KGiSL Institute of Technology
(Affiliated to ANNA University, Chennai and Approved by AICTE, New Delhi)
365, KGiSL Campus, Thudiyalur Road, Saravanampatti
Coimbatore – 641035

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Naan Mudhalvan -IOT

SMART WATER MANAGEMENT

SUBMITTED BY

MOHAMMED SHEBEEB T S	711721243062
ABHIJITH KUMAR N	711721243004
GURUJITH P M	711721243028
DEEPAK B	711721243021

CONTENTS

ABSTRACT

This project helps to regulate the proper maintenance of water tank information to monitoring section with proper updation of records. problem affects various processes in water management, such as water consumption, distribution, Water dust formed in the water tank. These problem can overcome by implementing proper monitoring system and information update system. Set of sensor like Turbidity, Salt sensor, pH sensor and Water flow sensors were used. This sensor informs about the water level tank and communicate to the monitor section. To maintain the tank without bacteria and microbes the Chlorine powder is sprayed if there is any changes in the ph value is found. If the water level reaches minimum position the motor automatically starts and when it reaches the maximum the motor stops automatically. A wireless sensor network is formed by connecting two and more water tanks using RF radio channel transceiver with monitoring section.

Chapter 1

INTRODUCTION

Internet of Things has been associated with cities, smart homes and also to manage traffic system.A unknown fact that about internet of things technology is also application across many other fields in our everyday life. Another such area where the internet of things technology can play a major role in water management. IOT is evolving fast and latest innovation occurring in wireless technology and embedded technology. This work focuses on a solution for water management in colleges, building and commercial area with the help of IOT. Water is precious and supply the needs to be regulated. To maintain the water in a proper way, should prevent the overflow of water in tanks and usage of the water in proper manner. In traditional days there is no proper maintenance of water. In conventional tanks there is need of human being to ON/OFF the motor. In this paper the automated system is introduced which is used to save the human work and cost. In this system the motor is automatically ON/OFF by using level sensor. The usage of water is observed by the water flow sensor..

1.1Project Definition

The planning, acquisition, construction, improvement or extension of water lines, pumps and related equipment and facilities, and any land, public ways or other interests in real estate, whether located within or outside of an assessment district, necessary or incident to the transportation and distribution of water from a water facility located within or outside of an assessment district to property located within an assessment district, all for the purpose of providing potable, sanitary water suitable for human consumption and use.

1.2Project Objectives

Encouraging settings for water resource use that are important socially, economically viable, and environmentally sustainable. Additionally, they aim to maximize the advantages and minimize the risks associated with the current hydraulic infrastructure. A common goal of all of these projects is to integrate policy approaches into other sectoral policies across a larger area of the nation. This frequently entails creating administrative, technical, and social water resource management tools

1.3 Project Specifications

1.Cleanliness maintenance
2.Integration with Weather Data
3.Alerts and Notifications
4.Data processing and transmission
5.Compliance and Safety

Chapter 2

LITERATURE SURVEY

In this paper Smart Water Management system using Microcontroller ZR16S08 as IoT Solution presents that system operates through the smart monitoring of the water flow in pipes of the water distribution network, aiming to ensure the quality of the water supply, knowing that water losses characterize one of the great problems in the world, as pipe holes may be open doors to water contaminants. In the Water Management system, IoT, automation is one of the essential attributes. This increases comfort and convenience in the lives of people. We would like to provide this in the domain of water management. Our motive is to help the readers understand the importance of using water judiciously and equipping them with the knowledge of the functioning of water management system which is done by using Internet of Things (IoT). OPC UA (Object Linking and Embedding for Process Control Unified Architecture) is a platform independent service-oriented architecture for the control of processes in the logistics and manufacturing sectors. Based on this standard we propose a smart water management model combining Internet of Things technologies with business processes coordination and decision support systems. They provide an architecture for sub-system interaction and a detailed description of the physical scenario in which we will test our implementation, allowing specific vendor equipment to be manageable and interoperable in the specific context of water management processes.

Chapter 3

SYSTEM STUDY

3.1 PROPOSED SYSTEM

In the Society Water Management with Automatic Bill Generation system, we proposed IoT based automation instead of manual water management. An electronic system is designed to control and monitor the level of water in a tank. The electronic system is designed to automatically control and display water levels. In our project the system achieves proper water management and enhances productivity from automation. Our idea is to detect the usage of water by individual flats and bill them accordingly using IOT based Flow sensor sensors

3.1.1 ARCHITECTURE OF PROPOSED SYSTEM

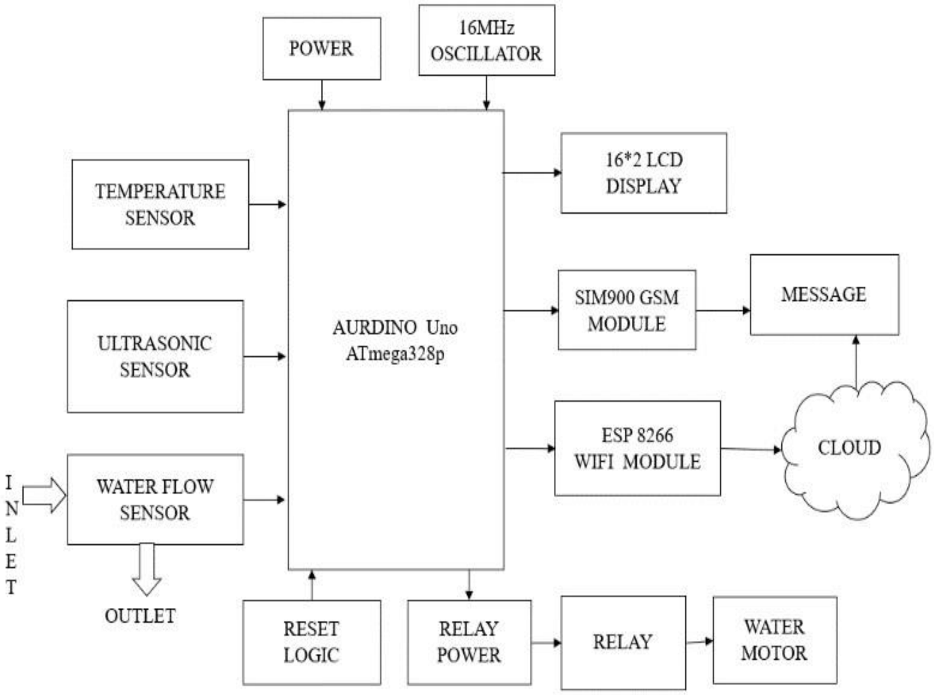


Fig1:smart water management system

Chapter4

SYSTEM ENVIRONMENT

4.1 Hardware Requirement

Hardware tools are required:

- Microcontroller ESP8266
- LCD Display
- Power supply
- Arduino UNO
- Water flow sensor
- Temperature sensor
- Ultrasonic sensor
- Ph sensor
- Buzzer

4.2Software Requirement

- Python 3.0

Chapter 5

SYSTEM DESIGN

5.1 IOT SYSTEM DESIGN:

5.1.1 Microcontroller

Encouraging settings for water resource use that are important socially, economically viable, and environmentally sustainable. Additionally, they aim to maximize the advantages and minimize the risks associated with the current hydraulic infrastructure. A common goal of all of these projects is to integrate policy approaches into other sectoral policies across a larger area of the nation. This frequently entails creating administrative, technical, and social water resource management tools

Fig2:ESP8266

5.1.2 LCD

LCD stands for Liquid Crystal Display. By using the LCD, all the outputs are displayed. LCD doesn’t know about the content (data or commands) supplied to its data bus. It is the user who has to specify whether the content at its data pins are data or commands.

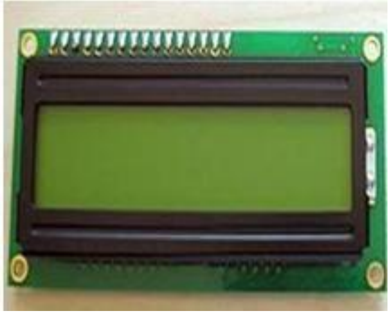


Figure 3: LCD Display

For this, if a command is inputted then a certain arrangement of 0’s and 1’s has to be applied to the Control lines so as to specify it is a command on the other hand if a data is inputted at the data lines then an another combination of 0s and 1s has to be applied to the control lines to require it is Data.

5.1.3 BUZZER

Buzzer is also called as Beeper. It is a sound signalling mechanical device.

Figure 4: Buzzer

5.1.4 TEMPERATURE SENSOR

Connecting power source(7-12v DC) to DC power jack □ Connecting a battery lead to Vin and Gnd. 5V and 3.3Vare used to provide power to sensors and modules when connecting it to. The temperature sensor LM35IC has been used for sensing the temperature. It is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature. The temperature can be measured more accurately with it than using a thermistor. The sensor circuit ryis sealed and not subject to oxidation, etc. It is a three terminal sensor used to measure the surrounding temperature ranging from -55 degree centigrade to 150 degree centigrade.

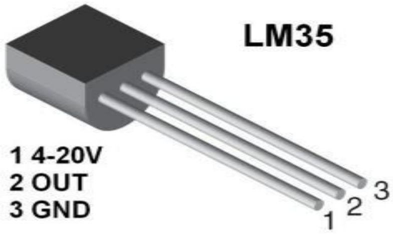


Figure 5: Temperature sensor

5.1.5 WATER FLOW SENSOR

Water flow sensor are installed at the water sources or pipes to measure the rate of flow of water and calculate the amount of water flowed through the pipe. Rate of flow of water is measured as liter per hour or cubic meters. Water flow sensor consists of plastic valve from which water can pass. A water rotor along with a hall effect sensor is present and measure the water flow. The main working principle behind the working of this sensor is the hall effect. According to this principle , in this sensor , a voltage difference is induced in the conductor due to rotation of the rotor. This induced voltage difference is transverse to the electric current.

Figure 6: Waterflow Sensor

5.1.6 SONIC SENSOR

The Sonic Sensor is used for computing the depth. Here it is used to measure the depth of the septic tank. The Sonic Sensor is fixed into the Septic tank. Then the Septic tank get filled means, it can sends the communications to particular organization. Then they will allot persons to clean the septic tank. Then septic tank cleaners will clean the tank. After cleaning it, the sensor can detect the level, and send messages to consistent organization.

Figure 8: Sonic sensor

This ultrasonic sensor can be used for measuring distance, object sensor, motion sensors etc. High sensitive module can be used with microcontroller to integrate with motion circuits to measure the distance, position & motion sensitive products.

In a nutshell, water depth sensing is using a sensor to measure the depth of water in a tank or container. Although various sensors can be used for this application, we will talk about ultrasonic sensor application.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 to the liquid and back. We can then subtract that distance from the total depth of the tank to determine the water depth.

.

BLOCK DIAGRAM OF PROPOSED SYSTEM:

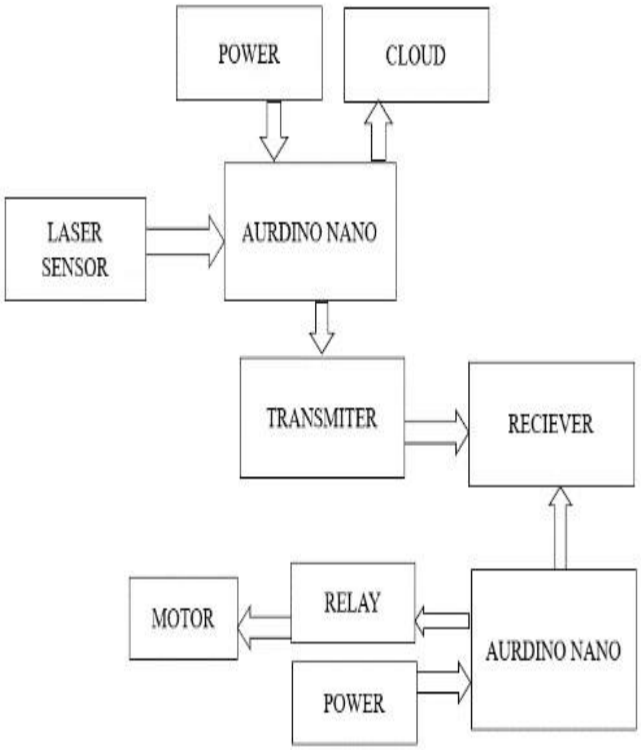


Figure11:Block diagram of proposed system

Chapter 6

Results and Discussion

This project helps detect the water flow and wastage of water and to conserve it clean and safe.

CONCLUSION :

Using sensor technology, provides online monitoring which helps in managing water resources more efficiently. IoT based automated water distribution and performance monitoring system focuses on various entities such as proper supply, over consumption alert and water quality assurance. IoT domain combines with cloud computing and Machine learning algorithm, analyze water quality, water flow pattern and make alert system to avoid wastage of water. Our literature review paper has summarized the work carried out by various researchers related to smart water management, associated problems, and their solutions, in the last two decades. Our proposed IOT based Society Water Management with Automatic Bill Generation system aim is to identify the problems and find out real time solutions to make effective smart water system and user has to pay effective cost of water as per their usage.

APPENDIX

SOURCE CODE :

```
#define BLYNK_TEMPLATE_ID "TMPLlcLQu4bQ"
#define BLYNK_TEMPLATE_NAME "water monitor"
#define BLYNK_AUTH_TOKEN "OgvenxCWu9sG7-
9deFGLFCLE4rWCGW7N"

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Wokwi-GUEST";   //WiFi Name
char pass[] = "";   //WiFi Password

//Set Water Level Distance in CM
int emptyTankDistance = 150 ;  //Distance when tank is empty
int fullTankDistance =  40 ;  //Distance when tank is full (must be
greater than 25cm)

//Set trigger value in percentage
int triggerPer =   10 ;  //alarm/pump will start when water level drop
below triggerPer

#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <AceButton.h>
using namespace ace_button;

// Define connections to sensor
#define TRIGPIN    27  //D6
#define ECHOPIN    26  //D7
#define wifiLed    2  //D0
#define BuzzerPin  13  //D3
#define RelayPin   14 //D5
#define ButtonPin1 12   //RX   //Mode
#define ButtonPin2 33  //SD3  //Relay
```

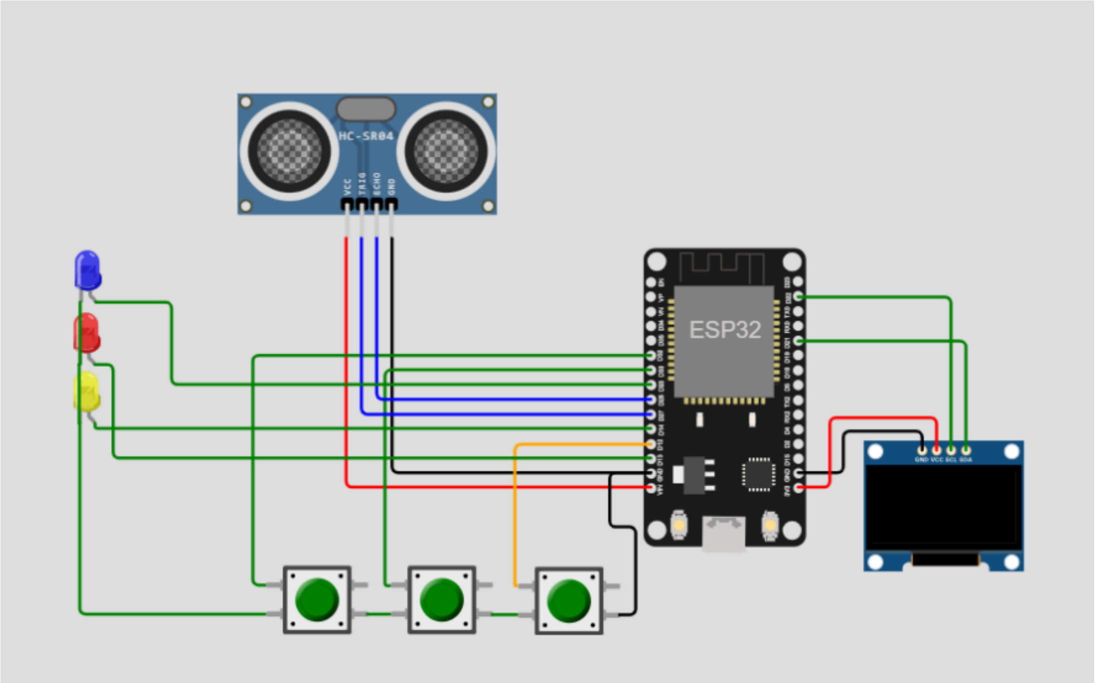
```
#define ButtonPin3 32  //D4   //STOP Buzzer
#define fullpin    25

//Change the virtual pins according the rooms
#define VPIN_BUTTON_1    V1
#define VPIN_BUTTON_2    V2
#define VPIN_BUTTON_3    V3
#define VPIN_BUTTON_4    V4
#define VPIN_BUTTON_5    V5

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL
pins)
#define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino
reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);

float duration;
float distance;
int   waterLevelPer;
bool  toggleBuzzer = HIGH; //Define to remember the toggle state

bool toggleRelay = false; //Define the toggle state for relay
bool modeFlag = true;
bool conection = true;
String currMode;

char auth[] = BLYNK_AUTH_TOKEN;

ButtonConfig config1;
AceButton button1(&config1);
ButtonConfig config2;
AceButton button2(&config2);
ButtonConfig config3;
```

```
AceButton button3(&config3);

void handleEvent1(AceButton*, uint8_t, uint8_t);
void handleEvent2(AceButton*, uint8_t, uint8_t);
void handleEvent3(AceButton*, uint8_t, uint8_t);

BlynkTimer timer;

void checkBlynkStatus() { // called every 3 seconds by SimpleTimer

  bool isconnected = Blynk.connected();
  if (isconnected == false) {
    //Serial.println("Blynk Not Connected");
    digitalWrite(wifiLed, LOW);
    conection = true;

  }
  if (isconnected == true) {
    digitalWrite(wifiLed, HIGH);
    //Serial.println("Blynk Connected");
    conection = false;
  }
}

// When App button is pushed - switch the state

BLYNK_WRITE(VPIN_BUTTON_3) {
  modeFlag = param.asInt();
  if(!modeFlag && toggleRelay){
      digitalWrite(RelayPin, LOW);  //turn off the pump
      toggleRelay = false;
    }
    controlBuzzer(500);
    currMode = modeFlag ? "AUTO" : "MANUAL";
}

BLYNK_WRITE(VPIN_BUTTON_4) {
  if(!modeFlag){
    toggleRelay = param.asInt();
    digitalWrite(RelayPin, toggleRelay);
    controlBuzzer(500);
  }
  else{
    Blynk.virtualWrite(VPIN_BUTTON_4, toggleRelay);
  }
}

BLYNK_WRITE(VPIN_BUTTON_5) {
  toggleBuzzer = param.asInt();
  digitalWrite(BuzzerPin, toggleBuzzer);
}

BLYNK_CONNECTED() {
  Blynk.syncVirtual(VPIN_BUTTON_1);
  Blynk.syncVirtual(VPIN_BUTTON_2);

  Blynk.virtualWrite(VPIN_BUTTON_3, modeFlag);
  Blynk.virtualWrite(VPIN_BUTTON_4, toggleRelay);
  Blynk.virtualWrite(VPIN_BUTTON_5, toggleBuzzer);
}

void displayData(){
  display.clearDisplay();
  display.setTextSize(3);
  display.setCursor(30,0);
  display.print(waterLevelPer);
  display.print(" ");
  display.print("%");
  display.setTextSize(1);
  display.setCursor(0,25);
  display.print(conection ? "OFFLINE" : "ONLINE");
  display.setCursor(60,25);
  display.print(currMode);
  display.setCursor(110,25);
  display.print(toggleRelay ? "! ON" : "OFF");
  display.display();
}

void measureDistance(){
  // Set the trigger pin LOW for 2uS
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2);

  // Set the trigger pin HIGH for 20us to send pulse
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(20);

  // Return the trigger pin to LOW
  digitalWrite(TRIGPIN, LOW);

  // Measure the width of the incoming pulse
  duration = pulseIn(ECHOPIN, HIGH);

  // Determine distance from duration
  // Use 343 metres per second as speed of sound
  // Divide by 1000 as we want millimeters

  distance = ((duration / 2) * 0.343)/10;

  if (distance > (fullTankDistance - 10)  && distance <
emptyTankDistance ){
    waterLevelPer = map((int)distance ,emptyTankDistance,
fullTankDistance, 0, 100);
    Blynk.virtualWrite(VPIN_BUTTON_1, waterLevelPer);
    Blynk.virtualWrite(VPIN_BUTTON_2, (String(distance) + " cm"));

    // Print result to serial monitor
//    Serial.print("Distance: ");
//    Serial.print(distance);
//    Serial.println(" cm");
```

```
  if (waterLevelPer < triggerPer){

    if(modeFlag){
      if(!toggleRelay){
        controlBuzzer(500);
        digitalWrite(RelayPin, HIGH); //turn on relay
        toggleRelay = true;
        Blynk.virtualWrite(VPIN_BUTTON_4, toggleRelay);
      }
    }
    else{
      if (toggleBuzzer == HIGH){
        digitalWrite(BuzzerPin, HIGH);
        Serial.println(" BuzzerPin high");
      }
    }
  }

  if (distance < fullTankDistance){
    digitalWrite(fullpin, HIGH);
    if(modeFlag){
      if(toggleRelay){
        digitalWrite(RelayPin, LOW); //turn off relay

        toggleRelay = false;
        Blynk.virtualWrite(VPIN_BUTTON_4, toggleRelay);
        controlBuzzer(500);
      }
    }
    else{
      if (toggleBuzzer == HIGH){
      digitalWrite(BuzzerPin, HIGH);
      }
    }
  }
  if (distance > (fullTankDistance + 5) && waterLevelPer > (triggerPer
+ 5)){
      toggleBuzzer = HIGH;
      Blynk.virtualWrite(VPIN_BUTTON_5, toggleBuzzer);
      digitalWrite(BuzzerPin, LOW);
    }
    if (distance = fullTankDistance){
    Serial.println(" udh bang ");


    }
  }
  displayData();
  delay(100);
}

void controlBuzzer(int duration){
  digitalWrite(BuzzerPin, HIGH);
  Serial.println(" BuzzerPin HIT");
  delay(duration);
  digitalWrite(BuzzerPin, LOW);
}

void setup() {
  // Set up serial monitor
  Serial.begin(9600);

  // Set pinmodes for sensor connections
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
  pinMode(wifiLed, OUTPUT);
  pinMode(RelayPin, OUTPUT);
  pinMode(BuzzerPin, OUTPUT);
  pinMode(fullpin, OUTPUT);

  pinMode(ButtonPin1, INPUT_PULLUP);
  pinMode(ButtonPin2, INPUT_PULLUP);
  pinMode(ButtonPin3, INPUT_PULLUP);
  digitalWrite(wifiLed, HIGH);
  digitalWrite(RelayPin, LOW);
  digitalWrite(BuzzerPin, LOW);

  config1.setEventHandler(button1Handler);
  config2.setEventHandler(button2Handler);
  config3.setEventHandler(button3Handler);

  button1.init(ButtonPin1);
  button2.init(ButtonPin2);
  button3.init(ButtonPin3);

  currMode = modeFlag ? "AUTO" : "MANUAL";

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(1000);
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.clearDisplay();

  WiFi.begin(ssid, pass);
  timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is connected every 2 seconds
  timer.setInterval(1000L,  measureDistance); // measure water level every 1 seconds
  Blynk.config(auth);
  delay(1000);

  Blynk.virtualWrite(VPIN_BUTTON_3, modeFlag);
  Blynk.virtualWrite(VPIN_BUTTON_4, toggleRelay);
  Blynk.virtualWrite(VPIN_BUTTON_5, toggleBuzzer);

  delay(500);
}
```

```
 void loop() {

  Blynk.run();
  timer.run(); // Initiates SimpleTimer

  button1.check(); //mode change
  button3.check(); //buzzer reset

  if(!modeFlag){  //if in manual mode
    button2.check();
  }

}
void button1Handler(AceButton* button, uint8_t eventType, uint8_t
buttonState) {
  Serial.println("EVENT1");
  switch (eventType) {
    case AceButton::kEventReleased:
      //Serial.println("kEventReleased");
      if(modeFlag && toggleRelay){
        digitalWrite(RelayPin, LOW);  //turn off the pump
        toggleRelay = false;
        controlBuzzer(500);
      }
      modeFlag = !modeFlag;
      currMode = modeFlag ? "AUTO" : "MANUAL";
      Blynk.virtualWrite(VPIN_BUTTON_3, modeFlag);
      controlBuzzer(200);
      break;
  }
}

void button2Handler(AceButton* button, uint8_t eventType, uint8_t
buttonState) {
  Serial.println("EVENT2");
  switch (eventType) {
    case AceButton::kEventReleased:
      //Serial.println("kEventReleased");
      if(toggleRelay){
        digitalWrite(RelayPin, LOW);  //turn off the pump
        toggleRelay = false;
      }
      else{
        digitalWrite(RelayPin, HIGH);  //turn on the pump
        toggleRelay = true;
      }
      Blynk.virtualWrite(VPIN_BUTTON_4, toggleRelay);
      controlBuzzer(500);
      delay(1000);
      break;
  }
}

void button3Handler(AceButton* button, uint8_t eventType, uint8_t
buttonState) {
  Serial.println("EVENT3");
  switch (eventType) {
    case AceButton::kEventReleased:
      //Serial.println("kEventReleased");
      digitalWrite(BuzzerPin, LOW);
      toggleBuzzer = LOW;
      Blynk.virtualWrite(VPIN_BUTTON_5, toggleBuzzer);
      break;
  }
}
```

SIMULATION PROCESS:



REFERENCES :

[1] Michel R. Machado, Tiago Ribas Júnior, Michele R. Silva, João B. Martins; “Smart Water Management System using Microcontroller ZR16S08 as IoT Solution”, 2019 IEEE 10th Latin American Symposium on Circuits & Systems (LASCAS),IEEE 18 March 2019

[2] Kaushik Gupta, Mandar Kulkarni, Manas Magdum, Yash Baldawa, Prof. Shivprasad Patil; “Smart Water Management in Housing Societies using IoT”; 2nd International Conference on Inventive Communication and Computational Technologies, April 2018

[3] Chanda Rajurkar, S R S Prabaharan, S. Muthulakshmi, “IoT based water management”, IEEE 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2), 16 October 2017

[4] Thinagaran Perumal, Md Nasir Sulaiman, Leong.C.Y; “Internet of Things (IoT) Enabled Water Monitoring System”, 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), 04 February 2016

[5] Muhannad Al-Jabi Haya Sammaneh IoT-Enabled Adaptive Smart Water Distribution Management System 2019 International Conference on Promising Electronic Technologies (ICPET)

[6] Monira Mukta, Samia Islam, Surajit Das Barman, Ahmed Wasif Reza, M Saddam Hossain Khan IoT based Smart Water Quality Monitoring System 2019 IEEE 4th International Conference on Computer and Communication Systems

[7] 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC) GLA University, Mathura, UP, India. Feb 28-29, 2020 IoT and Cloud Computing based Smart Water Metering System

.