

F2-4AD2DA

4-Ch. In / 2-Ch. Out Combination Analog

14

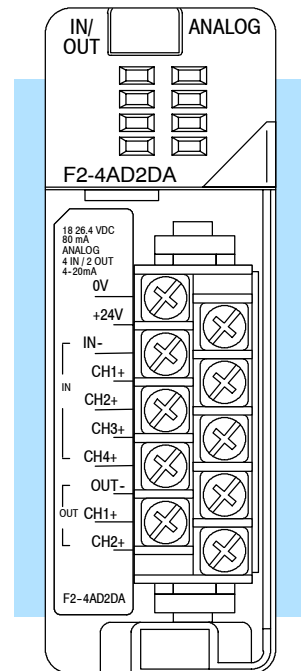
In This Chapter. . . .

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

Module Specifications

The F2-4AD2DA Analog Input/Output module provides several hardware features:

- On-board 250 ohm, 1/2 watt precision resistors provide substantial over-current-protection for 4–20mA current loops.
- Analog inputs and outputs are optically isolated from the PLC logic.
- The module has a removable terminal block so the module can be easily removed or changed without disconnecting the wiring.
- With a DL240/250–1/260 CPU, you can update all input and output channels in one scan.
- On-board active analog filtering and RISC-like microcontroller provide digital signal processing to maintain precision analog measurements in noisy environments.
- Low-power CMOS design requires less than 80mA from an external 18–26.4 VDC power supply.



The following tables provide the specifications for the F2-4AD2DA Analog Input/Output Module. Review these specifications to make sure the module meets your application requirements.

Input Specifications

Number of Input Channels	4, single ended (one common)
Range	4 to 20 mA current
Resolution	12 bit (1 in 4096)
Input Impedance	250Ω, ±0.1%, 1/2W, 25ppm/°C current input resistance
Maximum Continuous Overload	-40 to +40 mA, each current input
Input Stability	± 1 count
Crosstalk	-70 dB, 1 count maximum
Common Mode Rejection	-50 dB at 800 Hz
Active Low-Pass Filter	-3 dB at 50Hz, 2 poles (-12 dB per octave)
Step Response	10 mS to 95%
Full Scale Calibration Error	± 12 counts maximum, at 20 mA current input
Offset Calibration Error	± 8 counts maximum, at 4 mA current input
Maximum Input Inaccuracy	± 0.3% @ 25°C (77°F) ± 0.45% @ 0 to 60°C (32 to 140°F)
Recommended External Fuse	0.032A, series 217 fast-acting, current inputs

Output Specifications

Number of Output Channels	2, single ended (one common)
Range	4 to 20 mA current
Resolution	12 bit (1 in 4096)
Peak Withstanding Voltage	75 VDC, current outputs
External Load Resistance	0Ω minimum, current outputs
Loop Supply Voltage Range	18-30VDC, current outputs
Maximum Load / Power Supply	910Ω/24V, 620Ω/18V, 1200Ω/30V, current outputs
Linearity Error (best fit)	± 1 count (± 0.025% of full scale) maximum
Settling Time	100 μs maximum (full scale change)
Maximum Inaccuracy	± 0.1% @ 25°C (77°F) ± 0.3% @ 0 to 60°C (32 to 140°F)
Full Scale Calibration Error	± 5 counts at 20 mA current output
Offset Calibration Error	± 3 counts at 4 mA current output

General Module Specifications

Digital Input and Output Points Required	16 point (X) inputs 16 point (Y) outputs
PLC Update Rate	4 input channels per scan maximum (D2-240/250-1/260 CPU) 2 output channels per scan maximum (D2-240/250-1/260 CPU) 1 input and 1 output channel per scan maximum (D2-230 CPU)
Power Budget Requirement	60 mA @ 5 VDC (supplied by base)
External Power Supply Requirement	18 to 26.4 VDC, 80 mA maximum plus 20 mA per loop output
Accuracy vs. Temperature	± 45 ppm/°C full scale calibration range (including maximum offset change)
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).

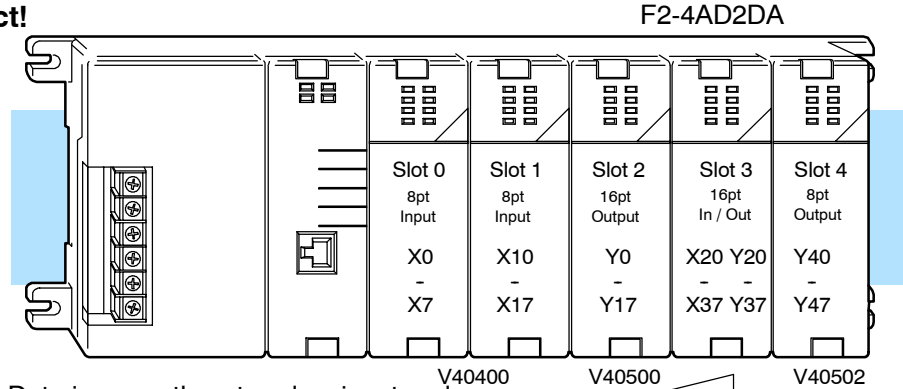
Combination Analog Configuration Requirements

The F2-4AD2DA Analog module requires 16 discrete input and 16 discrete output points. The module can be installed in any slot of a DL205 system, except when you use the DL230 programming method. The available power budget may also be a limiting factor. 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.

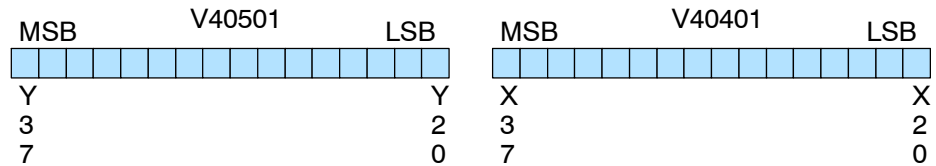
Special Placement Requirements (DL230 and Remote I/O Bases)

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 can see in the section on writing the program, you use V-memory locations to extract the analog data. If you place the module so that either the input or output points do not start on a V-memory boundary, the instructions cannot access the data. This also applies when the module is placed in remote I/O bases (D2-RSSS in CPU slot).

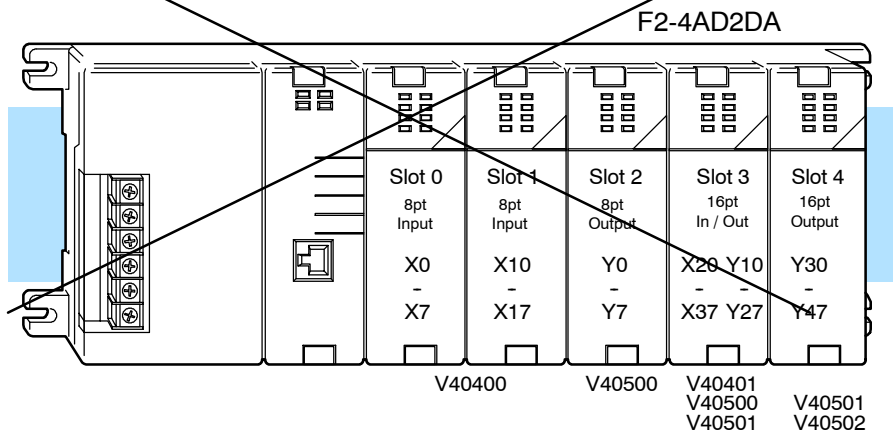
Correct!



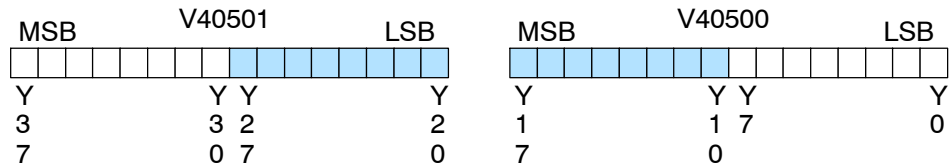
Output Data is correctly entered so input and output points start on a V-memory boundary.



Incorrect



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



F2-4AD2DA
4-Ch. In / 2-Ch. Out

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

X	X0	X20	X40	X60	X100	X120	X140	X160
V	V40400	V40401	V40402	V40403	V40404	V40405	V40406	V40407
Y	Y0	Y20	Y40	Y60	Y100	Y120	Y140	Y160
V	V40500	V40501	V40502	V40503	V40504	V40505	V40506	V40507

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 ideas to consider:

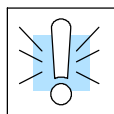
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the load or source.
- Do not run the signal wiring next to large motors, high current switches, or transformers. This may cause noise 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

The F2-4AD2DA requires at least one field-side power supply. You may use the same or separate power sources for the module supply and loop supply. The module requires 18–26.4VDC, at 80 mA. In addition, each current loop requires 20 mA (a total of 120 mA for six current loops). If you want to use a separate power supply make sure that it meets these requirements.

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 one combination module. The current required is 80 mA (module) plus up to 120 mA (six current loops) for a total of 200 mA.

It is desirable in some situations to power the loops separately in a location remote from the PLC. This will work as long as the loop's power supply meets the voltage and current requirements, and its 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, you should 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 ± 1 count.

Current Loop Transmitter Impedance

Standard 4 to 20 mA transmitters and transducers can operate from a wide variety of power supplies. Not all transmitters are alike and the manufacturers often specify a minimum loop or load resistance that must be used with the transmitter.

The F2-4AD2DA provides 250 ohm resistance for each input channel. If your transmitter requires a load resistance below 250 ohms, you do not have to make any adjustments. However, if your transmitter requires a load resistance higher than 250 ohms, you need to add a resistor in series with the module.

Consider the following example for a transmitter being operated from a 30 VDC supply with a recommended load resistance of 750 ohms. Since the module has a 250 ohm resistor, you need to add an additional resistor.

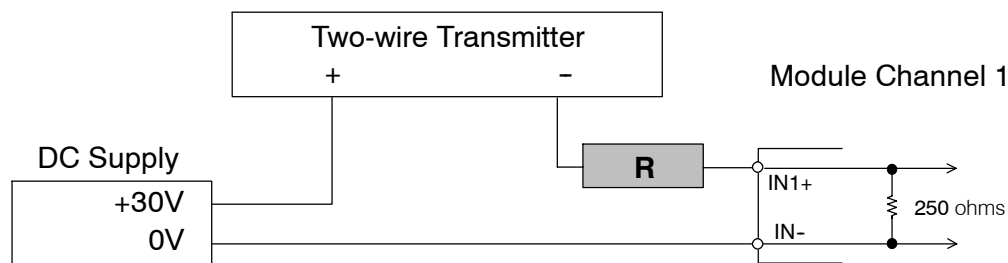
Example:

$$R = Tr - Mr$$

$$R = 750 - 250$$

$$R \geq 500$$

R - resistor to add
Tr - Transmitter total resistance requirement
Mr - Module resistance (internal 250 ohms)



In the example, add a 500 ohm resistor (R) in series with the module.

Wiring Diagram The F2-04AD2DA 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. The diagram shows separate module and loop power supplies. If you desire to use only one field-side supply, just combine the supplies' positive (+) terminals into one node, and remove the loop supply.

Note 1: Shields should be connected at their respective signal source.

Note 2: Unused channels should remain open (no connections) for minimum power consumption.

Note 3: More than one external power supply can be used provided all the power supply commons are connected together.

Note 4: A series 217, 0.032A, fast-acting fuse is recommended for 4-20 mA current input loops.

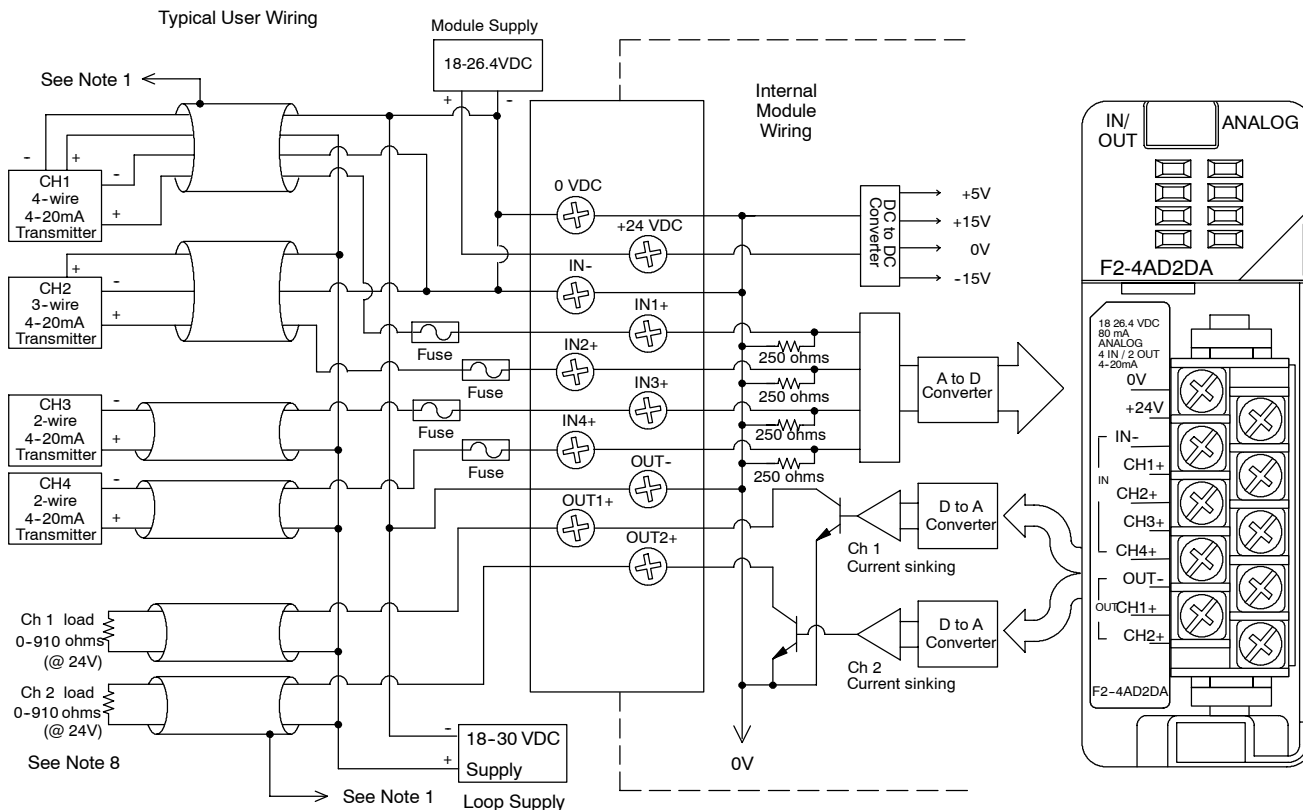
Note 5: If the power supply common of an external power supply is not connected to 0V on the module, then the output of the external transmitter must be isolated. To avoid "ground loop" errors, recommended 4-20mA transmitter types are:

- a. For 2 or 3 wire: Isolation between input signal and power supply.
- b. For 4 wire: Isolation between input signal, power supply, and 4-20 mA output.

Note 6: If an analog channel is connected backwards, then incorrect data values will be returned for that channel. Input signals in the -4 to +4 mA range return a zero value. Signals in the -4 to -40 mA range return a non-zero value.

Note 7: To avoid small errors due to terminal block losses, connect 0V, IN- and OUT- on the terminal block as shown. The module's internal connection of these nodes is not sufficient to permit module performance up to the accuracy specifications.

Note 8: Choose a output transducer resistance according to the maximum load / power supply listed in the Output Specifications table.



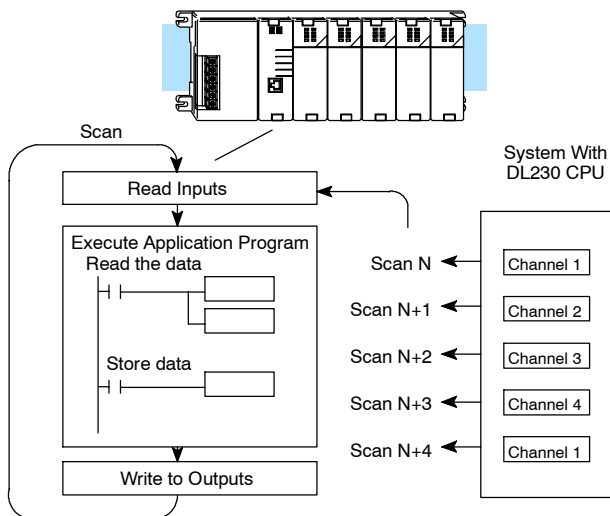
F2-4AD2DA
4-Ch. In / 2-Ch. Out

Module Operation

Input Channel Scanning Sequence for a DL230 CPU (Multiplexing)

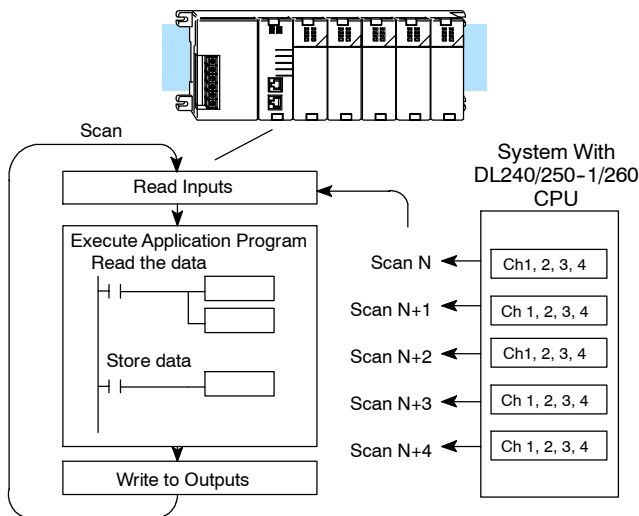
Before you begin writing the control program, it is important to take a few minutes to understand how the module processes and represents the analog signals.

The F2-4AD2DA 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 input data per CPU scan. Since there are four channels, it can take up to four 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.



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

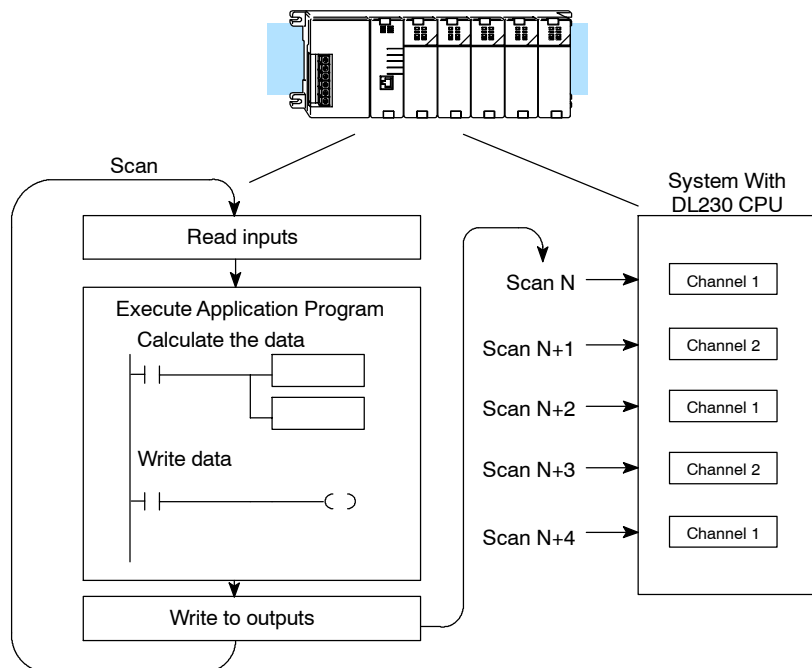
If you are using a DL240, DL250-1 or DL260 CPU, you can obtain all four channels of input data in one scan. This is because the DL240/250-1/260 CPU supports special V-memory locations that are used to manage the data transfer. This is discussed in more detail in the section on Writing the Control Program.



F2-4AD2DA
4-Ch. In / 2-Ch. Out

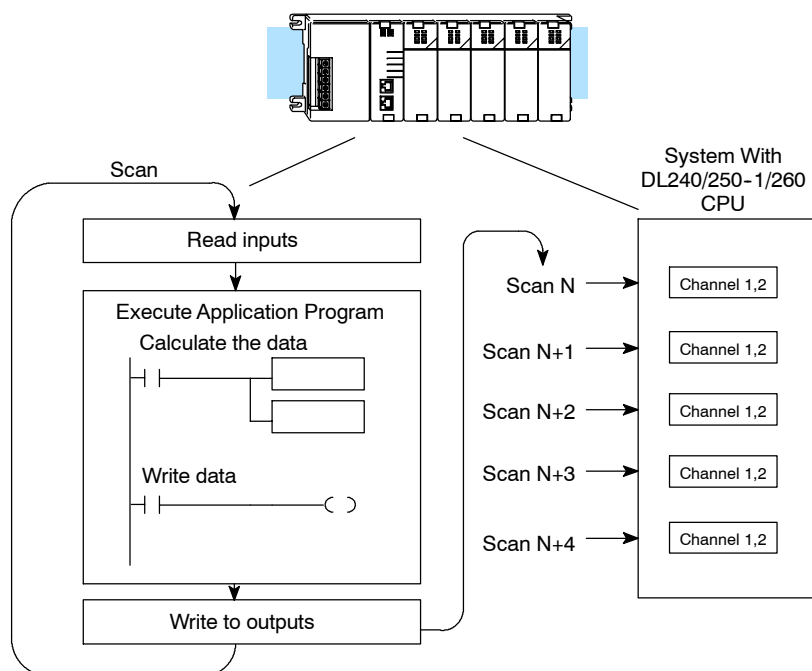
Output Channel Update Sequence for a DL230 CPU (Multiplexing)

If you are using a DL230 CPU, you can send one channel of data to the output module on each scan. Since there are two channels, it can take two scans to update both channels. However, if you are only using one channel, then you can update that channel on every scan.



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

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

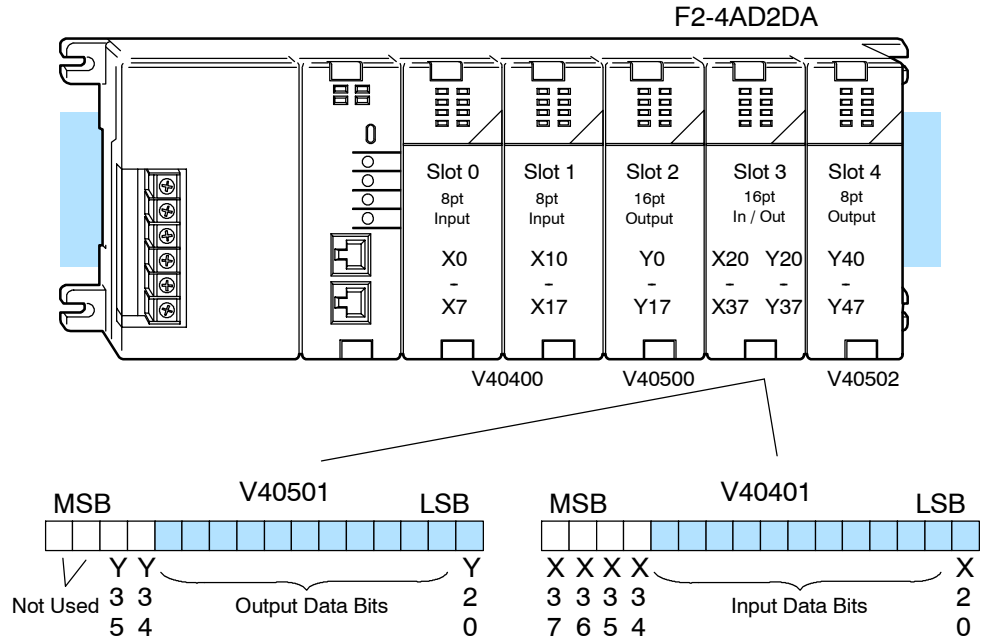


F2-4AD2DA
4-Ch. In / 2-Ch. Out

Understanding the I/O Assignments

You may recall the F2-4AD2DA module appears to the CPU as 16 discrete input and 16 discrete output points. These points provide the data value and channel identification. Note, if you are using a DL240/250 CPU, you may never have to use these bits, but it may help you understand the data format.

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

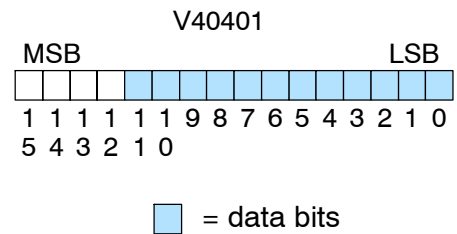


Within this word location, the individual bits represent specific information about the analog signal.

Input Data Bits

The first twelve bits of the input 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

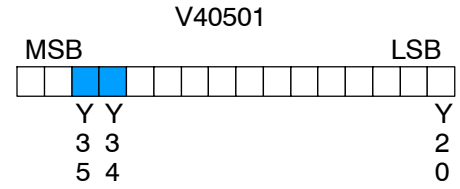


F2-4AD2DA
4-Ch. In / 2-Ch. Out

Output Channel Selection Bits

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

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

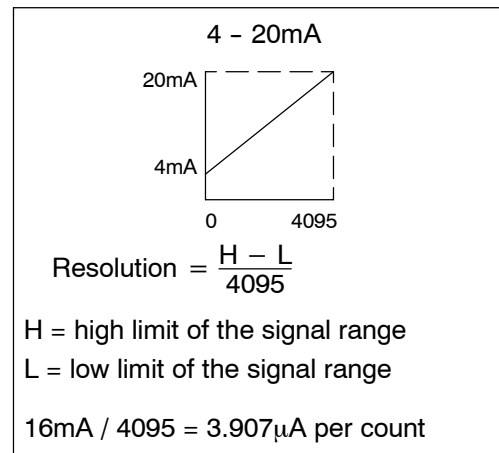


■ = channel control outputs

Module Resolution

Since the module has 12-bit resolution, the analog signal is converted into 4096 counts ranging from 0 - 4095 (2^{12}). For example, a 4mA signal would be 0, and a 20mA signal would be 4095. This is equivalent to a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF hexadecimal. The diagram shows how this relates to the signal range.

Each count can also be expressed in terms of the signal level by using the equation shown.



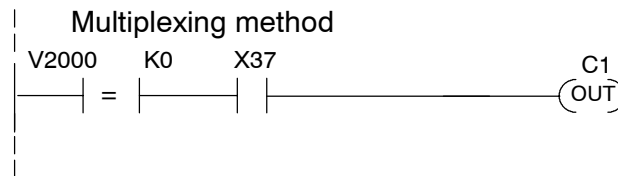
Writing the Control Program

Before you begin writing the program, there are a few supplemental examples that can be extremely beneficial. They include:

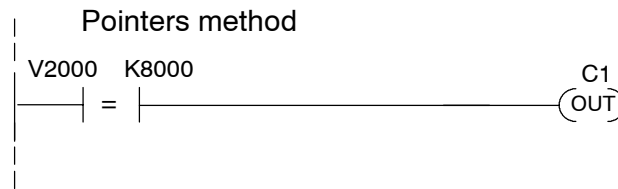
- Input power failure detection
- Output data calculation
- Input data scaling

Analog Input Power Failure Detection

Take a close look at these examples. They may be helpful for your application. The analog module has a 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 and following examples.



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



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

Calculating the Output Data

Your program has to calculate the digital value to send to the analog output channels. There are many ways to do this, but most applications are understood more easily if you use measurements in engineering units. This is accomplished by using the conversion formula shown.

You may have to make adjustments to the formula depending on the scale you choose for the engineering units.

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

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.

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

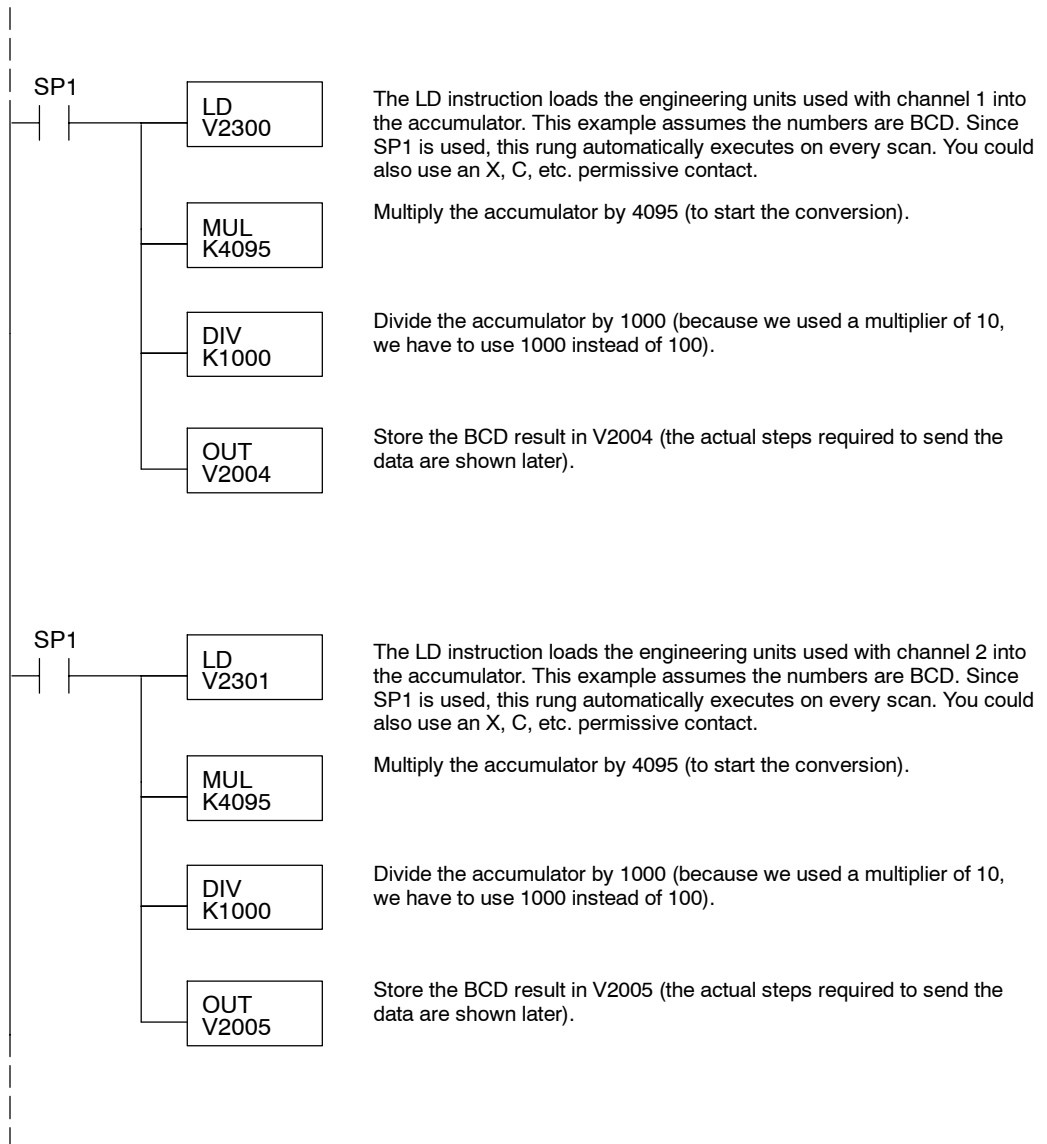
$$A = 494 \frac{4095}{1000 - 0}$$

$$A = 2023$$

The example program below shows how you would write the program to perform the engineering unit conversion. This example will work with all CPUs and assumes that you have calculated or loaded the engineering unit values and stored them in V2300 and V2301 for channels 1 and 2 respectively. Also, we move the final values to V2004 and V2005, which are memory locations that are used in the following examples. You can use any user V locations, but they must match the locations that are specified as the source for the output data (see the next section for an example).



NOTE: The DL205 offers instructions that allow you to perform math operations using BCD format. It is usually easier to perform any math calculations in BCD.



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

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

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

$$\text{Units} = 494$$

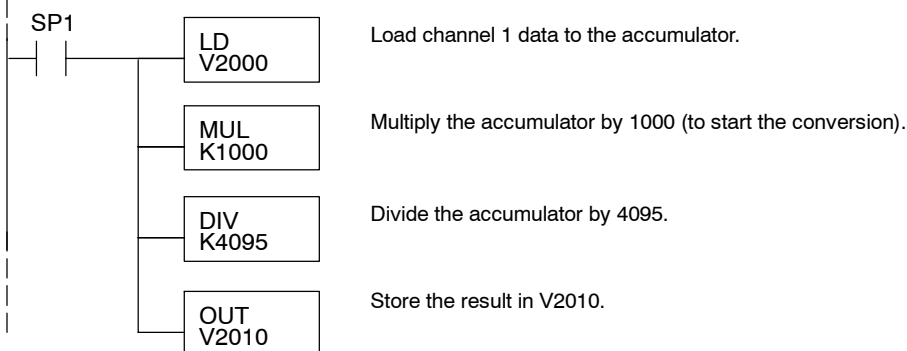
Handheld Display

V 2001	V 2000
0000	0494

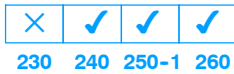
This value is more accurate

The example program 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.



Read / Write Program (Pointer Method)



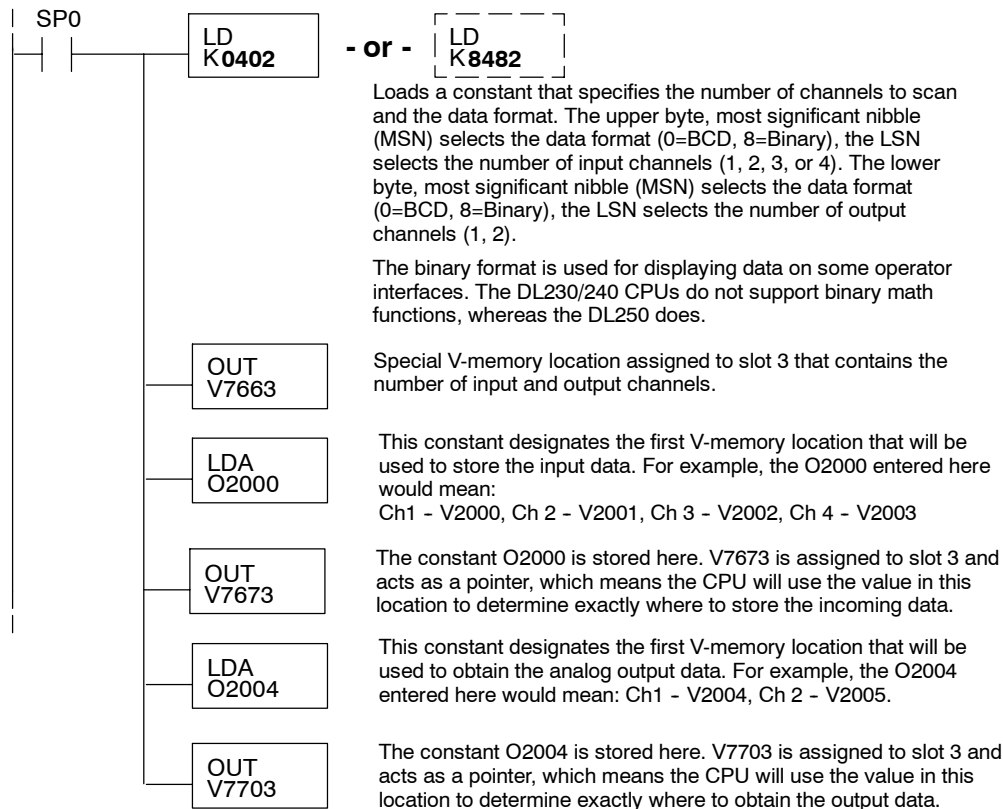
The DL240, DL250-1 and DL260 CPUs have special V-memory locations assigned to each base slot that greatly simplify the programming requirements. These V-memory locations:

- specify the number of input and output channels to scan.
- specify the storage location for the input data.
- specify the source location for the output data.

NOTE: To use the pointer method, DL250 CPUs must have firmware revision 1.09 or later, and F2-AD2DA modules must be revision C1 or later.

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 input data into V-memory locations. The CPU automatically converts the binary input data to BCD format. Once the input data is in V-memory, you can perform math on the data, compare the data against preset values, and so forth. For the output data, you have to calculate the digital value in BCD (as shown previously) before you send the data to the module, unless you select the binary data format option shown below.

V2000 and V2004 are used as the beginning of the data areas in the example, but you can use any user V-memory locations. Also, in the previous examples the module was installed in slot 3. You should use the V-memory locations for your application. The pointer method automatically converts values to BCD.



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

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

CPU Base: Analog In/Out Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V7660	V7661	V7662	V7663	V7664	V7665	V7666	V7667
Input Pointer	V7670	V7671	V7672	V7673	V7674	V7675	V7676	V7677
Output Pointer	V7700	V7701	V7702	V7703	V7704	V7705	V7706	V7707

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

Expansion Base D2-CM #1: Analog In/Out Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36000	V36001	V36002	V36003	V36004	V36005	V36006	V36007
Input Pointer	V36010	V36011	V36012	V36013	V36014	V36015	V36016	V36017
Output Pointer	V36020	V36021	V36022	V36023	V36024	V36025	V36026	V36027

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

Expansion Base D2-CM #2: Analog In/Out Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36100	V36101	V36102	V36103	V36104	V36105	V36106	V36107
Input Pointer	V36110	V36111	V36112	V36113	V36114	V36115	V36116	V36117
Output Pointer	V36120	V36121	V36122	V36123	V36124	V36125	V36126	V36127

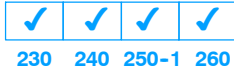
The Table below applies to the DL260 CPU expansion base 3.

Expansion Base D2-CM #3: Analog In/Out Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36200	V36201	V36202	V36203	V36204	V36205	V36206	V36207
Input Pointer	V36210	V36211	V36212	V36213	V36214	V36215	V36216	V36217
Output Pointer	V36220	V36221	V36222	V36223	V36224	V36225	V36226	V36227

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

Expansion Base D2-CM #4: Analog In/Out Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V36300	V36301	V36302	V36303	V36304	V36305	V36306	V36307
Input Pointer	V36310	V36311	V36312	V36313	V36314	V36315	V36316	V36317
Output Pointer	V36320	V36321	V36322	V36323	V36324	V36325	V36326	V36327

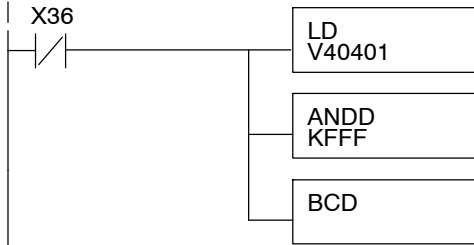
Reading Input Values (Multiplexing)



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 installed in another I/O arrangement. You can place these rungs anywhere in the program or if you are using stage programming, place them in a stage that is always active.

Load data when module is not busy.

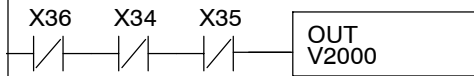


Loads the complete data word into the accumulator. The V-memory location depends on the I/O configuration. See Appendix A for the memory map.

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

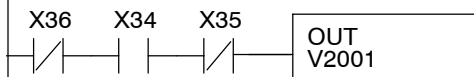
It is usually easier to perform math operations in BCD, You can leave out this instruction if your application does not require it.

Store Channel 1



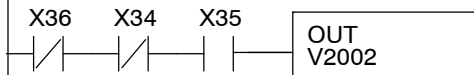
When the module is not busy and X36, X34 and X35 are off, channel 1 data is stored in V2000.

Store Channel 2



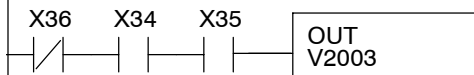
When the module is not busy and X34 is on and X35 and X36 are off, channel 2 data is stored in V2001.

Store Channel 3



When the module is not busy and X34 and X36 are off and X35 is on, channel 3 data is stored in V2002.

Store Channel 4

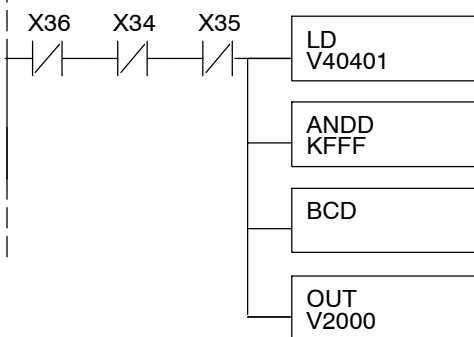


When the module is not busy and both X34 and X35 are on and X36 is off, channel 4 data is stored in V2003.

Single Input Channel Selected (Multiplexing)

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

Store channel 1 when module is not busy.



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. 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.

Writing Output 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 to write. Since the module appears as Y output points to the CPU, it is very easy to use the channel selection outputs to determine which channel to update.

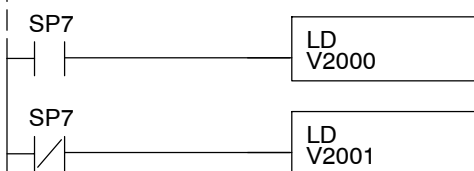
Note, this example is for a module installed as shown in the previous examples. The addresses used would be different if the module was used in a different I/O arrangement. You can place these rungs anywhere in the program or if you are using stage programming, place them in a stage that is always active.

This example is a two-channel multiplexer that updates each channel on alternate scans. SP7 is a special relay that is on for one scan, then off for one scan.



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

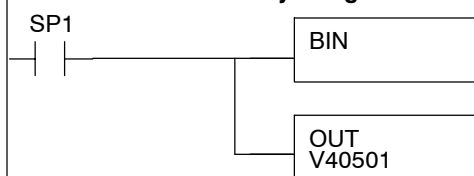
Load data into the accumulator.



Loads the data for channel 1 into the accumulator.

Loads the data for channel 2 into the accumulator.

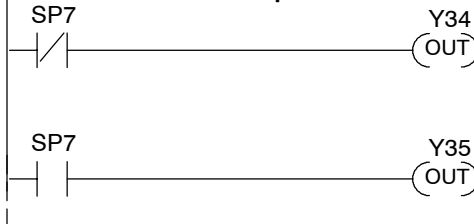
Send data to V-memory assigned to the module.



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

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

Select the channel to update.



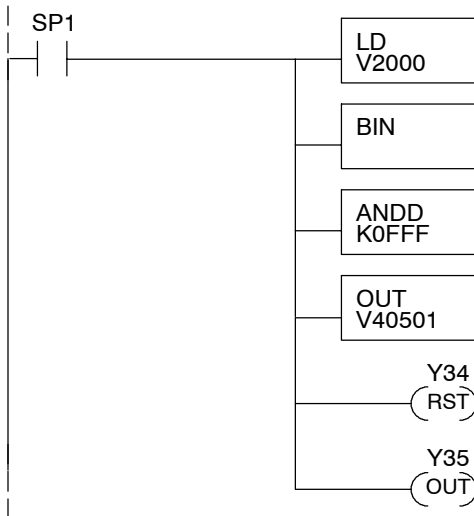
Selects channel 1 for update when Y34 is OFF (Y35-ON deselects channel 2). Note, Y34 and Y35 are used due to the previous examples. If the module was installed in a different I/O arrangement, the addresses would be different.

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

F2-4AD2DA
4-Ch. In / 2-Ch. Out

Sending Data to One Channel (Multiplexing)

If you are not using both channels, or if you want to control the updates separately, use the following program.



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

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

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

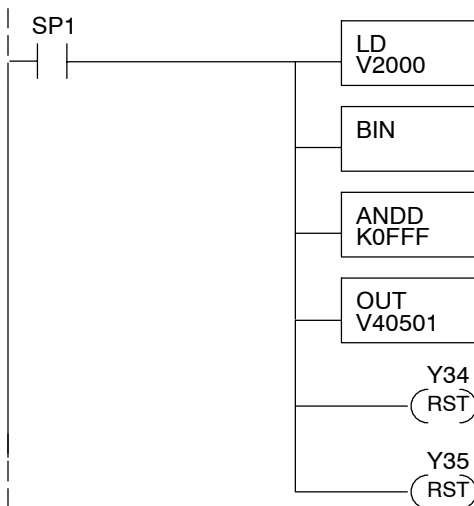
The OUT instruction sends the data to the module. Our example starts with V40501, but the actual value depends on the location of the module in your application.

Y34-OFF selects channel 1 for updating.

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

Sending the Same Data to Both Channels (Multiplexing)

If both channel selection outputs are off, both channels will be updated with the same data.



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

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

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

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

Y34-OFF selects channel 1 for updating.

Y35-OFF selects channel 2 for updating.

Analog and Digital Value Conversions

Sometimes it is useful to be able to quickly convert between the signal levels and the digital values. This is especially helpful during machine startup or troubleshooting. The 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 have measured the signal at 10mA, you could use the formula to easily determine the digital value (D) that should be stored in the V-memory location that contains the data.

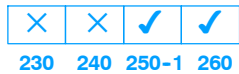
$$D = \frac{4095}{16}(A - 4)$$

$$D = \frac{4095}{16}(10\text{mA} - 4)$$

$$D = (255.93) (6)$$

$$D = 1536$$

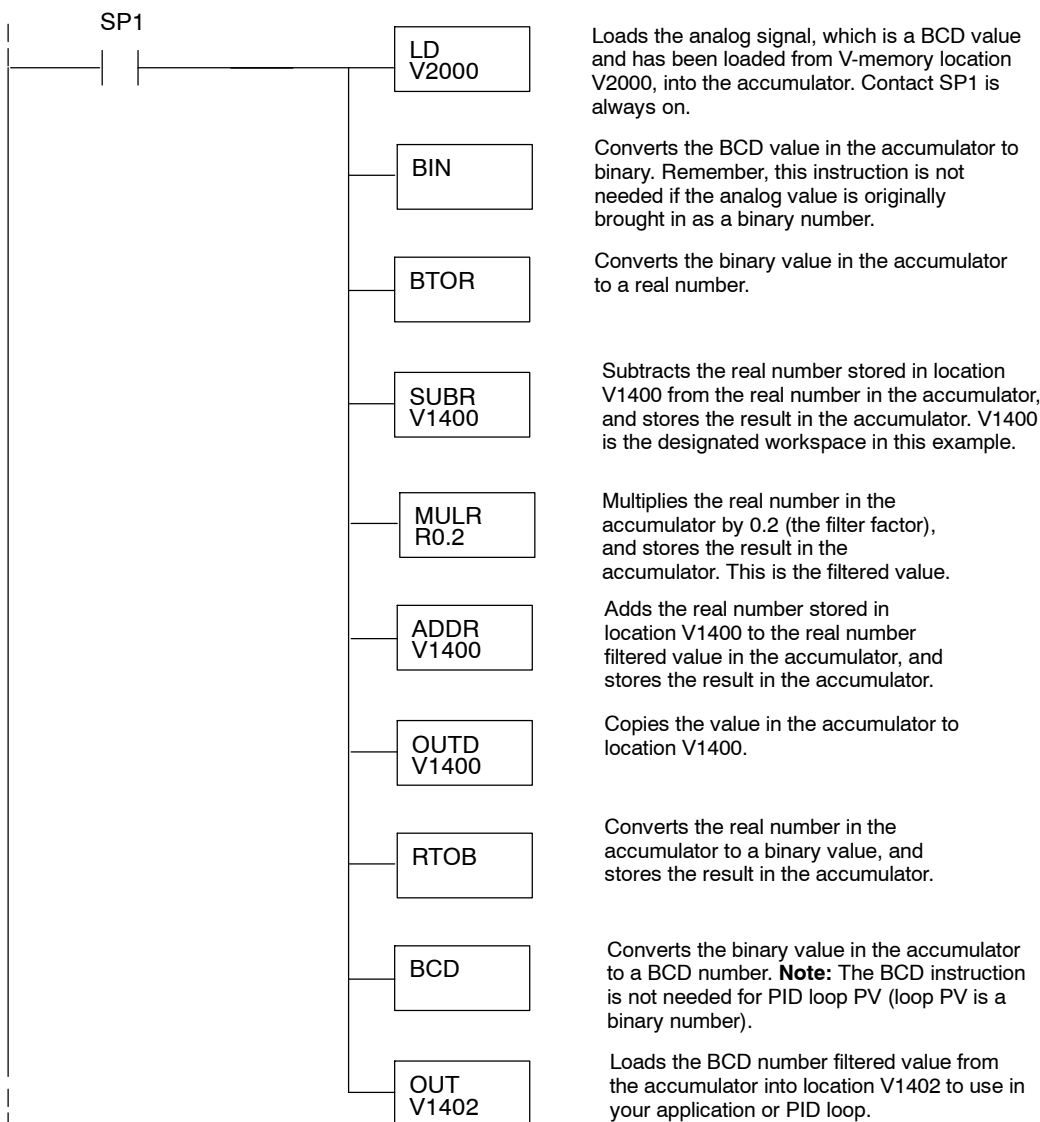
Filtering Input Noise (DL250-1, DL260 CPUs Only)



Add the following logic to filter and smooth analog input noise in DL250-1 or 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.



F2-4AD2DA
4-Ch. In / 2-Ch. Out