COGNEX

SLX WebUI

Reference Manual

2025 October 17
Revision: 25.2.0.45

Table of Contents

Legal Notices

About the SLX WebUI

The SLX WebUI provides an intuitive interface for managing your SLX device. The WebUI simplifies configuration and monitoring tasks.

The WebUI has the following key features:

- Fast setup with real-time feedback: SLX devices deploy in minutes. You can explore how different options affect the device in real time.
- Live display and results: View real-time images of codes read by the device and decode statistics.
- No additional software required: The WebUI only requires a supported modern web browser:
    - Google Chrome
    - Microsoft Edge
    - Apple Safari
    - Mozilla Firefox
    - Opera

The following SLX device models are compatible with WebUI:

- SLX-280D
- SLX-290

For more information about hardware features, see the *Reference Manual* of your SLX device.

Connecting to a SLX Device

The following topics describe how to connect to a SLX device and create a daisy chain.

Connecting to an SLX-280D Device

The SLX-280D ships with the following network configuration:

Port	Network Configuration
Network port 1	DHCP
Network port 2	DHCP
USB-C port	Static IP address 192.168.111.2 (subnet 255.255.255.0)

Due to this default configuration, connect to your device through direct USB-C connection for first time communication. After the first connection, you can configure and use the two network ports for any future connection.

Note: The following procedure applies to establishing connection using a PC running Windows 11.

Required equipment

You need the following equipment to first connect to one or more devices:

- Cognex X-coded Ethernet cable: M12 X-coded female to RJ-45
- USB-C cable

Note:

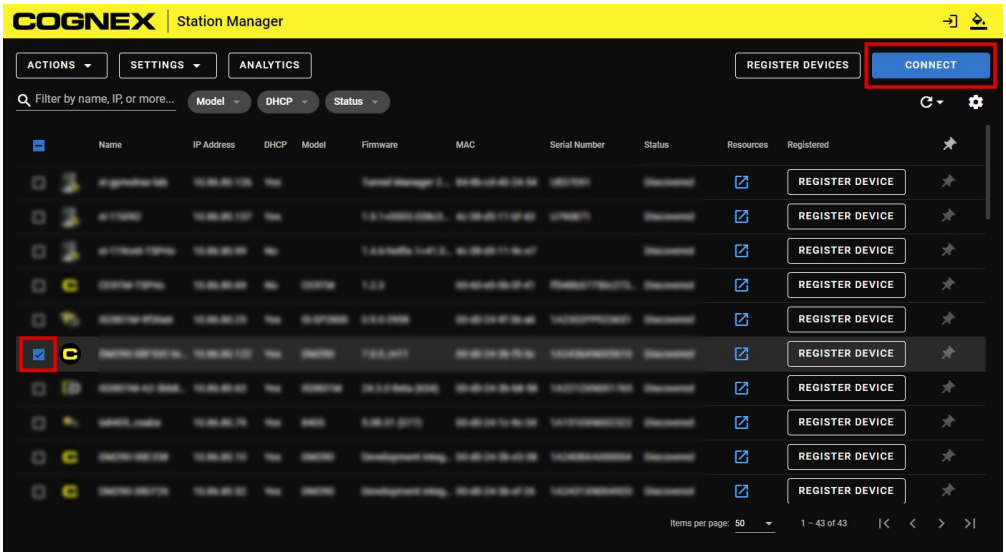
- Choose USB-C to USB-A or USB-C to USB-C depending on the ports available on the configuration PC.
- This cable is not provided by Cognex.

Connect to Station Manager

To connect to the Station Manager:

1. Go to support.cognex.com/en/downloads/edge-intelligence and select Cognex Station Manager.
2. Download the Station Manager installer and follow the on-screen steps.
3. Find and select your device.

4. Click **Connect** and set up the device.

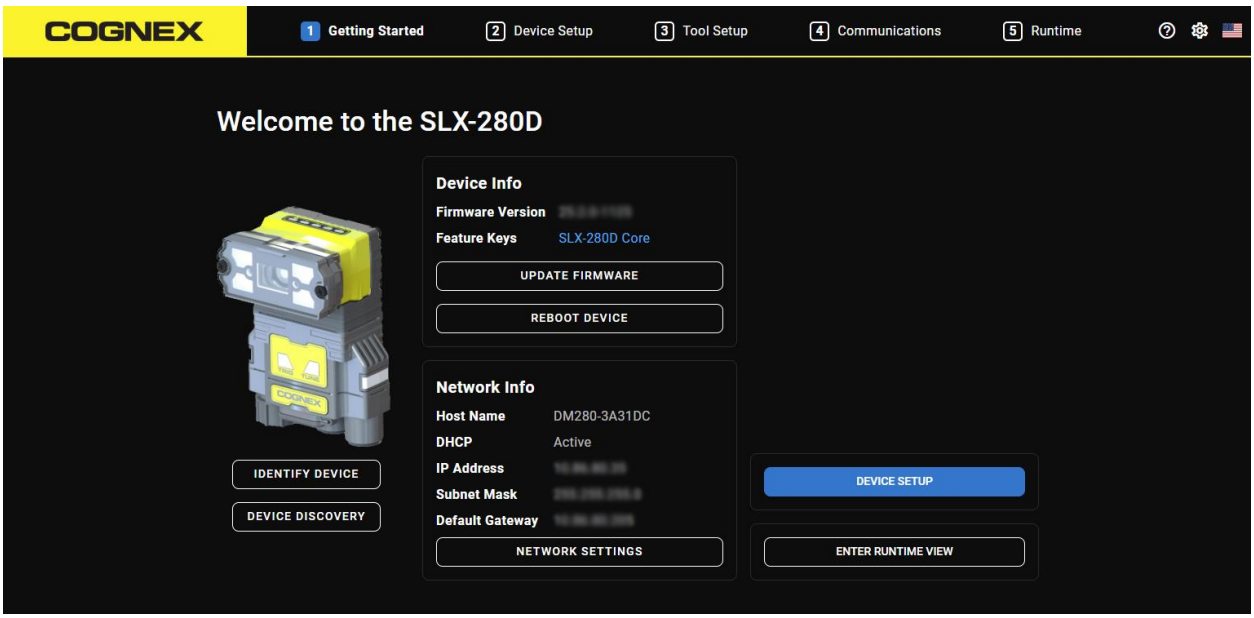


Establishing Initial USB-C Connection and Configuring the Network Ports

To establish connection and configure the network ports:

1. Switch on the device.
2. Connect the device to a USB port on your configuration PC using the USB-C cable.
3. Once connected, the device registers as network device on your PC and creates a new network adapter. Take note of the new adapter. You will need to configure it in the next step.
4. Locate the network adapter from Step 3 and configure it with a compatible IP setting, for example 192.168.111.1.
5. To view the landing page of the device in the web UI, navigate to http://192.168.111.2 using a web browser.

The following screenshot shows the **Getting Started** step. You can navigate to this page any time by clicking the **Getting Started** step at the top of the web UI.



Adding the SLX-280D Device to an Ethernet network

To add the device to a network connected to a DHCP server:

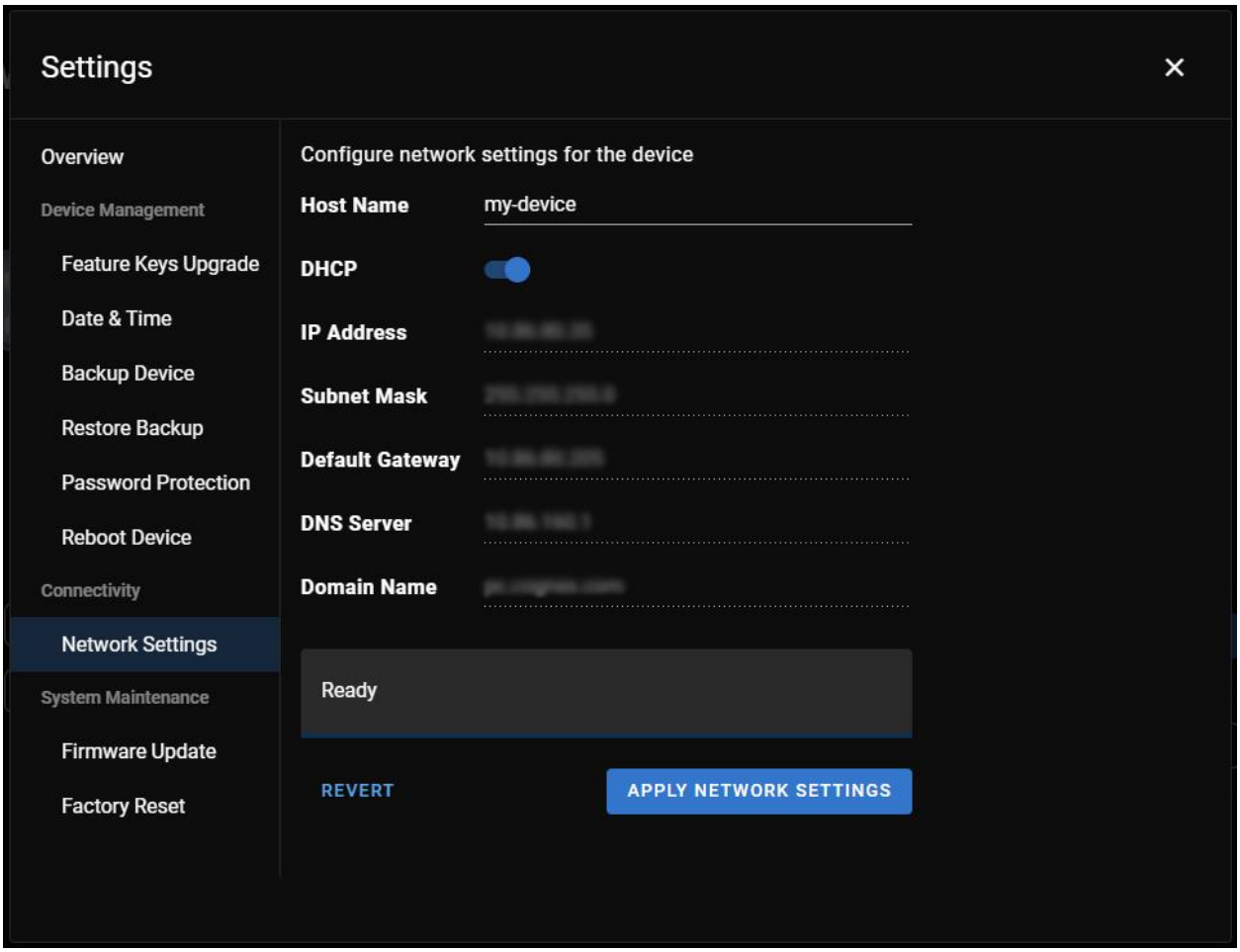
1. Connect device to your network using the X-coded Ethernet cable.
2. The device automatically receives an IP address. The **Network Info** section of theWebUI updates with the new IP address of the device.

To add the device to a network not connected to a DHCP server:

1. Connect device to your network using the X-coded Ethernet cable.
2. The device obtains a random link-local IP address (169.254.xxx.xxx).

   Note: Depending on your IT policy, you may need to be on the 169.254.xxx.xxx subnet to connect to the device.

3. To modify this, click **Network Settings**. Set the desired IP settings in the dialog box that pops up, then click **Apply Network Settings**.



Creating an SLX-280D Daisy Chain

Daisy chaining is a method of connecting multiple devices in a sequence. Each device connects to the next device, allowing data to flow through the chain without needing a network switch.

Creating a Daisy Chain

To create a daisy chain of devices:

1. Configure the first device to operate on the network through one of its network ports.
2. Connect a second device to the second network port of the first device.

   Note: The order of the ports used does not matter when creating a daisy chain.

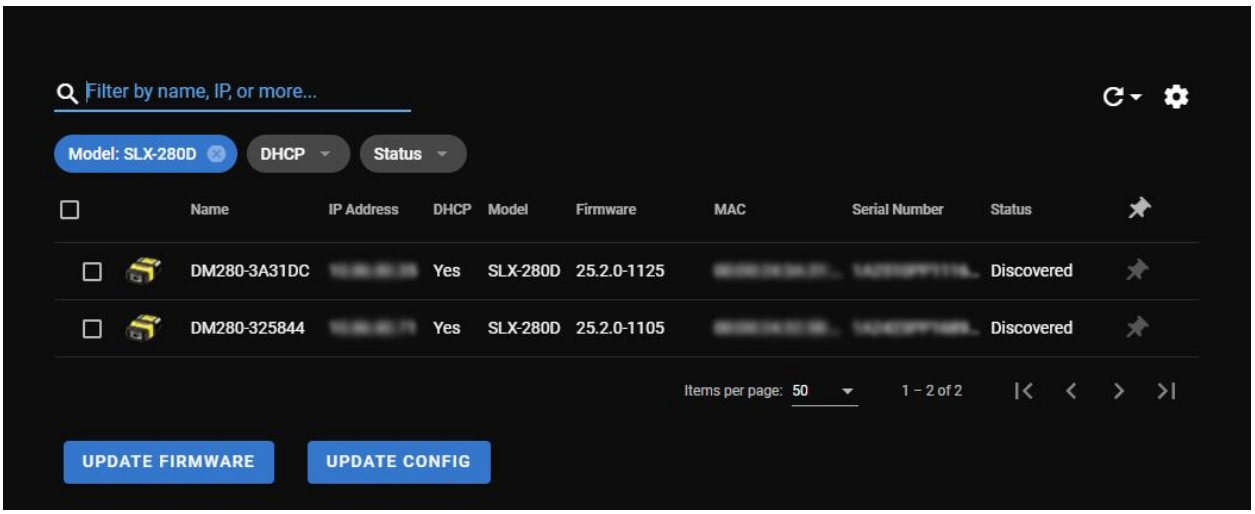
3. Repeat this process to create a daisy chain of devices.

Discovering Devices in a Daisy Chain

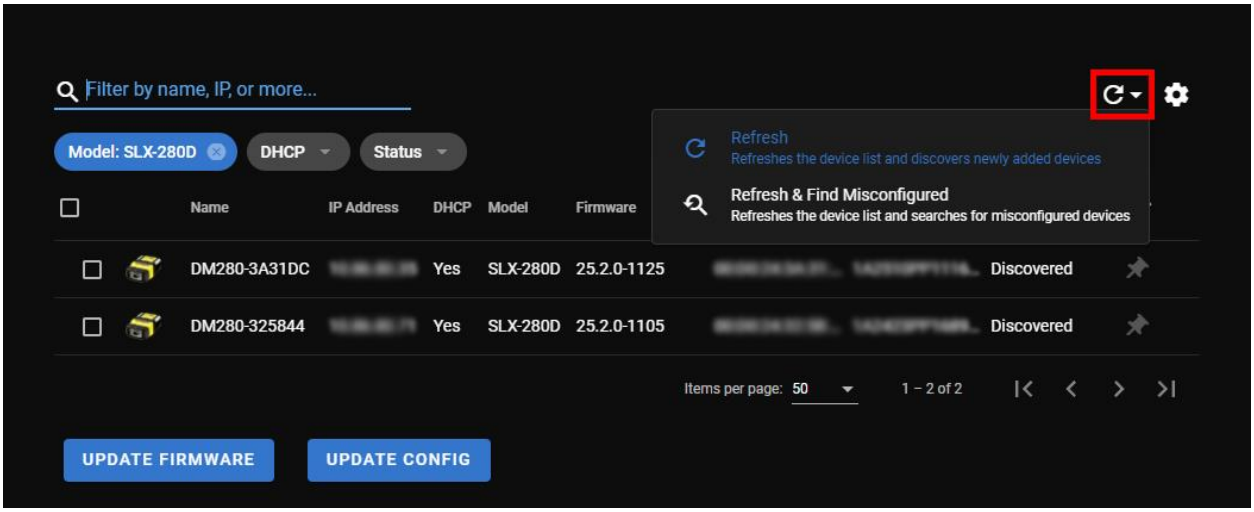
Subsequent devices in the daisy chain also require network configuration using device discovery in theWebUI.

To discover devices in a daisy chain:

1. From the WebUI of the first device, go to the **Getting Started** step. You can navigate to this page any time by clicking the **Getting Started** step at the top of the web UI.
2. Click **Device Discovery** to launch a dialog that displays a list of devices that are connected to the same network.



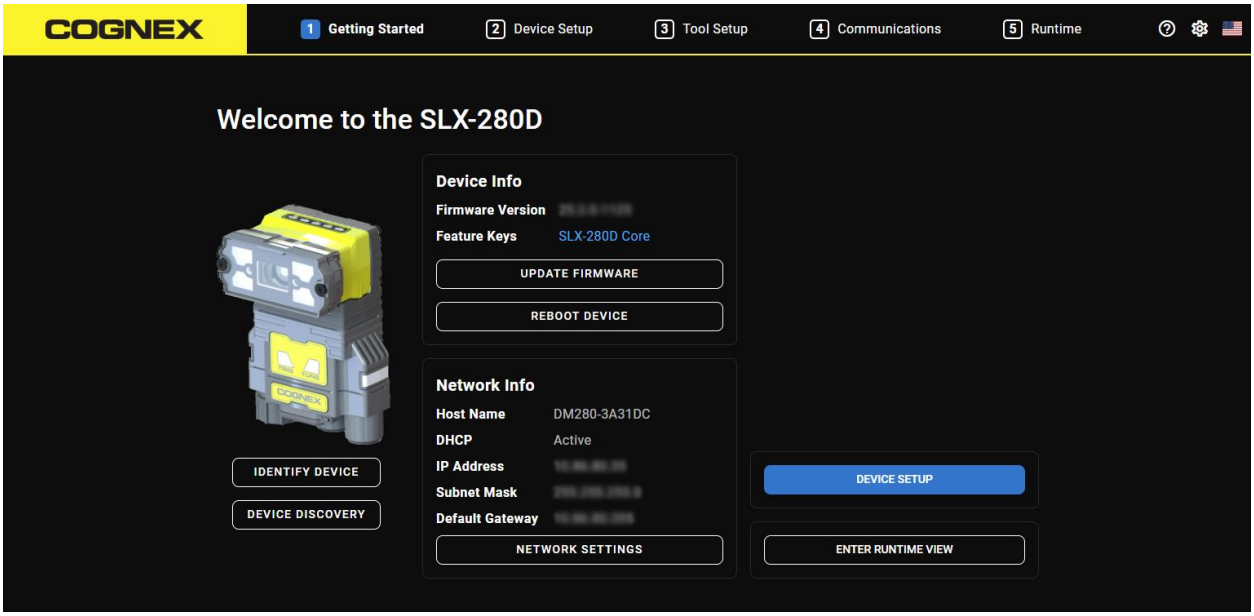
3. If a connected device is missing from the list, click the refresh button in the top right corner of the dialog and select the **Refresh & Find Misconfigured** option.



Application Steps

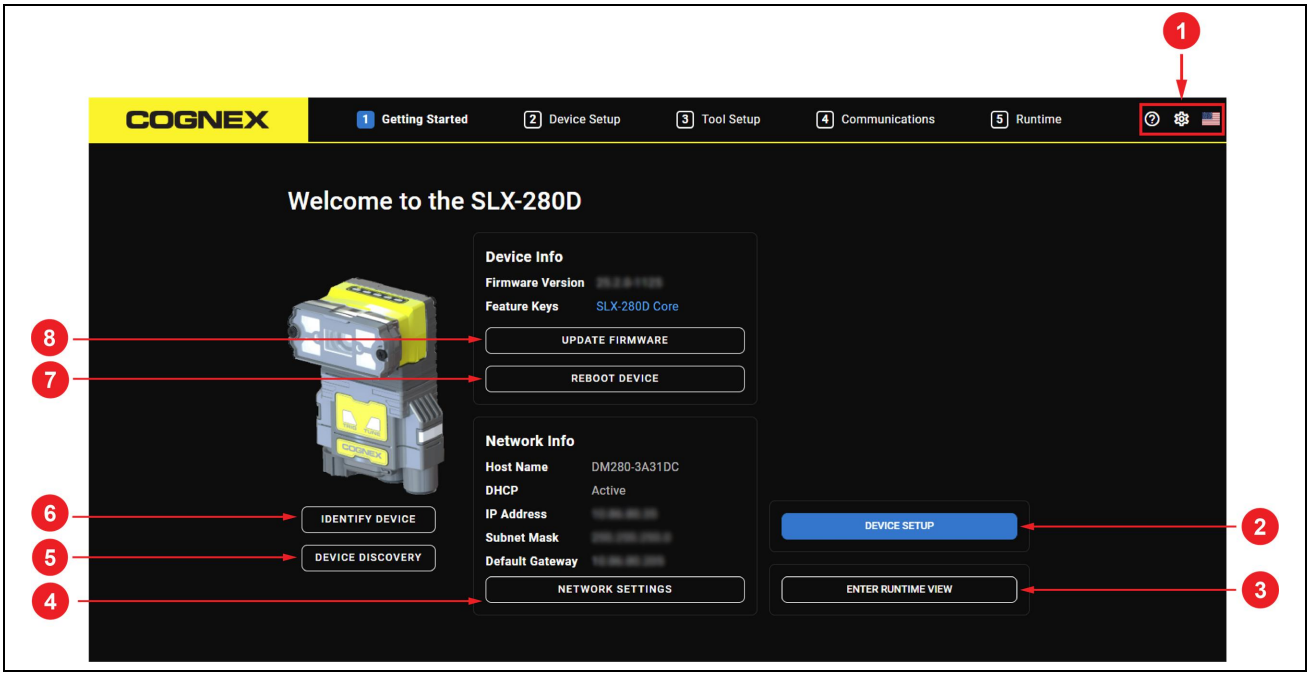
The application steps streamline the setup and management of your device. The application steps are displayed at the top of the web UI. You can navigate between the application steps any time by clicking on one of the steps.

The following screenshot shows the **Getting Started** step with the five application steps displayed horizontally on the top of the web UI.



Getting Started

The **Getting Started** step provides an overview of your device and serves as a central hub for device management and configuration.



Number	Description
1	Additional settings are available in this toolbar, located in the top-right of the WebUI: 1. Help: Documentation links 2. Settings: Displays the settings pop-up window, providing access to the following configuration features: • Network Settings • Restore Backup • Backup Device • Reboot Device • Firmware Update • Factory Reset 3. Language: Allows the user to switch the language of the WebUI (default is English). 4. Theme: Allows the user to switch between dark and light themes (default is dark theme).
2	**Device Setup**: Transitions the WebUI to the Device Setup step.
3	**Enter Runtime View**: Transitions the WebUI to the Runtime step, if the device has already been configured.
4	**Network Settings**: Launches a dialog that allows modification of the device network configuration.
5	**Device Discovery**: Launches a dialog that displays a list of devices that are co-located on the same network as the current device. For more information, see *Creating an SLX-280D Daisy Chain* on page 8.
6	**Identify Device**: Flashes the LEDs on the devices to assist with visual identification.
7	**Reboot Device**: Launches a dialog that allows a remote reboot of the device.
8	**Update Firmware**: Launches a dialog that allows updating of the device firmware.

Device Discovery

Open the utility and browse the list to find the device that you are looking for:

- Filter the device list by DHCP, status, model type, or begin typing an IP address or host name.
- Refresh the list or enable the **Discover Misconfigured Devices** option to find devices with static IP address that do not reside on the same subnet.

- Open the Settings panel with the gear icon to add common subnets that you can enable or disable for advanced device discovery queries.

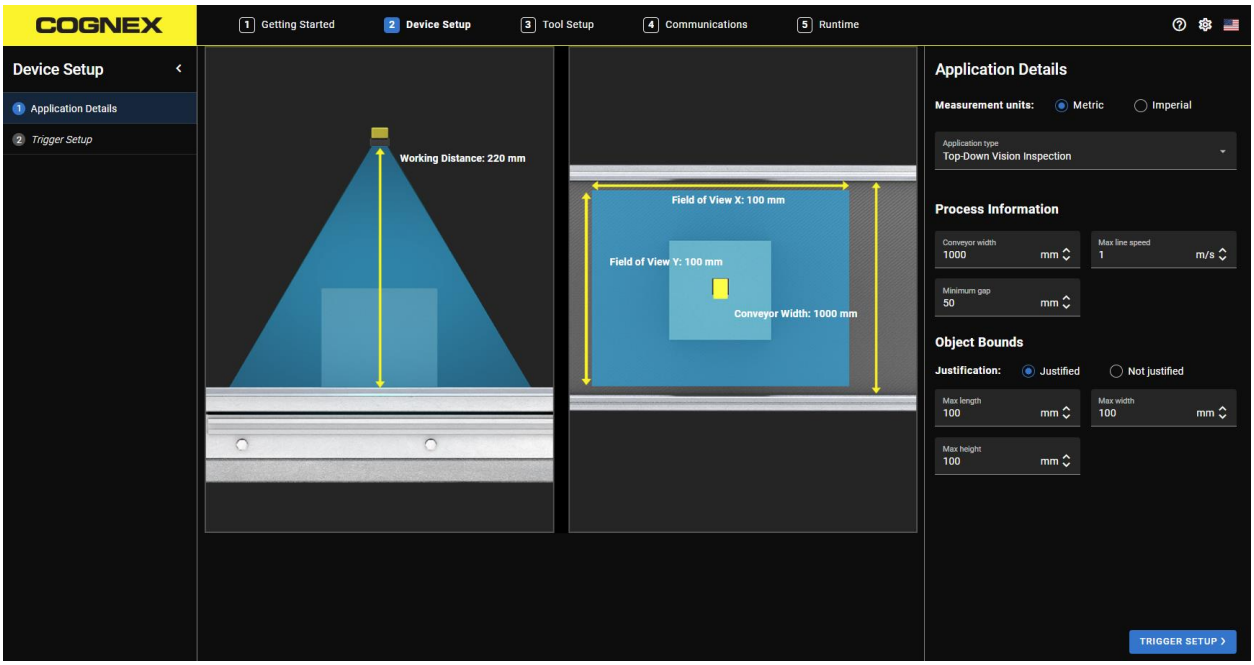
You can update the firmware or configuration on one or more devices.

Device Setup

The **Device Setup** step allows you to prepare the integration of the device into your production environment. This application step guides you through aligning your device with the object you want to inspect and setting up the trigger.

Application Details

The **Application Details** substep allows you to verify the device mounting position for your use case.



To get a mounting recommendation:

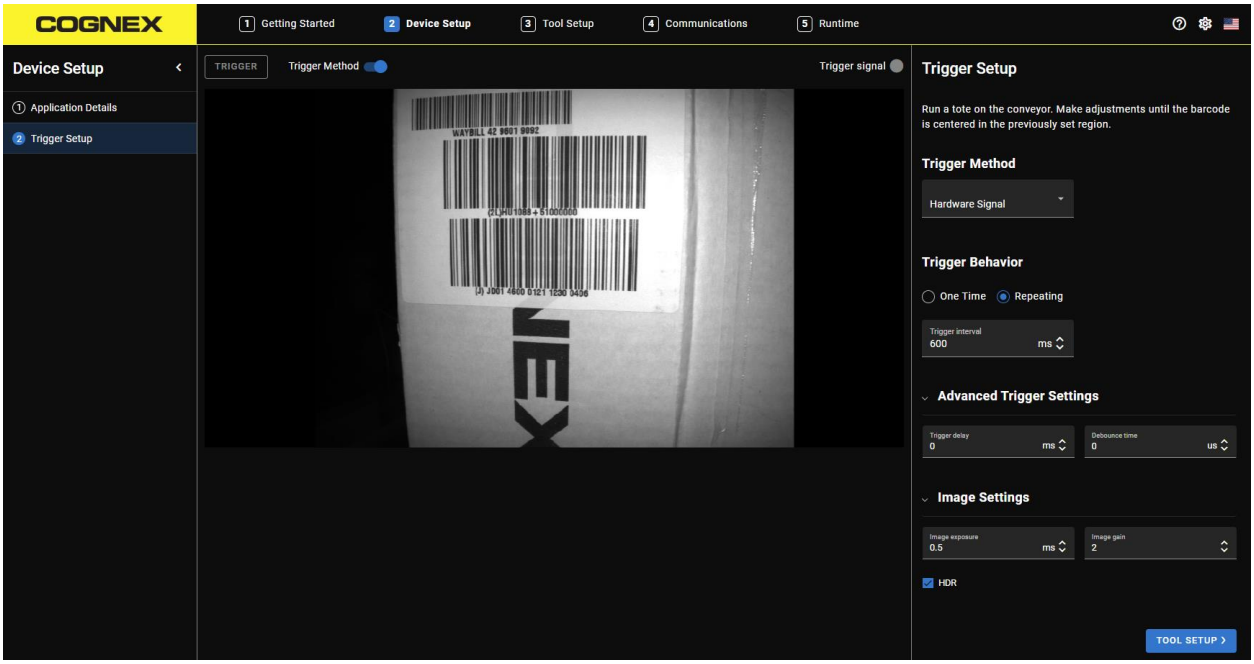
1. Choose a measurement unit between metric and imperial.
2. Select the **Application type** that corresponds to your use case.

   Note: Only hybrid device models can perform the combined barcode reading and vision inspection application types.

3. Enter the conveyor data in the **Process Information** fields.
4. Depending on the selected **Application type**, provide the following information:
   a. **Code Information** for use cases involving barcode reading. Configure 1D or 2D code information, or select **Both** to configure both.
   b. **Object Bounds** for vision inspection uses cases.
5. Observe the mounting distance displayed on the graphics. Cognex recommends to use these values if your production environment allows.

Trigger Setup

The **Trigger Setup** substep allows you to configure the trigger source of the device and to fine-tune the triggering behavior. It also provides image acquisition settings.



The **Trigger Setup** substep provides the following configuration options:

Setting	Description
Trigger Method	To trigger the device manually, disable the **Trigger Methods** toggle and click the **Trigger** button. To trigger the device automatically, configure one of the following **Trigger Methods**: • **Hardware Signal**: the device acquires images when it receives a signal from a photoeye or an external sensor on input line 0. • **Software Command**: the device acquires images when it receives a trigger command through an industrial protocol from a PLC or TCP Client Connection. • **Internal Timer**: the device automatically acquires images at a configurable interval. This enables continuous inspection or code reading without an external signal.
Trigger Behavior	The **Hardware Signal** and **Software Command** trigger modes support the following trigger behaviors: • **One Time**: The device acquires and processes one image for each trigger using the currently enabled tools. The device relies on an external trigger source. • **Repeating**: The device acquires and processes images continuously after one trigger. The device stops image acquisition when a tool returns a result or the trigger signal becomes inactive. **Note**: If you enable the Barcode Reading and Multi-Class Classifier tools at the same time, the device automatically switches to One Time mode.
Communication Protocols	Only applicable with the **Software Command** trigger method. Choose between **TCP Server** and **Industrial Protocols** in the Trigger Select drop-down. For more information, see *Protocols* on page 19.

Setting	Description
Advanced Trigger Settings	Under the **Advanced Trigger** settings, you can configure optional delay and debounce behavior: • **Trigger delay**: specifies how many milliseconds to wait before starting an image acquisition once the device receives a trigger. Using a delay can be useful when you know the device receives a trigger before the tote is completely under the Field of View. • **Debounce time**: defines how long the trigger signal must be detected to be recognized as valid. Use a shorter value to compensate for ESD line noise and higher values to compensate for noise in electromechanical relays. You might have to experiment with the setting to choose the most appropriate value for your production environment.
Image Settings	Configure the following acquisition options under the **Image Settings** dropdown menu: • **Image exposure**: the time the imager of device uses to acquire each image. • **Image gain**: the level of brightness in each image. • **HDR**: High Dynamic Range functionality uses the CMOS image sensor technology to enhance image quality and contrast.

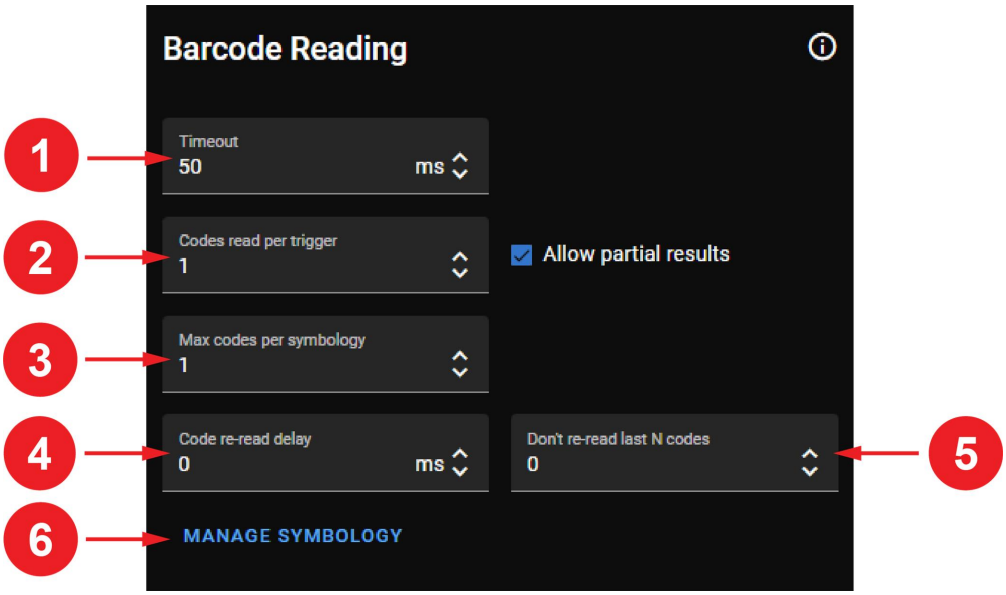
Tool Setup

The **Tool Setup** step allows you to configure the vision tools used in your application. You can use this step to test your device before deployment.

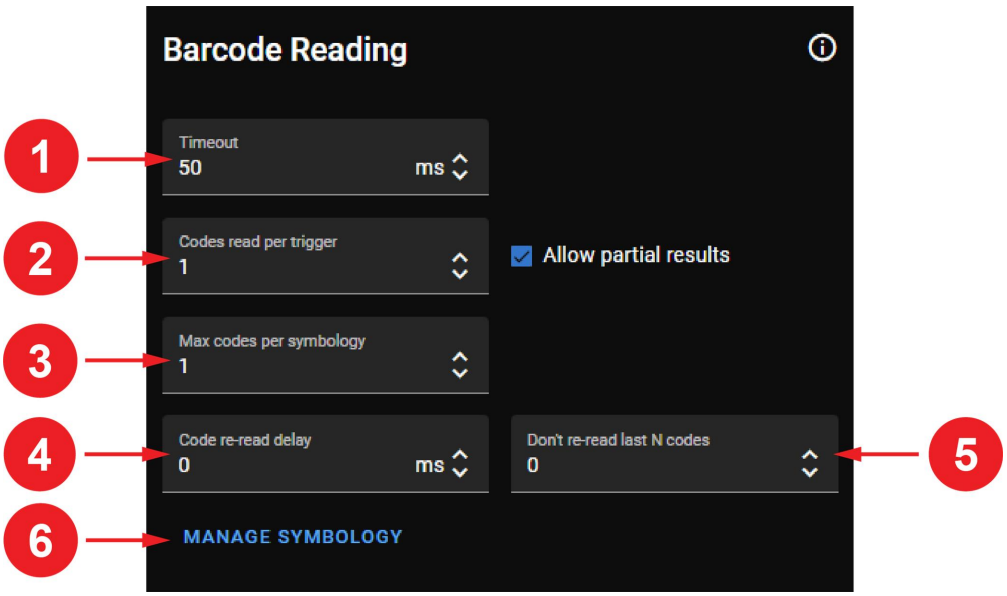
Barcode Reading

The **Barcode Reading** tool allows you to configure barcode reading options.

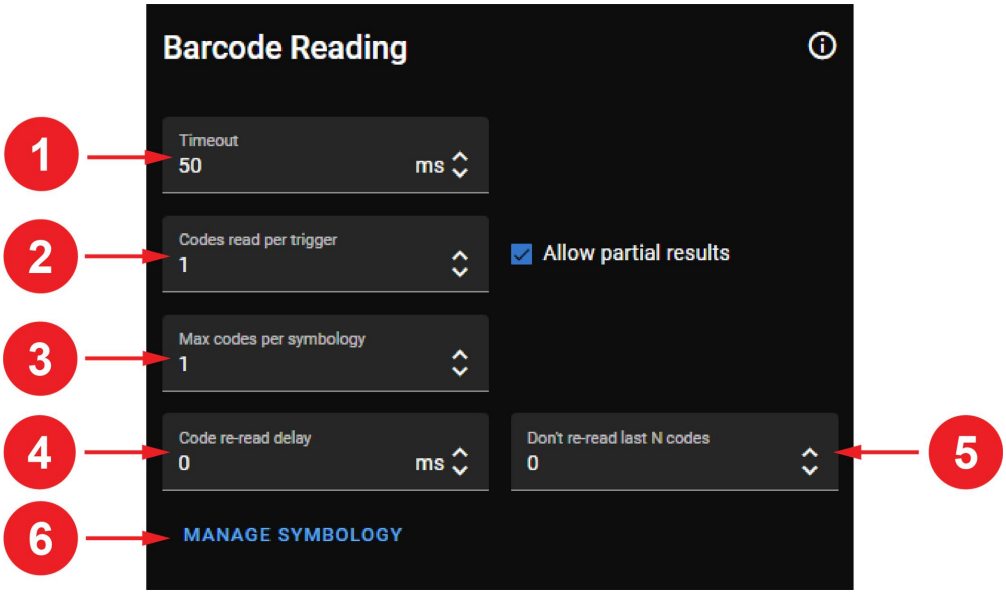
Set your Region of Interest (ROI) by resizing and rotating the rectangle. To configure the reading, set the following parameters:



Number	Description



1	The **Timeout** value specifies a fixed amount of time the device attempts to decode any symbol it detects. If the device is unable to decode the symbol within this timeout period, it returns a NoRead status as if the image contained no detectable symbol. Performance can decrease if you do not specify a timeout, as the device attempts to decode any code candidates in the current image before moving on to the next. The default value is 50 milliseconds.
2	The **Codes read per trigger** value specifies the number of codes the device has to read during a trigger. Enable the **Allow partial results** checkbox to make the device report a Good Read even if it could not find all the codes set up under the **Number of codes**.
3	The **Max code per symbology** value specifies the maximum number of each code type the device has to read during a trigger. For example, if you enable Data Matrix code and set the **Codes read per trigger** to two and the **Max code per symbology** is set to one, the device still reads one code.
4	The **Code Re-Read Delay** option lets you specify the delay between attempting to read the same code again. Specifying a non-zero value can help prevent data from overwhelming an external device connected to your SLX device. If your device is in One Time trigger mode, you can configure a re-read delay to avoid repeated outputs for the same code until it exits the Field of View. If the same code is presented before this time runs out, the device returns a NoRead.
5	The **Don't Reread Last N Codes** option prevents the device from reading codes that were read in the last N number of reads. You have to provide N, otherwise, if the number is 0, the option is disabled. The option applies to all read codes within one trigger sequence.



6	Click the **Manage Symbology** button to specify the barcode symbologies the device must look for. The available selection of 1D or 2D codes is determined by the **Code Information** setting in **Application Details** substep. Disabling unused symbologies can improve the performance of your device. The list of available 1D codes: • Code 128 • Code 25 • Code 39 • Code 93 • Codabar • Interleaved 2 of 5 • MSI • UPC/EAN The list of available 2D codes: • Data Matrix • DotCode • QR Code

Multi-Class Classifier

The **Multi-Class Classifier** tool detects the presence of items and process issues, and sorts objects into a maximum of ten classes. You can customize these classes any time during the training.

Set your Region of Interest (ROI) by resizing and rotating the rectangle. Once you train a class, the ROI is locked.

The Classification Score below the image represents the confidence the device has in correctly labeling the images. This is not the accuracy value, rather, the confidence in the prediction. The confidence tends to improve as you label more images.

Procedure

To train the Multi-Class Classifier, do the following:

1. Plan what classes you need.
2. Create the classes:
    a. Click **Add New** to create a new class.
    b. Click **Edit** to manage your classes. In the **Edit Classes** dialog, you can add classes, change the color of the labeling buttons, or rename the classes.
3. Label minimum two objects for each class. The **Confidence Indicator** result shows how confident the device is about the assigned label.
4. At this point, the device assigns a label for new objects.
5. Label more objects to improve the performance of the device.
6. Continue until you are satisfied with the results.

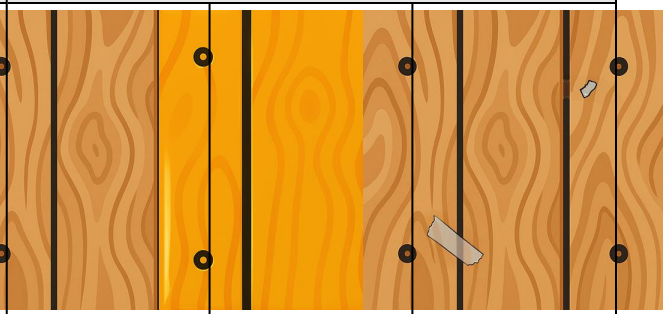
Best Practices

The following best practices can help you train the Multi-Class Classifier:

- Acquire images that resemble your production environment.
- Use objects that occupy at least 15% of the Field of View (FoV).
- Label a diverse set of objects into different classes, for example:
    - Capture objects of different sizes, different colors, different places within the FoV.
    - Add different backgrounds, if applicable, such as classification within a tray or tote.
    - Present totes with different levels of damage in the background.
- Verify that the device correctly applies labels for new objects.

Examples for Training Images

The following table shows examples images for detecting the presence or absence of an object with the corresponding classes: empty or occupied trays. The empty row shows an empty tray, meaning the absence of an object. The occupied row shows objects on the tray, meaning the presence of an object:

Class	Poor Training Set	Better Training Set
En y T		

Class	Poor Training Set	Better Training Set
Oc pie Tra		

A better training set captures variation in:

- Tray condition: the tray can have a different color or some debris left ton it
- Item type, color, and position, instead of just different angles
- Make sure to include various items if you have them in your production

Troubleshooting
If you are not satisfied with the performance of the tool, consider the following:

- The confidence sometimes stays low when the images in the classes are very different. However, the device might still be able to sort the images into the correct class.
- If the device does not recognize an image, it labels the image as Unclassified. This is normal behavior.
- If the confidence is still low after labeling many images, try a different tool that could be better suited to the job.
- Click **Labeled Images** to open the Labeled Images dialog and filter, delete, or reassign labels as needed.

After Training

To check how well the device performs in practice, click **Trigger** or turn on your conveyor belt and see the results in the **Runtime** substep. The training is complete when the device consistently and correctly labels new images with two or three bars.

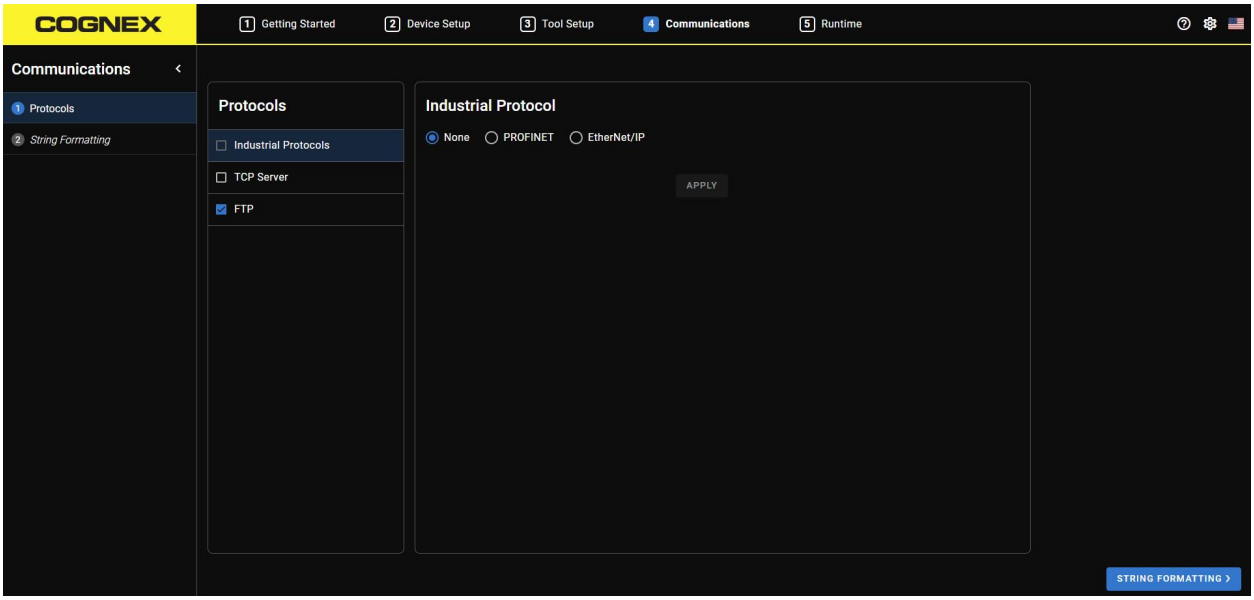
You can return to this substep at any time to improve the training with more images.

Communications

The **Communications** step includes options to configure device communications.

Protocols

The **Protocols** substep allows you to set up different communications protocols.



The device supports using multiple protocols at the same time. To enable a protocol, check the checkbox of the protocol.

Industrial Protocols

Industrial protocols enable the device to exchange I/O data, alarms, and diagnostics with a PLC. Configure the industrial protocol you want to use:

Protocol	PLC
PROFINET	Siemens S7
EtherNet/IP	Rockwell ControlLogix

To select the industrial protocol, click the radio button. To revert the settings to the previous state, click **Revert**.

Note: When changing industrial protocol settings, you have to restart the device for the changes to take effect.

For more information, see *Industrial Communications* on page 32.

TCP Server

Your SLX device can act as a TCP server. Provide the **Connection Port** to which TCP clients can connect.

FTP

The FTP function allows the device to transfer result images to an FTP server.

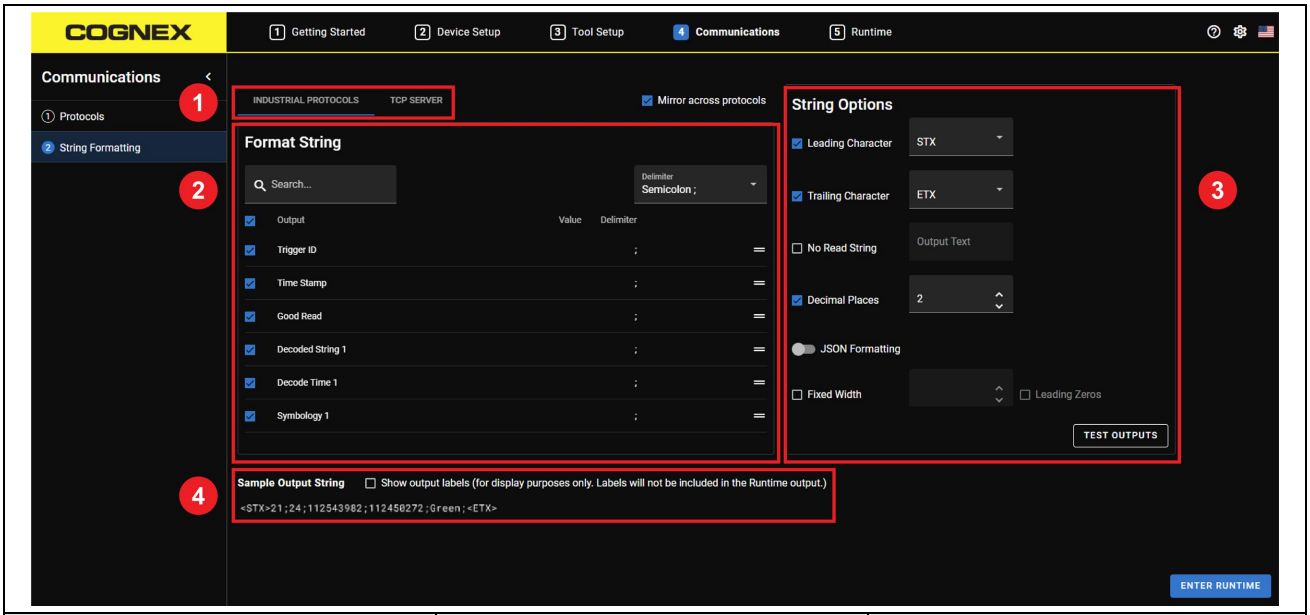
To configure the **FTP Settings**:

- **Server Address**: the address and port number of the FTP server.
- **File Path**: the a path in relation to the root directory of the FTP server, for example, */failed_images/image*.
- **Username**: the name of the user that the device can use to access the FTP server.
- **Password**: the password for the user that the device can use to access the FTP server.
- **File Name**: the file name for each result image. The device automatically appends a time stamp to the end of the file name.
- **File Type**: the file format of the result images.

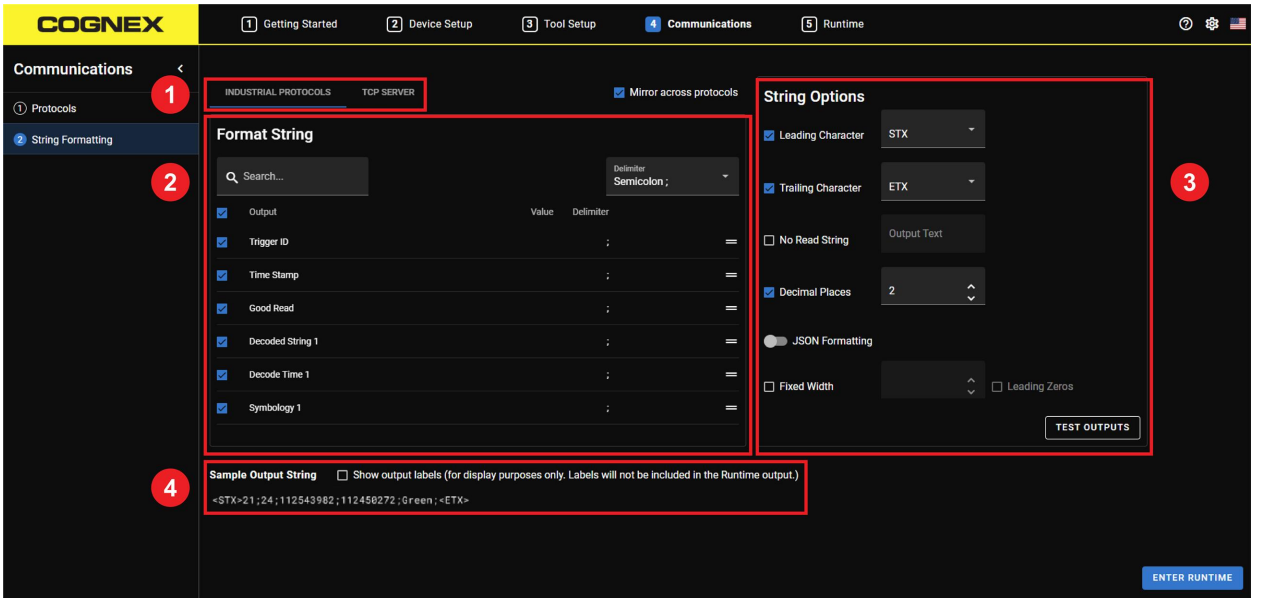
Click **Apply** to save your settings.

String Formatting

The **String Formatting** substep allows you to customize the output string that the device sends in response to a trigger.

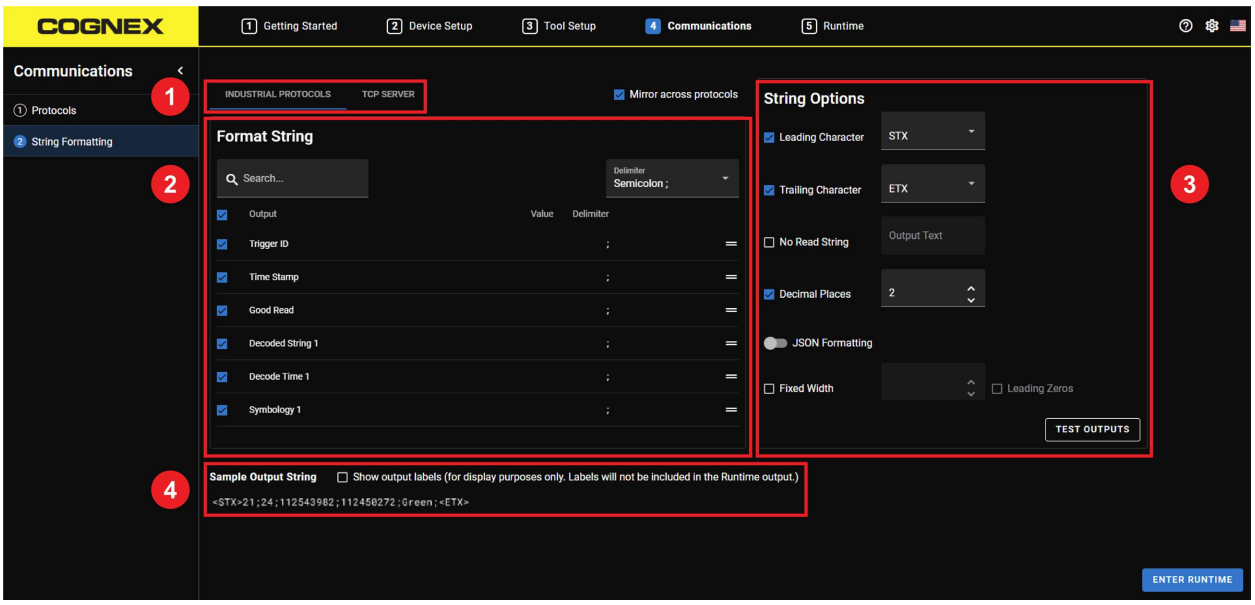


Number	Option Group	Settings
1	Protocol	Configure the output string for each individual protocol that the device uses. Check the **Mirror across protocols** checkbox to use the same settings for all protocols.
2	**Format String**	The rows display the available data results gathered from the tools. Check the checkboxes next to the data results to include them in the output string. To change the delimiter, select the delimiter character from the **Delimiter** dropdown. To use a substring or a script to create a custom string, click **Custom Output**.



Number	Option Group	Settings
3	**String Options**	You can specify the format of the output string:

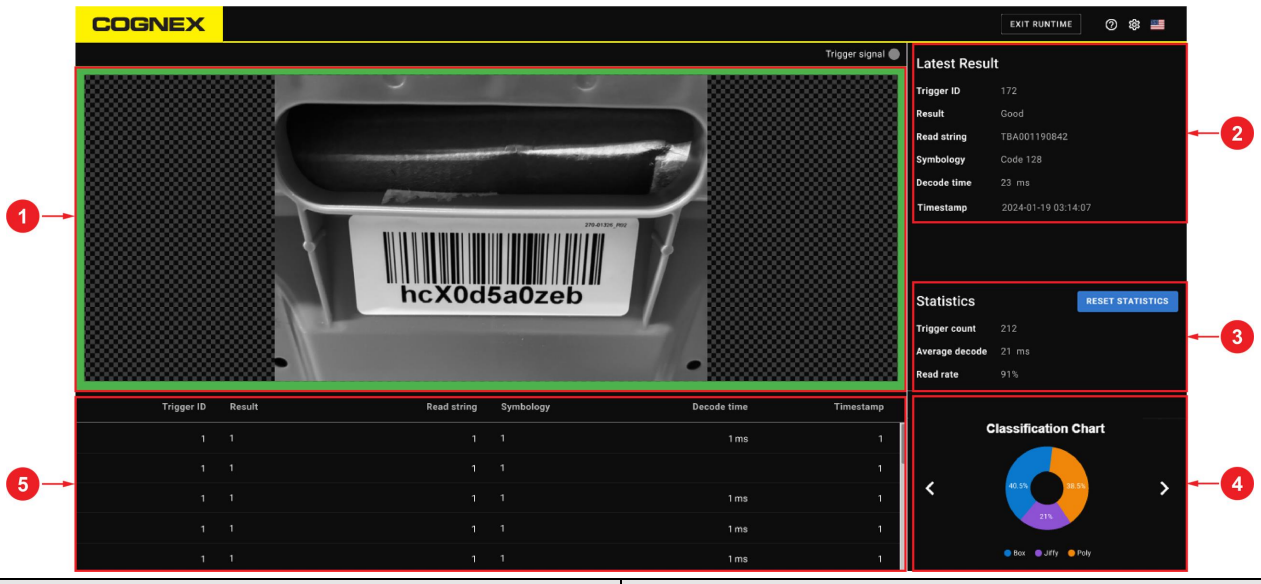
Option	Description
Leading Character	Specifies the character that starts the string.
Trailing Character	Specifies the character that ends the string.
Decimal Places	Specifies the number of decimal places to display for decimal numbers.
JSON Formatting	To use JSON formatting for the output string, check **JSON Formatting**. The default format of the output is ASCII.
Fixed Width	Specifies the number of characters to display for string values.
Leading Zeroes	Specifies the number of zeros to pad the string values if the length of the string is below the **Fixed Width**.



Number	Option Group	Settings
4	Output String	Shows the formatted output string. To see the labels for each data result in the output string, check **Show output labels**.

Runtime

The **Runtime** step places the device into an active mode and allows you to track the performance.



Number	Description
1	The screen shows the **Last Read** image of the device, or the image acquired for the **Selected Read**. You can move the image within the screen and zoom with the mouse wheel.

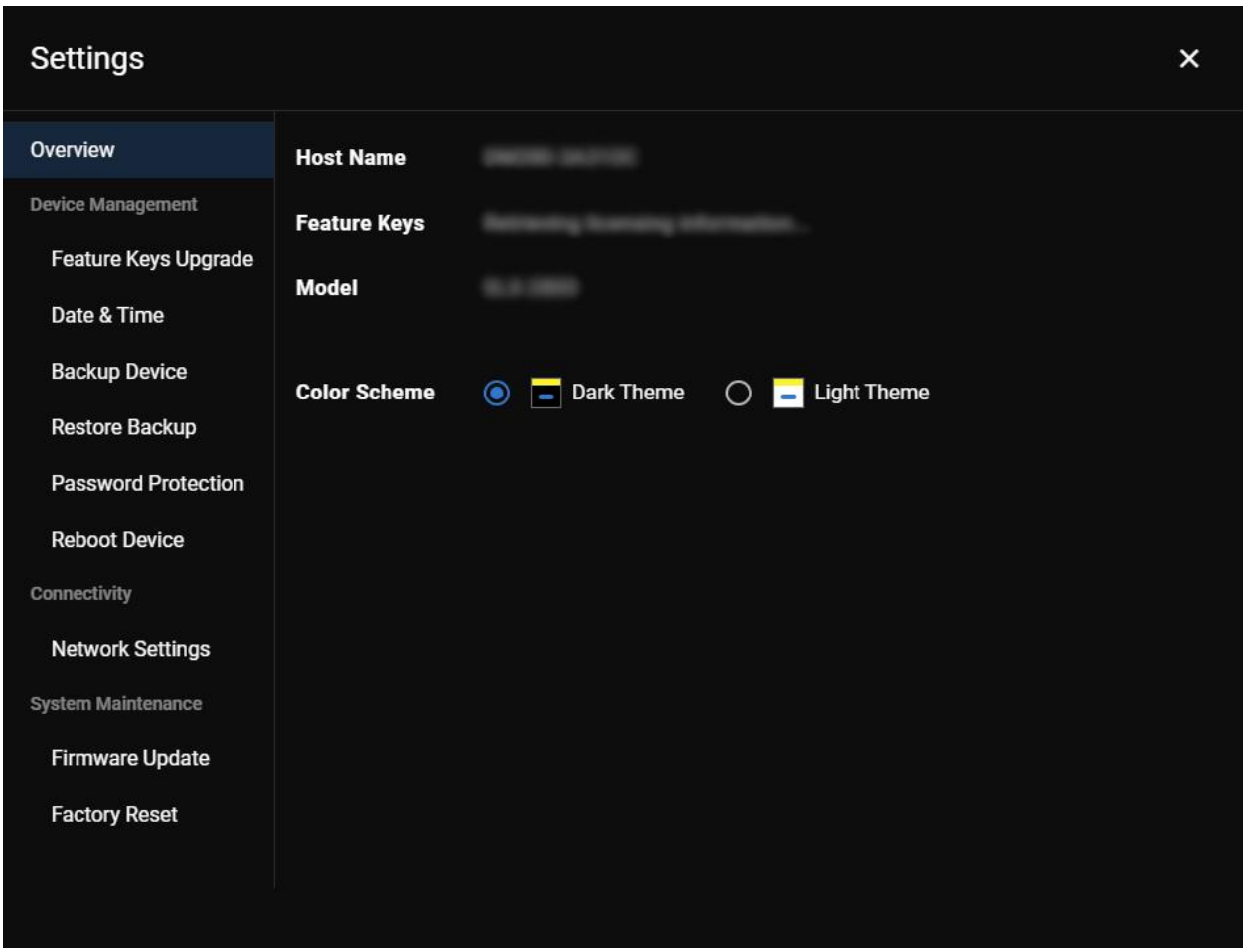
2	Result Data, including: • **Trigger ID**: a reference ID of the image, decoded string, and other result information • **Classification**: the class that the classifier tool predicts • **Read String**: the formatted decoded string • **Symbology**: the symbology type of the code • **Decode Time**: the time taken to decode the symbol • **Timestamp**: the date and time when the image was acquired **Note**: • The results displayed depend on the tools you enable. • Hybrid device models can have multiple tools enabled in single trigger mode.
3	The **Statistics** pane shows overall result statistics. The statistics are reset when the device reboots or if you click **Reset Statistics**.
4	Visual breakdown of different read statistics. Click on the arrows to switch between the results for different tools. You can click on each result category to toggle whether the chart displays that category.
5	A short result history list of previous trigger results. **Note**: Selecting an entry recalls the results of that trigger, but does not re-run the decode or retransmit the result.

Settings

The following subsections describe the available settings in the WebUI.

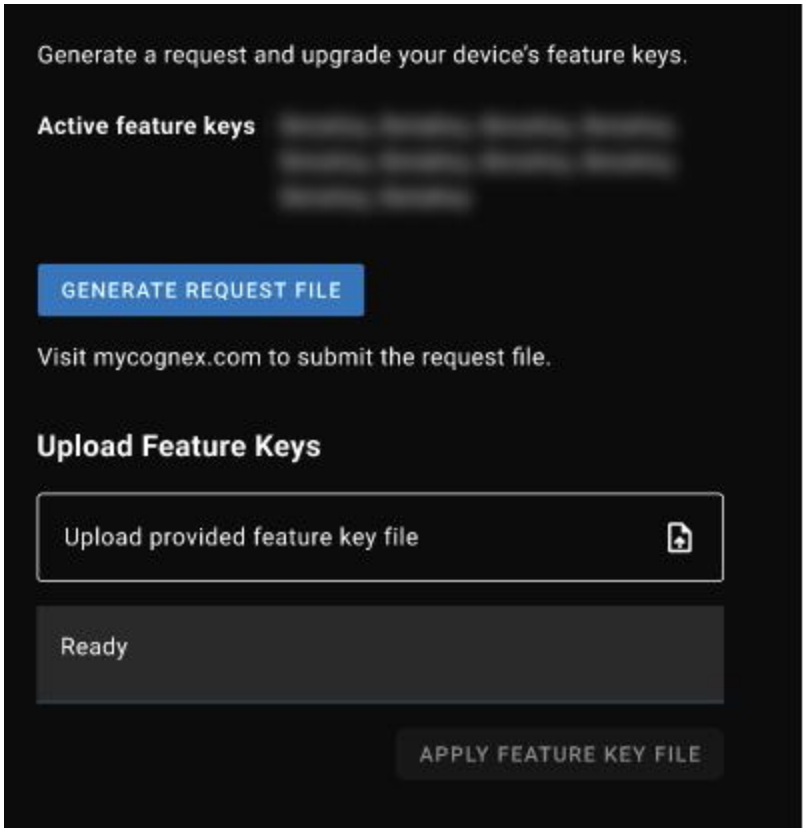
Overview

The **Overview** tab of **Settings** allows you to check the details of the device and select the theme of the UI.



Feature Keys Upgrade

The **Feature Keys Upgrade** tab of **Settings** allows you to manage the feature keys of the device.



Active feature keys lists the feature keys activate on the device.

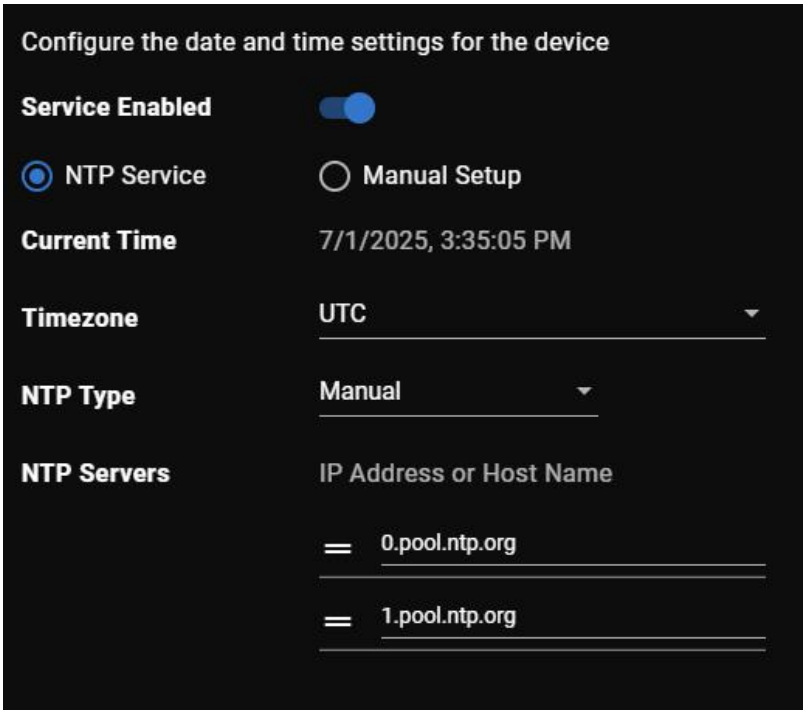
To activate feature keys:

1. Click **Generate Request File** to download the request file.
2. Go to mycognex.com and log in and upload the request file.
3. Download the feature key file.
4. Go back to **Settings** in the WebUI, and click **Upload provided feature key file** to upload the file.

   Select the feature key file downloaded from the Cognex website.
5. Click **Apply Feature Key File**.

Date & Time

The **Date & Time** tab of **Settings** allows you to configure date and time settings for your device.



NTP Service

To configure clock time sources between your servers, select **NTP Service**. You have the following configuration options:

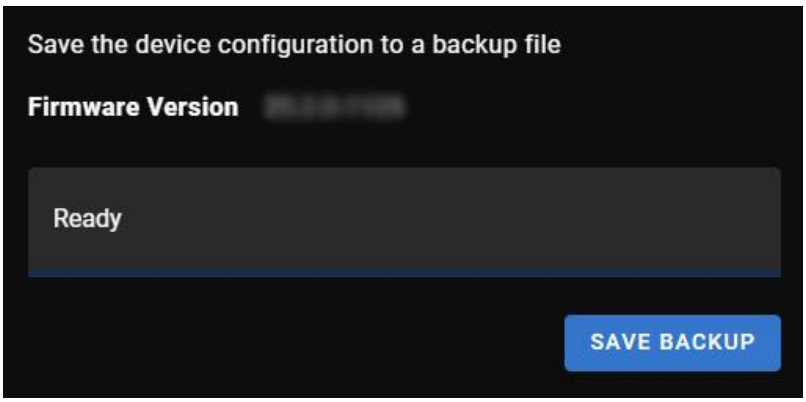
- **Current Time**: Displays the current time based on the settings.
- **Timezone**: Specifies the time zone.
- **NTP Type**: To enter the server addresses manually, select **Manual**. To configure the server addresses automatically, select **DHCP**.
- **NTP Servers**: Enter the server names manually.

Manual Setup

To set the date and time manually, select **Manual Setup** and click **Edit Time**.

Backup Device

The **Backup** tab of **Settings** allows you to save the device configuration to a backup file by clicking on **Save Backup**.



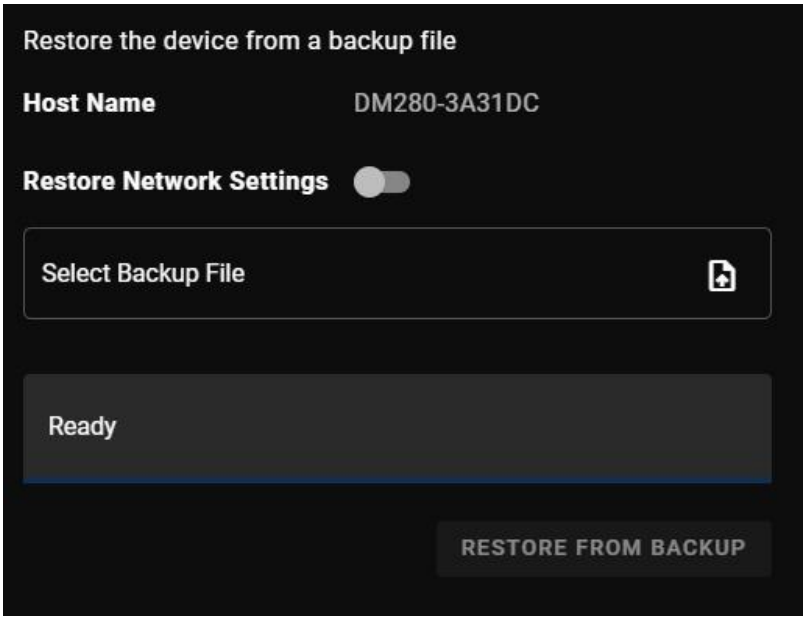
Restore Backup

The **Restore Device** tab of **Settings** allows you to restore the device from a backup file by selecting your backup file and clicking on **Restore**.

To override the current network settings of the device with the network settings of the backup file, enable the **Restore Network Settings** toggle.

Note:

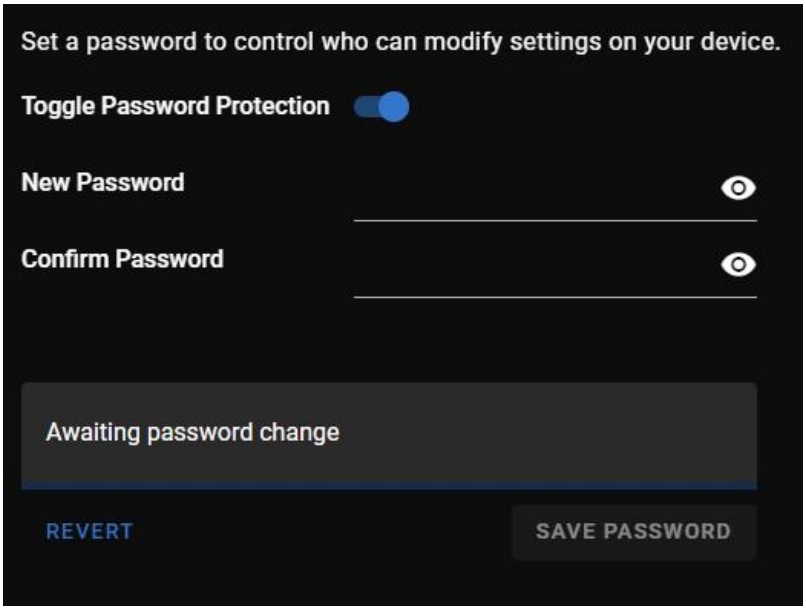
- If you enable the **Restore Network Settings**, the secure communication settings are copied from the backup file to the target device.
- The secure communication files are only restored if the **Restore Network Settings** toggle is enabled.



Note:
If restoring your device changes the IP address of the reader, your connection to the device is lost when the restore process finishes. You can find the new IP address of the device using device discovery in Station Manager.

Password Protection

The **Password Protection** tab of **Settings** allows you to set up a password for the device.

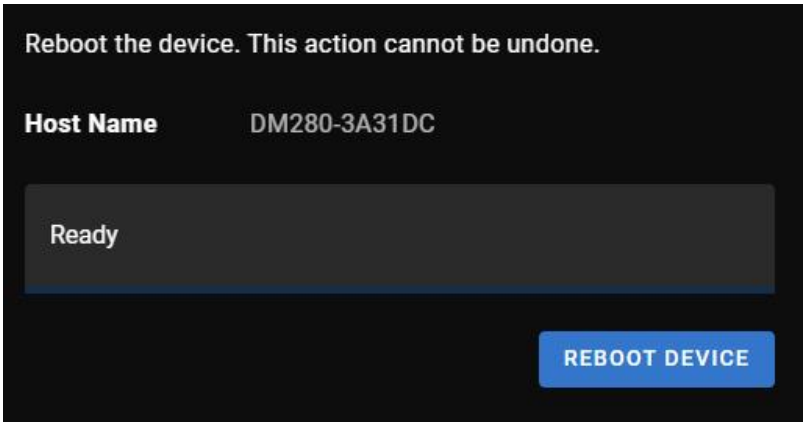


To set up a password:

1. Enable the **Toggle Password Protection** toggle.
2. Enter the new password in the **New Password** and **Confirm Password** fields.
3. Click **Save Password**.

Reboot Device

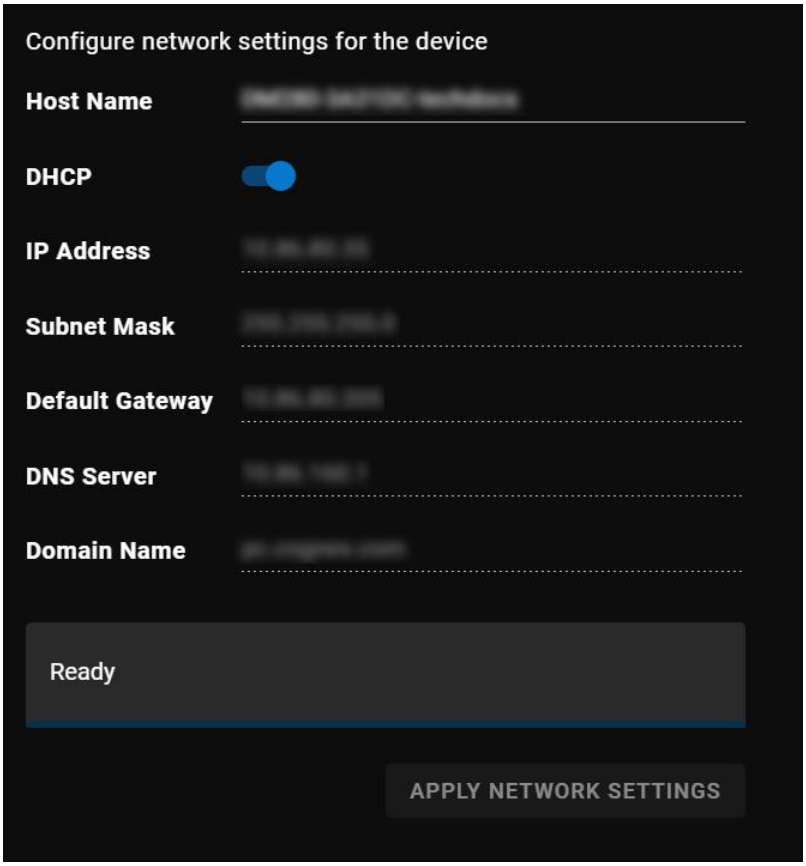
The **Reboot** tab of **Settings** allows you to reboot the device.



After clicking **Reboot Device**, the screen confirms that the device initiated the reboot. Rebooting takes a few seconds to finish.

Network Settings

The **Network Settings** tab of **Settings** allows you to configure network settings for your device. To save network settings, click on **Apply**.



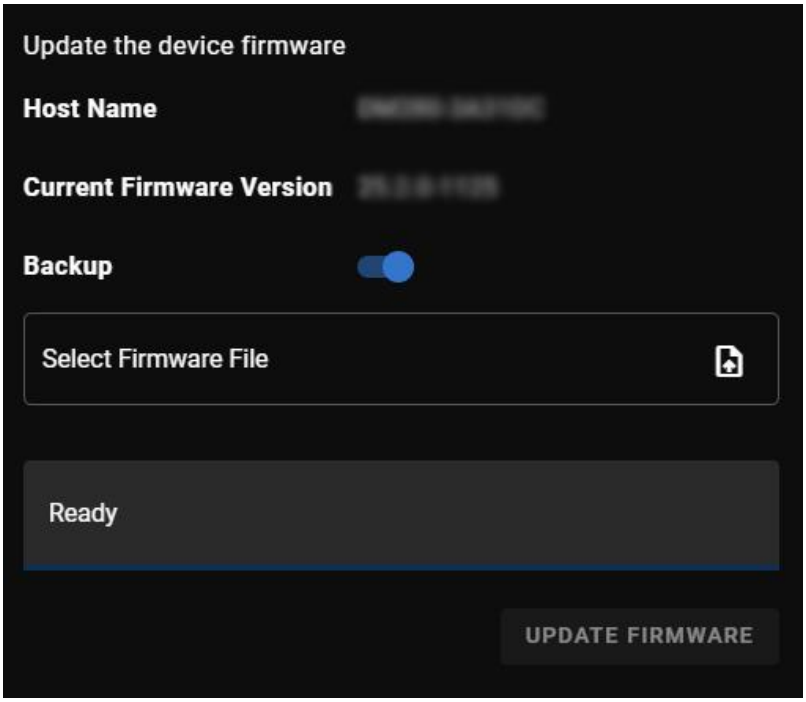
You have the following network settings for the device:

- **Host Name**: Specifies the host name.
- **DHCP**: Enables or disables DHCP.
- **IP Address**: Specifies the IP address.
- **Default Gateway**: Specifies the default gateway.
- **DNS Server**: Specifies the DNS server.
- **Domain Name**: specifies the domain name.

Firmware Update

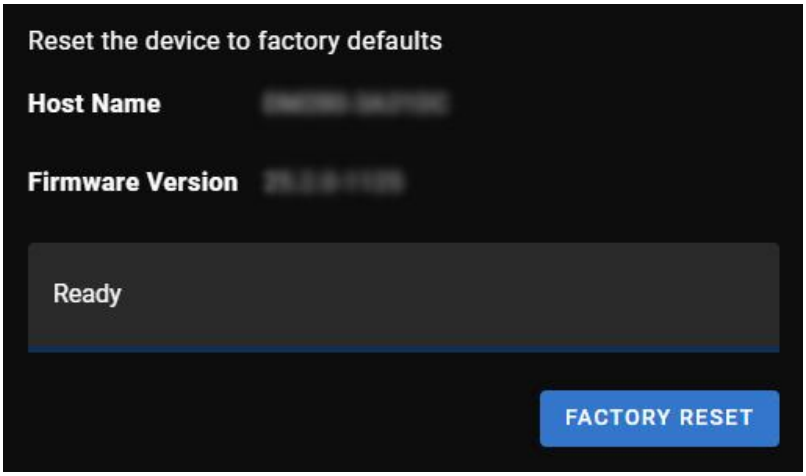
The **Firmware Update** tab of **Settings** allows you to update the firmware of your device by uploading the firmware file from your PC and then clicking on **Update Firmware**.

Enable the **Backup** toggle to save a backup file before the update.



Factory Reset

The **Factory Reset** tab of **Settings** allows you to reset the device to factory settings by clicking **Factory Reset**.



Note:
If restoring your device changes the IP address of the reader, your connection to the device is lost when the restore process finishes. You can find the new IP address of the device using device discovery in Station Manager.

Industrial Communications

The SLX series devices support the EtherNet/IP and PROFINET industrial communications protocols.

Note: The name LS280D is used for SLX-280D for industrial protocol configuration.

EtherNet/IP

The device supports EtherNet/IP™, an application level protocol based on the Common Industrial Protocol (CIP). EtherNet/IP provides an extensive range of messaging options and services transferring data and I/O over Ethernet. All devices on an EtherNet/IP network present their data to the network as a series of data values called attributes. Attributes can be grouped with other related data values into sets called Assemblies.

By default the device has the EtherNet/IP protocol disabled. The protocol can be enabled via DMCC, or in the **Communications** application step.

Tip: You can download the Electronic Data Sheet (EDS) file for your device from the Cognex Support Site.

Preparing to Use Ethernet/IP

Perform these steps to start using EtherNet/IP:

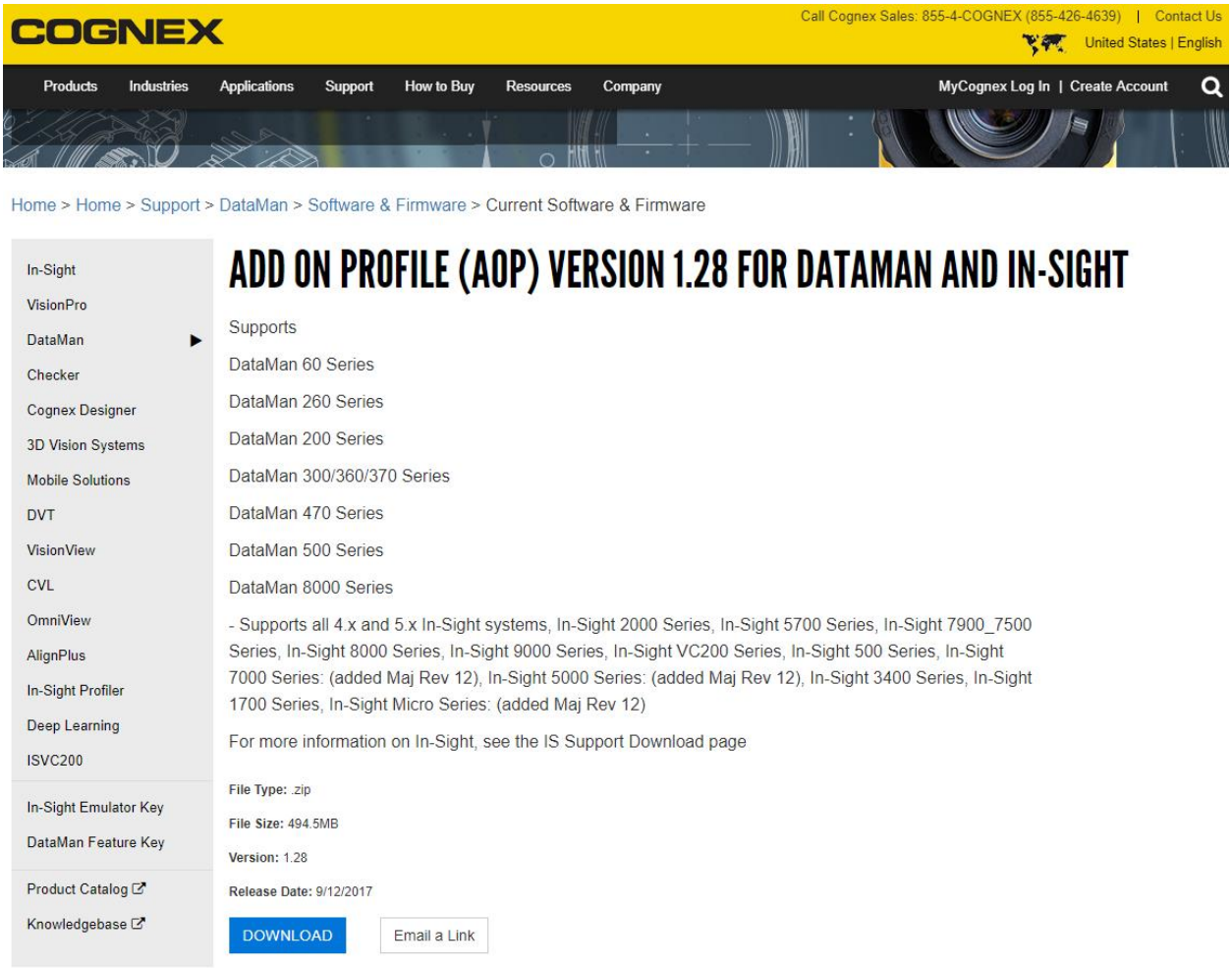
- Install the Rockwell Software tool.
- Set up the Rockwell Software tool so that it recognizes your device.
- Install the SLX Electronic Data Sheet (EDS) for the device.

Note: Cognex recommends using an existing DataMan Add-on Profile (AOP) and disabling electronic keying on the PLC side. The signal mapping for SLX devices is the same as for DataMan readers.

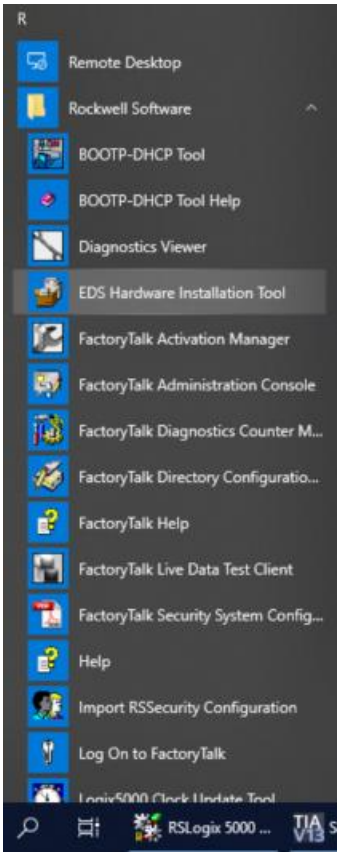
Perform the following steps to set up EtherNet/IP:

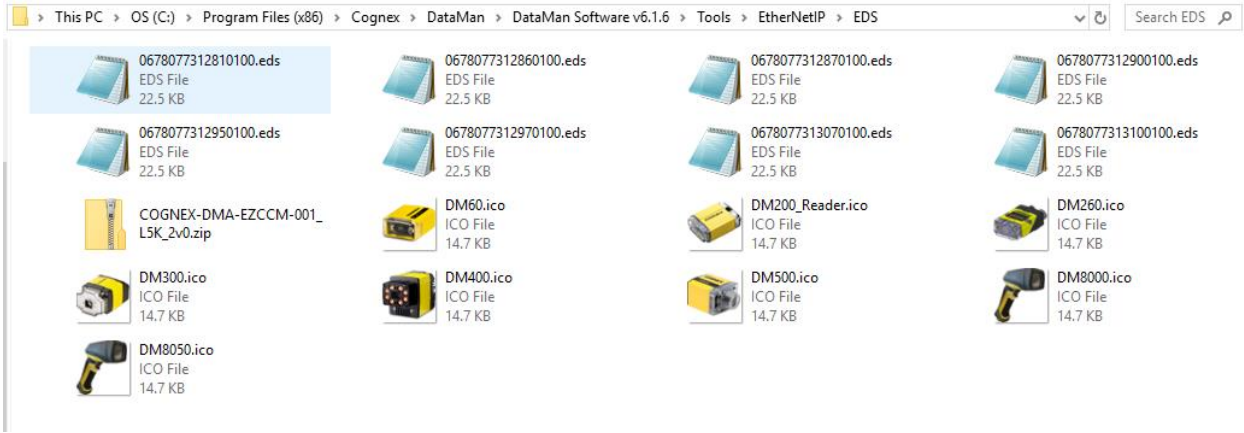
1. Make sure that you select the **Add-on Profile** (AOP) installation and the **Samples** installation. The Add-on Profile is only used with Rockwell ControlLogix or CompactLogix PLCs.
2. Install the Rockwell Add-on Profiles by navigating to the following directory on Cognex.com: https://support.cognex.com/en/downloads/dataman/software-firmware.

3. In the search box, type **Add-on Profile**. Click the file and download it from the following page:



4. From the Start menu, go to Programs -> Rockwell Software -> EDS Hardware Installation Tool.

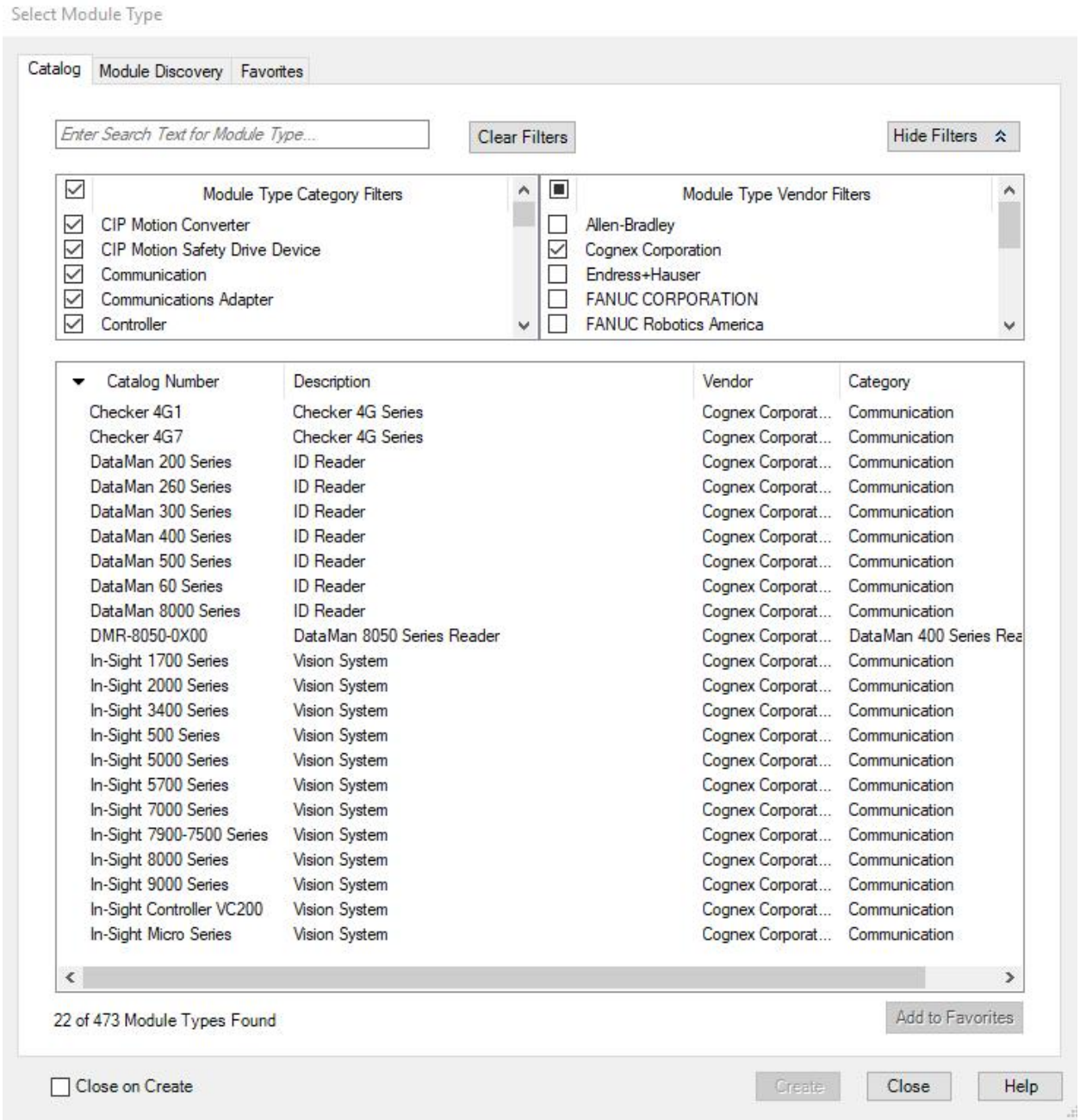




5. Run the EDS Installation Tool.

> **Note**: If you have an existing EDS file, uninstall it first, then install the latest version of the EDS.

6. Check if the device firmware is up-to-date.
7. Make sure that the **EtherNet/IP** option is enabled in the **Communications** application step.
8. In order for the changes to take effect, save your settings and cycle power. Go to **System** and click **Save Settings**.
9. Reboot your device.
10. Your device is visible now in the RSLinx.



Note: If your device is visible, but the icon is a question mark, repeat the EDS Installation.

11. Open one of the sample jobs and integrate your device into your program using the **Add-on Profile**.

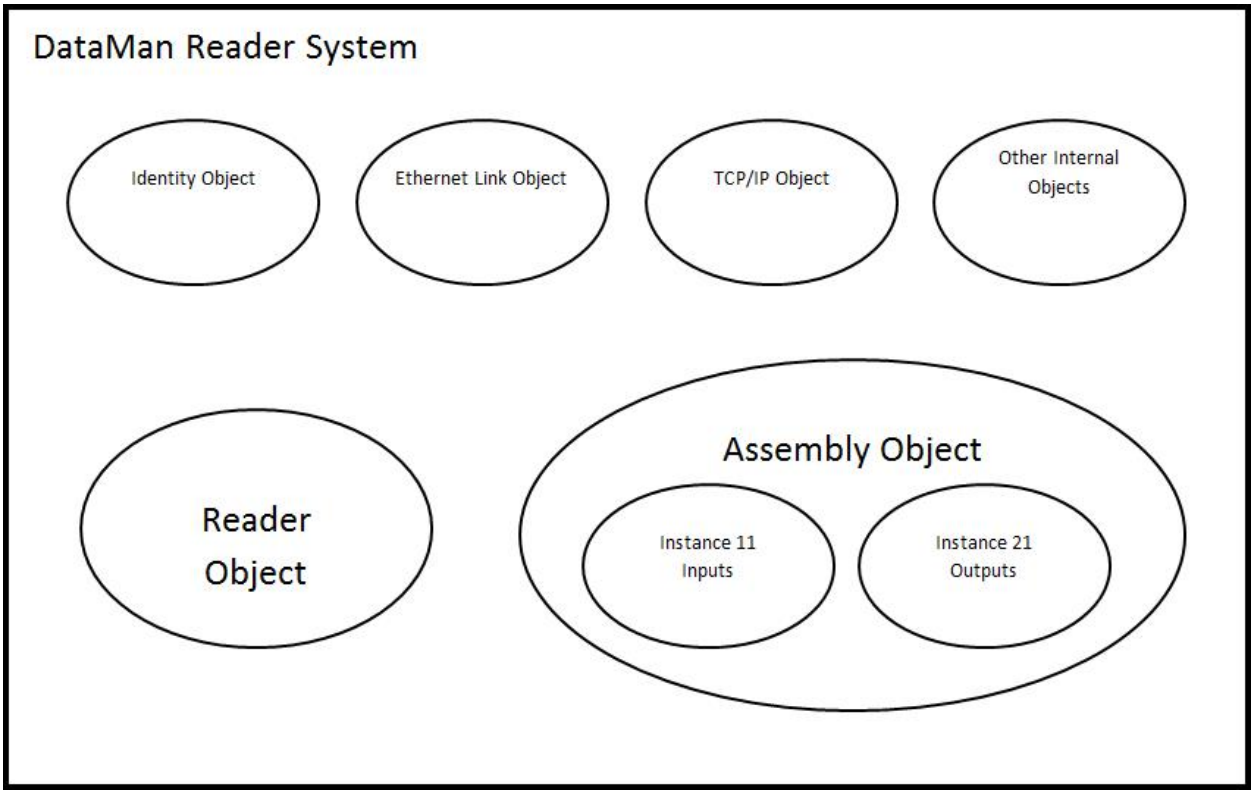
Ethernet/IP Object Model

SLX devices implement the Ethernet/IP object model with the ID Reader Object. The ID Reader Object is a vendor specific object class that is also used by DataMan readers. This means that it is not part of the CIP common (public) architecture but an extension. It is a custom object that Cognex has added to the EtherNet/IP architecture on the SLX device. All the data and functionality of this object model are available in the device. This includes triggering, status, events, errors, and result data.

The ID Reader Object is identified by its vendor-specific class code: DataMan ID Reader Object Class Code: 0x79

Objects are made up of attributes (data) and services (functionality). These can be defined at the class level (common to all instances of the class) or the instance level (unique to an individual instance). The CIP specification defines common attributes and services that apply to all objects (often these are optional). Vendors may also define their own attributes and services for their vendor specific classes.

The ID Reader Object attributes and services can be individually accessed via explicit messaging. In addition, a number of the ID Reader Object attributes are exposed in the DataMan assembly objects which allow them to be accessed as a group via implicit messaging.



Attributes

The DataMan ID Reader Object (Class Code: 0x79) has the following attributes:

Attribute ID	Access Rule	Name	Data Type	Description
0x9	Set	AcqTriggerEnable	BOOL	0 = EtherNet/IP triggering is disabled 1 = EtherNet/IP triggering is enabled
0xA	Set	AcqTrigger	BOOL	Acquire an image when this attribute changes from 0 to 1.
0xB	Get	AcqStatusRegister	BYTE	Bit0: Trigger Ready Bit1: Trigger Ack Bit2: Acquiring Bit3: Missed Acquisition Bit4-7: Reserved
0xC	Set	UserData	ARRAY of BYTE	User defined data that can be used as an input to the acquisition/decode.
0xD	Set	BufferResultsEnable	BOOL	Not supported on SLX devices.

Attribute ID	Access Rule	Name	Data Type	Description
0xE	Get	DecodeStatusRegister	BYTE	Not supported on SLX devices.
0xF	Set	ResultsAck	BOOL	Acknowledges that the client received the decode results.
0x10	Get	DecodeResults	STRUCT of	The last decode results.
		ResultsID	UINT	Decode results identifier. Corresponds to the TriggerID of the decoded image.
		ResultCode	UINT	Not supported on SLX devices.
		ResultExtended	UINT	Not supported on SLX devices.
		ResultLength	UINT	Current number of result data bytes.
		ResultData	ARRAY of BYTE	Result data from last decode.
0x12	Set	SoftEvents	BYTE	SoftEvents act as virtual inputs (execute action on 0 to 1 transition) Bit0: Not supported on SLX devices. Bit1: Not supported on SLX devices. Bit2: Not supported on SLX devices. Bit3: Not supported on SLX devices. Bit4: Not supported on SLX devices. Bit5: Not supported on SLX devices. Bit6: Execute DMCC command Bit7: Not supported on SLX devices.
0x15	Get	TriggerID	UNIT	Trigger identifier. ID of the next trigger to be issued.
0x16	Set	UserDataOption	UINT	Not supported on SLX devices.
0x17	Set	UserDataLength	UINT	Current number of user data bytes.
0x18	Get	SoftEventAck	BYTE	Not supported on SLX devices.

SoftEvents

SoftEvents act as “virtual” inputs. When the value of a SoftEvent changes from 0 to 1, the action associated with the event is executed. When it is done, the corresponding SoftEventAck bit changes from 1 to 0 to mark completion.

The SoftEvent and SoftEventAck form a logical handshake. After SoftEventAck changes to 1 the original SoftEvent is set back to 0. When that occurs, SoftEventAck is automatically set back to 0.

The “ExecuteDMCC” SoftEvent action require user supplied data. This data must be written to the UserData and UserDataLength area of the Input Assembly prior to invoking the SoftEvent.

Services

The ID Reader Object supports the following Common CIP services:

Service Code	Service Name	Description
0x05	Reset	Resets the ID Reader object.
0x0E	Get_Attribute_Single	Returns the contents of the specified attribute.
0x10	Set_Attribute_Single	Modifies the specified attribute.

The ID Reader Object supports the following vendor specific services:

Service Code	Service Name	Description
0x32	Acquire	Triggers a single acquisition.
0x34	SendDMCC	Sends a DMCC command to the device.
0x35	GetDecodeResults	Gets the content of the DecodeResults attribute.

Acquire Service

The Acquire Service triggers an acquisition (if the acquisition system is ready to acquire an image). If the acquisition could not be triggered, then the Missed Acquisition bit of the Trigger Ready will be set until the next successful acquisition.

SendDMCC Service

The SendDMCC Service sends a DMCC command string to the device. The request data consists of the DMCC command string that is to be sent to the device. The reply data will contain the string result of the DMCC command. Additionally, the service provides a numeric result status for the call. Most of these result codes relate to the basic success/failure of the service execution. However, the service also maps the actual DMCC status codes. This allows the PLC to interpret the service request without having to parse the actual DMCC return string.

Service Return Code	Description	DMCC Return Code
0	Success – No error	0
1	Bad Command	-
4	No Answer – System too busy	-
100	Unidentified error	100
101	Command invalid	101
102	Parameter invalid	102
103	Checksum incorrect	103
104	Parameter rejected/altered due to device state	104

Note: The DMCC command string must be in the CIP STRING2 format (16-bit integer indicating the string length in characters followed by the actual string characters, no terminating null required).

GetDecodeResults Service

The GetDecodeResults service reads data from the DecodeResults attribute of the ID Reader Object. This service takes parameters indicating the “size” (number of bytes to read) and the “offset” (offset into the DecodeResults attribute to begin reading). This gives the service the flexibility to be used with PLCs that have different restrictions on the amount of data allowed in an explicit message. It also allows you to access very large codes that cannot be completely transferred with implicit messaging (assembly object).

GetDecodeResults Request Data Format

Name	Type	Description
Size	UINT	The number of bytes of the DecodeResults attribute to read.
Offset	UINT	The offset into the DecodeResults attribute. This specifies the first byte of the DecodeResults attribute to begin reading (0 based offset).

Acquisition Sequence

The device can be triggered to acquire images implicitly through the assembly object.

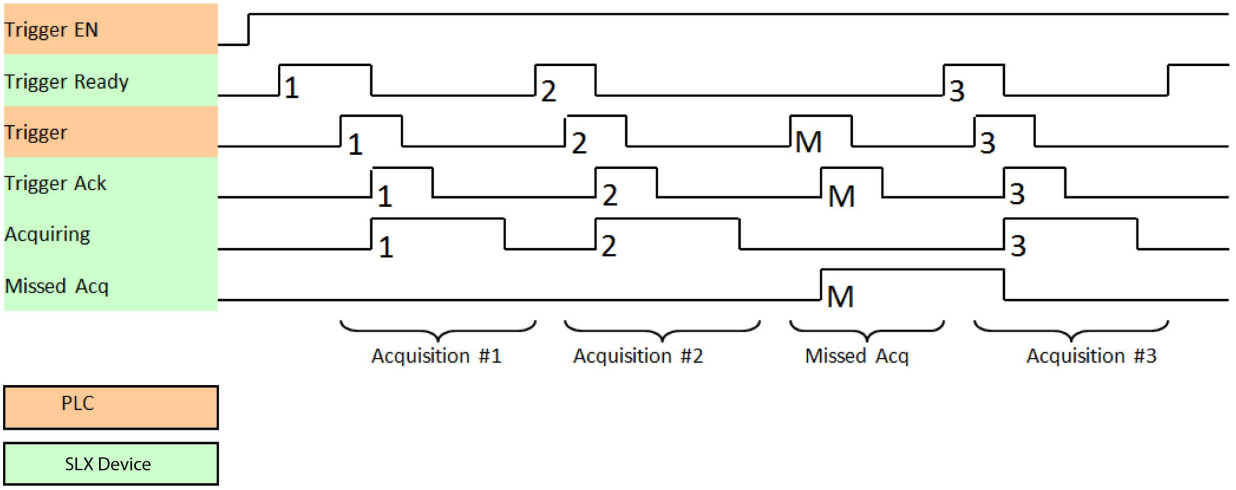
The ID Reader attributes are discussed in this section. These same values can also be accessed through the assembly objects. On startup, the AcqTriggerEnable attribute is False. Set the attribute to True to enable triggering. When the device is ready to accept triggers, the Trigger Ready bit in the AcqStatusRegister is set to True.

While the Trigger Ready bit is True, each time the ID Reader object sees the Trigger attribute change from 0 to 1, it initiates an image acquisition.

During an acquisition, the Trigger Ready bit is cleared and the Acquiring bit is set to True. When the acquisition is completed, the Acquiring bit is cleared. The Trigger Ready bit is again set to True once the device is ready to begin a new image acquisition.

The device allows acquisitions to overlap with the decoding of previous acquisitions. The Trigger Ready bit is set high after the acquisition is complete, while decoding may still be in progress. The Decoding bit is deprecated and only mirrors the behavior of the Acquiring bit. If trigger queuing is active or other trigger sources can interfere, the Trigger Ready bit may not be reliable. As result, conflicting trigger overruns may occur, which are reported in Bit 3 of the ResultCode attribute of the ID Reader object.

In certain cases, you can cancel an acquisition by clearing the Trigger signal before the read operation is finished. This allows you to cancel reads in Presentation and Manual mode if no code is in the field of view. To ensure that a read is not unintentionally cancelled, make sure that the PLC holds the Trigger signal True until both TriggerAck and ResultsAvailable are True (or DecodeComplete toggles state).



To force a reset of the trigger mechanism, set the TriggerEnable to False until the TriggerReady is 0. Then the TriggerEnable can be set to True to re-enable acquisition.

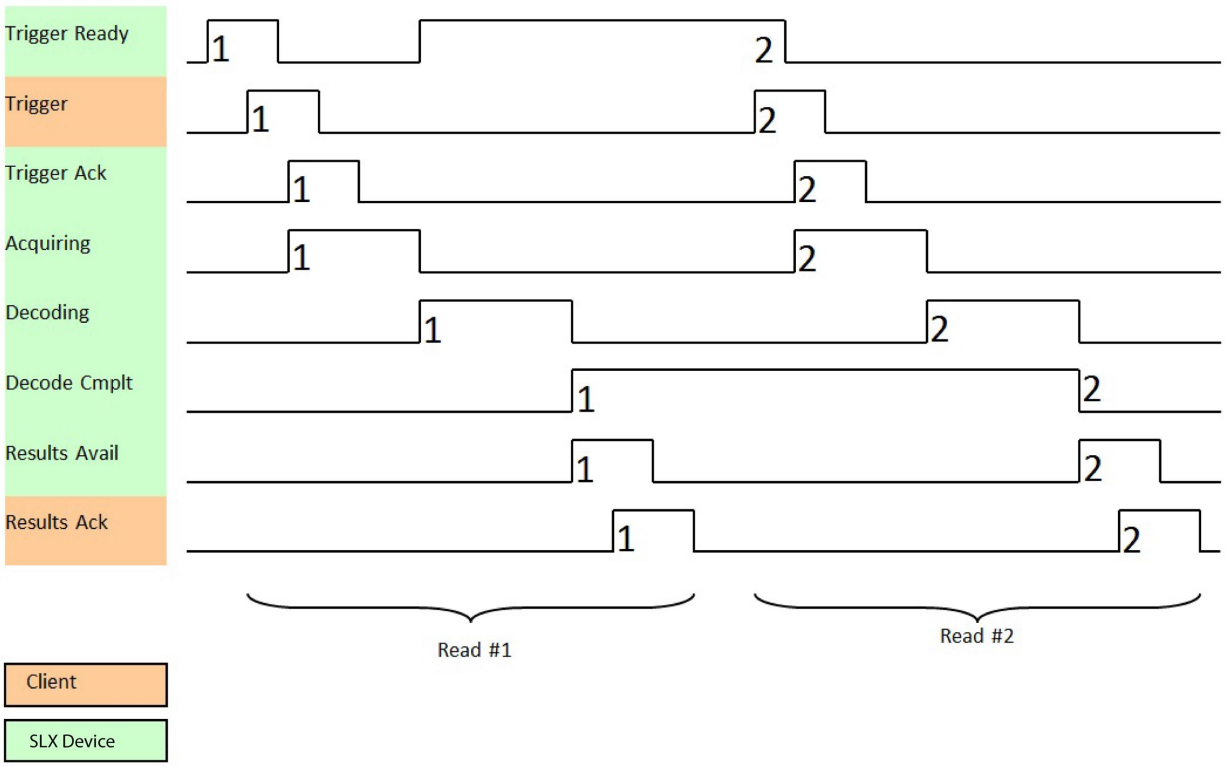
Decode / Result Sequence

After an image is acquired, it is decoded. When the decoding is complete and a result is ready to be delivered, the device toggles the DecodeComplete bit.

Decode results are reported asynchronously to the ID Reader Object.

If the BufferResultsEnable attribute is set to True, the new results are queued. The earlier decode results remain in the DecodeResults attribute until they are acknowledged by the client setting the ResultsAck attribute to True. After the Results Available bit is cleared, the client sets the ResultsAck attribute back to False to allow the next queued results to be placed in to the DecodeResults attribute. This is a necessary handshake to ensure that the client of the SLX device, which is the PLC, receives the results, even if short gaps occur between results.

Bit	Bit Name	Description
1	Decoding	Set when acquiring and decoding an image. The value of the Decoding bit is always the same as the Acquiring bit. The value of the **Decoding** bit is identical to the value of the **Acquiring** bit. Both bits are true if an acquisition, or series of acquisitions of the same trigger, is in progress.
2	Decode Complete	Toggled on completion of an image decode.
3	Results Buffer Overflow	Remains set to zero.
4	Results Available	Set when new results are placed in the DecodeResults attribute. Stays set until the results are acknowledged by setting ResultsAck to true.



Assembly Object

Assemblies are combinations of selected attributes (data items) from CIP objects within a device. The device vendor defines assemblies according to their needs and combine data together in useful groupings according to the requirements of the application.

The device is an I/O adapter class device. The convention for adapters is that Input Assemblies produce (transmit) data for another device (that is, device to PLC) and Output Assemblies consume (receive) data from another device (that is, PLC to device). The device acts as an I/O module for another device such as a PLC.

Assembly objects use implicit messaging. They are blocks of data which are transmitted as the raw payload of implicit messaging packets. These implicit messaging packets are produced (transmitted) repeatedly at a predefined chosen rate (for example, 100ms or 200ms).

The device have a single input assembly and single output assembly. These assemblies combine selected attributes (data) of the DataMan ID Reader Object into groupings that minimize network bandwidth, and still allow for efficient control and

processing. The data in these assemblies can also be accessed individually from the ID Reader Object. However, using the assembly objects is much more efficient, thus they are the primary means of runtime communication between a device and a PLC.

Input Assembly

The Input assembly provides status information, process state, and decode results.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11	0	Reserved				Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	General Fault	Reserved			Results Available	Results Buffer Overrun	Decode Complete Toggle	Decoding
	2	SoftEvent Ack 7	SoftEvent Ack 6	SoftEvent Ack 5	SoftEvent Ack 4	SoftEvent Ack 3	SoftEvent Ack 2	SoftEvent Ack 1	SoftEvent Ack 0
		Note: SLX devices only support SoftEvent Ack 6.							
	3-5	Reserved							
	6	Trigger ID (16-bit integer)							
	7								
	8	Result ID (16-bit integer)							
	9								
	10	Result Code (16-bit integer)							
	11								
	12	Result Extended (16-bit integer)							
	13								
	14	Result Data Length (16-bit integer)							
	15								
	16	Result Data 0							
	...								
	499	Result Data 483							

Output Assembly

The Output assembly contains control signals, software event signals, and any user data required for the trigger and decode.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Reserved				Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	SoftEvent 7	SoftEvent 6	SoftEvent 5	SoftEvent 4	SoftEvent 3	SoftEvent 2	SoftEvent 1	SoftEvent 0
		Note: SLX devices only support SoftEvent Ack 6.							
	2	Reserved							
	3								
	4	User Data Option (16-bit integer)							
	5	**Note**: Not supported on SLX devices.							
	6	User Data Length (16-bit integer)							
	7								
	8	User Data 0							
	...								
	499	User Data 491							

Industrial Communications Signals

The device uses a fixed set of pre-named signals which transfer data and results between a PLC and the device. The signals are shared between the supported protocols, but they are mapped into the systems differently.

- **TriggerReady**: Indicates when the SLX device can accept a new trigger. This tag is True when the Control tag “TriggerEnable” has been set , and the sensor is not acquiring an image.
- **TriggerAck**: Indicates when the SLX device has been triggered (that is, the Control tag “Trigger” has been set to True). This tag stays set until the Trigger tag is cleared.
- **Acquiring**: Indicates when the SLX device is acquiring an image either by setting the Trigger bit or by an external trigger.
- **MissedAcq**: Indicates when the SLX device misses an acquisition trigger. This happens when trigger on through the Trigger On bit of Ethernet/IP fails. It is cleared when the next successful acquisition occurs.
- **Decoding**: Deprecated. Same behavior as **Acquiring**.
- **DecodeCompleted**: Tag value is toggled (1 to 0 or 0 to 1) when a decode is completed.
- **ResultsBufferOverrun**: Not supported on SLX devices.
- **ResultsAvailable**: Indicates when a set of decode results are available (that is, the ResultID, ResultCode, ResultLength and ResultsData tags contain valid data).
- **TrainCodeAck** :Not supported on SLX devices.
- **TrainMatchStringAck**: Not supported on SLX devices.
- **TrainFocusAck**: Not supported on SLX devices.
- **TrainBrightnessAck**: Not supported on SLX devices.
- **UnTrainAck**: Not supported on SLX devices.

- **ExecuteDmccAck**: Indicates that the SoftEvent “ExecuteDMCC” has completed.
- **SetMatchStringAck**: Not supported on SLX devices.
- **TriggerID**: Value of the next trigger to be issued. Used to match triggers issued with corresponding result data received later.
- **ResultID**: The value of TriggerID when the trigger that generated these results was issued. Used to match TriggerID’s with result data.
- **ResultCode**:
- **ResultExtended**: Not supported on SLX devices.
- **ResultLength**: Number of bytes of result data contained in the ResultData tag.
- **ResultData**: Decode result data. Control tag group is the data sent from the ControlLogix to the SLX device:

◢ SLX290:O	{...}	{...}		CC:DataMan_SINT32:...
◢ SLX290:O.Control	{...}	{...}		CC:DataMan_Control...
SLX290:O.Control.TriggerEnable	0		Decimal	BOOL
SLX290:O.Control.Trigger	0		Decimal	BOOL
SLX290:O.Control.ResultsBufferEnable	0		Decimal	BOOL
SLX290:O.Control.ResultsAck	0		Decimal	BOOL
SLX290:O.Control.TrainCode	0		Decimal	BOOL
SLX290:O.Control.TrainMatchString	0		Decimal	BOOL
SLX290:O.Control.TrainFocus	0		Decimal	BOOL
SLX290:O.Control.TrainBrightness	0		Decimal	BOOL
SLX290:O.Control.Untrain	0		Decimal	BOOL
SLX290:O.Control.ExecuteDMCC	0		Decimal	BOOL
SLX290:O.Control.SetMatchString	1		Decimal	BOOL
▷ SLX290:O.Control.UserDataOption	0		Decimal	INT
▷ SLX290:O.Control.UserDataLength	8		Decimal	INT
▷ SLX290:O.UserData	{...}	{...}	ASCII	SINT[32]

- **TriggerEnable**: Setting this tag enables EtherNet/IP triggering. Clearing this field disables the EtherNet/IP triggering.
- **Trigger**: Setting this tag triggers an acquisition if:
  - The TriggerEnable tag is set.
  - No acquisition/decode is currently in progress.
  - The device is ready to trigger.
- **ResultsBufferEnable**: Not supported on SLX devices.
- **ResultsAck**: The ResultsAck tag is used to acknowledge that the PLC read the latest results. When ResultsAck is set, the ResultsAvailable tag will be cleared. If results buffering is enabled, the next set of results are made available when the ResultsAck tag is cleared again.
- **TrainCode**: Not supported on SLX devices.
- **TrainMatchString**: Not supported on SLX devices.
- **TrainFocus**: Not supported on SLX devices.
- **TrainBrightness**: Not supported on SLX devices.
- **Untrain**: Not supported on SLX devices.

- **ExecuteDMCC**: Changing this tag from 0 to 1 invokes the DMCC operation. A valid DMCC command string must be written to UserData prior to invoking this SoftEvent.
- **SetMatchString**: Not supported on SLX devices.
- **UserDataOption**: Not supported on SLX devices.
- **UserDataLength**: Number of bytes of user data contained in the UserData tag.
- **UserData**: This data is sent to the SLX device to support acquisition and/or decode.

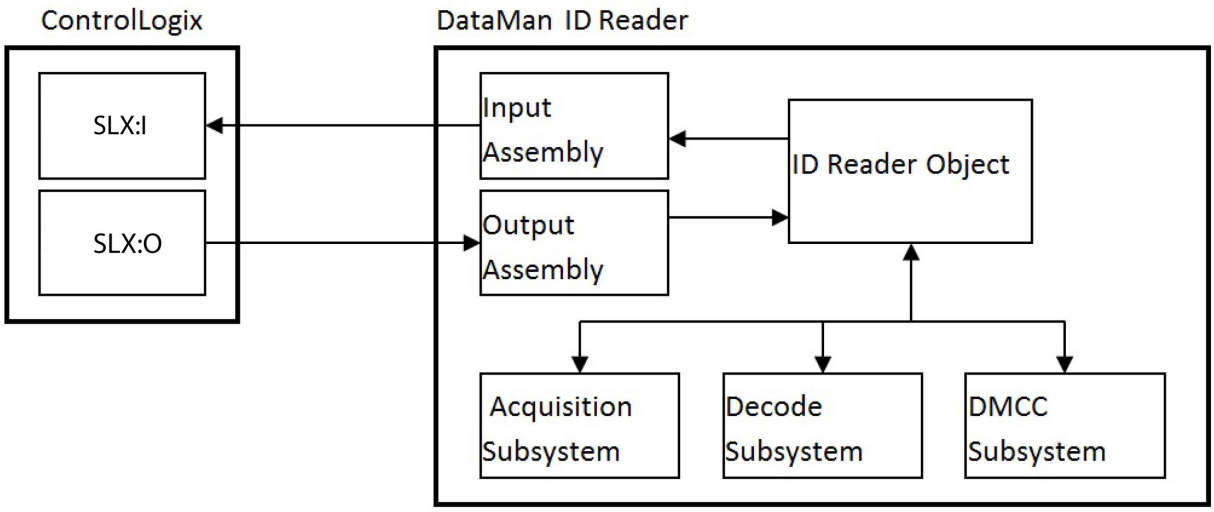
Rockwell ControlLogix Examples

Implicit messages transmit time-critical application specific I/O data and can be point-to-point or multicast. Explicit messages require a response from the receiving device. As a result, explicit messages are better suited for operations that occur less frequently. An instruction to send a DMCC command is an example of an explicit message.

Implicit Messaging

EtherNet/IP implicit messaging allows mapping the inputs and outputs of a SLX device into tags in the ControlLogix PLC. Once these connections are established, the data is transferred cyclically at a user defined interval (10 ms, 50 ms, 100 ms, and so on).

The figure below represents Ethernet-based I/O through EtherNet/IP:



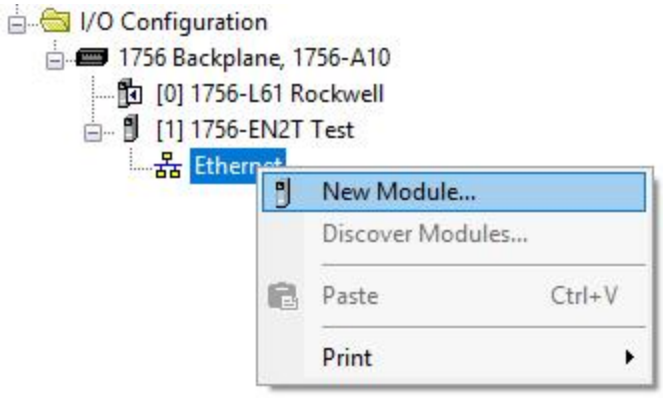
The *Input Assembly* and *Output Assembly* map various attributes (data) from the ID Reader object: The Input Assembly is the collection of SLX device data values sent to the PLC (PLC inputs); and the Output Assembly is the collection of data values received by the SLX device from the PLC (PLC outputs).

Establishing an Implicit Messaging Connection

To setup an EtherNet/IP implicit messaging connection between a SLX device and a ControlLogix controller, the SLX device must first be added to the ControlLogix I/O Configuration tree. The most efficient method is to use the Add-on Profile. This example assumes that the Add-on Profile has already been installed.

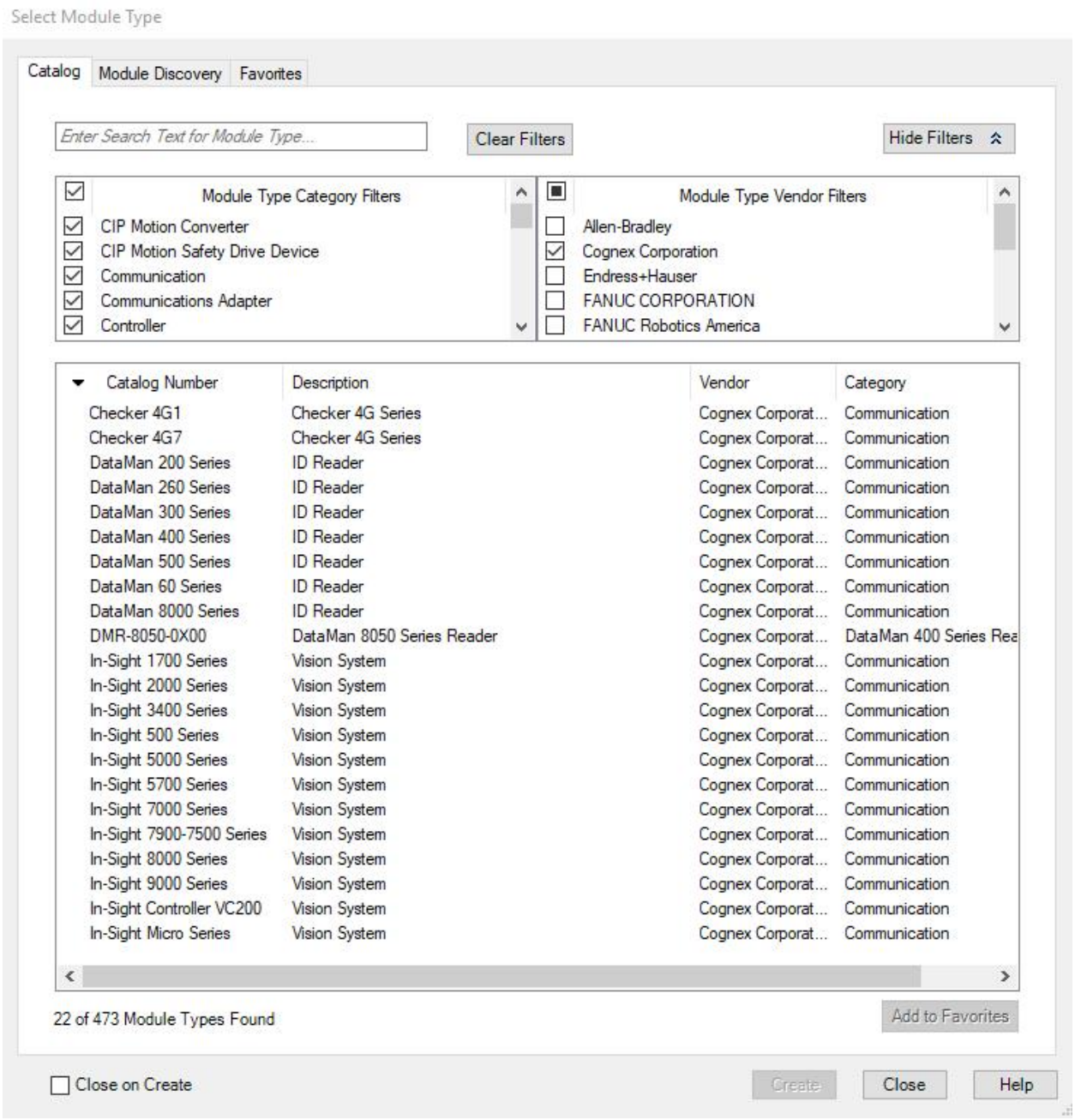
To establish an implicit messaging connection with a ControlLogix PLC:

1. Open RSLogix5000 and load your project (or select “File -> New…” to create a new one).
   From the I/O Configuration node, select the Ethernet node under the project Ethernet Module, right-click on the icon and select **New Module...** from the menu:



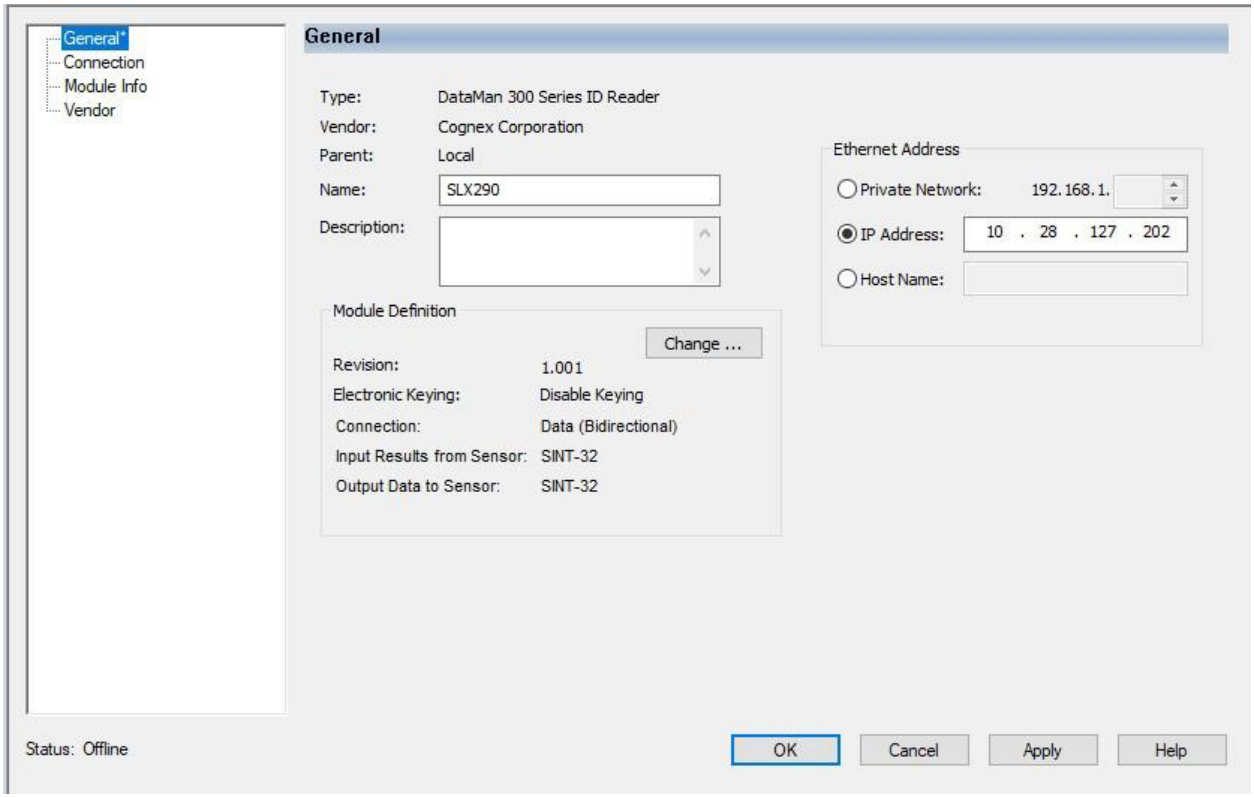
2. From the Select Module dialog, choose your model of SLX device from the list.

> **Note**: This option is only available after you install the DataMan Add-On Profile. SLX devices use the same modules as Cognex DataMan readers.

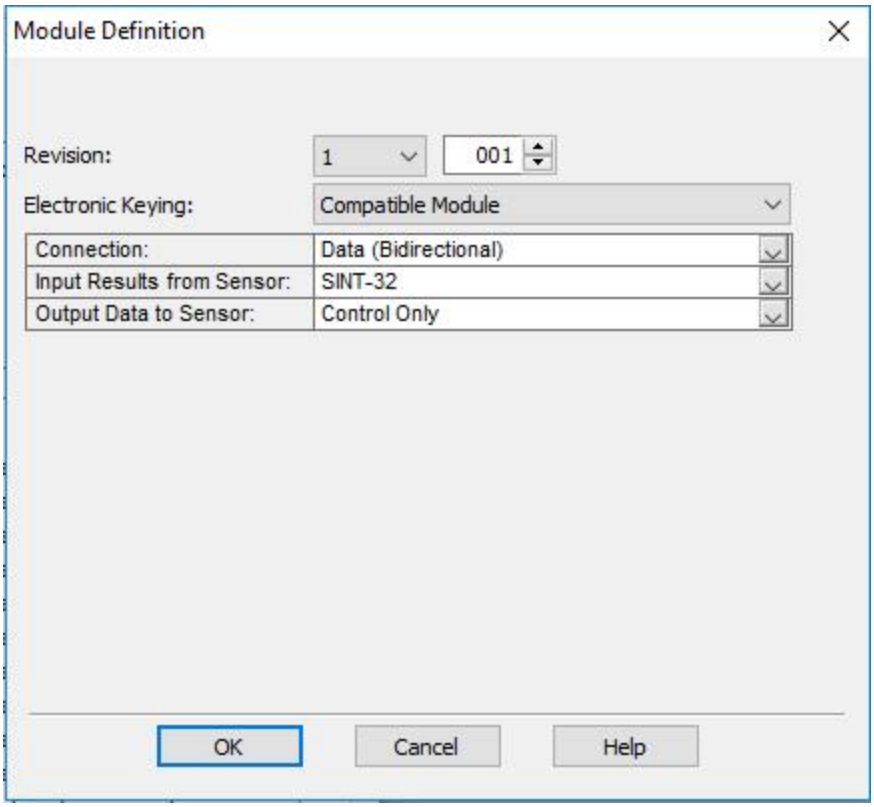


Note: The rest of the steps is identical regardless of which SLX model is selected.

3. After selecting the device, the configuration dialog for the SLX device system is displayed. Give the module a name and enter the SLX device's IP address. The default is a bidirectional (send/receive) connection consisting of control, status, and 32 bytes of result data with keying disabled. To change this default connection, click the **Change…** button. If no change is required, skip the next step.


General*
Connection
Module Info
Vendor
General
Type: DataMan 300 Series ID Reader
Vendor: Cognex Corporation
Parent: Local
Name: SLX290
Description:
Ethernet Address
Private Network: 192.168.1.
IP Address: 10 . 28 . 127 . 202
Host Name:
Module Definition
Change ...
Revision: 1.001
Electronic Keying: Disable Keying
Connection: Data (Bidirectional)
Input Results from Sensor: SINT-32
Output Data to Sensor: SINT-32
Status: Offline
OK
Cancel
Apply
Help

4. Clicking the **Change…** button brings up the **Module Definition** dialog. This dialog is used to alter the connection configuration. You can change:
    - SLX revision
    - Electronic keying
    - Connection type (bidirectional/receive-only)
    - Amount of data received (from the SLX device)
    - Amount of data sent (to the SLX device)



Electronic Keying: Defines the level of module type checking that is performed by the PLC before a connection is established.

Note: Cognex recommends to set **Electronic Keying** to **Disabled** for SLX devices.

Exact Match – All of the parameters must match or the connection will be rejected.

- Vendor
- Product Type
- Catalog Number
- Major Revision
- Minor Revision

Compatible Module – To prevent the inserted module from rejecting the connection:

- The Module Types have to match
- Catalog Number has to match
- Major Revision has to match
- The Minor Revision of the module has to be equal to or greater than the one specified in the software.

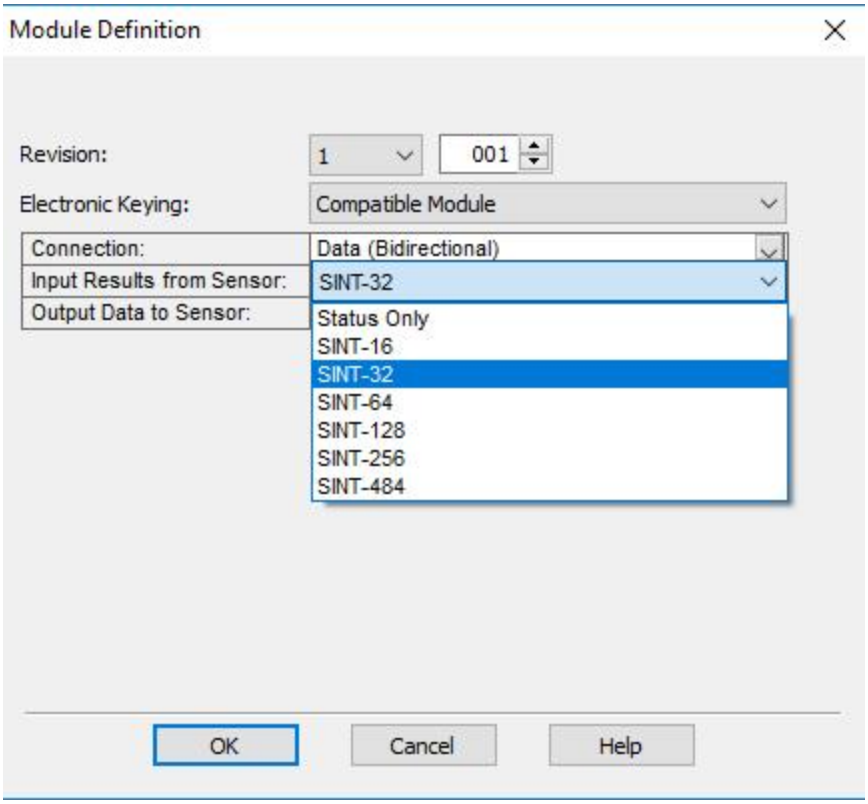
Disable Keying – The controller does not employ keying at all.

Connection: Defines the type of data flow.

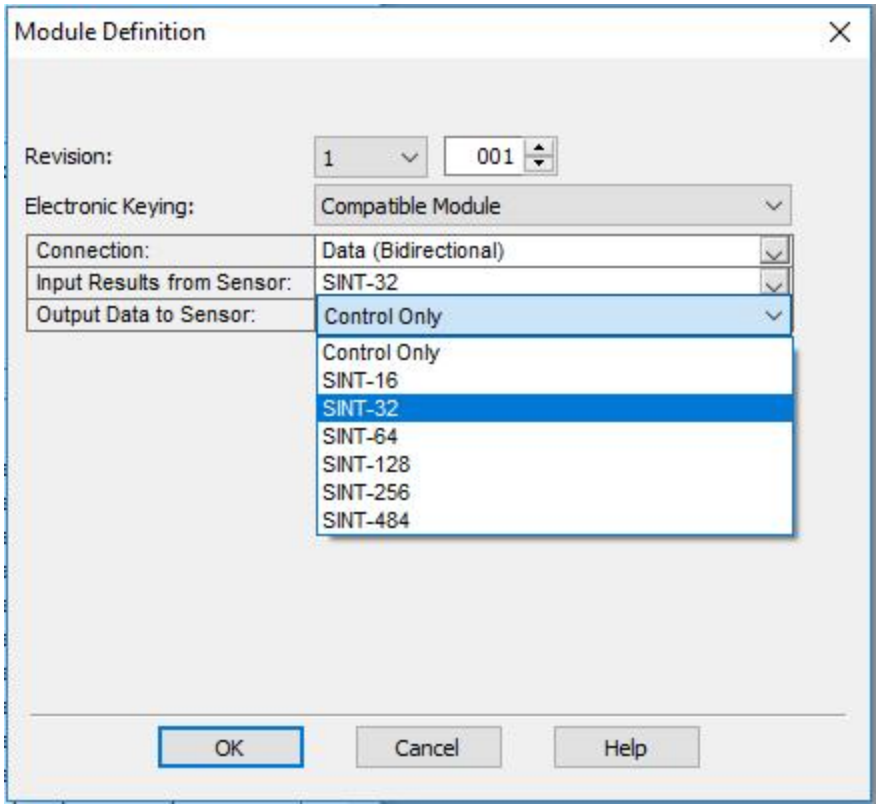
Data (Bidirectional) – The connection sends data to the SLX device and receives data from the SLX device.

Input (Results only) – The connection only receives data from the SLX device. If more than one PLC needs to receive data from the same SLX device, choose the Input connection option.

Input Results from Sensor: Defines the amount of data received on the connection from the SLX device. The minimum amount is the Status data only. The connection can be configured to also receive read result data. The amount of result data received is defined in fixed increments (16 bytes, 32 bytes, 64 bytes, and so on). Select the size to return no more than the largest code size to be read by the application. Setting the size larger wastes network bandwidth and diminishes performance.



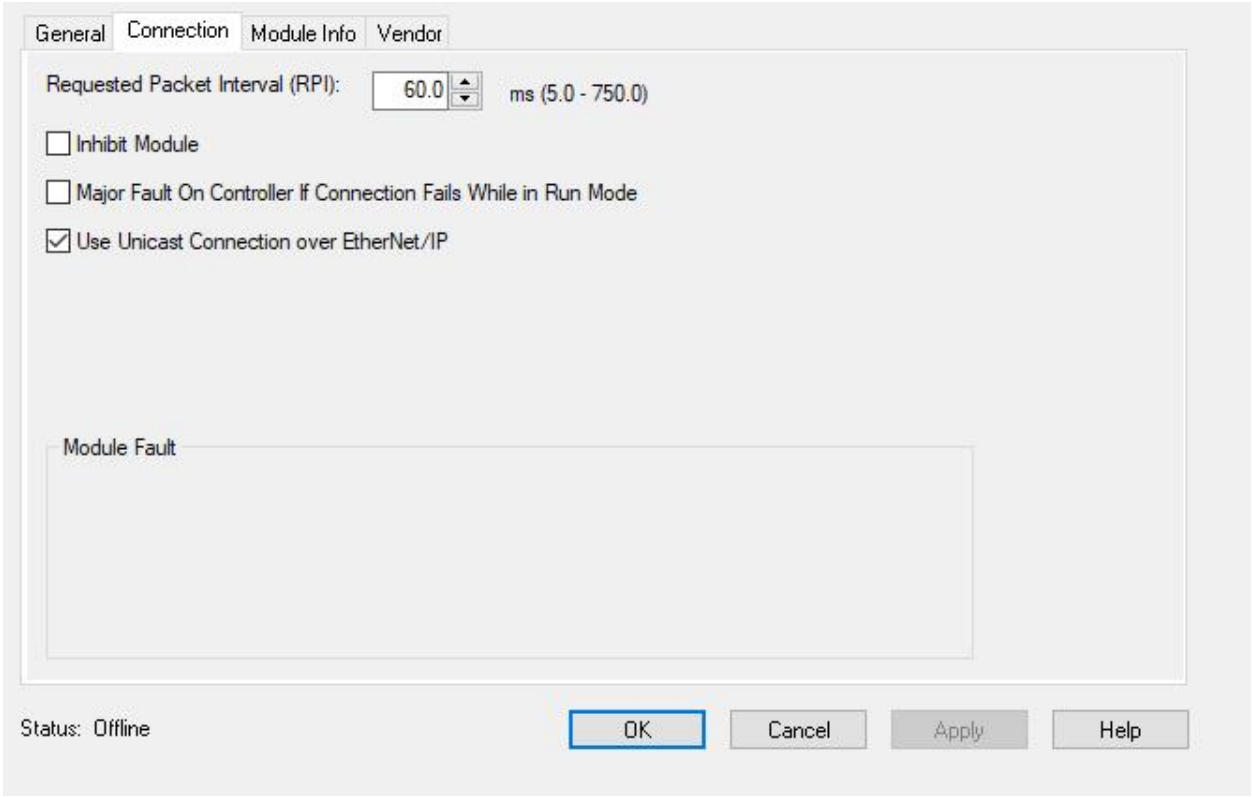
Output Data to Sensor: Defines the amount of data transmitted on the connection (to the SLX device). The minimum amount is the Control data only. The connection can be configured to also send user data. The amount of user data sent is defined in fixed increments (16 bytes, 32 bytes, 64 bytes, and so on). To enable User Data output, right-click the SLX module and then go to Properties -> Change -> Output Data to Sensor.



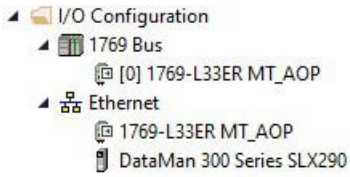
5. The final step is configuring the connection rate. The rate at which data is transmitted/received is defined as the Requested Packet Interval (RPI). The RPI defines how frequently the data is transmitted/received over the connection. To optimize network performance, do not set this rate lower than required by a given application, or lower than half the expected maximum read rate of the user application. Setting it lower wastes bandwidth and does not improve processing performance.

Note: Cognex recommends using a minimum RPI of 32 ms.

6. Select the “Connection” tab of the “New Module” dialog to set the rate.

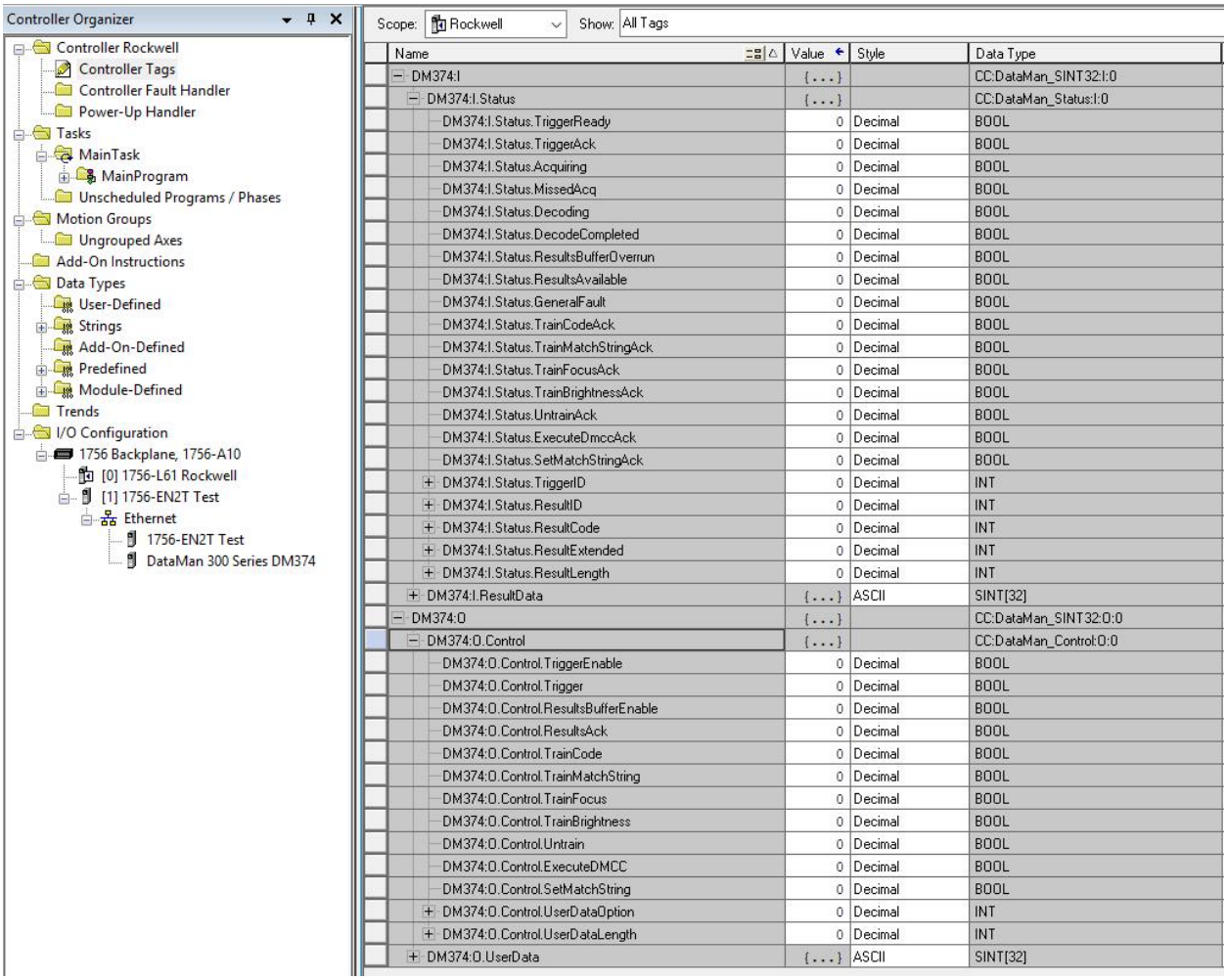


7. After adding the module to ControlLogix, the I/O tree should appear as follows:



8. When the SLX module is added to the I/O tree, RSLogix 5000 creates tags that map to the SLX device Input and Output Data (that is, the Input and Output Assembly Objects in the SLX device). These tags can be found under the “Controller Tags” node of the project tree.

> **Note**: The base name of these tags is the name you gave to the SLX Module that you added to the I/O Configuration in the previous steps.



Name	Value	Style	Data Type
DM374:I	{...}		CC:DataMan_SINT32:I:0
DM374:I.Status	{...}		CC:DataMan_Status:I:0
DM374:I.Status.TriggerReady	0	Decimal	BOOL
DM374:I.Status.TriggerAck	0	Decimal	BOOL
DM374:I.Status.Acquiring	0	Decimal	BOOL
DM374:I.Status.MissedAcq	0	Decimal	BOOL
DM374:I.Status.Decoding	0	Decimal	BOOL
DM374:I.Status.DecodeCompleted	0	Decimal	BOOL
DM374:I.Status.ResultsBufferOverrun	0	Decimal	BOOL
DM374:I.Status.ResultsAvailable	0	Decimal	BOOL
DM374:I.Status.GeneralFault	0	Decimal	BOOL
DM374:I.Status.TrainCodeAck	0	Decimal	BOOL
DM374:I.Status.TrainMatchStringAck	0	Decimal	BOOL
DM374:I.Status.TrainFocusAck	0	Decimal	BOOL
DM374:I.Status.TrainBrightnessAck	0	Decimal	BOOL
DM374:I.Status.UntrainAck	0	Decimal	BOOL
DM374:I.Status.ExecuteDmccAck	0	Decimal	BOOL
DM374:I.Status.SetMatchStringAck	0	Decimal	BOOL
DM374:I.Status.TriggerID	0	Decimal	INT
DM374:I.Status.ResultID	0	Decimal	INT
DM374:I.Status.ResultCode	0	Decimal	INT
DM374:I.Status.ResultExtended	0	Decimal	INT
DM374:I.Status.ResultLength	0	Decimal	INT
DM374:I.ResultData	{...}	ASCII	SINT[32]
DM374:O	{...}		CC:DataMan_SINT32:O:0
DM374:O.Control	{...}		CC:DataMan_Control:O:0
DM374:O.Control.TriggerEnable	0	Decimal	BOOL
DM374:O.Control.Trigger	0	Decimal	BOOL
DM374:O.Control.ResultsBufferEnable	0	Decimal	BOOL
DM374:O.Control.ResultsAck	0	Decimal	BOOL
DM374:O.Control.TrainCode	0	Decimal	BOOL
DM374:O.Control.TrainMatchString	0	Decimal	BOOL
DM374:O.Control.TrainFocus	0	Decimal	BOOL
DM374:O.Control.TrainBrightness	0	Decimal	BOOL
DM374:O.Control.Untrain	0	Decimal	BOOL
DM374:O.Control.ExecuteDMCC	0	Decimal	BOOL
DM374:O.Control.SetMatchString	0	Decimal	BOOL
DM374:O.Control.UserDataOption	0	Decimal	INT
DM374:O.Control.UserDataLength	0	Decimal	INT
DM374:O.UserData	{...}	ASCII	SINT[32]

The tags are organized in two groups: Status and Control. The Status group represents all the data being received from the SLX device. The Control group represents all the data being sent to the SLX device.

> **Note**: SLX devices use the same modules and naming conventions as Cognex DataMan readers.

These tags are the symbolic representation of the DataMan Assembly Object contents. The PLC ladder is written to access these tag values. By monitoring or changing these tag values the PLC ladder is monitoring and changing the DataMan Assembly Object contents.

> **Tip**: Based on the configured RPI, there is a time delay between the SLX device and the PLC tag values. Take this time delay into account when writing all PLC ladders.

Accessing Implicit Messaging Connection Data

One aspect of the Add-on Profile is that it automatically generates ControlLogix tags representing the connection data.

The generated tags are divided into two groups: Status and Control. The Status group represents all the data being received from the SLX device. The Control group represents all the data being sent to the SLX device. For more information, see *Industrial Communications Signals* on page 42.

Status tag group is the data the ControlLogix receives from the SLX device:

◢ SLX290:I	{...}	{...}		CC:DataMan_SINT32:I:0
◢ SLX290:I.Status	{...}	{...}		CC:DataMan_Status:I:0
SLX290:I.Status.TriggerReady	0		Decimal	BOOL
SLX290:I.Status.TriggerAck	0		Decimal	BOOL
SLX290:I.Status.Acquiring	0		Decimal	BOOL
SLX290:I.Status.MissedAcq	0		Decimal	BOOL
SLX290:I.Status.Decoding	0		Decimal	BOOL
SLX290:I.Status.DecodeCompleted	1		Decimal	BOOL
SLX290:I.Status.ResultsBufferOverrun	0		Decimal	BOOL
SLX290:I.Status.ResultsAvailable	1		Decimal	BOOL
SLX290:I.Status.GeneralFault	0		Decimal	BOOL
SLX290:I.Status.TrainCodeAck	0		Decimal	BOOL
SLX290:I.Status.TrainMatchStringAck	0		Decimal	BOOL
SLX290:I.Status.TrainFocusAck	0		Decimal	BOOL
SLX290:I.Status.TrainBrightnessAck	0		Decimal	BOOL
SLX290:I.Status.UntrainAck	0		Decimal	BOOL
SLX290:I.Status.ExecuteDmccAck	0		Decimal	BOOL
SLX290:I.Status.SetMatchStringAck	1		Decimal	BOOL
▷ SLX290:I.Status.TriggerID	10		Decimal	INT
▷ SLX290:I.Status.ResultID	9		Decimal	INT
▷ SLX290:I.Status.ResultCode	1		Decimal	INT
▷ SLX290:I.Status.ResultExtended	0		Decimal	INT
▷ SLX290:I.Status.ResultLength	9		Decimal	INT
▷ SLX290:I.ResultData	{...}	{...}	ASCII	SINT[32]

Note:
Configure the SLX module in RSLogix 5000 to manually add **UserData** to the output assembly.

Perform the following steps on a CompactLogix or ControlLogix PLC:

1. Right click the SLX module and select **Properties**.
2. Under **Module Definition**, click **Change**.
3. The Module Definition window pops up. Under the **Output Data to Sensor** drop-down menu, select **SINT-484**.
4. Click **OK**. RSLogix 500 is now updating the Module Definition.

The output assembly controller tags now list UserData as part of the output assembly.

Verifying Implicit Messaging Connection Operation

After the SLX device is added as an I/O device in a ControlLogix project and the project is downloaded to the controller, the I/O connection needs to be established. Once a successful connection is established, cyclic data transfers are initiated at the requested RPI.

To verify a proper I/O connection, follow these steps:

1. Download the project to the ControlLogix controller.
2. When the download completes, the project I/O indicator is **I/O OK**, indicating that the I/O connection is successfully completed.

   To verify the correct, two-way transfer of I/O data, go to the controller tags in RSLogix and change the state of the TriggerEnable bit from 0 to 1:

DM374:O	{...}		CC:DataMan_SINT32:O:0
DM374:O.Control	{...}		CC:DataMan_Control:O:0
DM374:O.Control.TriggerEnable	1	Decimal	BOOL
DM374:O.Control.Trigger	0	Decimal	BOOL
DM374:O.Control.ResultsBufferEnable	0	Decimal	BOOL
DM374:O.Control.ResultsAck	0	Decimal	BOOL
DM374:O.Control.TrainCode	0	Decimal	BOOL
DM374:O.Control.TrainMatchString	0	Decimal	BOOL
DM374:O.Control.TrainFocus	0	Decimal	BOOL
DM374:O.Control.TrainBrightness	0	Decimal	BOOL
DM374:O.Control.Untrain	0	Decimal	BOOL
DM374:O.Control.ExecuteDMCC	0	Decimal	BOOL
DM374:O.Control.SetMatchString	0	Decimal	BOOL
DM374:O.Control.UserDataOption	0	Decimal	INT
DM374:O.Control.UserDataLength	0	Decimal	INT

3. The TriggerReady tag changes to 1. Triggering is now enabled.
4. Whenever the Trigger tag is changed from 0 to 1, the SLX device acquires an image.

Note: The current TriggerID value is 1. Make sure that the results of the next trigger to be issued come back with a corresponding ResultID of 1.

5. After the acquisition/decode has completed, the DecodeCompleted tag will toggle, and the ResultsAvailable tag goes to 1. This example shows a successful read (ResultCode bit 0 = 1) and the read has returned 9 bytes of data (ResultLength=9). The data is in the ResultData tag.

◢ SLX290:I	{...}	{...}		CC:DataMan_SINT32:I:0
◢ SLX290:I.Status	{...}	{...}		CC:DataMan_Status:I:0
SLX290:I.Status.TriggerReady	0		Decimal	BOOL
SLX290:I.Status.TriggerAck	0		Decimal	BOOL
SLX290:I.Status.Acquiring	0		Decimal	BOOL
SLX290:I.Status.MissedAcq	0		Decimal	BOOL
SLX290:I.Status.Decoding	0		Decimal	BOOL
SLX290:I.Status.DecodeCompleted	1		Decimal	BOOL
SLX290:I.Status.ResultsBufferOverrun	0		Decimal	BOOL
SLX290:I.Status.ResultsAvailable	1		Decimal	BOOL
SLX290:I.Status.GeneralFault	0		Decimal	BOOL
SLX290:I.Status.TrainCodeAck	0		Decimal	BOOL
SLX290:I.Status.TrainMatchStringAck	0		Decimal	BOOL
SLX290:I.Status.TrainFocusAck	0		Decimal	BOOL
SLX290:I.Status.TrainBrightnessAck	0		Decimal	BOOL
SLX290:I.Status.UntrainAck	0		Decimal	BOOL
SLX290:I.Status.ExecuteDmccAck	0		Decimal	BOOL
SLX290:I.Status.SetMatchStringAck	1		Decimal	BOOL
▷ SLX290:I.Status.TriggerID	10		Decimal	INT
▷ SLX290:I.Status.ResultID	9		Decimal	INT
▷ SLX290:I.Status.ResultCode	1		Decimal	INT
▷ SLX290:I.Status.ResultExtended	0		Decimal	INT
▷ SLX290:I.Status.ResultLength	9		Decimal	INT
▷ SLX290:I.ResultData	{...}	{...}	ASCII	SINT[32]

Explicit Messaging

Unlike implicit messaging, explicit messages are sent to a specific device that always sends a reply to that message. As a result, explicit messages are better suited for operations that occur infrequently. Explicit messages can be used to read and write the attributes (data) of the ID Reader Object. They can also be used for acquiring images, sending DMCC commands and retrieving result data.

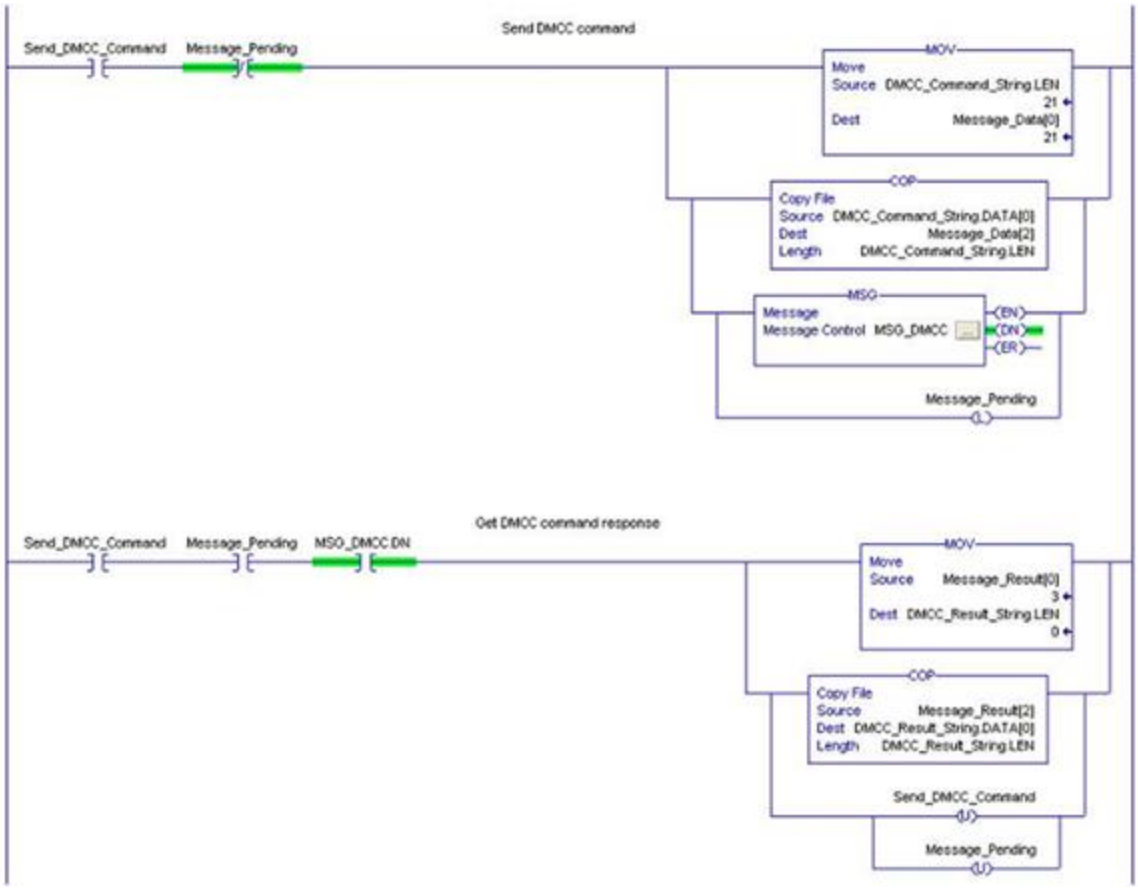
Issuing DMCC Commands

One of the more common explicit messages sent to a SLX device is an instruction to execute a DMCC command. Explicit messages are sent from ControlLogix to a SLX device using MSG instructions. There are two different paths for invoking DMCC messages with explicit messaging: through the CIP Object, or through the ID Reader Object “SendDMCC” service. The example shows the SendDMCC service.

The CIP STRING2 format is required for transmission across EtherNet/IP: a 16-bit length value followed by actual string characters, no null terminator. Logix stores strings in a slightly different format: a 32-bit length value followed by actual string characters, no null terminator. Therefore, some of the sample ladder involves converting to/from the two different string formats.

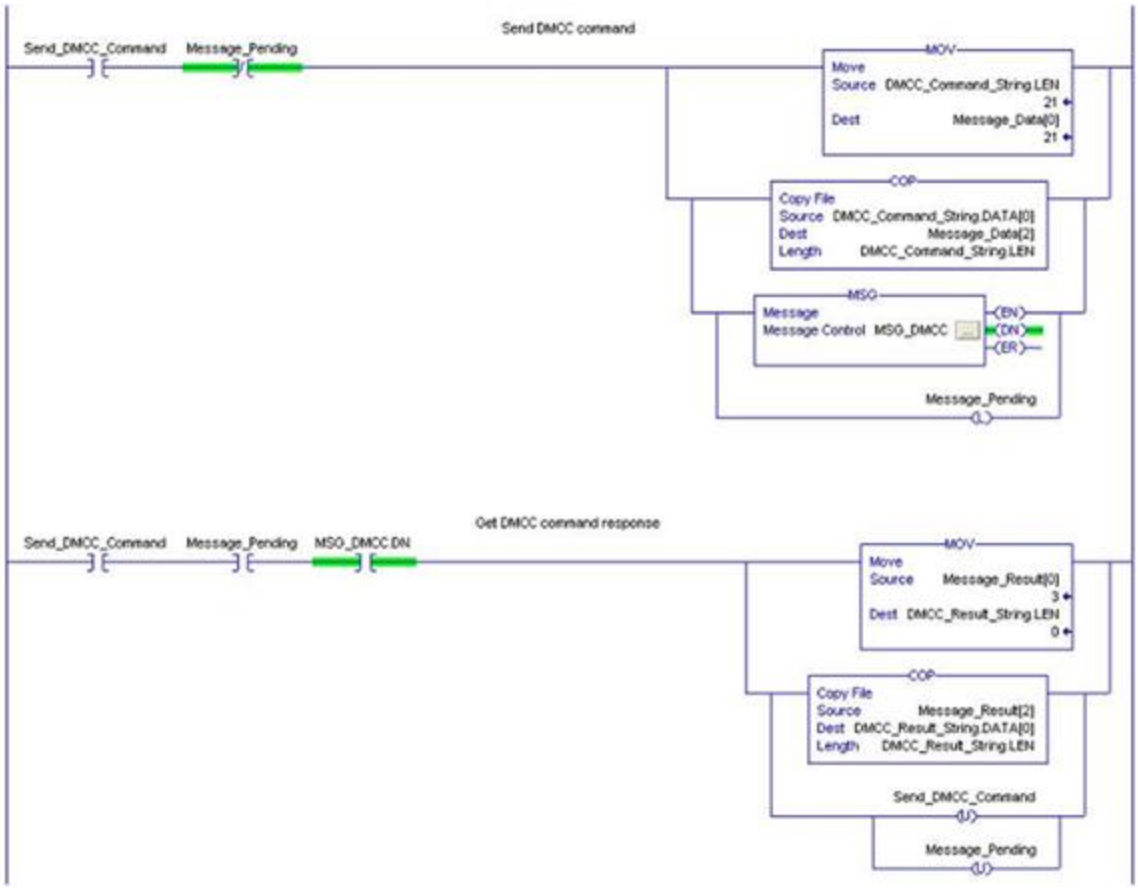
Note: The instruction to send a DMCC is intended as a demonstration of SLX explicit messaging behavior. This same operation could be written in a much more efficient ladder but would be less useful as a learning tool.

1. Add the following tags to the ControlLogix Controller Tags dialog:

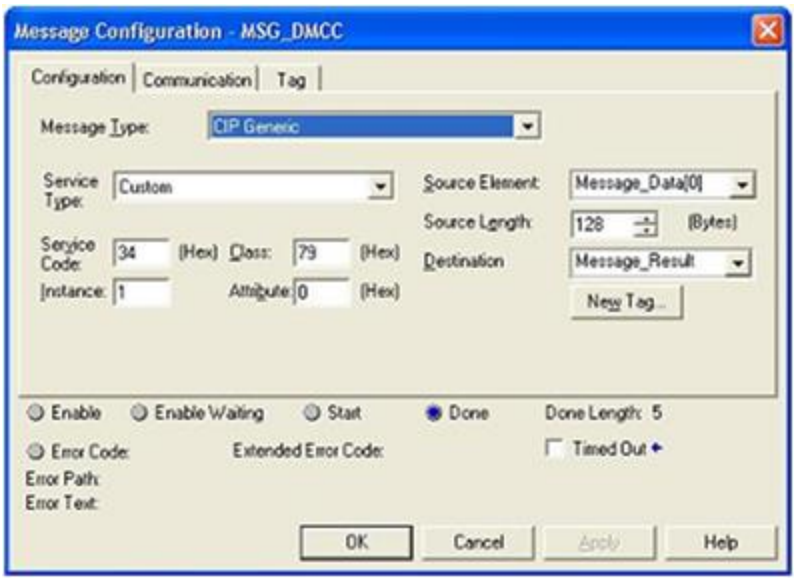


- **Send_DMCC_Command**: Boolean flag used to initiate the command.
- **DMCC_Command_String**: String containing the DMCC command to execute.
- **DMCC_Result_String**: String receiving the DMCC command results.
- **Message_Data**: Temp buffer holding the data to send via the MSG instruction.
- **Message_Result**: Temp buffer holding the data received via the MSG instruction.
- **Message_Pending**: Boolean flag used to indicate that a message is in process.
- **MSG_DMCC**: Data structure required by the Logix MSG instruction.

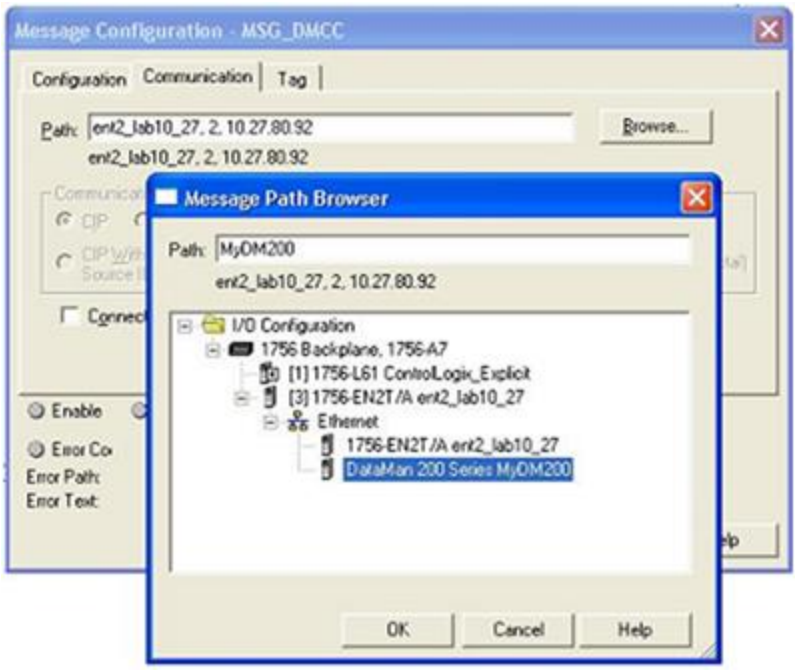
2. Add the following two rungs to the MainRoutine of your ControlLogix project:



3. Edit the MSG instruction. Configure it for “CIP Generic”, service 0x34 “SendDMCC”, class 0x79 “ID Reader Object” and instance 1. Set the source to “Message_Data” and the destination to “Message_Result”.



4. On the MSG instruction **Communication** tab, browse for and select the SLX device which you added to the project I/O Configuration tree. This tells Logix where to send the explicit message.



5. Download the project to the ControlLogix and place in “Run Mode”.

To operate:

1. Place a DMCC command in the “DMCC_Command_String” tag. For example “||>SET EXPOSURErl”. Note the rl at the end of the string. This is how Logix represents a CRLF.
2. Toggle the “Send_DMCC_Command” tag to 1.
3. When the “Send_DMCC_Command” tag goes back to 0 execution is complete. The DMCC command results can be found in “DMCC_Result_String”.

PROFINET

PROFINET is an application-level protocol used in industrial automation applications. This protocol uses standard Ethernet hardware and software to exchange I/O data, alarms, and diagnostics.

The device supports PROFINET I/O. This is one of the two “views” contained in the PROFINET communication standard. PROFINET I/O performs cyclic data transfers to exchange data with Programmable Logic Controllers (PLCs) over Ethernet. The second “view” in the standard, PROFINET Component Based Automation (CBA), is not supported.

By default, the device has the PROFINET protocol disabled. The protocol can be enabled in the **Communications** application step.

Tip: You can download the General Station Description (GSD) file for your device from the Cognex Support Site.

Preparing to Use PROFINET

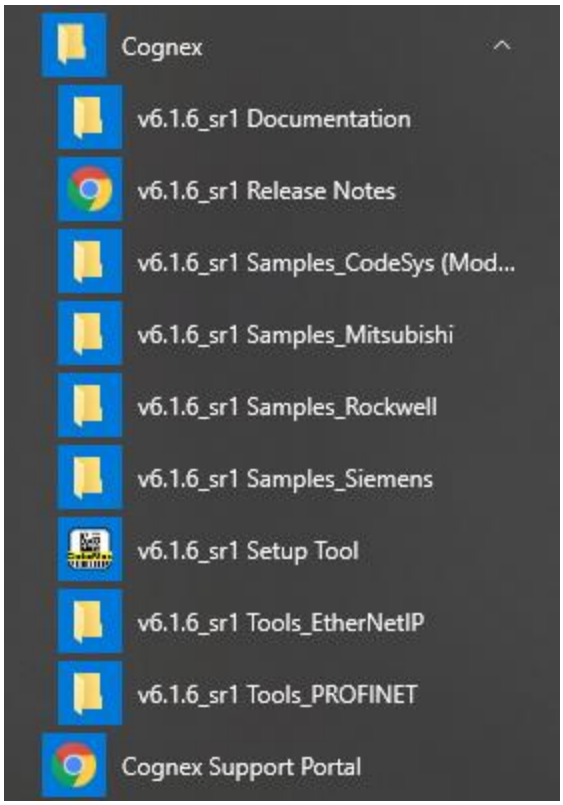
Preparing to use PROFINET involves the following main steps:

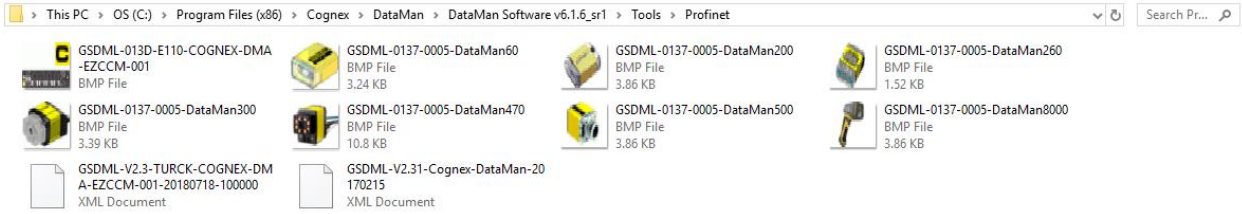
- Make sure that you have either the SIMATIC Step 7 programming software or TIA Portal installed.
- Set up the Siemens Software tool so that it recognizes your device device.
- Install the Generic Station Description (GSD) file.

Note: Expanding the process image can have a performance impact on the PLC scan cycle time. If your scan time is critical, use the minimal acceptable module sizes and manually remap them down lower in the process image.

Perform the following steps to set up PROFINET using SIMATIC or TIA Portal:

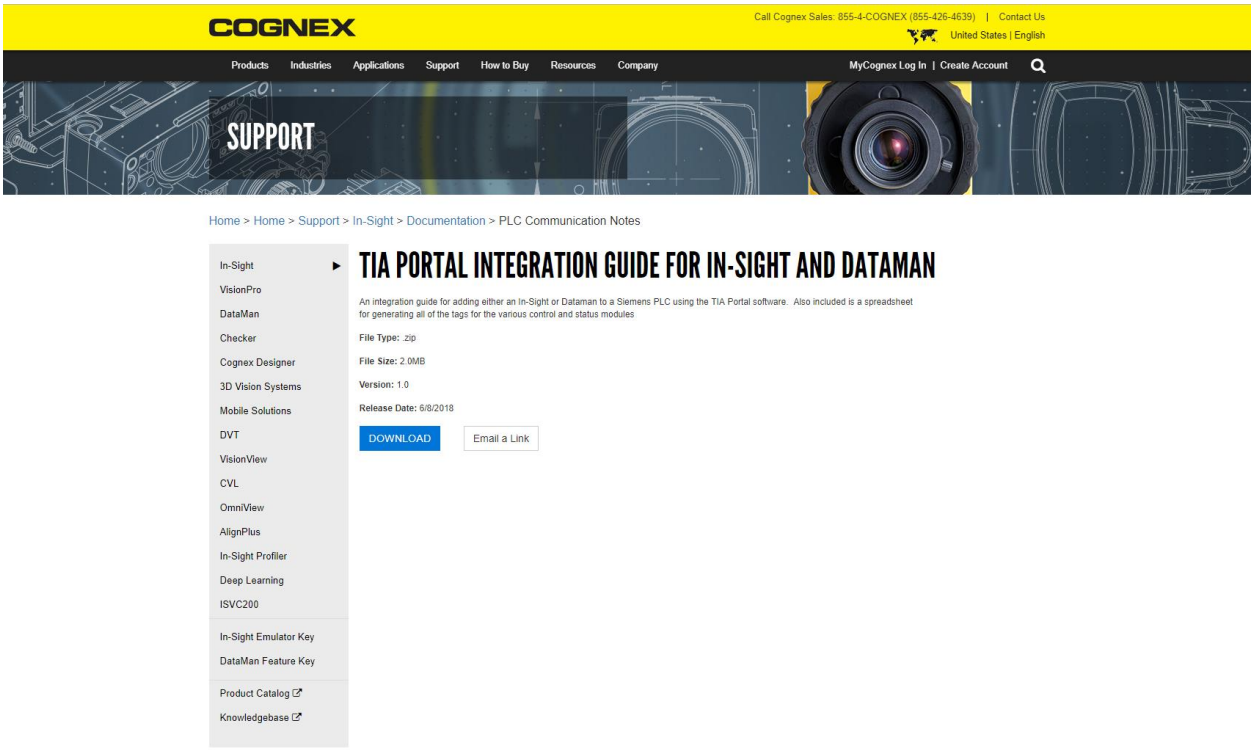
1. Verify that SIMATIC is on your machine.
2. From the Windows **Start** menu, go to Cognex and select the folder that contains the PROFINET tools.



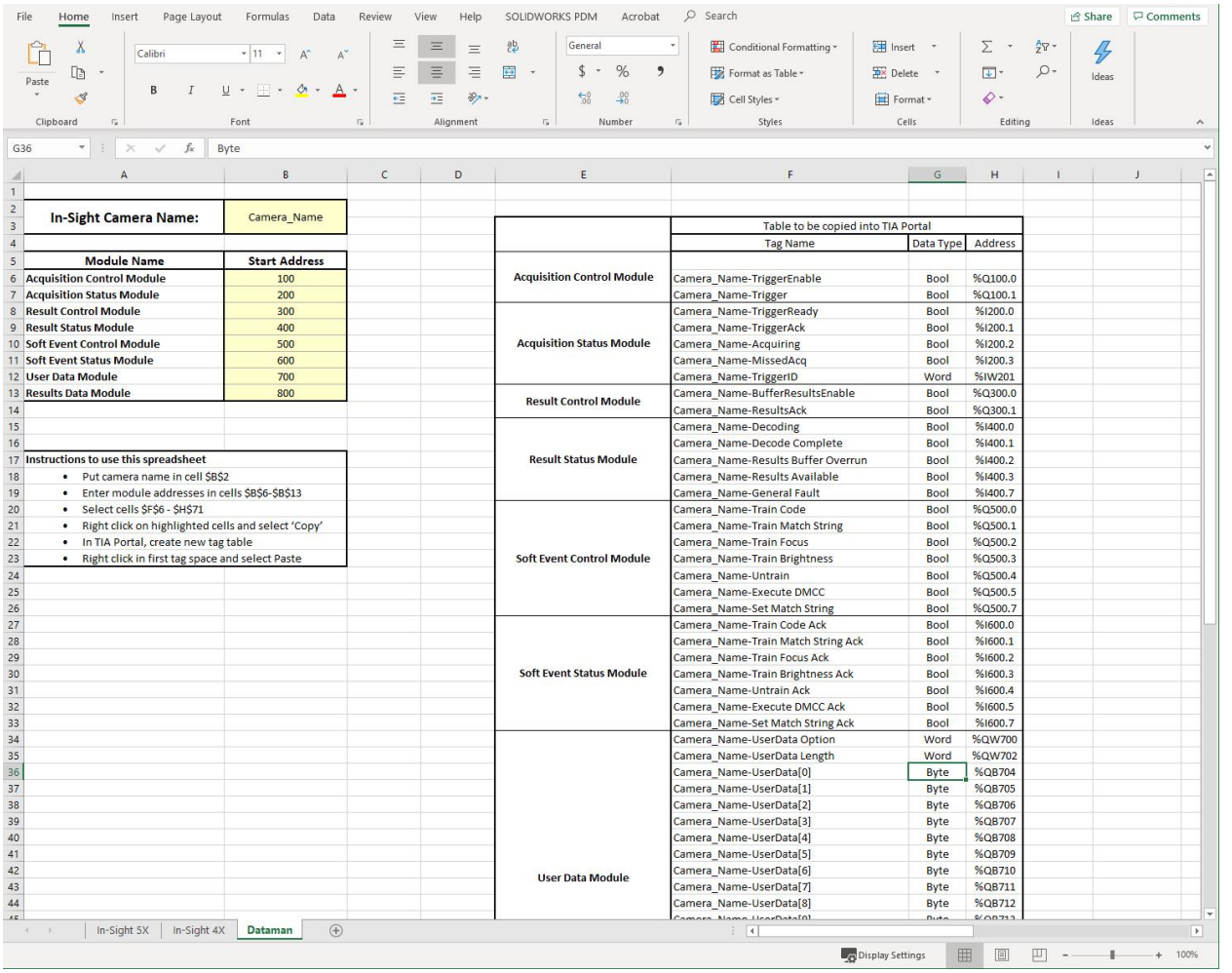


3. Ensure that the device firmware is up-to-date.
4. Ensure that the **PROFINET** option is enabled in the **Communications** application step.

5. Download the *TIA Portal Integration Guide* from https://support.cognex.com/en/downloads/detail/in-sight/3687/1033.



6. Open the *PROFINET Tag Generator* file that is included in the *TIA Portal Integration Guide*.



In-Sight Camera Name:	Camera_Name

Module Name	Start Address
Acquisition Control Module	100
Acquisition Status Module	200
Result Control Module	300
Result Status Module	400
Soft Event Control Module	500
Soft Event Status Module	600
User Data Module	700
Results Data Module	800

Instructions to use this spreadsheet

- Put camera name in cell B2
- Enter module addresses in cells B6-B13
- Select cells F6 - H71
- Right click on highlighted cells and select 'Copy'
- In TIA Portal, create new tag table
- Right click in first tag space and select Paste

	Table to be copied into TIA Portal		
	Tag Name	Data Type	Address
Acquisition Control Module	Camera_Name-TriggerEnable	Bool	%Q100.0
	Camera_Name-Trigger	Bool	%Q100.1
Acquisition Status Module	Camera_Name-TriggerReady	Bool	%I200.0
	Camera_Name-TriggerAck	Bool	%I200.1
	Camera_Name-Acquiring	Bool	%I200.2
	Camera_Name-MissedAcq	Bool	%I200.3
	Camera_Name-TriggerID	Word	%IW201
Result Control Module	Camera_Name-BufferResultsEnable	Bool	%Q300.0
	Camera_Name-ResultsAck	Bool	%Q300.1
Result Status Module	Camera_Name-Decoding	Bool	%I400.0
	Camera_Name-Decode Complete	Bool	%I400.1
	Camera_Name-Results Buffer Overrun	Bool	%I400.2
	Camera_Name-Results Available	Bool	%I400.3
	Camera_Name-General Fault	Bool	%I400.7
Soft Event Control Module	Camera_Name-Train Code	Bool	%Q500.0
	Camera_Name-Train Match String	Bool	%Q500.1
	Camera_Name-Train Focus	Bool	%Q500.2
	Camera_Name-Train Brightness	Bool	%Q500.3
	Camera_Name-Untrain	Bool	%Q500.4
	Camera_Name-Execute DMCC	Bool	%Q500.5
	Camera_Name-Set Match String	Bool	%Q500.7
Soft Event Status Module	Camera_Name-Train Code Ack	Bool	%I600.0
	Camera_Name-Train Match String Ack	Bool	%I600.1
	Camera_Name-Train Focus Ack	Bool	%I600.2
	Camera_Name-Train Brightness Ack	Bool	%I600.3
	Camera_Name-Untrain Ack	Bool	%I600.4
	Camera_Name-Execute DMCC Ack	Bool	%I600.5
	Camera_Name-Set Match String Ack	Bool	%I600.7
User Data Module	Camera_Name-UserData Option	Word	%QW700
	Camera_Name-UserData Length	Word	%QW702
	Camera_Name-UserData[0]	Byte	%QB704
	Camera_Name-UserData[1]	Byte	%QB705
	Camera_Name-UserData[2]	Byte	%QB706
	Camera_Name-UserData[3]	Byte	%QB707
	Camera_Name-UserData[4]	Byte	%QB708
	Camera_Name-UserData[5]	Byte	%QB709
	Camera_Name-UserData[6]	Byte	%QB710
	Camera_Name-UserData[7]	Byte	%QB711
	Camera_Name-UserData[8]	Byte	%QB712

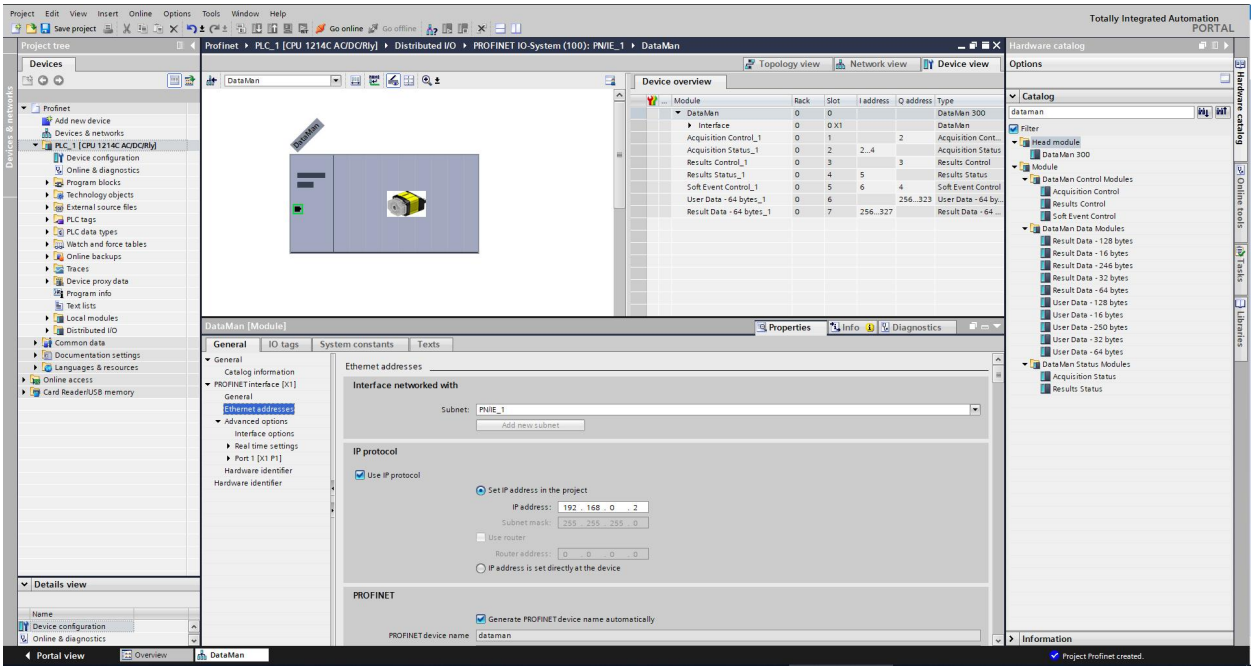
7. From the Windows **Start** menu, launch the SIMATIC Manager or TIA Portal.

You now have all the resources ready to integrate your device into your PLC project.

PROFINET Object Model

The PROFINET implementation on the device consists of seven I/O modules:

- Acquisition Control Module
- Acquisition Status Module
- Results Control Module
- Results Status Module
- SoftEvent Control Module
- User Data Module
- Result Data Module



Acquisition Control Module

Controls image acquisition. This module consists of data sent from the PLC to the device.

Slot number: 1

Total Module size: 1 byte

Bit	Name	Description
0	Trigger Enable	Setting this bit enables triggering via PROFINET. Clearing this bit disables triggering.
1	Trigger	Setting this bit triggers an acquisition when the following conditions are met: • Trigger Enable is set. • No acquisition is currently in progress. • The device is ready to trigger.
2-7	Reserved	Reserved for future use

Acquisition Status Module

Indicates the current acquisition status. This module consists of data sent from thedevice to the PLC.

Slot number: 2

Total Module size: 3 bytes

Bit	Name	Description
0	Trigger Ready	Indicates when the device is ready to accept a new trigger. Bit is True when “Trigger Enable” is set and the device is ready to accept a new trigger.
1	Trigger Ack	Indicates that thedevice has received a new Trigger. This bit remains True as long as the “Trigger” bit remains True (that is, it is interlocked with the Trigger bit).
2	Reserved	Reserved for future use.

3	Missed Ack	Indicates that the device was unable to successfully trigger an acquisition. Bit is cleared when the next successful acquisition occurs.
4-7	Reserved	Reserved for future use.
8-23	Trigger ID	ID value of the next trigger to be issued (16-bit integer). Used to match issued triggers with corresponding result data received later. This same value is returned in ResultID of the result data.

Note: The Missed Ack bit in the Acquisition Status Register will only be set if an acquisition triggered from the Acquisition Control Module could not get executed.

Results Control Module

Controls the processing of result data. This module consists of data sent from the PLC to the device.

Slot number: 3

Total Module size: 1 byte

Bit	Name	Description
0	Results Buffer Enable	Not supported on SLX devices.
1	Results Ack	Bit is used to acknowledge that the PLC has successfully read the latest result data. When set True the “Result Available” bit will be cleared. If result buffering is enabled, the next set of result data will be pulled from the queue and “Result Available” will again be set True.
2-7	Reserved	Reserved for future use.

Results Status Module

Indicates the acquisition and result status. This module consists of data sent from the device to the PLC.

Slot number: 4

Total Module size: 1 byte

Bit	Name	Description
0	Decoding	Indicates that the device is decoding an acquired image. The value of the **Decoding** bit is identical to the value of the **Acquiring** bit. Both bits are true if an acquisition, or series of acquisitions of the same trigger, is in progress.
1	Decode Complete	Bit is toggled on the completion of a decode operation when the new results are made available (0 -> 1 or 1 -> 0).
2	Result Buffer Overrun	Not supported on SLX devices.
3	Results Available	Indicates that a new set of read results are available, that is, the contents of the Result Data module are valid. Cleared when the results are acknowledged.
4-6	Reserved	Reserved for future use
7	General Fault	Indicates that a fault has occurred.

SoftEvent Control Module

Used to initiate a SoftEvent and receive acknowledgment of completion. Note that this is a bi-directional I/O module. Module data sent from the PLC initiates the SoftEvent. Module data sent by the device acknowledges completion.

Slot number: 5

Total Module size: 1 byte (input) and 1 byte (output)

Data written from the PLC to device:

Bit	Name	Description
0	Train Code	Not supported on SLX devices.
1	Train Match String	Not supported on SLX devices.
2	Train Focus	Not supported on SLX devices.
3	Train Brightness	Not supported on SLX devices.
4	Untrain	Not supported on SLX devices.
5	Reserved	Reserved for future use
6	Execute DMCC	Bit transition from 0 -> 1 causes the DMCC operation to be invoked. Note that a valid DMCC command string must first be placed in “User Data” before invoking this event.
7	Set Match String	Not supported on SLX devices.

Data written from the device to PLC:

Bit	Name	Description
0	Train Code Ack	Not supported on SLX devices.
1	Train Match String Ack	Not supported on SLX devices.
2	Train Focus Ack	Not supported on SLX devices.
3	Train Brightness Ack	Not supported on SLX devices.
4	Untrain Ack	Not supported on SLX devices.
5	Reserved	Reserved for future use
6	Execute DMCC Ack	Indicates that the “Execute DMCC” operation is complete
7	Set Match String Ack	Not supported on SLX devices.

User Data Module

Data sent from a PLC to adevice to support acquisition, decode and other special operations. Currently this module is only used to support the “Execute DMCC” and “Set Match String” SoftEvents.

Note: There are 5 versions of the User Data module. Only one instance can be configured for use in a given application. The “User Data Option” and “User Data Length” fields are the same for each module. The “User Data” field varies in size based on the selected module. Choose the module which is large enough to exchange the amount of data your application requires.

Slot number: 6

Total Module size:

4 + 16 (16 bytes of User Data)

4 + 32 (32 bytes of User Data)

4 + 64 (64 bytes of User Data)

4 + 128 (128 bytes of User Data)

4 + 250 (250 bytes of User Data)

Byte	Name	Description
0-1	User Data Option	Not supported on SLX devices.
2-3	User Data Length	Number of bytes of valid data actually contained in the “User Data” field (16-bit Integer).
4...	User Data	Data sent from the PLC to the device to support acquisition, decoding, and other special operations (array of bytes).

Result Data Module

Read result data sent from a device to a PLC.

Look into the Result Data Module for non-PROFINET related data if there is an acquisition problem on the device. The "Result Code" part contains information about trigger overruns or buffer overflows that have occurred on the device.

Note: There are 5 versions of the Result Data module. Only a single instance can be configured for use in a given application. The “Result ID”, “Result Code”, “Result Extended” and “Result Length” fields are the same for each module. The “Result Data” field varies in size based on the selected module. Choose the module which is large enough to exchange the amount of result data your application requires.

Slot number: 7

Total Module size:

8 + 16 (16 bytes of Result Data)

8 + 32 (32 bytes of Result Data)

8 + 64 (64 bytes of Result Data)

8 + 128 (128 bytes of Result Data)

8 + 246 (246 bytes of Result Data)

Byte	Name	Description
0-1	Result ID	The value of the “Trigger ID” when the trigger that generated these results was issued. Used to match up triggers with corresponding result data (16-bit Integer).
2-3	Result Code	Indicates the success or failure of the read that produced these results. A result of 1 means good read. A result of 0 means no read.
4-5	Result Extended	Currently unused (16-bit Integer).
6-7	Result Length	Actual number of bytes of read data contained in the “Result Data” field (16-bit Integer).
8...	Result Data	Decoded read result data (array of bytes)

Operation

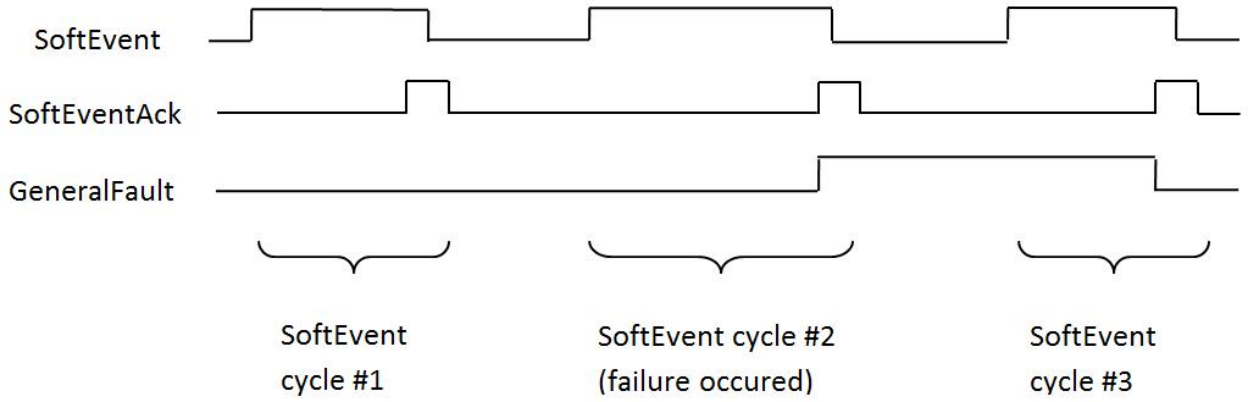
SoftEvents

SoftEvents act as “virtual” inputs. When the value of a SoftEvent changes from 0 -> 1 the action associated with the event will be executed. When the action completes the corresponding SoftEventAck bit will change from 0 -> 1 to signal completion. The acknowledge bit will change back to 0 when the corresponding SoftEvent bit is set back to 0.

The “ExecuteDMCC” SoftEvent action requires user supplied data. This data must be written to the UserData and UserDataLength area of the UserData Module prior to invoking the SoftEvent. Only one SoftEvent can be invoked at a time because these SoftEvents depend on the UserData.

General Fault Indicator

When a communication related fault occurs the “GeneralFault” bit will change from 0 -> 1. Currently the only fault conditions supported are SoftEvent operations. If a SoftEvent operation fails, the fault bit will be set. The fault bit will remain set until the next SoftEvent operation or until triggering is disabled and again re-enabled.



Acquisition Sequence

The device can be triggered to acquire images by several methods. It can be done explicitly by manipulating the Trigger bit of the Acquisition Control Module, it can be triggered by external hardwired input, and it can be triggered via DMCC. This section describes manipulating the Acquisition Control Module bits.

On startup the “Trigger Enable” bit will be False. It must be set to True to enable triggering. When the device is ready to accept triggers, the “Trigger Ready” bit will be set to True.

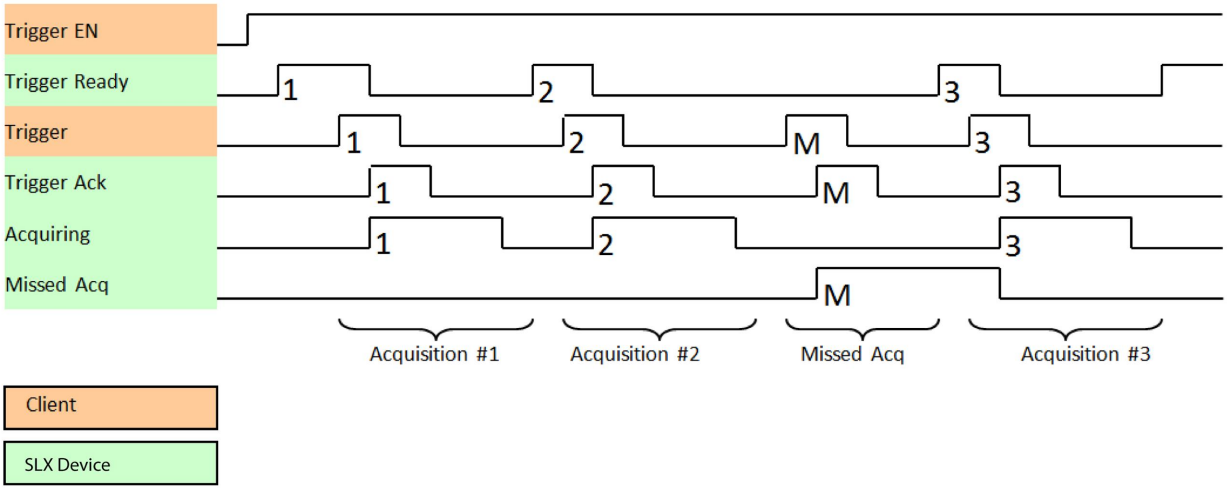
While the Trigger Ready bit is True, each time the device detects the “Trigger” bit change from 0 to 1, it initiates an image acquisition. Make sure that the client (PLC) holds the bit in the new state until that same state value is seen back in the Trigger Ack bit. This is a necessary handshake to guarantee that the device detects the change.

During an acquisition, the Trigger Ready bit will be cleared and the Acquiring bit will be set to True. When the acquisition is completed, the Acquiring bit will be cleared. The Trigger Ready bit is again set to True once the device is ready to begin a new image acquisition.

The device allows acquisitions to overlap with the decoding of previous acquisitions. The Trigger Ready bit is set high after the acquisition is complete, while decoding may still be in progress. The Decoding bit is deprecated and only mirrors the behavior of the Acquiring bit. If trigger queuing is active or other trigger sources can interfere, the Trigger Ready bit may not be reliable.

To force a reset of the trigger mechanism, set the Trigger Enable bit to False, until the Trigger Ready bit is 0. Then, Trigger Enable can be set to True to re-enable acquisition.

As a special case, an acquisition can be cancelled by clearing the Trigger signal before the read operation is complete. This allows cancelling reads in Presentation and Manual mode if no code is in the field of view. To ensure that a read is not unintentionally cancelled, it is advised that the PLC hold the Trigger signal True until both TriggerAck and ResultsAvailable are True or DecodeComplete toggles state.



Decode / Result Sequence

After an image is acquired, it is decoded. When the decoding is complete and a result is ready to be delivered, the device toggles the DecodeComplete bit.

Decode results are reported asynchronously to the Result Status module.

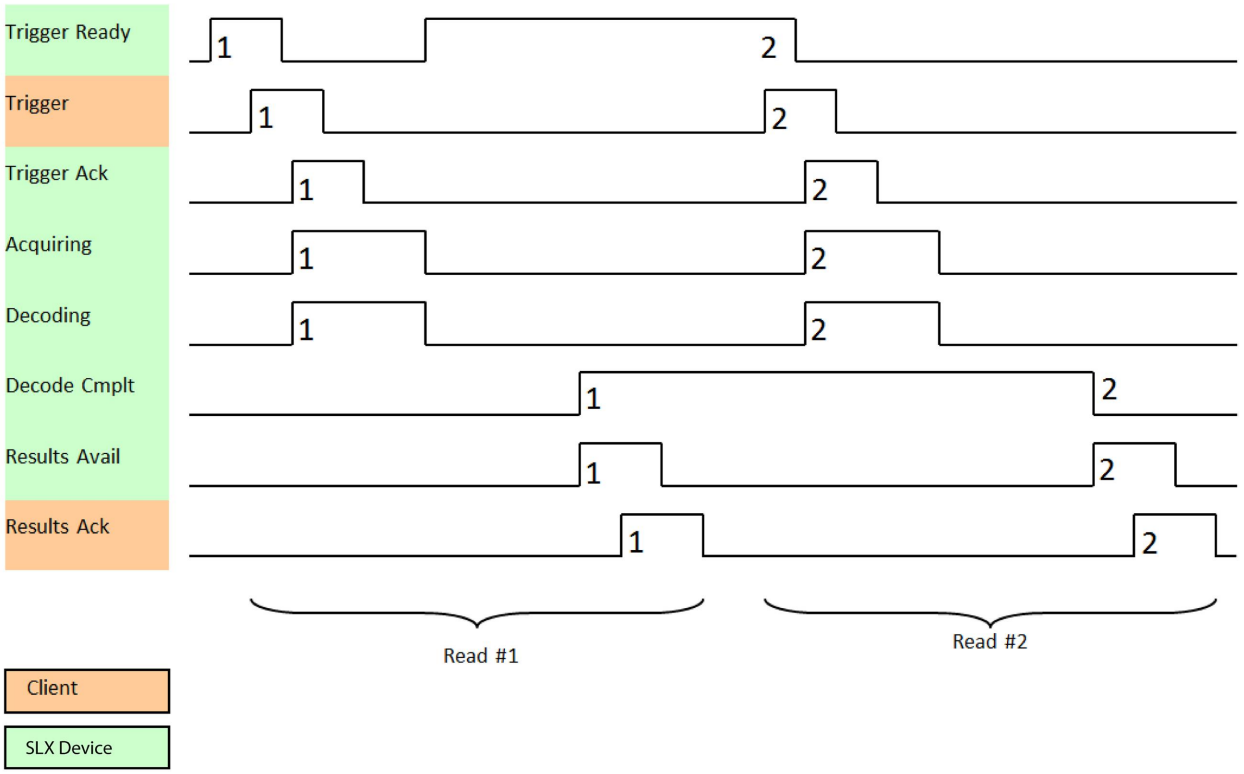
If BufferResultsEnable is set to False, then the decode results are immediately placed into DecodeResults, and the Results Available bit is set to True.

Note: The only way to ensure that you are not losing results is to use BufferResults. Make sure that your application can read each DecodeResult in time if you do not use BufferResultsEnable.

If the Results Buffer Enable bit is set to True, the new results are queued. The earlier decode results remain in the Results Module until the client acknowledges them by setting the “Results Ack” bit to True. After the Results Available bit is cleared, make sure that the client sets the Results Ack bit back to False to allow the next queued results to be placed into the Results Module. This is a necessary handshake to ensure the results are received by the client (PLC), even if short gaps occur between results.

Behavior of DecodeStatusRegister

Bit	Bit Name	Results if Buffering Disabled	Results if Buffering Enabled
0	Decoding	Set when acquiring and decoding an image. The value of the Decoding bit is always the same as the Acquiring bit.	Set when acquiring and decoding an image. The value of the Decoding bit is always the same as the Acquiring bit.
1	Decode Complete	Toggled on completion of an image decode.	Toggled on completion of an image decode.
2	Results Buffer Overflow	Remains set to zero.	Set when decode results could not be queued because the client failed to acknowledge a previous result. Cleared when the decode result is successfully queued.
3	Results Available	Set when new results are placed in the Results Module. Stays set until the results are acknowledged by setting Results Ack to true.	Set when new results are placed in the Results Module. Stays set until the results are acknowledged by setting Results Ack to true.



SIMATIC Examples

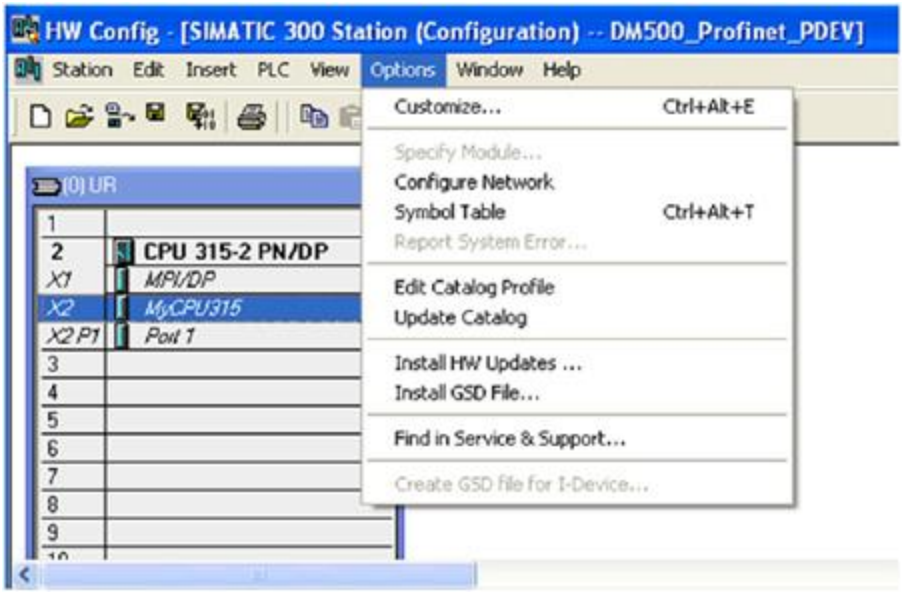
This section gives some examples of using the device with a Siemens S7-300 PLC assuming that you are familiar with the S7-300 and the SIMATIC programming software.

Adding a Device to a SIMATIC Project

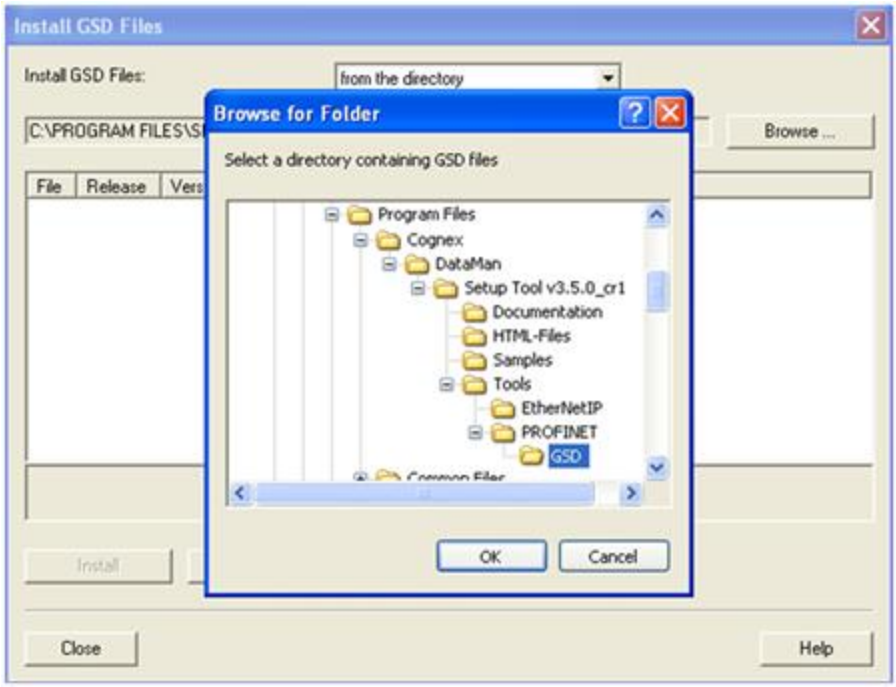
To add a device to a project:

1. If you already have a project, click **Cancel** to skip past the New Project wizard. Otherwise, let the wizard guide you through creating a new project.

2. In the opened project, double-click the Hardware icon to open the **HW Config** dialog. From the main menu, select **Options** -> **Install GSD File…**.

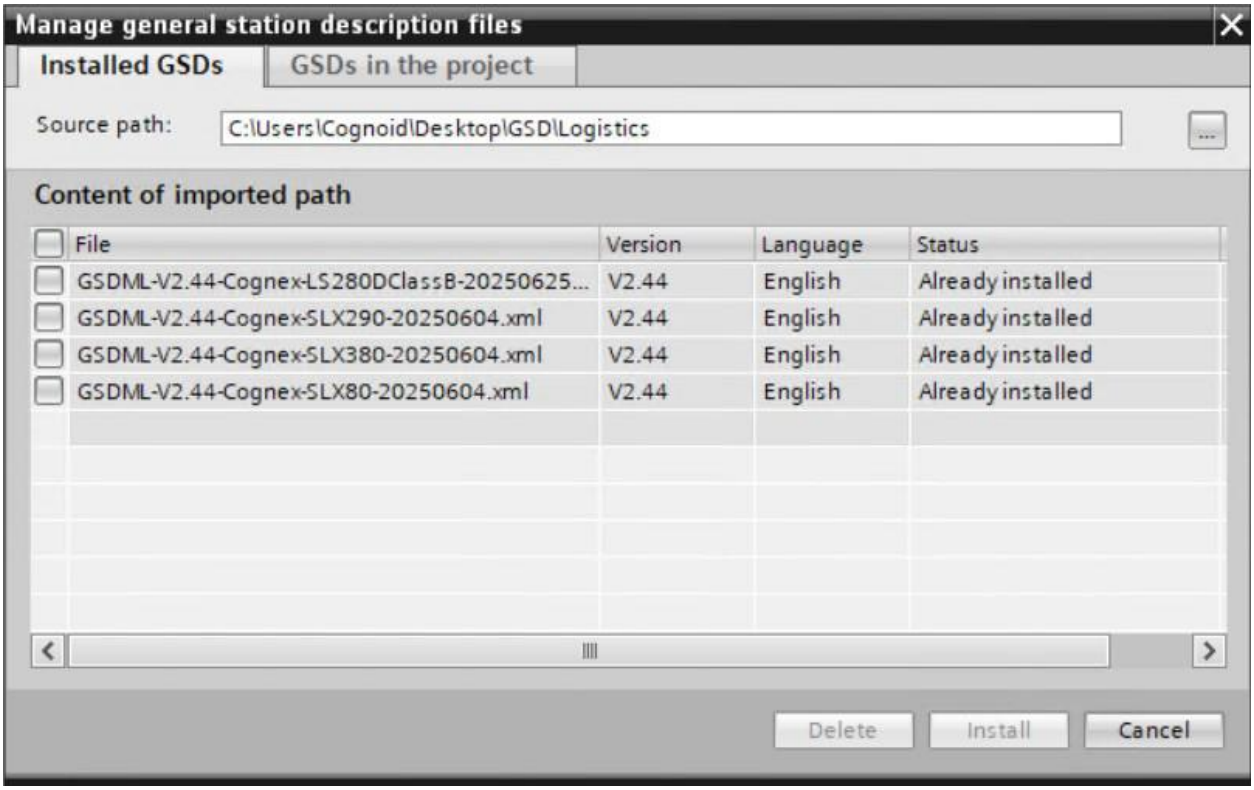


3. Browse to the installation folder of the GSD file (or the folder where you saved the GSD file if you downloaded it from the web).



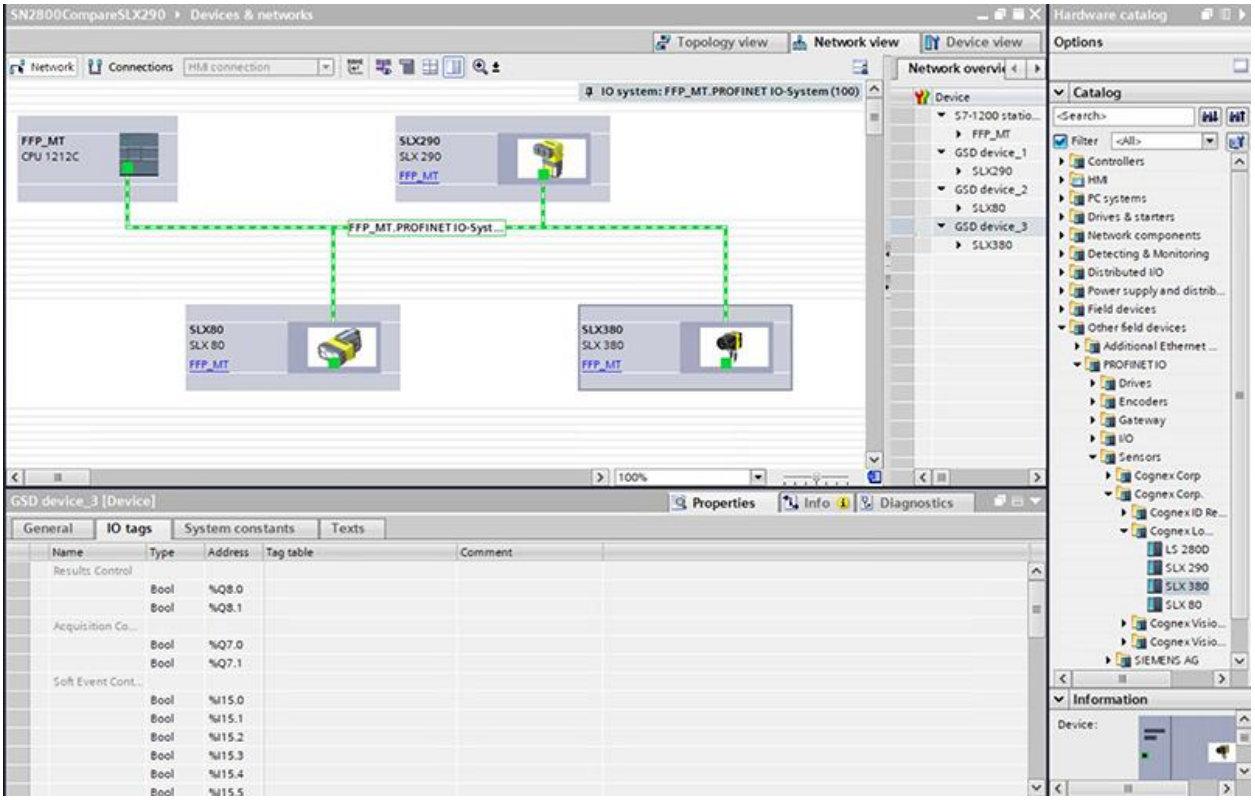
4. Select the GSD file you wish to install and follow the displayed instructions to complete the installation.

> **Note**: If there is more than one GSD file in the list, and you are unsure which to install, choose the one with the most recent date.



5. Add your device to your project. This makes the SLX device available in the Hardware Catalog. Launch the SIMATIC Hardware Config tool.
6. In the main menu, select View -> Catalog.
7. The catalog is displayed. Expand the **PROFINET IO** tree to the **Cognex ID Readers** node.

8. Press and hold the left mouse button to drag the device and drop it on the PROFINET IO network symbol in the left pane.

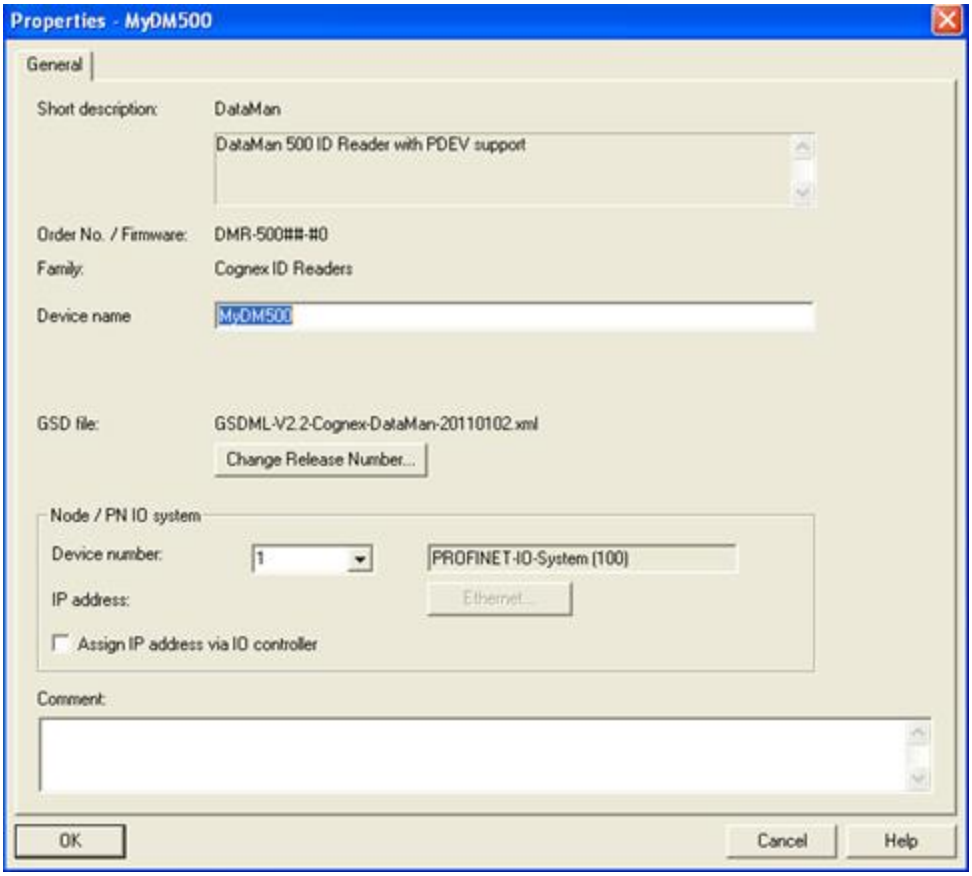


The HW Config tool automatically maps thedevice I/O modules into the memory space.

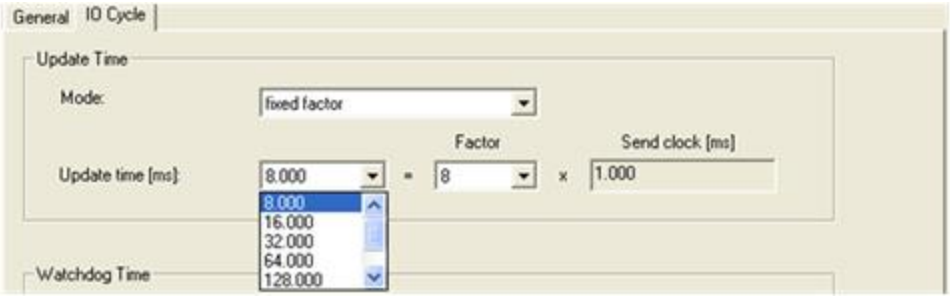
> **Note**: By default, the 64 byte User Data and 64 byte Result Data Modules are inserted. There are multiple sizes available for both of these modules. To optimize performance, use the module size that most closely matches the actual data requirements of your application. You can change the module by deleting the one in the table and inserting the appropriate sized module from the catalog.

9. Right-click the device icon and select **Object Properties…**.
10. Give thedevice a name. This must match the name of your actual device. The name must be unique and follow DNS naming conventions. For details, see the SIMATIC Software help.

11. If your device is configured to use its own static IP, uncheck **Assign IP address via IO controller**. As an alternative, if you wish the PLC to assign an IP address, select the Ethernet button and configure the appropriate address.



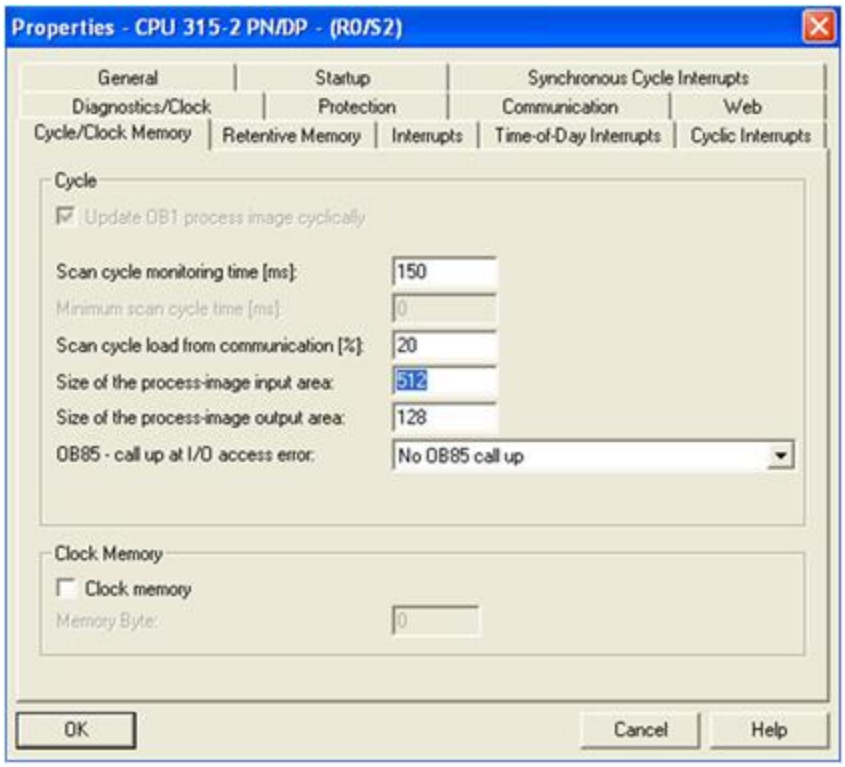
12. In the **IO Cycle** tab, select the appropriate cyclic update rate for your application.



Note: Cognex recommends using a minimum update time of 32 ms.

13. By default, the SIMATIC software maps the User Data and Result Data Modules to offset 256. This is outside of the default process image area size of 128. That is, data in these modules are inaccessible by some SFCs such as BLKMOV. As a solution, either remap the modules to lower offsets within the process image area or expand the process image area to include these modules.

If you choose to expand the process image area, make the size large enough for the module size plus the default 256 offset.



Note: Expanding the process image can have a performance impact on the PLC scan cycle time. If your scan time is critical, use the minimal acceptable module sizes and manually remap them down lower in the process image.

Triggering and Getting Results in SIMATIC

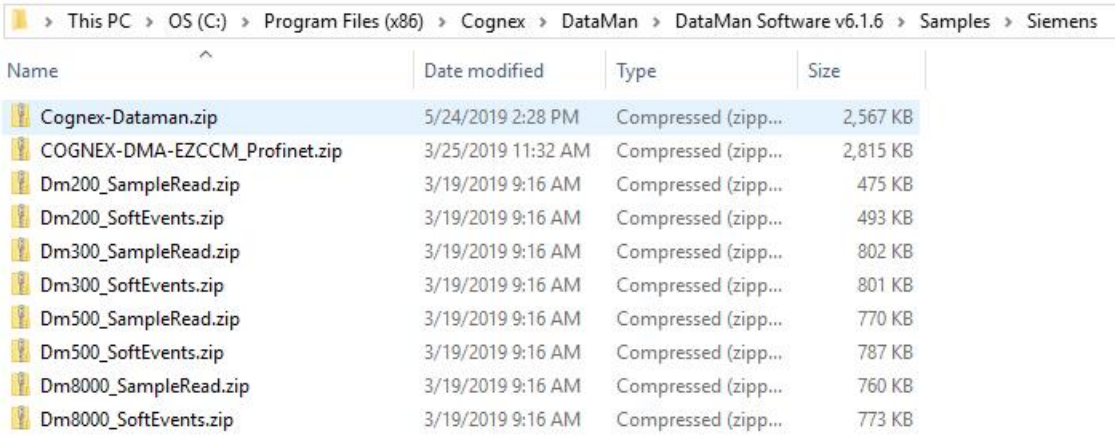
Run the sample program “DM200_SampleRead” for the complete example program.

Note: This sample can be used with any PROFINET enabled device.

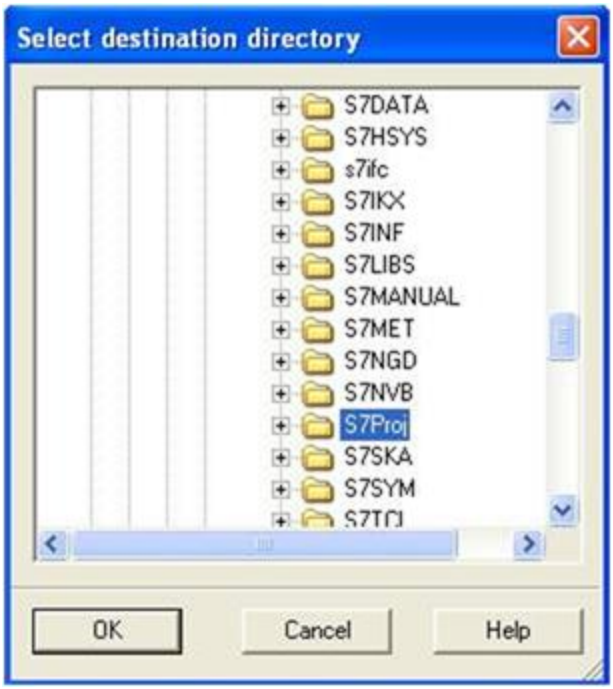
Perform the following steps to install the program:

1. Start the SIMATIC Manager software.
2. Close any open applications.
3. From the main menu, select **File -> Retrieve…**

4. Browse to find the sample file on your PC.



5. Save the project on your PC.



6. The Siemens software extracts the sample archive and makes it available.

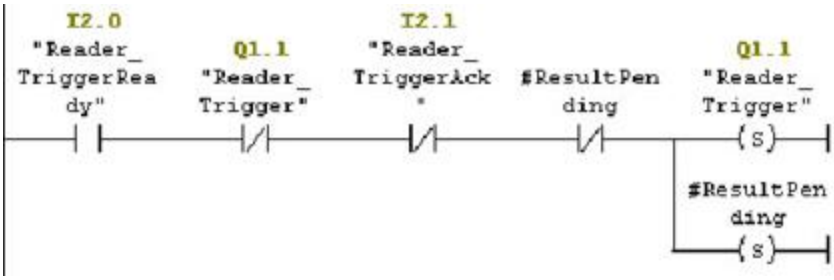
Reduced to the basics, the process of reading and retrieving results consists of the following:

1. Define an area in your application to save read results. There are many options regarding how and where result data can be stored. In this example, a Data Block (DB) is defined containing the fields of the Result Data module that are relevant for our application.

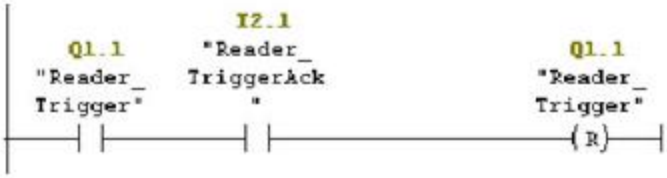
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	ID	INT	0	ID of this read result
+2.0	Flags	INT	0	Flags indicating success or failure
+4.0	Length	INT	0	Number of data bytes in the array Value[]
+6.0	Value	ARRAY[1..63]		Code value read
*1.0		CHAR		
=70.0		END_STRUCT		

2. Enable the device.

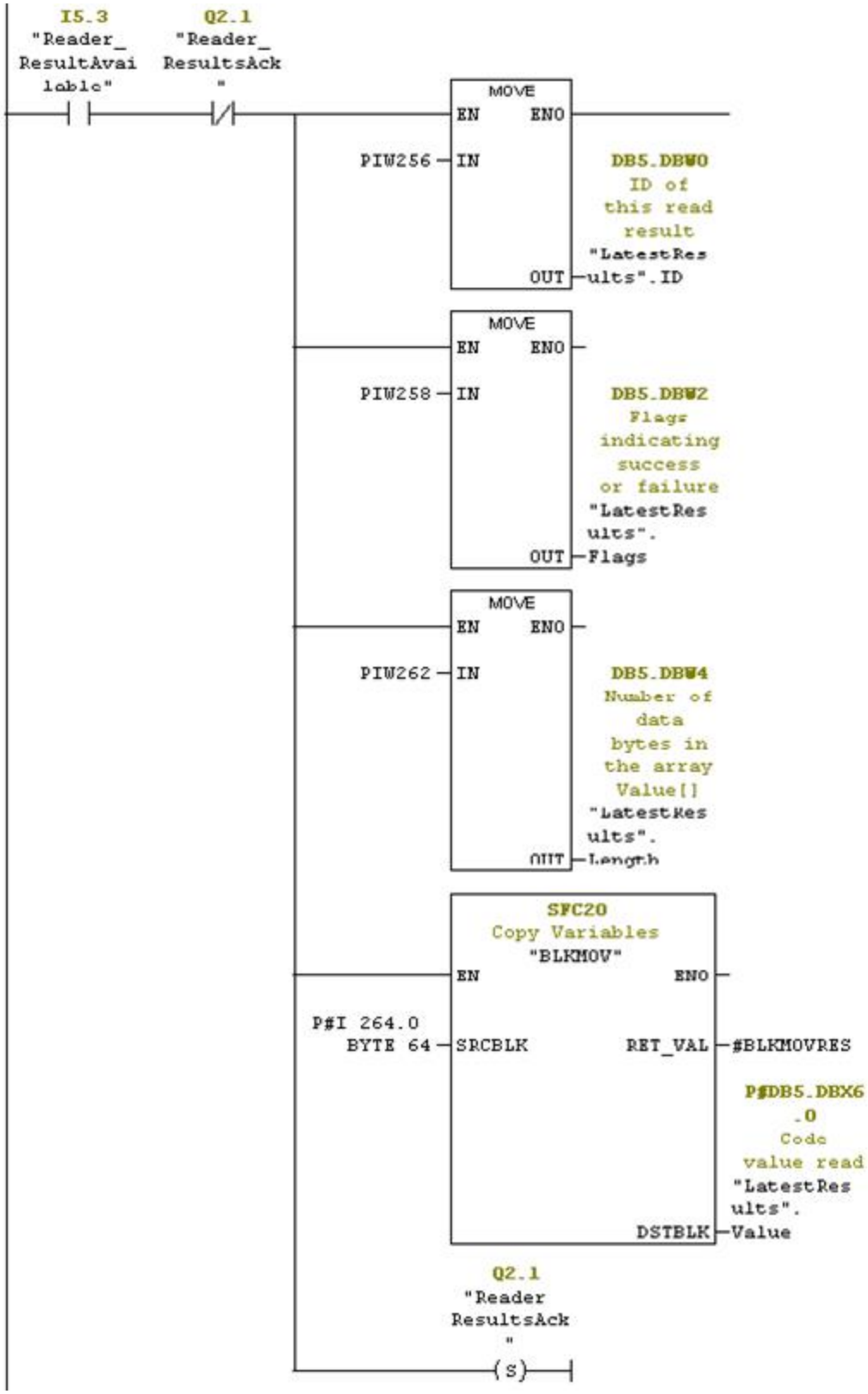
3. Set the trigger signal and set coil to indicate a read is pending.



4. As soon as the trigger signal is acknowledged, clear the trigger signal.

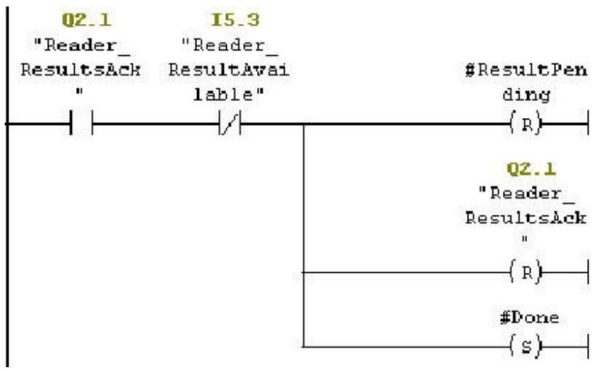


5. As soon as the results are available, save a copy of the result data and set the results acknowledge signal.



When the device sees the result acknowledge signal, clear result acknowledge, clear the read pending coil, and signal that the read process is complete.

The device clears “Results Available” as soon as it sees the PLC’s “Results Ack” signal.



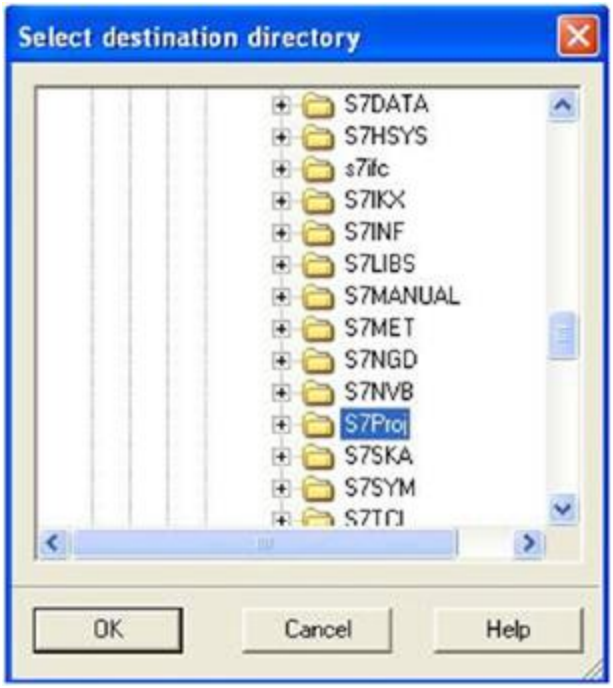
Using SoftEvents in SIMATIC

Run the sample program “DM200_SoftEvents” for the complete example program.

Note: This sample can be used with any PROFINET enabled device.

Perform the following steps to install the program:

1. Start the SIMATIC Manager software.
2. Close any open applications.
3. From the main menu, select **File -> Retrieve…**
4. Browse to find the sample file on your PC.
5. Select Dm200_SoftEvents.zip from the Siemens folder.
6. Select a destination folder to save the project on your PC.



7. The Siemens software extracts the sample archive and makes it available.

SoftEvents are a means of invoking an activity by manipulating a single control bit. The activity for each bit is predefined (for more information, see *SoftEvents* on page 66). With the exception of “Execute DMCC” and “Set Match String” all SoftEvents may be invoked in the same way. “Execute DMCC” and “Set Match String” require the added step of loading the User Data module with application data before invoking the event.

Reduced to the basics, the process of invoking a SoftEvent consists of the following:

```
FC3 : Train Focus
Initiate the "Train Focus" operation and monitor it to completion.

Network 1 : Title:
Issue "Train Focus" signal.

   Q3.2          I1.2                    Q3.2
 "Reader_      "Reader_                "Reader_
TrainFocus    TrainFocus              TrainFocus
    "             Ack"                    "
 ----|/|----------|/|--------------------(S)----

Network 2 : Title:
Release "Train Focus" signal as soon as it is acknowledged by the reader.

   I1.2                                  Q3.2
 "Reader_                              "Reader_
TrainFocus                            TrainFocus
   Ack"                                   "
 ----| |-------------------------------(R)----
                                       #Done
                                       ( )----

Network 3 : Title:
Set the return FC state.

Note, this will only return TRUE after the acknowledge has been received from
the reader. Otherwise it will return FALSE.

   #Done
 ----| |-------------------------------(SAVE)----
```

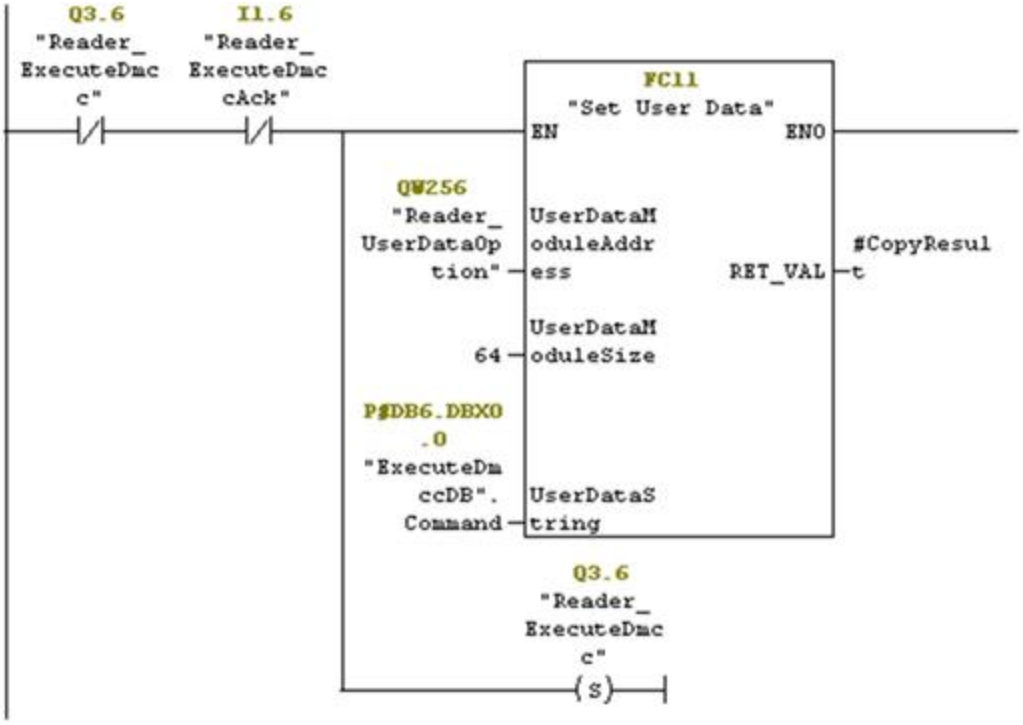
Executing DMCC Commands in SIMATIC

Refer to sample program “DM200_SoftEvents” for the complete example program. For information on how to install it, see *Using SoftEvents in SIMATIC* on page 78.

Note: This sample can be used with any PROFINET enabled device.

“Execute DMCC” is a SoftEvent which requires the added step of loading the User Data module with the desired DMCC command string before invoking the event. Note that the SoftEvent mechanism does not provide a means of returning DMCC response data other than a failure indication. This mechanism cannot be used for DMCC “||>GET…” commands.

The process of executing a DMCC command is the same for all other SoftEvents as in the example above except the step of invoking the SoftEvent also includes copying the command string to the User Data Module. In this example the command string is in a Data Block. This example can be expanded to utilize a Data Block with an array of command strings that the copy function can reference by an index value. It allows the user to pre-define all DMCC commands that are required by the application and invoke them by index.

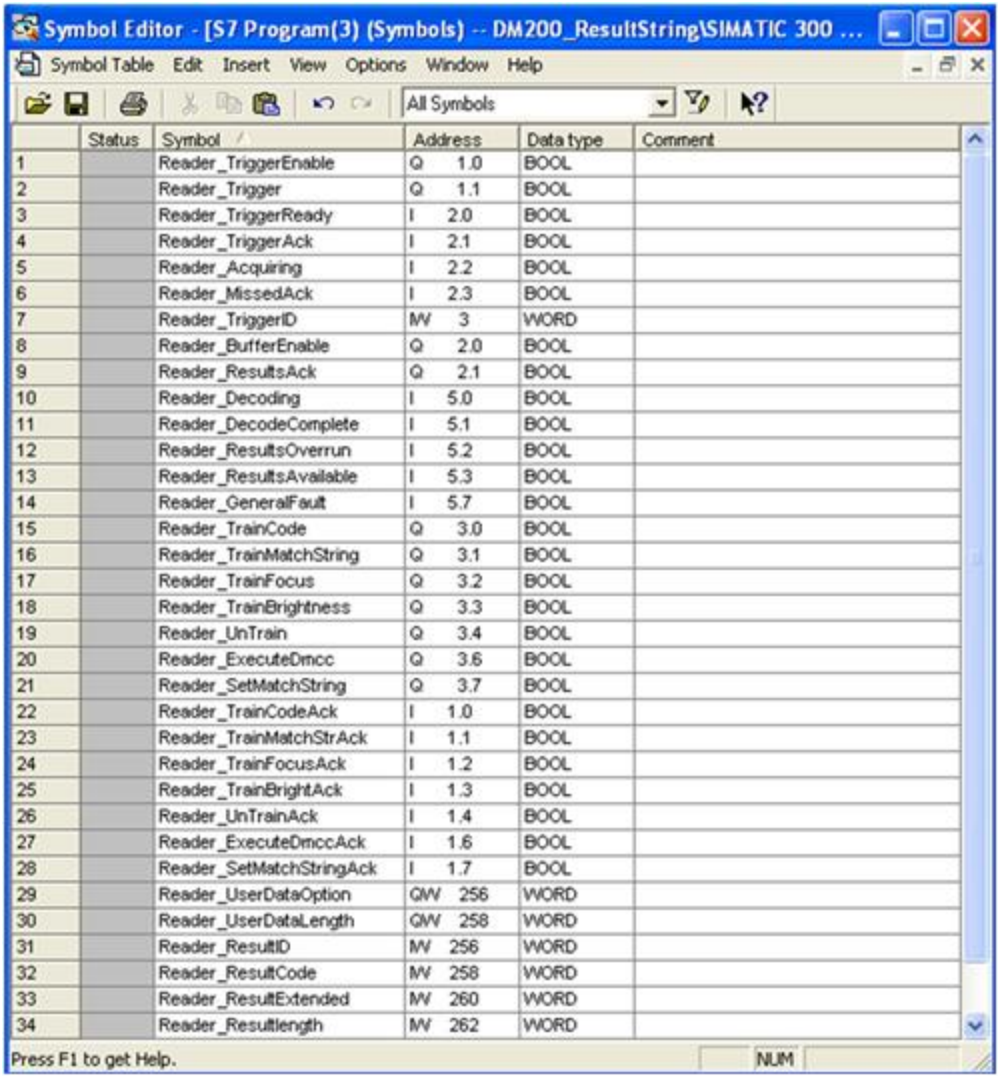


The function “Set User Data” (FC11) copies the provided string to the User Data module. Refer to the example program for the actual STL code.

SIMATIC Example Symbol Table

It is recommended that you define symbols for the device I/O module elements to make the code much easier to read and reduce mistakes. This sample table shows symbols defined for a typical instance of a SLX device. It is possible that device I/O modules are at different addresses in your project.

Make sure to adjust your symbol definitions based on the specific offsets of the I/O modules.



	Status	Symbol	Address	Data type	Comment
1		Reader_TriggerEnable	Q 1.0	BOOL	
2		Reader_Trigger	Q 1.1	BOOL	
3		Reader_TriggerReady	I 2.0	BOOL	
4		Reader_TriggerAck	I 2.1	BOOL	
5		Reader_Acquiring	I 2.2	BOOL	
6		Reader_MissedAck	I 2.3	BOOL	
7		Reader_TriggerID	MW 3	WORD	
8		Reader_BufferEnable	Q 2.0	BOOL	
9		Reader_ResultsAck	Q 2.1	BOOL	
10		Reader_Decoding	I 5.0	BOOL	
11		Reader_DecodeComplete	I 5.1	BOOL	
12		Reader_ResultsOverrun	I 5.2	BOOL	
13		Reader_ResultsAvailable	I 5.3	BOOL	
14		Reader_GeneralFault	I 5.7	BOOL	
15		Reader_TrainCode	Q 3.0	BOOL	
16		Reader_TrainMatchString	Q 3.1	BOOL	
17		Reader_TrainFocus	Q 3.2	BOOL	
18		Reader_TrainBrightness	Q 3.3	BOOL	
19		Reader_UnTrain	Q 3.4	BOOL	
20		Reader_ExecuteDmcc	Q 3.6	BOOL	
21		Reader_SetMatchString	Q 3.7	BOOL	
22		Reader_TrainCodeAck	I 1.0	BOOL	
23		Reader_TrainMatchStrAck	I 1.1	BOOL	
24		Reader_TrainFocusAck	I 1.2	BOOL	
25		Reader_TrainBrightAck	I 1.3	BOOL	
26		Reader_UnTrainAck	I 1.4	BOOL	
27		Reader_ExecuteDmccAck	I 1.6	BOOL	
28		Reader_SetMatchStringAck	I 1.7	BOOL	
29		Reader_UserDataOption	QW 256	WORD	
30		Reader_UserDataLength	QW 258	WORD	
31		Reader_ResultID	MW 256	WORD	
32		Reader_ResultCode	MW 258	WORD	
33		Reader_ResultExtended	MW 260	WORD	
34		Reader_Resultlength	MW 262	WORD	

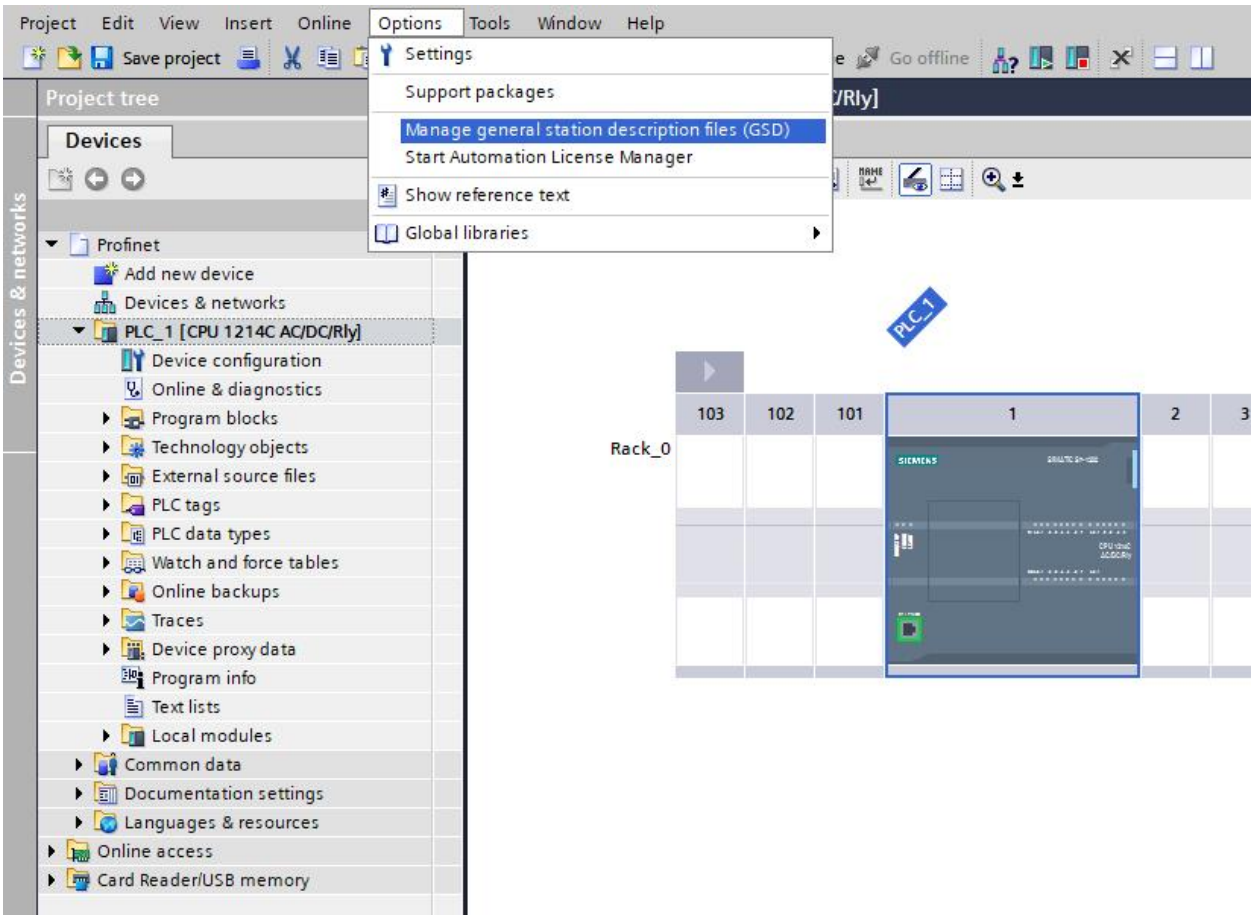
TIA Portal Examples

This section gives some examples of using the SLX device with a Siemens S7-1200 PLC, assuming that you are familiar with the S7-1200 and TIA Portal.

Adding a Device to a TIA Portal Project

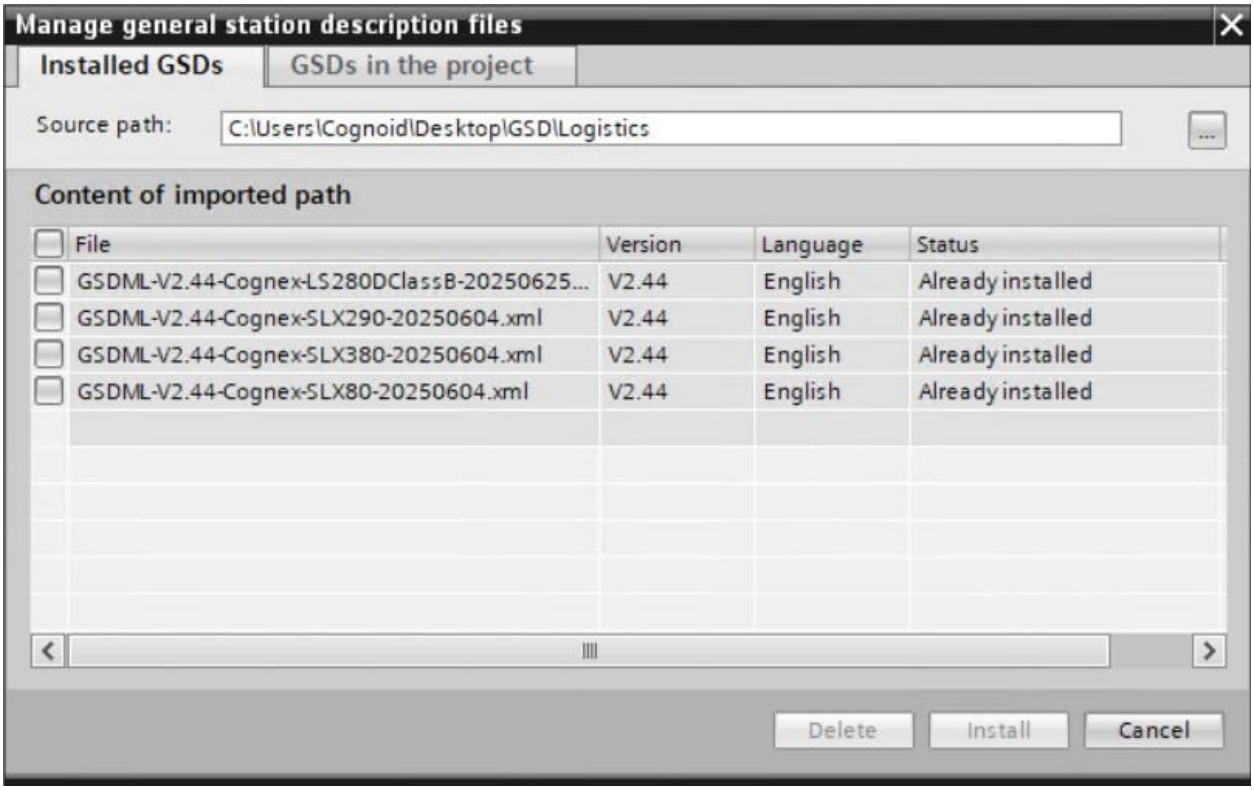
To add a device:

1. In TIA Portal, either create a new project, or open an existing one.
2. In **Project View**, click **Options** and click **Manage general station description files (GSD).**

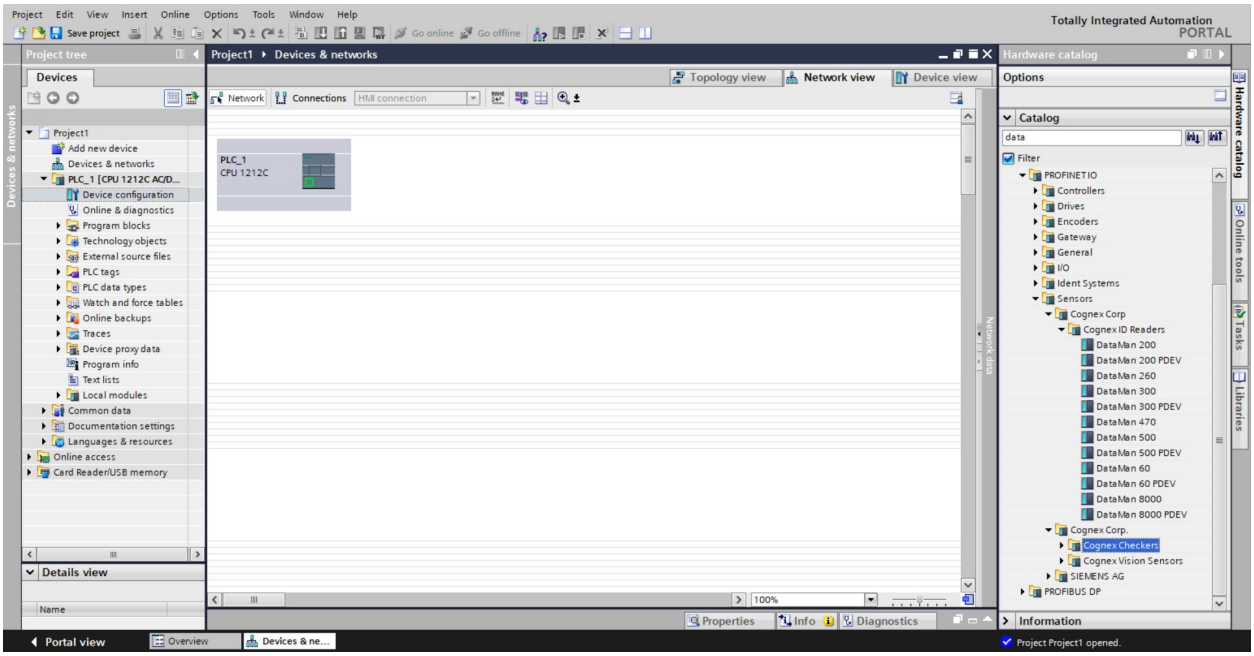


3. Browse to the installation folder of the GSD file (or the location where you saved the GSD file if you downloaded it from the web).
4. Select the GSD file you wish to install and follow the displayed instructions to complete the installation.

Note: If there is more than one GSD file in the list and you are unsure which to install, choose the one with the most recent date.



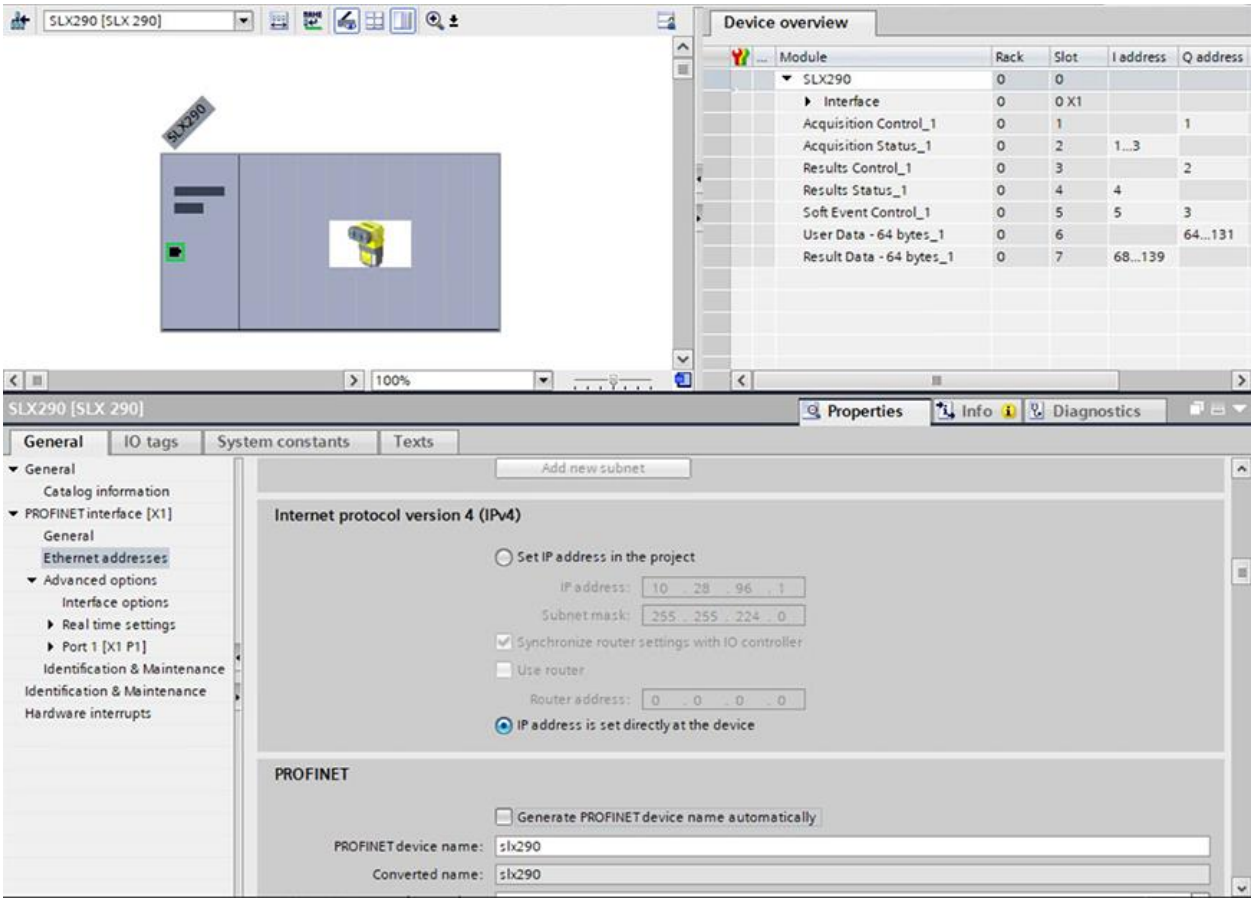
5. Go to **Device Configuration** and add your device in the **Network View** tab.
6. The **Hardware catalog** is displayed on the right. Expand the “PROFINET IO” tree to the “Cognex ID Readers” node.
7. Press and hold the left mouse button to drag the device and drop it on the PROFINET IO network symbol in the left pane.



The HW Config tool automatically maps the device I/O modules into the memory space.

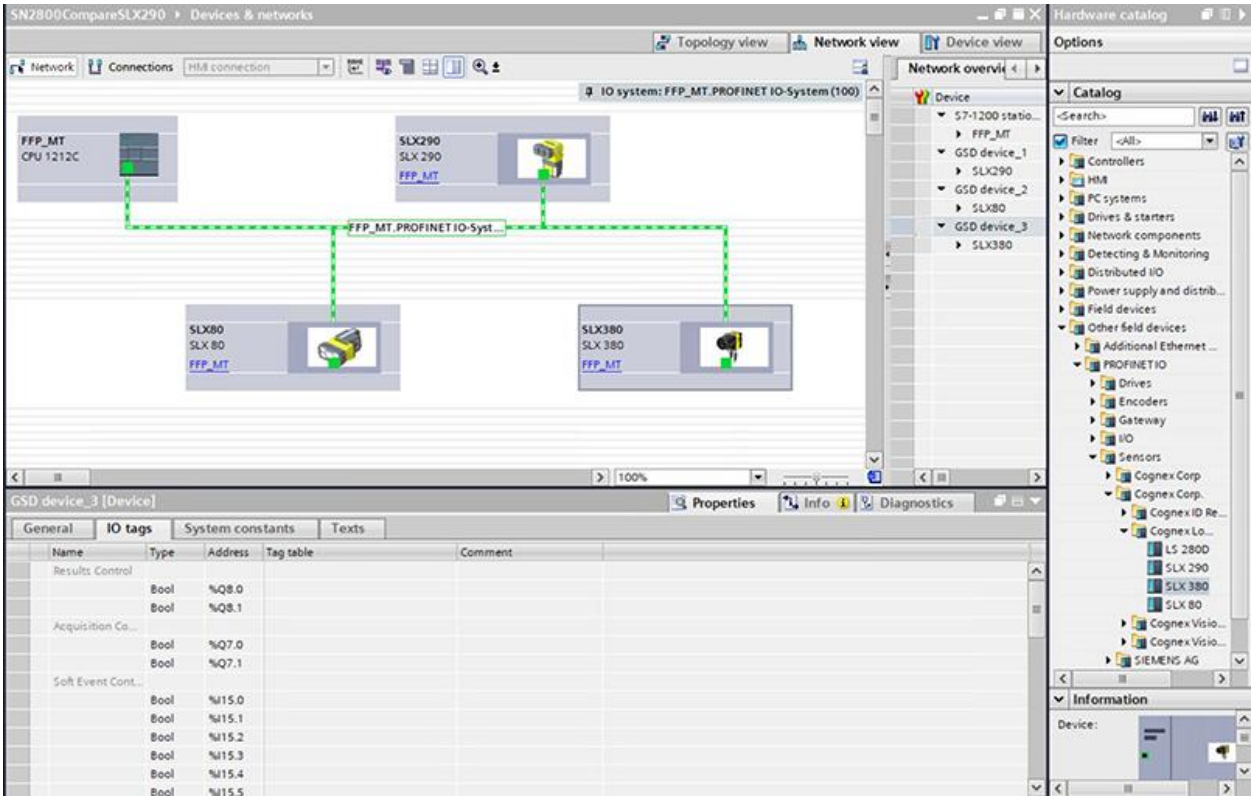
Note: By default, the 64 byte User Data and 64 byte Result Data Modules are inserted. There are multiple sizes available for both of these modules. To optimize performance, use the module size that most closely matches the actual data requirements of your application. You can change the module by deleting the one in the table and inserting the appropriate sized module from the catalog.

8. Right-click the device icon and select **Properties…**.



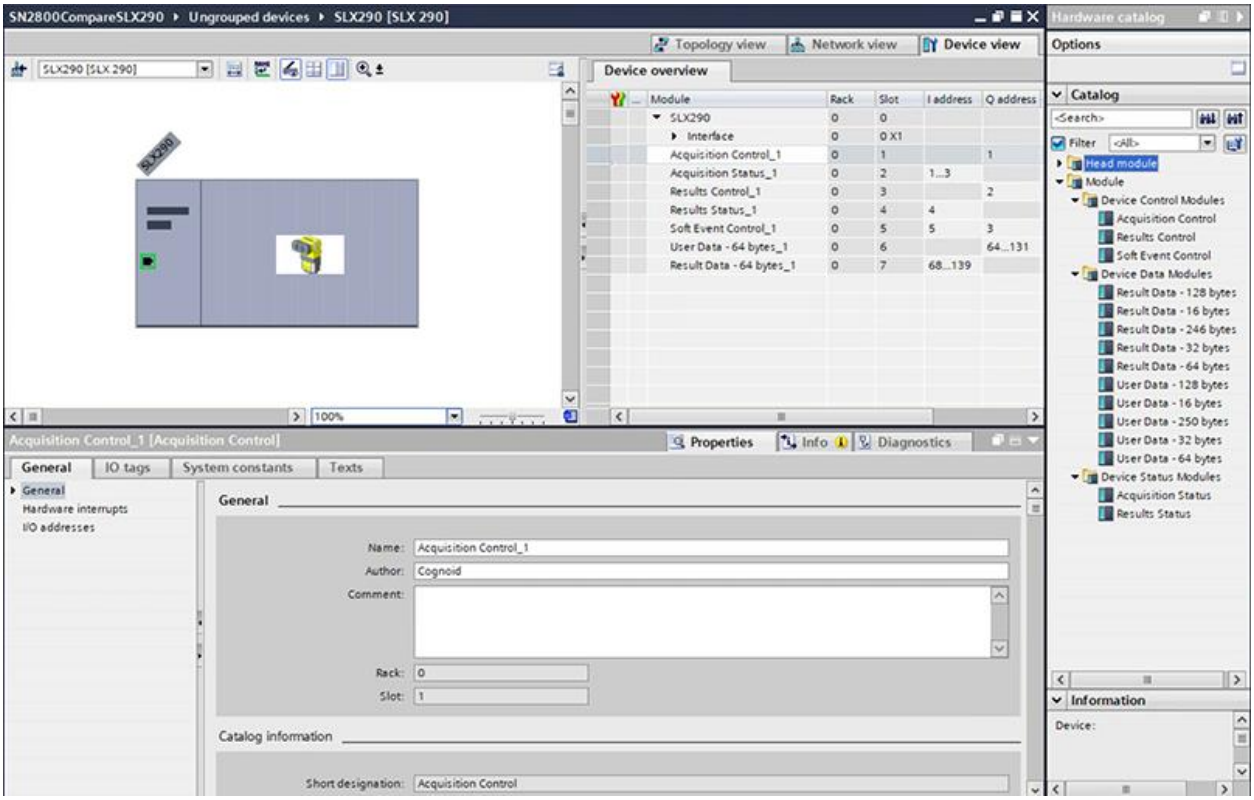
9. Give the device a name. This must match the name of your actual device. The name must be unique and follow DNS naming conventions. For details, see the TIA Portal help documentation.
10. If your device is configured to use its own static IP, select the **IP address is set directly at the device** radio button. As an alternative, if you wish the PLC to assign an IP address, select the Ethernet button and configure the appropriate address.

11. Check the checkbox next to the device image and connect the selected device with the PLC by drawing a line with your mouse.



12. By default, the SIMATIC software maps the User Data and Result Data Modules to offset 256. This is outside of the default process image area size of 128. That is, data in these modules are inaccessible by some SFCs such as BLKMOV. As a solution, either remap the modules to lower offsets within the process image area or expand the process image area to include these modules.

If you choose to expand the process image area, make the size large enough for the module size plus the default 256 offset.



Note: Expanding the process image can have a performance impact on the PLC scan cycle time. If your scan time is critical, use the minimal acceptable module sizes and manually remap them down lower in the process image.

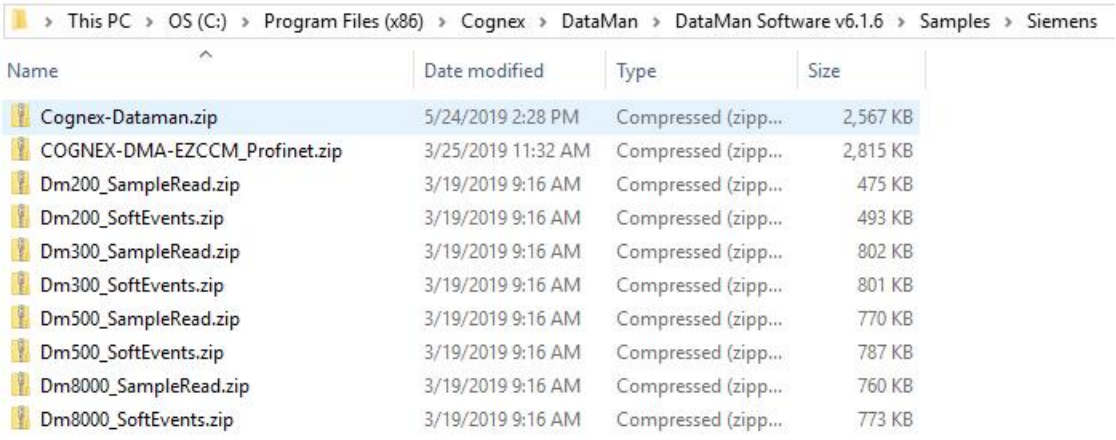
Triggering and Getting Results in TIA Portal

Open the sample program “Cognex-Dataman” for the complete example program.

Note: This sample can be used with any PROFINET-enabled SLX device.

Perform the following steps to install the program:

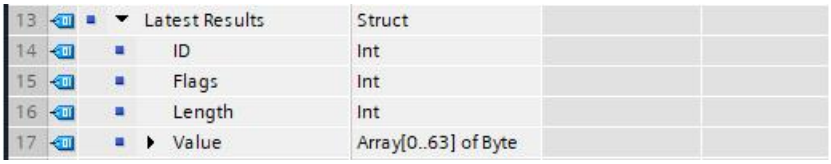
1. Start TIA Portal.
2. Browse to find the sample file on your PC.



3. Save the project on your PC. TIA Portal extracts the sample archive and makes it available.

Reduced to the basics, the process of reading and retrieving results consists of the following:

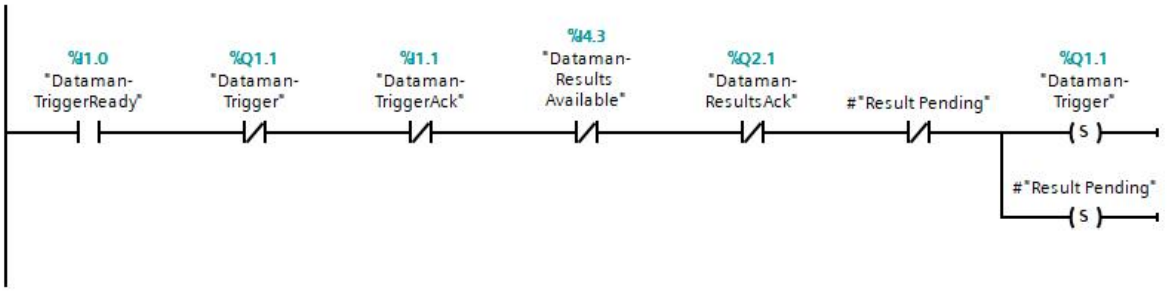
1. Define an area in your application to save read results. There are many options regarding how and where result data can be stored. In this example, a temporary tag is defined containing the fields of the Result Data module that are relevant for our application.



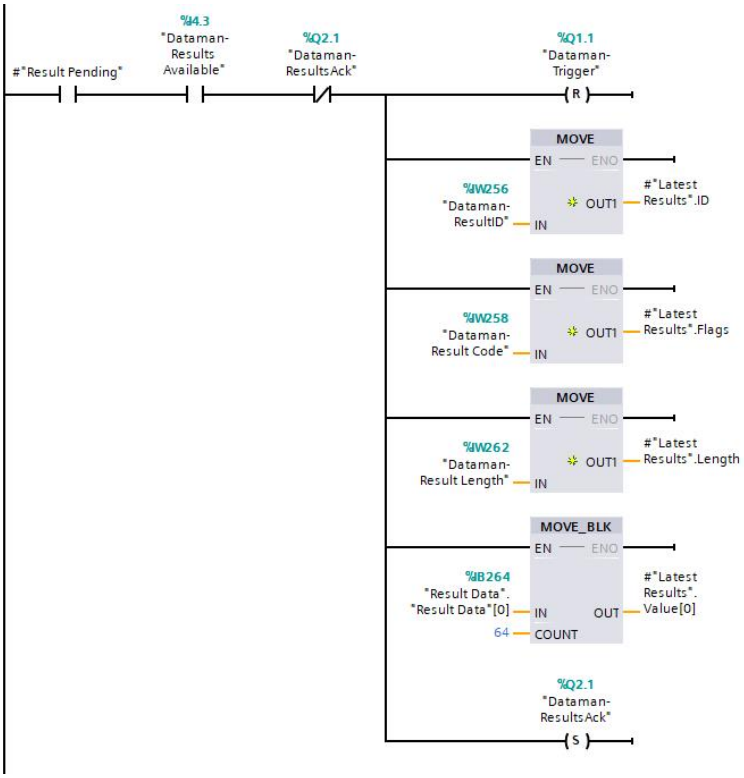
13	Latest Results	Struct	
14	ID	Int	
15	Flags	Int	
16	Length	Int	
17	Value	Array[0..63] of Byte	

2. Enable the device.

3. Set the trigger signal and set coil to indicate a read is pending.

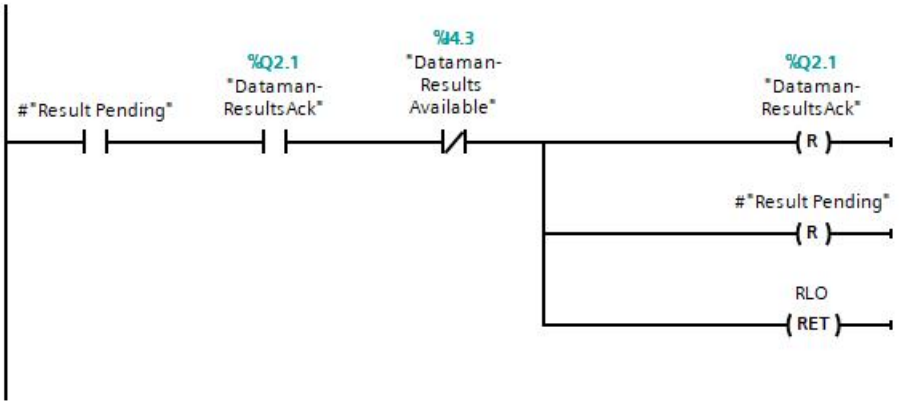


4. As soon as the results are available, clear the trigger signal and save a copy of the result data and set the results acknowledge signal.



5. When the device sees the result acknowledge signal, clear result acknowledge, clear the read pending coil, and signal that the read process is complete.

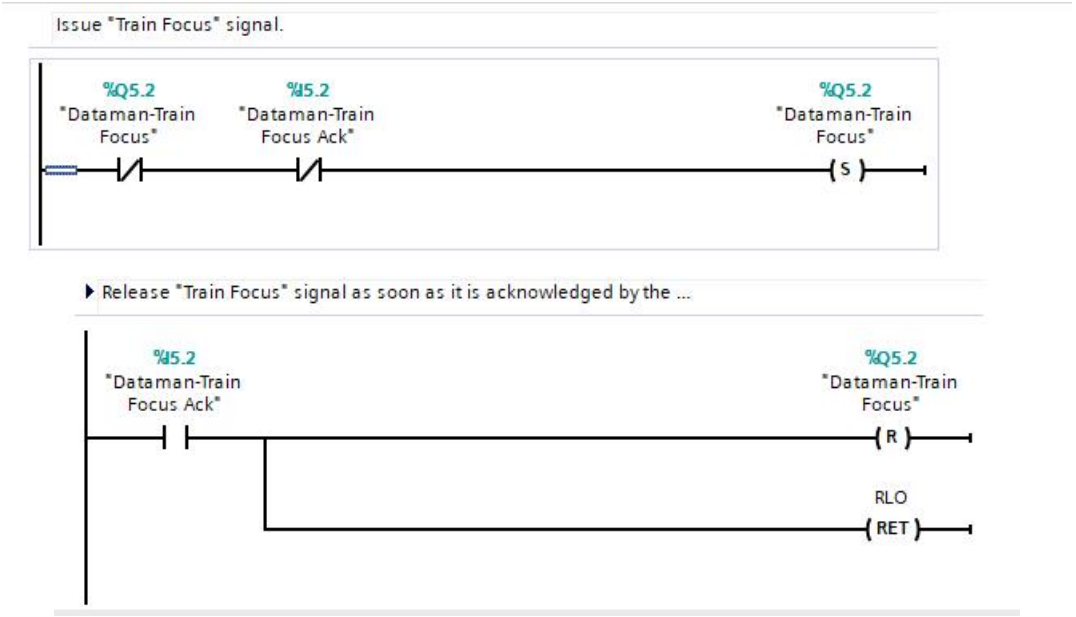
The device clears “Results Available” as soon as it sees the PLC’s “Results Ack” signal.



Using SoftEvents in TIA Portal

SoftEvents are a means of invoking an activity by manipulating a single control bit. The activity for each bit is predefined. For more details, see section SoftEvents. With the exception of “Execute DMCC” and “Set Match String” all SoftEvents may be invoked in the same way. “Execute DMCC” and “Set Match String” require the added step of loading the User Data module with application data before invoking the event.

Reduced to the basics, the process of invoking a SoftEvent consists of the following:



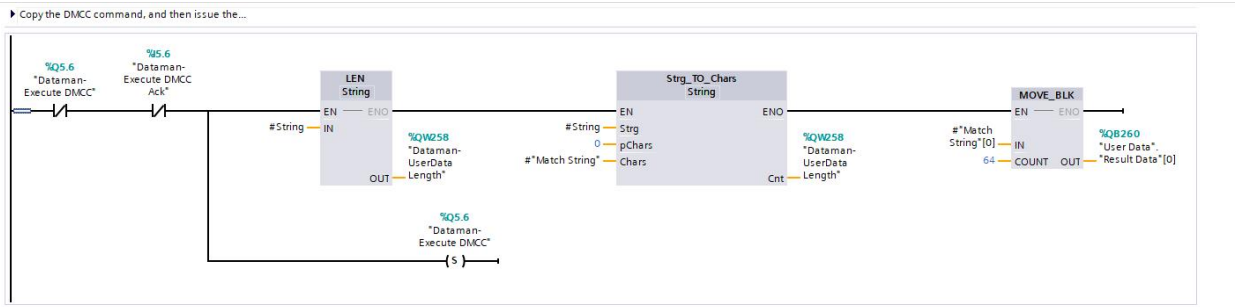
Executing DMCC Commands in TIA Portal

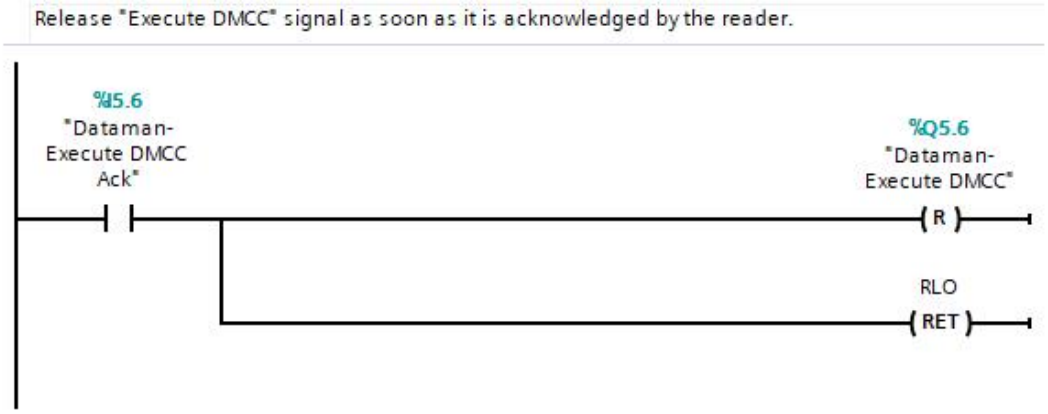
Refer to sample program “Cognex-Dataman” for the complete example program. For information on how to install it, see section Using SoftEvents.

Note: This sample can be used with any PROFINET enabled SLX device.

“Execute DMCC” is a SoftEvent which requires the added step of loading the User Data module with the desired DMCC command string before invoking the event. Note that the SoftEvent mechanism does not provide a means of returning DMCC response data (other than a failure indication). So this mechanism cannot be used for DMCC “||>GET…” commands.

The process of executing a DMCC command is the same for all other SoftEvents (see example above) except the step of invoking the SoftEvent also includes copying the command string to the User Data Module. In this example the command string is exists in a Data Block. This example can be expanded to utilize a Data Block with an array of command strings that the copy function can reference by an index value. This allows the user to pre-define all DMCC commands that are required by the application and invoke them by index.





TIA Portal Example Symbol Table

It is recommended that you define symbols for the device I/O module elements to make the code much easier to read and reduce mistakes. This sample table shows symbols defined for a typical instance of a SLX device. It is possible that device I/O modules are at different addresses in your project. Make sure to adjust your symbol definitions based on the specific offsets of the I/O modules.

For more information, see the *TIA Portal Integration Guide and Tag Generator* on the Cognex Suport Site.

Start ▸ PLC_1 [CPU 1212C AC/DC/Rly] ▸ PLC tags ▸ Tag table_1 [30]

Tag table_1

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Dataman-TriggerEnable	Bool	%Q2.0		☑	☑	☑	
2	Dataman-Trigger	Bool	%Q2.1		☑	☑	☑	
3	Dataman-TriggerReady	Bool	%I2.0		☑	☑	☑	
4	Dataman-TriggerAck	Bool	%I2.1		☑	☑	☑	
5	Dataman-Acquiring	Bool	%I2.2		☑	☑	☑	
6	Dataman-MissedAcq	Bool	%I2.3		☑	☑	☑	
7	Dataman-TriggerID	Word	%IW3		☑	☑	☑	
8	Dataman-BufferResultsEnable	Bool	%Q3.0		☑	☑	☑	
9	Dataman-ResultsAck	Bool	%Q3.1		☑	☑	☑	
10	Dataman-Decoding	Bool	%I5.0		☑	☑	☑	
11	Dataman-Decode Complete	Bool	%I5.1		☑	☑	☑	
12	Dataman-Results Buffer Overrun	Bool	%I5.2		☑	☑	☑	
13	Dataman-Results Available	Bool	%I5.3		☑	☑	☑	
14	Dataman-General Fault	Bool	%I5.7		☑	☑	☑	
15	Dataman-Train Code	Bool	%Q6.0		☑	☑	☑	
16	Dataman-Train Match String	Bool	%Q6.1		☑	☑	☑	
17	Dataman-Train Focus	Bool	%Q6.2		☑	☑	☑	
18	Dataman-Train Brightness	Bool	%Q6.3		☑	☑	☑	
19	Dataman-Untrain	Bool	%Q6.4		☑	☑	☑	
20	Dataman-Execute DMCC	Bool	%Q6.6		☑	☑	☑	
21	Dataman-Set Match String	Bool	%Q6.7		☑	☑	☑	
22	Dataman-Train Code Ack	Bool	%I4.0		☑	☑	☑	
23	Dataman-Train Match String Ack	Bool	%I4.1		☑	☑	☑	
24	Dataman-Train Focus Ack	Bool	%I4.2		☑	☑	☑	
25	Dataman-Train Brightness Ack	Bool	%I4.3		☑	☑	☑	
26	Dataman-Untrain Ack	Bool	%I4.4		☑	☑	☑	
27	Dataman-Execute DMCC Ack	Bool	%I4.6		☑	☑	☑	
28	Dataman-Set Match String Ack	Bool	%I4.7		☑	☑	☑	
29	Dataman-UserData Option	Word	%QW256		☑	☑	☑	
30	Dataman-UserData Length	Word	%QW258		☑	☑	☑	

Open Network Ports and Enabled Ports and Services

The following table summarizes open network ports and services:

Protocol	Port Number	Program Name	Notes
TCP	20	vsftpd	Temporarily open for active FTP connections
TCP	21	vsftpd	FTP
TCP	23		Telnet
TCP	53		dnsmasq
TCP	80	nginx	HTTP
TCP	2222		Ethernet/IP
TCP	20023		Device Recover
TCP	44818		Ethernet/IP
TCP	51000-51050	vsftpd	Temporarily open for passive FTP connections
UDP	53		dnsmasq
UDP	161/162		PROFINET - SNMP
UDP	2222		Ethernet/IP Implicit
UDP	1069		Device Discovery
UDP	34964		PROFINET
UDP	44818		Ethernet/IP
UDP	51069		Device Discovery

DMCC Communication

SLX devices support the following DMCCs:

DMCC	Description	Action
CAMERA.EXPOSURE	Camera parameters where exposure is in microseconds.	SET/GET
CAMERA.GAIN	Gets or sets the sensitivity of the image sensor. A higher number means higher sensitivity, which also increases image noise. The gain has no effect if the camera runs with automatic exposure.	SET/GET
SYMBOL.DATAMATRIX	Enable DataMatrix Symbology.	SET/GET
SYMBOL.C128	Enable Code 128 Symbology.	SET/GET
TRIGGER	Software trigger.	ON/OFF