OMRON
MICROSCAN

AutoVISION and VISIONSCAPE®
Industrial Protocol Manual

v9.3.0, December 2022

84-9220126-02 Rev E

Disclaimer

The information and specifications described in this manual are subject to change without notice.

Latest Manual Version or Technical Support

For the latest version of this manual, or for technical support, see your local Omron website. Your local Omron website can be located by visiting **https://www.ia.omron.com** and selecting your region from the Global Network panel on the right side of the screen.

Security Measures

Anti-Virus Protection

Install the latest commercial-quality antivirus software on the computer connected to the control system and maintain to keep the software up to date.

Security Measures to Prevent Unauthorized Access

Take the following measures to prevent unauthorized access to our products:

- Install physical controls so that only authorized personnel can access control systems and equipment.
- Reduce connections to control systems and equipment via networks to prevent access from untrusted devices.
- Install firewalls to shut down unused communications ports and limit communications hosts and isolate control systems and equipment from the IT network.
- Use a virtual private network (VPN) for remote access to control systems and equipment.
- Adopt multifactor authentication to devices with remote access to control systems and equipment.
- Set strong passwords and change them frequently.
- Scan for viruses to ensure safety of USB drives or other external storage devices before connecting them to control systems and equipment.

Data Input and Output Protection

Validate backups and ranges to cope with unintentional modification of input/output data to control systems and equipment.
- Check the scope of data.
- Check validity of backups and prepare data for restore in case of falsification or abnormalities.
- Safety design, such as emergency shutdown and fail-soft operation in case of data tampering or abnormalities.

Data Recovery

Back up and update data periodically to prepare for data loss.
When using an intranet environment through a global address, connecting to an unauthorized terminal such as a SCADA, HMI or to an unauthorized server may result in network security issues such as spoofing and tampering.
You must take sufficient measures such as restricting access to the terminal, using a terminal equipped with a secure function, and locking the installation area by yourself.
When constructing an intranet, communication failure may occur due to cable disconnection or the influence of unauthorized network equipment. Take adequate measures, such as restricting physical access to network devices, by such means as locking the installation area.
When using a device equipped with the SD Memory Card function, there is a security risk that a third party may acquire, alter, or replace the files and data in the removable media by removing or unmounting the removable media. Please take sufficient measures, such as restricting physical access to the controller or taking appropriate management measures for removable media, by means of locking the installation area, entrance management, etc.

Software

To prevent computer viruses, install antivirus software on the computer where you use this software. Make sure to keep the antivirus software updated.
Keep your computer's OS updated to avoid security risks caused by a vulnerability in the OS.
Always use the latest version of this software to add new features, increase operability, and enhance security.
Manage usernames and passwords for this software carefully to protect them from unauthorized uses.
Set up a firewall (e.g., disabling unused communication ports, limiting communication hosts, etc.) on a network for a control system and devices to separate them from other IT networks.
Make sure to connect to the control system inside the firewall.
Use a virtual private network (VPN) for remote access to a control system and devices from this software.

Contents

Contents

PREFACE ⸻ Welcome

Purpose of This Manual

This manual contains detailed information about how to configure and deploy EtherNet/IP and PROFINET I/O-based applications using AutoVISION, Visionscape, and Omron Microscan Smart Cameras.

Manual Conventions

The following typographical conventions are used throughout this manual.

- Items emphasizing important information are **bolded**.

- Menu selections, menu items and entries in screen images are indicated as: Run (triggered), Modify..., etc.

Preface

1

CHAPTER 1 — Enabling Industrial Protocols

This section describes how to enable EtherNet/IP for an Omron Microscan Smart Camera, and how to switch the camera's protocol between EtherNet/IP and PROFINET I/O.

Enabling Protocols

FrontRunner

Go to the **File** menu and select **Configure Device**. Go to the **Industrial Protocols** tab. Select the industrial protocol you intend to use with the camera.
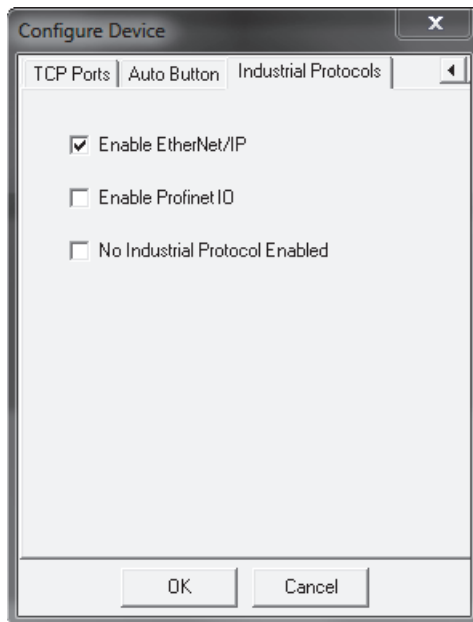
Note: When enabling, changing, or disabling protocols, the camera must be rebooted before the change will take effect. After clicking **OK**, you will be given the option to reboot the camera now or at a later time.

If you choose **No**, the change will not take effect until you manually reboot the camera.
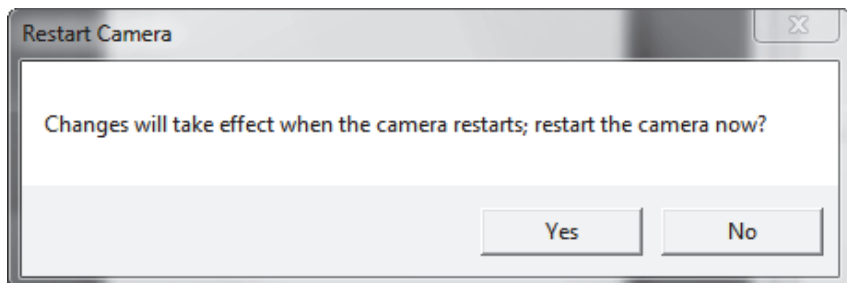
AutoVISION

In the **Connect** view, and with a camera selected, click the button to the left of **Details** to view camera settings. Click **Modify** to change camera settings. Select the desired protocol from the **Industrial Protocol** dropdown menu, and then click **Apply**.

2

Using EtherNet/IP

This section provides information necessary for using an Omron Microscan Smart Camera in an EtherNet/IP environment.

Notes:

- The camera communications protocol must be enabled for EtherNet/IP before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

- AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate EtherNet/IP communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

EtherNet/IP

Overview

The EtherNet/IP interface will be identified as Vendor Specific (100). The interface is designed to support Class 1 Implicit I/O data exchange, and Class 3 Explicit messages for serial commands not accessible with Implicit messaging.

Necessary Tools

The following tools are helpful for configuring EtherNet/IP:

• AutoVISION and FrontRunner

• EtherNet/IP Messaging Tool – can be a PLC or Software Tool, must be capable of sending explicit messages and establishing Class 1 connections. EIPScan from Pyramid Solutions is an example of such a tool.

• Terminal emulation or serial communication tool that can connect to serial uart and TCP socket, such as HyperTerminal or Putty.

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 **www.odva.org** for details.

EtherNet/IP Object Model

The HAWK MV-4000 and MicroHAWK use Class 1 connected messaging to communicate most of their data and services in a single connection.

EtherNet/IP Identity

Device Type: Device type is 100, Vendor-Specific, Machine Vision Smart Camera.

Vendor ID: Omron Microscan's ODVA Vendor ID is 1095.

Product Code: See table below.

Interface Revision: See table below.

Connection Properties: Class 1 Implicit Messaging

Input Assembly Instance (to PLC/client): 102

Output Assembly Instance (to camera): 114

Size: Fixed, 320 bytes in both directions

Input Trigger/Trigger Mode: Cyclic

RPI (Requested Packet Interval): 8 ms is the minimum RPI for the F440-F. **16 ms** is the minimum RPI for the MV-40, F430-F, and F330-F. **4 ms** is the minimum RPI for the HAWK MV-4000.

Input Type/Connection Type:

- Point-to-Point (PLC OUT, O > T)
- Point-to-Point (PLC IN, T > O)

Connection Priority: Scheduled

EtherNet/IP						
Product	**Code Version**	**EDS File**	**Version**	**Product Code**	**Device Major Rev**	**Device Minor Rev**
MV-40	5.1.0	MicroHAWKMV40(35-9000035-10).eds	1.0	6901	1	1
	5.2.0	MicroHAWKMV40(35-9000035-10).eds	1.0	6901	1	1
	5.2.2	MicroHAWKMV40_5_2_2_20210922.eds	1.0	6901	2	1
F430-F	5.2.2	F430-F_5_2_2_20210922.eds	1.0	6903	2	1
F330-F	5.2.2	F330-F_5_2_2_20210922.eds	1.0	6904	2	1
F440-F	5.3.0	F440-F_5_3_0_20220516.eds	1.0	6905	2	1
MV-4000	5.1.0	HAWK_MV-4000(8ABS-LFFA-LPPP).eds	1.1	6902	1	1
	5.2.0	HAWK_MV-4000(8ABS-LFFA-LPPP).eds	1.1	6902	1	1
	5.2.2	HAWK_MV-4000(8ABS-LFFA-LPPP).eds	1.1	6902	1	1

Assembly Layout

Input Assembly

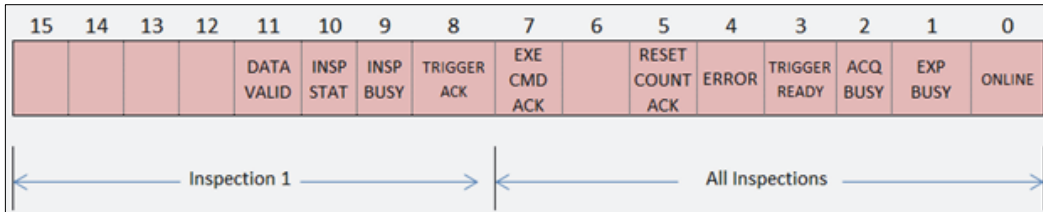The **input assembly** layout is described below.

Bytes	Name	Description
0...1	STATUS	Status register of the camera, each bit of this register represents a different state item. See *Camera Status Register* for bit descriptions
2...3	ECHO	This 16 bit word value reflects back to the PLC the value that the PLC wrote to the output assembly ECHO register. The PLC can verify the output assembly has been written to the camera when this value matches the written value.
4...7	CmdCodeRslt	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdCodeRslt reflects the result of the command invoked by Control.CmdCode. See *CmdCodeRslt* for definitions.
8...11	CmdRet	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdRet contains the data returned from the command invoked by Control.CmdCode. See *CmdRet* for definitions.
12...13	reserved	Reserved for future use.
14...15	State	Device State register. Depending on the current state of the camera, certain STATUS and CONTROL features may or may not be operational. See *State* for definitions.
16...17	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 145, the most significant bit vio point 160
18...19	reserved	Reserved for future use.
20...27	bool1-64	Each bit represents a bool value. The least significant bit of byte 20 reads the value of bool1. The most significant bit of byte 27 reads bool64.
28...47	int1-10	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 28 & 29 for the value of int1 through bytes 46 & 67 for the value of int10.
48...87	long1-10	Each group of 4 bytes represents a 32 bit signed integer value. The 40 bytes represent 10 long integers. From bytes 48-51 for the value of long1 through bytes 84-87 for the value of long10.
88...127	float1-10	Each group of 4 bytes represents a floating point value. The 40 bytes represent 10 floating point values. From bytes 88-91 for the value of float1 through bytes 124-127 for float10.
128...223	string1	These 96 bytes can store a string of up to 92, 8 bit characters, with the first 4 bytes containing the length value.
224...255	string2	Each of these 32 byte groups can store a string of up to 28, 8 bit characters, with the first 4 bytes containing the length value.
256...287	string3	
288...319	string4	

The input assembly layout is shown here:

Byte		Byte		Byte		Byte		Byte	
0	STATUS	64	long5	128		192		256	
2	ECHO	66		130		194		258	
4	CMD CODE RSLT	68	long6	132		196		260	
6		70		134		198		262	
8	CMD RET	72	long7	136		200		264	
10		74		138		202		266	
12	reserved	76	long8	140		204		268	
14	STATE	78		142		206	string1	270	string3
16	VIO	80	long9	144		208	(cont)	272	
18	reserved	82		146		210		274	
20	bool1...16	84	long10	148		212		276	
22	bool17...32	86		150		214		278	
24	bool33...48	88	float1	152		216		280	
26	bool49...64	90		154		218		282	
28	int1	92	float2	156		220		284	
30	int2	94		158		222		286	
32	int3	96	float3	160	string1	224		288	
34	int4	98		162		226		290	
36	int5	100	float4	164		228		292	
38	int6	102		166		230		294	
40	int7	104	float5	168		232		296	
42	int8	106		170		234		298	
44	int9	108	float6	172		236		300	
46	int10	110		174		238	string2	302	string4
48	long1	112	float7	176		240		304	
50		114		178		242		306	
52	long2	116	float8	180		244		308	
54		118		182		246		310	
56	long3	120	float9	184		248		312	
58		122		186		250		314	
60	long4	124	float10	188		252		316	
62		126		190		254		318	

Status: Camera Status Register (16-bit)

Each bit of this register represents a different state of the camera's operation. A high value of **1** indicates that state is active (true).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DATA VALID	INSP STAT	INSP BUSY	TRIGGER ACK	EXE CMD ACK		RESET COUNT ACK	ERROR	TRIGGER READY	ACQ BUSY	EXP BUSY	ONLINE

Inspection 1 ← → All Inspections

Bit	Name	Description
0	ONLINE	Inspections are running
1	EXP BUSY	The camera is busy capturing an image. The camera should not be triggered or the part under inspection moved during this time if illuminated. F440-F: EXP BUSY equivalent to ACQ BUSY.
2	ACQ BUSY	The camera is busy acquiring an image. The camera cannot be triggered while busy.
3	TRIGGER READY	The camera is ready to be triggered. This is equivalent to ONLINE == 1 and ACQ BUSY == 0.
4	ERROR	An error has occurred. Set the RESET ERROR control bit high to clear.
5	RESET COUNT ACK	This bit mirrors the RESET COUNT control bit. The PLC can be certain the reset command was received by the camera when this goes high. The PLC can then bring the RESET COUNT control signal back low.
7	EXE CMD ACK	This bit mirrors the EXE CMD control bit.
8	TRIGGER ACK	This bit mirrors the TRIGGER control bit.
9	INSP BUSY	This bit is high when inspection 1 is busy processing an image.
10	INSP STAT	This bit represents the inspection 1 status result. It is 1 if the inspection passes. It is only valid when DataValid goes high.
11	DATA VALID	This bit goes high when inspection 1 is complete. The PLC should clear this signal by setting RESET DV high once it has read results.

CmdCodeRslt (32-bit)

The value of **CmdCodeRslt** is only valid when **ExeCmdAck** is active (**1**), in response to **ExeCmd** being active.

CmdCodeRslt value (base 16 hex)	Meaning
0x0000_0000	Success
0x0100_0000	Fail. Possible reasons: Camera under PC control. Job cannot be changed.
0x0200_0000	Fail: No Job in slot.
0x0300_0000	Fail: Unknown cmd.

CmdRet (32-bit)

The value of **CmdRet** is only valid when **ExeCmdAck** is active (**1**), in response to **ExeCmd** being active, and **CmdCodeRslt** is **0** (**Success**). The following table shows which CmdCodes return data in the CmdRet register.

CmdRet value (32 bit)	Associated CmdCode	Meaning
0	0x1000_0000 to 0x1300_0000 (Job Change type)	Na
1 – 255	0x1800_0000 (Query Active Job Slot)	Active Job Slot #

State (16-bit)

State reflects the following operational condition of the camera:

State value (16 bit)	Meaning	Typical action required by the client (plc), or system operator
0	Offline	Perform job change or put camera online.
1	Online	Normal runtime operation: Monitor TriggerReady and DataValid signals. Trigger the camera.

2	Changing Vision Job	If camera is under pc control:
		    Wait until State changes to Offline or Online.
		If plc is controlling the job change:
		    Use ExeCmd, CmdCode, ExeCmdAck, and CmdCodeRslt to complete the operation.

3	Booting*	Wait for camera to transition to Online or Offline.
4	Empty (no Vision Job)	Load a new job from AutoVISION or Front Runner.

*Booting (3) State: This will rarely be seen by the PLC.

The value of **State** determines which **Control** and **Status** signals are available:

Control/Status Signal	State				
	0 (Offline)	1 (Online)	2 (Job Change)	3 (Booting)	4 (Empty)
Control.GO ONLINE	Y				
".GO OFFLINE		Y			
".RESET ERROR	Y	Y			Y
".RESET COUNT	Y	Y			
".EXE CMD	Y	Y	Y		Y
".TRIGGER		Y			
".RESET DATA VALID		Y			
Status.ONLINE	Y	Y	Y	Y	Y
".ERROR	Y	Y			Y
".RESET COUNT ACK	Y	Y			
".EXE CMD ACK	Y	Y	Y		Y
".EXP BUSY		Y			
".ACQ BUSY		Y			
".TRIGGER READY		Y			
".TRIGGER ACK		Y			
".INSP BUSY		Y			
".INSP STAT		Y			
".DATA VALID		Y			

Where:

Y = Signal is valid for this State.

Empty Table Cell = Signal is not valid for this State.

VIO Register Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v160	v159	v158	v157	v156	v155	v154	v153	v152	v151	v150	v149	v148	v147	v146	v145

Output Assembly

The **output assembly** layout is described below and shown in the following diagram.

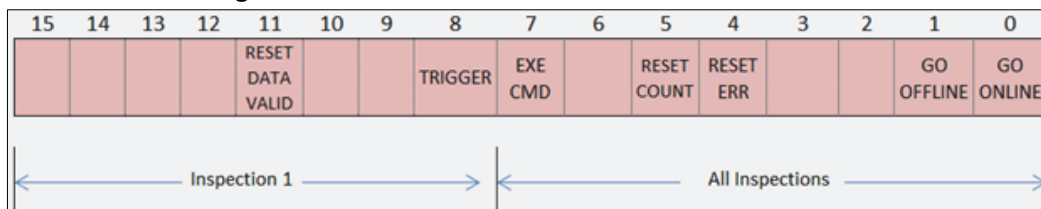| Bytes	Name	Description
0...1	CONTROL	Control register of camera. Each bit of this register represents a different status item. See *Camera Control Register* for bit descriptions
2...3	ECHO	This 16 bit value is reflected back to the PLC in the input assembly ECHO register. The PLC can verify the output assembly has been written to the camera when the input assembly matches this written value.
4...7	CmdCode	Specifies the process invoked in the camera when Control.ExeCmd goes active. See *CmdCode* for definitions.
8...11	CmdArg	Additional argument data for the CmdCode. See *CmdArg* for definition.
12...15	reserved	Reserved for future use.
16...17	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 129, the most significant bit is vio point 144
18...19	Reserved	Reserved for future use.
20...27	bool	Each bit represents a bool value. The least significant bit of byte 20 writes the value of bool101. The most significant bit of byte 27 writes bool164.
28...47	int101-110	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 28 & 29 to write the value of int101 through bytes 46 & 67 for the value of int110.
48...87	long101-110	Each group of 4 bytes represents a 32 bit signed integer value. The 40 bytes represent 10 long integers. From bytes 48-51 for the value of long101 through bytes 84-87 for the value of long110.
88...127	float101-110	Each group of 4 bytes represents a floating point value. The 40 bytes represent 10 floating point values. From bytes 88-91 for the value of float101 through bytes 124-127 for the value of float110.
128...223	string101	These 96 bytes can store a string of up to 92 bytes, with the first 4 bytes containing the length value.
224...255	string102	Each of these 32 byte groups can store a string of up to 28 bytes, with the first 4 bytes containing the length value.
256...287	string103	
288...319	string104	

The output assembly layout is shown here:

Byte		Byte		Byte		Byte		Byte	
0	CONTROL	64	long105	128		192		256	
2	ECHO	66		130		194		258	
4	CMD CODE	68	long106	132		196		260	
6		70		134		198		262	
8	CMD ARG	72	long107	136		200		264	
10		74		138		202		266	
12	reserved	76	long108	140		204		268	
14		78		142		206	string101	270	string103
16	VIO	80	long109	144		208	(cont)	272	
18	reserved	82		146		210		274	
20	bool101...116	84	long110	148		212		276	
22	bool117...132	86		150		214		278	
24	bool133...148	88	float101	152		216		280	
26	bool149...164	90		154		218		282	
28	int101	92	float102	156		220		284	
30	int102	94		158	string101	222		286	
32	int103	96	float103	160		224		288	
34	int104	98		162		226		290	
36	int105	100	float104	164		228		292	
38	int106	102		166		230		294	
40	int107	104	float105	168		232		296	
42	int108	106		170		234		298	
44	int109	108	float106	172		236		300	
46	int110	110		174		238	string102	302	string104
48	long101	112	float107	176		240		304	
50		114		178		242		306	
52	long102	116	float108	180		244		308	
54		118		182		246		310	
56	long103	120	float109	184		248		312	
58		122		186		250		314	
60	long104	124	float110	188		252		316	
62		126		190		254		318	

Control: Camera Control Register (16-bit)

Each bit of this register controls a function on the camera. Transitions from a low state of **0** to a high state of **1**, initiates the associate operation. The PLC should return the state of the control bit back to **0** after it has acknowledged the camera has processed the control. Unused bits should remain **0**. Setting the **Reset Data Valid** bit (bit **11**) will also reset the **Error** bit (bit **4**) in the **Camera Status Register**.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RESET DATA VALID			TRIGGER	EXE CMD		RESET COUNT	RESET ERR			GO OFFLINE	GO ONLINE

← Inspection 1 → ← All Inspections →

Bit	Name	Description
0	GO ONLINE	Start all inspections running
1	GO OFFLINE	Stop all inspections
4	RESET ERROR	Reset ERROR in the Status register
5	RESET COUNT	Reset all inspection counts
7	EXE CMD	Execute the command specified by Control.CmdCode
8	TRIGGER	Trigger Inspection 1. The inspection must be configured for a triggered image acquisition.
11	RESET DATA VALID	Reset the Data Valid signal of the Status register

CmdCode and CmdArg (32-bit)

Specifies the process invoked in the camera when **Control.ExeCmd** goes active. The **CmdCode** and **CmdArg** must be set, at least **2 RPI**, before setting the **EXE CMD** bit in the control register. Do not set all the values within the same RPI.

List of available CmdCodes, and associated CmdArg:

CmdCode value	CmdArg	Operations performed
0x1000_0000	Job Slot (1-255)	Go Offline, Load job from specified slot
0x1100_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Go Online
0x1200_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job
0x1300_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job, and Go Online
0x1800_0000	na	Query active job slot. *CmdRet* will contain the active job slot number when the operation is done.

CmdCode and ExeCmd Operation

Event key:

A. If DataValid or Error are present, clear them.
   Set the following control signals idle, and keep them idle while the command is processed by the camera:
      GoOffline, GoOnline, Trigger, ResetDataValid, ResetCount, ResetError.
   If the command operation is a job change, populate the output tags required to configure the new job (bool, int, long, float, string).
B. Populate CmdCode and CmdArg at least 2 RPI before activating ExeCmd.
C. Camera executes the command (may take up to a minute). While processing a Job Change command, State will be 2. Camera activates ExeCmdAck when it is done processing the command.
D. When the PLC sees an active ExeCmdAck, verify CmdCodeRslt is 0, and Error is 0. Process CmdRet if needed, then clear ExeCmd.
E. Camera clears ExeCmdAck when ExeCmd goes inactive. When ExeCmdAck goes inactive, CmdCodeRslt and CmdRet are no longer valid, and it may take a few seconds for the camera State and Online signals to settle to a final value (typically Online or Offline).
F. Camera can now be put online and triggered.

Notes:
st = PLC program scan time.
ct = Command processing time in the camera. May take up to a minute for some commands.
rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.
All signals represent the state of plc tags.

VIO Register Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v144	v143	v142	v141	v140	v139	v138	v137	v136	v135	v134	v133	v132	v131	v130	v129

Connection Properties: Class 3 Explicit Messaging

All Class 1 I/O assembly data and additional data are accessible via Explicit message. Input data (camera to PLC/Client) occupies attributes **1** to **100** of the classes. Output data (PLC/Client to camera) occupies attributes **101** to **200**.

Note: Any explicit message that causes an action taking longer than 3 seconds will time out. Omron Microscan recommends using implicit messaging in these cases.

Service:

- Get Attribute Single (0xE)

- Set Attribute Single (0x10)

Classes:

- bool = 104 (0x68)

- int = 105 (0x69)

- long = 106 (0x6A)

- float = 107 (0x6B)

- string = 108 (0x6C)

- control/status (mixed data types) = 109 (0x6D)

Instance: 1

Attribute:

- 1 to 100 = In to PLC/Client

- 101 to 200 = Out to Camera

Attribute Layout

When using explicit EtherNet/IP messaging, all global data objects can be read or written. Each data type is stored in its own class object and an instance of 1 to read the global data. For example, to read **float2**, the EtherNet/IP request would be for **Service Code 14 (0xE)**, **Class 107 (0x6B)**, **Instance 1**, **Attribute 2**.

Class 104		Class 105		Class 106		Class 107		Class 108		Class 109	
Attr#		Attr#		Attr#		Attr#		Attr#		Attr#	
1	bool1	1	int1	1	long1	1	float1	1	string1	1	CONTROL
2	bool2	2	int2	2	long2	2	float2	2	string2	2	STATUS
3	bool3	3	int3	3	long3	3	float3	3	string3	3	
4	bool4	4	int4	4	long4	4	float4	4	string4	4	
5	bool5	5	int5	5	long5	5	float5	5	string5	5	
6	bool6	6	int6	6	long6	6	float6	6	string6	6	ECHO
7	bool7	7	int7	7	long7	7	float7	7	string7	7	CMD CODE
8	bool8	8	int8	8	long8	8	float8	8	string8	8	CMD ARG
9	bool9	9	int9	9	long9	9	float9	9	string9	9	CMD CODE RSLT
10	bool10	10	int10	10	long10	10	float10	10	string10	10	CMD RET
...	...	...	...	...	...	...	...	...	...	11	STATE
										...	
199	bool199	199	int199	199	long199	199	float199	199	string199	199	
200	bool200	200	int200	200	long200	200	float200	200	string200	200	

The value received in response to **Get Attribute Single** depends on the type:

- **Bool** will return a 16-bit word with **0** for false or **1** for true

- **Ints** will return a 16-bit signed integer

- **Longs** will return a 32-bit signed integer

- **Floats** will return a 32-bit floating point number

- **Strings** will return a counted string. Total size of a string data item is **2048 bytes**. This includes a 4 byte "length" field followed by 2044 eight bit characters. When accessing strings explicitly, they are not limited to the size in the I/O assemblies. For example, **string3** is limited to **28 bytes** in the input assembly. If the actual string is longer than 28 bytes, it will be truncated when reading via the assembly, but not truncated when reading the same string via an attribute explicitly.

Assembly Class 109 can be used to read and write special EtherNet/IP-specific registers.

Attr#	Name	Description
1	CONTROL	The control register (16 bit). See *Camera Control Register* for bit definitions.
2	STATUS	The status register (16 bit). See *Camera Status Register* for bit definitions.
6	ECHO	The ECHO register (16 bit) (read only if implicit write is enabled)
7	CMD CODE	The command code register (32 bit). See *CmdCode*.
8	CMD ARG	The command argument register (32 bit). See *CmdArg*.
9	CMD CODE RSLT	The command code result register (32 bit). See *CmdCodeRslt*.
10	CMD RET	The command return value register (32 bit). See *CmdRet*.
11	STATE	The device state register (16 bit). See *State* for definitions.

EtherNet/IP Control/Status Signal Operation

Event key:

A. On rising edge of system trigger, the user app activates Scan_MV_IO_user.OUT.Control.Trigger to trigger the demo code.

B. Demo code detects rising edge of Scan_MV_IO_user.OUT.Control.Trigger, and if the camera is ready, sends a trigger to the camera.

C. Camera acquisition begins (may be delayed by one rpi).

D. If the camera's exposure time is shorter than the rpi, no change will be seen in TriggerReady and AcqBusy plc IN tags.

E. Camera firmware acks the trigger. The demo code may not see the ack until two rpi after the trigger was sent (event B).

F. Demo code detects TriggerAck and clears the Trigger.

G. Demo code detect falling edge of TriggerAck and clears the user Trigger.

H. Camera internal signal DataValid will go high when InspBusy goes low

I. Plc logic must delay one rpi time before re-asserting ResetDataValid

Notes:

1. The chart shows the workings of the Trigger and ResetDataValid Control signals, and the TriggerAck and DataValid Status signals.

2. st = plc program scan time

3. rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.

4. All signals represent the state of plc tags, except where noted as "(camera)". The cam signals shown are visible in the EIP interface, but the state of the plc tags and internal firmware signals will be different for at least one or two requested packet intervals (rpi).

5. The plc is running the demo code distributed with the camera. The demo code and user app use the Scan_MV_IO_user tag set as the primary control, status, and data interface for the user app. All signal operations are still true even if the plc demo code is not used.

6. TriggerReady/!AcqBusy: Camera exposure times can range from less than 1 ms, up to 100 ms.

7. On F440-F platform ExpBusy and AcqBusy are the same signals.

Data Type Descriptions and Equivalents in PLC and EDS/CIP Environments

AV	Description	RSLogix equivalent	Description	EDS / EIP equivalents	Description
Bool	1 bit	BOOL	1 bit	BOOL	1 bit
				WORD	16 BOOLs
				LWORD	64 BOOLs
Int	16 bit signed integer	INT	16 bit signed integer	INT	16 bit signed integer
Long	32 bit signed integer	DINT	32 bit signed integer	DINT	32 bit signed integer
Float	32 bit floating point	REAL	32 bit floating point	REAL	32 bit floating point
String	32 bit length field followed by 8 bit ASCII characters	STRING	32 bit length field followed by 8 bit ASCII characters	DINT + USINT[]	DINT (length) + USINT array of characters. USINT = 8 bit integer

PLC Tags and Serial Command Names

PLC tags are separated into **IN** and **OUT** for data direction. Within the IN and OUT groups, the tags are sub-divided into fixed **Status** and **Control** fields, plus user-defined linked data fields. This table shows how PLC tag names correspond to serial commands.

IN			OUT		
PLC tag prefix	Serial cmd prefix	Tag name	PLC tag prefix	Serial cmd prefix	Tag name
IN.Status.	eip.status.	Online (1)	OUT.Control.	eip.control.	GoOnline [i]
IN.Status.	eip.status.	Online (0)	OUT.Control.	eip.control.	GoOffline [ii]
IN.Status.	eip.status.	Error	OUT.Control.	eip.control.	ResetError
IN.Status.	eip.status.	ResetCountAck	OUT.Control.	eip.control.	ResetCount
IN.Status.	eip.status.	TriggerAck	OUT.Control.	eip.control.	Trigger
IN.Status.	eip.status.	DataValid	OUT.Control.	eip.control.	ResetDataValid
IN.Status.	eip.status.	ExeCmdAck	OUT.Control.	eip.control.	ExeCmd
IN.Status.	eip.status.	TrigReady[iii]	-	-	-
IN.Status.	eip.status.	AcqBusy	-	-	-
IN.Status.	eip.status.	ExpBusy	-	-	-
IN.Status.	eip.status.	InspBusy	-	-	-
IN.Status.	eip.status.	InspStat	-	-	-
IN.Status.	eip.	Echo	OUT.Control.	eip.	Echo
IN.Status.	eip.	CmdCodeRslt	OUT.Control	eip.	CmdCode
IN.Status	eip.	CmdRet	OUT.Control	eip.	CmdArg
IN.Status.	eip.	State	-	-	-
IN.vio.	io.	v[145-160]	OUT.vio.	io.	v[129-144]
IN.bool.	eip.	bool[1-100]	OUT.bool.	eip.	bool[101-200][iv]
IN.int.	eip.	int[1-100]	OUT.int.	eip.	int[101-200][v]
IN.long.	eip.	long[1-100]	OUT.long.	eip.	long[101-200]
IN.float.	eip.	float[1-100]	OUT.float.	eip.	float[101-200]
IN.string.	eip.	string[1-100]	OUT.string.	eip.	string[101-200]

[i] When GoOnline is changed from 0 to 1, Online goes to 1.

[ii] When GoOffline is changed from 0 to 1, Online goes to 0.

[iii] TrigReady, AcqBusy, ExpBusy, InspBusy, and InspStat are all IN-direction data only.

[iv] bool1-bool64 are mapped to PLC tags in the IN assembly. Bool101-bool164 are mapped to PLC tags in the OUT assembly. Bool members numbered 65-100 and 165-200 are accessible via Explicit Message only.

[v] For int, long, float, and string data:

Data members numbered 1-10 are mapped to PLC tags in the IN assembly.

Data members numbered 101-110 are mapped to PLC tags in the OUT assembly.

Data members numbered 11-100 and 111-200 are accessible via Explicit Message only.

Chapter **2**   Using EtherNet/IP

3

Allen-Bradley AOI (Add-On Instructions) for EtherNet/IP Operation

This section provides additional instructions helpful for using an Omron Microscan Smart Camera in an EtherNet/IP environment.

Notes:

• The camera communications protocol must be enabled for EtherNet/IP before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

• AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate EtherNet/IP communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

Rockwell RSLogix 5000 AOI (Add-On Instructions) for Omron Microscan Devices

The AOI file has been created as basic instructions and as a main rung import. The AOI instructions read and write data when called. The main rung import has global tags and the AOI itself to demonstrate how the AOI is used for beginners with the RSLogix system. The AOI can be used with the EDS file.

Note: Examples in this section have been created using RSLogix 5000 version 20.

EDS Files (Located in Default Directory C:\Omron\Vscape\Firmware \eds\"Product Name"):

The F330-F, F430-F, and F440-F EDS files are:

F330-F_5_2_2_20210922.eds

F430-F_5_2_2_20210922.eds

F440-F_5_3_0_20220516.eds

The HAWK MV-4000's EDS file is: **HAWK_MV-4000(8ABS-LFFA-LPPP).eds**

The MV-40's EDS file is: **MicroHAWKMV40_5_2_2_20210922.eds**

AOI File (Located in Default Directory C:\Omron\Vscape\Firmware \aoi\"Product Name"):

The AOI file is: **OMRONMicroscan_MV_AOI.L5X**.

Steps

- Install EDS File
- Import AOI File
- Test Communications and Review Data

Install EDS File

In RSLogix 5000, select the **EDS Hardware Installation Tool** under the main menu **Tools**. Click the **Next** button. Make sure the **Register an EDS Files(s)** radio button is selected.
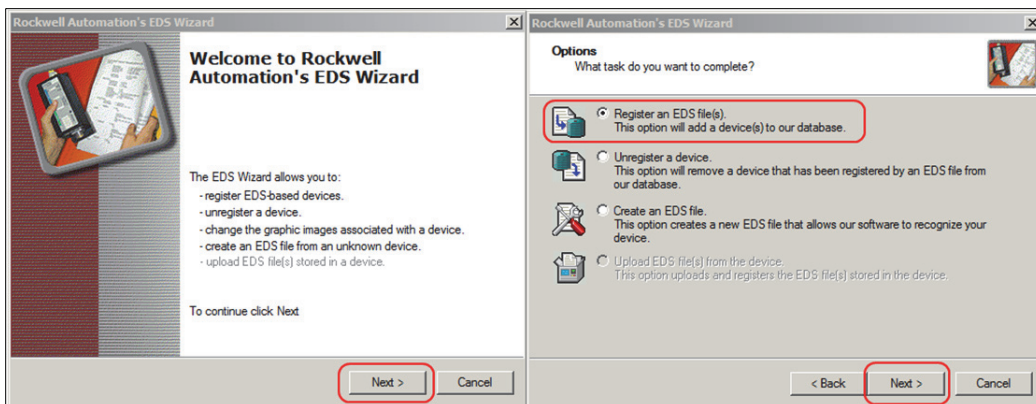
Click the **Browse** button to locate the EDS file (C:\Omron\Vscape\Firmware\eds\).
Once the EDS file is located and selected, click the **Next** button.
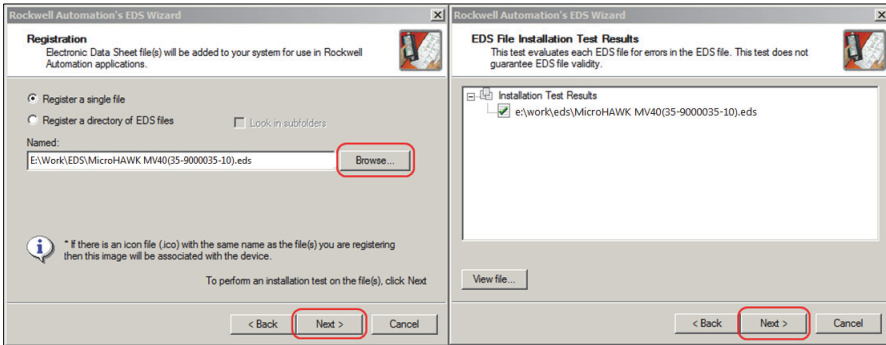
Click the **Next** button for the image. Click the **Next** button for the summary.
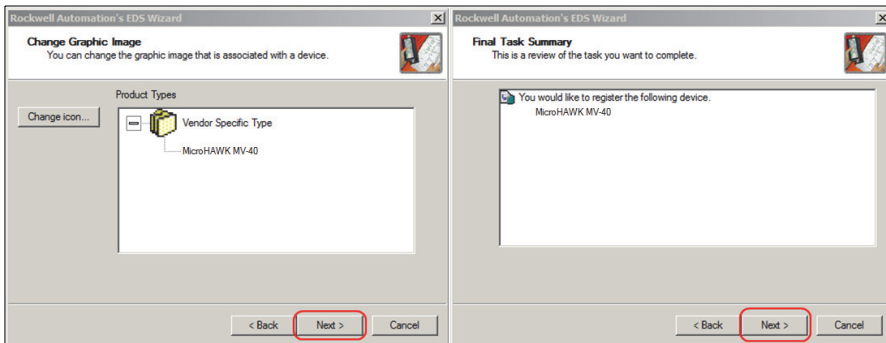
Click the **Finish** button to complete the EDS installation.
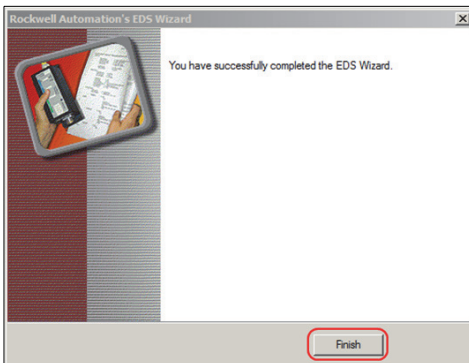
Right-click the Ethernet node on the left pane and select **New Module**. Type **Omron Microscan** in the filter box to list the device. Select the device from the newly added EDS file. Double-click the device or select and click the Create button to add to the project.
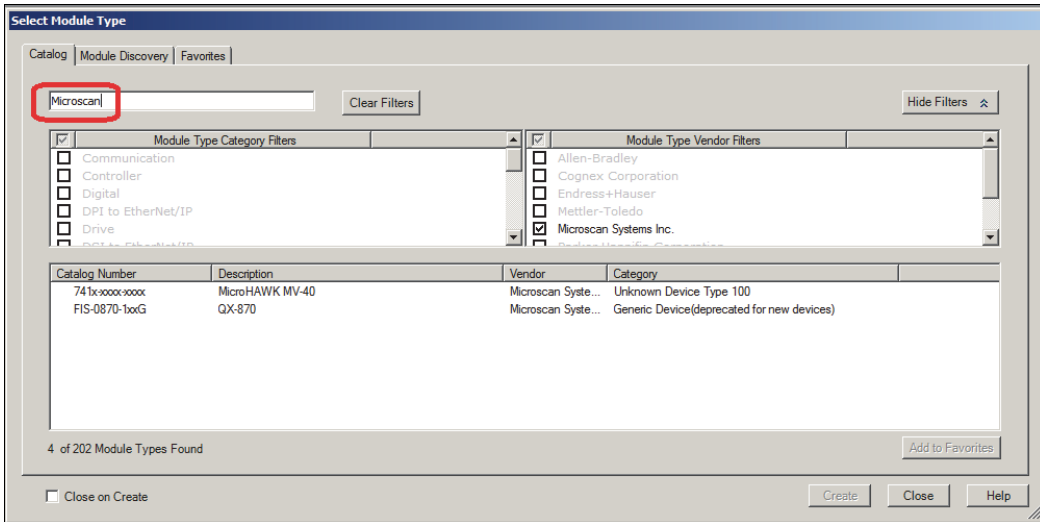
Enter the name for your device and the IP address, then click **OK**.
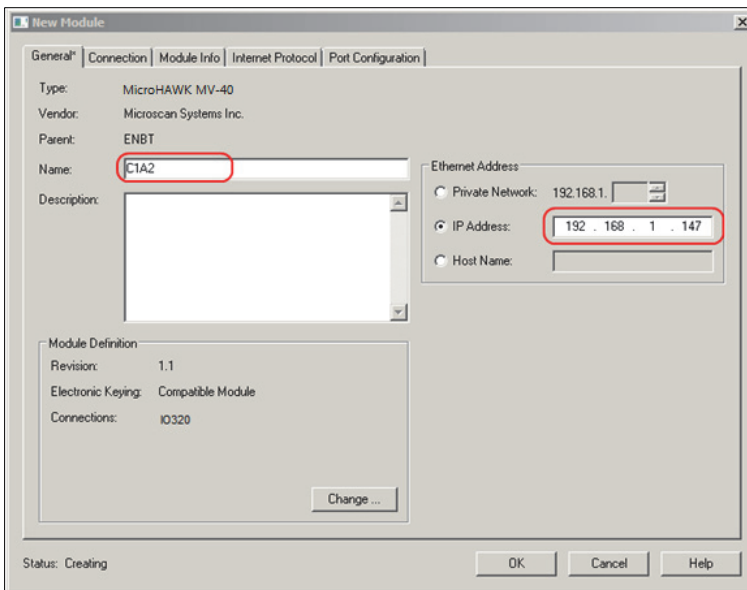
If the camera has multiple assembly sizes, the **Change** button allows you to select the other assembly formats.

Click the **Close** button on the module selection dialog to continue. Now the device has been added to the project and will be visible in the tree view under the **Ethernet** node.
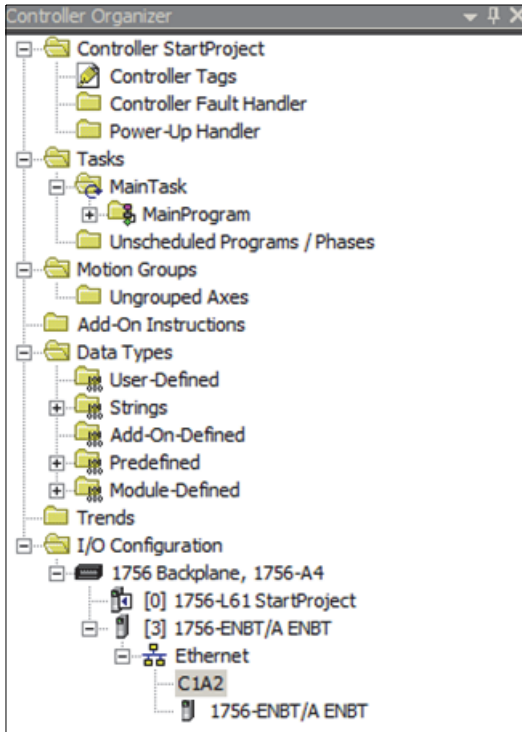
Import AOI File

Right-click the **Add-On Instructions** node in the tree view in the left pane and select **Import Add-On Instruction**. Locate the **L5X file** (C:\Omron\Vscape\Firmware\aoi\ "Product Name") and click the **Import** button.
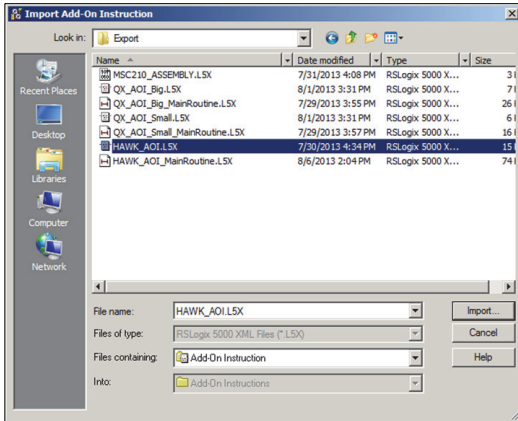
The **Import Configuration** dialog will prompt you for information regarding the AOI file. Select the **Data Types** to view the new tags and their attributes. Click **OK** when ready to continue.
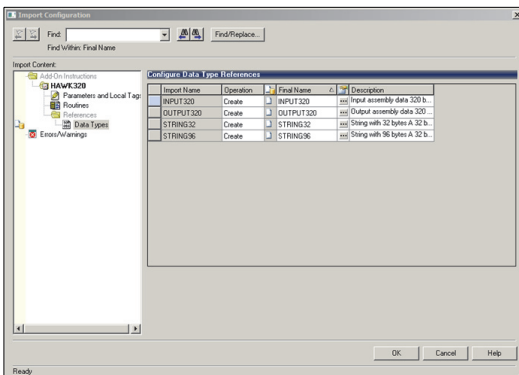
New tags and logic will now be added to the project.
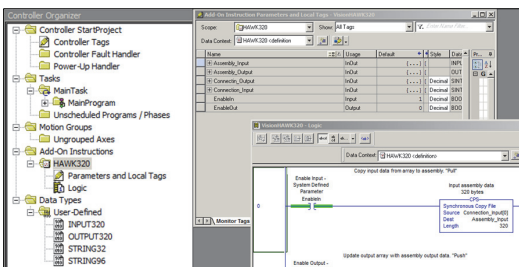
Test Communications and Review Data

After the reader module has been installed, you can start with a basic ladder logic program to test the data to and from the device. Two steps are involved to test the communication: (1.) Download an autovision job; (2.) Add logic to the PLC ladder logic.

Start by opening up **AutoVISION** and select the camera.

Select **Load a Job**.
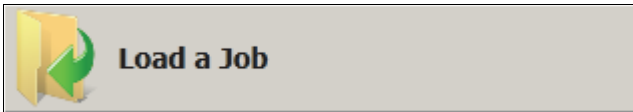
Navigate to **Omron > Vscape > Tutorials and Samples > "Product Name" > EIP Demo > SimpleTrigger_Job.avp**.
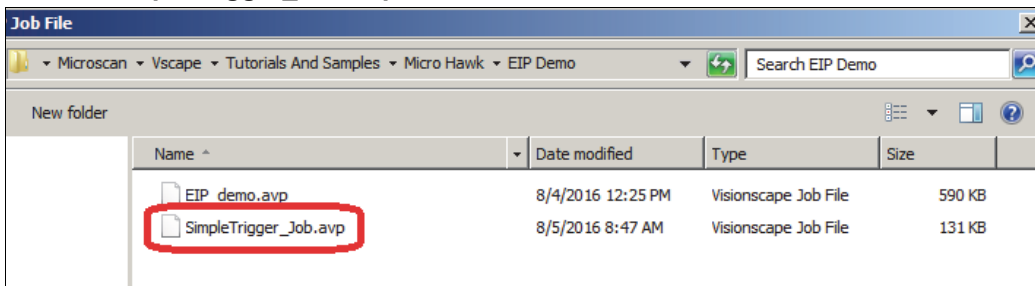
This job has no tools loaded and is a simple job that allows the PLC to trigger the camera. You can also set another Trigger action in the Trigger panel on the right side of the screen.

Now download the job to the camera by selecting the Save Job to Camera icon and selecting **New Slot**.
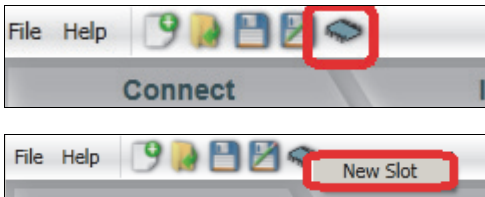
Select **Yes** to downloading the program to flash to the camera.
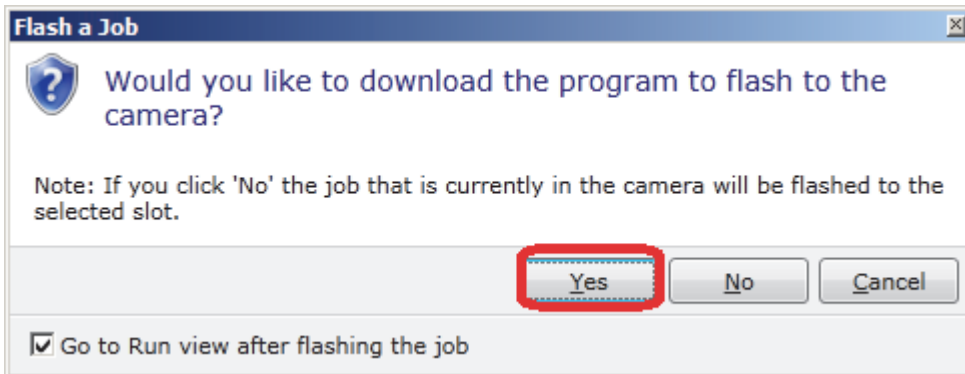
Then add the logic to the PLC. Do this by opening the **MainRoutine** editor and create an **Add-On-Instructions**.
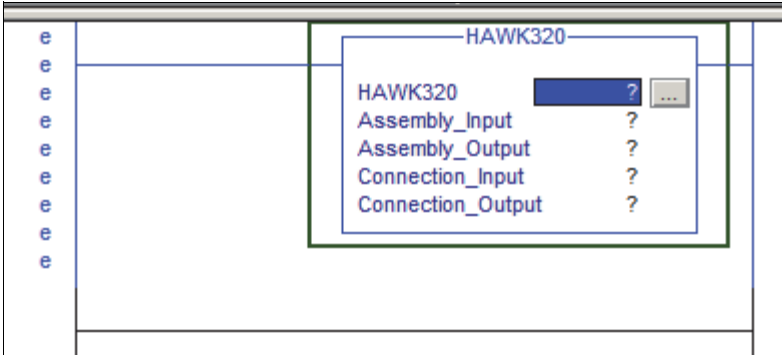
Right-click in the first element and create a **New Tag**.
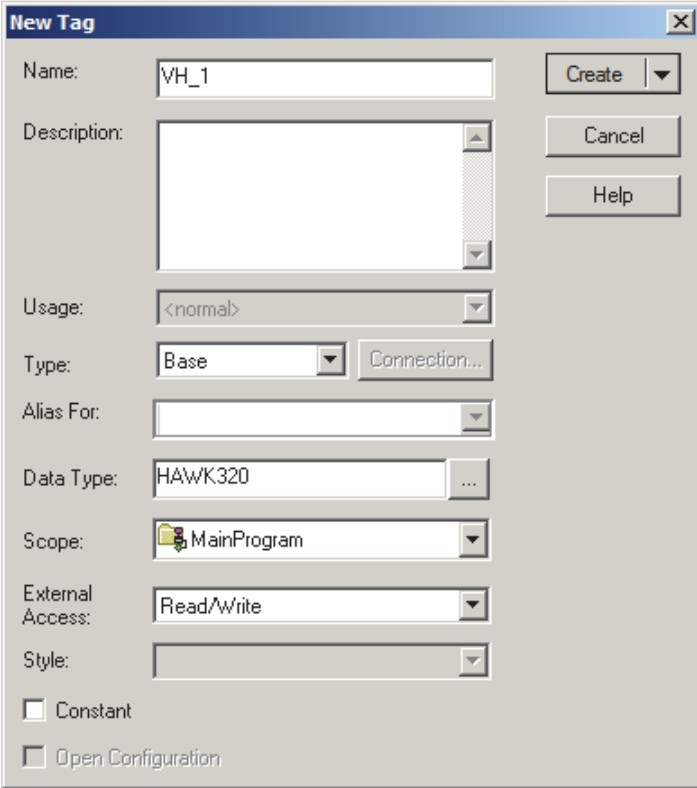
Right-click the second element and create a new tag for the **Assembly_Input**. It will default to the **AOI INPUT320** data type. Right-click the third element and create a new tag for the **Assembly_Output**. It will default to the **AOI OUTPUT320** data type.

New Tag		New Tag	
Name: ReadData	Create ▼	Name: WriteData	Create ▼
Description:	Cancel	Description:	Cancel
	Help		Help
Usage: <normal>		Usage: <normal>	
Type: Base	Connection...	Type: Base	Connection...
Alias For:		Alias For:	
Data Type: INPUT320	...	Data Type: OUTPUT320	...
Scope: MainProgram		Scope: MainProgram	
External Access: Read/Write		External Access: Read/Write	
Style:		Style:	
☐ Constant		☐ Constant	
☐ Open Configuration		☐ Open Configuration	

Double-click the fourth element and click the down arrow in the combo box to link the **Connection_Input** to the reader input data. The link should be the **[product name]: I.Data**.

⊟-Camera:I	_0447:6800
—Camera:I.ConnectionFaulted	BOOL
⊞-Camera:I.Data	SINT[320]
⊞-Camera:O	_0447:6800

Note: Do not connect to the **ConnectionFault** item.

Double-click the fifth element and click the down arrow in the combo box to link the **Connection_Output** to the reader output data. The link should be the **[product name]: O.Data**.

⊟-Camera:I	_0447:6800
—Camera:I.ConnectionFaulted	BOOL
⊞-Camera:I.Data	SINT[320]
⊞-Camera:O	_0447:6800

Add a new rung by right clicking and selecting **Add Rung**.
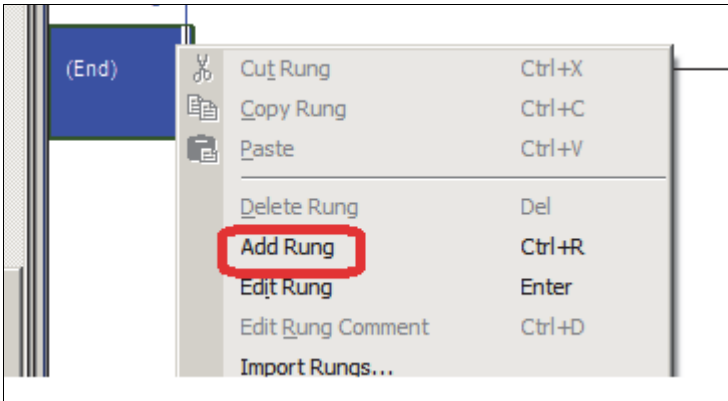
Create an **Examine On** trigger tag.
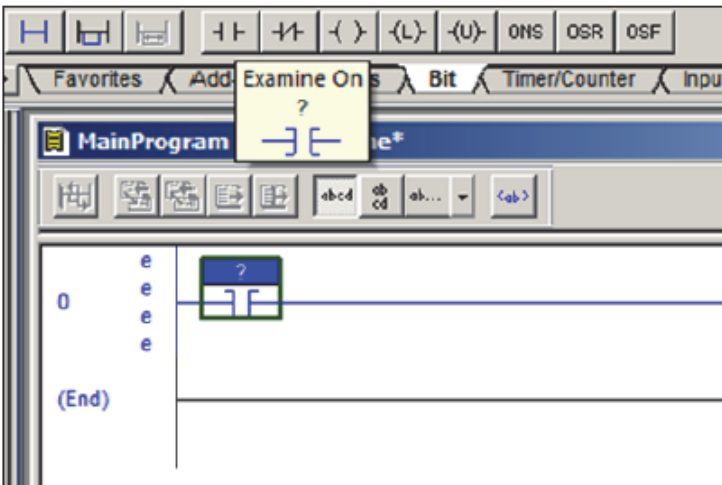
Right-click and create a **Boolean Trigger**.
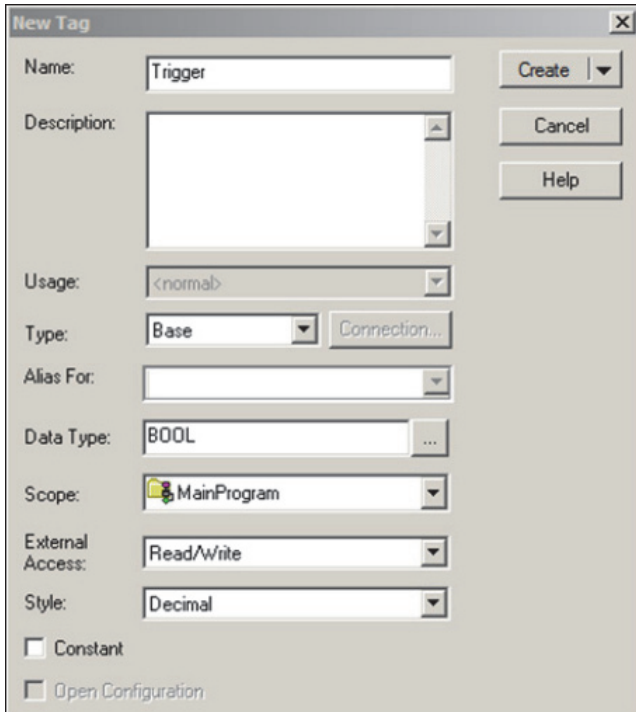
Add an **Output Energized** Tag to the end of the rung.

Assign the **Output Energized** tag to the **WriteData.CONTROL.8** bit

Now download the program to the PLC. Once the program has downloaded, set the PLC to **Run Mode**.
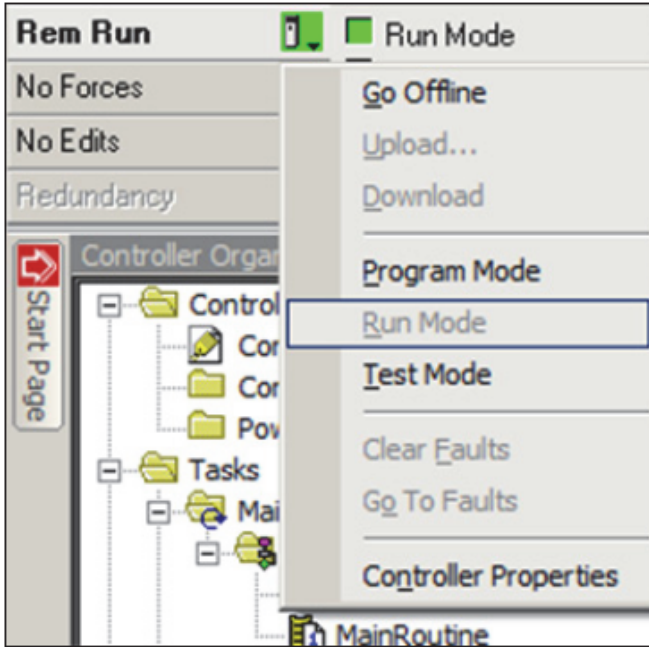
Note: Be sure the path to the PLC has been set in the project so that communications to the PLC can be established.
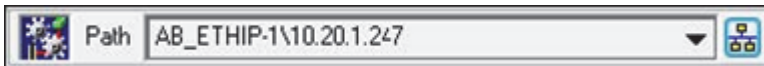
Toggle the bit **Trigger** by right-clicking and selecting **Toggle Bit** or by pressing **Ctrl + T**.

Trigger		
✂	Cu_t Instruction	Ctrl+X
🖹	_C_opy Instruction	**Ctrl+C**
📋	_P_aste	Ctrl+V
	_D_elete Instruction	Del
	_A_dd Ladder Element...	Alt+Ins
✎	Edit _M_ain Operand Description	Ctrl+D
	Save Instruction Defaults	
	Clear Instruction Defaults	
	Toggle Bit	**Ctrl+T**
	_F_orce On	
	F_o_rce Off	
	Re_m_ove Force	
	_G_o To...	Ctrl+G
	Instruction _H_elp	F1
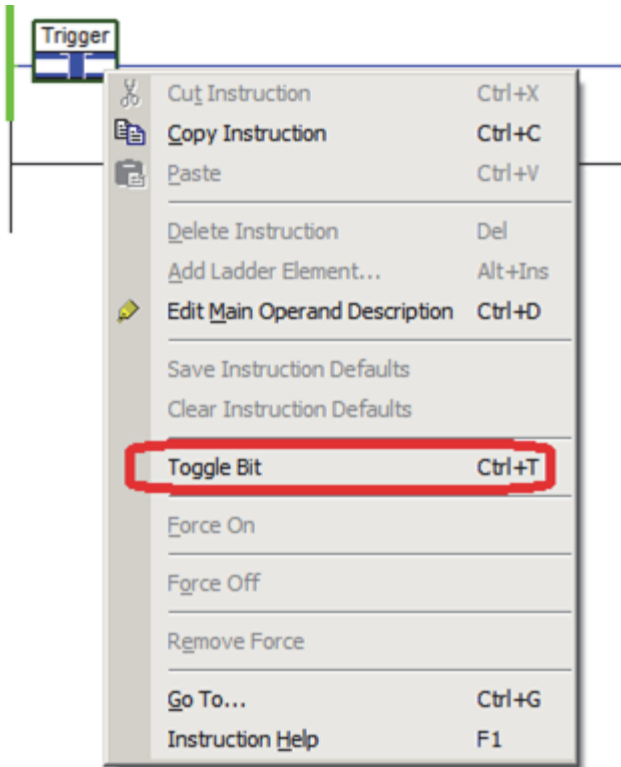
This action will set the **Trigger** tag high causing the trigger output bit to go high in the Output Assembly. The camera will trigger on each rising edge (ie when the **Trigger** tag is set to 1). This will confirm that communication is established between the PLC and the camera.

4

Allen-Bradley PLC Setup via EDS for EtherNet/IP Operation

This section describes how to use an EDS file to set up an Allen-Bradley PLC for EtherNet/IP operation.

Notes:

- The camera communications protocol must be enabled for EtherNet/IP before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

- AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate EtherNet/IP communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

- AutoVISION 5.2.3 and earlier only.

AB Rockwell RSLogix 5000 v20 PLC Integration with EDS

This section was created and run on the following Allen Bradley/Rockwell components:

* RSLogix 5000 Version 20.00.00 (CPR 9 SR 5)

* 756-L61 ControlLogix5561 Controller, firmware rev 20.11

* 1756-ENBT/A EtherNet/IP interface card, firmware rev 4.1

Run the Rockwell **EDS Hardware Installation Tool**.

Select **Add**.

Rockwell Automation - Hardware Installation Tool

**This tool allows you to change the hardware description
information currently installed on your computer.**

Add	Launch the EDS Wizard and add selected hardware description files and associated components only.
Remove	Launch the EDS Wizard and remove selected hardware description files and associated components only.
Remove All	Remove all previously installed hardware description files and associated components from your computer.

Exit

Select **Browse**.

Rockwell Automation's EDS Wizard

Registration
Electronic Data Sheet file(s) will be added to your system for use in Rockwell Automation applications.

○ Register a single file

○ Register a directory of EDS files ☐ Look in subfolders

Named:

[   ] Browse...

ⓘ * If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.

To perform an installation test on the file(s), click Next

Next >   Cancel

Navigate to the EDS file, then **Open** it. The default install location is:
For MicroHAWK: C:\Omron\Vscape\Firmware\eds\MicroHAWK
For HAWK MV-4000: C:\Omron\Vscape\Firmware\eds\HAWK MV4000
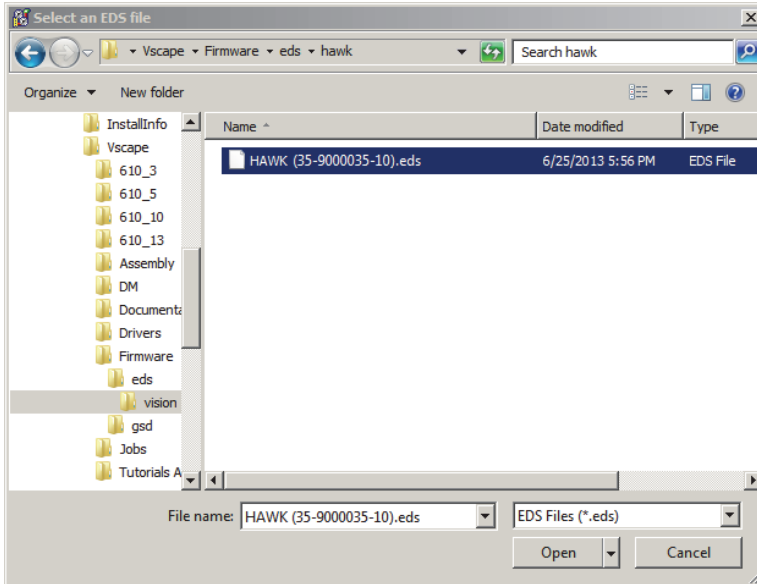
Keep clicking **Next** until the **Finish** button is displayed.
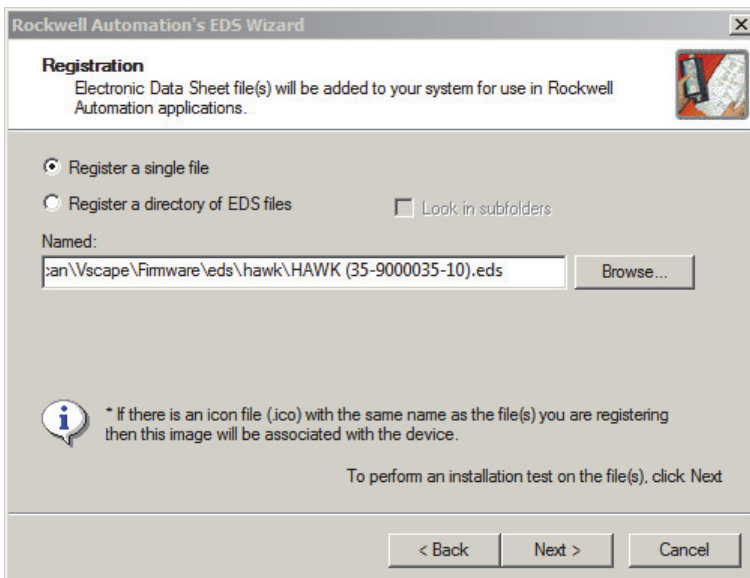
Click **Finish**.
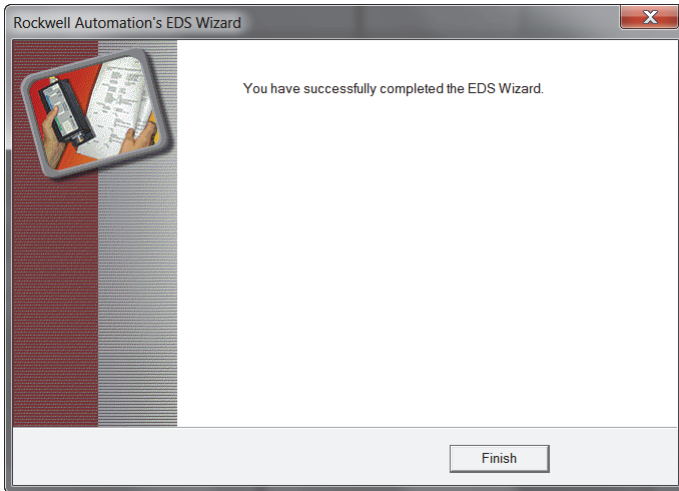
Open RSLogix 5000 v20 and create the **I/O Configuration** for the base system, including the system's Ethernet interface.
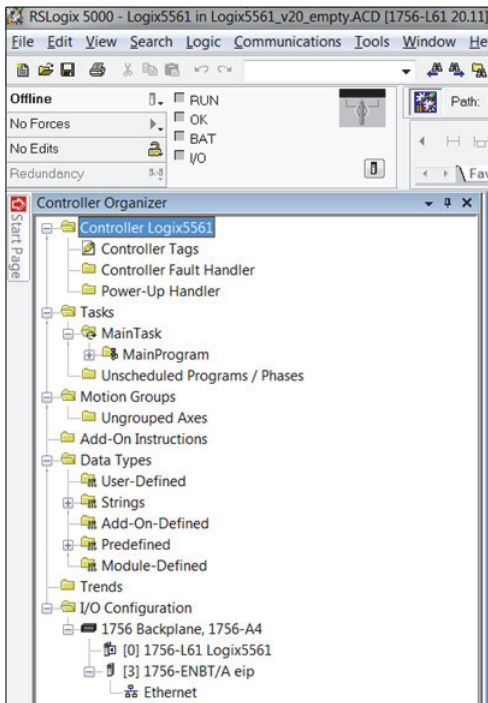
Right-click **Ethernet** and select **New Module**.
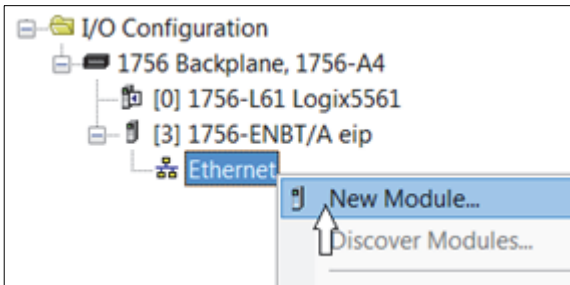
The **Select Module Type** dialog will be displayed.
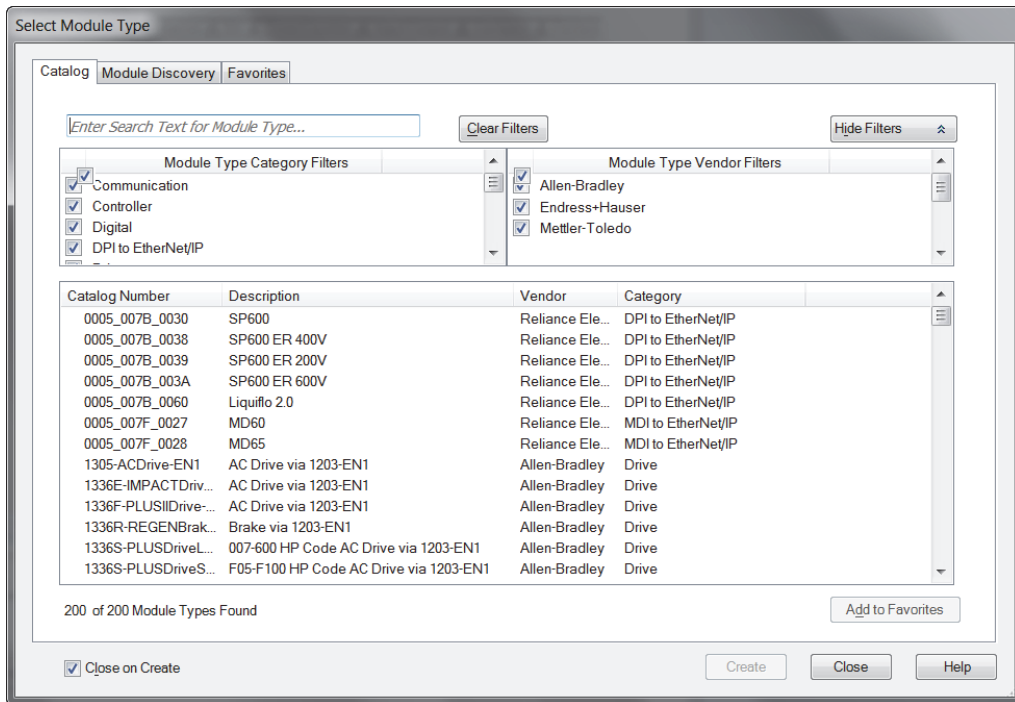
Clear the **Module Type Vendor Filters**.
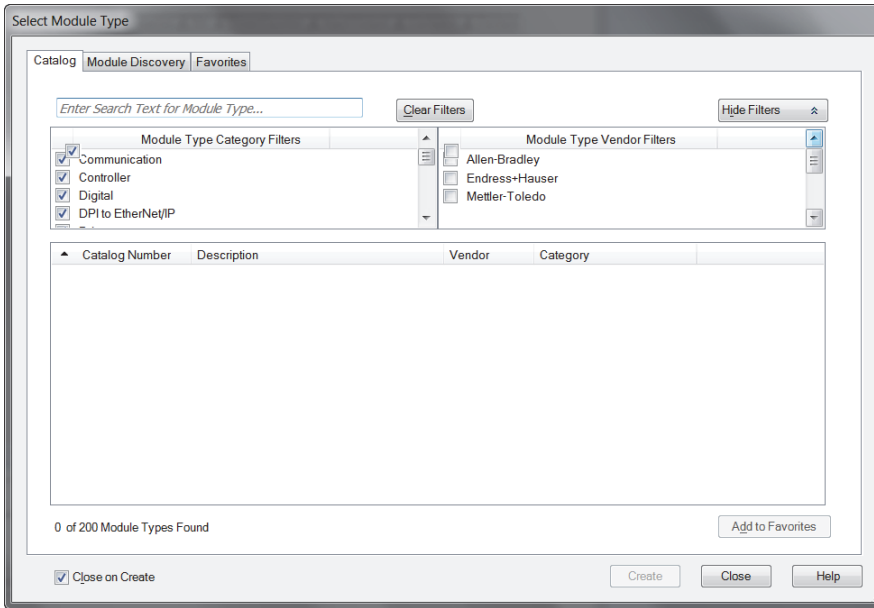
Scroll down the Module Type Vendor Filters until **Omron Microscan** comes into view, then select Omron Microscan.
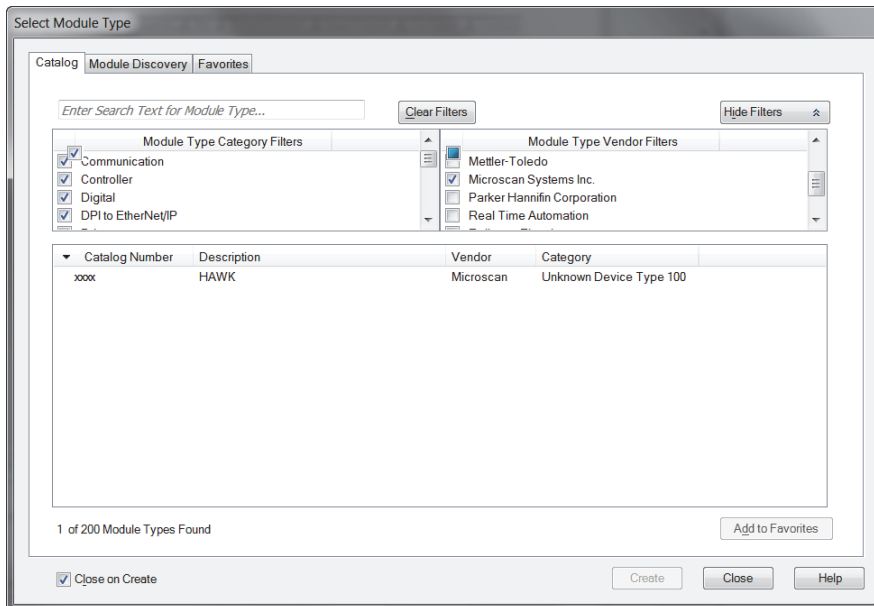
Click the required camera and click **Create**.
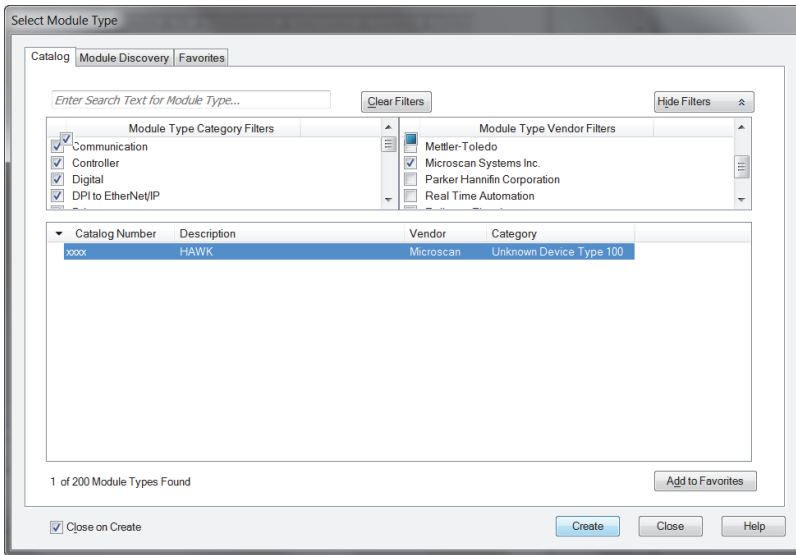
The **New Module** dialog is displayed. Type a unique name for this camera and its IP address.
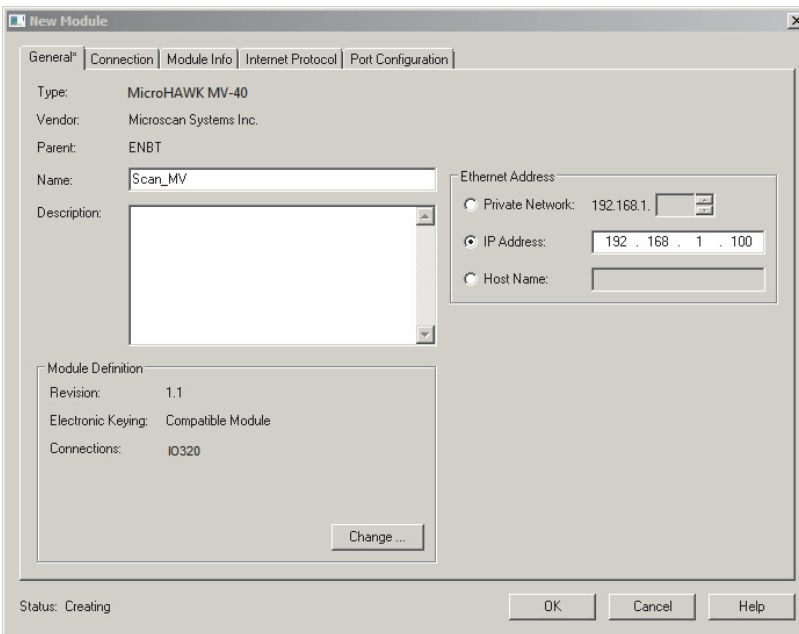
Click **OK**, verify the camera was added to the Ethernet network, then open the controller tags to verify that **:I** and **:O** tag sets were created.
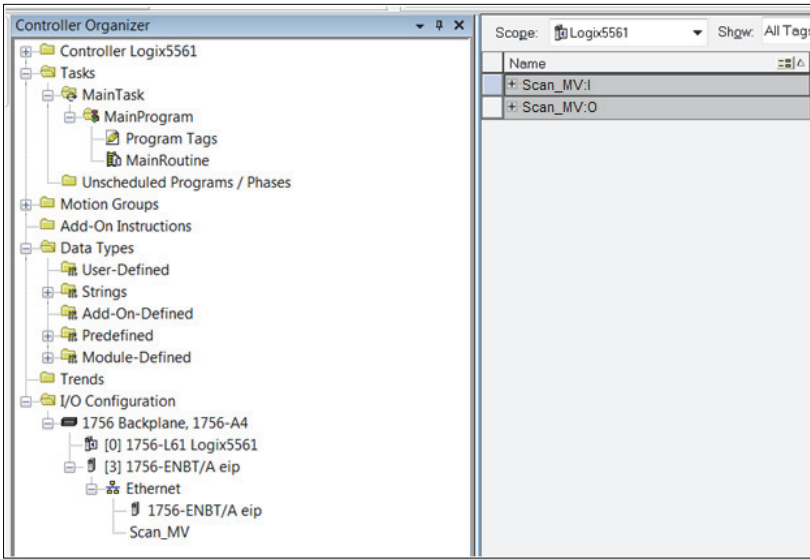
Open the **Main Routine**.
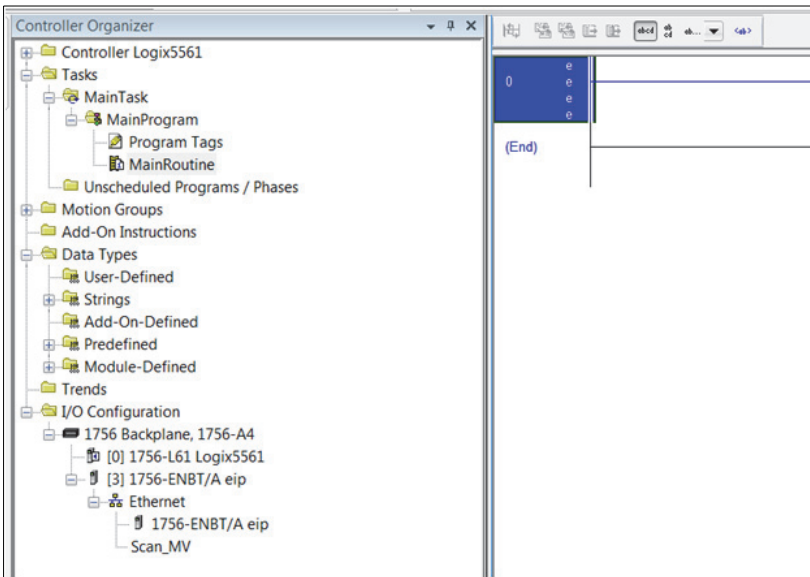
Right-click rung **0**, and select **Import Rungs**.
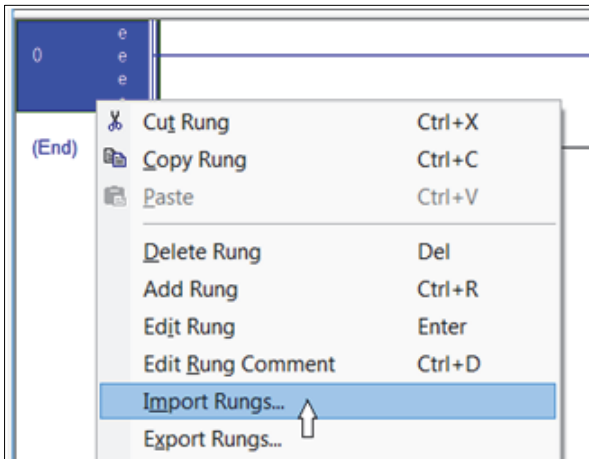
Navigate to the product example routine file and select **Import**. The default install directory is:

MicroHAWK: **C:\Omron\Vscape\Tutorials and Samples\MicroHAWK\EIP Demo**.

HAWK MV-4000: **C:\Omron\Vscape\Tutorials and Samples\HAWK MV-4000\EIP Demo.**

Note: If you are using a HAWK MV-4000 Smart Camera, you can use this example routine as well.
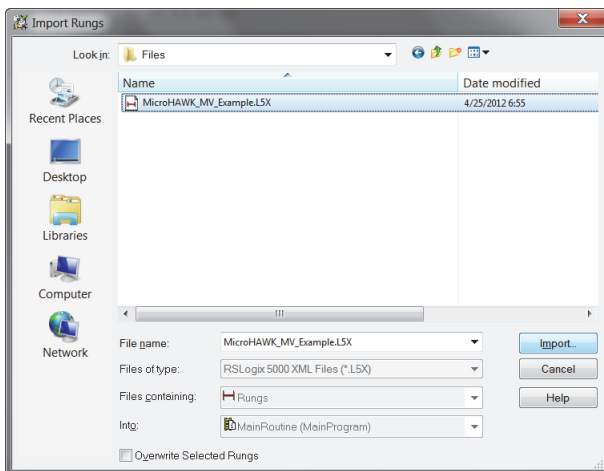
The **Import Configuration** dialog will be displayed.
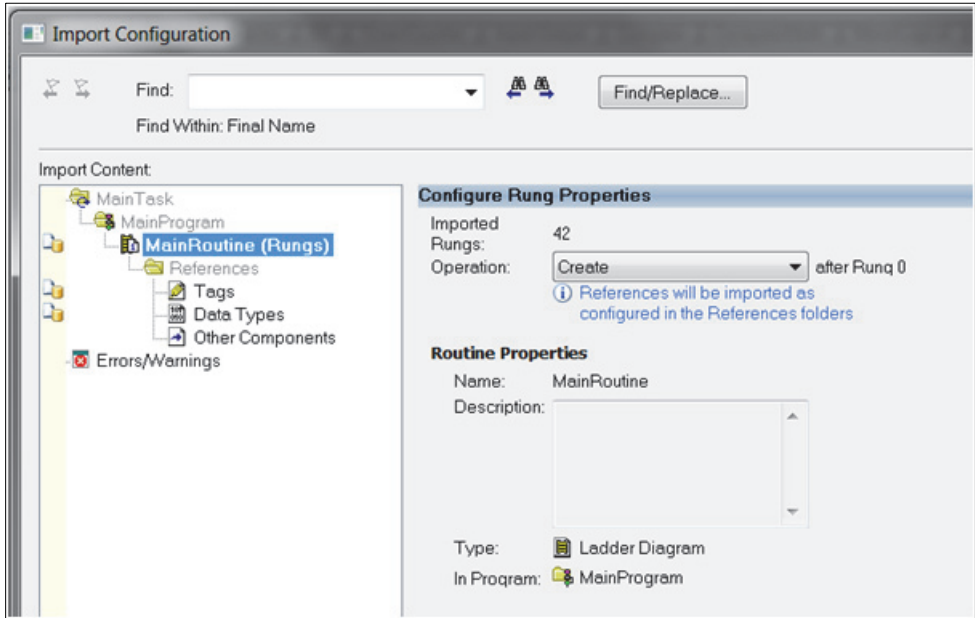
Select **Tags**.
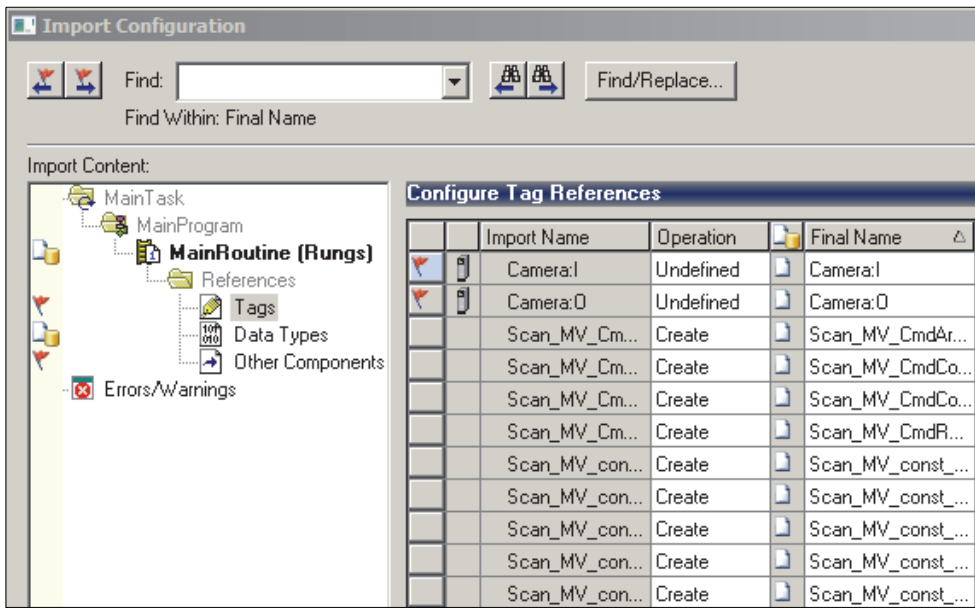
In the **Final Name** column, click **Camera:I**, then click the down arrow that appears on the right.
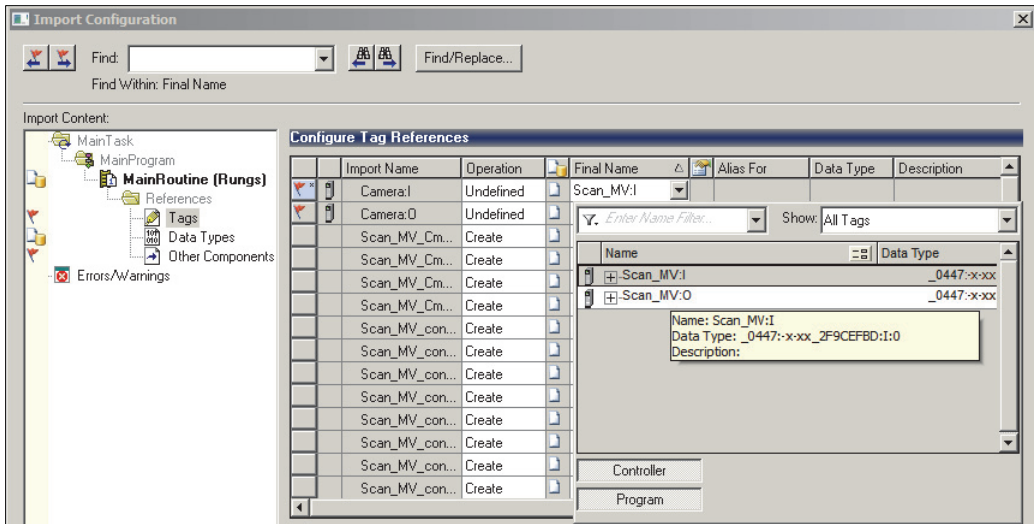
Double-click the camera name input tag assigned earlier.
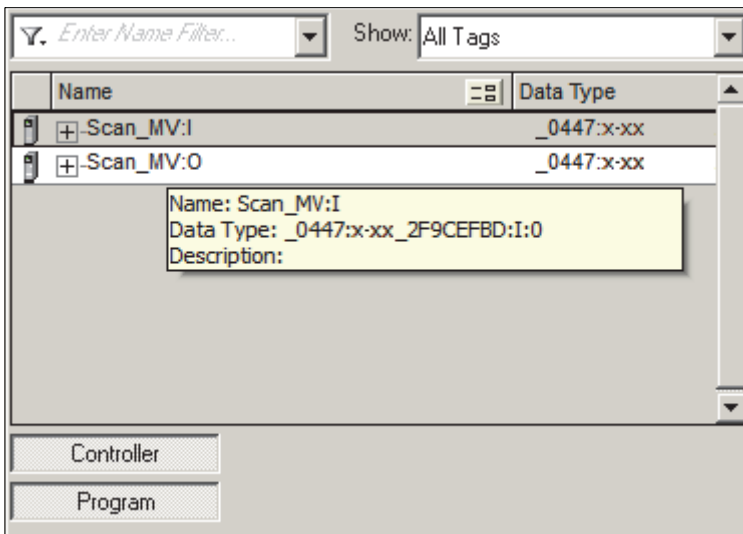
In the **Final Name** column, click **Camera:O**, then click the down arrow that appears on the right.
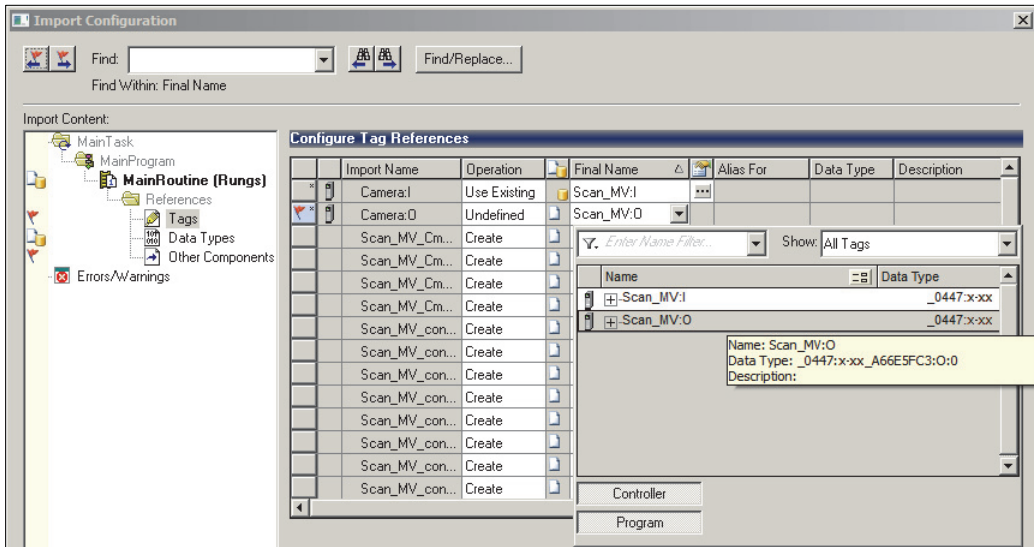
Double-click the camera name output tag assigned earlier.
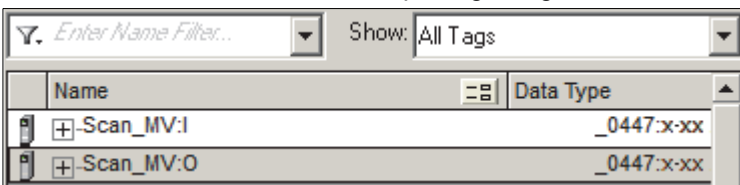
Click the **Other Components** icon in the tree view to select the **Component References**.
Select the camera in the **Final Name** column. Click **OK** to complete the import.
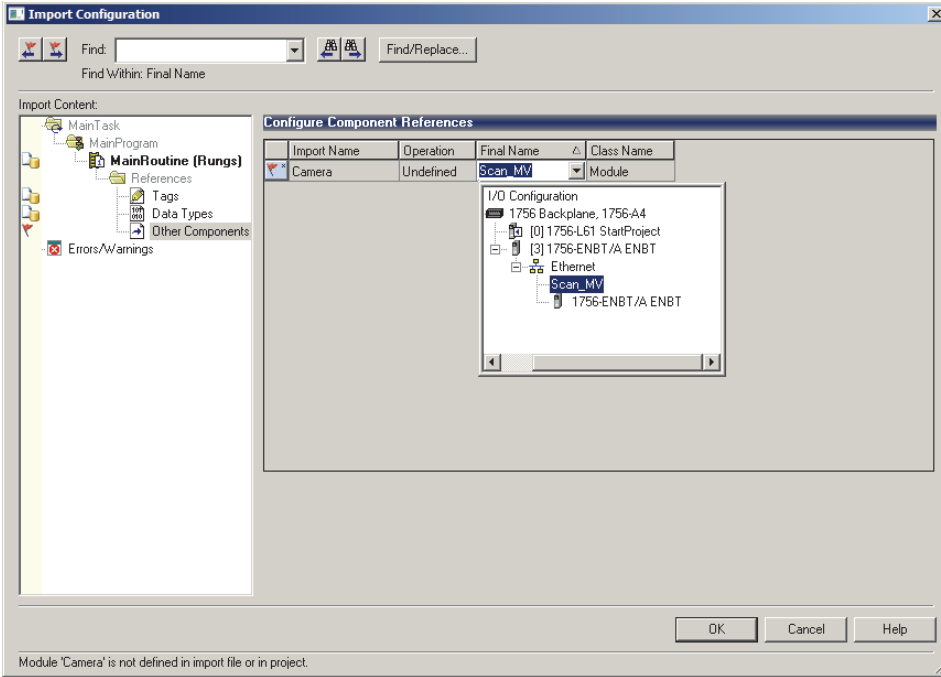
Click **OK** and the **Main Routine** and **User-Defined tags** will be populated.
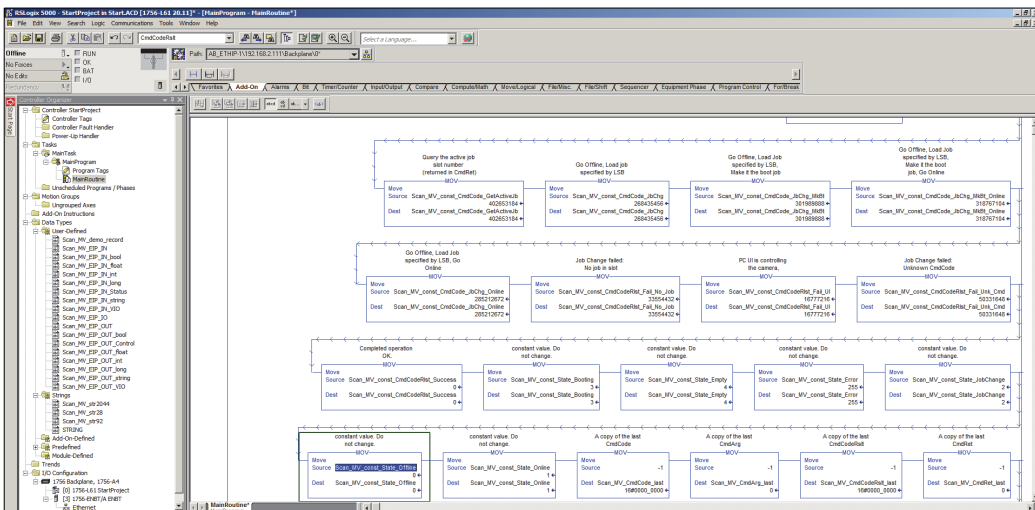
Delete any empty rungs (check rung **0**).
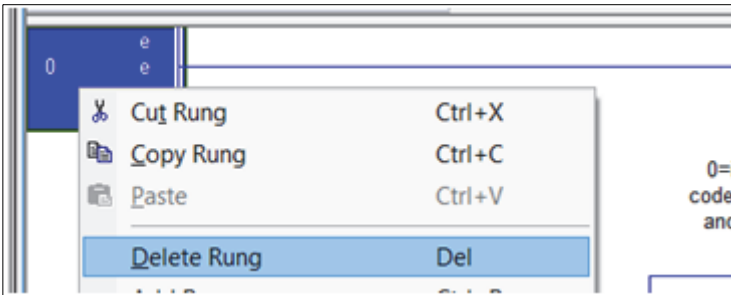
Download the project to the PLC.
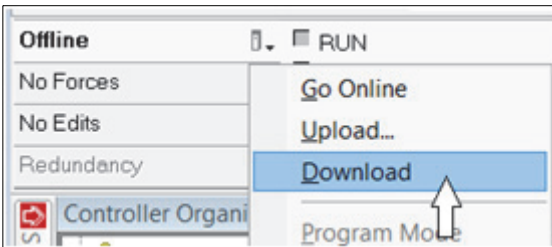
Note: Be sure the path to the PLC has been set in the project so that communications to the PLC can be established.
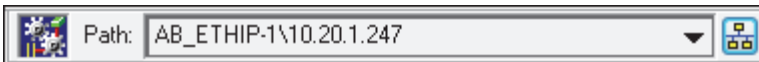
Put the PLC in **Run Mode**.
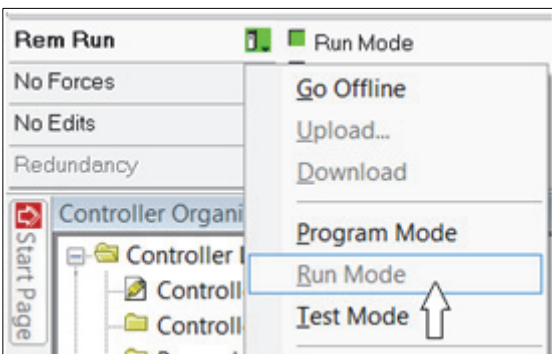
Open the **Program Tags** window and select **Monitor Tags**.

Name	Value
⊞-Scan_MV_const_State_JobChange	2
⊞-Scan_MV_const_State_Offline	0
⊞-Scan_MV_const_State_Online	1
⊞-Scan_MV_demo_blob	{...}
⊞-Scan_MV_demo_decode	{...}
⊞-Scan_MV_demo_InspStat	{...}
⊞-Scan_MV_demo_measure	{...}
⊞-Scan_MV_demo_mode	1
⊞-Scan_MV_dv_err_count	{...}
⊞-Scan_MV_dv_fall_count	{...}
⊞-Scan_MV_dv_rise_count	{...}
Scan_MV_ExeCmd_fall	0
Scan_MV_ExeCmd_rise	0
Scan_MV_ExeCmdAck_fall	0
Scan_MV_ExeCmdAck_rise	0
⊞-Scan_MV_IO_internal	{...}
⊞-Scan_MV_IO_user	{...}
⊞-Scan_MV_matchcode	'LABEL CHECK'
⊞-Scan_MV_ons_internal	-2080374528
⊞-Scan_MV_status_err_count	{...}
⊞-Scan_MV_trigger_count	{...}
⊞-Scan_MV_trigger_delay_timer	{...}
⊞-Scan_MV_trigger_err_count	{...}
⊞-Scan_MV_user_events	1

Program Tags - MainProgram

Scope: MainProgram   Show: All Tags

Monitor Tags / Edit Tags

4

Expand **Scan_MV_IO_user** so that the **Echo** in the **.IN.Status** and **.OUT.Control** structures is visible.

Name	Value
Scan_MV_IO_user.IN.Status.reserved15	0
Scan_MV_IO_user.IN.Status.Echo	0
Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000
Scan_MV_IO_user.IN.Status.CmdRet	0
Scan_MV_IO_user.IN.Status.reserved96_103	0
Scan_MV_IO_user.IN.Status.reserved104_111	0
Scan_MV_IO_user.IN.Status.State	1
Scan_MV_IO_user.IN.Status.reserved120_127	0
Scan_MV_IO_user.IN.VIO	{...}
Scan_MV_IO_user.IN.bool	{...}
Scan_MV_IO_user.IN.int	{...}
Scan_MV_IO_user.IN.long	{...}
Scan_MV_IO_user.IN.float	{...}
Scan_MV_IO_user.IN.string	{...}
Scan_MV_IO_user.OUT	{...}
Scan_MV_IO_user.OUT.Control	{...}
Scan_MV_IO_user.OUT.Control.GoOnline	0
Scan_MV_IO_user.OUT.Control.GoOffline	0
Scan_MV_IO_user.OUT.Control.reserved2	0
Scan_MV_IO_user.OUT.Control.reserved3	0
Scan_MV_IO_user.OUT.Control.ResetError	0
Scan_MV_IO_user.OUT.Control.ResetCount	0
Scan_MV_IO_user.OUT.Control.reserved6	0
Scan_MV_IO_user.OUT.Control.ExeCmd	0
Scan_MV_IO_user.OUT.Control.Trigger	0
Scan_MV_IO_user.OUT.Control.reserved9	0
Scan_MV_IO_user.OUT.Control.reserved10	0
Scan_MV_IO_user.OUT.Control.ResetDataValid	0
Scan_MV_IO_user.OUT.Control.reserved12	0
Scan_MV_IO_user.OUT.Control.reserved13	0
Scan_MV_IO_user.OUT.Control.reserved14	0
Scan_MV_IO_user.OUT.Control.reserved15	0
Scan_MV_IO_user.OUT.Control.Echo	0
Scan_MV_IO_user.OUT.Control.CmdCode	16#0000_0000
Scan_MV_IO_user.OUT.Control.CmdArg	0

Change **.OUT.Control.Echo** to non-zero.

—Scan_MV_IO_user.OUT.Control.reserved15		0
⊞-Scan_MV_IO_user.OUT.Control.Echo	▼	4321
⊞-Scan_MV_IO_user.OUT.Control.CmdCode		16#0000_0000

Verify that **Scan_MV_IO_user.IO.IN.Status.Echo** is the same value as the **.OUT.Control.Echo**.

—Scan_MV_IO_user.IN.Status.reserved15	0
⊞-Scan_MV_IO_user.IN.Status.Echo	4321
⊞-Scan_MV_IO_user.IN.Status.CmdCodeRslt	16#0000_0000

This confirms that the PLC and camera have successful two-way communication.

The demo code expects a demo vision job to be loaded on the camera, which populates the following input tags (camera to PLC) with vision tool results:

- .IN.bool.bool1, bool2, and bool3

- .IN.long.long1

- .IN.float.float1

- .IN.string.string1

The demo code will operate the **Control** and **Status** signals of the camera regardless of the vision job that is loaded. For a more detailed overview of the demo code and vision job, see **Allen-Bradley PLC Setup via Generic Ethernet Module for EtherNet/IP Operation**.

To send a trigger to the camera, scroll to **Scan_MV_IO_user.Control.Trigger**.

—Scan_MV_IO_user.OUT.Control.ExeCmd	0
—Scan_MV_IO_user.OUT.Control.Trigger	0
—Scan_MV_IO_user.OUT.Control.reserved9	0

Set the Trigger to **1**. This causes the demo code to trigger the camera, process the new inspection data, record the results in the **Scan_MV_demo_xxxx** tags, and clear the **DataValid** status signal.

The **Trigger** control changes to **0** when the camera is triggered. All processing is done when the counter **Scan_MV_dv_fall_count** increments, along with the pass/fail counters in the **Scan_MV_demo_xxxx** tags.

Name	=🔲 △	Value ←
⊟-Scan_MV_demo_blob		{...}
⊞-Scan_MV_demo_blob.pass_count		{...}
⊟-Scan_MV_demo_blob.fail_count		{...}
⊞-Scan_MV_demo_blob.fail_count.PRE		0
⊞-Scan_MV_demo_blob.fail_count.ACC		30
-Scan_MV_demo_blob.fail_count.CU		0
-Scan_MV_demo_blob.fail_count.CD		0
-Scan_MV_demo_blob.fail_count.DN		1
-Scan_MV_demo_blob.fail_count.OV		0
-Scan_MV_demo_blob.fail_count.UN		0
-Scan_MV_demo_blob.bool		0
⊞-Scan_MV_demo_blob.long		6
⊞-Scan_MV_demo_blob.long_max		6
⊞-Scan_MV_demo_blob.long_min		4
-Scan_MV_demo_blob.float		0.0
-Scan_MV_demo_blob.float_min		0.0
-Scan_MV_demo_blob.float_max		0.0
⊞-Scan_MV_demo_blob.string		''
⊞-Scan_MV_demo_decode		{...}
⊞-Scan_MV_demo_InspStat		{...}
⊞-Scan_MV_demo_measure		{...}
⊞-Scan_MV_demo_mode		1
⊞-Scan_MV_dv_err_count		{...}
⊟-Scan_MV_dv_fall_count		{...}
⊞-Scan_MV_dv_fall_count.PRE		0
⊞-Scan_MV_dv_fall_count.ACC	▼[	59]
-Scan_MV_dv_fall_count.CU		0

5

Allen-Bradley PLC Setup via Generic Ethernet Module for EtherNet/IP Operation

This section describes how to use the Generic Ethernet Module to set up an Allen-Bradley PLC for EtherNet/IP operation.

Notes:

- The camera communications protocol must be enabled for EtherNet/IP before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

- AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate EtherNet/IP communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

Integrating the Camera into a PLC Environment

This section assumes you are using an Allen Bradley PLC with Rockwell RSLogix 5000 v16 or newer. RSLogix v19 and v20 may look slightly different than the screen shots shown, but the integration process is similar.

Create the **I/O Configuration** for the base system, including the system's Ethernet interface.
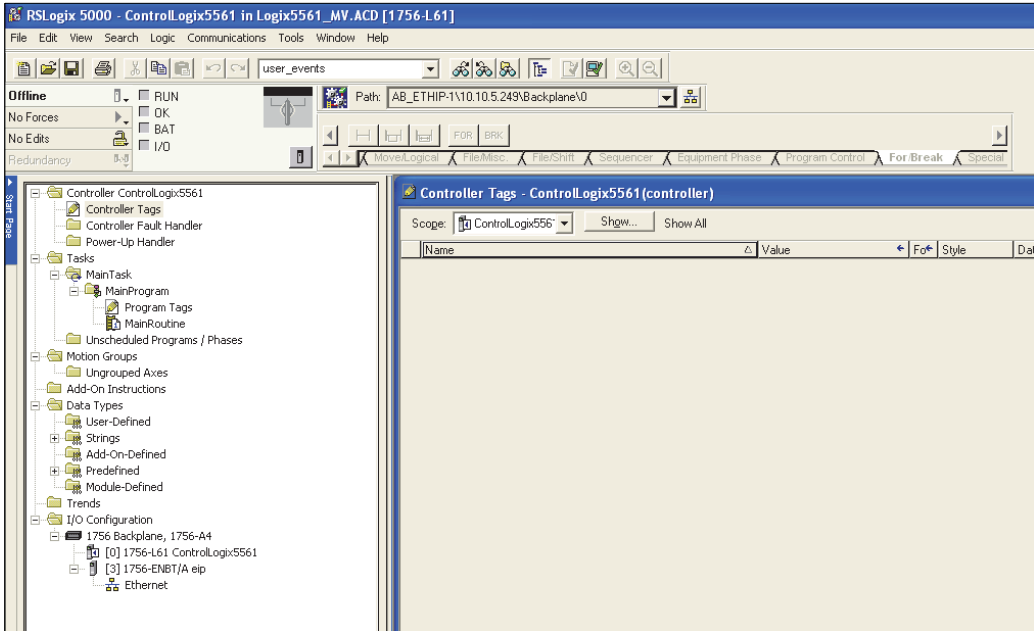
Add the camera by right-clicking the Ethernet interface and selecting **New Module**.
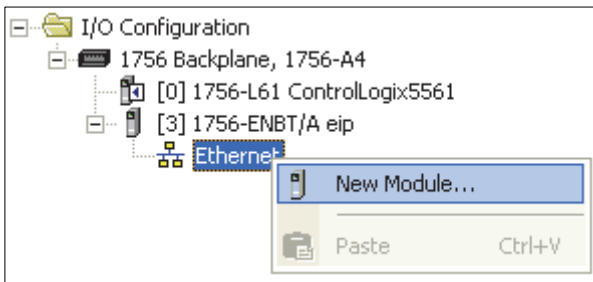
Select **ETHERNET-MODULE – Generic Ethernet Module**, and click **OK**.
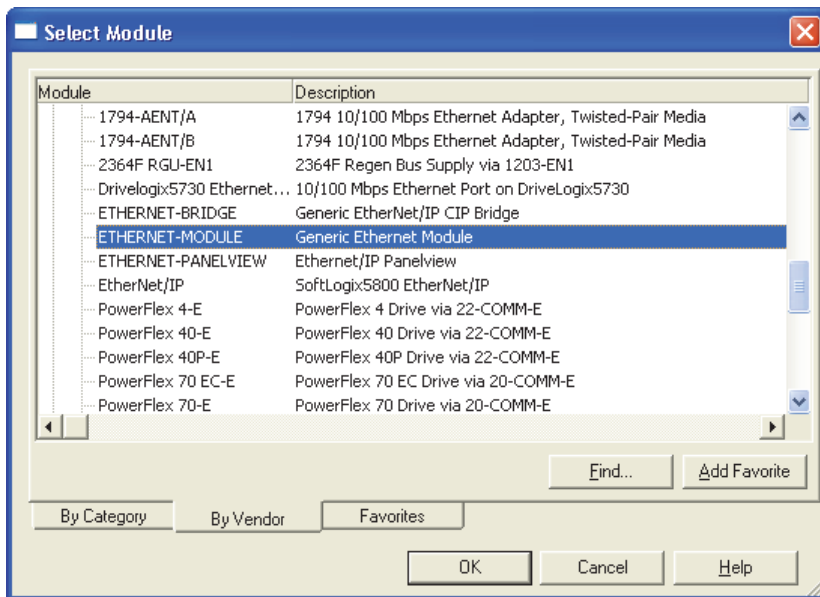
Configure the following fields:

Name: A useful name to rememberfor the camera. The example here is **Scan_MV**.

IP Address: The IP Address of the camera

Comm Format: "Data – DINT"

Input, Assembly Instance: 102

Input, Size: 80

Output, Assembly Instance: 114

Output, Size: 80

Configuration, Assembly Instance: 1

Configuration, Size: 0 (none)

Click **OK** when done.

Example:

Configure the **Required Packet Interval (RPI)** and click **OK**.

8 ms is the minimum RPI for the F440-F. **16 ms** is the minimum RPI for the MV-40, F430-F, and F330-F. **4 ms** is the minimum RPI for the HAWK MV-4000.
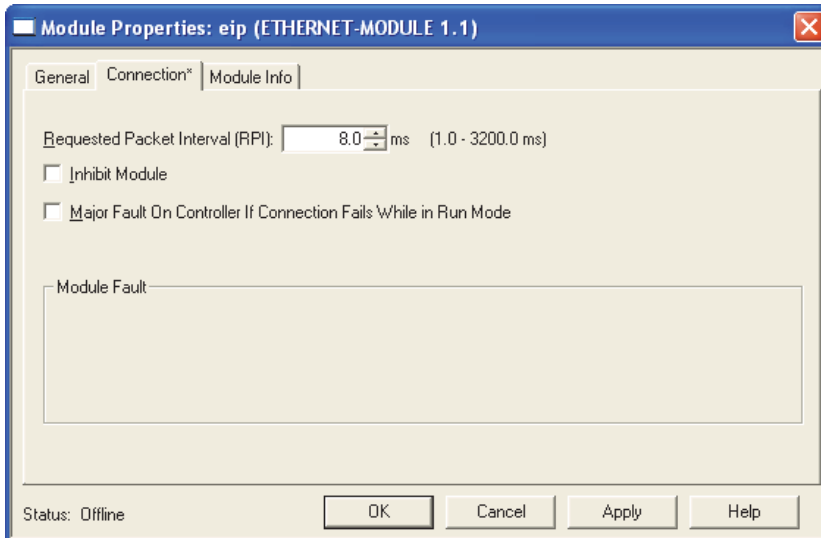
Verify that the camera was added to the Ethernet network, then open the controller tags to verify that **:I** and **:O** tag sets were created.

6
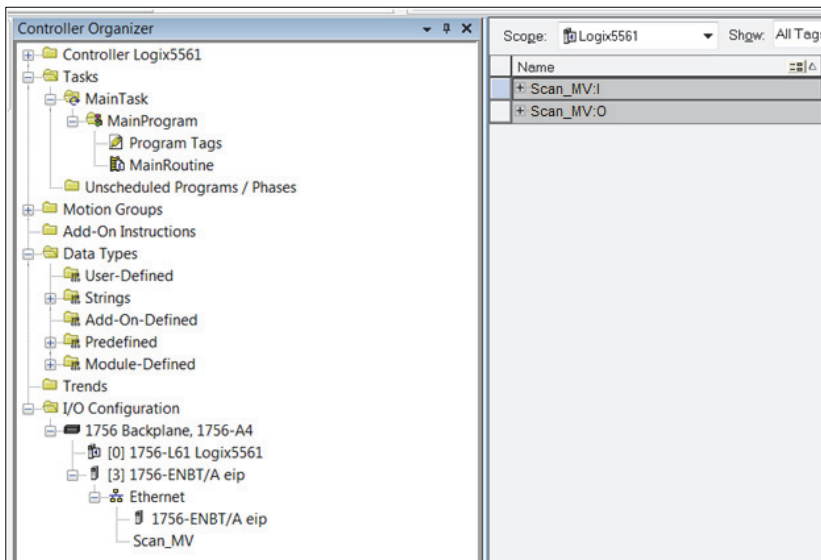
Omron PLC Setup for EtherNet/IP Operation

This section describes how to set up an Omron PLC for EtherNet/IP operation using an Omron Microscan Smart Camera and CX-One software.

Notes:

• The camera communications protocol must be enabled for EtherNet/IP before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

• AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate EtherNet/IP communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

Setting Up an Omron PLC

This section was created using the following Omron software and hardware:

- CX-Programmer version 9.43

- Network Configuration version 3.55

- PLC CJ2M CPU31

Start the CX-Programmer application and select menu item **File > New**. Enter the desired **Device Name**. Select the **Device Type** and then click **Settings** to the right of the Device Type menu.
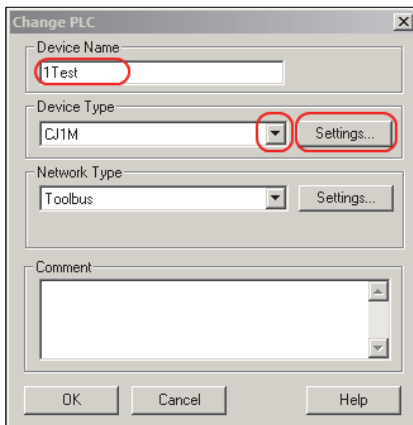
In the **Device Type Settings** dialog, select the correct **CPU Type** and click **OK**.
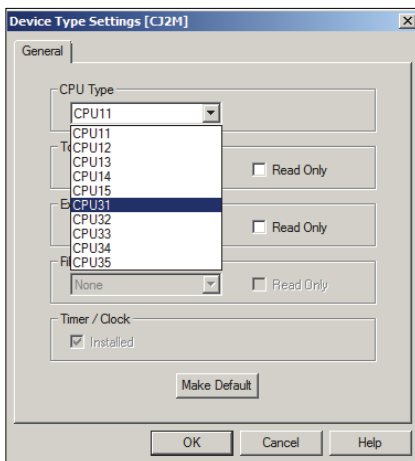
Select **USB** from the **Network Type** menu and click **OK**.
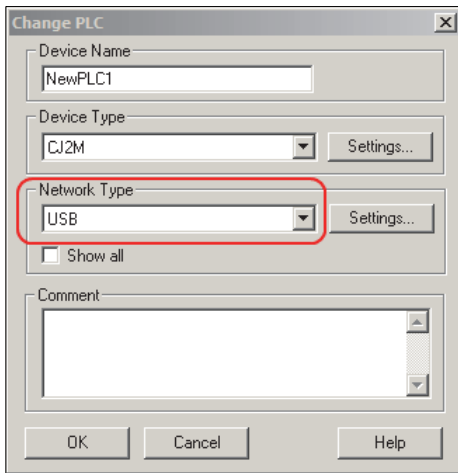
Connect to the PLC via USB connection. Select the menu item **PLC > Work Online** or click the online icon in the tool bar. When prompted, click **Yes** to complete the connection.
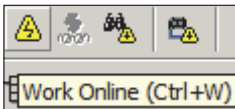
Once successfully connected, the background of the right pane will turn gray and the online icons in the ribbon will remain clicked.

To complete the next steps the PLC must be in **Program Mode**. Right-click the PLC node in the tree view in the left pane and select **Operating Mode > Program**.
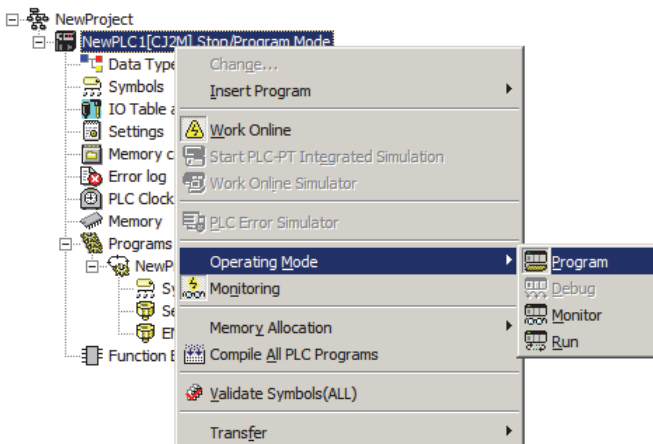
To register the I/O table, double-click **IO Table and Unit Setup**.
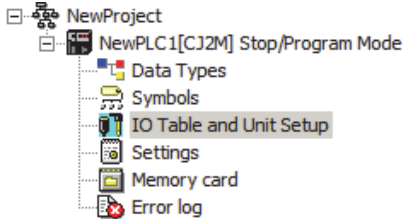
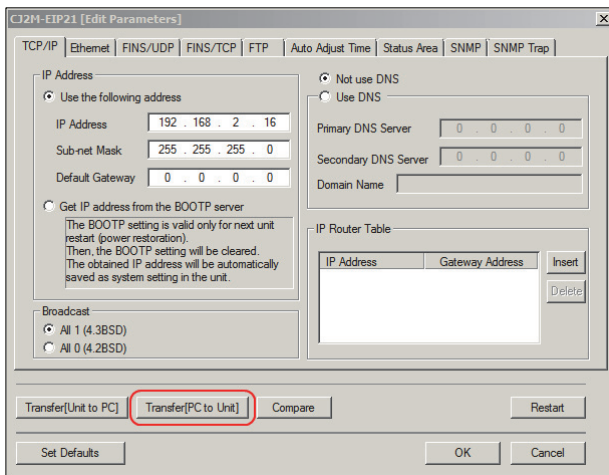
Select **Options > Create**.
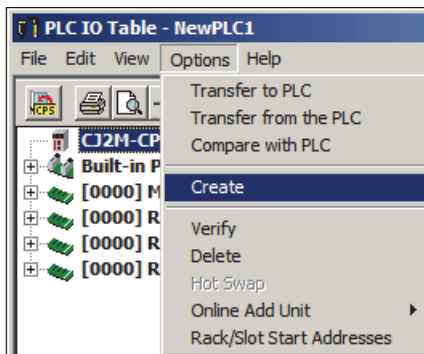
Click **Yes** at the I/O table creation prompt. Click **Yes** at the Initialize CPU bus settings prompt. Click **Transfer** at the transfer prompt. Click **OK** at the results prompt. The IO table will now be updated with the current PLC hardware settings.
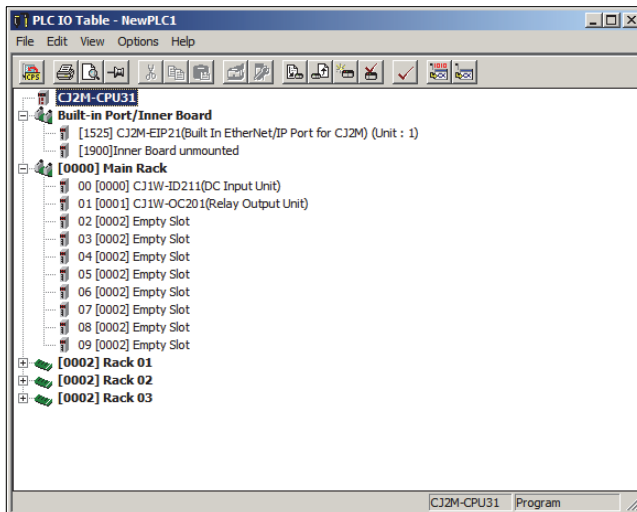
To edit the EtherNet/IP items and mapping, double-click the EtherNet/IP node.

- CJ2M-CPU31
- Built-in Port/Inner Board
  - [1525] CJ2M-EIP21(Built In EtherNet/IP Port for CJ2M) (Unit : 1)
  - [1900]Inner Board unmounted
- [0000] Main Rack
  - 00 [0000] CJ1W-ID211(DC Input Unit)
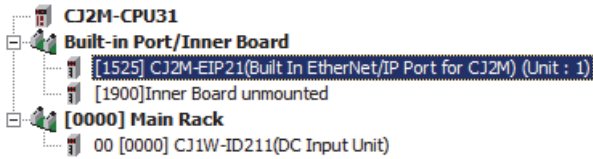
Input the desired IP settings and then click **Transfer [PC to Unit]**.

When prompted to transfer, click **Yes**. After the transfer prompt click **Close**. When prompted to restart the unit click **Yes**. Once the unit resets, click **OK** at the prompt. Close the IO edit dialog.

Physically power down the unit and adjust the rotary switches to match the last octet of the new IP address from above. Then power the unit back on.

Set the CX-Programmer application to offline by selecting **PLC > Work Offline** or the online icon in the ribbon.

In the CX-Programmer application, double-click the PLC node. In the Network Type menu, select **EtherNet/IP**. Click **Settings** to the right of Network Type and enter the PLC's new IP address. Click the **OK** buttons to close the dialogs.
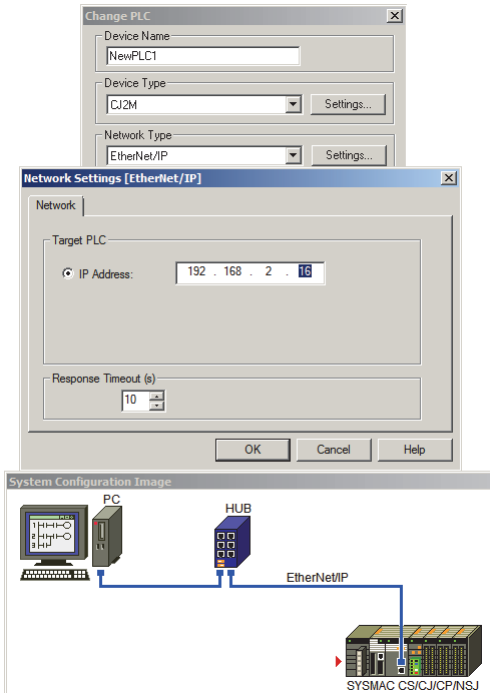

Set the CX-Programmer back to online. You will be prompted with a connection via EtherNet/IP. Click **Yes** to complete the connection.

Double-click the **IO Table and Unit Setup** node. Expand the **Built-In Port/Inner Board** node. Right-click and select **Start Special Application > Start with Settings Inherited**.
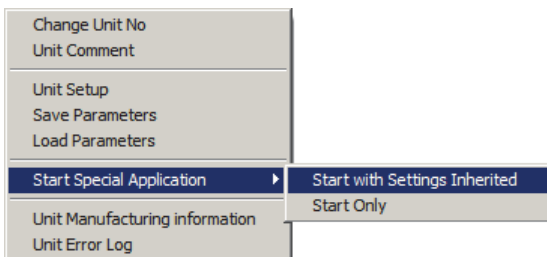
Select **Network Configurator** and click **OK**. Select port **TCP:2** and click **OK**. Then click
OK for **EtherNet/IP_1** connection.

To install the EDS file, select **EDS File > Install**. Navigate to the EDS folder
C:\Omron\Vscape\Firmware\eds\MicroHAWK or
C:\Omron\Vscape\Firmware\eds\HAWK. Select the correct file and click **Open** to load
the file. All other EDS files can be downloaded from the website.
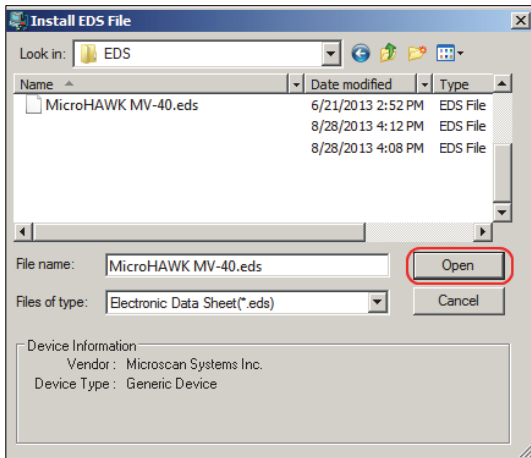
Expand the left tree view to open the **OMRON Corporation** files. Locate **CJ2M-EIP21** for
this example and drag it to the line in the right pane.
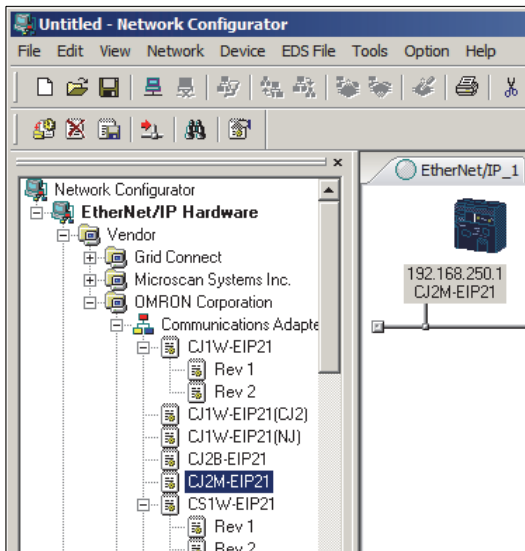
Expand the **Vendor Collection** node for the camera connected to the PLC and drag it to the line in the right pane.

Right-click near the newly-added icons and select **Change IP Address**. Enter the IP address for the PLC and the camera or reader attached and click **OK**.
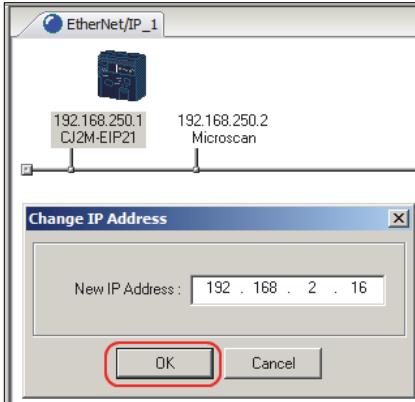

Double-click the PLC icon to edit the device parameters. This is where you will be linking and mapping the EtherNet/IP assembly data to the internal memory of the PLC. Select the **Tag Sets** tab. Select the **In - Consume** tab at the top. Click **Edit Tags** below. Then click **New** to edit/create a new tag. In this example we are naming this tag **300** for the peripheral memory linked to the input data. Select the size (**320 bytes**) for the entire input assembly. Click **Regist** then **Close** to continue.
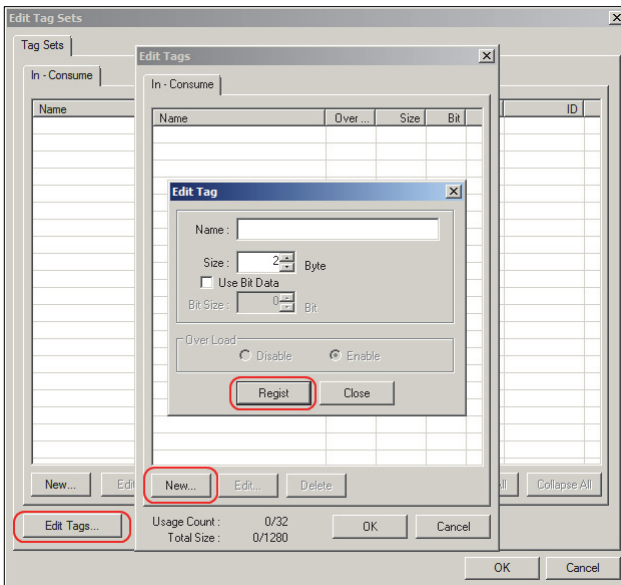
Select the **Out - Produce** tab and then click **New**. For the output assembly you are going to map to peripheral memory address **500** with **320 bytes**. Click **Regist** and then **Close**. Click **OK** on the **Edit Tags** dialog. When prompted to register new tags click **Yes**.
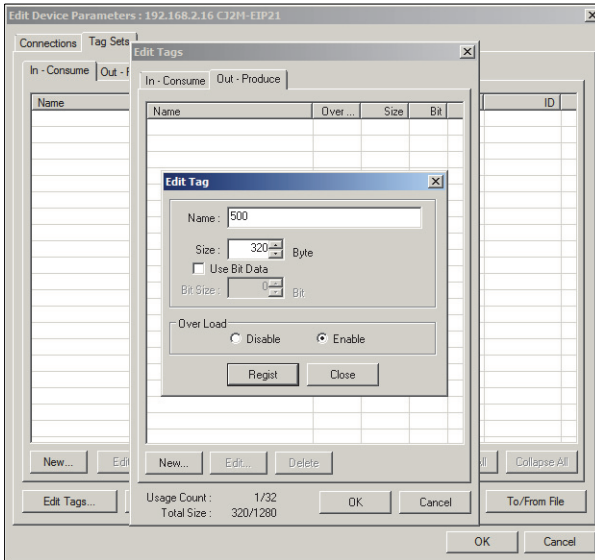
Click the **Connections** tab on the **Edit Device Parameters** dialog. Click the download button in the middle to register the device.
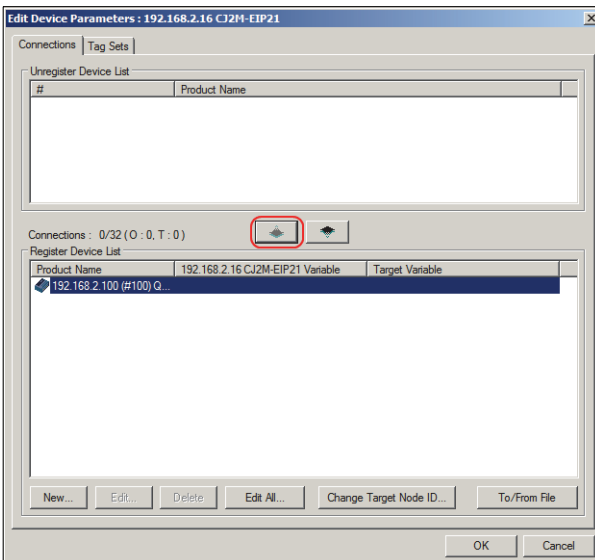
In the lower pane, double-click the PLC in the registered device list. This will open the linking dialog. If there are multiple connection types, they can be selected from the **Connection I/O Type** menu. In the **Originator Device** section, select the **Input Tag** then the **Output Tag**. Adjust the RPI if needed. When done, click **Regist** and then **Close**.
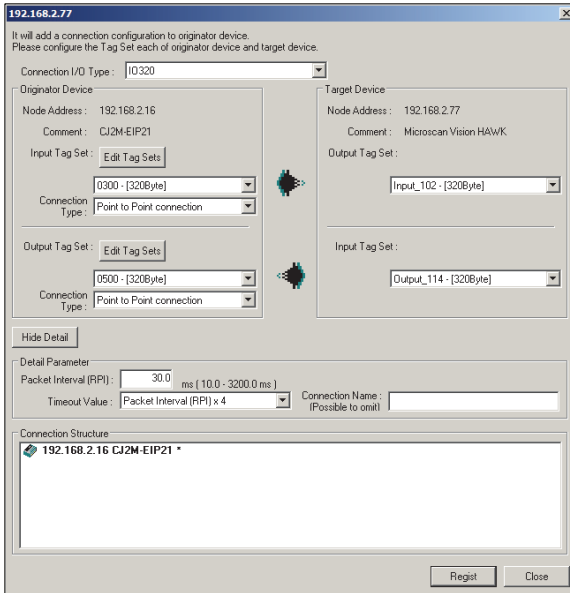

The registered device will now display the tags that are linked. Click **OK** to continue.
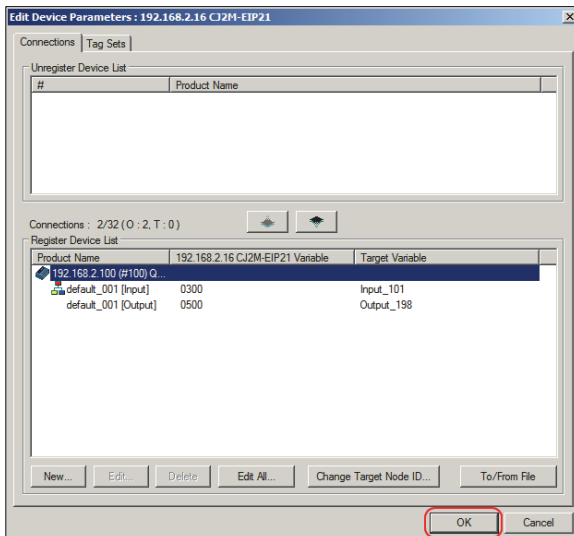
Download the new tags and links to the PLC by selecting **Device > Parameter > Download** or by clicking the download icon in the ribbon. Click **Yes** when prompted to download.

When the **List of Devices That Are Executing** dialog appears, select the PLC and click **Download after Changed to Program Mode**. When prompted to return the state, click **Yes** to continue.
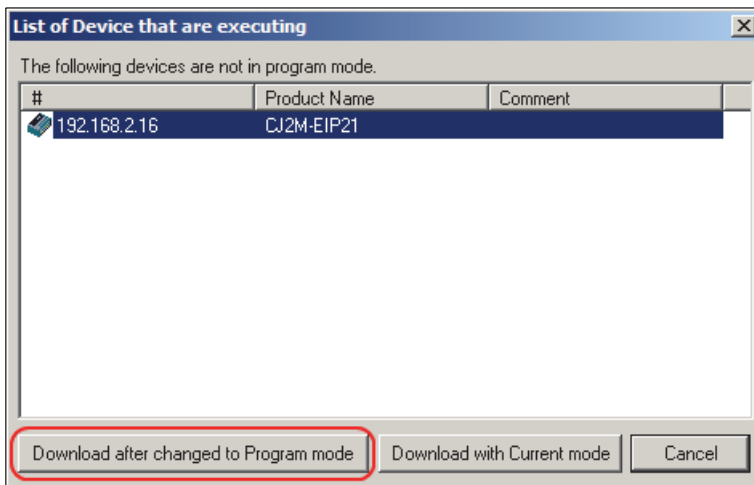
Select **Network > Check Connection** in the Network Configurator to ensure that there are no connection problems.

7

EtherNet/IP Operation Using Sysmac Studio and Omron NX/NJ Controller

This section provides information necessary for using an Omron Microscan Smart Camera in an EtherNet/IP environment.

Notes:

*   The camera communications protocol must be enabled for EtherNet/IP before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

*   AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate EtherNet/IP communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

Using AutoVISION to Set the Smart Camera Network Settings

1. Launch **AutoVISION** software and connect to the desired device.
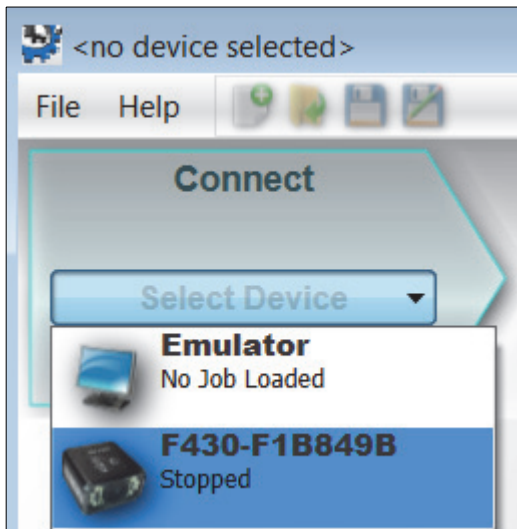
2. Click on the lock and then on **Details** for setting up the correct parameters.
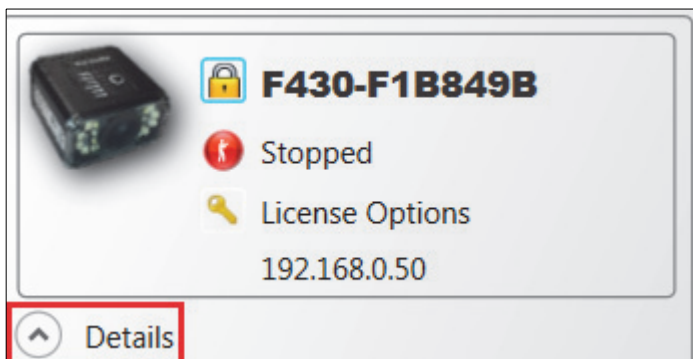
3. Click on **Modify** to edit the required parameters. Make sure the **IP Address** is in the desired range and the **Industrial Protocol** is set to **EtherNet/IP**. Click **Apply** when the correct parameters are set.

⌃ Details	
Model	F430-F WVGA
Category	SmartCamera
Version	9.2.0.3005
Memory	256 MB
Flash	2048 MB
IP Address	192.168.0.50
MAC Address	00:0B:43:1B:84:9B
Subnet Mask	255.255.0.0
Gateway	0.0.0.0
DHCP	Disabled
Number of serial TCP ports	4
Starting serial TCP Port	49211
Industrial Protocol	EtherNet/IP
Serial Port	RS232-1
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None
Auto Button	True

Modify

4. **Create a New Job**.
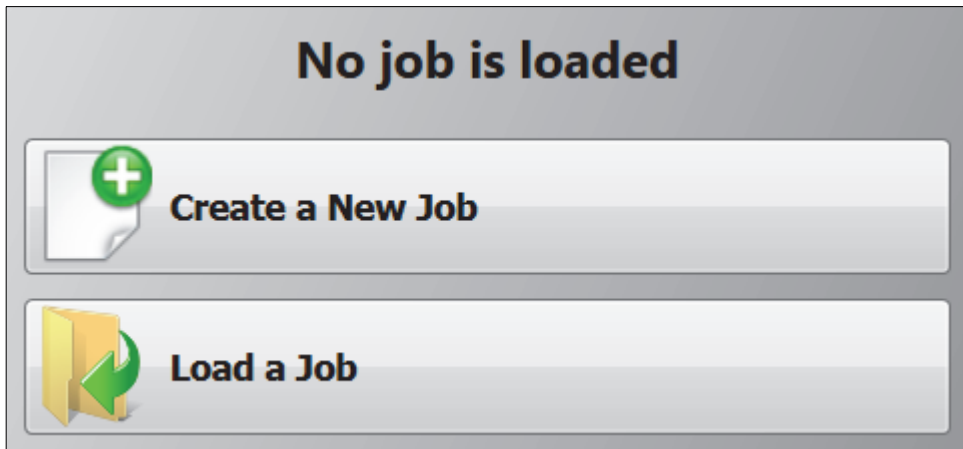
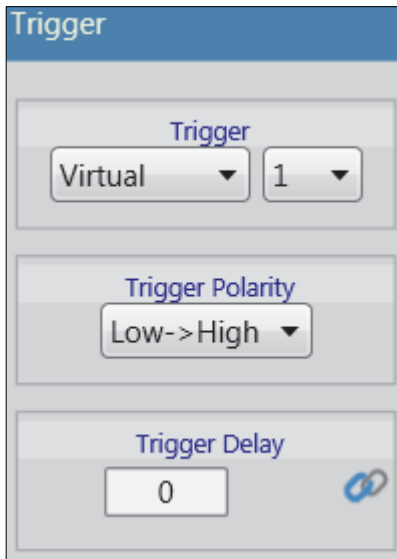**5.** Go to the **Edit** mode, select the **Camera** item and edit the **Trigger** settings. Select the type of trigger you wish to use – **Virtual** in this example.

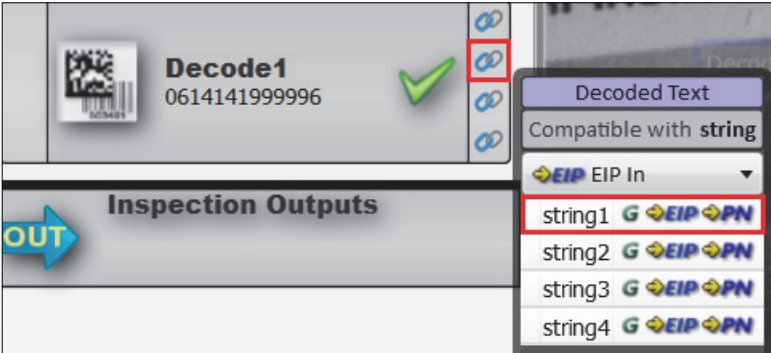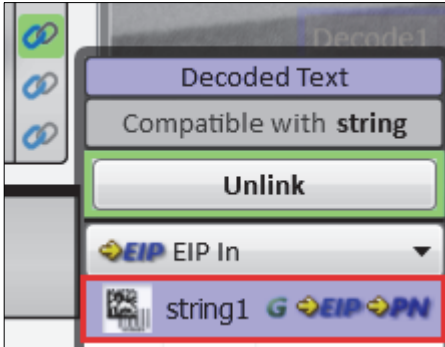6. Select the **Decode Tool** in the tool bar.

7. Click the blue camera icon to take a picture, and drag the symbol decode box fully around the code.

8. On the **Decode1** bar on the left, click the second link icon for decoded text.

9. Click **String 1** to link the **Decode** string to the EIP assembly **string1**.
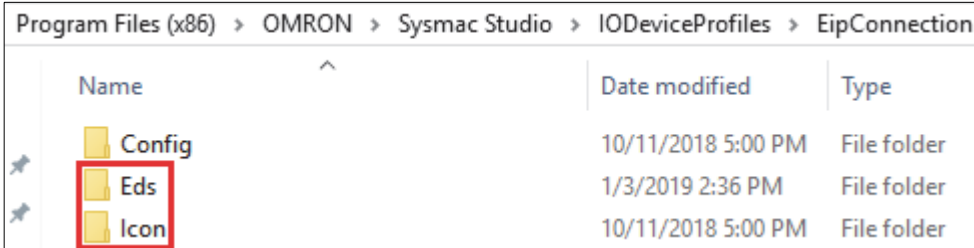
10. Go to **Run** mode so the device is waiting for a new **Trigger**.
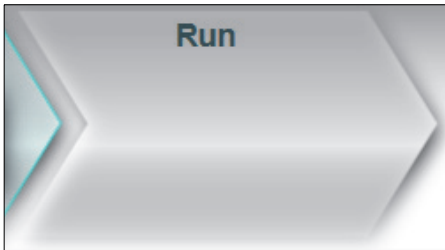
Setting Up the Controller

1. Obtain and install the F430-F EDS file so it can be used from Sysmac Studio. Copy the EDS and ICON files to **C:\Program Files(x86)\OMRON\Sysmac Studio\IODeviceProfiles\EipConnection**. Copy the **.eds** file into the **Eds** folder. Copy the **.ico** file into the **Icon** folder.

2. Create a new Sysmac Studio project.

3. Go to **Tools** and click **EtherNet/IP Connection Settings**.
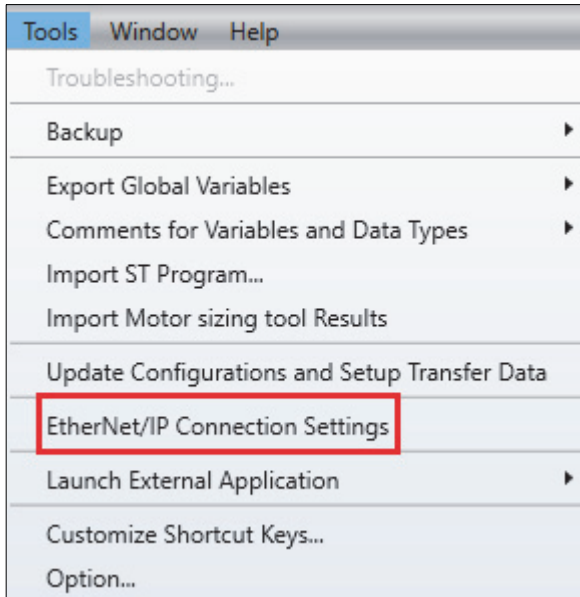
4. Double-click the controller **Device**. Be sure the PLC is on the same subnet as the reader.

5. On the **Toolbox** (on the right side) click on the **+** to add a **Target Device**. Enter the proper **Node Address** and select **Model Name** and **Revision**. Then click **Add** at the bottom.
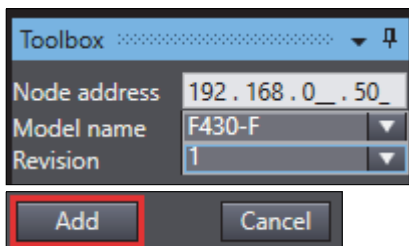
6. On the **Connection** tab, add a new connection by clicking on the **+**. Click on **Target Device** and select the newly added device.
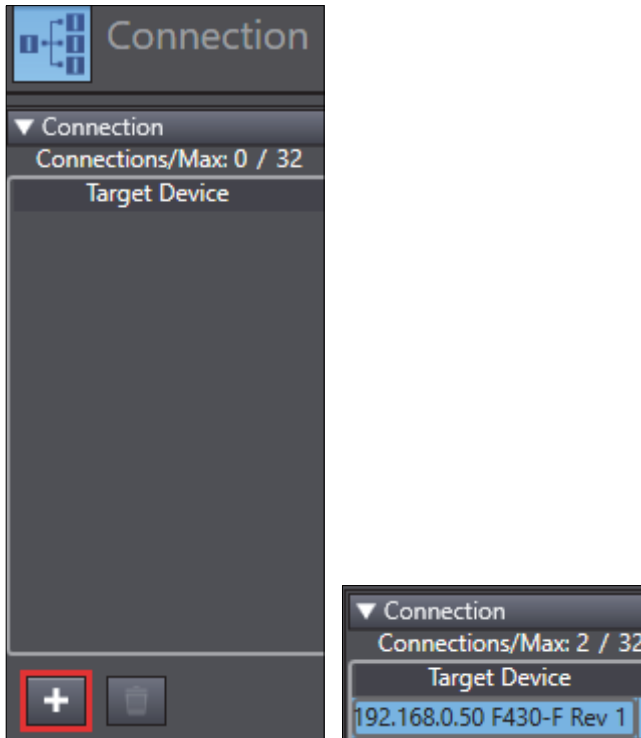
Setting the Global Variables

Set the global variables to use for the tag data links.

1. Double-click **Data Type** under **Programming > Data** in the **Multiview Explorer**.
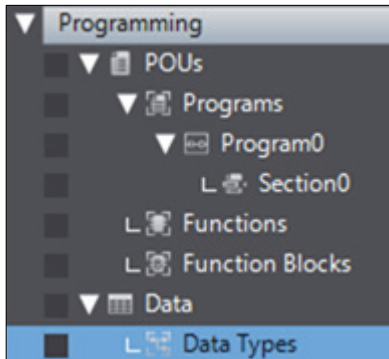
2. Follow the Industrial Protocol Manual to obtain the required Input and Output Assembly formats.

   Note: These may be created in an Excel spreadsheet and pasted into the Data Type to speed future creation.

3. Double-click **Global Variables** under **Programming > Data** in the **Multiview Explorer**.
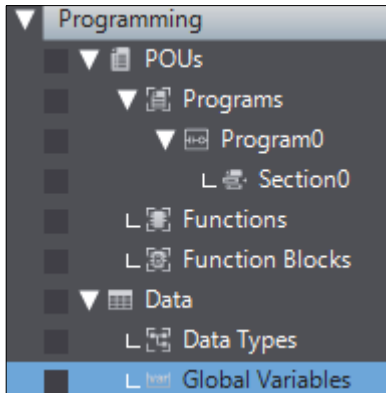
4. Add two new variable names here of the types created in step **2**. Be sure to set the **Network Publish** for **EIPInput** to input, and **EIPOutput** to output.

Name	Data Type
EIPInput	Input_Assembly_102
EIPOutput	Output_Assembly_114

Network Publish
Input
Output
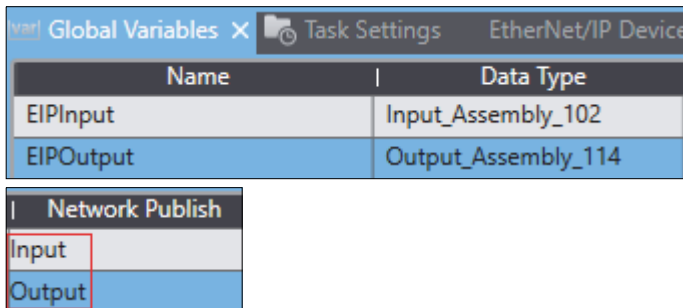
5. Double-click **Task Settings** under **Configurations and Setup** in the **Multiview Explorer**.
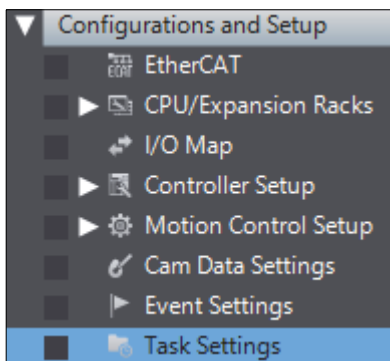
6. Under the **VAR** tab add the two recently created variables.
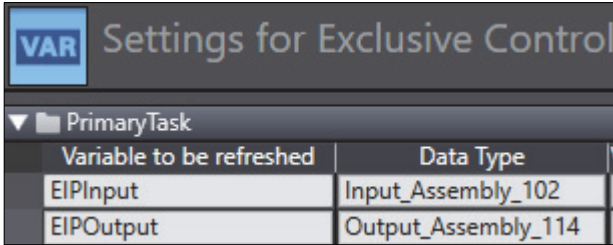
7. Go back to **Tools**, **EtherNet/IP Connection Settings**. Double-click the **Device**, then click on the **Tag Set** tab. It is now possible to register the newly-created variables. Click on **Registration All**.
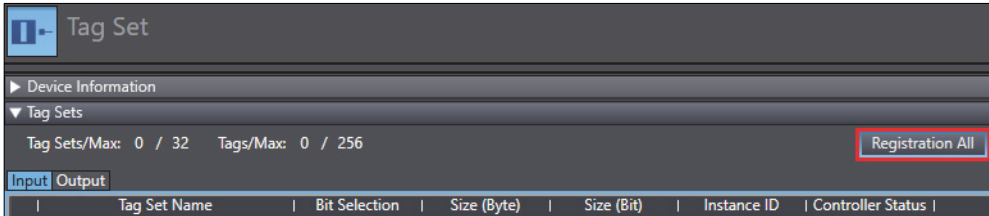
8. When the **Tag Set Registration Setting** window pops up, click **Register**.
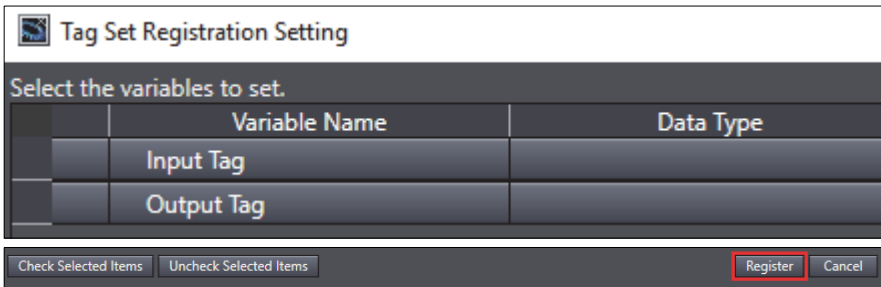
9. Under the **Connection** tab add the **Target Variable** and the **Originator Variable**.

Na	Connection I/O	Input/Out	Target Variable	Size [Byte]	Originator Variable	Size [Byte]
IO320		Input	102	320	EIPInput	320
		Output	114	320	EIPOutput	320

10. Go online and **Transfer to Controller**.
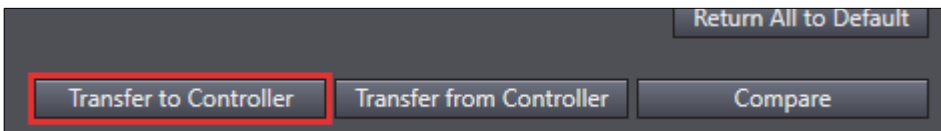
Checking EtherNet/IP Communications

Confirm that the EtherNet/IP tag data links are operating normally.

1. Go to the **Tools > EtherNet/IP Connection Settings**.
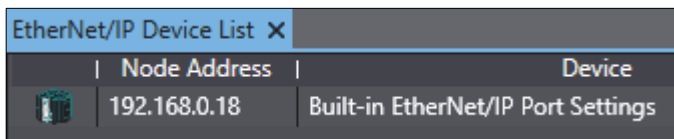
2. Right-click and select **Monitor**.
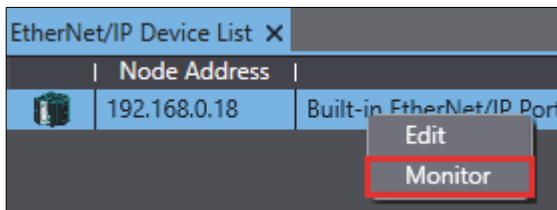
3. On the **Status** tab at the bottom, **Target Node Status** should be blue.
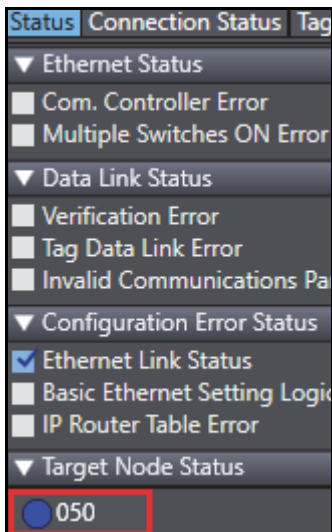
4. Click on the **Connection Status** tab. There should be a blue dot to the left of the controller.
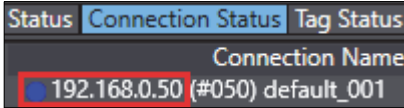
5. If either of the dots in step 3 or 4 are red, unplug the PLC and re-power it.

6. Click the **View** menu item and select **Watch Tab Page**. This will open a watch view area where you can view your assembly data.
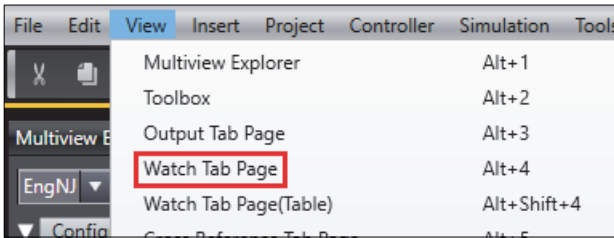
7. Add both **EIPInput** and **EIPOutput** Global Variables to the watch table.
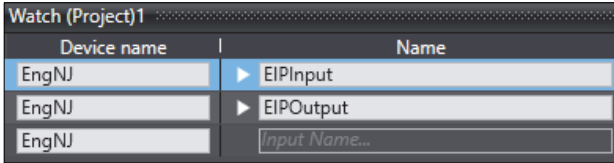
8. Expand the **EIPOutput** variable and scroll to **Trigger**. Set trigger to **TRUE**. You will see the camera trigger.
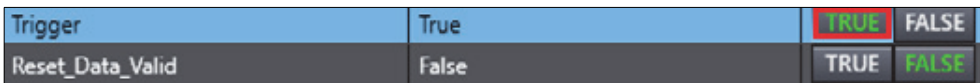
9. Expand the **EIPInput** variable and scroll down to **String 1**. You will see the decode string here.

10. Under **EIPOutput**, you must manually clear the trigger bit back to **FALSE**. You must also set the **Reset_Data_Valid** bit to **TRUE**, then back to **FALSE**, before you will be allowed to trigger again.
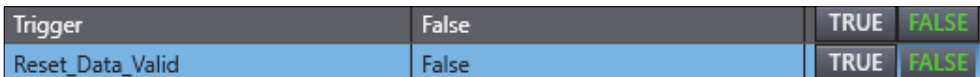
Using PROFINET I/O

This section provides information necessary for using an Omron Microscan Smart Camera in a PROFINET I/O environment.

Notes:

- The camera communications protocol must be enabled for PROFINET I/O before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

- AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate PROFINET I/O communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

Important: PROFINET I/O allows the IP address and subnet mask of the camera to be assigned automatically by the PLC. In order to allow this, the camera may not have an assigned IP address on reboot until the PLC is set to Run mode. During this time, the camera will not be visible on the network for AutoVISION or Visionscape FrontRunner.

Important: When the device is in PROFINET mode, the PLC will control the device's IP address settings, and setting the IP address via FrontRunner, AutoVISION, or the Device Discovery Utility will not be possible.

PROFINET Device Discovery Using Siemens TIA Portal

You can discover the PROFINET devices in TIA Portal by clicking on the search icon in the top menu bar.

It will open the following screen. Select the network interface you want under **PG/PC Interface** (**1**) and then click the Start search button (**2**).

This will list all PROFINET devices seen on the PN/IE network, as in the example below. You can see the device's name and IP settings in this screen.

PROFINET Device Discovery Using OMRON CX-Configurator-FDT

- Add your PNT21 device to your Omron CJ/NJ and configure it per the appropriate user guides.

- Run **CX-Configurator-FDT**.

- In the **Network View** pane on the right, click **MyNetwork > Add**.
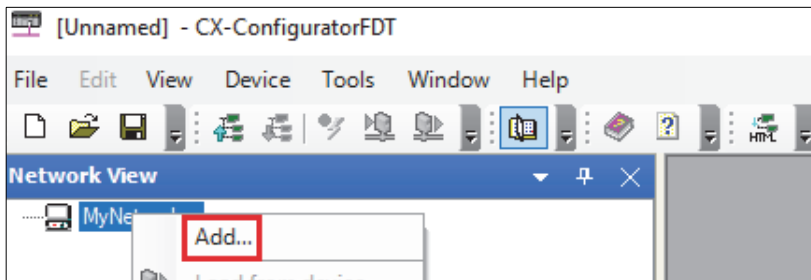
- Select **CJ1W-PNT21 PROFINET IO Controller**.

- The controller will now appear under **MyNetwork** as shown below.
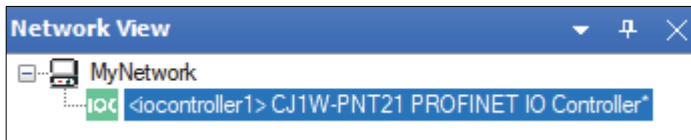
- Double-click **<iocontroller>** and the Controller Configuration screen will display as shown below.
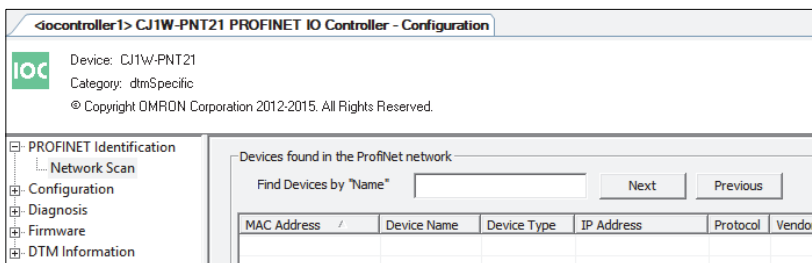
- Click on **Start Scan** at the bottom of the window.

Click "Start Scan" to begin searching for Devices in the ProfiNet network.

Scan Progress: [   ]   Start Scan

Click "Configure" to select the available Network Devices for Scanning.

Configure
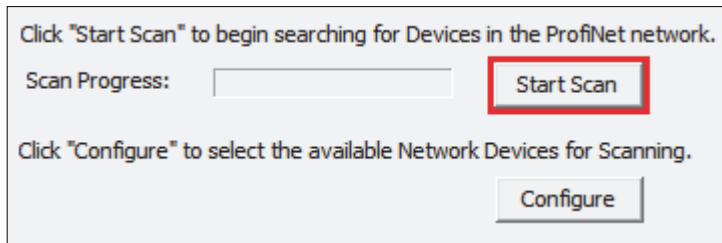
- A list of PROFINET devices will be displayed.

MAC Address	Device Name	Device Type	IP Address	Protocol	Vendor ID	Device ID	Device
00-20-FC-32-05-2E		Smart Camera	0.0.0.0	DCP	599	28676	Device
00-0C-29-1F-A1-C3	win-sn7tk0u…	SIMATIC-PC	192.168.0.25	DCP	42	514	-
00-0C-29-1F-A1-B9	win-sn7tk0u…	SIMATIC-PC	192.168.0.75	DCP	42	514	-
00-0B-43-17-5D-C1		Smart Camera	10.20.1.63	DCP	599	28674	Device

Default PROFINET Name and IP Address for Machine Vision Products

- The default PROFINET Name is not set and will display as "Accessible Device" in TIA Portal or an empty string in PROFINET Commander.

- The default IP address for products other than the HAWK MV-4000 is 192.168.188.2, with a subnet mask of 255.255.255.0.

- The HAWK MV-4000 IP address is based on the device MAC address. For example, a device with MAC 00:0B:43:01:02:03.

- The default IP is 192.168.02.03, using the 5th and 6th octet of the MAC as the last two IP digits.

PROFINET Device Naming and IP Address Configuration

PROFINET requires all devices to have a PROFINET name. This name can be, but is not required to be, the same as the device name. If a name is not set, or if it is set differently than what the PLC expects, a connection will not happen until all the names match.

Note: Many Antivirus / system security software platforms may block the DCP protocol, which will not allow you to connect or configure your device. You can temporarily turn off the firewall software briefly to see if it is blocking your DCP protocol, and make appropriate change as required.

Setting a device name can be done in one of two ways:

- Via the PLC;

- Via a PROFINET tool such as PROFINET Commander.

Setting the PROFINET Device Name and IP Via PLC Using Siemens TIA Portal Software

1. In the **Network View**, double-click the camera icon. This will open the **Device View**.

2. Click the camera icon then select the **Properties Tab**.
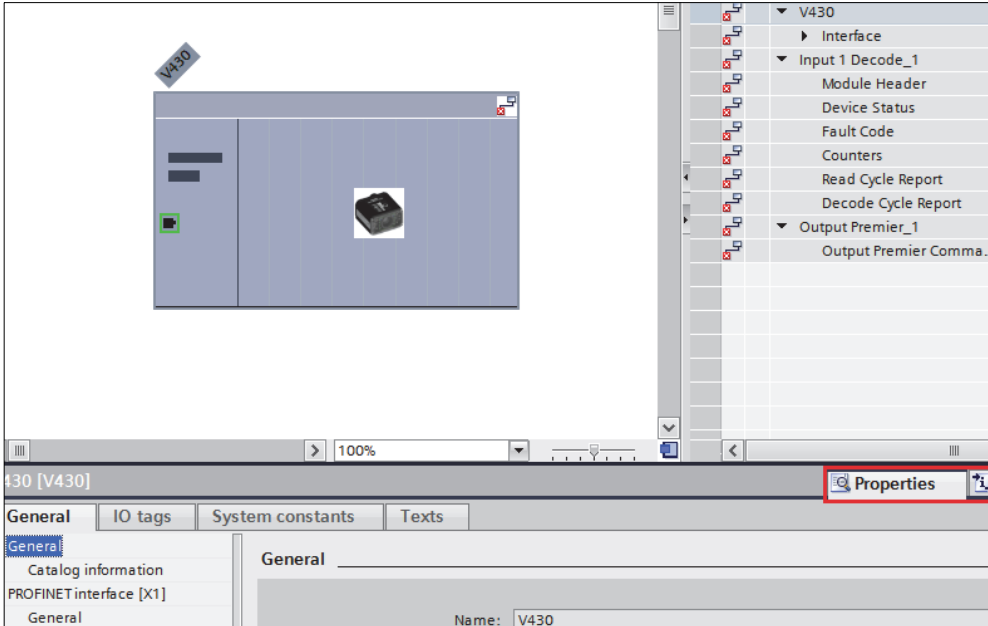
3. Select the **General Tab** and change the name of the camera in the **Name:** field to a unique name for the PROFINET network. There cannot be another device with the same name on the same network.
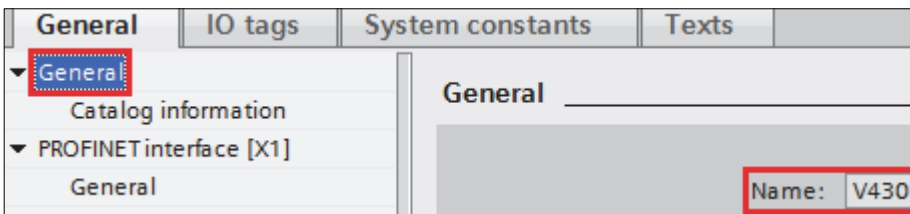
4. Select PROFINET interface **[X1] > Ethernet** address and scroll down to **IP Protocol**. Set the **IP Address** to the desired address for the network. In this example, it is **192.168.0.50**.
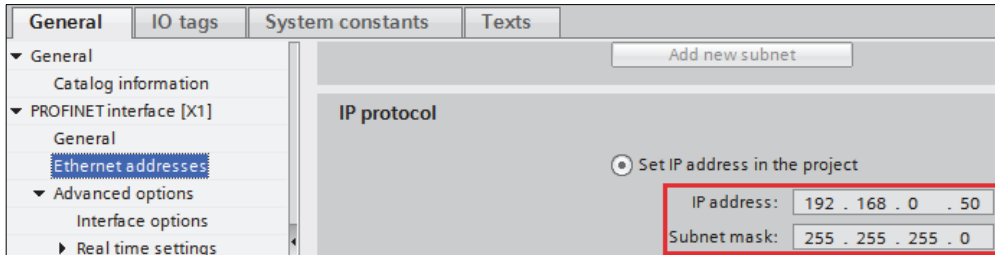
5. Save the project by clicking the **Save Project** icon.
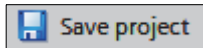
Setting the PROFINET Device Name and IP Via PROFINET Commander

1. Run the **PROFINET Commander** program.

2. Double-click on the **Browse** icon.
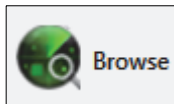
3. Select the **Network Interface** (**1**) in the selector on the right, then click **DCP Identify All** (**2**) on the left.
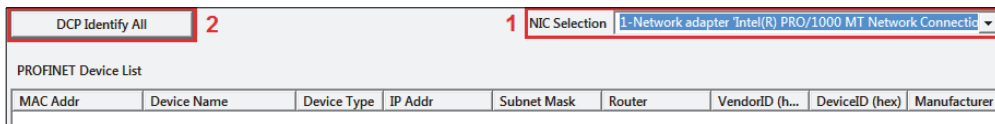
4. The program will list all PROFINET devices discovered on the network. An example is shown below.

MAC Addr	Device Name	Device Type	IP Addr	Subnet Mask	Router	VendorID (h...	DeviceID (hex)	Manufacturer
20:87:56:97:17:43	b	SCALANCE...	192.168.0.99	255.255.255.0	192.168.0.99	002a	0a01	SIEMENS AG
00:0B:43:39:39:44	v430	Ident Syste...	192.168.0.50	255.255.255.0	192.168.0.50	0257	3411	Microscan
28:63:36:48:AD:40	a	S7-1500	192.168.0.100	255.255.255.0	192.168.0.100	002a	0114	SIEMENS AG

5. Select the device you wish to change.

MAC Addr	Device Name	Device Type	IP Addr	Subnet Mask	Router	VendorID (h...	DeviceID (hex)	Manufacturer
20:87:56:97:17:43	b	SCALANCE...	192.168.0.99	255.255.255.0	192.168.0.99	002a	0a01	SIEMENS AG
00:0B:43:39:39:44	v430	Ident Syste...	192.168.0.50	255.255.255.0	192.168.0.50	0257	3411	Microscan
28:63:36:48:AD:40	a	S7-1500	192.168.0.100	255.255.255.0	192.168.0.100	002a	0114	SIEMENS AG

6. Enter the name you wish to assign in the **DCP Set Name** box and select whether you want this setting to be temporary or permanent. Be aware that this name can be any simple text string you want, as long as the camera and the PLC project are not using the same name for the same device.

DCP Set Name	myname	☐ Write Device Name Temporary

7. Click the **DCP Set Name** button.

8. Enter the IP Address you want the device to have into the **DCP Set Params** block and select whether you want this IP set to be temporary or permanent.

DCP Set IP Params	192 . 168 . 0 . 1	IP Address	☐ Write IP Settings Temporary
	255 . 255 . 255 . 0	Subnet Mask	
	0 . 0 . 0 . 0	Router	

DCP Get Info	Start LED Flash	DCP Reset to Factory

9. Press the **DCP Set IP Params** button.

10. At the top of the screen click the **DCP Identify All** button again.

11. You will see that the data you set is now listed under the device you modified.

Setting Device Name and IP via OMRON CX-Configurator-FDT

1. Follow instructions above for device discovery using **OMRON CX-Configurator-FDT**.

2. Select a device from the **Device List**.

MAC Address	Device ...	Device Type	IP Address	Protocol	Vendor ID	Device ID	Device I
❗ 00-0C-29-1F-A1-C3	win-sn7tk0u...	SIMATIC-PC	192.168.0.25	DCP	42	514	-
❗ 00-0C-29-1F-A1-B9	win-sn7tk0u...	SIMATIC-PC	192.168.0.75	DCP	42	514	-
❗ 00-20-FC-32-05-2E		Smart Camera	0.0.0.0	DCP	599	28676	Device
❗ 00-0B-43-17-5D-C1		Smart Camera	10.20.1.63	DCP	599	28674	Device

3. Right-click the selection and select **Edit Device Name** or **Edit Device IP Address**.
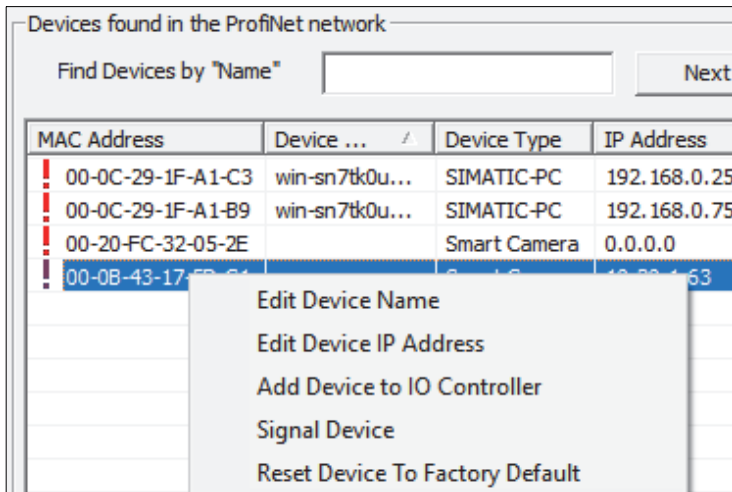
4. **Edit Device Name**. Set **Station Name** in field, Select **Temporary Setting** if required, then click **OK**.
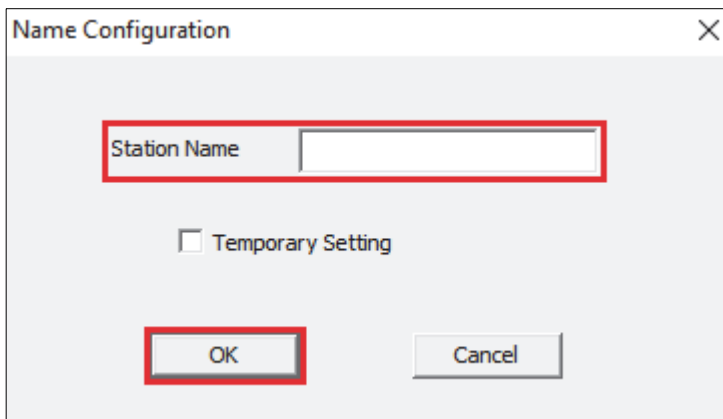
5. **Edit IP Address**. Set the **IP Address**, the **Subnet Mask**, and the **Default Gateway**. Select **Temporary Setting** if required, then click **OK**.

PROFINET I/O

PROFINET I/O Identity

Vendor ID: Omron Microscan's Systems, Inc. Vendor ID is 0x0257.

Device ID: See table below.

Main Family:

F440-F: General

MicroHAWK: General

HAWK MV-4000: General

Product Family:

F440-F: Smart Camera

MicroHAWK: Smart Camera

HAWK MV-4000: Smart Camera

GSDML Files

See table below.

Product	FW Version	GSD File	Version	Device ID
MV-40	5.0.0	GSDML-V2.32-Microscan-MicroHawkMV40-20160824.xml	V2.32	0x7000
	5.1.0*	GSDML-V2.32-Microscan-MicroHawkMV40-20190311.xml	V2.32	0x7000
		GSDML-V2.33-Microscan-MicroHawkMV40-20190311.xml	V2.33	0x7000
	5.2.0*	GSDML-V2.32-Microscan-MicroHawkMV40-20190311.xml	V2.32	0x7000
		GSDML-V2.33-Microscan-MicroHawkMV40-20190311.xml	V2.33	0x7000
	5.2.2	GSDML-V2.35-OmronMicroscan SystemsInc-MicroHAWK-MV-20210302.xml	V2.35	0x7000
F430-F	5.2.2	GSDML-V2.35-OmronMicroscan SystemsInc-MicroHAWK-MV-20210302.xml	V2.35	0x7000
F330-F	5.2.2	GSDML-V2.35-OmronMicroscan SystemsInc-MicroHAWK-MV-20210302.xml	V2.35	0x7000
F440-F	5.3.0	GSDML-V2.42-OmronMicroscan SystemsInc-F440-F-20221027.xml	V2.42	0x7005
MV-4000	5.2.0	GSDML-V2.33-Omron_Microscan_Systems_Inc-HAWKMV4000-20180530.xml	V2.33	0x7004
	5.2.2	GSDML-V2.33-Omron_Microscan_Systems_Inc-HAWKMV4000-20180530.xml	V2.33	0x7004

PROFINET

*Depends on older or newer PLC/TIA Portal.

Connection Properties: RT Cyclic Messaging

Odd slot numbers are input to the PLC, even slot numbers are output from the PLC.

Maximum data size in either direction is 518 bytes. The data size can be reduced by removing slots that are not used, except in the case of the F440-F, which has a fixed layout.

Cycle update time for MicroHAWK F330-F: **16 ms**

Cycle update time for MicroHAWK F430-F: **16 ms**

Cycle update time for F440-F: **16 ms**

Cycle update time for HAWK MV-4000: **16 ms**

Definition: The GSD file contains element MinDeviceInterval, which is 512. Multiply this by 31.25 µs. This is the cycle time. See the PROFINET GSDML specification for more information.

Slot/Subslot Layout Descriptions

Slot	Dir	Bytes	Name	Description
1	In	2	STATUS	Status register of the camera, each bit of this register represents a different state item. See *Camera Status Register* for bit descriptions
3	In	2	ECHO	This 16 bit word value reflects back to the PLC the value that the PLC wrote to the output assembly ECHO register. The PLC can verify the output assembly has been written to the camera when this value matches the written value.
5	In	4	CmdCodeRslt	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdCodeRslt reflects the result of the command invoked by Control.CmdCode. See *CmdCodeRslt* for definitions.
7	In	4	CmdRet	When Status.ExeCmdAck goes active in response to Control.ExeCmd, CmdRet contains the data returned from the command invoked by Control.CmdCode. See *CmdRet* for definitions.
9	In	2	State	Device State register. Depending on the current state of the camera, certain STATUS and CONTROL features may or may not be operational. See *State* for

				definitions.
2	Out	2	CONTROL	Control register of camera. Each bit of this register represents a different status item. See *Camera Control Register* for bit descriptions
4	Out	2	ECHO	This 16 bit value is reflected back to the PLC in the input assembly ECHO register. The PLC can verify the output assembly has been written to the camera when the input assembly matches this written value.
6	Out	4	CmdCode	Specifies the process invoked in the camera when Control.ExeCmd goes active. See *CmdCode* for definitions.
8	Out	4	CmdArg	Additional argument data for the CmdCode. See *CmdArg* for definition.
11	In	2	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 145, the most significant bit vio point 160
10	Out	2	VIO	Each bit reflects the state of a virtual IO point. The least significant bit reflects vio point 129, the most significant bit is vio point 144
13	In	8	bool1-64	Each bit represents a bool value. The least significant bit of byte 0 reads the value of bool1. The most significant bit of byte 7 reads bool64.
12	Out	8	Bool101-164	Each bit represents a bool value. The least significant bit of byte 0 writes the value of bool101. The most significant bit of byte 7 writes bool164.
15	In	20	int1-10	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 0-1 for the value of int1 through bytes 18-19 for the value of int10.

14	Out	20	int101-110	Each pair of sequential bytes represents a 16 bit signed integer value. The 20 bytes represent 10 integers. From bytes 0-1 to write the value of int101 through bytes 18-19 for the value of int110.
17	In	64	long1-16	Each group of 4 bytes represents a 32 bit signed integer value. The 64 bytes represent 16 long integers. From bytes 0-3 for the value of long1 through bytes 60-63 for the value of long16.
16	Out	64	long101-116	Each group of 4 bytes represents a 32 bit signed integer value. The 64 bytes represent 16 long integers. From bytes 0-3 for the value of long101 through bytes 60-63 for the value of long116.
19	In	96	float1-24	Each group of 4 bytes represents a 32 bit signed integer value. The 96 bytes represent 24 long integers. From byte offsets 0-1 for the value of float1 through byte offsets 92-95 for the value of float24.
18	Out	96	float101-124	Each group of 4 bytes represents a 32 bit signed integer value. The 96 bytes represent 24 long integers. From bytes 0-3 for the value of float101 through bytes 92-95 for the value of float124.
21	In	96	string1	These 96 bytes can store a string of up to 94, 8 bit characters, with the first 2 bytes containing the storage length and string length values.
20	Out	96	string101	These 96 bytes can store a string of up to 94, 8 bit characters, with the first 2 bytes containing the storage length and string length values.
23	In	96	string2-string7	6 consecutive strings, each of 32 bytes can store a string of up to 30, 8 bit characters, with the first 2 bytes of each string group containing the storage length and string length values.
22	Out	96	string102-string107	6 consecutive strings, each of 32 bytes can store a string of up to 30, 8 bit characters, with the first 2 bytes of each string group containing the storage length and string length values.

Slot Data Layout Diagrams

PLC Input

Slot	Byte Offset	Data
1	0	STATUS
3	0	Echo In
5	0	CMD CODE RSLT
7	0	CMD RET
9	0	STATE
11	0	VIO 145.. 160
13	0	bool 1.. 16
13	2	bool 17.. 32
13	4	bool 33.. 48
13	6	bool 49.. 64
15	0	int 1
15	2	int 2
15	4	int 3
15	6	int 4
15	8	int 5
15	10	int 6
15	12	int 7
15	14	int 8
15	16	int 9
15	18	int 10
17	0	long 1
17	4	long 2
17	8	long 3
17	12	long 4
17	16	long 5
17	20	long 6
17	24	long 7
17	28	long 8
17	32	long 9
17	36	long 10
17	40	long 11
17	44	long 12
17	48	long 13
17	52	long 14
17	56	long 15
17	60	long 16

PLC Output

Slot	Byte Offset	Data
2	0	CONTROL
4	0	Echo Out
6	0	CMD CODE
8	0	CMD ARG
10	0	VIO 129.. 144
12	0	bool 101.. 116
12	2	bool 117.. 132
12	4	bool 133.. 148
12	6	bool 149.. 164
14	0	int 101
14	2	int 102
14	4	int 103
14	6	int 104
14	8	int 105
14	10	int 106
14	12	int 107
14	14	int 108
14	16	int 109
14	18	int 110
16	0	long 101
16	4	long 102
16	8	long 103
16	12	long 104
16	16	long 105
16	20	long 106
16	24	long 107
16	28	long 108
16	32	long 109
16	36	long 110
16	40	long 111
16	44	long 112
16	48	long 113
16	52	long 114
16	56	long 115
16	60	long 116

PLC Input

Slot	Byte Offset	Data
	0	float 1
	4	float 2
	8	float 3
	12	float 4
	16	float 5
	20	float 6
	24	float 7
	28	float 8
	32	float 9
	36	float 10
	40	float 11
	44	float 12
19	48	float 13
	52	float 14
	56	float 15
	60	float 16
	64	float 17
	68	float 18
	72	float 19
	76	float 20
	80	float 21
	84	float 22
	88	float 23
	92	float 24

PLC Output

Slot	Byte Offset	Data
	0	float 101
	4	float 102
	8	float 103
	12	float 104
	16	float 105
	20	float 106
	24	float 107
	28	float 108
	32	float 109
	36	float 110
	40	float 111
	44	float 112
18	48	float 113
	52	float 114
	56	float 115
	60	float 116
	64	float 117
	68	float 118
	72	float 119
	76	float 120
	80	float 121
	84	float 122
	88	float 123
	92	float 124

PLC Input

Slot	Byte Offset	Data
	0	94
	1	<str 1 len>
21	2	string 1
	95	
	0	30
	1	<str 2 len>
	2	string 2
	31	
	32	30
	33	<str 3 len>
	34	string 3
	63	
	64	30
	65	<str 4 len>
23	66	string 4
	95	
	96	30
	97	<str 5 len>
	98	string 5
	127	
	160	30
	161	<str 6 len>
	162	string 6
	191	

PLC Output

Slot	Byte Offset	Data
	0	94
	1	<str 101 len>
20	2	string 101
	95	
	0	30
	1	<str 102 len>
	2	string 102
	31	
	32	30
	33	<str 103 len>
	34	string 103
	63	
	64	30
	65	<str 104 len>
22	66	string 104
	95	
	96	30
	97	<str 105 len>
	98	string 105
	127	
	160	30
	161	<str 106 len>
	162	string 106
	191	

PLC Slot Layout for Omron Microscan Smart Cameras

▼ dut	0	0
▼ Interface	0	0 X1
Port 1	0	0 X1 P1
Status_1	0	1
Control_1	0	2
Echo In_1	0	3
Echo Out_1	0	4
Cmd Code Rslt_1	0	5
Cmd Code_1	0	6
Cmd Ret_1	0	7
Cmd Arg_1	0	8
State_1	0	9

Status: Camera Status Register (16-bit)

Each bit of this register represents a signal that displays the camera's operational status. A high value of **1** indicates that the signal is **active** (true).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DATA VALID	INSP STAT	INSP BUSY	TRIGGER ACK	EXE CMD ACK		RESET COUNT ACK	ERROR	TRIGGER READY	ACQ BUSY	EXP BUSY	ONLINE

← Inspection 1 → ← All Inspections →

Bit	Name	Description
0	ONLINE	Inspections are running
1	EXP BUSY	The camera is busy capturing an image. The camera should not be triggered or the part under inspection moved during this time if illuminated. F440-F: EXP BUSY equivalent to ACQ BUSY.
2	ACQ BUSY	The camera is busy acquiring an image. The camera cannot be triggered while busy.
3	TRIGGER READY	The camera is ready to be triggered. This is equivalent to ONLINE == 1 and ACQ BUSY == 0.
4	ERROR	An error has occurred. Set the RESET ERROR control bit high to clear.
5	RESET COUNT ACK	This bit mirrors the RESET COUNT control bit. The PLC can be certain the reset command was received by the camera when this goes high. The PLC can then bring the RESET COUNT control signal back low.
7	EXE CMD ACK	This bit mirrors the EXE CMD control bit.
8	TRIGGER ACK	This bit mirrors the TRIGGER control bit.
9	INSP BUSY	This bit is high when inspection 1 is busy processing an image.
10	INSP STAT	This bit represents the inspection 1 status result. It is 1 if the inspection passes. It is only valid when DataValid goes high.
11	DATA VALID	This bit goes high when inspection 1 is complete. The PLC should clear this signal by setting RESET DV high once it has read results.

CmdCodeRslt (32-bit)

The value of CmdCodeRslt is only valid when **ExeCmdAck** is active (**1**), in response to **ExeCmd** being active.

CmdCodeRslt value (base 16 hex)	Meaning
0x0000_0000	Success
0x0100_0000	Fail. Possible reasons: Camera under PC control. Job cannot be changed.
0x0200_0000	Fail: No Job in slot.
0x0300_0000	Fail: Unknown cmd.

CmdRet (32-bit)

The value of **CmdRet** is only valid when **ExeCmdAck** is active (**1**), in response to **ExeCmd** being active, and **CmdCodeRslt** is **0** (Success). The following chart shows which **CmdCodes** return data in the **CmdRet** register.

CmdRet value (32 bit)	Associated CmdCode	Meaning
0	0x1000_0000 to 0x1300_0000 (Job Change type)	Na
1 – 255	0x1800_0000 (Query Active Job Slot)	Active Job Slot #

State (16-bit)

State reflects the following operational condition of the camera:

State value (16 bit)	Meaning	Typical action required by the client (PLC), or system operator
0	Offline	Perform job change or put camera online.
1	Online	Normal runtime operation: Monitor TriggerReady and DataValid signals. Trigger the camera.
2	Changing Vision Job	If camera is under pc control: Wait until State changes to Offline or Online. If PLC is controlling the job change: Use ExeCmd, CmdCode, ExeCmdAck, and CmdCodeRslt to complete the operation.
3	Booting*	Wait for camera to transition to Online or Offline.
4	Empty (no Vision Job)	Load a new job from AutoVISION or Front Runner.

*Booting (3) State: This will rarely be seen by the PLC.

The value of State determines which Control and Status signals are available:

Control/Status Signal	State				
	0 (Offline)	**1** (Online)	**2** (Job Change)	**3** (Booting)	**4** (Empty)
Control.GO ONLINE	Y				
".GO OFFLINE		Y			
".RESET ERROR					
".RESET COUNT	Y	Y			
".EXE CMD	Y	Y	Y		Y
".TRIGGER		Y			
".RESET DATA VALID		Y			
Status.ONLINE	Y	Y	Y	Y	Y
".ERROR					
".RESET COUNT ACK	Y	Y			
".EXE CMD ACK	Y	Y	Y		Y
".EXP BUSY		Y			
".ACQ BUSY		Y			
".TRIGGER READY		Y			
".TRIGGER ACK		Y			
".INSP BUSY		Y			
".INSP STAT		Y			
".DATA VALID		Y			

Where:
Y = Signal is valid for this State
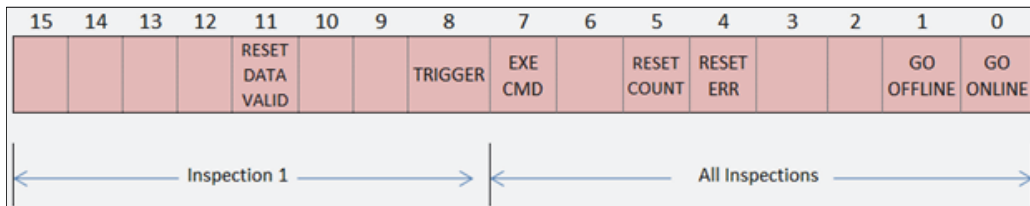Empty cell = Signal is not valid for this State

VIO Output Register Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v144	v143	v142	v141	v140	v139	v138	v137	v136	v135	v134	v133	v132	v131	v130	v129

VIO Input Register Bits

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v160	v159	v158	v157	v156	v155	v154	v153	v152	v151	v150	v149	v148	v147	v146	v145

Control: Camera Control Register (16-bit)

Each bit of this register controls a function on the camera. Transitions from a low state of **0** to a high state of **1** initiate the associated operation. The PLC should return the state of the control bit back to **0** after it has acknowledged the camera has processed the control. Unused bits should remain **0**. Setting the **Reset Data Valid** bit (bit **11**) will also reset the **Error** bit (bit **4**) in the **Camera Status Register**.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RESET DATA VALID			TRIGGER	EXE CMD		RESET COUNT	RESET ERR			GO OFFLINE	GO ONLINE

←———— Inspection 1 ————→ ←———————— All Inspections ————————→

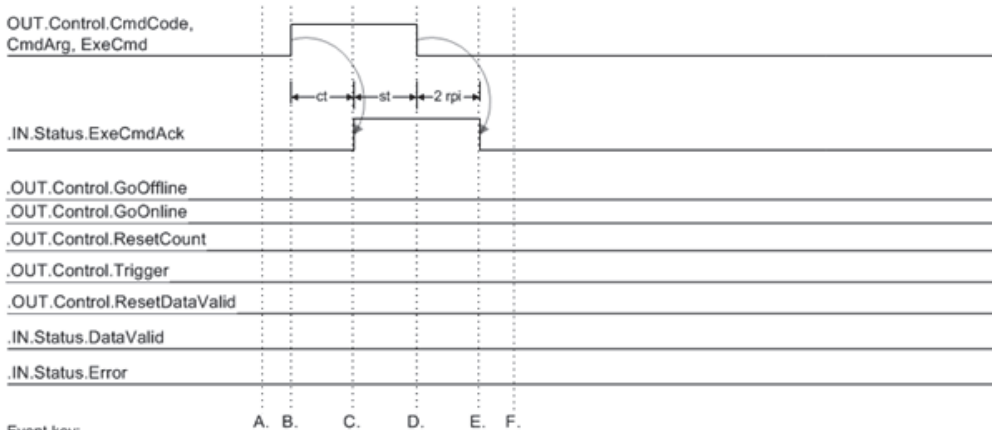| Bit	Name	Description
0	GO ONLINE	Start all inspections running
1	GO OFFLINE	Stop all inspections
4	RESET ERROR	Reset ERROR in the Status register
5	RESET COUNT	Reset all inspection counts
7	EXE CMD	Execute the command specified by Control.CmdCode
8	TRIGGER	Trigger Inspection 1. The inspection must be configured for a triggered image acquisition.
11	RESET DATA VALID	Reset the Data Valid signal of the Status register

CmdCode and CmdArg (32-bit)

Specifies the process invoked in the camera when **Control.ExeCmd** goes active. The **CmdCode** and **CmdArg** must be set before setting the EXE CMD bit in the control register. Do not set all the values within the same update time.

List of Available CmdCodes and Associated CmdArg

CmdCode value	CmdArg	Operations performed
0x1000_0000	Job Slot (1-255)	Go Offline, Load job from specified slot
0x1100_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Go Online
0x1200_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job
0x1300_0000	Job Slot (1-255)	Go Offline, Load job from specified slot, Make it the boot job, and Go Online
0x1800_0000	na	Query active job slot. *CmdRet* will contain the active job slot number when the operation is done.

CmdCode and ExeCmd Operation

Event key:
A. If DataValid or Error are present, clear them.
   Set the following control signals idle, and keep them idle while the command is processed by the camera:
      GoOffline, GoOnline, Trigger, ResetDataValid, ResetCount, ResetError.
   If the command operation is a job change, populate the output tags required to configure the new job (bool, int, long, float, string).
B. Populate CmdCode and CmdArg, then activate ExeCmd.
C. Camera executes the command (may take up to a minute). While processing a Job Change command, State will be 2. Camera activates
   ExeCmdAck when it is done processing the command.
D. When the PLC sees an active ExeCmdAck, verify CmdCodeRslt is 0, and Error is 0. Process CmdRet if needed, then clear ExeCmd.
E. Camera clears ExeCmdAck when ExeCmd goes inactive. When ExeCmdAck goes inactive, CmdCodeRslt and CmdRet are no longer valid,
   and it may take a few seconds for the camera State and Online signals to settle to a final value (typically Online or Offline).
F. Camera can now be put online and triggered.

Notes:
st = PLC program scan time
ct = Command processing time in the camera. May take up to a minute for some commands.
rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.
All signals represent the state of plc tags.

PROFINET I/O Control/Status Signal Operation

Event key:

A. On rising edge of system trigger, the user app activates Scan_MV_IO_user.OUT.Control.Trigger to trigger the demo code.

B. Demo code detects rising edge of Scan_MV_IO_user.OUT.Control.Trigger, and if the camera is ready, sends a trigger to the camera.

C. Camera acquisition begins (may be delayed by one rpi).

D. If the camera's exposure time is shorter than the rpi, no change will be seen in TriggerReady and AcqBusy plc IN tags.

E. Camera firmware acks the trigger. The demo code may not see the ack until two rpi after the trigger was sent (event B).

F. Demo code detects TriggerAck and clears the Trigger.

G. Demo code detect falling edge of TriggerAck and clears the user Trigger.

H. Camera internal signal DataValid will go high when InspBusy goes low

I. Plc logic must delay one rpi time before re-asserting ResetDataValid

Notes:

1. The chart shows the workings of the Trigger and ResetDataValid Control signals, and the TriggerAck and DataValid Status signals.

2. st = plc program scan time

3. rpi = Requested Packet Interval. Configured in the plc's EIP module connection properties. Allowed rpi is 10 ms to 3.2 s.

4. All signals represent the state of plc tags, except where noted as "(camera)". The cam signals shown are visible in the EIP interface, but the state of the plc tags and internal firmware signals will be different for at least one or two requested packet intervals (rpi).

5. The plc is running the demo code distributed with the camera. The demo code and user app use the Scan_MV_IO_user tag set as the primary control, status, and data interface for the user app. All signal operations are still true even if the plc demo code is not used.

6. TriggerReady/!AcqBusy: Camera exposure times can range from less than 1 ms, up to 100 ms.

9

PROFINET I/O Operation Using Sysmac Studio and Omron NJ Controller

This section provides information necessary for using an Omron Microscan Smart Camera in a PROFINET I/O environment.

Notes:

• The camera communications protocol must be enabled for PROFINET I/O before it can be used in this environment. Refer to **Chapter 1**, **Enabling Industrial Protocols**, for information about enabling communications protocols for the camera, and information about switching camera communications protocols.

• AutoVISION and FrontRunner jobs use Omron Microscan Link functionality to accommodate PROFINET I/O communications between the camera and the PLC. For information about how to connect job parameters and outputs to Omron Microscan Link tags, refer to the **Edit > Omron Microscan Link > Link Menus** topic in **AutoVISION Help**, accessible from AutoVISION Software. See also **Linking Datums to Omron Microscan Link Tags** in **Chapter 2** of the *Visionscape FrontRunner User Manual*.

Important: PROFINET I/O allows the IP address and subnet mask of the camera to be assigned automatically by the PLC. In order to allow this, the camera may not have an assigned IP address on reboot until the PLC is set to Run mode. During this time, the camera will not be visible on the network for AutoVISION or Visionscape FrontRunner.

Important: The PNT21 module with the NJ controller allows Omron Microscan Smart Cameras to communicate via PROFINET I/O. The maximum Input CR size for the PNT21 is 450 bytes. Due to the fixed layout for the F440-F Smart Camera, it is not supported by the PNT21.

PROFINET I/O Operation with Omron NJ Controller

1. Configure the device's IP address.

2. Create a job using **AutoVISION** or FrontRunner. Refer to AutoVISION Help to configure the data to interact with the PLC.


3. Start **Sysmac Studio**. If the **User Account Control Dialog** box is displayed at startup, select the option to start Sysmac Studio.

4. Sysmac Studio starts. Click **[New Project]**.
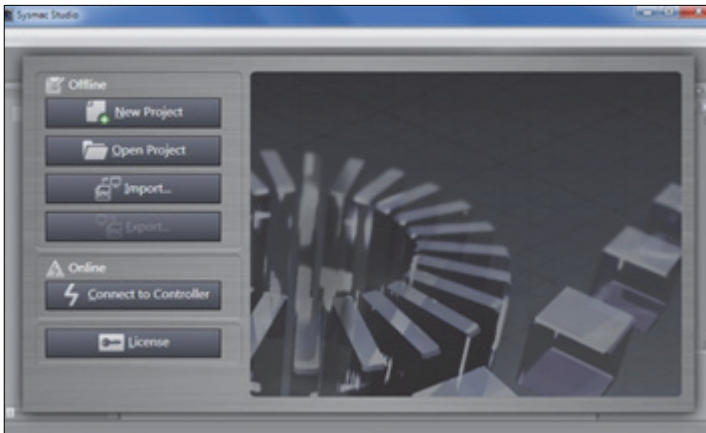
5. The **Project Properties** screen is displayed. In this document, **"New Project"** is used as the project name. Select the device to use from the **[Select Device]** dropdown list. Select an applicable version from the **[Version]** dropdown list. Although 1.09 is used as an example in this document, select the version actually required by the application.
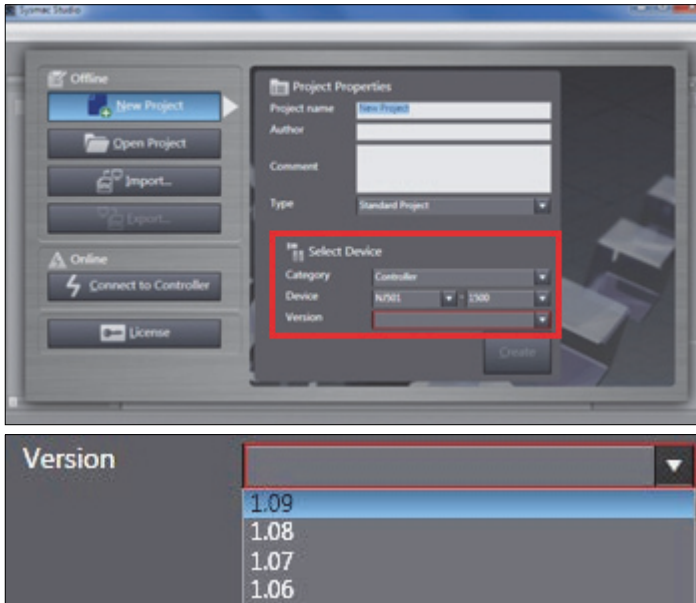
6. Click **Create**.
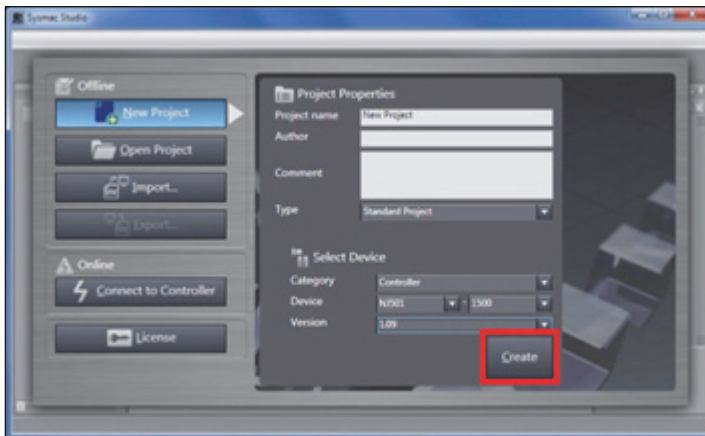
7. The new project is displayed. The following panes are displayed in this window.

   - **Left: Multiview Explorer**.

   - **Top right: Toolbox**.

   - **Bottom right: Controller Status** pane.

   - **Middle Top: Edit** pane.

   The following tab pages are displayed at the middle bottom of the window.

   - **Output Tab** page.

   - **Build Tab** page.
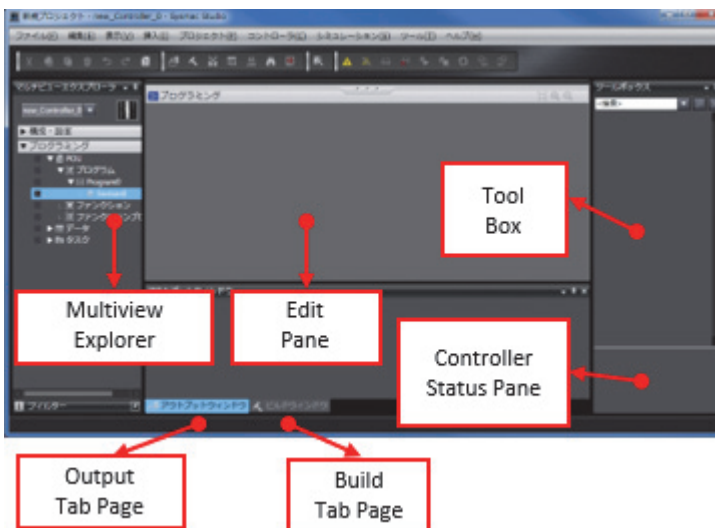
8. Double-click **Built-in EtherNet/IP Port Settings** under **Configurations and Setup > Controller Setup** in the **Multiview Explorer**.
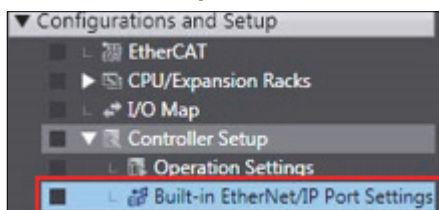
9. The **Built-in EtherNet/IP Port Settings** tab page is displayed in the **Edit** pane. For the **IP Address** settings.

- **IP Address: 192.168.10.18**.

- **Subnet Mask: 255.255.0.0**.

Verify that the controller can be pinged and that Sysmac Studio can communicate with it.
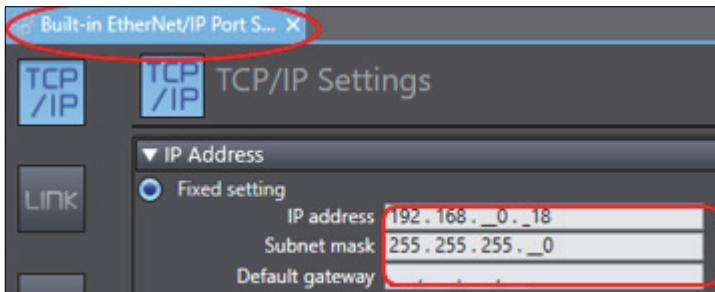
# Adding the PNT21 to the Project in Sysmac Studio

1. Go online with the controller. In the **Multiview Explorer**, double-click on **CPU/Expansion Racks**.
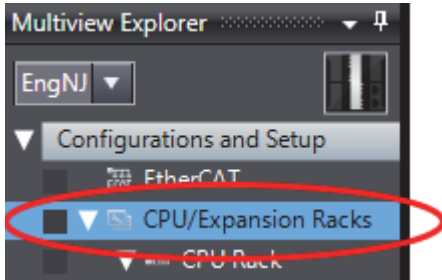
2. A graphical representation of the CPU Rack will be shown. Right-click on the CPU module. Click **Compare and Merge with Actual Unit Configuration**.
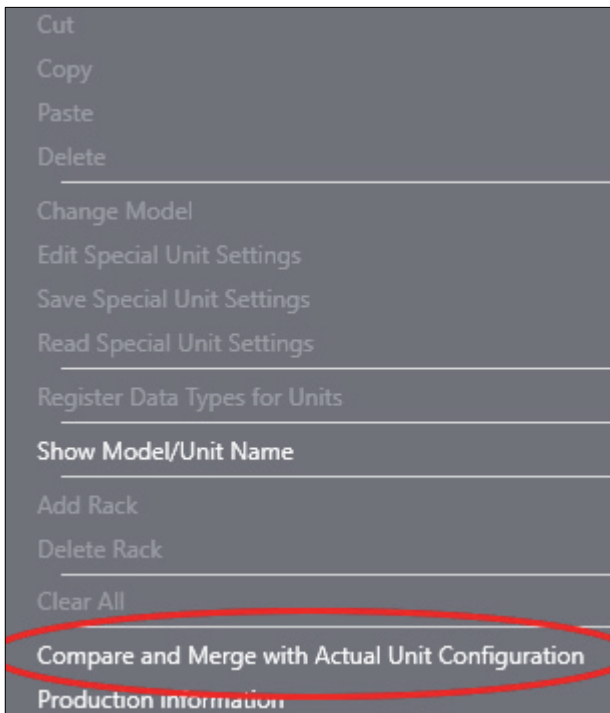
3. A display of the current CPU Rack vs the actual CPU rack will be displayed. The PNT21 will be displayed in the **Actual Unit Configuration** on the bottom. Click the **Apply Actual Unit Configuration** button.
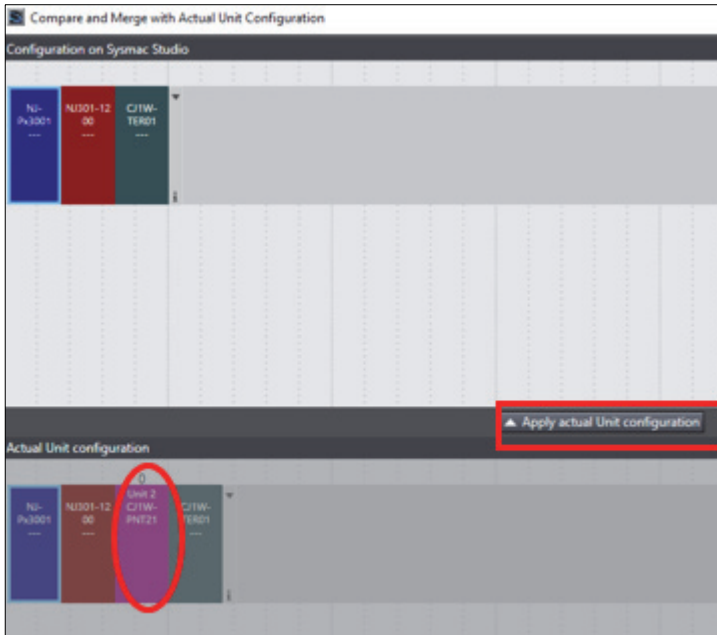
4. An updated comparison is now displayed. Be sure the **PNT21** is in both displays.

5. Save the project and synchronize the program to the controller.
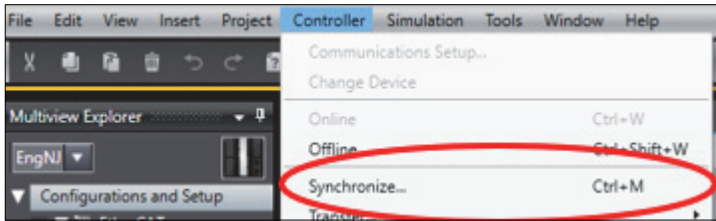
**6.** Transfer the program to the controller.
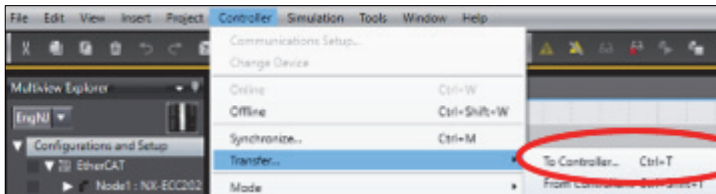
PNT21 PROFINET Controller and IO Device Setup Using CX-Configurator-FDT

## Installing the GSD File and Network Build

1. Open **CX-Configurator**. If there is no **Device Catalog** screen to the right, click the Device Catalog icon to display it.

On the bottom right click **Install Device Description Files**. Under the **Device Catalog** on the right, browse to the folder where the FXX0-F GSD file is located, select it, and click **Open**. Follow the prompts to add the file to the system.
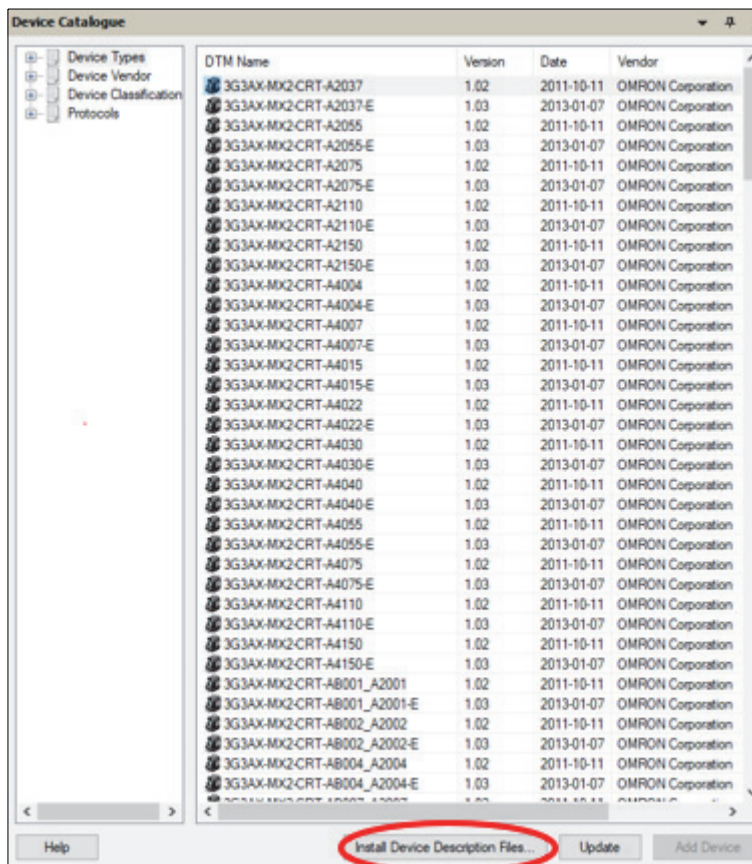
2. Expand **Device Types** and click **Communication**. Select **CJ1W-PRNT21 PROFINET IO Controller**. Drag and drop this device into **My Network** on the left.
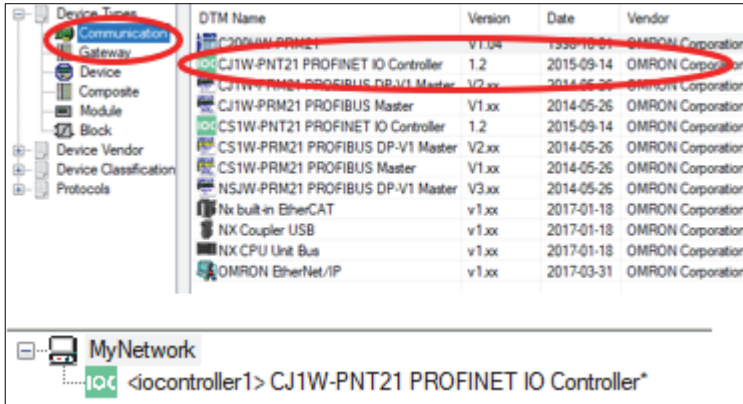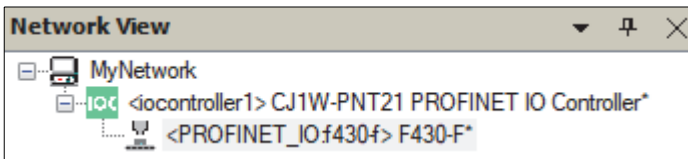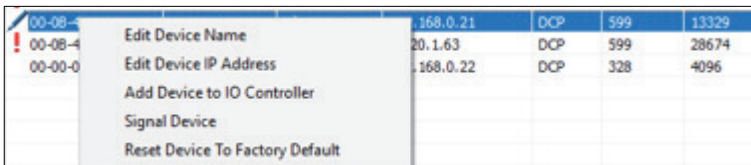

3. Double-click **<iocontroller1>**. This will open the PNT21 configuration page. Expand **PROFINET Identification**, then click **Network Scan**, then **Start Scan** below. This will provide a list of devices found on the network.
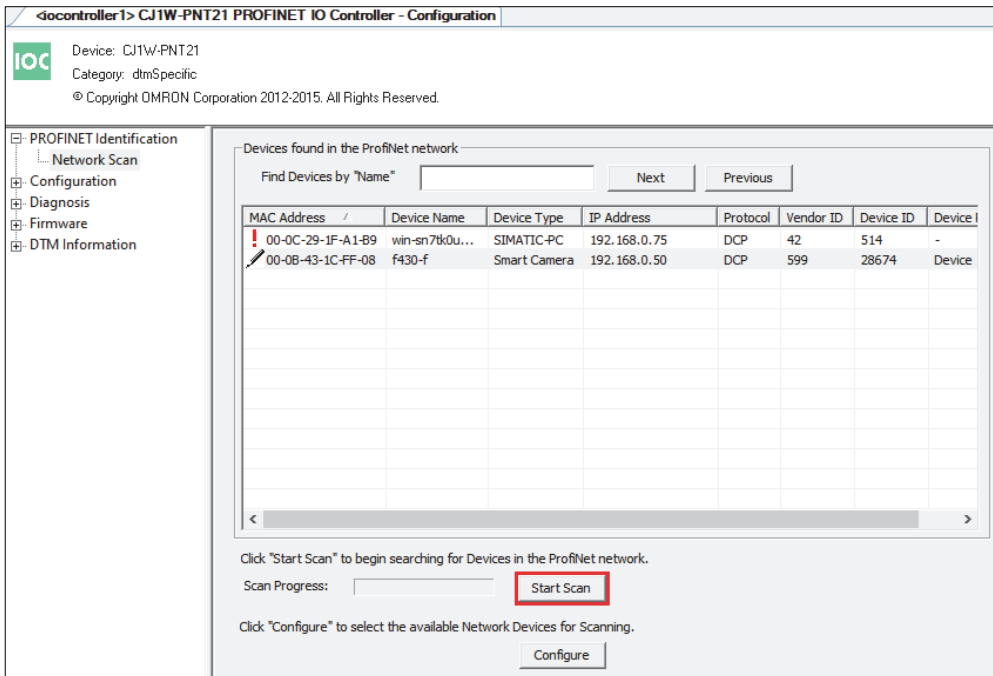
4. Locate the MAC address of the device to be added and right click on it.

Edit Device Name to set the name of the IO device. In this case, it was set to **F430-F**. **Edit Device IP Address** to set the PROFINET IO IP address of the device. In this case it was set to **192.168.0.50**. After the device is configured, click **Add Device to IO Controller**. The following will be displayed under **My Network**.

MAC Address /	Device Name	Device Type	IP Address	Protocol	Vendor ID	Device ID	Device
! 00-0C-29-1F-A1-B9	win-sn7tk0u...	SIMATIC-PC	192.168.0.75	DCP	42	514	-
00-0B-43-1C-FF-08	f430-f	Smart Camera	192.168.0.50	DCP	599	28674	Device

00-0B-4	Edit Device Name		168.0.21	DCP	599	13329
! 00-0B-4	Edit Device IP Address		20.1.63	DCP	599	28674
00-00-0	Add Device to IO Controller		168.0.22	DCP	328	4096
	Signal Device					
	Reset Device To Factory Default					

Network View ▼ ⊞ ✕

□ 🖳 MyNetwork
    □ 🔲 <iocontroller1> CJ1W-PNT21 PROFINET IO Controller*
        ⋯ 🖳 <PROFINET_IO:f430f> F430-F*
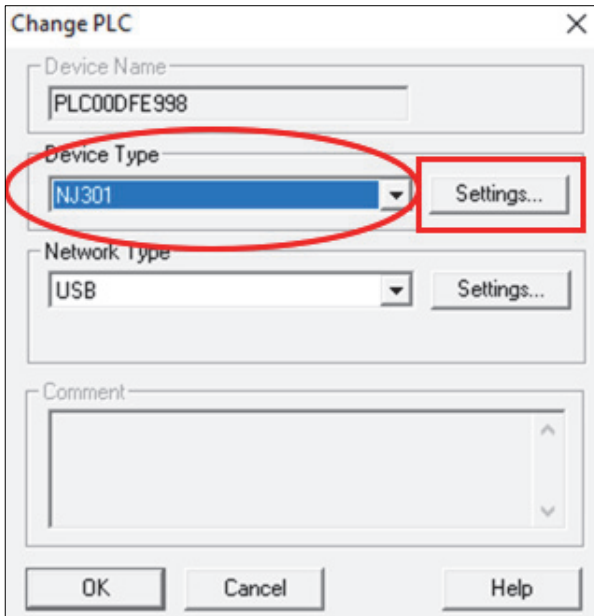
PNT21 Module Configuration

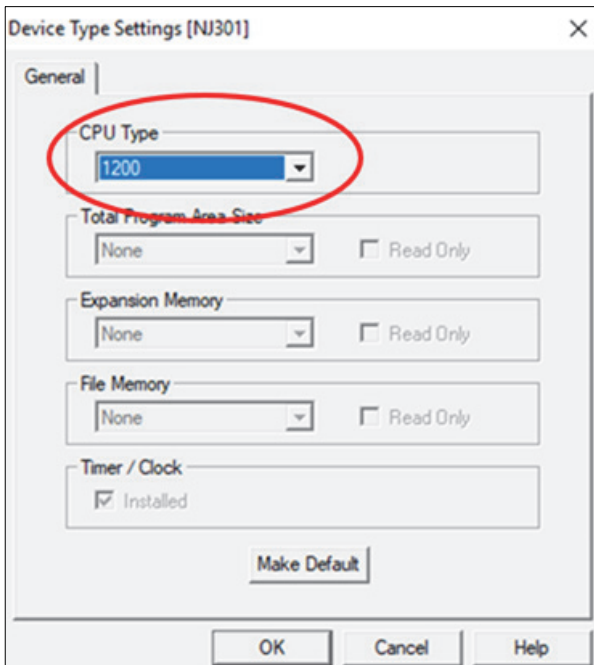1. In the **<iocontroller1>** configuration screen, expand **Configuration** and click **PLC Setup**. Select the **Unit Number** that has been set to the **PROFINET Controller**, **2** in this case. This must match the rotary switches set on the PNT21 front panel.
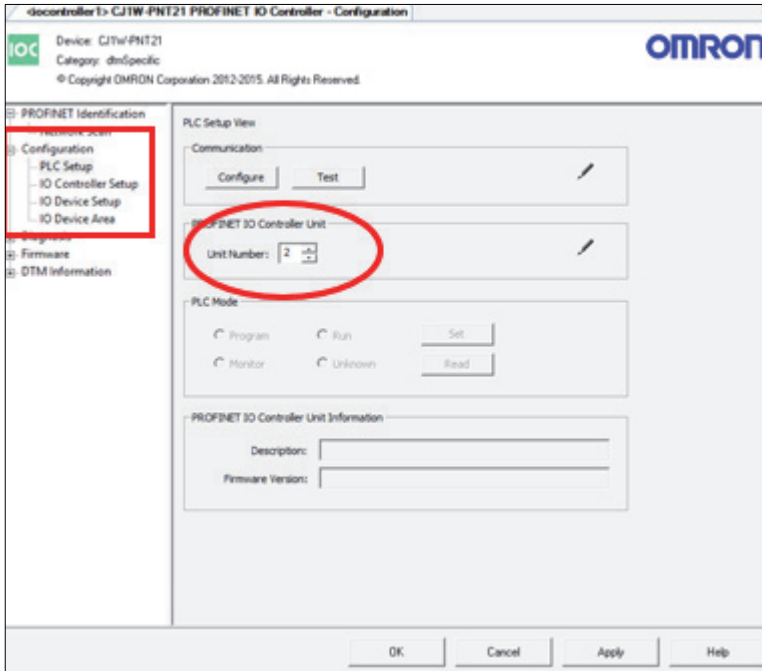
2. Click **Configure**. Select the PLC type, then click **Settings**. Select the **CPU Type**, then click **OK**.
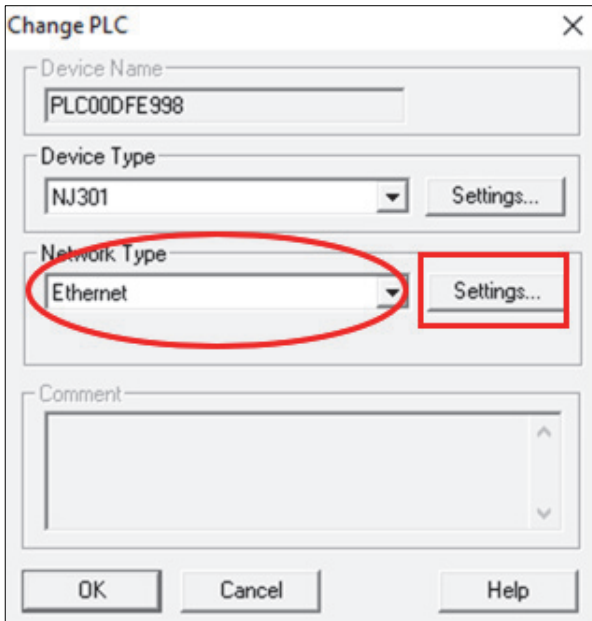
3. Set the **Network Type** to **Ethernet**, then click Settings. Select the **Driver Tab**. Enter the IP address of the controller that was set in Sysmac Studio earlier. Under **Workstation Node Number**, uncheck the **Auto Detect** box and put in the last octect of the PC's IP address. In this example, the PC interface IP is **192.168.0.220**. Enter **220** here, click **OK**, and then **OK** again.
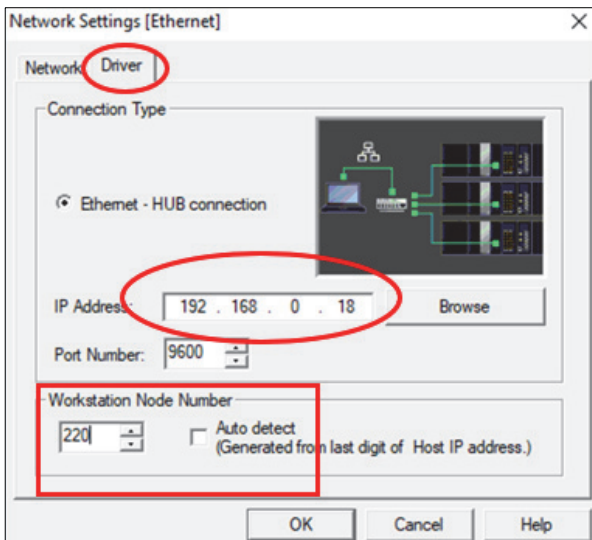
4. On the left under **Configuration**, click **IO Controller Setup**.

5. In the **IP Address** field enter the IP address you want the PNT21 to have (not the PLC IP). In this example the IP address is **192.168.0.22**. Click **OK**.

6. Under **Configuration**, click **IO Device Setup**. The selected device is displayed here. Click the IP address field and enter the F430-F IP address. In this example, the IP address is **192.168.0.50**. Click **Apply**.

IO Device Setup		Device No. ∠	Device Name	Device Type	IP Address	Update Rate (ms)
IO Device Area		1	f430-f	F430-F	192.168.0.50	16
Diagnosis						

7. Under **My Network**, double-click **<PROFINET_IO:f430-f> F430-F**. Click the **Add Module** button. Add the modules required. Click **OK** when finished.

<PROFINET_IO:f430-f> F430-F - Configuration   <iocontroller1> CJ1W-PNT21 PROFINET IO Controller - Configuration

IO Device: F430-F
Vendor: Omron Microscan Systems, Inc.

Navigation Area
- Configuration
  - General
  - → Modules
- Description
  - Device Info
  - Module Info
  - GSDML Viewer

	Slot	Sub Slot	!	
⊞	0			F430-F
⊞	1			Status
⊞	2			Control
⊞	3			Echo In
⊞	4			Echo Out
⊞	5			Cmd Code Rslt
⊞	6			Cmd Code
⊞	7			Cmd Ret
⊞	8			Cmd Arg
⊞	9			State

Add Module   Add Submodule   Remove   Duplicate

Navigation Area
- Configuration
  - General
  - → Modules
- Description
  - Device Info
  - Module Info
  - GSDML Viewer

	Slot	Sub Slot	!	
⊞	0			F430-F
⊞	1			Status
⊞	2			Control
⊞	3			Echo In
⊞	4			Echo Out
⊞	5			Cmd Code Rslt
⊞	6			Cmd Code
⊞	7			Cmd Ret
⊞	8			Cmd Arg
⊞	9			State
⊟	10			VIO Out
		1		VIO
⊟	11			VIO In
		1		VIO

Add Module   Add Submodule   Remove   Duplicate

Use of slots: 24/25

8. Double-click **<iocontroller1>** again, then go to **Configure**, then **IO Device Area**. Click the **Output** tab.
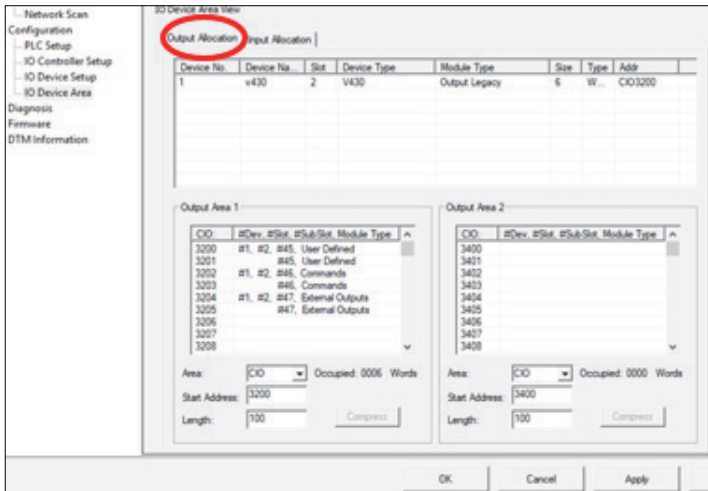
9. Select in **Output Area 1**:
   • **Area: CIO**
   • **Start Address: 3200**
   • **Length: 100**
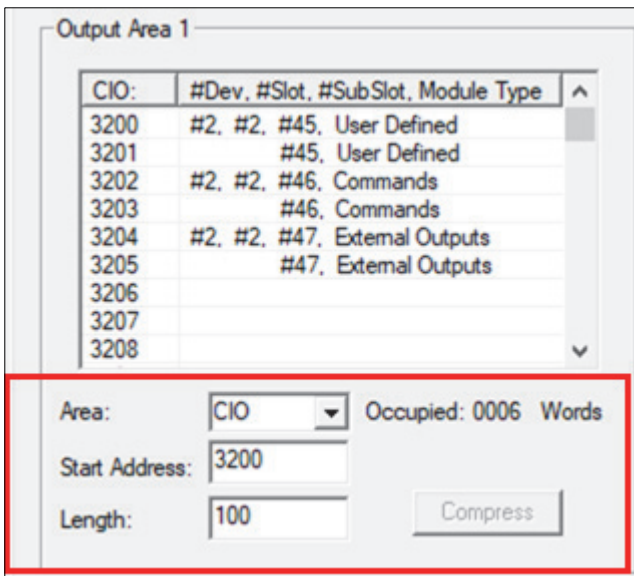
10. Select in **Output Area 2**:
    • **Area: CIO**
    • **Start Address: 3400**
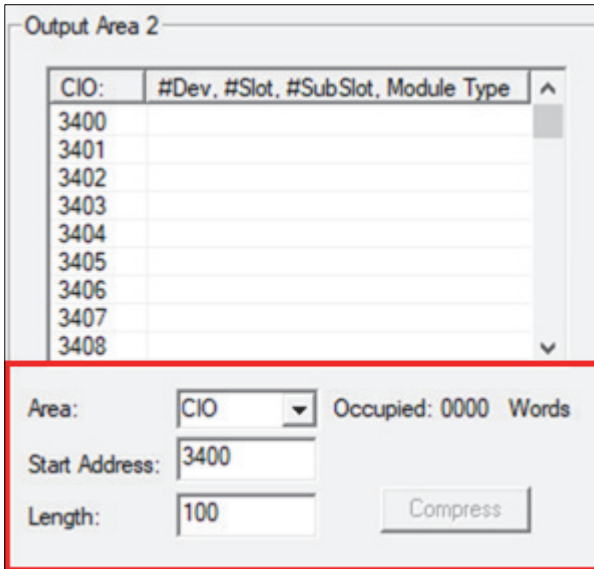
11. Double-click **<iocontroller1>** again, then go to **Configure**, then **IO Device Area**. Click the **Input** tab.
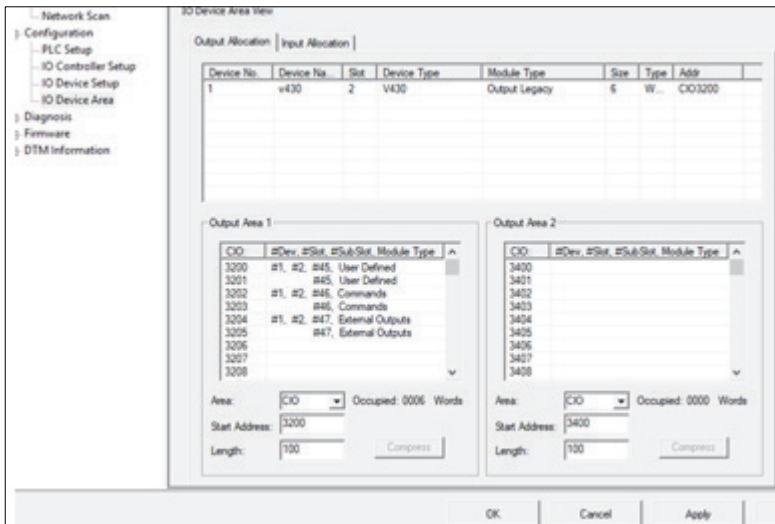
12. Select in **Input Area 1**:

- **Area: CIO**
- **Start Address: 3300**
- **Length: 100**

13. Select in **Input Area 2**:

- **Area: CIO**
- **Start Address: 3500**
- **Length: 100**

Click **OK**.

14. Click on **main menu > File**, then **SaveAs**. **Save** the project.

On the left under **My Network**, right-click **<iocontroller1>** and click **Go Online**.

Next, click **Store to Device**. The PROFINET configuration is now loaded and ready to run. An **r.** will appear in the LED display on the PNT21 module, indicating that the device is in run mode.
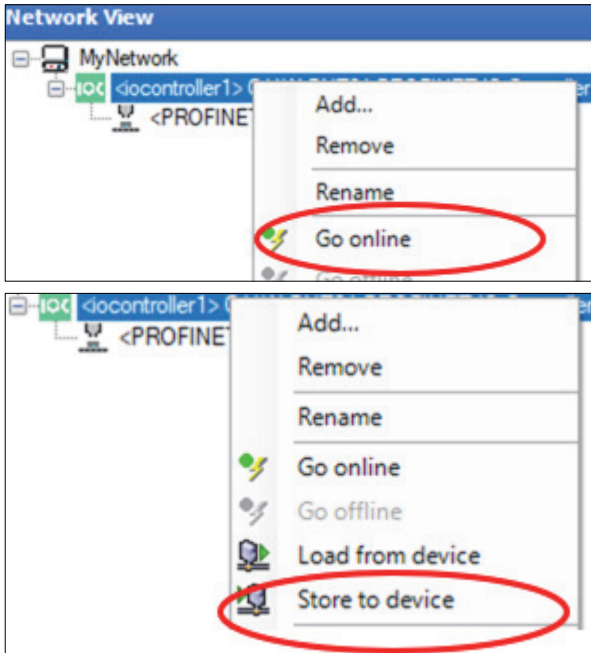
Finalizing the Sysmac Studio Project for PROFINET I/O Operation

Adding Data Types

In this example, the **Large Input Module** and the **Legacy Output Module** are used as seen below.

Structures		Name		Base Type		Offset Type		Offset Byte
Union	▶	V430_Input_Structure		STRUCT		User		
Enumerated	▶	V430_Output_Structure		STRUCT		User		

Adding Global Variables

Global variables need to be added for both the input and output modules.

Name		Data Type		Initial Value		AT		Retain	Constant	Network Publish
V430FInput		V430_Input_Structure				%3300		☐	☐	Do not publish
V430FOutput		V430_Output_Structure				%3200		☐	☐	Do not publish

In PROFINET configuration, publishing these variables is not required. In the **AT** field, notice that there is a % that refers to the CIO area. The number represents the **Start Addresses** used in **CX-ConfiguratorFDT** settings for **Input** and **Output**.

Area:	CIO
Start Address:	3300
Length:	100

Input

Area:	CIO
Start Address:	3200
Length:	100

Output

Verify PROFINET I/O Communication Status

Check the Connection Status

The PNT21 module should show **r.** on its display, with LEDs MS green, NS green, comm yellow, and 100 MB yellow if connected to 100 MB. Sysmac online status should be green, with a green ERR/ALRM indicator.

Check the Send and Receive Data

Add a watch and add the Input and Output modules.

Device name	I	Name
EngNJ	▶	V430FInput
EngNJ	▶	V430FOutput
EngNJ		*Input Name...*

Expand the **V430FOutput** module and look for the **Trigger** bit. Toggle this bit to **TRUE**, then to **FALSE**. The camera should trigger, causing the LEDs to flash.

User_Tag_8	False	TRUE	FALSE
Clear_Read_Cycle_Report_And_Counter	False	TRUE	FALSE
Unlatch_Outputs	False	TRUE	FALSE
Disable_Scanning	False	TRUE	FALSE
Trigger	False	TRUE	FALSE
New_Master	False	TRUE	FALSE
Out1	False	TRUE	FALSE

Find the **User_Tag_25** bit in the **Output Module** and set it to **TRUE**. Open the **V430FInput** module and verify that the **User_Tag_25** bit is now set to **TRUE**. Set it to **FALSE** in the **Output Module** and the **Input Module** should return to **FALSE**. This indicates that the controller was able to set a value to the **V430**, and then read the new value back from the V430. This proves that communication is taking place.

Output Module

EngNJ	▼ V430FOutput		
	User_Tag_25	True	TRUE FALSE

Input Verification

▼ V430FInput		
User_Tag_25_Echo	True	TRUE FALSE

Output Module

▼ V430FOutput		
User_Tag_25	False	TRUE FALSE

Input Verification

▼ V430FInput		
User_Tag_25_Echo	False	TRUE FALSE

Chapter **9** PROFINET I/O Operation Using Omron NJ Controller

A

APPENDIX A # Serial Commands

Serial commands can be sent via **TCP port**, **AutoVISION Terminal**, or **HyperTerminal**.

Important: The HAWK MV-4000 and F440-F Smart Cameras do not support focus commands.

Serial Commands

Serial Command Syntax

< > = Required argument. Replace appropriately.

For example:

-u <DB_User_name> becomes **-u av** where **av** replaces **DB_User_name**.

| = Mutually exclusive arguments. Choose one from the list.

{ } = Used with **|** to specify a list of choices for an argument.

[ ] = Optional parameter.

Important: Unless otherwise stated, commands will respond with **!OK** on success and **!ERROR** on failure.

AUTOCAL [-exp={0|1}] [-expval={60-100000}] [-gain={0|1}] [-gainval ={0-100}] [-focus={0|1}] [-focval{0-9999}]

Initiates camera calibration of gain, exposure, and focus. Each parameter is independent. Ranges are device-dependent.

-exp enable=1 or **disable=0** autocalibrate exposure.

-expval value of exposure in µs.

-gain enable=1 or **disable=0** autocalibrate gain.

-gainval value of gain in percentage.

-focus enable=1 or **disable=0** autocalibrate focus

-focval value of focus in mm.

Example 1:

Command: AUTOCAL

Response: 0;4632;134;50;300 (gain=0, exposure=4632 µs, focus=134, min allowable focus=50, max allowable focus=300)

Example 2:

Command: OFFLINE

Response: !OK

Command: QUERYAUTOCAL

Response: 0;4632;134;50;300 (Gain was 0.)

Command: AUTOCAL -exp=1 -gain=0 -gainval=18 (Fixed gain at 18%.)

Response: 18;3308;128;50;300 (Gain did not change and exposure has changed from 4632 µs to 3308 µs.)

Example 3:

Command: OFFLINE

Response: !OK

Command: QUERYAUTOCAL

Response: 0;3478;226;50;300 (Exposure was 3478 µs.)

Command: AUTOCAL -exp=0 -expval=1000 (Fixed exposure at 1000 µs.)

Response: 31;1000;98;50;300 (Exposure stayed at 1000 µs and gain has changed from 0% to 31%.)

Important: AUTOCAL only functions when the camera is OFFLINE.

GET {tagname|service|service.tagname}

Gets value of a global tag.

The tagname must correspond to one of the supported tags within the device. Use the **INFO** command to get a full list of tags and services, as well as attributes of the tag and list of subtags.

The command is terminated by a carriage return and/or line feed character.

Include an index to get a single value from an array such as **GET int1**. If the index is omitted, the full array of values will be returned in a comma-separated list of values.

Send **Get {tagname|service.tagname|service}** to get the value of a tag within the global data service. To get the value of a tag within another service, prefix the tagname with the service name. For example, a **GET <service.tagname>** command such as **GET eip.input** for the EIP input assembly.

The AVP service allows retrieval of step and datum information from the job tree using forward slash '/' in the symbolic name path. **GET avp/insp1/snapshot1/status** paths are not case-sensitive and do not need to be fully qualified if unique.

GET avp/snapshot1/status will return the same result if there is only one inspection.

When issued against a step, **GET avp/snapshot1** will return the values for all datums.

Success Return: On success will return the value stored in the tag.

For example: ABCD

Fail Return: On failure will return !ERROR followed by the reason for the failure.

For example: !ERROR Tag matchstring66 not found

Important: This command only functions when the camera is ONLINE.

For TCP Connection:
GETIMAGE [-format={jpg|png|tif|raw}] [-quality ={0-100}] [-woi=left,top,right,bottom] [-inspection=n]

For UART Connection:
GETIMAGE <-transfer=ymodem> [-format={jpg|png}] [-quality ={0-100}] [-woi=left,top,right,bottom] [-inspection=n]

Initiates serial transfer of inspection image.

Note: This command always returns the last (most recent) image.

-transfer=ymodem uses Ymodem protocol over the serial port. If the -transfer option is omitted completely, the transfer mode is over the TCP and Ethernet port.

Important: YModem transfer option is not supported on the HAWK MV-4000.

-format={jpg|png|raw|tif} specifies the format of the image. **RAW** and **TIF** are not supported over a UART Connection. If omitted, the image format is **JPG**.

Note: For monochrome cameras, the only formats available are TIF, PNG, and JPG. For color cameras, the only formats available are RAW, and PNG.

Note: All image file types return complete file information that can be saved directly to disk except the RAW file type, which requires explicit conversion.

-quality=*n* specifies a JPG compression quality of n less than or equal to 100. The default quality is 80 if not specified. This setting is only supported for the JPG file type.

Note: PNG, RAW, and TIF formats provide lossless image compression. If format is set to PNG, RAW, or TIF, the quality setting does not apply.

woi=left,top,right,bottom specifies a rectangular area of the image to be included in the output image. If omitted, the full image buffer is returned.

Note: -woi is only supported for TIF and JPG formats, as shown in the table below.

	Color Full Image	Color WOI	Mono Full Image	Mono WOI
PNG	Y	N	Y	N
RAW	Y	N	Y	N
TIF	Y	Y	Y	Y
JPG	Y	Y	Y	Y

-inspection=*n* specifies the inspection from which to retrieve an image. The image will be from the first snapshot within that inspection. If not specified, the image will be from the first inspection that contains a snapshot.

The following example will retrieve an image from the camera with these settings:
Protocol: ymodem; **Format:** png; **Quality:** N/A; **Inspection:** second inspection.

GETIMAGE –transfer=ymodem –format=png –inspection=2

The following example will retrieve an image from the camera with these settings:
Protocol: ymodem; Format: jpg (default); Quality: 50; Inspection: first inspection (default).

GETIMAGE –transfer=ymodem –quality=50

Important: This command only functions when the camera is ONLINE.

HELP

Returns a list of all serial commands showing correct syntax and functionality descriptions.

INFO [tagname|service]

Gets information about a tag or service.

INFO with no arguments gets a list of services.

INFO <service> gets a list of tags in that service.

INFO <service.tagname> gets attributes of the tag as well as a list of subtags.

The AVP service allows retrieval of step and datum information from the job tree using forward slash '/' in the symbolic name path. **INFO avp/insp1/snapshot1/status** paths are not case-sensitive and do not need to be fully qualified if unique.

INFO avp/snapshot1/status will return the same result if there is only one inspection.

When issued against a step, **INFO avp/snapshot1** returns properties of the step, a list of child datums, and a list of child steps. Child steps are indicated by a trailing forward slash.

JOBBOOT [-slot=]<n>

Sets bootup job slot *n* (RS-232 only).

JOBDELETE {[-slot=]n|-all}

Deletes job in slot *n*, or all jobs if **-all**.

Important: Does not delete the current job loaded in camera memory.

JOBDOWNLOAD <-transfer={ymodem|ftp}> [-size=value] [-c]

Important: JOBDOWNLOAD only supports FTP on the HAWK MV-4000.

Downloads a **.avz** job file via the specified transfer method (ymodem supported only over RS-232; FTP supported only over network connection).

The **ymodem transfer method** only requires that the user send the .avz file via the ymodem protocol over RS-232, and the job will load automatically after the transfer is complete.

The **FTP transfer method** requires the user to perform the following steps to load the job:

- **JOBDOWNLOAD: -transfer=ftp [-size=avpsizeinbytes]**

Pre-creates a fixed-size /streamd0 RAMdisk to receive the .avz over FTP. If size is omitted, the default RAM disk size is used to create /streamd0. The size of /streamd0 is limited to (available contiguous RAM – minimum target contiguous RAM) / 2.

- User FTPs the job to /streamd0
- **JOBLOAD: -mem -r**

Loads .avz from /streamd0 into RAM, deletes the RAMDisk /streamd0, and optionally starts the job (if -r is specified).

JOBINFO [[-slot=]n] [-v]

Gets job summary or info about slot *n*.

JOBINFO with no arguments returns a list of all jobs on the device.

-v = Verbose *n*. This option shows the amount of space that would be freed if the job were deleted. It also lists the total disk space and free disk space.

JOBLOAD {[-slot=]<n>|-mem} [-r]

Loads a job from slot *n* or from memory when used with the JOBDOWNLOAD command via FTP.

-r = Start inspections.

JOBSAVE [-slot=]<n>

Saves current job to slot *n.*

MEMAVAIL [-cp]

Returns available memory for device or coprocessor.

MEMCONTIG [-cp]

Returns maximum memory block for device or coprocessor.

MEMFRAGS [-cp]

Returns memory fragments for device or coprocessor.

Important: MEMFRAGS is not supported by the HAWK MV-4000. It will return **!ERROR**.

MEMINFO [-cp] [-v]

Returns memory summary "avail/contig/frags" for device or coprocessor. Verbose.

OFFLINE

Stops all inspections.

ONLINE

Starts all inspections.

ONLINE? [-insp=n]

Queries if each inspection on the camera is online. Defaults to all inspections if no inspection is specified. If the camera is running in a multi-inspection job, this command will return **!1** if all inspections are online and **!0** otherwise.

-insp=n specifies the inspection to query if it is online.

QUERYAUTOCAL

Returns photometry settings: Gain, Exposure, and Focus.

QUERYFOCUSUNITS

Queries the units being used for autofocus, mm (0) or inches (1).

QUERYWHITEBAL

Returns white balance settings: RED gain, BLUE gain, and GREEN gain.

QUICKFOCUS [x] [y]

Performs an autofocus by analyzing the area around the point specified by **x** and **y**.

The response is in the format of the camera's current focus, min. allowable focus on the camera, max. allowable focus on the camera.

Example:

Perform a quick focus on point (640,480) in the image.

Command: QUICKFOCUS 640 480

Response: 124;50;300 (Current focus is set to 124 mm with an allowable focus range of 50 – 300 mm on the current camera.)

Important: This command only functions when the camera is OFFLINE.

READY? [-insp=n]

Queries if inspection is waiting for a trigger. **!1** if all inspections are ready or **!0** if not all inspections are ready.

-insp=n specifies the inspection to query if it is ready.

REBOOT [-noload]

Reboots the device.

-noload = do not load BOOT job.

RESTOREWBAL

Restores preset white balance parameters: RED gain, BLUE gain, and GREEN gain.

SET <tagname> <value>

Sets value of a global tag.

The tagname must correspond to one of the supported tags within the device. Use the **INFO** command to get a full list of tags and services, as well as attributes of the tag and list of subtags.

The value can contain spaces.

The command is terminated by a carriage return and/or line feed character.

The value can be a list of comma-separated items to set a sequence of tags:

Send **SET int1 1, 2, 3** to set int1 = 1, int2 = 2, int3 = 3.

The AVP service allows setting of step and datum information from the job tree using forward slash '/' in the symbolic name path. **SET avp/insp1/snapshot1/acq1/gain 2.0** paths are not case-sensitive and do not need to be fully qualified if unique.

SET avp/acq1/gain 2.0 will set the same gain value if there is only one acquire.

Control tags in the AVP service such as **START**, **STOP**, and **TRIGGER** act as momentary switches. **SET avp.start 1** is equivalent to the **ONLINE** command. **avp.start** will reset immediately and always read as **0**.

Success Return: On success will return **!OK** followed by an echo of the command.

For example:

!OK SET matchstring1

Fail Return: On failure will return **!ERROR** followed by the reason for the failure.

For example:

!ERROR Tag matchstring66 not found

SETFOCUSUNITS

Sets units used for autofocus, **mm (0)** or **inches (1)**.

Important: The MicroHAWK MV-40 only supports mm so SETFOCUSUNITS will only accept **0** and anything else will respond with **!ERROR**.

TARGET {0|1|off|on}

Turns targeting LEDs On or Off.

target 1 = Turn Target On

target 0 = Turn Target Off

TRIGGER

Triggers an inspection.

VERSION

Returns Visionscape software version.

vt [n]

Triggers an inspection by pulsing a Virtual I/O point.

For example: **vt 1**

will return pulse **VIO1**. The inspection will run if it is configured to use **VIO 1** as a trigger.

If specified, the VIO index must be in the allowed range for Virtual I/O points within Visionscape. The virtual I/O line will be set high then low.

If VIO Index is not specified, VIO1 is assumed.

Fail Return: Return **!ERROR** followed by the reason for the failure.

For example: **!ERROR No such trigger** when the index specified '**n**' is out of range of virtual triggers.

WHITEBAL

Performs automatic calibration of white balance settings: RED gain, BLUE gain, and GREEN gain.

Important: This command only functions when the camera is OFFLINE.

B

APPENDIX B WinPcap License

The HAWK MV-4000 Smart Camera uses **WinPcap** for PROFINET implementation. This section contains the WinPcap license, which can also be found at:

https://www.winpcap.org/misc/copyright.htm

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the University of California, Berkeley and its contributors."

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 1983 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Portions Copyright (c) 1995, 1996, 1997 Kungliga Tekniska Högskolan (Royal Institute of Technology, Stockholm, Sweden). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by the Kungliga Tekniska Högskolan and its contributors."

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 1997 Yen Yen Lim and North Dakota State University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes software developed by Yen Yen Lim and North Dakota State University."
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 1993 by Digital Equipment Corporation.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies, and that the name of Digital Equipment Corporation not be used in advertising or publicity pertaining to distribution of the document or software without specific, written prior permission.

THE SOFTWARE IS PROVIDED "AS-IS" AND DIGITAL EQUIPMENT CORP. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions Copyright (C) 1995, 1996, 1997, 1998, and 1999 WIDE Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright (c) 1996 Juniper Networks, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that: (1) source code distributions retain the above copyright notice and this paragraph in its entirety, (2) distributions including binary code include the above copyright notice and this paragraph in its entirety in the documentation or other materials provided with the distribution. The name of Juniper Networks may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Appendix **B**   WinPcap License

Portions Copyright (c) 2001 Daniel Hartmeier All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTOR "AS-IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright 1989 by Carnegie Mellon.

Permission to use, copy, modify, and distribute this program for any purpose and without fee is hereby granted, provided that this copyright and permission notice appear on all copies and supporting documentation, the name of Carnegie Mellon not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that copying and distribution is by permission of Carnegie Mellon and Stanford University. Carnegie Mellon makes no representations about the suitability of this software for any purpose. It is provided "as-is" without express or implied warranty.