

EtherNet/IP™ Programming

#34 Programming with RSLogix 500

7/3/2014
Aerotech Inc.
www.aerotech.com

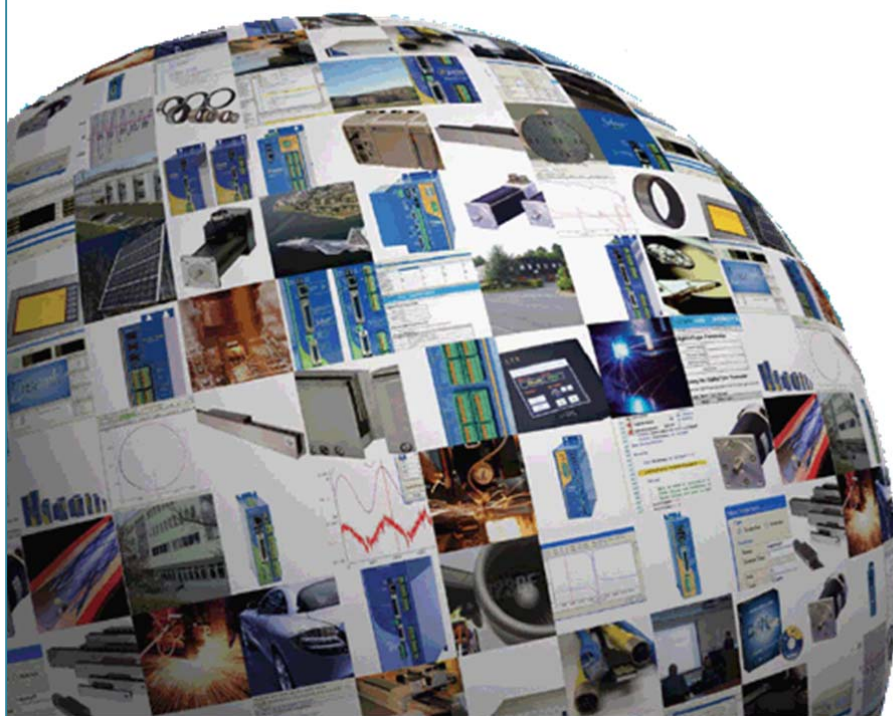


TABLE OF CONTENTS

ETHERNET/IP™	2
ETHERNET/IP PROGRAMMING RSLOGIX 500	2
RSLogix 500 Ladder example	3
Programming the ladder	3
Writing to a single Integer Register	5
Writing to Multiple Integer Registers.....	6
Reading a Single Register.....	7

ETHERNET/IP™

EtherNet/IP™ (Ethernet Industrial Protocol) is a communications protocol developed by Rockwell Automation, and is currently managed by the Open DeviceNet Vendors Association (ODVA). It is designed for use in industrial automation applications.

The Aerotech Ensemble and Soloist controllers have an EtherNet/IP™ plugin that can be installed. These controllers have 4 tasks. This plugin runs on Task 3 of the controller. It also takes up the Task 4 slot as well for communications. This is not inherent in the interface. This leaves Task 1 and Task 2 for user programs.

ODVA requires that users be registered. By doing this a vendor ID is created that can be read over an EtherNet/IP™ program to verify that the component is licensed. The Aerotech Soloist and Ensemble are ODVA certified. The Aerotech EtherNet/IP™ Vendor ID issued by ODVA is 935 for these controllers.

ETHERNET/IP PROGRAMMING RSLOGIX 500

Often machine or line control requires both a separate PLC and motion controller. The Ensemble, a multi-axis coordinated motion controller, and the Soloist, a single axis motion controller, can be used to interface with an Allen-Bradley SLC 5/05 using standard EtherNet/IP™. This application note demonstrates how to setup the SLC 5/05 and the Ensemble in a Master/Slave configuration. The Ensemble and Soloist are EtherNet/IP™ certified. The Soloist would run identical, and be programmed identical on the Allen Bradley side as the Ensemble. The only difference would be in the Soloist command syntax, which is slightly different than the Ensemble.

If you are using an Allen-Bradley Logix (and RS Logix 5000), refer to the application note “Using Ethernet/IP™ with RS Logix 5000”.



**When the PLC module is in PROG mode the PLC is no longer running.
Ensure that conditions are safe before you put your module into this mode.**

In this demonstration we will go through the configuration procedure to send commands to the Ensemble over EtherNet/IP™ through the RSLOGIX 500 software.

This example uses a SLC 5/05 Processor with FRN 10 or higher and an Ensemble controller with the EtherNet/IP™ plugin installed and running.

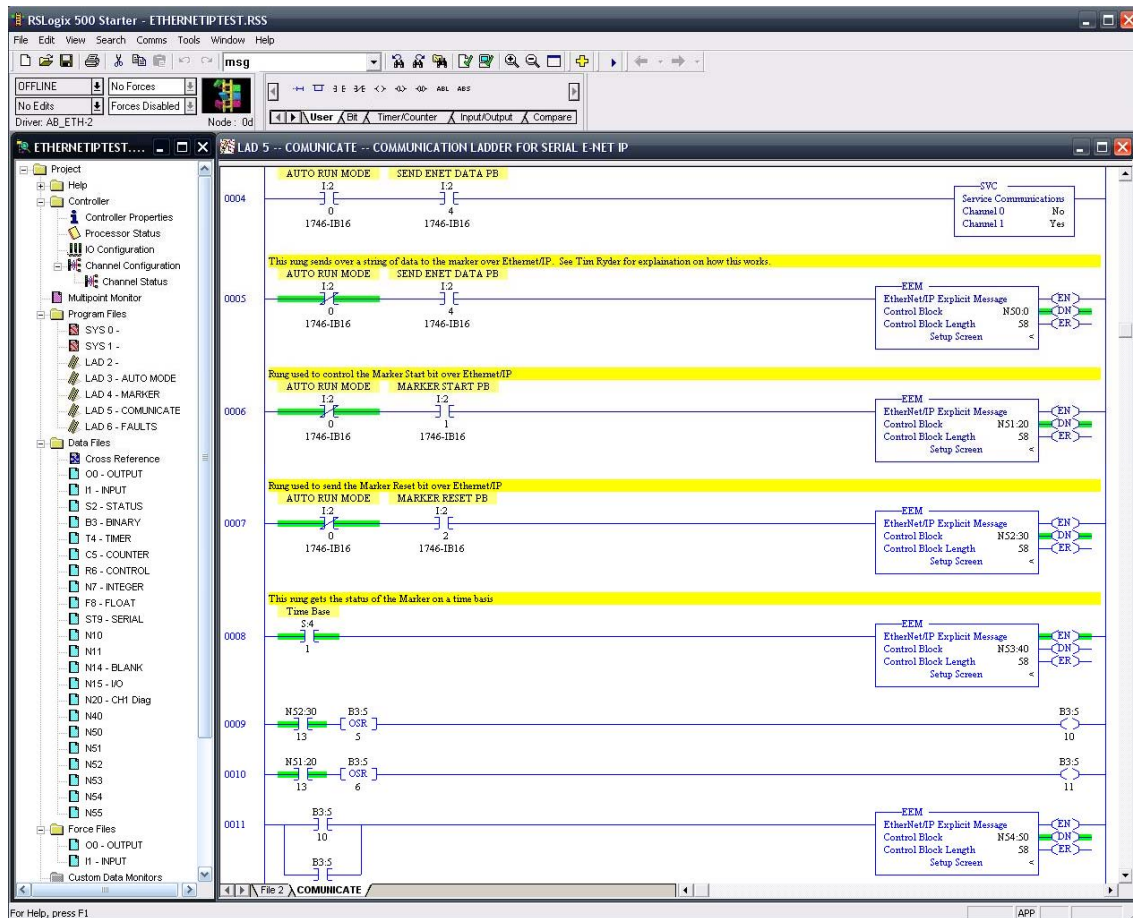
The SLC 5/05 IP address must be configured to be within the first 3 octets of the IP address on the Ensemble, and should have the same subnet mask of the Ensemble.

Example:

SLC 5/05	IP address: 192.168.254.200	Subnet mask: 255.255.255.0
Ensemble	IP address: 192.168.254.100	Subnet mask: 255.255.255.

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 the correct processor is not setup for the ladder program, see below for more information.)



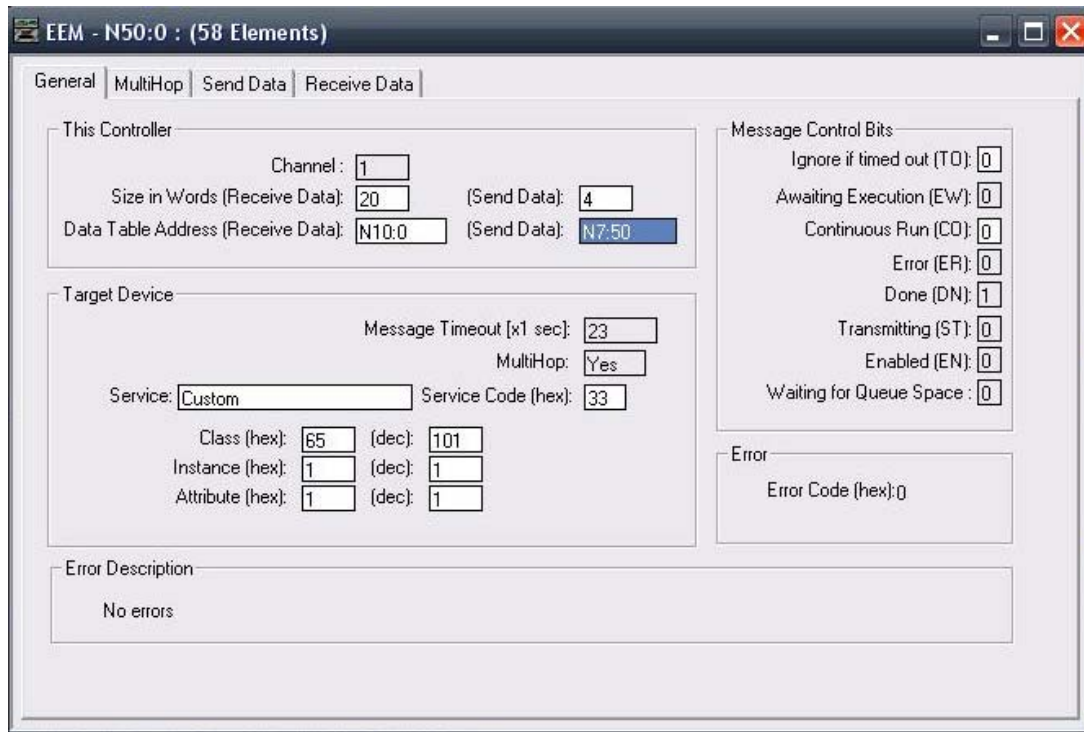
PROGRAMMING THE LADDER

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 the ladder and 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 is specified it has enough available registers to accommodate without overlap. To avoid confusion it is suggested that for each EEM Instruction used a new Integer Data File is created.

After a valid register address is entered the software will automatically open up the Setup Screen where the exact details of the packet to be assembled can be entered.



The Setup Screen gives complete control of the type of packet being sent, the data to be sent, and the location of where that packet will go in the Ensemble Register map.

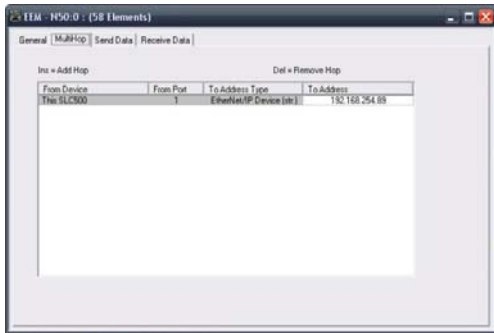
The first step is to setup the appropriate registers for holding the data going out and coming in from this EEM instruction.

Size in Words (Receive Data): This will map out how many words are expected to come back from the Ensemble in response to the command being sent, if the number of words that will be returned is not known use a large arbitrary number, i.e. 50. For this setting the RSLogix-500 software **will not** calculate to see if the registers will overlap another instructions’.

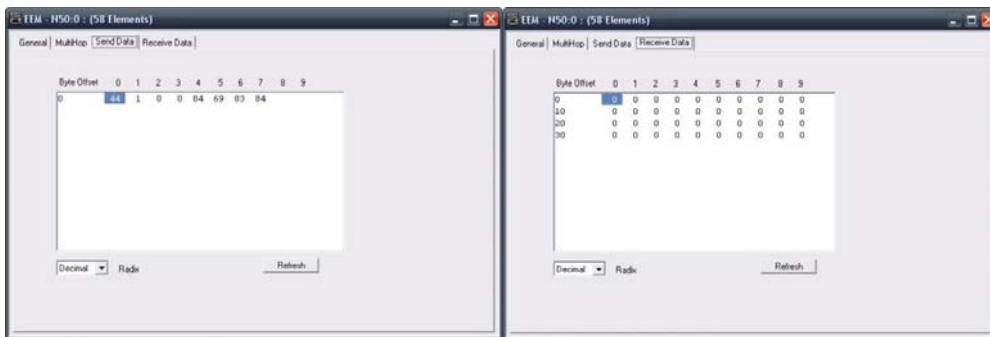
Data Table Address (Receive Data): Use one large Integer Data File for this and all EEM instructions, it makes it easier to visually find the “Received Data” for all the instructions.

Send Data: The top Send Data asks for how many words you want to send starting at the bottom Send Data address below. The SLC processor uses 16 bit words while most everything else including the Ensemble Controller uses 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. This issue is solved by using double the amount of words locally in the SLC for everything entered.

Click on the second tab in the Setup screen called “Multihop” and enter the IP address of the Ensemble controller.



The Send Data and Receive Data tabs will show the 16 bit words within the SLC of the raw Decimal data that is being “Sent” and “Received” from this instruction (according to the register specified on the General Tab).

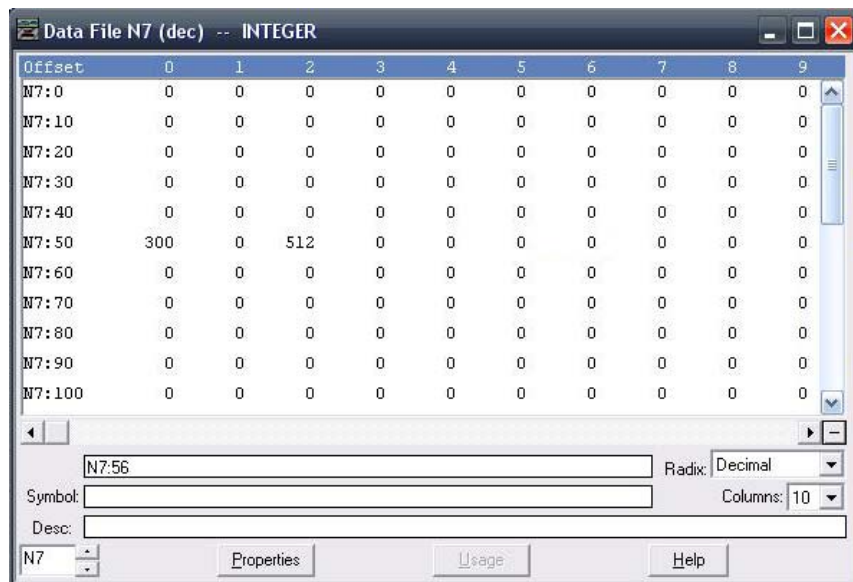


WRITING TO A SINGLE INTEGER REGISTER

The Ensemble help file states that in order to write a value to a single integer register on the Ensemble it requires 2 words of data to be sent 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 **4** words are needed to be sent to accomplish this. The first is the address of the register to be written, the second is a blank word. The third is the data to be written 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 an integer register in the



Ensemble. As referenced from the EEM Instruction on the previous page, all of the fields have been setup correctly to send the value “512” to register “300” in the Ensemble.

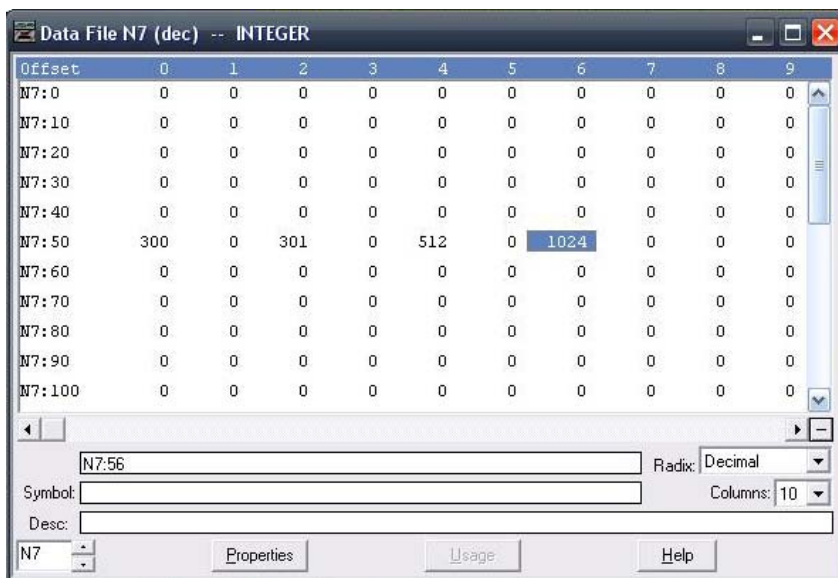
The Chart to the right is taken from RSLogix and shows how to compose the 4 words of data with the first 2 words as the desired integer register and the second 2 words as the value for the 32 bit register. The blank register only applies if the value being sent doesn't exceed the maximum value of the 16 bit register. If the value being written exceeds the value of 32767 then fill the first register with the maximum value of 32767 and add the remaining value in the 2nd register. For example if the value of 56878 needed entered into register 300 on the Ensemble, the first word in N7:50/2 would have a value of 32767 and then 24111 would be written in N7:50/3. The combination of the two registers together would equal one 32 bit word on the Ensemble Side totaling 56878.

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

WRITING TO MULTIPLE INTEGER REGISTERS

Keeping in mind the Single Integer Register procedure, writing to double registers is very similar. One case needs to be accounted for to do this. For requesting permission to write the multiple values to a range of registers within the Ensemble, Aerotech requires that the following words within the packet are sent.

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



Remember that double the amount of words are being sent on the PLC 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 sent across. This causes complication if the

size or amount of words that are being sent changes with the change in data size.

This example shows the values 512 and 1024 being written to registers 300, 301. The data table within RSLogix-500 would look like this.

The first two 16 bit words are the starting register in the Ensemble; the next two 16 bit words are the ending register in the Ensemble. The next four words are the actual data being sent to the above set of registers.

This gives a total of eight 16 bit words being sent from the SLC to the Ensemble, which interprets it as four 32 bit words.

READING A SINGLE REGISTER

When reading a register the EEM message is still used. Two 16 bit words will be sent. The data table to the right has N15:50 highlighted which contains the register address being requested from the Ensemble. The next row below it is assigned to be the received data address. For this packet the

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

Ensemble will respond with two 32bit words, the first word contains any errors which might occur and the second is the data within the requested register. In this example there were no errors and the value of register 299 was 1.