American International University- Bangladesh (AIUB)
Department of Electrical and Electronic Engineering
EEE4103: Microprocessor and Embedded Systems Laboratory

Course Outcome Mapping with the OEL:

CO/ CLO Number	CO/CLO Statement	K	P	A	Assessed Program Outcome Indicator	BNQF Indicator	Teaching-Learning Strategy	Assessment Strategy
1	Simulate laboratory experiments using microcontrollers, sensors, actuators switches, display devices, etc., and a suitable simulator related to the fields of electrical and electronic engineering.		P1, P4, P5		P.e.2.P4	FS.6	Practical Demonstration	OEL Report

Course Name:	Microprocessor and Embedded System Laboratory	**Course Code:**	EEE4103
Semester:	Summer 2022-2023	**Section:**	F
Faculty Member:	Md Sajid Hossain	**Group #**	6

OEL Lab Title:	Temperature Dependent Motor Control
Submission Link:	https://forms.microsoft.com/r/H2qfsUXZat

SL	Student ID #	Student Name	Obtained Marks
1.	20-42461-1	Rifat Hossain	
2.	20-42831-1	Ishrat Jahan	
3.	21-45019-2	Ahnaf Abdullah Zayad	
4.	21-45038-2	Srabone Raxit	
5.	21-45206-2	Kazi Ramisa Samiha	
6.	21-45263-2	Shakibul Hasan	

Assessment Materials and Marks Allocation:

COs	CO Statement	Assessment Materials	POIs	Marks
CO1	*Simulate laboratory experiments using microcontrollers, sensors, actuators switches, display devices, etc., and a suitable simulator related to the fields of electrical and electronic engineering.*	Open Ended Laboratory Report	P.e.2.P4	10

Assessment Rubrics:

COs-POIs	Excellent [9-10]	Proficient [7-8]	Good [5-6]	Acceptable [3-4]	Unacceptable [1-2]	No Response [0]	Secured Marks
CO1 P.e.2.P4	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are clearly demonstrated combining all input patterns with several outcomes.	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are clearly demonstrated with some outcomes and limited input patterns.	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are not clearly demonstrated with some outcomes and input patterns.	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. The simulation and implementation processes are not clearly demonstrated with a few outcomes for a few patterns.	The OEL developed as a process for complex engineering problems considering microcontrollers, sensors, switches, display devices, etc. are not appropriate. The simulation and implementation processes are not demonstrated with any outcomes and not for any pattern.	No responses at all	
Comments					**Total marks (10)**		

Temperature Dependent Motor Control

I. Objective

The main purpose of this experiment is to adjust the speed of a motor according to the temperature, i.e., if the temperature is high, motor speed will increase and if temperature is low, motor speed will decrease.

II. Theory and Methodology

To drive machine shafts, motors are commonly used as rotating equipment. Direct current motors are one of many different types of motors. The Direct current motor is essential to modern industry because it may be used in a variety of applications where the load changes with speed and necessitates a strong dynamic response. As temperature increases, the resistance of materials within the motor also increases. This elevated resistance leads to greater power losses, reducing the motor's efficiency and potentially diminishing its overall performance. Maintaining optimal operating temperatures is essential to ensure efficient and reliable motor operation.

In this experiment, a DC motor is setup where, as the value of temperature detected increases, the speed of the motor also increases. According to the Figure 4.1/4.2/4.3 in experimental setup, the circuit includes a DC motor, Arduino UNO, L298N H-bridge and a BMP180 temperature sensor. The Arduino UNO board is a collection of the essential Integrated Circuits and components for creating both simple and advanced control circuits. A L298N Dual H-bridge, often used with Arduino, can control 2 DC motors at the same time, along with their speed and direction. The module has two screw terminal blocks for motor A and B, another screw terminal block for the Ground pin, the VCC for the motor, and a 5V pin which can either be an input or output. The BMP180 or MPL115A is a well-suited device with its pressure range and resolution. Barometric pressure changes can directly correlate to changes in the weather.

For this experiment the concepts of SPI, USART, I2C and PWM are also implemented. A temperature sensor can be linked to the microcontroller using SPI, USART and I2C. They are used to transfer temperature values from the sensor to the microcontroller. Based on temperature, the microcontroller analyzes the data and modifies motor control parameters. USART offers adaptable serial connection frequently in point-to-point configurations, SPI is utilized for high-speed full-duplex communication, and I2C is built for multi-device communication with a small number of cables, making it ideal for connecting devices close together. PWM can control the speed of a motor's fan. The microprocessor may raise the PWM duty cycle as temperature rises, which will increase the fan speed. Increased fan speed improves cooling, keeping the motor from overheating.

III. Apparatus

a. Arduino UNO
b. LED Indicators (3 pieces - Red, Yellow, and Green)
c. Temperature Sensor
d. Breadboard
e. DC Motor
f. H-bridge Motor Driver
g. 1 KΩ Resistors (3 pieces)
h. Connecting Wires

IV. Experimental Setup

The pictures below show the implemented circuit.

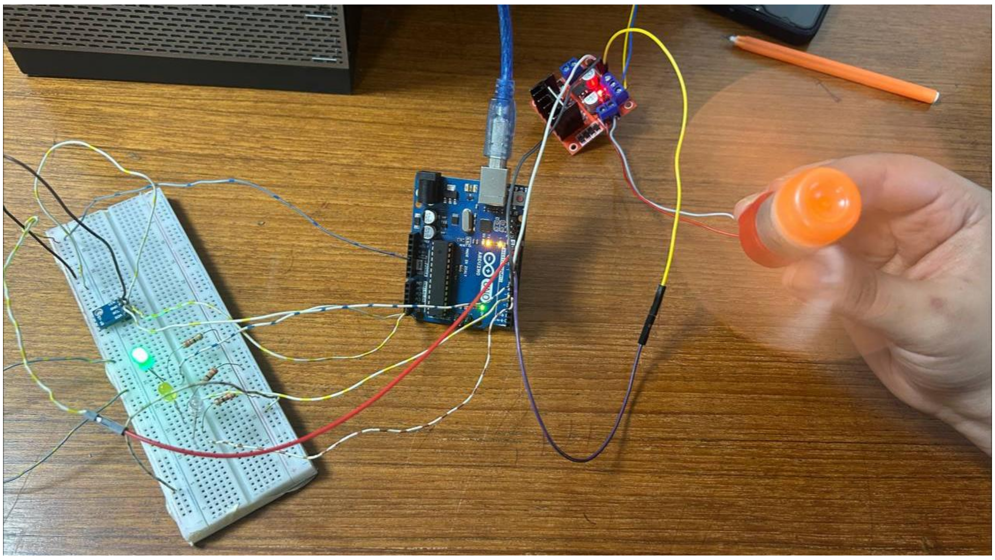

Figure 4.1

As shown in Figure 4.1, Green LED is ON. This means that the temperature is below 25 °C. Hence, speed (in Rpm) is decreasing.

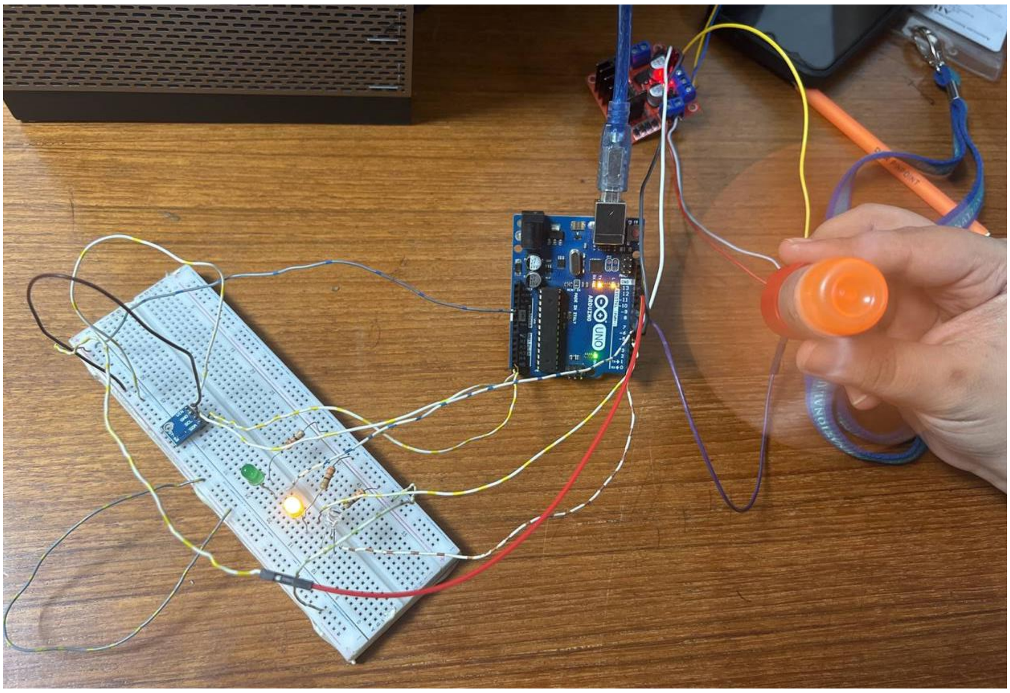

Figure 4.2

As shown in Figure 4.2, Yellow LED is ON. This means that the temperature is between 25 °C to 30 °C. Hence, speed (in Rpm) is increasing.

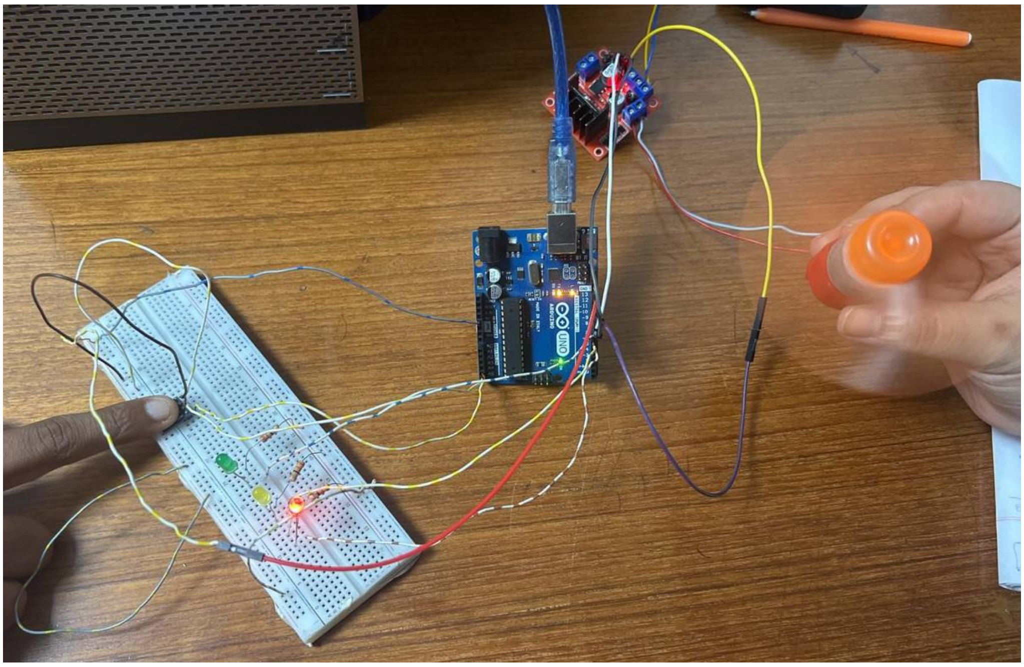

Figure 4.3

As shown in Figure 4.3, Red LED is ON. This means that the temperature is above 30 °C. Hence, speed (in Rpm) is further increasing.

V. Simulation

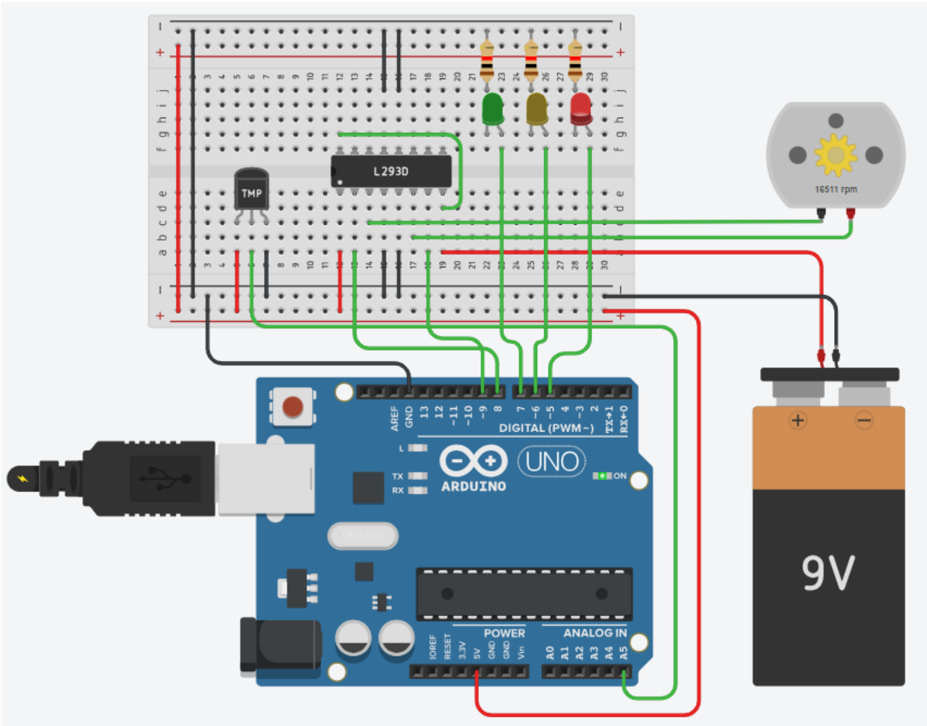

Figure 5.1: Red LED ON

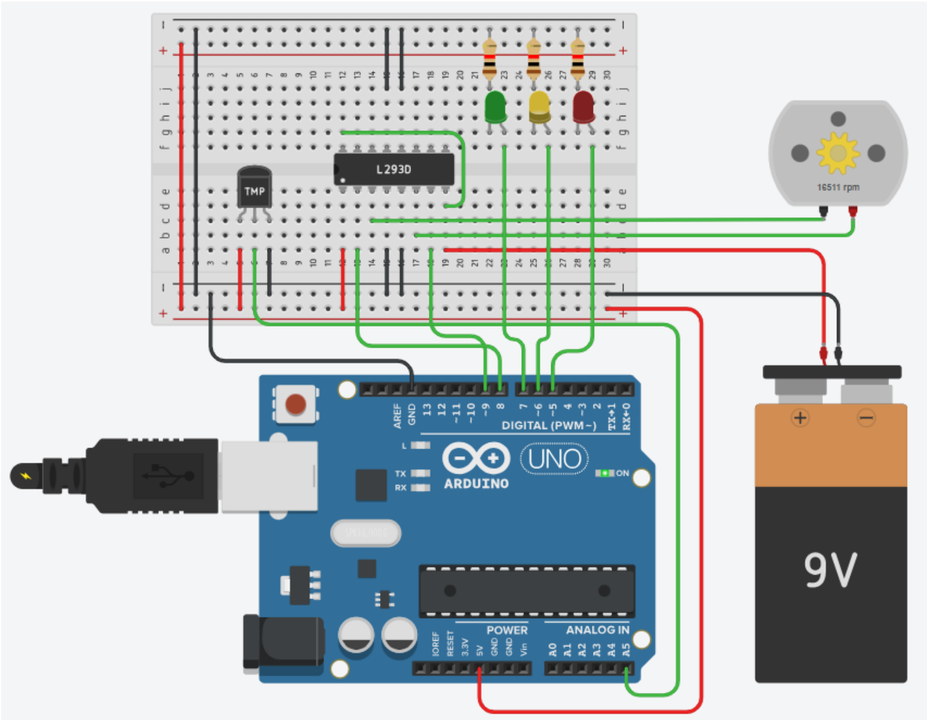

Figure 5.2: Yellow LED ON

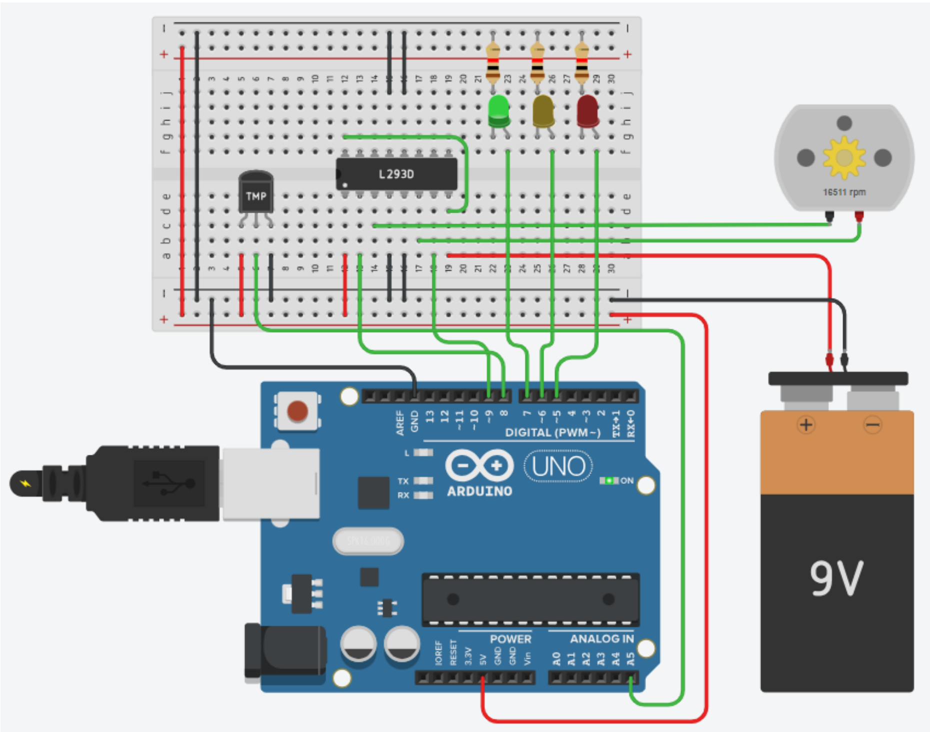

Figure 5.3: Green LED ON

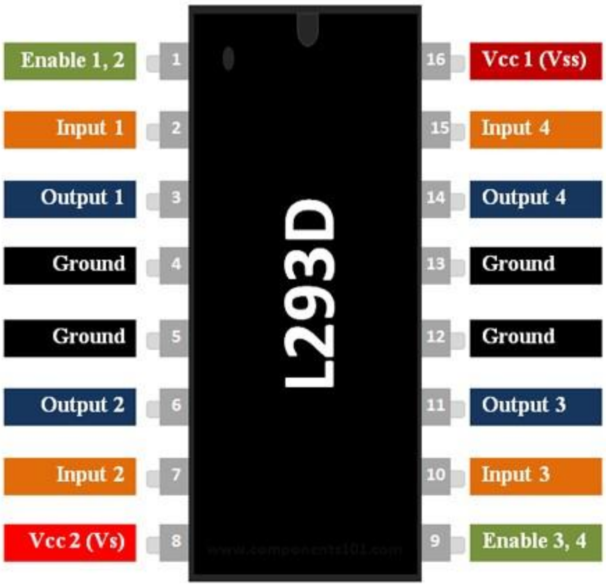

Figure 5.4: L293D IC Chip (H-Bridge Motor Driver)

VI. Code of the Program

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_BMP085.h>

int in1 = 8;
int in2 = 9;
int ConA = 10;

Adafruit_BMP085 bmp;

void setup() {
  Serial.begin(9600);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);

  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP085 sensor, check wiring!");
    while (1) {}
  }
}

void TurnMotor(int power) {
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
```

```
  analogWrite(ConA, power);
}

void loop() {
   int temp = bmp.readTemperature();
   Serial.println(temp);

   if (temp > 30) {
     digitalWrite(5, HIGH);
     digitalWrite(6, LOW);
     digitalWrite(7, LOW);

     int motorSpeed = map(temp, 30, 35, 250, 250);
     // Adjust the upper temperature limit as needed

     TurnMotor(motorSpeed);
     Serial.println(motorSpeed);
   }
   else if (temp >= 25 && temp <= 30) {
     digitalWrite(5, LOW);
     digitalWrite(6, HIGH);
     digitalWrite(7, LOW);

     int motorSpeed = map(temp, 25, 30, 150, 200);
     // Adjust the temperature range and speed values as needed

     TurnMotor(motorSpeed);
     Serial.println(motorSpeed);
   }
   else if (temp < 25) {
     digitalWrite(5, LOW);
     digitalWrite(6, LOW);
     digitalWrite(7, HIGH);

     int motorSpeed = map(temp, 20, 25, 100, 100);
     // Adjust the lower temperature limit as needed

     TurnMotor(motorSpeed);
     Serial.println(motorSpeed);
   }
}
```

VII. Data Collection Table

Table 7.1: Data Collected from Serial Monitor

Speed (Rpm – rotation per minute)	Temperature (°C – Degree Centigrade)
100	24
150	25
160	26
170	27
180	28
190	29
200	30
250	31

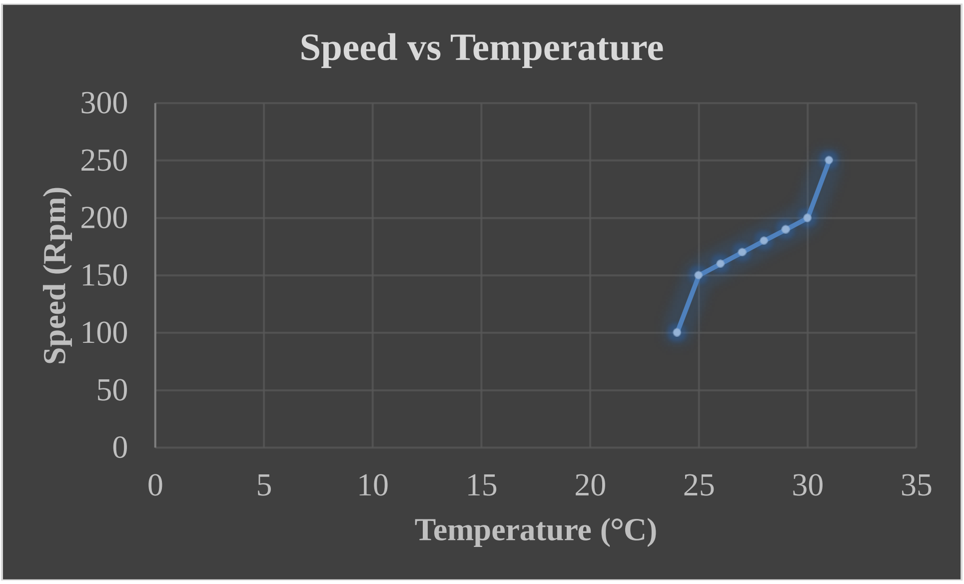

Figure 7.1: Speed vs Temperature Graph

From Figure 7.1, it can be illustrated that as temperature increases, speed is increasing. If the graph is more carefully observed, there is a steep increase in speed between 24 °C to 25 °C and 30 °C to 31 °C. This is because according to the implemented code, temperatures below 25 °C will have lower speed, temperatures between 25 °C to 30 °C will have moderate speed (increasing steadily throughout the range), and temperatures above 30 °C will have higher speed.

VIII. Discussions

In this experiment, the goal was to implement a circuit where motor speed would be changed according to temperature. As the temperature was being changed, the motor speed changed accordingly. Expected outcomes were observed from the experiment. Hence, the experiment was successful. To further validate the experimenter, simulation of the same circuit was also conducted. That too produced coherent results.

IX. References

[1] Esam Faleh Esam Alajmi1 and Soud Mohammed alhajri (2021) *Control System design for motor based on temperature dynamics - ijisrt.com*, *Control System Design for Motor based on Temperature Dynamics*. Available at: https://ijisrt.com/assets/upload/files/IJISRT21JUL649.pdf (Accessed: 11 August 2023).

[2] Admin (2022) *Temperature based fan speed controller using Arduino*, *Temperature Based Fan Speed Controller using Arduino*. Available at: https://how2electronics.com/temperature-based-fan-speed-controller-using-arduino/ (Accessed: 12 August 2023).

X. Appendix

Table 10.1: Contribution Table

Student Name	ID	Contribution
Rifat Hossain	20-42461-1	Simulation
Ishrat Jahan	20-42831-1	Objective, Apparatus, and Discussions
Ahnaf Abdullah Zayad	21-45019-2	Experimental Setup
Srabone Raxit	21-45038-2	Code of the Program
Kazi Ramisa Samiha	21-45206-2	Theory and Methodology, and References
Shakibul Hasan	21-45263-2	Data Collection and Analysis