

## Programmable Analog Input

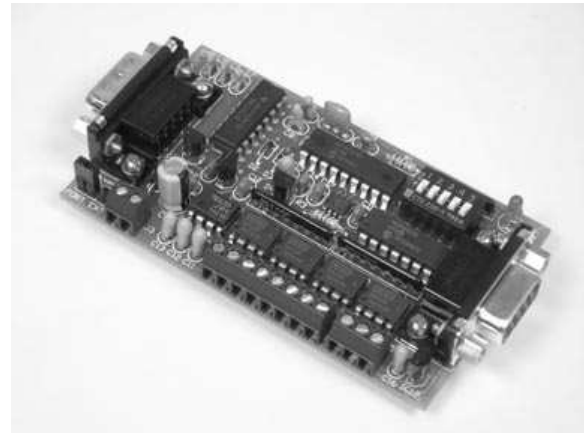
### FEATURES

- 4 differential input channels with a  $\pm 10V$  common mode range and protection to  $\pm 40V$ .
- 3 factory calibrated modes: +10.000 V to -8.000 V,  $\pm 600.0$  mV, and  $\pm 600.00$  mV.
- 2 user programmable modes which display readings in direct engineering units of choice.
- User programmable decimal point position.
- 20-bit A/D converter provides resolutions down to  $1.4 \mu V$  (user programmable modes).
- 8-times averaging smoothes out readings.
- 5-Volt excitation output for bridge sensors.
- Software calibrated; no trim-pots to adjust.
- All user configuration and calibration data stored in non-volatile memory.
- DIP switch addressable; stack up to 32 modules on the same serial port for 128 differential input channels.
- Industry standard RS-232 interface. Meets all EIA/TIA-232E and V.28 specifications.
- Wide power supply range (15 to 30 VDC).

### DESCRIPTION

Connects to the RS-232 serial port of a PC, laptop, or other host. Simple command strings sent from the host will read the voltage at the input channels. The value returned to the host can be shown as a voltage, or programmed by the user for any linear engineering units desired. The readings can also contain a decimal point if needed, the position programmable by the user.

Interface directly with pressure transducers, load cells, strain gages, temperature sensors, 4-20 mA devices, potentiometers, etc. Ideal for direct data acquisition applications without requiring data conversions at the output.



### SPECIFICATIONS

|                       |  |
|-----------------------|--|
| Analog Inputs         | 4 differential channels                  |
| ADC Resolution        | 20-bit                                   |
| Sample Rate           | 60 sps total                             |
| Accuracy (FCM)        | $\pm (0.05\% + 20\mu V)$                 |
| Input Range           | -8V to +10V                              |
| Over Voltage Range    | Each Input protected from -40V to +40V   |
| Common Mode Rej.      | 80 dB                                    |
| Input Impedance       | 10 M $\Omega$                            |
| Processor             | PIC16F628                                |
| Clock                 | 20 MHz                                   |
| Communications        | 9600 Baud, N, 8, 1                       |
| Power Requirements    | +15 to +30 VDC                           |
| Current Draw          | 35 to 45 mA, plus any excitation current |
| Operating Temperature | -20°C to +80°C                           |
| Board Dimensions      | 3.5" x 2.0" x 0.7"                       |
| Weight                | 1.9 oz                                   |

**TABLE 1: ADDRESS SETTING**

| HEADER CHARACTER<br>ASCII (HEX) | DIP SWITCH SETTING |
|---------------------------------|--------------------|
|                                 | 1=on, 0=off        |
|                                 | 1 2 3 4 5          |
| <b>A</b> (41)                   | <b>0 0 0 0 0</b>   |
| <b>B</b> (42)                   | <b>0 0 0 0 1</b>   |
| <b>C</b> (43)                   | <b>0 0 0 1 0</b>   |
| <b>D</b> (44)                   | <b>0 0 0 1 1</b>   |
| <b>E</b> (45)                   | <b>0 0 1 0 0</b>   |
| <b>F</b> (46)                   | <b>0 0 1 0 1</b>   |
| <b>G</b> (47)                   | <b>0 0 1 1 0</b>   |
| <b>H</b> (48)                   | <b>0 0 1 1 1</b>   |
| <b>I</b> (49)                   | <b>0 1 0 0 0</b>   |
| <b>J</b> (4A)                   | <b>0 1 0 0 1</b>   |
| <b>K</b> (4B)                   | <b>0 1 0 1 0</b>   |
| <b>L</b> (4C)                   | <b>0 1 0 1 1</b>   |
| <b>M</b> (4D)                   | <b>0 1 1 0 0</b>   |
| <b>N</b> (4E)                   | <b>0 1 1 0 1</b>   |
| <b>O</b> (4F)                   | <b>0 1 1 1 0</b>   |
| <b>P</b> (50)                   | <b>0 1 1 1 1</b>   |
| <b>a</b> (61)                   | <b>1 0 0 0 0</b>   |
| <b>b</b> (62)                   | <b>1 0 0 0 1</b>   |
| <b>c</b> (63)                   | <b>1 0 0 1 0</b>   |
| <b>d</b> (64)                   | <b>1 0 0 1 1</b>   |
| <b>e</b> (65)                   | <b>1 0 1 0 0</b>   |
| <b>f</b> (66)                   | <b>1 0 1 0 1</b>   |
| <b>g</b> (67)                   | <b>1 0 1 1 0</b>   |
| <b>h</b> (68)                   | <b>1 0 1 1 1</b>   |
| <b>i</b> (69)                   | <b>1 1 0 0 0</b>   |
| <b>j</b> (6A)                   | <b>1 1 0 0 1</b>   |
| <b>k</b> (6B)                   | <b>1 1 0 1 0</b>   |
| <b>l</b> (6C)                   | <b>1 1 0 1 1</b>   |
| <b>m</b> (6D)                   | <b>1 1 1 0 0</b>   |
| <b>n</b> (6E)                   | <b>1 1 1 0 1</b>   |
| <b>o</b> (6F)                   | <b>1 1 1 1 0</b>   |
| <b>p</b> (70)                   | <b>1 1 1 1 1</b>   |

**STACKABLE DATA MODULES**

All modules in this series incorporate two EIA/TIA-232E serial ports which communicate at 9600 baud, no parity, 8 data bits and 1 stop bit. DB9 connectors are jumpered to satisfy hardware handshaking. The port labeled "HOST" is configured as a DCE device and should be connected to a PC's serial port. The port labeled "SLAVE" is a DTE device and can be left open, or connected to another module's host port. Up to 32 modules can be chained together in this fashion to form a network. Either plugged together end to end, or separated by a cable. Because a module contains two individual bi-directional ports which pass data through, it also acts as a repeater, extending the total allowable length of the RS-232 communications line.

A modem can serve as the host for remote operation, but since a modem uses a DCE port, a "null modem" adapter must be placed between the modem and the data module's host port. A gender changer may also be required. In addition, any hardware/software flow control must be disabled in the terminal program.

Each module in a network should be set to a different address using the on-board 32-position DIP switch. A module will only respond to data packets that begin with its' own unique header character, which is determined by this DIP switch setting. See Table 1. Data packets transmitted by a module will also begin with this header character. The host PC can use the header character to address each individual module in a network, and to identify a module which is talking.

**COLLISION CONTENTION**

The utilization of the communications line can be thought of more as a single, bi-directional, data bus, operated in a multi-drop mode rather than a standard RS-232 data link. A transmission from a data module travels in both directions, upstream to the host, and downstream to signal other modules that it has seized the line. Before transmitting, a module will listen to the communications line and wait for quiescence. After a silent period equal to the length of one byte, the waiting module will send its data packet using a Carrier Sense Multiple Access with Collision Detection communications protocol. See the application note (AN100) at the back of this manual for more details.

## COMMAND SET

The host PC communicates with the Analog Input Module using a command set comprised of standard ASCII character strings as depicted in Table 2. Each command must be preceded with the header character which is determined by the DIP switch setting on the circuit card (see Table 1), and terminated with a carriage return.

**READ** - Reads the current voltage applied to a specified input channel and converts it to the format which was previously set by the MODE command. The results will be returned to the host and will contain a decimal point if it was previously set by the DECIMAL command. If the voltage applied to the input channel is beyond the range of the MODE setting plus 5%, the error symbol will be returned to the host.

**MODE** - Sets the conversion mode and voltage range for a specific input channel. This mode will be used each time the READ command is issued for this channel. There are three factory calibrated modes which will return the results listed in voltage, and two user programmable custom modes which will return the results in direct linear engineering units of choice. If using a custom mode, the user must set the offset null by using the ZERO command, and calibrate the channel as discussed in the section titled "USER PROGRAMMABLE MODES" later in this text.

**DECIMAL** - Sets the decimal point position for a specific input channel. Each time the READ command is issued for this channel, the value returned to the host will contain a decimal point at the position specified. If this setting equals 0, no decimal point will be used. Note, using a decimal point will not increase the number of

**TABLE 2: COMMAND SET**

| TITLE   | COMMAND            | DESCRIPTION  |
|---|--------------------|--|
| READ  | <i>R chn</i>       | Read the current value of an input channel. <i>chn</i> = A-D. Returns data in format set by MODE and decimal point position set by DECIMAL.  |
| MODE  | <i>M chn value</i> | Sets the mode and voltage range for a specific input channel which will be used each time the READ command is issued. <i>chn</i> = A-D, <i>value</i> = 1-5. 1 = +10.000V to -8.000V, 2 = ±600.0mV, 3 = ±600.00mV, 4 = custom mode using +10 volt to -8 volt input range, 5 = custom mode using ±0.6 volt input range. Default = 1. (Note 3, 4) |
| DECIMAL   | <i>D chn value</i> | Sets the decimal point position for a specific input channel which will be used each time the READ command is issued. <i>chn</i> = A-D, <i>value</i> = 0-7. 0 = none, 1 = 0.0, 2 = 0.00, etc. Default = 0. (Note 3, 4)   |
| ZERO  | <i>Z chn</i>       | If MODE is set to custom for a specific input channel, this command will automatically adjust the offset null so that the current voltage being applied to its input will be displayed as zero. <i>chn</i> = A-D. (Note 3)   |
| SPAN  | <i>S chn value</i> | If MODE is set to custom for a specific input channel, this command will calibrate the channel so that the current voltage being applied to its input will be displayed as <i>value</i> . <i>chn</i> = A-D, <i>value</i> = 0 to ±8388607. Use ZERO to set the offset null before using this command. (Note 3)                                  |
| FACTOR  | <i>F chn value</i> | If MODE is set to custom for a specific input channel, this command will calibrate the channel so that <i>value</i> will be displayed as one unit. <i>chn</i> = A-D, <i>value</i> = 0 to ±8388607 listed in mV including decimal point. Use ZERO to set the offset null before using this command. (Note 3)                                    |
| ERROR   | ?                  | This character will be returned after an invalid command or variable.  |
| RESET   | !                  | This character will be returned after a power-on reset, or brownout.   |
| <p><b>Note 1:</b> All command strings sent to the data module should be preceded with the header character (see Table 1), and terminated with a carriage return. All responses from the data module will also appear in this format.</p> <p><b>Note 2:</b> Any spaces shown above in the listing of the command strings are for clarity only. They should not be included in the actual transmission from the host, nor expected in a response from the data module.</p> <p><b>Note 3:</b> After successful execution, this command will be echoed back to the host in the same format as received.</p> <p><b>Note 4:</b> If <i>value</i> is omitted, reads the current setting which will be returned to the host in the same format as above.</p> |                    |  |

**TABLE 3: TERMINAL / CONNECTOR DESCRIPTION**

| NAME   | TYPE            | ELECTRICAL SPECS                                | COMMENTS:   |
|--|-----------------|---|---|
| HOST   | DB9<br>(female) | EIA/TIA-232E Standard                           | RS-232 serial port configured as DCE. Connects to host PC. Hardware handshake jumpered.                   |
| SLAVE  | DB9<br>(male)   | EIA/TIA-232E Standard                           | RS-232 serial port configured as DTE. Can be connected to another data module's HOST port for networking. |
| Power Source   | Jumper          | N/A   | Power source selection jumper. Selects either external, or port powered. (Note 1)                         |
| +  | Screw Term      | +15 to +30 VDC                                  | External unregulated power supply input.  |
| -  | Screw Term      | GND   | External power supply ground.   |
| A – D  | Screw Term      | Range = -8V to +10V<br>Max input = -40V to +40V | Analog input channels. (Note 2)   |
| V+   | Screw Term      | Tied to Power Supply Input                      | Provides power to 4-20mA devices.   |
| E+   | Screw Term      | +5 Volt Regulated Output                        | Provides excitation to bridge sensors.  |
| GND  | Screw Term      | Tied to Power Supply Ground                     | Used as ground return for V+ and E+.  |
| <p><b>Note 1:</b> Selecting "port powered" will draw from the power supply source of an upstream data module. Caution, the USB or COM port of a PC or laptop does not supply enough current to serve as the power supply source.</p> <p><b>Note 2:</b> If <math>V_{IN+} &gt; V_{IN-}</math> result will be shown as a positive number. If <math>V_{IN+} &lt; V_{IN-}</math> result will be shown as a negative number.</p> |                 |   |   |

digits shown on the screen, but will simply insert a decimal point within the original number.

**ZERO** - If an input channel is set to one of the custom modes, this command is used to calibrate the offset null so that the desired input level will be displayed as zero.

**SPAN** - If an input channel is set to one of the custom modes, this command is used to generate the calibration coefficient needed to convert a given input voltage to any display value of choice. For more info, see the section titled "USER PROGRAMMABLE MODES".

**FACTOR** - If an input channel is set to one of the custom modes, this command is used to generate the calibration coefficient needed to convert one display unit to any input voltage of choice. The voltage is listed in mV and may contain a decimal point. For more info, see the section titled "USER PROGRAMMABLE MODES" later in this text.

**ERROR** - Any data string sent from the host containing the correct header character but an invalid command or variable will be responded to with this error indicator.

**RESET** - Upon power-up or any other reset condition, this indicator is transmitted to the host. Note, all user configuration and calibration data is stored in non-volatile memory. Therefore, a reset or loss of power will not corrupt these settings.

## OPERATION

To hook the data module to a host PC, use a standard RS-232 cable with male and female DB9 connectors on opposite ends. This cable should be wired straight through (pin to pin) with no crossover of the data lines. In other words, not a null modem cable. Connect a suitable DC power source to the + and - terminals of the data module. It is highly recommended to use an ungrounded AC adapter such as that which is available from Weeder Technologies. This will provide isolation and prevent ground loops which are commonly created if the power supply and computer are grounded at different points.

When the data module is first powered up, the red LED will flash briefly. This indicates that the on-board microcontroller has booted up, successfully completed its internal diagnostic

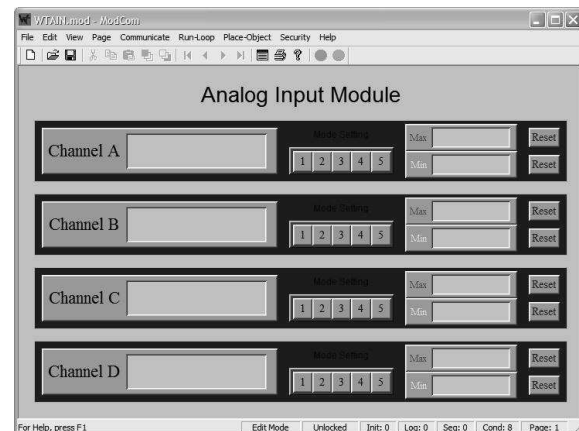
test, and has transmitted the reset character to the host to signal that it is up and running. The red LED will also flash anytime the module receives or transmits any data packet, thus making it a valuable diagnostic tool when troubleshooting communications problems.

An easy-to-use Windows software package called "ModCom" is available and can be downloaded from [www.weedtech.com](http://www.weedtech.com). This program will allow the user to quickly set up buttons which transmit commands, dials and meters which automatically poll for data, and a variety of other screen objects such as slider controls, event counters and timers, chart recorders that plot the data from any screen object, and more. In addition, conditional statements can be set up to watch for and act upon specific conditions, program scripts can be written by the user to perform complex tasks in the background, and data can be logged to a file or web page at user-defined intervals.

Once ModCom is installed and running, go to the <Communicate> menu item at the top of the screen and click on <Send/Receive>. A dialog box will pop up which you can use to type in the commands from Table 2, transmit them directly to the data module, and see the response coming back. Use this dialog box to familiarize yourself with the command set and to experiment with the various features supported by the module. The experience gained here is significant since these are the same command strings you will use when setting up the other objects in ModCom.

To read voltages at the input channels, start with the sample application "WTAIN.mod" which can be found in the ModCom subfolder called "Samples". After this file is opened, it will appear as shown in Figure 1. To start the main run-loop, click on the green toolbar button at the top of the screen. At this time the large Data Windows at the left will begin polling for data and showing the input voltages on the screen. The Button Arrays next to them can be used to change the MODE of each channel on the fly. The smaller Data Windows to the right can be used to track and hold the maximum and minimum values of the main Data Windows. They are initialized by a program Script called "SetupMaxMin" and use Conditional statements for updating. To examine these and to see how they operate, halt the main run-loop and then go to the <Run-Loop> menu item at the top of the screen and click on "Scripts" and "Conditionals"

**FIGURE 1: MODCOM APPLICATION**



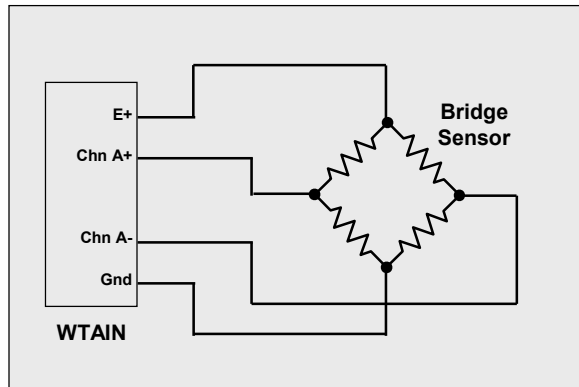
Each Data Window can be set up to signal an alarm condition by changing colors if the reading exceeds a predefined high or low limit. To do this, first halt the run-loop by clicking on the red toolbar button at the top of the screen, then right-click on the object to pull up its properties dialog box. Within this box are a number of options which can be modified by the user. For more details, read the help files which can be accessed by clicking on the <Help> menu item at the top of the screen.

The value shown in each Data Window can also be logged to a file. Go to the <Run-Loop> menu item at the top of the screen and click on <Data Logging>. Within this dialog box you can specify the data log file name, set the time interval between log events, and select those screen objects that you wish to log to file. The contents of the file can be viewed or printed by clicking on the corresponding button shown in this dialog box, or by clicking on a toolbar button at the top of the main screen.

## INPUT CHANNEL SCANNING

During normal operation, the Analog Input Module continuously scans all 4 channels in the background regardless of whether or not they are being polled by the host. The full 20-bit A/D results of each channel is stored in its own circular buffer which will always hold the most recent 8 samples. At any time a READ command is issued, the 8 samples associated with that particular channel are averaged together and the results will be the data which is returned to the host. This 8-times averaging smoothes out the readings and reduces the jitter shown on the screen.

**FIGURE 2: USING BRIDGE SENSORS**



**USER PROGRAMMABLE MODES**

The Analog Input Module incorporates a 32-bit floating point math routine which provides data conversions using calibration coefficients stored in non-volatile memory. Each of the modes selected with the MODE command has its own calibration coefficients. There are two modes which can be programmed by the user to display the results of each READ command in the linear engineering units preferred. The span of the A/D converter is extended across the range of each mode and determines the maximum resolution available. Mode 4 has a range of +10 to -8 volts and a resolution of 22 μV. Mode 5 has a range of +0.6 to -0.6 volts and a resolution of 1.4 μV. Each mode can be individually programmed using one of two methods, SPAN or FACTOR.

SPAN can be used if a known output reading can be generated by the sensor. Sensor manufacturers often include a shunt resistor for this purpose and when attached to the sensor's bridge, will cause it to output a voltage which represents a specific reading. Applying a known load or stimulus to a sensor also works well. After setting up the sensor so that it outputs the known reading, transmit the SPAN command with the *value* field containing the reading that should be displayed on the screen. Multiplying the value by multiples of 10 will increase the resolution. For example, if applying a 50-lbs load to a pressure transducer, transmitting a value of 5000 in the SPAN command will set up the channel to display its readings in 0.01-lbs increments. The DECIMAL command can then be used to set up the channel to insert a decimal point in the correct position of each reading.

FACTOR can be used if the actual voltage that equals one unit of measurement can be

determined. To calculate this voltage in mV, use the factory-listed output of the bridge sensor in the following equation.

$$\text{value} = \text{Out} * E / \text{FS} \quad \text{where:}$$

Out = Output of the sensor in mV/V.

E = Excitation voltage applied to the bridge.

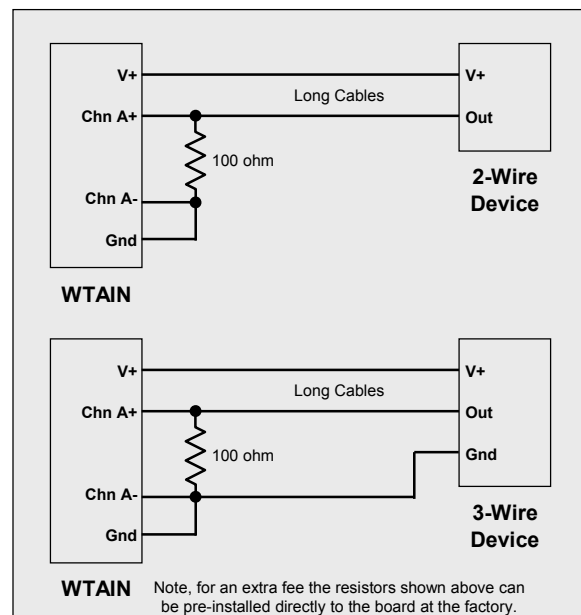
FS = Full scale capacity of the sensor.

For best results, the excitation voltage should be measured directly at the bridge using the same analog input channel and lead wires that will be attached to the bridge output itself. Transmit the FACTOR command with the *value* field containing the results of this equation. Dividing the value by multiples of 10 will increase the resolution displayed on the screen in the same way as multiplying the value used in the SPAN command, as discussed earlier.

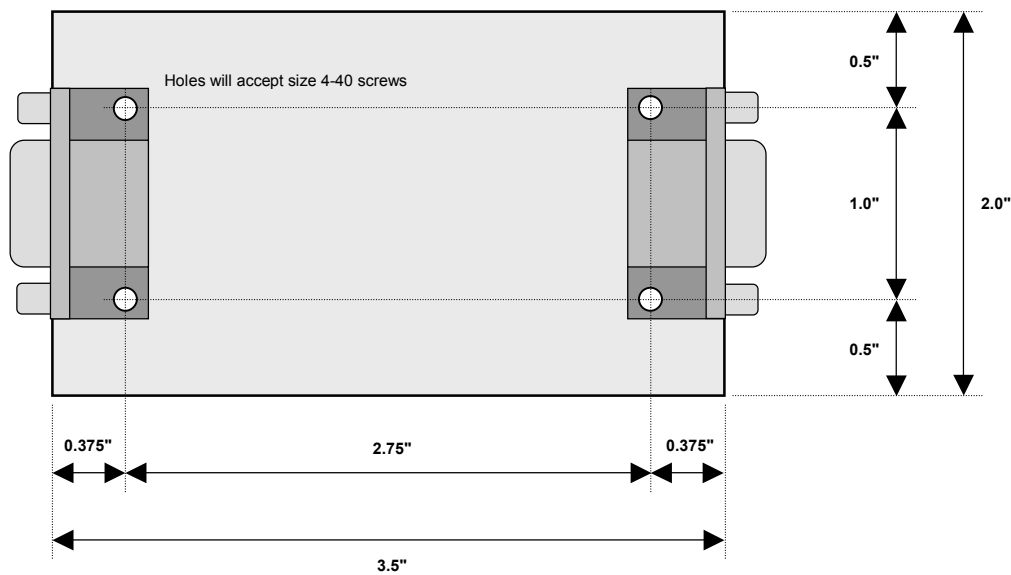
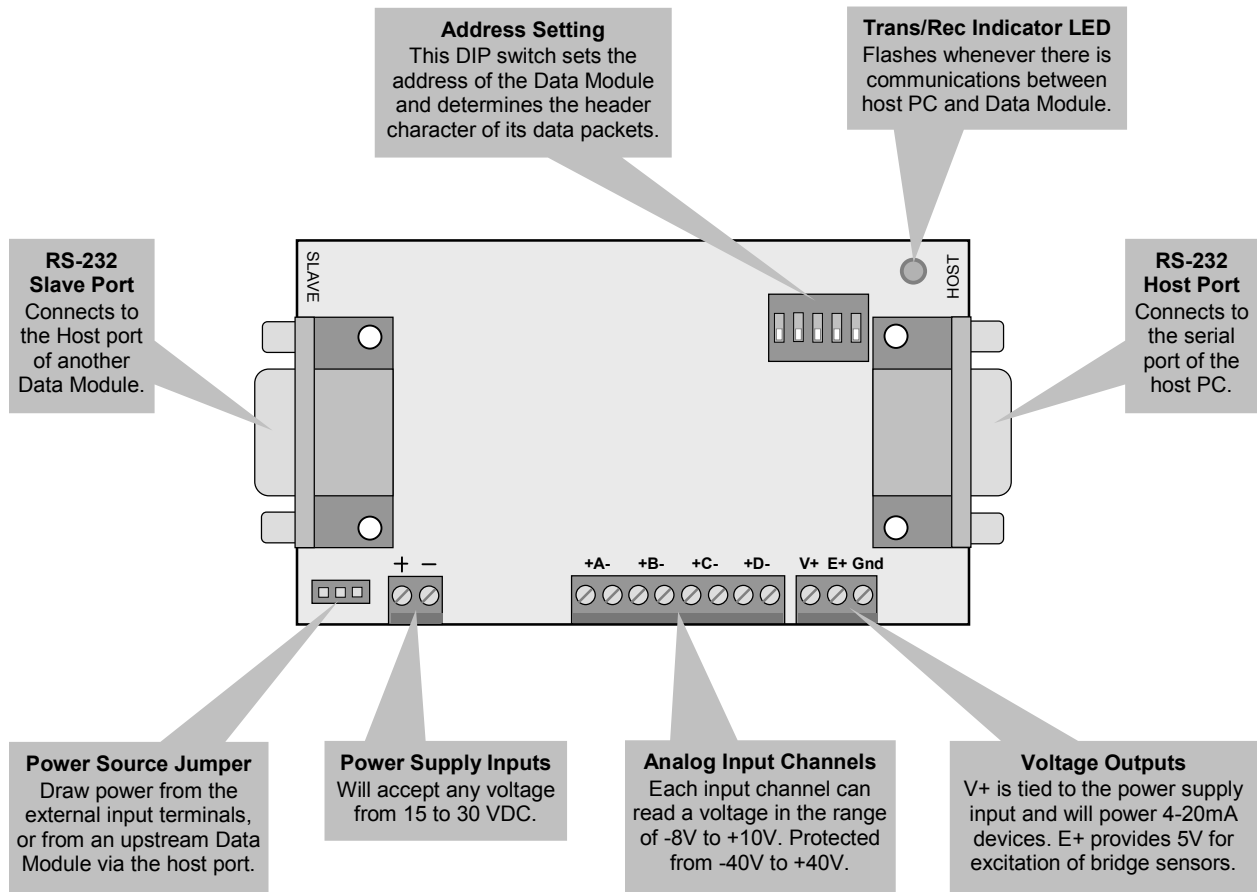
**4-20 mA CURRENT TRANSMITTERS**

To read the data from a 4-20 mA current transmitter, place a 100-ohm resistor in the current loop and use an input channel to read the voltage across the resistor as shown in Figure 3. The results returned to the host will be from 400mV to 2000mV representing the 4mA to 20mA current range. A user programmable mode can be used to automatically convert this voltage reading to any linear engineering units of choice and include a decimal point if needed.

**FIGURE 3: USING 4-20 mA DEVICES**



Note, for an extra fee the resistors shown above can be pre-installed directly to the board at the factory.



### PROGRAMMABLE ANALOG INPUT WTAIN-M

