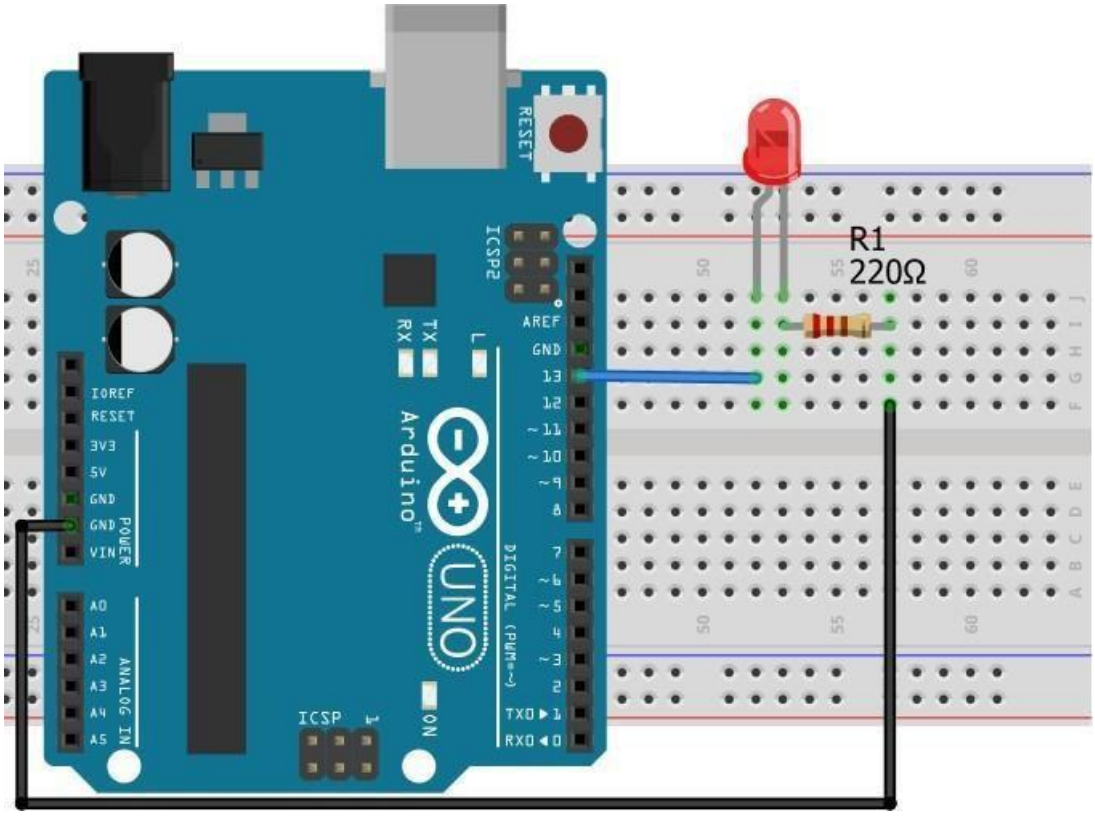
Experiment : Open-source prototype platform- Raspberry-Pi/Beagle board/Arduino -Simple program digital read/write using LED and Switch -Analog read/write using sensor and actuators.

/* 1. Digital Read/Write*/

Program For On -OFF the LED

```
int led1 = 12;
void setup() {
  // put your setup code here, to run once:
  pinMode(led1, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(100);                // wait for a second
  digitalWrite(led1, LOW);  // turn the LED off by making the voltage LOW
  delay(100);                // wait for a second
}
```



Program For On -OFF the LED by using Swich

```
#define ledPin 13 // choose the pin for the LED
#define switchPin 8 // choose the input pin (for a pushbutton)
int val = 0; // variable for reading the pin status
void setup()
{
pinMode(ledPin, OUTPUT); // declare LED as output
```

```
pinMode(switchPin, INPUT); // declare pushbutton as input
}
void loop()
{
val = digitalRead(switchPin); // read input value
if (val == HIGH)        // check if the input is HIGH (button released)
{
digitalWrite(ledPin, LOW); // turn LED OFF
}
else {
digitalWrite(ledPin, HIGH); // turn LED ON
}
}
```

OR

```
int buttonPin = 2;     // choose the pin for the LED
int ledPin = 12;    // choose the input pin (for a pushbutton)

int buttonState = 0;
void setup() {

  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT);
}
void loop() {
 buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {

    digitalWrite(ledPin, HIGH);
  } else {
      digitalWrite(ledPin, LOW);
  }
}
```

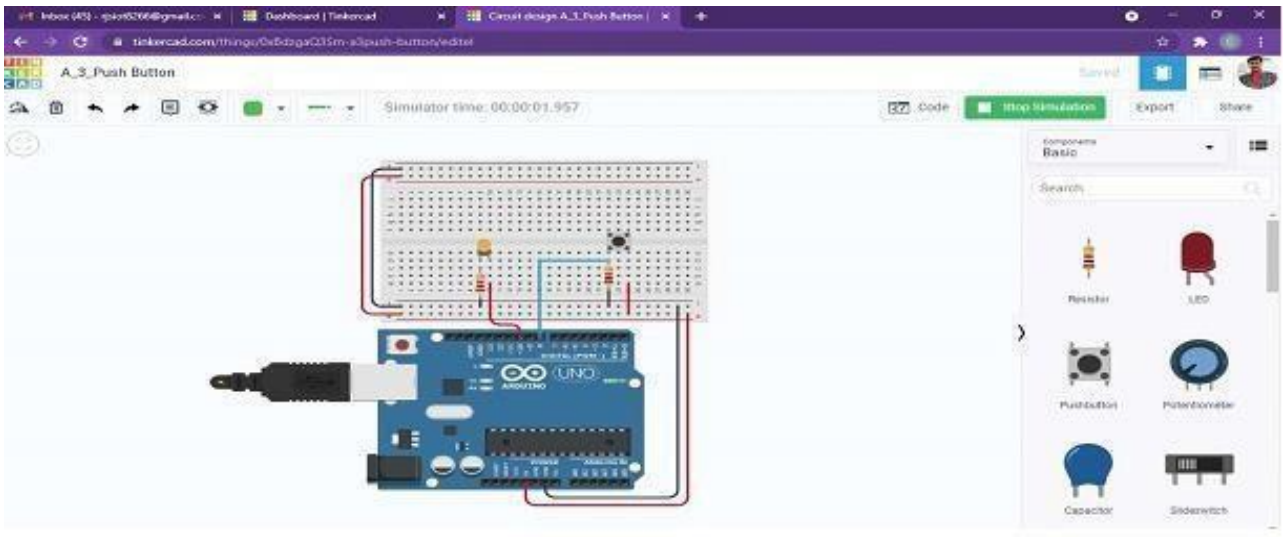
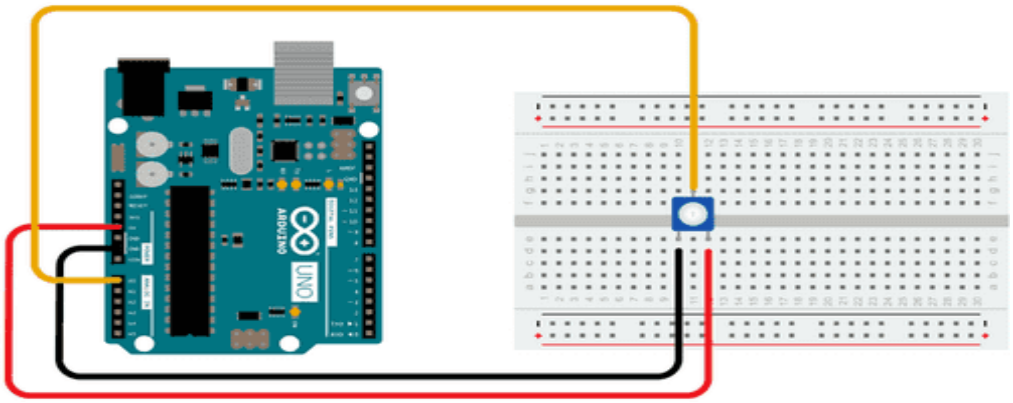


Fig.3 Simulated Result of LED & Push Button

/* 2. Analog Read
***/**
/* Reads an analog input on pin 0, prints the result to the Serial Monitor. Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu). Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.*/
// the setup routine runs once when you press reset:

Program For read the data from Potentiometer and display on serial monitor:

```
void setup() {
// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
}
// the loop routine runs over and over again forever:
void loop() {
// read the input on analog pin 0:
int sensorValue = analogRead(A0);
// print out the value you read:
Serial.println(sensorValue);
delay(100); // delay in between reads for stability
 float voltage = sensorValue * (5.0 / 1024.0);  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
Serial.println(voltage);
}
```



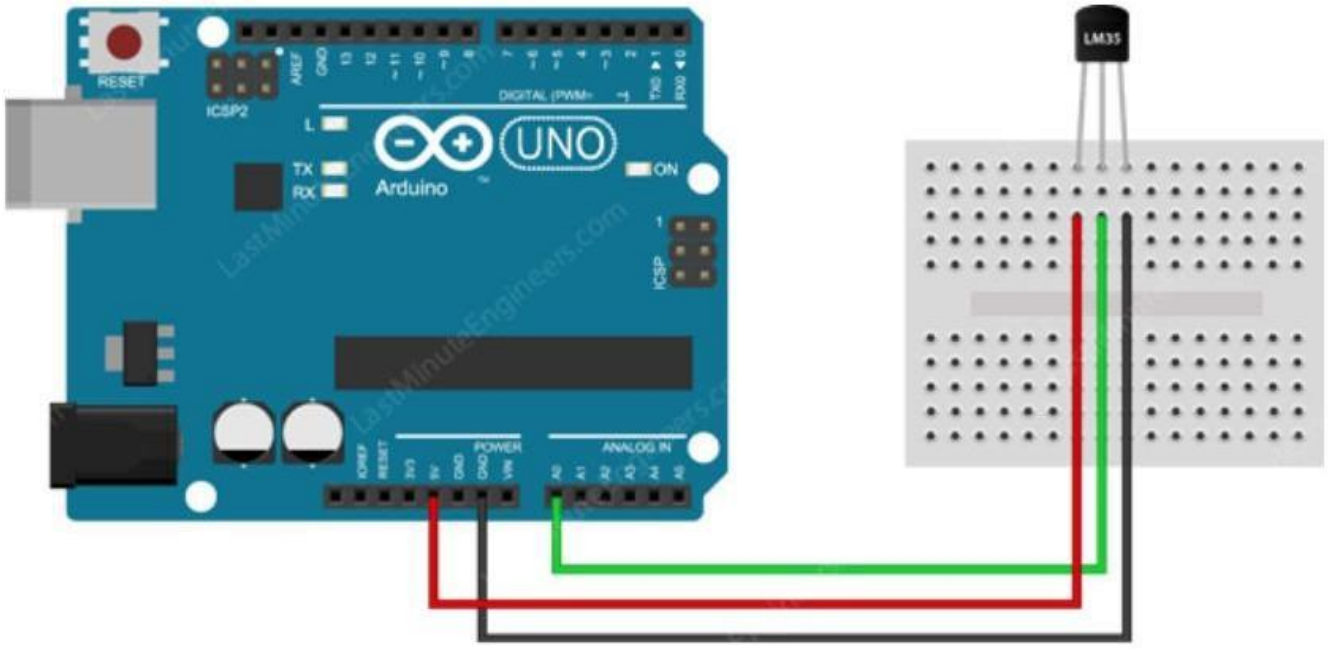
/* 3. Analog Read/Write*/

```
int PWM_Output = 10;
void setup() {
pinMode(PWM_Output,OUTPUT);
}
void loop() {
int analogValue = analogRead(A0);
for(int i=0;i<=255; i++)
{
analogWrite(PWM_Output,analogValue);
analogWrite(PWM_Output,i);
delay(100);
} }
```

Experiment: Interfacing sensors and actuators with Arduino/Raspberry-pi.
PROGRAM CODE:

// 1. Program for Interfacing of Temperature Sensor (LM35) with Arduino and display the reading on the serial monitor:

```
#define sensorPin A0
void setup() {
// Begin serial communication at 9600 baud rate
Serial.begin(9600);
}
void loop() {
// Get the voltage reading from the LM35
int reading = analogRead(sensorPin);
// Convert that reading into voltage
float voltage = reading * (5.0 / 1024.0);
// Convert the voltage into the temperature in Celsius
float temperatureC = voltage * 100;
// Print the temperature in Celsius
Serial.print("Temperature: ");
Serial.print(temperatureC);
Serial.print("\xC2\xB0"); // shows degree symbol
Serial.print("C | ");
// Print the temperature in Fahrenheit
float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
Serial.print(temperatureF);
Serial.print("\xC2\xB0"); // shows degree symbol
Serial.println("F");
delay(1000); // wait a second between readings
}
```



// 2.Program for Interface IR Sensor with Arduino and display the information about obstacle object on the serial monitor

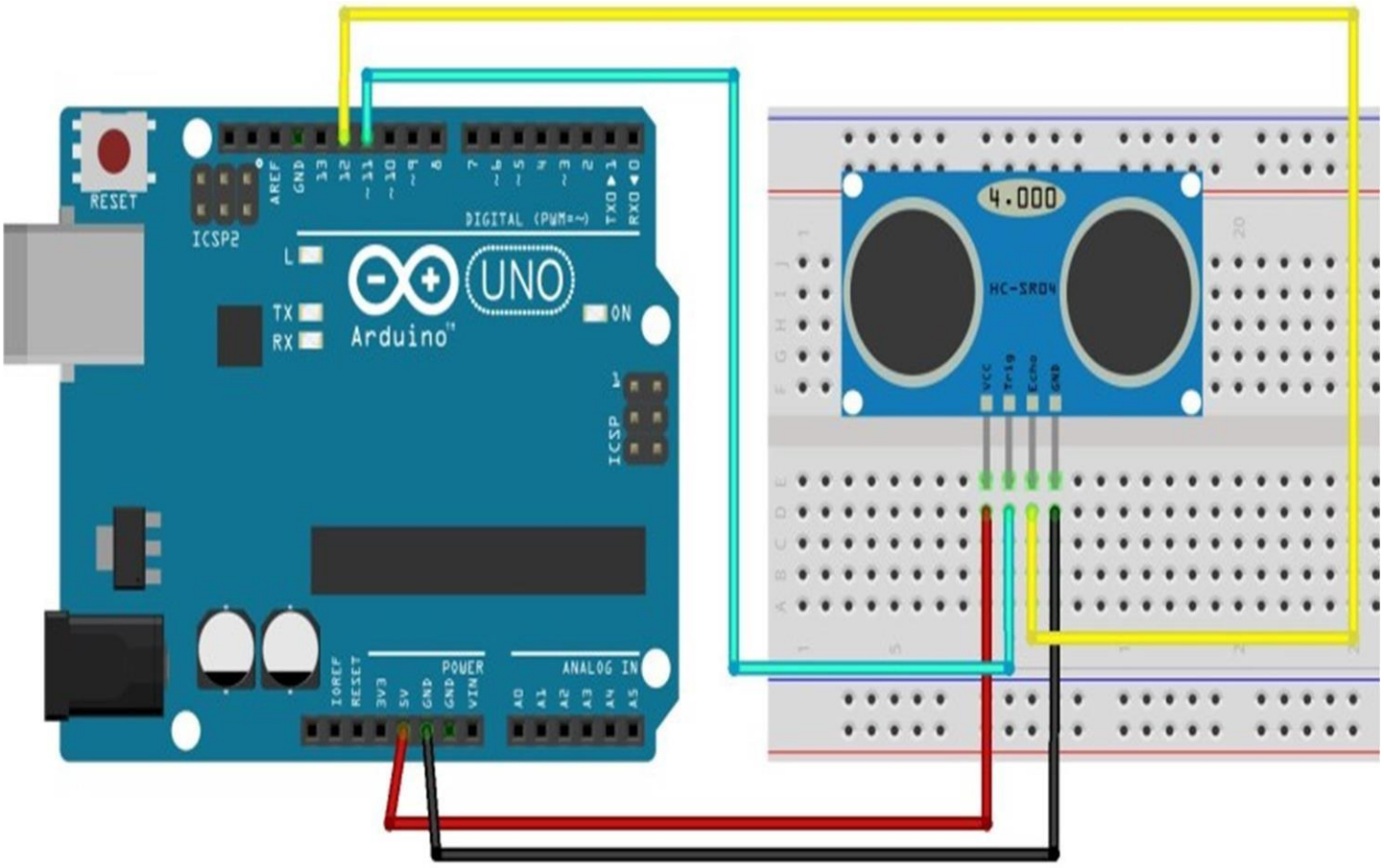
```
int ledPin=13;
int inputPin=2;
int val=0;
void setup()
{
pinMode(ledPin,OUTPUT);
pinMode( inputPin, INPUT);
Serial.begin(9600);
}
void loop()
{
val=digitalRead(inputPin); // check the pin status (High=1/Low=0) //Active Low output
if(val==HIGH)
{
Serial.print(“Object Absent\n");
digitalWrite(ledPin,LOW);
}
else
{ Serial.print("Object Present\n");
digitalWrite(ledPin,HIGH);
} }
```

Interfacing of IR Sensor

/*3. Program for Interface of Ultrasonic Sensor with Arduino and display the distance of obstacle using the serial monitor

```
Ultrasonic Sensor HC-SR04 interfacing with Arduino.*/
// defining the pins
const int trigPin = 9;
const int echoPin = 2;
// defining variables
long duration;
int distance;
void setup() {
```

```
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
}
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
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;
// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```



Experiment: IoT based Stepper Motor/DC Motor Control with Arduino/Raspberry Pi.
PROGRAM CODE:

/* 1 Code to spin the mini servomotor*/

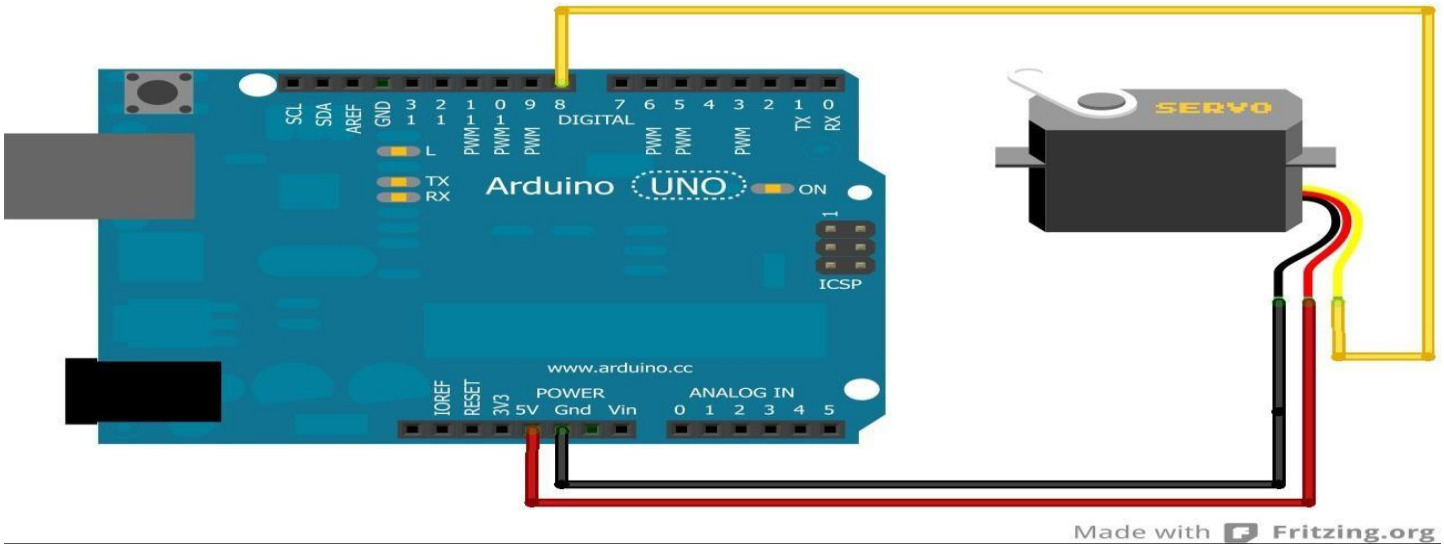
```
#include <Servo.h>

Servo myservo;
int value;
double angle;
void setup()
{
  Serial.begin(9600);
  myservo.attach(9);
}

void loop()
{
  value = analogRead(A0);
  angle = map(value, 0, 1023, 0, 180);
  Serial.println(angle);
  myservo.write(angle);
  delay(15);
}
```

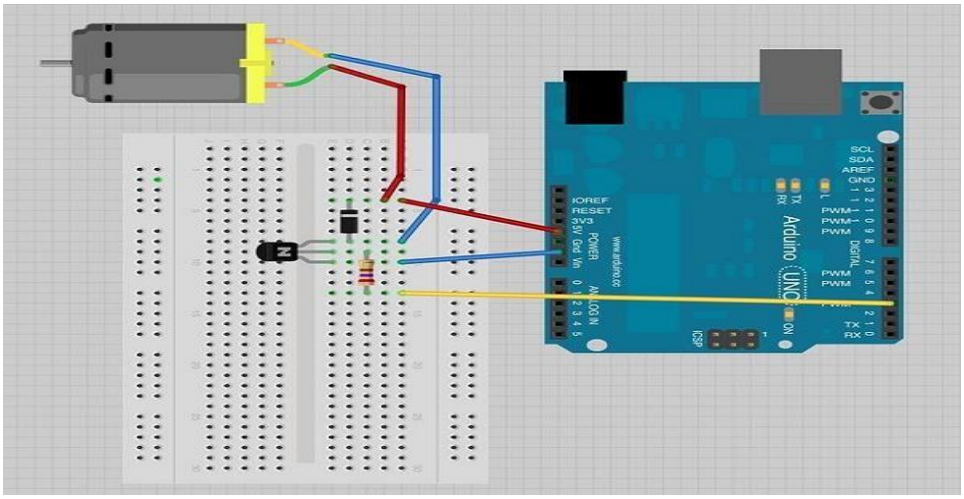
/* 2 Code to controlling spin the mini servomotor in degree */

```
#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
  }
}
```

/* 3 Code to controlling spin the DC Motor */

```
#include <Servo.h> //Include the servo library
Servo servoblue; //The servo gets the name “servoblue”
void setup()
{
servoblue.attach(8); //The signal line of the servo is on pin 8
}
void loop()
{
  servoblue.write(0); //Position 1 with an angle of 0°
delay(3000); //Wait 3 seconds
servoblue.write(90); //Position 2 with an angle of 90°
delay(3000); //Wait 3 seconds
servoblue.write(180); //Position 3 with an angle of 180°
delay(3000); //Wait 3 seconds
servoblue.write(20); //Position 4 with an angle of 20°
delay(3000); //Wait 3 seconds
}
```





/* 4 Code to spin the motor*/

```
int motorPin = 3;
void setup() { }
void loop() { digitalWrite(motorPin, HIGH);}
```

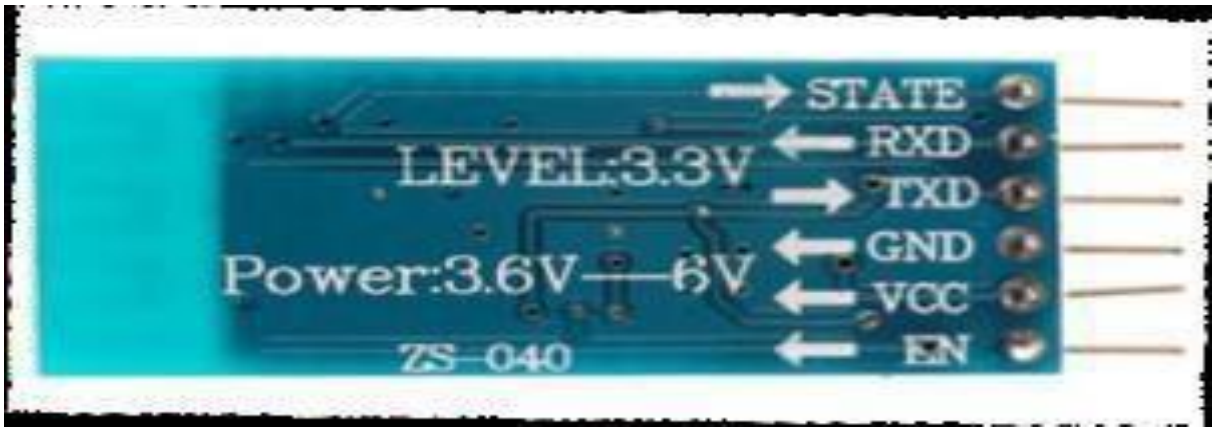
Instead of spinning at full speed, to control the DC motor speed, the code can be written as:

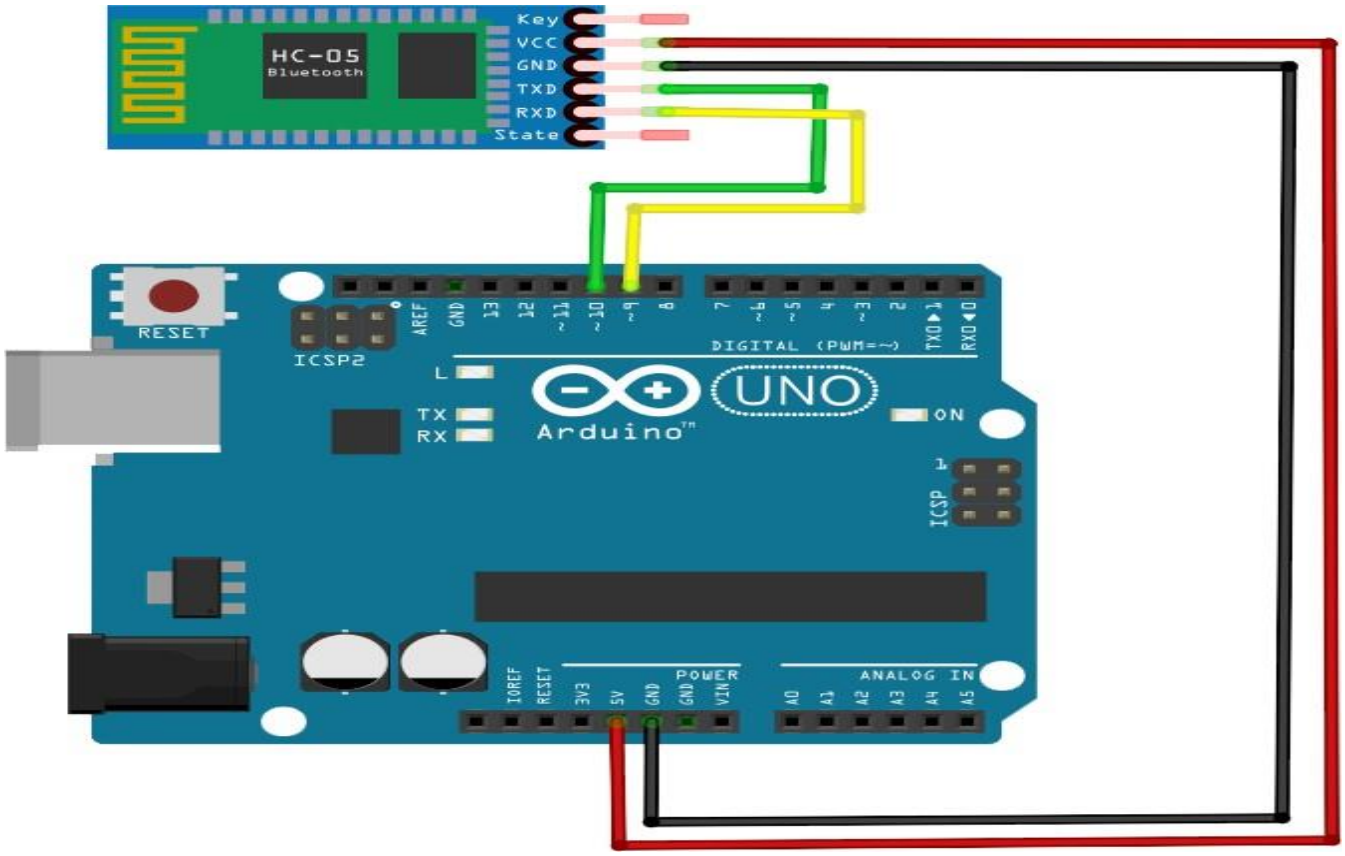
/*5 Code to controlling MOTOR SPEED CONTROL*/

```
int motorPin = 9;
void setup() {
pinMode(motorPin, OUTPUT);
Serial.begin(9600);
while (! Serial);
Serial.println("Speed 0 to 255");
}
void loop() {
if (Serial.available()) {
int speed = Serial.parseInt();
if (speed >= 0 && speed <= 255) {
analogWrite(motorPin, speed); } }}
```

Experiment : Interfacing Arduino to Bluetooth Module

PROGRAM CODE:





/*1 // Basic Bluetooth sketch HC-05
// Connect the Hc-05 module and communicate using the serial monitor
// First powered on set the default baud rate to 9600 and perform the by using Mobile*/

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3); (10,9)
```

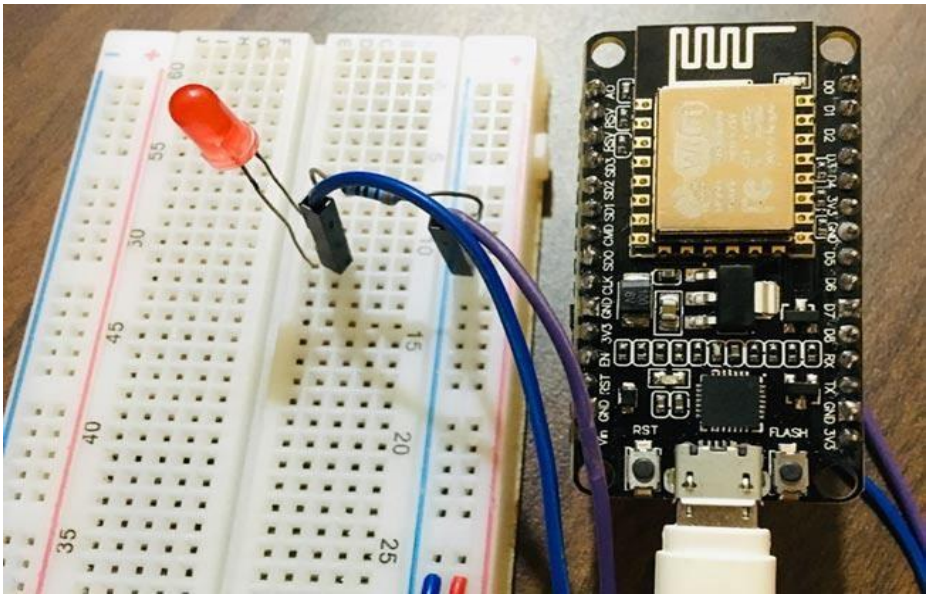
```
void setup() {
  Serial.begin(9600); //open the serial port
  mySerial.begin(9600); // open the bluetooth serial port
}

void loop() {
  if(mySerial.available()){
    Serial.println(mySerial.readString()); // send from serial to bluetooth
  }
```

/*2 // Basic Bluetooth sketch HC-05
// Connect the Hc-05 module and communicate using the serial monitor
// First powered on set the default baud rate to 9600 and perform the communication as well as ON /OFF the LED by using Mobile*/

```
#include <SoftwareSerial.h>
SoftwareSerial Bluetooth(10, 9); // RX, TX
int LED = 13; // the on-board LED
int Data; // the data received

void setup() {
  Bluetooth.begin(9600);
  Serial.begin(9600);
  Serial.println("Waiting for command...");
  Bluetooth.println("Send 1 to turn on the LED. Send 0 to turn Off");
  pinMode(LED,OUTPUT);

}

void loop() {
  if (Bluetooth.available()){ //wait for data received
    Data=Bluetooth.read();
    if(Data=='1'){
      digitalWrite(LED,1);
      Serial.println("LED On!");
      Bluetooth.println("LED On!");
    }
    else if(Data=='0'){
       digitalWrite(LED,0);
       Serial.println("LED Off!");
       Bluetooth.println("LED  On D13 Off ! ");
    }
    else{;}
  }
delay(100);
 }
```

Experiment: Get the status of a LED on web page.

Program code:




```
#include <ESP8266WiFi.h>

const char* ssid = "USERNAME";     // Your Wi-Fi Name
const char* password = "PASSWORD";  // Wi-Fi Password

int LED = 2; // led connected to GPIO2 (D4)
WiFiServer server(80);

void setup()
{
  Serial.begin(115200); //Default Baudrate
  pinMode(LED, OUTPUT);
  digitalWrite(LED, LOW);

  Serial.println("Connecting to the Newtork");
  WiFi.begin(ssid, password);
  Serial.print(ssid);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");
  server.begin();  // Starts the Server
  Serial.println("Server started");
```

```
  Serial.print("IP Address of network: "); // will IP address on Serial Monitor
  Serial.print("http://");
  Serial.println(WiFi.localIP());
  Serial.print("Copy and paste the following URL: https://"); // Will print IP address in URL format
  Serial.print(WiFi.localIP());
  Serial.println("/");
}

void loop()
{
  WiFiClient client = server.available();
  if (!client)
  {
    return;
  }
  Serial.println("Waiting for new client");
  while(!client.available())
  {
    delay(1);
  }

  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();

  int value = LOW;
  if(request.indexOf("/LED=ON") != -1)
  {
    digitalWrite(LED, HIGH); // Turn LED ON
    value = HIGH;
  }
  if(request.indexOf("/LED=OFF") != -1)
  {
    digitalWrite(LED, LOW); // Turn LED OFF
    value = LOW;
  }
client.println("HTTP/1.1 200 OK"); //
  client.println("Content-Type: text/html");
  client.println("");
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

  client.print(" CONTROL LED: ");

  if(value == HIGH)
  {
    client.print("ON");
  }
  else
  {
    client.print("OFF");
  }
  client.println("<br><br>");
  client.println("<a href=\"/LED=ON\"\"><button>ON</button></a>");
  client.println("<a href=\"/LED=OFF\"\"><button>OFF</button></a><br />");
```

```
  client.println("</html>");

  delay(1);
  Serial.println("Client disonnected");
  Serial.println("");
}
```