TEMPERATURE BASED FAN SPEED CONTROL

OBJECTIVE:

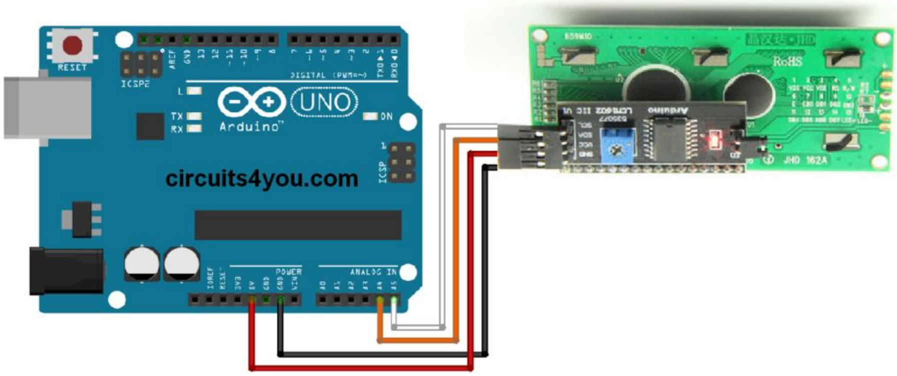
The idea behind the project TEMPERATURE BASED FAN SPEED CONTROL is to control the speed of the fan using microcontroller based on the variation in temperature detected by the temperature sensor.

COMPONENTS USED:

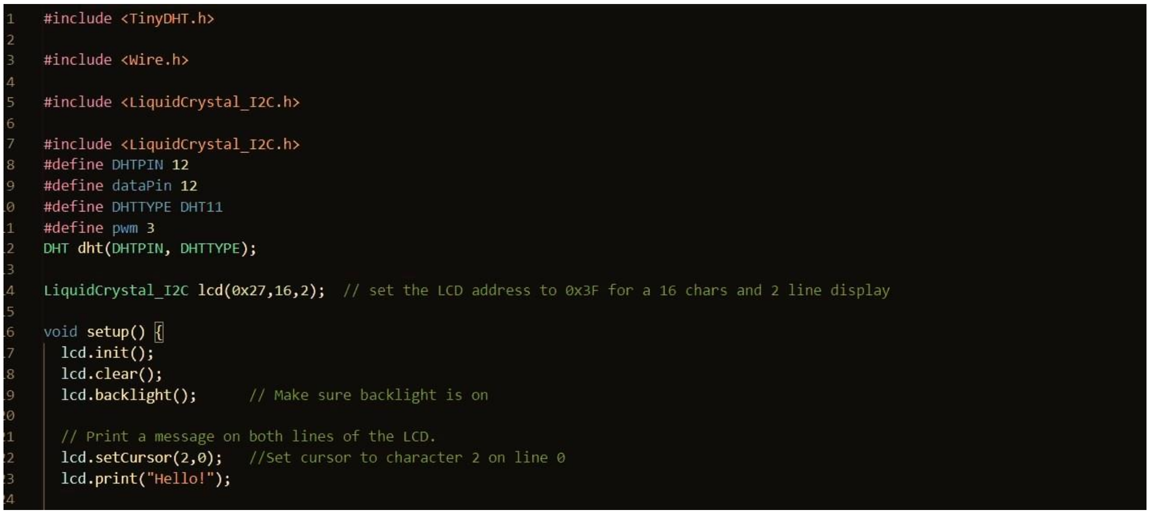
1. Arduino UNO board
2. DHT 11 Temperature sensor
3. 5V DC Fan
4. 2N2222 Transistor
5. I2C microcontroller
6. LCD display
7. Bread Board
8. Jumper Cables
9. 3 blade plastic fan

PROCEDURE:

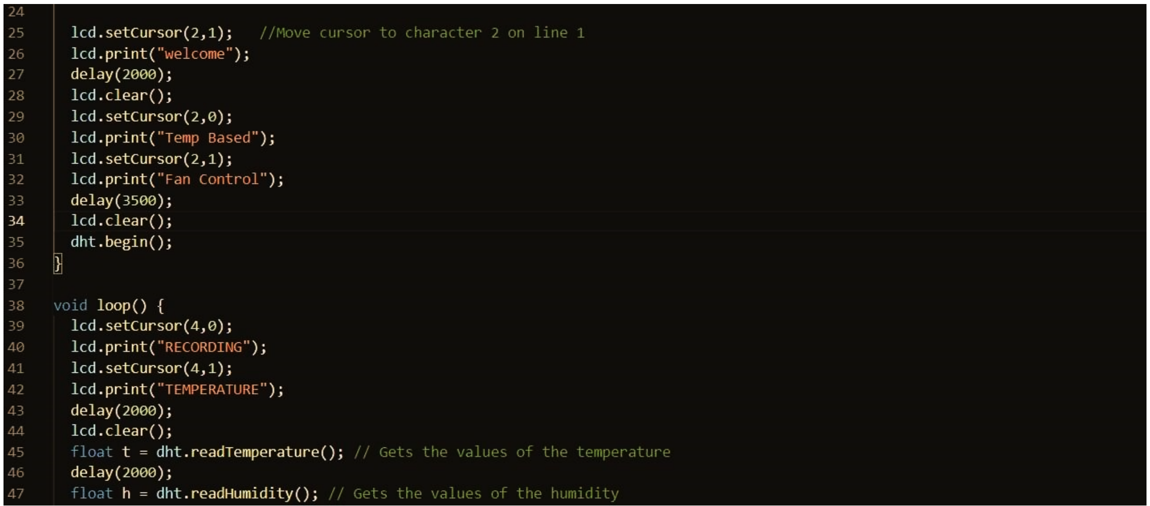
1. First take two jumper cables and connect 5V and gnd pins or arduino to the first two vertical slots on the breadboard respectively. This gives 5V potential to all the slots on first row and 0V to all the slots in the second row of bread board
2. Now connect the DHT11 temperature sensor, by connecting the positive terminal of the sensor to 5V on the breadboard and negative terminal to 0V pin. Now connect the out pin from DHT sensor to 12th pin on breadboard using jumper cable.
3. Solder I2C microcontroller to lcd display with pins in the correct order as shown in the picture
4. Now connect the soldered I2C lcd panel to the arduino board accordingly. Connect VCC pin to 5V , GND pin to 0V and SDA,SCL PINS to A4 and A5 pins on arduino board respectively.



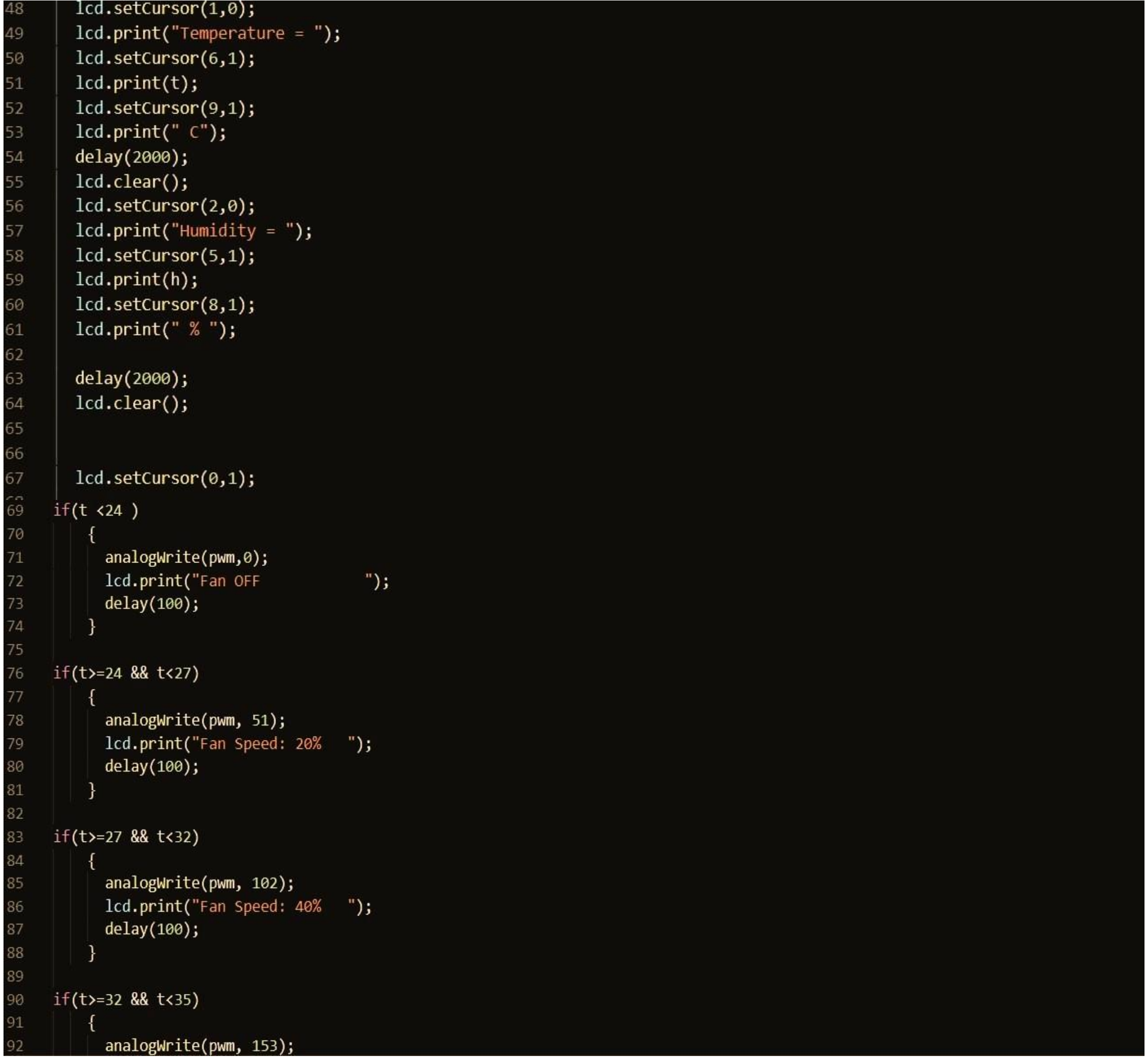
5. Now place 2N2222 transistor on breadboard and connect collector pin on transistor to 5V and base pin to 3rd pin on arduino board and emitter pin to positive terminal of DC fan.
6. After emitter pin is connected to positive terminal of DC fan connect negative pin to the ground or 0V.
7. Below is the schematic diagram of all the connections made and final model after all connections
8. After connecting all components, its time to write the code in the arduino IDE environment. Blow the code for the arduino board. Then connect arduino board to PC where IDE is installed. The IDE detects the board and then we need to select the type of board i.e arduino uno. Then upload the code to the board.
9. After uploading once you need not connect it to your PC. you just need a power connection.



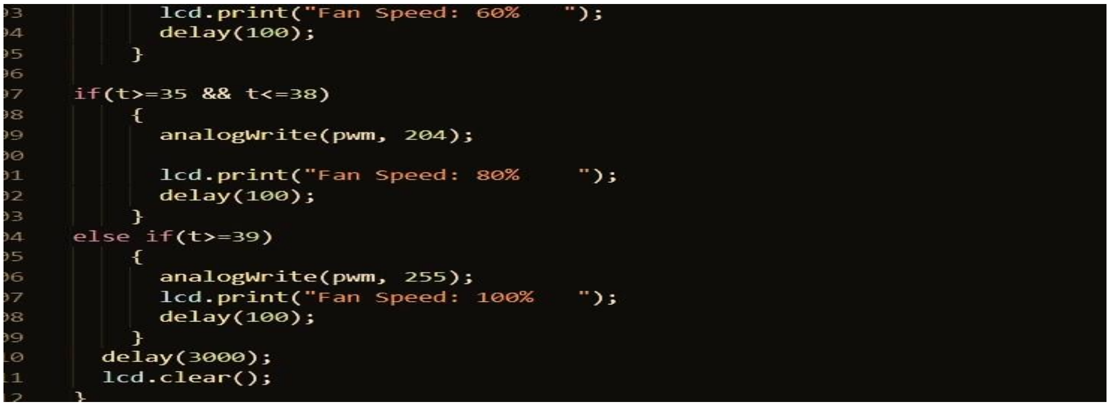
```
#include <TinyDHT.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <LiquidCrystal_I2C.h>
#define DHTPIN 12
#define dataPin 12
#define DHTTYPE DHT11
#define pwm 3
DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x3F for a 16 chars and 2 line display

void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();      // Make sure backlight is on

  // Print a message on both lines of the LCD.
  lcd.setCursor(2,0);   //Set cursor to character 2 on line 0
  lcd.print("Hello!");

```



```

  lcd.setCursor(2,1);   //Move cursor to character 2 on line 1
  lcd.print("welcome");
  delay(2000);
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Temp Based");
  lcd.setCursor(2,1);
  lcd.print("Fan Control");
  delay(3500);
  lcd.clear();
  dht.begin();
}

void loop() {
  lcd.setCursor(4,0);
  lcd.print("RECORDING");
  lcd.setCursor(4,1);
  lcd.print("TEMPERATURE");
  delay(2000);
  lcd.clear();
  float t = dht.readTemperature(); // Gets the values of the temperature
  delay(2000);
  float h = dht.readHumidity(); // Gets the values of the humidity
```

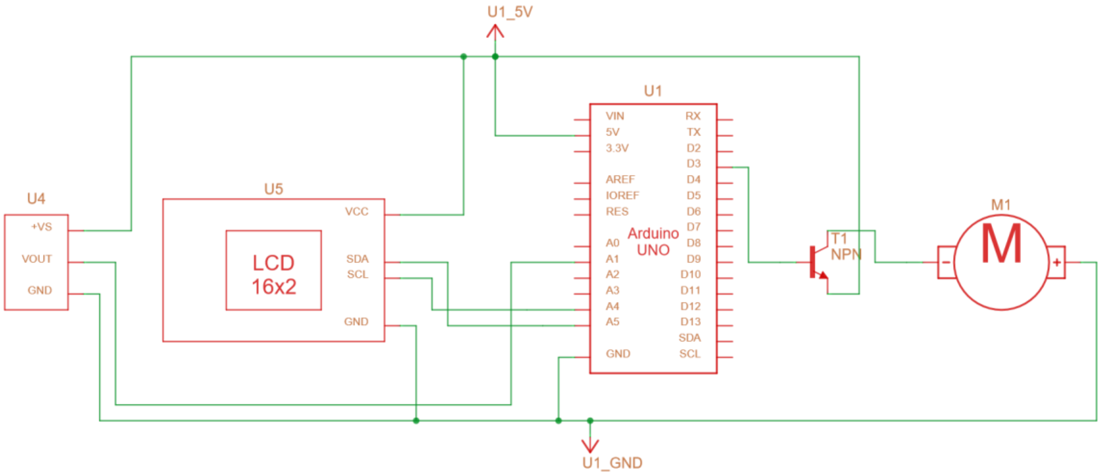


```
  lcd.setCursor(1,0);
  lcd.print("Temperature = ");
  lcd.setCursor(6,1);
  lcd.print(t);
  lcd.setCursor(9,1);
  lcd.print(" C");
  delay(2000);
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Humidity = ");
  lcd.setCursor(5,1);
  lcd.print(h);
  lcd.setCursor(8,1);
  lcd.print(" % ");

  delay(2000);
  lcd.clear();


  lcd.setCursor(0,1);
if(t <24 )
    {
      analogWrite(pwm,0);
      lcd.print("Fan OFF            ");
      delay(100);
    }

if(t>=24 && t<27)
    {
      analogWrite(pwm, 51);
      lcd.print("Fan Speed: 20%   ");
      delay(100);
    }

if(t>=27 && t<32)
    {
      analogWrite(pwm, 102);
      lcd.print("Fan Speed: 40%   ");
      delay(100);
    }

if(t>=32 && t<35)
    {
      analogWrite(pwm, 153);
```

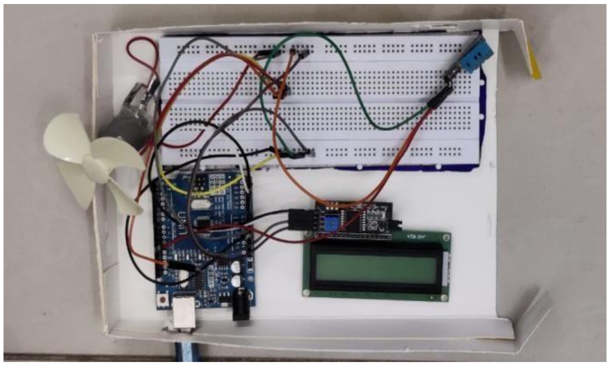


```
        lcd.print("Fan Speed: 60%    ");
        delay(100);
      }

  if(t>=35 && t<=38)
      {
        analogWrite(pwm, 204);

        lcd.print("Fan Speed: 80%     ");
        delay(100);
      }
  else if(t>=39)
      {
        analogWrite(pwm, 255);
        lcd.print("Fan Speed: 100%    ");
        delay(100);
      }
    delay(3000);
    lcd.clear();
  }
```

SCHEMATIC DIAGRAM:



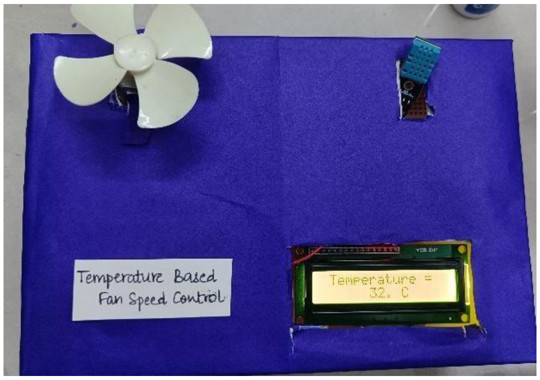
PROTOTYPE:



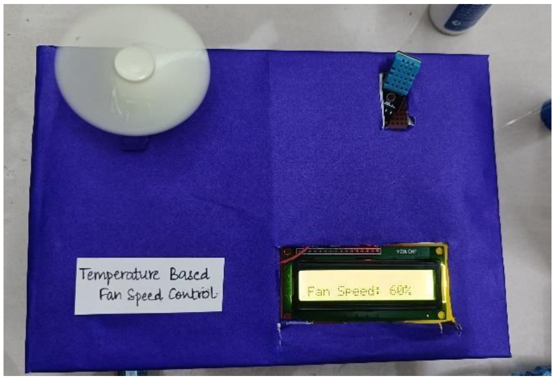
INSIDE VIEW
OF MODEL

RECORDING
TEMPERATURE

TEMPERATURE



FAN SPEED



ADVANTAGES:

1. **NO CONTROL ADJUSTMENTS:**

With automated fan control, you no longer need to disrupt your workflow to adjust the speed and direction of your fans. Automating the fan speed saves time and maintains a consistently comfortable environment.

Control fan speeds with a fully integrated LCD touchscreen display panel. This display panel makes repairs quicker and easier with error codes and diagnostic capabilities. With this state-of-the-art technology and a remote, you have the ability to control the fan speed and direction of up to 30 fans. Control on/off, fan speeds, forward/reverse and memory storage of the last used settings all with a touch of a button.

2. **OPTIMIZED FAN PERFORMANCE:**

Identifying the best speed and direction to operate your fans can be difficult, especially during the spring and fall seasons when the temperatures frequently fluctuate. The automation software calculates all of this for you and perfectly optimizes your fans’ performance for the current climate.

We offer optional control systems for the Macro-Air Controller 30. The comfort control thermostat automatically maintains the room temperature. It gauges the heat index and measures the temperature and humidity between the ceiling and floor. Finally, it calculates each fan’s ideal fan speed and direction.

3. **SAVING ON HEATING AND COOLING:**

If you couple your fans with air conditioning, the cooling effect created by the fans will enable you to set your thermostat higher while maintaining the same level of comfort. This combined with other building enhancements will cut down your costs. In the winter, the fans run in reverse to mix the heated air, eliminating hot and cold spots and decreasing your heating costs. Automation running on a power supply of 110-240v maximizes energy efficiency by lowering energy costs across your climate control solutions.

CONCLUSION:

This temperature-based fan speed control model has a great future scope. With the help of this we can monitor more parameters like humidity, light and at the same time control them. We can send this data to a remote location using mobile or internet. When the temperature exceeds the limit, we can alert the areas. This model will also be widely used in Electronic Industries in the future.

GROUP: **DUNE**

1. Bommiditha Jyothsnavi - 2021MEB1277
2. Guguloth Krishna Prasad - 2021MEB1284
3. Nakka Nishika – 2021MEB1301
4. Murki Rishi - 2021MEB1300
5. Jakkam Parnika - 2021MEB1289
6. Mitta Vasavi Srija - 2021MEB1297