MICROSCAN

MicroHAWK ID-40

Connectivity for MicroHAWK ID-40 with the Omron CJ2 Series

Version2.1
12/09/2016

1 Contents

1 Introduction

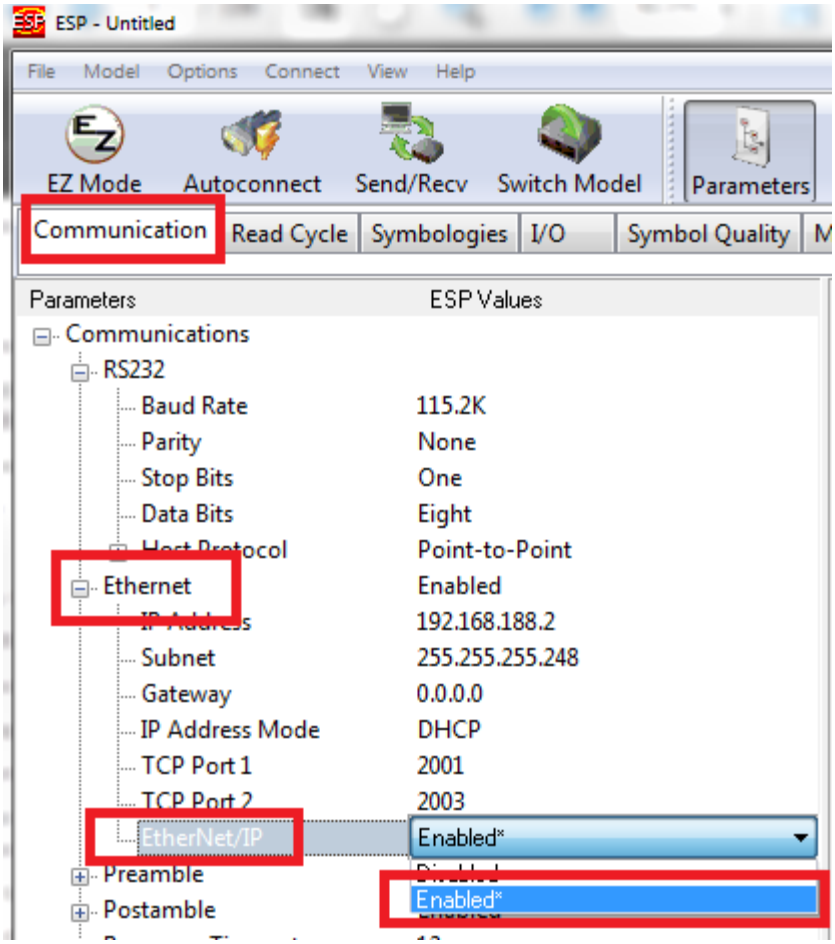
This guide explains how to setup the Microscan MicroHAWK Auto-ID product with the Omron CJ2 Series CPU. All files required for setup can be found on your MicroHAWK unit in the Industrial Protocols/EthernetIP/Omron CJ2. Files are also downloadable at: http://www.microscan.com/en-us/ServiceAndSupport/DownloadCenter.aspx

2 Protocol Switching in ESP and Weblink

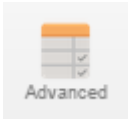
This section describes how to enable EtherNet/IP in ESP and Weblink.

2.1 ESP

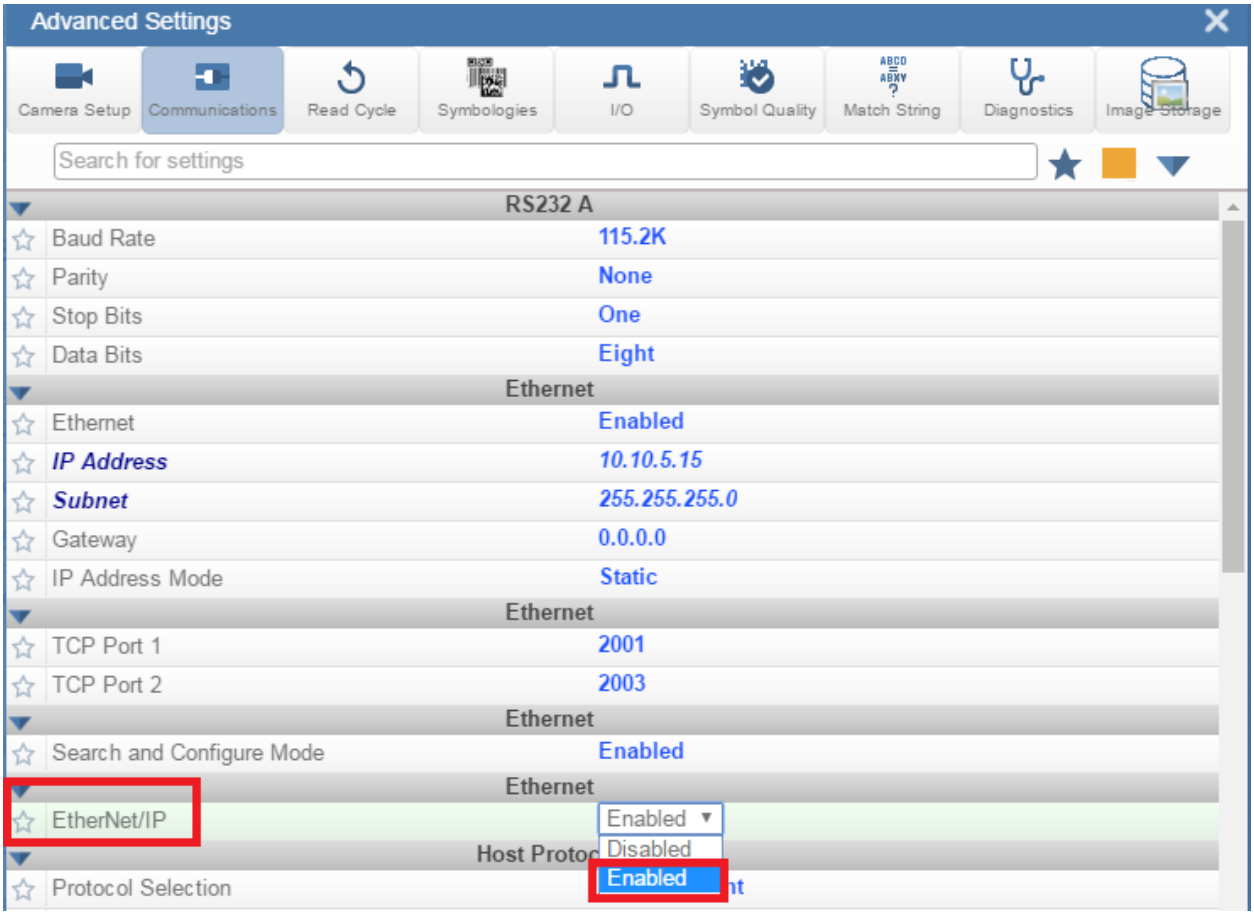
Go to the communications tab in ESP and under Ethernet there will be a node called EtherNet/IP. To the right click the dropdown box and select **Enabled**.



2.2 Weblink

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

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

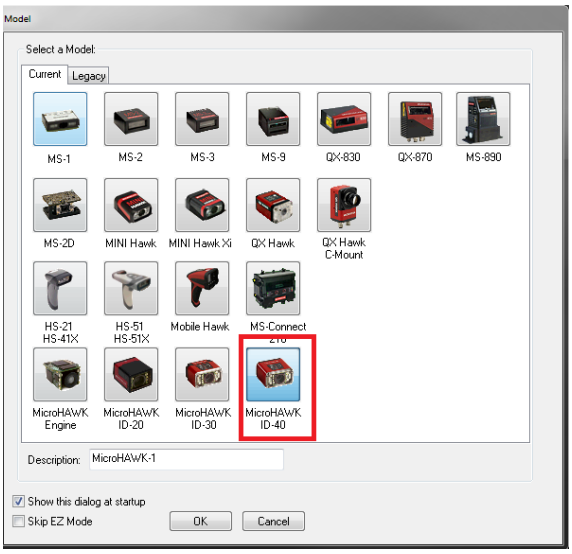


3 Update MicroHAWK Byte Swapping

To read data correctly in the Omron CJ2 controller, the unit will need to enable byte swapping. To do this follow these steps.

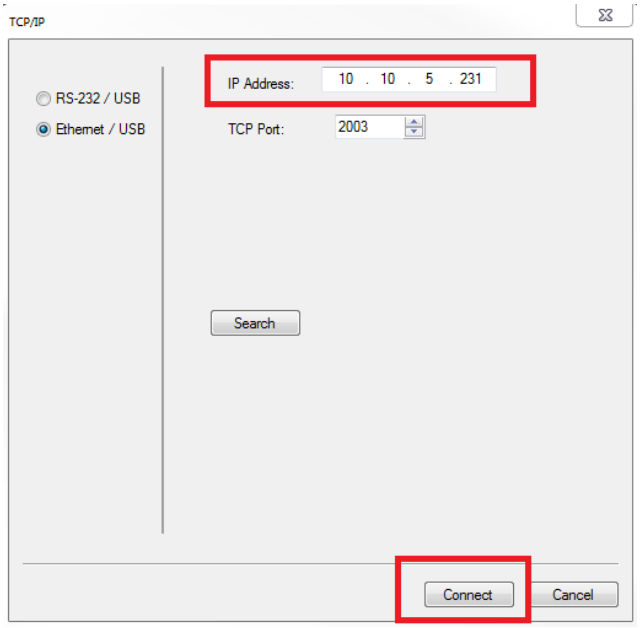
3.1 Open ESP

Open ESP and Select MicroHAWK ID-40

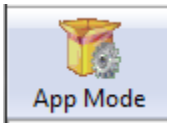


3.2 Connect to MicroHAWK

Connect to your MicroHAWK ID-40 using the correct IP Address and clicking **Connect**



3.3 App Mode

Go to App mode by clicking the App Mode button 

3.4 Terminal

Go to the terminal by clicking the Terminal button 

3.5 Ethernet/IP Byte Swapping K Command

Type the K command <K163,1> to enable byte swapping. This is required to read the data correctly in the Ormon PLC.

NOTE: This command is not transferable at the current release. This means one cannot copy the configuration file from one reader to the next. This command will need to be issued for each new MicroHAWK added to the PLC.

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). The interface is designed to support remote serial transmit and receive using implicit and explicit messaging.

4.2 Necessary Tools

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

- EtherNet/IP Messaging Tool – can be a PLC or Software Tool, must be capable of sending 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

The MicroHAWK EtherNet/IP device type is 0x00, Generic Device.

4.5 Vendor ID

Microscan's Vendor ID is 1095

4.6 Product code

The Product code for the MicroHAWK is 3410

4.7 MicroHAWK EtherNet/IP Object Model

The MicroHAWK uses Class 1 connected messaging to communicate most data in eight different IO assemblies. The user chooses one of six input assemblies, and one of two output assemblies, to create a Class 1 connection.

4.7.1 Connection Properties Supported

Class: 1
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

4.8 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

4.9 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

4.9.1 Table 3.11.1 Input Big Assembly Table

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

Total Size: 84 Bytes

4.9.2 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.

4.9.3 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.

4.9.4 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.

4.9.5 Decode Length

The number of characters found in the decode string

4.9.6 Decode Data

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

4.9.7 Assembly Member Location

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

4.9.7.1 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**

4.10 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.

4.10.1 Table 3.11.1 Input Big Assembly Table

SHORT DESCRIPTION	SIZE (BYTES)
USER-DEFINED TAG ECHO	4
COMMAND ECHO	4
OUPUT 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

4.10.2 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.

4.10.3 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.

4.10.4 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.

4.10.5 External Input Status

The current status of the physical input pins on the unit

4.10.5.1 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

4.10.6 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

4.10.7 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

4.10.8 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.

4.10.9 Trigger Counter

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

4.10.10 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

4.10.11 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

4.10.12 NoRead Counter

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

4.10.13 Decode Length

The number of characters found in the decode string

4.10.14 Decode Data

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

4.10.15 Assembly Member Location

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

4.10.15.1 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**

4.11 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 used in the configuration tag (C→ReadCycle.Multi→Symbol Delimiter).

4.11.1 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

4.11.2 Input Assembly Description

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

4.11.3 Input Assembly Module Header

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

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

4.11.3.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.11.3.2 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.

4.11.3.3 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.

4.11.3.4 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.

4.11.4 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.

4.11.5 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.

4.11.5.1 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	Buffer Overflow
21-31	Reserved

4.11.5.2 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.

4.11.5.3 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.

4.11.5.4 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.

4.11.5.5 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.

4.11.5.6 Data is Ready

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

4.11.5.7 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.

4.11.5.8 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.

4.11.5.9 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.

4.11.5.10 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.

4.11.5.11 Match Code Enabled

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

4.11.5.12 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.

4.11.5.13 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.

4.11.5.14 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.

4.11.5.15 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.

4.11.5.16 Optimizing

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

4.11.5.17 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.

4.11.5.18 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.

4.11.5.19 AutoImage Photometry Complete

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

4.11.5.20 Output 1 Status

Current status of the physical output 1 signal

4.11.5.21 Output 2 Status

Current status of the physical output 2 signal

4.11.5.22 Output 3 Status

Current status of the physical output 3 signal

4.11.5.23 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.

4.11.6 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.

4.11.7 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.

4.11.7.1 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.

4.11.7.2 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

4.11.7.3 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

4.11.7.4 NoRead Counter

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

4.11.7.5 Trigger Counter

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

4.11.7.6 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

4.11.7.7 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

4.11.8 Read Cycle Report

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

4.11.8.1 Table 3.11.4 Read Cycle Report Data

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

4.11.8.2 Capture Time

Total time it tool to capture the image

4.11.8.3 Total Decode Time

Total time spent decoding the symbol(s)

4.11.8.4 Total ReadCycle Time

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

4.11.9 Decode Cycle Report

Information on the decoded symbol

4.11.9.1 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

4.11.9.2 Decode Location Top

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

4.11.9.3 Decode Location Left

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

4.11.9.4 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.

4.11.9.5 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.

4.11.9.6 Code Type

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

4.11.9.6.1 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

4.11.9.7 Pixels Per Element

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

4.11.9.8 Decode Length

The number of characters found in the decode string

4.11.10 Decode Length

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

4.11.11 Decode Data

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

4.11.12 Assembly Member location

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

4.11.12.1 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**

4.12 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 used in the configuration tag (C→ReadCycle.Multi→Symbol Delimiter).

4.12.1 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

4.12.2 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.

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

4.12.3.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.12.3.2 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.

4.12.3.3 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.

4.12.3.4 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.

4.12.4 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.

4.12.5 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.

4.12.5.1 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	Buffer Overflow
21-31	Reserved

4.12.5.2 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.

4.12.5.3 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.

4.12.5.4 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.

4.12.5.5 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.

4.12.5.6 Data is Ready

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

4.12.5.7 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.

4.12.5.8 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.

4.12.5.9 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.

4.12.5.10 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.

4.12.5.11 Match Code Enabled

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

4.12.5.12 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.

4.12.5.13 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.

4.12.5.14 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.

4.12.5.15 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.

4.12.5.16 Optimizing

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

4.12.5.17 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.

4.12.5.18 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.

4.12.5.19 AutoImage Photometry Complete

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

4.12.5.20 Output 1 Status

Current status of the physical output 1 signal

4.12.5.21 Output 2 Status

Current status of the physical output 2 signal

4.12.5.22 Output 3 Status

Current status of the physical output 3 signal

4.12.5.23 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.

4.12.6 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.

4.12.6.1 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

4.12.7 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.

4.12.7.1 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.

4.12.7.2 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

4.12.7.3 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

4.12.7.4 NoRead Counter

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

4.12.7.5 Trigger Counter

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

4.12.7.6 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

4.12.7.7 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

4.12.8 Read Cycle Report

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

4.12.8.1 Table 3.12.5 Read Cycle Report Data

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

4.12.8.2 Capture Time

Total time it tool to capture the image

4.12.8.3 Total Decode Time

Total time spent decoding the symbol(s)

4.12.8.4 Total ReadCycle Time

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

4.12.9 Decode Cycle Report

Information on the decoded symbol

4.12.9.1 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

4.12.9.2 Decode Location Top

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

4.12.9.3 Decode Location Left

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

4.12.9.4 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.

4.12.9.5 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.

4.12.9.6 Code Type

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

4.12.9.6.1 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

4.12.9.7 Pixels Per Element

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

4.12.10 Decode Length

The number of characters found in the decode string

4.12.11 Decode Data

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

4.12.12 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**

4.13 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: The inspection will not need to have 4 decode symbols to use this input assembly.

4.13.1 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

4.13.2 Input Assembly Description

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

4.13.3 Input Assembly Module Header

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

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

4.13.3.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.13.3.2 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.

4.13.3.3 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.

4.13.3.4 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.

4.13.4 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.

4.13.5 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.

4.13.5.1 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	Buffer Overflow
21-31	Reserved

4.13.5.2 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.

4.13.5.3 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.

4.13.5.4 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.

4.13.5.5 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.

4.13.5.6 Data is Ready

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

4.13.5.7 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.

4.13.5.8 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.

4.13.5.9 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.

4.13.5.10 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.

4.13.5.11 Match Code Enabled

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

4.13.5.12 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.

4.13.5.13 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.

4.13.5.14 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.

4.13.5.15 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.

4.13.5.16 Optimizing

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

4.13.5.17 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.

4.13.5.18 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.

4.13.5.19 AutoImage Photometry Complete

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

4.13.5.20 Output 1 Status

Current status of the physical output 1 signal

4.13.5.21 Output 2 Status

Current status of the physical output 2 signal

4.13.5.22 Output 3 Status

Current status of the physical output 3 signal

4.13.5.23 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.

4.13.6 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.

4.13.6.1 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

4.13.7 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.

4.13.7.1 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.

4.13.7.2 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

4.13.7.3 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

4.13.7.4 NoRead Counter

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

4.13.7.5 Trigger Counter

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

4.13.7.6 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

4.13.7.7 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

4.13.8 Read Cycle Report

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

4.13.8.1 Table 3.13.5 Read Cycle Report Data

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

4.13.8.2 Capture Time

Total time it tool to capture the image

4.13.8.3 Total Decode Time

Total time spent decoding the symbol(s)

4.13.8.4 Total ReadCycle Time

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

4.13.9 Decode 1 Cycle Report

Information on the decoded symbol

4.13.9.1 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

4.13.9.2 Decode Location Top

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

4.13.9.3 Decode Location Left

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

4.13.9.4 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.

4.13.9.5 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.

4.13.9.6 Code Type

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

4.13.9.6.1 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

4.13.9.7 Pixels Per Element

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

4.13.10 Decode 1 Length

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

4.13.11 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.

4.13.12 Decode 2 Cycle Report

Information on the decoded symbol

4.13.12.1 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

4.13.12.2 Decode Location Top

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

4.13.12.3 Decode Location Left

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

4.13.12.4 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.

4.13.12.5 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.

4.13.12.6 Code Type

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

4.13.12.6.1 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

4.13.12.7 Pixels Per Element

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

4.13.13 Decode 2 Length

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

4.13.14 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.

4.13.15 Decode 3 Cycle Report

Information on the decoded symbol

4.13.15.1 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]**	

4.13.15.2 Decode Location Top

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

4.13.15.3 Decode Location Left

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

4.13.15.4 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.

4.13.15.5 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.

4.13.15.6 Code Type

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

4.13.15.6.1 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

4.13.15.7 Pixels Per Element

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

4.13.16 Decode 3 Length

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

4.13.17 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.

4.13.18 Decode 4 Cycle Report

Information on the decoded symbol

4.13.18.1 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

4.13.18.2 Decode Location Top

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

4.13.18.3 Decode Location Left

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

4.13.18.4 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.

4.13.18.5 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.

4.13.18.6 Code Type

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

4.13.18.6.1 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

4.13.18.7 Pixels Per Element

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

4.13.19 Decode 4 Length

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

4.13.20 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.

4.13.21 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**

4.14 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).

4.14.1 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

4.14.2 Input Assembly Description

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

4.14.3 Input Assembly Module Header

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

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

4.14.3.1 Run Mode

0 = not Run Mode, 1 = Run Mode

4.14.3.2 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.

4.14.3.3 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.

4.14.3.4 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.

4.14.4 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.

4.14.5 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.

4.14.5.1 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

4.14.5.2 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.

4.14.5.3 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.

4.14.5.4 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.

4.14.5.5 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.

4.14.5.6 Data is Ready

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

4.14.5.7 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.

4.14.5.8 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.

4.14.5.9 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.

4.14.5.10 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.

4.14.5.11 Match Code Enabled

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

4.14.5.12 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.

4.14.5.13 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.

4.14.5.14 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.

4.14.5.15 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.

4.14.5.16 Optimizing

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

4.14.5.17 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.

4.14.5.18 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.

4.14.5.19 AutoImage Photometry Complete

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

4.14.5.20 Output 1 Status

Current status of the physical output 1 signal

4.14.5.21 Output 2 Status

Current status of the physical output 2 signal

4.14.5.22 Output 3 Status

Current status of the physical output 3 signal

4.14.5.23 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.

4.14.6 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.

4.14.6.1 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

4.14.7 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.

4.14.7.1 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.

4.14.7.2 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

4.14.7.3 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

4.14.7.4 NoRead Counter

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

4.14.7.5 Trigger Counter

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

4.14.7.6 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

4.14.7.7 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

4.14.8 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.

4.14.8.1 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.

4.14.8.1.1 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

4.14.8.1.2 Capture Time

Total time it tool to capture the image

4.14.8.1.3 Total Decode Time

Total time spent decoding the symbol(s)

4.14.8.1.4 Total ReadCycle Time

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

4.14.8.1.5 Number of Decodes in Read Cycle

The total number of decoded symbols found during the read cycle

4.14.8.1.6 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.

4.14.8.1.7 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

4.14.9 Decode Cycle Report

Information on the decoded symbol

4.14.9.1 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

4.14.9.2 Decode Location Top

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

4.14.9.3 Decode Location Left

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

4.14.9.4 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.

4.14.9.5 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.

4.14.9.6 Code Type

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

4.14.9.6.1 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

4.14.9.7 Pixels Per Element

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

4.14.9.8 Decode Length

The number of characters found in the decode string

4.14.9.9 Decode Data

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

4.14.10 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	

4.15 MicroHAWK Output Assembly (Legacy)

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.

4.15.1 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

4.15.2 Output Assembly Description

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

4.15.2.1 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.

4.15.2.2 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.

4.15.2.2.1 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

4.15.2.2.2 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

4.15.2.2.3 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.

4.15.2.2.4 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.

4.15.2.2.5 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.

4.15.2.2.6 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.

4.15.2.3 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.

4.15.3 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

4.16 MicroHAWK Output Assembly

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.

4.16.1 Table 3.16.1 Output Assembly Table

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

Total Size: 4 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 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.

4.16.2.1.1 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

4.16.2.1.2 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.

4.16.2.1.3 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

4.16.2.1.4 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.

4.16.2.1.5 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).

4.16.2.1.6 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.

4.16.2.1.7 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.

4.16.2.1.8 Clear Noread Counter

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

4.16.2.1.9 Clear Trigger Counter

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

4.16.2.1.10 Clear Match Code Counter

Resets the Match Code/Good Read Counter to 000000000.

4.16.2.1.11 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.

4.16.2.1.12 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.

4.16.2.1.13 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.

4.16.2.1.14 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.

4.16.2.2 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

4.17 MicroHAWK Serial Command Assembly

Microscan has created an assembly to send and receive K Commands to the MicroHAWK. To be able to receive K Command request the units 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).

4.17.1 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

4.17.2 Command Length

Total number of characters in the Command string

4.17.3 Command String

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

4.17.4 Assembly Member Description

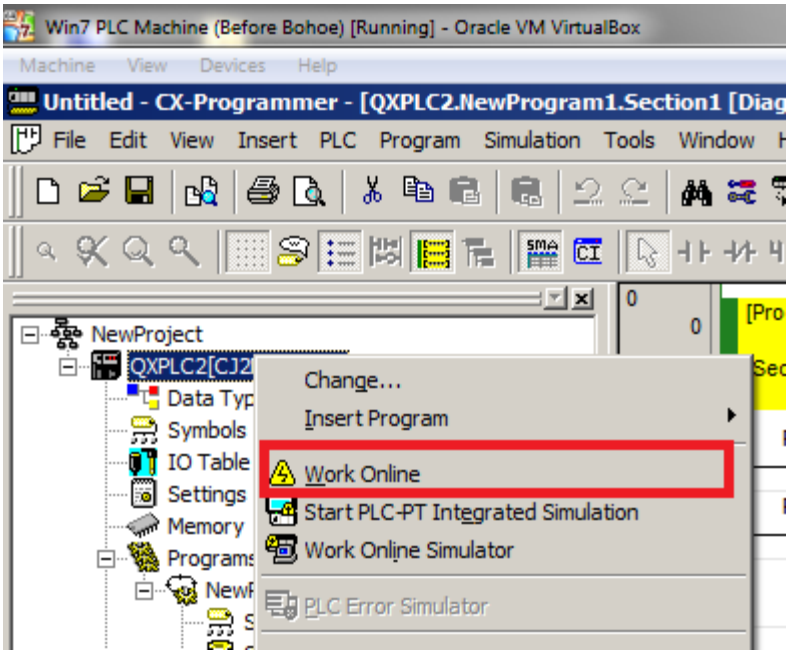
The following table is where members for the Serial Command Assembly are located.

	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**

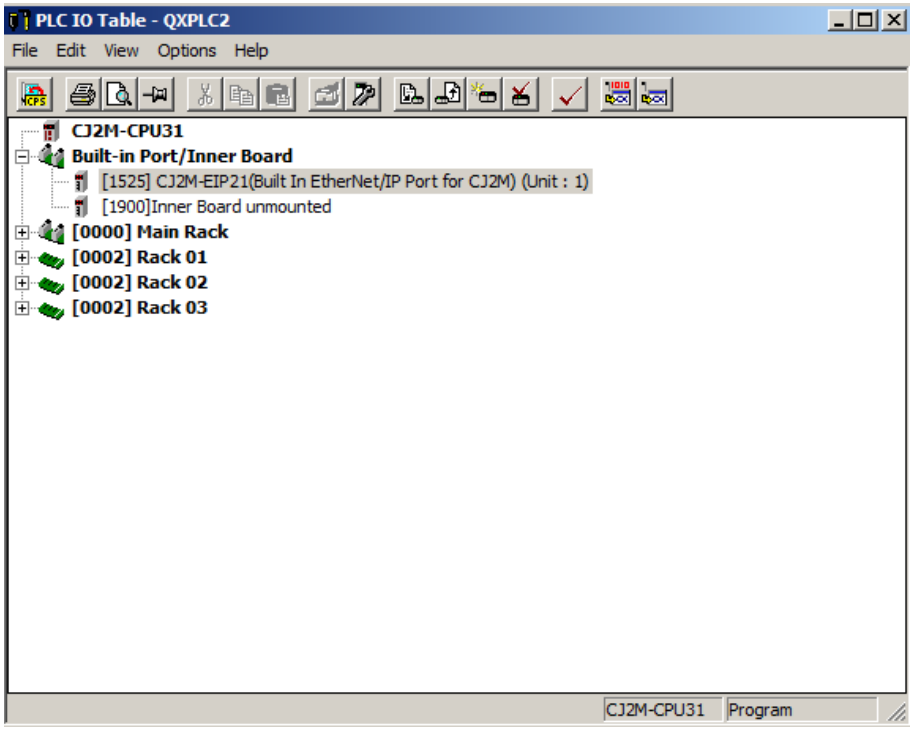
5 Adding the MicroHAWK to the CJ2 Ethernet/IP network

This section will cover installing the unit to an OMRON CJ2 controller network. These screen shots are from CX-Programmer and Network Configurator

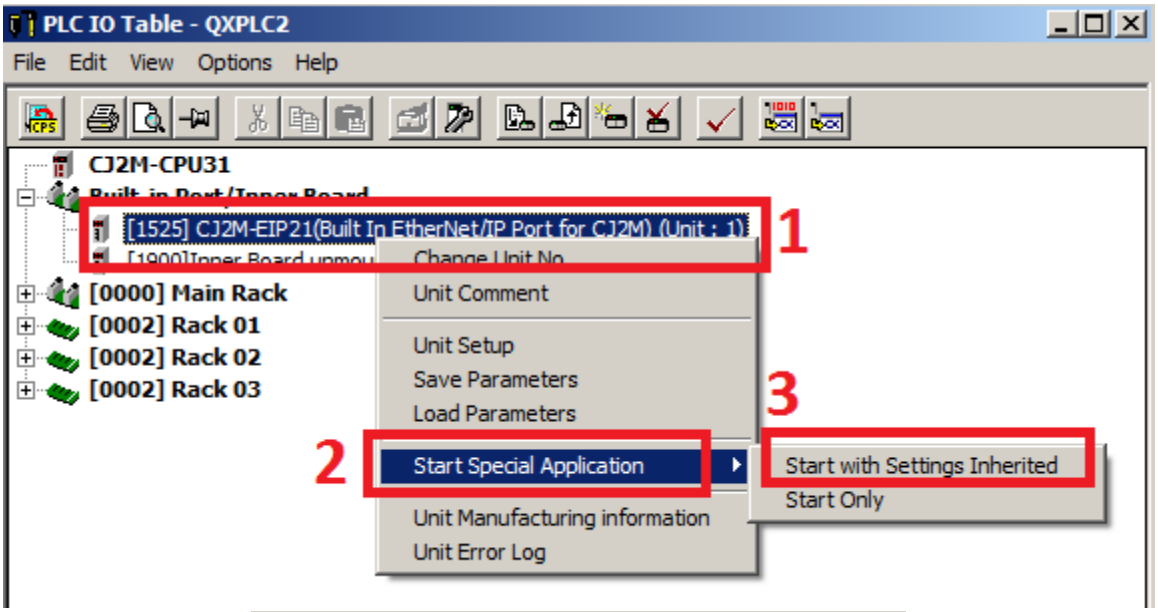
1) Go online with the controller. The example below is done be right clicking the Controller and selecting **Work Online**

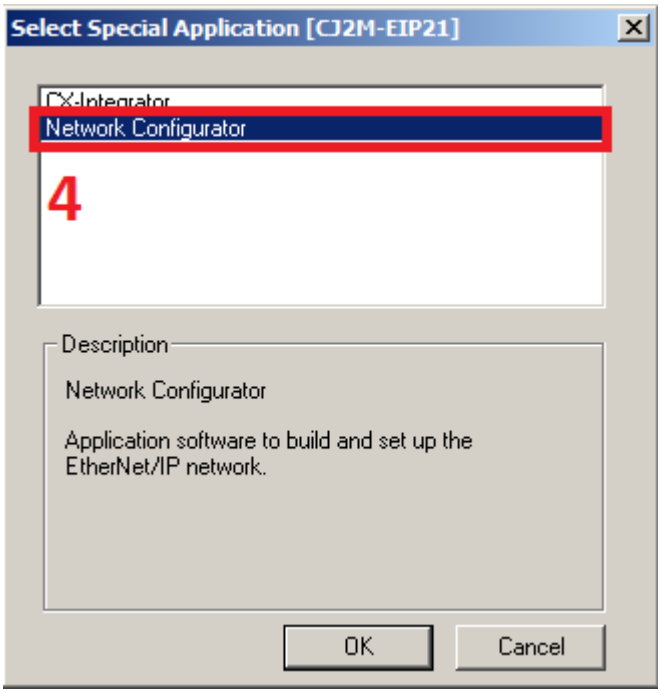


2) Double Click the IO Table and Unit Setup to bring up the IO Table

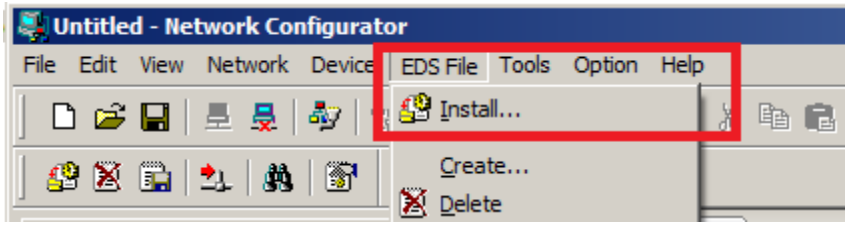


3) We will need to get the Network Configurator Application running. One way to bring it up is the following. Right Click the Ethernet/IP Port (1), select Start Special Application (2) and Start with Settings Inherited (3). Click on Network Configurator when the selection screen appears (4).

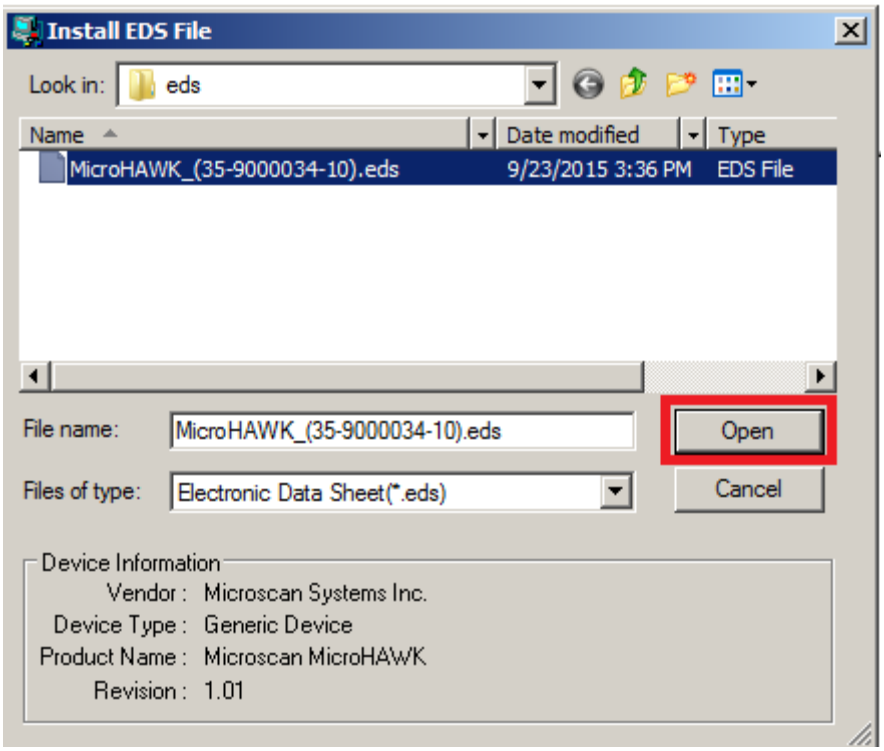




4) Microscan offers an Electronic Data Sheet (EDS) file to help with the installation of the unit to the controller. To install simply go to EDS File→Install…

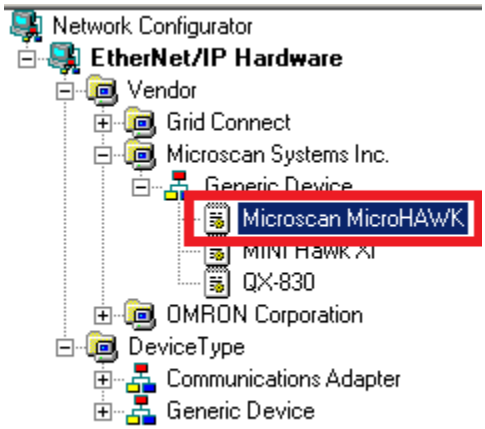


5) Navigate to the MicroHAWK EDS file ***MicroHAWK_(35-9000034-10).eds*** and click **Open**

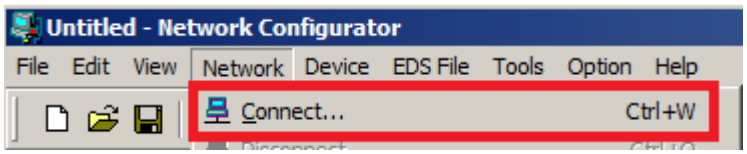


6) When prompted to install Icon if you wish to install another icon other than the default MicroHAWK icon than click YES, otherwise click NO.

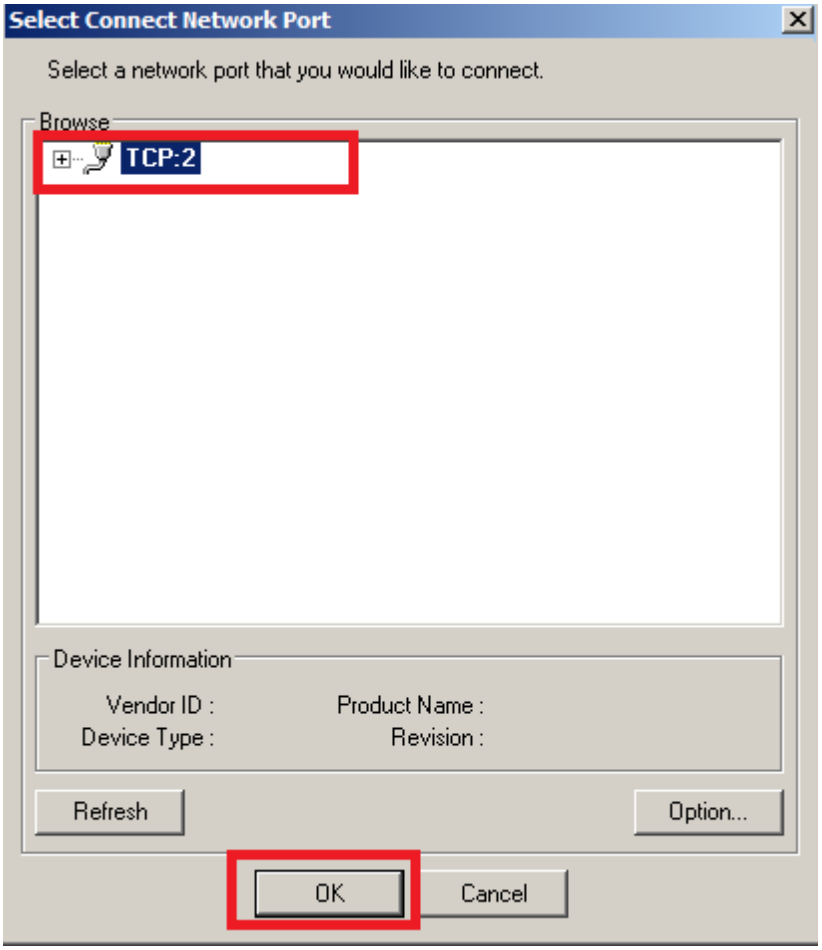
7) In the Hardware Configurator find and select the Microscan Systems Inc. Select the Folder and Generic Devices to check that the MicroHAWK configuration has been added.



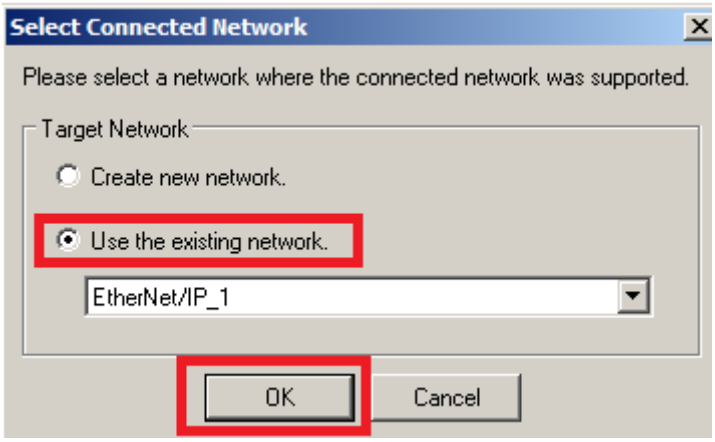
8) Connect to the network when the Network Configuration Application starts up.



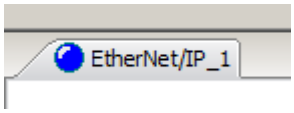
9) Select the TCP Port for connections and click OK



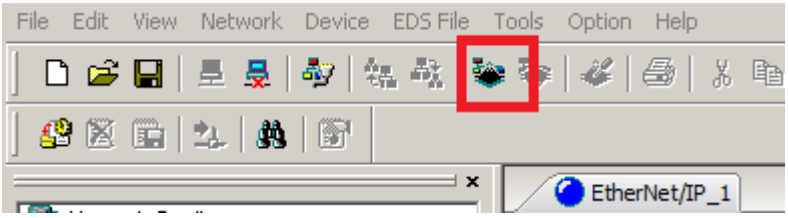
10) Choose the selection that suites your needs. In this example we'll select **Use the existing network**. Click OK.



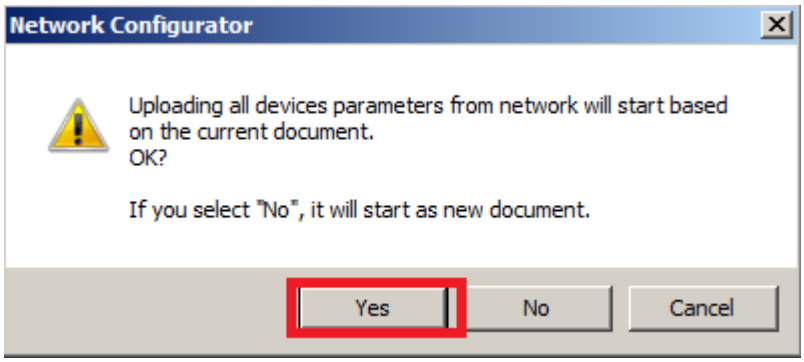
11) The Ethernet/IP lamp will turn on indicating a connection to the network



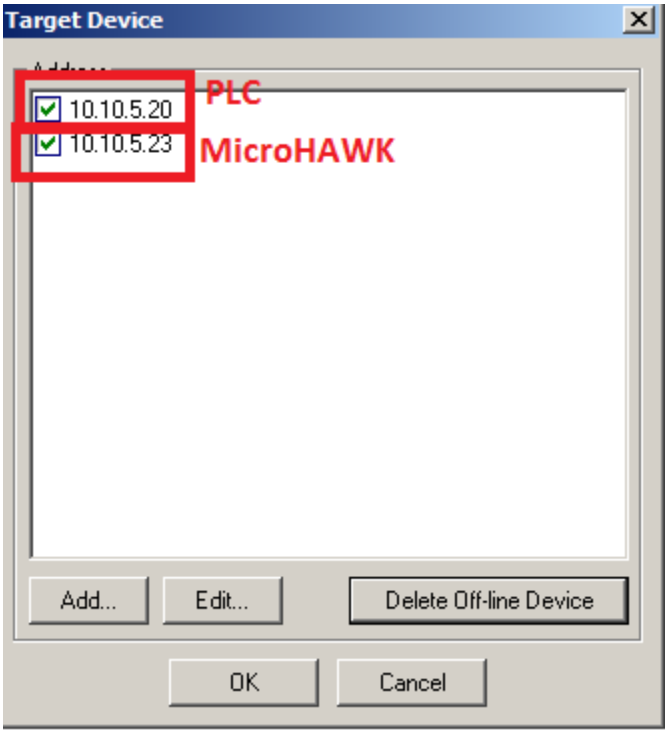
12) Click on the Upload button to upload the current devices on the network.



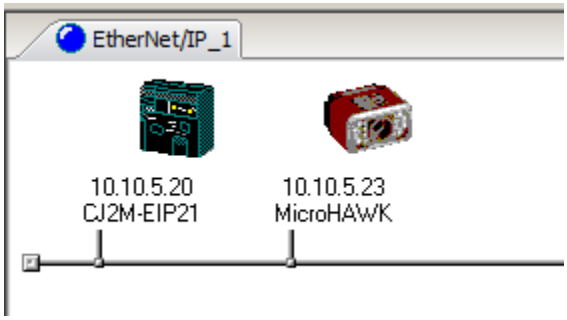
13) When prompted click on **Yes** to Upload all device parameters.



14) Make sure you select the PLC IP Address and the MicroHAWK IP Address in the selection screen. If using an existing program, simply add the MicroHAWK IP address to the network.

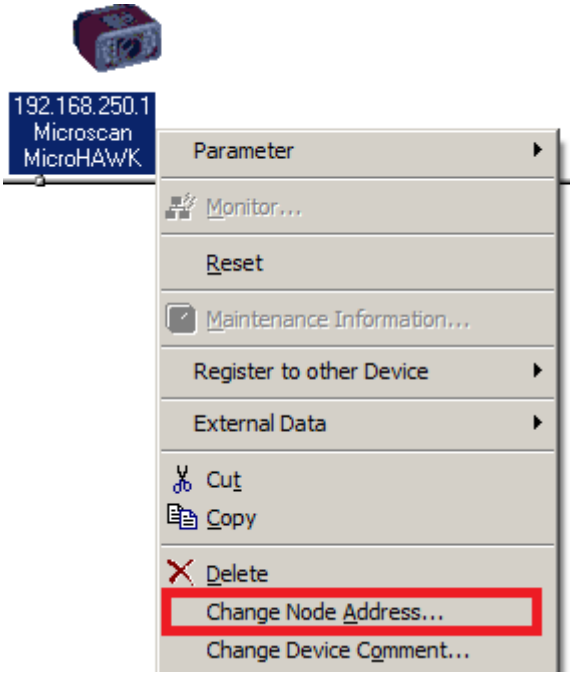


15) The system will automatically detect and load the Ethernet Models detected. The MicroHAWK should be displayed in the network node

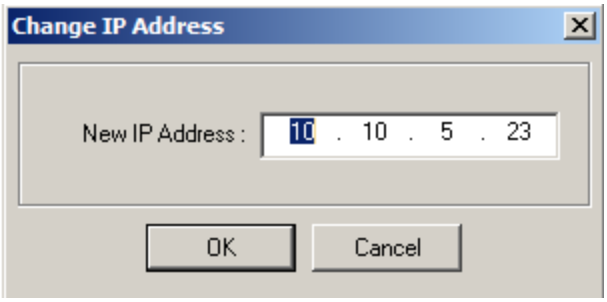


a. NOTE: To manually add the module simply drag the Microscan MicroHAWK file from the Network Configurator tree and drop it on the Network node. Complete step 16 in this section to change the IP Address.

16) If the IP Address needs to be changed then right click the MicroHAWK on the Network node and select **Change Node Address…**

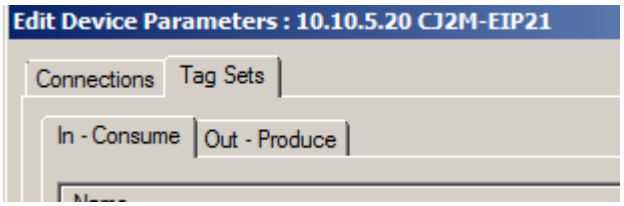


17) Change the IP Address to the MicroHAWK IP address (NOTE: Default is 192.168.188.2) In this example the unit IP address is 10.10.5.23

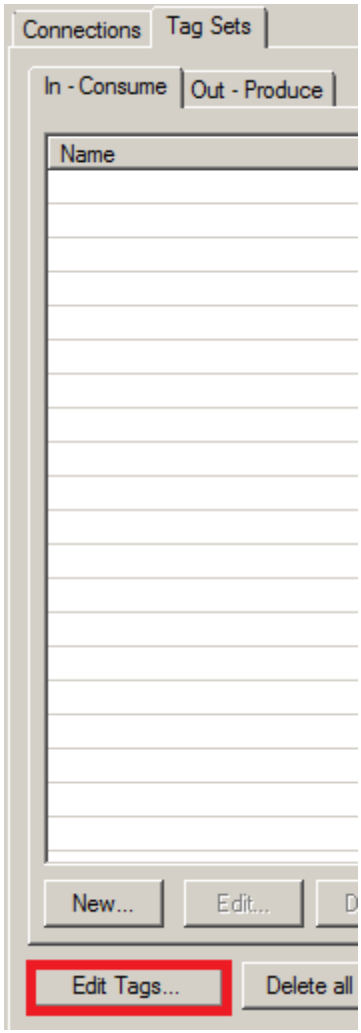


18) Configure the tag setting for the sending/receiving area by either double clicking the PLC or by right clicking on the PLC icon (1) and selecting Parameter(2)→Edit…(3)

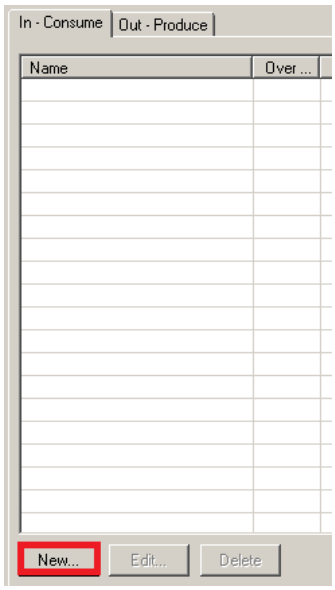
19) In Device Parameter select the Tag Set's Tab.



20) In the **In - Consume** tab click on the tag name and Click **Edit Tags…**



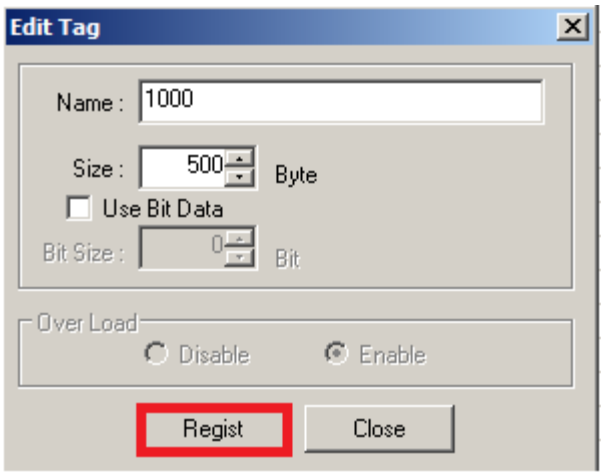
21) Click **New…**



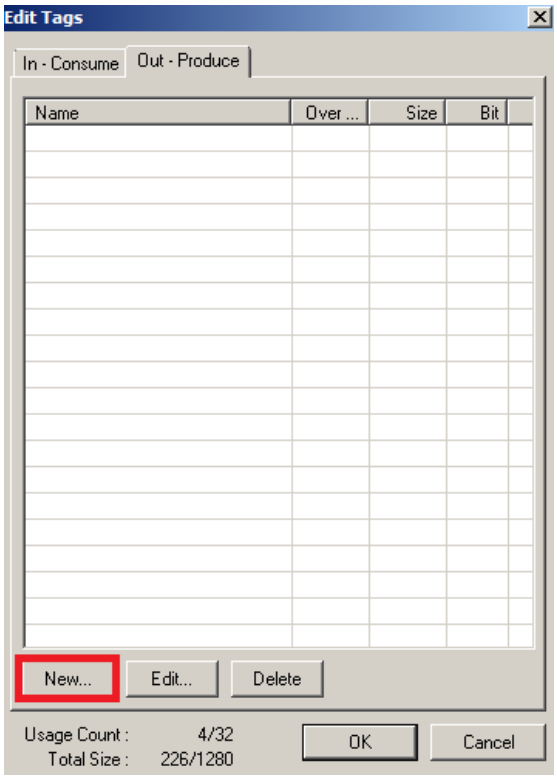
22) Change the input assembly size accordingly and click Regist. (In this example Input 1 Decode is used)

a. Reference the table for the correct input size

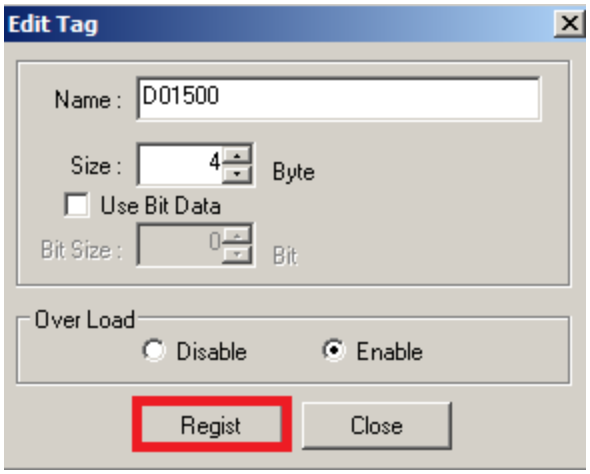
Assembly Name	Size (In Bytes)
Input Small	84
Input Big	176
Input MXL/SLC	248
Input 1 Decode	500
Input 4 Decode	500
Input N Decode	500



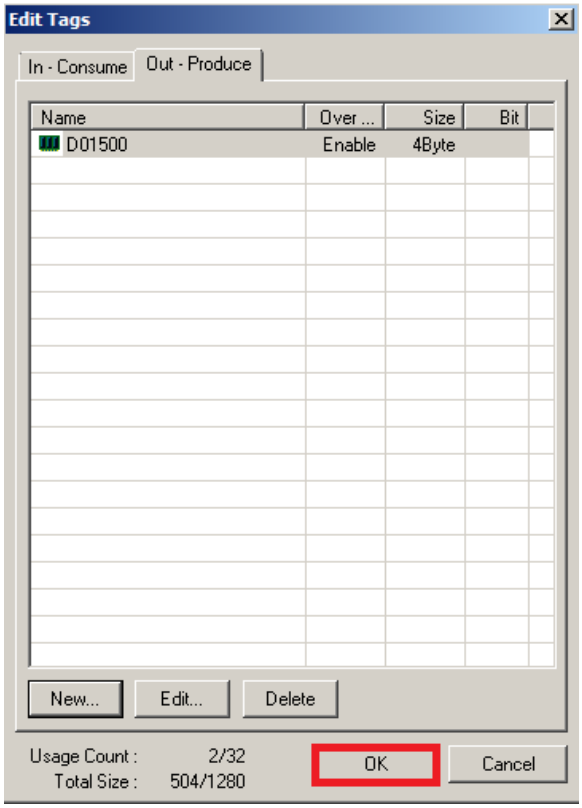
23) In the Out – Product tab click on the tag name and click **New…**



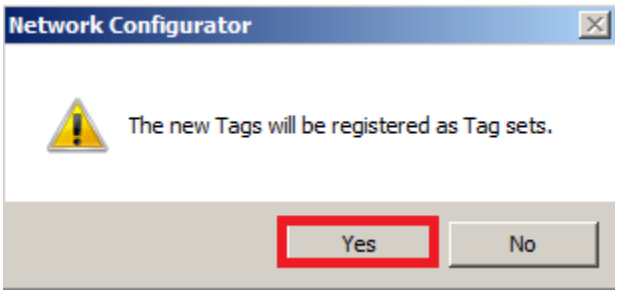
24) Change the size to either 12 Bytes if using the Output Assembly (Legacy) or to 4 Bytes for Output Assembly. Click **Regist** when done (In this example the new Output Assembly was selected).



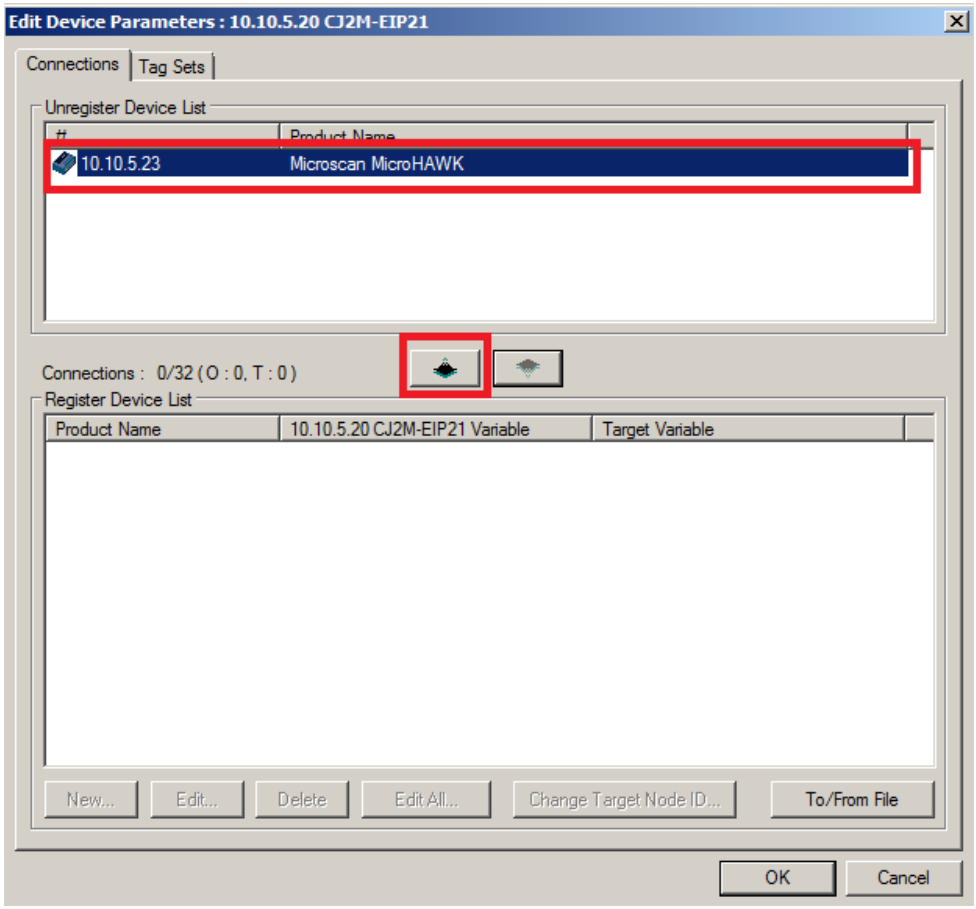
25) Click OK to close the Edit Tags Window



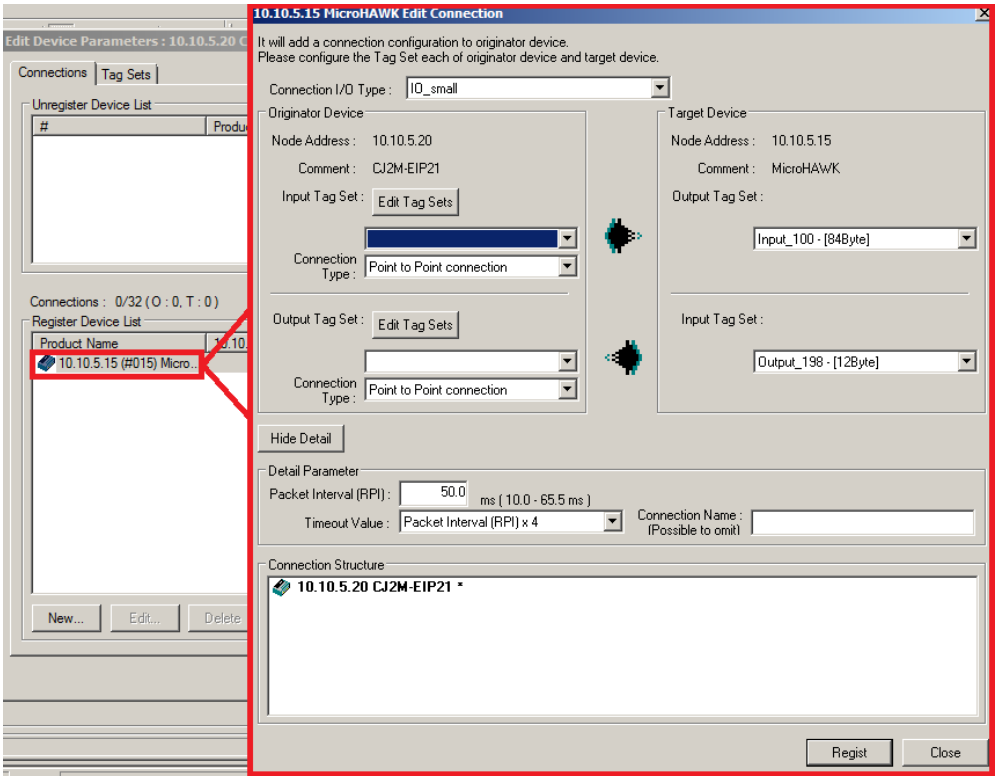
26) Click Yes to register the new Tags



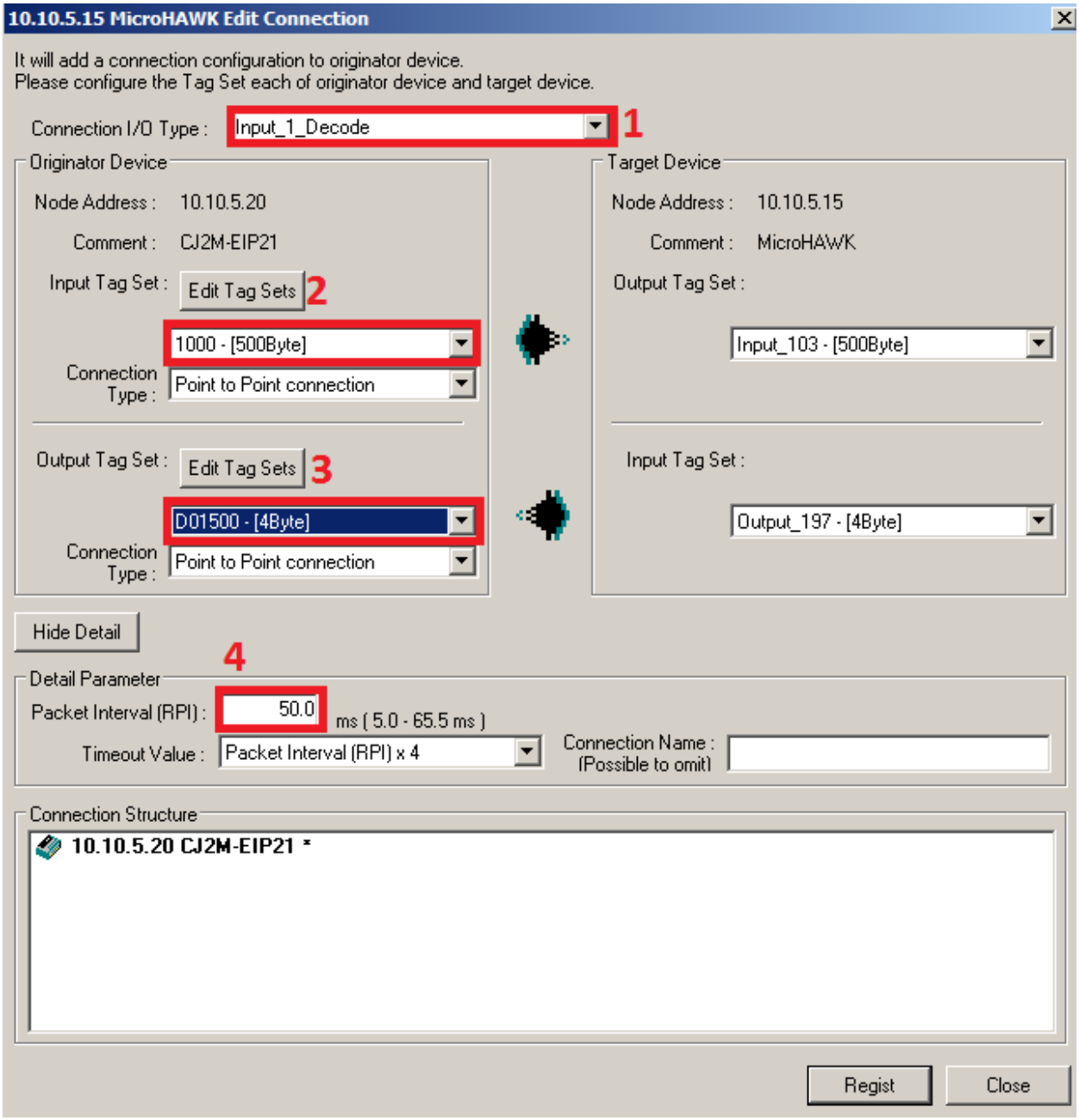
27) Click on the Connections Tab and click the Download button in the middle of the Edit Device Parameters Window



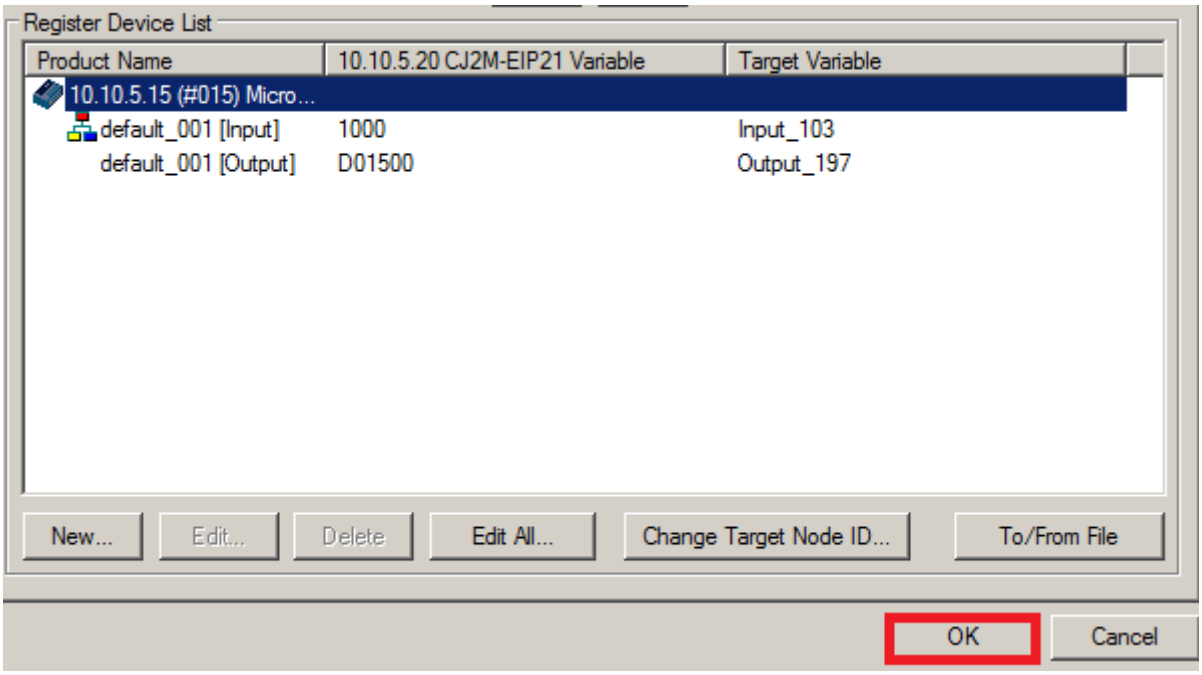
28) In the Register Device List Panel, double click on the MicroHAWK Device to bring up the Configuration



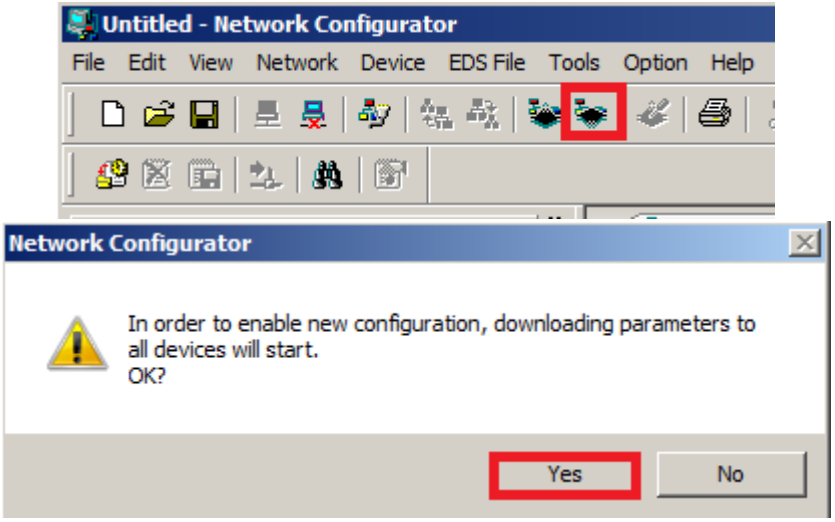
29) Selection the I/O Type (Big input or Small Input) (1) and select the correct tags (3 and 4) mapped out in steps 22 and 24. This will also be where you set your RPI(4) for the PLC. In this example we use 20ms.



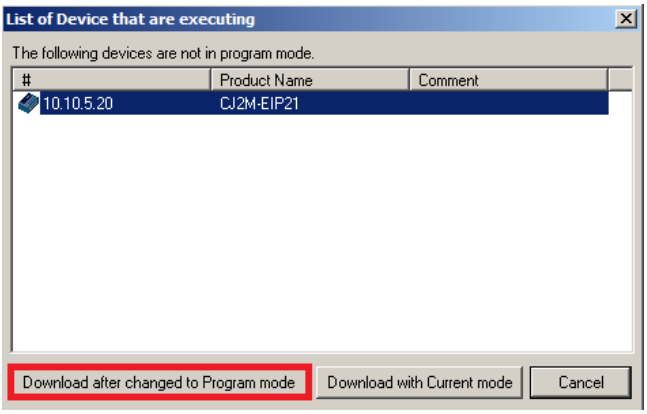
30) When complete Click Regist than close. The Device will be mapped out.



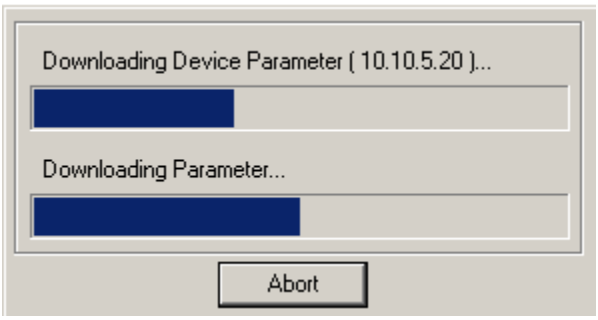
31) Download to Device by selecting the Download button. Click Yes when prompted to download.



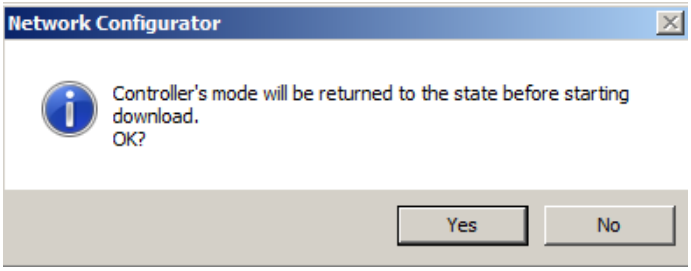
32) Highlight the PLC and select **Download after changed to Program mode**



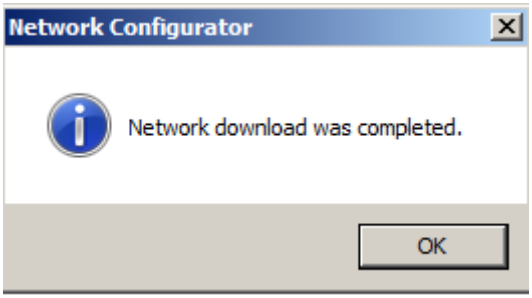
33) The Controller will download the new devices to the network



34) Click Yes when prompted to return the controller to the original state



35) Click OK when download is complete

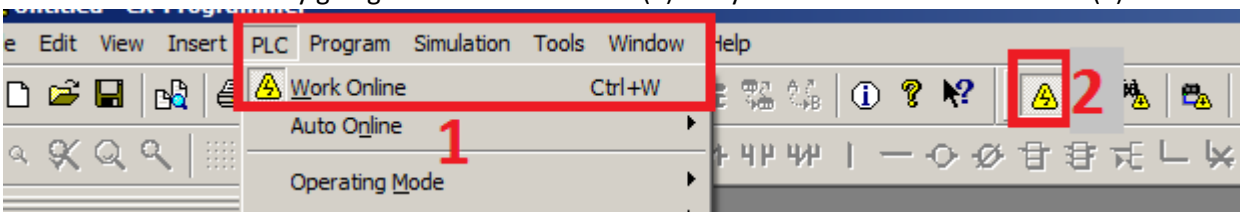


36) Close the Network Configurator and the PLC IO Table. This completes adding the MicroHAWK to the ORMON CRJ2 controller

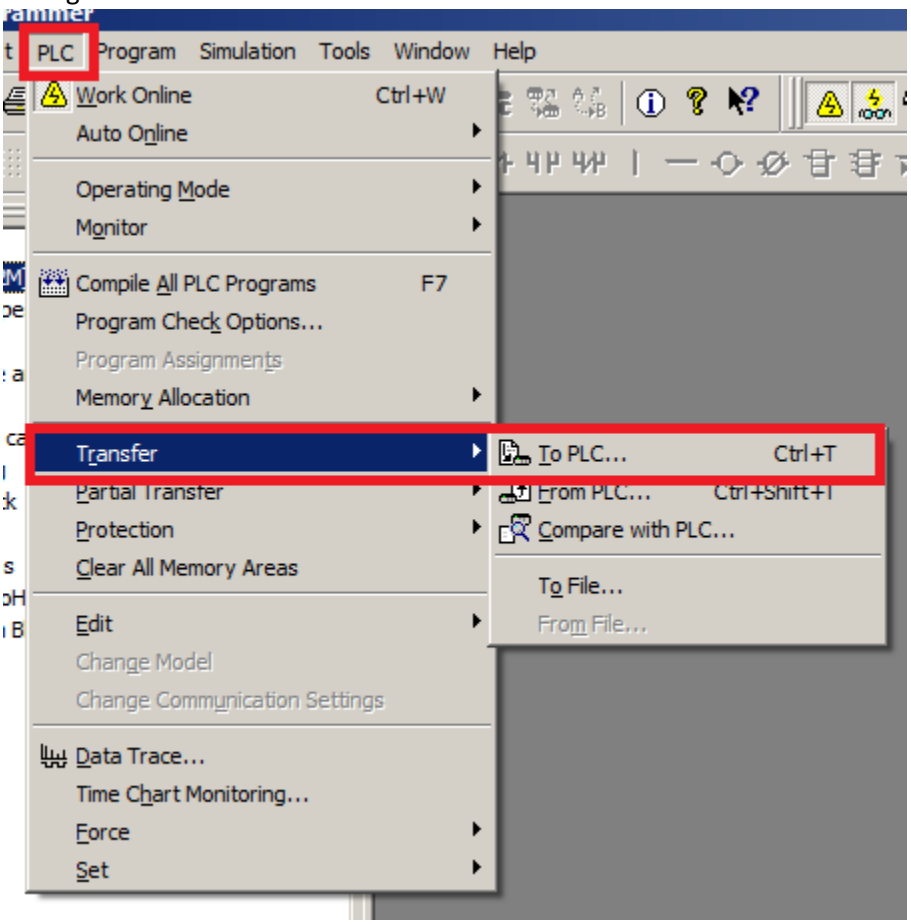
6 Downloading to PLC

This section will describe how to download to the Omron CJ2 PLC.

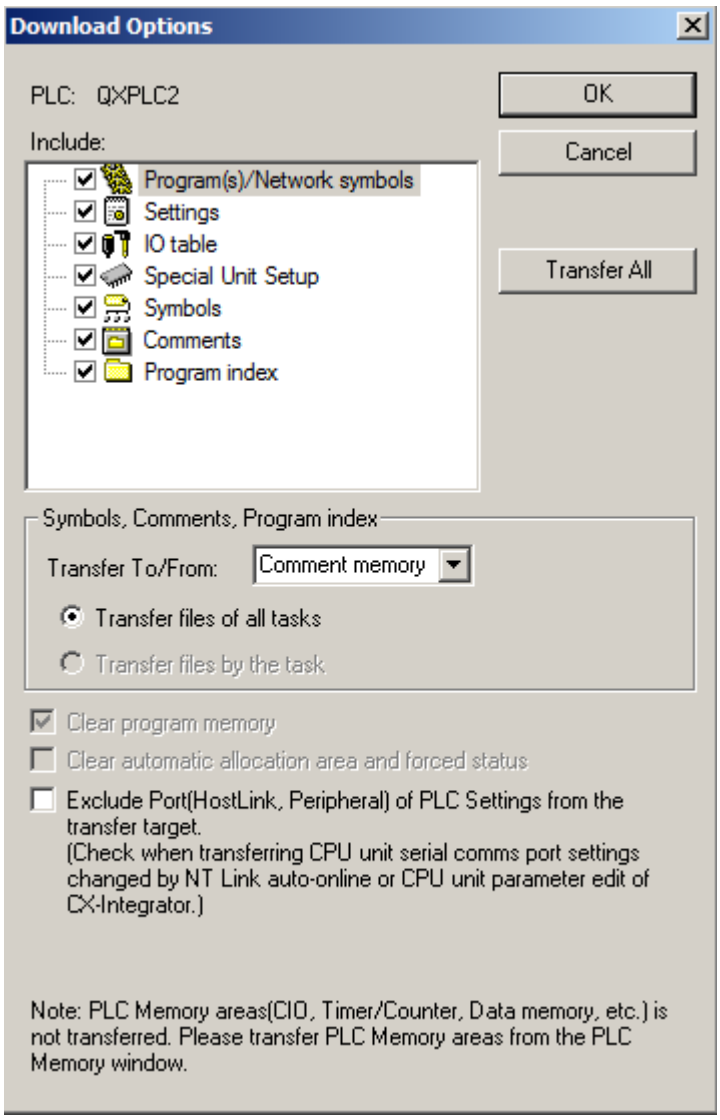
1) Go online with the PLC by going to PLC→Work Online(1) or by click the Work Online button(2)



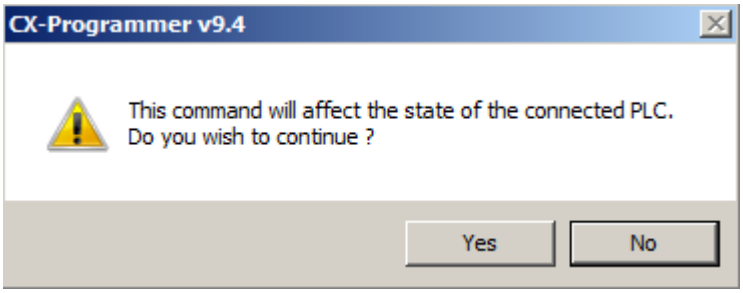
2) Next go to PLC→Transfer→To PLC…



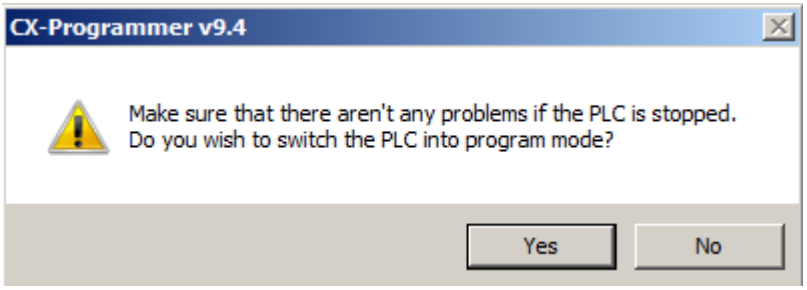
3) In this example we are including everything to download. This may not be needed for you. Click OK to proceed.



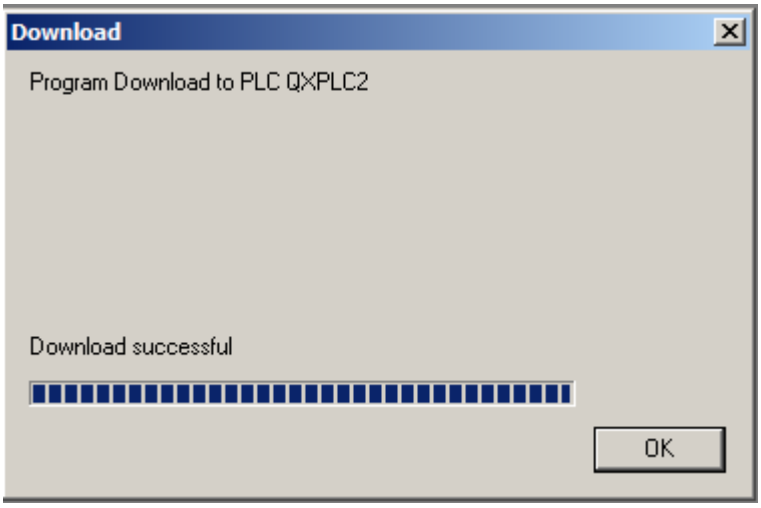
4) Click YES to continue



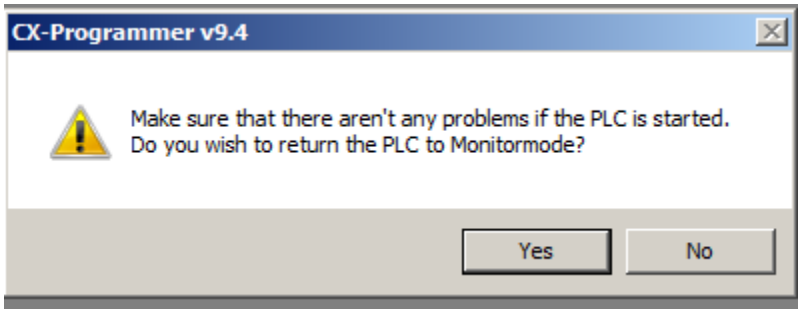
5) Select YES to switch the PLC to program mode.



6) When the download is complete, Click OK



7) Select YES to bring the PLC to Monitoring Mode



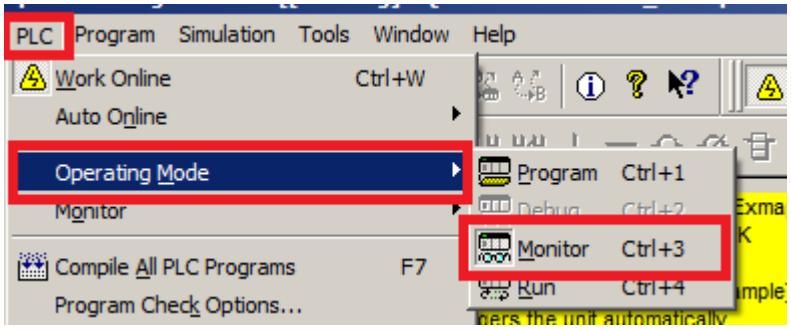
8) This completes downloading to the PLC. Please see section 5 and 6 to test the MicroHAWK on the Omron CJ2

7 Testing the MicroHAWK via the PLC Memory Table

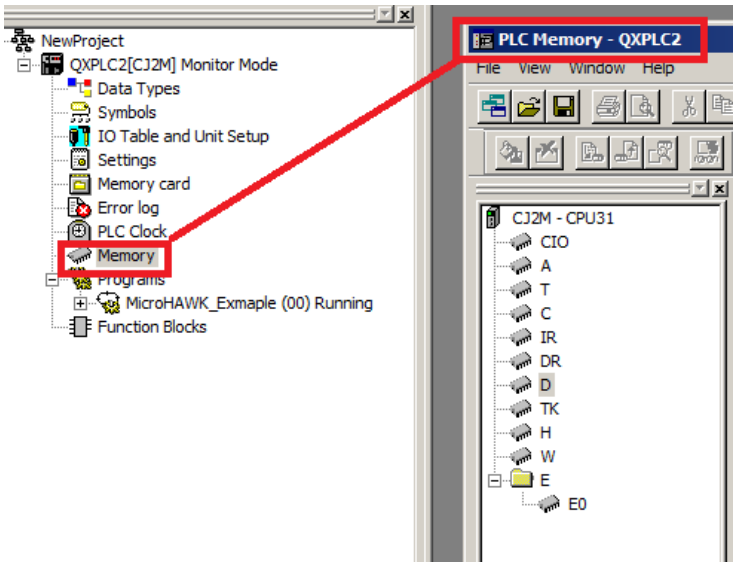
This section will describe how to test the newly added MicroHAWK on the OMRON CRJ2 controller via the PLC Memory Node. In this example our input are mapped to tag D01000 and the outputs are mapped to D01100

NOTE:You must complete Section 4 Download to PLC for this to work.

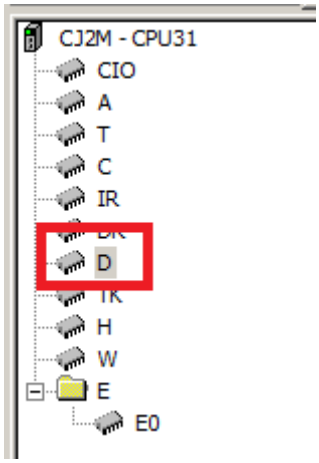
1) Set the PLC to Monitor mode by going to PLC→Operating Mode→Monitor



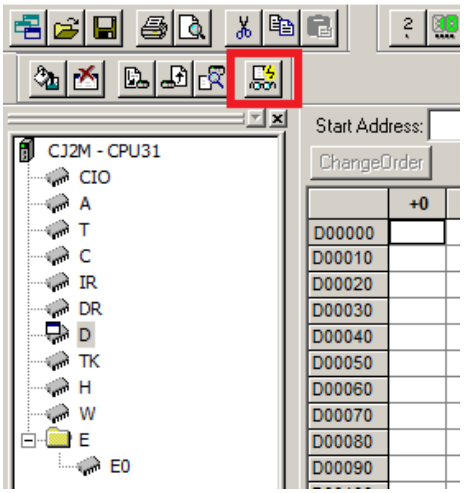
2) Double Click the Memory Node (1) in the Project Tree to bring up the memory watch.



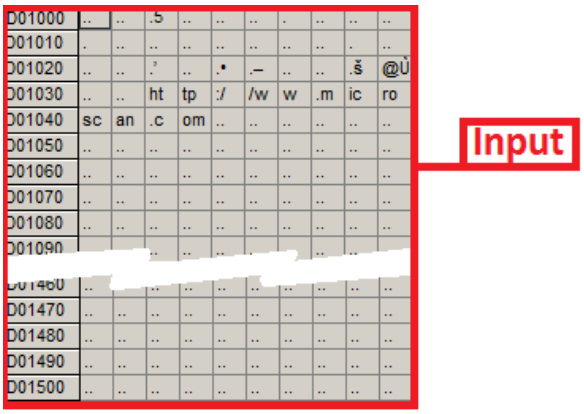
3) Double click the "D"

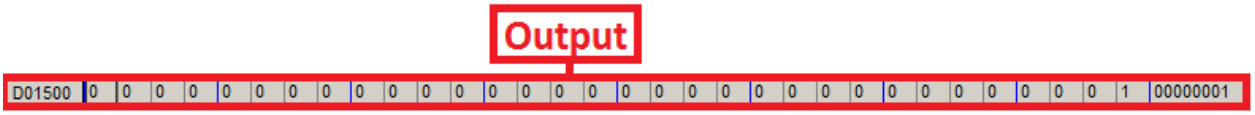


4) Click the Watch button to read/write to the PLC the memory



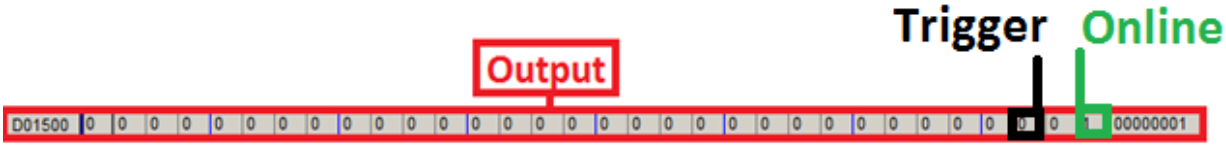
5) Mapping is as follows (This example we uses Input 1 Decode Instance 0x67 (500 bytes) and Output instance 0xC5 (4 bytes))





Decoded Data is here

6) Using the mapping above and referencing the Output Assembly in Section 7. The trigger is D01500.2 and Online is at D01500.0



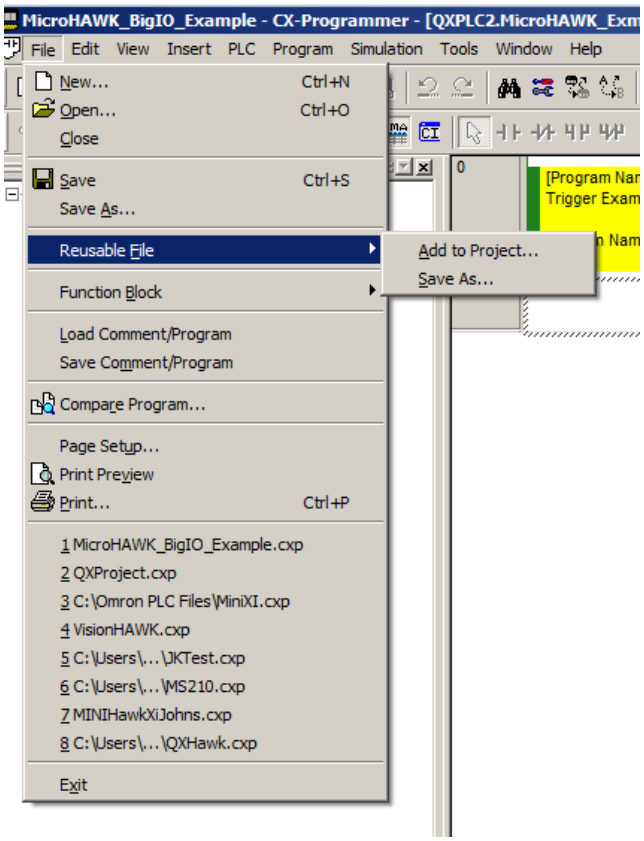
7) Toggling D01500.2 bit from 1 to 0 will trigger the unit when the unit is Online

8 Reusable Files

This section describes how to add the reusable files created by Microscan Inc. The files supplied only use one of the Input Assemblies outlined in sections 4.9 to 4.16. When adding these files to your project please verify that the tags created in section 3 must be used for these examples to work.

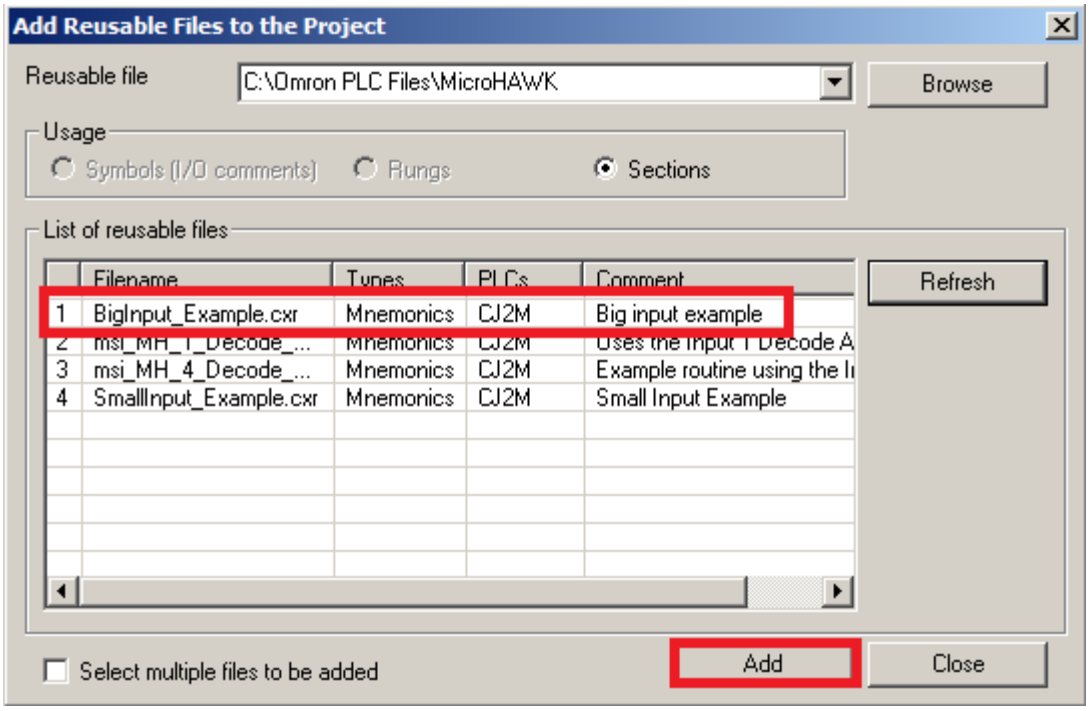
<u>These files are created for demonstration purposes only and should not be used in a production environment!</u>

1) Go to File→Reusable File→Add to Program…

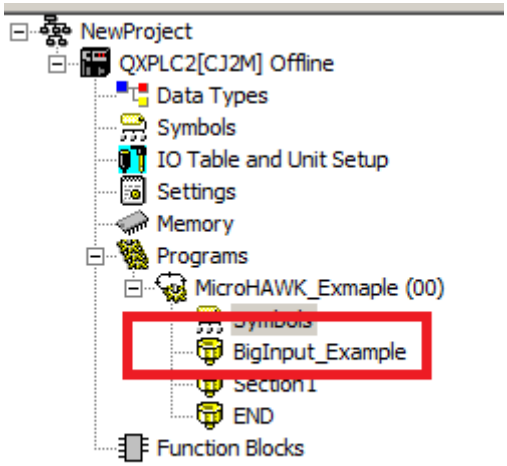


2) Browse to the location of the *.cxr files referenced in the beginning of this section. These files are located in the MicroHawk SD card under Industrial Protocols/EthernetIP/Omron CJ2/Example Routine. Files are also downloadable at: http://www.microscan.com/en-us/ServiceAndSupport/DownloadCenter.aspx

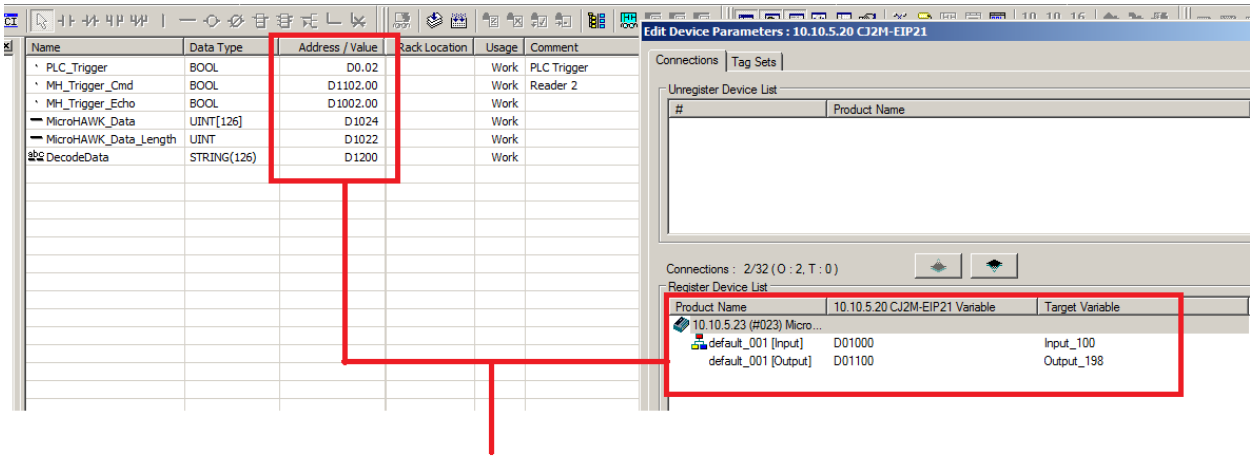
3) Once you have clicked OK they will appear in the List of Reusable Files. Click on the file you wish to use and click Add.



4) The added Reusable file will appear in the Programs dropdown tree

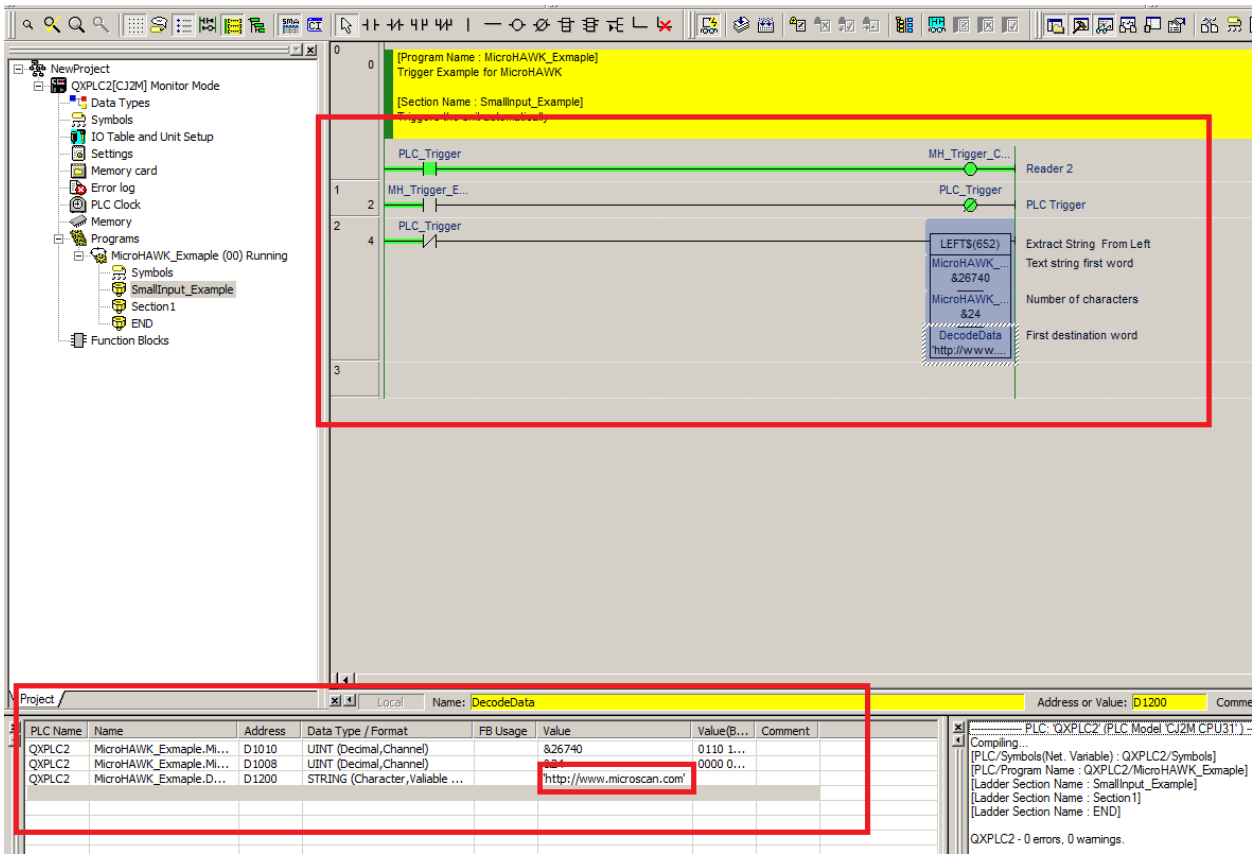


5) Any error's that appear will need to be corrected. Please verify that the tags located in **Programs→Symbols** that were imported match those you created in Section 3 in this guide.



These values may not match the ones you created in Section 3. Please update these values to match yours. You can check this by going to Network Connections and referencing the Tag Variables that were created

6) Follow Section 4 to download the example to the Omron CRJ2. The MicroHAWK will begin triggering and data will be displayed in Rung 3 of the example.

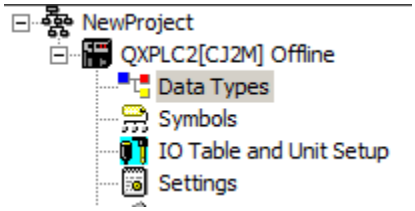


8.1 Input N Decode Example

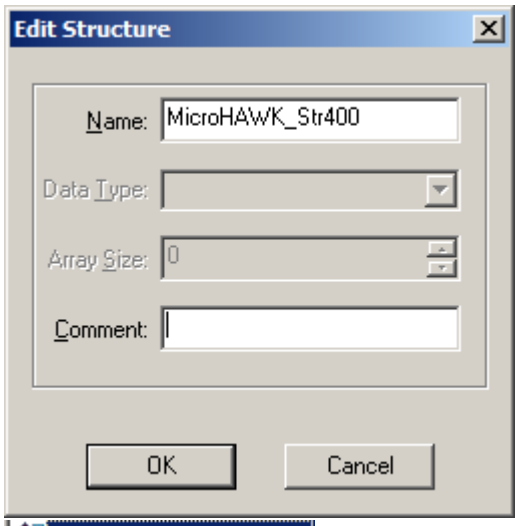
Due to the complexity of using the Input N Decode Assembly Microscan has included three Function Blocks to help the programmer locate the decode data and place the data into a user defined structure that is able to hold up to 10 decode symbols. The structure can be resized to any value the programmer deems necessary and can be resized to any value. This section will go over how to create the routine to get the programmer started on using the Input N Decode Assembly in the OMRON CX-Programmer project.

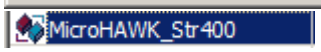
8.1.1 Creating Custom Data Types

This section will cover on how to create the custom data type to place the decode data in the Input N Assembly. The programmer should have knowledge of the CX-Programmer.

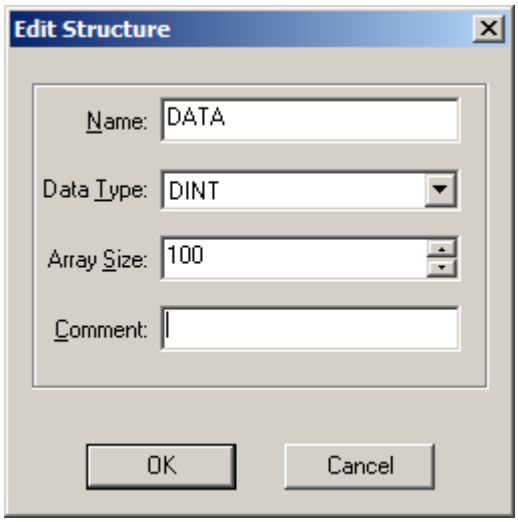
1. Double Click Data Types in the CX-Programmer Project.
   a. 

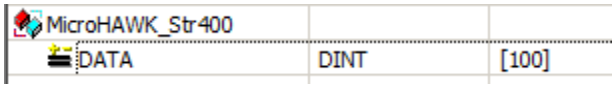
2. Right Click in the grid and select **Insect→Stuct…**
   a. 

3. Type in **MicroHAWK_Str400** in the edit structure window and click **OK.** This will add the structure MicroHAWK_Str400
   a. 

   b. 

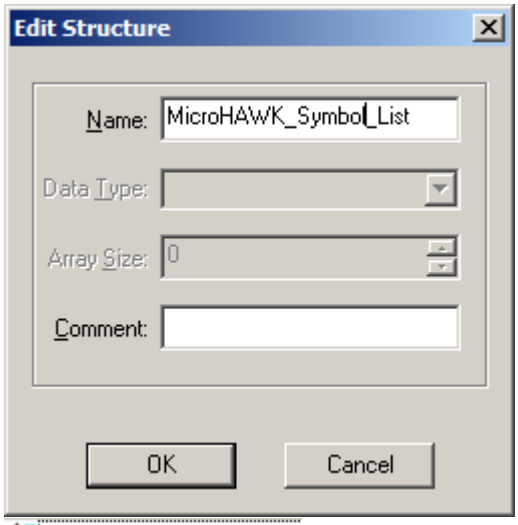
4. Right Click the newly added structure **MicroHAWK_Str400** and select **Add→Member…**
   a. 

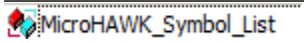
5. Type the following into the Edit Structure Window and click **OK**
   a. 

6. The member DATA that is a DINT array of 100 is now added to the structure MicroHAWK_Str400
   a. 

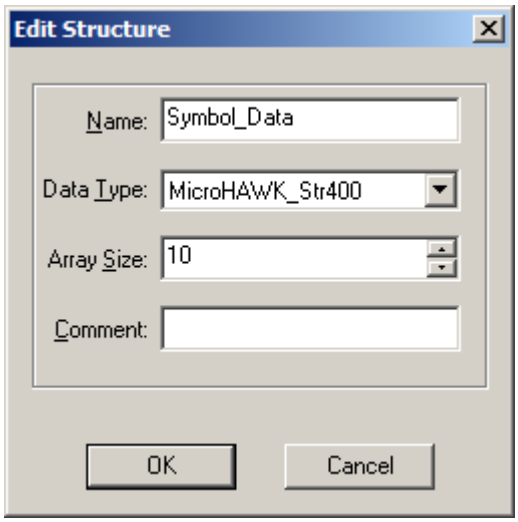
USER NOTE: This allows the system to hold a 400 bytes of data into this data type

7. Now we need to add a structure to hold a list of MicroHAWK_Str400 members. This allows the programmer to place the symbol data into an array the programmer can reference in the project. Right Click in the grid and select **Insect→Stuct…**
8. 

9. In the Edit Structure window type in **MicroHAWK_Symbol_List** and click **OK**. This will add the structure **MicroHAWK_Symbol_List**
   a. 

   b. 

10. Right Click the newly added structure **MicroHAWK_Symbol_List** and select **Add→Member…**
    a. 

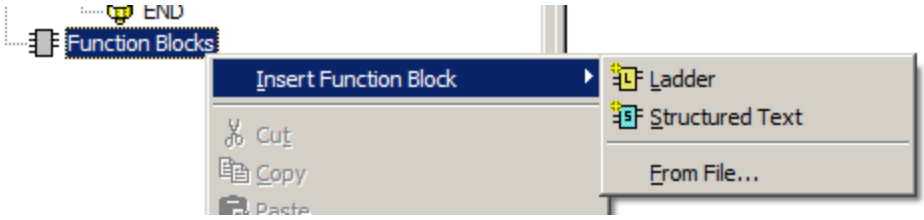
11. Type the following into the Edit Structure Window and click **OK**
    a. 

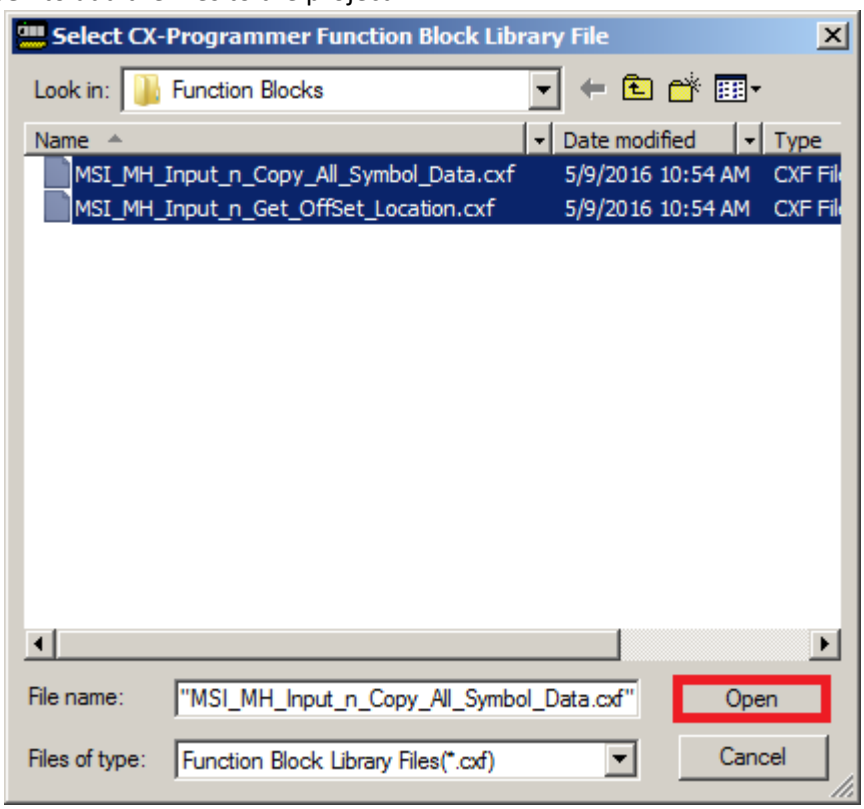
    b. **NOTE: The programmer should make the array size the maximum number of decode strings that will be decoded in one inspection.**
        i. **Example: In Weblink, the reader is set to look for 10 symbols. Therefore in this example the array size will be set for 10. This may not be the case for all application and it is the programmer's responsibility to size the array correctly.**
        ii. 

12. Now the member **Symbol_Data** has been added to the structure **MicroHAWK_Symbol_List**
    a. 

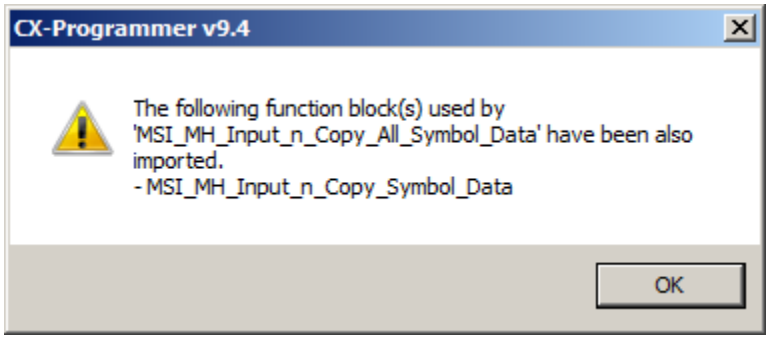
8.1.2 Adding Microscan Functions Blocks to the CX-Programmer

This section will cover adding the functions blocks to the CX-Programmer to get the Input N Decode Example to work in the CX-Programmer. These functions are only meant to aid the programmer in finding the decode symbol data and placing it into a tag type of MicroHAWK_Symbol_List created in section 8.1.1. Please reference this section on how to create the data type before proceeding in this section.

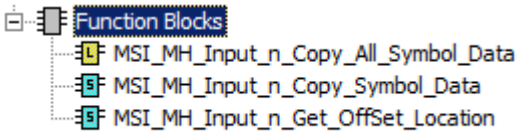
1. In the Project Tree, right click on **Function Blocks** and select Include **Function Blocks→From File…**
   a. 

2. Browse to the files located in the MicroHawk SD card under Industrial **Protocols/EthernetIP/Omron CJ2/Example Routine/Function Blocks** and highlight all the files. Click **Open** to add the files to the project.
   a. 

   b. A warning will display stating another function block is going to be added. Click OK to proceed as this function block is required.

c. 

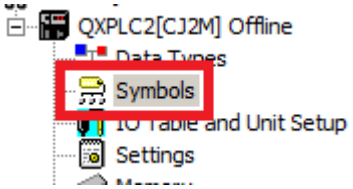
d. The Project should now contain three function blocks
   i. MSI_MH_Input_n_Copy_All_Symbol_Data
   ii. MSI_MH_Input_n_Copy_Symbol_Data
   iii. MSI_MH_Input_n_Get_Offset_Location

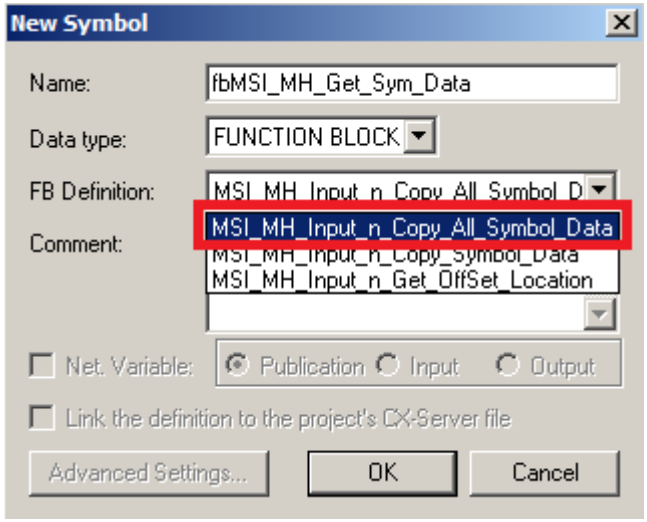


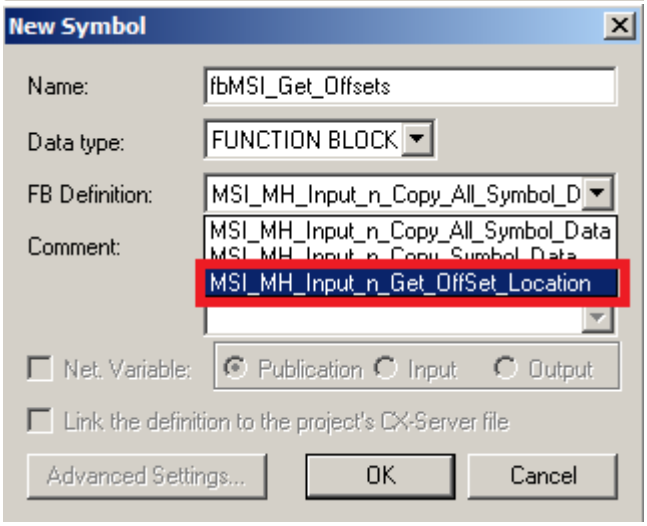
1.

8.1.3 Adding Required Tags

This section will go over adding the required tags to run the example routine Input N Decode. The programmer needs to complete sections 8.1.1 and 8.1.2 before completing this section.

1. Double click the Global Symbol Label below the Data Types node in the Project Tree
   a. 

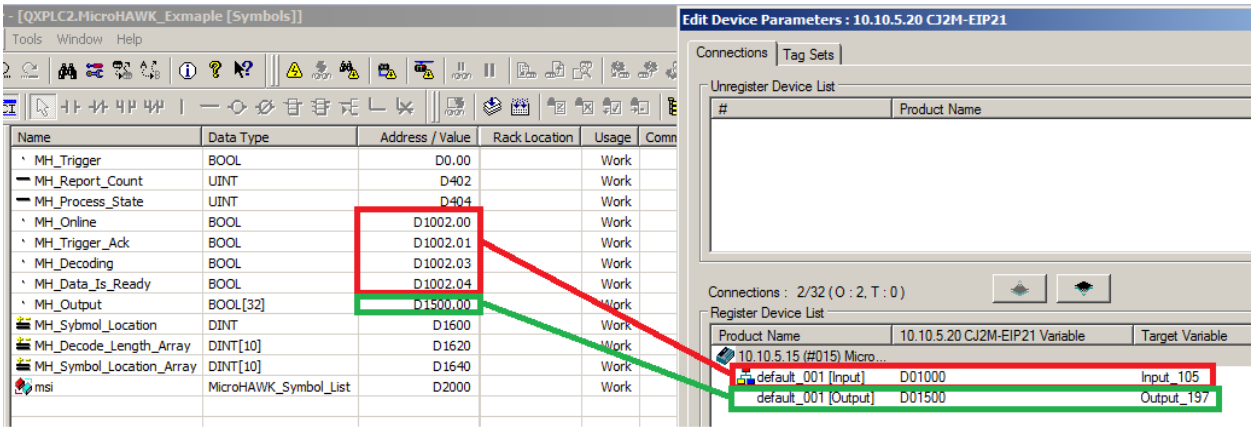
2. Add the following tags that reference the Functions blocks below
   a. 

   b. 

   c. You should have the following two tags added to the Global Project Symbol Table.

fbMSI_MH_Get_Sym_Data	FB [MSI_MH_Input_n_Copy_All_Symbol_Data]	N/A [Auto]
fb_MSI_Get_Offsets	FB [MSI_MH_Input_n_Get_OffSet_Location]	N/A [Auto]

3. In the Programs Symbol Table, add the following tags.

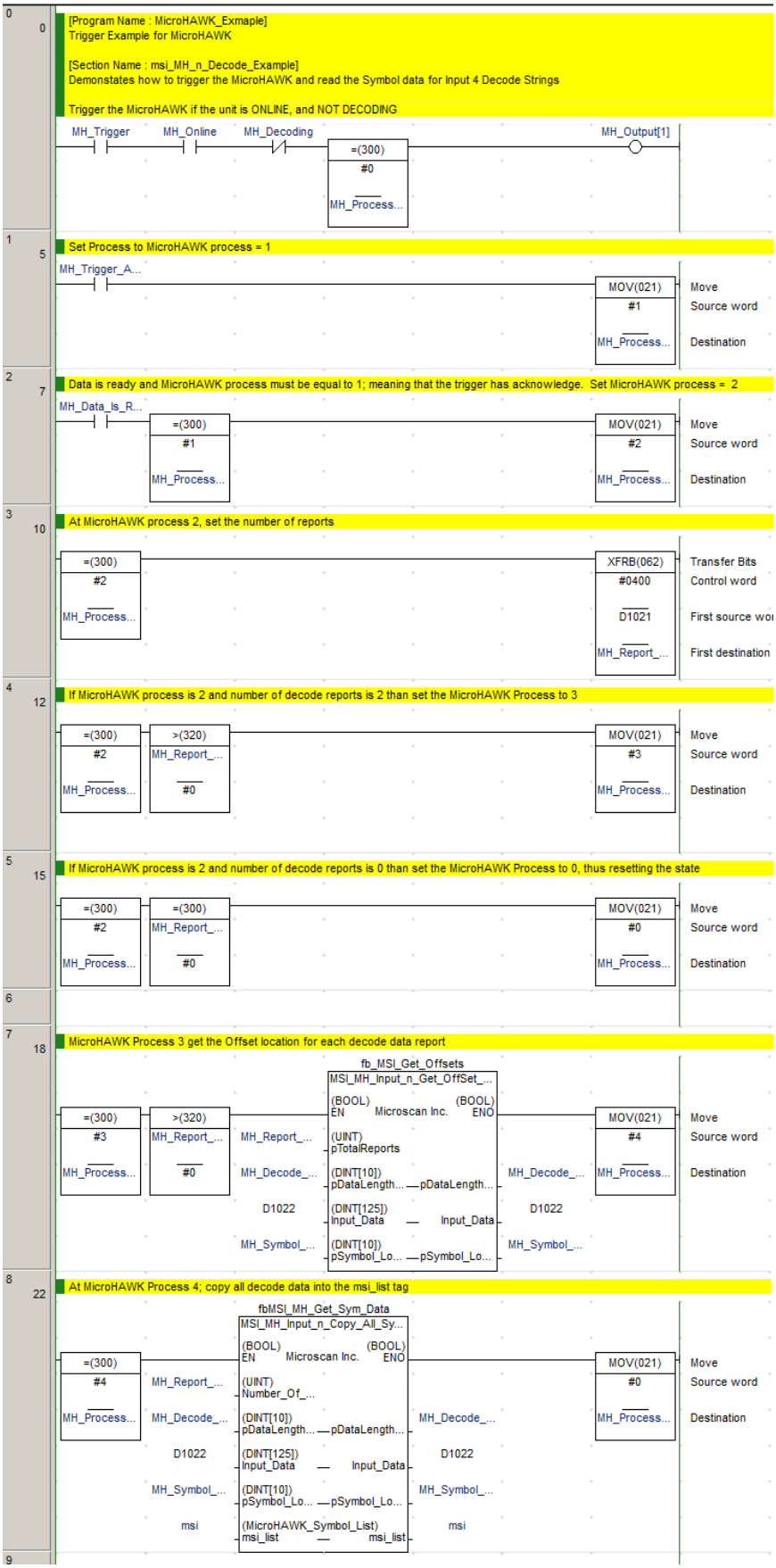


These variables may not match the ones you created in Section 5. Please update these values to match the memory location in your project. You can check this by going to Network Connections and referencing the Tag Variables that were created

8.1.4 Adding Required Rungs

This section will go over the rungs needed to get the Input N Decode assembly to work in the CX-Programmer. The programmer should complete sections 8.1.1 through 8.1.3 before proceeding in this section.

1. Create the rungs displayed below. The sections in Yellow are comments to help the programmer understand what each rung is doing.


[Program Name : MicroHAWK_Exmaple]
Trigger Example for MicroHAWK
[Section Name : msi_MH_n_Decode_Example]
Demonstates how to trigger the MicroHAWK and read the Symbol data for Input 4 Decode Strings
Trigger the MicroHAWK if the unit is ONLINE, and NOT DECODING
MH_Trigger
MH_Online
MH_Decoding
=(300)
#0
MH_Process...
MH_Output[1]
Set Process to MicroHAWK process = 1
MH_Trigger_A...
MOV(021)
#1
Move
Source word
Destination
Data is ready and MicroHAWK process must be equal to 1; meaning that the trigger has acknowledge. Set MicroHAWK process = 2
MH_Data_Is_R...
#2
At MicroHAWK process 2, set the number of reports
XFRB(062)
#0400
D1021
MH_Report_...
Transfer Bits
Control word
First destination
If MicroHAWK process is 2 and number of decode reports is 2 than set the MicroHAWK Process to 3
>(320)
#3
If MicroHAWK process is 2 and number of decode reports is 0 than set the MicroHAWK Process to 0, thus resetting the state
MicroHAWK Process 3 get the Offset location for each decode data report
fb_MSI_Get_Offsets
MSI_MH_Input_n_Get_OffSet_...
(BOOL) EN
(BOOL) ENO
Microscan Inc.
(UINT) pTotalReports
MH_Report_...
(DINT[10]) pDataLength...
MH_Decode_...
pDataLength...
(DINT[125]) Input_Data
D1022
Input_Data
(DINT[10]) pSymbol_Lo...
MH_Symbol_...
pSymbol_Lo...
At MicroHAWK Process 4; copy all decode data into the msi_list tag
fbMSI_MH_Get_Sym_Data
MSI_MH_Input_n_Copy_All_Sy...
(UINT) Number_Of_...
msi
(MicroHAWK_Symbol_List) msi_list
#4

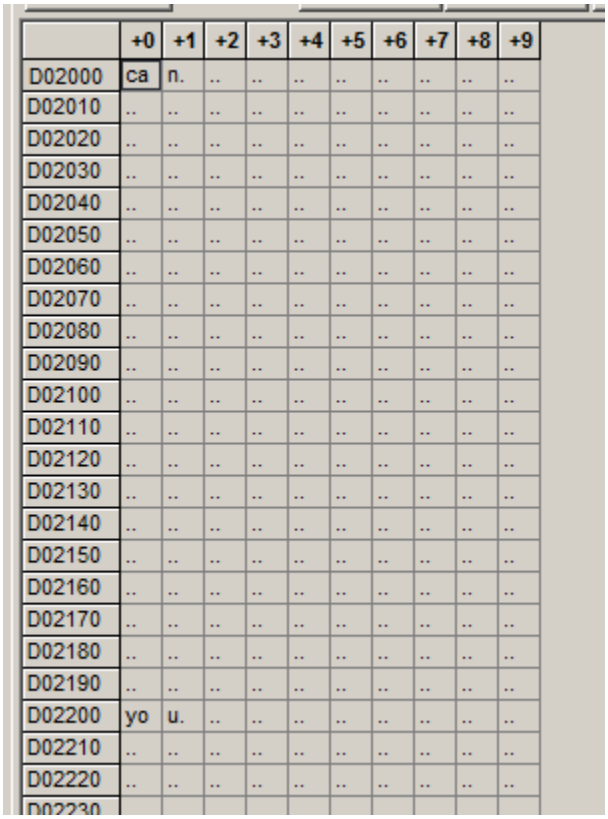
When completed, compile and download to the PLC. Reference section 8.1.5 on how to execute the routine.

8.1.5 Description of Input N Decode Example

The section will go over the example routine “Input N Decode” supplied by Microscan Inc. This example is a baseline for the programmer to reference. The programmer should be have full understand of EtherNet/IP and how to access the data in the assembly. Please reference sections 4.9 through 4.17 for all EtherNet/IP Assemblies for the unit.

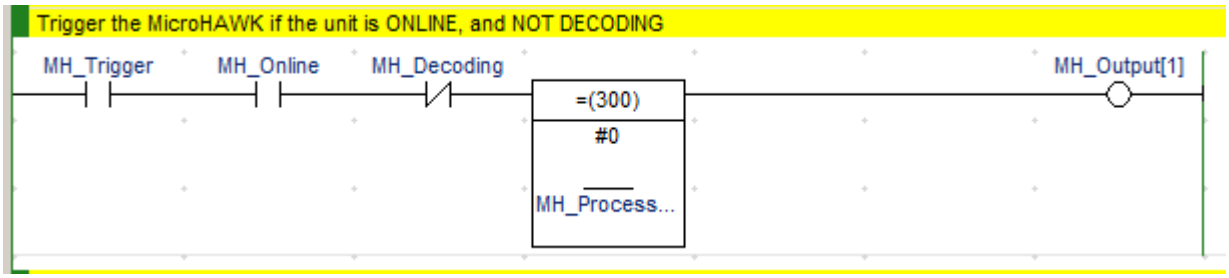
8.1.5.1 Definitions of MH_Process States

Adsfasdf

- MH_Process = 0
  - Unit can be triggered
- MH_Process = 1
  - Unit has acknowledge a trigger from the PLC
- MH_Process = 2
  - PLC is getting the total number of decode reports in the Input Assembly
- MH_Process = 3
  - PLC is getting the offset location of each decode report
- MH_Process = 4
  - PLC is transferring the decode symbol data to the tag msi. The decode symbols can be read at memory location D2000
  - 

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D02000	ca	n.	..	..	..	..	..	..	..	..
D02010	..	..	..	..	..	..	..	..	..	..
D02020	..	..	..	..	..	..	..	..	..	..
D02030	..	..	..	..	..	..	..	..	..	..
D02040	..	..	..	..	..	..	..	..	..	..
D02050	..	..	..	..	..	..	..	..	..	..
D02060	..	..	..	..	..	..	..	..	..	..
D02070	..	..	..	..	..	..	..	..	..	..
D02080	..	..	..	..	..	..	..	..	..	..
D02090	..	..	..	..	..	..	..	..	..	..
D02100	..	..	..	..	..	..	..	..	..	..
D02110	..	..	..	..	..	..	..	..	..	..
D02120	..	..	..	..	..	..	..	..	..	..
D02130	..	..	..	..	..	..	..	..	..	..
D02140	..	..	..	..	..	..	..	..	..	..
D02150	..	..	..	..	..	..	..	..	..	..
D02160	..	..	..	..	..	..	..	..	..	..
D02170	..	..	..	..	..	..	..	..	..	..
D02180	..	..	..	..	..	..	..	..	..	..
D02190	..	..	..	..	..	..	..	..	..	..
D02200	yo	u.	..	..	..	..	..	..	..	..
D02210	..	..	..	..	..	..	..	..	..	..
D02220	..	..	..	..	..	..	..	..	..	..
D02230										

8.1.5.2 Rung 0



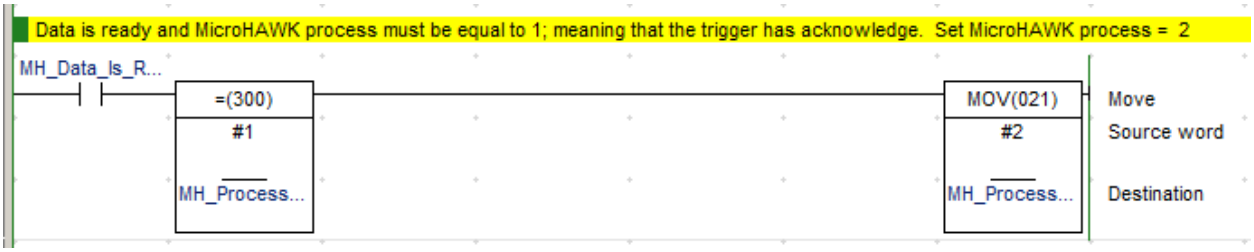
Triggers the unit if the following conditions are met

1. Unit is Online
2. Unit is Not Decoding
3. MH_Process state is Zero (0)

8.1.5.3 Rung 1

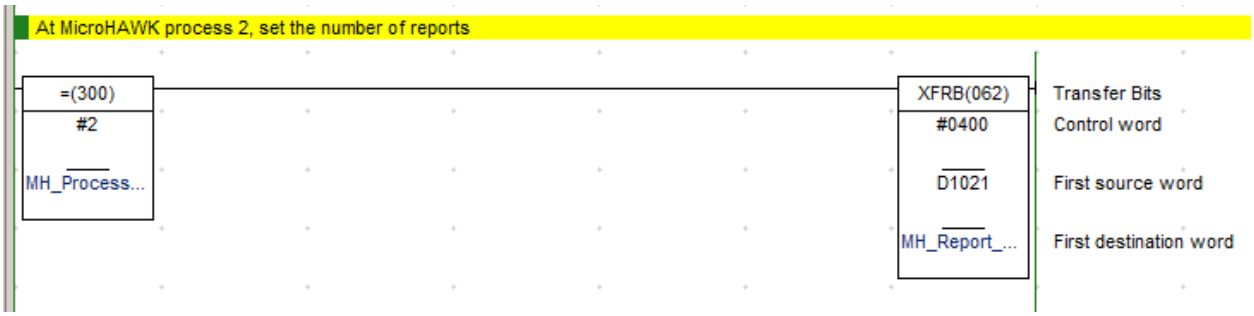
When the unit acknowledges the trigger it will set the MH_Process state to 1

8.1.5.4 Rung 2



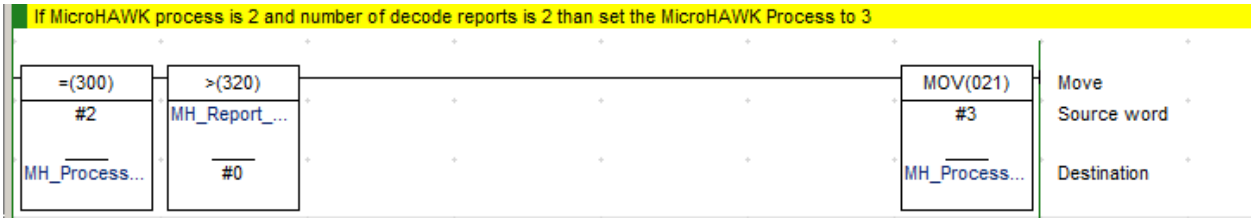
When the unit has acknowledged that the Data is Ready to be read by the PLC it will raise the Data_Is_Ready bit. This will alert the PLC and the MH_Process will be set to 2

8.1.5.5 Rung 3



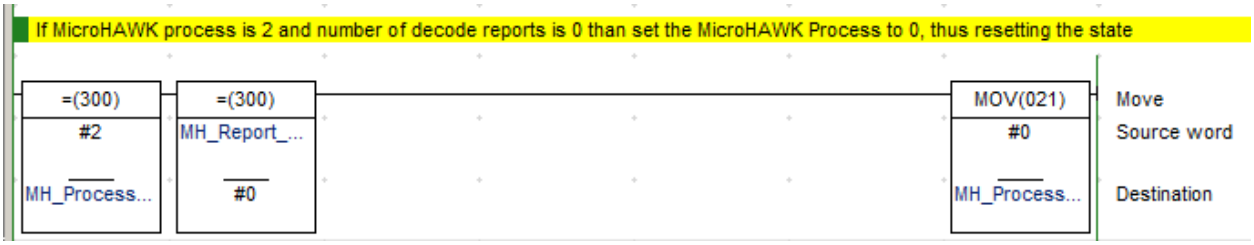
The PLC will read how many Decode Reports are in the Input N Assembly. Byte position 42 (WORD position 21) is the location of the decode report number in the Input N Assembly. This rung sets the tag MH_Report_Count to the correct value by using a transfer bit instruction.

8.1.5.6 Rung 4



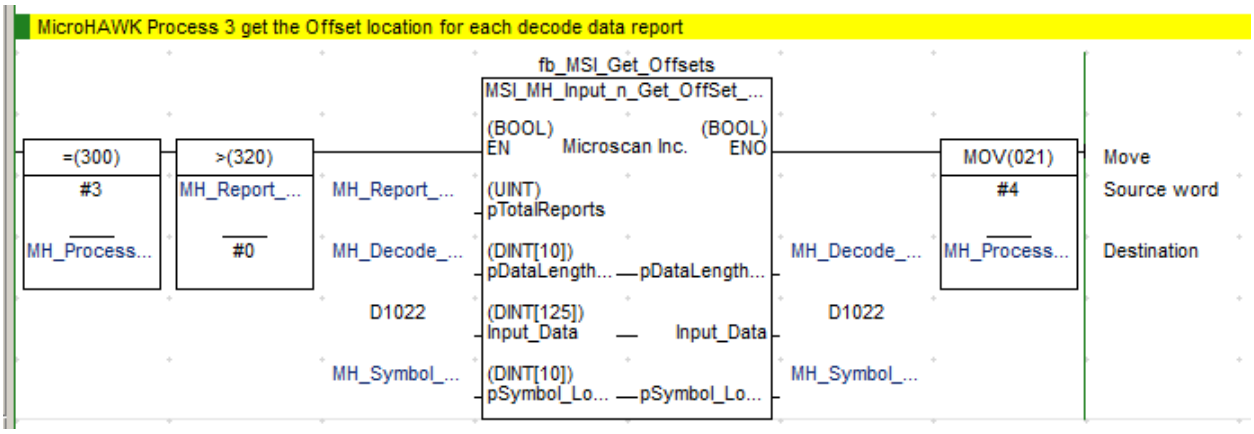
Increments that MH_Process state to 3 if there is any decode data to process. MH_Process state must be set to 2 for this rung to be evaluated.

8.1.5.7 Rung 5



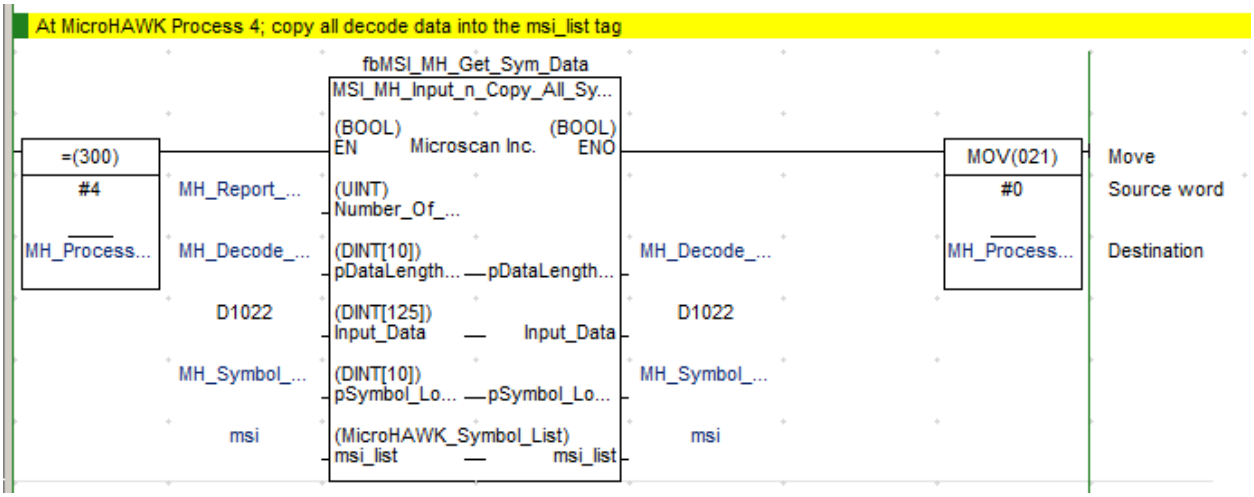
If the MH_Report_Count is 0 and MH_Process state is 2 than this resets the MH_Process back to 0, thus resetting the state

8.1.5.8 Rung 6



If the MH_Process state is 3 and the MH_Report_Count is greater than 0, than the function block will execute retrieving the offset location of the decode strings. After the function block as completed the MH_Process state will be set to 4

8.1.5.9 Rung 7

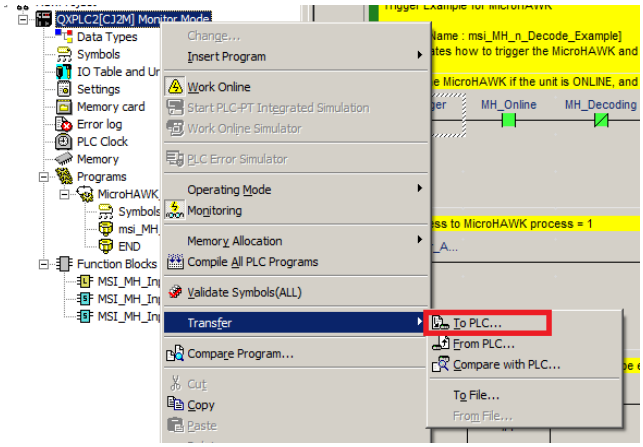


If the MH_Process state is 4 than the state machine has successfully triggered the unit, retrieved the total number of decode reports and the offset location for each decode string. The function block will move the decode data to the tag **msi** where the program can access the decode data. After the function block has executed the MH_Process state is reset to 0, thus resetting the state machine. If the MH_Trigger bit is HIGH than the process will start again.

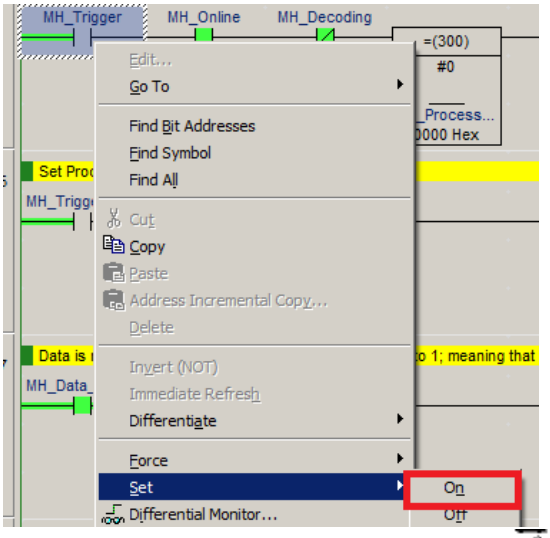
To test the routine do the following:

- Complete sections 8.11 through 8.1.4.
- Go online with the PLC
- Compile the program

- Download to the PLC



- Set MH_Trigger to ON



- Monitor the data in the Memory window