MidiTron™ Wireless

by Eroktronix

Manual version: 1.0

Introduction

MidiTron™ Wireless is a wireless sensor-to-MIDI interface. It consists of a compact sensor interface/transmitter unit and a corresponding receiver/output unit which interfaces via USB or MIDI. It provides 20 inputs in any combination of up to 10 analog inputs and 20 digital inputs.

MidiTron™ Wireless is designed to be a small, flexible, robust, long-range solution for creating wireless sensor controllers for real-time use in performance, installations and other scenarios.

MidiTron™ Wireless features

- Small transmitter (2 5/8” x 1 1/2” x 1 1/2”); encloses internal antenna and 9v battery
- Can be worn on body
- 1000 foot range under ideal conditions; several hundred foot range possible even with obstructions
- Uses 900 MHz frequency band for less interference than the increasingly crowded 2.4 GHz band
- Uses highly reliable RF chip set proven in live performance situations
- Eight selectable channels allow eight units to be used simultaneously or channel switching if interference is encountered
- Continuous re-transmit of sensor values virtually eliminates drop-out or lost data
- Very low latency transmit (maximum 7 ms with all sensors enabled)
- Very low latency USB and MIDI output (actual latency system dependent)
- Normal (7-bit) and high (10-bit) resolution analog modes available
- Easily configured with menu-driven programming patch
- No network configuration required as with Bluetooth or WiFi
- Standard MIDI and USB-MIDI input/output ports
- With USB, appears on computer as a standard MIDI device with no special drivers required
- Once configured, can be used as a standalone MIDI device without a computer
- Several interchangeable styles of connector boards allow flexibility in making sensor connections
- Three-pin style connector board allows direct interfacing of popular sensors available from other manufacturers

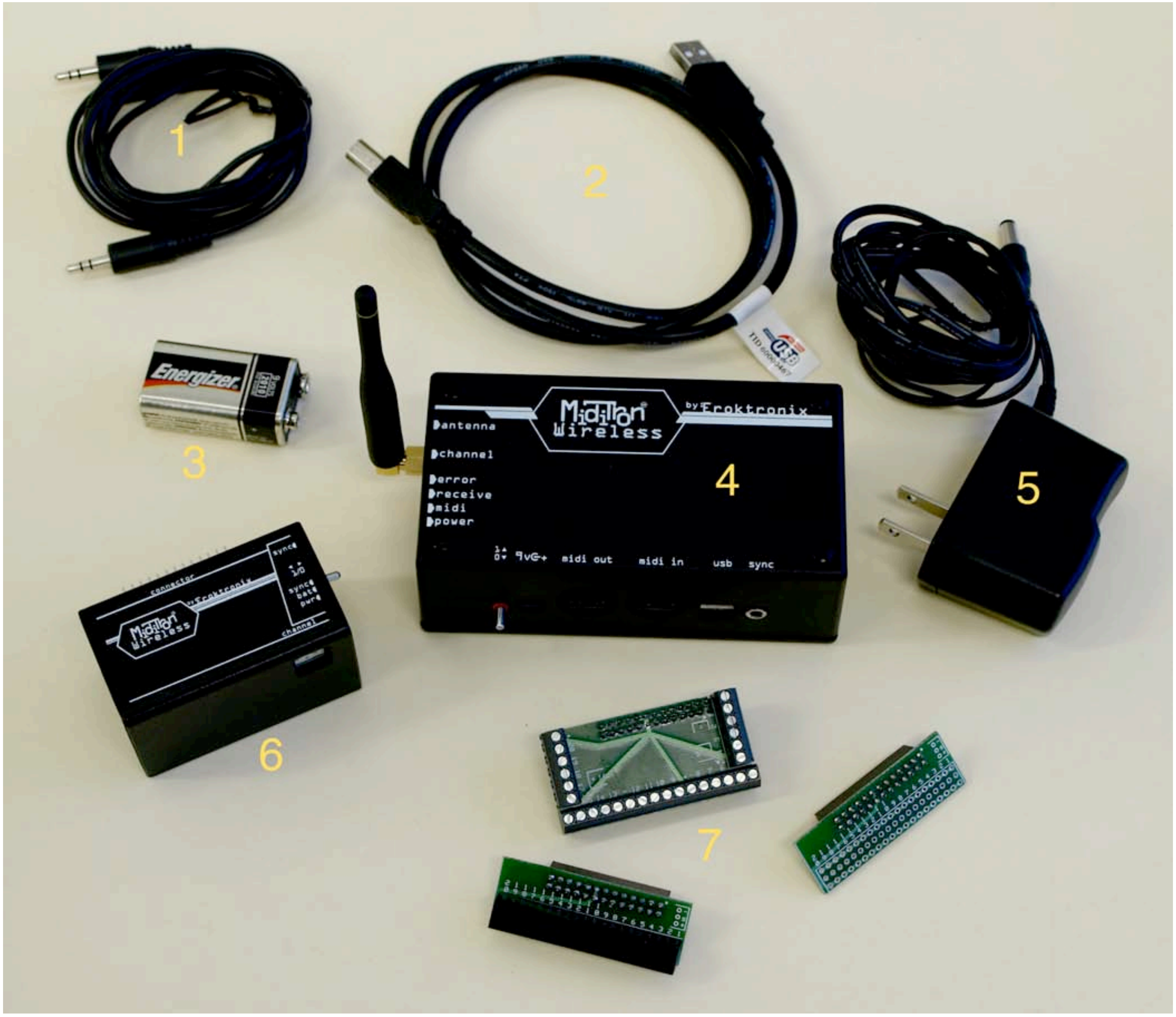


Figure 1: MidiTron™ Wireless with all included components

Included components

1) Sync cable
2) USB cable
3) 9v battery
4) MidiTron™ Wireless receiver
5) 9v receiver power supply
6) MidiTron™ Wireless transmitter
7) Connector boards

MidiTron™ Wireless Quick Start Guide

1) Gather together all components of your system, including the following:
- MidiTron™ Wireless transmitter and receiver units
- MidiTron™ Wireless connector board of your choice
- receiver power supply
- receiver USB cable, or a MIDI interface and two MIDI cables
- sync cable (1/8” stereo cable)
- sensors and associated components

2) Plug power into the receiver, power it on, then connect the receiver to your computer using USB or MIDI. If using USB, you should see a new MIDI device called “MidiTron Wireless 000” in your system dialog. On the Mac, look in Applications:Utilities:Audio MIDI Setup:MIDI Devices. In Windows, look in Control Panels:System:Hardware:Device Manager:Sound, Video and Game Controllers, then open the device called USB Audio Device. If using MIDI, you will communicate through your MIDI interface device. For MIDI, make sure to connect MIDI In to MIDI Out and vice-versa between the receiver and your interface.

3) Connect analog sensors to the connector board, beginning with input 1 and moving up consecutively. Follow these with your digital sensors, also connected consecutively.

4) Plug the connector board into the transmitter. When oriented properly, the connector board will hang down, away from the printed face of the transmitter.

5) Turn on the receiver. The green light should blink twice, and the red light should remain on. (If the yellow light comes on, this means the battery is low and should be replaced.)

6) Connect the transmitter and receiver together using the sync cable.

7) Open the programming patch. In the MIDI In and MIDI Out menus, select the ports to which the receiver is connected.

8) Enable each input you have connected to the transmitter using the Mode menus. As you enable inputs (in order beginning from 1), you will see each input’s transmitted value shown in its Receive box to the right of the red line.

9) Click *Open Max Examples*. If you haven’t changed the default parameter values (filled in each time you change an input’s Mode menu), you should see the sensor values on the corresponding check box (digital) or number box (analog).

MidiTron™ Wireless Detailed Guide

Transmitter

The MidiTron™ Wireless transmitter is a small unit that interfaces to sensors and transmits sensor data to the receiver. It has an internal antenna and is powered by a 9v battery. It has a 26-pin connector that interfaces with various connector boards to enable sensors to be connected. It can accommodate up to 20 analog or digital sensors (with a limit of 10 analog).

Figure 2: MidiTron™ Wireless Transmitter

1) Indicator lights
2) Power switch
3) Sync jack
4) Channel selector (8 position rotary dip switch)
5) 26-pin connector (dual row, .1” centers)

Indicator LEDs: The red LED is power. The yellow LED goes on when the battery is low. The green LED blinks each time sync data is received.

Channel selector: This switch must be set to the same channel number as the receiver (0-7). It is factory set to 0.

26-pin header: This header is for attaching one of a variety of connector boards. Each style of connector board provides a different way of connecting sensors (available styles are shown below). You may also construct your own cable, such as a 26-pin to DB-25 ribbon cable. The pin-out of this header is as follows:

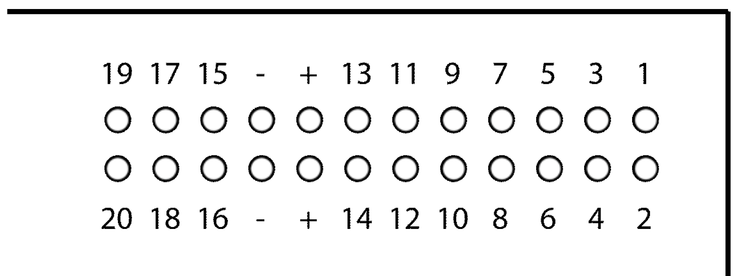


Figure 3: Pinout, looking at header in upper right corner of side of transmitter

Receiver

The MidiTron™ Wireless receiver receives sensor information continuously from the transmitter. It converts this information into MIDI commands based on user configuration. MIDI commands are then sent to a computer or other device via the receiver's MIDI Out port or to a computer via its USB port.

The receiver's USB port can be connected to either a Mac or PC and requires *no special drivers*. It will appear on the computer as a standard MIDI device with one In and one Out port. MIDI data will then be available for use in any program which can access the computer's MIDI devices.

1) Antenna
2) Channel selector (8 position rotary dip switch)
3) Indicator lights
4) Power switch
5) Power jack (9-15v, 200mA or greater, 2.1mm ID, 5.5mm OD, center +)
6) MIDI Out port
7) MIDI In port
8) USB port
9) Sync jack

Channel selector: This switch must be set to the same channel number as the transmitter (0-7). It is factory set to 0.

Indicator LEDs: (From right to left) The red LED is power. The yellow LED blinks when MIDI is received. The green LED turns on when there is a transmitter unit transmitting on the receiver's channel. The orange LED blinks on a receive error i.e. when bad data is received.

Receiver power

The receiver is powered by an external, wall-wart style, 9v, 200mA power supply with a 2.1mm ID, 5.5mm OD center-positive connector. If you replace this supply with your own one, make sure it is rated for at least 200mA and that it is center-positive. Its voltage can be higher than 9v (up to about 15v is safe).

Please note that the unit is not powered by USB. It must have the power supply connected.

Receiver status indicators

The green and orange LEDs indicate receive status. When a transmitter and receiver are on the same channel and in sync, the green LED should remain on and the orange LED should remain off. If the orange LED stays on or blinks often, it usually indicates that the transmitter and receiver are not in sync. Reprogramming the units should fix this.

If no transmitter is transmitting on a receiver's channel and the green light blinks often, it may indicate that there is interference on this channel. This may be insignificant, though you may want to switch the receiver to a different channel to find one without interference. Make sure to switch the transmitter to the same channel as well.

Troubleshooting USB-MIDI in Windows

In Windows XP, there are a number of factors that may cause MidiTron™ Wireless not to be available as a useable MIDI device. Here is a list of potential problems and solutions:
- MidiTron™ Wireless is likely to appear in your MIDI menu as the generically named "USB Audio Device." If it appears in a MIDI menu, it should work.
- Applications which monitor the USB port may interfere with proper operation. Uninstall them.
- Other devices which use the USB generic audio drivers (like webcams with audio) may interfere with operation. Uninstall them.
- Sometimes, plugging the MidiTron™ Wireless receiver into a USB hub, then the hub into the computer, may solve the problem.
- If all else fails, your best bet is to use a MIDI interface and the standard MIDI ports on the receiver, instead of direct USB-MIDI.

Connector boards

Sensors are normally connected to the transmitter using one of several available styles of connector boards, shown below. Connector boards must be plugged into the transmitter so that they hang down, away from the printed face of the transmitter (as in Figure 5). Each connector board provides access to the twenty input terminals and several access points each to power (labeled +) and ground (labeled -).

The three types of connector boards are as follows:
- solder, which provides +, terminal and - solder pads for each terminal
- 3-pin, which provides three pin holes on .1" centers for each terminal
- screw terminal, which provides a screw connection for each terminal plus several each for + and -

The 3-pin board will accept sensors from other manufacturers as long as they conform to the connector board and MidiTron™ Wireless specs.

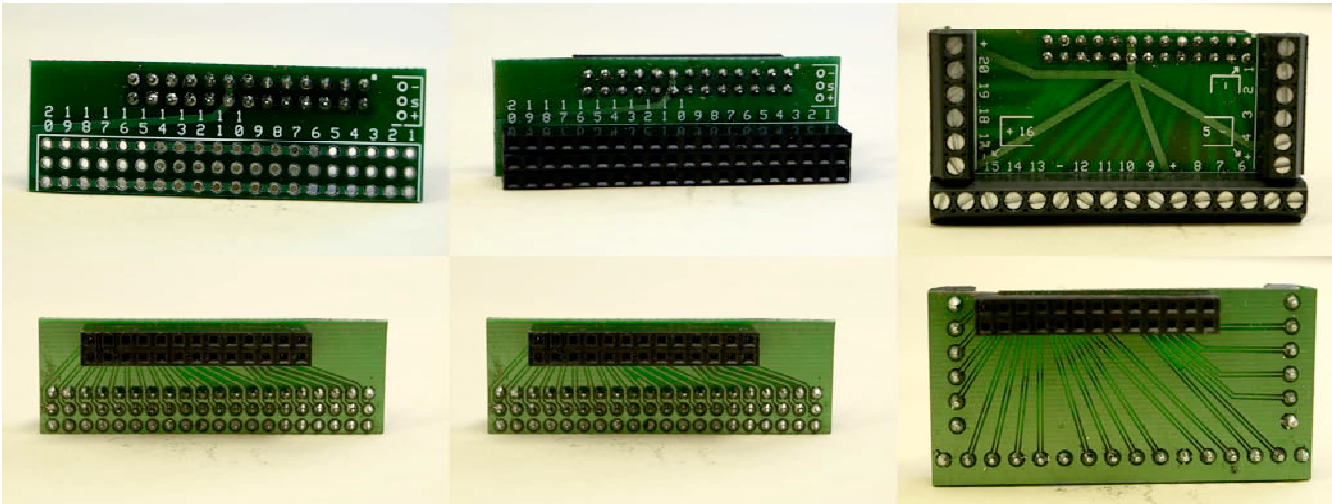


Figure 4: Connector boards front and back (L->R: solder, 3-pin, screw terminal)

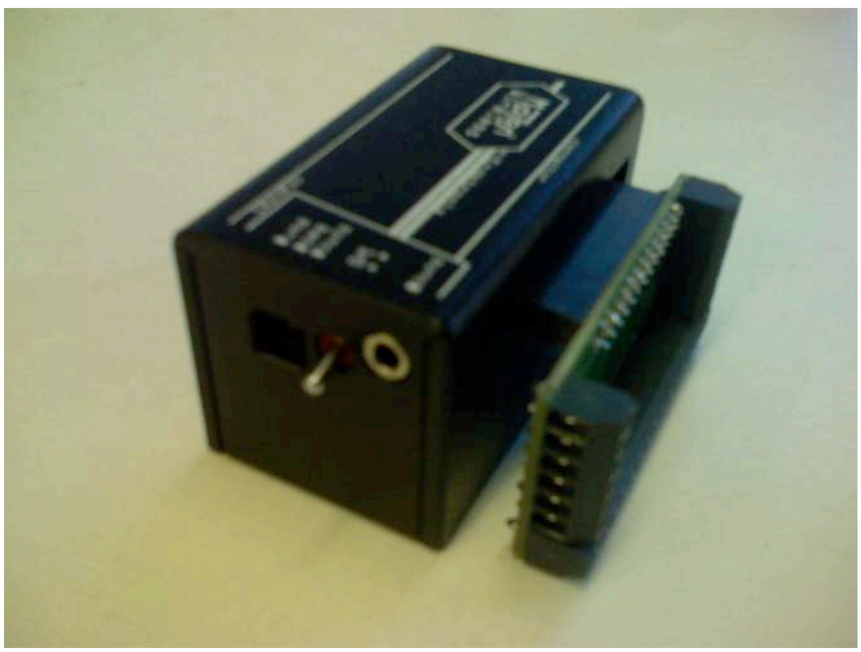


Figure 5: How to orient connector boards

Programming and syncing the units

Before the MidiTron™ Wireless system will work, it must be programmed and in sync. The system is programmed using a menu patch written in Max. This patch is supplied as a standalone application for Mac and PC, as well as a patch that can be run under Max or the free Max Runtime application, both available from http://cycling74.com. (You do not need to own Max to program the units.)

The programming patch (henceforth "programmer") lets you configure the system to indicate the number and type of inputs and what MIDI messages they should send, as well as some other system parameters. The patch is designed so that recommended default values for each input come up when you enable the input.

Before programming, the receiver must be connected to the computer via its USB or MIDI ports. Also, the receiver and transmitter **must be connected together** using the supplied sync cable. If the transmitter is not connected to the receiver during programming, it will not receive the necessary programming information; the units will be out of sync, causing the transmitter to send incorrect data to the receiver.

Once programmed, both units store the programming parameters in persistent ("flash") memory. The units will retain their programming even when powered off, until changed by reprogramming. Thus, the programming step is necessary only during setup, not during normal operation.

The transmitter's first ten inputs can be in one of three modes: Off, Digital Input, Analog Input. Its second set of ten inputs can be in one of two modes: Off, Digital Input.

For the transmitters's inputs to function properly, all analog inputs must come before any digital inputs and all inputs must be consecutive, starting at input 1. For example, if you use three analog inputs and three digital inputs, the analog inputs must be 1, 2 and 3. The digital inputs must then follow as 4, 5 and 6. Each mode has a set of configuration parameters specific to the context of the mode. Modes and configuration parameters are described in detail below.

MidiTron™ Wireless programmer software

The MidiTron™ Wireless programming software is implemented in Cycling 74's Max program and is supplied both as a stand-alone program for Mac and PC, as well as a set of Max patches which can be run in Max or Max Runtime.

After opening the programmer, you must first set the MIDI In and MIDI Out port menus. This tells the programmer to which ports the MidiTron™ Wireless receiver is connected.

The main body of the programmer screen consists of 20 configuration lines, corresponding to the 20 inputs on the MidiTron™ Wireless transmitter. Use the menus on the left side to set up any inputs you wish to connect (analog inputs first followed by digital). Leave unconnected inputs set to Off for maximum efficiency and to avoid having the system read unconnected ("floating") inputs.

When you change an input's mode, the line to the right of the menu will be filled in with options pertaining to the mode. By default, the settings are initialized with recommended values for the input and mode. These settings will work with the Max subpatch in the "Max Examples" window; Max users may copy and paste from this window to assist in their patch building. Depending on your application, of course, you may want to make changes to the default values. Every time you make a change to an input's mode or settings, the configuration for that input is sent immediately to MidiTron™ via sysex and stored in flash memory on both units.

For each enabled input, a number box (for analog) or check box (for digital) will appear to the right of the red line at the end of the configuration line. These boxes will automatically display input values received from the transmitter inputs.

Along the top of the window to the right of the port settings are the *Resend* and *Open Max Examples* buttons. Along the bottom of the window are controls labeled *Factory Reset, Read Config, Output Rate, Signal Strength Channel, Signal Strength Ctrl #, Signal Strength* meter and *Unit ID*. Here is a description of each:

Resend: Sends the current programmer settings to the receiver, and to the transmitter too, if it is connected via the sync cable (it should *always* be connected while programming). You might need to do this if the units aren't responding properly, or if you programmed the receiver and forgot to connect the transmitter sync.

Open Max Examples: Opens an example patch which (1) can program the units to desirable settings for all inputs, and (2) gives examples of how to read inputs in Max based on these settings. Max users can cut and paste from the example patch to help in building their own patches.

Factory Reset: Turns off all inputs and returns settings to their factory values. Use this if nothing is working properly, and you want to get back to a base state.

Read Config: Reads the current configuration from the receiver and sets the menus to these settings. Use this if you have just opened the programmer and wish to view or modify settings on the units.

Output Rate: When *Output Rate* is set to 0 (the default), MidiTron™ Wireless outputs data as quickly as possible, for all values that have changed since they were last output. When *Output Rate* is non-zero, it will send all values at a fixed interval. The interval is *Output Rate* * 2 milliseconds (approximately). This has two effects: (1) it can slow down the rate of MIDI, should you so desire, and (2) it effectively samples the inputs at a constant interval, allow you to compute the rate of change of inputs. Most users will not have a use for this, so don't worry if you don't understand it.

Signal Strength Channel and *Signal Strength Ctrl #*: These controls set the MIDI channel and controller number, respectively, for the receiver to output the strength of the signal received from the transmitter. This can be useful in checking for interference (when the transmitter is switched off) and transmission distance. When the channel is set to 0 (the default), the receiver will not output signal strength.

Signal Strength meter: The number box and slider in this section display the signal strength value from the receiver when *Signal Strength Channel* is set to a non-0 value. You can also receive and utilize this value using a *ctlin* box in Max or a control change input in any program that receives MIDI.

Unit ID: Used to set unique IDs for units in a multi-unit setup. This is described in detail below.

Floating inputs

If you have an input enabled but nothing connected to it, its value will "float." This means that the input will continuously read and transmit random values, which is confusing and (almost) never desirable. Set unused inputs to Off. Just to reiterate this point, a disconnected input is **not** zero; it is random and should be set to Off.

Digital input mode

When set to Digital, an input will accept 0v (or near 0v) to switch off and 5v (or near 5v) to switch on. When the input is switched on, it will send its “On” MIDI command; when switched off, it will send its “Off” MIDI command. Available commands for this mode are Note Off, Note On, Poly Pressure, Control Change, Program Change, Channel Pressure and Pitch Bend.

Analog input (AD) mode

When set to Analog, an input will accept a range of 0 to 5 volts and convert this voltage to a corresponding value using A/D conversion. It will send a MIDI message each time the converted value changes. Available commands for this mode are Poly Pressure, Control Change, Channel Pressure and Pitch Bend.

When using Poly Pressure, Control Change or Channel Pressure messages, the converted output value can range from 0-127. When using Pitch Bend, it can range from 0-16383 (but will jump by 16s, as the true resolution is 10 bits). Even in Pitch Bend mode though, the display will read 0-127 (i.e. low-res pitch bend). However, high-res pitch bend values are available to the program (Max or any other MIDI program you are using).

Analog inputs are available on inputs 1-10. Analog inputs must be consecutive, starting at input 1. That is, if you want 3 analog inputs, you must use inputs 1-3. MidiTron™ Wireless will automatically disable any inputs not configured in this way. For example, if you set inputs 1, 2 and 4 to be analog inputs, but not 3, input 4 will be disabled.

Analog input mode details

When setting up Analog input mode, you can control its input and output ranges to get the best results for the sensor you are using.

The input min and max are used to compensate for the fact that some sensor configurations do not output 0v at their minimum or 5v at their maximum. To adjust the input range properly, do the following:

1) To begin with, set the range values as follows:

    Input Min = 0
    Input Max = 127
    Output Min = 0
    Output Max = 127

2) Cause your sensor to output its minimum voltage (this is dependent on sensor type, of course; for example, it might mean darkening a photocell). Adjust Input Min until the input’s MIDI value reads 0.

You can adjust it so the sensor just hits zero at its minimum, or a little higher to provide a “dead zone” at the bottom of your sensor’s range.

3) Cause your sensor to output its maximum voltage. (Note: make sure this voltage never goes above 5v!) Adjust Input Max until the input’s MIDI value reads 127. Similarly to step 2, you can adjust it so the sensor just hits 127 at its maximum, or a little lower to provide a “dead zone” at the top of your sensor’s range.

4) Optional: You can also choose to limit the output range of the sensor and/or reverse its data direction. To limit the output range, adjust Output Min and Output Max. To reverse data direction, set Output Min to be higher than Output Max (for example, Output Min = 127, Output Max = 0). MidiTron™ Wireless maps the sensor’s minimum value to Output Min and its maximum value to Output Max, and scales intermediate values accordingly. (For pitch bend, although Output Min and Output Max are set using a range of 0-127, the output is mapped to 0-16383.)

MidiTron™ Wireless’s A/D conversion is done using 10-bit resolution. It then converts this value to the appropriate output value based on your range settings. When using pitch bend, the 10-bit A/D value is converted to a 14-bit pitch bend value. Thus, you will see stepping in the pitch bend value (i.e. the lowest four bit are insignificant), but the resolution is still higher than 7-bit (low-resolution) pitch bend.

Safe hookup practices

When hooking up devices to MidiTron™ Wireless, observe the following practices to minimize the possibility of damaging the unit or your devices.

1) Always turn the transmitter off when connecting and disconnecting devices.

2) Double-check your connections before switching the transmitter back on.

3) If you are concerned about your connections, it is recommended to have a volt meter connected to read the transmitter’s voltage between power and ground. This should read between about 4.9 and 5.1 volts. If it drops significantly below 5 volts, you may be creating a short from power to ground due to a wiring error.

Schematics

This collection of schematics covers the basics of hooking up most types of sensors. To learn how to wire sensors without using schematics, see the guide after this section.

Digital input: Switches

Switches provide a digital input and can be hooked up in the configurations shown in Figures 6 and 7. Figure 6 is termed *active high*, and Figure 7 is *active low*. Either configuration should work equally well for most applications. In the active high configuration, the voltage at the terminal will read 5v, or high, when the switch is depressed, causing MidiTron™ Wireless to send the input's *On* value. When the switch is released the terminal voltage will read 0v, or low, causing the *Off* value to be sent. If the circuit in Figure 7 is used, the opposite will occur (0v when pressed causing *Off* value to be sent, 5v when released causing *On* value to be sent).

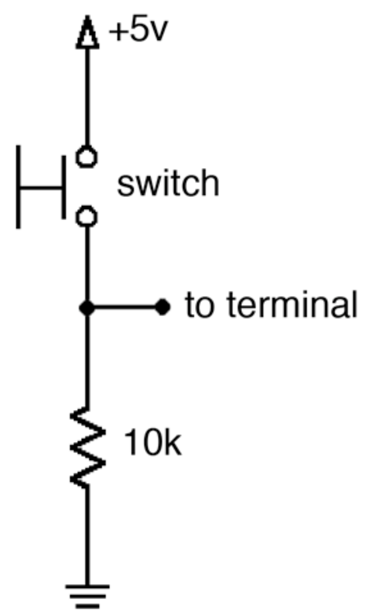


Figure 6: Switch, active high

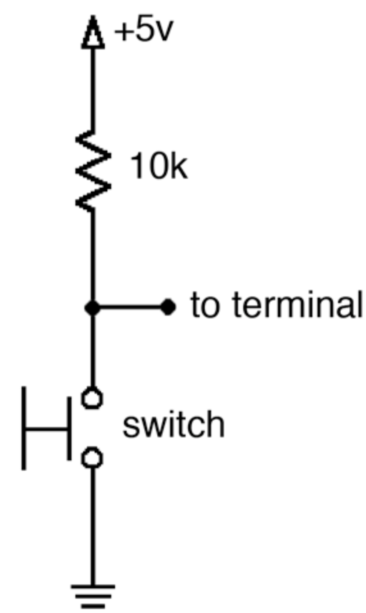


Figure 7: Switch, active low

Analog input: Potentiometers

A potentiometer (or “pot”) can be used to provide an analog voltage to a MidiTron™ input. Using the configuration in Figure 8, a pot will vary the voltage at the terminal from 0-5v - in other words, the full range of the input. 10k is a typical value for use in this configuration, but anything from 1k to 1M should work fine.

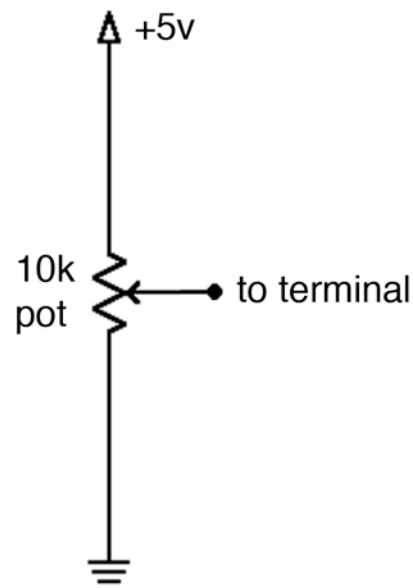


Figure 8: Potentiometer

Analog input: Resistive sensors

Resistive sensors, such as photoresistors, force sensing resistors (FSRs) or flex sensors, can be wired in either configuration shown in Figures 9 and 10. Using the Figure 9 configuration, the voltage to the terminal will increase with increasing resistance. Using the Figure 10 configuration, the terminal voltage will decrease with increasing resistance. Use whichever configuration is most convenient for your application.

You will also need to select an appropriate value for the fixed resistor R based on the resistance range of your sensor. In general, it is best to select a value for R that gives the largest voltage swing for the sensor. The optimal value is $R = sqrt(R_{min} * R_{max})$; that is, the square root of the sensor's minimum resistance times the sensor's maximum resistance.

Once you compute the optimal value, select a standard resistance value that is closest to it. This will give the maximum voltage range for your sensor and thus the best resolution. Use the Input min and max adjustments in the programmer to compensate for voltage offsets, as described above in the section **Analog input mode details**.

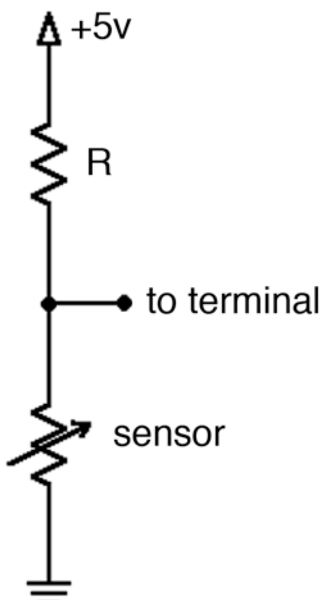


Figure 9: Resistive sensor (one way)

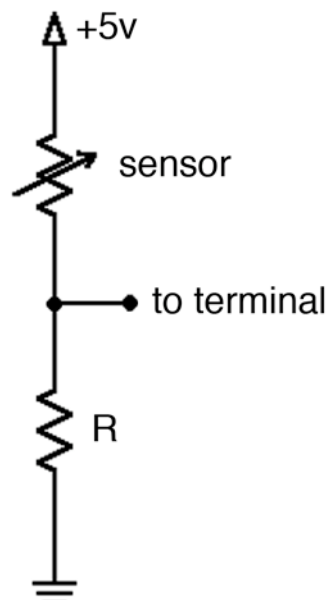


Figure 10: Resistive sensor (another way)

Resistive sensors and voltage dividers

Resistive sensors rely on the voltage divider principle to generate a varying voltage. It is therefore useful to understand the concept of a voltage divider circuit.

A voltage divider is created when an input voltage Vin is connected to two resistors, R1 and R2, and the output voltage Vout is taken at the point between the two resistors:

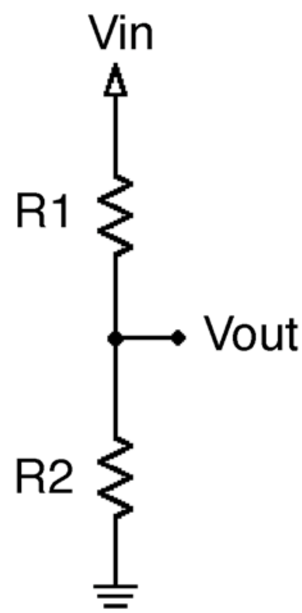


Figure 11: Voltage divider schematic

The voltage divider formula gives us the relationship between these four quantities:

Vout = Vin * R2 / (R1 + R2)

A simple example illustrates how this works. If we connect 5v to two 5k resistors in series (in a row), and then to ground, we have created a voltage divider. The voltage between the resistors (output voltage) will be 5v * 5k / (5k + 5k) = 5v * 1/2 = 2.5v.

We can consider a potentiometer to be a ready-made voltage divider, with the center tap being the Vout point. R1 + R2 is constant and equals the pot's value (as R1 goes up, R2 goes down and the total stays constant). Thus, we can see that the pot directly and linearly controls output voltage by varying R2.

In the case of two-terminal resistive sensors, we use a fixed resistor to create the voltage divider. Consider the case where R1 is the fixed resistor and R2 varies. (This works the same if reversed, but the voltage will move in the opposite direction.) Vin is constant at 5v, R1 is a constant value we select and R2 varies. Again, changing R2 (the resistive sensor) causes Vout to change. The only difference, which one can see by analyzing the formula, is that Vout will change logarithmically or exponentially with a change in R2. This is not a problem in most cases but can be compensated for in software such as Max if necessary.

Learning to wire sensors

If you want to wire your own sensors but don't wish to tackle schematic reading, this section will help. All sensors connected to the MidiTron™ Wireless transmitter need to provide a voltage between 0v and 5v. MidiTron™ Wireless accepts two different classes of sensors: digital and analog. Digital sensors provide an *on* or *off* signal by outputting voltages near 5v and 0v respectively. Analog sensors provide a varying voltage in the range 0-5v.

This section shows how to wire most common types of sensors to provide the necessary voltages and connections to MidiTron™ Wireless. In the discussions below, we will use the model of creating a device with three pins, which connect to +, a numbered terminal, and -, respectively (that is, power or 5v, a numbered transmitter terminal, and ground). We will refer to the center connection as "terminal" with the understanding that the resulting three-pin device can be plugged into any appropriate terminal by using the +/terminal/- connections of a connector board.

We will also refer to wires coming from your sensor. These could be wires you solder to the sensor terminals, wires clipped on by means of alligator or hook clips or wires preconnected by the manufacturer. We will assume that you understand how to make such connections. If you don't, see the section below on soldering sensors or refer to books or internet resources on making basic electronic connections.

Switches (digital)

Switches come in a huge variety of forms and functions. Some of the types you may encounter include manual (pushbuttons, slide, rocker, toggle, lever, etc.), tilt (mercury), reed (magnetic), alarm (floor pad) and more. They may have a single pole or circuit (meaning one set of contacts, designated single pole or SP), two (double pole or DP) or more (3P, 4P, etc.). They may provide two contacts for each pole (single throw or ST) or three contacts (common center and two outside contacts where one is on at a time while the other one is off, designated double throw or DT).

For example, a simple pushbutton switch is typically SPST. A DPDT slide switch would have two independent sets of three contacts (six total), with the slider connecting each center (common) terminal to either its "left" or "right" outer contact, depending on whether it is slid left or right.

Furthermore, momentary pushbutton switches (the type that spring back when released) are designated normally open (NO) or normally closed (NC), depending on whether their contacts are disconnected (NO) or connected (NC) when not being pushed.

Mechanical switches share one common property that we will utilize: they have at least one set of contacts that switches between being connected or disconnected. In one of these states, we need to generate 5v out; in the other, we need to generate 0v.

To do this, we'll first consider how to wire an NO SPST pushbutton. This switch has two contacts which are normally disconnected. In creating our three-pin device, the first step is to wire one contact to + and one to terminal. This allows us to send 5v to the terminal when the switch is on.

When the switch is off, however, we must ensure that the terminal has a connection to ground (see **Floating inputs** above). To do this, we connect a 10k resistor between terminal and -. This means we will have two connections to terminal (one leg of the switch and one of the resistor). This can be accomplished easily by splitting the insulation on the terminal wire of the switch an inch or two above the end and connecting the resistor lead there. You will probably then want to insulate this connection with shrink tubing or electrical tape.

Incidentally, the purpose of using a resistor instead of a direct connection to - is that a direct connection will lead to a short between + and - when the switch is pressed. A 10k resistor is sufficient to "pull down" or hold the terminal at ground when the switch is open, and draw very little current when the switch is closed.

Now that you have a switch wired as a three-pin device, it can be connected to a set of +/terminal/- connections on a connector board.

Active sensors (analog or digital)

Active sensors are ones that generate their own voltage output. These include most accelerometers, gyroscope and compass sensors, infrared and ultrasonic distance sensors, Hall-effect (magnetic) sensors and more.

Most sensors that generate a voltage within the range 0-5v can be used with MidiTron™ Wireless. These will be the easiest for beginners to work with.

(With some electronics knowledge, sensors that generate voltages higher than 5v can be voltage-divided or limited to 5v, but this should be done carefully, with a voltmeter used to ensure 5v is not exceeded. Sensors that output PWM can sometimes be RC filtered to generate a voltage. Sensors, such as certain ultrasonics, which require timing a pulse, and piezos, which require sampling and holding a pulse, can not be used.)

Most active sensors will have three wires or connection points, for power , output and ground. Typically, if prewired, power and ground wires will be red and black, respectively, but check the data sheet for the sensor before connecting. To use these sensors, simply connect their power, output and ground points or wires to +, terminal and -, respectively. Then configure the connected terminal for analog or digital, depending on whether the sensor provides an analog voltage range or digital (0v or 5v) output.

Potentiometers (analog)

Potentiometers, or pots, are variable resistors with three terminals. Two fixed outer terminals provide connections to the two ends of the pot's resistive material, and a moving center terminal provides a connection to a "wiper" contact which moves along the resistive material. This has the effect of creating a "voltage divider" circuit (see **Resistive sensors and voltage dividers** above).

Like switches, pots come in many shapes and styles, such as rotary (dial), slide (fader), multiturn, trimmer, ribbon (touch), etc. Almost all will have three terminals. To wire, identify the pot's wiper (center, moving terminal) and connect it to a consecutive analog input. Then connect the two outer contacts to + and -. It doesn't matter which outer terminal is connected to + or -. This simply controls the direction of voltage movement and can be reversed as desired.

Passive resistive sensors (analog)

It is likely that many of the sensors you will use will fall into this category. Almost all sensors of this type will have two leads, wires or terminals, and the resistance across these terminals will vary with the property being sensed. This is the type of sensor we will discuss in this section.

Examples of resistive sensors include photoresistors, force sensing resistors (FSRs), bend or flex sensors, thermistors and more. As these sensors function as a single variable resistor, and we need two resistance values to create a voltage divider (see **Resistive sensors and voltage dividers** above), we will need to add an appropriate fixed-value resistor to our wiring scheme.

To do so, use a multimeter to measure the resistance of your sensor at both ends of its scale. For example, if using a photoresistor, measure the resistance with the sensor in full light and total darkness. You can then calculate the optimal resistance value using the formula described above, in **Analog input: resistive sensors**, or simply select a resistor value somewhere between the min and max, which will generally yield acceptable results.

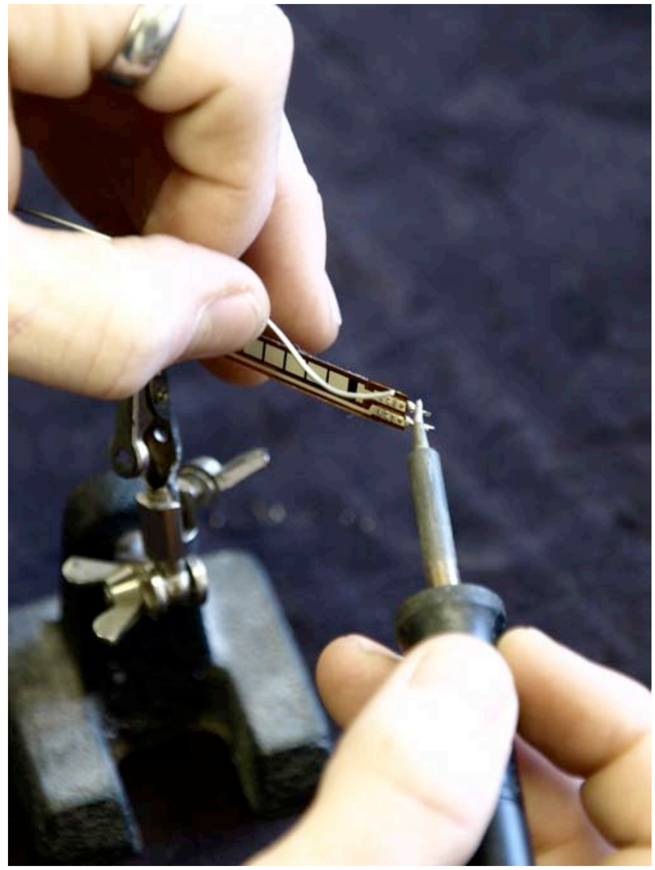
To wire, connect a red wire to one sensor terminal and a blue (or any other color besides black) wire to the other terminal. Next, connect a black wire to one end of the fixed resistor. Then, split the insulation on the blue wire at a convenient point in the middle. At this split point, connect the other end of the fixed resistor.

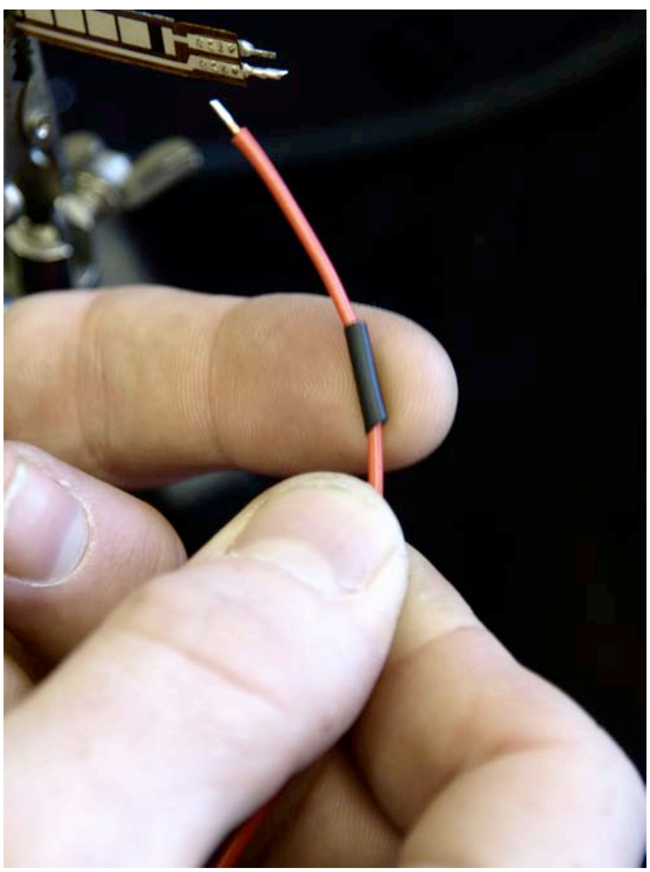
The result will be a three-wire device: a resistive voltage divider with your sensor and a corresponding fixed resistor. As usual, you can now connect red to +, black to - and blue to a consecutive analog input terminal. If you find that your sensor operates in the opposite direction that you wish (such as the analog value decreasing when you increase the bend of a flex sensor), it is permissible to exchange the red and black leads. This will reverse the operational direction of the sensor.

Sensor Wiring: A Visual Guide

The following is a step-by-step pictorial guide to wiring a sensor. Though the specific sensor shown is a passive resistive type, the general techniques are translatable to other types of sensors.

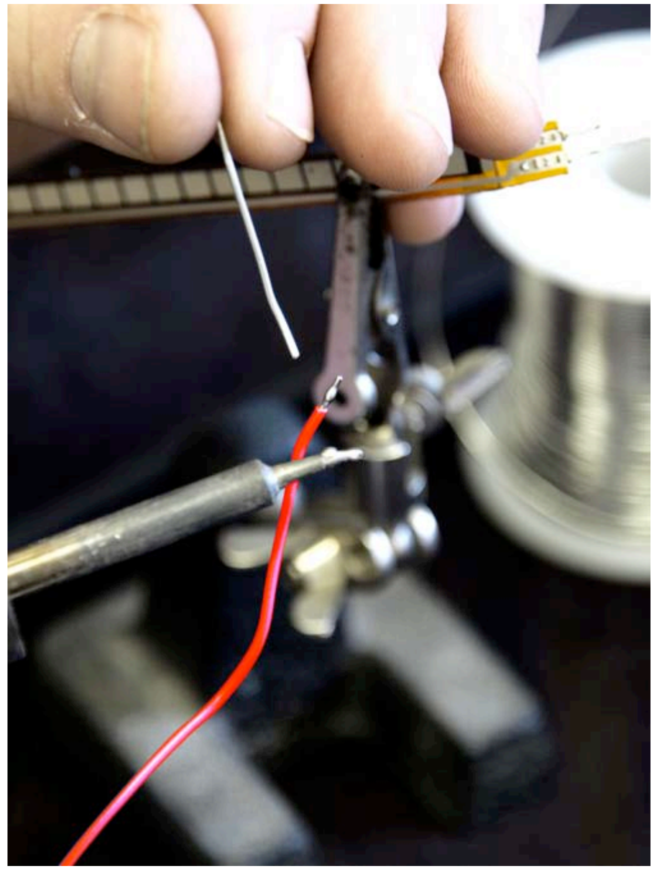
1) Get a good quality soldering iron, preferably with heat control. Get some 60/40 (or close) tin/lead rosin core solder, .031” diameter (or close). Begin by tinning the contacts on your sensor. Tinning means coating with solder. Use medium-low heat on your soldering iron. Touch the iron tip to the contact then melt solder over the tip onto the contact. Do not simply melt solder and drip it onto the contact. You need to heat the contact, have the rosin from the core flow over it followed by the solder. The catch is that you should accomplish the operation in 1-3 seconds, especially with delicate plastic body sensors. It is strongly advised that you practice soldering on throw-away components or wire first.

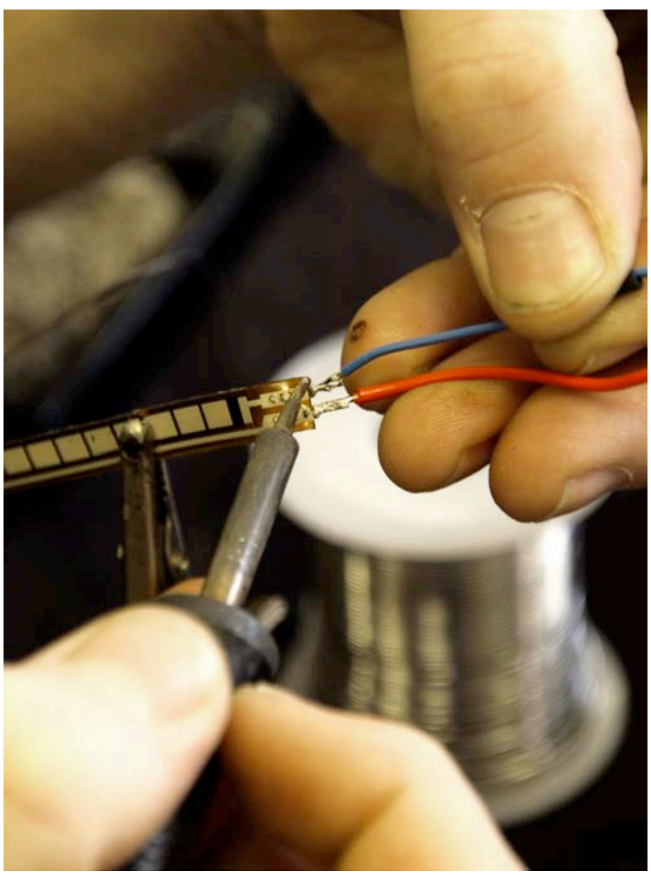




2) Cut some 22 AWG red stranded wire to your desired length. Strip about 1/4” or so of insulation off the end. Place a short length of heat shrink tubing over it. The heat shrink should slide freely over the wire but not be too much larger in diameter than the wire insulation.

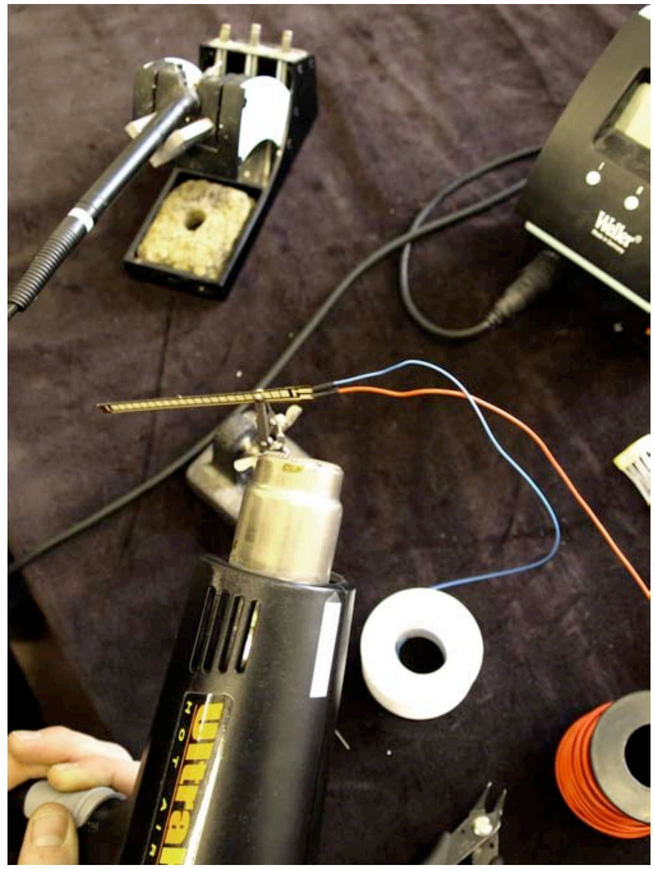
3) Tin the wire. That is, soak the stripped end thoroughly with solder.

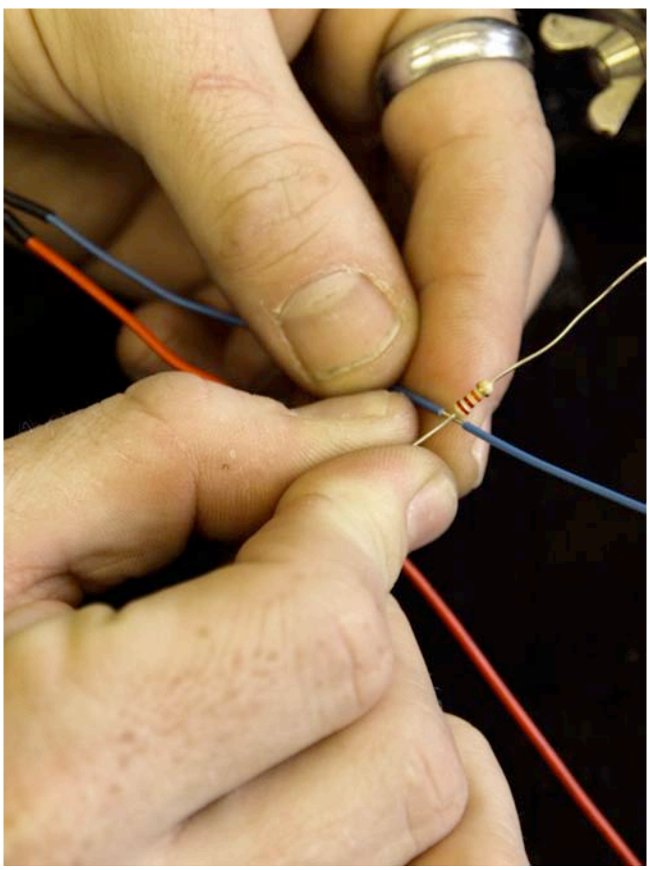




4) You will need “helping hands” for this step, or at least a weight to hold the sensor while you solder. Lay the tinned wire end on the tinned contact. Heat them briefly until the solder on both liquifies and melts together. Remove the heat, and hold the assembly perfectly still until the solder cools and solidifies (usually up to about 5 seconds). Repeat steps 2-4 for the second contact using a blue wire (or any other color besides black).

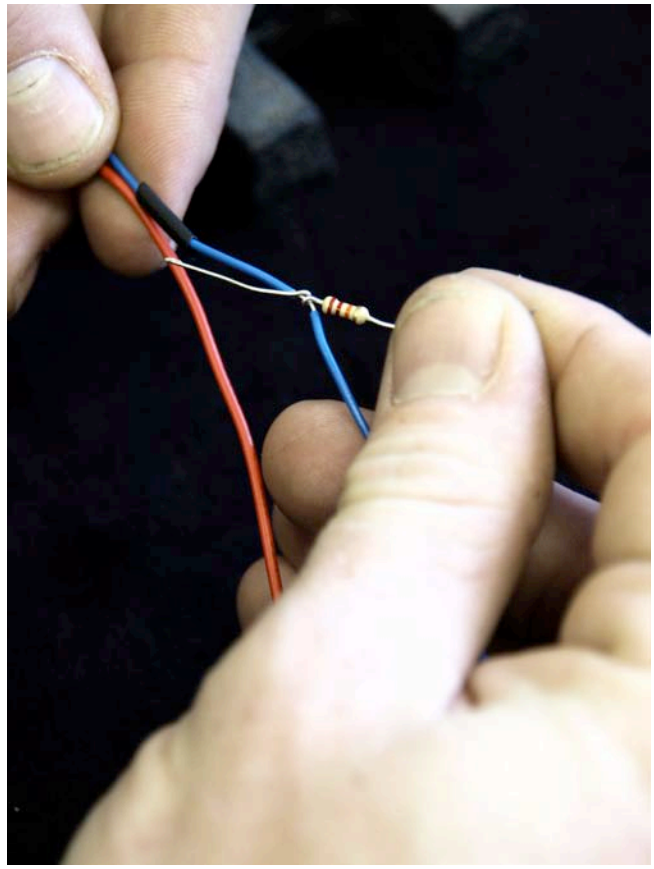
5) Move the heat shrink lengths over the soldered contacts. Shrink them by heating with a heat gun or hot hair dryer or by carefully waving a lighter back and forth under them.

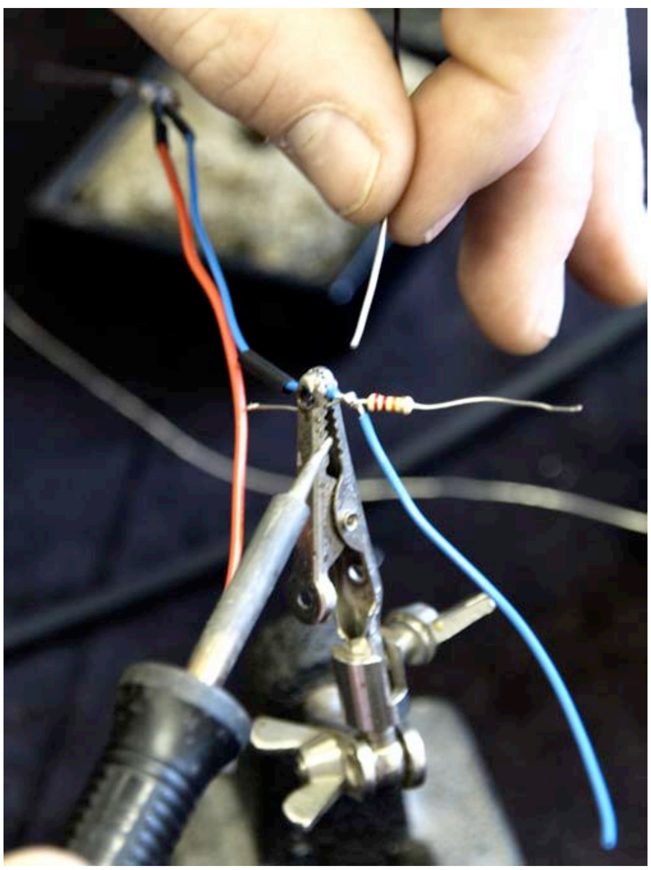




6) A few inches from the unsoldered end of the blue wire, strip about a 1/4” gap into the insulation. This can be done with wire strippers or with a knife. Select an appropriate value resistor for your sensor.

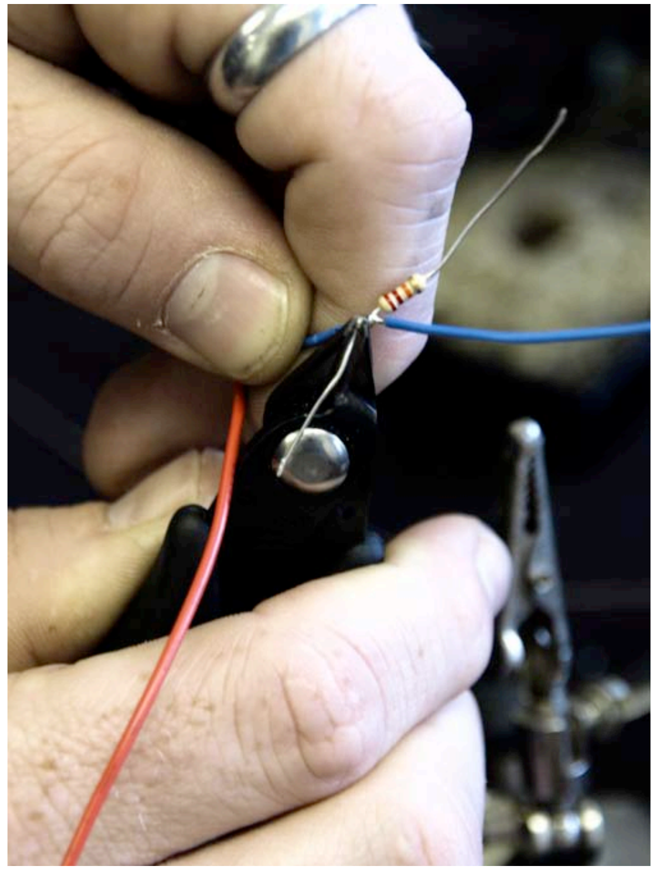
7) Wrap one end of the resistor around the gap in your blue wire.

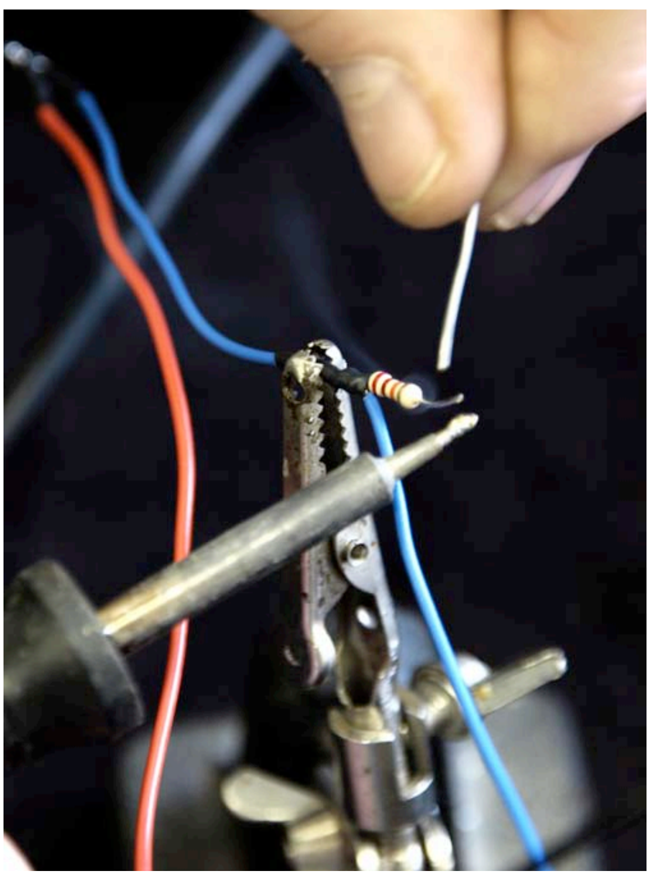




8) Solder the resistor to the gap using a good soaking of solder.

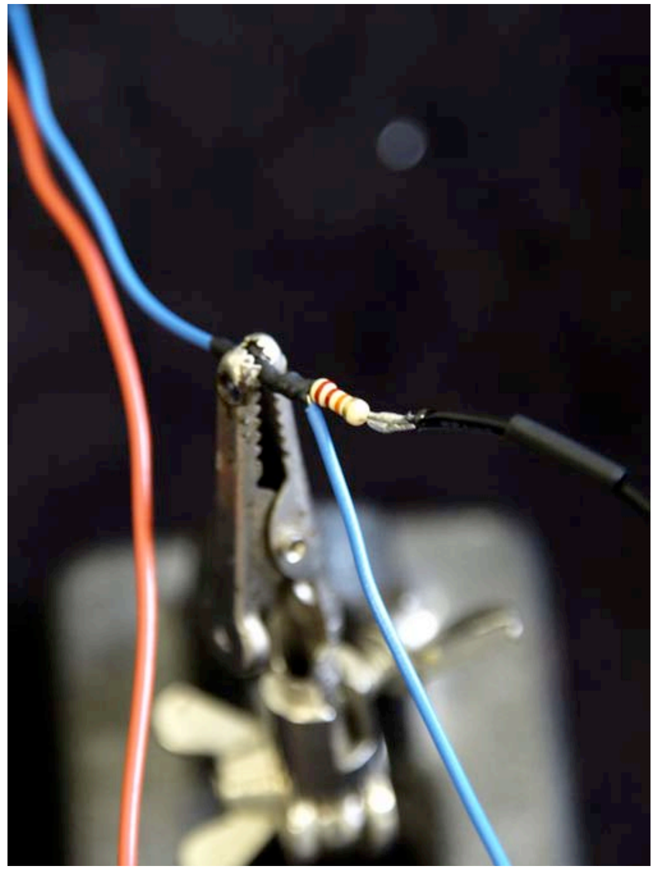
9) Trim the extra resistor wire from the soldered end.

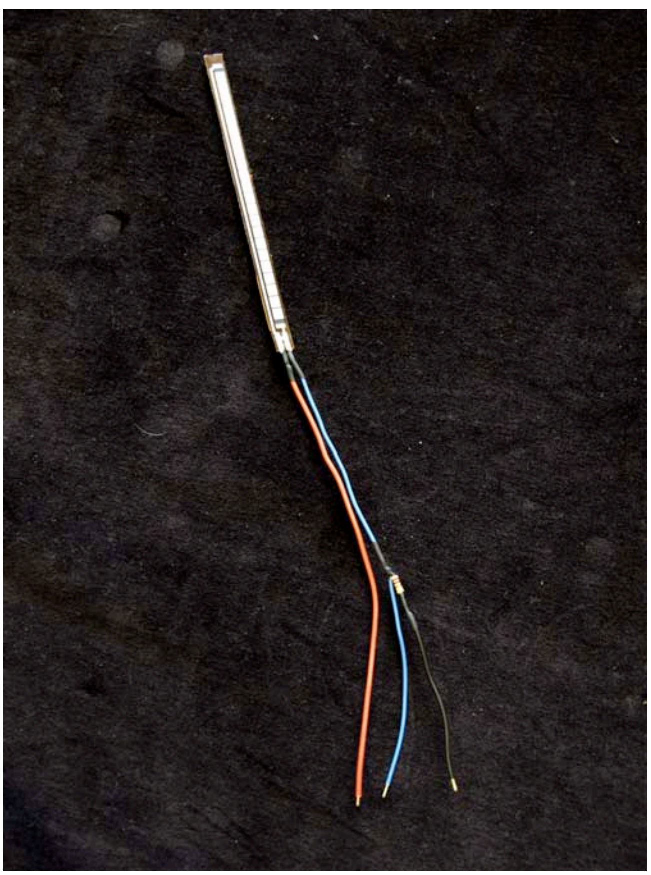




10) Trim the unsoldered end of the resistor to about 1/4”. Tin this end.

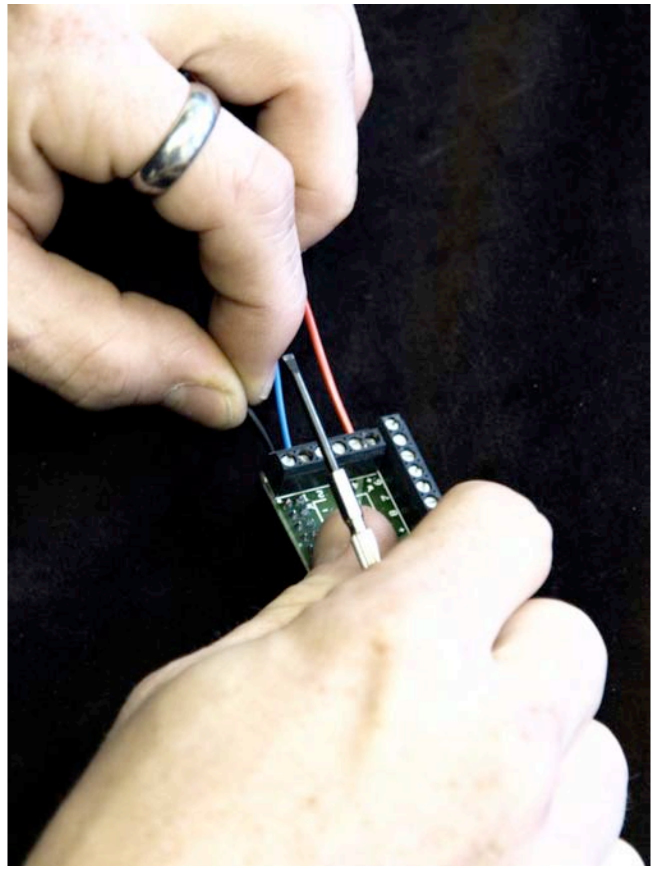
11) Cut a length of black wire long enough to reach from the end of your resistor to the ends of the other wires. Strip and tin one end of the wire as before. Bring this end to the resistor's free end and heat the two to meld the solder. Hold steady until solidified. Though shown differently in this picture, it is now preferable to put shrink tubing over the blue and black wires and the resistor. The shrink tubing should cover the resistor and both connections. Shrink as before.

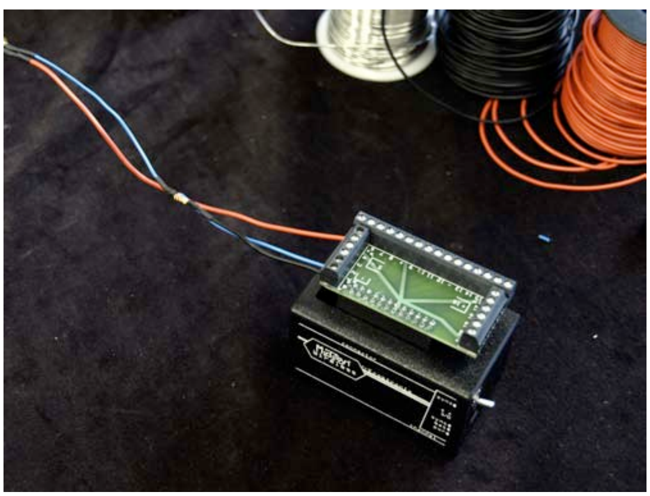




12) Trim all three wires to the same length. Strip 1/4" of insulation off each wire, then tin. You now have a wired sensor that's ready for connection.

13) Select the MidiTron™ Wireless connector board of your choice. Connect the red wire to +, black to - and blue to 1 (or if you have other analog sensors already connected, to the next consecutive numbered terminal). (Note: if you are using the 3-pin plug-in connector board, you will probably need to solder a three pin connector onto the wires in order to plug it in.)





14) This shows the completed sensor wired and attached to the transmitter.

Troubleshooting

This section will help you solve the most common problems you may encounter using MidiTron™ Wireless. The following checklist will help you get your units are working properly.

You should have a multimeter when testing. They are inexpensive and are an essential troubleshooting tool. If you have never used one, look for an online tutorial or check the manual if one came with your meter.

Disconnect the connector board from your transmitter if one is connected.

Make sure the battery in your transmitter is fresh. Normally, the low battery (yellow) LED will light when the battery is low. However, if it is too low, it may not. You can check the battery by switching the transmitter on, opening the battery door, then testing the voltage across the battery terminals (black multimeter lead to -, red to +). Your meter should read at least 7.5 volts.

On each unit, make sure its green light blinks twice when you turn it on, and that the red light stays on.

With both units powered on, connect the receiver to your computer using either USB or MIDI cables. Connect the sync cable between both units.

Check the channel dials on both units. Make sure they are both set to the same channel number (0-7). Make sure the antenna is connected securely to the receiver unit.

Open the MIDI control panel for your computer (see **MidiTron™ Wireless Quick Start Guide** for instructions on accessing this). If connecting via USB, make sure that your panel shows the “MidiTron Wireless 000” device. If it doesn’t, try disconnecting, power cycling and reconnecting the receiver.

On the Mac, you might want to install the freeware program “USB Monitor”. This will tell you when USB devices are connected and disconnected.

On Windows, the first time the device is connected, Windows should report that new hardware has been connected, then the device name, then that the new hardware is ready to use. If it doesn’t or if “USB Audio Device” doesn’t show up as an available device in Max and other MIDI programs, you may have USB driver problems with Windows. Sometimes these can be solved by deinstalling any software that may modify or interfere with access to the USB ports (such as port monitor software) or by reinstalling Windows. If these suggestions don’t solve your problem, you may have to use a USB-MIDI interface and connect via MIDI cables instead. (See the section **Troubleshooting USB-MIDI in Windows** above for more advice.)

If connecting via MIDI, make sure that MIDI Out on your interface is connected to MIDI In on the receiver, and MIDI In on your interface is connected to MIDI Out on the receiver. Also make sure that your interface shows up in the MIDI control panel on your computer.

Open the programming patch (as a standalone or in Max or Max Runtime). Select “MidiTron Wireless 000” on the Mac or “USB Audio Device” on Windows if connected via USB or the port on your MIDI interface that is connected to the receiver if connected via MIDI in both the *MIDI in port* and *MIDI out port* menus. Note that you must click and release both menus so they change from red to black and stop flashing for the ports to be selected.

Click the *Factory Reset!* button in the programmer and observe the lights on each unit. Each time you click, you should see the yellow light on the receiver and the green light on the transmitter blink. If they don't, check all cables and connections, swap out your MIDI and USB cables if you have extras, and make sure the port settings in the programmer are correct. (Also, all other values on the bottom line of the programmer window should be set to 0.)

Click *Open Max Examples*, then click the button pointed to by the text *Analog inputs =>*. This will enable the first ten inputs as analog on the transmitter. Since there is nothing connected to them, they are "floating" and will read random values. Run your fingers along the exposed pins on the transmitter. This should cause the values in the analog section of the Examples window (and likewise, the Receive boxes in the main window) to fluctuate wildly.

If you are not receiving values, try clicking the Resend button. Also, check to make sure that the green Receive light on the receiver is on constantly and that the orange Error light is flashing rarely or not at all. If this is not the case, change to a different (and identical) channel on both units.

If you have followed the above steps and your units are working, congratulations. If they are not, run through the steps again carefully. If they are still not working, email support@eroktronix.com for help.

Connecting multiple MidiTron™ Wireless units

Connecting multiple units to a single computer is easy. First, make sure each transmitter/receiver pair is set to a different transmit channel. Then, if using MIDI, simply connect each receiver to a different set of MIDI ports on your multi-port MIDI interface.

If using USB, you will need to take some additional steps to give each unit a different ID (and therefore a different USB name). Quit the programmer, then connect one (and only one) receiver to the computer via USB. Reopen the programmer and select the receiver in the MIDI in/out port menus.

Now, change the Unit ID to a unique number (0-127) for this unit. Quit the programmer again, then disconnect USB from the receiver and switch it off for a few seconds. Switch it back on, reconnect USB and reopen the programmer. The unit should now appear with the new ID in the name. For example, if you set it to 1, the name will be "MidiTron Wireless 001".

Repeat this process for each receiver you wish to connect, one at a time. These IDs are memorized by the receivers so you will only need to do this process once. It is also not a bad idea to set the ID of each unit to the same as the channel, and label the corresponding receiver and transmitter with the channel/ID number.

Once all units have unique IDs, you can connect all of them to your computer directly or through a USB hub, and they should appear as unique MIDI devices.

Sensors and sources

Book
Physical Computing: Sensing and Controlling the Physical World with Computers: This is the bible on the subject of sensors, microprocessors and more. Consider it an essential guide if you're getting started with this type of work.

Web sites
http://www.tigoe.net/pcomp: the web site by one of the authors of *Physical Computing* - all things about hooking up electronics, sensors, microcontrollers, etc.
http://www.ircam.fr/equipes/analyse-synthese/wanderle/Gestes/Externe/index.html: A comprehensive reference on sensor based instruments
http://nime.org: information on the New Interfaces for Musical Expression conference, including online proceedings
http://www.pacificsites.com/~brooke/Sensors.shtml: fairly exotic list of sensors

Electronics (new)
http://digikey.com: one of the most extensive selections of electronics
http://jameco.com: less overwhelming than digikey.com and often has most of what you might need, good for bend sensors and motors
http://mouser.com: extensive selection, some hard-to-find items, good source for sliders (linear pots)
http://sparkfun.com: great source for sensors and other items for the experimenter
http://acroname.com: another good source for sensors

Electronics (surplus)
http://allelectronics.com and http://goldmine-elec.com: Great sources for inexpensive parts

Electronics search engines
http://findchips.com: searches about 20 electronics companies at once for what you want
http://globalspec.com: "The Engineering Search Engine"

Force Sensing Resistors (FSRs)
http://interlinkelec.com: They sell a kit with a lot of FSRs. They will also sell them individually (with a $60 minimum order), but you have to call or email them, as this is not listed on the site.

Capacitive
http://qprox.cox: Similar to Theremin-type sensing, they offer both on-off and continuous sensing.

Video Game Parts
http://happcontrols.com: the source for all things related to video games (buttons, joysticks, etc.)

Some sensors and devices you might like to try
Most of these devices can be found from the electronics suppliers listed above.
switches: pushbutton, momentary, on-off, snap action, lever, video game, joysticks, reed (magnetic), Hall effect (magnetic), carpet, floor, window, door
potentiometers: standard, multiturn, linear (slider), joystick

variable resistors: flex/bend, force sensing (FSR), photocell, thermistor (temperature)
other sensors: wind/air pressure, accelerometer, infrared (proximity), ultrasonic (proximity)
lights and control: relays, LEDs, electroluminescent wire, low-voltage DC bulbs (with transistor drivers), AC switching (with relays)
robotics: DC motors, stepper motors, solenoids

Electrical Specifications

Receiver power requirements	9-15 VDC, 200 mA or greater
Transmitter power output	5 VDC from "+" terminal, 1A max (though heat sinking may be required to achieve maximum current)
Terminal digital input voltage	low level = 0.0 - 0.8v; high level = 2.0 - 5.0v
Terminal analog input voltage	0.0 - 5.0v
Terminal input leakage current	1 μA