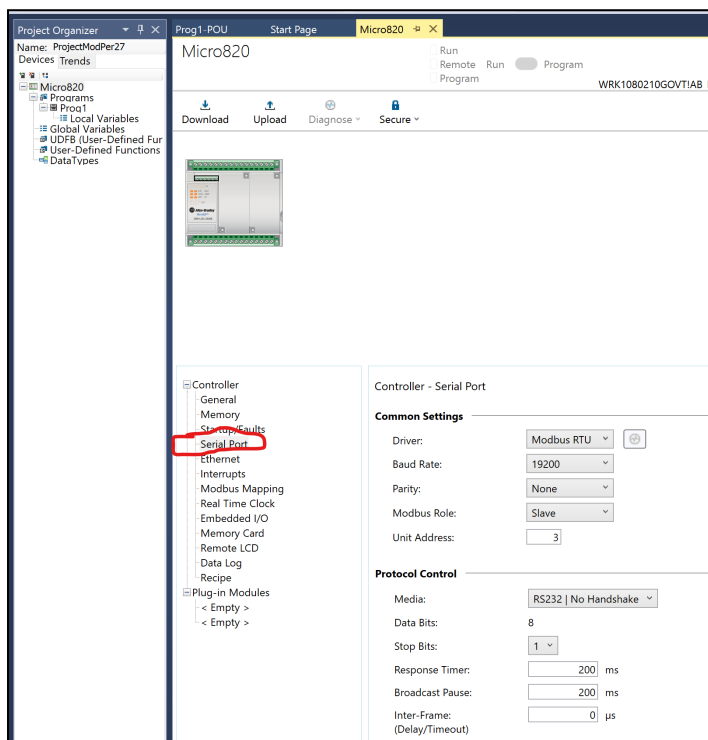


MICS Lab 07 Outline

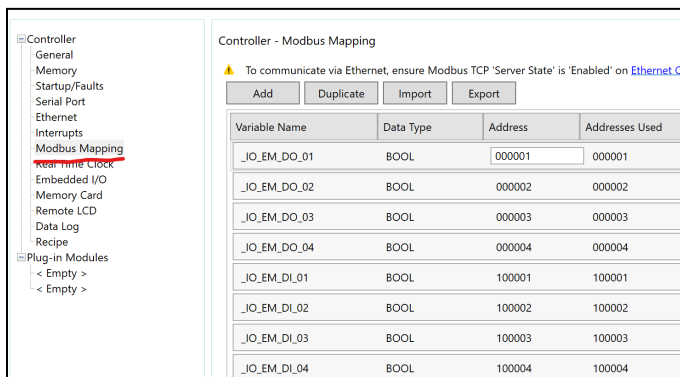
This is roughly following the following video: <https://www.youtube.com/watch?v=ARg2QHn3IB0>

Our goal is to configure the settings so that one PLC can manipulate the Coil Outputs of a remote PLC and also read in its Discrete Inputs.

1. Configure the middle PLC 192.168.0.27 to be the Peripheral ("Slave") device
 - a. Create a new CCW project, select m820 QBB controller
 - b. On the device Micro820 tab, select the Serial Port item from the tree
 - i. If required, click the Remote LCD link to uncheck the top box that is defaulted on
 - c. Select Modbus RTU for Driver, select "Slave" for the Modbus Role, and enter the unit address of "2" this device.

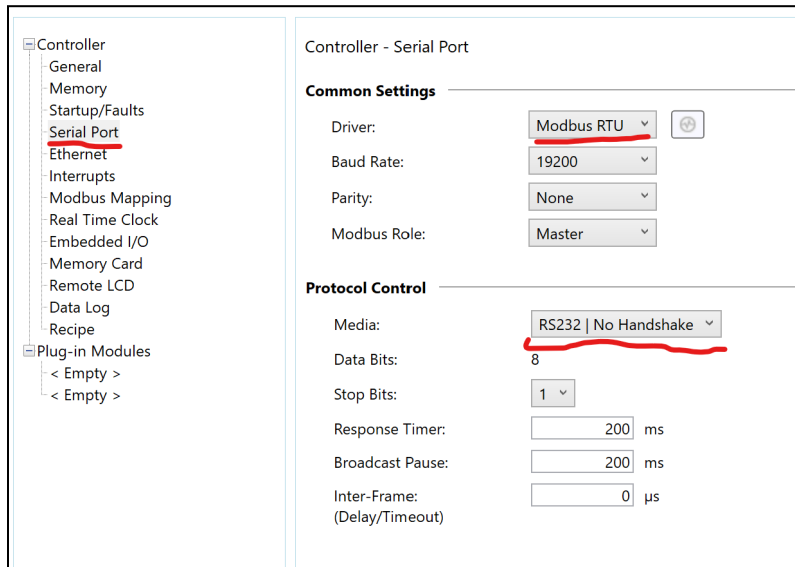


- d. In the Protocol section select RS232, and leave everything else default.
- e. Next select the Modbus Mapping item, click the Add button and click the DO_01 output (the blue light) and type in an address for the "first" coil = 000001

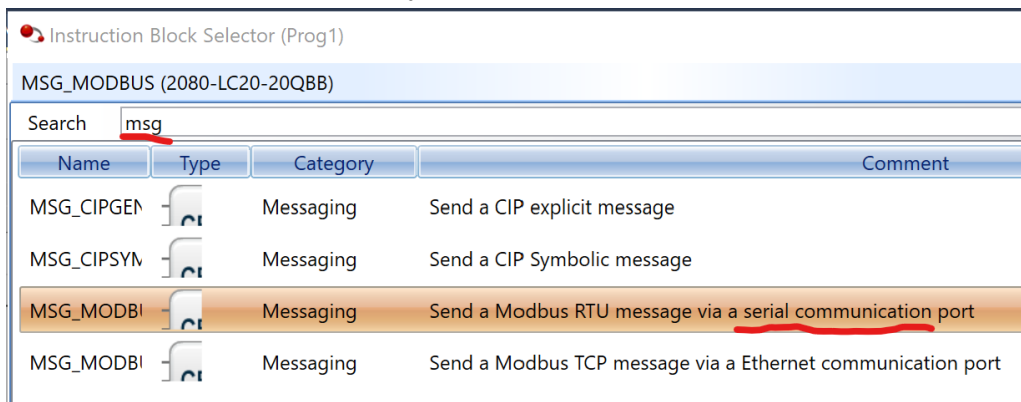


- f. Repeat for the three other discrete coil outputs. Then, "Add" the four discrete inputs and link them to the addresses 100001-100004, as shown above.

- i. You can export this mapping to a file for later use if you care to.
 - g. Save this project and Build it to make sure there are no errors.
 - h. Don't download it to the PLC (unless directed).
2. Next work on the left most PLC with IP address 192.168.0.25 which is going to be the "Master"
 - a. Create a new project and select the appropriate PLC type.
 - b. On its Controller tab, Select the Serial Port item, and configure the Driver to Modbus RTU, "Master" Role and the RS232 for Media.



- c. Create a new LD Program
 - d. Drag down an Instruction Block from the item list (or from the Toolbox). Start typing "msg" and select the option that says Modbus RTU.

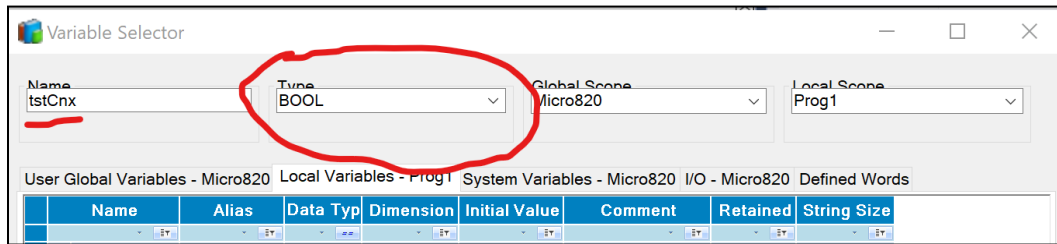


- e. Double click the block, and rename it something like "WriteCoils"
 - f. With the MB Block selected, hit the F1 key to open up the help page, read through that help document. Also open the bottom link that describes how the Local Parameters are set up.
 - g. For each of the inputs to the MB block, click on the upper half and type in a name and hit Enter

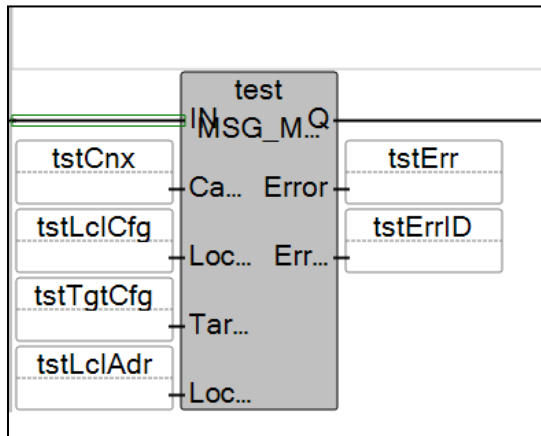


- h. To fix the yellow warning triangle, double click in the lower half of the input, which will open up the Variable Selector window. You should see the name you typed and the appropriate "type"

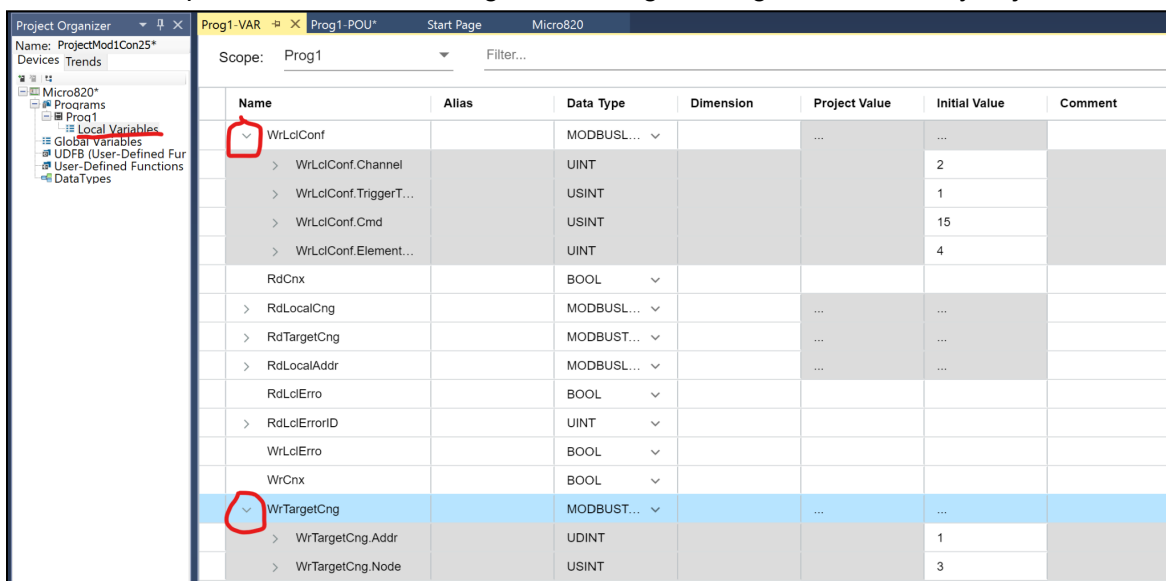
automatically selected. Hit the Ok button at the bottom, and that should bring you back to the ladder diagram and the yellow warning triangle should be gone.



- i. Repeat for all four inputs and two outputs. Try to have "Write" or something similar so you can tell it apart from the next message block we will build later.



- j. Once done, Double Click on the Local Variables entry in the tree on the left edge of the window and expand the WrLocalConfig and WrTargetConfig variables that you just created.



- k. From the top of the Local Config, under Initial Value put a "2" for .Channel. This refers to which Modbus port you want to send the message on, here the "2" is used to indicate we want to use the built in serial pins on the top front of the PLC.
- l. Next is the .TriggerType, you can start it with a "1" which means we want it to continually send this message not just when triggered.
- m. Below is the .Cmd entry and this should line up with one of the Modbus Function Codes that are described in the help, but also in the [Modbus Protocol Specifications](#). Here we want to be able to write to the other PLC coil outputs, so enter 0x0F or "15".
- n. The next entry is for .Element and this is how many coils we want to write to, here type in "4"

- o. Next under the Target Configuration, the .Addr is the address within the coil address region you want to start writing to. And since we mapped DO_01 through DO_04 to 100001-100004, enter a "1" here to start with DO_0.
 - p. Finally under the .Node enter in the "Slave" unit address from the other unit, here use "2" (the Unit Address we assigned to .27)
 - q. Build the program and download it to the PLC which is attached via the serial cable to .27 (should be .25)
 - r. When the program runs you should be able to manipulate the value of the first word in this MB Block's Local Address (~WrLclAddr[1]) memory space which should, in turn, make the lights on the other PLC go on or off.
3. Provided that worked, repeat the steps starting with 2.d and create another Modbus message block that can read in the first four Discrete Input that we mapped to DI_01 through DI_04. (FC = 0x04)
 - a. If this works you should be able to have a partner flick the switches on the other PLC and it will show up in the first word of the ReadLocalAddress block of memory on the "Master" device.
4. If that worked, add a TON block, as shown in the video, to a rung above the message blocks and change the .TriggerType to "0" and that should make the inputs and outputs a little slower to reach the other PLC. Or if you feel fancy you can trigger it with one of the inputs on the .25 Controller.

Other things to try:

- Look at the RS-232 Serial data using the oscilloscope (try to manually decode)
- Try the RS-485 connection (you will need to move the wires to the other pins on each)
- Read in the Analog Input from the other PLC (AI_01?)
- Read and write to the Holding Memory (which you will have to map in the "Slave" device)

Or this video and use the serial to communicate with your PC.

https://www.youtube.com/watch?v=Zb2C4Zsq2_s

<https://www.win-tech.com/>