UNIVERSITI TEKNOLOGI MARA

IoT Based Monitoring Hydroponic Wick System

FAKHITAH BINTI AZMAN

BACHELOR OF SCIENCE (Hons.)

INFORMATION TECHNOLGY

July 2020

SUPERVISOR’S APPROVAL

IOT BASED MONITORING HYDROPONIC WICK SYSTEM

By

FAKHITAH BINTI AZMAN
2017668934

This thesis is prepared under the direction of thesis supervisor, En. Mohammad Hafiz bin Ismail. It was submitted to the Faculty of Computer and Mathematical Sciences and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Science (Hons) Information Technology.

Approved by:

……………………………….
En. Mohammad Hafiz bin Ismail
Thesis Supervisor

JULY, 2020

STUDENT DECLARATION

I certify that this report and the research to which it refers are the product of my own work and that any ideas or quotation from the work of other people, published or otherwise are fully acknowledged in accordance with the standard referring practices of the discipline.

……………………………….

FAKHITAH BINTI AZMAN

201766893

ACKNOWLEDGEMENT

With the assistance and co-operation of various authorities, this project has been successfully carried out. Hence, I wish to express my greatest appreciation to those people who guided, helped and supported in completing in this project. First of all, I would like to thank Universiti Teknologi Mara (UiTM) for giving me the opportunity to involve in this project. Throughout this project, I gain a lot of experience and knowledge which definitely going helpful in future. Secondly, I would like to express my deepest gratitude to my supervisor, En Mohammad Hafiz bin Ismail, for his guidance in this project. Special appreciation also goes to my beloved parents, En Azman bin Ali and Puan Wan Rodiyah binti Wan Majid for their motivation and financial support in this project. Last but not least, I would like to give my gratitude to my friends for helping me by discussing, sharing and exchange ideas and everyone who are directly involved in this project.

ABSTRACT

Hydroponic is a modern agriculture which can produce healthy plant and vegetables. Most people nowadays are willing to have garden at their house. However, most people have difficulty in growing plants at their house. It is because of the limited area for planting at their house. Besides that, they have difficulties in determining how much water is needed for the plants as well as the presence of unpredictable weather which can cause the plant to become wilt. To solve these problems, by developing the IoT Based Monitoring Hydroponic Wick System. IoT Based Monitoring Hydroponic Wick System helps by monitoring temperature, humidity, water level and soil moisture. Sensors that were used in this project are DHT22, water level sensor and soil moisture sensor. All the sensors are controlled by a microcontroller, NodeMCU ESP8266 and the data are stored in MySQL database. In addition, the user can monitor status hydroponic on dashboard which display the current and past data.

TABLE OF CONTENT

CONTENT **PAGE**

LIST OF FIGURES

LIST OF TABLES

TABLE **PAGE**

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter provides the background of the research and motivation to develop the IoT Based Hydroponic Wick System. This chapter also provides about the problem of the current situation relating to the project, objectives of the project and project significance.

1.2 Background of Study

Indoor garden is an act of growing plants inside the house, restaurant or any enclosed area. It is a way of overcoming the lack of natural feel by planting plants inside an enclosed place. This technique is beneficial for those living in apartments or home where there is no space for a garden. Besides, it also gives beneficial to the human health in growing plant inside their house.

Plants are use sunlight and chemical inside the leaves which called chlorophyll to convert carbon dioxide and water into glucose and oxygen. This process called photosynthesis. Chemically the equation for photosynthesis process is:

$$\underset{\text{Carbon dioxide}}{6CO_2} + \underset{\text{water}}{6H_2O} \xrightarrow[\text{chlorophyll}]{\text{Sunlight}} \underset{\text{carbohydrate}}{C_6H_{12}O_6} + \underset{\text{oxygen}}{6O_2}$$

Figure 1.1 Photosynthesis Process

Based on Figure 1.1, the equation shows what is needed for plant to growth which is water and light. So, the water level sensor and light sensor are required to monitor the plant. Since, the plant release water, temperature and humidity sensor also required to monitor the environment of plant. Other than that, soil moisture sensor also required to monitor the moisture of plant soil.

The benefit for hydroponic is most plant can be generated twice as quickly in a well-manage hydroponic system. Besides, hydroponic is the indoor farming in a climate-controlled environment which means farms can exist in places where weather and soil conditions are not favourable. There are many types of hydroponic which is wick system, drip system, ebb and flow system, deep water culture system and nutrient film technique (NFT)(Lee & Lee, 2015). There are a lot of benefit in hydroponic.

Wick system is water or nutrient solution supplied in a reservoir through a wick or fibrous materials (usually nylon) capable of absorbing and transporting water from the reservoir to the root area by capillary action(Lee & Lee, 2015). This project focus on wick system because it is easy for develop and saving cost.

In this project, the technique use is Internet of Things (IoT) for hydroponic wick system. IoT is to monitor the plants with different types of sensor including temperature and humidity sensor, water level sensor, light sensor and soil moisture. The implementing of IoT has allowed user to monitor the hydroponic culture. The temperature sensor deployed over the hydroponics wick system can sense the temperature loss and alert the user accordingly(Mehra et al., 2018). For humidity, it can measure the moisture of plant which is important for plant to growth. Besides, it also implements water level sensor that can notify the user if the level of water decrease. Light sensor also important for the measure plant light level and soil moisture sensor also required to monitor the moisture of plant soil. By implementing IoT in this project, it will notify to the user about the condition of plant.

Thus, this project proposes to develop an IoT Based Monitoring Hydroponic Wick System. This project which will help the user to monitor the condition of plants in real-time.

1.3 Problem Statement

The main problem for this project is limited area for planting especially for people who lives in the city area. People who live in the city mostly living in condo and apartment. Besides, the management of condominium does not allow the resident for planting at the corridor area. In the city, it become compact with a development and a lot of people. Based on the research by Hafiz Talib, Aidil Azlin Abd Rahman, and Shahrizal Dollah (2018), area town has been used for development projects and its limited the agricultural activity.

Other than that, the unpredictable weather is also one of the problems which can negatively affect the plants. There are 5 major climate risks that cities are exposed which are sea-level rise, extreme events, health, energy use and water availability. Out of five of these risks, extreme events and water availability are major challenges to conventional farming methods. Extreme events such as droughts, heat-wave, wind-storms, and flood events make farming outdoor difficult to grow (Hunt & Watkiss, 2011).

Besides, the problem in planting plant is difficult to monitor the level of water. Non-self-watering planting also required daily attention to track soil moisture and water as appropriate. If it is also common to occur under and overwater if the plants are not tested often enough if there is too much water. People usually watering the plant without know the level of water. Then, the plant become wilt if the plant get more or less water. It also can affect the humidity of the plant.

1.4 Objectives

The aim of this project is to develop an IoT-Based Hydroponic Wick System to help the people for indoor planting. In order to achieve the aim, the following objectives must be achieved:

i. To design a web-based interface for status monitoring of indoor garden.
ii. To develop prototype of IoT monitoring system for hydroponics in small space and at any time to monitor the plant.
iii. To evaluate the functionality of IoT monitoring system for hydroponic using Blackbox Testing method.

1.5 Project Scope

This research is focusing on growing plant with hydroponic wick system using Internet of Things. The aim of this project is to help the people who are interested in home indoor gardening. The scope of this project is limited to 3 mini pots mint plant. It is more suitable for people live in the city which has limited space. Besides, this project enables to automatic environment monitoring for hydroponics with different type of sensors including temperature and humidity sensor, water level sensor, soil moisture sensor and light sensor. The system controller unit is embedded with Wi-Fi module to transmit data to the cloud database. Microcontroller will display the condition of the plants. The user can monitor their plant through the web-based in the real-time.

1.6 Significance

It is important to develop the IoT Based Hydroponic Wick system which may lead ease to people. The people will be able planting the plant by their own.

They can plant in inside their house. This project will give the benefit to the user especially who live in condominium or apartment since they limited space to planting. They also not required to water the plant every day. Lastly, this project also can help user to monitor the condition of plants in real-time which is the user can keep update about the condition of plant.

1.7 Summary

In a conclusion, this project develops the IoT Based Monitoring Hydroponic Wick System which can help the user to have indoor garden in their house. The research objective will guide to complete the project. Besides, there were several objectives will act as a guidance in completing the project in accordance to the scope. Lastly, this project will give a benefit to user to monitor their plant and researcher to learn and make some improvement to their project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses what has been done on previous research. Smart indoor gardening, hydroponic, related work with this project and development techniques to be used to develop the hydroponic wick system are the subtopic that will be discussed. This chapter contains a lot of information and research done through the internet, newspapers, paper works and articles.

2.2 Smart Indoor Garden

Indoor gardening is act of growing plants indoors. Due to the problem statement in Chapter 1 which is limited area for planting especially people for people who lives in a city area, unpredictable weather and difficult to monitor the level of water, smart indoor garden will help in solving the problem. Smart technology in indoor gardening such as temperature and humidity sensor, soil moisture sensor, light intensity sensor and water level sensor will help the people in monitoring their plant (Min & Park, 2018).

Indoor Garden benefits from overall health and quality of life, physical strength, flexibility in fitness, cognitive ability, and socialization (Tse, 2010). Indoor gardening can also improve air quality by filtering toxins, thereby reducing the risk of health problems with respiratory disorders and chronic headaches and irritation of the eyes (Min & Park, 2018). Environmental studies have shown that connecting with plants improves human mental well-being as well as productivity levels: and further suggests that our levels of empathy and compassion are even improved (Wang & MacMillan, 2013). In addition, indoor gardening can also provide the fresh ingredients in the kitchen to reduce the risk of chemical intake. Finally, stress reduction is the only advantage of indoor gardening when people are in a room with a few containerized indoor plants (Virginia I. Lohr, 2010).

2.3 Type of Hydroponic

Hydroponics is a method for growing soilless plants rather than using mineral nutrient solution in water solvent (Ullah et al., 2019). Hydroponics system are technologies of agriculture that use nutrient solutions instead of soil substrates (Lee & Lee, 2015). For hydroponic, there are 6 hydroponic types that are the wick system, the drip method, the ebb and flow system, the deep-water culture system, the nutrient film technique, and the aeroponic system. Figure 2.1 shows the type of hydroponic system.

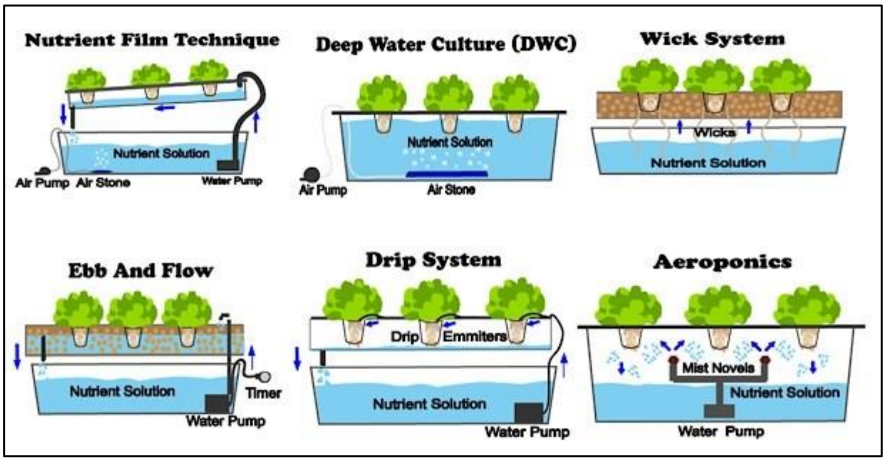


Figure 2.1 The Type of Hydroponic

Nutrient Film Technique systems have a constant flow of nutrient solution so no timer required for the submersible pump. The nutrient solution is pumped into the growing tray (usually a tube) and flows over the roots of the plants, and then drains back into the reservoir. Besides, Deep Water Culture is the simplest of all active hydroponic systems. The platform that hold the plants is usually made of Styrofoam and floats directly on the nutrient solution. Next, wick system is the simplest type of hydroponic system. This is a passive system, which means there are no moving parts. The nutrient solution is drawn into the growing medium from the reservoir with a wick. Other than that, Ebb and flow is the one of hydroponic type. Ebb and flow system work by temporarily flooding the grow tray with nutrient solution and then the draining solution back into the reservoir. This action is normally done with a submerged pump that is connected to the timer. Another hydroponic type is drip system. Drip system are probably the most widely used type of hydroponic system in the world. The operation of the drip system is simple, a timer controls a submersed pump. The timer turns the pump on the nutrient solution is dripped onto the base of each plant by a small drip line. The last one of hydroponic type is aeroponic system. Aeroponic system is probably the most high-tech type of hydroponic gardening. The roots hang in the air and are misted with nutrient solution. The misting is usually done every few minutes to the roots.

Each type of hydroponics has its own advantage. An outstanding model for indoor plant cultivation is for wick method or passive system. Water or a nutrient solution is supplied in a reservoir through a wick or fibrous substance (typically nylon) capable of absorbing and transporting water from the reservoir by capillary action to the root region (Lee & Lee, 2015). The wick system benefits from being easy and cheap to set up and maintain. Although it tends to keep the growing medium to humid, which does not allow the optimum oxygen content in the root system (*Types of Hydroponics Systems. - Soilless*, n.d.)

This project will develop the hydroponic wick system because it is suitable for short-term plant due to the problem statement in Chapter 1. In addition, the hydroponic wick system does not require a greater quantity of plant water.

Other than that, the hydroponic wick system is also acceptable to people who want to have an indoor garden that is not hard to monitor their plant.

2.4 Related Work

Keerthana, S. Devika, K. Sathiyadevi, S. Priyanka (2018) has developed the sensors, light-emitting diode lighting, water spray, and pump that could effectively lower the concentration of dioxide, temperature, and dramatically increase the water level. The paper also stated that in a greenhouse or indoor farming environment, hydroponic farming can be better controlled because of the surrounding conditions. The paper also stated that in a greenhouse or indoor farming environment, hydroponic farming can be better controlled because of the surrounding conditions. Through ensuring that the plant receives all nutrients from the water solution, the unit area parameters are mechanically regulated. In addition, by using IOT software, cultivator can apprehend the conditions of plant growth and remotely control the parameters. For each plant 1 and plant 2 this project implemented Arduino microcontroller with 3 detector forms such as temperature sensor, pH detector and LDR. In addition, ESP8266 is a wi-fi module to communicate with the server using the internet of things, and GSM module is to communicate and relay is used to automatically switch on / off the pumping motor's water supply.

Rajkumar, Dharmaraj, & Scholar, (2018) proposed work presents a Hydroponic style of farming which is the method of growing plants without soil & sunlight. A farming method which needs lesser requirements in cost factor and also it easy to maintain and control the important factors such as light, water level temperature, and humidity throughout the year is needed. Based on Figure 2.2, this project has been implemented as a NodeMCU microcontroller kit that connects to the Wireless Sensor Network with the internet that can senses humidity, temperature and water level. To collect real-time data from the smart hydroponic farm, the following sensors and other peripherals are used. This project had used Blynk and ThingSpeak, the IoT-based APi, as the framework for controlling and tracking the smart hydroponic farm.

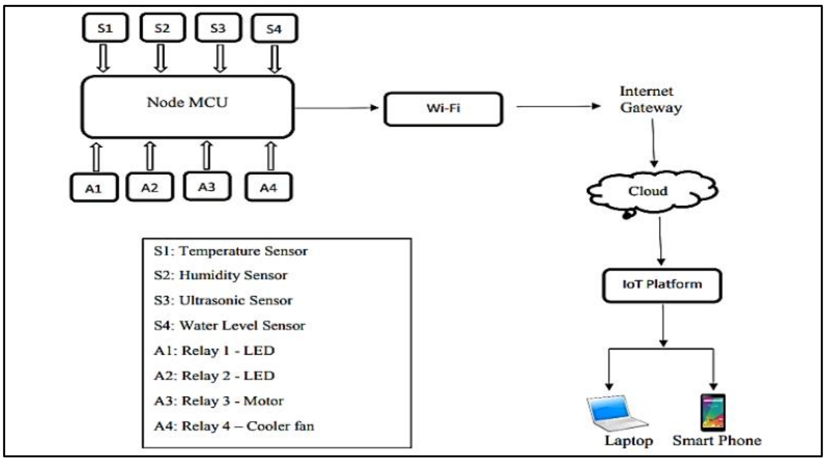


Figure 2.2 Proposed Architecture (Source: Rajkumar et al., 2018)

This paper proposes an IoT-based intelligent hydroponic plant factory solution called PlantTalk by Van et al. (2019). Van et al.(2019) The stated PlantTalk can flexibly configure connections through a smartphone for different plant sensors and actuators. Python programs for plant-care intelligence were conveniently written through the smartphone through this project. The plant-care intelligence includes automatic LED lighting, water spray, water pump and so on. This project enabled the gardener to monitor the sensor data in real-time via smartphone, and the user had to control the actuators manually. Figure 2.3 shows the functional block diagram of PlantTalk connected to the control board by the variety of sensor and actuator. To monitor the crop, the sensors were very critical. Temperature sensor, pH sensor, humidity sensor, CO2 sensor, O2 sensor, water level sensor and timer have been added to the control board. This project also enabled a camera to be integrated into their smartphone applications to observe the plants that the sensors and actuators took care of remotely. Connections of all plant IoT devices can be set up by using any smartphone's browser through the web-based PlantTalk GUI. The control board sends the data via Ethernet or Wi-Fi to the PlantTalk server.

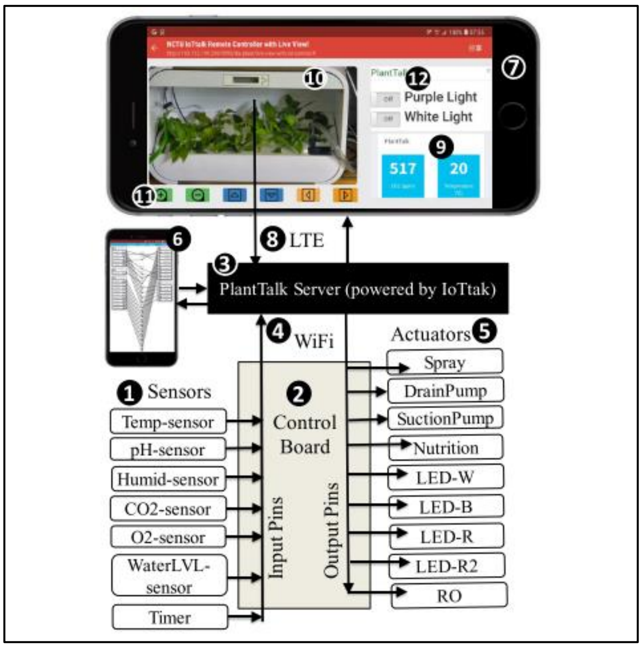


Figure 2.3 PlantTalk functional block diagram (Source: L. Da Van et al., (2019))

This project had been developed by Aliac & Maravillas, (2019) which process in pH, water level, air temperature and relative humidity are constantly monitored to provide the ideal environment for plants to grow. System provides irrigation controlled by water and nutrient intake. Users can store, manage, apply, and share information through the internet through the sensor-collected data and using cloud-based technology as the backend. The methodology of this project used Raspberry Pi to collect data from the server, transmit data from the temperature and humidity sensor, the pH level sensor, and the water level sensor. To monitor system nutrient conditions, water pump was also used to regulate machine water inflow and sensor temperature and humidity. The hardware control systems send data to Firebase in two modes, real-time and batch time. Information is then collected from the Firebase cloud service and used to view information on a website app. The app has been designed in a friendly desktop and mobile environment. In short, Raspberry Pi is used to monitor components in the hydroponics system based on the command obtained from the Google Firebase, a cloud service for real-time sensor logging.

Sisyanto, Suhardi, & Kurniawan, (2018) provide the collaborative work between hydroponic farmers and Cyber Physical Social System (CPSS) in their project. CPSS is a combination of physical and social systems that are transmitted through cyber or internet connection (Smirnov et al., n.d.). All sensors need to be incorporated with the use of Raspberry Pi. The introduction of the theory of CPSS made it possible for hydroponic farmers to operate in a team. The sensors used are light sensor, sensor of temperature and sensor of moisture, and sensor of nutrients. In addition, the Telegram messenger was also used as social media for interaction activities between farmer to farmer and farmer to a Bot. A Bot is a Python-based telegram bot for interaction between sensors and farmers.

IoT Planting consisting of a model gardening system with soil moisture, temperature, water sensor, growing light and Android for the elderly (Lekjaroen et al., 2016). This design has used Arduino Uno Wi-Fi, which has integrated Wi-Fi module, according to Figure 2.4. To the Arduino, which is DHT22, soil moisture and water detector, the sensors are applied. DHT22 has been used for temperature and humidity measurements. In addition, soil moisture was used to measure the volumetric water content at the root level in the soil stick and the water sensor used to detect water level. The grow light and water pump with power supply shown in Figure 2.4 to provide the light for the photosynthesis cycle and supply the plant with water respectively. The cloud technology and phpMYSQL were also used in this project to store data.

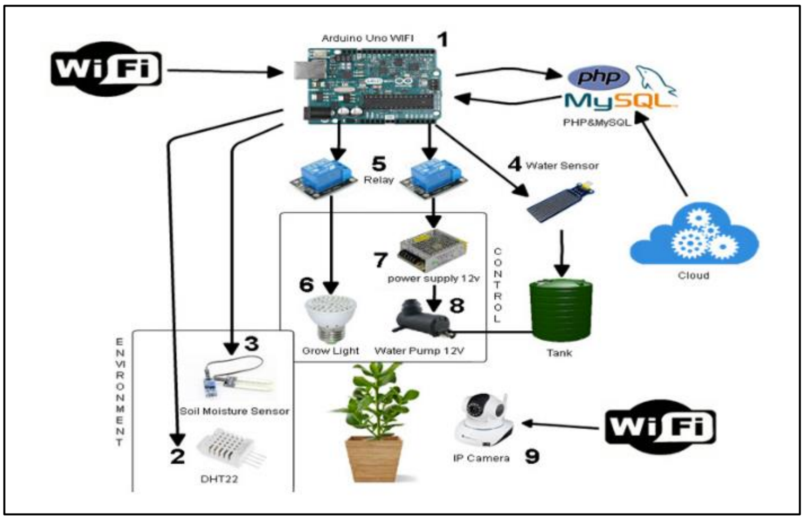


Figure 2.4 IoT Planting Components (Source: Lekjaroen et al., 2016)

Table 2.1 Comparison previous work

Title	Author	Category	Focus Area	Control Board	Type of Sensor
Automating and Analysing Greenhouse Hydroponic Farms using IOT	(Keerthana, S. Devika, K. Sathiyadevi, S. Priyanka, 2018)	Hydroponic	-	Arduino	• Temperature • pH • LDR
A Novel Approach for Smart Hydroponic Farming Using IoT	(Rajkumar et al., 2018)	Hydroponic	Home	NodeMCU	• Humidity • Temperature • Water Sensor

PlantTalk: A Smartphone-Based Intelligent Hydroponic Plant Box	(L.-D. Van et al., n.d.)	Hydroponic	Home	Arduino ESP8266 ESP-12F	• Temperature • pH • Humidity • CO_2 • O_2 • Water Level
IOT Hydroponics Management System	(Aliac & Maravillas, 2019)	Hydroponic	-	Raspberry Pi	• Temperature • Humidity • pH • Water Sensor
Hydroponic Smart Farming Using Cyber Physical Social System with Telegram Messenger	(Sisyanto et al., 2018)	Hydroponic	-	Raspberry Pi 3	• LDR • Temperature • Humidity • pH
IoT Planting: Watering System Using Mobile Application for the Elderly	(Lekjaroen et al., 2016)	Non-hydroponic	-	Arduino Uno	• Temperature • Humidity • Soil moisture • Water sensor

In addition, the hydroponic and non-hydroponic plant is being performed by many researchers. Although it has two types, Arduino, NodeMCU and Raspberry Pi use the variety of control board in their design. Microcontroller is the one that is important to the project's development. The microcontroller will be incorporated with a sensor. We also use the variety of sensors that can aid the growth of the plant. Temperature sensor, humidity sensor, pH level sensor, CO2 level, O2 level, water level sensor as well as LDR are the sensors they are used for the plant. The data will be transmitted from the sensor based on analysis will be processed in the cloud.

2.5 Notification

2.5.1 Email Notification

To measure the level and values for Gas and Dust sensors, this project was created. Notification by email as the means of viewing the information. The sensors are attached to the Gizduino Mini microcontroller and this microcontroller is linked to Raspberry Pi, which channels are on high level when sending an email message to the Barangay(Caya et al., 2018). Another design has been using email notification. The project is about the system of home security. In this project, Raspberry Pi 3 was used and another module, PIR Motion sensor and camera, was also introduced(Rani et al., 2018). Other than that, the project developed by Malche (2017) was used Arduino Nano microcontroller and the type of connection is Wi-Fi and Bluetooth. This project was developed for building smart home system by implemented Internet of Things (IoT).

2.5.2 SMS Notification

SMS notification nowadays can help people to communicate. SMS also can help in healthcare based on IoT. Gupta, Patchava, & Menezes (2016) developed the project was used microcontroller which is Raspberry Pi that connected to other sensors compared to other project which is used Arduino, Arduino Uno and Arduino Mega microcontroller (Shingala & Patel, 2017)(Singh et al., 2018)(Abagissa et al., 2018). However, project that develop by Singh et al. (2018) using Arduino Mega and NodeMCU ESP8266.

2.5.3 Telegram Messenger

Telegram notification are widely used nowadays. Telegram can help the user gain the notification. There are 3 research paper that used Telegram notification as medium to get the data. According to Anvekar et al. (n.d.), P.N.V.S.N, Rao, & Rao, (2017) and (Moretti et al., 2017), they are used Telegram notification to the their project. They also used Raspberry Pi microcontroller as the control board.

Table 2.2 Comparison of notification

Title	Author	Type of Notification	Type of Control Board
Air Pollution and Particulate Matter Detector Using Raspberry Pi with IoT Based Notification	(Caya et al., 2018)	Email	• Gizduino • Raspberry Pi 2
Automatic Home Appliances and Security of Smart Home with RFID, SMS, Email and Real Time Algorithm Based on IOT	(Shingala & Patel, 2017)	SMS	• Arduino

Design Alternatives for End User Communication in IoT Based System Model	(Anvekar et al., n.d.)	Telegram	• Raspberry Pi
Healthcare based on IoT using Raspberry Pi	(Gupta et al., 2016)	SMS	• Raspberry Pi
Home Automation using Telegram	(P.N.V.S.N et al., 2017)	Telegram	• Raspberry Pi
Energy-Efficient IoT-Enabled Fall Detection System with Messenger-Based Notification	(Moretti et al., 2017)	Telegram	• Raspberry Pi 3
IoT Based Home Security System Using Raspberry Pi with Email and Voice Alert	(Rani et al., 2018)	Email	• Raspberry Pi 3
Internet of Things (IoT) for building Smart Home System	(Malche, 2017)	Email	• Arduino Nano
IoT based Smart Home Automation System using Sensor Node	(Singh et al., 2018)	SMS	• NodeMCU ESP8266 • Arduino Uno

IoT Based Smart Agricultural Device Controlling System	(Abagissa et al., 2018)	SMS	• Arduino Mega

2.6 Development Techniques

In the IT arena, the Internet of Things is a new paradigm shift. The phrase "Internet of Things" is coined from the two words i.e. the first word is "Internet" and the second word is "Things" (Madakam et al., 2015). The Internet of Things does not have a common description available that is appropriate to the world user community. In fact, there are many different groups that have defined the term, including academics, researchers, practitioners, innovators, developers and business people. The best definition for the Internet of Things is "A robust and open network of intelligent objects capable of self-organizing, sharing information, data and resources, reacting and acting in the face of environmental situations and changes." With IoT, which is connectivity, there is a lot of benefit. IoT facilitates contact between devices, known as the Machine-to-Machine. The physical systems are therefore able to stay connected and therefore the total transparency is possible with lower inefficiencies and higher quality. In addition, due to the interaction and control of physical objects to wireless infrastructure electronically and centrally, there is a large amount of automation and control in the work. The machines can communicate with each other without human intervention, leading to faster and timely output. Finally, IoT is very efficient and time-saving. The interaction of the machine to machine offers improved efficiency, hence: accurate results can be achieved quickly. This leads to valuable time savings. This allows people to do other creative jobs instead of doing the same tasks every day.

2.6.1 Internet of Things (IoT)

In the IT arena, the Internet of Things is a new paradigm shift. The phrase "Internet of Things" is coined from the two words i.e. the first word is "Internet" and the second word is "Things" (Madakam et al., 2015). The Internet of Things does not have a common description available that is appropriate to the world user community. In fact, there are many different groups that have defined the term, including academics, researchers, practitioners, innovators, developers and business people. The best definition for the Internet of Things is "A robust and open network of intelligent objects capable of self-organizing, sharing information, data and resources, reacting and acting in the face of environmental situations and changes." With IoT, which is connectivity, there is a lot of benefit. IoT facilitates contact between devices, known as the Machine-to-Machine. The physical systems are therefore able to stay connected and therefore the total transparency is possible with lower inefficiencies and higher quality. In addition, due to the interaction and control of physical objects to wireless infrastructure electronically and centrally, there is a large amount of automation and control in the work. The machines can communicate with each other without human intervention, leading to faster and timely output. Finally, IoT is very efficient and time-saving. The interaction of the machine to machine offers improved efficiency, hence: accurate results can be achieved quickly. This leads to valuable time savings. This allows people to do other creative jobs instead of doing the same tasks every day.

2.6.2 Sensor

2.6.2.1 NodeMCU (ESP8266)

A microcontroller with a processor, memory and peripherals can be considered as a self-contained system and can be used as an embedded system. NodeMCU is an open-source firmware and development kit that helps you to create or prototype IoT products. It includes ESP8266, a low-cost, full TCP / IP Wi-Fi chip.

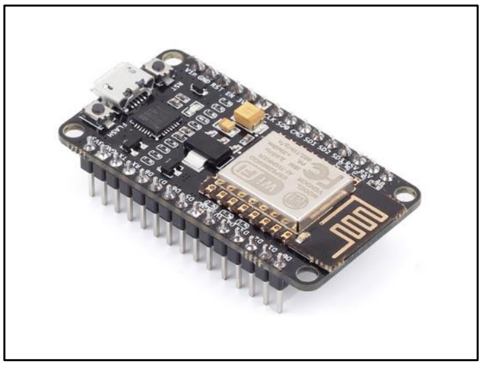


Figure 2.5 NodeMCU (ESP8266)

2.6.2.2 Temperature and Humidity Sensor

The sensors for temperature and humidity are used to provide information on temperature and humidity. These are the key sensors used to measure the system's nutrient conditions. IT is important for these sensors to be very accurate, as it is the basis for the plant's growth and safety management and maintenance

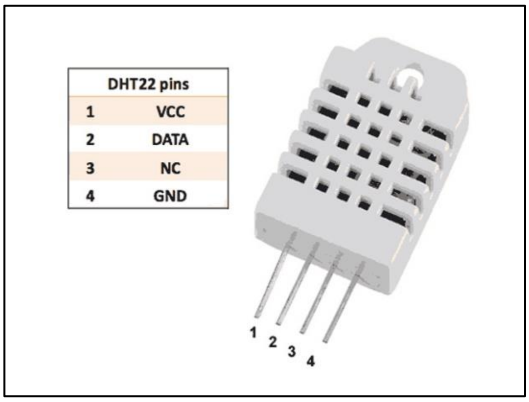


Figure 2.6 Temperature and Humidity Sensor (DHT22)

2.6.2.3 Water Level

The sensors of water level are used to detect the level of substances that may flow. The sensors notify the operator of any possible property damage resulting from any leakage and also allow to know when a container is approaching empty(Lakhiar et al., 2018).

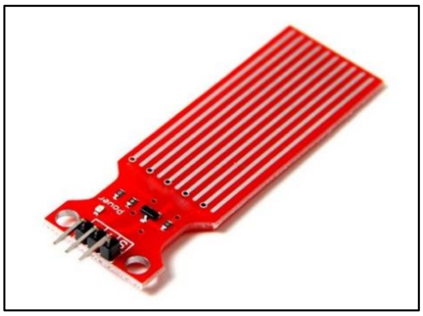


Figure 2.7 Water sensor level

2.6.2.4 Soil Moisture Sensor

The moisture sensor in the soil senses the soil's moisture content. It has two conductors, commonly known as electrodes, separated by a small distance.

Together with the digital pins, it has Vcc, ground and signal pins that provide the output as high or low. The signal pin gives the analog value proportional to the amount of soil moisture (Mithya, Aishwarya, Gayathri, Mahalakshmi, & Pavithra, 2019) .

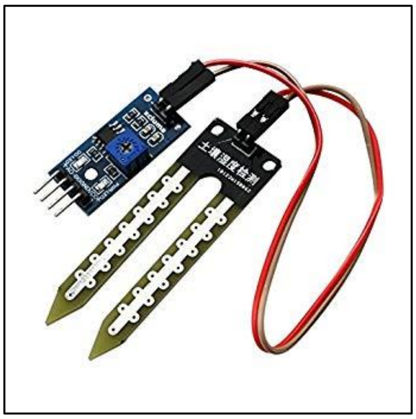


Figure 2.8 Soil Moisture Sensor

2.6.3 Thingspeak

Thingspeak is a web based open API IoT source information platform that comprehensive in storing the sensor data of varied 'IoT applications' and conspire the sensed data output in graphical form at the web level. Thingspeak communicate with the help of internet connection which acts as a 'data packet' carrier between the connected 'things' and the Thingspeak cloud retrieve, save/store, analyse, observe and work on the sensed data from the connected sensor to the host microcontroller. The most primary feature of Thingspeak functionality is the term 'Channel' that have field for data, field for location, field for status for varied sensed data. Once channels are created in the 'Thingspeak' the data can be implemented and alternately one can process and visualize the information (Pasha, 2016).

There are several strengths in using the Thingspeak which is the key thing which separates ThingSpeak from any of the mentioned competitors is that it creates a sense of community through the possibility of creating public channels. The API allows for very easy visualization of collected data through using spline charts. Therefore, it is visually appealing and is much easier when examining collected data compared to other open source APIs.

Besides, Thingspeak also has the weakness which is uploading data to the API there is a limit up one update per channel every fifteen seconds. We postulate that the reason for this limit in uploads is due to the excess bandwidth that could be used, and therefore would end up costing ThingSpeak additional funds for a non-profit service (Gómez Maureira et al., 2014).

2.6.4 Black box Testing

Black box testing is based on the requirements specifications and there is no need to examining the code in black box testing. This is purely done based on customers view point only tester knows the set of inputs and predictable outputs. Black box testing plays a significant role in software testing, it aids in overall functionality validation of the system. Black box testing is done based on end user perspective. Black box testing is done from beginning of the software project life cycle. All the testing team members need to be involved from beginning of the project. The main advantage of black box testing is that, testers no need to have knowledge on specific programming language, not only programming language but also knowledge on implementation. Another advantage of black box testing is that it helps to expose any ambiguities or inconsistencies in the requirements specifications (Nidhra & Dondeti, 2012).

2.7 Summary

The Literature Review chapter dealt with previous research on this project's topic. We can see that many works have been done to improve IoT in hydroponics by doing some work. In the hydroponic system, various techniques have been applied. There is also a notification that the data can be sent to the user.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter is about the methodology proposed to develop the IoT Based Hydroponic Wick System. Methodology describe the ways and methods applied in completing the project, according to the objectives of this project. Generally, methods and techniques were elaborated in specification, design, programming and testing.

3.2 Methodology

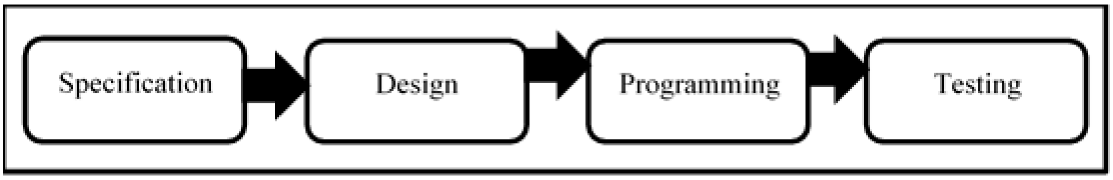


Figure 3.1 Methodology general phases

Figure 3.1 showed the four phases that involved in completing this project. The first phase was specification related to the project. In the specification, a background of study has been conducted by identify the factor and variables to be used in completing the project. In this project, background of study was conducted on hydroponic wick system that will implement the IoT. The second phase is Design. This phase will proceed after the first phase was done. This technique describes the how to design the prototype of project. The next phase is programming which is develop the hardware and software component. This phase also describes the technique and tools that will be used to integrate the hardware and software component. Testing is the last phase in developing this project. In this phase, it will be describing the tasks that will be perform in testing phase.

3.3 Phase 1: Specification

Specification is the first phase in Methodology general phases. Figure 3.2 shows the important processes involved in this phase. This phase consists of conduct theoretical study and literature research in develop IoT Based Hydroponic Wick System.

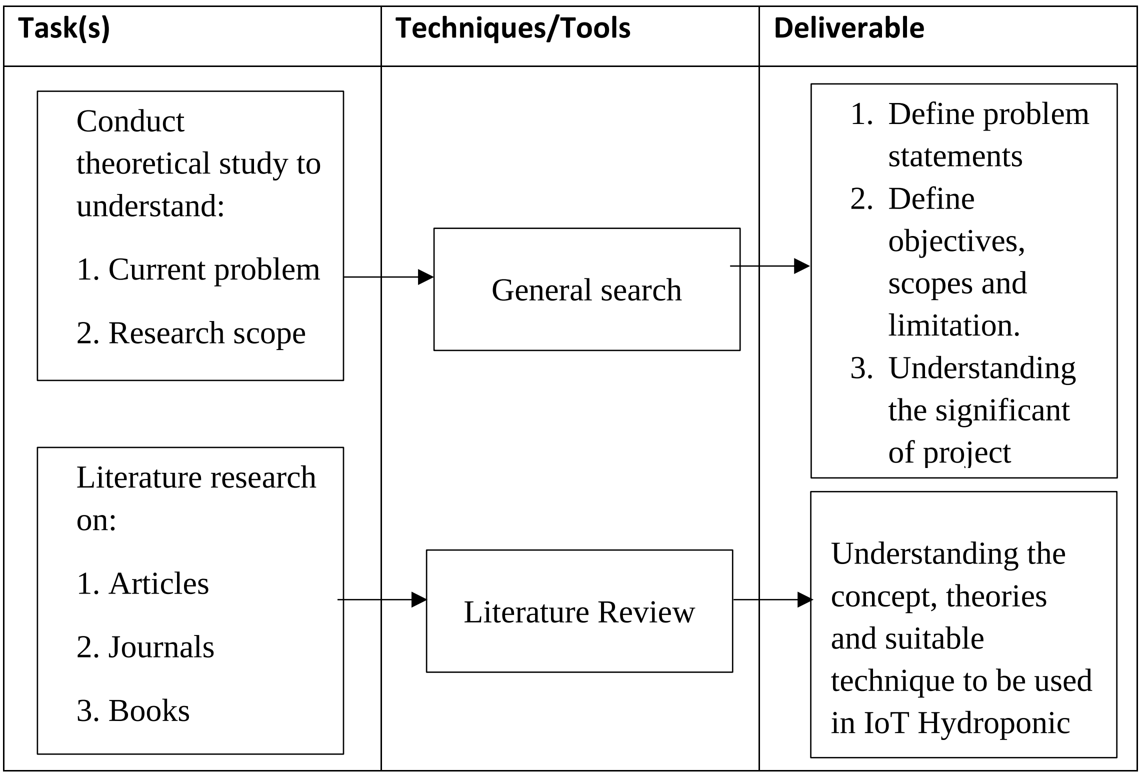


Figure 3.2 The details of specification phase.

3.3.1 Theoretical study

The first task of Phase 1 is the theoretical study. This purpose of the first task is to understand the current problem and research scope for develop the project. In order to achieve this phase is make some general research. By doing some general search, problem statements, objectives, scopes and limitation can be determined. Besides, it will be more understanding about the significant of project

3.3.2 Literature Research

The second task if phase 2 is conduct literature research. After the first task is done and all the expected function clearly defined, literature research is needed to be achieved. This can be done by doing the literature review on articles, journals and books. By doing the related works in the literature review, this project will be more understanding the concept, theories and suitable technique to be used in developing this project.

3.4 Phase 2: Design

The next phase of methodology is design. In this phase consists of designing the flow of the system, designing sensor node and designing the circuit diagram. It will help the developer to develop the project easily.

Task(s)	Techniques/Tools	Deliverable

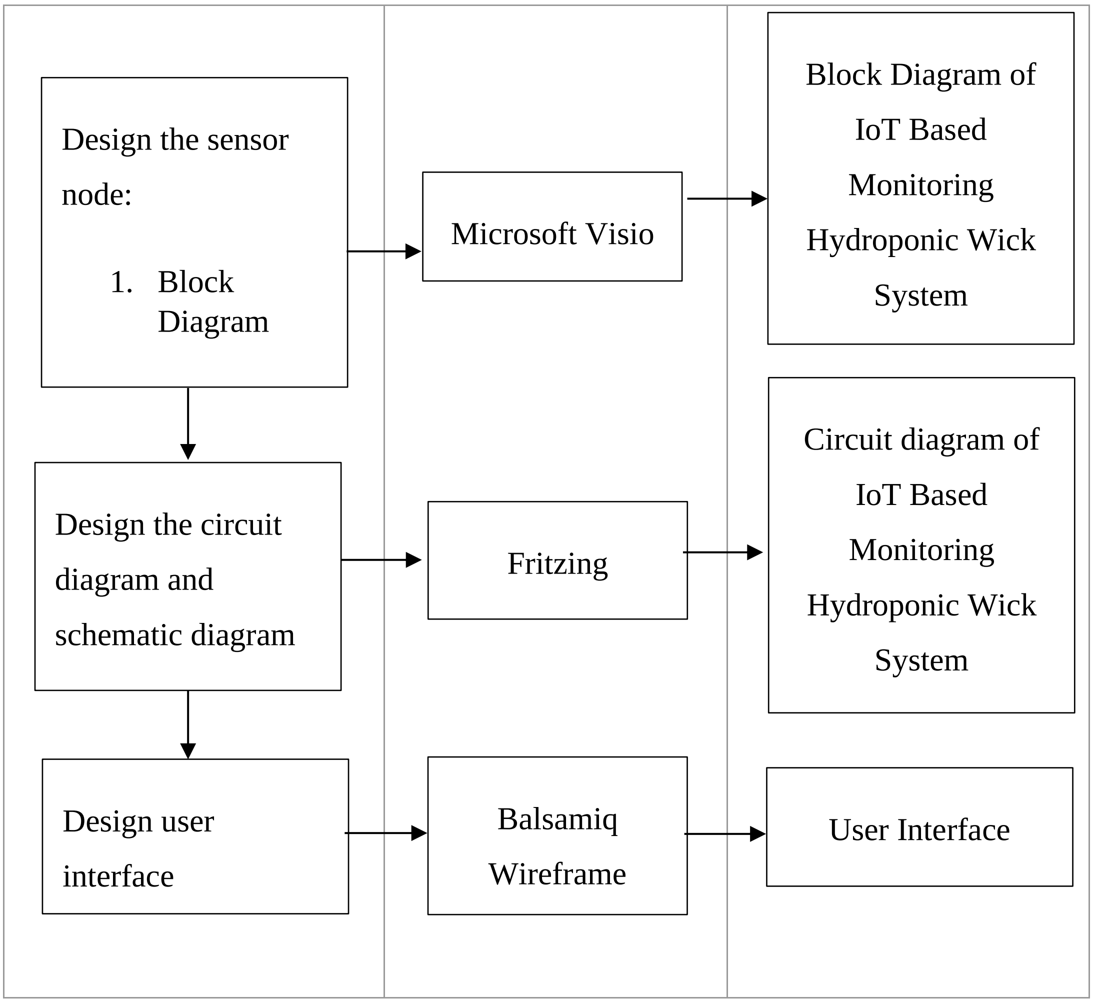


Figure 3.3 The details of design phases

3.4.1 Design Sensor Node

The second task for the Phase 2 is designing the sensor node which is block diagram. This second task can be also done using Microsoft Visio. Designing Block Diagram is the basic element of IoT architecture. This phase show how they are connected to collect, store and process the data.

3.4.2 Design Circuit Diagram and Schematic Diagram

After the second task is done, the next task will be proceeded which is design the circuit diagram and schematic diagram. The second part of the task deals with mainly the producing the schematic circuit diagram and this is done by using Fritzing. Fritzing will be used to sketch and view the schematic diagram of the hardware component prototype.

3.4.3 Design user interface

The last task will be proceeded for this phase is design user interface. Design user interface is needed to produce the prototype by using Balsamiq Wireframe. By design user interface which drawn with basic shape and lines. The main focus for design the user interface is to figuring out layout, content placement, solving navigation and functionality problems in a format that is easy to adjust.

3.5 Phase 3: Programming

Last but not least, programming is the phase before proceed to the last phase. This process describes about techniques and tools that required in develop hardware and software and also describe about integrating between hardware and software component. The Figure 3.4 shows the details of programming phases.

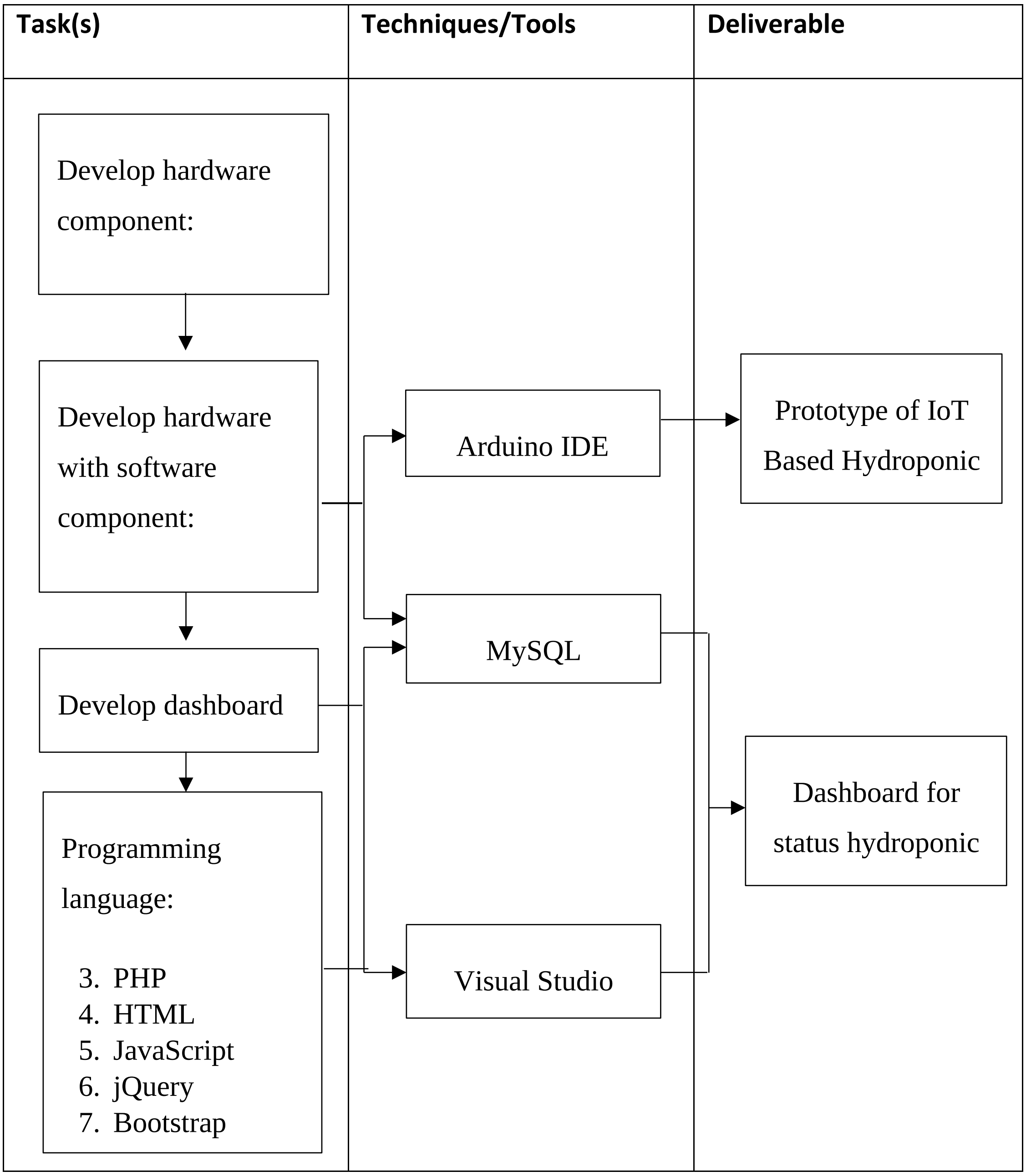


Figure 3.4 Detail programming phase

3.5.1 Develop Hardware

The hardware that required for this project are NodeMCU (ESP266) and variety of the sensor. The table show the description of hardware that required for developing the project.

Table 3.1 Description of hardware specification

Hardware	**Description**
NodeMCU (ESP8266)	NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.
Water Level Sensor	Water Level Sensor are used to detect the level of water that may flow.
DHT22	To measure and provide about temperature and humidity.
Soil Moisture Sensor	The soil moisture sensor is to senses the soil moisture content.

3.5.2 Develop Hardware with Software Component

The second task for Phase 3 is develop hardware with software component. After developing the hardware component is done, develop hardware with software component will be proceed in order to achieve the desired function. The software component is hardcoding and cloud based. Hardcoding can be done by Arduino IDE. Arduino IDE is commonly use IDEs for building IoT projects because of Arduino IDE support many controller boards. Lastly, this project also required database to store the data that transmitted from the sensor. Database that will be used in this project is MySQL database.

3.5.3 Develop dashboard status monitoring

The last task for Phase 3 is develop dashboard status monitoring. This task can be done by using the Visual Studio Code and MySQL database. In this task, the programming language thar required are PHP, HTML, JavaScript, jQuery and Bootstrap. Visual Studio Code is the software to develop the dashboard. Moreover, Visual Studio Code support many programming languages and it is including the programming language that are need to develop the dashboard. For previous task, MySQL database are for store data from the sensor. However, in developing the dashboard for status monitoring, data that stored in MySQL will be display on the dashboard.

3.6 Phase 4: Testing

The final phase of methodology is Phase 4 whish is testing phase. In this phase involve two task which is sensor testing and connectivity between sensor and cloud based. For testing the project, Black box Testing method are used in evaluate the respond time and delivering a fully functional and usefulness of developing the project.

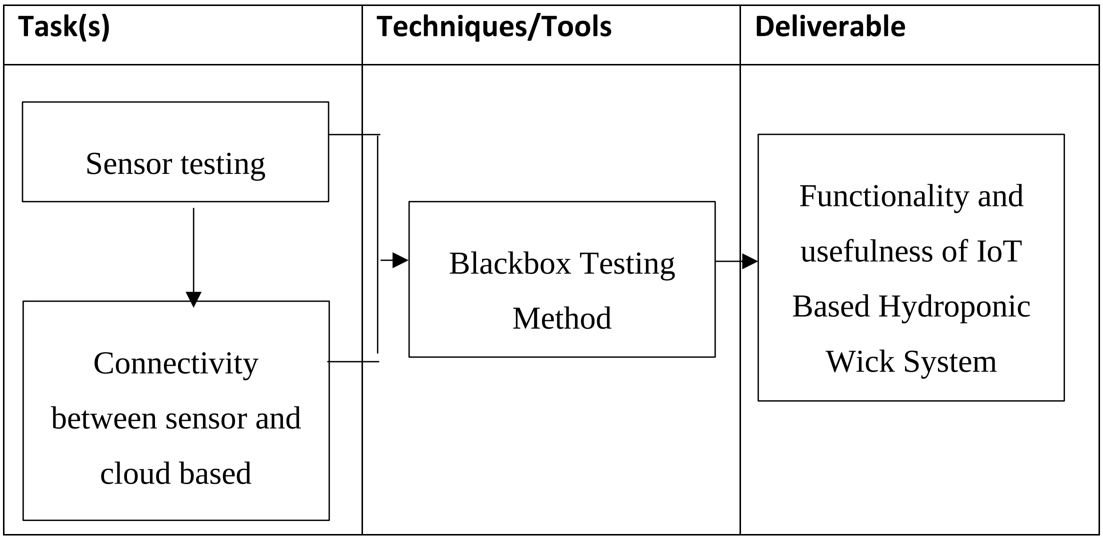


Figure 3.5 The details of testing phase

3.6.1 Sensor Testing

The first task of Phase 4 is the sensor testing. After developing the hardware component, all the sensor component will be tested in order to know whether the sensor is function or not. Four different sensors will be tested which is temperature and humidity sensor, water level sensor, light intensity sensor and soil moisture sensor. The test will be conducted where the researches will provide the stimuli to the sensor.

3.6.2 Connectivity between Sensor and Cloud Based

The second task which is connectivity between the sensor and cloud based. The testing will be conducted after the sensor testing is perfectly working. This task will determine whether the connection between the hardware component and software components are correct. The researchers will check individual connection between sensor, database and monitoring dashboard which is shows the data in statistic.

3.7 Summary

Chapter 3 basically described the general what were the steps taken in completing prototype. The general phases have four phases which is specification, design, programming and testing. The significance of methodology is to plant the research from the beginning of project until the scope and objective are achieved. This methodology also gives guidance to the researches to complete their project as planned.

CHAPTER 4

CONSTRUCTION

4.1 Introduction

This chapter will provide further discussion about project design and development process. The project design and development that implemented from the beginning to the final process until the application is ready to use will be shown in stages.

4.2 Hardware Requirements

This subtopic will discuss the requirement to build the hardware part of prototype. There are four main requirement which is NodeMCU ESP8266, DHT22, water level sensor and soil moisture sensor.

4.2.1 NodeMCU ESP8266

A microcontroller can be considered a self-contained system with a processor, memory and peripherals and can be used as an embedded system. NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT product. This is included ESP8266 is a low-cost Wi-Fi chip with full TCP/IP.

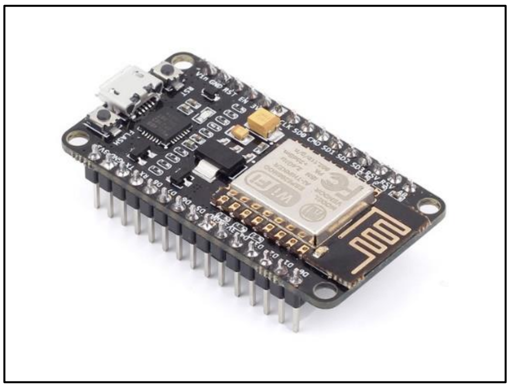


Figure 4.1 NodeMCU ESP8266

4.2.2 Temperature and Humidity (DHT22)

DHT22 is the sensor that senses the temperature and humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air. The temperature range for DHT22 sensor are between -40 °C to 80 °C. Besides, the range for humidity are between 0% to 100%. The accuracy for DHT22 sensor is ±0.5°C for temperature and ±1% for humidity. This sensor will be able to display the more accuracy output for this project. Other than that, the user may have the proper temperature and humidity for successful hydroponic plant.

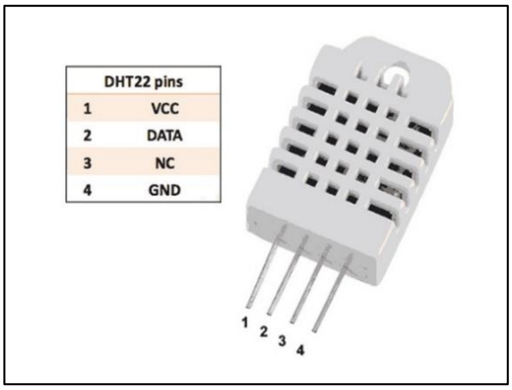


Figure 4.2 DHT22 sensor

4.2.3 Water Level Sensor

This project also required water level sensor to measure the level of water in hydroponic container. This sensor can measure the water level until 40mm. This sensor will help user to notify the level of water.

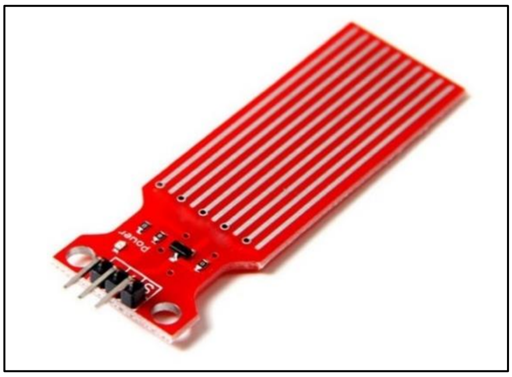


Figure 4.3 Water Level Sensor

4.2.4 Soil Moisture Sensor

The soil moisture sensor senses the moisture content of the soil. Three soil moisture will be use in this project because there are three mini hydroponic pot. This sensor will determine the whether the plant are needed to watering or does no need watering.

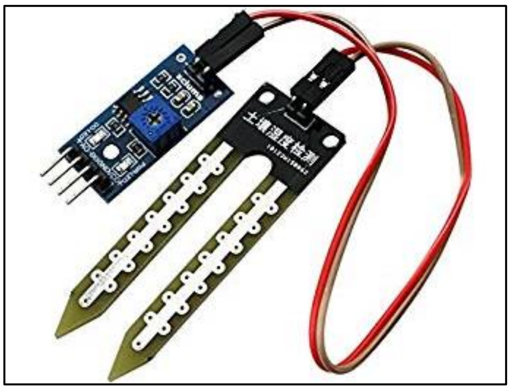


Figure 4.4 Soil Moisture Sensor

4.3 Software Requirements

This sub-chapter will discuss the software requirements in developing IoT Based Monitoring Hydroponic Wick System. There are three software were used to developing this project which is Arduino IDE, Fritzing and Visual Studio Code.

4.3.1 Arduino IDE

Arduino IDE or Arduino Integrated Development Environment is the basic tools to written the program and test scripts. For this project, Arduino IDE is chosen to complete this prototype. Arduino IDE are supported many boards and NodeMCU is included. The sketch that had been written will be upload into NodeMCU through USB cable that connected to NodeMCU. Figure 4.5 shows an example of interface Arduino IDE.

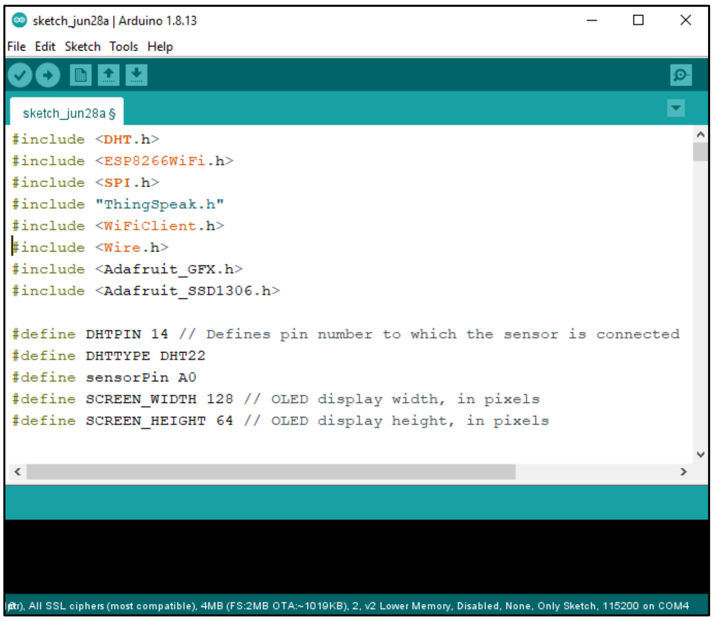


Figure 4.5 Interface of Arduino Ide

4.3.2 Fritzing

Fritzing is the open-source software that help designer to create a prototype. The designer can design the circuit on breadboards. In Fritzing, there are many popular components such as microcontroller board and core part. Besides, designer can also import the component that are not provide in Fritzing. Other than that, Fritzing help the designer to convert circuit into schematic diagram. However, schematic will not be neat and requires some rearranging the component to form the connection. Last stage of creating a circuit in Fritzing is PCB layout. This stage also required to rearrange the components. Lastly, Fritzing also offer to run the code which can determine whether the code is work or not on the circuit. Figure 4.6 shows the interface of Fritzing.

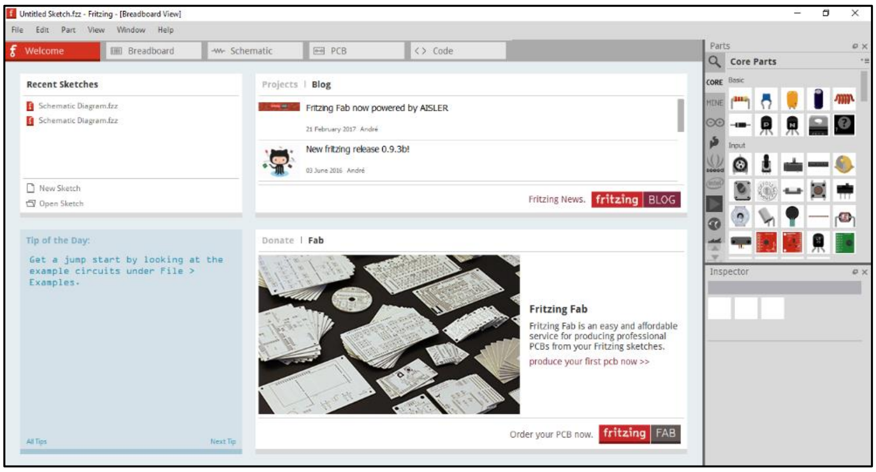


Figure 4.6 Interface of Fritzing

4.3.3 Visual Studio Code

Visual Studio Code is the open-source software to design the dashboard for this project. Visual Studio Code have support many of programming language which is include PHP, HTML, JavaScript and more. The language that are needed in developing this project are supported by Visual Studio Code. In addition, Visual Studio Code supports Windows, MacOS and Linux environments. This is shows that it is flexible and can be edited on it, regardless of what the operating system of the device is. Figure 4.7 shows the interface on Visual Studio Code.

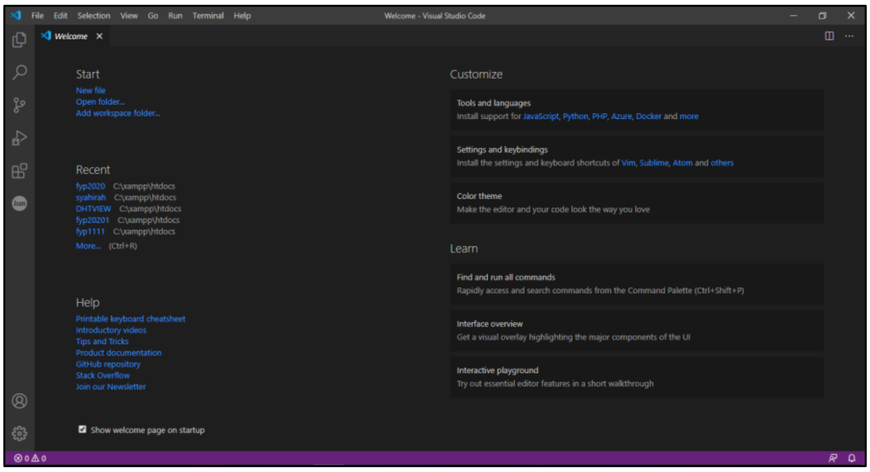


Figure 4.7 Interface of Visual Studio Code

4.4 Network Setup

Developer need to set up the web server before develop the system. A web server is a computer on which a web site is hosted and a program that runs on a such computer. Web server that will be use for this project is XAMPP.

4.4.1 Web Server

XAMPP is chosen as the web server. XAMPP is open source software and it is basically a local server. This local server work on the own laptop or desktop. XAMPP helps developer to develop the local web. Besides, XAMMP are easily connected to MySQL database. Figure 4.8 shows the control panel of XAMPP.

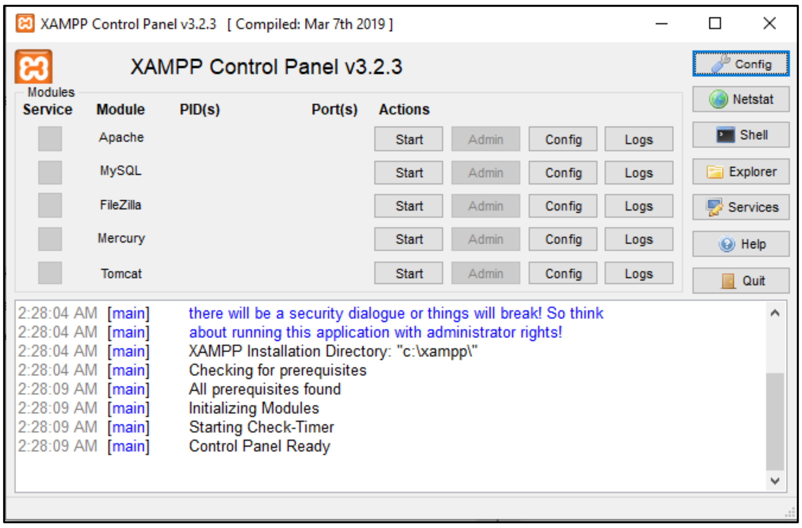


Figure 4.8 Interface Xampp

4.5 User Interface Design

User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in a website, or technology product. User interface designs must not only be attractive to potential users, but must also be functional and created with users in mind. For this project, user may monitor their condition of plant by through the dashboard. Interface design are needed to give an idea rather than develop by using actual application.

Figure 4.9 shows the interface of the main page. The main page of this interface called as Dashboard. User can monitor the current status of temperature, humidity, water level and soil moisture of each pot in the dashboard. Besides, user also can monitor the status based on graph that display on dashboard.

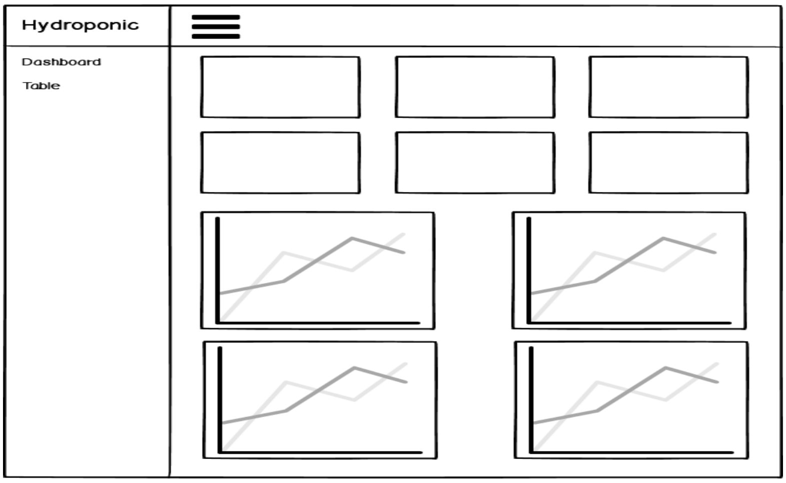


Figure 4.9 Interface Dashboard

Next interface design is for table design. This interface will ease user to monitor the past data of their hydroponic plant. Furthermore, user may search the time or the data of the sensor. Figure 4.10 shows the interface for table design.

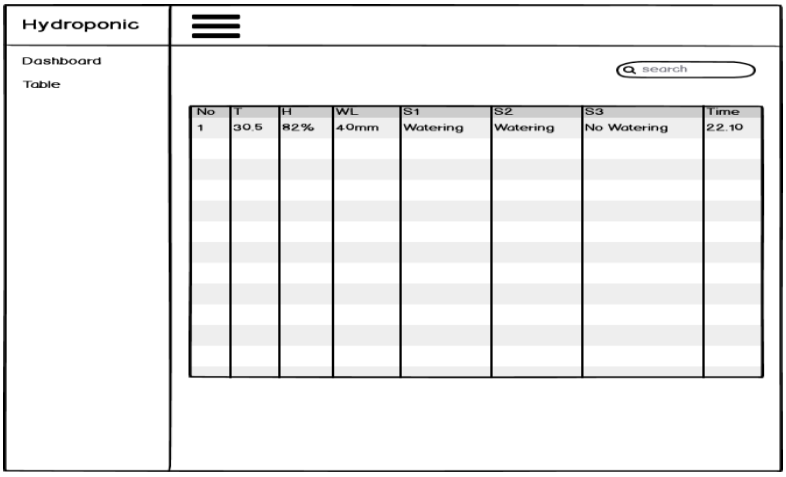


Figure 4.10 Interface Data Table

4.6 Database Design

A database is an organized collection of structured information, or data, usually stored in a computer system. A database is a systematic collection of data that can easily to access and manage. Database design is the process of producing a detailed data model of database. Table 4.1 shows the list the attribute in the value table.

Table 4.1 Database Attribute

Field Name	Data Type	Description
id(PK)	bigint(20)	Id number
temperature	float	Temperature data
humidity	float	Humidity data
waterlevel	float	Water level data
soilmoisture1	float	Soil moisture for pot 1
soilmoisture2	float	Soil moisture for pot 2
soilmoisture3	float	Soil moisture for pot 3
date	timestamp	Time data save to database

4.6.1 Block Diagram

A block is specialized, high level flowchart used in engineering. It is used to design new systems or to describe and improved existing ones. For IoT Based Monitoring Wick System, block diagram is needed to shows the overview of the major component system and important working relationship. Figure 4.11 shows the block diagram of IoT Based Monitoring Wick Hydroponic System.

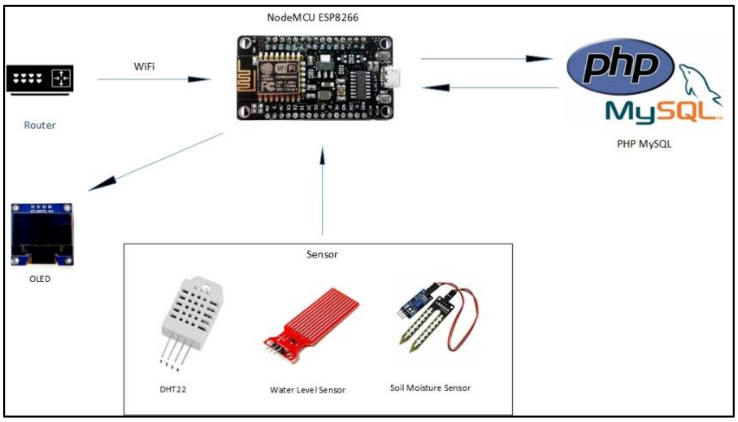


Figure 4.11 Block Diagram of IoT Based Monitoring Wick Hydroponic System

4.6.2 Fritzing Diagram

Fritzing diagram is created to show how the hardware component of the prototype will be connected. This must be done as visual guide for assemble the component later. Figure 4.12 show the fritzing diagram for this project.

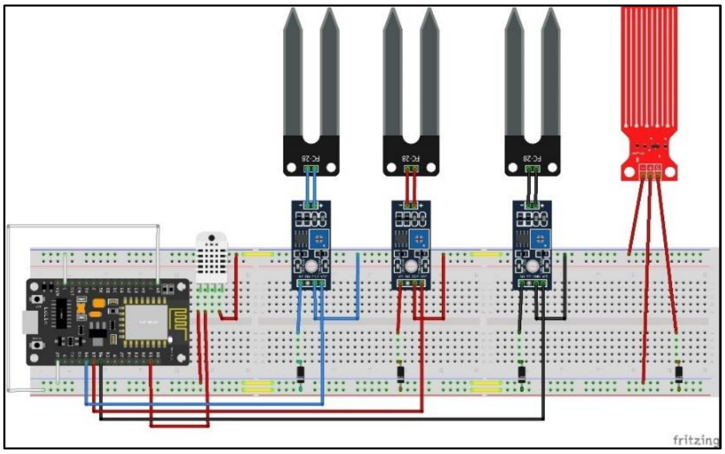


Figure 4.12 Fritzing Diagram

4.6.3 Schematic Diagram

Figure 4.13 showed the schematic diagram for IoT Based Monitoring Hydroponic Wick System. The schematic diagram is helping in understanding on how to linked the sensor and electronic function without detailing the software and hardware used.

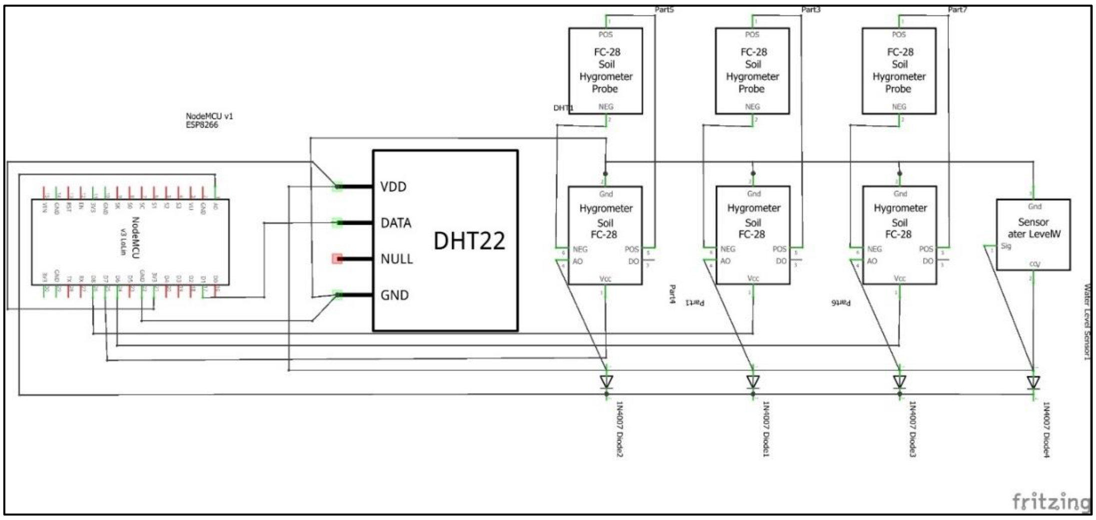


Figure 4.13 Schematic Diagram

4.6.4 Prototype Assembling

After the fritzing and schematic diagram have be define, this project will assemble the prototype hardware component.

Firstly, DHT22 which is temperature and humidity sensor are assemble to NodeMCU ESP8266. DHT22 is digital sensor that have three pins. The '+' pin on DHT22 was assemble to 3V3 on NodeMCU ESP8266. Besides, the 'out' and '- 'pins are pinned to D1 and GND on NodeMCU ESP8266 respectively. Figure 4.14 shows the sensor pinned to the NodeMCU ESP8266.

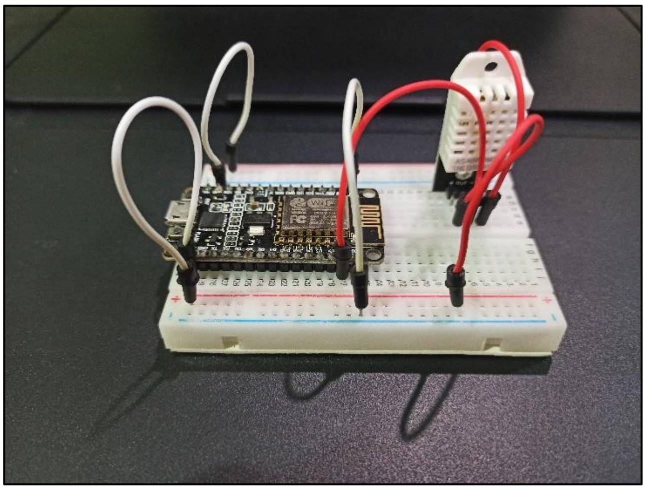


Figure 4.14 Assemble DHT22 sensor

Secondly, soil moisture sensor is assembling to NodeMCU ESP8266. For this project required three soil moisture that will read the data from three pots. Soil moisture sensor need to pinned to the analog pin. Unfortunately, NodeMCU ESP8266 only has one analog pin. If, all the three sensors are pinned directly to analog pin (A0), the output of data will be the same. So, diode 1N4007 are required to make sure the data will be read correctly. VCC pin of each sensor is pinned to D6, D7, and D8 respectively. Besides, the GND pin on soil moisture sensor pinned to GND on NodeMCU ESP8266. The Figure 4.15 shows the three soil moistures assemble to NodeMCU ESP8266 with DHT22 sensor.

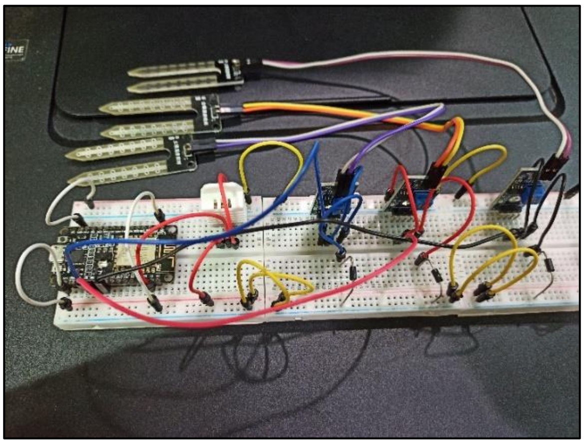


Figure 4.15 Assemble Soil Moisture Sensor

Figure 4.16 shows the last sensor assemble to this project is water level sensor. This sensor is to measure the level of water in this project. This sensor also required analog pin which is A0. So, diode 1N4007 is used to make sure the data will be read one by one of the sensors.

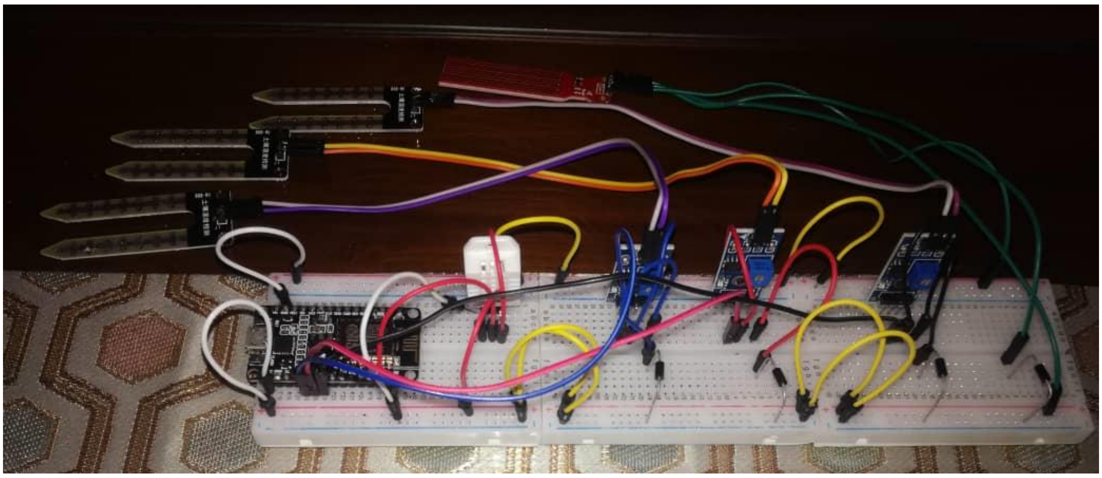


Figure 4.16 Assemble Water Level Sensor

4.7 Prototype Development

In this development process, the software Arduino IDE was used. The sketches are written in Arduino IDE and uploaded to NodeMCU ESP8266 through USB cable. There are 3 main part which are setup, loop and Sending_to_database.

Before proceed to setup part, libraries and variable must be defined first. The libraries that are involve in this script are ‘DHT.h’, ‘ESP8266WiFi.h’, ‘SPI.h’ and ‘WiFiClient.h’. These libraries can be seen through the #include syntax in the program. ‘DHT.h’ is the library for sensor DHT22 while the other libraries are for NodeMCU ESP8266 which can connect through WiFi. Besides, there are the #define syntax on this program for DHT22 sensor. #define is useful C++ component that allow to give a name to constant value before the program is compiled. Figure 4.17 shows the library definition part.

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <SPI.h>
#include <WiFiClient.h>

#define  DHTPIN D1 // Defines pin number to which the sensor is connected
#define  DHTTYPE DHT22
```

Figure 4.17 Library definition

The next step before the setup method proceed is defined the connection for the prototype. NodeMCU ESP8266 is built with WiFi module. So, the connection of the protype must be define. Figure 4.18 shows the connection definition part which required WiFi’s SSID and password of SSID. The Figure 4.18 shows the connection definition part in Arduino IDE.

```
const char *ssid =  "Honor 5C";      // replace with your
const char *pass =  "0139894984";
```

Figure 4.18 Connection Definition

The last step is variable declaration. This part is which each variable used will be pin-mapped to the pin where the components that representing are connected. The Figure 4.19 shows the all the variable declaration on Arduino IDE.

```
DHT dht(DHTPIN, DHTTYPE);
const int sensor_pin = A0;
int readData, Value_D6, Value_D7, Value_D8, Value_D5;
int Pin_D6 = 12, Pin_D7 = 13, Pin_D8 = 15, Pin_D5 = 14;
float humidity, temperature;
```

Figure 4.19 Variable Declaration

After all libraries and variable are defined, this development proceeds to the main part which is void setup(). This part can be seen on Figure 4.20 which is the pinMode is ‘OUTPUT’ or ‘INPUT’. Besides, Figure 4.20 also shows script for connecting to WiFi.

```
void setup() {

  Serial.begin(9600);
  delay(100);
  dht.begin();
  pinMode(Pin_D6,OUTPUT);
  pinMode(Pin_D7,OUTPUT);
  pinMode(Pin_D8,OUTPUT);
  pinMode(Pin_D5,OUTPUT);
  pinMode(A0,INPUT);

  Serial.println("Connecting to ");
       Serial.println(ssid);


       WiFi.begin(ssid, pass);

       while (WiFi.status() != WL_CONNECTED)
       {
            Serial.print("wifi could not connect");
            delay(500);
       }

      Serial.println("");
      Serial.println("WiFi connected");
      Serial.println(WiFi.localIP());
}
```

Figure 4.20 Script for connecting to Wi-Fi

The next part after setup method done is loop method. The loop method written where all the main functions of the prototype will be defined. The code on loop method will be run until the NodeMCU is switched off. Figure 4.21 shows the code for loop method on Arduino IDE.

```
void loop() {
  readData = dht.read(DHTTYPE); // Reads the data from the sensor
  humidity = dht.readHumidity();
  temperature = dht.readTemperature();

  digitalWrite(Pin_D6, HIGH);     //Turn D6 On
  delay(100);                     //Wait for sensor
  Value_D6 = analogRead(0);       //Read Analog pin as D8
  digitalWrite(Pin_D6, LOW);      //Turn D6 Off

  digitalWrite(Pin_D7, HIGH);     //Turn D7 On
  delay(100);                     //Wait for sensor
  Value_D7 = analogRead(0);       //Read Analog pin as D7
  digitalWrite(Pin_D7, LOW);      //Turn D7 Off

  //Repeat for D8
  digitalWrite(Pin_D8, HIGH);     //Turn D8 On
  delay(100);                     //Wait for sensor
  Value_D8 = analogRead(0);       //Read Analog pin as D8
  digitalWrite(Pin_D8, LOW);      //Turn D8 Off

  digitalWrite(Pin_D5, HIGH);     //Turn D8 On
  delay(100);                     //Wait for sensor
  Value_D5 = analogRead(0);       //Read Analog pin as D8
  digitalWrite(Pin_D5, LOW);      //Turn D8 Off
  delay(100);                     //Wait for sensor
  Sending_to_database();
  delay(100);
}
```

Figure 4.21 Loop Method

The last part is void Sending_to_database(). This part is to display the data. In Figure 4.22 shows the part of 'Serial.print' and 'client.print'. The different between this part is 'Serial.print' is the script for display on the serial monitor while 'client.print' is to insert data into database. This project used MySQL database to store the data.

```
void Sending_to_database()
{
  if (client.connect(server, 80)) {
    Serial.println("connected");
    // Make a HTTP request:
    Serial.print("Temperature :");
    Serial.print(temperature);
    Serial.println(" C ");
    client.print("GET /DHTVIEW/dhtvalue.php?temperature=");     //YOUR URL
    client.print(temperature);
    Serial.print("Humidity :");
    Serial.print(humidity);
    Serial.println(" % ");
    client.print("&humidity=");
    client.print(humidity);
    Serial.print("Water Level = ");
    Serial.println(Value_D5);
    client.print("&waterlevel=");
    client.print(humidity);
    Serial.print("Soil Moisture (Pot 1) = ");
    Serial.println(Value_D6);
    client.print("&soilmoisture1=");
    client.print(humidity);
    Serial.print("Soil Moisture (Pot 2) = ");
    Serial.println(Value_D7);
    client.print("&soilmoisture2=");
    client.print(humidity);
    Serial.print("Soil Moisture (Pot 3) = ");
    Serial.println(Value_D8);
    client.print("&soilmoisture3=");
    client.print(humidity);
```

Figure 4.22 Void Sending_to_database

Besides, in the Sending_to_database() part also have the code for developer to know the data from the sensor are store or not into the database. This can be known by the connection to the server. If there are no connection to the server, so the data are not be stored to database. Besides, the data from the sensor also does not display on serial monitor. The Figure 4.23 shows the code segment of connection server.

```
  client.print(" ");        //SPACE BEFORE HTTP/1.1
  client.print("HTTP/1.1");
  client.println();
  client.println("Host: 192.168.43.236");
  client.println("Connection: close");
  client.println();

  }
 else {
  // if you didn't get a connection to the server:
  Serial.println("connection failed");
}
```

Figure 4.23 Connection Server

In Arduino IDE, the script has been written to store the data to database. However, there are also need the php code to execute on the web server. Figure 4.24 shows php code for insert data into database MySQL.

```
<?php
class value{
 public $link='';
 function __construct($soilmoisture3,$soilmoisture2,$soilmoisture1, $waterlevel, $humidity, $temperature){
  $this->connect();
  $this->storeInDB($soilmoisture3,$soilmoisture2,$soilmoisture1, $waterlevel, $humidity, $temperature);
 }

 function connect(){
  $this->link = mysqli_connect('localhost','root','') or die('Cannot connect to the DB');
  mysqli_select_db($this->link,'hydroponic') or die('Cannot select the DB');
 }

 function storeInDB($soilmoisture3,$soilmoisture2,$soilmoisture1, $waterlevel, $humidity, $temperature){
  $query = "insert into value set temperature='".$temperature."', humidity='".$humidity."',
  waterlevel='".$waterlevel."', soilmoisture1='".$soilmoisture1."', soilmoisture2='".$soilmoisture2."',
  soilmoisture3='".$soilmoisture3."'";
  $result = mysqli_query($this->link,$query) or die('Errant query:  '.$query);
 }

}

if (isset($_GET ["soilmoisture3"])) if ($_GET ["soilmoisture3"] != "")
{
if (isset($_GET ["soilmoisture2"])) if ($_GET ["soilmoisture2"] != "")
{
if (isset($_GET ["soilmoisture1"])) if ($_GET ["soilmoisture1"] != "")
{
if (isset($_GET ["waterlevel"])) if ($_GET ["waterlevel"] != "")
{
if (isset($_GET ["humidity"])) if ($_GET ["humidity"] != "")
{
if (isset($_GET ["temperature"])) if ($_GET ["temperature"] != "")

$value=new value($_GET['soilmoisture3'],$_GET['soilmoisture2'],$_GET['soilmoisture1'],
$_GET['waterlevel'],$_GET['humidity'],$_GET['temperature']);
}}}}}
?>
```

Figure 4.24 Script for database

CHAPTER 5

RESULT AND FINDINGS

5.1 Introduction

In this chapter, results and findings of this project will be discussed in detail. This will be done based on analysis and statistic of conducted evaluation. This chapter will be discussed about the functionality evaluation which is fulfills the requirements of objective three (3): To evaluate the functionality of IoT Monitoring Hydroponic Wick System using Black Box Testing Method.

However, objective one (1) and objective two (2) requirements had been fulfilled before proceeding with testing phase. Figure 5.1 until Figure 5.3 proved that objective (1): To design a web-based interface for status monitoring of hydroponic wick system has been fulfill up to the requirements. Moreover, Figure 5.3 shows the prototype of IoT monitoring system for hydroponics in small space which fulfills the requirements for objective (2).

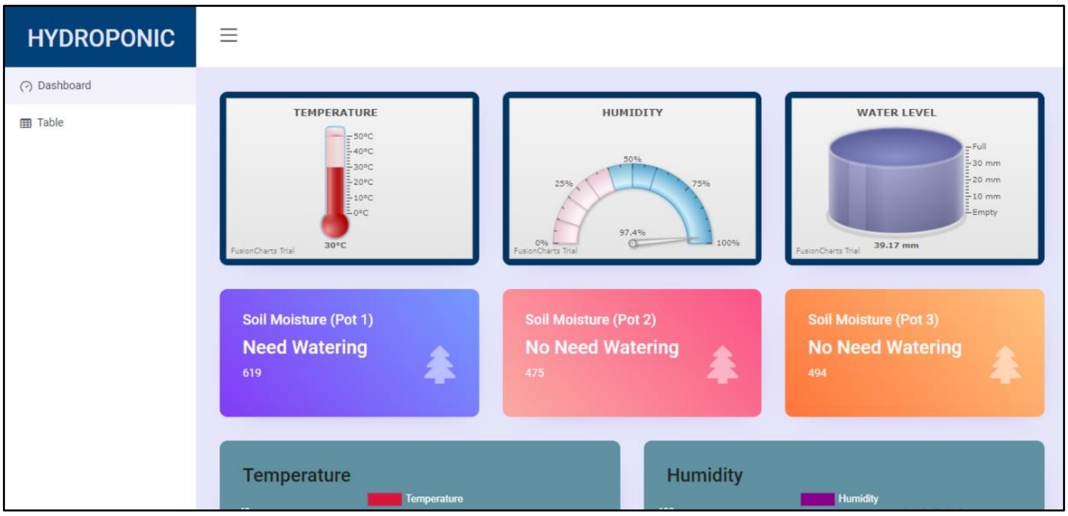


Figure 5.1 Dashboard IoT Based Monitoring Hydroponic Wick System

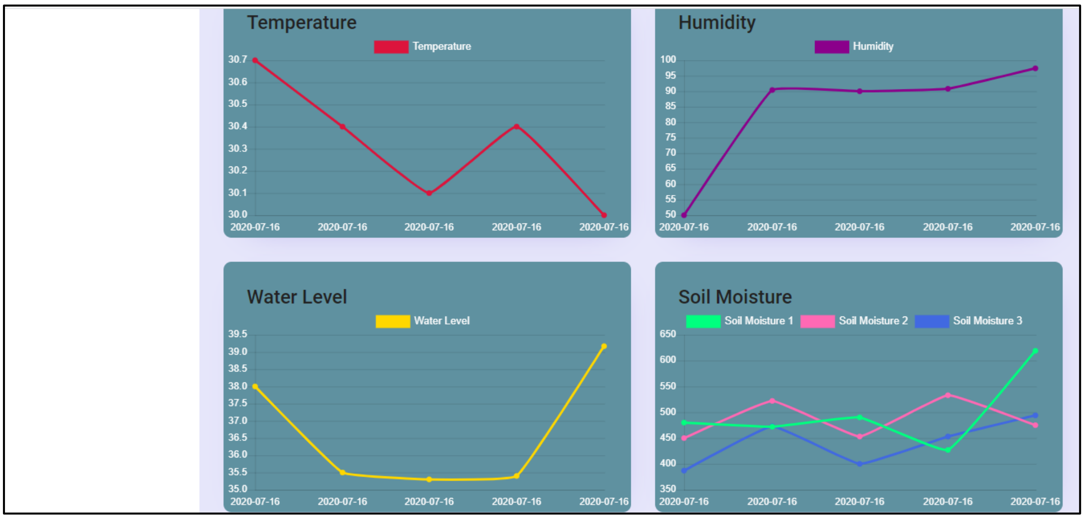
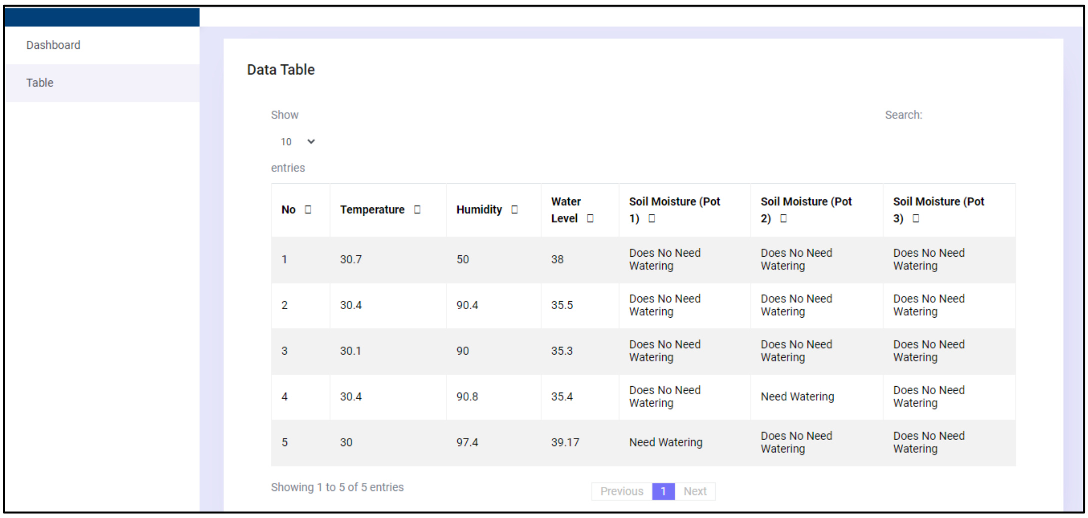


Figure 5.2 Dashboard IoT Based Monitoring Hydroponic Wick System



Dashboard

Table

Data Table

Show

10

entries

Search:

No	Temperature	Humidity	Water Level	Soil Moisture (Pot 1)	Soil Moisture (Pot 2)	Soil Moisture (Pot 3)
1	30.7	50	38	Does No Need Watering	Does No Need Watering	Does No Need Watering
2	30.4	90.4	35.5	Does No Need Watering	Does No Need Watering	Does No Need Watering
3	30.1	90	35.3	Does No Need Watering	Does No Need Watering	Does No Need Watering
4	30.4	90.8	35.4	Does No Need Watering	Need Watering	Does No Need Watering
5	30	97.4	39.17	Need Watering	Does No Need Watering	Does No Need Watering

Showing 1 to 5 of 5 entries

Previous 1 Next

Figure 5.3 Table data IoT Based Monitoring Hydroponic Wick System

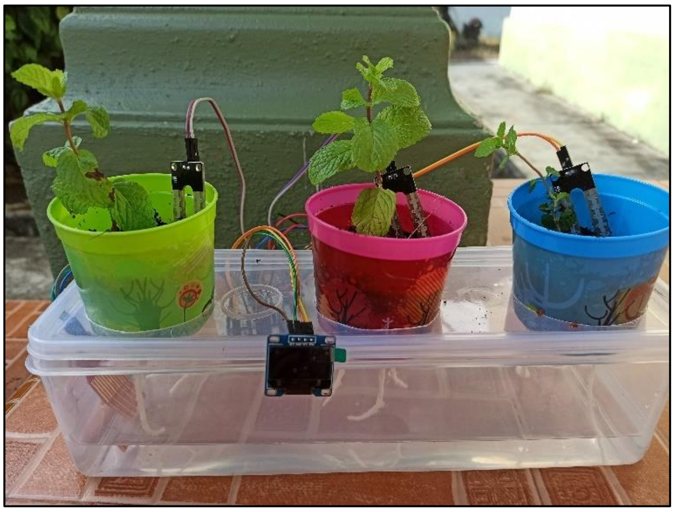


Figure 5.4 Hydroponic Wick System with completed prototype

5.2 Black box Testing

Black Box Testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. In Black Box Testing, this project focus on input and output data of sensor.

5.3 Testing Description

Black Box Testing is used to test the system to make sure the system is work well. The testing has been done by seven days to collect the data from the sensor. In seven days, the testing has been done one hour for every day which is at 10AM. The testing consists temperature and humidity sensor, water level sensor and three soil moisture sensors for each pot. For this project testing, the testing is more focus more on water level sensor and soil moisture sensor. However, this data for DHT22 also will be collected. For each sensor data for expected result must be keyed in first before testing the sensor to get the actual result.

The first sensor that are used in this project is DHT22. DHT22 is to determine the temperature and humidity of the surrounding. For having a healthy plant, temperature and humidity are important because it can affect the plant growth. However, there is no specific task for DHT22 because temperature and humidity cannot determine the expected result. Temperature and humidity are only for collecting the data.

The second sensor is water level sensor. Water level sensor is to determine the level of the water in the container. Before collecting the data, the level of water must be measure first as expected result. The level of water will be measure using the ruler.

The last sensor is soil moisture sensor. This sensor is to determine whether the soil ‘Need Watering’ or ‘Does Not Need Watering’. Before starting testing phase, this sensor has been tested on fresh dry soil. This is because to determine whether the soil is in need of watering or not. If the data for soil moisture are more than 533, the plant ‘Need Watering’ and if data of soil moisture sensor is less than 533, the plant ‘Does Not Need Watering’. If on that day, the soil moisture sensor shows more than 533, the plant will be watered directly to identify the soil moisture sensor is working well or not.

Table 5.1 shows the form that needed to be filled as expected result before starting the testing. The next table is Table 5.2. Table 5.2 shows the form which needed to be filled during the testing. The testing to fill in the form are only 10 data.

Table 5.1 Expected Result form

Day					
Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
-	-				

Table 5.2 Form Testing

Day 1						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

5.4 Result and Findings

This subchapter presents the result for DHT22, water level sensor and soil moisture sensor for each pot. Data for each sensor will be presented in the table which has two table. The table is showing data collected at 10 AM. Furthermore, all the data were summarized in graph.

5.4.1 DHT22 sensor

Table 5.3 and Table 5.4 shows the data from DHT22 sensor. Table 5.3 shows the data of temperature and Table 5.4 shows the data of humidity. Both of the table shows the temperature and humidity of surrounding.

Table 5.3 Data of Temperature

Days	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
1	30.7	30.4	30.1	30.9	30.4	30.3	30.3	30.3	30.5	30.8
2	31.4	31.7	31.6	31.7	31.2	31	31	31.5	31.9	31.6
3	31	31.6	32	31.7	32	31.1	31.7	32	31.6	31
4	29.2	29.6	29.5	29.9	29.7	29.1	29.2	29.9	30	30
5	29.3	29	29.3	29.2	29.2	29.6	29.5	29.1	29.3	29.7
6	30.5	30	31	30.3	30.9	30.3	30.5	30.5	30.5	30.9
7	31	31.8	31.6	31	31	31.9	32	31.7	31.5	31.9

Table 5.4 Data of Humidity

Days	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
1	82	82.2	82.1	82	81.9	82	81.8	82	81.9	81.4
2	90.4	90.1	90	90.5	90.4	90.8	90.5	90.8	90.3	90.9
3	94.8	94.1	94.4	94.3	94.3	94.5	94.4	94.6	94.8	94.8
4	93.3	93.3	93.1	93.9	93.2	93.2	93.8	94	93.5	93.8
5	94.3	94.4	94	94.2	94.6	94	95	94	94	94.4
6	90	90	90.6	90.4	90.4	90.2	90.8	90.8	90.9	90.8
7	79	79.2	79.2	79.2	79.4	79.1	79.4	80	79.9	80

Figure 5.5 shows the graph of temperature of DHT22 sensor. The graph shows the highest temperature was on Day 3 and the lowest temperature was on Day 5. Furthermore, the data of temperature shows increase and decrease on the graph. There are changing of temperature for each day. However, the range of temperature change are 0.1°C until 0.5°C.

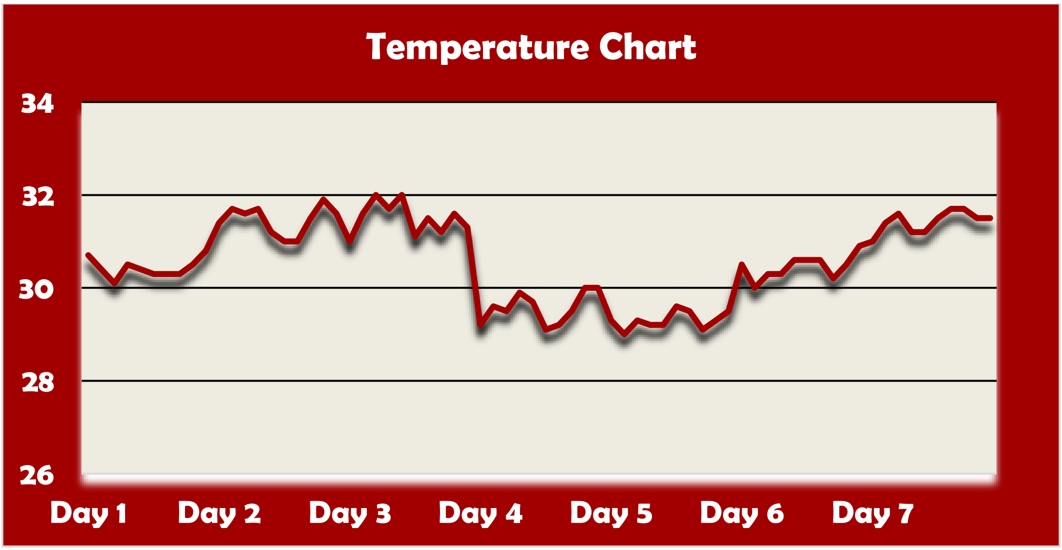


Figure 5.5 Temperature Chart

Figure 5.6 shows the humidity data in the graph. Based on graph, the lowest humidity is at Day 7. Besides, highest data of humidity is at Day 3. Based on the data in Figure 5.2, the data are slightly change in a day. The changes of data in a day are not more than 1%.

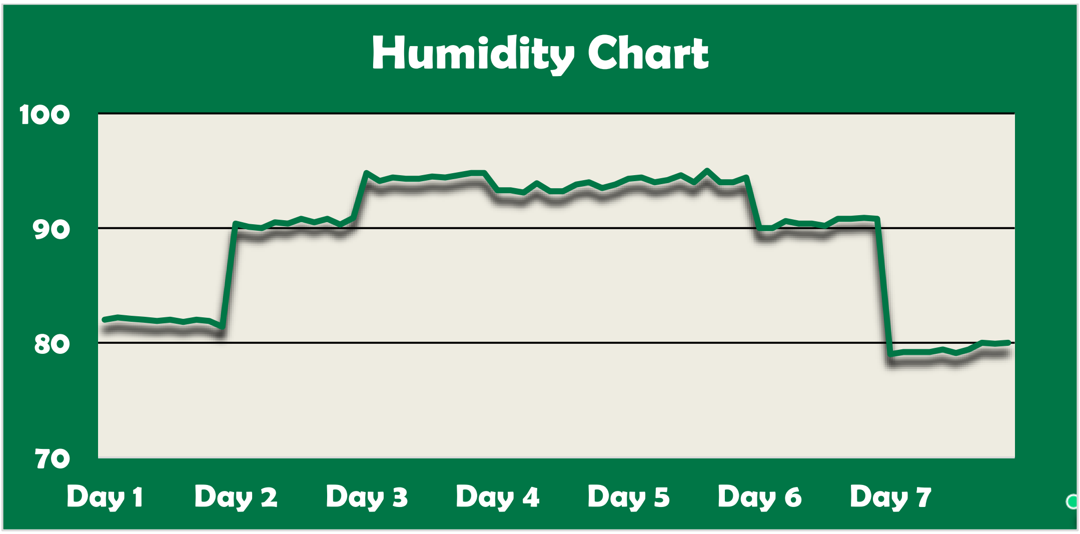


Figure 5.6 Humidity Chart

From this analysis, it has proven that DHT22 sensor can detect the surrounding temperature and humidity. It is because, the changes for temperature are not more than or less than 0.5°C and changes of humidity are not more than or less than 1%. Based on the previous research, this sensor can be proven that temperature and humidity sensor work well because the changes of data are slightly different.

5.4.2 Water Level Sensor

Table 5.5 shows the expected value for water level sensor. Water level sensor is the sensor to detect the water level in hydroponic container. The expected value is taken before power on the sensor which is make calibration for the water level. Furthermore, the expected value is determine using a ruler. Besides, Table 5.6 shows the data that collected in 7 days for water level sensor. The data of the sensor are determined in millimeters(mm).

Table 5.5 Expected Result for Water Level

Days	Expected Result
1	35
2	34
3	34
4	33
5	32
6	32
7	31

Table 5.6 Data of Water Level

Days	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
1	38	37.9	35.7	35.1	35.1	34.9	35.2	35.2	35.1	35.1
2	35.4	36.7	35.4	35.4	37.1	36.8	34.7	34.7	34.7	34.7
3	36.9	35.2	34.5	34.5	34.1	34.3	34.5	34.4	34.4	34.4
4	33.8	37.3	33.8	33.4	33.8	33.8	34.1	33.8	33.8	33.8
5	34.7	37.6	33.7	33.2	33.2	33.5	33.2	33.2	33.2	33.2
6	38.2	35.5	32.9	32.9	32.2	33.1	33.1	33.1	32.9	32.9
7	33.3	37.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5

Figure 5.7 show the graph based on table data in Table 5.6. On the Day 6, water level of data shows the highest data which is 38.2 mm. The data are changes in a day for each day are sometimes rapidly. The highest changing water level sensor are on Day 6 which is from 33.2mm until 38.2mm and on the Day 7 which is from 37.5mm until 32.3mm. The differences for Day 6 and Day 7 are 5mm.

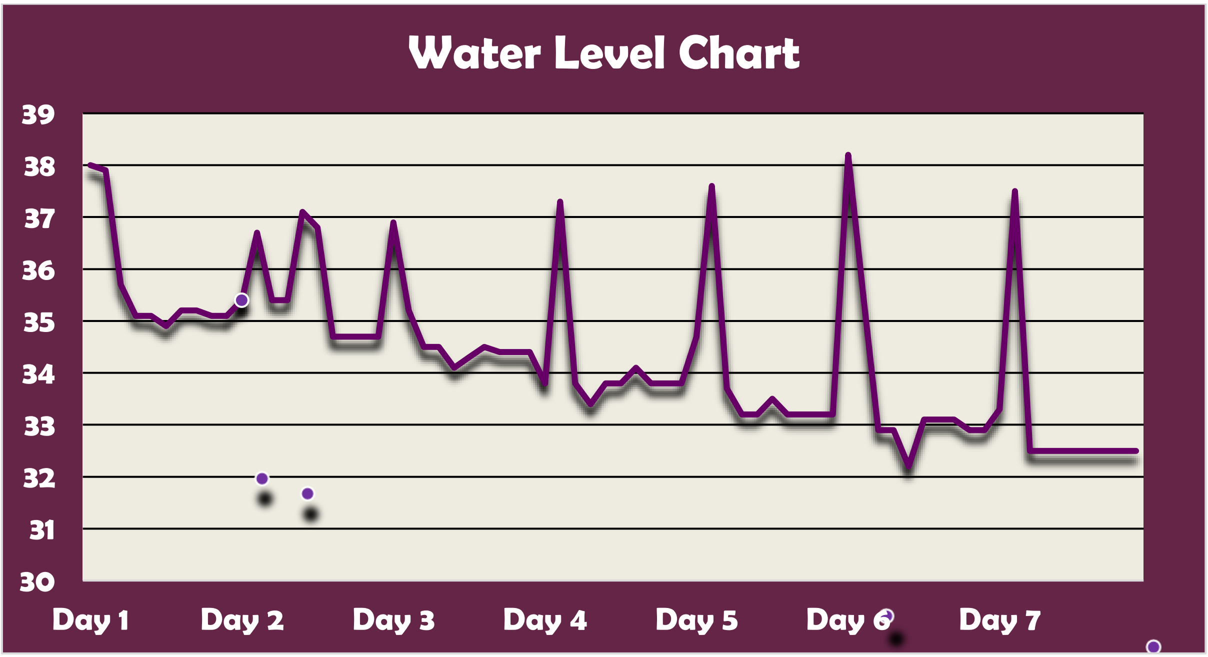


Figure 5.7 Water Level Chart

From this analysis, the water level sensor is not functioning well. Based on observations during testing phase, the level of water can be change rapidly if it has water waves on the container. That is why the water level sometimes change rapidly. Besides, to get the accurate result, the sensor position must be in upright position. In an addition, the sensor sometimes cannot detect the actual value which it needs to immersed into water slowly.

5.4.3 Soil Moisture Sensor

There are three pots mini hydroponic in this project. Hence, three soil moisture sensors had been tested for the soil in mini hydroponic pots. The Table 5.6 shows the expected result for each sensor. All the expected results are indicating the soil 'Does Not Need Watering'. This is because, this hydroponic project is using the wick to transport the nutrient solution into the soil in mini hydroponic pot. By that, the soil is should be 'Does No Need Watering'. The soil moisture sensor will become 'Need Watering' if there is no nutrient solution in the container.

Table 5.7 Expected Soil Moisture

Days	Soil 1	Soil 2	Soil 3
1	Does No Need Watering	Does No Need Watering	Does No Need Watering
2	Does No Need Watering	Does No Need Watering	Does No Need Watering
3	Does No Need Watering	Does No Need Watering	Does No Need Watering
4	Does No Need Watering	Does No Need Watering	Does No Need Watering
5	Does No Need Watering	Does No Need Watering	Does No Need Watering
6	Does No Need Watering	Does No Need Watering	Does No Need Watering
7	Does No Need Watering	Does No Need Watering	Does No Need Watering

Table 5.8 shows the summarized result of soil moisture sensor for mini pot 1. From this table, most of the data shows the plant in pot are 'Does No Need Watering'. However, there are only two data in Day 2 indicates the plant is 'Need Watering'.

Table 5.8 Data of Soil Moisture 1

Days	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
1	480	490	494	480	489	484	493	490	493	484
2	472	492	527	517	512	522	536	538	524	522
3	438	441	440	439	440	442	438	441	451	451
4	430	432	495	480	457	464	467	456	438	473
5	456	481	530	531	518	509	481	501	486	501
6	476	477	484	485	483	477	490	483	490	483
7	488	501	496	497	488	509	494	492	499	506

Table 5.9 shows the summarized for soil moisture sensor. On the Day 2, Day 4 and Day 5 are shows the data is lower than 533. This is indicating that the plant is 'Need Watering'. However, in most of the day are showing the soil is moist.

Table 5.9 Data of Soil Moisture 2

Days	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
1	450	453	455	460	460	455	453	459	457	452
2	522	532	511	530	532	505	492	523	532	564
3	473	479	478	487	483	479	473	478	476	480
4	481	518	514	535	546	541	532	529	504	534
5	476	525	506	484	500	534	504	480	493	496
6	403	405	414	412	405	418	407	405	417	417
7	456	459	481	484	464	469	469	463	467	488

Table 5.10 shows the summarized data of soil moisture sensor for mini pot. As expected, the plant is 'Does Not Need Watering'. The plant on Day 4 and Day 6 are 'Need Watering' compared with another days. Besides, the lowest data shows on Day 1 which indicates 'No Need Watering'.

Table 5.10 Data of Soil Moisture 3

Days	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
1	387	400	393	394	388	390	390	400	399	396
2	437	442	437	443	439	442	443	437	437	439
3	387	396	392	400	395	389	388	395	395	389
4	489	544	532	543	534	522	539	523	492	487
5	402	411	413	415	409	404	414	414	410	417
6	432	535	527	545	540	538	523	510	494	498
7	423	443	442	439	433	442	435	424	444	455

Figure 5.8 shows the summarized data of soil moisture on the mini pot for Soil 1, Soil 2 and Soil 3. The graph is based on Table 5.7, Table 5.8 and Table 5.9. The value for each day is mostly change. Although the data are mostly change, the value are not rapidly change.

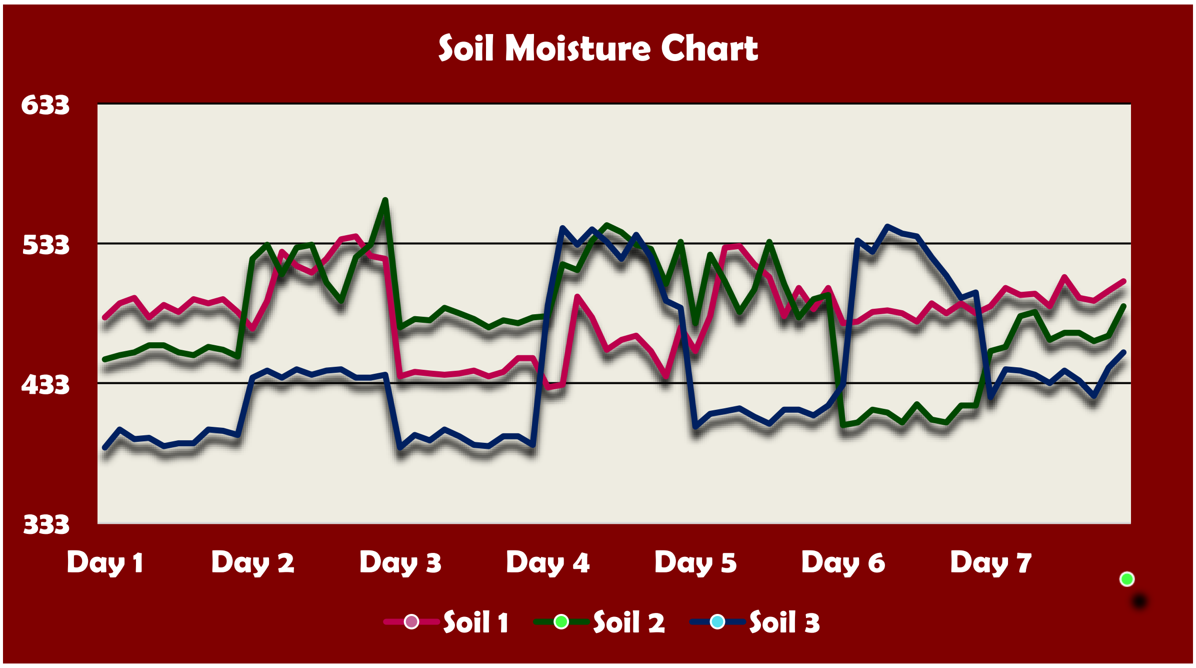


Figure 5.8 Soil Moisture Chart

From the analysis, this can be summarized that three soil moisture sensor for each pot are can detect the soil is moist or not. It is because, when the plant of that day needs watering, the plant will be watered directly. As the result, the data will be drop which indicates the plant does no need watering. This wick hydroponic project actually does no need watering directly to the plant because the wick is the medium to transport nutrient solution into soil. The soil may remain moist for a long time period. Hence, the soil which watering directly will maintain moist on the next day.

5.5 Conclusion

Based on the testing has been done on hydroponic wick system, this project is proven that DHT22 sensor and soil moisture sensor for each pot are works well. In an addition, this will help the user to monitor the condition of their plants. However, the water level sensor was not work well as expected. It needs the user to immersed the water level sensor slowly to get the correct result.

CHAPTER 6

CONCLUSION

6.1 Introduction

This chapter will cover the summarization of all the IoT Based Monitoring Hydroponic Wick System. In addition, the recommendation for improvement of this system will be also discussed in this chapter. Hence, indirectly it will be useful reference for a future developer who will develop the same of this project.

6.2 Project Contributions

This project has developed the IoT-Based Monitoring Wick Hydroponic System to ease user in monitoring their hydroponic plant. User will be notified the temperature and humidity of surrounding which is important for plant growth. Besides, the hydroponic plants are more focus on the level of water and the moisture of soil. The water which contain the nutrient solution will be transferred to soil for plant gain the nutrient. That is important for user to know the level of water for make sure the soil always moist. The way for the user to know the temperature, humidity, water level and soil moisture for each pot is using the sensor. This project used DHT22 sensor for temperature and humidity, water level sensor and soil moisture sensor to provide the data for people.

Furthermore, there are web-based system for people to monitoring their hydroponic plants. When the user power on the microcontroller, the data from sensor will sent to MySQL database. Moreover, user may know the current data and can read the data in graph. Thus, the user probably knows the data from sensors are increasing or decreasing.

6.3 Conclusion

In conclusion, this project has met the first objective which is to design web-based interface for status monitoring of indoor garden. The web application was created based on windows platform using PHP and JavaScript language. This project also achieved the second objective which is develop prototype of IoT monitoring system for hydroponic in small space and at any time to monitor the plant. By developing the prototype in small space, this will help the user to grow hydroponic plant in their house which only required a small space. Lastly, the third objective of this project is to evaluate the functionality of IoT monitoring for hydroponic using Blackbox Testing method. This has been through the sensor validation done. These results showed that prototype worked the way it was supposed to be and in a consistent manner.

6.4 Problems and Limitations

There are a few problems and limitation of this project throughout the process of developing and testing. The problems and limitations are listed as follows:

i) DHT22 sensor which read the temperature and humidity of surrounding does not have the expected value. This cannot be done for testing DHT22 sensor.

ii) The water level sensor found very difficult in determine the level of water. The formula for determine the level of water to make sure the max value is 40mm had to be changed several times.

6.5 Recommendation for Future Work

In the process to develop this prototype, there are a lot of things that have completed in order to achieved the objective of the project. This project was developed in the limited time scope in order to complete this project. Future enhancement of the project that can be recommended to developed a better system for the future are:

i) Provide the automatic pump of nutrient solution for maintain the soil moisture. This feature will be very needed for user where they do no need to watering the plant when the plant is drying.

ii) Provide the other sensor such as light sensor and pH sensor. Both sensors are needed for hydroponic plant growth which light sensor required for the plant to get the amount of light and pH sensor for having the correct pH of nutrient solution.

REFERENCES

Abagissa, A. T., Behura, A., & Pani, S. K. (2018). IoT Based Smart Agricultural Device Controlling System. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018, Icicct*, 26–30. https://doi.org/10.1109/ICICCT.2018.8473126

Aliac, C. J. G., & Maravillas, E. (2019). IOT hydroponics management system. *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*, 1–5. https://doi.org/10.1109/HNICEM.2018.8666372

Anvekar, R. G., Banakar, R. M., Hgx, E. E. Y. E., Bhat, R. R., Frp, U. J., Lvw, D. H., Ri, P., Xvh, Z., & Xqltxh, W. (n.d.). *In IoT Based System Model*. 5–9.

Caya, M. V. C., Babila, A. P., Bais, A. M. M., Im, S. J. V., & Maramba, R. (2018). Air pollution and particulate matter detector using raspberry Pi with IoT based notification. *HNICEM 2017 - 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management*, *2018-Janua*, 1–4. https://doi.org/10.1109/HNICEM.2017.8269490

Gómez Maureira, M. A., Oldenhof, D., & Teernstra, L. (2014). ThingSpeak – an API and Web Service for the Internet of Things. *World Wide Web*. https://staas.home.xs4all.nl/t/swtr/documents/wt2014_thingspeak.pdf

Gupta, M. S. D., Patchava, V., & Menezes, V. (2016). Healthcare based on IoT using Raspberry Pi. *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015*, 796–799. https://doi.org/10.1109/ICGCIoT.2015.7380571

Hafiz Talib, M., Aidil Azlin Abd Rahman, K., & Shahrizal Dollah, M. (2018). Analyzing hydroponic rack design for apartment house. In *International Journal of Engineering & Technology* (Vol. 7, Issue 2.14). https://doi.org/10.14419/ijet.v7i2.14.12812

Hunt, A., & Watkiss, P. (2011). Climate change impacts and adaptation in cities: A review of the literature. *Climatic Change*, *104*(1), 13–49. https://doi.org/10.1007/s10584-010-9975-6

Keerthana, S. Devika, K. Sathiyadevi, S. Priyanka, M. (2018). Automating and Analysing Greenhouse Hydroponic Farms using IOT. *International Journal for Research in Applied Science and Engineering Technology*, *6*(3), 3325–3329.

https://doi.org/10.22214/ijraset.2018.3706

Lakhiar, I. A., Jianmin, G., Syed, T. N., Chandio, F. A., Buttar, N. A., & Qureshi, W. A. (2018). Monitoring and control systems in agriculture using intelligent sensor techniques: A review of the aeroponic system. *Journal of Sensors*, *2018*. https://doi.org/10.1155/2018/8672769

Lee, S., & Lee, J. (2015). Beneficial bacteria and fungi in hydroponic systems: Types and characteristics of hydroponic food production methods. *Scientia Horticulturae*, *195*, 206–215. https://doi.org/10.1016/j.scienta.2015.09.011

Lekjaroen, K., Ponganantayotin, R., Charoenrat, A., Funilkul, S., Supasitthimethee, U., & Triyason, T. (2016). Application for the Elderly. *2016 International Computer Science and Engineering Conference (ICSEC)*, 1–6. https://doi.org/10.1109/ICSEC.2016.7859873

Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, *03*(05), 164–173. https://doi.org/10.4236/jcc.2015.35021

Malche, T. (2017). *System*. 65–70.

Mehra, M., Saxena, S., Sankaranarayanan, S., Tom, R. J., & Veeramanikandan, M. (2018). IoT based hydroponics system using Deep Neural Networks. *Computers and Electronics in Agriculture*, *155*(August), 473–486. https://doi.org/10.1016/j.compag.2018.10.015

Min, B., & Park, S. J. (2018). A smart indoor gardening system using IoT technology. *Lecture Notes in Electrical Engineering*, *474*, 683–687. https://doi.org/10.1007/978-981-10-7605-3_110

Moretti, S., Jesuthasan, N., Caroppo, A., Zangiacomi, A., Leone, A., Sacco, M., & Vezzoli, A. (2017). *Wireless Mobile Communication and Healthcare*. *192*, 367–374. https://doi.org/10.1007/978-3-319-58877-3

Nidhra, S., & Dondeti, J. (2012). *B LACK BOX AND W HITE B OX T ESTING T ECHNIQUES – A L ITERATURE R EVIEW*. *2*(2), 29–50.

P.N.V.S.N, M., Rao, S. T., & Rao, G. M. (2017). Home Automation using Telegram. *Ijarcce*, *6*(6), 64–69. https://doi.org/10.17148/ijarcce.2017.6613

Pasha, S. (2016). Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis. *Int. J. New Technol. Res.*, *2*(6), 19–23. https://doi.org/ISSN : 2454-4116

Rajkumar, R., Dharmaraj, R., & Scholar, P. (2018). A Novel Approach for Smart Hydroponic Farming Using IoT. *International Journal of Engineering Research in Computer Science and Engineering*, *5*(5), 18–23. https://www.technoarete.org/common_abstract/pdf/IJERCSE/v5/i5/Ext_71306.pdf

Rani, R., Lavanya, S., & Poojitha, B. (2018). IoT Based Home Security System Using Raspberry Pi with Email and Voice Alert. *International Journals of Advanced Research in Computer Science and Software Engineering*, *e-8*(4), 119–123.

Shingala, K., & Patel, J. (2017). Automatic Home Appliances and Security of Smart Home with RFID , SMS , Email and Real Time Algorithm Based on IOT. *International Research Journal of Engineering and Technology(IRJET)*, *4*(4), 1958–1964. https://www.irjet.net/archives/V4/i4/IRJET-V4I4409.pdf

Singh, H., Pallagani, V., Khandelwal, V., & Venkanna, U. (2018). IoT based smart home automation system using sensor node. *Proceedings of the 4th IEEE International Conference on Recent Advances in Information Technology, RAIT 2018*, 1–5. https://doi.org/10.1109/RAIT.2018.8389037

Sisyanto, R. E. N., Suhardi, & Kurniawan, N. B. (2018). Hydroponic smart farming using cyber physical social system with telegram messenger. *2017 International Conference on Information Technology Systems and Innovation, ICITSI 2017 - Proceedings*, *2018-Janua*, 239–245. https://doi.org/10.1109/ICITSI.2017.8267950

Smirnov, A., Levashova, T., Shilov, N., & Sandkuhl, K. (n.d.). *Ontology for Cyber-Physical-Social Systems Self- Organisation*.

Tse, M. M. Y. (2010). Therapeutic effects of an indoor gardening programme for older people living in nursing homes. *Journal of Clinical Nursing*, *19*(7–8), 949–958. https://doi.org/10.1111/j.1365-2702.2009.02803.x

Types of Hydroponics Systems. - Soilless. (n.d.). Retrieved November 15, 2019, from http://www.soilless.org/hydroponics/types-hydroponics-systems/

Ullah, A., Aktar, S., Sutar, N., Kabir, R., & Hossain, A. (2019). *Cost Effective Smart Hydroponic Monitoring and Controlling System Using IoT*. 142–154. https://doi.org/10.4236/ica.2019.104010

Van, L. Da, Lin, Y. B., Wu, T. H., Lin, Y. W., Peng, S. R., Kao, L. H., & Chang, C. H. (2019). Planttalk: A smartphone-based intelligent hydroponic plant box. *Sensors (Switzerland)*, *19*(8). https://doi.org/10.3390/s19081763

Van, L.-D., Lin, Y.-B., Wu, T.-H., Lin, Y.-W., Peng, S.-R., Kao, L.-H., & Chang, C.-H. (n.d.). *PlantTalk: A Smartphone-Based Intelligent Hydroponic Plant Box*. https://doi.org/10.3390/s19081763

Virginia I. Lohr. (2010). What are the benefits of REDD + and why do they. *October, 881*(October), 675–682.

Wang, D., & MacMillan, T. (2013). The Benefits of Gardening for Older Adults: A Systematic Review of the Literature. *Activities, Adaptation and Aging*, *37*(2), 153–181. https://doi.org/10.1080/01924788.2013.784942

APPENDICES

Appendix A : Testing Result

Day 1						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	30.7	82	38	480	450	387
2	30.4	82.2	37.9	490	453	400
3	30.1	82.1	35.7	494	455	393
4	30.5	82	35.1	480	460	394
5	30.4	81.9	35.1	489	460	388
6	30.3	82	34.9	484	455	390
7	30.3	81.8	35.2	493	453	390
8	30.3	82	35.2	490	459	400
9	30.5	81.9	35.1	493	457	399
10	30.8	81.4	35.1	484	452	396

Day 2						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	31.4	90.4	35.4	472	522	437
2	31.7	90.1	36.7	492	532	442
3	31.6	90	35.4	527	511	437
4	31.7	90.5	35.4	517	530	443
5	31.2	90.4	37.1	512	532	439
6	31	90.8	36.8	522	505	442
7	31	90.5	34.7	536	492	443
8	31.5	90.8	34.7	538	523	437
9	31.9	90.3	34.7	524	532	437
10	31.6	90.9	34.7	522	564	439

Day 3						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	31	94.8	36.9	438	473	387
2	31.6	94.1	35.2	441	479	396
3	32	94.4	34.5	440	478	392
4	31.7	94.3	34.5	439	487	400
5	32	94.3	34.1	440	483	395
6	31.1	94.5	34.3	442	479	389
7	31.5	94.4	34.5	438	473	388
8	31.2	94.6	34.4	441	478	395
9	31.6	94.8	34.4	451	476	395
10	31.3	94.8	34.4	451	480	389

Day 4						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	29.2	93.3	33.8	430	481	489
2	29.6	93.3	37.3	432	518	544
3	29.5	93.1	33.8	495	514	532
4	29.9	93.9	33.4	480	535	543
5	29.7	93.2	33.8	457	546	534
6	29.1	93.2	33.8	464	541	522
7	29.2	93.8	34.1	467	532	539
8	29.5	94	33.8	456	529	523
9	30	93.5	33.8	438	504	492
10	30	93.8	33.8	473	534	487

Day 5						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	29.3	94.3	34.7	456	476	402
2	29	94.4	37.6	481	525	411
3	29.3	94	33.7	530	506	413
4	29.2	94.2	33.2	531	484	415
5	29.2	94.6	33.2	518	500	409
6	29.6	94	33.5	509	534	404
7	29.5	95	33.2	481	504	414
8	29.1	94	33.2	501	480	414
9	29.3	94	33.2	486	493	410
10	29.5	94.4	33.2	501	496	417

Day 6						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	30.5	90	38.2	476	403	432
2	30	90	35.5	477	405	535
3	30.3	90.6	32.9	484	414	527
4	30.3	90.4	32.9	485	412	545
5	30.6	90.4	32.2	483	405	540
6	30.6	90.2	33.1	477	418	538
7	30.6	90.8	33.1	490	407	523
8	30.2	90.8	33.1	483	405	510
9	30.5	90.9	32.9	490	417	494
10	30.9	90.8	32.9	483	417	498

Day 7						
No.	Temperature	Humidity	Water Level	Soil 1	Soil 2	Soil 3
1	31	79	33.3	488	456	423
2	31.4	79.2	37.5	501	459	443
3	31.6	79.2	32.5	496	481	442
4	31.2	79.2	32.5	497	484	439
5	31.2	79.4	32.5	488	464	433
6	31.5	79.1	32.5	509	469	442
7	31.7	79.4	32.5	494	469	435
8	31.7	80	32.5	492	463	424
9	31.5	79.9	32.5	499	467	444
10	31.5	80	32.5	506	488	455

Appendix B : Coding

dbconnect.php

```
<?php
$host = "localhost";
$userName = "root";
$password = "";
$dbName = "hydroponic";
// Create database connection
$conn = new mysqli($host, $userName, $password, $dbName);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
```

dashboard.php

```
<?php
                    include 'dbconnect.php';


     $query = 'SELECT * FROM value ORDER BY id DESC LIMIT 1';


     $result = mysqli_query($conn,$query) or die('SQL error');
     $row = mysqli_fetch_array($result)



?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>Dashboard</title>
  <!-- Favicon icon -->
  <link rel="icon" type="image/png" sizes="16x16" href="images/leave.png">
  <!-- Pignose Calender -->
  <link href="./plugins/pg-calendar/css/pignose.calendar.min.css" rel="stylesheet">
  <!-- Chartist -->
  <link rel="stylesheet" href="./plugins/chartist/css/chartist.min.css">
  <link rel="stylesheet" href="./plugins/chartist-plugin-tooltips/css/chartist-plugin-
tooltip.css">
  <!-- Custom Stylesheet -->
  <link href="css/style.css" rel="stylesheet">
```

```
  <link href="styles1.css" rel="stylesheet">
</head>


<body>


  <!--*******************
    Preloader start
  ********************-->
  <div id="preloader">
    <div class="loader">
      <svg class="circular" viewBox="25 25 50 50">
        <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="3" stroke-
miterlimit="10" />
      </svg>
    </div>
  </div>
  <!--*******************
    Preloader end
  ********************-->



  <!--**********************************
    Main wrapper start
  ***********************************-->
  <div id="main-wrapper" style="background-color:      #E6E6FA;">


    <!--**********************************
      Nav header start
    ***********************************-->
```

```
<div class="nav-header" style="background-color:    #02407a;">
   <div class="brand-logo">
      <a href="index.html">
         <h2 style="color:       #E6E6FA;">HYDROPONIC</h2>
      </a>
   </div>
</div>
<!--**********************************
   Nav header end
***********************************-->

<!--**********************************
   Header start
***********************************-->
<div class="header">
   <div class="header-content clearfix">

      <div class="nav-control">
         <div class="hamburger">
            <span class="toggle-icon"><i class="icon-menu"></i></span>
         </div>
      </div>



   </div>
</div>
<!--**********************************
   Header end ti-comment-alt
***********************************-->
```

```
<!--**********************************
    Sidebar start
***********************************-->
<div class="nk-sidebar">
    <div class="nk-nav-scroll">
        <ul class="metismenu" id="menu">

            <li>
                <a href="dashboard.php" aria-expanded="false">
                    <i class="icon-speedometer menu-icon"></i><span class="nav-
text">Dashboard</span>
                </a>
            </li>
            <li>
                <a href="table.php" aria-expanded="false">
                    <i class="fa fa-table menu-icon"></i><span class="nav-
text">Table</span>
                </a>
            </li>
        </ul>
    </div>
</div>
<!--**********************************
    Sidebar end
***********************************-->

<!--**********************************
    Content body start
```

```
************************************-->

<div class="content-body" style="background-color:           #E6E6FA;">

  <div class="container-fluid mt-3">
    <div class="row">
      <div class="col-lg-4 col-sm-6">
        <div class="card gradient-11" style="  position: relative;
        width: 100%;
        height: 220px;
        overflow: hidden;
        padding-top: 56.25%;
        background-color: #003566">
          <div class="card-body">
            <div class="d-inline-block">
              <iframe class="responsive-iframe" src="gaugetemp.php"></iframe>
            </div>
          </div>
        </div>
      </div>
      <div class="col-lg-4 col-sm-6">
        <div class="card gradient-11" style="  position: relative;
        width: 100%;
        height: 220px;
        overflow: hidden;
        padding-top: 56.25%;
        background-color: #003566">
          <div class="card-body">
            <div class="d-inline-block">
```

```
                        <iframe class="responsive-iframe" src="gaugehumid.php"></iframe>
                    </div>
                </div>
            </div>
        </div>
        <div class="col-lg-4 col-sm-6">
            <div class="card gradient-11" style="  position: relative;
            width: 100%;
            height: 220px;
            overflow: hidden;
            padding-top: 56.25%;
            background-color: #003566">
                <div class="card-body">
                    <div class="d-inline-block">
                        <iframe class="responsive-iframe"
src="gaugewaterlvl.php"></iframe>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-lg-4 col-sm-6">
            <div class="card gradient-1">
                <div class="card-body">
                    <h3 class="card-title text-white">Soil Moisture (Pot 1)</h3>
                    <div class="d-inline-block">
                        <h3 class="text-white"><?php
```

```
                        $soil1 = $row['soilmoisture1'];
                          if ($soil1 < "533") {
                          echo "No Need Watering";
                          }
                          else{
                            echo "Need Watering";
                          }
                        ?></h3>
                  <p class="text-white mb-0"><?php echo
$row['soilmoisture1'];?></p>
                </div>
                <span class="float-right display-5 opacity-5"><i class="fa fa-
tree"></i></span>
              </div>
            </div>
          </div>
          <div class="col-lg-4 col-sm-6">
            <div class="card gradient-2">
              <div class="card-body">
                <h3 class="card-title text-white">Soil Moisture (Pot 2)</h3>
                <div class="d-inline-block">
                  <h3 class="text-white"><?php
                        $soil2 = $row['soilmoisture2'];
                          if ($soil2 < "533") {
                          echo "No Need Watering";
                          }
                          else{
                            echo "Need Watering";
                          }
```

```
                              ?></h3>
                        <p class="text-white mb-0"><?php echo
$row['soilmoisture2'];?></p>
                      </div>
                      <span class="float-right display-5 opacity-5"><i class="fa fa-
tree"></i></span>
                    </div>
                  </div>
                </div>
                <div class="col-lg-4 col-sm-6">
                  <div class="card gradient-3">
                    <div class="card-body">
                      <h3 class="card-title text-white">Soil Moisture (Pot 3)</h3>
                      <div class="d-inline-block">
                        <h3 class="text-white"><?php
                              $soil3 = $row['soilmoisture3'];
                                if ($soil3 < "533") {
                                echo "No Need Watering";
                                }
                                else{
                                  echo "Need Watering";
                                }
                              ?></h3>
                        <p class="text-white mb-0"><?php echo
$row['soilmoisture3'];?></p>
                      </div>
                      <span class="float-right display-5 opacity-5"><i class="fa fa-
tree"></i></span>
                    </div>
                  </div>
```

```
            </div>


          </div>


          <div class="row">
            <div class="col-lg-12">
              <div class="row">
                <div class="col-6">
                  <div class="card" style="background-color:      rgb(95, 145, 160);">
                    <div class="d-inline-block">
                      <div class="card-body pb-0 d-flex justify-content-between">
                        <div>
                          <h3 class="m-0">Temperature</h3>
                        </div>
                        <div>
                          <ul>


                          </ul>
                        </div>
                      </div>
                      <div class="chart-wrapper">
                        <canvas id="mycanvas"></canvas>
                      </div>
                      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
                      <script type="text/javascript" src="js/Chart.min.js"></script>
                      <script type="text/javascript" src="js/tsptemp.js"></script>
                    </div>
                  </div>
```

```
                </div>


                <div class="col-6">
                  <div class="card" style="background-color:  rgb(95, 145, 160);">
                    <div class="d-inline-block">
                      <div class="card-body pb-0 d-flex justify-content-between">
                        <div>
                          <h3 class="m-0">Humidity</h3>
                        </div>
                        <div>
                          <ul>


                          </ul>
                        </div>
                      </div>
                      <div class="chart-wrapper">
                        <canvas id="mycanvas2"></canvas>
                      </div>
                      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
                      <script type="text/javascript" src="js/Chart.min.js"></script>
                      <script type="text/javascript" src="js/tsphumid.js"></script>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <div class="row">
```

```
            <div class="col-lg-12">
              <div class="row">
                <div class="col-6">
                  <div class="card" style="background-color:  rgb(95, 145, 160);">
                    <div class="d-inline-block">
                      <div class="card-body pb-0 d-flex justify-content-between">
                        <div>
                          <h3 class="m-0">Water Level</h3>
                        </div>
                        <div>
                          <ul>


                          </ul>
                        </div>
                      </div>
                      <div class="chart-wrapper">
                        <canvas id="mycanvas3"></canvas>
                      </div>
                      <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
                      <script type="text/javascript" src="js/Chart.min.js"></script>
                      <script type="text/javascript" src="js/tspwaterlevel.js"></script>
                    </div>
                  </div>
                </div>


                <div class="col-6">
                  <div class="card" style="background-color:  rgb(95, 145, 160);">
                    <div class="d-inline-block">
```

```
                    <div class="card-body pb-0 d-flex justify-content-between">
                      <div>
                        <h3 class="m-0">Soil Moisture</h3>
                      </div>
                      <div>
                        <ul>
                        </ul>
                      </div>
                    </div>
                    <div class="chart-wrapper">
                      <canvas id="mycanvas4"></canvas>
                    </div>
                    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
                    <script type="text/javascript" src="js/Chart.min.js"></script>
                    <script type="text/javascript" src="js/tspsoilmoisture.js"></script>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
        </div>
        </div>
      </div>
      <!-- #/ container -->
    </div>
    <!--**********************************
      Content body end
```

```
   ***********************************-->


   <!--**********************************
      Footer start
   ***********************************-->


   <!--**********************************
      Footer end
   ***********************************-->
</div>
<!--**********************************
   Main wrapper end
***********************************-->


<!--**********************************
   Scripts
***********************************-->
<script src="plugins/common/common.min.js"></script>
<script src="js/custom.min.js"></script>
<script src="js/settings.js"></script>
<script src="js/gleek.js"></script>
<script src="js/styleSwitcher.js"></script>


<!-- Chartjs -->
<script src="./plugins/chart.js/Chart.bundle.min.js"></script>
<!-- Circle progress -->
<script src="./plugins/circle-progress/circle-progress.min.js"></script>
<!-- Datamap -->
<script src="./plugins/d3v3/index.js"></script>
```

```
    <script src="./plugins/topojson/topojson.min.js"></script>

    <script src="./plugins/datamaps/datamaps.world.min.js"></script>

    <!-- Morrisjs -->

    <script src="./plugins/raphael/raphael.min.js"></script>

    <script src="./plugins/morris/morris.min.js"></script>

    <!-- Pignose Calender -->

    <script src="./plugins/moment/moment.min.js"></script>

    <script src="./plugins/pg-calendar/js/pignose.calendar.min.js"></script>

    <!-- ChartistJS -->

    <script src="./plugins/chartist/js/chartist.min.js"></script>

    <script src="./plugins/chartist-plugin-tooltips/js/chartist-plugin-
tooltip.min.js"></script>

    <script src="./js/dashboard/dashboard-1.js"></script>

</body>

</html>
```

table.php

```
<?php
      include 'dbconnect.php';


      $query ="SELECT * FROM  value";


      $result = mysqli_query($conn,$query) or die('SQL error');
?>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width,initial-scale=1">
   <title>Dashboard</title>
   <!-- Favicon icon -->
   <link rel="icon" type="image/png" sizes="16x16" href="images/leave.png">
   <!-- Pignose Calender -->
   <link href="./plugins/pg-calendar/css/pignose.calendar.min.css" rel="stylesheet">
   <!-- Chartist -->
   <link rel="stylesheet" href="./plugins/chartist/css/chartist.min.css">
   <link rel="stylesheet" href="./plugins/chartist-plugin-tooltips/css/chartist-plugin-
tooltip.css">
   <!-- Custom Stylesheet -->
   <link href="css/style.css" rel="stylesheet">
   <link href="styles1.css" rel="stylesheet">
</head>
<body>
```

```
<!--*******************
  Preloader start
********************-->
<div id="preloader">
  <div class="loader">
    <svg class="circular" viewBox="25 25 50 50">
      <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="3" stroke-
miterlimit="10" />
    </svg>
  </div>
</div>
<!--*******************
  Preloader end
********************-->
 <!--**********************************
  Main wrapper start
***********************************-->
<div id="main-wrapper">
  <!--**********************************
    Nav header start
  ***********************************-->
  <div class="nav-header" style="background-color:   #02407a;">
    <div class="brand-logo">
      <a href="index.html">
        <h2 style="color:   #E6E6FA;">HYDROPONIC</h2>
      </a>
    </div>
  </div>
  <!--**********************************
```

```
    Nav header end
  ***********************************-->
  <!--***********************************
    Header start
  ***********************************-->
  <div class="header">
    <div class="header-content clearfix">
            <div class="nav-control">
        <div class="hamburger">
          <span class="toggle-icon"><i class="icon-menu"></i></span>
        </div>
      </div>
              </div>
  </div>
  <!--***********************************
    Header end ti-comment-alt
  ***********************************-->
  <!--***********************************
    Sidebar start
  ***********************************-->
  <div class="nk-sidebar">
    <div class="nk-nav-scroll">
      <ul class="metismenu" id="menu">


        <li>
          <a href="dashboard.php" aria-expanded="false">
            <i class="icon-speedometer menu-icon"></i><span class="nav-
text">Dashboard</span>
          </a>
```

```
            </li>

            <li>

              <a href="table.php" aria-expanded="false">

                <i class="fa fa-table menu-icon"></i><span class="nav-
text">Table</span>

              </a>

            </li>

          </ul>

        </div>

      </div>

      <!--*********************************

        Sidebar end

      ***********************************-->


      <!--*********************************

        Sidebar start

      ***********************************-->


      <!--*********************************

        Content body start

      ***********************************-->

      <div class="content-body"style="background-color:   #E6E6FA;">

<!-- row -->

<div class="container-fluid">

   <div class="row">

      <div class="col-12">

        <div class="card">

          <div class="card-body"style="background-color:  #FFFFFF;">

            <h4 class="card-title">Data Table</h4>
```

```
<div class="table-responsive"style="background-color:   #FFFFFF;">
  <table class="table table-striped table-bordered zero-configuration">
    <thead>
      <tr>
        <th style="color:   #000000;">No</th>
        <th style="color:   #000000;">Temperature</th>
        <th style="color:   #000000;">Humidity</th>
        <th style="color:   #000000;">Water Level</th>
        <th style="color:   #000000;">Soil Moisture (Pot 1)</th>
        <th style="color:   #000000;">Soil Moisture (Pot 2)</th>
        <th style="color:   #000000;">Soil Moisture (Pot 3)</th>
        <th style="color:   #000000;">Date & Time</th>
      </tr>
    </thead>
    <tbody>
        <tr>
          <?php
           $count = 0;
           while($row = mysqli_fetch_array($result, MYSQLI_ASSOC))
           {
             ?>
             <td style="color:   #000000;">
             <?php $count = $count+1; print($count);?>
             </td>
             <td style="color:   #000000;"><?php echo $row['temperature'];?
></td>
             <td style="color:   #000000;"><?php echo $row['humidity'];?></t
d>
             <td style="color:   #000000;"><?php echo $row['waterlevel'];?><
/td>
```

```
<td style="color:   #000000;"><?php
   $soil1 = $row['soilmoisture1'];
      if ($soil1 < "533") {
      echo "Does No Need Watering";
      }
      else{
         echo "Need Watering";
      }
   ?></td>
<td style="color:   #000000;"><?php
   $soil2 = $row['soilmoisture2'];
      if ($soil2 < "533") {
      echo "Does No Need Watering";
      }
      else{
         echo "Need Watering";
      }
        ?></td>
<td style="color:   #000000;"><?php
   $soil3 = $row['soilmoisture3'];
      if ($soil3 < "533") {
      echo "Does No Need Watering";
      }
      else{
         echo "Need Watering";
      }
      ?></td>
<td style="color:   #000000;"><?php echo $row['date'];?></td>
                              </tr>
```

```
                      <?php
                     }
                      ?>
                </tbody>
            </table>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- #/ container -->
</div>
    <!--**********************************
      Content body end
    ***********************************-->
     <!--**********************************
      Footer start
    ***********************************-->
    <!--**********************************
      Footer end
    ***********************************-->
  </div>
  <!--**********************************
    Main wrapper end
  ***********************************-->

  <!--**********************************
    Scripts
```

```
    **********************************-->
    <script src="plugins/common/common.min.js"></script>
    <script src="js/custom.min.js"></script>
    <script src="js/settings.js"></script>
    <script src="js/gleek.js"></script>
    <script src="js/styleSwitcher.js"></script>
    <script src="./plugins/tables/js/jquery.dataTables.min.js"></script>
    <script src="./plugins/tables/js/datatable/dataTables.bootstrap4.min.js"></script>
    <script src="./plugins/tables/js/datatable-init/datatable-basic.min.js"></script>
</body>
</html>
```