19

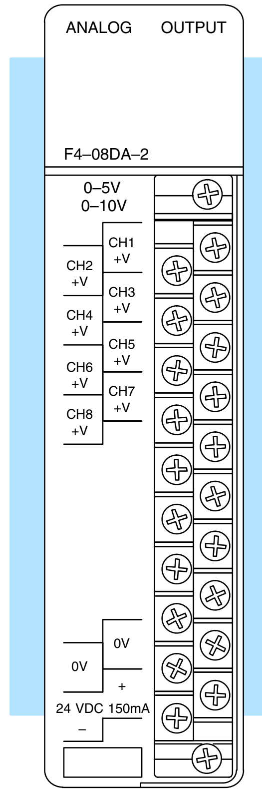
F4–08DA–2 8-Channel Analog Voltage Output

In This Chapter. . . .

Module Specifications

The F4–08DA–2 Analog Voltage Output Module provides several features and benefits.

- It provides eight channels of 0–5V or 0–10V single ended voltage outputs.
- Analog outputs are optically isolated from PLC logic components.
- The module has a removable terminal block, so the module can be easily removed or changed without disconnecting the wiring.
- From one to eight analog outputs may be updated in one CPU scan (D4–440 and D4–450 CPUs only).



Analog Output Configuration Requirements

The F4–08DA–2 Analog Output requires 16 discrete output points in the CPU. The module can be installed in any slot of a DL405 system, including remote bases. The limitations on the number of analog modules are:

- For local and expansion systems, the available power budget and discrete I/O points.
- For remote I/O systems, the available power budget and number of remote I/O points.

Check the user manual for your particular model of CPU for more information regarding power budget and number of local or remote I/O points.

The following tables provide the specifications for the F4–08DA–2 Analog Voltage Output Module. Review these specifications to ensure the module meets your application requirements.

Output Specifications

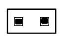
Number of Channels	8, single ended (one common)
Output Range	0–5VDC, 0–10VDC
Resolution	12 bit (1 in 4095)
Output Type	Voltage Sourcing 10mA max.
External Load	1kΩ maximum / 10kΩ minimum (for example: 10 volts at 1kΩ = 10mA load; 10 volts at 10kΩ = 1mA load)
Crosstalk	–70 dB, ±1 count maximum
Linearity Error (end-to-end) and Relative Accuracy	±1 count maximum (10VDC at 25°C)
Full Scale Calibration Error (offset error included)	±6 counts maximum (10VDC at 25°C)
Offset Calibration Error	±3 counts maximum (0VDC at 25°C)
Maximum Inaccuracy	±0.2% at 25°C (77°F) ±0.4% at 0° to 60°C (32° to140°F)
Conversion Time	400 μS maximum, for full scale change 4.5 to 9 mS for digital output to analog out

General Module Specifications

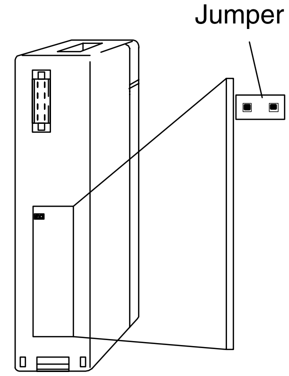
Digital Output Points Required	16 point (Y) outputs, 12 bits binary data, 3 bits channel select, 1 bit output enable
Power Budget Requirement	80 mA at 5 VDC (supplied by base power supply)
External Power Supply	21.6 to 26.4 VDC, 150 mA max., class 2
Accuracy vs. Temperature	±57 ppm / °C full scale calibration range (including maximum offset change, 2 counts)
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

Setting the Module Jumper

Before installing and wiring the module, you may need to change the internal jumper setting. The module has one jumper, located in the open cutout at the rear of the housing. When the jumper is installed (which is the factory default setting), the module operates in 0–5VDC mode for all eight channels. When the jumper is removed, the module operates in 0–10VDC mode. When removed, store the jumper by placing it over one terminal as shown below to prevent losing it.

 Installed = 0–5VDC Mode

 Removed = 0–10VDC Mode



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.
- 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.

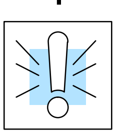
User Power Supply Requirements

The F4–08DA–2 requires a field-side power supply. The module requires 21.6 – 26.4 VDC, Class 2, 150mA max. current.

The D4–430/440/450 CPUs, D4–RS Remote I/O Controller, H4–EBC, and D4–EX Expansion Units have built-in 24 VDC power supplies that provide up to 400mA of current. You may use one of these instead of a separate supply if there is only a couple of analog modules. The current required is 150mA max.: eight outputs driving 1kΩ loads to 10V (10mA x 8); plus 70mA for the module's internal circuitry.

Load Requirements

Each channel in use must have a load impedance of 1kΩ to 10kΩ. Unused channels must be left disconnected.

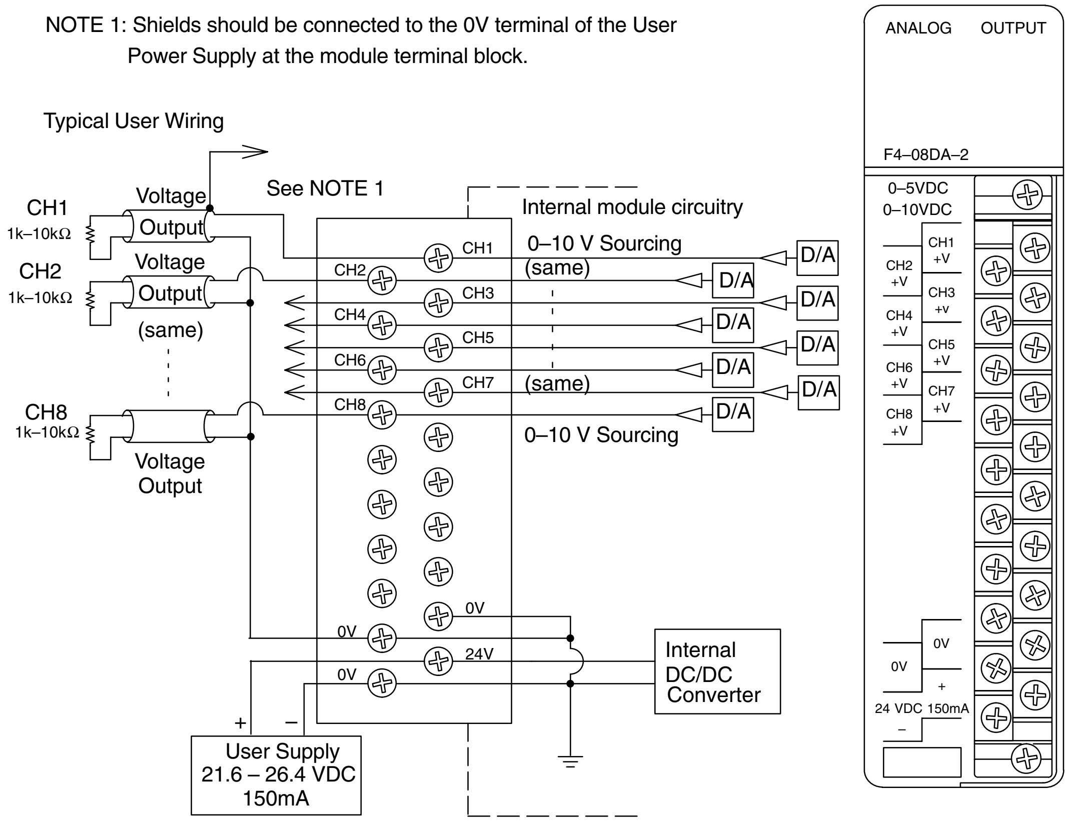


WARNING: If you are using the 24 VDC base power supply, make sure you calculate the power budget. Exceeding the power budget can cause unpredictable system operation that can lead to a risk of personal injury or damage to equipment.

Removable Connector

The F4–08DA–2 module has a removable connector to make wiring easier. Simply loosen the retaining screws and gently pull the connector from the module.

Wiring Diagram

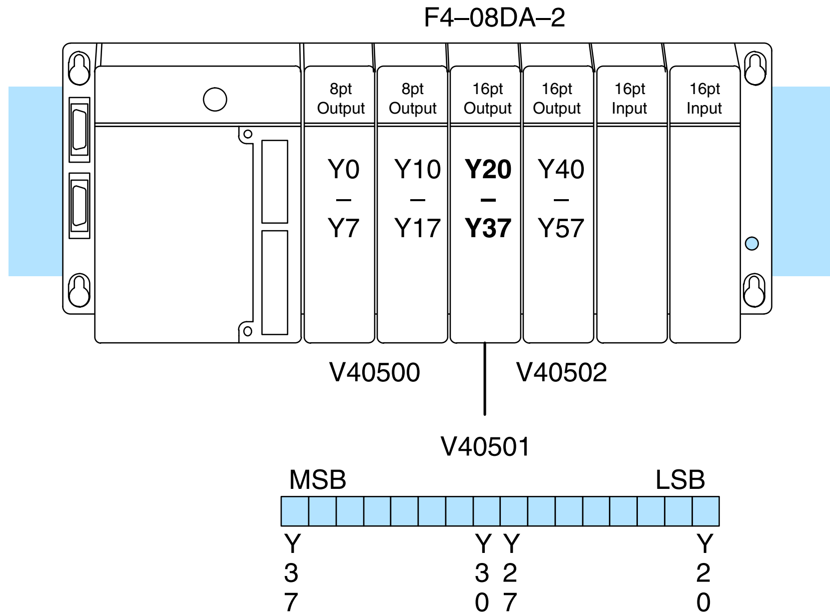


Module Operation

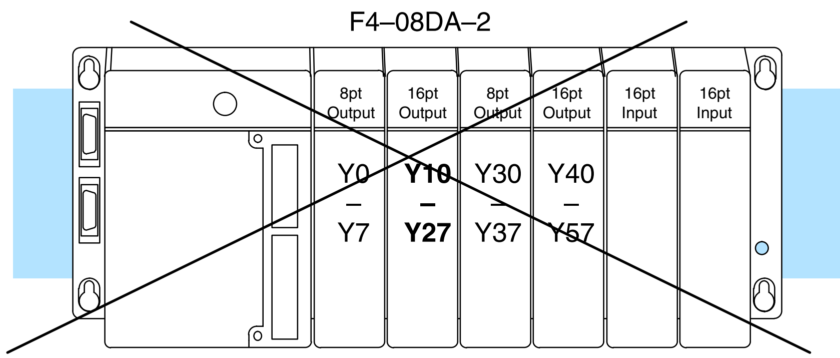
D4–430 Special Requirements

Even though the module can be placed in any slot, it is important to examine the configuration if you are using a D4–430 CPU. As you'll see in the section on writing the program, you use V-memory locations to send the analog data. As shown in the following diagram, if you place the module so the output points do not start on a V-memory boundary, the instructions cannot access the data.

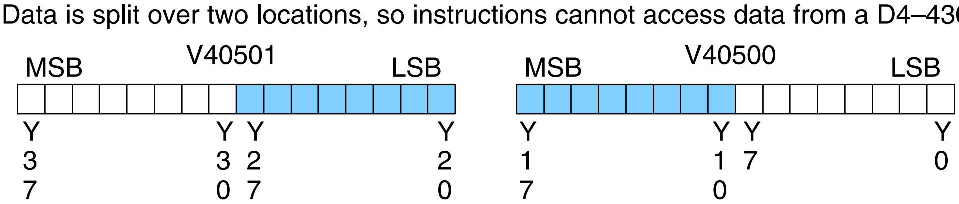
Correct!



Wrong!



Data is split over two locations, so instructions cannot access data from a D4–430.

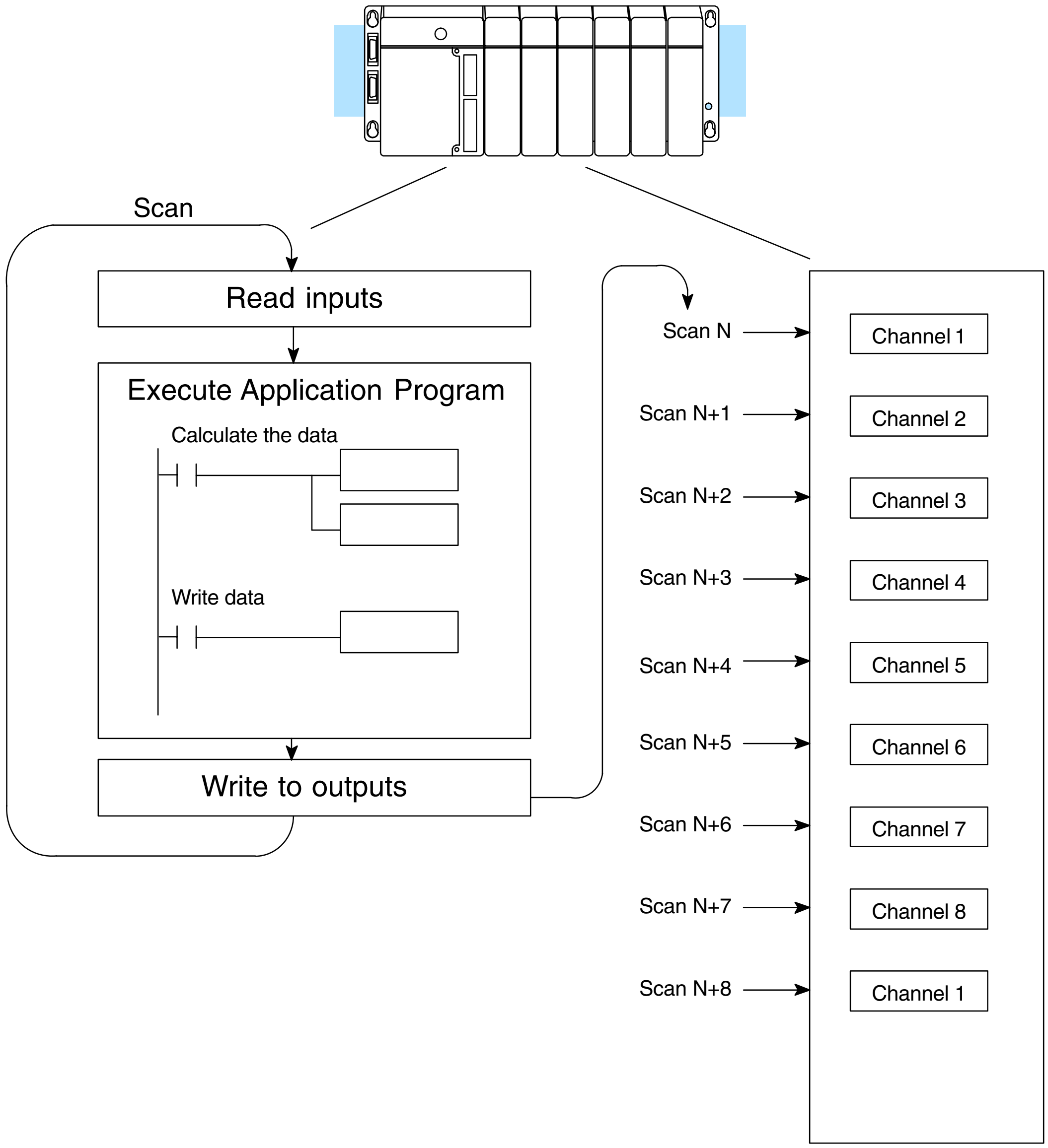


F4–08DA–2
8-Ch. Analog Output

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 F4–08DA–2 module allows you to update the channels in any order. Your control program determines which channels get updated on any given scan. A typical ladder program will update one channel per CPU scan. So, all eight channels can be updated every eight scans. With a D4–440 or D4–450 CPU, you can use Immediate instructions to update all eight channels in the same scan (we'll show you how to do this later).

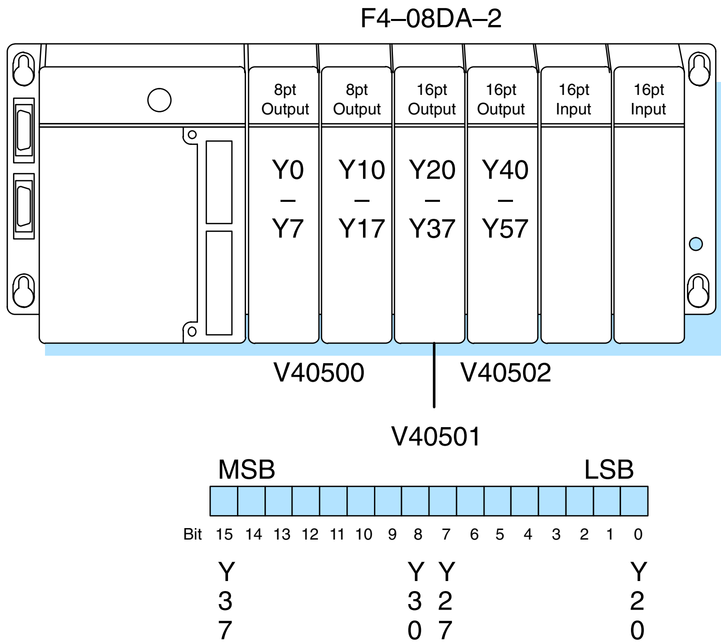


Output Bit Assignments

You may recall the F4–08DA–2 module requires 16 discrete output points from the CPU. These points provide:

- The digital representation of one analog signal per scan.
- Identification of the channel that is to receive the data.
- Output enable control for all channels.

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.

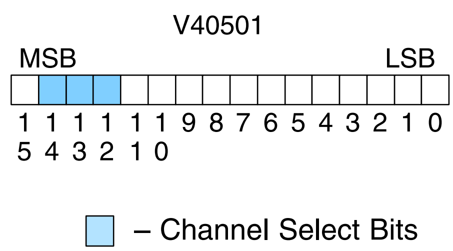


Within this V-memory location the individual bits represent specific information about the channel selected and the analog signal.

Channel Select Bits

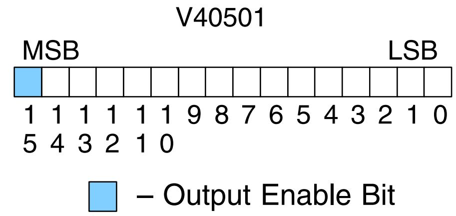
Output bits 12, 13, and 14 of the data word are the channel select outputs. They are binary encoded to select the channel from 1 to 8 that will be updated with the data.

Bit 14	Bit 13	Bit 12	Channel
Off	Off	Off	1
Off	Off	On	2
Off	On	Off	3
Off	On	On	4
On	Off	Off	5
On	Off	On	6
On	On	Off	7
On	On	On	8



Output Enable Bit

Output bit 15 is the Output Enable control bit for all eight channels. When the bit is off, all eight channel output voltage levels drop to 0VDC. Disabling the outputs also clears all eight output data registers. To resume analog output levels, first the Output Enable control bit must turn on. Then, the CPU must write new data to each channel to restore the output voltage for that channel.

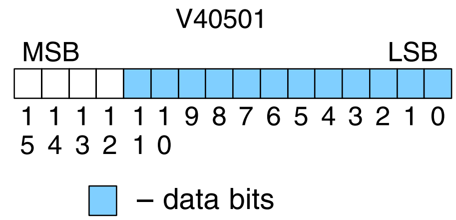


OFF = Disable (and clear)
ON = Enable

Analog Data Bits

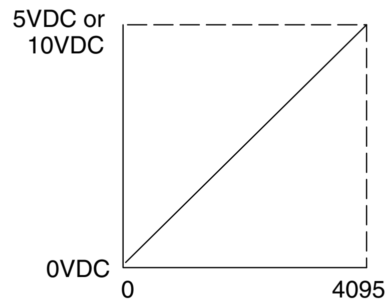
The first twelve bits of the data word represent the analog data in binary format.

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



Module Resolution

Since the module has 12-bit resolution, the analog signal is made of 4096 counts ranging from 0–4095 (2^{12}). For the 0 to 5V scale, sending a 0 produces a 0VDC signal, and 4095 sends a 5VDC signal. This is equivalent to a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF hexadecimal. The graph to the right shows the linear relationship between the data value and output signal level.



Each count can also be expressed in terms of the signal level by using the equation shown. The following table shows the smallest signal change that occurs when the digital value is increased by 1 LSB.

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

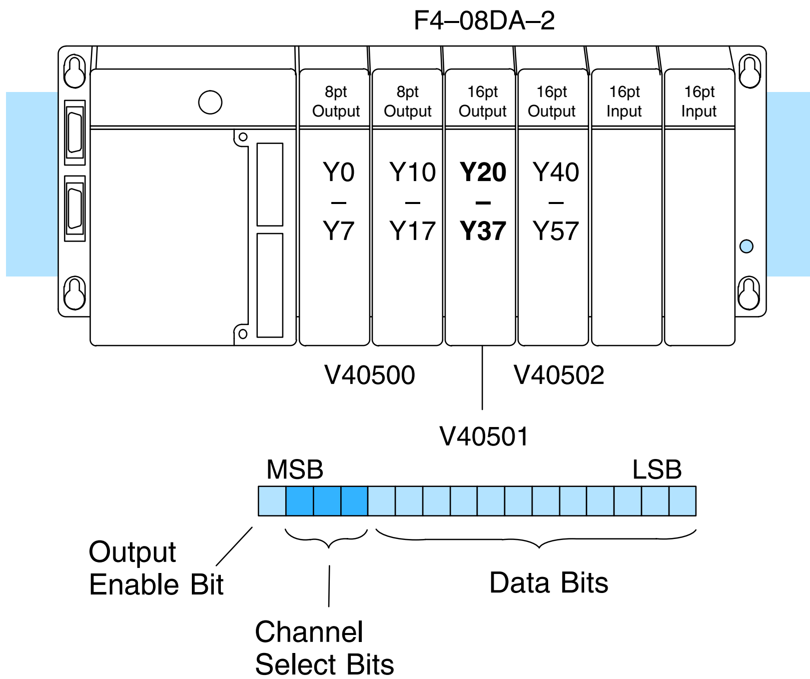
H = high limit of the signal range
L = low limit of the signal range

Signal Range	Span (H – L)	Divide By	Smallest Change
0 to 5VDC	5VDC	4095	1.221mV
0 to 10VDC	10VDC	4095	2.442mV

Writing the Control Program

Update Any Channel

As mentioned earlier, you can update any channel per scan using regular I/O instructions, or any number of channels per scan using Immediate I/O instructions. The following diagram shows the data locations for an example system. You use the channel select outputs to determine which channel gets updated (more on this later).



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.

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.

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

A = analog value (0 – 4095)

U = engineering units

H = high limit of the engineering unit range

L = low limit of the engineering unit range

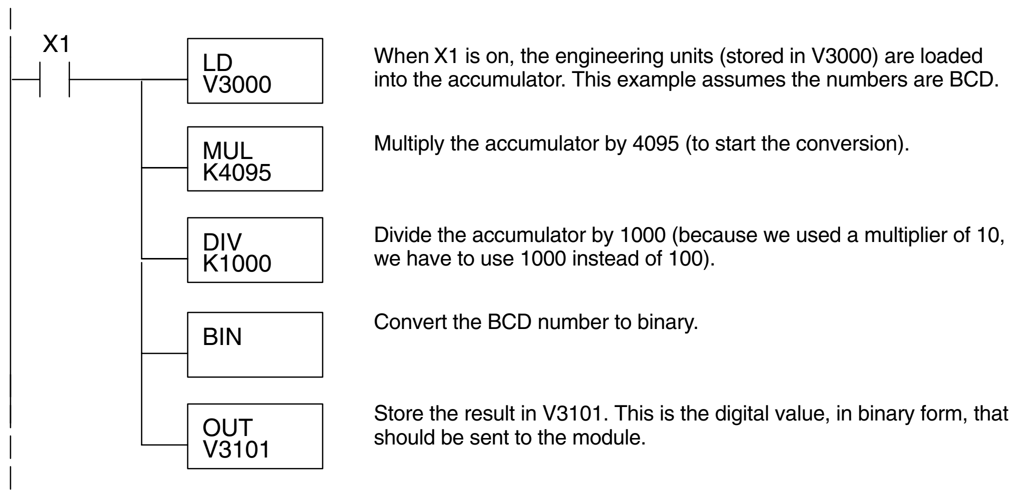
$$A = 10U \frac{4095}{10(H - L)}$$

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

$$A = 2023$$

Here is 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 V3000. Also, you have to perform this for all eight channels if you are using different data for each channel.

NOTE: The DL405 offers various instructions that allow you to perform math operations using binary, BCD, etc. It's usually easier to perform any math calculations in BCD and then convert the value to binary before you send the data to the module. If you are using binary math, you do not have to include the BIN conversion.



V-Memory Registers

The ladder program examples that follow occasionally use certain V-memory register addresses in the CPU that correspond to 16-bit Y output modules. Use the table below to find the V-memory address for the particular slot of your analog module. See Appendix A for additional addresses available in the D4–450 CPU.

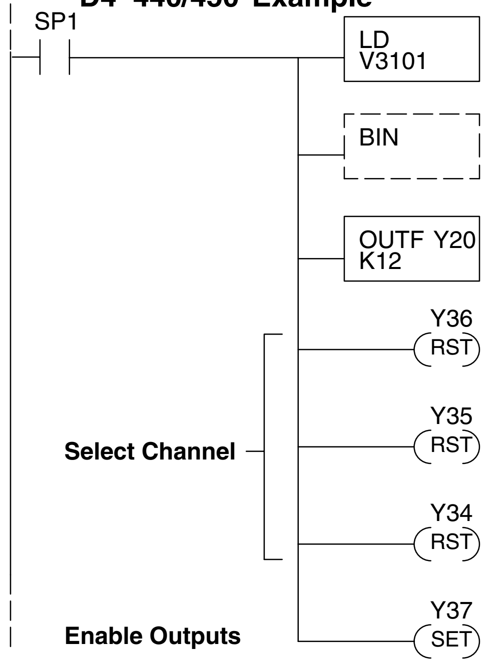
V-Memory Register Addresses for 16-Point Output (Y) Locations										
Y	000	020	040	060	100	120	140	160	200	220
V	40500	40501	40502	40503	40504	40505	40506	40507	40510	40511
Y	240	260	300	320	340	360	400	420	440	460
V	40512	40513	40514	40515	40516	40517	40520	40521	40522	40523

Sending Data to One Channel

The following programs show you how to update a single channel. Notice the D4–430 CPU requires a slightly different program than the D4–440/D4–450 CPUs. Since the D4–430 does not support the OUTF instruction, the program must be modified to make sure the channel select bits are not accidentally changed by the data in the accumulator. These examples assume you already have the data loaded in V3101.

430 ✗ 440 ✓ 450 ✓

D4–440/450 Example



The LD instruction loads the data for channel 1 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've already converted the data elsewhere).

The OUTF sends the 12 bits to the data word. Our example starts with Y20, but the actual value depends on the location of the module in your application.

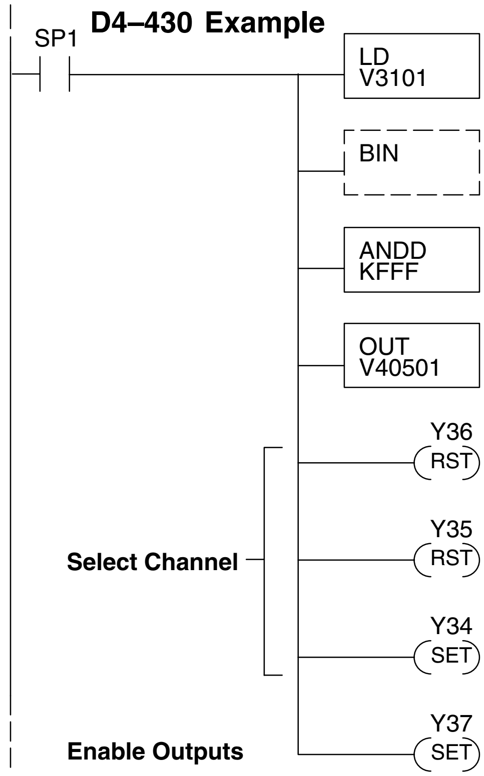
Turn Y36, Y35, and Y34 off to update Channel 1.

Y36	Y35	Y34	Channel
Off	Off	Off	Ch. 1
Off	Off	On	Ch. 2
Off	On	Off	Ch. 3
Off	On	On	Ch. 4
On	Off	Off	Ch. 5
On	Off	On	Ch. 6
On	On	Off	Ch. 7
On	On	On	Ch. 8

Turn on Y37 to enable all eight output channels.

430 ✓ 440 ✓ 450 ✓

D4–430 Example



The LD instruction loads the data for channel 1 into the accumulator. Since SP1 is used, this rung automatically executes 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've 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.

Turn Y36, Y35, and Y34 off to update Channel 1.

Y36	Y35	Y34	Channel
Off	Off	Off	Ch. 1
Off	Off	On	Ch. 2
Off	On	Off	Ch. 3
Off	On	On	Ch. 4
On	Off	Off	Ch. 5
On	Off	On	Ch. 6
On	On	Off	Ch. 7
On	On	On	Ch. 8

Turn on Y37 to enable all eight output channels.

Sequencing the Channel Updates

The next four example programs show you how to send digital values to the module when you have more than one channel. These examples will automatically update all eight channels over eight scans.

The first two sequencing examples, examples 1 and 2, are fairly simple and will work in almost all situations. We recommend these for new users. They use control relays C1 through C10 as index numbers corresponding to the channel updated on any particular scan. At the end of each scan, only one control relay C1 through C10 is on. On each subsequent scan, the next control relay energizes. The channel sequencing automatically begins with channel 1 on the first scan, or after any disruption in the logic. You must use example 2 with D4–430 CPUs. Either example will work with D4–440 or D4–450 CPUs.

The next two examples, 3 and 4, are slightly more complex. However, they do not depend on the use of control relays to provide channel sequencing. Instead, they use function boxes to increment a channel pointer value in V-memory. Then, other instructions perform bit manipulations to position the channel select bits properly in the output word to the module. You must use example 4 with D4–430 CPUs. Either example will work with D4–440 or D4–450 CPUs.

In the fifth example, we show you how you can update all eight channels in the same scan with D4–440 and D4–450 CPUs. However, this can increase the scan time and you may not always need to update all eight channels on every scan.

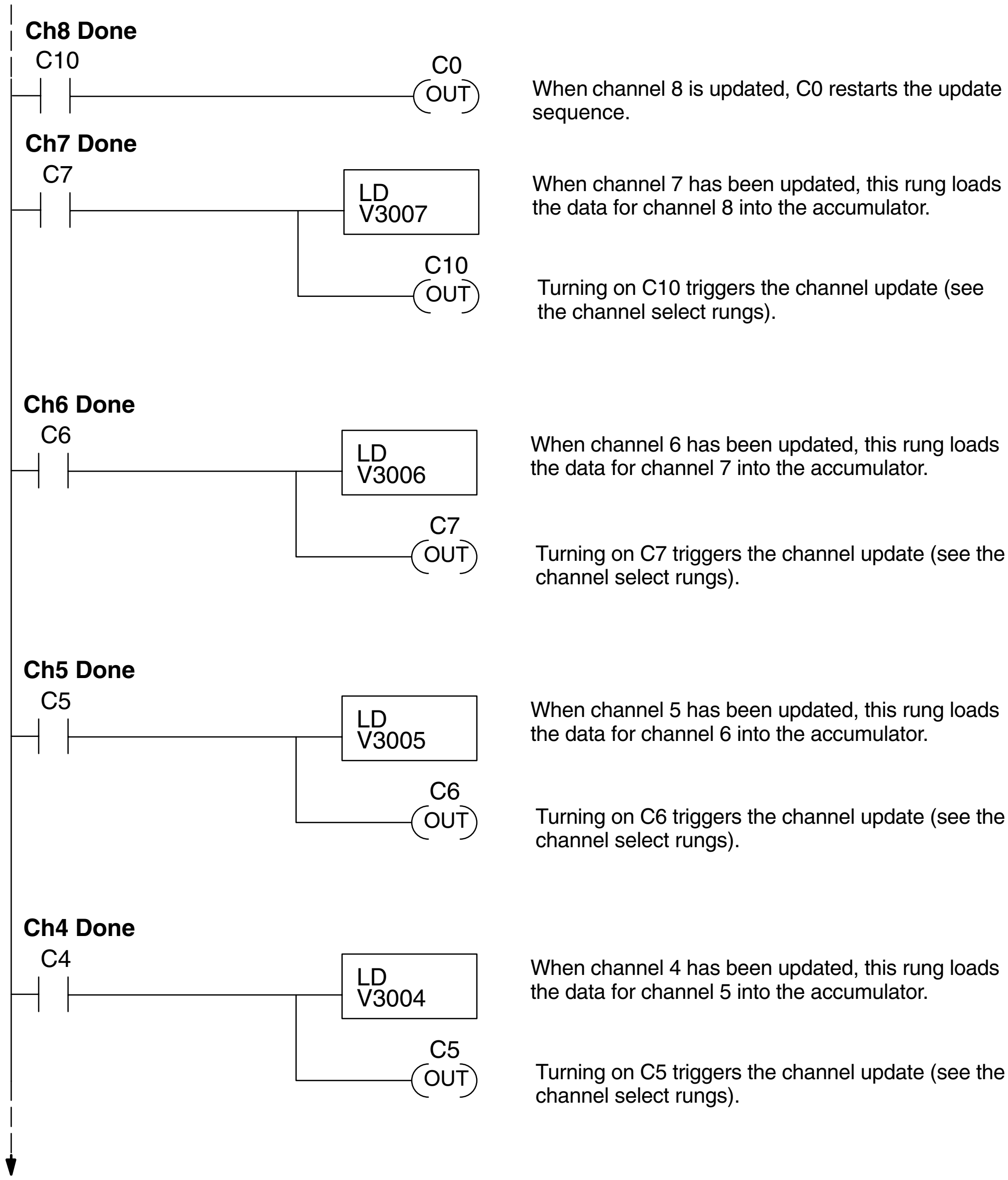
In the last example, we show you how you can update a single channel during the scan with D4–440 and D4–450 CPUs using the Immediate instructions.

Sequencing Example 1, D4–440/450

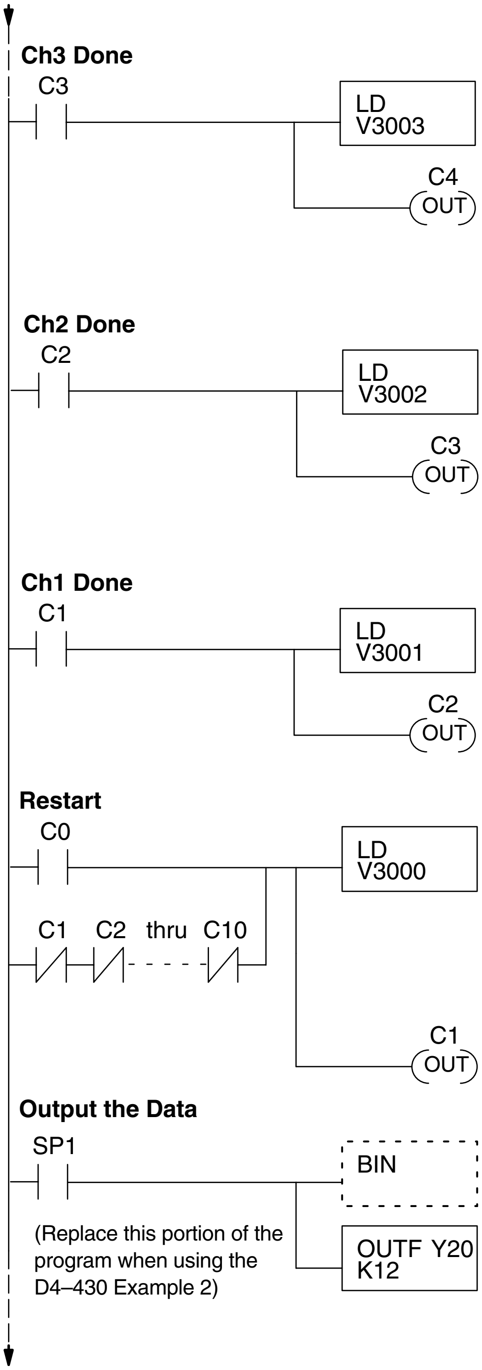
430 440 450

The following program example shows how to send digital values to the module when you have more than one channel. This example assumes you have already loaded the data according to the following table. It is important to use the rungs in the order shown for the program to work. This example will not work with D4–430 CPUs.

V-Memory Locations for Output Data in Examples 1 and 2								
Channel Number	1	2	3	4	5	6	7	8
V–Memory Storage	3000	3001	3002	3003	3004	3005	3006	3007



Example 1 Continued



When channel 3 has been updated, this rung loads the data for channel 4 into the accumulator.

Turning on C4 triggers the channel update (see the channel select rungs).

When channel 2 has been updated, this rung loads the data for channel 3 into the accumulator.

Turning on C3 triggers the channel update (see the channel select rungs).

When channel 1 has been updated, this rung loads the data for channel 2 into the accumulator.

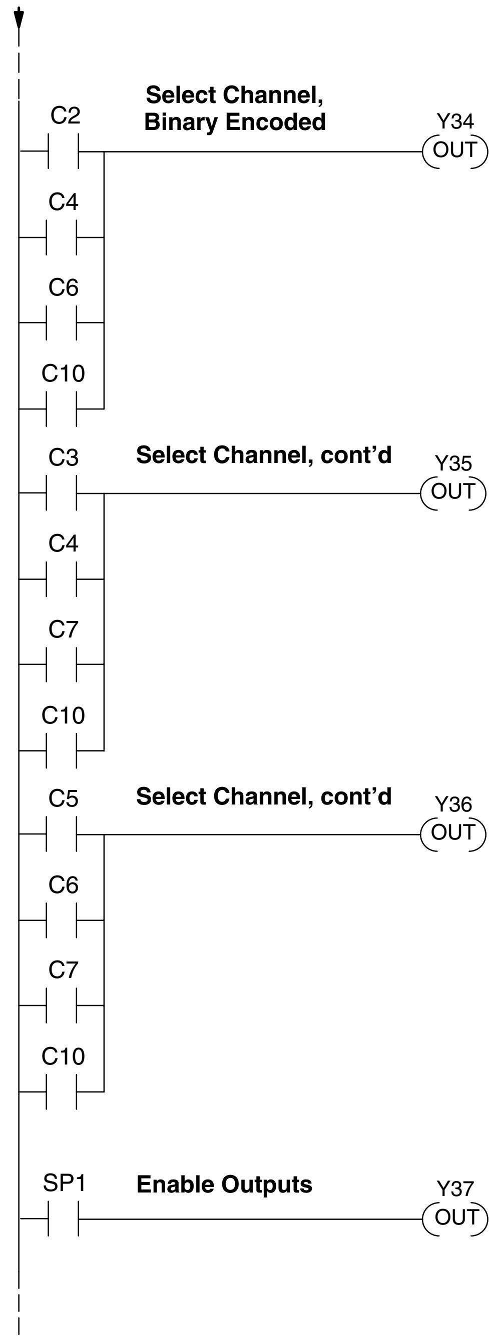
Turning on C2 triggers the channel update (see the channel select rungs).

This rung loads the data for channel 1 into the accumulator. C0 restarts the sequence after channel 8 is done (see the top rung). The first scan or any interruption in control relay sequencing is detected when control relays C1 through C10 are off (all eight contacts not shown here due to space constraints). In this case, we also start the sequence with channel 1.

Turning on C1 triggers the channel update (see the channel select rungs).

This rung converts the accumulator data for all channels (one per scan) to binary (you must omit this step if you've already converted the data elsewhere). It also loads the data to the appropriate bits of the data word. Our example starts with Y20, but the actual value depends on the location of the module in your application.

Example 1 Continued



Set Y36, Y35, and Y34 to the binary code which selects the output channel 1 through 8, based on the control relay status.

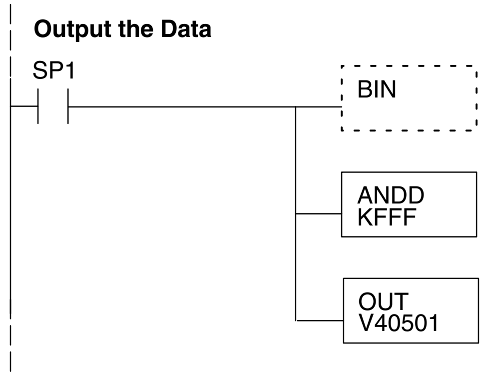
CR(on)	Y36	Y35	Y34	Channels
C1	Off	Off	Off	Ch. 1
C2	Off	Off	On	Ch. 2
C3	Off	On	Off	Ch. 3
C4	Off	On	On	Ch. 4
C5	On	Off	Off	Ch. 5
C6	On	Off	On	Ch. 6
C7	On	On	Off	Ch. 7
C10	On	On	On	Ch. 8

Enable all channels. SP1 is always on.

Sequencing Example 2, D4–430

✓ ✓ ✓
430 440 450

Since the D4–430 does not support the OUTF instruction, the previous program must be modified to make sure the channel select bits or the output enable bits are not accidentally changed by the data in the accumulator. Replace the "Output the Data" rung in the middle of Example 1 with the new rung below. Be sure to retain the original order of the rungs shown in Example 1 for the program to work. This example will also work with D4–440 and D4–450 CPUs.



This rung converts the accumulator data for channels 1 through 8 (one per scan) to binary (You must omit this step if you've 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 for the first bank of 8 channels, but the actual value depends on the location of the module in your application.

Sequencing Example 3, D4–440/450

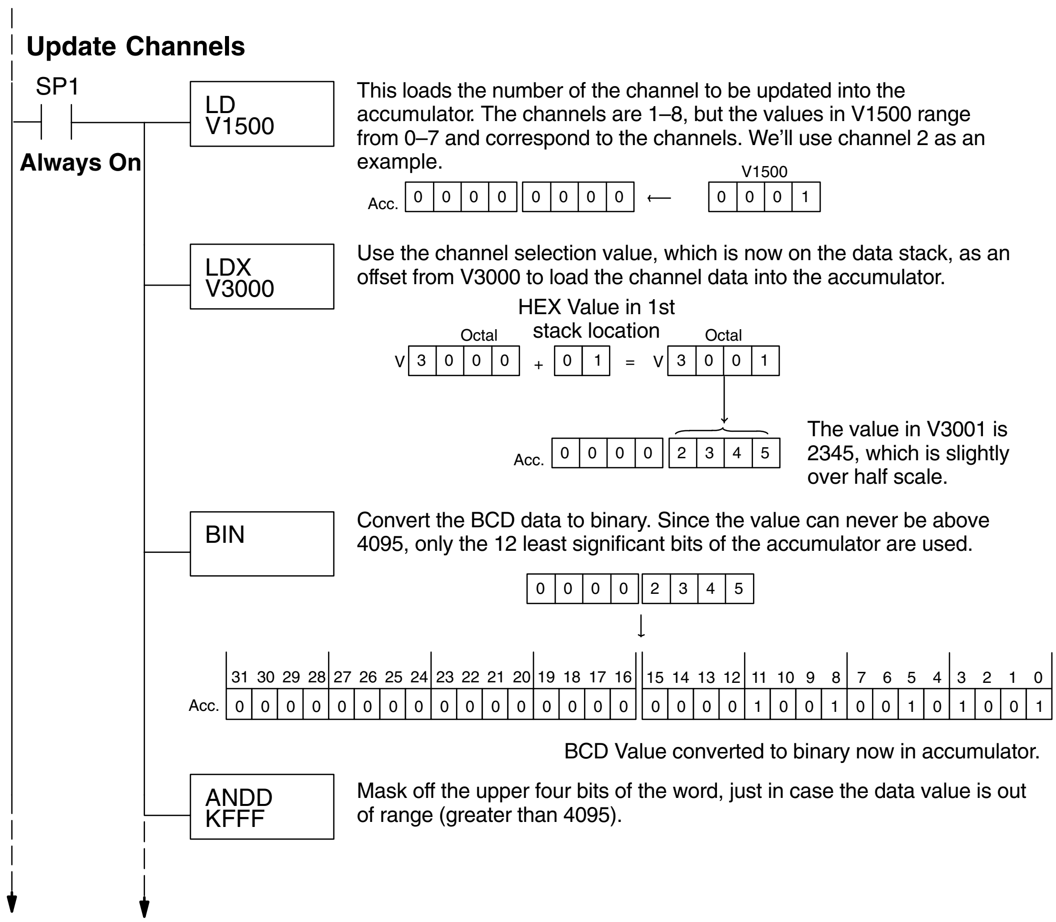
✕ 430 ✓ 440 ✓ 450

The following program example shows how to send digital values to the module when you have more than one channel. This example works only for D4–440 and D4–450 CPUs. It assumes you are using the following data locations.

V-Memory Locations for Output Data in Example 3								
Channel Number	1	2	3	4	5	6	7	8
V-Memory Storage	3000	3001	3002	3003	3004	3005	3006	3007

The channel index is stored in V1500. It varies from 0 to 7, pointing to channels as shown: 0 – Ch. 1, 1 – Ch. 2, and 7 – Ch. 8. This example assumes V1500 is initialized to "0" earlier in the program.

This example program updates one channel during each scan. The program comments for this portion also shows the accumulator status at each step. The last portion of the program increments the channel index number and resets it after eight scans.



Example 3 Continued

LD V1500

Load the number of the channel to be updated back into the accumulator again (the channel data is moved to the first data stack location).

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SHFL K12

This instruction moves the channel select bit(s) into the proper location. We'll use it later when we send the 16-bit data word to the module.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

ORD K8000

Set the Output Enable bit, by combining the value of 8000 hex with the accumulator value. This sets bit 15 to "1", enabling all channels.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

ADDBS

Earlier in the program the data value was placed into the first data stack location. The ADDBS instruction adds the value currently in the accumulator with the value in the first data stack location.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

+

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Stack	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

↓

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	1	0	0	1

Data for Analog Module

OUTF Y20 K16

Send the lower 16 bits stored in the accumulator to the analog module. The lowest 12 bits contain the analog data. Bits 12, 13, and 14 are the channel selection bits. Bit 15 is the Output Enable bit.

Increment Channel Index

INCB V1500

Increment the channel index value. This allows the logic to cycle through all eight channels.

Reset Channel Index

V1500 K8 ─┤ = ├─

LD K0

When channel 8 has been updated, then reset the channel selection memory location to 0 (remember, 0 represents channel 1).

OUT V1500

Sequencing Example 4, D4–430

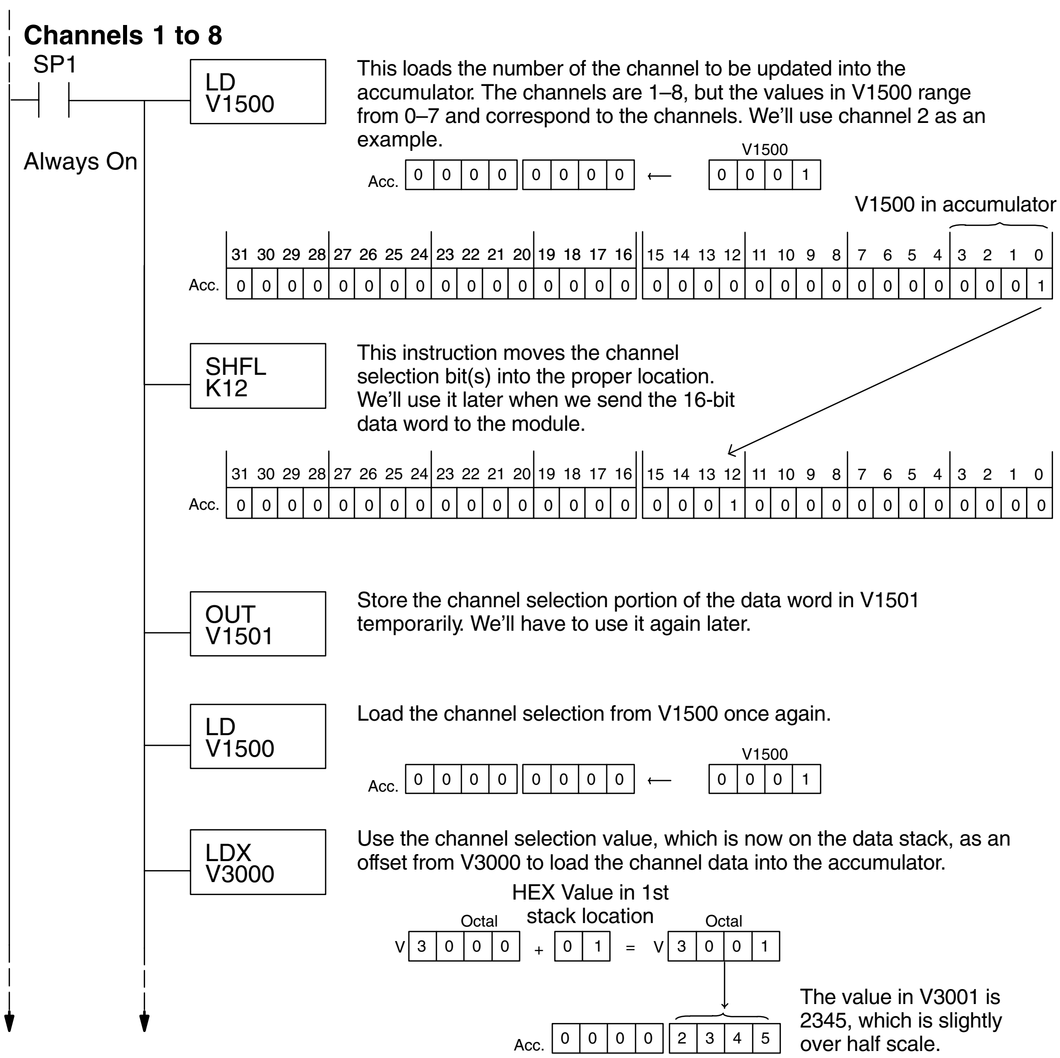
430 440 450

The following program example shows how to send digital values to the module when you have more than one channel. This example works for D4–430, D4–440 or D4–450 CPUs. It assumes you are using the following data locations.

V-Memory Locations for Output Data in Example 4								
Channel Number	1	2	3	4	5	6	7	8
V–Memory Storage	3000	3001	3002	3003	3004	3005	3006	3007

The channel index is stored in V1500. It varies from 0 to 7, pointing to channels as shown: 0 – Ch. 1, 1 – Ch. 2, and 7 – Ch. 8. This example assumes V1500 is initialized to “0” earlier in the program.

The first portion of the program updates one channel during each scan. The program comments show the accumulator status at each step. The last portion of the program increments the channel index number and resets it after eight scans.



Example 4 Continued

BIN

Convert the BCD data to binary. Since the value can never be above 4095, only the least significant 12 bits of the accumulator are used.

0	0	0	0	2	3	4	5

↓

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

BCD Value converted to binary now in accumulator.

ANDD KFFF

Mask off the upper four bits of the word, just in case the data value is out of range (greater than 4095).

OR V1501

Earlier in the program the channel selection portion of the data word was created and stored in V1501. Now we can OR this location with the data word currently in the accumulator to get the final data word that is ready to send to the analog module.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1

+

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V1501	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

↓

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0	1

Data for Analog Module

ORD K8000

Set the Output Enable bit, by combining the value of 8000 hex with the accumulator value. This sets bit 15 to “1”, enabling all channels.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Acc.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	1	0	1	0	0	1

OUT V40501

Send the data stored in the lower half of the accumulator to the analog module (the OUT instruction ignores the upper 16 bits of the accumulator). The most significant four bits of the analog word contain the channel selection bits. The remaining 12 bits contain the analog data.

Increment Channel Index

INCB V1500

Increment the channel index value. This allows the logic to cycle through all eight channels.

Reset Channel Index

V1500K8 ─┤ = ├─

LD K0

When channel 8 has been updated, then reset the channel index memory location to 0 (remember, 0 represents channel 1).

OUT V1500

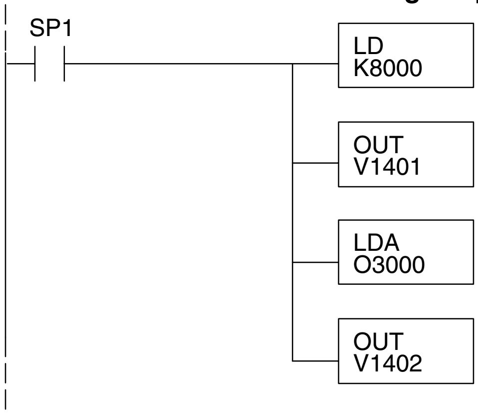
Updating all Channels in a Single Scan, D4–440/450

430 440 450

By using the Immediate instructions found in the D4–440 and D4–450 CPUs, you can easily update all eight channels in a single scan. Before choosing this method, remember it slows CPU scan time (approximately 12 mS). To minimize this impact, change the SP1 (Always On) contact to an X, C, etc. permissive contact that only updates the channels as required. This example assumes you already have the data loaded in V3000 to V3007 for channels 1 to 7 respectively.

NOTE: This program will not work in a remote/slave arrangement. Use one of the programs shown that reads one channel per scan.

Initialize the Immediate Analog Output Pointers



The LD instruction loads the data into the accumulator. The constant K8000 represents channel 1 selected, Output Enable bit is on.

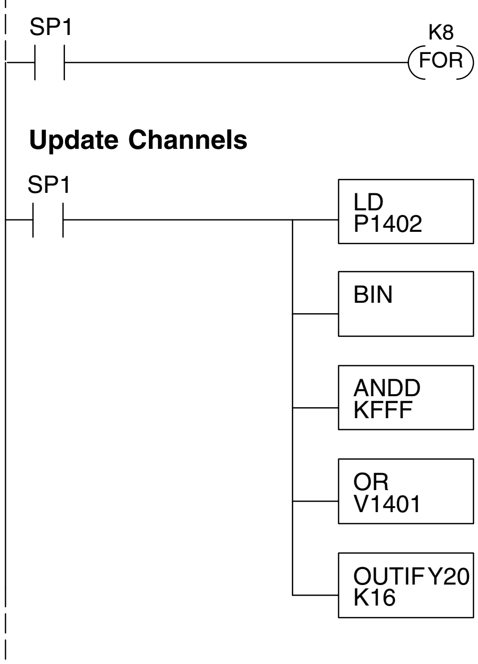
We store the channel index pointer in V1401. The program increments this in each pass, resetting it after eight scans.

The Load Address instruction takes the octal 3000 and converts it to hex, then puts it in the accumulator. V3000 is the location which contains the data for channel 1.

V1402 contains the pointer for channel 1 to 8 data.

The following FOR–NEXT loop updates all eight channels in a single scan.

FOR–NEXT Loop



The following FOR–NEXT loop updates channels 1 through 8. The valid range for the constant K is 1 to 8 for the module. For example, a value of 1 will update channel 1.

Load the analog output value into the accumulator. V3000 = Ch. 1, V3001 = Ch. 2, ... V3007 = Ch. 8. The number at V1402 is a pointer to the address of the value.

Convert the data to binary. This step is optional, and your program may do this conversion elsewhere.

Mask off the channel select and output enable bits, so they are not corrupted by data out of range.

Combine the channel select and output enable bits. for Y36, Y35, and Y34 with the data in the accumulator.

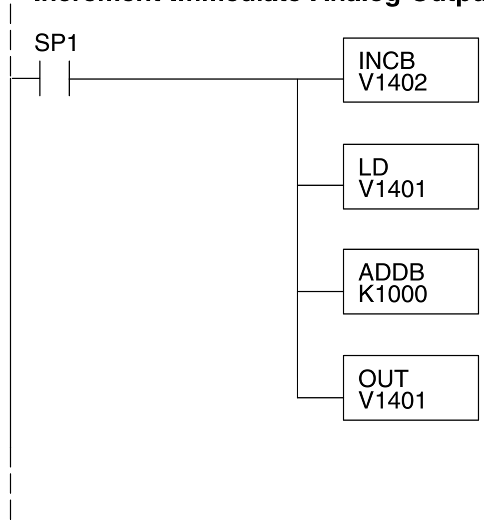
Immediately write the lower 16 bits of the accumulator to the module's lower data word. This updates channels 1 through 8 during the FOR–NEXT loop.

Updating all Channels in a Single Scan, Continued

430 440 450

Now we increment the Immediate Analog Output Pointers for the channel, before the next pass through the FOR–NEXT loop.

Increment Immediate Analog Output Pointers



Increment the analog output data pointer.

Load the curent channel index count into the accumulator.

Add 1000 hex to the channel index value, since the channel select field is in the most significant four bits.

Save the incremented channel index value for the next pass through the FOR–NEXT loop.

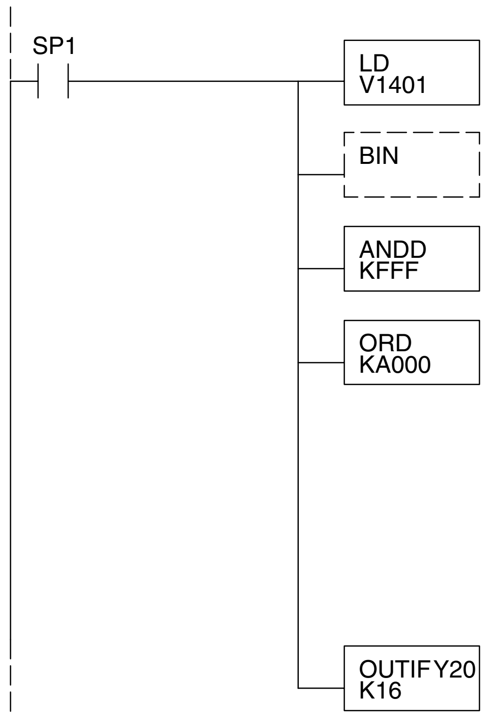
Here is the end of the FOR–NEXT loop.

Execute the loop above the number of times specified in the FOR instruction.

Updating a Single Channel During a Scan, D4–440/450

430 440 450

You can also update just a single channel during a ladder logic scan by using the Immediate instructions found in the D4–440 and D4–450 CPUs. By removing the FOR–NEXT rungs and a couple of other rungs, we create the example below. This example assumes the data is already loaded in V1401.



Load the analog output value for the channel from V1401 into the accumulator.

Convert the data to binary. This step is optional, and your program may do this conversion elsewhere.

Mask off the channel select and output enable bits, so they are not corrupted by data out of range.

Combine the channel select and output enable bits (channel index) with the desired channel with the data in the accumulator. We chose channel 3 here.

Index	Channel
8000	Ch. 1
9000	Ch. 2
A000	Ch. 3
B000	Ch. 4
C000	Ch. 5
D000	Ch. 6
E000	Ch. 7
F000	Ch. 8

Immediately write the lower 16 bits of the accumulator to the module's lower data word. This updates the selected channel.

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 useful 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 5VDC	$A = \frac{5D}{4095}$	$D = \frac{4095}{5}A$
0 to 10VDC	$A = \frac{10D}{4095}$	$D = \frac{4095}{10}A$

For example, if you need a 3V signal level with the module set for 0–5V, you would use the following formula to determine the digital value that should be stored in the V-memory location that contains the data.

$$D = \frac{4095}{5}A$$

$$D = \frac{4095}{5}(3V)$$

$$D = (819)\ (3)$$

$$D = 2457$$