CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

AUTOMATIC FAN

GROUP B-8

Michael Winston Tjahaja	**2106732170**
Rian Abrar Makarim Lubis	**2106708242**
Raditya Ihsan Dhiaulhaq	**2106733912**
Adrien Ardra Ramadhan	**2106731485**

PREFACE

Praise and gratitude be to Allah SWT, for His blessings and guidance that has enabled our team to successfully complete our final project on Cyber-Physical Systems titled "Automatic Fan" right on time.

This project was made and done with the aim to conclude the final project of Cyber-Physical Systems. Walaupun demikian, penyusun berusaha dengan semaksimal mungkin demi kesempurnaan penyusunan laporan ini.

Within this report, we explain the introduction, implementation, and the complete evaluation of the project that we've worked on.

Both suggestions and criticism with the intent of helping us build off of our mistakes and strengths is well appreciated in the pursuit of reaching a near perfect report.

All in all, we hope that this report that we've made as a group brings many benefits for it's future and current readers as to increase their understanding of the topic Cyber-Physical Systems.

We give thanks to all parties that have helped us finish this report and we apologize for any mistakes that we've made within this report.

Depok, 14 Mei, 2023

Group B-8

TABLE OF CONTENTS

CHAPTER 1 .... **4**

INTRODUCTION **4**

1.1 PROBLEM STATEMENT 4

1.3 ACCEPTANCE CRITERIA 5

1.4 ROLES AND RESPONSIBILITIES 5

1.5 TIMELINE AND MILESTONES 5

CHAPTER 2 .... **7**

IMPLEMENTATION **7**

2.1 HARDWARE DESIGN AND SCHEMATIC 7

2.2 SOFTWARE DEVELOPMENT 7

2.3 HARDWARE AND SOFTWARE INTEGRATION 8

CHAPTER 3 .... **9**

TESTING AND EVALUATION **9**

3.1 TESTING 9

3.2 RESULT 9

3.3 EVALUATION 10

CHAPTER 4 .... **11**

CONCLUSION **11**

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

During everyday life, we obviously need technology to be able to take care of our heat problems, such as the room being too hot, or if you've just finished exercising, especially with the very recent and extreme climate change Indonesia has been facing. The most recent record of heat was on 14'th May 2023 where temperatures reached 35 degrees Celsius and in this case, it would be very convenient if you had an automatic fan that would help with the heat.

There is a problem however, where conventional fans will only create winds at a steady level without being able to detect their surroundings such as the current room temperature and humidity. That will obviously create an ineffective environment where we would have to manually set the speed of the wind. Therefore, we've come up with a solution to tackle this problem when using a conventional fan.

1.2 PROPOSED SOLUTION

To overcome the previously stated problem, where conventional fans feel less ineffective to make a room feel cooler, we within this B-8 group have decided to take initiative to tackle this problem. That's why we're coming up with what we call the Auomated Fan. This tool will be able to automatically control the speed of its rotations depending on the current room temperature.

Within this project, we are using an Arduino as the main controller to control the speed of the fan-blades depending of the area of where the tool is currently situated in. With that being said, this project is hoped to be able to give an effective and efficient solution to handling heat and make fans less of a hassle to use thanks to automation.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. Sistem must be able to detect heat using the DHT11
2. MAX7219 must be able to display the detected temperature from the DHT11 sensor
3. Servo / Motor must be able to spin when the temperature hits a certain threshold
4. LED indicates the level of temperature
5. The HCSR04 sensor is capable of detecting distance to determine whether motor turned on or not

1.4 ROLES AND RESPONSIBILITIES

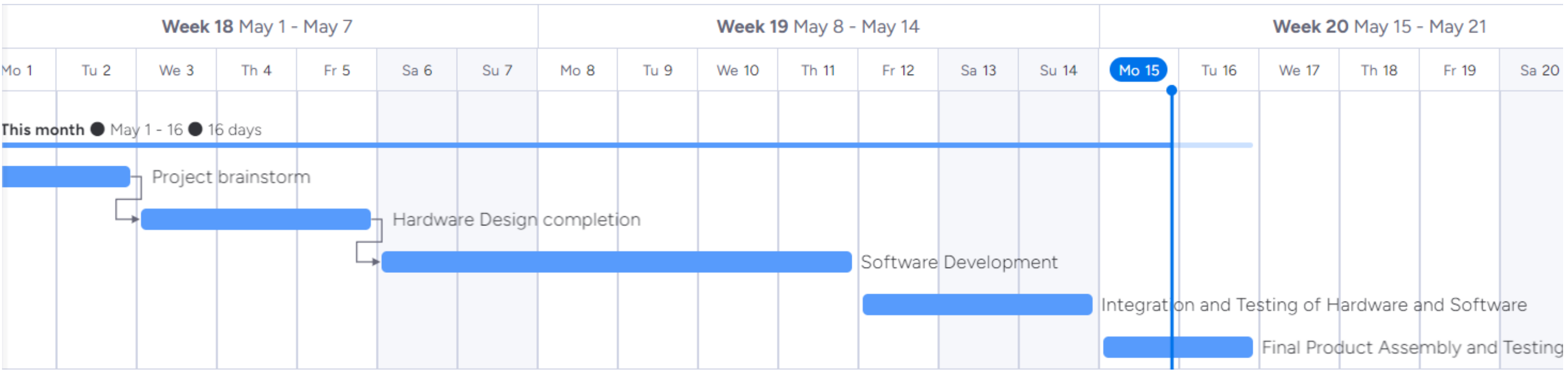
The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Practitioner
Role 1	Building the physical circuit, writing the project report, circuit testing, creating the PPT, and participating for the combination of all the codes into a singular code.	Adrien Ardra Ramadhan
Role 2	Creating the READMe, building the physical circuit, simulating the Proteus code, running the main code, creating the condition to run the DC motor, PPT creation, writing and translating the report into English.	Michael Winston Tjahaja
Role 3	Creating the DHT11 + MAX7219 code,	Raditya Ihsan Dhiaulhaq

	combining the entirety of the code, creating and simulating the Proteus code, PPT creation, debugging most of the problems that appeared within the final code.	
Role 4	Participated in creating the PPT, making the servo code and combining the main program.	Rian Abrar

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES



CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

The hardware that we’ve managed to come up with has an Arduino Uno as a that would include all the necessary libraries and logic to be able to control the servo, read the DHT11 sensor’s data, and to display the information gained from said sensor on the Max7219 display. There is also a HCSR04 sensor that will determine whether or not the motor will be turned on or off.

There will also be 3 LEDs connected to the Arduino Uno that will show what the current state of the temperature sensor is in and will be the main physical sign that the controller has switched modes.

2.2 SOFTWARE DEVELOPMENT

The software development is written in Assembly code, specifically for an ATmega8 microcontroller. The code we have written defines a few global functions, including `SPI_MAX7219_init, MAX7219_disp_text, DHT11_sensor, HC_SR04_Sensor and defines the entry point of the program as main.

The `main` function is a loop that calls the function DHT11_sensor that will jump back to the beginning of the loop. The function itself is used to read the incoming data from DHT11 in the form of both temperature and humidity. The main microcontroller will then send a start and response signal towards the DHT11 and will read both temperature and humidity values and send those values back to the microcontroller. Based on the temperature value, it branches into three different conditions: “first_condition,” “second_condition,” and “third condition.”

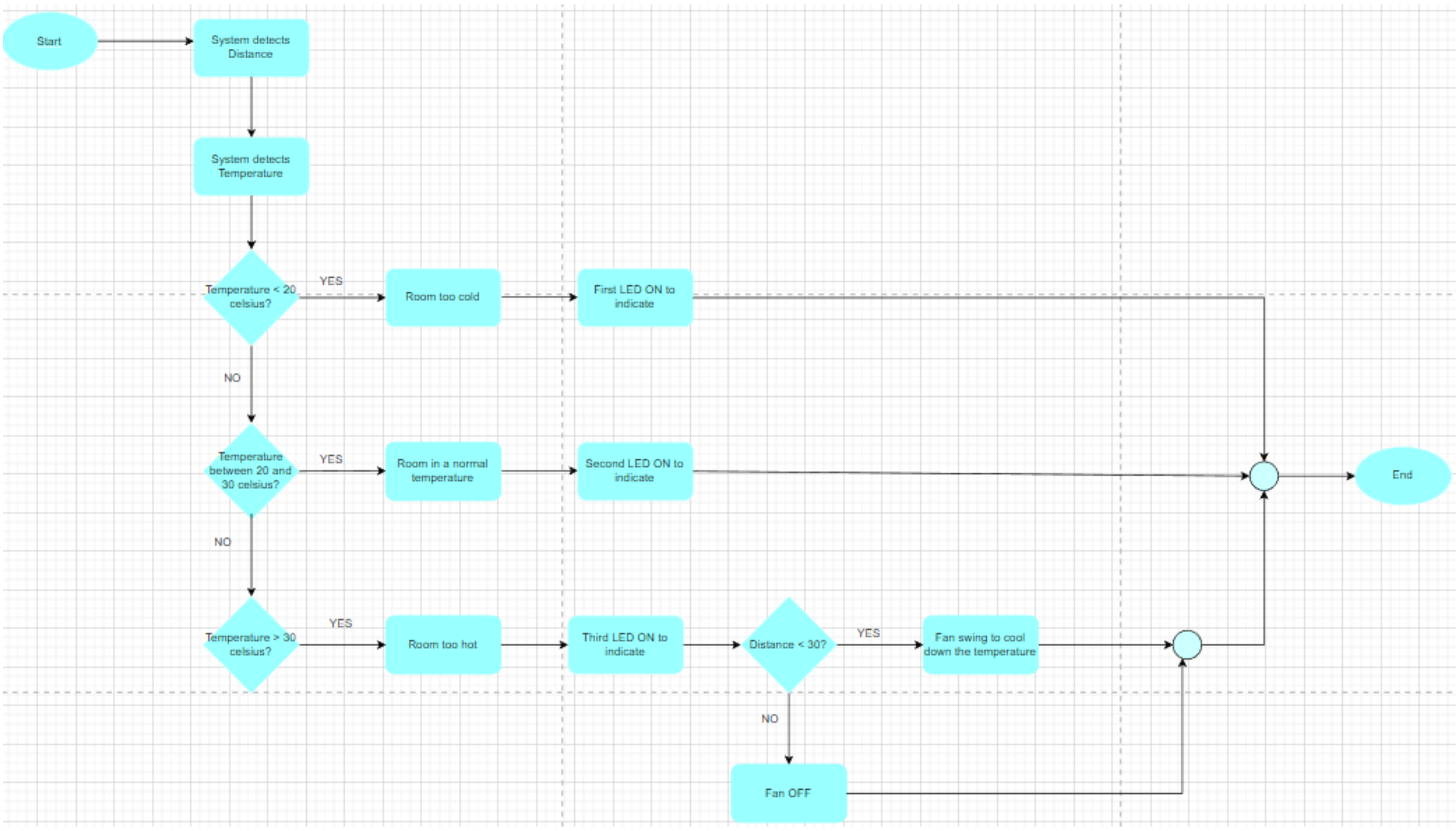
The first condition represents cold temperatures underneath 20 degrees celsius. When this condition appears, the bits 5 and 6 will be cleared and the motor will be turned off. Bit 4 will then be set to turn on an LED situated on the left to signal that this condition is on

The second condition corresponds to normal weather conditions around 25 degrees celsius. This will then clear bit 4 and 6 (in PORTD) to turn off the left and right LEDs if they were on and set bit 5 (in PORTD) to turn on the middle LED. It will also lower the speed of which the motor spins at.

The third condition represents hot temperature where the temperature is above 30 degrees. This will clear bits 4 and 5 and will set bit 6 to turn on the right LED to show that this condition is true. It then sets the motor or the DC motor to spin at maximum speed to cool off the room at maximum efficiency.

Both the temperature and humidity values from the DHT11 will be sent from the Arduino Uno to the MAX7219 display using the MAX7219_disp_text function using the SPI protocol.

The function HC_SR04 will first initialize the sensor HCSR04 first. It will then receive the data given by said sensor and will be the main determining factor which turns on/off the motor. The given parameter is whether or not the sensor detects an object within 30 centimeters off its field.



2.3 HARDWARE AND SOFTWARE INTEGRATION

To integrate both the hardware and software, we will be programming an ATmega328p microcontroller using the Assembly Language. First, we started by building the main circuit using the 'Proteus' software to be able to have a clear map of what we were going to make using an actual breadboard. We used a few hardware components such as; a DHT11 sensor, 3 LEDs, a DC Motor, and a HCSR04 sensor. After we've built the virtual circuit, we programmed using Arduino IDE to determine the way the circuit works. The LEDs are set to turn on/off depending on the temperature of where the circuit is situated in. When the temperature is below 20 degrees celsius the far left LED will turn on and the DC Motor will stay off. The next condition is where the temperature is between 20 degrees celsius and 30 degrees celsius which will turn on the middle LED while also turning on the DC Motor. The final condition is when the circuit detects temperatures above 30 degrees Celsius where the far right LED will turn on and the motor will also turn on.

There's also another factor in whether or not the circuit turns on or not, where if the HCSR04 detects an object within 30 centimeters of it's range, it will turn on. If there are no objects detected within the range of the sensor, the circuit will turn off (This does not affect whether or not the LEDs turn on or off).

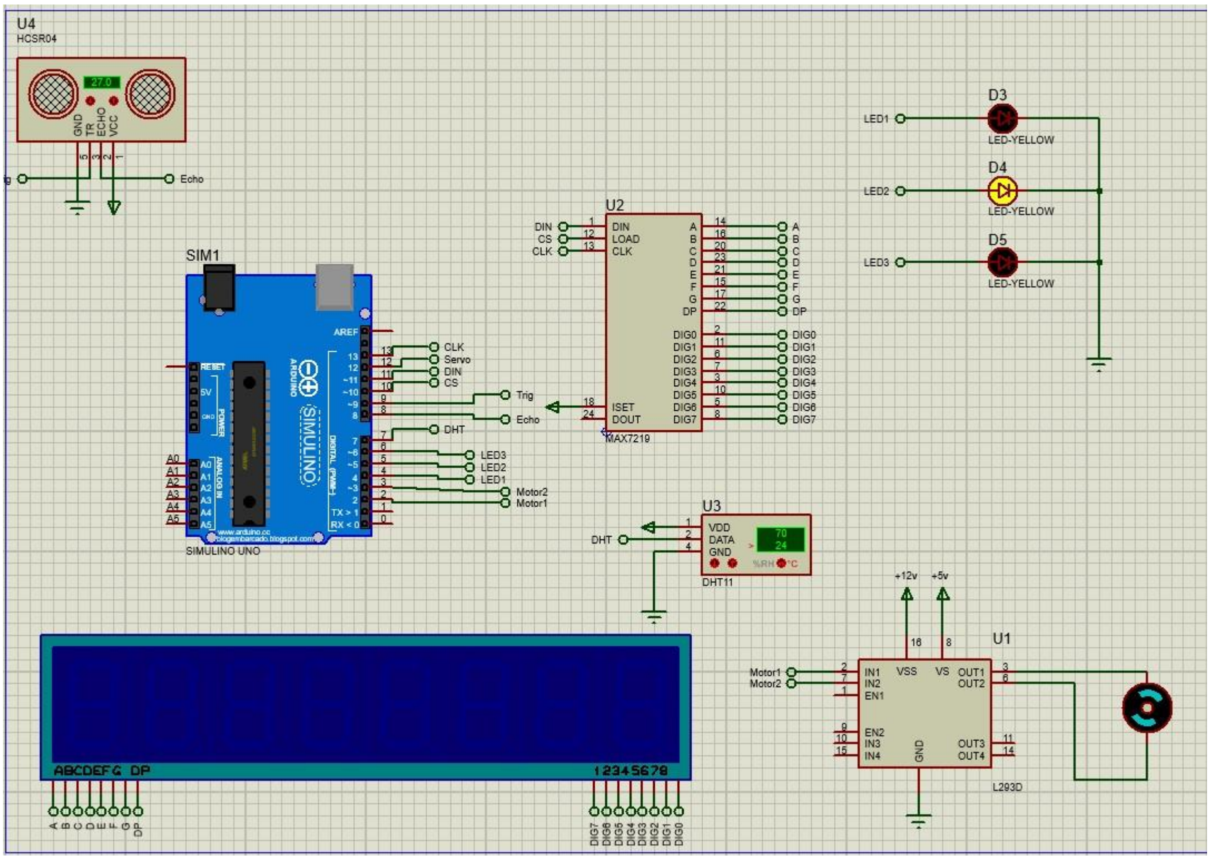
Once the code runs accordingly within the Proteus Simulation, the actual circuit is then made. We then started the testing on the real circuit to see if there were any mistakes and to troubleshoot those problems within the circuit or the code itself. Once that was done the project had concluded.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

For the testing, we made the following Proteus Simulation Circuit:

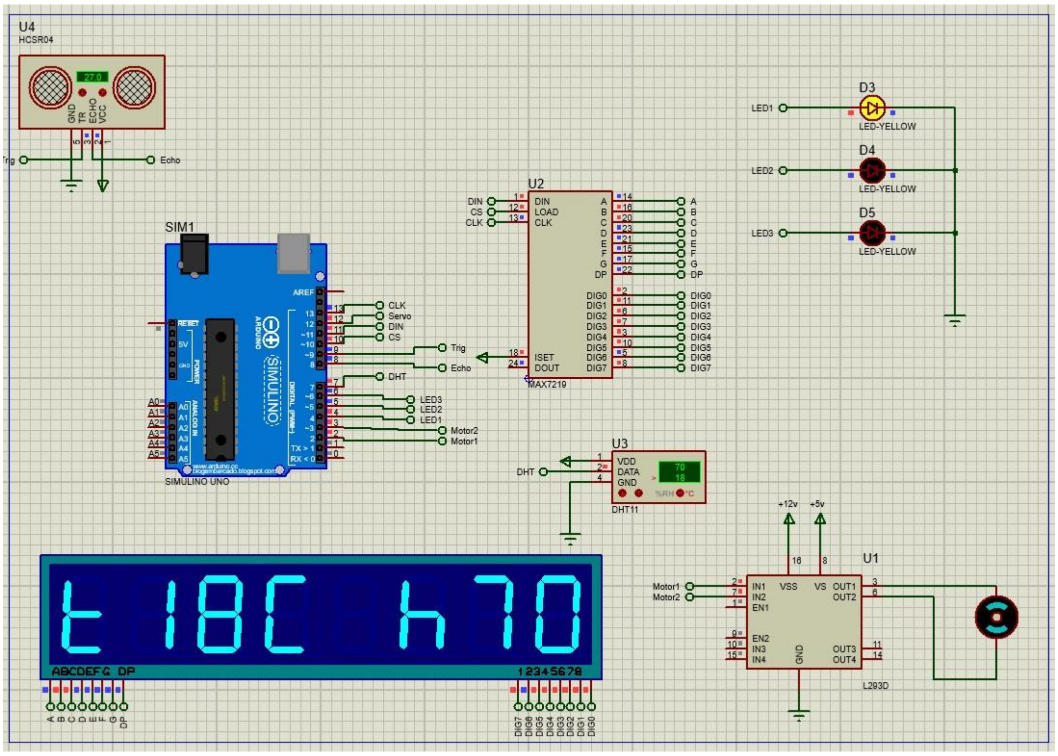


The testing we did followed a few conditions which were; when the temperature was below 20 degrees celsius and if there was an object >/< 30 centimeters away from the HCRS04 sensor, the temperature was between 20 and 30 degrees celsius and if there was an object >/< 30 centimeters away from the HCRS04 sensor, and finally, when the temperature is above 30 degrees celsius and if there was an object >/< 30 centimeters away from the HCRS04 sensor.

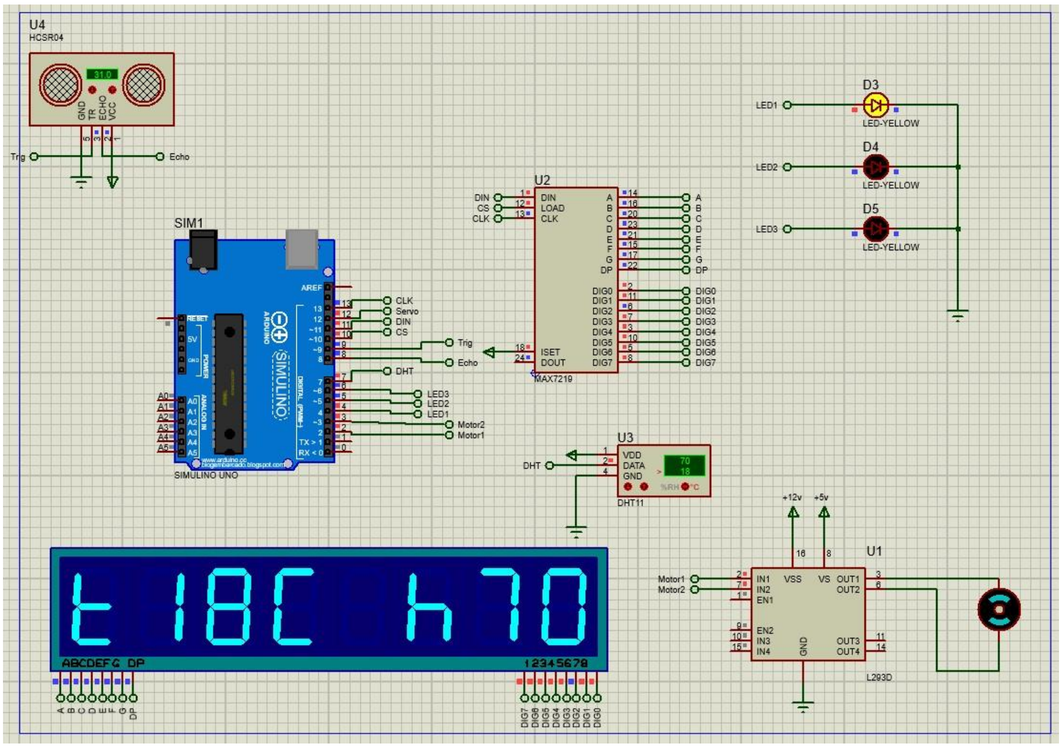
3.2 RESULT

The following were the results of the testing

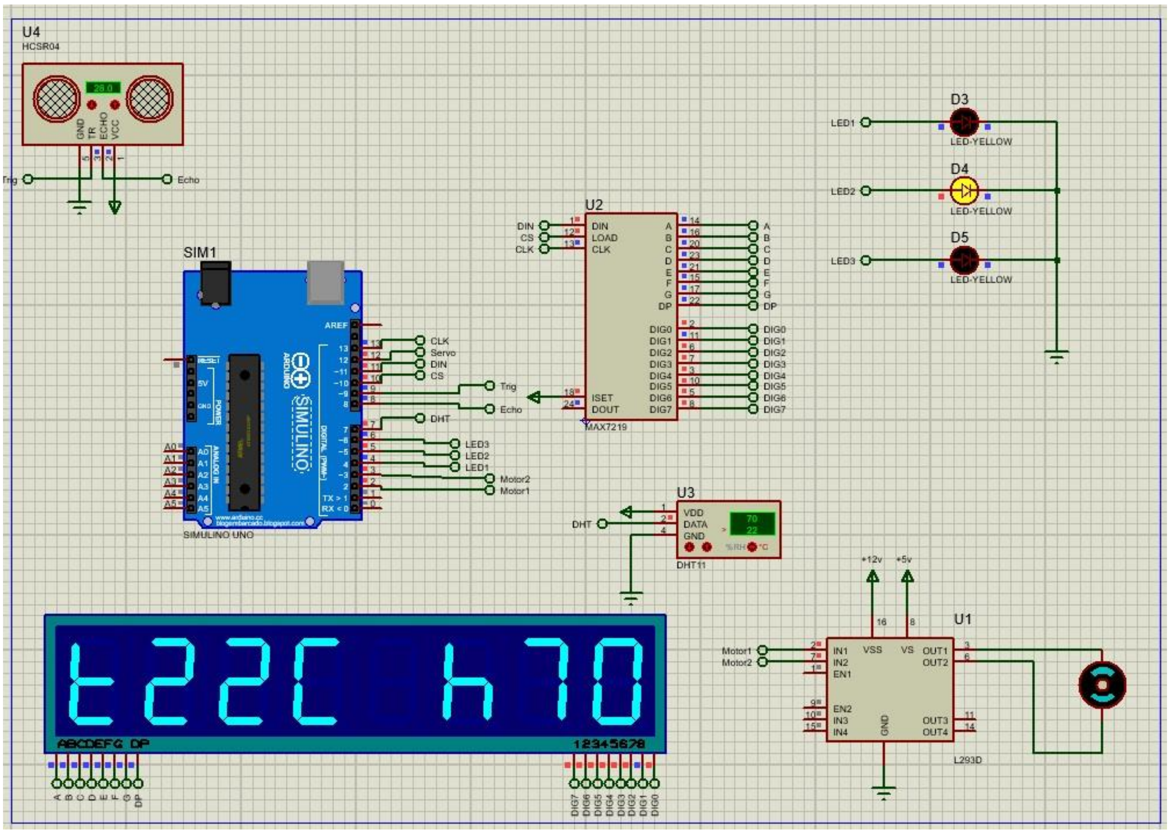
- Temperature is below 20 degrees celsius with an object within the 30 centimeter range



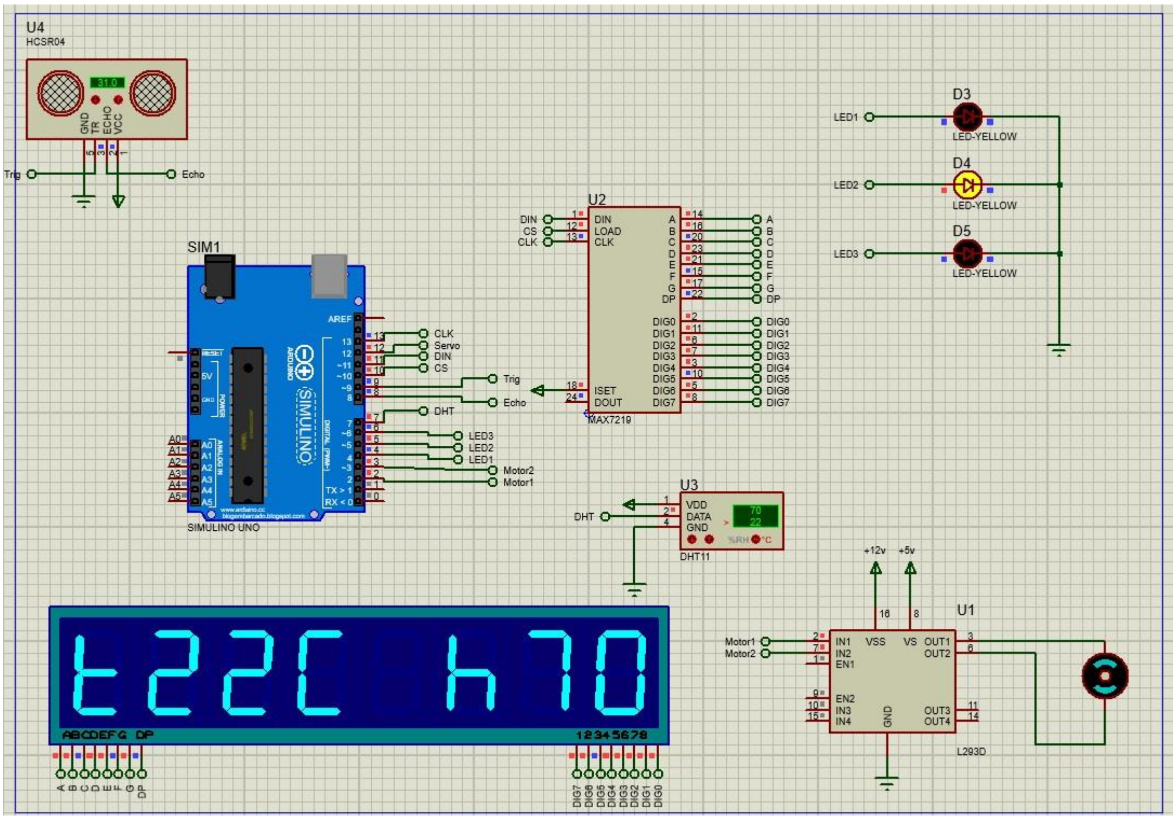
- Temperature is below 20 degrees celsius with an object outside of the 30 centimeter range



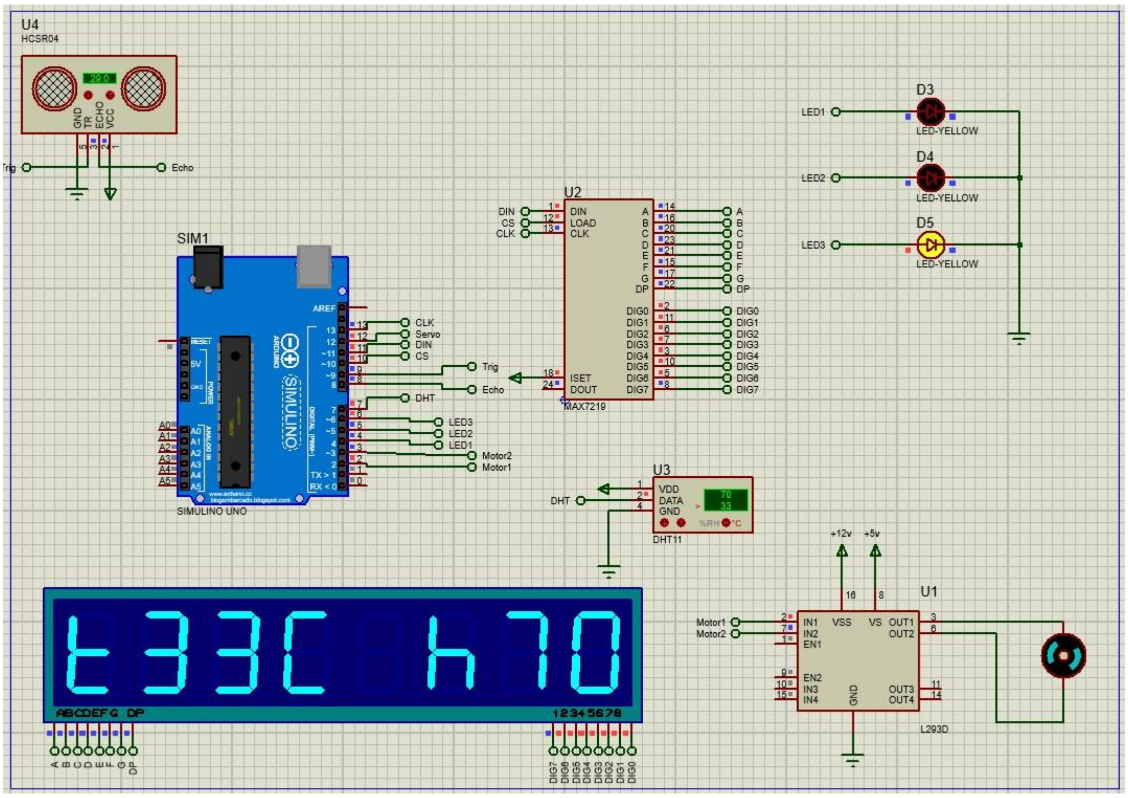
- Temperature is between 20 - 30 degrees celsius with an object within the 30 centimeter range



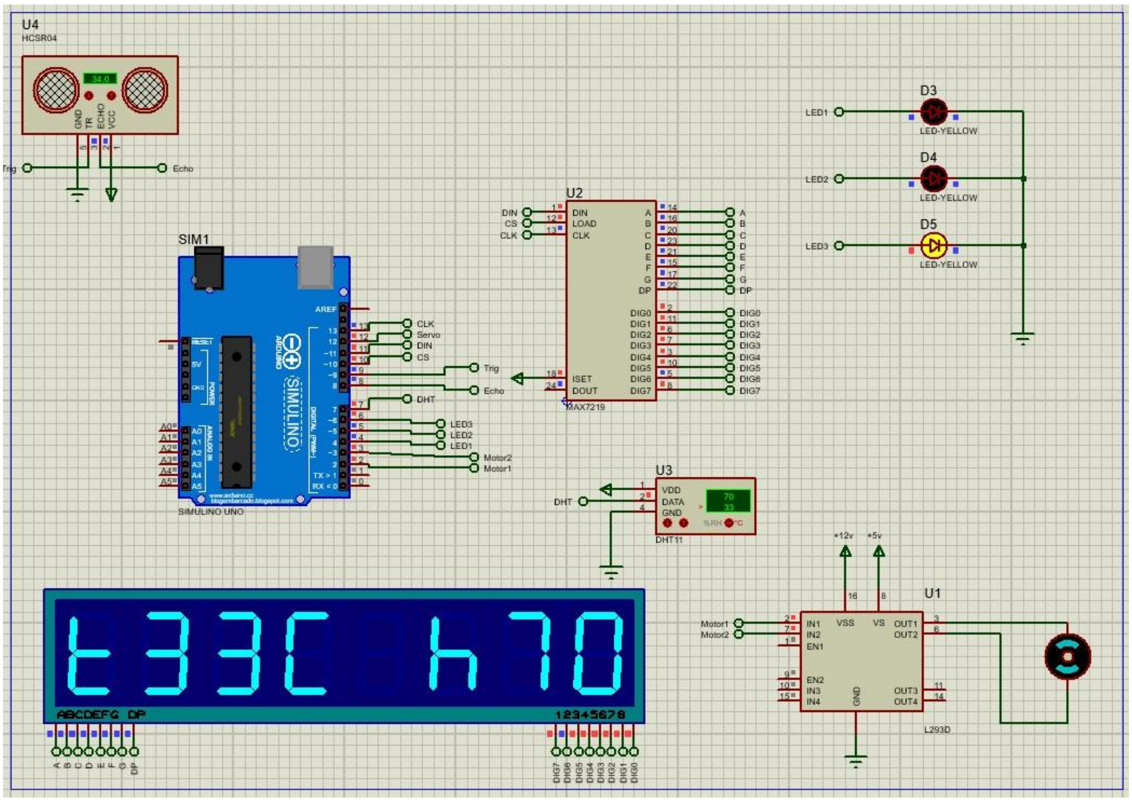
- Temperature is between 20 - 30 degrees celsius with an object outside of the 30 centimeter range



- Temperature is above 30 degrees celsius with an object within the 30 centimeter range



- Temperature is above 30 degrees celsius with an object outside of the 30 centimeter range



3.3 EVALUATION

Once the practitioners finished the Automatic Fan project, starting from designing the Proteus Circuit, writing the Assembly code, and finalizing the physical circuit, there were a few evaluation points that can be pointed out. Because of the lack of experience with making the physical circuit, there were a few mistakes with designing it; The Arduino Uno microcontroller is unable to turn on the motor. There was also a problem with turning on the servo (a component that was previously within the main circuit) where the spin of said servo was inconsistent; sometimes it would spin and sometimes it wouldn’t. Practitioners also found a mistake when displaying the heat and humidity using the MAX7219 display where the seven segments could not display the writing correctly. Practitioners also failed to implement the humidity from the DHT11 into the way the entire circuit worked. With that being said, the practitioners have gained a good amount of experience through this project to be used as evaluation material for future and upcoming projects.

CHAPTER 4

CONCLUSION

This Automatic Fan project is well suited to use for a wide range of people as it tackle a very important problem that current conventional fans lack; automation. This Automatic Fan can be used to detect the surrounding heat of a room and will turn on accordingly if there is a person within the fan itself to conserve the power to turn on the fan itself.

The Automatic Fan can detect 3 levels of heat; The first level being where the left LED turns on when temperatures are low (< 20 degrees celsius. The second level is when the temperatures are between 20 degrees celsius and 30 degrees celsius which will turn on the motor and light up the middle LED. The last level will turn on when temperatures are above 30 degrees celsius where the right LED will turn on which states that the room is far too hot and will turn on the motor accordingly to cool down the room.

REFERENCES

[1] Assembly via Arduino (part 26) - servo motor SG90 (2021) YouTube. Available at: https://www.youtube.com/watch?v=ZqbKUDZw6vM (Accessed: 11 May 2023).

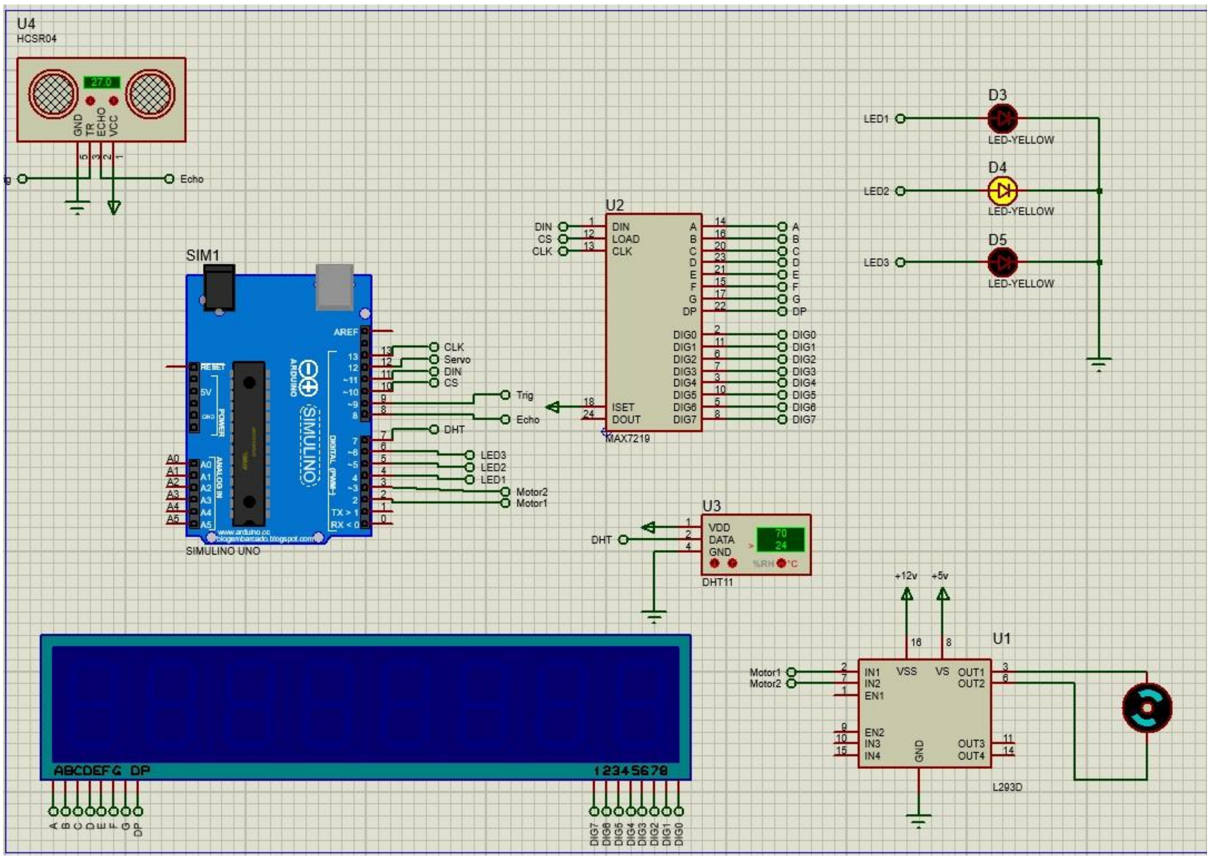
[2] *Assembly via Arduino (part 21) - HC SR04 sensor* (2021) *YouTube*. Available at: https://www.youtube.com/watch?v=dIIBG7Qz-_E&pp=ygUdaGNzcjA0IGFzc2VtYmx5IGFuYXMga3V6ZWNoaWU%3D (Accessed: 11 May 2023).

[3] Assembly via Arduino (part 20) - DHT11 data on MAX7219 display (2021) YouTube. Available at: https://www.youtube.com/watch?v=UjWbE2Q75ms&pp=ugMICgJpZBABGAHKBR5tYXg3MjE5IGFzc2VtYmx5IGFuYXMga3V6ZWNoaWU%3D (Accessed: 11 May 2023).

[4] *Assembly via Arduino (part 19) - dht11 sensor* (2021) *YouTube*. Available at: https://www.youtube.com/watch?v=vnLpzvkCUq8&pp=ugMICgJpZBABGAHKBRxESFQxMSBhc3NlbWJseSBhbmFzIGt1emVjaGll (Accessed: 11 May 2023).

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation