

F3–04DAS

4-Channel Isolated Analog Output

In This Chapter. . . .

- Module Specifications
 - Setting the Module Jumpers
 - Connecting the Field Wiring
 - Module Operation
 - Writing the Control Program (DL340/DL350)
 - Writing the Control Program (DL350)
-

Module Specifications

The following table provides the specifications for the F3-04DAS Analog Output Module. Review these specifications to make sure the module meets your application requirements.

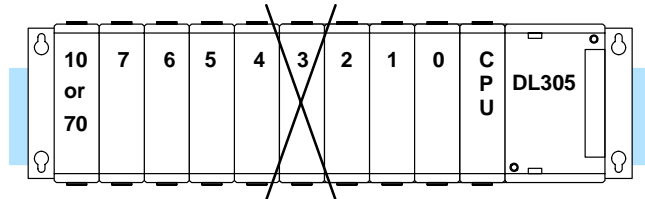
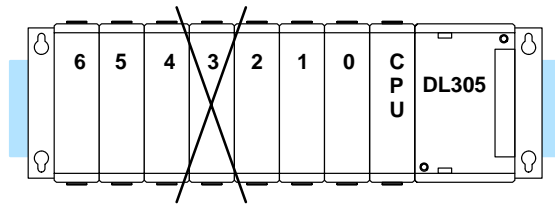
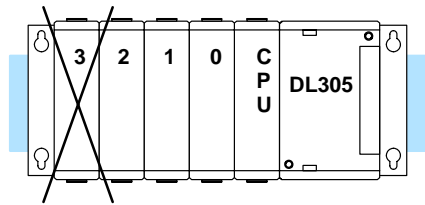
Number of Channels	4
Output Ranges	$\pm 5\text{V}$, $\pm 10\text{V}$, $0-5\text{V}$, $0-10\text{V}$, $1-5\text{V}$, $0-20\text{ mA}$, $4-20\text{ mA}$
Resolution	12 bit (1 in 4096)
Output Type	Isolated, 750 VDC channel-to-channel 750 VDC channel-to-logic
Output Current	$\pm 5\text{ mA}$, voltage output
Short-circuit Current	$\pm 20\text{ mA}$ typical, voltage output
Capacitive Load Drive	$0.1\mu\text{F}$ typical, voltage output
Load Impedance	470Ω maximum, current output $2\text{K}\Omega$ minimum, voltage output
Isolation Mode Rejection	140 dB at 60Hz
Linearity Error	± 1 count ($\pm 0.03\%$ maximum)
Calibration Error	$\pm 0.15\%$ typical, $\pm 0.75\%$ maximum of span $\pm 10\text{ ppm} / ^\circ\text{C}$ maximum of full scale
Calibrated Offset Error	± 1 count maximum, current output $\pm 5\text{ mV}$ typical, $\pm 50\text{ mV}$ max., voltage output $\pm 0.2\text{ mV}$ typical / $^\circ\text{C}$
Conversion Time	$30\mu\text{S}$ maximum, 1 channel/scan
Power Budget Requirement	154 mA @9V, 145 mA @ 24V (maximum)
External Power Supply	None required
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 F3-04DAS Analog Output appears as a 16-point module. The module can be installed in any slot configured for 16 points, but **should not be installed in Slot 3 of any DL305 base**. 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.

WARNING: You should not install this module in Slot 3 of any DL305 base. The module has traces on the edge card connector that may become damaged if the module is repeatedly installed and removed. The solder mask that protects the traces may be scraped off, which may cause a short circuit on the I/O bus. The short circuit can lead to unpredictable system operation or cause damage to the CPU or power supply.



Setting the Module Jumpers

Jumper Locations The module is set at the factory for a 0–10V signal on all four channels. If this is acceptable you do not have to change any of the jumpers.

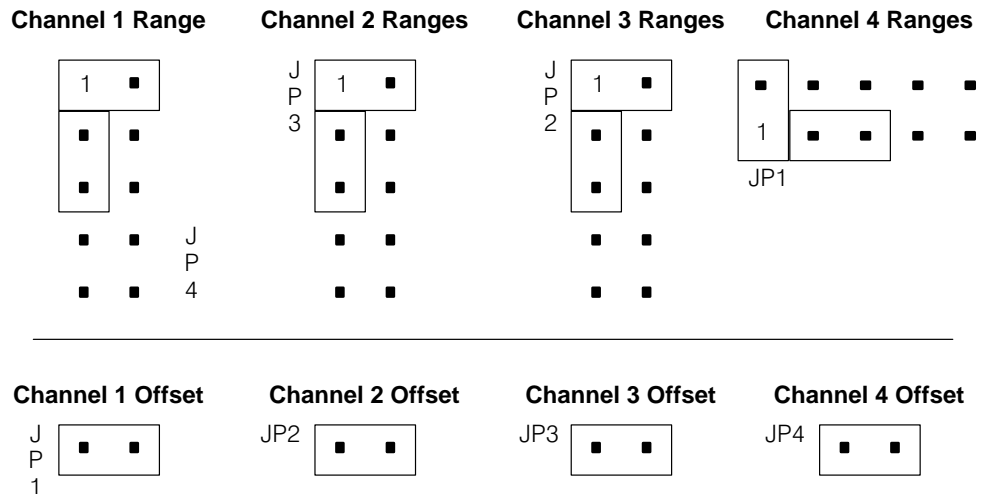
If you examine the top board on the module you will notice four sets of jumpers. The jumpers are assigned to the channels as follows.

- Channel 1 — Jumper JP4
- Channel 2 — Jumper JP3
- Channel 3 — Jumper JP2
- Channel 4 — Jumper JP1

NOTE: At first glance it might appear we have the channel / jumper assignments out of order. Your eyes do not deceive you. Channel 1 *is* controlled by JP4.

Each channel also has a jumper located on the bottom board of the module. These jumpers select a 1V (or 4mA) offset for each channel. Remove the jumper for any range that requires an offset. These jumpers are assigned as expected. JP1 selects an offset for channel 1, JP2 selects an offset for channel 2, etc.

The following diagram shows how the jumpers are assigned. It also shows the factory settings.



Selecting Input Signal Ranges

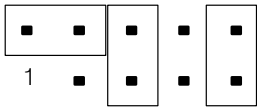
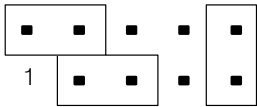
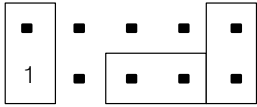
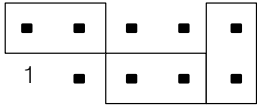
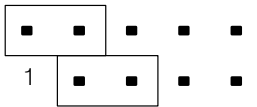
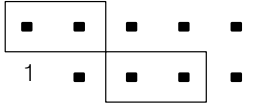
The following tables show the jumper selections for the various ranges. (Only channel 1 is used in the example, but all channels must be set.)

Bipolar Signal Range	Jumper Settings
-5 VDC to +5 VDC	<p>Channel 1 (JP4) Offset Jumper (JP1)</p>
-10 VDC to +10 VDC	<p>Channel 1 (JP4) Offset Jumper (JP1)</p>
Unipolar Signal Range	Jumper Settings
4 to 20 mA (1 VDC to 5 VDC)	<p>Channel 1 (JP4) Offset Jumper (JP1)</p>
0 VDC to +5 VDC (0 to +20 mA)	<p>Channel 1 (JP4) Offset Jumper (JP1)</p>
0 VDC to +10 VDC	<p>Channel 1 (JP4) Offset Jumper (JP1)</p>

F3-04DAS
 4 Ch. Isolated Analog Out.

Special Output Signal Ranges

The following tables show the jumper selections for some additional ranges that are not normally found in many applications. Notice you can install or remove the offset jumper to change the settings. (Only channel 1 is used in the example, but all channels must be set.)

Signal Range Offset Installed	Signal Range Offset Removed	Jumper Settings
-10 VDC to +6 VDC	-9 VDC to +7 VDC	Channel 1 (JP4) 
-5 VDC to +3 VDC	-4 VDC to +4 VDC	Channel 1 (JP4) 
-2.5 VDC to +2.5 VDC	-1.5 VDC to +3.5 VDC	Channel 1 (JP4) 
-2.5 VDC to +1.5 VDC	-1.5 VDC to +2.5 VDC	Channel 1 (JP4) 
0 VDC to 8 VDC	1 VDC to 9 VDC	Channel 1 (JP4) 
0 VDC to 4 VDC	1 VDC to 5 VDC	Channel 1 (JP4) 

F3-04DAS
4 Ch. Isolated Analog Out.

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 F3-04DAS receives all power from the base. A separate power supply is not required.

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

- Current transducers must have an impedance less than 470 ohms.
- Voltage transducers must have an impedance greater than 2K ohms.

Removable Connector

The F3-04DAS module has a removable connector to make wiring easier. Simply squeeze the top and bottom tabs and gently pull the connector from the module.

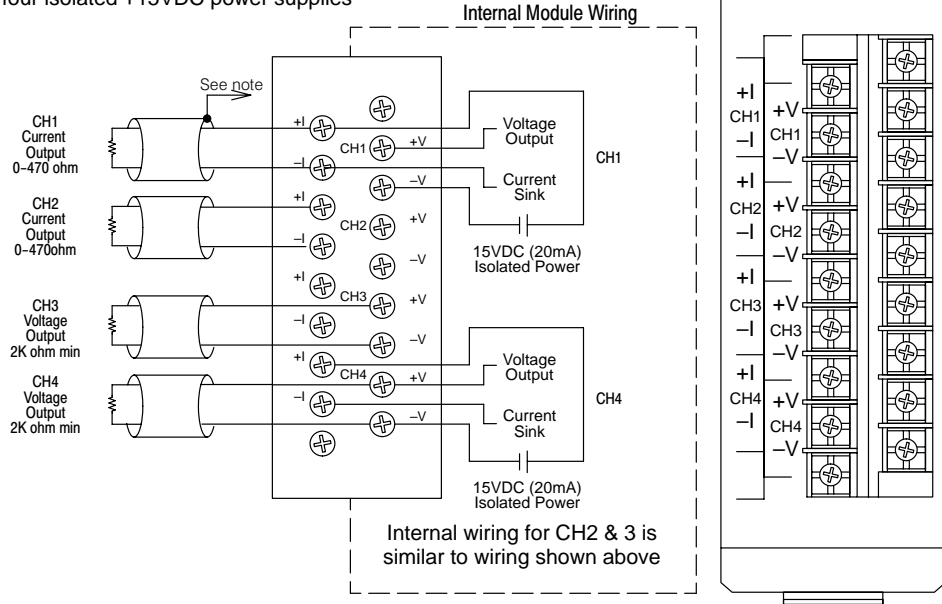
Wiring Diagram

Note 1: Shields should be connected to the respective channel's -V terminal of the module.

Note 2: Each isolated output channel may have either a voltage or current load, but not both

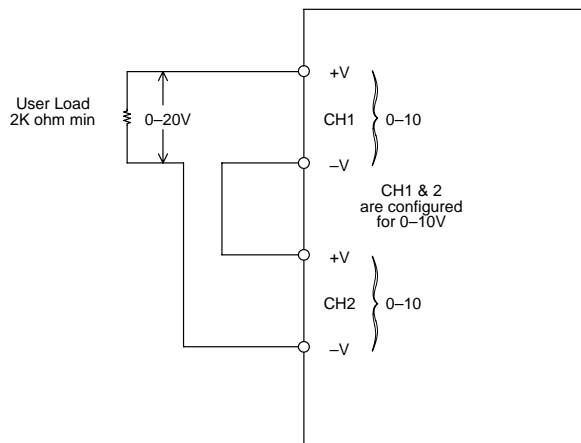
Note 3: An external 0.31 Amp fast-acting fuse in series with the isolated +I terminal (+15VDC) is recommended to protect against accidental shorts to the -V terminal (15VDC common)

Note 4: Do not attempt to source more than 20mA from any one of the four isolated +15VDC power supplies



Combining Voltage Outputs

You may occasionally encounter transmitters that have a very unusual signal range. Since each channel is isolated, you can “daisy chain” the channels to provide output voltage signals that are outside of the normal operating range. For example, you could connect the first two channels to provide a voltage output from 0 to 20 VDC.



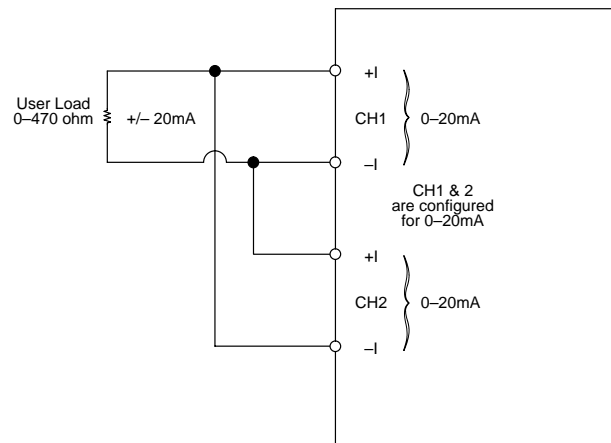
F3-04DAS
4 Ch. Isolated Analog Out.

Combining Current Outputs

You cannot connect the current outputs in series (like the voltage outputs) but you can achieve unusual ranges with a few wiring and programming tricks. For example, let's say an application requires a ± 20 mA range. By completing the following steps, you could easily accommodate this requirement.

1. Configure channel 1 and channel 2 for 0–20mA.
2. Connect the +I of channel 1 to the –I of channel 2.
3. Connect the –I of channel 1 to the +I of channel 2.
4. Send 0 (digital value) to channel 2 while you send 0–4095 (digital value) to channel 1. To reverse the power flow, send 0 to channel 1 while you send the 0–4095 value to channel 2. (See the section on Writing the Control Program for information on sending data values.)

WARNING: The isolated +15 VDC power supplies are rated at a maximum of 20 mA. Current ratings that exceed 20 mA will damage the module beyond repair. For example, if you used the 0–10 VDC range for the example, the current would approach 40 mA which would cause damage to the module.

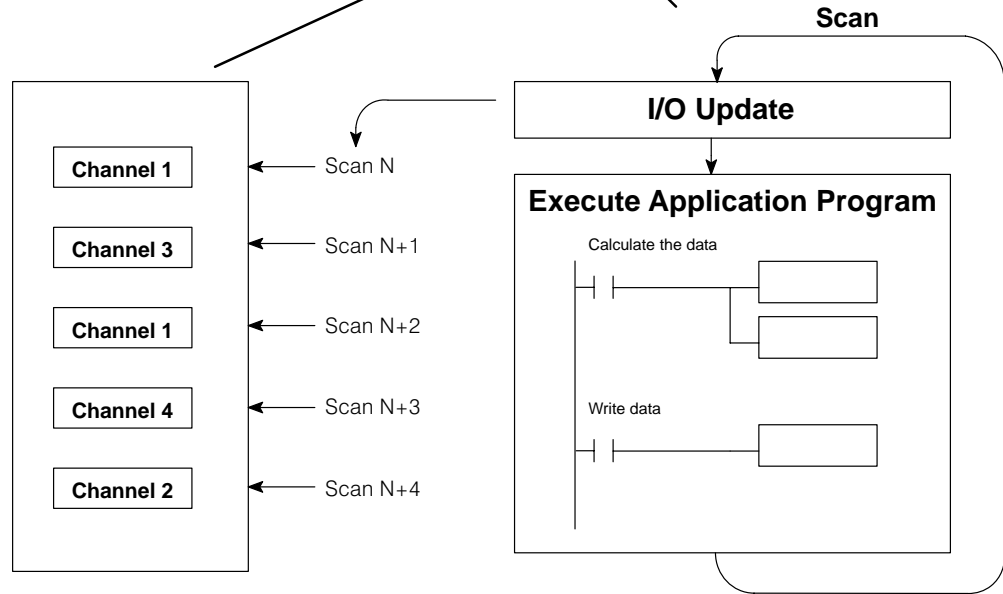
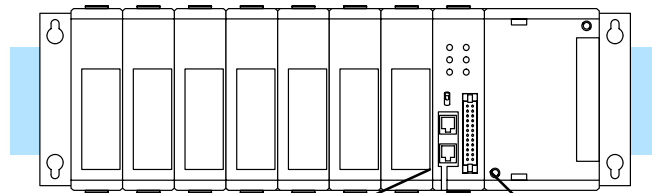


Module Operation

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 F3-04DAS module can update one channel per CPU scan. Your RLL program selects the channel to update, so you have complete flexibility in solving your application requirements.



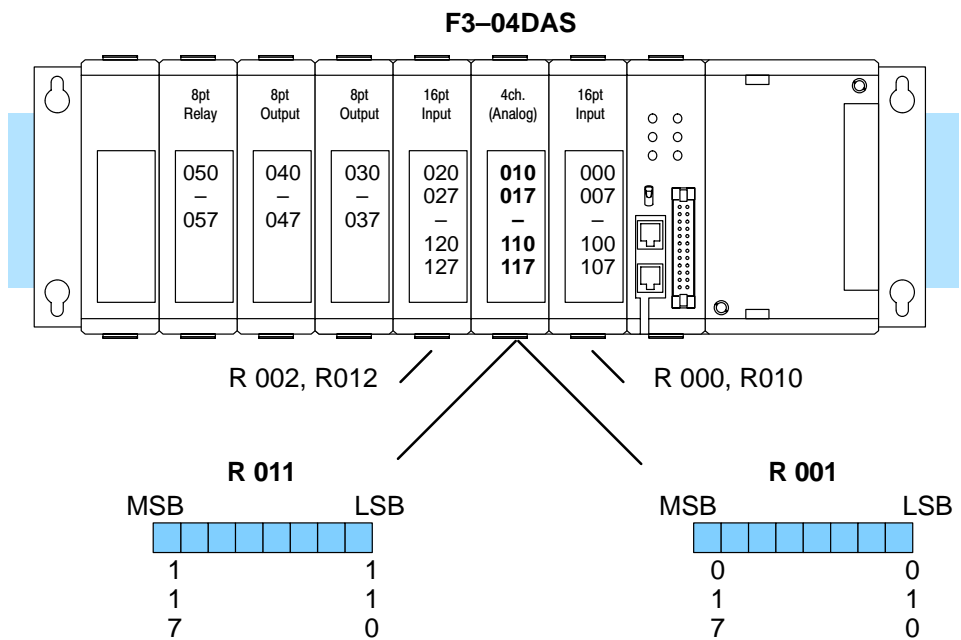
F3-04DAS
4 Ch. Isolated Analog Out.

Understanding the I/O Assignments

You may recall the F3-04DAS module appears to the CPU as a 16-point module. These 16 points provide:

- the digital representation of the analog signal.
- identification of the channel to receive the data.

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.

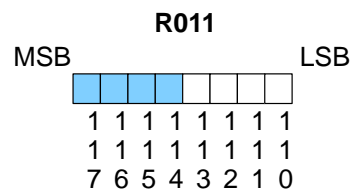


Within these two word locations, the individual bits represent specific information about the analog signal.

Channel Selection Inputs

The last four points of the upper register are used as outputs to tell the module which channel to update. In our example, when output 114 is on, channel 1 will be updated. Here's how the outputs are assigned.

Output	Channels
114	1
115	2
116	3
117	4



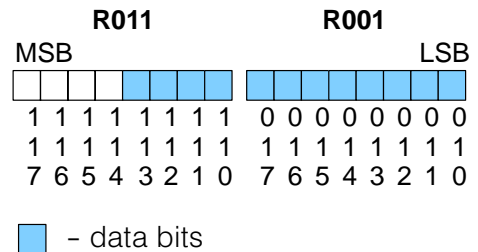
- channel selection inputs

F3-04DAS 4 Ch. Isolated Analog Out.

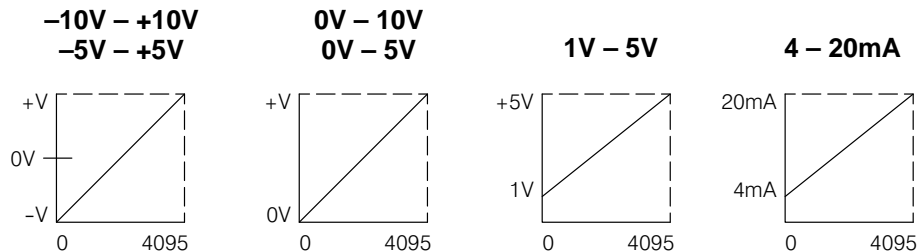
Analog Data Bits

The remaining twelve bits represent the analog data in binary format.

Bit	Value	Bit	Value
0 (LSB)	1	6	64
1	2	7	128
2	4	8	256
3	8	9	512
4	16	10	1024
5	32	11	2048



Since the module has 12-bit resolution, the analog signal is converted into 4096 “pieces” ranging from 0 – 4095 (2^{12}). For example, with a 0 to 10V scale, a 0V signal would be 0, and a 10V signal would be 4095. This is equivalent to a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF 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 possibly result in a change in the data value for each signal range.

$$\text{Resolution} = \frac{H - L}{4095}$$

H = high limit of the signal range

L = low limit of the signal range

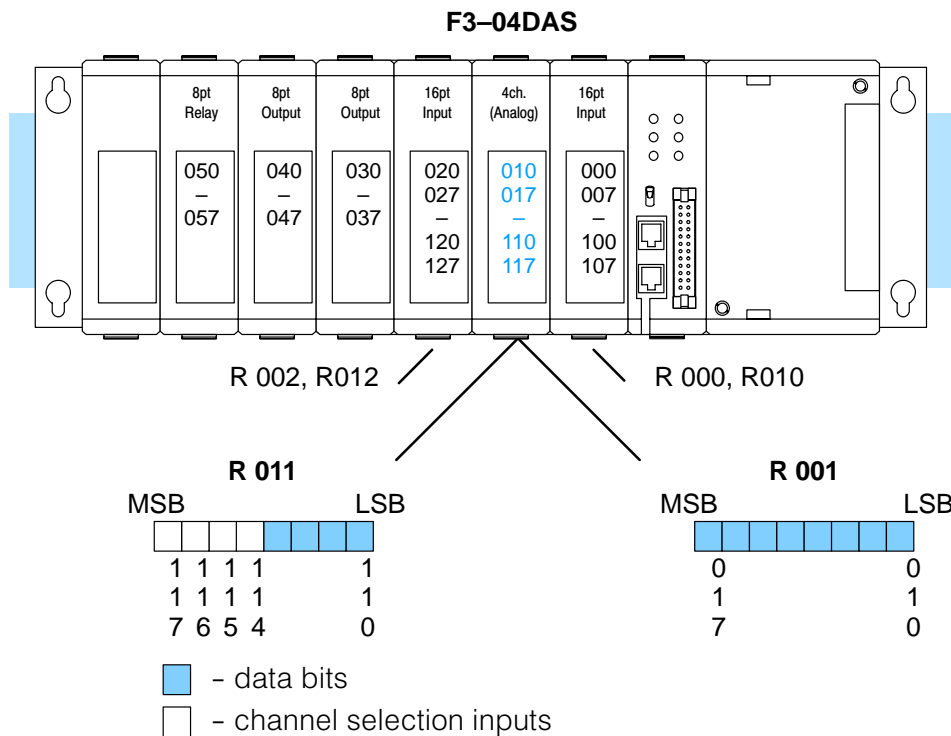
Range	Highest Signal	Lowest Signal	Smallest Change
-10 to +10V	+10V	-10V	4.88 mV
-5 to +5V	+5 V	-5V	2.44 mV
0 to 5V	5V	0V	1.22 mV
0 to 10V	10V	0V	2.44 mV
1 to 5V	5V	1V	0.98 mV
4 to 20mA	20mA	4mA	3.91 μ A

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 use the channel selection bits to determine which channels will be updated. The following diagram shows the location for both the channel selection bits and data bits.



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.

$$A = 4096 \frac{U}{H - L}$$

- A = Analog value (0 – 4095)
- 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.

$$A = 4096 \frac{U}{H - L}$$

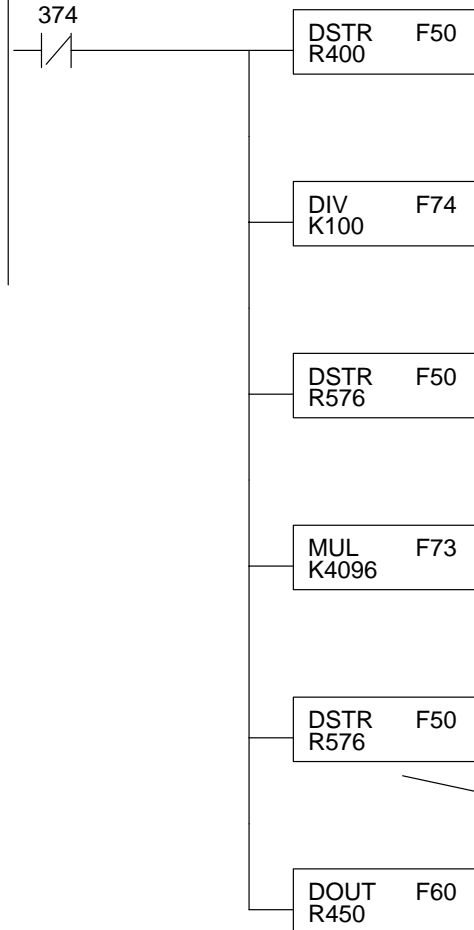
$$A = 4096 \frac{42}{100 - 0}$$

$$A = 1720$$

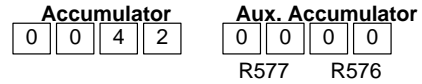
Here's how you would write the program to perform the Engineering Unit conversion. This example assumes you have calculated or loaded the engineering unit value and stored it in R400. Also, you have to perform this for all channels if you're using different data for each channel.

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

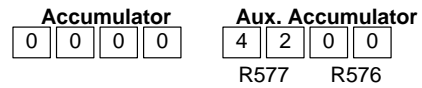
Scale the data



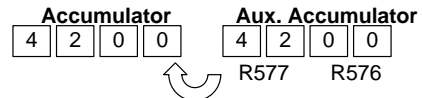
This instruction loads Engineering unit value into the accumulator on every scan.



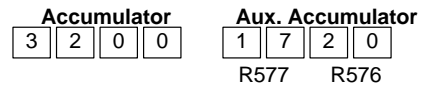
The Engineering unit value is divided by the Engineering unit range, which in this case is 100. ($100 - 0 = 100$)



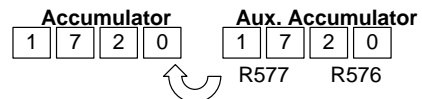
This instruction moves the two-byte decimal portion into the accumulator for further operations.



The accumulator is then multiplied by the module resolution, which is 4096. ($4096 \times 4200 = 17203200$). 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.



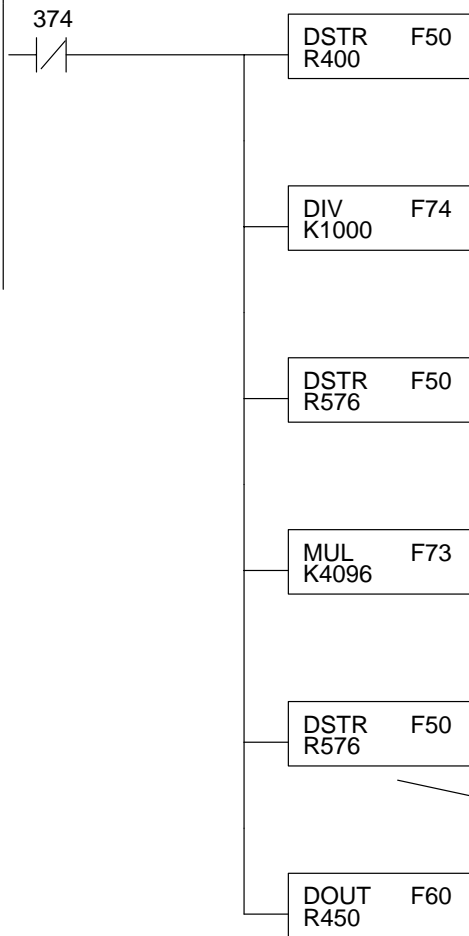
This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the digital value, which is 1720.



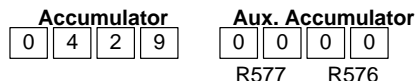
There will probably be times when you need more precise control. For example, maybe your application requires 42.9 PSI, not just 42 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve entered 429 as the Engineering unit value 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.

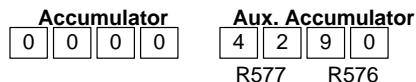
Scale the data



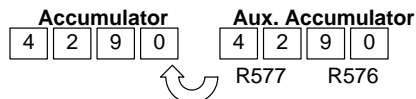
This instruction loads Engineering unit value into the accumulator on every scan.



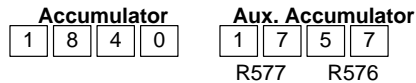
The Engineering unit value is divided by the Engineering unit range, which in this case is 1000. (100.0 implied range)



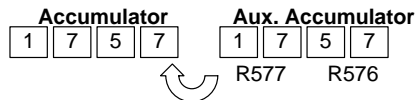
This instruction moves the two-byte decimal portion into the accumulator for further operations.



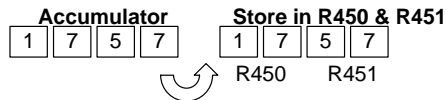
The accumulator is then multiplied by the module resolution, which is 4096. (4096 x 4290 = 17571840). Notice the most significant digits are now stored in the auxilliary accumulator. (This is different from the Divide instruction operation.)



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



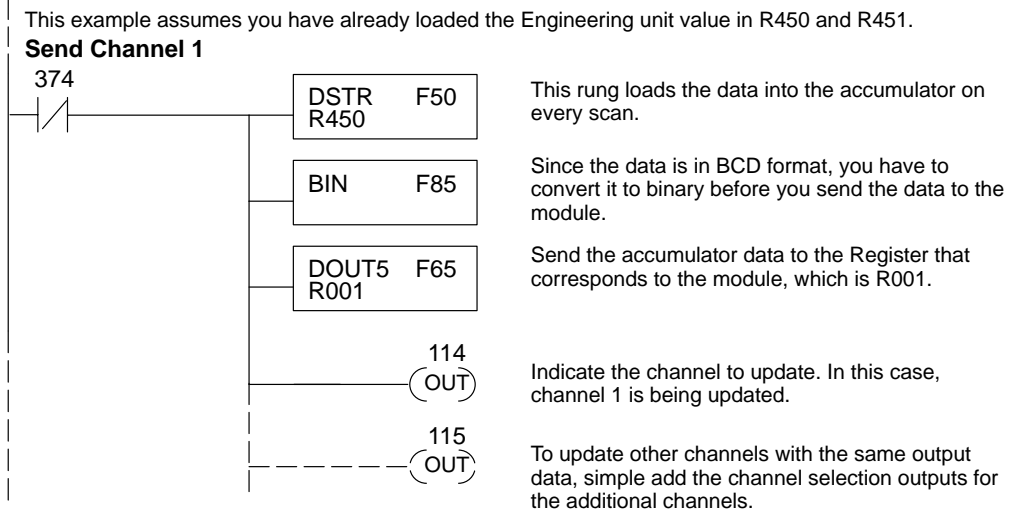
This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the digital value, which is 1757.



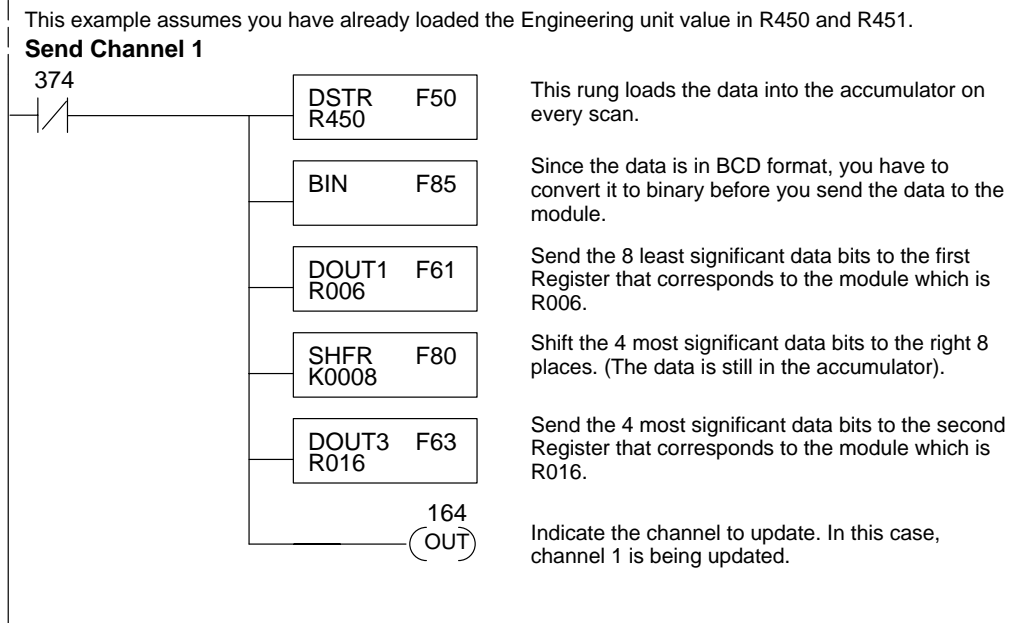
4 Ch. Isolated Analog Out. F3-04DAS

Sending Data to a Single Channel

The following program example shows how to send the digital value to a single channel.



If you install the F3-04DA-1 in the slot corresponding to registers 6 and 16, you have to make a slight program adjustment. This is because the DOUT5 instruction is not supported for this slot.

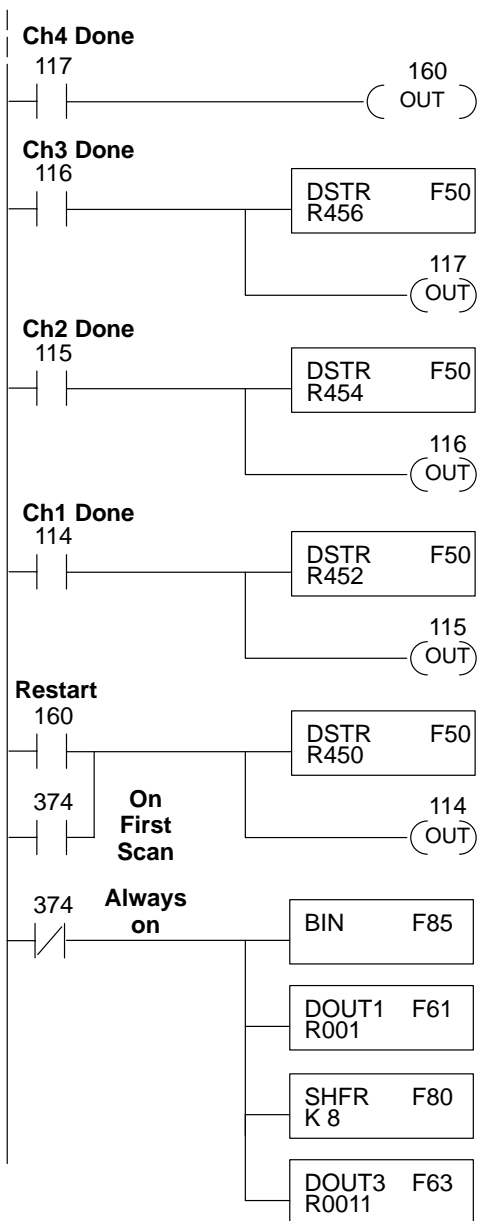


Sequencing the Channel Updates

This example shows how to send digital values to the module when you have more than one channel. This example will automatically update all four channels over four scans. The example is fairly simple and will work in most all situations, but there are instances where problems can occur. The logic must be active on the first CPU scan and all subsequent scans. If the logic gets stopped or disabled for some reason, there is no way to restart it. If you're using an RLL^{PLUS} (Stage) program, put this logic in an initial stage that is always active. Also, you should avoid using the this example if you require the analog output logic to be used inside a Master Control Relay field of control. You could also accidentally disable the analog output logic by inadvertently writing to the multiplexing control relays with an operator interface or intelligent module, such as an ASCII BASIC module, etc.

Sequencing Example

The following program example shows how to send the digital values to multiple channels. With this program, all channels will be updated within four scans. You must use the rungs in the order shown, but you can include them anywhere in the program.



When channel 4 has been updated, 160 restarts the update sequence.

When channel 3 has been updated, this rung loads the data for channel 4 into the accumulator. By turning on 117, this triggers the channel update. (Since 117 is also used as an input, this results in a one-shot.)

When channel 2 has been updated, this rung loads the data for channel 3 into the accumulator. By turning on 116, this triggers the channel update. (Since 116 is also used as an input, this results in a one-shot.)

When channel 1 has been updated, this rung loads the data for channel 2 into the accumulator. By turning on 115, this triggers the channel update. (Since 115 is also used as an input, this results in a one-shot.)

This rung loads the data for channel 1 into the accumulator. Since 374 is used, this rung automatically executes on the first scan. After that, 160 restarts this rung. If you examine the first rung, you'll notice 160 only comes on after channel 4 has been updated.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module. (You can omit this step if you've already converted the data elsewhere.)

Send the 8 least significant data bits to the first Register that corresponds to the module which is R001.

Shift the 4 most significant data bits to the right 8 places. (The data is still in the accumulator).

Send the 4 most significant data bits to the second Register that corresponds to the module which is R011.

F3-04DAS
4 Ch. Isolated Analog Out.

Writing the Control Program (DL350)

Reading Values: Pointer Method and Multiplexing

There are two methods of reading values:

- The pointer method (**all system bases must be D3-xx-1 to support the pointer method**)
- Multiplexing

You must use the multiplexing method with remote I/O modules (the pointer method will not work). You can use either method when using DL350 CPU, but for ease of programming it is strongly recommended that you use the pointer method.

Pointer Method

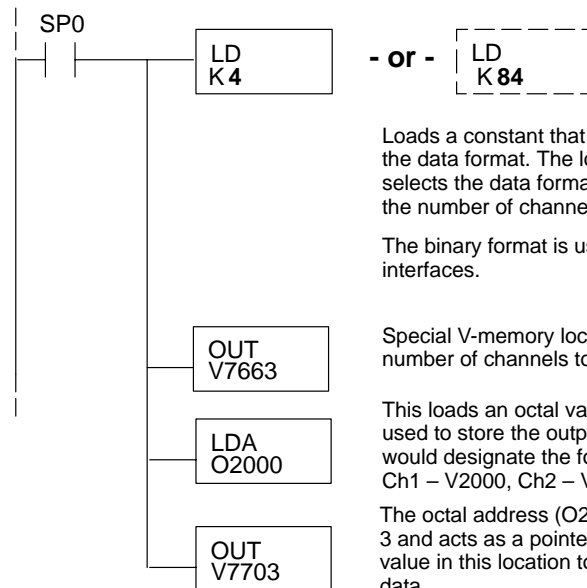
The DL350 has special V-memory locations assigned to each base slot that greatly simplifies the programming requirements. By using these V-memory locations you can:

- specify the number of channels to update.
- specify where to obtain the output data.



NOTE: Do not use the pointer method and the PID Control Output auto transfer to I/O module function together for the same module. If using PID loops, use the pointer method and ladder logic code to map the analog output data from the PID loop to the output module memory location(s).

The following program example shows how to set up these locations. Place this rung anywhere in the ladder program, or in the initial stage when using stage programming.



Loads a constant that specifies the number of channels to scan and the data format. The lower byte, most significant nibble (MSN) selects the data format (i.e. 0=BCD, 8=Binary), the LSN selects the number of channels (1 or 2).

The binary format is used for displaying data on some operator interfaces.

Special V-memory location assigned to slot 3 that contains the number of channels to scan.

This loads an octal value for the first V-memory location that will be used to store the output data. For example, the O2000 entered here would designate the following addresses.
Ch1 - V2000, Ch2 - V2001

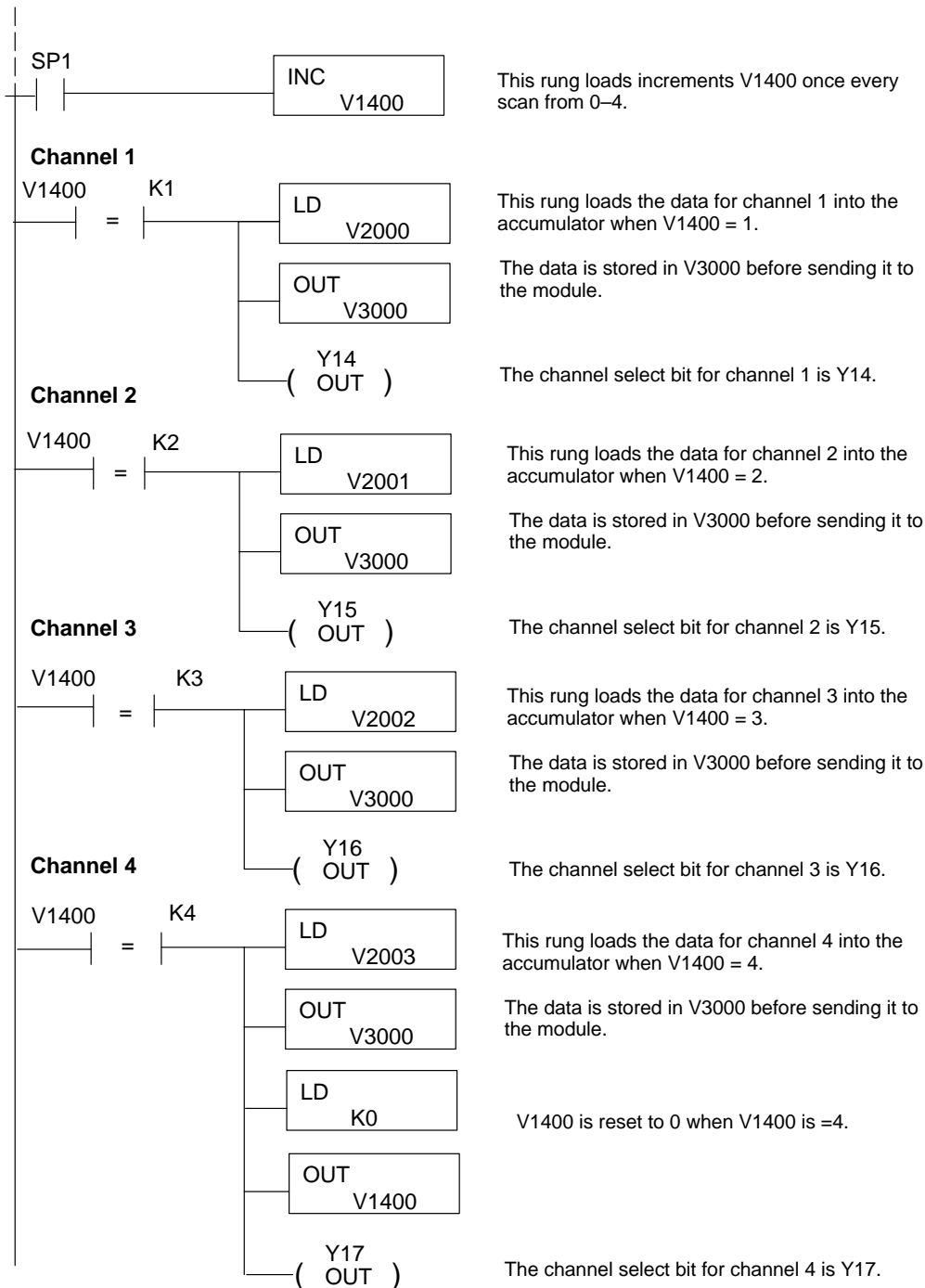
The octal address (O2000) is stored here. V7703 is assigned to slot 3 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the output data.

The table shows the special V-memory locations used with the DL350. Slot 0 (zero) is the module next to the CPU. Remember, the CPU only examines the pointer values at these locations after a mode transition. The pointer method is supported on expansion bases (all bases must be D3-xx-1) up to a total of 8 slots away from the DL350. The pointer method is not supported in slot 8 of a 10 slot base.

Analog Output Module Slot Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V7660	V7661	V7662	V7663	V7664	V7665	V7666	V7667
Storage Pointer	V7700	V7701	V7702	V7703	V7704	V7705	V7706	V7707

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

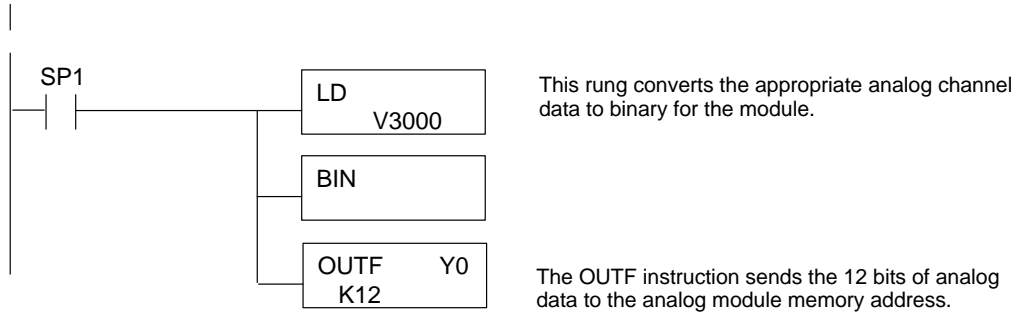
This example assumes the module is in Y0 address slot of a D3-xx-1. In this example V2000 contains the data for channel 1 and V2001 for channel 2, etc. in BCD. If any expansion bases are used in the system, they must all be D3-xx-1 to be able to use this example. Otherwise, the conventional base addressing must be used.



example program continued on next page

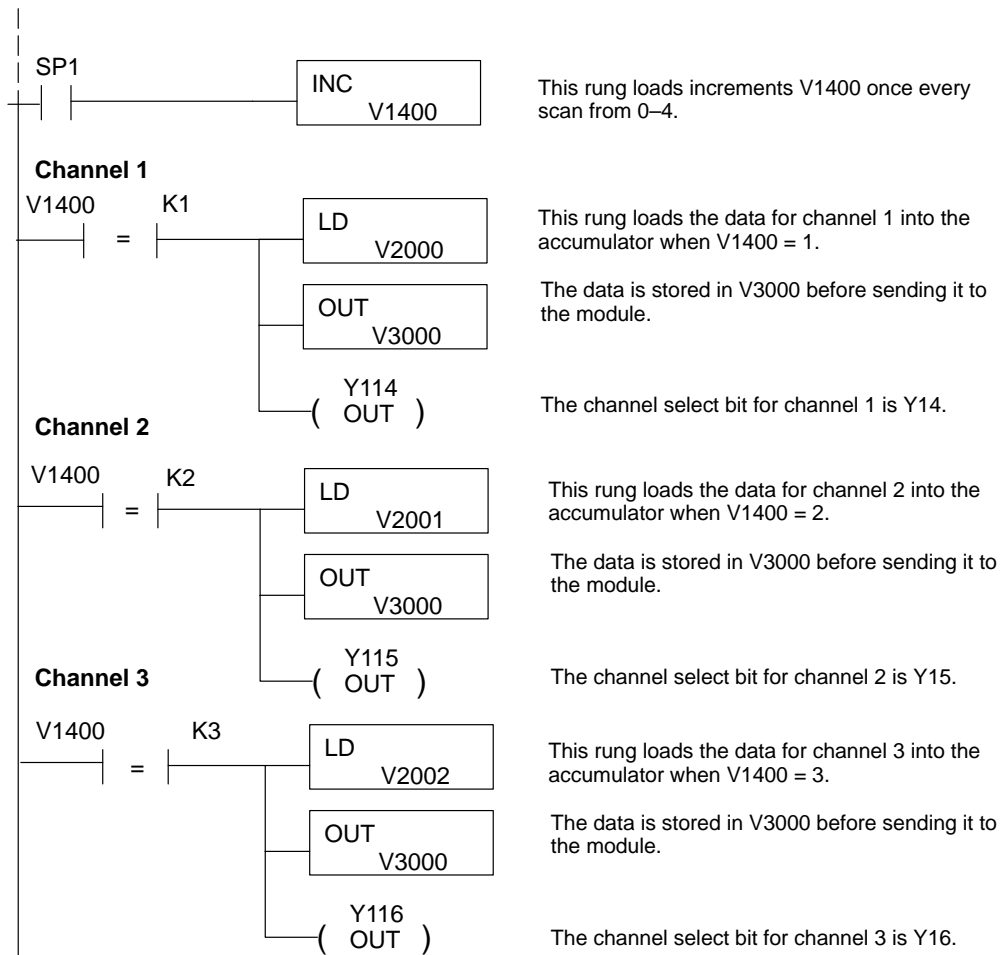
F3-04DAS
4 Ch. Isolated Analog Out.

example program continued from previous page



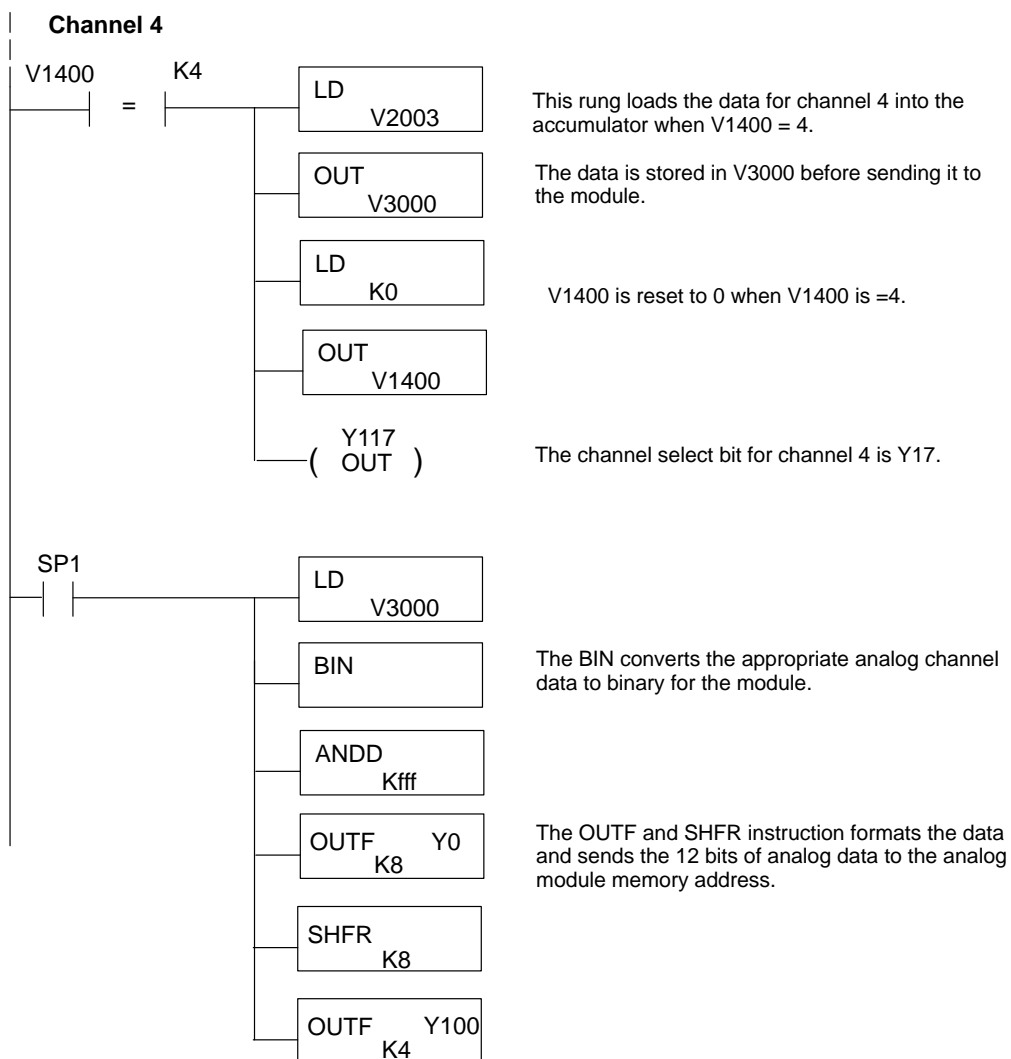
Multiplexing: DL350 with Conventional DL305 Base

This example assumes the module is in the Y0-10 / Y100-107 slot of a 305 conventional base. In this example V2000 contains the BCD data for channel 1 and V2001 contains the data for channel 2, etc. One more rung would be necessary for channel 4.



example program continued on next page

example program continued from previous page



F3-04DAS
4 Ch. Isolated Analog Out.

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.

$$A = U \frac{4095}{H - L}$$

A = Analog value (0 – 4095)

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.

$$A = 10U \frac{4095}{10(H - L)}$$

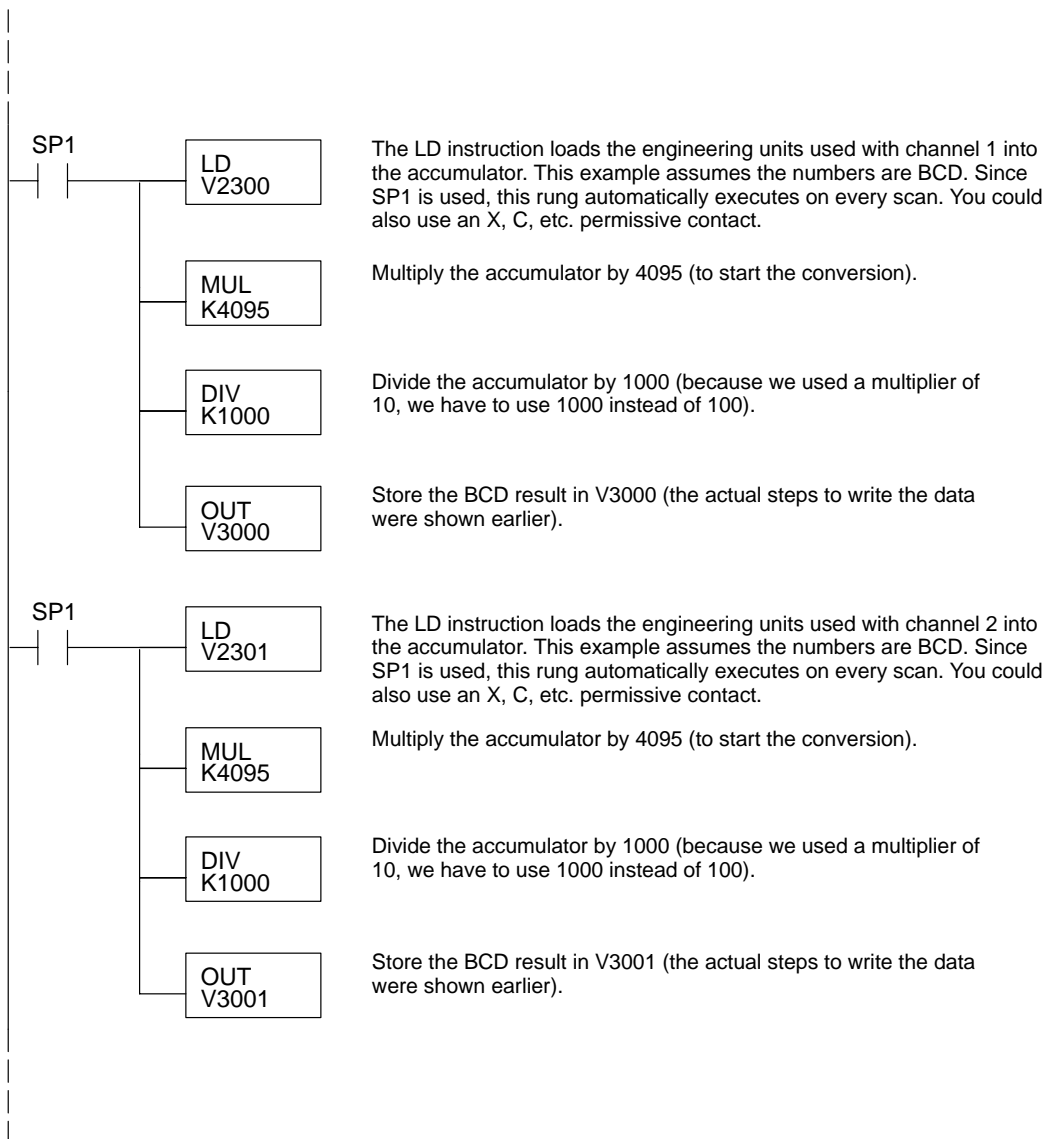
$$A = 494 \frac{4095}{1000 - 0}$$

$$A = 2023$$

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.



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 signal level ...
-10V to +10V	$A = \frac{20D}{4095} - 10$	$D = \frac{4095}{20}(A + 10)$
-5V to +5V	$A = \frac{10D}{4095} - 5$	$D = \frac{4095}{10}(A + 5)$
0 to 5V	$A = \frac{5D}{4095}$	$D = \frac{4095}{5} A$
0 to 10V	$A = \frac{10D}{4095}$	$D = \frac{4095}{10} A$
1 to 5V	$A = \frac{4D}{4095} + 1$	$D = \frac{4095}{4}(A - 1)$
4 to 20mA	$A = \frac{16D}{4095} + 4$	$D = \frac{4095}{16}(A - 4)$

For example, if you are using the -10 to +10V range and you have measured the signal at 6V, you would use the following formula to determine the digital value that should be stored in the register location that contains the data.

$$D = \frac{4095}{20}(A + 10)$$

$$D = \frac{4095}{20}(6V + 10)$$

$$D = (204.75)(16)$$

$$D = 3276$$