# 4-Port USB Relay Controller with 6-Channel Temperature/Analog/Digital I/O Interface (Data Acquisition Module)

## INTRODUCTION

This USB Relay Controller / Data Acquisition Module allows computer controlled switching of external devices as well as full bi-directional communication with the external world (ideal for Data Acquisition applications) using the USB port of your computer. The controller is very flexible and can be used in many custom applications including weather stations as well as temperature monitoring, logging and control as it can be easily connected to temperature and other types of sensors.

It is compatible with both Windows and Apple OS X, as well as various Linux flavors and appears as a USB CDC (Communications Device Class) device which creates a Virtual Serial (COM) port allowing easy communication with the board through any programming language that supports serial communications (VB, VB.NET, C#, C, C++, Perl, Java, etc). A complete set of easy to use commands are available for complete control of all relays, I/O channels and sensors.

The controller provides four relay outputs with a current rating of 5A each. In addition, it provides a 6-channel Input/Output interface, with each channel individually configurable as Digital Input, Digital Output, Analog Input (10-bit Resolution), or Temperature Sensor (Dallas/Maxim 1-Wire Series).

In Digital Input/Output modes, each channel can support a TTL compatible or ST (Schmitt Trigger) input (depending on the channel) or a 5V output signal.

In Analog Input mode, each channel can convert a voltage of between 0 to 5V (or higher voltage using a voltage-divider) into a 10-bit digital representation.

Finally, in Temperature Sensor mode, each channel can be connected to a Maxim/Dallas 1-Wire Digital Temperature Sensor device (sold separately) such as the DS1822, DS18S20, DS18B20 or DS1821 and provides temperature readings in both Degree Celsius or Degree Fahrenheit.

Windows Drivers as well as sample code and test applications can be found at:

**http://www.canakit.com/uk1104**

# TECHNICAL SPECIFICATIONS

Supported Operating Systems     : Microsoft Windows 2000 / XP / 2003 / 7
     Any Operating System supporting USB CDC
     Virtual COM Ports (e.g. Linux, Mac OS X, etc.)

Number of Relays     : 4

Relay Rating     : 5A 110V AC / 24V DC (Resistive Loads)

Number of I/O Channels     : 6

I/O Channel Modes     : Digital Output
     Digital Input (TTL/ST)*
     Analog Input
     Temperature Sensor

Supported Temp. Sensors     : Maxim/Dallas DS18 Series

Max. Current per I/O Channel     : 25 mA

Max. Total Current for I/O Channels  : 75 mA (45 mA if all Relays are ON)

Max. # of Temperature Sensors     : 6

A/D Conversion Resolution     : 10-bit (5V DC Scale)

Max. Recommended Impedance
   for Analog Sources     : 10K Ohm

Supply Voltage
   Controller Circuit     : USB powered**
   Relay Circuit     : 9V DC ***

* CH1, CH2, CH3 and CH6 are ST (Schmitt Trigger) inputs, CH4 and CH5 are TTL compatible inputs
** The controller can also be configured to be powered by the external supply voltage used for the relay circuitry.
*** If the relays will not be used, the 9V DC supply voltage is not required.

## IMPORTANT DISCLAIMER

This device connects to the USB port of your computer and can be used to control external devices connected to its onboard relays.  Incorrect wiring or shorts on the board can potentially cause damage to the controller itself, your computer's USB controller and/or your computer's motherboard if an external voltage makes its way to the USB bus or the USB port is shorted.  Extreme care must be taken when using this device to avoid any damage to your equipment.  **In particular, make sure you always disconnect the device from the USB port as well as any other power source when working on the device.**

CANA KIT CORPORATION, it's shareholders, employees, suppliers, distributors and/or resellers are not liable for any damage or loss of data as a result of the use of this device, including special, incidental, or consequential damages resulting from the use of this device, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs or recovering or reproducing any data stored in computers connected to this device.

**Your use of this circuit indicates your acceptance of these terms.**

# COMMANDS REFERENCE

## Individual Relay Commands

| | | |
|---|---|---|
| RELx.ON | Turn Relay (x) ON | ::REL1.ON |
| RELx.OFF | Turn Relay (x) OFF | ::REL2.OFF |
| RELx.TOGGLE | Toggle Relay (x) | ::REL3.TOGGLE |
| RELx.GET | Get Relay (x) Current State | ::REL4.GET<br>*1* |

## Bulk Relay Commands

| | | |
|---|---|---|
| RELS.ON | Turn All Relays ON | ::RELS.ON |
| RELS.OFF | Turn All Relays OFF | ::RELS.OFF |
| RELS.GET | Get All Relay Current Statuses | ::RELS.GET<br>*0 1 1 0* |

## Individual Channel Commands

| | | |
|---|---|---|
| CHx.ON | Set CH (x) ON | ::CH1.ON |
| CHx.OFF | Set CH (x) OFF | ::CH3.OFF |
| CHx.TOGGLE | Toggle CH (x) | ::CH6.TOGGLE |
| CHx.GET | Get CH (x) Current State | ::CH6.GET<br>*1* |
| CHx.SETMODE | Sets the Channel Type for CH (x) | ::CH1.SETMODE(1) |
| | Valid Types are:      1 - Digital Output<br>2 - Digital Input<br>3 - Analog Input*<br>4 - Temperature Sensor | |
| CHx.GETANALOG | Get current analog value | ::CH1.GETANALOG<br>*511* |
| CHx.GETTEMP | Get current temperature value | ::CH1.GETTEMP<br>*27.0000* |

## Bulk Channel Commands

| | | |
|---|---|---|
| CHS.ON | Set All Channels ON | ::CHS.ON |
| CHS.OFF | Set All Channels OFF | ::CHS.OFF |
| CHS.GET | Get All Channels Current Statuses | ::CHS.GET<br>*0 1 1 0 1 1* |

# COMMANDS REFERENCE (CONTINUED)

**Misc Commands**

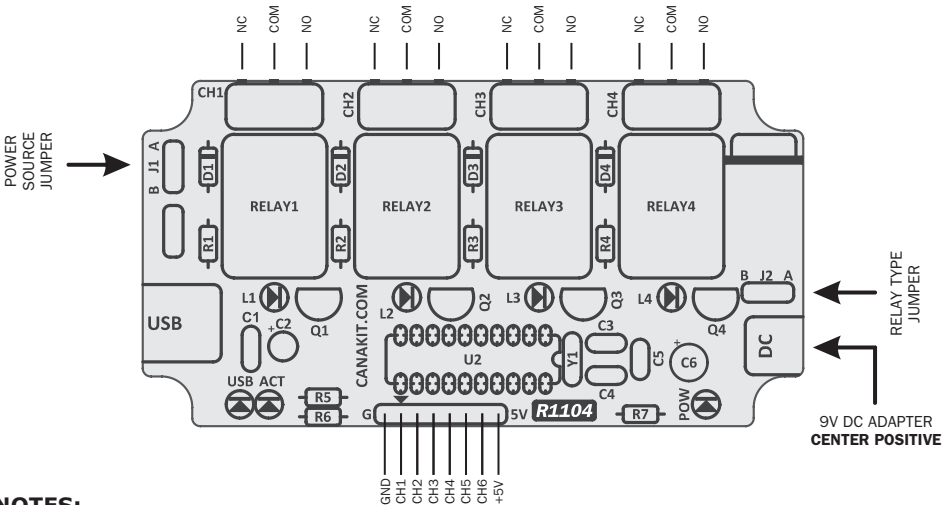| CHS.SETTEMPRES | Set Temperature Sensor Resolution | ::CHS.SETTEMPRES(12) |
|---|---|---|
| | Valid Resolutions are 9, 10, 11 or 12-bit | |
| SETID | Assigns a 2 character Unit ID to the unit | ::SETID(A0) |
| ABOUT | Outputs current version and Unit ID | |

::ABOUT
*UK1104 | V2.1 | WWW.CANAKIT.COM*
*ID: A0*

# CONTROLLER OVERVIEW



**NOTES:**

- J1:    A – Use USB bus voltage to power the controller circuit
         B – Use external power source to power the controller circuit

- J2:    A – 9V DC Relays Installed
         B – 5V DC Relays Installed (Optional - Relays will need to be replaced)

- GND and +5V pins of the header can be used as a 5V DC regulated supply voltage for any external circuitry.  When the controller circuit is being powered from the USB bus (J1 in the "A" position), you should not draw any more than about 400 mA from the GND and +5V pins as exceeding this may exceed USB standard limits.

- A 9V DC adapter (center-positive) is required to power the relay circuitry.  If the relays are not being used and the controller circuit is being powered from the USB bus (J1 in the "A" position), then this 9V DC supply is not required.

# IMPORTANT NOTES AND TIPS

**1.** You must always wait for the prompt "::" before sending any commands to the unit.

**2.** All commands must be followed by a Carriage Return (CR) and Line Feed (LF) character (ASCII character 13 followed by a ASCII character 10).

**3.** When you first connect to the controller, simply send an initial CR and LF to initialize the controller and get to the initial "::" prompt.

**4.** Note that there is an exception when setting a channel to Analog mode. The controller will automatically set all channels up to and including the channel set to Analog mode to the Analog mode. That is, if you issue the command CH3.SETMODE(3), then channels 1 through 3 are ALL set to Analog mode. This behavior is by design. When allocating the channels for different modes, you should allocate all analog channels first and then the remaining channels may be used in any other mode.

**5.** Note that the CHx.ON, CHx.OFF, CHS.ON and CHS.OFF commands set the mode of the corresponding channel(s) to Digital Output even if the mode was something else. Do not use these commands on channels whose mode has been changed to something other than Digital Output.

**6.** The controller does not support changing of a channel mode from the Analog mode back to another mode. That is, if you issue the command CH1.SETMODE(3), you cannot subsequently issue a command such as CH1.SETMODE(1). Note that this is only the case for the Analog channel model. You can safely change the mode of a channel multiple times when using the other channel modes.

**7.** You can use the SETID command to assign a unique Unit ID to a controller. The ID is stored on the controller's EEPROM memory. This allows you to be able to attach multiple controllers to a single computer and be able to distinguish which unit you are communicating with at any given time.

**8.** The default temperature sensor resolution is set to 9-bit. With this resolution, the conversion time is approximately 94ms and reading resolution is in 0.5°C increments. Note that the higher the reading resolution, the higher the conversion time. The table below shows the different resolutions and corresponding conversion times and reading increments. Note that the reading resolution is different than the accuracy of the reading. For example, the DS18B20 has a **worst case** accuracy of ±0.5°C (in the -10°C to +85°C range).

| | | |
|---|---|---|
| 9-bit | ~94mS | Readings in 0.5000 °C increments |
| 10-bit | ~188mS | Readings in 0.2500 °C increments |
| 11-bit | ~375mS | Readings in 0.1250 °C increments |
| 12-bit | ~750mS | Readings in 0.0625 °C increments |

# WINDOWS DRIVER INSTALLATION

1.  Download and extract the driver files into a directory on your hard drive. The required Windows driver files can be found at: **http://www.canakit.com/uk1104 .**

2.  Using a standard USB cable, connect the controller to your computer. The following window should now appear if the device driver has not been previously installed.
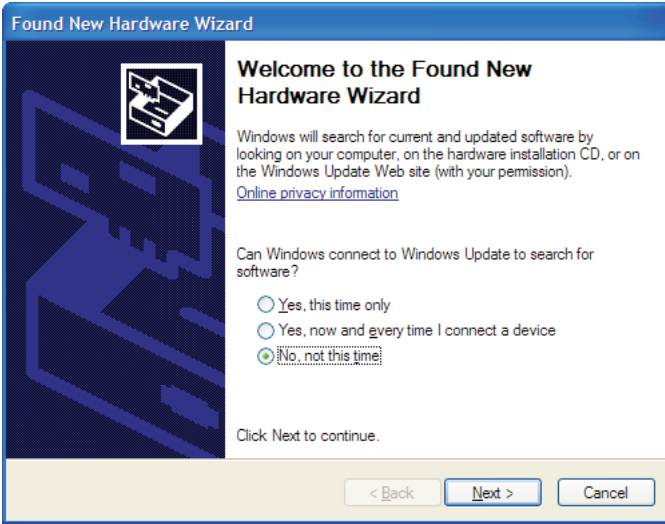


3.  Select "No, not this time" and click "Next". The following window should now be displayed allowing you to specify where the drivers for the device can be found.
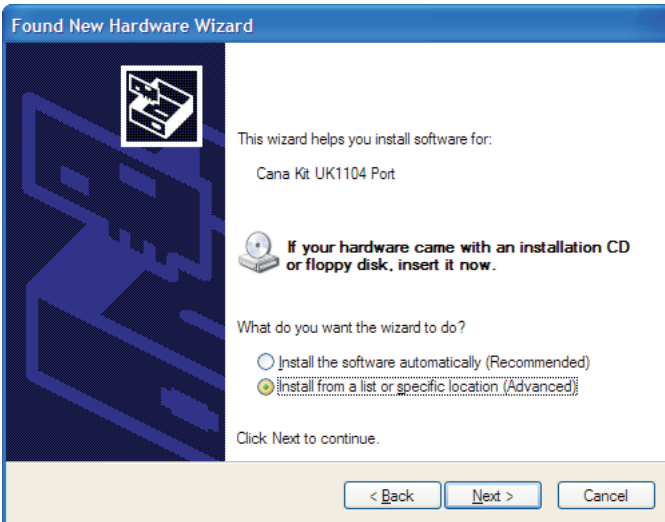
**4.** Select "Install from a list or specific location (Advanced)" and click "Next". The following window should now be displayed.
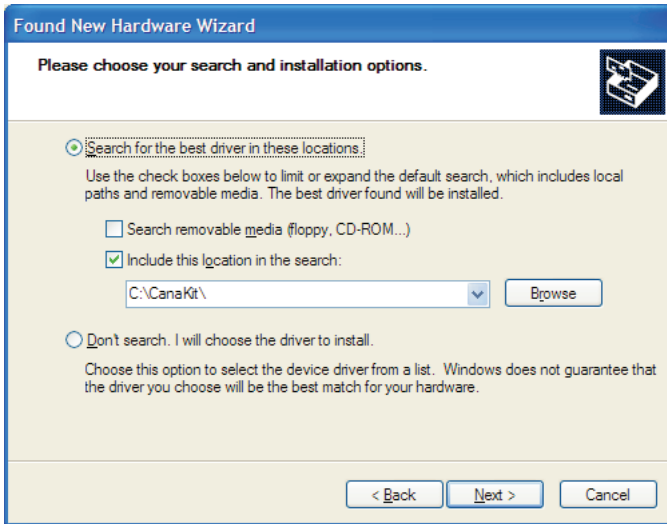


**5.** Click "Browse" and then select the location where you saved the UK1104 driver files and then click "Next". You may receive a message indicating that the driver has not passed Windows Logo testing. If so, click "Continue Anyway" to continue with the installation.

**6.** The necessary driver will now be installed on your system and the controller will appear as a Virtual Serial (COM) port on the computer. To verify that the device was successfully installed and identify the assigned port, go to your Device Manager and expand the "Ports" node. In this example, the assigned port is COM14.
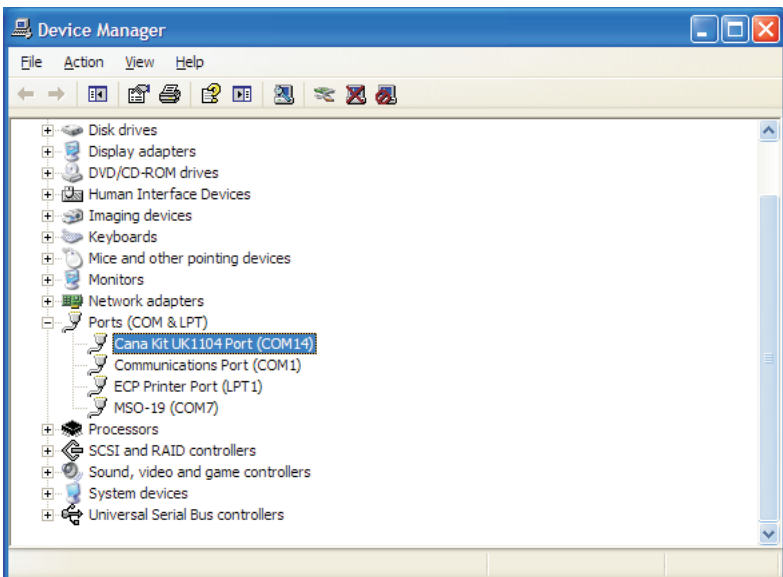
# COMMUNICATING WITH THE CONTROLLER

To communicate with the controller in an interactive way you need a Terminal Emulation program to connect to the assigned serial port. Windows comes with a basic such program called Hyperterminal (not installed by default). Other available and recommended Terminal Emulation programs are PuTTY, RealTerm and AbsoluteTerminal. Download links for these can be found at:

**http://www.canakit.com/uk1104**

You may connect to the device using your favorite Terminal Emulation program on the assigned serial port. The actual serial port speed and parameters are not important as the port is a virtual port. If required, set the Baud Rate to 115,200, Data Bits to 8, Parity to N and Stop Bits to 1. Below is a sample output of an interactive session.

```
UK1104 | V2.1 | WWW.CANAKIT.COM
ID: 00

::REL3.ON

::REL3.GET
1

::REL2.TOGGLE

::RELS.GET
0 1 1 0

::RELS.OFF

::CH2.ON

::CH2.GET
1

::CHS.GET
0 1 0 0 0 0

::CH1.SETMODE(3)

::CH1.GETANALOG
608

::CH6.SETMODE(4)

::CH6.GETTEMP
23.5000

::CH6.SETTEMPRES(12)

::CH6.GETTEMP
23.6250
```

```
::CH3.SETMODE(2)

::CH3.GET
1

::CH3.GET
0

::CHS.OFF

::CHS.GET
0 0 0 0 0 0

::ABOUT
UK1104 | V2.1 | WWW.CANAKIT.COM
ID: 00

::SETID(C1)

::ABOUT
UK1104 | V2.1 | WWW.CANAKIT.COM
ID: C1

::
```
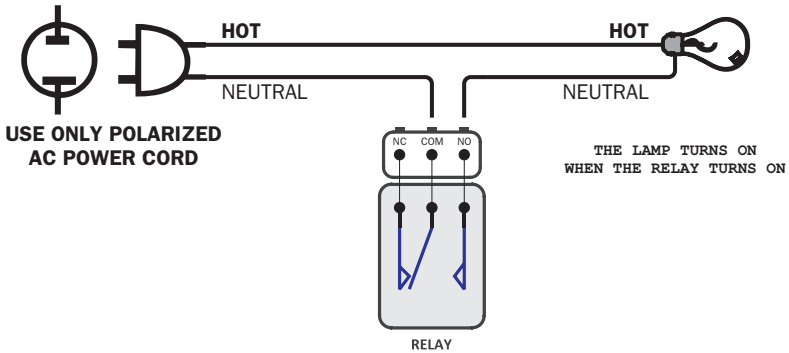
# Using Relay Outputs

The controller offers four SPDT relays incorporated right on the board. Each relay can handle up to 5A @ 110V AC or 24V DC. The three standard relay terminals, namely NO (Normally Open), NC (Normally Closed) and COM (Common) have been made available using four blocks of 3-pin screw terminals.

A relay is simply an electrically operated switch that uses an electromagnet to operate a switching mechanism. Each relay has a Normally Open terminal as well as a Normally Closed terminal. When the relay is Off, the NC and COM terminals are connected together. When the relay turns On, the NC terminal is disconnected and the NO and COM terminals are connected together. Refer to the diagram below for an example connection of an AC lamp.

**When connecting the relay to an AC main power source, it is highly recommended to bring in the neutral wire into the relay for safety reasons.**

HOT                    HOT

NEUTRAL                NEUTRAL

**USE ONLY POLARIZED
AC POWER CORD**

NC   COM   NO

THE LAMP TURNS ON
WHEN THE RELAY TURNS ON

RELAY

*Please note that this controller is designed to be connected to resistive loads as opposed to inductive loads. Connection of inductive loads to the controller, although technically possible, may cause unpredictable results due to introduction of undesired EMI (Electro Magnetic Interference) into the USB bus which can interfere to the USB operation and can cause the USB controller to disconnect the device.*

### IMPORTANT NOTE

**This circuit involves high voltage AC main power wiring when used to control high voltage AC devices. Please note that handling, testing and operating high voltage AC powered equipment can be dangerous and fatal when basic safety rules are not followed. ALWAYS disconnect the circuit from the AC line prior to performing any work on the circuit.**

**If you are inexperienced or not confident in working with high voltage AC powered circuits, we strongly recommend that you DO NOT attempt to use this circuit and seek the help of a licensed electrician.**
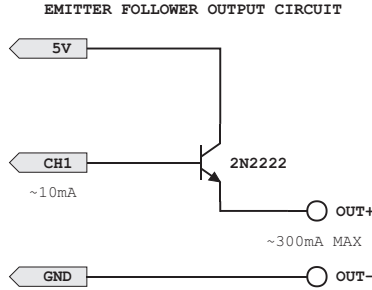
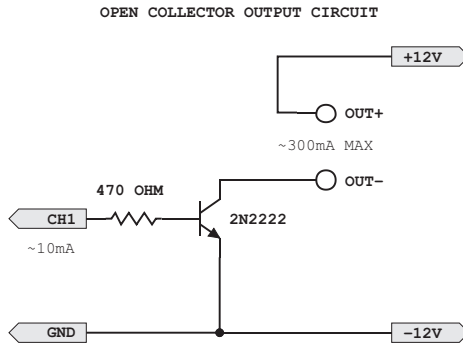**PLEASE REFER TO THE RELAY USAGE & HIGH-VOLTAGE SAFETY GUIDE INCLUDED**

# USING DIGITAL INPUTS & OUTPUTS

The controller offers up to 6 channels that can be used as Digital Outputs as well as Inputs. When used as an output, each channel can sink or source up to 25mA. However, in total, you should not sink nor source more than about 75 mA from all channels combined together (45 mA if all relays are ON). You can handle higher currents by using a simple Emitter Follower circuit as shown below. In this example, you will only draw less than 10 mA from the channel, but can switch a load of up to about 300 mA.

EMITTER FOLLOWER OUTPUT CIRCUIT

```
5V ──────────┐
             │
             ├─ 2N2222
CH1 ─────────┤
~10mA        └──── ○ OUT+
                   ~300mA MAX

GND ──────────────── ○ OUT−
```

You can also switch on or off a voltage other than 5V, if required for your application. As an example, the following simple circuit can be used if you wish to switch on or off a 12V DC load. *Note that the ground of both supplies will need to connected together.*

OPEN COLLECTOR OUTPUT CIRCUIT

```
                        ┌──── +12V
                        │
                   ○ OUT+
                   ~300mA MAX
                   ○ OUT−
         470 OHM        │
CH1 ─────WWW──── 2N2222
~10mA

GND ──────────────── −12V
```

Each channel can also be configured as a Digital Input. Channels 1, 2, 3 and 6 are Schmitt Trigger inputs, while Channels 4 and 5 are TTL compatible inputs. The Schmitt Trigger inputs have different transition voltages, depending on whether the input signal is changing from high to low or low to high. The table below summarizes the voltage thresholds for the different input types:

| | | |
|---|---|---|
| TTL | Low (0): 0.8V (Max) | High (1): 2V (Min) * |
| ST | Low (0): 1V (Max) | High (1): 4V (Min) ** |

* For TTL inputs, voltages between 0.8V and 2.0V would produce an "undefined" value.
** For ST inputs, the input will retail its current value until the threshold is crossed. For example, if the current input voltage is 5V, the input state will only be changed if the voltage drops below 1V. Conversely, if the current input voltage is 0V, the input value will only be changed if the voltage goes above 4V. On the other hand, if the input was a TTL type input, the input state will be "undefined" (could be high or low) during the transition from 0.8 to 2V.

# USING ANALOG INPUTS

The controller offers a 10-bit Analog-to-Digital converter that can be used on up to 6 Channels. To set a channel to the Analog mode, first issue the CHx.SETMODE(3) command to configure the channel to the Analog mode. You can then issue the CHx.GETANALOG command to get a value of between 0 to 1023 representing a voltage of between 0 to 5V DC on the corresponding channel. *It is important to note that when you issue the CHx.SETMODE(3) command, all channels up to and including the specified channel become Analog inputs, not just the specified channel. This behavior is by design.*
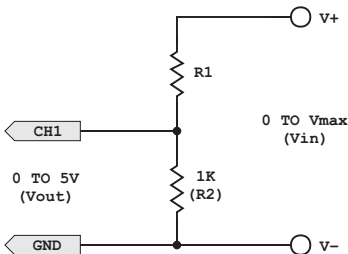
The maximum recommended impedance for analog sources is 10K Ohm and the absolute maximum voltage on in the input is 5V DC. **Exceeding 5V DC on any input will damage the controller.** Note that the accuracy of the analog to digital conversion depends on many factors, including the impedance of the analog source, keeping steady the voltage to be measured during the conversion as well as the stability of the circuit's supply voltage. Keep in mind that the voltage reference source for the A/D converter is the controller's supply voltage. Depending on the position of Jumper J1, this would be the USB bus voltage, or the controller's voltage regulator output when using an external DC supply. Some USB controllers or hubs may not put out exactly 5V which will affect the A/D conversion accuracy.

You can measure this voltage on the "G" and "5V" pins of the 8-pin header. The measured voltage would represent the full-scale value of the conversion. For example, if you measure a voltage of 4.95V and the A/D conversion returns 1023, it represents a voltage of 4.95V and not 5V.
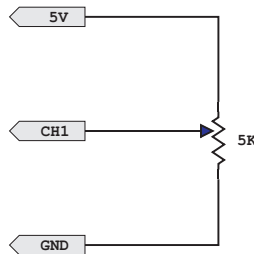
**Note that the ground of the controller will always need to be connected to the ground of the voltage to be measured. Furthermore, make sure you never apply a negative voltage or an AC voltage to any of the inputs. Doing so will damage the controller.**

If you need to measure a voltage higher than 5V DC, you may use a voltage divider as shown in the reference circuit below. An appropriate value for R1 should be selected based on the maximum voltage (Vmax) you expect to measure. Note that it is important to choose an appropriate value for R1 so that the voltage applied to the channel never exceeds 5V DC. You can also connect a variable resistor or potentiometer as show below.

```
        VOLTAGE DIVIDER                              POTENTIOMETER

                          O V+                         5V

                 R1

    CH1                  0 TO Vmax         CH1
                           (Vin)                        5K

   0 TO 5V        1K
    (Vout)       (R2)

    GND                   O V-             GND
```

```
Vout = [ R2 / (R1 + R2) ] * Vin ...
5 = [ 1000 / (R1 + 1000) ] * Vmax ...

       R1 = (Vmax x 200) - 1000

               Examples:

Vmax = 10V      R1 = 1000 Ohm  (1.0K Ohm)
Vmax = 12V      R1 = 1400 Ohm  (1.4K Ohm)
Vmax = 20V      R1 = 3000 Ohm  (3.0K Ohm)
```

# USING TEMPERATURE SENSORS

The controller supports connection to up to 6 independent temperature sensors and is compatible with Dallas/Maxim 1-Wire Digital Temperature Sensors.

We recommend using the DS18B20 High-Precision Temperature Sensor (available from Cana Kit).  The DS18B20 can measure temperatures from -55°C to +125°C (-67°F to +257°F) and has an accuracy of ±0.5°C in -10°C to +85°C (+14°F to +185°F) range.

The sensor should be connected to the corresponding input as shown in the diagram below. A 1K pull up resistor is required between the DQ and VDD pins.

When connecting the sensor to the controller with a long cable, it is recommended to add the resistor on the sensor side as opposed to the controller side.  Furthermore, for cable lengths of more than about 10 meters (~30 feet), it is recommended to also add a 10 to 47uF Tantalum (or Electrolytic) bypass capacitor (on the sensor side).  This helps provide power supply filtering and removes possible voltage fluctuations that may occur when the sensor is communicating with the controller.

Please note that the type of cable used can affect the digital signals when going over long distances.  Stereo shielded audio cable could be used for shorter distances of up to about 15 meters (~ 50 feet).  Flat telephone type cable can be used for distances of up to about 75 meters (~ 245 feet).  Finally, for longer distances, use CAT5 networking cable.  Using CAT5 networking cable, you can attach up to 6 sensors using the same network by using two wires for the GND and VDD pins of each sensor, and the remaining 6 wires for the DQ pins of each sensor.  In our labs, we have tested connecting up to 6 temperature sensors using CAT5 cable with a length of 100 meters (~325 feet).
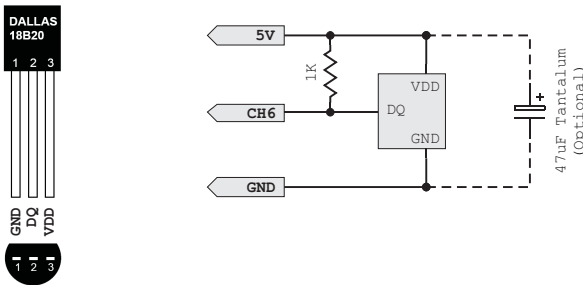
# SAMPLE CODE & DEMO APPLICATION

You can find sample code and a demo application, as well as additional information and downloads at:

**http://www.canakit.com/uk1104**