جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology
Electrical & Computer Engineering Department

Computer Design Laboratory Project

Smart home automation

Prepared by: ShahEd Jamhour 1180654

Instructor: Dr. Abdallatif Abuissa
Assistant: Eng. Heba Qadah

Section:5
Date: May 2021

Abstract:

The aim of this project is to build a home automation system using 2 Arduinos, a breadboard and the following components.

1.Ultrasonic Sensor.
2.Light Dependent Resistor.
3.Passive Infrared sensor.
4. Temperature Sensor.
5. DC Motor.
6.Bulb.
7.Switch.

Table of Contents

Theory & Background:

This part contains the components and their usage.

1. Ultrasonic sensor

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1” to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module. Figure 1 shows an ultrasonic sensor.

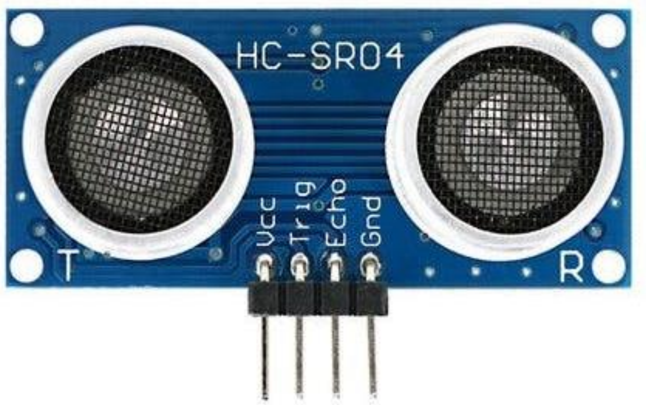


Figure (1)

2. Light Dependent Resistor – LDR

In order to detect the intensity of light or darkness, The LDR is a special type of resistor that allows higher voltages to pass through it (low resistance) whenever there is a high intensity of light, and passes a low voltage (high resistance) whenever it is dark. We can take advantage of this LDR property and use it in this project. Figure 2 shows LDR.

Figure (2)

3. Passive Infrared sensor - PIR

   PIR sensor detects a human being moving around within approximately 10m from the sensor. This is an average value, as the actual detection range is between 5m and 12m.PIR are fundamentally made of a pyro electric sensor, which can detect levels of infrared radiation. For numerous essential projects or items that need to discover when an individual has left or entered the area. PIR sensors are incredible, they are flat control and minimal effort, have a wide lens range, and are simple to interface with. Figure 3 shows PIR.

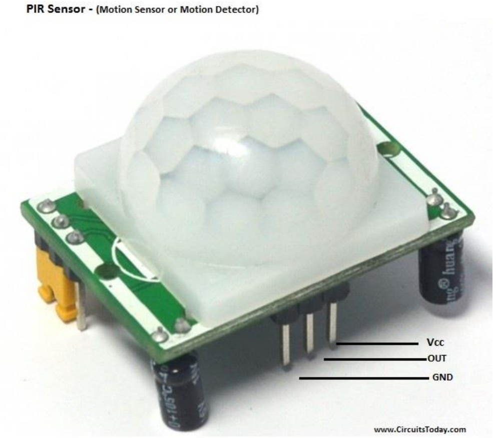


Figure (3)

4. A DC motor (Direct Current motor) is the most common type of motor. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate.

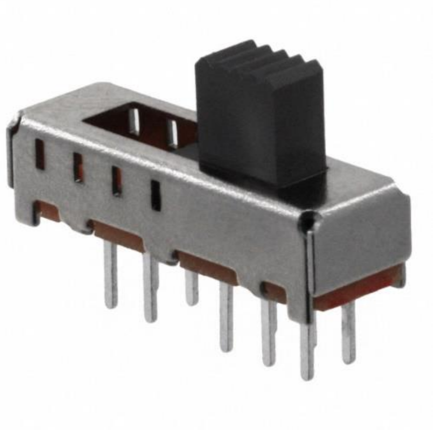
If you switch the leads, the motor will rotate in the opposite direction. Figure 4 shows a DC motor.

Figure (4)

5. Slide Switch

Slide switch is to connect or disconnect the circuit by sliding its switch handle so as to switch the circuit. The common types of slide switch include single pole double throw, single pole triple throw, double pole double throw, and double pole triple throw and so on. Generally, it is used in circuits with a low voltage and features flexibility and stabilization. Slide switches are commonly used in all kinds of instruments/meters equipment, electronic toys and other fields related.

Figure 5 shows a slide switch.



Slide switch (5)

Design and Implementation:

In this project 2 Arduinos were used (As shown in fig 6), the first Arduino will tell the second Arduino, if we are using the manual or automatic mode (By using the switch, if the switch is low , it will work on the automatic mode , if high, it will work on manual mode).

If in Automatic mode the circuit will start working immediately after starting the simulation, but the manual mode requires the user to use the remote in order to complete any actions.

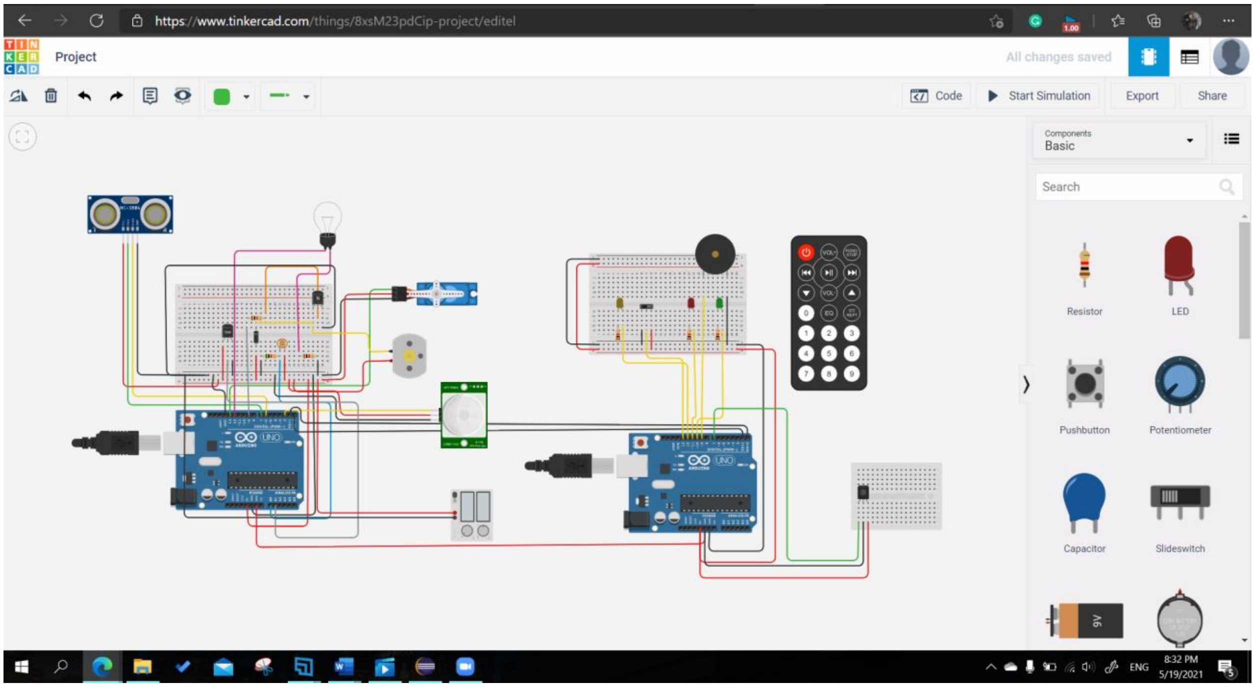
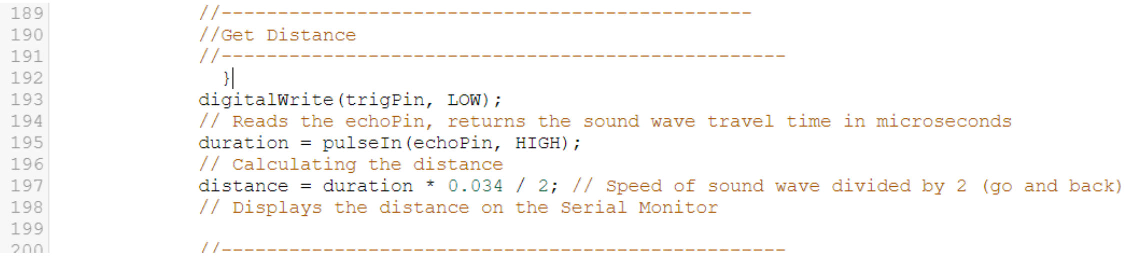


Fig 6

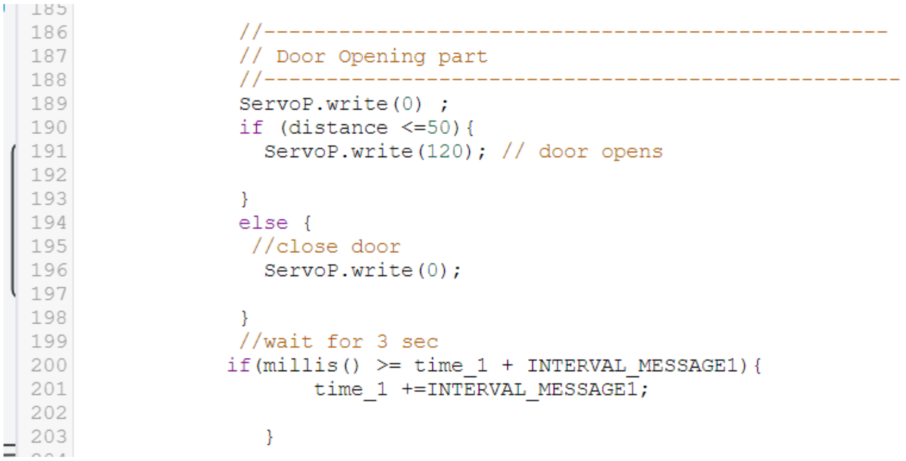
How does the system work?

1. Automatic mode

   a. An Ultrasound Sensor was connected in order to measure the distance, if there is something in distance of 50 cm or less, a door (Servo meter) will open for 3 seconds.

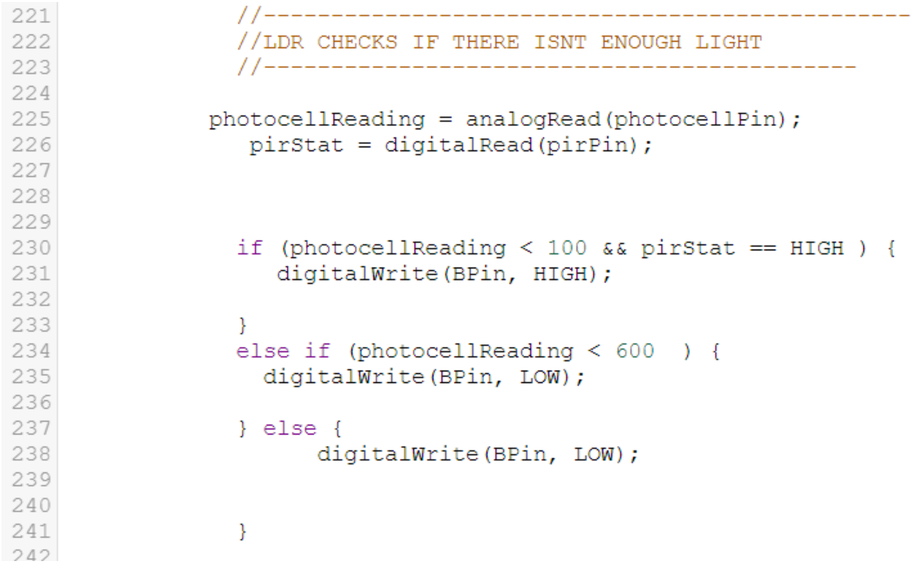


```
                //-----------------------------------------------
                //Get Distance
                //--------------------------------------------------
                  }|
                digitalWrite(trigPin, LOW);
                // Reads the echoPin, returns the sound wave travel 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)
                // Displays the distance on the Serial Monitor

                //--------------------------------------------------
```



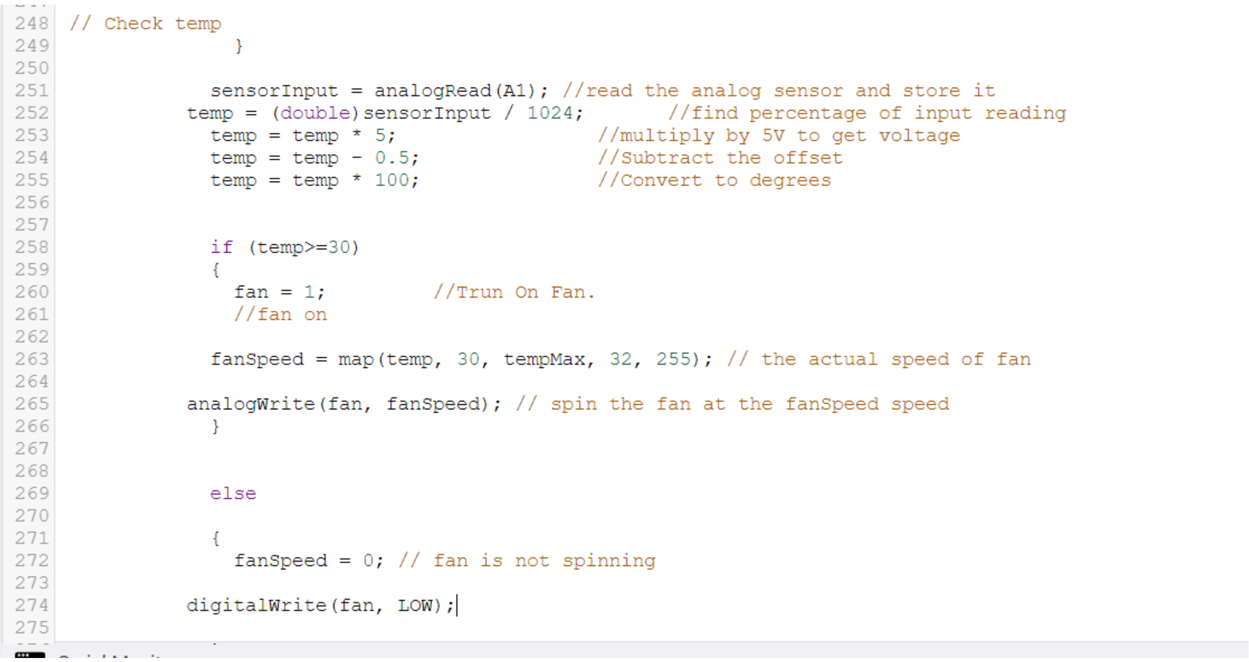
```
            //---------------------------------------------------
            // Door Opening part
            //----------------------------------------------------
            ServoP.write(0) ;
            if (distance <=50){
              ServoP.write(120); // door opens

            }
            else {
             //close door
              ServoP.write(0);

            }
            //wait for 3 sec
           if(millis() >= time_1 + INTERVAL_MESSAGE1){
                   time_1 +=INTERVAL_MESSAGE1;

              }
```

b. A Light Dependent Resistor and PIR were connected in order to detect light and movement. If there is no enough light “and” there is a movement then a bulb will be turned on for 1 minute.



```
            //-------------------------------------------------
            //LDR CHECKS IF THERE ISNT ENOUGH LIGHT
            //---------------------------------------------

          photocellReading = analogRead(photocellPin);
             pirStat = digitalRead(pirPin);



            if (photocellReading < 100 && pirStat == HIGH ) {
               digitalWrite(BPin, HIGH);

            }
            else if (photocellReading < 600  ) {
              digitalWrite(BPin, LOW);

            } else {
                  digitalWrite(BPin, LOW);


            }
```

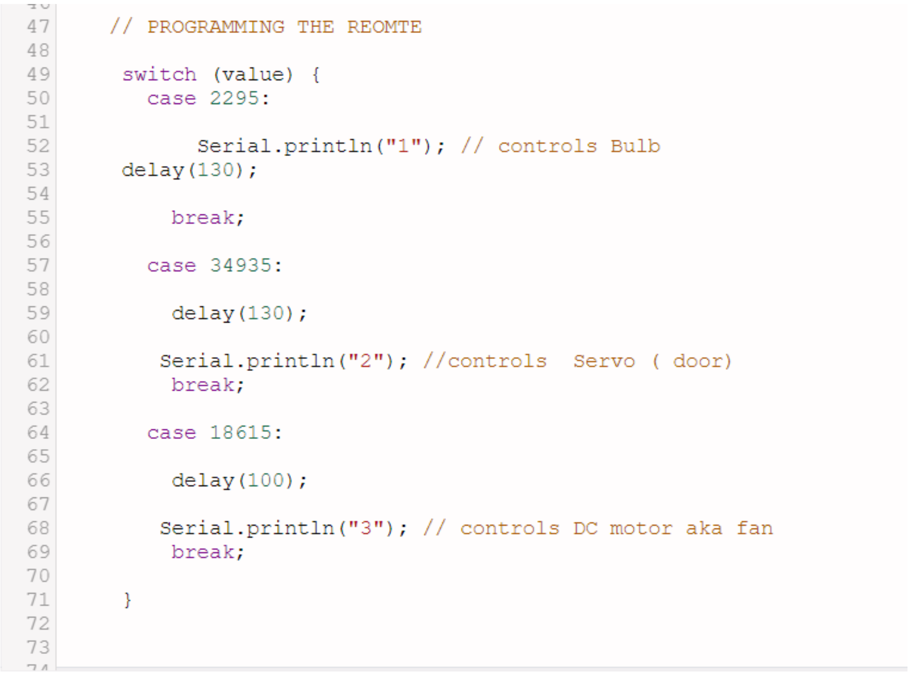
c. Temperature Sensor was connected to measure temperature, if the temperature is more than 30 degree, then a fan (DC Motor) will be turned on, and the fan will go faster as the temperature increases.



```
// Check temp
              }

            sensorInput = analogRead(A1); //read the analog sensor and store it
          temp = (double)sensorInput / 1024;       //find percentage of input reading
            temp = temp * 5;                 //multiply by 5V to get voltage
            temp = temp - 0.5;               //Subtract the offset
            temp = temp * 100;               //Convert to degrees


            if (temp>=30)
            {
              fan = 1;         //Trun On Fan.
              //fan on

            fanSpeed = map(temp, 30, tempMax, 32, 255); // the actual speed of fan

          analogWrite(fan, fanSpeed); // spin the fan at the fanSpeed speed
            }


            else

            {
              fanSpeed = 0; // fan is not spinning

          digitalWrite(fan, LOW);
```

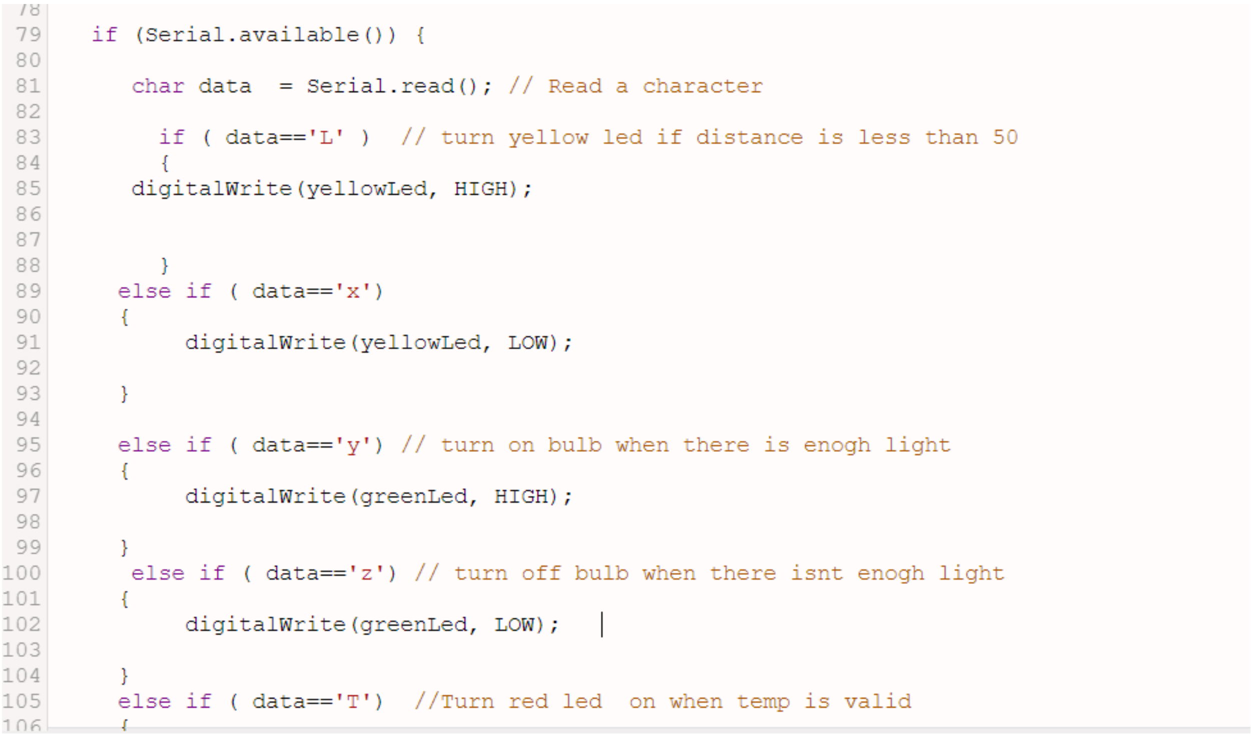
2. Manual mode

   a. As mentioned before if the switch is high it will work in the manual mode.
      LEDS and Buzzers will notify the user if anything needs an action.
      The Arduino#2 will receive different alarms about the situation of Arduino#1, and the user can use the remote control to turn on/off the bulb (By clicking 1), the door (servo motor) (by clicking 2), and the fan (DC motor) (By clicking 3).



```
// PROGRAMMING THE REOMTE

 switch (value) {
   case 2295:

       Serial.println("1"); // controls Bulb
 delay(130);

     break;

   case 34935:

     delay(130);

    Serial.println("2"); //controls  Servo ( door)
     break;

   case 18615:

     delay(100);

    Serial.println("3"); // controls DC motor aka fan
     break;

 }
```

Notification will be sent as follows;
 If the distance was less than 50 yellow led will be on, when the temperature is valid red led is on , and so one ..



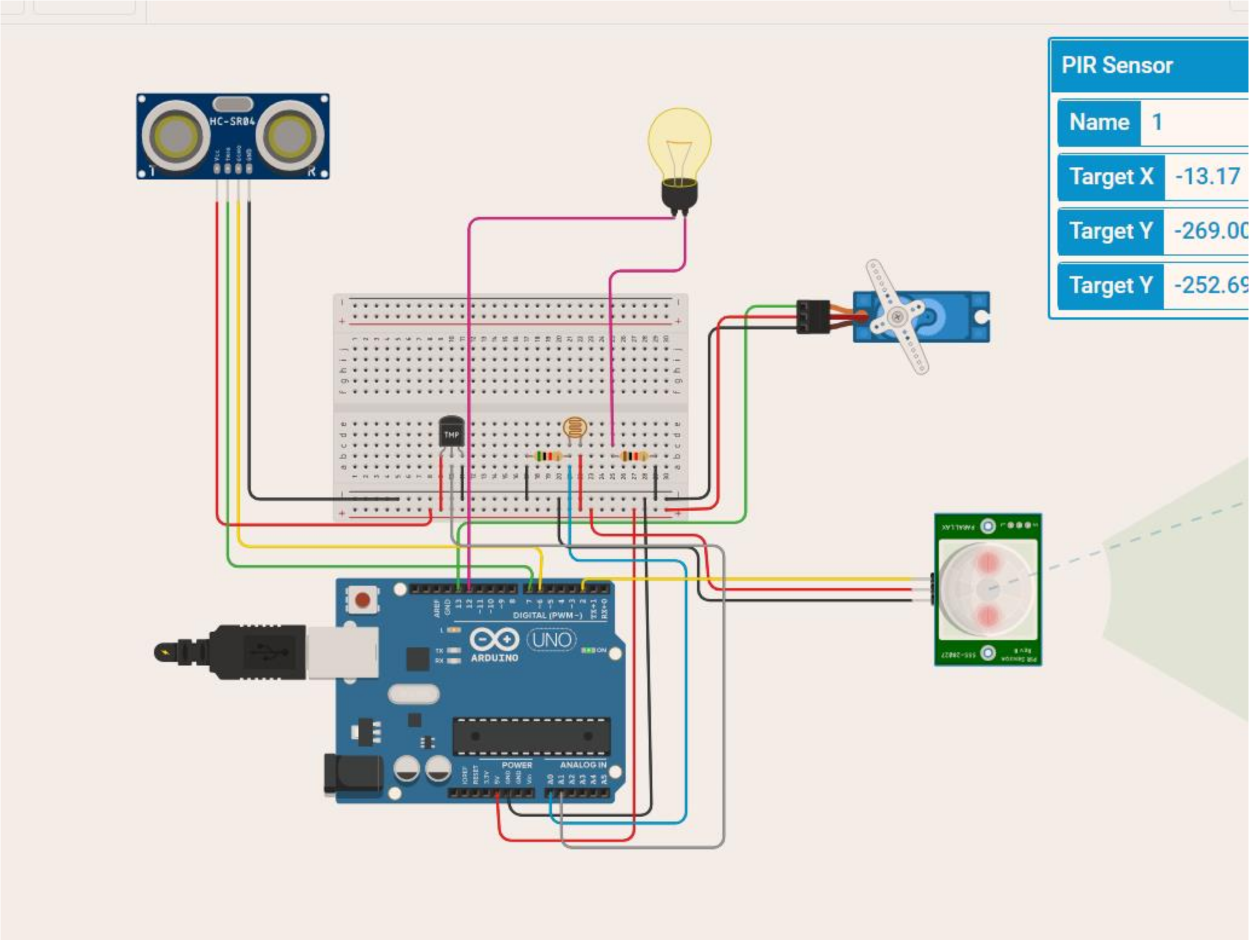
```
  if (Serial.available()) {

     char data  = Serial.read(); // Read a character

       if ( data=='L' )  // turn yellow led if distance is less than 50
       {
     digitalWrite(yellowLed, HIGH);


       }
    else if ( data=='x')
    {
         digitalWrite(yellowLed, LOW);

    }

    else if ( data=='y') // turn on bulb when there is enogh light
    {
         digitalWrite(greenLed, HIGH);

    }
     else if ( data=='z') // turn off bulb when there isnt enogh light
    {
         digitalWrite(greenLed, LOW);

    }
    else if ( data=='T')  //Turn red led  on when temp is valid
    {
```

Testing

A. Automatic mode

- (Switch is low)

- As shown in fig 7 Distance is more than 50 cm, so the door is moving.
- The light is valid and there is a movement so the light bulb is turned on

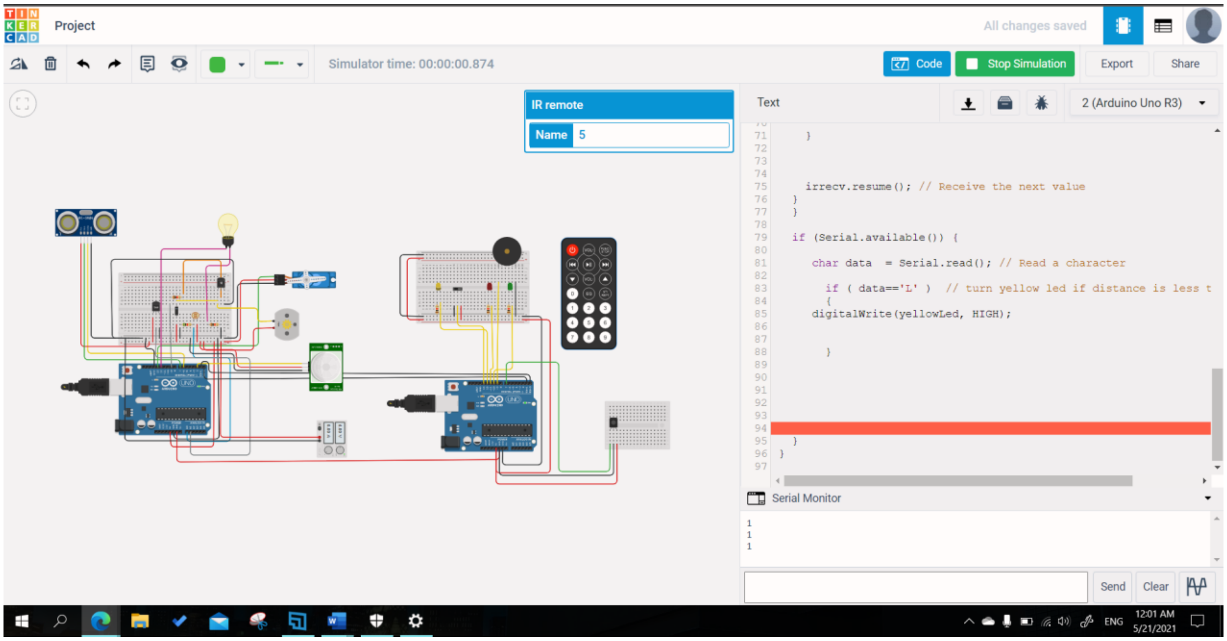


PIR Sensor	
Name	1
Target X	-13.17
Target Y	-269.00
Target Y	-252.69

Fig 7

B.Manual mode

- Switch is high
- If I pressed 1 on the remote it will be printed on the serial as shown in the figure belowand the yellow led .
- The bulb is now on since 1 controls the Bulb.

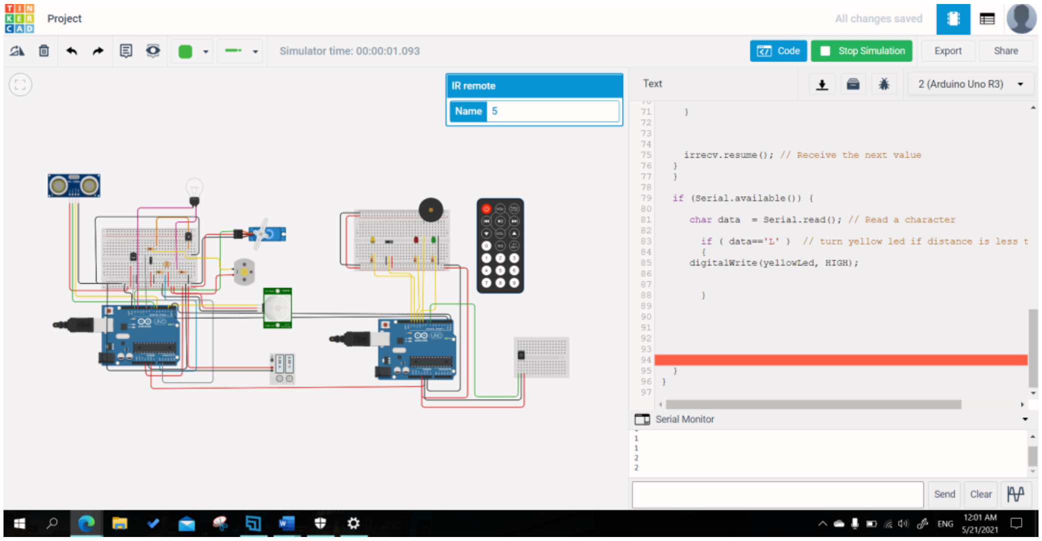


```
      }



    irrecv.resume(); // Receive the next value
  }
  }

  if (Serial.available()) {

     char data  = Serial.read(); // Read a character

       if ( data=='L' )  // turn yellow led if distance is less t
       {
     digitalWrite(yellowLed, HIGH);


       }






  }
}
```

```
1
1
1
```

Fig 8

If 2 is pressed the door will start moving

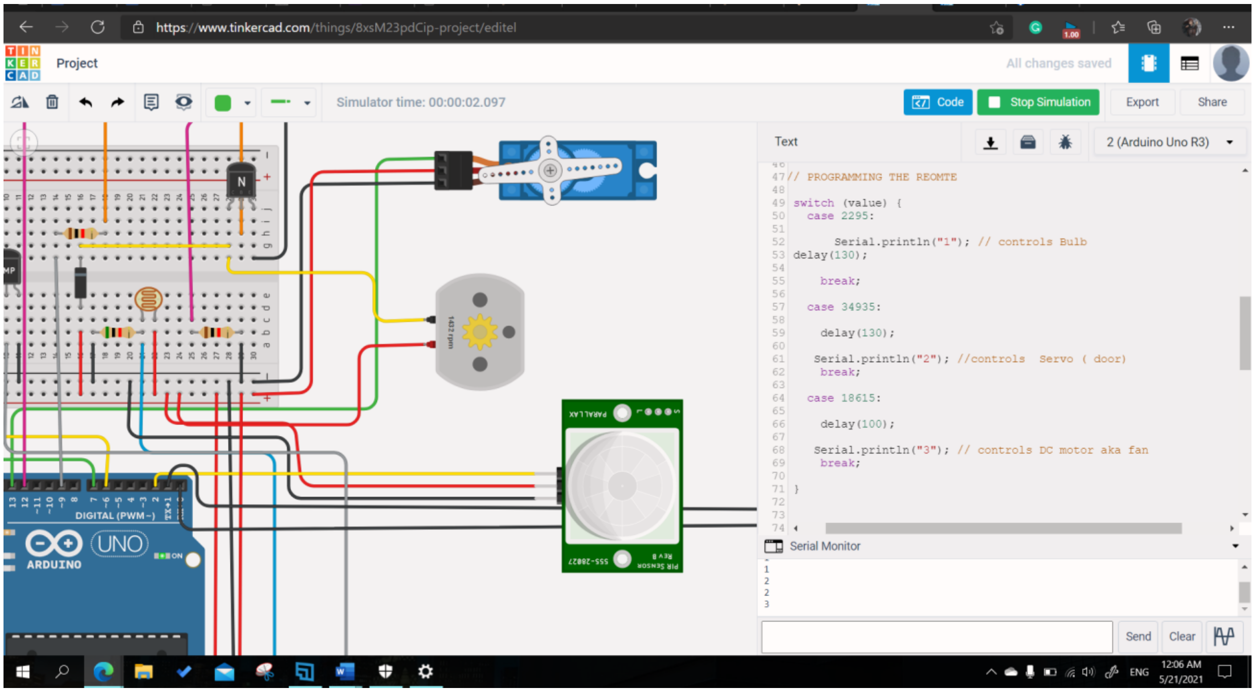


```
      }



    irrecv.resume(); // Receive the next value
  }
  }

  if (Serial.available()) {

     char data  = Serial.read(); // Read a character

       if ( data=='L' )  // turn yellow led if distance is less t
       {
     digitalWrite(yellowLed, HIGH);


       }






  }
}
```

```
1
1
2
2
```

Fig 3

If 3 is pressed then the motor will spin



```
// PROGRAMMING THE REOMTE

 switch (value) {
   case 2295:

       Serial.println("1"); // controls Bulb
 delay(130);

     break;

   case 34935:

     delay(130);

    Serial.println("2"); //controls  Servo ( door)
     break;

   case 18615:

     delay(100);

    Serial.println("3"); // controls DC motor aka fan
     break;

 }
```

```
1
2
2
3
```

Link for Full project

https://www.tinkercad.com/things/8xsM23pdCip-project/editel?sharecode=XH5xfpKZA0vyW1GzZa_ZxtV5fUvkMHOBkLQCuyQH-2Y

Conclusion

In this project I learnt how to build a home automation system using 2 Arduinos and Using the Serial functions.
Also I became more familiar with the following components:

1.Ultrasonic Sensor.
2.Light Dependent Resistor.
3.Passive Infrared sensor.
4. Temperature Sensor.
5. DC Motor.
6.Bulb.
7.Switch.

.

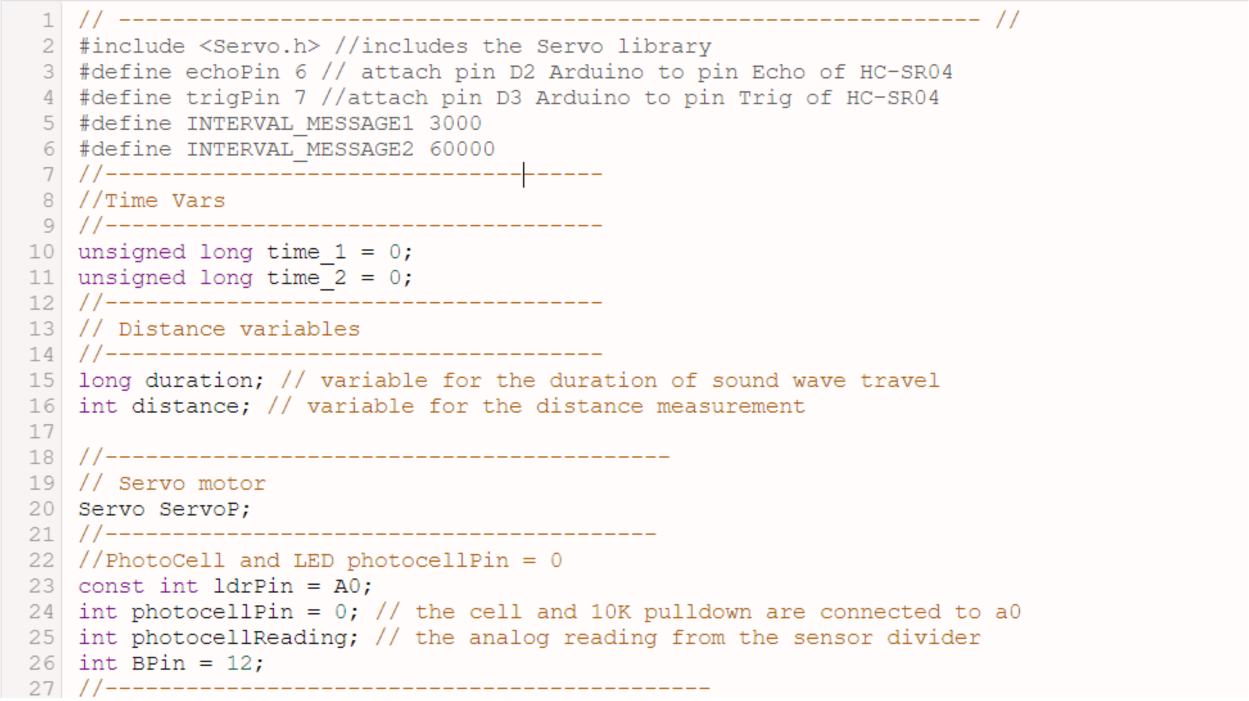
Reverences

[1] Sweep | Arduino accessed on 19-5-2021
[2] Using A TMP36 Temperature Sensor With Arduino - BC Robotics (bc-robotics.com) accessed on 19-5-2021
[3] Arduino - PIR Sensor - Tutorialspoint accessed on 20-5-2021
[4] How to Use an LDR Sensor With Arduino | Arduino | Maker Pro accessed on 20-5-2021
[5] Arduino Infrared Remote Tutorial : 7 Steps - Instructables accessed on 20-5-2021

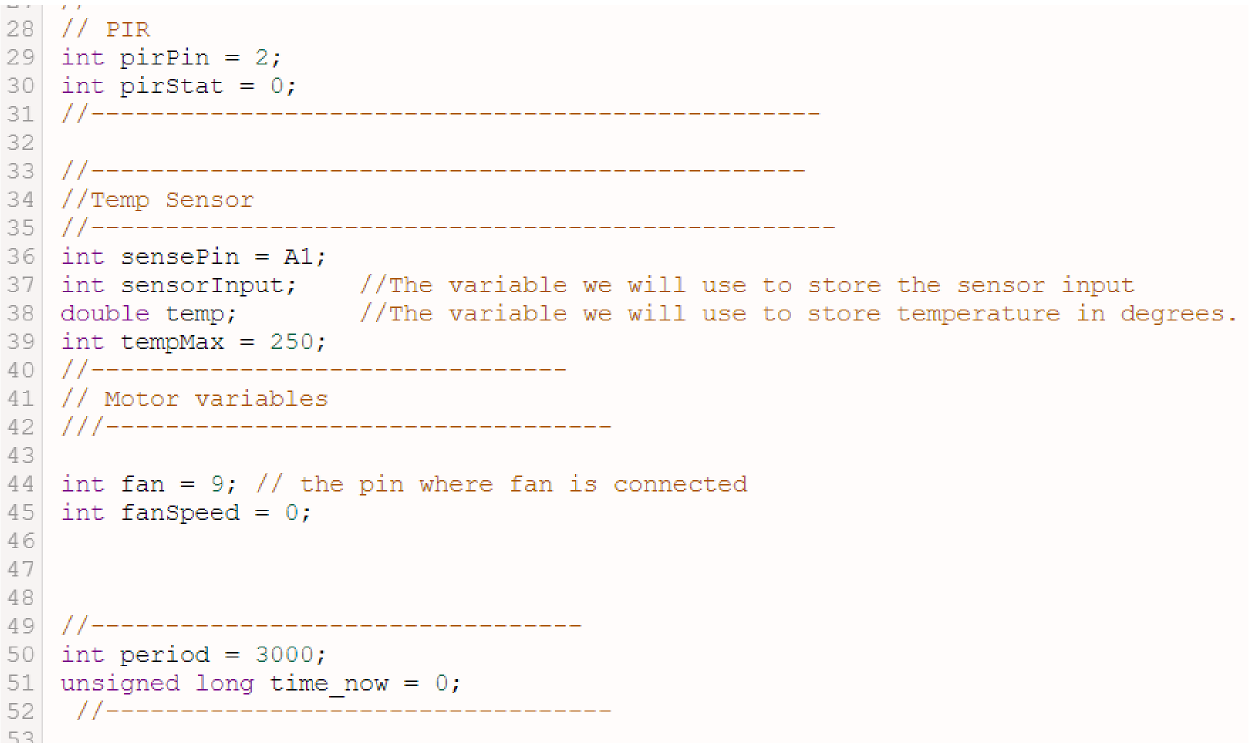
Appendix

Full code

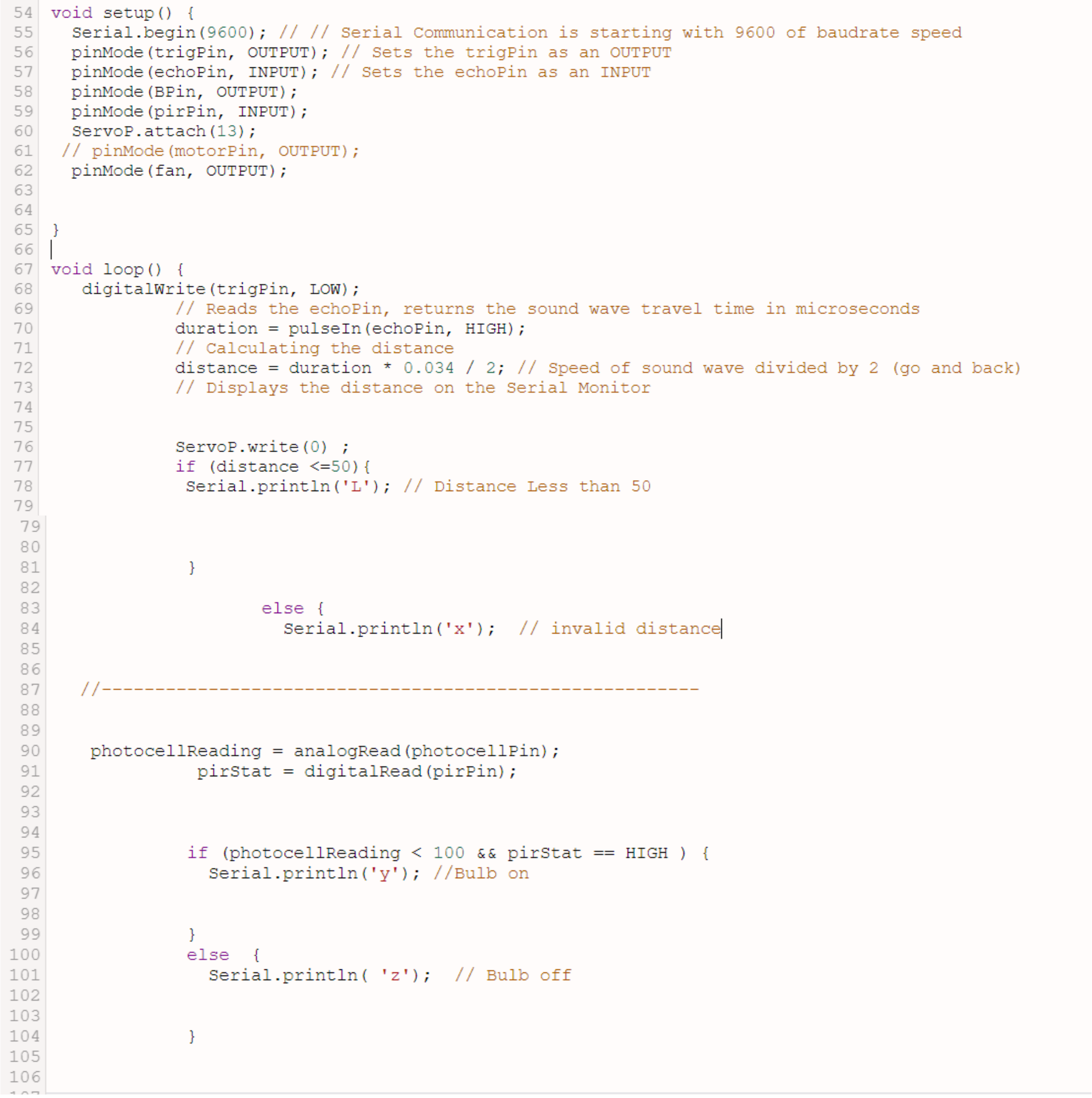
Arduino #1



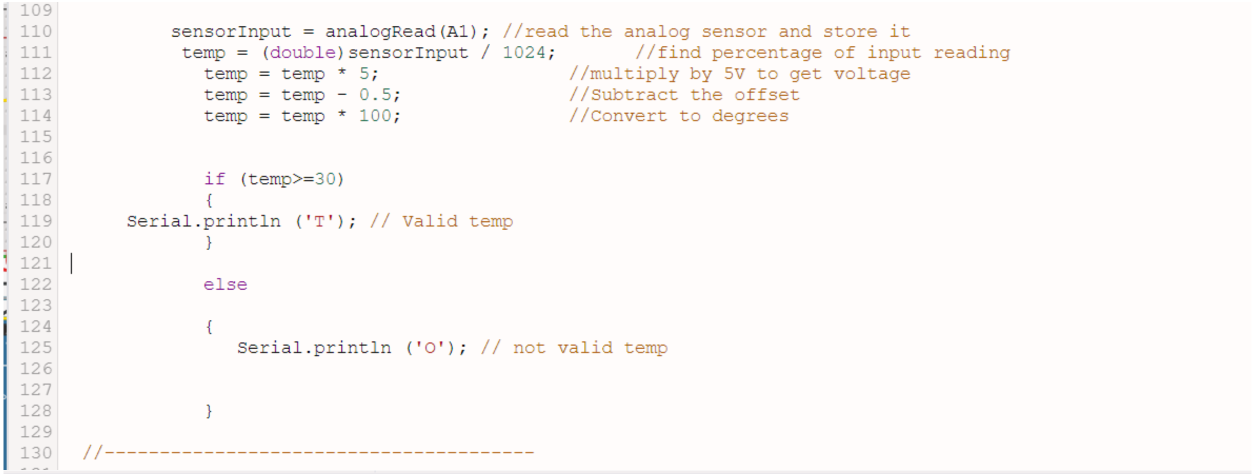
```
// ---------------------------------------------------------------- //
#include <Servo.h> //includes the Servo library
#define echoPin 6 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 7 //attach pin D3 Arduino to pin Trig of HC-SR04
#define INTERVAL_MESSAGE1 3000
#define INTERVAL_MESSAGE2 60000
//-----------------------------------
//Time Vars
//-----------------------------------
unsigned long time_1 = 0;
unsigned long time_2 = 0;
//-----------------------------------
// Distance variables
//-----------------------------------
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement

//----------------------------------------
// Servo motor
Servo ServoP;
//---------------------------------------
//PhotoCell and LED photocellPin = 0
const int ldrPin = A0;
int photocellPin = 0; // the cell and 10K pulldown are connected to a0
int photocellReading; // the analog reading from the sensor divider
int BPin = 12;
//-------------------------------------------
```



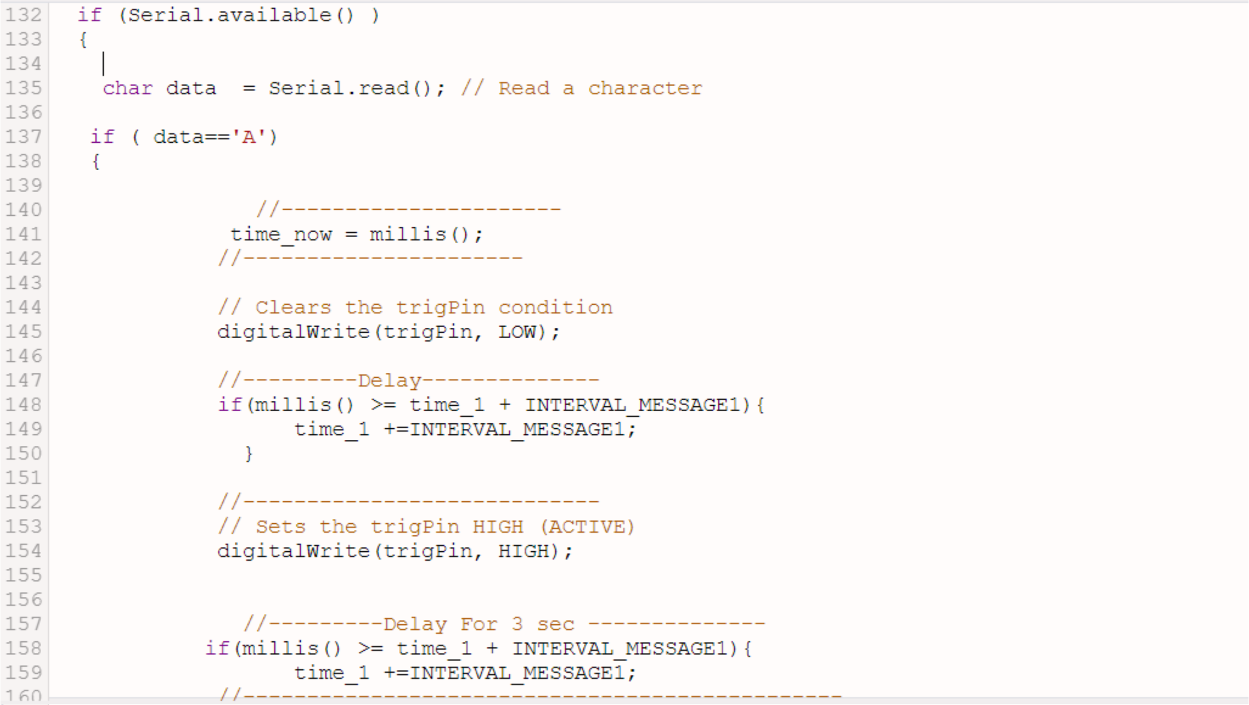
```
//
// PIR
int pirPin = 2;
int pirStat = 0;
//-------------------------------------------------

//------------------------------------------------
//Temp Sensor
//--------------------------------------------------
int sensePin = A1;
int sensorInput;    //The variable we will use to store the sensor input
double temp;        //The variable we will use to store temperature in degrees.
int tempMax = 250;
//---------------------------------
// Motor variables
///----------------------------------


int fan = 9; // the pin where fan is connected
int fanSpeed = 0;



//---------------------------------
int period = 3000;
unsigned long time_now = 0;
 //---------------------------------
```



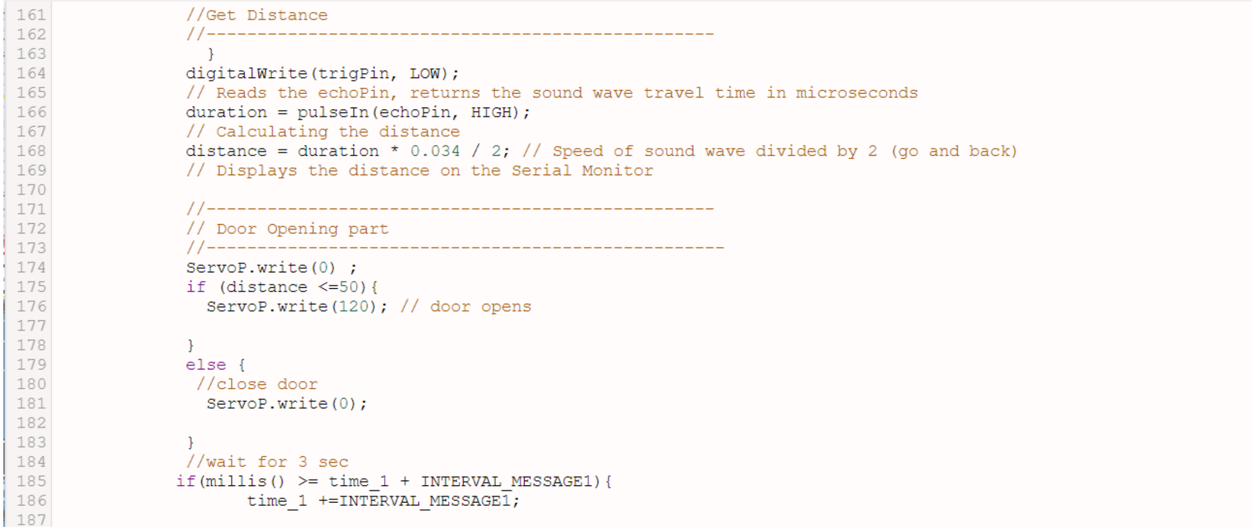
```
void setup() {
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  pinMode(BPin, OUTPUT);
  pinMode(pirPin, INPUT);
  ServoP.attach(13);
 // pinMode(motorPin, OUTPUT);
  pinMode(fan, OUTPUT);


}

void loop() {
   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)
            // Displays the distance on the Serial Monitor


            ServoP.write(0) ;
            if (distance <=50){
             Serial.println('L'); // Distance Less than 50



             }

                    else {
                      Serial.println('x');  // invalid distance


   //---------------------------------------------------------


    photocellReading = analogRead(photocellPin);
              pirStat = digitalRead(pirPin);



             if (photocellReading < 100 && pirStat == HIGH ) {
               Serial.println('y'); //Bulb on


             }
             else  {
               Serial.println( 'z');  // Bulb off


             }


```



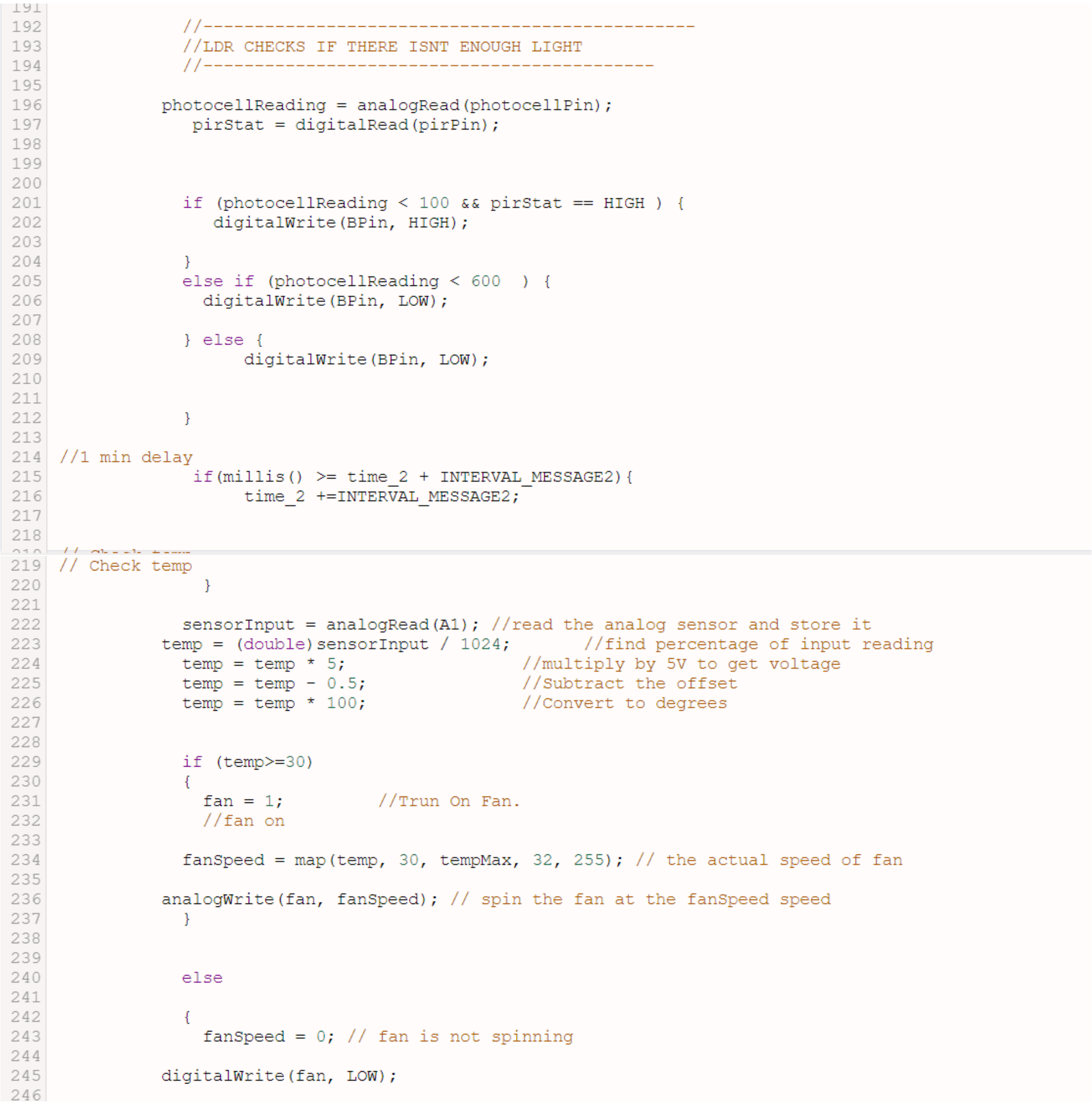
```
         sensorInput = analogRead(A1); //read the analog sensor and store it
          temp = (double)sensorInput / 1024;       //find percentage of input reading
            temp = temp * 5;                //multiply by 5V to get voltage
            temp = temp - 0.5;              //Subtract the offset
            temp = temp * 100;              //Convert to degrees


            if (temp>=30)
            {
     Serial.println ('T'); // Valid temp
            }

            else

            {
               Serial.println ('O'); // not valid temp


            }

 //----------------------------------------
```



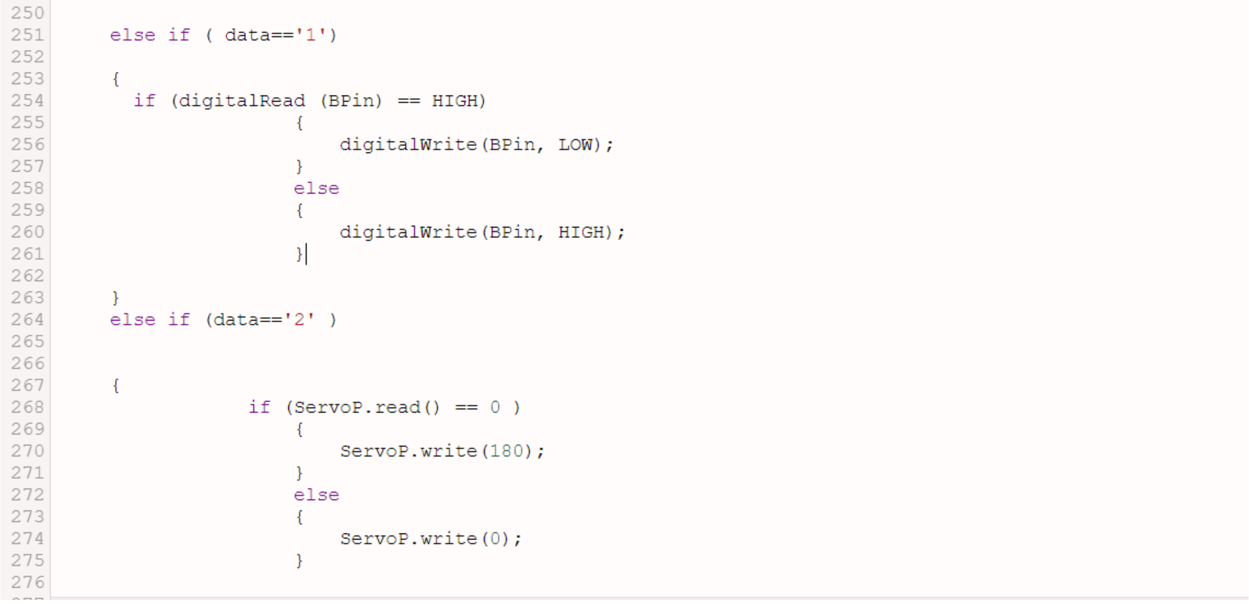
```
 if (Serial.available() )
 {

   char data  = Serial.read(); // Read a character

  if ( data=='A')
  {

                //----------------------
              time_now = millis();
             //----------------------

             // Clears the trigPin condition
             digitalWrite(trigPin, LOW);

             //---------Delay--------------
             if(millis() >= time_1 + INTERVAL_MESSAGE1){
                   time_1 +=INTERVAL_MESSAGE1;
               }

             //----------------------------
             // Sets the trigPin HIGH (ACTIVE)
             digitalWrite(trigPin, HIGH);


               //---------Delay For 3 sec --------------
            if(millis() >= time_1 + INTERVAL_MESSAGE1){
                   time_1 +=INTERVAL_MESSAGE1;
             //---------------------------------------------
```



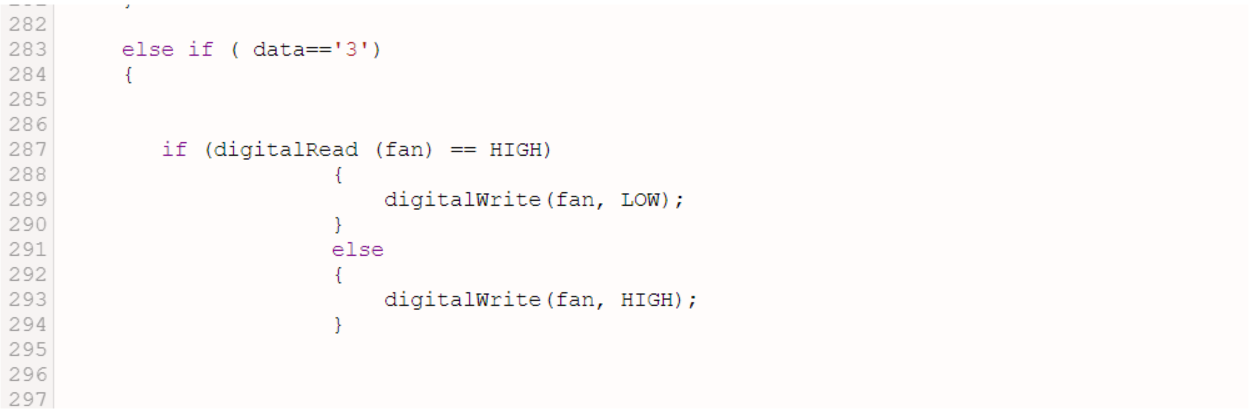
```
             //Get Distance
             //---------------------------------------------------
               }
             digitalWrite(trigPin, LOW);
             // Reads the echoPin, returns the sound wave travel time in microseconds
             duration = pulseIn(echoPin, HIGH);
             // Calculating the distance
             distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
             // Displays the distance on the Serial Monitor

             //---------------------------------------------------
             // Door Opening part
             //----------------------------------------------------
             ServoP.write(0) ;
             if (distance <=50){
               ServoP.write(120); // door opens

             }
             else {
              //close door
               ServoP.write(0);

             }
             //wait for 3 sec
            if(millis() >= time_1 + INTERVAL_MESSAGE1){
                   time_1 +=INTERVAL_MESSAGE1;

```



```
            //------------------------------------------------
            //LDR CHECKS IF THERE ISNT ENOUGH LIGHT
            //--------------------------------------------

          photocellReading = analogRead(photocellPin);
             pirStat = digitalRead(pirPin);


            if (photocellReading < 100 && pirStat == HIGH ) {
               digitalWrite(BPin, HIGH);

            }
            else if (photocellReading < 600  ) {
              digitalWrite(BPin, LOW);

            } else {
                  digitalWrite(BPin, LOW);


            }

//1 min delay
             if(millis() >= time_2 + INTERVAL_MESSAGE2){
                  time_2 +=INTERVAL_MESSAGE2;

// Check temp
              }

            sensorInput = analogRead(A1); //read the analog sensor and store it
          temp = (double)sensorInput / 1024;       //find percentage of input reading
            temp = temp * 5;                 //multiply by 5V to get voltage
            temp = temp - 0.5;               //Subtract the offset
            temp = temp * 100;               //Convert to degrees


            if (temp>=30)
            {
              fan = 1;         //Trun On Fan.
              //fan on

            fanSpeed = map(temp, 30, tempMax, 32, 255); // the actual speed of fan

          analogWrite(fan, fanSpeed); // spin the fan at the fanSpeed speed
            }


            else

            {
              fanSpeed = 0; // fan is not spinning

          digitalWrite(fan, LOW);
```

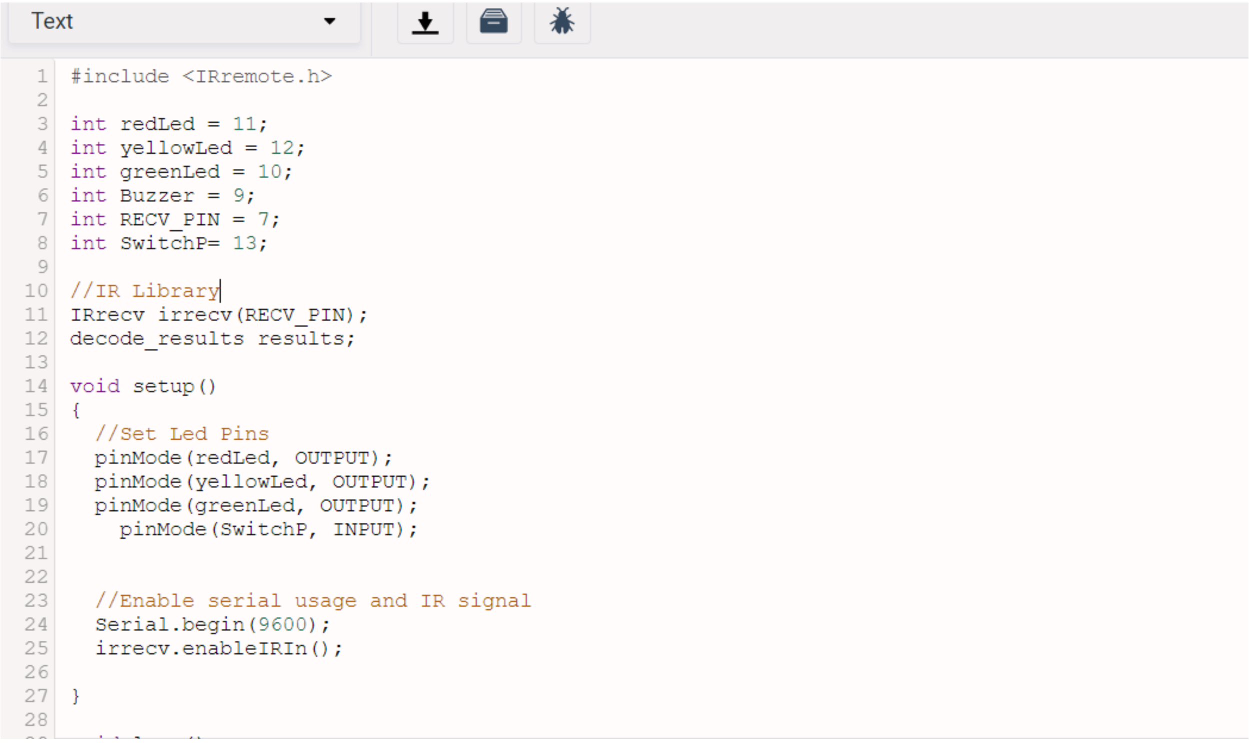


```

else if ( data=='1')

{
  if (digitalRead (BPin) == HIGH)
                {
                    digitalWrite(BPin, LOW);
                }
                else
                {
                    digitalWrite(BPin, HIGH);
                }

}
else if (data=='2' )


{
            if (ServoP.read() == 0 )
                {
                    ServoP.write(180);
                }
                else
                {
                    ServoP.write(0);
                }

```

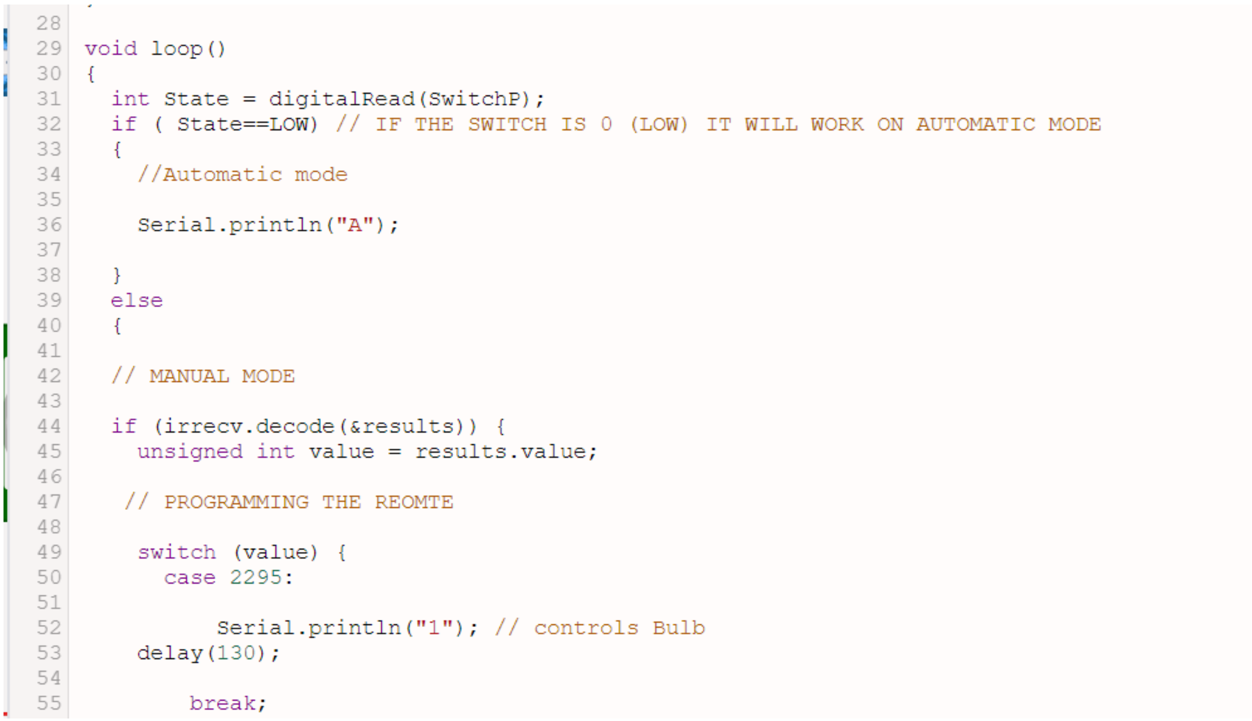


```

else if ( data=='3')
{


   if (digitalRead (fan) == HIGH)
                {
                    digitalWrite(fan, LOW);
                }
                else
                {
                    digitalWrite(fan, HIGH);
                }



```

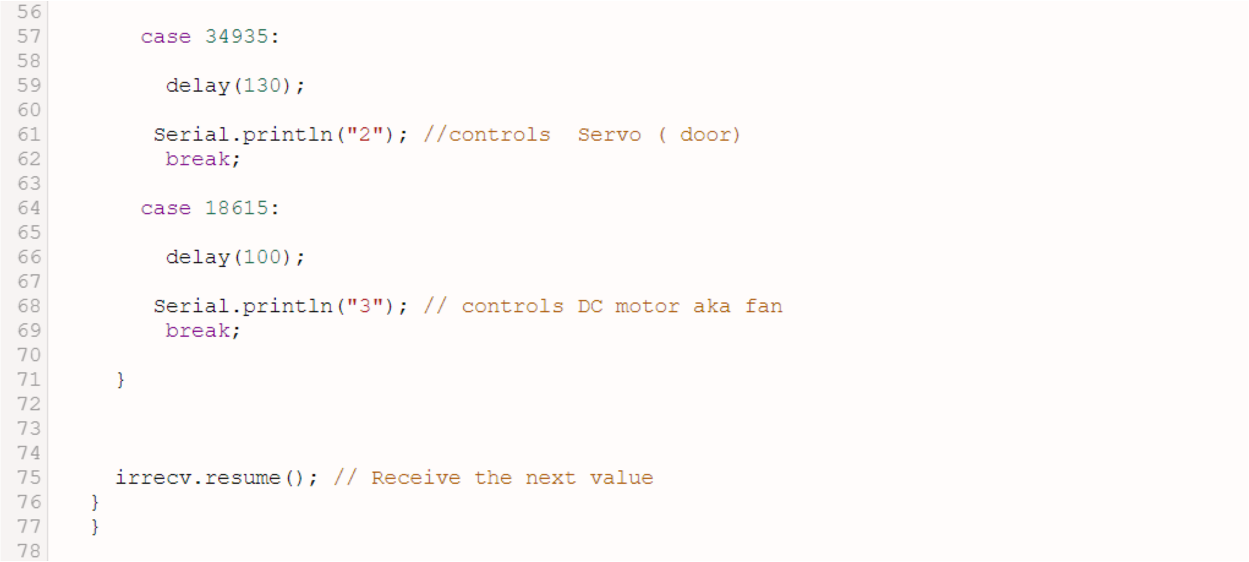
Arduino 2



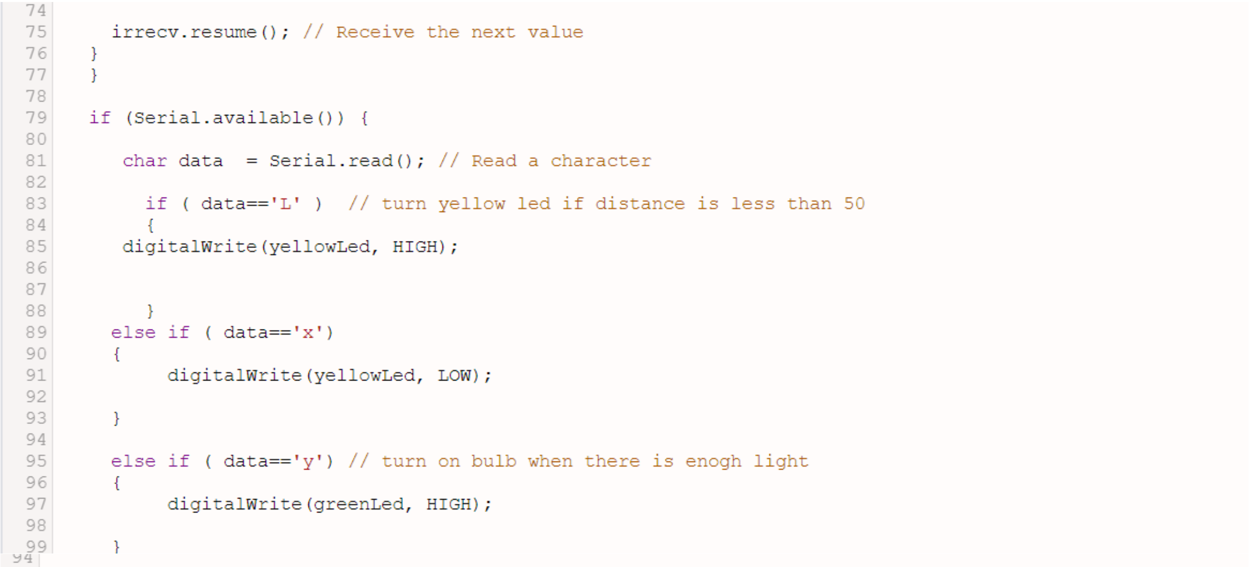
```
#include <IRremote.h>

int redLed = 11;
int yellowLed = 12;
int greenLed = 10;
int Buzzer = 9;
int RECV_PIN = 7;
int SwitchP= 13;

//IR Library
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup()
{
  //Set Led Pins
  pinMode(redLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
    pinMode(SwitchP, INPUT);


  //Enable serial usage and IR signal
  Serial.begin(9600);
  irrecv.enableIRIn();

}

```



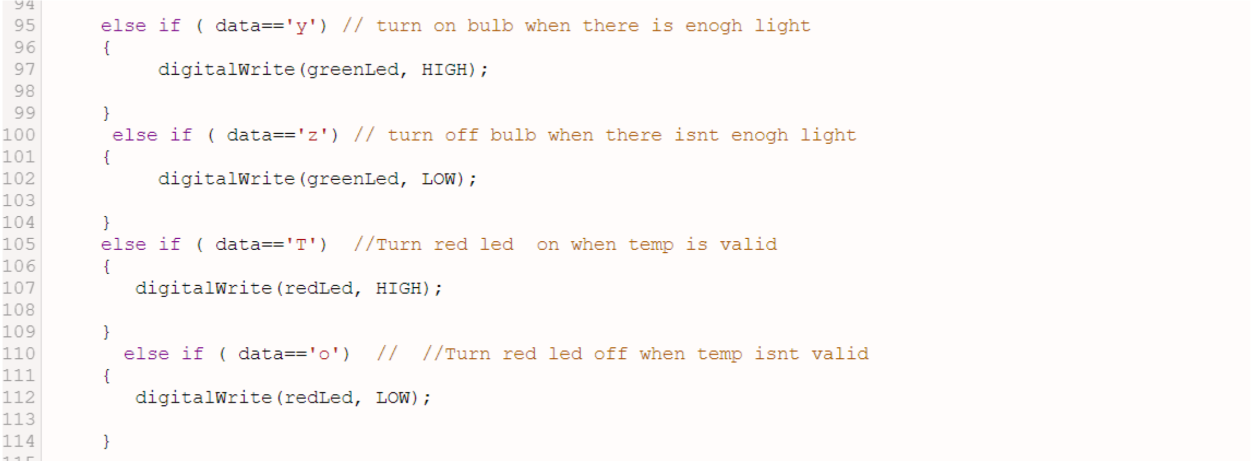
```

void loop()
{
  int State = digitalRead(SwitchP);
  if ( State==LOW) // IF THE SWITCH IS 0 (LOW) IT WILL WORK ON AUTOMATIC MODE
  {
    //Automatic mode

    Serial.println("A");

  }
  else
  {

  // MANUAL MODE

  if (irrecv.decode(&results)) {
    unsigned int value = results.value;

   // PROGRAMMING THE REOMTE

    switch (value) {
      case 2295:

          Serial.println("1"); // controls Bulb
    delay(130);

        break;
```



```
      case 34935:

        delay(130);

       Serial.println("2"); //controls  Servo ( door)
        break;

      case 18615:

        delay(100);

       Serial.println("3"); // controls DC motor aka fan
        break;

    }



    irrecv.resume(); // Receive the next value
  }
  }
```



```
    irrecv.resume(); // Receive the next value
  }
  }

  if (Serial.available()) {

     char data  = Serial.read(); // Read a character

       if ( data=='L' )  // turn yellow led if distance is less than 50
       {
     digitalWrite(yellowLed, HIGH);


       }
    else if ( data=='x')
    {
         digitalWrite(yellowLed, LOW);

    }

    else if ( data=='y') // turn on bulb when there is enogh light
    {
         digitalWrite(greenLed, HIGH);

    }
```



```
    else if ( data=='y') // turn on bulb when there is enogh light
    {
         digitalWrite(greenLed, HIGH);

    }
     else if ( data=='z') // turn off bulb when there isnt enogh light
    {
         digitalWrite(greenLed, LOW);

    }
    else if ( data=='T')  //Turn red led  on when temp is valid
    {
       digitalWrite(redLed, HIGH);

    }
      else if ( data=='o')  //  //Turn red led off when temp isnt valid
    {
       digitalWrite(redLed, LOW);

    }
```