



**POLITEKNIK SULTAN SALAHUDDIN ABDUL AZIZ SHAH**

**AUTOMATED POND WATER LEVEL  
CONTROL SYSTEM USING APPS AND MOTOR**

NAME  
ENGKU EZRYATUL ANIISA  
BT ENGU ZAILANI

REGISTRATION NO  
08DEU20F2027

**JABATAN KEJURUTERAAN ELEKTRIK**

**SESI 2 2022/2023**

**POLITEKNIK**

**SULTAN SALAHUDDIN ABDUL AZIZ SHAH**

**AUTOMATED POND WATER LEVEL CONTROL  
SYSTEM USING APPS AND MOTOR**

**NAME**

**ENGKU EZRYATUL ANISA  
BT ENGKU ZAILANI**

**REGISTRATION NO**

**08DEU20F2027**

This report submitted to the Electrical Engineering Department in fulfillment of the requirement for a Diploma in Electronic Engineering (Medical)

**JABATAN KEJURUTERAAN ELEKTRIK**

**SESI 2 2022/2023**

## CONFIRMATION OF THE PROJECT

The project report titled "Automated Pond Water Level Control System using Apps and Motor " has been submitted, reviewed and verified as a fulfills the conditions and requirements of the Project Writing as stipulated



Checked by:

Supervisor's name : Puan Irma Baizuri bt Mohd Akhir

Supervisor's signature:

Date :

Verified by:

Project Coordinator name :

Signature of Coordinator :

Date :



“I acknowledge this work is my own work except the excerpts I have already explained to our source”

1. Signature :

Name : **ENGKU EZRYATUL ANIISA BINTI ENGKU ZAILANI**

Registration Number : **08DEU20F2027**

Date :

**DECLARATION OF ORIGINALITY AND OWNERSHIP**

**TITLE : AUTOMATED POND WATER LEVEL CONTROL SYSTEM USING APPS AND MOTOR**

**SESSION: JUNE 2023**

1. I, **1.Im ENGKU EZRYATUL ANIISA BINTI ENGKU ZAILANI (NO KP: 020510-03-0030)**

is a final year student of **Diploma in Electronic Engineering (Medical), Department of Electrical, Politeknik Sultan Salahuddin Abdul Aziz Shah**, which is located at **Persiaran Usahawan,40140 Shah Alam Selangor Darul Ehsan**. (Hereinafter referred to as 'the Polytechnic').

- 2. I acknowledge that 'The Project above' and the intellectual property therein is 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 Electronic Engineering (Medical)** to me.

Made and in truth that is recognized by;

a) **ENGKU EZRYATUL ANIISA BINTI ENGKU ZAILANI**  
(Identification card No: - 020510-03-0030)

(.....)  
**ENGKU EZRYATUL ANIISA BINTI ENGKU ZAILANI**



In front of me, **PUAN IRMA BAIZURI BT MOHD AKHIR** (Click here to enter text.)  
As a project supervisor, on the date:

(.....)  
**PUAN IRMA BAIZURI BT MOHD AKHIR**

## **ACKNOWLEDGEMENTS**

I have taken efforts in this Project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I am highly indebted to (Puan Irma Baizuri Bt Mohd Akhir) for their guidance and constant supervision as well as for providing necessary information regarding the Project & also for their support in completing the Project.

I would like to express my gratitude towards my parents & member of (DEU5A) for their kind co-operation and encouragement which help me in completion of this Project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the Project and people who have willingly helped me out with their abilities.

## **ABSTRACT**

This paper focusing on implement a water level control system that is used for water tanks in agricultural facilities to perform crop watering activities. This will improve monitoring of water level in water tank if this sector facing a water crisis such as water supply cut off. This will help the farmers to prepare the next step with help water level control system that gives accurate water quantity in water tank or storage. Water Level Control System is helpful to the farmers knowing the actual amount of water in the water tank for agricultural usage. Blynk application as the Internet of things (IoT) platform that will give all the information about the water level and monitoring the activities of water consumption at a certain time. Hence, from the findings of the study, the water level monitoring system is successfully developed according to monitor the water level in the water tank. Based on the result analysis shows the Water Level Control System is capable of monitoring water level and also the Blynk application successfully displays the water level and notifies if the water level is at higher condition.



## **ABSTRAK**

*Kertas kerja ini memfokuskan kepada pelaksanaan sistem kawalan paras air yang digunakan untuk tangki air di kemudahan pertanian untuk melakukan aktiviti penyiraman tanaman. Ini akan meningkatkan kawalan paras air dalam tangki air sekiranya sektor ini menghadapi krisis air seperti bekalan air terputus. Ini akan membantu petani untuk menyediakan langkah seterusnya dengan bantuan sistem kawalan paras air yang memberikan kuantiti air yang tepat dalam tangki atau simpanan air. Sistem Kawalan Paras Air membantu petani mengetahui jumlah sebenar air dalam tangki air untuk kegunaan pertanian. Aplikasi Blynk sebagai platform Internet of things (IoT) yang akan memberikan segala maklumat tentang paras air dan kawalan aktiviti penggunaan air pada masa tertentu. Justeru, daripada dapatan kajian, sistem kawalan aras air berjaya dibangunkan mengikut pemantauan paras air dalam tangki air. Berdasarkan analisis keputusan menunjukkan Sistem Kawalan Paras Air mampu memantau paras air dan juga aplikasi Blynk berjaya memaparkan paras air dan memberitahu jika paras air berada pada keadaan lebih tinggi.*

## **TABLE OF CONTENTS**

<b>CONFIRMATION OF THE PROJECT</b>	<b>i</b>
<b>DECLARATION OF ORIGINALITY AND OWNERSHIP</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>ABSTRAK</b>	<b>vii</b>
<b>TABLE OF CONTENTS</b>	<b>viii</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF SYMBOLS</b>	Error! Bookmark not defined.
<b>LIST OF ABBREVIATIONS</b>	Error! Bookmark not defined.
<b>CHAPTER 1</b>	<b>1</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Background Research	2
1.3 Problem Statement	3
1.4 Research Objectives	3
1.5 Scope of Research	3
1.6 Project Significance	4
1.7 Chapter Summary	4
<b>CHAPTER 2</b>	<b>5</b>
<b>2 LITERATURE REVIEW</b>	<b>5</b>
2.1 Introduction	5
2.2 Control System	5
2.2.1 Microcontroller	6
2.2.2 Programmable Logic Control (PLC)	7
2.2.3 Arduino	7
2.3 Chapter Summary	8
<b>CHAPTER 3</b>	<b>9</b>
<b>3 RESEARCH METHODOLOGY</b>	<b>9</b>
3.1 Introduction	9
3.1.1 Block Diagram of the Project	9
3.1.2 Flowchart of the Project	10
3.2 Project Hardware	11
3.2.1 Software Development	11
3.2.2 Description of Main Component	12
3.2.2.1 Ultrasonic Sensor (HC-SR04)	12
3.2.2.2 Battery 9V	13
3.2.2.3 Battery Holder	13
3.2.3 Circuit Operation	14
3.3 Project Software	14
3.4 Prototype Development	15
3.5 Sustainability Element in The Design Concept	15
	viii

<b>CHAPTER 4</b>	<b>16</b>
<b>4 RESULTS AND DISCUSSION</b>	<b>16</b>
4.1 Introduction	16
4.2 Result and Analysis	17
<b>CHAPTER 5</b>	<b>19</b>
<b>5 CONCLUSION AND RECOMMENDATIONS</b>	<b>19</b>
5.1 Conclusion	19
5.2 Recommendation	19
5.3 Chapter Summary	19
<b>CHAPTER 6</b>	<b>20</b>
<b>6 PROJECT MANAGEMENT AND COSTING</b>	<b>20</b>
6.1 Introduction	20
6.2 Gantt Chart	20
6.3 Cost and Budgeting	21
6.4 Chapter Summary	21
<b>REFERENCES</b>	<b>22</b>
<b>7 APPENDICES</b>	<b>24</b>
APPENDIX A- PROGRAMMING CODING	24

## LIST OF TABLES

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
	Table 3-1: List of Component .....	11
	Table 6-1: Components and Materials .....	21

## LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1:	Block diagram of open loop and closed loop system	<b>Error! Bookmark not defined.</b>
Figure 3.1:	Flow chart of operation of the system	<b>Error! Bookmark not defined.</b>
Figure 3.2:	Circuit Diagram	<b>Error! Bookmark not defined.</b>
Figure 3.3:	Front view of the project	<b>Error! Bookmark not defined.</b>

# CHAPTER 1

## 1 INTRODUCTION

### 1.1 Introduction

The topic of our project is Automated Pond Water Level Control System using Apps and Motor. We intend to design a system which can monitor the level of water from water tanks. We know that the resources of water are continuously depleting, so we need to keep a check on our usage so that no extra water goes unutilized. If we monitor the water level in tanks we ensure that the tank never overflows and hence no extra water loss would happen. Water is one in all the foremost vital basic desires for all living beings. However sadly an enormous quantity of water is being wasted by uncontrolled use. The major losses happen either in homes or industries. So the system can be applied there to fill the void. Our main aim is to design a system which will be versatile, economical and simply configurable which will be able to solve water losing issues. The water level data can be used for various purposes for better management of water source. Monitoring water level from remote location may be very useful when it is not possible to visit location physically every time. The system uses a NodeMCU microcontroller and a HC-SR04 Ultrasonic sensor to do the work. The sensor sends a pulse of ultrasonic waves and the waves after hitting the water source, again are reflected back towards the sensor. The NodeMCU microcontroller computes the time taken for the journey and also computes the distance. After that a pump is used to always help to maintain the water level between 30- 70 % so that the users can always have adequate amount of water supply at their home.

## 1.2 Background Research

Automated pond water level control systems have become increasingly popular in recent years, as they offer a convenient and efficient way to maintain optimal water levels in ponds. These systems typically use sensors to detect the water level in the pond, and then activate a motor to add or remove water as necessary. In addition to providing a more consistent water level, automated pond water level control systems can help reduce water waste and prevent damage to the pond ecosystem. To implement an automated pond water level control system using apps and motors, several components are required. First, a sensor must be installed in the pond to detect the water level. There are many types of sensors available for this purpose, including float switches, pressure sensors, and ultrasonic sensors. Once the sensor detects that the water level has fallen below a certain threshold, it sends a signal to a microcontroller or other control device. The control device then activates a motor, such as a submersible pump or solenoid valve, to add water to the pond. The motor can be controlled using a variety of methods, including relays, transistors, or dedicated motor control circuits. The control device can also be connected to a mobile app or other software interface, allowing the user to monitor and control the water level remotely. One of the key benefits of using an automated pond water level control system is the ability to maintain a consistent water level, which is important for the health of aquatic plants and animals in the pond. Additionally, automated systems can help prevent water waste by only adding water when necessary, and can also prevent damage to the pond ecosystem by avoiding overfilling or draining the pond. There are several factors to consider when designing an automated pond water level control system. These include the size and shape of the pond, the type of sensor and motor to be used, and the control method and software interface. Additionally, care must be taken to ensure that the system is installed and operated safely, and that it does not pose a risk to people or animals in the area. Overall, an automated pond water level control system using apps and motors can provide a convenient and effective way to maintain optimal water levels in ponds, while also promoting water conservation and protecting the pond ecosystem.

### **1.3 Problem Statement**

Maintaining the right water level in a pond can be a challenging task, especially if the pond is large or located in an area that is difficult to access. Uncontrolled water levels can lead to the pond overflowing, damaging nearby structures, or causing erosion. On the other hand, too little water in the pond can harm aquatic life and affect the overall health of the ecosystem.

### **1.4 Research Objectives**

The main objective of this Project is to maintain the water level in the pond at a normal water level.

More specifically the principle objective of this research are:

1. To design an automated pond water level control system using apps and motor.
2. To reduce water waste by preventing overflowing of the loss of water due to evaporation.
3. To develop the proper water level in a pond is critical for the health of the ecosystem.

### **1.5 Scope of Research**

1. This Project is focusing to conserve water and reduce the cost of water usage.
2. The emphasis is to improve the overall health of the pond and promote the growth of aquatic life.
3. The main controller is using Blynk apps to control the water level.



## **1.6 Project Significance**

1. Efficiency: The system can help optimize water usage by ensuring that the pond is only filled with the required amount of water. This can help reduce water wastage and improve the efficiency of the pond.
2. Convenience: The use of an app to control the water level in the pond can provide a convenient way to manage the pond remotely. This can be particularly useful for larger ponds or for those located in hard-to-reach areas.
3. Cost savings: By optimizing water usage and reducing wastage, the system can help reduce water bills and save money in the long run.
4. Improved pond health: Maintaining the correct water level in a pond is important for the health of aquatic life and the overall ecosystem. An automated water level control system can help ensure that the water level is always at the optimal level for the pond's inhabitants.
5. Increased productivity: For ponds used for aquaculture or other commercial purposes, an automated water level control system can help improve productivity by ensuring that the pond is always at the optimal water level for growth and production.

## **1.7 Chapter Summary**

An automated pond water level control system using apps and a motor is a system that uses a combination of technology and mechanical components to maintain a consistent water level in a pond or other water feature. The system typically consists of a water level sensor, a microcontroller, a motor, and a mobile application. The water level sensor is used to detect the water level in the pond and send the data to the microcontroller. The microcontroller then analyzes the data and sends commands to the motor to adjust the water level as needed. The mobile application allows users to monitor and control the water level remotely. Users can set the desired water level and receive alerts if the water level falls below or exceeds the set level. Overall, an automated pond water level control system using apps and a motor provides an easy and efficient way to maintain a consistent water level, which is important for the health and wellbeing of aquatic plants and animals.

## **CHAPTER 2**

### **2 LITERATURE REVIEW**

#### **2.1 Introduction**

In the modern era, people are increasingly creating advanced technologies. Therefore, in order to produce a technology, it is necessary to conduct an in-depth study. The literature review is a reference and guide used to ensure that the design is designed to meet the engineering specifications and meet the needs of current users in order to be commercialized in the country and abroad. This study needs to be carried out to meet the permitted characteristics. It is important that the design matches the engineering aspect of the design. At the beginning of the study, planning is done to ensure that the design will be fulfilled. The concept of project scope in order to meet the desired project. The effort in finding reference materials is an important factor to guarantee the effectiveness of the entire project and the completed report. So with this, some information related to the project has been searched and collected so that this project is known and can function properly. The results of the study will be processed and applied in the design to meet the exact specifications and fulfill the requirements of the diploma project.

#### **2.2 Control System**

A control system is a set of mechanical or electronic devices that regulates other devices or systems by way of control loops. Typically, control systems are computerized. Control system are a central part of production and distribution in many industries. Automation technology plays a big role in these systems. The types

of control loops that regulate these processes include industrial control systems, such as supervisory control and data acquisition, systems and distributed control systems.

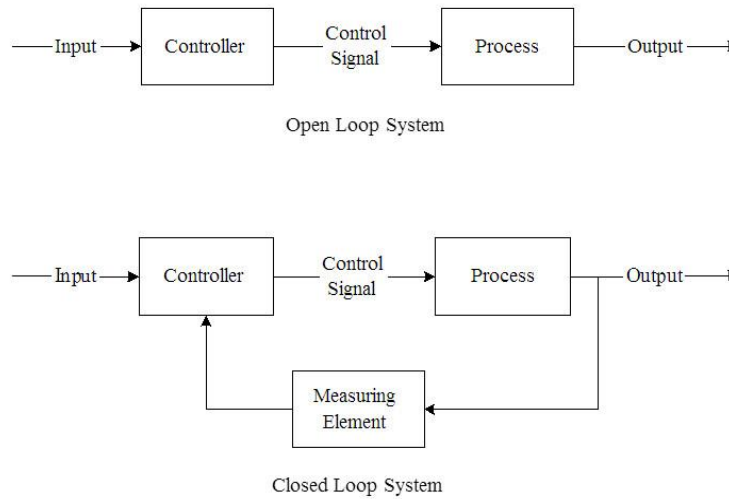


Figure 2-1: Block diagram of open loop and closed loop system

### 2.2.1 Microcontroller

A microcontroller is a small computer on a single integrated circuit (IC) chip. It contains a processor, memory, and input/output peripherals, all on a single chip. Microcontrollers are used in a wide range of applications, from controlling simple devices like thermostats and light switches to more complex systems like automotive electronics and industrial control systems. Microcontrollers are designed to be low-cost, low-power, and easy to use. They typically have a small amount of memory, ranging from a few hundred bytes to a few kilobytes, and are programmed using a specialized programming language or environment. Microcontrollers can be programmed to perform a wide variety of tasks, from simple data processing to complex control algorithms. They are commonly used in embedded systems, which are computer systems designed to perform specific tasks within a larger system.

### **2.2.2 Programmable Logic Control (PLC)**

A programmable logic controller (PLC) is a specialized computer system used in industrial automation and control applications. PLCs are designed to control a wide range of electromechanical processes, such as manufacturing assembly lines, robotic systems, traffic lights, and wastewater treatment plants. PLCs consist of a central processing unit (CPU), input/output (I/O) modules, and various communication interfaces. The CPU is responsible for executing the control program, which is typically created using a specialized programming language or graphical programming environment. The I/O modules provide a means of interfacing with the external world by accepting inputs from sensors and switches and providing outputs to actuators and other control devices. PLCs are highly reliable and ruggedized, making them suitable for use in harsh industrial environments. They are also easily programmable and can be reconfigured to accommodate changes in the manufacturing process or other control requirements.

### **2.2.3 Arduino**

Arduino is an open-source electronics platform designed for building prototypes and interactive electronic projects. It consists of both hardware and software components, including a microcontroller board, a programming environment, and a set of libraries and tools for building and uploading code to the board. The Arduino board is typically based on an Atmel AVR microcontroller, although other microcontroller architectures have also been used. It provides a range of digital and analog input and output pins that can be used to interface with various sensors, actuators, and other electronic components. The Arduino programming language is based on Wiring, a similar language that simplifies the process of writing code for microcontrollers. The Arduino software environment includes a code editor, a compiler, and a bootloader that allows code to be uploaded to the board over a serial connection. Arduino boards are widely used by hobbyists, students, and professionals alike for a wide range of applications, including robotics, home automation, environmental monitoring, and many others.

### **2.3 Chapter Summary**

This section is discovered about the technical part, including the selection of the type of controller. In this modern era, Internet of Things (IoT) is everywhere. Hence, the uses of selected microcontroller are in accordance with the rising of technologies nowadays.

## CHAPTER 3

### 3 RESEARCH METHODOLOGY

#### 3.1 Introduction

This section presents the framework planning that might be used to work plan regarding the topic. it comprises all of the information of the sensors and components including from microcontroller to the type of Wi-Fi module. this section also discusses the method used in order to achieve the objectives of this project. the main focuses of this chapter incorporate the design of the water monitoring system which is powered by Arduino Uno as a microcontroller as well as the application of the IoT technology that includes transmitting the data of water level in the water tank and display the data on the user device base on Blynk application.

##### 3.1.1 Block Diagram of the Project

In this section, the project design, components preparation, application of software, and project flow will be discussed thoroughly. Referring Figure 1 shows the block diagram of the project design that is using Arduino Uno as a microcontroller and ESP8266 as Wi-Fi module that will transfer data from the Ultrasonic sensor to Blynk Application.

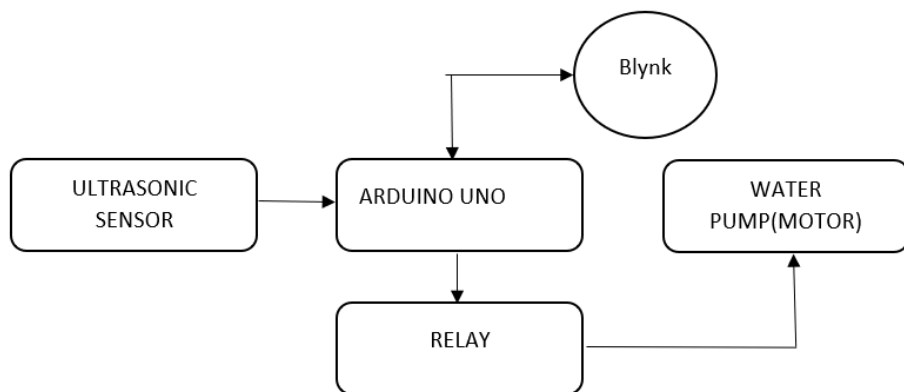


Figure 3-1: Block Diagram

### 3.1.2 Flowchart of the Project

Figure 2 shows the flowchart of the proceeds for the Automated Pond Water Level Control System. The process will start when the Ultrasonic Sensor measures the distance between the surface of the water and the Ultrasonic sensor. The data from the Ultrasonic sensor will transfer to Arduino Uno and calculate the water level based on the formula. If the level of the water tank is in a high condition, the water pump DC will be activated. At the same time, the data will transfer to the Wi-Fi module NodeMCU (ESP32). Lastly, the data will be received on the user's smartphone via the Blynk application.

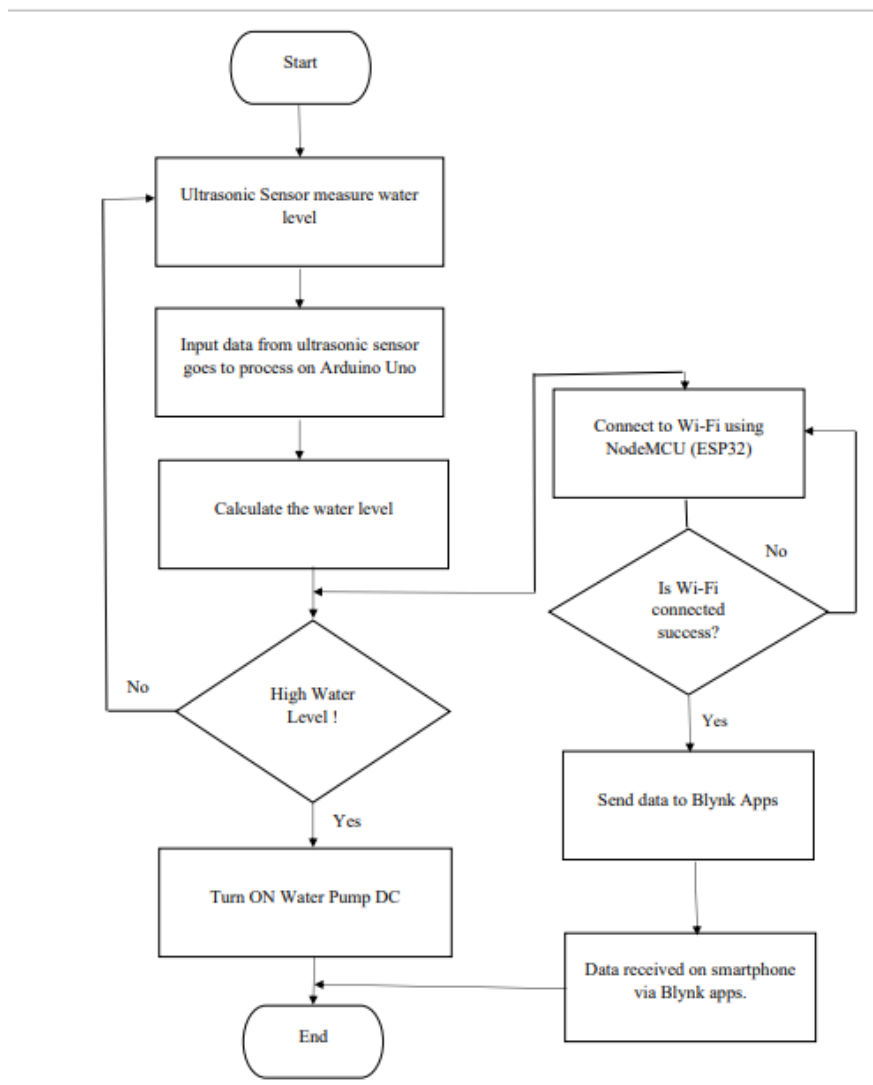


Figure 3-2: Flowchart of Automated Pond Water Level Control System

### 3.2 Project Hardware

In order to develop the Water Level Monitoring System, a few suitable components with high performance is used to achieve the objectives. All of the components were at a reasonable cost to buildan Internet of things (IoT) platform of water Level Monitoring System and each part or component has its advantages. Table 1 shows that the unit of component used in this project.

<b>Component</b>	<b>Unit</b>
Arduino Uno	1
NodeMCU (ESP32)	1
Ultrasonic Sensor (HC-SR04)	1
Single Relay	1
Water Pump DC	1
Battery	1
Battery Holder	1
Rubber Pipe	1

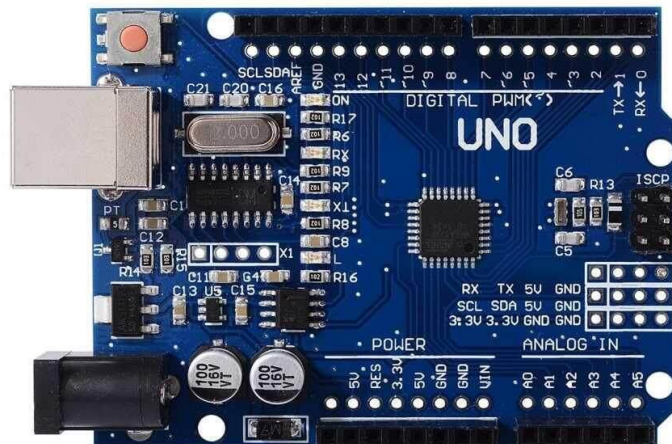
Table 3-1: List of Component

#### 3.2.1 Software Development

The software implemented and programmed for this project includes the Arduino Software IDE which allows the Arduino program to be uploaded to the Arduino board, as well as the Blynk IoT platform which allows users to monitor real time data on Blynk application. Blynk platform can allowthe user to create a digital dashboard where can build the interface based on the function of the project. Blynk can compatible with many types of microcontrollers such as Node MCU ESP32. The blynk application has three major components that are very useful for this project. First, Blynk application is used to control and display data on widgets. Next, Blynk serve is a cloud service responsible for all communication between smartphones and projects. Lastly, Blynk libraries, which include various widgets enable a device to send data obtained from a sensor to be displayed on a mobile application inconvenient way.



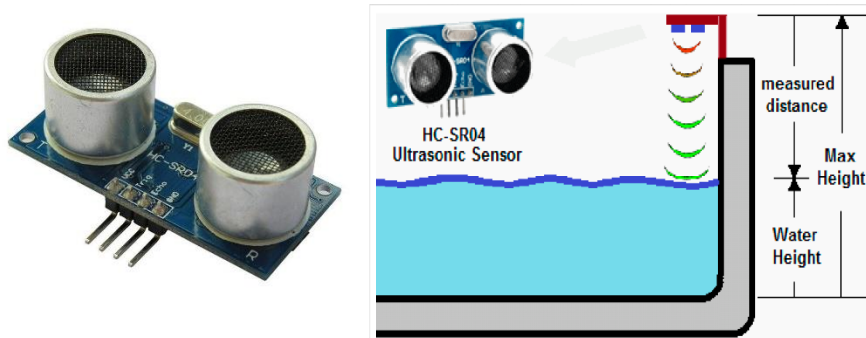
### 3.2.2 Description of Main Component



**Arduino Uno**

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.

#### 3.2.2.1 Ultrasonic Sensor (HC-SR04)



**Ultrasonic Sensor (HC-SR04)**

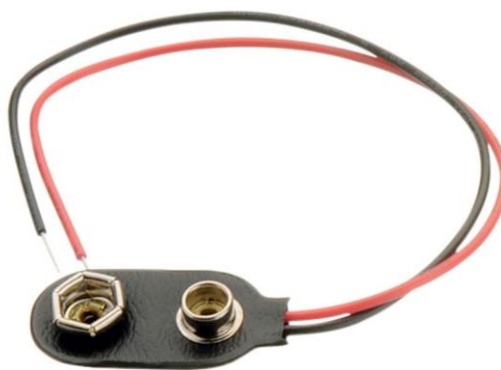
The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. This sensor reads from 2cm to 400cm (0.8inch to 157inch) with an accuracy of 0.3cm (0.1inches), which is good for most hobbyist projects. In addition, this particular module comes with ultrasonic transmitter and receiver modules.

### 3.2.2.2 Battery 9V



For the first transistor radios, the nine-volt battery, also known as the 9-volt battery, was developed. It has a top polarised snap connector and a rectangular prism form with rounded sides. Clocks, smoke alarms, and walkie-talkies all frequently employ this type. In primary carbon-zinc and alkaline chemistry, primary lithium iron disulfide, nickel-cadmium, nickel-metal hydride, and lithium-ion rechargeable forms, nine-volt batteries are frequently used. This type of mercury-oxide battery, which was once widely used, hasn't been produced in a long time due to its mercury concentration. This format has the designations NEDA 1604, IEC 6F22 (for zinc- carbon), or MN 1604 6LR61 (for alkaline). No of the chemistry, the size is typically referred to as PP3; this name was formerly only used for carbon-zinc or, in some countries, E or E- block.

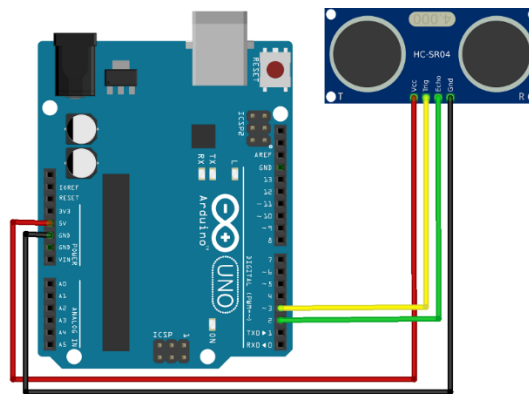
### 3.2.2.3 Battery Holder



A battery holder is a chamber or compartment that can accommodate one or more batteries. The holder must also make electrical contact with the battery terminals when using dry cells. Cables are frequently attached to the battery terminals for wet cells, as is the case with emergency lighting equipment or autos. A battery holder can be a separate plastic holder that is mounted using screws, eyelets, adhesive, double-sided

tape, or another method, or it can be a plastic case with the shape of the housing moulded as a compartment or compartments that receive a battery or batteries. Battery holders may have a lid to keep the batteries within and preserve them, or they may be sealed to stop battery leakage from harming the circuitry and other components. The two most popular ways to connect electrically within a holder are coils of spring wire or flat tabs that press against the battery terminals. On battery holders, external connections are typically established using pin contacts, surface mount feet, solder lugs, or wire leads.

### 3.2.3 Circuit Operation



### 3.3 Project Software

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a huge variety of Arduino-based projects.

### 3.4 Prototype Development



Figure 3-3 Front view of the Prototype

### 3.5 Sustainability Element in The Design Concept

The term sustainability is broadly used to indicate programs, initiatives and actions aimed at the preservation of a particular resource. However, it actually refers to four distinct areas: human, social, economic and environmental.

To avoid future water shortages, we need to be better about preserving water. Wasted water depletes already limited water resources, contributes to environmental degradation such as soil erosion and habitat destruction and results in increased energy consumption for water treatment and distribution.

IoT can tackle the issue of water waste in several ways: smart irrigation systems can monitor soil moisture and adjust watering schedules accordingly. Leak detection sensors provide alerts when a leak is detected in a water pipe. Smart water meters can track usage in real time and send alerts when usage patterns indicate a potential leak. Remote control of water valves can turn off the water supply in case of emergency or when not needed. And finally, predictive maintenance can detect potential failures in water infrastructure and prevent leaks before they occur.

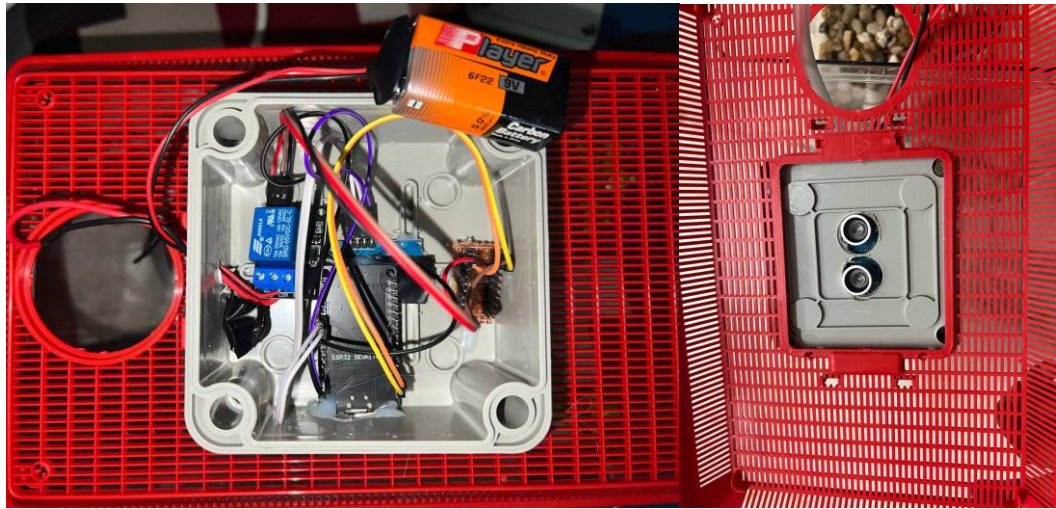
## CHAPTER 4

### 4 RESULTS AND DISCUSSION

#### 4.1 Introduction

In this project, we created an Automated Pond Water Level Control System Using Apps and Motor to modernize the agricultural industry with technology to make work easier. To create this prototype, we need to design circuits and applications using IoT. We design on breadboard and the main components used in this project are WiFi ESP32 NodeMCU, Relay and ultrasonic sensor and water pump. WiFi ESP32 NodeMCU was used so that we can communicate with Blynk application and it helps to build IoT products. The ESP32 library must first be installed in the Arduino Uno to use this command. For the water level monitoring systems, the measurement is ultrasonic. The sensor is important and giving high accuracy for ultrasonic reading to measure tank water level. With ultrasonic sensors, we can find the water depth calculation by finding the distance between the transceiver and the surface of the water. The sensor will transmit a short ultrasonic pulse, and we can measure the travel time of that pulse (the echo) to the liquid and back. The water pump works using water suction method which drain the water through its inlet and released it through the outlet. You can use the water pump as exhaust system for your aquarium and controlled water flow fountain.

## 4.2 Result and Analysis

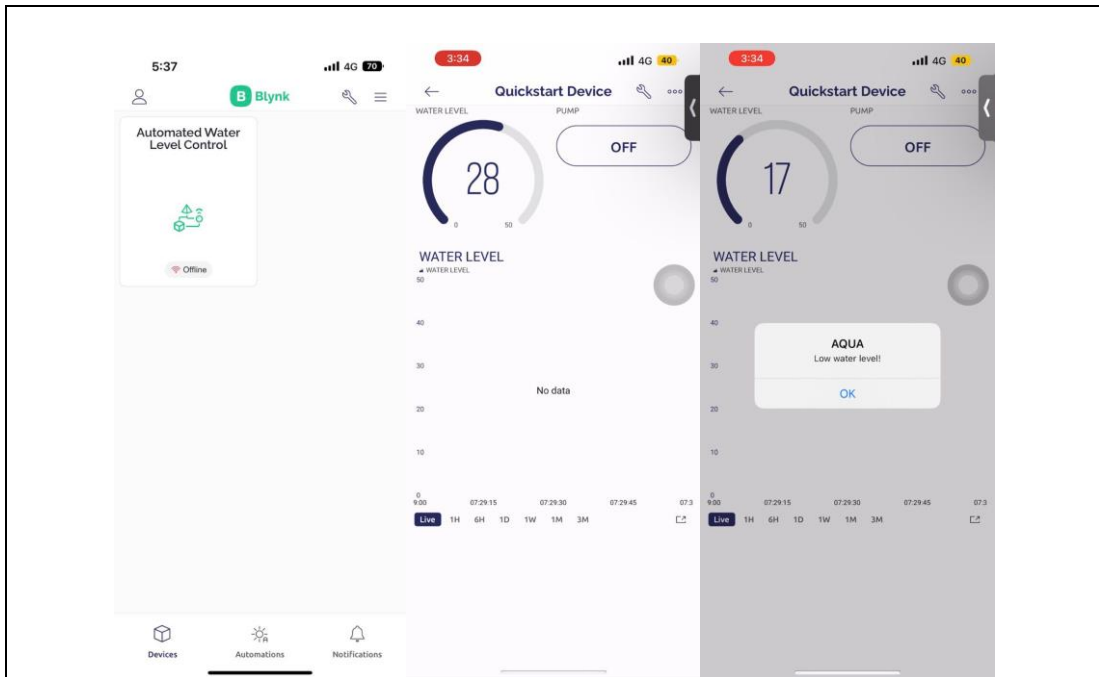


The image shown above is the components used in this project which is Arduino Uno, NodeMCU (ESP32), Ultrasonic Sensor (HC-SR04), Single Relay, Battery 9V and Battery Holder.

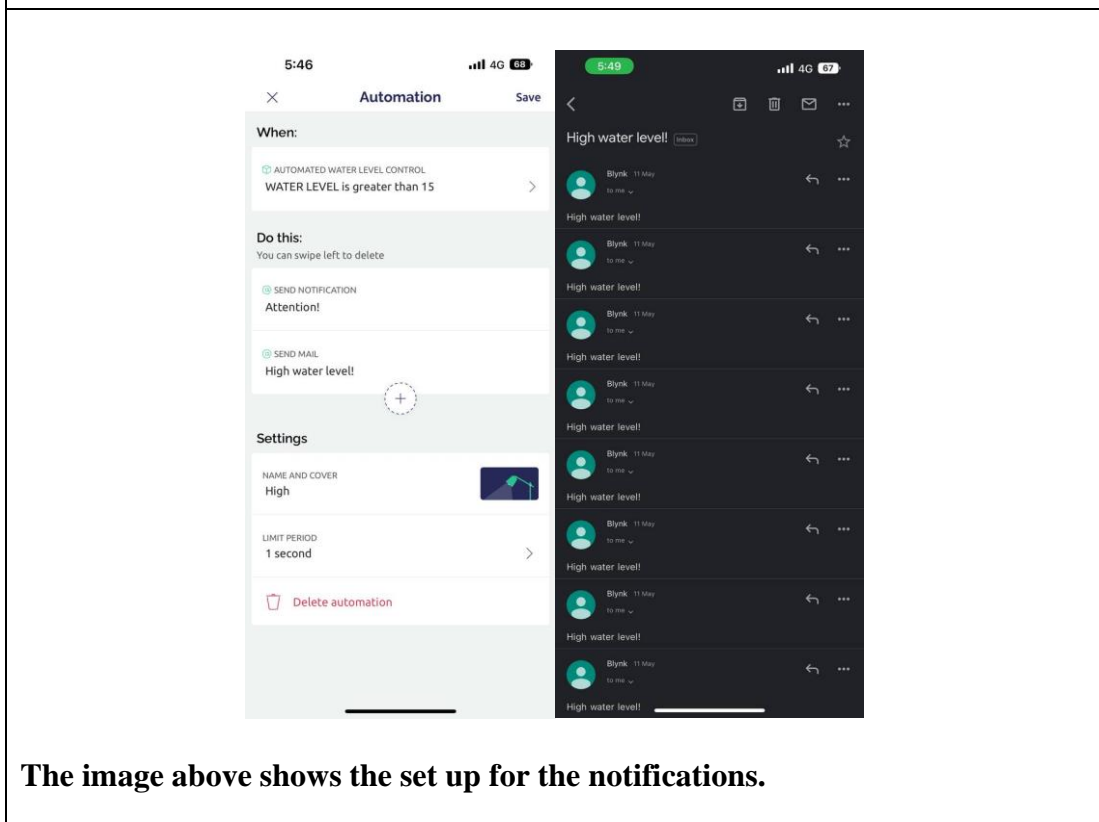


The image shown above is the connections of the relay and the water pump.

## Blynk Application



The image shown above is the Components used in the Blynk Apps.



The image above shows the set up for the notifications.

## CHAPTER 5

### 5 CONCLUSION AND RECOMMENDATIONS

#### 5.1 Conclusion

The project enables us to observe the level of water from a distant location and helps us to track it and protect it from overflowing and thereby enabling the user to ensure that no extra water gets used and there is no excess loss of water. We know that the major place where water gets wasted is industries and homes. So using this proposed system will help to minimize the water loss to a large extent. If the user knows about the water level in real-time he/she has the power to maintain the water loss to an extent by maintaining the water at a sustainable height, i.e., between 30-70% of the height of the original tank to ensure no overflowing.

#### 5.2 Recommendation

This project has extensive scope in the industrial field and the housing sector; by using a level control loop it can be ensured that the water level does not cross a certain limit and also by using a pump it is ensured that a minimum height is always achieved. These days in our busy life we need messages as reminder in our phones or laptops that are connected by internet and this system can give reminders regarding the water level to the user every-time in his/her hand. It will also help in creating a smart city centered around the idea of water preservation.

#### 5.3 Chapter Summary

At the end of this project , we hope that this project called An Automatic Pond Water Level Control using Apps and Motor will contribute and benefit everyone, especially to our country. This project has application in the home. Lastly , the other benefit is it can reduce wasting time , water and energy and also give the best quality for smart home.



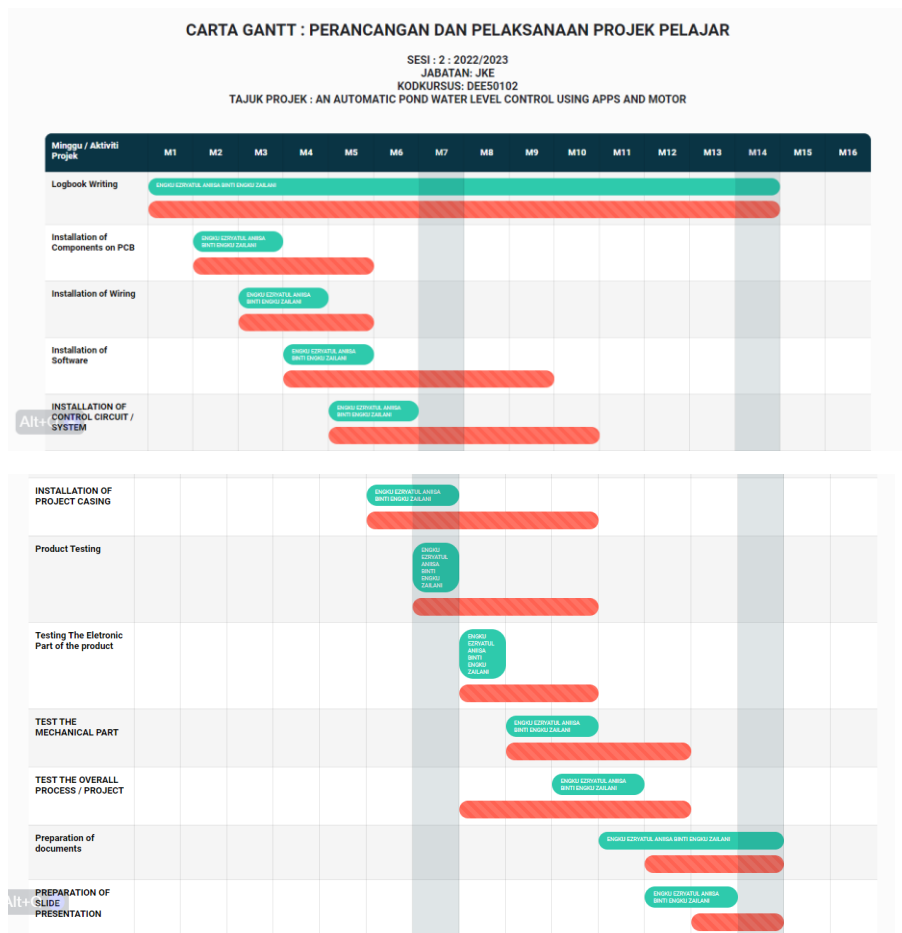
# CHAPTER 6

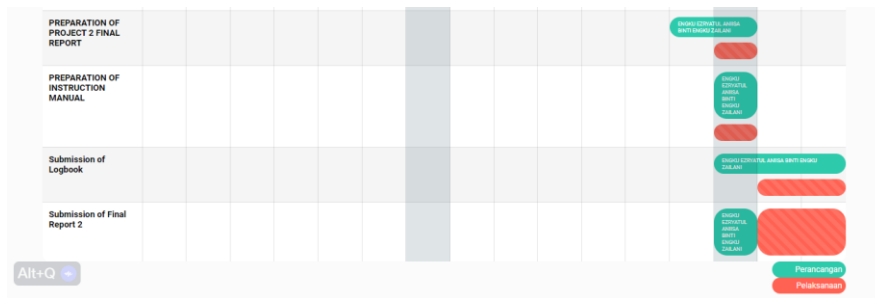
## 6 PROJECT MANAGEMENT AND COSTING

### 6.1 Introduction

The main advantage of the automatic pond water level control system using apps and motor is it has very low cost than the conventional one available in markets . For example, some commercial controllers use microcontrollers which alone costs around RM200. Some controllers even have a price range of RM200-RM400. But for our system, the components used are less in number and easily available. Hence losses will be less leading to a better efficiency.

### 6.2 Gantt Chart





### 6.3 Cost and Budgeting

No.	Components and materials	The unit price	Quantity	Total
1	Arduino Uno	RM 39.90	1	RM 39.90
2	NodeMCU (ESP32)	RM 29.90	1	RM 29.90
3	Ultrasonic Sensor (HC-SR04)	RM5.69	1	RM5.69
4	Single Relay 12V	RM 3.49	1	RM 3.49
5	Water Pump DC 5V	RM 1.86	1	RM 1.86
6	Battery 9V	RM 4.30	1	RM 4.30
7	Battery Holder	RM 0.40	1	RM 0.40
8	Rubber Pipe	RM 1.86	1	RM 1.86
9	Mini Aquarium	RM 16.90	1	RM 16.90
			<b>Total :</b>	<b>RM 104.30</b>
	List of other costing			
1	Transportation			
2	Postage			
3	Craft Work			
4	Internet			
5	Application			
			<b>Total :</b>	<b>RM 104.30</b>
			<b>Overall total</b>	<b>RM104.30</b>

Table 6-1: Components and Materials

### 6.4 Chapter Summary

The overall gross budget estimate in the implementation of this project is RM 104.30 as shown in Table 1. According to this budget cost, this project can be considered as a less costly project compared to other projects that can cost over a thousand ringgit. The cost of the project is also in line with one of the key features of a good project developer that is low cost but have a high quality project.

## REFERENCES

1. Asaad Ahmed Mohammedahmed Eltaieb, Zhang JianMin, "Automatic Water Level Control System", China – Tianjin
2. Ria Sood, Manjit Kaur, Hemant Lenka, "Design and Development of Automatic Water Flowmeter", Mohali, India
3. Sanam Pudasaini, Anuj Pathak, Sukirti Dhakal, Milan Paudel. "Automatic Water Level Controller with Short Messaging Service (Sms) Notification", Kathmandu University, Nepal
4. Lukman Adewale Ajao, Blessing Olatunde Abisoye, James Agajo, Abdulzeez Olorundare Ajao, Muhammed Bashir Muazu, Abdulzeez Femi Salami, Automated Multiple Water Tanks Control System Using Atmega And Fpga Technology, IEEE Mac 2019, Pp 347-353.
5. R. M. Shrenika, S. S. Chikmath, A. V. R. Kumar, Y. V. Divyashree And R. K. Swamy, "Non-Contact Water Level Monitoring System Implemented Using Labview And Arduino," 2017 International Conference On Recent Advances In Electronics And Communication Technology (Icraect), Bangalore, 2017, Pp. 306-309. [CrossRef]
6. Teetla Asha, Vidadala Srija, Design And Implementation Of Wireless Based Water Level Monitoring Using Arduino And Bluetooth, International Research Journal Of Motorering And Technology (Ijert), Volume: 07, Issue: 01 Jan 2020.
7. Siddartha Shankar, Dr M Dakshayini, Iot-Mobile Enabled Smart Water Level Controlling System To Regulate Water Wastage, 2018 Teee, Pp 2045-2048. [CrossRef]
8. Shandong Jianzhu "Design Of Intrinsically Safe Intelligent Water-Level Monitor Used In Coal Mine ", IEEE Trans. On Industrial Applications, Vol. 19, Pp. 1052 -1056 1983
9. K. Santhosh Kumar, G. Mukesh, K. Deepti, Microcontroller Based Automatic Water Level Control System, International Refereed Journal of Motorering And Science (Irjes) Volume 4, Issue 11, November 2015.
10. Beza Negash Getu and Hussain A. Attia, Automatic Water Level Sensor and Controller System. A02016 IEEE.
11. Teetla Asha, Vidadala Srija, Design And Implementation Of Wireless Based Water Level Monitoring Using Arduino And Bluetooth, International Research Journal Of Motorering And Technology (Ijert), Volume: 07, Issue: 01, Jan 2020.
12. Priya J, Sailusha Chekuri, Water Level Monitoring System Using Tot, International Research Journal Of Motorering And Technology (Irjet) Volume: 04 Issue: 12, Dec-2017.

13. Madhurima Santra, Sanjoy Biswas, Sibasis Bandhapadhyay And Kaushik Palit, Smart Wireless Water Level Monitoring & Pump Controlling System, International Journal Of Advances In Scientific Research And Motorering (Ijasre), Vol. 03, Issue 4, May -2017

## 7 APPENDICES

### APPENDIX A- PROGRAMMING CODING

```
#include <Wire.h>
// 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          "TMPL6EvKgg9X9"
#define BLYNK_TEMPLATE_NAME        "Quickstart Template"
#define BLYNK_AUTH_TOKEN           "v0WJA-cULzGPDipnkJcPnN0wFWNBZex2"
// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#define trigPin1 18
#define echoPin1 19
#define RXD2 16
#define TXD2 17
#define PUMP 5
#define LedY 4
#define vCalibration 83.3
#define currCalibration 0.50

int RIVER=0;
int dist;// LiDAR actually measured distance value
int strength;// LiDAR signal strength
int check;// check numerical value storage
int i;
int uart[9];// store data measured by LiDAR
const int HEADER=0x59;// data package frame header

int mode=0;

float COL=0;
// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;
const int potPin2 = 35;
const int potPin3 = 32;
const int potPin4 = 33;
const int potPin5 = 25;
```

```

float ADC1,ADC2,ADC3,ADC4;
// variable for storing the potentiometer value
int potValue = 0;
int NStat=0,MStat=0,DStat=0;
float h=0,t=0;
float hx=0,tx=0;
int PIRSTAT=0;
int BIT=0;
int ALM1=0,ALM2=0,ALM3=0,ALM4=0;
int Ready=0;
int Ml=0;
String MinS="00";
String HourS="00";
String SecS="00";
int DataIn=0;
String DATA="";
String Temp1x="";
String PHx="";
String Temp2x="";
String Temp1y="";
String PHy="";
String Temp2y="";
String Temp3y="";
String Temp3x="";
String Temp4y="";
String Temp4x="";
String currentTime;
String currentDate;
String TimerGet="00:00:00";
int MODE=0;

float LEVEL=0;
int Sec=0;

int ALM=0;
int Val=100;
int Index=0;
float CV=0;
int CKN=0;
//-----
int TDIS=0;
int Rly1=0;
int wait=0;
int Rly2=0;
int Rly3=0;
int Rly4=0;
int Rly5=0;
int pos1=0;

```

```

int pos=0;
int CK=0;
int STAT=0;
//-----
char auth[] = BLYNK_AUTH_TOKEN;

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

BlynkTimer timer;

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

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

  // process received value
}

BLYNK_WRITE(V10)
{
  int pinValue = param.asInt(); // assigning incoming value from pin V1
to a variable
  Rly1=pinValue;
  if (pinValue==1){
    digitalWrite(PUMP,HIGH);

  }
  if (pinValue==0){
    digitalWrite(PUMP,LOW);
  }

}

BLYNK_WRITE(V11)
{
  int pin2Value = param.asInt(); // assigning incoming value from pin
V1 to a variable
  Rly2=pin2Value;
  if (pin2Value==1){

  }
  if (pin2Value==0){

```

```

    }

}
BLYNK_WRITE(V12)
{
  int pin3Value = param.asInt(); // assigning incoming value from pin
V1 to a variable
  Rly3=pin3Value;
  if (pin3Value==1){
    MODE=1;

  }
  if (pin3Value==0){
    MODE=0;

  }
  // process received value
}

BLYNK_WRITE(V13)
{
  int pin4Value = param.asInt(); // assigning incoming value from pin
V1 to a variable
  Rly4=pin4Value;
  if (Rly4==1){
    ALM1=0;

  }

}

BLYNK_WRITE(V14)
{
  int pin5Value = param.asInt(); // assigning incoming value from pin
V1 to a variable
  Rly5=pin5Value;

  // process received value
}
//-----
-----

// 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_press
ed.png");
    // Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-
started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

void myLIDAR()
{

    //-----
}
void myTimerEvent()
{
    //-----
-
static unsigned long timepoint = millis();

    if (millis() - timepoint > 1000U) //time interval: 1s
    {
        long duration1x, distance1, duration2x, distance2, duration3x,
distance3, duration4x, distance4, duration5x, distance5;
        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;
//LEVEL=distance1/100.0;

        if (distance1<=30){
            LEVEL=30-distance1;

        }
        if (distance1>30){
            LEVEL=0;

        }

        Serial.println(LEVEL);
//-----

//    Serial.print(Smoke);
//    Serial.print("\t");

```

```

    // Serial.print(Mot);
    // Serial.print("\t");
    // Serial.print(ReedD);
    // Serial.print("\t");
    // Serial.println(ReedW);
    Blynk.virtualWrite(V1, LEVEL);

}
//-----
}

void setup()
{

    int i,k;

        pinMode(trigPin1, OUTPUT);

    pinMode(echoPin1, INPUT);

pinMode(PUMP,OUTPUT);

    delay(3000);

    Serial.begin(9600);

    while (!Serial);

    Serial.println ("Initializing...");
    //
    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);

delay(300);
}

void loop()
{

    Blynk.run();
    timer.run()
}

```