Lesson 12 Infrared Sensors

The Big Idea:

With the addition of a small electronic device called the infrared sensor, we can add to an Arduino™ the ability to detect infrared radiation. This, in a sense, brings eyes to an Arduino.™ Infrared is simply light, although it is light outside the frequency range visible to humans. Future lessons will use infrared two ways: first, to detect objects and, later, to detect and decode messages. This lesson is about detection. Detection of objects is one of the first exercises with the rolling robot of Lesson 15 and decoding messages of Lesson 16.

Background:

Previous lessons introduced the Serial Monitor, the push button, and the potentiometer as means for bringing information into an Arduino™ sketch. This lesson introduces *infrared*, the first of two wireless technologies used in these lessons to connect the Arduino™ to the surrounding world. The second technology, to be addressed later, is radio.

Infrared

Infrared (IR) is a particular frequency range of electromagnetic radiation, similar to that of visible light. IR, in fact, sits just before the red end of the visible light.

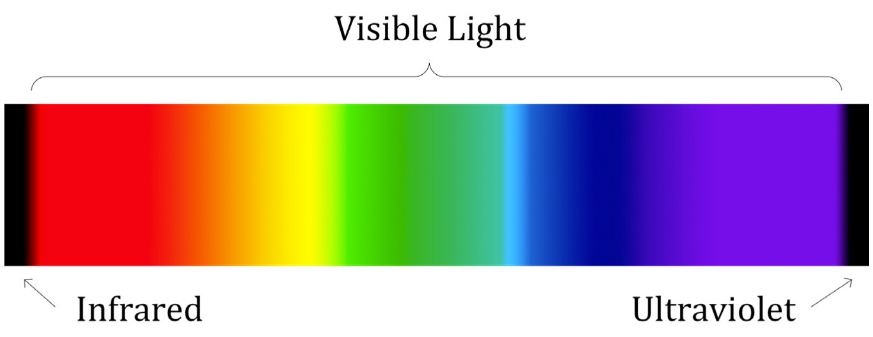


Figure 12-1. Visible light spectrum

Infrared is ubiquitous. It is present in sunshine and is generally associated with objects radiating heat, which is everything from a human body to freshly baked bread. And it is by far the most commonly used technology for remote control of electronic devices, including home entertainment systems.

This presents a problem. How is it that the television receiver can respond instantly to an infrared message from a remote control but is somehow able to ignore a warm chocolate chip cookie?

The answer is that infrared radiation as it is used in common remote controls and will be used in these lessons is *modulated*. In other words, when it is on it is actually being turned on and off 38,000 times per second. The infrared receivers used in this lesson are specially designed to detect only infrared that is flashing on and off at this frequency. The infrared from the cookie, which is not modulated, is ignored.

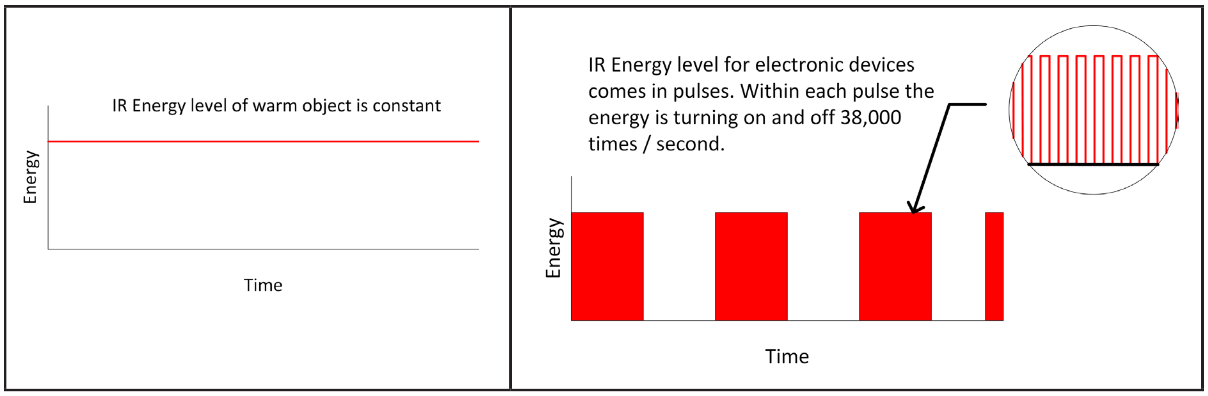


Figure 12-2. Unmodulated vs. modulated infrared radiation

This lesson deals only with detection of infrared energy being transmitted by an electronic device. But note that actual messages can be encoded into the transmitted infrared by changing the lengths of the pulses and the times between them. This is how a television remote is able to perform specific tasks. Message encoding will be used in later lessons to remotely control a rolling robot and play a sort of musical instrument.

The rate at which some signal is turned on and off is called the *frequency*. One on and off pair is called a *cycle*. Frequency is the number of cycles per some unit in time, in this case cycles per second. The unit for cycles per second is the *Hertz*, named after Heinrich Hertz, a German physicist generally credited with proving the existence of electromagnetic waves. The abbreviation for the Hertz is Hz. For convenience, frequency is usually referred to in thousands of cycles per second, for which the abbreviation is kHz.

Table 12-1. Vocabulary

Term	Description
cycle	A single pulse followed by a period of no pulse just before another pulse.
frequency	The number of times the pulse is turned on and off per second. Infrared used for control of electronic devices is typically modulated at a frequency of 38,000 cycles per second.
Hertz	The unit of measure for frequency. One Hertz is equal to one cycle per second. The abbreviation is Hz.

Term	Description
infrared	The frequency of electromagnetic radiation just below light visible to humans. The first visible color is red, hence the name infrared (infra means below).
infrared sensor	Electronic component with an output pin that is normally at +5 volts. When this device detects modulated infrared, this output voltage drops to zero.
modulate	The process of turning a pulse on and off very rapidly when that pulse is in the `ON`, or `HIGH`, state.

Description:

The *infrared sensor* is a small electronic component with three wires. One is connected to the Arduino™ +5, one to ground (GND), and the third is the device output, typically connected to an Arduino™ digital pin configured for `INPUT` mode.

The output pin is `HIGH` when no modulated infrared is detected (or sensed). When the receiver detects infrared modulated at 38 kHz, the voltage on its output pin drops to `LOW`.

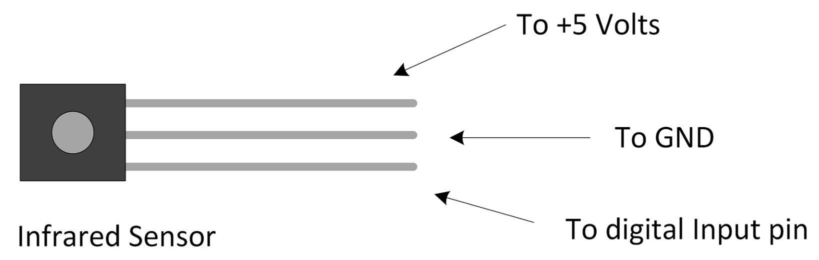


Table 12-2. Pin 1 status and meaning

Pin 1 status	Meaning
`HIGH`	No modulated infrared detected.
`LOW`	Modulated infrared is detected.

Caution	Generally speaking, pin 1 of the IR sensor may be connected to any Arduino™ Uno digital port configured for `INPUT`. But the Uno also has a built-in LED and current-limiting resistor connected to pin 13. As a result, special care must be taken before using it for `INPUT`. See http://arduino.cc/en/Tutorial/DigitalPins for specifics. These lessons do not attempt to use this pin for input.

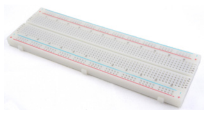
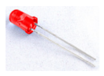
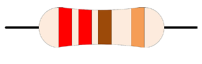
This lesson does not decode incoming infrared messages. It merely detects them. Refer to Lesson 15 for decoding of messages.

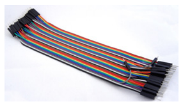
Goals:

By the end of this lesson readers will:

1. Be able to identify an infrared detector and the individual pins by pin number.
2. Know how an infrared detector is wired for power and how to feed its output to an Arduino™ digital pin.
3. Know how to configure an Arduino™ digital pin for reading the infrared sensor signal.
4. Write a sketch that detects and responds to an infrared signal from a remote control.

Materials:

Quan-tity	Part	Image	Notes	Catalog Number
1	Arduino™ Uno		Single-board computer. This board is delicate and should be handled with care. When you are not using it, keep it in a box or plastic bag.	3102
1	USB Cable		This is a standard USB adapter cable with a flat connector on one end and a square connector on the other.	2301
1	Computer with at least one USB port and access to the Arduino™ website, http://www.arduino.cc.	---	The operating system of this computer must be Windows, Macintosh OS/X, or Linux.	---
1	Bread-board		Used for prototyping.	3104
2	Light-emitting diodes (LEDs)		Single color, about 0.02 amps rated current, diffused.	1301
2	Resistors, 220 ohm		1/4 watt, 5% tolerance, red-red-brown-gold.	0102
1	Television remote control		Any Sony-compatible remote	---

Quantity	Part	Image	Notes	Catalog Number
1	Infrared sensor		3-pin, 38kHz.	1302
As req'd	Jumper wires		Used with bread-boards for wiring the components.	3105

Procedure:

1. Construct the circuit as shown in Figure 12-4.

Using a bread-board, wire the infrared sensor, LEDs, and current-limiting resistors.

Figure 12-4 is a schematic. The IR sensor's pin 1 is NOT the center pin. It just sort of looks that way because a schematic is drawn for a simple diagram of the electronic components.

2. Connect the Arduino™ to the computer via the USB cable and start the Arduino™ IDE.

3. Create a new Arduino™ sketch. Name it `Lesson12IRSensor`.

4. Enter the header comments and define the pins to be used

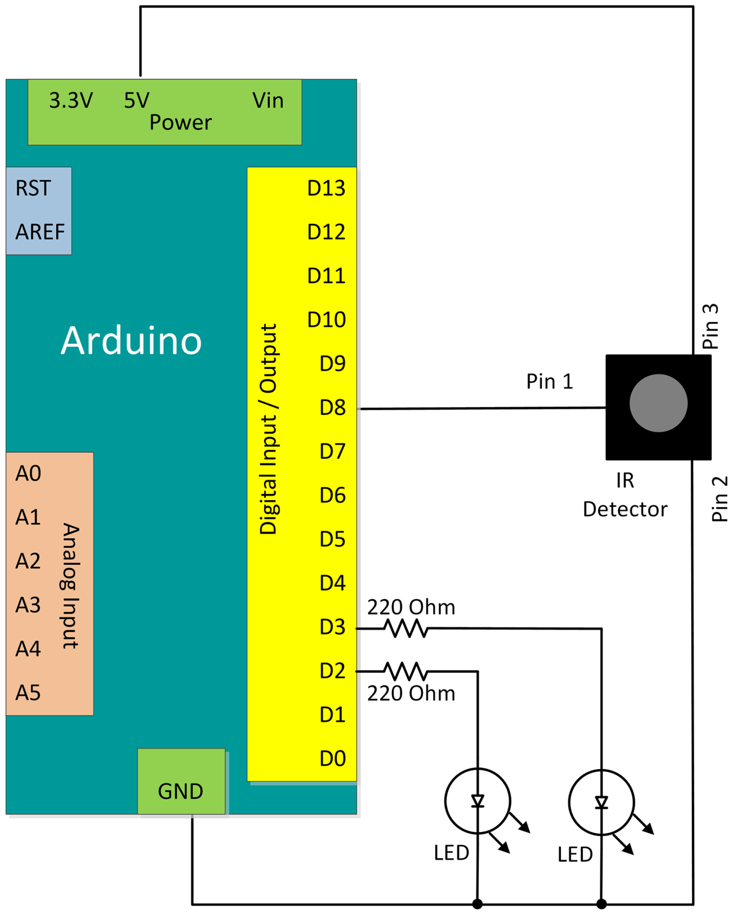


Figure 12-4. Schematic Diagram

Snippet 12-1.

```
/* Lesson12IRSensor
   by W. P. Osborne
   6/30/15
*/

#define pinLED1 2
#define pinLED2 3
#define pinIRSensor 8
...
```

5. Add the `setup()` method and use it to initialize the three digital pins. Notice that the pins being used for the LEDs must be initialized as `OUTPUT`, while the pin for sensing the infrared must be `INPUT`.

Snippet 12-2.

```
...
void setup(){
    pinMode(pinLED1, OUTPUT);
    pinMode(pinLED2, OUTPUT);
    pinMode(pinIRSensor, INPUT);
}
...
```

6. Finally, add the `loop()` method. Use an `if-else` pair to either light or put out the first LED depending on if the sensor detects infrared. The LED should light when infrared is present.

Snippet 12-3.

```
...
void loop(){
    if( digitalRead(pinIRSensor) == LOW){
        // infrared is detected. Light the LED
        digitalWrite(pinLED1, HIGH);
    } else {
        digitalWrite(pinLED1, LOW);
    }
}
```

7. Save the sketch then upload it to the Arduino.™ Neither LED should be lit.

8. Point the remote control at the infrared sensor, then press a button. The LED should flash.

Notice that the LED isn't on all the time. The remote control is sending the command message associated with the button being pushed. This message is composed of pulses. The LED is lighting in response to these pulses.

Experiment with the remote control to see how far away it can get and still be detected. Determine if one side of the sensor is more sensitive than the other.

Exercises:

Exercise 12-1. No infrared detected

Modify the `Lesson12IRSensor` sketch to have the second LED light when no infrared is detected but go out when infrared is detected. The two LEDs, then, will be exact opposites. When one is on the other is off. Save the sketch as `Lesson12Exercise1`.

Note: This second diode will be on most of the time.

Exercise 12-2. Smooth infrared detected

Modify the sketch from Exercise 12-1 to keep the IR detected LED on for one second each time infrared is detected. Ask yourself if this this makes the sensor easier to use. That is,

- Does the sensor seems to respond immediately when infrared is detected?
- Does the detection LED go out in a timely manner when infrared is no longer detected?

Exercise 12-3. Distance front and back

Infrared sensors have a front, where the raised area of plastic is found, and a back, which is flat. Measure how far the remote control can be taken from the sensor and still have detection occur reliably. Record these differences:

Distance from front: ____________ meters / feet

Complete listing 12.1. `Lesson12IRSensor`

```
/* Lesson12IRSensor
   by W. P. Osborne
   6/30/15
*/

#define pinLED1 2
#define pinLED2 3
#define pinIRSensor 8

void setup(){
    pinMode(pinLED1, OUTPUT);
    pinMode(pinLED2, OUTPUT);
    pinMode(pinIRSensor, INPUT);
}

void loop(){
    if( digitalRead(pinIRSensor) == LOW){
        // infrared is detected. Light the LED
        digitalWrite(pinLED1, HIGH);
        digitalWrite(pinLED2, LOW);
    } else {
        digitalWrite(pinLED1, LOW);
        digitalWrite(pinLED2, HIGH);
    }
}
```