F2-02DA-2, F2-02DA-2L 2-Channel Analog Voltage Output

9

In This Chapter. . . .

Module Specifications

The F2-02DA-2 and F2-02DA-2L Analog Output modules provide several hardware features:

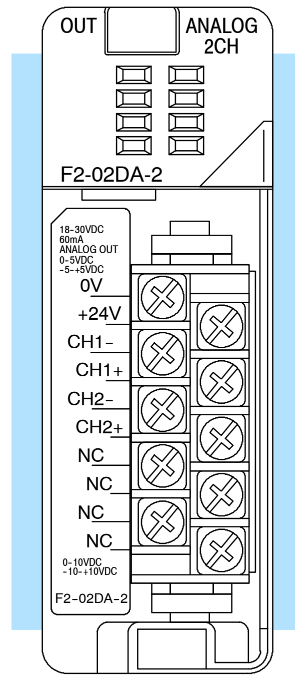
- Analog outputs are optically isolated from the PLC logic.
- The module has a removable terminal block, so the module can be easily removed or changed without disconnecting the wiring.
- With a DL240, DL250-1 or DL260 CPU, you can update both channels in one scan.
- F2-02DA-2: Low-power CMOS design requires less than 60mA from an external 18-30 VDC power supply.
- F2-02DA-2L: Low-power CMOS design requires less than 70mA from an external 10-15 VDC power supply.
- Outputs can be independently configured for any of these four ranges:

  1) 0 to 5 VDC

  2) 0 to 10 VDC

  3) ±5 VDC

  4) ±10 VDC



F2-02DA-2

NOTE: The F2-02DA-2 and F2-02DA-2L modules look very similar and it is very easy to mistake one module for the other. If your module does not work, check the terminal label to see if you have a 12 volts (L) or a 24 volts model and that it is being supplied with the proper input voltage.

The following tables provide the specifications for the F2-02DA-2 and F2-02DA-2L Analog Output Modules.

Output Specifications

Number of Channels	2
Output Ranges	0 to 5V, 0 to 10V, ±5V, ±10V
Resolution	12 bit (1 in 4096)
Output Type	Single ended, 1 common
Peak Output Voltage	15VDC (clamped by transient voltage suppressor)
Load Impedance	2000Ω minimum
Load Capacitance	.01µF maximum
Linearity Error (end to end)	±1 count (±0.025% of full scale) maximum
Conversion Settling Time	5 µs maximum (full scale change)
Full-Scale Calibration Error (offset error included)	±12 counts max. unipolar @ 25°C (77°F) ±16 counts max. bipolar @ 25°C (77°F)
Offset Calibration Error	±3 counts maximum @ 25°C (77°F) unipolar ±8 counts maximum @ 25°C (77°F) bipolar
Accuracy vs. Temperature	±50 ppm/°C full scale calibration change (including maximum offset change of 2 counts)
Maximum Inaccuracy	Unipolar ranges ±0.3% @ 25°C (77°F) ±0.45% 0-60°C (32-140°F) Bipolar ranges ±0.4% @ 25°C (77°F) ±0.55% 0-60°C (32-140°F)

General Specifications

PLC Update Rate	1 channel per scan maximum (D2-230 CPU) 2 channels per scan maximum (D2-240/250-1/260 CPU)
Digital Outputs Output Points Required	12 binary data bits, 2 channel ID bits, 1 sign bit 16 point (Y) output module
Power Budget Requirement	40 mA @ 5 VDC (supplied by base)
External Power Supply	F2-02DA-2: 18-30 VDC, 60 mA (outputs fully loaded) F2-02DA-2L: 10-15 VDC, 70 mA (outputs fully loaded)
Operating Temperature	0 to 60° C (32 to 140° F)
Storage Temperature	-20 to 70° C (-4 to 158° F)
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

One count in the specification table is equal to one least significant bit of the analog data value (1 in 4096).

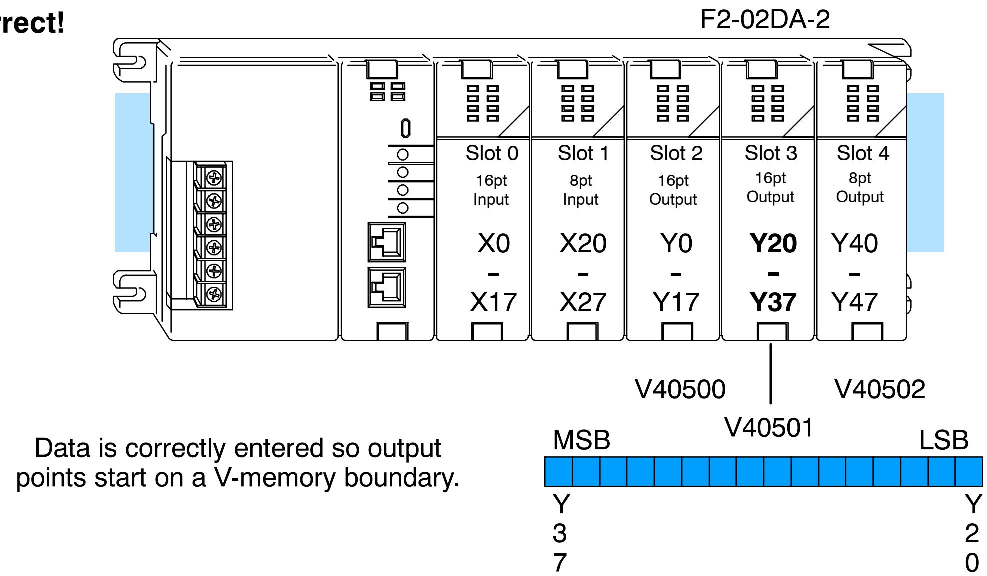
Analog Output Configuration Requirements

The F2-02DA-2 (L) analog output requires 16 discrete output points. The module can be installed in any slot of a DL205 system, but the available power budget and discrete I/O points can be limiting factors. Check the user manual for your particular model of CPU and I/O base for more information regarding power budget and number of local, local expanison or remote I/O points.

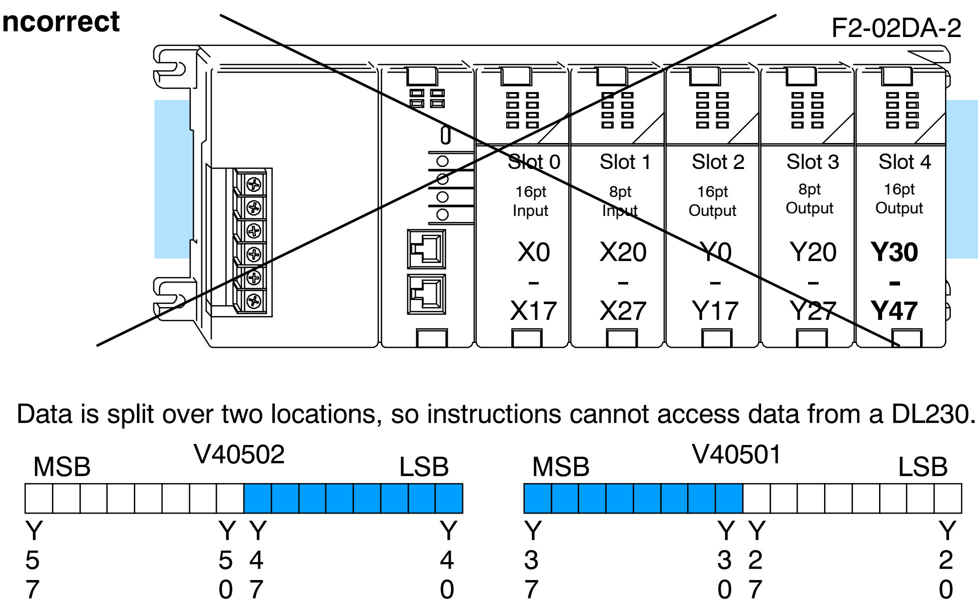
Special Placement Requirements (DL230 and Remote I/O Bases)

Even though the module can be placed in any slot, it is important to examine the configuration if you are using a DL230 CPU. As you can see in the section on writing the program, you use V-memory locations to extract the analog data. If you place the module so the output points do not start on a V-memory boundary, the instructions cannot access the data. This also applies when placing this module in a remote I/O base using a D2-RSSS in the CPU slot.

Correct!



Incorrect



To use the V-memory references required for a DL230 CPU, the *first* output address assigned to the module must be one of the following Y locations. The table also shows the V-memory addresses that correspond to these Y locations.

Y	Y0	Y20	Y40	Y60	Y100	Y120	Y140	Y160
V	V40500	V40501	V40502	V40503	V40504	V40505	V40506	V40507

Setting the Module Jumpers

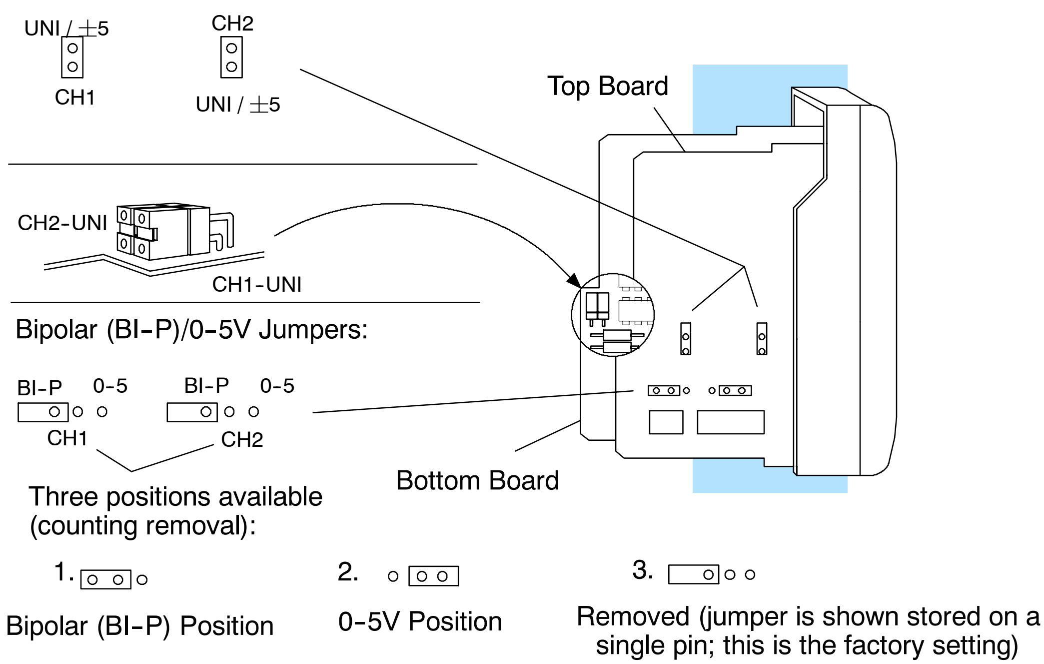
The F2-02DA-2 (L) Analog Output module uses jumpers for selecting the voltage ranges for each channel. The range of each channel can be independently set. Available operating ranges are 0–5V, 0–10V, ±5V, and ±10V.

There are three jumpers for each channel. Two sets are on the top board, and the third set is along the edge of the bottom board *with* the black D-shell backplane connector. Install or remove these jumpers to select the desired range. Unused jumpers can be stored on a single pin so they do not get lost.

- Two of the top board jumpers are labeled "UNI / ±5" and there is one for each channel.
- The two bottom board jumpers are labeled "UNI" and there is one for each channel. These jumpers determine the format of the channel output data, and the effect of their settings is independent from that of the other jumpers on the module. With a UNI jumper removed, the corresponding channel requires data values in the range of ±2047. With a UNI jumper installed, the channel requires data values in the range of 0 to 4095.
- The other two top board jumpers are labeled "BI-P 0-5" and there is one for each channel. These jumpers each have three possible settings (including jumper removed) since there are three pins.

NOTE: It is important to set the module jumpers correctly. The module will not operate correctly if the jumpers are not properly set for the desired voltage range.

This figure shows the jumper locations. See the table on the following page to determine the proper settings for your application.



Voltage Range and Output Combinations

The table lists the eight possible combinations of voltage ranges and data formats, along with the corresponding jumper settings. For most applications, use one of the four standard selections shown in the shaded blocks in the table. Standard unipolar voltage ranges accept a data format of 0 to 4095. Standard bipolar ranges accept a data format of -2047 to +2047.

Voltage Range	Output Data Format	UNI / ± 5V Jumpers Settings (top board)	UNI Output Format Jumpers Settings (bottom board)	BI-P 0-5V Jumpers Settings (top board)	
				BI-P (Bipolar) Position	0-5V Position
0 to 5V	0 to 4095	Install	Install		Install here
0 to 10V	0 to 4095	Install	Install	Completely remove	
0 to 5V	±2047	Install	Remove		Install here
0 to 10V	±2047	Install	Remove	Completely remove	
±5V	±2047	Install	Remove	Install here	
±10V	±2047	Remove	Remove	Install here	
±5V	0 to 4095	Install	Install	Install here	
±10V	0 to 4095	Remove	Install	Install here	

Standard selections are shown in shaded cells in the table.

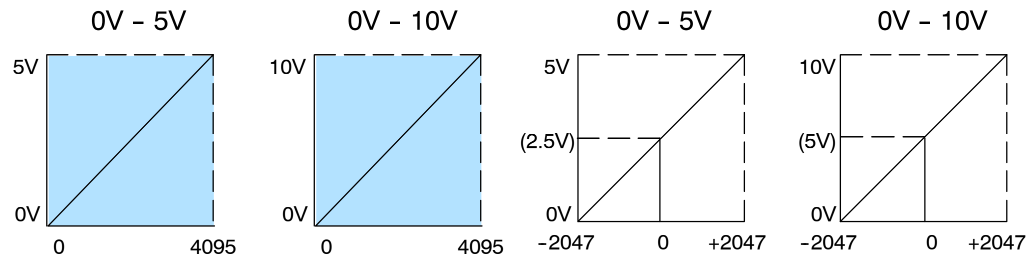
For example, to select settings of "±5V" voltage range with a "±2047" output data format for channel 1, refer to the table above and the figure on the previous page and arrange the jumpers as follows:

- Install the "CH1" "UNI / ±5V" jumper.
- Remove the "CH1-UNI" jumper. Store the jumper so it does not get lost by placing it on one pin.
- Install the "CH1" "BI-P 0-5" jumper in the BI-P (bipolar) position on the left pin and center pin.

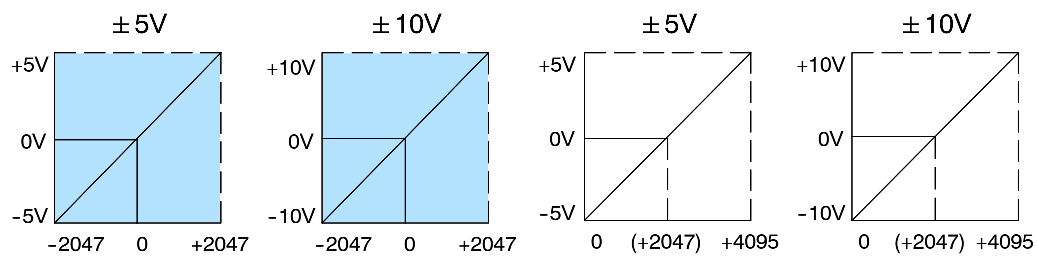
The non-standard selections in the table provide the opposite data format for both unipolar and bipolar voltage ranges. If you are using unipolar output (0-5V or 0-10V) on one channel and bipolar output (±5V, ±10V) on the other channel, then one of the outputs will use a non-standard data format.

The graphs show the voltage range to output data format relationship for each of the eight selections.

Unipolar Ranges



Bipolar Ranges



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 signal source. *Do not* ground the shield at both the module and the load.
- Do not 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 F2-02DA-2 (L) requires a separate field-side power supply. Each module requires 18–30 VDC at up to 60mA current. The DL205 bases have built-in 24 VDC power supplies that provide up to 300mA of current. If you are using only a couple of analog modules, you can use this power source instead of a separate supply. If you want to use a separate supply, choose one that meets the power requirements of your application.

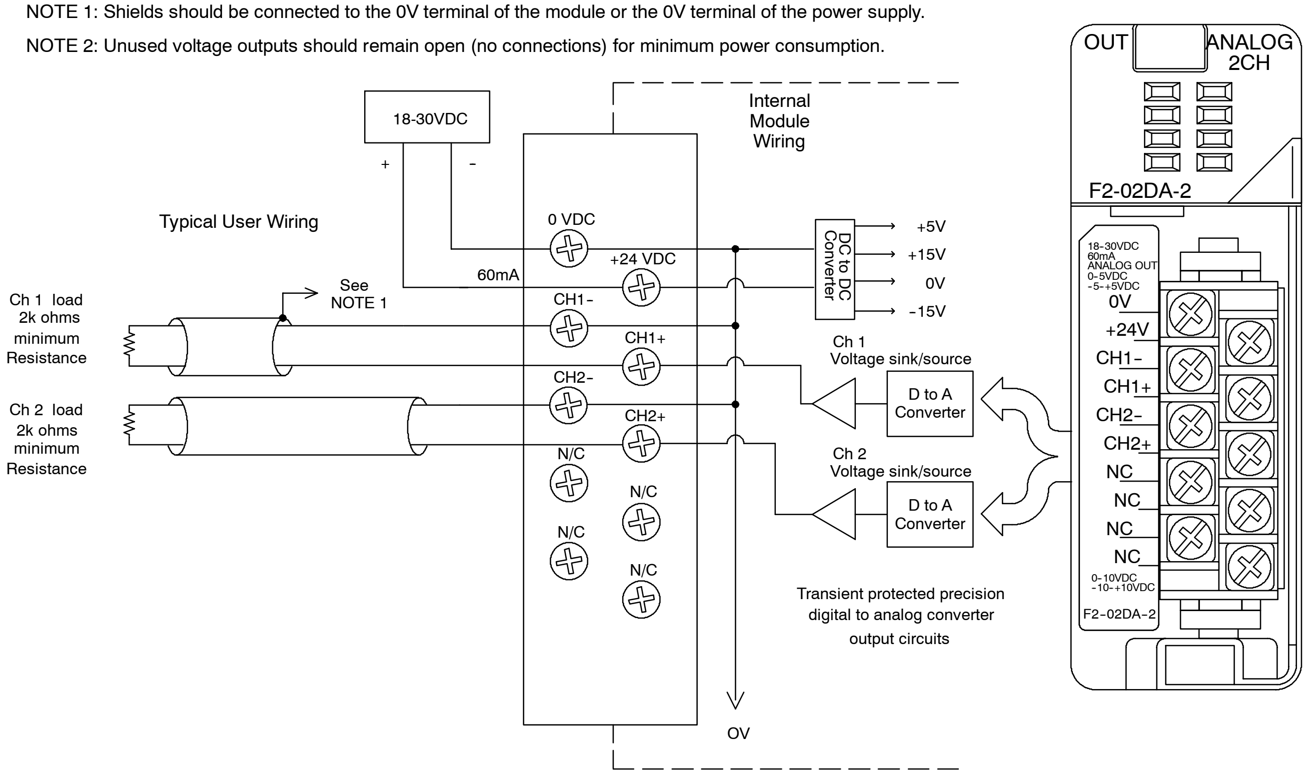
WARNING: If you are using the 24 VDC base power supply, make sure you calculate the power budget. Exceeding the power budget can cause unpredictable system operation that can lead to a risk of personal injury or damage to equipment.

Wiring Diagram

The F2-02DA-2 (L) module has a removable connector to make wiring easier. Simply remove the retaining screws and gently pull the connector from the module. Use the following diagram to connect the field wiring.

NOTE 1: Shields should be connected to the 0V terminal of the module or the 0V terminal of the power supply.

NOTE 2: Unused voltage outputs should remain open (no connections) for minimum power consumption.

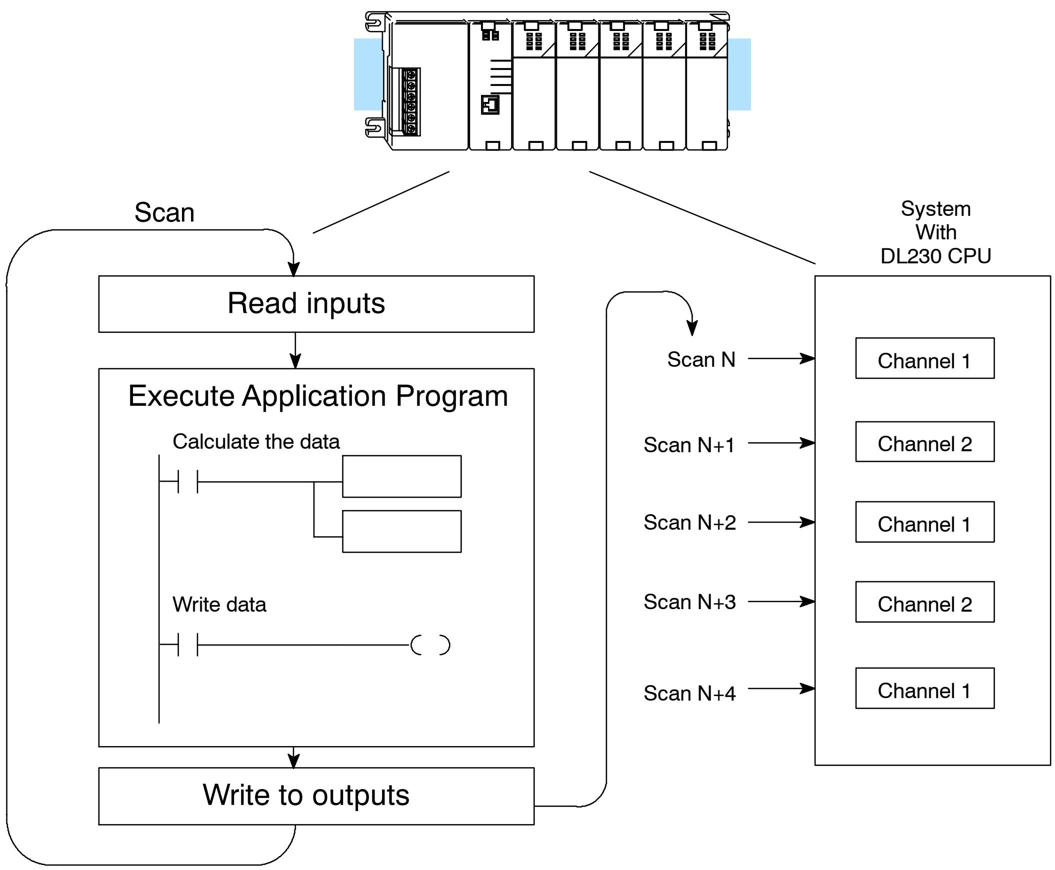


Module Operation

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.

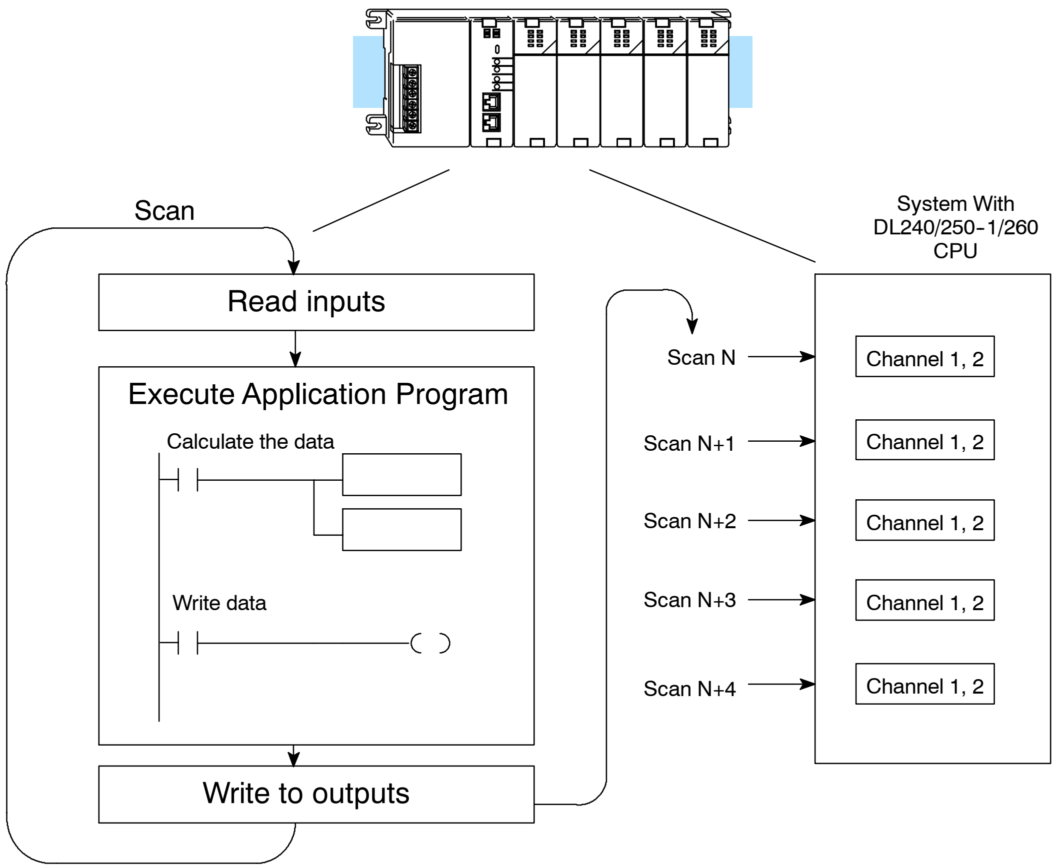
Channel Update Sequence for a DL230 CPU (Multiplexing)

If you are using a DL230 CPU, you can send one channel of data to the output module on each scan. The module refreshes both field devices on each scan, but you can only get new data from the CPU at the rate of one channel per scan. Since there are two channels, it can take two scans to update both channels. However, if you are only using one channel, you can update that channel on every scan. The multiplexing method can also be used for DL240/250-1/260 CPUs.



Channel Update Sequence for a DL240, DL250-1 or DL260 CPU (Pointer Method)

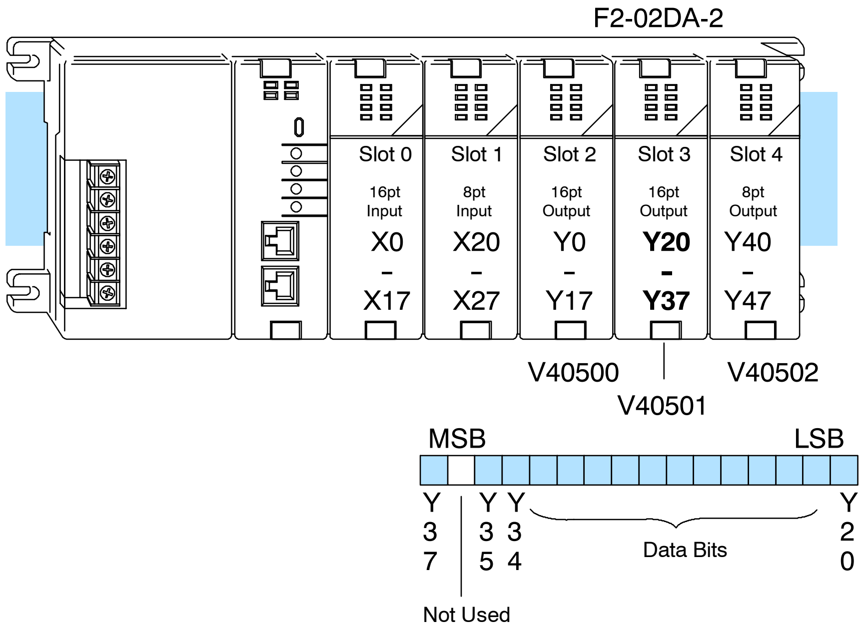
If you are using a DL240, DL250-1 or DL260 CPU, you can update both channels on every scan. This is because the DL240/250-1/260 CPU supports special V-memory locations that are used to manage the data transfer. This is discussed in more detail in the section on Writing the Control Program.



Understanding the Output Assignments

You may recall the F2-02DA-2 (L) module requires 16 discrete output points in the CPU. These points provide the data value and an indication of which channel to update. Note, if you are using a DL240/250–1/260 CPU, you may never have to use these bits, but it may help you understand the data format.

Since all output points are automatically mapped into V-memory, it is very easy to determine the location of the data word that will be assigned to the module.

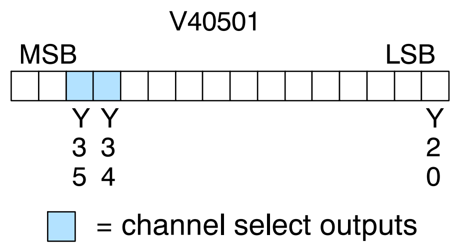


Within this word location, the individual bits represent specific information about the analog signal.

Channel Select Outputs

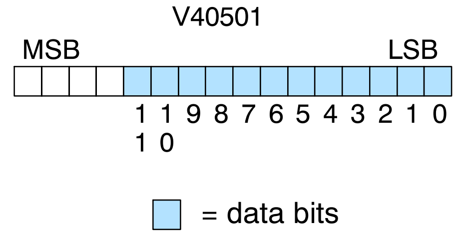
Two of the outputs select the active channel. Remember, the V-memory bits are mapped directly to discrete outputs. Turning a bit OFF selects its channel. By controlling these outputs, you can select which channel(s) gets updated.

Y35	Y34	Channel
On	Off	1
Off	On	2
Off	Off	1 & 2 (same data to both channels)
On	On	none (both channels hold current values)



Analog Data Bits The first twelve bits represent the analog data in binary format.

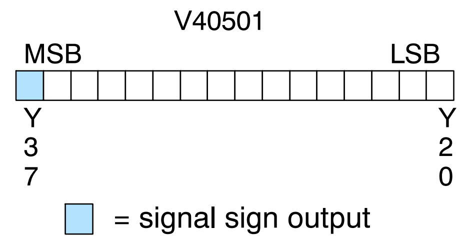
Bit	Value	Bit	Value
0	1	6	64
1	2	7	128
2	4	8	256
3	8	9	512
4	16	10	1024
5	32	11	2048



Signal Sign Output

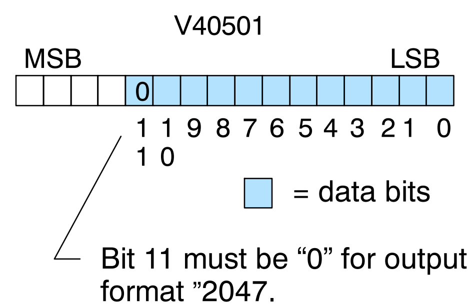
The last output can be used to select the signal sign (+ or –) for bipolar ranges. By controlling this output, you can easily select positive or negative data values.

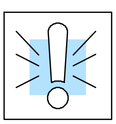
Programming examples in the next section show how easy it is to make the sign selection part of your data value.



Bipolar Output Data

If an output channel is configured for an output format of 0 – 2047, the maximum valid value for the lower 12 bits is 2047. This means the 12'th bit (bit 11) *must* always be "0".



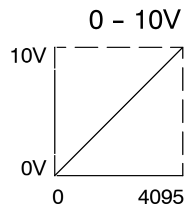


WARNING: If the data value exceeds 2047, the 12th bit becomes a "1", and the other eleven bits start over at "00000000000". At this point the module's channel output voltage also goes *back to the bottom of its range* and begins increasing again. The RLL program will be expecting a maximum output, but it will be minimum instead. This can have serious consequences in some applications, and may result in personal injury or damage to equipment. Therefore, in standard bipolar ranges (or whenever the output format is ±2047 in general), be sure that your RLL program does not create numbers with absolute values greater than 2047.

Module Resolution

Since the module has 12-bit resolution, the analog signal is converted from 4096 counts ranging from 0-4095 (2^{12}). For example, with a 0 to 10V range, send a 0 to get a 0V signal, and send 4095 to get a 10V signal. This is equivalent to a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF hexadecimal.

Each count can also be expressed in terms of the signal level by using the following equation:



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

H = high limit of the signal range

L = low limit of the signal range

The following table shows the smallest change in signal level due to a digital value change of 1 LSB count.

Voltage Range	Signal Span	Divide By	Smallest Output Change
0 to 5V	5 volts	4095	1.22 mV
0 to 10V	10 volts	4095	2.44 mV
±5V	10 volts	4095	2.44 mV
±10V	20 volts	4095	4.88 mV

Writing the Control Program

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 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.

$$A = U \frac{4095}{H - L}$$ for 0–4095 output format

$$A = U \frac{2047}{H - L}$$ for 0–2047 output format

A = Analog value (0 – 4095)

U = Engineering units

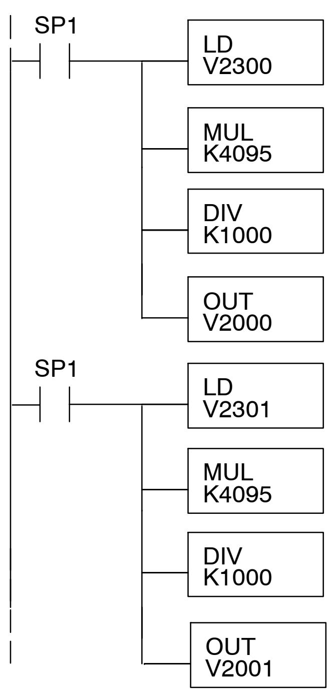
H = High limit of the engineering unit range

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 must adjust the formula to compensate for it.

$$A = 10U \frac{4095}{10(H - L)} \qquad A = 494 \frac{4095}{1000 - 0} \qquad A = 2023$$

The following example program shows how you would write the program to perform the engineering unit conversion to output data formats 0–4095. This example assumes you have calculated or loaded the engineering unit values in BCD format and stored them in V2300 and V2301 for channels 1 and 2 respectively. The DL205 offers instructions that allow you to perform math operations using BCD format. It is usually easier to perform any math calculations in BCD and then convert the value to binary before you send the data to the module.



The LD instruction loads the engineering units used with channel 1 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 4095 (to start the conversion).

Divide the accumulator by 1000 (because we used a multiplier of 10, we have to use 1000 instead of 100).

Store the BCD result in V2000 (the actual steps required to send the data are shown later).

The LD instruction loads the engineering units used with channel 2 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 4095 (to start the conversion).

Divide the accumulator by 1000 (because we used a multiplier of 10, we have to use 1000 instead of 100).

Store the BCD result in V2001 (the actual steps required to send the data are shown later).

Negative Values with Bipolar Range

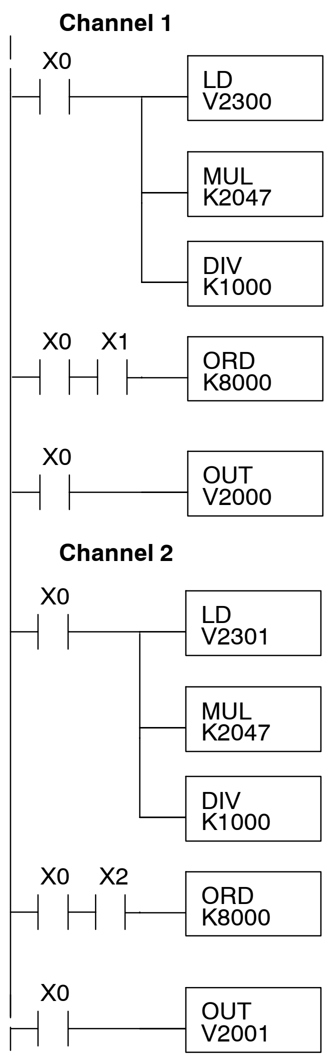
If you are using the bipolar ranges (±5V, ±10V) or an output data format of ±2047, you also need to specify whether the value is positive or negative. There are two ways to show that the value is negative:

- Turn on the sign output (Y37 in the examples, DL230 only).
- Embed the sign output in the data value (required for the DL240/250–1/260 using the pointer method, an optional method for the DL230).

To embed the sign output in the data values, just OR 8000 to the value. This has the same effect as turning on Y37. Remember, the V-memory location is mapped directly to the outputs.

If you are going to use bipolar ranges, you also need to add logic to handle the positive and negative values. The logic would be similar for both values, but you should use some type of permissive contact to select the appropriate section of logic. Here is an example that re-scales a variable from a 0–1000 range to a 0–2047 range. It includes a step that combines 8000 with the value to make it negative.

NOTE: Do not exceed a value of 2047 for ±2047 output formats.



The LD instruction loads the engineering units used with Channel 1 into the accumulator. This example assumes the numbers are BCD. Since X0 is used, this rung only executes when X0 is on (X1 would be the input that would indicate a negative value should be used).

Multiply the accumulator by 2047 (to start the conversion).

Divide the accumulator by 1000 (because we used a multiplier of 10, we have to use 1000 instead of 100).

This ORD instruction imbeds the sign output in the data value when X0 and X1 are on. It combines the BCD value (8000) with the accumulator value to make it negative. Omit this rung if you choose to control the sign bit of the module (Y37) directly.

Store the result in V2000. This is the digital value, in BCD form, that should be sent to the module (the actual steps required to send the data are shown later).

The LD instruction loads the engineering units used with Channel 2 into the accumulator. This example assumes the numbers are BCD. Since X0 is used, this rung only executes when X0 is on (X2 would be the input that would indicate a negative value should be used).

Multiply the accumulator by 2047 (to start the conversion).

Divide the accumulator by 1000 (because we used a multiplier of 10, we have to use 1000 instead of 100).

This ORD instruction imbeds the sign output in the data value when X0 and X1 are on. It combines the BCD value (8000) with the accumulator value to make it negative. Omit this rung if you choose to control the sign bit of the module (Y37) directly.

Store the result in V2001. This is the digital value, in BCD form, that should be sent to the module (the actual steps required to send the data are shown later).

Writing Values: Pointer Method and Multiplexing

There are two methods of reading values:

- The pointer method
- Multiplexing

You *must* use the multiplexing method when using a DL230 CPU. You must also use the multiplexing method with remote I/O modules (the pointer method will not work). You can use either method when using DL240, DL250–1 and DL260 CPUs, but for ease of programming it is strongly recommended that you use the pointer method.

Writing Values (Pointer Method)

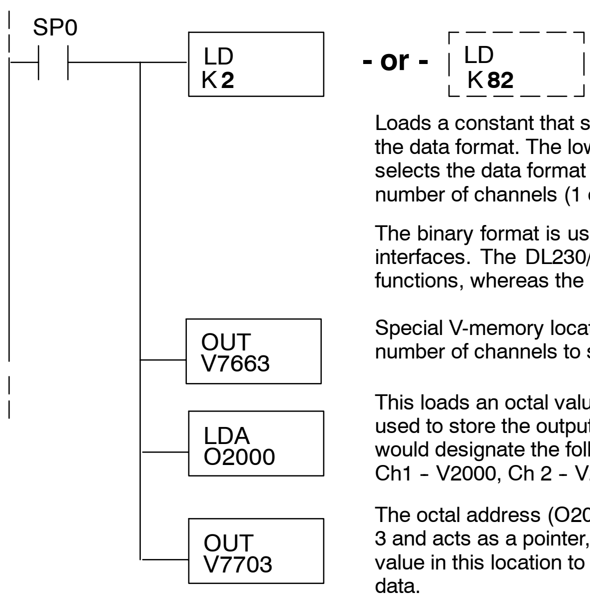
230	240	250–1	260
✗	✓	✓	✓

Once you have calculated the data values (shown previously) you must enter the program that actually updates the module. The DL240/250–1/260 has special V-memory locations assigned to each base slot that greatly simplify 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: DL240 CPUs with firmware release 1.5 or later supports this method. DL250 CPUs with firmware release version 1.06 or later support this method. If you must use the DL230 example, module placement in the base is very important. Review the section earlier in this chapter for guidelines.

The following program example shows how to setup these locations. Place this rung anywhere in the ladder program, or in the initial stage if you are using stage programming. You may recall in the previous example we used V2000 and V2001 to store the calculated values. Also, in the previous examples we had the analog module installed in slot 3. You should use the appropriate memory locations for your application. The pointer method automatically converts values to BCD.



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 (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. The DL230/240 CPUs do not support binary math functions, whereas the DL250 does.

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, Ch 2 – 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 tables below show the special V-memory locations used by the DL240, DL250-1 and DL260 for the CPU base and local expansion base I/O slots. Slot 0 (zero) is the module next to the CPU or D2-CM module. Slot 1 is the module two places from the CPU or D2-CM, and so on. Remember, the CPU only examines the pointer values at these locations after a mode transition. Also, if you use the DL230 (multiplexing) method, verify that these addresses in the CPU are zero.

The Table below applies to the DL240, DL250-1 and DL260 CPU base.

CPU 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

The Table below applies to the DL250-1 or DL260 expansion base 1.

Expansion Base D2-CM #1: Analog Output Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36000	V36001	V36002	V36003	V36004	V36005	V36006	V36007
Storage Pointer	V36020	V36021	V36022	V36023	V36024	V36025	V36026	V36027

The Table below applies to the DL250-1 or DL260 expansion base 2.

Expansion Base D2-CM #2: Analog Output Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36100	V36101	V36102	V36103	V36104	V36105	V36106	V36107
Storage Pointer	V36120	V36121	V36122	V36123	V36124	V36125	V36126	V36127

The Table below applies to the DL260 CPU expansion base 3.

Expansion Base D2-CM #3: Analog Output Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36200	V36201	V36202	V36203	V36204	V36205	V36206	V36207
Storage Pointer	V36220	V36221	V36222	V36223	V36224	V36225	V36226	V36227

The Table below applies to the DL260 CPU expansion base 4.

Expansion Base D2-CM #4: Analog Output Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36300	V36301	V36302	V36303	V36304	V36305	V36306	V36307
Storage Pointer	V36320	V36321	V36322	V36323	V36324	V36325	V36326	V36327

Writing Data (Multiplexing)

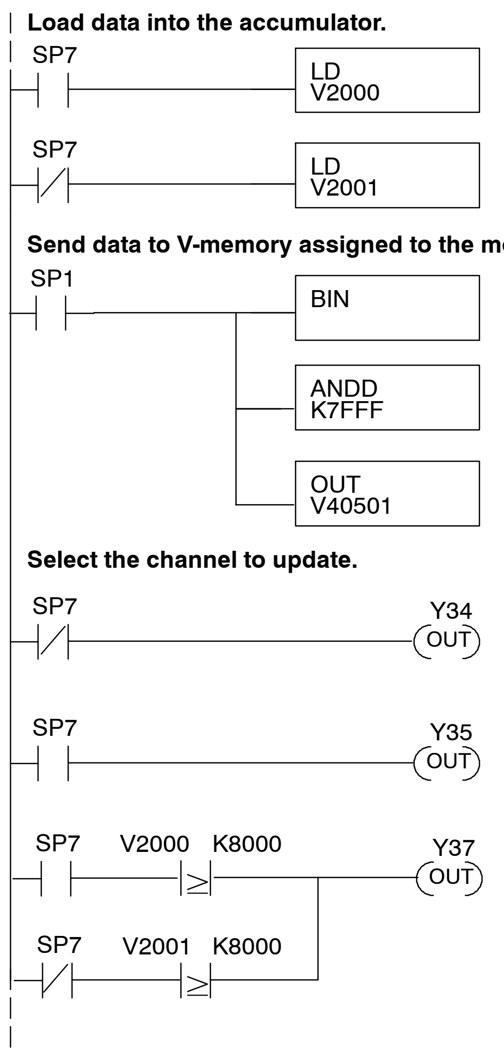
✓ 230 ✓ 240 ✓ 250–1 ✓ 260

The DL230 CPU *does not* have the special V-memory locations that allow you to automatically enable the data transfer. Since all channels are multiplexed into a single data word, the control program must be set up to determine which channel to write. Since the module appears as Y output points to the CPU, it is very easy to use the channel selection outputs to determine which channel to update.

Note, this example is for a module installed as shown in the previous examples. The addresses used would be different if the module was used in a different I/O arrangement. You can place these rungs anywhere in the program or if you are using stage programming, place them in a stage that is always active.

This example is a two-channel multiplexer that updates each channel on alternate scans. SP7 is a special relay that is on for one scan then off for one scan. A permissive contact on the last rung handles an embedded sign bit.

NOTE: You must send binary data to the module. If the data is already in binary format, you should not use the BIN instruction shown in this example.



Loads the data for channel 1 into the accumulator.

Loads the data for channel 2 into the accumulator.

Convert the data to binary (you must omit this step if you have converted the data elsewhere).

Remove sign bit for BCD to binary conversion. SP1 is always on.

The OUT instruction sends the data to the module. Our example starts with V40501 but the actual value depends on the location of the module in your application.

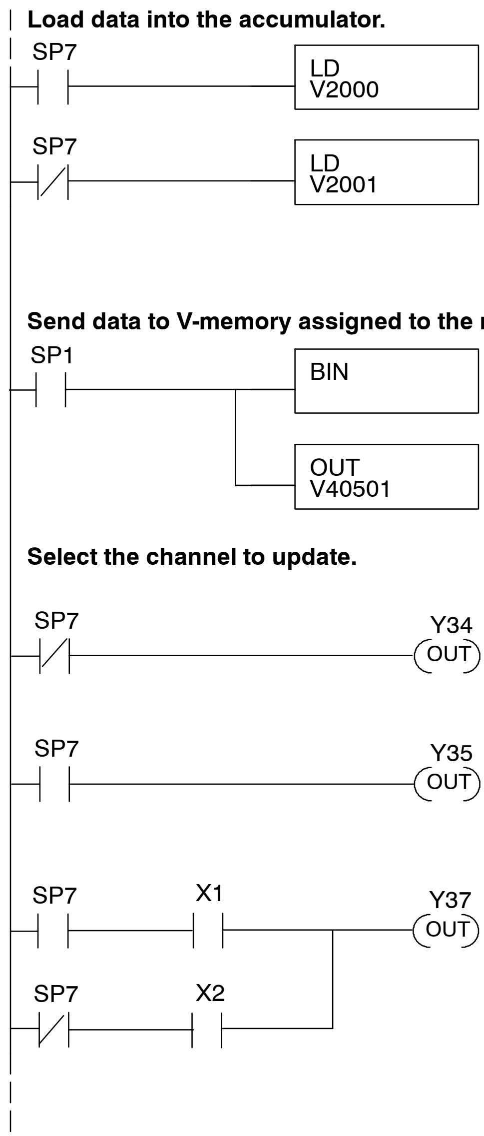
Selects channel 1 for update when Y34 is OFF (Y35–ON deselects channel 2). Note, Y34 and Y35 are used due to the previous examples. If the module was installed in a different I/O arrangement, the addresses would be different.

Selects channel 2 for update when Y35 is OFF (Y34–ON deselects channel 1). Note, Y34 and Y35 are used due to the previous examples. If the module was installed in a different I/O arrangement, the addresses would be different.

If the output format is –2047 to +2047, include this rung to embed the sign bit. For the output format 0 to 4095, omit this rung.

If you are using an output format range of ±2047 (most commonly used with bipolar voltage ranges), you also must specify whether the values are positive or negative. You could use the previous example with a simple addition to activate the sign output bit, or the following example uses individual contacts to determine the sign bit status for each channel.

NOTE: If you embed the sign information into the data value (by adding 8000 to the data value) you should not use this method. Use the previous example.



Loads the data for channel 1 into the accumulator.

Loads the data for channel 2 into the accumulator.

Convert the data to binary (you must omit this step if you have converted the data to binary).
SP1 is always on.

The OUT instruction sends the data to the module. Our example starts with V40501, but the actual value depends on the location of the module in your application.

Selects channel 1 for update when Y34 is OFF (Y35-ON deselects channel 2). Note, Y34 and Y35 are used due to the previous examples. If the module was installed in a different I/O arrangement, the addresses would be different.

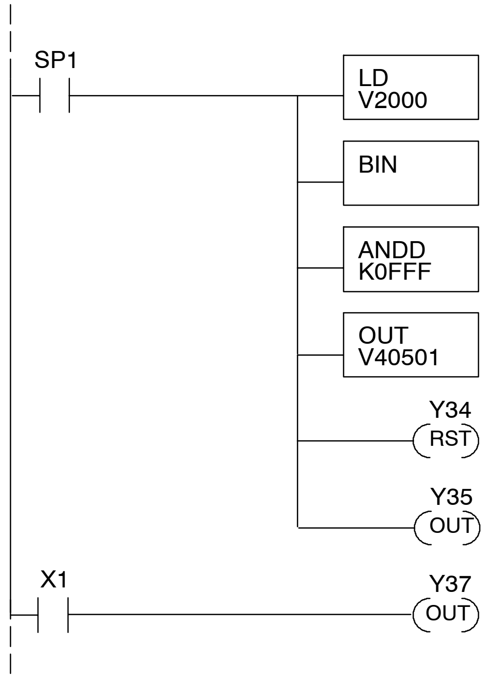
Selects channel 2 for update when Y35 is OFF (Y34-ON deselects channel 1). Note, Y34 and Y35 are used due to the previous examples. If the module was installed in a different I/O arrangement, the addresses would be different.

The permissive X1 activates Y37 (sign bit) during a channel 1 update scan. The permissive X2 activates Y37 during a channel 2 update scan. The sign bit (Y37 ON) indicates that the value is negative. You could use another permissive, such as a CR, etc.

NOTE: Do not exceed a value of 2047 for ±2047 output data formats.

Sending Data to One Channel

If you are not using both channels, or if you want to control the updates separately, use the following program. Remember, for bipolar ranges you either have to embed the sign information or use the sign output bit.



The LD instruction loads the data into the accumulator. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

The BIN instruction converts the accumulator data to binary (you must omit this step if you have already converted the data elsewhere).

The ANDD instruction masks off the channel select bits to prevent an accidental channel selection.

The OUT instruction sends the data to the module. Our example starts with V40501, but the actual value depends on the location of the module in your application.

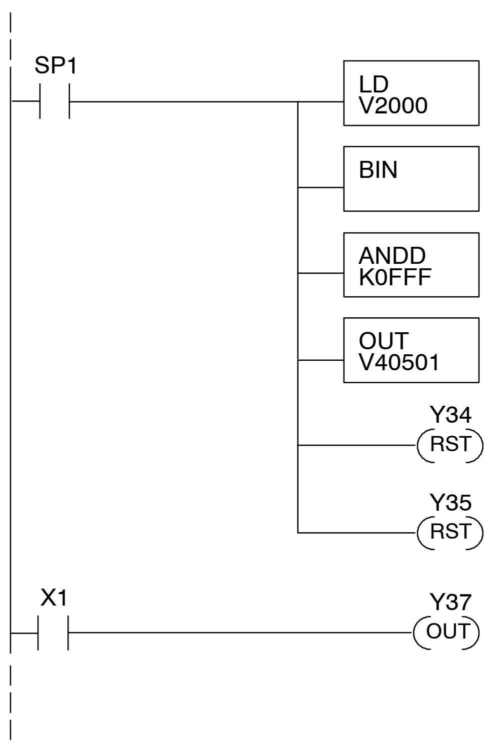
Y34-OFF selects channel 1 for updating.

Y35-ON deselects channel 2 (do not update).

The permissive X1 activates Y37, which is the sign bit. The sign bit indicates that the value is negative. You could use another permissive, such as a CR, etc. Omit this rung if you are using the 0 to +4095 output format.

Sending the Same Data to Both Channels

If both channel selection outputs are off, both channels will be updated with the same data. Remember, for bipolar ranges you either have to embed the sign information or use the sign output bit.



The LD instruction loads the data into the accumulator. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

The BIN instruction converts the accumulator data to binary (you must omit this step if you have already converted the data elsewhere).

The ANDD instruction masks off the channel select bits to prevent an accidental channel selection.

The OUT instruction sends the data to the module. Our example starts with V40501, but the actual value depends on the location of the module in your application.

Y34-OFF selects channel 1 for updating.

Y35-OFF selects channel 2 for updating.

The permissive X1 activates Y37, which is the sign bit. The sign bit indicates that the value is negative. You could use another permissive, such as a CR, etc. Omit this rung if you are using the 0 to +4095 output format.

Analog and Digital Value Conversions

Sometimes it is useful to be able to quickly convert between the 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. Remember, if you embed the sign information into the data value, you must adjust the formulas accordingly.

Range	If you know the digital value ...	If you know the signal level ...
0 to 10V	$A = \frac{10D}{4095}$	$D = \frac{4095}{10}(A)$
±10V (output format ±2047)	$A = \frac{10D}{2047}$	$D = \frac{2047}{10}(A)$
0 to 5V	$A = \frac{5D}{4095}$	$D = \frac{4095}{5}(A)$
±5V (output format ±2047)	$A = \frac{5D}{2047}$	$D = \frac{2047}{5}(A)$

For example, if you are using the ±10V range with an output format of ±2047, and you know you need a 6V signal level, use this formula to determine the digital value (D) that will be stored in the V-memory location that contains the data.

$$D = \frac{2047}{10}(A)$$

$$D = \frac{2047}{10}(6V)$$

$$D = (204.7)(6)$$

$$D = 1228$$