A very quick and dirty introduction to Sensors, Microcontrollers, and Electronics

Part Three: how sensors and actuators work and how to hook them up to a microcontroller

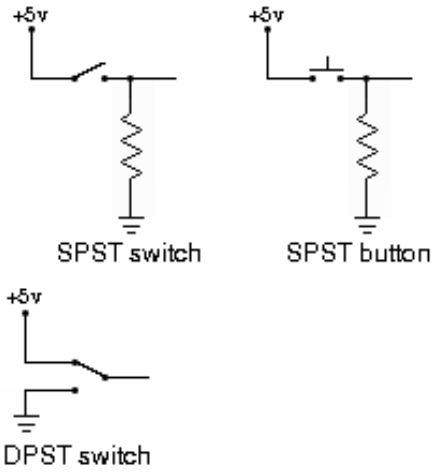
There are gazillions of different kinds of sensors and actuators. You probably use at least a hundred of them a day. Luckily for us they mostly fall into a few broad categories, as we saw in part one. It generally not necessary to know the intimate details of how a sensor/actuator works in order to use it. Most of the time it's sufficient to simply understand how to get information and electricity into and out of the device.

Sensors

button/switch

These are easy. Most buttons/switches have either two or three terminals. When you press the button or flip the switch contacts between the terminals are made and/or broken. The details of what happens depends on the exact configuration of the button/switch. Some of the variables are: number of switch positions (poles), number of switches (throws), normal state (open or closed) of each positions.

A single pole, single throw (SPST) switch is the simplest kind. A double pole, single throw switch is slightly more complicated. They look like this:



To use a SPST switch, you attach a voltage source to one terminal of the switch. When you flip the switch a connection is made between the two terminals, which means that current can now flow from one terminal to the other. That doesn't sound very exciting, until you remember that a microcontroller has digital input pins. By hooking up a button between a +5v voltage source and the input pin on a microcontroller you can sense whether the button is pressed or not. When it's not pressed, the microcontroller will see no voltage on its input pin (a "0"). But when it's pressed it'll see +5v (a "1"). (Sometimes you need a "pulldown resistor" on the pin so that the pin really sees ground when the button isn't pressed. It depends on the microcontroller.)

There are lots prefabricated buttons and switches available. You can also make your own pretty easily -- all you need is two pieces of conducting metal! Pretty much any instrument or device that has a make/break contact mechanism (clarinet, piano, hi-hat, phone, door, foot on the floor, window, etc.)

Can be wired up as a button/switch sensor.

potentiometer

A potentiometer (or pot) is a simple knob that works as a voltage divider with two variable resistors. Small potentiometers are called trim pots. They come in many shapes and size and value ranges. A potentiometer will be labeled with its total resistance. Potentiometers are either linear or logarithmic ("audio"). When using a linear pot Vout will change linearly as the knob is turned. When using a log pot Vout will change logarithmically as the knob is turned.

A pot has three terminals (or legs), which correspond to Vin, Vout and ground in the voltage divider diagram in the previous section. You hook your +v up to the first leg (Vin), ground up to the third leg (ground) and use the middle leg as your output (Vout). As you turn the knob on the pot the voltage on the middle leg will get bigger/smaller.

Potentiometers are very useful for setting parameters, like the speed of a process or the brightness of a light. They're also good for measuring the rotational angle of an object about an axis. Log pots are good for controlling non-linear parameters like pitch; they give you more resolution (change slowly) in the bass and less (change quickly) in the treble.

pressure sensor

A pressure sensor is generally a variable resistor that changes resistance based on how hard it is pressed. They are also known as Force Sensing Resistors or FSRs. They come in many shapes and sizes, from tiny little dots to long strips. They're pretty flexible, and can usually be cut to size.

Often you use an FSR as the variable resistor in a voltage divider circuit. By using one FSR and one fixed resistor and hooking Vout to an A/D converter you can give the microcontroller some idea of how much pressure is being put on the FSR. If you don't have an A/D converter handy, there's a simple RCtime circuit that you can use to do more-or-less the same thing.

FSRs can be used in many places that a button is used, with the advantage that you can not only tell that the button is being pressed, but also how hard. They're used a lot in things like velocity-sensitive keyboards, but are also good for use in floor mats, on the sides of instruments, on finger tips...Basically anywhere there's going to be a force that you want to measure.

light sensor

The most common light sensor is the CDC photo sensor. It works like a variable resistor with a value that changes based on how much light it receives. Other types are photo-transistors and photo-diodes.

Photo sensors are used in much the same way as pressure sensors. If you hook one up to a microcontroller via a voltage divider (or RCtime) circuit, the microcontroller can sense how much light is falling on the device.

beam breaker

A beam breaker is a non-physical switch or trigger. The simplest way to make one is to aim a cheap laser pointer at a photo-sensor. When the beam is broken, the resistance of the photo-resistor changes. If the photo resistor is hooked up to a microcontroller, it can sense whether the beam is intact or broken. So simple!

proximity sensor

There are a variety of ways to implement proximity sensing. The most common are infrared and ultrasound. They're both usually used as packaged devices that hook up to a microcontroller via one or more digital i/o lines.

An infrared sensor works by sending out pulses of infrared (invisible) light. It then tries to detect reflections of that light from nearby objects. If it detects a reflection then it assumes that there's an object nearby and puts out a digital "1". Otherwise it puts out a "0". There are some tricks to get a rough distance measurement out of an infrared detector, but most often they're used for simple object detection.

An ultrasonic proximity detector works by putting out ultrasonic (inaudible) pulses of sound. It then measures the length of time it takes those pulses to hit nearby objects and return as echos. The longer the time, the farther away the object. Ultrasonic proximity detectors are good for measuring short distances (maybe a couple yards). It's usually up to the microcontroller to do the counting and distance computations; the detector takes care of sending and detecting the ultrasonic pulses.

accelerometer

Accelerometers are cool. They're small ICs with tiny little suspended weights embedded in them. By measuring the force of gravity/inertia on those weights the device can emit a digital signal that tells you about its orientation in space.

Like proximity sensors, accelerometers are hooked up to a microcontroller via a few digital i/o lines.

microphone

Most microcontrollers aren't fast enough to be useful for recording sound via a mic. But they can do some basic things like listen for suddens sounds or listen for silence. They can also be used to control sound recording devices.

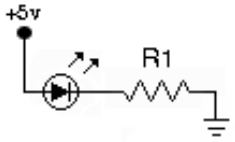
contact mic

These are made from piezo elements. A piezo element is made from a curious type of ceramic material that gives off electricity when it's deformed (this is called the piezo-electric effect). The voltage created is (roughly) proportional to how much the element is deformed. This property lets it be used as a contact mic. You can scream all you want into a contact mic and it won't hear you. But rub it against something, and it will wail! There are other types of piezo materials, like piezo rubber, that could be fun to play with.

actuators

LEDs

LEDs are Light Emitting Diodes. They work just like regular diodes, except that they give off light when you put current through them. The brightness of the light is determined by the amount of current. But watch out; you can't just crank up an LED to eleven. If you put too much current through an LED it will blow up. Bad. To limit the current through an LED you use a current limiting resistor. As so:



If the LED specs say it can only handle 20 milliamps of current (0.02 Amps), and the voltage source is at +5v, what's the smallest current limiting resistor you can use?

R = V/I
R = 5/.02
R = 250 Ohms

Remember, that's the *smallest* resistor you can safely use. You'd probably want to use something bigger just to be safe.

motor

A motor transfers electrical energy into motion. There are tiny motors everywhere -- cell phones (vibrate mode), printers, hair dryers, cameras, robo-dogs.

Most microcontrollers can't provide enough current to drive a motor directly. You have to use some additional circuitry to power the motor. The microcontroller then controls that circuitry. There are a number of different kinds of motors. More on motors later.

solenoid

A solenoid is kind of like a cross between a speaker and a motor. When you energize a solenoid its shaft is either pushed in or pulled out, depending on the type. When you de-energize it, the shaft returns to its starting point (usually via a spring or other return mechanism). Things like electrical latches are often driven by solenoids.

Like motors, solenoids can't be driven directly by a microcontroller; you need extra circuitry to power the device.

speaker

You know what a speaker does. You can't generally hook a speak directly to a microcontroller. You need some sort of amplifier in between them, as the microcontroller won't have enough power to drive the speaker directly. Most microcontrollers aren't fast enough to put out high-quality sound (see microphones, above.) But you can use them to generate simple bleeps and buzzes, or you can use them to control other sound generating devices (CD players, samplers, etc.)

piezo element speaker

We saw piezo elements above as contact mics. They can also be used as speakers, since the piezo electric effect goes both ways: deform the ceramic and it puts out a voltage. Put a voltage in and the ceramic deforms. Cool.

The nice thing about using a piezo element as a speaker is that it takes very little power to drive it. You can hook a piezo element directly to a pin on a microcontroller and then generate a PWM waveform by toggling that pin on and off (from +5v to ground). Sound will then come out of the piezo element! It won't be very loud, but it's good enough for simple buzzers and whatnot. Many cheapo electronic devices have piezo buzzers in them.

What's next?

You have to actually do something now!

Part Four: the assignment