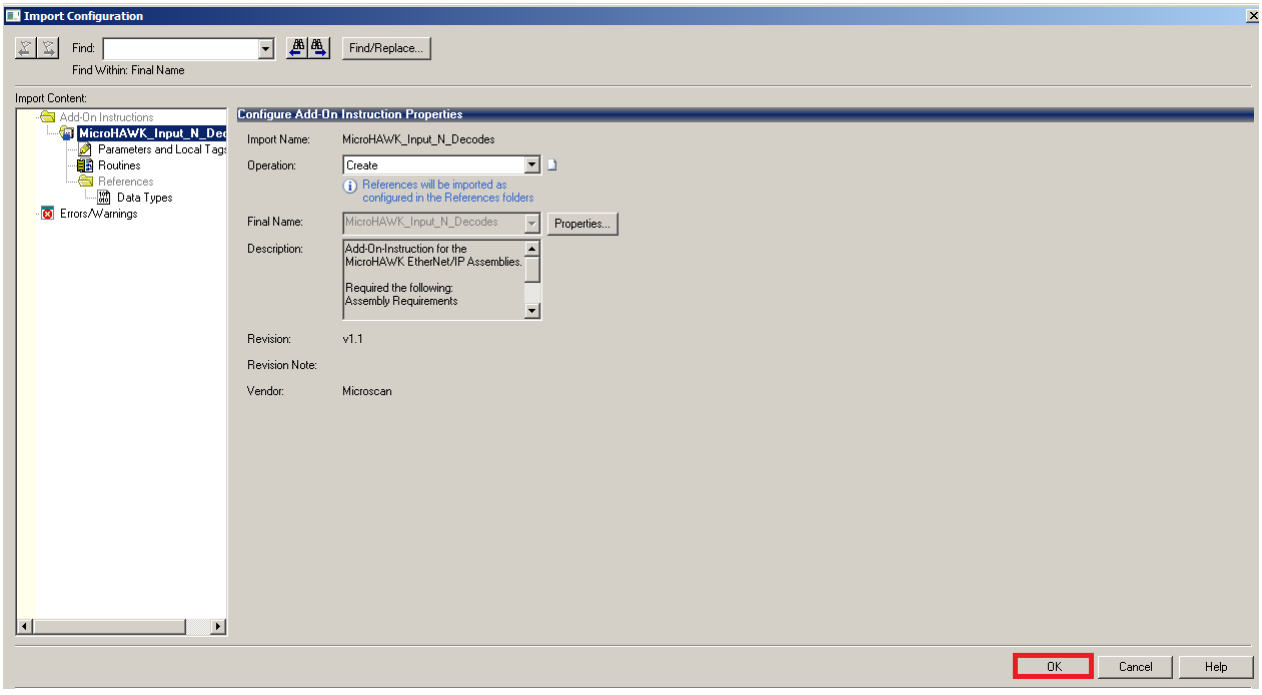
1) In the Controller Organizer Window, right Click the Add-On Instruction folder and select Import Add-On Instructions…



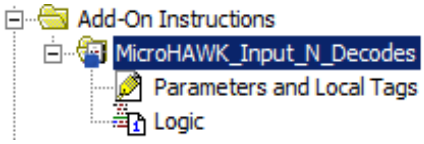
2) Select the correct AOI file you wish to use for your PLC logic and click Import. In this example we will be using the Big IO assembly.



3) In the Import Configurator, select the Errors/Warnings and check that nothing happened during the import. Click OK to continue



4) Verify that the AOI was imported by browsing your Add-On Instructions folder.



5) You are now able to begin programming the PLC with the MicroHAWK. Microscan has provided example routines for the programmer to import into a project. Please reference Section 6 Example Routine Import on the routines and how to import them into the RSLogix 5000 project.

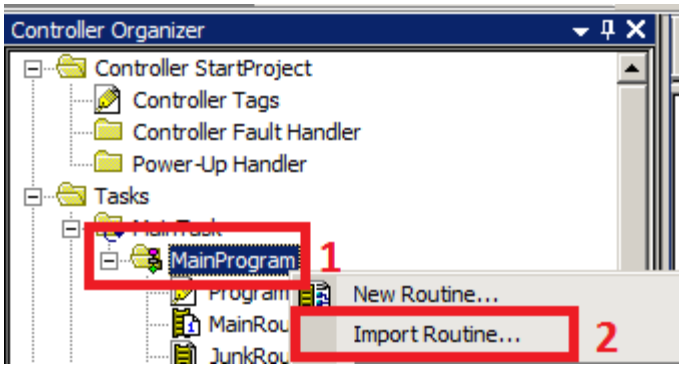
7 Example Routine Import

This section describes how to import the example routine created by Microscan Inc. These example routines will demonstrate implicit and explicit messaging. They also already map the output assemblies to a user data tag the programmer can reference, and use, in throughout the project.. These routines are a good starting point to get the MicroHAWK up and running in the Rockwell Compactlogix platform with minimal startup time. In this example we will import the example routine into the Main program but it could be imported into other areas, such as other programs, Period Task etc…

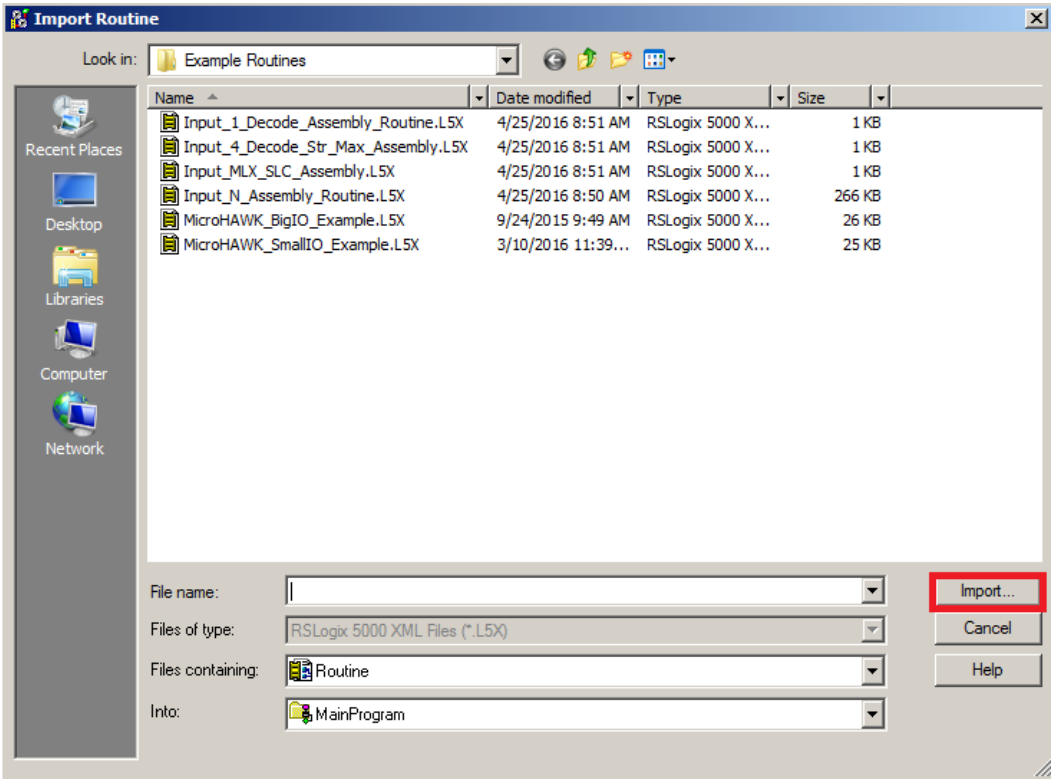
NOTE1: It is the user's responsibility to correctly program the PLC logic. These examples are for demonstration purposes only and are not intended for production release.

NOTE2: This example imports the routine in the MainProgram but the examples can be imported into other areas of the project per the programmer's requirements.

- Start by going to the Controller Organizer window and right click on the MainProgram Folder(1) and selecting Import Routine(2).



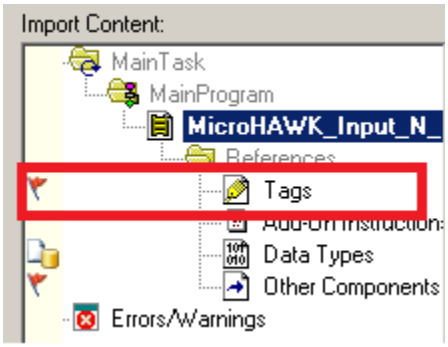
- Navigate to the Example Routines Folder in Microscan Connectivity→Rockwell CompactLogix→Example Routine downloaded from the website and select the desired routine. To help the programmer the table below describes the correct file name associated to the Input/Output Assemblies that were selected. Click IMPORT after selecting the file name.



7.1 Table 6.1 Example Routine file selection

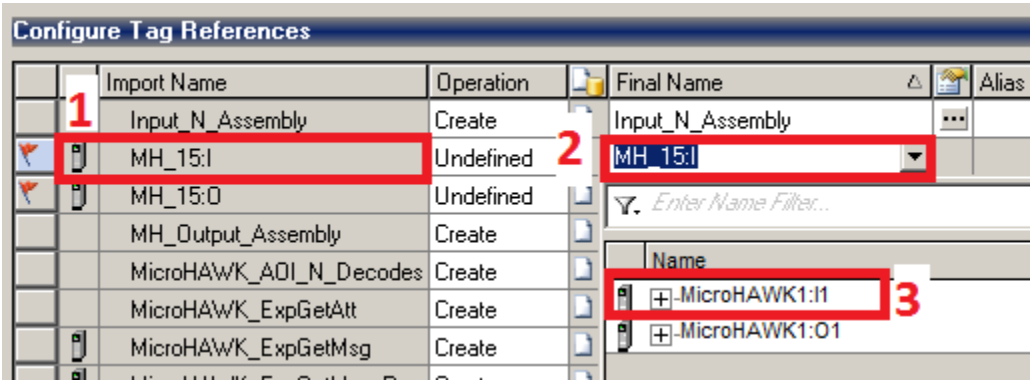
Input/Output Assemblies Used		Example Routine to Import
Input	**Output**	
Small (0x64, Decimal 100)	Output Legacy (0xC6, Decimal 198)	MicroHAWK_SmallIO_Example.L5X
Big (0x65, Decimal 101)	Output Legacy (0xC6, Decimal 198)	MicroHAWK_BigIO_Example.L5X
Input MXL/SLC (0x66, Decimal 102)	Output (0xC5, Decimal 197)	MicroHAWK_Input_MXLSLC_Example.L5X
Input 1 Decode (0x67, Decimal 103)	Output (0xC5, Decimal 197)	MicroHAWK_ Input_Input_1_Example.L5X
Input 1 Decode (0x68, Decimal 104)	Output (0xC5, Decimal 197)	MicroHAWK_ Input_Input_4_Example.L5X
Input 1 Decode (0x69, Decimal 105)	Output (0xC5, Decimal 197)	MicroHAWK_ Input_Input_N_Example.L5X

- In the Import Configuration Window there will be some errors. This is okay as this step will address them below.
    a. Tags



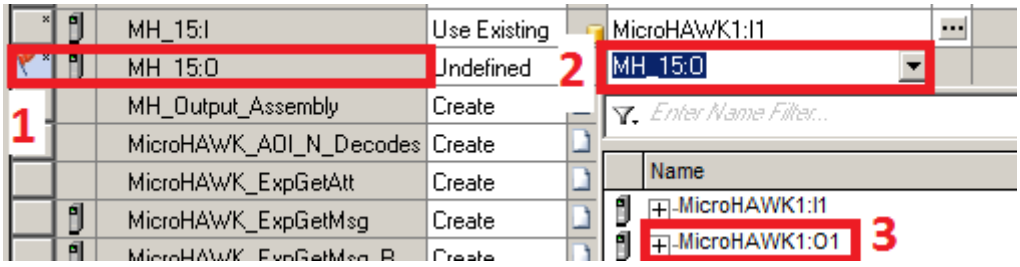
i. Change the Ethernet Module Input tag(1) by clicking the Final Name(2) and selecting the Ethernet Module Input tag for the MicroHAWK.

Example Changing the Ethernet Module Input Data: In this example the Ethernet Module was named **MicroHAWK1** so the input tag that was auto-generated is **MicroHAWK1:I** (ModuleName:I)

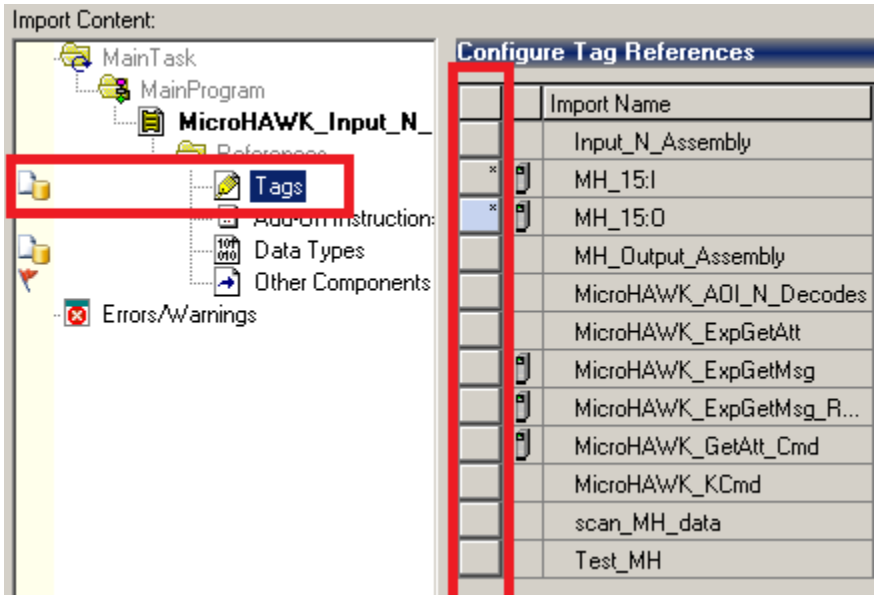


ii. Perform the same action to the Ethernet Module Output tag (1) by clicking the Final Name(2) and selecting the Ethernet Module Output tag((3)

Example Changing the Ethernet Module Output Data: In this example the Ethernet Module was named **MicroHAWK1** so the input tag that was auto-generated is **MicroHAWK1:O** (ModuleName:O)

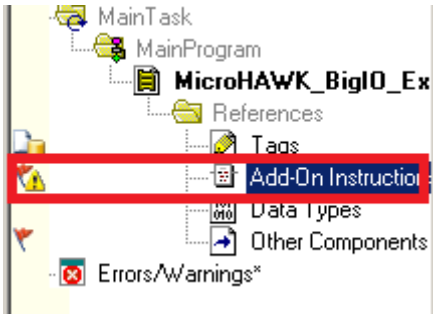


iii. When complete there should be no red flags in the Tags Section

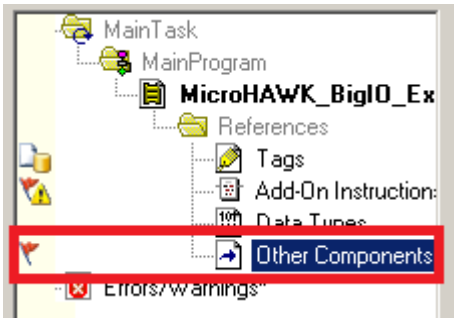


b. Add-On Instructions Section

If the PLC already as the AOI installed from Section 3 than a red flag will be noticed here. This can be ignored as it will use the existing AOI

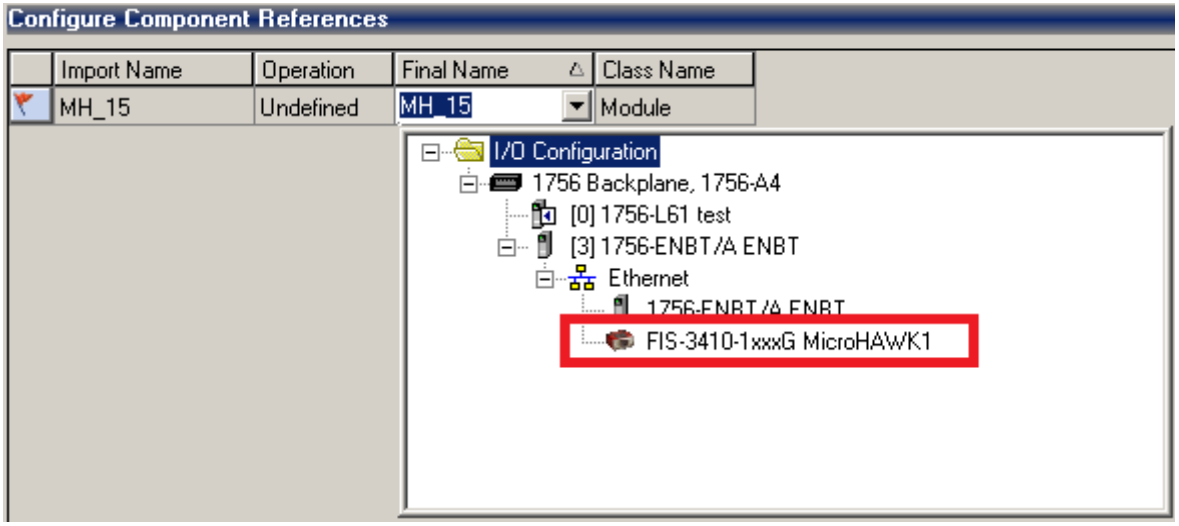


c. Other Components

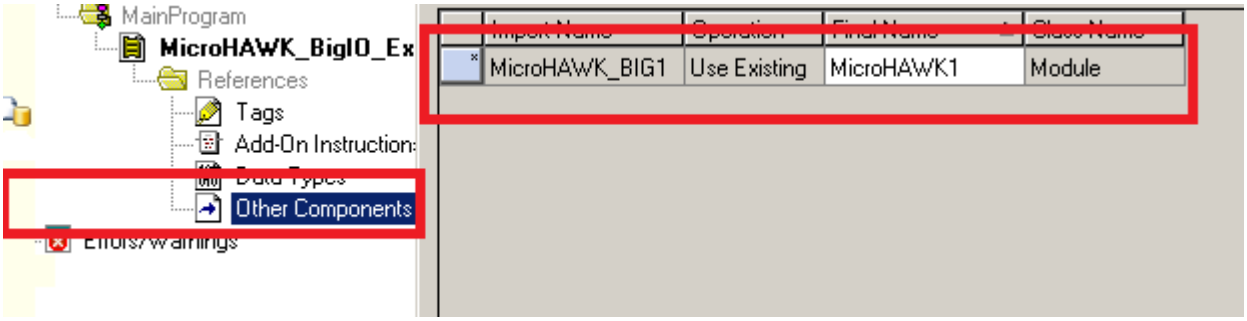


i. Change the Ethernet Module to the module created in Section 4.1 or 5.2

Example Changing the Ethernet Module: In this example the Ethernet Module was named **MicroHAWK1** so the Final Name is **MicroHAWK1** (ModuleName)

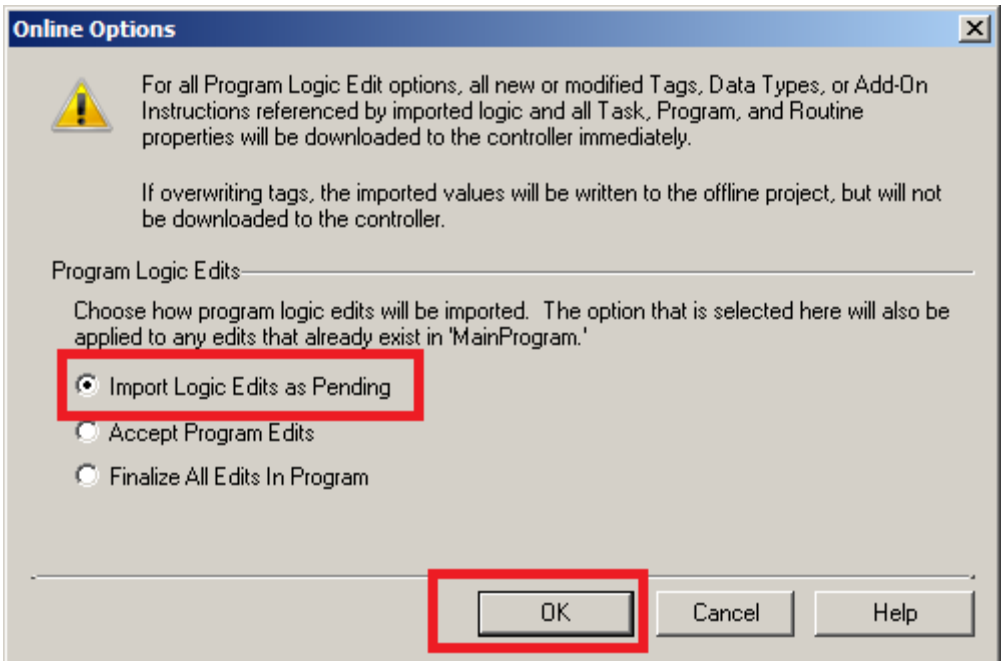


ii. When complete there should be no red flags in the Others Section

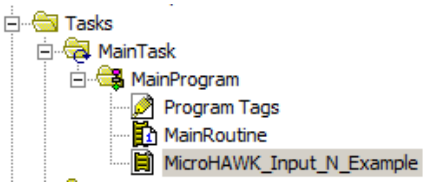


Note[1]: If you have a red flag in Add-on Instructions because you already imported the AOI into your PLC. It is okay to proceed. The PLC will just use the existing AOI in the routine that will be imported.

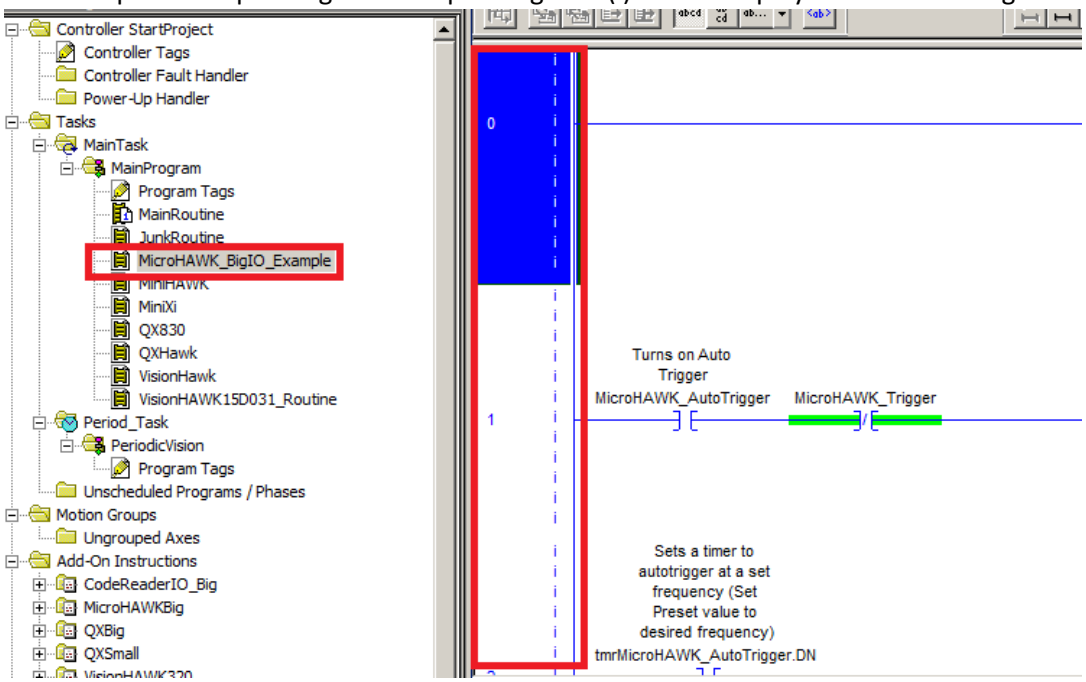
- After addressing all the error's above, Click on OK to Import the routine. If the PLC is **Online** than the **Online Options** windows will display. choose In this example we will select Import Logic Edits as Pending. Click OK to finalize



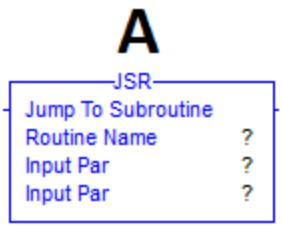
- The Routine Should now be imported to the Program.



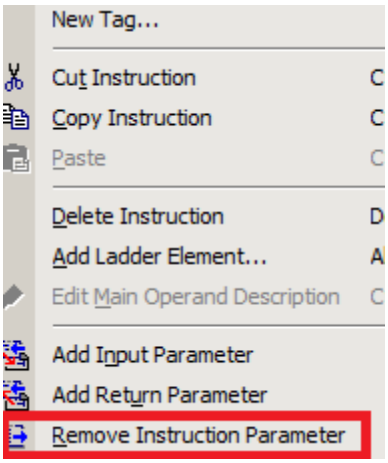
a. If imported as pending than the pending icon (i) will be displayed on all the rungs.



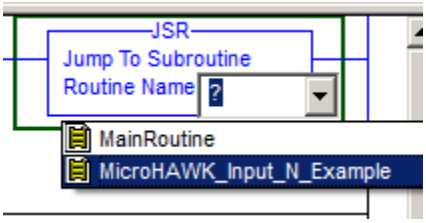
- To execute this routine, insert a JSR (Jump to Subroutine) instruction in an area that the program will execute it. In this example, the JSR is inserted at the MainRoutine by doing the following:
    a. Add a rung  to the bottom of the MainRoutine ladder
    b. Add a JSR instruction (A) to the routine



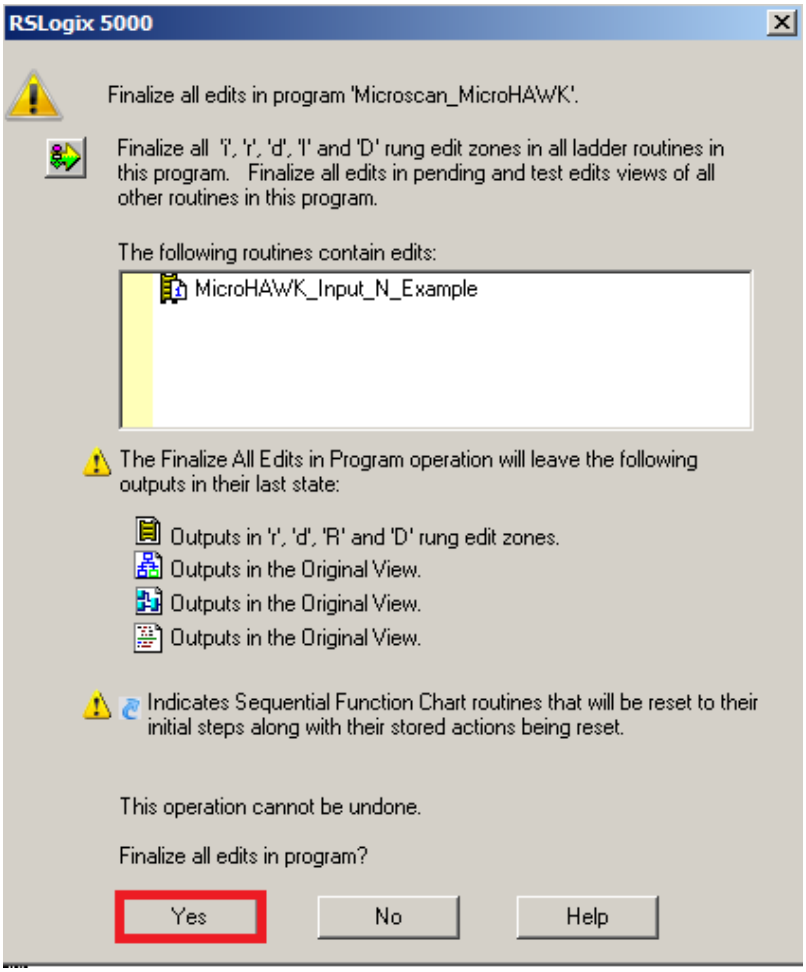
    c. Remove all input parameters as they are not needed for either example routines. You can remove the parameter by right clicking on the Input Par and selecting Remove Instruction Parameter



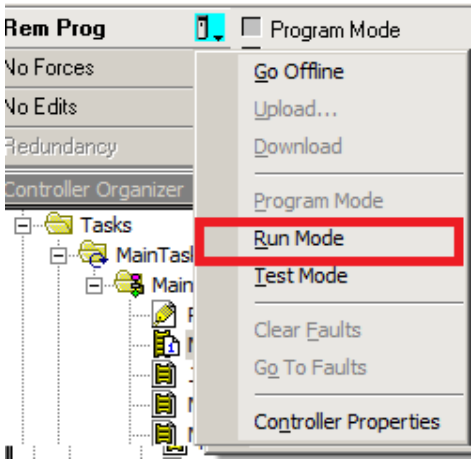
    d. Select the example routine imported (In this example, the MicroHAWK_Input_N_Example was imported).



- If no Error's occurred during the Import Process and the MicroHAWK is powered up and communicating to the PLC than save the pending edits to your PLC by clicking the Finalize all edits button  to begin using the Example Routine.

a. 

- Set the PLC to Run Mode if it is not already.



- The routine will automatically start running and is ready for use. Please refer to Section 7 for each routines description.

8 Supplied Example Routines

This section will describe the example routines supplied by Microscan for the Rockwell CompactLogix controller. Please refer to the Section 3 Using EtherNet/IP for more detailed information regarding the input/output assemblies.

8.1 MicroHAWK_SMALLIO_Example

This section describes the key points in the example routine MicroHAWK_SMALLIO_Example.L5X supplied by Microscan for the MicroHAWK unit. Although there is more logic than described, the programmer should have enough knowledge to know how to read the ladder logic provided and MicroHAWK unit.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The controller passes the information from the user tags into the AOI. The AOI simply performs a CPS from the user tags to the Ethernet Module tags. Inputs will be copied from the module to the user tags and outputs will be copied from the user tags to the module.

Rung 0: This is where the data is copied from the User tags to the Module tags. Inputs will be copied from the user data to the module data and the outputs will be copied from the Module to the user data.

Rung 3: Example on how to trigger the MicroHAWK reader. Notice that the trigger does not check any device status! When using the Small Input Assembly caution should be issued because a trigger could occur when the unit is still decoding which will cause false failures of the previous sequence.

Rung 4: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rung 5: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

8.2 MicroHAWK_BIGIO_Example

This section describes the key points in the example routine MicroHAWK_BIGIO_Example.L5X supplied by Microscan for the MicroHAWK unit. Although there is more logic than described, the programmer should have enough knowledge to know how to read the ladder logic provided and MicroHAWK unit.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The controller passes the information from the user tags into the AOI. The AOI simply performs a CPS from the user tags to the Ethernet Module tags. Inputs will be copied from the module to the user tags and outputs will be copied from the user tags to the module.

Rung 0: This is where the data is copied from the User tags to the Module tags. Inputs will be copied from the user data to the module data and the outputs will be copied from the Module to the user data.

Rung 3: Example on how to trigger the MicroHAWK reader. Notice that the trigger will not be sent if the unit is not Disabled and the unit is not current decoding an image.

Rung 4: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rung 5: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

8.3 MicroHAWK_Input_MXLSLC_Example

This section describes the key points in the example routine MicroHAWK_Input_MXLSLC_Example.L5X supplied by Microscan for the MicroHAWK unit. Although there is more logic than described, the programmer should have enough knowledge to know how to read the ladder logic provided and MicroHAWK unit.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The controller passes the information from the user tags into the AOI. The AOI simply performs a CPS from the user tags to the Ethernet Module tags. Inputs will be copied from the module to the user tags and outputs will be copied from the user tags to the module.

scan_MH_MXLSLC_Data: User tag that is accessible across your program. It is up to the programmer's discretion if they would like to use this tag in their application and/or to make it a controller global tag. Benefit is the tag's already mapped to the output data of the unit, so all the user needs to do is map the data to the application requirements.

Rung 0: This is where the data is copied from the User tags to the Module tags. Inputs will be copied from the user data to the module data and the outputs will be copied from the Module to the user data.

Rung 1: Demonstrates how to trigger the unit through the output command. Notice that before triggering the unit, the rung checks if the device status tag in the input assembly. The logic verifies that the unit is: online; not decoding; and not triggering. It is imperative that additional logic for the user's application is added, so to not receive false negative inspections.

Rung 2: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rung 3: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

Rungs 4 through 14: Map the user data tag **scan_MH_MXLSLC_Data** to the units Ethernet Module data.

Decode Symbol data can be found in **MH_InputMXLSLC_Asm.Decode_Report.Decode_Data**. This is a 184 character long string and is delimited by the delimiter set in the unit. This value is set from the K Command <K222> second parameter **Multi-label Separator**

8.4 MicroHAWK_Input_1_Decode_Example

This section describes the key points in the example routine MicroHAWK_Input_1_Decode_Example.L5X supplied by Microscan for the MicroHAWK unit. Although there is more logic than described, the programmer should have enough knowledge to know how to read the ladder logic provided and MicroHAWK unit.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The controller passes the information from the user tags into the AOI. The AOI simply performs a CPS from the user tags to the Ethernet Module tags. Inputs will be copied from the module to the user tags and outputs will be copied from the user tags to the module.

scan_MH1_Data: User tag that is accessible across your program. It is up to the programmer's discretion if they would like to use this tag in their application and/or to make it a controller global tag. Benefit is the tag's already mapped to the output data of the unit, so all the user needs to do is map the data to the application requirements.

Rung 0: This is where the data is copied from the User tags to the Module tags. Inputs will be copied from the user data to the module data and the outputs will be copied from the Module to the user data.

Rung 1: Demonstrates how to trigger the unit through the output command. Notice that before triggering the unit, the rung checks if the device status tag in the input assembly. The logic verifies that the unit is: online; not decoding; and not triggering. It is imperative that additional logic for the user's application is added, so to not receive false negative inspections.

Rung 2: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rung 3: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

Rungs 4 through 14: Map the user data tag **scan_MH1_Data** to the units Ethernet Module data.

Decode Symbol data can be found in **MH_Input1_Asm.Decode_Report.Decode_Data**. This is a 436 character long string and is delimited by the delimiter set in the unit. This value is set from the K Command <K222> second parameter **Multi-label Separator**

8.5 MicroHAWK_Input_4_Decode_Example

This section describes the key points in the example routine MicroHAWK_Input_4_Decode_Example.L5X supplied by Microscan for the MicroHAWK unit. Although there is more logic than described, the programmer should have enough knowledge to know how to read the ladder logic provided and MicroHAWK unit.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The controller passes the information from the user tags into the AOI. The AOI simply performs a CPS from the user tags to the Ethernet Module tags. Inputs will be copied from the module to the user tags and outputs will be copied from the user tags to the module.

scan_MH4_Data: User tag that is accessible across your program. It is up to the programmer's discretion if they would like to use this tag in their application and/or to make it a controller global tag. Benefit is the tag's already mapped to the output data of the unit, so all the user needs to do is map the data to the application requirements.

Rung 0: This is where the data is copied from the User tags to the Module tags. Inputs will be copied from the user data to the module data and the outputs will be copied from the Module to the user data.

Rung 1: Demonstrates how to trigger the unit through the output command. Notice that before triggering the unit, the rung checks if the device status tag in the input assembly. The logic verifies that the unit is: online; not decoding; and not triggering. It is imperative that additional logic for the user's application is added, so to not receive false negative inspections.

Rung 2: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rung 3: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

Rungs 4 through 14: Map the user data tag **scan_MH4_Data** to the units Ethernet Module data.

Decode Symbol data can be found in 4 locations for this Input Assembly due to this input assembly statically assign 4 locations for each decoded symbol.

- **MH_Input4_Asm.Decode_Report1.Decode_Data**. This is a 160 maximum character long string
- **MH_Input4_Asm.Decode_Report2.Decode_Data**. This is a 72 maximum character long string
- **MH_Input4_Asm.Decode_Report3.Decode_Data**. This is a 72 maximum character long string
- **MH_Input4_Asm.Decode_Report4.Decode_Data**. This is a 72 maximum character long string

8.6 MicroHAWK_Input_N_Decode_Example

This section describes the key points in the example routine MicroHAWK_Input_N_Decode_Example.L5X supplied by Microscan for the MicroHAWK unit. Although there is more logic than described, the programmer should have enough knowledge to know how to read the ladder logic provided and MicroHAWK unit.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The controller passes the information from the user tags into the AOI. The AOI simply performs a CPS from the user tags to the Ethernet Module tags. Inputs will be copied from the module to the user tags and outputs will be copied from the user tags to the module.

scan_MH_Data: User tag that is accessible across your program. It is up to the programmer's discretion if they would like to use this tag in their application and/or to make it a controller global tag. Benefit is the tag's already mapped to the output data of the unit, so all the user needs to do is map the data to the application requirements.

Rung 0: This is where the data is copied from the User tags to the Module tags. Inputs will be copied from the user data to the module data and the outputs will be copied from the Module to the user data.

Rung 1: Demonstrates how to trigger the unit through the output command. Notice that before triggering the unit, the rung checks if the device status tag in the input assembly. The logic verifies that the unit is: online; not decoding; and not triggering. It is imperative that additional logic for the user's application is added, so to not receive false negative inspections.

Rung 2: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rung 3: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

Rungs 4 through 14: Map the user data tag **scan_MH_Data** to the units Ethernet Module data.

Decode Symbol data can be found in the **Input_N_Assembly.Decode_Report_List**. The list contains 20 Decode reports with a maximum string of 436 characters. Each report contains information regarding a specific decode string. In **Input_N_Assembly.Read_Cycle_Report** the tag **Number_Of_Decode_Reports** informs the program of how many Decode Reports are in the list. The Read Cycle Report also states how many decoded symbols were found as well as some diagnostic data of Capture time, total decode time and total read cycle time (in milliseconds). The AOI **MicroHAWK_Input_N_Decodes** parses all the information from the unit and places this information in to the list without any additional programing from the programmer. If the controller does not support the Add-On-Instructions than please review the structured text routine and create a routine in the project. The structured text routine can be used in the project with some slight modifications.

8.7 MicroHAWK_Explicit_Example

This section describes the key points in the example routine MicroHAWK_Explicit_Example.L5X supplied by Microscan for the MicroHAWK unit. In some use cases the explicit messaging would need to be used, such as, if the controller being used does not support implicit messages.

It is up to the programmer on how to integrate the device to the controller in the RSLogix program. The example routine mimics a producer/consumer model by writing data from the device to the controller and data from the controller to the device using an explicit messaging. Data from the device is passed through an Add-On-Instruction to parse the data and put into more manageable user tags that the program can use.

NOTE[1]: The table below states what each assembly supports in regards to service types

ASSEMBLY NAME	GET (0X0E)	SET (0X10)	GET/SET (0X45)
INPUT SMALL	Yes	No	No
INPUT BIG	Yes	No	No
INPUT MXL/SLC	Yes	No	No
INPUT 1 DECODE	Yes	No	No
INPUT 4 DECODE	Yes	No	No
INPUT N DECODE	Yes	No	No
OUTPUT (LEGACY)	Yes	Yes	No
OUTPUT	Yes	Yes	No
SERIAL COMMANDS	Yes	Yes	Yes

NOTE[2]: The supplied example uses the Input 1 Decode Assembly described in Section 3.12 MicroHAWK Input 1 Decode Assembly, the Output Assembly as described in Section 3.16 Output Assembly and the Serial Command Assembly described in Section 3.17 Serial Command Assembly.

8.7.1 Example Tag Definitions:

ExpInputArray: A SINT array with a length of 500 bytes (SINT[500]). This input array is where the information is store from the explicit message **MicroHAWK_GetInputMsg** in rung 4

ExpOutputArray: A SINT array with a length of 500 bytes (SINT[500]). This Output array is where the information is passed from the application data to the device in an explicit message instruction **MicroHAWK_SetOutputMsg** in rung 5

user_MH_MLXSLC_tag: User tag that is accessible across your program. It is up to the programmer's discretion if they would like to use this tag in their application and/or to make it a controller global tag. Benefit is the tag's already mapped to the output data of the unit, so all the user needs to do is map the data to the application requirements.

8.7.2 Example Program

Rung 0: This is where the data is copied from the User tags to the controller tags **ExpInputArray** and **ExpOutputArray**. Data in these arrays will be sent to the device after their respective timers are done.

Rung 1: Updates the polling timers to a new preset value. These timers can be updated to any value but be aware that a slower timer might miss some important data transactions. It's suggested to keep these timers to around 5 milliseconds.

Rung 2 - 3: Polling timers. When the timers reach DONE, then the explicit messages for their respective flags will be issues.

Rung 4: When the timer **tmr_MHExpInput** DONE flag is raised an explicit message **MicroHAWK_GetInputMsg** is sent. This receiving data will populate the temp SINT array **ExpInputArray** with the raw data. Rung 0 will parse this data into the user data tag **user_MH_MLXSLC_tag**. The timer is than reset so begin the next polling cycle.

Rung 5: When the timer **tmr_MHExpOutput** DONE flag is raised an explicit message **MicroHAWK_SetOutputMsg** is sent. This sends data from the temp SINT array **ExpOutputArray** to the device. The timer is than reset so begin the next polling cycle.

Rung 6: Example on how to send a K Command to the device. For firmware versions greater than 1.1, the unit will now properly respond to a command if applicable.

Example: Sending a <?> will have a valid response <?/…>

Rung 7: Demonstrates how to trigger the unit through the output command. Notice that before triggering the unit, the rung checks if the device status tag in the user data. The logic verifies that the unit is: online; not decoding; and not triggering. It is imperative that additional logic for the user's application is added, so to not receive false negative inspections.

Rung 8: Example on how to unlatch a user trigger tag after the unit has responded with an acknowledgment. The unit is triggered on a rising edge so it is important to reset the trigger bit when done.

Rungs 9 through 19: Map the user data tag **user_MH_MLXSLC_tag** to the units Ethernet Module data.

Rung 20: Copies the mapped user output data to the temp SINT array **ExpOutputArray.** This will then be sent to the device on the next polling cycle.