

This appnote only applies to DirectSoft Ver 5 users, and the CPUs that can support IBOX's. If both conditions aren't met, then Chapter 5 in the ECOM manual deals with the original RX/WX instructions and Modbus TCP. This document is meant to show how to use newer instructions and more detail into the process of using ECOM100s for ModbusTCP.

For DirectSoft 5 users, there are 2 specific IBOX's that can help with Modbus TCP communications and one more IBOX that must be used to sequence these instructions: ECRX, ECWX, and ECOM100. This appnote doesn't give full details about these instructions, that information is already in the PLC User Manual, Chap 5 of the particular PLC/CPU that you are using.

The ECOM100 IBOX must be placed at the top of ladder, with no input logic. You will need one box for each ECOM100 you wish to use. The slot location of the ECOM100 is assigned to an ECOM # here, as well as the address ranges needed by the instruction. Note that this range **MUST** be unique and cannot be used for any other purpose. The same is true for ANY workspace V-memory assignment in any IBOX's used.

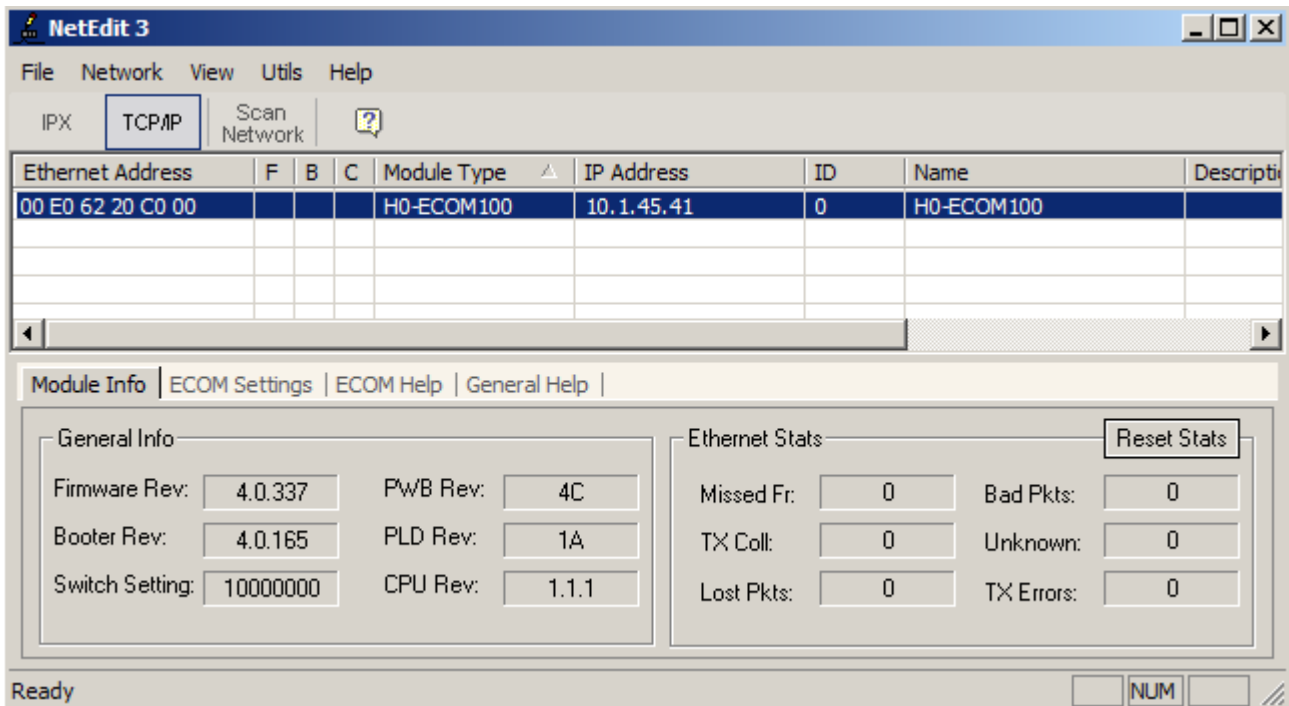
These instructions auto-sequence themselves, but DirectLogic octal addressing still must be used. There is a Modbus spreadsheet located on our Tech Support site that will be needed to convert Modbus addresses in the slaves to the octal addressing that must be used in the ECxX boxes.

Link is here: http://support.automationdirect.com/docs/modbus_conversion.xls

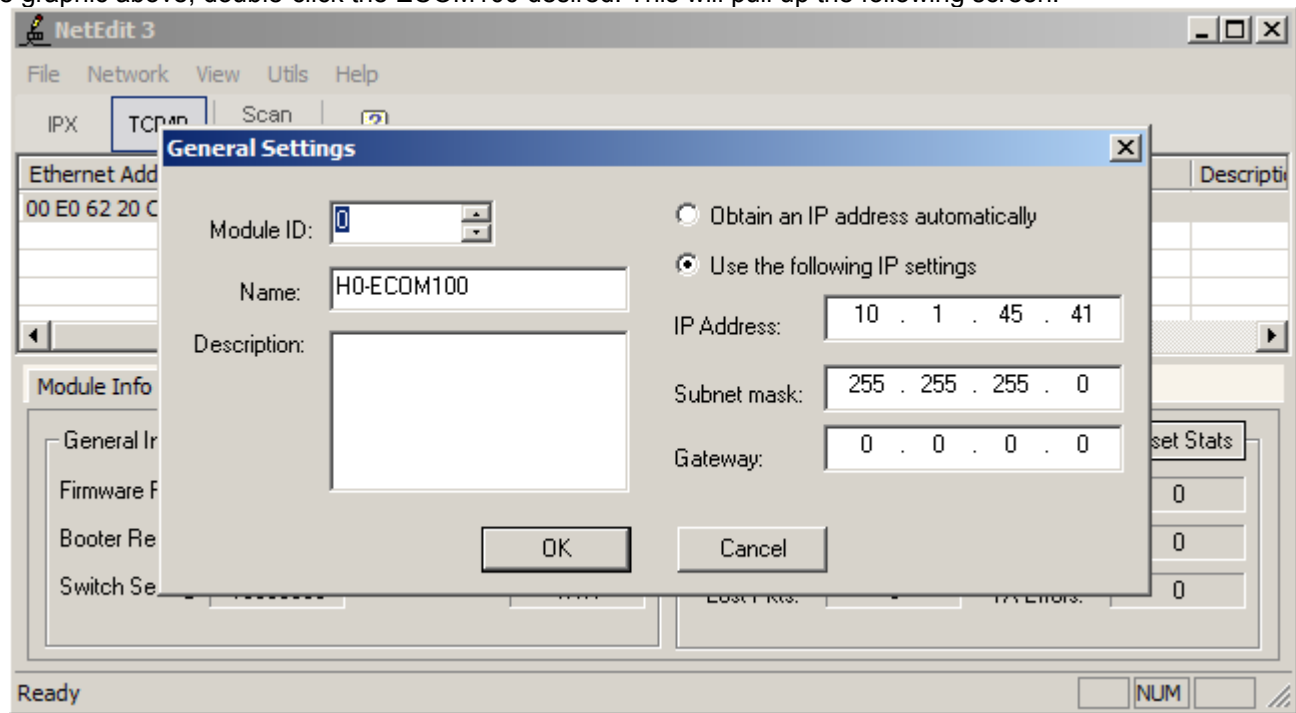
Once you have used the spreadsheet to determine the appropriate DL address to use for your desired Modbus address, place this value into the "From Slave Element" field. Enter the # of bytes you wish to retrieve, and enter the appropriate DL address you want to receive this data. The "Slave ID" at this time has no bearing to the actual slave, it will be tied to the IP address of the Modbus device in the NetEdit 3 and Peer-Peer setup below.

For example, if you want to read the first Modbus coils, you would place "GY1" in "From Slave Address", the # of bytes, and you could place the data into C400, for example. For Holding Registers, you would place the data into V-memory locations. The example program at the bottom will READ V40001 and V40002 from the Modbus slave, and immediately WRITE V7766 and V7767 (RTC Seconds and Minutes) to 40003 and 40004. So if the slave has no data there, the PLC will send its' Seconds and Minutes values (except the DL05 which will send 0 unless it has the Real-Time Clock module installed).

For the final steps, you must setup the ECOM100 IP settings and "Peer-Peer" table with NetEdit3.

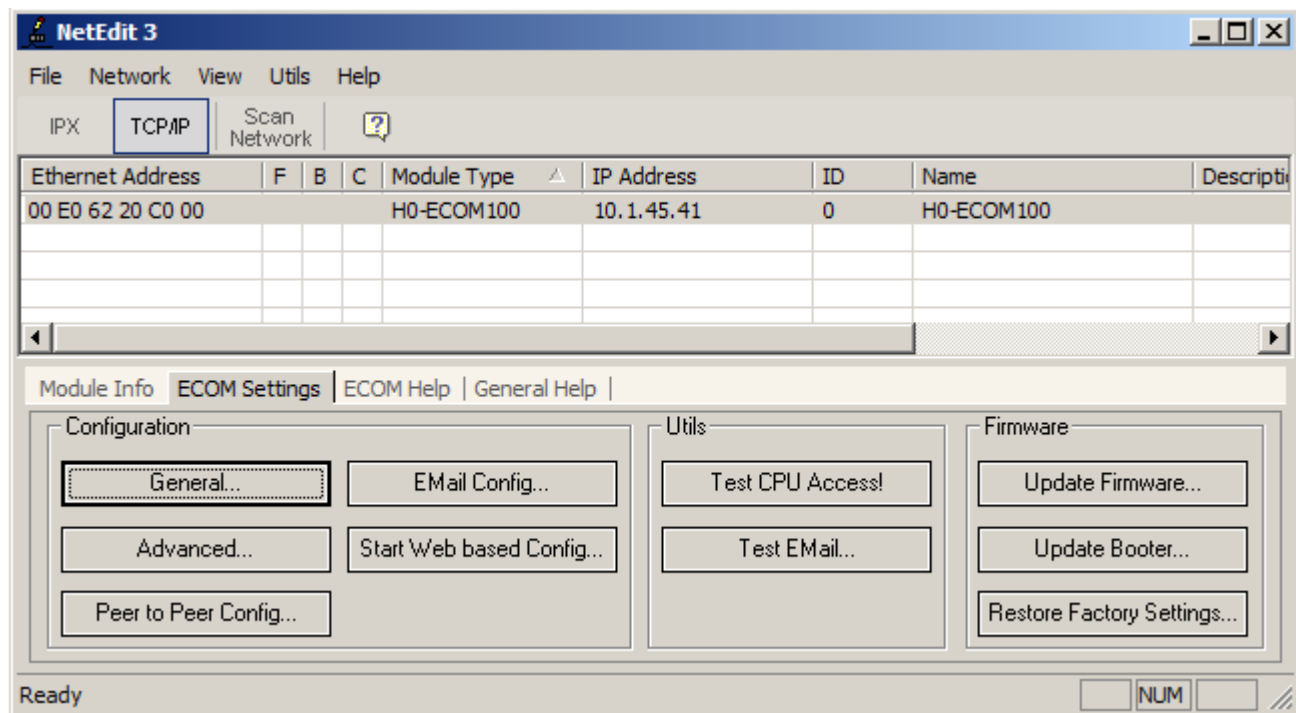


In the graphic above, double-click the ECOM100 desired. This will pull up the following screen:

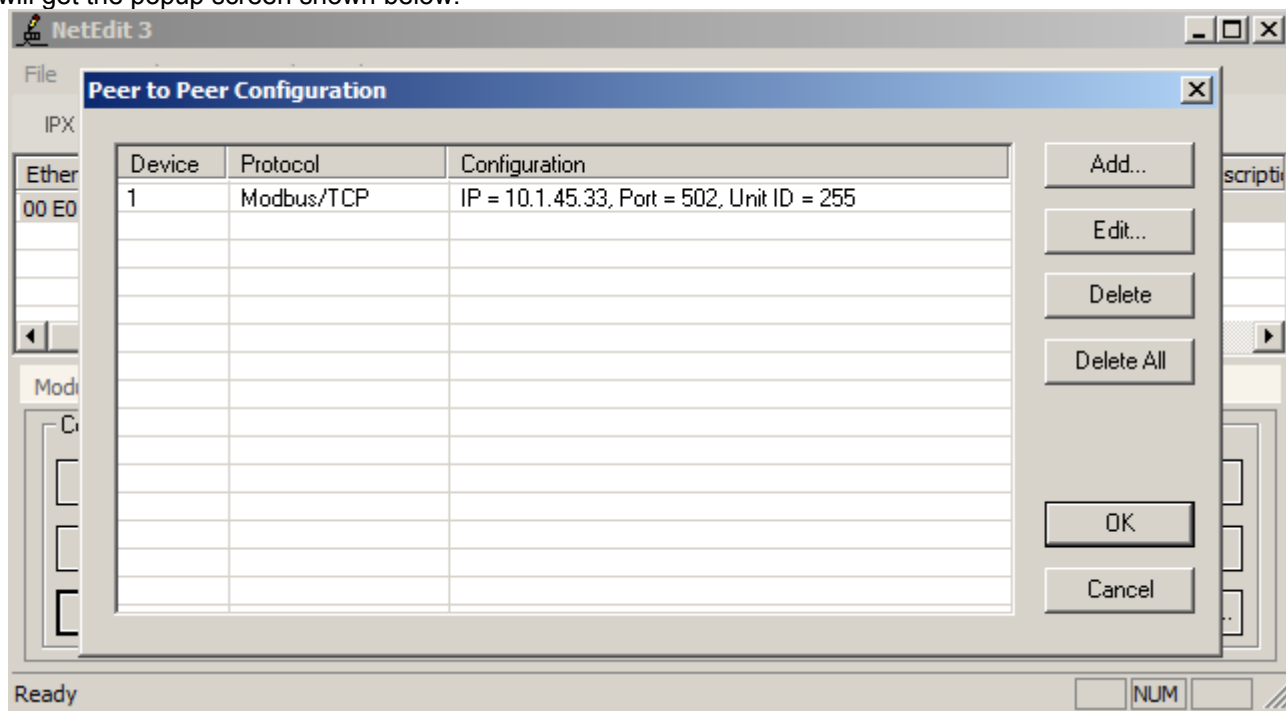


Here, the user can assign the IP address and subnet mask for his network. The Modbus slaves will need to have compatible settings of course. When finished , click OK to go back to NetEdit3 main screen.

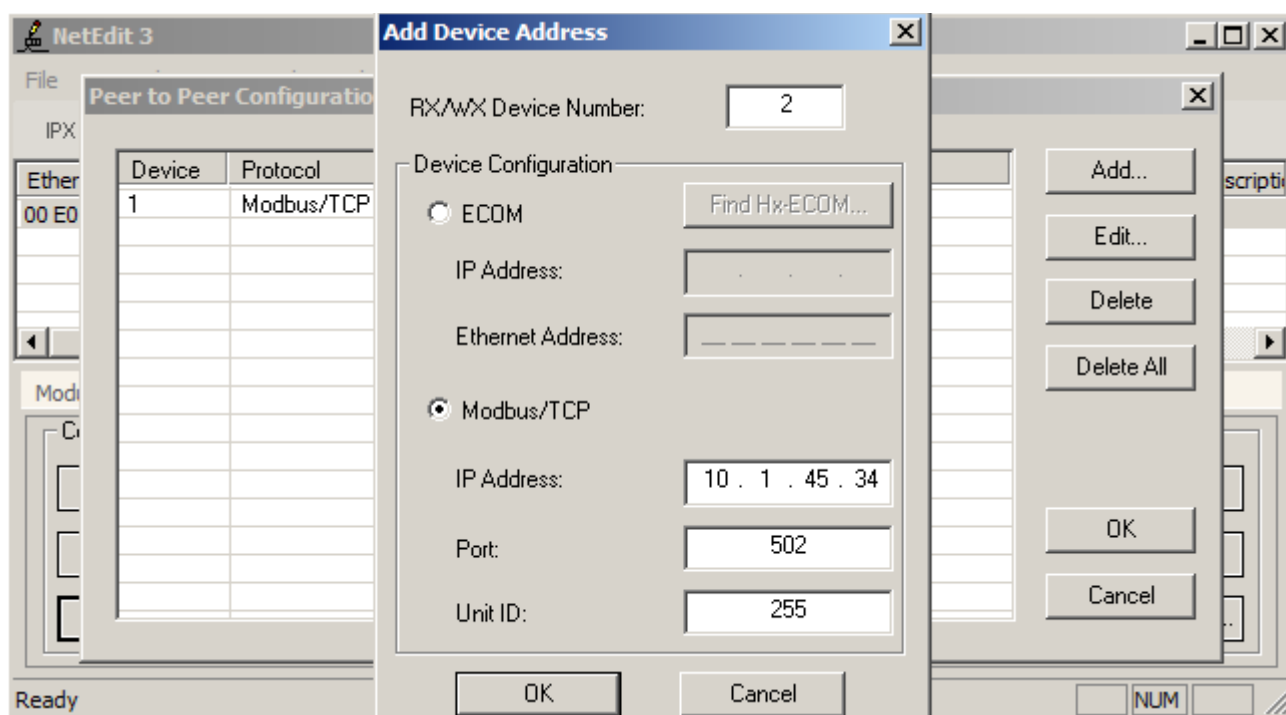
To begin the ModbusTCP setup, highlight the desired ECOM100 module , and select the “ECOM Settings” tab at the bottom of NetEdit, and click the button marked “Peer to Peer Config”.



You will get the popup screen shown below:



Here, to add an entry, click the “Add” button, or “Edit” to make changes to the existing settings.

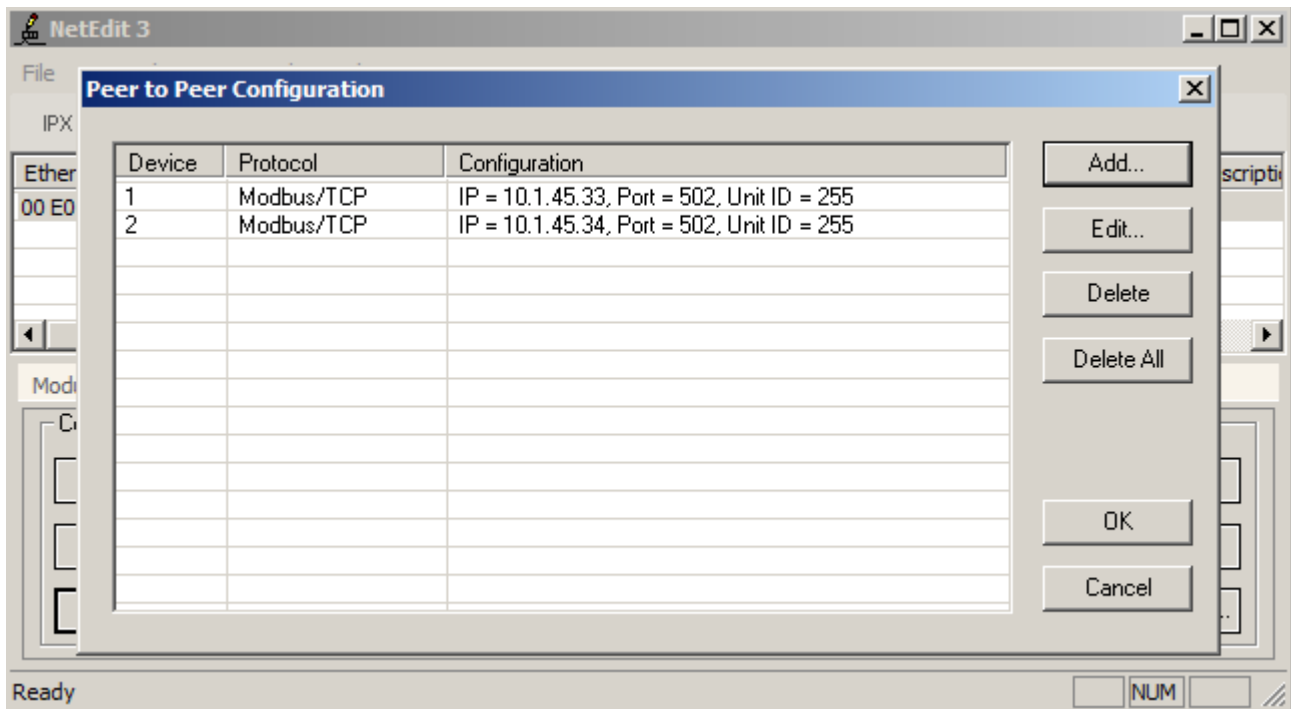


Make sure to check the box for “ModbusTCP” and make “Device #” equal to the “Slave ID” that was used in the ECxX box(s), and enter your IP address of the Modbus slave. Leave the UnitID at 255 and port at 502 unless your slave documentation instructs otherwise.

NOTE: UnitID is normally used with ethernet-serial gateways, where one IP address (the gateway) may be used for many serial slaves. Here, the UnitID would represent the serial slave address, and would need to be changed to match the various slaves.

So the IP address might be 10.1.45.34 for a Modbus gateway and the UnitID would be “1” for the first serial slave, “2” for the second, and so on.

Once you hit “OK”, the ECOM100 will be updated, there is no further action to take place with NetEdit until you are ready to add additional information to the table.



At the bottom is the example program. Note that it is only using Device ID 1, the user could easily add additional IBOXs.

After making sure your program has an END statement, "Accept" (F8), and write the project to the PLC (Shift+F9). Make sure after you have written the project, that the PLC is placed into Program mode, then back into Run mode (many IBOXes are only processed at Program-to-Run transitions). If everything is correct, V505 (RX/WX OK Count) using the Success bit should be incrementing very fast.

Troubleshooting:

Modbus Addressing: You must know the addressing of your device. Some devices give addresses in hex values, which can often appear like a decimal value (310 is a valid hex or decimal value). Asian drives often use hex. You can use Windows calculator “scientific” view to convert the addresses.

Also, many devices use addresses as “offsets”, particularly Holding Registers. The address 40001 means the first Holding Register, but your device may term this as Holding Register 0 or 1 (the 40000 is assumed). Whether this is actually an address of 0 or 1 is hard to predict.

An excellent method of troubleshooting is to try to only read from the middle of a known address range, that will have non-zero values. If you know the device has 10 Holding registers starting at 1, try to read #3, and compare that to the values in the device. If that is one more or one less than you expect, then that is the offset you will have to use in the DirectLogic numeric conversion.

Make sure you can ping your device.

Make sure the PLC has made a Program-to-Run transition.

Make sure the LinkGood light is ON, on the ECOM100 module,

Make sure Dipswitch 7 is ON, on the ECOM100 module.

On rare occasions, and almost exclusively with Festo or Numatics devices, their Holding Registers start at a VERY high number, around 45,000. There isn't an equivalent octal address to convert that high a value, so a Z constant was introduced that allows the use of a hex value in the "From Slave Element" field. So a 45,392 address is actually the offset added to 40,000... $45,392 = B150$ hex, so the entry in the "From Slave Element" field would be ZB150.

There are 3rd party Modbus shareware programs available on the internet, and we have a free ModbusTCP tester at the link below. You might need to use one of these programs to test to your device, to make sure they can work successfully to the expected addressing.

http://ftp.automationdirect.com/pub/Modbus_TCP_Master.zip

To read Modbus Input Registers, you must change the number of bytes in any instruction to the next odd number. For example, if you are reading 16 bytes (8 Registers), then you would increase the number by 1 to 17 bytes, and the instruction will then be trying to read Modbus 30000 addresses.

Network #1 uses the Hx-ECOM100 in Slot 1. This would be the only slot in 05, first slot in 06, and second slot in 205/405 models.

It will use the range of V-memory from V400 - V502 as the working status, workspace and buffer. These locations must not be used anywhere else.

Make sure Dipswitch 7 is turned ON in the ECOM100.

The Modbus converter spreadsheet from ADC Tech Support site will be extremely useful to convert Modbus addresses into the octal-based addressing required in the instructions.

NOTE: NetEdit 3 MUST be used to setup the ECOM100 "Peer-to-Peer Config" table in the ECOM100. This is what determines if the communications are ModbusTCP or ECOM.

1	ECOM100 Config	
	ECOM100	IB-710
	ECOM100 #	K1
	Slot	K1
	Status	V400
	Workspace	V401
	Msg Buffer (65 WORDs)	V402 - V502

On the first PLC scan, set the Comm Success & Comm Error count registers to 0.

Also SETS C106, which is the enable logic to the ECxX boxes.



Once the ECRX and ECWX IBoxes are enabled, the ECOM100 IBox will automatically sequence them, no manual control of the port busy bits is required.

This example uses C106 with a SET on First Scan, it could be changed to whatever logic the user desires such as SP1.



The ECRX will read from Slave ID "1", and will target address TA0 (V0) which is the Modbus equivalent address 40001 (first Holding Register).

It will get 4 bytes (2 registers) and place the data in V2000-2001. This data will likely be in decimal format if coming from 3rd party devices. The DataView window at left has V2000-2001 set for Decimal format.

Note that Workspace V location must be unique.



The ECWX will write to Slave ID "1", and will target address TA2 (V2) which is the Modbus equivalent of 40003 (third Holding Register).

It will write 4 bytes (2 registers) from the PLC addresses V7766-V7767 . These 2 addresses are the Seconds and Minutes from the PLC RealTime Clock, so they are non-zero most of the time. Note the data will be in BCD/Hex format.

The DL05 will only have data here if using the Option module D0-01MC, otherwise it will be zeroes.

Note that Workspace location must be unique.



NOTE: If trying to read Modbus Input Registers (Function Code 4 or 30001 addressing), the # of bytes must be increased by 1 to the next odd number. This is how the DirectLogic and ECOM100 recognize the Modbus address is an Input Register.



