Design lab (EE:396)

Design and Development of an Arduino-based Water Reservoir Monitoring System

Domala Yamini - 210102030

Jupaka Gayathri - 210102041

Abstract: The objective of our project is to design a water level controlling and monitoring system for preserving water using Arduino. There is a lot of wastage of water around the world. Today we need to preserve water at any cost. We can see many houses as overhead tanks and they keep on overflowing water after full tank. It wastes a lot of water as well as electricity. To solve this problem, we are trying to implement a project. In this project, we are going to implement a water level tank controlling system that helps in monitoring the level in the tank.

1 Introduction

To conserve water, we implemented an Arduino-based project that detects the water level continuously. If the water level is above fullThreshold, it turns off the motor pump, and if the water level is below emptyThreshold, it automatically turns on the motor pump.

2 System Architecture

2.1 Key Components: Description and Uses

2.1.1 Microcontroller (Arduino)

Arduino receives signals from the ultrasonic sensor and responds by turning the motor pump OFF and ON based on the threshold values of water levels.

2.1.2 Ultrasonic sensor

The sensor emits ultrasonic waves that travel through the air until they encounter the water surface. Upon hitting the water, the waves are reflected back to the sensor, which calculates the distance based on the time it takes for the waves to return. By continuously measuring the distance between the sensor and the water surface, the sensor can provide real-time data on the water level.

2.1.3 Relay

Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. Arduino cannot control high-voltage and current, but a relay can do this job. There are three configurations of relay: 1) COM = Common Connection 2) NC = Normally Closed Connection 3) NO= Normally Open Connection

2.1.4 OLED Display

OLED is used to display the output. It uses less power than LCDs and is very thin and lightweight, making it ideal for portable devices. OLEDs use fewer layers than LCDs, making them thinner and lighter without sacrificing performance or battery life.

3 Method

The design of this system consists of hardware design and software design. Figure 1 is a block diagram of a water distribution monitoring system design based on Arduino.

3.1 Circuit Diagram

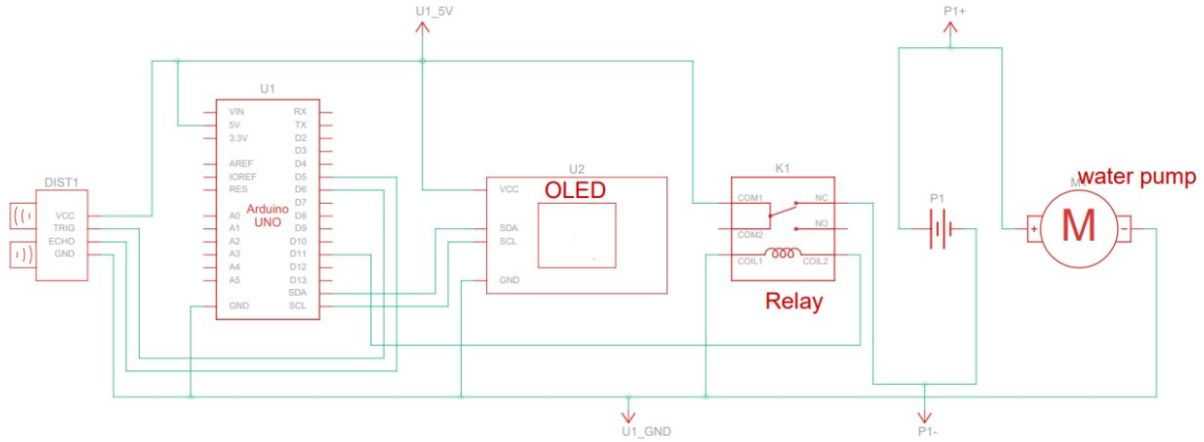


Figure 1: Circuit

3.2 Arduino Code

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
#define OLED_RESET     -1
#define SCREEN_ADDRESS 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
#define NUMFLAKES     10
#define LOGO_HEIGHT   16
#define LOGO_WIDTH    16
#define trigPin 6
#define echoPin 5
const int relayPin = 11;
int emptyThreshold =10 ;
int fullThreshold =20 ;
void displayheight(int a){
```

```
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("H="));
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print(a);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print(F("cm"));
  display.display();
  delay(300);
  }
  void displayfull(){
  display.clearDisplay();
  display.setTextSize(3);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("FULL"));
  display.display();
  delay(300);
  }
  void displayempty(){
  display.clearDisplay();
  display.setTextSize(3);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print(F("EMPTY"));
  display.display();
  delay(300);
  }

void setup() {
  Serial.begin(9600);
    if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
    pinMode(relayPin, OUTPUT);
    // Initially, turn off the relay
    digitalWrite(relayPin, LOW);
    pinMode(trigPin,OUTPUT);
    pinMode(echoPin,INPUT);
}

  display.display();
  delay(500);
 // Clear the buffer
  display.clearDisplay();
 // Draw a single pixel in white
  display.drawPixel(10, 10, SSD1306_WHITE);
  display.display();
  delay(5000);
}
```

```
void loop() {
long duration, distance ,h ;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
Serial.println(distance);
delay(500);
h = 30 - distance; // let 30 be height of tank
Serial.println(h);
if(distance >= 30){
  displayempty();
}
else if(distance <= 5){
 displayfull();
}
else{ displayheight(h);}
  // Check water level and control pump
  if (h < emptyThreshold) {
    // Tank is empty, turn on pump
    digitalWrite(relayPin, HIGH);
    Serial.println("Pump turned ON");
  } else if (h >= fullThreshold) {
    // Tank is full, turn off pump
    digitalWrite(relayPin, LOW);
    Serial.println("Pump turned OFF");
  }
}
```

3.3 Flow chart

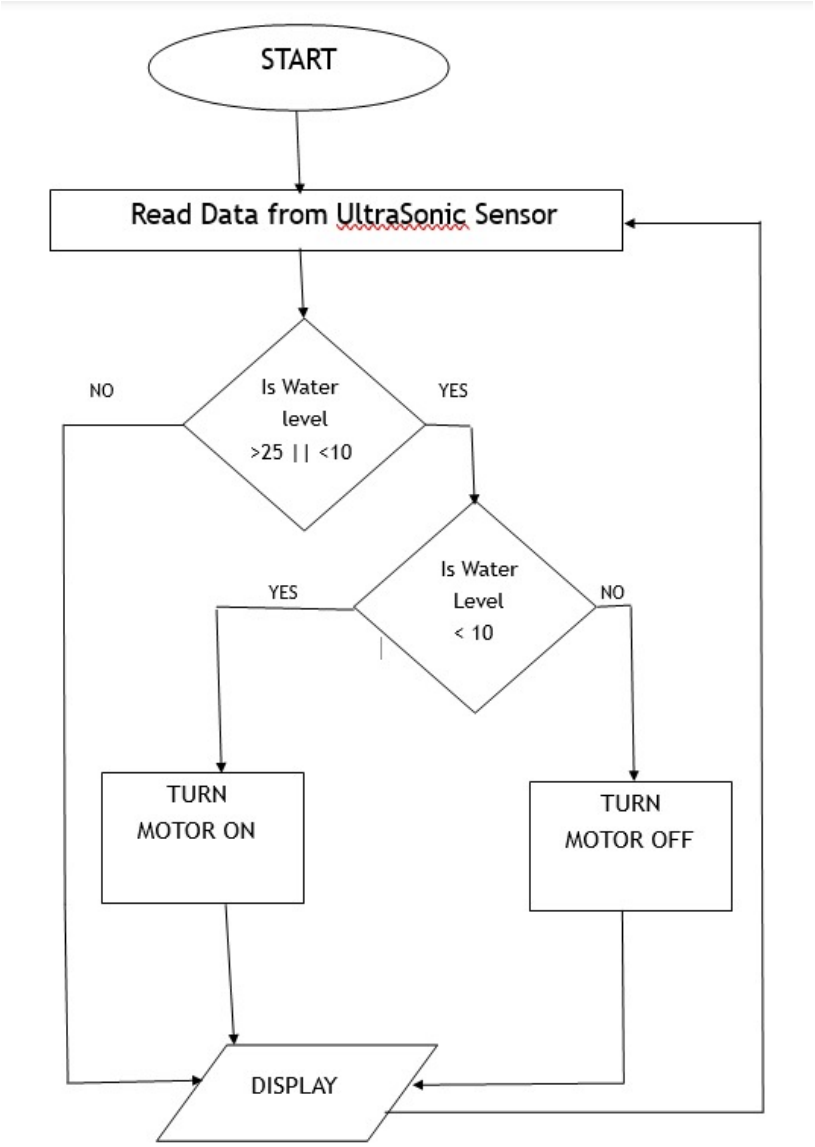
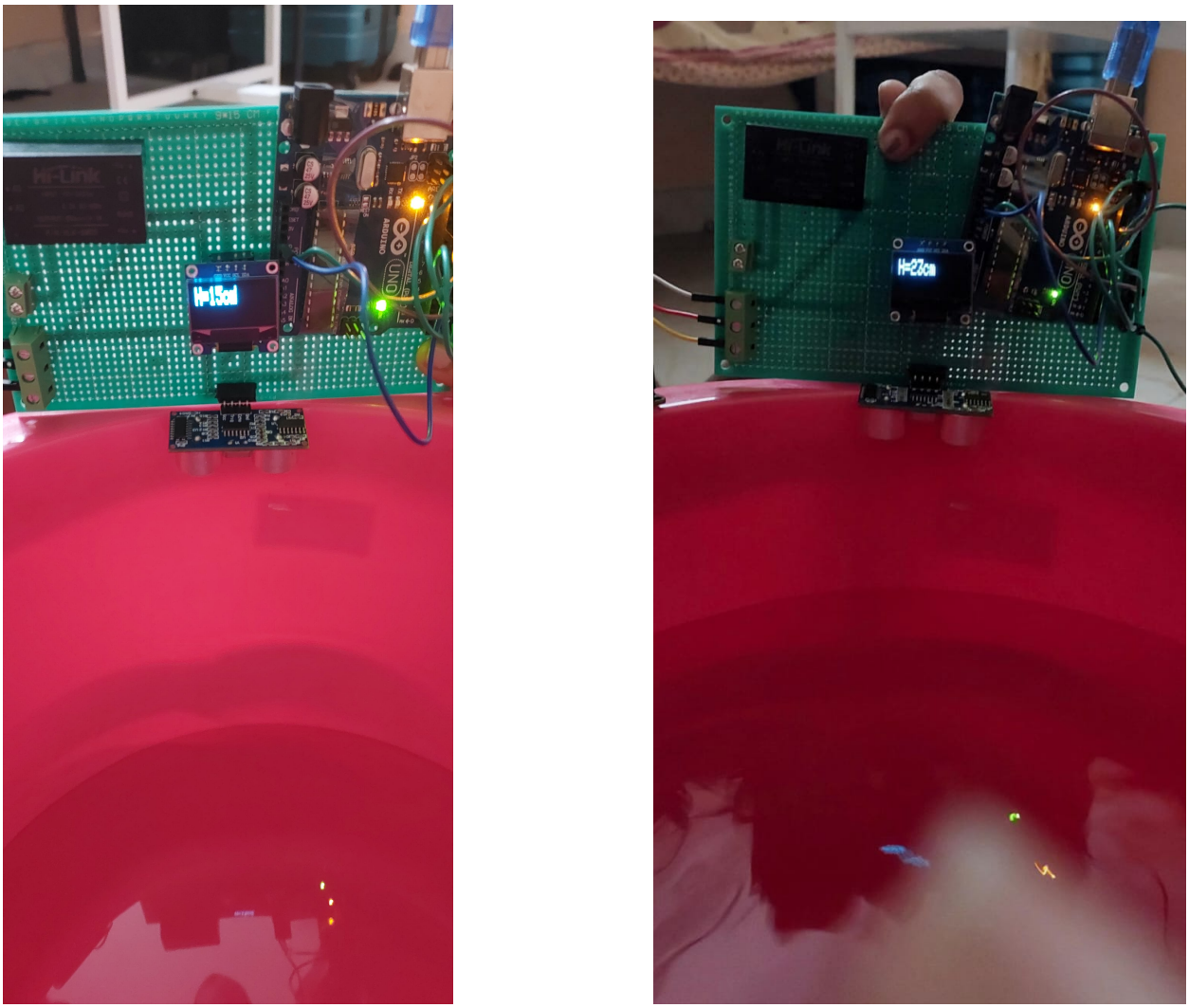


Figure 2: Circuit

4 Experimental Results

The following images show the level of water in the tank on the OLED display.



(a) at 15 cm

(b) at 23cm

Figure 3: output

5 Challenges Faced

- **Sensor Accuracy**: The accuracy of the water level sensor can be affected by factors such as sensor quality, calibration errors, and environmental conditions like temperature and humidity.
-