SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

ELECTRICAL AND ELECTRONIC ENGINEERING

SOLAR POWERED AUTOMATED HYDROPONIC CULTIVATION

Thesis submitted to Sri Lanka Institute of Information Technology in partial fulfilment of the requirement for the EC3061 Design Project Module semester II.

Jayatissa K.T.D

EN19379616

Group Number: 33

Supervised by: Dr. Lasantha Seneviratne

Coordinated by: Mr. Nushara Wedasingha

Acknowledgements

Countless people supported my effort on this project. First of all, I would like to express my sincere gratitude to my supervisor, Dr. Lasantha Senevirathna, for your invaluable advice, guidance, encouragement, motivation, and support throughout this project. I would also like to thank the lecturers, lab assistants, colleagues of the Sri Lanka Institute of Information Technology for helping me to make this project a success. And finally, I would like to say a special thank you to my group members, my family, and to everyone who helped us throughout this project.

Abstract of Thesis presented by Jayatissa K.T.D to Sri Lanka Institute of Information Technology.

SOLAR POWERED AUTOMATED HYDROPONIC CULTIVATION

ABSTRACT:

Throughout human history, the soil has been the foundation of agriculture. Soil cultivation is the most common cultivation method throughout the past and even at the present. Due to many reasons soil cultivation is facing many challenges and is turning out to be a tough challenge to the farmers all over the globe.

In the first semester, we have implemented an alternative solution for soil cultivation. The future development and modification for that product using more efficient technologies will be discussed in this thesis. The product we implemented to tackle the soil cultivation problems is an IOT based solar-powered automated hydroponic cultivation system in which all the necessary parameters needed for plant growth will be monitored and supplied adequately through an automated system. This product is developed using fuzzy logic controllers to make the parameter controlling more efficient and two essential parameters are monitored and controlled in addition to the parameters controlled before. Our product would prove highly valuable to this emerging hydroponic consumer market, as we plan to construct a fully automated, inexpensive, and compact hydroponic system.

List of Abbreviations

Abbreviations

IOT – Internet of Things

FLC - Fuzzy Logic Controller

pH – Potential of Hydrogen

Table of Contents

Chapter 1 10

INTRODUCTION 10

1.1 Research Objectives 14

1.2 Contribution of the Thesis 15

1.3 Structure of the Thesis 16

Chapter 2 17

LITERATURE REVIEW 17

2.1 Introduction 17

2.2 Background and Motivation 17

2.3 Similar Existing Projects 19

2.3.1 Temperature Control using Fuzzy Logic 19

2.3.2 Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse 20

2.3.3 Fully Automated Hydroponic System for Indoor Plant Growth 23

2.4 Summary 24

Chapter 3 25

METHODOLOGY OF OUR WORK 25

3.1 Introduction 25

3.2 Overview of the proposed work 25

3.3 Individual Contribution 27

3.3.1 Implementation of the water temperature and the CO2 level monitoring systems 27

3.3.2 Designing Fuzzy Logic Controllers 30

3.3.3 Physical implementation 58

3.4 Summary 59

Chapter 4 60

RESULTS 60

Limitations 61

Chapter 5 62

FINAL CONCLUSION 62

References 63

APPENDIX 66

List of Figures

Figure 1.1: Effect of use of hydroponic system compared to soil grown tomato plants on height.

Figure 1.2: Expected global hydroponic system market by 2027

Figure 1.3: The basic configuration of a fuzzy system

Figure 2.1: Structure of the Nutrient Film Technique hydroponic cultivation method

Figure 2.2: The membership function of the input variable of the project “Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse”

Figure 2.3: The membership functions of the output variables of the project “Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse”

Figure 2.4: dynamic model of the agricultural greenhouse modeled using Matlab-Simulink environment of the project “Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse”

Figure 2.5: Web interface of the project “Fully Automated Hydroponic System for Indoor Plant Growth”

Figure 3.1: Basic overview diagram of the proposed project

Figure 3.2: DS18B20 temperature sensor

Figure 3.3: Basic circuit for measuring water temperature

Figure 3.4: mq135 gas sensor

Figure 3.5: Basic circuit for measuring the CO2 level

Figure 3.6: Membership functions for the fuzzy input temperature in Matlab toolbox

Figure 3.7: Membership functions for the fuzzy input light intensity in Matlab toolbox

Figure 3.8: Membership functions for the fuzzy output fan speed in Matlab toolbox

Figure 3.9: Formation of rules in the Matlab toolbox of air temperature FLC

Figure 3.10: Rule viewer for air temperature FLC

Figure 3.11: surface viewer of the rule base of air temperature FLC

Figure 3.12: Simulink model of the air temperature FLC

Figure 3.13: Fuzzy algorithm for the air temperature FLC

Figure 3.14: Membership functions for the fuzzy input humidity in Matlab toolbox

Figure 3.15: Membership functions for the fuzzy output humidity valve in Matlab toolbox

Figure 3.16: Membership functions for the fuzzy output humidity fan in Matlab toolbox

Figure 3.17: Formation of rules in the Matlab toolbox of air humidity FLC.

Figure 3.18: Rule viewer for air humidity FLC

Figure 3.19: surface viewer of the rule base of air humidity FLC

Figure 3.20: Simulink model of the air humidity FLC

Figure 3.21: Fuzzy algorithm for the air humidity FLC

Figure 3.22: Membership functions for the fuzzy input CO2 air state in Matlab toolbox

Figure 3.23: Membership functions for the fuzzy output air pump in Matlab toolbox

Figure 3.24: Formation of rules in the Matlab toolbox of CO2 level FLC

Figure 3.25: Rule viewer for CO2 level FLC

Figure 3.26: surface viewer of the rule base of CO2 FLC

Figure 3.27: Simulink model of the CO2 level FLC

Figure 3.28: Fuzzy algorithm for the CO2 level FLC

Figure 3.29: Membership functions for the fuzzy input water level in Matlab toolbox

Figure 3.30: Membership functions for the fuzzy output water valve in Matlab toolbox

Figure 3.31: Formation of rules in the Matlab toolbox of water level FLC

Figure 3.32: The rule viewer and the surface viewer of water level FLC

Figure 3.33: Simulink model of the water level FLC

Figure 3.34: Fuzzy algorithm for the water level FLC

Figure 3.35: Membership functions for the fuzzy input water temperature in Matlab toolbox

Figure 3.36: Membership functions for the fuzzy output coil in Matlab toolbox

Figure 3.37: Formation of rules in the Matlab toolbox of water temperature FLC

Figure 3.38: The rule viewer and the surface viewer of water temperature FLC

Figure 3.39: Simulink model of the water temperature FLC

Figure 3.40: Fuzzy algorithm for the water temperature FLC

Figure 3.41: Membership functions for the fuzzy input pH value in Matlab toolbox

Figure 3.42: Membership functions for the fuzzy output Albert solution valve in Matlab toolbox

Figure 3.43: Formation of rules in the Matlab toolbox of pH level FLC

Figure 3.44: The rule viewer and the surface viewer of pH level FLC

Figure 3.45: Simulink model of the pH level FLC

Figure 3.46: Fuzzy algorithm for the pH level FLC

Figure 3.47: Photos taken during the fabrication of the product

Figure 3.48: The Final Product

List of Tables

Table 1.1: Comparison between hydroponic cultivations and soil cultivations.

Table 2.1: Input linguistic variables of “Temperature Control using Fuzzy Logic” project

Table 2.2: Fuzzy rules “Temperature Control using Fuzzy Logic” project

Table 3.1: Fuzzy sets for the input temperature

Table 3.2: Fuzzy sets for the input light intensity

Table 3.3: Fuzzy sets for the output fan speed

Table 3.4: Fuzzy sets of the input variable humidity

Table 3.5: Fuzzy sets for the output Humidity valve

Table 3.6: Fuzzy sets for the output Humidity fan

Table 3.7: Fuzzy sets of the input variable CO2 air state

Table 3.8: Fuzzy sets for the output air pump

Table 3.9: Fuzzy sets of the input variable water level

Table 3.10: Fuzzy sets for the output water valve

Table 3.11: Fuzzy sets of the input variable water temperature

Table 3.12: Fuzzy sets for the output coil

Table 3.13: Fuzzy sets of the input variable pH value

Table 3.14: Fuzzy sets for the output albert solution valve

Chapter 1

INTRODUCTION

Farming or gardening isn't limited to a particular region anymore. Many of the fresh fruits and vegetables we buy in the market are grown hydroponically, in hydroponic farms all over the world during different climatic seasons. Hydroponics is a method of growing plants without the use of soil as a medium. Instead of soil, a nutrient-rich water solution is used as the medium and the plant roots are submerged in it. Plants absorb the necessary nutrients by standing with their roots in the nutrient-rich solution. Plants mainly need food, water, and air for their growth. It has been proven that plant roots can absorb nutrients from a nutrient-rich solution much effectively than absorbing it from a soil medium because, in hydroponic cultivations, all the food and water plants need are given directly to their roots around the clock. [1, 2]

They are many types of hydroponic cultivations such as deep water culture, wick system, nutrient film technique, ebb and flow, drip hydroponics, aeroponics, etc. The most suitable and common plants to grow in a hydroponic system are lettuce, spinach, bell peppers, herbs, and strawberries. It has been proven that these plants are durable in hydroponic environments. [3, 4, 5]

When deciding between hydroponic cultivations and soil cultivation s, there are many factors that we can consider to understand the worth of using hydroponics. It's only we can decide which way to choose by considering those factors. The differences between the two systems should lead you to the right option. [6]

Hydroponics vs Soil?

Table 1.1: Comparison between hydroponic cultivations and soil cultivations.

Hydroponic Cultivations	**Soil Cultivations**
There is no land usage. So there is no risk of reduction of land suitable for agriculture.	There is huge risk in depletion of land using for agriculture. And this has resulted in wild habitats turning into agricultural lands.
Very less amount of water usage. The plant's roots will only take up the sufficient amount of water needed at any one time and leave the rest in the water tank for later use.	More water is required and that has caused a risk of water scarcity throughout the globe.
Possibly produces higher and quicker yields.	Less direct access to essential nutrients. Therefore possibly lower and slow yields.
Hydroponics Allows for year-round crop growth for any climatic season.	Limited to climatic changes.
Low risk of plant diseases.	High risk of various kinds of plants and soil diseases.
Less chemicals and no weeding required. So we can get healthy and natural yields.	Chemical fertilizers are used in a big manner. So the yields are not so healthy to consume.
Time savings.	Consumes a lot of time.

Table 1.1 above shows the summary of the comparison between the hydroponic cultivation and soil cultivation. [6]

Figure 1.1 below shows the comparison of average height each day of tomato plants grown in hydroponic environment and in conventional method. [7]

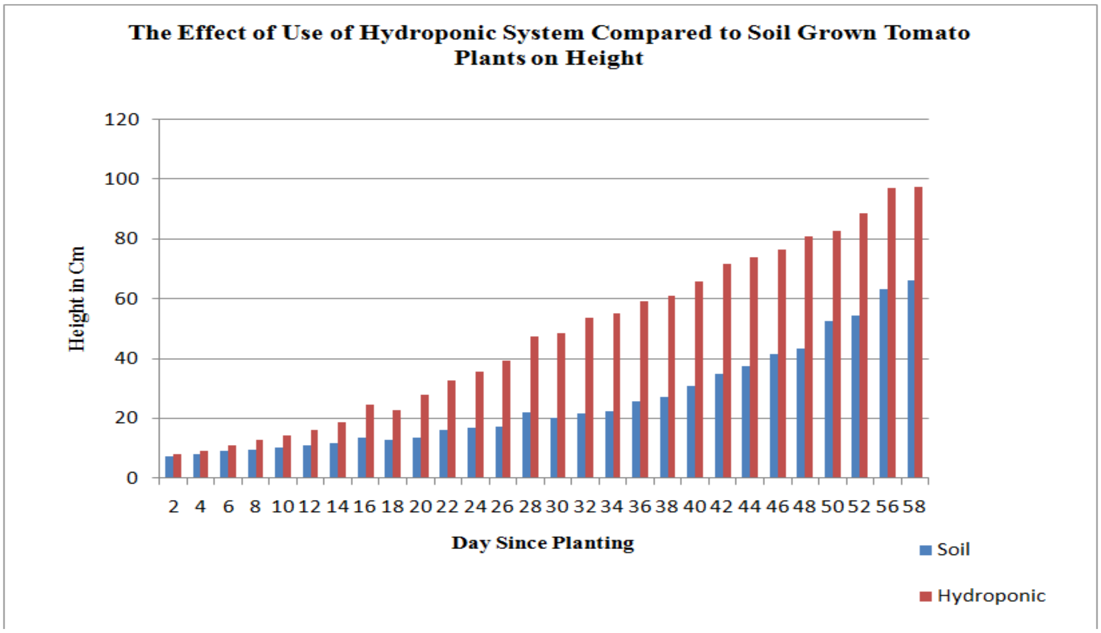

Figure 1.1: Effect of use of hydroponic system compared to soil grown tomato plants on height.

Rise in hydroponics

Hydroponics have experienced a huge rise in the last 5-6 years. Farmers are considering hydroponic cultivations over conventional cultivations.

Figure 1.2 below shows the expected global hydroponic system market by 2027 according to a research done by data bridge market. [8]

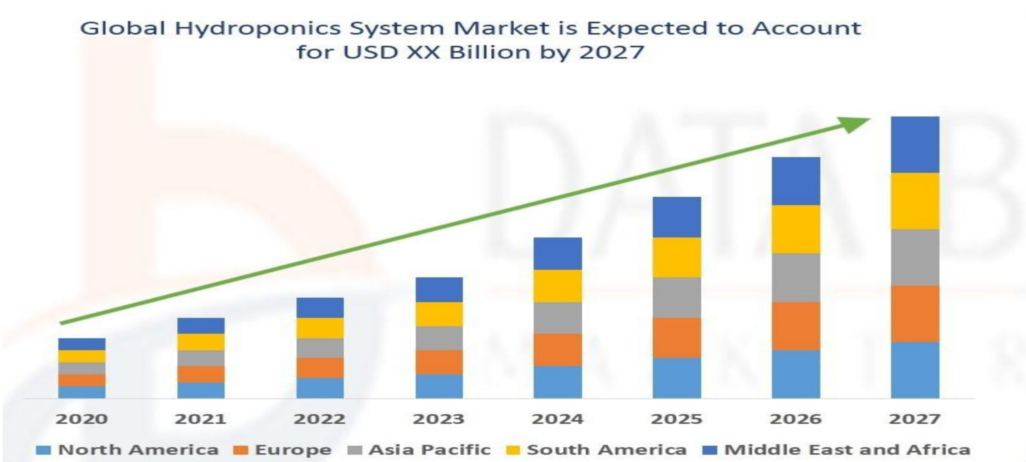

Figure 1.2: Expected global hydroponic system market by 2027

Introduction to fuzzy logic

Fuzzy logic is an extension of Boolean logic by Lotfi Zadeh in 1965. It was based on the mathematical theory of fuzzy sets which is somewhat similar to the concept of classical crisp set theory. Unlike classical crisp sets, fuzzy sets allow a partial number to a set from 0 to 1. Fuzzy logic enables a condition to be in a state other than true or false. It gives the luxury of the notion of degree in the verification of a condition. This makes fuzzy logic more flexible when compared to Boolean logic. Due to this reason, fuzzy logic has been used in many complex systems to make them simpler. [10]

Fuzzy logic has rapidly turn out to be one of the most successful technologies for developing sophisticated control systems. A common fuzzy system consists of four main parts such as a fuzzification, a knowledge base, an inference engine, and a defuzzification. Fuzzification is the process of converting a crisp input into a fuzzy input according to the membership functions or the fuzzy sets. The knowledge base consists of the rule base of the fuzzy system. So the knowledge base can be called the key element of a fuzzy system as it contains all the necessary information to execute a task. The inference engine provides the decision-making logic of the controller. Defuzzification is the process of finally converting the set of output values from the controller into a single point wise value. [11, 12, 17]

The figure 1.3 below shows the basic configuration of a fuzzy system.

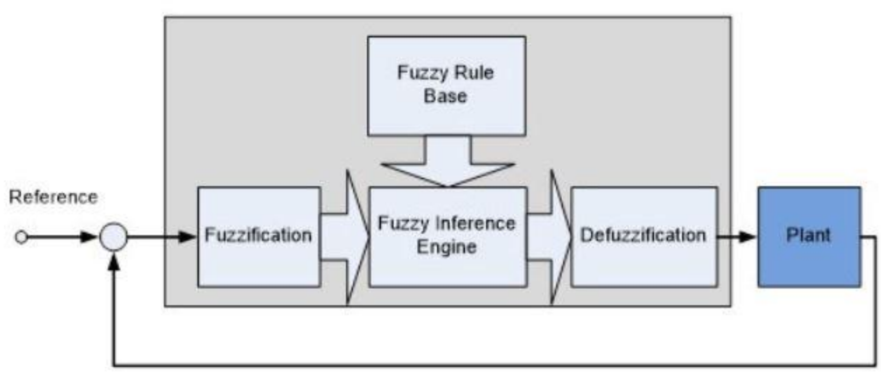

Figure 1.3: The basic configuration of a fuzzy system

1.1 Research Objectives

The foremost objective of the proposed work is to develop and modify the product that we already implemented in the first semester, using new technologies, making it more efficient and user-friendly. To address this, the thesis focuses on the following specific objectives:

- To build a new table top structure to our product which can be in indoors without consuming much space.
- To study and get a clear understanding of what should be the exact range of values for each parameter to maximize plant growth in a hydroponic environment.
- To investigate and find out more essential parameters needed for plant growth in addition to the parameters controlled before.
- To get familiar with matlab fuzzy Simulink software.
- To study and examine about the fuzzy logic algorithms.
- To build fuzzy logic algorithm systems for control and monitor each parameter instead of the relays.
- To investigate and study more deeply about web page development and app development.
- To configure systems to monitor and control the water temperature level of the nutrient rich water solution and the carbon dioxide level of the environment.
- To develop skills such as time management, team work, project development, problem solving, adaptability, leadership and etc.
- To add additional mechanisms to control the parameters which are measured up to now.
- To solar power final our product.

- Complete the Automated Hydroponic Cultivation web application.

- To make the web interface more user friendly.

- To build a final quality product which can be introduced to the market.

1.2 Contribution of the Thesis

The thesis provides the following contributions:

- A fully automated hydroponic cultivation system which will monitor and control all the necessary parameters needed for plant growth in an adequate manner.

- A final quality product that is very user friendly and can be kept indoors without consuming much space.

- Fully automated systems to control each parameter using fuzzy logic algorithms.

- A web application which stores and displays the state of the automated hydroponic cultivation.

- Additional mechanisms are implemented to control the parameters which are measured up to now to improve the efficiency of the system.

1.3 Structure of the Thesis

Below is a short introduction about the structure of thesis.

- **Chapter 2:** The background readings and literature reviews about automated hydroponic cultivation projects, which are studied to carry out this project are represented in this chapter. This chapter also includes an overview of literature reviews about the projects implemented using fuzzy logic algorithms.

- **Chapter 3:** This chapter includes a broad summary of the software implementation and the hardware implementation of the proposed product. The technologies and the theories used in implementing the product is also discussed.

- **Chapter 4:** In this chapter, the results of our software and hardware implementations are discussed and analyzed. This chapter also addresses the challenges faced during the project and the limitations in the proposed work.

- **Chapter 5**: This chapter presents the final conclusion of our project.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the already existing projects about automated hydroponic cultivations and projects which have been done using fuzzy logic algorithms. The software and hardware implementation techniques and theories used in those projects are discussed shortly.

2.2 Background and Motivation

Hydroponic systems have attracted increased attention from the farmers currently as it provides an alternative agricultural method to conventional soil cultivation of crops. The word hydroponics comes from the roots “hydro”, meaning water, and “ponos”, meaning labor. The earliest reference to hydroponics was an idea put up by a man called William Frederick Gericke. He has come up with the idea that plants could be grown in a solution of nutrients instead of soil. The history of hydroponic cultivations dates back to the Floating Gardens of China and the Hanging Gardens of Babylon. So it is clear that the ancient people have used these techniques thousands of years ago. The general theory behind hydroponic cultivations still remains the same although there have been few modifications made using modern technology. [13, 14]

Hydroponic environments have solved many conventional agricultural problems such as water scarcity, weeding problems, pest attacks and diseases, labor costs and etc. In hydroponic environments, artificial technologies are used to provide the appropriate environmental factors for the crops. Environmental factors such as temperature, humidity, light intensity and etc must be managed using manual technologies in hydroponic cultivations. So a manual hydroponic system needs some attention and regular maintenance. So the demand for automated hydroponic cultivation systems has evolved. [15]

The main goal of our design is to create a fully automated, table-top hydroponic cultivation system. A successful product would provide farmers with the ability to grow any crops indoors whatever the climate or season is. We used the Nutrient Film Technique (NFT) hydroponic cultivation type for our product because it looks more efficient and it is the leading hydroponic cultivation method at present. Nutrient Film Technique method is popular among both commercial and home hydroponic farmers due to many reasons. In our product, the nutrient water is constantly flowing through to the upper tray which contains the plants from the lower reservoir. The upper tray is an enclosed channel where crops sit in small openings made in it. Then there is an overflow drain to get the excess water back to the lower reservoir, ready to be used again. [14, 16]

Figure 2.1 below shows the structure of the Nutrient Film Technique hydroponic cultivation method.

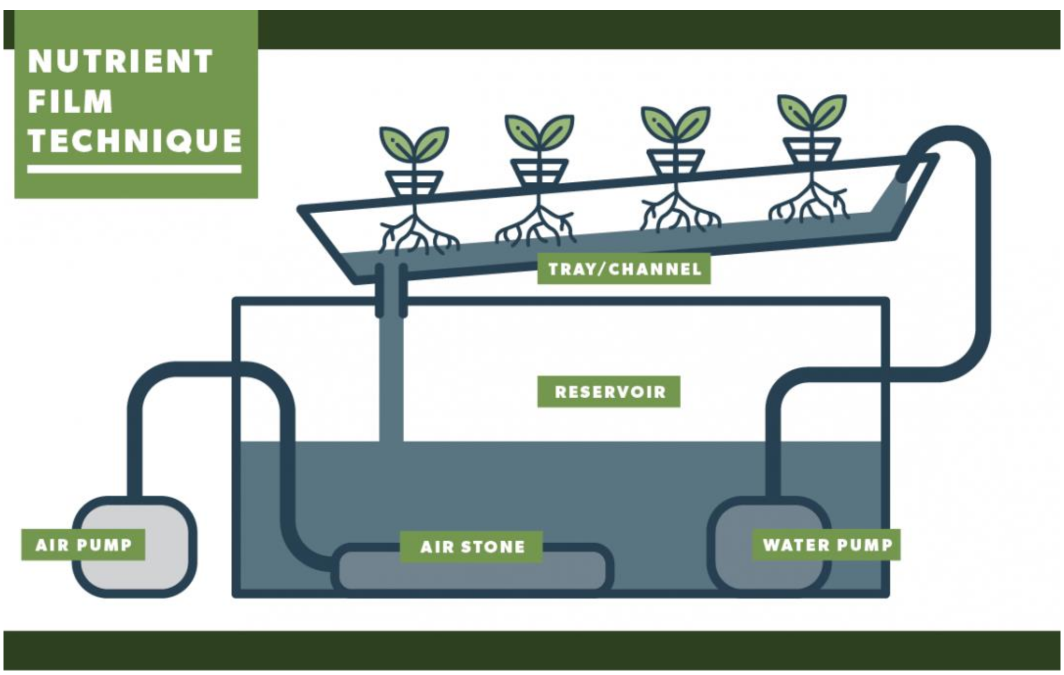

Figure 2.1: Structure of the Nutrient Film Technique hydroponic cultivation method

2.3 Similar Existing Projects

2.3.1 Temperature Control using Fuzzy Logic

By P. Singhala1 , D. N. Shah2 , B. Patel

This project aims to heat the system to the required temperature, afterward hold it at that temperature in an insured manner. The product consists of a heater, fan, and temperature sensor. In this product, the set point of the temperature is given into the system by an operator using a 4x4 keypad. The temperature is measured using the LM35 temperature sensor. There is an analog to digital converter to convert the analog value from the sensor into a digital value and give it to the fuzzy controller. The input of the fuzzy controller is the difference between the setpoint value and the current temperature measured by the sensor. This input value is called the error. The amount of power delivered to the heater and the fan is controlled by passing a command through a serial port via the microcontroller. The microcontroller generates a PWM signal for the MOSFET to deliver the desired amount of power to the fan and the heater which is needed to keep the temperature stable. [18]

Table 2.1 below shows the input linguistic variables of the fuzzy controller

.

Table 2.1: Input linguistic variables of “Temperature Control using Fuzzy Logic” project

No.	Crisp Input Range (Error = Set Point – Current Temperature)	Fuzzy Variable Name
1	-15 to -50	NEG
2	0 to +30	SNEG
3	-15 to +15	ZERO
4	0 to +30	SPOZ
5	+15 to +50	POZ

Table 2.2 below shows the fuzzy rules with the relevant output linguistic variables.

Table 2.2: Fuzzy rules "Temperature Control using Fuzzy Logic" project

No	Fuzzy variable range output	Corresponding	Fuzzy variable name
1	165.75 to 255	65% to 100%	VH
2	127 to 204	50% to 80%	H
3	165.75 to 89.25	65% To 35%	M
4	127 to 51	50% to 20%	L
5	89.25 to 0	35% to 0%	Z

This product can be called a fuzzy control system using a small number of rules and a simple implementation to solve a temperature control problem.

2.3.2 Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse

By Rim BEN ALI, Emna ARIDHI, Mehdi ABBES, Abdelkader MAMI

In this system, a fuzzy logic controller is developed to control and monitor the inside temperature and relative humidity inside an agricultural greenhouse according to the dynamics of input and output variables inside the greenhouse environment using membership functions. At first, a thermal dynamic model has been developed to relate the greenhouse environment and to predict the inside temperature and relative humidity. This system was modeled using Matlab-Simulink. [19]

The fuzzy controller temperature system has only one input which is the temperature error.

$$\Delta T = T_{setpoint} - T_{inside\ temperature}$$

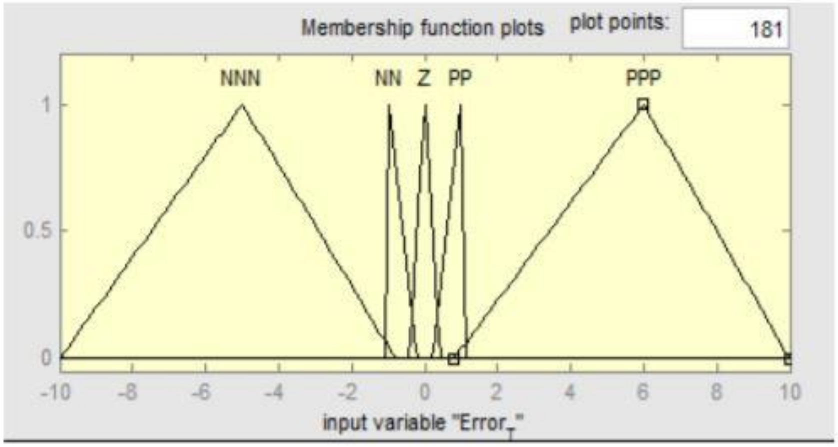

Figure 2.2: The membership function of the input variable of the project “Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse”

Figure 2.2 above shows the membership function of the only input variable, the temperature error plotted in matlab.

There are two output variables, which are the ventilation rate and the heating rate. The membership functions of the output variables are shown in the below figure 2.3.

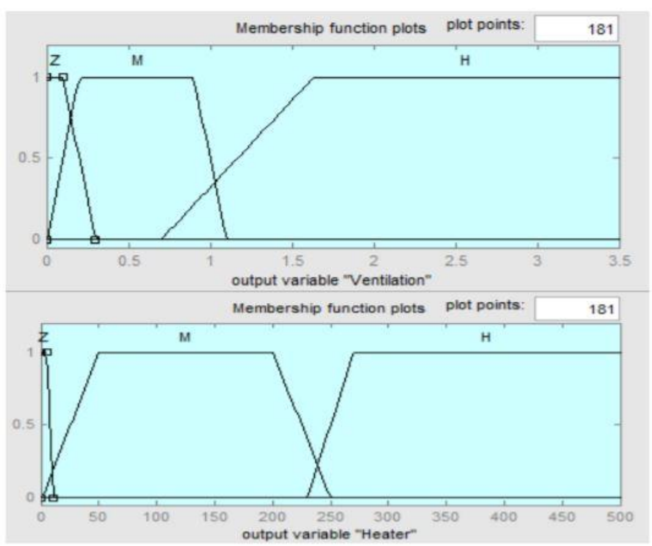

Figure 2.3: The membership functions of the output variables of the project “Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse”

The input and output variables have been similarly initialized for the relative humidity FLC also. Separate fuzzy rule bases have been made to control the two parameters.

Figure 2.4 below shows the dynamic model of the agricultural greenhouse modeled using Matlab-Simulink environment.

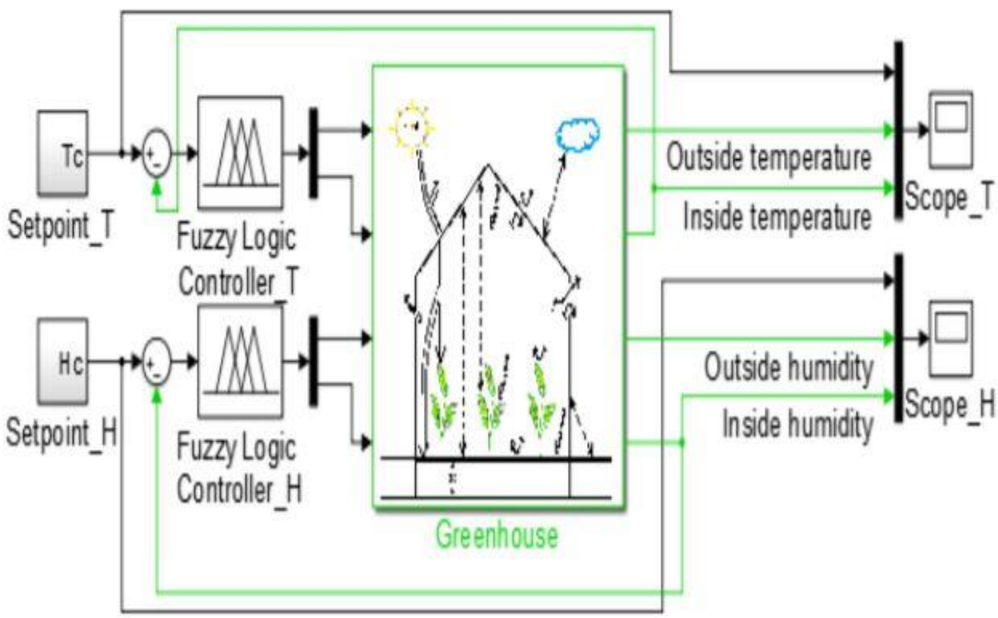

Figure 2.4: dynamic model of the agricultural greenhouse modeled using Matlab-Simulink environment of the project "Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse"

In this project, a simplified dynamic model was developed to control the inside climate behavior of an agricultural greenhouse. The FLC controller was modeled using Matlab-Simulink.

2.3.3 Fully Automated Hydroponic System for Indoor Plant Growth

By Vaibhav Palandea , Adam Zaheera , and Kiran George

The main goal of this project has been to overcome the problem of growing certain plants and vegetables in remote areas such as deserts and the north and south poles due to the extreme outside weather. In this project, the hydroponic system is monitored using the Internet of Things (IoT) network. The user should only plant a seed and input the relevant parameters. This system uses two Arduino boards for the analysis of the received data and control and a Raspberry Pi is used to run an open-source automation software called Domoticz. The system also uses a mobile app for the remote monitoring process which supports both ios and android devices. The system consists of four sensors such as an electrical conductivity probe, a pH sensor, a water temperature sensor, and an air temperature/humidity sensor. Water is heated up to around 25°C using a water heater, which is optimum for healthy root growth. [20, 21]

Figure 2.5 below shows the web interface of the system.

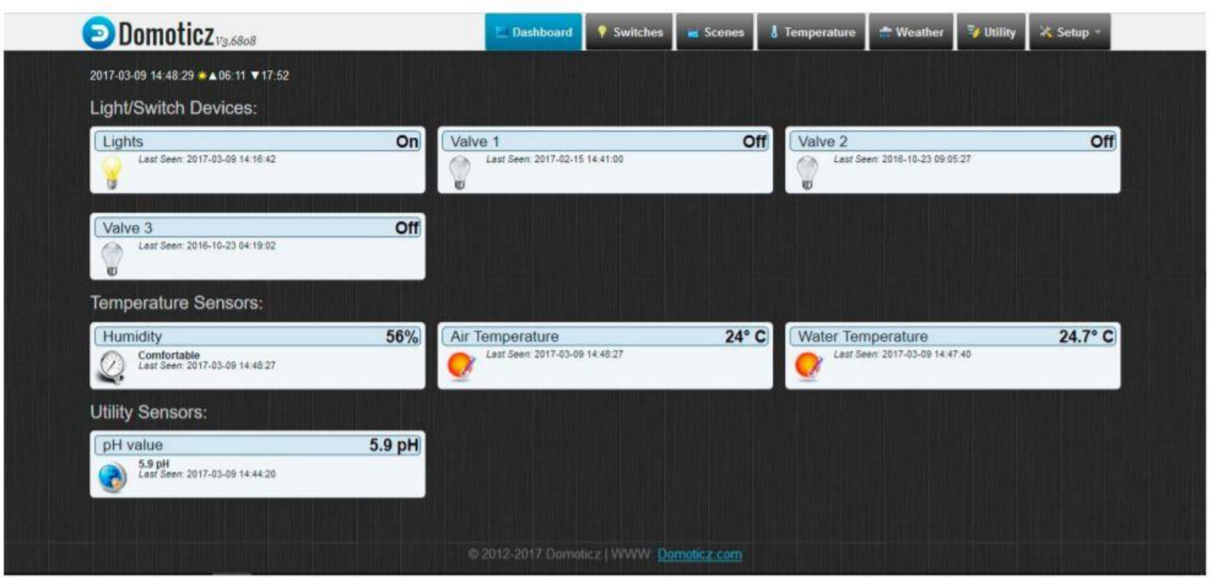

Figure 2.5: Web interface of the project “Fully Automated Hydroponic System for Indoor Plant Growth”

This system highlights its importance through its complete automation aspect and its small size.

2.4 Summary

This chapter discussed the background information for our project and some similar approaches to our product. It has shortly addressed the methodologies and theories used by those projects. Chapter 3 will be addressing the hardware implementation and the software implementation of our approach and the methodologies and techniques used.

Chapter 3

METHODOLOGY OF OUR WORK

3.1 Introduction

This chapter includes a comprehensive overview of the software implementation and the hardware implementation of the project and the methodologies and principal techniques used in the project are also discussed.

3.2 Overview of the proposed work

The proposed work is to develop and modify the IoT-based solar-powered automated hydroponic cultivation system using new technologies, which we implemented in the first semester. In the first semester, we controlled five parameters that are essential for plant growth. They are the temperature of the air, the humidity of the air, light intensity, the water level of the nutrient-rich water solution, and the pH value of the water solution.

In this semester, we implemented two systems that control another two essential parameters for plant growth in addition to the parameters controlled in the first semester. The water temperature of the water solution and the carbon dioxide level of the environment are those two parameters. Water temperature is a very important factor since the roots are submerged in the water solution. The temperature of the water should be kept in the perfect range because it helps the roots to absorb nutrients more effectively. The carbon dioxide level of the environment is also a key factor in plant growth because CO_2 is needed for the photosynthesis process of plants. So the hydroponic environment must supply the adequate amount of CO_2 needed for the photosynthesis process. We also implemented fuzzy logic controllers to all the parameter controlling units instead of the relays to make the parameter controlling process more efficient. In addition to those, we implemented a disease-identifying system and an algae-controlling system to our product.

Figure 3.1 below shows the basic overview diagram of our proposed work. The DS18B20 sensor and the mq135 sensor was used to monitor the water temperature and the CO2 level of air respectively.

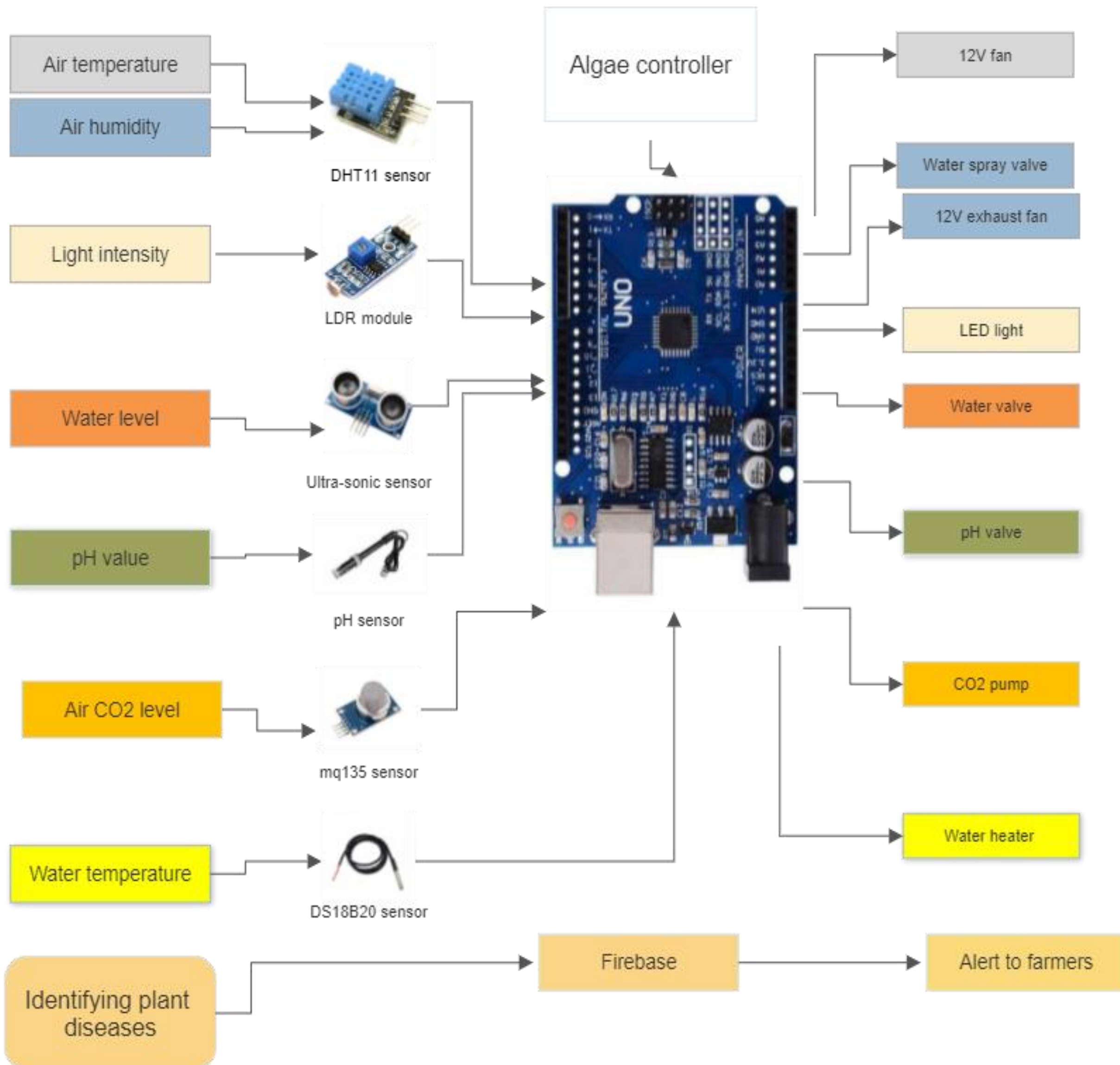

Figure 3.1: Basic overview diagram of the proposed project

3.3 Individual Contribution

My part was to design fuzzy logic controllers for all the parameter controlling systems. And I implemented the systems for controlling water temperature and controlling the CO2 level of air.

3.3.1 Implementation of the water temperature and the CO2 level monitoring systems

Water temperature monitoring system

The DS18B20 temperature sensor was used to measure the temperature of the nutrient water solution. It's a one-wire digital temperature sensor from maxim integrated. It gives a digital output as a reading. DS18B20 is widely used to measure temperatures in hard environments such as chemical solutions, mines, soil, etc due to its user-friendly characteristics such as small size, high accuracy, low hardware overhead, etc. Each DSB18B20 sensor has a unique serial number. This allows for measuring the temperature at multiple points without using much of the digital pins in the microcontroller. The DS18B20 sensor works with the mechanism of one-wire communication. Only the data pin should be connected to the microcontroller. The other two pins are used to power the sensor. A pull-up resistor must be connected in parallel between the data pin and the VCC pin. [21, 22, 23]

Figure 3.2: DS18B20 temperature sensor

Figure 3.2 above shows the DS18B20 sensor and its three pins.

Some features of the DS18B20 sensor are:

- Has the ability to measure degrees in Celsius with 9 to 12 bit precision, from -55 to +125 Celsius.
- Power supply range from 3V to 5.5V.
- ±0.5°C Accuracy from -10°C to +85°C.
- No external components are required.

Figure 3.3 below shows the basic circuit for measuring water temperature.

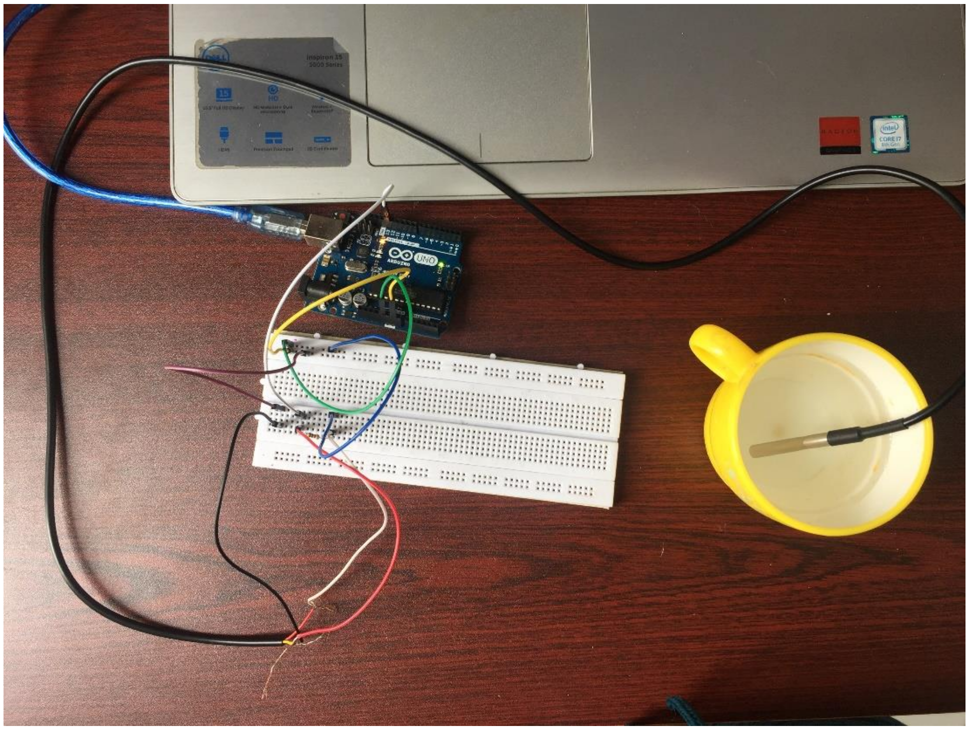

Figure 3.3: Basic circuit for measuring water temperature.

CO_2 level monitoring system

The mq135 is a gas sensor that can be used for measuring a wide range of gases such as CO_2, NH_3, NO_x, alcohol, smoke, etc. It is widely used in air quality monitoring systems in various industries. The mq135 gas sensor is very popular because of its low cost. The mq135 sensor uses a small heater inside with an electrochemical sensor to measure the various kinds of gases. The mq135 sensor consists of four pins such as Vcc, GND, an analog pin, and a digital pin. [24, 25]

Figure 3.4 below shows the mq135 gas sensor.

Figure 3.4: mq135 gas sensor

Some features of the mq135 sensor are:

- Long life and low cost.
- High sensitivity to ammonia, benzene and etc.
- Simple structure and a circuit.
- High accuracy
- Wide detection range
- Operates at 5V and consumes 150mA.

Figure 3.5 below shows the basic structure for CO2 level monitoring system.

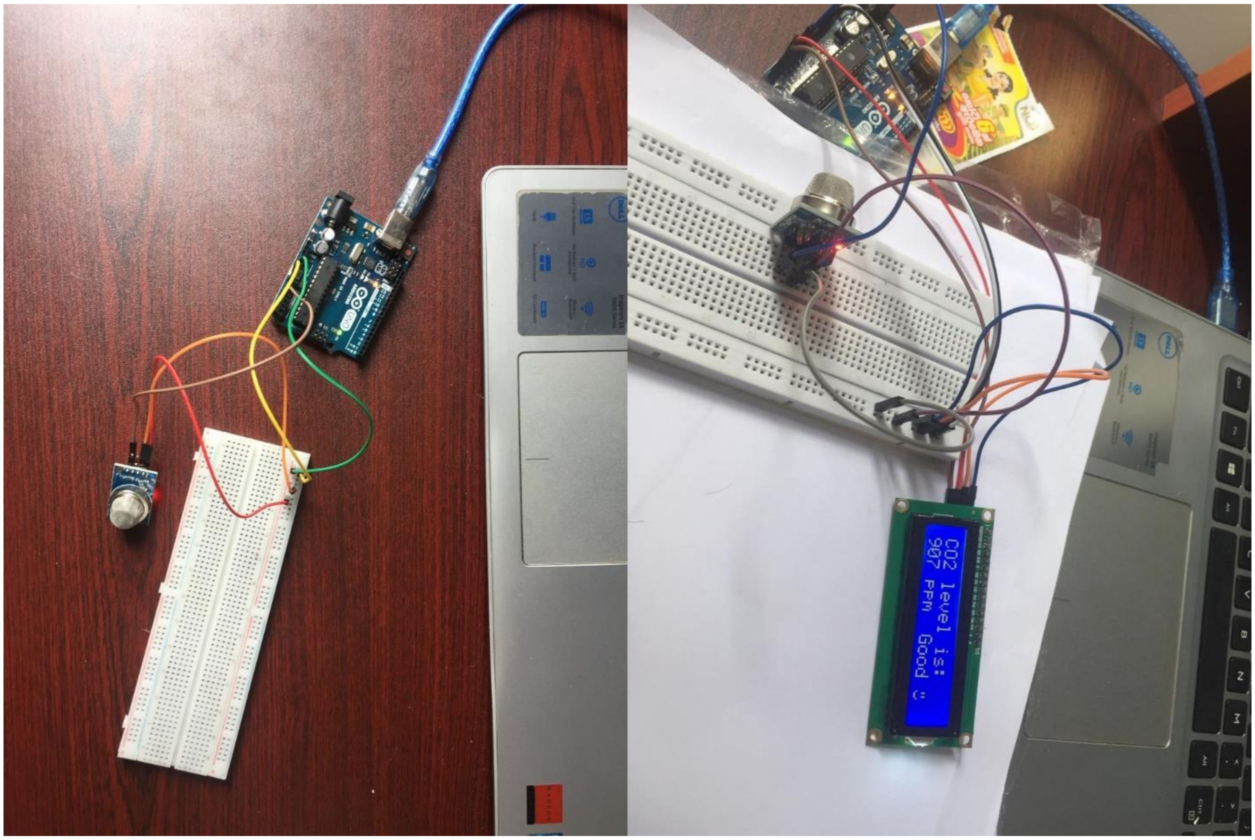

Figure 3.5: Basic circuit for measuring the CO2 level

3.3.2 Designing Fuzzy Logic Controllers

Fuzzy logic seems closer to the way how the human brains think. So fuzzy logic controllers help programming machines to think as humans do. In this project, I implemented six fuzzy logic controller systems to control and monitor six different parameters needed for plant growth. Those parameters are air temperature, air humidity, pH value of the nutrient solution, water level of the solution, CO2 level of air, the Water temperature of the solution. [22]

At first, I investigated and got an idea of what should be the perfect range for each parameter that we should maintain to get a maximum effect on the plant growth in a hydroponic environment. According to those values I created fuzzy sets for each parameter. Then I created membership functions for the input variables and the output variables in the Matlab fuzzy toolbox. After that, the rule base was created in the Matlab fuzzy toolbox for each parameter. Then the fuzzy controller for each parameter was modeled using the Matlab Simulink using the created fuzzy toolbox file.

Fuzzy logic controller for the air temperature monitoring system

In our proposed work the air temperature was controlled using a 12V dc fan. For this fuzzy controller system, I used two input variables and one output variable. The two inputs were the temperature reading from the DHT11 sensor and the light intensity reading from the LDR module. The reason behind taking the light intensity reading as an input was because our system consists of LED lights. When those lights are on it slightly affects the temperature. That means when the lights are on the temperature can get a slight increase in value. So I thought the air temperature fuzzy controller will be more effective when we take light intensity also as input to the system. The output of the fuzzy controller was the 12V dc fan speed.

Table 3.1 and table 3.2 below shows the fuzzy sets for the two fuzzy inputs temperature and the light intensity respectively.

Table 3.1: Fuzzy sets for the input temperature

No	Input Range	Variable name
1	0 to 5	verysmall
2	0 to 10	small
3	5 to 15	mid
4	10 to 30	big
5	20 to 30	verybig

Table 3.2: Fuzzy sets for the input light intensity

No	Input Range	Variable name
1	0 to 600	lowldr
2	300 to 900	midldr
3	900 to 1500	highldr

Figure 3.6 and figure 3.7 below shows the membership functions created in the Matlab fuzzy toolbox according to the input values for the two fuzzy inputs temperature and the light intensity respectively.

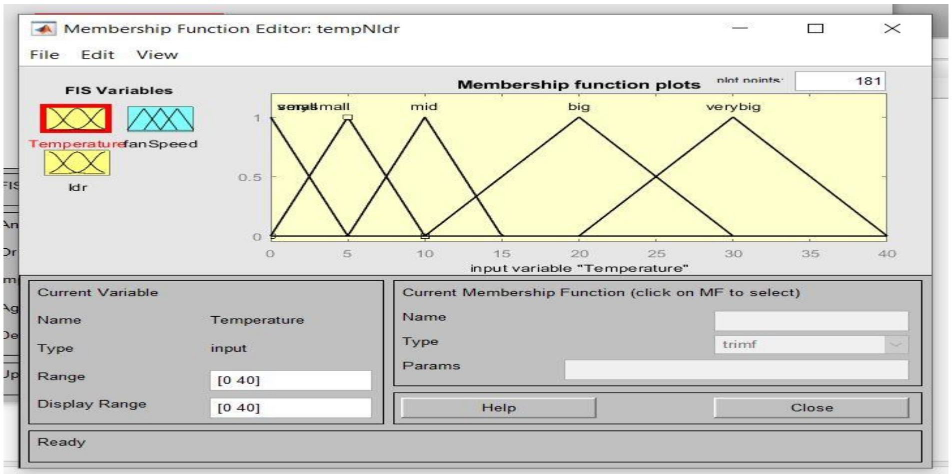

Figure 3.6: Membership functions for the fuzzy input temperature in Matlab toolbox

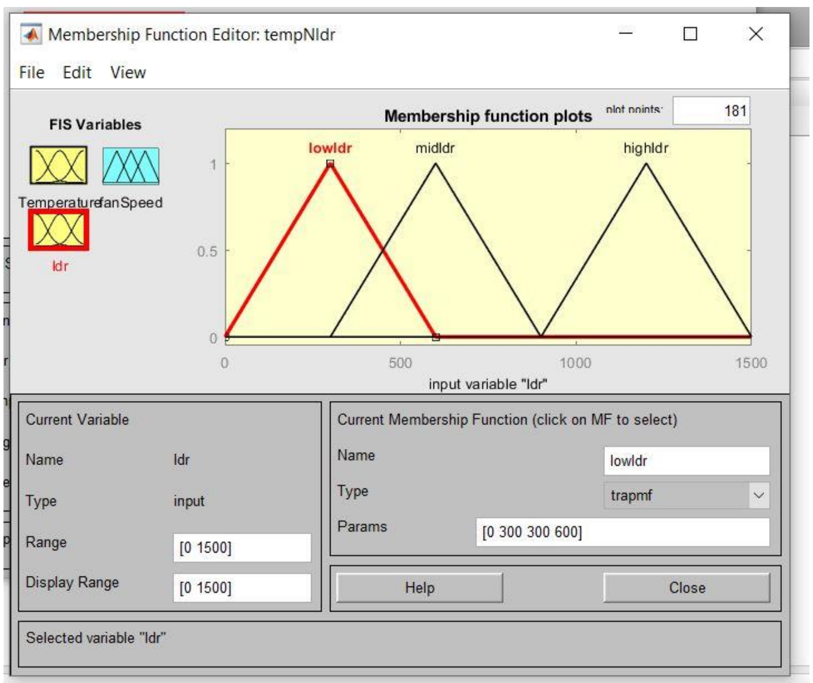

Figure 3.7: Membership functions for the fuzzy input light intensity in Matlab toolbox

Then the fuzzy sets for the output variable fan speed was created. Table 3.3 below shows the fuzzy sets for the output variable fan speed.

Table 3.3: Fuzzy sets for the output fan speed

No	Output range	Variable name
1	0 to 100	lowb
2	50 to 150	midb
3	100 to 200	highb
4	150 to 255	veryhighb

Then the membership functions was created for the output variable fan speed in the Matlab toolbox according to the fuzzy sets. Figure 3.8 below shows the membership functions created in the Matlab fuzzy toolbox for the output variable fan speed.

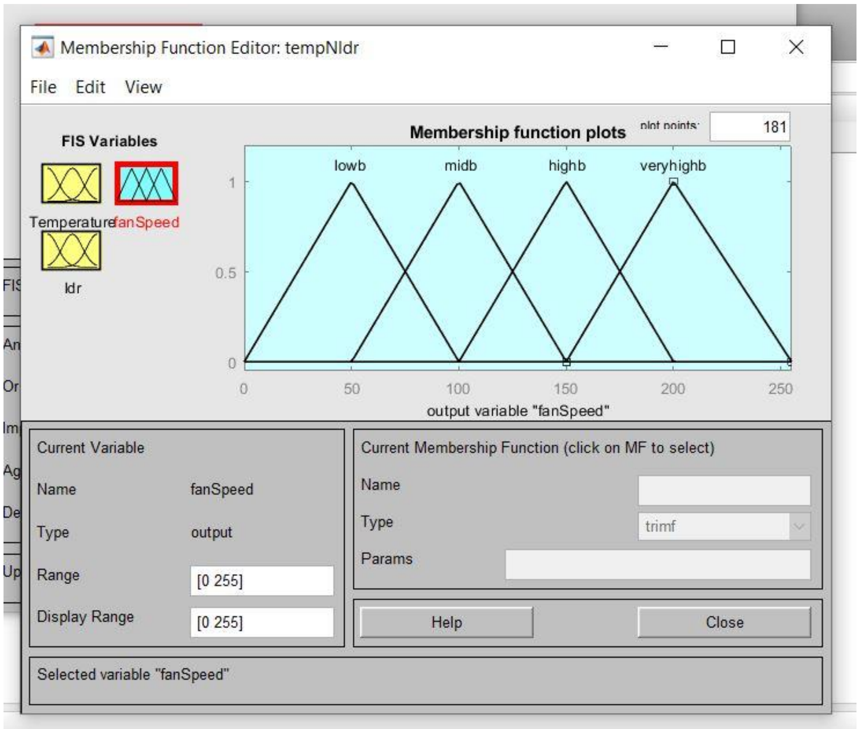

Figure 3.8: Membership functions for the fuzzy output fan speed in Matlab toolbox

Then the fuzzy rule base was created. The rule base for this fuzzy controller consists of 15 fuzzy rules.

The two figures below show the implementation of the rule base. Figure 3.9 shows the formation of the rules and figure 3.10 shows the rule viewer in which we can observe the output value by changing the input values.

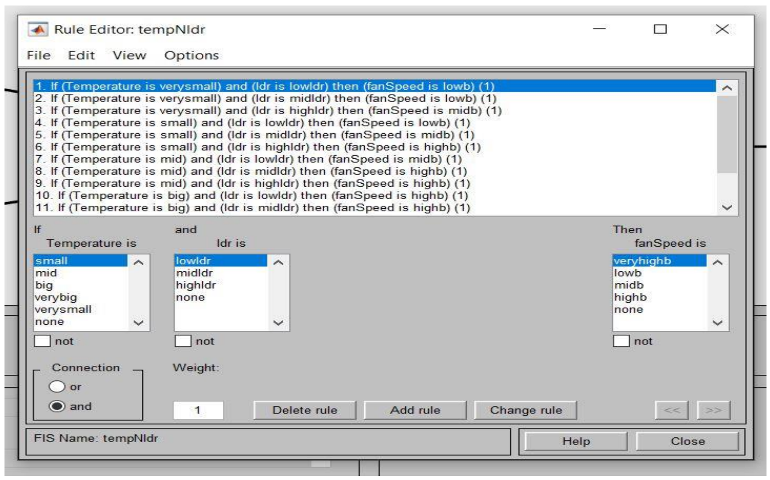

Figure 3.9: Formation of rules in the Matlab toolbox of air temperature FLC.

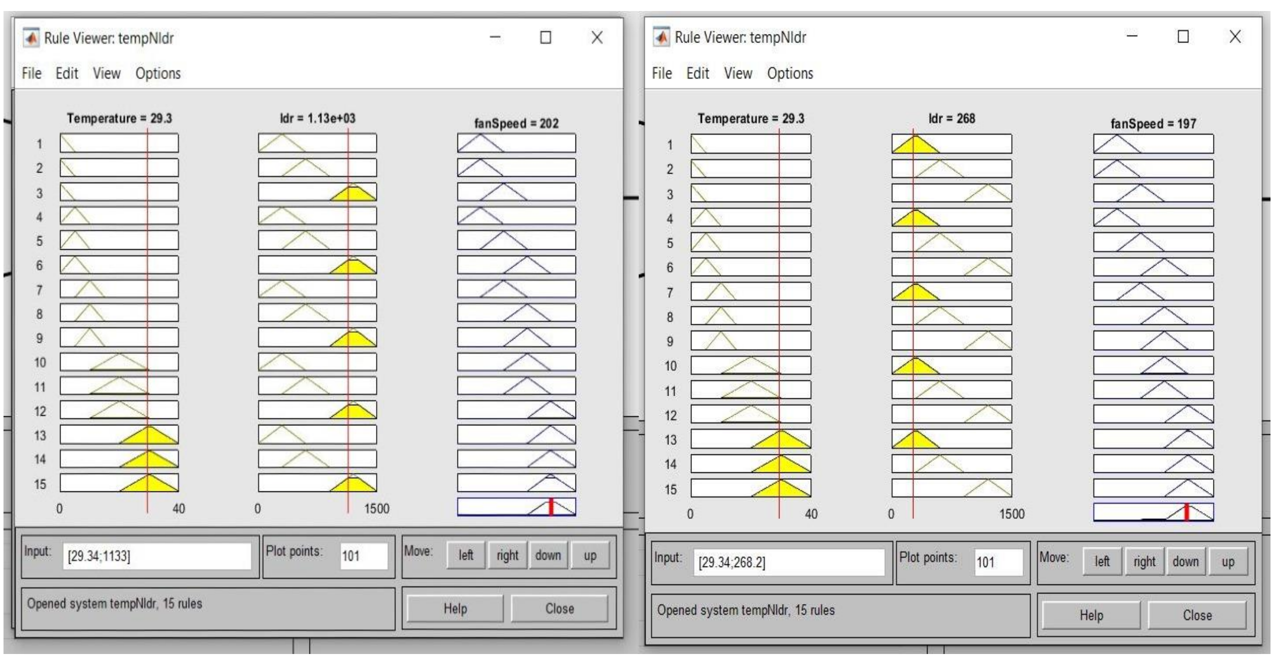

Figure 3.10: Rule viewer for air temperature FLC

In the above figure 3.10, there are two instances of the rule viewer. The input temperature is the same in both instances. In the left instance, the light intensity input reading is higher than the instance on the right side. So we can see the output (fan speed) has reduced slightly with the decrease of the light intensity input reading. In this way, we can change the inputs and observe the output in the rule viewer. By doing this, we can confirm that the rule base is implemented correctly and effectively.

Figure 3.11 below shows surface viewer of the rule base.

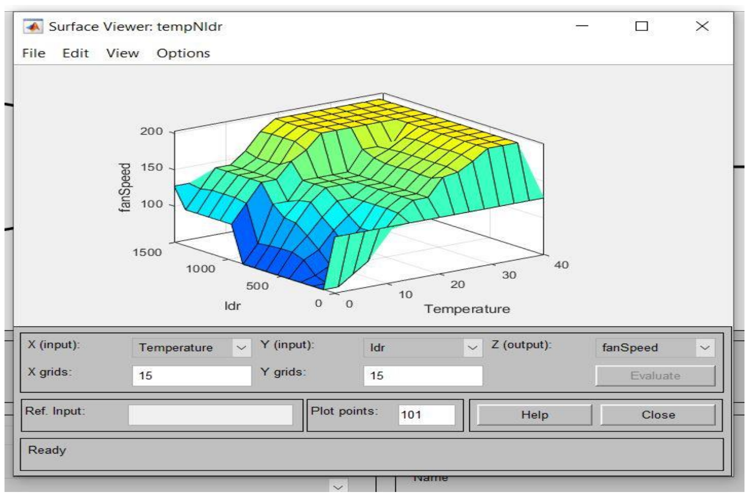

Figure 3.11: surface viewer of the rule base of air temperature FLC

Then the system was modeled using the Matlab Simulink software. Figure 3.12 below shows the Simulink model created.

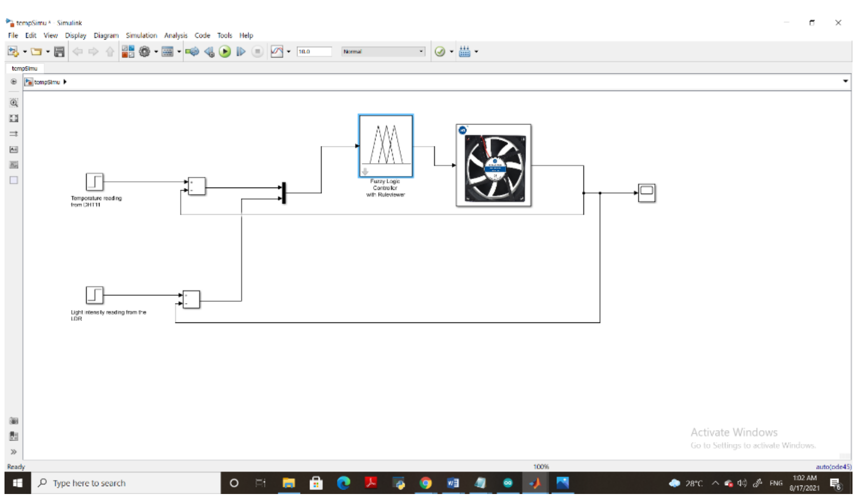

Figure 3.12: Simulink model of the air temperature FLC

Finally the fuzzy algorithm was coded in the Arduino IDE software.

Figure 3.13 below shows the fuzzy algorithm for the air temperature FLC. In this screenshot only a part of the code is visible. The full code will be in the appendix part of the thesis.

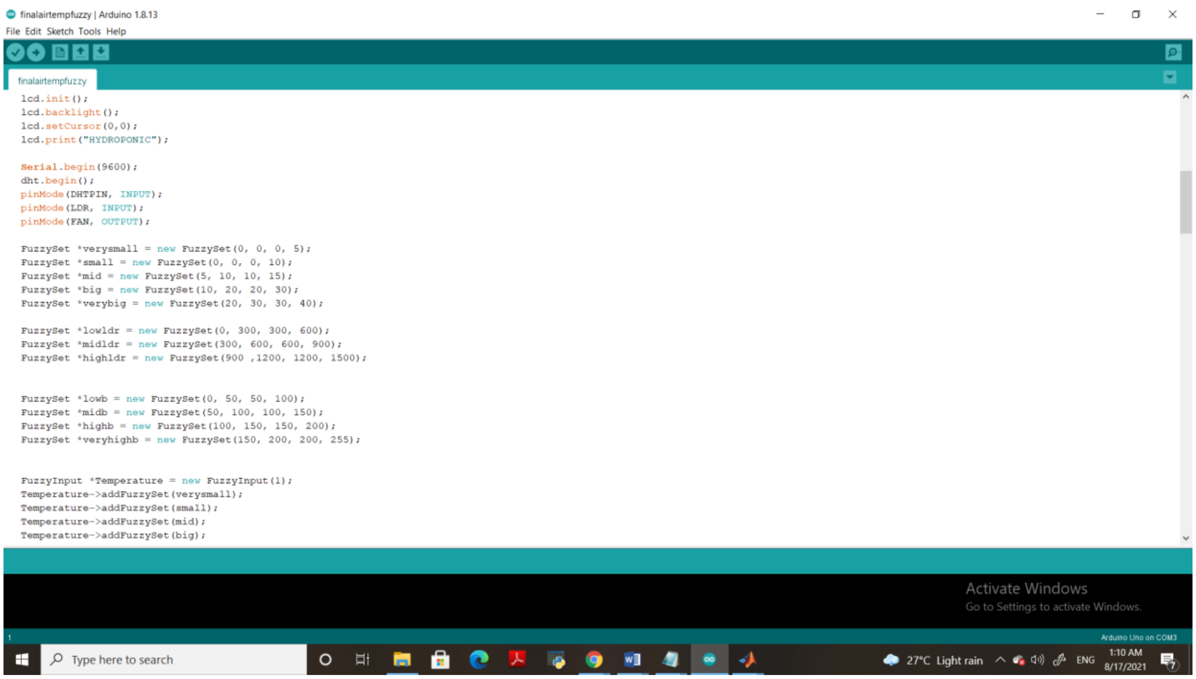

Figure 3.13: Fuzzy algorithm for the air temperature FLC

Fuzzy logic controller for the air humidity monitoring system

In our product, the air humidity is controlled through a 12V solenoid water spraying valve and a 12V exhaust fan. The 12V solenoid water spraying valve can increase humidity by adding moisture into the environment and the 12V exhaust fan can decrease humidity by removing the moisture out. In this FLC controller, the humidity reading from the DHT11 sensor is taken as the input variable. There are two output variables. They are the valve and the exhaust fan.

So firstly I created the fuzzy sets for the input variable.

Table 3.4 and figure 3.14 below show the fuzzy sets and the membership functions of the input variable humidity respectively.

Table 3.4: Fuzzy sets of the input variable humidity

No	Input Range	Variable name
1	0 to 10	verysmall5
2	0 to 40	small5
3	30 to 50	mid5
4	40 to 80	big5
5	60 to 100	verybig5

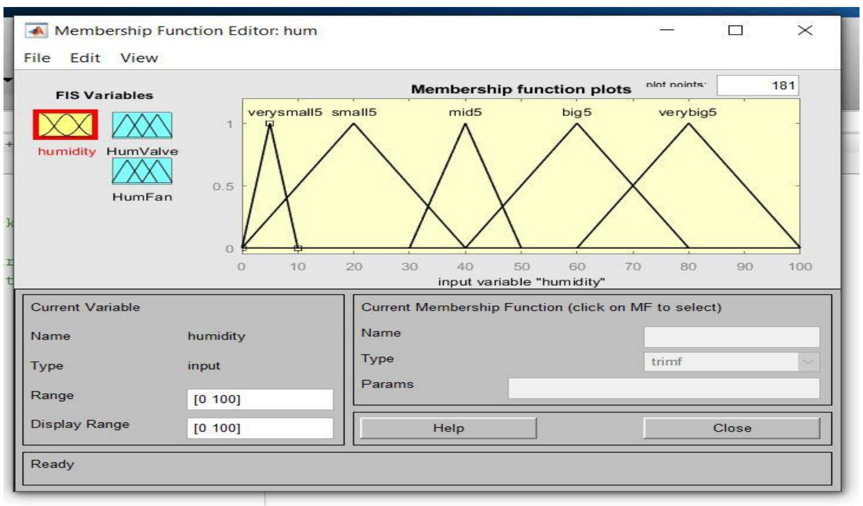

Figure 3.14: Membership functions for the fuzzy input humidity in Matlab toolbox

Then the fuzzy sets for the two output variables was created. The table 3.5 and table 3.6 below shows the fuzzy sets for the two output variables Humidity valve and the humidity fan respectively.

Table 3.5: Fuzzy sets for the output Humidity valve

No	Output range	Variable name
1	0 to 100	lowb5
2	50 to 150	midb5
3	100 to 200	highb5
4	150 to 255	veryhighb5

Table 3.6: Fuzzy sets for the output Humidity fan

No	Output range	Variable name
1	0 to 100	lowb8
2	50 to 150	midb8
3	100 to 200	highb8
4	150 to 255	veryhighb8

Then the membership functions was created in the Matlab toolbox. Figure 3.15 and figure 3.16 below shows the membership functions for the two output variables.

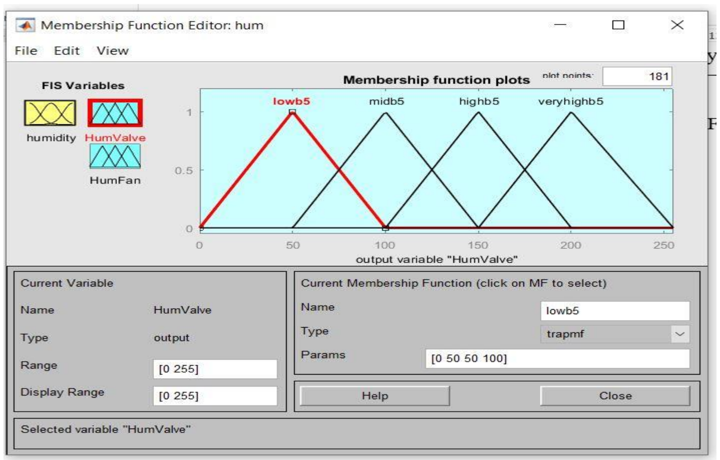

Figure 3.15: Membership functions for the fuzzy output humidity valve in Matlab toolbox

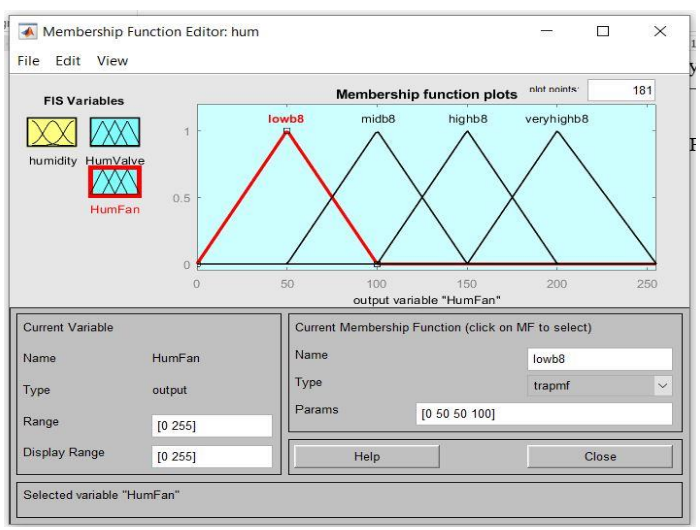

Figure 3.16: Membership functions for the fuzzy output humidity fan in Matlab toolbox

Then the fuzzy rule base was created. Figures 3.17 and 3.18 below shows the formation of rules and rule viewer of the Matlab fuzzy tool box of the air humidity FLC.

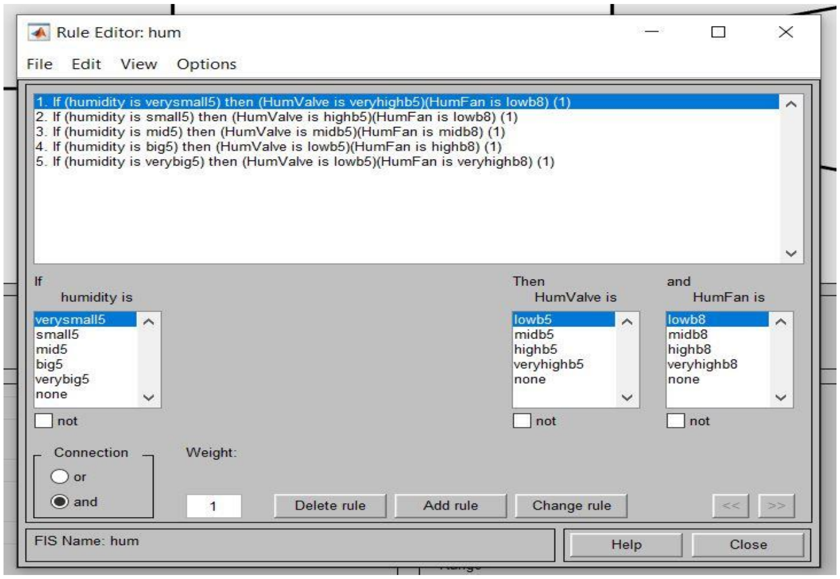

Figure 3.17: Formation of rules in the Matlab toolbox of air humidity FLC.

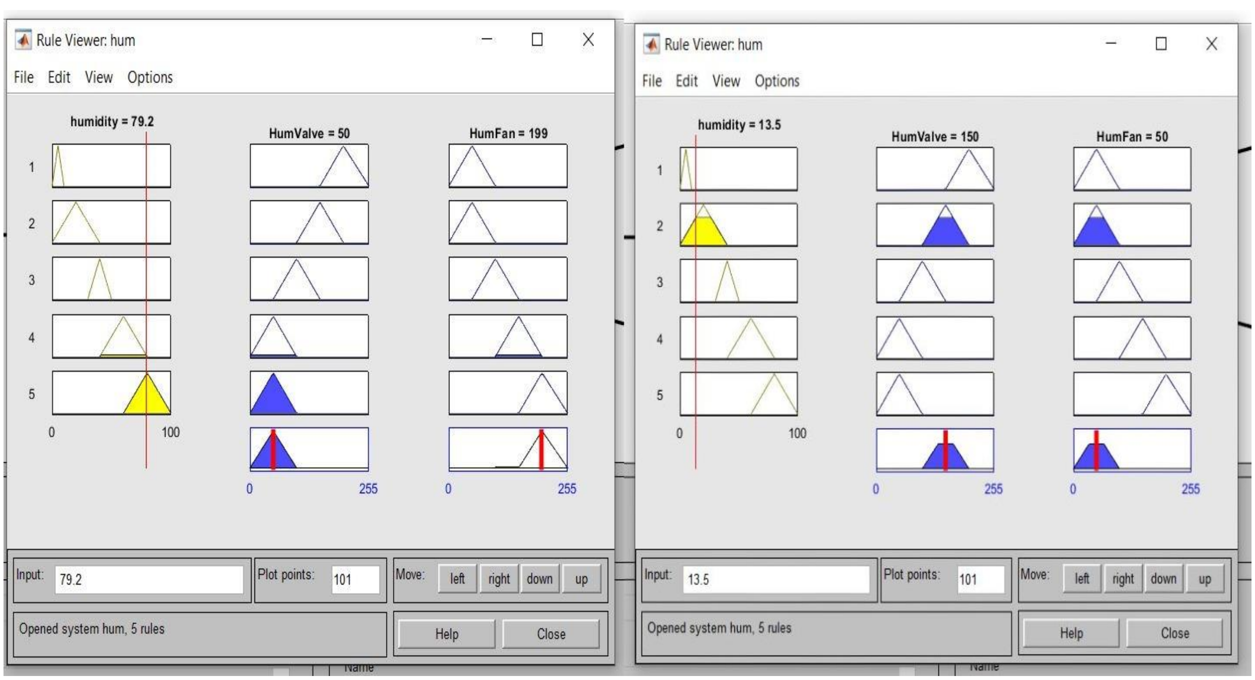

Figure 3.18: Rule viewer for air humidity FLC

The perfect air humidity range that supports plant growth is around 50-60 %. In the above figure 3.18, there are two cases in the rule viewer. In the first case or the case on the left side, the humidity reading is 79.2 which is higher than the range that we should maintain. So the system must decrease the air humidity. So we can see the humidity fan has got a higher voltage and the humidity valve has got a lower voltage. In the right side case, the air humidity is low. Therefore the system must increase the air humidity. So the humidity valve has a higher voltage while the humidity fan has a lower voltage.

Figure 3.19 below shows the surface view of the rule base.

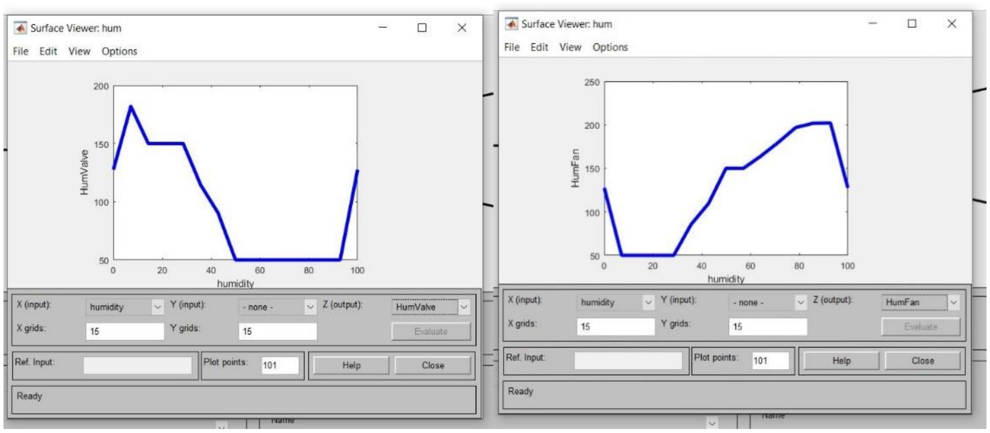

Figure 3.19: surface viewer of the rule base of air humidity FLC

Then the system model was created in the Matlab Simulink. Figure 3.20 below shows the air humidity model created.

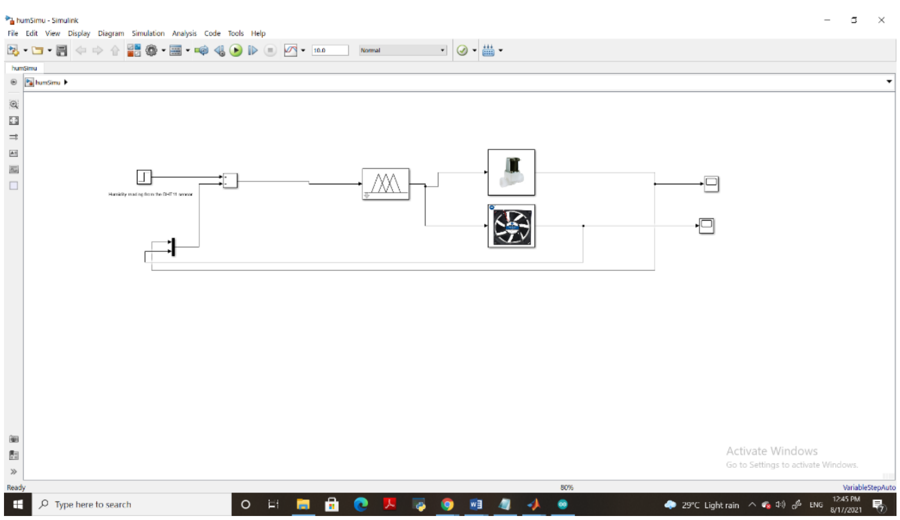

Figure 3.20: Simulink model of the air humidity FLC

Finally the fuzzy algorithm was coded in the Arduino IDE. Below figure 3.21 shows the algorithm.

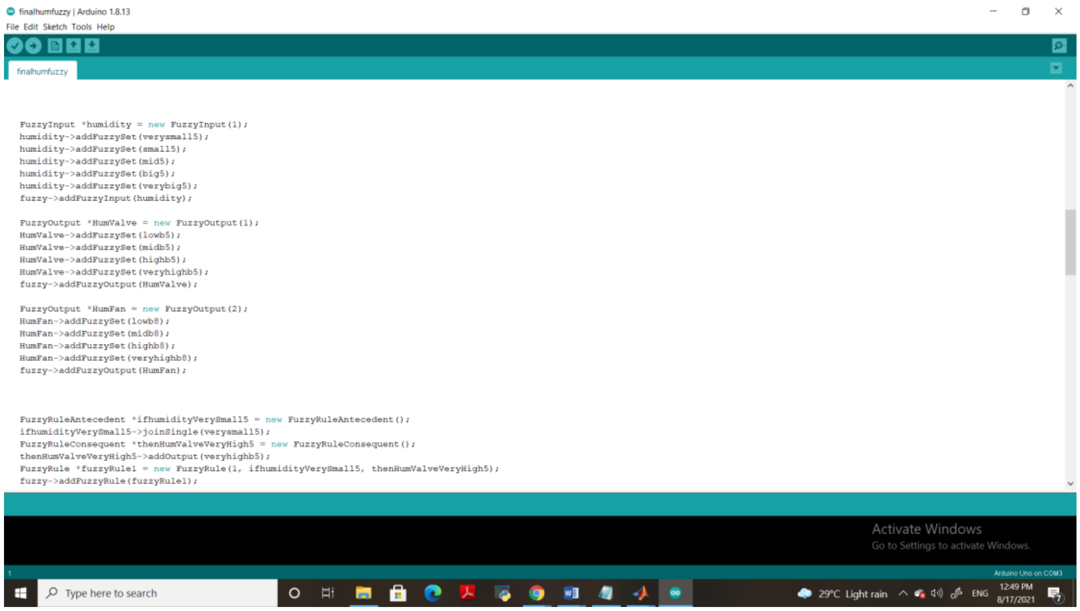

Figure 3.21: Fuzzy algorithm for the air humidity FLC

Fuzzy logic controller for the CO2 of air monitoring system

This fuzzy logic controller consists of one input and one output. The input is the CO2 level in the air measured by the mq135 air quality sensor. And the output is an air pump. The perfect CO2 level range that supports plant growth is around 600 to 1300 ppm.

At first, the fuzzy sets and membership functions for the input and output variables were created.

Table 3.7 and figure 3.22 below show the fuzzy sets and the membership functions for the input variable.

Table 3.7: Fuzzy sets of the input variable CO2 air state

No	Input Range	Variable name
1	0 to 300	verysmall1
2	100 to 600	small1
3	300 to 900	mid1
4	600 to 1200	big1
5	900 to 1500	verybig1

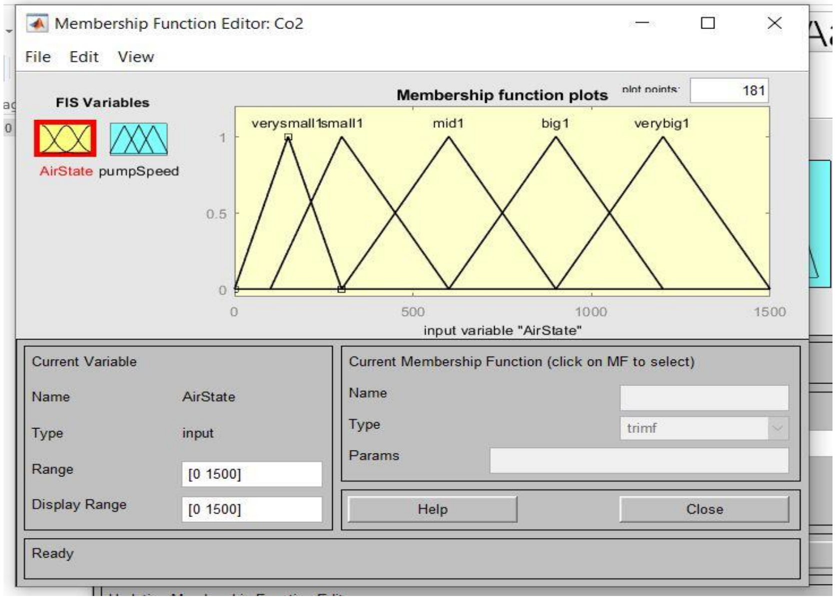

Figure 3.22: Membership functions for the fuzzy input CO2 air state in Matlab toolbox

Then the fuzzy sets and the membership functions for the output variable were created.

Table 3.8 and figure 3.23 below show the fuzzy sets and the membership functions for the output variable.

Table 3.8: Fuzzy sets for the output air pump

No	Output range	Variable name
1	0 to 100	lowb1
2	50 to 150	midb1
3	100 to 200	highb1
4	150 to 255	veryhighb1

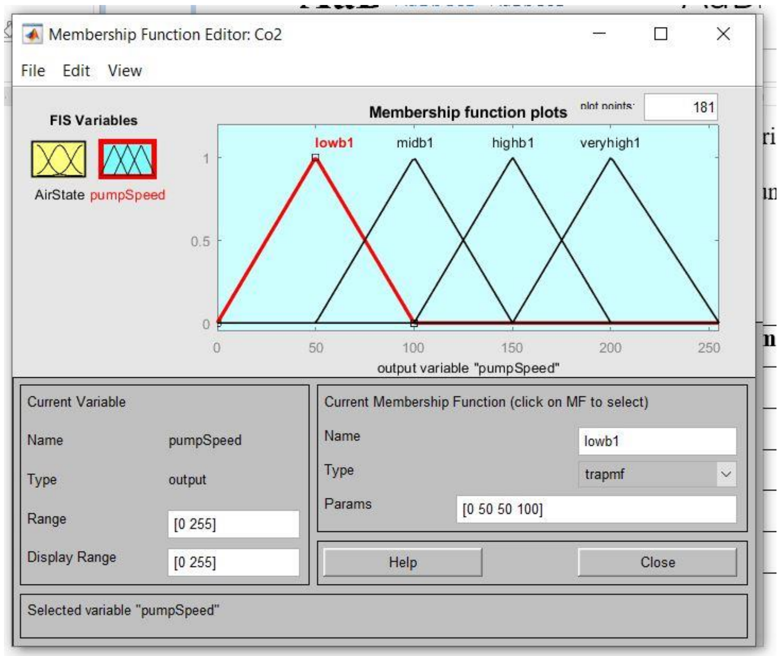

Figure 3.23: Membership functions for the fuzzy output air pump in Matlab toolbox

After that the rule base was created in the Matlab fuzzy toolbox. This CO2 level controlling FLC consists with five fuzzy rules.

Figures 3.24 and 3, 25 below shows the formation of rules and the rule viewer respectively.

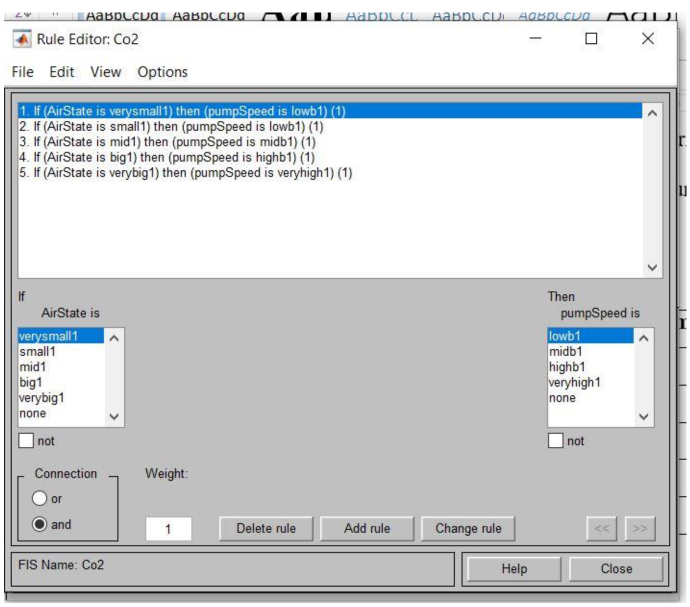

Figure 3.24: Formation of rules in the Matlab toolbox of CO2 level FLC

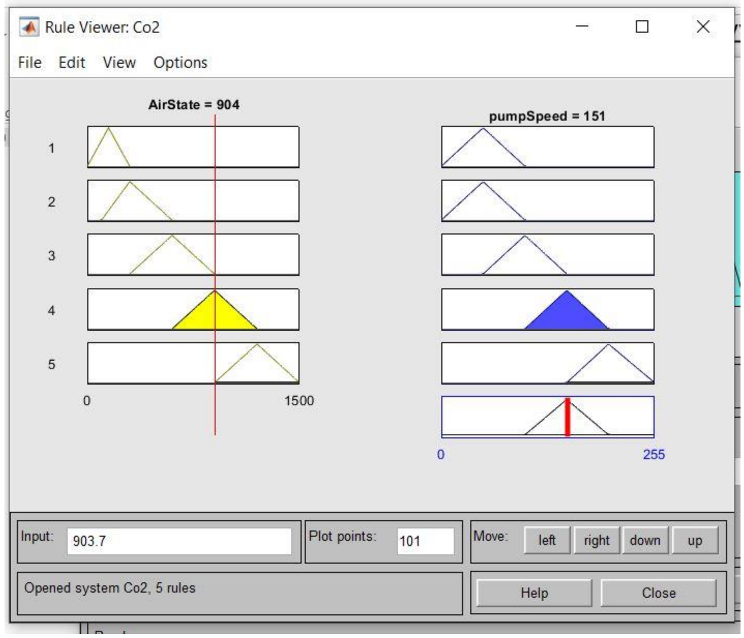

Figure 3.25: Rule viewer for CO2 level FLC

Surface view of the rule base of CO2 level FLC in shown by the below figure 3.26.

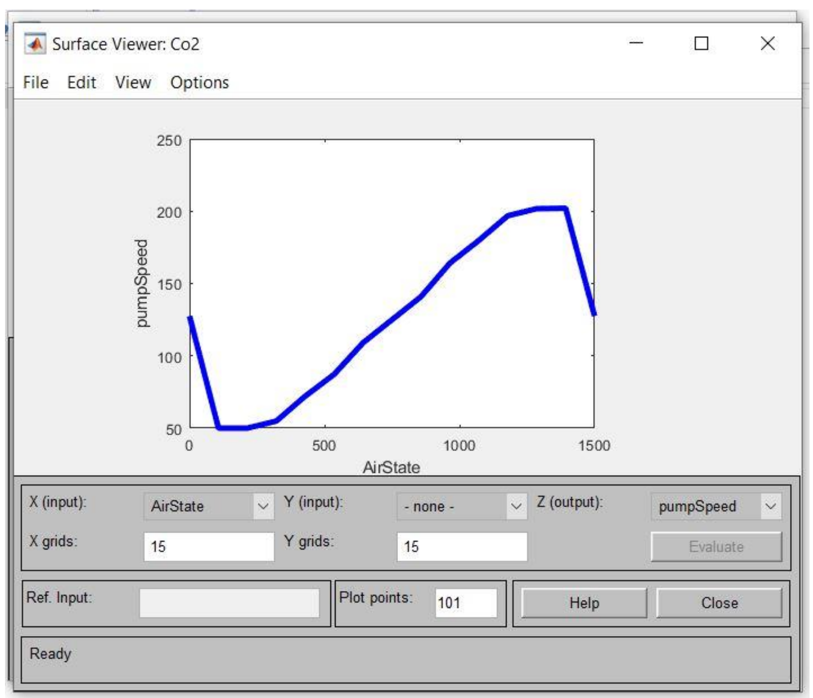

Figure 3.26: surface viewer of the rule base of CO2 FLC

Then the CO2 level FLC controller was modeled in Matlab Simulink. The figure 3.27 below shows the Matlab Simulink model.

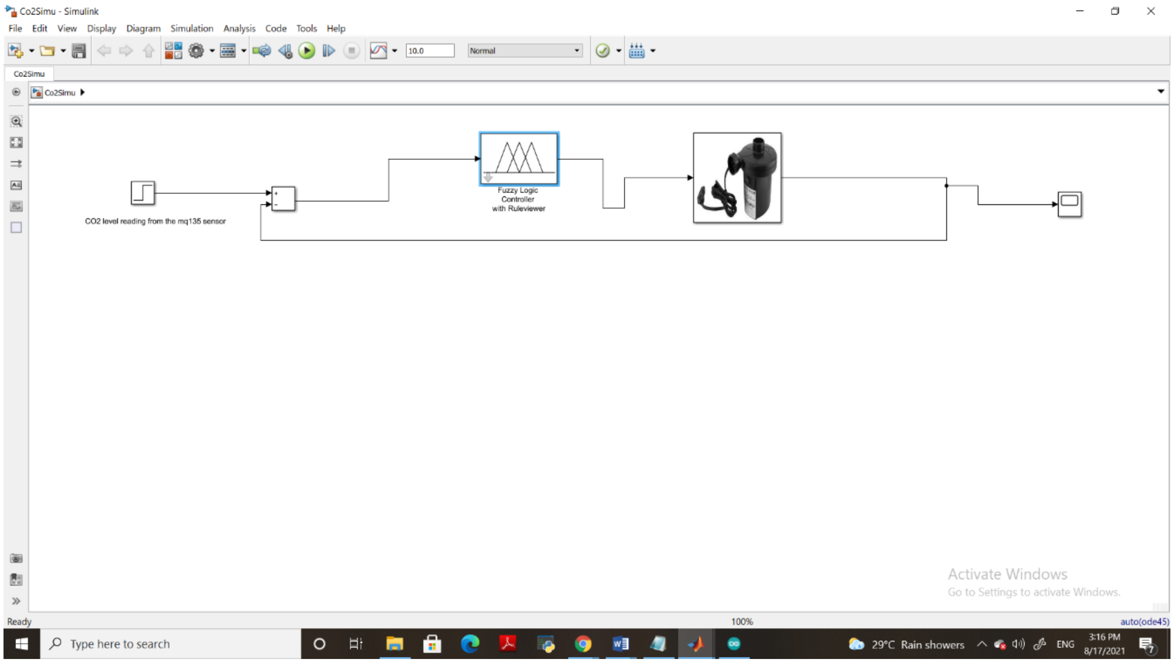

Figure 3.27: Simulink model of the CO2 level FLC

Then the fuzzy algorithm was coded in the Arduino IDE. Figure 3.28 below shows the Coding part.

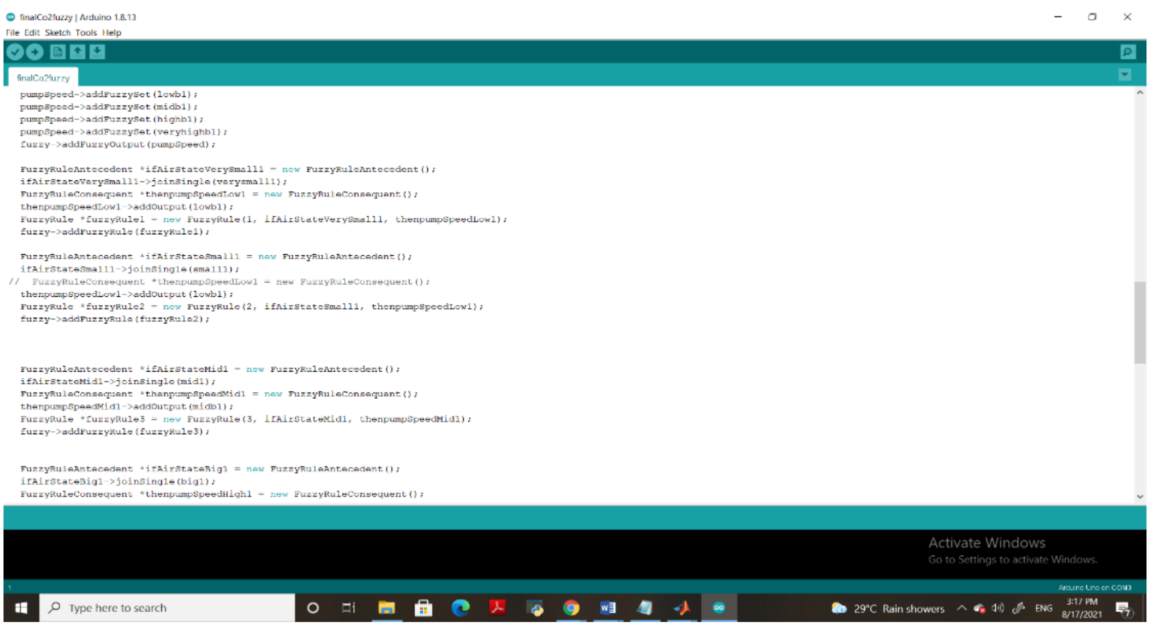

Figure 3.28: Fuzzy algorithm for the CO2 level FLC

Fuzzy logic controller for the water level monitoring system

The water level of the reservoir is measured using an ultrasonic sensor and a 12V valve is used to flow water when the water level is low in the reservoir. In the water level fuzzy logic controller there are one input variable and one output variable each. The input variable is the water level reading taken from the measurement of the ultrasonic sensor and the output variable is the water flow valve. Table 3.9 and figure 3.29 below show the fuzzy sets and the membership functions for the input variable respectively.

Table 3.9: Fuzzy sets of the input variable water level

No	Input Range	Variable name
1	0 to 4	verysmall3
2	2 to 6	small3
3	4 to 12	mid3
4	8 to 18	big3
5	13 to 25	verybig3

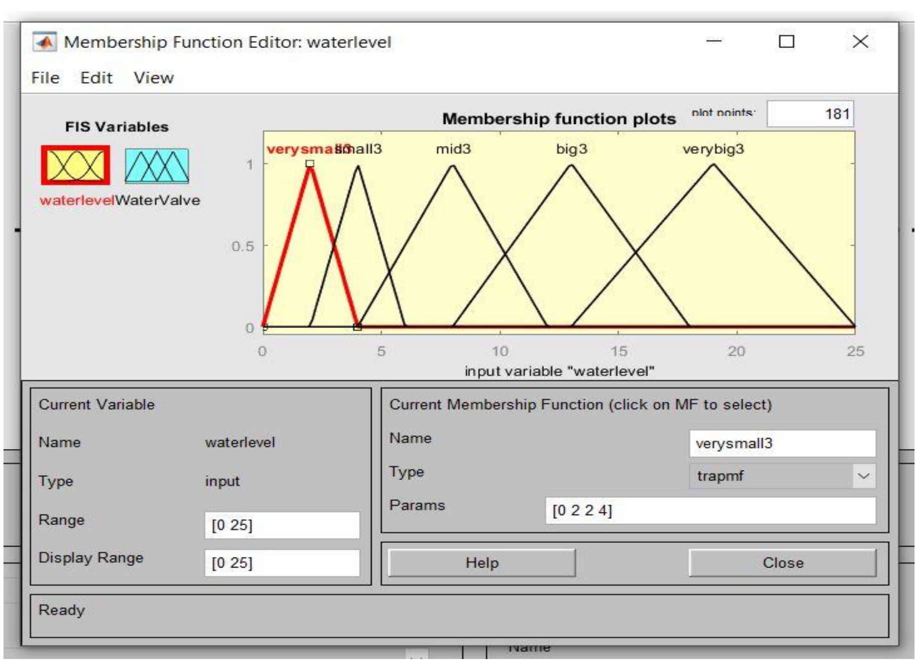

Figure 3.29: Membership functions for the fuzzy input water level in Matlab toolbox

Then I created the fuzzy sets and the membership functions for the output variable which is the water valve.

Table 3.10 and figure 3.30 below show the fuzzy sets and the membership functions for the output variable.

Table 3.10: Fuzzy sets for the output water valve

No	Output range	Variable name
1	0 to 100	lowb3
2	50 to 150	midb3
3	100 to 200	highb3
4	150 to 255	veryhighb3

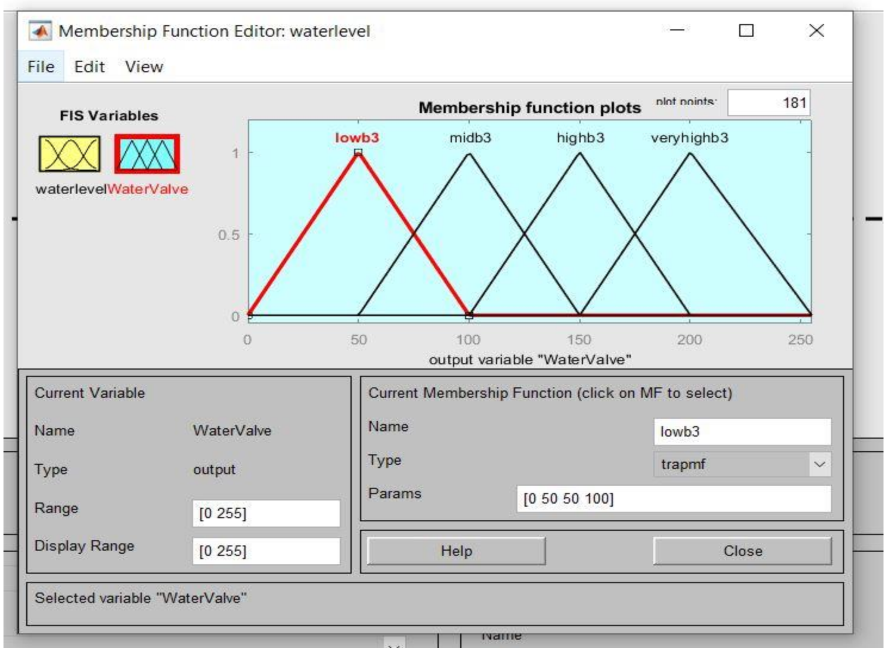

Figure 3.30: Membership functions for the fuzzy output water valve in Matlab toolbox

Then the rule base for the water level fuzzy logic controller was created in Matlab fuzzy toolbox.

Figure 3.31 below shows the formation of rules. The FLC controller consists with five fuzzy rules.

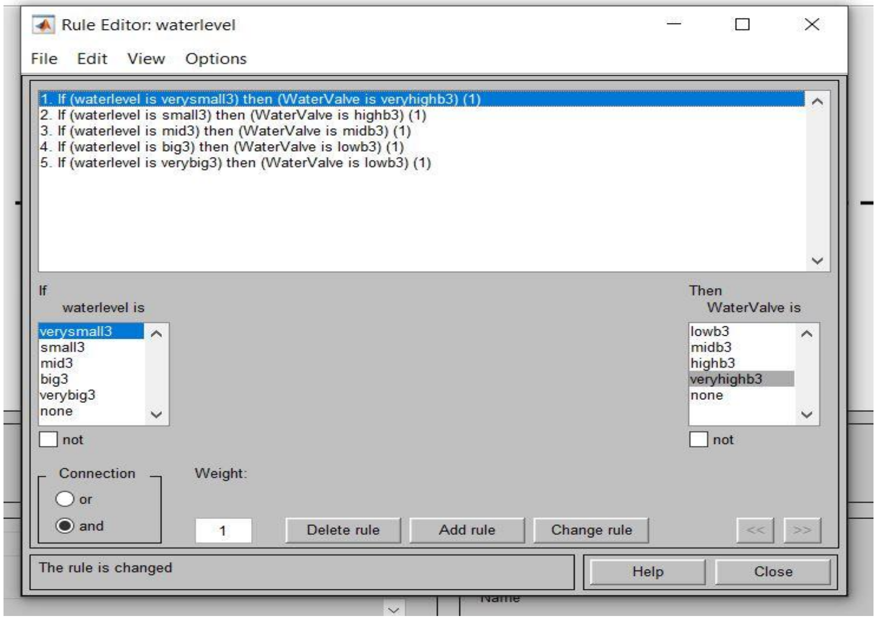

Figure 3.31: Formation of rules in the Matlab toolbox of water level FLC

Figure 3.32 below shows the rule viewer and the surface viewer.

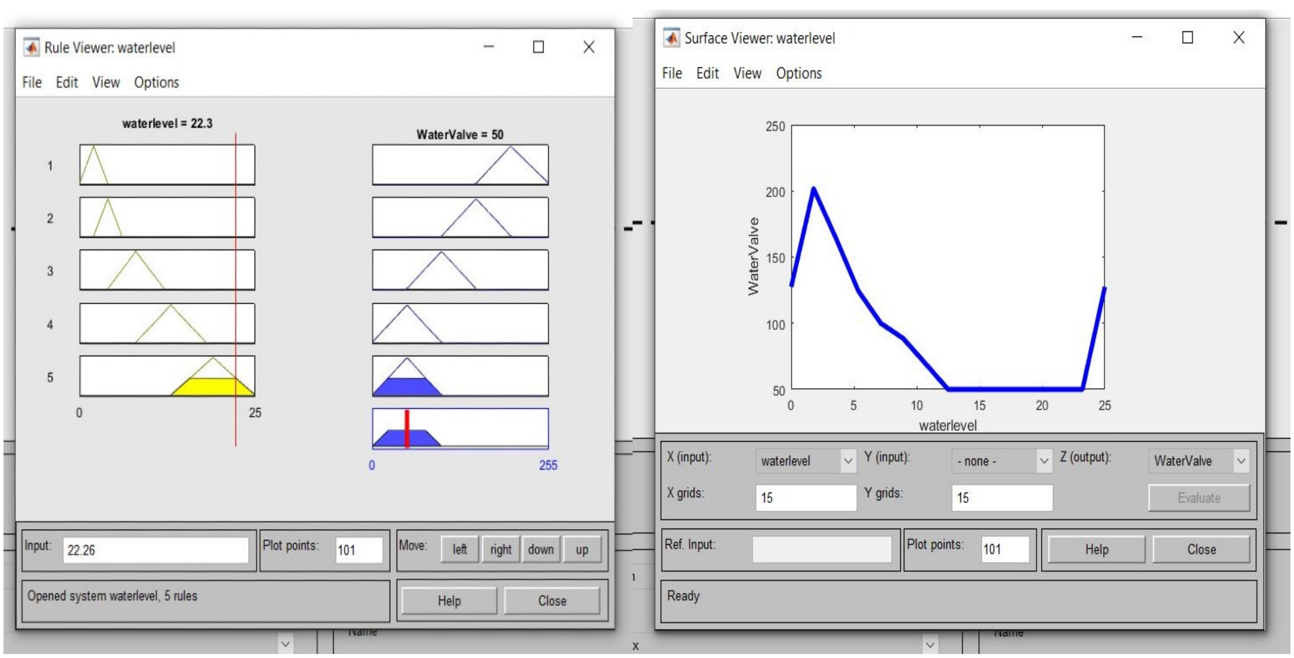

Figure 3.32: The rule viewer and the surface viewer of water level FLC

Then the model of the fuzzy logic controller was implemented in the Matlab Simulink. Finally the fuzzy algorithm was coded in the Arduino IDE.

Figures 3.33 and 3.34 shows the Simulink model and the code part.

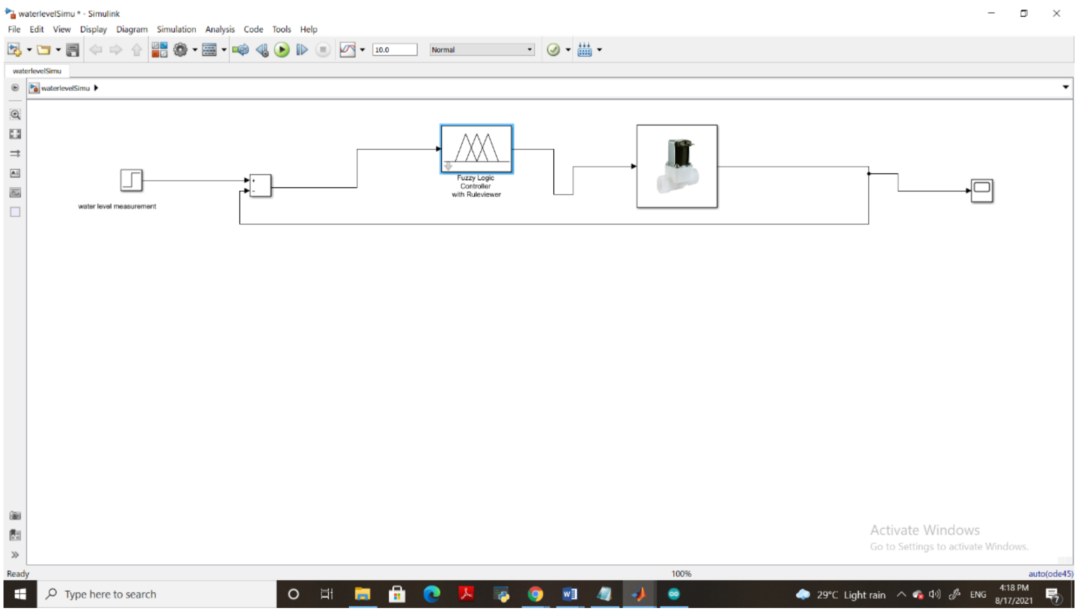

Figure 3.33: Simulink model of the water level FLC

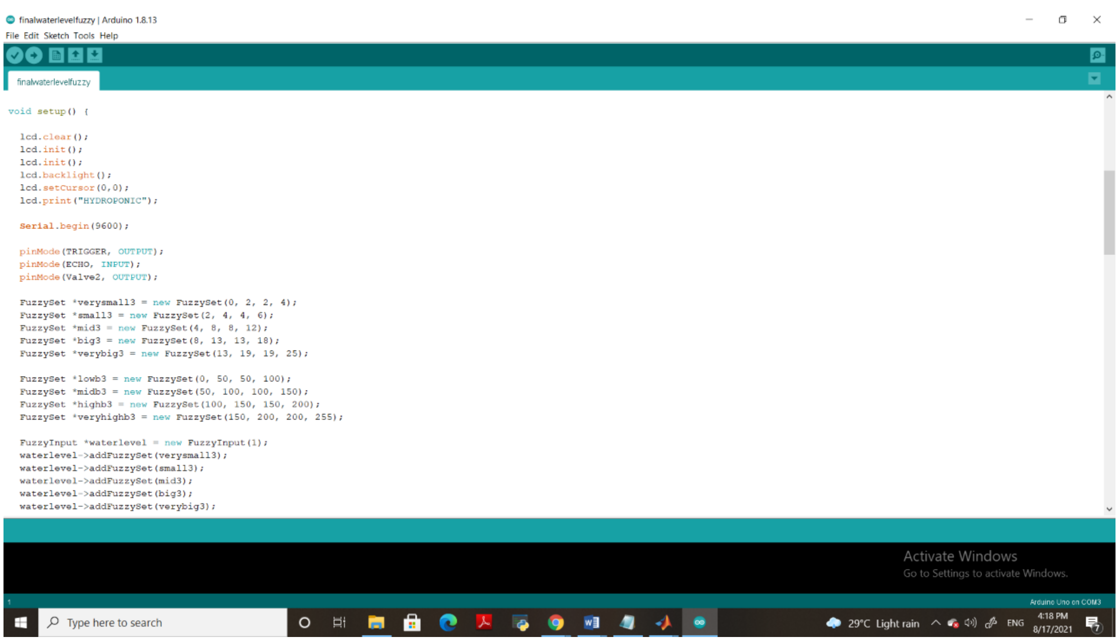

Figure 3.34: Fuzzy algorithm for the water level FLC

Fuzzy logic controller for the water temperature monitoring system

Extreme water temperatures around the root will decrease a plant's ability to absorb nutrients. It has been proven by professionals that the water temperature must be kept around 18-28 Celcius to support the roots to absorb nutrients more effectively in a hydroponic environment. This temperature range provides an ideal environment for healthy roots and effective nutrient absorption. The water temperature fuzzy logic controller consists of one input variable and one output variable. The input is the water temperature reading from the DS18B20 temperature sensor and the output is a heater coil.

Table 3.11 and figure 3.35 below shows the fuzzy sets and membership functions for the input variable.

Table 3.11: Fuzzy sets of the input variable water temperature

No	Input Range	Variable name
1	0 to 10	verysmall4
2	5 to 15	small4
3	10 to 30	mid4
4	20 to 40	big4
5	30 to 60	verybig4

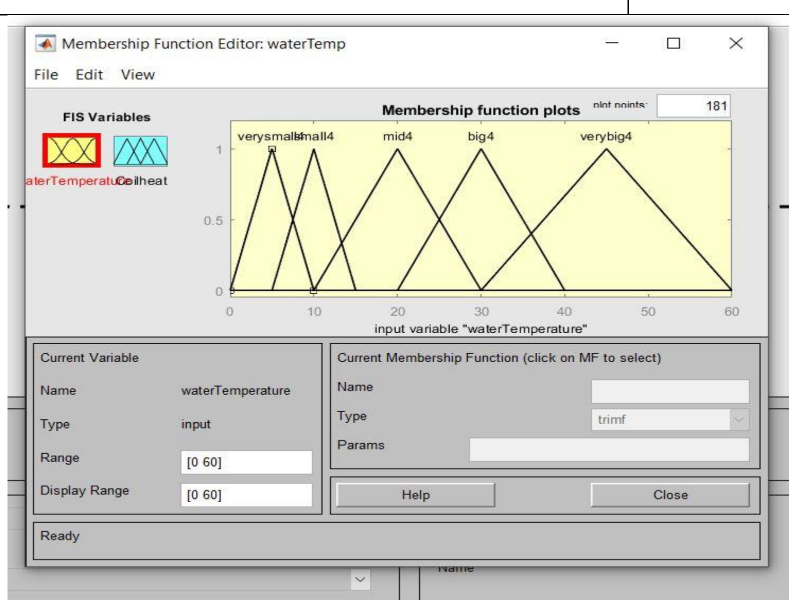

Figure 3.35: Membership functions for the fuzzy input water temperature in Matlab toolbox

Then the fuzzy sets and the membership functions for the output variable was created.

Table 3.12 and figure 3.36 below show the fuzzy sets and the membership functions for the output variable.

Table 3.12: Fuzzy sets for the output coil

No	Output range	Variable name
1	0 to 100	lowb4
2	50 to 150	midb4
3	100 to 200	highb4
4	150 to 255	veryhighb4

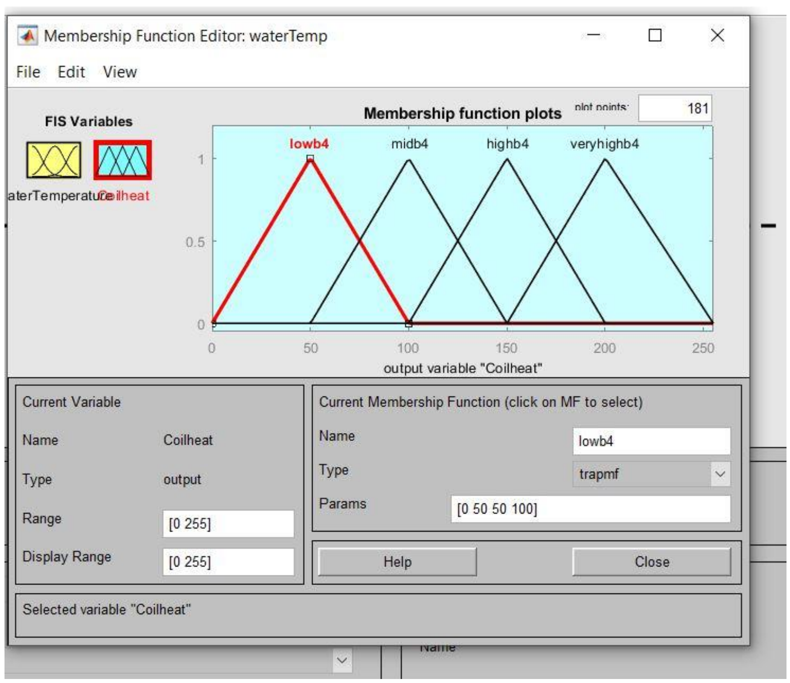

Figure 3.36: Membership functions for the fuzzy output coil in Matlab toolbox

Then the rule base was created for the FLC controller using the Matlab fuzzy toolbox.

Figures 3.37 below show the formation of rules and figure 3.38 below shows the rule viewer and the surface view of the rule of base of the water temperature fuzzy logic controller.

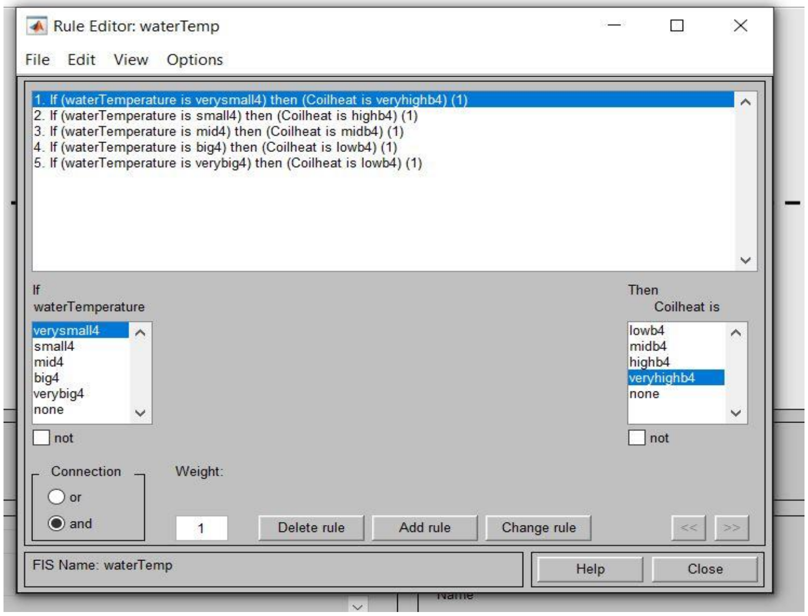

Figure 3.37: Formation of rules in the Matlab toolbox of water temperature FLC

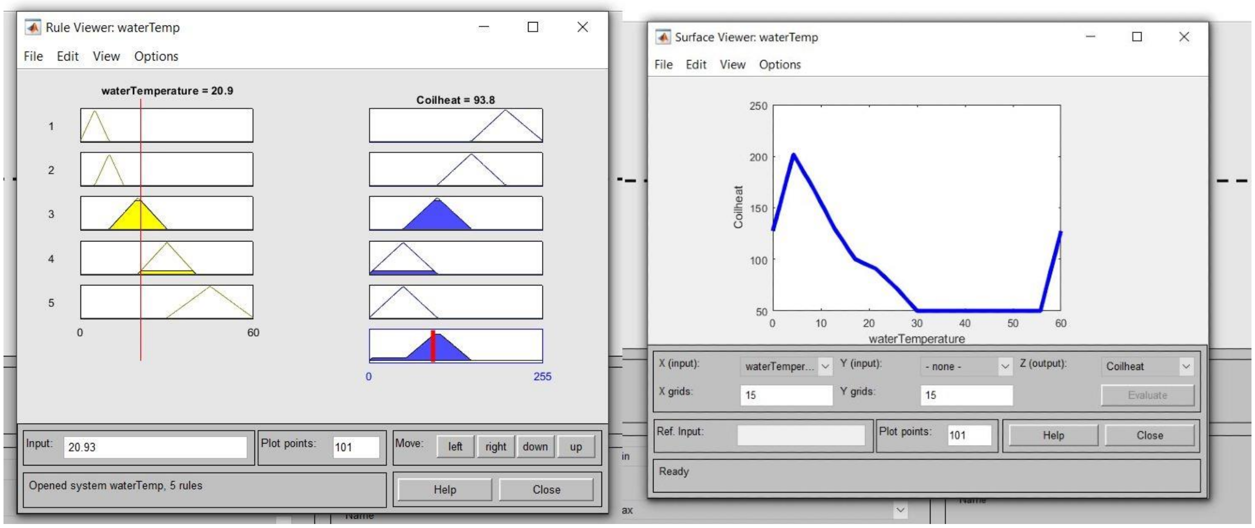

Figure 3.38: The rule viewer and the surface viewer of water temperature FLC

Then the water temperature FLC was modeled in the Matlab Simulink. Figure 3.39 below shows the Simulink model.

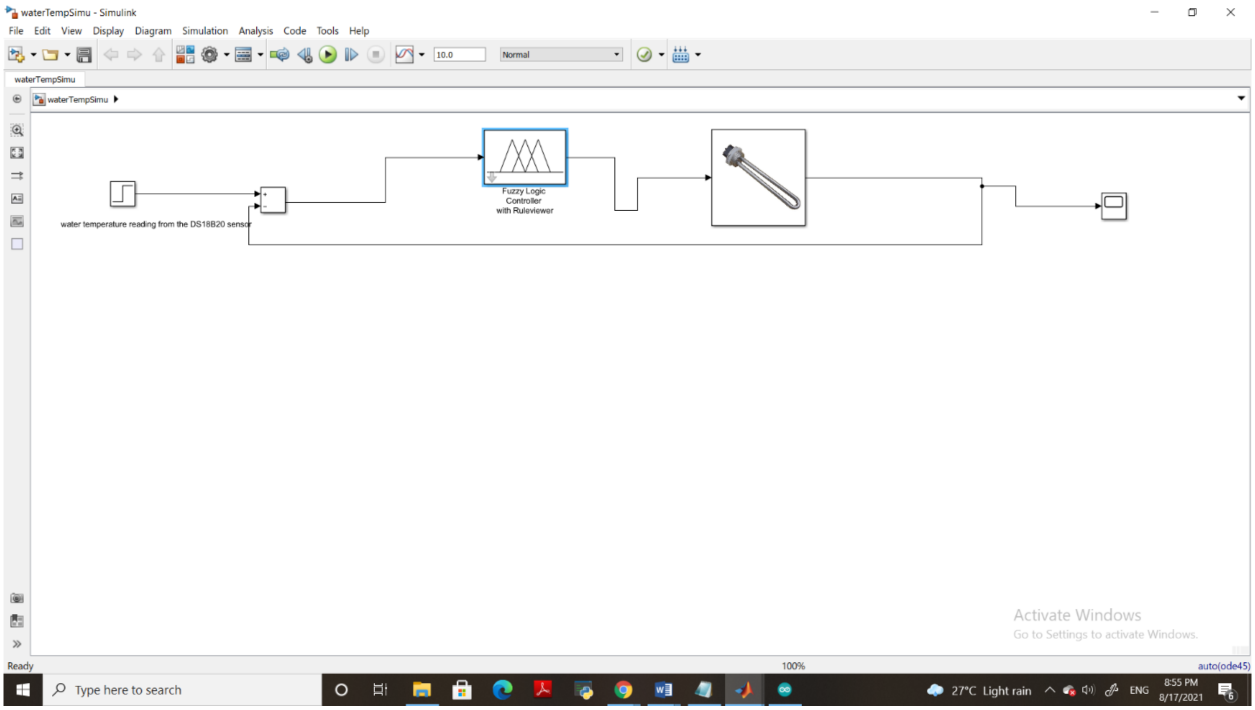

Figure 3.39: Simulink model of the water temperature FLC

Finally the fuzzy algorithm was coded in the Arduino IDE. Below figure 3.40 shows the coding part.

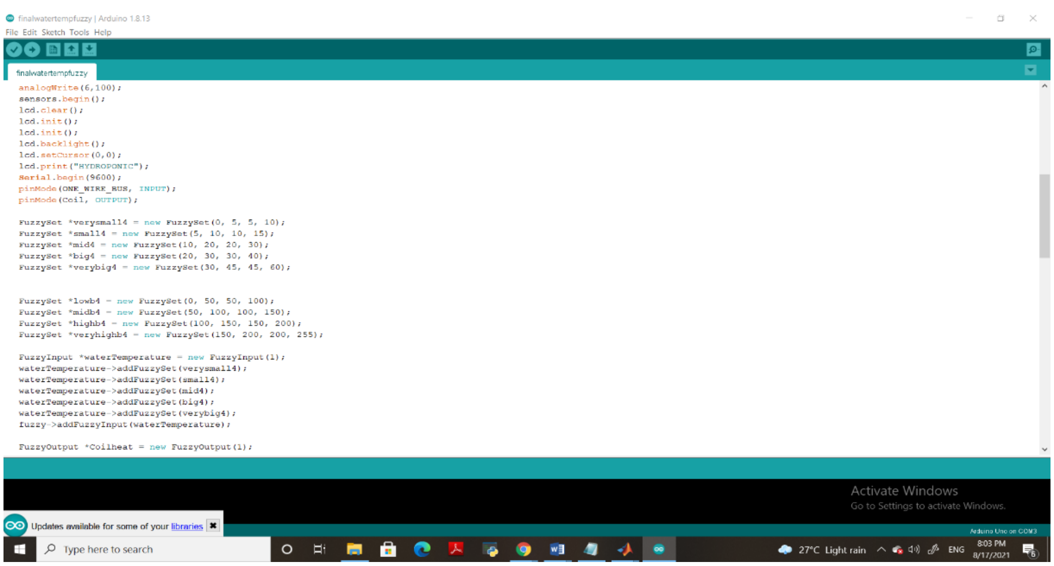

Figure 3.40: Fuzzy algorithm for the water temperature FLC

Fuzzy logic controller for the pH level monitoring system

The pH level fuzzy logic controller system consists of one input variable and one output variable. The input variable is the reading from the pH sensor and the output variable is the albert solution valve that flows the Albert solution to the main reservoir. The pH value of the nutrient-rich water solution must be maintained between the 6-7 range to support healthy and effective plant growth. This has been proven by experts. Firstly the fuzzy sets and the membership functions were created for the input and output variables. Table 3.13 and figure 3.41 below shows the fuzzy sets and membership functions for the input variable.

Table 3.13: Fuzzy sets of the input variable pH value

No	**Input Range**	**Variable name**
1	0 to 2	verysmall2
2	2 to 4	small2
3	2 to 10	mid2
4	4 to 12	big2
5	10 to 14	verybig2

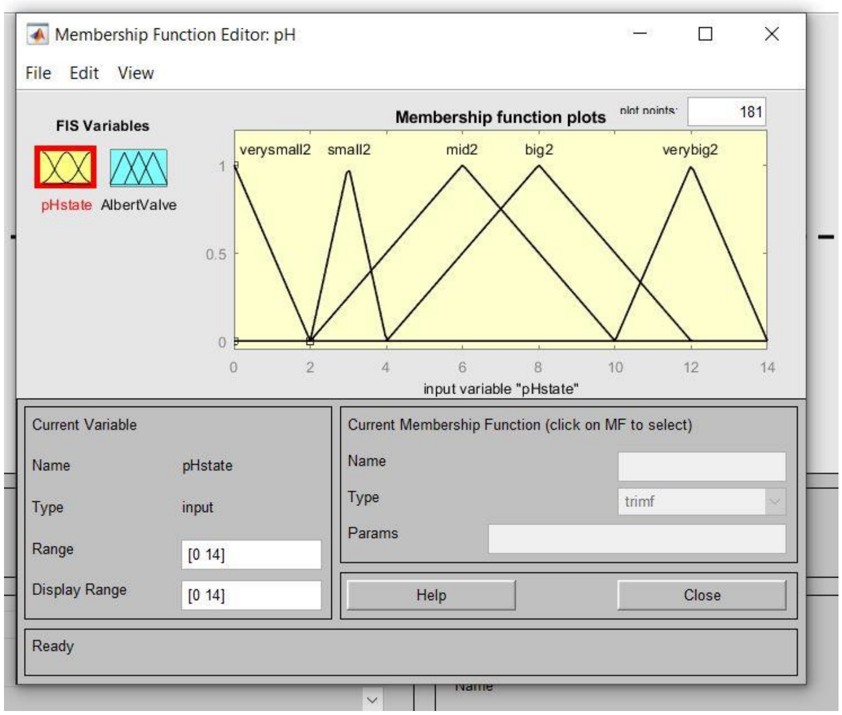

Figure 3.41: Membership functions for the fuzzy input pH value in Matlab toolbox

Table 3.14 and figure 3.42 below show the fuzzy sets and the membership functions for the output variable.

Table 3.14: Fuzzy sets for the output albert solution valve

No	Output range	Variable name
1	0 to 100	lowb2
2	50 to 150	midb2
3	100 to 200	highb2
4	150 to 255	veryhighb2

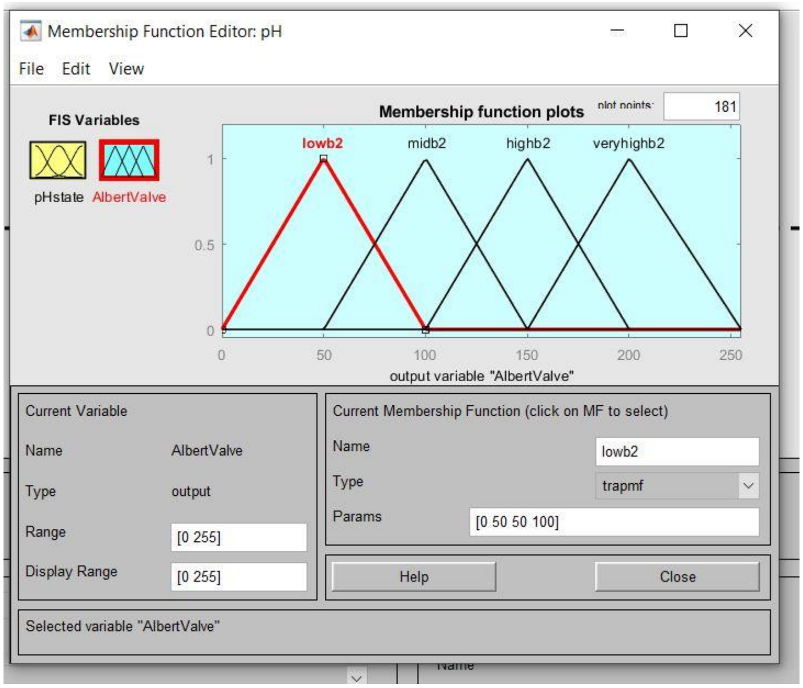

Figure 3.42: Membership functions for the fuzzy output Albert solution valve in Matlab toolbox

Then the rule base was created. Figures 3.43 below show the formation of rules and figure 3.44 below shows the rule viewer and the surface view of the rule of base of the pH level FLC.

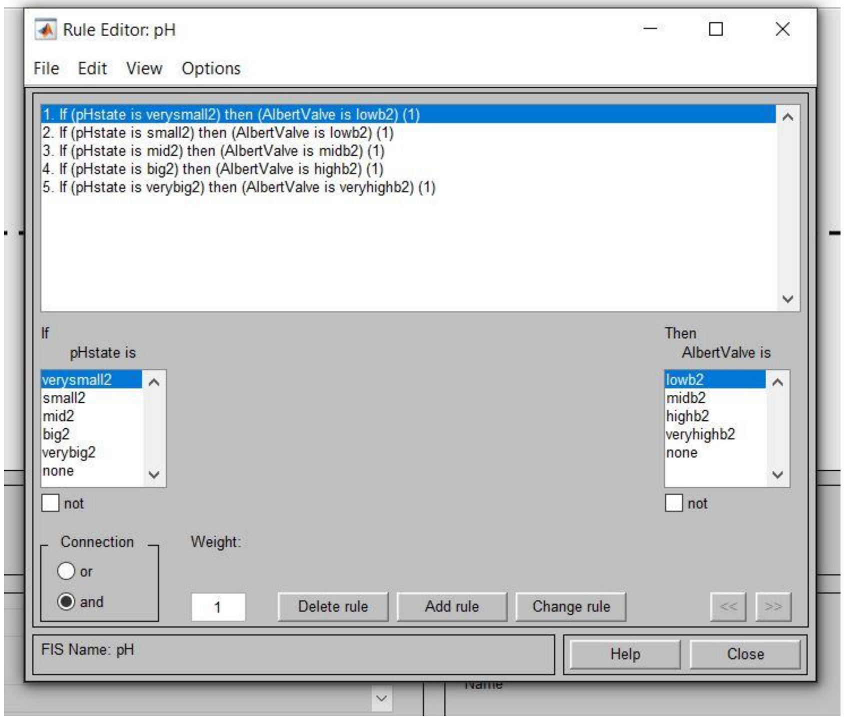

Figure 3.43: Formation of rules in the Matlab toolbox of pH level FLC

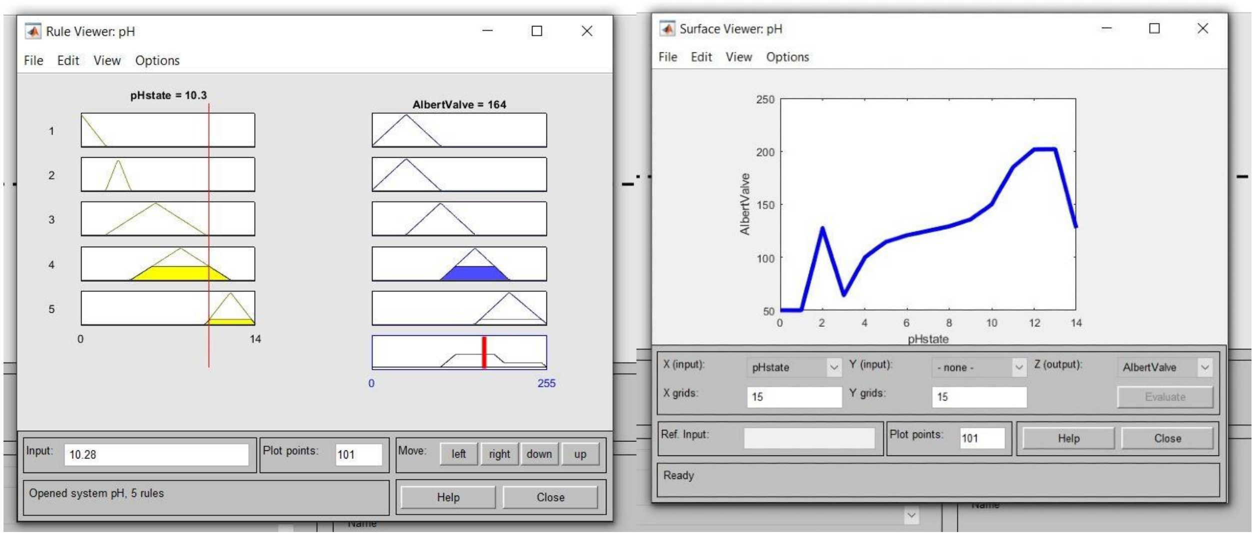

Figure 3.44: The rule viewer and the surface viewer of pH level FLC

Then the model of the fuzzy logic controller was implemented in the Matlab Simulink and finally the fuzzy algorithm was coded in the Arduino IDE.

Figures 3.45 and 3.46 shows the Simulink model and the code part.

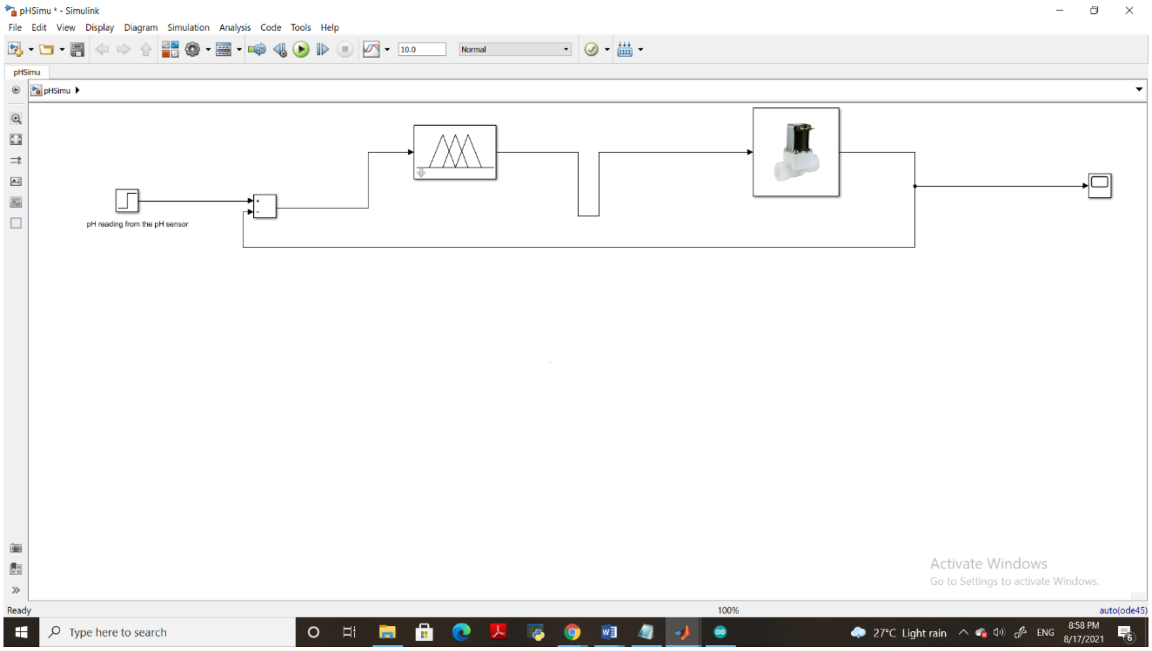

Figure 3.45: Simulink model of the pH level FLC

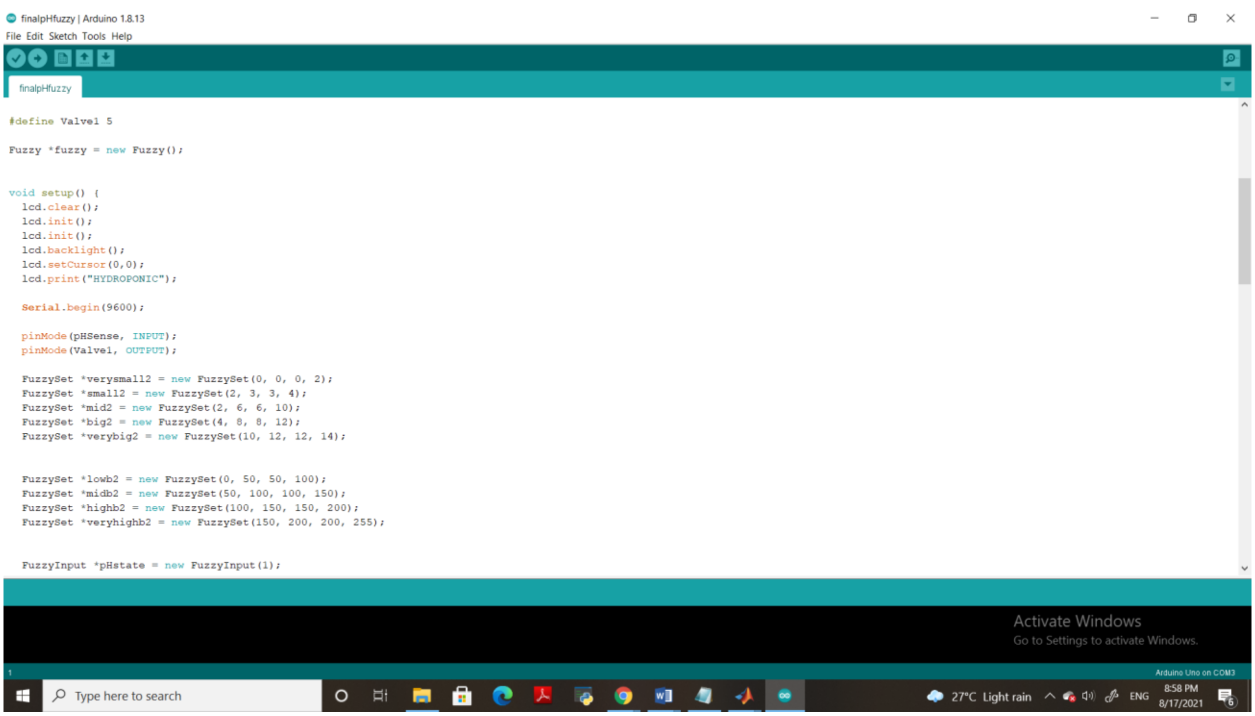

Figure 3.46: Fuzzy algorithm for the pH level FLC

3.3.3 Physical implementation

The structure we implemented in the first semester was big and was consuming a huge space. In this semester, we created a tabletop hydroponic structure that can be kept indoors without consuming much space and which is portable.

Figure 3.47 below shows some photos taken during fabrication.

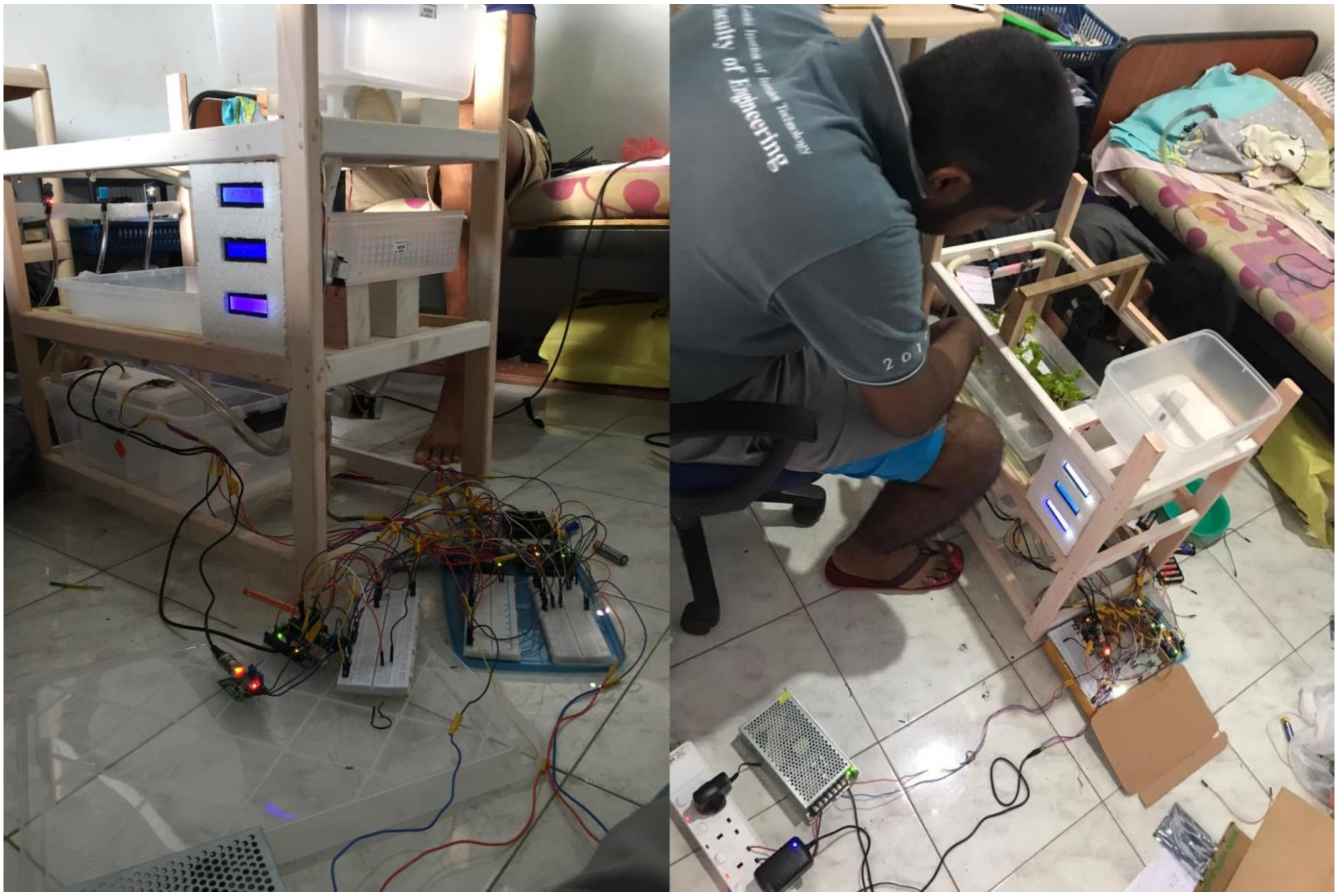

Figure 3.47: Photos taken during the fabrication of the product

Figure 3.48 below shows the final product.

Figure 3.48: The Final Product

3.4 Summary

This chapter has addressed the overview methodology of our work. The individual contribution has been discussed broadly using the technologies and theories used for implementation. Chapter 4 will be discussing the results and limitations of our product.

Chapter 4

RESULTS

Further development on the IoT-based automated hydroponic cultivation system using new technologies and theories was addressed in this study. The product was reconstructed as a tabletop structure, which gave the benefits such as consuming less space, being portable, making it more user friendly to the consumers, can be kept indoors, etc. The developed hydroponic system is controlling seven essential environmental factors which are mainly affecting plant growth, using relevant sensors and other required mechanisms. A plant disease identification system and an algae control system were also implemented in the product. The nutrient film technique hydroponic method was used in the product as it has proven to be the most efficient hydroponic method in the world at present and the nutrient film technique hydroponic method met our design requirements effectively. The water temperature of the nutrient solution and the CO2 level of air are the two additional parameters added further to the existed parameters. The parameters controlled before are the water level of the nutrient solution, temperature of the air, pH value of the nutrient solution, humidity of the air, and light intensity. The fuzzy logic controllers were implemented to control all seven parameters instead of the relays. This improved the efficiency of parameter controlling systems since, with fuzzy logic controllers, parameters can be maintained in the required range constantly. Additional mechanisms are added to the system such as exhaust fans to decrease air humidity and heater coils to control the water temperature.

Limitations

- The system was implemented to support only a single crop type cultivation. That means the parameter controlling systems are programmed only for controlling the parameters needed for a single crop type.

- The consumers do not have the option of inserting user input parameter levels into the system through a keypad or something manually.

Chapter 5

FINAL CONCLUSION

The objective of our study was to present information about the challenges in traditional soil cultivation and the use of automated hydroponic systems as an alternative for soil cultivation. This study has been identified some challenges in traditional soil-based crop cultivation, such as limited agricultural lands, unsuitable soil conditions for the desired crop type, pollution of the environment due to the usage of chemical fertilizers, unsuitable climatic factors for the cultivation, and mainly low efficiency of the harvest. After doing several pieces of research on this, Hydroponic Cultivation System has been identified as a solution, which can address the high extend of the above challenges. This system can allow producing food the whole year without any season or interval. The main barrier in Hydroponic cultivation is manpower with complexity handling capability. In this study Automation, Hydroponic Cultivation System has been proposed and implemented as a solution that addresses all these challenges. The automation of the system solves the issues of complicated manual monitoring and controlling processes. This system measuring and maintaining the main environmental factors (water level, water temperature, pH level of water base, Temperature, Humidity level, Light level, Co2 level of the air), and the system is using fuzzy controllers for the parameter controlling which gain the highest accuracy for the system.

References

[1] Nature.com. 2021. *Soil: The Foundation of Agriculture | Learn Science at Scitable*. [online] Available at: <https://www.nature.com/scitable/knowledge/library/soil-the-foundation-of-agriculture-84224268/> [Accessed 14 August 2021].

[2] Diynetwork.com. 2021. *Notification*. [online] Available at: <https://www.diynetwork.com/how-to/outdoors/gardening/introduction-to-hydroponics> [Accessed 14 August 2021].

[3] Stories.pinduoduo-global.com. 2021. *Hydroponics vs. Soil: Which Will Grow Plants Faster*. [online] Available at: <https://stories.pinduoduo-global.com/agritech-hub/hydroponics-vs-soil> [Accessed 14 August 2021].

[4] Mr Stacky Vertical Gardens. 2021. *10 reasons to choose hydroponic vs soil - Mr Stacky Vertical Gardening*. [online] Available at: <https://www.mrstacky.com.au/hydroponic-gardening-information/hydroponics-vs-soil/\> [Accessed 14 August 2021].

[5] saferbrand. 2021. *top-plants-hydroponic*. [online] Available at: <https://www.saferbrand.com/articles/top-plants-hydroponic> [Accessed 14 August 2021].

[6] Soil, H., 2021. *Hydroponics vs. Soil*. [online] Greenhouse Today. Available at: <https://www.greenhousetoday.com/hydroponics-vs-soil/> [Accessed 14 August 2021].

[7] Sites.google.com. 2021. *Data - Hydroponic and Soil Tomato Growth*. [online] Available at: <https://sites.google.com/site/hydroponicandsoiltomatogrowth/data> [Accessed 15 August 2021].

[8] Databridgemarketresearch.com. 2021. *Hydroponics System Market – Global Industry Trends and Forecast to 2027 | Data Bridge Market Research*. [online] Available at: <https://www.databridgemarketresearch.com/reports/global-hydroponics-system-market> [Accessed 15 August 2021].

[10] Aisii.azc.uam.mx. 2021. [online] Available at: <http://aisii.azc.uam.mx/mcbc/Cursos/IntCompt/Lectura15.pdf> [Accessed 15 August 2021].

[11] Medium. 2021. *A very brief introduction to Fuzzy Logic and Fuzzy Systems*. [online] Available at: <https://towardsdatascience.com/a-very-brief-introduction-to-fuzzy-logic-and-fuzzy-systems-d68d14b3a3b8> [Accessed 15 August 2021].

[12] Automation, I. and applications, W., 2021. *What are fuzzifier and Defuzzifier?*. [online] Industrial Automation, PLC Programming, scada & Pid Control System. Available at: <https://forumautomation.com/t/what-are-fuzzifier-and-defuzzifier/4148> [Accessed 15 August 2021].

[13]"History of Hydroponics: When Was Hydroponics Invented?", *Epic Gardening*, 2021. [Online]. Available: https://www.epicgardening.com/history-of-hydroponics/. [Accessed: 15- Aug- 2021].

[14]"Project Background – MECE E3430 Hydropotomous", *Blogs.cuit.columbia.edu*, 2021. [Online]. Available: https://blogs.cuit.columbia.edu/hydropotomous/project-components/. [Accessed: 15- Aug- 2021].

[15] "Autonomous Hydroponic System", Csus.edu, 2020. [Online]. Available: https://www.csus.edu/indiv/t/tatror/senior_design/SD%20F19-S20%20Reports/Team_3_F19S20_EOP_Report.pdf. [Accessed: 15- Aug- 2021].

[16]L. Soto, "How Nutrient Film Technique (NFT) Hydroponic Systems Work - Pure Greens: Custom Container Farms", *Pure Greens: Custom Container Farms*, 2021. [Online]. Available: https://puregreensaz.com/nutrient-film-technique/. [Accessed: 15- Aug- 2021].

[17]"Free research papers and projects on fuzzy logic", *Engpaper.com*, 2021. [Online]. Available: https://www.engpaper.com/free-research-papers-and-projects-on-fuzzy-logic.htm. [Accessed: 15- Aug- 2021].

[18]"Temperature Control using Fuzzy Logic", *Arxiv.org*, 2021. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1402/1402.3654.pdf. [Accessed: 15- Aug- 2021].

[19]"Fuzzy Logic Controller of temperature and humidity inside an agricultural greenhouse", *ResearchGate*, 2021. [Online]. Available: https://www.researchgate.net/publication/303563634_Fuzzy_logic_controller_of_temperature_and_humidity_inside_an_agricultural_greenhouse/link/59d6191d458515140ee7b25b/download. [Accessed: 15- Aug- 2021].

[20]"Fully Automated Hydroponic System for Indoor Plant Growth", *ResearchGate*, 2021. [Online]. Available: https://www.researchgate.net/publication/324508835_Fully_Automated_Hydroponic_System_for_Indoor_Plant_Growth. [Accessed: 16- Aug- 2021].

[21]*Domoticz.com*, 2021. [Online]. Available: https://www.domoticz.com/DomoticzManual.pdf. [Accessed: 16- Aug- 2021].

[22]"Reshaping the World with Fuzzy Logic", *We Are WorldQuant*, 2021. [Online]. Available: https://www.weareworldquant.com/en/thought-leadership/reshaping-the-world-with-fuzzy-logic/. [Accessed: 16- Aug- 2021].

[21]"How Does DS18B20 Temperature Sensor Work?", *Apogeeweb.net*, 2021. [Online]. Available: https://www.apogeeweb.net/circuitry/ds18b20-working-principle.html#2-2-ds18b20-features. [Accessed: 18- Aug- 2021].

[22]"DS18B20 (Digital Temperature Sensor) and Arduino", *Arduino Project Hub*, 2021. [Online]. Available: https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806. [Accessed: 18- Aug- 2021].

[23]"DS18B20 Temperature Sensor", *Components101*, 2021. [Online]. Available: https://components101.com/sensors/ds18b20-temperature-sensor. [Accessed: 18- Aug- 2021].

[24]"MQ135 Air Quality Gas Sensor buy online at Low Price in India - ElectronicsComp.com", *Electronicscomp.com*, 2021. [Online]. Available: https://www.electronicscomp.com/mq135-air-quality-control-gas-sensor. [Accessed: 18- Aug- 2021].

[25]"MQ135 Alcohol Sensor Circuit and Its Working", *ElProCus - Electronic Projects for Engineering Students*, 2021. [Online]. Available: https://www.elprocus.com/mq-135-alcohol-sensor-circuit-and-working/. [Accessed: 18- Aug- 2021].

APPENDIX

Combined code for Air Temperature and Air Humidity Fuzzy Logic Controllers

```
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
LiquidCrystal_I2C lcd(0x27,16,2);

#define LDR A0
#define DHTPIN 7
#define DHTTYPE DHT11
#define FAN 5
#define Valve3 6
#define EFan 3

String T = "AirTemp: ";
String L = "Light_level:";
String H = "Humidity: ";

DHT dht(DHTPIN,DHTTYPE);
Fuzzy *fuzzy = new Fuzzy();
void setup() {
  lcd.clear();
  lcd.init();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
```

```
lcd.print("HYDROPONIC");
Serial.begin(9600);

dht.begin();

pinMode(DHTPIN, INPUT);
pinMode(LDR, INPUT);
pinMode(FAN, OUTPUT);
pinMode(Valve3, OUTPUT);
pinMode(EFan, OUTPUT);

FuzzySet *verysmall = new FuzzySet(0, 0, 0, 5);
FuzzySet *small = new FuzzySet(0, 0, 0, 10);
FuzzySet *mid = new FuzzySet(5, 10, 10, 15);
FuzzySet *big = new FuzzySet(10, 20, 20, 30);
FuzzySet *verybig = new FuzzySet(20, 30, 30, 40);

FuzzySet *lowldr = new FuzzySet(0, 300, 300, 600);
FuzzySet *midldr = new FuzzySet(300, 600, 600, 900);
FuzzySet *highldr = new FuzzySet(900 ,1200, 1200, 1500);

FuzzySet *lowb = new FuzzySet(0, 50, 50, 100);
FuzzySet *midb = new FuzzySet(50, 100, 100, 150);
FuzzySet *highb = new FuzzySet(100, 150, 150, 200);
FuzzySet *veryhighb = new FuzzySet(150, 200, 200, 255);

FuzzySet *verysmall5 = new FuzzySet(0, 5, 5, 10);
FuzzySet *small5 = new FuzzySet(0, 20, 20, 40);
FuzzySet *mid5 = new FuzzySet(30, 40, 40, 50);
FuzzySet *big5 = new FuzzySet(40, 60, 60, 80);
FuzzySet *verybig5 = new FuzzySet(60, 80, 80, 100);

FuzzySet *lowb5 = new FuzzySet(0, 50, 50, 100);
FuzzySet *midb5 = new FuzzySet(50, 100, 100, 150);
FuzzySet *highb5 = new FuzzySet(100, 150, 150, 200);
FuzzySet *veryhighb5 = new FuzzySet(150, 200, 200, 255);
```

```
FuzzySet *lowb8 = new FuzzySet(0, 50, 50, 100);
FuzzySet *midb8 = new FuzzySet(50, 100, 100, 150);
FuzzySet *highb8 = new FuzzySet(100, 150, 150, 200);
FuzzySet *veryhighb8 = new FuzzySet(150, 200, 200, 255);

FuzzyInput *Temperature = new FuzzyInput(1);
Temperature->addFuzzySet(verysmall);
Temperature->addFuzzySet(small);
Temperature->addFuzzySet(mid);
Temperature->addFuzzySet(big);
Temperature->addFuzzySet(verybig);
fuzzy->addFuzzyInput(Temperature);

FuzzyInput *ldr = new FuzzyInput(2);
ldr->addFuzzySet(lowldr);
ldr->addFuzzySet(midldr);
ldr->addFuzzySet(highldr);
fuzzy->addFuzzyInput(ldr);

FuzzyInput *humidity = new FuzzyInput(3);
humidity->addFuzzySet(verysmall5);
humidity->addFuzzySet(small5);
humidity->addFuzzySet(mid5);
humidity->addFuzzySet(big5);
humidity->addFuzzySet(verybig5);
fuzzy->addFuzzyInput(humidity);

FuzzyOutput *fanSpeed = new FuzzyOutput(1);
fanSpeed->addFuzzySet(lowb);
fanSpeed->addFuzzySet(midb);
fanSpeed->addFuzzySet(highb);
fanSpeed->addFuzzySet(veryhighb);
fuzzy->addFuzzyOutput(fanSpeed);

FuzzyOutput *HumValve = new FuzzyOutput(2);
HumValve->addFuzzySet(lowb5);
HumValve->addFuzzySet(midb5);
HumValve->addFuzzySet(highb5);
```

```
HumValve->addFuzzySet(veryhighb5);
fuzzy->addFuzzyOutput(HumValve);

FuzzyOutput *HumFan = new FuzzyOutput(3);
HumFan->addFuzzySet(lowb8);
HumFan->addFuzzySet(midb8);
HumFan->addFuzzySet(highb8);
HumFan->addFuzzySet(veryhighb8);
fuzzy->addFuzzyOutput(HumFan);

FuzzyRuleAntecedent *ifTemperatureVerySmallAndLdrIsLow = new FuzzyRuleAntecedent();
ifTemperatureVerySmallAndLdrIsLow->joinWithAND(verysmall, lowldr);
FuzzyRuleConsequent *thenFanSpeedLow = new FuzzyRuleConsequent();
thenFanSpeedLow->addOutput(lowb);
FuzzyRule *fuzzyRule1 = new FuzzyRule(1, ifTemperatureVerySmallAndLdrIsLow, thenFanSpeedLow);
fuzzy->addFuzzyRule(fuzzyRule1);

FuzzyRuleAntecedent *ifTemperatureVerySmallAndLdrIsMid = new FuzzyRuleAntecedent();
ifTemperatureVerySmallAndLdrIsMid->joinWithAND(verysmall, midldr);
thenFanSpeedLow->addOutput(lowb);
FuzzyRule *fuzzyRule2 = new FuzzyRule(2, ifTemperatureVerySmallAndLdrIsMid, thenFanSpeedLow);
fuzzy->addFuzzyRule(fuzzyRule2);

FuzzyRuleAntecedent *ifTemperatureVerySmallAndLdrIsHigh = new FuzzyRuleAntecedent();
ifTemperatureVerySmallAndLdrIsHigh->joinWithAND(verysmall, highldr);
FuzzyRuleConsequent *thenFanSpeedMidb = new FuzzyRuleConsequent();
thenFanSpeedMidb->addOutput(midb);
FuzzyRule *fuzzyRule3 = new FuzzyRule(3, ifTemperatureVerySmallAndLdrIsHigh, thenFanSpeedMidb);
fuzzy->addFuzzyRule(fuzzyRule3);

FuzzyRuleAntecedent *ifTemperatureSmallAndLdrIsLow = new FuzzyRuleAntecedent();
ifTemperatureSmallAndLdrIsLow->joinWithAND(small, lowldr);
thenFanSpeedLow->addOutput(lowb);
FuzzyRule *fuzzyRule4 = new FuzzyRule(4, ifTemperatureSmallAndLdrIsLow, thenFanSpeedLow);
fuzzy->addFuzzyRule(fuzzyRule4);

FuzzyRuleAntecedent *ifTemperatureSmallAndLdrIsMid = new FuzzyRuleAntecedent();
ifTemperatureSmallAndLdrIsMid->joinWithAND(small, midldr);
```

```
thenFanSpeedMidb->addOutput(midb);
FuzzyRule *fuzzyRule5 = new FuzzyRule(5, ifTemperatureSmallAndLdrIsMid, thenFanSpeedMidb);
fuzzy->addFuzzyRule(fuzzyRule5);

FuzzyRuleAntecedent *ifTemperatureSmallAndLdrIsHigh = new FuzzyRuleAntecedent();
ifTemperatureSmallAndLdrIsHigh->joinWithAND(small, highldr);
FuzzyRuleConsequent *thenFanSpeedHigh = new FuzzyRuleConsequent();
thenFanSpeedHigh->addOutput(highb);
FuzzyRule *fuzzyRule6 = new FuzzyRule(6, ifTemperatureSmallAndLdrIsHigh, thenFanSpeedHigh);
fuzzy->addFuzzyRule(fuzzyRule6);

FuzzyRuleAntecedent *ifTemperatureMidAndLdrIsLow = new FuzzyRuleAntecedent();
ifTemperatureMidAndLdrIsLow->joinWithAND(mid, lowldr);
thenFanSpeedMidb->addOutput(midb);
FuzzyRule *fuzzyRule7 = new FuzzyRule(7, ifTemperatureMidAndLdrIsLow, thenFanSpeedMidb);
fuzzy->addFuzzyRule(fuzzyRule7);

FuzzyRuleAntecedent *ifTemperatureMidAndLdrIsMid = new FuzzyRuleAntecedent();
ifTemperatureMidAndLdrIsMid->joinWithAND(mid, midldr);
thenFanSpeedHigh->addOutput(highb);
FuzzyRule *fuzzyRule8 = new FuzzyRule(8, ifTemperatureMidAndLdrIsMid, thenFanSpeedHigh);
fuzzy->addFuzzyRule(fuzzyRule8);

FuzzyRuleAntecedent *ifTemperatureMidAndLdrIsHigh = new FuzzyRuleAntecedent();
ifTemperatureMidAndLdrIsHigh->joinWithAND(mid, highldr);
thenFanSpeedHigh->addOutput(highb);
FuzzyRule *fuzzyRule9 = new FuzzyRule(9, ifTemperatureMidAndLdrIsHigh, thenFanSpeedHigh);
fuzzy->addFuzzyRule(fuzzyRule9);

FuzzyRuleAntecedent *ifTemperatureBigAndLdrIsLow = new FuzzyRuleAntecedent();
ifTemperatureBigAndLdrIsLow->joinWithAND(big, lowldr);
thenFanSpeedHigh->addOutput(highb);
FuzzyRule *fuzzyRule10 = new FuzzyRule(10, ifTemperatureBigAndLdrIsLow, thenFanSpeedHigh);
fuzzy->addFuzzyRule(fuzzyRule10);

FuzzyRuleAntecedent *ifTemperatureBigAndLdrIsMid = new FuzzyRuleAntecedent();
ifTemperatureBigAndLdrIsMid->joinWithAND(big, midldr);
thenFanSpeedHigh->addOutput(highb);
```

```
  FuzzyRule *fuzzyRule11 = new FuzzyRule(11, ifTemperatureBigAndLdrIsMid, thenFanSpeedHigh);
  fuzzy->addFuzzyRule(fuzzyRule11);

  FuzzyRuleAntecedent *ifTemperatureBigAndLdrIsHigh = new FuzzyRuleAntecedent();
  ifTemperatureBigAndLdrIsHigh->joinWithAND( big, highldr);
  FuzzyRuleConsequent *thenFanSpeedVeryHigh = new FuzzyRuleConsequent();
  thenFanSpeedVeryHigh->addOutput(veryhighb);
  FuzzyRule *fuzzyRule12 = new FuzzyRule(12, ifTemperatureBigAndLdrIsHigh, thenFanSpeedVeryHigh);
  fuzzy->addFuzzyRule(fuzzyRule12);

  FuzzyRuleAntecedent *ifTemperatureVeryBigAndLdrIsLow = new FuzzyRuleAntecedent();
  ifTemperatureVeryBigAndLdrIsLow->joinWithAND( verybig, lowldr);
  thenFanSpeedVeryHigh->addOutput(veryhighb);
  FuzzyRule *fuzzyRule13 = new FuzzyRule(13, ifTemperatureVeryBigAndLdrIsLow, thenFanSpeedVeryHigh);
  fuzzy->addFuzzyRule(fuzzyRule13);

  FuzzyRuleAntecedent *ifTemperatureVeryBigAndLdrIsMid = new FuzzyRuleAntecedent();
  ifTemperatureVeryBigAndLdrIsMid->joinWithAND( verybig, midldr);
  thenFanSpeedVeryHigh->addOutput(veryhighb);
  FuzzyRule *fuzzyRule14 = new FuzzyRule(14, ifTemperatureVeryBigAndLdrIsMid, thenFanSpeedVeryHigh);
  fuzzy->addFuzzyRule(fuzzyRule14);

  FuzzyRuleAntecedent *ifTemperatureVeryBigAndLdrIsHigh = new FuzzyRuleAntecedent();
  ifTemperatureVeryBigAndLdrIsHigh->joinWithAND( verybig, highldr);
  thenFanSpeedVeryHigh->addOutput(veryhighb);
  FuzzyRule *fuzzyRule15 = new FuzzyRule(15, ifTemperatureVeryBigAndLdrIsHigh, thenFanSpeedVeryHigh);
  fuzzy->addFuzzyRule(fuzzyRule15);

 FuzzyRuleAntecedent *ifhumidityVerySmall5 = new FuzzyRuleAntecedent();
  ifhumidityVerySmall5->joinSingle(verysmall5);
  FuzzyRuleConsequent *thenHumValveVeryHigh5 = new FuzzyRuleConsequent();
  thenHumValveVeryHigh5->addOutput(veryhighb5);
  FuzzyRule *fuzzyRule16 = new FuzzyRule(16, ifhumidityVerySmall5, thenHumValveVeryHigh5);
  fuzzy->addFuzzyRule(fuzzyRule16);

  FuzzyRuleAntecedent *ifhumiditySmall5 = new FuzzyRuleAntecedent();
  ifhumiditySmall5->joinSingle(small5);
  FuzzyRuleConsequent *thenHumValveHigh5 = new FuzzyRuleConsequent();
```

```
thenHumValveHigh5->addOutput(highb5);
FuzzyRule *fuzzyRule17 = new FuzzyRule(17, ifhumiditySmall5, thenHumValveHigh5);
fuzzy->addFuzzyRule(fuzzyRule17);

FuzzyRuleAntecedent *ifhumidityMid5= new FuzzyRuleAntecedent();
ifhumidityMid5->joinSingle(mid5);
FuzzyRuleConsequent *thenHumValveMid5 = new FuzzyRuleConsequent();
thenHumValveMid5->addOutput(midb5);
FuzzyRule *fuzzyRule18 = new FuzzyRule(18, ifhumidityMid5, thenHumValveMid5);
fuzzy->addFuzzyRule(fuzzyRule18);

FuzzyRuleAntecedent *ifhumidityBig5 = new FuzzyRuleAntecedent();
ifhumidityBig5->joinSingle(big5);
FuzzyRuleConsequent *thenHumValveLow5 = new FuzzyRuleConsequent();
thenHumValveLow5->addOutput(lowb5);
FuzzyRule* fuzzyRule19 = new FuzzyRule(19, ifhumidityBig5, thenHumValveLow5);
fuzzy->addFuzzyRule(fuzzyRule19);

FuzzyRuleAntecedent *ifhumidityVeryBig5 = new FuzzyRuleAntecedent();
ifhumidityVeryBig5->joinSingle(verybig5);
thenHumValveLow5->addOutput(lowb5);
FuzzyRule* fuzzyRule20 = new FuzzyRule(20, ifhumidityVeryBig5, thenHumValveLow5);
fuzzy->addFuzzyRule(fuzzyRule20);

ifhumidityVerySmall5->joinSingle(verysmall5);
FuzzyRuleConsequent *thenHumFanLow8 = new FuzzyRuleConsequent();
thenHumFanLow8->addOutput(lowb8);
FuzzyRule *fuzzyRule21 = new FuzzyRule(21, ifhumidityVerySmall5, thenHumFanLow8);
fuzzy->addFuzzyRule(fuzzyRule21);

ifhumiditySmall5->joinSingle(small5);
thenHumFanLow8->addOutput(lowb8);
FuzzyRule *fuzzyRule22 = new FuzzyRule(22, ifhumiditySmall5, thenHumFanLow8);
fuzzy->addFuzzyRule(fuzzyRule22);

ifhumidityMid5->joinSingle(mid5);
FuzzyRuleConsequent *thenHumFanMid8 = new FuzzyRuleConsequent();
thenHumFanMid8->addOutput(midb8);
```

```
  FuzzyRule *fuzzyRule23 = new FuzzyRule(23, ifhumidityMid5, thenHumFanMid8);
  fuzzy->addFuzzyRule(fuzzyRule23);

  ifhumidityBig5->joinSingle(big5);
  FuzzyRuleConsequent *thenHumFanHigh8 = new FuzzyRuleConsequent();
  thenHumFanHigh8->addOutput(highb8);
  FuzzyRule* fuzzyRule24 = new FuzzyRule(24, ifhumidityBig5, thenHumFanHigh8);
  fuzzy->addFuzzyRule(fuzzyRule24);

  ifhumidityVeryBig5->joinSingle(verybig5);
  FuzzyRuleConsequent *thenHumFanVeryHigh8 = new FuzzyRuleConsequent();
  thenHumFanVeryHigh8->addOutput(veryhighb8);
  FuzzyRule* fuzzyRule25 = new FuzzyRule(25, ifhumidityVeryBig5, thenHumFanHigh8);
  fuzzy->addFuzzyRule(fuzzyRule25);

}


  float temperature() {
     float temp = dht.readTemperature();
    return temp;
}
int intensity() {
  return analogRead(LDR);
}
float humidity() {
    float humidity = dht.readHumidity();
    return humidity;
}
void loop() {
  float temp = temperature();
  int light = intensity();
  float hum = humidity();
     delay(3000);
     lcd.clear();
     lcd.setCursor(0,0);
     lcd.print(T + temp );
     lcd.print(" C");
```

```
    delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(L + light );

    delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print( H + hum );

  if (temp < 0 || temp > 40 || light>600 ) return;
  if (hum < 0 || hum > 100 ) return;
 fuzzy->setInput(1, temp);
 fuzzy->setInput(2, light);
 fuzzy->setInput(3, hum);

 fuzzy->fuzzify();

 int output1 = fuzzy->defuzzify(1);
 int output2 = fuzzy->defuzzify(2);
 int output3 = fuzzy->defuzzify(3);
 analogWrite(FAN, output1);
 analogWrite(Valve3, output2);
 analogWrite(EFan, output3);
}
```

Combined code for pH level and Water Temperature Fuzzy Logic Controllers

```
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>
#include <Arduino.h>
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
#include <OneWire.h>
#include <DallasTemperature.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#define Valve1 5
#define ONE_WIRE_BUS 13
#define Coil 6
int pHSense = A0;
int samples = 10;
float adc_resolution = 1024.0;

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

String water = "WaterTemp: ";

Fuzzy *fuzzy = new Fuzzy();
void setup() {
  lcd.clear();
  lcd.init();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("HYDROPONIC");
  Serial.begin(9600);
  analogWrite(6,100);
  sensors.begin();

  pinMode(pHSense, INPUT);
  pinMode(Valve1, OUTPUT);
  pinMode(ONE_WIRE_BUS, INPUT);
  pinMode(Coil, OUTPUT);

  FuzzySet *verysmall2 = new FuzzySet(0, 0, 0, 2);
  FuzzySet *small2 = new FuzzySet(2, 3, 3, 4);
  FuzzySet *mid2 = new FuzzySet(2, 6, 6, 10);
  FuzzySet *big2 = new FuzzySet(4, 8, 8, 12);
  FuzzySet *verybig2 = new FuzzySet(10, 12, 12, 14);
```

```
FuzzySet *lowb2 = new FuzzySet(0, 50, 50, 100);
FuzzySet *midb2 = new FuzzySet(50, 100, 100, 150);
FuzzySet *highb2 = new FuzzySet(100, 150, 150, 200);
FuzzySet *veryhighb2 = new FuzzySet(150, 200, 200, 255);

FuzzyInput *pHstate = new FuzzyInput(1);
pHstate->addFuzzySet(verysmall2);
pHstate->addFuzzySet(small2);
pHstate->addFuzzySet(mid2);
pHstate->addFuzzySet(big2);
pHstate->addFuzzySet(verybig2);
fuzzy->addFuzzyInput(pHstate);

FuzzyOutput *AlbertValve = new FuzzyOutput(1);
AlbertValve->addFuzzySet(lowb2);
AlbertValve->addFuzzySet(midb2);
AlbertValve->addFuzzySet(highb2);
AlbertValve->addFuzzySet(veryhighb2);
fuzzy->addFuzzyOutput(AlbertValve);

 FuzzySet *verysmall4 = new FuzzySet(0, 5, 5, 10);
FuzzySet *small4 = new FuzzySet(5, 10, 10, 15);
FuzzySet *mid4 = new FuzzySet(10, 20, 20, 30);
FuzzySet *big4 = new FuzzySet(20, 30, 30, 40);
FuzzySet *verybig4 = new FuzzySet(30, 45, 45, 60);

FuzzySet *lowb4 = new FuzzySet(0, 50, 50, 100);
FuzzySet *midb4 = new FuzzySet(50, 100, 100, 150);
FuzzySet *highb4 = new FuzzySet(100, 150, 150, 200);
FuzzySet *veryhighb4 = new FuzzySet(150, 200, 200, 255);

FuzzyInput *waterTemperature = new FuzzyInput(2);
waterTemperature->addFuzzySet(verysmall4);
waterTemperature->addFuzzySet(small4);
waterTemperature->addFuzzySet(mid4);
waterTemperature->addFuzzySet(big4);
waterTemperature->addFuzzySet(verybig4);
fuzzy->addFuzzyInput(waterTemperature);
```

```
FuzzyOutput *Coilheat = new FuzzyOutput(2);
Coilheat->addFuzzySet(lowb4);
Coilheat->addFuzzySet(midb4);
Coilheat->addFuzzySet(highb4);
Coilheat->addFuzzySet(veryhighb4);
fuzzy->addFuzzyOutput( Coilheat);

FuzzyRuleAntecedent *ifpHstateVerySmall2 = new FuzzyRuleAntecedent();
ifpHstateVerySmall2->joinSingle(verysmall2);
FuzzyRuleConsequent *thenAlbertValveLow2 = new FuzzyRuleConsequent();
thenAlbertValveLow2->addOutput(lowb2);
FuzzyRule *fuzzyRule1 = new FuzzyRule(1, ifpHstateVerySmall2, thenAlbertValveLow2);
fuzzy->addFuzzyRule(fuzzyRule1);

FuzzyRuleAntecedent *ifpHstateSmall2 = new FuzzyRuleAntecedent();
ifpHstateSmall2->joinSingle(small2);
thenAlbertValveLow2->addOutput(lowb2);
FuzzyRule *fuzzyRule2 = new FuzzyRule(2, ifpHstateSmall2, thenAlbertValveLow2);
fuzzy->addFuzzyRule(fuzzyRule2);

FuzzyRuleAntecedent *ifpHstateMid2 = new FuzzyRuleAntecedent();
ifpHstateMid2->joinSingle(mid2);
FuzzyRuleConsequent *thenAlbertValveMid2 = new FuzzyRuleConsequent();
thenAlbertValveMid2->addOutput(midb2);
FuzzyRule *fuzzyRule3 = new FuzzyRule(3, ifpHstateMid2, thenAlbertValveMid2);
fuzzy->addFuzzyRule(fuzzyRule3);

FuzzyRuleAntecedent *ifpHstateBig2 = new FuzzyRuleAntecedent();
ifpHstateBig2->joinSingle(big2);
FuzzyRuleConsequent *thenAlbertValveHigh2 = new FuzzyRuleConsequent();
thenAlbertValveHigh2->addOutput(highb2);
FuzzyRule* fuzzyRule4 = new FuzzyRule(4, ifpHstateBig2, thenAlbertValveHigh2);
fuzzy->addFuzzyRule(fuzzyRule4);

FuzzyRuleAntecedent *ifpHstateVeryBig2 = new FuzzyRuleAntecedent();
ifpHstateVeryBig2->joinSingle(verybig2);
FuzzyRuleConsequent *thenAlbertValveVeryHigh2 = new FuzzyRuleConsequent();
thenAlbertValveVeryHigh2->addOutput(veryhighb2);
```

```
  FuzzyRule* fuzzyRule5 = new FuzzyRule(5, ifpHstateVeryBig2, thenAlbertValveHigh2);
  fuzzy->addFuzzyRule(fuzzyRule5);

  FuzzyRuleAntecedent *ifwaterTemperatureVerySmall4 = new FuzzyRuleAntecedent();
  ifwaterTemperatureVerySmall4->joinSingle(verysmall4);
  FuzzyRuleConsequent *thenCoilheatVeryHigh4 = new FuzzyRuleConsequent();
  thenCoilheatVeryHigh4->addOutput(veryhighb4);
  FuzzyRule *fuzzyRule6 = new FuzzyRule(6, ifwaterTemperatureVerySmall4, thenCoilheatVeryHigh4);
  fuzzy->addFuzzyRule(fuzzyRule6);

  FuzzyRuleAntecedent *ifwaterTemperatureSmall4 = new FuzzyRuleAntecedent();
  ifwaterTemperatureSmall4->joinSingle(small4);
  FuzzyRuleConsequent *thenCoilheatHigh4 = new FuzzyRuleConsequent();
  thenCoilheatHigh4->addOutput(highb4);
  FuzzyRule *fuzzyRule7 = new FuzzyRule(7, ifwaterTemperatureSmall4, thenCoilheatHigh4);
  fuzzy->addFuzzyRule(fuzzyRule7);

  FuzzyRuleAntecedent *ifwaterTemperatureMid4= new FuzzyRuleAntecedent();
  ifwaterTemperatureMid4->joinSingle(mid4);
  FuzzyRuleConsequent *thenCoilheatMid4 = new FuzzyRuleConsequent();
  thenCoilheatMid4->addOutput(midb4);
  FuzzyRule *fuzzyRule8 = new FuzzyRule(8, ifwaterTemperatureMid4, thenCoilheatMid4);
  fuzzy->addFuzzyRule(fuzzyRule8);

  FuzzyRuleAntecedent *ifwaterTemperatureBig4 = new FuzzyRuleAntecedent();
  ifwaterTemperatureBig4->joinSingle(big4);
  FuzzyRuleConsequent *thenCoilheatLow4 = new FuzzyRuleConsequent();
  thenCoilheatLow4->addOutput(lowb4);
  FuzzyRule* fuzzyRule9 = new FuzzyRule(9, ifwaterTemperatureBig4, thenCoilheatLow4);
  fuzzy->addFuzzyRule(fuzzyRule9);

  FuzzyRuleAntecedent *ifwaterTemperatureVeryBig4 = new FuzzyRuleAntecedent();
  ifwaterTemperatureVeryBig4->joinSingle(verybig4);
  thenCoilheatLow4->addOutput(lowb4);
  FuzzyRule* fuzzyRule10 = new FuzzyRule(10, ifwaterTemperatureVeryBig4, thenCoilheatLow4);
  fuzzy->addFuzzyRule(fuzzyRule10);

}
```

```
float ph (float voltage) {
  return 7 + ((2.5 - voltage) / 0.18);
}
float waterTemperature() {
   sensors.requestTemperatures();
   float Wtemp = sensors.getTempCByIndex(0);
   return Wtemp;
}
void loop() {
  int measurings=0;
  for (int i = 0; i < samples; i++)
  {
    measurings += analogRead(pHSense);
    delay(10);
  }
  float voltage = 5 / adc_resolution * measurings/samples;
  float WT = waterTemperature();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("pH= ");
    lcd.print(ph(voltage));
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print( water + WT );
    delay(2000);

if (ph(voltage) < 0 || ph(voltage) > 14 ) return;
if (WT < 0 || WT > 60 ) return;

  fuzzy->setInput(1, ph(voltage));
  fuzzy->setInput(2, WT);
  fuzzy->fuzzify();
  int output1 = fuzzy->defuzzify(1);
  int output2 = fuzzy->defuzzify(2);
  analogWrite(Valve1, output1);
  analogWrite(Coil, output2);
}
```

Combined code for Water Level and CO2 Level Fuzzy Logic Controllers

```
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>
#include <Arduino.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

#define sensor A1
#define TRIGGER 4
#define ECHO 5
#define Valve2 6
#define PUMP 7
int gas, co2lvl;

Fuzzy *fuzzy = new Fuzzy();

String W = "waterLevel:";
int Height = 25;
String carbon = "CO2Level:";

void setup(){
  lcd.clear();
  lcd.init();
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("HYDROPONIC");

  Serial.begin(9600);
```

```
delay(1000);

 pinMode(sensor, INPUT);
 pinMode(TRIGGER, OUTPUT);
 pinMode(ECHO, INPUT);
 pinMode(Valve2, OUTPUT);
 pinMode(PUMP, OUTPUT);

  lcd.clear();
  lcd.print("Warming coil");

  for(int i =0;i<=100;i++)
  {
    lcd.setCursor(12,0);
    if (i<100) lcd.print(" ");
    if (i<10) lcd.print(" ");
    lcd.print(i);
    lcd.print("%");
    delay(1000);
  }
  FuzzySet *verysmall1 = new FuzzySet(0, 150, 150, 300);
  FuzzySet *small1 = new FuzzySet(100, 300, 300, 600);
  FuzzySet *mid1 = new FuzzySet(300, 600, 600, 900);
  FuzzySet *big1 = new FuzzySet(600, 900, 900, 1200);
  FuzzySet *verybig1 = new FuzzySet(900, 1200, 1200, 1500);

  FuzzySet *lowb1 = new FuzzySet(0, 50, 50, 100);
  FuzzySet *midb1 = new FuzzySet(50, 100, 100, 150);
  FuzzySet *highb1 = new FuzzySet(100, 150, 150, 200);
  FuzzySet *veryhighb1 = new FuzzySet(150, 200, 200, 255);

  FuzzyInput *AirState = new FuzzyInput(1);
  AirState->addFuzzySet(verysmall1);
  AirState->addFuzzySet(small1);
  AirState->addFuzzySet(mid1);
  AirState->addFuzzySet(big1);
  AirState->addFuzzySet(verybig1);
  fuzzy->addFuzzyInput(AirState);
```

```
FuzzyOutput *pumpSpeed = new FuzzyOutput(1);
pumpSpeed->addFuzzySet(lowb1);
pumpSpeed->addFuzzySet(midb1);
pumpSpeed->addFuzzySet(highb1);
pumpSpeed->addFuzzySet(veryhighb1);
fuzzy->addFuzzyOutput(pumpSpeed);

FuzzySet *verysmall3 = new FuzzySet(0, 2, 2, 4);
FuzzySet *small3 = new FuzzySet(2, 4, 4, 6);
FuzzySet *mid3 = new FuzzySet(4, 8, 8, 12);
FuzzySet *big3 = new FuzzySet(8, 13, 13, 18);
FuzzySet *verybig3 = new FuzzySet(13, 19, 19, 25);

FuzzySet *lowb3 = new FuzzySet(0, 50, 50, 100);
FuzzySet *midb3 = new FuzzySet(50, 100, 100, 150);
FuzzySet *highb3 = new FuzzySet(100, 150, 150, 200);
FuzzySet *veryhighb3 = new FuzzySet(150, 200, 200, 255);

FuzzyInput *waterlevel = new FuzzyInput(1);
waterlevel->addFuzzySet(verysmall3);
waterlevel->addFuzzySet(small3);
waterlevel->addFuzzySet(mid3);
waterlevel->addFuzzySet(big3);
waterlevel->addFuzzySet(verybig3);
fuzzy->addFuzzyInput(waterlevel);

FuzzyOutput *WaterValve = new FuzzyOutput(1);
WaterValve->addFuzzySet(lowb3);
WaterValve->addFuzzySet(midb3);
WaterValve->addFuzzySet(highb3);
WaterValve->addFuzzySet(veryhighb3);
fuzzy->addFuzzyOutput(WaterValve);


FuzzyRuleAntecedent *ifAirStateVerySmall1 = new FuzzyRuleAntecedent();
ifAirStateVerySmall1->joinSingle(verysmall1);
FuzzyRuleConsequent *thenpumpSpeedLow1 = new FuzzyRuleConsequent();
thenpumpSpeedLow1->addOutput(lowb1);
```

```
FuzzyRule *fuzzyRule1 = new FuzzyRule(1, ifAirStateVerySmall1, thenpumpSpeedLow1);
 fuzzy->addFuzzyRule(fuzzyRule1);

 FuzzyRuleAntecedent *ifAirStateSmall1 = new FuzzyRuleAntecedent();
 ifAirStateSmall1->joinSingle(small1);
 thenpumpSpeedLow1->addOutput(lowb1);
 FuzzyRule *fuzzyRule2 = new FuzzyRule(2, ifAirStateSmall1, thenpumpSpeedLow1);
 fuzzy->addFuzzyRule(fuzzyRule2);

 FuzzyRuleAntecedent *ifAirStateMid1 = new FuzzyRuleAntecedent();
 ifAirStateMid1->joinSingle(mid1);
 FuzzyRuleConsequent *thenpumpSpeedMid1 = new FuzzyRuleConsequent();
 thenpumpSpeedMid1->addOutput(midb1);
 FuzzyRule *fuzzyRule3 = new FuzzyRule(3, ifAirStateMid1, thenpumpSpeedMid1);
 fuzzy->addFuzzyRule(fuzzyRule3);

 FuzzyRuleAntecedent *ifAirStateBig1 = new FuzzyRuleAntecedent();
 ifAirStateBig1->joinSingle(big1);
 FuzzyRuleConsequent *thenpumpSpeedHigh1 = new FuzzyRuleConsequent();
 thenpumpSpeedHigh1->addOutput(highb1);
 FuzzyRule* fuzzyRule4 = new FuzzyRule(4, ifAirStateBig1, thenpumpSpeedHigh1);
 fuzzy->addFuzzyRule(fuzzyRule4);

 FuzzyRuleAntecedent *ifAirStateVeryBig1 = new FuzzyRuleAntecedent();
 ifAirStateVeryBig1->joinSingle(verybig1);
 FuzzyRuleConsequent *thenpumpSpeedVeryHigh1 = new FuzzyRuleConsequent();
 thenpumpSpeedVeryHigh1->addOutput(veryhighb1);
 FuzzyRule* fuzzyRule5 = new FuzzyRule(5, ifAirStateVeryBig1, thenpumpSpeedHigh1);
 fuzzy->addFuzzyRule(fuzzyRule5);

 FuzzyRuleAntecedent *ifwaterlevelVerySmall3 = new FuzzyRuleAntecedent();
 ifwaterlevelVerySmall3->joinSingle(verysmall3);
 FuzzyRuleConsequent *thenWaterValveVeryHigh3 = new FuzzyRuleConsequent();
 thenWaterValveVeryHigh3->addOutput(veryhighb3);
 FuzzyRule *fuzzyRule6 = new FuzzyRule(6, ifwaterlevelVerySmall3, thenWaterValveVeryHigh3);
 fuzzy->addFuzzyRule(fuzzyRule6);
```

```
  FuzzyRuleAntecedent *ifwaterlevelSmall3 = new FuzzyRuleAntecedent();
  ifwaterlevelSmall3->joinSingle(small3);
  FuzzyRuleConsequent *thenWaterValveHigh3 = new FuzzyRuleConsequent();
  thenWaterValveHigh3->addOutput(highb3);
  FuzzyRule *fuzzyRule7 = new FuzzyRule(7, ifwaterlevelSmall3, thenWaterValveHigh3);
  fuzzy->addFuzzyRule(fuzzyRule7);

  FuzzyRuleAntecedent *ifwaterlevelMid3 = new FuzzyRuleAntecedent();
  ifwaterlevelMid3->joinSingle(mid3);
  FuzzyRuleConsequent *thenWaterValveMid3 = new FuzzyRuleConsequent();
  thenWaterValveMid3->addOutput(midb3);
  FuzzyRule *fuzzyRule8 = new FuzzyRule(8, ifwaterlevelMid3, thenWaterValveMid3);
  fuzzy->addFuzzyRule(fuzzyRule8);

  FuzzyRuleAntecedent *ifwaterlevelBig3 = new FuzzyRuleAntecedent();
  ifwaterlevelBig3->joinSingle(big3);
  FuzzyRuleConsequent *thenWaterValveLow3 = new FuzzyRuleConsequent();
  thenWaterValveLow3->addOutput(lowb3);
  FuzzyRule* fuzzyRule9 = new FuzzyRule(9, ifwaterlevelBig3, thenWaterValveLow3);
  fuzzy->addFuzzyRule(fuzzyRule9);

  FuzzyRuleAntecedent *ifwaterlevelVeryBig3 = new FuzzyRuleAntecedent();
  ifwaterlevelVeryBig3->joinSingle(verybig3);
  thenWaterValveLow3->addOutput(lowb3);
  FuzzyRule* fuzzyRule10 = new FuzzyRule(10, ifwaterlevelVeryBig3, thenWaterValveLow3);
  fuzzy->addFuzzyRule(fuzzyRule10);
}

float Co2(){
  gas = analogRead(sensor);
  co2lvl = gas-120;
  co2lvl = map(co2lvl,0,1024,400,5000);
  return co2lvl;
}
```

```
int waterlevel() {
  digitalWrite(TRIGGER,LOW);
  delay(1);
  digitalWrite(TRIGGER,HIGH);
  delay(1);
  digitalWrite(TRIGGER,LOW);
  long t = pulseIn(ECHO,HIGH);
  long cm = t / 29 / 2;
  return (Height - cm);
}

void loop() {
 float Air = Co2();
 int WL = waterlevel();
 delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print( W + (Height-WL) );
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print( carbon + Air );

if((co2lvl >= 350)&&(co2lvl <= 1000))
  {
    lcd.setCursor(0,1);
    lcd.print("Good level");
  }
  else if((co2lvl >= 1000)&&(co2lvl <= 2000))
  {
    lcd.setCursor(0,1);
    lcd.print("Bad");
  }
if (Air < 0 || Air > 1600 ) return;
if (WL < 0 || WL > 25 ) return;
```

```
 fuzzy->setInput(1, Air);
 fuzzy->setInput(2, WL);
 fuzzy->fuzzify();

 int output1 = fuzzy->defuzzify(1);
 int output2 = fuzzy->defuzzify(2);

 analogWrite(PUMP, output1);
 analogWrite(Valve2, output2);

}
```