Smart Home System For Saving Electricity

Project Report

Internet of Things

BTech ICT (Sem VII)

Yug Patel (1741017)

Aditya Bajaj (1741028)

Shivansh Nautiyal (1741037)

Nisarg Gandhi (1741099)

Under the guidance of

Instructor Name

Anurag Lakhlani

School of Engineering and Applied Science

Ahmedabad University

Ahmedabad - 380009

CHAPTER_1

INTRODUCTION

1.1 OVERVIEW

The project deals with an interesting manner of how energy can be saved by just turning instruments of room off when not in used.

1.2 MOTIVATION

In India most of the people forgets to switch off fans and lights when they leave the room which cause a big loss of energy. Currently India Faces a huge Problem of Electricity Shortage And there are many Villages in India which still Doesn't have electricity. So, saving Electricity is a huge task and very important for a developing Country like India.

1.3 OBJECTIVE

- Our main objective is to control lights, fans and A/C's on the basis of human presence.

- We will be using IR sensors to count humans inside the room.

- As the person moves inside the lights will be controlled in that way.

- And as the temperature changes fans and A/C's will be operating accordingly.

- Also, we have 4x4 Keypad for Entering the Password. And Servo Motor for Opening the Door.

CHAPTER 2

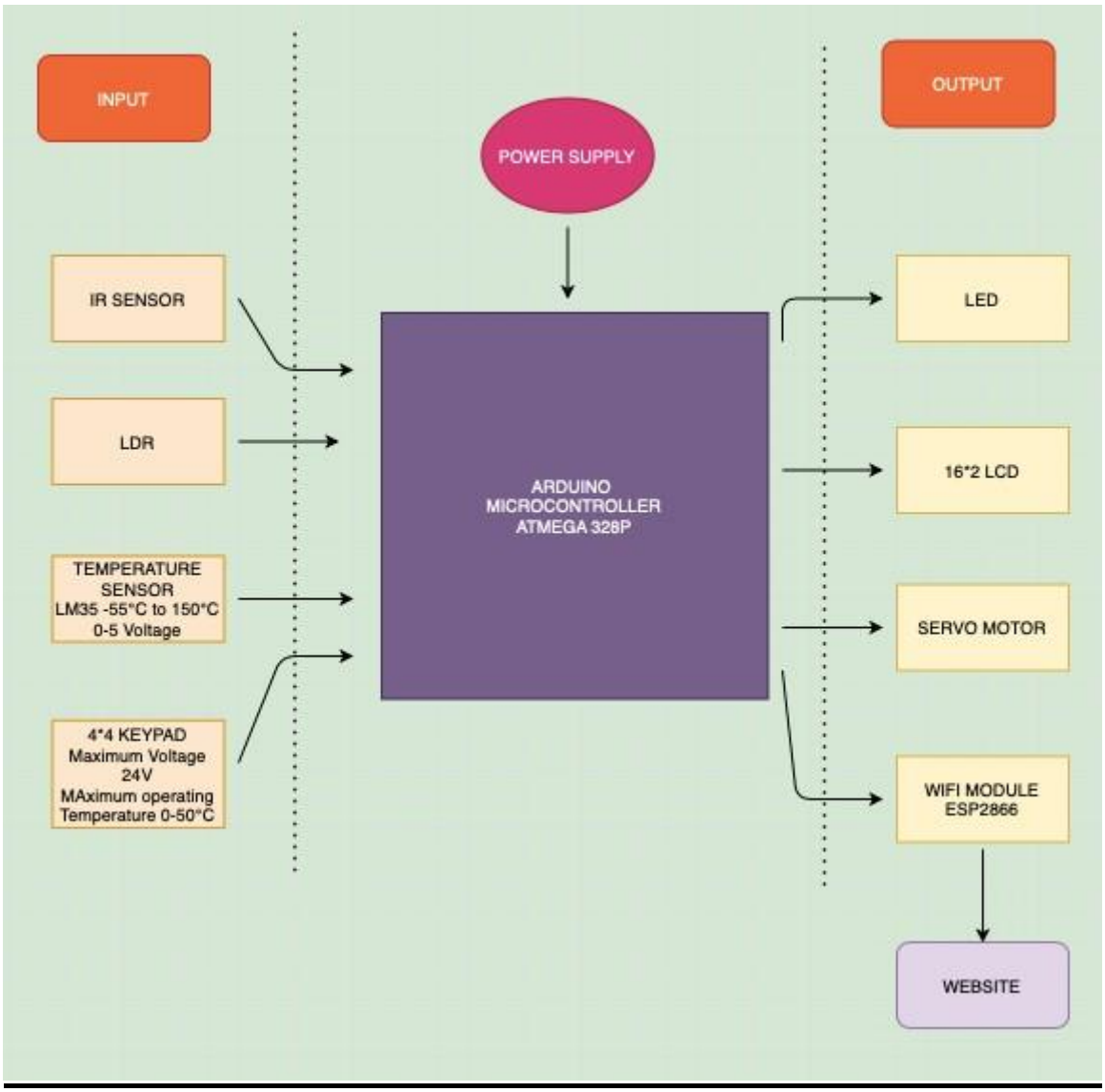
MARKET SURVEY

Sr No.	System	Communication Interface	Controller	User Interface	Applications	Benefits

1.	Philips Hue Bridge	Zig bee wireless Network	Home gateway and Router	Smart device	Monitoring and Controlling Home Appliances	Effectively Manages and Controls Home Appliances and Other devices
2.	LIFX Lighting Systems	Wi-Fi Module	Works on Wi-Fi System.	Led Lights	Switching LED	Smart, Economic and Efficient
3.	Comfy Lights	Wi-Fi Network	Build in Sensors (Motion Sensor, LED, Wi-Fi, Light Sensor).	Android Phone	Comfy Light simulates Human presence at home realistically through light, so deterring intruders.	Convenience, safety, and Power-saving

CHAPTER 3

BLOCK DIAGRAM



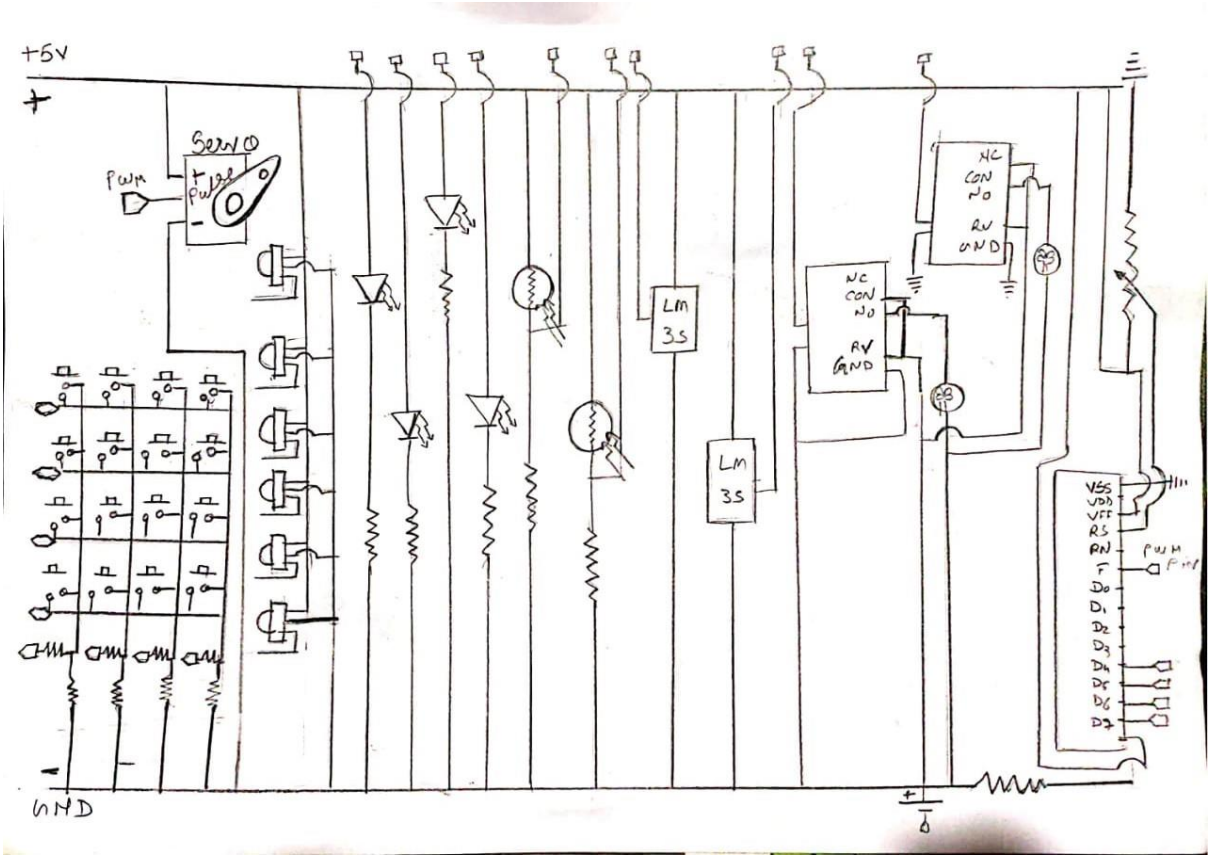
- First, Person will Enter the Password in 4X4 KeyPad. If the password is Correct the door (Servo Motor) will open (Rotate 90 Degree).
- After That when person will pass through the Door, IR Sensor will detect it and display the Count of No of Persons in the Room on 16X2 LCD.
- The Room will be divided in 2 Parts, Left & Right and there will be an IR Sensor in each part which will detect presence of person in each part.
- In Each part there will be a LM35 and a LDR to check the temperature and light intensity.

- Fan and A/c's will be Controlled and turned ON and OFF on the bases of temperature readings provided by LM35.
- Lights will be Controlled and turned ON and OFF on the bases of Natural Light Intensity readings provided by LDR.
- All this data will be available to the user on his mobile phone. This data will be transferred using internet by ESP8266WiFi Module.
- Input for the System will be password on the 4X4 KeyPad, The Human Presence detected IR Sensors, Natural Light Intensity detected by LDR, Temperature Readings Taken by LM35.
- Output for the System will be Counts of No of Humans & Current Temperature on 16X2 LCD, Fans, AC's and Lights as LED's and Servo Motor as door.

CHAPTER 4

SYSTEM ARCHITECTURE

CIRCUIT DIAGRAM



HARDWARE AND SOFTWARE REQUIREMENTS

● Hardware Components

1. ARDUINO MEGA
2. 16x2 CHARACTER LCD
3. IR SENSOR
4. LDR
5. LM35 TEMPERATURE SENSOR
6. 12V FAN
7. SERVO MOTOR
8. 4x4 KEYPAD

9. RELAY DRIVER
10. RESISTORS
11. LEDS
12. JUMPER WIRES
13. BREAD BOARD

● Software Requirements

1. Embedded C programming
2. ARDUINO
3. TinkerCad

- There Are Several Inputs and Several Outputs in the System
- Inputs: IR Sensors, LM35, LDR, 4X4 KeyPad.
- Outputs: 16X2 Lcd, Led, Servo Motor.
- Selection Criteria:
  A) LM35: Minimum and Maximum Input Voltage is 35V and -2V respectively. Typically, 5V. Can measure temperature ranging from -55°C to 150°C. Output voltage is directly proportional (Linear) to temperature (i.e.) there will be a rise of 10mV (0.01V) for every 1°C rise in temperature. ±0.5°C Accuracy. Drain current is less than 60uA. Low cost temperature sensor. Small and hence suitable for remote application. Available in TO-92, TO-220, TO-CAN and SOIC package.
  B) LDR: Can be used to sense Light. Easy to use on Breadboard or Perf Board. Easy to use with

Microcontrollers or even with normal Digital/Analog IC. Small, cheap and easily available. Available in PG5, PG5-MP, PG12, PG12-MP, PG20 and PG20-MP series.

C) IR Sensor: IR sensors read moving objects. Contact-based temperature sensors do not work well on moving objects. Infrared temperature sensors are ideally suited for measuring the temperatures of tires, brakes and similar devices. IR sensors don't wear. No contact means no friction. Infrared sensors experience no wear and tear, and consequently have longer operating lives. IR sensors can provide more detail. An IR sensor can provide greater detail during a measurement than contact devices, simply by pointing it at different spots on the object being read. IR sensors can be used to detect motion by measuring fluctuations in temperature in the field of view.

D) 4X4 KeyPad: Maximum Voltage across EACH SEGMENT or BUTTON: 24V. Maximum Current through EACH SEGMENT or BUTTON: 30mA. Maximum operating temperature: 0°C to + 50°C. Ultra-thin design. Adhesive backing. Easy interface. Long life.

E) 16X2 LCD: Operating Voltage is 4.7V to 5.3V. Current consumption is 1mA without backlight. Alphanumeric LCD display module, meaning can display alphabets and numbers. Consists of two rows and each row can print 16 characters. Each character is built by a 5×8-pixel box. Can work on both 8-bit and 4-bit mode. It can also display any custom generated characters. Available in Green and Blue Backlight

F) Servo Motor: The servo motor is specialized for high-response, high-precision positioning. As a motor capable

of accurate rotation angle and speed control, it can be used for a variety of equipment.

CHAPTER 5

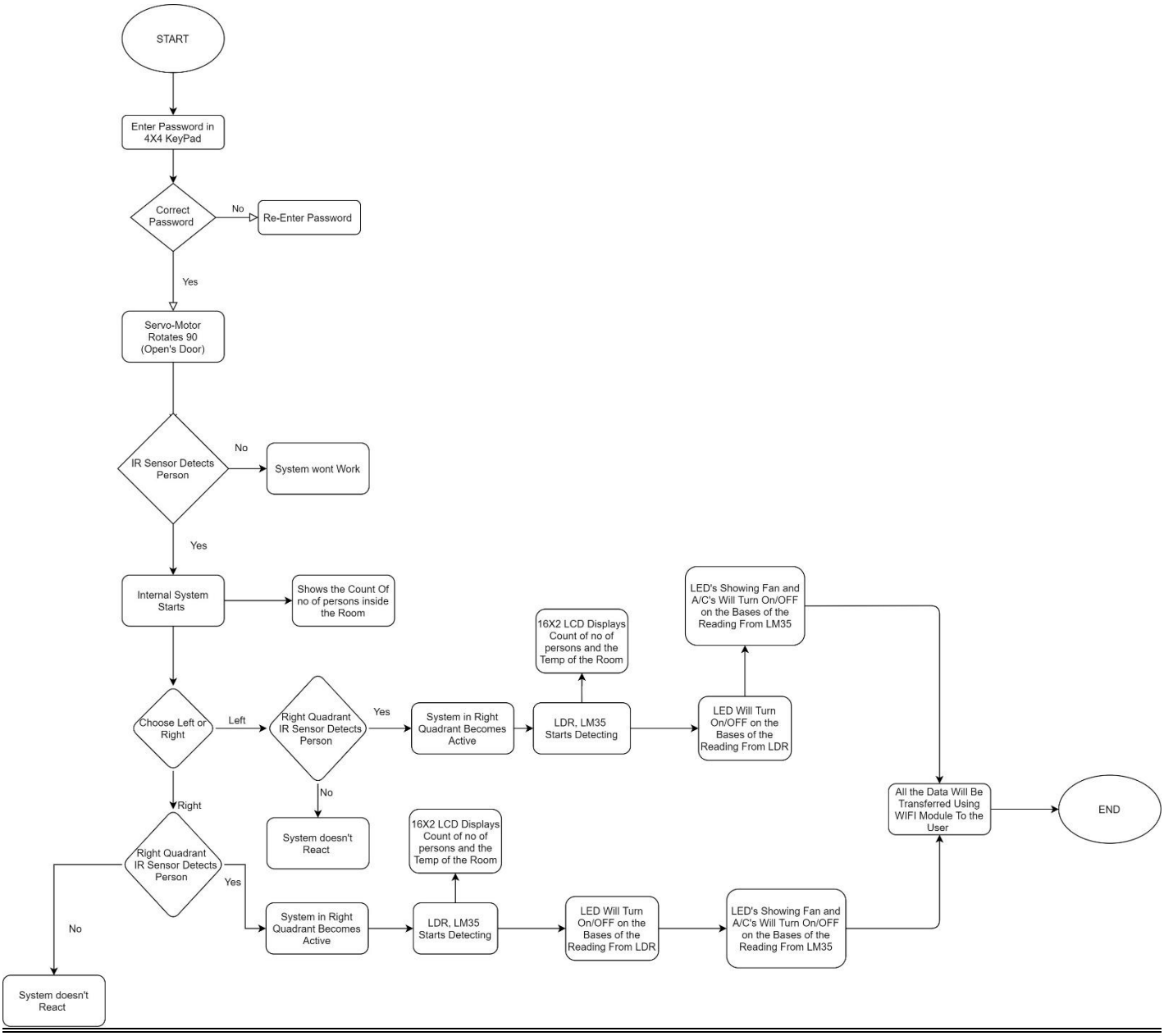
MICROCONTROLLER FEATURES

Microcontroller / Features	Arduino UNO	Raspberry Pi	Beagle Bone	Intel's Galileo	Intel's Edison
Microprocessor	ATmega 32 8	1.2 GHz 64-bit quad-core ARM Cortex-A53	1GHz AM335x ARM® Cortex-A 8	Intel Quark SoC X1000 Application Processor	Intel Atom 500MHz dual-core, dual-threaded CPU and an Intel Quark 100MHz microcontroller
Clock Speed	16MHz	700 MHz to 1.4 GHz	300 MHz to 1 GHz	400 MHz	500 MHz
Operating Voltage	7 to 12 V	5.1 V	5 V	3.3V or 5V	1.8V
Flash Memory	32 KB	4 GB	4 GB/256 MB	8M	
SRAM	2 KB			512 KB	192 KB
EEPROM	1 KB			8 KB	
Digital I/O pins	14	26	92	14	20
Analog input pins	6	-	8	6	6

Programming Languages	C, C++	Python, C, C++, Java, Scratch, and Ruby	C, C++, Python, Perl, Ruby, Java, or even a shell script	C, C++, Python, Node.js/ JavaScript	Python, Node.JS, C/C++

CHAPTER 6

FLOW CHART



CHAPTER 7

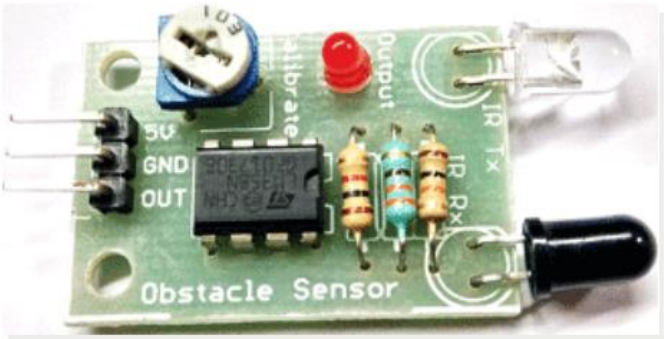
SENSOR'S DETAILS

IR Sensor

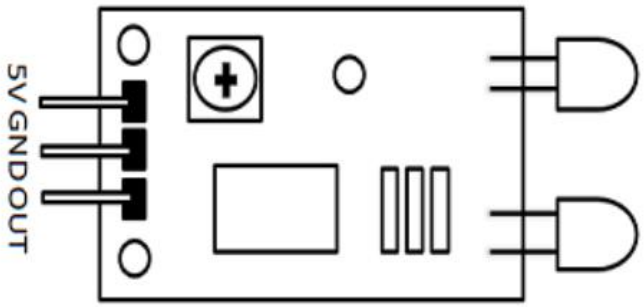
1. **Operating Principle with Diagram**

   An infrared sensor is an electronic instrument that is used to sense certain characteristics of its surroundings. It does this by either emitting or detecting infrared radiation. Infrared sensors are also capable of measuring the heat being emitted by an object and detecting motion.

   Infrared technology is found not just in industry, but also in every-day life. Televisions, for example, use an infrared detector to interpret the signals sent from a remote control. Passive Infrared sensors are used for motion detection systems, and LDR sensors are used for outdoor lighting systems. The key benefits of infrared sensors include their low power requirements, their simple circuitry and their portable features.



IR Sensor Module



IR Sensor Module Pinout

2. **Measurement Range**

   The Range of IR Sensor is up to 20cm.

3. **Physical Dimension**

   Size: 50 x 20 x 10 mm (L x B x H)

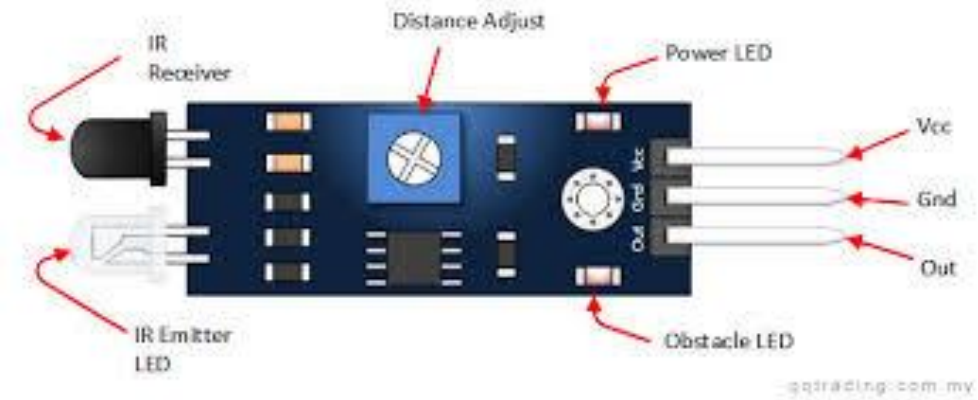
   Hole size: φ2.5mm

4. **Power rating details**

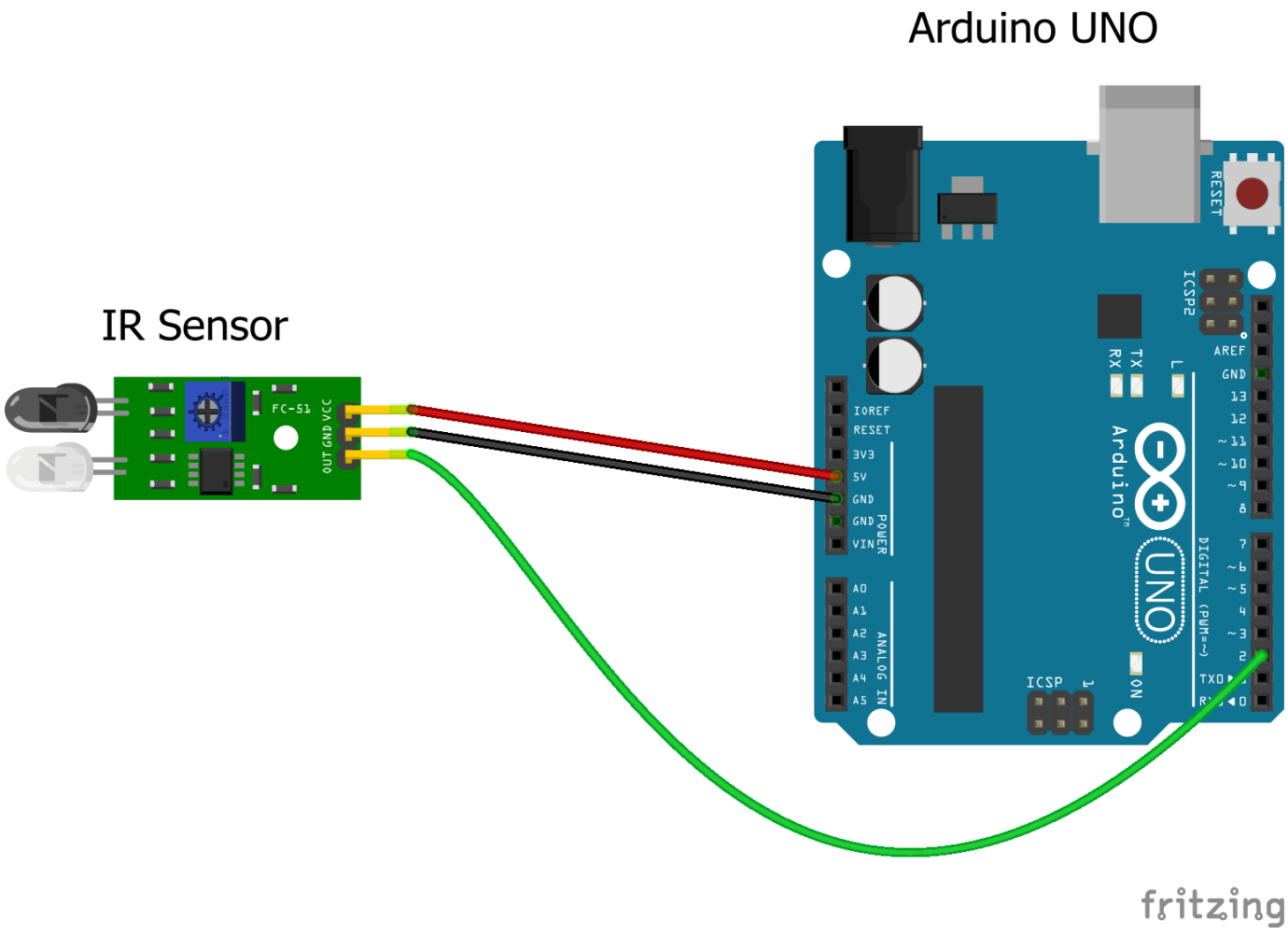
   Operating voltage 5VDC

   I/O pins are 5V and 3.3V compliant

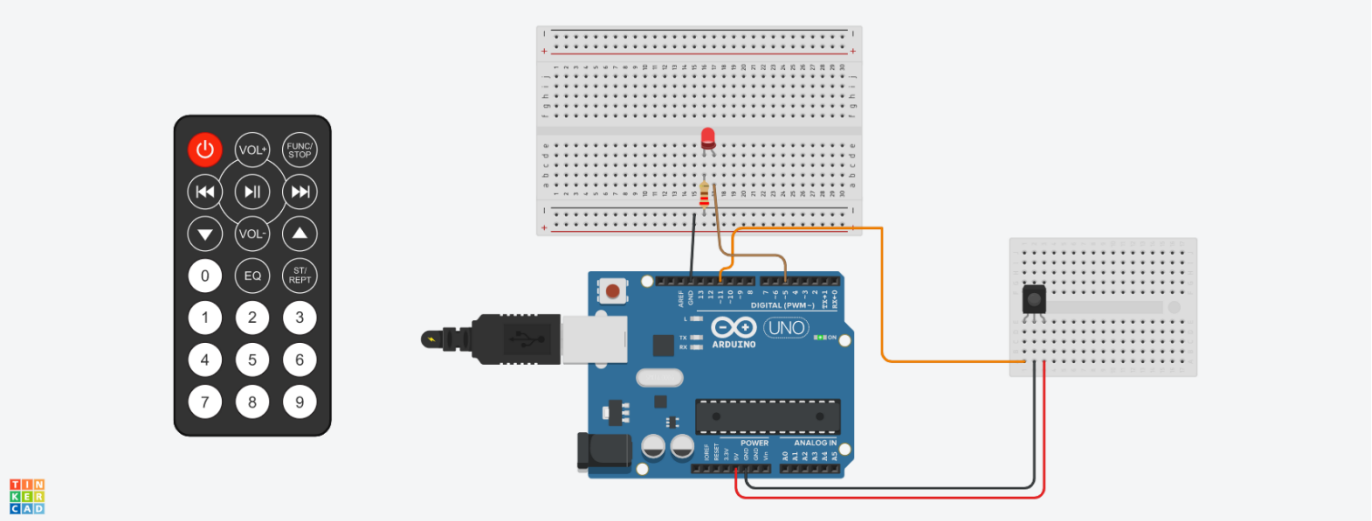
5. Pin Diagram



6. Interfacing with Arduino



Simulation Interface of IR And Arduino



LDR

1. Operating Principle with Diagram

Photoresistor or LDR (Light Dependent Resistor), as the name suggests will change it resistance based on the light around it. That is when the resistor is placed in a dark room it will have a resistance of few Mega ohms and as we gradually impose light over the sensor its resistance will start to decrease from Mega Ohms to few Ohms. This property helps the LDR to be used as a Light Sensor. It can detect the amount of light falling on it and thus can predict days and nights or can be used for changing the intensity of lights.

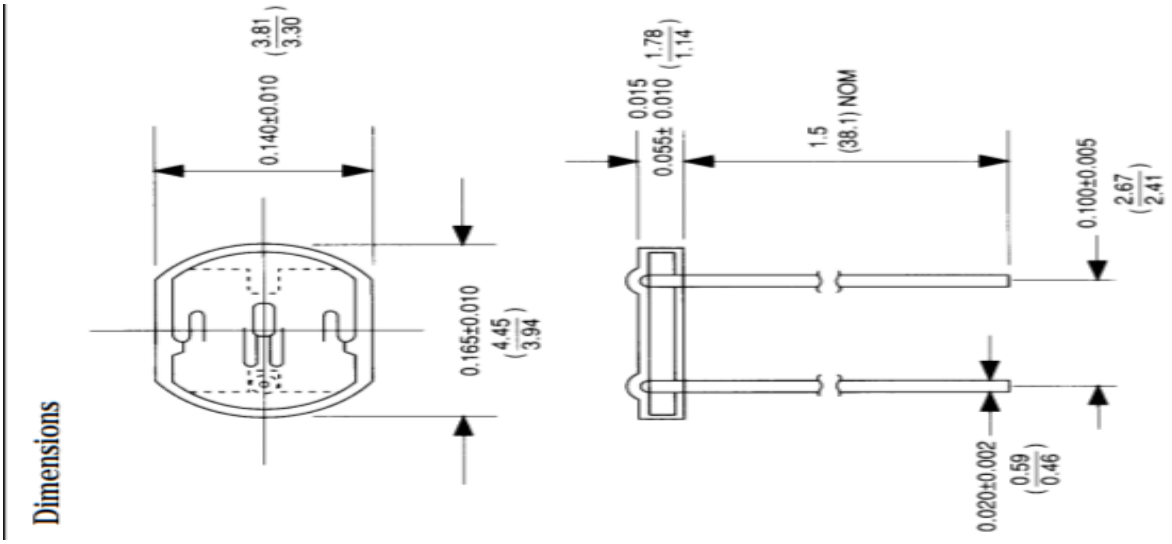
2. Measurement Range

Operating temperature: -25 Degree Celsius to +75 Degree Celsius

Guide to source illuminations

Light source Illumination	LUX
Moonlight	0.1
60W Bulb at 1m	50
1W MES Bulb at 0.1m	100
Fluorescent Lighting	500
Bright Sunlight	30,000

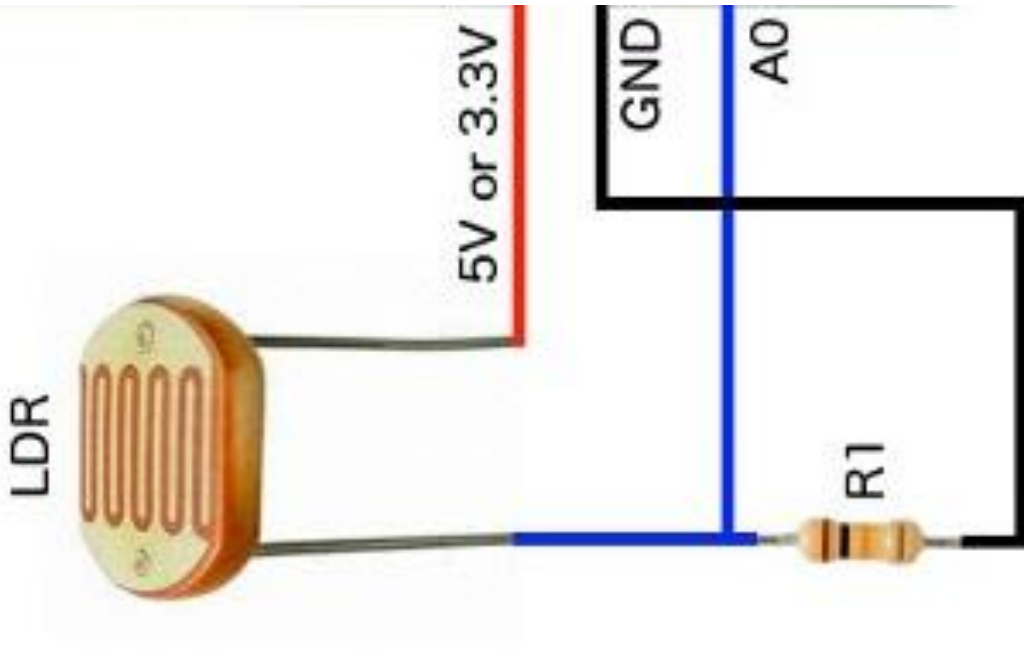
3. Physical Dimension



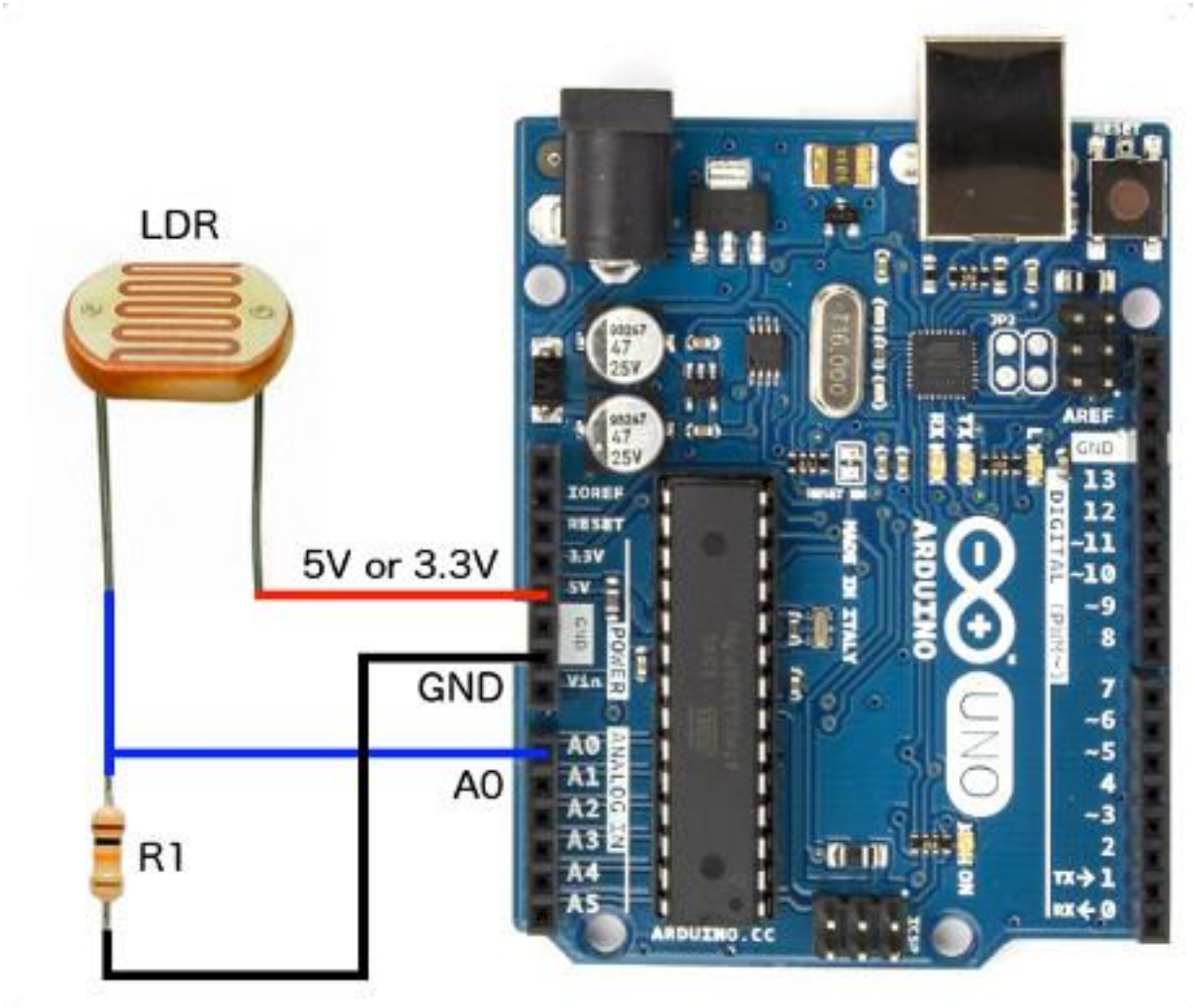
4. Power rating details

Peak Voltage: 100V Max
Current: 5mA

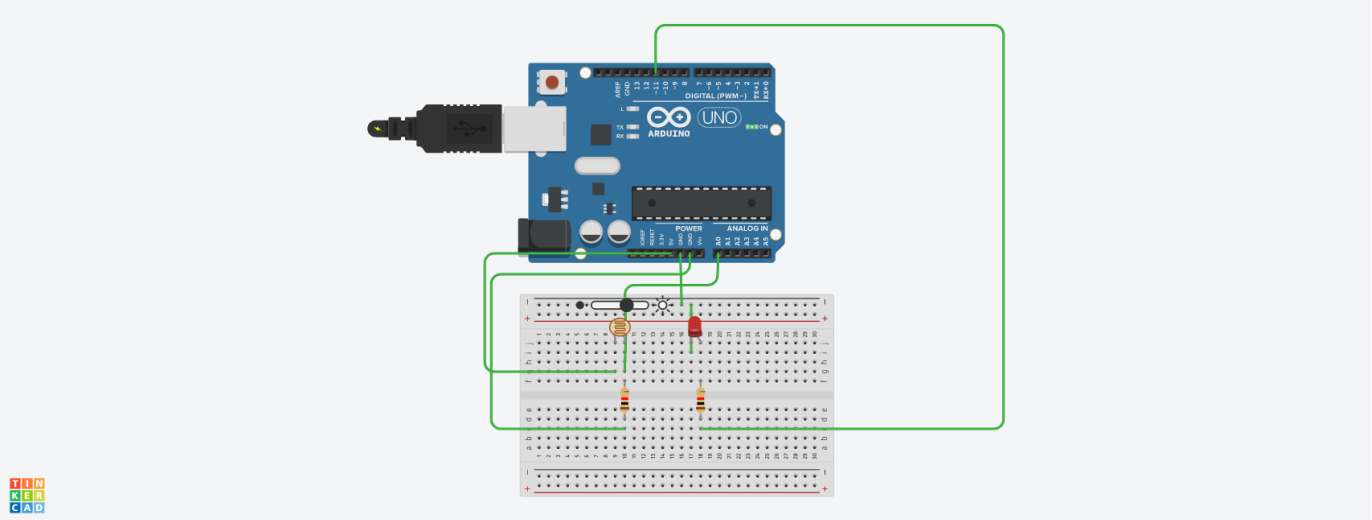
5. Pin Diagram



6. Interfacing with Arduino



Simulation Interface of LDR And Arduino



LM35

1. **Operating Principle with Diagram**

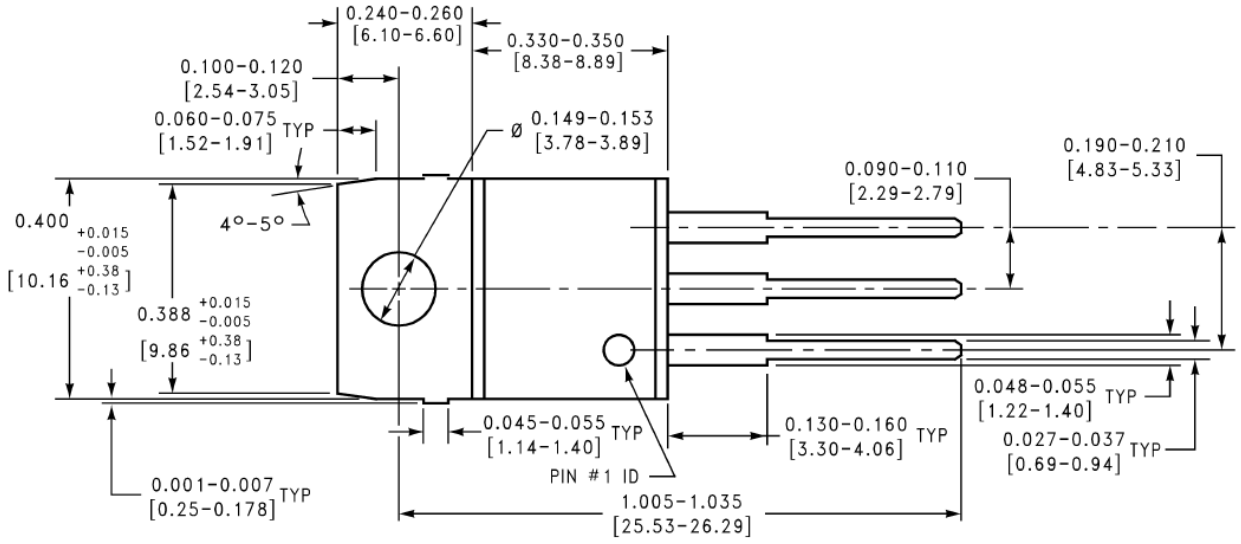
   In principle, the sensor will perform sensing when the temperature changes every 1 ºC temperature will show a voltage of 10 mV. In placing the LM35 can be affixed with adhesive or can be cemented on the surface but the temperature will be slightly reduced by about 0.01 ºC being absorbed in the surface temperature. In this way the expected difference between the air temperature and the surface temperature can be detected by a sensor LM35 same temperature as the surrounding, if the surrounding air temperature is much higher or much lower than the surface temperature, the LM35 is the surface temperature and the temperature of the surrounding air.

2. Measurement Range

LM35 Measurement Range is from -55 degree Celsius to +150 Degree Celsius

3. Physical Dimension

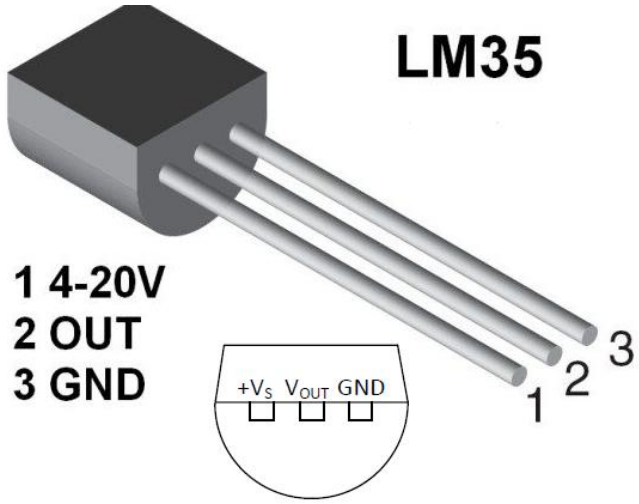
-



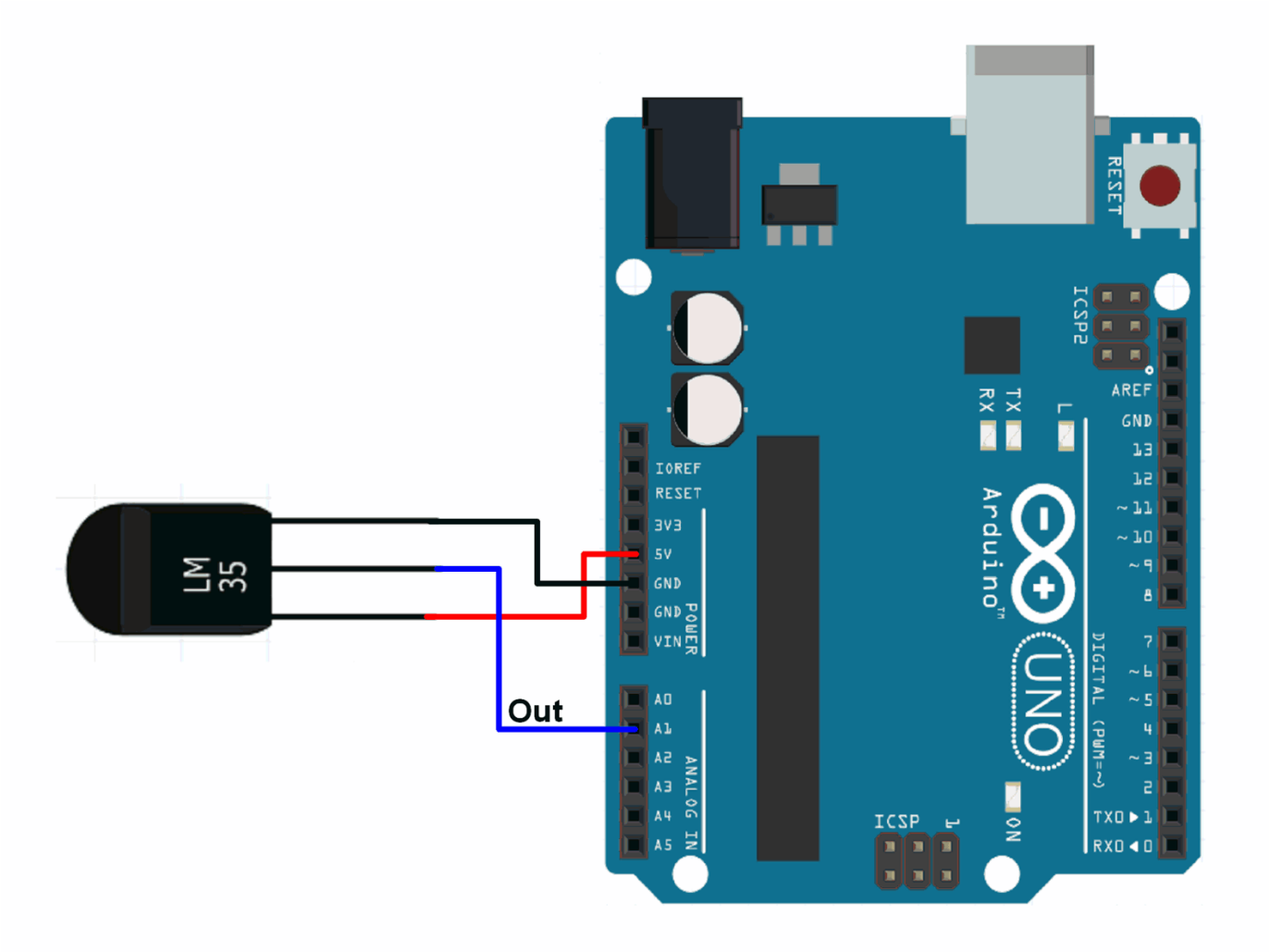
4. Power rating details

Operating Voltage: 4V – 30V
Supply Voltage: 5V

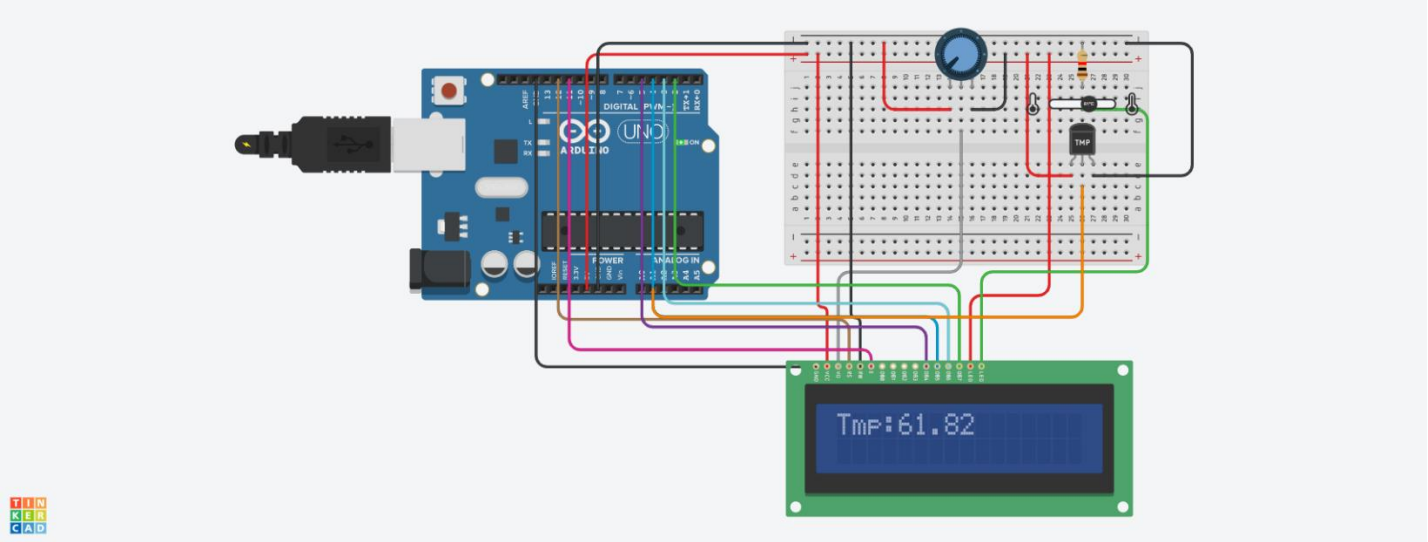
5. Pin Diagram



6. Interfacing with Arduino



Simulation Interface of LM35 And Arduino



CHAPTER 8

ACTUATORS AND DISPLAYS

Liquid crystal display (LCD 16x2)

1. Operating Principle with diagram

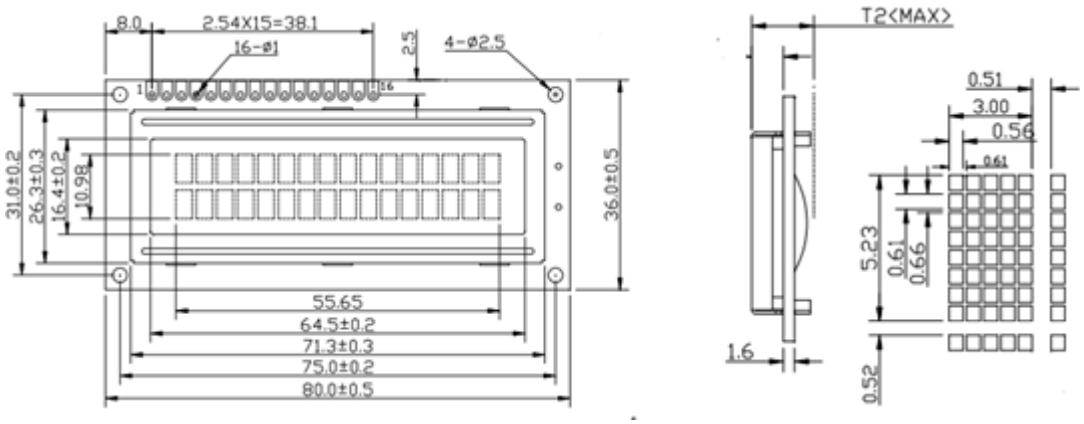
LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO's or calculators. The appearance and the pinouts have already been visualized above now let us get a bit technical.
16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots. A Single character with all its Pixels is shown in the below picture.

Now, we know that each character has (5×8=40) 40 Pixels and for 32 Characters we will have (32×40) 1280 Pixels. Further, the LCD should also be instructed about the Position of the Pixels. Hence it will be a hectic task to handle everything with the help of MCU, hence an Interface IC like HD44780is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the Commands and Data from the MCU and process them to display meaningful information onto our LCD Screen. You can learn how to interface an LCD using the above-mentioned links. If you are an advanced programmer and would like to create your own library for interfacing your Microcontroller with this LCD module then you have to understand the HD44780 IC is working and commands which can be found its datasheet.

2. Measurement Range

This Bluetooth module can cover maximum 9 meter of signal.

3. Physical Dimension

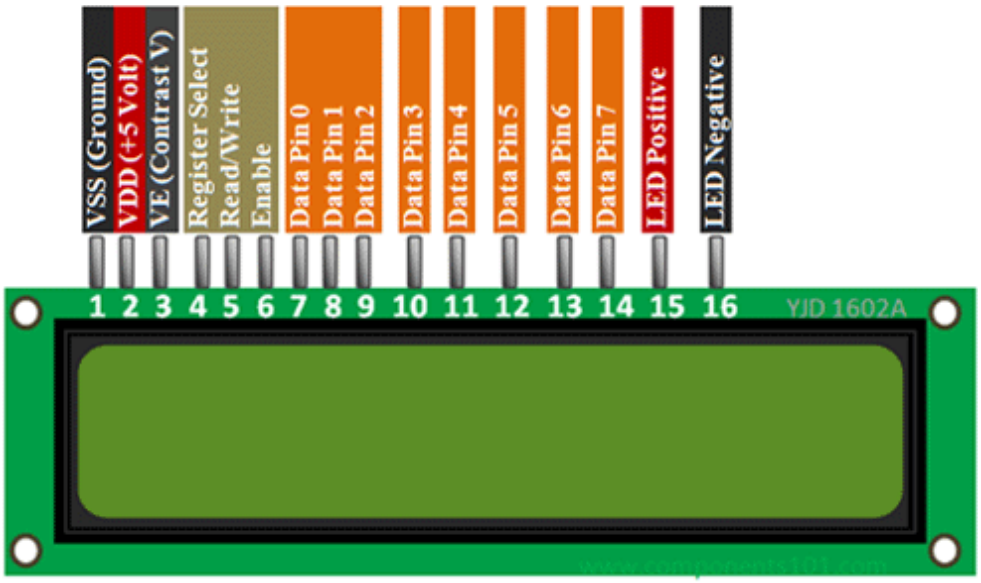


4. Power rating details

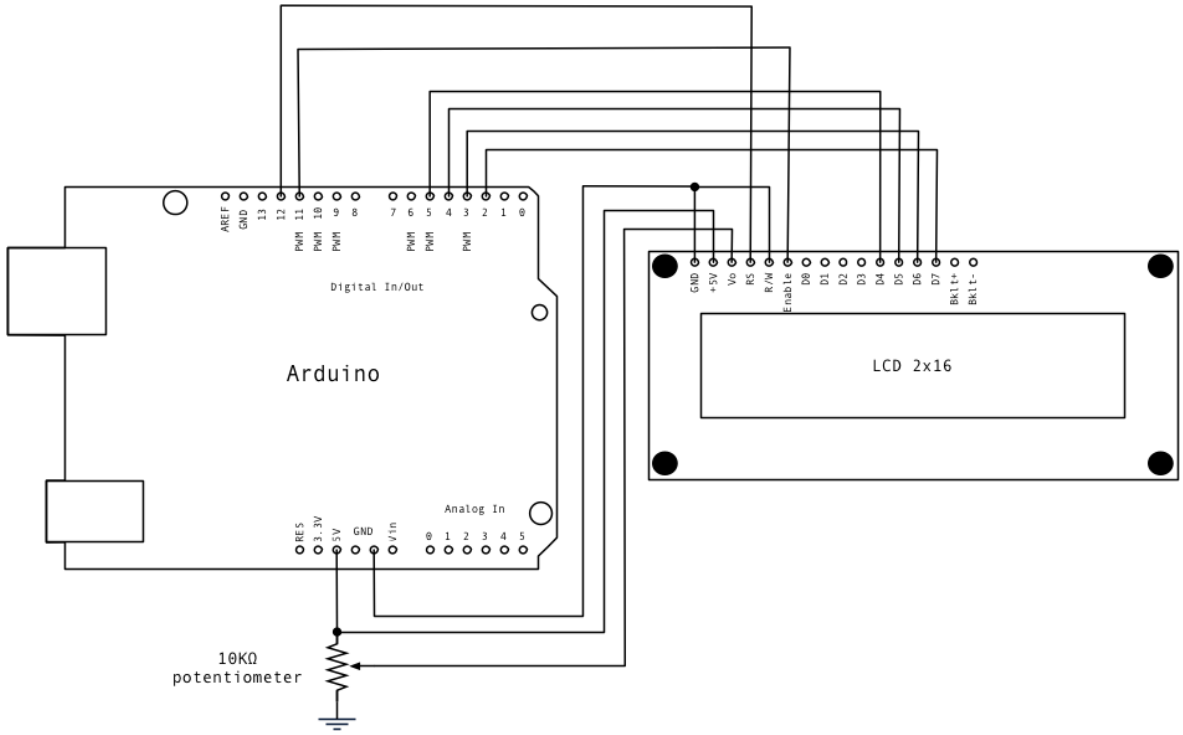
Operating voltage : 4.7V – 5.3V

DC Supply voltage : 5V DC

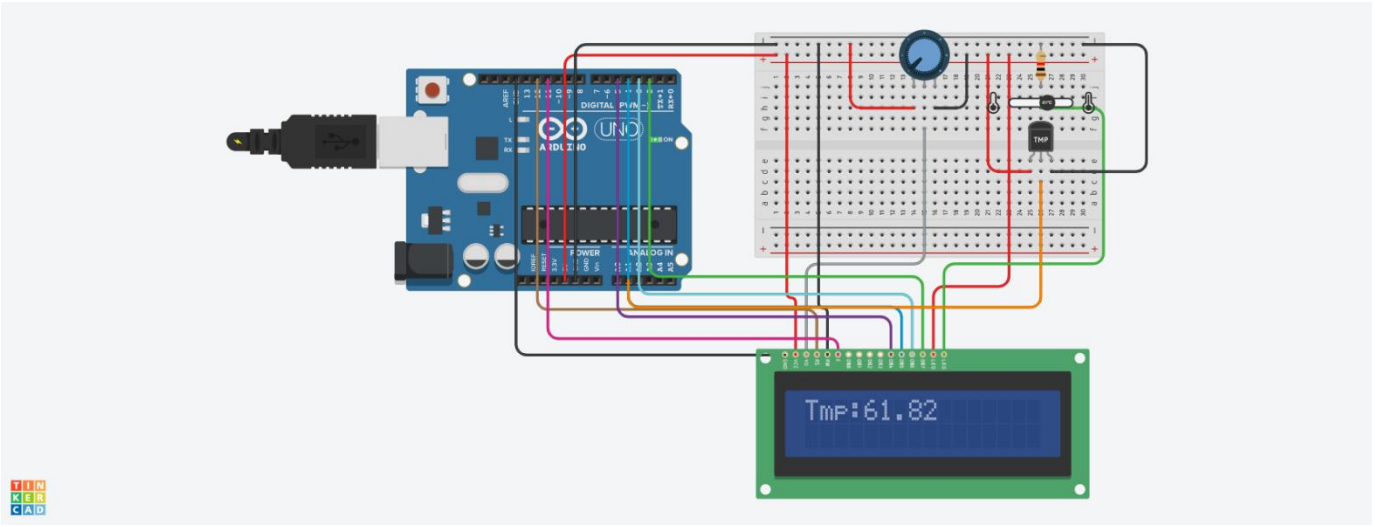
5. Pin Diagram



6. Interface with Arduino



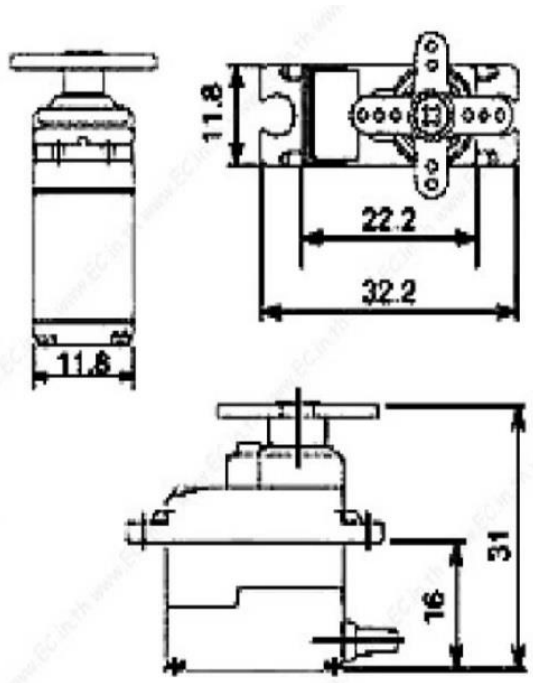
Simulation Interface Of 16X2 LCD Display and Arduino



Servo Motor (Actuator)

1. Operating Principle with diagram

Servo motor works on the PWM (Pulse Width Modulation) principle, which means its angle of rotation is controlled by the duration of pulse applied to its control PIN. Basically, servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears.

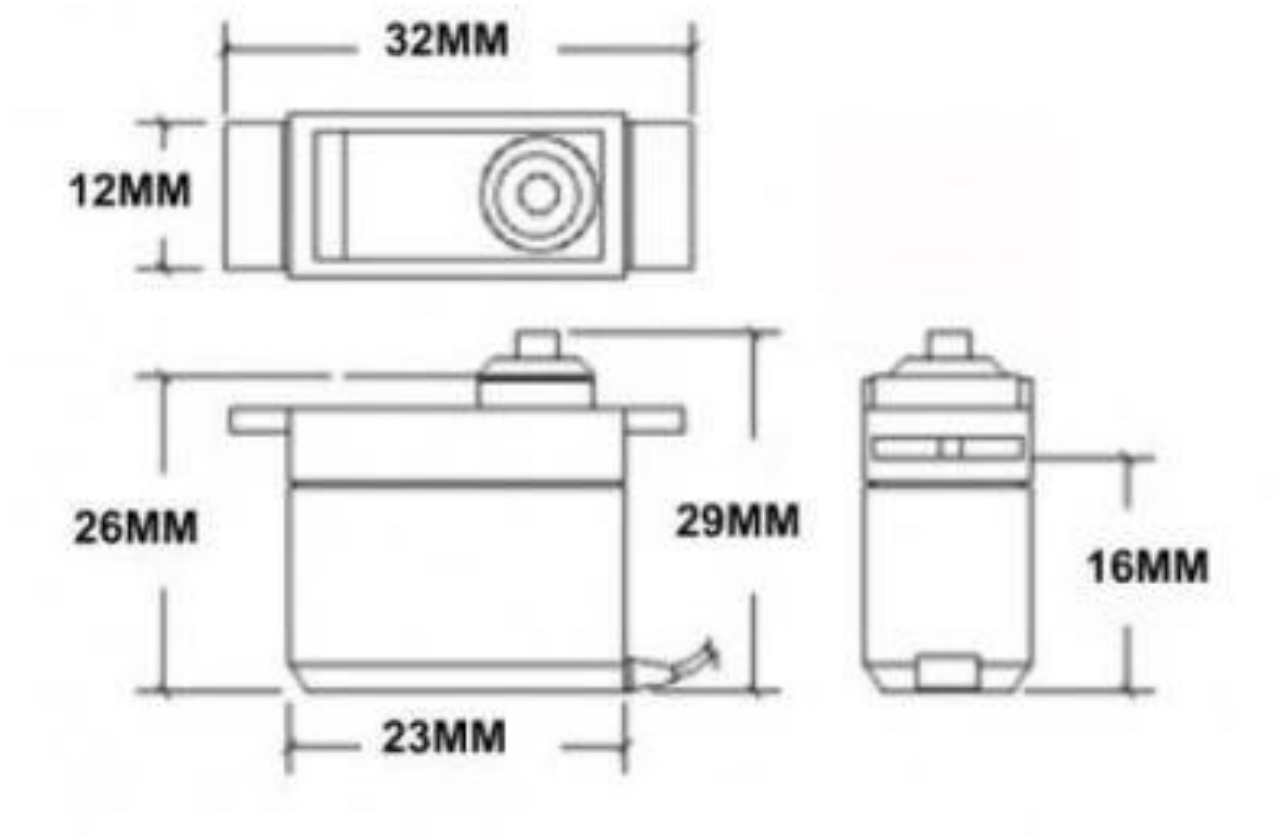


2. Measurement Range

Rotational Range Is 180 Degree

3. Physical Dimension

Length: 0.91 in (23.1 mm)
Width: 0.48 in (12.2 mm)
Height:1.14 in (29.0 mm)

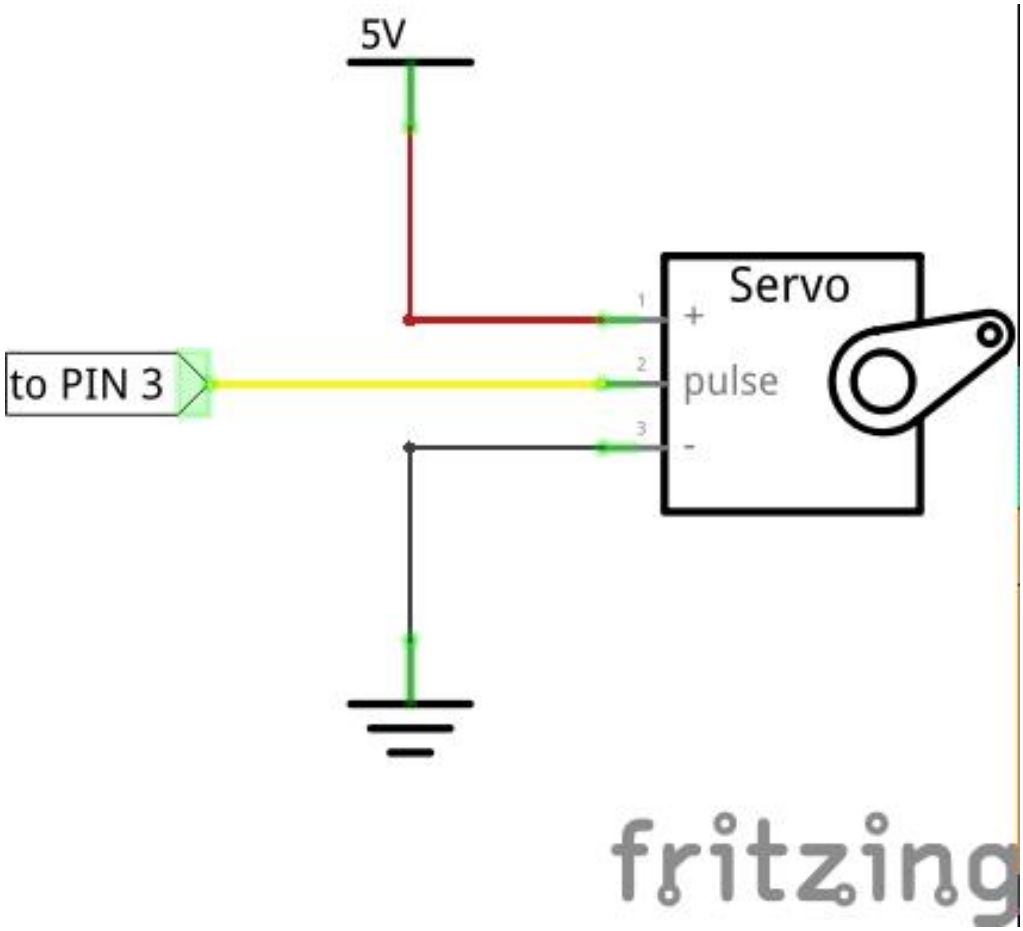


4. Power rating details

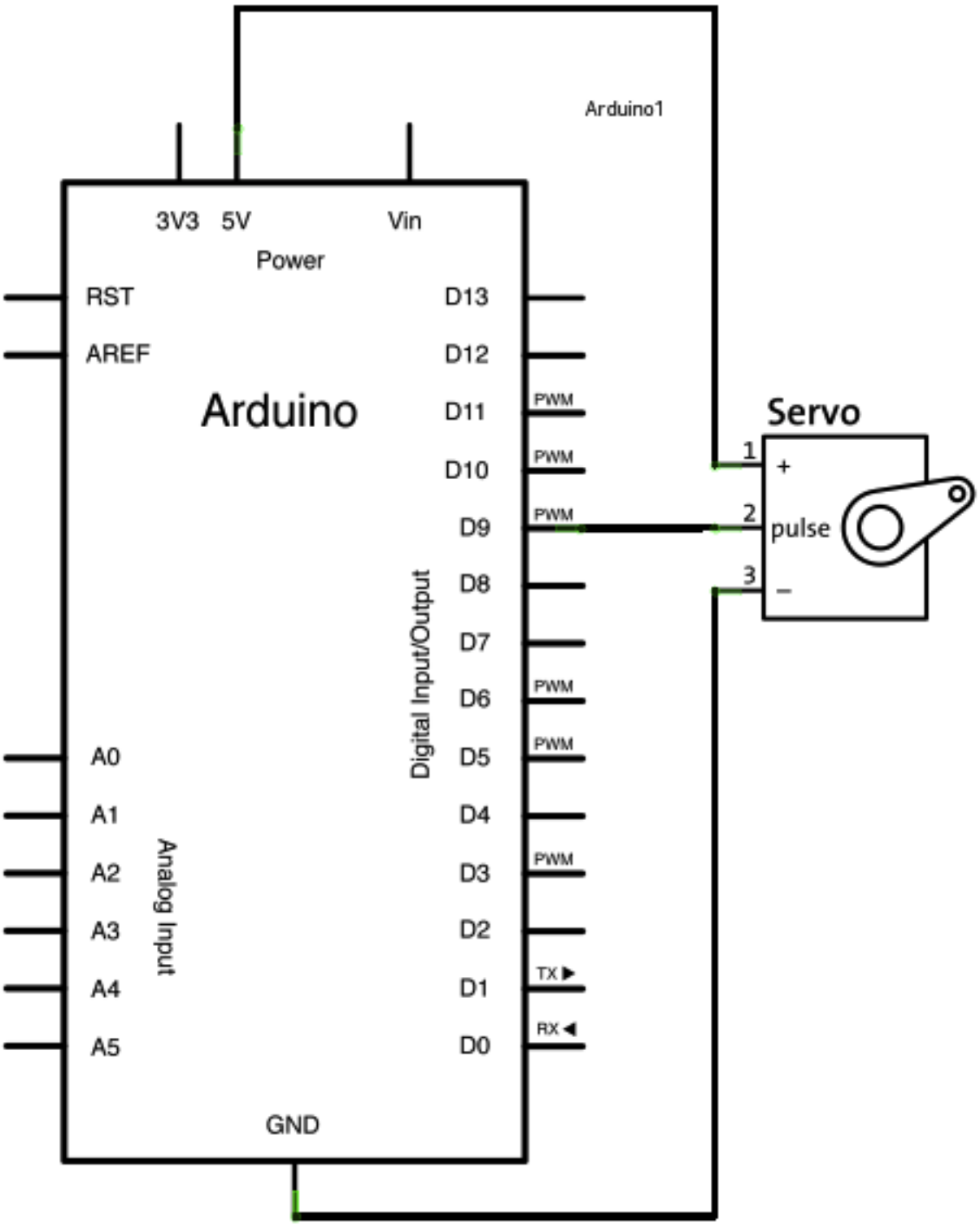
Operating voltage : 4.8V DC

Supply voltage : 5V DC

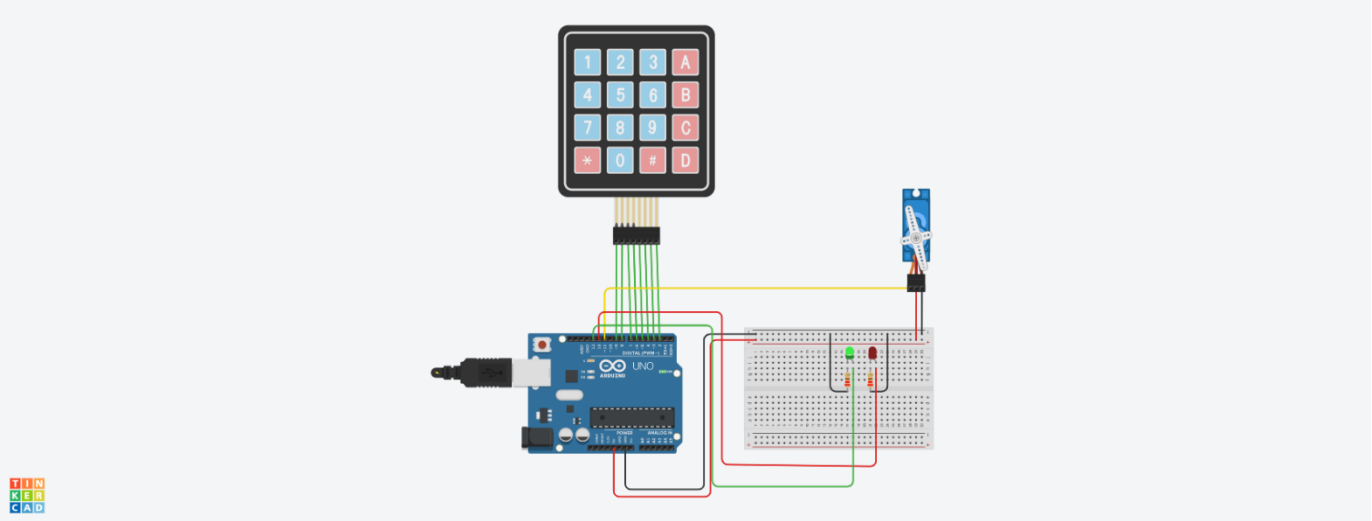
5. Pin Diagram



6. Interface with Arduino



Simulation Interface of Arduino And Servo Motor.



4X4 KeyPad

1. Operating Principle with Diagram

A matrix keypad is a small compact input device that accepts user inputs and processed by Microcontrollers. You might have seen this in most commonly used devices like Calculators, Digital locks, Gas pumps and DIY projects. It comes in different types, one of them is membrane keypads, it is thinner in size and you can paste it on top of your creative projects.

If we have to connect 16 buttons to the microcontroller, then each button takes 1 GPIO pin. But if we use the matrix keypad, we just need 8 pins only.

Initially, all rows are set to 0(LOW) and all Columns are set to 1(HIGH). When a key press occurs the **column pin** will get contacted to the **row pin** and makes the entire column state to low. To identify the exact pin at the column, we need to **scan each row by sending 1 (HIGH)** and **read the state at Column pins**. The column which changes the state from 0(LOW) to 1(HIGH) then that is the location of the pressed key (Passes the HIGH signal from Row to Column pin). Let's see this in detail with an example

In the idle state, the row and column will be like this,

Row R1 R2 R3 R4 – 0000

Column C1 C2 C3 C4– 1111

2. **Measurement Range**
3. **Physical Dimension**

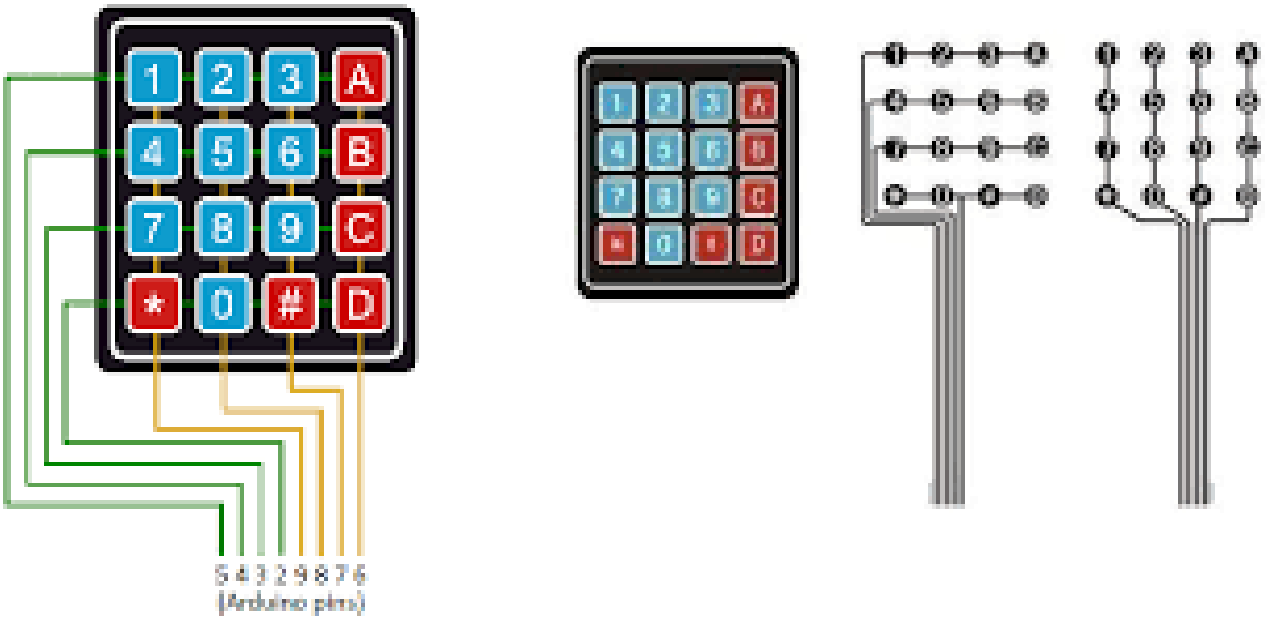
Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

4. **Power rating details**

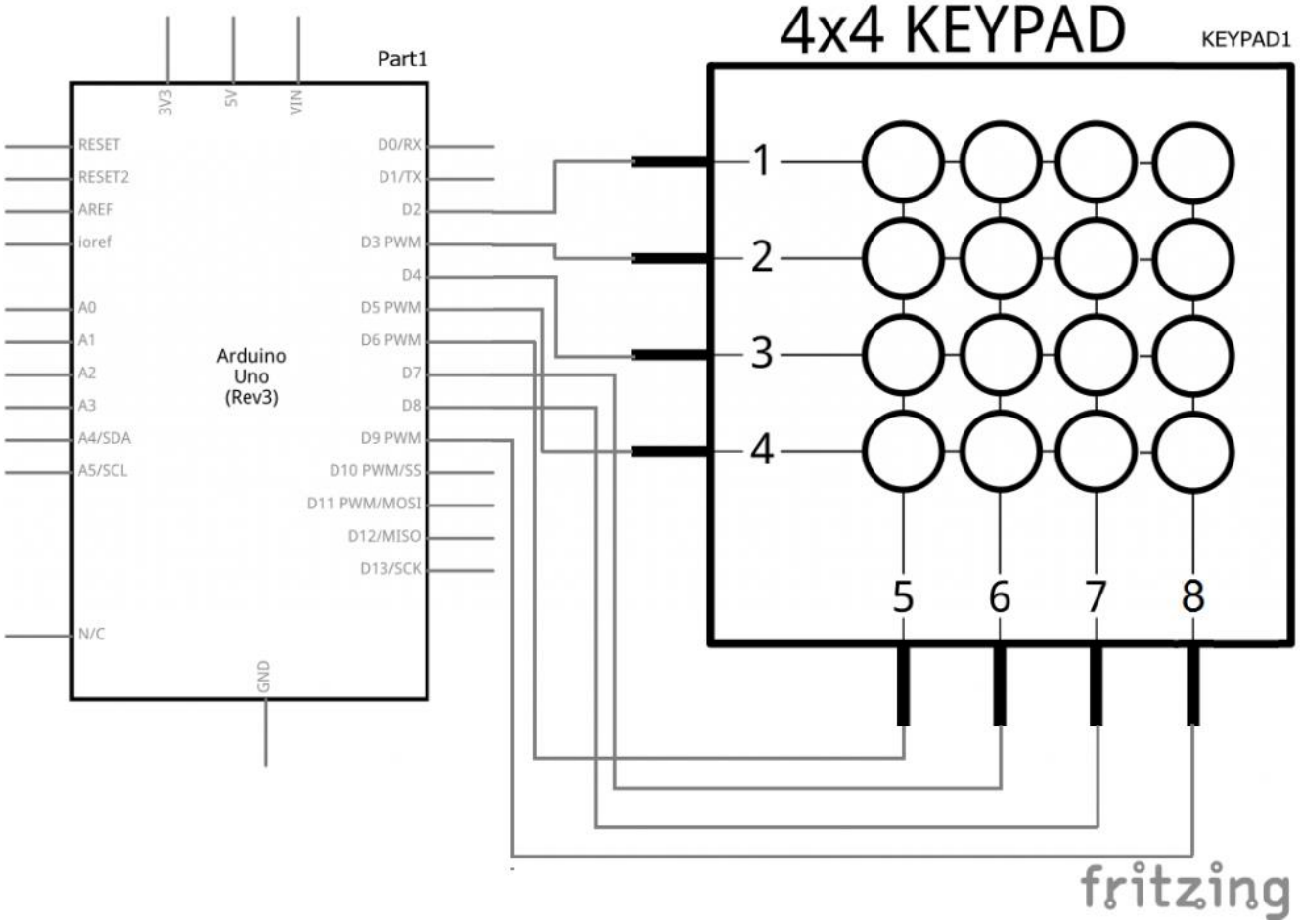
Maximum **Voltage** across EACH SEGMENT or BUTTON: 24V.

Maximum Current through EACH SEGMENT or BUTTON: 30mA.

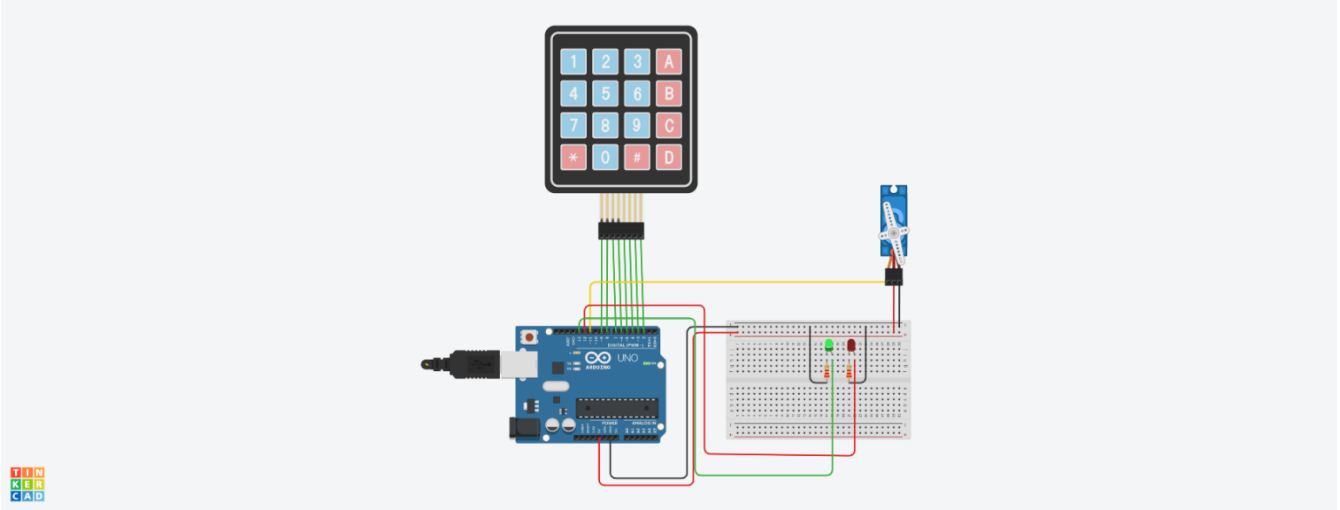
5. **Pin Diagram**



6. **Interfacing with Arduino**



Simulation Interface of 4X4 KeyPad and Arduino



CHAPTER 9

DETAILS OF WEBSITE

For our project we made a Single Webpage using the HTML for a Webpage, CSS for making the page attractive, and JavaScript for interacting the page. This Single Webpage is fully responsive so it would open in Mobile as easily as it opens in Desktop. The Bootstrap framework is used so that all the icons and text look more appealing. We have used different icons like font-Awesome, fav-icons to make icons look more eye-catchy.

Code for Bootstrap:

```
<!-- Favicon-->
<link rel="icon" type="image/x-icon" href="assets/img/favicon.ico" />
<!-- Font Awesome icons (free version)-->
<script src="https://use.fontawesome.com/releases/v5.13.0/js/all.js"
crossorigin="anonymous"></script>
<!-- Google fonts-->
<link href="https://fonts.googleapis.com/css?family=Merriweather+Sans:400,700"
rel="stylesheet" />
<link href="https://fonts.googleapis.com/css?family=Merriweather:400,300,300italic,400italic,
700,700italic" rel="stylesheet" type="text/css" />
<!-- Third party plugin CSS-->
<link href="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/magnific-popup.
min.css" rel="stylesheet" />
<!-- Core theme CSS (includes Bootstrap)-->
<link href="css/styles.css" rel="stylesheet" />
```

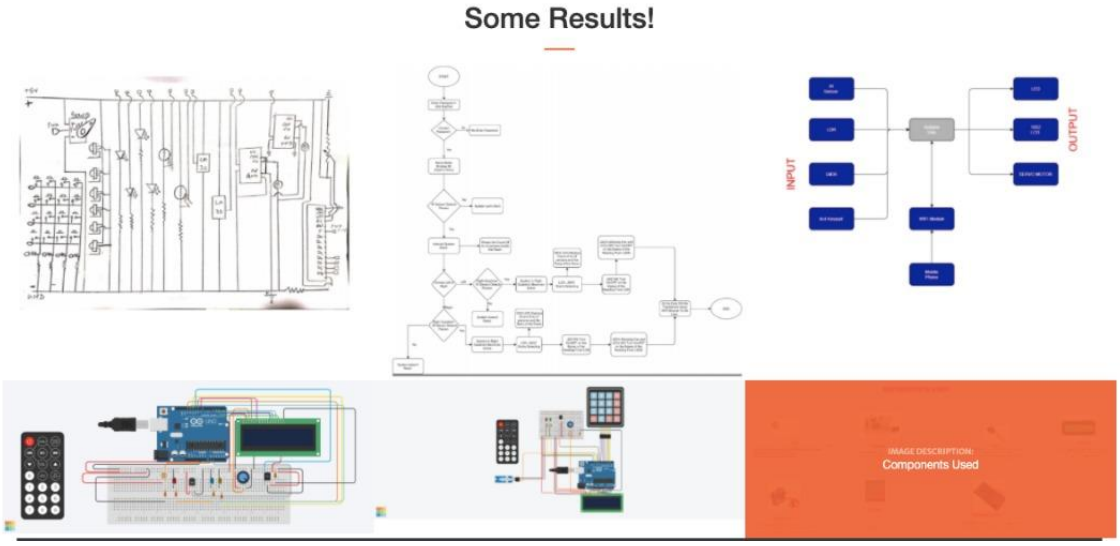
Code for icons:

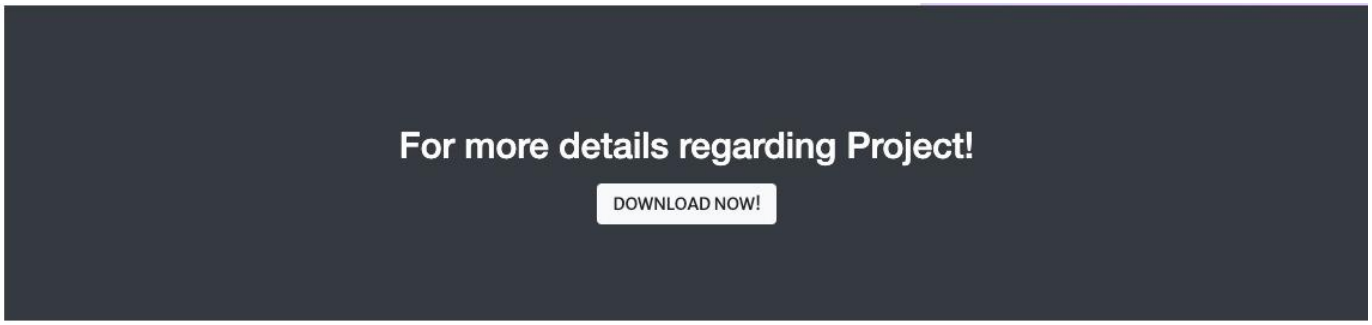
```
<!-- Bootstrap core JS-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.bundle.min.
js"></script>
<!-- Third party plugin JS-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.4.1/jquery.easing.min.
js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/jquery.
magnific-popup.min.js"></script>
<!-- Core theme JS-->
<script src="js/scripts.js"></script>
```

About Us!

The system as the name indicates, 'Smart Home System to Save Electricity', makes the system more flexible and provides an attractive user interface. In this system, we integrate various Sensors for home automation system. A novel architecture for a home automation system is proposed using relatively new technologies. The system consists of mainly 5 components, That is an IR Sensor, LM35, LDR, DC Motor and 16x2 LCD. We hide the complexity of the notions involved in the home automation system by including them into a simple, but comprehensive set of related concepts. In our project, we are controlling the lights, fans and A/C's based on human presence. We are using IR sensors to detect human presence, LDR to control lights and LM35 temperature sensor to control Fans and A/C's and DC Motor for depiction of Fans.

GET STARTED!





CHAPTER 10

DETAILS OF COMMUNICATION PROTOCOLS

Protocol Used: Wi-Fi

Wi-Fi is the name of a wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. A common misconception is that the term Wi-Fi is short for "*wireless fidelity*," however this is not the case. Wi-Fi is simply a trademarked phrase that means *IEEE 802.11x*.

Wi-Fi networks have no physical wired connection between sender and receiver by using radio frequency (RF) technology -- a frequency within the electromagnetic spectrum associated with radio wave propagation. When an RF current is supplied to

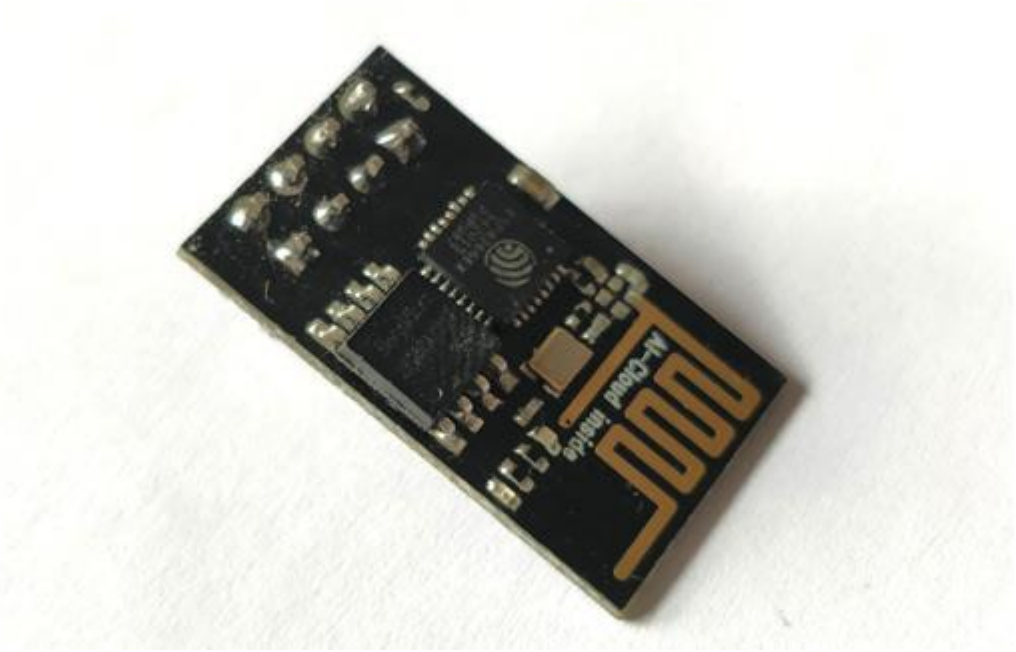
an antenna, an electromagnetic field is created that then is able to propagate through space.

The cornerstone of any wireless network is an access point (AP). The primary job of an access point is to broadcast a wireless signal that computers can detect and "tune" into. In order to connect to an access point and join a wireless network, computers and devices must be equipped with wireless network adapters.

The Wi-Fi Alliance, the organization that owns the Wi-Fi registered trademark term, specifically defines Wi-Fi as any "*wireless local area network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards.*"

Initially, Wi-Fi was used in place of only the 2.4GHz 802.11b standard, however the Wi-Fi Alliance has expanded the generic use of the Wi-Fi term to include any type of network or WLAN product based on any of the 802.11 standards, including 802.11b, 802.11a, dual-band and so on, in an attempt to stop confusion about wireless LAN interoperability.

In Our Project We implemented this Protocol Using ESP8266WiFi Module



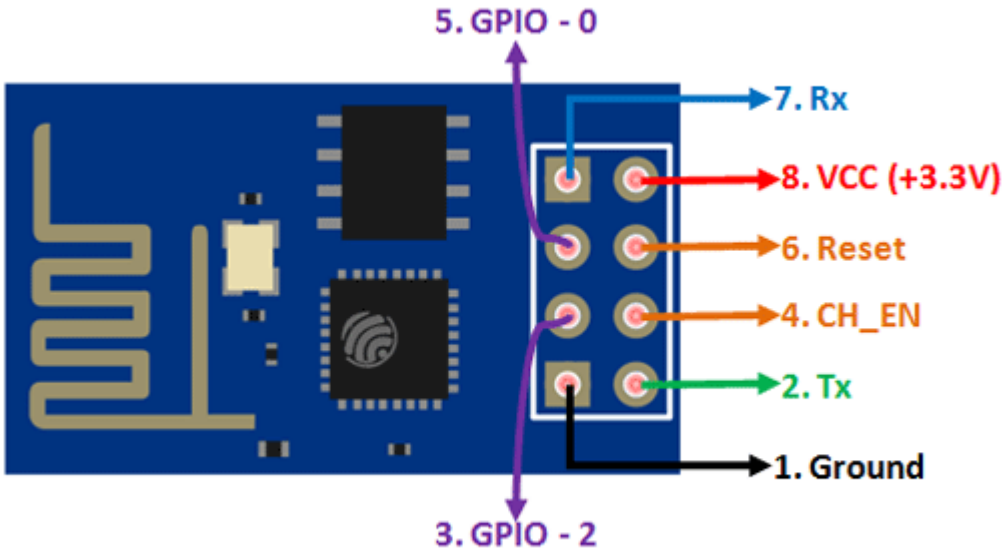
The ESP8266 is a very user friendly and low-cost device to provide internet connectivity to your projects. The module can work both as an Access point (can create hotspot) and as a station (can connect to Wi-Fi), hence it can easily fetch data and upload it to the internet making Internet of Things as easy as possible.

We Can Configure It Using Arduino IDE which makes it more User Friendly. It can also fetch data from internet using API's because of which system could access any information that is available in the internet, thus making it smarter.

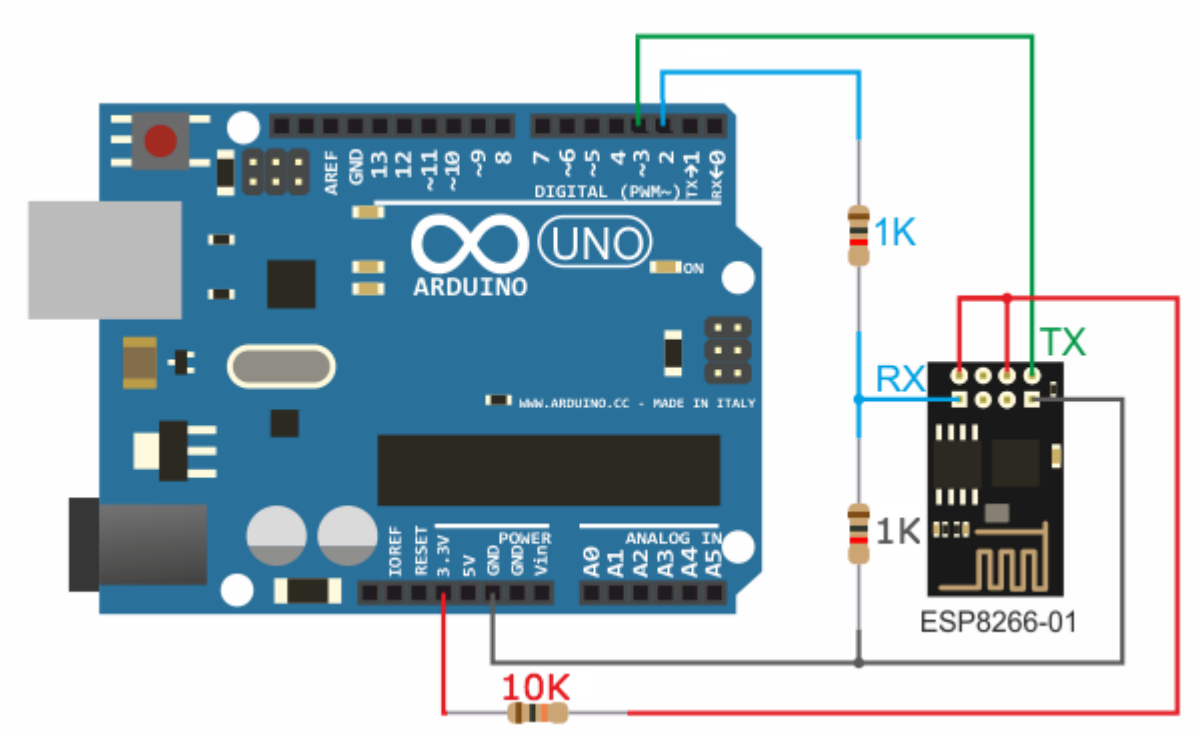
However this version of the module has only 2 GPIO pins so we have to use it along with another microcontroller like Arduino, else we can look onto the more standalone ESP-12 or ESP-32 versions.

The ESP8266 module works with 3.3V only, anything more than 3.7V would kill the module.

Pin Layout:



Interfacing with Arduino:



Demo Code:

The Following Code Will Receive data AT commands from the Arduino's serial window to send them to the ESP8266, and to print the ESP8266's response to the command or to other actions

```
#include <SoftwareSerial.h>


SoftwareSerial esp8266(2,3); // make RX Arduino line is pin 2, make TX
Arduino line is pin 3.

                                    // This means that you need to connect the TX line from the esp to the
                                Arduino's pin 2

                                    // and the RX line from the esp to the Arduino's pin 3

void setup()

{

 Serial.begin(9600);

 esp8266.begin(9600); // your esp's baud rate might be different

}


void loop()

{

 if(esp8266.available()) // check if the esp is sending a message

 {
```

```
  while(esp8266.available())
  {
    // The esp has data so display its output to the serial window
    char c = esp8266.read(); // read the next character.
    Serial.write(c);
  }
}
```

```
if(Serial.available())
{
  delay(1000);

  String command="";

  while(Serial.available()) // read the command character by character
  {
      // read one character
    command+=(char)Serial.read();
  }
  esp8266.println(command); // send the read character to the esp8266
}
}
```

Sr.no	Bluetooth	Zigbee	Wi-Fi
1	Bluetooth frequency ranges between 2.4 GHz to 2.483 GHz.	Zigbee frequency range is mostly 2.4 GHz.	Wi-Fi frequency ranges between 2.4 GHz to 5 GHz.

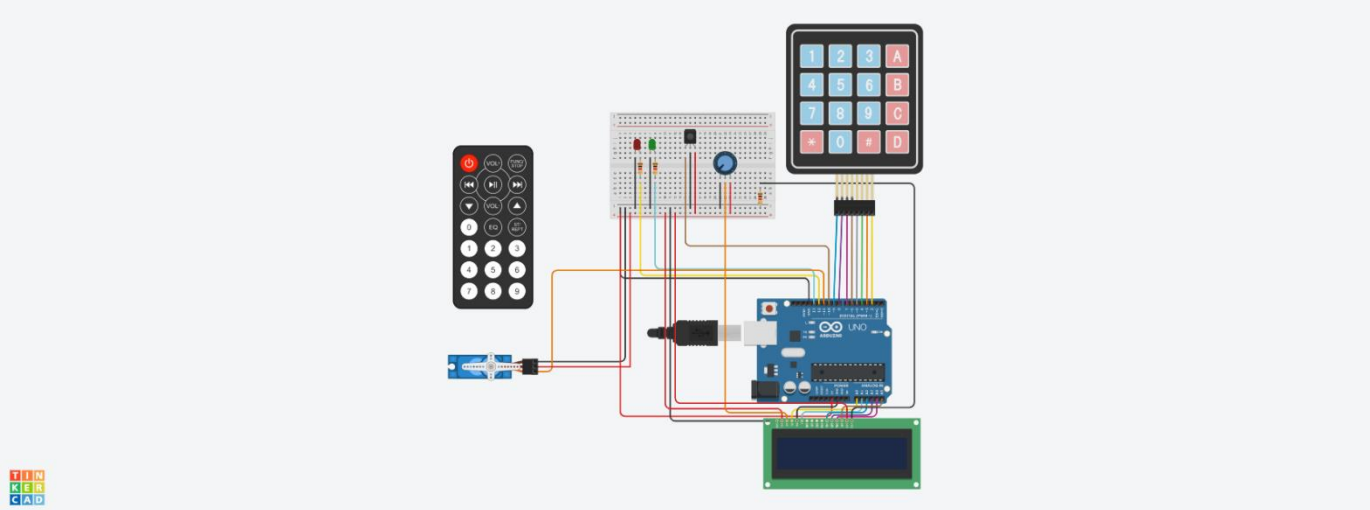
2	Bluetooth has 79 RF Channels.	Zigbee has 16 RF Channels.	
3	Bluetooth uses GFSK modulation technique.	Zigbee uses GFSK, BPSK and QPSK modulation techniques.	Wi-Fi uses BPSK and QPSK modulation techniques.
4	Bluetooth has maximum 8 cell nodes.	Zigbee also has more than 65000 cell nodes.	
5	Bluetooth needs low bandwidth.	Zigbee needs low bandwidth but higher than Bluetooth.	Wi-Fi needs high bandwidth.
6	Bluetooth follows IEEE 802.15.1.	Zigbee follows IEEE 802.15.4.	Wi-Fi follows IEEE 802.11
7	Bluetooth radio signal range is up to 10 meters.	Zigbee radio signal range is up to 100 meters.	Wi-Fi coverage area is up to 32 meters.

CHAPTER 12

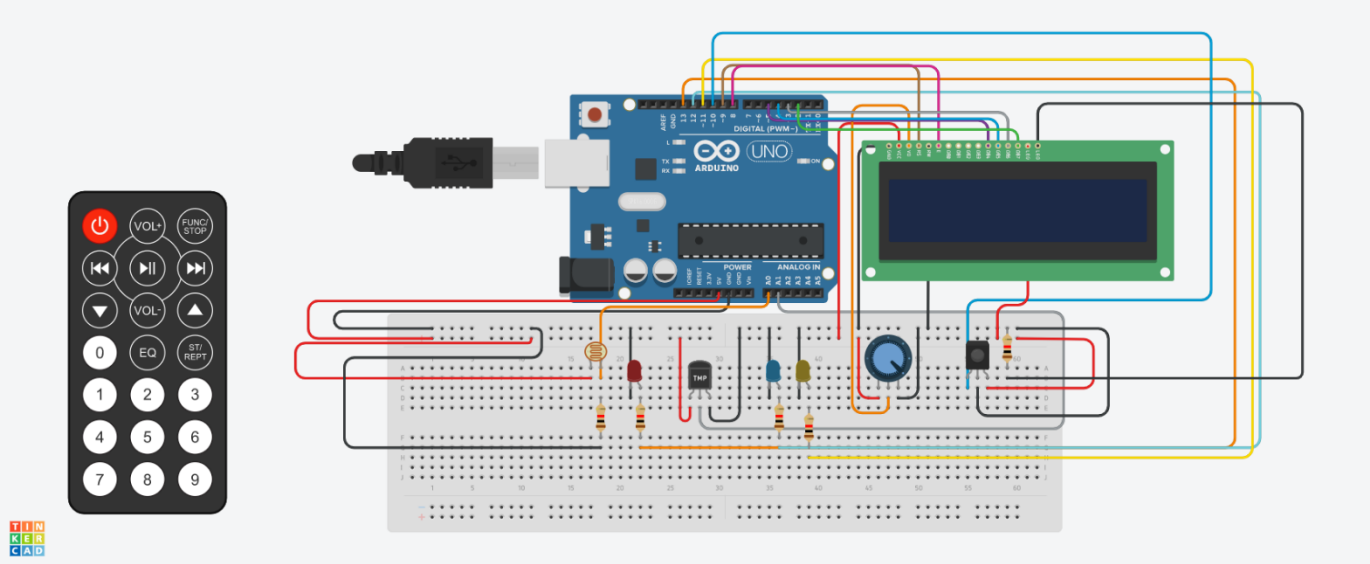
COMPLETE CODE OF THE PROJECT

The are Two Circuits in Our Project

1. **Outer Circuit**
2. **Inner Circuit**



Outer Circuit



Inner Circuit

Code for Outer Circuit.

```
        #include <LiquidCrystal.h>
        #include <IRremote.h>
#include <Keypad.h>
#include <Servo.h>

        int RECV_PIN = 10;
        IRrecv irrecv(RECV_PIN);
        decode_results results;

        const int rs=A0,en=A1,d4=A2,d5=A3,d6=A4,d7=A5;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

Servo servo_Motor;
        unsigned int count;
char* password = "852";
int position = 0;
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
{'1','2','3','A'},
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};

byte rowPins[ROWS] = { 9, 8, 7, 6 };
byte colPins[COLS] = { 5, 4, 3, 2 };
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

const int red = 12;
const int green = 13;

void setup(){
 pinMode(red, OUTPUT);
 pinMode(green, OUTPUT);
          Serial.begin(9600);
 servo_Motor.attach(11);
 setLocked(true);
 irrecv.enableIRIn();
 lcd.print("No Of Person:");
 lcd.setCursor(13,0);
           lcd.print(count);
}
void loop(){
 char key = keypad.getKey();
 if (key == '*' || key == '#') {
    position = 0;
    setLocked(true);
```

```
  }
  if (key == password[position]){
      position ++;
  }
  if (position == 4){
      setLocked(false);
  }
  delay(100);
  }
  void setLocked(int locked){
  if (locked){
     digitalWrite(red, HIGH);
     digitalWrite(green, LOW);
     servo_Motor.write(0);
  }
  else{
     digitalWrite(green, HIGH);
     digitalWrite(red, LOW);
     servo_Motor.write(90);
  }

   if (irrecv.decode(&results)) //irrecv.decode(&results) returns true if anything is recieved, and stores info
in varible results
   {
          unsigned int value = results.value; //Get the value of results as an unsigned int, so we can use switch
case
      irrecv.resume(); // Receive the next value
      switch (value)
      {
                  case 12495:
                                    count++;
                                    break;
       case 2295:
                                    count++;
                                    break;
      case 34935:
                                    count++;
                                    break;
      case 18615:
                                    count++;
                                    break;
      case 10455:
                                    count++;
                                    break;
      case 43095:
                                    count++;
                                    break;
      case 26775:
                                    count++;
                                    break;
      case 3675:
```

```
                                count++;
                                break;
     case 39015:
                                count++;
                                break;
     case 22695:
                                count++;
                                break;


}
   switch (value)
    {
                    case 255:
                                count--;
                                break;
     case 32855:
                                count--;
                                break;
     case 36975:
                                count--;
                                break;
     case 16575:
                                count--;
                                break;
     case 41055:
                                count--;
                                break;
     case 8415:
                                count--;
                                break;
     case 24735:
                                count--;
                                break;
     case 4325:
                                count--;
                                break;
     case 20655:
                                count--;
                                break;
     case 45135:
                                count--;
                                break;
    case 28815:
                                count--;
                                break;



}
```

```
            lcd.setCursor(13,0);
            lcd.print(count);
    Serial.print("No of Persons:");
    Serial.println(count);
   }
  }
```

Code for Inner Circuit.

```
#include <IRremote.h>
#include<LiquidCrystal.h>
int ir5=10,ledpin1=13,relay1=12,acpin1=11;
int count=0,flag1=0,flag2=0,i;
float temp1,val1, value;
int temppin1=A1,ldrpin1=A0;
int ldrstatus1;
IRrecv irrecv(10);
decode_results results;

 const int rs=9,en=8,d4=5,d5=4,d6=3,d7=2;
 LiquidCrystal lcd(rs,en,d4,d5,d6,d7);



void setup()
{
 Serial.begin(9600);
 lcd.begin(16,2);

 pinMode(ir5,INPUT);

 pinMode(ledpin1,OUTPUT);

 pinMode(relay1,OUTPUT);

 pinMode(acpin1,OUTPUT);

 pinMode(temppin1,INPUT);

 pinMode(ldrpin1,INPUT);

 lcd.print("No:");
 lcd.setCursor(4,0);
 lcd.print(count);
 lcd.setCursor(7,0);
 lcd.print("TEMP1:");
 lcd.setCursor(7,1);
 lcd.print("LIGHT:");
 irrecv.enableIRIn();
```

```
}
void loop()
{

 if(count>=0)
 {

    if(irrecv.decode(&results))//irrecv.decode(&results) returns true if anything is recieved, and stores info
in varible results
    {
        unsigned int value = results.value; //Get the value of results as an unsigned int, so we can use switch
case
        //Serial.println(value);
        irrecv.resume(); // Receive the next value


     ldrstatus1=analogRead(ldrpin1);

        if(ldrstatus1<=307)
        {
            analogWrite(ledpin1,255 - (ldrstatus1)/4);
            Serial.println(ldrstatus1/4);
        }
        else
        {
            digitalWrite(ledpin1,LOW);
        }
        val1=analogRead(temppin1);
        temp1=val1/4;
        Serial.println(temp1);
        if(temp1<5)
        {
        digitalWrite(acpin1,LOW);
        digitalWrite(relay1,LOW);
        }
        else if(temp1<29)
        {
            digitalWrite(acpin1,LOW);
            digitalWrite(relay1,HIGH);
        }
        else if(temp1>=30)
        {
        digitalWrite(acpin1,HIGH);
        digitalWrite(relay1,HIGH);
        }

    }

  }
  else
  {
```

```
    digitalWrite(ledpin1,LOW);

    digitalWrite(acpin1,LOW);

    digitalWrite(relay1,HIGH);

    temp1=0;

  }

  lcd.setCursor(13,0);
  lcd.print(temp1);
  lcd.setCursor(13,1);
  lcd.print((ldrstatus1/4));
}
```

CHAPTER 13

SUMMARY OF THE PROJECT

So in this Project we have Designed a System to save electricity. The project deals with an interesting manner of how energy can be saved by just turning instruments of room off when not in used. Our main objective is to control lights, fans and A/C's on the basis of human presence. We will be using IR sensors to count humans inside the room. As the person moves inside the lights will be controlled in that way. And as the temperature changes fans and A/C's will be operating accordingly. Also, we have 4x4 Keypad for Entering the Password. And Servo Motor for Opening the Door.

We also wanted to transfer all the data collected by the Sensors and Actuators and send it to our WebSite Using the ESP8266WIFI Module. Because of this the User can Monitor the activities happening inside the house from anywhere. But Due to the unavailability of the hardware due to the Current Scenario of Covid-19 and unavailability of WIFI module in the Simulator (TinkerCad), we were Unable to Implement the WIFI module Part. But we have created a website and provided Visualization of the data on that website using ThinkSpeak.

CHAPTER 14

REFERENCES

1. https://www.pjrc.com/teensy/td_libs_IRremote.html

2. https://components101.com/wireless/esp8266-pinout-configuration-features-datasheet

3. https://www.arduino.cc/reference/en/libraries/irremote/

4. https://www.ti.com

5. *M. Asadullah and A. Raza, "An overview of home automation systems," 2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), Rawalpindi, 2016, pp. 27-31, doi: 10.1109/ICRAI.2016.7791223.*

6. *Suseelan, Angel & Palaniappan, Satish & Hariharan, Naveen & Kesh, Naren & S, Vidhyalakshimi. (2015). Home Automation Systems - A Study. International Journal of Computer Applications. 116. 11-18. 10.5120/20379-2601.*

7. *Modrzyk, Nicolas. (2020). Vision and Home Automation. 10.1007/978-1-4842-5722-7_5*

Appendix A

DataSheets

1. IR Sensor

1. Descriptions

The Multipurpose Infrared Sensor is an add-on for your line follower robot and obstacle avoiding robot that gives your robot the ability to detect lines or nearby objects. The sensor works by detecting reflected light coming from its own infrared LED. By measuring the amount of reflected infrared light, it can detect light or dark (lines) or even objects directly in front of it. An onboard RED LED is used to indicate the presence of an object or detect line. Sensing range is adjustable with inbuilt variable resistor.

The sensor has a 3-pin header which connects to the microcontroller board or Arduino board via female to female or female to male jumper wires. A mounting hole for easily connect one or more sensor to the front or back of your robot chassis.

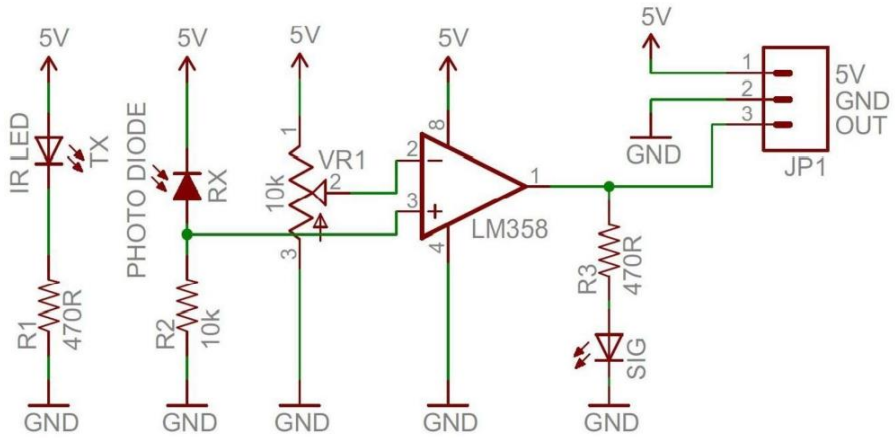
2. Features

- 5VDC operating voltage.
- I/O pins are 5V and 3.3V compliant.
- Range: Up to 20cm.
- Adjustable Sensing range.
- Built-in Ambient Light Sensor.
- 20mA supply current.
- Mounting hole.

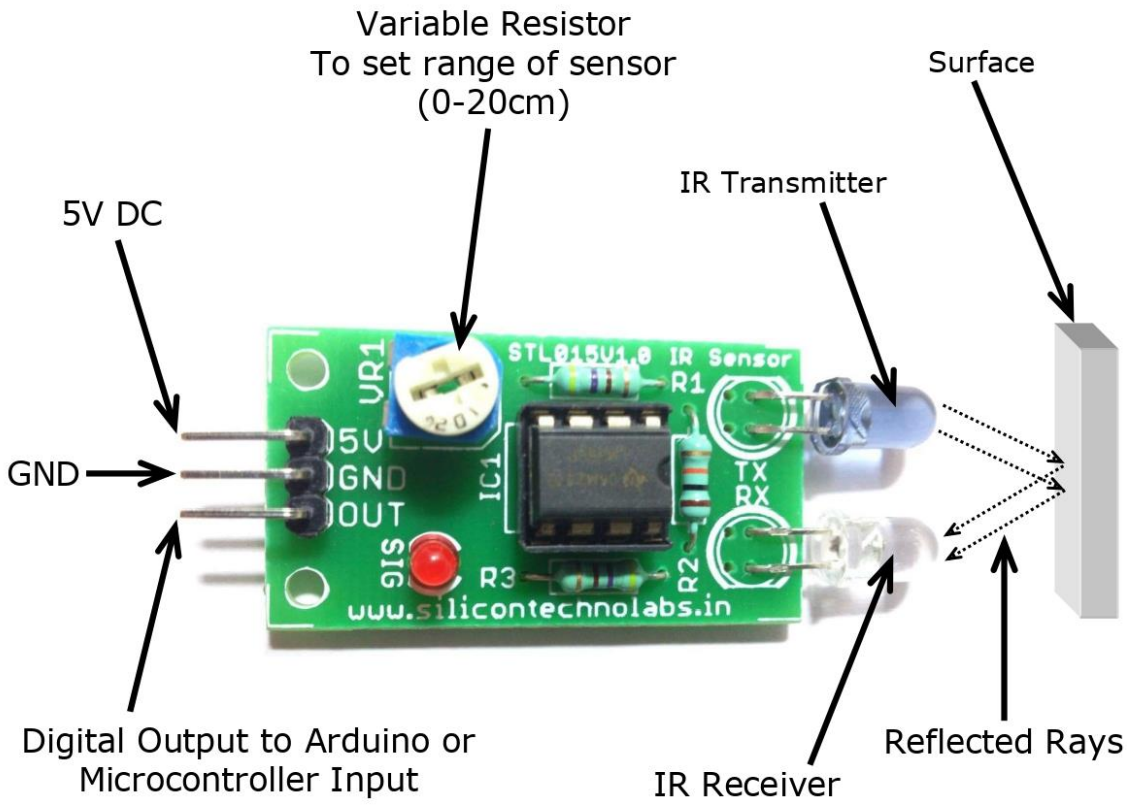
3. Specifications

- Size: 50 x 20 x 10 mm (L x B x H)
- Hole size: ϕ2.5mm

4. Schematics

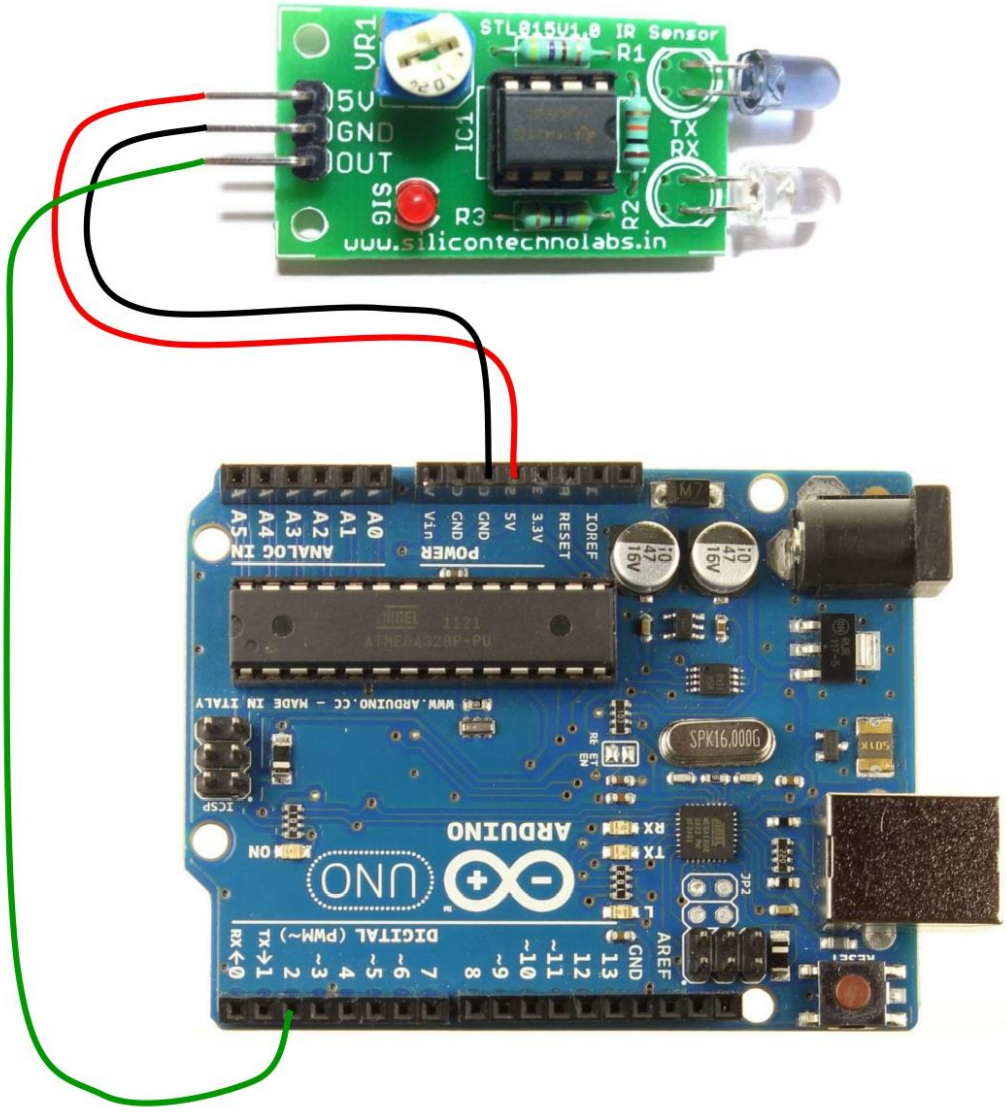


5. Hardware Details



6.Interface to Arduino

Now let's we build simple object counter using IR Proximity Sensor that's counts the Number of objects.Connect Silicon TechnoLabs IR Proximity Sensor to your arduino board as shown in below image.



2. LDR

Data Sheet

Light dependent resistors

NORP12 RS stock number 651-507
NSL19-M51 RS stock number 596-141

Two cadmium sulphide (cdS) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity. Applications include smoke detection, automatic lighting control, batch counting and burglar alarm systems.

Guide to source illuminations

Light source	Illumination (Lux)
Moonlight	0.1
60W bulb at 1m	50
1W MES bulb at 0.1m	100
Fluorescent lighting	500
Bright sunlight	30,000

Circuit symbol

Light memory characteristics

Light dependent resistors have a particular property in that they remember the lighting conditions in which they have been stored. This memory effect can be minimised by storing the LDRs in light prior to use. Light storage reduces equilibrium time to reach steady resistance values.

NORP12 (**RS** stock no. 651-507)

Absolute maximum ratings

Voltage, ac or dc peak	320V
Current	75mA
Power dissipation at 30°C	250mW
Operating temperature range	-60°C to +75°C

Electrical characteristics

T_A = 25°C. 2854°K tungsten light source

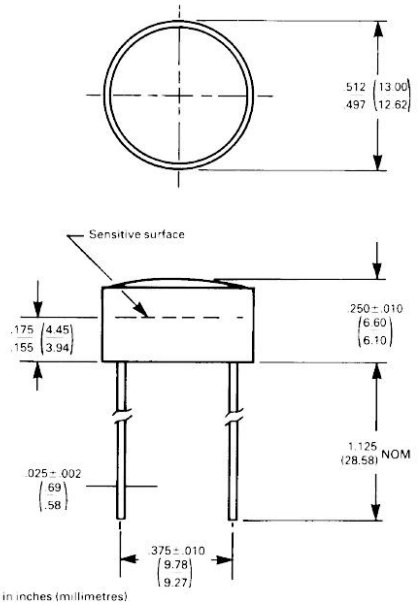
Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	1000 lux 10 lux	- -	400 9	- -	Ω kΩ
Dark resistance	-	1.0	-	-	MΩ
Dark capacitance	-	-	3.5	-	pF
Rise time 1	1000 lux 10 lux	- -	2.8 18	- -	ms ms
Fall time 2	1000 lux 10 lux	- -	48 120	- -	ms ms

1. Dark to 110% R_L
2. To 10 × R_L
R_L = photocell resistance under given illumination.

Features

- Wide spectral response
- Low cost
- Wide ambient temperature range.

Dimensions



Absolute maximum ratings

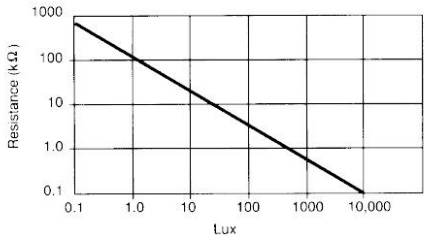
Voltage, ac or dc peak _____ 100V
Current _____ 5mA
Power dissipation at 25°C _____ 50mW*
Operating temperature range _____ -25°C +75°C

*Derate linearly from 50mW at 25°C to 0W at 75°C.

Electrical characteristics

Parameter	Conditions	Min.	Typ.	Max.	Units
Cell resistance	10 lux 100 lux	20 -	- 5	100 -	kΩ kΩ
Dark resistance	10 lux after 10 sec	20	-	-	MΩ
Spectral response	-	-	550	-	nm
Rise time	10ftc	-	45	-	ms
Fall time	10ftc	-	55	-	ms

Figure 4 Resistance as a function illumination



Dimensions

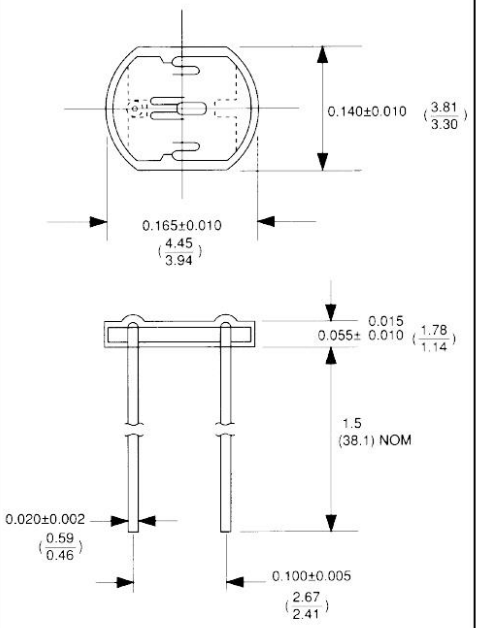
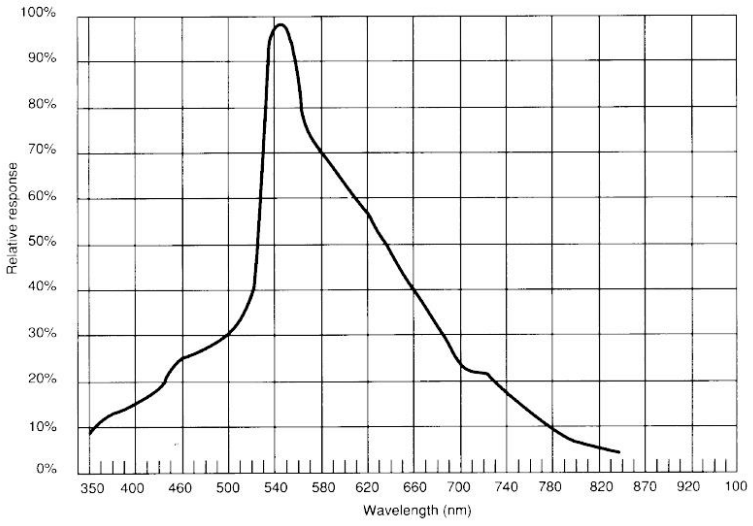


Figure 5 Spectral response



3. LM35

November 2000

National Semiconductor

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

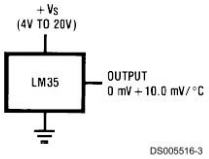
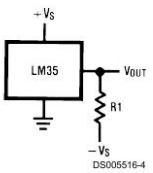


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S/50\ \mu A$
V_{OUT}=+1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

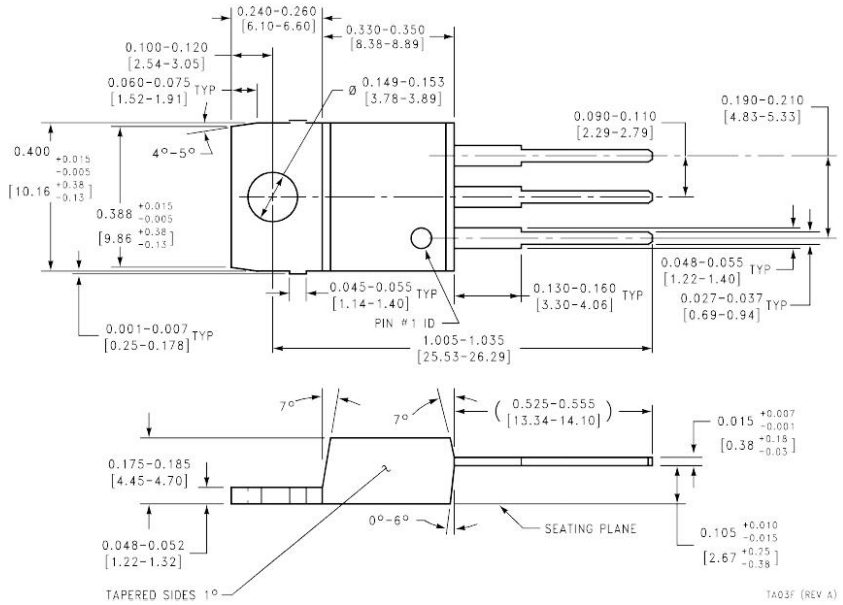
Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.;	
TO-46 Package,	−60˚C to +180˚C
TO-92 Package,	−60˚C to +150˚C
SO-8 Package,	−65˚C to +150˚C
TO-220 Package,	−65˚C to +150˚C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300˚C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260˚C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215˚C
Infrared (15 seconds)	220˚C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55˚C to +150˚C
LM35C, LM35CA	−40˚C to +110˚C
LM35D	0˚C to +100˚C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy	T_A=+25˚C	±0.2	±0.5		±0.2	±0.5		˚C
(Note 7)	T_A=−10˚C	±0.3			±0.3		±1.0	˚C
	T_A=T_{MAX}	±0.4	±1.0		±0.4	±1.0		˚C
	T_A=T_{MIN}	±0.4	±1.0		±0.4		±1.5	˚C
Nonlinearity (Note 8)	$T_{MIN} \le T_A \le T_{MAX}$	**±0.18**		**±0.35**	**±0.15**		**±0.3**	˚C
Sensor Gain (Average Slope)	$T_{MIN} \le T_A \le T_{MAX}$	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/˚C
Load Regulation (Note 3) $0 \le I_L \le 1$ mA	T_A=+25˚C	±0.4	±1.0		±0.4	±1.0		mV/mA
	$T_{MIN} \le T_A \le T_{MAX}$	**±0.5**		**±3.0**	**±0.5**		**±3.0**	mV/mA
Line Regulation (Note 3)	T_A=+25˚C	±0.01	±0.05		±0.01	±0.05		mV/V
	$4V \le V_S \le 30V$	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25˚C	56	67		56	67		µA
	V_S=+5V	**105**		**131**	**91**		**114**	µA
	V_S=+30V, +25˚C	56.2	68		56.2	68		µA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	µA
Change of Quiescent Current (Note 3)	$4V \le V_S \le 30V$, +25˚C	0.2	1.0		0.2	1.0		µA
	$4V \le V_S \le 30V$	**0.5**		**2.0**	**0.5**		**2.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	µA/˚C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	˚C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			˚C

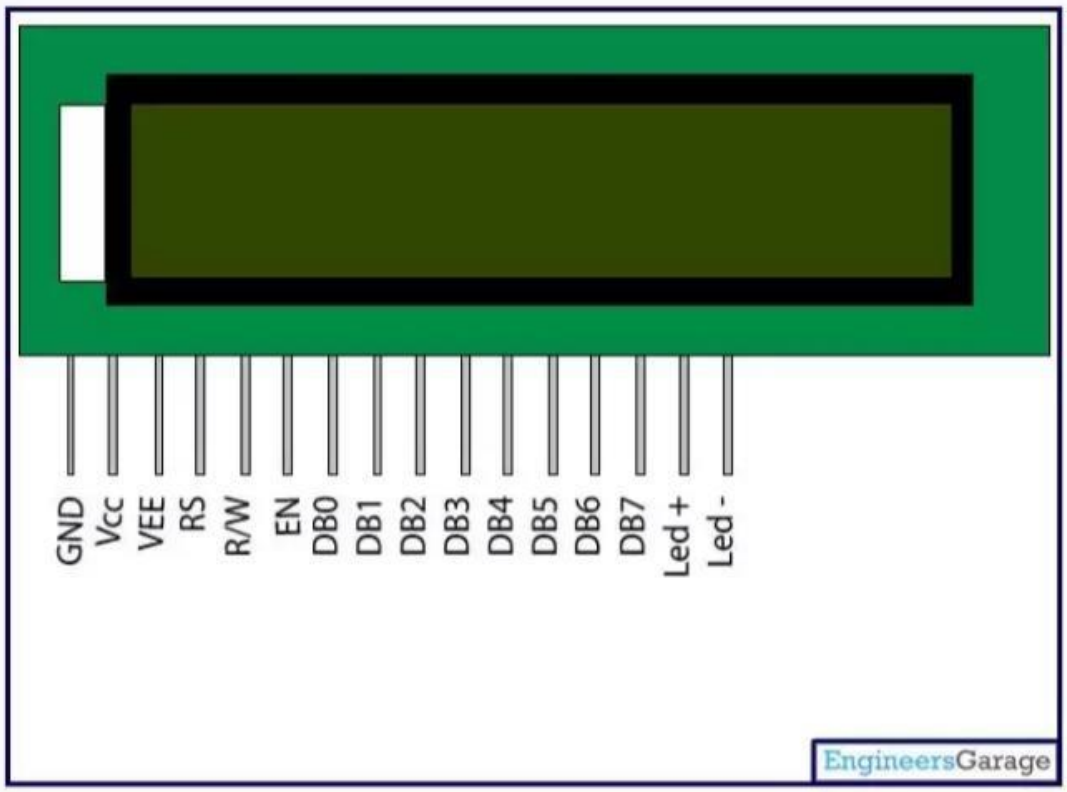
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Power Package TO-220 (T)
Order Number LM35DT
NS Package Number TA03F

4. 16X2 LCD Display

Pin Diagram:



Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

5. Servo Motor

TowerPro SG90 - Micro Servo

Basic Information

Modulation: Analog

Torque: **4.8V:** 25.0 oz-in (1.80 kg-cm)

Speed: **4.8V:** 0.10 sec/60°

Weight: 0.32 oz (9.0 g)

Dimensions:

Length: 0.91 in (23.1 mm)

Width: 0.48 in (12.2 mm)

Height:1.14 in (29.0 mm)

Motor Type: 3-pole
Gear Type: Plastic
Rotation/Support: Bushing

Additional Specifications

Rotational Range: 180°
Pulse Cycle: ca. 20 ms
Pulse Width: 500-2400 µs

6. 4X4 KeyPad

PARALLAX

Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

4x4 Matrix Membrane Keypad (#27899)

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

Features

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:
  Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
  Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Application Ideas

- Security systems
- Menu selection
- Data entry for embedded systems

Appendix B

Programming review

Programming languages are classified as:

1. Machine language
2. Assembly language
3. High level language

MACHINE LANGUAGE:

The language of 0's and 1's is called as machine language. The machine language is system independent because there are different set of binary instruction for different types of computer systems.

Limitations of machine languages:

It is very tedious and error prone process of writing programs in machine languages.

ASSEMBLY LANGUAGES:

It is low level programming language in which the sequence of 0s and 1s are replaced by mnemonic (ni-monic) codes. Typical instruction for addition and subtraction.

Example: ADD for addition, SUB for subtraction etc.

Since our system only understand the language of 0s and 1s. therefore a system program is known as assembler. Which is designed to translate an assembly language program into the machine language program?

HIGH LEVEL LANGUAGE:

High level languages are English like statements and programs. Written in these languages are needed to be translated into machine language before to their execution using a system software compiler.

Programs written in ***high level languages*** are much easier to maintain and modify as per requirements.

Sr.no	C++	Java	Python	C
1	Compiled Programming language	Compiled Programming Language	Interpreted Programming Language	Compiled Programming language
2	Supports Operator overloading	Does not support Operator Overloading	Supports Operator overloading	Supports Operator overloading
3	Provide both single and multiple inheritance	Provide partial multiple inheritance using interfaces	Provide both single and multiple inheritance	Provide both single and multiple inheritance
4	Platform dependent	Platform Independent	Platform Independent	Platform dependent
5	Does Not support threads	Has in build multithreading support	Supports multithreading	Does Not Supports thread

Appendix C
Trouble-Shooting

The first and major challenge that we faced was deciding the components with certain specifications required to make our project.

The second was interfacing IR sensor and understanding it's working.

Another challenge we faced was in writing code for counting of person at entry and exit and how to use one door as both entry and exit.

During Simulation, As TinkerCad Does not have Arduino Mega so instead of that we have to use Two Arduino Uno to Compensate for less no of Pins.

Even in Simulation we have problems with the operation of IR Sensors as if we send signal from IR remote and we wanted it to be Received by any Specific IR or Multiple IR sensors then also the Signal will go to the default IR sensor.

While making website there was many times when we got an error but that were fixed but among them there was on which was when we were adding Bootstrap Framework to make the website responsive the link that need to be added contained some random numbers we tried two to three times by copy pasting again and again but then we realize when we copy and paste some link from google to our editor which is visual studio it takes random number so we had to type that manually and then it resolved.

Appendix D
Real Life Mounting of Our system

The sensor will be mounted with a different power supply than the appliances.

We will use transformers for the sensors in our circuit as the sensors will not be able to handle the excessive voltage of the main household system.

The IR sensors will be mounted on the entry points of the room. Also along with the entry and exit points, IR will also be mounted inside rooms to detect the presence and keep the count of number of people present inside the room or hallway.

We will use copper wiring which will minimise effects of water seepage in the house. Most of the sensors are mounted inside rooms which already prevents them from weather and unexpected damages.

Appendix E
Real Life Scenarios

The main issue that we would face is that most of the sensors operate at very low temperatures than the actual main household voltage. This will lead us to modify our circuit.

The number of sensors would strictly depend upon the size of the room. If we typically use 2 IR sensors for an average sized room then there may be a possibility that the sensor is unable to detect the person due to the limited range of it.

The follow up problem would come up if we use more sensors than there are actually needed. The reading will differ in this case and it may give inaccurate results.