Automatic water dispenser and level monitoring system

Mini Project Report submitted in partial fulfillment
of the requirement for the degree of
T. E. (Information Technology)

Submitted By

Tanya Shrivastava (18101A0010)
Saamiya Newrekar (18101A0004)
Kushagra Soni (18101A0053)

Under the Guidance of
Prof. Kanchan Dhuri
Department of Information Technology

Vidyalankar Institute of Technology
Wadala(E), Mumbai 400 037

University of Mumbai

2020-2021

CERTIFICATE OF APPROVAL

For

Mini Project Report

This is to Certify that

Tanya Shrivastava
Saamiya Newrekar
Kushagra Soni

Have successfully carried out Mini Project entitled
"Water level monitoring system and contactless
water dispenser" in partial fulfillment of degree
course in Information Technology
As laid down by University of Mumbai during the academic year
2020-2021

Under the Guidance of
"Prof. Kanchan Dhuri"

Signature of Guide

Head of Department

Examiner 1

Examiner 2

Principal

Dr. S. A. Patekar

ACKNOWLEGEMENT

We express our profound gratitude and sincere thanks to **Prof. Kanchan Dhuri** our respectable project guide, for her gigantic support and guidance. Without her counselling our project would not have seen the light of the day.

We extend our sincere thanks to **Dr. Deepali Vora** Head of the Department of Information Technology for offering valuable advice at every stage of this undertaking. We would like to thank all the staff members who willingly helped us. We are grateful to VIDYALANKAR INSTITUTE OF TECHNOLOGY for giving us this opportunity.

1. **Tanya Shrivastava**
2. **Saamiya Newrekar**
3. **Kushagra Soni**

Table of Contents

1. Introduction 2
2. Aim and Objectives 3
3. Problem Statement 4
4. Proposed System 5
   4.1 Block Diagram 5
   4.2 Flow Chart 5
5. Components 6
   5.1 Hardware 6
      5.1.1 NodeMCU 6
      5.1.2 12V Relay Module 7
      5.1.3 Ultrasonic sensor 8
      5.1.4 Arduino Nano 8
      5.1.5 Bread Board: 8-9
      5.1.6 Jumper Wire 9
      5.1.7 Arduino Uno 9
      5.1.8 Water pump 9
      5.1.9 12V battery 9

   5.2 Software 9
      5.2.1 Arduino IDE 9
      5.2.2 Blynk App 100
6. Implementation 111
   6.1 Circuit Diagram 111
   6.2 Working 111
7. Results & Discussion .....
   7.1 Discussion 133
8. Conclusion & Future Scope 144
   References 165
   Appendix 166
      Code 166

List of Figures

Figure 4.1 Block Diagram 5

Figure 4.2 Flow Chart 5

Figure 5.1.1A NODEMCU 6

Figure 5.1.1B NODEMCU Pin Diagram 6

Figure 5.1.2 12V 4-Channel Relay Module 7

Figure 5.1.3 Ultrasonic sensor 7

Figure 5.1.4 Arduino Nano 7

Figure 5.1.5 Bread Board 9

Figure 5.1.6 Jumper Wire 9
Figure 5.1.7 Arduino Uno 10
Figure 5.1.8 Water pump 10
Figure 5.1.9 5V battery 10

Figure 5.2.1 Arduino IDE 11

Figure 5.2.2 Blynk App 11

Figure 6.1 Circuit Diagram 11

Figure 7.1 Apparatus with connection 12

Figure 7.2 High water level 13

Figure 7.3 Medium water level 13

Figure 7.4 Low water level 13

Abstract

This project aims at making people's lives easier, more convenient and safer (keeping in mind the post-COVID conditions). The main purpose of making this project is to make a water dispenser (and deploy it in our college's water dispensers) so that students can have/drink clean water without physically touching the water dispenser. This will be done using sensors (ultrasonic sensor) which calculates the distance between the nearest object and the dispenser and once the minimum distance condition is satisfied, the relay motor starts working and the water flow starts. This water dispenser system also helps in saving water as there's no need to turn the tap off manually. Everything is done by the sensors and the motors.

In addition to this contactless water dispenser, we've made an app (using blynk app) which tells us the water level in the dispenser in real time. The data from the sensor (ultrasonic sensor) is sent via Wi-Fi, using the Wi-Fi module on the nodeMCU. The data is transferred via the public nodeMCU servers.

The highlights of this app (IoT platform) include:

- Bar representation and gauge representation of water level in the container.

- Email system to notify about the overflowing water or low water levels.

- Notifications on the phone, about water overflow, which has the app installed.

So overall, the project is aimed at making water dispensing process more handy and safer.

1. Introduction

Approximately 71 percent of the Earth's surface is water-covered, and the oceans hold about 96.5 percent of all Earth's water. From that only 2.5 percent can be used to drink. Therefore, only 2.5 percent is fresh water. So as human beings we must save the water. This online web-based system was developed to reduce the water wastage during the transition. Further, this can be developed as an emergency notification during flood situations. One of the major problems faced by most of the countries is the issue of water scarcity in the world and wastage during transmission. Basically, human interference is required to control the pumps and other devices to deploy computing techniques in creating a barrier wastage in order to provide more financial gains and help the environment and the water cycle which in turn ensures water conservation for our future. The idea of automatic controlling involves designing a control system to function with minimal or no human interference. IoT based Water Level Monitoring system is an innovative system which will inform the users about the level of water and will prevent it from overflowing.

This simple water level indicator uses a simple circuit to get the details of the water level of the tank. The system has only used sensors and cables to get results and it also contains a software to retrieve data.
When the water level is high or too low, the app sends a notification on the phone along with an email.

In this system water level is monitored by using ultrasonic sensors. The basic principal of ultrasonic distance measurement is based on ECHO (Stoddard & Jeo, 2007). When sound waves are transmitted in environment, they return back to the origin as ECHO after striking on any obstacle. The only calculation of this system is the traveling time of both sound waves (outgoing time and returning time to origin after striking on any obstacle). And after few calculations it is possible to get the results.

2. Aim and Objectives

The aim of the project is to develop a Water level monitoring system and integrate it with a contactless water dispenser system using NodeMCU which will send data about water level from anywhere and we can also view water levels on the blynk app.

The core objectives are:

- Gather system requirements
- Evaluate and study the platform required for the system
- Evaluate and study suitable development language, technologies and tools
- Evaluate Methods of Interface
- Program NodeMCU
- Interface board for Relay Module.
- Program Blynk App
- Evaluate and test the system
- Maintain system

3. Problem Statement

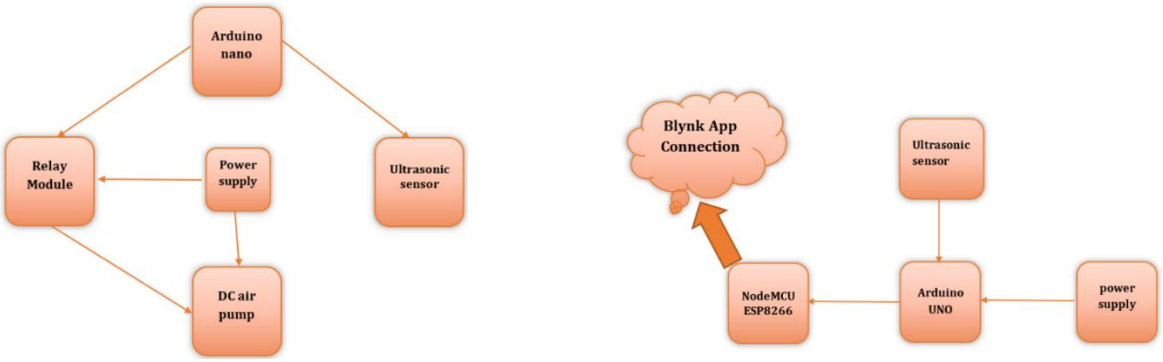
In the present-day scenario water conservation and its preservation has become a vital responsibility for a sustainable future of ours and the coming generations. So, for this we've made a simple project which would help dispense water more safely , tracing the live water level non our phones.

More formally,

To offer convenience and ease in dispensing water to users keeping in mind the social distancing norm and avoid overflowing of water using NodeMCU and BLYNK app

4. Proposed System

4.1 Block Diagram



Water dispenser

Water level monitor

Figure 4.1 Block Diagram

4.2 Flow Chart

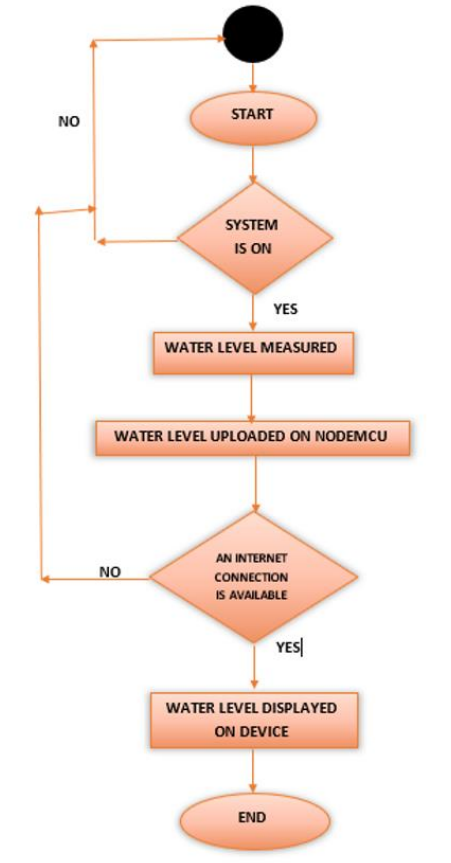


Figure 4.2 Flow Chart

5. Components

5.1 Hardware:

5.1.1 NodeMCU

NodeMCU v3 is a development board which runs on the ESP8266 with the Espress if Non-OS SDK, and hardware based on the ESP-12 module. The device features 4MB of flash memory, 80MHz of system clock, around 50k of usable RAM and an on chip WIFI Transceiver.

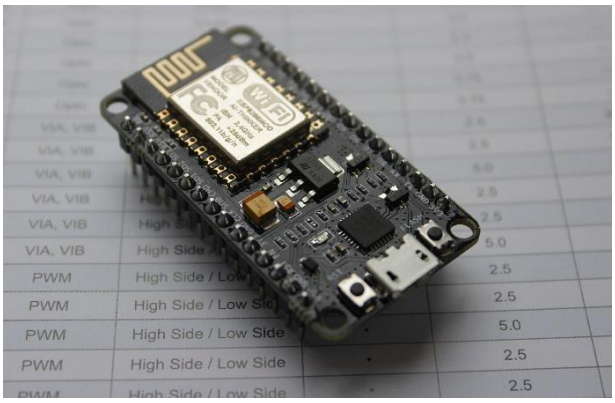


Figure 5.1.1A NODEMCU

Features:

Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106

Operating Voltage: 3.3V

Input Voltage: 7-12V

Digital I/O Pins (DIO): 16

Analog Input Pins (ADC): 1

UARTs: 1

SPIs: 1

Flash Memory: 4 MB

SRAM: 64 KB

Clock Speed: 80 Mhz

Wi-Fi: IEEE 802.11 b/g/n:

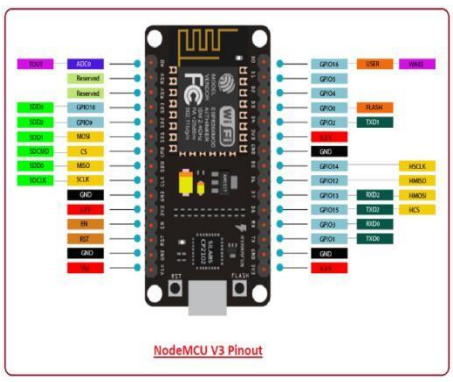


Figure 5.1.1B NodeMCU PIN Diagram

5.1.2 12V 4-Channel Relay Module

This is a 12V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equiped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by micro-controller.

Figure 5.1.2 12V 4-Channel Relay Module

5.1.3 Ultrasonic sensor

An ultrasonic sensor/transducer is a device used to make measurements based on the transmission and reception of UV rays. The concept used is ECHO, which uses the time difference between transmission and reception of the UV rays.

Figure 5.1.3 12V Ultrasonic sensor

5.1.4 Arduino Nano

The **Arduino Nano** is a small, complete, and breadboard-friendly board based on the ATmega328P. It offers the same connectivity and specs of the Arduino Uno board in a smaller form factor.

Figure 5.1.4 Arduino Nano

5.1.5 Bread Board:

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

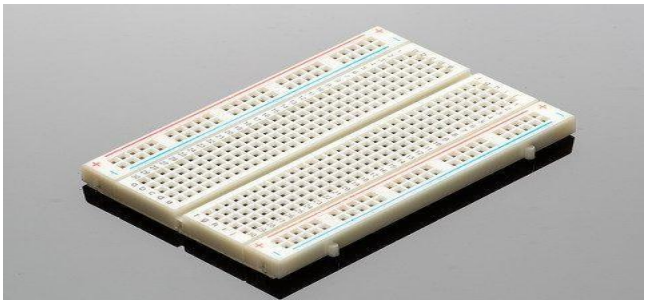


Figure 5.1.7 BreadBoard

5.1.6 Jumper Wire

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering.

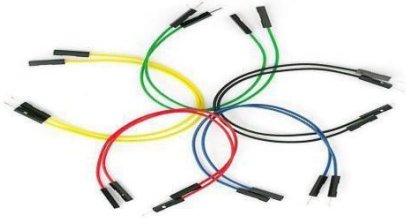
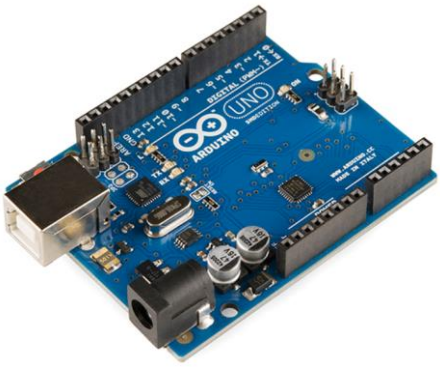


Figure 5.1.8 Jumper Wires

5.1.7 Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.[2][3] The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.



5.1.8 water pump

Micro DC 3-6V Micro Submersible Pump Mini water pump For Fountain Garden Mini water circulation System DIY project. This is a low cost, small size Submersible Pump Motor which can be operated from a 3 ~ 6V power supply.

5.1.9 Battery

12V battery for power supply to the motor.

5.2 Software

5.2.1 Arduino IDE

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.

Figure 5.2.1 Arduino IDE

5.2.2 Blynk App

Blynk is a new platform that allows you to quickly build interfaces for controlling and monitoring your hardware projects from your iOS and Android device. After downloading the Blynk app, you can create a project dashboard and arrange buttons, sliders, graphs, and other widgets onto the screen. Using the widgets, you can turn pins on
and off or display data from sensors.

Figure 5.2.2 Blynk App

6. Implementation

6.1 Circuit Diagram

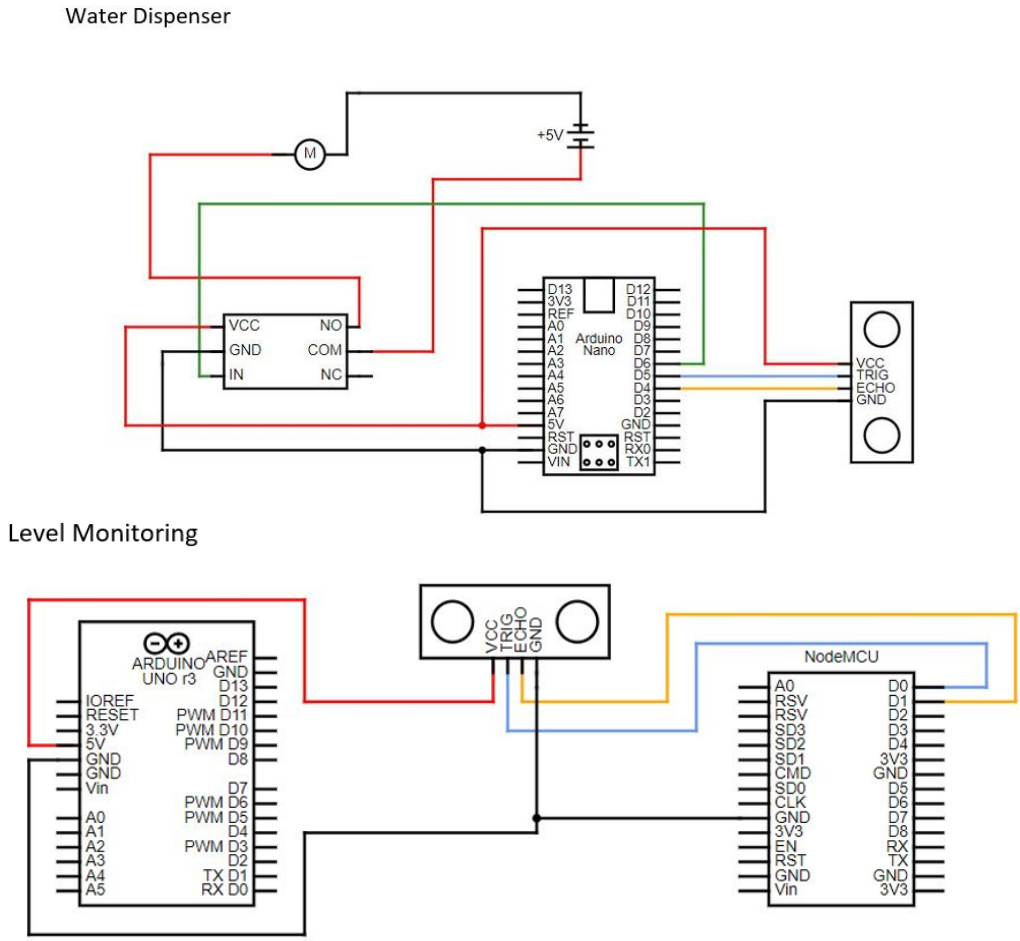


Figure 6.1 Circuit Diagramu

6.2 Working

1. Water Dispenser

> Components used are Arduino Nano, Ultrasonic sensor, Relay module and water pump
> The 5V and GND pins of Arduino Nano are connected to positive and negative of sensor, relay module and water pump. Also allows the data transfer.
> Ultrasonic sensor acts as key component which helps in detecting the object later water flows. Here pins trig and echo are connected to data pin D5 and D4 of Arduino NANO.
> Relay is controlled through Arduino which is connected to pin D6 from IN pin. The water pump is connected to the relay using a normally open configuration.

2. Level Monitoring

> Components used are Arduino UNO, Ultrasonic Sensor, NodeMCU ESP8266
> The 5V and GND pins of Arduino UNO are connected to VCC and GND pins of Ultrasonic sensor and NodeMCU which provides power across whole circuit.
> Ultrasonic sensor is controlled by NodeMCU were pins trig and echo are connected to the data pin D0 and D1 of ESP8266 Wi-fi module.
> NodeMCU act as a Wi-fi module which connects the hardware component to software component (Blynk app) through code. The final result is displayed on the Blynk App.

7. Results & Discussion

Figure 7.1 Apparatus with connection

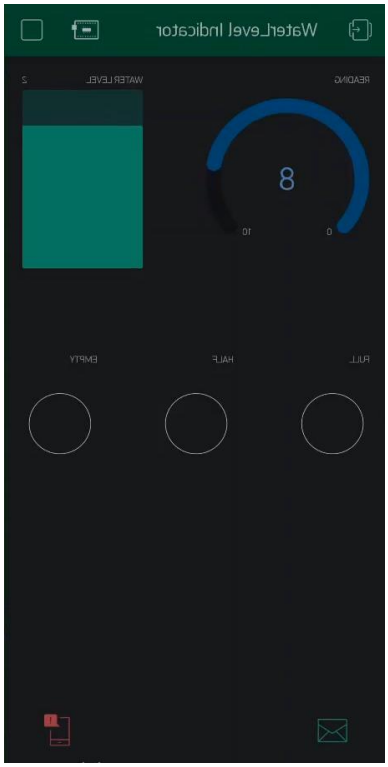


Figure 7.2 good water level

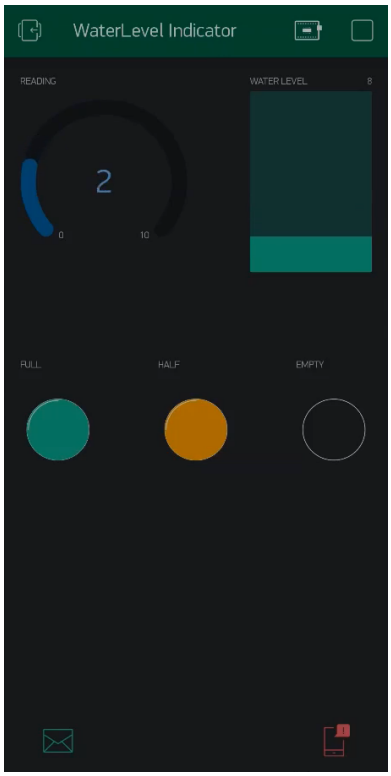


Figure 7.3 medium water level

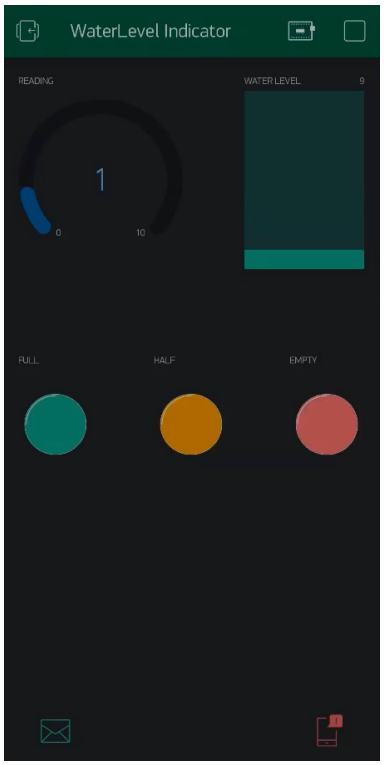


Figure 7.4 low water level

Figure 7.1: Initial apparatus of the project with all the connections made

Figure 7.2: Blynk application showing high water level in the container

Figure 7.3: Application showing medium water level

Figure 7.4: Application showing low water level.

7.1 Discussion

During the whole period of the project we gained a lot of knowledge on the NodeMCU board and programming in Arduino IDE. If we talk about the achievements out of the project when starting to do the project it was to interface it with the Relay Module. We came to know a lot about the relay that it convert AC voltage to DC. NodeMCU can be connected the Blynk app which a very good app and there are lots of components which can be used to make better IoT projects in the future and also it can send data to its server and able to collect the data and send the alerts accordingly .There are many Blynk libraries in Arduino itself which can be used Out of which all the work was completed.

8. Conclusion & Future Scope

The aim of this project was to make a contactless water dispenser system and an extension was made. A blynk app was made to check the live water level in the same dispenser on the phone. In conclusion, we can say that both the components of this project work fine and as expected. The app shows live water level. The data shown on the app is delivered to it using the Wi-fi module of nodeMCU. The water dispenser also works fine.

Scope of improvement in this project includes:

- Making an entire ecosystem of such water dispensers such that if a dispenser has less amount of water, the user can get the information about the nearest water dispenser which has adequate amount of water.
- Giving a timely analysis of water saved/wasted. A good analysis using graphs and similar visuals can be used to give deeper insights of the user is performing. This would make use of time-series data.
- Credits can be added. Users who perform exceptionally good can be compensated by some coupons or other relevant gifts/resources. This would inculcate a sense of competition for saving more water, which would in turn be beneficial for all.

References

[1] https://youtu.be/uJ9r2lP-ARA - for water dispenser

[2] https://www.youtube.com/watch?v=62-ufbh24m4&list=WL&index=90&t=83s – for water levelling system.

[3] https://nodemcu.readthedocs.io/en/master/

[4] https://blynk.io/

[5] https://stackoverflow.com/

[6] Saloni Khamdelwal and Preteek Dubey 'IOT based Water Monitoring System', *IETE,* Year 2020.

[7] Dissanayaka Rmsm, Helani Wickramaarachchi, 'IoT Based Water Level Monitoring System Using NodeMCU', 11th Symposium on Applied Science, Business & Industrial Research 2019.

[8] Gideon Okoro, Gloria Ezenne 'DEVELOPMENT OF A LOW COST AUTOMATIC WATER LEVEL MONITORING SYSTEM', Faculty of Engineering University of Nigeria International Conference, April, 2018.

Appendix

Code

```
//WATER DISPENSER
#define trigger 5
#define echo 4
#define Relay 6
float time=0,distance=0;
void setup()
{
Serial.begin(9600);
 pinMode(trigger,OUTPUT);
 pinMode(echo,INPUT);
 pinMode(Relay,OUTPUT);
 delay(2000);
}
void loop()
{
 measure_distance();

 if(distance<5)
 {
   digitalWrite(Relay,LOW);
 }
 else
 {
```

```
   digitalWrite(Relay,HIGH);
 }
 delay(500);
}
void measure_distance()
{
 digitalWrite(trigger,LOW);
 delayMicroseconds(2);
 digitalWrite(trigger,HIGH);
 delayMicroseconds(10);
 digitalWrite(trigger,LOW);
 delayMicroseconds(2);
 time=pulseIn(echo,HIGH);

 distance=time*200/20000;
}
```

```
// water level monitoring
#define trigger D0
#define echo D1
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "x8yT0c1ZiVvEIl8DSoaDkEPzJ1ieRvP1";
char ssid[] = "********";
```

```
char pass[] = "*******";
void setup()
{
  Serial.begin (9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
}
void loop()
{

  long duration, distance, reading;
  digitalWrite(trigger, LOW);
  delayMicroseconds(2);

  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);

  digitalWrite(trigger, LOW);
  duration = pulseIn(echo, HIGH);
  distance = (duration/2) / 29.1;
  reading = (10-distance);
  Serial.println(reading);
  Blynk.virtualWrite(V3, distance);
  Blynk.virtualWrite(V4, reading);
  if (distance >= 9) {
    Blynk.virtualWrite(V0, 255);
```

```
    if (distance == 9){
    Blynk.email("tanyashrivastava2000@gmail.com","EMPTY!!,Please refill the tank");
    Blynk.notify("EMPTY!!,Please refill the tank");
    }
}
  else {
    Blynk.virtualWrite(V0, 0);
  }
 if (distance >= 5) {
    Blynk.virtualWrite(V1, 255);
}
  else {
    Blynk.virtualWrite(V1, 0);
  }
   if (distance >=3) {
    Blynk.virtualWrite(V2, 255);
}
  else {
    Blynk.virtualWrite(V2, 0);
  }
Blynk.run();
}
```