Water Tank Monitoring & Pump Controlling System Project Report

SUBMITTED BY:

- KIC-HNDCSAI-Y2-231-F-002
- KIC-HNDCSAI-Y2-231-F-013
- KIC-HNDCSAI-Y2-231-F-011

INTERNET OF THINGS

Date:

Start : 2023.09.01 – End:2023.09.12

Higher National Diploma In

Computer Science With Artificial Intelligence.

NIBM

KANDY INNOVATION CENTER

ABSTARCT

This project refers to an automatic water tank system, That was named “water tank monitoring and pump control system”. It enables efficient water level management through Arduino and ESP32 microcontrollers, ultrasonic sensors, LEDs, relays and Wi-Fi connectivity. The system continuously measures the water level in the tank using ultrasonic sensors and uses LEDs to inform the user of the current water level.

So what happen is the module has been programmed to constantly trigger and echo the Ultrasonic sensor on the tank to measure the distance to the water level. For each water level the module programmed to lit up the LEDs to notify the owner about the water level. So when the the water level reach to the lowest level, it triggers the turn on the pump condition in the module program. When that happens it send a signal to the relay to start the the pump. After the water level reach to the maxium level, it triggers the turn off condition in the program and send a signal to the relay to stop the pump. Also this module connected to a cloud computer using Blynk service. Because of that you can monitor the tank real time water level from anywhere at anytime. Also you can manually control the water pump too. For now this module is programmed to a 13cm height water tank but we can reprogram it for a higher tall tank too

Table of Contents:

INTRODUCTION

Water has been an essential resource since the beginning of the biosphere till today. Water level monitoring is necessary in all water bodies in the country, agricultural lands, reservoirs etc. We have implemented an automatic water tank management system to solve such problems in the country.

It is made to be easy to use by all people using advanced technologies like Arduino, ESP32, relay, ultrasonic sensor, LEDs, and Wi-Fi. The system is designed to be effective in monitoring and managing the water level in the tank and to take automatic solutions. Effectively designed to monitor and manage real-time information about water levels in the tank and take automated solutions. User can control the water through remote control.

The primary objectives of this project are as follows:

1. To design a cost-effective and reliable water tank management system.
2. To develop a user-friendly interface for monitoring water levels remotely.
3. To implement automated control of the water pump based on predefined water level thresholds.
4. To enable adaptability for tanks of varying heights.

This report looks at the hardware and software components of the system and its future developments including its functionality

METHODOLOGY

Approach:

This automatic water tank management system was started from "computer design". The main components of the system and its functions are covered. Mainly it was focus on Real time monitoring and Automated control of the water pump.

All parts are connected and then run through coding.As Blink cloud service is used, the entire system can be controlled by remote control.Used in remote monitoring and controling by blynk cloud service.Documentation was maintained for reference.

Tools and Technologies Used:

Arduino IDE: Used for programming the Arduino microcontroller.

ESP32: Employed as the main controller for wireless communication and remote access.

Ultrasonic Sensor: Utilized for precise distance measurement to determine water levels.

Relay: Used for controlling the water pump based on predefined conditions.

LEDs: Provided visual notifications to indicate water levels.

Power Pack: Supplied power to the components reliably.

Wi-Fi Connectivity: Enabled communication with the Blynk cloud service.

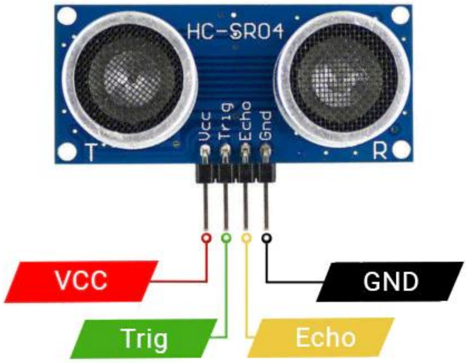
Blynk Service: Used for remote monitoring and control through a user-friendly mobile app.

Project Components

- Arduino for programming
- ESP32 microcontroller
- Ultrasonic sensor for water level measurement
- LEDs for visual status indicators
- Relay for pump control
- Wi-Fi connectivity for remote monitoring
- Blynk cloud service for data visualization and control

❖ ESP32 microcontroller

❖ Ultrasonic sensor



System Functionality

Distance Measurement: The water level is continuously monitored by ultrasonic sensor. Its distance measurements are accurately tracked and recorded.

Different colored LEDs are used to illuminate the gauges to provide individual readings

1. LED1: Low water level
2. LED2: Medium-low water level
3. LED3: Medium-high water level
4. LED4: High water level

Pump control: When the water level reaches a low level, the system triggers the pump to run by sending a signal to the relay. The pump remains active until the water level reaches the maximum level, at which point the system sends a signal to the relay to turn the pump off.

Remote Monitoring: Remote monitoring is working with Blynk cloud service.it was the big advantage of system

Users can manually control the water pump through the Blynk app.

Hardware Setup

- **Ultrasonic Sensor**: Connected to Arduino pins trig and echo for distance measurement.

An ultrasonic sensor is a distance measuring device that uses ultrasonic waves to measure distance by sending and receiving ultrasonic waves.

Ultra sonic sensor features:

Output voltage: 5V (DC).

Out current: 15mA.

Modulation frequency: 40Hz.

Output: 0 – 5V (Output high when obstacle detected in range).

Beam Angle: Max 15 degrees.

Distance: 2cm – 400cm.

Accuracy: 0.3cm.

Communication: Positive TTL pulse.

- LEDs: Indicators for water level status, connected to pins LED1, LED2, LED3, and LED4.
- **Relay:** Controls the water pump, connected to pin relay.

- Wi-Fi Connectivity: ESP32 provides Wi-Fi connectivity for remote access.

Code Overview

All mentioned functions are implemented by Arduino code as explained below by Arduino code.

```
// Define the Blynk template ID and authentication token for your Blynk project
#define BLYNK_TEMPLATE_ID "TMPL4zHYlvJkF"
#define BLYNK_TEMPLATE_NAME "Water Level Monitoring System"
#define BLYNK_AUTH_TOKEN "PtkFWsjd2nnP3rQOuvVppocye4xCK8xM"

// Include the necessary library files
#include <Wire.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Define pins for LEDs, ultrasonic sensor, and relay
#define LED1 2
#define LED2 4
#define LED3 5
#define LED4 18
#define trig 12
#define echo 13
#define relay 14

// Define the maximum water level (in CM) for your tank
int MaxLevel = 13;

// Calculate different water level thresholds as a percentage of MaxLevel
```

```
int Level1 = (MaxLevel * 75) / 100;
int Level2 = (MaxLevel * 65) / 100;
int Level3 = (MaxLevel * 55) / 100;
int Level4 = (MaxLevel * 35) / 100;

// Initialize a Blynk timer
BlynkTimer timer;

// Define your Wi-Fi credentials (SSID and password)
char ssid[] = "isurulkh1";
char pass[] = "123456789";

void setup() {
  // Start the serial communication for debugging
  Serial.begin(115200);

  // Initialize Blynk with your authentication token and Wi-Fi credentials
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass, "blynk.cloud", 80);

  // Set pin modes for LEDs, ultrasonic sensor, and relay
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(relay, OUTPUT);
  // Initially, turn off the relay (HIGH means off)
  digitalWrite(relay, HIGH);
}

// Function to get the ultrasonic sensor values
void ultrasonic() {
  // Trigger the ultrasonic sensor to measure distance
  digitalWrite(trig, LOW);
  delayMicroseconds(4);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  // Measure the duration of the echo pulse to calculate distance
  long t = pulseIn(echo, HIGH);
  int distance = t / 29 / 2;

  // Print the distance to Serial for debugging
  Serial.println(distance);

  // Calculate the distance relative to MaxLevel
  int blynkDistance = (distance - MaxLevel) * -1;

  // Send the distance data to the Blynk app
  if (distance <= MaxLevel) {
    Blynk.virtualWrite(V0, blynkDistance);
  } else {
    Blynk.virtualWrite(V0, 0);
```

```
}

// Control LEDs based on water level
if (Level1 <= distance) {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
} else if (Level2 <= distance && Level1 > distance) {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
} else if (Level3 <= distance && Level2 > distance) {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, LOW);
} else if (Level4 <= distance && Level3 > distance) {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
}

// Control the relay (pump) based on water level
if (Level1 < distance) {
  digitalWrite(relay, LOW); // Turn on the pump (LOW means on)
  } else if (Level4 > distance) {
    digitalWrite(relay, HIGH); // Turn off the pump (HIGH means off)
  }
}

// Function to handle manual pump control from the Blynk app
BLYNK_WRITE(V1) {
  bool Relay = param.asInt();
  if (Relay == 1) {
    digitalWrite(relay, LOW); // Turn on the pump (LOW means on)
  } else {
    digitalWrite(relay, HIGH); // Turn off the pump (HIGH means off)
  }
}

void loop() {
  ultrasonic(); // Call the ultrasonic function
  Blynk.run();  // Run the Blynk library to handle Blynk communication
}
```

FUTURE IMPROVEMENTS

- Based on the technique we are currently using, it is only possible to obtain a high water level of 13 cm. In the future, more sophisticated sensors and technology may be used to measure larger tanks.

- And by using more efficient sensors and advanced algorithms, water overflows can be prevented.

- Users can be set to receive specific information via SMS or email.In many complex situations, all information can be viewed and monitored through the phone.

Making these improvements will make water tank system more secure and user-friendly and will lead to more people using it.

CONCLUTION

This project is considered as an excellent project, it is considered as a successful and useful one.This includes remote monitoring and control, which can be a solution to various problems in the water resources sector.