

UE21CS251B
Microprocessor and Computer Architecture

PROJECT REPORT
Water Level Controller

Team Members:

1. Gayathri Manoj - PES1UG21CS904
2. R. S. Moumitha - PES1UG21CS908
3. Sai Manasa Nadimpalli - PES1UG21CS910
4. Shreya Kashyap - PES1UG21CS913

Water Level Controller

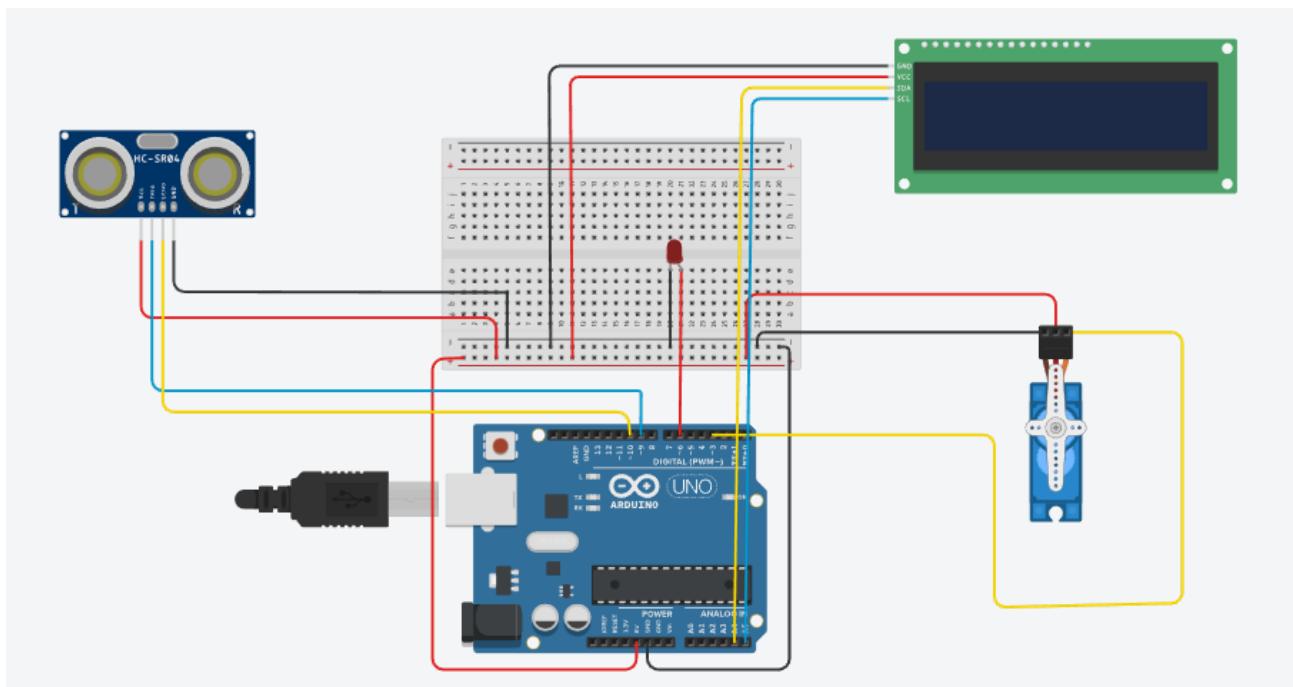
Objective:

To automate home water pump systems that pump water from the sump to overhead tank by using water-level sensor and ultrasonic sensor.

Abstract:

The water pump automation system has been built using an arduino uno microcontroller board. A water level sensor and ultrasonic sensor are installed on the overhead tank to enable detection of water level. A water level threshold is set; when the readings of the sensors fall below the threshold a signal is sent to the motor and the motor is switched on. An LCD module is also installed which allows us to continuously monitor water level readings.

Circuit Diagram:



Source Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
int servoPin = 3;
Servo Servo1;
LiquidCrystal_I2C lcd(0x3F, 16, 2);
```

```
#define sensorPin A0
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distance;

void setup()
{
    Serial.begin(9600);
    lcd.init();
    lcd.backlight();
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    Servo1.attach(sensorPin);
}

void loop()
{
    unsigned int sensorValue = analogRead(sensorPin);
    Serial.println(sensorValue);
    uint8_t outputValue = map(sensorValue, 400, 800, 0, 255);
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculating the distance
    distance = duration * 0.034 / 2;
    Serial.print("Distance: ");
    Serial.println(distance);

    if (sensorValue >= 10)
    {
        Servo1.write(0); // motor switch off
    }
    else if (distance > 4)
    {
        // switch on condition
```

```
Servol.write(0);
delay(500);

// Make servo go to 90 degrees
Servol.write(90);
delay(500);

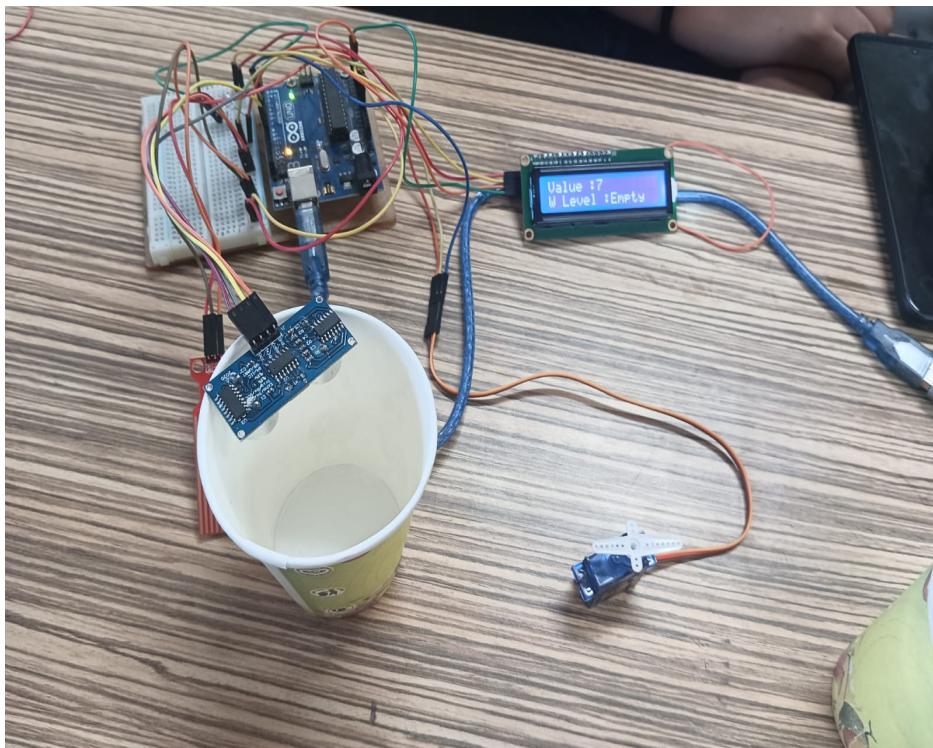
// Make servo go to 180 degrees
Servol.write(180);
delay(500);
}

lcd.setCursor(0, 0);
lcd.print("Value :");
lcd.print(distance);
lcd.print("    ");
lcd.setCursor(0, 1);
lcd.print("          ");
lcd.setCursor(0, 1);
lcd.print("W Level :");

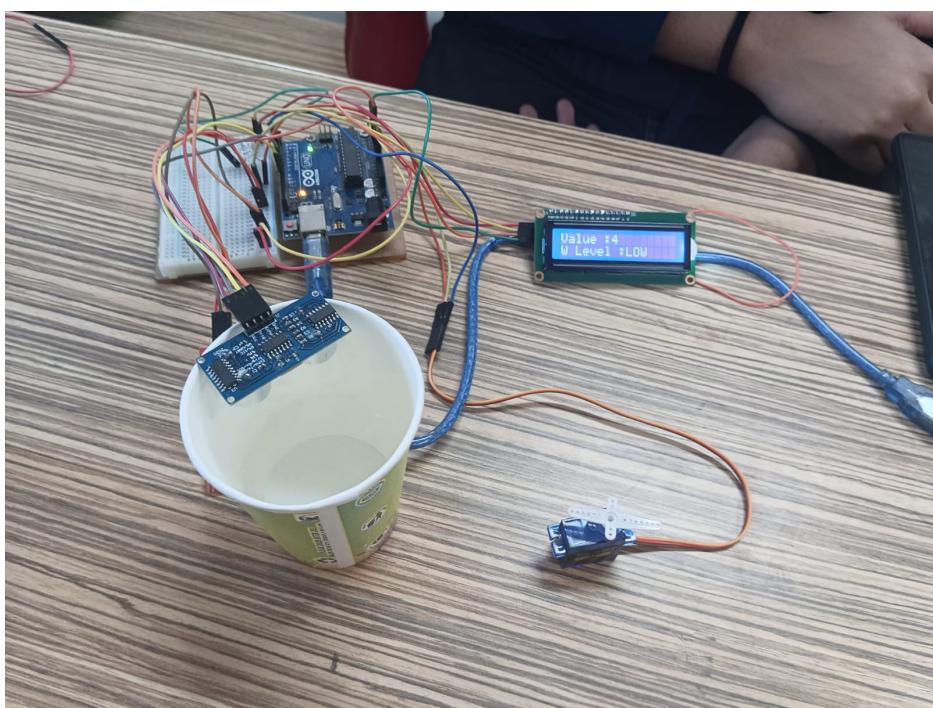
// printing the content on the LCD
if (distance >= 7)
{
    lcd.print("Empty ");
}
else if (distance > 3)
{
    lcd.print("LOW    ");
}
else if (distance <= 3)
{
    lcd.print("HIGH    ");
}
}
```

Output and Demonstration:

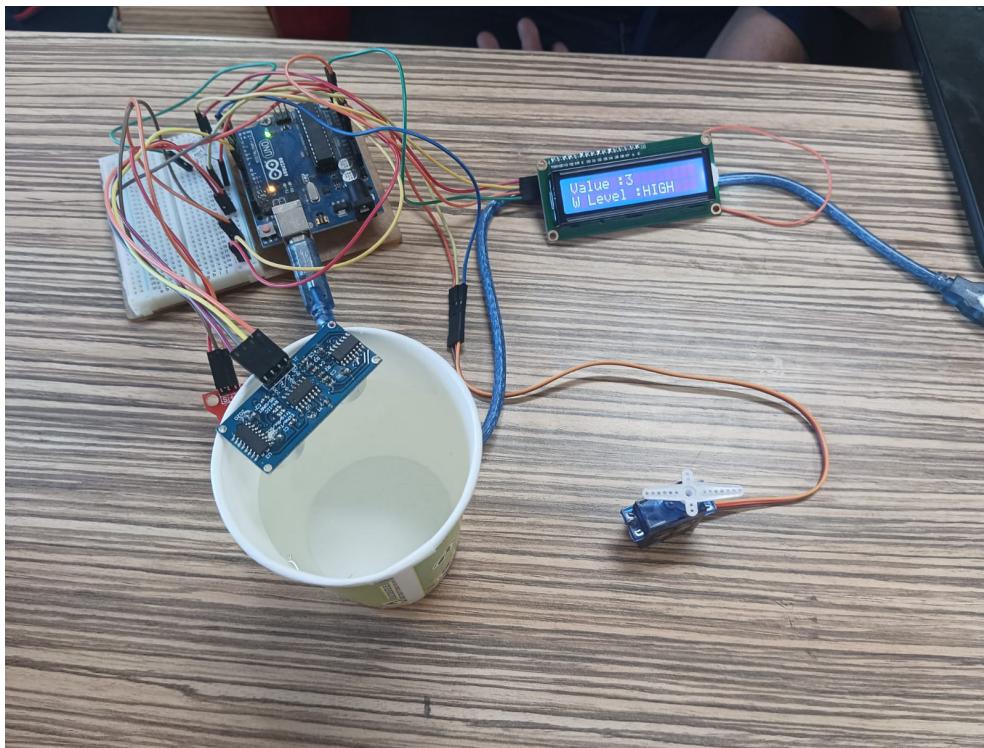
1. When tank is empty: (Servo Motor **SWITCHED ON**)



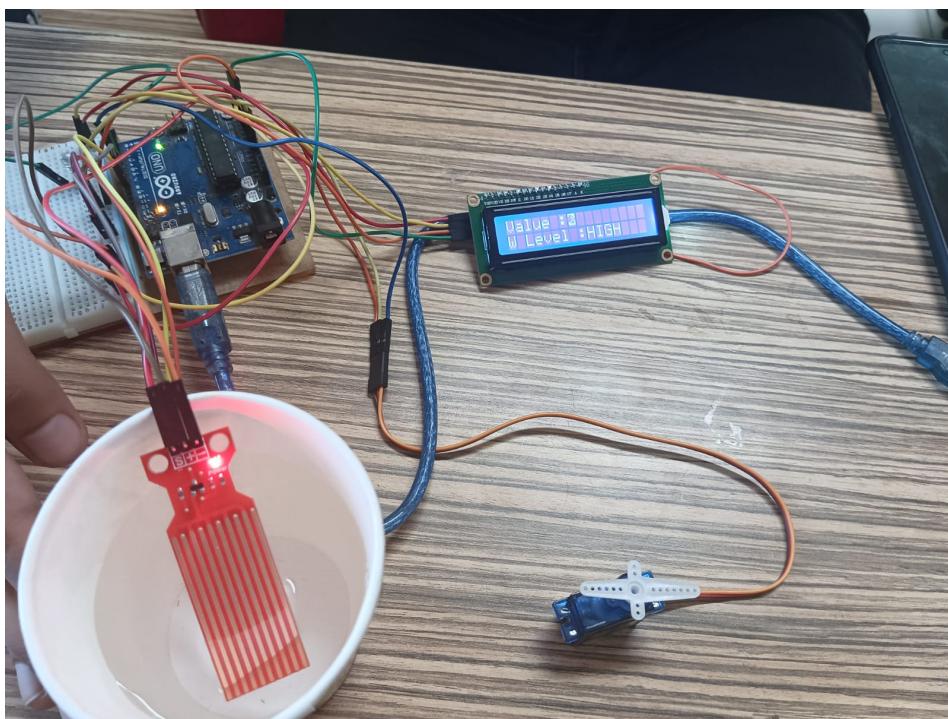
2. When tank water level is LOW: (Servo Motor **SWITCHED ON**)



3. When tank water level is HIGH: (Servo Motor **SWITCHED OFF**)



4. Using Water-level sensor:



Github repository containing source code and dependencies:

<https://github.com/shreya241103/Water-Level-Controller-using-Arduino-Uno>