School of Engineering and Applied Science
Ahmedabad University
6th April 2019

Home Automation Project
Final Project Report

EMBEDDED SYSTEM DESIGN

In

ICT

By

Mudra Suthar(1741022)
Rutvi Tilala (1741091)
Hardi Trivedi(1741092)
Shivansh Nautiyal(1741037)

Under the guidance of

Prof. Anurag Lakhlani

Index

Acknowledgment

We extend our gratitude towards Prof. Anurag Lakhlani for sharing the academic knowledge related to the majority of the components which proves to be a vital part of this project. We thank our classmates for lending us components from time to time. Thank you to the ESD Team for allowing access to the lab for several hours and throughout the day.

1. INTRODUCTION

1.1 MOTIVATION::

Many of us might be thinking that how convenient would it be if our electrical appliances at our homes were smart enough. Like the fan can change its speed automatically on the basis of the temperature or the tube lights would reduce its intensities in case of the presence of the sunlight. This would make a huge difference at the micro level in the day to day chores. A Home automated system your life easy going as you won't have to go and change the settings of the electrical appliances manually.

Nowadays, the humankind is moving towards the new technologies by replacing the manual operations to automatically controlled devices. One of the basic requirements of the people during hot weather is a cooling fan. But, the speed of the fan can be controlled by manual operation using a manual switch namely fan regulator or dimmer. By turning the dimmer, the fan speed can be altered. It can be watched in some places like where the temperature is high during the morning, though the temperature falls down radically at night time. The users do not understand the difference in temperature. Same is the situation with the tube lights. The light intensity is way less in the presence of the daylight. So to overcome these problems we came up with the idea of making Home Automation System.

1.2 DESCRIPTION:

The whole Project can be divided into three systems:

1) Temperature controlled fan
2) Fading LED using an LDR
3) Controlling fan and intensity of LED on the basis of number of person in room

1)TEMPERATURE CONTROLLED FAN

In this system, a DHT22 sensor is used to sense the room temperature and then we adjust the speed of a DC fan/motor accordingly using PWM (Pulse Width Modulation). Arduino Uno is the heart of this project and an L293D driver IC is used to drive the DC fan/motor.

The final outcome to run the DC fan would be based on four different conditions. If the temperature is less than 25°C, then the DC fan will remain off and details will be displayed on the LCD. If the temperature is between 25°C and 30°C, then the DC fan will start working at a low speed (30% duty cycle). Similarly, if the temperature is between 30°C and 35°C, then the DC fan will run at a medium speed (60% duty cycle). If the temperature is greater than or equal to 35°C, then the DC fan will run at full speed (100% duty cycle).

2)FADING LED USING A LDR

In this system, a LDR, a LED and most importantly an arduino is used. This system determines the current/voltage allowed by light sensor to pass through it and outputs this voltage on digital pin of arduino to fade led the given output voltage is dependent on the current/voltage allowed to pass by light sensor(Or intensity of light). Sp when the intensity of the day light will be more the brightness of the led will be less and so on.

3) CONTROLLING SPEED OF THE FAN AND INTENSITY OF LED ON THE BASIS OF PEOPLE PRESENT IN THE ROOM

In this system, two IR sensors are used which determine the number of people entering and leaving the room. One IR sensor is placed at the entry of the room which basically behave as “up counter” by incrementing the number of person entering the room. The IR sensor which is placed at exit of the room is "down counter" which decrement the count of person entered in the room. On the basis of number of person present in the room we control the speed of the fan which is also determined by temperature of the place and intensity of light. Also, we are displaying number of person present in the room on LCD.

COMPONENTS USED

The following components will be used by us in this project

Hardware Requirements

- Arduino MEGA
- L293D Motor Driver IC
- DHT22 Sensor
- 16×2 LCD Display
- DC Fan/motor
- 9V Battery
- 10KΩ Potentiometer
- 220Ω Resistor
- Breadboard
- LDR
- LED
- Two IR sensors

Software Requirements

- Arduino CC to edit and upload new code

2. MARKET SURVEY

There have been some products in the market which work on the concept of controlling a fan based on the current temperature of the room. None of the exact products in the market are available for sale.

There is a CPU fan, which works on the same principle as our topic of the project at much higher cost. The product can be controlled according to the ambient temperature of the fan speed, no manual control, automatic stepless speed, the higher the ambient temperature, the faster the fan speed control, the ambient temperature is below the set the start temperature, the governor can control the fan stops running, in order to save power and eliminate unwanted noise generated by the fan.

Available on : Amazon.com
(Here are some snapshots of the same)

Product description

This DC12V governor (JCF1201 type) can be controlled according to the ambient temperature of the fan speed, no manual control, automatic stepless speed, the higher the ambient temperature, the faster the fan speed control, the ambient temperature is below the set the start temperature, the governor can control the fan stops running, in order to save power and eliminate unwanted noise generated by the fan. Variety of 2, 3, 4-wire 12V fans are applicable.

This governor's main features:
1. With anti-reverse power protection, power line reversed, the board will not burn.
2. New dual-mode function: fixed temperature mode and adjustable temperature mode.
3. Fit for 2,3,4 line 12V fan, stepless temperature control can be achieved.
4. Single temperature sensor, easier to use, more responsive control.
5. Fixed mode temperature at 30 degrees, the fan speed control starts from 30 to 50 degrees fan speed linear increase, 50 degrees fan at full speed.
6. Start mode adjustable temperature range: 15 to 50 degrees, the full speed range of 35 to 70 degrees (start-up and full of temperature difference of 20 degrees).
7. The maximum operating temperature of 110 degrees temperature probe beyond this temperature, the probe will be damaged! !
8. On board there is a mode switch, the switch to the F position, the governor fixed temperature mode, switch to the A position, the governor of temperature adjustable mode.
9. Temperature adjustment potentiometer, when the mode switch is in the A position, rotating the potentiometer, you can change the fan-start temperature.
10. Board size: 45x25x30MM (LXWXH), Sensor probe length: 325mm

Package include:
1x Fan Temperaure Control Speed Controller with Sensor Probe

American Journal of Engineering Research (AJER) published a paper, which threw light upon the function of a microcontroller in building a temperature-controlled fan.[1]

Abstract: Automation is a field that continues to expand and generate interest. This is so as a result of the constant need to reduce human effort, especially in the control of electrical appliances. This work involves the construction of a circuit using a temperature sensor and a microcontroller that will automatically control the speed of the fan whenever there is a change in ambient temperature. An Arduino program is used to program the microcontroller based on the desired function. The Arduino language is an open source project that creates microcontroller-based kit for building digital device and interactive objects that can sense physical quantities and control devices. The original control unit of the fan is disabled and replaced with the constructed circuit. The fan is given 3 different speed levels, with each speed level being activated by a certain degree of temperature change. This will eventually reduce human stress as well as reduce energy wastage.

-A somewhat similar research paper was published. It was about a temperature controlled fan speed controller using PT-100.[2]

Abstract: Now a day's technology is running with time, it completely occupied the life style of human beings. Even though there is such an
Importance for technology in our routine life there are even people whose life styles are very far to this well known term technology. So it is our
responsibility to design few reliable systems, which can be even efficiently used by them. Automatic Room Temperature Controlled Fan Speed
Controller is one of them. The developed system provides an environment in which no user needed to control the fan speed. Automatically control

the fan speed by sensing the room temperature. These fascinating efforts to create intelligent system are to provide human being a more
convenient life. The circuit was designed using electronic components available in local market to keep the cost at low level.

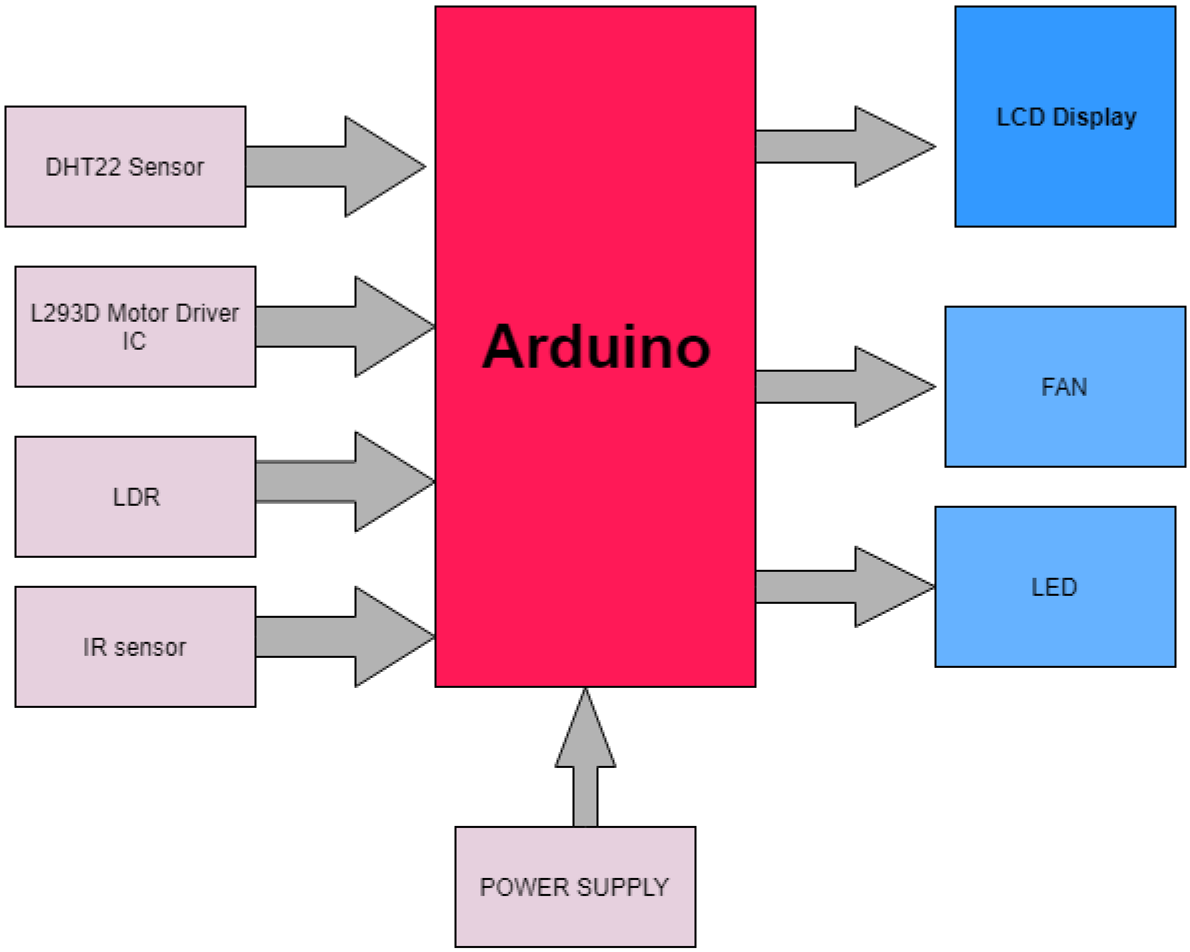
3. BLOCK DIAGRAM

EXPLANATION

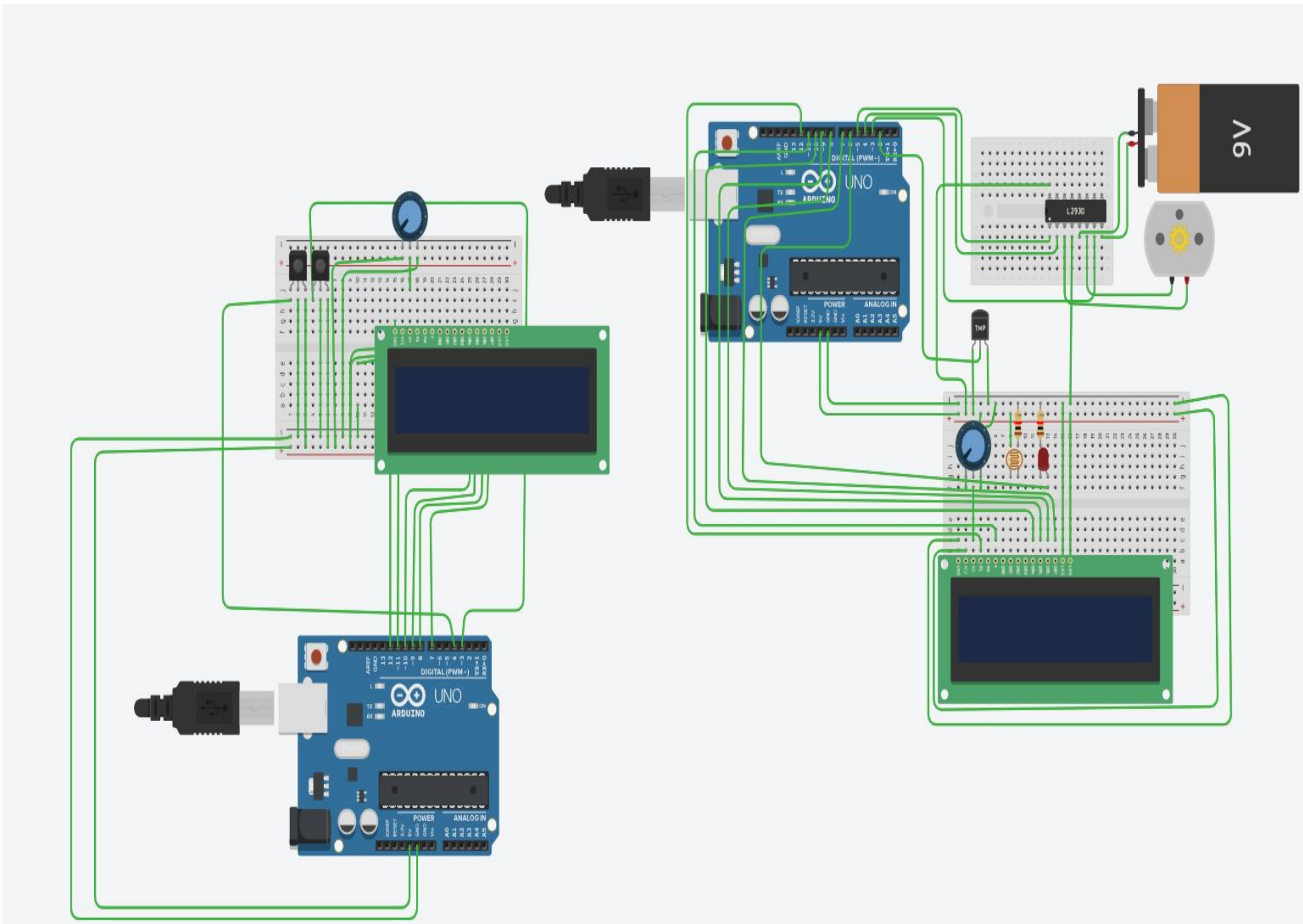
As we want our system automatically control the speed of fan and the intensity of the light we will be needing the following sensors to collect the input data:

1)DHT 11 (Temperature sensor)
2)LDR (Light dependent Resistor)
3) IR sensor

We will show the output on the following:
1)LED 2)LCD 3) DC Motor(fan)


DHT22 Sensor
L293D Motor Driver IC
LDR
IR sensor
Arduino
LCD Display
FAN
LED
POWER SUPPLY

4. CIRCUIT DIAGRAM



(As the tools used by us didn't had Arduino Mega Microcontroller , We have used two Arduino Uno in here, But our model is made on Arduino Mega)

EXPLANATION

The connections of the First Arduino (Which Controls the Fan and LED)

First, we can connect the L293D motor driver IC to Arduino as below.

- Connect pin 1 (Enable 1) of L293D to pin 5 of Arduino.
- Connect pin 2 (Input 1) of L293D to the pin 4 of Arduino.
- Connect pin 3 (Output 1) of L293D to one terminal of the DC motor.

- Pins 4 and 5 of the L293D are ground pins, connect these to common ground (battery ground + Arduino ground).
- Connect pin 6 (Output 2) of L293D to the remaining terminal of the DC motor.
- Connect pin 7 (Input 2) of L293D to the pin 3 of Arduino.
- Pin 8 is Vcc2, which is the driver/motor power input, connect it to positive of the battery.
- Pin 16 is Vcc1, which is logic voltage input (voltage level of control signals provided by Arduino). Connect it to 5V output of Arduino.

After that connect the DHT22 sensor to the Arduino, it is using a single wire bus for communication.

- The first pin is VCC power input, connect it to 5V output of Arduino.
- The second pin is DATA output, connect it to pin 2 of Arduino.
- The third pin is not used.
- And last, the ground pin is connected to the ground.

Now we can connect 16×2 LCD 1 to the Arduino.(Temp and fan speed Display)

- Pin 1 is VSS, connect it to the ground.
- Pin 2 is VDD, connect it to 5V output of Arduino.
- Pin 3 is VEE, for adjusting display contrast. Connect it to the variable terminal of a potentiometer whose fixed terminals are connected to ground and 5V.

- Pin 4 is RS (Register Select), it is used to select data or command register. Connect it to pin 12 of Arduino.
- Pin 5 is R/W (Read/Write). Connect it to ground since we are only writing data to LCD in this project.
- Pin 6 is EN (Enable), it is used to indicate a valid data/command in data lines (D0 ~ D7). Connect it to pin 11 of Arduino.
- Pin 7 ~ 10 are data pins (D0 ~ D3), used to transmit data/command to LCD controller. But these pins are not used in 4 bit LCD interfacing, so connect it to ground.
- Pin 11 ~ 14 are data pins (D4 ~ D7), used to transmit data/commands to the LCD controller. Connect it to Arduino pins 10, 9, 8 and 7 respectively.

The connection for the LDR and LED is as follows

- One terminal of LDR is connected to the analog input pin A0 and the other is connected to the ground via register
- Connect led anode to pin# 9 of arduino and cathode to ground in series with a resistor

The connections of the Second Arduino (Which controls the counter)

The connection of the Led 2 is.(Counter Display)

- Pin 1 is VSS, connect it to the ground.
- Pin 2 is VDD, connect it to 5V output of Arduino.

- Pin 3 is VEE, for adjusting display contrast. Connect it to the variable terminal of a potentiometer whose fixed terminals are connected to ground and 5V.
- Pin 4 is RS (Register Select), it is used to select data or command register. Connect it to pin 12 of Arduino.
- Pin 5 is R/W (Read/Write). Connect it to ground since we are only writing data to LCD in this project.
- Pin 6 is EN (Enable), it is used to indicate a valid data/command in data lines (D0 ~ D7). Connect it to pin 11 of Arduino.
- Pin 7 ~ 10 are data pins (D0 ~ D3), used to transmit data/command to LCD controller. But these pins are not used in 4 bit LCD interfacing, so connect it to ground.
- Pin 11 ~ 14 are data pins (D4 ~ D7), used to transmit data/commands to the LCD controller. Connect it to Arduino pins 10, 9, 8 and 7 respectively.

The connection of the IR sensors are as follows:

- IR1- out- pin4 , Vc to 5v of arduino and , Gnd to gnd of the Arduino
- IR2- out- pin5 , Vc to 5v of arduino and , Gnd to gnd of the Arduino

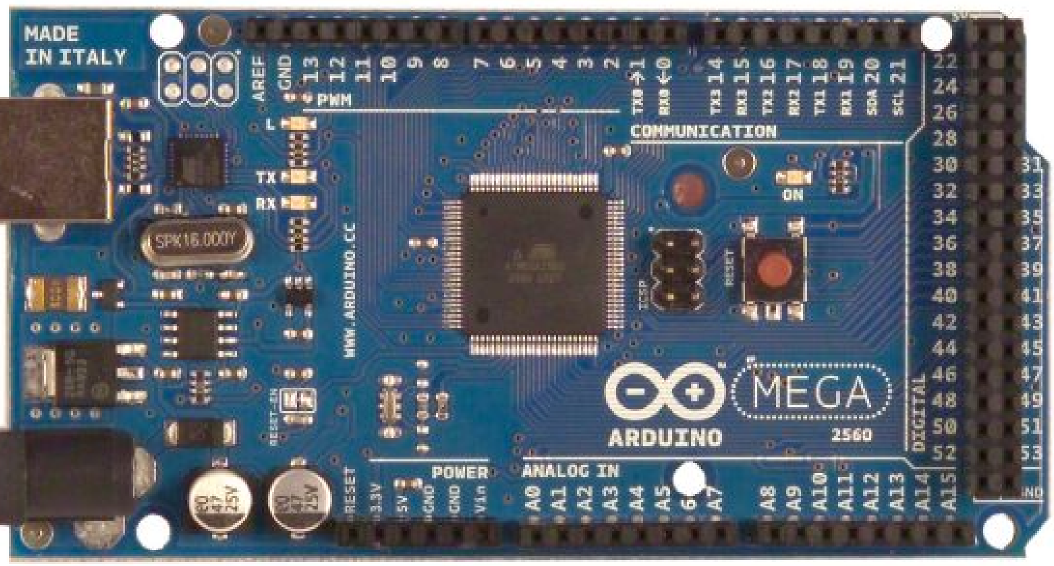
5). ARDUINO (mega)

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a

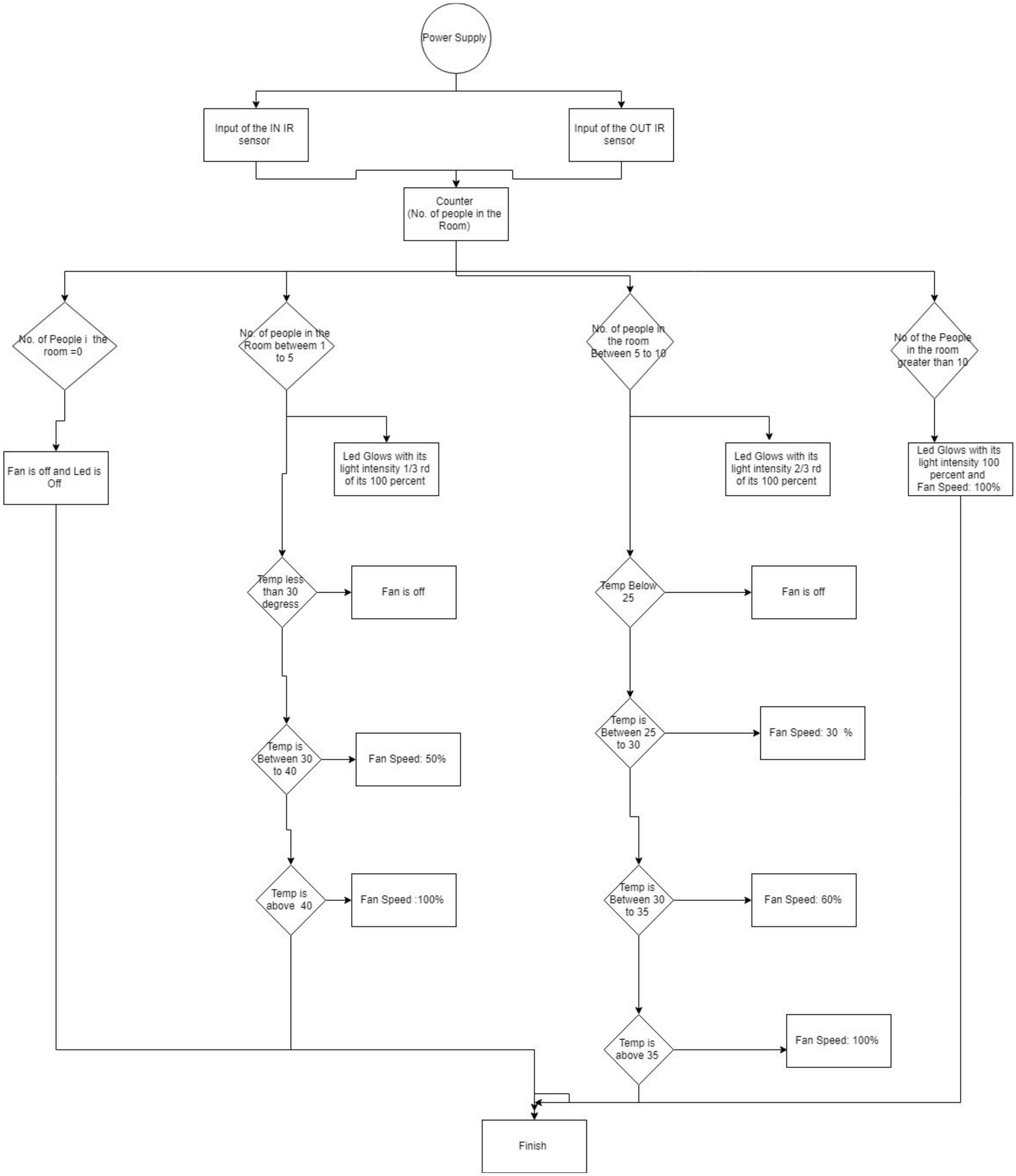
USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

The Mega 2560 is an update to the Arduino Mega, which it replaces.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g


MADE
IN ITALY
AREF
GND
PWM
COMMUNICATION
L
TX
RX
ON
SPK16.000Y
WWW.ARDUINO.CC
ICSP
RESET
MEGA
ARDUINO
2560
DIGITAL
POWER
ANALOG IN

6) PROGRAM FLOWCHART



7) SENSORS

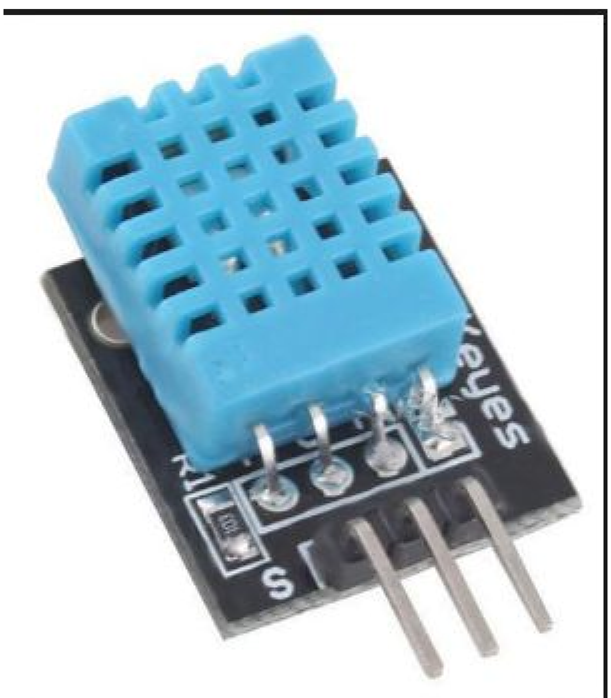
1.DHT11 sensor:

DHT11 is a low-cost humidity and temperature sensor which provides high reliability and long term stability. The DHT11 Humidity and Temperature Sensor consists of 3 main components. A resistive type humidity sensor, an NTC (negative temperature coefficient) thermistor (to measure the temperature) and an 8-bit microcontroller, which converts the analog signals from both the sensors and send out a single digital signal.

DHT11 sensor is used to read the temperature to control fan speed. This sensor is very easy to use and has very good accuracy compared to other sensors.

The specifications of DHT 11 is given as follows:

PCB size	22.0mm X 20.5mm X 1.6mm
Working voltage	3.3 or 5V DC
Operating voltage	3.3 or 5V DC
Measurement range	20-95%RH；0-50℃
Resolution	8bit（temperature），8bit（humidity）
Compatible interfaces	2.54 3-pin interface and 4-pin Grove interface(1)



2. LDR(LIGHT DEPENDENT RESISTORS)

Light dependent resistors, LDRs or photoresistors are often used in circuits where it is necessary to detect the presence or the level of light.
A photoresistor or light dependent resistor is a component that is sensitive to light. When light falls upon it then the resistance changes. Values of the resistance of the LDR may change over many orders of magnitude the value of the resistance falling as the level of light increases.
LDRs are made from semiconductor materials to enable them to have their light sensitive properties. Many materials can be used, but one popular material for these photoresistors is cadmium sulphide, CdS.

Specification:

Max power dissipation	200mW
Max voltage @ 0 lux	200V
Peak wavelength	600nm
Min. resistance @ 10lux	1.8kΩ
Max. resistance @ 10lux	4.5kΩ
Typ. resistance @ 100lux	0.7kΩ
Dark resistance after 1 sec	0.03MΩ
Dark resistance after 5 sec	0.25MΩ

3. IR SENSORS

IR Sensor module has great adaptive capability of the ambient light, having a pair of infrared transmitter and the receiver tube, the infrared emitting tube to emit a certain frequency, encounters an obstacle detection direction (reflecting surface), infrared reflected back to the receiver tube receiving, after a comparator circuit processing, the green LED lights up, while the signal output will output digital signal (a low-level signal), through the potentiometer knob to adjust the detection distance, the effective distance range 2 ~ 10cm working voltage of 3.3V-5V. The detection range of the sensor can be adjusted by the potentiometer, with little interference, easy to assemble, easy to use features, can be widely used robot obstacle avoidance, obstacle avoidance car assembly line count and black-and-white line tracking and many other occasions.

Features of IR Sensor Module:-

- When the module detects obstacles in front of the signal, the circuit board green indicator light level, while the OUT port continuous output low-level signals, the module detects a distance of 2 ~ 10cm, detection angle 35 °, the detection distance can be potential adjustment with adjustment potentiometer clockwise, the increase in detection distance; counterclockwise adjustment potentiometer, the detection distance decreased.
- the sensor active infrared reflection detection, target reflectivity and shape of the detection distance of the key. The black minimum detection range, white maximum; small area object distance is small, a large area from the large.
- The sensor module output port OUT can be directly connected with the microcontroller IO port can also be driven directly to a 5V relay; Connection: VCC-VCC; GND-GND; OUT-IO.
- The comparator using LM393, stable.
- 3-5V DC power supply module can be used. When the power is turned on, the red power LED is lit.
- With the screw holes of 3mm, easy to install.
- Board size: 3.1CM * 1.5CM.
- Each module in the delivery has threshold comparator voltage adjustable via potentiometer, special circumstances, please do not adjust the potentiometer.

Connection of IR sensors:

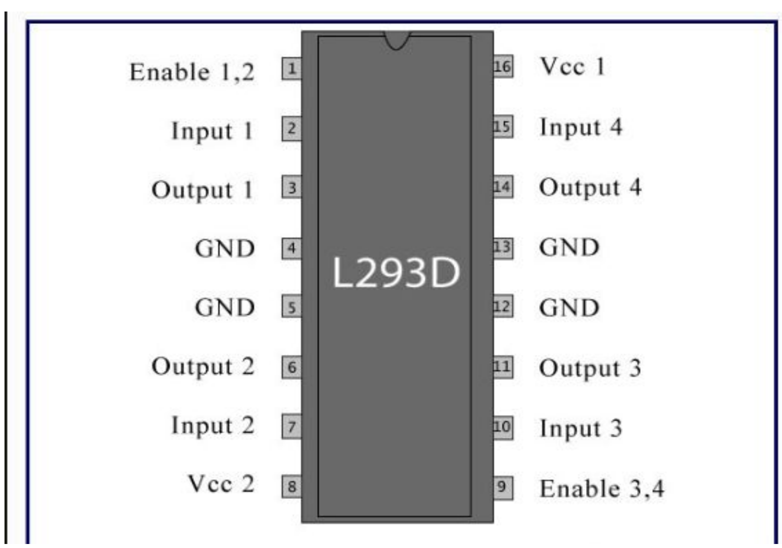
- VCC external 3.3V-5V voltage (can be directly connected with the a 5v microcontroller and 3.3v microcontroller).
- GND external GND.
- OUT board digital output interface (0 and 1).

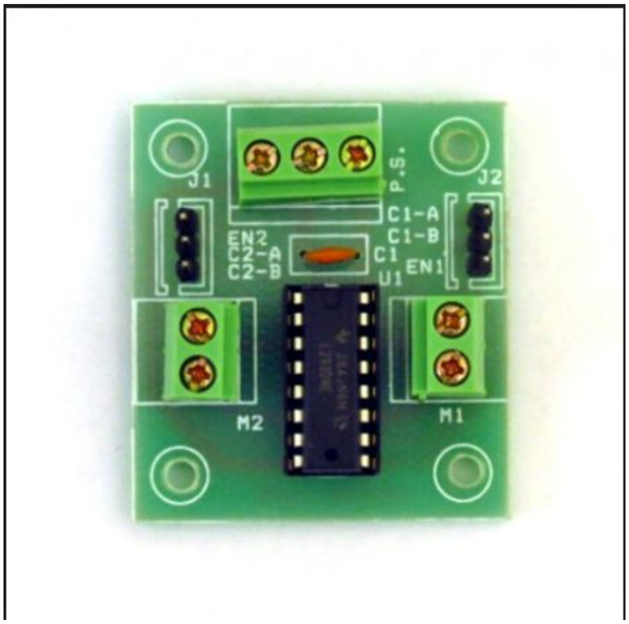
8) ACTUATORS AND DISPLAYS

L293D Motor Driver IC: We are using L293D motor driver IC for controlling DC fan/motor with Arduino which can drive 2 DC motors. So, for the future work if we want to control two fans at the same time, then it is of immense use.

Specifications

- Supply Voltage Range 4.5V to 36V
- 600-mA Output current capability per driver
- Separate Input-logic supply
- It can drive small DC-geared motors, bipolar stepper motor.
- Pulsed Current 1.2-A Per Driver
- Thermal Shutdown
- Internal ESD Protection
- High-Noise-Immunity Inputs

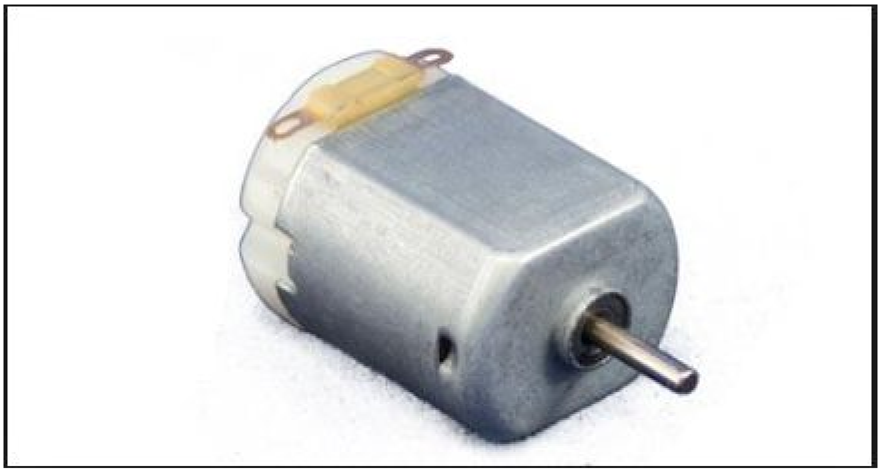




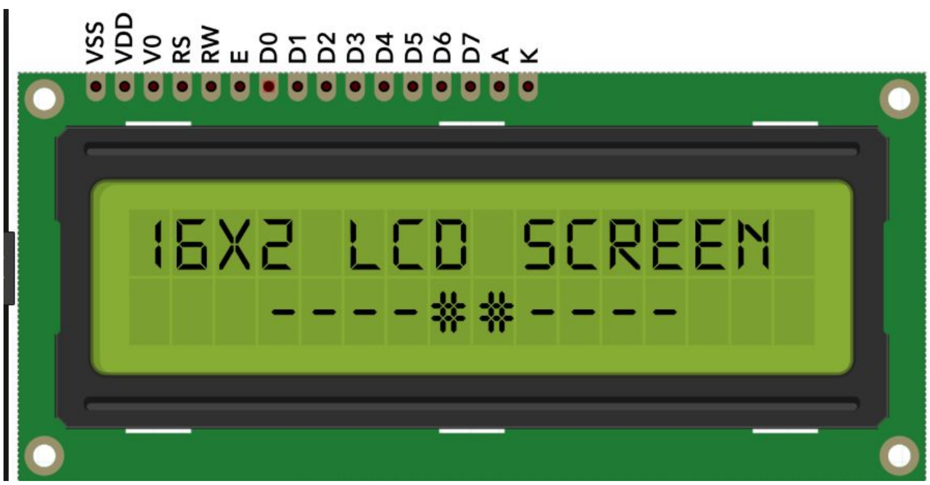
DC motor: Here, we have selected DC motor which can be treated as a fan and we can control the speed by providing PWM signals on the basis of change in temperature.

Specifications

Parameter	Value	Unit
Input Voltage	12	V
Speed	26	Rpm
Torque	588	mNm
Weight	160	G
Power	1.1	W
Diameter	37	mm
Length	27	mm



LCD Display: Liquid crystal display (LCD), an electronic display device that is used to display temperature and fan speed at particular an instance of the time. We have used 16*2 LCD dispaly.



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

LED

A light-emitting diode is a semiconductor light source that emits light when current flows through it.

10. COMPLETE C PROGRAM

```
#include "DHT.h"          // Including DHT22 library
#include "LiquidCrystal.h" // Including LCD library
#include "Adafruit_Sensor.h" // Including the DHT22 library

#define DHTPIN 2     // Declaring pin 6 for communicating to DHT22
sensor
#define DHTTYPE DHT11 // Declaring the type of DHT sensor we
are using (DHT22 or DHT11)

DHT dht(DHTPIN, DHTTYPE);         // Declaring DHT connection
and type
LiquidCrystal lcd_count(12,11,10,9,8,7); // Declaring LCD
connections for counting no. of people
LiquidCrystal lcd_temp(22,23,24,25,26,27);// Declaration for LCD
connections for displaying temp and fan speed



int LDR = A0;
int LED =  44;
```

```
int count=0;

int Motor_Pin1 = 29;  // Input 1 on L293D
int Motor_Pin2 = 28;  // Input 2 on L293D
int Enable = 30;      //Enable 1 on L293D

void IN()
{
   count++;
   lcd_count.clear();
   lcd_count.print("Person In Room:");
   lcd_count.setCursor(0,1);
   lcd_count.print(count);

}

void OUT()
{
  count--;
   lcd_count.clear();
   lcd_count.print("Person In Room:");
   lcd_count.setCursor(0,1);
   lcd_count.print(count);

}

void setup()
{
  //temperature controlled fan section

  pinMode(Motor_Pin1, OUTPUT);
  pinMode(Motor_Pin2, OUTPUT);
```

```
  pinMode(Enable, OUTPUT);
  lcd_temp.begin(16,2); // Initializes the 16x2 LCD
  dht.begin();     // Initializes DHT sensor

  // ir sensor and led lights section
  lcd_count.begin(16,2);
  lcd_count.print("Visitor Counter");
  delay(2000);
  pinMode(4, INPUT);
  pinMode(5, INPUT);
  pinMode(6, OUTPUT);
  lcd_count.clear();
  lcd_count.print("Person In Room:");
  lcd_count.setCursor(0,1);
  lcd_count.print(count);
}

void loop()
{

  if(digitalRead(4)== LOW )
  IN();
  if(digitalRead(5)== HIGH)
  OUT();



  if(count<=0)
  {
    //temperature section
    analogWrite(Enable,0);  // fan is off
    lcd_temp.clear();
```

```
    lcd_temp.print("Nobody In Room");
    lcd_temp.setCursor(0,1);
    lcd_temp.print("Fan Is Off");
    delay(200);

    // counter section
    lcd_count.clear();
    digitalWrite (LED , LOW );
    lcd_count.clear();
    lcd_count.print("Nobody In Room");
    lcd_count.setCursor(0,1);
    lcd_count.print("Light Is Off");


  }
 else{
    lcd_temp.clear();                        // Clear LCD
    float temp = dht.readTemperature(); // Reading the temperature in
Celsius

    // Validating received data
    if (isnan(temp)) {

      lcd_temp.print("Failed to read");
      delay(1000);
      return;
        }
   // setting the lcd_temp
    lcd_temp.setCursor(0,0);
    lcd_temp.print("Per:");
    lcd_temp.print(count);      // Writing temperature in the first row
    lcd_temp.print("Tmp:");
```

```
    lcd_temp.print(temp);      // Writing temperature in the first row
    lcd_temp.print(" C");
    lcd_temp.setCursor(0,1);   // Setting the position to display fan
speed

    if (count > 1  && count < 5)
          {
           int reading=analogRead(LDR);
           int bright=reading;
           delay(500);
           analogWrite(LED, bright);

           if(temp <30 ) {                // If the temperature less than 30

              analogWrite(Enable,0);        // 0% PWM duty cycle
              lcd_temp.print("Fan OFF ");
              delay(100);
                        }
           else if(temp>=30 & temp<40) {      // If the temperature is
between 30 & 40

              analogWrite(Enable, 127);        // 50% of maximum duty
cycle value (255).
              lcd_temp.print("Fan Speed:50% ");
              delay(100);
               }

           else if(temp>=40) {              // If the temperature is above 40

              analogWrite(Enable, 255);        // 100% duty cycle
              lcd_temp.print("Fan Speed: 100%   ");
              delay(100);
```

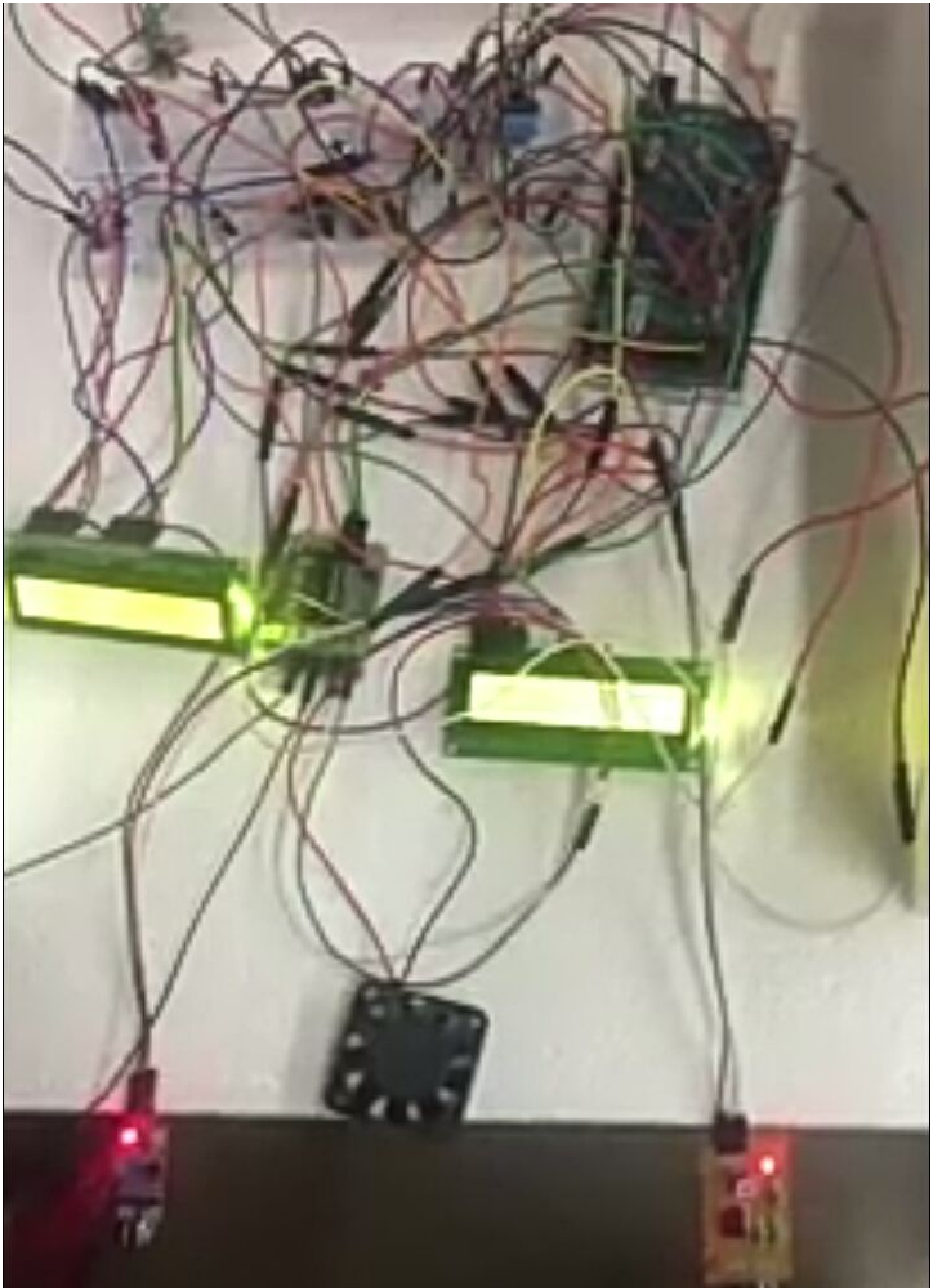
```
            }
          }

    else if (count > 5  && count < 10)
          {
          int reading=analogRead(LDR);
          int bright=reading;
          delay(500);
          analogWrite(LED, bright);

          if(temp <25 ) {                // If the temperature less than 25

             analogWrite(Enable,0);        // 0% PWM duty cycle
             lcd_temp.print("Fan OFF ");
             delay(100);
  }
          else if(temp>=25 & temp<30) {      // If the temperature is
between 25 & 30

             analogWrite(Enable, 77);         // 30% of maximum duty
cycle value (255).
             lcd_temp.print("Fan Speed: 30% ");
             delay(100);
  }
          else if(temp>=30 & temp<35) {      // If the temperature is
between 30 & 35

             analogWrite(Enable, 157)        // 60% of maximum duty
cycle value (255).
             lcd_temp.print("Fan Speed: 60% ");
             delay(100);
  }
```

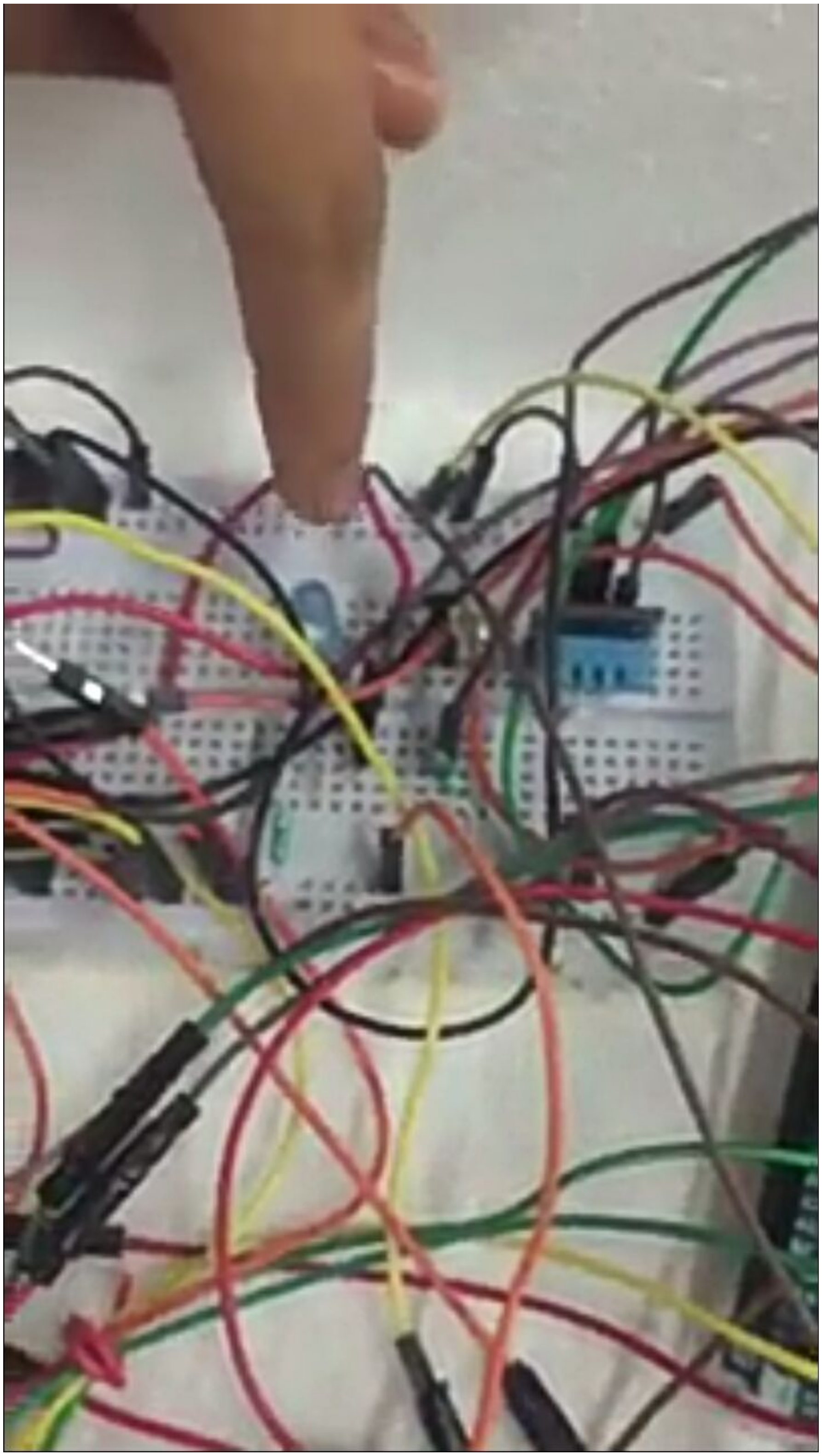
```
          else if(temp>=35) {              // If the temperature is above 35

             analogWrite(Enable, 255);       // 100% duty cycle
             lcd_temp.print("Fan Speed: 100%   ");
             delay(100);
  }


           }

    else if (count > 10 )
          {
          int reading=analogRead(LDR);
          int bright=reading;
          delay(500);
          analogWrite(LED, bright);
          analogWrite(Enable, 255);       // 100% duty cycle
          lcd_temp.print("Fan Speed: 100%   ");
          delay(100);
           }

 }
   digitalWrite(Motor_Pin1, LOW);  // To drive the motor in a
particular direction
   digitalWrite(Motor_Pin2, HIGH); // To drive the motor in a
particular direction

   delay(2000); // 2 seconds delay



}
```

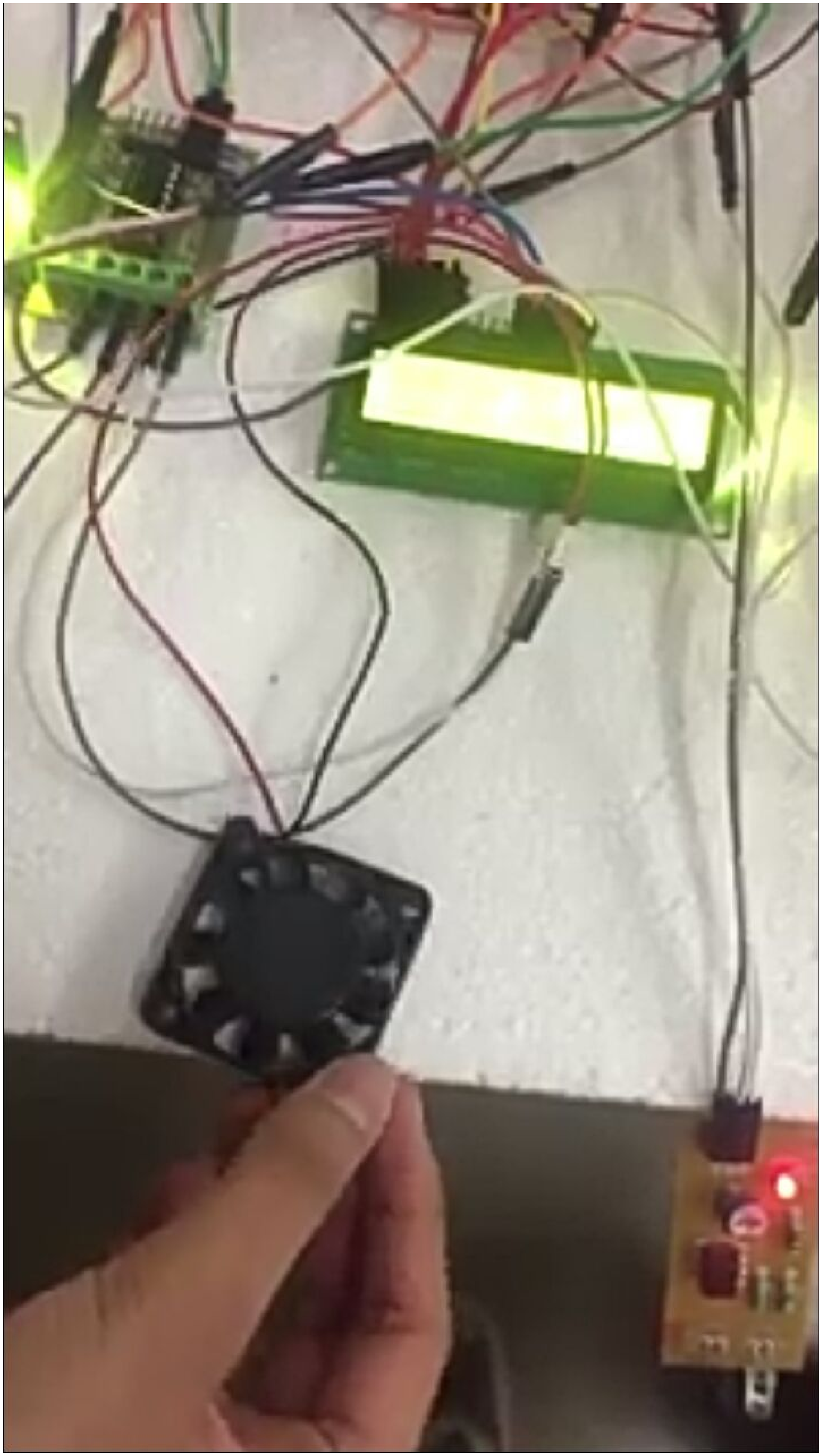
11. SUMMARY

Our group is working on a project based on home automation. A home automation system can control lighting, climate, entertainment systems, and appliances. It may also include home security such as access control and alarm systems when connected with the Internet; home devices are an important constituent of the Internet of Things. A home automation system typically connects controlled devices to a central hub or "gateway". While there are many competing vendors, there are very few worldwide-accepted industry standards and the smart home space is heavily fragmented. Manufacturers often prevent independent implementations by withholding documentation and by litigation.
We aim to contribute to this industry, which has tremendous potential.
We have a system where we are working with two appliances, fan and light bulb (led). Power consumption in the world has increased and is increasing continuously. We automated the system of controlling fans and lights. In our model, we have used two IR sensors. One IR sensor senses the number of persons entering the room. It keeps a track of the number of person and keeps incrementing as the number of persons increase. Similarly another IR sensor tracks the number of persons leaving the room. As people leave it keeps the track and reduces the count of the first IR sensor. This way we have the exact number of people who are present in the room. There is a temperature sensor, which senses the temperature in the room. What exactly it does is that whenever a person enters the room, the temperature sensor senses the temperature of the room and starts the fan at a certain speed, say 25%. If the temperature is high, the fan will rotate at comparatively higher speed. If number of persons in the room increase, the speed of the fan also increases. This is to reduce the humidity and maintain the airflow when there are too many people in the room. The light works in a same way. As the number of people in the room increase, the intensity of the led increases. As the number of people in the room leave, the intensity of the fan and even the led decrease. This will help us reduce the human efforts of switching on and off the fan and lights every time they leave the room. This helps the environment by controlling the consumption of electricity.

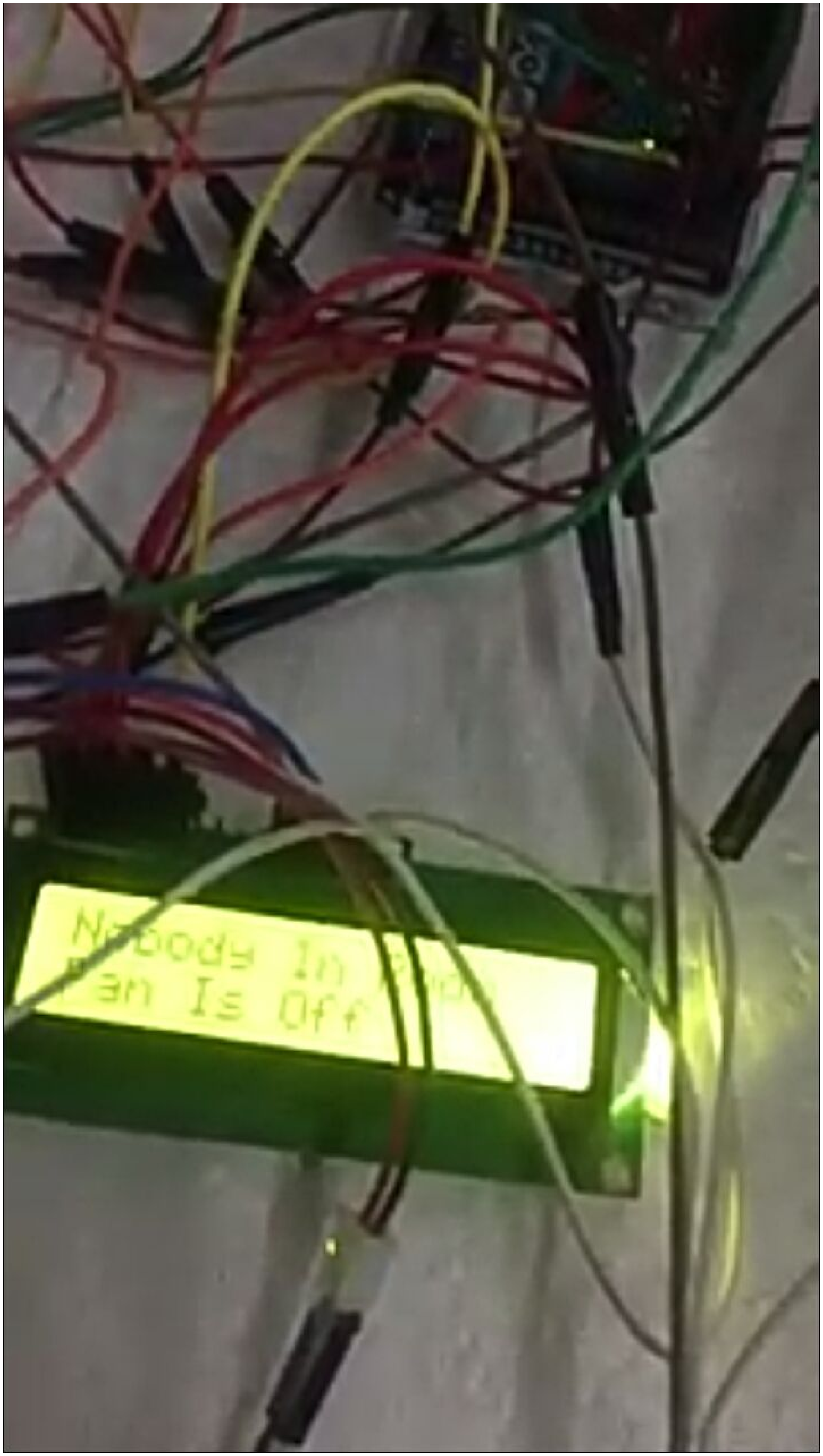
Initially, there will be no person present in the room which implies that fan and light will remain switched off till the person enters the room

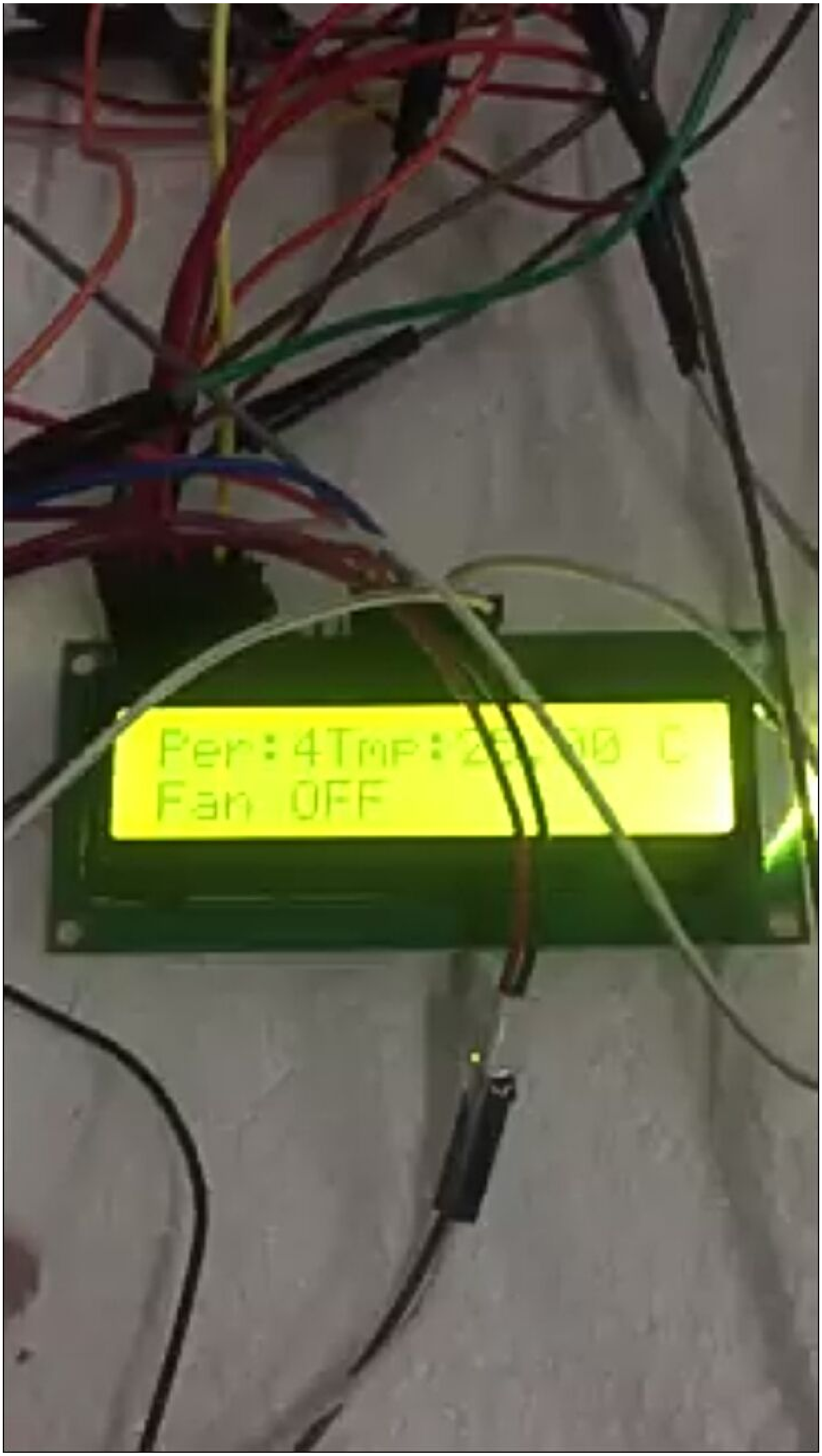




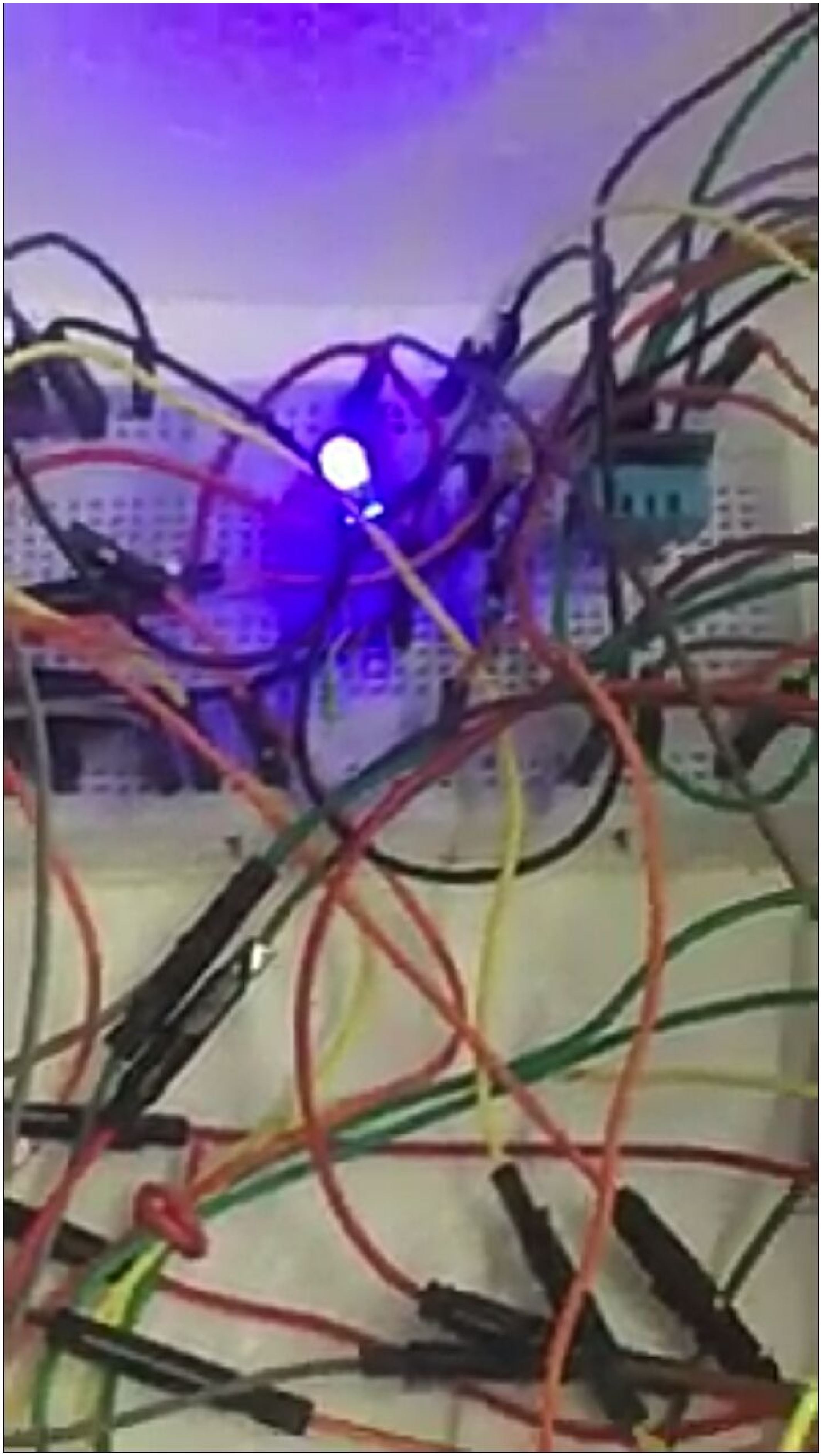


Here, two LCD are used for tracking number of person present in the room and temperature of the room with the status of fan(i.e speed of the fan which can be 0%, 30%, 60% and 100% controlled with temperature parallelly) respectively.

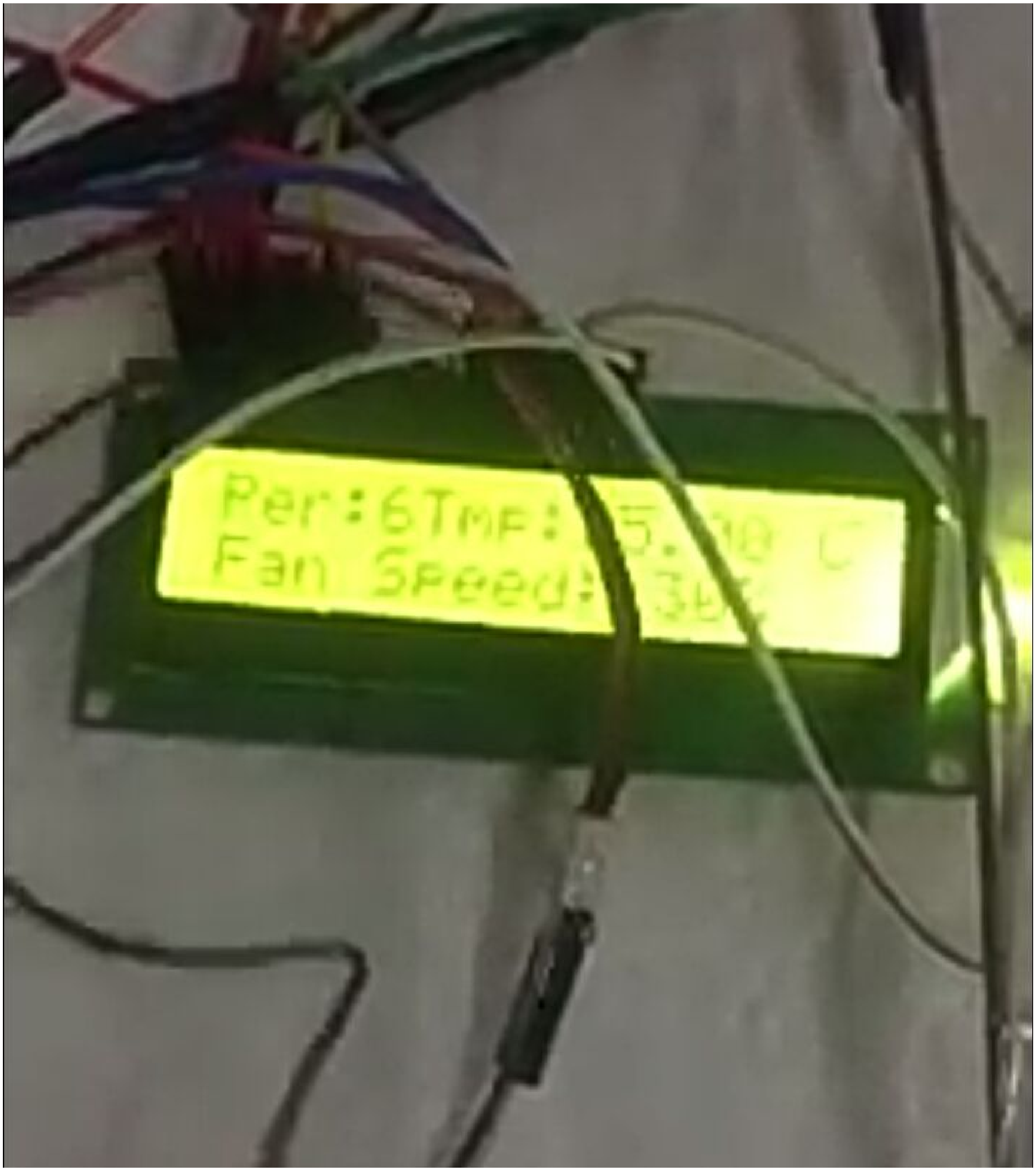

In
Is Off


Per:4Tmp:
Fan OFF

Now, as soon as IR sensor present at the entry of the room senses a person entering in the room it will increment the person and turn on the LED and fan with its minimum intensity .



Now, as number of person in the room increases speed of fan and intensity of light increases simultaneously which is controlled by DHT22 temperature sensor and LDR respectively.



This process of automation can be further used with different devices and contribute to the betterment of the people and on the other hand the environment.

Timeline of the Project

No.	DATE	WORK DUE	COMPLETED
1.	30/01/2019	• Forming the groups	✓
2.	07/02/2019	• Deciding the topic of the project, its specification • The technical flow of the project • What components will be needed	✓ ✓ ✓
3.	04/03/2019	• Deciding the specification of the components and get them. • Develop the algorithm of the project • Come up with the circuit diagram of the project • Finding ways for how the concepts of Embedded systems will make the project more efficient.	✓ ✓ ✓
4.	30/03/2019	• Completing major parts of the project. ➔ Programming ➔ Hardware Implementation	✓
5.	06/04/2019	• Last Finish up, Testing and Debugging	✓
6.	10/04/2019	• Public Display of the Project	-

12. REFERENCES

[1] Adeloye, M. (2017, 09 28). Construction Of A Temperature Controlled Fan Using A
Microprocessor. American Journal of Engineering Research (AJER), 6(9).

[2] Mashud, M. A. A., Yasmin, D., Razzaque, M. A., & Uddin, M. H. Automatic Room
Temperature Controlled Fan Speed Controller using PT-100. International Journal of
Scientific & Engineering Research, 6.

APPENDIX A

1. ***DHT11:***

DHT11 - Humidity and Temperature Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

Features

- Full range temperature compensated
- Relative humidity and temperature measurement
- Calibrated digital signal
- Outstanding long-term stability
- Extra components not needed
- Long transmission distance
- Low power consumption
- 4 pins packaged and fully interchangeable

Details

This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process.
The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.

Specifications

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃，1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

2. *L293D Motor Driver IC:*

L293D Motor Driver

Introduction:

The L293D motor driver is available for providing User with ease and user friendly interfacing for embedded application. L293D motor driver is mounted on a good quality, single sided non-PTH PCB. The pins of L293D motor driver IC are connected to connectors for easy access to the driver IC's pin functions. The L293D is a Dual Full Bridge driver that can drive up to 1Amp per bridge with supply voltage up to 24V. It can drive two DC motors, relays, solenoids, etc. The device is TTL compatible. Two H bridges of L293D can be connected in parallel to increase its current capacity to 2 Amp.

Features:

- Easily compatible with any of the system
- Easy interfacing through FRC (Flat Ribbon Cable)
- External Power supply pin for Motors supported
- Onboard PWM (Pulse Width Modulation) selection switch
- 2pin Terminal Block (Phoenix Connectors) for easy Motors Connection
- Onboard H-Bridge base Motor Driver IC (L293D)

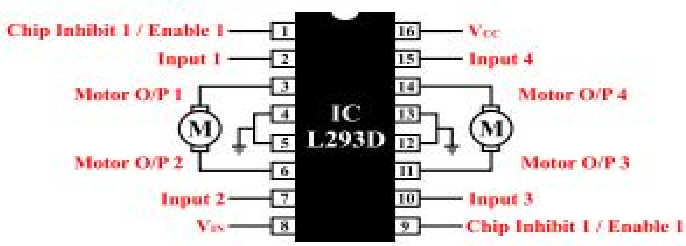
Technical Specification:

- Power Supply : Over FRC connector 5V DC
  External Power 9V to 24V DC
- Dimensional Size : 44mm x 37mm x 14mm (l x b x h)
- Temperature Range : 0°C to +70 °C

L293D IC:

The driver IC L293D is quad push-pull drivers capable of delivering output currents to 1A per channel respectively. Each channel is controlled by a TTL-compatible logic input and each pair of drivers (a full bridge) is equipped with an inhibit input available at pin 1 and pin 9. The motor will run only when chip inhibit is at high logic i.e. chip inhibit is enabled.

The connection diagram is shown below:



3. DC Motor

1. Introduction

Motor 1 Specification – 12V DC Motor

Features

- 12V – 200RPM –
- 3.6KG•CM torque DC gearhead motor
- 30:1 Gear Ratio
- 2mm rear encoder shaft
- Good compromise between speed and torque for small robotic designs.

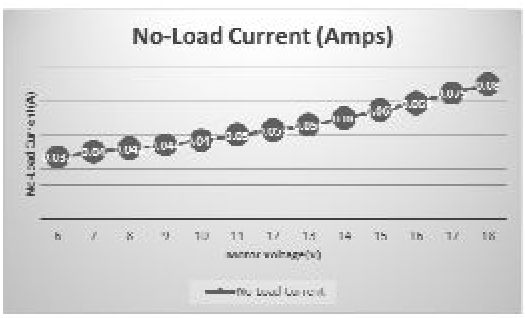
1.1 Description

Motor1 is a 12V DC geared motor with a .25" motor output shaft and a 2mm rear encoder shaft. The 2mm shaft works with our ENC300 quadrature encoder to allow the motor to be used in position control applications. Motor controllers that are rated for 12V@2A are ideal for controlling this motor. However, motor controllers with lower current ratings can also be used if they have over-current and over-temperature protection.

2. Motor1 Specifications

2.1 DC Gearhead Motor Characteristics

Characteristic	Value	Unit
Operating voltage	4.5-18	V
Startup torque (kilogram-force centimeter)	3.6	KG•CM
Startup torque (inch-pound)	3.1	Inch•lbf
Gear ratio	30:1	
No-Load Current (12V)	0.053	A
Stall Current	1.5	A
No-Load Speed (12V)	200	RPM



4. LCD

1. Features

1. 5x8 dots with cursor
2. 16characters *2lines display
3. 4-bit or 8-bit MPU interfaces
4. Built-in controller (ST7066 or equivalent)
5. Display Mode & Backlight Variations
6. ROHS Compliant

LCD type	☐TN			
	☐FSTN	☑FSTN Negative		
	☐STN Yellow Green	☐STN Gray	☐STN Blue Negative	
View direction	☑6 O'clock	☐12 O'clock		
Rear Polarizer	☐Reflective	☐Transflective	☑Transmissive	
Backlight Type	☑LED	☐EL	☐Internal Power	☑3.3V Input
		☐CCFL	☑External Power	☐5.0V Input
Backlight Color	☑White	☐ Blue	☐ Amber	☐Yellow-Green
Temperature Range	☑Normal	☐Wide	☐Super Wide	
DC to DC circuit	☐Build-in	☑Not Build-in		
Touch screen	☐With	☑Without		
Font type	☑English-Japanese	☐English-Europen	☐English-Russian	☐other

2. MECHANICAL SPECIFICATIONS

Module size	80.0mm(L)*36.0mm(W)* Max13.5(H)mm
Viewing area	64.5mm(L)*16.4mm(W)
Character size	3.00mm(L)*5.23mm(W)
Character pitch	3.51mm(L)*5.75mm(W)
Weight	Approx.

5. 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	-	400	-	Ω
	10 lux	-	9	-	kΩ
Dark resistance	-	1.0	-	-	MΩ
Dark capacitance	-	-	3.5	-	pF
Rise time 1	1000 lux	-	2.8	-	ms
	10 lux	-	18	-	ms
Fall time 2	1000 lux	-	48	-	ms
	10 lux	-	120	-	ms

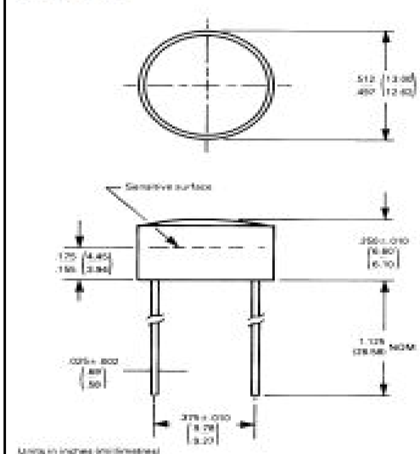
1. Dark to 110% R_L
2. To 10 × R_L

R_L = photocell resistance under given illumination.

Features

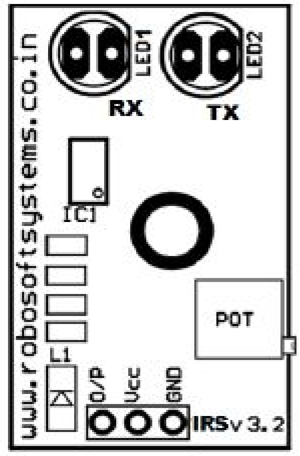
- Wide spectral response
- Low cost
- Wide ambient temperature range.

Dimensions



5. IR sensors

IR Sensor - Single



General Description

The IR Sensor-Single is a general purpose proximity sensor. Here we use it for collision detection. The module consist of a IR emitter and IR receiver pair. The high precision IR receiver always detects a IR signal.

The module consists of 358 comparator IC. The output of sensor is high whenever it IR frequency and low otherwise. The on-board LED indicator helps user to check status of the sensor without using any additional hardware.

The power consumption of this module is low. It gives a digital output.

Pin Configuration

The figure to the right is a top view of the IR Sensor module. The following table gives its pin description.

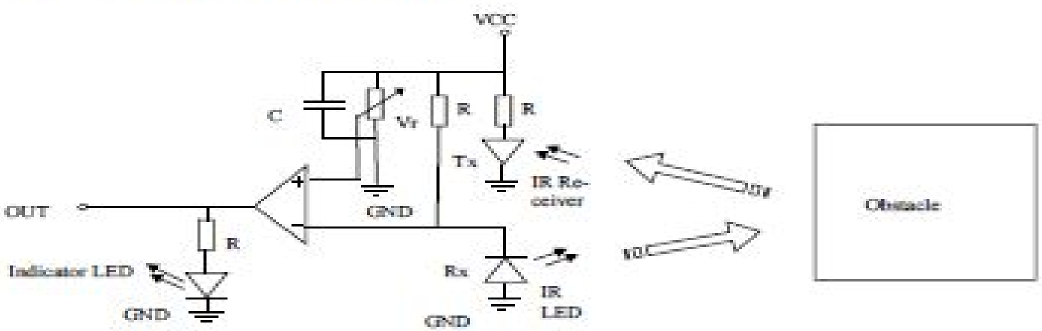
Pin No.	Connection	Description
1	Output	Digital Output (High or Low)
2	VCC	Connected to circuit supply
3	Ground	Connected to circuit ground

Application Ideas

- Obstacle detection
- Shaft encoder
- Fixed frequency detection

IR Sensor - Single

Functional Block Diagram /Schematic Diagram



Overview of Schematic

The sensitivity of the IR Sensor is tuned using the potentiometer. The potentiometer is tuneable in both the directions. Initially tune the potentiometer in clockwise direction such that the Indicator LED starts glowing. Once that is achieved, turn the potentiometer just enough in anti-clockwise direction to turn off the Indicator LED. At this point the sensitivity of the receiver is maximum. Thus, its sensing distance is maximum at this point. If the sensing distance (i.e., Sensitivity) of the receiver is needed to be reduced, then one can tune the potentiometer in the anti-clockwise direction from this point.

Further, if the orientation of both Tx and Rx LED's is parallel to each other, such that both are facing outwards, then their sensitivity is maximum. If they are moved away from each other, such that they are inclined to each other at their soldered end, then their sensitivity reduces.

Tuned sensitivity of the sensors is limited to the surroundings. Once tuned for a particular surrounding, they will work perfectly until the IR illumination conditions of that region nearly constant. For example, if the potentiometer is tuned inside room/building for maximum sensitivity and then taken out in open sunlight, its will require retuning, since sun's rays also contain Infrared (IR) frequencies, thus acting as a IR source (transmitter). This will disturb the receiver's sensing capacity. Hence it needs to be retuned to work perfectly in the new surroundings.

The output of IR receiver goes low when it receives IR signal. Hence the output pin is normally low because, though the IR LED is continuously transmitting, due to no obstacle, nothing is reflected back to the IR receiver. The indication LED is off. When an obstacle is encountered, the output of IR receiver goes low, IR signal is reflected from the obstacle surface. This drives the output of the comparator low. This output is connected to the cathode of the LED, which then turns ON.

APPENDIX B

C vs Assembly

Portability: Portability is the most important factor in language selection. If the source code is not portable then we have to pay a lot of time.

Maintainability: Language should be easy to understand otherwise it will take a huge amount time for small changes.

Availability: Compiler and IDE should be easily available in the market and most important thing is that it should be cheap.

Efficiency: Language has a good efficiency and bug rate should be less.

Development time: Take less amount of time to develop the project

There are following points which describe the difference between
<u>C vs Assembly.</u>

- The c language provides portability and does not depend on the specific platform. It is the biggest advantage of C and this property make people helpless to use C.The code which was written in c could be easily reused on a different platform, beside it Assembly does not provide the portability and source code specific to a processor because assembly instruction depends on the processor architecture.

- Software which has written in assembly perform well as compared to C. Now's days C compiler generate more optimize assembly code. C compilers generally do better optimizations than you can sustain for hours and days. In general, we cannot say compiler-generated assembly code always better, because when human writes a software in assembly then he thinks a lot of scenarios specific to the requirement and processor which did not handle by the compiler.
- In case of micro-controller IDE generate an assembly code (startup code) to initialize the stack, heap, and NVIC . And further, it is hard to maintain Assembly code. So when stack and heap have initialized we used to call a C function to initialize different
system clocks.

- I believe API which has written in assembly would be fast. That is the biggest reason to write memset () and memcpy () routine in Assembly.

- I believed executable generated by assembly language have smaller size compare to c language but not always true.

C vs Verilog

Therefore, I have a few questions

1. A firmware really can be written either in HDL or in a software programming language, or its just another way to perform the same mission?I would love to real-world examples. What constraints resulting from each option?
2. I know that a common use of firmware over software is in hardware accelerators (such as GPUs, network adapters, SSL accelerators, etc). As I understand it, this acceleration is not always necessary, but only recommended (for example, in the case of SSL and acceleration of complex algorithms). Can one choose between firmware and software in all cases? If
not, we would be happy to cases in which firmware is clearly and unequivocally appropriate.
3. I might have read that the firmware mostly burned on ROM or flash. How it is represented in there? In bits, like software? If so, what's the profound difference? Is it the availability of adapted circuits in the case of firmware?

APPENDIX C

Troubleshooting

(1) Trying to operate two Arduino uno at the same time became a bit complex so we simplified the model by using one arduino mega instead of two arduino uno.

(2) The two IR sensors used by us were of two different types , When they sensed any obstacle , one would return logic 0 to the controller while the other would return logic 1 to the microcontroller.

(3) The inside circuit of the Motor Driver Ic was such that when 9 volt battery was connected to it for the fan (Motor), it supplied the power to the Arduino so the fan ended up getting less power supply , and Short - circuit would take place hence garbage value was displayed on the LCD. We solved it by removing the vcc and ground connections from Motor driver IC to the Arduino.

VIDEO :

https://drive.google.com/file/d/1cdiIEnULS444r8Rq5PxaqnV4hV_GS5Uc/view?usp=sharing