Temperature Controlled Fan

Project report submitted in partial fulfillment of the degree of
Electronics and Communication Engineering

For the course

Design Lab 1

Name of the student: Animesh Jain
Roll No. : 20UEC020

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

Date of Submission: 24/11/2021

Abstract

Humans are increasingly adopting new technology, such as substituting manual processes with automatic and computer-controlled gadgets. This project is a self-contained automated fan speed controller that regulates the speed of an electric fan to meet our needs. Temperature control is necessary at many places around us, including air conditioners, water heaters, snow melters, ovens, heat exchangers, mixers, furnaces, incubators, thermal baths, etc. As a result, this project can be quite useful in learning the fundamentals of how to control the temperature in your home. The temperature-controlled fan will react to temperature variations in that particular area. It has a compact design. This will be done using a temperature sensor, the LM35 IC. It can also be done in a variety of other ways but this way is easy to perform, cheap, and the parts used are easily available in the market. The system is user-friendly because of the liquid crystal display (LCD). On the LCD panel, the measured temperature and fan speed level measurements are presented at the same time. In this project, I'll show you how to use Arduino to make a temperature-controlled fan. We will be able to modify the fan speed in our home or anywhere else based on the temperature of the room and display the temperature and fan speed changes on a 16x2 LCD display with this circuit. To do this, I am going to use an Arduino UNO Board, LM35 sensor, 16x2 LCD, DC fan, some resistors, switch, potentiometer, 1N4007 diode, and a buzzer.

1 Motivation

People are progressively receiving modern innovation, such as substituting manual forms with programmed and computer-controlled gadgets. During hot weather, a cooling fan is one of the most fundamental needs of individuals. However, the fan's speed may be adjusted manually via a fan regulator or dimmer switch. The fan speed may be adjusted by adjusting the dimmer. It may be observed in some locations, such as those where the temperature is high in the morning but rapidly drops at night. Users do not understand the temperature differences. So, to deal with the fan's speed here's a way to adjust it based on the temperature. This concept is especially applicable for zones where temperature changes drastically during day and night time. This extension made by this project will change over the manual fan into programmed fans. The programmed fans will change their speed according to the temperature within the room. This circuit, the temperature-controlled fan using a microcontroller is utilized to control the speed of the fan according to the temperature and show the current temperature and the speed of the fan. The required components are an Arduino UNO Board, LM35 sensor, 16x2 LCD, DC fan, some resistors, switch, potentiometer, 1N4007 diode, and a buzzer. Arduino is the heart of the circuit because it controls all functions. LM35

IC is a temperature sensor whose yield voltage is straightly relative to Celsius (Centigrade) temperature. It is appraised to function over a -55°C to 150°C temperature extend. Temperature sensor LM35 faculties the temperature and changes over it into an electrical (analog) flag, which is connected to the microcontroller, the Arduino UNO Board. The analog value is changed over into digital value. Hence the detected values of the temperature and speed of the fan are shown on the LCD. When the temperature surpasses a particular value the fan begins rotating.

2 Proposed Solution

The temperature-controlled fan circuit can be made using commonly used items in the electronics world. The heart of this circuit is ARDUINO UNO. The circuit is designed to find the temperature of the room or surroundings and send that information to the Arduino board. Then the Arduino Uno board executes the code with the current temperature and then processes it to get the speed of the fan suitable for the surrounding. The outcome obtained from the LM35 IC sensor and the speed of the fan which is progressing will be displayed in a 16X2 LCD DISPLAY. This is a brief overview of the system/circuit. I am using 4 Sub-Parts in this whole circuit. Now explaining the components used in the circuit:

Components Used:

1) Arduino Uno

2) Fan - DC

3) LCD Display 16X2

4) Potentiometer

5) 1N4007 Diode

6) Mosfet

7) Speaker

8) Relay

9) Diode

10) BC547 BJT

11) LM35 IC

12) Switch

13) Resistor

14) DC Source

15) Ground

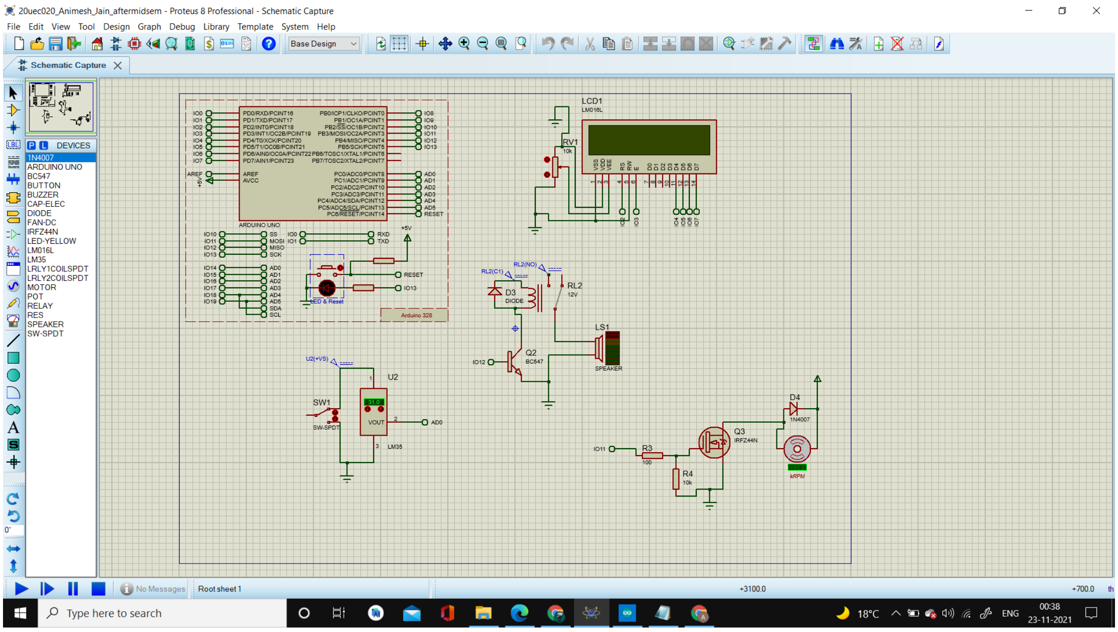

Figure 2.1: Circuit Diagram of the Temperature Controlled Fan using the above mentioned components on Proteus

2.1 1ST SUB PART: THE TEMPERATURE SENSOR CIRCUIT

This small part of the circuit consists of a switch, LM35 IC, Ground, and some connecting wires.

Switch: An electric switch could be a gadget that hinders the electron stream in a circuit. Switches are fundamentally binary gadgets: either completely on or off and light switches have a basic plan. When the switch is turned off, the circuit breaks, and the control stream is interrupted. Circuits comprise a source of power and load.

LM35 IC: LM35 is a precession Integrated circuit Temperature sensor, whose output voltage varies, based on the temperature around it. It is a small and cheap IC that can be used to measure temperature anywhere between -55°C to 150°C. It can easily be interfaced with any Microcontroller that has ADC function or any development platform like Arduino.

Ground: In electrical engineering, ground or earth is a reference point in an electrical circuit from which voltages are measured, a common return path for electric current, or a direct physical connection to the earth. Electrical circuits may be connected to the ground for several reasons.

2.2 2ST SUB PART: THE BUZZER CIRCUIT

This small part of the circuit consists of a relay, BC547 BJT, Speaker, and Ground.

Relay: Electrical relay switches open and closes the circuits by receiving electrical signals from external sources. Some people may relate a “relay” to a race when the team members take turns beating the sticks to complete the race. "Relays" installed in electrical products operate in a similar manner; they receive an electrical signal and send a signal to other devices by turning on and off the switch.

BC547 BJT: BC547 is an NPN transistor which means that when power is applied to the base (control pin) it will flow from the collector to the emitter. NPN transistors are usually used to "change location" on a device, that is, they are placed behind a load in a circuit.

Speaker: The speaker consists of a baffle attached to a cable coil tied in the middle of the magnet. When electricity is transmitted to the coil it moves, pushes or pulls the baffle, which in turn moves the air and emits an audible sound. Speakers have very low resistance and therefore require capacitors to block current DC.

2.3 3ST SUB PART: THE LED 16X2 CIRCUIT

This small part of the circuit consists of a 16X2 LED, Potentiometer, Ground.

16X2 LED: A 16x2 LCD means it can display 16 characters in each row and there are 2 such lines. On this LCD each character is displayed on a matrix of 5x7 pixels. The 16 x 2 smart alphanumeric dot matrix display is capable of displaying 224 different characters and symbols.

Potentiometer: The potentiometer is the electronic part of the passive. Potentiometers work by changing the location of the slide connection to the same resistance. In a potentiometer, the entire input voltage is applied to the entire length of the resistor, and the output voltage is the voltage drop between the fixed and smooth contact.

2.4 4ST SUB PART: THE DC FAN CIRCUIT

This small part of the circuit consists of a DC Fan, 1N4007 Diode, Mosfet, Resistors, Ground.

DC Fan: A DC motor is an engine that converts electrical energy into mechanical energy by drawing a direct current. DC motors make a circular motion on a machine with electromagnetism. DC motors with inductors (electromagnet) inside them form a magnetic field that helps rotate the engine.

1N4007 Diode: 1N4007 is a regenerative diode, specifically designed for circuits that need to convert alternating current into a direct current. It can exceed up to 1 A waves and have a peak inverse voltage (PIV) of 1,000 V.

Mosfet: MOSFET represents a metal-oxide-semiconductor field-effect transistor. Transistors for small electrical devices are used, among other things, alarm clocks, counters, and, perhaps most popularly, computers; they are some of the most basic building blocks of modern times. A few MOSFETs amplify or process analog signals. Most are used in digital electronics.

Resistors: A resistor is an electrical component of two idle terminals that uses electrical resistance as part of a circuit. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, voltage divisions, bias active elements, and cross-transmission lines, among other uses.

2.5 5ST SUB PART: THE ARDUINO CODE

Arduino IDE is special software that works on your system that allows you to write drawings (same program name in Arduino language) for different Arduino boards. Arduino programming language is based on a very simple hardware programming language called processing, similar to C-language. I am attaching a photo of the code written in Arduino IDE.

```
// the starting of code, including the required library for th functionong of the 16X2 LCD
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);   // defining the pins required in the circuit for the working of the LCD
int temperature_Pin = A0;  // defining the output pin of LM35 Temperature IC
int fan = 11;     // defining the pin where the fan is
int buzzer = 12;   // defining the buzzer pin
int temperature;
int temperature_Min = 30;  // defining the temperature when fan needs to start
int temperature_Max = 50;  // Defining the maximum temperature when fan speed reaches at 100%
int fan_Speed; // defining a variable
int fan_LCD;   // defining a variable

void setup() {    // the setup function of the Arduino code, defining the pin whether it is input or output
 pinMode(fan, OUTPUT);
 pinMode(buzzer, OUTPUT);
 pinMode(temperature_Pin, INPUT);
 lcd.begin(16,2); // setting the cursor of 16X2 LED
 lcd.setCursor(5,0);
 lcd.print("WELCOME");   // printing first words on 16X2 LCD display
 lcd.setCursor(4,1);
 lcd.print("20UEC020");  // printing the roll no.
 delay(2000);
 lcd.clear();   // clearing the display
}

void loop() {
  temperature = readTemp();    // the loop function of the Arduino code, reading the temperature from LM35 IC

  if((temperature >= temperature_Min) && (temperature <= temperature_Max)) { // if temperature is higher than minimum temp
     fan_Speed = map(temperature, temperature_Min, temperature_Max, 32, 255); // the actual speed of fan
     fan_LCD = map(temperature, temperature_Min, temperature_Max, 0, 100); // speed of fan to display on LCD
     analogWrite(fan, fan_Speed); // spin the fan at the fanSpeed speed
  }

  if(temperature < temperature_Min) {  // if temp is lower than minimum temp
   fan_Speed = 0;    // fan is not spinning
   fan_LCD = 0;
   digitalWrite(fan, LOW);     // giving the digital value to fan
  }

  if(temperature > temperature_Max) {     // if temp is higher than tempMax
   digitalWrite(fan, HIGH);
   digitalWrite(buzzer, HIGH); // turn on buzzer
  } else {             // else turn of bubber
    digitalWrite(buzzer, LOW);   // giving the digital value to buzzer
  }

  lcd.setCursor(0,0);   // setting the cursor of 16X2 LED
  lcd.print("Temperature:");
  lcd.print(temperature); // display the temperature
  lcd.write(223); // the degree sign is given by this command
  lcd.print("C  ");

  lcd.setCursor(0,1); // move cursor to next line
  lcd.print("Fan Speed:");
  lcd.print(fan_LCD); // display the fan speed
  lcd.print("%   ");
  delay(200);
}

int readTemp() { // get the temperature and convert it to celsius
 temperature = analogRead(temperature_Pin);
 return temperature * 0.48828125;  // returning the required temperature
}
```

Figure 2.2: Code written in Arduino IDE

2.6 Working and other details

The working of the circuit is as follows: The circuit works when the temperature detected by LM35 IC reaches a level and then through arduino it gives command to the DC fan to switch on with a particular speed. As the temperature rises the speed of the fan increases and after a particular level the speed of fan reaches its 100% and then the buzzer beeps to tell that the 100% limit of the fan is reached.
The minimum cut-off temperature I used is 30°C and the maximum cut-off temperature is 50°C so when the temperature goes above 30°C the fan turns on at a particular speed which increments at a regular interval according to the initial temperature, minimum cut-off temperature and maximum cut-off voltage. The buzzer beeps when the fan reaches its top speed which means reaches at the 100% of its speed means the temperature reaches at 50°C. The other thing is the user can see the temperature and the speed of fan continuously on the 16X2 LED Display. I am using MOSFET for controlling the speed of the fan and a relay and BC547 Transistor for controlling Buzzer/Speaker.

3 Results

I am attaching the screenshots of the simulation for better understanding.

1) The screenshot below is when the LM35 temperature sensor senses 25°C and the buzzer and fan both are off. The speed of the fan is 0% and the buzzer is off.

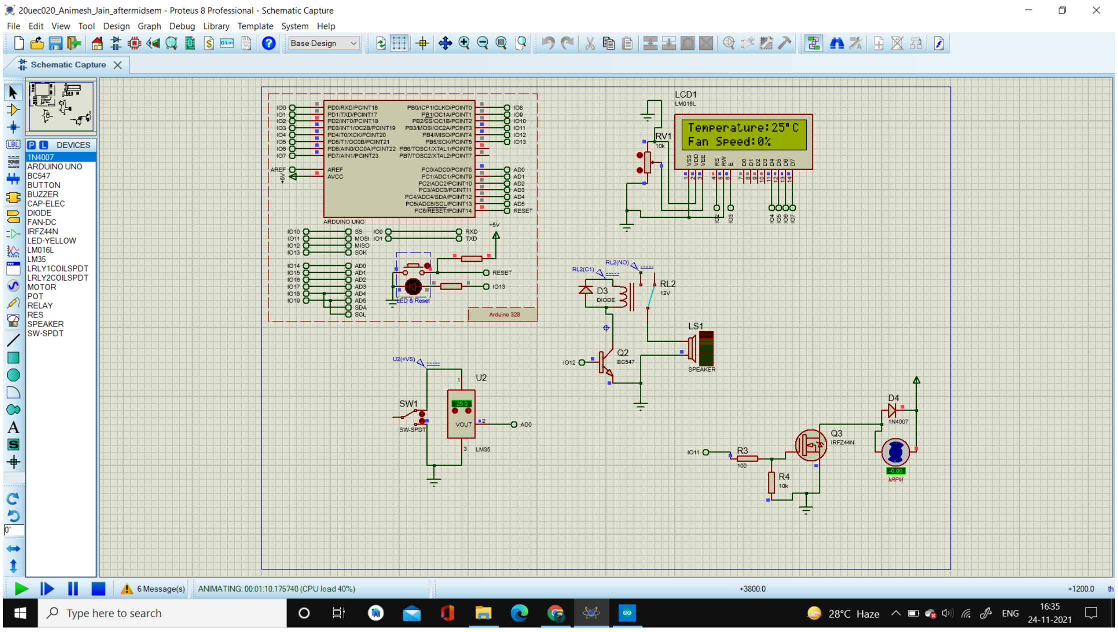

Figure 3.1: Circuit diagram at 25°C

2) Now when the temperature reaches just above the minimum cut-off temperature which is 30°C so the surrounding temperature reaches 31°C and the fan start rotating at a speed of 5% and the buzzer is still off.

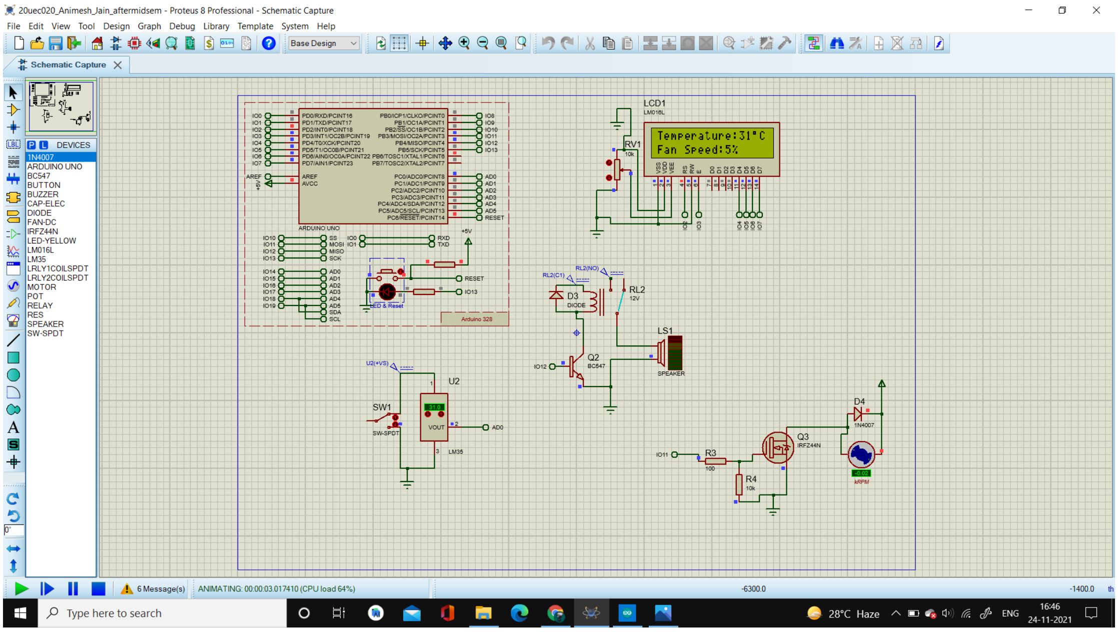

Figure 3.2: Circuit diagram at 31°C

3) Then I increased the temperature to 40°C and the speed of the fan gradually increases and reaches its half potential means at 50% of its top speed and the buzzer is still not ringing. It's off.

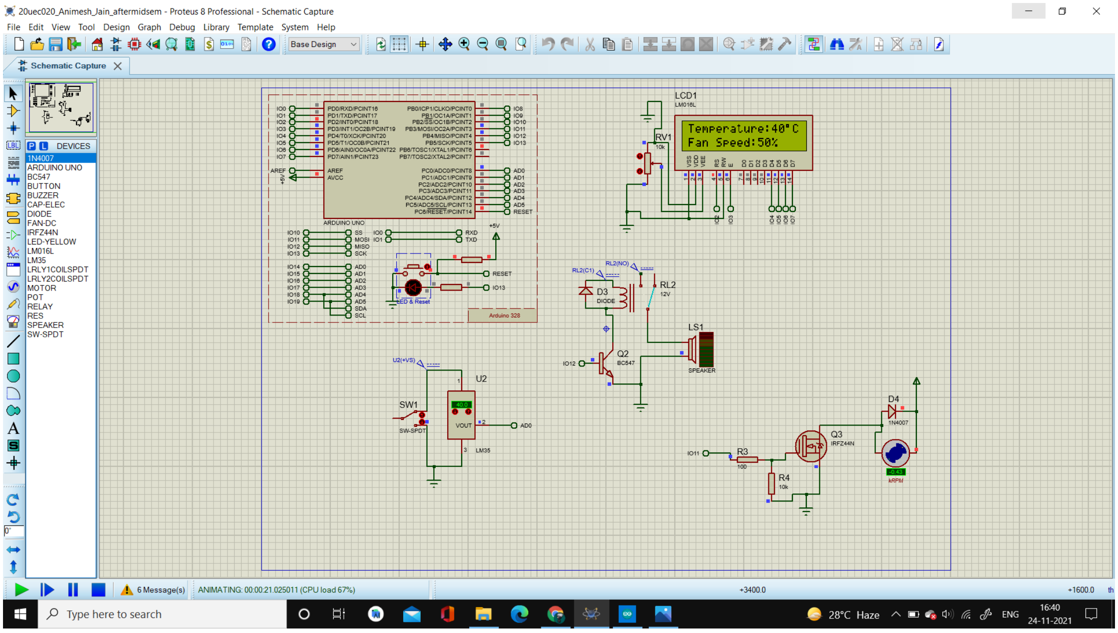

Figure 3.3: Circuit diagram at 40°C

4) Then when the temperature reaches the top level means 50°C then the fan reaches its top speed means 100%. The temperature at that point is equal to the maximum temperature which is 50°C and the speed at 100%. The buzzer is still off.

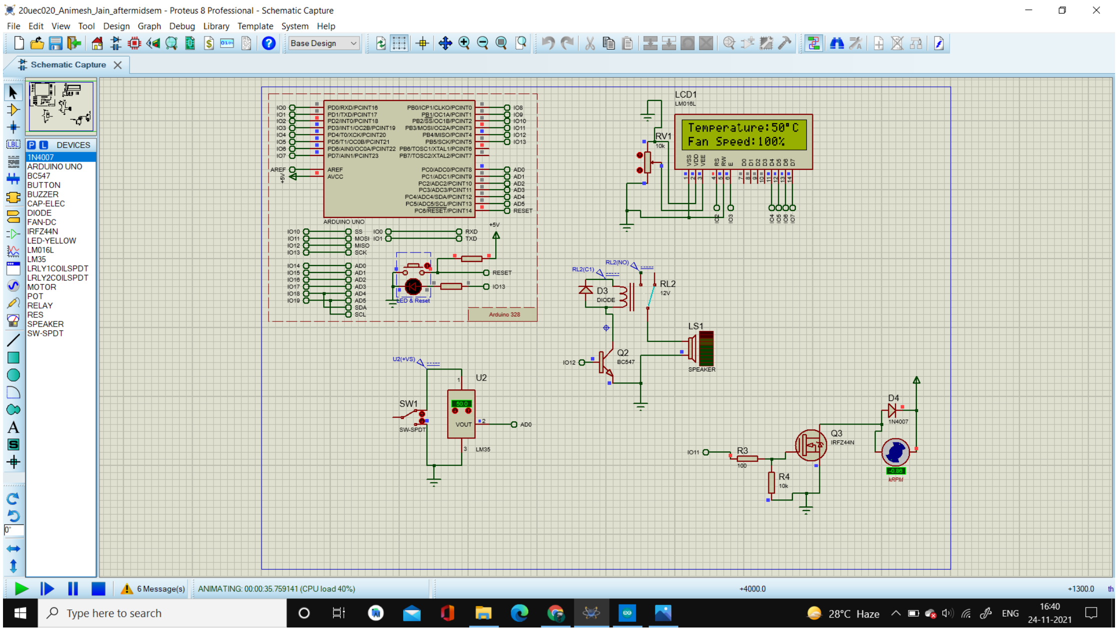

Figure 3.4: Circuit diagram at 50°C

5) As soon as the temperature rises one more °degree the temperature exceeds the maximum temperature and becomes 51°C and the speed of the fan is 100%. As the temperature exceeds the maximum temperature the buzzer will start ringing and the user can know that the temperature has exceeded the maximum temperature and the speed of the fan now can't be increased.

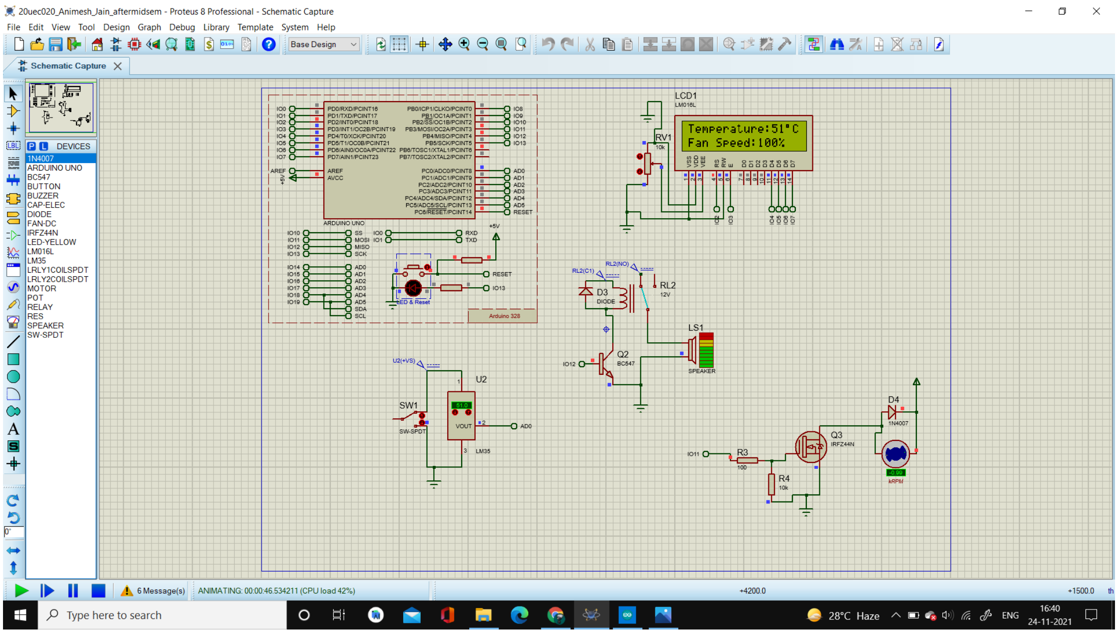

Figure 3.5: Circuit diagram at 51°C

The fan works automatically taking the instructions from the Arduino Uno and it becomes an easy and useful item for day-to-day life.

4 Conclusion and future work

The temperature controlled fan circuit is useful, easy to make and very efficient. The components used are easy to find and cheap. The designed circuit works when the temperature is greater than 30°C, at that point the fan starts running and the speed increases in regular interval according to the increase in temperature between 30°C to 50°C. When the temperature reaches 50°C the 100% of the speed of the fan is reached and then when the temperature increases with a single more °degree celsius than the buzzer starts ringing as the speed of fan can't increase now with that increment in the temperature. Everytime from the start of the circuit, the temperature and the speed of the fan is visible in the 16X2 LED display. This is the main base of the project. For future work we can increase the efficiency and the gap of the temperature where there fan is working. For future we can attach another devices which can be very helpful in our day-to-day life. We can add some more features to our project in future if needed.

5 REFERENCE

1) https://www.arduino.cc/

2) https://components101.com/sensors/lm35-temperature-sensor

3) https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld

4) https://youtu.be/p34w6ISouZY

5) https://en.wikipedia.org/wiki/Arduino_Uno