ID: QA18224 | Access Levels: Everyone

Logix and Micro820/850/870 controller Ethernet/IP Messaging

Document ID	QA18224
Published Date	09/23/2022

Summary

Logix and Micro820/850/870 controller Ethernet/IP Messaging

Question

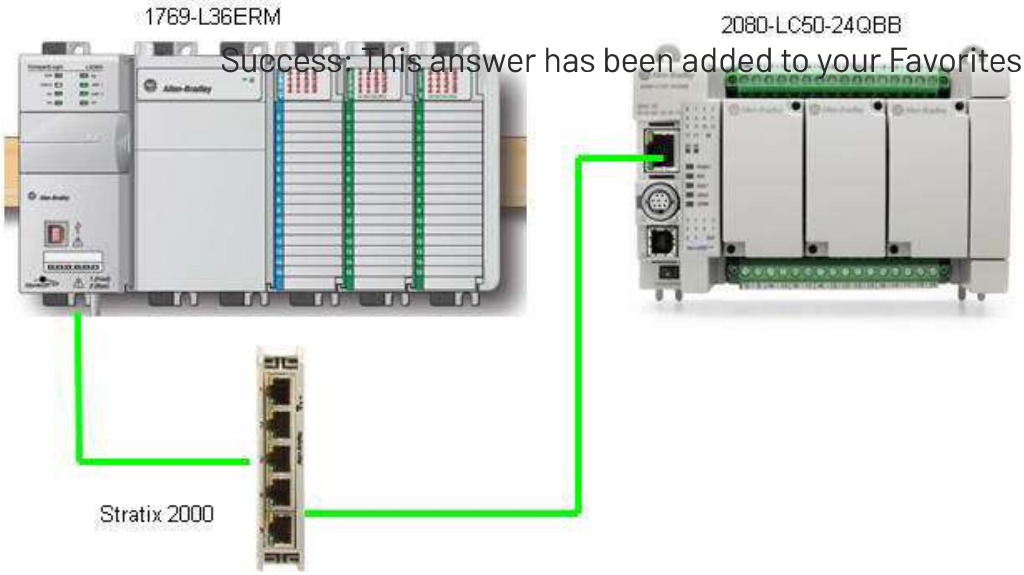
- How does a Logix controller communicate with a Micro820/850 controller over Ethernet/IP?
- Why does a message instruction in ControlLogix to a Micro820/850 have error `16#0005 Class or instance not supported?`
- How to message from Micro800 controller to Logix controller?

Background

This application note details how to create a message from a Logix controller to a Micro850 PLC (global variables only) over Ethernet/IP. If the Micro820/850 tags are not globally defined an error `16#0005 Class or instance not supported` will occur on the message instruction.

Note: Connected Components Workbench 4.0 and above support this.

This application note is based on the following hardware:



Note: It is assumed that the user is familiar with the RSLogix 5000 and CCW programming software, as many of the steps for setup are not shown.

Environment

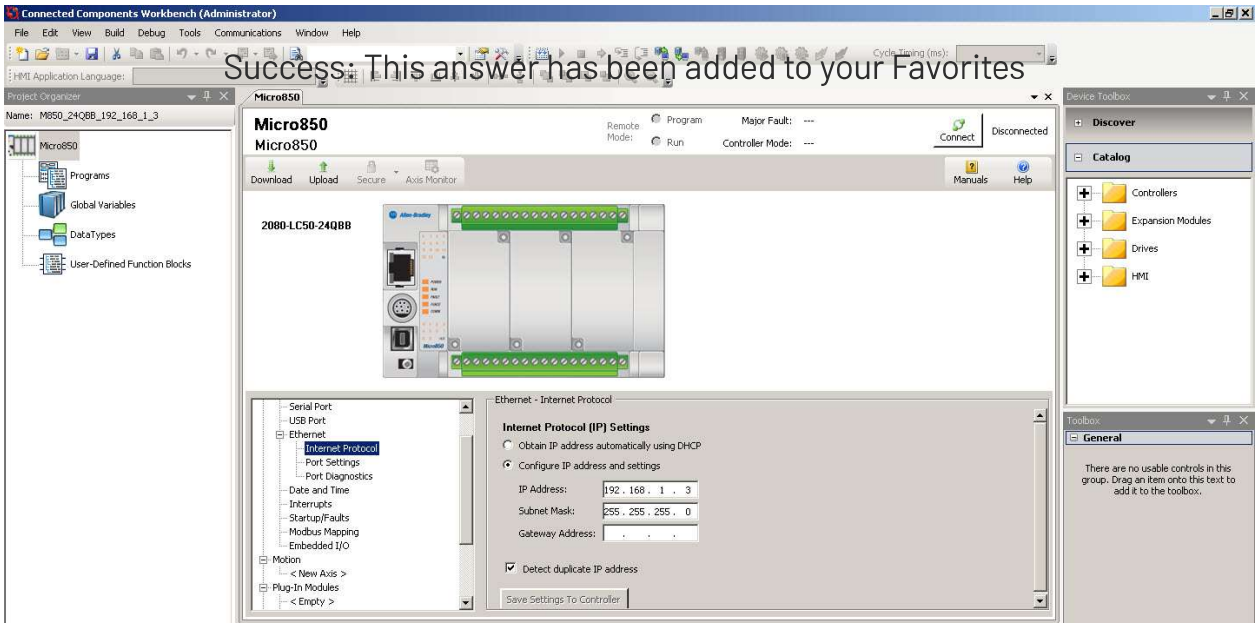
- Micro820 Controller
- Micro850 Controller
- Logix5000 Controllers
- Connected Component Workbench
- RSLogix 5000
- Studio 5000 Logix Designer

Answer

Logix to Micro800 Messaging

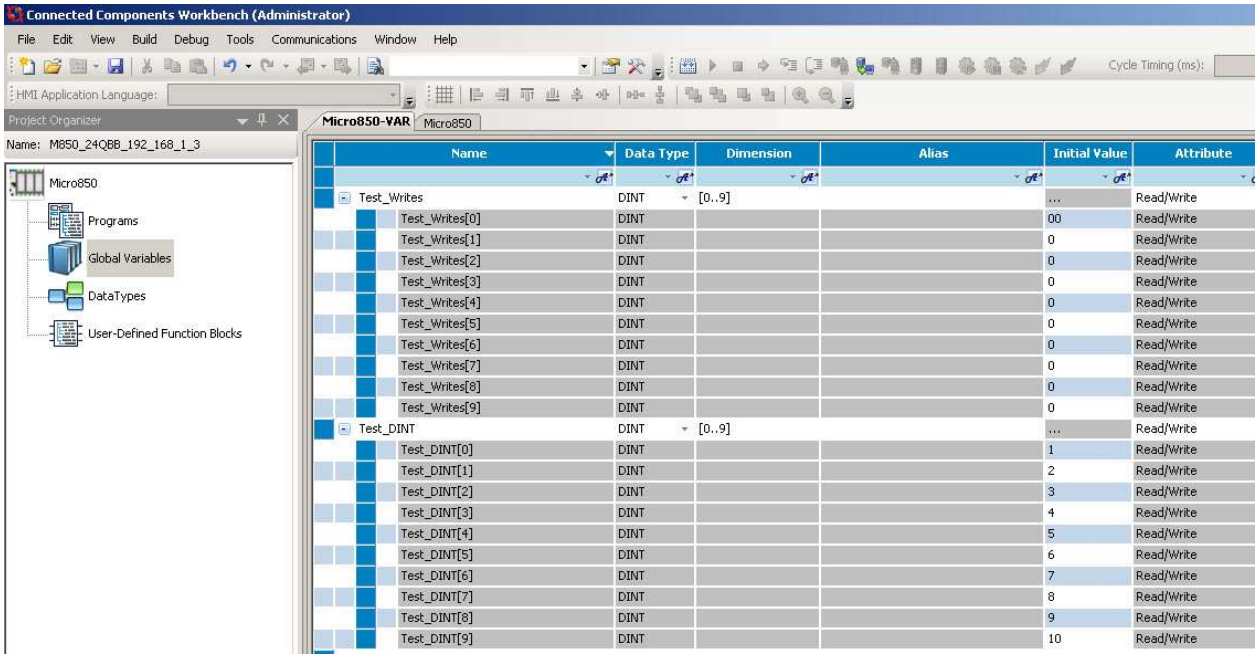
In order to communicate with a Micro850 over Ethernet/IP it is necessary to setup the Micro850 with an IP address in the same address range as the CompactLogix.

In this case as our CompactLogix is setup 192.168.1.12, then we need to setup the Micro850 perhaps to 192.168.1.3 with subnet 255.255.255.0.

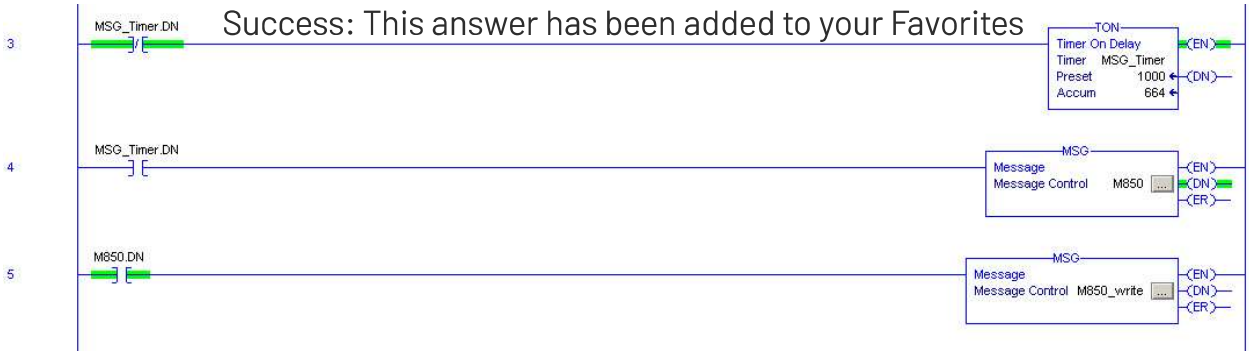


To test the messaging capability, first we need to setup some test variables in the Micro850. Enter two arrays of ten DINT's into the global variables of the M850. So create `Test_DINT` for the read, and create `Test_Writes` for the writing of values from the Logix. If the Micro 850 tags are not globally defined an error `16#0005 Class or instance not supported` will occur on the message instruction.

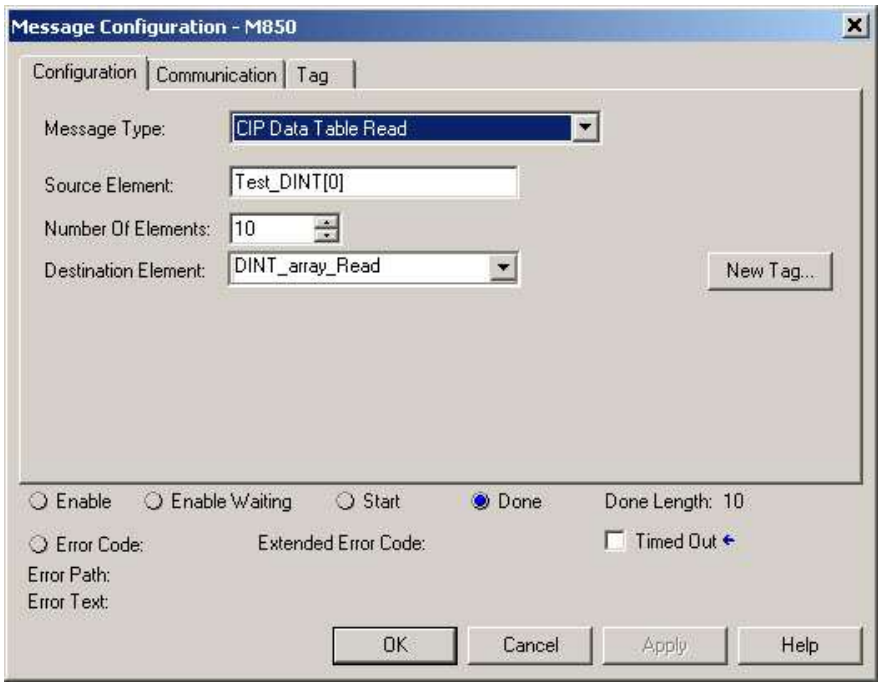
Note: that each value of the Test_DINT is provided with an initial value. Now build, download, and go to Debug mode so that you can see the global variables.



Go to the RSLogix 5000 software and create a project with the following ladder:
Create a new tag for the MSG instructions called `M850` and `M850_write`.

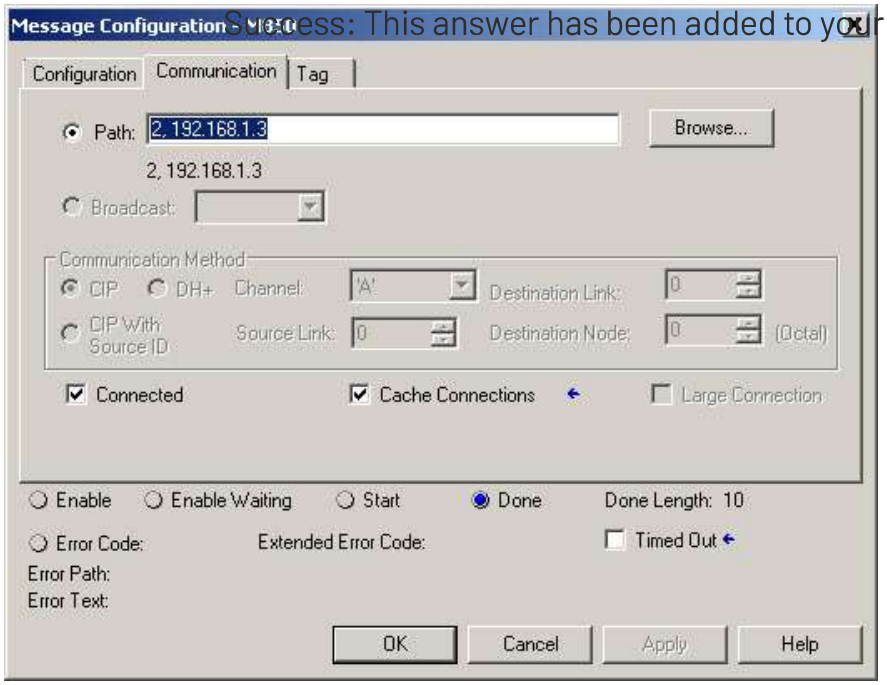


Double click on the properties of the MSG instruction M850 for the Configuration tab. Select a **CIP Data Table Read** and then enter into the **Source Element**, the name of the tag created in the M850. Specify the **Number of Elements**, in this case 10. Then enter the Destination Element where you will send the values. As we have an array of 10 values, so you will need to create a controller scoped array, in this case `DINT_array_Read[10].`



Now go to the **Communication** tab, to set the **Path**. The message path will depend on the family of controller you are using, the Ethernet port = 2, and then the IP address of the M850 = 192.168.1.3.

Note: For a Micro 820 you may have to use an unconnected message. To do this go to the **Communication** tab and uncheck **Connected.** This was tested in v29 and v30 of Studio 5000.



Double click on the properties of the MSG instruction M850_write for the Configuration tab. Select a **CIP Data Table Write**.

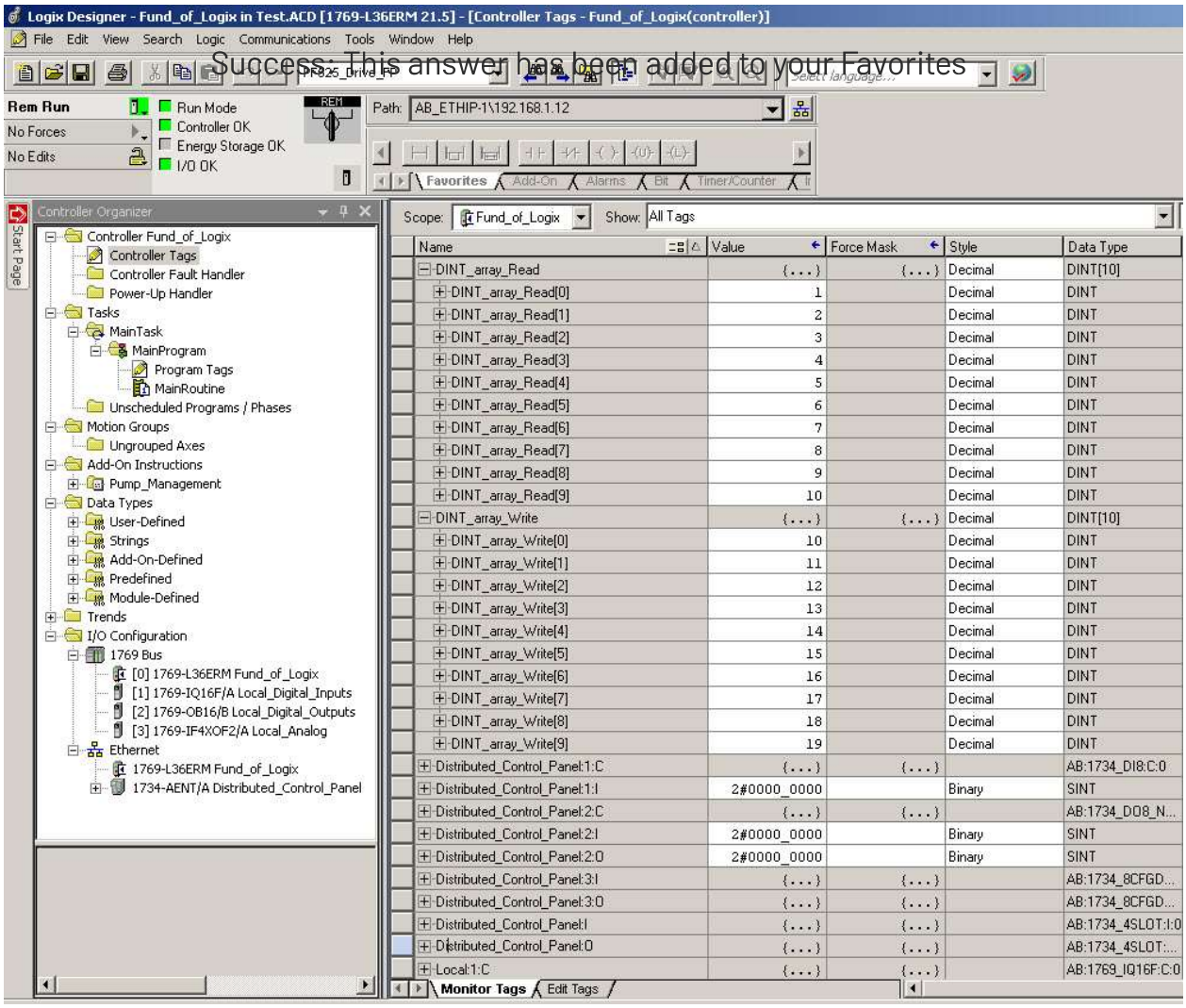
Now create a controller tag in the Logix to hold our write values `DINT_array_Write[10].` and then enter this into the **Source Element**. Specify the **Number of Elements**, in this case 10. Then enter the Destination Element which is the name of the tag created in the M850 where you will send the values, `Test_Writes`

As before go to the **Communication** tab, to set the **Path**. The Ethernet port = 2, and then the IP address of the M850 = 192.168.1.3

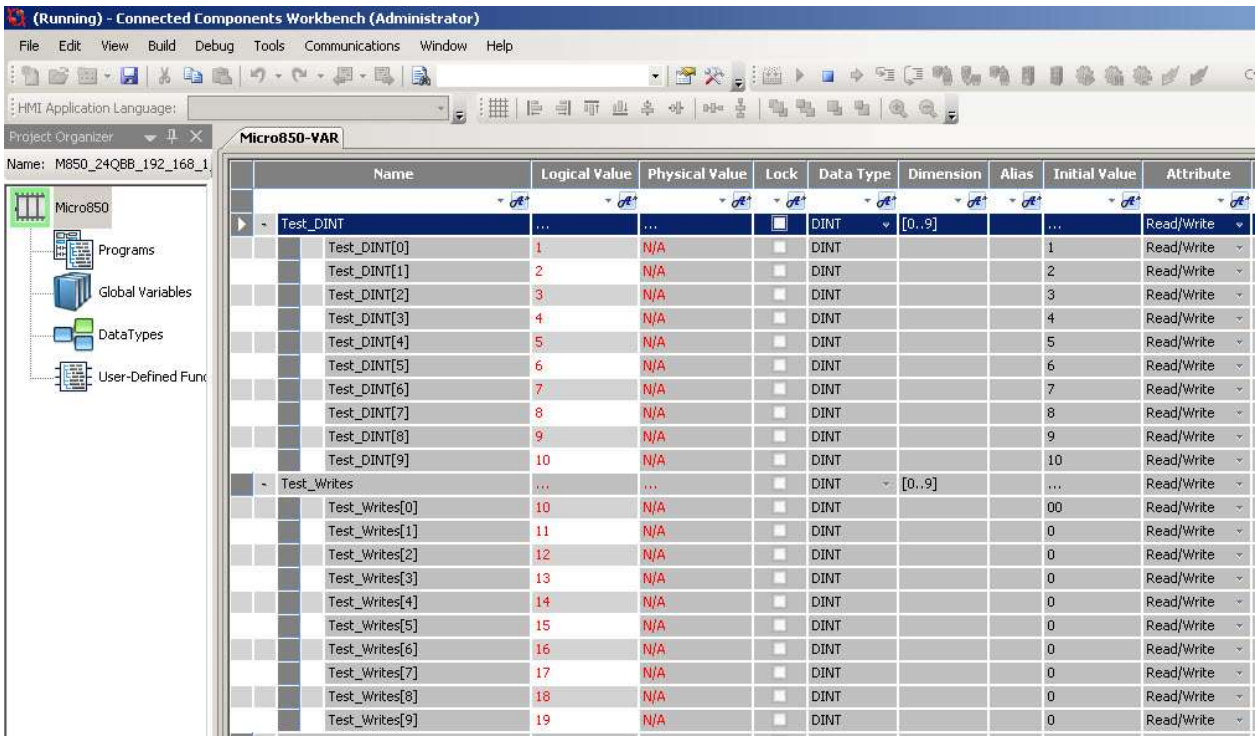
Verify your controller has no errors, download and go online.

Go to the Controller Tags in the Monitor tab, and expand your tags DINT_array_Read and to see the actual values.

Expand the tags DINT_array_Write and enter values to send to the M850.



The write values sent to the M850 are shown below in debug mode.

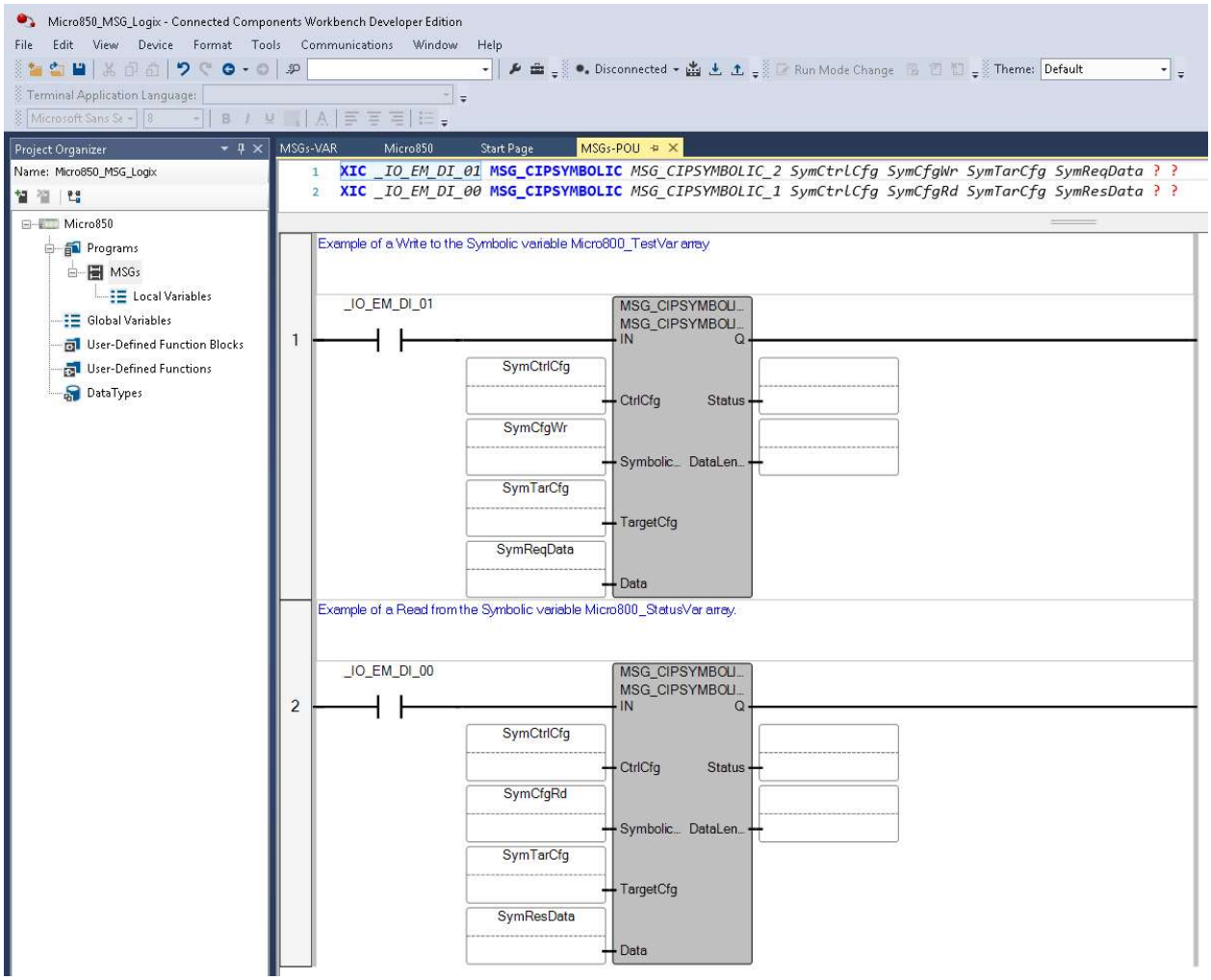


NOTE: It is not possible to read or write to the Micro800 System Variables (_SYSVAR). The same stands for User Defined Data Types created on Micro800 unit.

Micro800 to Logix Messaging (available from CCW4).

Using the same setup on our Micro850 as before with IP address set to 192.168.1.3 with subnet 255.255.255.0. The CompactLogix used in this update, is a 1769-L18ERM at IP address 192.168.1.10 with subnet 255.255.255.0.

Note that CCW 11.0 is being used in the following pictures, so the layout will look different.
The program shows 2 off Symbolic MSG instructions, the first for writing values and the second for reading values.

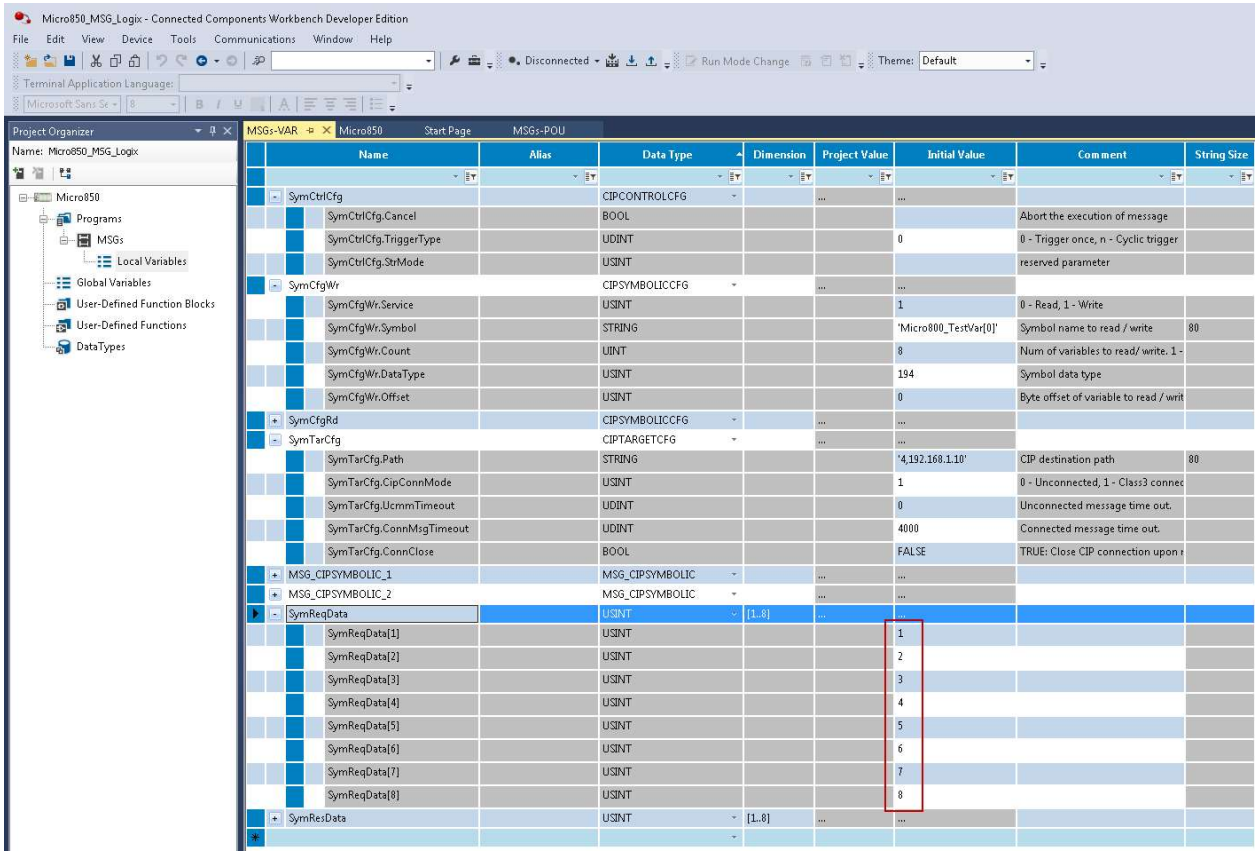


The variables for the Write instruction are shown below:

- **SymCfgWr**
  - .**Server** set to 1 for **Write**.
  - **.Symbol** set to the exact tag name you will be writing to. In this case `Micro800_TestVar[0]`.

- **.Count** set to the length or number of elements to be sent. In this case `8`.
- **.DataType** set to the corresponding value from the helpfiles referenced below. In this case `194`.
- **.Offset** set to the starting offset (in bytes) of the read variable. In this case `0`.

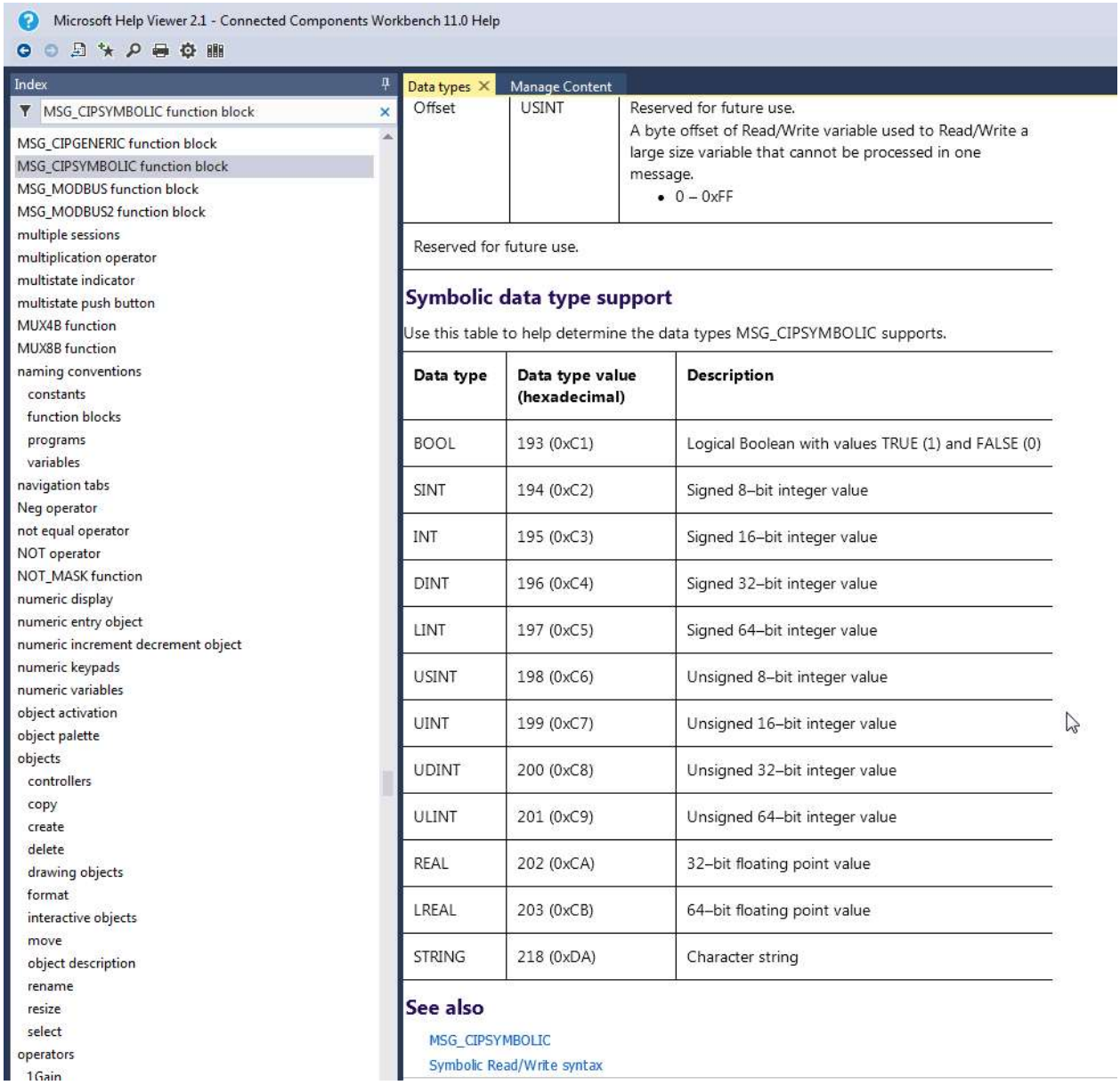
- **SymTarCfg**
  - **.Path** set to the path from the Micro800 to the controller. In our example to the 1769-L36ERM our path is `4,192.168.1.10`. For older CompactLogix controllers (for example 1769-L35E), path after IP address should point to backplane and slot 0 like `4,192.168.1.10,1,0`
  - **.CipConnMode** set to **0** for unconnected or **1** for class 3 connected messaging. In this case `1`.
  - **.UcmmTimeout** set to the timeout value for an unconnected message (**.CipConnMode 0**). In this case, since we are using connected message, `0`.
  - **.ConnMsgTimeout** set to the timout value for a connected message (**.CipConnMode 1**). In this case `4000` (4 seconds).
  - **.ConnClose** set to whether the connection should be closed after completion. In this case **FALSE** as we will not need the connection closed.



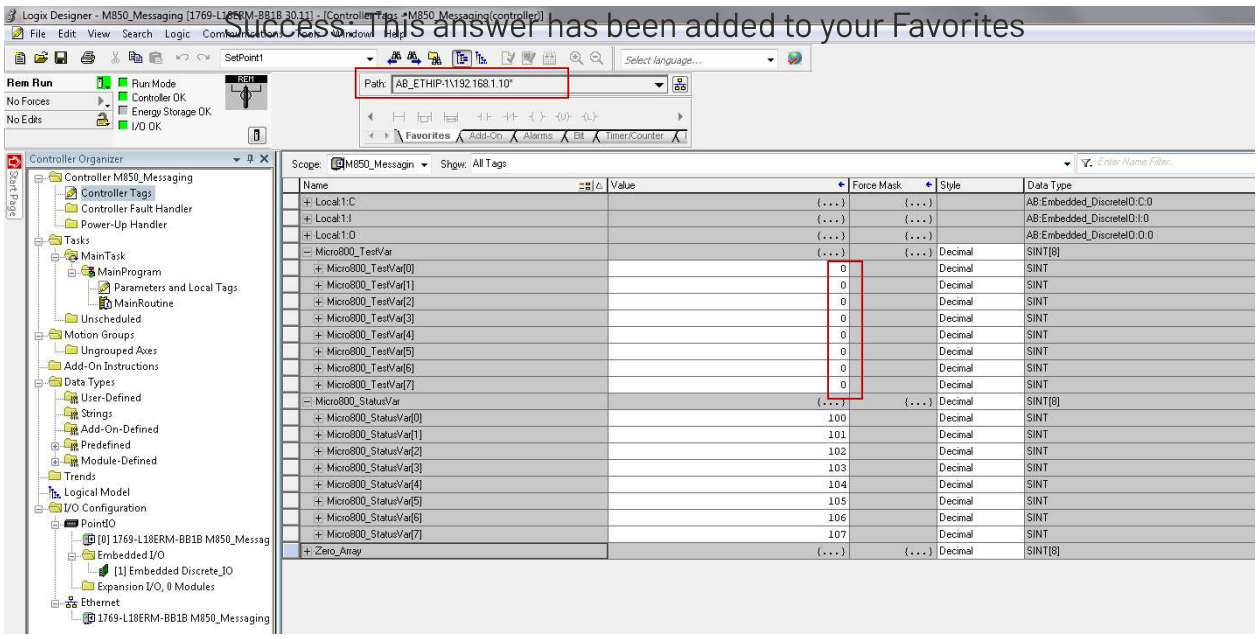
Note: the SymReqData array initial values to be sent to the Logix controller.

The DataType comes from the CCW Help files:

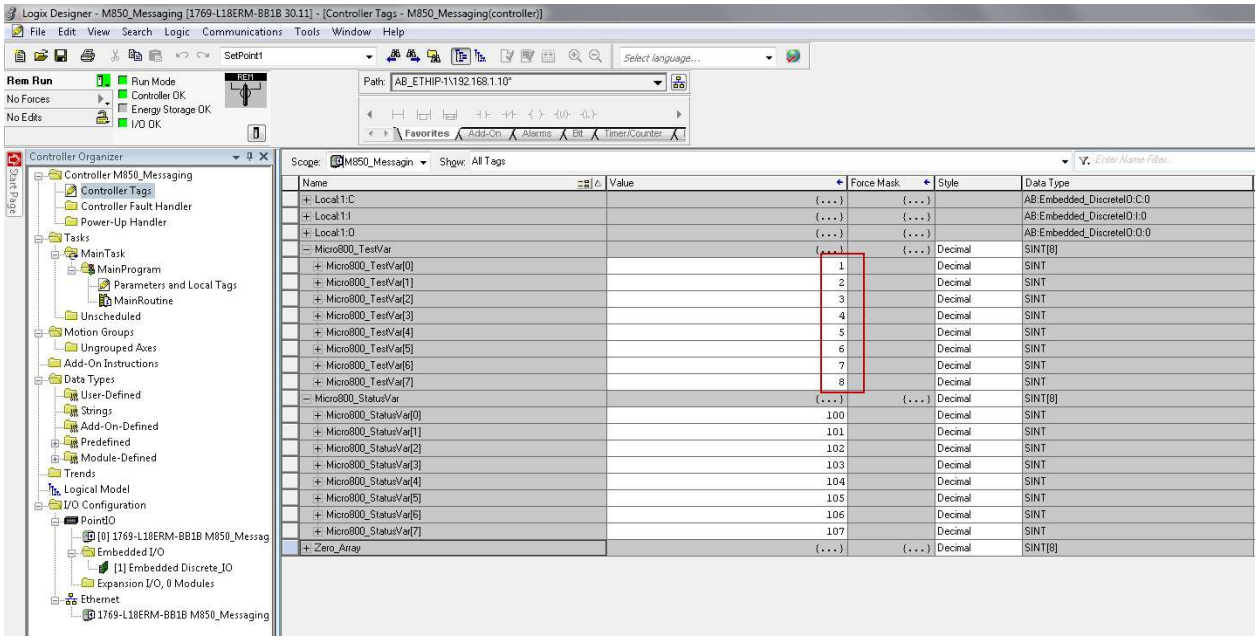
Success: This answer has been added to your Favorites



Download the program, and connect via Ethernet to the Logix controller: The Micro850_TestVar array tags show zeros.



Toggling the input **_IO_EM_DI_01**, enables the Write MSG instruction, and provides the following change to the array values:



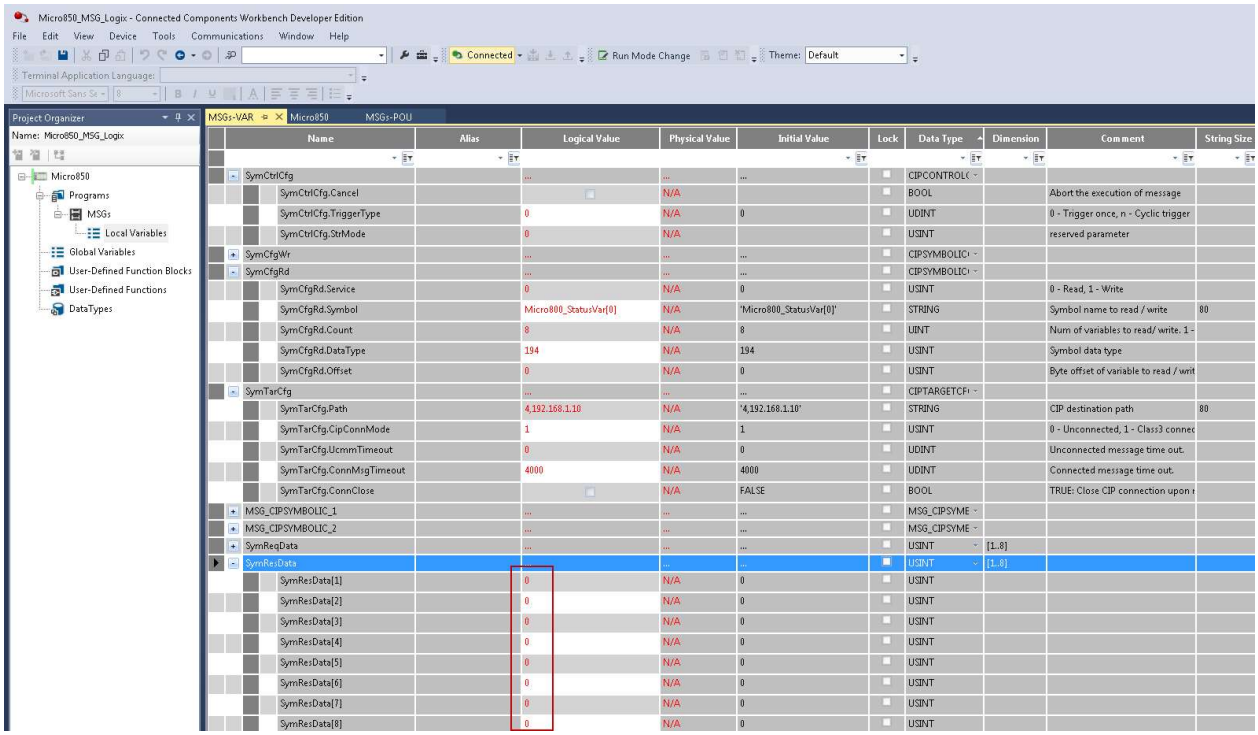
The variables for the Read instruction are shown below:

- **SymCfgRd**
  - .**Server** set to 0 for **Read**.
  - **.Symbol** set to the exact tag name you will be writing to. In this case `Micro800_StatusVar[0]`.
  - **.Count** set to the length or number of elements to be sent. In this case `8`.

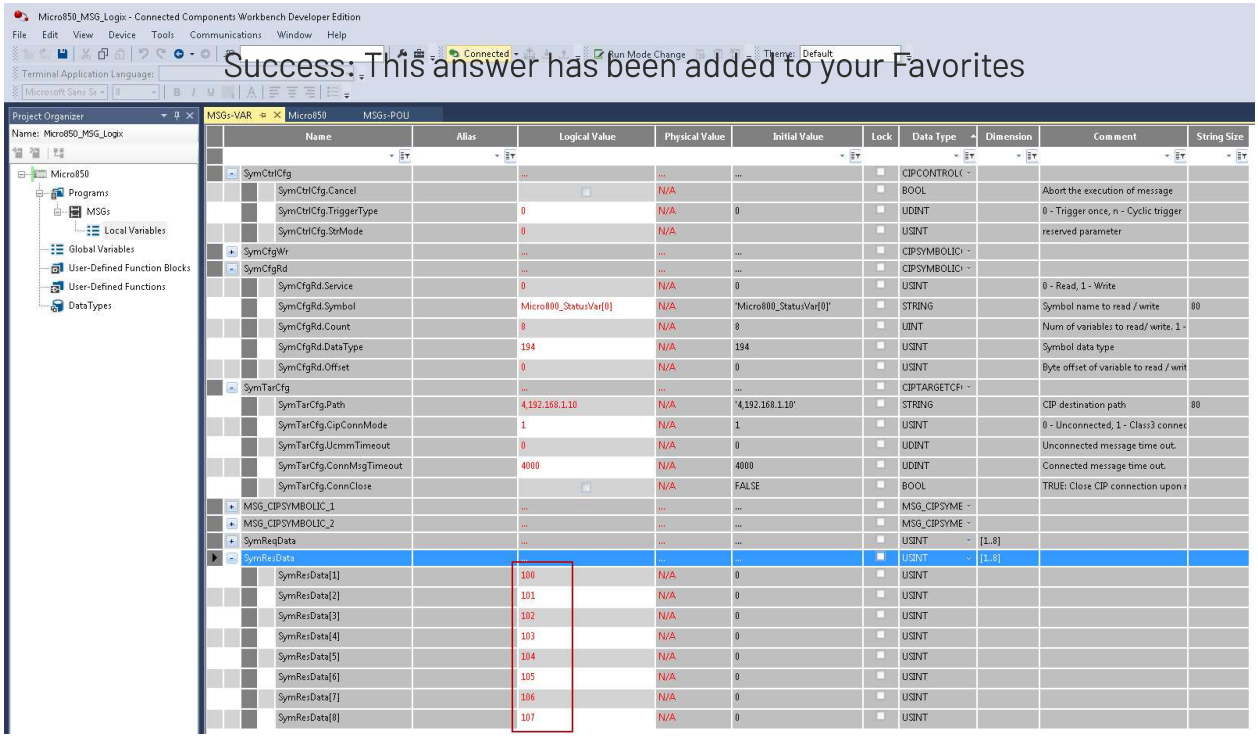
- **.DataType** set to the corresponding value from the helpfiles referenced above. In this case 194.
- **.Offset** set to the starting offset (in bytes) of the read variable. In this case `0`.

The **SymTarCfg** can be the same as for the write message.

Note: the **SymResData** array tags are all at zeros for the start.



Toggling input **_IO_EM_DI_00**, enables the Read MSG instruction, and provides the following change to the array values:



Attached are similar examples of messages between a Logix controller and a Micro850.

Attachments

File

Messaging_to_Micro850.ACD

Success: This answer has been added to your Favorites