SPECTRUM CONTROLS
An Allient Company

Universal Industrial Gateway Application Note
Using a Gateway to Replace a 1761-NET-ENI Module Application Note

Document No: 0100340-01 Rev. A0

Heading

The 1761-NET-ENI module is commonly used to bridge a communications path between a DF-1 Full Duplex device and Ethernet/IP (TCP). However, it is no longer available from Rockwell and there is no direct replacement. With many applications still using DF-1, there remains a need to provide a means of bridging it to EtherNet/IP.

One option is to use the Spectrum Controls Universal Industrial Gateway (Gateway). One advantage of using the Gateway is that in the application where it will be used, it offers increased flexibility and future expandability rather than simply replace the NET-ENI without any enhancements. However, the Gateway is not a “drop in” replacement. It does not provide a link between RSLinx and the DF-1 PLC and it does require that the PLC application be modified by way of minor changes in the programming. This document provides the simple steps needed to replace the NET-ENI with the Gateway.

Please note that it is assumed that the NET-ENI being replaced is from an application that either does or did work. When using the NET-ENI one or both of the PLC’s in the network is doing the messaging. When replacing the NET-ENI with a Gateway, the messaging is handled by the Gateway. As a result, the programming within the PLC needs to be altered to remove or disable the messaging within the PLC. This is not a requirement of the Gateway but because the NET-ENI is no longer in the system and the path that the message instructions used is no longer available and thus they will not work.

When setting up a Gateway to replace a NET-ENI, there are three primary components that need to be addressed:

- In the Gateway setup there needs to be a device configured that represents the link between the Gateway and the PLC.
- For each device there needs to be at least one Tag configured that is linked to a Tag or Data File in the PLC.
- A Tag Map needs to be created to execute the movement of data between the two PLCs.

The information needed to configure a Device in the Gateway is:

- The type of PLC.
- The communications type and protocol.
- An IP address for the PLC connection using Ethernet.
- The port number the DF-1 PLC is connected to.

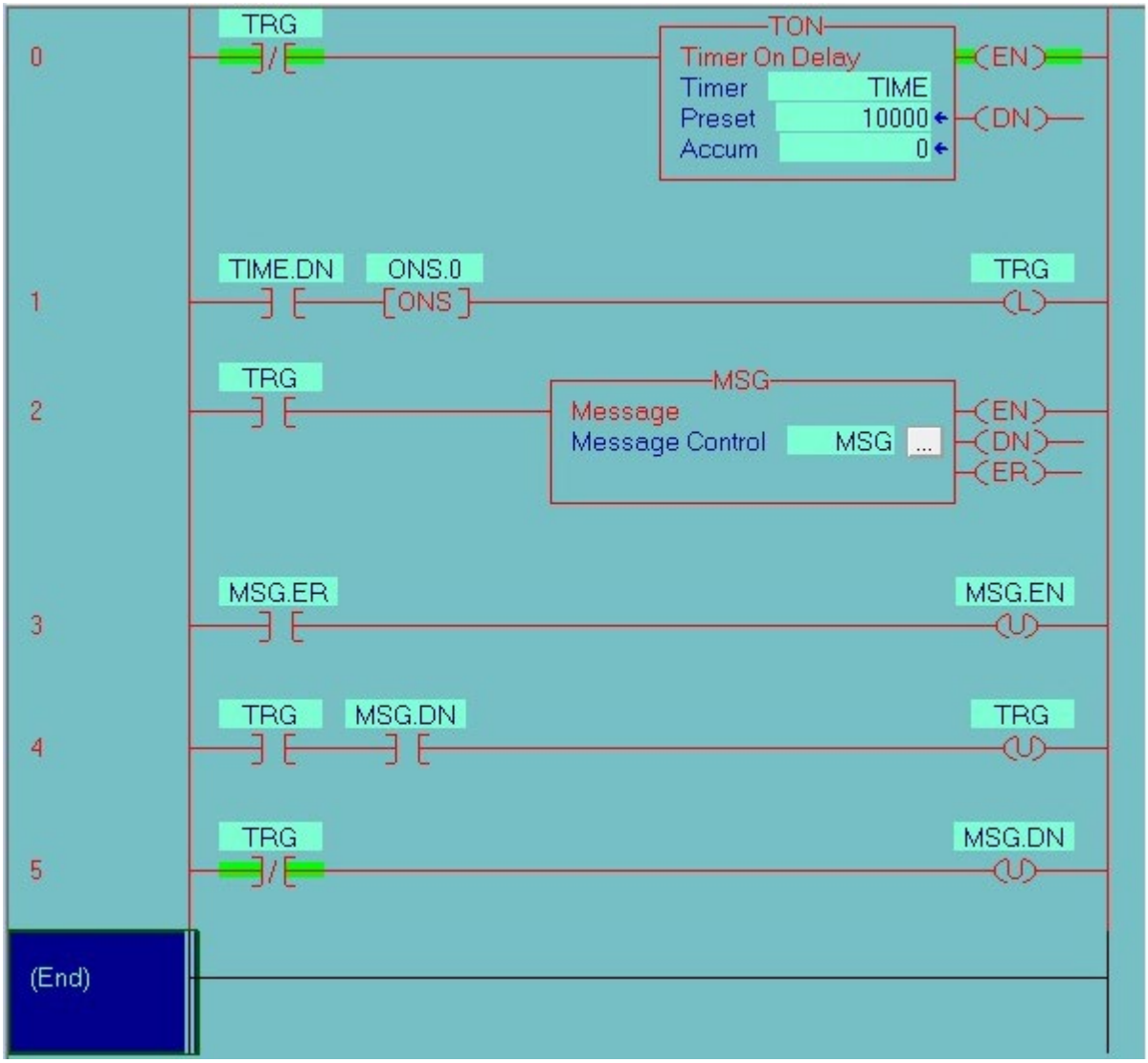
In order to correctly configure the Tags and Tag Maps, you need to know how the message instruction in the PLC is configured and how it’s triggered. How the message instruction is configured will give you the information on where the data is being taken from and where it is being moved (e.g. the name of a tag in a Logix controller and the address of a Data File in a SLC or MicroLogix). How the message instruction is triggered will tell you if you need to trigger the Tag Map using the change of state of data within one of the PLCs or via time.

In the following examples we will look at two types of messaging done in the Logix PLC and how the Gateway would be configured to take over communications. Please note that the PLC configuration examples are two of many possible ways to configure PLC messaging and are not intended to represent the best methods available. Both examples represent a CompactLogix writing the value of a tag (N9) to a data file (N10:0) in a SLC5/03 with a NET-ENI bridging the Ethernet/IP protocol used by the CompactLogix to the SLC5/03’s DF-1 port. Because the CompactLogix is the PLC doing the messaging it is the only one that will need minor programming changes in its ladder logic routine.

Spectrum Controls, Inc.
1705 132nd Ave NE
Bellevue, WA 98005 USA

Telephone: (425) 746-9481
Fax: (425) 641-9473
Web Site: www.spectrumcontrols.com

In the first example the messaging in the PLC is triggered once every 10 seconds using a timer:

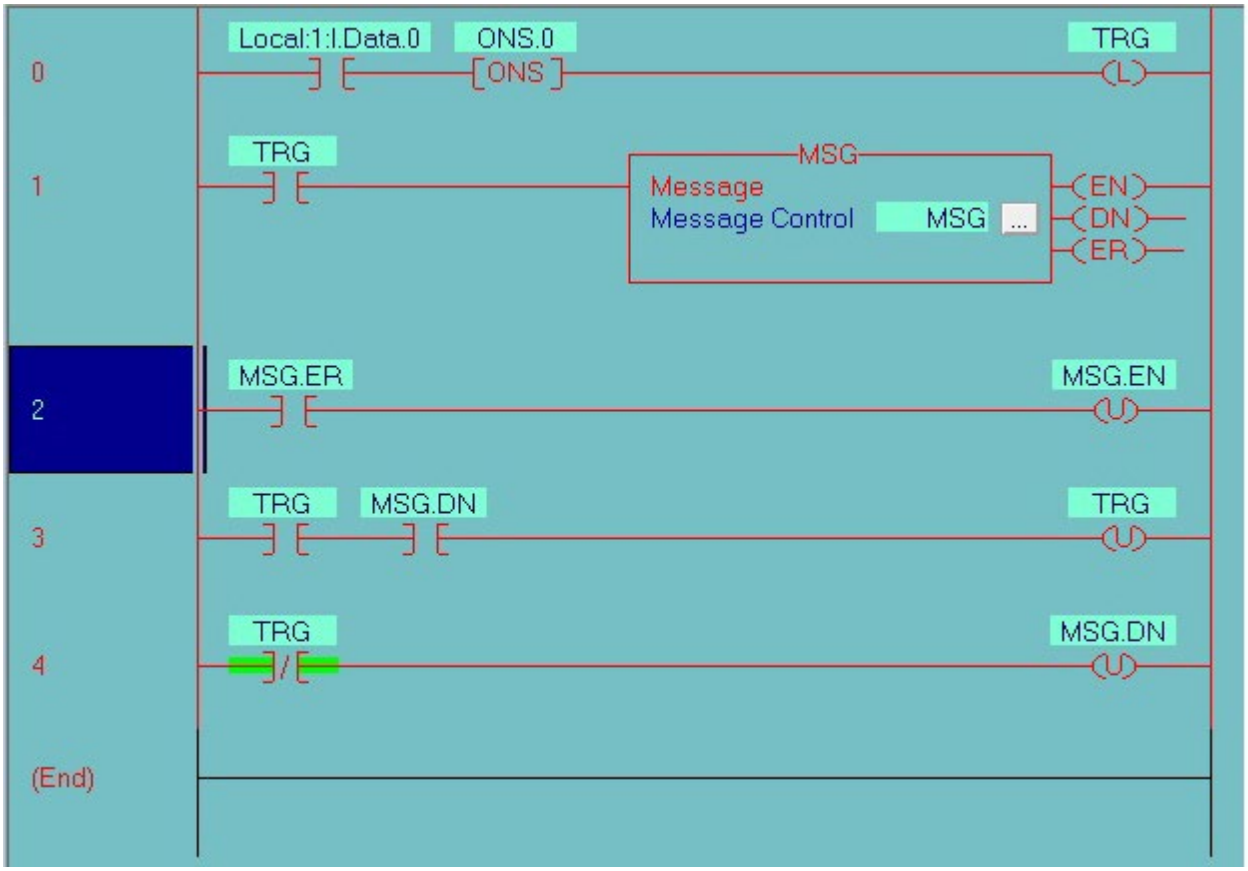


When the bit “TRG” is low the timer is enabled and runs for 10 seconds. When the timer is done the bit “TIME.DN” is set which latches the bit “TRG” enabling the message instruction. If the message instruction errors, it is reset and runs again. When the message instruction is successful it sets the bit “MSG.DN” which unlatches the “TRG” bit which in turn unlatches the “MSG.DN” bit and the process starts over again.

Spectrum Controls, Inc.
1705 132nd Ave NE
Bellevue, WA 98005 USA

Telephone: (425) 746-9481
Fax: (425) 641-9473
Web Site: www.spectrumcontrols.com

In the example that uses an event to trigger the message instruction, the basic process is the same except that a discrete input is used to set the “TRG” bit:



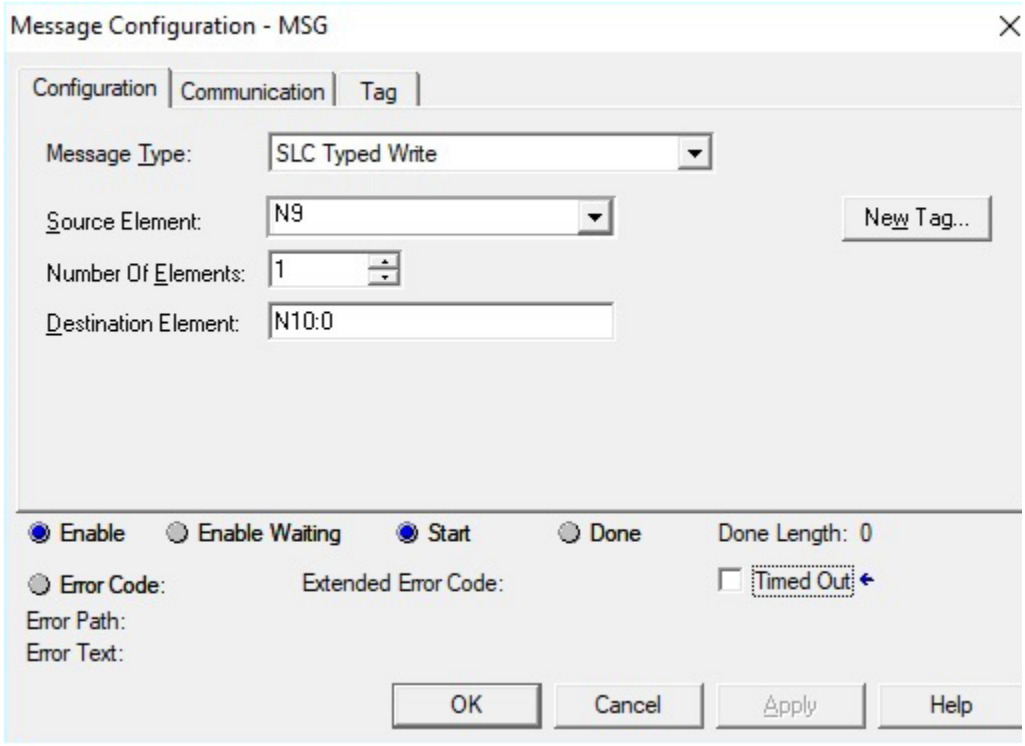
As with the time-based routine when the message instruction is successful the system is reset but will not execute the messaging until the discrete input opens then closes again. In both examples, in order to set up the Gateway to replace the NET-ENI, you need to know where the data is being read from and written to.

One way to determine this is to look at the message instructions configuration (which is the same in both examples).

Spectrum Controls, Inc.
1705 132nd Ave NE
Bellevue, WA 98005 USA

Telephone: (425) 746-9481
Fax: (425) 641-9473
Web Site: www.spectrumcontrols.com

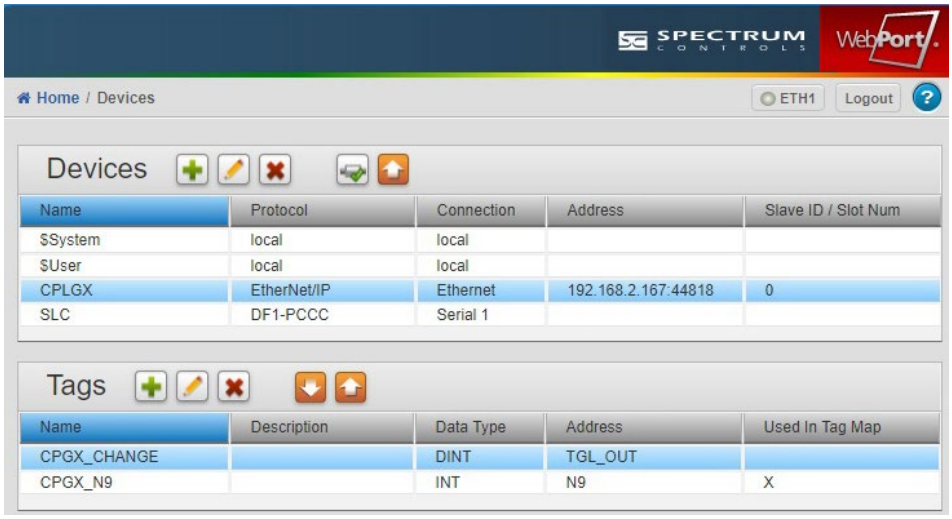
Because the Message Type is a SLC Typed Write, data from the Tag in the Logix PLC, “N9” or the “Source Element”, is being moved from the Logix PLC to the Data File “N10:0” or the “Destination Element” in the SLC/MicroLogix:



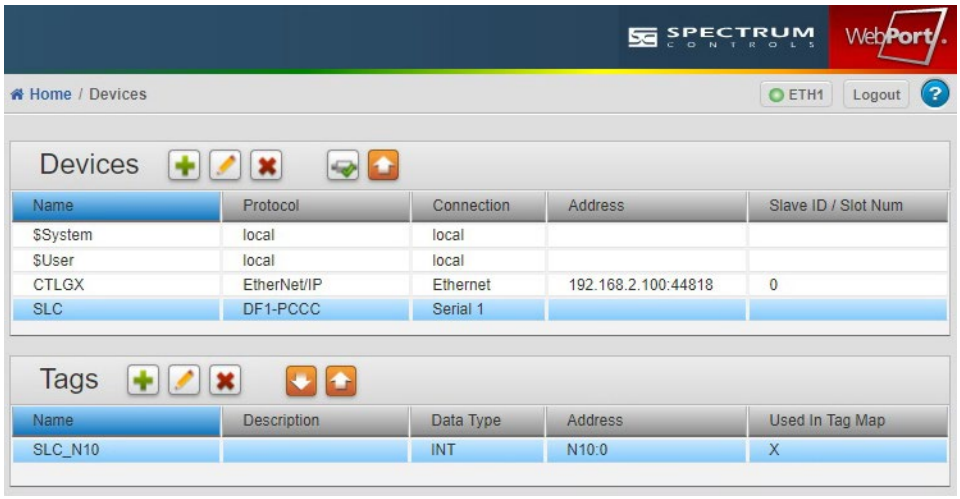
With the information from the two examples we now can configure the Gateway to take over the messaging between the two PLCs. The first step is to create two devices in the Gateway with one representing the CompactLogix and the other the SLC5/03. The SLC5/03 device will have a Tag that will link to the location of the Data File and the CompactLogix device will have a Tag linked to the tag “N9”. It will also have a Tag labeled “TLG_OUT” which will only be used in the application that uses a discrete input to trigger messaging. The only remaining item after that will be the Tag Map which will be slightly different between the two examples.

For the CompactLogix a Device is created that defines the communications protocol and the IP address of the PLC.

As mentioned above, this Device has two Tags. One is labeled “CPGX_CHANGE” which is linked to the PLC tag “TGL_OUT” and the other is labeled “CPGX_N9” which is linked to the PLC tag “N9”.



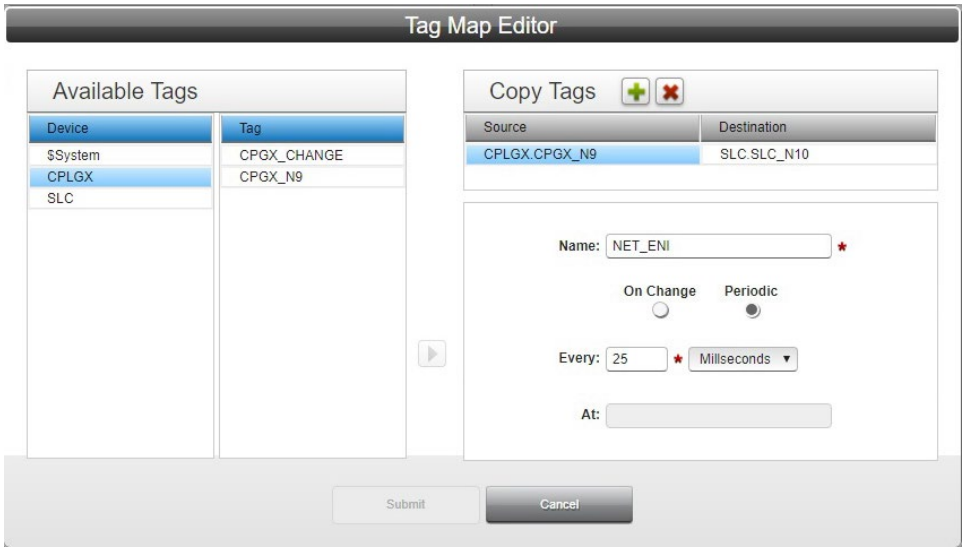
For the SLC5/03 a Device is created that defines which port[1] it is connected to. The Tag for the SLC5/03 is labeled “SLC_N10” and is linked to the Data File “N10:0”.



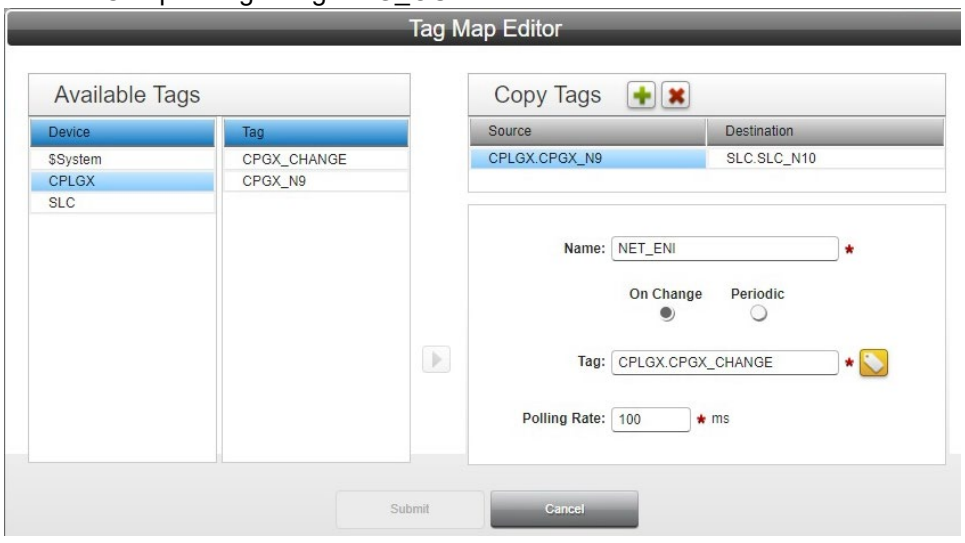
Now that we have created the Devices with associated Tags, we can create the Tag Map. Each Tag Map can be set to trigger using the “Periodic” setting, which is a time basis ranging from once every 1ms to once every day or using “On Change” which monitors the state of a Tag and triggers the Tag Map when the value of that Tag changes.

[1] The serial ports are configured using a separate configuration page which allows you to set the Baud rate and word protocol

For the example using a trigger setting of “Periodic” you can see that the source Tag is the one associated with the CompactLogix Tag “N9”, the destination is the Tag associated with the Data File “N10:0” and the Periodic trigger is set to once every 25ms.



For the event trigger example, the source and destination are the same as with the time based example however the trigger is set to “On Change” and the Tag it is monitoring (once every 100ms) is the one associated with the CompactLogix Tag “TLG_OUT”.



Spectrum Controls, Inc.
1705 132nd Ave NE
Bellevue, WA 98005 USA

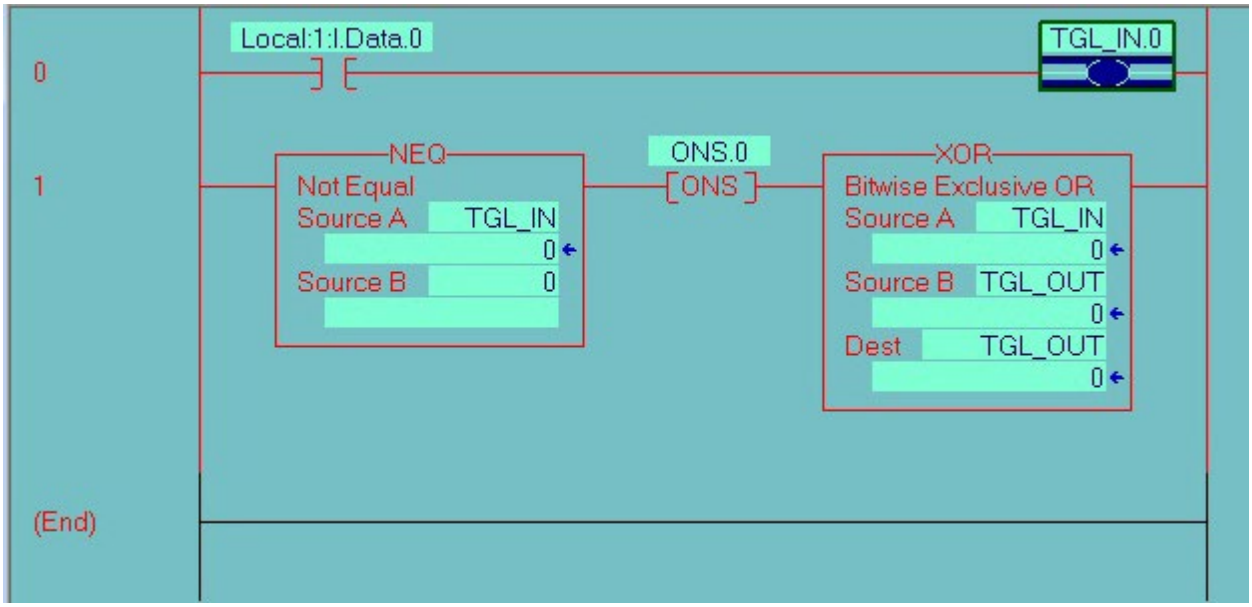
Telephone: (425) 746-9481
Fax: (425) 641-9473
Web Site: www.spectrumcontrols.com

With the Devices, Tags and Tag Maps configured we now need to change the configuration in the PLC[2].

For the application using a timer to trigger the message instruction, those lines of ladder logic need to be removed or disabled. Disabling is recommended in case the messaging needs to be re-enabled for any reason.

For applications using the On Change trigger, the messaging needs to be removed or disabled and the event that triggered the message instruction needs to be tied to the Tag within the PLC that the Gateway will use to trigger the Tag Map.

For this example, the discrete input used to trigger the message instruction is now used to toggle the bit "TGL_IN.0". When the discrete input goes from open to closed, the comparator "NEQ" (Not Equal) combined with the "XOR" function toggles the bit "TGL_OUT" from 0 to 1 and then from 1 to 0 when the switch again goes from open to closed. This method is preferred over simply tying the discrete input directly to the Tag being monitored because the On Change trigger only looks at a change of state, not when it changes from 0 to 1. If the discrete were used by itself then the Tag Map would trigger when the discrete in goes from open to closed as well as when it goes from closed to open effectively triggering it twice with only one execution of the discrete input. Please note that this is only one of many methods of generating an event-based trigger. Any method that results in a monitored Tag changing its value will work as far as the Gateway is concerned.



These two examples show how the Universal Industrial Gateway can be utilized in applications that previously had a 1761-NET-ENI module. It demonstrates bridging a PLC communicating via EtherNet/IP to one using DF-1 as well as applications that need to bring a DF-1 PLC into an EtherNet/IP environment. However, it is also worth mentioning that adding the Gateway to these types of applications enhances them by allowing future upgrades and expansion with minimal cost and effort. Because the Gateway simultaneously supports multiple protocols if the DF-1 PLC is replaced with one that does have EtherNet/IP capability, you can still rely on the Gateway to handle the communications and in most applications that is simpler and less time consuming than reprogramming a PLC to incorporate and use message instructions. It also allows you to add devices that normally would not be simple and easy to add such as a Modbus RTU product.

[2] It is highly recommended that before making any changes to the PLC program, that the program be saved as a backup in case there is a need to restore the PLC's program back to how it was configured before the change was made.