13

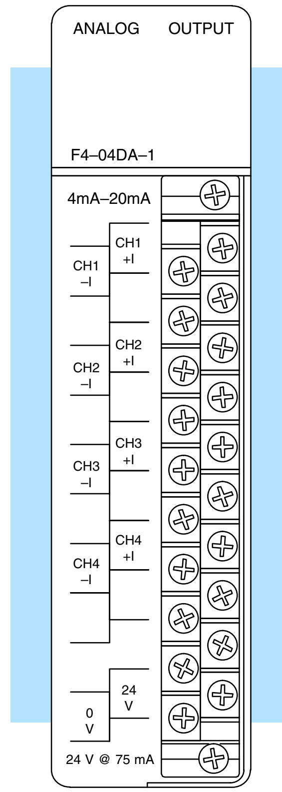
F4-04DA-1 4-Channel Analog Current Output

In This Chapter. . . .

Module Specifications

The F4–04DA–1 Analog Current Output Module provides several features and benefits.

- It is a direct replacement for the popular F4–04DA module in applications set for 4–20 mA output range.
- It provides four channels of 4–20 mA single ended current 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.
- All four analog outputs may be set in one CPU scan (DL440 and DL450 CPUs only).



The following tables provide the specifications for the F4–04DA–1 Analog Output Module. Review these specifications to ensure the module meets your application requirements.

Output Specifications

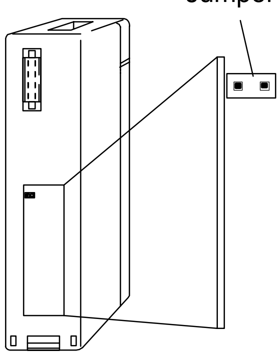
Number of Channels	4, single ended (one common)
Output Range	4–20 mA
Resolution	12 bit (1 in 4095)
Output Type	Outputs sink 4-20 mA from external supply
External Load Resistance	0Ω minimum
Maximum Loop Supply	30 VDC
Peak Output Voltage	40 VDC (clamped, transient suppressed)
Maximum Load / Power Supply	620Ω/18V, 910Ω/24V, 1200Ω/30V
Linearity Error (best fit)	±1 count (±0.025%) maximum
Gain Calibration Error	±5 counts maximum
Offset Calibration Error	±3 counts maximum
Maximum Inaccuracy	±0.1% @ 25°C (77°F) ±0.3% @ 0 to 60°C (32 to140°F)
Conversion Time	100 μs maximum, settling time 2.0 ms maximum, digital out to analog out

General Module Specifications

Digital Output Points Required	16 point (Y) outputs, 12 bits binary data and 4 active channel bits
Power Budget Requirement	70 mA @ 5 VDC (from base)
External Power Supply	21.6–26.4 VDC, 75 mA, class 2 (add 20 mA for each current loop used)
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'll need to decide the proper jumper setting for your application. 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 Standard Mode. In this mode, the channel select bits are binary encoded, and you have access to the Output Enable control bit. We recommend this setting for new applications, since it offers more overall features and easier CPU programming.



Installed = Standard Mode

Removed = F4–04DA Compatibility Mode

When the jumper is removed, the module operates in the F4–04DA Compatibility Mode. If you have an existing F4–04DA application that uses 4–20 mA outputs only, choosing this mode will allow existing ladder logic to work with this module. In the F4–04DA Compatibility Mode, each channel has an individual channel select output bit (the Output Enable control bit is not accessible).

For either mode, the module requires 16 (Y) output points. Choose the mode of operation that best fits your application.

NOTE: If you have selected the F4–04DA Compatibility Mode (jumper removed), refer to the chapter on the F4–04DA for output bit assignments and ladder logic examples. **The remainder of the information in this chapter applies only to the F4–04DA–1 operating in the Standard Mode (with jumper installed).**

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 F4–04DA–1 requires at least one field-side supply. You may use the same or separate power sources for the module supply and loop supply. The module requires 21.6 to 26.4 VDC, Class 2, at 75 mA current. The four current loops require 18 to 30 VDC, at 20 mA each.

The DL430/440/450 CPUs, D4–RS Remote I/O Controller, 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 you have only a couple of analog modules. The current required is 75 mA (module), plus 80 mA (four current loops) for a total of 155 mA.

In some situations it's desirable to power the loops separately due to power budget or due to their remote location from the PLC. This will work, as long as the loop supply meets the voltage and current requirements, and its minus (–) side and the module supply's (–) side are connected together.

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.

Load Requirements

Each channel in use must have a load impedance less than 620 ohms at18V, 910 ohms at 24V, or 1200 ohms at 30V. Unused channels must be left disconnected.

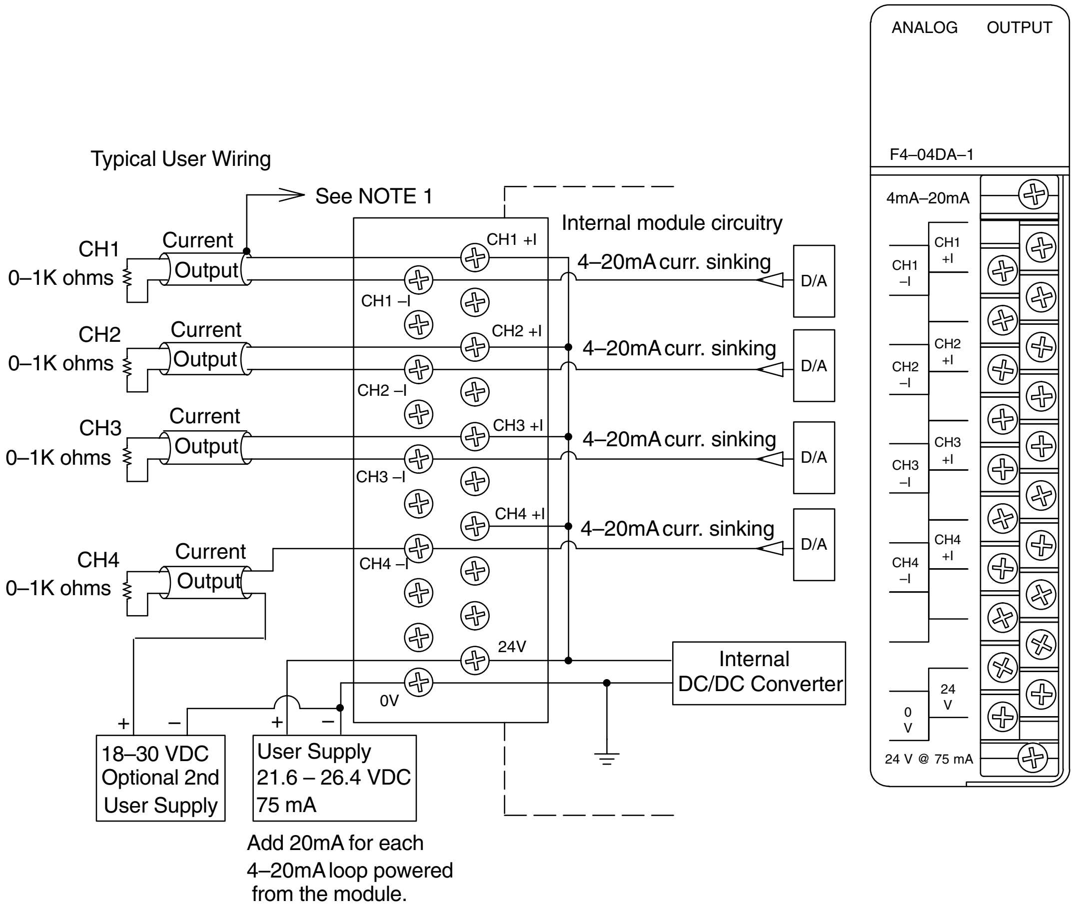
Removable Connector

The F4–04DA–1 module has a removable connector to make wiring easier. Simply loosen the retaining screws and gently pull the connector from the module. Use the following diagram to connect the field wiring. The diagram shows separate module and loop supplies for channel 4. If you only want to use one field-side supply, just combine the supplies' positive (+) terminals into one node, and remove the loop supply.

Wiring Diagram

NOTE 1: Shields should be connected to the 0V terminal of the module terminal block.

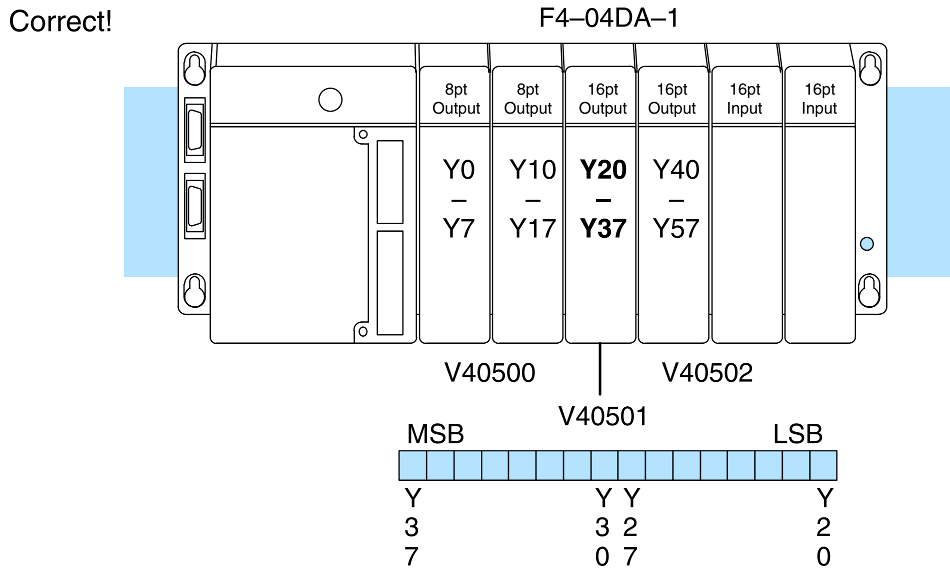
NOTE 2: Unused current outputs should remain open (no connections).

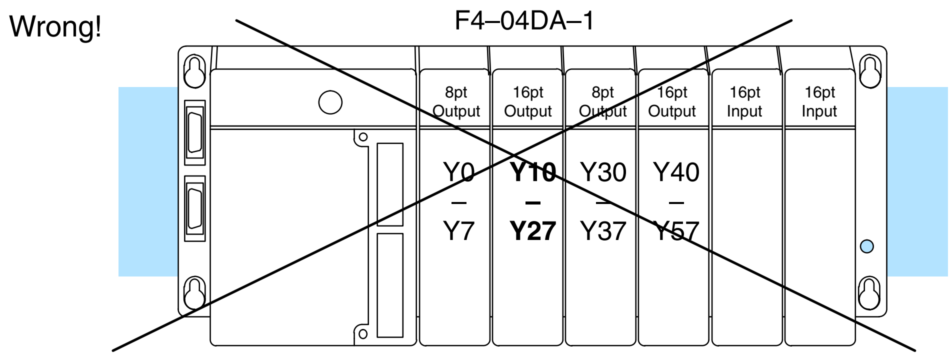


Module Operation

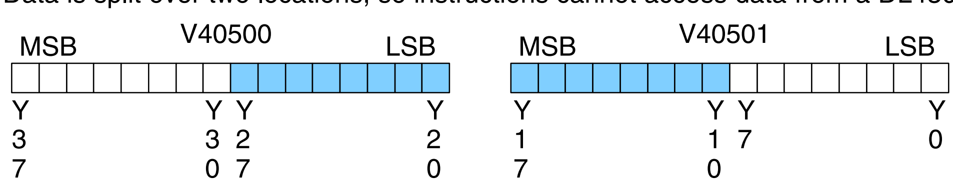
DL430 Special Requirements

Even though the module can be placed in any slot, it is important to examine the configuration if you are using a DL430 CPU. As you will 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.





Data is split over two locations, so instructions cannot access data from a DL430.



Analog Output Configuration Requirements

The D4–04DA–1 Analog Output module 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:

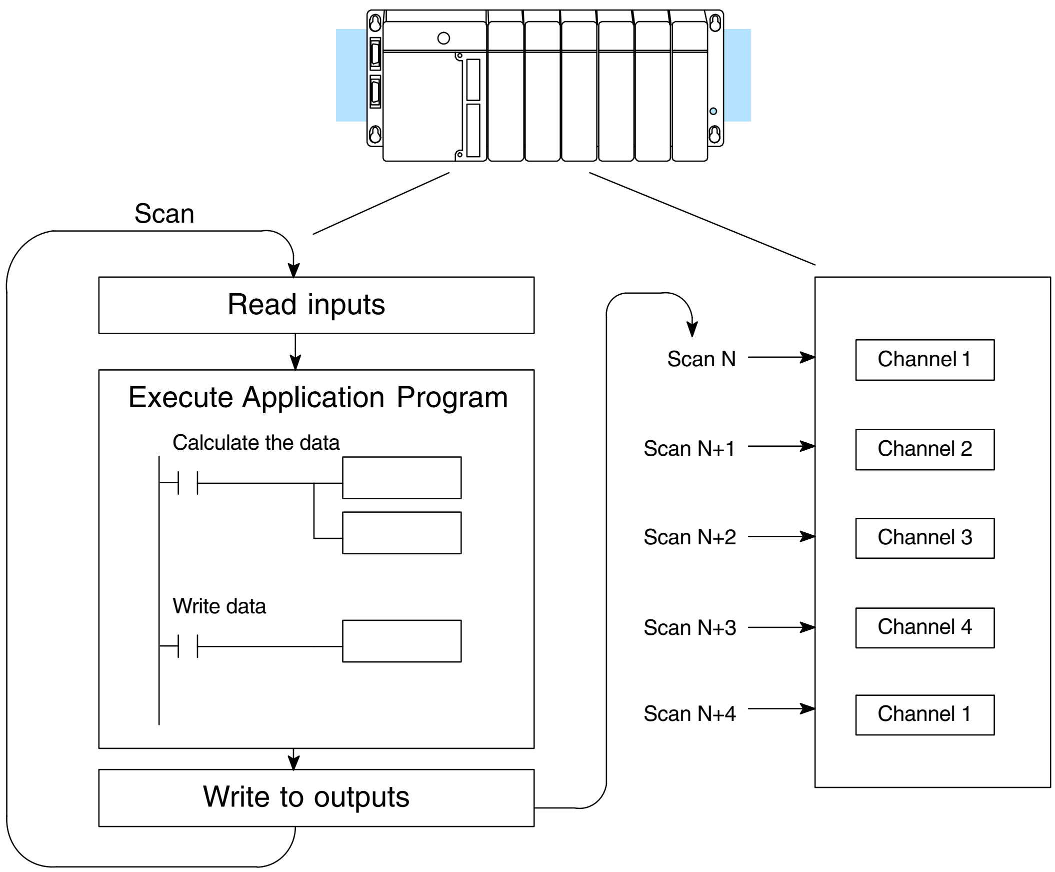
- For local and expansion systems, the available power budget and discrete I/O points.
- For remote I/O systems, the available power budget and number of remote I/O points.

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.

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

The F4–04DA–1 module allows you to update the channels in any order. Your control program determines which channel gets updated on any given scan. The exact method depends on the operating mode you selected when setting the jumper. With a DL440 or DL450 CPU, you can use immediate instructions to update all four channels in the same scan (we'll show you how to do this later).

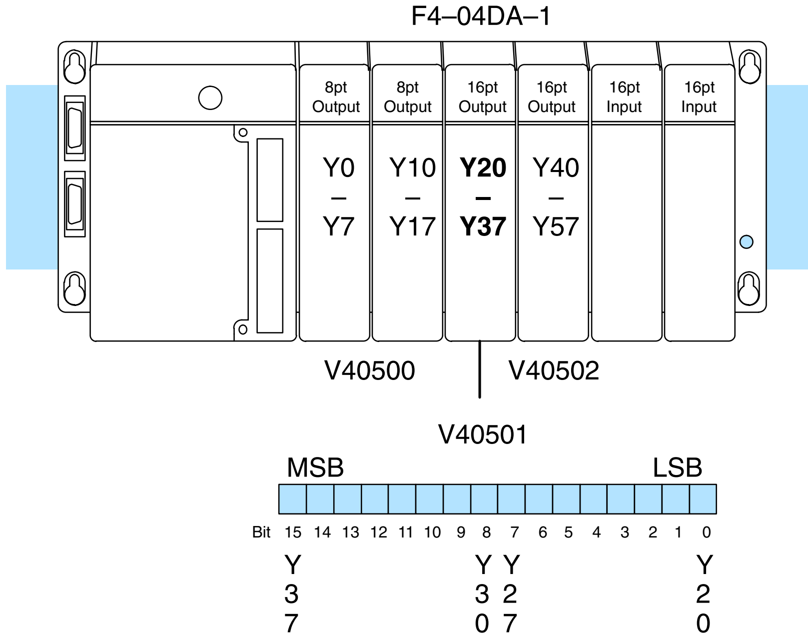


Output Bit Assignments

You may recall the F4–04DA–1 requires 16 discrete output points from the CPU. These points provide:

- The digital representation of the analog signal.
- Identification of the channel that is to receive the data.

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.

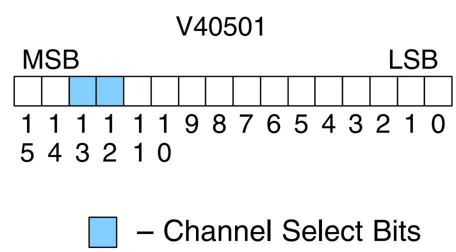


Within this V-memory location the individual bits represent specific information about the channel selected and the analog signal.

Channel Select Bits

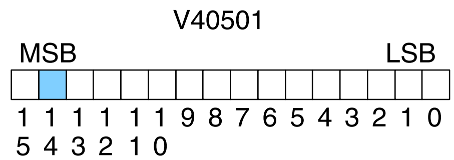
Output bits 12 and 13 are the channel select outputs. They are binary encoded to select the channel that will be updated with the data. The bits are assigned as follows.

Bit 13	Bit 12	Channel
Off	Off	1
Off	On	2
On	Off	3
On	On	4



Output Enable Bit

Output bit 14 is the Output Enable control bit for all four channels. When it is off, all channel output currents decrease to their lowest level, which is 4 mA for connected loads. Disabling the outputs also clears the module's output data registers for each channel. 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 current for that channel.



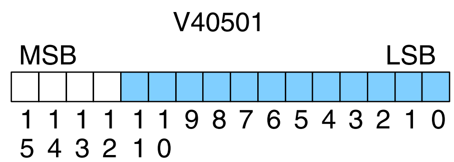
– Output Enable Bit

OFF = Disable (and clear)
ON = Enable

Analog Data Bits

The first twelve bits of the V-memory location represent the analog data in binary format. Each bit has a binary weight according to the following table.

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

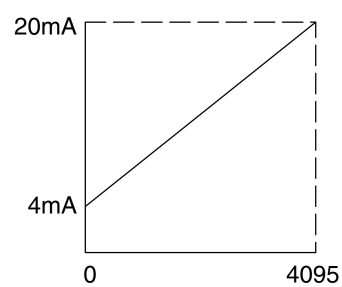


– data bits

The remaining bit (bit 15) is not used and is ignored by the module.

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 4 to 20 mA scale, sending a 0 produces a 4 mA signal, and 4095 gives a 20 mA 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 increases by 1 LSB.

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

H = high limit of the signal range

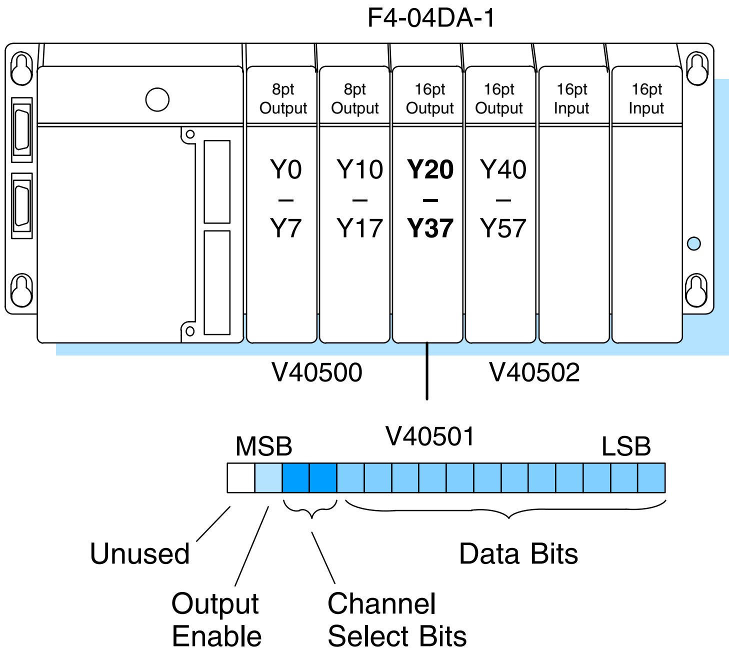
L = low limit of the signal range

Signal Range	Span (H – L)	Divide By	Smallest Change
4 to 20mA	16mA	4095	3.91 µA

Writing the Control Program

Update Any Channel

As mentioned earlier, you can update any channel each 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 almost 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.

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

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.

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

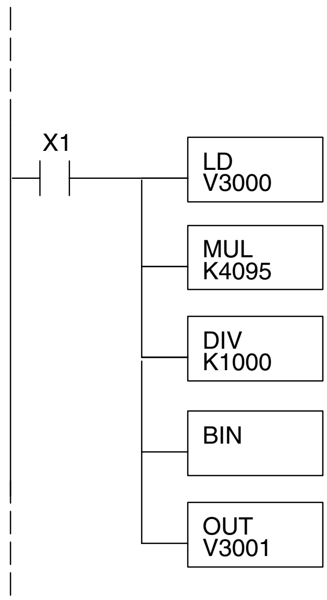
$$A = 494 \frac{4095}{1000 - 0}$$

$$A = 2023$$

430 440 450

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 four 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 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. If you are using binary math, you do not have to include the BIN conversion.



When X1 is on, the engineering units (stored in V3000) are loaded into the accumulator. This example assumes the numbers are BCD.

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).

Convert the BCD number to binary.

Store the result in V3001. This is the digital value, in binary form, that should be sent to the module.

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 location of your analog module. See Appendix A for additional addresses available for the DL450 CPU.

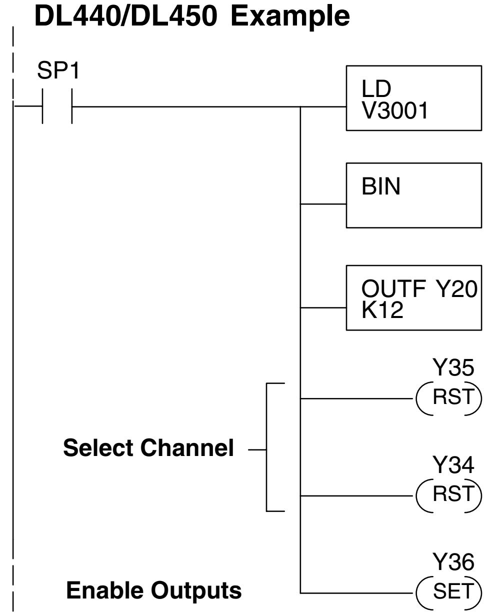
V-Memory Register Addresses for 16-Point Output (Y) Locations										
Y	000	020	040	060	100	120	140	160	200	220
V	40500	40501	40502	40503	40504	40505	40506	40507	40510	40511
Y	240	260	300	320	340	360	400	420	440	460
V	40512	40513	40514	40515	40516	40517	40520	40521	40522	40523

Sending Data to One Channel

The following programs show you how to update a single channel. Notice the DL430 CPU requires a slightly different program than the DL440 and DL450 CPUs. Since the DL430 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. The DL430 example will also work with DL440 and DL450 CPUs. This example assumes you already have the data loaded in V3001.

✗ ✓ ✓
430 440 450

DL440/DL450 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.

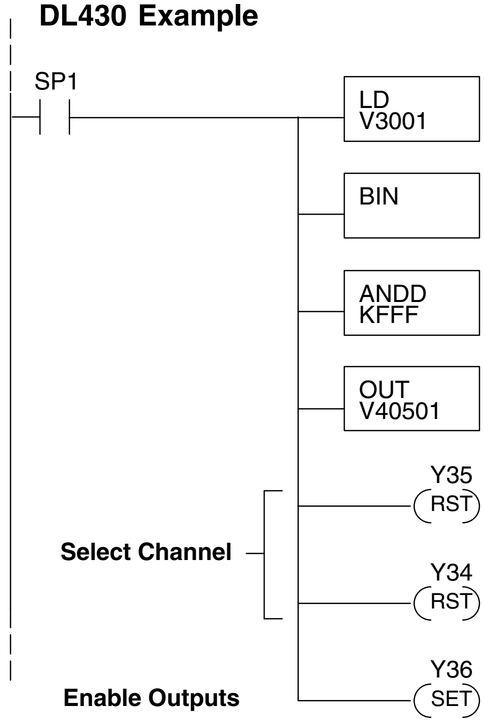
Turn Y35 off and Y34 off to update Channel 1.

Y35	Y34	Channel
Off	Off	Ch. 1
Off	On	Ch. 2
On	Off	Ch. 3
On	On	Ch. 4

Turn on Y36 to enable all four output channels.

✓ ✓ ✓
430 440 450

DL430 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 Y35 off and Y34 off to update Channel 1.

Y35	Y34	Channel
Off	Off	Ch. 1
Off	On	Ch. 2
On	Off	Ch. 3
On	On	Ch. 4

Turn on Y36 to enable all four 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 four channels over four 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 C4 as index numbers corresponding to the channel updated on any particular scan. At the end of each scan, only one control relay C1 through C4 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 1 with DL430 CPUs. Either example will work with DL440 or DL450 CPUs.

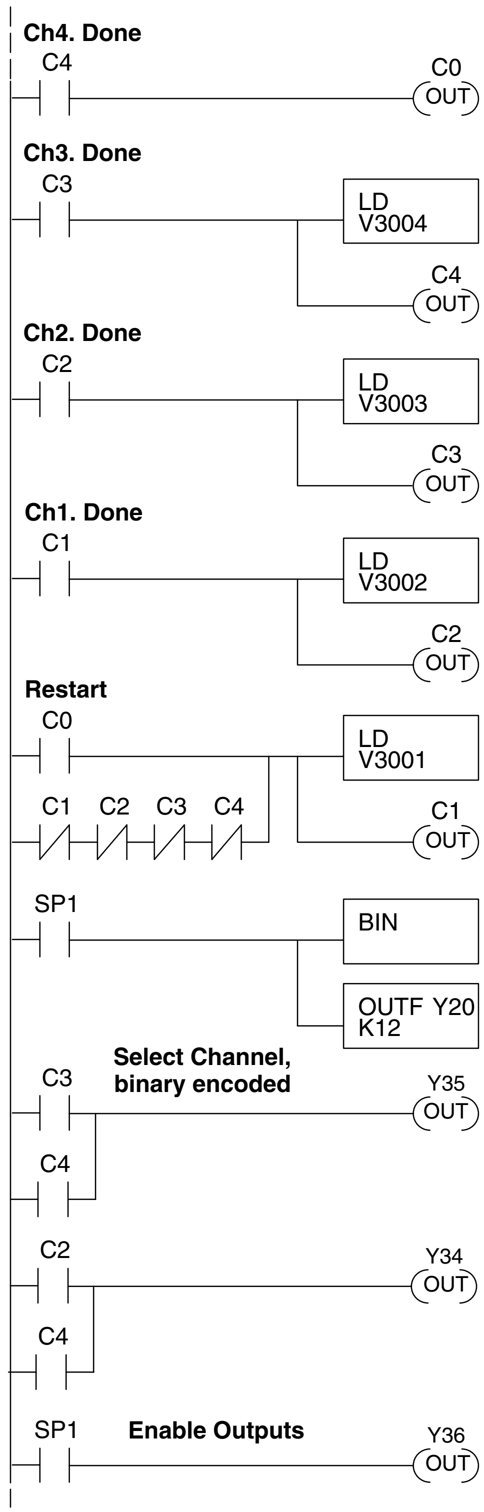
The next two examples, 3 and 4, are slightly more complex. However, they do not depend on the use of control relays to provide channel sequencing. Instead, they use function boxes to increment a channel pointer value in V-memory. Then, other instructions perform bit manipulations to position the channel select bits properly in the output word to the module. You must use example 3 with DL430 CPUs. Either example will work with DL440 or DL450 CPUs.

In the last example, we show how you can update all four channels in the same scan with DL440 and DL450 CPUs. However, this can increase the scan time and you may not always need to update all four channels on every scan.

Sequencing Example 1, DL440/450

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 already have the data loaded in V3001, V3002, V3003, and V3004 for channels 1 – 4 respectively. It is important to use the rungs in the order shown for the program to work. This example will not work with DL430 CPUs.



When channel 4 has been updated, C0 restarts the update sequence.

When channel 3 has been updated, this rung loads the data for channel 4 into the accumulator. By turning on C4, this triggers the channel update (see the channel select rungs below).

When channel 2 has been updated, this rung loads the data for channel 3 into the accumulator. By turning on C3, this triggers the channel update (see the channel select rungs below).

When channel 1 has been updated, this rung loads the data for channel 2 into the accumulator. By turning on C2, this triggers the channel update (see the channel select rungs below).

This rung loads the data for channel 1 into the accumulator. C0 restarts the sequence after channel 4 is done (see the top rung). The first scan or any interruption in control relay sequencing is detected when control relays C1 through C4 are off. In this case, we also start the sequence with channel 1.

This rung converts the accumulator data 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.

Set Y35 and Y34 to select the output channel, based on the control relay status.

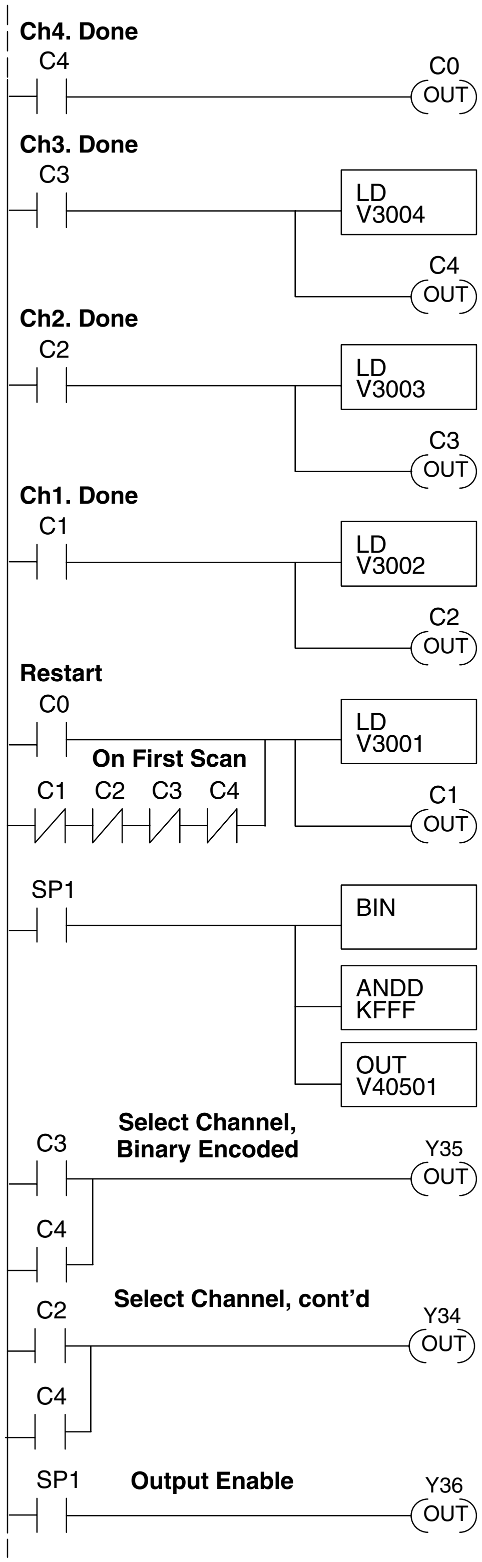
CR(on)	Y35	Y34	Channel
C1	Off	Off	Ch. 1
C2	Off	On	Ch. 2
C3	On	Off	Ch. 3
C4	On	On	Ch. 4

Enables all four output channels. SP1 is always on.

Sequencing Example 2, DL430

430 440 450

Since the DL430 does not support the OUTF instruction, the previous program must be modified to make sure the channel select bits are not accidentally changed by the data in the accumulator. It is important to use the rungs in the order shown for the program to work. This example will also work with DL440 and DL450 CPUs.



When channel 4 has been updated, C0 restarts the update sequence.

When channel 3 has been updated, this rung loads the data for channel 4 into the accumulator. By turning on C4, this triggers the channel update (see the channel select rungs below).

When channel 2 has been updated, this rung loads the data for channel 3 into the accumulator. By turning on C3, this triggers the channel update (see the channel select rungs below).

When channel 1 has been updated, this rung loads the data for channel 2 into the accumulator. By turning on C2, this triggers the channel update (see the channel select rungs below).

This rung loads the data for channel 1 into the accumulator. C0 restarts the sequence after channel 4 is done (see the top rung). The first scan or any interruption in control relay sequencing is detected when control relays C1 through C4 are off. In this case, we also start the sequence with channel 1.

This rung 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.

Set Y35 and Y34 to select the output channel, based on the control relay status.

CR(on)	Y35	Y34	Channel
C1	Off	Off	Ch. 1
C2	Off	On	Ch. 2
C3	On	Off	Ch. 3
C4	On	On	Ch. 4

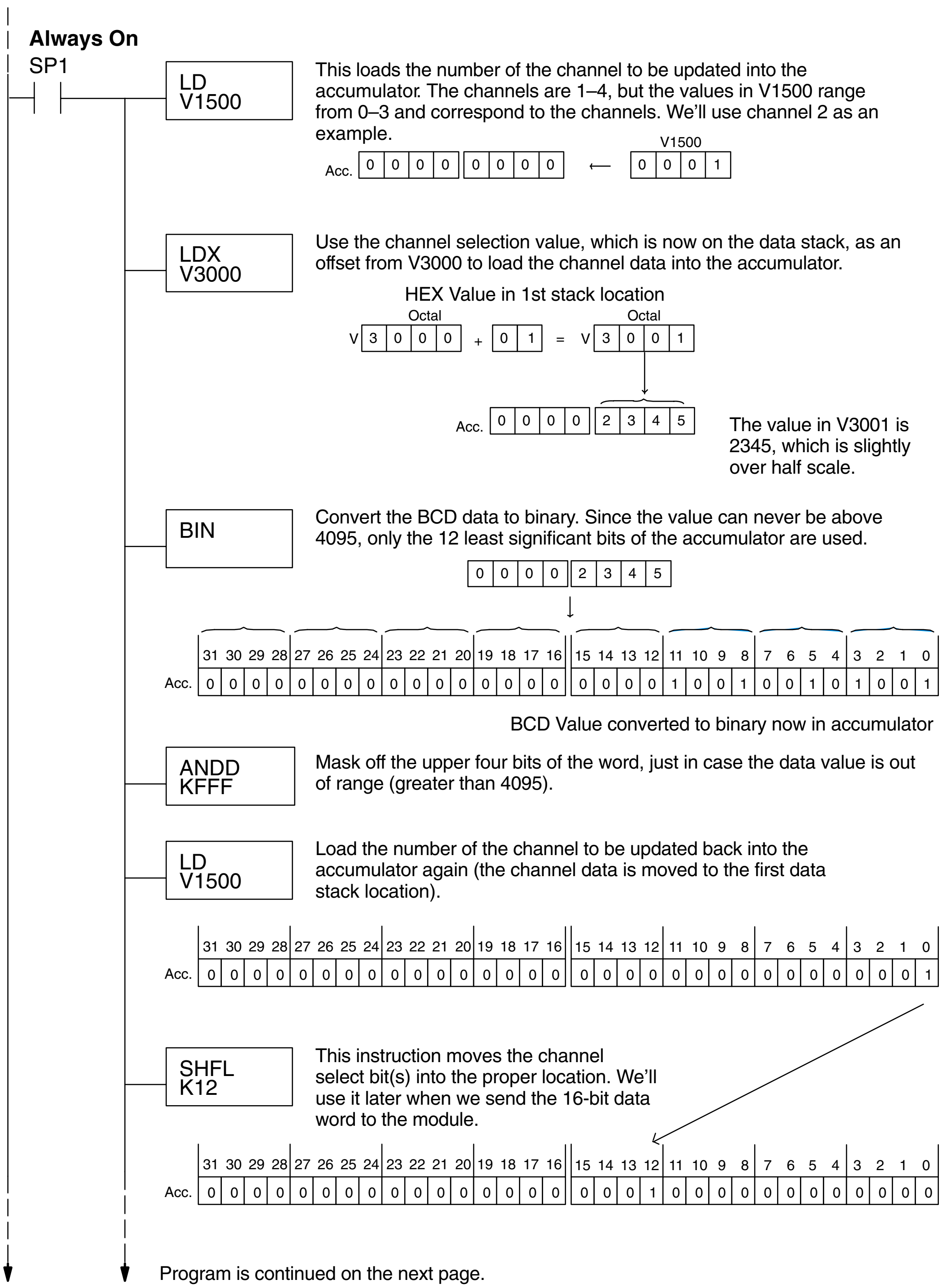
Enables all four output channels. SP1 is always on.

Sequencing Example 3, DL440/DL450

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 will not work with DL430 CPUs. This example assumes you are using the following data locations.

- V3000 – channel 1 data V3001 – channel 2 data
- V3002 – channel 3 data V3004 – channel 4 data
- V1500 – channel to update: 0 = ch. 1, 1 = ch. 2, 2 = ch. 3, 3 = ch. 4



Example 3 Continued

ORD K4000

Set the Output Enable bit, by combining the value of 4000 hex with the accumulator value. This sets bit 14 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	0	1	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	0	1	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	0	1	0	1	1	0	0	1	0	0	1	0	1	0	0	1

Data for Analog Module

OUTF Y20 K15

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

INCB V1500

Increment the channel selection value. This allows the logic to cycle through all four channels.

V1500 = K4 → **LD K0**

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

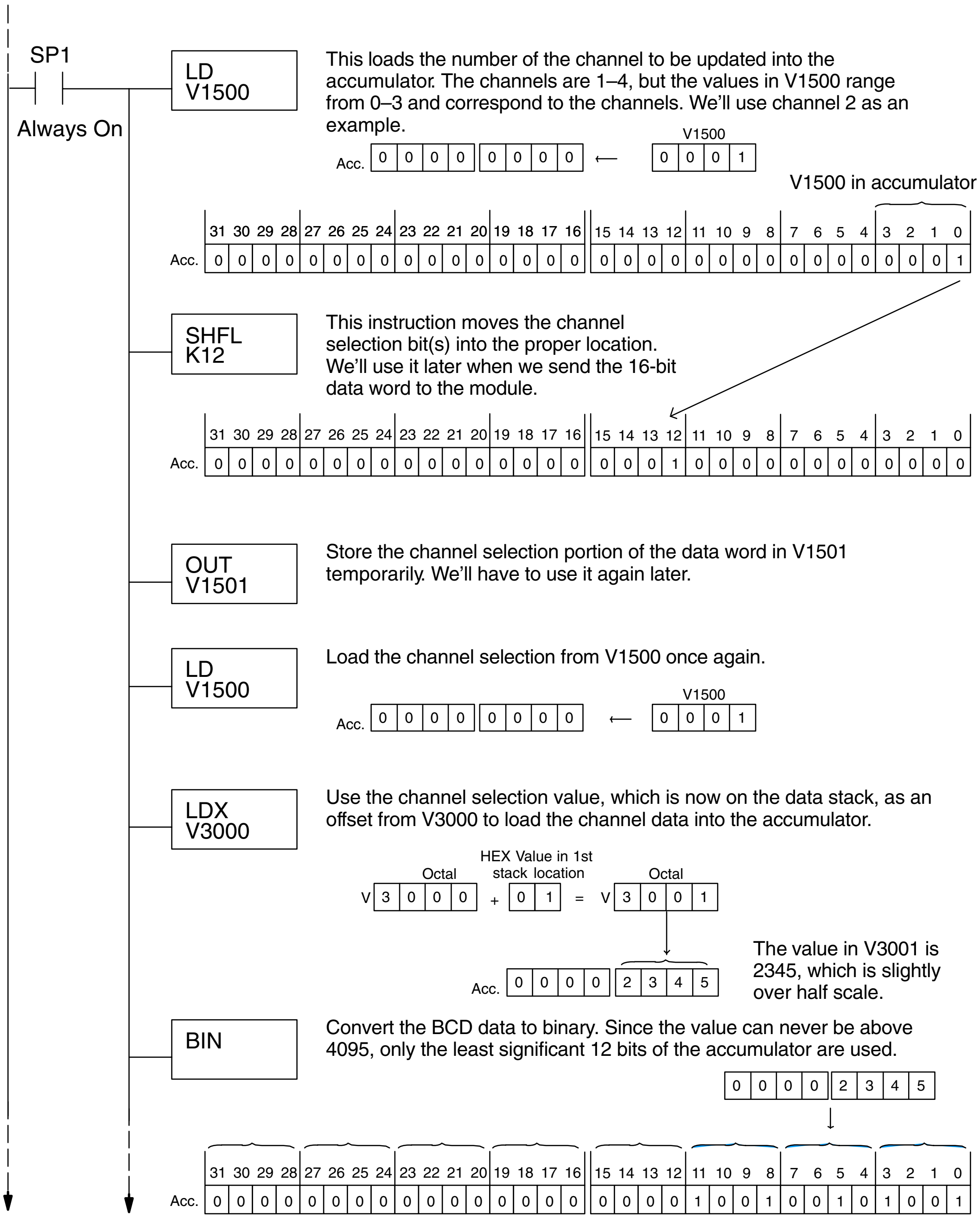
OUT V1500

Sequencing Example 4, DL430

✓ 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 will also work with DL440 and DL450 CPUs. This example assumes you are using the following data locations.

- V3000 – channel 1 data V3001 – channel 2 data
- V3002 – channel 3 data V3004 – channel 4 data
- V1500 – channel to update: 0 = ch. 1, 1 = ch. 2, 2 = ch. 3, 3 = ch. 4
- V1501 – temporary location for the channel selection



Program is continued on the next page.

Example 4 Continued

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 be sent 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
K4000

Set the Output Enable bit, by combining the value of 4000 hex with the accumulator value. This sets bit 14 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	0	1	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.

INCB
V1500

Increment the channel selection value. This allows the logic to cycle through all four channels.

V1500 K4
=

LD
K0

When channel four has been updated, this instruction resets the channel selection memory location to 0 (remember, 0 represents channel 1).

OUT
V1500

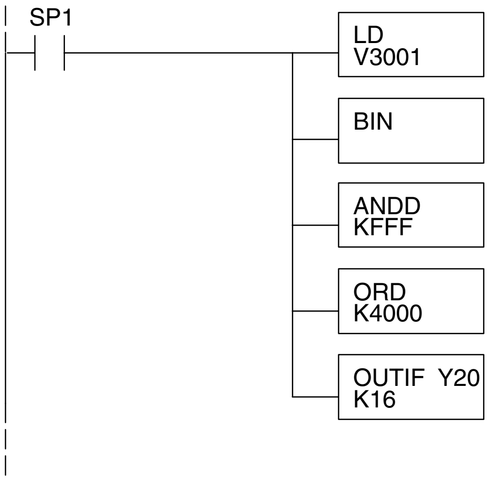
Updating all Channels in a Single Scan

430 440 450

By using the Immediate instructions found in the DL440 and DL450 CPUs (not DL430s), you can easily update all four channels in a single scan. Before choosing this method, remember that it slows the CPU scan time. 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 V3001, V3002, V3003, and V3004 for channels 1–4 respectively.

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

Channel 1 Example



The LD instruction loads the data for channel 1 into the accumulator.

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.

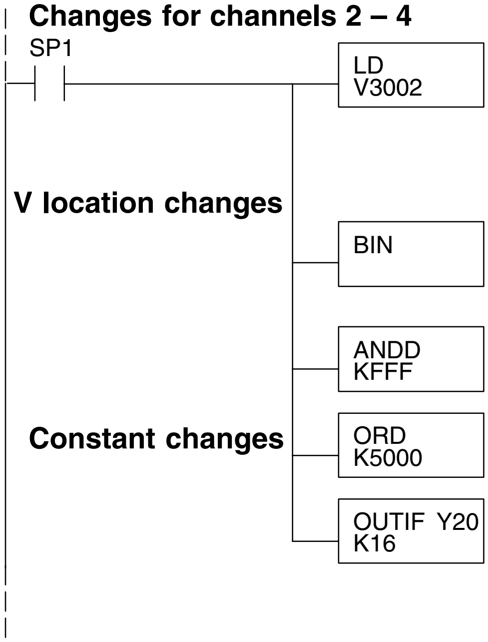
The ORD instruction (with K4000) sets the Output Enable bit. Outputs Y34 and Y35 are left off to select channel 1 for updating with the data.

The OUTIF sends 16 bits to the data word. Our example starts with Y20, but the actual value depends on the location of the module in your application.

You have to send 16 bits with the OUTIF instruction. If you don't send all 16 bits, the module will ignore the data.

The remaining channels are updated with a similar program segment. The only changes are the location of the data for each channel (V3002, V3003, and V3004) and the ORD instruction. The constant loaded with the ORD instruction is different for each channel. The following example shows where these differences occur.

Changes for channels 2 – 4



The LD instruction loads the data for channel 2 into the accumulator.

Location	Channel
V3001	1
V3002	2
V3003	3
V3004	4

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

Mask off the upper four bits, so bad data cannot corrupt the channel select bits, output enable bit, or sign bit.

The ORD instruction with the constants as specified selects the appropriate channel to be updated, and sets the Output Enable bit. The following constants are used.

Constant	Channel
K 4000	1
K 5000	2
K 6000	3
K 7000	4

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 ...
4 to 20mA	$A = \frac{16D}{4095} + 4$	$D = \frac{4095}{16}(A - 4)$

For example, if you need a 9mA signal level, you would use the following formula to determine the digital value that should be stored in the V-memory location that contains the data.

$$D = \frac{4095}{16}(A–4)$$

$$D = \frac{4095}{16}(9mA–4)$$

$$D = (255.94)\ (5)$$

$$D = 1280$$