

D3–02DA

2–Channel

Analog Output

In This Chapter. . . .

- Module Specifications
 - Connecting the Field Wiring
 - Module Operation
 - Writing the Control Program
-

Module Specifications

The following table provides the specifications for the D3-02DA Analog Output Module. Review these specifications to make sure the module meets your application requirements.

Number of Channels	2 (independent)
Output Ranges	0 – 10V, 4 – 20 mA
Resolution	8 bit (1 in 256)
Output Type	Single ended
Output Impedance	.5 Ω maximum, voltage output
Output Current	10 mA minimum, voltage output @ 10 VDC
Load Impedance	550 Ω maximum, 5 Ω minimum, current output
Total Inaccuracy	\pm 0.4% maximum at 25° C
Accuracy vs. Temperature	\pm 50 ppm / °C maximum
Conversion Time	100 μ s maximum (2 channels/scan)
Power Budget Requirement	80 mA @9V
External Power Supply	24 VDC, \pm 10%, 170 mA, class 2
Operating Temperature	32° to 140° F (0° to 60° C)
Storage Temperature	-4° to 158° F (-20° to 70° C)
Relative Humidity	5 to 95% (non-condensing)
Environmental air	No corrosive gases permitted
Vibration	MIL STD 810C 514.2
Shock	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304

Analog Output Configuration Requirements

The D3-02DA Analog Output appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

- For local and expansion systems, the available power budget and 16-point module usage are the limiting factors.

Connecting the Field Wiring

Wiring Guidelines Your company may have guidelines for wiring and cable installation. If so, you should check those before you begin the installation. Here are some general things to consider.

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the module or the power supply return (0V). *Do not* ground the shield at both the module and the transducer.
- Don't run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.
- Route the wiring through an approved cable housing to minimize the risk of accidental damage. Check local and national codes to choose the correct method for your application.

User Power Supply Requirements The D3-02DA requires a separate power supply. Choose a supply that meets the following requirements: 24 VDC $\pm 10\%$, Class 2, 170mA current (or greater, depending on the number of modules being used.)

Load Requirements Each channel can be wired independently for a voltage or current transducer.

- Current transducers must have an impedance between 5 and 550 ohms
- Voltage transducers must have an impedance greater than 1K ohms.

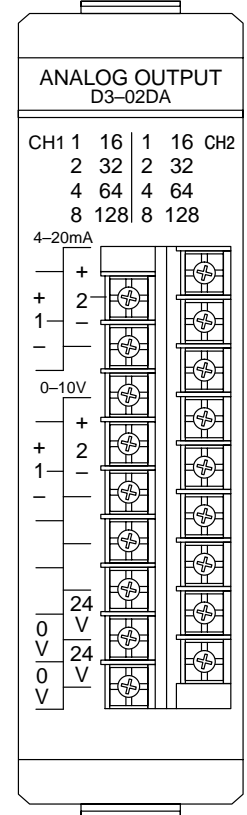
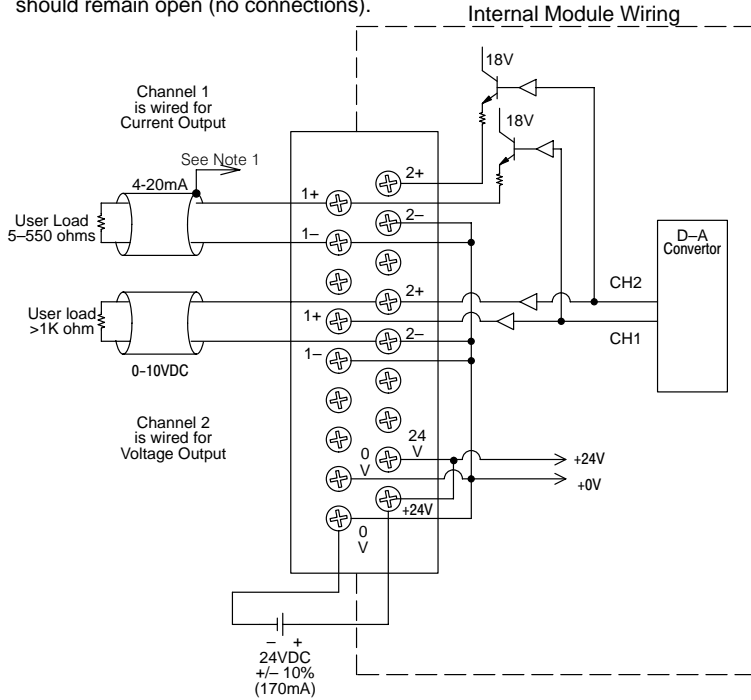
Removable Connector

The D3-02DA module has a removable connector to make wiring easier. Simply remove the retaining screws and gently pull the connector from the module.

Wiring Diagram

Note 1: Shields should be connected to the 0V of the module or to the 0V of the P/S.

Note 2: Unused voltage and current outputs should remain open (no connections).

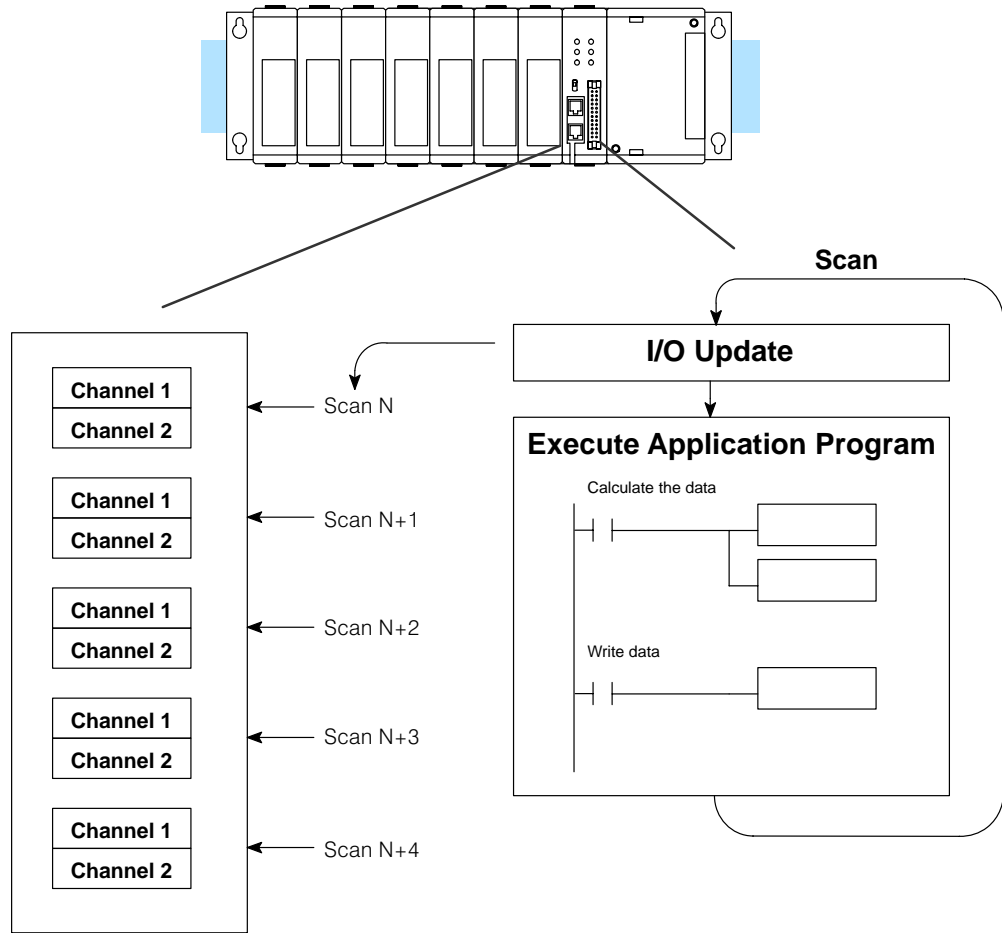


Module Operation

Before you begin writing the control program, it is important to take a few minutes to understand how the module processes and represents the analog signals.

Channel Scanning Sequence

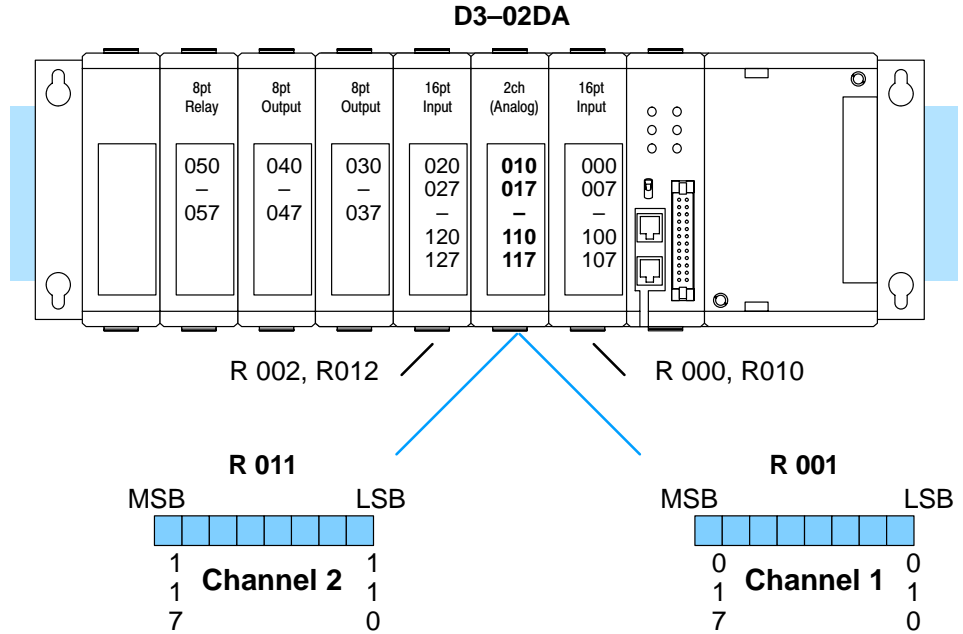
The D3-02DA module updates both channels in the same scan. The control program updates the two channels of this module independent of each other and each channel does not have to be refreshed on each scan.



Understanding the I/O Assignments

You may recall the D3-02DA module appears to the CPU as a 16-point module. These 16 points provide the digital representation of the analog signal.

Since all I/O points are automatically mapped into Register (R) memory, it is very easy to determine the location of the data word that will be assigned to the module.

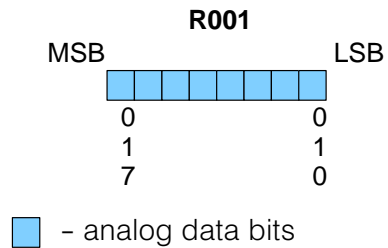


Within these two word locations, the individual bits represent specific information about the analog signal.

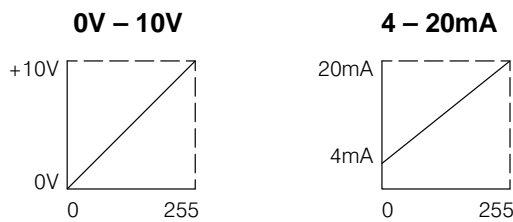
Analog Data Bits

The first register contains the data for channel one (R001). The second register contains the data for channel two (R011).

Bit	Value	Bit	Value
0	1	4	16
1	2	5	32
2	4	6	64
3	8	7	128



Since the module has 8-bit resolution, the analog signal is converted into 256 "pieces" ranging from 0 – 255 (2⁸). For example, with a 0 to 10V scale, a 0V signal would be 0, and a 10V signal would be 255. This is equivalent to a binary value of 0000 0000 to 1111 1111, or 00 to FF hexadecimal. The following diagram shows how this relates to each signal range.



Each "piece" can also be expressed in terms of the signal level by using the equation shown. The following table shows the smallest signal levels that will result in a change in the data value for each signal range.

$$\text{Resolution} = \frac{H - L}{255}$$

H = high limit of the signal range
L = low limit of the signal range

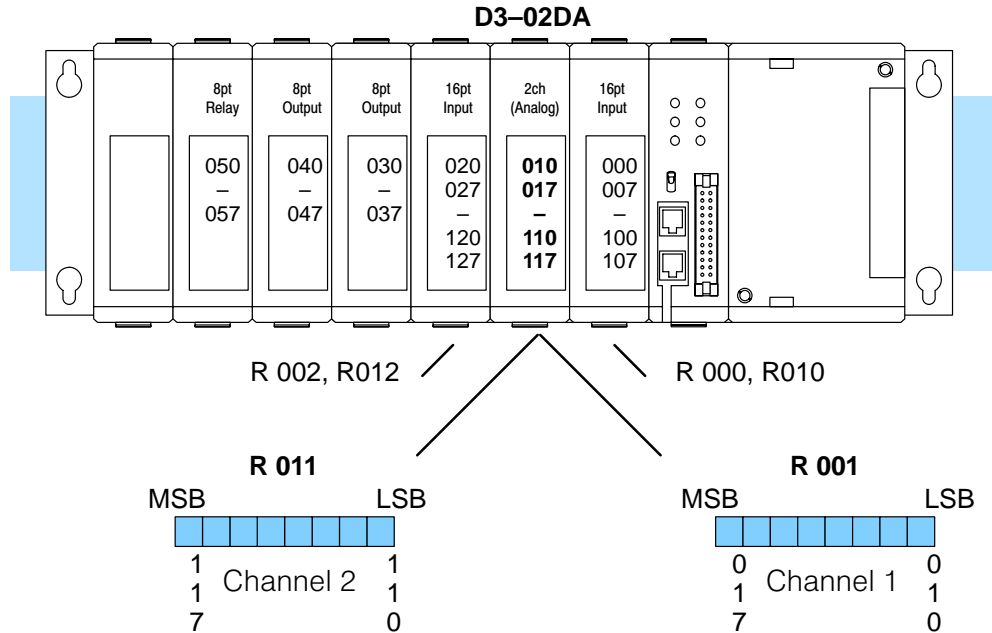
Range	Highest Signal	Lowest Signal	Smallest Change
0 to 10V	10V	0V	39 mV
4 to 20mA	20mA	4mA	62.5 μA

Now that you understand how the module and CPU work together to gather and store the information, you're ready to write the control program.

Writing the Control Program (DL330 / DL340)

Identifying the Data Locations

As mentioned earlier, you can update either channel or both channels during the same scan. Since the module does not have any channel select bits, you just simply determine the location of the data word and send the data word to the output module whenever you need to update the data.



Calculating the Digital Value

Your program has to calculate the digital value to send to the analog module. There are many ways to do this, but most all applications are understood more easily if you use measurements in engineering units. This is accomplished by using the conversion formula shown. You may have to make adjustments to the formula depending on the scale you choose for the engineering units.

$$A = 256 \frac{U}{H - L}$$

- A = Analog value (0 - 255)
- U = Engineering Units
- H = high limit of the Engineering unit range
- L = low limit of the Engineering unit range

The following example shows how you would use Engineering Units to obtain the digital value to represent pressure (PSI) from 0 to 100. This example assumes you want to obtain a pressure of 42 PSI, which is slightly less than half scale.

$$A = 256 \frac{U}{H - L}$$

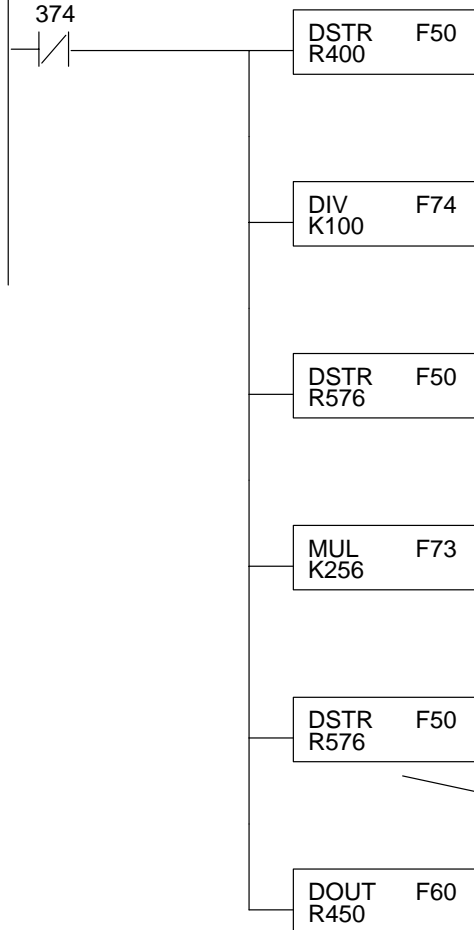
$$A = 256 \frac{42}{100 - 0}$$

$$A = 107.5 \text{ (or 108)}$$

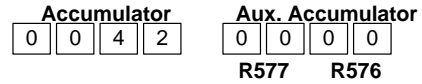
Here's how you would write the program to perform the Engineering Unit conversion. This example assumes you have calculated or loaded the engineering unit value and stored it in R400. Also, you have to perform this for both channels if you're using different data for each channel.

This example assumes you have already loaded the Engineering unit value in R400.

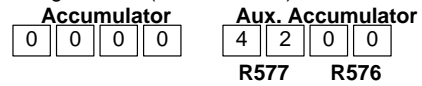
Scale the data



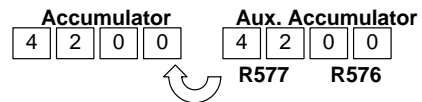
This instruction loads Engineering unit value into the accumulator.



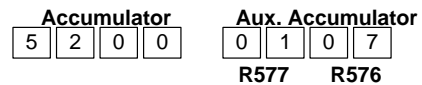
The Engineering unit value is divided by the Engineering unit range (42/100=.42). In this case the range is 100. (100 - 0 = 100)



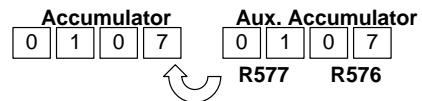
This instruction moves the two-byte decimal portion into the accumulator for further operations.



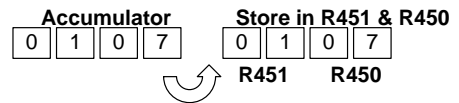
The accumulator is then multiplied by the module resolution, which is 256. (256 x 4200 = 1075200). Notice the most significant digits are now stored in the auxiliary accumulator. (This is different from the Divide instruction operation.)



This instruction moves the two-byte auxiliary accumulator for further operations.



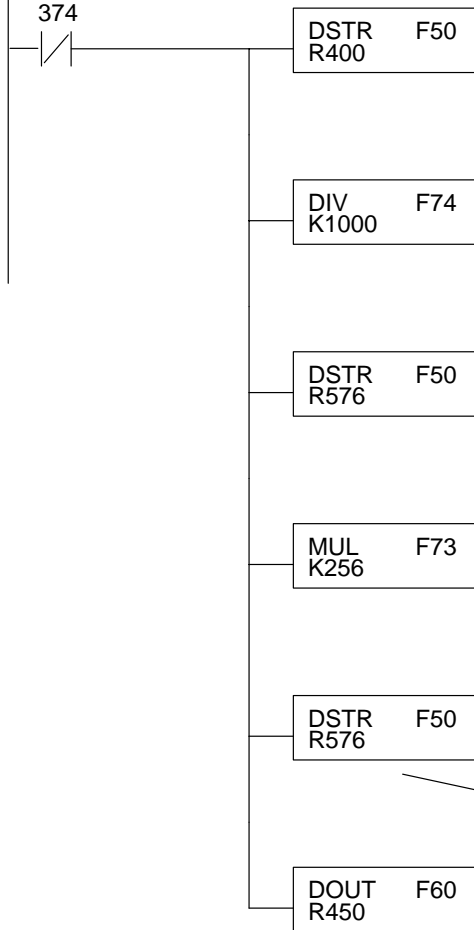
This instruction stores the accumulator to R451 and R450. R451 and R450 now contain the digital value, which is 107.



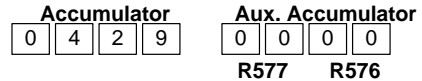
There will probably be times when you need more precise control. For example, maybe your application requires 42.9 PSI, not just 42 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve entered 429 as the Engineering unit value and we’ve added another digit to the scale. Instead of a scale of 100, we’re using 1000, which implies 100.0 for the PSI range.

This example assumes you have already loaded the Engineering unit value in R400.

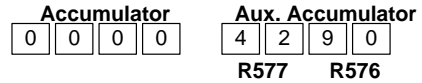
Scale the data



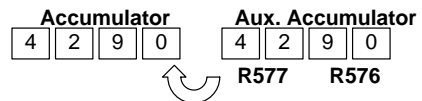
This instruction loads Engineering unit value into the accumulator.



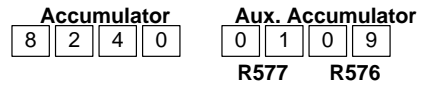
The Engineering unit value is divided by the Engineering unit range, which in this case is 1000. (100.0 implied range) (429/1000 = .429)



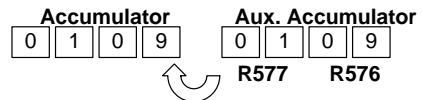
This instruction moves the two-byte decimal portion into the accumulator for further operations.



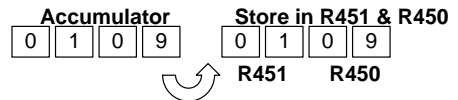
The accumulator is then multiplied by the module resolution, which is 256. (256 x 4290 = 1098240). Notice the most significant digits are now stored in the auxiliary accumulator. (This is different from the Divide instruction operation.)



This instruction moves the two-byte auxiliary accumulator for further operations.



This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the digital value, which is 109.

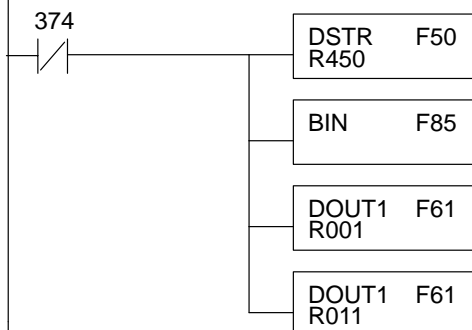


Sending the Same Data to Both Channels

In some applications, you'll want to send the same output values to both channels. The following program example shows how to send the digital values to the module.

This example assumes you have already loaded the Engineering unit value in R450 and R451.

Send Channel 1 & 2



This rung loads the data into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

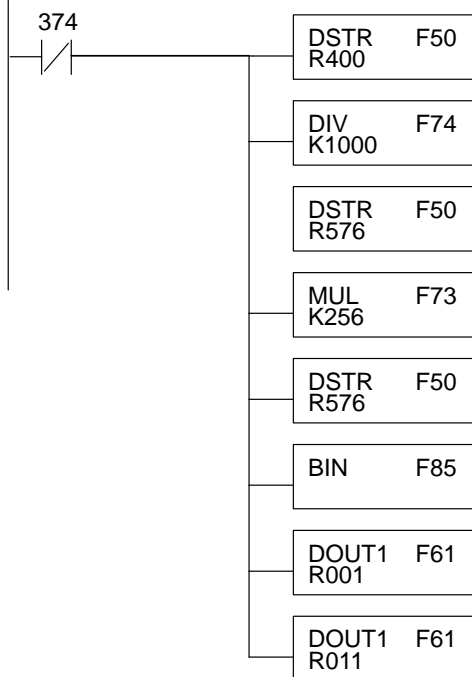
Send the accumulator data to the Register that corresponds to channel 1, which is R001.

Send the accumulator data to the Register that corresponds to channel 2, which is R011.

If you want a shorter program, just combine the data scaling and output instructions.

This example assumes you have already loaded the Engineering unit value in R400.

Send Channel 1 & 2



This instruction loads Engineering unit value into the accumulator.

The Engineering unit value is divided by the Engineering unit range, which in this case is 1000. (100.0 implied range)

This instruction moves the two-byte decimal portion into the accumulator for further operations.

The accumulator is then multiplied by the module resolution, which is 256.

This instruction moves the two-byte auxiliary accumulator into the regular accumulator.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

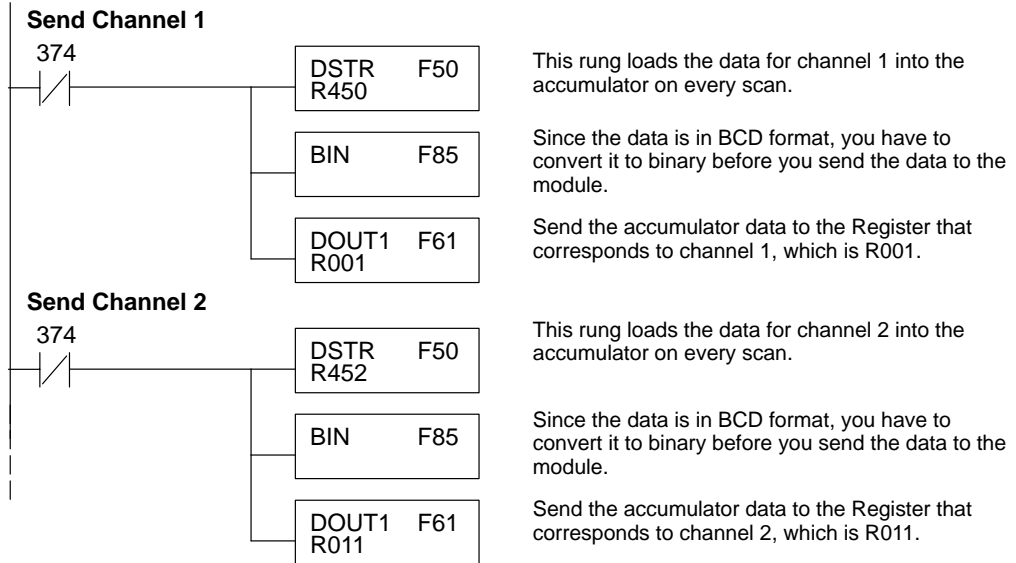
Send the accumulator data to the Register that corresponds to channel 1, which is R001.

Send the accumulator data to the Register that corresponds to channel 2, which is R011.

Sending Specific Data to Each Channel

In this case, the example logic is setup to send different data to each channel. Of course, you would have to have separate routines to calculate the output data and you would have to store the different values in separate registers.

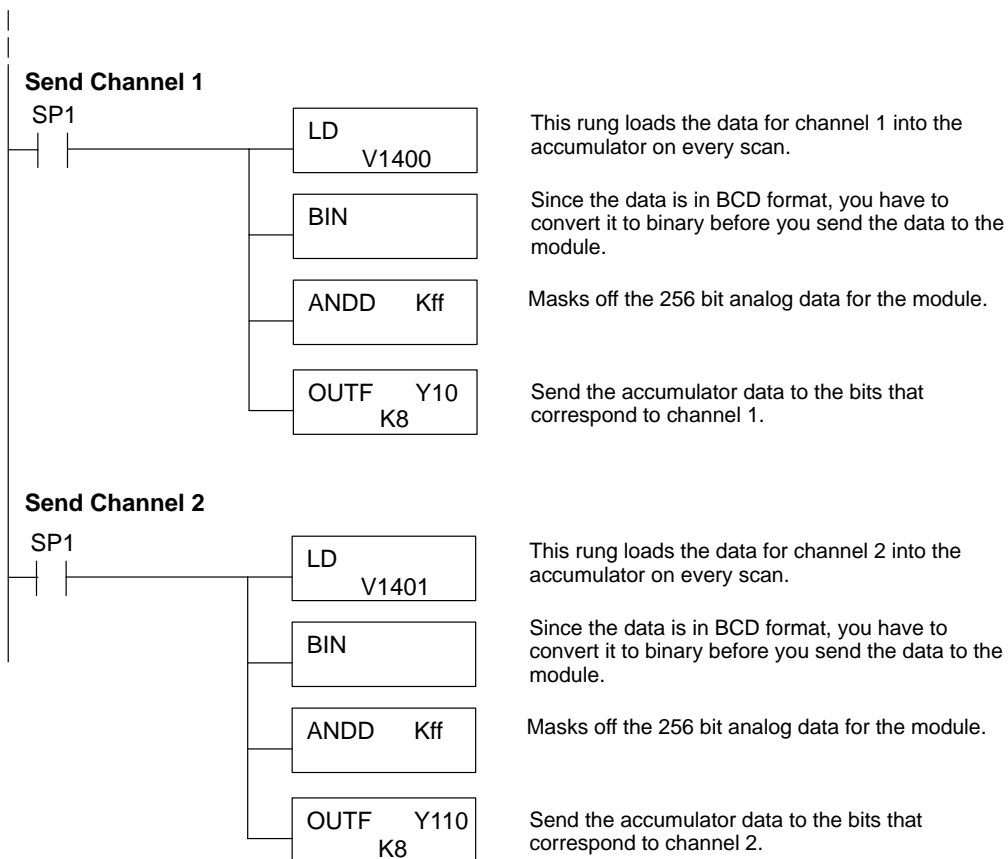
This example assumes you have already loaded the Engineering unit value for Channel 1 in R450 and R451 and the data for Channel 2 in R452 and R453.



Writing the Control Program (DL350)

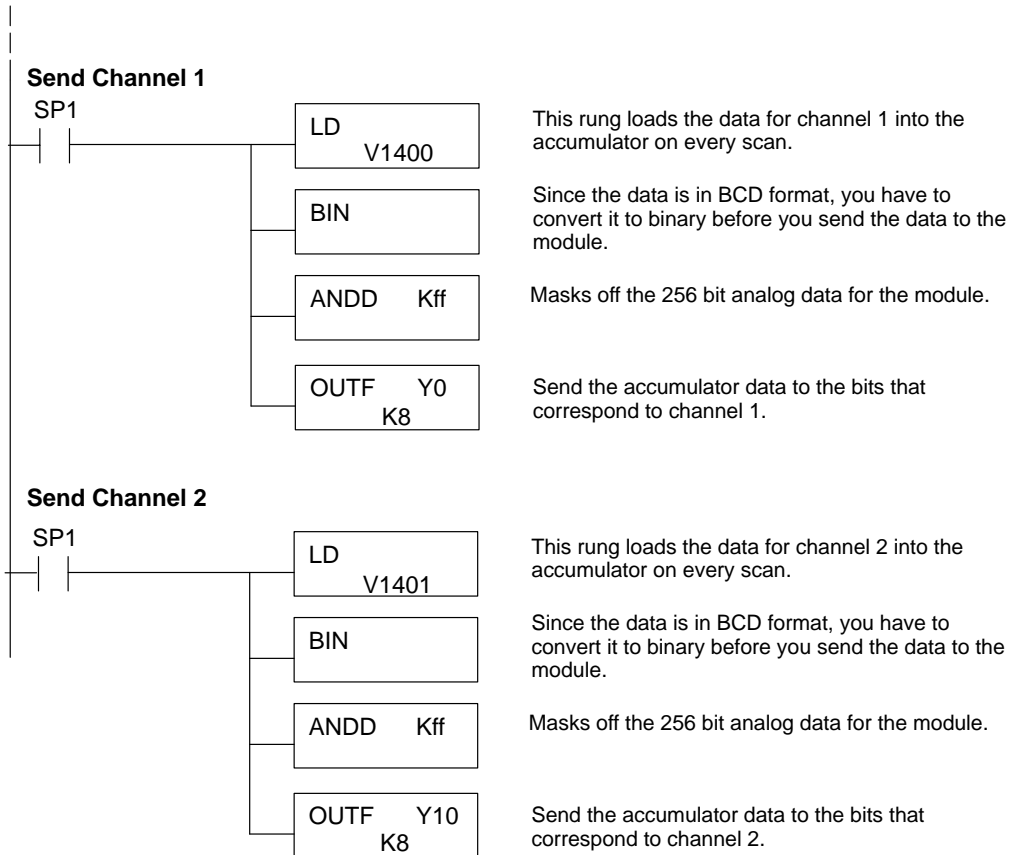
Multiplexing: DL350 with a Conventional DL305 Base

This example assumes the module is in the Y10-17 / Y110-117 slot of a 305 conventional base. In this example V1400 contains the BCD data for channel 1 and V1401 contains the data for channel 2.



**Multiplexing:
DL350 with a
D3-xx-1 Base**

This example assumes the module is in Y0 address slot of a D3-xx-1 base . In this example V1400 contains the BCD data for channel 1 and V1401 contains the data for channel 2. If any expansion bases are used in the system, they must all be D3-xx-1 to be able to use this example. Otherwise, the conventional base addressing must be used.



Analog and Digital Value Conversions

Sometimes it is helpful to be able to quickly convert between the voltage or current signal levels and the digital values. This is especially helpful during machine startup or troubleshooting. The following table provides formulas to make this conversion easier.

Range	If you know the digital value ...	If you know the analog signal level ...
0 to 10V	$A = \frac{10D}{255}$	$D = \frac{255}{10} A$
4 to 20mA	$A = \frac{16D}{255} + 4$	$D = \frac{255}{16}(A - 4)$

For example, if you are using the 4–20mA range and you know you need a 10mA signal level, you would use the following formula to determine the digital value that should be sent to the module.

$$D = \frac{255}{16}(A - 4)$$

$$D = \frac{255}{16}(10\text{mA} - 4)$$

$$D = (15.93) (6)$$

$$D = 96$$

Calculating the Digital Value

Your program must calculate the digital value to send to the analog module. There are many ways to do this, but most applications are understood more easily if you use measurements in engineering units. This is accomplished by using the conversion formula shown.

You may have to make adjustments to the formula depending on the scale you choose for the engineering units.

$$A = U \frac{255}{H - L}$$

A = Analog value (0 – 255)

U = Engineering Units

H = high limit of the engineering unit range

L = low limit of the engineering unit range

Consider the following example which controls pressure from 0.0 to 99.9 PSI. By using the formula, you can easily determine the digital value that should be sent to the module. The example shows the conversion required to yield 49.4 PSI. Notice the formula uses a multiplier of 10. This is because the decimal portion of 49.4 cannot be loaded, so you adjust the formula to compensate for it.

$$A = 10U \frac{255}{10(H - L)}$$

$$A = 494 \frac{255}{1000 - 0}$$

$$A = 126$$



The example program below shows how you would write the program to perform the engineering unit conversion. This example assumes you have calculated or loaded the engineering unit values in BCD and stored them in V2300 and V2301 for channels 1 and 2 respectively.

NOTE: The DL350 offers various instructions that allow you to perform math operations using BCD format. It is easier to perform math calculations in BCD and then convert the value to binary before sending the data to the module.

