

Using NI LabVIEW with ioLogik E2210 servers

Alex Chen, Senior Engineer, MOXA Technical Support Dept.

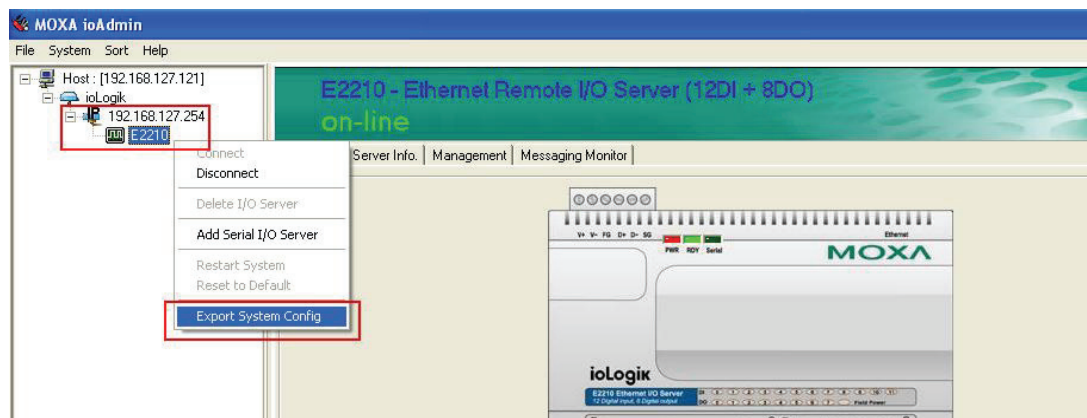
In this Technical Note, we cover the following topics:

1. Obtaining Modbus addresses from the ioLogik E2210 configuration file
2. Configuring NI LabVIEW for the ioLogik E2210

1. Obtaining Modbus addresses from the ioLogik E2210 configuration file

- 1.1. It is possible to obtain the Modbus address of each I/O channel on the ioLogik E2210 by exporting the system configuration. This information may be required when developing a Modbus application.

Run ioAdmin by clicking **Start → Program Files → ioLogik → Utility → ioAdmin**. In the left panel, right click on the ioLogik E2210 whose address table you wish to export, and then select **Export System Config** to save the configuration file.



Copyright © 2007 The Moxa Group

Released on February 9, 2007

About MOXA

The Moxa Group manufactures one of the world's leading brands of device networking solutions. Products include serial boards, USB-to-serial Hubs, media converters, device servers, embedded computers, Ethernet I/O servers, terminal servers, Modbus gateways, industrial switches, and Ethernet-to-fiber converters. Our products are key components of many networking applications, including industrial automation, manufacturing, POS, and medical treatment facilities.

How to Contact MOXA

Tel: 1-714-528-6777

Web: www.moxa.com

Fax: 1-714-528-6778

E-mail: info@moxa.com

This document was produced by the MOXA Technical Writing Center (TWC). Please send your comments or suggestions about this or other MOXA documents to twc@moxa.com.

- 1.2. The exported system configuration will appear as follows. This table can be used to retrieve the ioLogik E2210 system, I/O configuration, and Modbus address table.

```

ik2210.txt - Notepad
File Edit Format View Help
ioLogik E2210 Network I/O Server Configuration
=====
Date: 1/29/2007
Time: 3:32:23 PM
Firmware: 1.2.9.0
[1. Model]
MOD_TYPE=E2210 - Active Ethernet I/O Server (12DI + 8DO)
MOD_LOC=
MOD_NAME=

[2. I/O Configurations]
-----
DI00=0, (DI), DI00_FILTER=100, (50.00ms)
DI01=0, (DI), DI01_FILTER=100, (50.00ms)
DI02=0, (DI), DI02_FILTER=100, (50.00ms)
DI03=0, (DI), DI03_FILTER=100, (50.00ms)
DI04=0, (DI), DI04_FILTER=100, (50.00ms)
DI05=0, (DI), DI05_FILTER=100, (50.00ms)
DI06=0, (DI), DI06_FILTER=100, (50.00ms)
DI07=0, (DI), DI07_FILTER=100, (50.00ms)
DI08=0, (DI), DI08_FILTER=100, (50.00ms)
DI09=0, (DI), DI09_FILTER=100, (50.00ms)
DI10=0, (DI), DI10_FILTER=100, (50.00ms)
DI11=0, (DI), DI11_FILTER=100, (50.00ms)

DO00=0, (DO), DO00_PWN=0, (Off), DO00_SAFE=0, (Off)
DO01=0, (DO), DO01_PWN=0, (Off), DO01_SAFE=0, (Off)
DO02=0, (DO), DO02_PWN=0, (Off), DO02_SAFE=0, (Off)
DO03=0, (DO), DO03_PWN=0, (Off), DO03_SAFE=0, (Off)
DO04=0, (DO), DO04_PWN=0, (Off), DO04_SAFE=0, (Off)
DO05=0, (DO), DO05_PWN=0, (Off), DO05_SAFE=0, (Off)
DO06=0, (DO), DO06_PWN=0, (Off), DO06_SAFE=0, (Off)
DO07=0, (DO), DO07_PWN=0, (Off), DO07_SAFE=0, (Off)

[3. Modbus address table]
-----
CHANNEL I/O TYPE MODBUS REFERENCE MODBUS ADDRESS (Dec, Hex)
DI00 Input 10001 0000, 0x0000
DI01 Input 10002 0001, 0x0001
DI02 Input 10003 0002, 0x0002
DI03 Input 10004 0003, 0x0003
DI04 Input 10005 0004, 0x0004
DI05 Input 10006 0005, 0x0005
DI06 Input 10007 0006, 0x0006
DI07 Input 10008 0007, 0x0007
DI08 Input 10009 0008, 0x0008
DI09 Input 10010 0009, 0x0009
DI10 Input 10011 0010, 0x000A
DI11 Input 10012 0011, 0x000B

DO00 output 00001 0000, 0x0000
DO01 output 00002 0001, 0x0001

```

Use the information in the configuration file to determine each channel's Modbus address. The second section of the file contains information on each channel's configuration. In the example, we can obtain the following information about digital input channels 00 and 01 and digital output channels 00 and 01:

Channel No.	Configuration
DI00	DI DI00_FILTER=100, (50.00ms)
DI01	DI DI01_FILTER=100, (50.00ms)
DO00	DO DO00_PWN=0, (Off), DO00_SAFE=0, (Off)
DO01	DO DO01_PWN=0, (Off), DO00_SAFE=0, (Off)

The third section of the configuration file contains the Modbus address for each channel, under Modbus reference:

```
[3. Modbus address table]
```

```
-----
CHANNEL NO.      I/O TYPE      MODBUS REFERENCE  MODBUS ADDRESS (DEC, HEX)
DI00             Input         10001             0000, 0x0000
DI01             Input         10002             0001, 0x0001
DO00             Output        00001             0000, 0x0000
DO01             Output        00002             0001, 0x0001
```

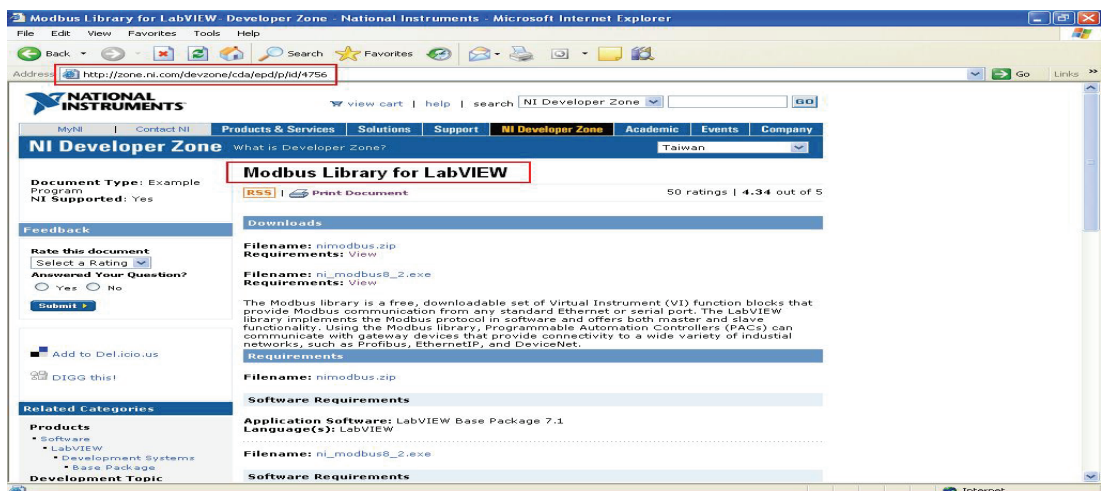
- 1.3. The ioLogik E2210 supports Modbus function codes as defined by the Modbus protocol. Each data type has a specific address range. A quick overview of the address types and function codes can be found in the following table. For more detailed information on Modbus address mapping, please refer to the User's Manual.

I/O Type	Address Range	Data Type	Common Name	Read/Write Behavior	Function Codes
Event Counter	30001 to 39999	Read Only Registers	analog inputs	16-bit quantity, provided by an I/O system, read-only	04=Read Input Register
Digital Input	10001 to 19999	Read Only Coils	binary inputs	single bit, provided by an I/O system, read-only	02=Read Discrete Inputs
Pulse Output	40001 to 49999	Read/Write Registers	analog values, variables, registers	16-bit quantity, alterable by an application program, read-write	03=Read Holding Registers 06 = Write Single Register 16 = Write Multiple Registers
Digital Output	00001 to 09999	Read/Write Coils	bits, binary values, flags	single bit, alterable by an application program, read-write	01=Read Coil 05 = Write Single Coil 15 = Write Multiple Coils

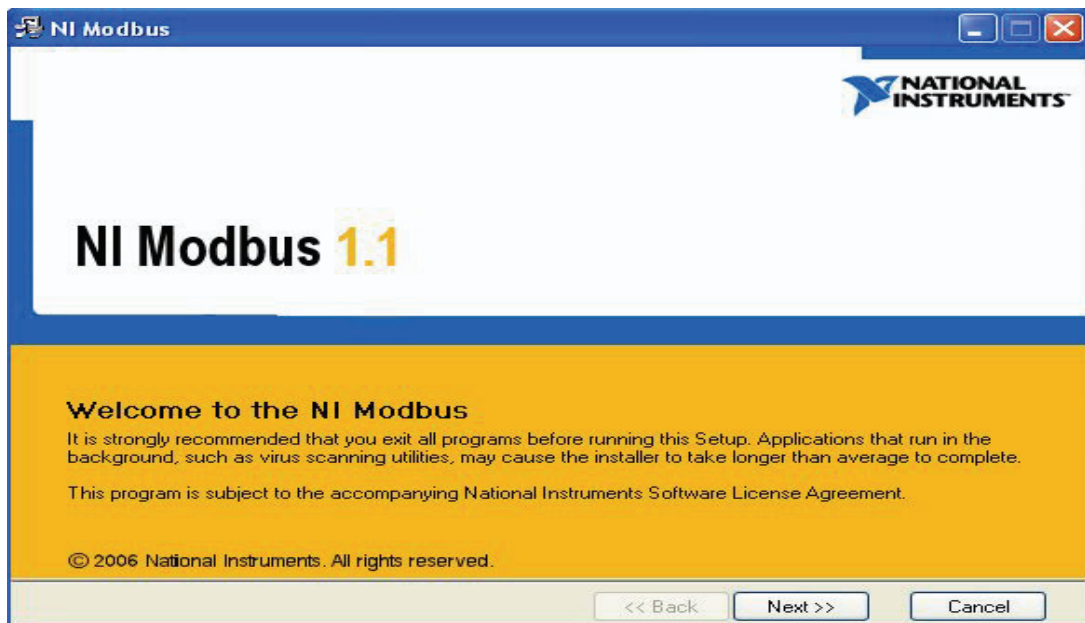
In the previous example, the value of digital input channel 00 would be read by referencing Modbus address 10001 and function code 02. To read/write digital output channel 00, you would reference Modbus address 00001 and function codes 01 and 05.

2. Configuring NI LabVIEW for the ioLogik E2210

- 2.1. The LabVIEW Modbus library implements the Modbus protocol in software and offers both master and slave functionality. Using the Modbus library, Programmable Automation Controllers (PACs) can communicate with ioLogik E2210 servers. You must first download the Modbus library from the NI web site. (<http://zone.ni.com/devzone/cda/epd/p/id/4756>)

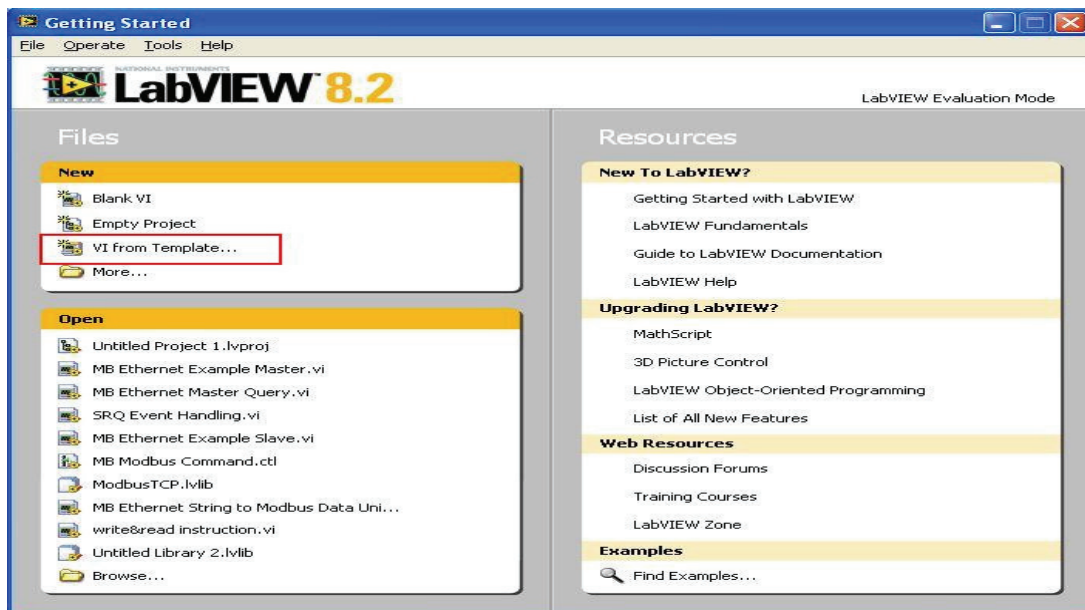


- 2.2. After downloading Modbus library, run the installation program.

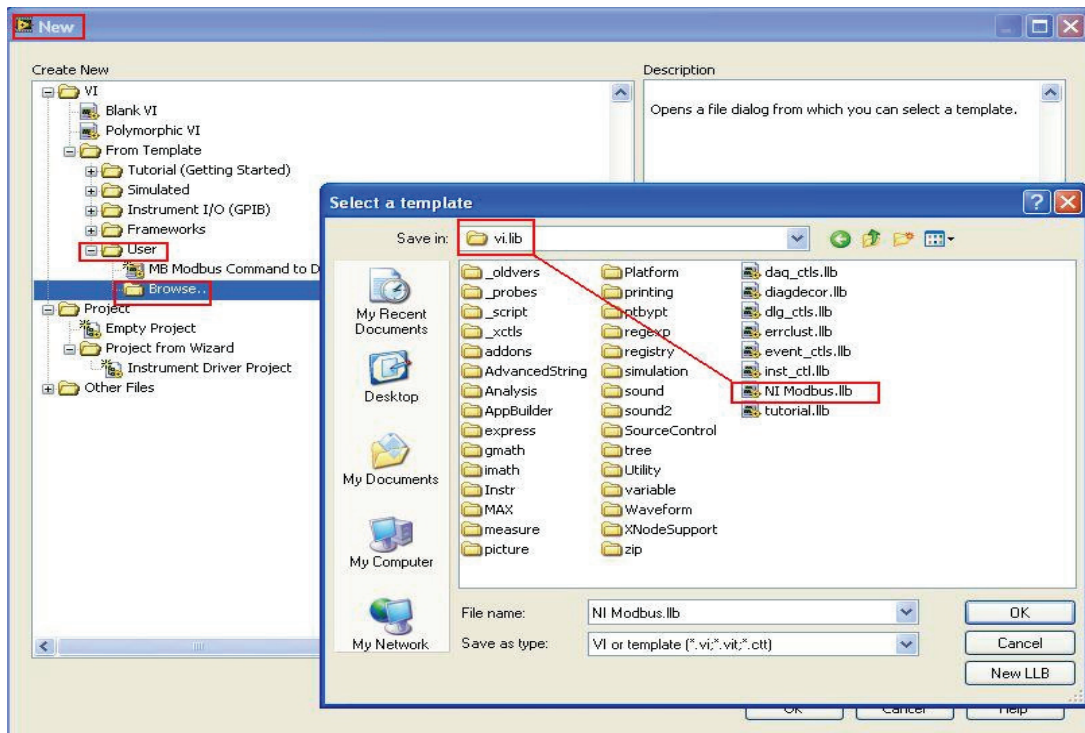


- 2.3. Start NI LabVIEW under **Start → All Programs → National Instruments → LabVIEW 8.2.**

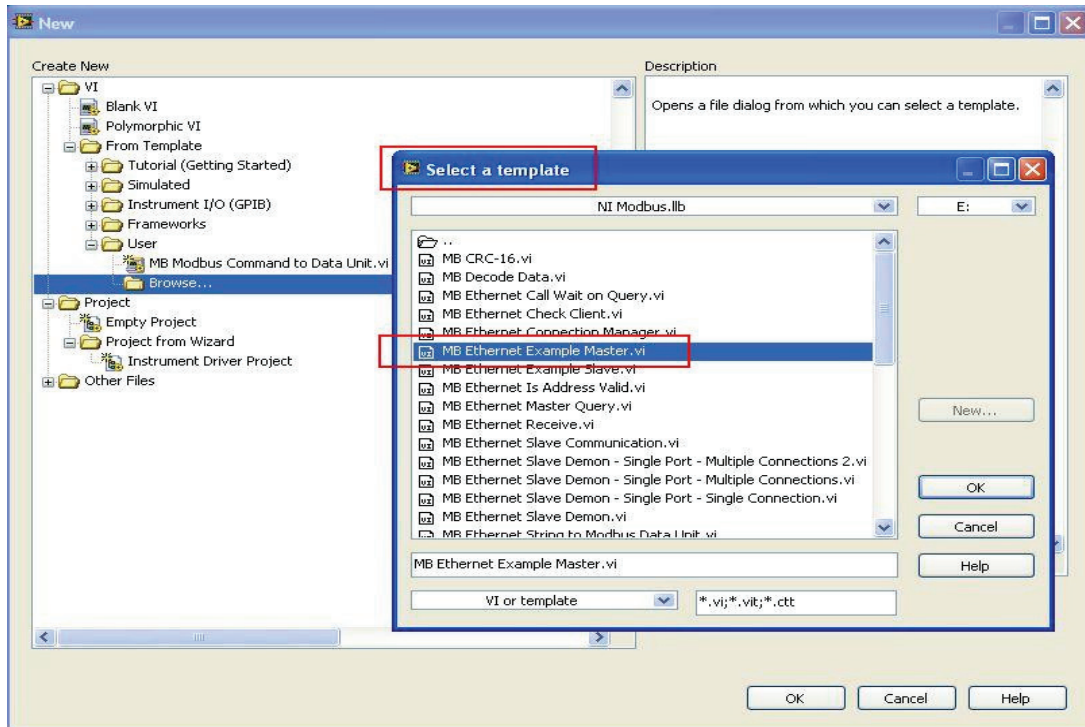
- 2.4. Select **VI from Template...** from the main screen.



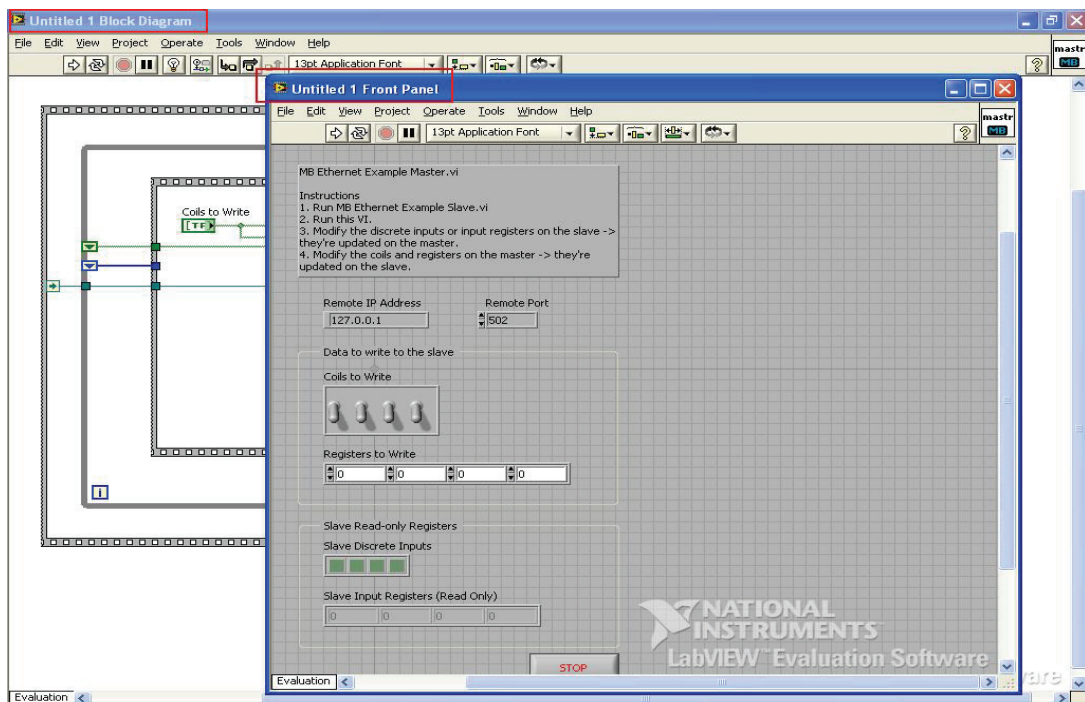
- 2.5. Go to **User → Browse...** to select a file to open. Find and open **NI Modbus.lib**, which should be located in **C:\Program Files\National Instruments\LabVIEW 8.2\vi.lib**.



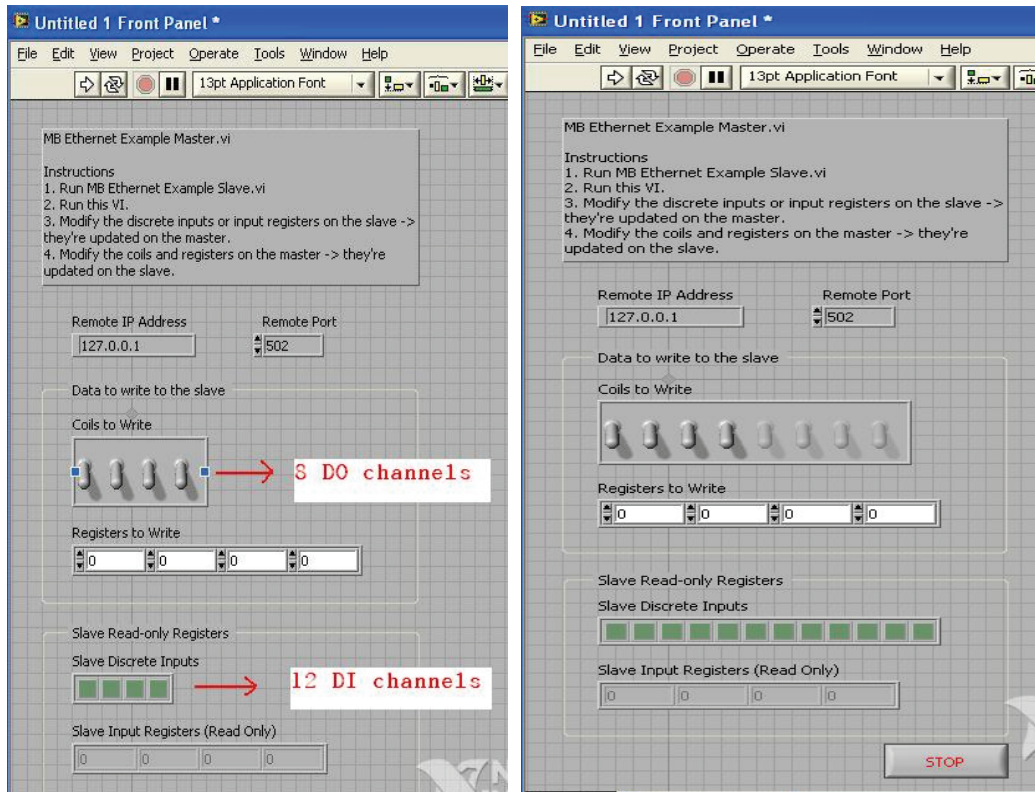
- 2.6. You will be prompted to select a template. Select **MB Ethernet Example Master.vi**. This example will be used to query and control the ioLogik E2210 server's DI and DO channels.



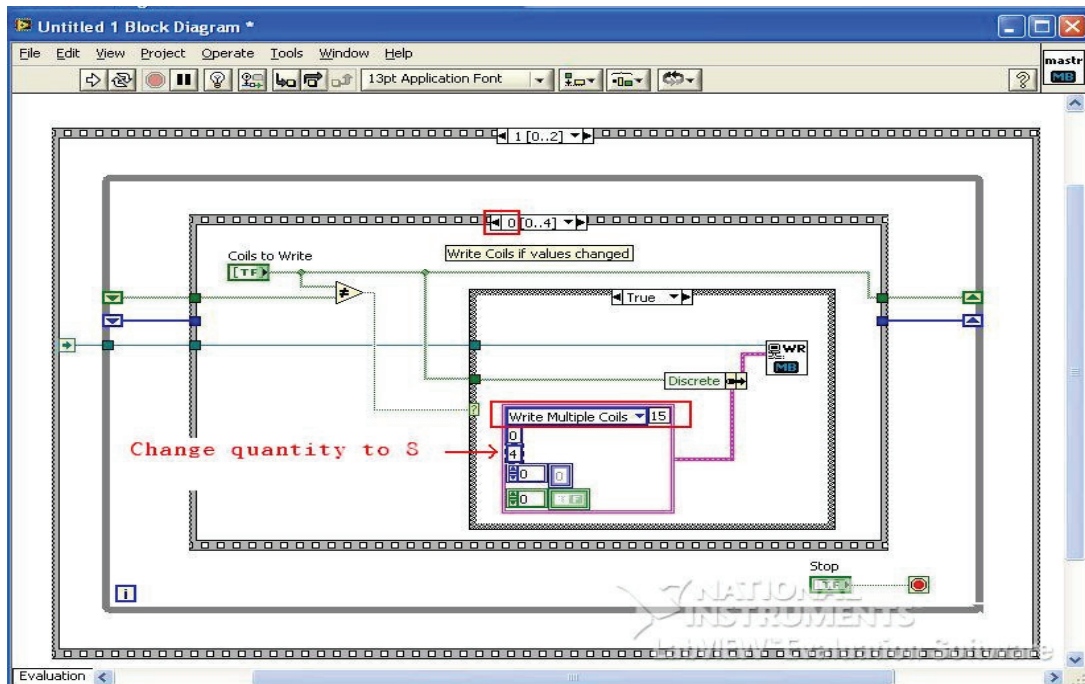
- 2.7. When the file opens, the Front Panel and Block Diagram will be displayed.



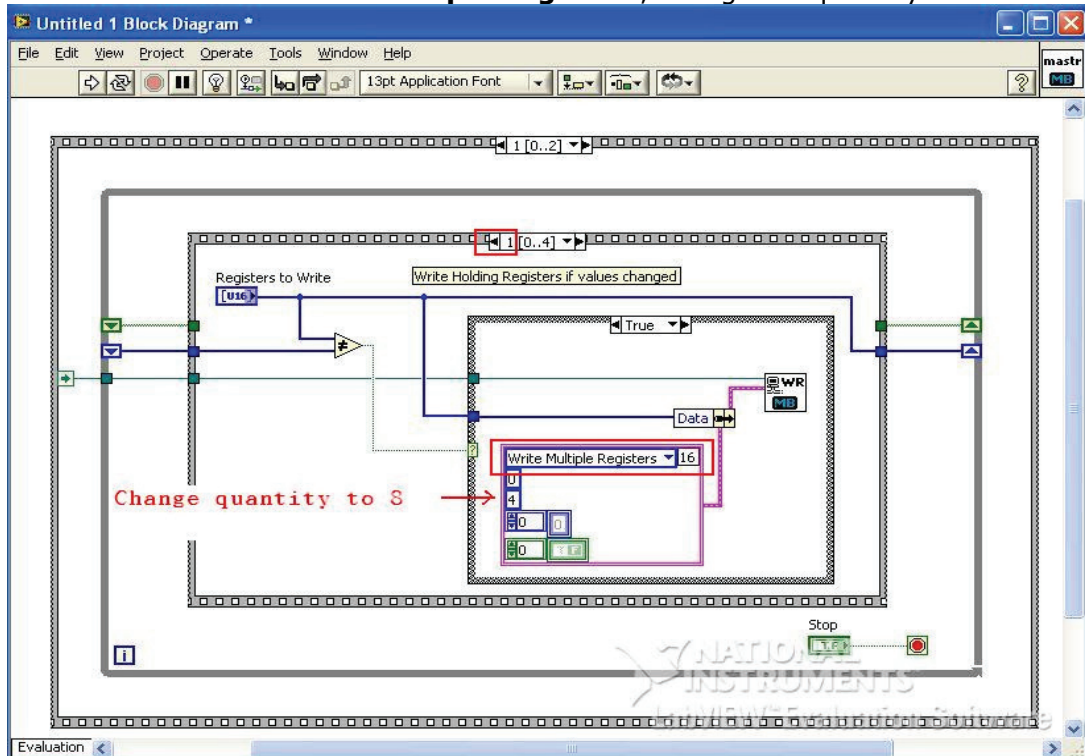
- 2.8. The iLogik E2210 has 8 DO channels and 12 DI channels. In the Front Panel window, click and drag the **Coils to Write** box until 8 channels are showing. Click and drag the **Slave Discrete Inputs** box until 12 channels are showing.



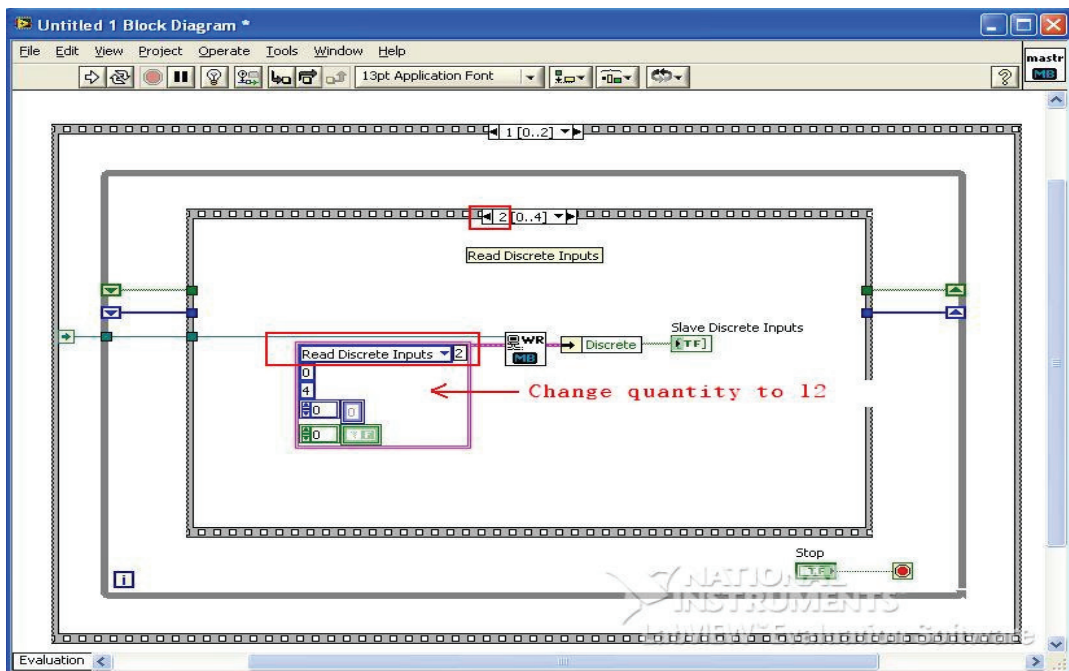
- 2.9. In the Block Diagram, find **Label 0**. Under **Write Multiple Coils**, change the quantity from 4 to **8**.



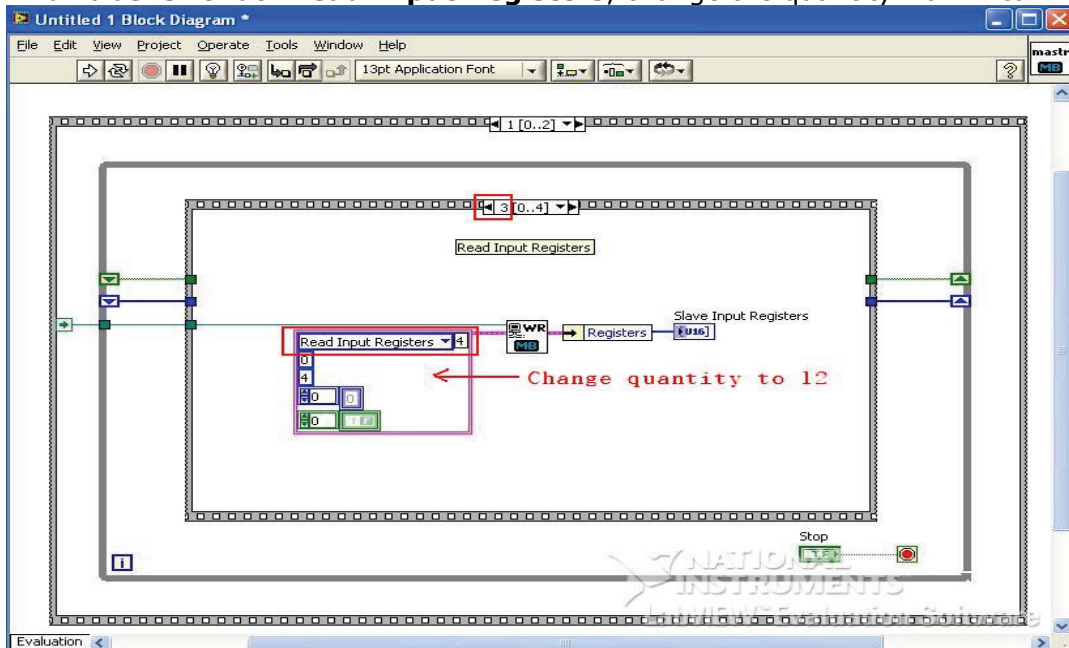
- 2.10. Find **Label 1**. Under **Write Multiple Registers**, change the quantity from 4 to **8**.



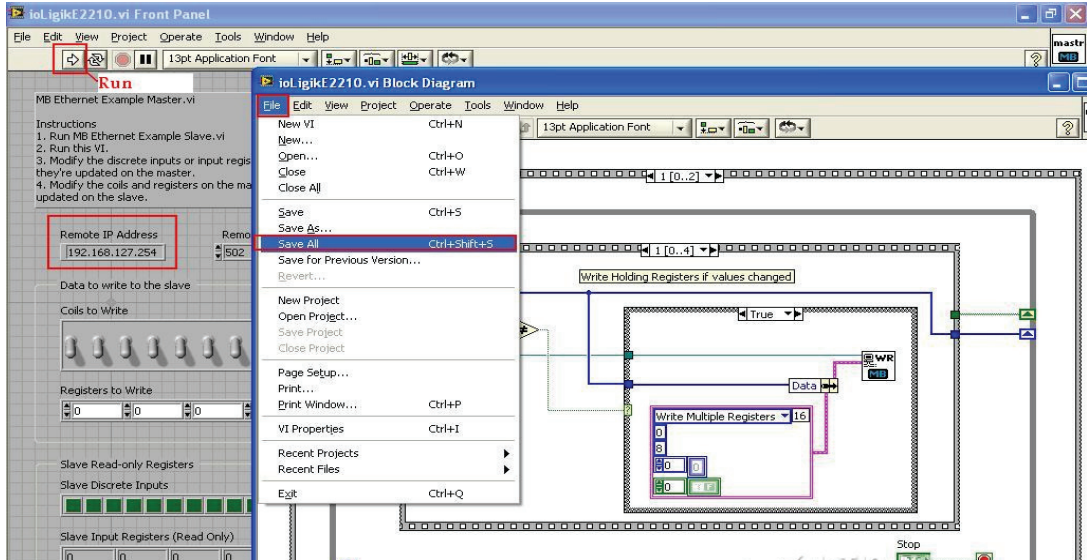
2.11. Find **Label 2**. Under **Read Discrete Inputs**, change the quantity from 4 to **12**.



2.12. Find **Label 3**. Under **Read Input Registers**, change the quantity from 4 to **12**.



- 2.13. Save the configuration through **File → Save All**. After the configuration is saved, enter the ioLogik E2210's IP address (default = 192.168.127.254) in the **Remote IP Address** field. Press the **Run** button to execute this VI.



- 2.14. In the Front Panel, the **Coils to Write** buttons can be manipulated to control DO values. You also can monitor DI status through the **Slave Discrete Inputs** lights. If you monitor I/O status with the ioAdmin utility, you should be able verify that the results match.

