AUTOMATIC PLANT IRRIGATION AND MONITORING SYSTEM USING NODE MCU

ACYUTA KRISHNA RAO A/L SATINARAYANAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

AUTOMATIC PLANT IRRIGATION AND MONITORING SYSTEM USING NODE MCU

ACYUTA KRISHNA RAO A/L SATINARAYANAN

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Computer Networking) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I hereby declare that this project report entitled

AUTOMATIC PLANT IRRIGATION AND MONITORING SYSTEM USING NODE MCU

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : Date :

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of Computer Science (Computer Networking) With Honours.

SUPERVISOR : Date :

DEDICATION

To my beloved parents, friends and the individuals who guided and motivated me on my educational path here in Universiti Teknikal Malaysia Melaka (UTeM).

ACKNOWLEDGMENT

Thanks to God for his blessing who gives guidance, strength, mental power, security and abilities for giving us good health and simplify everything to complete this final year project. First of all, thanks to my beloved mother and sister who give full support with their uncounted prayers, encouragement, excitement and invaluable help. I might not be able to finish this final year project completely without their assistance.

Second, special thanks to my project supervisor Dr. Nurul Azma Binti Zakaria who supervise and give me the opportunity to complete my final year project. Not forgotten for all her guidance, advices, suggestions, supports, ideas and all comments that have helped to ensure this project to a successful completion. Next, I should like to thanks to all the lecturers most sincerely for the useful thoughts, advice and extra expertise that they have given to finalizing this project.

Other than that, I want to thank my friends especially my best friends and my housemates who give their opinions, suggestions and supports. Thank you for the friendship with me and pleasure-learning throughout the study at UTeM. Lastly, I excuse all the others who heled me complete this project in different ways.

ABSTRACT

Automatic plant irrigation system emphasizes in implementing an automated watering system which will affect the growth of plant and simultaneously maintain the level of soil moisture and temperature of the plant. The system incorporates the Node MCU paired with Blynk to monitor the automated watering system in ensuring the plant is not overwatered and vice versa through the usage of soil humidity and temperature sensors. The system also allows control for manual intervention through buttons that operate as switch while remote intervention through the Blynk application. The sensor data is transmitted to the Blynk cloud and exported as CSV files for data processing. The analysis to measure the reactivity of the plant irrigation system to different soil moisture and temperature levels using Fittonia Albivenis plant is studied. Smartphone device that runs the Blynk application must be connected to a mobile data or Wi-Fi to ensure smooth data transmission from the Node MCU microcontroller and Blynk cloud.

ABSTRAK

Sistem pengairan loji automatik menekankan dalam melaksanakan sistem penyiraman automatik yang akan menjejaskan pertumbuhan tumbuhan dan serentak mengekalkan tahap kelembapan tanah dan suhu tumbuhan. Sistem ini menggabungkan MCU Node yang dipasangkan dengan Blynk untuk memantau sistem penyiraman automatik dalam memastikan kilang tidak dikeringkan dan sebaliknya melalui penggunaan kelembapan tanah dan sensor suhu. Sistem ini juga membolehkan kawalan untuk campur tangan manual melalui butang yang berfungsi sebagai suis sementara campur tangan jauh melalui aplikasi Blynk. Data sensor dihantar ke awan Blynk dan dieksport sebagai fail CSV untuk pemprosesan data. Analisis untuk mengukur kereaktifan sistem pengairan tanaman kepada kelembapan tanah dan paras suhu yang berbeza menggunakan tumbuhan Fittonia Albivenis dikaji. Peranti ponsel pintar yang menjalankan aplikasi Blynk harus dihubungkan ke data seluler atau Wi-Fi untuk memastikan transmisi data yang lancar dari mikrokontroler MCU Node dan awan Blynk.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I: Introduction

1.1 Project Introduction

In the age of advanced electronics and technology, incorporating convenience would be an aid the daily lifestyle of humans. There is a need for many automated systems that are capable of replacing or reducing human effort in their daily activities and jobs (Duzic & Dumic, 2017) .

In the form of unique intersection between biological engineering and electronics, the solution requires only a little bit knowledge of electronics as well as that knowledge related to botany and plant physiology (Duzic & Dumic, 2017).

The task is to irrigate the system for the facility as an automated system and to obtain the statistics for a user. The technology software will use the Node MCU (Arduino) as a microcontroller to address this challenge. The main activities included a moisture sensor and a temperature sensor when the Node MCU is associated to provide the signal to dry or moist soil situation.

Once the sensor detects the soil, the Node MCU sends out the sign and the buzzer can start and then contact the relay to allow the water pump being launched and keep the soil moist. The system should also allow manual intervention of an operator to control a water pump. This manual intervention should be possible to be performed both locally and remotely via the Internet.

1.2 Background study

Automated plant watering system estimate and measure the existing plant and then supplies desired amount of water needed by that plant. It is minimizing the excess water use as well as keeping plants healthy. The continuous increasing demand of food requires the rapid improvement in food production technology

The constant exploitation of water from the soil lowers the level of water due to which a lot of land gradually appears in unirrigated areas. Another important reason is the unplanned use of water, which contributes to substantial waste of water. Plants can easily be tracked and controlled by an automated plant irrigation system, in which the DC motor is used, in modern irrigation systems. Arduino UNO is used for this implementation. (Shifa, 2018).

Arduino is a prototyping tool for electrical open-source electrics that consists of both the interactive development environment and the physical programmable circuit board IDE. The Arduino interface allows the user to execute this on the device, to write and transfer computer code to the Arduino Uno physical board. The Arduino IDE programming language is C++. The Arduino takes sensor inputs and monitors the output.

1.3 Problem Statement

The plant needs to be always sensitive to the needs of the consumer. It is essential to water the plants at the right rate for the plant. Nevertheless, this watering ritual many people forget. Working people always forget to water the plants because the timetable is strict. In fact, irrigation plants are repetitive repeated activities and can lead busy people to tire.

In fact, the critical amount of water required for a plant to recover the soil moisture appropriate for plants is not expected by men. Then the field is irrigated regularly, the plant always dies. This may be attributed to over-watering or loss of the farm. Persons with no watering habits always have a concern.

The manual watering of plants does provide an advantage to people in the contexts of saving money, water, labour conservation and overall convenience when compared to big scales such as agriculture. The wastage produced from this directly impacts on a large scale.

Although other alternatives exist, such as regularly employing a gardener to water the plant, it costs money. In addition, if the owners are not present, there is a problem leaving a third person inside their home or premise. This increases the owner's security worries about the safety of their estate or premises, and without the owner being supervised is hard to trust gardeners or employees.

Table 1.1 Summary of Problem Statement

PS	Problem Statement
P1	Busy people always forget to water the plants due to tight schedule, especially those who constantly travel.
P2	People are not able to predict the essential amount of water needed to restore the soil moisture level in the plants.
P3	Concerns over safety of their premises, and without the owner supervising is hard to trust gardeners or employees.

1.4 Research Question

Problem question is use to distinguish each problem that being stated. Below elaborates the research questions based on the problem statement above.

1. ***Is there a significant impact of an automatic plant watering system for people by using an Arduino Node MCU Microcontroller?***

   The study is mainly focused restoring moisture levels to plants on an automated basis which will directly affect the periodic watering of plants manually. This project will study on the impact of the automatic plant watering system on their daily task execution.

2. ***Is there a significant relationship by implementing an auto watering system using soil moisture and temperature sensors?***

   The study emphasizes on implementing and ensuring an automated watering system which will positively affect the growth of plant and simultaneously maintaining the soil moisture and monitoring the temperature.

3. ***Is there a significant relationship between the performance of plant moisture and monitoring the system using Arduino Node MCU with Blynk?***

By incorporating the Arduino Node MCU paired with Blynk to monitor the automated watering system and assesing the plant moisture at the same time ensuring the plant is not overwatered and vice versa.

Table 1.2 Summary of Project Question

PS	PQ	Project Question
P1	**Q1**	Is there a significant impact of an automatic plant watering system for people by using an Arduino Node MCU Microcontroller?
P2	**Q2**	Is there a significant relationship by implementing an auto watering system using soil moisture and temperature sensors?
P3	**Q3**	Is there a significant relationship between the performance of plant moisture and monitoring the system using Arduino Node MCU with Blynk

1.5 Research Objective

The objectives of the project is to determine what needs to be done. To order to achieve the goals of the project, the problem statement and also the project concern of this project as defined in the previous paragraph should be seen. These are three objectives which in this project need to be achieved:

1. ***To develop an automatic plant watering system for people by using an Arduino Node MCU Microcontroller.***

   The objective to lessen the hassle for manual watering for plants for people is the emphasis on why an automated plant watering system is designed by incorporating the Arduino Node MCU Microcontroller to measure the efficiency.

2. ***To implement an auto watering system using soil moisture and temperature sensors.***

   In efforts to establish an automated watering system without disregarding the maintenance of soil and temperature to ensure the plan benefits holistically.

3. ***To analyse the performance of plant moisture and monitoring the system using Arduino Node MCU with Blynk.***

   Understanding the relationship betwen moisture and temperature and how it directly affects plants well being.

Table 1.3 Summary of Project Objectives

PS	PQ	PO	Project Objective
PS1	**PQ1**	**PO1**	To develop an automatic plant watering system for people by using an Arduino Node MCU Microcontroller
PS2	**PQ2**	**PO2**	To implement an auto watering system using soil moisture and temperature sensors.
PS3	**PQ3**	**PO3**	To analyse the performance of plant moisture and monitoring the system using Arduino Node MCU with Blynk.

1.6 Research Scope

This project is an improvement of conventional method of watering plants to the auto watering system. The auto watering system development is divided into software and hardware implementation and the development of this system is done through an Arduino Node MCU Board and with the conventional Arduino programming language Java.

It also requires input sensors like soil moisture sensor, soil temperature and humidity sensor DHT22, waterproof soil temperature DS18B20. Buttons like ON/OFF to control the irrigation of the pump. Water pump and relay actuators to control the flow of water from the reservoir. This system can be integrated with smart home utilities, be extended to the enterprise level usage since it is fully automated and is connected to a cloud monitoring system.

1.7 Project Contribution

Project contribution defines the expected from this project and significant contribution of this project. This project should produce a dedicated automatic plant watering system for people who look to have advanced gardening care system. This project also offer solution to analyse the performance of plant moisture and monitoring the system using Arduino Node MCU.

1.8 Report Organization

There are five chapters in this thesis which are included of introduction, literature review, methodology, result and discussion and finally a conclusion and recommendation. Each chapter will discuss its own aspects that related to the project.

Chapter I: Introduction

This chapter introduces the project which consists of introduction, background of the project. It explains briefly about the problem statement, objective, scope of the project, the expected outcome and conclusion to make sure the project can be understood easily.

Chapter II: Literature Review

This chapter focuses on the explanation of the reading material and publish thesis. Proceeding with where previous studies are reviewed. This chapter is discussing about the approaches and methods used in previous studies. The comparison of strength and weakness can be used as the guidelines to develop an efficient automatic watering system.

Chapter III: Project Methodology

Chapter three focuses on the methodology and approaches on the project. This includes the software implementation and hardware development of the project. This chapter also focuses on the project method which explain each stage of the selected methodology like waterfall methodology. The chapter also describe the milestone of this project.

Chapter IV: Analysis and Design

The problem research, analysis methodology and the nature of the project are discussed in this chapter. For the problem analysis, all the problem description in chapter I addressed was based on the two sides of the project research criteria, which are hardware and software. The conceptual design of the project and process flow also in this chapter are added for the design of the project.

Chapter V: Implementation

In this chapter focus on how the project is implemented. It shows how the project is gathered, installed, configured and setup according to the design and structure created in the chapter 4. The entire process of installation is showed and documented in detail here.

Chapter VI: Testing

The project development schedule, prototype design and test results, and project test plan are discussed in this chapter. The study schedule is made up of the test organisation, test environment and test schedule. The above is defined in the test data. The test design. To conclude the test phase, the test results are evaluated.

Chapter VII: Conclusion

This chapter discusses about the summary of the project. The conclusion includes the observation on weaknesses and strengths, proposition for improvement which are future work, project contribution and the limitation of the project

1.9 Conclusion

Briefly, from this chapter discussed about the Internet of Things (IoT) and the implementation of IoT inside agriculture and home farming environment. The study emphasizes on implementing and ensuring an automated watering system which will positively affect the growth of plant and simultaneously maintaining the soil moisture and monitoring the temperature. By incorporating the Arduino Node MCU paired with Blynk to monitor the automated watering system and assessing the plant moisture at the same time ensuring the plant is not overwatered and vice versa. In this chapter was explained the background project, problem statement, objective, scope and the summary. The next chapter being chapter 2 will discuss about literature review of the project.

Chapter II: Literature Review

2.1 Introduction

This chapter discusses on the related topics concerning the mentioned project objectives. This literature review functions in giving a depth to understanding the project. The review will be divided into two parts that will discuss on the software and the hardware implemented in the project build and its application. Next will be the review on the Arduino Uno and Node MCU microcontroller. The following subtopics will elaborate further on the project application. The table shows the chapter 2 overview.

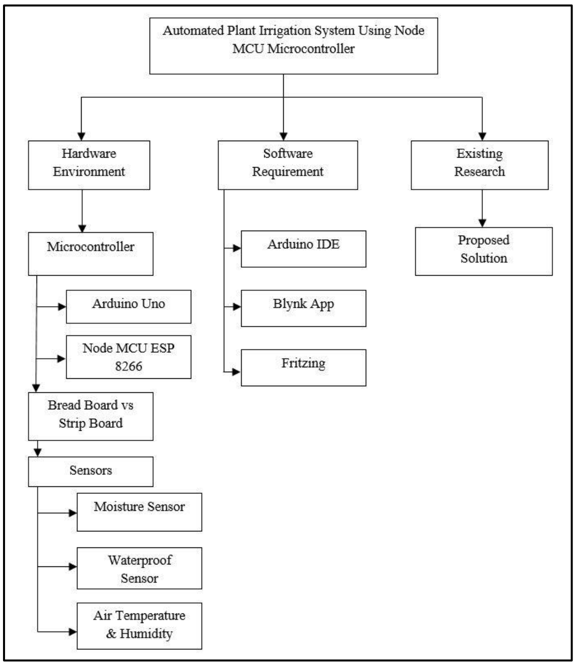


Figure 2.1 Literature Review Overview

2.2 Hardware Review

2.2.1 Arduino

The included development environment of Arduino software contains a textual content editor specified for writing code, message location, textual content console, toolbar with buttons for common features and menu series. 3D printer electronics are Arduino compatible and they utilise Atmega chip to empower the transfer in code utilizing the Arduino (Meijer 2014).

The programs written using Arduino software (IDE) are designed to facilitate drawings. Such illustrations are inside the text editor and are retained with the extension of the script. In addition to full errors and other information, the console shows the output of text material using the Arduino software program (IDE). The lowest right nook in the window shows the monitor and serial port that are installed. You can confirm and attach programs, make, open and manage drawings and open the serial monitor with toolbar buttons.

2.2.2 Arduino and Node MCU

The programming of NodeMCU with its IDE is almost as routine as an Arduino. Every new "custom firmware," including the initial firmware installed at the factory where AT commands are widely used, would overwrite all contained in the flash memory of the processor. While SDK from the company can be used for custom firmware creation, the good old Arduino IDE is much simpler to use. Figure 2.2 shows the Node MCU microcontroller board.

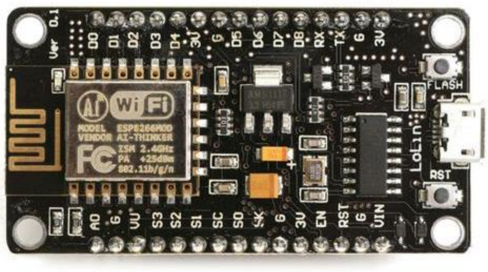


Figure 2.2 Node MCU ESP8266 Microcontroller

Arduino processor boards are excellent components microcontroller. Both of this microcontroller has their own character which is an Arduino microcontroller motherboard. A microcontroller is an easy laptop which could run one software at a time, again and again once more. Table 2.1 shows the specification comparison between the Arduino Uno and node MCU microcontroller.

Table 2.1 Arduino Uno and Node MCU comparison table.

	Arduino Uno	Node MCU
Price	RM50	RM35
Size	7.6x1.9x6.4 cm	49 × 24.5
Memory	0.002 MB	4MB ROM
Network	No	Yes
Multitasking	No	Yes
Voltage Input	7 to 12 V	4.5V to 9V
USB	Only one input	Two, peripherals OK
Operating System	No	No
Integrated Development Environment	Arduino	Arduino

2.3 Sensors

2.3.1 Soil Moisture Sensor

There are two channels in the soil moisture sensor which are used to measure the amount of water in the soil. This leads allow the current to move through the soil and determine in return the resistance to measure the level of moisture (Bisen, 2018). Figure 2.3 shows the soil moisture sensor.

Figure 2.3 Soil Moisture Sensor (Bisen, 2018)

When more water is present in the soil, more electricity is produced by the soil, which means less resistance and high moisture. If there is less surface water, the soil can contribute to less energy, which means high resistance and low humidity.

2.3.2 Waterproof temperature sensor DS18B20

For a two-way communication with a microcontroller, the waterproof temperature sensor needs just one wireless pin. The sensor is fairly accurate and does not need additional parts. It can measure temperatures from **-55°C to +125°C** with ±0.5°C Accuracy (Bisen, 2018). Figure 2.4 shows the waterproof temperature sensor DS18B20.

Figure 2.4 Waterproof Temperature Sensor DS18B20 (Bisen, 2018)

The temperature sensor resolution is 9, 10, 11 or 12 bits user-configurable. The sensor can be operated from 3V to 5.5V and only absorbs 1mA for successful conversions to temperature.

2.3.3 Temperature and Humidity Sensor DHT22

DHT22 digital signal calibration output. The DHT 22 is ideal for all kinds of tough applications thanks to its small size, low consumption and long transmission range (20 m). Single-row with four pins, which makes the connection very easy. Figure 2.5 shows the DHT22 temperature and humidity sensor.

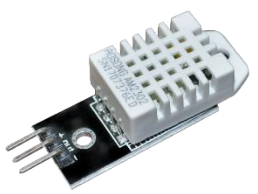


Figure 2.5 Temperature and Humidity Sensor DHT22

2.4 Board

2.4.1 Breadboard and Strip Board

There are different kinds of boards to create prototypes in electronics. Board is the medium for prototyping through construction of robots or any other electronic task. Many projects have a bread board in place of a panel because the part such as LEDs and resistors must not be soldered to the panel but only placed into holes designed according to schemas. It can prevent damage to the leg component and prevent it to build other prototypes using this type of breadboard. Figure 2.6 shows the breadboard.

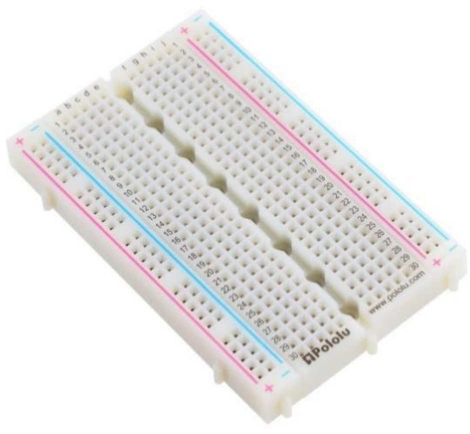


Figure 2.6 Breadboard

2.5 Software Review

2.5.1 Arduino IDE

Arduino IDE is used for microcontroller integration of programs. Source code is an important part of the automated watering system's entire process. The Arduino consists of java programming language and Arduino IDE also has a library called a C / C++, commonly known as a wiring. The Box Arduino typically includes a USB downloader that connects directly the flash program to the microcontroller.

2.5.2 Blynk

The Blynk mobile application serves as a control panel for your hardware visualization and control. The app offers a highly productive interface and numerous widgets for various purposes. Blynk is a forum for Internet of Things designed to quickly and easily develop and implement intelligent IoT tools. Figure 2.7 shows the Blynk application.

Figure 2.7 Blynk Application

It can be used for the remote reading, saving and simulation of sensor data and hardware. Blynk is an IoT platform application which is installed on a user's smartphone to send notifications about watering device status whether it is activated or deactivated.

2.6 Growth Requirements for the Plant

Indoor gardening could be the best option for a new starter in gardening. In indoor gardening, there is a lot of benefit that the planter can control and track the plant's growth to grow healthy and prosper. Furthermore, the soil used for growing can be monitored and controlled so that it is always moist and well maintained.

The right tools and equipment will help protect the plant and provide what the plant needs to grow. To ensure the plant grows safely and prosperously, there are so many requirements. The surrounding temperature, fertilizer, soil moisture, surrounding humidity and adequate lighting are some of the factors which influence plant growth. (of, 2015).

2.6.1 Fittonia Albivenis

The Fittonia albivenis is a plant species native to Colombia, Peru, Bolivia, Ecuador, and northern Brazil, Fittonia albivenis has been a flowering plant of the family Acanthaceae An herbaceous vine, with strong contrasting white or red leaves, it is noteworthy for its dark green foliage. It's often referred to as herb of the nerve or mosaic. It must be cultivated as a household in temperate regions with a temperature below 10 ° C (50 ° F). Figure 2.8 shows the Fittonia Albivenis plant.

Figure 2.8 Fittonia Albivenis Plant

It has to be regularly watered. It is considered to be "faint" without water for a few days but is quickly resuscitated by a quick watering and its health resumes. Fittonia albivenis is considered to be difficult to grow, so it is better bought in a nursery than cared for. It is suitable for groundcover because of its spreading habit.

2.6.2 Sufficient Lighting

The indoor gardening plant requires 300 lux to 500 lux of light to thrive. The plant will be tiny if there is not enough light and it will lose its flavour. To conduct photosynthesis, the plants require sunlight. There are two photosynthesis stages that are dark reaction and light reaction. In the dark reaction, the carbon dioxide from carbohydrates and hydrogen from the light reaction will be mixed while the energy from the sunlight is consumed and separated into hydrogen and oxygen in the light reaction (of, 2015). Figure 2.9 shows the photosynthesis and respiration process for day and night.

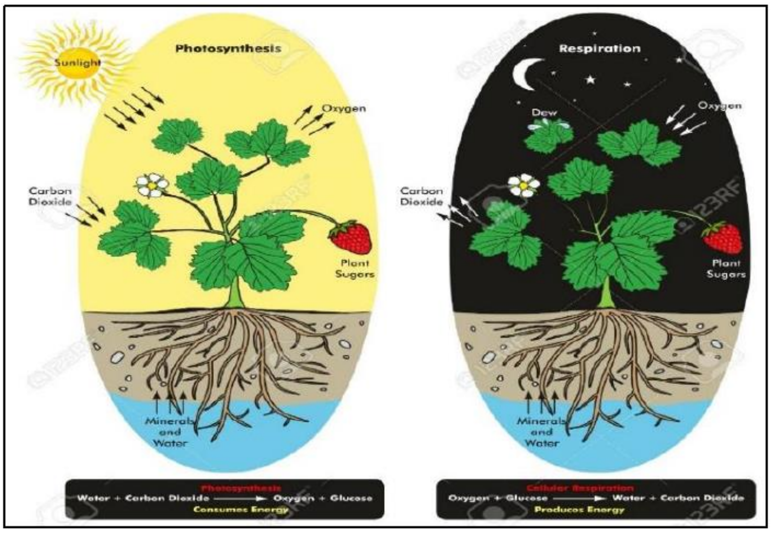


Figure 2.9: Photosynthesis process and Respiration process for day and night (of, 2015)

In order to maintain the quality, quantity and duration of light, consideration should be given to lighting the plants. Plant without enough light will be light, lighter green leaves and smaller leaves, while plants with sufficient light will quickly produce new leaves with dark green leaves and short internodes.

The florescent light can be used for indoor gardening to start seedlings in additional light or back up light to supplement daylight. Compared to other light form, it is more effective. When the light emitted from the warm object is produced at the light coating, a bright white coloured light is produced. The light should be no more than 30 cm from the plants and the light use should be between 20 watts and 40 watts. Figure 2.10 shows the distance from the plant to the light (of, 2015)

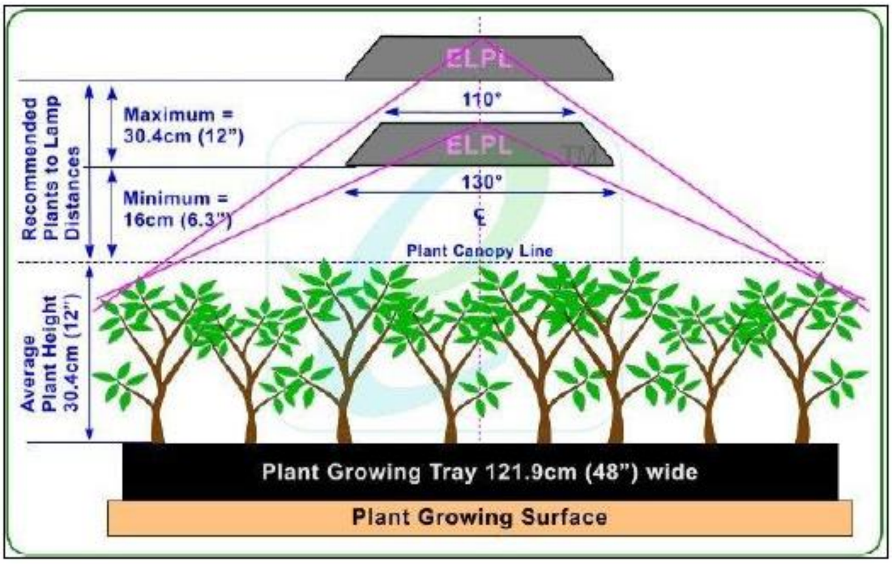


Figure 2.10: Distance from the plant to the light

2.6.3 Temperature

Temperature is one of the most significant growing and thriving requirements for the crop. As the temperature changes continuously between the plants and the surrounding air, the plant temperature is generally the same as the surrounding temperature. Plant temperature acceptance ranges vary significantly. All plants generally have a range of tolerance depending on their age, condition and water balance to different temperatures. The standard indoor gardening temperature needed by plants is between 25°C and 33°C.

The plant is more easily damaged when the plant is affected by high temperature because the temperature is higher than the maximum temperature for growth. The effect of high temperature on growth was associated with the influence of other environmental factors, specifically sufficient light. Usually high temperature causes damage to the sunscald seen in the sunlight.

The plant also can damage due to low temperature surrounding the plant because the temperature surrounding the plant are at below freezing points. This causes a variety of damage to the plant (of, 2015) (McConnell).

2.6.4 Relative Humidity

The relative humidity is the amount of humidity in the air. The humidity above 50% is high, 40% to 50% is medium, and 20% is considered low. It's very important and easily overlooked factor. There are two situations concerning relative humidity that should be noted. First, the less humidity, the more water from the leaf is lost. Below is the comparative figure between the 10% lower humidity and the 50% higher humidity where both plants obtain the same 25°C temperature (McConnell). Figure 2.11 shows the different relative humidity with same temperature.

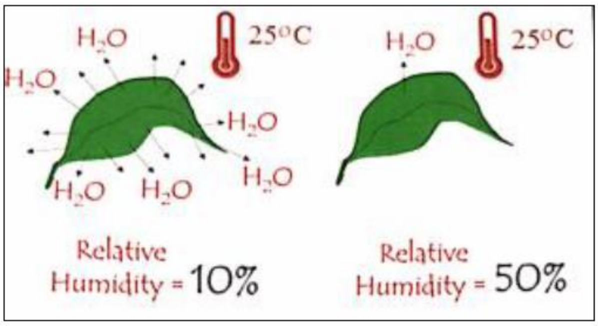


Figure 2.11. The different relative humidity with same temperature.

In the second place, the higher the temperature, the more moisture from the crop will be lost and the more water vapour the air will produce. Below is the comparative figure between high temperature of 32°C and low temperature of 25°C, where both obtain 50% of the same relative humidity (McConnell). Figure 2.12 shows the different temperature with same relative humidity.

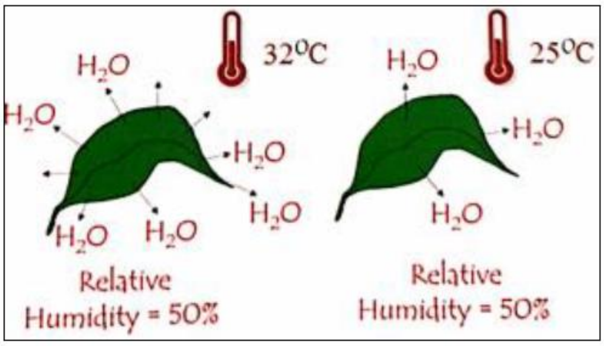


Figure 2.12. The different temperature with same relative humidity

2.6.5 Water

The indoor gardening required regular watering. This is because the plant's roots are restricted to water in the pot. When the ground is dry, it indicates that the plants need water. The indication to avoid watering in the pot is when the water begins to flow out under the former. Improper irrigation causes a lot of trouble (McConnell).

2.6.6 Fertilizer

It is important for plants to use fertilizers correctly because it influences revenue and bioactive ingredients. The fertilizer can be used to stimulate plant growth by spraying on plant leaves, putting it on the soil, or applying it directly on the roots. There are two category of fertilizer which is both natural fertilizer and chemical fertilizer. Organic fertilizer is recommended to use to plant (of, 2015).

2.7 Previous Research of Existing System

2.7.1 Precise automated water dispenser for potted plants

A Dyck study shows that, until the pot and plant reach their original pot weight, one common method to return moisture tension to the desired level is to add water periodically to the container.

This method requires manually measuring or adding water until the initial weight is met. Changes in plant weight as growth occurs may be needed to change initial weight. (F.B, 1977).

The study proposes the idea of an apparatus which dispenses water automatically into the pot restoring it to the set weight (F.B, 1977). This reduces substantially the time needed to water multiple potted plants. The system was originally meant to return a wide range of potted plants with an equivalent original weight to moisture levels every day without testing the water used.

2.7.2 Controlled Irrigation System based of Microcontoller for Plantation

Research by Kumbhar et.al in efforts to study the irrigation system in India are operated manually. The study suggested an automated concept of irrigation to use water efficiently and effectively.

The thesis proposed to build a framework that would develop a software program for reading data on channels in assembly language. When the data is small, the controller must set the motor to ON state by transmitting the relay signal. The motor with a pipe is attached with a different valve to each sector. The valve is opened simultaneously with data for the stepper motor. This process continues as long as there are high channel results. When channel data are small, the next channel will be screened and the cycle will continue. This system helps provide plants with water when moisture falls below the stated value (Khumbar & P.Ghatule, 2013).

The findings of the study revealed that developed microcontroller based irrigation system can work constantly for indefinite time period, even in certain abnormal circumstances (Agrawal, Dewangan, Deshmukh, Yadav, & Chandrakar, 2018). If water is supplied to the plants in the right time, productivity is increased by 25% to 30%. This machine can be used to irrigate very large areas, only to split the entire country into several sectors and the whole process can be managed by a single microcontroller. It saves energy, time and costs for people. (Khumbar & P.Ghatule, 2013).

2.7.3 Automatic Irrigation Based on Monitoring Plant Transpiration

In a separate study conducted by Ayoade et al, an automatic irrigation to monitor plant transpiration was researched incorporating micro irrigation as an artificial means of supplying water to the root of plant.

In dry areas and times of insufficient rainfall, drainage was used to promote the planting of agricultural crops, the preservation of the landscape and the reverberation of disturbed soils. Irrigation helps protect plants from frost, suppresses the growth of weeds in grain fields and avoids erosion of soils. The old irrigation system was the use of irrigation bottles, water channels to open and closed by hand, or sprinklers for the rucksack. Most water is being lost in this situation. (Agbetuyi, Awelewa, Evwieroghene, & Tita, 2017).

In attempts to improve the existing and old forms of irrigation, an automated irrigation system is necessary to curb wastage of water in the process. In the study, the automatic irrigation system was designed to continuously sense the moisture level of the soil (Agbetuyi, Awelewa, Evwieroghene, & Tita, 2017). The machine correctly responds by watering the soil with the exact amount of water needed and then shuts the supply down when the necessary soil moisture level is reached.

The findings produced that, the system helps to eliminate the stress of manual irrigation and irrigation control while at the same time conserving the available water supply.

2.7.4 Plant Watering Autonomous Robot

According to Adeoudu et al, An intelligent watering system was able to avoid obstacles on its way to a semi-structured environment. The robot architecture combines the module moisture sensor with the infrared moisture sensor and an interface between the Arduino-Xbee boards to avoiding obstruction. The robot's mobility is only limited to push the direction forward. The research is for experimentation in the field only.

The system was designed in such a way where the plant irrigation mobile robot is designed to be able to water a given area of land without human intervention. The robot action is divided into two major phases, which are moisture sensing phase and irrigation phase, implemented by two different modules (Adeodu, 2019). The measurement of moisture is accomplished with the soil sensorYL-69, powered by the arduino platform with ATmega328 microcontroller.

DC motors and water pump relay integrated with another Arduino microcontroller monitor the irrigation and mobility of the robot for the watering operation, based on soil humidity info. The operation of the robot can be separated into two plant watering activities and orientation tasks of the robot. Table 2.2 shows the comparison between all the reviewed studies.

No	Name of Author	Name of Project	Board Used	Sensors Used	OLED	Cloud	Mobile Interface
1.	Nermin Đuzić1, Dalibor Đumić2	Automatic Plant Watering System via Soil Moisture Sensing by means of Suitable Electronics. (2017)	Arduino Nano Arduino IDE	Moisture	No	No	No
2.	American Journal of Engineering Research (AJER) Tasneem Khan Shifa.	Moisture Sensing Automatic Plant Watering System Using Arduino Uno (2018)	Arduino Uno Arduino IDE	Moisture	No	No	No
3.	Jacquline M.S. Waworundeng, Novian Chandra Suseno, Roberth Ricky Y Manaha.	Automatic Watering System for Plants with IoT Monitoring and Notification. (2018)	WeMos D1 (Arduino Uno) Arduino IDE	Moisture YL-69	No	Thingspeak IoT Monitoring	Blynkk
4.	Adeoudu A.O, Bodunde O.P, Daniyan I A, Omitola O.O, Akinyoola J.O, Adie U.C	Development of an autonomous mobile plant irrigation robot for semi structured environment.	Vero Board	Moisture	1602 Liquid Crystal Display (LCD)	Monitoring and Graph	Xbee Series 1
5.	Lokesh Agrawal, Bhitika Dewangan, Shilpi Desmukh, Priyanka Yadav, Mukesh Kumar Chandrakar	Smart Irrigation System Using 8051 Microcontroller and Fencing Control Using Bluetooth Module.	Arduino Uno	Moisture	Bluetooth text command	Monitoring	Interfaced GSM modem

6.	Noor Sarah Binti Mohd Shah, UTeM.	Auto Watering System Using Arduino. (2015)	Arduino Uno Arduino IDE	Moisture	No	No	No
7.	SITI NURATIQAH ALIANA BT ABIDIN, UteM.	DEVELOPMENT OF PLANT MOISTURE MONITORING SYSTEM USING ARDUINO	Arduino Uno Arduino IDE	Moisture Temperature	No	No	No
8.	S.R Khumbar, Arjun P Ghatule.	Microcontroller based Controlled Irrigation System for Plantation.	PIC Microcontroller	Yes, HIH 4030 Humidity Sensors	No	Data reading and monitoring	Interfacing card PCL series card 207.
9.	F.B Dyck	Precise Automatic Water Dispenser for Plants Growing in Pots.	No	Photoelectric Controller, V4-12 Micro Switch	No	Data reading and Monitoring	No

10	Jacquline M.S. Waworundeng, Novian Chandra Suseno, Roberth Ricky Y Manaha.	Automatic Watering System for Plants with IoT Monitoring and Notification. (2018)	No	Watering System Soil moisture sensor detects the water content from the soil and gets analog microcontroller feedback signal. The relay provided a signal that leads to the opening or closure of the solenoid from the Microcontroller. When the relay is on, the solenoid valve has opened the valve and when the relay is off locked. When the solenoid valve opens, water flows across the watering pot. Monitoring System The Wemos D1 microcontroller from the soil moisture sensor provided the signal for sensing ground moisture. Sending cloud data to ThingSpeak.com is a Wemos D1 microcontroller, which has bluetooth interface and wireless connectivity. Notification System. Blynk apps collect data from Wemos D1, store and forward the data to the smartphone of the Customer. Users will check whether they are switched on or off on the watering app screen on the mobile on Blynk Applications.
11	Adeoudu A.O, Bodunde O.P, Daniyan I A, Omitola O.O, Akinyoola J.O, Adie U.C	Development of an autonomous mobile plant irrigation robot for semi structured environment.	No	Three types of land tested included: computerized system-building simulation, physical circuit simulation using a breadboard to ensure correct operation and the final realization of the circuit on the Vero board.

12	Lokesh Agrawal, Bhitika Dewangan, Shilpi Desmukh, Priyanka Yadav, Mukesh Kumar Chandrakar.	Smart Irrigation System Using 8051 Microcontroller and Fencing Control Using Bluetooth Module.	No	The deficiency of water in the field is sensed by the op-amp based sensor. Whenever there is need of water in the particular field, the high signal appears on the output pin of the sensor of that particular field.
13	S.R Khumbar, Arjun P Ghatule.	Microcontroller based Controlled Irrigation System for Plantation.	No	Measuring the change in the humidity (wetness), the resistance changes and the output of sensor increase which is fed to the comparator.
14.	F.B Dyck	Precise Automatic Water Dispenser for Plants Growing in Pots	No	An apparatus which dispenses water automatically into the pot restoring it to the set weight. . As water is added and balance position is approached the upward movement of the balance arm releases the limit switch.

Table 2.3 Proposed Solution Table

	Board	**Sensors Used**	**OLED**	**Cloud**	**Mobile**	**Manual Intervention**
Proposed Solution	Node MCU, Arduino IDE	YL-69 Moisture Sensor, DS18B20, Soil Temperature Sensor.	Yes, Digital Display.	Blynk Cloud	Blynk, Notification & Control	Yes, with Buttons, & Blynk Apps.

2.8 Proposed Solution

Based on the literature review studied regarding to the hardware to be selected for this project, many hardware has been reviewed for the microcontroller and microcomputer, from many of the microcomputer the Node MCU ESP8266 is selected. The reasons for the Node MCU ESP8266 are chosen is first, the cost of the Node MCU 8266 itself, it is compatible and reasonable compared to the other boards available. It also has adequate functionalities that supports the cause of the project like the integrated Wi-Fi module and enough number of Pins to support all sensors. It is also very expandable for future upgrades.

Secondly the waterproof soil temperature sensor DS18B20, is used to take measurements of the soil temperature. The waterproof temperature sensor requires only one digital pin for two way communication with a microcontroller. Temperature sensor is fairly precise and needs no external components to work. The soil moisture sensor is used to take the measurements of soil moisture. These leads allow the current to pass through the soil and in return calculates the resistance value to measure the moisture level.

We have switches used to switch on and off the water pump to water the plant, these switches are used as a medium to provide manual intervention for the plant. An automation mode to water the plant according to its moisture in soil will also be developed so it functions in the case of sensors malfunctioning or crisis.

Blynk is integrated into the project to offer an advanced notification interface to further monitor the plant watering system. An alert and notification will be sent to the user about the sensors readings and the plants watering frequency. Data will be sent to Blynk applications every time there is an activity in the system. This will ensure the plant owner stays informed about the plant welfare and reduces the human effort and hassle hiring someone to water the plant in his absence. Figure 2.13 shows the proposed solution project setup.

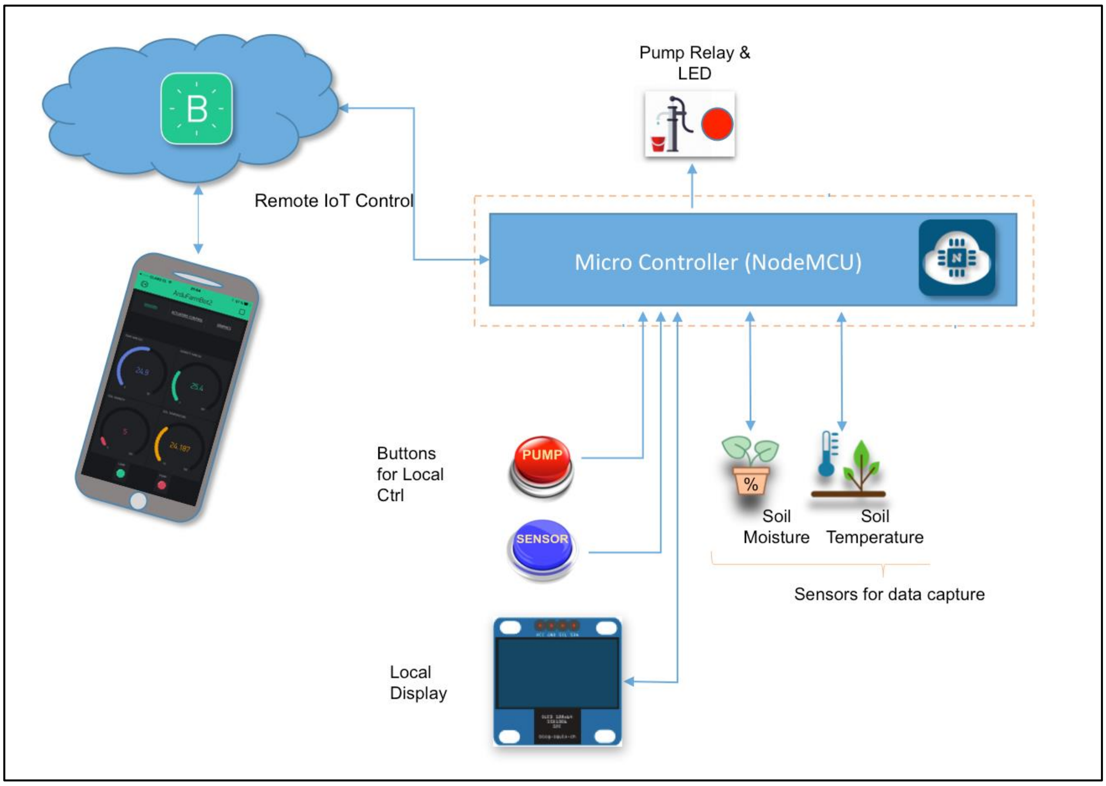


Figure 2.13 Proposed Solution for Project Setup

2.9 Conclusion

As conclusion, this chapter briefly explain about Internet of Things and the multiple hardware devices that are conventionally used to build an automated plant irrigation system. In order to archive the automated plant irrigation system using Internet of Things, it required to complete this task to deepen understanding and compare most suitable application or devices to use. By doing the literature review it help in understanding and choosing the most suitable implementation for the project. The next chapter will be chapter 3 which will be discussing about the project's methodology in detail.

CHAPTER III: METHODOLOGY

3.1 Introduction

Each chapter chooses and describes the methods used in the study. The two main parts of this section are the project methodology and the project milestone. The project methodology involves the process and system flow used throughout the project. This includes the milestone for the schedule for the entire project. The milestone listed is all the tasks included in this process, and a Gantt charts will be created to display project preparation in the allocated time.

3.2 Project Development

The fundamental method is to gather all information related to the development of Automatic Plant Irrigation system using Node MCU through the journal, articles, internet and books. The goal for the development Automatic Plant Irrigation is to irrigate the system for the facility as an automated system and to obtain the statistics for a user. The technology software will use the Node MCU (Arduino) as a microcontroller to address this challenge.

This project begins on early semester where for the first half of this semester, is the part of introduction phase where author will do the research from the information that already gather and start to analyse for required information and find the concept that related to the project. The purpose is to determine the best solution for the project in terms of hardware selection and software related.

The empirical phase to the development when the design is implemented into the real hardware and software then into the testing of function follow based on objective. The survey is performed regarding to the design tool and project utility that use on the project.

3.3 Project Methodology

The methodology selected for this project is waterfall model which is known as the linear-sequential life cycle. This model required to complete each of the phase before can proceed to the next phase and there would be no overlapping for each phase. The waterfall model often uses for the project that does not required for incremental on each phase and one of the popular strategies for Software Development Life Cycle (SDLC). The waterfall model consists of seven phase which is, requirement, analysis, design, implementation, testing, deployment and maintenance.

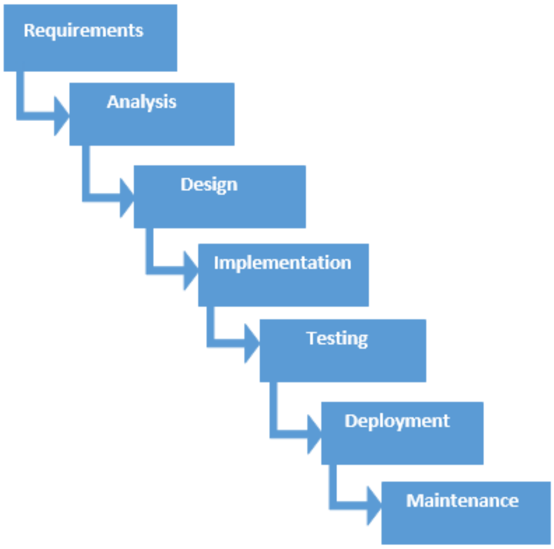


Figure 3.1 Waterfall Model Methodology (Royce, 1970)

3.3.1 Phase I: Requirements

The main objective of this phase is to document everything that related to project in order to understand about the problem statement, objective, scope of the project. The requirement for hardware and software also being discuss in this phase where all information is gathering and try to match with the project objective.

i. **Software Requirement**

Arduino IDE - Arduino IDE is used to insert programs into the microcontroller. Arduino are made from java programming language and Arduino IDE also comes with a library C/C++ commonly called wiring which makes input operation and output easier. The Arduino package usually include USB downloader so that the flash program to the microcontroller is directly connect to the USB laptop.

Blynk - It can be used for the remote reading, saving and simulation of sensor data and hardware. Blynk is an IoT platform application which is installed on a user's smartphone to send notifications about watering device status whether it is activated or deactivated.

ii. Hardware Requirement

NodeMCU - Will be programmed almost as a regular Arduino, using its IDE. Any new “custom firmware” will replace anything previously stored in the chip’s flash memory, including the original firmware loaded at factory where the AT commands were common used.

Soil Moisture Sensor - The soil moisture sensor consists of two leads that are used to measure volume of water content in soil. These leads allow the current to pass through the soil and in return calculates the resistance value to measure the moisture level.

Waterproof Temperature Sensor DS18B20 - The waterproof temperature sensor requires only one digital pin for two way communication with a microcontroller. Temperature sensor is fairly precise and needs no external components to work. It can measure temperatures from -55°C to +125°C with ±0.5°C Accuracy.

Mini DC Water Pump - This DC 3V-5V micro dipper water pump will easily be built into your project water system. The water pump is used to drain and unload water through the inlet using the water aspiration process.

3.3.2 Phase II: Analysis

During this phase the required information that already gather is analyse and breakdown into the part in order to get the best solution. The author study in related field such as IoT for agriculture and IoT for smart home farming to understand and find the significant of the project. While the project already understands by the author, the author begins to make comparison of the technology that available to satisfy the objective of the project.

For example, the hardware to use for microcomputer is compare with the recent that available in the market and the lowest cost with the highest benefit to the project is selected for the project.

For the software development, the author needs to determine what is the best programming language to be use in order to achieve the objective, and the programming language that selected is Arduino programming language which is the primary running environment. This phase is important phase since it will determine the direction of the project in the future. And the decision taken during this phase need to be based on the research and information that already gather.

3.3.3 Phase III: Design

The data that already analyse and studied from previous phase is taking into the design model which is include the technical and system design model. At this phase it will determine in details which hardware and how hardware will be connected with each other's. For the software design model there is logical design that will have design related to the system to be develop. Flowchart are developed during this phase that will help during the development of the software. The basic interface of the software also designs during this phase.

3.3.4 Phase IV: Implementation

The objective of the implementation phase is to complete from the detailed design model and implement into the real development. The design model of the proposed product is implemented in this phase which is include the hardware and software implementation. For this project the implementation phase is divided into three parts where the first part is focus on the installation and configuration of the hardware.

The second part of this implementation is coding development for the plant irrigation system which will allow the system to respond to the plants soil condition automatically and at the same time allow manual intervention for user through switches.

The third part is the integration part, during this part the hardware is combined with the cloud connectivity with Blynk acting as the cloud server and Blynk app as the mobile application that provides the monitoring function. The implementation part is the longest phase for this project since it required a lot of work to be done. And during the development the author should face with a lot of error, bug and system logic problem. This phase later will be past to the testing phase where the system or hardware will be testing the functionality and any possibility that would happen.

3.3.5 Phase V: Testing

The objective of this phase is to test what have been implement. The Question will be bringing up relate to the hardware and the system and making the list of possibility that would happen in order to test the recent development. The functionality of hardware and software will be test in details and aspect. For the testing phase, the automated plant irrigation system is tested with integration to the Blynk cloud and mobile applications. The system is checked for its implementation performance and results.

3.3.6 Phase VI: Deployment

The functional and non-functional of the hardware and system is done testing, the project is now ready for the deployment and used in real environment. This phase is coming when the testing phase is clear from the bug or error or any issues and it will deploy for the production and used by the end users.

3.3.7 Phase VII: Maintenance

The objective of the maintenance phase is to keep to system and hardware is functional correctly. This phase is required the author to monitor the product that already produce and provide with the bug fixes and ready to receive the report from the end user regarding to the product. It is necessary to provide new upgrades and for the prototype when it is available to the end user.

3.4 Project Milestones

Project milestone was timeline that will ensure the completeness of the system based on the specific timeframe. The project milestone consists of the project flow, which is the start and completion time of the project. The milestone was also being use to estimate the duration its need to complete the project. Table 3.1 shows the Project Milestone for PSM 2.

Table 3.1 Project Milestones for PSM 2

Week	**Date**	**Activities**	**Deliverables**
1.	09/09 - 13/02	- Proposal PSM: Discussion - Proposal assessment & verification. - Change of supervisor letter	Proposal Preparation
2.	16/09 – 20/09	- Meeting the supervisor. - Proposal correction/improvement.	
3.	23/09 – 27/09	- Proposal Submission - Chapter 1	Chapter 1 Completion
4.	30/9 – 4/10	- Chapter 2	
5.	7/10 – 11/10	- Chapter 2 - Meeting Supervisor	Chapter 2 Completion
6.	14/10 – 18/10	- Chapter 3	
7.	21/10 – 25/11	- Chapter 3 - Meeting Supervisor	Chapter 3 Completion
8.	28/10 – 3/11	- Chapter 4	
9.	4/11 – 8/11	- Chapter 4 Completion	
10.	11/11 – 16/11	- Chapter 5 - Project Demo	Progress Presentation 1

11.	18/11 – 22/11	- Chapter 5 completion	Testing Method & Expected Results
12.	25/11 – 29/11	- Meeting with supervisor - Testing method results - PSM 2 Report	Chapter 6 – Test Results
13.	2/12 – 6/12	- Meeting with supervisor - PSM 2 Report - Claim form submission	Chapter 6 Correction and Presentation
14.	9/12 – 13/12	- Project Demonstration - PSM 2 Report	Project Demonstration PSM 2 Report Submission
15.	16/12 – 20/12	- Final Presentation	Second Presentation of PSM 2 and Report Submission

3.5 Gantt Chart

Table 3.2 Gantt chart for PSM 2

Activity	Week													
	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12	W 13	W 14
- Proposal Submission														
- Meeting with supervisor - Proposal correction and improvement.														
- Proposal Presentation & submission via PSM Online System. - Chapter 1														
- Chapter 2														
- Chapter 2 - Meeting with supervisor														
- Chapter 3														
- Chapter 4														
- Meeting with supervisor - Chapter 4														
- Meeting with supervisor - Chapter 5														
- Chapter 6 - Claim form Submission														
- Project Demonstration PSM 2														
- Final Project Presentation														

3.6 Conclusion

As the conclusion it is crucial to have the project methodology to act as the guide to complete the development of the project. For this chapter the methodology use is the waterfall model taking from SDLC and the most suitable methodology use since the product does not required for circular process for each phase. The milestones that set is to finish the work according to the time frame that already decided so the project will not be delay. The next chapter 4 will discuss about the analysis and design which related to the problem analysis, hardware and software to be use in the project and the high-level of design.

CHAPTER IV: ANALYSIS DESIGN

4.1 Introduction

The analysis and design of the automatic irrigation plant system are discussed here. First of all the task is evaluating and splitting into smaller components and researching the solution of each problem in the associated dossier. This analysis technique will help the project to develop. The next stage is the development process in which the planned project results are stored. Hardware and software requirements are all contained in this chapter, which outlines the system and hardware's functionality and architecture. In this chapter the system design will explain how the system works and understanding the flow of the system.

4.2 Problem Analysis

The continuous extraction of water from earth is reducing the water level due to which lot of land is coming slowly in the zones of un-irrigated land. Another very important reason of this project is due to unplanned use of water due to which a significant amount of water goes to waste. In modern irrigation system, plants can be easily monitored and can be taken care of by automatic plant irrigation system, in where DC motor is used. For this implementation Node MCU is used to control the water irrigation of the plant to provide more nuanced

4.3 Requirement Analysis

Requirement analysis is related to the process of determine the requirement of the project. It is and important task and process and need to provide with the accurate information. In this subtopic it discuss about the data requirement, function requirement, non-functional requirement and other related requirement which is software and hardware requirement for the project.

4.3.1 Data Requirement

The product that design should have the input and output process where it required to happen since the system will be interact with the hardware. The hardware will require the input to process the information and display the output into the system as the result from the process. For this project the input data would be the command coming from the buttons and sensors etc. The data is sent by the microcontroller then transmitted into the cloud server where the data would be stored and recorded. It will be displayed and showed in the Blynk application for monitoring purposes.

4.4 Other Requirement

This subtopic will discuss about the software and hardware requirement for the development of the project. It is important to understand the purpose for each hardware and software will be used for this project.

4.4.1 Software Requirement

Arduino IDE - Arduino IDE is used to insert programs into the microcontroller. Arduino are made from java programming language and Arduino IDE also comes with a library C/C++ commonly called wiring which makes input operation and output easier. The Arduino package usually include USB downloader so that the flash program to the microcontroller is directly connect to the USB laptop.

Blynk - It can be used for the remote reading, saving and simulation of sensor data and hardware. Blynk is an IoT platform application which is installed on a user's smartphone to send notifications about watering device status whether it is activated or deactivated.

4.4.2 Hardware Requirement

I. **Node MCU -** Will be programmed almost as a regular Arduino, using its IDE. Any new "custom firmware" will replace anything previously stored in the chip's flash memory, including the original firmware loaded at factory where the AT commands were common used.

II. **Soil Moisture Sensor** - The soil moisture sensor consists of two leads that are used to measure volume of water content in soil. These leads allow the current to pass through the soil and in return calculates the resistance value to measure the moisture level.

III. **Waterproof Temperature Sensor DS18B20** - The waterproof temperature sensor requires only one digital pin for two way communication with a microcontroller. Temperature sensor is fairly precise and needs no external components to work. It can measure temperatures from -55°C to +125°C with ±0.5°C Accuracy.

IV. **Mini DC Water Pump -** This DC 3V-5V micro dipper water pump will easily be built into your project water system. The water pump is used to drain and unload water through the inlet using the water aspiration process.

V. **Monitor** For the output display of the raspberry it required monitor, the monitor is use actively while configuring the Raspberry Pi. The Raspberry Pi use the HDMI port as the display output and the monitor to use is required to support the HDMI or alternatively can use the converter to convert VGA to the HDMI.

VI. **Keyboard and Mouse**

For the input devices, the keyboard and mouse were used. It was used on the configuration phase where the command that need to be run is entered by using the keyboard and for the interaction with the GUI it can be use the mouse.

4.5 High-Level Design

This subtopic is the high-level design where it will contains the structure view of the system to be developed which is have the architecture design, circuit design and user interface design.

4.5.1 Architecture Design

The prototype of automated plant watering system should be able to detect the level of moisture and temperature in the soil and respond to either turn on the water pump. The water pump has a reservoir of water attached to it. The pump has a controlled flow rate of water that regulates the amount of water to irrigate. The commands are controlled by the microcontroller.

Every data is recorded and sent to the Blynk application where the entire irrigation activity is logged and stored for monitoring purpose and the owner gets notified through the Blynk app about the watering routines of the plant.

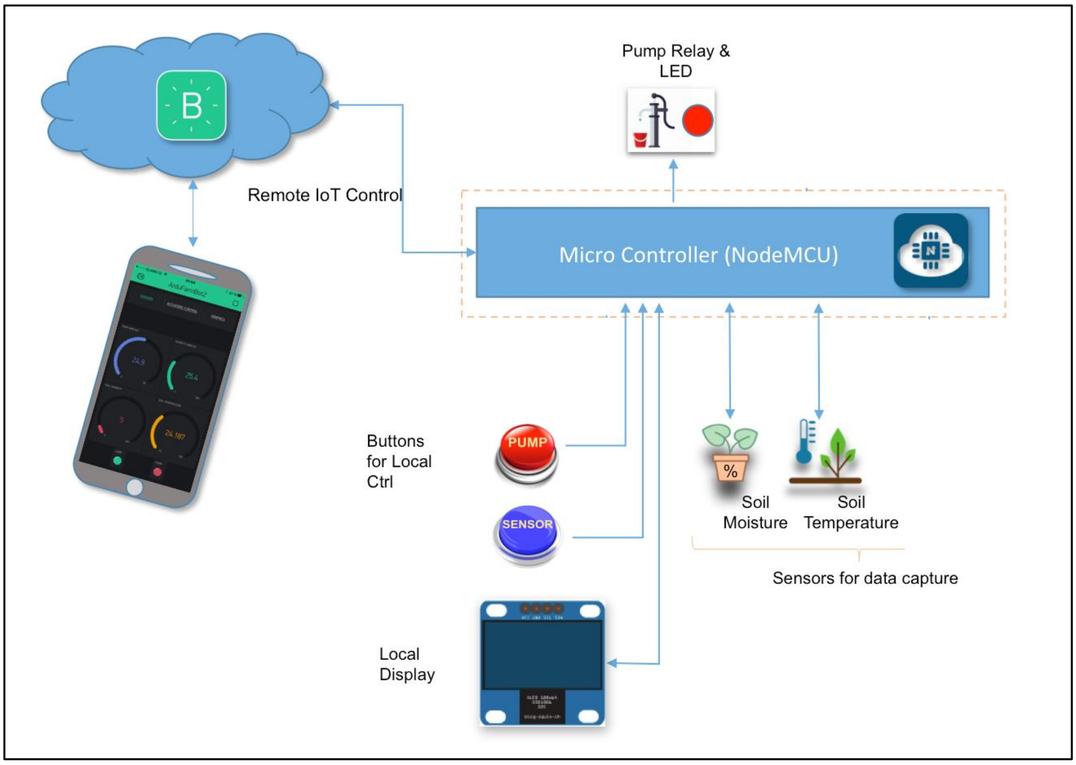


Figure 4.1. The Architecture Design of the Project

4.5.2 Circuit Diagram

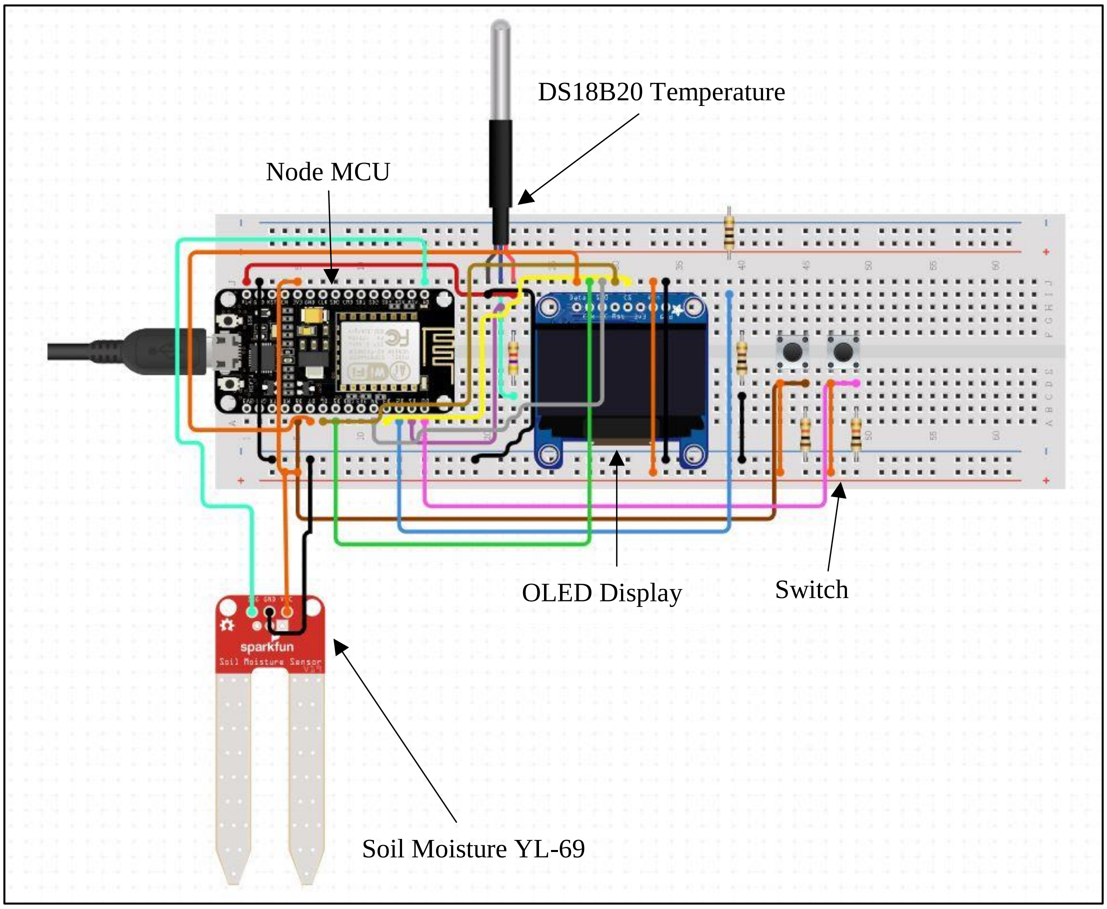


Figure 4.2. Circuit Diagram of the prototype

4.5.3 OSI Layer Interpretation

OSI (Open Systems Interconnection) is the model that show how the application is communicate in the network environment. It comes with seven layer that start with physical layer which involve the hardware until the last layer which is the application layer where the user interacts directly. The figure 4.7 shows the OSI model for this project where at the layer seven to layer five is related to the user interface and user interaction the system. For this project user can access to the system through any type of devices and platform. For the layer four until layer one is show about the hardware setup and data communication.

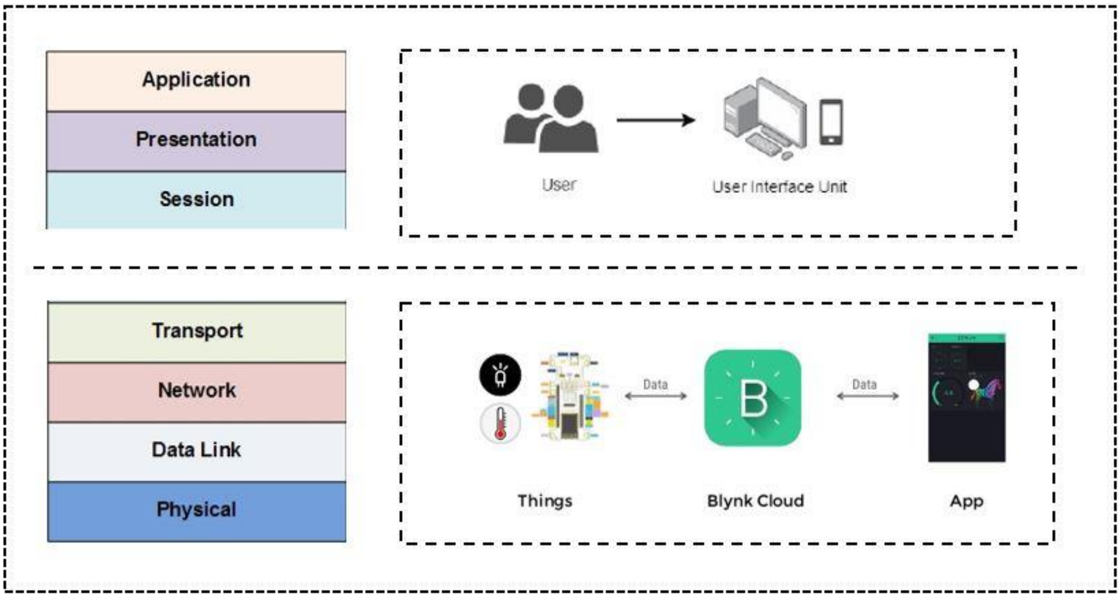


Figure 4.3 The OSI Layer Interpretation

4.5.4 IoT Protocols in OSI layers

1. **Wi-Fi (Data Link Layer 2) -** For IoT connectivity, WiFi is a preferred choice for many software designers. It is due to the equipment this holds. It has fast data transmission rates and the ability to manage a large amount of data. The widely available WiFi 802.11 standard allows you to transfer hundreds of megabits in just a second.

   In this project the automatic plant irrigation system is connected to the Wi-Fi or a Wi-Fi hotspot to ensure digital data transmission happens from the Node MCU to the Blynk cloud. These data needs to be sent over the network to make the monitoring system possible.

2. **MQTT (Application Layer 7) -** MQTT is a message protocol and full form is Message Queue Telemetry Transport. The main task of MQTT is to obtain data from so many power supplies e.g moisture sensor and temperature sensor. It operates in addition to the TCP to provide stable and simple data streams.

   This protocol MQTT consists of three core elements or processes: subscriber, publisher and broker. The publisher's work produces data and transmits the data with the aid of the broker to the user.

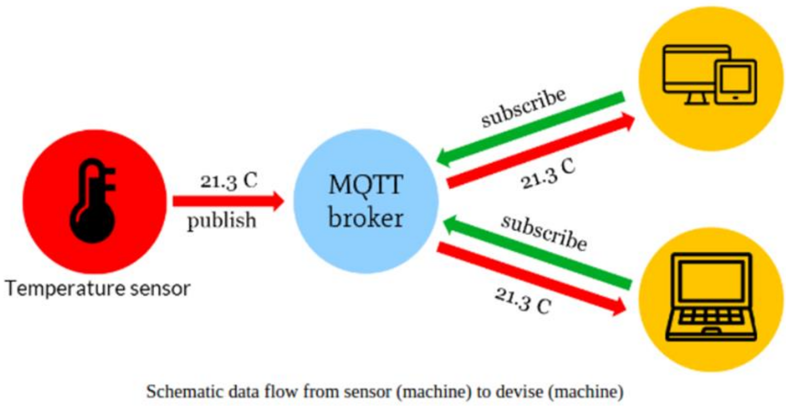


Figure 4.4 MQTT schematic data flow from sensor to device

In this project, there is a temperature control and moisture control sensors. The sensors will send the broker it’s readings. On the other hand, this temperature value is required by a phone or desktop application, Blynk. 2 things will therefore happen: The device specifies the subject to be published, e.g. "temp." The message "temperature value" is then written.

The Blynk app uses the "hot" theme. Then the response that is the temperature value of the unit is sent. Once, the broker's job is to take the "temperature interest" message and send it to the Blynk application.

4.5.5 User Interface Design

User interface design will show how the interface of the system will looks. This interface design should support for the responsive view, interactive and user friendly. The navigation panel and the sensor data must be displayed interactively. The design should be direct and show it functionality with the minimal of noise. This subtopic will show the navigation design, input design and the output design.

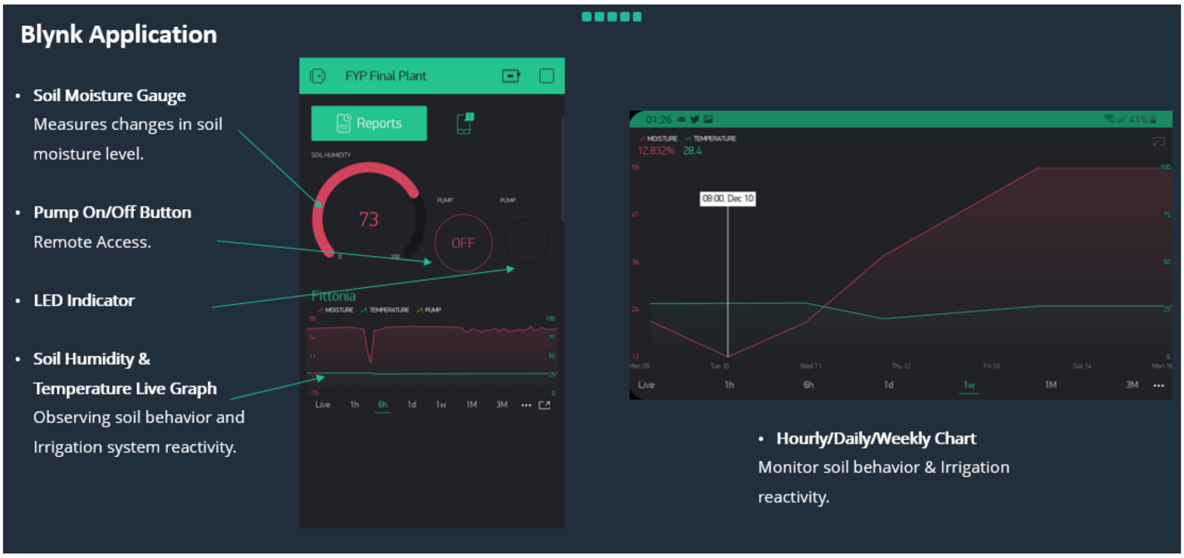


Figure 4.5 User Interface Design Blynk

4.5.6 Flowchart Design

Flowchart is a formalized graphic representation of a logic sequence of the system. In the design phase flowchart is one part of the project development. This flowchart will show how the flow of the system functionality is.

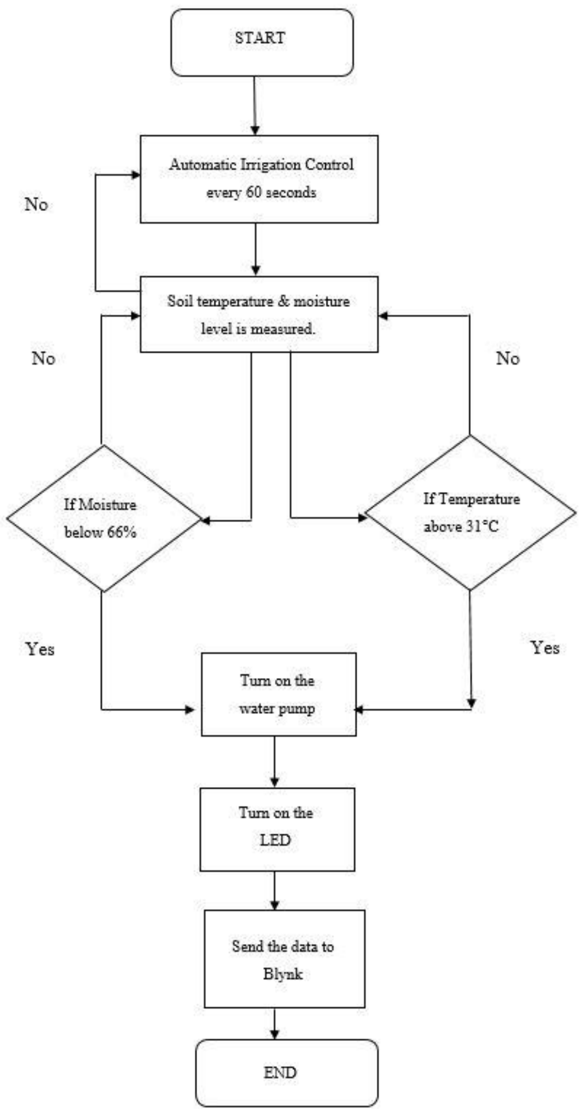


Figure 4.6 Flowchart of Project

4.6 Conclusion

As the conclusion, this chapter is important to be clearly discuss before going to the next chapter which the implementation phase. This chapter discuss about all the requirement include the software and hardware requirement. Moreover, this chapter show the design of the system and the flow chart of the system. The chapter 5 will be discussing regarding the implementation process in detail.

CHAPTER V: IMPLEMENTATION

5.1 Introduction

This chapter describes about the implementation status of the system. The implementation phase of this system divides into three parts, which the first is the Node MCU and Arduino IDE configuration, development of the coding, and the third part is the software or Blynk App development. This chapter will describe in details about the configuration and the expected output of the project.

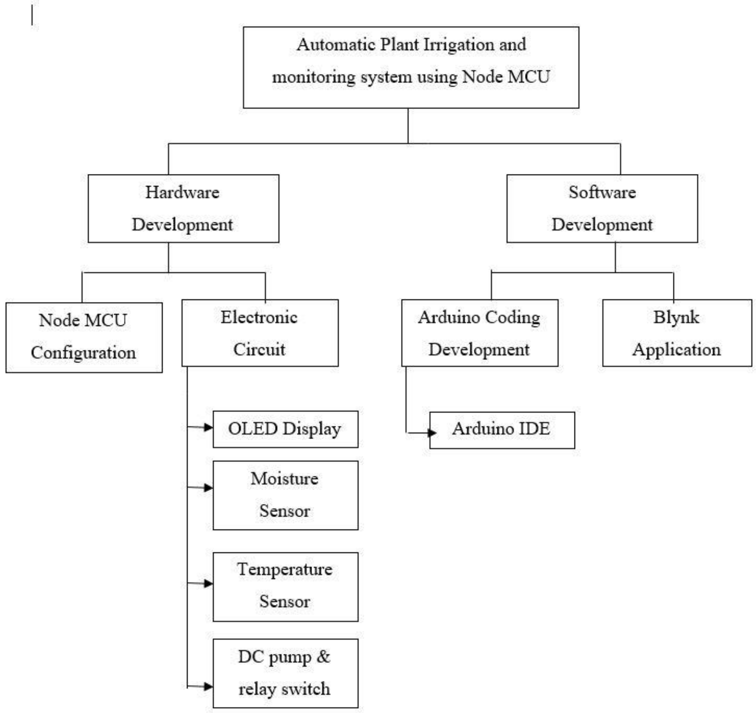


Figure 5.1 Overview of chapter 5

5.2 Development Environment Setup

This section describes about development process which is involving the development for the hardware and system. The setup is divided into two phase, hardware setup and software and the detail will be explained in the subsection below.

5.2.1 Hardware Development

In the hardware development, The Node MCU microcontroller is used as the main processor unit that will process all information that receive and transmit. The Wi-Fi module, breadboard, AC Adapter and USB cable will be use to provide the communication medium between the devices.

i) Arduino Node MCU

Node MCU is used as the main unit for this project, it will accept the request and response with the appropriate result. The Node MCU need to be installed with the appropriate libraries that the sensors and other communicating devices requires to aid the coding development. The communication between the sensor modules and Node MCU happens through breadboard while the integration between the microcontroller and the Blynk happens through the Wi-Fi module.

The recommended specification for the project is explained in the below section. By using the recommend setup environment it would help to eliminate any errors during the hardware setup.

A) Hardware Requirement

- NodeMCU ESP8266 12-E
- Soil Moisture Sensor
- Waterproof Digital Temperature Temp Sensor Thermal Probe DS18B20
- 0.96″ I2C IIC SPI Serial 128X64 Yellow & Blue OLED LCD ($9.00)
- Buttons (2X)
- LEDs (1X)
- DC 12V 2CH 2 Channel Isolated Optocoupler High/Low Level Trigger Relay
- Module
- Jump wires
- Breadboard
- 4.7K ohms resistor
- 10K ohms resistor
- 220 ohms resistor
- 5V/ 2A External Power Supply
- Mini DC Water Pump

B) Software Requirement

- Arduino IDE
- Blynk

Figure 5.2 below show the hardware development for the Node MCU, which has a USB data connection to the laptop and power cable connect to the port to supply excess power supply. This recommended setup will allow Node MCU to be access through the laptop for programming purpose.

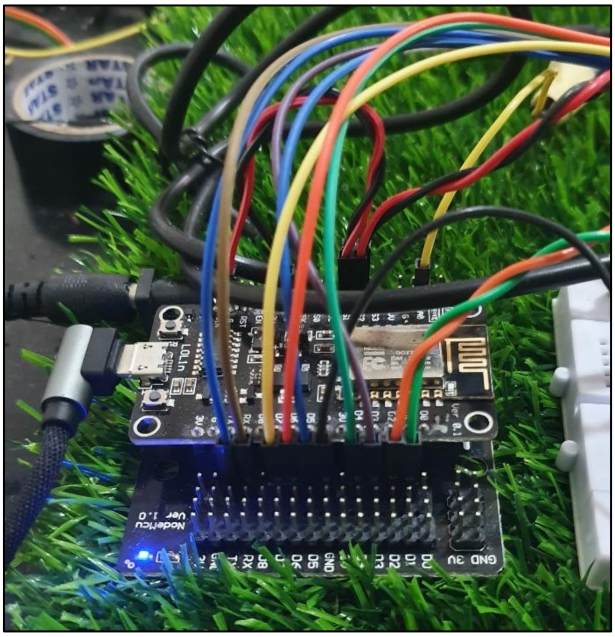


Figure 5.2 Node MCU Setup

5.3 Project Configuration Management

This section describes about the configuration setup of the hardware and system. Every step of the configuration and coding implementation will be explained in this section.

5.3.1 Hardware Configuration Setup

NodeMCU ESP-12E is the integrated version of the popular ESP8266, a Serial to Wi-Fi “System on a Chip” (SoC) Main modules are without doubt: the ESP-01 and the ESP-12E. On this project, the ESP-12E Development Board (NodeMCU DevKit 1.0) is used. Its Pinout is shown below. Figure 5.3.2 shows the node MCU pin out.

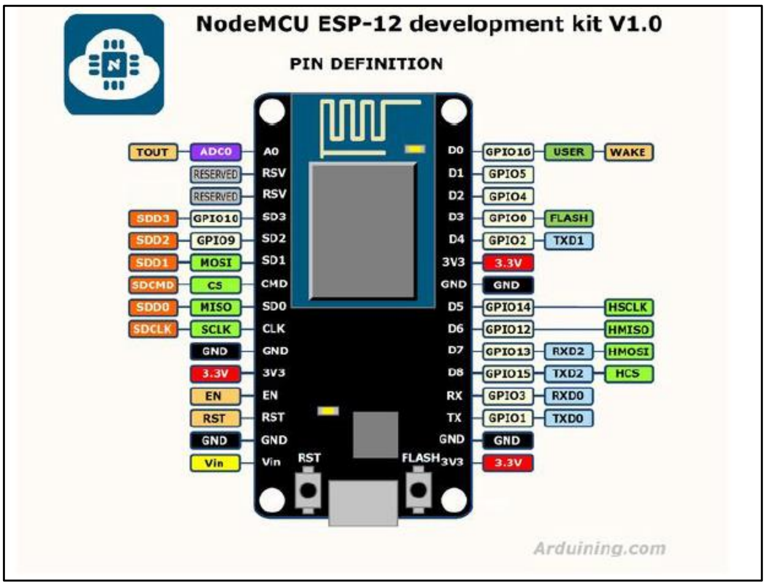


Figure 5.3.1 Node MCU ESP-12 Pin Out

5.3.2 Using Arduino IDE with Node MCU

The Node MCU is used as a regular Arduino, using its IDE. Although the manufacturer’s SDK to develop the custom firmware, it is much easier to use the good and old Arduino IDE.

1. In the Arduino IDE: Open PREFERENCES window and Enter below URL into the Additional Boards Manager URLs field, and http://arduino.esp8266.com/stable/package_esp8266com_index.json
2. Select OK. Figure 5.3.1 shows the Preference setup in Arduino IDE.

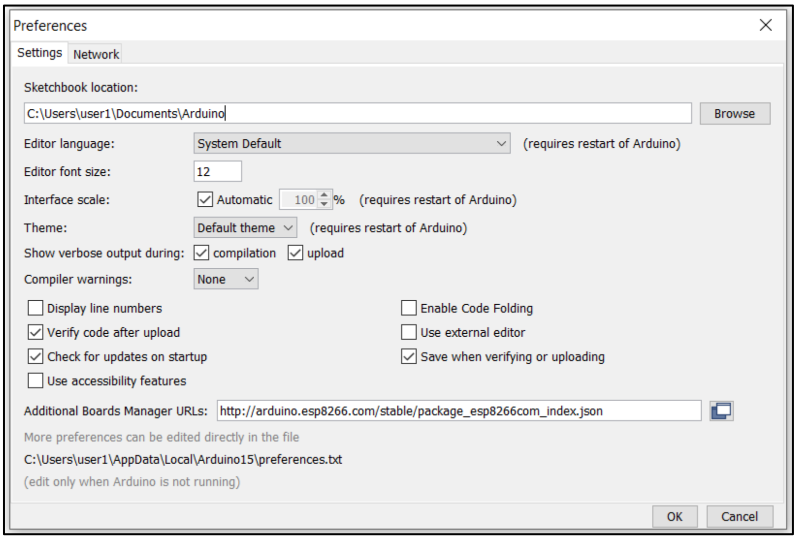


Figure 5.3.1 Preference configuration Arduino IDE

3. Select MENU option Tools → Board → Boards Manager and scroll down to locate the option esp8266 by ESP8266 Community which should be the last item on the list.
4. Click INSTALL. Figure 5.3.2 shows the library manager in Arduino IDE.

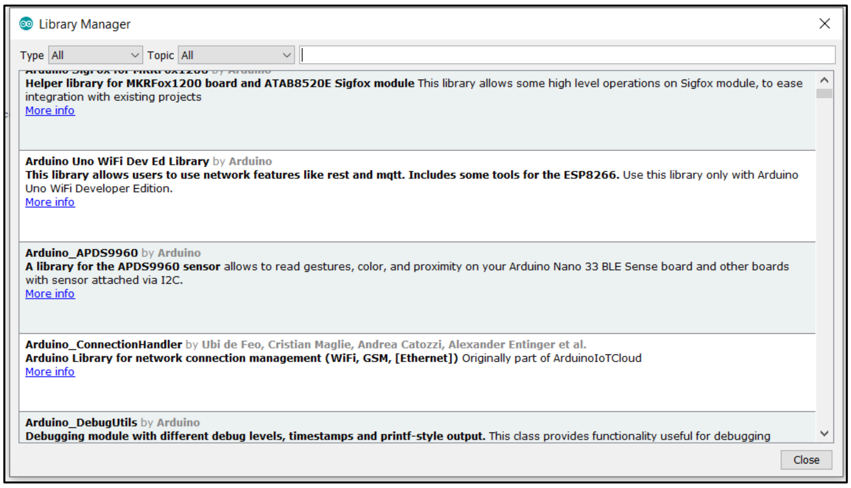


Figure 5.3.2 Library Manager in Arduino IDE

5.3.3 Installing the USB Drivers

The USB to Serial UART module included in the panel is the CP2012 of Silicon Labs, for which the Virtual COM Port (VCP) drivers are normally mounted. In the case of MAC, UART has the name /dev / cu. SLAB USB in the system folder generated to communicate with the CP2102.

1. After restarting the Arduino IDE we can now select the board we are using from the MENU:
2. Tools → Board → NodeMCU 1.0 (ESP-12E Module).

3. Then, we specify correct CPU Frequency Tools → CPU Frequency: → 80MHz and Upload Speed: Tools → Upload Speed: → 115200).

4. Finally, the last step is to select the correct option for the Port: Tools → Port → /dev/cu.SLAB_USBtoUART. Figure 5.3.3 shows the Board selection pane in Arduino IDE. Figure 5.3.3 shows the board selection in Arduino IDE.

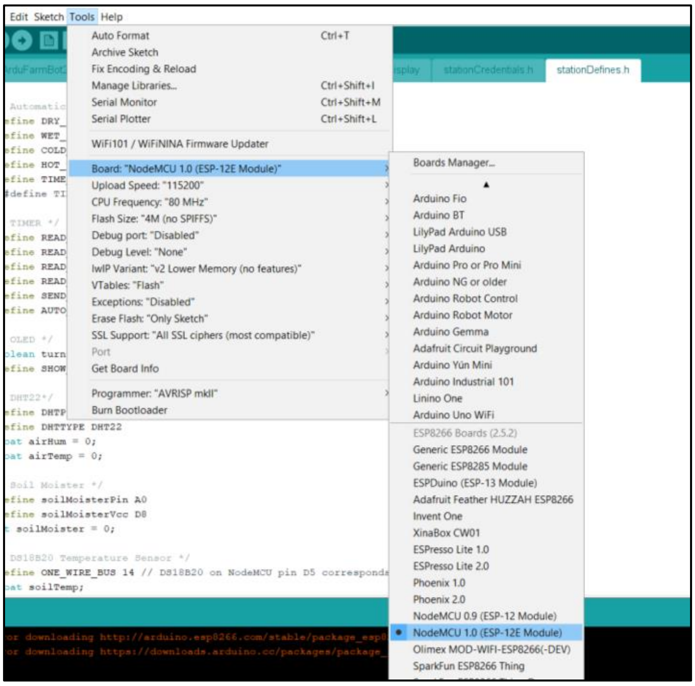


Figure 5.3.3 Board Selection in Arduino IDE

5. Write an own firmware and upload it, try one of the examples: File → Examples → ESP8266WiFi → WiFiScan.

6. After uploading it, the Serial Monitor window was opened and the results were obsreved.

5.3.4 Installing the OLED display

The used model is a 128x 64-pixel display that communicates via I2C, SSD 1306 with 128 pixels in horizontal position and 64 pixels in vertical position. So, we'll get a "16X8" display (8 lines of 16 characters each) if you use 8 pre-8 characters. The SSD1306 can be powered directly from the Node MCU module by 5V (external) or 3.3V. The first option for the project was the selected one (5V). It's an I2C screen, so we'll connect it to the I2C pins of NodeMCU using: SCL - D1 (A5 Arduino pin equivalent). SDA - D2 (Arduino pin A4 equivalent). After the displayed is linked, let us download and install the library on our Arduino IDE. Figure 5.3.4 shows the OLED display test code.

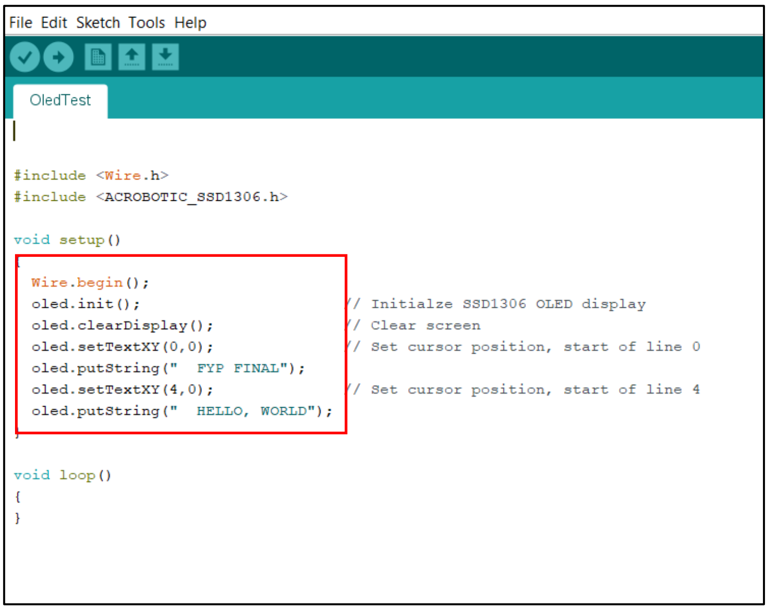


Figure 5.3.4 OLED Display Test Code

5.3.5 Capturing Soil Moisture Humidity

The LM393 module has 2 outputs, one digital (D0) that can be set-up using the potentiometer that exist on it and an analog one (A0). This module can be sourced with 3.3V, what is very convenient when working with a NodeMCU. Install the LM393 4 pins to NodeMCU as below:

- • LM393 A0 - NodeMCU A0
- • LM393 VCC - NodeMCU GPIO D3 or VCC (5V)
- • LM393 GND - NodeMCU GND
- • LM393 D0 - Not Connected

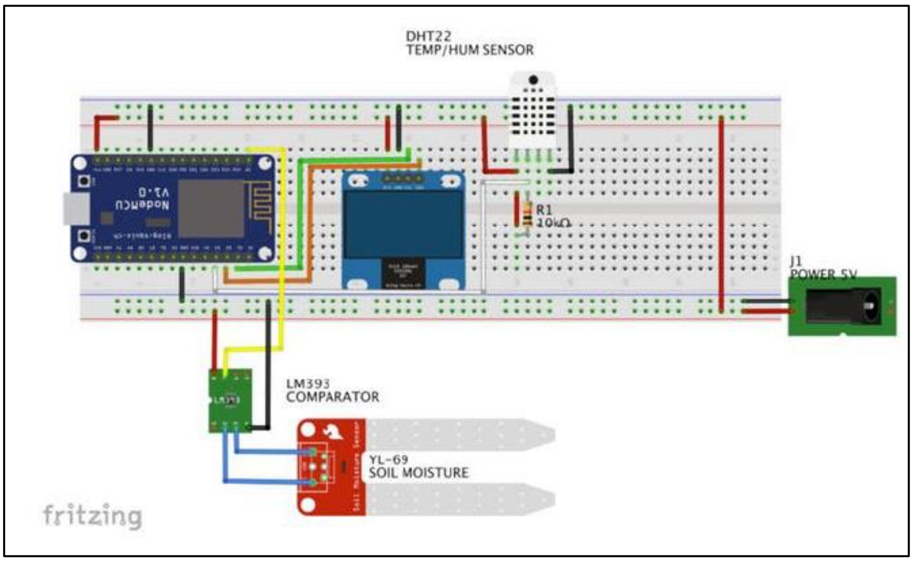


Figure 5.3.6 Soil Moisture Schematic Diagram

The MAP was to set up the range in percentage. When using a "short circuit" at the sensor samples ("100 percent humidity" equivalent) the value was of about 0 at the ADC output and would be around 600 (5V powered source). The value of this value would be "in the air." The data from the sensor is obtained 3 times, with an average.

```
/*****************************************************
 * Get Soil Moister Sensor data
 *****************************************************/
void getSoilMoisterData(void)
{
  soilMoister = 0;
  digitalWrite (soilMoisterVcc, HIGH);
  delay (500);
  int N = 3;
  for(int i = 0; i < N; i++) // read sensor "N" times and get the average
  {
    soilMoister += analogRead(soilMoisterPin);
    Serial.print (soilMoister);
    delay(150);
  }
  digitalWrite (soilMoisterVcc, LOW);
  soilMoister = soilMoister/N;
  soilMoister = map(soilMoister, 1000, 0, 0, 150); //LM393 on 5V (+Vin)
}

/*****************************************************
```

Figure 5.3.7 Soil Moisture Coding

5.3.6 Collecting Soil Temperature

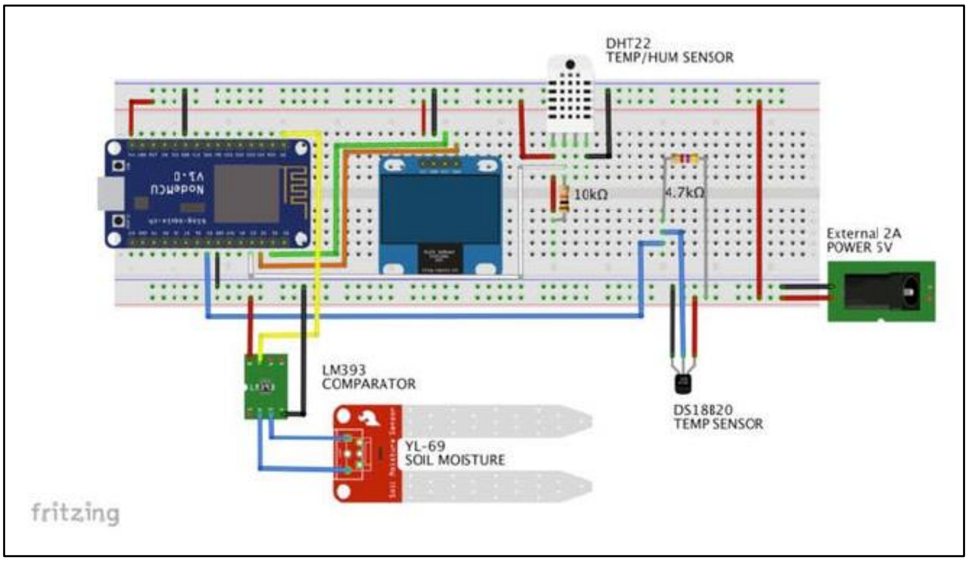


Figure 5.3.8 Soil Temperature Schematic Diagram

The DS18B20 is a powerful, long-distance digital sensor. The 1-wire digital temperature detectors are relatively accurate and can give up before 12 bit of the on-board digital/analog conversion accuracy (±0.5 °C over a large area of range). We can attach multiple sensors to the same pin using a single digital pin, until each have a special 64-bit ID burned into the factory to distinguish between them.

To use the DS18B20 properly, two libraries will be necessary:

- OneWire: https://github.com/adafruit/ESP8266Arduino/tree/esp8266/libraries/OneWire
- Dallas Temperature: https://github.com/milesburton/Arduino-Temperature-Control-Library

Set the two libraries to the directory in the Arduino IDE Library. After restarting IDE, the sensor was tested with the code "Simple.ino" as shown in the photograph, included in the Library Examples. Using the Serial Controller, add code to the Node MCU and controller temperature. Figure 5.3.9 shows the ‘Simple.ino’ example code that’s picked from the template.

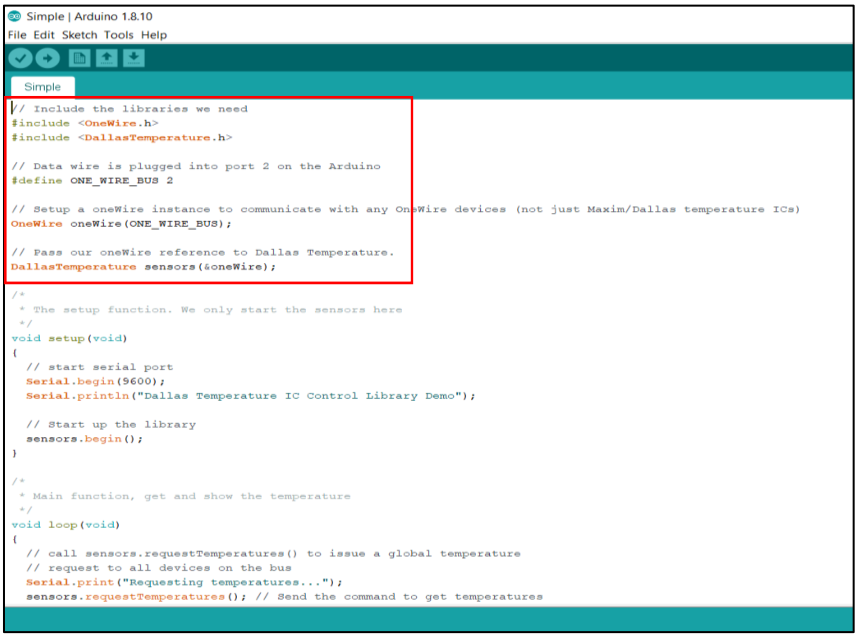


Figure 5.3.9 Dallas Temperature ‘Simple.ino’ example code

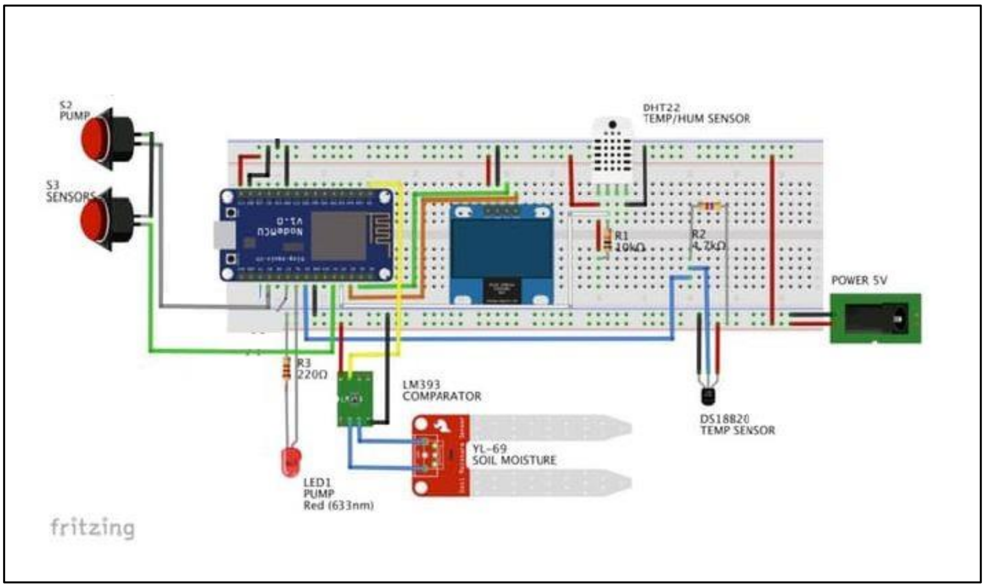


Figure 5.3.10 Full Schematic Diagram of the System

5.3.7 Completing the Circuit

A. LEDs

Remember that on the MCU node, LEDs are only to be checked. The pump (Red LED) is "simulated." Relays are attached to these inputs for the final circuit as explained in the next step.

B. Buttons

Due to the readings of the sensors, the operator can also control the pump manually. To do so, the plan will include three push-buttons:

- Switch 1: Pump Manual Ctrl
- Switch 2: Sensor Read

If the actuator is ON, the key will press "Turn-OFF," and vice versa: buttons operate on a "Toggle" mode. It means that the Node MCU Output is continuously "HIGH." The button's logic would be "normally closed." A "LOW" is introduced at the individual pin by pressing the button.

Figure 5.3.12 shows the code for debounce a key. A function **readLocalCmd** should be executed in order to read the local command. Every key of this function is read and the actuator variables **pumpStatus** is modified.

```
/*****************************************************
* Debouncing a key
*****************************************************/
boolean debounce(int pin)
{
  boolean state;
  boolean previousState;
  const int debounceDelay = 30;

  previousState = digitalRead(pin);
  for(int counter=0; counter< debounceDelay; counter++)
  {
    delay(1);
    state = digitalRead(pin);
    if(state != previousState)
    {
      counter = 0;
      previousState = state;
    }
  }
  return state;
}
```

Figure 5.3.12 Debounce key code

Figure 5.3.13 shows the command to trigger the pump. The function is called **aplyCmd** if a button is pressed. And according to its name, the correct control is applied, switching the actuators ON or OFF:

```
/*****************************************************
* Receive Commands and act on actuators
*****************************************************/
void aplyCmd()
{
  if (pumpStatus == 1)
  {
    digitalWrite(PUMP_PIN, HIGH);
    if (!turnOffOLED) displayData();
  }
  else
      {
        digitalWrite(PUMP_PIN, LOW);
        if (!turnOffOLED) displayData();
      }

}
```

Figure 5.3.13 Commands to indicate Pump status

C. Code considerations:

The 4 big "group of tasks" so far can be summarized into this:

- Read sensors
- Read buttons (local Command)
- Act on Pump
- Display all Data

The less tests we make, the easier for Soil Moisture Sensor (electrolysis causes due oxidation of the sample). The actuators, need a quick reaction when pressed a button. Using "timer" to manage these functions. Using milliseconds: SimpleTimer.h

To install the Library, follow the instructions on this link:

http://playground.arduino.cc/Code/SimpleTimer

The library is included on the main body of the code, following by a timer definition:

```
/***************************************************
* Starting Timers
****************************************************/
void startTimers(void)
{
  timer.setInterval(READ_BUTTONS_TM*1000, readLocalCmd);
  timer.setInterval(READ_SOIL_TEMP_TM*1000, getSoilTempData);
  timer.setInterval(READ_SOIL_HUM_TM*1000, getSoilMoisterData);
  timer.setInterval(DISPLAY_DATA_TM*1000, displayData);
}
```

Figure 5.3.14 Interval Timers for Sensor Readings

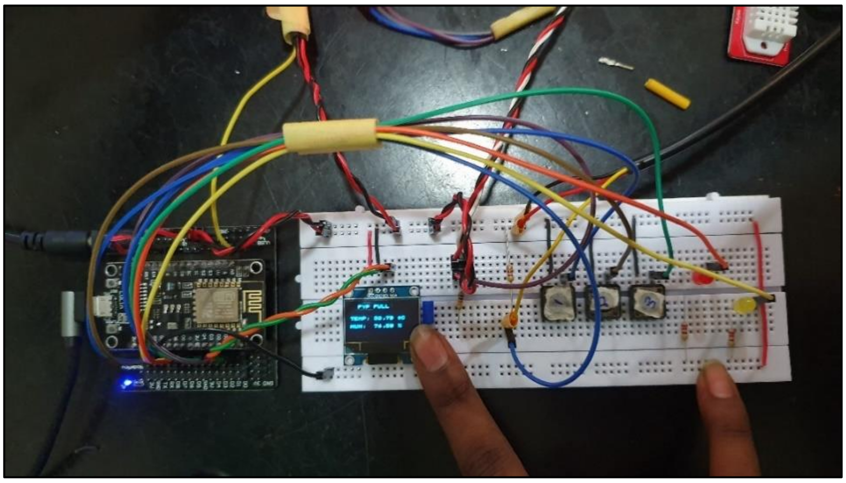


Figure 5.3.15 Breadboard and preliminary Sensor readings

5.3.8 Completing Local Display

A) Sensor Read Button

There is a potency to wait for long cycles between soil moisture sensor measurements, as we could see in the previous step. It is OK, but we don't want to "wait" for 10 to 15 seconds for our automated needs for manual operation.

Therefore put in place a 2nd key in our plan to show the actual sensor information at any time regardless of the predefined timing of the automated readings. When the sensors are changed, the same button to display information on the OLED. Figure 5.3.16 shows the read local command code.

```
void readLocalCmd()
{
  boolean digiValue = debounce(PUMP_ON_BUTTON);
  if (!digiValue)
  {
    pumpStatus = !pumpStatus;
    aplyCmd();
  }

  digiValue = debounce(SENSORS_READ_BUTTON);
  if (!digiValue)
  {
    turnOffOLED = !turnOffOLED;
    if (!turnOffOLED)
    {
      oled.setTextXY(0,0); oled.putString("UPDATING SENSORS");
      getSoilMoisterData();
      getSoilTempData();
      displayData();
    }else oled.clearDisplay();
}
```

Figure 5.3.16 Read Local Command Code

5.3.9 Making our Gardening System fully automatic

The system can be controlled by a local operator via buttons. The next thing to do is certainly introduce the automatic irrigation method and mechanism. Starting with initial interval where the sensors have to work. Those values are changed later using practical values obtained from a real garden:

Soil Moisture:

- "WET": Over 76-90% (no watering at all)
- "Target Humid": Between 45% and 75% (Where we want to work at) and
- "DRY": Below 10-35% (need to turn on the pump in order to increase humidity)

Figure 5.3.17 shows the parameter definition on the code for automatic irrigation:

```
/* Automatic Control Parameters Definition */
#define DRY_SOIL      10
#define WET_SOIL      35
#define COLD_TEMP     24
#define HOT_TEMP      35
#define TIME_PUMP_ON  5
```

Figure 5.3.17 Parameter Definition on Automatic Code.

TIME_PUMP_ON is the time in seconds that the pump must be ON during automatic operation. Based on above parameters, let's think about some very simple assumptions to be implemented on the code: If it's DRY => PUMP = ON

5.4 Building a Blynk App

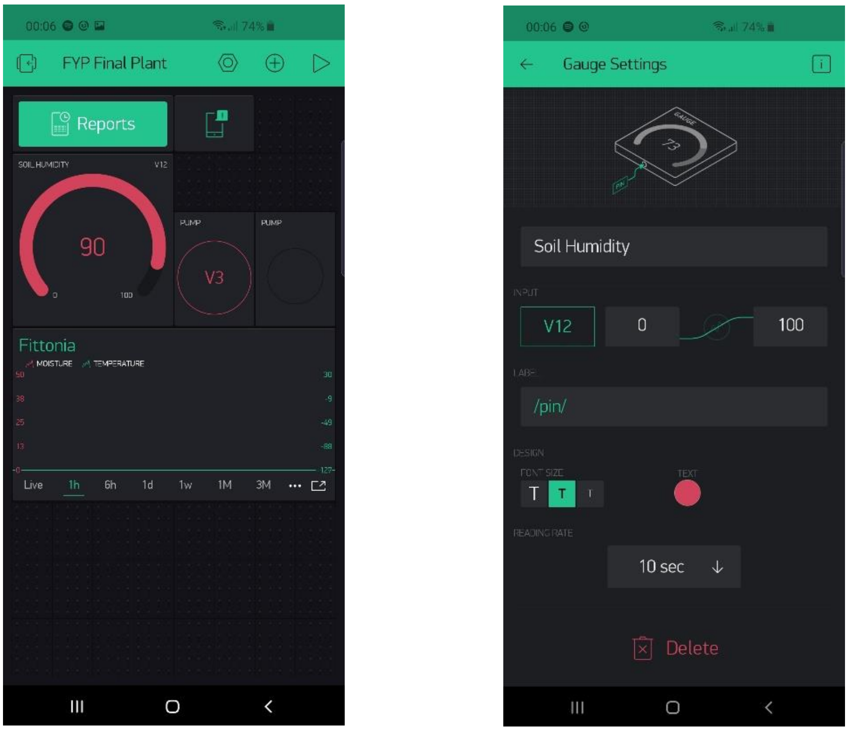


Figure 5.4.1 Main Interface and Gauge Widget on Blynk

1. Download BLYNK app for Apple iPhone or Google Android

2. Install BLYNK Library for Arduino:

https://github.com/blynkkk/blynk-library/releases/tag/v0.3.10

Now, go to the app at the Smart Phone and follow the steps:

3. Open Blynk app. Tap on “Create New Project” screen. Give a name for your project. Select an appropriated Hardware Model: “NodeMCU”. Take note from Authorization Token char auth[] = "YourAuthToken";

4. Press “OK”. A Blank screen with dots will appear. Tap the Screen to open the “Widget Box”. Figure 5.4.2 shows the graph and pump widget on Blynk.

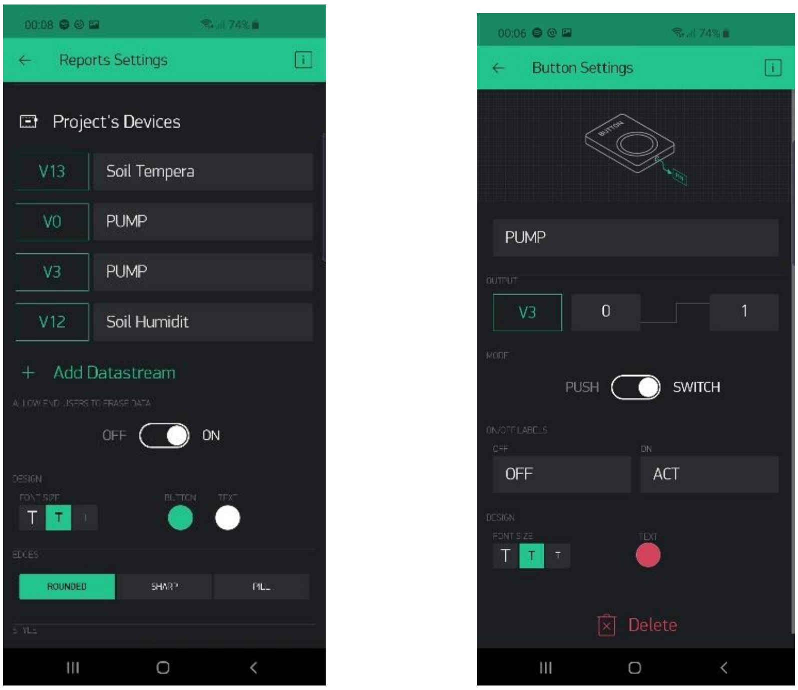


Figure 5.4.2 Graph and Pump Widget on Blynk

Four widgets were used which is the gauge widget, graph widget, Switch Widget, LED widget, Notification Widget and Report Widget. Revisiting general specification at introduction, we can summarize that our app will be needed for:

- Read all Sensors and verify actuators status.
- Take remote actions, “turning on/off” Pump and Lamp.
- Sending messages when System is “off-line” and/or an actuator is ON.
- Record general sensors data.

5.4.2 Changing code to introduce Blynk

To run a Blynk app together with your code these necessary inclusion has to be made to the original code:

1. Including BlynkSimpleEsp8266 library at beginning of the original code.
2. During setup(), initiate Blynk credentials: Blynk.begin(auth, ssid, pass);

```
char auth[] = "dO_ePkyxlTLKnUZg4Gies5LS3avi4Vxv"; // Blynk project: "FYP Final Plant"
char ssid[] = "wifi laju@unifi";
char pass[] = "Power123456";
```

Figure 5.4.3 SSID and Password Configuration for Blynk

3. Define a timing to send local data to Blynk server: timer.setInterval(5000L, sendUptime);
4. Call the function Blynk.run(); at loop()
5. Create the function sendUtime(); this is where the data will be sent to the Blynk Server: Blynk.virtualWrite(VirtualPin, sensor data);

The must now include the file StationCredentials.h:

Figure 5.4.3 shows how to define LED widget to use the “Virtual LED” on the Blynk app, it has to be defined at beginning of the code as below:

```
#define SW_VERSION "   SW Ver. 4.0" // SW version will appears at inni
#include "stationDefines.h"       // Project definitions
#include "stationCredentials.h"

/* ESP & Blynk */
#define BLYNK_PRINT Serial    // Comment this out to disable prints ar
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
WidgetLED PUMPs(V0);  // Echo signal to Sensors Tab at Blynk App
WidgetLED PUMPa(V5); // Echo signal to Actuators Tab at Blynk App
```

Figure 5.4.3 Defining the Widget LED

To “turn on” or “turn off” the PUMPs LED that relates to virtual PIN V0, for example, call functions respectively: PUMPs.on() or PUMPs.off().

The commands included at applyCmd() function, so LEDs on Blynk app will mimic real LEDs of our project. Figure 5.4.4 shows the code to turn the Pump On/Off function.

```
/*****************************************************
* Receive Commands and act on RELAYs - Logic LOW
*****************************************************/
void aplyCmd()
{
  if (pumpStatus == 1)
  {
    Blynk.notify("FYP Final: Warning ==>> Pump ON");
    digitalWrite(PUMP_PIN, LOW); // To be used with Relay module (inverted logic: activate with LOW)
    if (!turnOffOLED) displayData();
    PUMPs.on();
    PUMPa.on();
  }
  else
      {
        digitalWrite(PUMP_PIN, HIGH); // To be used with Relay module (inverted logic: normally HIGH)
        if (!turnOffOLED) displayData();
        PUMPs.off();
        PUMPa.off();
      }
}
```

Figure 5.4.4 Turning Pump On/OFF Function

To receive a command from a Blynk button, a function BLYNK_WRITE() must be defined outside a function, loop() or setup(). For that, a code was created, one for pump Blynk Button: Figure 5.4.5 shows the remote command to activate pump through Blynk.

```
/*****************************************************************
* Read remote commands
*****************************************************************/
BLYNK_WRITE(3) // Pump remote control
{
  int i=param.asInt();
  if (i==1)
  {
    pumpStatus = !pumpStatus;
    aplyCmd();
  }
}
```

Figure 5.4.5 Remote Command through Blynk to activate Pump

5.5 Complete Prototype

5.5.1 Hardware

The final prototype uses the Node MCU as the microcontroller use a 5V AC power adapter that powers up the breadboard and the integrated Wi-Fi module is used to connect the network. The sensors are all fixed to the breadboard according to the schematic diagram. The water reservoir is setup to place the water pump inside. The relay module is connect to the board and water pump. The sensors are all placed inside the plant pot. The Fittonia Albivenis indoor plant was used to experiment the prototype. Figure 5.4.1 shows the complete prototype of the system.

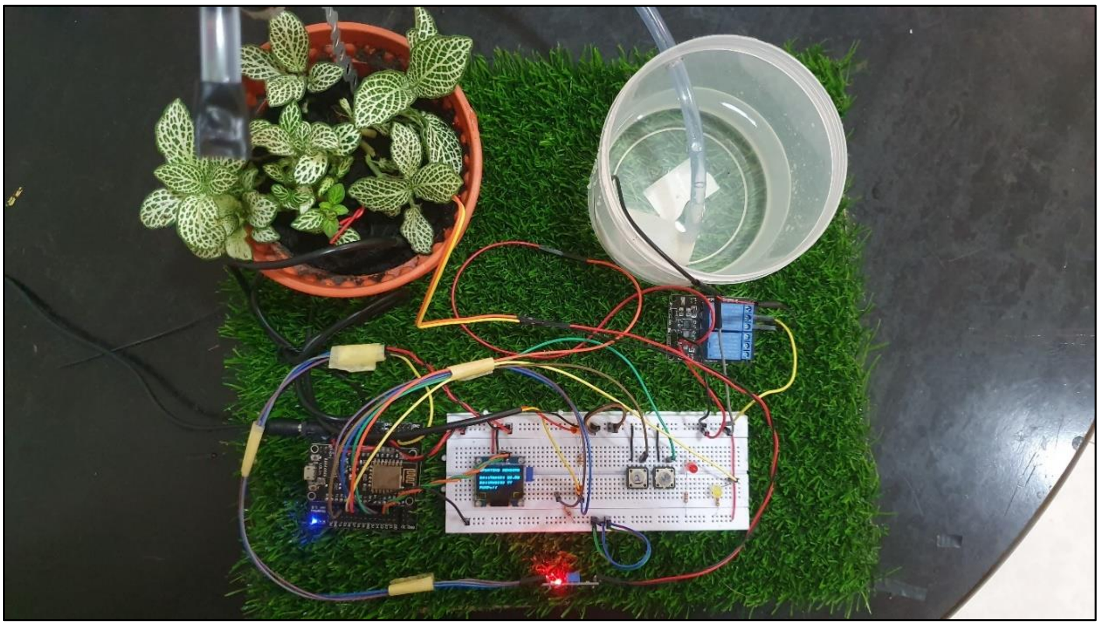


Figure 5.4.1. The complete Prototype of the System

5.5.2 Software Blynk

The Blynk app is developed to monitor the automatic plant irrigation system, the figure 5.5.1 shows the main control page of the system while the next figure shows the configuration of the graph widget.

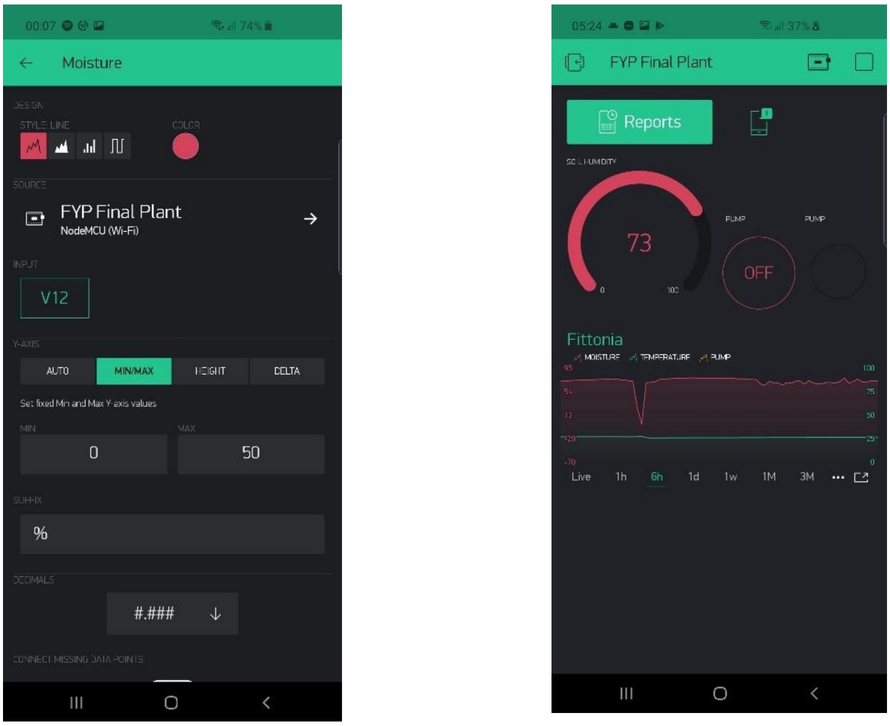


Figure 5.5.1 Blynk application interface

5.6 Implementation Setup

The implementation setup explained the status for each automatic plant irrigation system software and hardware development. Every development between prototype and software will be explained.

5.7 Conclusion

This chapter described about the implementation of the hardware and the system. Each step is described in details with the explanation. The step includes for the hardware setup is wiring the devices and make the connection to each of the requirement. The system setup is also described in details starting from what the application to use to begin the development until how to code the system. The diagram and figure in this chapter have been use to help the reader to understand about the development of the hardware and system. The next chapter will be chapter 6, the testing and analysis of the project will be discussed in detail there.

CHAPTER VI: TESTING

6.1 Introduction

The following chapter will discuss on the testing of the system. There will be three section in the test plans comprising of test environment and test schedule. The test will be conducted and distributed into different components to ensure the software system and the prototypes are running with zero errors. The chapter will include test design that includes analysis part to test the system functionality.

6.2 Test plan

The test plan is an extensive testing for the system that involves the procedure and scope of the system. The test plan will aid in rectifying the errors that is encountered when the system is running. In total there are two types of test in the project that are software testing and prototype testing. Prototype testing is more practical to ensure the runs smoothly. Software testing is to scan the system performances with the prototype and the appropriate data transfer.

6.2.1 Test Organization

Test organization explains on the responsible parties that are involved to conduct the testing process. In the project, system developer is the person who is an authority on hardware and software-based device testing. This person is already involved in the different test cases and will find any potential errors or developments.

6.2.2 Test Environment

In this sub section will discuss about the test environment of the project. It will be divided into two which is prototype and software.

i. Hardware

The initial part of the test involves the Node MCU Microcontroller and breadboard. Both this boards are connected to each other through jump wires. There is an OLED screen used as a local display to display the readings. The screen display function was tested. Then the sensor modules like soil temperature and soil moisture was connected and tested for its functionalities. Switches and relay modules are installed and connected to the DC pump and tested. C++ coding was programmed in Arduino IDE to enable the sensors and switch, record data into the Blynk cloud server for the monitoring system to be enabled through the Wi-Fi connection.

ii. Software

The next test is to setup the Blynk application on smartphone to enable the monitoring system and notify the user every time there is an activity on the plant irrigation. The users will be notified every time the plant is watered, a gauge meter to indicate the plant's moisture level and temperature is created, and a graph to monitor the levels of both this variables is created. The data is logged and exported into a csv file for analysis purpose.

iii. Network

The third test includes network monitoring, Node MCU with integrated Wi-Fi module and network setup which are both tested and can be used either mobile or wireless. The transmission data is tested from the wireless to wired media.

6.2.3 Test Schedule

In the test schedule describes how the testing was done during the testing phase by the system developer. During process if mistakes were to occur, the process will be carried back to the earlier stage which is the execution stage. The execution stage is where the trial and error checks is carried out to discover mistakes and solve them. The process is continued until the functionality scheme is developed.

6.3 Test strategy

This section describes about the test strategy to be use in this project. This project has three testing strategy which is, hardware, software and network. The hardware testing involves the functionalities and connectivity of all sensors, and electrical devices along with the microcontroller Node MCU. The software testing will involve to make sure the blynk application can interact with the hardware directly with the prototype without required for additional configuration for the user. Next, test is the network testing, the test will measure the connection between the hardware and the software and also other devices that have inside the network.

6.4 Test Design

The following section will elaborate on how to test the design system comprising of the test descriptions and test data.

6.4.1 Test description

The test descriptions describe the test cases and the results from the prototype and software for the functionality scenario of the sensors. In this case, the reading from both the Node MCU and the moisture sensors will be one test case testing.

6.4.2 Test Data

i. Hardware

The figure shows the connectivity of Arduino Node MCU with the moisture sensors using jumper wire and breadboard. The indicator LED light is to indicate the board power is on. Figure 6.1 shows the complete prototype of automatic plant irrigation system.

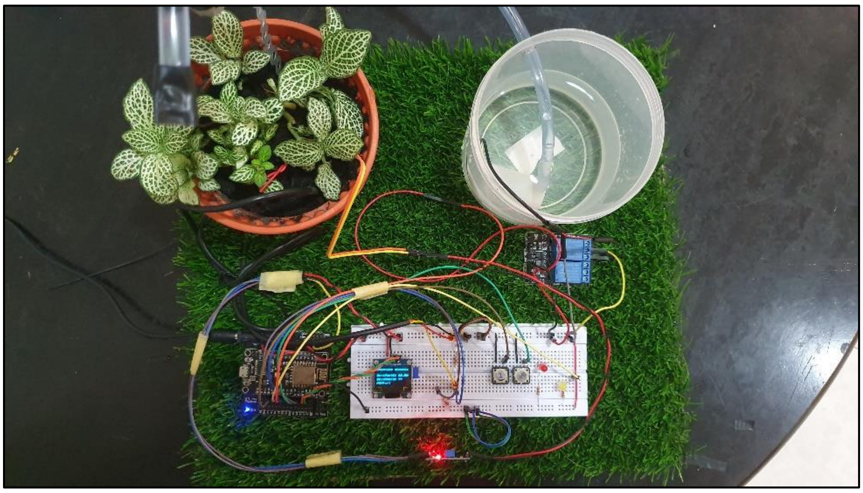


Figure 6.1 Complete Prototype of Automatic Plant Irrigation System

Table 6.1 Power connection of Node MCU

Test Case ID	TC001
Test Functionality	Test the power connectivity of the Node MCU with the power plug and USB
Precondition	The AC/DC adaptor connect to the power plug
Execution Steps	1. Connect the Node MCU with the mini USB power cable. 2. Switch on the plug.
Expected Results	The LED light in the Node MCU should light and blink to indicate the hardware is functioning.
Error Message	None
Result	Pass.

Table 6.2 OLED screen test with Node MCU

Test Case ID	TC002
Test Functionality	Test the connection between the Node MCU and the OLED screen.
Precondition	The Node MCU is switched one and USB connection is made to the computer. OLED screen is installed according to its schematic diagram.
Execution Steps	1. Connect the Node MCU with the computer through USB. 2. Upload the test code of the OLED screen to the Node MCU.
Expected Results	"Hello World" is printed on the display and preferred output can be printed according to our program.
Error Message	None
Result	Pass

Table 6.3 Soil moisture sensor test with Node MCU

Test Case ID	TC003
Test Functionality	Test the connection between Node MCU and the soil moisture sensor YL-69
Precondition	The moisture sensor is connected in the designated pin in Node MCU according to its schematic diagram.
Execution Steps	1. Connect the Node MCU with the computer through USB. 2. Upload the test code of the moisture sensor to the Node MCU.
Expected Results	Moisture sensor LED lights up and starts blinking. Reading is displayed on the OLED screen and Arduino's serial monitor.

Error Message	None
Result	Pass

Table 6.4 Soil Temperature test with Node MCU

Test Case ID	TC004
Test Functionality	Test the connection between Node MCU and the soil temperature sensor DS18B20.
Precondition	The soil temperature sensor is connected in the designated pin in Node MCU according to its schematic diagram.
Execution Steps	1. Connect the Node MCU with the computer through USB. 2. Upload the test code of the soil temperature sensor to the Node MCU.
Expected Results	Temperature reading is displayed on the OLED screen and Arduino's serial monitor.
Error Message	None
Result	Pass

Table 6.5 Button test with Node MCU

Test Case ID	TC005
Test Functionality	Test the buttons which act as manual switches on the Node MCU with the DC Pump.
Precondition	The buttons and DC pump setup is connected at the designated pin of Node MCU according to its schematic diagrams.
Execution Steps	1. Connect the Node MCU with the computer through USB. 2. Upload the partial test code of the switch and DC Pump to the Node MCU.
Expected Results	The relay module works on a toggle mode to switch on/off to trigger the DC pump. Water is pumped out of the water tube.
Error Message	None
Result	Pass.

ii. Software

Table 6.6 Moisture and Temperature Gauge Test on Blynk

Test Case ID	TC006
Test Functionality	Test the soil moisture and soil temperature gauge meter on blynk to display sensor data.
Precondition	The code is programmed to systematically send data every n-seconds once to blynk
Execution Steps	1. Open the blynk app and go to the menu then choose gauge to add a new gauge. 2. Label each gauge with their names respectively and assign them to appropriate digital output pin from Node MCU.
Expected Results	The moisture and temperature reading are displayed on the gauge and it refreshes every n-seconds.

Error Message	None.
Result	Pass.

Table 6.7 Moisture and Temperature Graph Test on Blynk

Test Case ID	TC007
Test Functionality	Test the soil moisture and soil temperature graph on blynk to display sensor data.
Precondition	The code is programmed to systematically send data every n-seconds once to blynk
Execution Steps	1. Open the Blynk app and go to the menu then choose graph to add a new graph. 2. Name the graph, and add the moisture and temperature, pump on the data stream. 3. Assign them to the appropriate pin output from the Node MCU.
Expected Results	1. The moisture and temperature reading are displayed on the graph and it refreshes every n-seconds. 2. The pump ON/OFF is connoted on the graph every time it is triggered and the duration is recorded.
Error Message	None.
Result	Pass.

Table 6.8 Pump switch and LED Test on Blynk

Test Case ID	TC008
Test Functionality	Test the Pump switch and the LED
Precondition	The code is programmed to systematically send data every n-seconds once to Blynk.
Execution Steps	1. Open the Blynk app and go to the menu then choose buttons to add a new button. 2. Name the button, and assign it to the appropriate pin output from the Node MCU. 3. Repeat the same steps to create an LED press the button once created.
Expected Results	1. Water is pumped out of the water tube. 2. LED Button lights up, on the Blynk app. 3. Pump shows “Act” when switched ON and “OK” when switched OFF.
Error Message	None.
Result	Pass.

Table 6.9 Notification Alert Test on Blynk

Test Case ID	TC009
Test Functionality	Test the Notification System on Blynk
Precondition	The code is programmed to systematically send data every n-seconds once to Blynk. The Node MCU is connected to Wi-Fi.
Execution Steps	The water pump button is pressed.
Expected Results	The Notification “Warning => Pump is on” message pops up.
Error Message	None.
Result	Pass.

Table 6.10 Notification System Test on Blynk

Test Case ID	TC010
Test Functionality	Test the notification system on Blynk
Precondition	The code is programmed to systematically send data every n-seconds once to Blynk. The Node MCU is connected to Wi-Fi.
Execution Steps	The System is switched OFF
Expected Results	The notification “System is Offline” message pops up.
Error Message	None.
Result	Pass.

iii. Network

Table 6.11 Wi-Fi Network Connectivity Test with Blynk

Test Case ID	TC010
Test Functionality	Test the connectivity between the Node MCU and the Blynk through Wi-Fi
Precondition	Node MCU is connected to the local Wi-Fi network according to the SSID and Password of the WiFi
Execution Steps	Start the Node MCU
Expected Results	The system is online on Blynk device.
Error Message	The Node MCU Led is not Blinking.
Result	Pass.

6.4.3 Hardware Test Data

a) TC001 - Test the power connectivity of the Node MCU with the power plug and USB

The 5V AC adapter wire and the USB wire is connected to the Node MCU and the power is switched on. The blue light powers up and begins blinking to indicate current passing through. USB connection is established to the laptop and the Arduino IDE is opened to check if the USB connection is detected under the port number. Once a test code is uploaded into the microcontroller, the transmission lights starts to blink constantly to indicate data transmission. Figure 6.2 shows the Node MCU connected to the power plug.

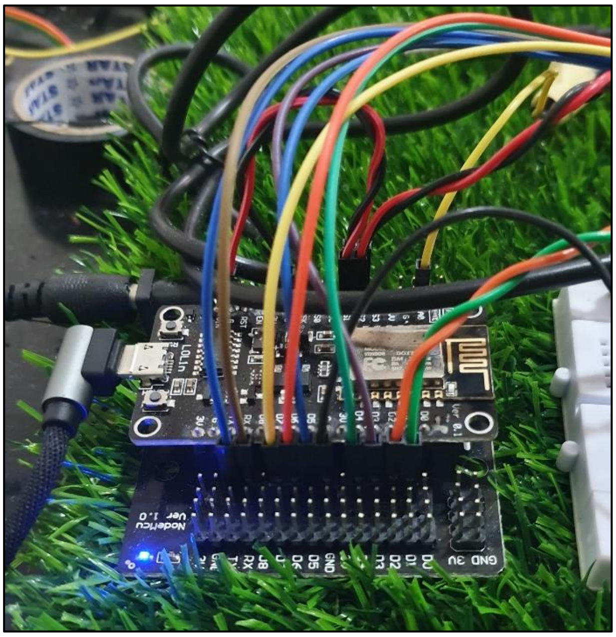


Figure 6.2 Node MCU connected to power plug and laptop connection

b) **TC002 - Test the connection between the Node MCU and the OLED screen.**

The OLED display is connected to the breadboard and jump wires was used to make the connection according to the schematic diagram and the test code was uploaded to test the display after configuration was done through the code. The outputs to display were programmed. Figure 6.3 shows the OLED screen output testing.

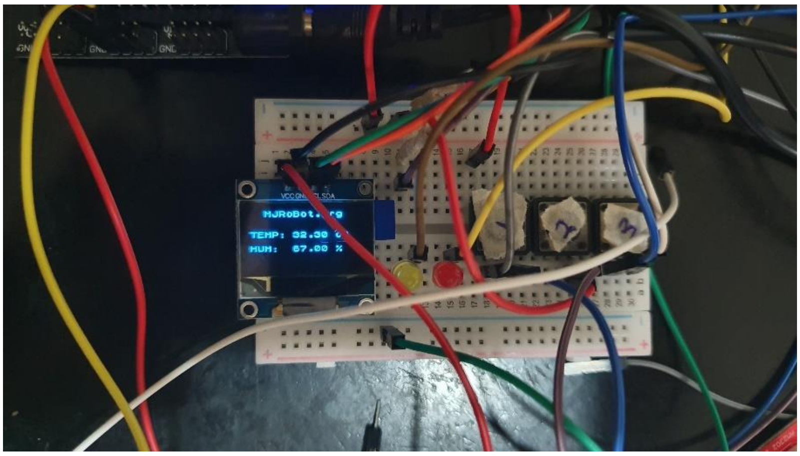


Figure 6.3 OLED screen output testing

c) TC003 - Test connection between the Node MCU and soil moisture sensor

Sensor calibration test was conducted Sensors for soil moisture return rather arbitrary values to be translated into something simple to use. The whole range for a soil moisture sensor based on Node MCU is 0-1023. Nevertheless, some of this range is lost in reality. For a percentage production, at least 100 wet and dry readings should be separated. Figure 6.4 shows the moisture sensor being calibrated.

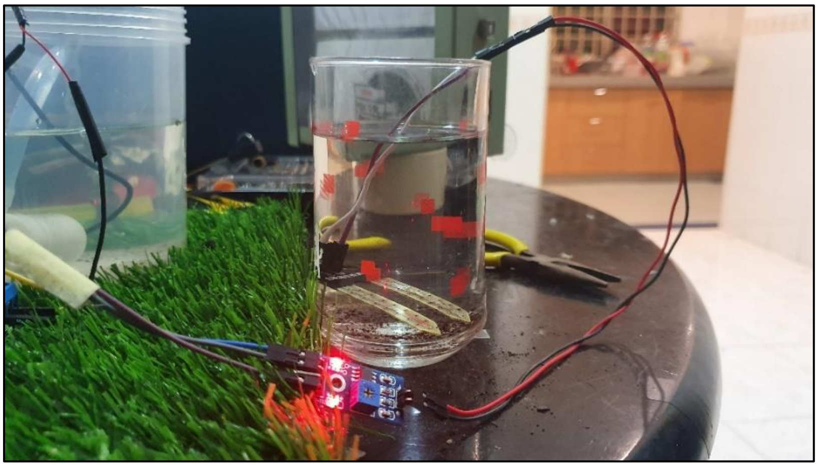


Figure 6.4 Moisture Sensor calibration to obtain maximum and minimum value.

The minimum and maximum values must first be determined for the sensor and the soil to be correctly calibrated. There may be different types of soil with different conductivity levels, which may require different calibration. Place the ground moisture sensor on dry soil and record the raw value. Place a sensor of soil humidity in wet soil and observe the raw value.

d) TC004 - Test the connection between Node MCU and the soil temperature sensor DS18B20.

The temperature sensor is connected to the breadboard and jump wires was used to make the connection according to the schematic diagram and the test code was uploaded to test the display after configuration was done through the code. The outputs to display were programmed. Figure 6.5 shows the soil temperature output testing.

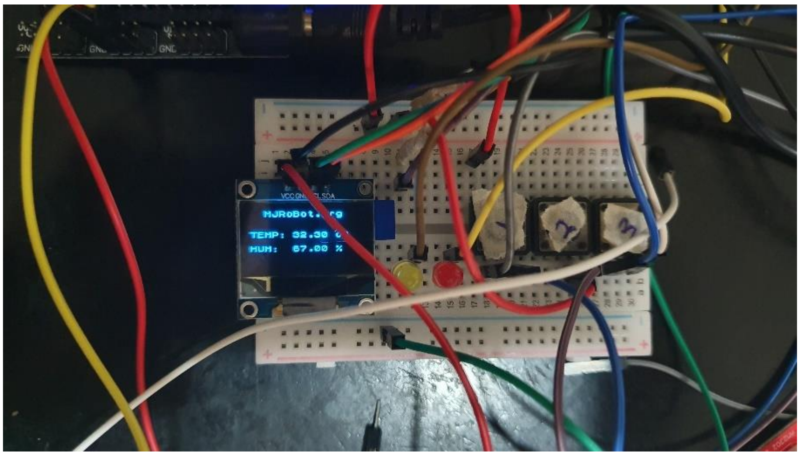


Figure 6.5 Soil temperature output testing

e) TC005 - Test the buttons which act as manual switches on the Node MCU with the DC Pump.

The buttons were fixed according to the schematic diagram, and the functionalities were tested. The pump releases water when the button is pressed, and the when the button is pressed again the pump stops. The relay module works on a toggle mode to switch on/off to trigger the DC pump. Figure 6.6 shows the water tube watering when the pump is triggered.

Figure 6.6 DC Pump is switched ON

6.4.4 Software Test Data

a) TC006 - Test the soil moisture and soil temperature gauge meter on Blynk

The code is programmed to systematically send data every n-seconds once to Blynk. Each labelled gauge with their names respectively and assign them to appropriate digital output pin from Node MCU. The moisture and temperature reading are displayed on the gauge and it refreshes every n-seconds. Figure 6.7 shows the soil moisture and temperature gauge on Blynk.

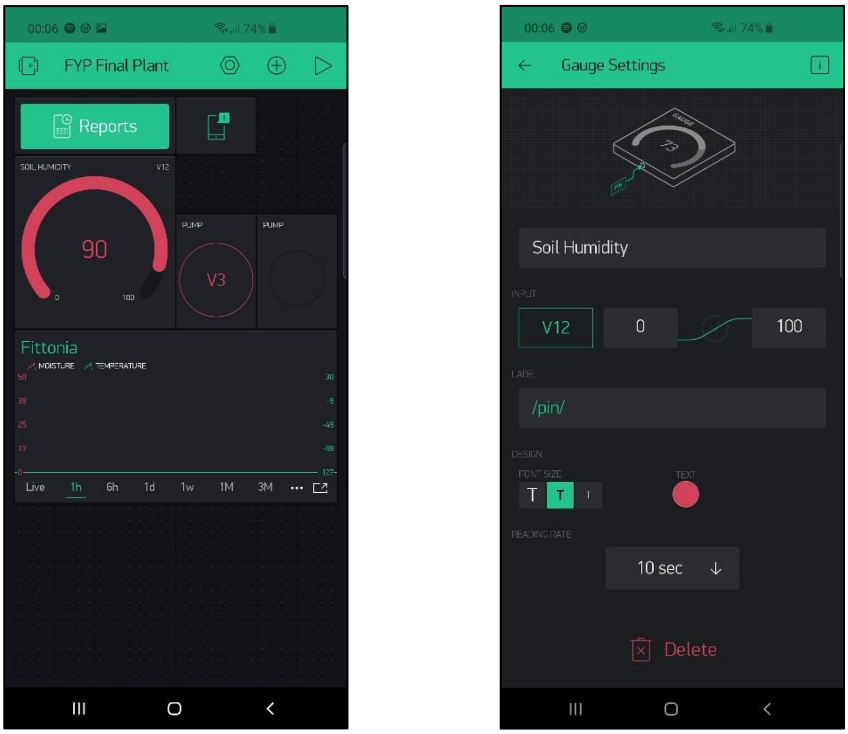


Figure 6.7 Soil Moisture and Temperature Gauge

b) TC007 - Test the Soil Moisture and Soil Temperature graph on Blynk.

The code is programmed to systematically send data every 600-seconds once to Blynk. The labelled graph, is checked for its sensitivity. The moisture and temperature, pump data are all added on the data stream. The moisture and temperature reading are displayed on the graph and it refreshes every 600-seconds. The pump ON/OFF is connoted on the graph every time it is triggered and the duration is recorded. Figure 6.8 shows the soil moisture and soil temperature graph on Blynk.

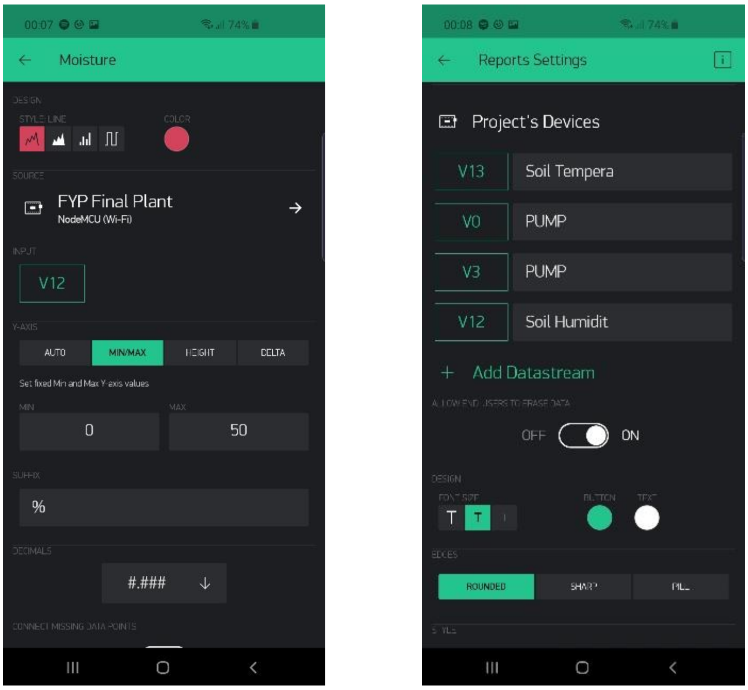


Figure 6.8 Soil Moisture and Temperature Graph

c) TC008 - Test the Pump switch and the LED

Water is pumped out of the water tube. LED Button lights up, on the Blynk app. Pump shows “Act” when switched ON and “OK” when switched OFF. Figure 6.9 shows the Pump switch and LED on Blynk.

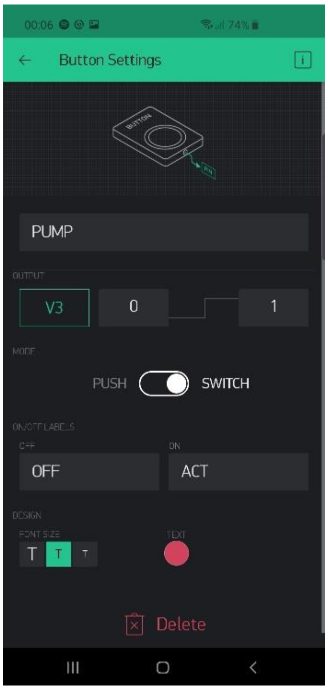


Figure 6.9 Pump Switch and LED on Blynk

d) TC009 - Test the Notification System on Blynk

The water pump button is pressed. The Notification “Warning => Pump is on” message pops up. The system goes offline and the notification “System is now offline” message pops up. Figure 6.10 shows the notification system on Blynk.

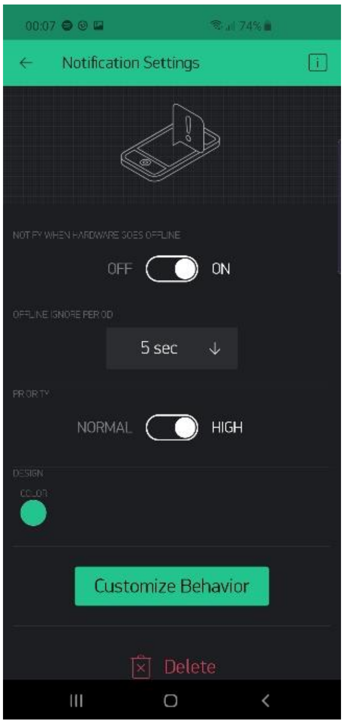
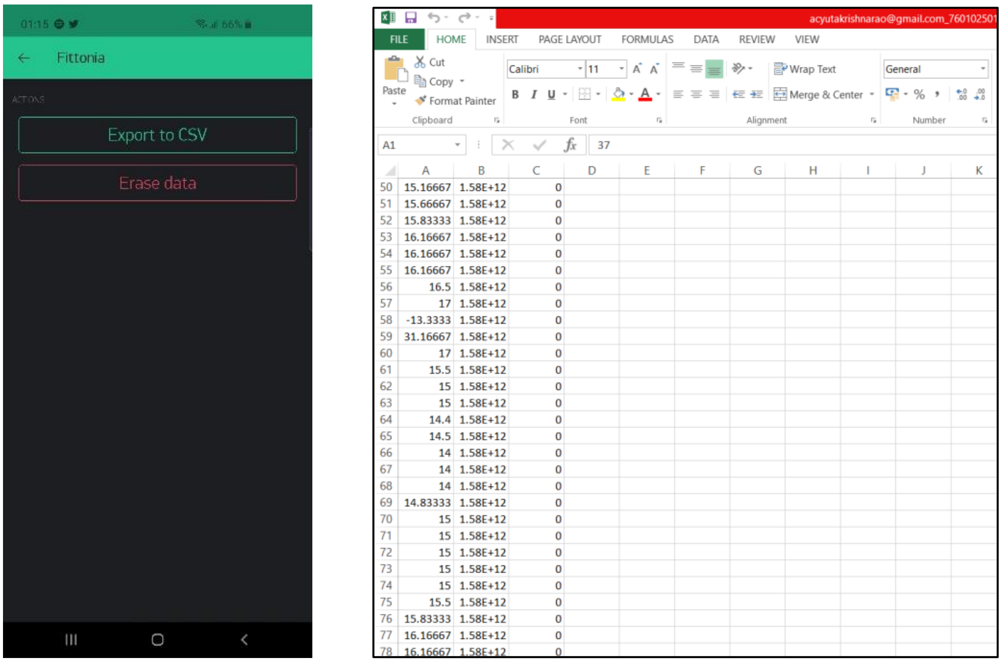


Figure 6.10 Notification System on Blynk

e) TC010 - Generate Report and Export as CSV file.

The report generate function was tested and the report was exported to csv file for further data processing and analysis. The csv file opened in the excel contains the moisture reading on the first column and the time measurement on the second column while the third column shows the pump on and off column. Figure 6.11 shows the export as CSV file function.



	A	B	C
50	15.16667	1.58E+12	0
51	15.66667	1.58E+12	0
52	15.83333	1.58E+12	0
53	16.16667	1.58E+12	0
54	16.16667	1.58E+12	0
55	16.16667	1.58E+12	0
56	16.5	1.58E+12	0
57	17	1.58E+12	0
58	-13.3333	1.58E+12	0
59	31.16667	1.58E+12	0
60	17	1.58E+12	0
61	15.5	1.58E+12	0
62	15	1.58E+12	0
63	15	1.58E+12	0
64	14.4	1.58E+12	0
65	14.5	1.58E+12	0
66	14	1.58E+12	0
67	14	1.58E+12	0
68	14	1.58E+12	0
69	14.83333	1.58E+12	0
70	15	1.58E+12	0
71	15	1.58E+12	0
72	15	1.58E+12	0
73	15	1.58E+12	0
74	15	1.58E+12	0
75	15.5	1.58E+12	0
76	15.83333	1.58E+12	0
77	16.16667	1.58E+12	0
78	16.16667	1.58E+12	0

Figure 6.11 Generate Report and Export as CSV on Blynk

6.5 Test Result and Analysis

The following section will elaborate on how to test the design system comprising of the test descriptions and test data. The test analysis was done by emphasizing how well the system functions indoor. This is essential to ensure that the quality and utility of the prototype is guaranteed. The categories that were evaluated were, Irrigation control, electric consumption, water consumption and plant growth rate.

6.5.1 System Developer Test result

From the test description and test data that done by the system developer, all test result will be describe in this section. The test result includes all the test environment which area hardware, system and network.

a) Hardware

Test Case ID	Test Description	Fail	Pass
TC001	Test the power connectivity of the Node MCU with the power plug and USB		/
TC002	Test the connection between the Node MCU and the OLED screen.		/
TC003	Test the connection between Node MCU and the soil moisture sensor YL-69		/
TC004	Test the connection between Node MCU and the soil temperature sensor DS18B20.		/
TC005	Test the buttons which act as manual switches on the Node MCU with the DC Pump.		/

Table 6.12 Hardware Testing

b) Software

Table 6.13 Software Blynk App Testing

Test Case ID	Test Description	Fail	Pass
TC006	Test the soil moisture and soil temperature gauge meter on Blynk.		/
TC007	Test the soil moisture and soil temperature graph on Blynk.		/
TC008	Test the Pump switch and the LED on Blynk.		/
TC009	Test the Notification System on Blynk.		/
TC010	Generate Report and Export as CSV file.		/

6.5.2 Irrigation Control

There are different modes that can be set to the auto irrigation control to ease the watering duration and instances of the plant. In this project, the duration for watering of the plant is set to 10 seconds and 5 seconds alternatively, threshold value for hot temperature is set at 34°C, and cold temperature is set at 26°C. Dry soil level is set at 20% and wet soil is set at 65% in the case of the plant being over watered. Optimal soil level is around 35%-60%. The program is customizable to different parameters and irrigation values. The pump will be triggered every time the soil moisture level drops below the threshold value of 20%. The below setting was used to test the system.

Table 6.14 Automatic Irrigation System Setting

Soil Moisture	Level (%)	Time (Sec)
Dry Soil	0%-20%	-
Optimal Soil	35%-60%	-
Wet Soil	65%-85%	-

Auto Pump ON	-	60 Seconds
OLED Refresh Rate	-	60 Sec/1 min
Send Data to Blynk	-	600 Sec/5 min
Watering Duration		5 Seconds

After all the system are completely set up, the prototype is tested and run to observe if the control system can be functioned and to collect data about the difference in soil moisture level and how the automatic irrigation responds to the difference.

For the system was mostly used indoors the main systems used was the auto watering system. Two different threshold values for dry moisture was used to test the irrigation system's response which are 10% and 15% respectively and results are recorded and reflected in the Blynk graph. The plant was placed in closed environment indoor with minimal sunlight and 24 hours fluorescent light exposure to aid the plants in carrying out photosynthesis as the Fittonia plant thrives under minimal light.

a) Soil Moisture On 1 Hour daily chart using 10% moisture as dry threshold value – Observation A

The daily chart shows hourly measurement of the plants moisture level on December 13th 2019 using 10% moisture as dry threshold value. The moisture level of the plants drops during the evening, the graph indicates the soil moisture level has dropped to 9.33% while the temperature is recorded at 29.5 degree. Once the soil moisture drops below 10%, the pump was triggered and the plant is watered right after that for 10 seconds through the water tube and the subsequent readings have evidently gone up where the soil moisture level shows steep increase in readings. Figure 6.12 shows the hourly chart using 10% moisture as dry soil threshold.

Figure 6.12 1 Hour daily chart using 10% Moisture as Dry soil Threshold

The subsequent measurement recorded at 1.00am on 13th December 2019, which is 1 hour after the pump was triggered shows an increase in the soil moisture level. The new reading shows 74.429% soil moisture levels and the temperature is maintained at 29.5 degree. The increase in soil moisture level indicates the plant has reached the wet level and therefore no further irrigation will be conducted. Figure 6.13 shows the changes in the hourly chart using 10% moisture as dry soil threshold.

Figure 6.13 1 Hour daily chart using 10% Moisture as Dry soil Threshold

b) Soil Moisture on 1 hour daily chart using 15% moisture as dry threshold value - Observation B

The daily chart shows hourly measurement of the plants moisture level on December 11th 2019 using 15% moisture as dry threshold value. The moisture level of the plant was low during the afternoon, the graph indicates the soil moisture level has dropped to 12.46% while the temperature is recorded at 28.4 degree. Once the soil moisture drops below 15%, the pump was triggered and the plant is watered right after that for 5 seconds and the subsequent readings have evidently gone up where the soil moisture level shows increase in readings. Figure 6.14 shows the 1 hour daily chart using 15% moisture as dry soil threshold value.

Figure 6.14 1 Hour daily chart using 15% Moisture as Dry Soil Threshold

The subsequent measurement recorded at 1.00pm on 11th December 2019, which is 1 hour after the pump was triggered shows an increase in the soil moisture level. The new reading shows 21.650% soil moisture levels and the temperature is maintained at 28.5 degree. The increase in soil moisture level indicates the plant has reached the optimal level and therefore no further irrigation will be conducted. Figure 6.15 shows the changes in the 1 hour daily chart using 15% moisture as dry soil threshold value.

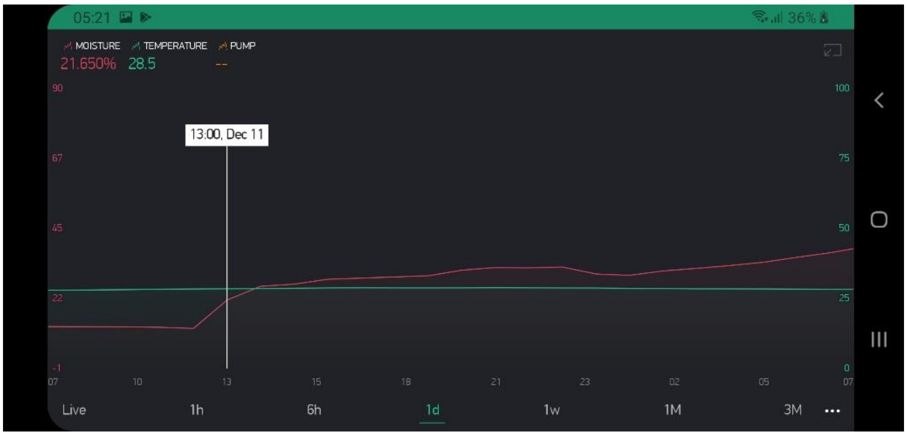


Figure 6.15 1 Hour daily chart using 15% Moisture as Dry Soil Threshold

c) Soil Moisture on Weekly Chart using 15% moisture as dry threshold value – Observation C

The weekly chart shows daily measurement of the plants moisture level on December 9th and December 10th 2019 respectively using 15% moisture as dry threshold value. The moisture level of the plant at 8.00am on 9th was 21.57% while the temperature was at 28.2 degrees. The following day the moisture level dropped to 12.83% while the temperature measured at 28.4 degrees. The automatic irrigation control is waiting to be triggered. Once the pump was triggered later in the day the subsequent readings have evidently gone up where the soil moisture level shows increase in readings. Figure 6.16 shows the weekly chart using 15% moisture as dry threshold value.

Chart 6.16 Weekly chart using 15% Moisture as Dry Soil Threshold

The subsequent measurement recorded the following day at 8.00am on 11th December 2019, which is day after the initial reading and after the auto pump was triggered shows an increase in the soil moisture level. The new reading shows 21.209% soil moisture levels and the temperature is measured at 28.5 degree. The increase in soil moisture level indicates the plant is slowly reaching the optimal level while the irrigation continues to happen until optimal level is reached. Figure 6.17 and figure 6.18 shows the changes in the weekly chart using 15% moisture as dry threshold value.

Chart 6.17 Weekly chart using 15% Moisture as Dry Soil Threshold

Chart 6.18 Weekly chart using 15% Moisture as Dry Soil Threshold

6.6 Discussion of Test Results

The test results show the response of the automatic irrigation system to the different level of moisture in the soil according to the threshold value that is programmed. The test was conducted with 2 different threshold values for dry soil that is 10% moisture and 15% moisture, and the difference and responsiveness of the irrigation system is recorded in the graph and logged.

Observation A records the responsiveness of the irrigation system when set at 10% dry moisture level as threshold value. The graph indicates the moisture level dropped to a 9.33% and the automatic irrigation system responded by watering the plant and the value increased until the soil becomes wet to 74.49% moisture level due to 10 seconds watering duration and therefore no further irrigation was conducted.

Observation B records the responsiveness of the irrigation system when set at 15% dry moisture level as threshold value. The graph indicates the moisture level dropped to a 12.46% and the automatic irrigation system responded by watering the plant and the value increased until the soil recovers to an optimal 21.650% moisture level due to 5 seconds watering duration and therefore no further irrigation was conducted.

Observation C records the responsiveness of the irrigation system when set at 15% dry moisture level as threshold value. The graph indicates there was a drop in the moisture level measured across 2 days when measured from 21.57% to 12.38%. The pump was triggered later during the day and the moisture level rise significantly and the following day measurement shows an increase at 21.209%.

All the conducted tests indicate the automatic plant irrigation is working according to the parameters, threshold value and principles programmed. The monitoring system on Blynk reflects the reading and measurement accurately. All the data was logged in a spreadsheet and exported to local library for data processing and analysis.

6.6.4 Water consumption

The plants only need to be watered when the moisture level of the soil drops below threshold value and the volume of the water reservoir tank can contain maximum of 1.824 litres which each watering of the plants took half of the maximum amount of reservoir tank. Then, taking the recent tariff of domestic water supply in Malaysia which is around RM0.57 for the first $20m^3$, the total consumed water for 31 days is calculated as below. Assuming that 1.824 litres is equivalent to $0.001824m^3$.

$$0.001824m^3 \text{ x } 31 \text{ days} = 0.056544m^2$$

The total price of the consumed water is equivalent to the calculation as below. This shows that in average the fertilizer tank consumes 3 cents per month for the water cost.

$$0.056544m^2 \text{ x RM}0.57 = \text{RM}0.03$$

It can be derived that this designed system is water saving as it only consumes 3 cents per month for the water cost.

6.7 Conclusion

In a whole, the chapter discusses all the functionality test and analysis of the results. The tests ensure the system works smoothly and the results that have been analysed and concluded based on the test organization of the prototype and the software. The next chapter discusses on the conclusion of the project.

CHAPTER VII: CONCLUSION

7.1 Introduction

This chapter describes the overall project summarizing the project details from the beginning of the project comprising the purpose of the project until the testing procedure of the project. The objective for this project was successfully achieved and in this chapter, it will discuss about the project weakness and the strength of the project. Project contribution and limitation will also describe in this chapter. The project will be improvisation in the future is discussion in this chapter.

7.2 Project Summarization

This project is about the setup and configure the hardware and software to produce a low-cost complete system that will support automated indoor plant watering system. The system is developed to control the plant irrigation according to the soil condition, which is connected to the Blynk application for monitoring and remote controlling purpose. The system will help to ease the usage and management of the Arduino MCU Node.

The first objective of the project is to develop an automatic plant watering system for indoor gardening by using an Arduino Node MCU Microcontroller. The second objective is to implement an auto watering system using soil moisture and temperature sensors. Third objective is to integrate and nalyse the performance of plant moisture and monitoring the system using Arduino Node MCU with Blynk.

The system was successfully tested and it fulfils all the objective. As the conclusion, the setup of the hardware and software will integrate each other and produce a complete solution to be implement in homes.

7.2.1 Project Limitations

- **The Implementation of DHT22 Air Temperature and Humidity Sensor.**

Not incorporating the use of air temperature and humidity sensors to provide optimal condition and get more precise environment to stimulate growth rate of the plant due to time limitation and more resource needed to setup, implement and test the outcome. More time needed to monitor the plant's growth rate according to its variables.

- **Test was conducted with one type of plant (Fittonia Albivenis).**

The test was done only on one plant hence the results may vary with different plants and requires more testing under different conditions, every plant has its own sensitivities and its own optimal conditions to thrive. Therefore a much bigger setup is needed.

- **Time Constraints**

The second constraint is the lack of time for the project development. The development of the project involves both software and hardware concurrently, and it takes time to develop software and hardware as it is set up in a different environments and platforms while all the processes and results need to be documented accordingly.

- **Budget Constraint**

The need to operate under strict budget while the purchasing cost of the project items and gadgets were expensive, and the cost of needing to fine tune the prototype was significantly costly hence there was restriction to expand the project to accommodate more than one plant.

7.2.2 Project Strength

- **Less Power consumption.**

Arduino MCU Node use less power compared to other microcontrollers, and the electricity consumed by the setup was significantly lower compared to most other setups and with less power consumption it may lead to the cost saving.

- **Minimised usage of Water**

Traditional methods of watering may typically waste 50% of the water used because of irrigation, evaporation and overwatering inefficiencies. Intelligent irrigation systems use sensors to warn watering routines in real time or historical data to adjust watering schedules in order to improve performance. It can be derived that this designed system is water saving as it only consumes 10 cents per month for the water cost.

- **Scalable to Adapt**

The system is scalable to accommodate different models of plants, and can be tweaked to suit different plant conditioning. The application is also highly customizable and scalable to navigate more than one plant. Direct control can be established to the irrigation system through the Blynk application.

7.3 Project Contribution

The complete combination of the Arduino MCU Node hardware and software which is aim to. The system can be expanded for indoor home gardening and possibly for outdoor irrigation and bigger farms. The system can be enhanced to use in agriculture and urban farming considering hydroponic contexts and etc.

In future, the system can be implemented on a bigger scale by experimenting and more plants and bigger sensors as they sense bigger volumes of plants.

7.5 Future Works

There are a few things that can be added in the future to improve the project for instance

- Testing on more than one Plants

  The project can be executed by testing and different plants to check the efficiency of the system as well as how it will positively affect the growth of the plant.

- Incorporating More Sensors

  Usage of more sensors such as the air temperature and air humidity sensors to ensure a precise and exact readings are obtain to get a better understanding on how to improve the system further.

- Location Variation

  The automated system may be tested outdoors to assess the balance of natural light and automated watering system.

- Incorporating Mobility to the System

  Future projects may investigate on how to generate a mobile automated watering system that may help for indoor environments for certain plants that require natural sunlight for growth.

7.6 Conclusion

As conclusion, purpose of this project is to develop prototype and software which that can enable an automated watering system to be successfully set up. This project may in mass plantation and the same time the wellbeing of plants are emphasized. The project meets all the objective as stated in chapter 1. Optimistically, this project can help in the improvement of the plantation sector in the long run.

REFERENCES

Adeodu, A. O. (2019). Development of an autonomous mobile plant irrigation robot for semi structured environment . *2nd International Conference on Sustainable Materials Processing and Manufacturing*, 1-7.

Agbetuyi, A. F., Awelewa, A., Evwieroghene, O. h., & Tita, S. W. (2017). Design and implementation of an automatic irrigation system based on monitoring soil moisture. *Journal of Electrical Engineering*, 2-11.

Agrawal, L., Dewangan, B., Deshmukh, S., Yadav, P., & Chandrakar, M. K. (2018). Smart Irrigation System Using 8051 Microcontroller and Fencing Control Using Bluetooth Module. *International Advance Research in Electrical ,Electronics, and Instrumentation Engineering, Vol.7, Issue 5*, 1-6.

Bisen, N. T. (2018, Jun 8). *Project Hub*. Retrieved from create arduino.cc: https://create.arduino.cc/projecthub/neetithakur/automatic-plant-watering-system-using-arduino-uno-8764ba

Duzic, N., & Dumic, D. (2017). Automatic Plant Watering Systems via Soil Moisture Sensing by means of suitable electronics and its application for anthropological and medical purposes. *Original scientific paper; Coll Antropol*, 2-5.

F.B, D. (1977). A PRECISE AUTOMATIC WATER DISPENSER FOR PLANTS GROWING IN POTS. *Canadian Journal of Plant Science*, 1-2.

Khumbar, S., & P.Ghatule, A. (2013). Microcontroller based Controlled Irrigation System for Plantation. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol II,* , 1-4.

McConnell, G. M. (n.d.). *Growing Indoor Plants with Success.* Georgia: The University of Georgia.

of, D. (2015). *Design of a Smart Indoor Gardening System Via Pid.*

Shifa, T. K. (2018). Moisture Sensing Automatic Plant Watering System Using Arduino Uno. *American Journal of Engineering Research (AJER)*, 1-5.