Using EtherNet/IP on an Allen-Bradley SLC 5/05 for communication with Ensemble

Written by Tim Ryder of Columbia Marking Tools. 07/12/2010

Requirements:

- A SLC 5/05 Processor with FRN 10 or Higher. *(You can find out what FRN version of the Processor you have by looking at the sticker on the backside of the unit before inserting it into the rack.)*
- Ensemble controller with the EtherNet/IP plugin installed by the factory and running.

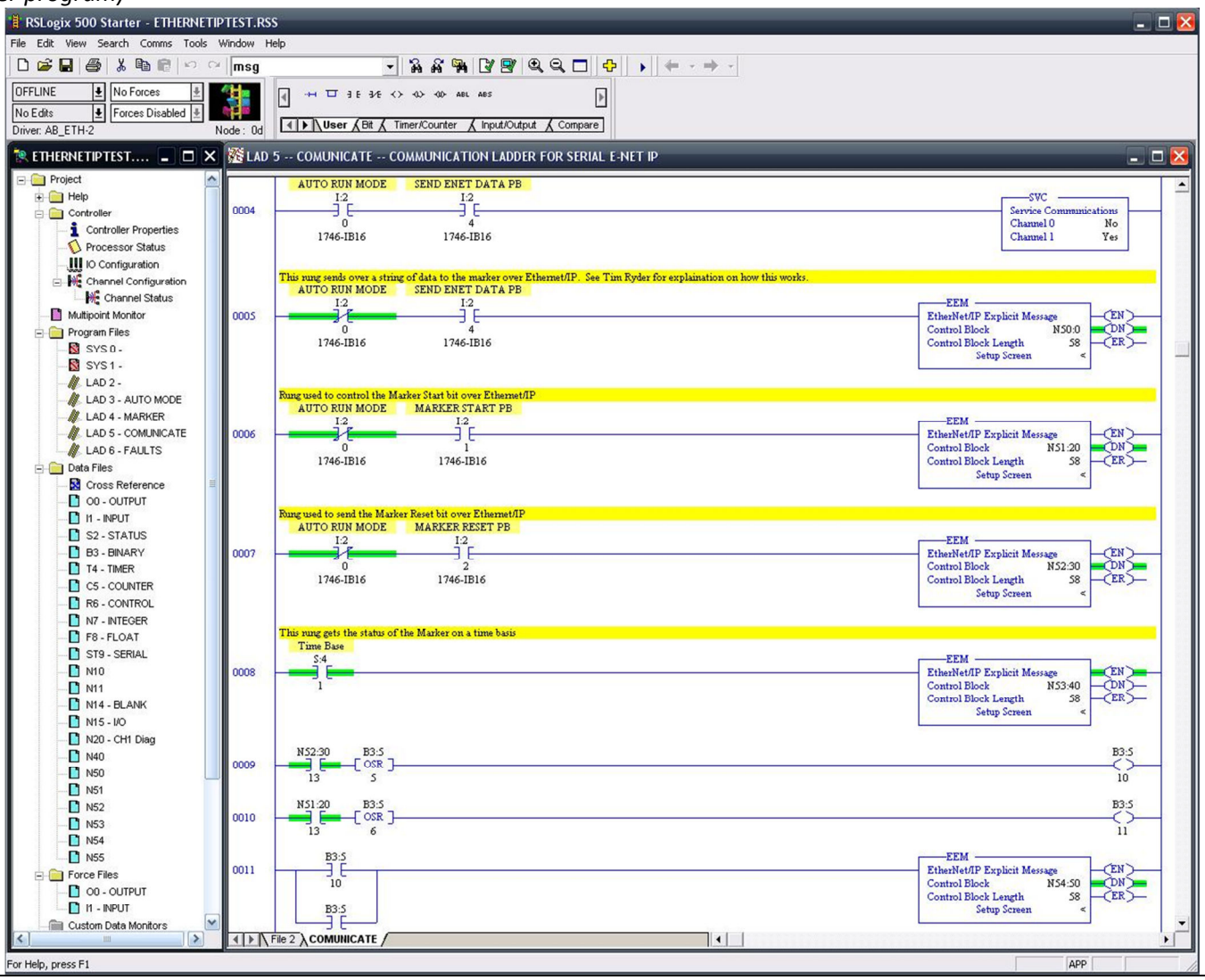
Note: If you have a SLC 5/05 but does not have the FRN 10 firmware, you can purchase an upgrade kit from your local Allen-Bradley dealer using the part number: 1747-DU501

Network Parameters:

The SLC 5/05's IP Address must be configured to be within the first 3 octets of the IP Address on the Ensemble Controller, It should also be within the same Subnet Mask unless you're accessing the Controller through a gateway to another network. *(Example: 192.168.254.XXX Subnet 255.255.255.0).*

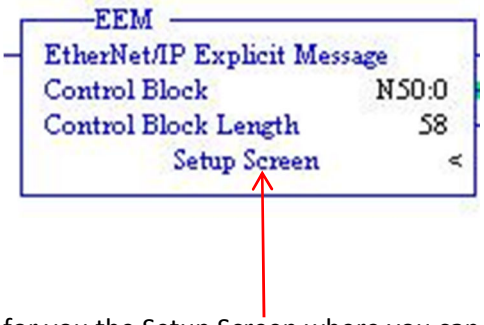
RSLogix-500 Ladder Example

Shown here is a screen shot of a working Logic communicating to an Aerotech Ensemble MP using the "EEM" Instruction found in RSLogix-500. *(Note: This instruction will not be active if you do not have the correct processor setup for your ladder program)*



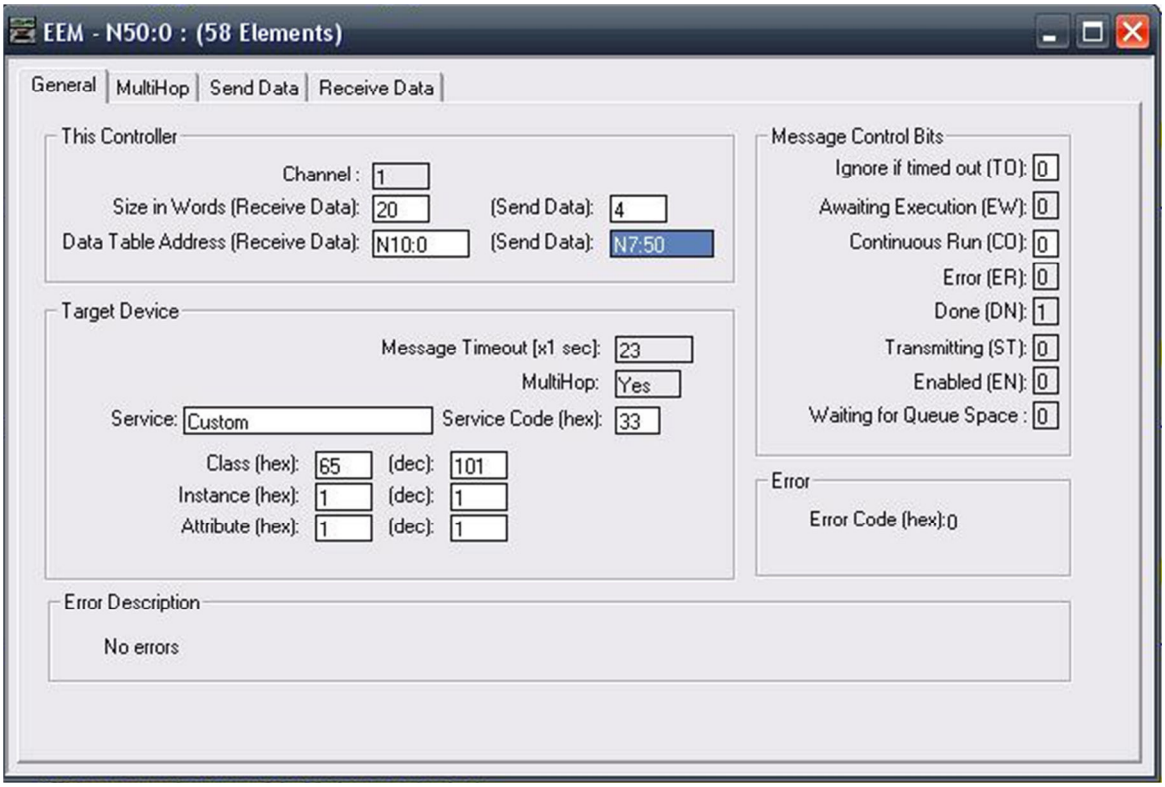
Programming the Ladder in RSLogix-500:

Because the SLC controller does not natively support the CIP protocol, Allen-Bradley has updated the RSLogix-500 software to include the "EEM" instruction. This instruction will compile a CIP type packet of data and transmit/receive it to the desired IP destination. The EEM Instruction is found in the "Input/Output" in the instructions pane of RSLogix. Drop one onto your ladder configure its' Control Block setting.



The control block must be an Integer Register Data File which can accommodate 58 registers of data. The Control Block Length field is NOT editable and will always be 58. Make sure that whichever integer register you specify it has enough available registers to accommodate without overlap. I suggest for each EEM Instruction you use that you create a new Integer Data File as to avoid confusion.

After you type in a good register address, the software will automatically open up for you the Setup Screen where you can configure the exact details of the packet to be assembled.



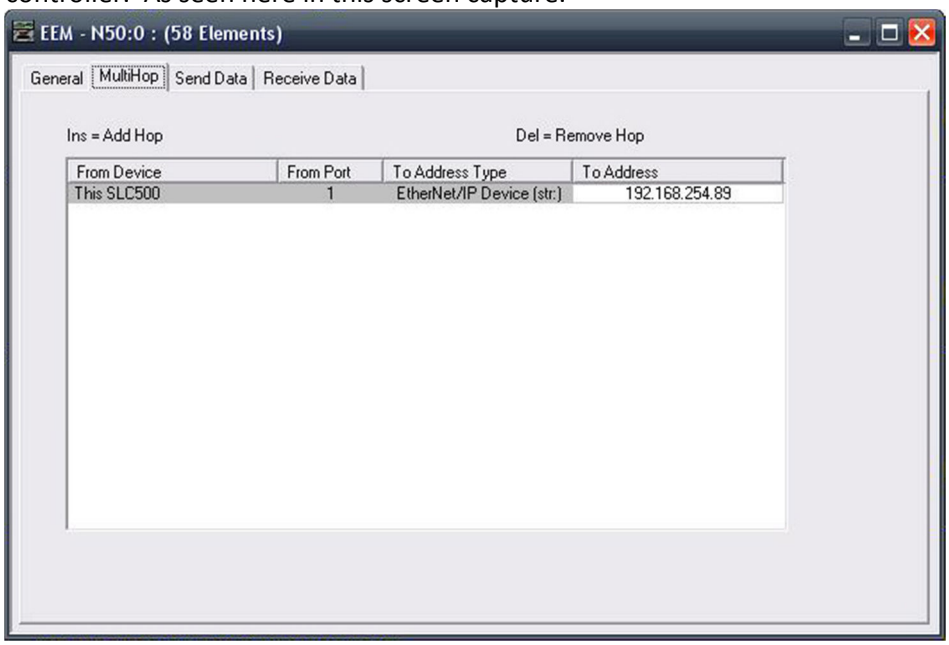
This Setup Screen gives us complete control as to what type of packet we're sending, the data to be sent, as well as where we want that packet to go in the Ensemble Register map.

The first step again would be to setup the Appropriate registers for holding the data going out and coming in from this EEM instruction. The fields here are a little vague so I will be more specific to explain them better. The First field on the General tab where it says "Size in words (Receive Data) this will map out how many words we're expecting to come back from the Ensemble in response to our command we're sending it. If you do not know how many to expect, you can always put some large arbitrary number like 50. For this setting the RSLogix-500 software will not calculate to see if the registers will overlap another instructions'. Likewise with the Data Table Address (Receive Data) I recommend using one large Integer Data File for this and all of your EEM instructions, it makes it easier to visually find the "Received Data" for all of the instructions.

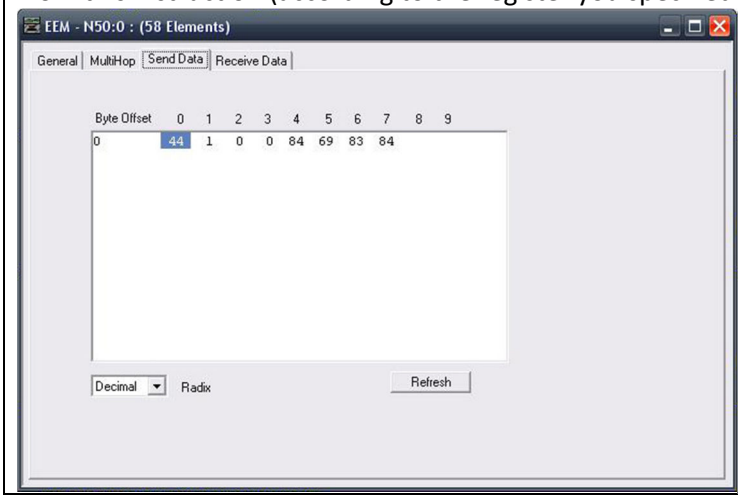
The (Send Data) field on the top is asking for how many words you want to send starting at the address below it. This is where it gets a little tricky. The SLC processor uses 16 bit words while most everything else in the modern world including the Ensemble Controllers are using 32 bit words. This creates a problem when trying to setup the data for the EEM instruction. All of the fields in this setup screen which require you to enter the amount of Words are referring to 16 bit words within the SLC memory. We get around this by using double the amount of words locally in the SLC for everything we enter.

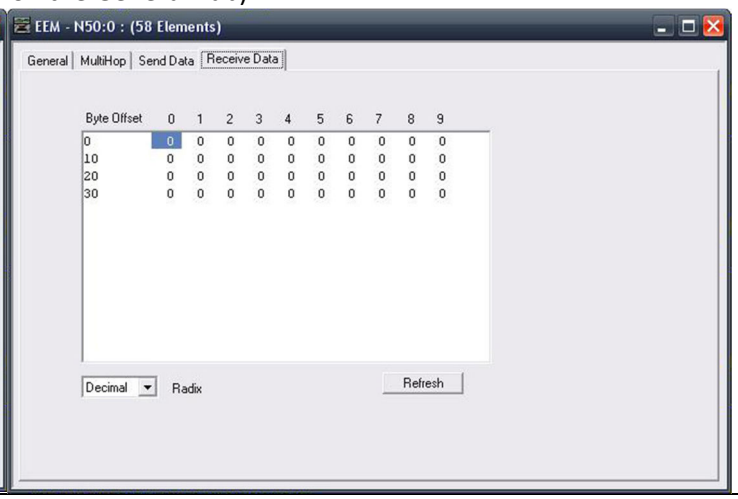
Next we click on the second tab in this Setup screen called "Multihop" and enter the IP address of the Ensemble

controller. As seen here in this screen capture.



The next 2 tabs will show you the 16 bit words within the SLC of the raw Decimal data you're "Sending" and "Receiving" from this instruction (according to the register you specified on the General Tab).

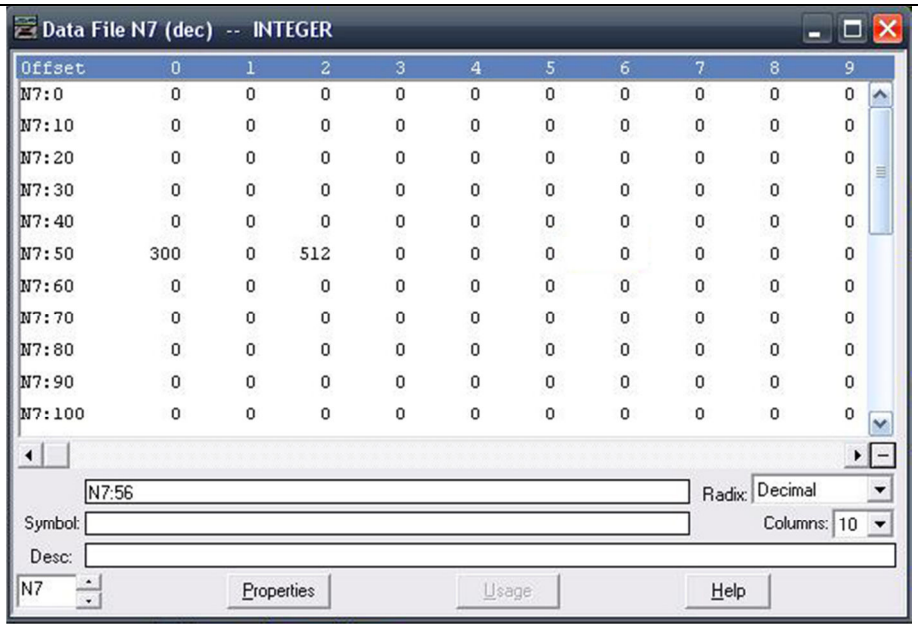




Writing to a Single Integer Register:

One example for applying the formatting rules from the previous section, the Ensemble help documents state that in order to write a value to a single integer register on the Ensemble (among other things) it requires you to send 2 words of data within the packet. The first word is the address in the Ensemble of the Register you wish to write to, the second word is the actual data you wish to populate this register with. Within the SLC processor however we actually need to send 4 words to accomplish this. The first is the address of the register you wish to write to, the second is a blank word. The Third is the data you would like to write to that register and the fourth is another blank register. Here is a screen capture of a programmed integer register data file within RSLogix-500 for writing a value to integer register in the Ensemble. For my example of the EEM Instruction on the previous page, I have setup all of the fields correctly to send the value "512" to register "300" in the Ensemble.

The Chart to the right is taken from RSLogix and shows how to compose the 4 words of data with the first 2 words as the desired integer register and the second 2 words as the value for the 32 bit register. The blank register only applies if the value you're trying to send doesn't exceed the maximum value of the 16 bit register. If you need to fill the first register with the maximum value of 32767 and add the remaining value in the 2nd register, this is ok as well. So for example if I wanted to enter the value of 56878 into register 300 on the Ensemble, I would have "32767" in N7:50/2 and then 24111 in N7:50/3 The combination of the 2 registers together would equal 1 32 bit word on the Ensemble Side totaling 56878.



Offset	0	1	2	3	4	5	6	7	8	9
N7:0	0	0	0	0	0	0	0	0	0	0
N7:10	0	0	0	0	0	0	0	0	0	0
N7:20	0	0	0	0	0	0	0	0	0	0
N7:30	0	0	0	0	0	0	0	0	0	0
N7:40	0	0	0	0	0	0	0	0	0	0
N7:50	300	0	512	0	0	0	0	0	0	0
N7:60	0	0	0	0	0	0	0	0	0	0
N7:70	0	0	0	0	0	0	0	0	0	0
N7:80	0	0	0	0	0	0	0	0	0	0
N7:90	0	0	0	0	0	0	0	0	0	0
N7:100	0	0	0	0	0	0	0	0	0	0

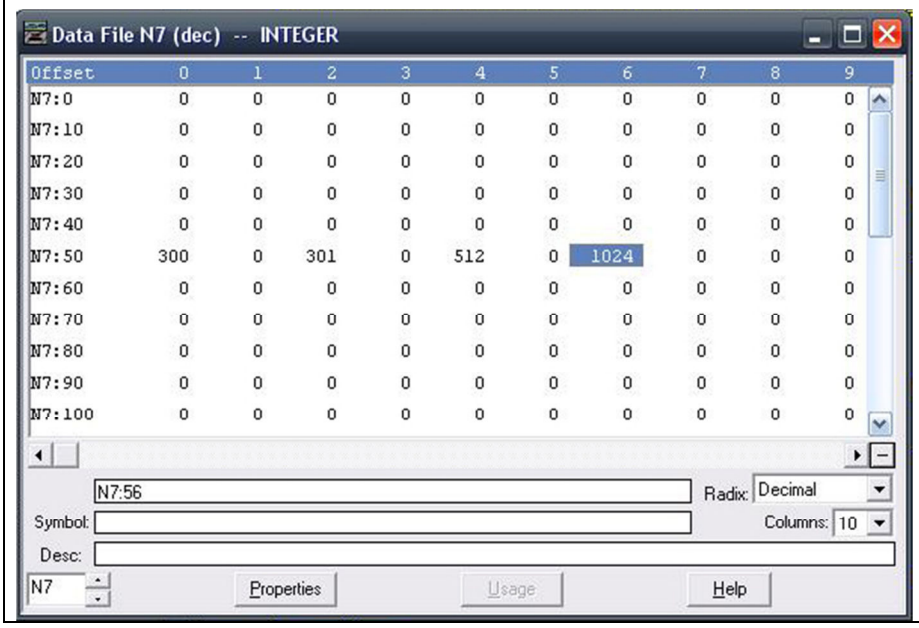
When you execute this instruction, the Ensemble will interpret the combination of the 2 16 bit registers as a single 32 bit register. Unfortunately this makes it a bit tricky on the SLC side to compose the packet data correctly especially if it is dynamic data.

Writing to Multiple Integer Registers:

Keeping in Mind of what you previously read in the Single Integer Register section, Writing to Double registers is basically the same thing. There is only hiccup which can cause some headaches. For requesting permission to write the multiple values to a range of registers with the Ensemble, Aerotech requires that we send the following words within the packet.

1.) The Starting Address of the first register to write to
2.) The Ending Address of the Last register to write to (this gives us a range for example 300-301)
3.) A word for each registers containing the data to be store within it.

As I had mentioned before this can get complicated remembering that we're actually sending double the amount of words on our end to get them correct on the Ensemble End. This gets VERY difficult writing multiple registers to the Ensemble using dynamic data for what is being across. This poses quite a headache if the size or amount of words being sent change with the change in data size…



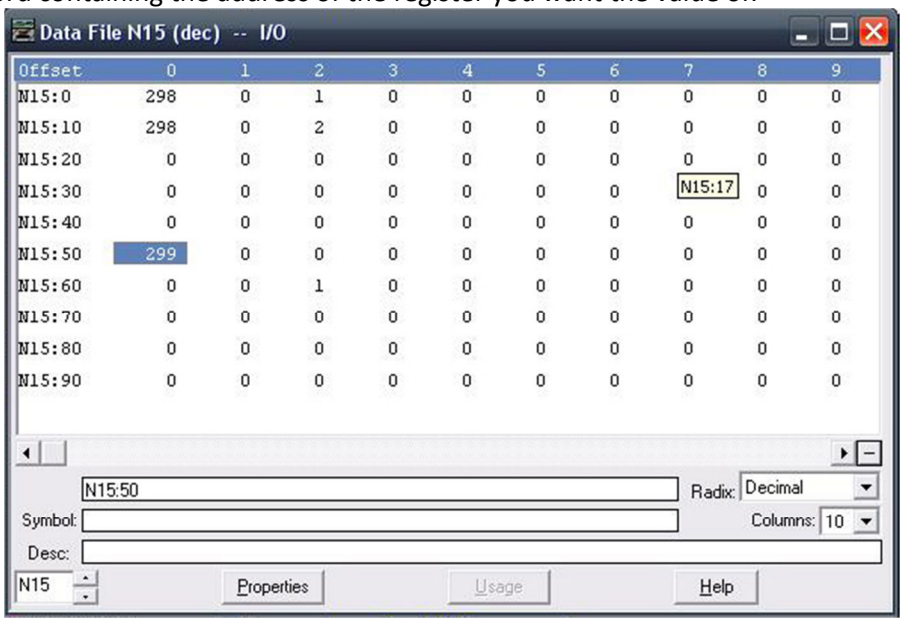
Offset	0	1	2	3	4	5	6	7	8	9
N7:0	0	0	0	0	0	0	0	0	0	0
N7:10	0	0	0	0	0	0	0	0	0	0
N7:20	0	0	0	0	0	0	0	0	0	0
N7:30	0	0	0	0	0	0	0	0	0	0
N7:40	0	0	0	0	0	0	0	0	0	0
N7:50	300	0	301	0	512	0	1024	0	0	0
N7:60	0	0	0	0	0	0	0	0	0	0
N7:70	0	0	0	0	0	0	0	0	0	0
N7:80	0	0	0	0	0	0	0	0	0	0
N7:90	0	0	0	0	0	0	0	0	0	0
N7:100	0	0	0	0	0	0	0	0	0	0

So for example, if I wanted to write the value 512 and 1024 to registers 300, 301. My data table within RSLogix-500 would look like this. Here you can see I have the first 2 16 bit words as the starting register in the ensemble, the next 2 16 bit words as the ending register in the ensemble and the next 4 words as the actual data to populate the range of registers I am allocating to write to.

This gives us a total of 8 16 bit words to be sent from the SLC while again the Ensemble interprets it as 4 32 bit words. You can image how more than this can get pretty hairy to program☺.

Reading a Single Register:

This little example is the easiest thing to do once you've already figured out the previous ones. Aerotech makes it very simple in requiring you to only send 1 32 bit word containing the address of the register you want the value of.



Offset	0	1	2	3	4	5	6	7	8	9
N15:0	298	0	1	0	0	0	0	0	0	0
N15:10	298	0	2	0	0	0	0	0	0	0
N15:20	0	0	0	0	0	0	0	0	0	0
N15:30	0	0	0	0	0	0	0	N15:17	0	0
N15:40	0	0	0	0	0	0	0	0	0	0
N15:50	299	0	0	0	0	0	0	0	0	0
N15:60	0	0	1	0	0	0	0	0	0	0
N15:70	0	0	0	0	0	0	0	0	0	0
N15:80	0	0	0	0	0	0	0	0	0	0
N15:90	0	0	0	0	0	0	0	0	0	0

So like in my previous examples, I'm going to use an EEM message but all I am sending is 2 16 bit words shown from this data table to the right here where I highlighted my register N15:50 containing the register address I want to know the value of in the Ensemble. The next row below it I have assigned to be the received data address. For this packet the Ensemble will respond with 2 32bit words, the first is any errors which might occur as you can see in N15:60 there were none and the second is the data within the requested register. Which is a nice and simple value of 1. This instruction happens very quickly over EtherNet/IP and can usually be execute every 50ms or so. From a logic point of view I would recommend setting up a special XIC bit with the address S:4/3 which is a free running timer within the SLC Processor and it will go true every 64ms. This is a simple way to constantly monitor the value of a register if you're using it for setting conditions within your logic.