

Smart Home System Using Mobile Application

Project Description

A smart home is a residence that uses connected devices to enable the remote monitoring and management of appliances and systems, such as lighting and heating.

In this project we will use an Android mobile application to control the following:

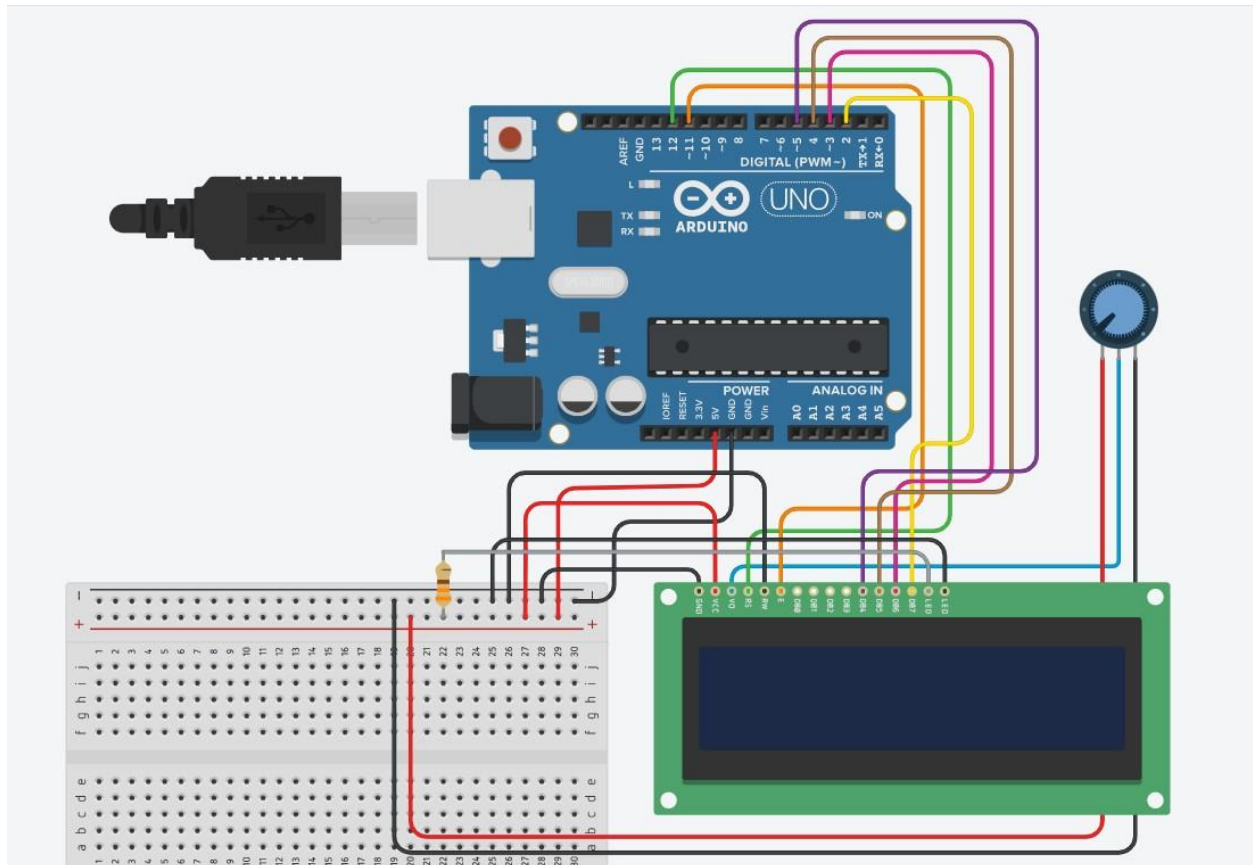
- How to Turn the Lights ON or OFF.
- How to Open or Close the door with password.
- How to Monitor the Human Heart Rate.
- How to Determine the Climate.

Components

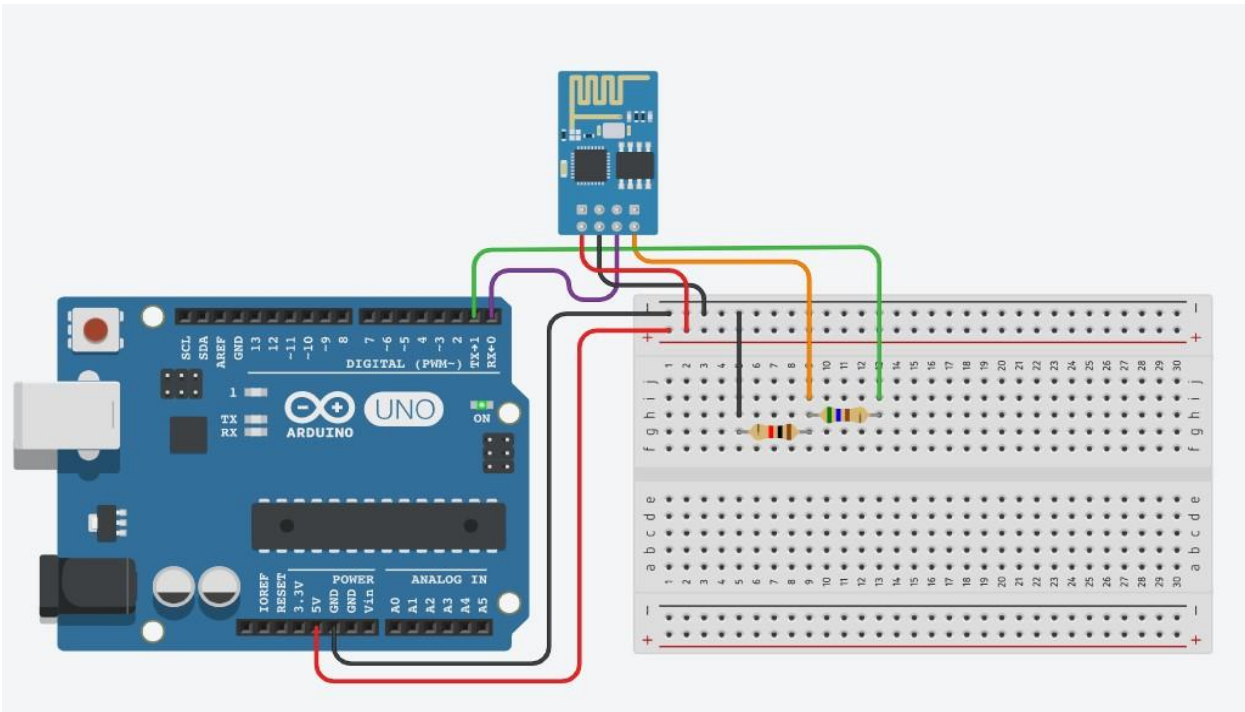
1. Arduino UNO.
2. LCD 16x2.
3. Leds.
4. Temperature Sensor (LM35).
5. Stepper Motor.
6. Buzzer.
7. Bluetooth Module (HC-06).
8. Potentiometer.
9. Max30102 Heart Rate Module.
10. Resistors (560-ohm, 330-ohm and 1K-ohm).
11. BreadBoard.

Connections

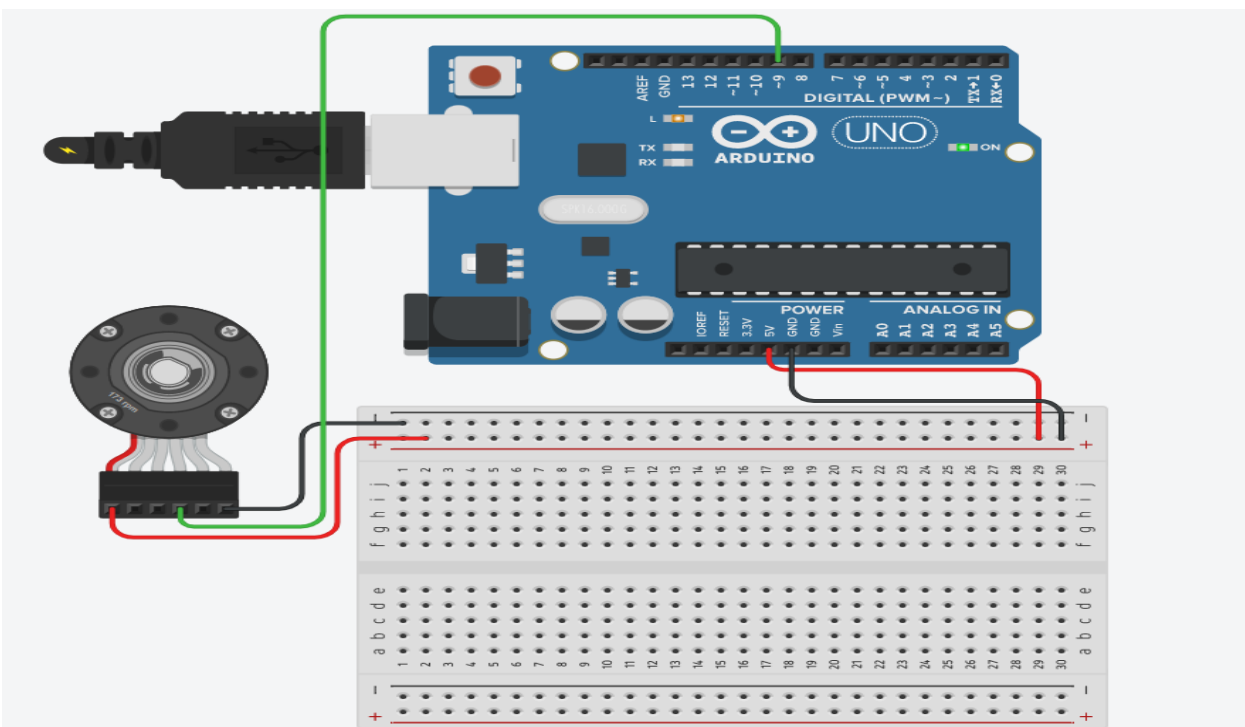
1. Arduino with LCD.



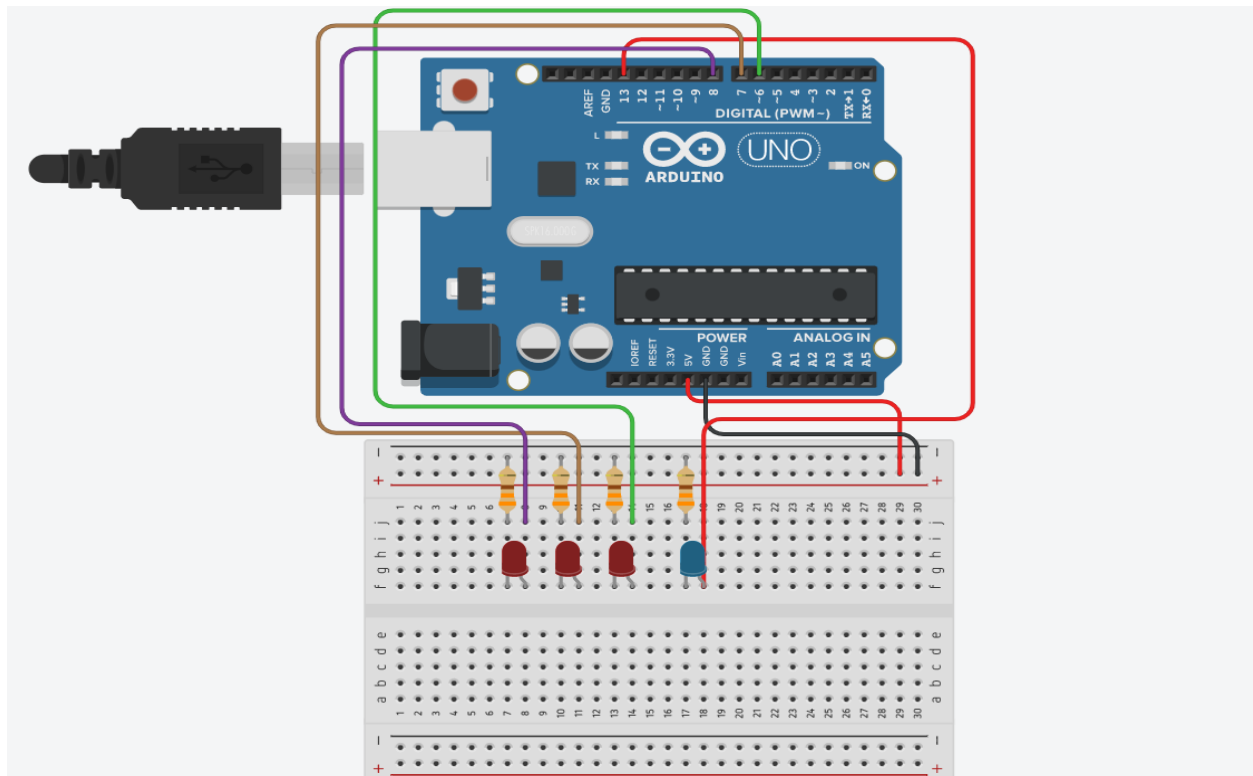
2. Arduino with Bluetooth Module.



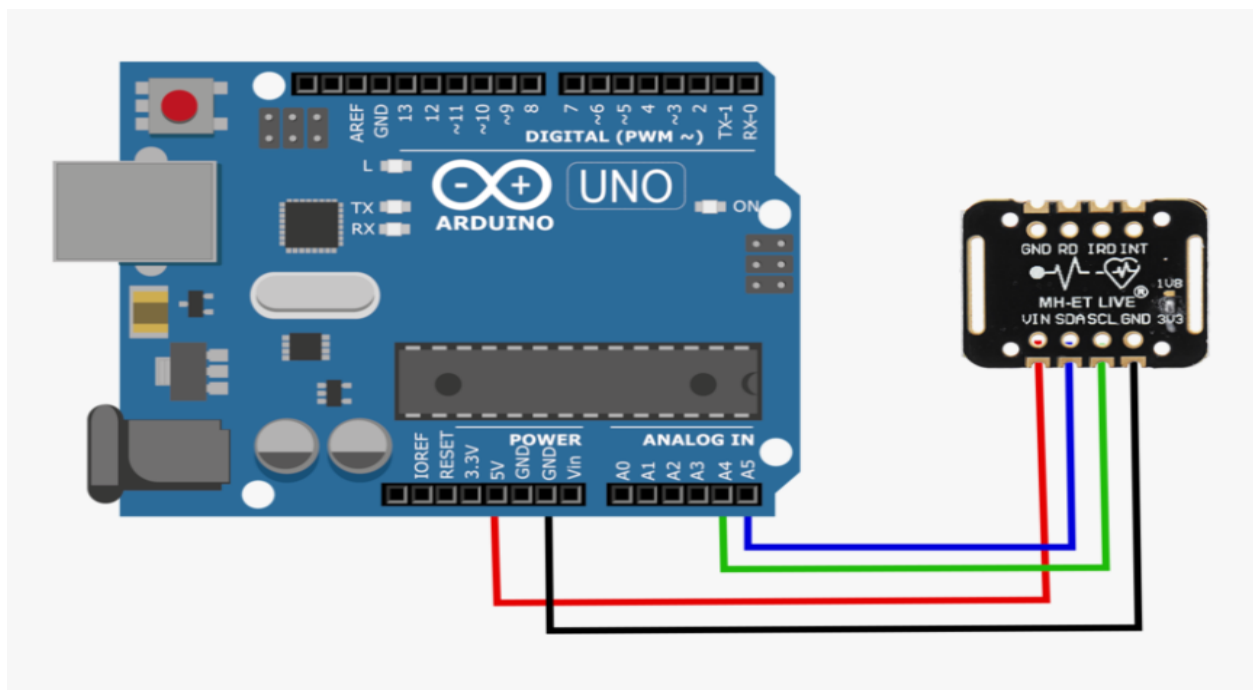
3. Arduino with Stepper Motor.



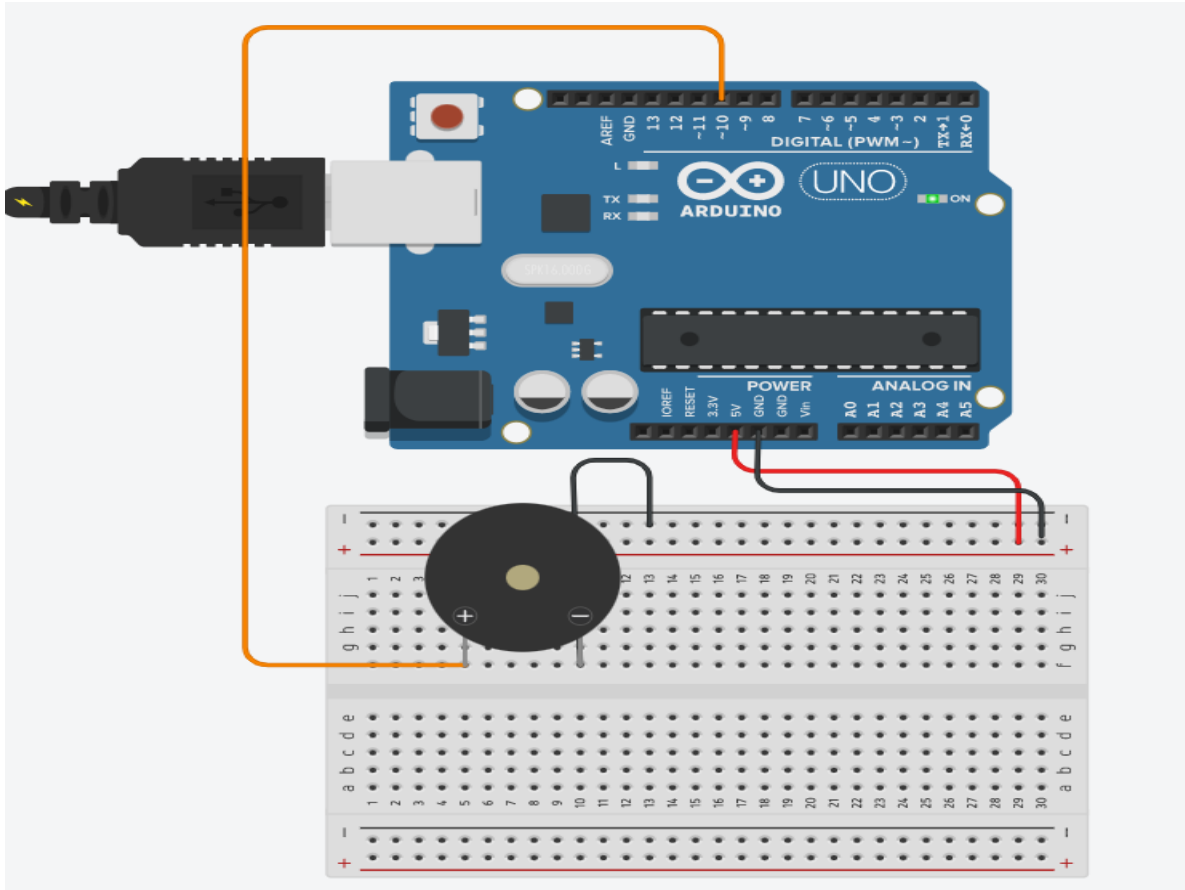
4. Arduino with Leds.



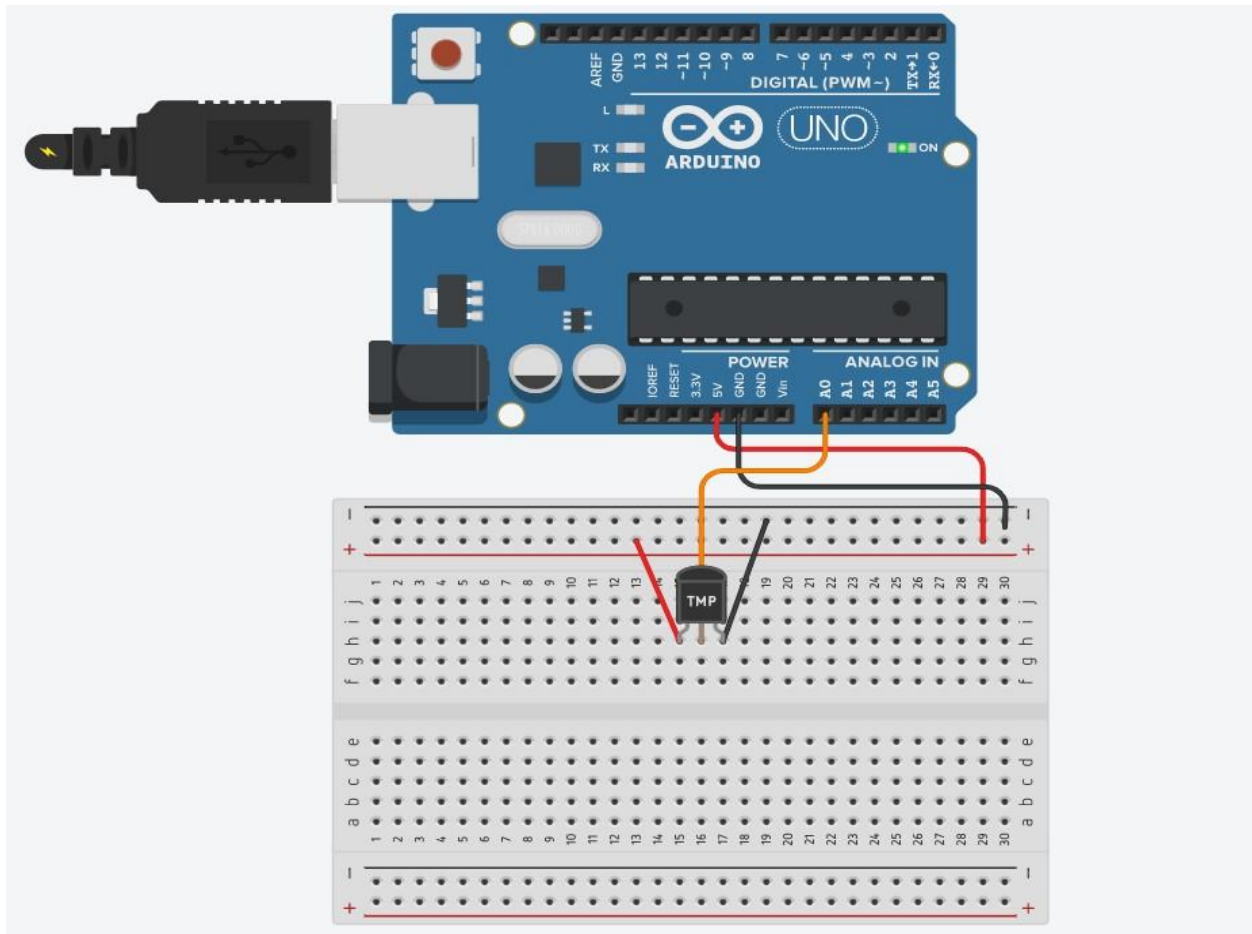
5. Arduino with Max30102.



6. Arduino with Buzzer.



7. Arduino with LM35.



Code

1. Include Libraries and define variables.

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <Servo.h> // servo library
#include <EEPROM.h>
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

//Bluetooth Serial
SoftwareSerial Blue = SoftwareSerial(0, 1);

//define servo motor
Servo myservo;

//define LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

//define max30102 sensor
MAX30105 particleSensor;
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;

int BUZZ = 10; // Buzzer at pin 10

String data = " ";
int opened = 0;
```


2. Setup Function.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(0, INPUT);    //Bluetooth RX  
    pinMode(1, OUTPUT);   //Bluetooth TX  
    pinMode(6, OUTPUT);   //Led  
    pinMode(7, OUTPUT);   //Led  
    pinMode(8, OUTPUT);   //Led  
    pinMode(9, OUTPUT);   //Servo motor  
    pinMode(10, OUTPUT);  //Buzzer  
    pinMode(13, OUTPUT);  //Led  
    pinMode(14, OUTPUT);  //Temperature sensor  
    pinMode(18, OUTPUT);  //Max30102 SCL  
    pinMode(19, OUTPUT);  //Max30102 SDA  
  
    myservo.attach(9);    //attaches the servo to digital pin 9  
    myservo.write(0);  
  
    Wire.begin();  
    Serial.begin(115200);  
    Blue.begin(9600);     //Set the baud rate as 9600  
    lcd.begin(16, 2);     //initialize the lcd for 16 chars 2 lines  
    lcd.clear();  
    noTone(10);  
}
```

3. Loop Function.

```
void loop() {  
  // put your main code here, to run repeatedly:  
  Blue.listen();  
  while (Blue.available() > 0)  
  {  
    data = Blue.readString();  
    //Door Opening  
    if (data == "0") {  
      if (opened == 0) {  
        digitalWrite(6, LOW);  
        digitalWrite(7, LOW);  
        digitalWrite(8, LOW);  
        lcd.clear();  
        lcd.setCursor(0, 0);  
        lcd.print("Right Password");  
        delay(1000);  
        lcd.clear();  
        for (int x = 0; x <= 180; x += 1) // Rotates the servo to the unlocked position  
        {  
          myservo.write(x);  
          delay(10);  
        }  
        opened = 1;  
        delay(1000);  
        lcd.setCursor(0, 0);  
        lcd.print("Opened");  
        delay(1000);  
        lcd.clear();  
      }  
    }  
  }  
}
```

```

else if (data == "W")
{
    if (digitalRead(6) == LOW && digitalRead(7) == LOW && digitalRead(8) == LOW)
    {
        digitalWrite(6, HIGH);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Wrong Password");
        delay(1000);
        lcd.clear();
    }
    else if (digitalRead(6) == HIGH && digitalRead(7) == LOW && digitalRead(8) == LOW)
    {
        digitalWrite(7, HIGH);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Wrong Password");
        delay(1000);
        lcd.clear();
    }
    else if (digitalRead(6) == HIGH && digitalRead(7) == HIGH && digitalRead(8) == LOW)
    {
        digitalWrite(8, HIGH);
        lcd.print("Wrong Password");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("The Door is Locked");
        digitalWrite(6, LOW);
        digitalWrite(7, LOW);
        digitalWrite(8, LOW);
        delay(3000);
        lcd.clear();
    }
    //Door Closing
    else if (data == "C")
    {
        if (opened == 1) {
            for (int x = 180; x >= 0; x -= 1) // Rotates the servo to the unlocked position
            {
                myservo.write(x);
                delay(10);
            }
            delay(2000);
            lcd.setCursor(0, 0);
            lcd.print("Closed");
            delay(1000);
            lcd.clear();
            opened = 0;
        }
    }
    //Lightning ON/OFF
    else if (data == "N")
    {
        digitalWrite(13, HIGH);
    }
    else if (data == "F")
    {
        digitalWrite(13, LOW);
    }
}

```

```

else if (data == "T")
{
    int temp = analogRead(A0);
    //Convert digital data into analog by multiplying by 5000 and dividing by 1024
    float voltage = temp * (5.0 / 1024.0);
    // Convert the voltage into the temperature in degree Celsius:
    float tempC = voltage * 100;
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temperature= ");
    lcd.print(tempC);
    lcd.print(" C");
    if (tempC > 50)
    {
        tone(BUZZ, 450);
        delay(500);
        noTone(BUZZ);
        delay(500);
        tone(BUZZ, 450);
        delay(500);
        noTone(BUZZ);
        delay(500);
        tone(BUZZ, 450);
        delay(500);
        noTone(BUZZ);
        for(int x = 0; x <= 180; x += 1)
        {
            myservo.write(x);
            delay(10);
        }
        opened = 1;
    }
}

```

```

//Heart Rate
else if (data == "H")
{
    //oximeter.listen();
    if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {
        Serial.println("MAX30102 was not found. Please check wiring/power. ");
        while (1);
    }
    particleSensor.setup(); //Configure sensor with default settings
    particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
    particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED

    Serial.println("Initializing...");
    while(Serial.available() > 0) {
        long irValue = particleSensor.getIR();
        if (checkForBeat(irValue) == true) {
            //We sensed a beat!
            long delta = millis() - lastBeat;
            lastBeat = millis();
            beatsPerMinute = 60 / (delta / 1000.0);
            //anArray[arrayIndex] = beatsPerMinute;
            //arrayIndex++;
            if (beatsPerMinute < 255 && beatsPerMinute > 20) {
                rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
                rateSpot %= RATE_SIZE; //Wrap variable
                //Take average of readings
                beatAvg = 0;
                for (byte x = 0 ; x < RATE_SIZE ; x++)
                {
                    beatAvg += rates[x];
                }
                beatAvg /= RATE_SIZE;
            }

            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("BPM = ");
            lcd.print(beatsPerMinute);
            Serial.print("IR = ");
            Serial.print(irValue);
            Serial.print(", BPM = ");
            Serial.print(beatsPerMinute);
            Serial.print(", Avg BPM = ");
            Serial.print(beatAvg);
            while (irValue < 50000) {
                lcd.clear();

                Serial.print("No finger");
            }
            Serial.println();
        }
    }
}
}
}

```