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

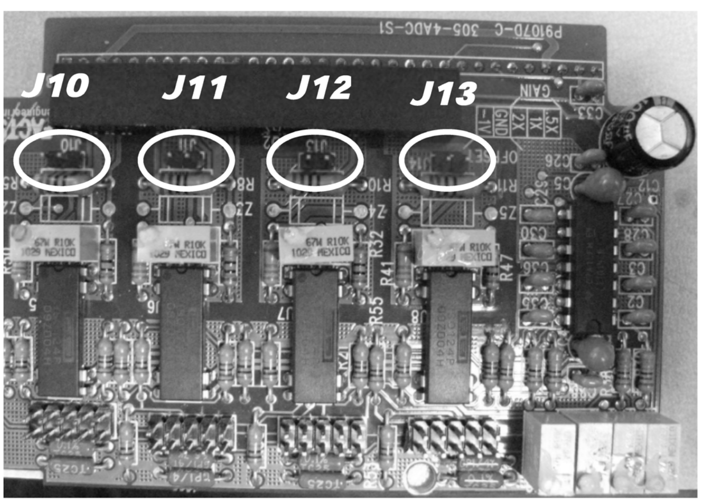
Changes to Chapter 3. F3-04ADS 4-Channel Isolated Analog Input

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

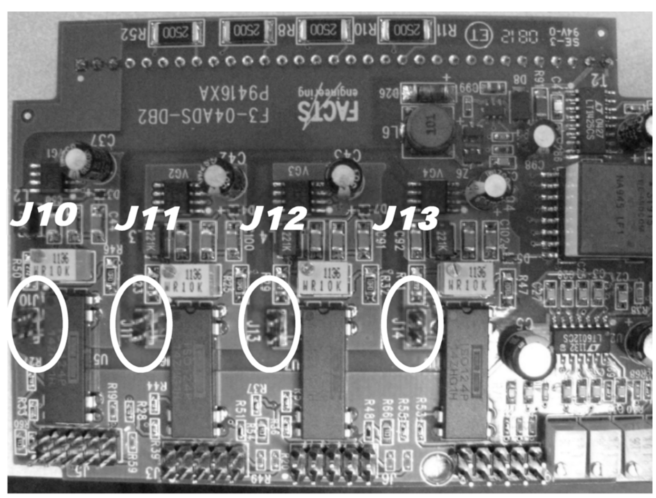
The PC board was redesigned and the locations of jumpers J10, J11, J12, and J13 changed. The jumpers were rotated 90 degrees and are closer to the back of the module than the original layout. The functionality of the jumpers did not change. The orientaton of the 5 pairs of pins for each channel is the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

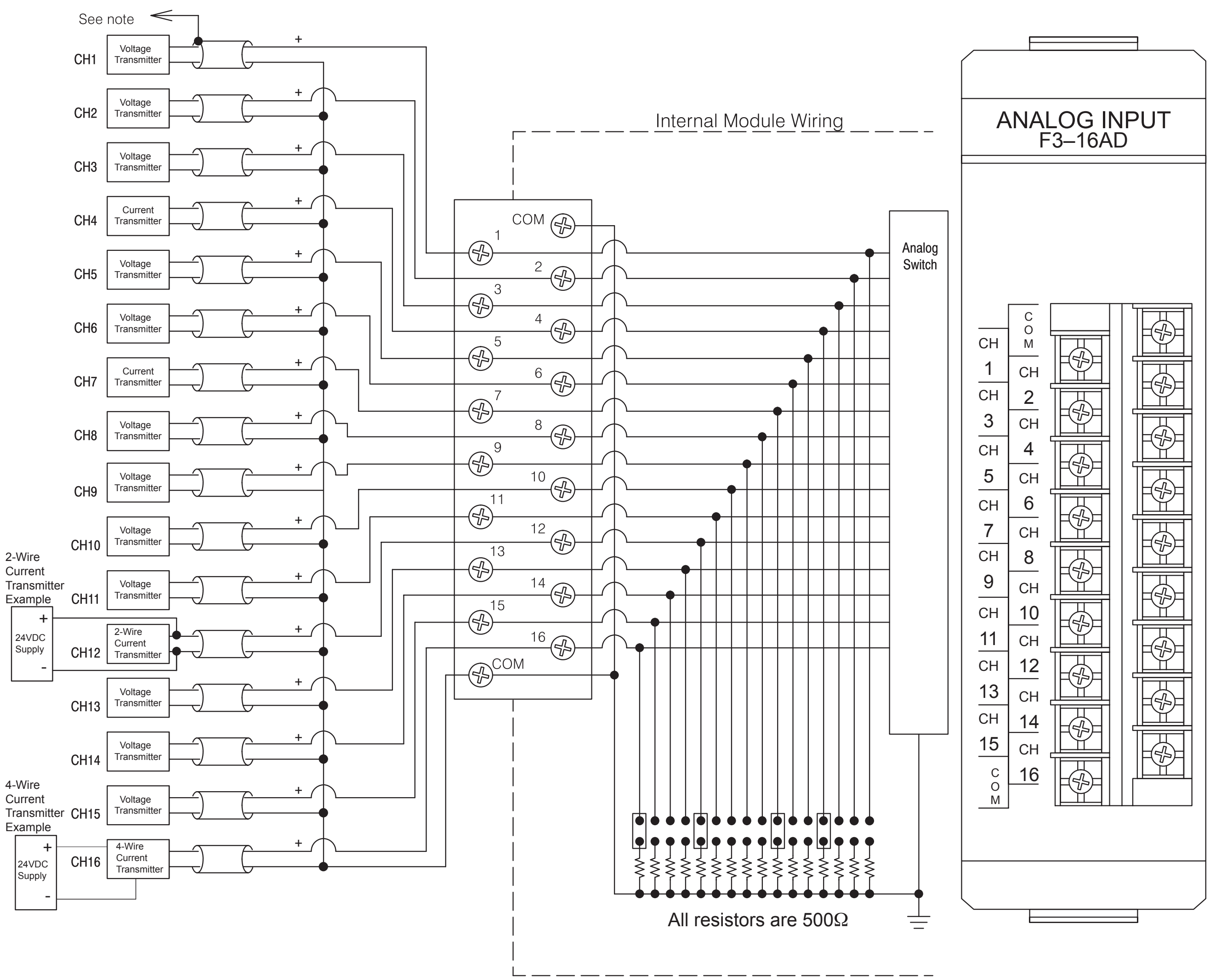
Page 5-9. Wiring Diagram

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.



D3–04AD 4-Channel Analog Input

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

2

In This Chapter. . . .

Module Specifications

The following table provides the specifications for the D3–04AD Analog Input Module. Review these specifications to make sure the module meets your application requirements.

Number of Channels	4
Input Ranges	1 – 5V, 4 – 20 mA
Resolution	8 bit (1 in 256)
Channel Isolation	Non-isolated (one common)
Input Type	Differential or Single ended
Input Impedance	1 MΩ minimum, voltage 250Ω current
Absolute Maximum Ratings	0 – +10V maximum, voltage 0 – 30 mA maximum, current
Linearity	± 0.8% maximum
Accuracy vs. Temperature	± 70 ppm / °C maximum
Maximim Inaccuracy	1% maximum at 25° C
Conversion Method	Sequential comparison
Conversion Time	2 ms maximum
Power Budget Requirement	55 mA @ 9V
External Power Supply	24 VDC, ± 10%, 65 mA, class 2
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
Noise Rejection Ratio	Normal mode: –6 dB/250Hz Common mode: 60dB/60Hz (–5 to 10V)

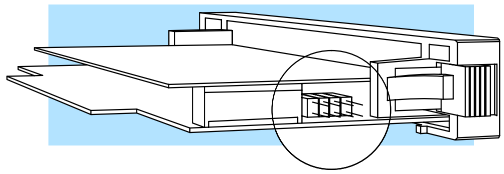
Analog Input Configuration Requirements

The D3–04AD 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:

- For local and expansion systems, the available power budget and 16-point module usage are the limiting factors.

Setting the Module Jumpers

There are four jumpers located on the module that select between 1–5V and 4–20 mA signals. The module is shipped from the factory for use with 1–5V signals.



If you want to use 4 – 20 mA signals, you have to install a jumper. No jumper is required for 1 – 5V operation. Each channel range may be selected independently of the others.

Range	Jumper
1 – 5V	Removed
4 – 20 mA	Installed

Connecting the Field Wiring

Wiring Guidelines

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

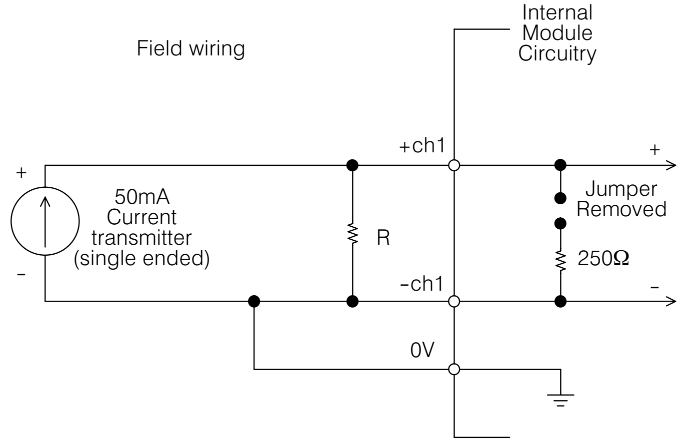
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the source.
- Don’t run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.
- Route the wiring through an approved cable housing to minimize the risk of accidental damage. Check local and national codes to choose the correct method for your application.

User Power Supply Requirements

The D3–04AD requires a separate power supply. The DL305 bases have built-in 24 VDC power supplies that provide up to 100 mA of current. If you only have one analog module, you can use this power source instead of a separate supply. If you have more than two analog modules, or you would rather use a separate supply, choose one that meets the following requirements: 24 VDC $\pm$10%, Class 2, 65mA current (or greater, depending on the number of modules being used.)

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 that 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, 1 – 5V range selected.

$$R = \frac{5V}{50mA} \qquad R = 100 \text{ ohms}$$

NOTE: Your choice of resistor can affect the accuracy of the module. A resistor that has ±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 D3–04AD 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

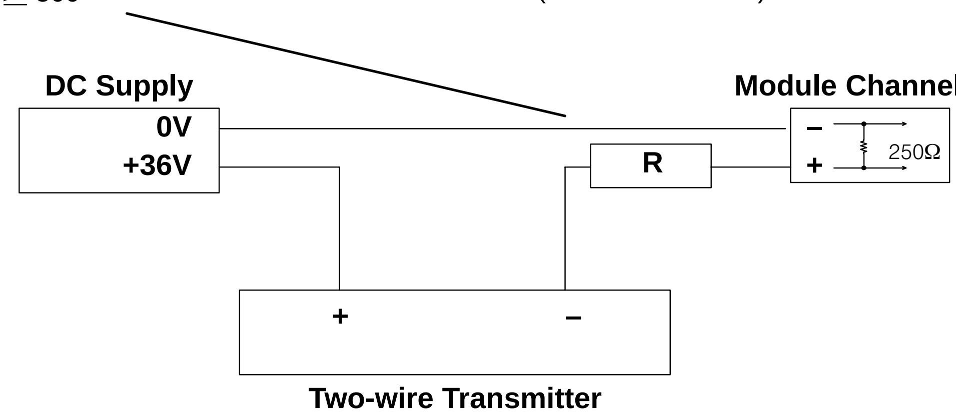
R = 750 – 250

$R \geq 500$

R – Resistor to add

Tr – Transmitter Requirement

Mr – Module resistance (internal 250 ohms)



Removable Connector

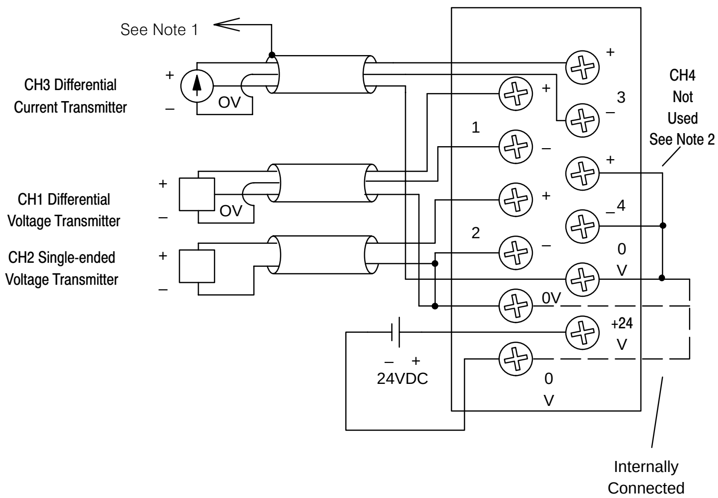
The D3–04AD module has a removable connector to make wiring easier. Simply squeeze the tabs on the top and bottom and gently pull the connector from the module.

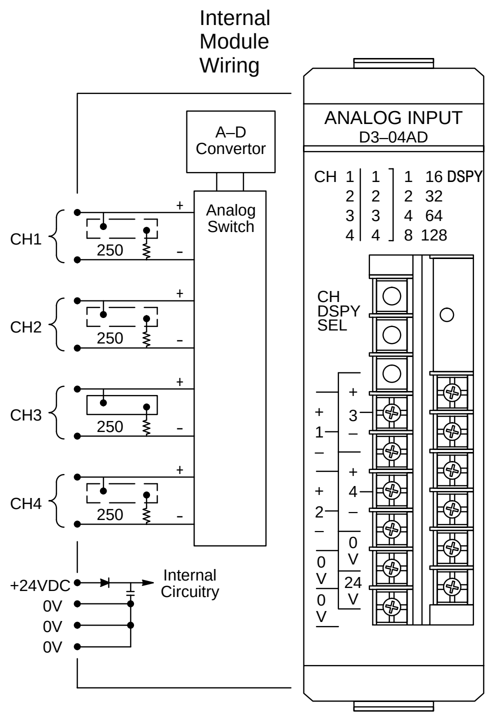
Wiring Diagram

Note 1: Terminate all shields of the cable at their respective signal source.

Note 2: Unused channels should be shorted to 0V or have the Jumper installed for current input for best noise immunity.

Note 3: When a differential input is not used 0V should be connected to the – of that channel.





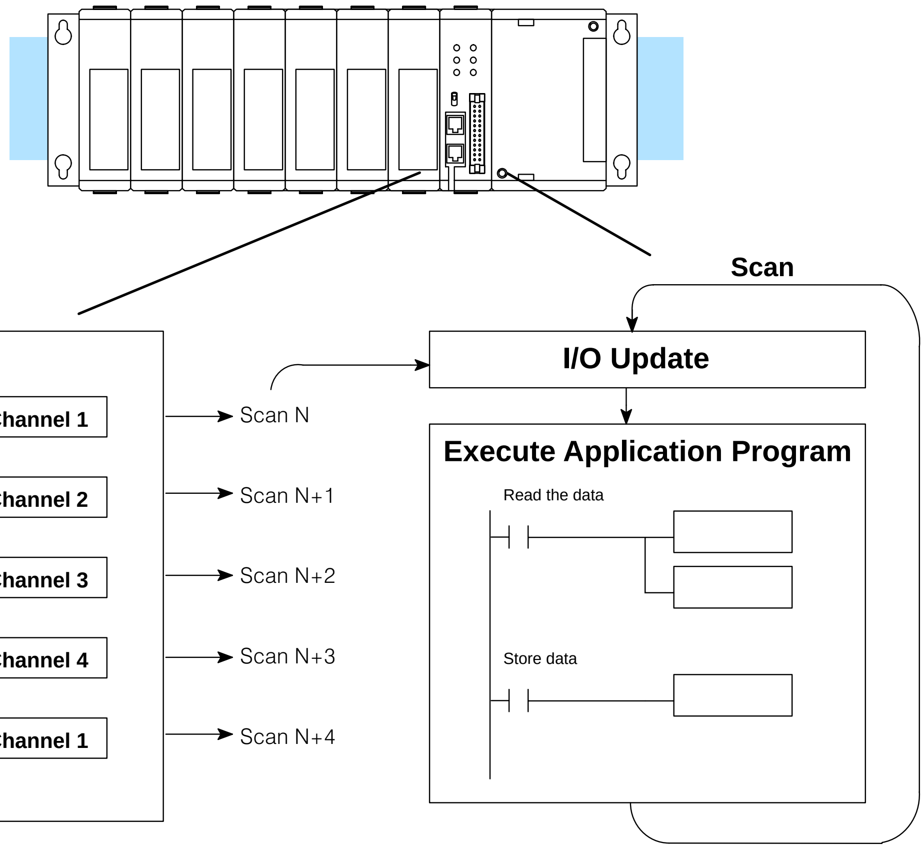
Module Operation

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 D3–04AD module supplies 1 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.



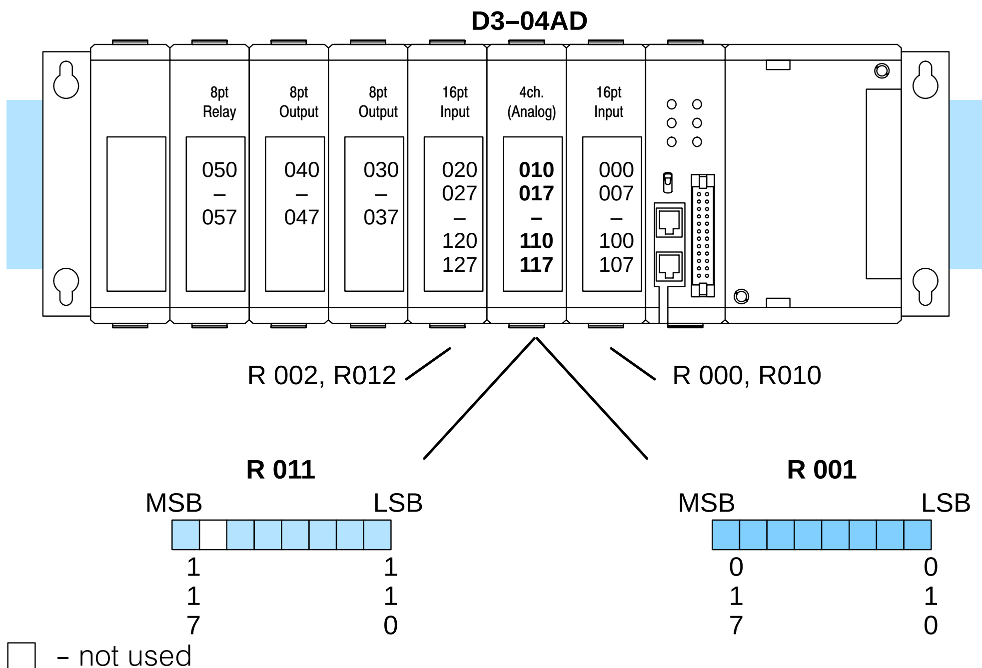
Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 8-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete control logic in the RLL program.

Understanding the I/O Assignments

You may recall the D3–04AD module appears to the CPU as a 16-point module. Some of the points are inputs to the CPU and some are outputs to the 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.

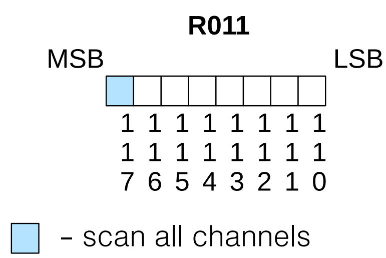


Within these two register locations, the individual bits represent specific information about the analog signal.

All Channel Scan Output

The most significant point (MSP) assigned to the module acts as an output to the module and controls the channel scanning sequence. This allows flexibility in your control program.

If this output is on, all channels will be scanned sequentially. If the output is off, you can use other points to select a single channel for scanning.



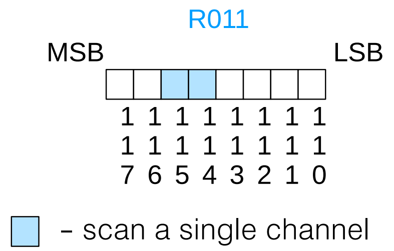
Scan	Out 117	Channel Input
N	Off	None
N+1	On	1
N+2	On	2
N+3	On	3
N+4	On	4
N+5	On	1
N+6	Off	None
N+7	Off	None

Single Channel Scan Outputs

The upper register also contains two additional outputs that can be used to choose a single channel for scanning. These outputs are *ignored* if the channel scan output is turned on.

(Note, our example shows outputs 114 and 115. Your output point will depend on where you have installed the module.)

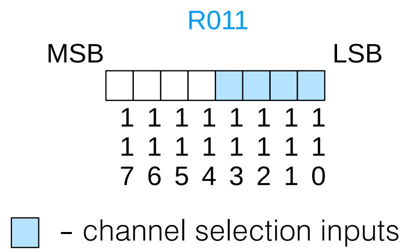
Out 114	Out 115	Channel
Off	Off	1
On	Off	2
Off	On	3
On	On	4



Active Channel Selection Inputs

The first four points of the upper register are used as inputs to tell the CPU which channel is being processed. (Remember, the previous bits only tell the module which channels to scan.) In our example, when input 110 is on the module is telling the CPU it is processing channel 1. Here's how the inputs are assigned.

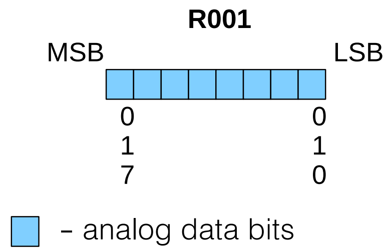
Input	Active Channel
110	1
111	2
112	3
113	4



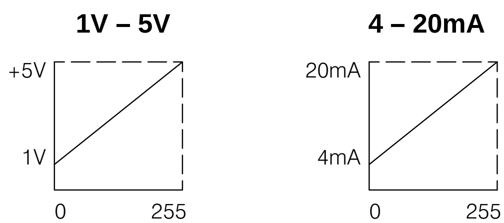
Analog Data Bits

The first register contains 8 bits which represent the analog data in binary format.

Bit	Value	Bit	Value
0	1	4	16
1	2	5	32
2	4	6	64
3	8	7	128



Since the module has 8-bit resolution, the analog signal is converted into 256 "pieces" ranging from 0 – 255 (2^8). For example, with a 1 to 5V scale, a 1V signal would be 0, and a 5V signal would be 255. This is equivalent to a a binary value of 0000 0000 to 1111 1111, or 00 to FF 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 could possibly result in a change in the data value for each signal range.

Resolution = (H–L)/255

H = high limit of the signal range

L = low limit of the signal range

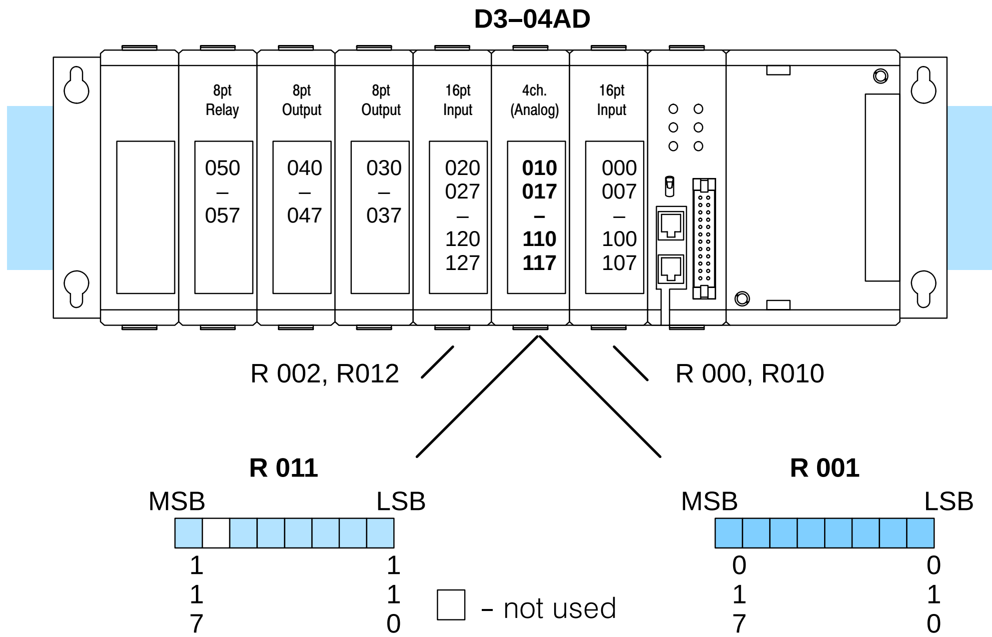
Range	Highest Signal	Lowest Signal	Smallest Change
1 to 5V	5V	1V	15.6 mV
4 to 20mA	20mA	4mA	62.7 µA

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)

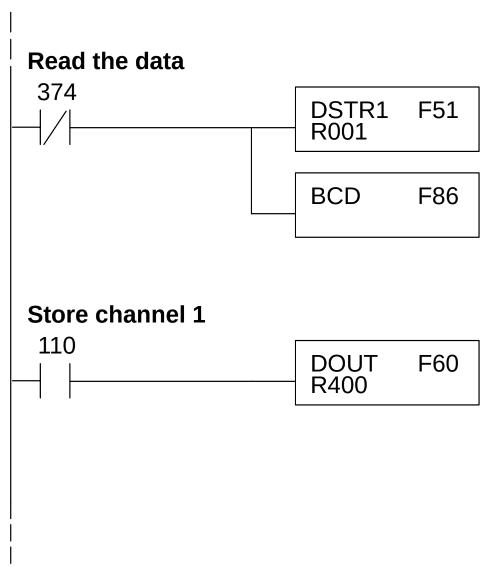
Identifying the Data Locations

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 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. If you choose another channel, you would have to add a rung (or rungs) that use the channel select bits to select the channel for scanning. You would also have to change the rung that stores the data.



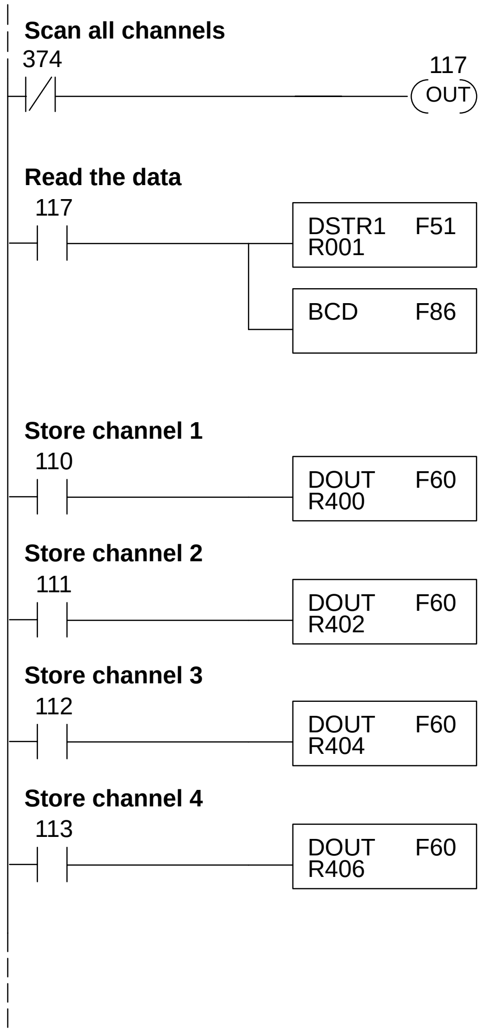
This rung loads the data into the accumulator on every scan. (You can use any permissive contact.)

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 channel selection inputs are used to let the CPU know which channel has been loaded into the accumulator. Channel 1 input has been used in the example, but you could easily use a different input for a different channel. By using these inputs 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 that is designed to read multiple channels of analog data into Register locations. This example reads one channel per scan. Once the data is in a Register, you can perform math on the data, compare the data against preset values, etc.



Turn on output 117, which instructs the module to scan all channels.

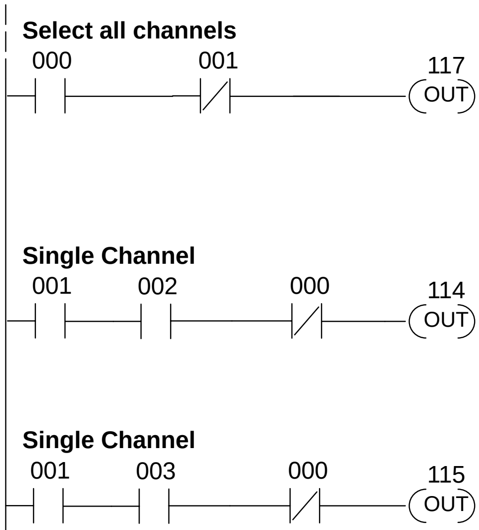
This rung loads the data into the accumulator. This rung executes for all channels.

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

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 that the DOUT instruction stores the data in two bytes. (Two bytes are required for four digit BCD numbers.)

Single or Multiple Channels

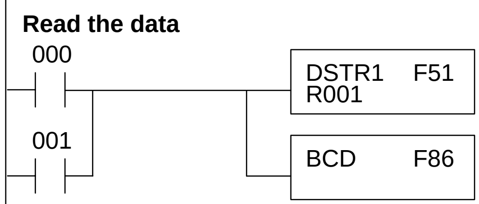
The following example shows how you can use the same program to read either all channels or a single channel of analog data into Register locations. Once the data is in a Register, you can perform math on the data, compare the data against preset values, etc.



Inputs 000 and 001 are used to select between single channel scanning and all channel scanning. These two points were arbitrarily chosen and could be any permissive contacts. When output 117 is on, all channels will be scanned.

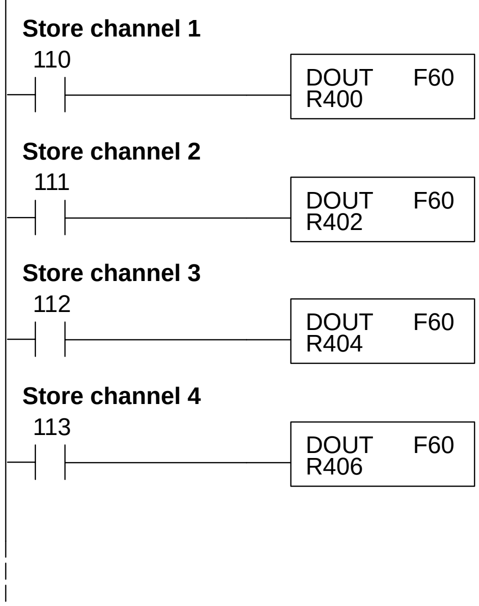
Input 001 selects single channel scan. Inputs 002 and 003 select which channel by turning on outputs 114 and 115 as necessary.

114	115	Channel
Off	Off	Ch. 1
On	Off	Ch. 2
Off	On	Ch. 3
On	On	Ch. 4



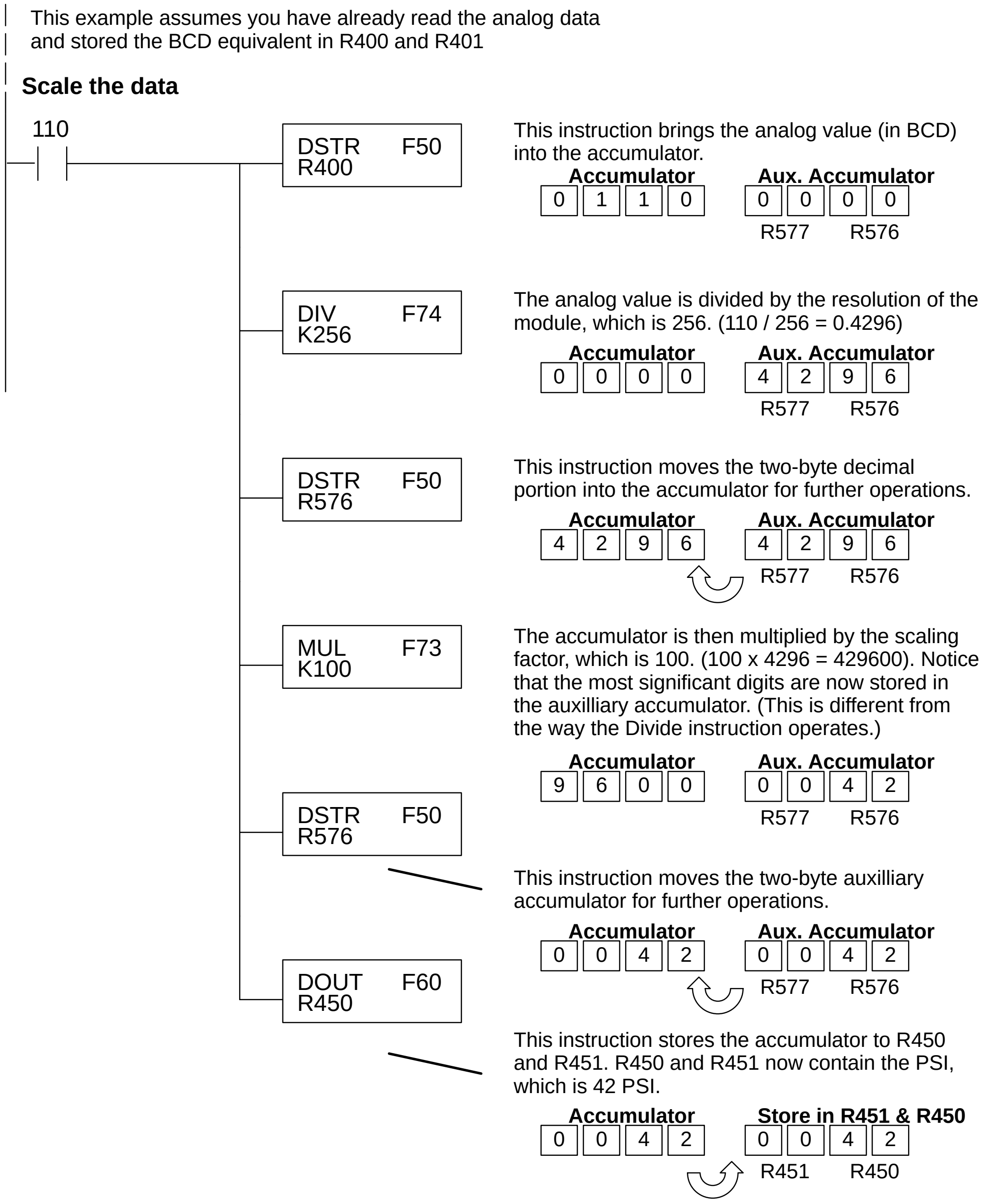
This rung loads the data into the accumulator. This rung executes for all channel scan or single channel scan.

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



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 that the DOUT instruction stores the data in two bytes. This is because two bytes are required to store the BCD number.

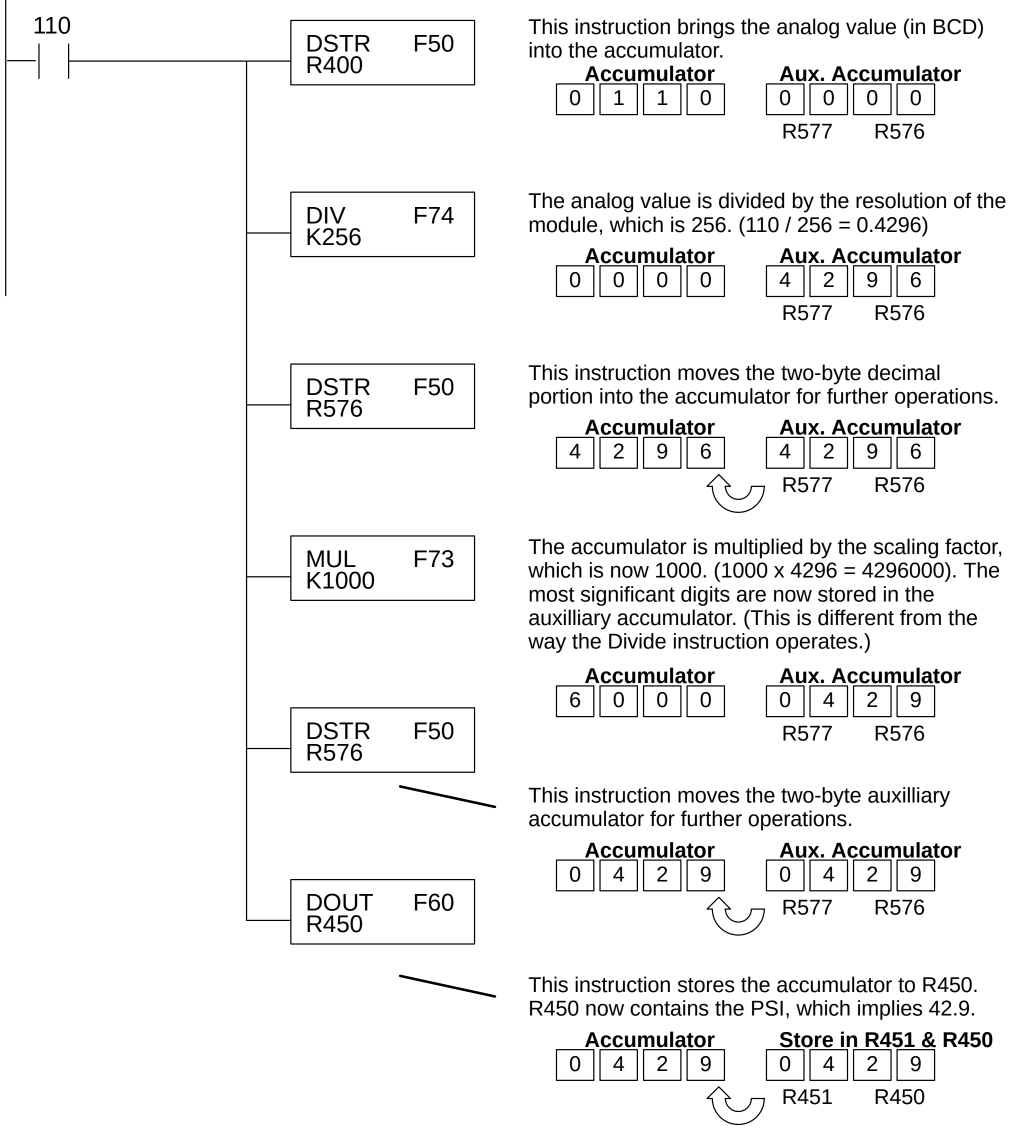
The following instructions are required to scale the data. We'll continue to use the 42.9 PSI example. In this example we're using channel 1. Input 110 is the active channel indicator for channel 1. Of course, if you were using a different channel, you would use the active channel indicator point that corresponds to the channel you were using.



You probably noticed that 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.

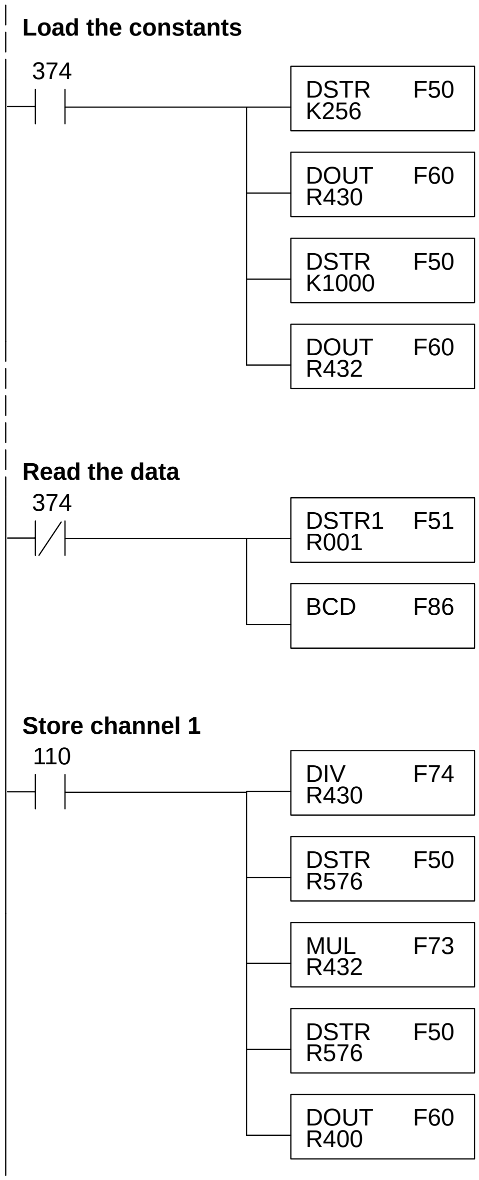
This example assumes you have already read the analog data and stored the BCD equivalent in R400 and R401

Scale the data



This example program shows how you can use the instructions to load the 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 256) 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 data into the accumulator on every scan. (You could use any permissive contact.)

The DL305 CPUs perform math operations in BCD. Since we will perform math on the data, the data must be converted from binary data to BCD.

The analog value is divided by the resolution of the module, 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, stored in R432.

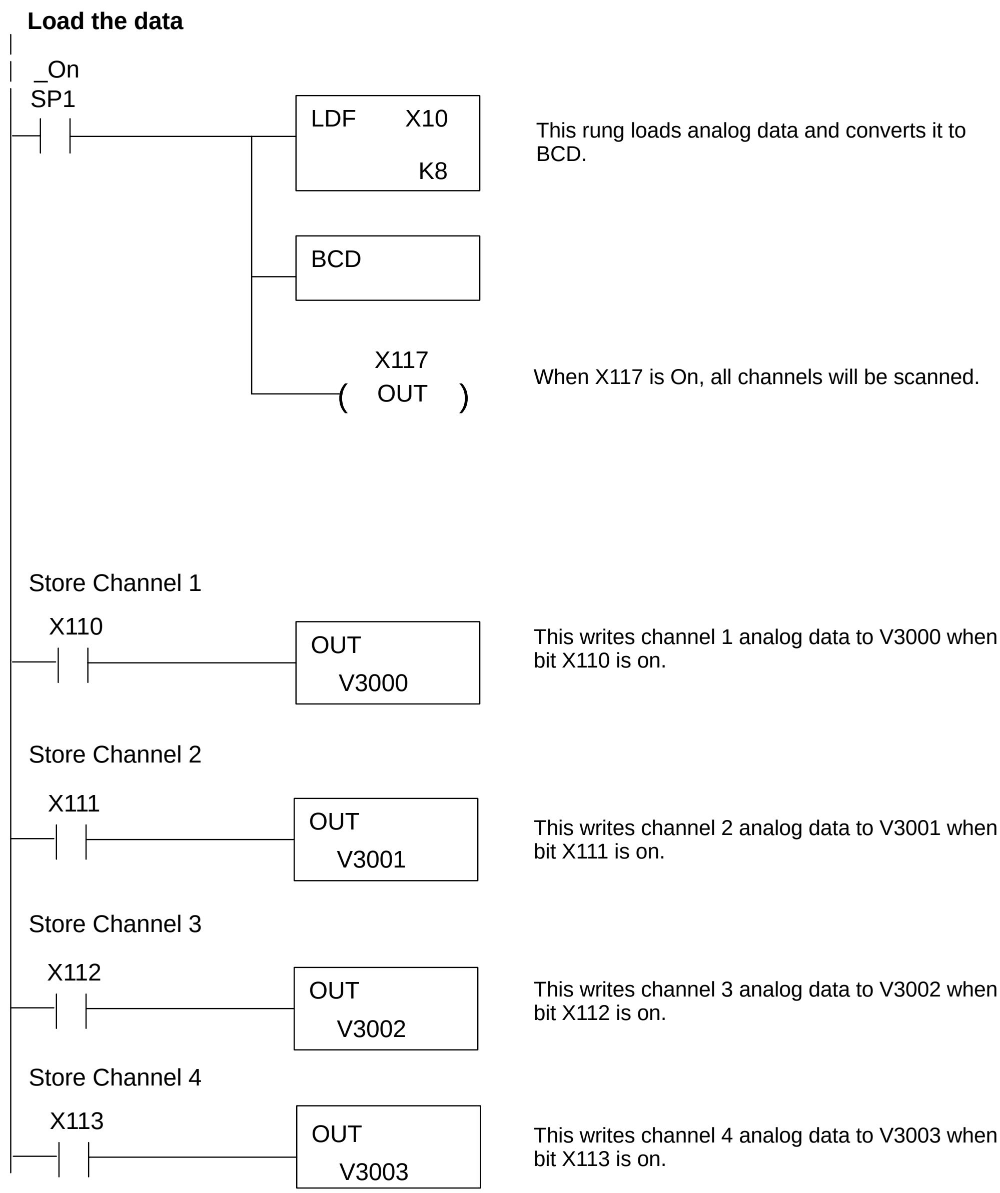
This instruction moves most significant digits (now stored in the auxilliary accumulator) into the regular accumulator for further operations.

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

Writing the Control Program (DL350)

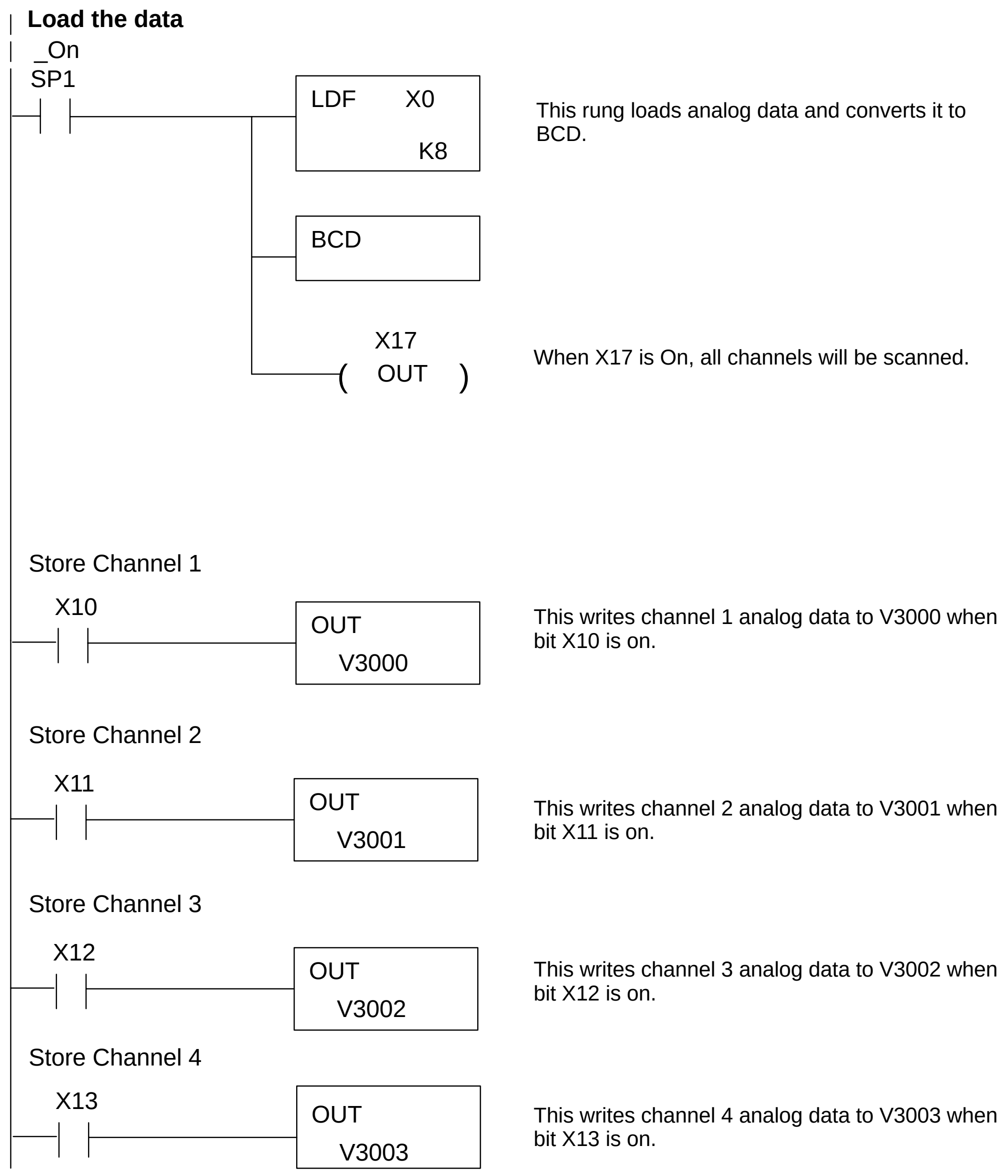
Multiplexing: DL350 with a Conventional DL305 Base

The example below shows how to read multiple channels on an D3–04AD Analog module in the 10–17/110–117 address slot. This module must be placed in a 16 bit slot in order to work.



Multiplexing: DL350 with a D3–xx–1 Base

The example below shows how to read multiple channels on an D3–04AD Analog module in the X0 address of the 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.



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.

Units = (A/255)*S

Units = value in Engineering Units
A = Analog value (0 – 255)
S = 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 110, which is slightly less than half scale. This should yield approximately 43 PSI.

Units = (A/255)*S

↓

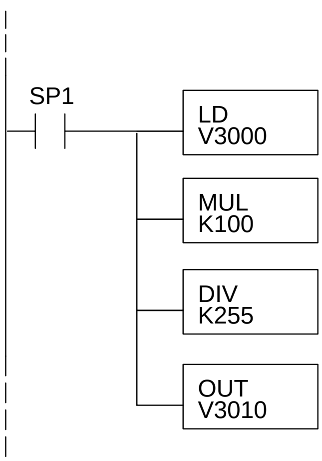
Units = (110/255)*100

↓

Units = 43

Here is how you would write the program to perform the engineering unit conversion. This example assumes you have the analog data in BCD format data loaded into V3000.

NOTE: This example uses SP1, which is always on. You could also use an X, C, etc. permissive contact.



When SP1 is on, load channel 1 data to the accumulator.

Multiply the accumulator by 100 (to start the conversion).

Divide the accumulator by 255.

Store the result in V3010.

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 ...
1 to 5V	A = (4D/255) + 1	D = (255/4)(A–1)
4 to 20mA	A = (16D/255) + 4	D = (255/16)(A–4)

For example, if you are using the 1 to 5V range and you have measured the signal at 3V, you would use the following formula to determine the digital value that should be stored in the register location that contains the data.

D = (255/4)(A–1)

D = (255/4)(3V–1)

D = (63.75) (2)

D = 127.5 (or 128)