NATIONAL INSTITUTE OF TECHNOLOGY AGARTALA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Mini-Project Titled

“*Temperature Controlled fan with voice Control*”

is submitted to the

Microprocessors & Microcontrollers Laboratory.

Academic Session: 2021-22

Submitted by:

a) **Arkajyoti Bhattacharya, 19UEC021**
b) **Sohel Kabir Rana, 19UEC022**
c) **Debarun Roy, 19UEC025**
d) **Debangshu De, 19UEC001**
e) **Souradeep Dey, 19UEC004**

Under the Supervision of

Dr. Apangshu Das

Assistant Professor, ECE, NIT Agartala

Group No.: 5

Name of the Project: *Temperature Controlled fan with Voice Control*

Information of the group members:

Sl No.	Name	Enrolment No	Email	Contact
1	Arkajyoti Bhattacharya	19UEC021	arkajyotibhattacharya026@gmail.com	7005476681
2	Sohel Kabir Rana	19UEC022	skr03ar@gmail.com	6009166436
3	Debarun Roy	19UEC025	debarunroy2000@gmail.com	6009256785
4	Debangshu De	19UEC001	dedebangshu2000@gmail.com	8256969141
5	Souradeep Dey	19UEC004	deys9603@gmail.com	6009439628

Summary of our project:

As we all know, today's technology necessitates the use of automatic systems and automation. Day by day, we're getting closer to automation. The goal of this project is to design, build, and debug a Microcontroller-based system that measures temperature and has a Voice Control feature for controlling and manoeuvring the speed of a DC Motor-based Fan. The Voice Control increases range of speeds at which the Fan can operate in, regardless of the Temperature sampled. As a result, the physical effort required is reduced, making the process considerably easier.

Introduction:

A cooling fan is one of the most fundamental needs of humans in hot weather. However, at the moment, the fan's speed can only be adjusted manually using a manual switch known as a fan regulator. Intelligent systems are being introduced every day as technology advances. Everything is becoming more complex and understandable. The need for cuttingedge technologies and sophisticated electrical systems is increasing. Microcontrollers are crucial in the creation of smart systems because they provide the system with a brain. Microcontrollers have become the brains behind many of the new technologies that are being released on a regular basis. A microcontroller is a single-chip microprocessor that is used to automate & control machines and processes. Microcontrollers are being used in a wide range of applications to perform more accurate automated activities. Microcontrollers are used in almost every contemporary equipment, including air conditioners, power tools, toys, and office machinery. Microcontroller essentially consists of Central Processing Unit (CPU), timers and counters, interrupts, memory, input/output ports, Analog to Digital Converters (ADC) on a single chip. With this single chip integrated circuit design of the microcontroller the size of control board is reduced and power consumption is low. This Project aims at the Design, Construction and Debugging of Microcontroller based system which samples temperature and has Voice Control Feature to Control and manoeuvre the speed of a DC Motor based Fan. The Voice Control increases range of speed at which the Fan can operate in, regardless of the Temperature sampled. The temperature-based fan speed control system may be implemented utilising an Arduino board and an electrical circuit. Because the Arduino board is now quite advanced among all electronic circuits, we used it to regulate the fan speed.

Background:

The following are the components and software that were utilised in this project:

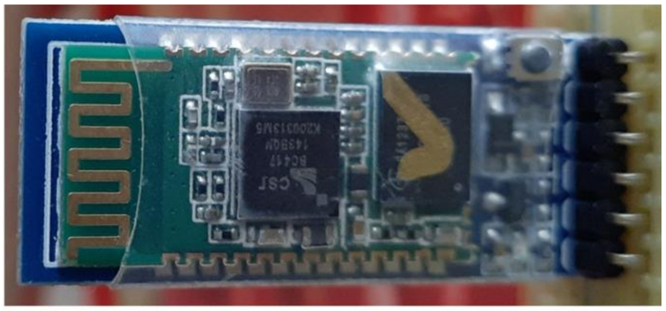
Arduino UNO:
Arduino UNO is based on an ATmega328P microcontroller. There are 20 Input/Output pins present on the Arduino UNO board. These 20 pins include 6 PWM pins, 6 analog pins, and 8 digital I/O pins.

The PWM pins are Pulse Width Modulation capable pins.
The crystal oscillator present in Arduino UNO comes with a frequency of 16MHz.
It also has a Arduino integrated Wi-Fi module. Such Arduino UNO board is based on the Integrated Wi-Fi ESP8266 Module and ATmega328P microcontroller.
The input voltage of the UNO board varies from 7V to 20V.

Arduino UNO automatically draws power from the external power supply. It can also draw power from the USB.
The Memory Structure of Arduino UNO:
Flash Memory (32 kB) + SRAM(2 kB) + EEPROM(1kB)+ Preinstalled Flash(0.5 kB).

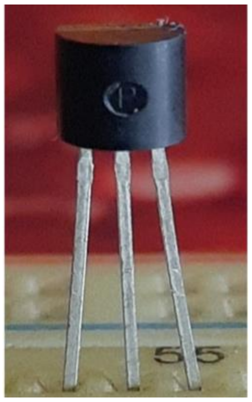
HC-06 Bluetooth Module:

HC-06 is a Bluetooth module designed for establishing short range wireless data communication between two microcontrollers or systems. The module works on Bluetooth 2.0 communication protocol and it can only act as a slave device. This is cheapest method for wireless data transmission and more flexible compared to other methods and it even can transmit files at speed up to 2.1Mb/s.



HC-06 uses frequency hopping spread spectrum technique (FHSS) to avoid interference with other devices and to have full duplex transmission. The device works on the frequency range from 2.402 GHz to 2.480GHz. The communication with this HC-06 module is done through UART interface. The data is sent to the module or received from the module though this interface. So we can connect the module to any microcontroller or directly to PC which has RS232 port (UART interface).

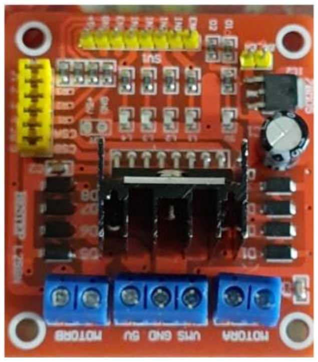
TMP 36 Temperature Sensor:



TMP 36 is an Analog Temperature Sensor.
These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetallic strips (like in some home thermometers or stoves), nor do they use thermistors (temperature sensitive resistors). Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the *Vbe* - of a transistor.) By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature.

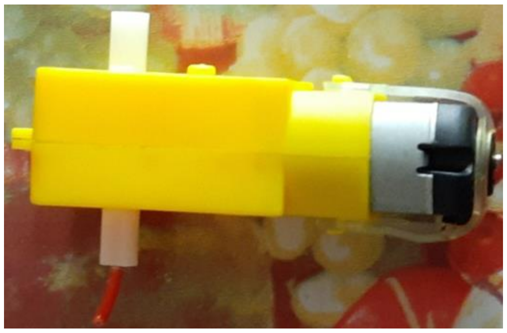
Size: TO-92 package (about 0.2" x 0.2" x 0.2") with three leads.
Temperature range: -40°C to 150°C / -40°F to 302°F.
Output range: 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C.Power supply: 2.7V to 5.5V only, 0.05 mA current draw.

<u>L298N Motor Driver</u>:



The L298N is a dual-channel H-Bridge motor driver capable of driving a 2x DC motors, making it ideal for building two-wheel robots. An H-Bridge circuit contains four switches with the motor at the centre forming an H, closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This leads to a change in the direction of the motor. From 'Vs' pin the H-Bridge gets its power for driving the motors which can be 5 to 35V. '*Vss'* is used for driving the logic circuitry which can be 5 to 7V. And they both sink to a common ground named 'GND'.
The L298N motor driver's output channels for the motor A and B are broken out to the edge of the module with two 3.5mm-pitch screw terminals.The IN1 and IN2 pins control the direction of the motor A while IN3 and IN4 control the direction of the motor B.ENA and ENB are used to turn the motors ON, OFF and control its speed. The module usually comes with a jumper on these pins. When this jumper is in place, the motor is enabled and spins at maximum speed.

<u>100 RPM Dual Shaft BO Motor</u>:

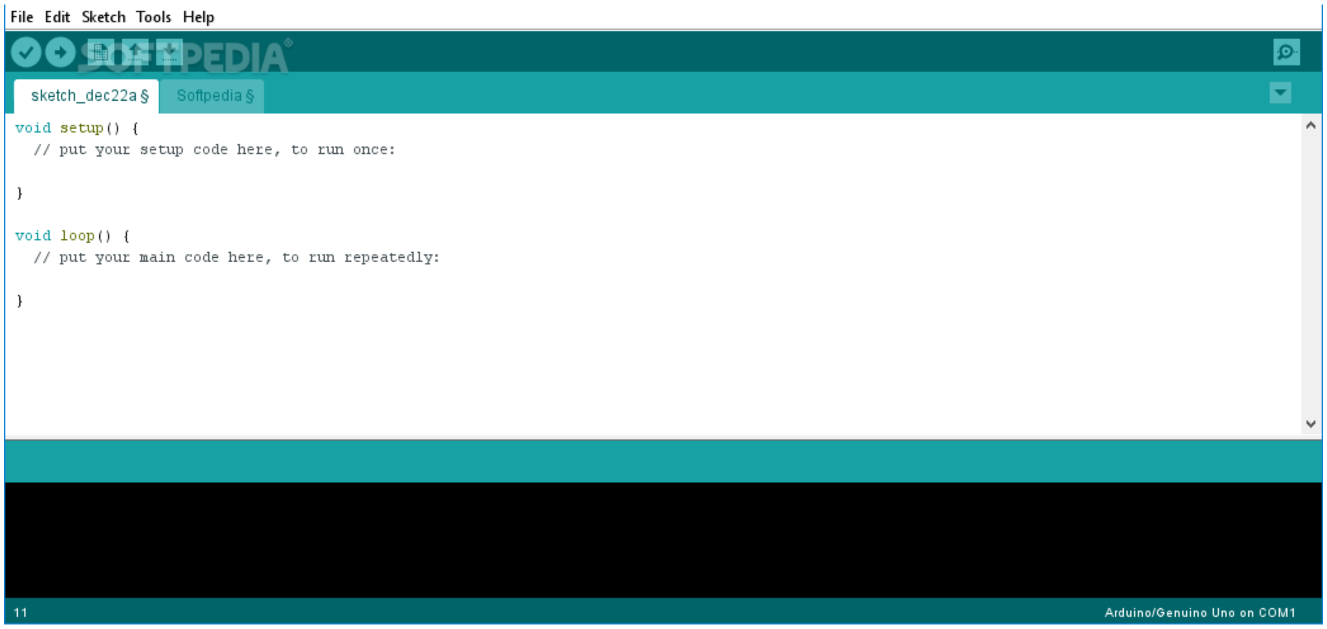


The 100 RPM Dual Shaft BO Motor - Straight motor gives good torque and rpm at lower operating voltages, which is the biggest advantage of these motors. Small shaft with matching wheels gives an optimized design for your application or robot. Mounting holes on the body & light weight makes it suitable for in-circuit placement. This motor can be used with 69mm Diameter Wheel for Plastic Gear Motors. It is an alternative to our metal gear DC motors. It comes with an operating voltage of 3-12V and is perfect for building small and medium robots.

Shaft length: 7 mm
Motor Design: Straight Dual Shaft
Shaft Diameter: 5.5 mm
Size: 55 x 48 x 23 mm.
Operating Voltage: 3 to 12V.
Current (without loading): 40-180mA.
RPM: 100 rpm.
Output Torque: 0.35 kg cm.

<u>Arduino IDE 1.8.16</u>:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.
Active development of the Arduino software is hosted by GitHub. See the instructions for building the code. Latest release source code archives are available here. The archives are PGP-signed so they can be verified using this gpg key.

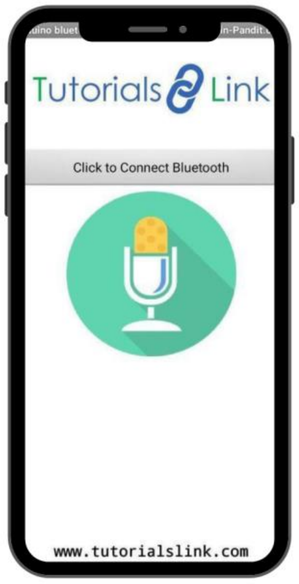


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

11 Arduino/Genuino Uno on COM1

<u>Arduino Bluetooth Voice Controller</u>:

This app (developed by Nitin Pandey) was used as a Master Device for the Slave HC 05 Bluetooth Module. The Voice was recognised and sent to Bluetooth Module and then serially sent to the Arduino.



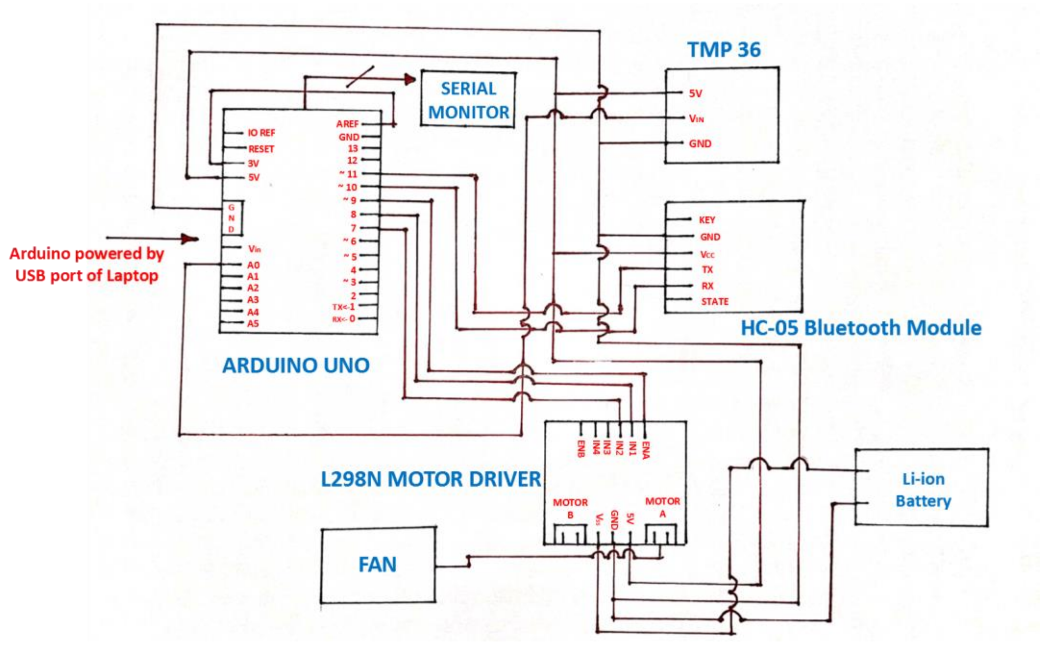
Related/Previous Works:

A project was done previously, with the help of 8051 microprocessor. The goal of the project was to create a temperature-controlled fan with an 8051 microcontroller that switched on and off automatically based on the temperature. The LM35 temperature sensor in the circuit would provide a continuous analogue output according to the temperature it sensed. The ADC received this analogue signal and transformed the analogue values to digital values. The ADC's digital output was the same as the analogue voltage observed. We had to do some calculations in the microcontroller programming to derive the temperature from the observed analogue voltage. The microcontroller would switch on the relay to start the fan if the temperature surpassed 50 degrees Celsius (as specified in the code).

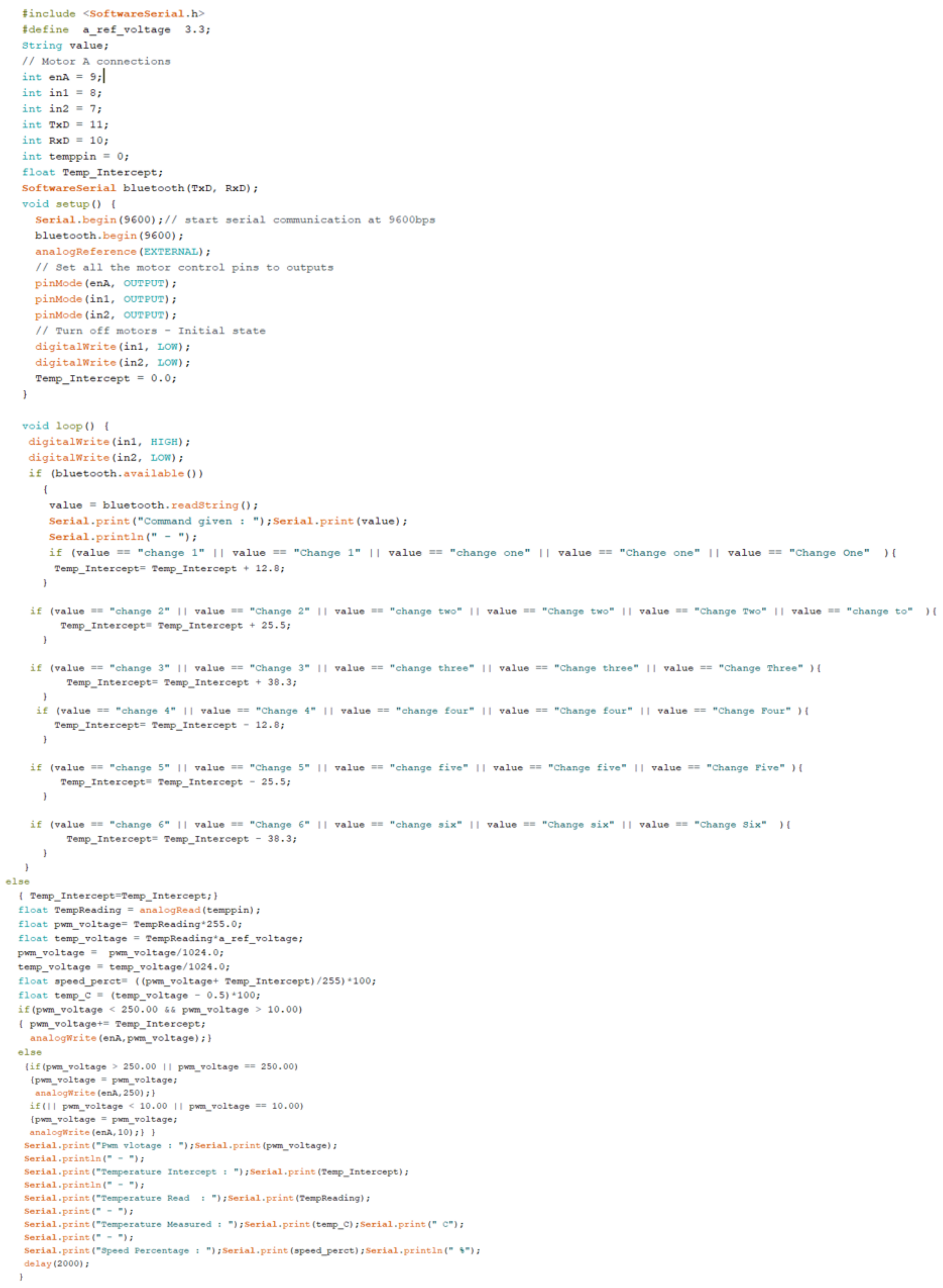
Another project utilising the 89s51 microcontroller was also created. The processing component constituted the microcontroller 89s51, which initially received data from the ADC. Through the amplifier, ADC received data from the temperature sensor. Then, according to the logic of the programme for which the microcontroller had already been programmed, microcontroller 89s51 compared the current temperature to the predetermined temperature. The output port of the 89s51 was used to send the outcome of the aforesaid operation to an LCD display of important data and created pulses according to the logic programme, which was then sent to the driver circuit to get the required output of the ceiling fan.

Description of Proposed work:

Circuit Diagram:



Code:



```
#include <SoftwareSerial.h>
#define  a_ref_voltage  3.3;
String value;
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;
int TxD = 11;
int RxD = 10;
int temppin = 0;
float Temp_Intercept;
SoftwareSerial bluetooth(TxD, RxD);
void setup() {
  Serial.begin(9600);// start serial communication at 9600bps
  bluetooth.begin(9600);
  analogReference(EXTERNAL);
  // Set all the motor control pins to outputs
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  // Turn off motors - Initial state
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  Temp_Intercept = 0.0;
}

void loop() {
 digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
 if (bluetooth.available())
   {
    value = bluetooth.readString();
    Serial.print("Command given : ");Serial.print(value);
    Serial.println(" - ");
    if (value == "change 1" || value == "Change 1" || value == "change one" || value == "Change one" || value == "Change One"  ){
     Temp_Intercept= Temp_Intercept + 12.8;
   }

 if (value == "change 2" || value == "Change 2" || value == "change two" || value == "Change two" || value == "Change Two" || value == "change to"  ){
      Temp_Intercept= Temp_Intercept + 25.5;
   }

 if (value == "change 3" || value == "Change 3" || value == "change three" || value == "Change three" || value == "Change Three" ){
       Temp_Intercept= Temp_Intercept + 38.3;
   }
  if (value == "change 4" || value == "Change 4" || value == "change four" || value == "Change four" || value == "Change Four" ){
     Temp_Intercept= Temp_Intercept - 12.8;
   }

 if (value == "change 5" || value == "Change 5" || value == "change five" || value == "Change five" || value == "Change Five" ){
      Temp_Intercept= Temp_Intercept - 25.5;
   }

 if (value == "change 6" || value == "Change 6" || value == "change six" || value == "Change six" || value == "Change Six"  ){
       Temp_Intercept= Temp_Intercept - 38.3;
   }
 }
else
  { Temp_Intercept=Temp_Intercept;}
  float TempReading = analogRead(temppin);
  float pwm_voltage= TempReading*255.0;
  float temp_voltage = TempReading*a_ref_voltage;
  pwm_voltage =  pwm_voltage/1024.0;
  temp_voltage = temp_voltage/1024.0;
  float speed_perct= ((pwm_voltage+ Temp_Intercept)/255)*100;
  float temp_C = (temp_voltage - 0.5)*100;
  if(pwm_voltage < 250.00 && pwm_voltage > 10.00)
  { pwm_voltage+= Temp_Intercept;
    analogWrite(enA,pwm_voltage);}
  else
   {if(pwm_voltage > 250.00 || pwm_voltage == 250.00)
    {pwm_voltage = pwm_voltage;
     analogWrite(enA,250);}
    if(|| pwm_voltage < 10.00 || pwm_voltage == 10.00)
    {pwm_voltage = pwm_voltage;
    analogWrite(enA,10);} }
   Serial.print("Pwm vlotage : ");Serial.print(pwm_voltage);
   Serial.println(" - ");
   Serial.print("Temperature Intercept : ");Serial.print(Temp_Intercept);
   Serial.println(" - ");
   Serial.print("Temperature Read  : ");Serial.print(TempReading);
   Serial.print(" - ");
   Serial.print("Temperature Measured : ");Serial.print(temp_C);Serial.print(" C");
   Serial.print(" - ");
   Serial.print("Speed Percentage : ");Serial.print(speed_perct);Serial.println(" %");
   delay(2000);
  }
```

Calculation:

The Temperature is sampled by the Sensor and the Analog Voltage is converted to 10 bit Digital Value (by Internal ADC of Arduino).

The PWM of the Arduino has 8 bit Resolution (has 256 values).Hence, the 10 bit Value has to be mapped to the
8 bit Value linearly.

Moreover, The Voice Control Commands can independently control the level of the PWM Voltage provided by the Arduino.

Hence, The PWM Value (written at EN1) is determined the Equation:

$$PWM\_Value = (256/1024)*Temp\_reading + Temp\_Intercept.$$

Temp_Intercept is shifted by Voice Control.

Six Voice Commands are given:

1. *Change* 1 (*increases Speed approx.* 5%).

   $Temp\_Intercept= Temp\_Intercept + 12.8;$

2. *Change* 2 (*increases Speed approx.* 10%).

   $Temp\_Intercept= Temp\_Intercept + 25.5;$

3. *Change* 3 (*increases Speed approx.* 15%)

   $Temp\_Intercept= Temp\_Intercept + 38.3;$

4. *Change* 4 (*decreases Speed approx.* 5%).

   $Temp\_Intercept= Temp\_Intercept -12.8;$

5. *Change* 5 (*decreases Speed approx.* 10%).

   $Temp\_Intercept= Temp\_Intercept -25.5;$

6. *Change* 6 (*decreases Speed approx.* 15%).

   $Temp\_Intercept= Temp\_Intercept -36.3;$

TMP 36 can range up to 5 Volts. But, for better precision and practically such high temperatures are not feasible, The Ref Voltage is taken up to 3.3 V from the Arduino.

Temperature is converted by the following Equation:

$$Temp\_Converted = ((3.3/\ 1024)*Temp\_reading - 0.5)*100$$

The 0.5 is taken as offset considering the accuracy of the Temparature Sensor.

The Speed percentage is calculated by the Equation:

$$Speed\_perct = (PWM\_Value/255)*100$$

Considering the operating conditions of the DC Motor (max 12V and 40 mA to 180mA Current)
The maximum allowable PWM Value (written at EN1) was kept at 250 and minimum allowable PWM Value (written at EN1) was kept at 10.

Result:

PWM_VALUE	SPEED %	VOLTAGE(drawn by Motor)	CURRENT(drawn by Motor)
56.78	22.27	5.34 V	64.3 mA
83.57	32.57	6.32 V	72.5 mA
95.58	37.84	7.16 V	80.02 mA
134.47	52.60	8.37 V	85.4 mA
172.42	67.62	8.96 V	88.6 mA
210.62	82.73	9.50 V	90.2 mA
249.53	97.75	10.12 V	91.5 mA

Conclusion:

The project is impeccable and may be used commercially on a big scale in the future, with both features, such as automated temperature adjustment and voice control, being utilised**.** Speech recognition is significantly more efficient to utilize than writing or typing since voice can be captured much faster than writing or typing. The data will be easier to collect if you use your voice instead of your body movement, since verbal input is significantly better. Instead of moving towards the electric fan alone, the device may collect the user's words at a higher rate. Making straight and concise notations is quite beneficial. It could minimize and eliminate constraints to handicap accessibility, and provide a safe environment for the users and nonusers.

References:

1. https://docs.arduino.cc/hardware/uno-rev3

2. https://components101.com/modules/l293n-motor-driver-module

3. https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-

4. https://learn.adafruit.com/tmp36-temperature-sensor