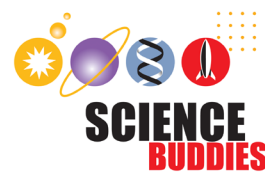


Build an Arduino Self-Driving Car

Lesson 6: Using the Infrared Sensors



Estimated time: 60 minutes

Introduction

In this lesson students will learn to use an infrared (IR) light sensor to detect nearby objects or differentiate between light and dark surfaces. They will learn how two sensors can be used to make a lane-following autonomous car.

Learning Objectives

- Use an IR emitter/detector (sensor) with an Arduino
- Connect a second sensor based on instructions for connecting the first sensor
- Understand how two sensors can be used for vehicle navigation

Preparation

- Try this activity yourself before you try it with students. You can keep a working circuit assembled for a demonstration.
- Have an assortment of objects available for students to test with the sensors. Make sure you have objects with different colors and surface textures (matte vs. glossy).

Materials

In addition to the materials from previous lessons, each group will need:

- Infrared sensors (2)
- Male-female jumper wires (8)
- Assorted male-male jumper wires
- LEDs (2)
- 220Ω resistors (2)

Procedure

Walk your students through the following steps or have them watch the video and follow along.

Step	Timestamp	Description
1	0:00	Demonstrate the working circuit for students.
2	0:40	Have students look closely at one of the sensors. Explain how the sensor operates and how to identify the different pins using the notch in one corner. You can also have students look at the sensor's datasheet.
3	4:00	Connect male-female jumper wires to the sensor's pins and connect it to the Arduino along with an LED (see Appendix 6A).
4	7:04	Enter the example code to control the LED using the sensor (see Appendix 6A). Go through the commented code and discuss how it works.
5	9:08	Upload the code to your Arduino and test your sensor by aiming it at different objects from different distances. Note how the sensor's output depends on both the infrared reflectivity of the object <i>and</i> the distance from the object.
6	10:59	Connect a second sensor and LED to the Arduino and modify the code to control the second LED with the second sensor (see Appendix 6B , but let students try this on their own first). Discuss how two sensors can be used for a line-following robot or autonomous car.

Assessment

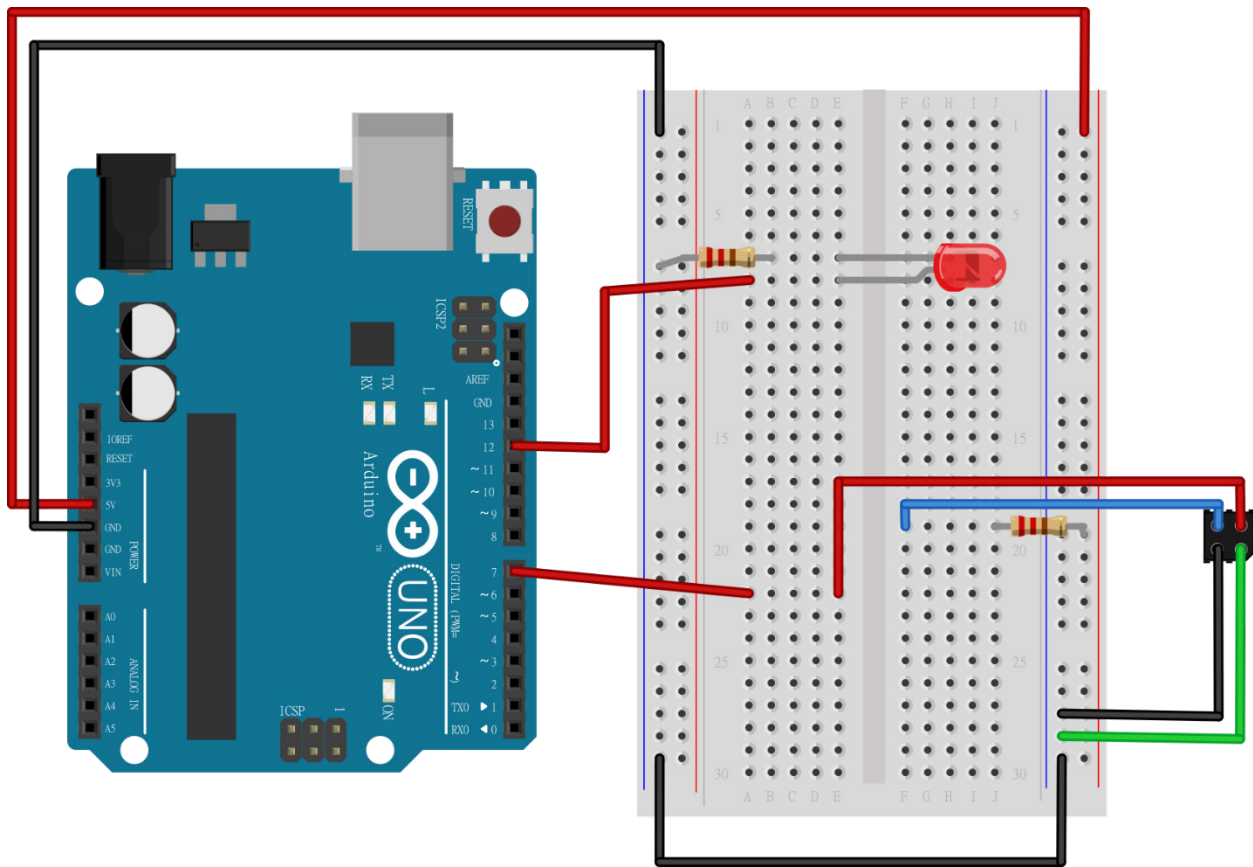
As you walk around the class:

- Confirm that students can get the sensor and example code working.
- Observe how students test different objects in front of the sensor.
- Make sure students can successfully add a second sensor and LED to the circuit.

Appendix 6A: Breadboard Diagram and Code for One Sensor

The following diagram shows a single IR sensor and LED connected to the Arduino.

Note: this diagram omits the H-bridge for clarity. If you already have the H-bridge circuit assembled, you may need to connect your sensor to different breadboard rows.



The following code will light up the LED when the sensor detects a surface that reflects sufficient IR light.

```
// declare constants for pins
const int sensorPin = 7;
const int ledPin = 12;

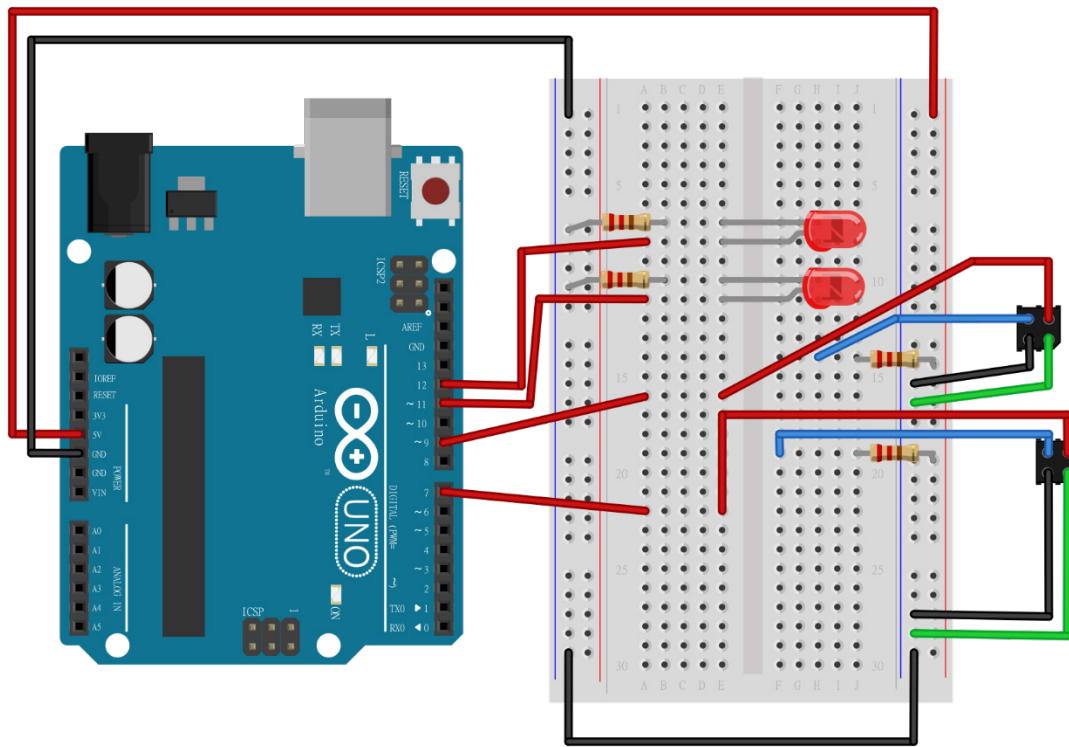
// declare variable for input pin state
int pinState;

void setup() { // setup code that only runs once
  // set sensor pin as input with internal pullup resistor
  // this way you do not need an external resistor
  pinMode(sensorPin, INPUT_PULLUP);
  // set LED pin as output
  pinMode(ledPin, OUTPUT);
}

void loop() { // code that loops forever
  pinState = digitalRead(sensorPin); // read the sensor pin
  if(pinState == HIGH){ // no IR light detected, sensor sees
                        // dark surface
    digitalWrite(ledPin, LOW); // turn the LED off
  }
  else{ // IR light detected, sensor sees light surface
    digitalWrite(ledPin, HIGH); // turn the LED on
  }
}
```

Appendix 6B: Breadboard Diagram and Code for Two Sensors

The following diagram shows two IR sensors and two LEDs connected to the Arduino.



The following code will control the two LEDs separately with the two sensors.

```
// declare constants for pins
const int sensor1Pin = 7;
const int sensor2Pin = 9;
const int led1Pin = 12;
const int led2Pin = 11;

// declare variable for input pin states
int sensor1State;
int sensor2State;

void setup() { // setup code that only runs once
  // set sensor pins as inputs with internal pullup resistors
  // this way you do not need a external resistors
  pinMode(sensor1Pin, INPUT_PULLUP);
  pinMode(sensor2Pin, INPUT_PULLUP);
  // set LED pins as output
  pinMode(led1Pin,OUTPUT);
  pinMode(led2Pin,OUTPUT);
}

void loop() { // code that loops forever
  // read the sensor pins
  sensor1State = digitalRead(sensor1Pin);
  sensor2State = digitalRead(sensor2Pin);

  // control the LEDs
  if(sensor1State == HIGH){ // no IR light detected, sensor sees
                           // dark surface
    digitalWrite(led1Pin,LOW); // turn the LED off
  }
  else{ // IR light detected, sensor sees light surface
    digitalWrite(led1Pin,HIGH); // turn the LED on
  }
}
```

```
if(sensor2State == HIGH){ // no IR light detected, sensor sees
                          // dark surface
    digitalWrite(led2Pin,LOW); // turn the LED off
}
else{ // IR light detected, sensor sees light surface
    digitalWrite(led2Pin,HIGH); // turn the LED on
}
}
```