Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

Product Family:	DL305	**Date**:	September 2018
Manual Number	D3-ANLG-M		
Revision and Date	3rd Edition, February 2003		

Changes to Chapter 2. D3-04AD 4-Channel Analog Input

This module is no longer available. Please consider the F3-08AD-1 or F3-04ADS as a replacement

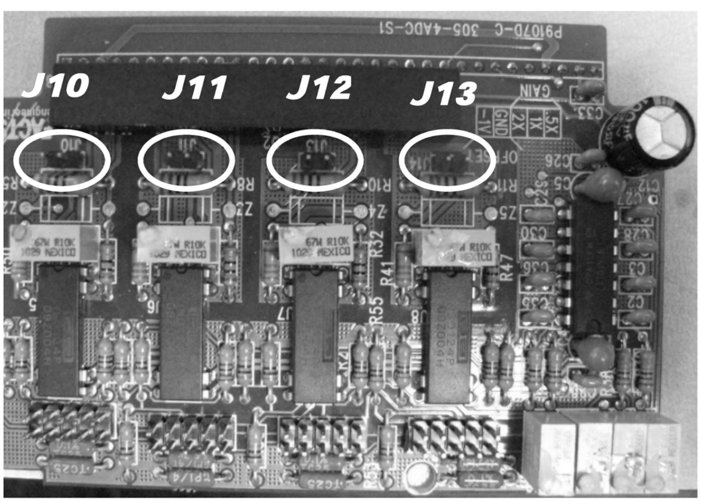
Changes to Chapter 3. F3-04ADS 4-Channel Isolated Analog Input

Page 3-3. Setting the Module Jumpers; Jumper Locations

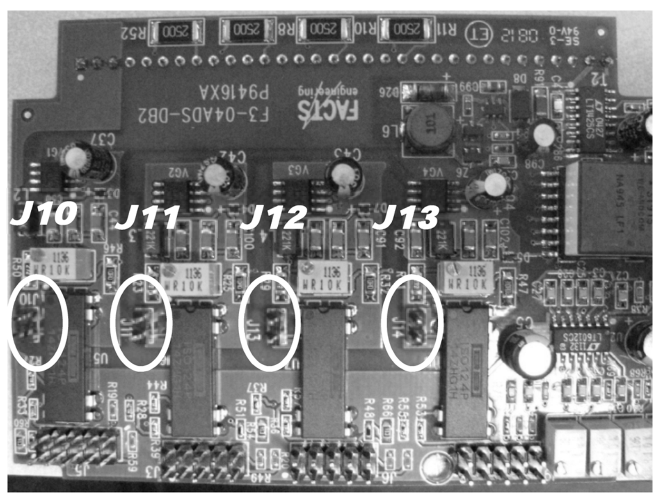
The PC board was redesigned and the locations of jumpers J10, J11, J12, and J13 changed. The jumpers were rotated 90 degrees and are closer to the back of the module than the original layout. The functionality of the jumpers did not change. The orientaton of the 5 pairs of pins for each channel is the same.

The photo on the right shows the new design, while the one on the left shows the original PC board. The photo on the left matches the drawing shown on page 3-3. The redesigned PC boards are in modules manufactured starting in mid-2012.0

Original PC Board Layout (Manufactured prior to mid-2012)



Redesigned PC Board Layout (Manufactured after mid-2012)



Errata Sheet

Changes to Chapter 5. F3–16AD 16-Channel Analog Input

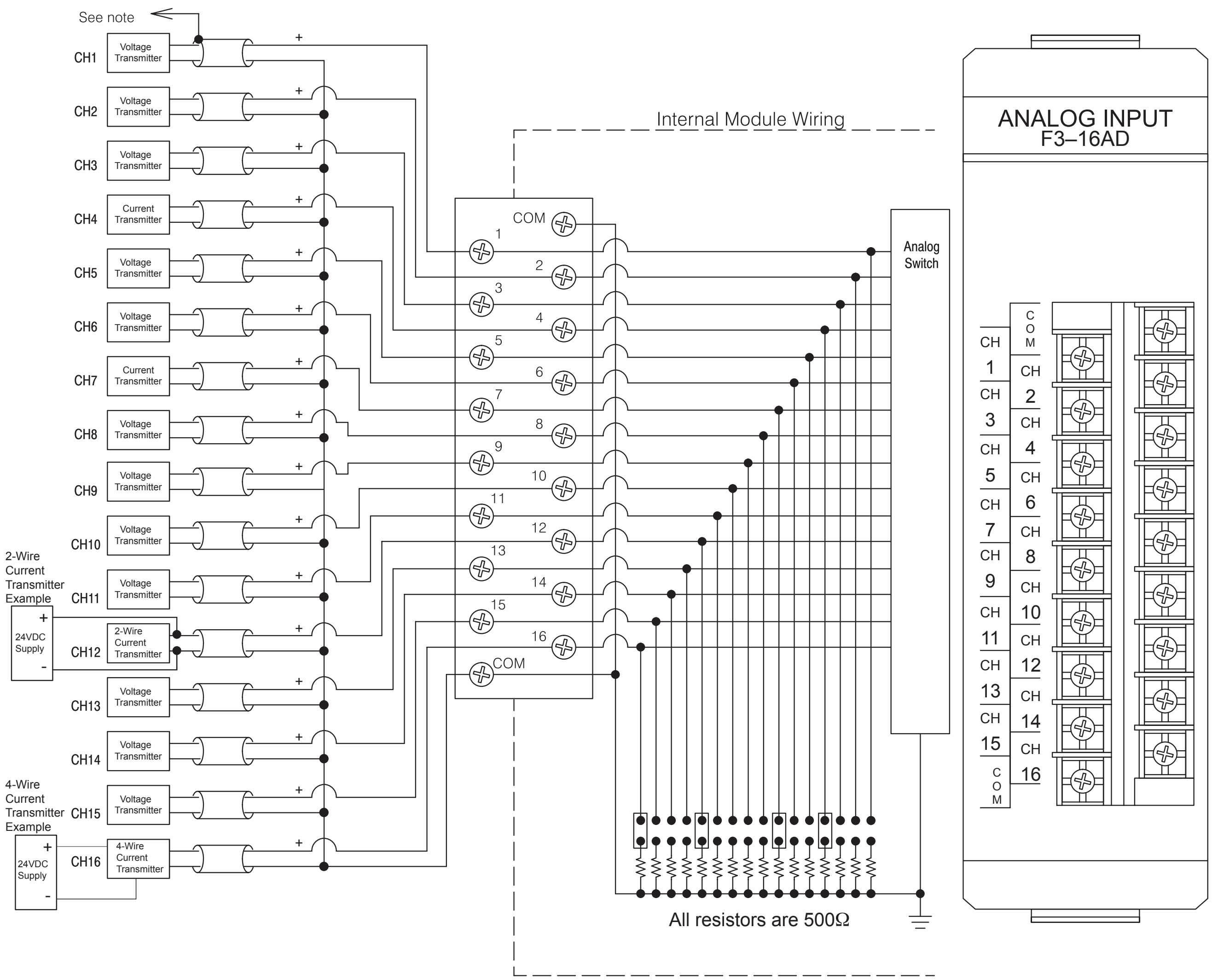
Page 5-9. Wiring Diagram

The wiring diagram shows “current transmitters” CH 4, 7, 12, and 16. The diagram should show external 24VDC power supplies for these current transmitters. A 2-wire current transmitter example of this has been added to the diagram below for CH12.

Also, CH16 has been changed to show a 4-wire current transmitter example.

Wiring Diagram

Note 1: Terminate all shields at their respective signal source.
Note 2: Jumpers for CH4, 7, 12 and 16 are installed for current input.



3

F3–04ADS 4-Channel Isolated Analog Input

In This Chapter. . . .

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

Module Specifications

The following table provides the specifications for the F3–04ADS Analog Input Module. Make sure the module meets your application requirements.

Number of Channels	4, isolated
Input Ranges	0 – 5V, 0 – 10V, 1 – 5V, ±5V, ±10V, 0 – 20 mA, 4 – 20 mA
Resolution	12 bit (1 in 4096)
Input Type	Differential
Max. Common mode voltage	±750V peak continuous transformer isolation
Noise Rejection Ratio	Common mode: –100 dB at 60Hz
Active Low-pass Filtering	–3 dB at 10Hz, –12 dB per octave
Input Impedance	250Ω ±0.1%, 1/2W current input 200KΩ voltage input
Absolute Maximum Ratings	±40 mA, current input ±100V, voltage input
Conversion Time	1 channel per scan, successive approximation, AD574
Linearity Error	±1 count (0.03% of full scale) maximum
Full Scale Calibration Error	±9 counts maximum
Offset Calibration Error	±4 counts maximum, bipolar ranges ±2 counts maximum, unipolar ranges
Accuracy vs. Temperature	57 ppm / °C maximum full scale
Recommended Fuse	0.032 A, Series 217 fast-acting, current inputs
Power Budget Requirement	183 mA @ 9 VDC, 50 mA @ 24 VDC
External Power Supply	None required
Operating Temperature	32° to 140° F (0° to 60° C)
Storage Temperature	–4° to 158° F (–20° to 70° C)
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

Analog Input Configuration Requirements

The F3–04ADS Analog Input appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

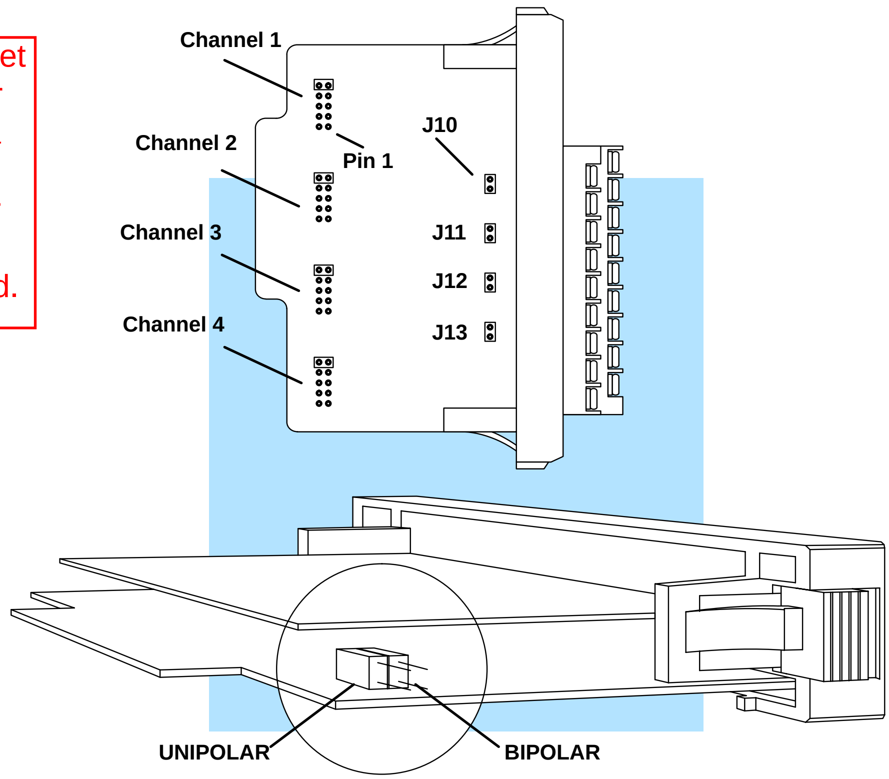
- The module should not be placed in the last slot of a rack (due to size constraints.)
- For local and expansion systems, the available power budget and 16-point module usage are the limiting factors.

Setting the Module Jumpers

Jumper Locations

The module is set at the factory for a 4–20 mA signal on all four channels. If this is acceptable you do not have to change any of the jumpers. The following diagram shows how the jumpers are set.

See the Errata Sheet on the first page for information about a circuit board re-design. The jumper locations on newer PC boards changed.



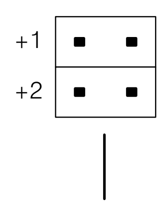
Selecting the Number of Channels

If you examine the rear of the module, you'll notice several jumpers. The jumpers labeled +1 and +2 (located on the larger board, near the terminal block) are used to select the number of channels that will be used.

Without any jumpers the module processes one channel. By installing the jumpers you can add channels. The module is set from the factory for four channel operation.

For example, if you install the +1 jumper, you add one channel for a total of two. Now if you install the +2 jumper you add two more channels for a total of four.

Any unused channels are not processed so if you only select channels 1, 2, and 3, channel 4 will not be active. The table shows which jumpers to install.



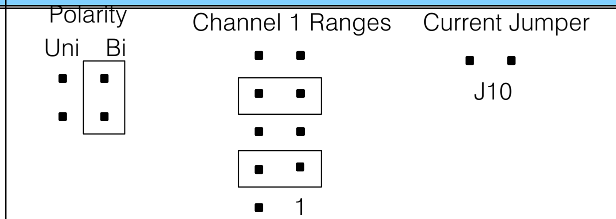
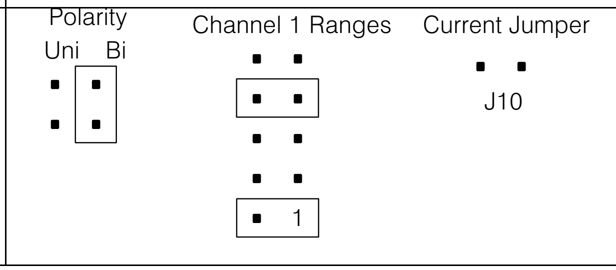
Jumpers installed as shown selects 4-channel operation

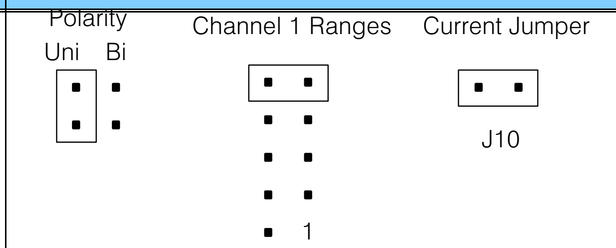
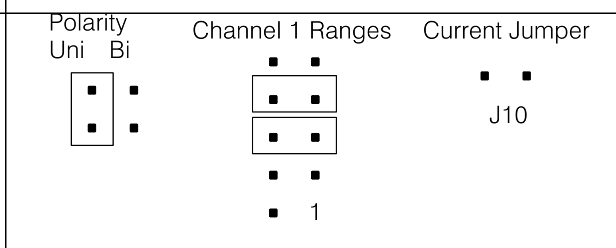
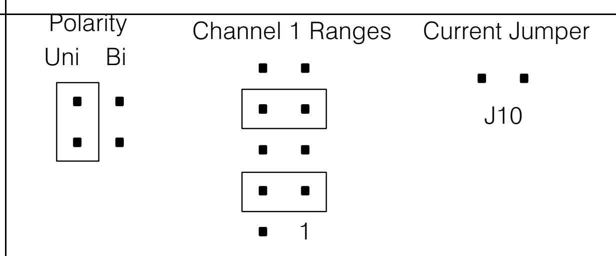
Channel	+1	+2
1	No	No
1, 2,	Yes	No
1, 2, 3	No	Yes
1, 2, 3, 4	Yes	Yes

Selecting Input Signal Ranges

As you examin the jumper settings, notice there are jumpers for each individual channel. These jumpers allow you to select the type of signal (voltage or current) and the range of the signal. The following tables show the jumper selections for the various ranges. Only channel 1 is used in the example, but all channels must be set.

NOTE: The Polarity jumper selects Unipolar or Bipolar operation for all channels.

Bipolar Signal Range	Jumper Settings
–5 VDC to +5 VDC (–20 to +20 mA)	
–10 VDC to +10 VDC	

Unipolar Signal Range	Jumper Settings
4 to 20 mA (1 VDC to 5 VDC, remove the current jumper)	
0 VDC to +5 VDC (0 to +20 mA, install the current jumper)	
0 VDC to +10 VDC	

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.

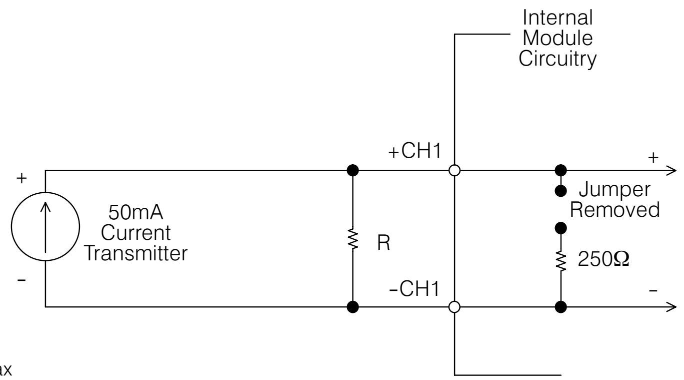
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the source.
- Do not run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.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

The F3–04ADS receives all power from the base. A separate power supply is not required.

Custom Input Ranges

Occasionally you may have the need to connect a transmitter with an unusual signal range. By changing the wiring slightly and adding an external resistor to convert the current to voltage, you can easily adapt this module to meet the specifications for a transmitter which does not adhere to one of the standard input ranges. The following diagram shows how this works.



$$R = \frac{V_{max}}{I_{max}}$$

R = value of external resistor

V_{max} = high limit of selected voltage range

I_{max} = maximum current supplied by the transmitter

Example: current transmitter capable of 50mA, 0 – 10V range selected.

$$R = \frac{10V}{50mA} \qquad R = 200 \text{ ohms}$$

NOTE: Your choice of resistor can affect the accuracy of the module. A resistor with a ±0.1% tolerance and a ±50ppm / °C temperature coefficient is recommended.

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 F3–04ADS provides 250 ohm resistance for each channel. If your transmitter requires a load resistance below 250 ohms, then you do not have to make any adjustments. However, if your transmitter requires a load resistance higher than 250 ohms, then you need to add a resistor in series with the module.

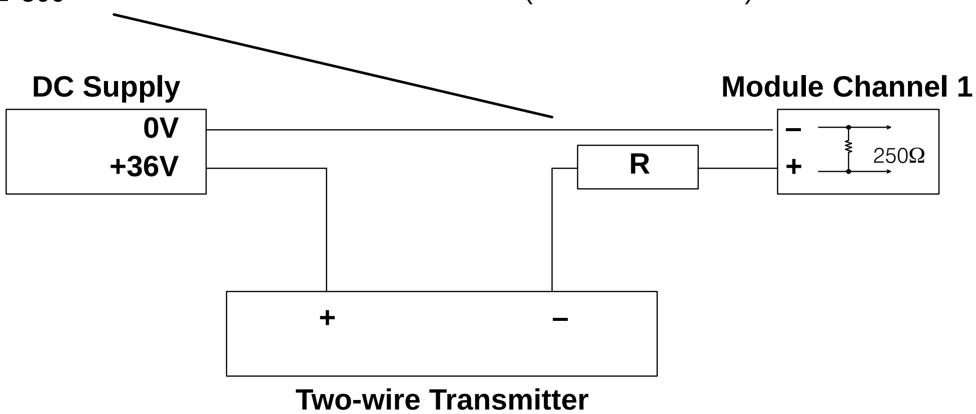
Consider the following example for a transmitter being operated from a 36 VDC supply with a recommended load resistance of 750 ohms. Since the module has a 250 ohm resistor, you need to add an additional resistor.

$R = Tr - Mr$

$R = 750 - 250$

$R \geq 500$

R – Resistor to add
Tr – Transmitter Requirement
Mr – Module resistance (internal 250 ohms)



Removable Connector

The F3–04ADS module has a removable connector to make wiring easier. Simply squeeze the top and bottom tabs and gently pull the connector from the module.

Wiring Diagram

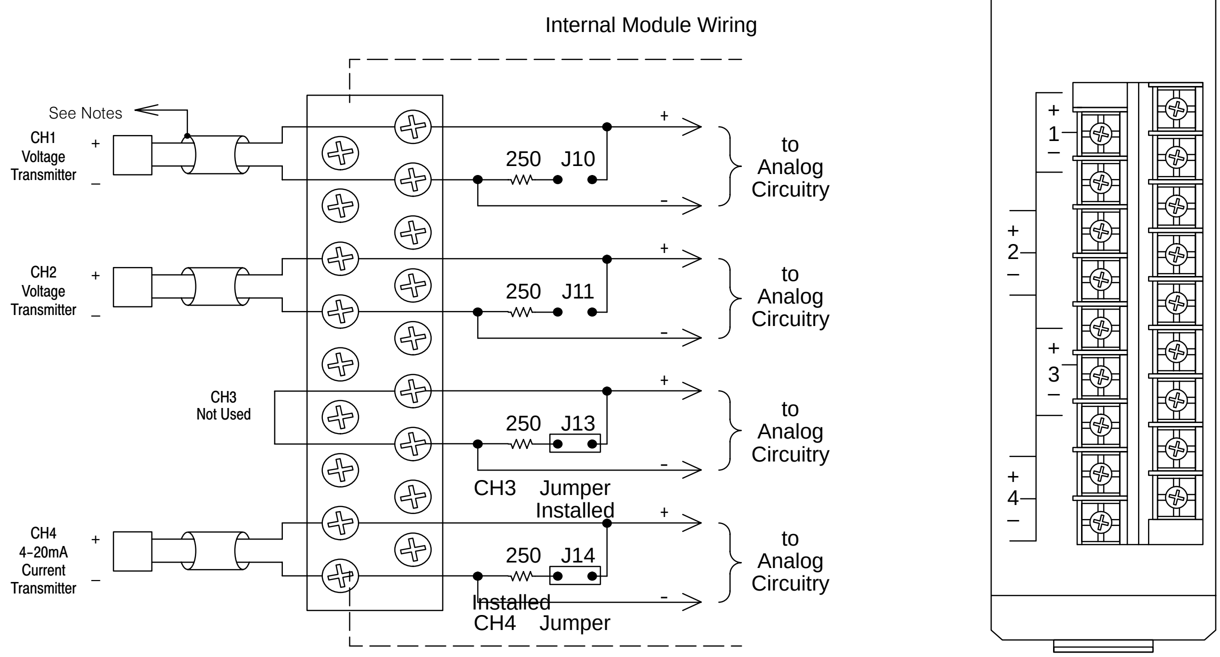
Note 1: Connect unused voltage or current inputs to 0VDC at terminal block or leave current jumper installed (see Channel 3).

Note 2: A Series 217, 0.032A, Fast-acting fuse is recommended for 4–20mA current loops.

Note 3: Transmitters may be 2, 3, or 4 wire type.

Note 4: Transmitters may be powered from separate power sources.

Note 5: Terminate all shields of the cable at their respective signal source.



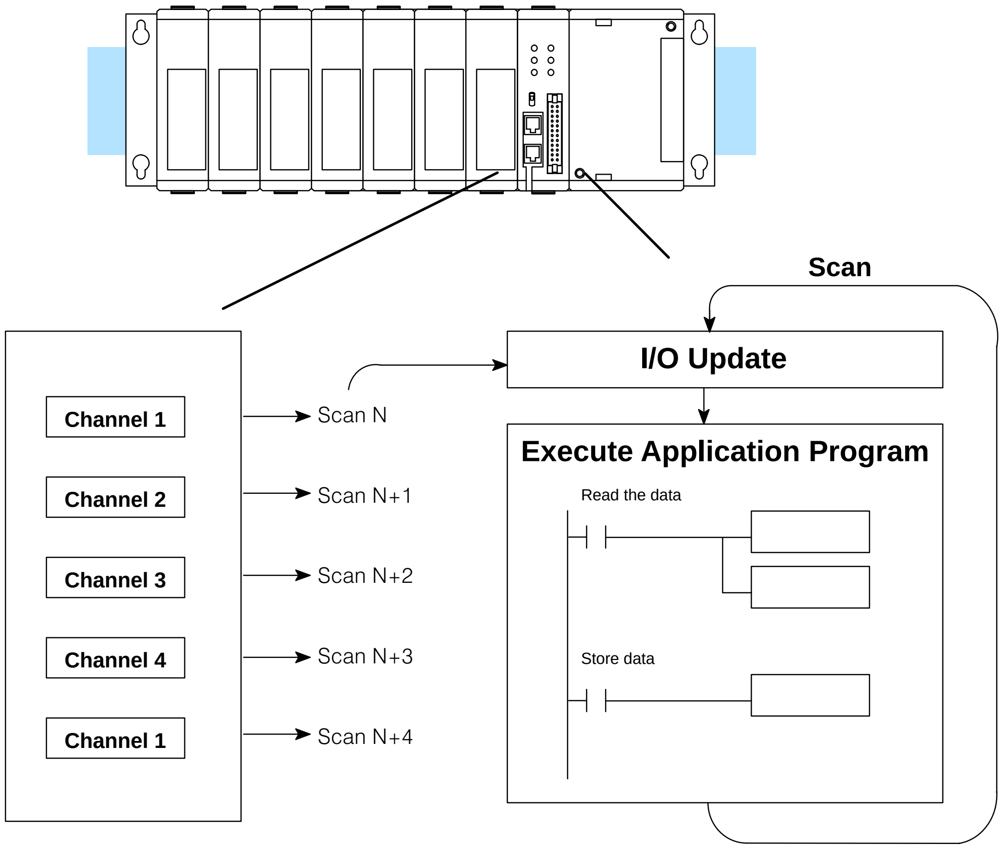
Module Operation

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

The F3–04ADS module supplies1 channel of data per each 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.

You do not have to select all of the channels. Unused channels are not processed, so if you select only two channels, then each channel will be updated every other scan.



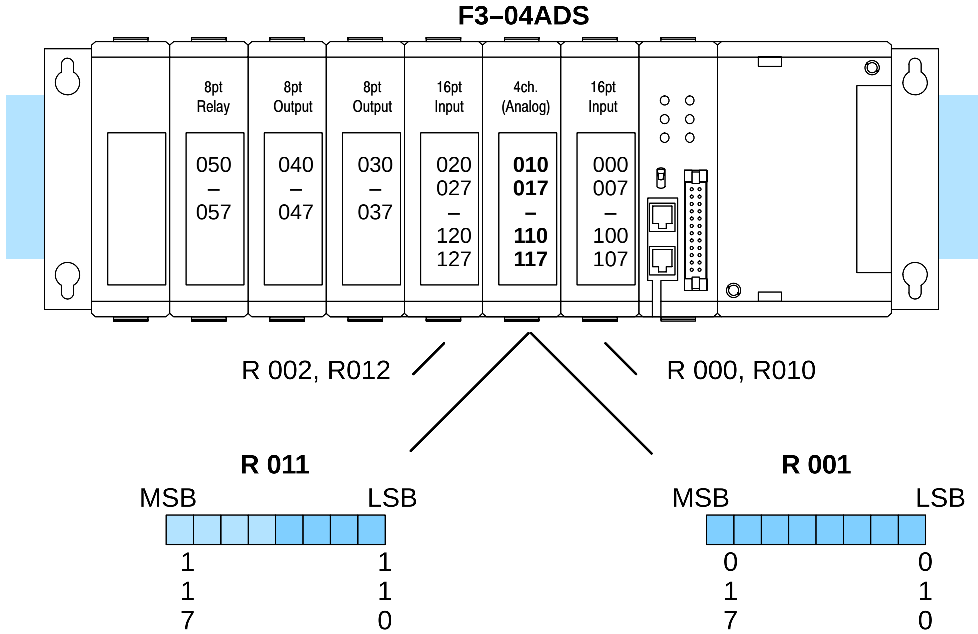
Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 12-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete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

Understanding the I/O Assignments

You may recall the F3–04ADS module appears to the CPU as a 16-point module. These 16 points provide:

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

Since all I/O points are automatically mapped into Register (R) memory, it is very easy to determine the location of the data word that will be assigned to the module.

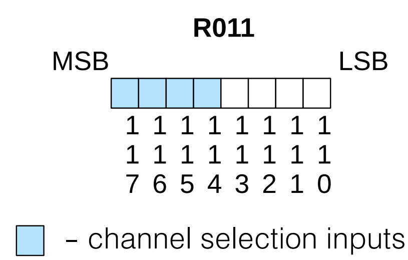


Within these two register locations, the individual bits represent specific information about the analog signal.

Active Channel Selection Inputs

The last four points of the upper register are used as inputs to tell the CPU which channel is being processed. In our example, when input 114 is on the module is telling the CPU it is processing channel 1. Here's how the inputs are assigned.

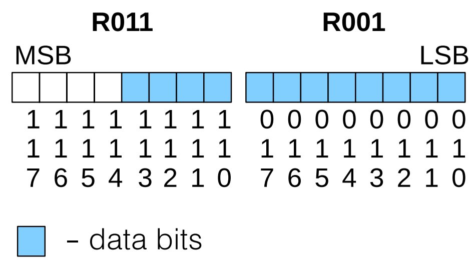
Input	Active Channel
114	1
115	2
116	3
117	4



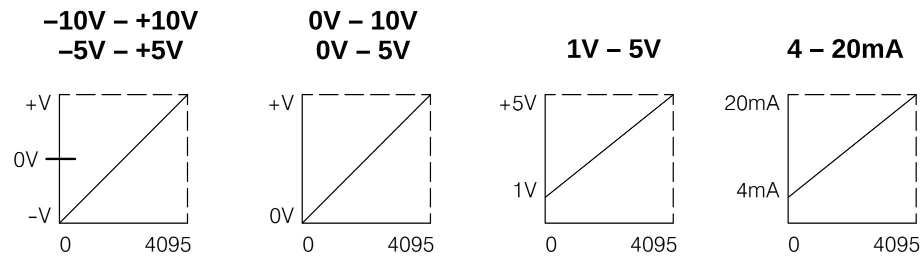
Analog Data Bits

The remaining twelve bits represent the analog data in binary format.

Bit	Value	Bit	Value
0 (LSB)	1	6	64
1	2	7	128
2	4	8	256
3	8	9	512
4	16	10	1024
5	32	11	2048



Since the module has 12-bit resolution, the analog signal is converted into 4096 “pieces” ranging from 0 – 4095 (2^{12}). For example, with a 0 to 10V scale, a 0V signal would be 0, and a 10V signal would be 4095. This is equivalent to a a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF hexadecimal. The following diagram shows how this relates to each signal range.



Each “piece” can also be expressed in terms of the signal level by using the equation shown. The following table shows the smallest signal levels that will result in a change in the data value for each signal range.

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

H = high limit of the signal range

L = low limit of the signal range

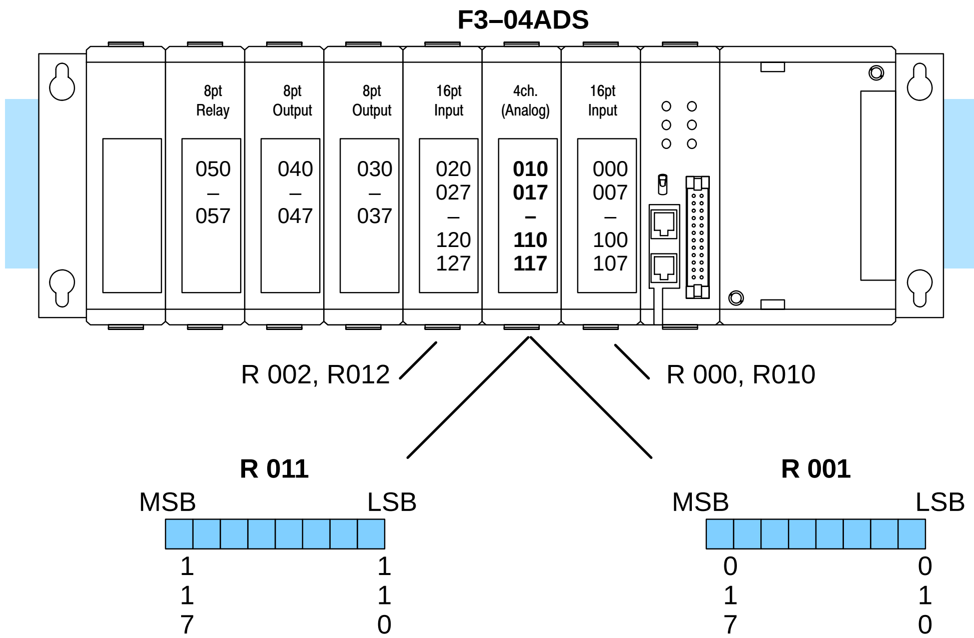
Range	Highest Signal	Lowest Signal	Smallest Change
–10 to +10V	+10V	–10V	4.88 mV
–5 to +5V	+5 V	–5V	2.44 mV
0 to 5V	5V	0V	1.22 mV
0 to 10V	10 V	0V	2.44 mV
1 to 5V	5V	1V	0.98 mV
4 to 20mA	20mA	4mA	3.91 μA

Now that you understand how the module and CPU work together to gather and store the information, you're ready to write the control program.

Writing the Control Program (DL330 / DL340)

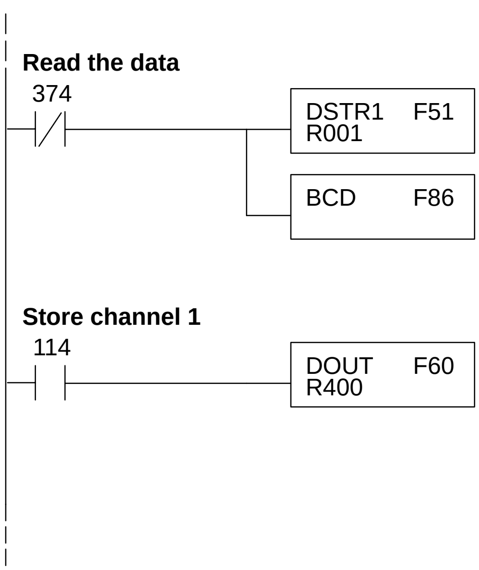
Identifying the Data Locations

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 provides input points to the CPU, it is very easy to use the active channel status bits to determine which channel is being monitored.



Single Channel on Every Scan

The following example shows a program that is designed to read a single channel of analog data into a Register location on every scan. Once the data is in a Register, you can perform math on the data, compare the data against preset values, etc. This example is designed to read channel 1. Since you use jumpers to select the number of channels to scan, this is the only channel that you can use in this manner.



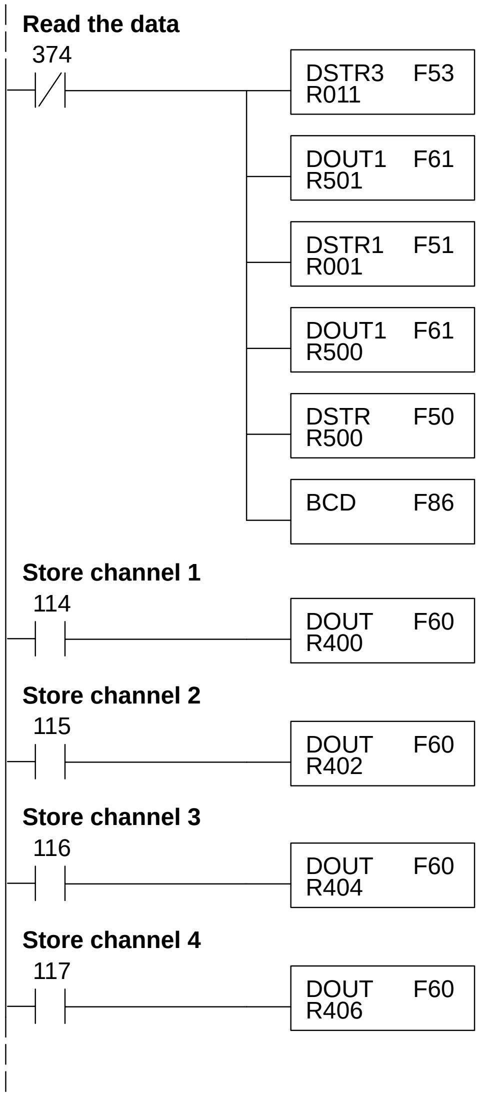
This rung loads the data into the accumulator on every scan. (You can use any permissive contact.)

The DL305 CPUs perform math operations in BCD. This instruction converts the binary data to BCD. (You can omit this step if your application does not require the conversion.)

The active channel inputs are used to let the CPU know which channel has been loaded into the accumulator. (Since you cannot isolate the individual channels for scanning, channel 1 is the only channel that can be used in this manner.) By using the input to control a DOUT instruction, you can easily move the data to a storage register. The BCD value will be stored in R400 and R401. (Two bytes are required for four digit BCD numbers.)

Reading Multiple Channels over Alternating Scans

The following example shows a program designed to read any of the available channels of analog data into Register locations. Once the data is in a Register, you can perform math on the data, compare the data against preset values, etc. Since the DL305 CPUs use 8-bit word instructions, you have to move the data in pieces. It's simple if you follow the example.



This rung loads the four most significant data bits into the accumulator from Register 011. (A normally closed 374 means it is loaded on every scan.)

Temporarily store the bits to Register 501.

This rung loads the eight least significant data bits into the accumulator from Register 001.

Temporarily store the bits to Register 500. Since the most significant bits were loaded into 501, now R500 and R501 contain all twelve bits in order.

Now that all the bits are stored, load all twelve bits into the accumulator.

Math operations are performed in BCD. This instruction converts the binary data to BCD. (You can omit this step if your application does not require the conversion.)

The channel selection inputs are used to let the CPU know which channel has been loaded into the accumulator. By using these inputs to control a DOUT instruction, you can easily move the data to a storage register. Notice the DOUT instruction stores the data in two bytes. (Two bytes are required for four digit BCD numbers.)

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.

$$\text{Units} = \frac{A}{4096} S$$

Units = value in Engineering Units

A = Analog value (0 – 4095)

S = high limit of the Engineering unit range

The following example shows how you would use the analog data to represent pressure (PSI) from 0 to 100. This example assumes the analog value is 1760. This should yield approximately 42.9 PSI.

$$\text{Units} = \frac{A}{4096} S$$

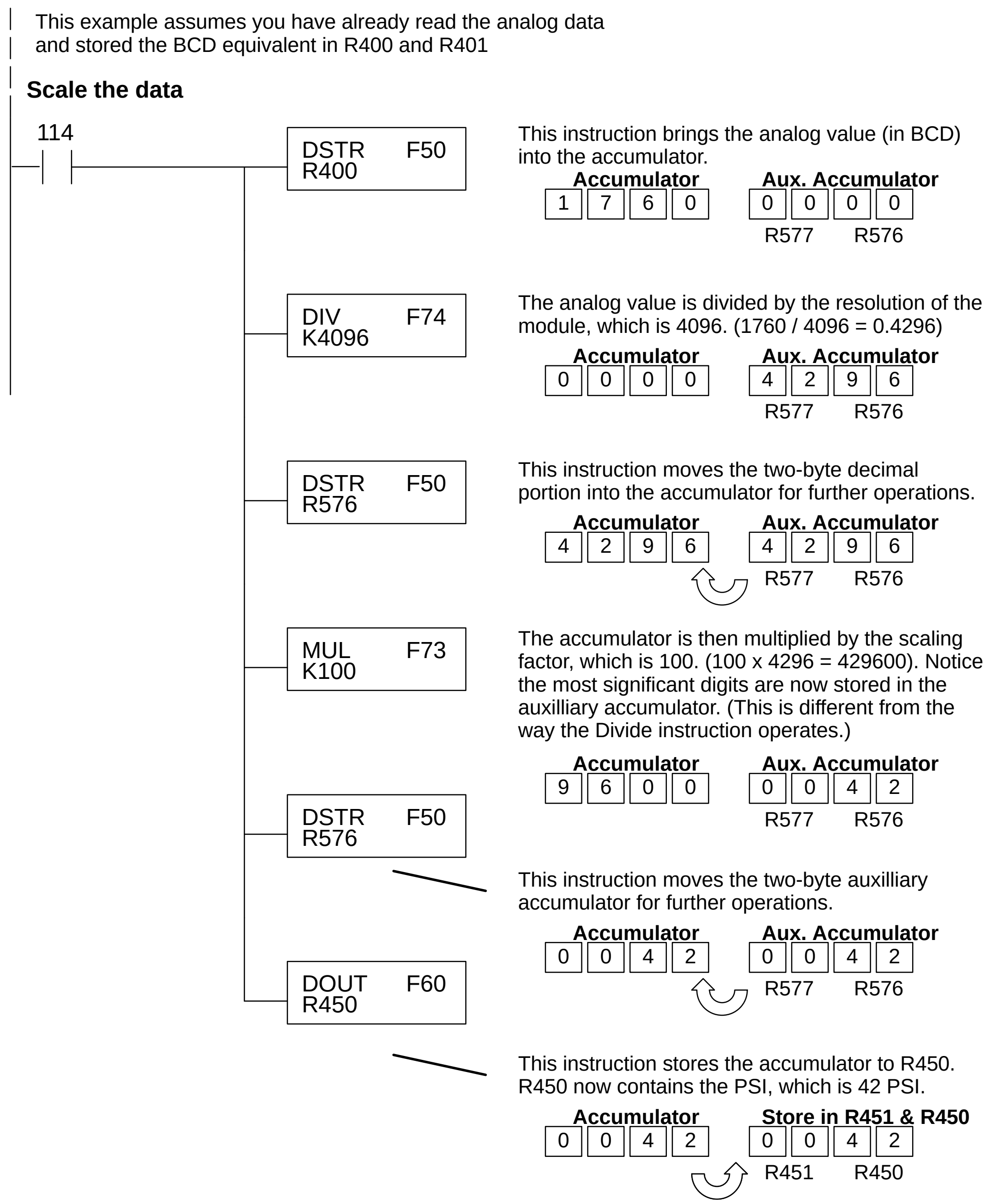
$$\downarrow$$

$$\text{Units} = \frac{1760}{4096} 100$$

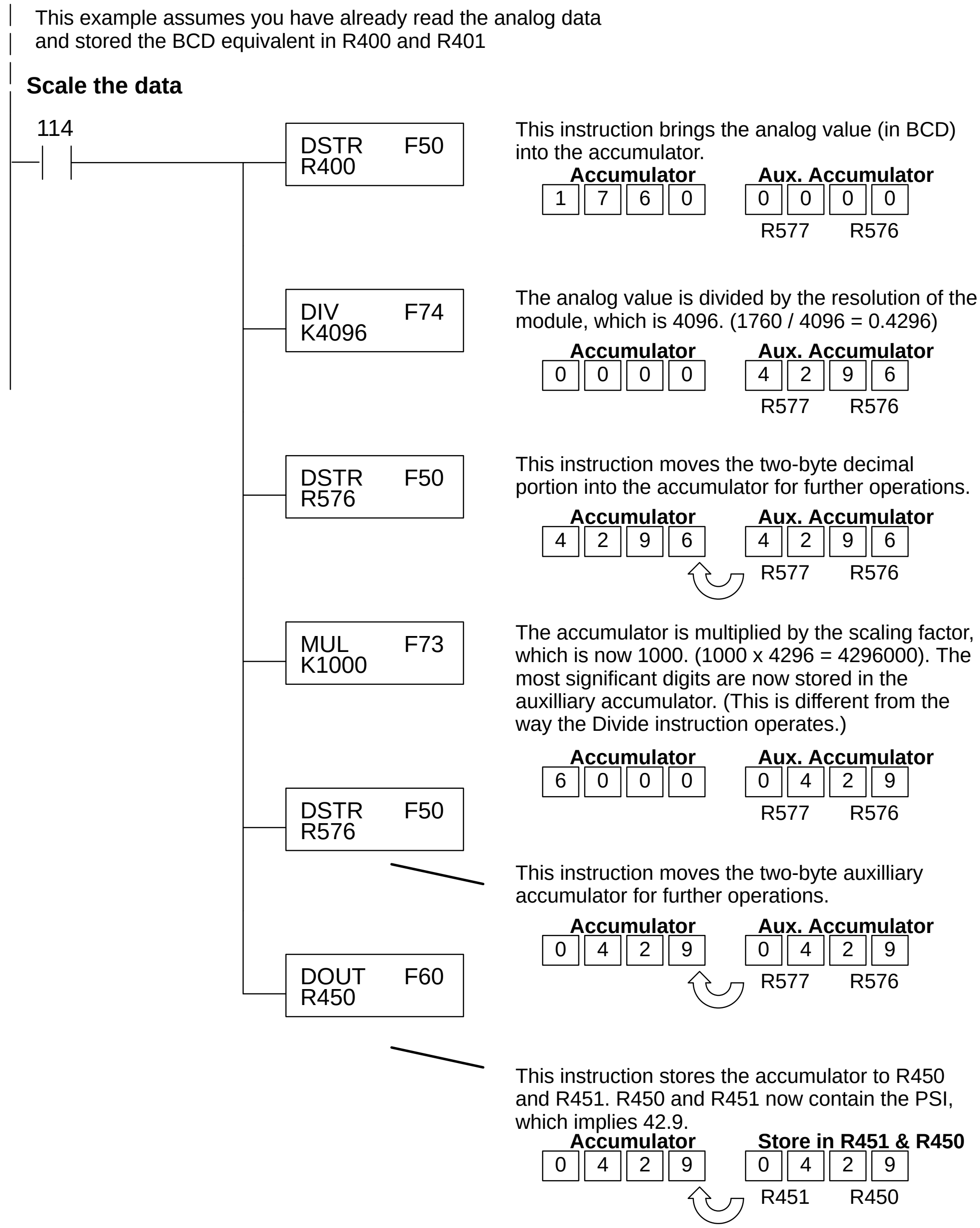
$$\downarrow$$

$$\text{Units} = 42.9$$

The following instructions are required to scale the data. We'll continue to use the 42.9 PSI example. In this example we're using channel 1. Input 114 is the active channel indicator for channel 1. Of course, if you were using a different channel, you would use the active channel indicator point that corresponds to the channel you were using.

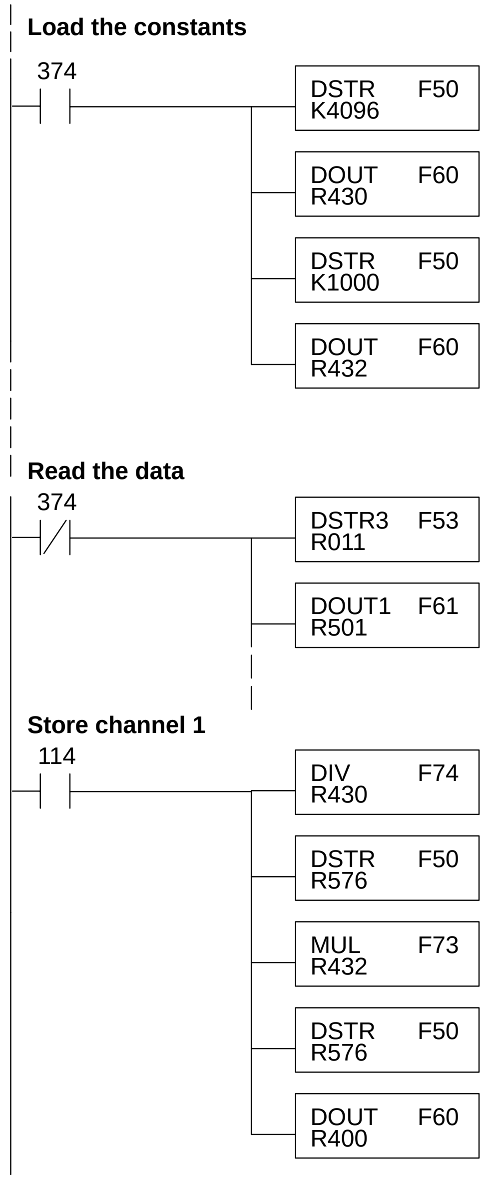


You probably noticed the previous example yielded 42 PSI when the real value should have been 42.9 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve added another digit to the scale. Instead of a scale of 100, we’re using 1000, which implies 100.0 for the PSI range.



This example program shows how you can use the instructions to load these equation constants into data registers. The example is written for channel 1, but you can easily use a similar approach to use different scales for all channels if required.

You may just use the appropriate constants in the instructions dedicated for each channel, but this method allows easier modifications. For example, you could easily use an operator interface or a programming device to change the constants if they are stored in Registers.



On the first scan, these first two instructions load the analog resolution (constant of 4096) into R430 and R431.

These two instructions load the high limit of the Engineering unit scale (constant of 1000) into R432 and R433. Note, if you have different scales for each channel, you'll also have to enter the Engineering unit high limit for those as well.

This rung loads the four most significant data bits into the accumulator from Register 011.

Temporarily store the bits to Register 501.

The analog value is divided by the resolution of the module, which is stored in R430.

This instruction moves the decimal portion from the auxilliary accumulator into the regular accumulator for further operations.

The accumulator is multiplied by the scaling factor, which is stored in R432.

This instruction moves most significant digits (now stored in the auxilliary accumulator) into the regular accumulator for further operations.

The scaled value is stored in R400 and R401 for further use.

Writing the Control Program (DL350)

Reading Values: Pointer Method and Multiplexing

There are two methods of reading values for the DL350:

- The pointer method **(all system bases must be D3–xx–1 bases to support the pointer method)**
- Multiplexing

You must use the multiplexing method with remote I/O modules (the pointer method will not work). You can use either method when using DL350, but for ease of programming it is strongly recommended that you use the pointer method.

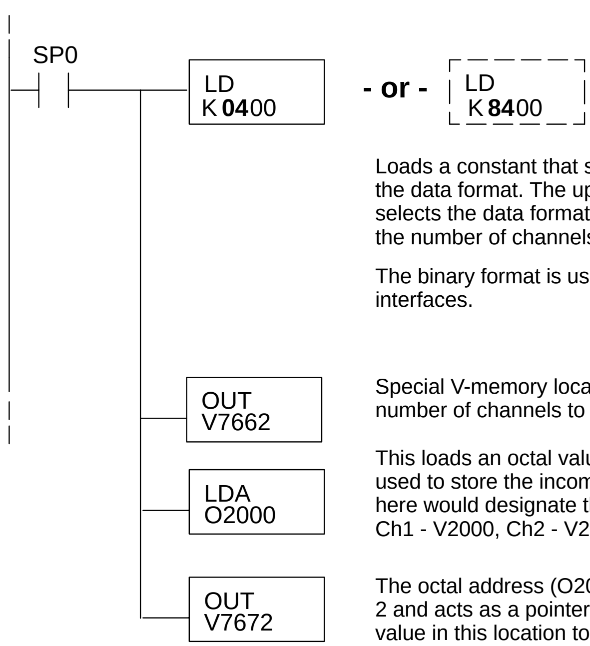
NOTE: Do not use the pointer method and the PID PV auto transfer from I/O module function together for the same module. If using PID loops, use the pointer method and ladder logic code to map the analog input data into the PID loop table.

Pointer Method

The DL350 has special V-memory locations assigned to each base slot that greatly simplifies the programming requirements. These V-memory locations allow you to:

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

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 RLL*PLUS* 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.



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).

The binary format is used for displaying data on some operator interfaces.

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, Ch 4 - V2003

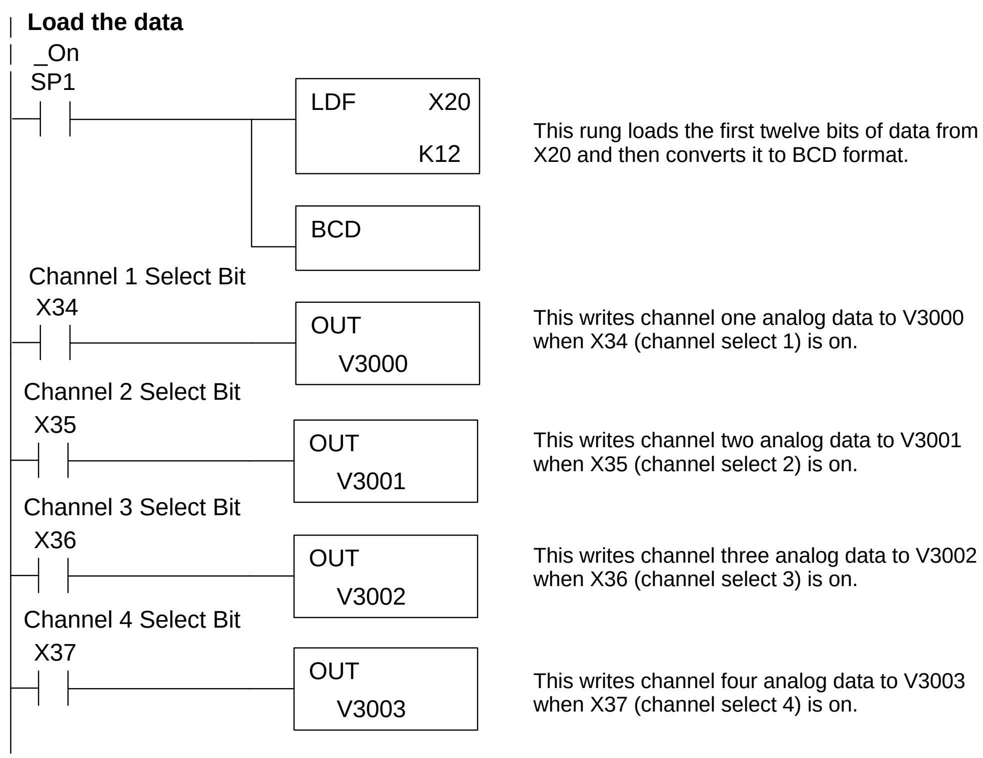
The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming data.

The table shows the special V-memory locations used with the DL350. Slot 0 (zero) is the module next to the CPU, slot 1 is the module two places from the CPU, and so on. Remember, the CPU only examines the pointer values at these locations after a mode transition. The pointer method is supported on expansion bases up to a total of 8 slots away from the DL350 CPU. The pointer method is not supported in slot 8 of a 10 slot base.

Analog Input 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
Storage Pointer	V7670	V7671	V7672	V7673	V7674	V7675	V7676	V7677

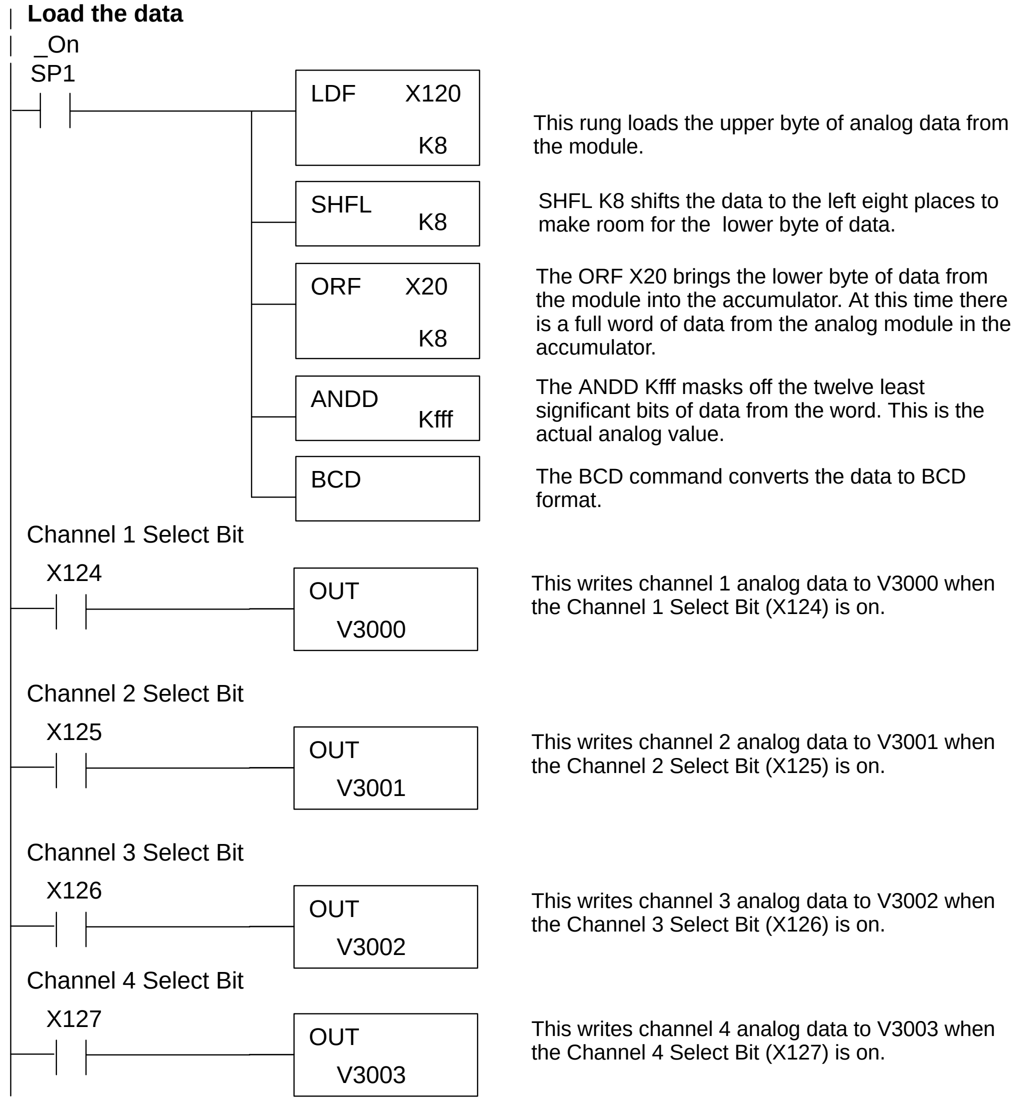
Multiplexing: DL350 with a D3–xx–1 Base

The example below shows how to read multiple channels on a F3–04ADS Analog module in the X20 adddress position of the D3–XX–1 base. If any expansion bases are used in the system, they must all be D3–xx–1 to be able to use this example. Otherwise, the conventional base addressing must be used.



Multiplexing: DL350 with a Conventional DL305 Base

The example below shows how to read multiple channels on an F3–04ADS Analog module in the 20–27/120–127 address slot. This module must be placed in a 16 bit slot in order to work.



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

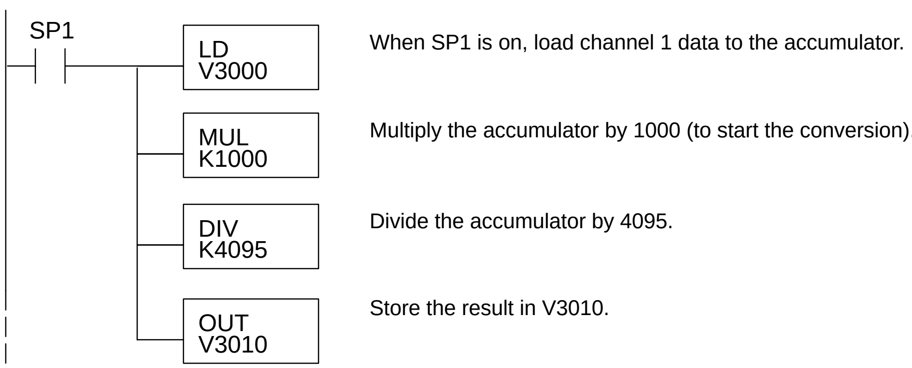
L = low limit of the engineering unit range

A = Analog value (0 – 4095)

For example, if you wanted to measure pressure (PSI) from 0.0 to 99.9 then 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.

Here is 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 helpful to be able to quickly convert between the signal levels and the digital values. This is especially helpful 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 ...
–10V to + 10V	$A = \frac{20D}{4095} - 10$	$D = \frac{4095}{20}(A + 10)$
–5V to + 5V	$A = \frac{10D}{4095} - 5$	$D = \frac{4095}{10}(A + 5)$
0 to 5V	$A = \frac{5D}{4095}$	$D = \frac{4095}{5}A$
0 to 10V	$A = \frac{10D}{4095}$	$D = \frac{4095}{10}A$
1 to 5V	$A = \frac{4D}{4095} + 1$	$D = \frac{4095}{4}(A - 1)$
4 to 20mA	$A = \frac{16D}{4095} + 4$	$D = \frac{4095}{16}(A - 4)$

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

$$D = \frac{4095}{20}(A + 10)$$

$$D = \frac{4095}{20}(6V + 10)$$

$$D = (204.75)\ (16)$$

$$D = 3276$$