IOT LAB EXPERIMENTS

Sensors Actuators

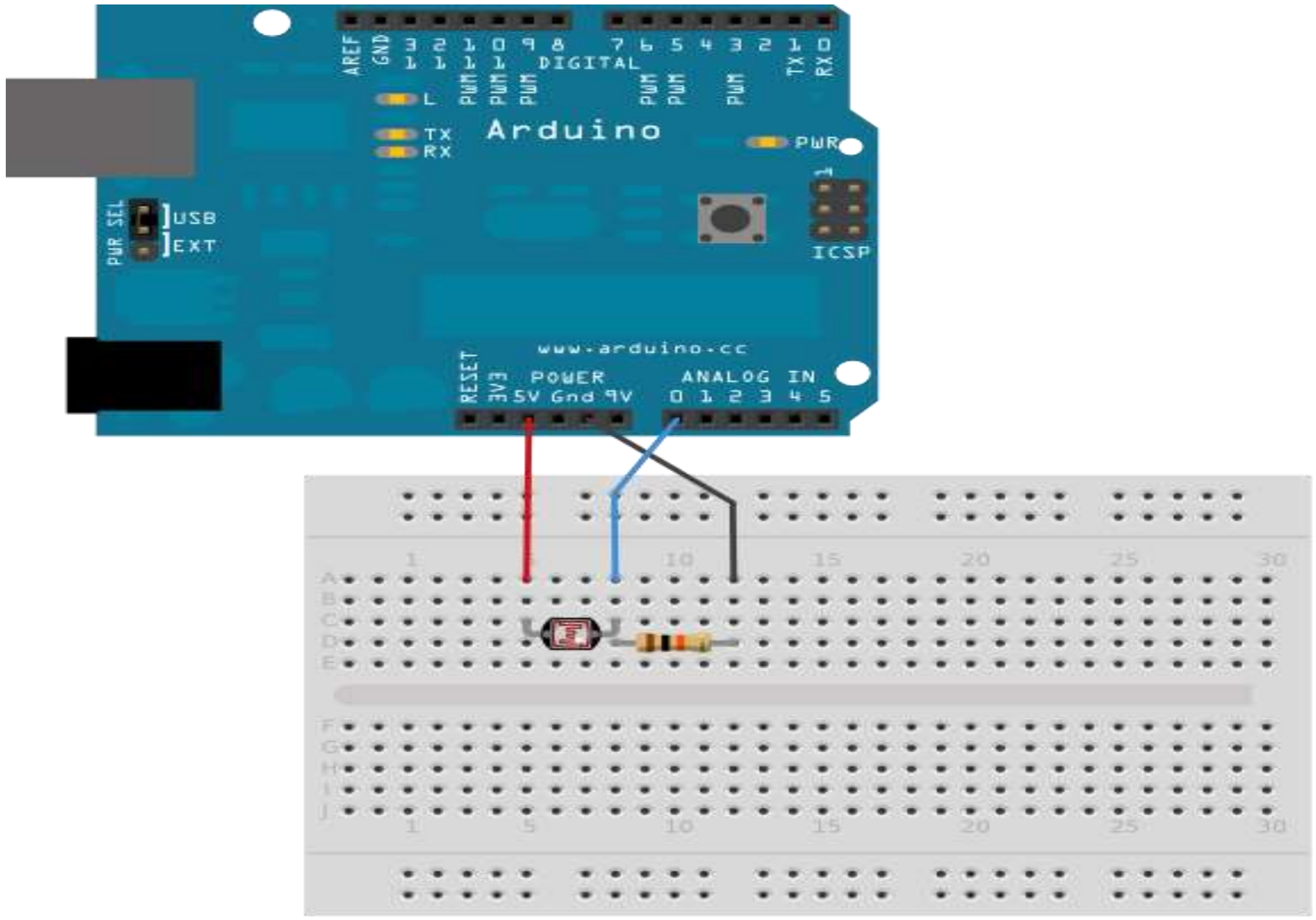
Department of Computer Science and Engineering, BMS College of Engineering

LDR(Light Dependent Resistor)

- A photoresistor (or light-dependent resistor, **LDR**, or photocell) is a **light-controlled variable resistor**.
- The **resistance of a photoresistor decreases with increasing incident light intensity**
- In other words, it exhibits photoconductivity.

Nightlight Simulation(Photo Sensor)

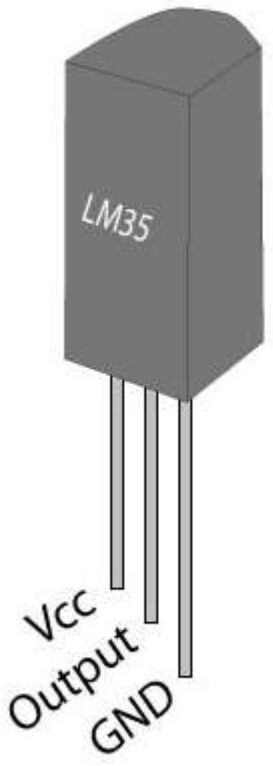
Nightlight Simulation(Photo Sensor)



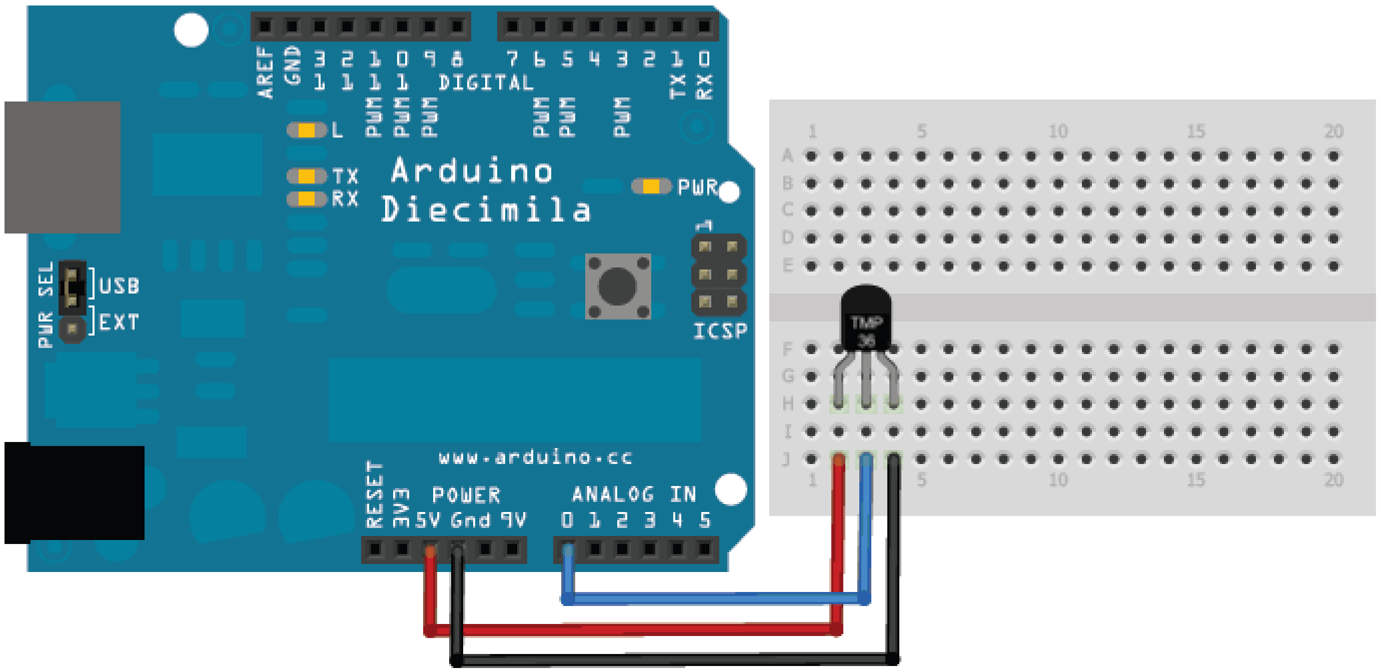
Code…

```
int light_sensitivity = 500;    //This is the approx value of light surrounding your LDR
 void setup()
  {
    Serial.begin(9600);          //start the serial monitor with 9600 buad
    pinMode(11, OUTPUT);
}
 void loop()
  {
    int LDRValue = analogRead(A0);      //reads the ldr's value through LDR
    Serial.println(LDRValue);       //prints the LDR values to serial monitor
    delay(50);        //This is the speed by which LDR sends value to arduino
     if (LDRValue < light_sensitivity)
       digitalWrite(11, HIGH);
    else
       digitalWrite(11, LOW);
      delay(1000);
  }
```

Temperature Sensor



Temperature Sensor(1)



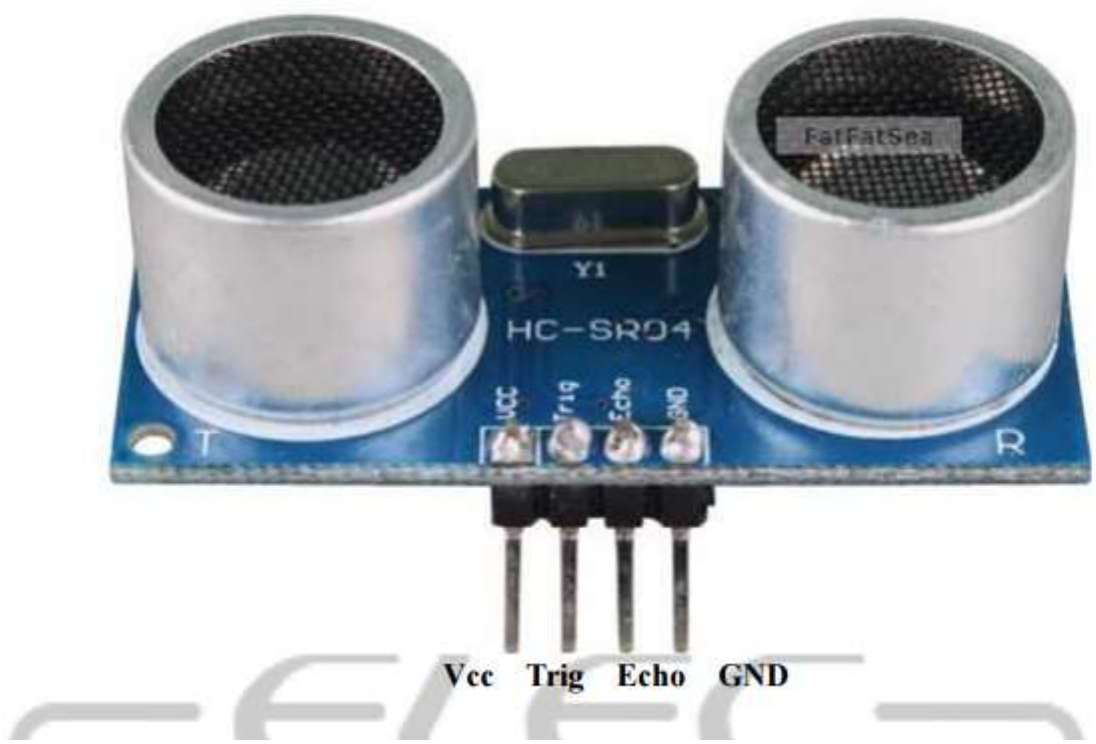
```
int outputpin= 0;
//this sets the ground pin to LOW and the input voltage pin to high
void setup()
{
Serial.begin(9600);
}
void loop()
{
int rawvoltage= analogRead(outputpin);
float millivolts= (rawvoltage/1024.0) * 5000;
float celsius= millivolts/10;
Serial.print(celsius);
Serial.print(" degrees Celsius, ");
Serial.print((celsius * 9)/5 + 32);
Serial.println(" degrees Fahrenheit");
delay(1000);
}
```

Temperature Sensor(3)

```
temp = (5.0 * analogRead(tempPin) * 100.0) / 1024;
```

The original equation came from taking the reading, finding what percentage of the range (1024) it is, multiplying that by the range itself(aRef, or 5000 mV), and dividing by ten (10 mV per degree Celcius(Linear temperature slope), according to the datasheet.

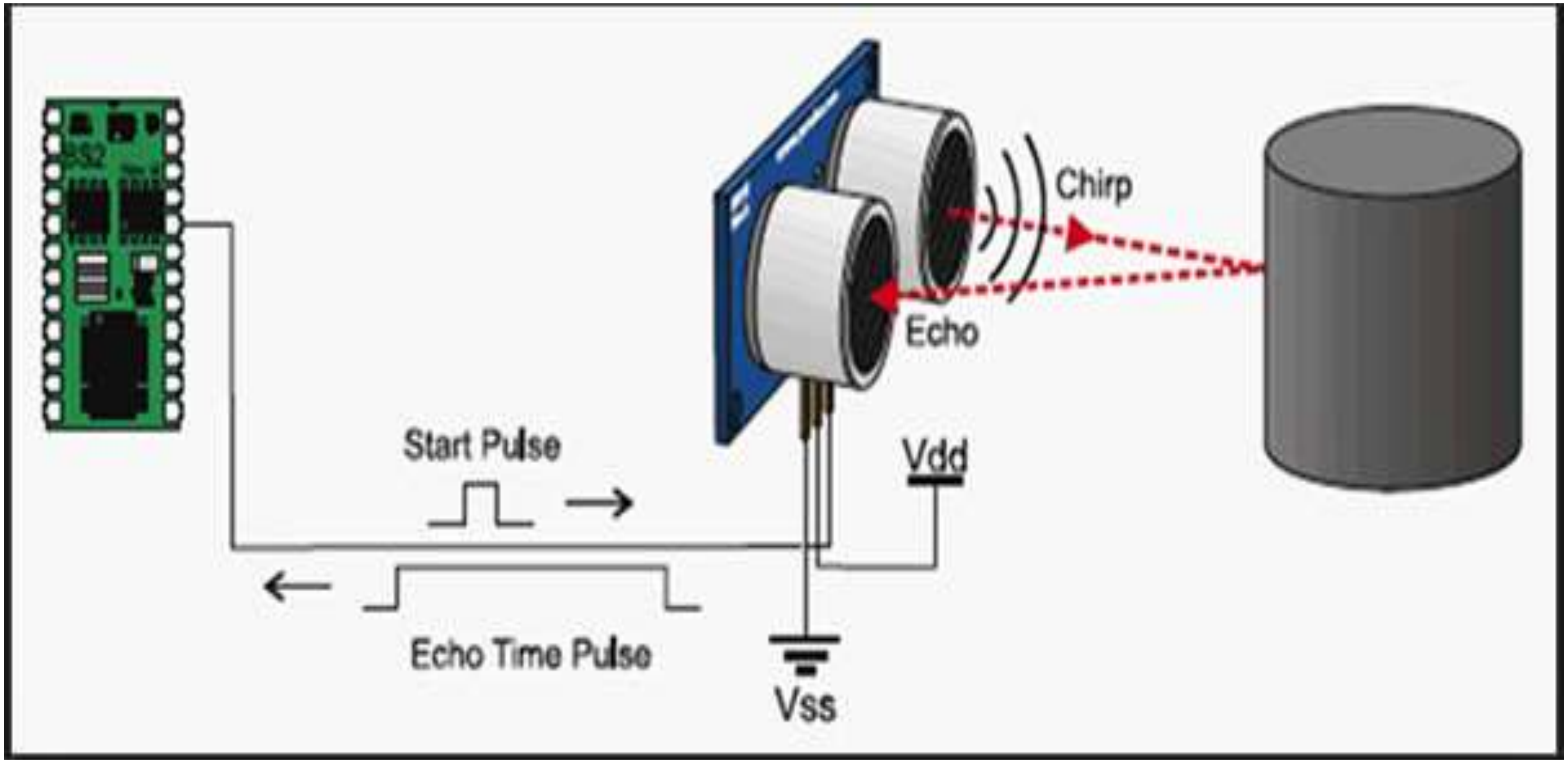
Ultrasound Sensor(1)



Ultrasound Sensor(2)

- Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function
- The ranging accuracy can reach to 3mm.
- The modules includes ultrasonic transmitters, receiver and control circuit.
- VCC: +5VDC
- Trig : Trigger (INPUT)
- Echo: Echo (OUTPUT)
- GND: GND

Working Principle(1)



Working Principle(2)

The modules includes ultrasonic transmitters, receiver and control circuit.

The basic principle of work:

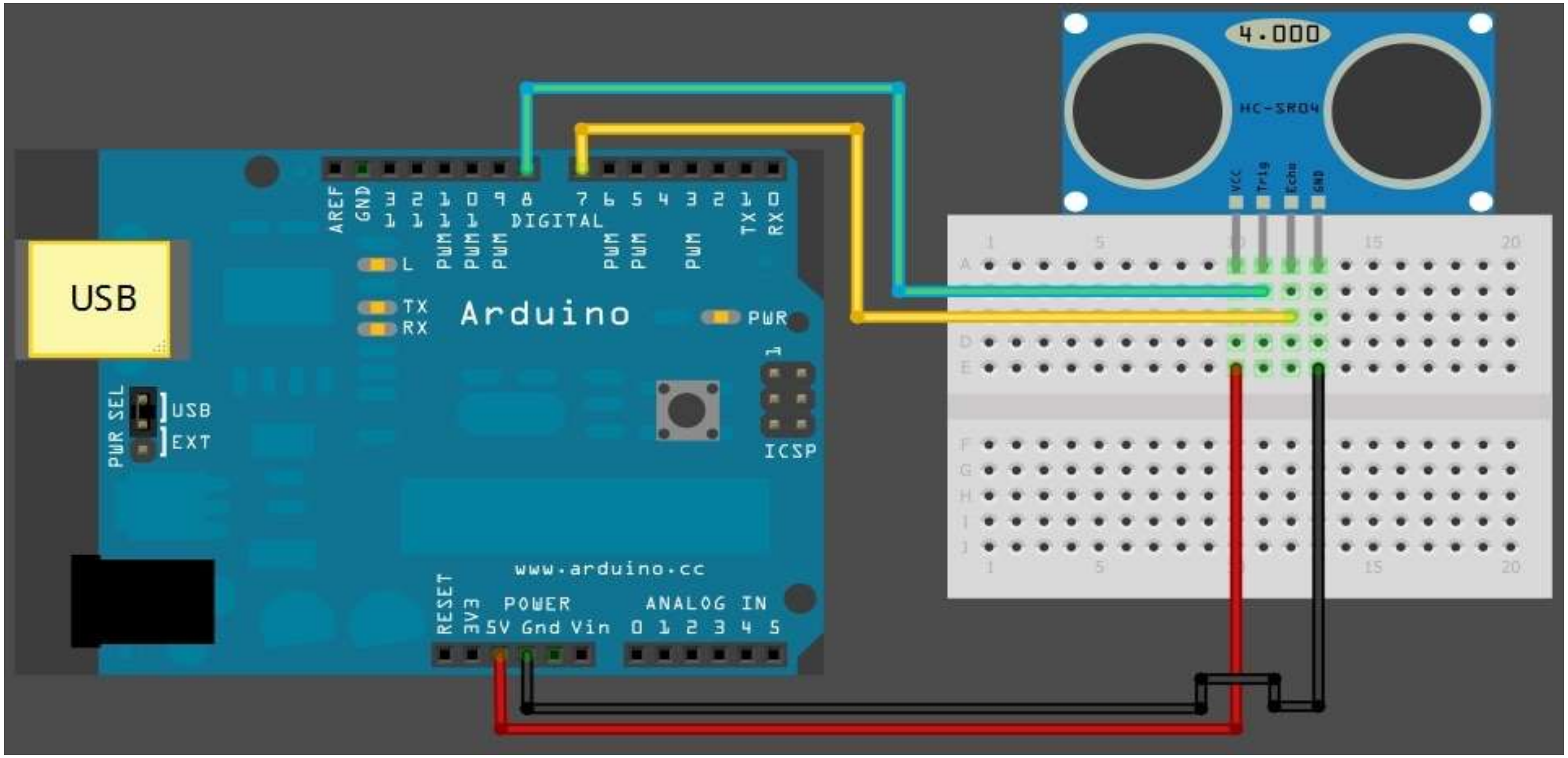
(1) Using IO trigger for at least 10us high level signal,

(2) The Module automatically sends eight 40 kHz pulse and detect whether there is a pulse signal back.

(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Distance Calculation with Ultrasound



Code for printing distance of an obstacle in serial monitor

```
const int trigPin = 2;
const int echoPin = 4;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
```

```
void loop()
{
pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  inches = microsecondsToInches(duration); // convert time  to  distance
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);     Serial.print("in, ");
  Serial.print(cm);          Serial.print("cm");
  Serial.println();
    delay(1000);
}
```

```
long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}

long microsecondsToInches(long microseconds)
{
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second).  This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-
v1.3.pdf
  return microseconds / 74 / 2;
}
```

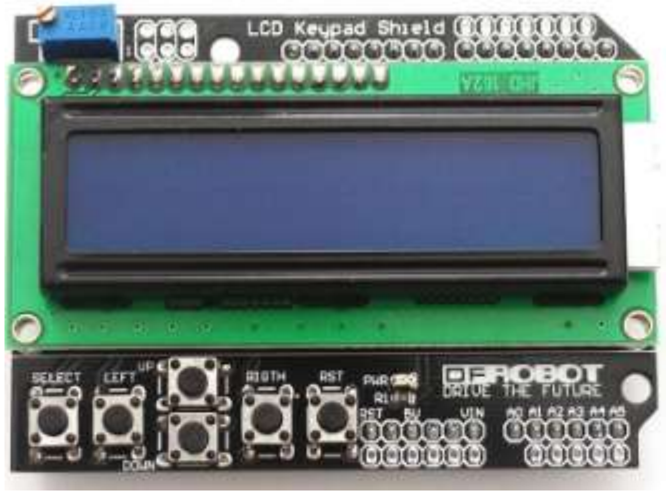
Reverse Parking Sensor

```
#include <NewPing.h>

int tonepin = 13;
//Attach positive leg of buzzer at 13 and negative leg to GND
-------------------------------------------------------
 if(cm >0 && cm <10)
{
Serial.print("Less Distance");
tone(tonepin,2000,1000); //tone(pin, frequency(hertz), duration(millisecon
}
if (cm > 10)
{
noTone(8);
}
```

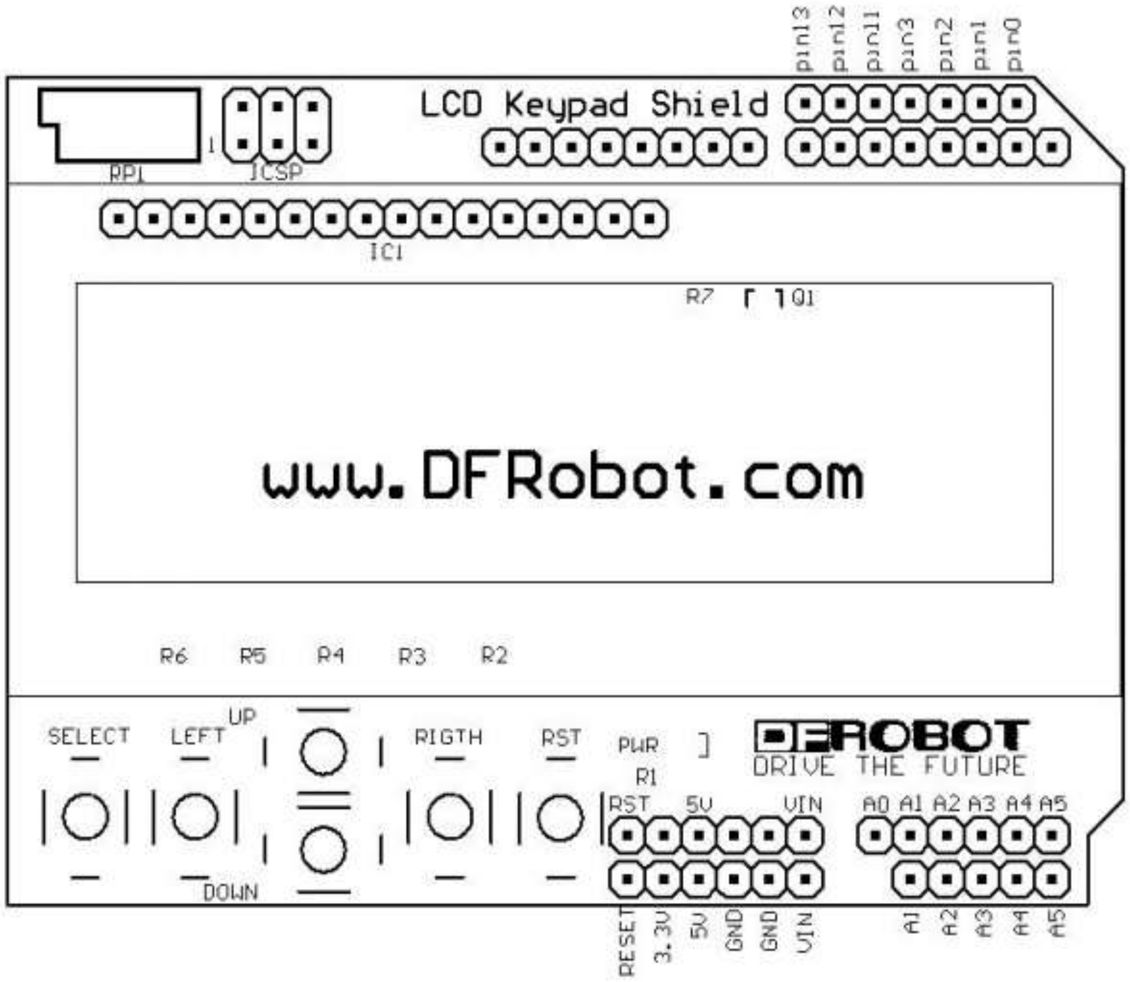
Procedure for including Newping.h

- **Instructions:**
- **Download NewPing library from here: Download NewPing Library**
- Put the "NewPing" folder in "libraries\".
- In the Arduino IDE, create a new sketch (or open one) and select from the menubar "Sktech->Import Library->NewPing".
- **Comment it all out like this in NewPing.cpp:**
-
```
/*
#if defined (__AVR_ATmega32U4__) // Use Timer4 for ATmega32U4
(Teensy/Leonardo).
ISR(TIMER4_OVF_vect) {
#else
ISR(TIMER2_COMPA_vect) {
#endif
	if(intFunc) intFunc(); // If wrapped function is set, call it.
}*/
```

Distance display(LCD)-Ultrasound sensor(1)

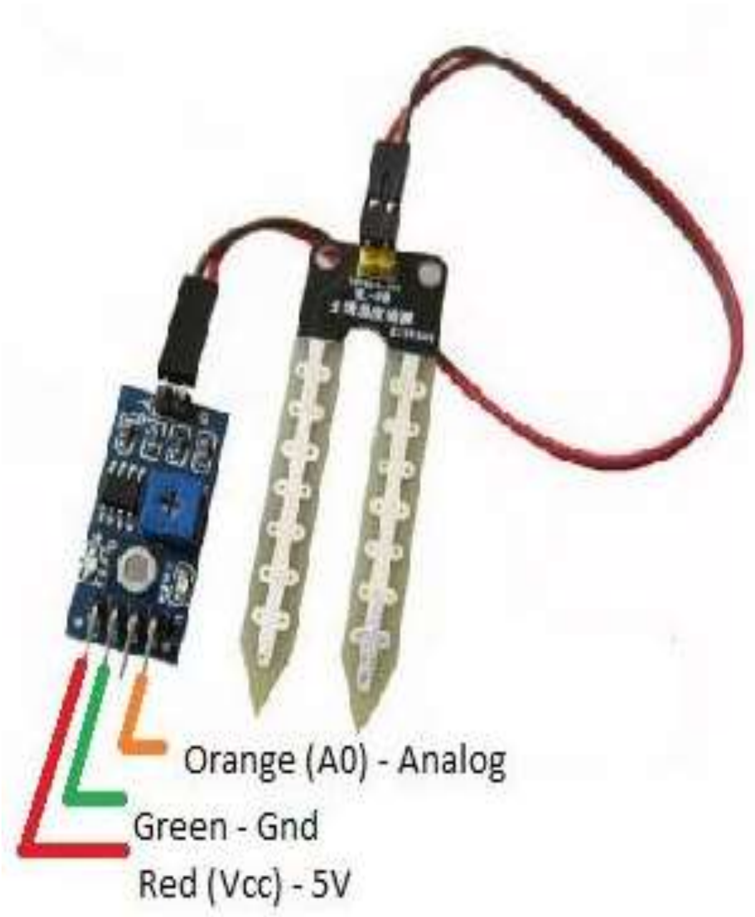


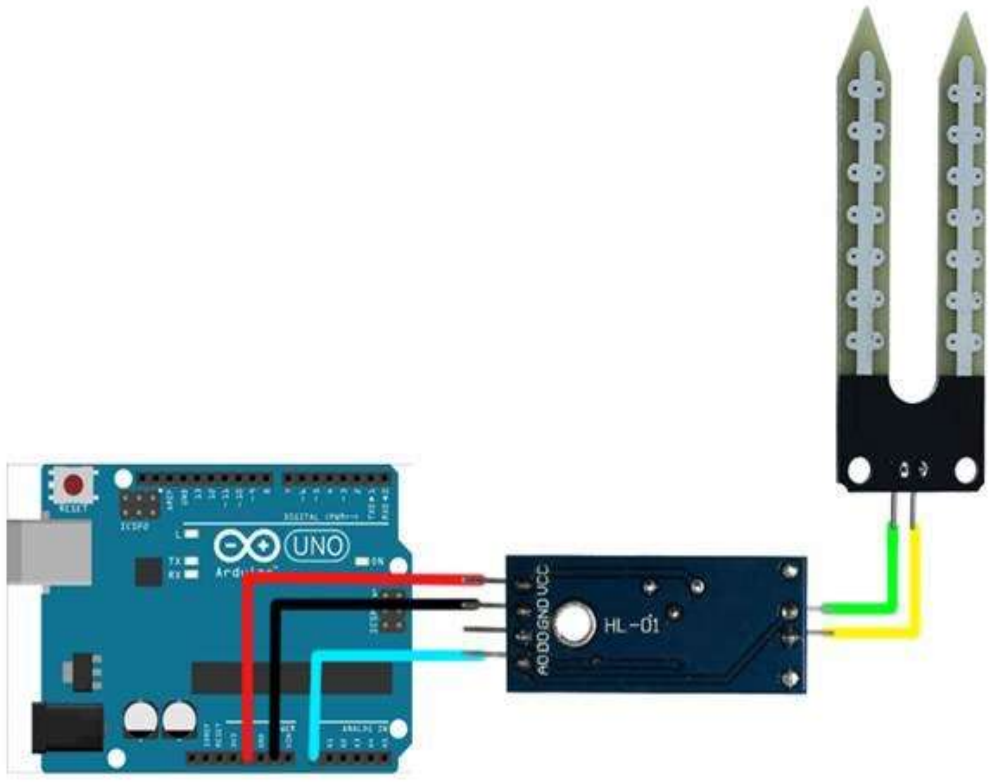
Distance display(LCD)-Ultrasound sensor(2)



Ultrasound	Arduino
----------	--------
Echo	2
Trigger	3
Vcc	5 V
Gnd	Gnd

Moisture Sensor



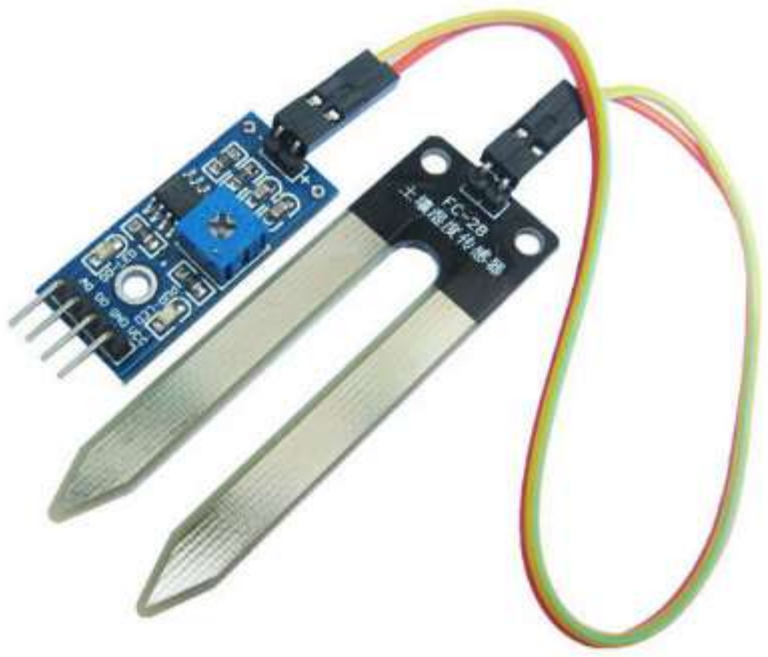


```
int Dpin0=5; //Digital Pin 5 connected to D0
int Apin0=1; //Analog Pin 1 connected to A0
int Apinval;

void setup()
{
pinMode(Dpin0, INPUT);
Serial.begin(9600);
}

void loop()
{
Serial.print("D0 Value:");
Serial.println(digitalRead(Dpin0)); // Read Digital Pin
Serial.print("A0 Value:");
Apinval=analogRead(Apin0); // Read Analog Pin for Moisture Sensor Value
Serial.println(Apinval);
delay(5000);
}
```

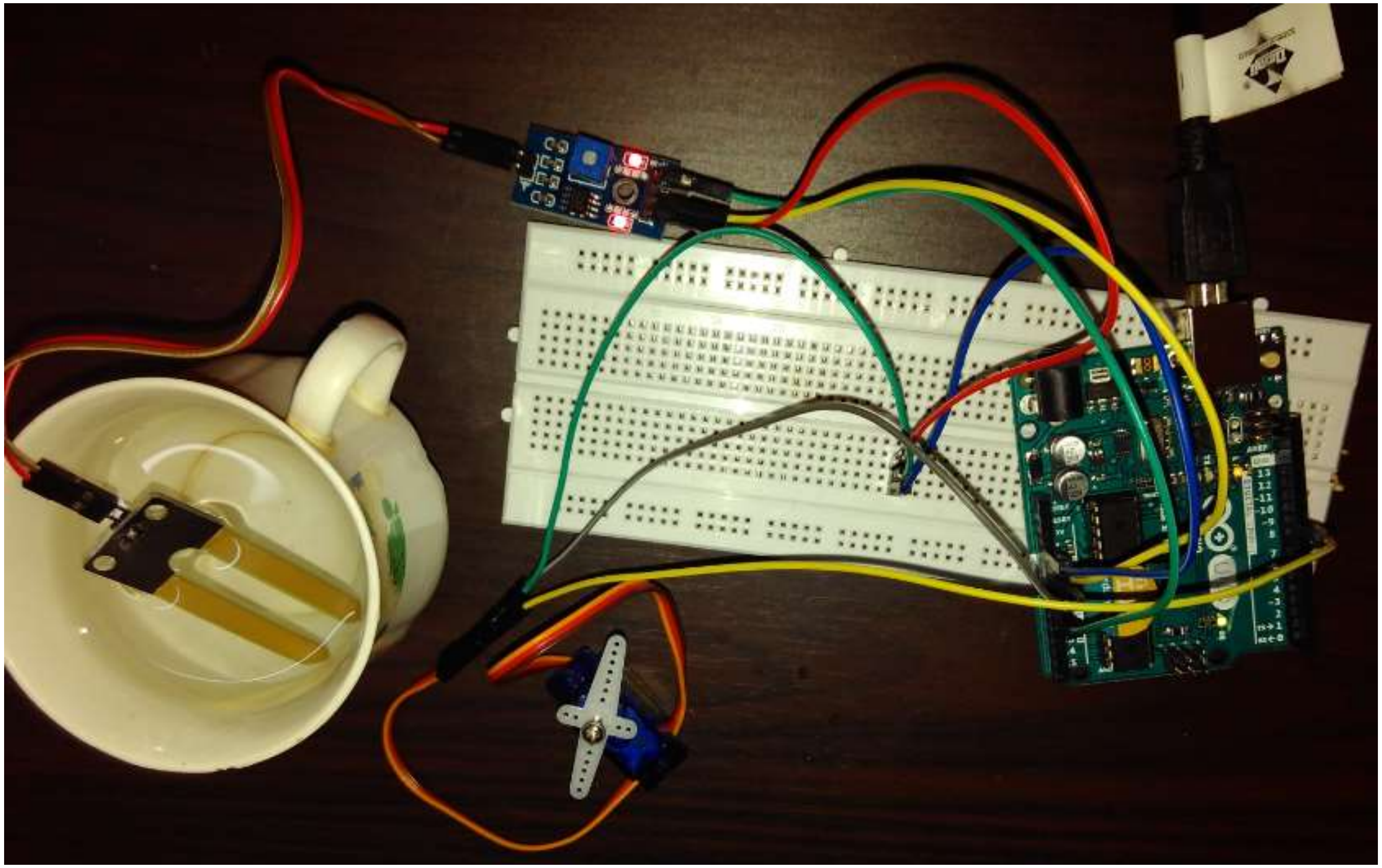
Servo Motor

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Automatic Irrigation Control Simulation(1)



Automatic Irrigation Control Simulation(2)



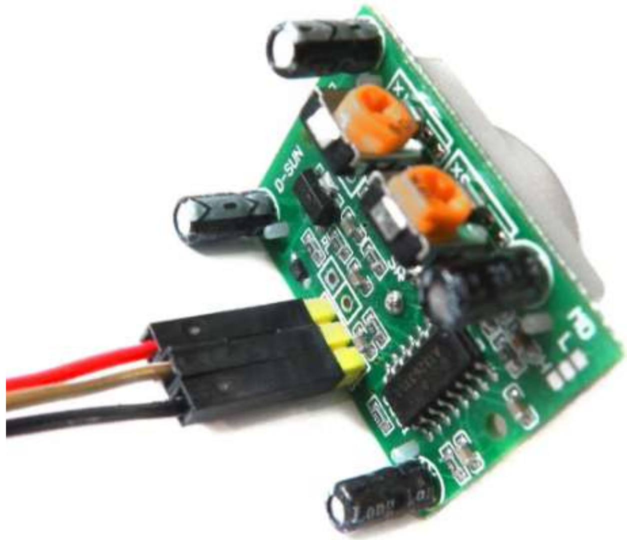
Automatic Irrigation Control Simulation(3)

- Moisture sensor VCC to Arduino 5V
- Moisture sensor GND to Arduino GND
- Moisture sensor A0 to Arduino A0

- Servo motor VCC to Arduino 5V
- Servo motor GND to Arduino GND
- Servo Motor Signal to Arduino digital pin 9

- Write the code to rotate servo motor when there is no moisture

PIR(Passive Infrared Sensor)



PIR Sensor(1)

- It measures infrared (IR) light radiating from objects in its field of view.
- All objects with a temperature above absolute zero emit heat energy in the form of radiation. Usually this radiation isn't visible to the human eye because it radiates at infrared wavelengths
- The term ***passive*** in this instance refers to the fact that PIR devices do not generate or radiate any energy for detection purposes. They work entirely by detecting the infrared radiation emitted or reflected from an object.

PIR Sensor(2)

- An individual PIR sensor detects changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor.
- When an object, such as a human, passes in front of the background, such as a wall, the temperature at that point in the sensor's field of view will rise from room temperature to body temperature, and then back again. The sensor converts the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.

PIR Sensor code(1)

```
//the time we give the sensor to calibrate (10-60 secs according to the
datasheet)
int calibrationTime = 30;

//the time when the sensor outputs a low impulse
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 5000;
boolean lockLow = true;  // variable used to calculate start time of the motion
boolean takeLowTime;  //variable used to calculate stop time of the motion

int pirPin = 3;    //the digital pin connected to the PIR sensor's output
int ledPin = 13;
```

PIR Sensor code(2)

```
void setup(){     //setup
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);

  Serial.print("calibrating sensor ");                    //give the sensor some
time to calibrate
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      delay(1000);
      }
    Serial.println(" done");
    Serial.println("SENSOR ACTIVE");
    delay(50);
  }
```

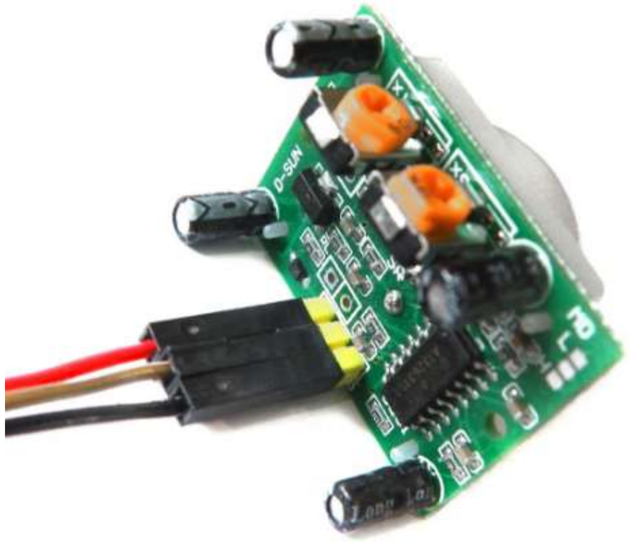
PIR Sensor code (3)

```
void loop(){                    //loop
if(digitalRead(pirPin) == HIGH){
digitalWrite(ledPin, HIGH);
     if(lockLow){
 //makes sure we wait for a transition to LOW before any further output is
made:
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(50);
      }
      takeLowTime = true;
     }
```

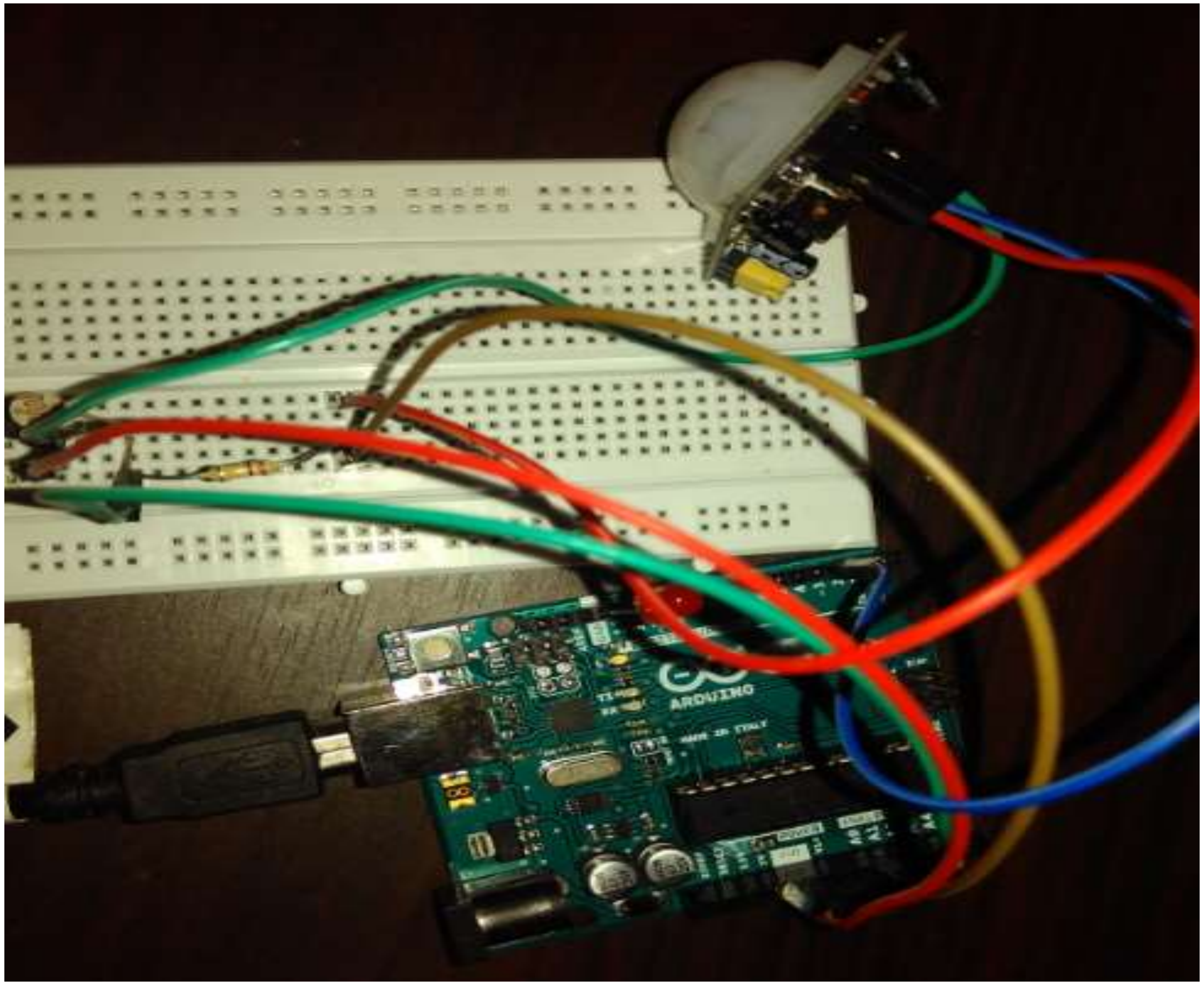
PIR Sensor code(4)

```
if(digitalRead(pirPin) == LOW){
  digitalWrite(ledPin, LOW);  //the led visualizes the sensors output pin state

  if(takeLowTime){
   lowIn = millis();          //save the time of the transition from high to LOW
   takeLowTime = false;       //make sure this is only done at the start of a
LOW phase
   }
  //if the sensor is low for more than the given pause,
  //we assume that no more motion is going to happen
  if(!lockLow && millis() - lowIn > pause){
     //makes sure this block of code is only executed again after
     //a new motion sequence has been detected
     lockLow = true;
     Serial.print("motion ended at ");      //output
     Serial.print((millis() - pause)/1000);
     Serial.println(" sec");
     delay(50);
     }
  }
}
```

Nightlight Simulation on Human Presence(1)



Nightlight Simulation on Human Presence(2)

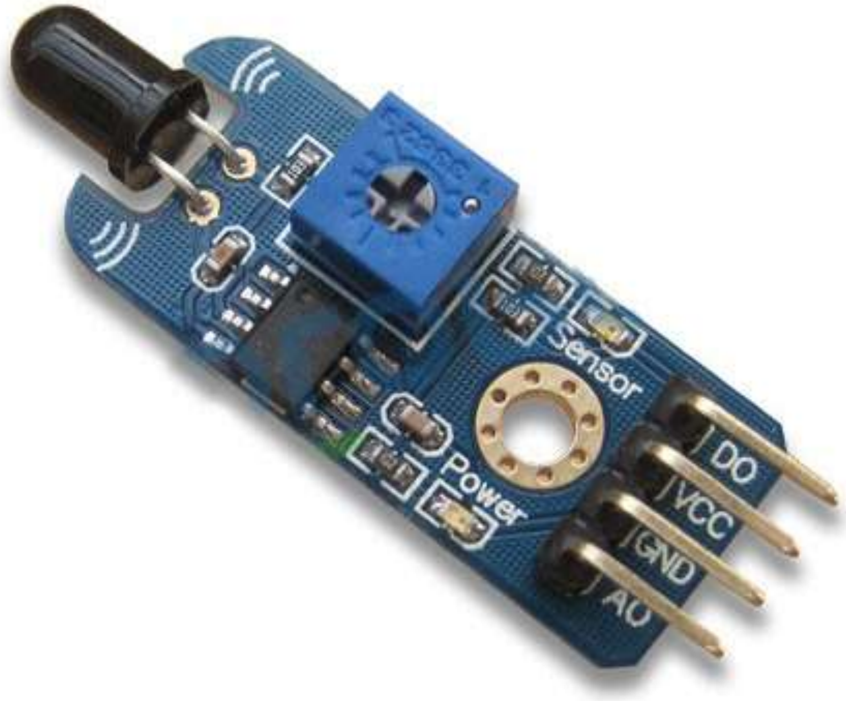


Nightlight Simulation on Human Presence(3)

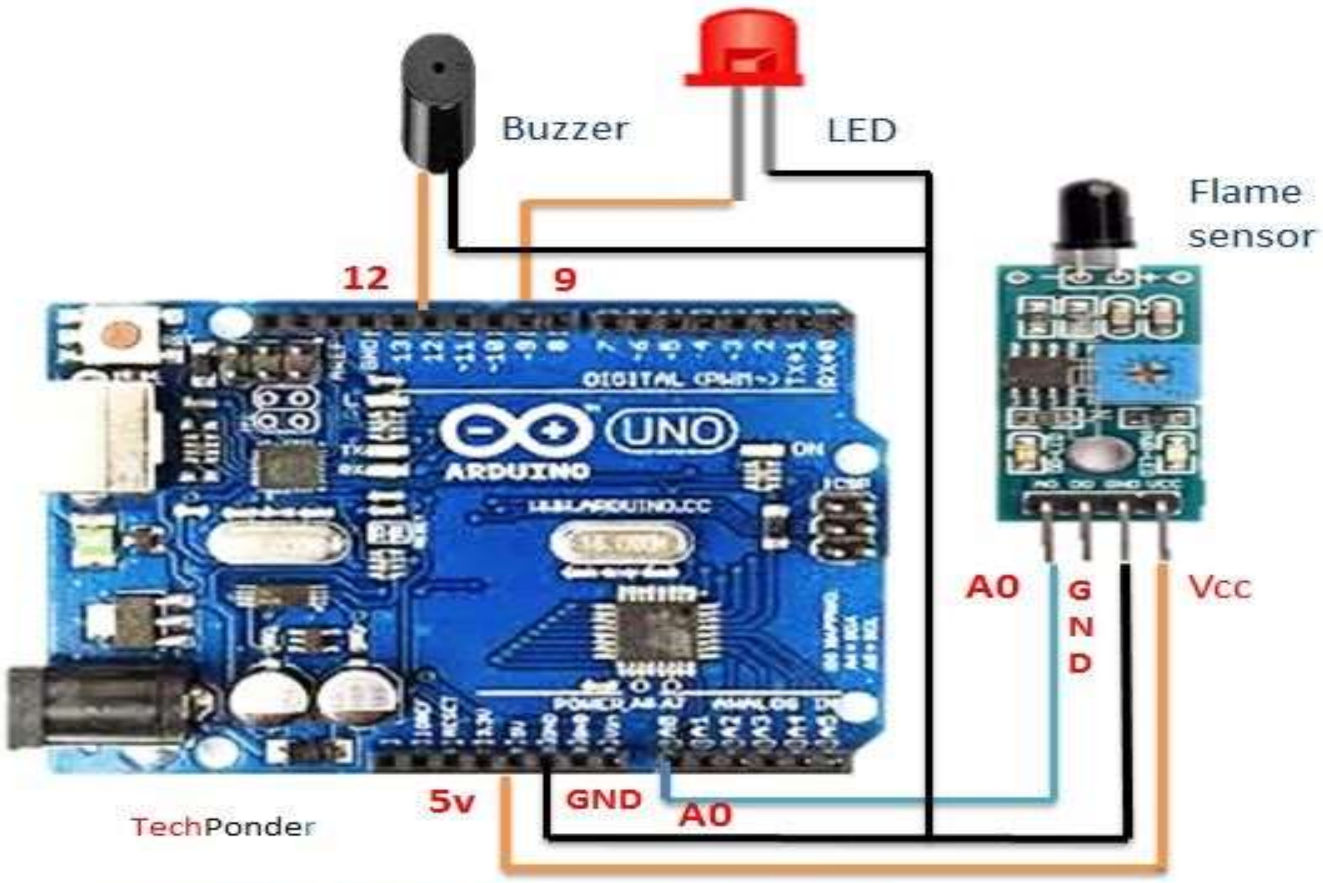
Connections:

- Attach one leg of LDR to 5V and another leg to Arduino Analog pin A0
- Attach one leg of 110K register with that leg of LDR connected to A0
- Attach another leg of register to the ground
- Connect the positive leg of LED to pin 11 and negative to GND
- **Connect positive leg of PIR to 5V**
- **Connect negative leg of PIR to GND**
- **Connect output pin of PIR to digital pin 3**
- **Write the code to switch on LED only when there is movement and less light.**

Fire Alert(1)

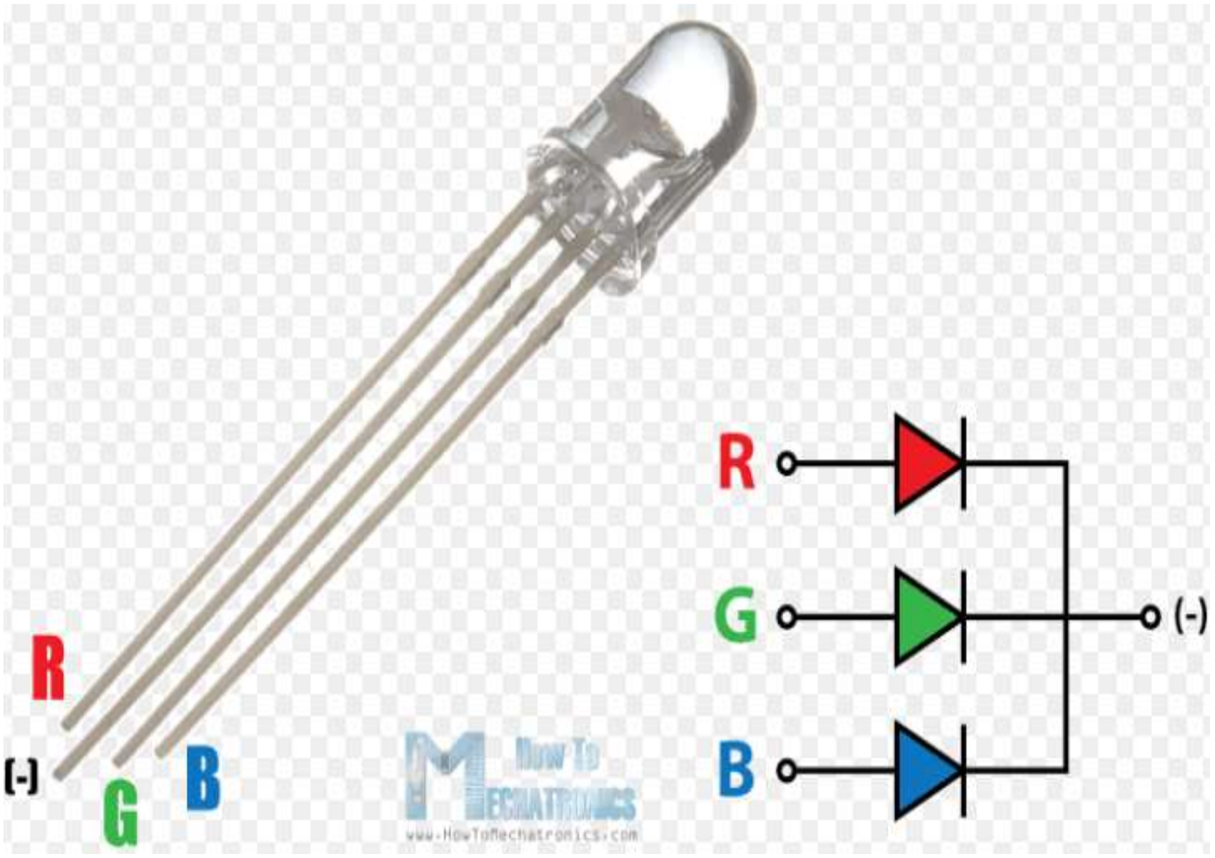


Fire Alert(2)



Flame Detection using Arduino

Color Recognition with RGB LED(1)



RGB LED Code(1)

```
int redPin = 11;
int greenPin = 10;
int bluePin = 9;
 void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
  void setColor(int red, int green, int blue)
{

  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

RGB LED Code(2)

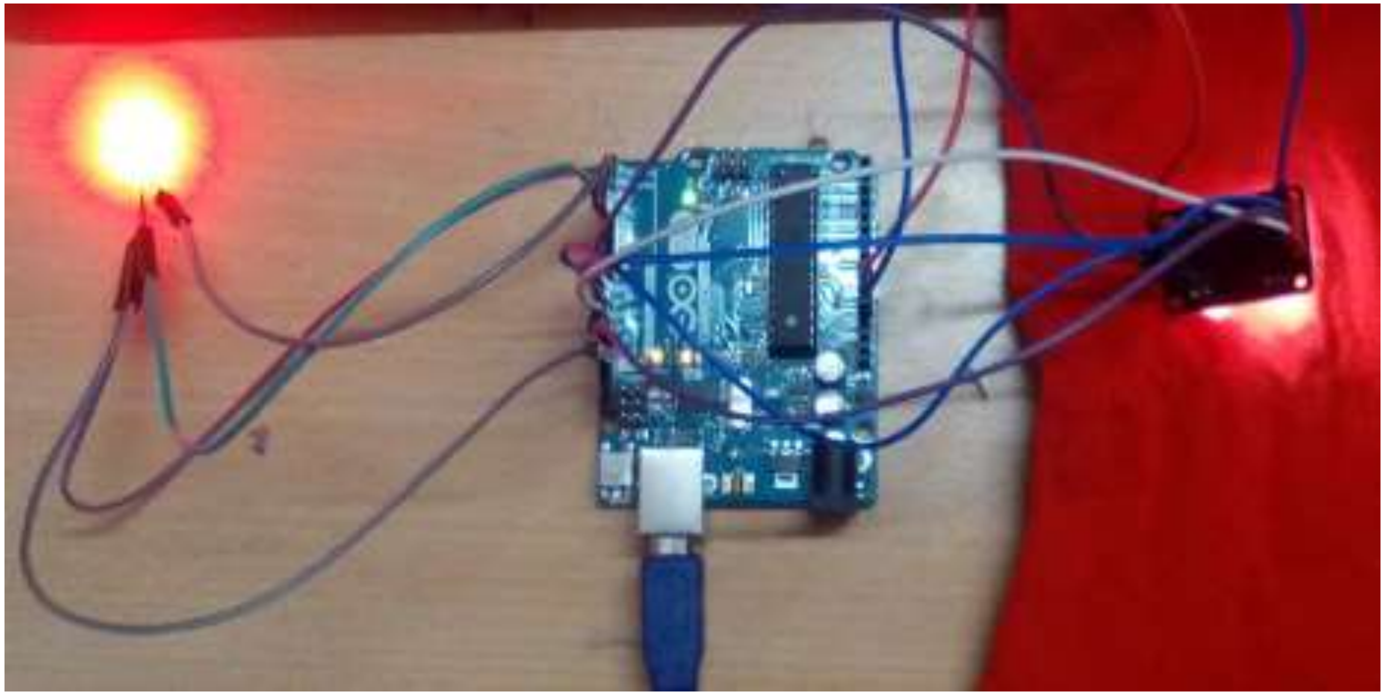
```
void loop()
{
  setColor(255, 0, 0);  // red
  delay(1000);
  setColor(0, 255, 0);  // green
  delay(1000);
  setColor(0, 0, 255);  // blue
  delay(1000);
  setColor(255, 255, 0);  // yellow
  delay(1000);
  setColor(80, 0, 80);  // purple
  delay(1000);
  setColor(0, 255, 255);  // aqua
  delay(1000);
}
```

Color Recognition with RGB LED(1)

Color Sensor	Arduino
-----------	--------
VCC	5V
GND	GND
s0	8
s1	9
s2	12
s3	11
OUT	10
OE	GND

RGB LED	Arduino
-----------	--------
R	2
G	3
B	4

Color Recognition with RGB LED(3)



Write the code to glow according to color in front of the sensor.