Sensors and Actuators

Sensors and actuators are electronic components that allow a piece of electronics to interact with the world.

As the microcontroller is a very simple computer, it can process only electric signals (a bit like the electric pulses that are sent between neurons in our brains). For it to sense light, temperature, or other physical quantities, it needs something that can convert them into electricity. In our body, for example, the eye converts light into signals that get sent to the brain using nerves. In electronics, we can use a simple device called a light-dependent resistor (an LDR or photoresistor) that can measure the amount of light that hits it and report it as a signal that can be understood by the microcontroller.

Once the sensors have been read, the device has the information needed to decide how to react. The decision-making process is handled by the microcontroller, and the reaction is performed by actuators. In our bodies, for example, muscles receive electric signals from the brain and convert them into a movement. In the electronic world, these functions could be performed by a light or an electric motor.

In the following sections, you will learn how to read sensors of different types and control different kinds of actuators.

Blinking an LED

The LED blinking sketch is the first program that you should run to test whether your Arduino board is working and is configured correctly. It is also usually the very first programming exercise someone does when learning to program a microcontroller. A light-emitting diode (LED) is a small electronic component that's a bit like a light bulb, but is more efficient and requires lower voltages to operate.

Your Arduino board comes with an LED preinstalled. It's marked "L". You can also add your own LED—connect it as shown in Figure 4-2.

K indicates the cathode (negative), or shorter lead; A indicates the anode (positive), or longer lead.

Once the LED is connected, you need to tell Arduino what to do. This is done through code, that is, a list of instructions that we give the microcontroller to make it do what we want.

Analogue Input

As you learned in the previous section, Arduino is able to detect whether there is a voltage applied to one of its pins and report it through the *digitalRead()* function. This kind of either/or response is fine in a lot of applications, but the light sensor that we just used is able to tell us not just whether there is light, but also how much light there is. This is the difference between an on/off sensor (which tells us whether something is there) and an analogue sensor, whose value continuously changes. In order to read this type of sensor, we need a different type of pin.

In the lower-right part of the Arduino board, you'll see six pins marked "Analog In"; these are special pins that can tell us not only whether there is a voltage applied to them, but if so, also its value. By using the *analogRead()* function, we can read the voltage applied to one of the pins. This function returns a number between 0 and 1023, which represents voltages between 0 and 5 volts. For example, if there is a voltage of 2.5 V applied to pin number 0, *analogRead(0)* returns 512.

If you now build the circuit that you see in Figure 5-6, using a 10k resistor, and run the code listed in Example 06A, you'll see the onboard LED (you could also insert your own LED into pins 13 and GND as shown in "Blinking an LED" in Chapter 4) blinking at a rate that's dependent upon the amount of light that hits the sensor.