

# **F2-08AD-2**

## **8-Channel Analog Voltage Input**

---

**5**

In This Chapter. . . .

- Module Specifications
  - Setting the Module Jumpers
  - Connecting the Field Wiring
  - Module Operation
  - Writing the Control Program
-

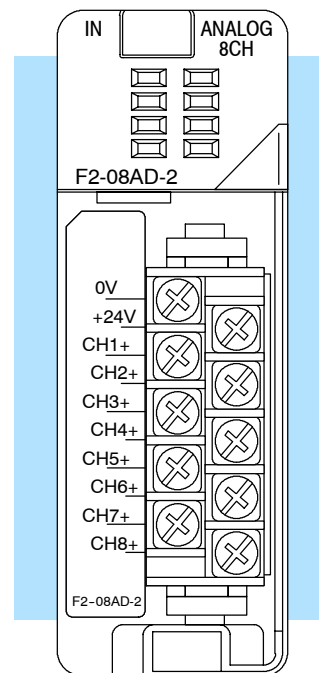
## Module Specifications

The F2-08AD-2 Analog Voltage Input module provides several hardware features:

- Analog inputs 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 all channels in one scan.
- Low-power CMOS design requires less than 80mA from an external 18–26.4 VDC power supply.

### Firmware Requirements:

To use this module, D2-230 CPUs must have firmware version 1.6 or later. To use the pointer method of writing values, D2-240 CPUs require firmware version 2.2 or later and D2-250 CPUs require firmware version 1.06 or later.



The following tables provide the specifications for the F2-08AD-2 Analog Input Module. Review these specifications to make sure the module meets your application requirements.

### Input Specifications

Number of Channels	8, single ended (one common)
Input Ranges	0 - 5V, 0 - 10V, $\pm 5V$ , $\pm 10V$ .
Resolution	12 bit (1 in 4096) unipolar (0 - 4095) 13 bit (1 in 8192) bipolar (-4095 - +4095)
Step Response	4 mS to 95% of full step change
Crosstalk	-70 dB, 1 count maximum
Input Filtering	$f_{-3dB}$ @ 200Hz, -6 dB / octave
Input Impedance	> 20M $\Omega$
Maximum Continuous Overload	-75 VDC to +75 VDC
Linearity Error (End to End)	$\pm 0.025\%$ of span ( $\pm 1$ count maximum unipolar) ( $\pm 2$ count maximum bipolar)
Input Stability	$\pm 1$ count
Full Scale Calibration Error (Offset error not included)	$\pm 3$ counts maximum
Offset Calibration Error	$\pm 1$ count maximum, @ 0 VDC
Maximum Inaccuracy	$\pm .1\%$ @ 25°C $\pm .3\%$ 0 to 60°C (32 to 140°F)
Accuracy vs. Temperature	$\pm 50$ ppm/°C maximum full scale calibration (including maximum offset change of 2 counts)

### General Specifications

PLC Update Rate	1 channel per scan maximum (DL230 CPU) 8 channels per scan maximum (DL240/250-1/260 CPU)
Data Acquisition Time	3ms/channel (asynchronous)
Digital Inputs	12 binary data bits, 1 sign bit, 3 channel ID bits, 1 diagnostic bit
Input points required	16 point (X) input module
Power Budget Requirement	60 mA maximum, 5 VDC (supplied by base)
External Power Supply	80 mA maximum, 18 to 26.4 VDC
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).

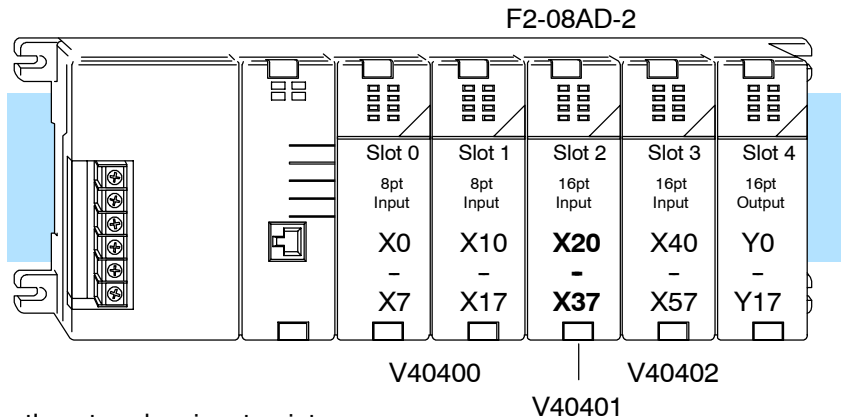
### Analog Input Configuration Requirements

The F2-08AD-2 Analog Input appears as a 16-point discrete input module. The module can be installed in any slot of a DL205 system. The available power budget and discrete I/O points are the 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 expansion or remote I/O points.

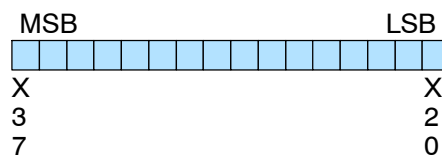
**Special Placement Requirements (DL230 and Remote I/O Bases)**

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 will see in the section on writing the program, you use V-memory locations to extract the analog data. If you place the module so that the input 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 base using a D2-RSSS in the CPU slot.

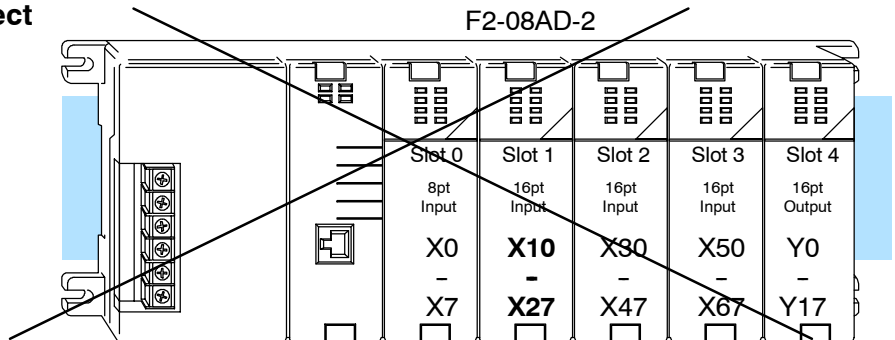
**Correct!**



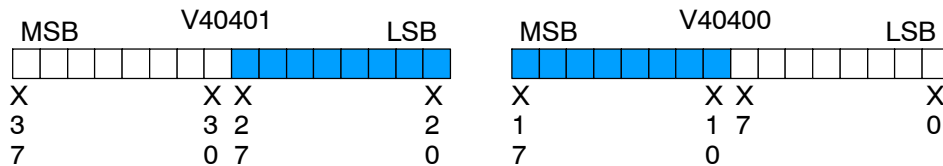
Data is correctly entered so input points start on a V-memory boundary.



**Incorrect**



Data is split over two locations, so instructions cannot access data from a DL230.



To use the required V-memory references, the *first* input address assigned to the module must be one of the following X locations. The table also shows the V-memory addresses that correspond to these X locations.

<b>X</b>	X0	X20	X40	X60	X100	X120	X140	X160
<b>V</b>	V40400	V40401	V40402	V40403	V40404	V40405	V40406	V40407

## Setting the Module Jumpers

### Selecting the Number of Channels

There are three jumpers, labeled +1, +2, and +4 that are used to select the number of channels that will be used. These jumpers are located on the motherboard, the one *with* the black D-shell style backplane connector. The module is set from the factory for eight channel operation (all three jumpers installed).

Any unused channels are not processed. For example, if you only select channels 1 thru 3, channels 4 thru 8 will not be active. The following table shows how to set the jumpers to select the number of channels.

No. of Channels	+1	+2	+4
1	No	No	No
1,2	Yes	No	No
1,2,3	No	Yes	No
1,2,3,4	Yes	Yes	No
1,2,3,4,5	No	No	Yes
1,2,3,4,5,6	Yes	No	Yes
1,2,3,4,5,6,7	No	Yes	Yes
1,2,3,4,5,6,7,8	Yes	Yes	Yes

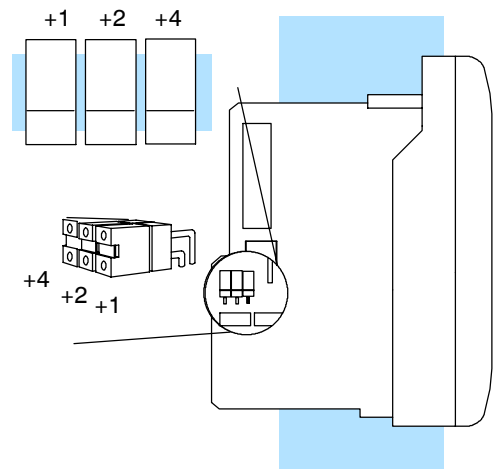
Yes = jumper installed

No = jumper removed

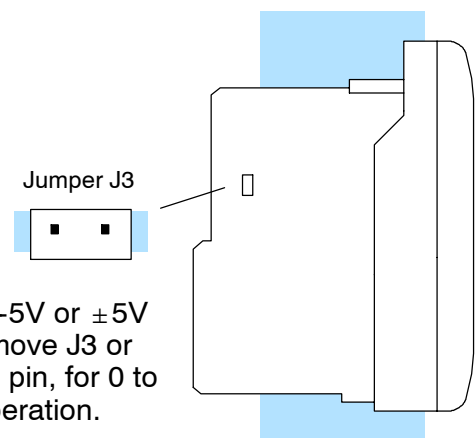
### Selecting Input Voltage Range

If you examine the smaller circuit board, which is on top of the motherboard, you will notice a jumper labeled "J3." Use this jumper to select between the 5V ranges and the 10V ranges. The module comes from the factory set for 10V operation (jumper is removed and is stored on one of the pins).

Install J3 for 0-5V or  $\pm 5V$  operation. Remove J3 or store on single pin, for 0 to 10 or  $\pm 10V$  operation.



For example, to select 8-channel operation, leave all three jumpers installed. To select only channel 1, remove (or store on a single post to prevent losing them) all three jumpers.



## Connecting the Field Wiring

### Wiring Guidelines

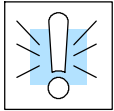
Your company may have guidelines for wiring and cable installation. If so, you should check those before you begin the installation. Here are some general things to consider:

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the transmitter source. *Do not* ground the shield at both the module and the source.
- 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

You may use the same or separate power source for the transmitter voltage supply. The DL205 bases have built-in 24 VDC power supplies that provide up to 300mA of current. You may use this instead of a separate supply if you are using only a couple of analog modules.

It is desirable in some situations to power the transmitters separately in a location remote from the PLC. This will work as long as the transmitter supply meets the voltage and current requirements, and the transmitter's minus (-) side and the module supply's minus (-) 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.

---

The DL205 base has a switching type power supply. As a result of switching noise, you may notice  $\pm 3-5$  counts of instability in the analog input data if you use the base power supply. If this is unacceptable, try one of the following:

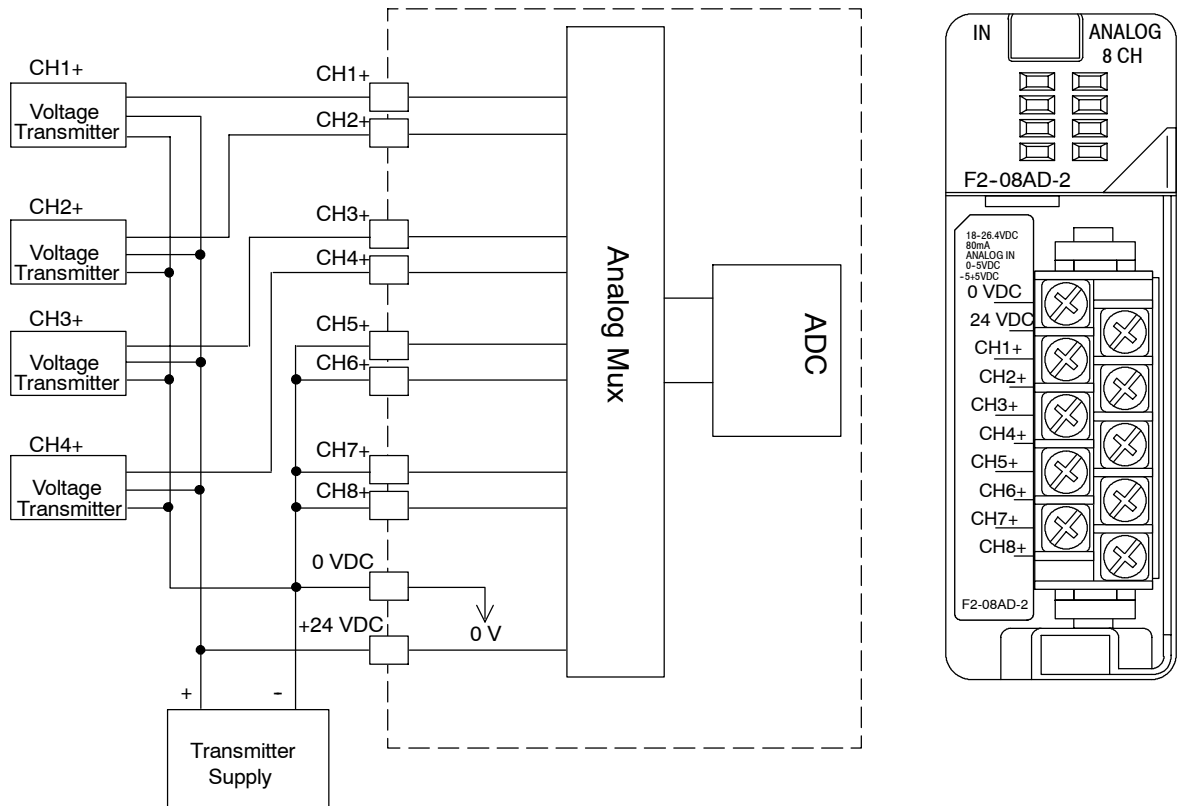
1. Use a separate linear power supply.
2. Connect the 24VDC common to the frame ground, which is the screw terminal marked "G" on the base.

By using these methods, the input stability is rated at  $\pm 1$  count.

Unused inputs should be shorted together and connected to common.

**Wiring Diagram**

The F2-08AD-2 module has a removable connector to make wiring easier. Simply squeeze the top and bottom retaining clips and gently pull the connector from the module. Use the following diagram to connect the field wiring.



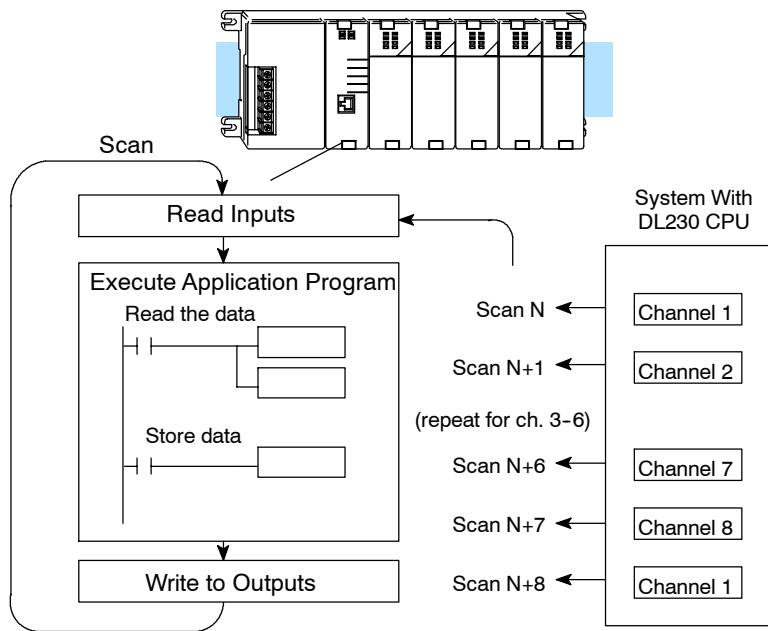
Note 1: Connect unused channels (CH5+, CH6+, CH7+, CH8+ in this diagram) to common (0 VDC).

# Module Operation

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

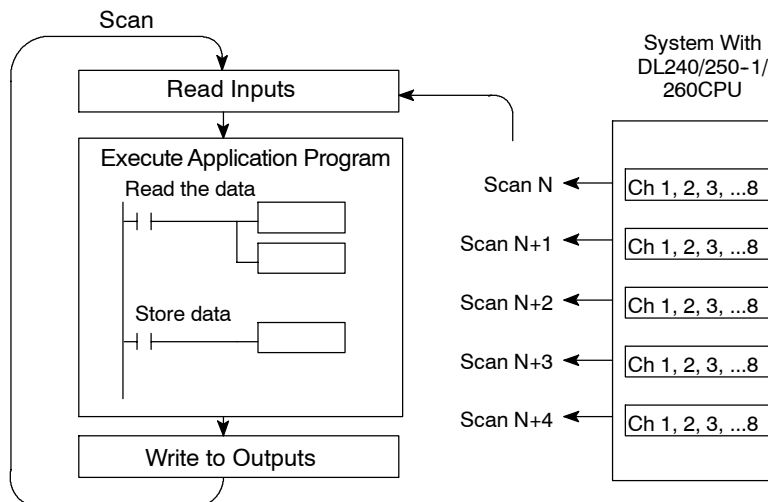
## Channel Scanning Sequence for a DL230 CPU (Multiplexing)

The F2-08AD-2 module can supply different amounts of data per scan, depending on the type of CPU you are using. The DL230 can obtain one channel of data per CPU scan. Since there are eight channels, it can take up to eight scans to get data for all channels. Once all channels have been scanned the process starts over with channel 1. Unused channels are not processed, so if you select only two channels, then each channel will be updated every other scan. The multiplexing method can also be used for DL240/250-1/260 CPUs.



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

If you are using a DL240, DL250-1 or DL260 CPU, you can obtain all eight channels of input data in one scan. This is because those CPUs support 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.)



F2-08AD-2  
8-Ch. Voltage Input



**Analog Module Updates**

Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 12-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete control logic in the RLL program.

For the vast majority of applications, the values are updated much faster than the signal changes. However, in some applications the update time can be important. The module takes approximately 4ms to sense 95% of the change in the analog signal.

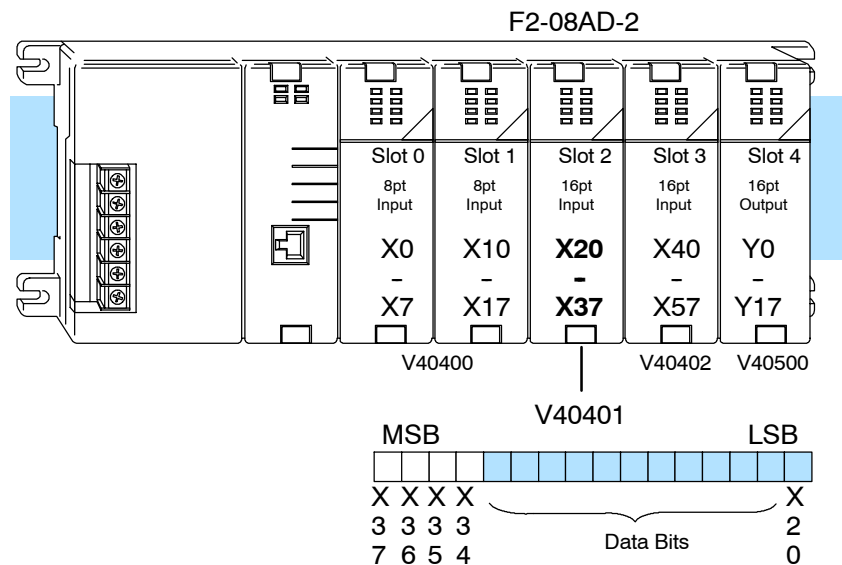
Note, this is *not* the amount of time required to convert the signal to a digital representation. The conversion to the digital representation takes only a few microseconds. Many manufacturers list the conversion time, but it is the settling time of the filter that really determines the update time.

**Understanding the Input Assignments**

You may recall the F2-08AD-2 module requires 16 discrete input points in the CPU. You can use these points to obtain:

- an indication of which channel is active.
- the digital representation of the analog signal.
- module diagnostic information.

Since all input 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.

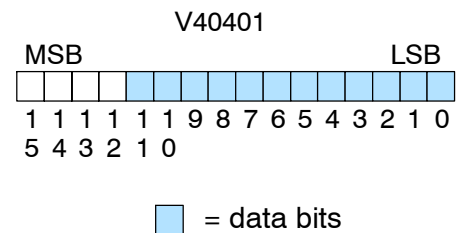


Within these word locations, the individual bits represent specific information about the analog signal.

**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



■ = data bits



# Writing the Control Program

## Reading 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.

### Pointer Method

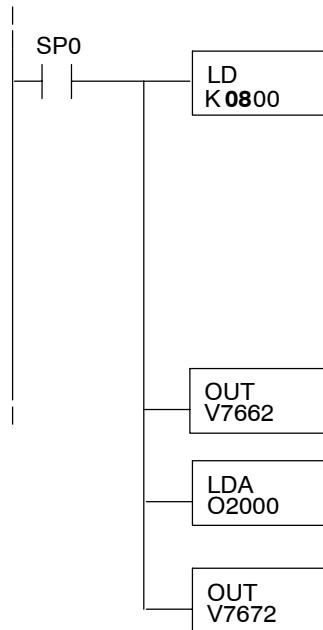


The DL240, DL250-1 and DL260 CPUs have special V-memory locations assigned to each base slot that greatly simplify the programming requirements. These V-memory locations:

- specify the data format.
- specify the number of channels to scan.
- specify the storage locations.

**NOTE:** DL240 CPUs with firmware release 2.2 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 example program 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 instructions. This is all that is required to read the data into V-memory locations. Once the data is in V-memory, you can perform math on the data, compare the data against preset values, and so forth. V2000 is used in the example, but you can use any user V-memory location. In this example the module is installed in slot 2. You should use the V-memory locations for your module placement. The pointer method automatically converts values to BCD.



- or - LD K8800

Loads a constant that specifies the number of channels to scan and the data format. The upper byte, most significant nibble (MSN) selects the data format (i.e. 0=BCD, 8=Binary), the LSN selects the number of channels (i.e. 1, 2, 3, 4, 5, 6, 7, or 8).

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 2 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 incoming data. For example, the O2000 entered here would designate the following addresses.

Ch1 - V2000, Ch2 - V2001, Ch3 - V2002, Ch4 - V2003  
Ch5 - V2004, Ch6 - V2005, Ch7 - V2006, Ch8 - V2007

The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming 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.

<b>CPU Base: Analog Input Module Slot-Dependent V-memory Locations</b>								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V7660	V7661	V7662	V7663	V7664	V7665	V7666	V7667
Storage Pointer	V7670	V7671	V7672	V7673	V7674	V7675	V7676	V7677

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

<b>Expansion Base D2-CM #1: Analog Input Module Slot-Dependent V-memory Locations</b>								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36000	V36001	V36002	V36003	V36004	V36005	V36006	V36007
Storage Pointer	V36010	V36011	V36012	V36013	V36014	V36015	V36016	V36017

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

<b>Expansion Base D2-CM #2: Analog Input Module Slot-Dependent V-memory Locations</b>								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36100	V36101	V36102	V36103	V36104	V36105	V36106	V36107
Storage Pointer	V36110	V36111	V36112	V36113	V36114	V36115	V36116	V36117

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

<b>Expansion Base D2-CM #3: Analog Input Module Slot-Dependent V-memory Locations</b>								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36200	V36201	V36202	V36203	V36204	V36205	V36206	V36207
Storage Pointer	V36210	V36211	V36212	V36213	V36214	V36215	V36216	V36217

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

<b>Expansion Base D2-CM #4: Analog Input Module Slot-Dependent V-memory Locations</b>								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36300	V36301	V36302	V36303	V36304	V36305	V36306	V36307
Storage Pointer	V36310	V36311	V36312	V36313	V36314	V36315	V36316	V36317

**Using Bipolar Ranges (Pointer Method)**

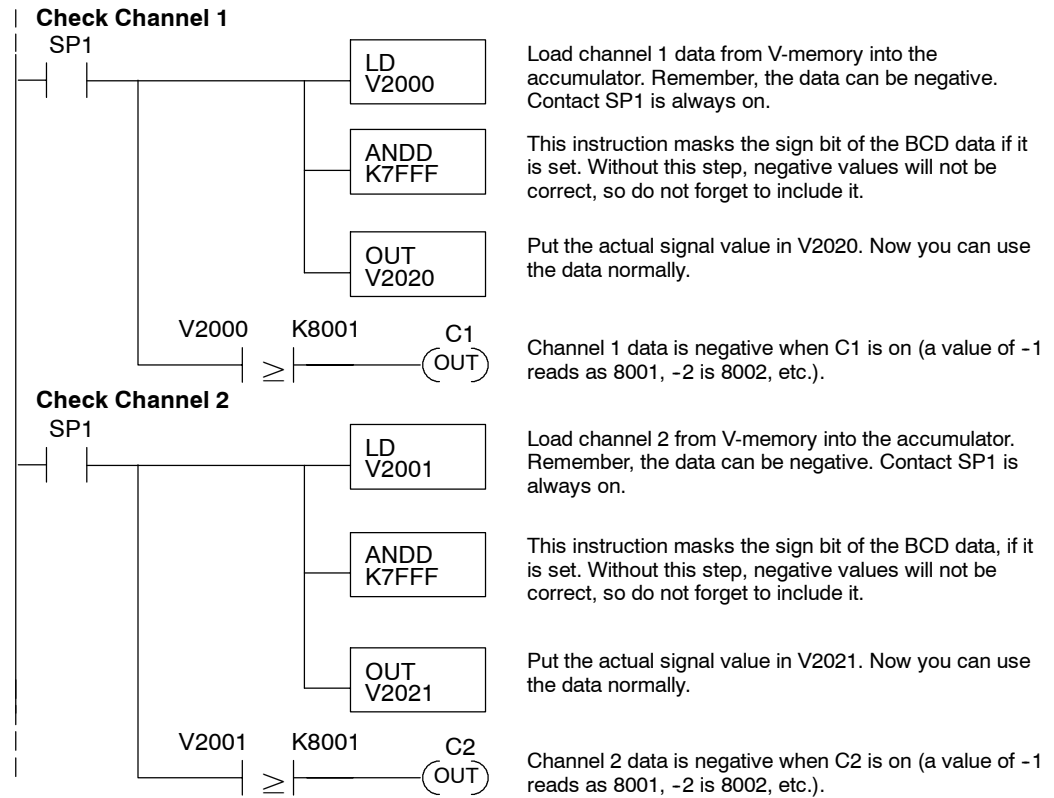
✗	✓	✓	✓
230	240	250-1	260

With bipolar ranges, you need some additional logic to determine whether the value being returned represents a positive or a negative voltage. For example, you may need to know the direction for a motor. With the pointer method, you *cannot* use the last input (X37 in the previous examples) to show the sign for each channel because the DL240/250-1/260 read all eight channels in one scan. If you tried to use X37, you would only be monitoring the last channel that was read. You would not be able to determine the sign for the previous channels. There is a simple solution:

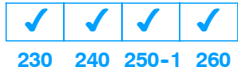
- If you get a value greater than or equal to 8001, the value is negative.

The sign bit is the most significant bit, which combines 8000 to the data value. If the value is greater than or equal to 8001, you only have to mask the most significant bit and the active channel bits to determine the actual data value.

The following program shows how you can accomplish this. Since you always want to know when a value is negative, these rungs should be placed *before* any other operations that use the data, such as math instructions, scaling operations, and so forth. Also, if you are using stage programming instructions, these rungs should be in a stage that is always active. Note, you only need this logic for each channel that is using bipolar input signals. The example only shows two channels.

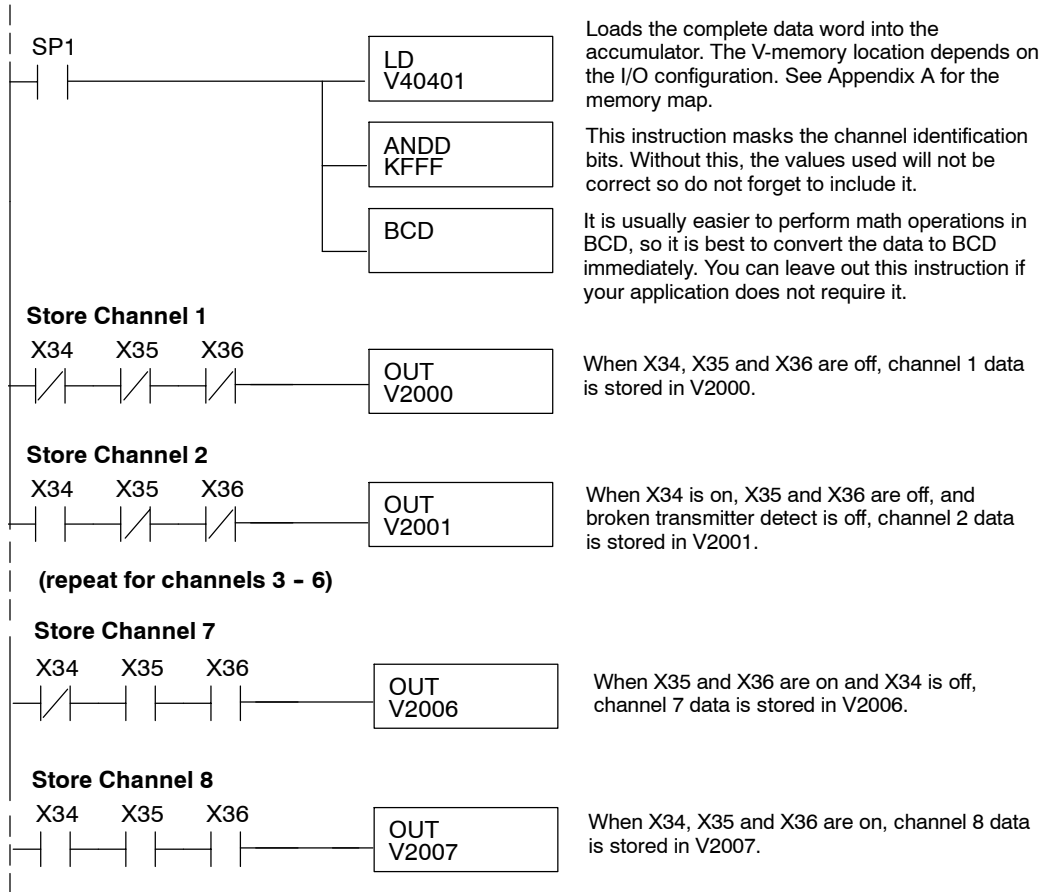


### Reading Values (Multiplexing)



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 setup to determine which channel is being read. Since the module appears as X input points to the CPU, it is very easy to use the active channel status bits to determine which channel is being monitored.

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 slot. You can place these rungs anywhere in the program, or if you are using stage programming instructions place them in a stage that is always active.



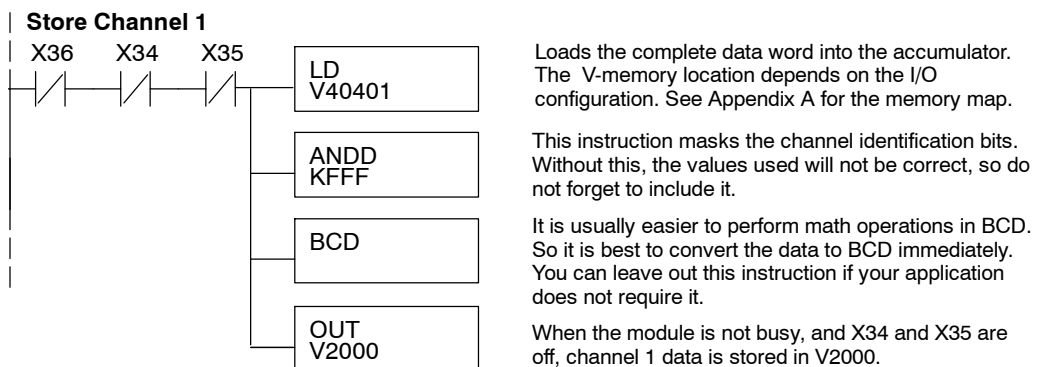
Loads the complete data word into the accumulator. The V-memory location depends on the I/O configuration. See Appendix A for the memory map.

This instruction masks the channel identification bits. Without this, the values used will not be correct so do not forget to include it.

It is usually easier to perform math operations in BCD, so it is best to convert the data to BCD immediately. You can leave out this instruction if your application does not require it.

### Single Channel Selected

Since you do not have to determine which channel is selected, the single channel program is even simpler.



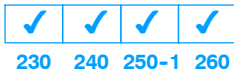
Loads the complete data word into the accumulator. The V-memory location depends on the I/O configuration. See Appendix A for the memory map.

This instruction masks the channel identification bits. Without this, the values used will not be correct, so do not forget to include it.

It is usually easier to perform math operations in BCD. So it is best to convert the data to BCD immediately. You can leave out this instruction if your application does not require it.

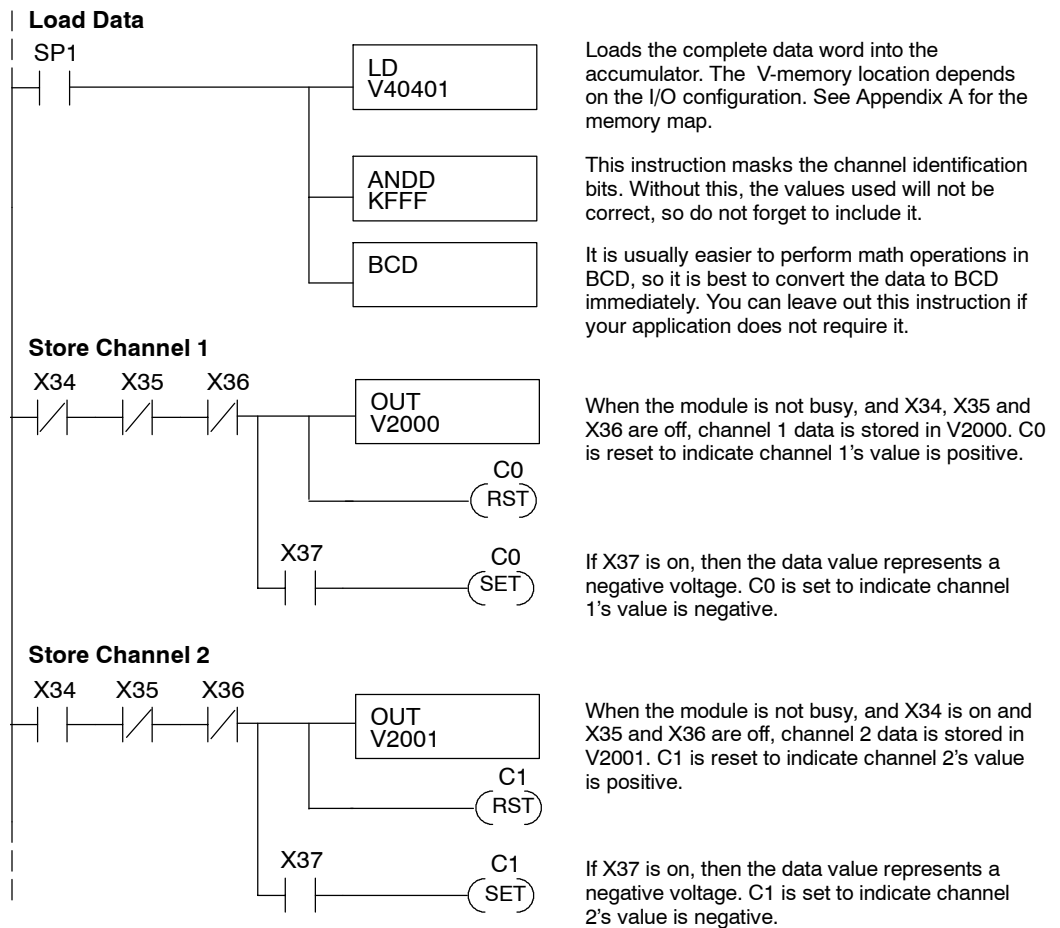
When the module is not busy, and X34 and X35 are off, channel 1 data is stored in V2000.

**Using Bipolar Ranges (Multiplexing)**

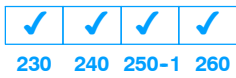


With bipolar ranges, you need some additional logic because you need to know if the value being returned represents a positive voltage or a negative voltage. For example, you may need to know the direction for a motor. Since the DL230 only reads one channel per scan, you can use the last input (X37 in the examples) to show the sign.

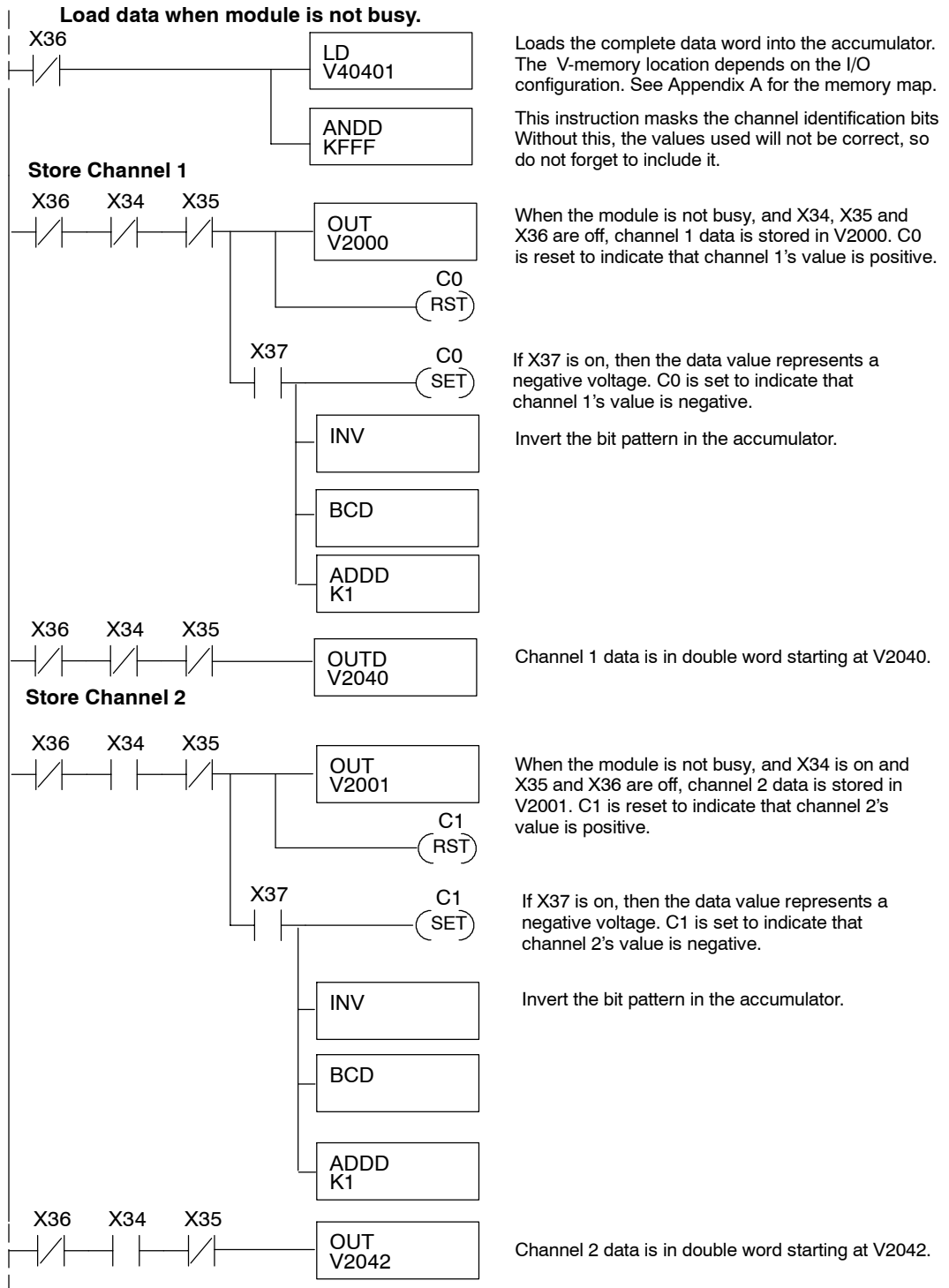
The following program shows how you can accomplish this. Since you always want to know when a value is negative, these rungs should be placed *before* any operations that use the data, such as math instructions, scaling operations, and so forth. Also, if you are using stage programming instructions, these rungs should be in a stage that is always active. Note, you only need the additional logic for those channels that are using bipolar input signals. The example shows two channels, but you can repeat these steps for all eight channels if necessary.



### Using 2's Complement (Multiplexing)



The 2's complement data format may be required to display negative values on some operator interface devices. It could also be used to simplify data averaging on bipolar signals. The example shows two channels, but you can repeat these steps for all eight channels if necessary.

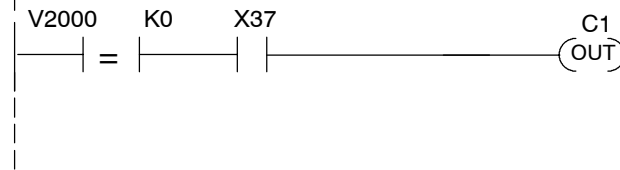




**Analog Power Failure Detection**

The analog module has an on-board RISC-like microcontroller that can diagnose analog input circuit problems. You can easily create a simple ladder rung to detect these problems. This rung shows an input point that would be assigned if the module was used as shown in the previous examples. A different point would be used if the module was installed in a different I/O arrangement.

**Multiplexing method**



V-memory location V2000 holds channel 1 data. When a data value of zero is returned and input X37 is on, the analog channel is not operating properly.

**Pointer method**



V-memory location V2000 holds channel 1 data. When a data value of 8000 is returned, the analog channel is not operating properly.

**Scaling the Input Data**

Most applications usually require measurements in engineering units, which provide more meaningful data. 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.

$$\text{Units} = A \frac{H - L}{4095}$$

H = high limit of the engineering unit range

L = low limit of the engineering unit range

A = Analog value (0 - 4095)

For example, if you wanted to measure pressure (PSI) from 0.0 to 99.9 you would have to multiply the analog value by 10 in order to imply a decimal place when you view the value with the programming software or a handheld programmer. Notice how the calculations differ when you use the multiplier.

Analog Value of 2024, slightly less than half scale, should yield 49.4 PSI

Example without multiplier

$$\text{Units} = A \frac{H - L}{4095}$$

$$\text{Units} = 2024 \frac{100 - 0}{4095}$$

$$\text{Units} = 49$$

Handheld Display

V 2001	V 2000
0000	0049

Example with multiplier

$$\text{Units} = 10 A \frac{H - L}{4095}$$

$$\text{Units} = 20240 \frac{100 - 0}{4095}$$

$$\text{Units} = 494$$

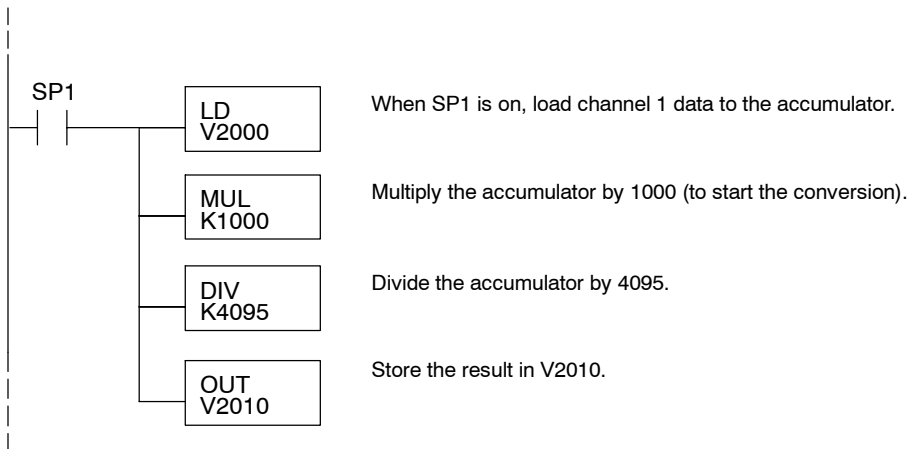
Handheld Display

V 2001	V 2000
0000	0494

This value is more accurate.

The example below shows how you would write the program to perform the engineering unit conversion. This example assumes you have BCD data loaded into the appropriate V-memory locations using instructions that apply for the model of CPU you are using.

Note, this example uses SP1, which is always on. You could also use an X, C, etc. permissive contact.



### 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. Remember, this module *does not* operate like other versions of analog input modules that you may be familiar with. The bipolar ranges use 0–4095 for both positive and negative voltages. The sign bit allows this, which actually provides better resolution than those modules that do not offer a sign bit. The following table provides formulas to make this conversion easier.

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

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

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

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

$$D = (409.5) (6)$$

$$D = 2457$$

**Filtering Input Noise (DL250-1, DL260 CPUs Only)**



Add the following logic to filter and smooth analog input noise in DL250-1 and DL260 CPUs. This is especially useful when using PID loops. Noise can be generated by the field device and/or induced by field wiring.

The analog value in BCD is first converted to a binary number because there is not a BCD-to-real conversion instruction. Memory location V1400 is the designated workspace in this example. The MULR instruction is the filter factor, which can be from 0.1 to 0.9. The example uses 0.2. A smaller filter factor increases filtering. You can use a higher precision value, but it is not generally needed. The filtered value is then converted back to binary and then to BCD. The filtered value is stored in location V1402 for use in your application or PID loop.

**NOTE:** Be careful not to do a multiple number conversion on a value. For example, if you are using the pointer method to get the analog value, it is in BCD and must be converted to binary. However, if you are using the conventional method of reading analog and are masking the first twelve bits, then it is already in binary and no conversion using the BIN instruction is needed.

