Using a Light Sensor

Read data from an analog light sensor.

Site: iCODE
Course: Machine Science Guides (Arduino Version)
Book: Using a Light Sensor
Printed by: Ivan Rudnicki
Date: Wednesday, July 30, 2014, 03:31 PM

Contents

About Sensors

Because they do not flash or make noise, sensors are less noticeable than the other electronic devices in the world around us, but they are becoming increasingly common. Often, they are used to monitor the environment and trigger actions that human beings may forget or neglect to do--for example, turning a car's windshield wipers on when it starts to rain, or flushing a toilet automatically when a user exits a stall. In other cases, sensors are used to maintain desired conditions when human beings are not around at all--for example, activating the furnace when a house's temperature drops, or turning on a lamp when it gets dark. Figure 1 shows some of these common sensors.

Figure 1. Electronic devices with sensors.

About the Analog Light Sensor

Earlier, you learned that the microcontroller has two essential abilities:

#1 It can *set* the voltage on any of its pins HIGH (5 volts) or LOW (0 volts).
#2 It can *detect* whether the voltage at any pin is HIGH (5 volts) or LOW (0 volts).

In fact, on certain pins, the microcontroller can measure the exact voltage at the pin, as long as the voltage is between 5 volts and 0 volts. These pins are called *analog* ports, because they can measure a range of values, rather than just detecting high or low values. The microcontroller then converts this voltage value into a number ranging from 0 to 1024, using its built-in analog-to-digital converter, or ADC. Figure 2 shows approximately how voltages relate to ADC values for the light sensor included in the kit.

Light Level	Voltage at Pin	ADC Value
Very Bright	0.00 volts	0
Bright	0.75 volts	256
Medium	1.50 volts	512
Dim	2.25 volts	768
Very Dim	3.00 volts	1024

Figure 2. Relationship among light level, voltage, and ADC value.

The light sensor in your kit does not sense visible light, but only radiation in the infrared portion of the electromagnetic spectrum. Infrared radiation is a large component of incandescent and sunlight, but a very small component of fluorescent light, as noted in Figure 3.

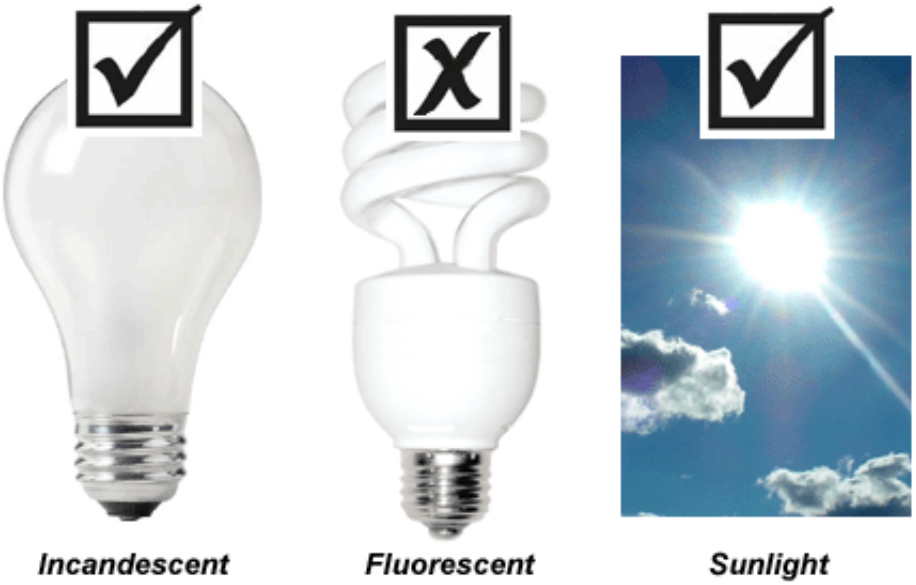

Figure 3. Sources of light detected by the sensor.

Adding the Light Sensor

Connect the light sensor to Port C5, as shown in Figure 4. Like the LED, the light sensor is a polarized device. The ground pin, marked by the flat section on the rim of the sensor's plastic housing, must connect to ground. Note that, in the schematic, Port C5 is also labeled ADC5, indicating that it is an analog port.

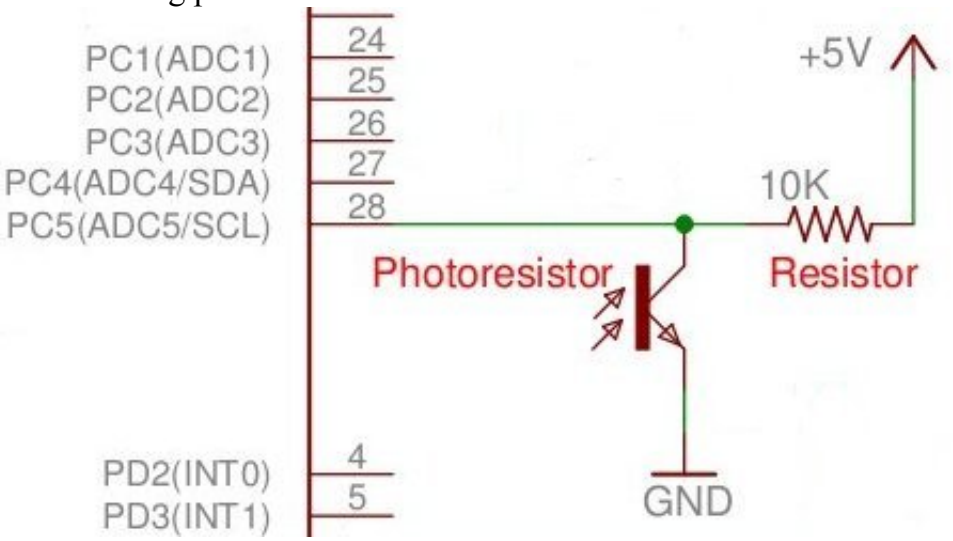

Figure 4. Adding a light sensor.

Controlling the Speaker

With the following code, the light level detected by the sensor determines the frequency of the tones produced by the piezo speaker. After you have downloaded this code to your board, test the sensor with different light sources. You will find that the device is most sensitive to light in the infrared portion of the electromagnetic spectrum. The sensor more easily detects sunlight, incandescent light, and the infrared signals sent from TV remote controls, than it does the light produced by fluorescent bulbs or LEDs.

1. Rename your code file sensorspeaker.c.
2. Modify your code file, as follows:

```
#include <mxapi.h>
#include <sound.h>
#include <adc.h>

int main(void)
{
   int x;                  //Declare a variable called x
   input_pin(PORT_C5);     //Set up Port C5 (sensor) as an input

   adc_init();             //Initialize analog to digital converter
   while(1==1)
   {
      x=adc_read(5);              //Read analog value on Port C5
      tone_out(PORT_D4, x, 40);   //Make x Hertz tone for 40 ms on Port D4
   }
}
```

3. Compile and test your new code.

Controlling an LED

Now imagine that the LED is a light in your house, and you want to turn it on whenever it gets dark in the room. The code in this section shows one way of doing this.

1. Rename your code file sensorled.c.
2. Modify your code file, as follows:

```
#include <mxapi.h>
#include <adc.h>

int main(void)
{
    int x;                //Declare a variable called x
    input_pin(PORT_C5);   //Set up Port C5 (sensor) as an input
    output_pin(PORT_B0); //Set the Port B0 (LED) as an output

    adc_init();                 //Initialize analog to digital converter
    while(1==1)
    {
        x=adc_read(5);          //Read ADC value for Port C5
        if(x>512)               //If it gets dark
        {
            high_pin(PORT_B0); //Turn the LED on
        }
        else                    //If it is bright
        {
            low_pin(PORT_B0);   //Turn the LED off
        }
    }
}
```

3. Compile and test your new code.

Programming Challenge

Program the board so that it makes a continuous tone in the dark, and goes silent when there is light. What would happen if you put a board programmed this way in the refrigerator?