Application Note #5477

Interfacing a DMC-40x0 with an Automation Direct Logic 205 PLC

Hardware

Controller: DMC-4040
Base: Automation Direct Logic 205
Brain: H2-EBC-100
Digital Input Module: D2-08ND3
Digital Output Module: D2-08TD1
Analog Input Module: F2-04AD-1
Analog Output Module: F2-02DA-2

Introduction

The following application note goes over the process for connecting an Automation Direct Logic 205 PLC to a DMC-40x0 via Ethernet. We will cover multiple steps from basic connectivity to reading and writing both digital and analog I/O. It is assumed that the user has knowledge of the Galil programming language, and specifically the SB, CB, MB, and IH commands. If not, please consult the controller command reference for a detailed description of each command. http://www.galilmc.com/literature/manuals.html

This note will talk about this hardware set up specifically, but in general any Galil controller can be used as a substitute for the DMC-40x0, such as the DMC-21x3, DMC-22x0, DMC-14x5…In addition, almost any PLC which can communicate over the ModBus protocol, or send ASCII strings, can be used instead of the Automation Direct part. If you have questions about interfacing to a specific PLC please contact Galil.

Connection

The easiest way to connect the Galil controller and the Automation Direct PLC is over an Ethernet connection. In our example, we have DMC-4040, a Automation Direct PLC, and a Windows based host PC all connected to a hub. It is important to note that when a Galil motion controller is communicating to a PLC using the ModBus protocol the Galil controller is the ModBus master and the PLC is the ModBus slave.

The first step is to establish a connection between the host PC and the controller. For details on this, please consult Chapter 2 of the Galil user manual. Once this connection is established, the next step is to have the controller communicate with the PLC. At this point you will need to know the IP address of the PLC.

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

Using the PLC's IP address and the IH command you can open a connection from the controller to the PLC. Note: When opening the connection you *must* use port 502. At this point you can confirm your connection by issuing a TH command to the controller and verifying that you are connected to the PLC. An example of this can be seen on the following page.

Addressing I/O

Before we begin controlling the I/O we need to discuss the numbering of the I/O. There are two important concepts to remember with this PLC:

1) The first digital input on the first digital input bank is input zero. The same is true of all other I/O as well.

2) When counting up digital inputs, ignore any modules in between. This means that if there is a bank of 8 digital inputs, then a bank of 8 digital outputs, and another bank of 8 digital inputs, the first bit on the second bank of digital inputs is input 8(inputs 0-7 reside on the first bank). Again, the same is true with regards to all other I/O.

Inputs		Outputs		Inputs	
0(0)	4(4)	X	X	0(8)	4(12)
1(1)	5(5)	X	X	1(9)	5(13)
2(2)	6(6)	X	X	2(10)	6(14)
3(3)	7(7)	X	X	3(11)	7(15)

*The number outside the parenthesis is the number physically written on the module. The number inside is the number used for addressing, otherwise known as the bitnumber.

*These rules are specific to this Automation Direct PLC, and for addressing for different PLC's you should consult their manual

ModBus Function Code Summary

Function Code	Used For
2	Reading digital inputs
3	Reading back values set for digital outputs
4	Reading analog inputs
5	Writing to digital outputs
6	Writing to analog ouputs

Digital Outputs

There are two different sets of commands that can be used for controlling digital I/O. The lowest level of control involves the user sending ModBus commands directly, using the Galil command MB. The alternative is to use the standard Galil commands for setting I/O on the controller, the SB and CB command.

SB & CB Commands

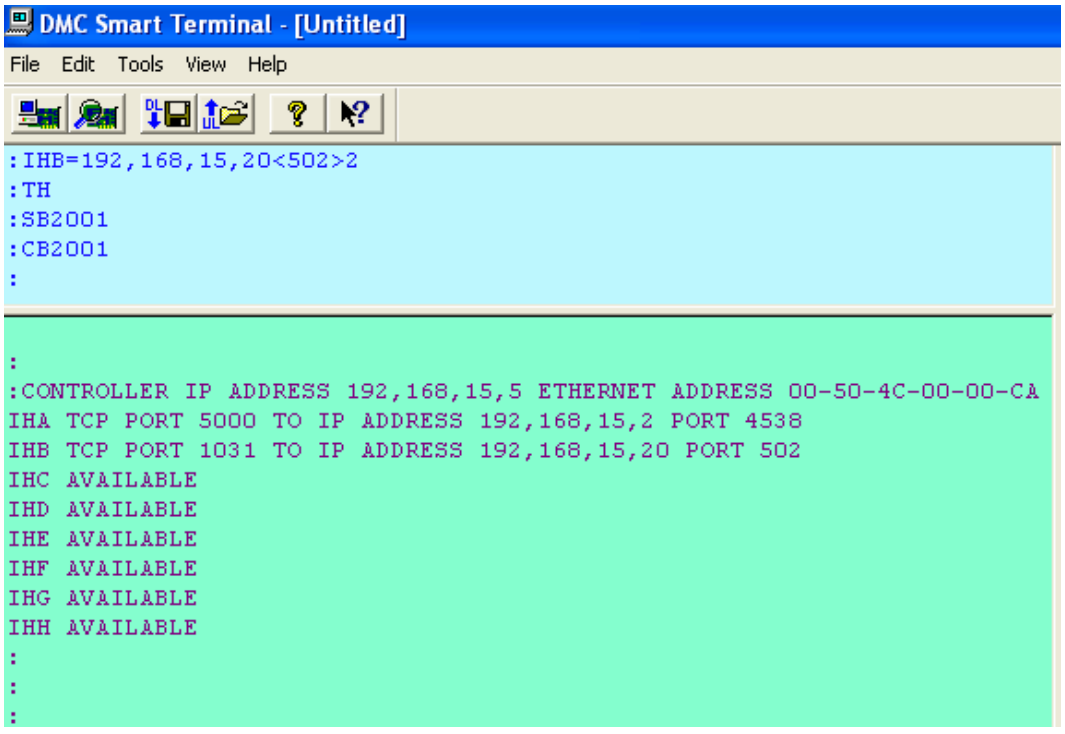
These commands are the common Galil commands to control I/O, but when connected on port 502, the controller then converts them to ModBus before they are sent out. With these commands there is one more catch to the addressing; the controller needs to know which handle the PLC is connected to, and that is done with the following equation-

Address = (HandleNumber)*1000 + BitNumber

HandleNumber: This is the number that corresponds to the letter used when opening the connection. For example, if a connection is opened with IHB=…, the handle number would be 2 because B is the second letter of the alphabet.

BitNumber: The bit number is determined with the rules from the section entitled "Addressing I/O".

Once the full address is determined, controlling the outputs is easy. The SB command turns the output on, and CB command clears the output. The following example demonstrates connecting from the controller to the PLC and then setting/clearing bit 1-



Note: In this example, handle A is used for the Ethernet connection to the computer and handle B is used for the connection to the PLC.

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

Digital Outputs and the MB Command

Besides the SB and CB commands, the I/O can also be controlled with direct ModBus commands using the Galil command MB. To do this, ModBus function code 5 is used. If your PLC is connected to handle B, and you want to set bit 6, the command would look like this-

MBB=0,5,6,1

Example Value	Reason
0	Slave Address not needed
5	ModBus function code for writing bits
6	BitNumber we wanted to write to
1	1 is for setting the bit, 0 is for clearing

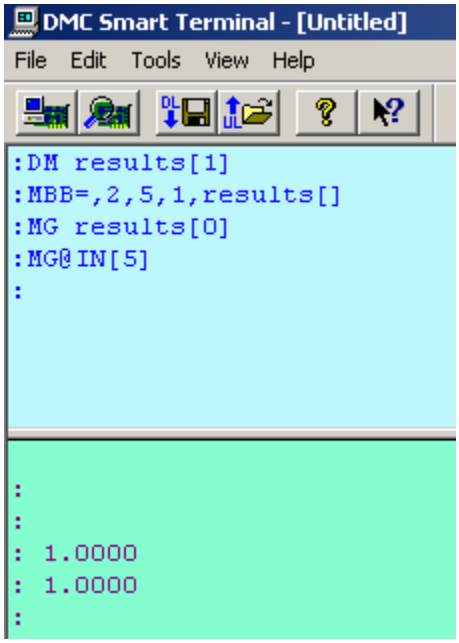
*For details on the MB command please see the command reference. In addition, when using the ModBus commands setting MW1 will insert a small delay between subsequent ModBus commands to allow devices time to process the commands.

Note: The bit number used is simply 6, and does not involve the equation needed for addressing with the SB and CB command. This is because the controller knows which handle to send it to by the letter after the MB.

Digital Inputs

Reading the status of the digital inputs also relies heavily on determining the address of the bit you are trying to read. Once you have the determine the address of the bit you are trying to read using the instruction in section above titled "Addressing I/O", you can then issue commands in two ways. The simplest way is by using MG@IN[address]. This is similar to the SB/CB in that is converts the command into the ModBus command that is sent to the PLC. Again, the user also has the option of issuing the commands in ModBus directly with the MB command. ModBus function 2 is used for reading digital inputs. The following example demonstrates both methods which accomplish the same thing-

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

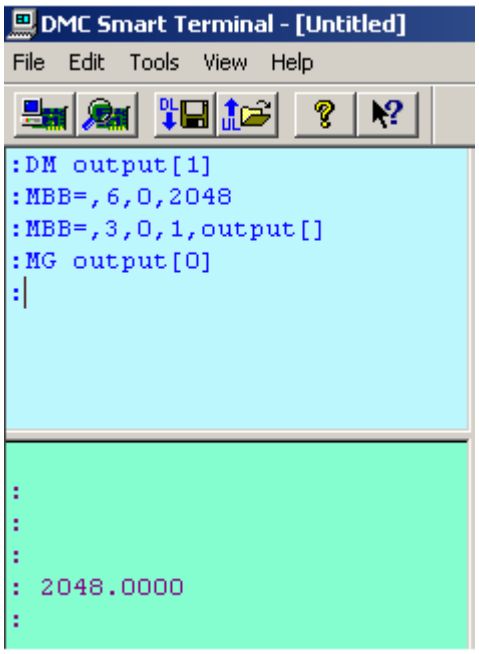


Example Value	Reason
0	Slave address not needed
2	ModBus function code for reading digital inputs
5	Starting bit number we want to read
1	Number of bits to read
results[]	Where the results are stored

Note: For the MB command, you must first dimension an array. The result will be stored in element 0 of the array.

Analog Outputs

Analog outputs follow the same bit numbering rules as described in the "Addressing I/O" section. Once you've determined the address of the channel you would like to control, you then use ModBus function 6 to write to the register. For this module the resolution is 12 bits, so the value you set must be in a range of 1 to 4096. You can also read back the value that you have set with the ModBus function code 3.
Note: Some Modbus devices will use floating point format to describe analog values and Galil does support this format.



First MB command

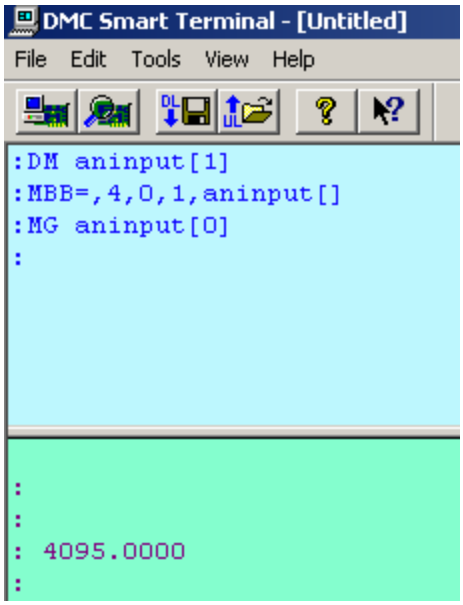
Example Value	Reason
0	Slave address not needed
6	ModBus function code for setting analog outputs
0	Bit number we chose to write to
2048	Value between 0 and 4096. This value sets our output to 5V.

Second MB Command

Example Value	Reason
0	Slave address not needed
3	Function code for reading back analog output settings
0	Bit number for analog output channel we are reading from
1	Number of words to read
output[]	Array where results are stored

Analog Inputs

The addressing for analog inputs follows the same rules outlined in "Addressing I/O". For reading analog inputs you use ModBus function 4. The module used in this example ranges from 4-20ma and has 12 bits of resolution ranging from 1 to 4095. Therefore when we read a value of 4095 it corresponds to 20ma.



Example Value	Reason
0	Slave address not needed
4	Function code for reading analog inputs
0	Bit number for analog input to read from
1	Number of words to read from
aninput[0]	Array where results are stored

Conclusion

By interfacing an Automation Direct Logic PLC to a Galil motion controller a user now has the power to control all their I/O through one program that can reside on the motion controller. In addition, the user now has the ability to coordinate different I/O events with motion events seamlessly. Mating these two products allows the customer to have a more complete motion control system.

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com