POLITEKNIK MALAYSIA
SULTAN SALAHUDDIN ABDUL AZIZ SHAH

SMART RESERVOIR TANK

NURUL NAJIHAH BINTI AZIZ

(08DEP20F1028)

DEPARTMENT OF ELECTRICAL ENGINEERING

(COMMUNICATION)

2 2022/2023

POLITEKNIK MALAYSIA
SULTAN SALAHUDDIN ABDUL AZIZ SHAH

SMART RESERVOIR TANK

NURUL NAJIHAH BINTI AZIZ

(08DEP20F1028)

This report was submitted to the Electrical Engineering Department in fulfilment of the requirement for a Diploma in Electrical Engineering (Communication) With Honours

DEPARTMENT OF ELECTRICAL ENGINEERING (COMMUNICATION)

2 2022/2023

CONFIRMATION OF THE PROJECT

The project report titled "SMART RESERVOIR TANK" has been submitted, reviewed, and verified as fulfils the conditions and requirements of the Project Writing as stipulated

Checked by:

Supervisor's name : PN ASLINDA BINTI ZAMAH SHARI

Supervisor's signature:

Date :

Verified by:

Project Coordinator name :

Signature of Coordinator :

Date :

"I acknowledge this work is my own work except for parts that have already been explained to our source"

Signature :

Name : **NURUL NAJIHAH BINTI AZIZ**

Registration Number : **08DEP20F1028**

Date :

DECLARATION OF ORIGINALITY AND OWNERSHIP

TITLE : SMART RESERVOIR TANK

SESSION: SESI 2 2022/2023

1. I, **NURUL NAJIHAH BINTI AZIZ (08DEP20F1028)**

is a final year student of **Diploma in Electrical Engineering (Communication), Department of Electrical, Politeknik Sultan Salahuddin Abdul Aziz Shah,** which is located at **Persiaran Usahawan, 40150 Shah Alam, Selangor. (Hereafter referred to as 'the Polytechnic').**

2. I acknowledge that 'The Project above' and the intellectual property therein are the result of our original creation /creations without taking or impersonating any intellectual property from the other parties.

3. I agree to release the 'Project' intellectual property to 'The Polytechnics' to meet the requirements for awarding the **Diploma in Electrical Engineering (Communication)** to me.

Made and in truth that is recognized by:	Signature:
NURUL NAJIHAH BINTI AZIZ (Identification card No: - 021201-03-0082)	..... (**NURUL NAJIHAH BINTI AZIZ**)
In front of me, **PN ASLINDA BINTI ZAMAH SHARI** (Identification card No: -) As a project supervisor, on the date:	..... **(PN ASLINDA BINTI ZAMAH SHARI)**

ACKNOWLEDGEMENTS

I devote time to this endeavour. However, it would not have been possible without the kind support and help of many people and organisations. I want to sincerely thank each and every one of you for your help. I owe a lot to Puan Aslinda Binti Zamah Shari, who served as my supervisor, as well as to my instructors, friends, and coworkers for their guidance, on-going supervision, and providing of the knowledge I required to complete the project.

In order to complete this project successfully, I would want to sincerely thank my parents, relatives, and other close relatives for their kind encouragement and support. I want to thank the business community from the bottom of my heart for their attention and time.

My project-development partner and the others who gave their expertise to help me are equally deserving of my gratitude and respect.

TABLE OF CONTENTS

CONFIRMATION OF THE PROJECT ..... **i**

DECLARATION OF ORIGINALITY AND OWNERSHIP ..... **ii**

ACKNOWLEDGEMENTS ..... **iii**

TABLE OF CONTENTS ..... **iv**

LIST OF FIGURES ..... **vi**

ABSTRACT **1**

CHAPTER 1: INTRODUCTION

1.1 Background of Project **2**

1.2 Problem Statement **2**

1.3 Objectives of the Project **2**

1.4 Scope of the Project **2**

1.5 Significance of the Project **3**

1.6 Summary **3**

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction **4**

2.2 Water Supply Disruption **4**

2.3 Potential Loss of Billions Of Ringgit Due To Water Disruption **5**

2.4 Microcontroller Based Automated Water Level Sensing and Controlling: Design and Implementation Issue **5**

2.5 An IOT based Water Supply Monitoring and Controlling System with Theft Identification **6**

2.6 An Internet of Things Based Model for Smart Water Distribution with Quality Monitoring **6**

2.7 Microcontroller Based Water Tank Management Monitoring System **6**

2.8 IOT Based Real time water Monitoring System for Smart City **7**

2.9 IOT Controlled Water Supply Management **7**

2.10 Summary **8**

CHAPTER 3: METHODOLOGY

3.1 Introduction **9**

3.2 Block Diagram **9**

3.3 Flowchart of Device Implementation 10

3.4 Schematic Diagram 11

3.5 Elements of the Components 12

3.5.1 Hardware 12

3.5.2 Software 14

3.6 Cost 14

3.7 Summary 15

CHAPTER 4: RESULT AND ANALYSIS

4.1 Introduction 16

4.2 Data Analysis 16

4.3 Result of Smart Reservoir Tank 16

4.3.1 Hardware Device 16

4.3.1 Software 19

4.4. Data Analysis for the Project Survey 20

4.5 Summary 22

CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1 Conclusion 23

REFERENCES 24

APPENDIX 26

APPENDIX A - SEMESTER 4 PROJECT 1 GANTT CHART 27

APPENDIX B - SEMESTER 5 PROJECT 2 GANTT CHART 28

APPENDIX C – PROJECT SURVEY 27

APPENDIX D – SOURCE CODE 30

LIST OF FIGURES

Figure 1: Number of unscheduled water cuts in Malaysia (2017) **4**

Figure 3.1: Block Diagram of the Project **9**

Figure 3.2: Flowchart of the Project **10**

Figure 3.3: Schematic Diagram of the project **11**

Figure 3.4: Node MCU ESP8266 ..... **12**

Figure 3.5: Ultrasonic Sensor **12**

Figure 3.6: Relay Module **13**

Figure 3.7: Solenoid Valve **13**

Figure 3.8: Arduino IDE **14**

Figure 3.9: Cost of Project **14**

Figure 4.1: View of Hardware (Upside View) **16**

Figure 4.2: View of Hardware (Side View) **17**

Figure 4.3: View of Solenoid Valve (Inlet) **17**

Figure 4.4: View of Solenoid Valve (Outlet) **17**

Figure 4.5: View of Ultrasonic Sensor **18**

Figure 4.6: View of Wiring Inside The Casing **18**

Figure 4.7: Graph Time Water Flow Through Valve **19**

Figure 4.8: Output Display On The Application **19**

Figure 4.9: Analysis of Question 1 ..... **20**

Figure 4.10: Analysis of Question 2 ..... **20**

Figure 4.11: Analysis of Question 3 ..... **21**

Figure 4.12: Analysis of Question 4 ..... **21**

ABSTRACT

The most valuable and precious resource is water because it is a necessity for all living things. If preparations are not made ahead of time, there won't be enough water supply. As shrewd consumers, we must take all necessary precautions before something like a disruption in the water supply occurs and has an impact on our everyday lives. In this project, we create a reservoir tank system that can be controlled by apps using the Node MCU ESP8266. We can activate the system using the built-in apps when the government issues a warning about water supply interruptions when no one is home, and the water ultrasonic sensor will be used to send a signal to the Node MCU ESP8266. We will then be alerted when the tank is full so we can deactivate the system using the same apps.

CHAPTER 1: INTRODUCTION

1.1 Background of Project

Malaysia was regarded as a populous, developing nation. As a result, there are more people using water, which has led to problems with the water supply in some areas. Water supply interruptions were frequently reported by Sinar Harian on January 28, 2022, and inhabitants did not have enough time to prepare. Residents were having trouble adjusting because they had to get up early to prepare. Our current reservoir tank can only be operated manually, therefore we chose to develop it.

Reservoir tank are defined as a storage space for fluids, reservoirs may hold water or gasses, including hydrocarbons. In this project, we specifically use reservoir tank to store water supply when there's a water supply disruption. Even there is no one at home to prepare before disruptions, they can still complete all the preparations easily and quickly.

1.2 Problem Statement

Malaysia was regarded as a populous, developing nation. As a result, there are more people using water, which has led to problems with the water supply in some areas. Water supply interruptions were frequently reported by Sinar Harian on January 28, 2022, and inhabitants did not have enough time to prepare. Due to an interruption in the water supply at Taman Bujang Utama, Merbok, Kedah, one of the residents had to get up as early as 5.30 am to collect water. From the aforementioned instance, we know that homeowners who suffered a disruption in their water supply must get up early in order to prepare, which is very inconvenient.

1.3 Objectives of the Project

This project was performed to develop a home reservoir system by using Node MCU ESP8266. Our aim is to help people making their preparation for the absence of water when there's no people at home. This system can be activated with a mobile application using Wifi which can save user time.

1.4 Scope of the Project

Our project scope is for people who happen to not be at home when water supply disruptions occur and causing they don't have any water supply to use. With this system, we can help people make preparations even when they were at work or somewhere else such as hometown.

1.5 Significance of the Project

Significance of our project is to help people that having difficulty in making preparations for water supply disruption while they were at work or somewhere else. This system can help them preparing for water supply disruption quickly and easily.

1.6 Summary

In the epilogue, chapter one of this study is briefly explained. The backdrop of the study is presented in the first section of this chapter, which goes on to discuss the research project's water supply disruption and potential loss of billions of Ringgit due to water disruption. The project's objectives, scope, and problem statements are listed afterwards. This chapter also ends with a discussion of the project's significance.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This project-related articles and research papers published by academic publishers will be analysed in the chapter on the literature review. This literature review is divided into the theory and prior studies sections. Then, in earlier studies, the focus was on products and devices that had already been created from other studies and might be used to advance this endeavour.

2.2 Water Supply Disruption

In Malaysia, water supply interruptions can either be planned (often for maintenance and facility upgrades) or unplanned. Selangor continuously experienced the most unscheduled water supply interruptions from 2014 to 2017, according to the most recent data from the Ministry of Water, Land, and Natural Resources (KATS).

According to KATS data, Selangor tops the list for unannounced water cuts, with 84,796 of 174,997 national occurrences in 2014 and 81,969 of 167,055 events countrywide in 2015; however, this number has significantly decreased to 19,061 out of 61,517 incidents in 2017.

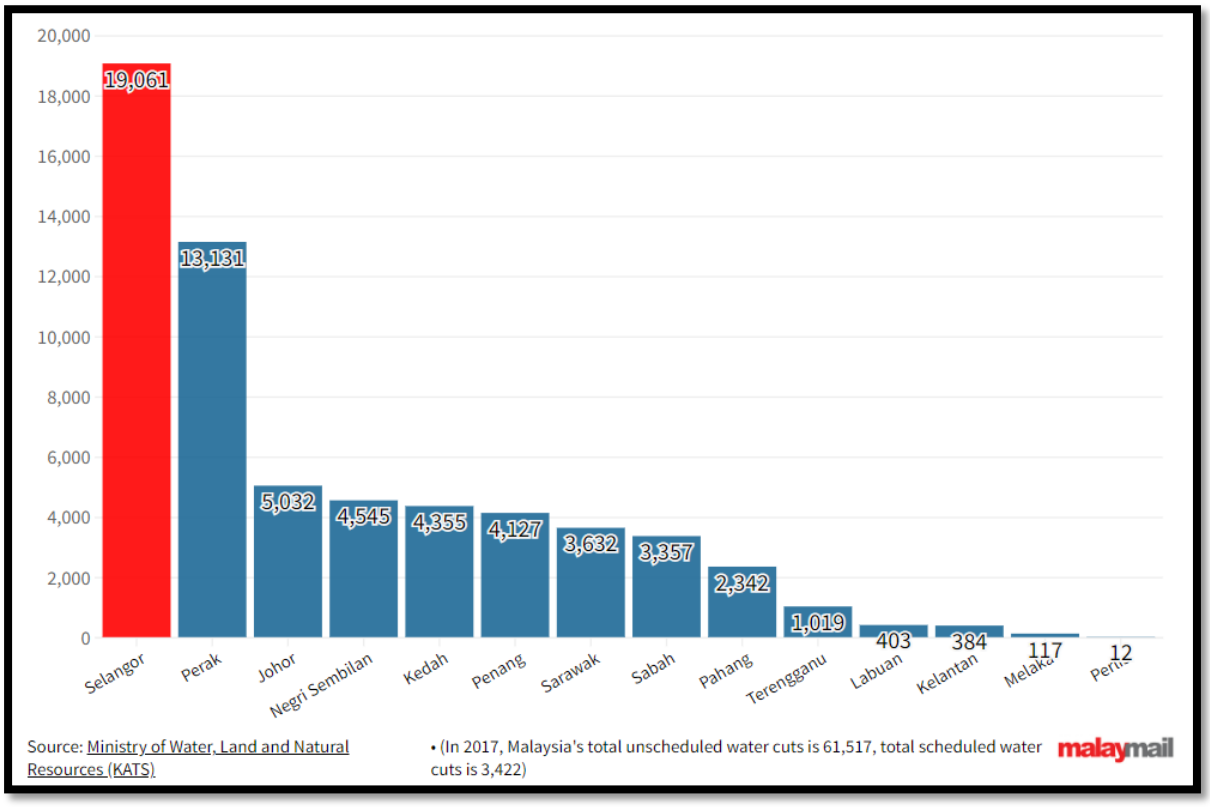


Figure 1: Number of unscheduled water cuts in Malaysia (2017)

2.3 Potential Loss Of Billions Of Ringgit Due To Water Disruption

Many major industries such as manufacturing, tourism, food and healthcare will suffer losses as operations cannot run normally, thus affecting the daily business target set. Failure to achieve the target, especially for manufacturers, will lead them to increase costs due to overtime operations to cover the affected production.

According to estimates, if the problem persists for a month, economic losses might total more than RM2 billion as a result of decreased production and effects on other industries. Manufacturing production has been negatively impacted by water rationing and unplanned outages during the past few years, resulting in the loss of millions of Ringgit due to the reduction in production.

Despite the nation receiving enough of rainfall each year, water shortages remain a common occurrence. In Selangor, a problem that has been dragging on for years, the state had not been able to sign a deal to buy concessionaires' water assets.

Water became a valuable resource due to the rapid industrial and residential expansion, particularly in areas like Selangor. The need to balance water demand with human demands, though, received little attention from the authorities.

2.4 Microcontroller Based Automated Water Level Sensing and Controlling: Design and Implementation Issue

S. M. Khaled Reza, Shah Ahsanuzzaman Md. Tariq, S.M. Mohsin Reza (2010) developed system that used micro-controller based water level sensing and controlling in a wired and wireless environment A water level management strategy would aid in lowering both household electricity usage and water overflow. An 8-bit microcontroller, an inverter, a reserve tank (res. Tank), a water tank, and a water pump were used in this design. Water level sensor has been used to operate the water pump.

The water level is determined using four homemade water level sensors. Microcontroller input used to be made up of inverted sensor data. We programmed the PIC 16F84A memory using MPLAB software. The use of a microcontroller to automatically operate the entire system lowers the complexity of the design and control. The sensor unit, which monitors the water level using an inverter, provides input to the microcontroller. After analysing the input variables, the output determines whether to turn on or off the water pump based on the tank's current water level.

2.5 An IOT based Water Supply Monitoring and Controlling System with Theft Identification

Mrs. M. S. Joshi and Pranita Vijaykumar Kulkarni devised and created an IOT-based water supply monitoring and regulating system with theft identification in 2016. The system was a low-cost, dependable, and effective way to distribute water properly while also being continuously monitored and controlled from a central computer to address water-related issues. The proposed system consists of a Raspberry Pi acting as a minicomputer and a variety of sensors, including flow, turbidity, and water level sensors. Raspberry Pi receives data from sensors that are collected by Arduino. This system handles issues with overflow, excessive use, water quality, and distribution in the right way. This solution enables continuous monitoring and management from a central server.

2.6 An Internet of Things Based Model for Smart Water Distribution with Quality Monitoring

After learning how water is historically distributed in a certain zone, Joy Shah (2017) developed the smart IOT-based model for smart water distribution with quality monitoring. Using a web interface, the control valve-equipped embedded device can be turned on or off. Each endpoint can be managed and regulated by the suggested IoT device so that it can get enough water under pressure. The Flow sensor uses the Controller to transmit data to the cloud over the internet. Control valves can be turned on and off to distribute water under pressure while the generated data is being watched in real time. A water flow sensor is used to calculate the flow rate. The quality of the water is monitored using a variety of sensors, including pH sensors, conductivity sensors, and temperature sensors. The raspberry PI controller periodically uploads the sensor's values to the cloud over an Internet channel. These values can be tracked in real time, location-wise. The source of the problem can be found if the supplied water is fresh and has passed all tests but the water that reaches the end points is contaminated. For example, if the supplied water is fresh but the water that reaches the end points is contaminated, we can determine the region or end point where the water is fresh and uncontaminated because that means there is no problem up until that point.

2.7 Microcontroller Based Water Tank Mangement Monitoring System

This water quality and quantity monitoring system's primary goal was to provide an early warning system for domestic water tanks. The main controller for the entire system is an Atmega328P microprocessor. The Ph probe and ultrasonic sensor are employed as the system's inputs. The primary controller will receive every signal input from an ultrasonic sensor and a pH probe. The microcontroller will analyse the input signal and decide what will happen as an output. This system outputs a buzzer, an LCD 16 x 2 character display, and LEDs. The remaining water level and water quality in the tank will be continuously displayed on the LCD. In order to distinguish between the three water level stages of high level, medium level, and low level, this project uses three LEDs as indicators. When the water level is the lowest, a buzzer that serves as an alarm will sound. The consumers will be made aware of the need to plan ahead. In general, this project will ensure that the water level in the reserve tank can be continuously checked for safety.

2.8 IOT Based Real time water Monitoring System for Smart City

They proposed a method that is based on IoT to meet the needs for water and monitoring water quality (Internet of Things). Sensors will be used to gather the data, which can then be viewed through a website in real time. The suggested system includes a raspberry PI 0W as the central controller and various sensors, including a water flow sensor, a pH sensor, a water control valve, and a water level sensor. A microcontroller processes the data it receives from the sensors and then uses a wireless communication module to send it to the cloud.

Our approach overcomes the problems by taking into account the limitations of earlier water management models and using the cloud to store sensor data. This will guarantee data security and stop data loss. Their design has a water level sensor that provides real-time data on the water level in the tank. Supply of safe water will be ensured via quality control. Data is shown on the website, therefore using the website rather than the application ensures faster access to the data from any location. In order to transmit data via the internet, the core controller accesses the sensor values and processes them. The core controller is a Raspberry PI. Utilizing cloud computing, the sensor data may be seen on the website.

2.9 IOT Controlled Water Supply Management

Pavithra Kumari, Chaithra R. M, Ranjitha, Kiran A.R (2018) designed IOT controlled water supply management. The technology employs a switch called a raindrop sensor module to detect water leaks in pipes. The water level in the tank is checked using a float sensor, which also guards against the pump running dry. The use of temperature sensors and voltage sensors protects DC motors from overheating and overvoltage. Through the WiFi module, all system issues will be communicated to the relevant party. Water flow is controlled by solenoid valves, which are electromagnetically actuated valves. The 12V DC tiny pump utilised in this instance is managed by a microcontroller and moves water from the tank to the needed solenoid valves.

The Wi-Fi module selects the data flowing over the internet. An arduino board can connect to the internet using Wi-Fi with the help of the Arduino Wi-Fi shield. Through an Arduino controller, sensors are controlled and sensed. The switching component is made up of relays. Relay is an electromechanical switch that operates on the energised electromagnet concept. Water flow is controlled by solenoid valves, which are electromagnetically actuated valves. The microcontroller-controlled, 12-volt tiny pump utilised in this instance lifts water from a tank and pushes it to the necessary solenoid valves. a system of direct current power supplies that maintains constant voltage regardless of changes in the primary supply.

With the aid of a high voltage detection circuit, it is feasible to protect dc motors from high voltage. Water loss could be brought on by pressure variations and certain main pipe damage. In order to keep the water level in a tank within a safe range, this is developed using a straightforward technique of wired conducting probes with transistor interface. If the water level drops below the set limits, the transistor enters a saturated condition and sends a signal to the Arduino. The entire procedure is performed using automated equipment.

2.10 Summary

At the start of chapter two, there is a synopsis of the subjects covered in that chapter or an introduction to those subjects. This chapter addresses all information relevant to the management of creation water and is primarily composed of research-based content. The study led to the term "Smart Reservoir Tank" being used to describe all methods for estimating how to manage a water supply.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This section examines the many research methods that are frequently used by researchers that study hardware and software systems. This task requires a number of different approaches to be completed. Data analysis, gadget development, testing, and research are some of the methods used in the study. The methods used to carry out this study will be covered in more detail in this section, which will also deal with the methodological research.

The structure and design of the product will also be covered in this section, along with the techniques used to provide readers with informational viewpoints that will deepen their knowledge. Because it is vital to the execution of this study, this component of the strategy is significant in this investigation. This section provides a detailed explanation of the technique.

3.2 Block Diagram

A block diagram of the hardware architecture for the Smart Reservoir Tank process is shown in Figure 3.1. The system's components are depicted in this block diagram and are grouped into three primary categories: inputs, processes, and outputs. Together, these elements constitute the management of the water supply. The block diagram in Figure 3.1 demonstrates how each component is connected to the other components and how each one carries out a certain task. This project will run on electricity from the adapter. When we switch it on, the ultrasonic sensor will detect the distance and volume of liquid in the tank so that it may be displayed in the application. At the same time, the tank will be connected to the app. The ultrasonic sensor will then send a signal to the Node MCU (ESP8266) in the process section with the assistance of a relay module. The tank's volume will be measured and shown on the mobile application for the final step.

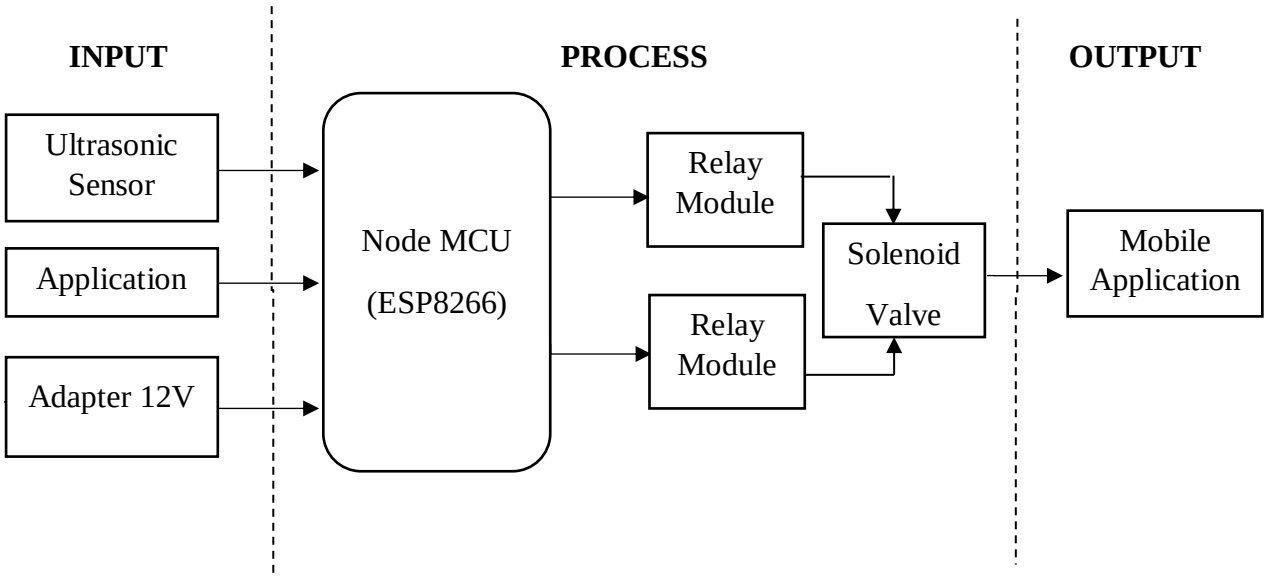


Figure 3.1: Block Diagram of the project

3.3 Flowchart of Device Implementation

We can observe from this flowchart that the project software's source of input data is Node MCU ESP8266. The water level sensor will then determine whether the tank is full of water or not. The valve will switch off if the tank is full and stay on if the tank is not full.

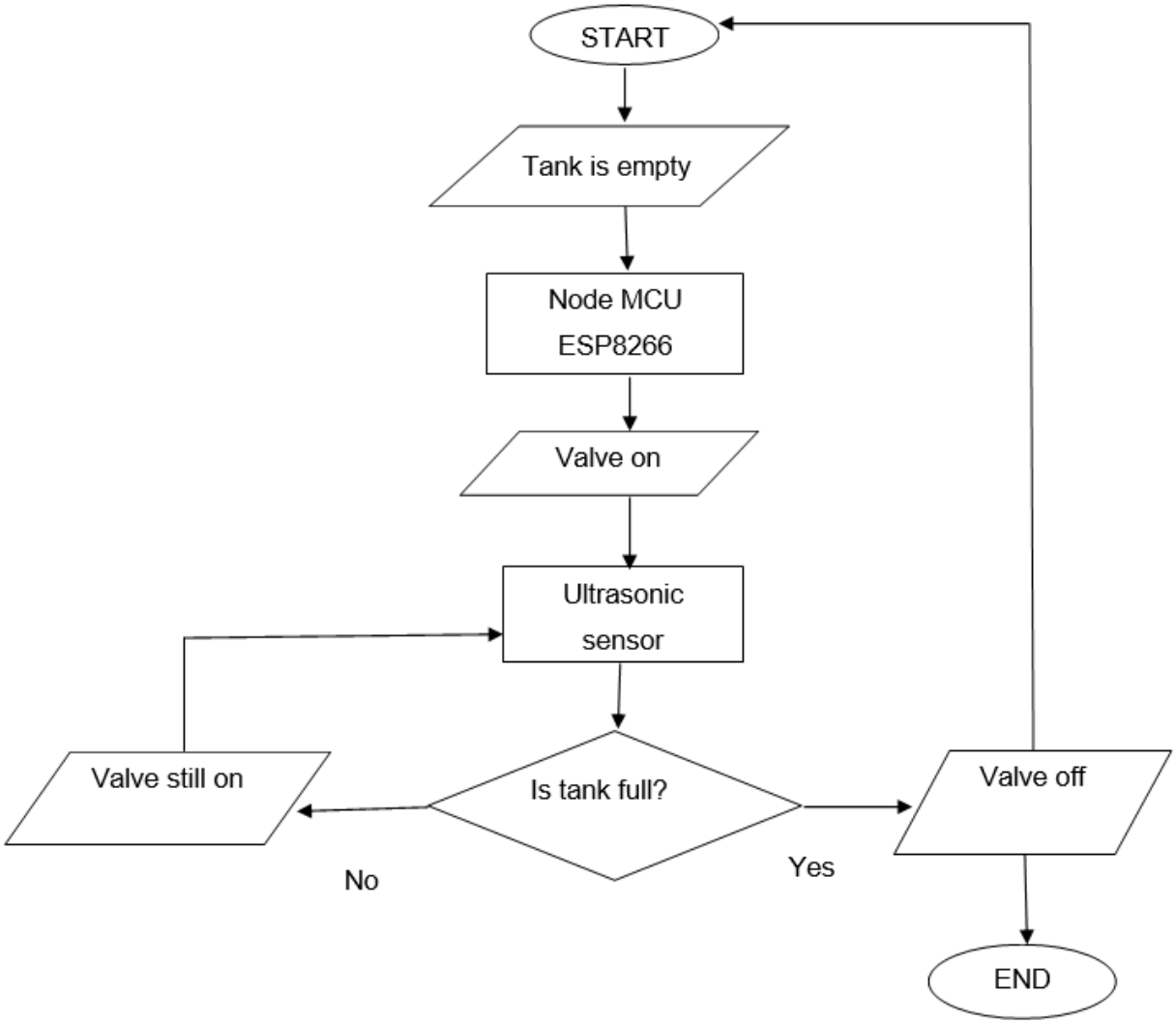


Figure 3.2: Flowchart of the Project

3.4 Schematic Diagram

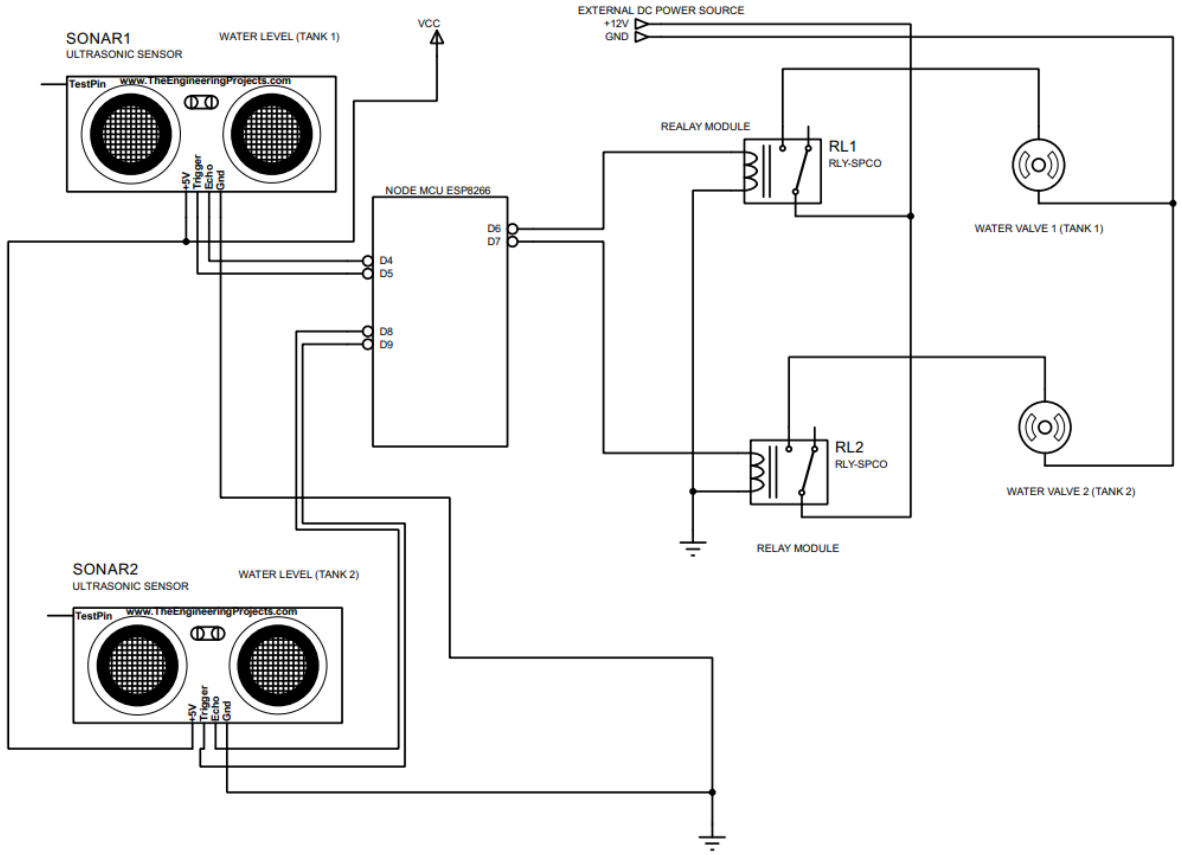


Figure 3.3: Schematic Diagram of the project

3.5 Elements of the Components

3.5.1 Hardware

1. **NODE MCU ESP8266**

Figure 3.4: NODE MCU ESP8266

A cheap open source IoT platform is Node MCU. It originally included hardware based on the ESP-12 module and firmware that runs on Espressif Systems' ESP8266 Wi-Fi SoC. Support for the 32-bit ESP32 MCU was later added. There are open source prototyping board designs for the Node MCU open source firmware. Node and MCU are combined to form the moniker "Node MCU" (micro-controller unit). To control the input and output of the project. It also can be connected using Wi-Fi.

2. **ULTRASONIC SENSOR**

Figure 3.5: ULTRASONIC SENSOR

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound. Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver.

3. RELAY MODULE

Figure 3.6: ULTRASONIC SENSOR

A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit. It is use to strengthen the power to the solenoid valve.

4. SOLENOID VALVE

Figure 3.7: SOLENOID VALVE

The definition of a solenoid valve is an electro-mechanical valve that is commonly employed to control the flow of liquid or gas. There are various solenoid valve types, but the main variants are either pilot operated or direct acting. In this project, it is use to control either opening or closing the valve so that the water can be flow or prevent from going through it

3.5.2 Software

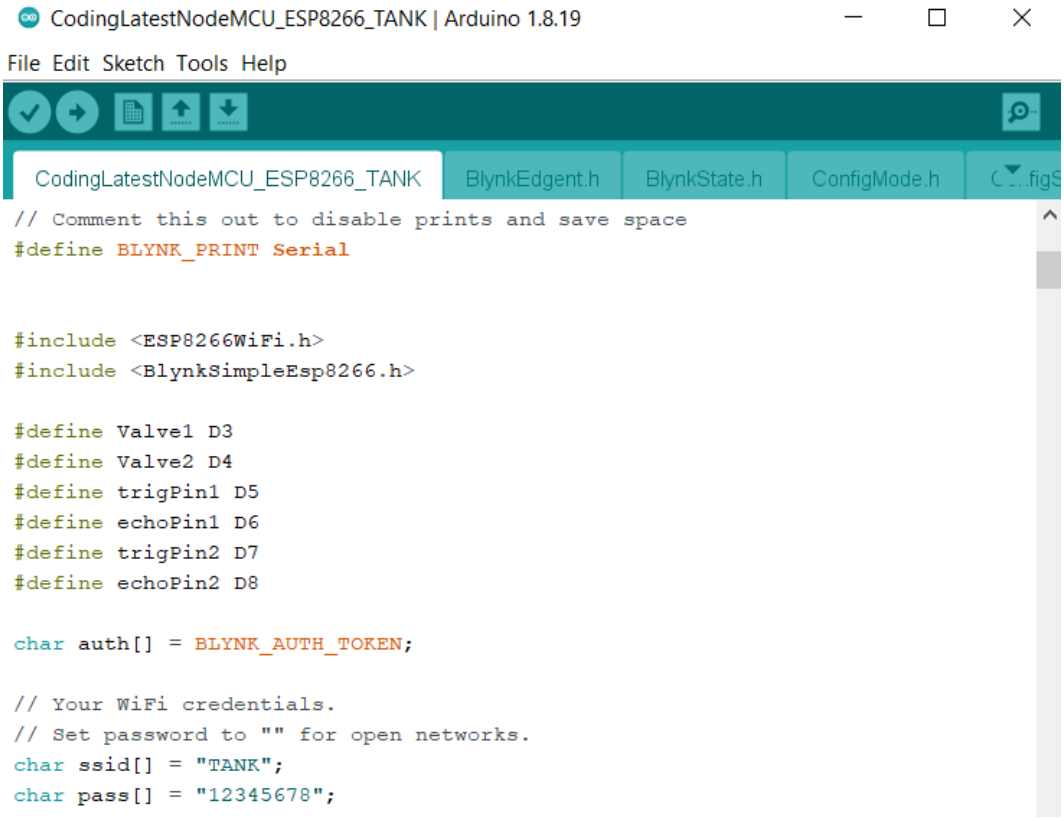


Figure 3.8: Arduino IDE

3.6 Cost of the Project

	Price (RM)
Component	230
Storage Box (Tank)	40
Rubber Pipe	25
Multi-Core Cable	32
Heat Shrink Tube	13
Adapter	10

Figure 3.9: Cost of project

3.7 Summary

This chapter largely covers the project development technique. The project's initial concept, which consists of the project's main elements or ideas, is where this chapter starts. A block diagram of the entire project is then displayed for better understanding, showing all of its inputs, processes, and outputs. This chapter explicitly lays out the hardware and software development procedures for each of these developments, together with the cost estimations that were used, to create the smart reservoir tank. The predicted progress of the entire project that will be used in the project is described in the chapter's last section.

CHAPTER 4: RESULT AND ANALYSIS

4.1 Introduction

This chapter discusses the analysis done for the results and findings of this research. When the interruption in the water supply was resolved, the smart reservoir tank was tested. The outcomes, as well as the analysis process, are fully described in this chapter. The analysis's findings are vital in helping us comprehend the efficacy in terms of accuracy and efficiency in terms of time, money, and energy savings.

4.2 Data Analysis

The device must be tested for usage after it has been entirely manufactured in order to evaluate whether or not the product is accurate. Data is gathered on how long it takes the smart reservoir tank to fill to a particular level. Because it was placed above the reservoir within a specified time that the user had chosen, the ultrasonic sensor detects the volume of water in the tank when it flows through the solenoid valve and compares it to the depth of the reservoir.

4.3 Result of Smart Reservoir Tank

4.3.1 Hardware Device

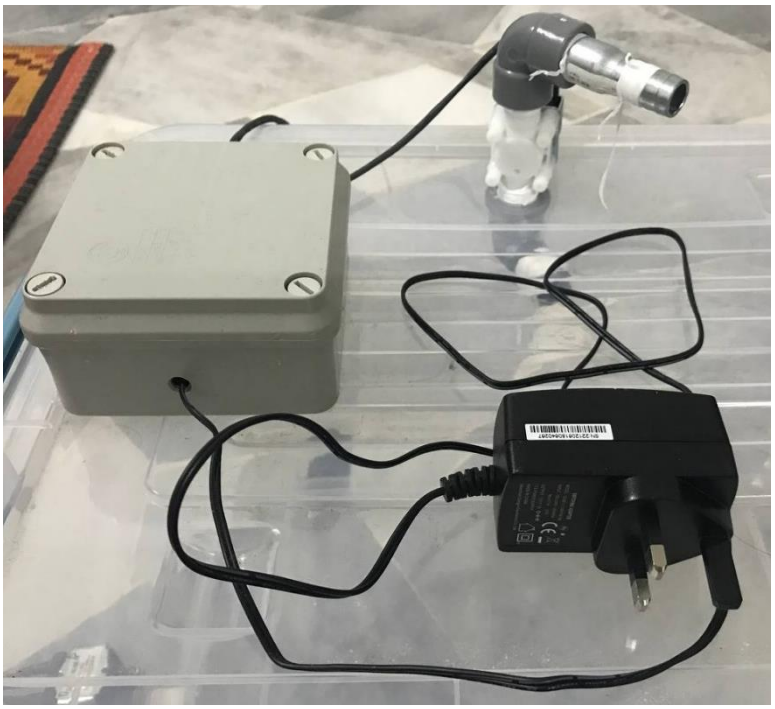


Figure 4.1: View of Hardware (Upside View)

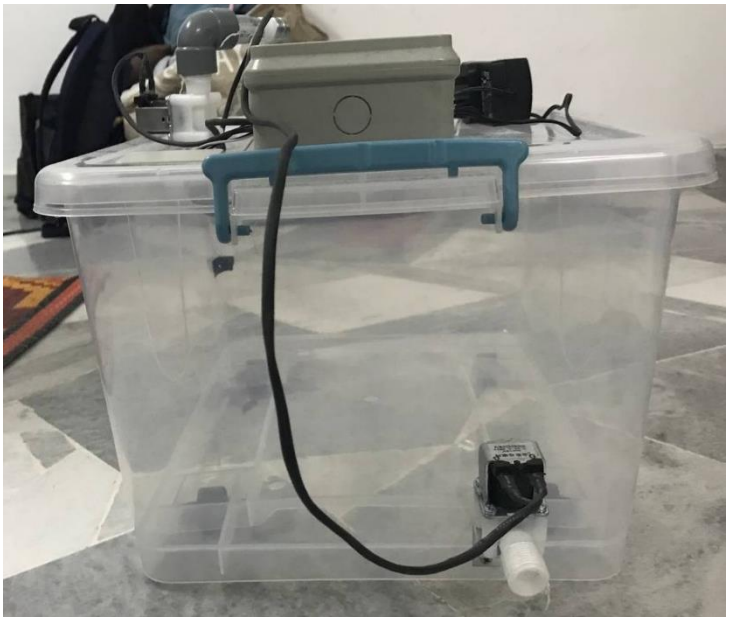


Figure 4.2: View of Hardware (Side View)

Figure 4.3: View of Solenoid Valve (Inlet)

Figure 4.4: View of Solenoid Valve (Outlet)

Figure 4.5: View of Ultrasonic Sensor

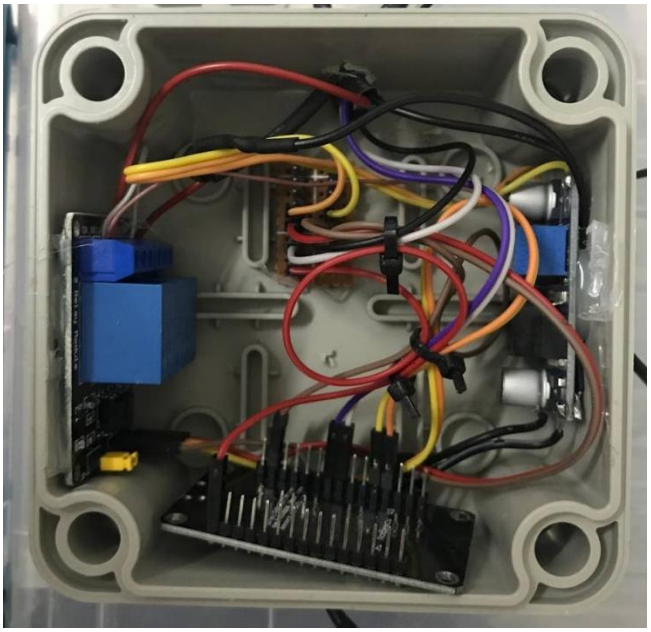


Figure 4.6: View of Wiring Inside The Casing

4.3.2 Software

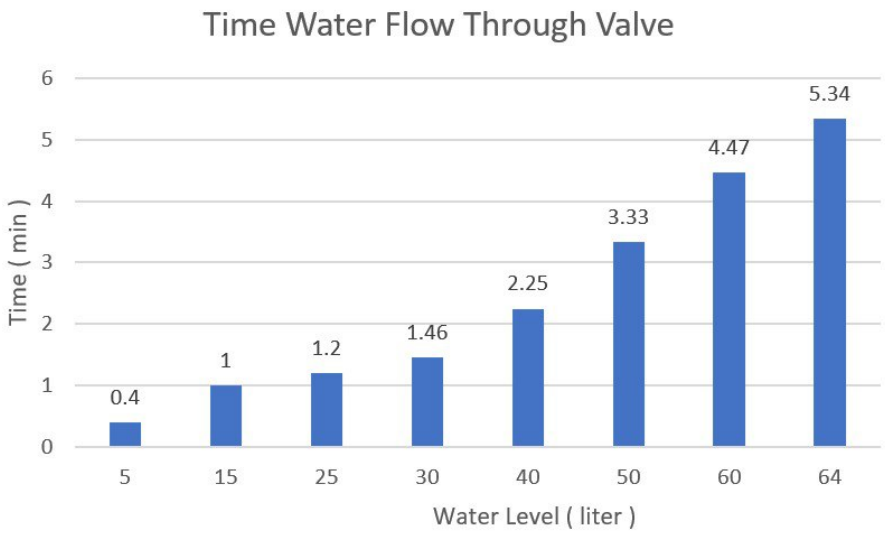


Figure 4.7: Graph Time Water Flow Through Valve

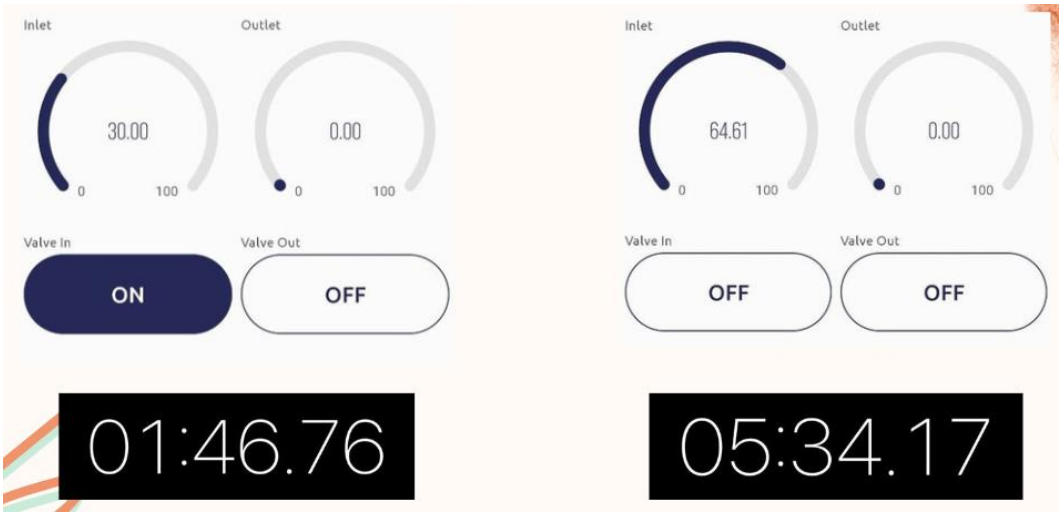


Figure 4.8: Output Display on the Application

4.4 Data Analysis for the Project Survey

We've just looked at one aspect of this survey's findings so far the disruption of the water supply. Since we want to affect the entire nation, not just one state, our target respondents for a water supply interruption are individuals who own a home and don't care where they are from. For this survey, we collected 17 responses from different states.

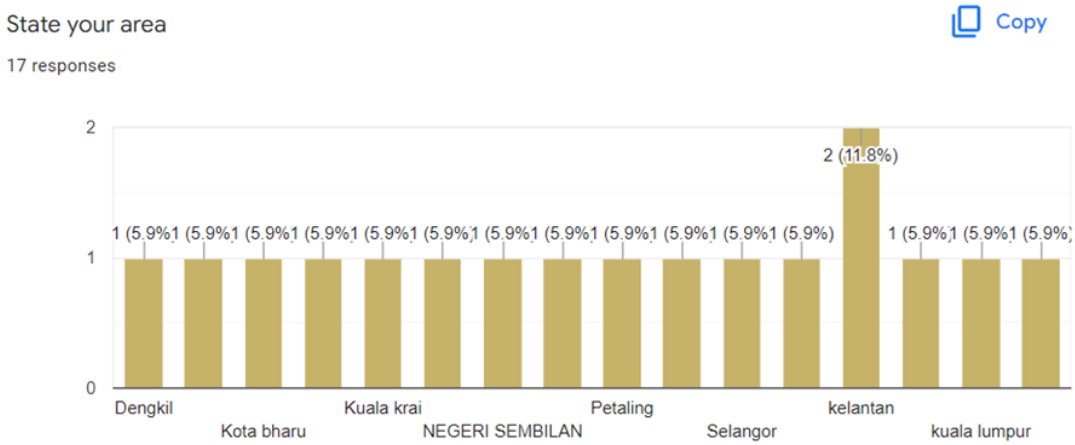


Figure 4.9: Analysis of Question 1

According to the bar graph in Figure 4.5, out of the 17 respondents, 4 of them are from Kelantan. Other 13 respondents come from various state that experienced the water supply disruption

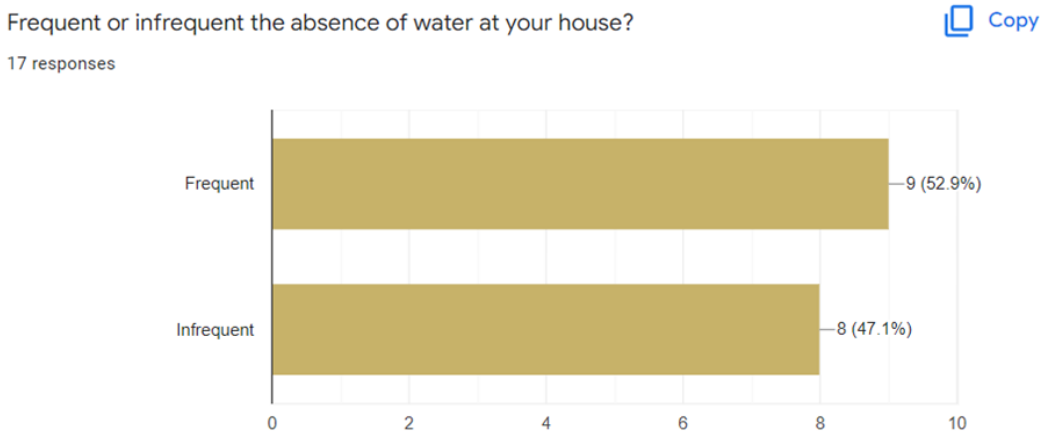


Figure 4.10: Analysis of Question 2

The bar graph in Figure 4.6 shows that, 9 out 0f 17 respondent frequently experienced the water supply disruption

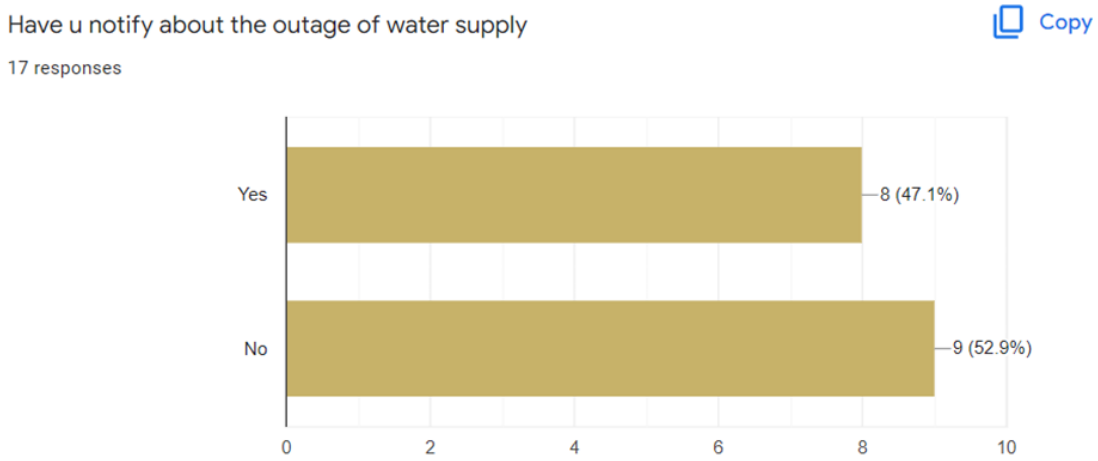


Figure 4.11: Analysis of Question 3

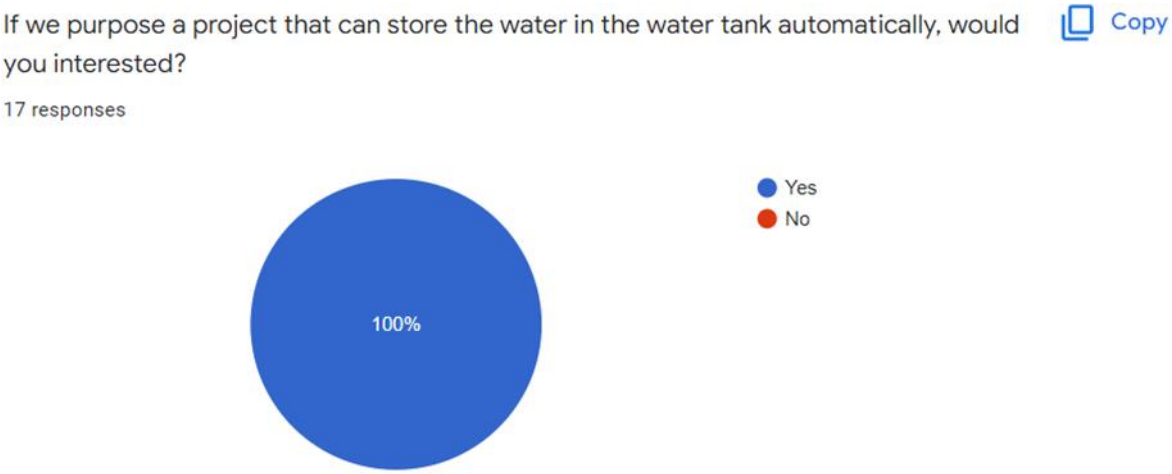


Figure 4.12: Analysis of Question 4

4.5 Summary

In conclusion, it can be noted that all four survey questions received positive and pleasing responses from all participants. They concurred that this device is incredibly advantageous for individuals who frequently experienced water supply disruption and always don’t have time to make a preparation at home. This system is activated by an application that causes the solenoid valve to open, allowing water to flow into the reservoir tank. The ultrasonic sensor will then send a notification to the application when the tank is full so that we may turn off the system. Whether or not the tank is filled with water will be determined by the ultrasonic sensor. The solenoid valve will be activated by the Node MCU to cease filling the tank with water after this sensor is linked to it. People will receive notifications from the application via apps the home's owner has downloaded to their phones.

CHAPTER 5: CONCLUSION

5.1 Conclusion

There are many different reservoir tanks available in today's technologically advanced world, but the Smart Reservoir Tank is designed specifically for users who were preoccupied with their outside lives and lacked the time to prepare for the disruption of the water supply. After that, the programme that allows online tank monitoring will be used. Additionally, this system is activated by a software that opens the solenoid valve and lets water flow into the reservoir tank. The ultrasonic sensor will then alert the application when the tank is full, enabling us to turn the gadget off. Whether or not the tank is filled with water will be determined by the ultrasonic sensor. When this sensor is linked to the Node MCU, the Node MCU will activate the solenoid valve to stop the tank from filling with water. The home's owner will download apps to mobile phones that will enable users to get programme notifications.

REFERENCES

[1] S. M. Khaled Reza, Shah Ahsanuzzaman Md. Tariq, S.M. Mohsin Reza,2016, Microcontroller Based Automated Water Level Sensing and Controlling : Design and Implementation issue .

[2] Pranita Vijaykumar Kulkarni1 , Mrs. M. S. Joshi2, 2016 ,An IOT Based Water Supply Monitoring and Controlling System with Theft Identification .

[3] Joy Shah1, 2017 , An Internet of Things Based Model for smart Water Distribution with Quality Monitoring .

[4] Siti Aishah Che Kar, Nur Idawati Md Enzai, Ahmad Faidhi Mohd Hatim, Aqib Fawwaz Mohd Amidon, Ahmad Faris Firdaus Mohd Sapawi , 2017 , Microcontroller Based Water Tank Management Monitoring System .

[5] Rupalir. Shevale , Shweta Karad, Maryam Merchant, Ashwini Kardile, Vijeyata Mishra , 2018 , IOT Bsed Real time water Monitoring System for Smart City .

[6] Pavithra kumari, Chaithra R. M, Ranjitha, Kiran A.R , 2018 , IOT Controlled Water Supply Management

[7] Siti Zubaidah Zakaraya, 2022, Water supply is often cut off, residents do not have time to prepare. Retrieved on 28th January 2022 from https://www.sinarharian.com.my/article/185246/EDISI/Bekalan-air-sering-putus-penduduk-tak-sempat-buat-persiapan

[8] Jan, F., Min-Allah, N., Saeed, S., Iqbal, S. Z., & Ahmed, R. (2022, January 20). IOT-based solutions to monitor water level, leakage, and motor control for Smart Water Tanks. MDPI. Retrieved December 20, 2022, from https://www.mdpi.com/2073-4441/14/3/309

[9] Lim, I. (2019, August 6). Klang Valley Water Disruption: Why does it happen so often? Malay Mail. Retrieved December 20, 2022, from https://www.malaymail.com/news/malaysia/2019/08/06/klang-valley-water-disruption-why-does-it-happen-so-often/1777926

[10] Water disruption may lead to billion ringgit losses. (n.d.). Retrieved December 20, 2022, from https://www.nst.com.my/news/nation/2018/03/343104/water-disruption-may-lead-billion-ringgit-losses

[11] Manufacturers cry over millions lost in water shortage. (n.d.). Retrieved December 20, 2022, from https://www.fmm.org.my/FMM_In_The_News-@-Manufacturers_Cry_Over_Millions_Lost_in_Water_Shortage.aspx

[12] Wikimedia Foundation. (2022, November 13). Nodemcu. Wikipedia. Retrieved December 20, 2022, from https://en.wikipedia.org/wiki/NodeMCU

[13] G@dmin. (2020, December 16). What is a power relay module: Relay Modules. GEP Power Products. Retrieved December 20, 2022, from https://www.geppowerproducts.com/standard-products/power-distribution-fuse-relay-holders-fuse-blocks/relay-modules/#:~:text=Relay%20Modules,-

Request%20a%20Quote&text=A%20power%20relay%20module%20is,or%20close%20an%20electrical%20circuit.

[14] Solenoid valve - how they work. Tameson. (n.d.). Retrieved December 20, 2022, from https://tameson.com/solenoid-valve-types.html

[15] Jost, D. (2019, October 7). What is an ultrasonic sensor? Fierce Electronics. Retrieved December 20, 2022, from https://www.fierceelectronics.com/sensors/what-ultrasonic-sensor

APPENDIX

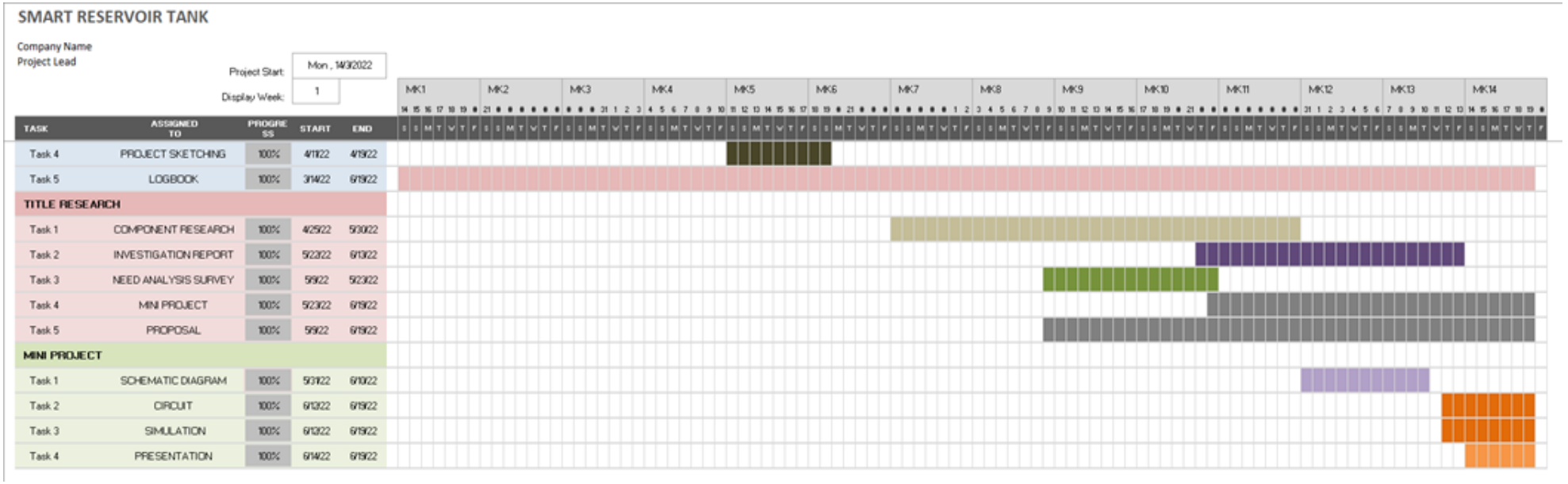
APPENDIX A - SEMESTER 4 PROJECT 1 GANTT CHART

APPENDIX B - SEMESTER 5 PROJECT 2 GANTT CHART

APPENDIX C – PROJECT SURVEY

APPENDIX D – SOURCE CODE

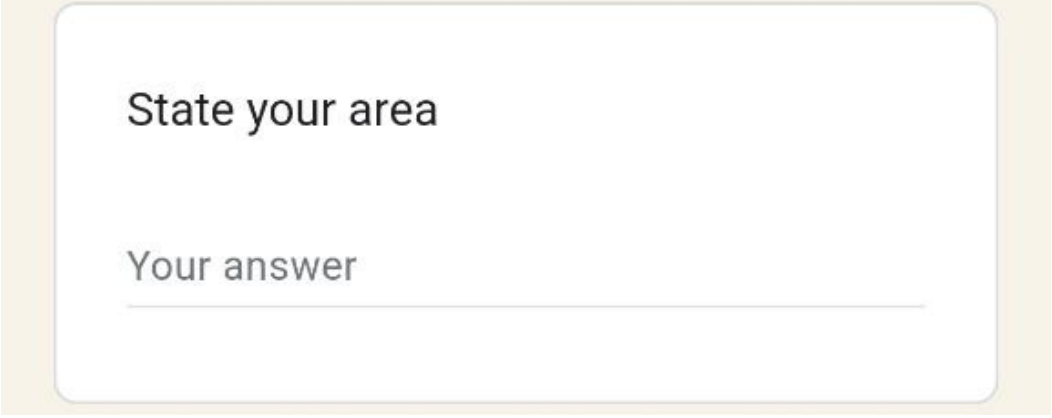
APPENDIX A - SEMESTER 4 PROJECT 1 GANTT CHART



TASK	ASSIGNED TO	PROGRESS	START	END
Task 4	PROJECT SKETCHING	100%	4/1/22	4/19/22
Task 5	LOGBOOK	100%	3/14/22	6/19/22
TITLE RESEARCH				
Task 1	COMPONENT RESEARCH	100%	4/25/22	5/30/22
Task 2	INVESTIGATION REPORT	100%	5/22/22	6/13/22
Task 3	NEED ANALYSIS SURVEY	100%	5/9/22	5/23/22
Task 4	MINI PROJECT	100%	5/23/22	6/19/22
Task 5	PROPOSAL	100%	5/9/22	6/19/22
MINI PROJECT				
Task 1	SCHEMATIC DIAGRAM	100%	5/31/22	6/10/22
Task 2	CIRCUIT	100%	6/13/22	6/19/22
Task 3	SIMULATION	100%	6/13/22	6/19/22
Task 4	PRESENTATION	100%	6/14/22	6/19/22

APPENDIX B - SEMESTER 5 PROJECT 2 GANTT CHART

APPENDIX C – PROJECT SURVEY



Frequent or infrequent the absence of water at your house?

- [ ] Frequent
- [ ] Infrequent

Have u notify about the outage of water supply

- [ ] Yes
- [ ] No

If we purpose a project that can store the water in the water tank automatically, would you interested?

- ○ Yes
- ○ No

Submit

Clear form

APPENDIX D – SOURCE CODE

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID           "TMPLCzRWk4Wx"
#define BLYNK_DEVICE_NAME           "Quickstart Device"
#define BLYNK_AUTH_TOKEN            "sUqlYfLkEkF5ZESmgi_QRUAVNcUS9qsh"


// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define Valve1 D3
#define Valve2 D4
#define trigPin1 D5
#define echoPin1 D6
#define trigPin2 D7
#define echoPin2 D8

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "TANK";
char pass[] = "12345678";

long duration1x, distance1, duration2x, distance2, duration3x, distance3, duration4x,
distance4, duration5x, distance5;
int Rly1=0, Rly2=0, Rly3=0, Rly4=0, Rly5=0, Rly6=0,  Rly7=0,  Rly8=0;
int Val1=90, Val2=0, Val3=0, Val4=0, Val5=0, Val6=0,  Val7=0,  Val8=0;
float TK1=26;
float TK2=26;
float LV1=0;
float LV2=0;
int MODE=0;

int AL1=0,AL2=0,AL3=0;
int TM=0;
BlynkTimer timer;

int pos=0;
bool led_set[2];
long timer_start_set[2] = {0xFFFF, 0xFFFF};
long timer_stop_set[2] = {0xFFFF, 0xFFFF};
```

```
unsigned char weekday_set[2];

long rtc_sec;
unsigned char day_of_week;

bool led_status[2];
bool update_blynk_status[2];
bool led_timer_on_set[2];

// This function is called every time the Virtual Pin 0 state changes


// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
{
  // Change Web Link Button message to "Congratulations!"
 // Blynk.setProperty(V3, "offImageUrl", "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
 // Blynk.setProperty(V3, "onImageUrl",  "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
 // Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-
need-to-blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{
digitalWrite(trigPin1, LOW);  // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin1, LOW);
  duration1x = pulseIn(echoPin1, HIGH);
  distance1 = ((duration1x/2) / 29.1); //* 0.26;

  digitalWrite(trigPin2, LOW);  // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin2, LOW);
  duration2x = pulseIn(echoPin2, HIGH);
  distance2 = ((duration2x/2) / 29.1); //* 0.26;
// Blynk.virtualWrite(V2, TM);
Serial.print(distance1);
Serial.print("\t");
Serial.println(distance2);

if (distance1<=TK1){
  LV1=((TK1-distance1)/26*100.0)-20;
}
```

```
if (distance2<=TK2){
  LV2=((TK2-distance2)/26*100.0)-20;
}
if (distance1>TK1){
  LV1=0;
}
if (distance2>TK2){
  LV2=0;
}

Blynk.virtualWrite(V4, LV1);
Blynk.virtualWrite(V5, LV2);

if (MODE==1){
  if (LV1<50){
    digitalWrite(Valve1,LOW);
  }
  if (LV1>=80){
    digitalWrite(Valve1,HIGH);
  }

   if (LV2<50){
    digitalWrite(Valve2,LOW);
  }
  if (LV2>=80){
    digitalWrite(Valve2,HIGH);
  }
}

}

BLYNK_WRITE(V0)
{
  Rly1 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly1==1){
 MODE=1;
  }
   if (Rly1==0){
 MODE=0;
  }

  // process received value
}

BLYNK_WRITE(V7)
{
  Rly7 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly7==1){
```

```
      digitalWrite(Valve2,LOW);                 // waits 15ms for the servo to reach the
position
  }
  if (Rly7==0){
      digitalWrite(Valve2,HIGH);                 // waits 15ms for the servo to reach the
position
  }
   // Blynk.logEvent("manual", String("MESSAGE"));


  // process received value
}


BLYNK_WRITE(V2)
{
  Rly2 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly2==1){

  delay(500);

  }

  // process received value
}

BLYNK_WRITE(V4)
{
  Rly4 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly4==1){
    digitalWrite(Valve1,LOW);
  }
  if (Rly4==0){
   digitalWrite(Valve1,HIGH);
  }

  // process received value
}

BLYNK_WRITE(V5)
{
  Rly5 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly5==1){
    digitalWrite(Valve2,LOW);
  }
  if (Rly5==0){
   digitalWrite(Valve2,HIGH);
```

```
  }

  // process received value
}

BLYNK_WRITE(V6)
{
  Rly6 = param.asInt(); // assigning incoming value from pin V1 to a variable

  if (Rly6==1){
    digitalWrite(Valve1,LOW);
  }
  if (Rly6==0){
    digitalWrite(Valve1,HIGH);
  }

  // process received value
}



BLYNK_WRITE(V1)
{
  Val1 = param.asInt(); // assigning incoming value from pin V1 to a variable


    Serial.print("");
    Serial.println(Val1);

  // process received value
}

BLYNK_WRITE(V9)
{
  unsigned char week_day;

  TimeInputParam t(param);

  if (t.hasStartTime() && t.hasStopTime() )
  {
    timer_start_set[0] = (t.getStartHour() * 60 * 60) + (t.getStartMinute() * 60) +
t.getStartSecond();
    timer_stop_set[0] = (t.getStopHour() * 60 * 60) + (t.getStopMinute() * 60) +
t.getStopSecond();

    Serial.println(String("Start Time: ") +
                   t.getStartHour() + ":" +
                   t.getStartMinute() + ":" +
                   t.getStartSecond());
```

```
    Serial.println(String("Stop Time: ") +
                   t.getStopHour() + ":" +
                   t.getStopMinute() + ":" +
                   t.getStopSecond());

    for (int i = 1; i <= 7; i++)
    {
      if (t.isWeekdaySelected(i))
      {
        week_day |= (0x01 << (i-1));
        Serial.println(String("Day ") + i + " is selected");
      }
      else
      {
        week_day &= (~(0x01 << (i-1)));
      }
    }

    weekday_set[0] = week_day;
  }
  else
  {
    timer_start_set[0] = 0xFFFF;
    timer_stop_set[0] = 0xFFFF;
  }
}

//
################################################################################
#####################################

void setup()
{


   pinMode(trigPin1, OUTPUT);
   pinMode(Valve1, OUTPUT);
    pinMode(Valve2, OUTPUT);
  pinMode(echoPin1, INPUT);
   pinMode(trigPin2, OUTPUT);
   pinMode(echoPin2, INPUT);
digitalWrite(Valve1,HIGH);
digitalWrite(Valve2,HIGH);

  Serial.begin(9600);


  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
```

```
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  // Setup a function to be called every second
  timer.setInterval(1000L, myTimerEvent);
/*
   for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  */

}

void loop()
{
  Blynk.run();
  timer.run();
  // You can inject your own code or combine it with other sketches.
  // Check other examples on how to communicate with Blynk. Remember
  // to avoid delay() function!


}
```