17

F4–04DAS–1 4-Channel Isolated 4–20mA Output

In This Chapter. . . .

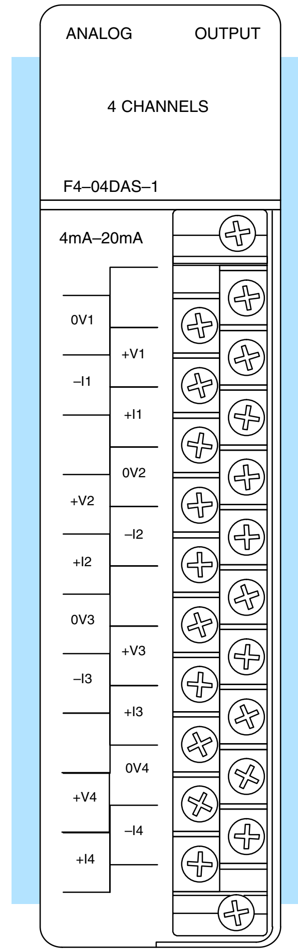
— Module Specifications
— Connecting the Field Wiring
— Module Operation
— Writing the Control Program

Module Specifications

The F4–04DAS–1 4-channel Isolated Analog Output module provides several features and benefits.

- Each analog output is isolated from the other 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).
- Provides four channels of isolated current outputs if used with independent loop power supplies.

Firmware Requirements:
When using this module with an H4–EBC, the H4–EBC must have firmware version 2.1.46 or later.



Analog Output Configuration Requirements

The F4–04DAS–1 Analog Output requires 32 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 table provides the specifications for the F4–04DAS–1 Analog Output Module. Review these specifications to ensure the module meets your application requirements.

Output Specifications

Number of Channels	4, isolated current sourcing
Output Ranges	4–20mA
Resolution	16 bit (1 in 65536)
Isolation Voltage	±750V continuous, channel to channel, channel to logic
Load Impedance	0–525Ω
Loop Supply	18–32VDC
Linearity Error (end to end)	±10 counts (±0.015%) of full scale
Offset Calibration Error	±13 counts (±0.02%)
Full Scale Calibration Error	±8 counts maximum (offset error included)
Maximum Inaccuracy	±0.07% at 25°C (77°F) ±0.18% at 0 to 60°C (32 to 140°F)
Conversion Settling Time	3 ms to 0.1% of full scale

General Module Specifications

Digital Output Output Points Required	16 data bits, 2 channel ID, 1 output enable 32(Y) output points
Power Budget Requirement	100mA @ 5 VDC (from base)
External Power Supply	50mA per channel, class 2
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

Connecting the Field Wiring

Wiring Guidelines

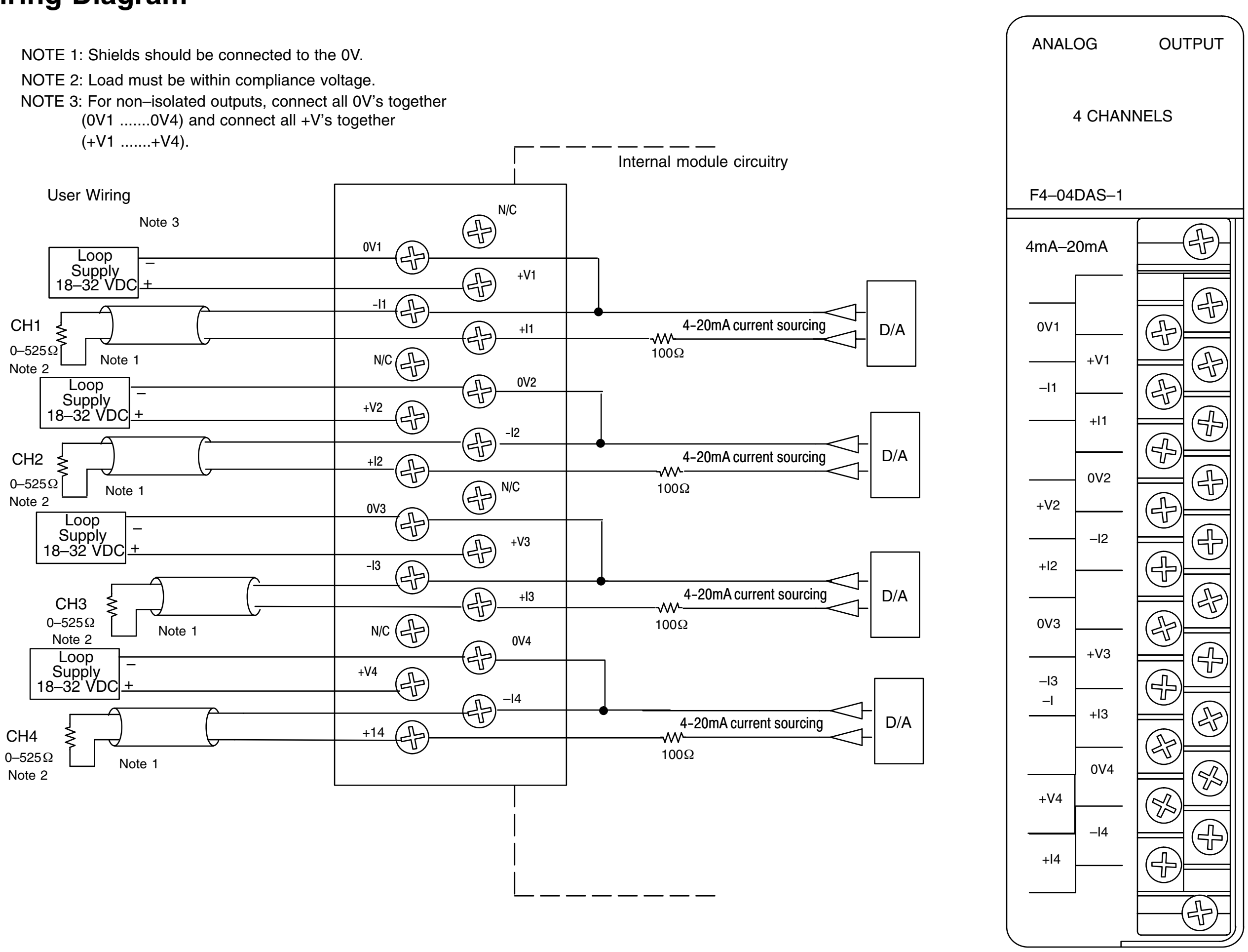
Your company may have guidelines for wiring and cable installation. If so, you should check those before you begin the installation. Here are some general things to consider.

- 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.

Removable Connector

The F4–04DAS–1 module has a removable connector to make wiring easier. Simply remove the retaining screws and gently pull the connector from the module.

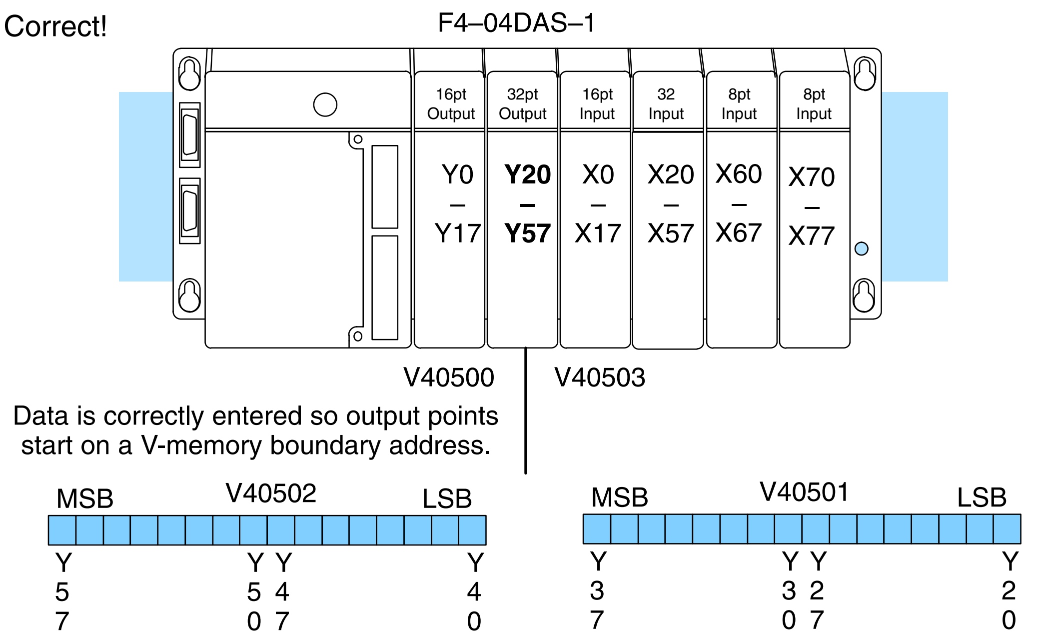
Wiring Diagram

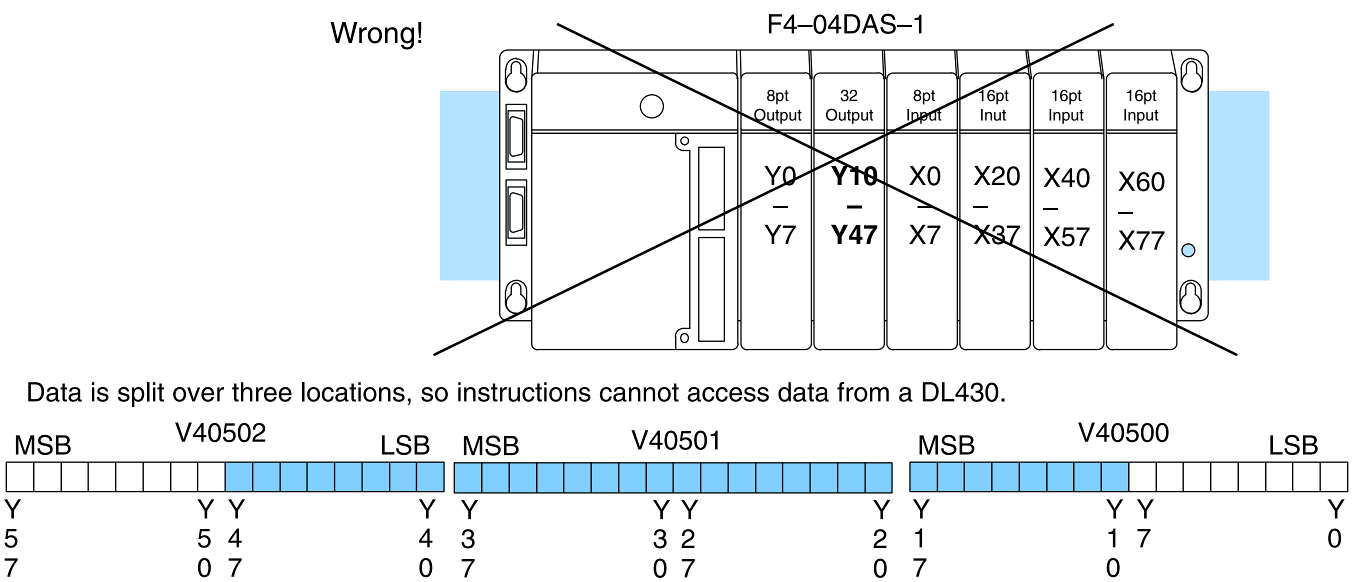


Module Operation

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.

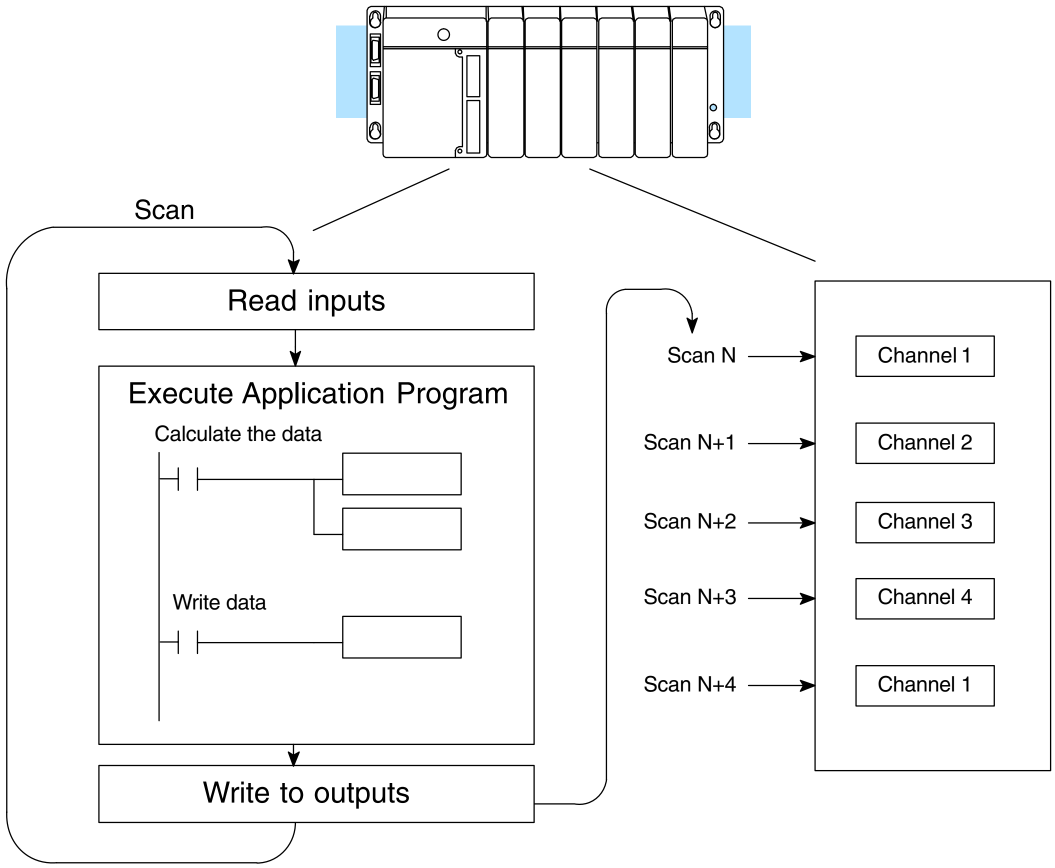




Before you begin writing the control program, it is important to take a few minutes to understand how the module processes and represents the analog signals.

Channel Scanning Sequence

The F4–04DAS–1 module allows you to update the channels in any order. Your control program determines which channel gets updated on any given scan by using two binary encoded output points. With a DL440 or DL450 CPU, you can use immediate instructions to update all four channels in the same scan (we will show you how to do this later).

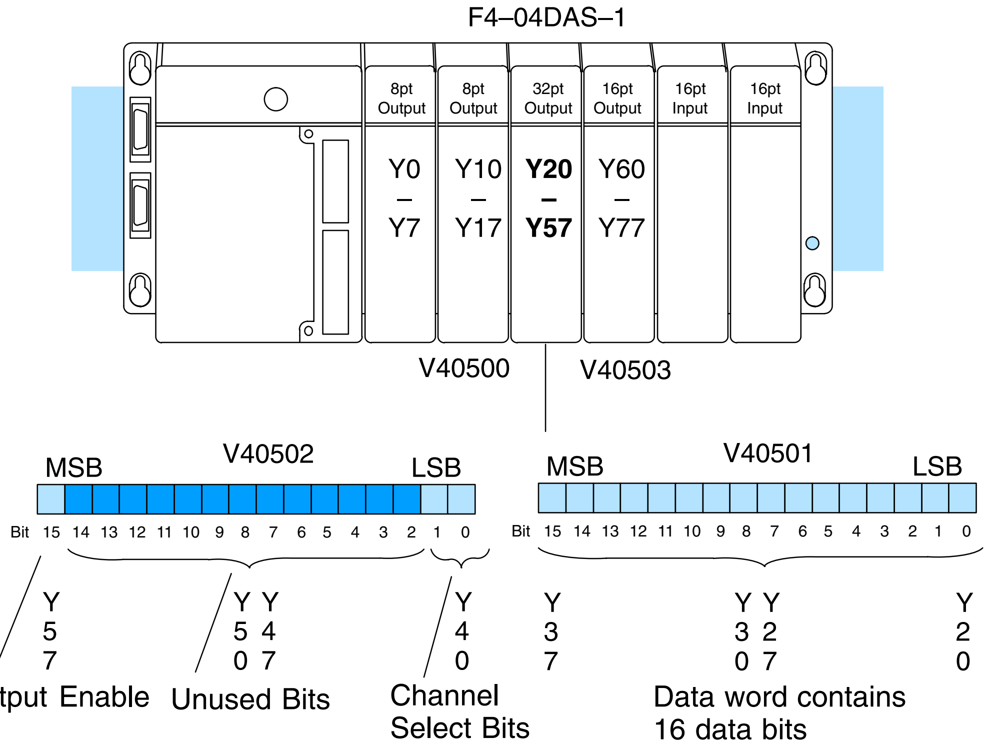


Output Bit Assignments

You may recall the F4–04DAS–1 module requires 32 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.

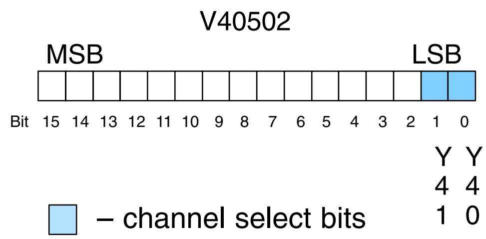
Since all output points are automatically mapped into V-memory, it is very easy to determine the location of the data word that will be assigned to the module.



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

Channel Select Bits

Bits 16 and 17 are binary encoded to select the channel that will be updated with the data. The bits are assigned as follows.

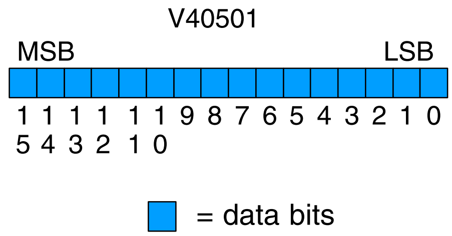


Y41	Y40	Channel Number
0	0	1
0	1	2
1	0	3
1	1	4

Analog Data Bits

The first sixteen bits of the V-memory location represent the analog data in binary format.

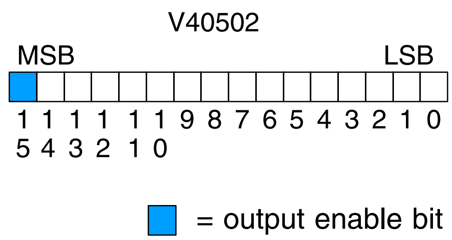
Bit	Value	Bit	Value
0	1	8	256
1	2	9	512
2	4	10	1024
3	8	11	2048
4	16	12	4096
5	32	13	8192
6	64	14	16384
7	128	15	32768



Output Enable Bit

The most significant bit of the second word (or MSW) is the Output Enable Bit. Turning it on enables all four channels to be updated. Turning it off causes all output signal levels to go to 4mA and clears the module's internal data registers for all channels.

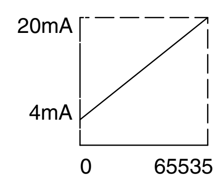
After an off-to-on transition of this bit, each output stays at 4mA until the channel and the CPU writes a non-zero value to it.



Module Resolution

Since the module has 16-bit resolution, the analog signal is converted into 65536 counts ranging from 0 – 65535 (2^{16}). For example, send a 0 to get a 4mA signal and 65535 to get a 20mA signal. This is equivalent to a binary value of 0000 0000 0000 0000 to 1111 1111 1111 1111, or 0000 to FFFF hexadecimal. The diagram shows how this relates to the signal range.

4 – 20mA



$$\text{Resolution} = \frac{H - L}{65535}$$

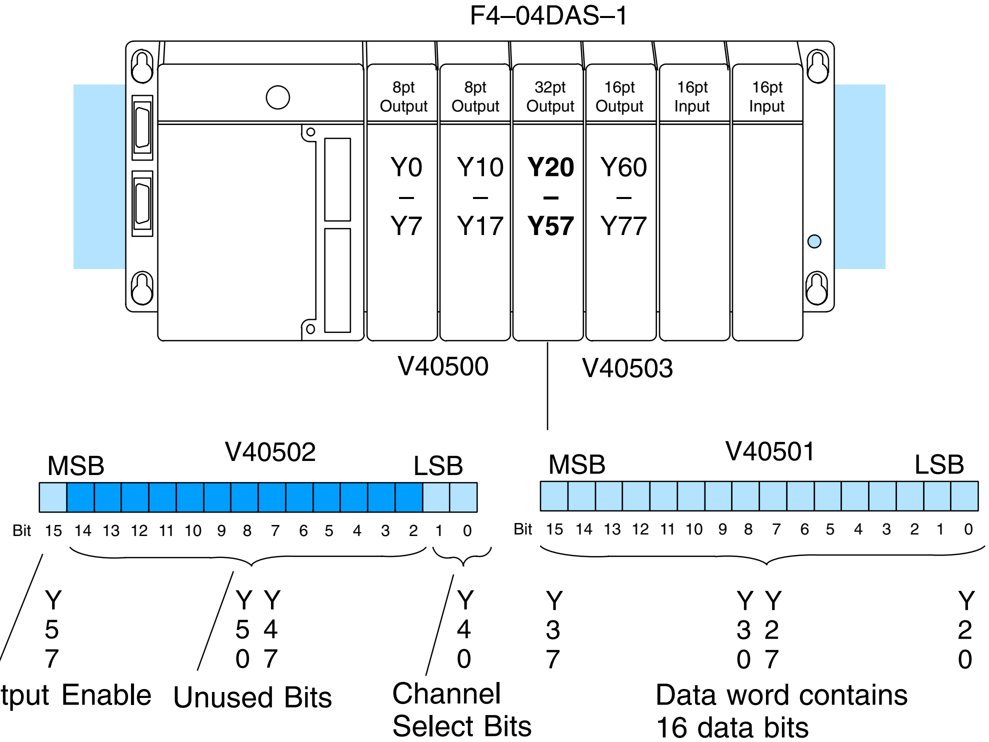
H = high limit of the signal range

L = low limit of the signal range

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 selection 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.

$$A = U \frac{65535}{H - L}$$

A = analog value (0 – 65535)

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.

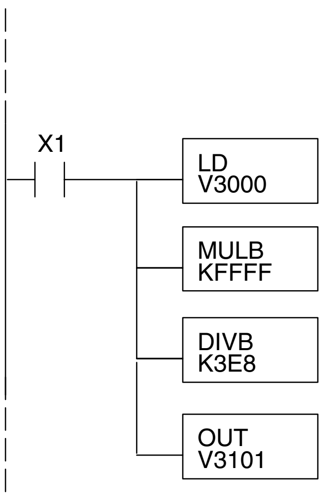
$$A = 10U \frac{65535}{10(H - L)} \qquad A = 494 \frac{65535}{1000 - 0} \qquad A = 32374$$

Engineering Unit Conversion

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. When using this module, it is usually easier to perform any math calculations in binary because of the large numbers involved.



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

Multiply the accumulator by 65535 (to start the conversion).

Divide the accumulator by 1000 (3E8 hex, because we used a multiplier of 10, we have to use 1000 instead of 100).

Store the result in V3101. 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 for DL450 CPUs.

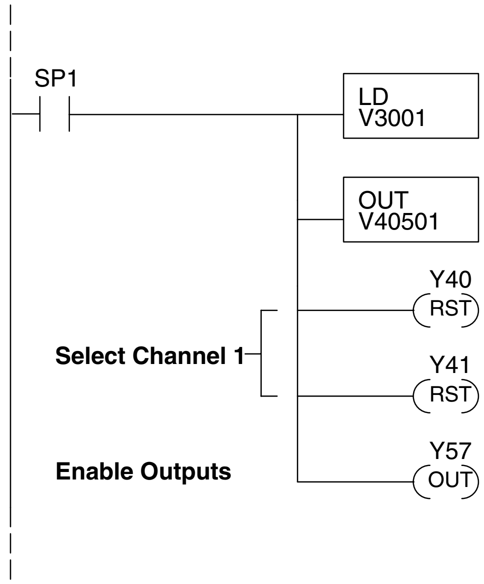
V-Memory Register Addresses for 16-Point Output (Y) Locations										
Y	000	020	040	060	100	120	140	160	200	220
V	40500	40501	40502	40503	40504	40505	40506	40507	40510	40511
Y	240	260	300	320	340	360	400	420	440	460
V	40512	40513	40514	40515	40516	40517	40520	40521	40522	40523

Sending Data to One Channel

✓ ✓ ✓
430 440 450

The following programs show you how to update a single channel. Notice that the BCD method uses a slightly different program than the binary method . Both examples assume you already have the data loaded in V3001.

Binary Example



Data is in a range of 0–FFFF (hex).

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.

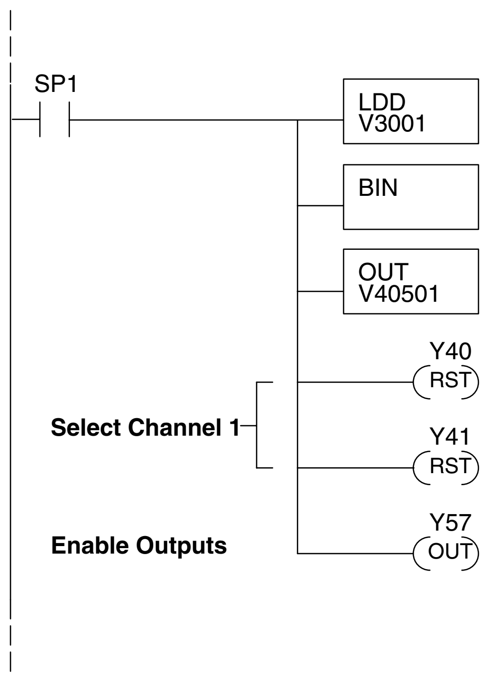
The OUT sends the 16 bits to the data word. Our example starts with Y20, but the actual value depends on the location of the module in your application.

Select channel 1 for updating.

Y41	Y40	Channel
Off	Off	Ch. 1
Off	On	Ch. 2
On	Off	Ch. 3
On	On	Ch. 4

Turn on the output enable bit, to enable all output channels.

BCD Example



Data is in a range of 0–65535 (2 words).

The LDD 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.

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.

Select channel 1 for updating.

Y41	Y40	Channel
Off	Off	Ch. 1
Off	On	Ch. 2
On	Off	Ch. 3
On	On	Ch. 4

Turn on the output enable bit, to enable all output channels.

Sequencing the Channel Updates

The next three example programs show you how to send digital values to the module when you have more than one channel. The first two examples will automatically update all four channels over four scans, while the last example updates all four channels in one scan.

The first sequencing example is fairly simple and will work in almost all situations. We recommend it for new users. It uses 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.

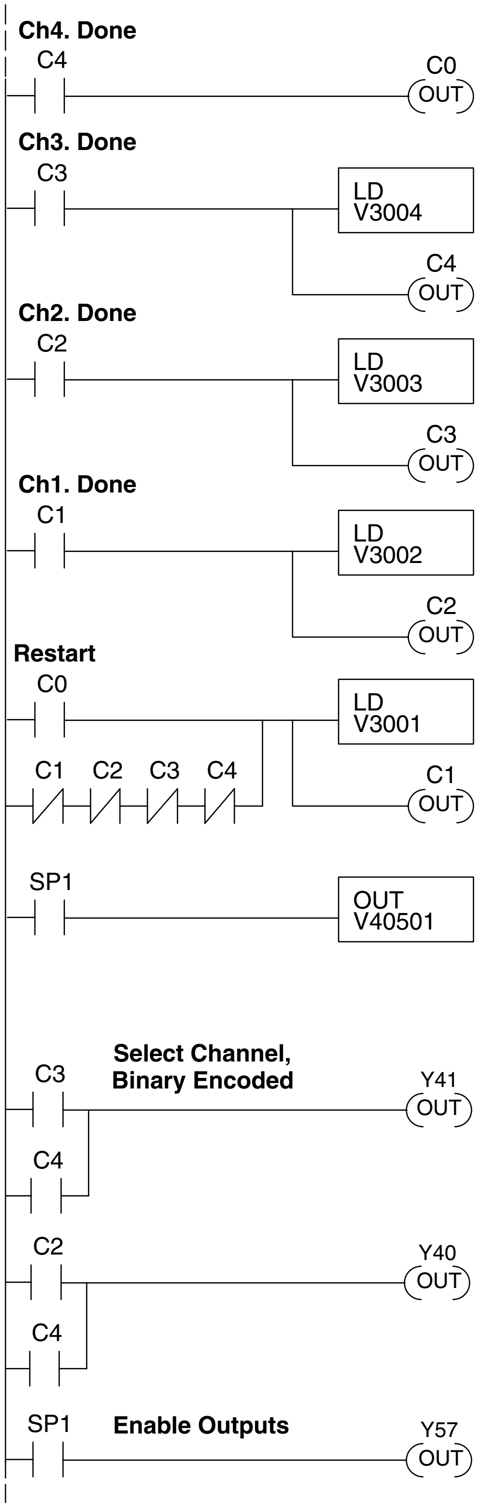
The second example is slightly more complex. However, it does not depend on the use of control relays to provide channel sequencing. Instead, it uses function boxes to increment a channel pointer value in V-memory. Then, other instructions perform bit manipulations to position the channel select bits properly in the output word to the module.

In the last example, we show you 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, DL430/440/450

430 440 450

This 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 binary format in V3001, V3002, V3003, and V3004 for channels 1 – 4 respectively (note that these locations are in a range of 0–FFFF hex). It is important to use the rungs in the order shown for the program to work.



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).

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).

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 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 Y41 and Y40 to select the output channel, based on the control relay status.

CR(on)	Y41	Y40	Channel
C1	Off	Off	Ch. 1
C2	Off	On	Ch. 2
C3	On	Off	Ch. 3
C4	On	On	Ch. 4

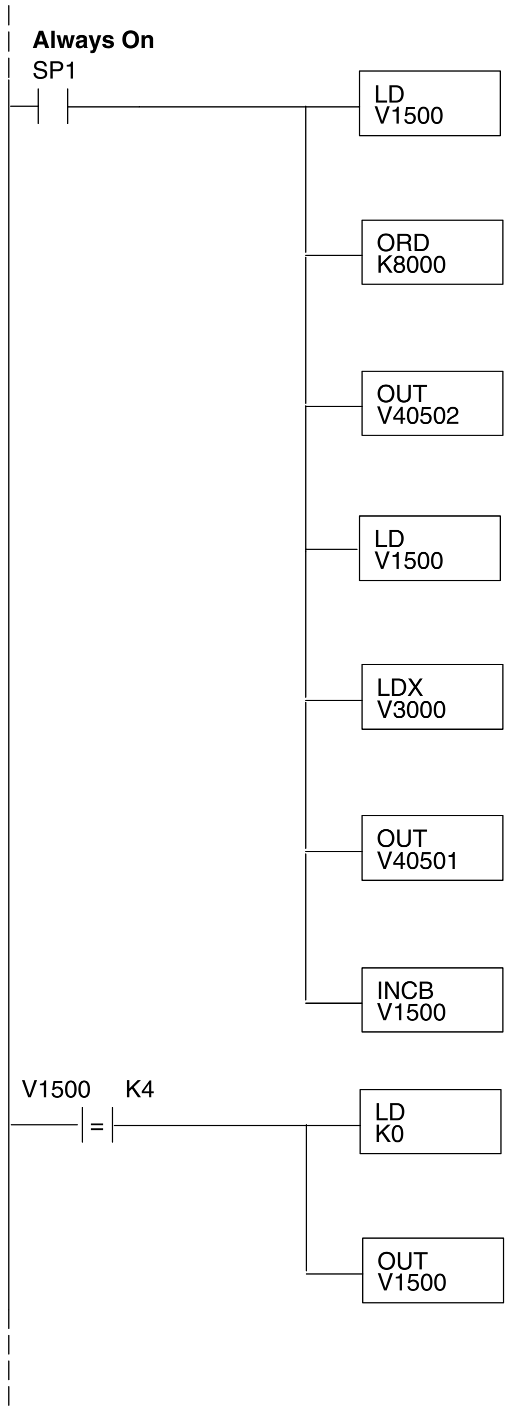
Enables all four output channels. SP1 is always on.

Sequencing Example 2, DL430/440/450

✓ ✓ ✓
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 the data in binary format and are using the following data locations.

- V3000 – channel 1 data V3001 – channel 2 data
- V3002 – channel 3 data V3003 – channel 4 data
- V1500 – channel to update: 0 = ch. 1, 1 = ch. 2, 2 = ch. 3, 3 = ch. 4



LD V1500: This loads the number of the channel to be updated into the accumulator. The channels are 1–4, but the values in V1500 range from 0–3 and correspond to the channels.

ORD K8000: Logically ORs the value in the accumulator with the constant 8000, which sets the Output Enable Bit.

OUT V40502: The result is stored in this location.

LD V1500: Again load the channel selection from V1500 back into the accumulator.

LDX V3000: Use the channel selection value as an offset from V3000 to load the channel data into the accumulator.

OUT V40501: Sends 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).

INCB V1500: Increments the channel selection value. This allows the logic to cycle through all four channels.

LD K0 / OUT V1500: When channel 4 has been updated, this instruction resets the channel selection memory location to 0 (0 is for channel 1).

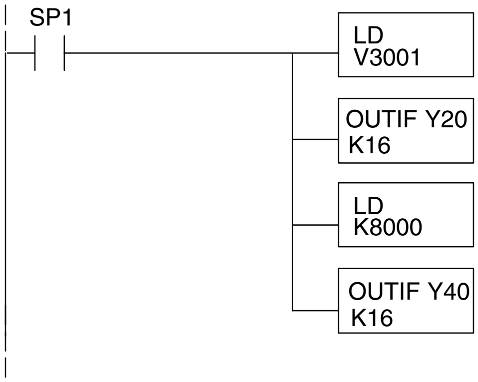
Updating all Channels in a Single Scan, DL440/450

430 440 450

By using the Immediate instructions found in the DL440 and DL450 CPUs, you can easily update all four channels in a single scan. Before choosing this method, remember the Immediate instructions slow 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 are using binary format and already have the data loaded in V3001, V3002, V3003, and V3004 for channels 1 – 4 respectively. This example will not work with DL430 CPUs.

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 into the accumulator. Specifiying V3001 selects channel 1.

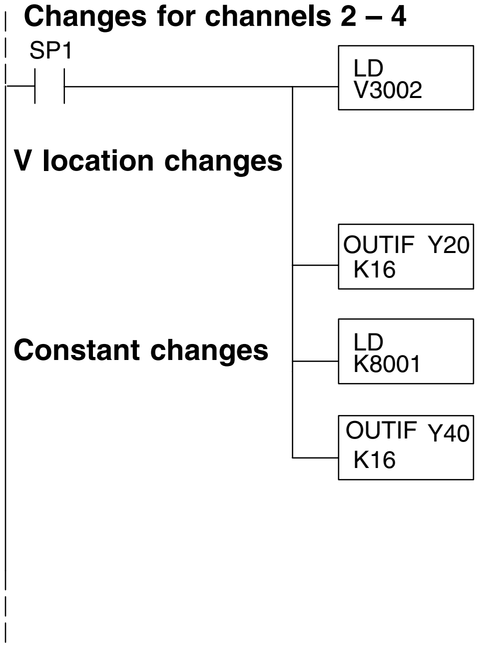
The OUTIF instruction sends 16 bits to the data word. Our example starts with Y20, but the actual value depends on the location of the module in your application.

Loads the constant 8000 into the accumulator.

The OUTIF instruction sends 16 bits to the data word. Our example starts with Y40, but the actual value depends on the location of the module in your application.

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 second LD instruction. The constant loaded with the second LD 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 into the accumulator. Specifying V3002 selects channel 2. Here are the locations for each of the four channels.

Location	Channel
V3001	1
V3002	2
V3003	3
V3004	4

The OUTIF instruction 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.

Loads the constant 8001 into the accumulator.

The OUTIF instruction sends 16 bits to the data word. Our example starts with Y40, but the actual value depends on the location of the module in your application. The following constants are used.

Constant	Channel
K 8000	1
K 8001	2
K 8002	3
K 8003	4

Analog and Digital Value Conversions

Sometimes it is helpful to be able to quickly convert between the 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.

Current Range	**Output Format**	**If you know the digital value ...**	**If you know the analog signal level ...**
4 – 20mA	0 to 65535	$A = \frac{16D}{65535} + 4$	$D = \frac{65535}{16}(A - 4)$