**EXPERIMENT 6**

**Objectives:**

* Control the LED using Ultrasonic sensor with Arduino.
* Connect PIR sensor with Arduino.
* Develop a device for Heart rate using Pulse sensor.

**Learning outcomes:**

* Turn ON a LED with the variation of distance using the Ultrasonic sensor in Arduino and print the distance.
* Program Arduino to detect any moment in the room or around the motion sensor.
* Monitor the Heart rate using Pulse sensor and Arduino.

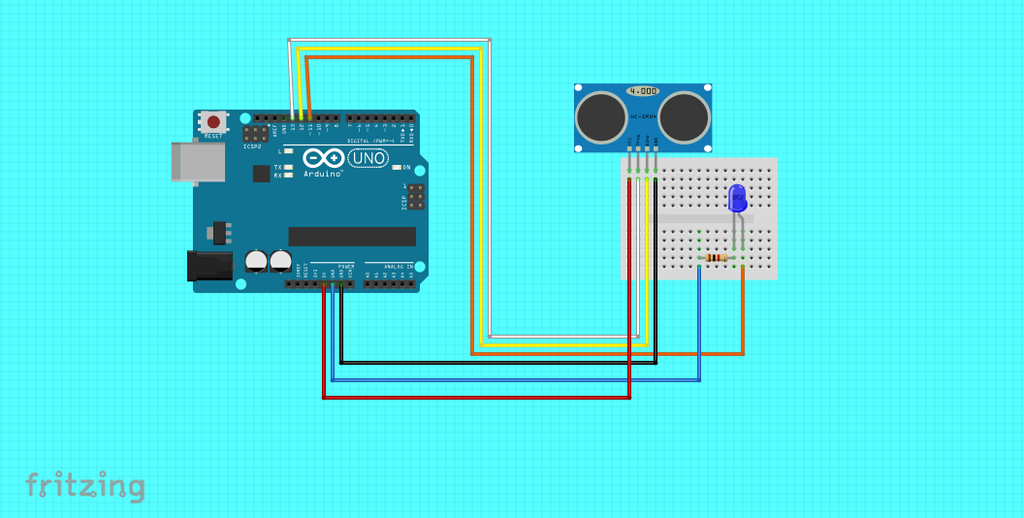
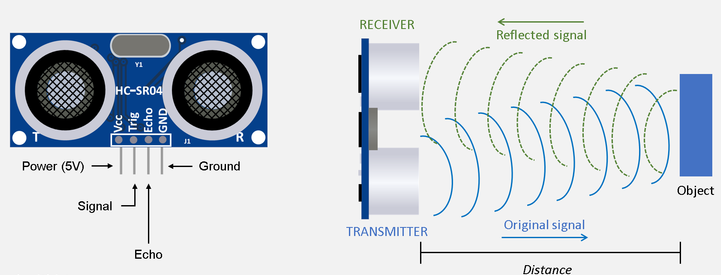
**Equipments/Components required:**

1. Arduino UNO with micro-USB cable - 1no
2. Ultrasonic sensor (HC-SR04) - 1no
3. PIR sensor - 1no
4. Pulse sensor (HW-827) - 1no
5. Jumper Wires - pack of wires
6. 1K ohm Resistor - 1no
7. Breadboard - 1no
8. LED - 1no

**PROCEDURE:**

**Activity1:** Control the LED using Ultrasonic sensor in Arduino IDE.

1. Connect the circuit as shown in the below figure.

1. Copy the below program in Arduino IDE.

#define echoPin 12 // attach pin D12 Arduino to pin Echo of HC-SR04

#define trigPin 13 //attach pin D13 Arduino to pin Trig of HC-SR04

// defines variables

long duration; // variable for the duration of sound wave travel

int distance; // variable for the distance measurement

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT

pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed

Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor

Serial.println("with Arduino UNO R3");

}

void loop() {

// Clears the trigPin condition

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin HIGH (ACTIVE) for 10 microseconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance = duration \* 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

if (distance < 10)

{

digitalWrite(11, HIGH);

}

else

{

digitalWrite(11, LOW);

}

// Displays the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.print(distance);

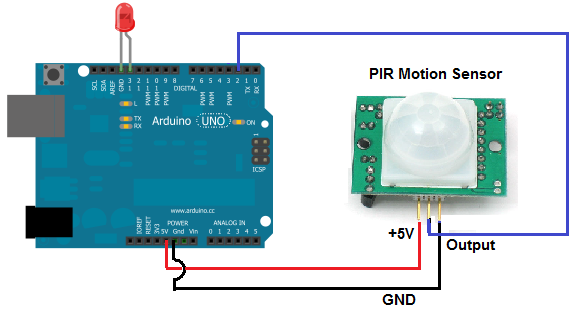
Serial.println(" cm");

}

1. Connect Hardware to your System Through Micro USB.
2. Select Board (**Arduino UNO**) and port in **select other board and port** menu.
3. Click on upload (arrow mark) option to compile and upload the program.
4. After uploading code, move any object in front of Ultrasonic sensor and observe the distance in Serial Monitor.
5. Check the LED whether it is turn ON or not.

**Activity2:** Connect PIR sensor with Arduino.

1. Connect the circuit as shown in the below figure.



1. Copy the below program in Arduino IDE.

int led = 13;                // the pin that the LED is atteched to

int sensor = 2;              // the pin that the sensor is atteched to

int state = LOW;             // by default, no motion detected

int val = 0;                 // variable to store the sensor status (value)

void setup()

{

  pinMode(led, OUTPUT);      // initalize LED as an output

  pinMode(sensor, INPUT);    // initialize sensor as an input

  Serial.begin(9600);        // initialize serial

}

void loop()

{

  val = digitalRead(sensor);   // read sensor value

  if (val == HIGH) {           // check if the sensor is HIGH

    digitalWrite(led, HIGH);   // turn LED ON

    delay(10);                // delay 10 milliseconds

    if (state == LOW) {

      Serial.println("Motion detected!");

      state = HIGH;       // update variable state to HIGH

    }

  }

  else {

      digitalWrite(led, LOW); // turn LED OFF

      delay(10);             // delay 10 milliseconds

      if (state == HIGH){

        Serial.println("Motion stopped!");

        state = LOW;       // update variable state to LOW

    }

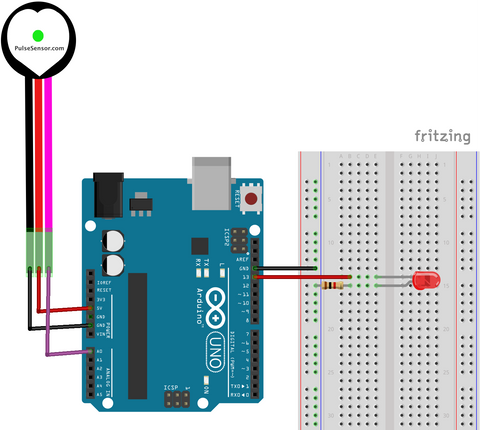
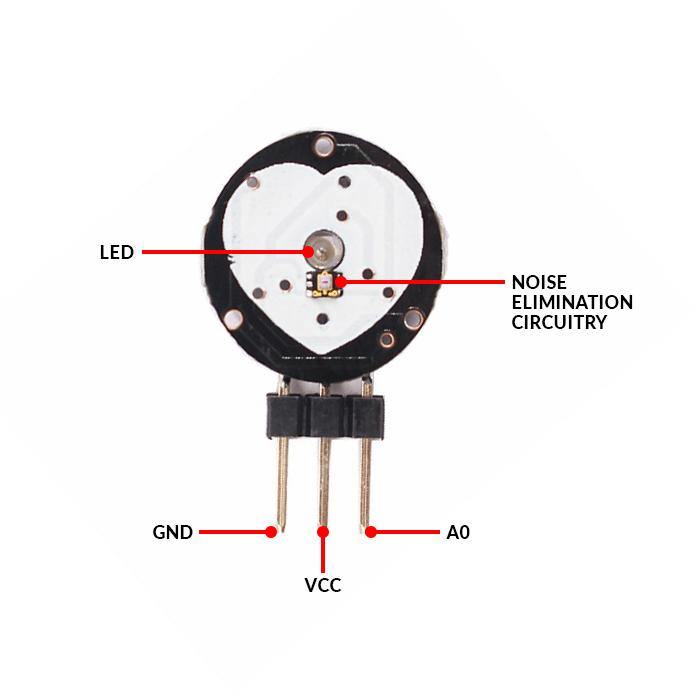
  }

}

1. Connect Hardware to your System Through Micro USB.
2. Select Board (**Arduino UNO**) and port in **select other board and port** menu.
3. Click on upload (arrow mark) option to compile and upload the program.
4. After upload the code, if any object moves around sensor then it automatically turn ON’s the LED.

**Activity3:** Develop a device for heart rate using Pulse sensor.

1. Connect the circuit as shown in the below figure.

1. Next, it’s necessary to install the **PulseSensor** library, which can be done through the Arduino Library Manager:

**Sketch** **→ Include Library → Manage Libraries….**

1. Enter “pulsesensor” in the search field and look through the list for the “PulseSensor Playground”.
2. Click the “Install” button, or “Update” from an earlier version.
3. For the code, follow the below procedure.

**Examples → PulseSensor Playground → GettingStartedProject**

1. Click on upload (arrow mark) option to compile the program.
2. After upload the code, hold the pulse sensor with your fingers and open Serial Monitor and Serial Plotter to check the Heart rate.