DL305

Analog I/O Modules

Manual Number D3–ANLG-M

WARNING

Thank you for purchasing automation equipment from ***Automationdirect.com***™. We want your new ***Direct***LOGIC™ automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation are in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault–tolerant and are not designed, manufactured or intended for use or resale as on–line control equipment in hazardous environments requiring fail–safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). ***Automationdirect.com***™ *specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.*

For additional warranty and safety information, see the Terms and Conditions section of our Desk Reference. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770–844–4200.

This publication is based on information that was available at the time it was printed. At ***Automationdirect.com***™ we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. ***Automationdirect.com***™ disclaims any proprietary interest in the marks and names of others.

AVERTISSEMENT

Nous vous remercions d'avoir acheté l'équipement d'automatisation de **Automationdirect.com**™. Nous tenons à ce que votre nouvel équipement d'automatisation ***Direct***LOGIC™ fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux ("activités à risque élevé"). La société **Automationdirect.com**™ nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770–844–4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société **Automationdirect.com**™, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerce

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. **Automationdirect.com**™ nie tout intérêt dans les autres marques et désignations.

Manual Revisions

If you contact us in reference to this manual, be sure to include the revision number.

Title: DL305 Analog I/O Modules, 2nd Edition, Rev. D
Manual Number: D3–ANLG–M

Issue	Date	Description of Changes
Original	1/94	Original Issue
2nd Edition	3/96	Corrections
Rev. A	4/96	Minor corrections
Rev. B	6/98	Downsized to spiral Corrected sequencing examples
Rev. C	11/99	Added example programs for the D3–350 CPU.
3rd Edition	2/03	Added pointer method and additional D3–350 programming examples

Table of Contents

Chapter 1: Getting Started

Chapter 2: D3–04AD 4-Channel Analog Input

Chapter 3: F3–04ADS 4-Channel Isolated Analog Input

Chapter 4: F3–08AD–1 8-Channel Analog Input

Chapter 5: F3–16AD 16-Channel Analog Input

Chapter 6: D3–02DA 2–Channel Analog Output

Chapter 7: F3–04DA–1 4-Channel Analog Output

Chapter 8: F3–04DAS 4-Channel Isolated Analog Output

Chapter 9: F3–08THM–n 8-Channel Thermocouple Input

Chapter 10: F3–08TEMP 8-Channel Temperature Input

Appendix A: DL305 Data Types and Memory Map

1

Getting Started

In This Chapter. . . .

Introduction

Purpose of this manual

This manual will show you how to select and install analog input and analog output modules. It also shows several ways to use the analog data in your PLC program.

Who should read this manual

If you understand the DL305 oand DL350 instruction sets and system setup requirements, this manual will provide all the information you need to install and use the analog modules. This manual *is not* intended to be a tutorial on analog signal theory, but rather, a user reference manual for the DL305 Analog I/O modules.

Supplemental Manuals

If you have purchased operator interfaces or ***Direct***SOFT™, you will need to supplement this manual with the manuals that are written for these products.

Technical Support

We realize that even though we strive to be the best, the information may be arranged in such a way you cannot find what you are looking for. First, check these resources for help in locating the information:

- **Table of Contents** – chapter and section listing of contents, in the front of this manual
- **Quick Guide to Contents** – chapter summary listing on the next page
- **Appendices** – reference material for key topics, near the end of this manual
- **Index** – alphabetical listing of key words, at the end of this manual

You can also check our online resources for the latest product support information:

- **Internet** – the address of our Web site is http://www.plcdirect.com
- **Bulletin Board Service(BBS)** – call (770)–844–4209

If you still need assistance, please call us at 800–633–0405. Our technical support group is glad to work with you in answering your questions. They are available Monday through Friday from 9:00 A.M. to 6:00 P.M. Eastern Standard Time. If you have a comment or question about any of our products, services, or manuals, please fill out and return the ‘Suggestions’ card that was shipped with this manual.

Chapters

The main contents of this manual are organized into the following nine chapters:

1	**Getting Started**	introduces the various DL305 Analog modules. Also includes tips on getting started and how to design a successful system.
2	**D3–04AD**	explains the 4 channel analog input module. Provides ladder logic examples for all bases and CPUs.
3	**F3–04ADS**	explains the 4 channel isolated analog input module. Provides ladder logic examples for all bases and CPUs.
4	**F3–08AD**	explains the 8 channel analog input module. Provides ladder logic examples for all bases and CPUs.
5	**F3–16AD**	explains the 16 channel analog input module. Provides ladder logic examples for all bases and CPUs.
6	**D3–02DA**	explains the 2 channel analog output module. Provides ladder logic examples for all bases and CPUs.
7	**F3–04DA–1**	explains the 4 channel analog output module. Provides ladder logic examples for all bases and CPUs.
8	**F3–04DAS**	explains the 4 channel isolated analog output module. Provides ladder logic examples for all bases and CPUs.
9	**F3–08THM–n**	explains the 8 channel Thermocouple input module. Provides ladder logic examples for all bases and CPUs.
10	**F3–08TEMP**	explains the 8 channel temperature input module. Provides ladder logic examples for all bases and CPUs.

Appendices

Additional reference information on the DL305 analog modules is in the following five appendices:

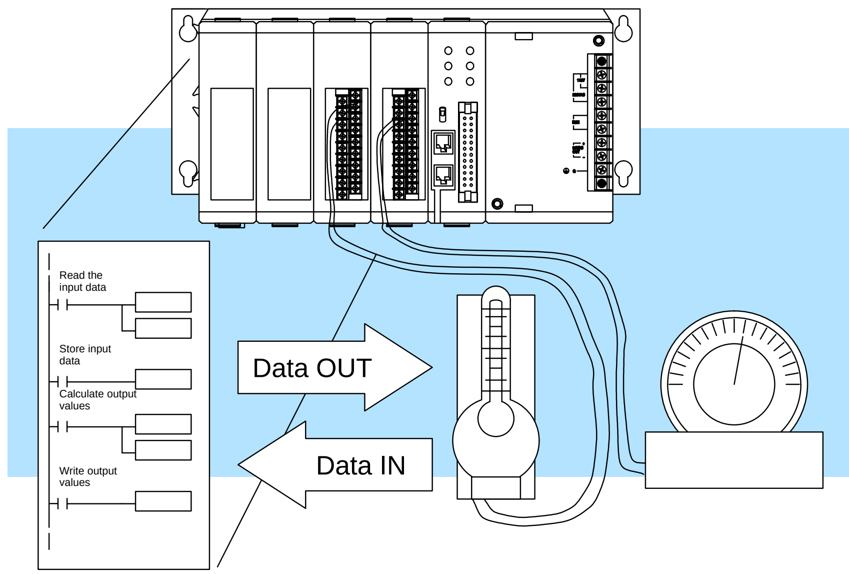
A B

Reference Appendices

- **A** – DL305C Data Types and Memory Map
- **B** – DL350 Data Types and Memory Map

DL305 Analog Components

There are a wide variety of Analog I/O modules available for use with the DL305 family of automation products. These modules are well suited for monitoring and controlling various types of analog signals such as pressure, temperature, etc. There are modules specifically designed for thermocouple and temperature input requirements. No complex programming or module setup software is required. Simply install the module, add a few lines to your RLL program, and you’re ready!



DL305 Analog I/O

The following is a list of the types of analog input and analog output modules that are available.

- D3–04AD — 4 channel input, 8-bit resolution
- F3–04ADS — 4 channel isolated input, 12-bit resolution
- F3–08AD — 8 channel input, 12-bit resolution
- F3–16AD — 16 channel input, 12-bit resolution
- D3–02DA — 2 channel output, 8-bit resolution
- F3–04DA–1 — 4 channel output, 12-bit resolution
- F3–04DAS — 4 channel isolated output, 12-bit resolution

Thermocouple Input

There is also an 8 channel thermocouple input module that converts type E, J, K, R, S, or T thermocouple signals into direct temperature readings. This module can also convert other types of low-level (millivolt range) signals into digital values. The part number for this module is F3–08THM–n, where n is the type of thermocouple. If you want a millivolt input version, simply replace n with a 1 (0–50 mV) or a 2 (0 – 100mV). All versions offer 12-bit resolution.

Temperature Input

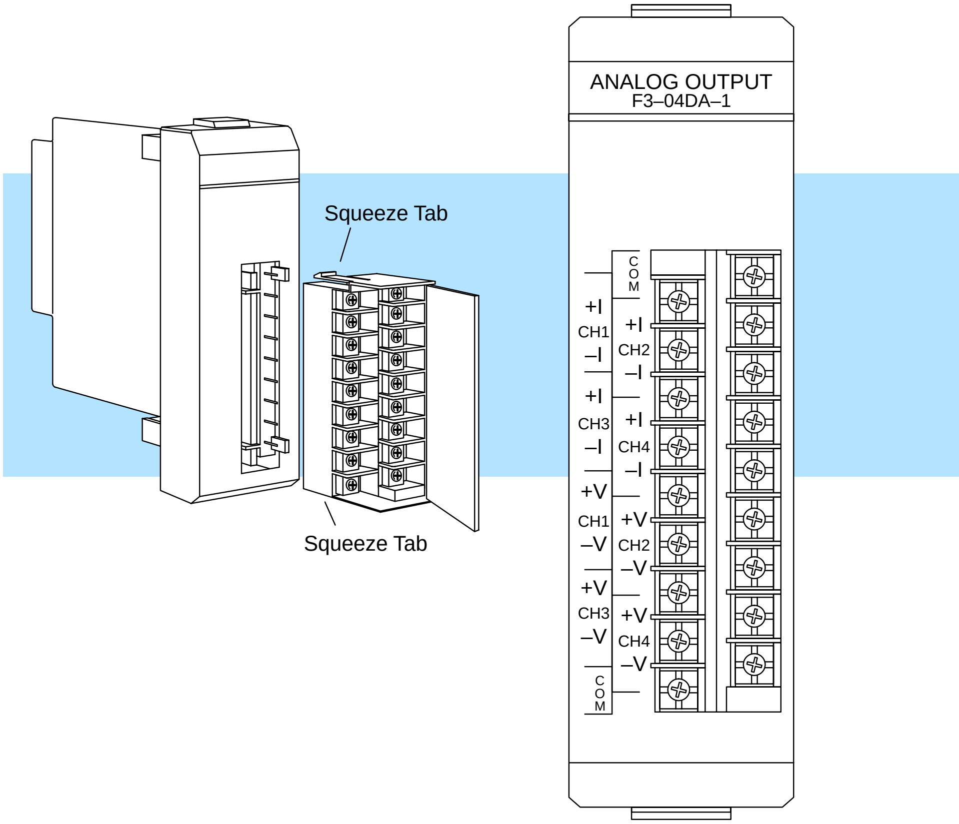
The Temperature Input module provides 8 channels for direct temperature measurement in either Celsius or Fahrenheit from –55° to 150° C. Order part number F3–08TEMP. This module offers 12-bit resolution.

Physical Characteristics

The DL305 Analog Modules provide many features that make the modules easier to use. For example, the terminal block can be removed making wiring a simple task. You can also use our **DIN***nector* product line to organize your wiring even further (see our catalog for details).

Some of the modules provide LEDs used to determine the signal level. Since there are not enough LEDs to show all of the channels at once, there is a small switch underneath the terminal cover that allows you to select the channel for monitoring. Not all of the modules have this feature.

Most of the modules also have jumpers that can be set to select between the various types of signals. Each chapter will show how to set these jumpers for the selections you need.



Selecting the Appropriate Module

The following tables provide a condensed version of the information you need to select the appropriate module. The most important thing is to simply determine the number of channels required and the signal ranges that must be supported. Once you've determined these parameters, look in the specific chapter for the selected module to determine the installation and operation requirements.

Analog Input

Specification	D3–04AD	F3–04ADS	F3–08AD	F3–16AD
Channels	4	4	8	16
Input Ranges	1 – 5V 4 – 20 mA	0 – 5V 1 – 5V 0 – 10V ±5V ±10V 0 – 20mA 4 – 20mA	4 – 20mA	0 – 5V 1 – 5V 0 – 10V ±5V ±10V 0 – 20mA 4 – 20mA[1]
Resolution	8 bit (1 in 256)	12 bit (1 in 4096)	12 bit (1 in 4096)	12 bit (1 in 4096)
Channel Isolation	Non-isolated (one common)	Isolated	Non-isolated (one common)	Non-isolated (one common)
Input Type	Differential	Differential	Single ended	Single ended
Maximum Inaccuracy at 77 °F (25 °C)	1%	±0.3%	0.35%	0.25% voltage 1.25% current
See Chapter . . .	2	3	4	5

[1] – resolution is reduced with 4–20 mA signals. You should use the F3–08AD if the primary application requires 4–20 mA signals.

Analog Output

Specification	D3–02DA	FACTS F3–04DA–1	FACTS F3–04DAS
Channels	2	4	4
Output Ranges	1 – 10VDC 4 – 20 mA	0 – 5V 0 –10V 4 – 12mA 4 – 20mA	0 – 5V 0 – 10V ± 5V ± 10V 4 – 20mA
Resolution	8 bit (1 in 256)	12 bit (1 in 4096)	12 bit (1 in 4096)
Channel Isolation	Non-isolated (one common)	Non-isolated (one common)	Isolated
Output Type	Single ended	Single ended	Differential
Maximum Inaccuracy at 77 °F (25 °C)	± 0.4%	± 0.2% voltage ± 0.6% current	± 0.8%
See Chapter	6	7	8

Special Input

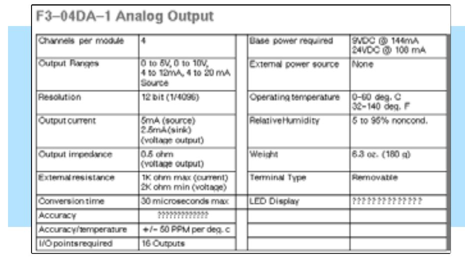
Specification	F3–08TEMP	FACTS F4–04DA
Channels	8, Temperature Input	8, Thermocouple Input
Input Ranges	0 – 1mA AD590 input types	E: –270/1000 °C, –450/1832 °F J: –210/760 °C, –350/1390 °F K: –270/1370 °C, –450/2500 °F R: 0/1768 °C, –32/3214 °F S: 0/1768 °C, –32/3214 °F T: –270/400 °C, –450/752 °F 50mV: 0 – 50 mV 100mV: 0–100 mV
Resolution	12 bit (1 in 4096)	12 bit (1 in 4096)
Channel Isolation	Non-isolated	Non-isolated
Input Type	Single ended	Differential
Maximum Inaccuracy at 77 °F (25 °C)	0.25%	0.35%
See Chapter	10	9

Getting Started

Analog Made Easy – Four Simple Steps

Once you've selected the appropriate module, use the chapter that describes the module and complete the following steps.

STEP 1. Take a minute to review the detailed specifications to make sure the module meets your application requirements.

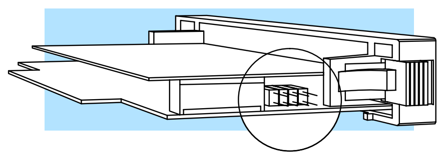

F3–04DA–1 Analog Output

Channels per module	4	Base power required	9VDC @ 144mA 24VDC @ 108 mA
Output Ranges	0 to 5V, 0 to 10V, 4 to 12mA, 4 to 20 mA Source	External power source	None
Resolution	12 bit (1/4096)	Operating temperature	0–60 deg. C 32–140 deg. F
Output current	5mA (source) 2.5mA (sink) (voltage output)	Relative Humidity	5 to 95% noncond.
Output impedance	0.5 ohm (voltage output)	Weight	6.3 oz. (180 g)
External resistance	1K ohm max (current) 2K ohm min (voltage)	Terminal Type	Removable
Conversion time	30 microseconds max	LED Display	??????????????
Accuracy	?????????????		
Accuracy/temperature	+/– 50 PPM per deg. c		
I/O points required	16 Outputs		

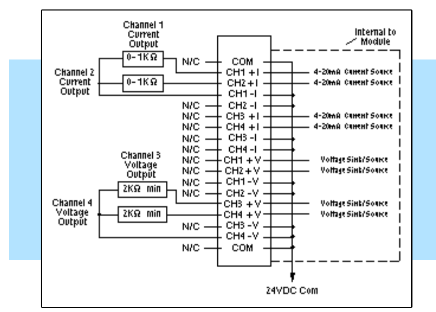
STEP 2. Set the module switches and/or jumpers to select:

- number of channels
- the operating ranges (voltage or current)

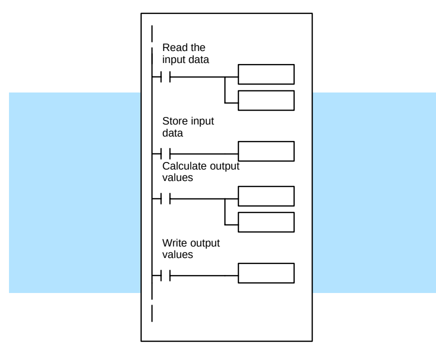
Note, some of the modules may not have switches.



STEP 3. Connect the field wiring to the module connector.



STEP 4. Review the module operating characteristics and write the control program.



Analog Input Terminology

We use several different terms throughout the rest of this manual. You don't have to be an expert on analog terms to use the products, but it may help make it easier to select the appropriate modules if you take a few minutes to review these definitions.

Channels per Module The number of analog channels or points available in the module to connect to field devices.

Input Ranges The input ranges in voltage and/or current that the module will operate properly within.

Resolution The number of binary weighted bits available on the digital side of the module for use in converting the analog value to a digital value.

Input Type Specifies if the module accepts single ended, bipolar or differential input signals.

Input Impedance The input impedance of the module using a voltage or current input signal.

Conversion Method The method the module uses to convert the analog signal to a digital value.

Conversion Time The amount of time required to complete the analog to digital conversion.

Linearity Error and Total Tolerance (Relative Accuracy) The linearity and accuracy of the digital representation over the entire input range.

Accuracy vs. Temperature The effect of temperature on the accuracy of the module.

LED Display LED indicators on the module

I/O Points Required The number of I/O points the CPU must dedicate to the module.

External Power Source Some modules require a separate 24VDC power source. The 24VDC output supply at the local or expansion base can be used as long as you do not exceed the current rating.

Base Power Required The amount of base current required by the module. Use this value in your power budget calculations.

Operating Temperature The minimum and maximum temperatures the module will operate.

Relative Humidity The minimum and maximum humidity the module will operate.

Terminal Type Indicates whether the terminal type is a removable or non-removable connector or a terminal.

Weight The weight of the module.

Analog Output Module Terminology

Channels per Module The number of analog channels or points available in the module to connect to field devices.

Output Ranges The output ranges in voltage and/or current modes the module will operate properly within.

Resolution The number of binary weighted bits available on the digital side of the module for use in converting the digital value to a analog signal.

Output Current The maximum current the module will drive using a voltage output signal.

Output Impedance The output impedance of the module using a voltage output signal.

Load Impedance The minimum and maximum resistance the module can drive using a current output signal.

Conversion Time The amount of time required to complete the digital to analog conversion.

Accuracy The linearity and calibrated accuracy of the digital representation over the entire output range.

Accuracy vs. Temperature The effect of temperature on the accuracy of the module.

LED Display LED indicators on the module

External Power Source Some modules require a separate 24VDC power source. The 24VDC output supply at the local or expansion base can be used as long as you do not exceed the current rating.

Base Power Required The amount of base current required by the module. Use this value in your power budget calculations.

Operating Temperature The minimum and maximum temperatures the module will operate.

Relative Humidity The minimum and maximum humidity the module will operate.

Terminal Type Indicates whether the terminal type is a removable or non-removable connector or a terminal.

Weight The weight of the module.

I/O Points Required The number of I/O points the CPU must dedicate to the module.

2

D3–04AD 4-Channel Analog Input

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

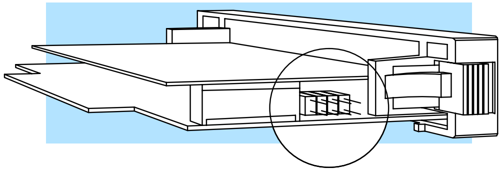
The D3–04AD Analog Input appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

- 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.

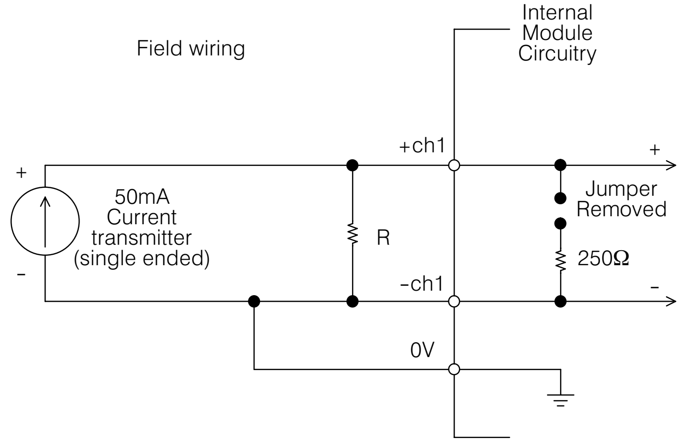
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the source.
- Don’t run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.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

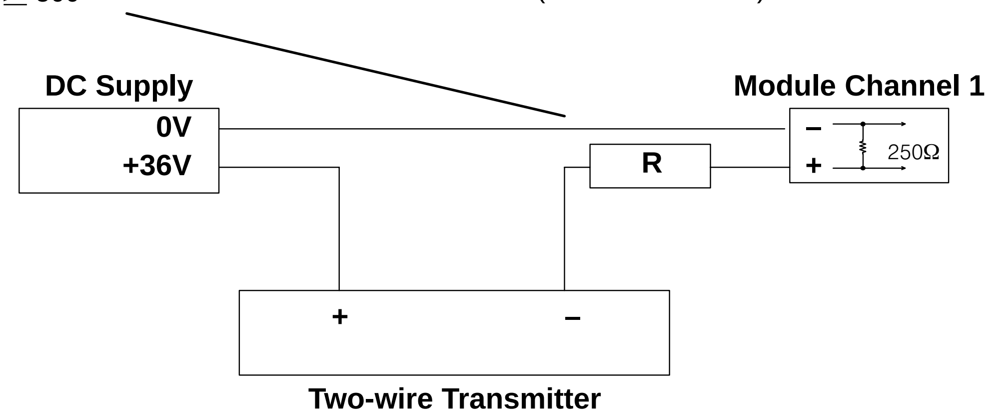
Standard 4 to 20 mA transmitters and transducers can operate from a wide variety of power supplies. Not all transmitters are alike and the manufacturers often specify a minimum loop or load resistance that must be used with the transmitter.

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

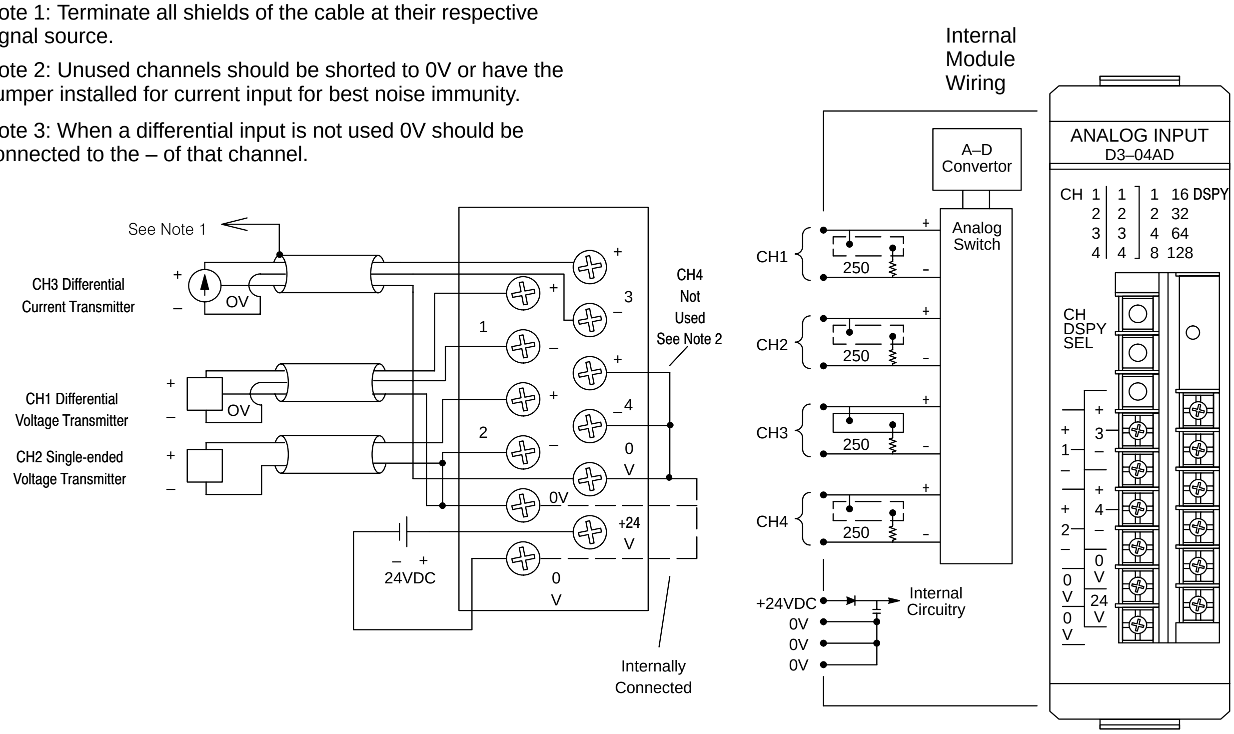
The D3–04AD module has a removable connector to make wiring easier. Simply squeeze the tabs on the top and bottom and gently pull the connector from the module.

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.



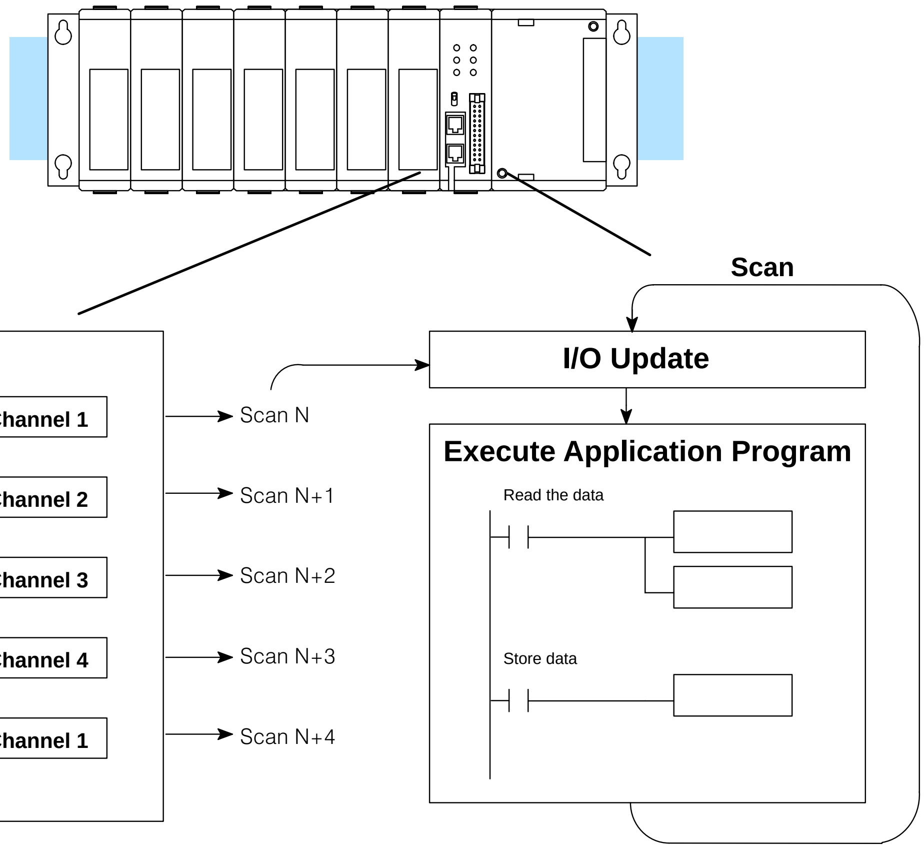
Module Operation

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.



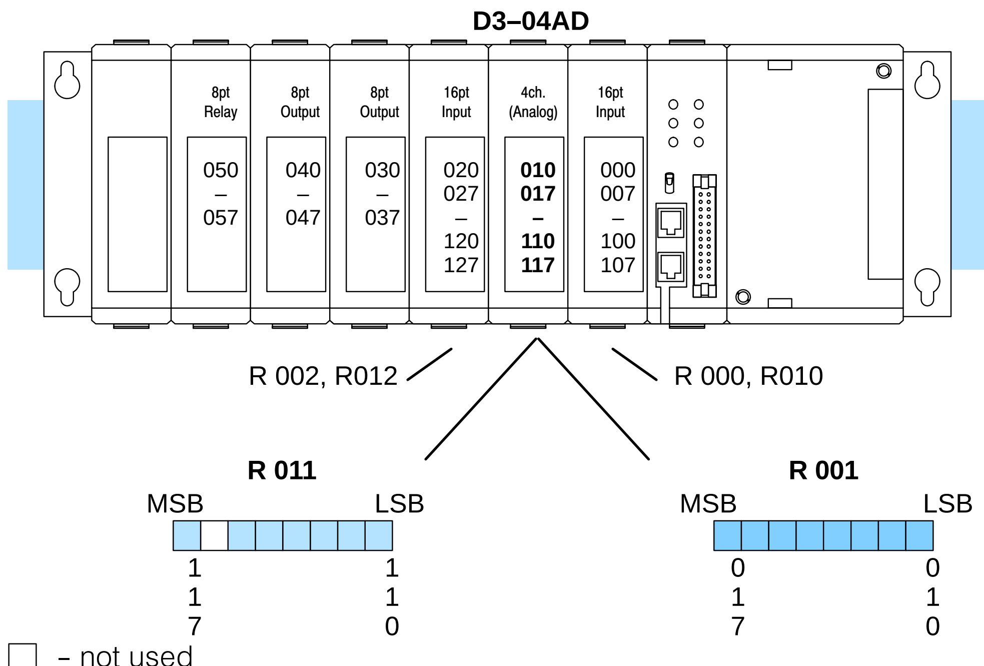
Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 8-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete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

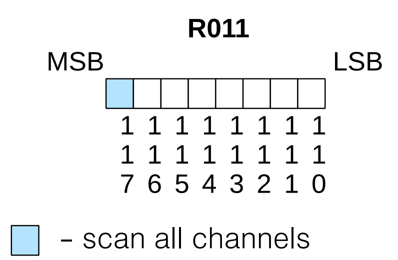


Within these two register locations, the individual bits represent specific information about the analog signal.

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.



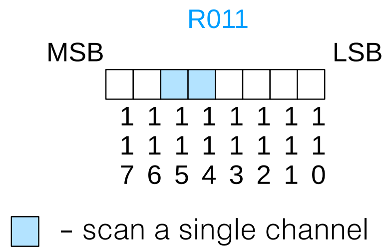
Scan	Out 117	Channel Input
N	Off	None
N+1	On	1
N+2	On	2
N+3	On	3
N+4	On	4
N+5	On	1
N+6	Off	None
N+7	Off	None

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.)

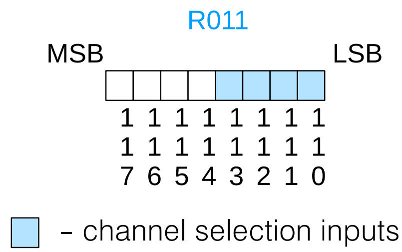
Out 114	Out 115	Channel
Off	Off	1
On	Off	2
Off	On	3
On	On	4



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.

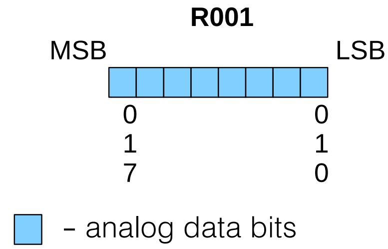
Input	Active Channel
110	1
111	2
112	3
113	4



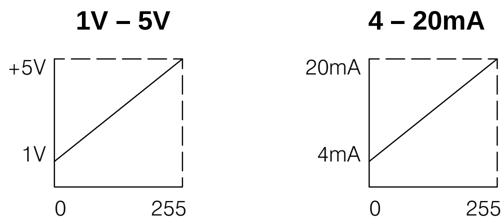
Analog Data Bits

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

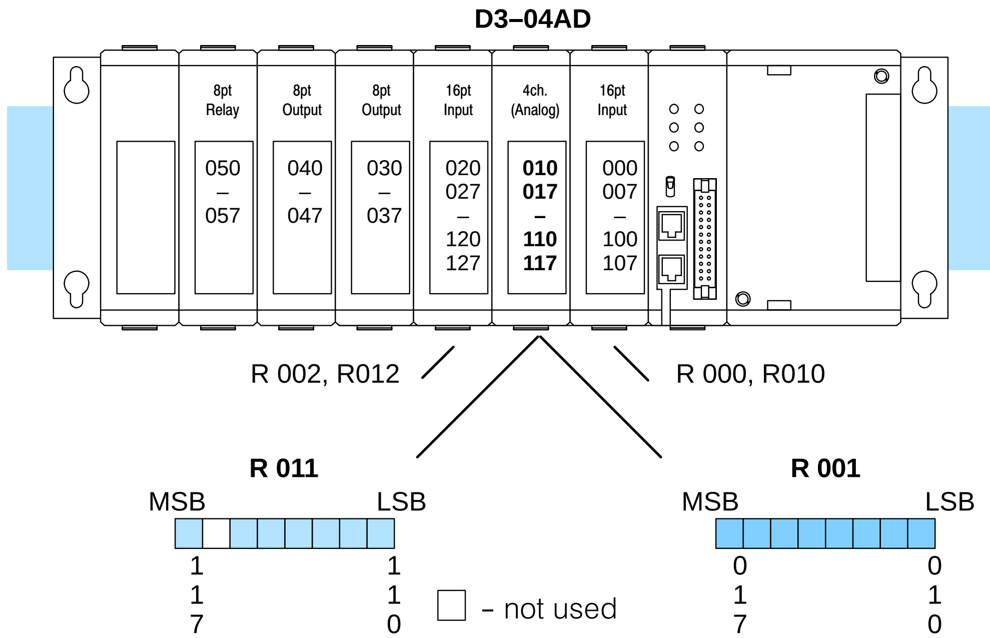
Range	Highest Signal	Lowest Signal	Smallest Change
1 to 5V	5V	1V	15.6 mV
4 to 20mA	20mA	4mA	62.7 µA

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)

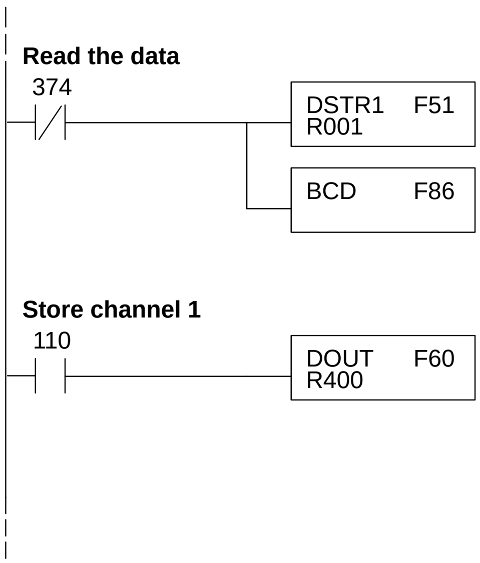
Identifying the Data Locations

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.



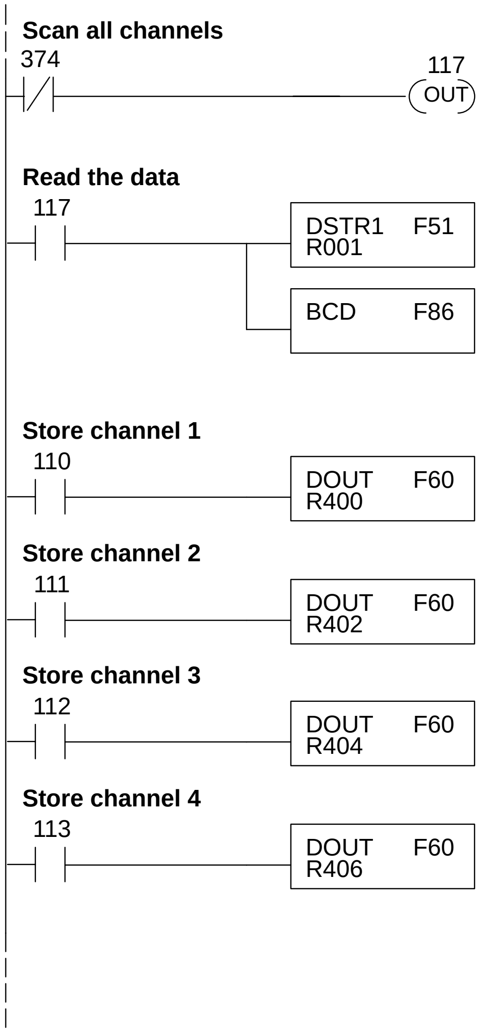
This rung loads the data into the accumulator on every scan. (You can use any permissive contact.)

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.

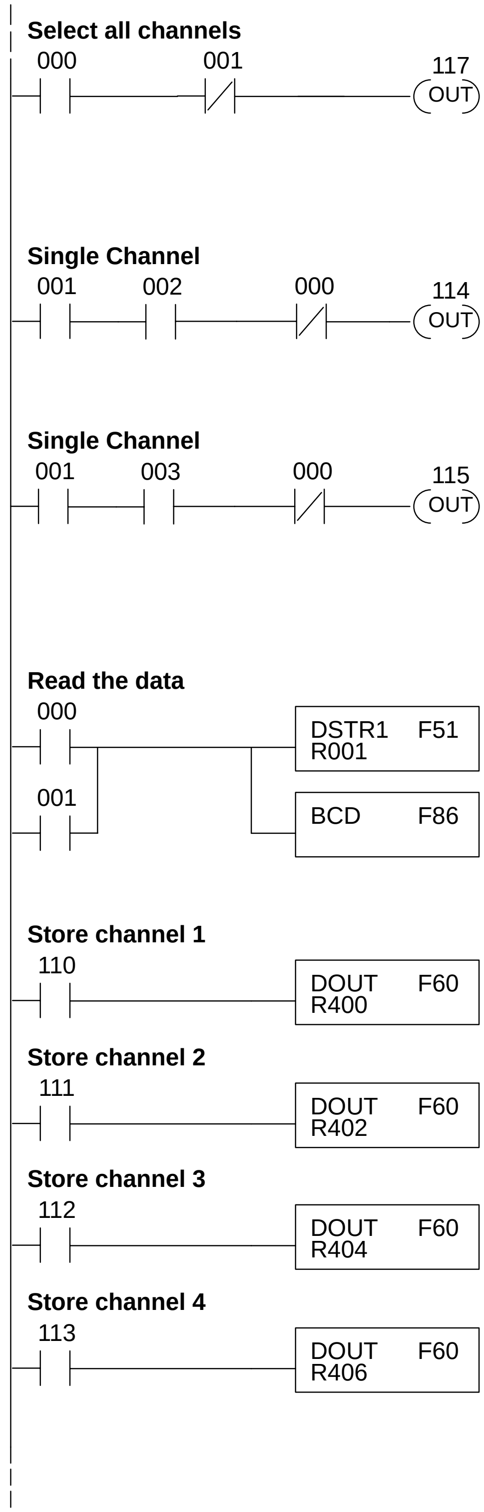
This rung loads the data into the accumulator. This rung executes for all channels.

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.

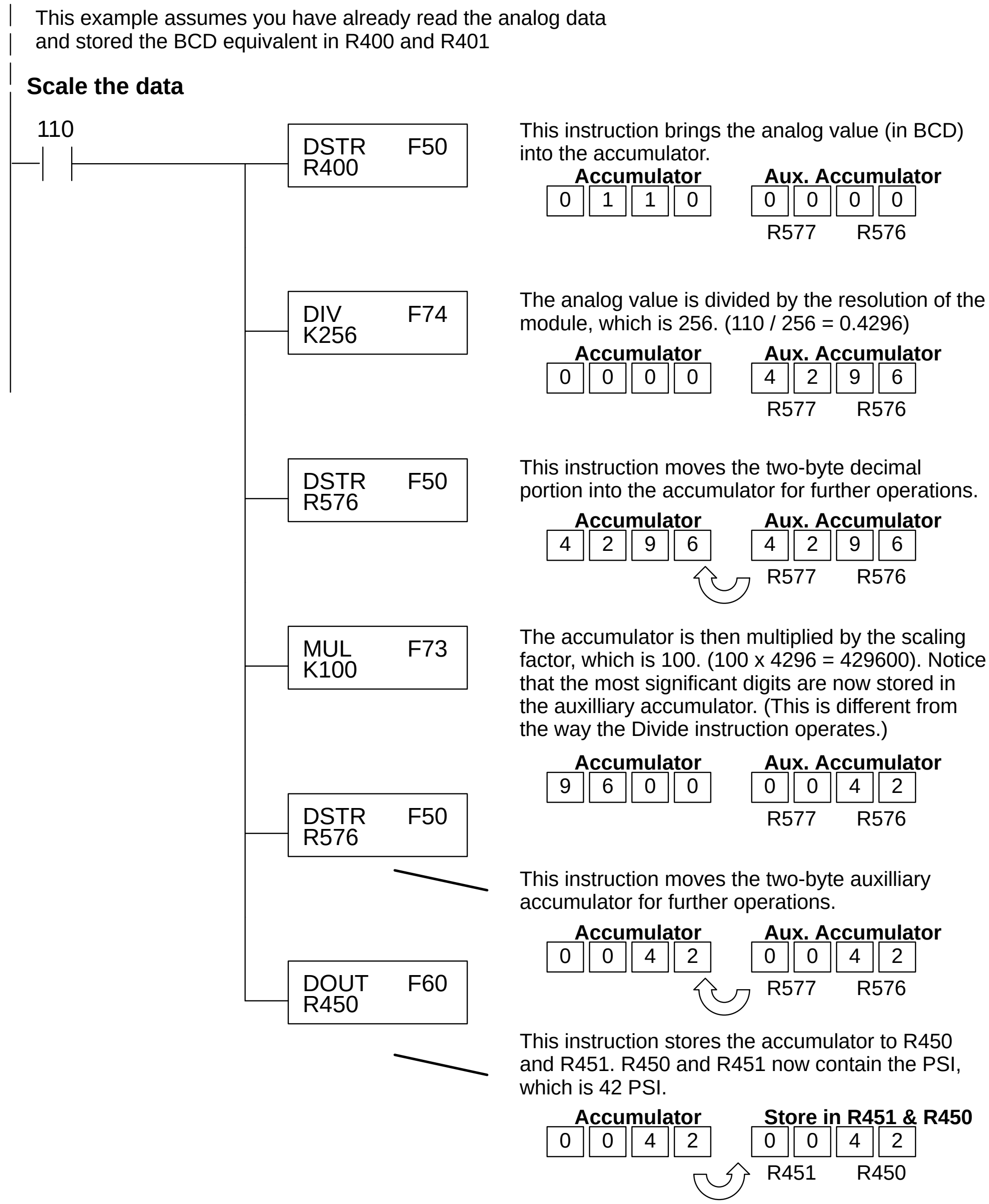
114	115	Channel
Off	Off	Ch. 1
On	Off	Ch. 2
Off	On	Ch. 3
On	On	Ch. 4

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.

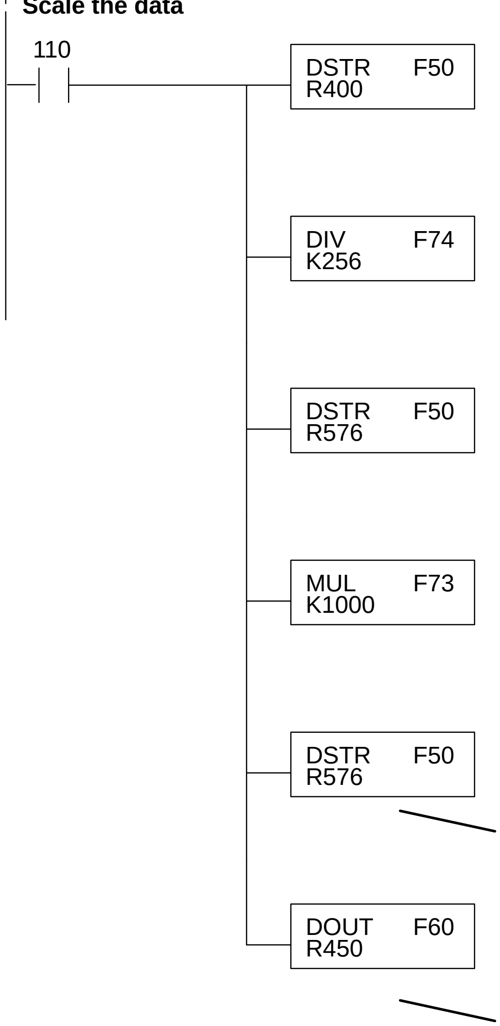
The following instructions are required to scale the data. We'll continue to use the 42.9 PSI example. In this example we're using channel 1. Input 110 is the active channel indicator for channel 1. Of course, if you were using a different channel, you would use the active channel indicator point that 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 instruction brings the analog value (in BCD) into the accumulator.

Accumulator				Aux. Accumulator			
0	1	1	0	0	0	0	0
				R577		R576	

The analog value is divided by the resolution of the module, which is 256. (110 / 256 = 0.4296)

Accumulator				Aux. Accumulator			
0	0	0	0	4	2	9	6
				R577		R576	

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator			
4	2	9	6	4	2	9	6
				R577		R576	

The accumulator is multiplied by the scaling factor, which is now 1000. (1000 x 4296 = 4296000). The most significant digits are now stored in the auxilliary accumulator. (This is different from the way the Divide instruction operates.)

Accumulator				Aux. Accumulator			
6	0	0	0	0	4	2	9
				R577		R576	

This instruction moves the two-byte auxilliary accumulator for further operations.

Accumulator				Aux. Accumulator			
0	4	2	9	0	4	2	9
				R577		R576	

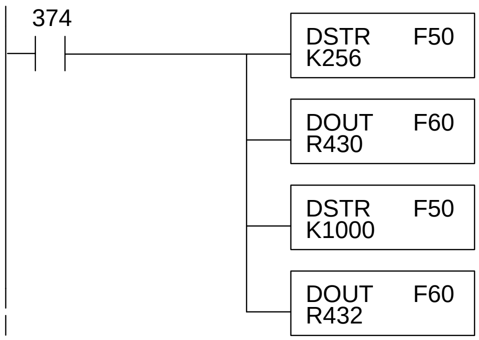
This instruction stores the accumulator to R450. R450 now contains the PSI, which implies 42.9.

Accumulator				Store in R451 & R450			
0	4	2	9	0	4	2	9
				R451		R450	

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.

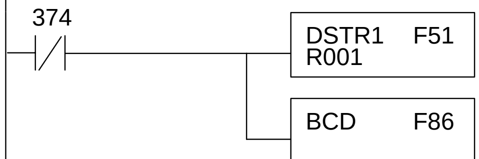
Load the constants



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.

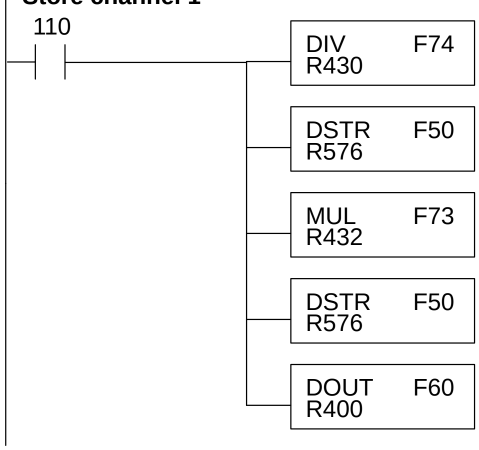
Read the data



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.

Store channel 1



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.

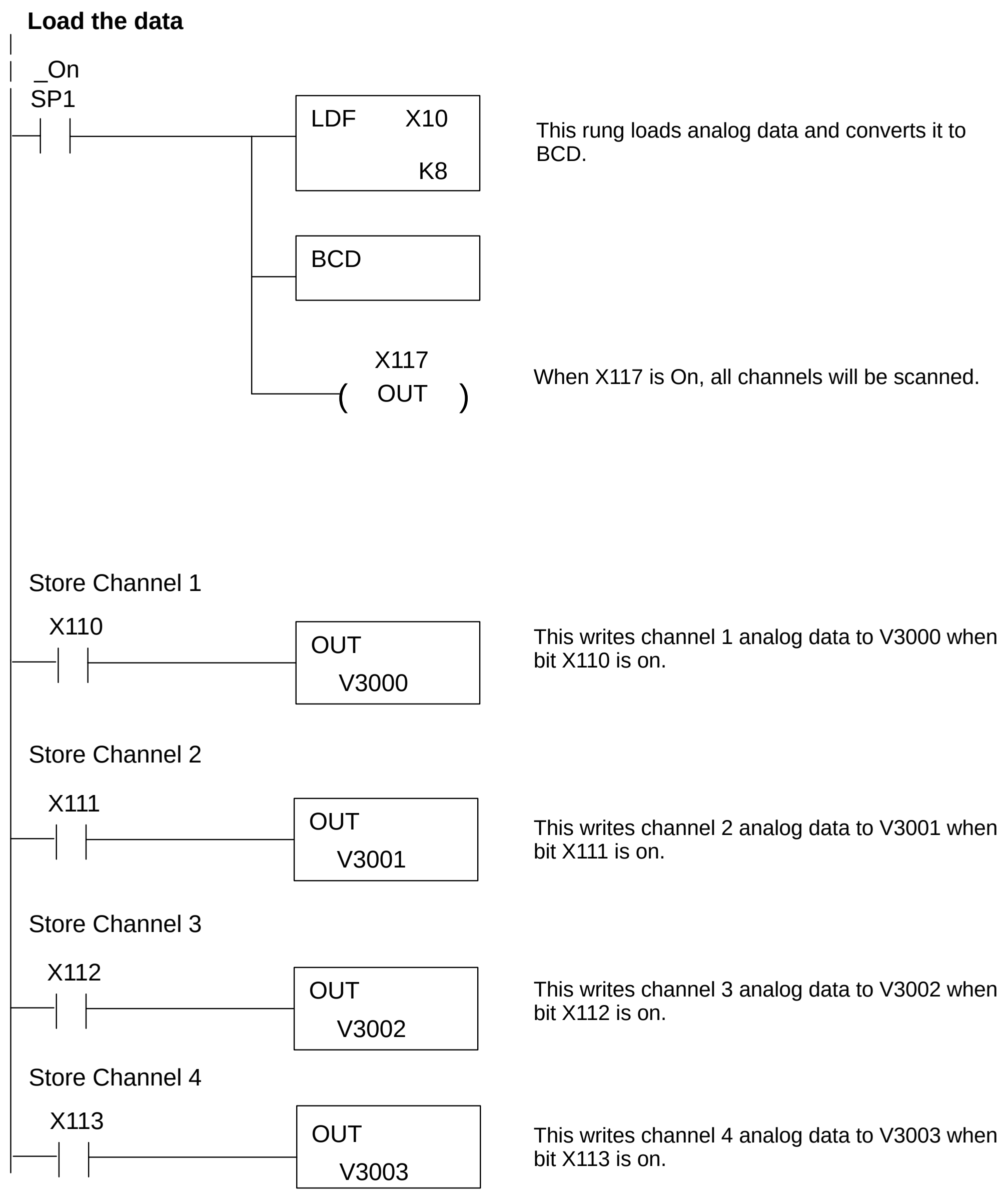
This instruction moves most significant digits (now stored in the auxilliary accumulator) into the regular accumulator for further operations.

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

Writing the Control Program (DL350)

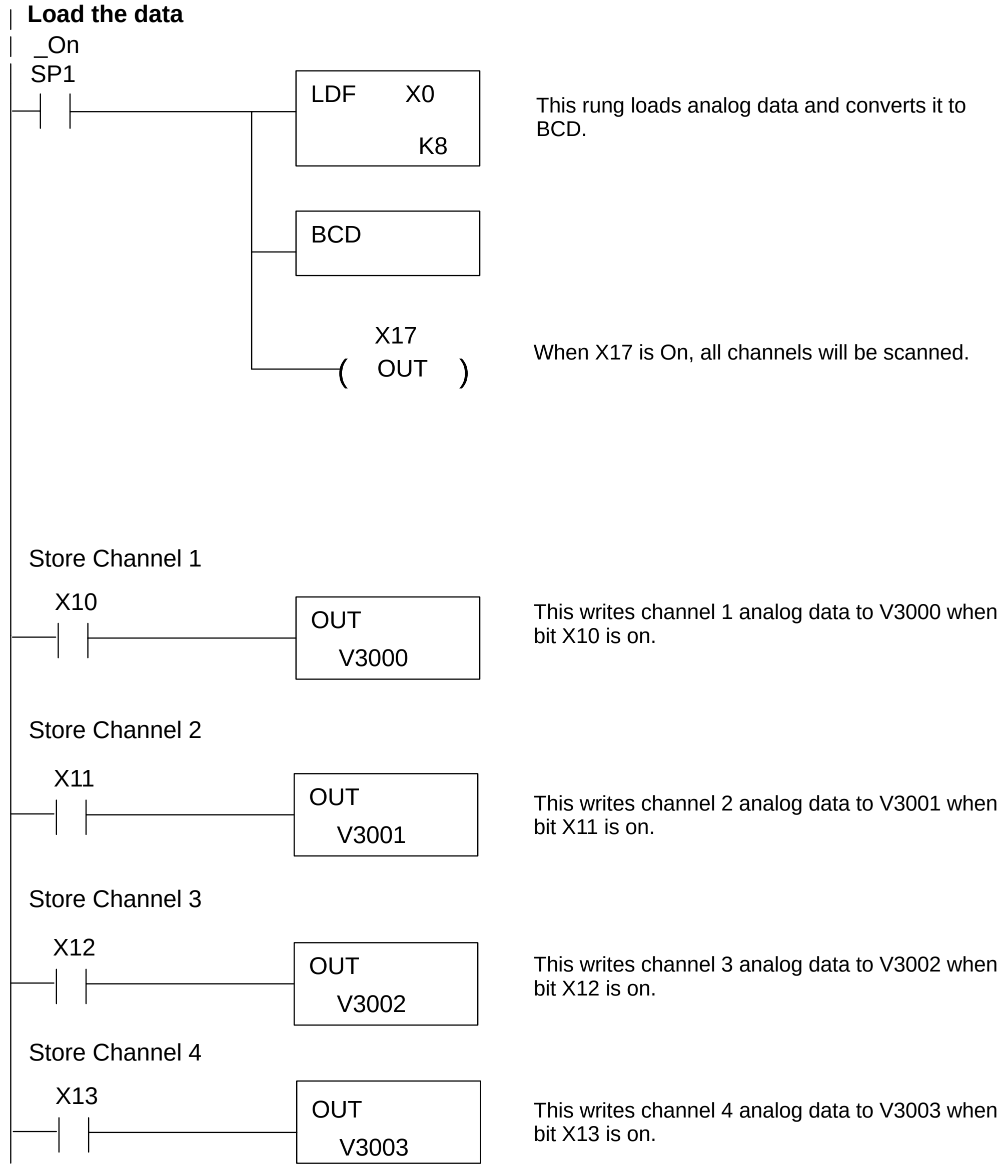
Multiplexing: DL350 with a Conventional DL305 Base

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

↓

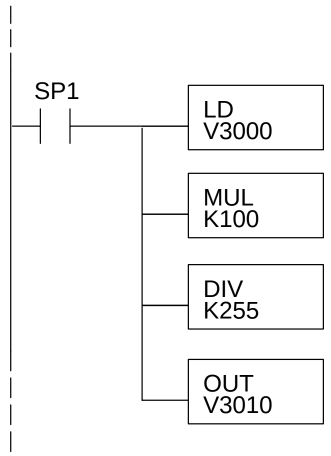
Units = (110/255)*100

↓

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)

3

F3–04ADS 4-Channel Isolated Analog Input

In This Chapter. . . .

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

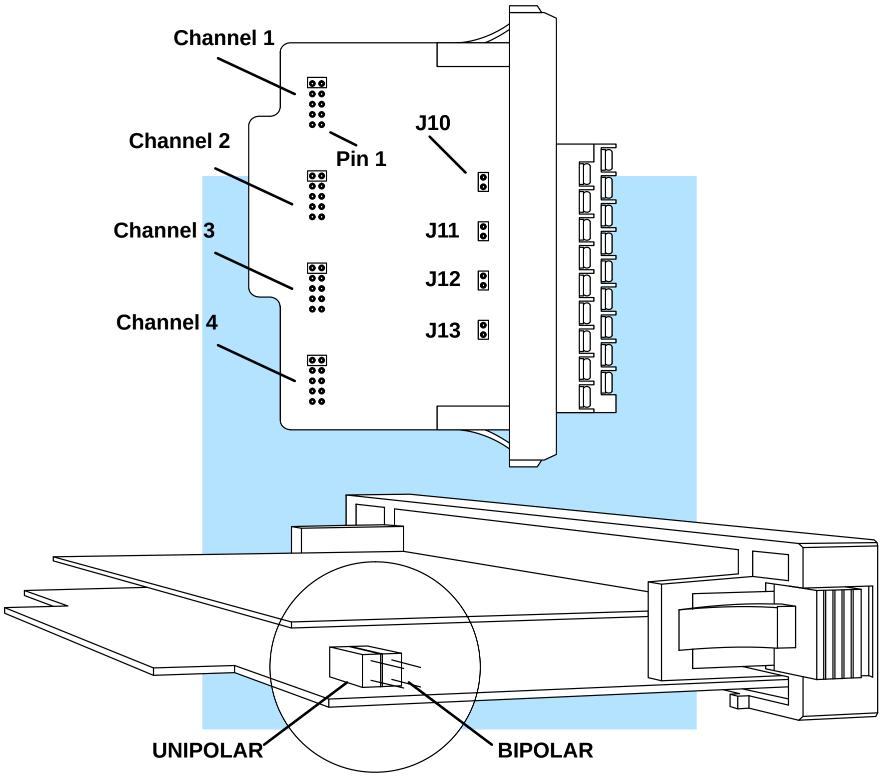
The F3–04ADS Analog Input appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

- 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.



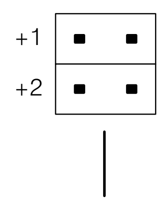
Selecting the Number of Channels

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.



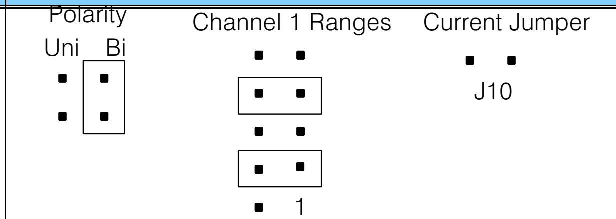
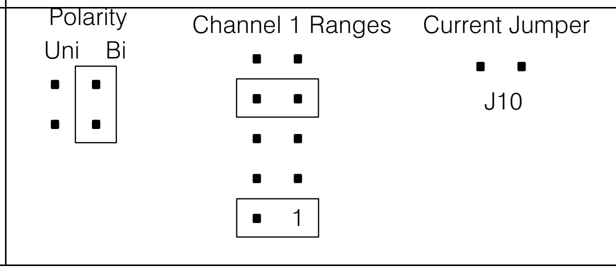
Jumpers installed as shown selects 4-channel operation

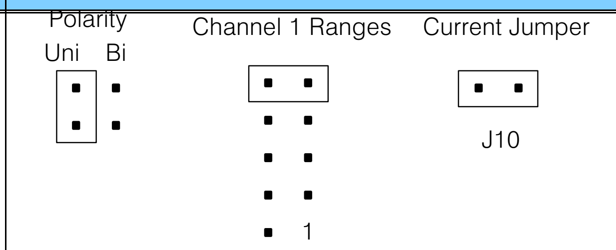
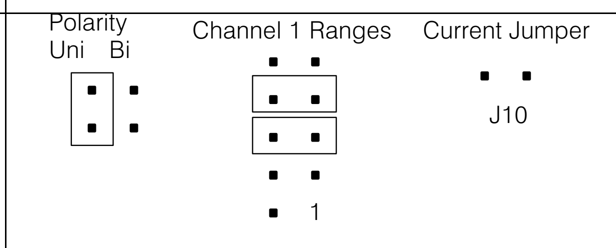
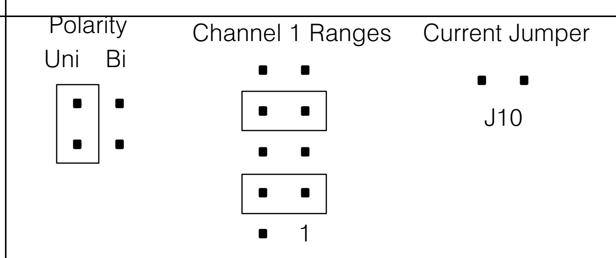
Channel	+1	+2
1	No	No
1, 2,	Yes	No
1, 2, 3	No	Yes
1, 2, 3, 4	Yes	Yes

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.

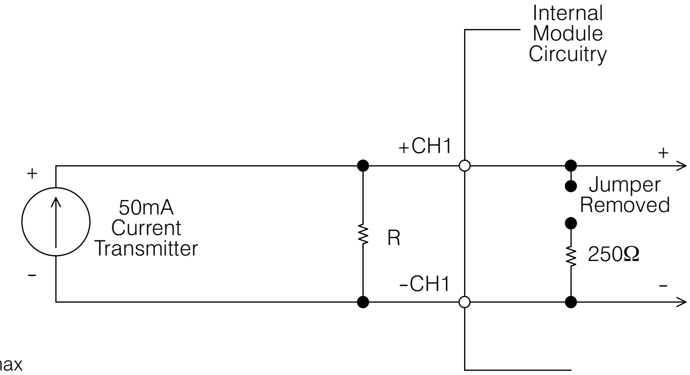
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the source.
- Do not run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.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

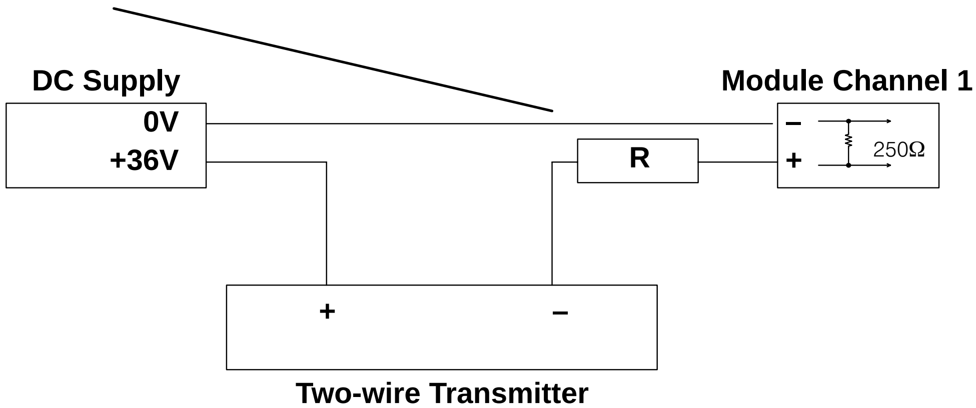
Consider the following example for a transmitter being operated from a 36 VDC supply with a recommended load resistance of 750 ohms. Since the module has a 250 ohm resistor, you need to add an additional resistor.

$R = Tr - Mr$

$R = 750 - 250$

$R \geq 500$

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

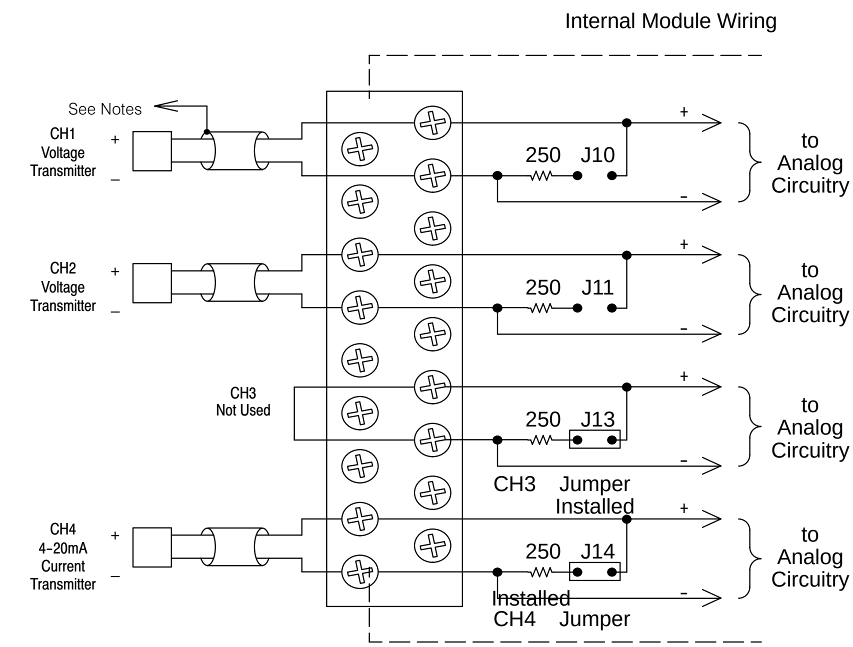
Note 1: Connect unused voltage or current inputs to 0VDC at terminal block or leave current jumper installed (see Channel 3).

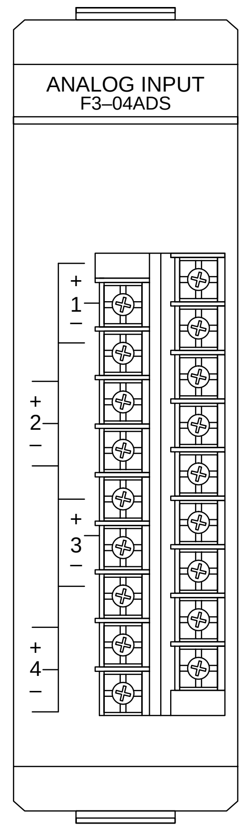
Note 2: A Series 217, 0.032A, Fast-acting fuse is recommended for 4–20mA current loops.

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.





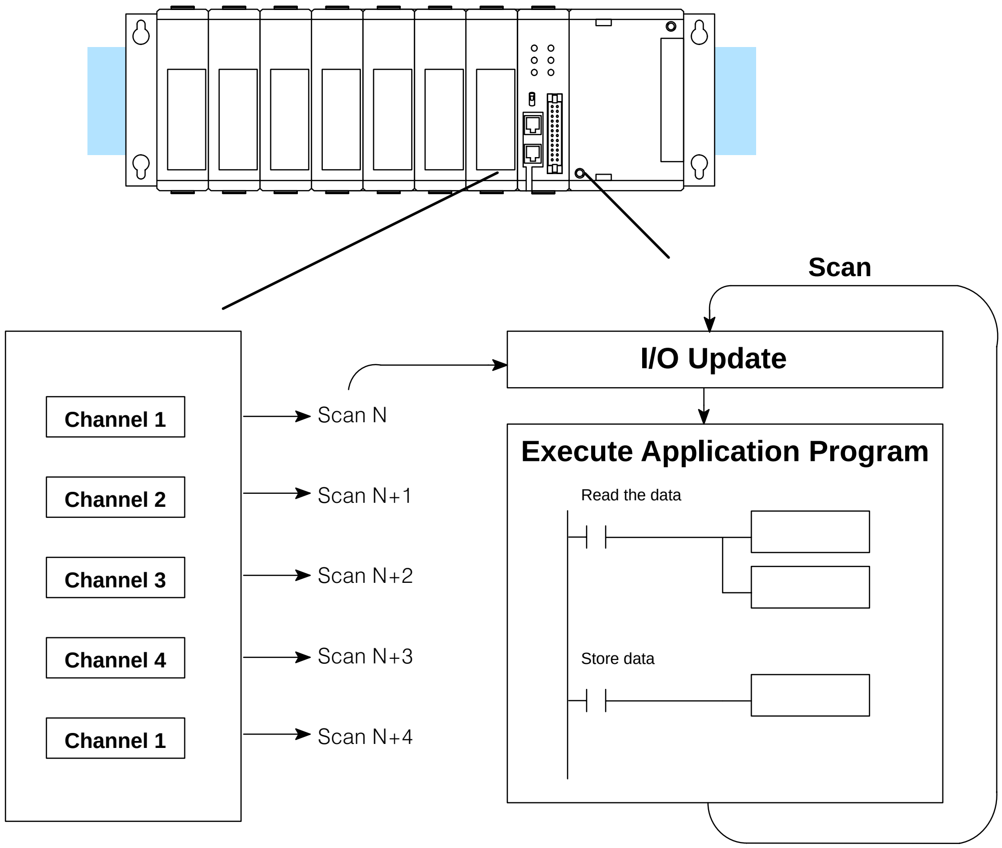
Module Operation

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.

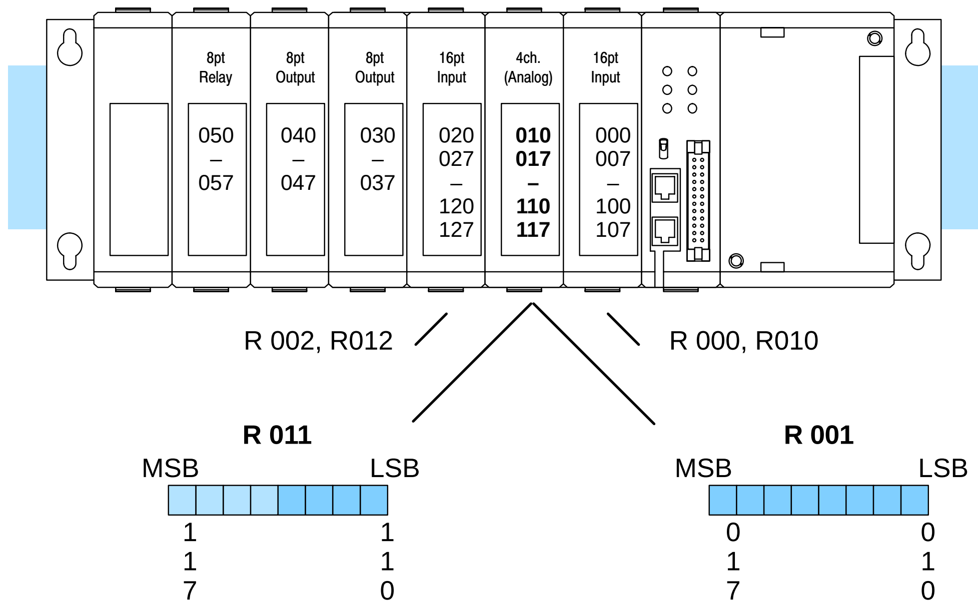
Understanding the I/O Assignments

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.

F3–04ADS

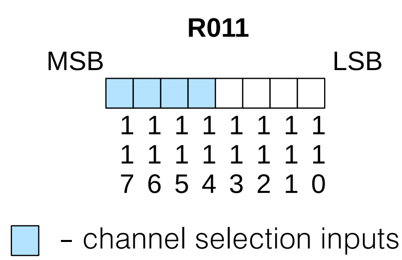


Within these two register locations, the individual bits represent specific information about the analog signal.

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.

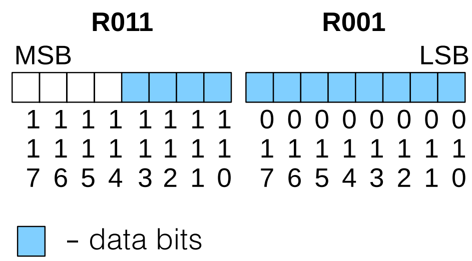
Input	Active Channel
114	1
115	2
116	3
117	4



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



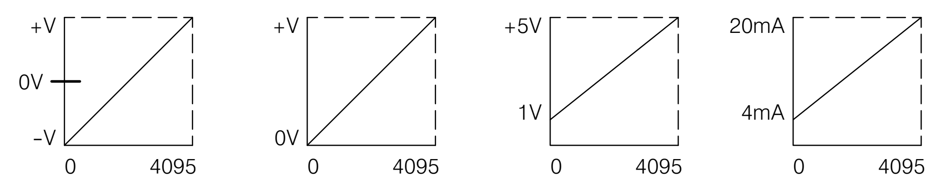
Since the module has 12-bit resolution, the analog signal is converted into 4096 “pieces” ranging from 0 – 4095 (2^{12}). For example, with a 0 to 10V scale, a 0V signal would be 0, and a 10V signal would be 4095. This is equivalent to a a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF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

–10V – +10V
–5V – +5V

0V – 10V
0V – 5V

1V – 5V

4 – 20mA



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

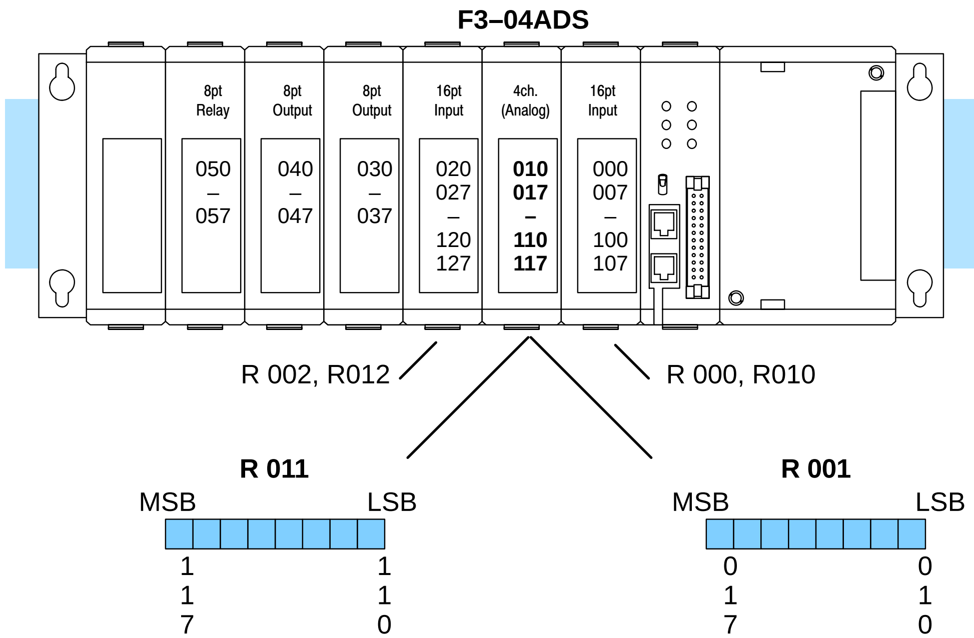
Range	Highest Signal	Lowest Signal	Smallest Change
–10 to +10V	+10V	–10V	4.88 mV
–5 to +5V	+5 V	–5V	2.44 mV
0 to 5V	5V	0V	1.22 mV
0 to 10V	10 V	0V	2.44 mV
1 to 5V	5V	1V	0.98 mV
4 to 20mA	20mA	4mA	3.91 μA

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)

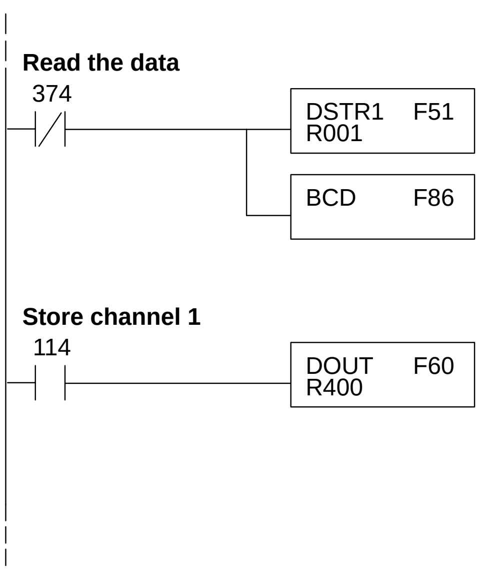
Identifying the Data Locations

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.



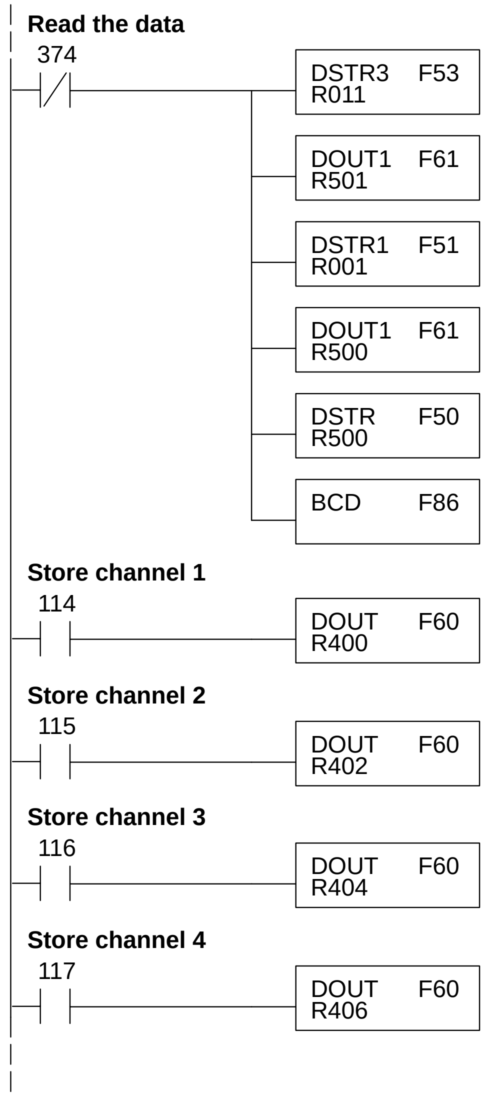
This rung loads the data into the accumulator on every scan. (You can use any permissive contact.)

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.

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

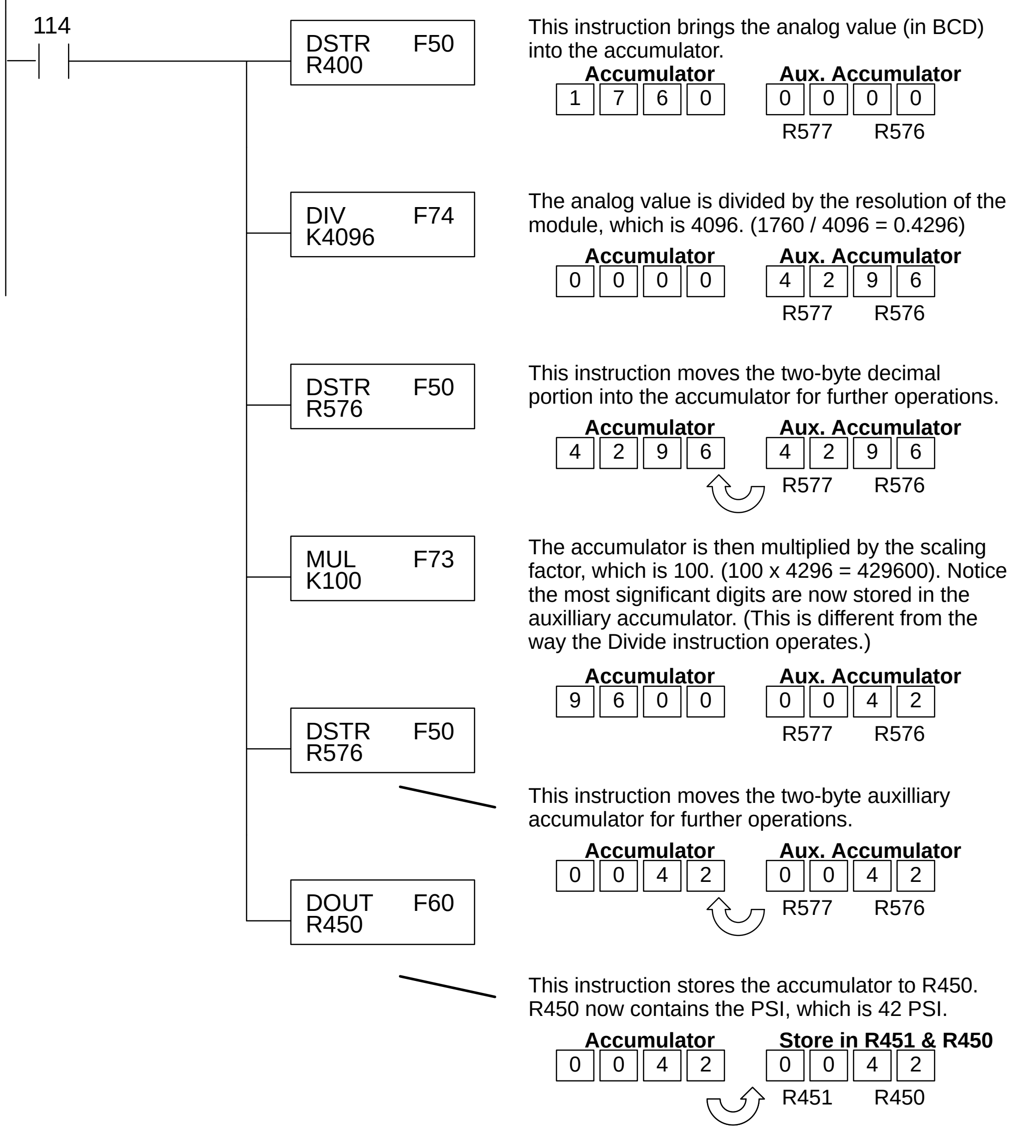
$$\text{Units} = \frac{1760}{4096}\,100$$

$$\text{Units} = 42.9$$

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.

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

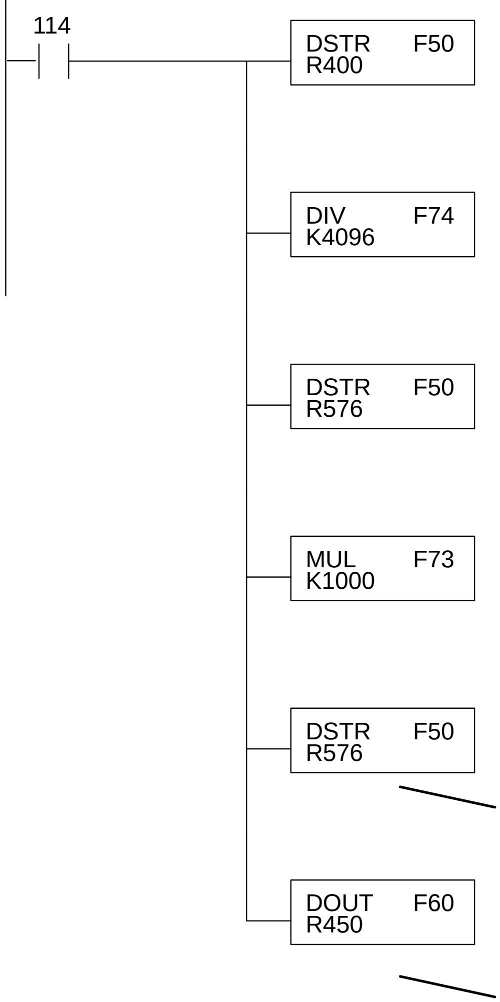
Scale the data



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

Scale the data



This instruction brings the analog value (in BCD) into the accumulator.

Accumulator				Aux. Accumulator			
1	7	6	0	0	0	0	0
				R577		R576	

The analog value is divided by the resolution of the module, which is 4096. (1760 / 4096 = 0.4296)

Accumulator				Aux. Accumulator			
0	0	0	0	4	2	9	6
				R577		R576	

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator			
4	2	9	6	4	2	9	6
				R577		R576	

The accumulator is multiplied by the scaling factor, which is now 1000. (1000 x 4296 = 4296000). The most significant digits are now stored in the auxilliary accumulator. (This is different from the way the Divide instruction operates.)

Accumulator				Aux. Accumulator			
6	0	0	0	0	4	2	9
				R577		R576	

This instruction moves the two-byte auxilliary accumulator for further operations.

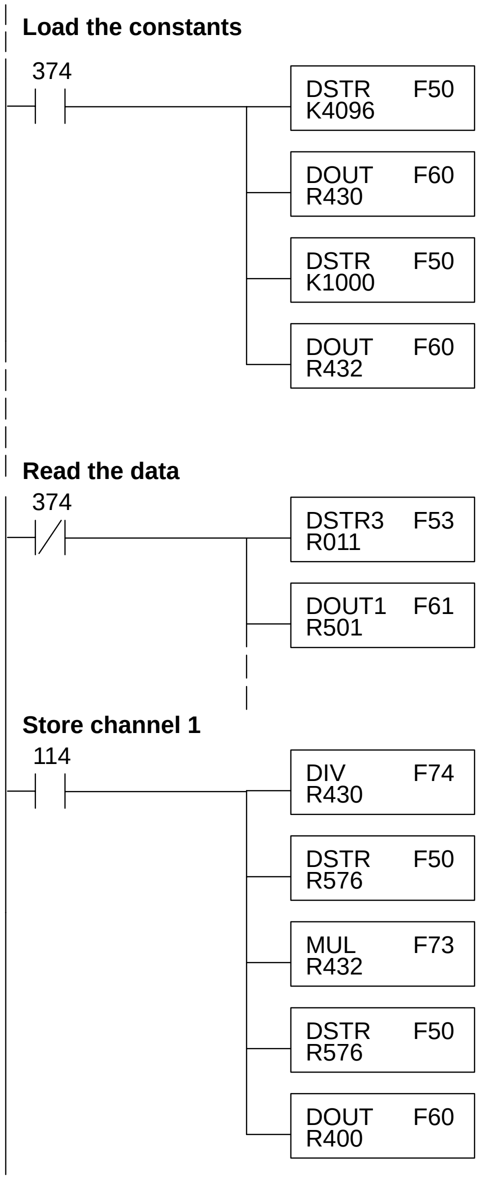
Accumulator				Aux. Accumulator			
0	4	2	9	0	4	2	9
				R577		R576	

This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the PSI, which implies 42.9.

Accumulator				Store in R451 & R450			
0	4	2	9	0	4	2	9
				R451		R450	

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.

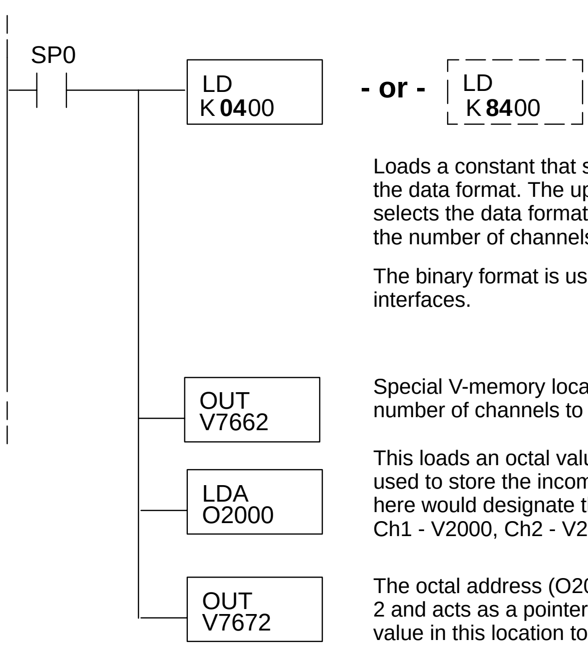
NOTE: Do not use the pointer method and the PID PV auto transfer from I/O module function together for the same module. If using PID loops, use the pointer method and ladder logic code to map the analog input data into the PID loop table.

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

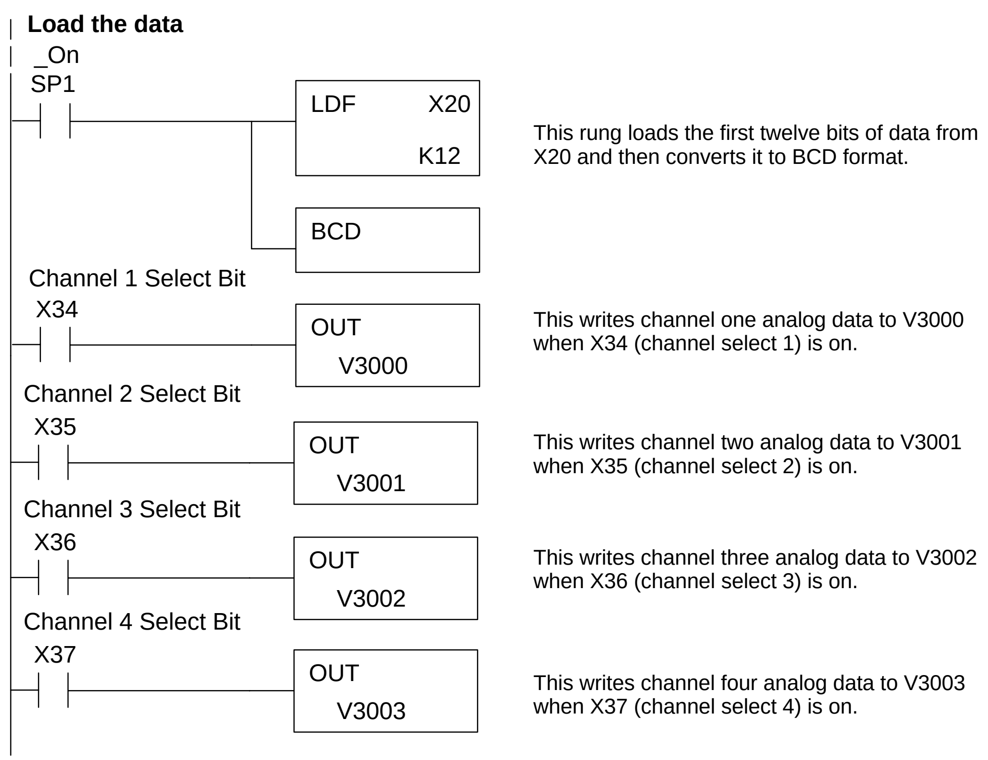
The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming data.

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

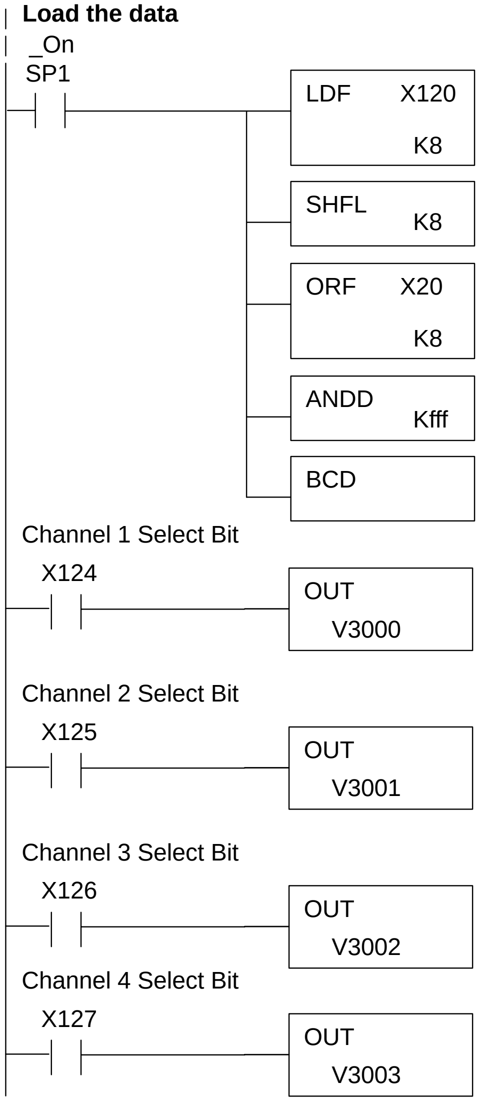
Multiplexing: DL350 with a D3–xx–1 Base

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.



This rung loads the upper byte of analog data from the module.

SHFL K8 shifts the data to the left eight places to make room for the lower byte of data.

The ORF X20 brings the lower byte of data from the module into the accumulator. At this time there is a full word of data from the analog module in the accumulator.

The ANDD Kfff masks off the twelve least significant bits of data from the word. This is the actual analog value.

The BCD command converts the data to BCD format.

This writes channel 1 analog data to V3000 when the Channel 1 Select Bit (X124) is on.

This writes channel 2 analog data to V3001 when the Channel 2 Select Bit (X125) is on.

This writes channel 3 analog data to V3002 when the Channel 3 Select Bit (X126) is on.

This writes channel 4 analog data to V3003 when the Channel 4 Select Bit (X127) is on.

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

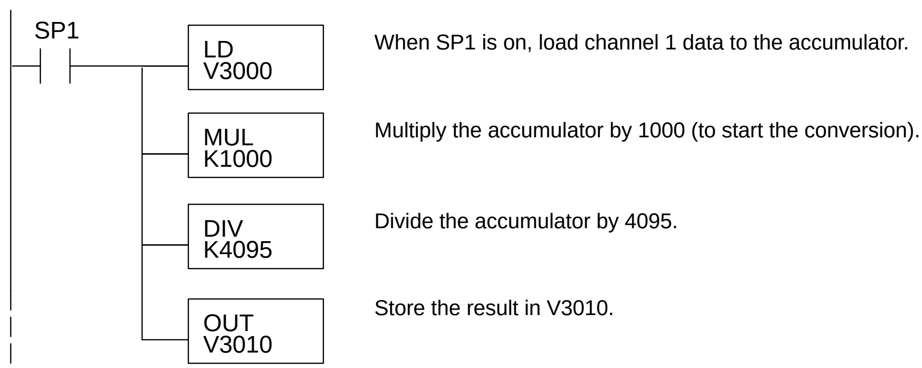
L = low limit of the engineering unit range

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.

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

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

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

$$D = 3276$$

4

F3–08AD–1
8-Channel
Analog Input

In This Chapter. . . .

Module Specifications

The following table provides the specifications for the F3–08AD Analog Input Module from FACTS Engineering. Review these specifications to make sure the module meets your application requirements.

Number of Channels	8, single ended (one common)
Input Ranges	4 – 20 mA
Resolution	12 bit (1 in 4096)
Input Impedance	250Ω ±0.1%, 1/2W current input
Absolute Maximum Ratings	±30mA
Conversion Time	35μs per channel 1 channel per CPU scan
Converter Type	Successive Approximation, AD574
Linearity Error	±1 count (0.03% of full scale) maximum
Maximum Inaccuracy	0.35% of full scale at 77 °F (25 °C)
Accuracy vs. Temperature	57 ppm / °C maximum full scale (including maximum offset change of 2 counts)
Recommended Fuse	0.032 A, Series 217 fast-acting
Power Budget Requirement	25 mA @ 9 VDC, 37 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

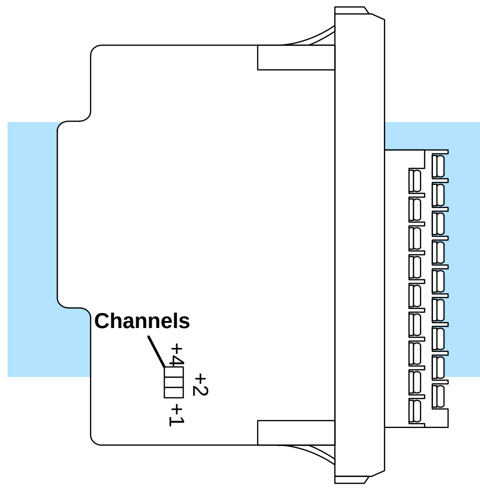
The F3–08AD Analog Input appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

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

Setting the Module Jumpers

Jumper Locations

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



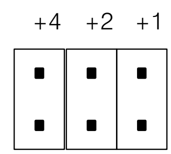
Selecting the Number of Channels

If you examine the rear of the module, you'll notice several jumpers. The jumpers labeled +1, +2 and +4 are used to select the number of channels that will be used. Without any jumpers the module processes one channel (channel 1). By installing the jumpers you can add channels. The module is set from the factory for eight 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 1–4, then the last four channels will not be active. The following table shows which jumpers to install.

Channel(s)	+4	+2	+1
1	No	No	No
1 2	No	No	Yes
1 2 3	No	Yes	No
1 2 3 4	No	Yes	Yes
1 2 3 4 5	Yes	No	No
1 2 3 4 5 6	Yes	No	Yes
1 2 3 4 5 6 7	Yes	Yes	No
1 2 3 4 5 6 7 8	Yes	Yes	Yes



Number of Channels

Jumpers installed as shown selects 8-channel operation

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 F3–08AD receives all power from the base. A separate power supply is not required.

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 F3–08AD 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.

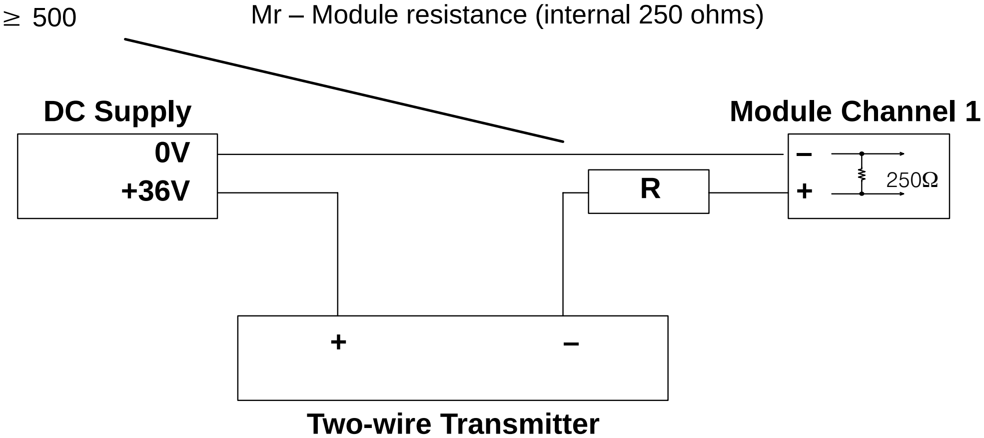
Consider the following example for a transmitter being operated from a 36 VDC supply with a recommended load resistance of 750 ohms. Since the module has a 250 ohm resistor, you need to add an additional resistor.

$R = Tr - Mr$

$R = 750 - 250$

$R \geq 500$

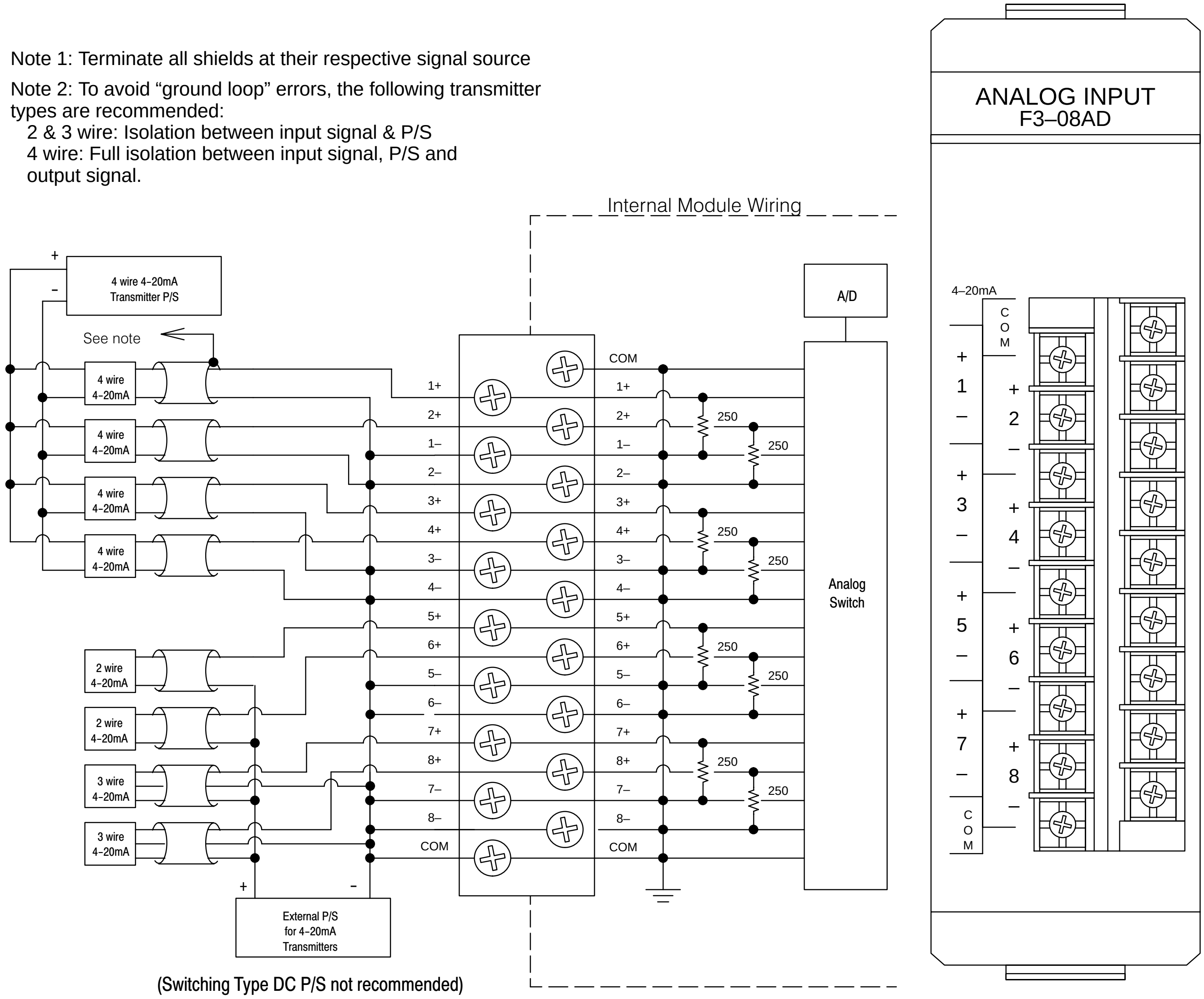
R – Resistor to add
Tr – Transmitter Requirement
Mr – Module resistance (internal 250 ohms)



Removable Connector

The F3–08AD 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



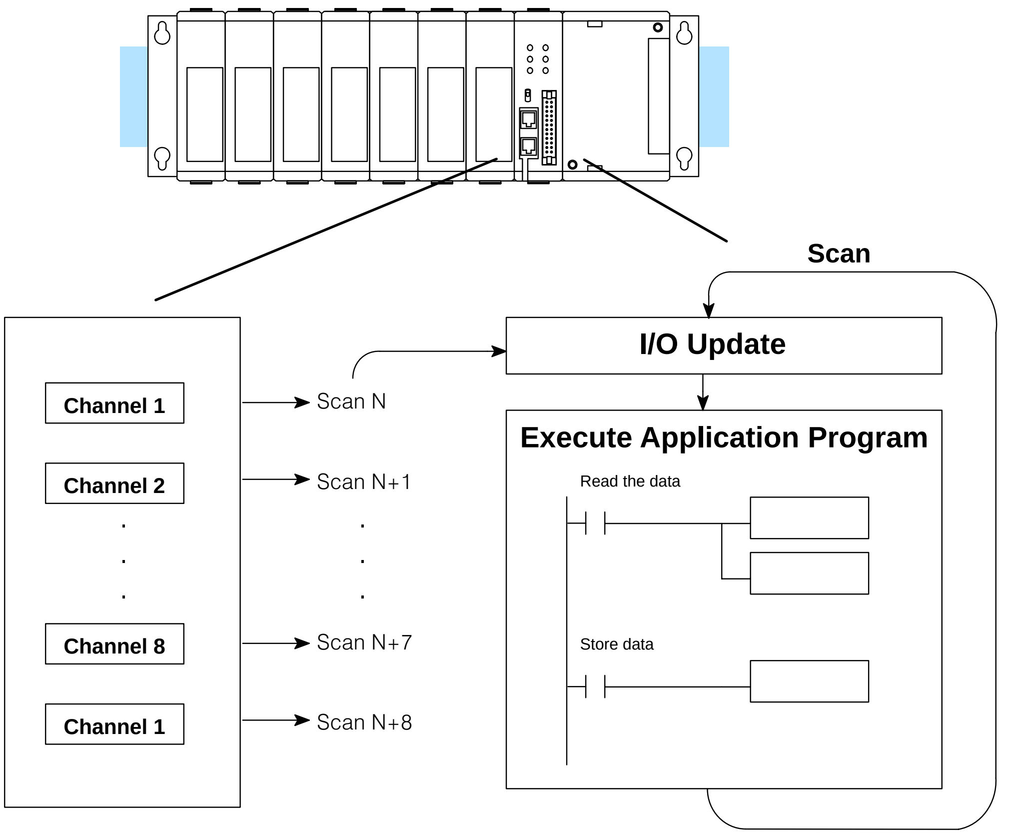
Module Operation

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 F3–08AD module supplies1 channel of data per each CPU scan. Since there are eight channels, it can take up to eight scans to get data for all channels. Once all channels have been scanned the process starts over with channel 1.

You do not have to select all of the channels. Unused channels are not processed, so if you select only four channels, then the channels will be updated within four scans.



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.

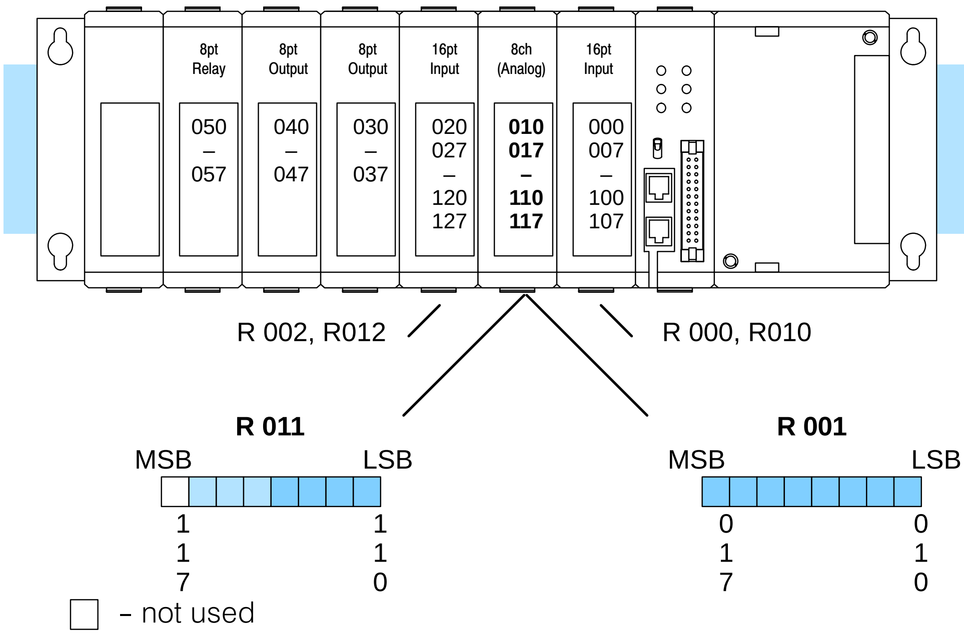
Understanding the I/O Assignments

You may recall the F3–08AD 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.

F3–08AD

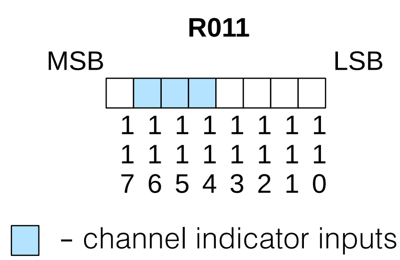


Within these two register locations, the individual bits represent specific information about the analog signal.

Active Channel Indication Inputs

The next to last three bits of the upper Register indicate the active channel. The indicators automatically increment with each CPU scan.

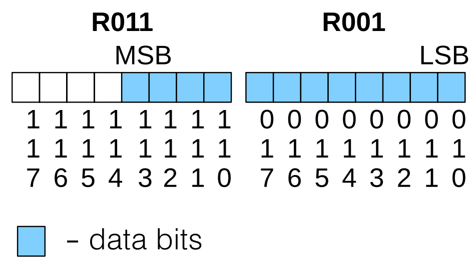
Scan	Channel Inputs	Active Channel
N	000	1
N+1	001	2
N+2	010	3
N+3	011	4
N+4	100	5
N+5	101	6
N+6	110	7
N+7	111	8
N+8	000	1



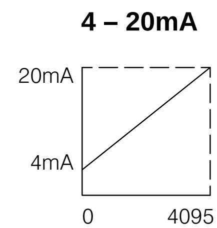
Analog Data Bits

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 4 – 20 mA scale, a 4 mA signal would be 0, and a 20 mA 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 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

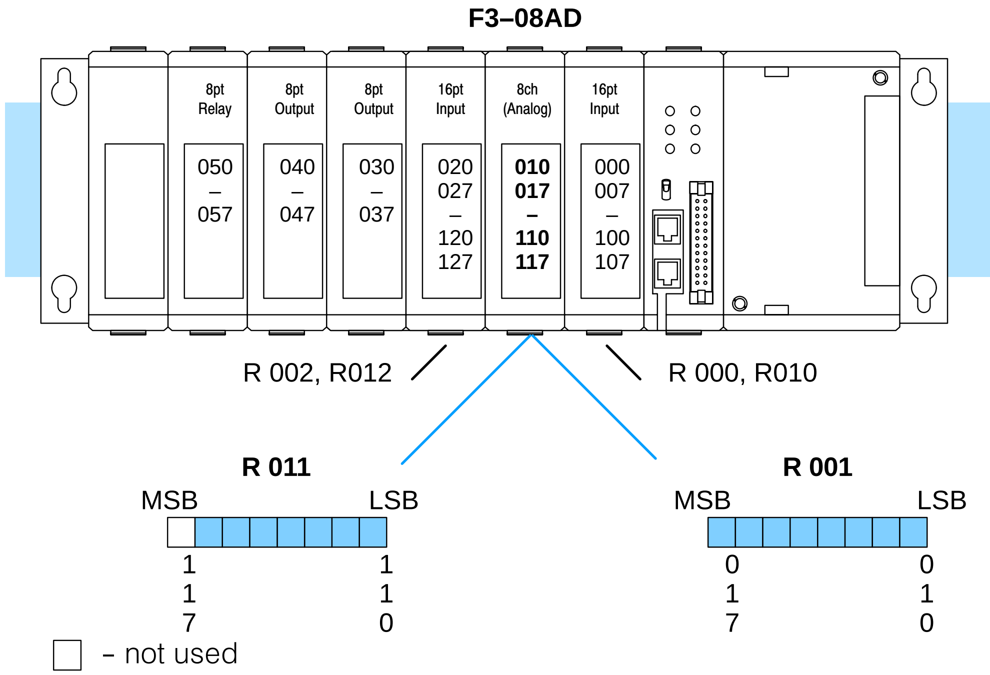
Range	Highest Signal	Lowest Signal	Smallest Change
4 to 20mA	20mA	4mA	3.91 μA

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)

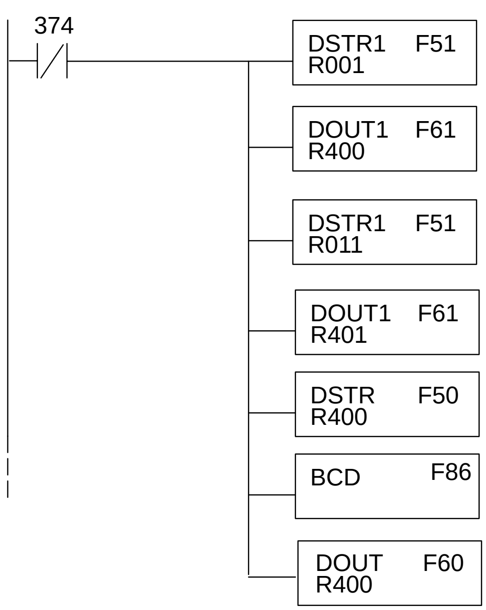
Identifying the Data Locations

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.



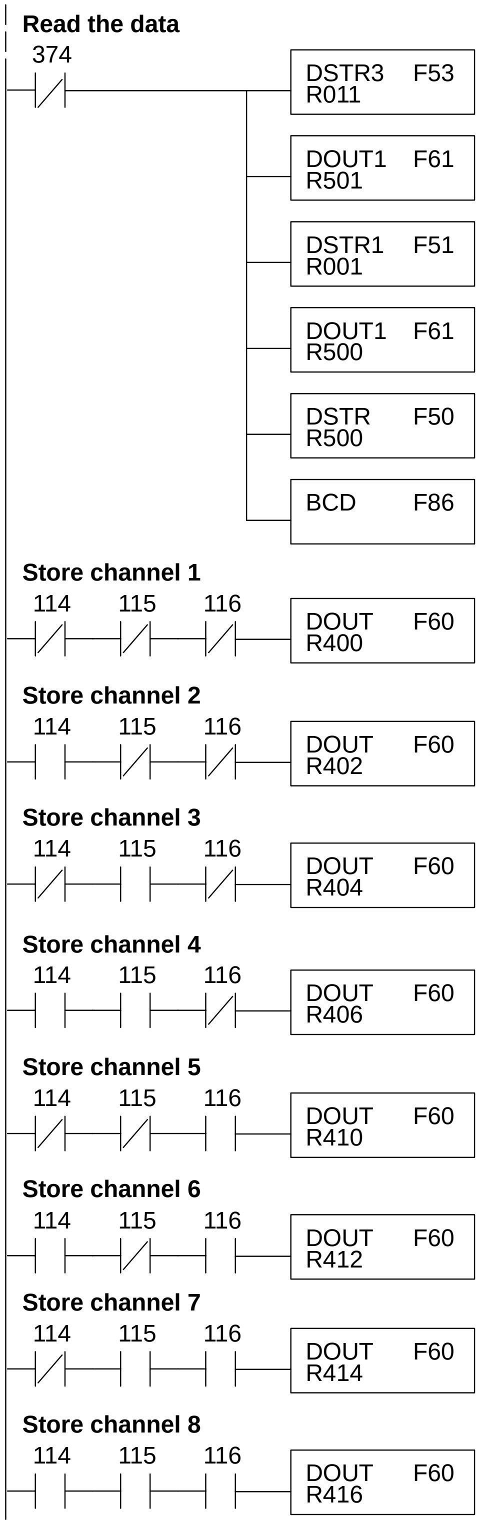
This rung loads the data into the accumulator on every scan. (You can use any permissive contact.)

Since the active channel indicators are all off when channel 1 is being read, you would not have to use them. (Since you cannot isolate the individual channels for scanning, channel 1 is the only channel that can be used in this manner.) The DOUT1 instruction moves 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.)

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.)

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 on every scan. (You could use any permissive contact.)

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 indicator 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.

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

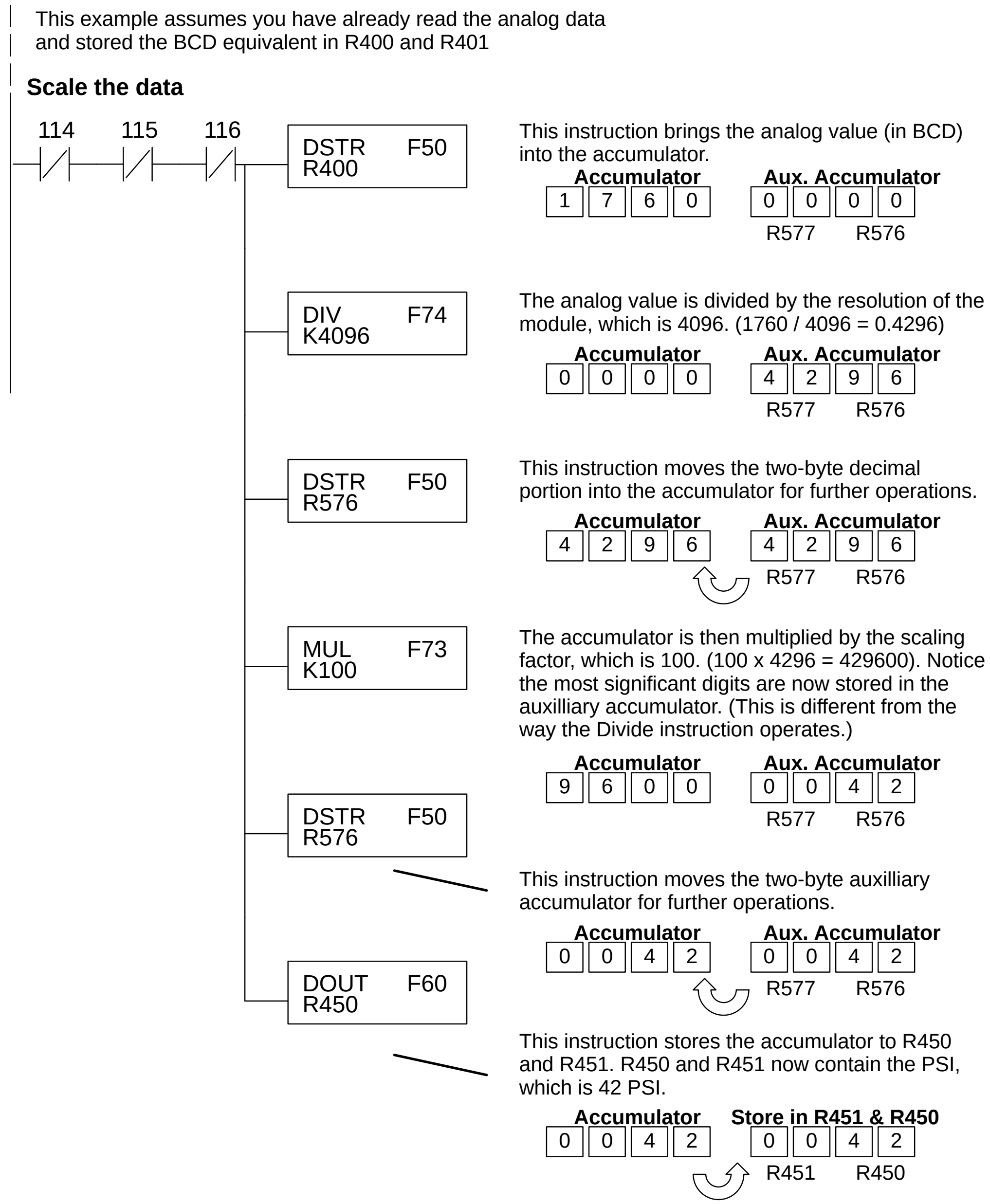
$$\downarrow$$

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

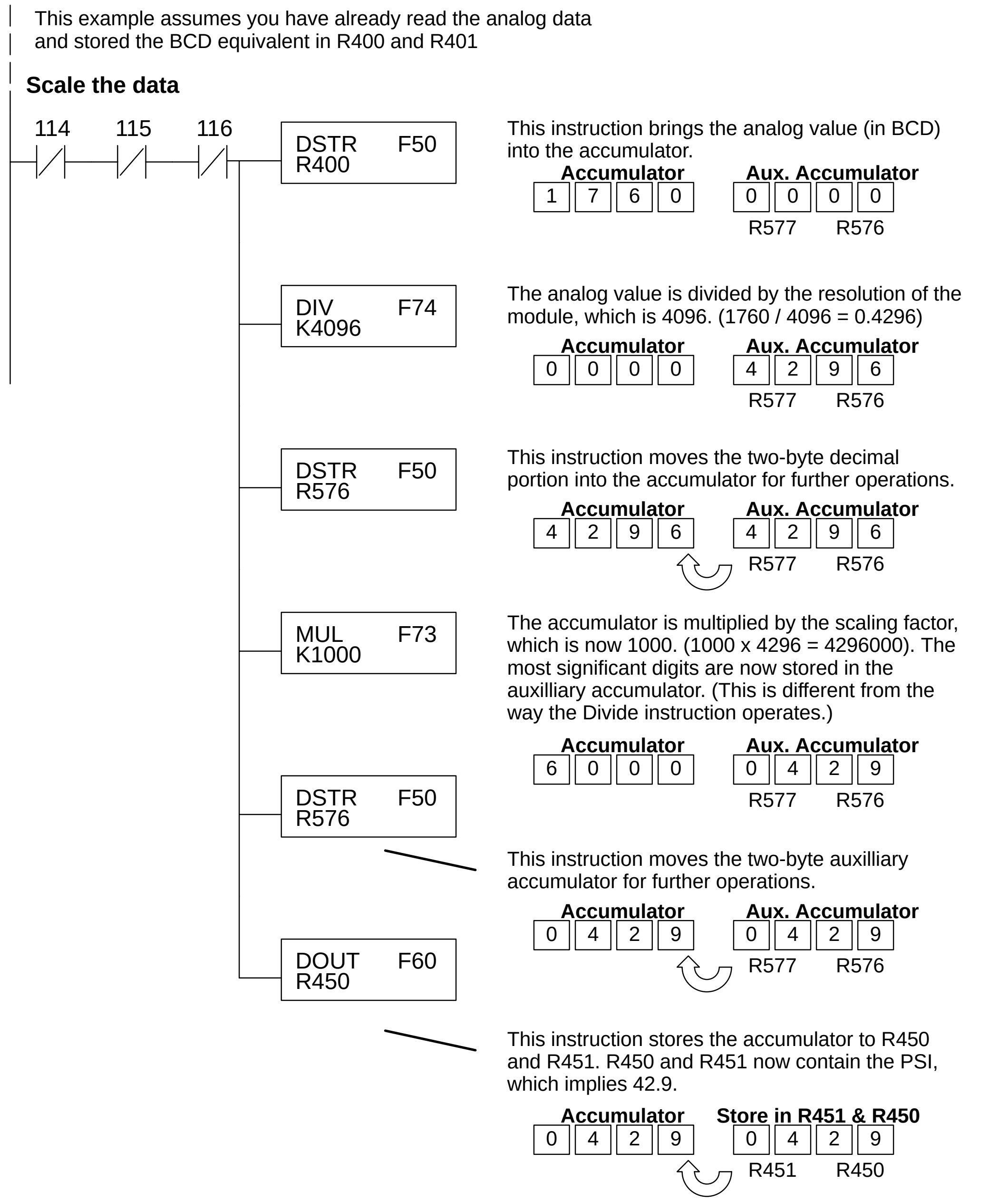
$$\downarrow$$

$$\text{Units} = 42.9$$

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, input 115, and input 116 are all off when channel 1 data is being read. Of course, if you were using a different channel, you would use the active channel indicator point combination that corresponds to the channel you were using.

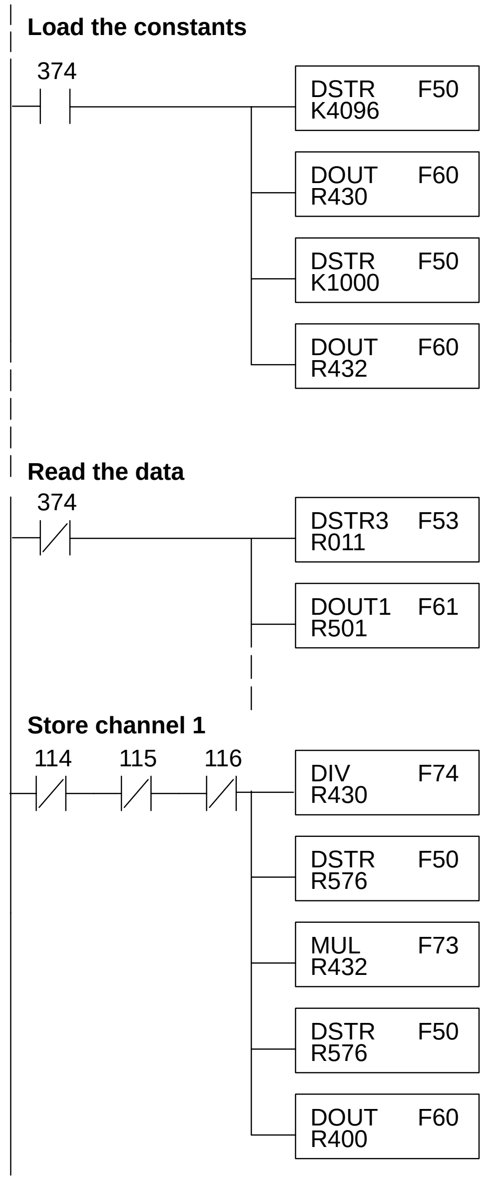


You probably noticed the previous example yielded 42 PSI when the real value should have been 42.9 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve added another digit to the scale. Instead of a scale of 100, we’re using 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 was written for channel 1, but you could easily use a similar approach to use different scales for all channels if required.

You could 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 on every scan. (You could use any permissive contact.)

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.

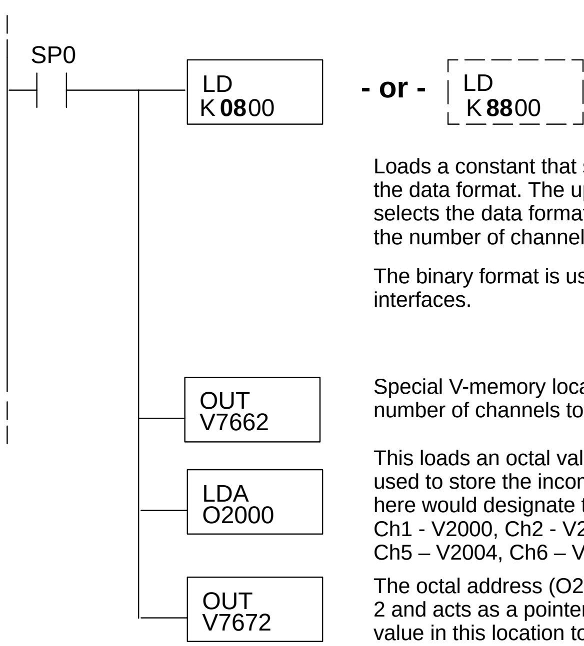
NOTE: Do not use the pointer method and the PID PV auto transfer from I/O module function together for the same module. If using PID loops, use the pointer method and ladder logic code to map the analog input data into the PID loop table.

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, 5, 6, 7, 8).

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, Ch4 - V2003,
Ch5 – V2004, Ch6 – V2005, Ch7 – V2006, Ch8 – V2007

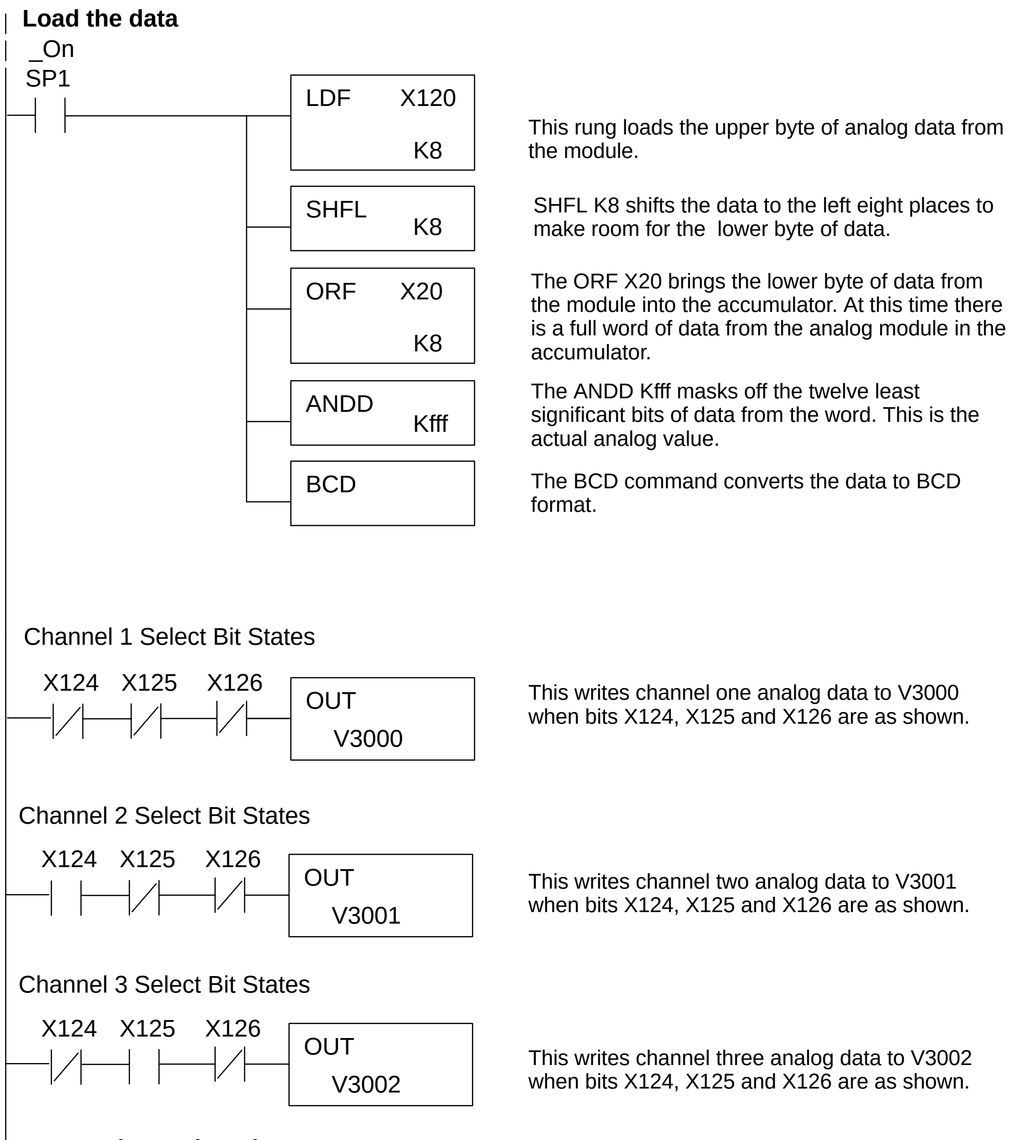
The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming data.

The 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 Conventional DL305 Base

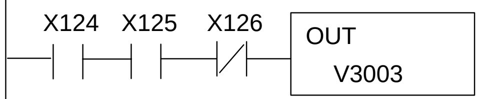
The example below shows how to read multiple channels on an F3–08AD Analog module in the X20–27 / X120–127 address slot. This module must be placed in a 16 bit slot in order to work.



example continued on next page

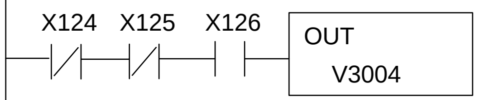
example continued from previous page

Channel 4 Select Bit States



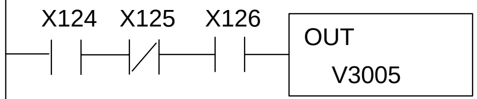
This writes channel four analog data to V3003 when bits X124, X125 and X126 are as shown.

Channel 5 Select Bit States



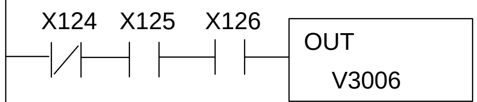
This writes channel five analog data to V3004 when bits X124, X125 and X126 are as shown.

Channel 6 Select Bit States



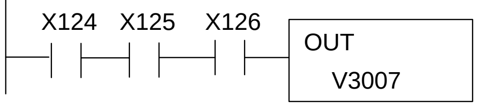
This writes channel six analog data to V3005 when bits X124, X125 and X126 are as shown.

Channel 7 Select Bit States



This writes channel seven analog data to V3006 when bits X124, X125 and X126 are as shown.

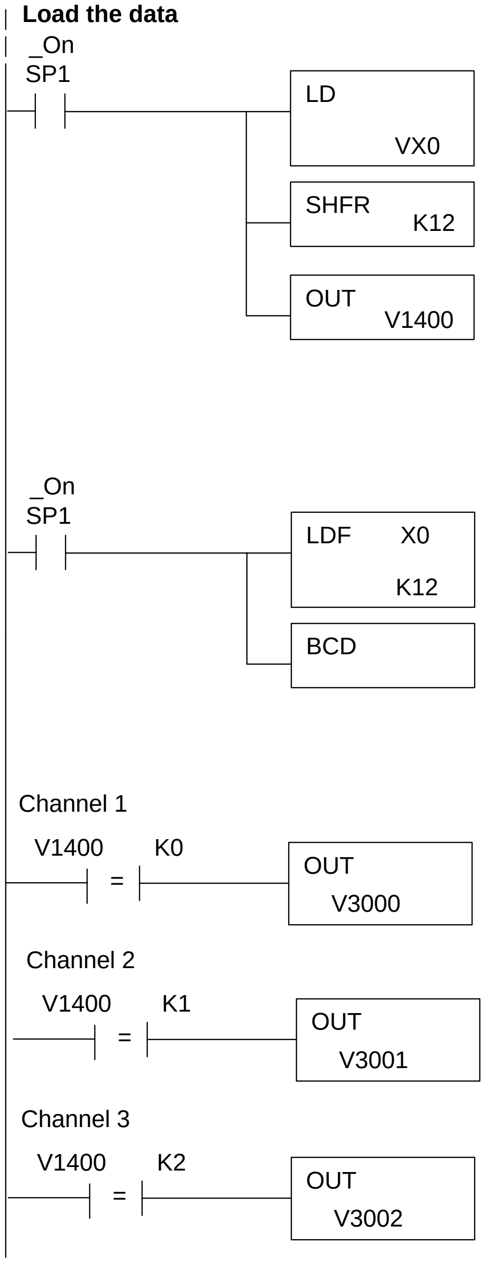
Channel 8 Select Bit States



This writes channel eight analog data to V3007 when bits X124, X125 and X126 are as shown.

Multiplexing: DL350 with a D3–xx–1 Base

The example below shows how to read multiple channels on an F3–08AD Analog module in the X0 address slot of a 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.



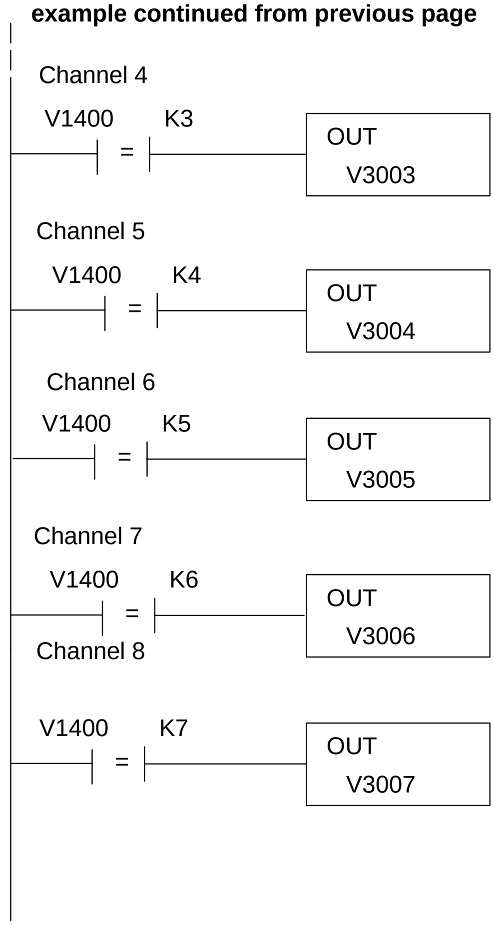
This rung loads the only the channel select bits into V1400. The SHFR shifts the analog data out of the word.

This rung loads the only the analog input data and converts it to BCD.

These rungs store the BCD analog input data into consecutive V memory registers. V1400 will increment once per scan from 0 to 7.

example continued on next page

example continued from previous page



These rungs store the BCD analog input data into consecutive V memory registers. V1400 will increment once per scan from 0 to 7.

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

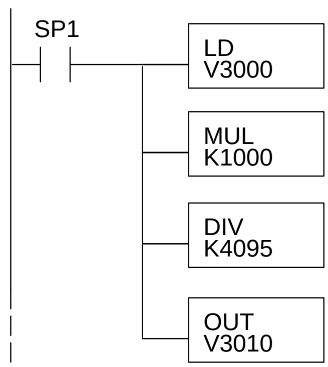
L = low limit of the engineering unit range

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.



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

Multiply the accumulator by 1000 (to start the conversion).

Divide the accumulator by 4095.

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 ...
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 would use the following formula to determine the digital value that should be stored in the register location that contains the data.

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

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

$$D = (255.93)\,(6)$$

$$D = 1536$$

5

F3–16AD 16-Channel 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 F3–16AD Analog Input Module from FACTS Engineering. Review these specifications to make sure the module meets your application requirements.

Number of Channels	16, single ended (one common)
Input Ranges	±5V, ±10V, 0–5V[1], 0–10V, 0–20 mA, 4 – 20 mA[2]
Resolution	12 bit (1 in 4096)
Input Impedance	2MΩ, voltage input 500Ω ±1%, current input
Absolute Maximum Ratings	±25V, voltage input ±30 mA, current input
Conversion Time	35µs per channel 1 channel per CPU scan
Converter Type	Successive Approximation, AD574
Linearity Error	±1 count maximum
Maximum Inaccuracy at 77 °F (25 °C)	0.25% of full scale, voltage input 1.25% of full scale, current input
Accuracy vs. Temperature	57 ppm / °C maximum full scale
Recommended Fuse	0.032 A, Series 217 fast-acting, current inputs
Power Budget Requirement	33 mA @ 9 VDC, 47 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

1 – requires gain adjustment with potentiometer.
2 – resolution is 3275 counts (instead of 4096). Allows easier broken transmitter detection

Analog Input Configuration Requirements

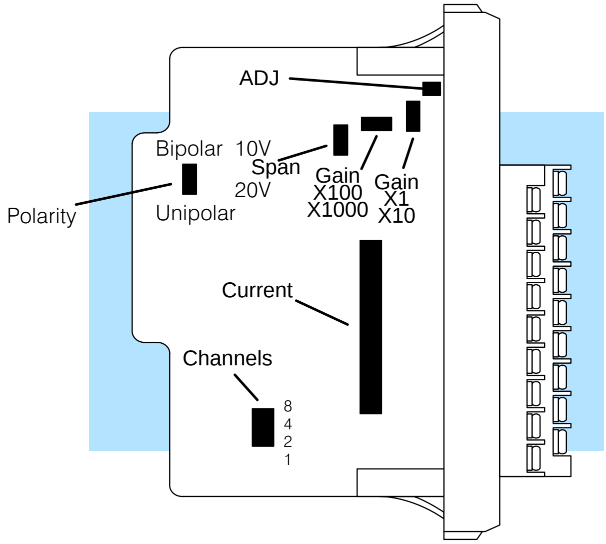
The F3–16AD Analog Input appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

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

Setting the Module Jumpers

Jumper Locations

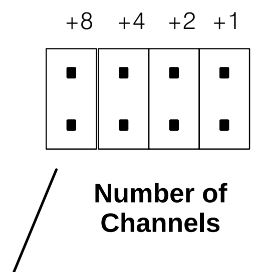
The module is set at the factory for a 0–20 mA signal on all sixteen channels. If this is acceptable you do not have to change any of the jumpers. The following diagram shows the jumper locations.



Selecting the Number of Channels

If you examine the rear of the module, you'll notice several jumpers. The jumpers labeled +1, +2, +4 and +8 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 sixteen channel operation.

Any unused channels are not processed so if you only select channels 1–8, then the last eight channels will not be active. The following table shows which jumpers to install.



Jumpers installed as shown selects 16-channel operation

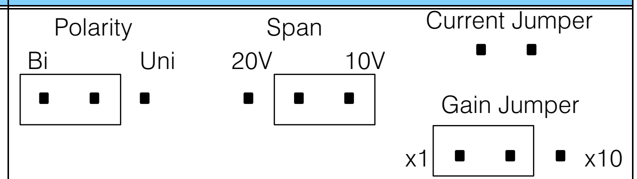
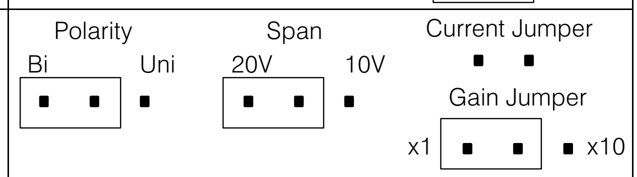
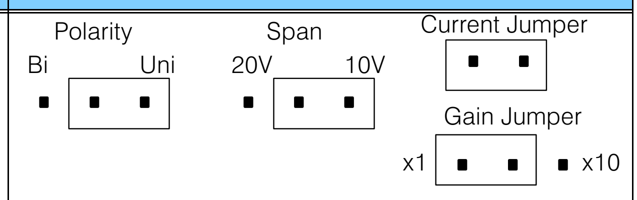
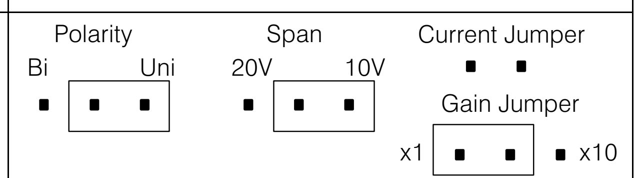
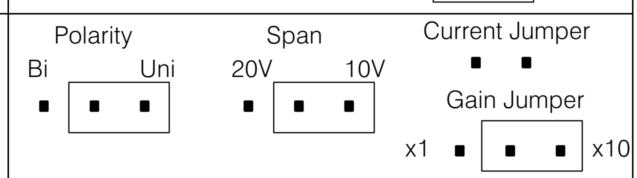
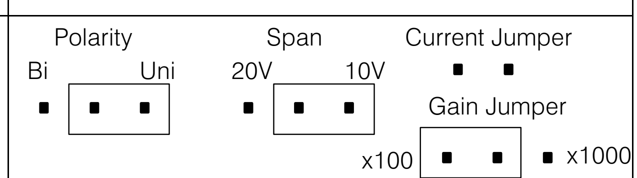
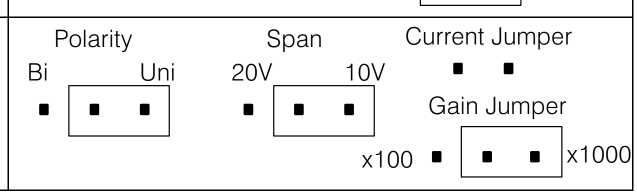
Channel(s)	Jumper			
	+8	+4	+2	+1
1	No	No	No	No
1 2	No	No	No	Yes
1 2 3	No	No	Yes	No
1 2 3 4	No	No	Yes	Yes
1 2 3 4 5	No	Yes	No	No
1 2 3 4 5 6	No	Yes	No	Yes
1 2 3 4 5 6 7	No	Yes	Yes	No
1 2 3 4 5 6 7 8	No	Yes	Yes	Yes

Channel(s)	Jumper			
	+8	+4	+2	+1
1 2 3 4 5 6 7 8 9	Yes	No	No	No
1 2 3 4 5 6 7 8 9 10	Yes	No	No	Yes
1 2 3 4 5 6 7 8 9 10 11	Yes	No	Yes	No
1 2 3 4 5 6 7 8 9 10 11 12	Yes	No	Yes	Yes
1 2 3 4 5 6 7 8 9 10 11 12 13	Yes	Yes	No	No
1 2 3 4 5 6 7 8 9 10 11 12 13 14	Yes	Yes	No	Yes
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Yes	Yes	Yes	No
1 2 3 4 5 6 7 8 9 10 11 12 13141516	Yes	Yes	Yes	Yes

Selecting Input Signal Ranges

As you examined the jumper settings, you may have noticed there are current jumpers for each individual channel. These jumpers allow you to select the type of signal (voltage or current).

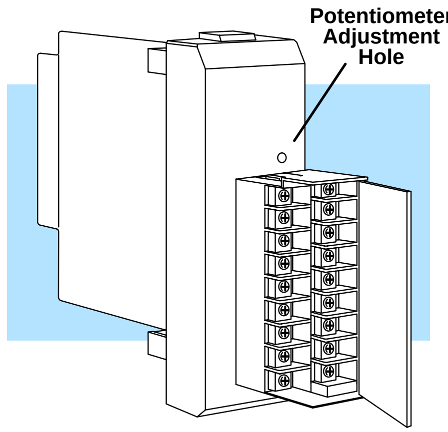
The span and polarity jumpers are used to select the signal range. The polarity and span selection affect all the channels. For example, if you select unipolar operation and a 10V span, you can use both 0 –10V and 0–20 mA signals at the same time. Channels that will receive 0–20 mA signals should have the current jumper installed. The following table shows the jumper selections for the various ranges. (Only channel 1 is used in the example, but all channels must be set.)

Bipolar Signal Range	Jumper Settings
–5 VDC to +5 VDC	
–10 VDC to +10 VDC	
Unipolar Signal Range	**Jumper Settings**
0 to 20 mA (these settings are also used for the 4–20mA range)	
0 VDC to +10 VDC	
0 VDC to +1 VDC	
0 VDC to +0.1 VDC	
0 VDC to +0.01 VDC	

Input Signal Range	Jumper Settings
0 VDC to +5 VDC (requires gain adjustment see instructions below)	Polarity: Bi, Uni; Span: 20V, 10V; Current Jumper; Gain Jumper: x1, x10
0 VDC to +12 VDC (requires gain adjustment see instructions below)	Polarity: Bi, Uni; Span: 20V, 10V; Current Jumper; Gain Jumper: x1, x10

Variable Gain Adjustment

If you look at the terminal block closely, you'll notice a small hole conceals an adjustment potentiometer. This small potentiometer is used to adjust the gain for certain situations.



For example, if you have 0–5V transmitters you have to use the 0–10V scale on the module. Since the module converts the signal to a digital value between 0 and 4095, a 5V signal would only yield a value of 2048. Fortunately, the variable gain feature provides a simple solution. Just complete the following steps.

1. Install a jumper on the gain adjustment pins. (This jumper location is labeled ADJ. This jumper will remain installed after the gain adjustment .)
2. Apply 5V to one of the channels.
3. Use a handheld programmer or ***Direct***SOFT to monitor the input register that contains the analog data. (If you're not familiar with this procedure, wait until you read the section on Writing the Control Program. This will show you how to get data into a register. You can come back to this procedure later.)
4. Adjust the potentiometer until the register value reads 4094 or 4095. The potentiometer is turned clockwise to increase the gain.

Now the module has been adjusted so a 5V signal provides a digital value of 4095 instead of 2048.

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.

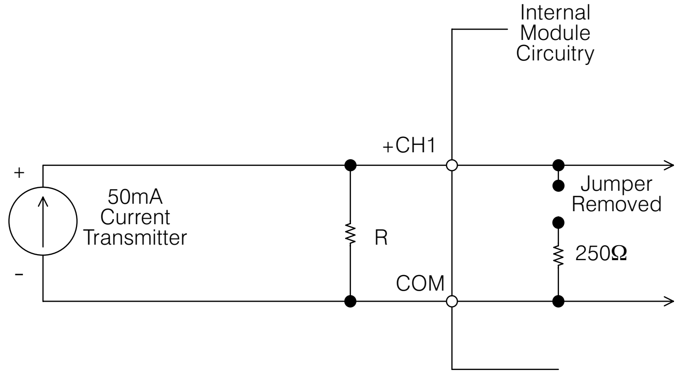
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the source.
- Don't run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.
- 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 F3–16AD 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 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, 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 that has $\pm 0.1\%$ 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 at the various voltages.

The F3–16AD provides 500 ohm resistance for each channel. If your transmitter requires a load resistance below 500 ohms, then you do not have to make any adjustments. However, if your transmitter requires a load resistance higher than 500 ohms, then you need to add a resistor in series with the module.

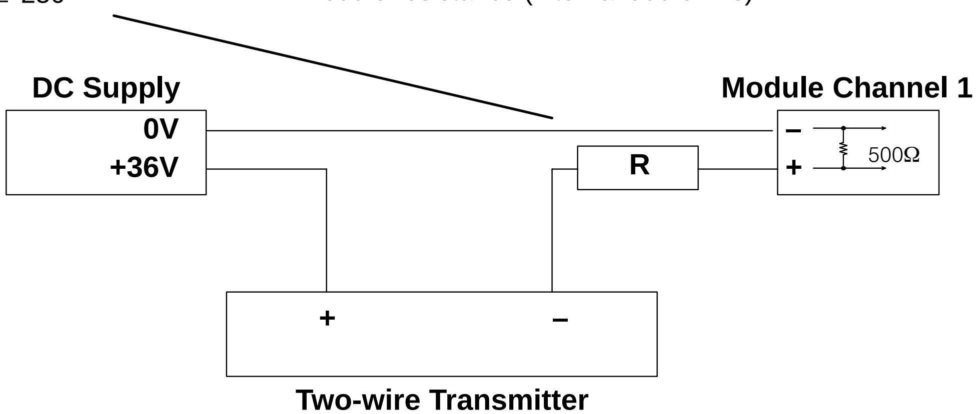
Consider the following example for a transmitter being operated from a 36 VDC supply with a recommended load resistance of 750 ohms. Since the module has a 500 ohm resistor, you need to add an additional resistor.

$R = Tr - Mr$

$R = 750 - 500$

$R \geq 250$

R – Resistor to add
Tr – Transmitter Requirement
Mr – Module resistance (internal 500 ohms)

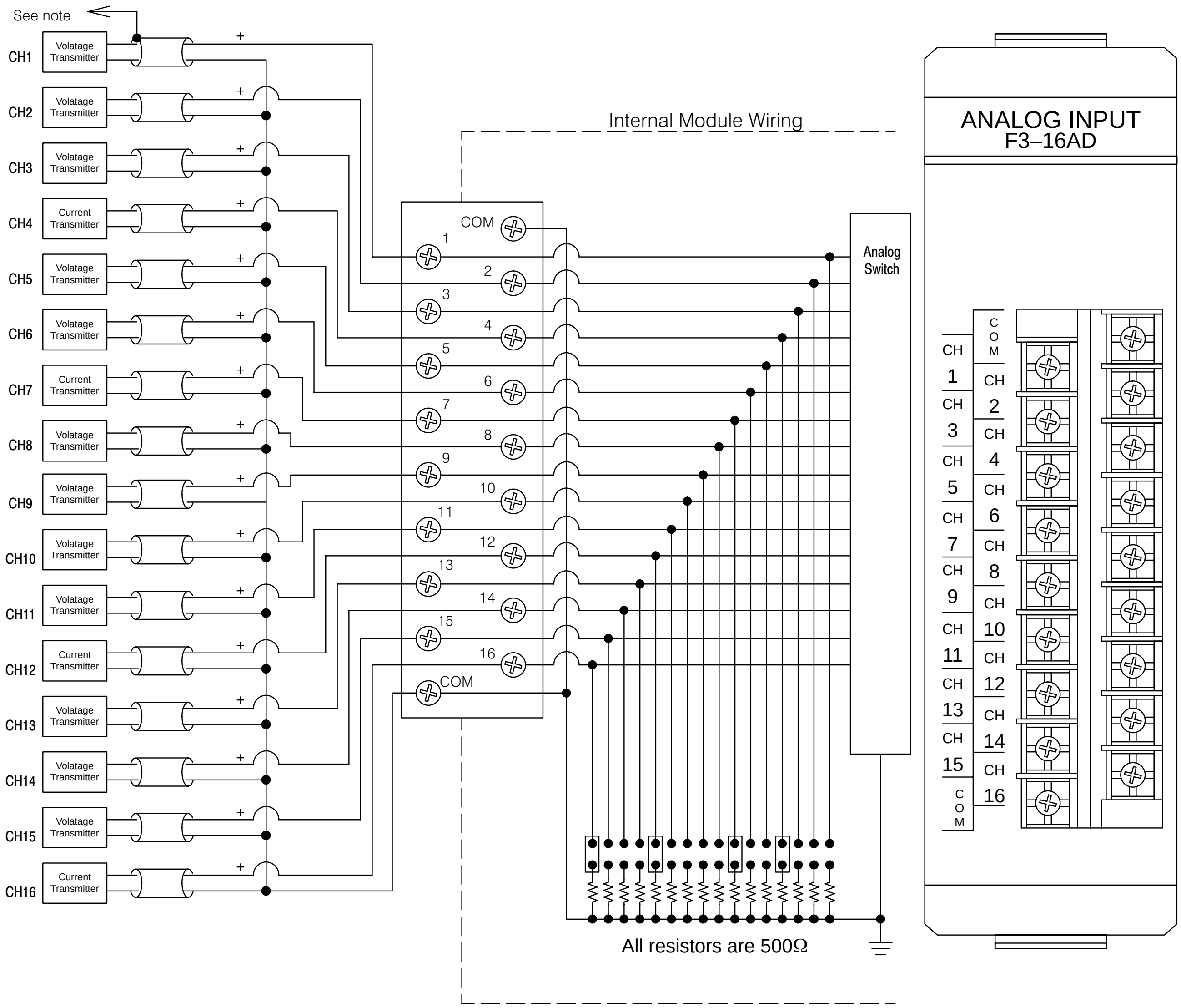


Removable Connector

The F3–16AD 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: Terminate all shields at their respective signal source.
Note 2: Jumpers for CH4, 7, 12 and 16 are installed for current input.



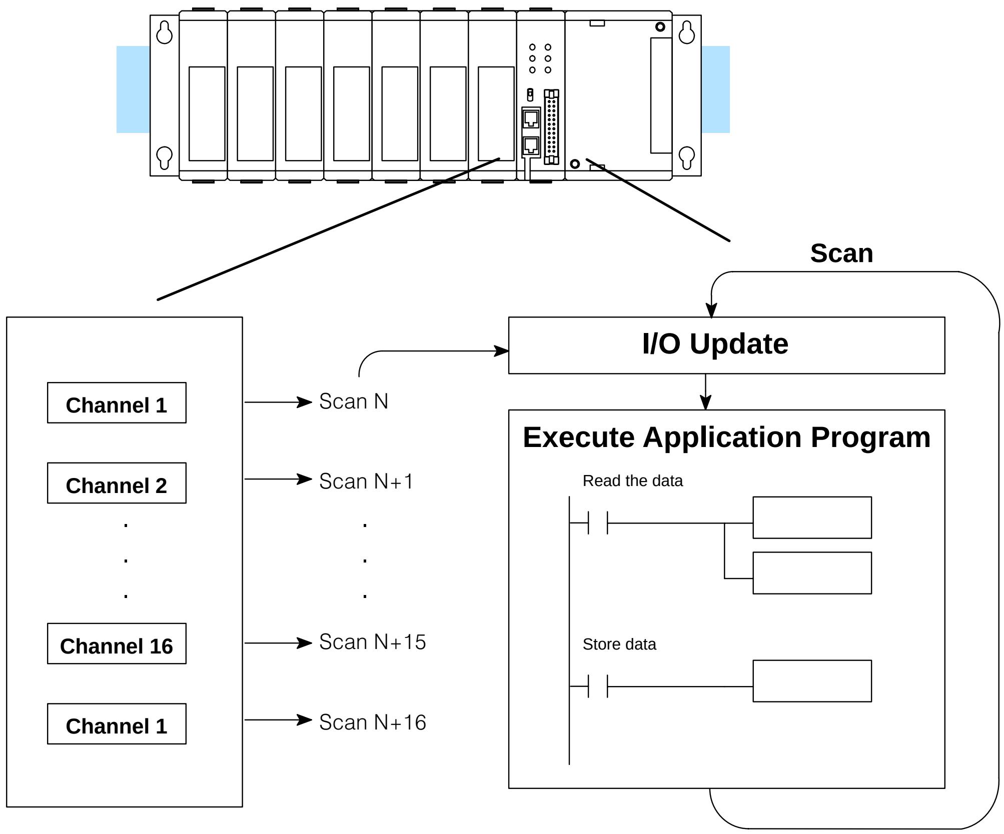
Module Operation

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 F3–16AD module supplies 1 channel of data per each CPU scan. Since there are sixteen channels, it can take up to sixteen 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 eight channels, then the channels will be updated within eight scans.



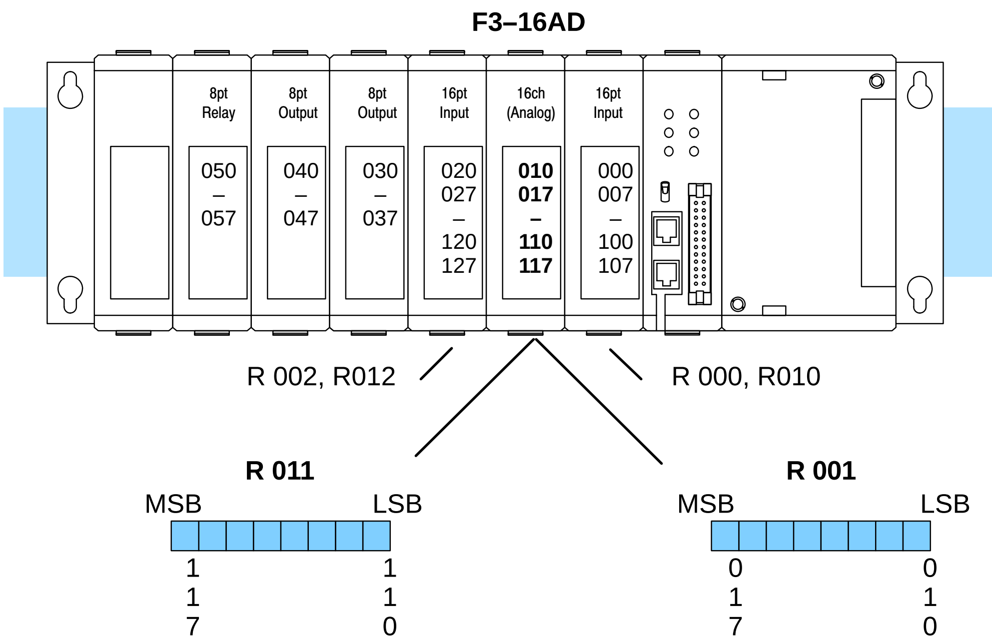
Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 12-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete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

Understanding the I/O Assignments

You may recall the F3–16AD 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.

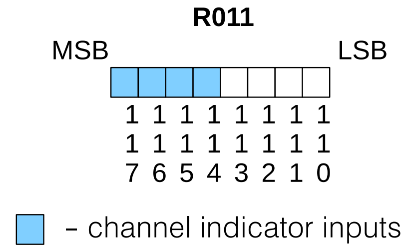


Within these two register locations, the individual bits represent specific information about the analog signal.

Active Channel Indicator Inputs

The last four inputs of the upper Register indicate the active channel. The indicators automatically increment with each CPU scan.

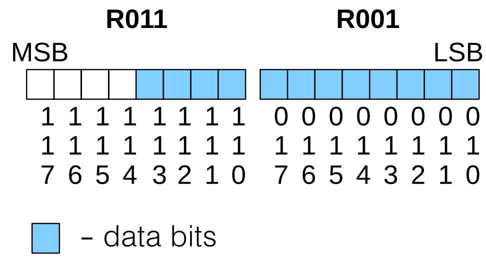
Scan	Channel Inputs	Active Channel
N	0000	1
N+1	0001	2
N+2	0010	3
N+3	0011	4
N+4	0100	5
N+5	0101	6
N+6	0110	7
N+7	0111	8
N+8	1000	9
N+9	1001	10
N+10	1010	11
N+11	1011	12
N+12	1100	13
N+13	1101	14
N+14	1110	15
N+15	1111	16



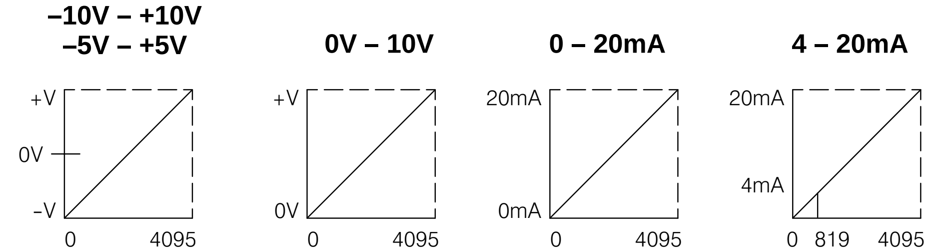
Analog Data Bits

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.



NOTE: When you use 4–20mA signals, you have to use the 0–20mA scale. You do not have resolution of 4096 if the 4–20mA signal is present. In this case, the range is 819 to 4095. This is because a 0 still represents 0mA, not 4mA.

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 possibly 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

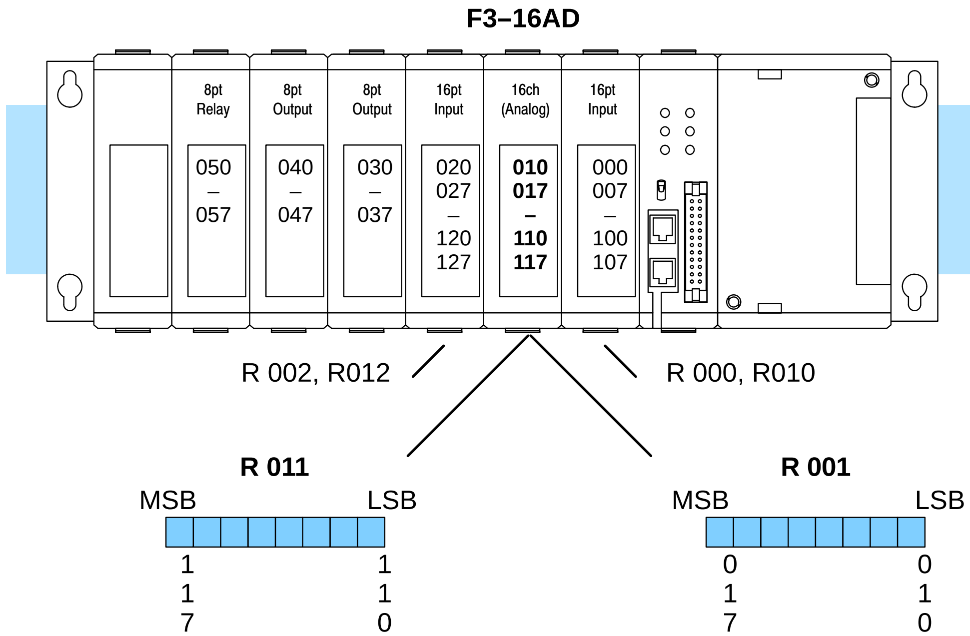
L = low limit of the signal range

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	10V	0V	2.44 mV
0 to 12V	12V	0V	2.90 mV
0 to 20mA (4 to 20mA also)	20mA	0mA	4.88 µA
0 to 1V	1 V	0 V	0.244 mV
0 to 0.1V	0.1 V	0 V	24.4 uV
0 to 0.01V	0.01 V	0 V	2.44 uV

Writing the Control Program (DL330 / DL340)

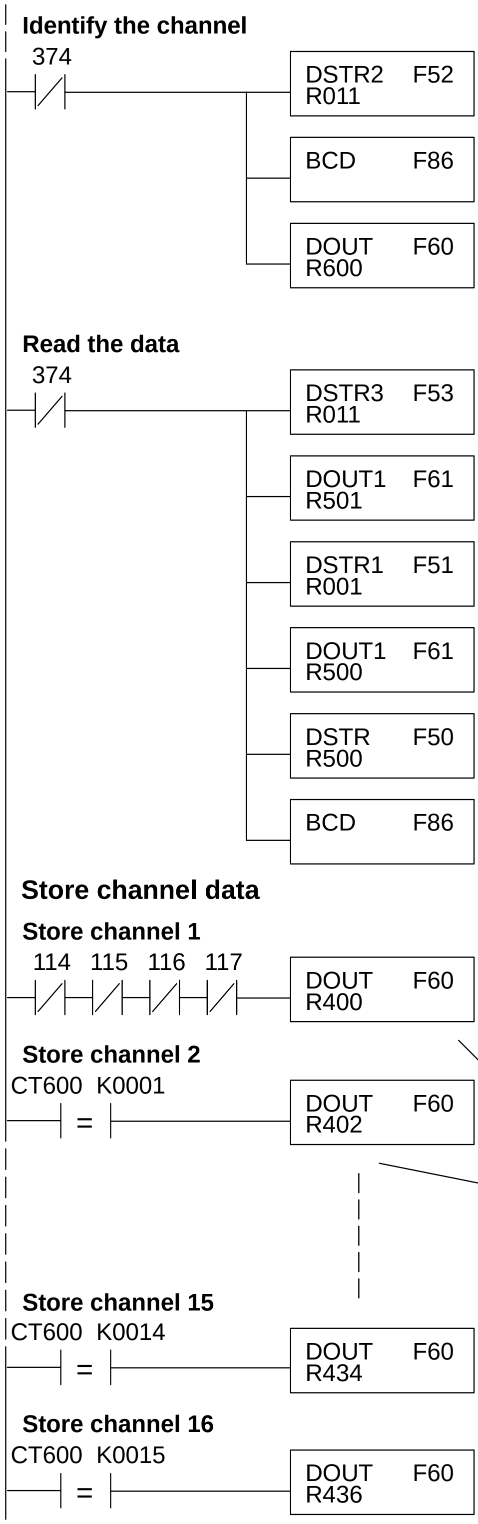
Identifying the Data Locations

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.



Example Program

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 pretty simple if you follow the example.



This rung loads the channel ID bits into the accumulator from Register 011 on every scan.

Convert the channel ID status to BCD. (We'll use relational contacts later to make the chanel selection much easier.)

Store the channel ID in R600. (Note, you don't absolutely have to do it this way. If you use R600, then you can't use Timer/Counter 600. You could just use the channel indicators. See the Store Channel 1 example that follows.)

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

This rung shows how you would use the channel indicator inputs as contacts to control the channel selection.

This rung shows an easier way. Earlier we loaded the channel ID bits (in BCD format) into R600. Now we can use one relational contact to examine this value. However, this method uses the register associated with Timer/Counter 600. If you use this method, make sure you don't use the Timer/Counter associated with the register elsewhere in the program.

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.

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

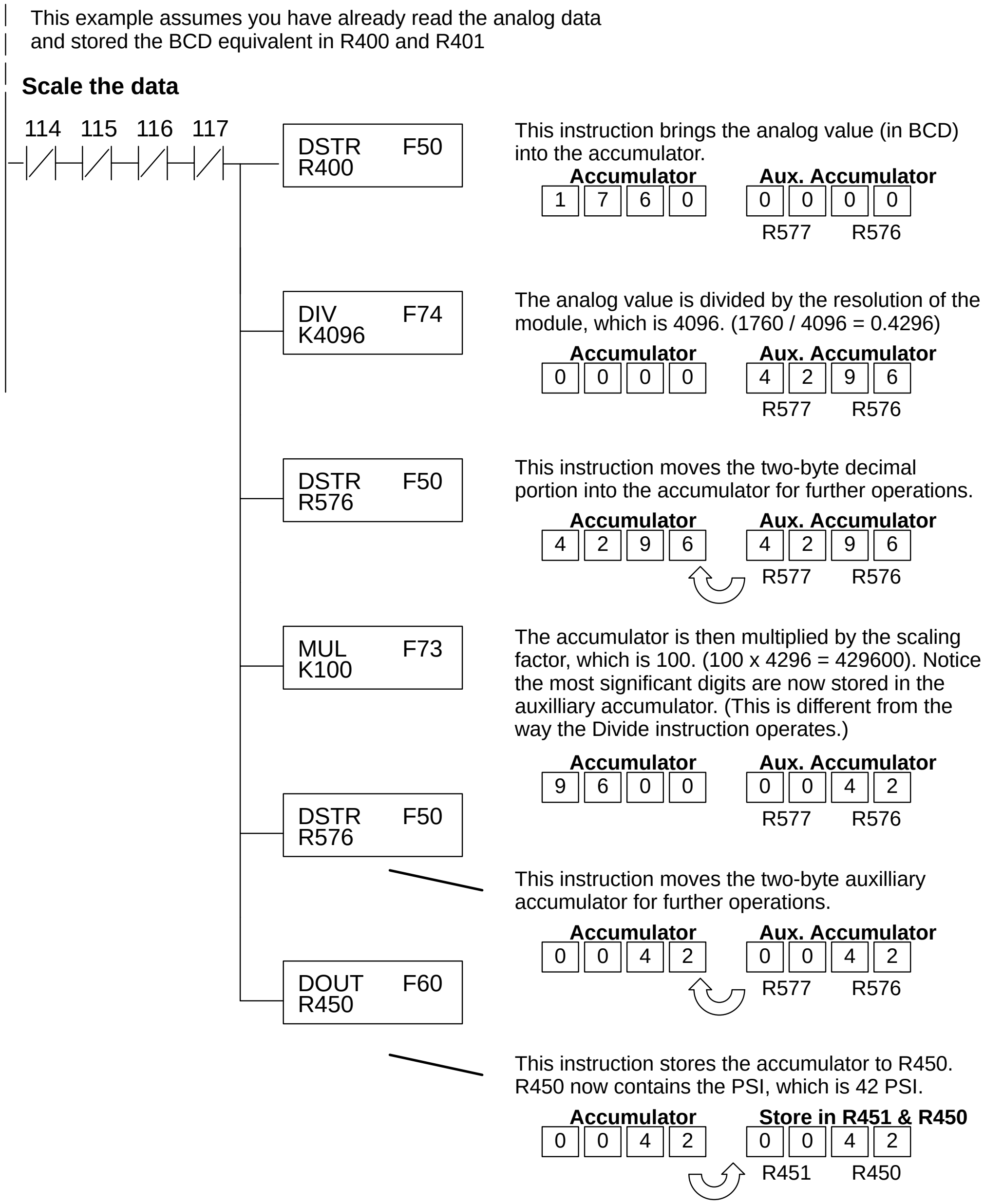
$$\downarrow$$

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

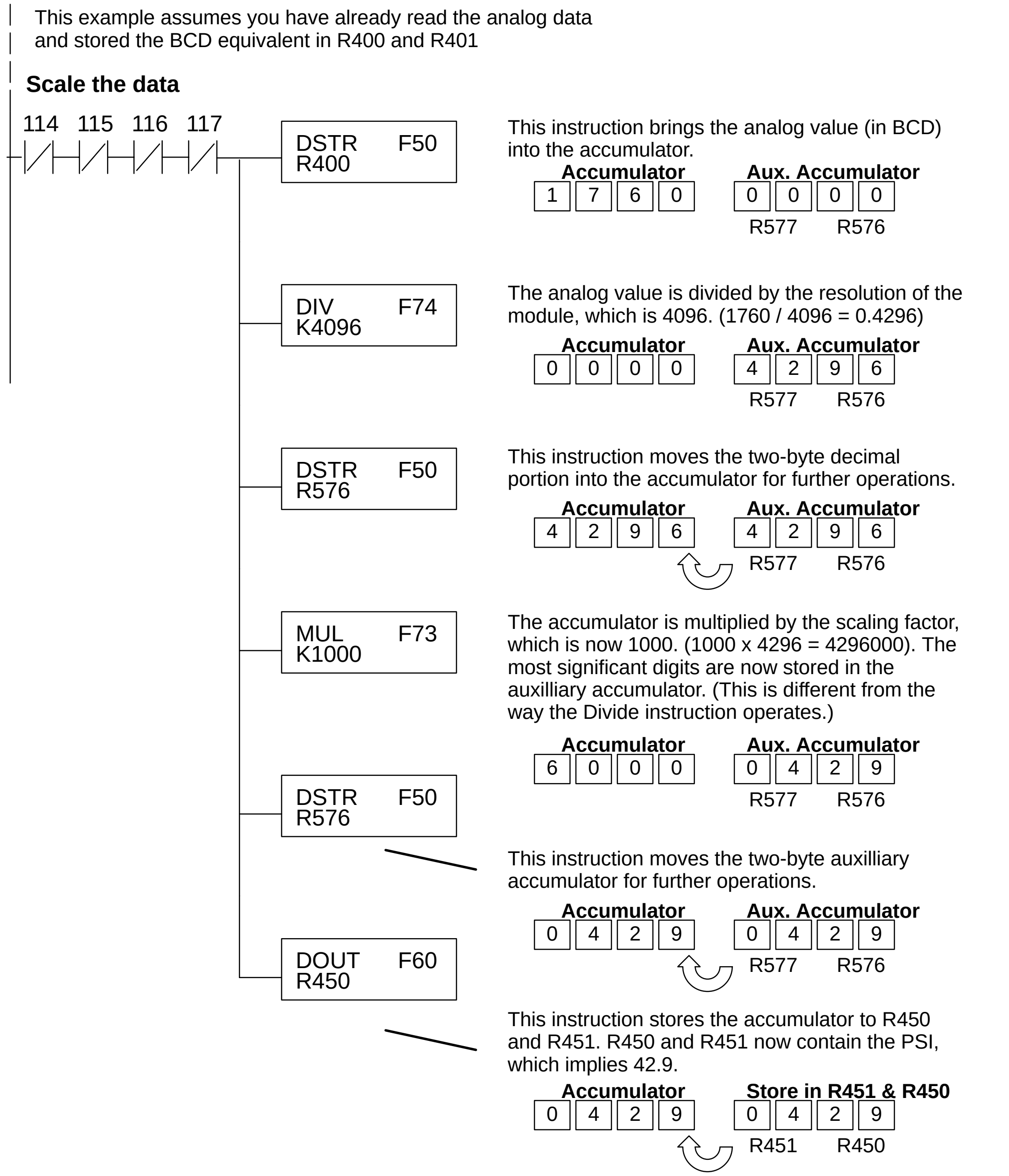
$$\downarrow$$

$$\text{Units} = 42.9$$

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. The active channel indicator inputs are all off when channel 1 data is being read. Of course, if you were using a different channel, you would use the active channel indicator point combination that corresponds to the channel you were using.

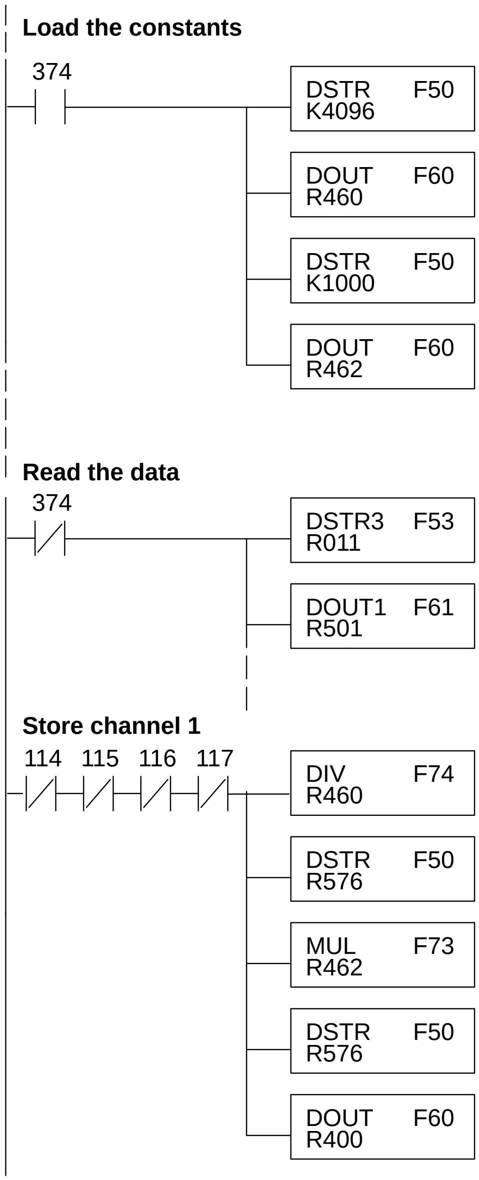


You probably noticed the previous example yielded 42 PSI when the real value should have been 42.9 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve added another digit to the scale. Instead of a scale of 100, we’re using 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 R460 and R461.

These two instructions load the high limit of the Engineering unit scale (constant of 1000) into R462 and R463. 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 on every scan.

Temporarily store the bits to Register 501.

The analog value is divided by the resolution of the module, which is stored in R460.

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 R462.

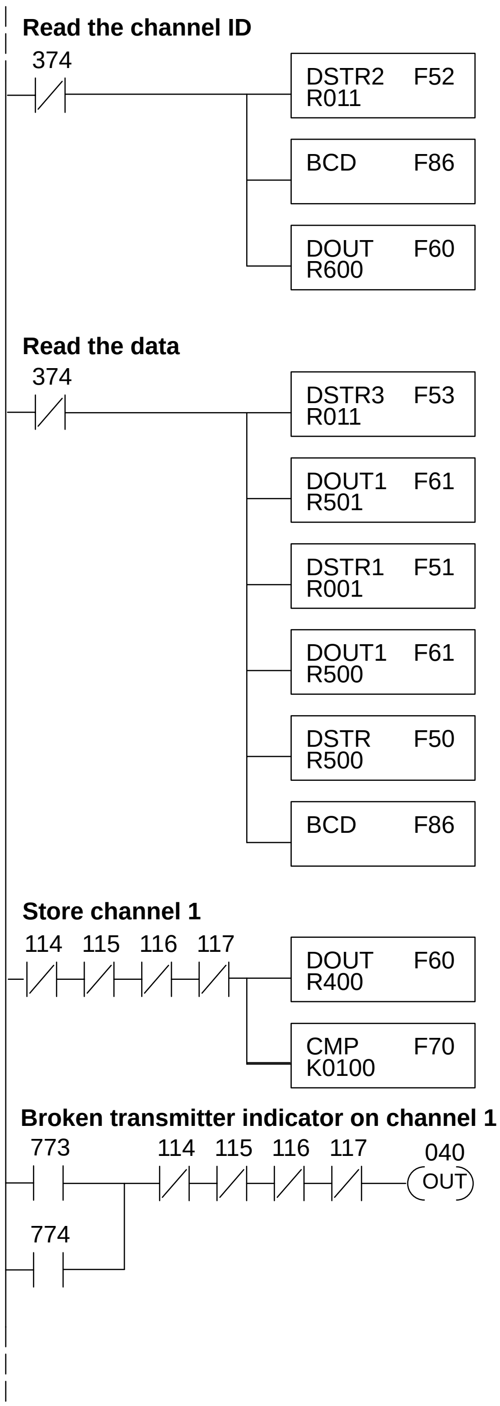
This instruction moves most significant digits (now stored in the auxilliary accumulator) into the regular accumulator for further operations.

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

Broken Transmitter Detection

If you use 4–20mA signals you can easily check for broken transmitter conditions. Since you have to use the 0–20mA range and the lowest signal for the 4–20mA transmitter is 4mA, the lowest digital value for the signal is not 0, but instead is 819.

If the transmitter is working properly the smallest value you should ever see is 819. If you see a value of less than about 750 (allowing for tolerance), then you know the transmitter is broken.



This rung loads the channel ID bits into the accumulator from Register 011 on every scan.

Convert the channel ID status to BCD. We'll use relational contacts later to make the chanel selection much easier.)

Store the channel ID in R600.

This rung loads the four most significant data bits into the accumulator from Register 011 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 DOUT instruction copies the accumulator data to R400 and R401. Since the data is still in the accumulator, we can compare it against a constant. Since the minimum value for a 4mA signal is 819 (minus the module tolerance), we can choose a value for the compare. We picked 100, but you could choose something else from 0 to about 750.

Flags 773 and 774 are used with the Compare instruction. In this example if the analog value is less than or equal to 100, then output 040 is turned on.

You may want to latch 040 to catch intermittent broken transmitters.

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.

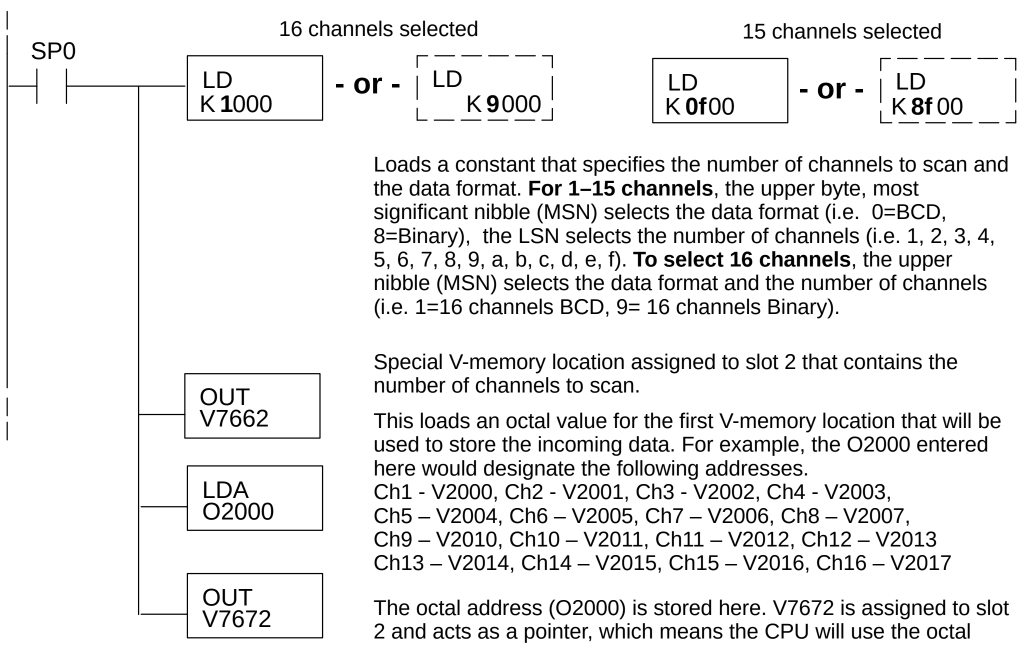
NOTE: Do not use the pointer method and the PID PV auto transfer from I/O module function together for the same module. If using PID loops, use the pointer method and ladder logic code to map the analog input data into the PID loop table.

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. **For 1–15 channels**, the upper byte, most significant nibble (MSN) selects the data format (i.e. 0=BCD, 8=Binary), the LSN selects the number of channels (i.e. 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f). **To select 16 channels**, the upper nibble (MSN) selects the data format and the number of channels (i.e. 1=16 channels BCD, 9= 16 channels Binary).

Special V-memory location assigned to slot 2 that contains the number of channels to scan.

This loads an octal value for the first V-memory location that will be used to store the incoming data. For example, the O2000 entered here would designate the following addresses.
Ch1 - V2000, Ch2 - V2001, Ch3 - V2002, Ch4 - V2003,
Ch5 – V2004, Ch6 – V2005, Ch7 – V2006, Ch8 – V2007,
Ch9 – V2010, Ch10 – V2011, Ch11 – V2012, Ch12 – V2013
Ch13 – V2014, Ch14 – V2015, Ch15 – V2016, Ch16 – V2017

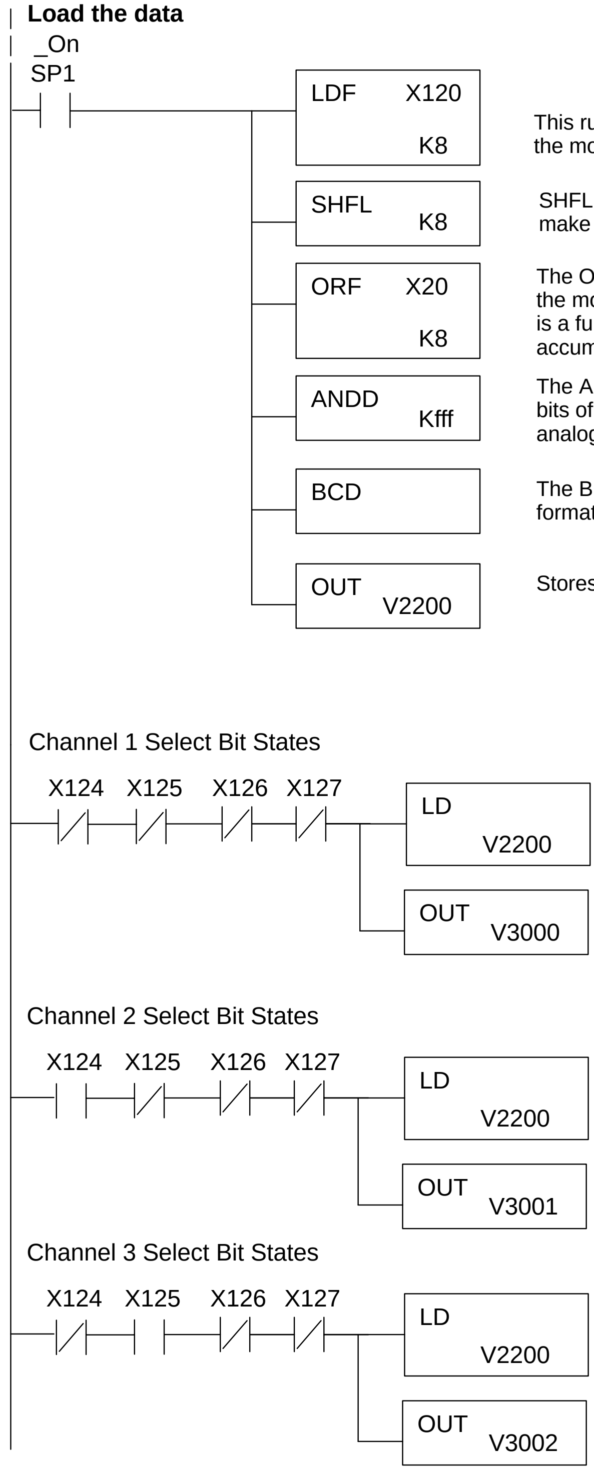
The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming data.

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 Conventional DL305 Base

The example below shows how to read multiple channels on an F3–08AD Analog module in the 20–27/120–127 address slot. This module must be placed in a 16 bit slot in order to work.



This rung loads the upper byte of analog data from the module.

SHFL K8 shifts the data to the left eight places to make room for the lower byte of data.

The ORF X20 brings the lower byte of data from the module into the accumulator. At this time there is a full word of data from the analog module in the accumulator.

The ANDD Kfff masks off the four most significant bits of data from the word. This leaves the actual analog value.

The BCD command converts the data to BCD format.

Stores the data in V2200.

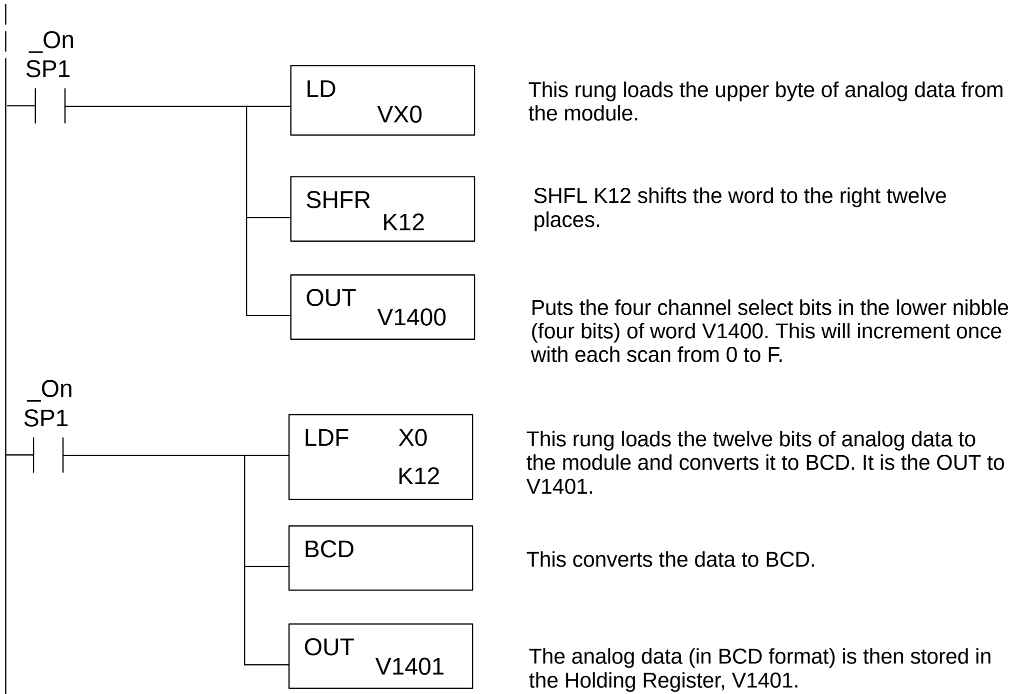
This sends channel one analog data to V3000 when bits X124, X125, X126 and X127 are as shown.

This sends channel two analog data to V3001 when bits X124, X125, X126 and X127 are as shown.

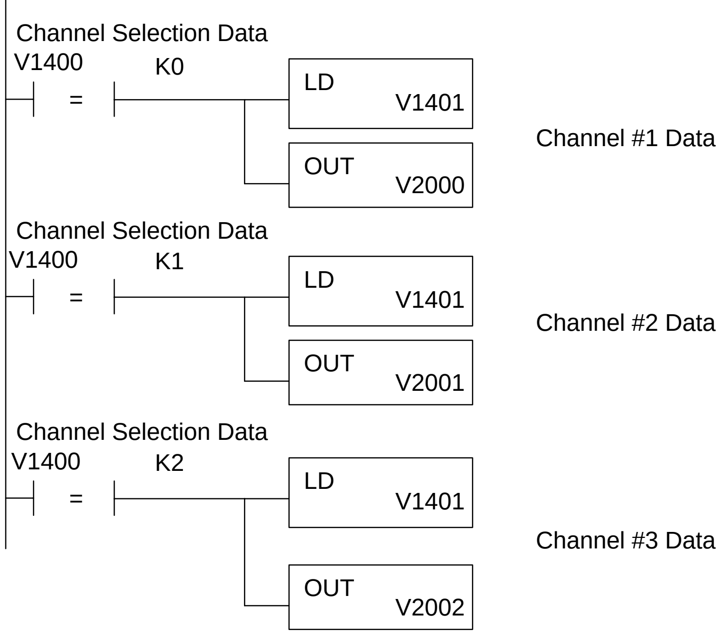
This sends channel two analog data to V3002 when bits X124, X125, X126 and X127 are as shown.

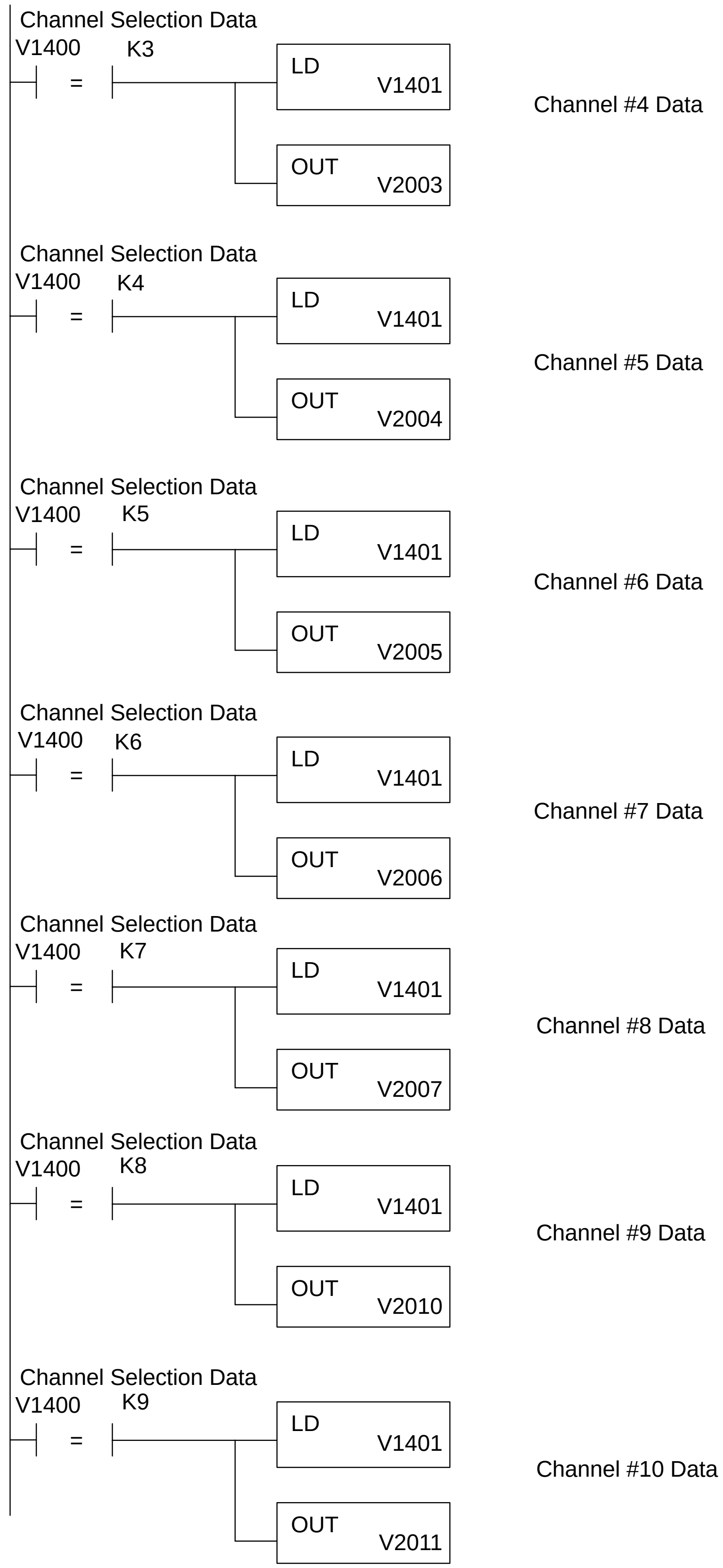
Multiplexing: DL350 with a D3–XX–1 Base

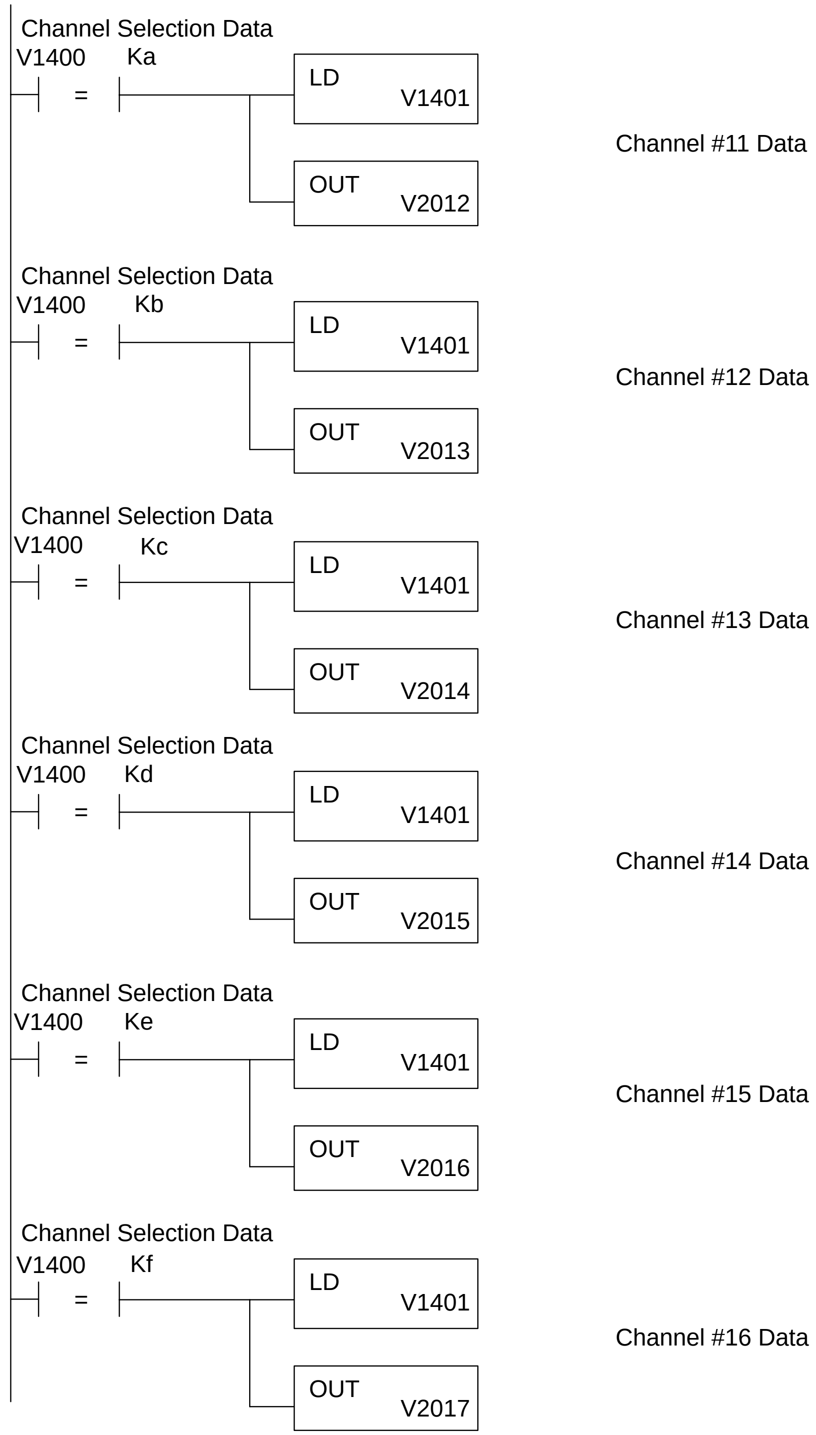
The example below shows how to read multiple channels on an F3–16AD Analog module in the X0 address slot of the D3–XX–1 base. If any expansion bases are used in the system, they must all be D3–xx–1 to be able to use this example. Otherwise, the conventional base addressing must be used.



Rungs 3–18 compare the count of the chennel select bits. When the corresponding bits are true, the channel data for that channel is stored in the proper V-memory location. For sixteen channels of analog data, the module will require sixteen scans in order to update all channels.




Channel Selection Data
V1400 K3
=
LD V1401
OUT V2003
Channel #4 Data
Channel Selection Data
V1400 K4
=
LD V1401
OUT V2004
Channel #5 Data
Channel Selection Data
V1400 K5
=
LD V1401
OUT V2005
Channel #6 Data
Channel Selection Data
V1400 K6
=
LD V1401
OUT V2006
Channel #7 Data
Channel Selection Data
V1400 K7
=
LD V1401
OUT V2007
Channel #8 Data
Channel Selection Data
V1400 K8
=
LD V1401
OUT V2010
Channel #9 Data
Channel Selection Data
V1400 K9
=
LD V1401
OUT V2011
Channel #10 Data


Channel Selection Data
V1400 Ka
=
LD V1401
OUT V2012
Channel #11 Data
Channel Selection Data
V1400 Kb
=
LD V1401
OUT V2013
Channel #12 Data
Channel Selection Data
V1400 Kc
=
LD V1401
OUT V2014
Channel #13 Data
Channel Selection Data
V1400 Kd
=
LD V1401
OUT V2015
Channel #14 Data
Channel Selection Data
V1400 Ke
=
LD V1401
OUT V2016
Channel #15 Data
Channel Selection Data
V1400 Kf
=
LD V1401
OUT V2017
Channel #16 Data

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

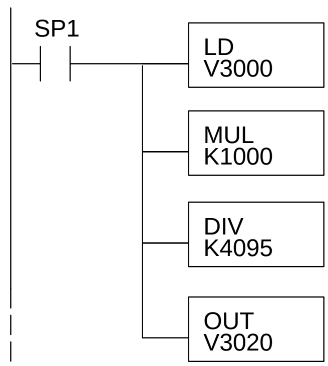
L = low limit of the engineering unit range

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.



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

Multiply the accumulator by 1000 (to start the conversion).

Divide the accumulator by 4095.

Store the result in V3020.

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	$A = \frac{10D}{4095}$	$D = \frac{4095}{10}A$
0 to 12V	$A = \frac{12D}{4095}$	$D = \frac{4095}{12}A$
0 to 20mA (or 4–20mA)	$A = \frac{20D}{4095}$	$D = \frac{4095}{20}A$
0 to 1V	$A = \frac{1D}{4095}$	$D = \frac{4095}{1}A$
0 to 0.1V	$A = \frac{0.1D}{4095}$	$D = \frac{4095}{0.1}A$
0 to 0.01V	$\ell A = \frac{0.01D}{4095}$	$D = \frac{4095}{0.01}A$

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.

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

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

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

$$D = 3276$$

6

D3–02DA 2–Channel Analog Output

In This Chapter. . . .

Module Specifications

The following table provides the specifications for the D3–02DA Analog Output Module. Review these specifications to make sure the module meets your application requirements.

Number of Channels	2 (independent)
Output Ranges	0 – 10V, 4 – 20 mA
Resolution	8 bit (1 in 256)
Output Type	Single ended
Output Impedance	.5Ω maximum, voltage output
Output Current	10 mA minimum, voltage output @ 10 VDC
Load Impedance	550Ω maximum, 5Ω minimum, current output
Total Inaccuracy	±0.4% maximum at 25° C
Accuracy vs. Temperature	±50 ppm / °C maximum
Conversion Time	100µs maximum (2 channels/scan)
Power Budget Requirement	80 mA @9V
External Power Supply	24 VDC, ±10%, 170 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

Analog Output Configuration Requirements

The D3–02DA Analog Output 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.

Connecting the Field Wiring

Wiring Guidelines

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

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the module or the power supply return (0V). *Do not* ground the shield at both the module and the transducer.
- 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 D3–02DA requires a separate power supply. Choose a supply that meets the following requirements: 24 VDC ±10%, Class 2, 170mA current (or greater, depending on the number of modules being used.)

Load Requirements

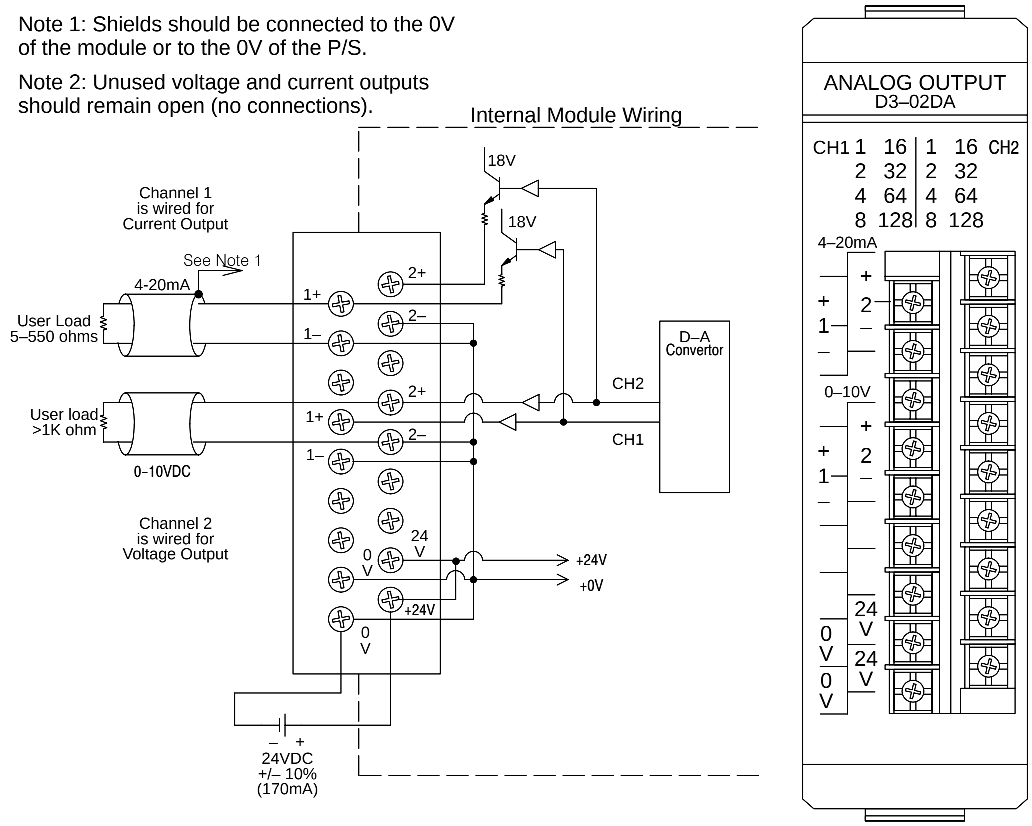
Each channel can be wired independently for a voltage or current transducer.

- Current transducers must have an impedance between 5 and 550 ohms
- Voltage transducers must have an impedance greater than 1K ohms.

Removable Connector

The D3–02DA module has a removable connector to make wiring easier. Simply remove the retaining screws and gently pull the connector from the module.

Wiring Diagram

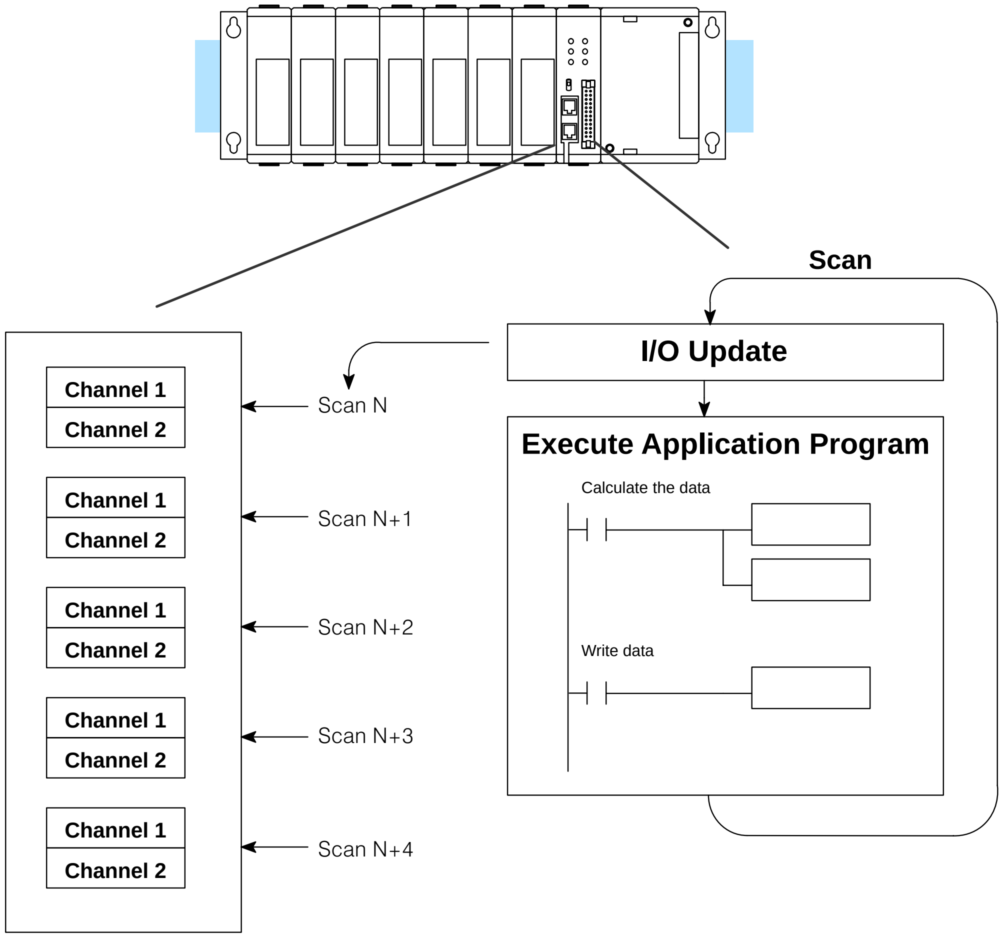


Module Operation

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

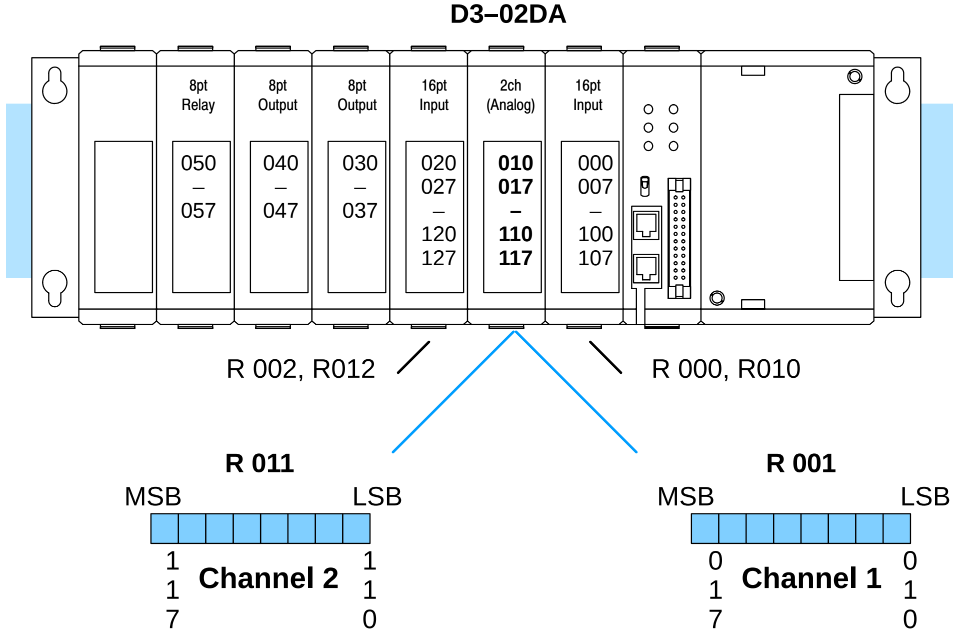
The D3–02DA module updates both channels in the same scan. The control program updates the two channels of this module independent of each other and each channel does not have to be refreshed on each scan.



Understanding the I/O Assignments

You may recall the D3–02DA module appears to the CPU as a 16-point module. These 16 points provide 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.

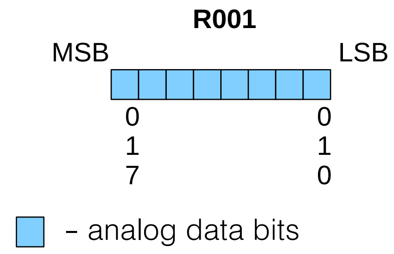


Within these two word locations, the individual bits represent specific information about the analog signal.

Analog Data Bits

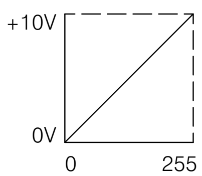
The first register contains the data for channel one (R001). The second register contains the data for channel two (R011).

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

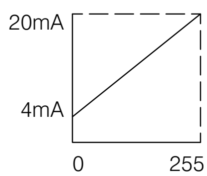


Since the module has 8-bit resolution, the analog signal is converted into 256 “pieces” ranging from 0 – 255 (2^8). For example, with a 0 to 10V scale, a 0V signal would be 0, and a 10V signal would be 255. This is equivalent to a a binary value of 0000 0000 to 1111 1111, or 00 to FF hexadecimal. The following diagram shows how this relates to each signal range.

0V – 10V



4 – 20mA



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.

$$\text{Resolution} = \frac{H - L}{255}$$

H = high limit of the signal range

L = low limit of the signal range

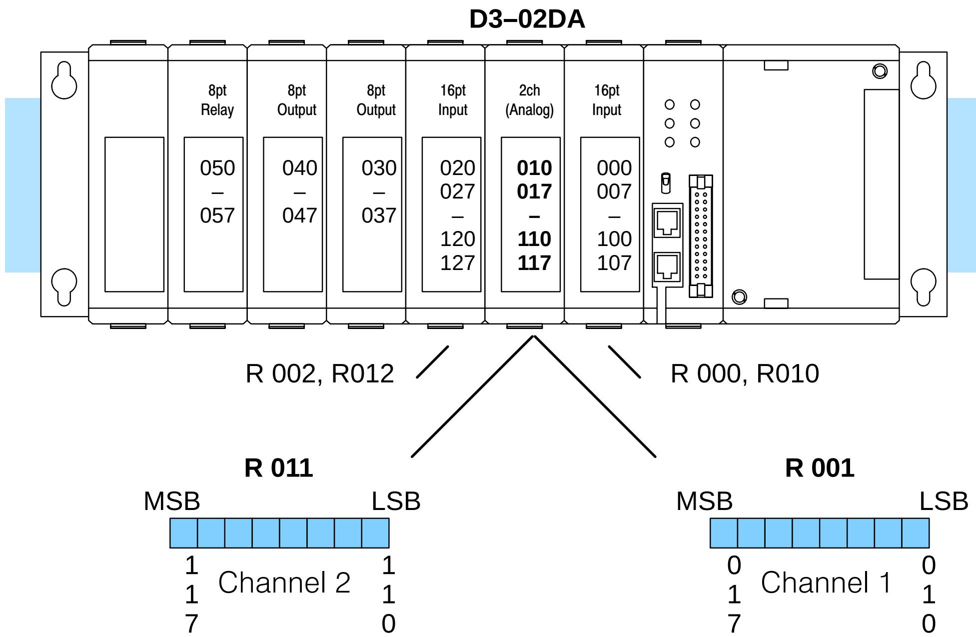
Range	Highest Signal	Lowest Signal	Smallest Change
0 to 10V	10V	0V	39 mV
4 to 20mA	20mA	4mA	62.5 µA

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

As mentioned earlier, you can update either channel or both channels during the same scan. Since the module does not have any channel select bits, you just simply determine the location of the data word and send the data word to the output module whenever you need to update the data.



Calculating the Digital Value

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

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

$$A = 256 \frac{U}{H - L}$$

A = Analog value (0 – 255)

U = Engineering Units

H = high limit of the Engineering unit range

L = low limit of the Engineering unit range

The following example shows how you would use Engineering Units to obtain the digital value to represent pressure (PSI) from 0 to 100. This example assumes you want to obtain a pressure of 42 PSI, which is slightly less than half scale.

$$A = 256 \frac{U}{H - L}$$

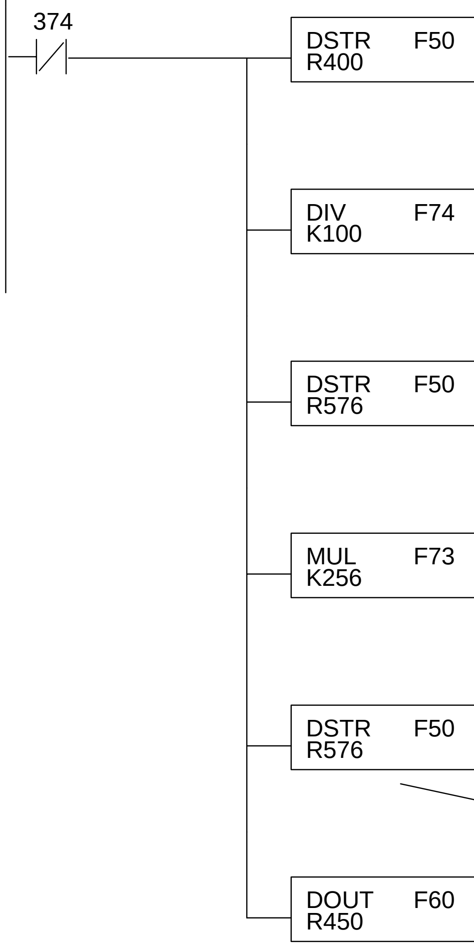
$$A = 256 \frac{42}{100 - 0}$$

$$A = 107.5 \text{ (or 108)}$$

Here's how you would write the program to perform the Engineering Unit conversion. This example assumes you have calculated or loaded the engineering unit value and stored it in R400. Also, you have to perform this for both channels if you're using different data for each channel.

This example assumes you have already loaded the Engineering unit value in R400.

Scale the data



This instruction loads Engineering unit value into the accumulator.

Accumulator				Aux. Accumulator (R577)		Aux. Accumulator (R576)	
0	0	4	2	0	0	0	0

The Engineering unit value is divided by the Engineering unit range (42/100=.42). In this case the range is 100. (100 – 0 = 100)

Accumulator				Aux. Accumulator (R577)		Aux. Accumulator (R576)	
0	0	0	0	4	2	0	0

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator (R577)		Aux. Accumulator (R576)	
4	2	0	0	4	2	0	0

The accumulator is then multiplied by the module resolution, which is 256. (256 x 4200 = 1075200). Notice the most significant digits are now stored in the auxilliary accumulator. (This is different from the Divide instruction operation.)

Accumulator				Aux. Accumulator (R577)		Aux. Accumulator (R576)	
5	2	0	0	0	1	0	7

This instruction moves the two-byte auxilliary accumulator for further operations.

Accumulator				Aux. Accumulator (R577)		Aux. Accumulator (R576)	
0	1	0	7	0	1	0	7

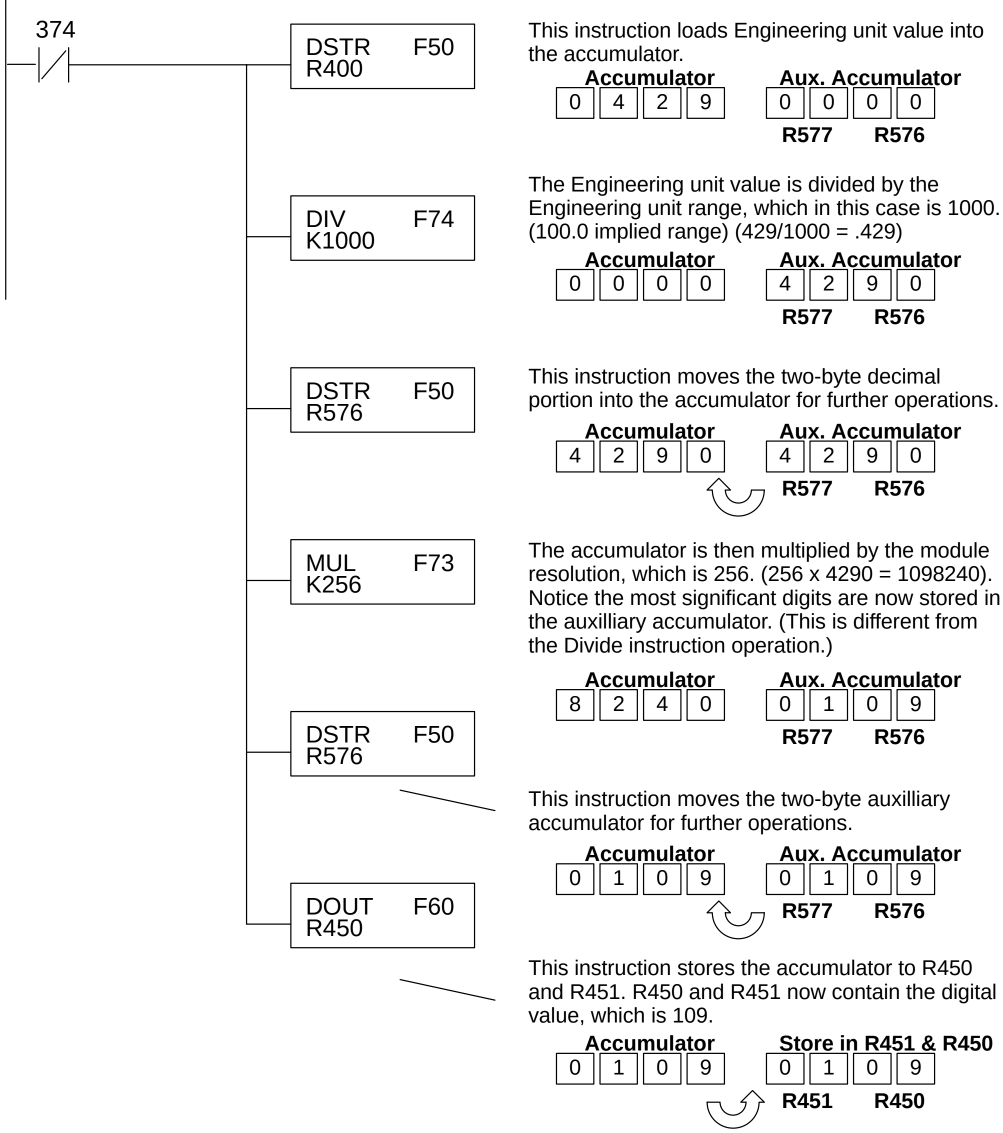
This instruction stores the accumulator to R451 and R450. R451 and R450 now contain the digital value, which is 107.

Accumulator				Store in R451 & R450 (R451)		Store in R451 & R450 (R450)	
0	1	0	7	0	1	0	7

There will probably be times when you need more precise control. For example, maybe your application requires 42.9 PSI, not just 42 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve entered 429 as the Engineering unit value and 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 loaded the Engineering unit value in R400.

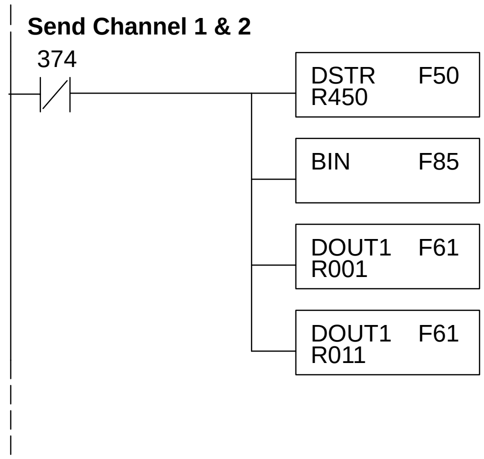
Scale the data



Sending the Same Data to Both Channels

In some applications, you'll want to send the same output values to both channels. The following program example shows how to send the digital values to the module.

This example assumes you have already loaded the Engineering unit value in R450 and R451.



This rung loads the data into the accumulator on every scan.

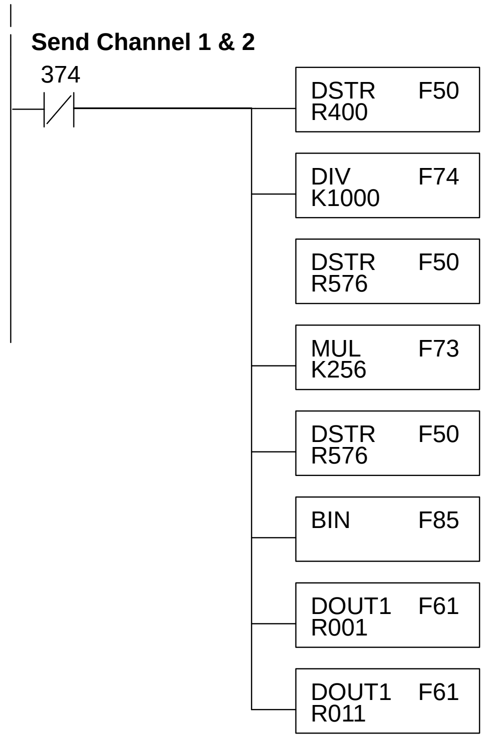
Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the accumulator data to the Register that corresponds to channel 1, which is R001.

Send the accumulator data to the Register that corresponds to channel 2, which is R011.

If you want a shorter program, just combine the data scaling and output instructions.

This example assumes you have already loaded the Engineering unit value in R400.



This instruction loads Engineering unit value into the accumulator.

The Engineering unit value is divided by the Engineering unit range, which in this case is 1000. (100.0 implied range)

This instruction moves the two-byte decimal portion into the accumulator for further operations.

The accumulator is then multiplied by the module resolution, which is 256.

This instruction moves the two-byte auxilliary accumulator into the regular accumulator.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

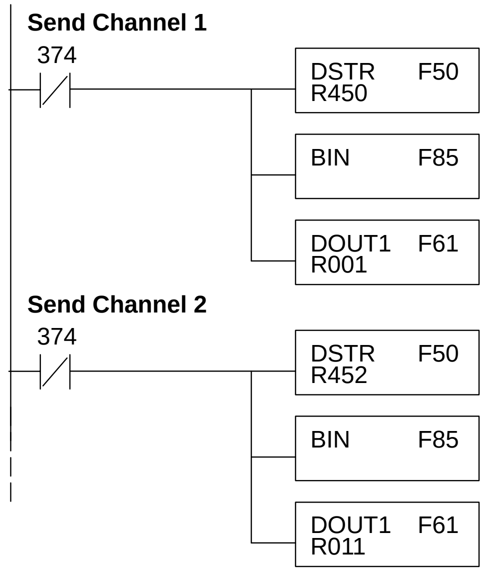
Send the accumulator data to the Register that corresponds to channel 1, which is R001.

Send the accumulator data to the Register that corresponds to channel 2, which is R011.

Sending Specific Data to Each Channel

In this case, the example logic is setup to send different data to each channel. Of course, you would have to have separate routines to calculate the output data and you would have to store the different values in separate registers.

This example assumes you have already loaded the Engineering unit value for Channel 1 in R450 and R451 and the data for Channel 2 in R452 and R453.



This rung loads the data for channel 1 into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the accumulator data to the Register that corresponds to channel 1, which is R001.

This rung loads the data for channel 2 into the accumulator on every scan.

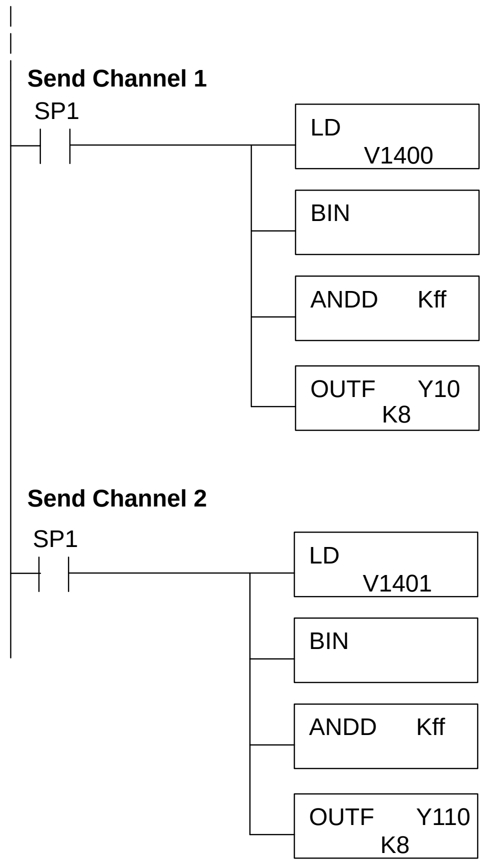
Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the accumulator data to the Register that corresponds to channel 2, which is R011.

Writing the Control Program (DL350)

Multiplexing: DL350 with a Conventional DL305 Base

This example assumes the module is in the Y10–17 / Y110–117 slot of a 305 conventional base. In this example V1400 contains the BCD data for channel 1 and V1401 contains the data for channel 2.



This rung loads the data for channel 1 into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Masks off the 256 bit analog data for the module.

Send the accumulator data to the bits that correspond to channel 1.

This rung loads the data for channel 2 into the accumulator on every scan.

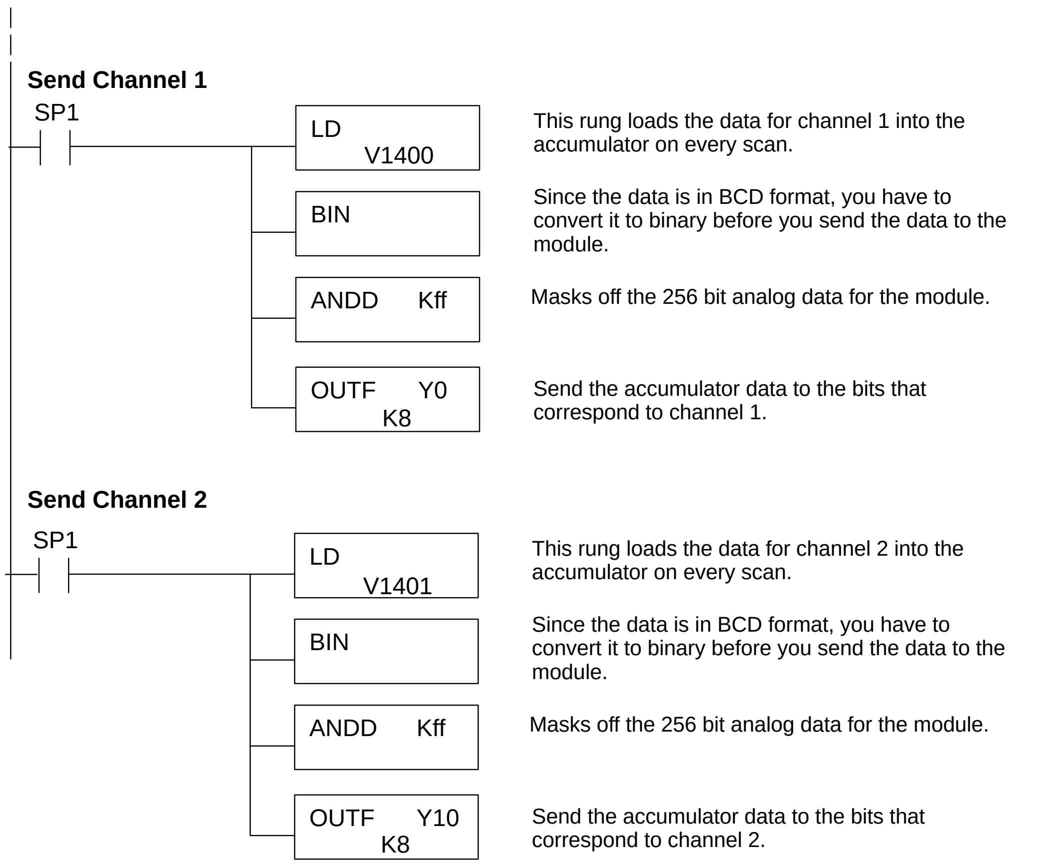
Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Masks off the 256 bit analog data for the module.

Send the accumulator data to the bits that correspond to channel 2.

Multiplexing: DL350 with a D3–xx–1 Base

This example assumes the module is in Y0 address slot of a D3–xx–1 base . In this example V1400 contains the BCD data for channel 1 and V1401 contains the data for channel 2. 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.



Analog and Digital Value Conversions

Sometimes it is helpful to be able to quickly convert between the voltage or current 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 ...
0 to 10V	$A = \frac{10D}{255}$	$D = \frac{255}{10} A$
4 to 20mA	$A = \frac{16D}{255} + 4$	$D = \frac{255}{16}(A - 4)$

For example, if you are using the 4–20mA range and you know you need a 10mA signal level, you would use the following formula to determine the digital value that should be sent to the module.

$$D = \frac{255}{16}(A - 4)$$

$$D = \frac{255}{16}(10mA - 4)$$

$$D = (15.93)\ (6)$$

$$D = 96$$

Calculating the Digital Value

Your program must calculate the digital value to send to the analog module. 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.

$$A = U \frac{255}{H - L}$$

A = Analog value (0 – 255)

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.

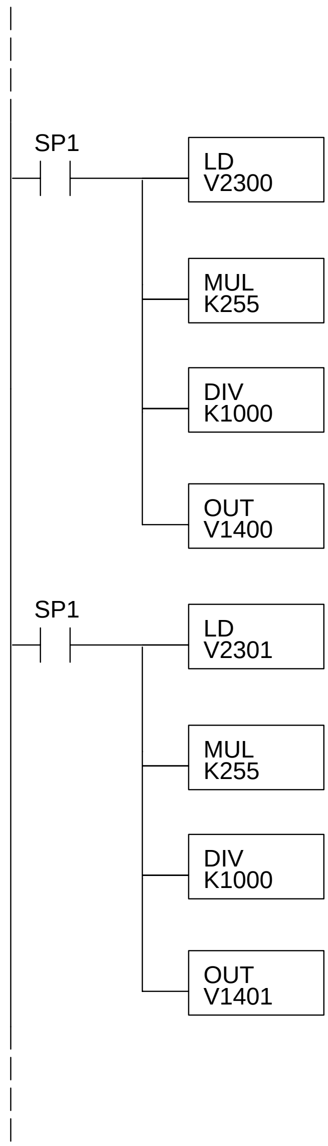
$$A = 10U \frac{255}{10(H - L)}$$

$$A = 494 \frac{255}{1000 - 0}$$

$$A = 126$$

The example program below shows how you would write the program to perform the engineering unit conversion. This example assumes you have calculated or loaded the engineering unit values in BCD and stored them in V2300 and V2301 for channels 1 and 2 respectively.

NOTE: The DL350 offers various instructions that allow you to perform math operations using BCD format. It is easier to perform math calculations in BCD and then convert the value to binary before sending the data to the module.



The LD instruction loads the engineering units used with channel 1 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 255 (to start the conversion).

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

Store the BCD result in V1400 (the actual steps to write the data were shown earlier).

The LD instruction loads the engineering units used with channel 2 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 255 (to start the conversion).

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

Store the BCD result in V1401 (the actual steps to write the data were shown earlier).

7

F3–04DA–1 4-Channel Analog Output

In This Chapter. . . .

Module Specifications

The following table provides the specifications for the F3–04DA–1 Analog Output Module. Review these specifications to make sure the module meets your application requirements.

Number of Channels	4
Output Ranges	0 – 5V, 0 – 10V, 4 – 12 mA, 4 – 20 mA (source)
Resolution	12 bit (1 in 4096)
Output Type	Single ended (one common)
Output Impedance	0.5Ω typical, voltage output
Output Current	5 mA source, 2.5 mA sink (voltage)
Short-circuit Current	40 mA typical, voltage output
Load Impedance	1KΩ maximum, current output 2KΩ minimum, voltage output
Linearity Error	± 1 count (± 0.03% maximum)
Maximum Inaccuracy at 77 °F (25 °C)	± 0.6% of span, current output ± 0.2% of span, voltage output
Accuracy vs. Temperature	± 50 ppm / °C maximum
Conversion Time	30μS maximum
Power Budget Requirement	144 mA @9V, 108 mA @ 24V
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 Output Configuration Requirements

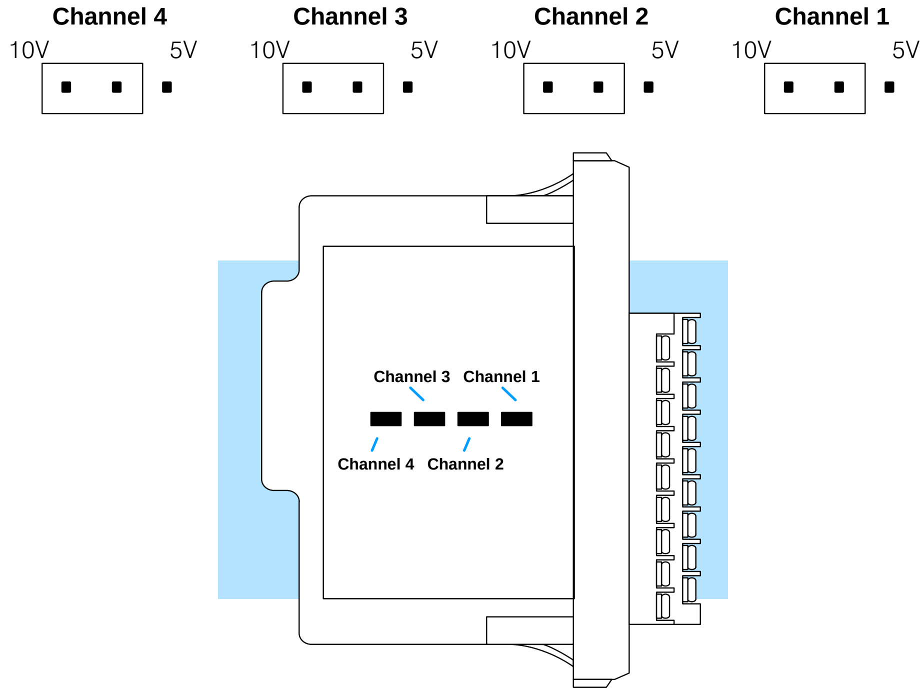
The F3–04DA–1 Analog Output appears as a 16-point module. The module can be installed in any slot configured for 16 points. See the DL305 User Manual for details on using 16 point modules in DL305 systems. The limitation on the number of analog modules are:

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

Setting the Module Jumpers

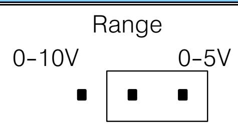
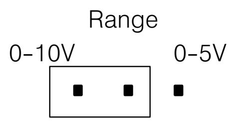
Jumper Locations

The module is set at the factory for a 0–10V signal on all four channels. (This range also allows 4–20 mA operation since there are separate I and V wiring terminals.) If this is acceptable you do not have to change any of the jumpers. The following diagram shows the jumper locations.



Selecting Output Signal Ranges

The jumper is set from the factory to allow either 0–10V or 4–20mA operation on all channels. In addition, you can select 0 – 5V or 4 – 12 mA operation by moving the jumper. (Only channel 1 is used in the example, but all channels must be set.)

Signal Range	Jumper Settings
0 to +5 VDC 4 to 12 mA	
0 VDC to +10 VDC 4 to 20 mA	

Connecting the Field Wiring

Wiring Guidelines

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

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the module or the power supply return (0V). *Do not* ground the shield at both the module and the transducer.
- 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 F3–04DA–1 receives all power from the base. A separate power supply is not required.

Load Requirements

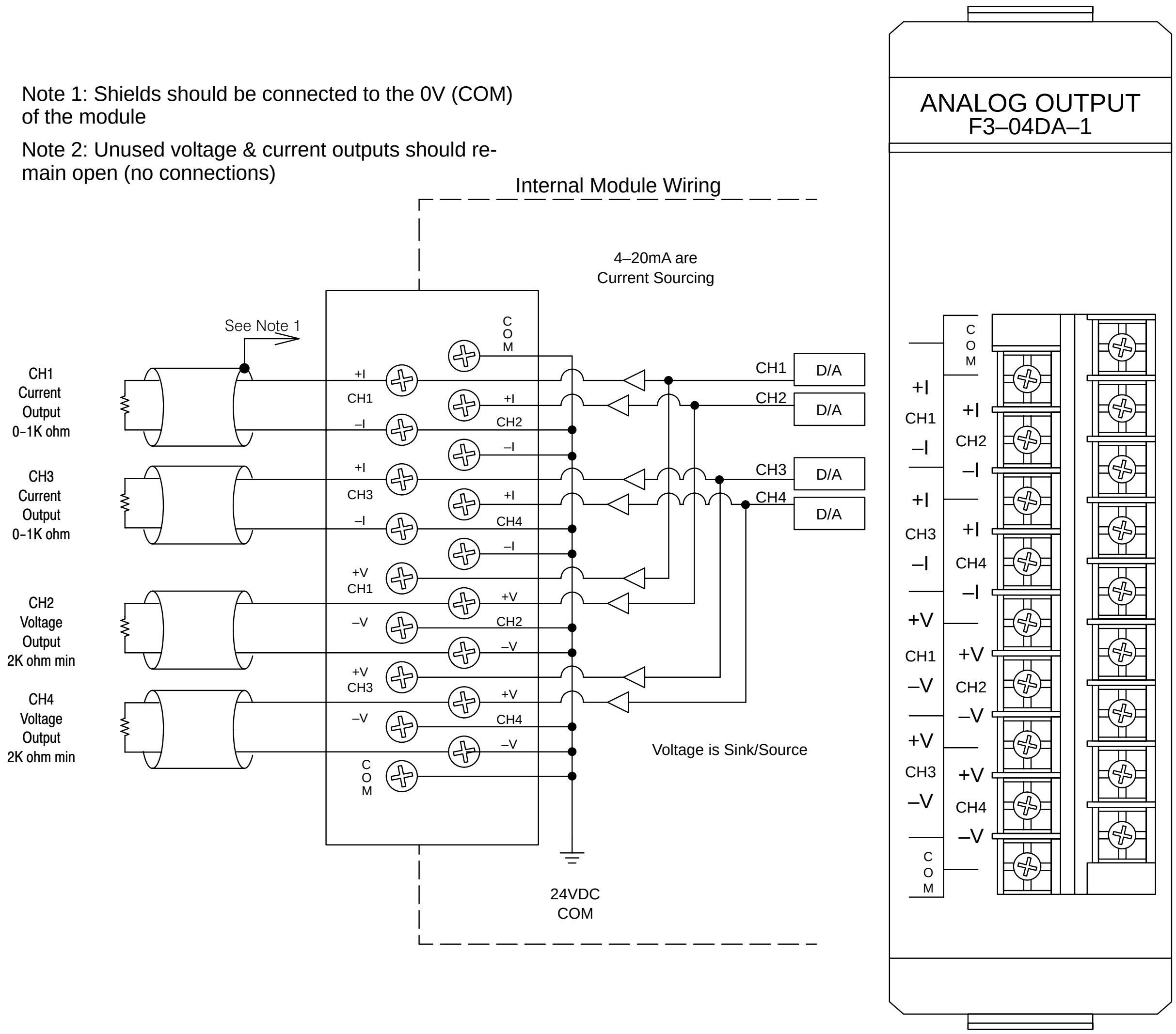
Each channel can be wired independently for a voltage or current transducer.

- Current transducers must have an impedance less than 1K ohm.
- Voltage transducers must have an impedance greater than 2K ohms.

Removable Connector

The F3–04DA–1 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

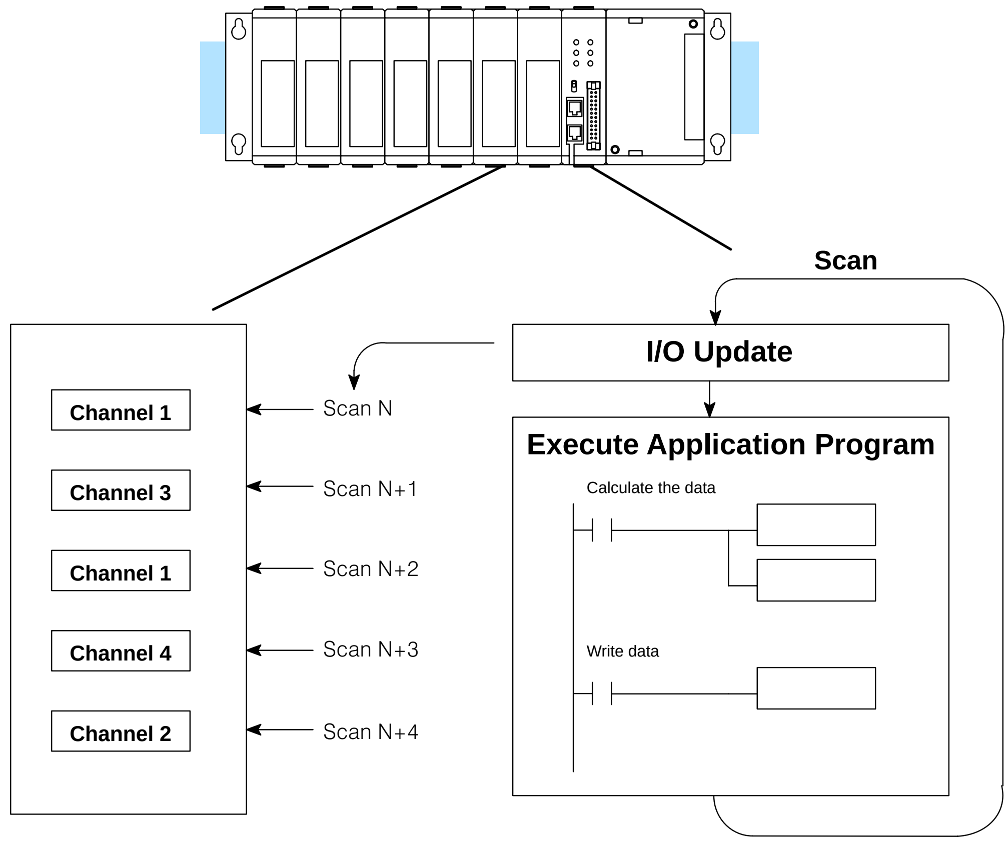


Module Operation

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 F3–04DA–1 module can update one channel per CPU scan. Your RLL program selects which channel to update, so you have complete flexibility to solve your application requirements.

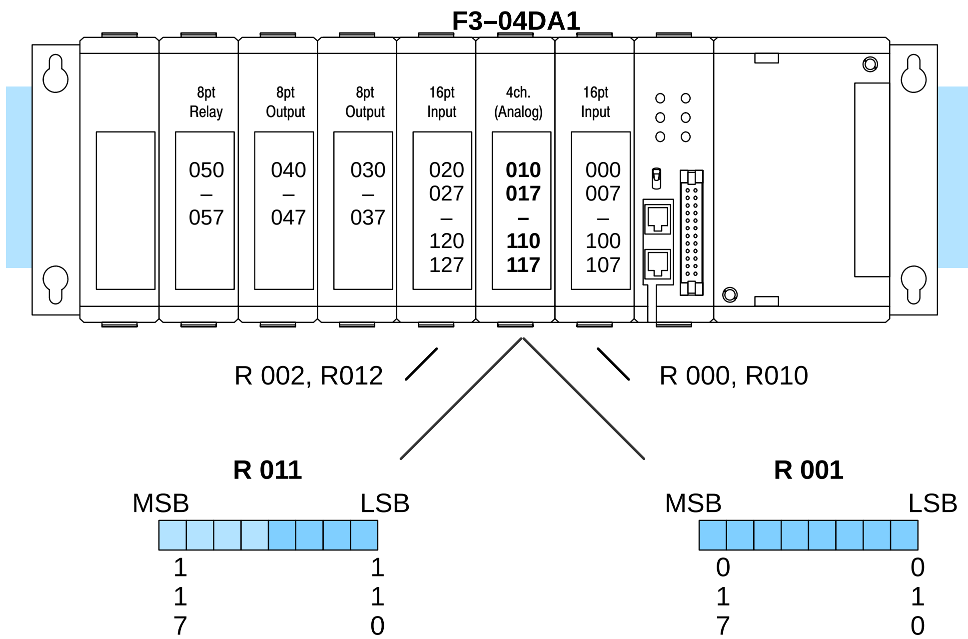


Understanding the I/O Assignments

You may recall the F3–04DA–1 module appears to the CPU as a 16-point module. These 16 points provide:

- the digital representation of the analog signal.
- identification of the channel to receive the data.

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.

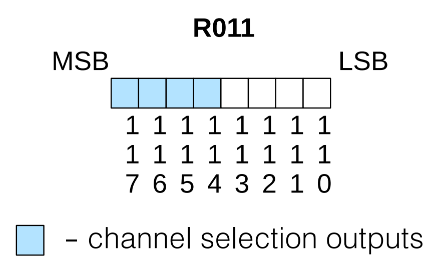


Within these two word locations, the individual bits represent specific information about the analog signal.

Channel Selection Inputs

The last four points of the upper register are used as outputs to tell the module which channel to update. In our example, when output 114 is on, channel 1 will be updated. Here's how the outputs are assigned.

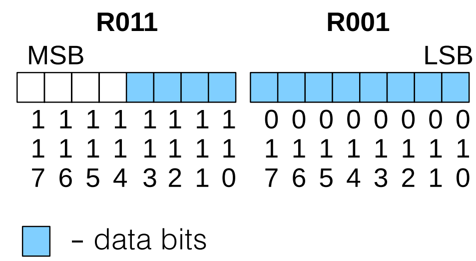
Output	Channels
114	1
115	2
116	3
117	4



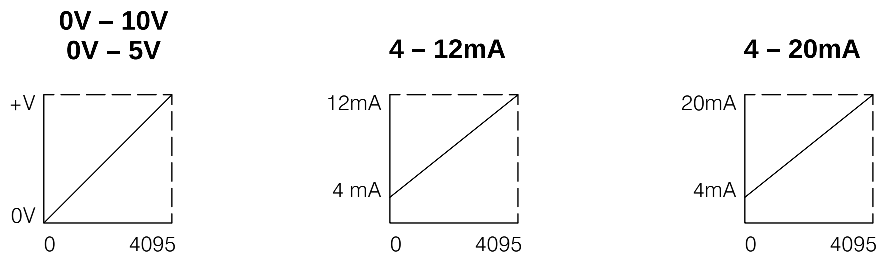
Analog Data Bits

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 possibly 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

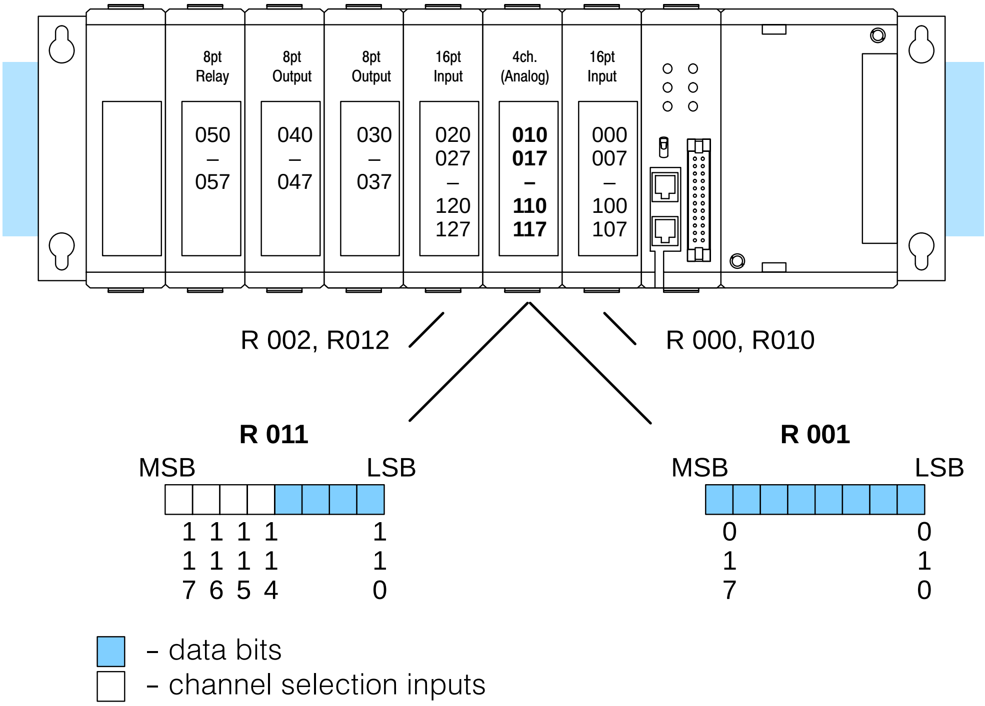
Range	Highest Signal	Lowest Signal	Smallest Change
0 to 5V	5V	0V	1.22 mV
0 to 10V	10V	0V	2.44 mV
4 to 12mA	12mA	4mA	1.95 µA
4 to 20mA	20mA	4mA	3.91 µA

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

As mentioned earlier, you can use the channel selection bits to determine which channels will be updated. The following diagram shows the location for both the channel selection bits and data bits.



Calculating the Digital Value

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

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

$$A = 4096 \frac{U}{H - L}$$

A = Analog value (0 – 4095)

U = Engineering Units

H = high limit of the Engineering unit range

L = low limit of the Engineering unit range

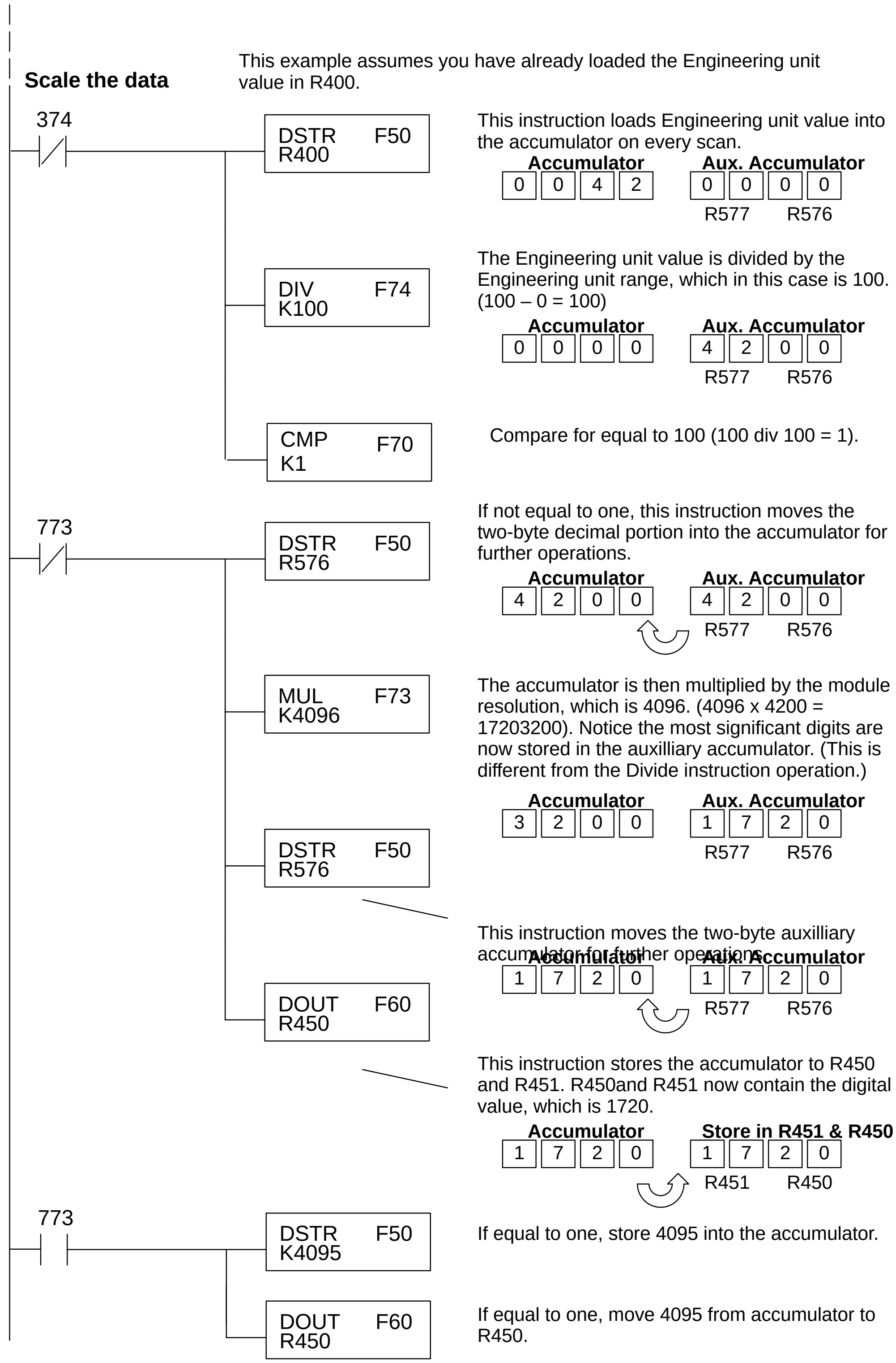
The following example shows how you would use Engineering units to obtain the digital value to represent pressure (PSI) from 0 to 100. This example assumes you want to obtain a pressure of 42 PSI, which is slightly less than half scale.

$$A = 4096 \frac{U}{H - L}$$

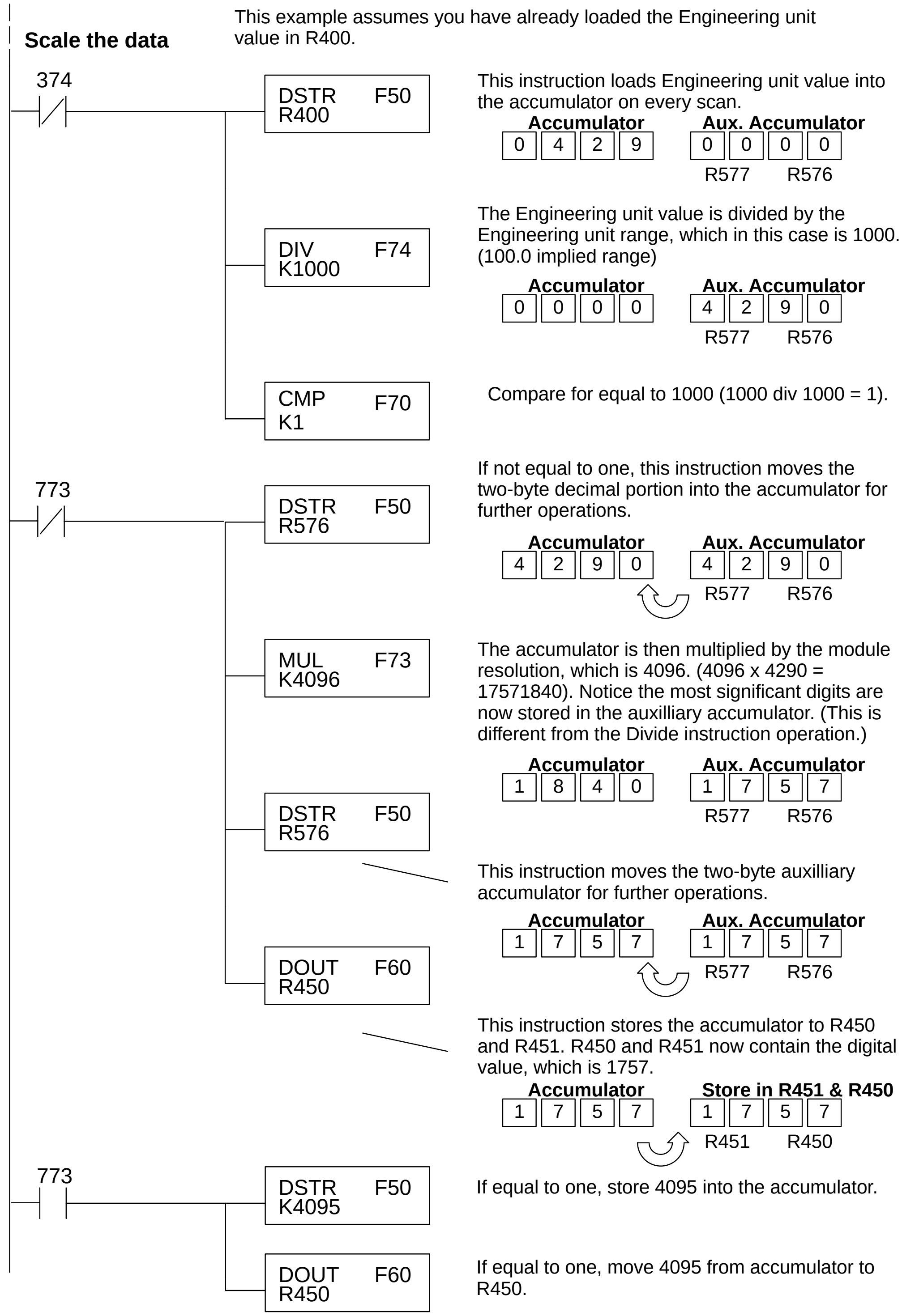
$$A = 4096 \frac{42}{100 - 0}$$

$$A = 1720$$

Here's how you would write the program to perform the Engineering unit conversion. This example assumes you have calculated or loaded the engineering unit value and stored it in R400. Also, you have to perform this for all channels if you're using different data for each channel.



There will probably be times when you need more precise control. For example, maybe your application requires 42.9 PSI, not just 42 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve entered 429 as the Engineering unit value and 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.

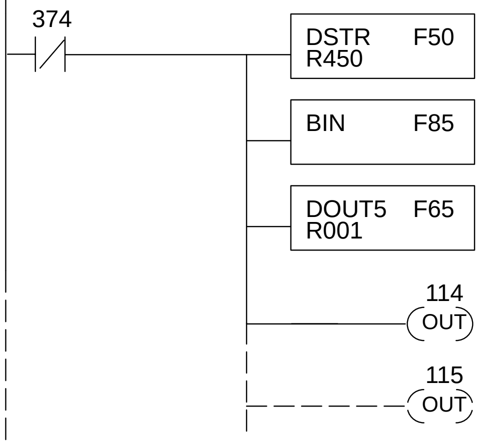


Sending Data to a Single Channel

The following program example shows how to send the digital value to a single channel.

This example assumes you have already loaded the Engineering unit value in R450 and R451.

Send Channel 1



This rung loads the data into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the accumulator data to the Register that corresponds to the module, which is R001.

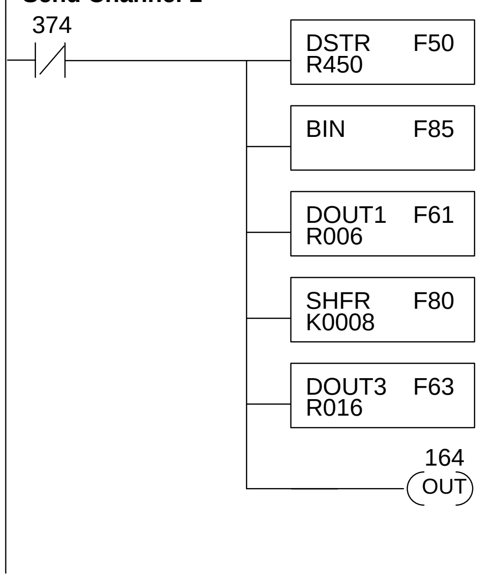
Indicate the channel to update. In this case, channel 1 is being updated.

To update other channels with the same output data, simple add the channel selection outputs for the additional channels.

If you install the F3–04DA–1 in the slot corresponding to registers 6 and 16, you have to make a slight program adjustment. This is because the DOUT5 instruction is not supported for this slot.

This example assumes you have already loaded the Engineering unit value in R450 and R451.

Send Channel 1



This rung loads the data into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the 8 least significant data bits to the first Register that corresponds to the module which is R006.

Shift the 4 most significant data bits to the right 8 places. (The data is still in the accumulator).

Send the 4 most significant data bits to the second Register that corresponds to the module which is R016.

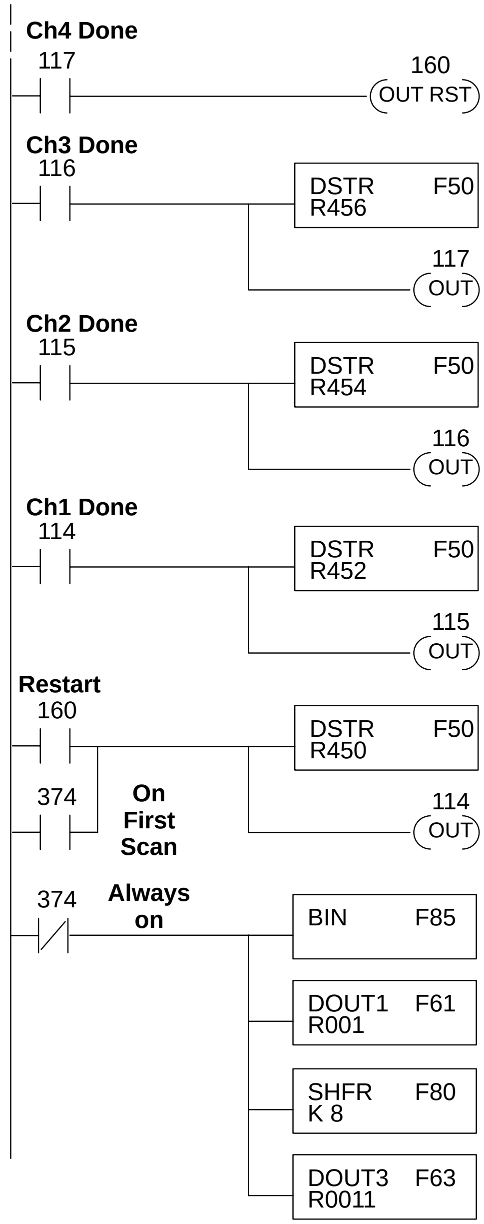
Indicate the channel to update. In this case, channel 1 is being updated.

Sequencing the Channel Updates

This example shows how to send digital values to the module when you have more than one channel. This example will automatically update all four channels over four scans. The example is fairly simple and will work in most all situations, but there are instances where problems can occur. The logic must be active on the first CPU scan and all subsequent scans. If the logic gets stopped or disabled for some reason, there is no way to restart it. If you're using an RLL*PLUS* (Stage) program, put this logic in an initial stage that is always active. Also, you should avoid using the this example if you require the analog output logic to be used inside a Master Control Relay field of control. Even if you do not have a need for the MCR, you can still accidentally disable the analog output logic by inadvertently writing to the multiplexing control relays with an operator interface or intelligent module, such as an ASCII BASIC module, etc.

Sequencing Example

The following program example shows how to send the digital values to multiple channels. With this program, all channels will be updated within four scans. You must use the rungs in the order shown, but you can include them anywhere in the program.



When channel 4 has been updated, 160 restarts the update sequence.

When channel 3 has been updated, this rung loads the data for channel 4 into the accumulator. By turning on 117, this triggers the channel update. (Since 117 is also used as an input, this results in a one-shot.)

When channel 2 has been updated, this rung loads the data for channel 3 into the accumulator. By turning on 116, this triggers the channel update. (Since 116 is also used as an input, this results in a one-shot.)

When channel 1 has been updated, this rung loads the data for channel 2 into the accumulator. By turning on 115, this triggers the channel update. (Since 115 is also used as an input, this results in a one-shot.)

This rung loads the data for channel 1 into the accumulator. Since 374 is used, this rung automatically executes on the first scan. After that, 160 restarts this rung. If you examine the first rung, you'll notice 160 only comes on after channel 4 has been updated.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module. (You can omit this step if you've already converted the data elsewhere.)

Send the 8 least significant data bits to the first Register that corresponds to the module which is R001.

Shift the 4 most significant data bits to the right 8 places. (The data is still in the accumulator).

Send the 4 most significant data bits to the second Register that corresponds to the module which is R011.

Writing the Control Program (DL350)

Reading Values: Pointer Method and Multiplexing

There are two methods of reading values:

- The pointer method **(all system bases must be D3–xx–1 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 CPU, but for ease of programming it is strongly recommended that you use the pointer method.

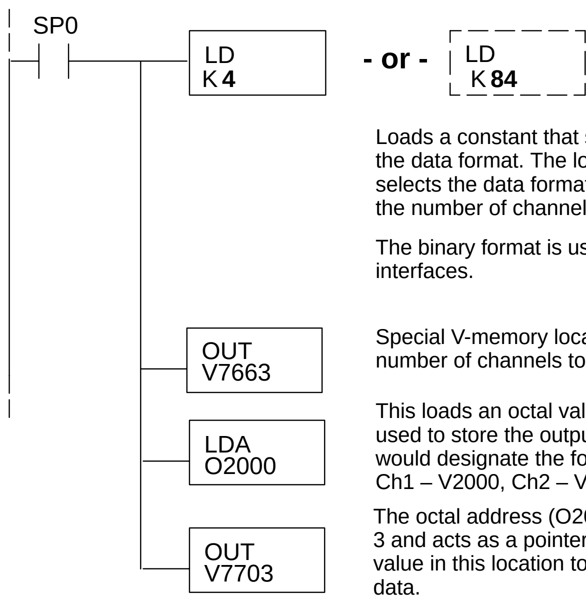
Pointer Method

The DL350 has special V-memory locations assigned to each base slot that greatly simplifies the programming requirements. By using these V-memory locations you can:

- specify the number of channels to update.
- specify where to obtain the output data.

NOTE: Do not use the pointer method and the PID Control Output auto transfer to 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 output data from the PID loop to the output module memory location(s).

The following program example shows how to set up these locations. Place this rung anywhere in the ladder program, or in the initial stage when using stage programming.



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

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

Special V-memory location assigned to slot 3 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 output data. For example, the O2000 entered here would designate the following addresses.
Ch1 – V2000, Ch2 – V2001, ch3 – V2002, ch4 – V2003

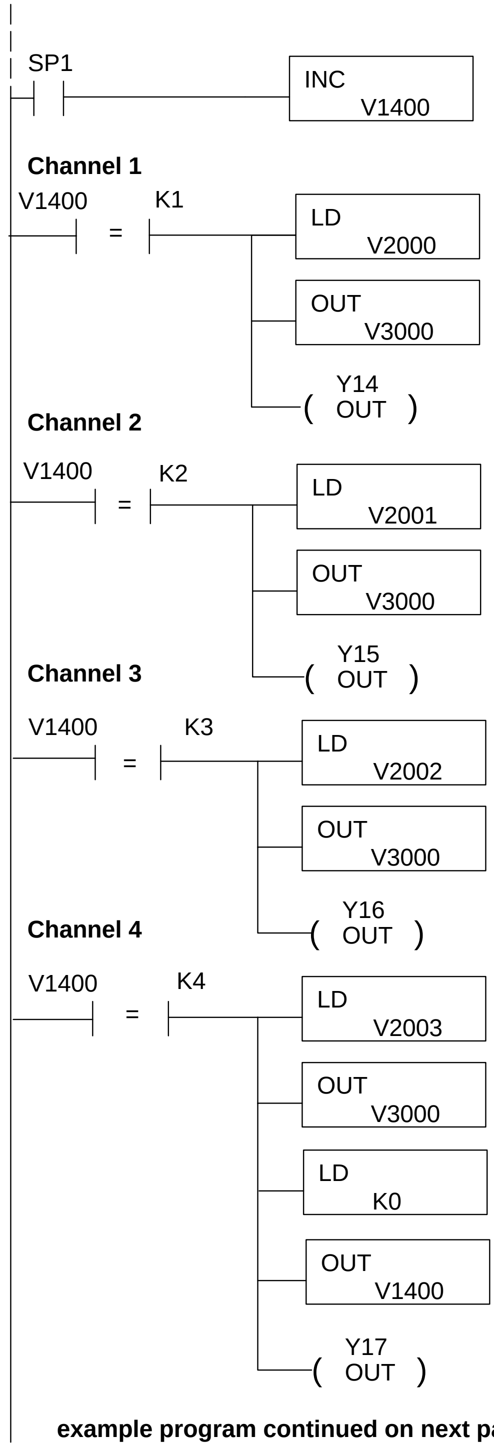
The octal address (O2000) is stored here. V7703 is assigned to slot 3 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the output data.

The table shows the special V-memory locations used with the DL350. Slot 0 (zero) is the module next to the CPU. Remember, the CPU only examines the pointer values at these locations after a mode transition. The pointer method is supported on expansion bases (all bases must be D3–xx–1) up to a total of 8 slots away from the DL350. The pointer method is not supported in slot 8 of a 10 slot base.

Analog Output 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	V7700	V7701	V7702	V7703	V7704	V7705	V7706	V7707

Multiplexing: DL350 with a D3–xx–01 Base

This example assumes the module is in Y0 address slot of D3–xx–1 base. In this example V2000 contains the data for channel and V2001 for channel 2, etc. 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.



This rung loads increments V1400 once every scan from 0–4.

This rung loads the data for channel 1 into the accumulator when V1400 = 1.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 1 is Y14.

This rung loads the data for channel 2 into the accumulator when V1400 = 2.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 2 is Y15.

This rung loads the data for channel 3 into the accumulator when V1400 = 3.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 3 is Y16.

This rung loads the data for channel 4 into the accumulator when V1400 = 4.

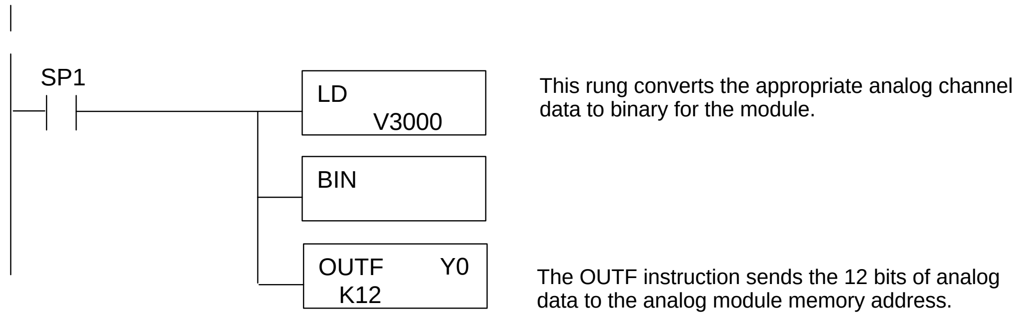
The data is stored in V3000 before sending it to the module.

When V1400 is reset to 0 when V1400 is =4.

The channel select bit for channel 4 is Y17.

example program continued on next page

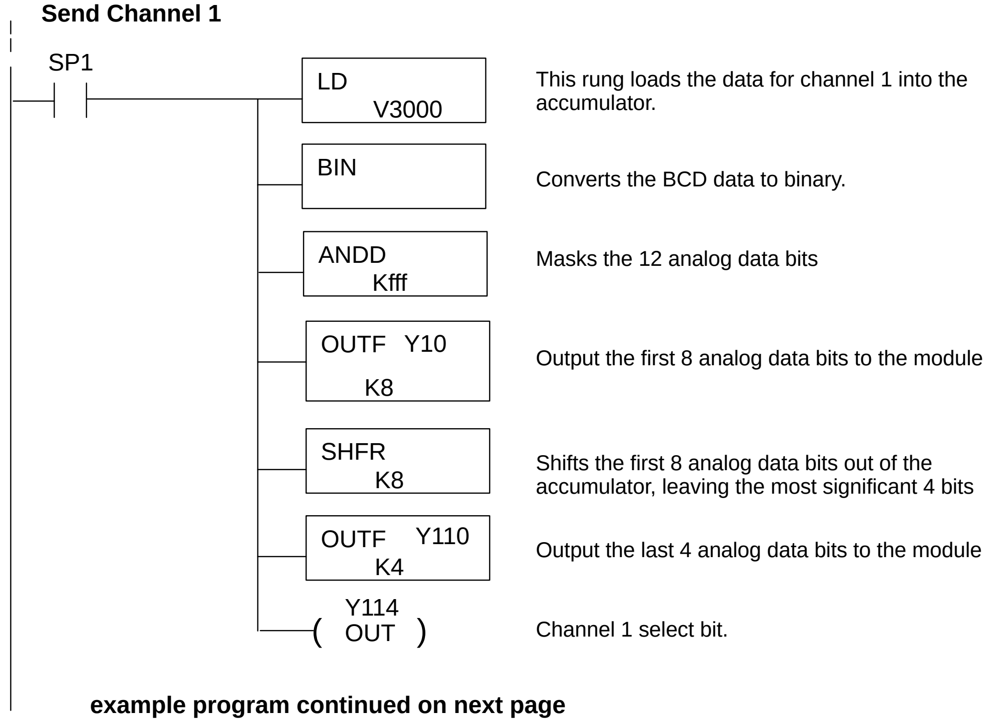
example program continued from previous page



Multiplexing: DL350 with a Conventional DL305 Base

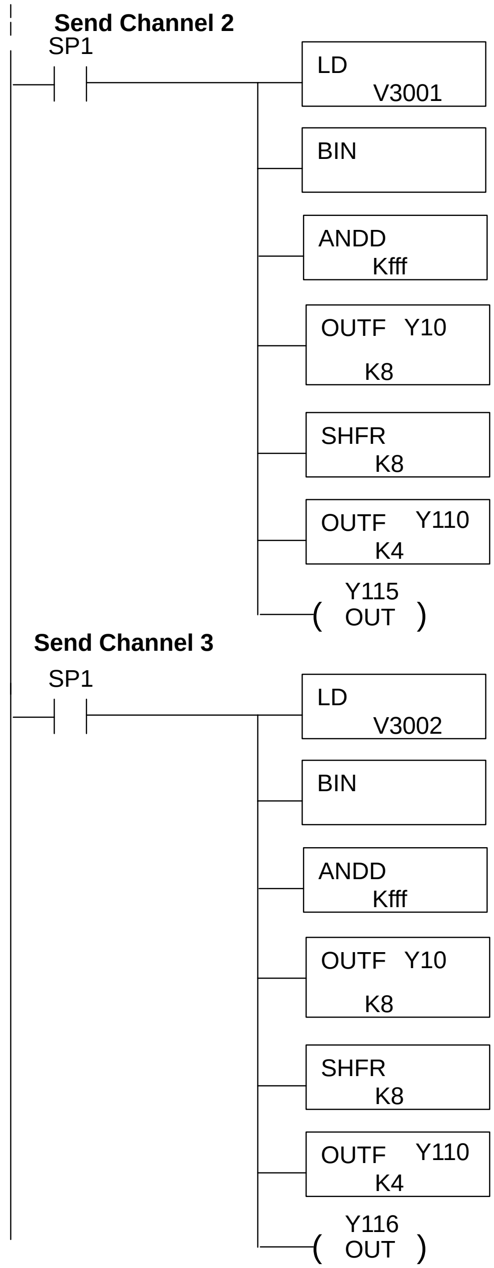
This example assumes the module is in the 10–17 / 110–117 slot of a 305 conventional base. In this example V3000 contains the BCD data for channel 1 and V3001 contains the data for channel 2, etc. One more rung would be necessary for channel 4.

Send Channel 1



example program continued on next page

example program continued from previous page.



This rung loads the data for channel 2 into the accumulator.

Converts the BCD data to binary.

Masks the 12 analog data bits

Output the first 8 analog data bits to the module

Shifts the first 8 analog data bits out of the accumulator, leaving the most significant 4 bits

Output the last 4 analog data bits to the module

Channel 2 select bit.

This rung loads the data for channel 3 into the accumulator.

Converts the BCD data to binary.

Masks the 12 analog data bits

Output the first 8 analog data bits to the module

Shifts the first 8 analog data bits out of the accumulator, leaving the most significant 4 bits

Output the last 4 analog data bits to the module

Channel 3 select bit.

Calculating the Digital Value

Your program must calculate the digital value to send to the analog module. 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.

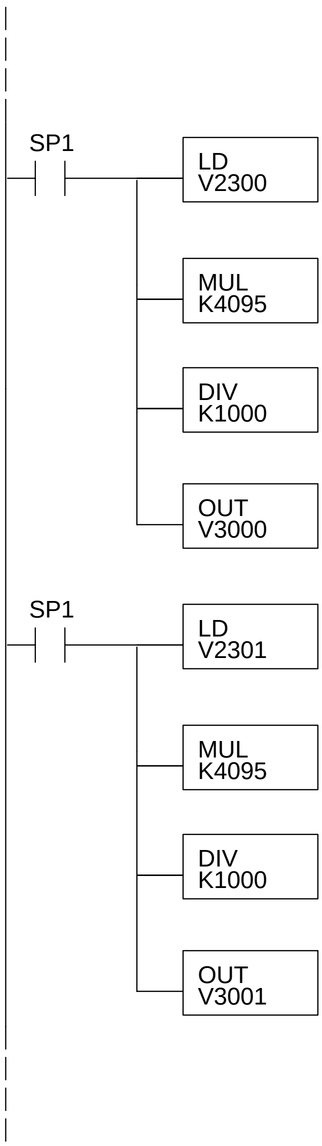
$$A = 10U \frac{4095}{10(H - L)}$$

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

$$A = 2023$$

The example program shows how you would write the program to perform the engineering unit conversion. This example assumes you have calculated or loaded the engineering unit values in BCD and stored them in V2300 and V2301 for channels 1 and 2 respectively.

NOTE: The DL350 offers various instructions that allow you to perform math operations using BCD format. It is easier to perform math calculations in BCD and then convert the value to binary before sending the data to the module.



The LD instruction loads the engineering units used with channel 1 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 4095 (to start the conversion).

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

Store the BCD result in V3000 (the actual steps to write the data were shown earlier).

The LD instruction loads the engineering units used with channel 2 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 4095 (to start the conversion).

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

Store the BCD result in V3001 (the actual steps to write the data were shown earlier).

Analog and Digital Value Conversions

Sometimes it is helpful to be able to quickly convert between the voltage or current 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 ...
0 to 5V	$A = \frac{5D}{4095}$	$D = \frac{4095}{5} A$
0 to 10V	$A = \frac{10D}{4095}$	$D = \frac{4095}{10} A$
4 to 12mA	$A = \frac{12D}{4095} + 4$	$D = \frac{4095}{12}(A - 4)$
4 to 20mA	$A = \frac{16D}{4095} + 4$	$D = \frac{4095}{16}(A - 4)$

For example, if you are using the 4–20mA range and you know you need a 10mA signal level, you would use the following formula to determine the digital value that should be sent to the module.

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

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

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

$$D = 1536$$

8

F3–04DAS 4-Channel Isolated Analog Output

In This Chapter. . . .

Module Specifications

The following table provides the specifications for the F3–04DAS Analog Output Module. Review these specifications to make sure the module meets your application requirements.

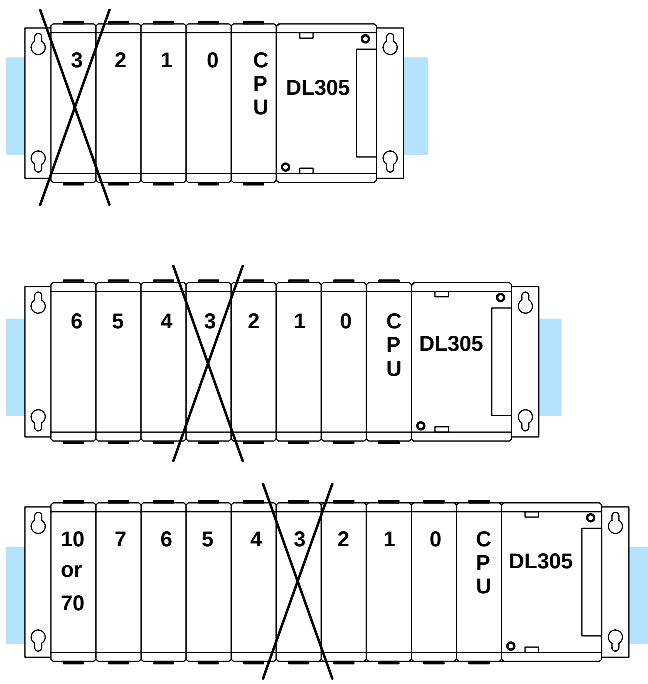
Number of Channels	4
Output Ranges	±5V, ±10V, 0–5V, 0–10V, 1–5V, 0–20 mA, 4–20 mA
Resolution	12 bit (1 in 4096)
Output Type	Isolated, 750 VDC channel-to-channel 750 VDC channel-to-logic
Output Current	±5 mA, voltage output
Short-circuit Current	±20 mA typical, voltage output
Capacitive Load Drive	0.1μF typical, voltage output
Load Impedance	470Ω maximum, current output 2KΩ minimum, voltage output
Isolation Mode Rejection	140 dB at 60Hz
Linearity Error	±1 count (±0.03% maximum)
Calibration Error	±0.15% typical, ±0.75% maximum of span ±10 ppm / °C maximum of full scale
Calibrated Offset Error	±1 count maximum, current output ±5 mV typical, ±50 mV max., voltage output ±0.2 mV typical / °C
Conversion Time	30μS maximum, 1 channel/scan
Power Budget Requirement	154 mA @9V, 145 mA @ 24V (maximum)
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 Output Configuration Requirements

The F3–04DAS Analog Output appears as a 16-point module. The module can be installed in any slot configured for 16 points, but **should not be installed in Slot 3 of any DL305 base**. 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.

WARNING: You should not install this module in Slot 3 of any DL305 base. The module has traces on the edge card connector that may become damaged if the module is repeatedly installed and removed. The solder mask that protects the traces may be scraped off, which may cause a short circuit on the I/O bus. The short circuit can lead to unpredictable system operation or cause damage to the CPU or power supply.



Setting the Module Jumpers

Jumper Locations

The module is set at the factory for a 0–10V signal on all four channels. If this is acceptable you do not have to change any of the jumpers.

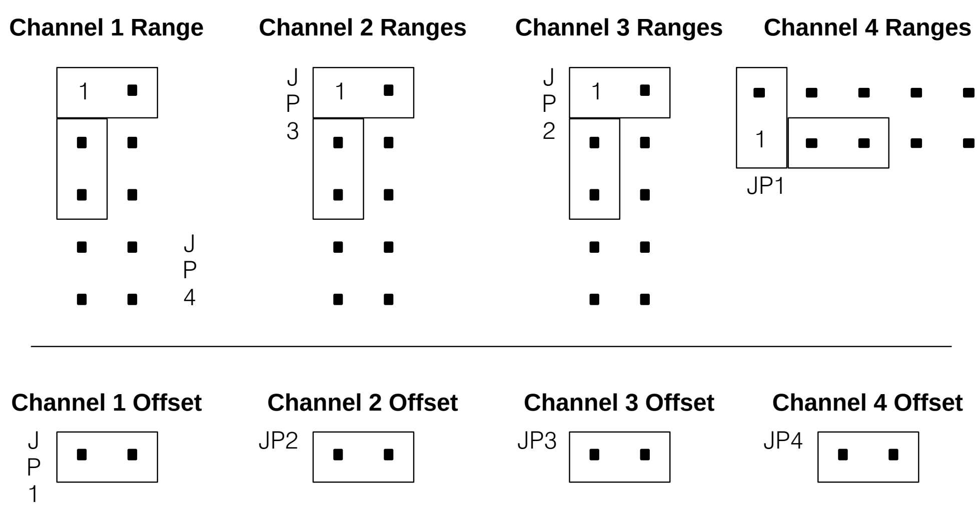
If you examine the top board on the module you will notice four sets of jumpers. The jumpers are assigned to the channels as follows.

- Channel 1 — Jumper JP4
- Channel 2 — Jumper JP3
- Channel 3 — Jumper JP2
- Channel 4 — Jumper JP1

NOTE: At first glance it might appear we have the channel / jumper assignments out of order. Your eyes do not deceive you. Channel 1 *is* controlled by JP4.

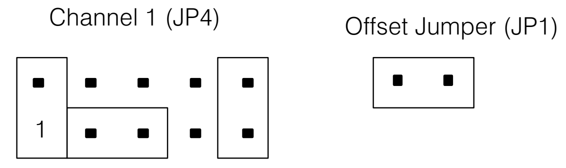
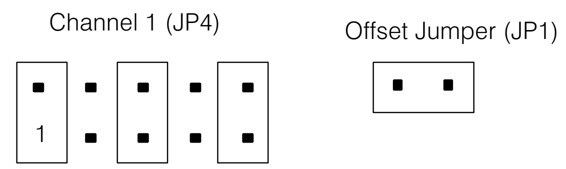
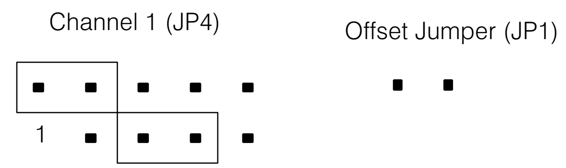
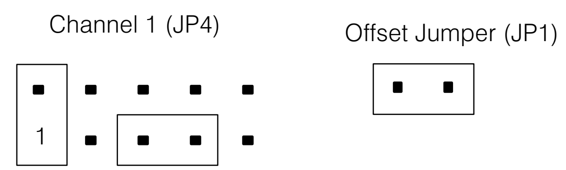
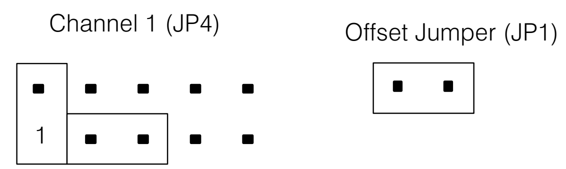
Each channel also has a jumper located on the bottom board of the module. These jumpers select a 1V (or 4mA) offset for each channel. Remove the jumper for any range that requires an offset. These jumpers are assigned as expected. JP1 selects an offset for channel 1, JP2 selects an offset for channel 2, etc.

The following diagram shows how the jumpers are assigned. It also shows the factory settings.



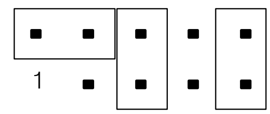
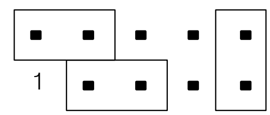
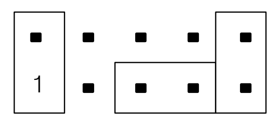
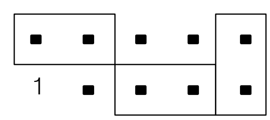
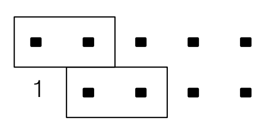
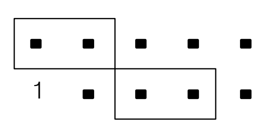
Selecting Input Signal Ranges

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.)

Bipolar Signal Range	Jumper Settings
–5 VDC to +5 VDC	
–10 VDC to +10 VDC	
Unipolar Signal Range	**Jumper Settings**
4 to 20 mA (1 VDC to 5 VDC)	
0 VDC to +5 VDC (0 to +20 mA)	
0 VDC to +10 VDC	

Special Output Signal Ranges

The following tables show the jumper selections for some additional ranges that are not normally found in many applications. Notice you can install or remove the offset jumper to change the settings. (Only channel 1 is used in the example, but all channels must be set.)

Signal Range Offset Installed	Signal Range Offset Removed	Jumper Settings
–10 VDC to +6 VDC	–9 VDC to +7 VDC	Channel 1 (JP4) 
–5 VDC to +3 VDC	–4 VDC to +4 VDC	Channel 1 (JP4) 
–2.5 VDC to +2.5 VDC	–1.5 VDC to +3.5 VDC	Channel 1 (JP4) 
–2.5 VDC to +1.5 VDC	–1.5 VDC to +2.5 VDC	Channel 1 (JP4) 
0 VDC to 8 VDC	1 VDC to 9 VDC	Channel 1 (JP4) 
0 VDC to 4 VDC	1 VDC to 5 VDC	Channel 1 (JP4) 

Connecting the Field Wiring

Wiring Guidelines

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

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the module or the power supply return (0V). *Do not* ground the shield at both the module and the transducer.
- 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 F3–04DAS receives all power from the base. A separate power supply is not required.

Load Requirements

Each channel can be wired independently for a voltage or current transducer.

- Current transducers must have an impedance less than 470 ohms.
- Voltage transducers must have an impedance greater than 2K ohms.

Removable Connector

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

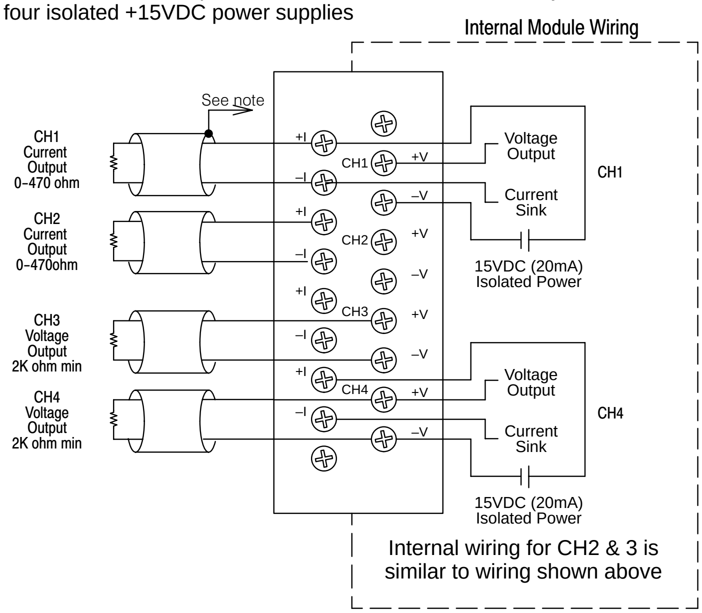
Wiring Diagram

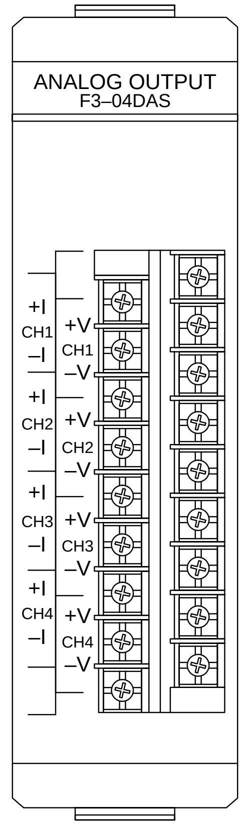
Note1: Shields should be connected to the respective channel's – V terminal of the module.

Note 2: Each isolated output channel may have either a voltage or current load, but not both

Note 3: An external 0.31 Amp fast-acting fuse in series with the isolated +I terminal (+15VDC) is recommended to protect against accidental shorts to the –V terminal (15VDC common)

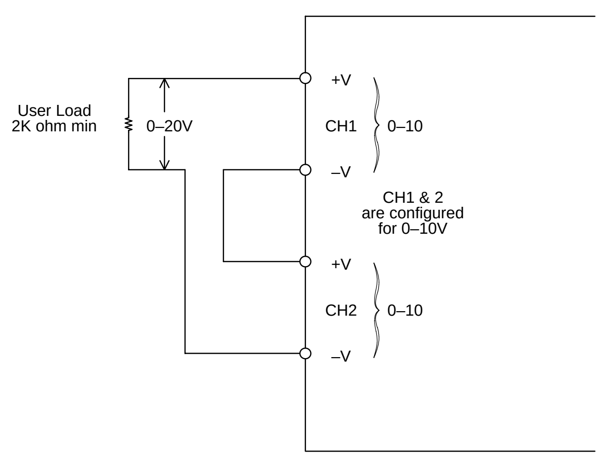
Note 4: Do not attempt to source more than 20mA from any one of the four isolated +15VDC power supplies





Combining Voltage Outputs

You may occasionally encounter transmitters that have a very unusual signal range. Since each channel is isolated, you can "daisy chain" the channels to provide output voltage signals that are outside of the normal operating range. For example, you could connect the first two channels to provide a voltage output from 0 to 20 VDC.

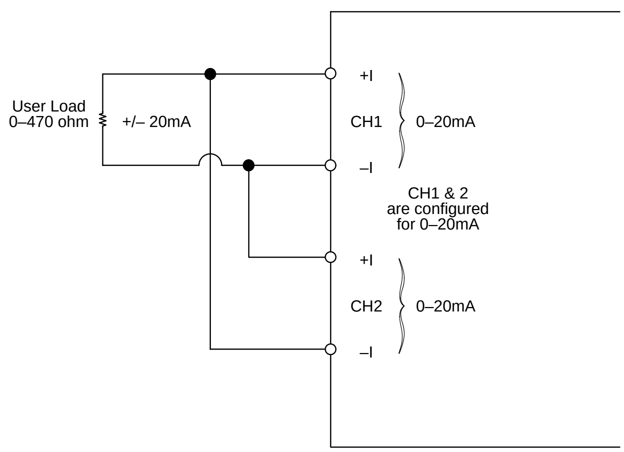


Combining Current Outputs

You cannot connect the current outputs in series (like the voltage outputs) but you can achieve unusual ranges with a few wiring and programming tricks. For example, let's say an application requires a ± 20 mA range. By completing the following steps, you could easily accommodate this requirement.

1. Configure channel 1 and channel 2 for 0–20mA.
2. Connect the +I of channel 1 to the –I of channel 2.
3. Connect the –I of channel 1 to the +I of channel 2.
4. Send 0 (digital value) to channel 2 while you send 0–4095 (digital value) to channel 1. To reverse the power flow, send 0 to channel 1 while you send the 0–4095 value to channel 2. (See the section on Writing the Control Program for information on sending data values.)

WARNING: The isolated +15 VDC power supplies are rated at a maximum of 20 mA. Current ratings that exceed 20 mA will damage the module beyond repair. For example, if you used the 0–10 VDC range for the example, the current would approach 40 mA which would cause damage to the module.

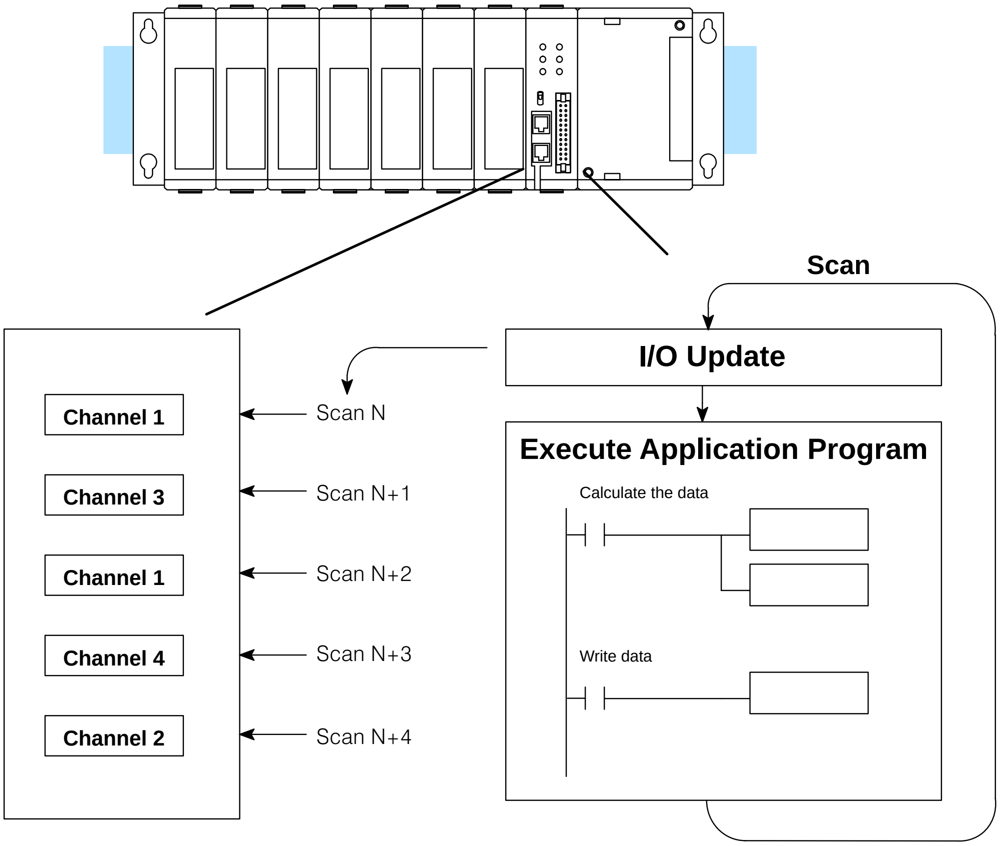


Module Operation

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 F3–04DAS module can update one channel per CPU scan. Your RLL program selects the channel to update, so you have complete flexibility in solving your application requirements.



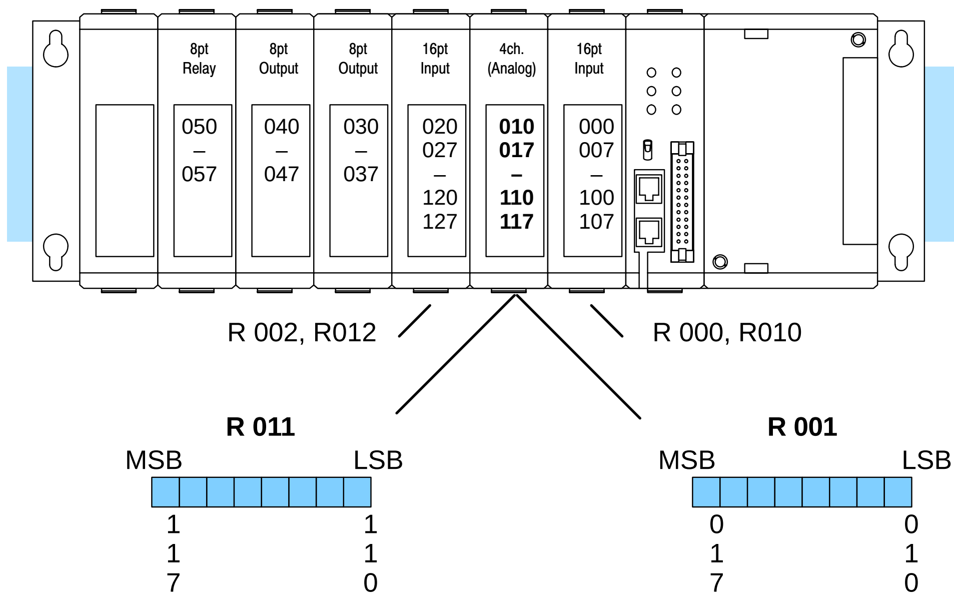
Understanding the I/O Assignments

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

- the digital representation of the analog signal.
- identification of the channel to receive the data.

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.

F3–04DAS

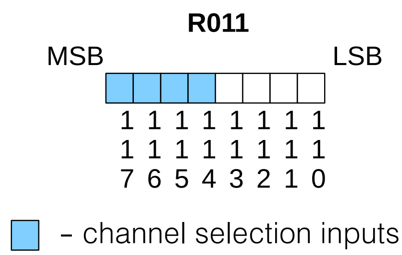


Within these two word locations, the individual bits represent specific information about the analog signal.

Channel Selection Inputs

The last four points of the upper register are used as outputs to tell the module which channel to update. In our example, when output 114 is on, channel 1 will be updated. Here's how the outputs are assigned.

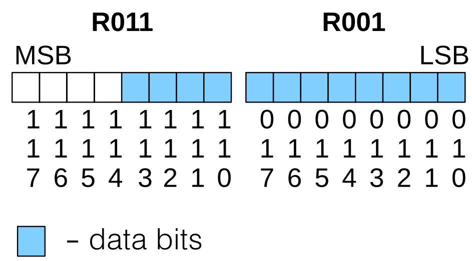
Output	Channels
114	1
115	2
116	3
117	4



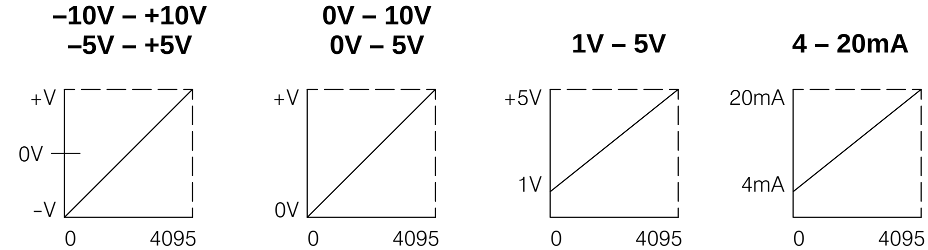
Analog Data Bits

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 possibly 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

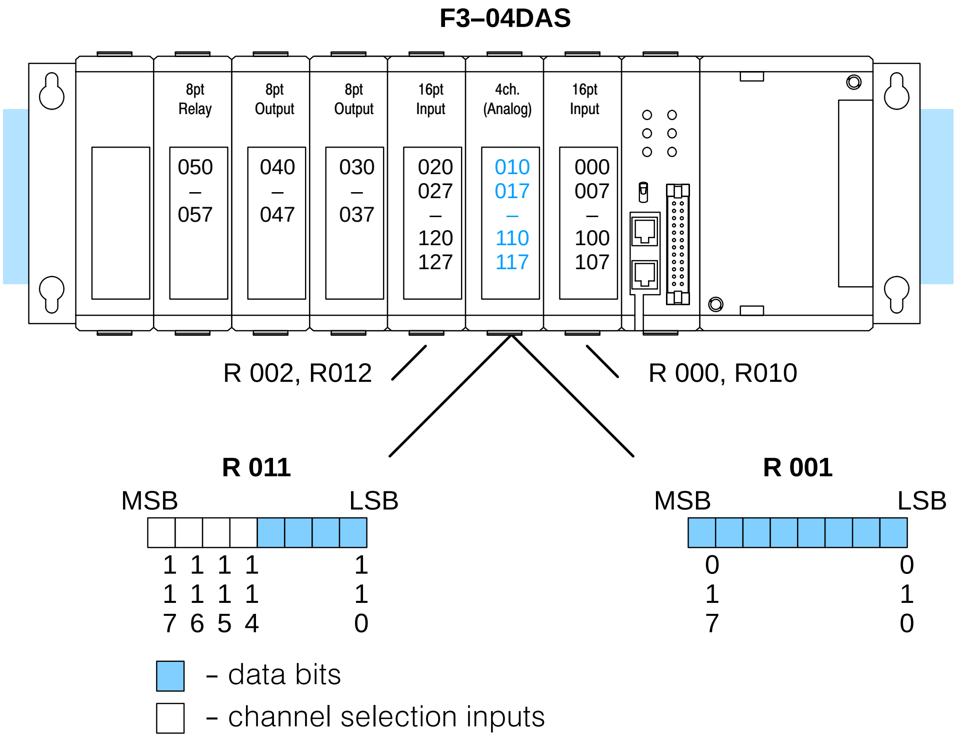
Range	Highest Signal	Lowest Signal	Smallest Change
–10 to +10V	+10V	–10V	4.88 mV
–5 to +5V	+5 V	–5V	2.44 mV
0 to 5V	5V	0V	1.22 mV
0 to 10V	10V	0V	2.44 mV
1 to 5V	5V	1V	0.98 mV
4 to 20mA	20mA	4mA	3.91 μA

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

As mentioned earlier, you can use the channel selection bits to determine which channels will be updated. The following diagram shows the location for both the channel selection bits and data bits.



Calculating the Digital Value

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

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

$$A = 4096 \frac{U}{H - L}$$

A = Analog value (0 – 4095)

U = Engineering Units

H = high limit of the Engineering unit range

L = low limit of the Engineering unit range

The following example shows how you would use Engineering Units to obtain the digital value to represent pressure (PSI) from 0 to 100. This example assumes you want to obtain a pressure of 42 PSI, which is slightly less than half scale.

$$A = 4096 \frac{U}{H - L}$$

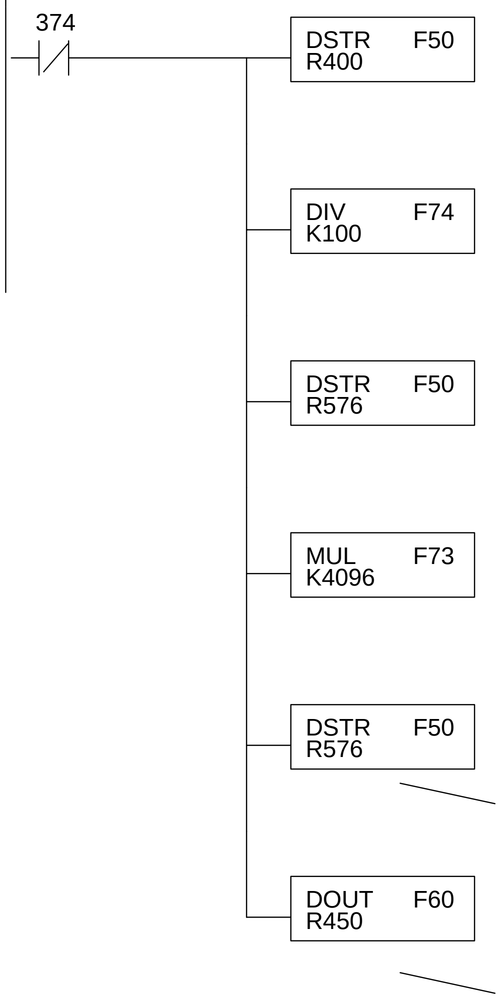
$$A = 4096 \frac{42}{100 - 0}$$

$$A = 1720$$

Here’s how you would write the program to perform the Engineering Unit conversion. This example assumes you have calculated or loaded the engineering unit value and stored it in R400. Also, you have to perform this for all channels if you’re using different data for each channel.

This example assumes you have already loaded the Engineering unit value in R400.

Scale the data



This instruction loads Engineering unit value into the accumulator on every scan.

Accumulator				Aux. Accumulator			
0	0	4	2	0	0	0	0
				R577		R576	

The Engineering unit value is divided by the Engineering unit range, which in this case is 100. (100 – 0 = 100)

Accumulator				Aux. Accumulator			
0	0	0	0	4	2	0	0
				R577		R576	

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator			
4	2	0	0	4	2	0	0
				R577		R576	

The accumulator is then multiplied by the module resolution, which is 4096. (4096 x 4200 = 17203200). Notice the most significant digits are now stored in the auxilliary accumulator. (This is different from the Divide instruction operation.)

Accumulator				Aux. Accumulator			
3	2	0	0	1	7	2	0
				R577		R576	

This instruction moves the two-byte auxilliary accumulator for further operations.

Accumulator				Aux. Accumulator			
1	7	2	0	1	7	2	0
				R577		R576	

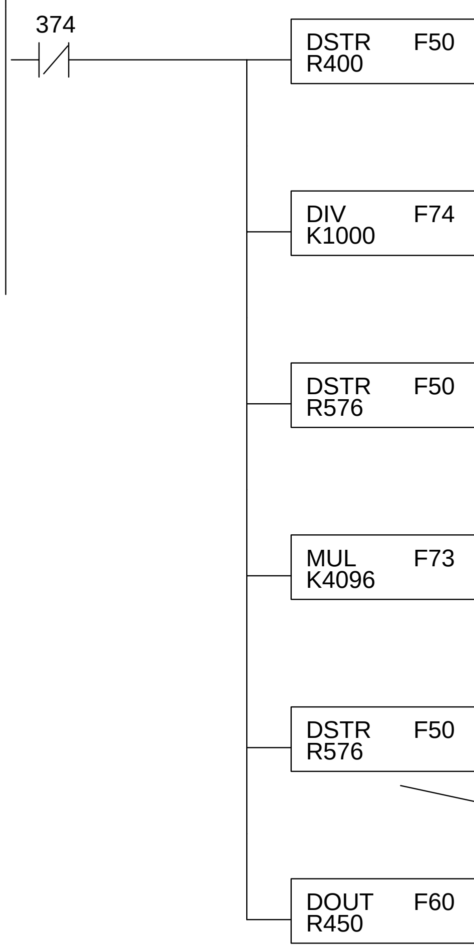
This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the digital value, which is 1720.

Accumulator				Store in R451 & R450			
1	7	2	0	1	7	2	0
				R451		R450	

There will probably be times when you need more precise control. For example, maybe your application requires 42.9 PSI, not just 42 PSI. By changing the scaling value slightly, we can “imply” an extra decimal of precision. Notice in the following example we’ve entered 429 as the Engineering unit value and 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 loaded the Engineering unit value in R400.

Scale the data



This instruction loads Engineering unit value into the accumulator on every scan.

Accumulator				Aux. Accumulator			
0	4	2	9	0	0	0	0
				R577		R576	

The Engineering unit value is divided by the Engineering unit range, which in this case is 1000. (100.0 implied range)

Accumulator				Aux. Accumulator			
0	0	0	0	4	2	9	0
				R577		R576	

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator			
4	2	9	0	4	2	9	0
				R577		R576	

The accumulator is then multiplied by the module resolution, which is 4096. (4096 x 4290 = 17571840). Notice the most significant digits are now stored in the auxilliary accumulator. (This is different from the Divide instruction operation.)

Accumulator				Aux. Accumulator			
1	8	4	0	1	7	5	7
				R577		R576	

This instruction moves the two-byte auxilliary accumulator for further operations.

Accumulator				Aux. Accumulator			
1	7	5	7	1	7	5	7
				R577		R576	

This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the digital value, which is 1757.

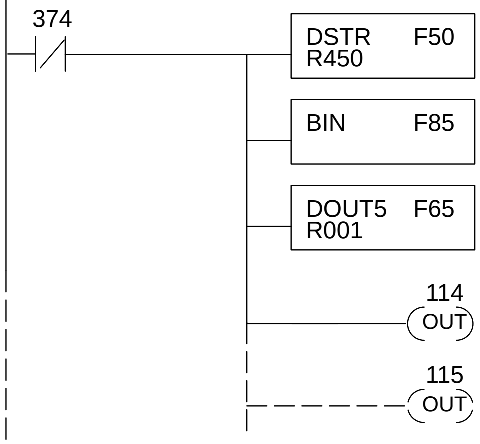
Accumulator				Store in R450 & R451			
1	7	5	7	1	7	5	7
				R450		R451	

Sending Data to a Single Channel

The following program example shows how to send the digital value to a single channel.

This example assumes you have already loaded the Engineering unit value in R450 and R451.

Send Channel 1



This rung loads the data into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the accumulator data to the Register that corresponds to the module, which is R001.

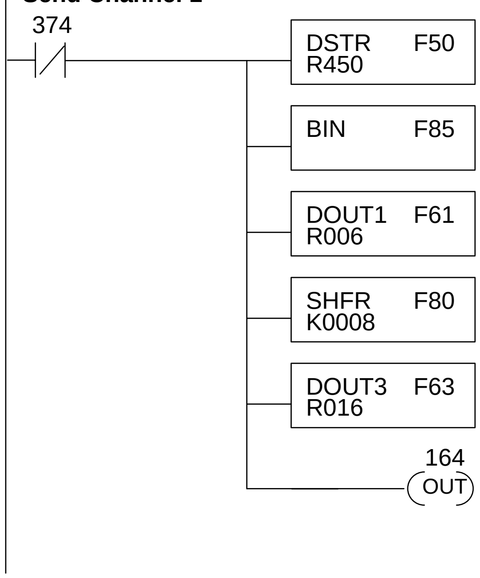
Indicate the channel to update. In this case, channel 1 is being updated.

To update other channels with the same output data, simple add the channel selection outputs for the additional channels.

If you install the F3–04DA–1 in the slot corresponding to registers 6 and 16, you have to make a slight program adjustment. This is because the DOUT5 instruction is not supported for this slot.

This example assumes you have already loaded the Engineering unit value in R450 and R451.

Send Channel 1



This rung loads the data into the accumulator on every scan.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module.

Send the 8 least significant data bits to the first Register that corresponds to the module which is R006.

Shift the 4 most significant data bits to the right 8 places. (The data is still in the accumulator).

Send the 4 most significant data bits to the second Register that corresponds to the module which is R016.

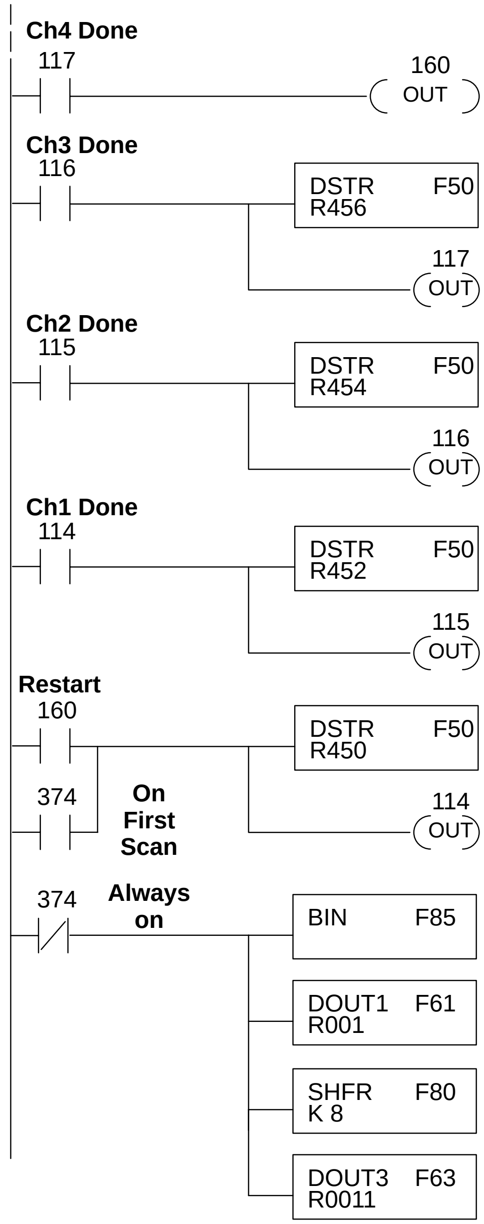
Indicate the channel to update. In this case, channel 1 is being updated.

Sequencing the Channel Updates

This example shows how to send digital values to the module when you have more than one channel. This example will automatically update all four channels over four scans. The example is fairly simple and will work in most all situations, but there are instances where problems can occur. The logic must be active on the first CPU scan and all subsequent scans. If the logic gets stopped or disabled for some reason, there is no way to restart it. If you're using an RLL*PLUS* (Stage) program, put this logic in an initial stage that is always active. Also, you should avoid using the this example if you require the analog output logic to be used inside a Master Control Relay field of control. You could also accidentally disable the analog output logic by inadvertently writing to the multiplexing control relays with an operator interface or intelligent module, such as an ASCII BASIC module, etc.

Sequencing Example

The following program example shows how to send the digital values to multiple channels. With this program, all channels will be updated within four scans. You must use the rungs in the order shown, but you can include them anywhere in the program.



When channel 4 has been updated, 160 restarts the update sequence.

When channel 3 has been updated, this rung loads the data for channel 4 into the accumulator. By turning on 117, this triggers the channel update. (Since 117 is also used as an input, this results in a one-shot.)

When channel 2 has been updated, this rung loads the data for channel 3 into the accumulator. By turning on 116, this triggers the channel update. (Since 116 is also used as an input, this results in a one-shot.)

When channel 1 has been updated, this rung loads the data for channel 2 into the accumulator. By turning on 115, this triggers the channel update. (Since 115 is also used as an input, this results in a one-shot.)

This rung loads the data for channel 1 into the accumulator. Since 374 is used, this rung automatically executes on the first scan. After that, 160 restarts this rung. If you examine the first rung, you'll notice 160 only comes on after channel 4 has been updated.

Since the data is in BCD format, you have to convert it to binary before you send the data to the module. (You can omit this step if you've already converted the data elsewhere.)

Send the 8 least significant data bits to the first Register that corresponds to the module which is R001.

Shift the 4 most significant data bits to the right 8 places. (The data is still in the accumulator).

Send the 4 most significant data bits to the second Register that corresponds to the module which is R011.

Writing the Control Program (DL350)

Reading Values: Pointer Method and Multiplexing

There are two methods of reading values:

- The pointer method **(all system bases must be D3–xx–1 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 CPU, but for ease of programming it is strongly recommended that you use the pointer method.

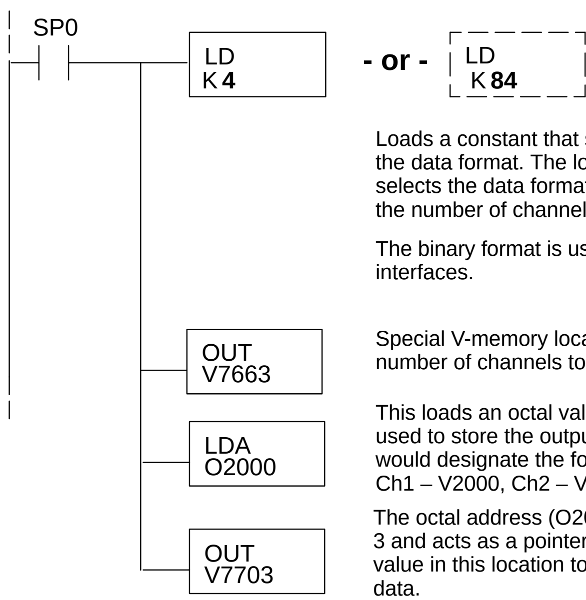
Pointer Method

The DL350 has special V-memory locations assigned to each base slot that greatly simplifies the programming requirements. By using these V-memory locations you can:

- specify the number of channels to update.
- specify where to obtain the output data.

NOTE: Do not use the pointer method and the PID Control Output auto transfer to 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 output data from the PID loop to the output module memory location(s).

The following program example shows how to set up these locations. Place this rung anywhere in the ladder program, or in the initial stage when using stage programming.



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

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

Special V-memory location assigned to slot 3 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 output data. For example, the O2000 entered here would designate the following addresses.
Ch1 – V2000, Ch2 – V2001

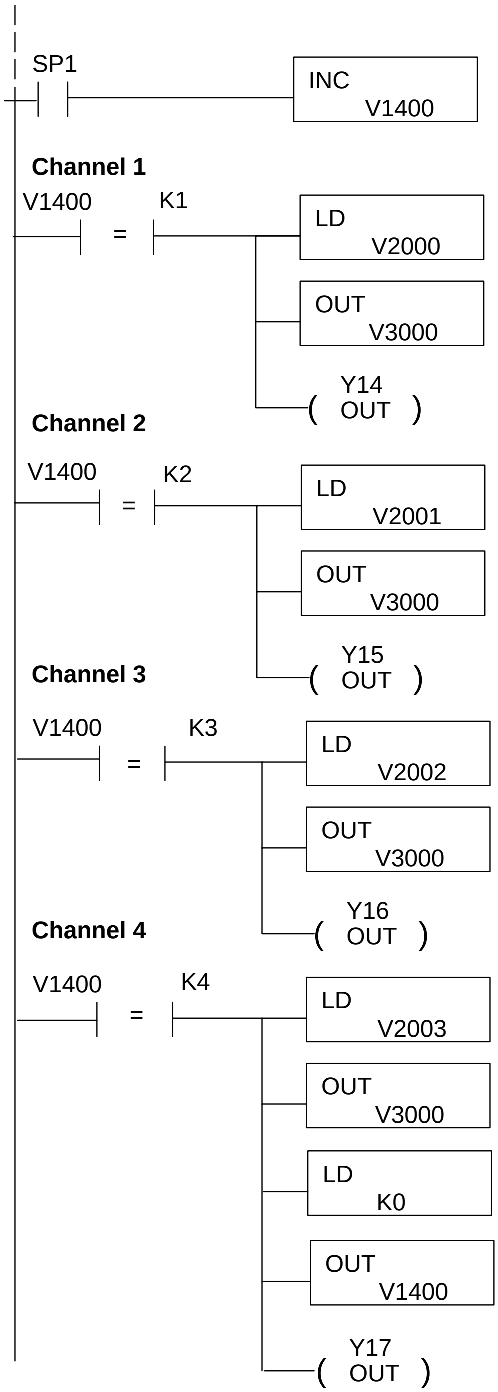
The octal address (O2000) is stored here. V7703 is assigned to slot 3 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the output data.

The table shows the special V-memory locations used with the DL350. Slot 0 (zero) is the module next to the CPU. Remember, the CPU only examines the pointer values at these locations after a mode transition. The pointer method is supported on expansion bases (all bases must be D3–xx–1) up to a total of 8 slots away from the DL350. The pointer method is not supported in slot 8 of a 10 slot base.

Analog Output 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	V7700	V7701	V7702	V7703	V7704	V7705	V7706	V7707

Multiplexing: DL350 with a D3–xx–1 Base

This example assumes the module is in Y0 address slot of a D3–xx–1. In this example V2000 contains the data for channel 1 and V2001 for channel 2, etc. in BCD. 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.



This rung loads increments V1400 once every scan from 0–4.

This rung loads the data for channel 1 into the accumulator when V1400 = 1.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 1 is Y14.

This rung loads the data for channel 2 into the accumulator when V1400 = 2.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 2 is Y15.

This rung loads the data for channel 3 into the accumulator when V1400 = 3.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 3 is Y16.

This rung loads the data for channel 4 into the accumulator when V1400 = 4.

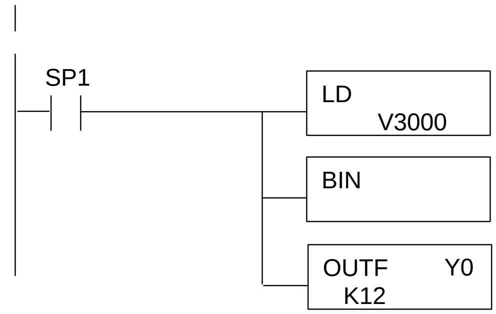
The data is stored in V3000 before sending it to the module.

V1400 is reset to 0 when V1400 is =4.

The channel select bit for channel 4 is Y17.

example program continued on next page

example program continued from previous page

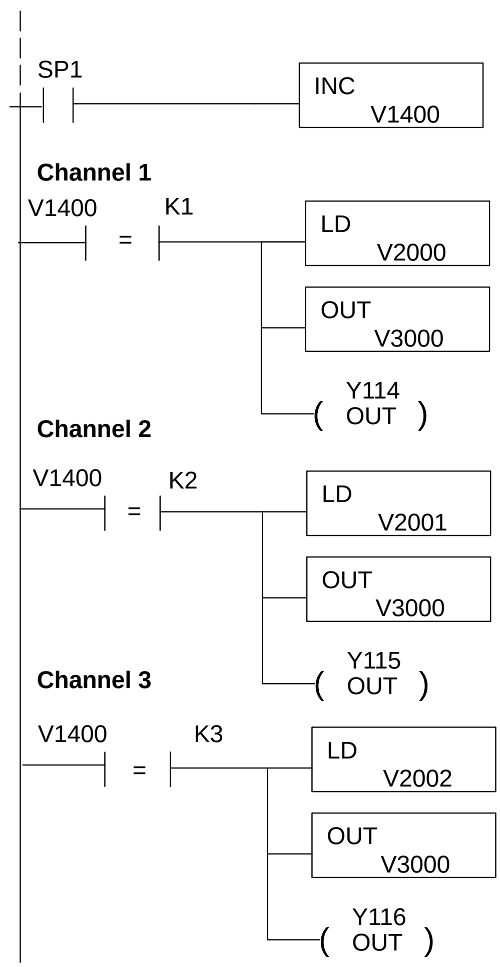


This rung converts the appropriate analog channel data to binary for the module.

The OUTF instruction sends the 12 bits of analog data to the analog module memory address.

Multiplexing: DL350 with Conventional DL305 Base

This example assumes the module is in the Y0–10 / Y100–107 slot of a 305 conventional base. In this example V2000 contains the BCD data for channel 1 and V2001 contains the data for channel 2, etc. One more rung would be necessary for channel 4.



This rung loads increments V1400 once every scan from 0–4.

This rung loads the data for channel 1 into the accumulator when V1400 = 1.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 1 is Y14.

This rung loads the data for channel 2 into the accumulator when V1400 = 2.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 2 is Y15.

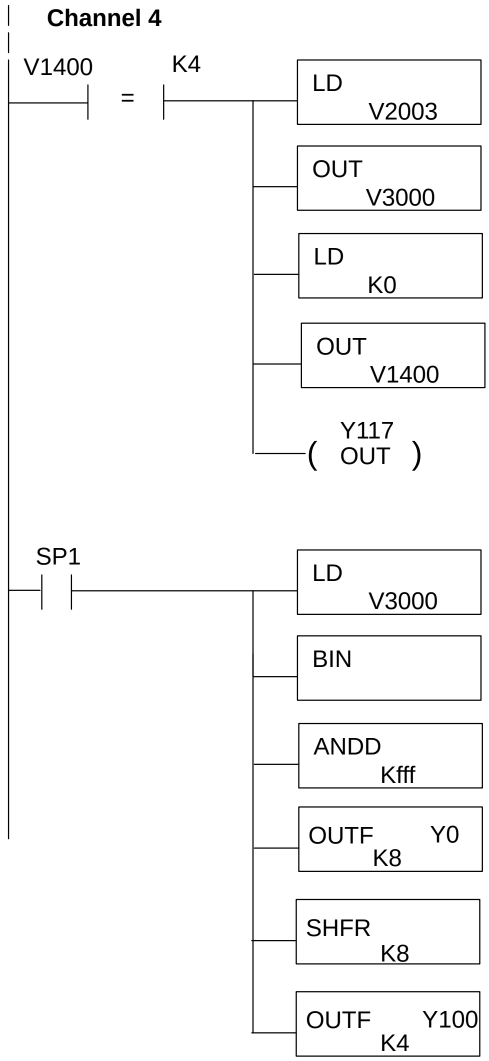
This rung loads the data for channel 3 into the accumulator when V1400 = 3.

The data is stored in V3000 before sending it to the module.

The channel select bit for channel 3 is Y16.

example program continued on next page

example program continued from previous page



This rung loads the data for channel 4 into the accumulator when V1400 = 4.

The data is stored in V3000 before sending it to the module.

V1400 is reset to 0 when V1400 is =4.

The channel select bit for channel 4 is Y17.

The BIN converts the appropriate analog channel data to binary for the module.

The OUTF and SHFR instruction formats the data and sends the 12 bits of analog data to the analog module memory address.

Calculating the Digital Value

Your program must calculate the digital value to send to the analog module. 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.

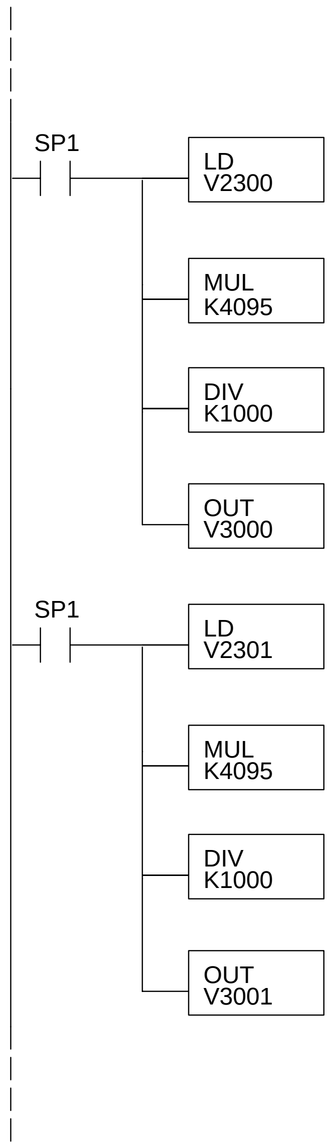
$$A = 10U \frac{4095}{10(H - L)}$$

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

$$A = 2023$$

The example program below shows how you would write the program to perform the engineering unit conversion. This example assumes you have calculated or loaded the engineering unit values in BCD and stored them in V2300 and V2301 for channels 1 and 2 respectively.

NOTE: The DL350 offers various instructions that allow you to perform math operations using BCD format. It is easier to perform math calculations in BCD and then convert the value to binary before sending the data to the module.



The LD instruction loads the engineering units used with channel 1 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 4095 (to start the conversion).

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

Store the BCD result in V3000 (the actual steps to write the data were shown earlier).

The LD instruction loads the engineering units used with channel 2 into the accumulator. This example assumes the numbers are BCD. Since SP1 is used, this rung automatically executes on every scan. You could also use an X, C, etc. permissive contact.

Multiply the accumulator by 4095 (to start the conversion).

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

Store the BCD result in V3001 (the actual steps to write the data were shown earlier).

Analog and Digital Value Conversions

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

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

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

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

$$D = 3276$$

9

F3–08THM–n 8-Channel Thermocouple Input

In This Chapter. . . .

Introduction

Automatic Conversion

The F3–08THM–n Thermocouple Input Module provides eight, differential thermocouple input channels (12-bit resolution). The module automatically converts type E, J, K, R, S or T thermocouple signals into direct temperature readings. No extra scaling or complex conversion is required. You can select between °F or °C operation.

This module is also available in versions specially designed to convert millivolt signal levels into direct digital values (0–4095). Two versions are available, one for 0–50mV and one for 0–100mV.

Hardware Features

The F3–08THM–n also features automatic cold junction compensation, thermocouple linearization, plus analog and digital filtering. The temperature calculation and linerazation are based on data provided by the National Bureau of Standards.

Diagnostic Features

Thermocouple burnout and other errors are automatically reported to the CPU. For example, if the thermocouple becomes disconnected, then a value of 4095 is assigned to that channel.

Module Specifications

The following table provides the specifications for the F3–08THM–n Thermocouple Input Module from FACTS Engineering. Review these specifications to make sure the module meets your application requirements.

Number of Channels	8, differential inputs
Input Ranges	Type E: –270/1000 °C, –450/1832 °F Type J: –210/760 °C, –350/1390 °F Type K: –270/1370 °C, –450/2500 °F Type R: 0/1768 °C, –32/3214 °F Type S: 0/1768 °C, –32/3214 °F Type T: –270/400 °C, –450/752 °F –1: 0 – 50 mV –2: 0–100 mV
Resolution	12 bit (1 in 4096)
Input Impedance	27KΩ DC
Absolute Maximum Ratings	Fault protected input, 130 Vrms or 100 VDC
Cold Junction Compensation	Automatic
Conversion Time	15ms per channel, minimum 1 channel per CPU scan
Converter Type	Successive Approximation, 574
Linearity Error	±1 count (0.03% of full scale) maximum
Maximum Inaccuracy at 77 °F (25 °C)	0.35% of full scale
Accuracy vs. Temperature	57 ppm / °C maximum full scale
Power Budget Requirement	50 mA @ 9 VDC, 34 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 F3–08THM–n Thermocouple 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

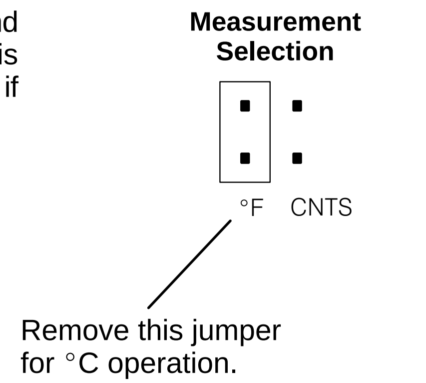
Jumper Locations

The module is set at the factory for °C thermocouple readings. If this is acceptable you do not have to change any of the jumpers. The following diagram shows how the jumpers are set.

WARNING: DO NOT change the calibration jumper settings. If you think this jumper has been changed, make sure it is NOT in the CAL position. All calibration is performed at the factory. Any changes to this may affect the module accuracy which could result in the risk of personal injury and/or equipment damage.

Selecting °F or °C Operation

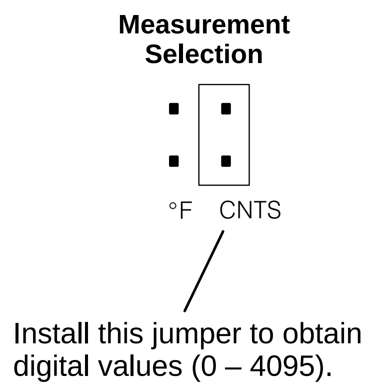
There is a jumper located on the bottom of the board that selects between °C and °F temperature measurements. This jumper (labeled °F) should be removed if you require °C measurements.



Selecting 0–4095 Operation

There is a jumper located on the bottom of the board that allows you to disable the direct temperature conversion feature. If you install a jumper on the CNTS pin, the temperature will be represented by a digital value between 0 and 4095. For example, an E type thermocouple would have a value of 0 for –450 °F and a value of 4095 for 1832 °F.

NOTE: If you are using the –1 (50mV) or the –2 (100mV) millivolt input versions, you should make sure this jumper is 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.

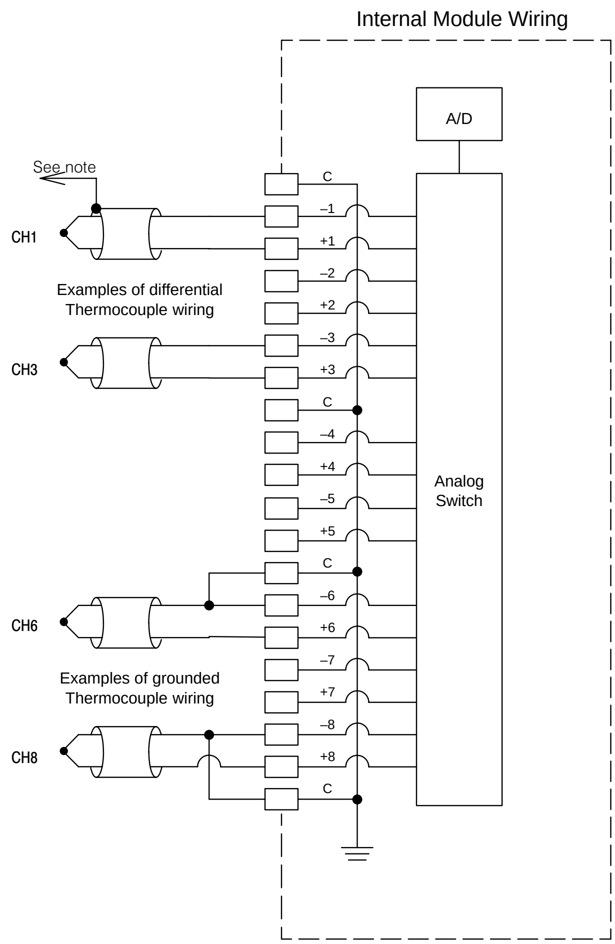
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the signal source. *Do not* ground the shield at both the module and the source.
- Don’t run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.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

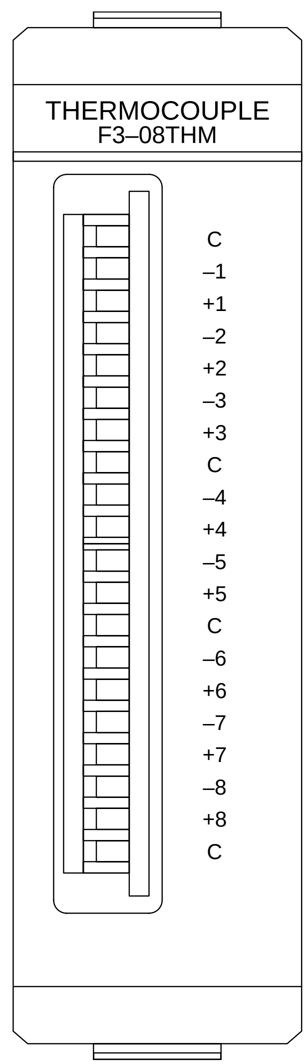
User Power Supply Requirements

The F3–08THM–n receives all power from the base. A separate power supply is not required.

Wiring Diagram

Note 1: Terminate shields at the respective signal source
Note 2: Leave unused channels open (no connection)



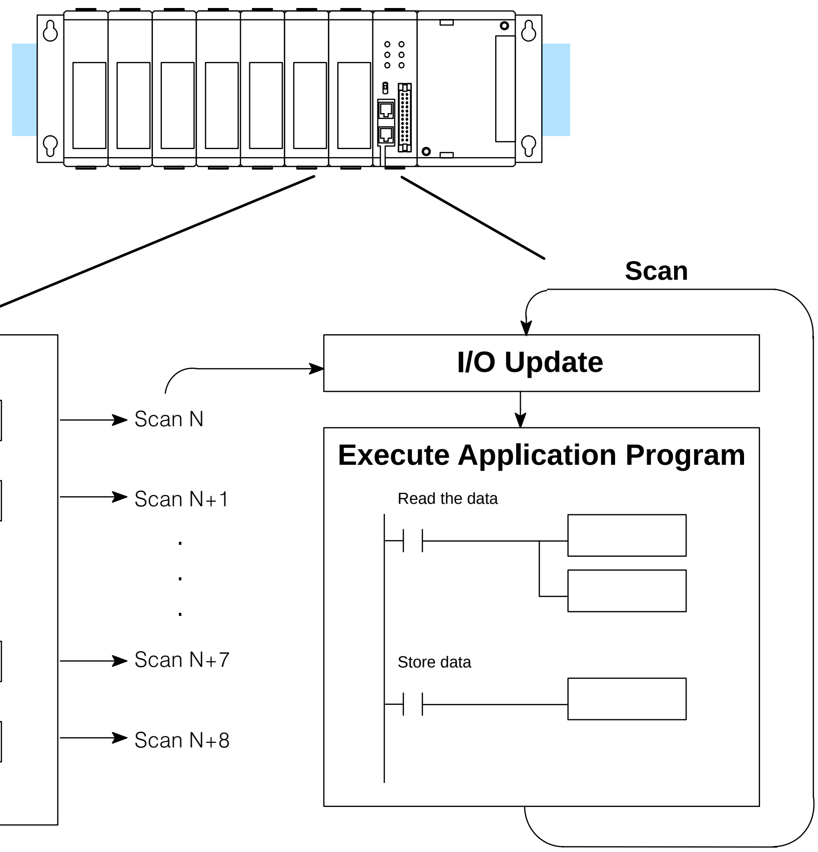


Module Operation

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 F3–08THM–n module supplies1 channel of data per each CPU scan. Since there are eight channels, it can take up to eight scans to get data for all channels. Once all channels have been scanned the process starts over with channel 1.



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

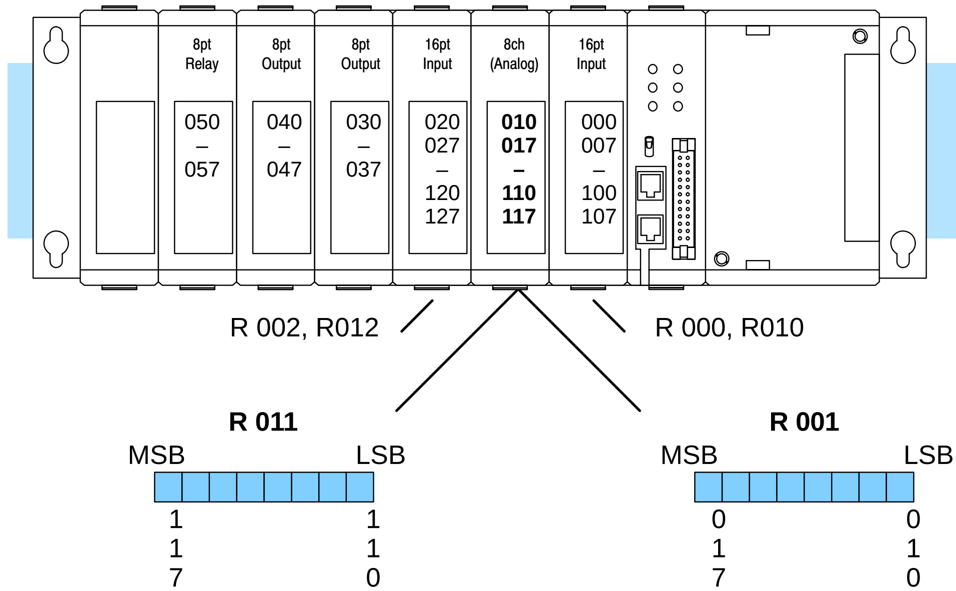
Understanding the I/O Assignments

You may recall the F3–08THM–n 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 temperature.

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.

F3–08THM

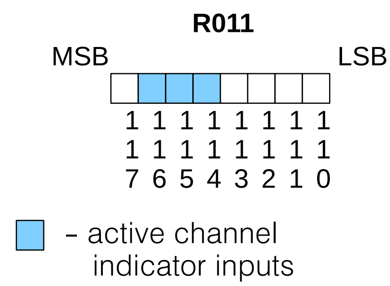


Within these two register locations, the individual bits represent specific information about the analog signal.

Active Channel Indicator Inputs

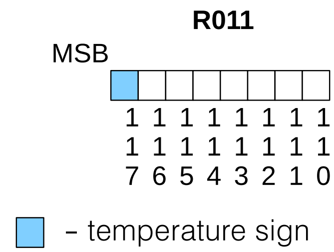
The next to last three bits of the upper Register indicate the active channel. The indicators automatically increment with each CPU scan.

Scan	Active Channel Inputs	Channel
N	000	1
N+1	001	2
N+2	010	3
N+3	011	4
N+4	100	5
N+5	101	6
N+6	110	7
N+7	111	8
N+8	000	1



Temperature Sign Bit

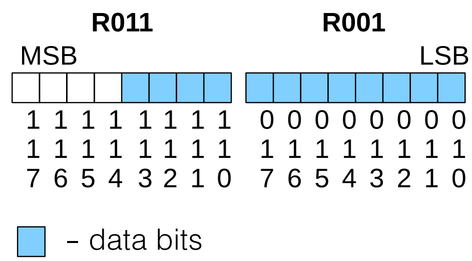
The most significant bit is used to note the sign of the temperature. If this bit is on, then the temperature is negative. If the bit is off, then the temperature is positive.



Analog Data Bits

The first twelve bits represent the temperature. If you have selected the 0–4095 scale, the following format is used.

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



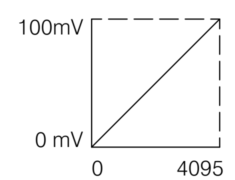
Temperature Input Resolution

Typically, the F3–08THM–n resolution enables you to detect a 1 °C change in temperature. The National Bureau of Standards publishes conversion tables that show how each temperature corresponds to an equivalent signal level.

Millivolt Input Resolution

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 –2 (100mV) module a signal of 0 mV would be 0, and a signal of 100 mV 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 diagram shows how this relates to the example signal range.

0–100 mV Scale



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

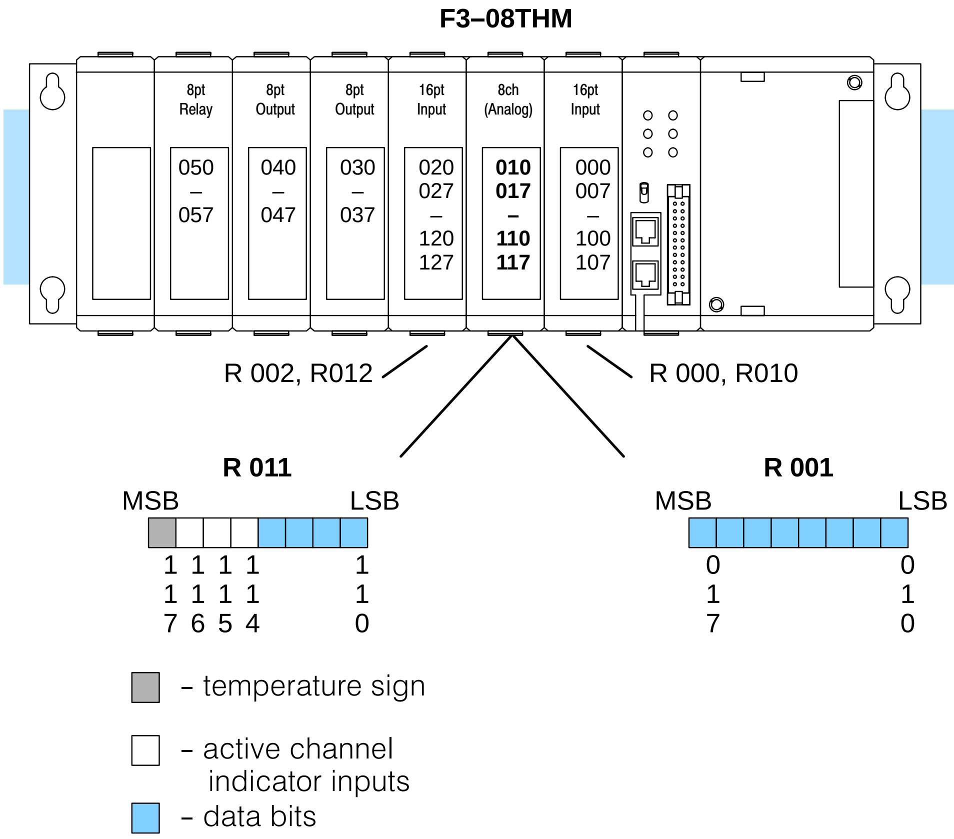
Range	Highest Signal	Lowest Signal	Smallest Change
0 – 50 mV	50 mV	0 mV	12.2 µV
0 – 100 mV	100mA	0mA	24.2 µV

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

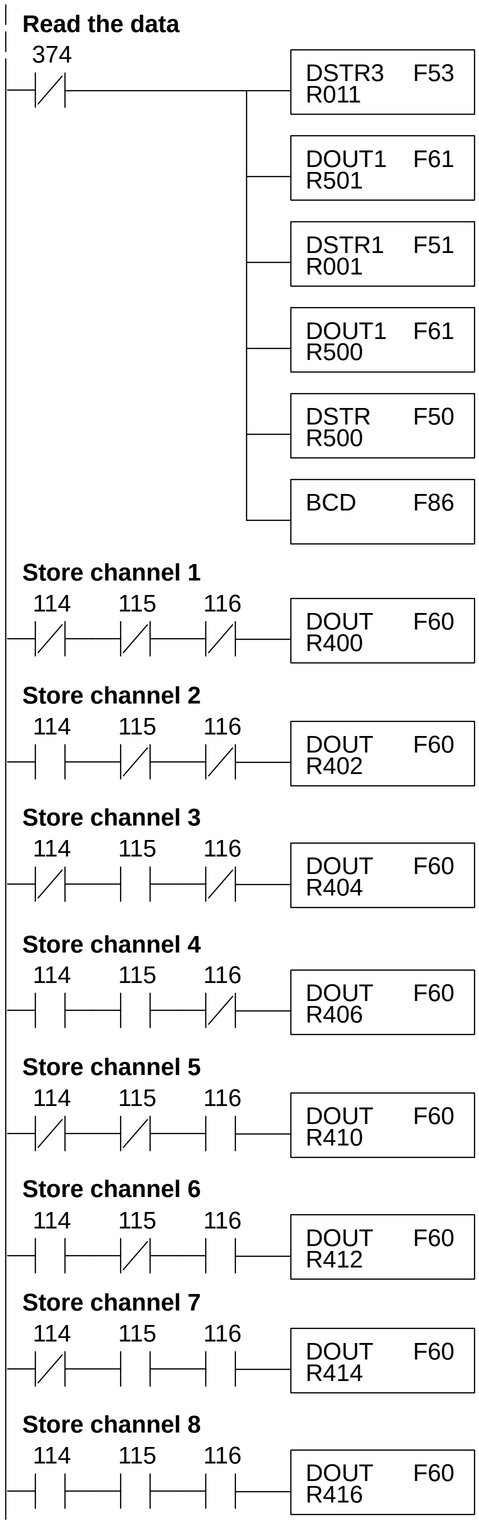
Since all channels are multiplexed into a single data word, the control program must be setup to determine which channel is being read. Since the module provides input points to the CPU, it is very easy to use the channel status bits to determine which channel is being monitored.



Automatic Temperature Conversion

If you are using the temperature scale (°F or° C) then you do not have to perform any scaling. Once you convert the binary temperature reading to a four-digit BCD number, you have the temperature.

The following example shows a program designed to read any of the available channels of 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 data bits into the accumulator from Register 011 on every scan.

Temporarily store the bits to Register 501.

This rung loads the eight 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.

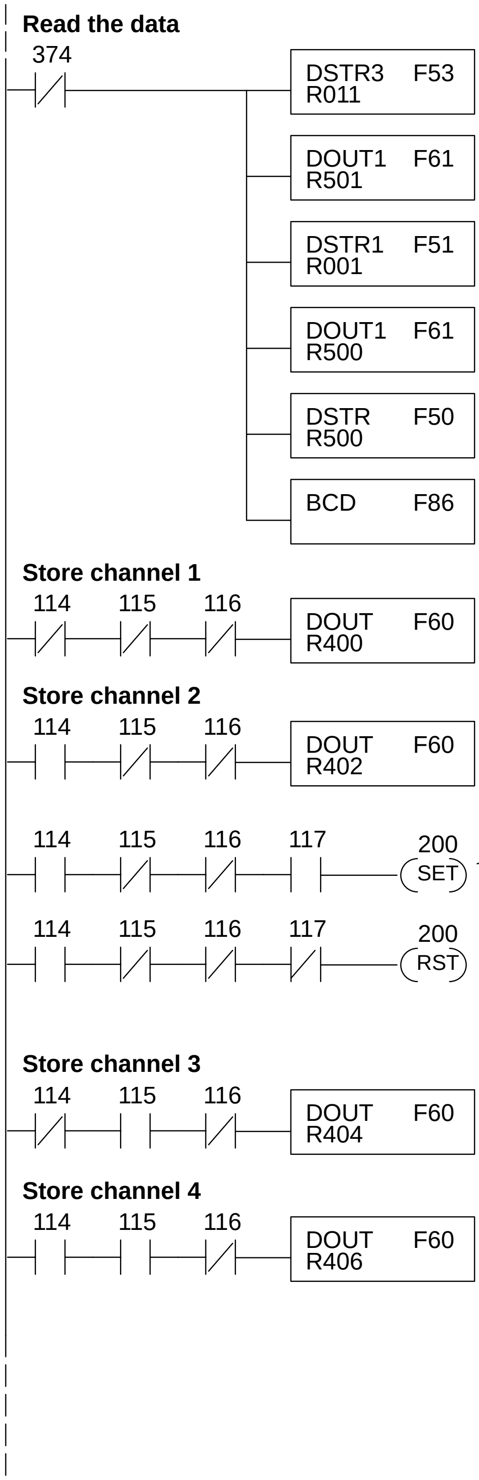
Now that all the bits are stored, load all twelve bits into the accumulator.

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.)

Using the Sign Bit

By adding a couple of simple rungs you can easily monitor the temperature for positive vs. negative readings. (For example, you have to know whether the temperature is +100 °F or –100 °F.) Notice how we've changed Channel 2 to control an output that denotes the sign of the temperature.



This rung loads the four data bits into the accumulator from Register 011 on every scan.

Temporarily store the bits to Register 501.

This rung loads the eight 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.)

If 117 is on, then the temperature on channel 2 is negative.

If 117 is off, then the temperature on channel 2 is positive.

Scaling the Input Data

If you are using the –1 (50mV) or the –2 (100mV) versions, you may want to scale the data to represent the 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

NOTE: The thermocouple versions automatically provide the correct temperature readings. Scaling is not required.

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.

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

$$\downarrow$$

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

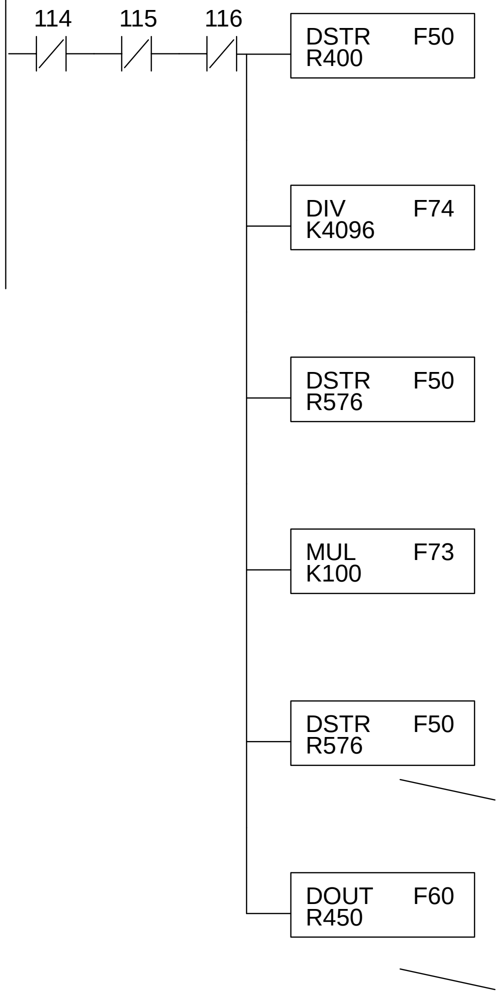
$$\downarrow$$

$$\text{Units} = 42.9$$

The following instructions are required to scale the data. (We'll continue to use the 42.9 PSI example.) Once we've explained how these instructions operate, we'll show an example program.

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

Scale the data



This instruction brings the analog value (in BCD) into the accumulator.

Accumulator				Aux. Accumulator			
1	7	6	0	0	0	0	0
				R577		R576	

The analog value is divided by the resolution of the module, which is 4096. (1760 / 4096 = 0.4296)

Accumulator				Aux. Accumulator			
0	0	0	0	4	2	9	6
				R577		R576	

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator			
4	2	9	6	4	2	9	6
				R577		R576	

The accumulator is then multiplied by the scaling factor, which is 100. (100 x 4296 = 429600). Notice the most significant digits are now stored in the auxilliary accumulator. (This is different from the way the Divide instruction operates.)

Accumulator				Aux. Accumulator			
9	6	0	0	0	0	4	2
				R577		R576	

This instruction moves the two-byte auxilliary accumulator for further operations.

Accumulator				Aux. Accumulator			
0	0	4	2	0	0	4	2
				R577		R576	

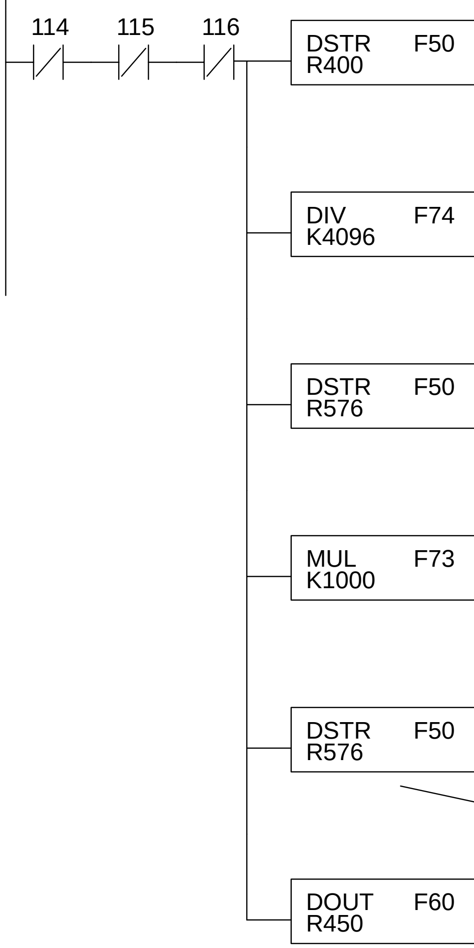
This instruction stores the accumulator to R450 and R451. R450 and R451 now contain the PSI, which is 42 PSI.

Accumulator				Store in R451 & R450			
0	0	4	2	0	0	4	2
				R451		R450	

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

Scale the data



This instruction brings the analog value (in BCD) into the accumulator.

Accumulator	Aux. Accumulator (R577 R576)
1 7 6 0	0 0 0 0

The analog value is divided by the resolution of the module, which is 4096. (1760 / 4096 = 0.4296)

Accumulator	Aux. Accumulator (R577 R576)
0 0 0 0	4 2 9 6

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator	Aux. Accumulator (R577 R576)
4 2 9 6	4 2 9 6

The accumulator is multiplied by the scaling factor, which is now 1000. (1000 x 4296 = 4296000). The most significant digits are now stored in the auxilliary accumulator. (This is different from the way the Divide instruction operates.)

Accumulator	Aux. Accumulator (R577 R576)
6 0 0 0	0 4 2 9

This instruction moves the two-byte auxilliary accumulator for further operations.

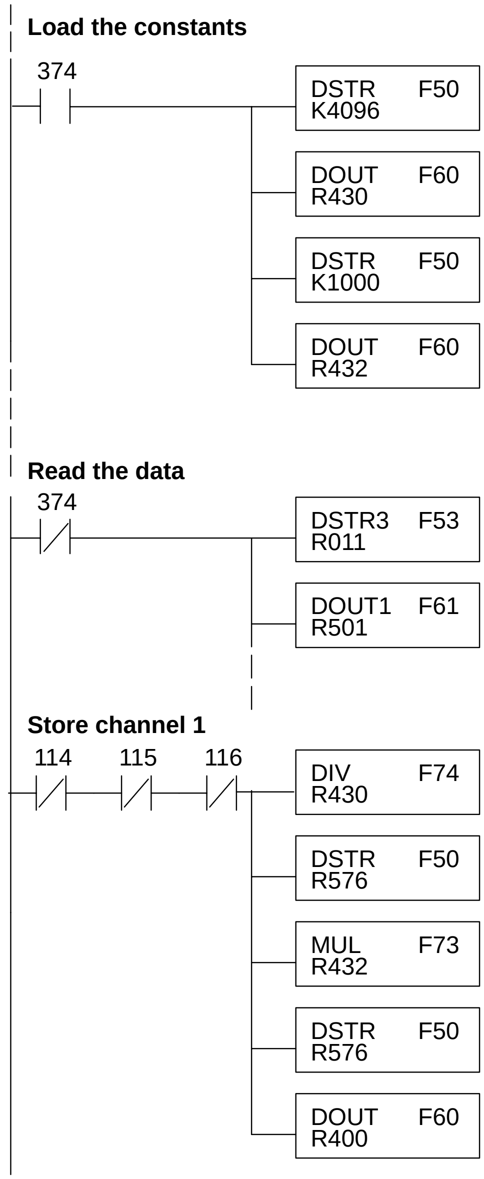
Accumulator	Aux. Accumulator (R577 R576)
0 4 2 9	0 4 2 9

This instruction stores the accumulator to R450 and R451. R450 and R451 now contains the PSI, which implies 42.9.

Accumulator	Store in R451 & R450 (R451 R450)
0 4 2 9	0 4 2 9

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 on every scan.

Temporarily store the bits to Register 501.

The analog value is divided by the resolution of the module, which is stored in R430 and R431.

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 and R433.

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

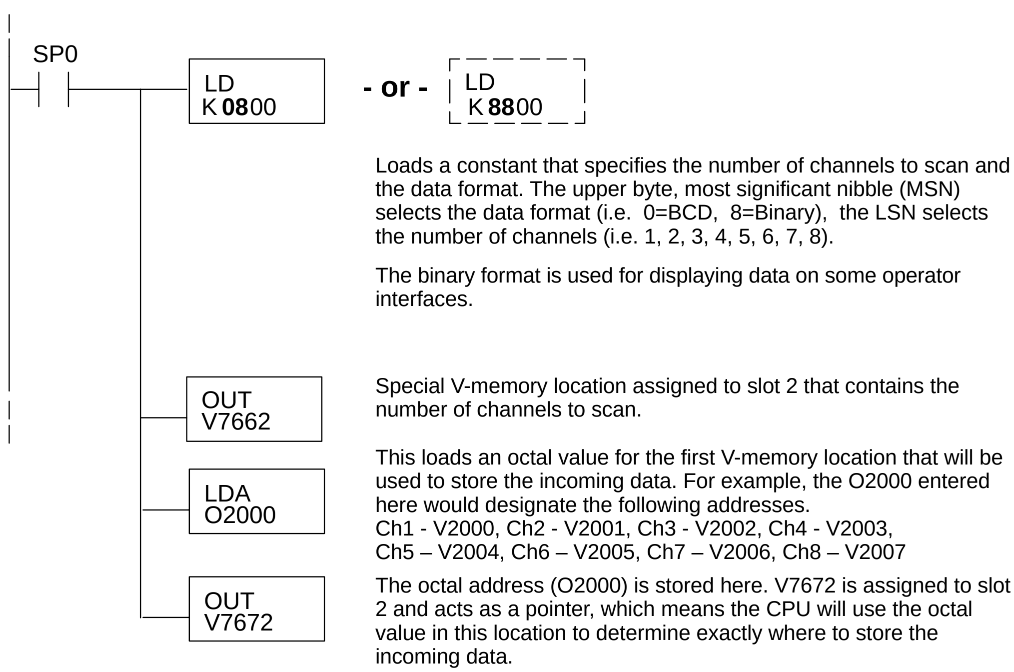
You must use the multiplexing method with remote I/O modules (the pointer method will not work). You can use either method when using DL350, but for ease of programming it is strongly recommended that you use the pointer method.

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.

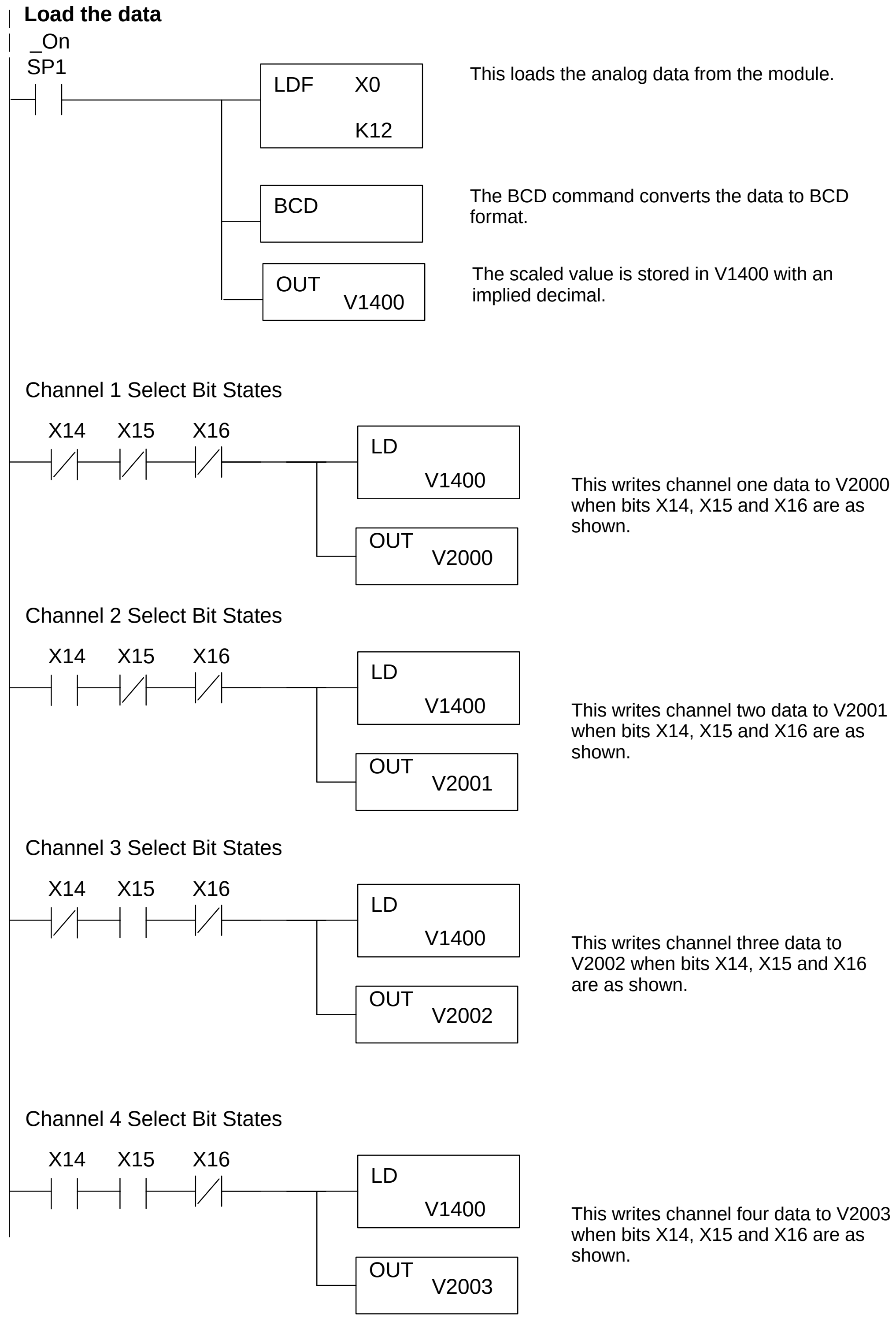


The table shows the special V-memory locations used with the DL350. Slot 0 (zero) is the module next to the CPU, slot 1 is the module two places from the CPU, and so on. Remember, the CPU only examines the pointer values at these locations after a mode transition. The pointer method is supported on expansion 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.

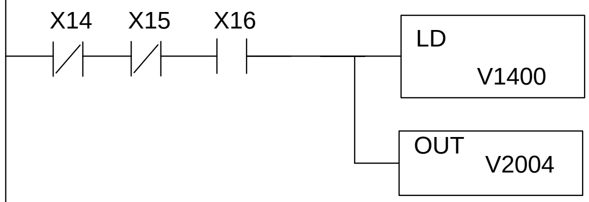
Analog Input Module Slot-Dependent V-memory Locations								
Slot	0	1	2	3	4	5	6	7
No. of Channels	V7660	V7661	V7662	V7663	V7664	V7665	V7666	V7667
Storage Pointer	V7670	V7671	V7672	V7673	V7674	V7675	V7676	V7677

Multiplexing: DL350 with a D3–XX–1 Base

The example below shows how to read multiple channels on an F3–08THM Thermocouple module in the X0 address slot 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.

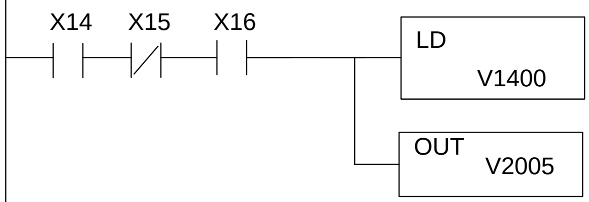


Channel 5 Select Bit States



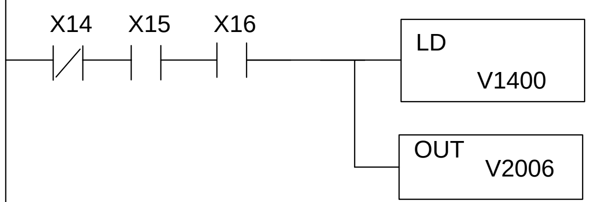
This writes channel five data to V2004 when bits X14, X15 and X16 are as shown.

Channel 6 Select Bit States



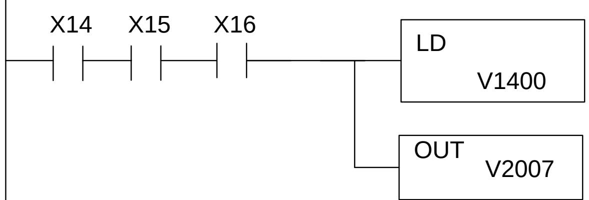
This writes channel six data to V2005 when bits X14, X15 and X16 are as shown.

Channel 7 Select Bit States



This writes channel seven data to V2006 when bits X14, X15 and X16 are as shown.

Channel 8 Select Bit States

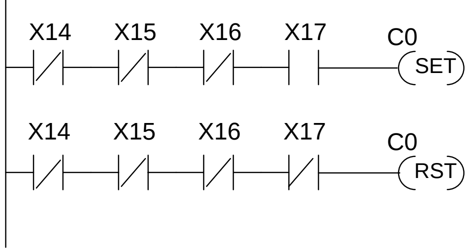


This writes channel eight data to V2007 when bits X14, X15 and X16 are as shown.

Using the Sign Bit

Channel 1 Selected

X17 is the sign bit when in module address 0.

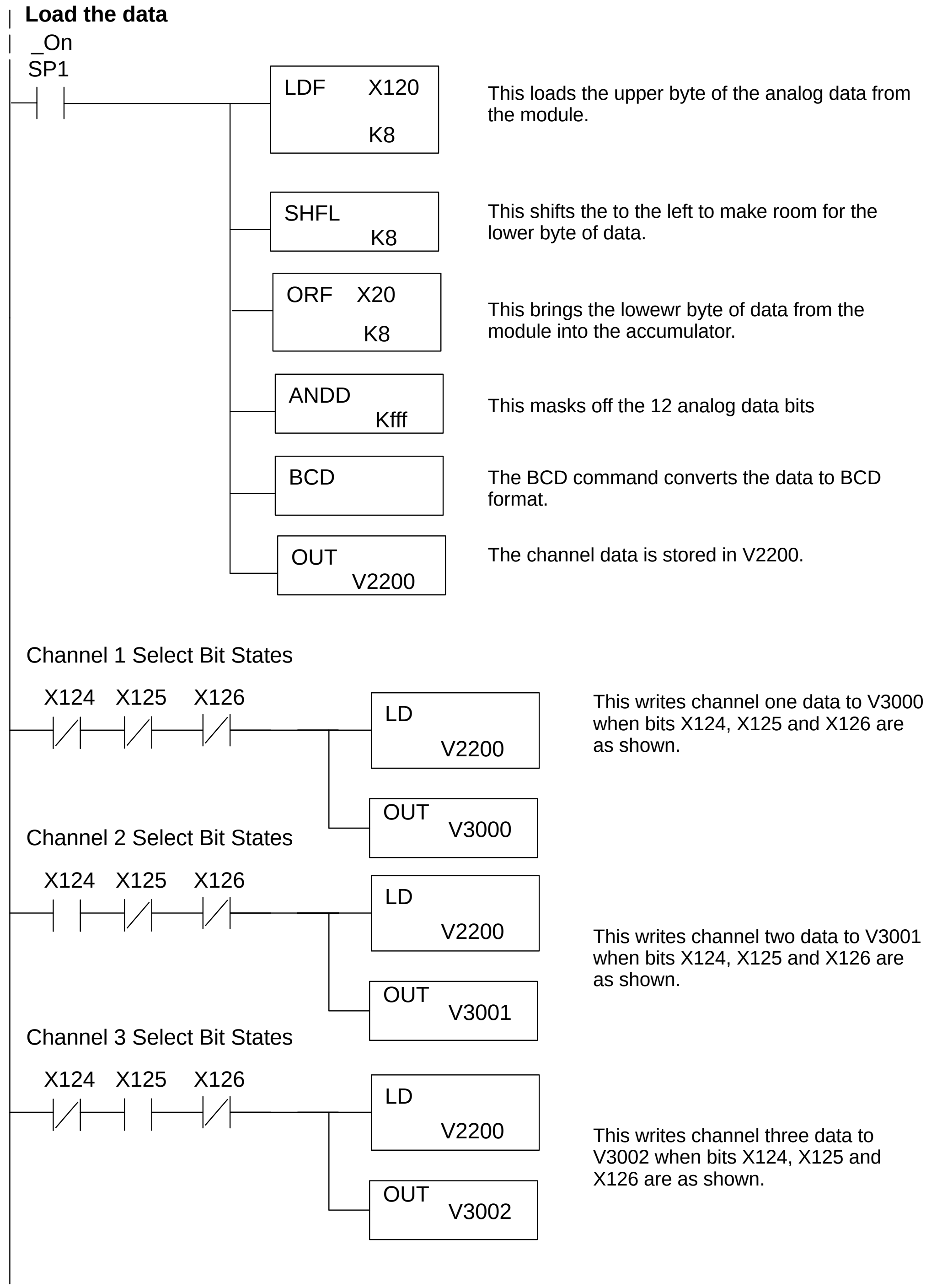


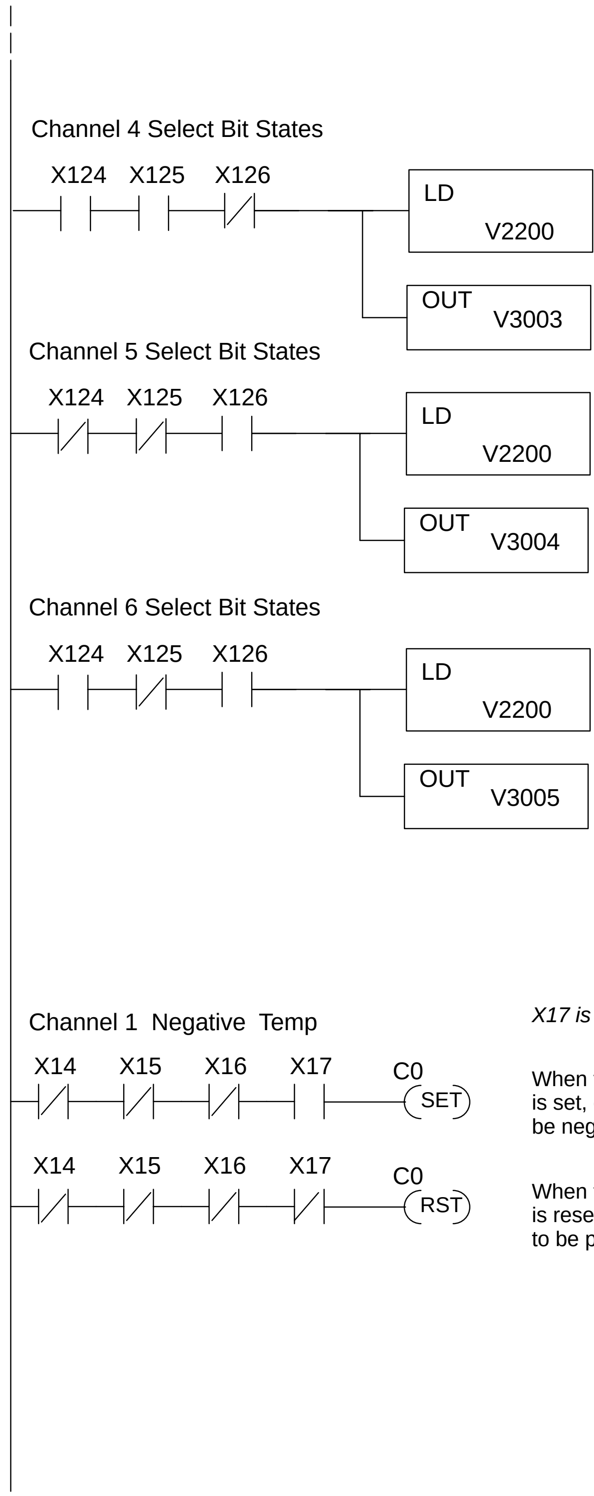
When the sign bit is on, the sign control relay (C0) is set, causing the temperature on channel one to be negative.

When the sign bit is not true, the sign bit control bit is reset, causing the temperature on channel one to be positive.

Multiplexing: DL350 with a Conventional DL305 Base

The example below shows how to read multiple channels on an F3–08THM Thermocouple module in the X20–X27 / 120 –127 address of a DL305 conventional base. The first six channels are shown.





This writes channel four data to V3003 when bits X124, X125 and X126 are as shown.

This writes channel five data to V3004 when bits X124, X125 and X126 are as shown.

This writes channel six data to V3005 when bits X14, X15 and X16 are as shown.

X17 is the sign bit when in module address 0.

When the sign bit is on, the sign control relay (C0) is set, causing the temperature on channel one to be negative.

When the sign bit is not true, the sign bit control bit is reset, causing the temperature on channel one to be positive.

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

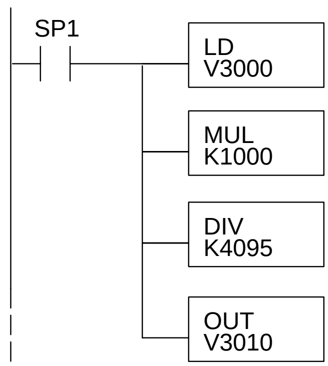
L = low limit of the engineering unit range

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.



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

Multiply the accumulator by 1000 (to start the conversion).

Divide the accumulator by 4095.

Store the result in V3010.

Temperature and Digital Value Conversions

Since the thermocouple devices are non-linear, it is much easier to rely on published standards for conversion information. The National Bureau of Standards publishes conversion tables that show how each temperature corresponds to an equivalent signal level.

Millivolt 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.

mV Range	If you know the digital value ...	If you know the analog signal level ...
MV50 0 to 50 mV	$A = \frac{50D}{4095}$	$D = \frac{4095}{50}A$
MV100 0 to 100 mV	$A = \frac{100D}{4095}$	$D = \frac{4095}{100}A$

For example, if you are using a –2 (100mV) version and you have measured the signal as 30 mV, you would use the following formula to determine the digital value that should be stored in the register location that contains the data.

$$D = \frac{4095}{100}A$$

$$D = \frac{4095}{100}(30)$$

$$D = (40.95)\ (30)$$

$$D = 1229$$

F3–08TEMP 8-Channel Temperature Input

In This Chapter. . . .

Module Specifications

The F3–08TEMP Temperature Input Module provides eight, single-ended temperature inputs for use with AD590 type temperature transmitters (range of 0–1mA.) The module provides 12-bit resolution. You can use the RLL control program to select between °F or °C operation.

The following table provides the specifications for the F3–08TEMP Temperature Input Module from FACTS Engineering. Review these specifications to make sure the module meets your application requirements.

Number of Channels	8, single-ended inputs
Input Ranges	0 – 1 mA
Resolution	12 bit (1 in 4096) No missing codes 0.25 °C with AD590M
Input Impedance	10KΩ ±0.1%
Absolute Maximum Ratings	±50 mA
Conversion Time	35μs per channel, maximum 1 channel per CPU scan
Converter Type	Successive Approximation, AD574
Linearity Error	±1 count (0.03% of full scale) maximum
Maximum Inaccuracy at 77 °F (25 °C)	0.25% of full scale
Accuracy at 25 °C	±1 °C with AD590M transmitter
Accuracy vs. Temperature	57 ppm / °C maximum full scale
Power Budget Requirement	25 mA @ 9 VDC, 37 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

Compatible Temperature Probe Specifications

The following table provides the specifications for input temperature probes compatible with this module.

Compatible Temperature Probe Specifications	
Transmitter Type	AD590
Input Temperature Range	–40° to 212° F (–40° to 100°C) – (Opto 22 PN ICTD) –67° to 302° F (–55° to 150°C) – (Analog Devices PN AC2626J)
Transmitter Output for Opto 22 and Analog Devices	1 μA / °K, 298.2 μA @ 25 °C 218 μA @ –55 °C, 423 μA @ 150 °C

Analog Input Configuration Requirements

The F3–08TEMP Temperature 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.

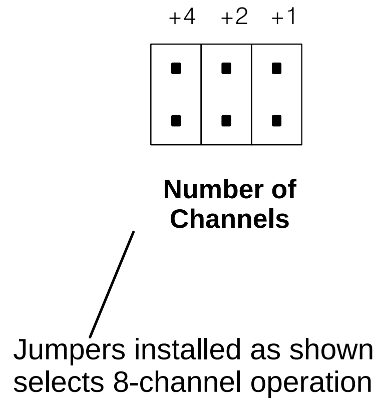
Setting the Module Jumpers

Factory Settings

The module is set at the factory for eight-channel operation. If this is acceptable you do not have to change any of the jumpers. The following diagram shows how the jumpers are set.

Selecting the Number of Channels

If you examine the rear of the module, you'll notice several jumpers. The jumpers labeled +1, +2 and +4 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 eight 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 1–4, then the last four channels will not be active. The following table shows which jumpers to install.

Channel(s)	+4	+2	+1
1	No	No	No
1 2	No	No	Yes
1 2 3	No	Yes	No
1 2 3 4	No	Yes	Yes
1 2 3 4 5	Yes	No	No
1 2 3 4 5 6	Yes	No	Yes
1 2 3 4 5 6 7	Yes	Yes	No
1 2 3 4 5 6 7 8	Yes	Yes	Yes

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

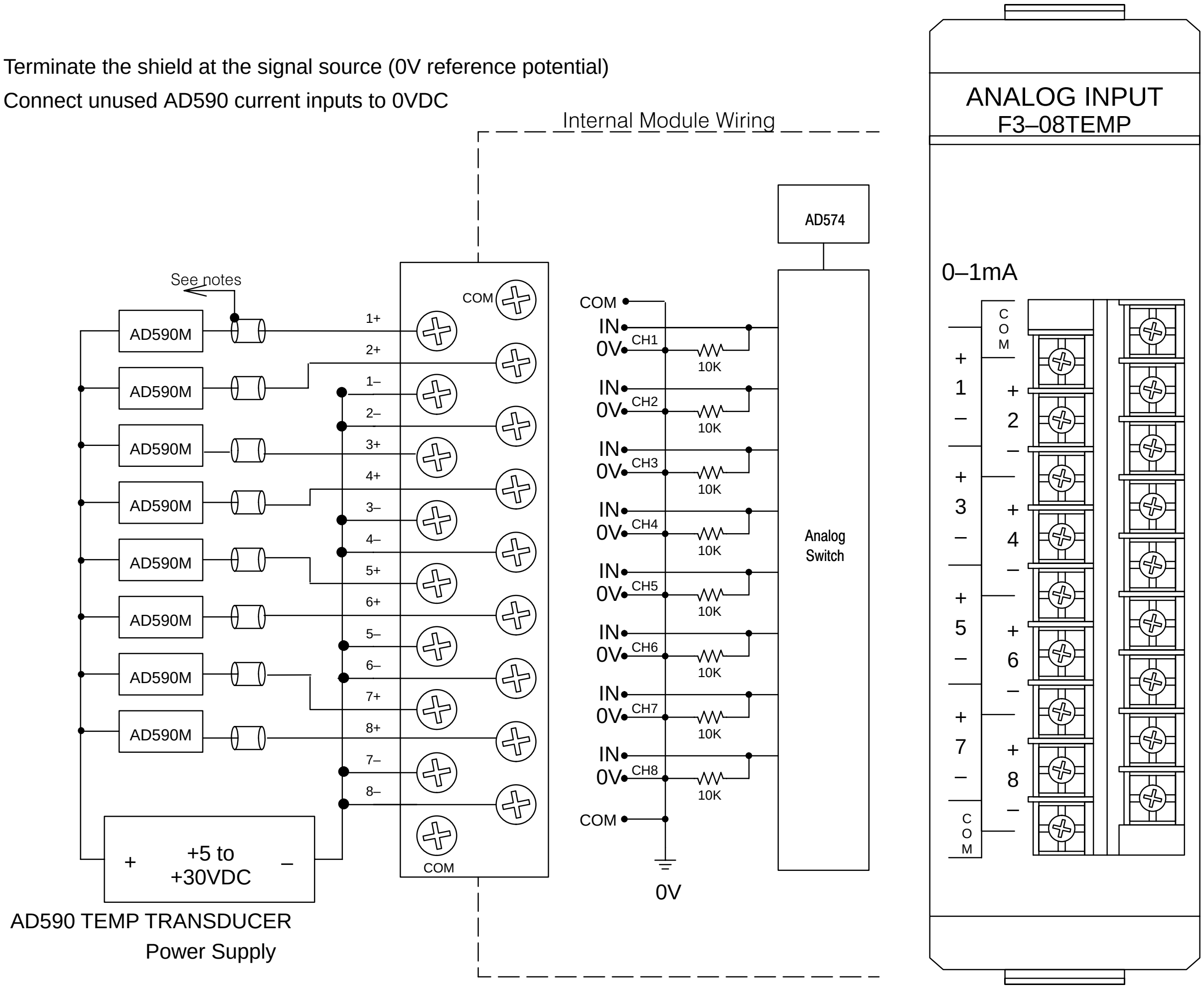
The F3–08TEMP receives all power from the base. A separate power supply is not required.

Removable Connector

The F3–08TEMP module has a removable connector to make wiring easier. Simply remove the retaining screws and gently pull the connector from the module.

Note 1: Terminate the shield at the signal source (0V reference potential)

Note 2: Connect unused AD590 current inputs to 0VDC



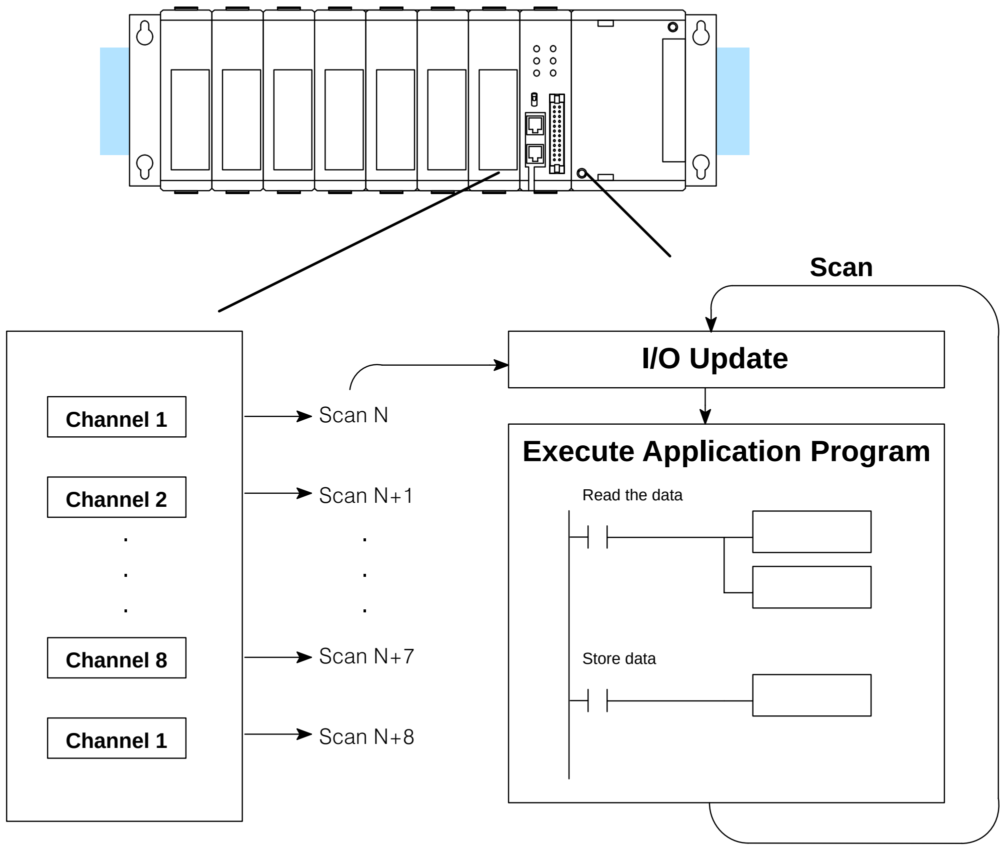
Module Operation

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 F3–08TEMP module supplies1 channel of data per each CPU scan. Since there are eight channels, it can take up to eight scans to get data for all channels. Once all channels have been scanned the process starts over with channel 1.

You do not have to select all of the channels. Unused channels are not processed, so if you select only four channels, then the channels will be updated within four scans.



Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the temperature 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.

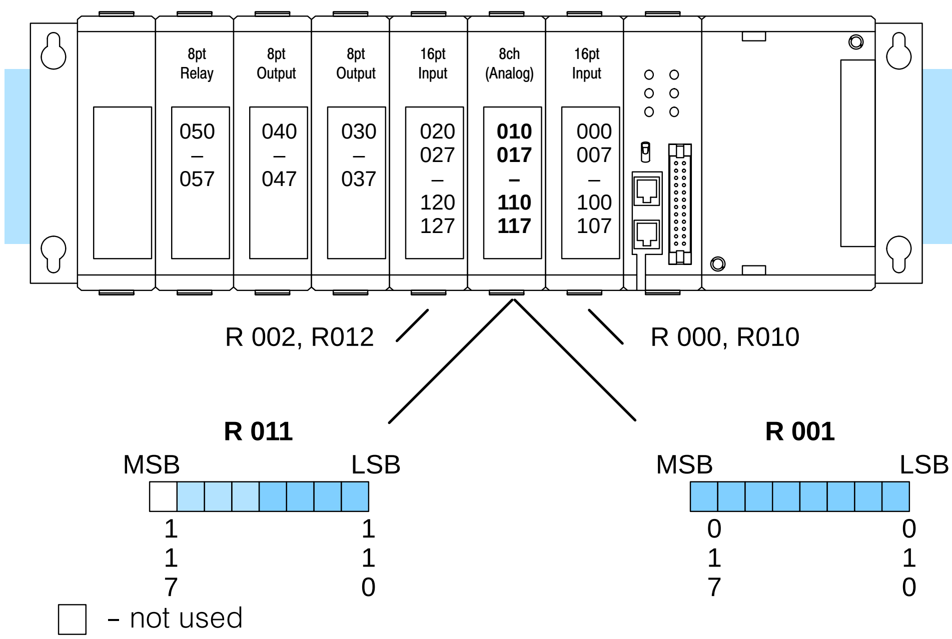
Understanding the I/O Assignments

You may recall the F3–08TEMP 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 temperature.

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.

F3–08TEMP

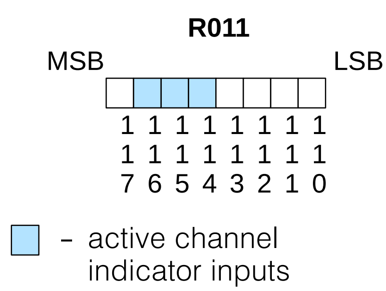


Within these two register locations, the individual bits represent specific information about the analog signal.

Active Channel Indicator Inputs

The next to last three bits of the upper Register indicate the active channel. The indicators automatically increment with each CPU scan.

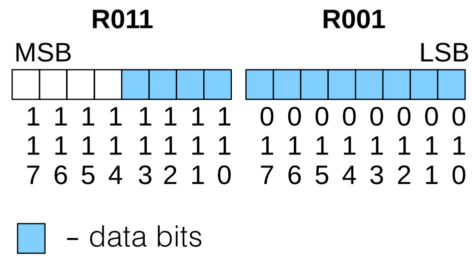
Scan	Active Channel Inputs	Channel
N	000	1
N+1	001	2
N+2	010	3
N+3	011	4
N+4	100	5
N+5	101	6
N+6	110	7
N+7	111	8
N+8	000	1



Analog Data Bits

The first twelve bits represent the temperature. The following format is used.

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



Temperature Input Resolution

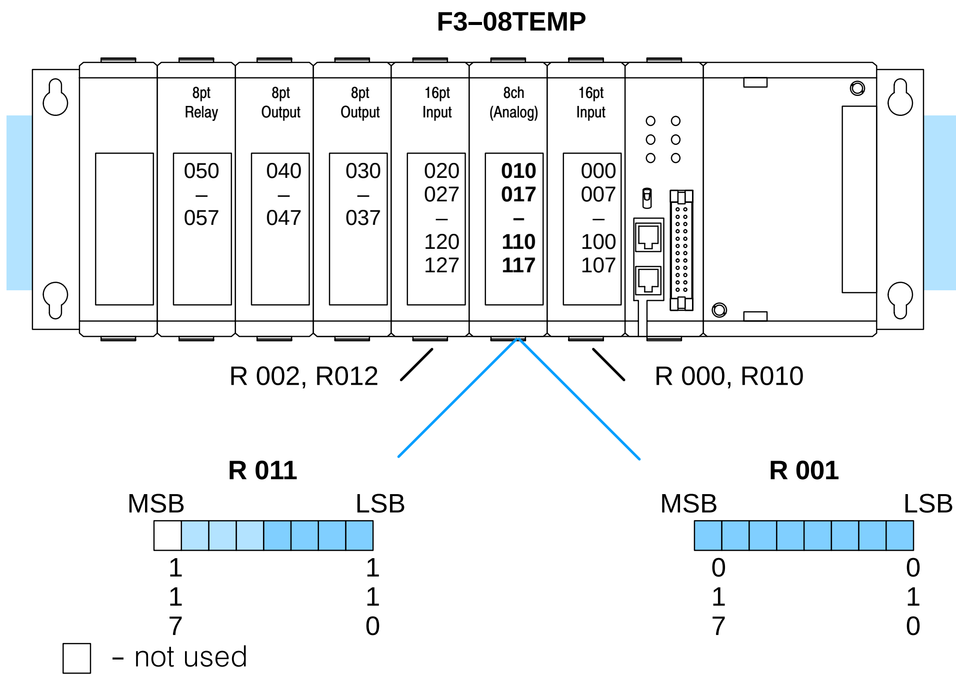
Typically, the F3–08TEMP resolution enables you to detect a 0.1 °F change in temperature.

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 (DL 330 / DL340)

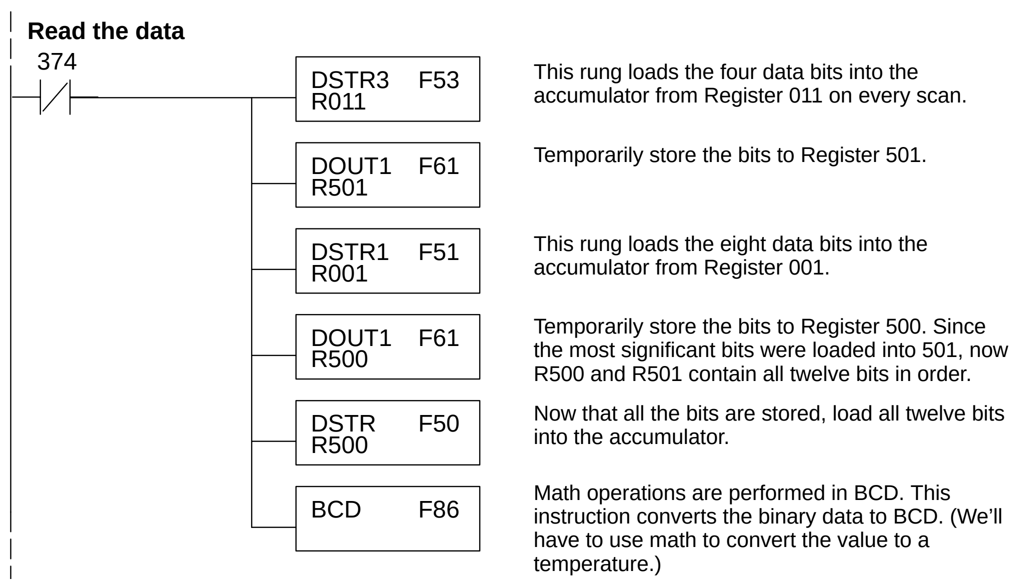
Identifying the Data Locations

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.



Reading the Digital Value

The following example program is designed to read any of the available channels of data . Once the data is read, you'll have to add some logic to convert the data into a °C or °F temperature. (More on the conversion in a minute. For now, let's just read the value into the accumulator.) 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.



Converting the Data to Temperature

Once the input data is stored in a register location, you will need to convert it to represent the temperature you are measuring. Use the formulas shown to convert the data to show the temperature in °C and °F.

For °C Readings

$$\text{Temp} = 1000\frac{A}{4096} - 273.2$$

Temp = temperature in °C

A = Analog value (0 – 4095)

273.2 = °K offset
(0 °K = –273.2 °C)

For °F Readings

$$\text{Temp} = 1000\frac{A}{2276} - 459.6$$

Temp = temperature in °F

A = Analog value (0 – 4095)

459.6 = °K offset
(0 °K = –459.6 °F)

The following example shows how you would use the analog data to represent the temperature. This example assumes the analog value is 1733. This should yield approximately 150 °C.

$$\text{Temp} = 1000\frac{1733}{4096} - 273.2$$

$$\downarrow$$

$$\text{Temp} = 149.9$$

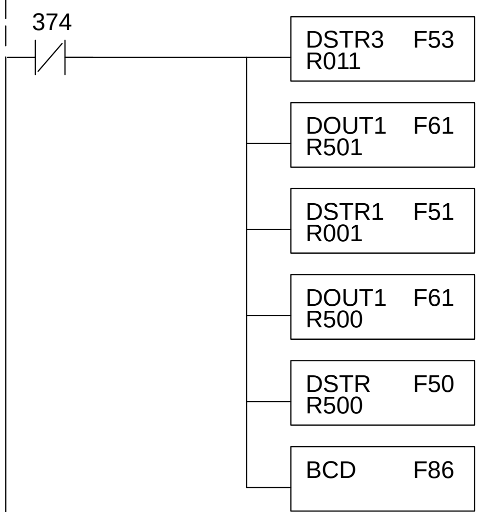
You can't quite enter the formula exactly as is with the DL305 instruction set. You have to use a value that implies the decimal point of precision. Plus, since we can move the decimal portion into the accumulator, we do not have to multiply the value by 1000.

The following instructions show you how to solve the conversion problem. (We'll continue to use the 150 °C example.)

NOTE: This example uses °C. To use °F, simply change the scaling factor and offset instructions to use the F formula.

- °**F scale** — Constant of 2276 for scaling factor, constant of 4596 for offset.
- °**C scale** — constant of 4096 for scaling factor, constant of 2732 for offset.

Read the data



This rung loads the four data bits into the accumulator from Register 011 on ever scan.

Temporarily store the bits to Register 501.

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

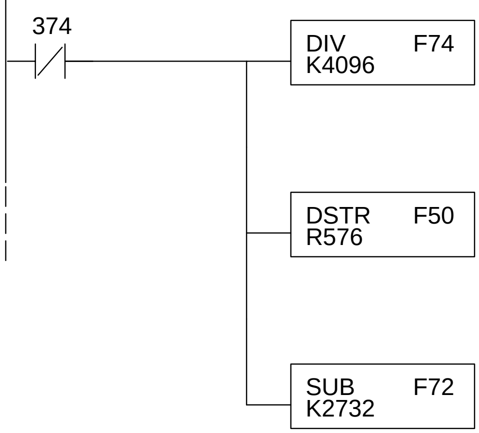
Temporarily store the bits to Register 500. Since the most significant bits were loaded into 501, now R500 and R501 contain all twelve bits in order.

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. (We'll have to use math to convert the value to a temperature.)

Accumulator				Aux. Accumulator			
1	7	3	3	0	0	0	0
				R577		R576	

Scale the data



The analog value is divided by °C scaling factor, which is 4096. (1733 / 4096 = 0.4230)

Accumulator				Aux. Accumulator			
0	0	0	0	4	2	3	0
				R577		R576	

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Accumulator				Aux. Accumulator			
4	2	3	0	4	2	3	0
				R577		R576	

Now subtract the °K offset from the accumulator. (The °K offset is 2732, which represents 273.2 °.

Accumulator					Constant			
4	2	3	0	—	2	7	3	2

Accumulator			
1	4	9	8

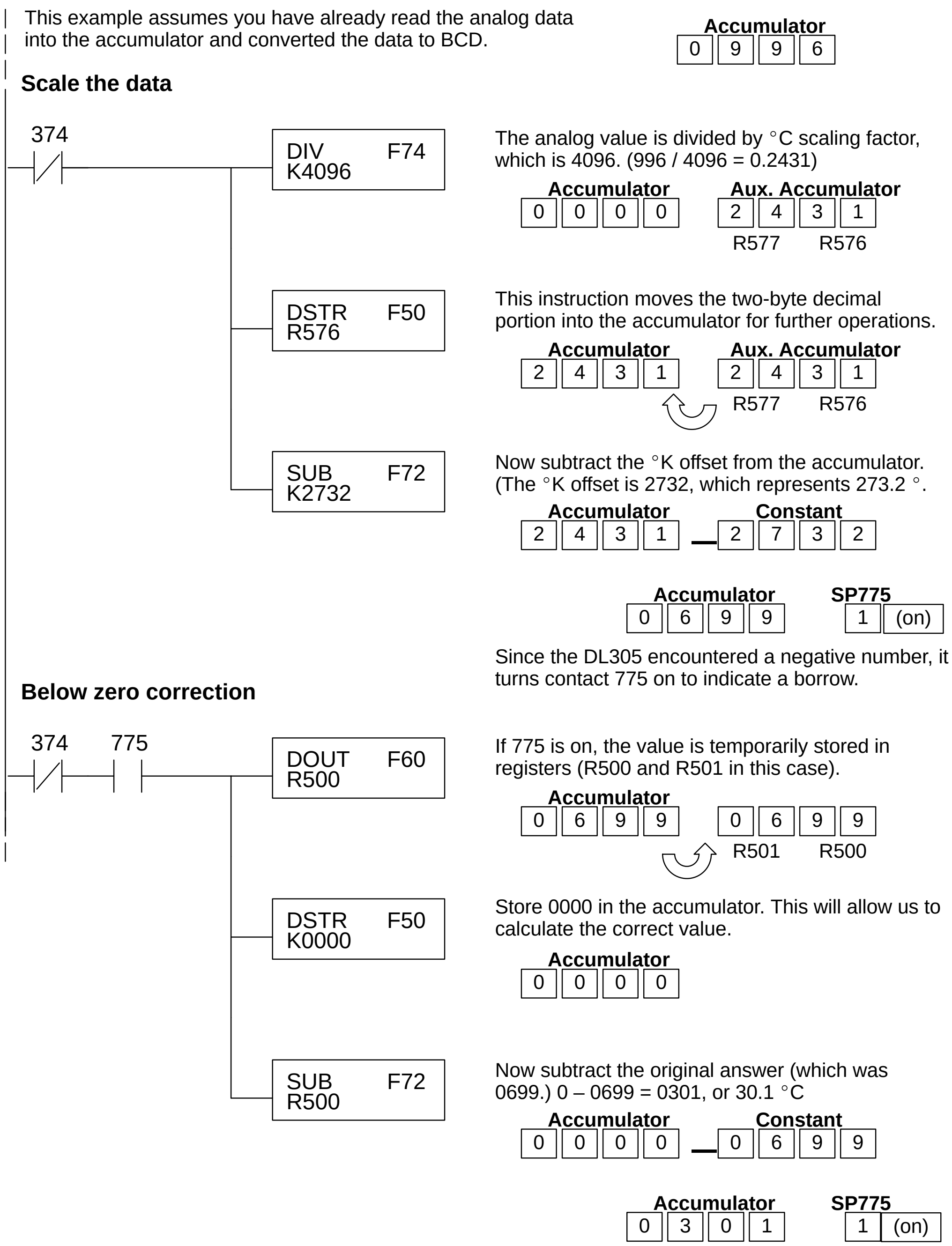
Reading Temperatures Below Zero

You have to perform some additional calculations if the temperature is below zero. Since the DL305 sets a special contact 775 if the subtraction results in a value below zero, you can use this to indicate further calculations are required. The following example shows the scaling and zero indication for a temperature of –30 C.

$$\text{Temp} = 1000\frac{996}{4096} - 273.2$$

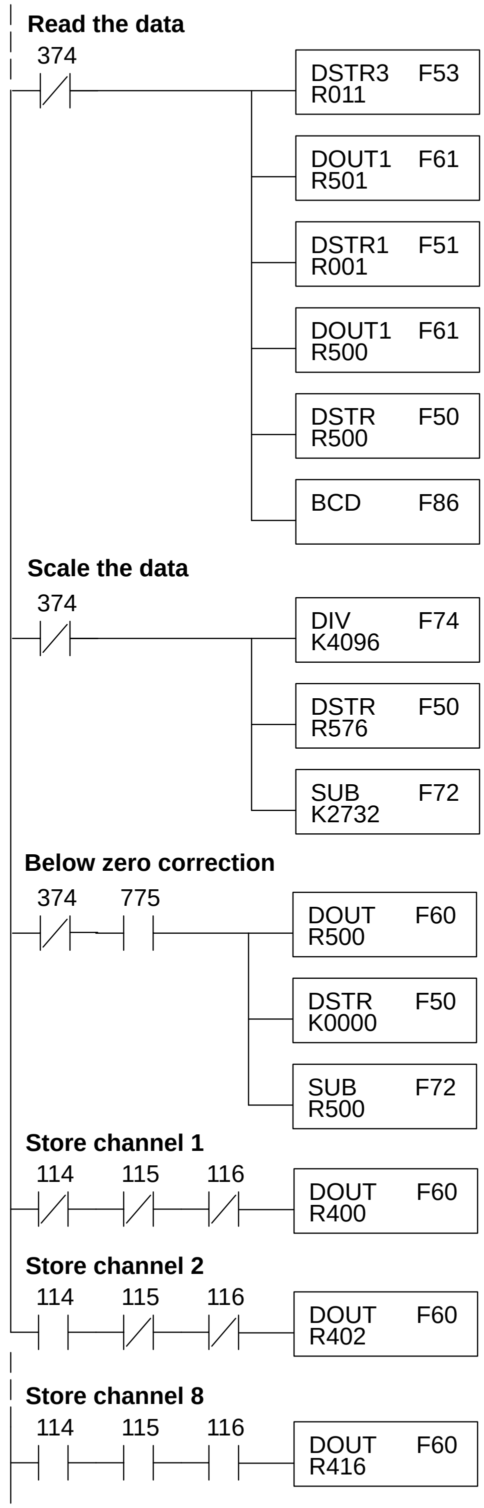
$$\downarrow$$

$$\text{Temp} = -30.0$$



Storing the Temperature

Once you've read the data and converted it to a temperature, you can use the channel selection inputs to store each of the eight channels. Once you've stored the data you can perform data comparisons, additional math, etc.



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

Temporarily store the bits to Register 501.

This rung loads the eight 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. (We'll have to use math to convert the value to a temperature.)

The analog value is divided by °C scaling factor, which is 4096. (1733 / 4096 = 0.4230)

This instruction moves the two-byte decimal portion into the accumulator for further operations.

Now subtract the °K offset from the accumulator. (The °K offset is 2732, which represents 273.2 °.

If 775 is on, the value is temporarily stored in registers (R500 and R501 in this case).

Store 0000 in the accumulator. This will allow us to calculate the correct value.

Now subtract the original answer (which was 0699.) 0 – 0699 = 0301, or 30.1 °C

Use the channel selection bits to determine which channel is active. For example, this rung stores channel 1 temperature in registers 400 and 401.

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

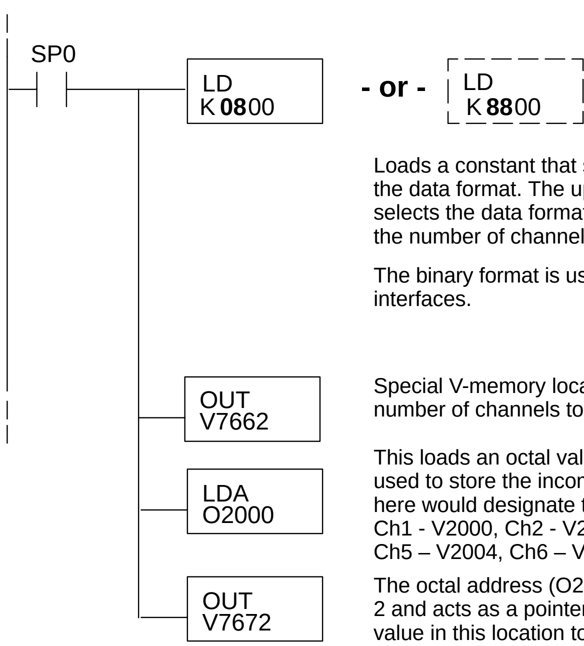
You must use the multiplexing method with remote I/O modules (the pointer method will not work). You can use either method when using DL350, but for ease of programming it is strongly recommended that you use the pointer method.

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, 5, 6, 7, 8).

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, Ch4 - V2003,
Ch5 – V2004, Ch6 – V2005, Ch7 – V2006, Ch8 – V2007

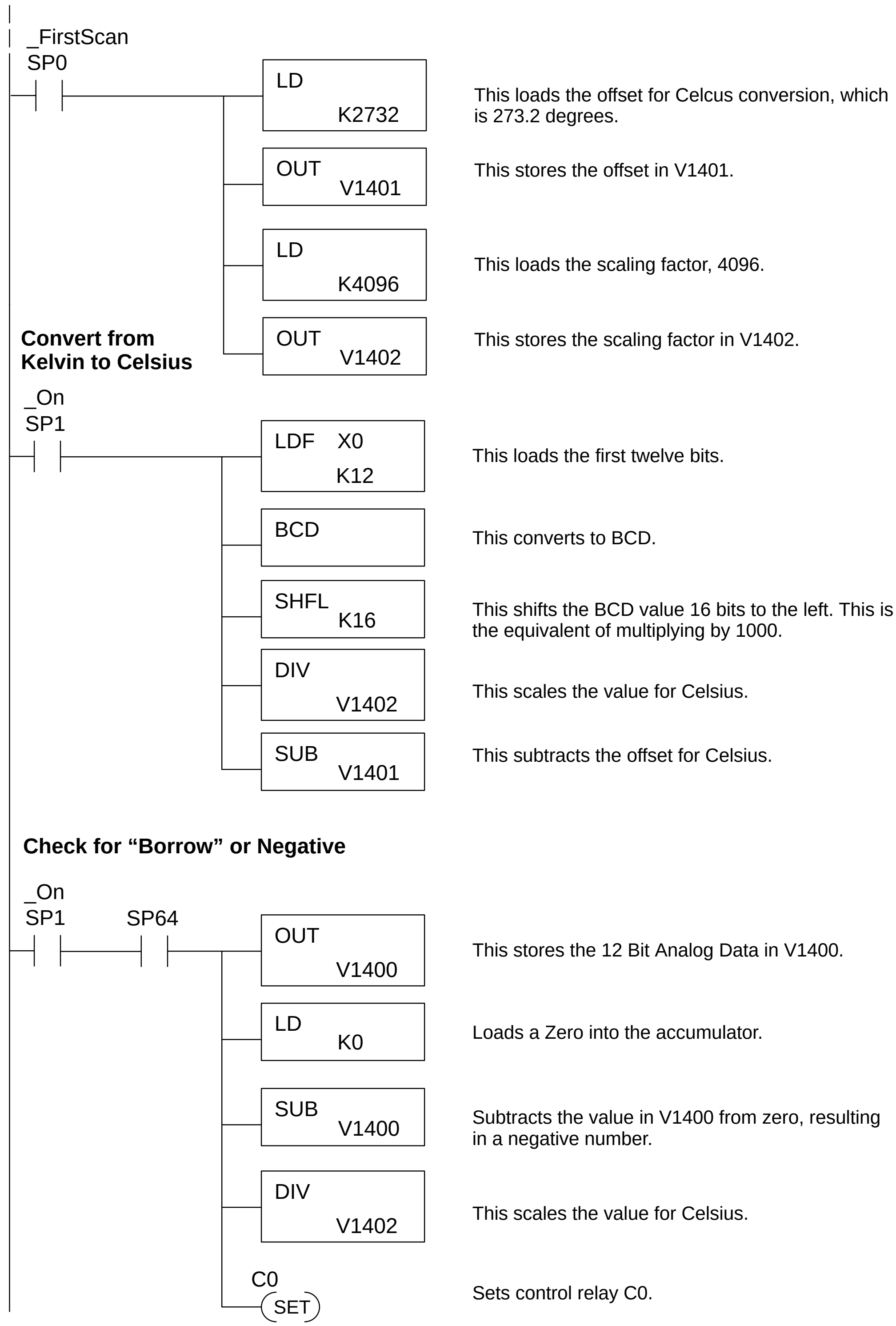
The octal address (O2000) is stored here. V7672 is assigned to slot 2 and acts as a pointer, which means the CPU will use the octal value in this location to determine exactly where to store the incoming data.

The 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

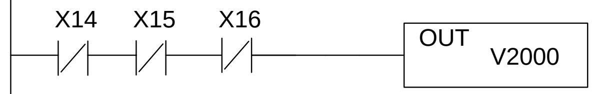
Multiplexing: DL350 with a D3–XX–1 Base

The example below shows how to read an Analog Devices AD590 temperature transducer on an F3–08TEMP Temperature Input module in the X0 address 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.



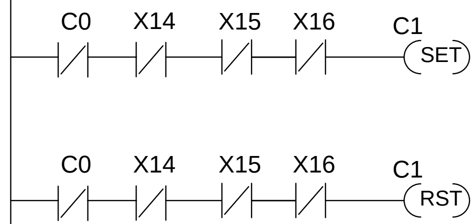
When the appropriate channel select bits are turned on, the converted analog data is then stored in V-memory locations, starting with V2000 (for channel 1).

Channel 1 Select Bit States



This writes channel one data to V2000 when bits X14, X15 and X16 are as shown.

The negative indicator bit

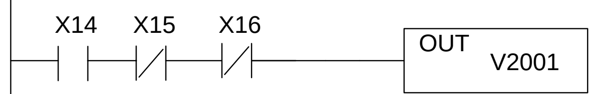


These two rungs control the negative indicator bit. When the channels select bits are true for a particular channel and C0 is on, the negative bit for that channel is set. When the temperature goes above 0 Celsius, the bit is reset.

Notice that this only applies to Channels 1.

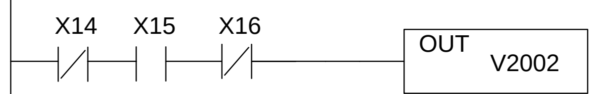
The remaining channels are shown below, without covering the negative bit logic.

Channel 2 Select Bit States

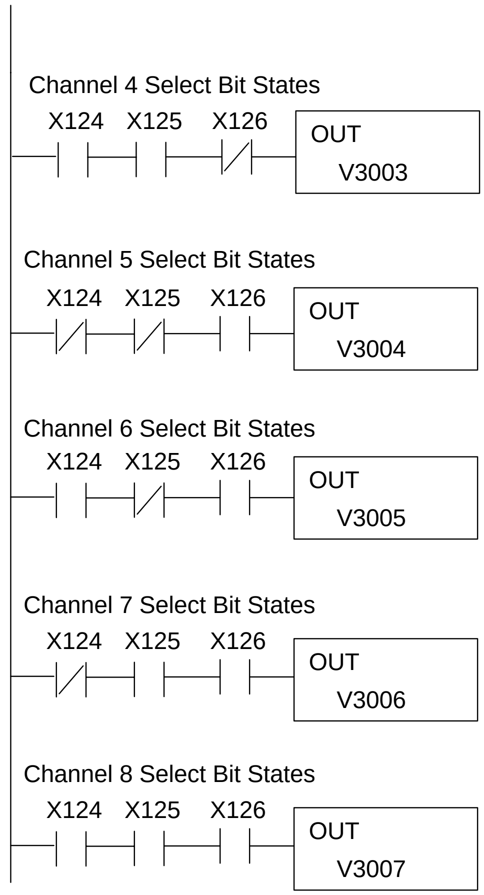


This writes channel two data to V2001 when bits X14, X15 and X16 are as shown.

Channel 3 Select Bit States



This writes channel three data to V2002 when bits X14, X15 and X16 are as shown.



This writes channel four analog data to V3003 when bits X124, X125 and X126 are as shown.

This writes channel five analog data to V3004 when bits X124, X125 and X126 are as shown.

This writes channel six analog data to V3005 when bits X124, X125 and X126 are as shown.

This writes channel seven analog data to V3006 when bits X124, X125 and X126 are as shown.

This writes channel eight analog data to V3007 when bits X124, X125 and X126 are as shown.

Temperature 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 temperature ...
–55 to150 °C	$T = \frac{1000D}{4095} - 273.2$	$D = \frac{4095}{1000}(T + 273.2)$
–67 to 302 °F	$T = \frac{1000D}{2276} - 459.6$	$D = \frac{2276}{1000}(T + 459.6)$

For example, if you have measured the temperature at 30 °C, you would use the following formula to determine the digital value that should be stored in the register location that contains the temperature.

$$D = \frac{4095}{1000}(T + 273.2)$$

$$D = \frac{4095}{1000}(30 + 273.2)$$

$$D = (4.095)\,(303.2)$$

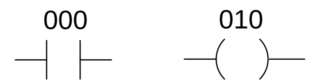
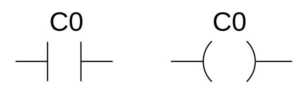
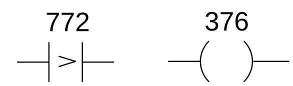
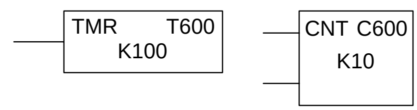
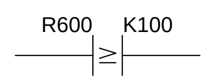
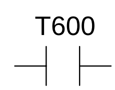
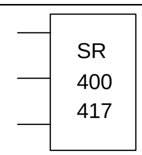
$$D = 1241$$

DL305 Data Types and Memory Map

A

In This Chapter. . . .

DL330 Memory Map

Memory Type	Discrete Memory Reference (octal)	Register Memory Reference (octal)	Qty. Decimal	Symbol
Input / Output Points	000 – 157 700 – 767	R000 – R015 R070 – R076	168 Total	
Control Relays	160 – 373	R016 – R037	140	
Special Relays	374 – 377 770 – 777	R037 R077	12	
Timers / Counters	600 – 673 674 – 677*	None	64	
Timer / Counter Current Values	None	R600 – R673 R674 – R677*	64	
Timer / Counter Status Bits	T600 – T673 T674 – T677*	None	64	
Data Words	None	R400 – R563	116	None specific, used with many instructions
Shift Registers	400 – 577	None	128	
Special Registers	None	R574 – R577	4	R574 – R575 used with FAULT R576 – R577 Auxiliary Accumulator

* T/ C Setpoint Unit Only. Can be used as data registers if the Timer/Counter Setpoint Unit or Thumbwheel Interface Module is not used. R564 – R573 contain the preset value used with the Timer / Counter Setpoint Unit. R674 – R677 contain the current values for these timers or counters.

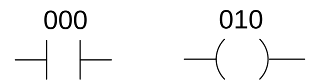
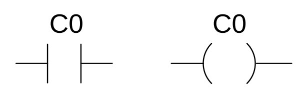
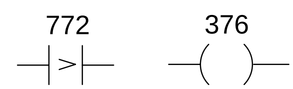
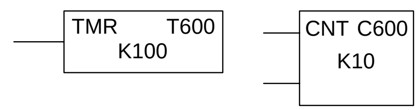
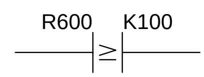
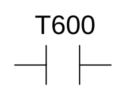
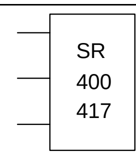
DL330P Memory Map

Memory Type	Discrete Memory Reference (octal)	Register Memory Reference (octal)	Qty. Decimal	Symbol
Input / Output Points	000 – 157 700 – 767	R000 – R015 R070 – R076	168 Total	000 010
Control Relays	160 – 174 200 – 277	R016 – R017 R020 – R027	77	C0 C0
Special Relays	175 – 177 770 – 777	R017 R077	11	772 > 176
Timers / Counters	600 – 673 674 – 677*	None	64	TMR T600 K100 CNT C600 K10
Timer / Counter Current Values	None	R600 – R673 R674 – R677*	64	R600 K100 ≥
Timer / Counter Status Bits	T600 – T673 T674 – T677*	None	64	T600
Data Words	None	R400 – R563	116	None specific, used with many instructions
Stages	S0 – S177	R100 – R117	128	SG S 001 S1
Special Registers	None	R574 – R577	4	R574 – R575 used with FAULT R576 – R577 Auxiliary Accumulator

* T/ C Setpoint Unit Only. Can be used as data registers if the Timer/Counter Setpoint Unit or Thumbwheel Interface Module is not used, which provides a total of 128 data registers.

R564 – R573 contain the preset value used with the Timer / Counter Setpoint Unit. R674 – R677 contain the current values for these timers or counters.

DL340 Memory Map

Memory Type	Discrete Memory Reference (octal)	Register Memory Reference (octal)	Qty. Decimal	Symbol
Input / Output Points	000 – 157 700 – 767	R000 – R015 R070 – R076	168 Total	
Control Relays	160 – 373 1000 – 1067	R016 – R037 R100 – R106	180	
Special Relays	374 – 377 770 – 777 1070 – 1077	R037 R077 R107	20	
Timers / Counters	600 – 673 674 – 677*	None	64	
Timer / Counter Current Values	None	R600 – R673 R674 – R677*	64	
Timer / Counter Status Bits	T600 – T673 T674 – T677*	None	64	
Data Words	None	R400 – R563 R700 – R767	172	None specific, used with many instructions
Shift Registers	400 – 577	None	128	
Special Registers	None	R574 – R577 R770 – R777	12	R574–R575 used with FAULT R576–R577 Auxiliary Accumulator R770–R777 Communications Setup

* T/ C Setpoint Unit Only. Can be used as data registers if the Timer/Counter Setpoint Unit or Thumbwheel Interface Module is not used. R564 – R573 contain the preset value used with the Timer / Counter Setpoint Unit. R674 – R677 contain the current values for these timers or counters.

I/O Point Bit Map

These tables provide a listing of the individual Input points associated with each register location for the DL330, DL330P, and DL340 CPUs.

MSB			I/O References				LSB	Register Number
007	006	005	004	003	002	001	000	R0
017	016	015	014	013	012	011	010	R1
027	026	025	024	023	022	021	020	R2
037	036	035	034	033	032	031	030	R3
047	046	045	044	043	042	041	040	R4
057	056	055	054	053	052	051	050	R5
067	066	065	064	063	062	061	060	R6
077	076	075	074	073	072	071	070	R7
107	106	105	104	103	102	101	100	R10
117	116	115	114	113	112	111	110	R11
127	126	125	124	123	122	121	120	R12
137	136	135	134	133	132	131	130	R13
147	146	145	144	143	142	141	140	R14
157	156	155	154	153	152	151	150	R15
167	166	165	164	163	162	161	160	n/a
177	176	175	174	173	172	171	170	n/a
707	706	705	704	703	702	701	700	R70
717	716	715	714	713	712	711	710	R71
727	726	725	724	723	722	721	720	R72
737	736	735	734	733	732	731	730	R73
747	746	745	744	743	742	741	740	R74
757	756	755	754	753	752	751	750	R75
767	766	765	764	763	762	761	760	R76

NOTE: 160 – 167 can be used as I/O in a DL330 or DL330P CPU under certain conditions. 160 – 177 can be used as I/O in a DL340 CPU under certain conditions. You should consult the DL305 User Manual to determine which configurations allow the use of these points.

These points are normally used as control relays. You cannot use them as both control relays and as I/O points. Also, if you use these points as I/O, you cannot access these I/O points as a Data Register reference.

Control Relay Bit Map

The following tables provide a listing of the individual control relays associated with each register location for the DL305 CPUs.

NOTE: 160 – 167 can be used as I/O in a DL330 or DL330P CPU under certain conditions. 160 – 177 can be used as I/O in a DL340 CPU under certain conditions. You should consult the DL305 User Manual to determine which configurations allow the use of these points.

You cannot use them as both control relays and as I/O points. Also, if you use these points as I/O, you cannot access these I/O points as a Data Register reference.

MSB			DL330 Control Relay References				LSB	Register Number
167	166	165	164	163	162	161	160	R16
177	176	175	174	173	172	171	170	R17
207	206	205	204	203	202	201	200	R20
217	216	215	214	213	212	211	210	R21
227	226	225	224	223	222	221	220	R22
237	236	235	234	233	232	231	230	R23
247	246	245	244	243	242	241	240	R24
257	256	255	254	253	252	251	250	R25
267	266	265	264	263	262	261	260	R26
277	276	275	274	273	272	271	270	R27
307	306	305	304	303	302	301	300	R30
317	316	315	314	313	312	311	310	R31
327	326	325	324	323	322	321	320	R32
337	336	335	334	333	332	331	330	R33
347	346	345	344	343	342	341	340	R34
357	356	355	354	353	352	351	350	R35
367	366	365	364	363	362	361	360	R36
				373	372	371	370	R37

* Control relays 340 – 373 can be made retentive by setting a CPU dipswitch. See the DL305 User Manual for details on setting CPU dipswitches.

MSB			DL330P Control Relay References				LSB	Register Number
167	166	165	164	163	162	161	160	R16
			174	173	172	171	170	R17
207	206	205	204	203	202	201	200*	R20
217	216	215	214	213	212	211	210	R21
227	226	225	224	223	222	221	220	R22
237	236	235	234	233	232	231	230	R23
247	246	245	244	243	242	241	240	R24
257	256	255	254	253	252	251	250	R25
267	266	265	264	263	262	261	260	R26
277*	276	275	274	273	272	271	270	R27

* Control relays 200 – 277 can be made retentive by setting a CPU dipswitch. See the DL305 User Manual for details on setting CPU dipswitches.

MSB			DL340 Control Relay References				LSB	Register Number
167	166	165	164	163	162	161	160	R16
177	176	175	174	173	172	171	170	R17
207	206	205	204	203	202	201	200	R20
217	216	215	214	213	212	211	210	R21
227	226	225	224	223	222	221	220	R22
237	236	235	234	233	232	231	230	R23
247	246	245	244	243	242	241	240	R24
257	256	255	254	253	252	251	250	R25
267	266	265	264	263	262	261	260	R26
277	276	275	274	273	272	271	270	R27
307	306	305	304	303	302	301	300	R30
317	316	315	314	313	312	311	310	R31
327	326	325	324	323	322	321	320	R32
337	336	335	334	333	332	331	330	R33
347	346	345	344	343	342	341	340*	R34
357	356	355	354	353	352	351	350	R35
367	366	365	364	363	362	361	360	R36
				373*	372	371	370	R37
1007	1006	1005	1004	1003	1002	1001	1000	R100
1017	1016	1015	1014	1013	1012	1011	1010	R101
1027	1026	1025	1024	1023	1022	1021	1020	R102
1037	1036	1035	1034	1033	1032	1031	1030	R103
1047	1046	1045	1044	1043	1042	1041	1040	R104
1057	1056	1055	1054	1053	1052	1051	1050	R105
1067	1066	1065	1064	1063	1062	1061	1060	R106

* Control relays 340 – 373 can be made retentive by setting a CPU dipswitch. See the DL305 User Manual for details on setting CPU dipswitches.

Special Relays

The following table shows the Special Relays used with the DL305 CPUs.

CPUs	Special Relay	Description of Contents
DL330P	175	100 ms clock, on for 50 ms and off for 50 ms.
	176	Disables all outputs except for those entered with the SET OUT instruction.
	177	Battery voltage is low.
DL330 DL340	374	On for the first scan cycle after the CPU is switched to Run Mode.
	375	100 ms clock, on for 50 ms and off for 50 ms.
	376	Disables all outputs except for those entered with the SET OUT instruction.
	377	Battery voltage is low.
DL330 DL330P DL340	770	Changes timers to 0.01 second intervals. Timers are normally 0.1 second time intervals.
	771	The external diagnostics FAULT instruction (F20) is in use.
	772	The data in the accumulator is greater than the comparison value.
	773	The data in the accumulator is equal to the comparison value.
	774	The data in the accumulator is less than the comparison value.
	775	An accumulator carry or borrow condition has occurred.
	776	The accumulator value is zero.
	777	The accumulator has an overflow condition.
DL340	1074	The RX or WX instruction is active.
	1075	An error occurred during communications with the RX or WX instructions.
	1076	Port 2 communications mode: on = ASCII mode, off = HEX mode
	1077	Port 1 communications mode: on = ASCII mode, off = HEX mode

Data Registers

The following 8-bit data registers are primarily used with data instructions to store various types of application data. For example, you could use a register to hold a timer or counter preset value.

Some data instructions call for two bytes, which will correspond to two consecutive 8-bit data registers such as R401 and R400. The LSB (Least Significant Bit) will be in register R400 as bit0 and the MSB (Most Significant Bit) will be in register R401 as bit17.

NOTE: Data Registers are retentive.

DL330 / DL330P 8-Bit Data Registers							
407	406	405	404	403	402	401	400
417	416	415	414	413	412	411	410
427	426	425	424	423	422	421	420
437	436	435	434	433	432	431	430
447	446	445	444	443	442	441	440
457	456	455	454	453	452	451	450
467	466	465	464	463	462	461	460
477	476	475	474	473	472	471	470
507	506	505	504	503	502	501	500
517	516	515	514	513	512	511	510
527	526	525	524	523	522	521	520
537	536	535	534	533	532	531	530
547	546	545	544	543	542	541	540
557	556	555	554	553	552	551	550
				563	562	561	560

DL340 8-Bit Data Registers							
407	406	405	404	403	402	401	400
417	416	415	414	413	412	411	410
427	426	425	424	423	422	421	420
437	436	435	434	433	432	431	430
447	446	445	444	443	442	441	440
457	456	455	454	453	452	451	450
467	466	465	464	463	462	461	460
477	476	475	474	473	472	471	470
507	506	505	504	503	502	501	500
517	516	515	514	513	512	511	510
527	526	525	524	523	522	521	520
537	536	535	534	533	532	531	530
547	546	545	544	543	542	541	540
557	556	555	554	553	552	551	550
				563	562	561	560
707	706	705	704	703	702	701	700
717	716	715	714	713	712	711	710
727	726	725	724	723	722	721	720
737	736	735	734	733	732	731	730
747	746	745	744	743	742	741	740
757	756	755	754	753	752	751	750
767	766	765	764	763	762	761	760

DL350 System V-memory

System V-memory	Description of Contents	Default Values / Ranges
V7620–V7627	Locations for DV–1000 operator interface parameters	
V7620	Sets the V-memory location that contains the value.	V0 – V3777
V7621	Sets the V-memory location that contains the message.	V0 – V3777
V7622	Sets the total number (1 – 16) of V-memory locations to be displayed.	1 – 16
V7623	Sets the V-memory location that contains the numbers to be displayed.	V0 – V3777
V7624	Sets the V-memory location that contains the character code to be displayed.	V0 – V3777
V7625	Contains the function number that can be assigned to each key.	V-memory for X, Y, or C
V7626	Reserved	0,1,2,3,12
V7627	Reserved	Default=0000
V7630–V7632	Reserved	–
V7633	User defined timer interrupt/operation of battery/Binary instruction sign flag* Bit 0–7 40H Setting Interrupt Bit 12 ON with battery sign flag. ON use sign flag – OFF no sign flag Bit 15 Binary instruction sign flag. ON use sign flag – OFF no sign flag	
V7634	User defined timer interrupt	
V7640	Loop Table Beginning address	V1400–V7340
V7641	Number of Loops Enabled	1–4
V7642	Error Code – V–memory Error Location for Loop Table	
V7643–V7647	Reserved	
V7650	Port 2 End–code setting Setting (A55A), Nonprocedure communications start.	
V7651	Port 2 Data format –Non–procedure communications format setting.	
V7652	Port 2 Format Type setting – Non–procedure communications type code setting.	
V7653	Port 2 Terminate–code setting – Non–procedure communications Termination code setting.	
V7654	Port 2 Store V–mem address – Non–procedure communication data store V–Memory address.	
V7655	Port 2 Setup area –0–7 Comm protocol (flag 0) 8–15 Comm time out/response delay time (flag 1)	
V7656	Port 2 setup area – 0–15 Communication (flag2, flag 3)	
V7657	Port 2 setup area – Bit to select use of parameter	
V7660–V7707	Set–up Information	
V7710–V7717	Reserved	
V7720–V7722	Locations for DV–1000 operator interface parameters.	
V7720	Titled Timer preset value pointer	
V7721	Title Counter preset value pointer	
V7722	HiByte-Titled Timer preset block size, LoByte-Titled Counter preset block size	
V7730–V7737	For slot 0 to 7 D3–DCM	
V7747	Location contains a 10ms counter. This location increments once every 10ms.	
V7750	Reserved	

System V-memory	Description of Contents
V7751	Fault Message Error Code — stores the 4-digit code used with the FAULT instruction when the instruction is executed.
V7752	Reserved
V7753	Reserved
V7754	Reserved
V7755	Error code — stores the fatal error code.
V7756	Error code — stores the major error code.
V7757	Error code — stores the minor error code.
V7760–V7762	Reserved
V7763–V7764	Location for syntax error information.
V7765	Scan — stores the total number of scan cycles that have occurred since the last Program Mode to Run Mode transition.
V7766	Contains the number of seconds on the clock. (00 to 59).
V7767	Contains the number of minutes on the clock. (00 to 59).
V7770	Contains the number of hours on the clock. (00 to 23).
V7771	Contains the day of the week. (Mon, Tue, etc.).
V7772	Contains the day of the month (1st, 2nd, etc.).
V7773	Contains the month. (01 to 12)
V7774	Contains the year. (00 to 99)
V7775	Scan — stores the current scan time (milliseconds).
V7776	Scan — stores the minimum scan time that has occurred since the last Program Mode to Run Mode transition (milliseconds).
V7777	Scan — stores the maximum scan time that has occurred since the last Program Mode to Run Mode transition (milliseconds).

DL350 Comm Port 2 Control Relays

The following system control relays are valid only for D3–350 CPU remote I/O setup on Communications Port 2.

System CRs	Description of Contents
C740	Completion of setups – ladder logic must turn this relay on when it has finished writing to the Remote I/O setup table
C741	Erase received data – turning on this flag will erase the received data during a communication error.
C743	Re-start – Turning on this relay will resume after a communications hang-up on an error.
C750 to C757	Setup Error – The corresponding relay will be ON if the setup table contains an error (C750 = master, C751 = slave 1... C757=slave 7
C760 to C767	Communications Ready – The corresponding relay will be ON if the setup table data is valid (C760 = master, C761 = slave 1... C767=slave 7

DL350 Memory Map

Memory Type	Discrete Memory Reference (octal)	Word Memory Reference (octal)	Qty. Decimal	Symbol
Input Points	X0 – X777	V40400 – V40437	512	X0
Output Points	Y0 – Y777	V40500 – V40537	512	Y0
Control Relays	C0 – C1777	V40600 – V40677	1024	C0 C0
Special Relays	SP0 – SP777	V41200 – V41237	512	SP0
Timer Current Values	None	V0 – V377	256	V0 K100 ≥
Timer Status Bits	T0 – T377	V41100 – V41117	256	T0
Counter Current Values	None	V1000 – V1177	128	V1000 K100 ≥
Counter Status Bits	CT0 – CT177	V41140 – V41147	128	CT0
Data Words	none	V1400 – V7377 V10000–V17777	3072 4096	None specific, used with many instructions
Stages	S0 – S1777	V41000 – V41077	1024	SG S 001 S0
System parameters	None	V7400–V7777	256	System specific, used for various purposes

DL 350 X Input / Y Output Bit Map

This table provides a listing of the individual Input points associated with each V-memory address bit.

MSB						DL350 Input (X) and Output (Y) Points									LSB	X Input Address	Y Output Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40400	V40500
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40401	V40501
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40402	V40502
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40403	V40503
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40404	V40504
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40405	V40505
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40406	V40506
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40407	V40507
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V40410	V40510
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V40411	V40511
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V40412	V40512
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V40413	V40513
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V40414	V40514
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V40415	V40515
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V40416	V40516
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V40417	V40517
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V40420	V40520
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V40421	V40521
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V40422	V40522
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V40423	V40523
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V40424	V40524
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V40425	V40525
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V40426	V40526
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V40427	V40527
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V40430	V40530
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V40431	V40531
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V40432	V40532
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V40433	V40533
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V40434	V40534
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V40435	V40535
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V40436	V40536
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V40437	V40537

DL350 Control Relay Bit Map

This table provides a listing of the individual control relays associated with each V-memory address bit.

MSB						DL350 Control Relays (C)									LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40600
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40601
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40602
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40603
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40604
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40605
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40606
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40607
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V40610
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V40611
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V40612
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V40613
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V40614
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V40615
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V40616
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V40617
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V40620
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V40621
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V40622
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V40623
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V40624
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V40625
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V40626
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V40627
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V40630
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V40631
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V40632
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V40633
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V40634
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V40635
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V40636
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V40637

MSB							Additional DL350 Control Relays (C)								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	V40640
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	V40641
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	V40642
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	V40643
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	V40644
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	V40645
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	V40646
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	V40647
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	V40650
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	V40651
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	V40652
1277	1276	1275	1274	1273	1272	1271	1270	1267	1266	1265	1264	1263	1262	1261	1260	V40653
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	V40654
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	V40655
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	V40656
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	V40657
1417	1416	1415	1414	1413	1412	1411	1410	1407	1406	1405	1404	1403	1402	1401	1400	V40660
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	V40661
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	V40662
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	V40663
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	V40664
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	V40665
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	V40666
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	V40667
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	V40670
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	V40671
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	V40672
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	V40673
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	V40674
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	V40675
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	V40676
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	V40677

DL350 Stage™ Control / Status Bit Map

This table provides a listing of the individual Stage™ control bits associated with each V-memory address.

MSB						DL350 Stage (S) Control Bits									LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41000
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41001
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41002
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41003
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41004
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41005
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41006
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41007
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V41010
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V41011
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V41012
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V41013
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V41014
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V41015
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V41016
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V41017
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V41020
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V41021
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V41022
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V41023
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V41024
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V41025
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V41026
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V41027
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V41030
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V41031
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V41032
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V41033
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V41034
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V41035
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V41036
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V41037

MSB							DL350 Additional Stage (S) Control Bits (continued)								LSB	Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	V41040
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	V41041
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	V41042
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	V41043
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	V41044
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	V41045
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	V41046
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	V41047
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	V41050
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	V41051
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	V41052
1277	1276	1275	1274	1273	1272	1271	1270	1267	1266	1265	1264	1263	1262	1261	1260	V41053
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	V41054
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	V41055
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	V41056
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	V41057
1417	1416	1415	1414	1413	1412	1411	1410	1407	1406	1405	1404	1403	1402	1401	1400	V41060
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	V41061
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	V41062
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	V41063
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	V41064
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	V41065
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	V41066
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	V41067
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	V41070
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	V41071
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	V41072
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	V41073
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	V41074
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	V41075
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	V41076
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	V41077

DL350 Timer and Counter Status Bit Maps

This table provides a listing of the individual timer and counter contacts associated with each V-memory address bit.

MSB						DL350 Timer (T) and Counter (CT) Contacts									LSB	Timer Address	Counter Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41100	V41140
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41101	V41141
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41102	V41142
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41103	V41143
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41104	V41144
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41105	V41145
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41106	V41146
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41107	V41147

This portion of the table shows additional Timer contacts available with the DL350.

MSB						DL350 Additional Timer (T) Contacts									LSB	Timer Address
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V41110
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V41111
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V41112
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V41113
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V41114
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V41115
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V41116
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V41117