



NOTE Some protocols on the Digi One IAP product family that are described in this document are being moved to legacy status because the protocols described are no longer actively supported. The products themselves continue to be active products.



Connecting Rockwell ControlLogix to Modbus slaves by Digi One IAP

March 2020

90000643

Contents

1	Introduction	5
1.1	Sample application.....	5
1.2	Powerful Digi One IAP features.....	5
1.3	Reading Modbus 0x coils	5
1.4	Writing Modbus 0x coils	6
1.5	Writing single bits (0x coils).....	6
1.6	Reading Modbus 1x and 3x inputs	6
1.7	Reading Modbus 4x values	6
1.8	Writing Modbus 4x values	7
2	Setting up the Digi One IAP	7
2.1	Overview.....	7
3	By web wizard.....	7
3.1	Select the Industrial Automation Wizard.....	7
3.2	Select to create a new group and reset all IA settings	8
3.3	Limit this group to any protocol that supports mapping	8
3.4	Define Ethernet/IP as the master or message source	9
3.5	Define timeouts for the incoming Ethernet/IP connection	9
3.6	Master summary—incoming Ethernet/IP is ready.....	10
3.7	Define Modbus/TCP as the master or message source.....	10
3.8	Master summary—both incoming masters all set	11
3.9	Define master priority	11
3.10	Define first message destination—our serial port 1 to PLC(s).....	12
3.11	Set the protocol and port number	12
3.12	First destination complete	13
3.13	Define second message destination—PLC at 192.168.1.xxx.....	13
3.14	Set the Modbus/TCP slave details	13
3.15	Define third message destination—PLC at 192.168.1.yyy	14
3.16	Finished—review the settings	14
3.1	Reboot the Digi One IAP	15

- 4 Setup by telnet..... 15
- 5 Setting up the ControlLogix MSG Block..... 15
 - 5.1 Create your MSG block rung in RSLogix 5000 16
 - 5.2 Configure the MSG for reading 0x, 1x, 3, or 4x areas..... 16
 - 5.3 Configure the MSG for writing 0x or 4x areas..... 17
 - 5.4 Configure the MSG for writing a single bit to 0x area..... 18
- 6 How to test this application..... 19
 - 6.1 Testing by simulator 19
- 7 Troubleshooting and FAQ 19
 - 7.1 Can I use PLC5 or CIF style commands? 19
 - 7.2 Why is my MSG block returning a bad status? 19
- 8 How Rockwell polls turn into Modbus polls.....20
 - 8.1 Given Modbus notation, determine your Rockwell notation.....20
 - 8.2 How to document.....20

1 Introduction

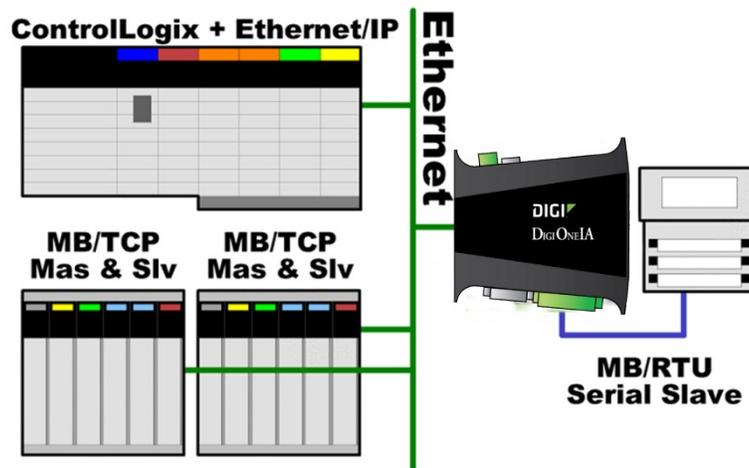
Abstract

This document describes how to use an MSG block within a ControlLogix PLC program to remotely read or write Modbus slaves using the Digi One IAP. The Digi One IAP makes the Modbus device appear as an SLC5 or MicroLogix DF1 device. The ControlLogix uses SLC5 Typed Reads/Writes to access the Modbus slaves.

1.1 Sample application

This application note uses a sample design with three masters and three slaves. The ControlLogix can poll the Modbus/RTU slave via Ethernet/IP.

- The ControlLogix can poll either Modbus/TCP PLC as slave via Ethernet/IP
- The Modbus/TCP PLC can poll the Modbus/RTU slave via Modbus/TCP



1.2 Powerful Digi One IAP features

Other vendors make DF1-to-Modbus/RTU converters. Adding an ENI allows you to poll the Modbus/RTU slave. However, this approach does not help you access the Modbus/TCP slaves. Additionally, this converter pair blocks direct Modbus/TCP access to the Modbus/RTU slave.

The Digi One IAP approach allows the ControlLogix and other Modbus/TCP masters to concurrently share the Modbus/RTU slave(s).

You can add Modbus/TCP cards to your ControlLogix, but these cards are expensive and require a second Ethernet cable, hub/switch port, and IP address. The Digi One IAP approach allows the existing ControlLogix Ethernet port to concurrently handle Rockwell and Modbus data.

1.3 Reading Modbus 0x coils

To read data in the Modbus 0x coil area with function 1, use an SLC Typed Read command to a B-file. This command reads full blocks of 16 x 1-bit data packed as words and word aligned. Misaligned bit blocks, partial words, and masked/scattered bit reads are not supported. Reading B10:0 returns the first

word of data—Modbus bits 0x00001 to 0x00016 packed as a word. Reading B10:1 returns the second word and so on. Files B0 to B9 are treated like B10.

1.4 Writing Modbus 0x coils

To write data in the Modbus 0x coil area with function 15, use an SLC Typed Write command to a B-file. This command writes full blocks of 16 x 1-bit data packed as words and word aligned. Misaligned bit blocks, partial words, and masked/scattered bit writes are not supported.

1.5 Writing single bits (0x coils)

The ControlLogix does not allow using a BOOL or bit data type in the MSG block. In addition, the way Modbus supports single-bit read/write is very different from the way DF1/PCCC requires masking bits within words.

To enable writing single bits, the Digi One IAP supports a special use of a string data type written to a B-file. You can send a string such as `/2=on` or `/11=1` to issue a Modbus function 5 to set or clear a single bit as referenced by the B-file word. Writing `/2=1` to B10:0 sets Modbus bit 0x00003 to 1 or on.

You can find more detail later in this application note.

1.6 Reading Modbus 1x and 3x inputs

The ControlLogix does not allow the MSG block to read or write “I” or input files. To enable access to Modbus input data in areas 1x and 3x, the Digi One IAP assumes that AB file numbers 250 to 255 map to these areas.

Read files B250 to B255 to read the Modbus 1x input status area. Misaligned bit blocks, partial words, and masked/scattered bit reads are not supported. B250:0 reads the first word: Modbus bits 1x00001 to 1x00016 packed as a word.

Read files N250 to N255 to read the Modbus 3x input register area. N250:0 reads the first word 3x00001, and so on.

1.7 Reading Modbus 4x values

The SLC Typed Read to N-file types map to a read of Modbus 4x holding registers. Below are the algorithms for mapping:

- For files N0-N10, use the starting element as the Modbus address. For example, N7:0 and N10:0 map to Modbus 4x00001, while N10:17 maps to Modbus 4x00018. Files N0 to N9 are treated like N10.
- For files N11 to N255, subtract 10 from the file number, multiply this number by 100 and add the starting element. For example, N21:0 is Modbus 4x01101 since $((21 - 10) * 100) + (0 + 1)$ is 1101. Register N170:17 is Modbus 4x16018 since $((170 - 10) * 100) + (17 + 1)$ is 16018. The maximum address is thus N255:255 for 4x24757. This may appear complex, but a simple Excel spreadsheet can clearly document your mappings.

If you need access to all 65,535 Modbus registers, use the **PLC2** or **CIF Unprotected Read**. These reads map directly into Modbus registers. A CIF read of offset 0 maps to Modbus 4x00001, offset 100 maps to Modbus 4x00101, and so on. This off-by-one behavior occurs because Modbus register 4x00001 is actually sent on the wire as address 0, not 1.

1.8 Writing Modbus 4x values

The SLC Typed Write command to an N-file maps to a Modbus function 6 or 16 to Modbus 4x holding registers. Writing one register uses Modbus function 6, while writing more than one register uses function 16.

If you need access to all 65,535 Modbus registers, use the PLC2 or CIF Unprotected Write. These writes map directly into Modbus registers. A CIF write of offset 0 maps to Modbus 4x00001, offset 100 maps to 4x00101, and so on. This off-by-one behavior occurs because Modbus register 4x00001 is actually sent on the wire as address 0, not 1.

2 Setting up the Digi One IAP

The instructions in this document are based on the assumption that you understand the basics of setting up and accessing your Digi product by Ethernet and TCP/IP. If you need help with these procedures, see the [Digi One IAP support page](#).

2.1 Overview

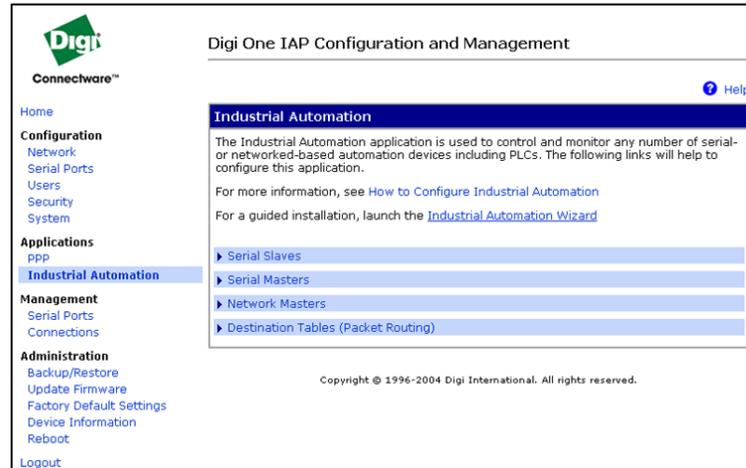
Here is a summary of the tasks for configuring the Digi One IAP:

- Configure to accept incoming Ethernet/IP masters (a message source).
- Configure for Modbus/RTU slaves 0-32 on port 1.
- Configure for Modbus/TCP slave/unit id 33 at remote IP 192.168.1.xxx.
- Configure for Modbus/TCP slave/unit id 34 at remote IP 192.168.1.yyy. By default, any EIP messages with DST=35-255 fail (as set up here).

3 By web wizard

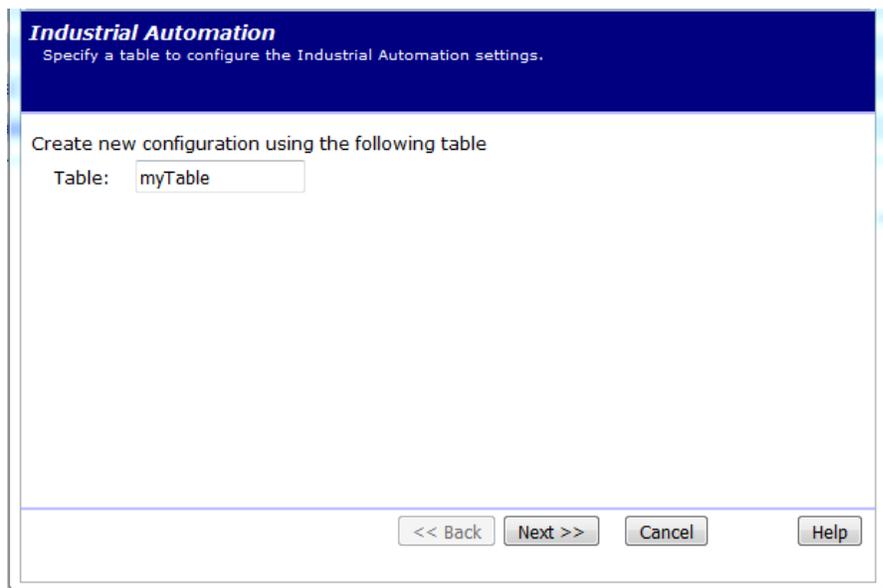
3.1 Select the Industrial Automation Wizard

1. In the left column, under **Applications**, click **Industrial Automation**.
2. Click **Industrial Automation Wizard**.



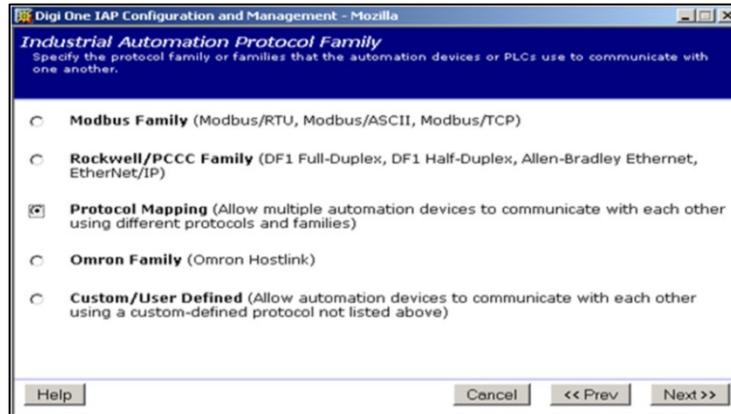
3.2 Select to create a new group and reset all IA settings

1. Create a new configuration group.
2. Clear all existing IA settings.
3. Click **Next**.



3.3 Limit this group to any protocol that supports mapping

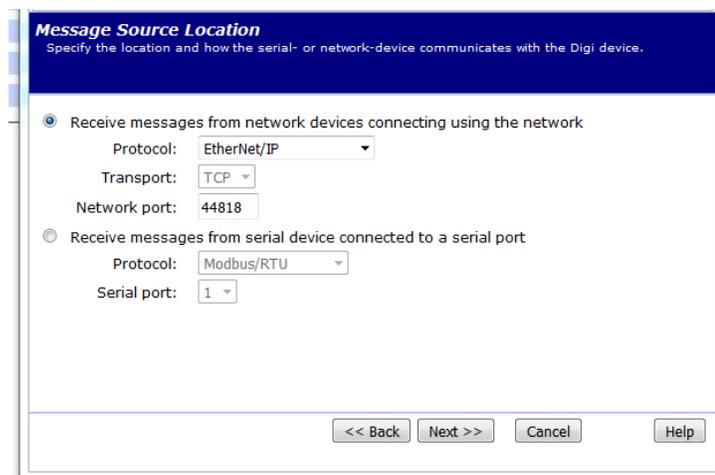
1. Click the **Protocol Mapping** radio button to map Rockwell protocols to Modbus. This setting allows you to select most protocols using the wizard.
2. Click **Next**.



3.4 Define Ethernet/IP as the master or message source

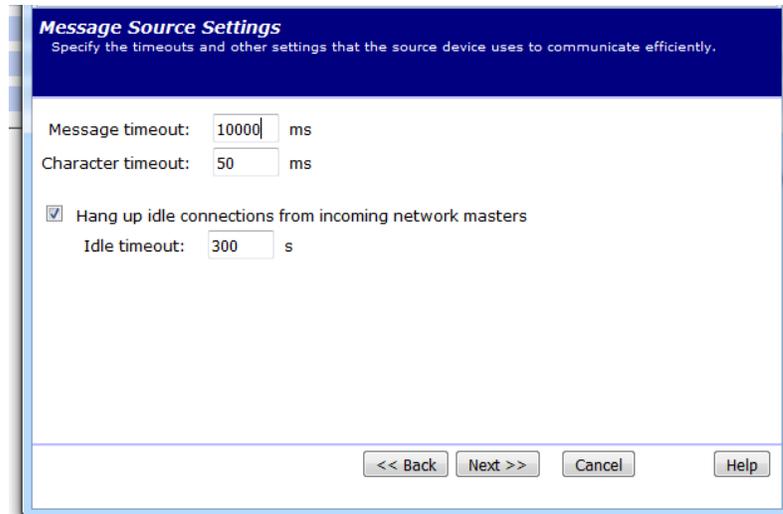
You see some windows about message sources that are not covered in this document.

1. Click **Next** until you see this window.
2. Click the **Receive messages from network devices connecting using the network** radio button.
3. Click the **Protocol input** dropdown list and select **Ethernet/IP**. The network port automatically sets to 44818.
4. Click **Next**.



3.5 Define timeouts for the incoming Ethernet/IP connection

1. In the **Message timeout** input textbox, enter **10000** milliseconds.
2. This setting makes manual testing with HyperTerminal easier. ControllLogix usually waits up to 30 seconds for a response. The other two settings are fine as is.
3. Click **Next**.



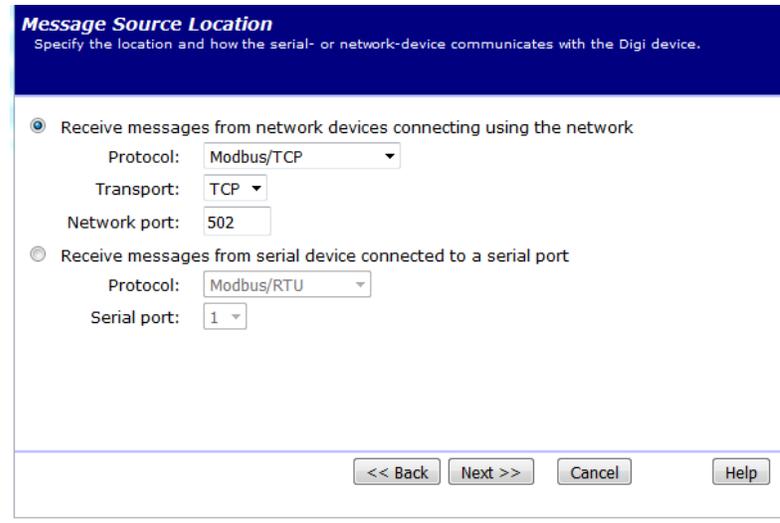
3.6 Master summary—incoming Ethernet/IP is ready

1. Click the Continue creating more message sources checkbox to enable incoming Modbus/TCP.
2. Click Next.



3.7 Define Modbus/TCP as the master or message source

1. Click the Receive messages from network devices using the network radio button.
2. Click the Protocol dropdown list and select Modbus/TCP. The Network port automatically sets to 502.
3. Click **Next**.



3.8 Master summary—both incoming masters all set

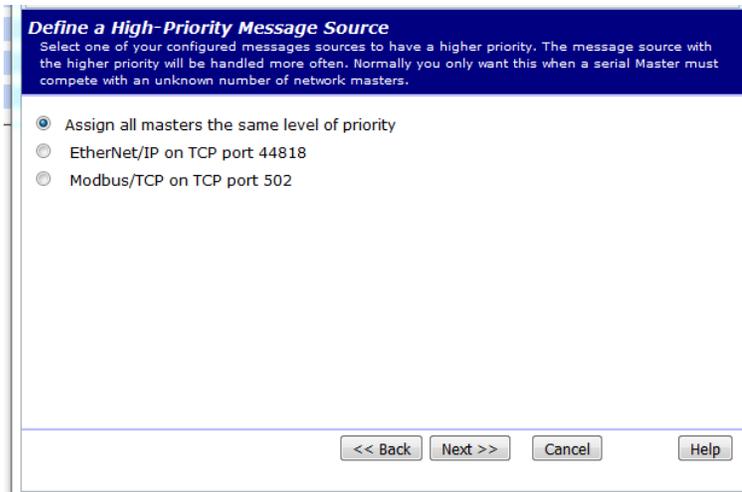
At this point you could add another master such as Allen-Bradley/Ethernet, an informal name for the CSP protocol used by pre-ControlLogix PLC5E and SLC5/05. But for this exercise, these two masters are enough.

Click **Next**.



3.9 Define master priority

You can assign a higher priority to Ethernet/IP or Modbus/TCP masters. For this exercise, however, accept the default to **leave the masters at the same priority**. Each incoming socket is given round robin access to the serial Modbus slaves.

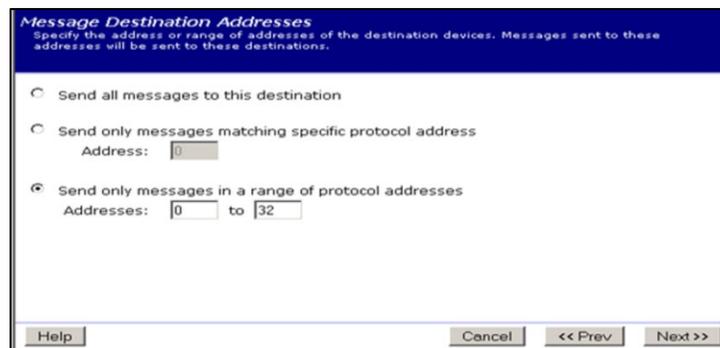


1. Click the Assign all the masters the same level of priority button.
2. Click **Next**.

3.10 Define first message destination—our serial port 1 to PLC(s)

You see the **Message Destinations** window, which is not shown in this document.

1. Click **Next** until you see the next window.
2. Click the **Send only messages in range of protocol addresses** radio button.
3. Enter **0** to **32** in the **Address** textboxes.
By default, address **0** maps to **1**, which is the DST or Destination Node number you will set in the ControlLogix MSG block.
4. Click **Next**.



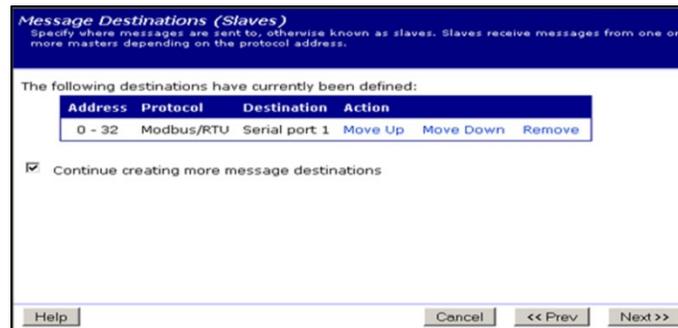
3.11 Set the protocol and port number

1. Click the **Protocol** dropdown list and select **Modbus/RTU**.
2. You'll move through the next few windows. The default settings are fine.
3. Click the **Serial port** dropdown list and select **1**.
4. Click **Next**.



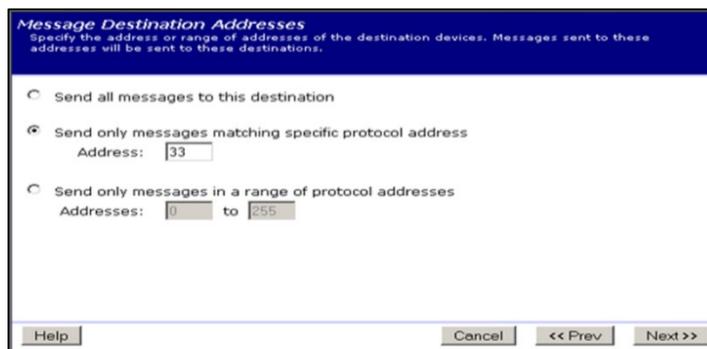
3.12 First destination complete

1. When you get to this summary, click the **Continue creating more message destinations** checkbox.
2. Click **Next**.



3.13 Define second message destination—PLC at 192.168.1.xxx

1. Click the Send only messages matching specific protocol address radio button.
2. Enter 33 in the Address textbox.
3. Click **Next**.



3.14 Set the Modbus/TCP slave details

1. Select **Modbus/TCP** to use either an IP address or DNS name.

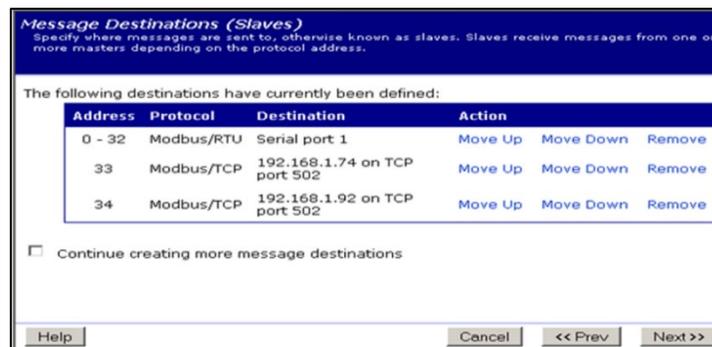
2. You'll move through a few more windows; the default settings are fine.
3. Click **Next**.



3.15 Define third message destination—PLC at 192.168.1.yyy

When you return to the summary, continue and create another destination for protocol address 24 to a **Modbus/TCP** slave. After you complete the third destination, you return to this summary.

1. Make sure the Continue creating more message destinations is unchecked.
2. Click **Next**.

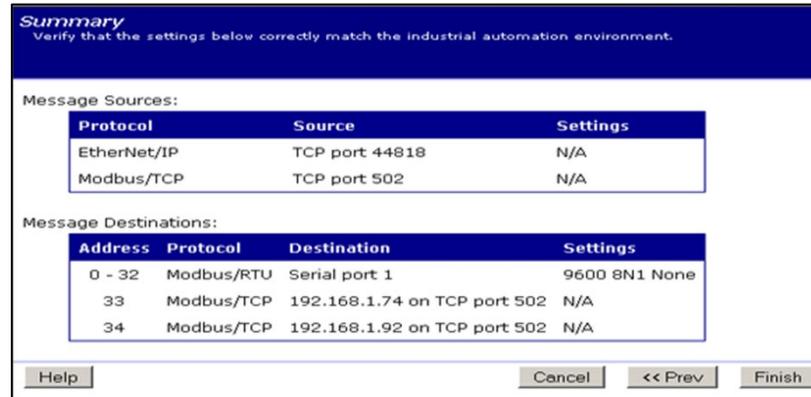


3.16 Finished—review the settings

Here is the final summary window. The Digi One IAP accepts messages by **Ethernet/IP** and **Modbus/TCP**, and it has these destinations:

- Up to 32 slaves on the **Serial port**
- Two remote Modbus/TCP slaves

1. Click **Finish**.



3.1 Reboot the Digi One IAP

You can make minor changes to the Digi One IAP configuration without rebooting. However, when you change the number or type of masters (message sources) or number or type of slaves (message destinations), it is safest to reboot. These changes, which affect the number and type of tasks running in the Digi One IAP RTOS, occasionally fail to take effect without a reboot.

4 Setup by telnet

1. To log into your Digi One IAP, use either HyperTerminal or telnet.
2. Enter the IP address of your DST and the well-known telnet port of 23.
3. Next is a text script you can cut and paste into an editor such as WordPad. Edit the script as required, then cut and paste again into HyperTerminal using **Edit | Paste to Host**.

```
# clear all IA config revert ia=factory

# setup port 1 as Modbus/RTU slave (baud = 9600,8,E,1) set port ra=1 dev=ia
set line ra=1 baud=9600 csize=8 parity=E stopb=1
set ia serial=1 protocol=mbrtu type=slave chrtout=50ms slvtout=1sec

# setup network for Ethernet/IP PCCC-encap and Modbus/TCP incoming
set ia master=1 active=on protocol=ethernetip transport=tcp ipport=44818 table=1 set ia
master=2 active=on protocol=modbustcp transport=tcp ipport=502 table=1

# setup destination table set ia table=1 name=table1
set ia table=1 addroute=1 active=on protocol=mbrtu
set ia table=1 route=1 protaddr=0-32 type=serial port=1
set ia table=1 addroute=2 active=on protocol=mbtcp protaddr=33 set ia table=1 route=2 type=ip
ipaddress=192.168.1.74 ipport=502 set ia table=1 addroute=3 active=on protocol=mbtcp
protaddr=34 set ia table=1 route=3 type=ip ipaddress=192.168.1.92 ipport=502

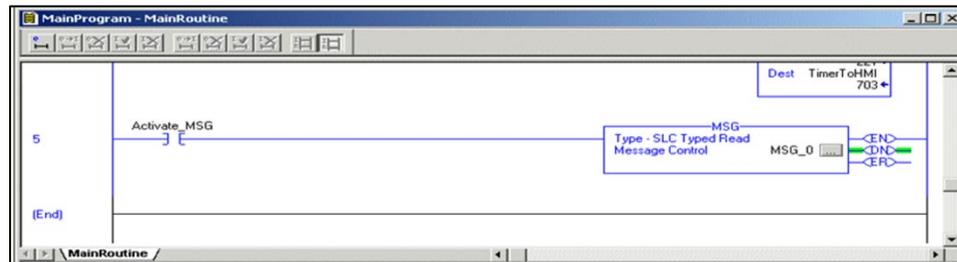
# reboot the Digi One IAP boot action=reset
```

5 Setting up the ControlLogix MSG Block

The instructions in this document are based on the assumption that you understand the basics of programming and accessing your ControlLogix PLC.

5.1 Create your MSG block rung in RSLogix 5000

1. Create a MSG block using whatever trigger is appropriate.
2. Create the tags you want to read to or write from. The tags can be either integer or string tags.

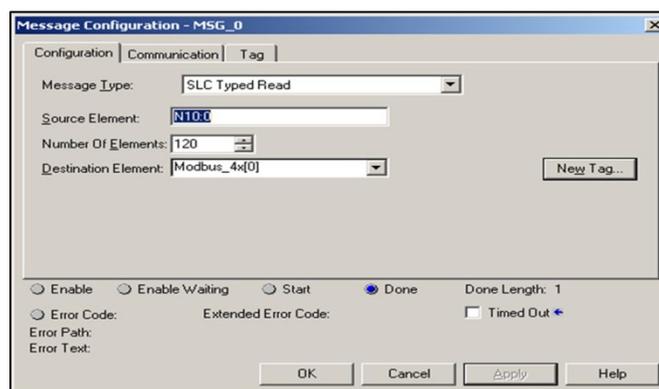


5.2 Configure the MSG for reading 0x, 1x, 3, or 4x areas

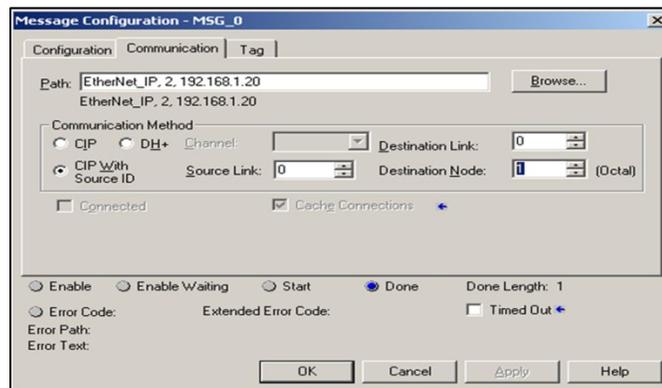
Reading these files	Map to
B0 to B249	Modbus function 1 to 0x area
B250 to B255	Modbus function 2 to 1x area
N0 to N249	Modbus function 3 to 4x area
N250 to N255	Modbus function 4 to 3x area

Select your MSG block in RSLogix and open it up for detailed configuration:

1. Click the Configuration tab.
2. From the Message Type pull down menu, select SLC Typed Read.
3. Set the Source Element as required. For example, N10:0 starts reading at 4x00001 or $((10-10)*100)+0+1$.
4. In the Number of Elements box, select the number of words to read. The SLC Typed Read is limited to 122 words, which is less than the 125-word limit of Modbus.
5. From the Destination Element pull down menu, select your ControlLogix tag—in this case, an integer array named Modbus_4x that can hold at least 120 words starting at [0].5.
6. Click **OK**.



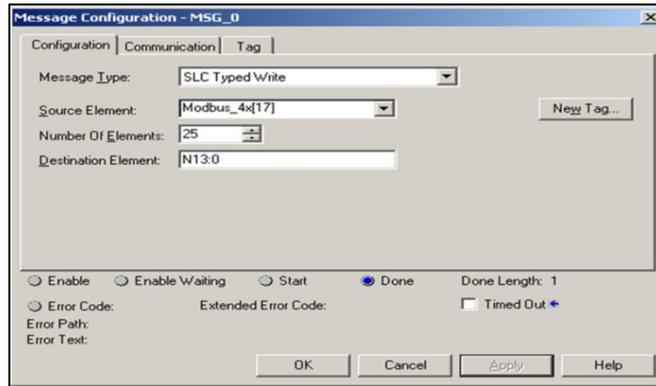
7. Click the **Communication** tab.
8. Set the **Path** as shown, using the Digi One IAP's IP address.
9. Click **CIP With Source ID**, and set the **Destination Node** to the slave address to index. In our example:
10. Modbus slaves 1 to 32 are Modbus/RTU serial slaves.
11. Modbus slaves 33 and 34 are remote Modbus/TCP slaves.
12. You'll need to use the appropriate slave address based on your Digi One IAP configuration.
13. Click **OK**.



5.3 Configure the MSG for writing 0x or 4x areas

Writing these files	Map to
B0 to B249	Modbus function 15 to 0x area
N0 to N249	Modbus function 16 to 4x area or Modbus function 6 if a single register

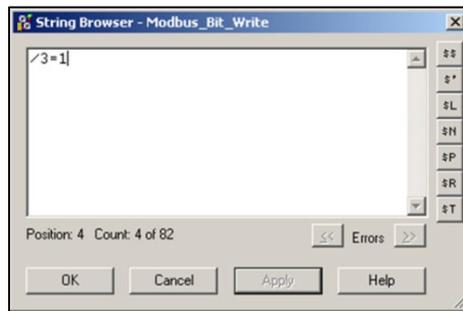
1. Select your MSG block in **RSLogix** and open it up for detailed configuration.
2. Click the **Configuration** tab.
3. Set **Message Type** to **SLC Typed Write**.
4. Set **Source Element** to your ControlLogix tag—in this case, the eighteenth integer in an array named Modbus_4x. ControlLogix arrays are zero-based.
5. Set **Number of Elements** to the number of words to write—25 in this case.
6. Set Destination Element as required. N13:0 starts writing at Modbus 4x00301 or $((13-10)*100)+0+1$.



7. Set the Communication Path, CIP with Source Id, and Destination Node on the Communication Tab as described above for the SLC Typed Read.

5.4 Configure the MSG for writing a single bit to 0x area

Writing single bits to Modbus is more challenging than writing words. Suppose you want to set Modbus bit 0x00036 to 1. The direct mapping to Rockwell file notation would be B10:2/3, but you cannot write this single bit from ControlLogix.

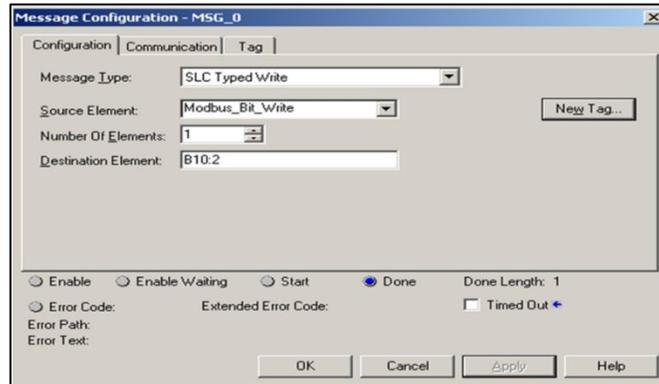


Create a string data tag and fill it with a string that defines:

- The bit to affect
- Whether to set the bit to 0 or 1

Then you write this string to word B10:2 with an SLC5 Typed Write.

1. Set Message Type to SLC Typed Write.
2. Set Source Element to your ControlLogix tag of type String.
3. Set Number of Elements to 1 element.
4. Enter a Destination Element as required.
5. B10:2 covers Modbus bits 0x00033 to 0x00048, and you are setting bit 3 (/3 is zero-based) or 0x00036.



6. Set the Communication Path, CIP with Source ID, and Destination Node on the Communication Tab as described above for the SLC Typed Read.

6 How to test this application

6.1 Testing by simulator

Protocol mapping is not as easy to test because it introduces many details. The easiest way to test may be to use a Modbus slave simulator, which allows you to see the actual polls. One simulator is ModSim32 from www.win-tech.com.

7 Troubleshooting and FAQ

7.1 Can I use PLC5 or CIF style commands?

Commands other than CIF/PLC2 unprotected reads/writes and SLC500 Typed Reads/Writes are processed by the Digi One IAP doing protocol mapping; however, the results may not be what you want.

The ControlLogix MSG block doesn't have any preference for PCCC commands. Therefore, you should just use the SLC Typed Read/Write if you want predictable results as documented in this application note.

7.2 Why is my MSG block returning a bad status?

If your Ethernet is working, and the ControlLogix is able to create an Ethernet/IP connection to the Digi One IAP, two conditions that can cause an error response include:

- The MSG block is sending something other than SLC Typed Reads/Writes.
- The MSG block is requesting file types other than "N" or "B."

Setting the timeouts

Setting a valid combination of timeouts requires the correct relationship between the four key timeouts you need to manage. The best defaults to start will be:

- The master application's slave timeout: 3 seconds
- The Digi One IAP's message timeout: default of 2.5 seconds

- The Digi One IAP's slave timeout: default of 1 second
- The Digi One IAP's character timeout: 50 milliseconds

These defaults safely allow two remote masters to share most slaves. For more information related to timeouts see the [support page for Digi One IAP](#) and the [Understanding Timeout Settings in Digi One IAP](#) application note.

8 How Rockwell polls turn into Modbus polls

Although your Rockwell master is issuing PCCC polls, the Digi One IAP must turn these into actual Modbus commands for Modbus slaves to be able to answer.

8.1 Given Modbus notation, determine your Rockwell notation

Download the [Digi Protocol Bridging Calculator for Master speaking Rockwell protocols](#) (part number 90000652), and use this Excel spreadsheet to convert your existing Modbus notations into the appropriate Rockwell SLC500 polls. With the Digi One IAP, it is important to determine the correct SLC500 address to poll the Modbus register.

8.2 How to document

If you supply a drive unit with Modbus/RTU communications, how can you enable Rockwell users to buy and manage your product? You refer them to the Digi One IAP and this application note, but that is only part of the solution.

The Digi One IAP allows customers to poll your driver concurrently by:

- DF1 serial in the second RS-232 pass-through port
- AB/Ethernet (CSP) from older SLC5 and PLC5E
- DF1 serial by remote DF1 encapsulation in TCP/IP
- ControlLogix using Ethernet/IP using SLC Typed Read/Write
- Modbus/RTU by serial or Modbus/TCP using the network

You also need to document your device in the AB file notation to simplify setup. Combining your Modbus/RTU product with a Digi One IAP and a good dual-protocol register map can open up new markets for your products with Rockwell Automation / Allen-Bradley users.

Here is a simple partial example:

AB	Modbus	Mode	Description	Values
N10:0	4x0001	Rd	Drive horsepower	In HP (i.e: 15 = 15hp)
N10:1	4x0002	Rd	Software version	BCD like 0101
N10:2	4x0003	Rd/Wr	Parameter lock	0/disable, 1/enable

Connecting Rockwell ControlLogix to Modbus slaves by Digi One IAP

AB	Modbus	Mode	Description	Values
N10:3 N10:3/0 N10:3/1 N10:3/2 N10:3/3 N10:3/4	4x0004	Rd/Wr	Run commands bit 1: Stop / Brake bit 2: Forward Run bit 3: Reverse Run bit 4: Fault Reset bit 5: Coast to Stop	Only 1 bit may be set per write, or the write is ignored. Read back shows current status.
N10:4	4x0005	Rd	Current	By 0.1A
N10:5	4x0006	Rd	Frequency	By 0.01 Hz
N10:6	4x0007	Rd	Voltage	Volts
N10:7	4x0008	Rd	Output power	Watts
N10:8	4x0009	Rd	Speed	RPM