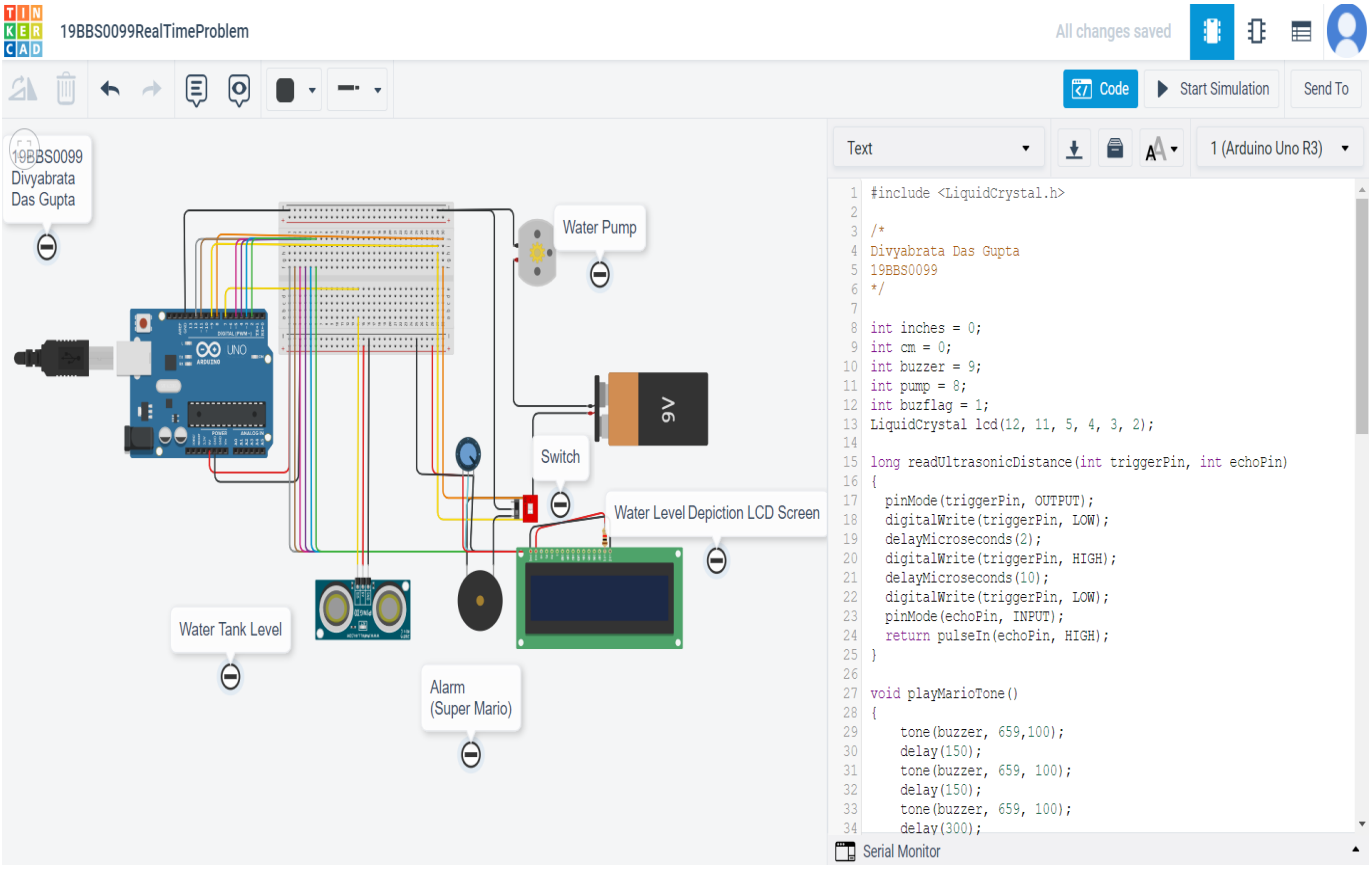
Arduino Board Real-time Problem Experiment:

Title: Underground Water Tank Level Problem

Tinkercad Link: *https://www.tinkercad.com/things/hMLm1yOZEOS-19bbs0099realtimeproblem*

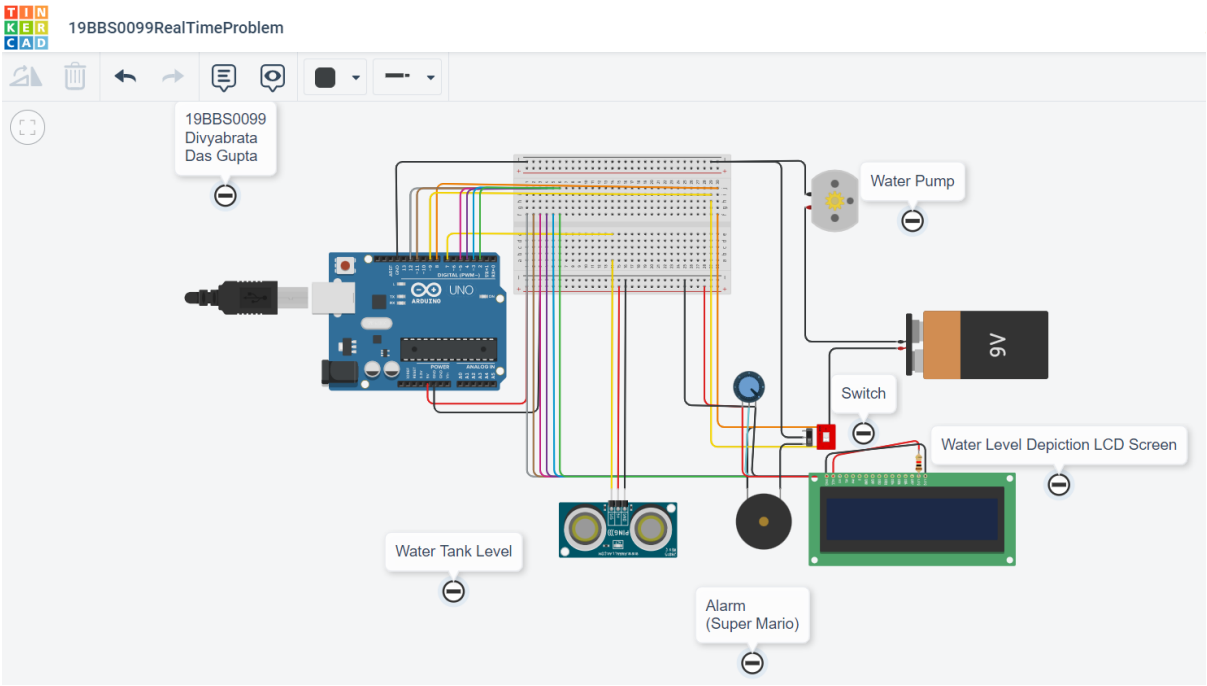
Description: There are several real-time problems that I found in my locality and in my house itself. Some of them are the underground water tank and pump problem, automatic sanitizing of amazon (or other E-commerce services) delivered products, automatic smart street lighting, and so on. All of these problems in my locality can be solved using IoT, but being a kind of village area under Panchayat and with less development, they have not been implemented. I found the first Underground water tank problem quite interesting to solve. It is a very common problem in almost all of our homes where the water pump needs to be manually switched off when the tank is full of water. Without a proper system to inform when the tank is full, this timing has to be mainly guessed. People, like my family, mainly do this by daily experience on how much time it takes the water pump to fill the underground water tank, half an hour, an hour, etc. Many times, people guess the wrong time and either switch the pump off too early, before the tank is full of water, leading to a shortage of water for daily use. Otherwise, it's too late leading to water overflowing from the tank. In fact, sometimes people forget to switch the pump off too which leads to excessive overflow of water outside of the house, into the streets. This leads to wastage of precious water and dirtying of the surroundings of the house. Either way, it's a real-time problem that persists in many houses, mine being one of them. This simple yet troublesome problem can be easily solved using IoT technology and sensors.

The proposed model is as follows, designed in the software Tinkercad:

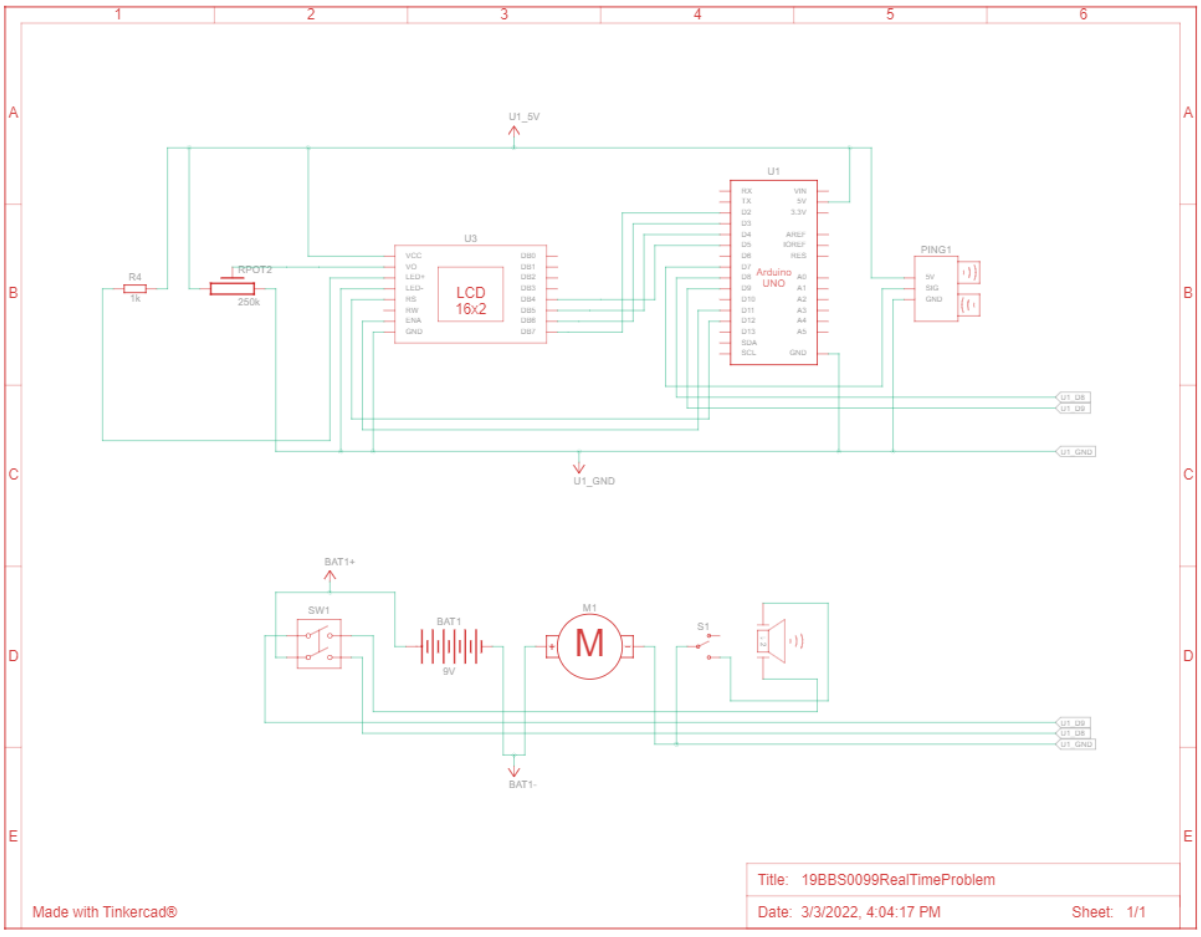



```
#include <LiquidCrystal.h>

/*
Divyabrata Das Gupta
19BBS0099
*/

int inches = 0;
int cm = 0;
int buzzer = 9;
int pump = 8;
int buzflag = 1;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}

void playMarioTone()
{
    tone(buzzer, 659,100);
    delay(150);
    tone(buzzer, 659, 100);
    delay(150);
    tone(buzzer, 659, 100);
    delay(300);
```

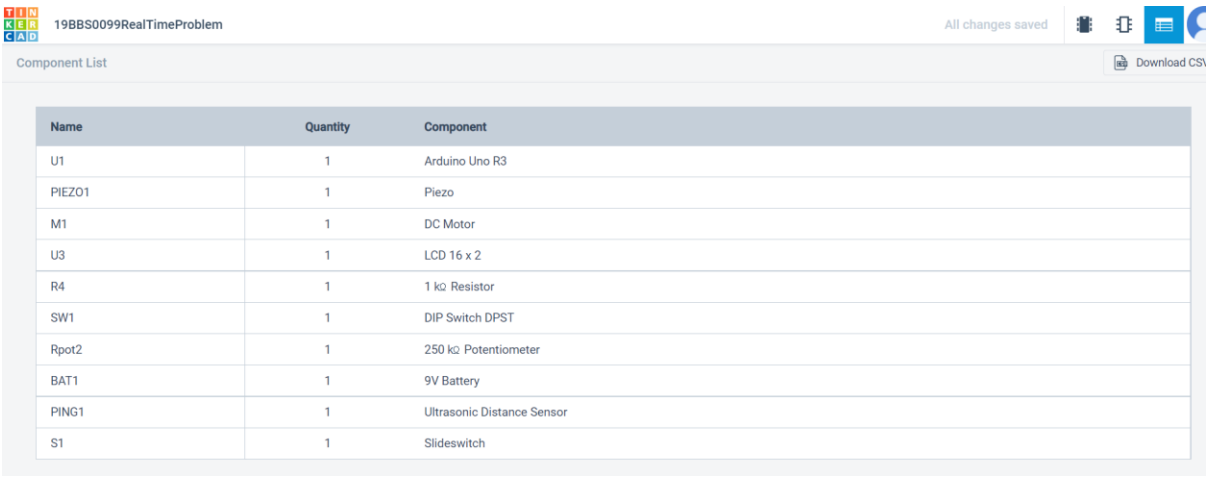
Circuit view:



Schematic view:

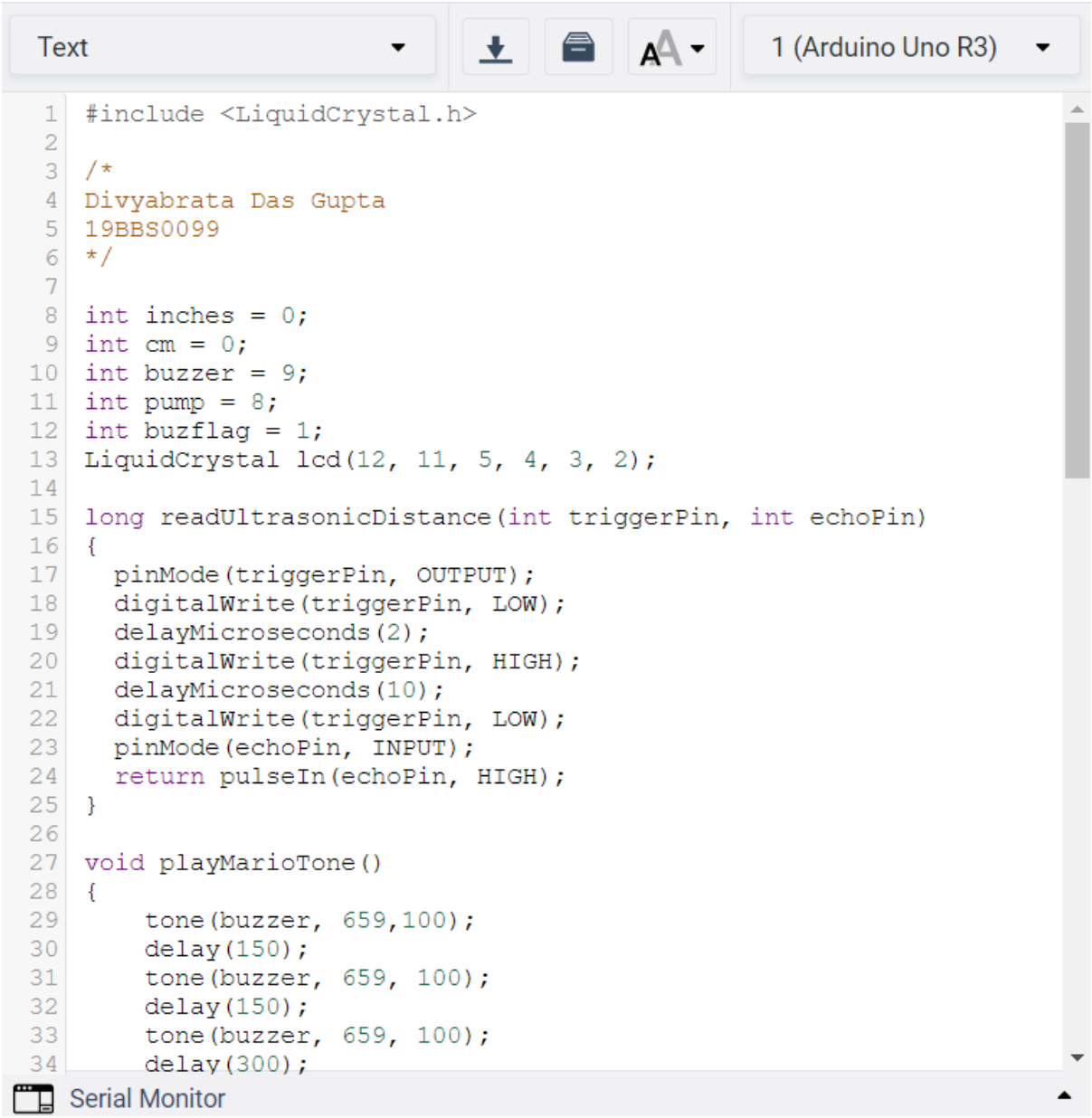


Hardware Component list:



Name	Quantity	Component
U1	1	Arduino Uno R3
PIEZO1	1	Piezo
M1	1	DC Motor
U3	1	LCD 16 x 2
R4	1	1 kΩ Resistor
SW1	1	DIP Switch DPST
Rpot2	1	250 kΩ Potentiometer
BAT1	1	9V Battery
PING1	1	Ultrasonic Distance Sensor
S1	1	Slideswitch

The DC Motor represents the water pump for filling the underground tank. The level of water is constantly measured using the Ultrasonic Distance Sensor situated on the top of the water tank, which measures the distance between itself and the surface of filled water below. This distance is converted into a percentage depicting the amount of water filled inside the tank based on its maximum depth, assumed here to be about 160 inches or 400 cm. This percentage is constantly shown on the LCD screen so that the user can know the amount of water filled inside the tank anytime and determine whether the pump needs to be switched on. When the pump is switched on, simultaneously and buzzer/piezo IoT node is also activated. This is kind of an alarm system in which when the tank is more than 75% filled, which is more than enough water for the day, it sounds an alarm with the Super Mario tune (just for gags) for the user to come to switch off the pump. However, if the user is not alert or present to switch off the pump after the tank is about 90% full, it automatically switches the pump and alarms off. If the switch is left on then the pump again works after the water level drops below 90%. The alarm can be switched on or off separately by user too.

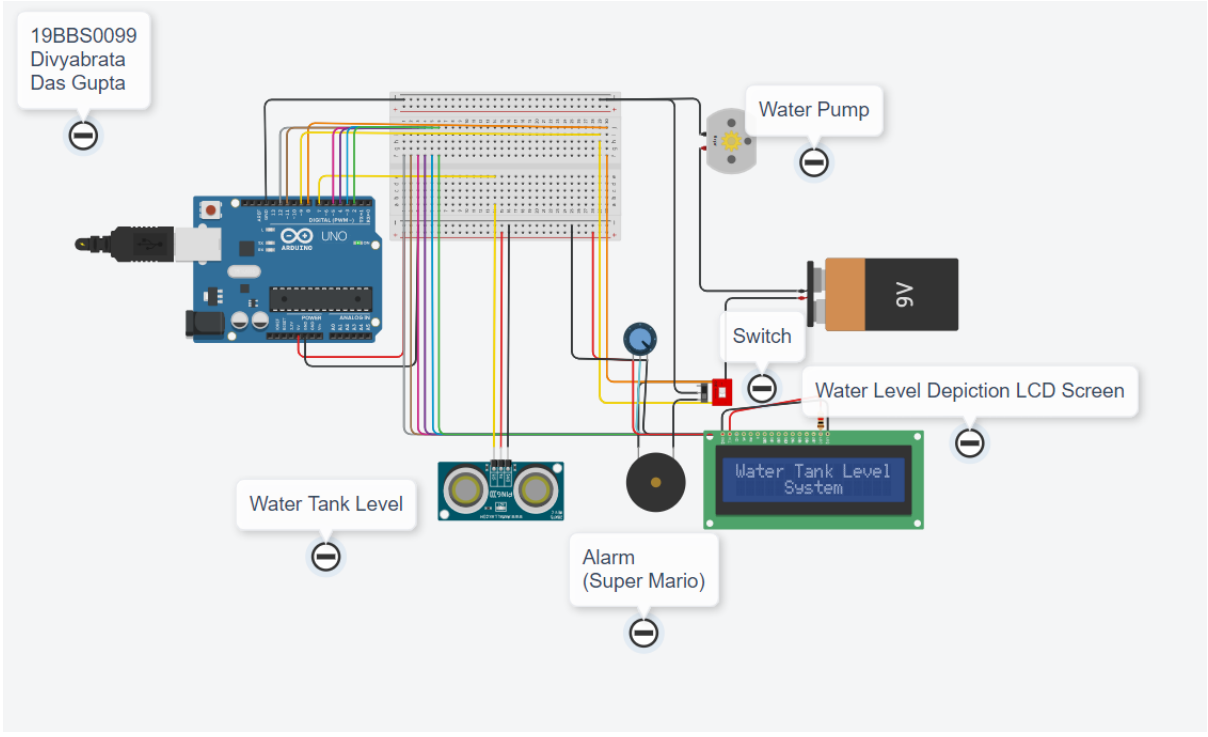



```
#include <LiquidCrystal.h>

/*
Divyabrata Das Gupta
19BBS0099
*/

int inches = 0;
int cm = 0;
int buzzer = 9;
int pump = 8;
int buzflag = 1;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

long readUltrasonicDistance(int triggerPin, int echoPin)
```

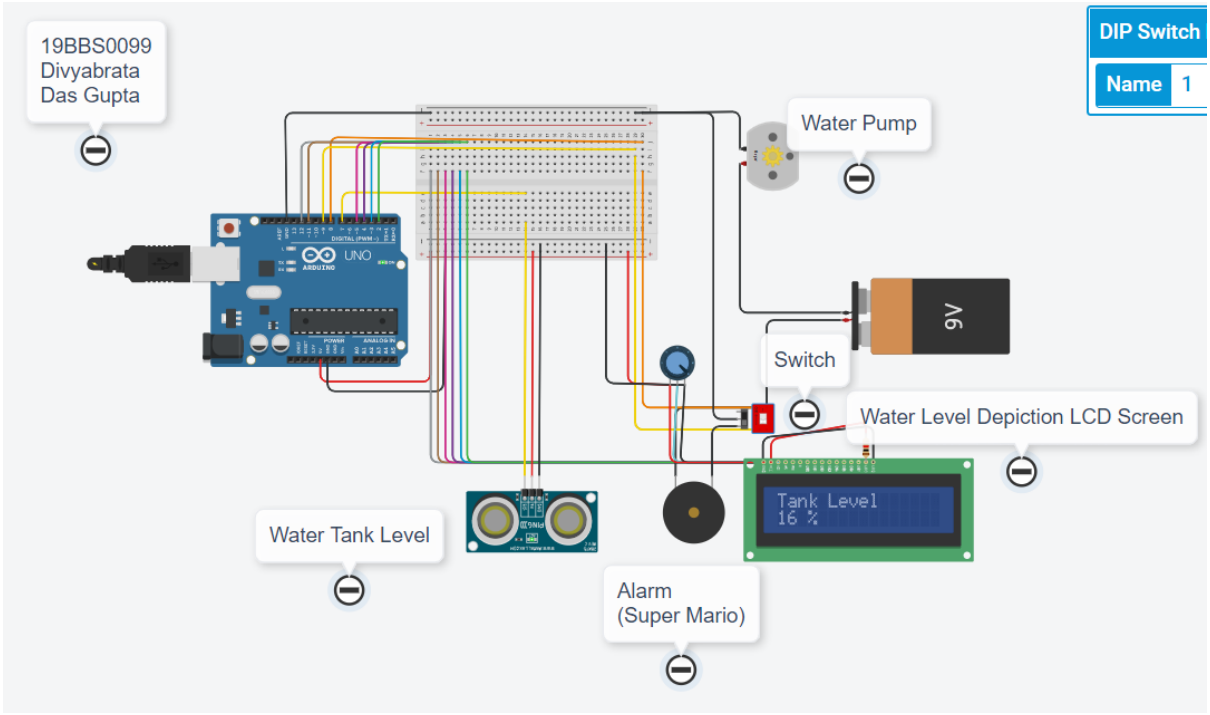
```
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}

void playMarioTone()
{
    tone(buzzer, 659,100);
    delay(150);
    tone(buzzer, 659, 100);
    delay(150);
    tone(buzzer, 659, 100);
    delay(300);
    tone(buzzer, 659, 100);
    delay(300);
    tone(buzzer, 523, 100);
    delay(100);
    tone(buzzer, 659, 100);
    delay(300);
    tone(buzzer, 784, 100);
    delay(550);
    tone(buzzer, 392, 100);
    delay(800);
}

void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  lcd.print("Water Tank Level");
  lcd.setCursor(5,1);
  lcd.print("System");
  delay(1000);
  lcd.clear();

}

void loop()
{
  cm = 0.01723 * readUltrasonicDistance(7, 7);
  int level = map(cm, 400 , 10, 0 , 100);
  lcd.setCursor(0, 0);
  lcd.print( "Tank Level");
  lcd.setCursor(0, 1);
  lcd.print(level);
  lcd.setCursor(3, 1);
```

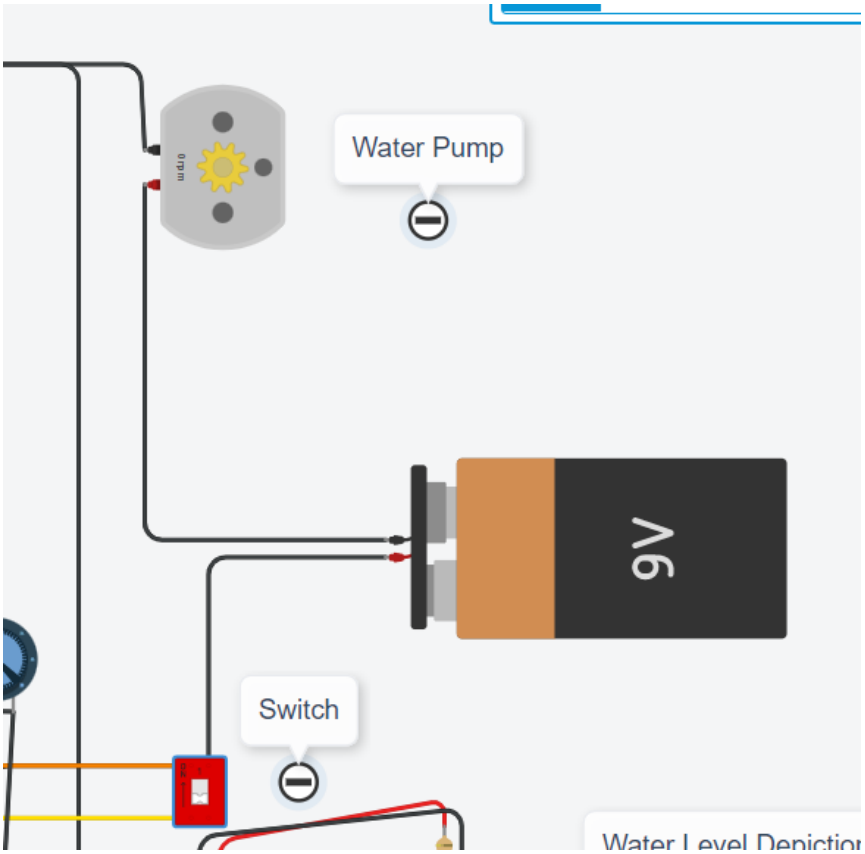
```
  lcd.print("%");
  Serial.print(cm);
  Serial.println("cm");
  if(cm >50){
    digitalWrite(pump, HIGH);
    buzflag=1;
  }
  else{
    digitalWrite(pump, LOW);
    buzflag=0;
  }
  delay(100);
  if((cm < 100)&&(buzflag == 1)){
    playMarioTone();
  }
  else{
    digitalWrite(buzzer,LOW);
    delay(100);
  }

}
```

Simulation:

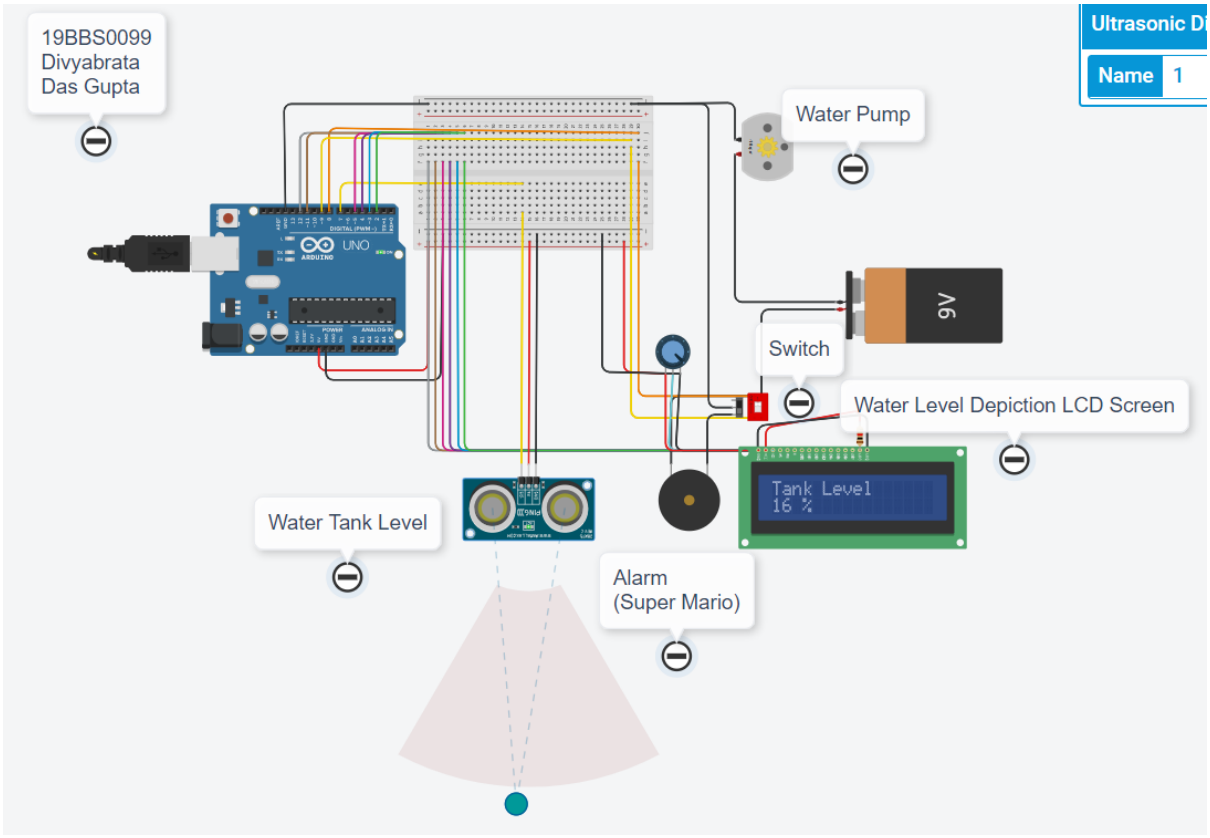
1. 

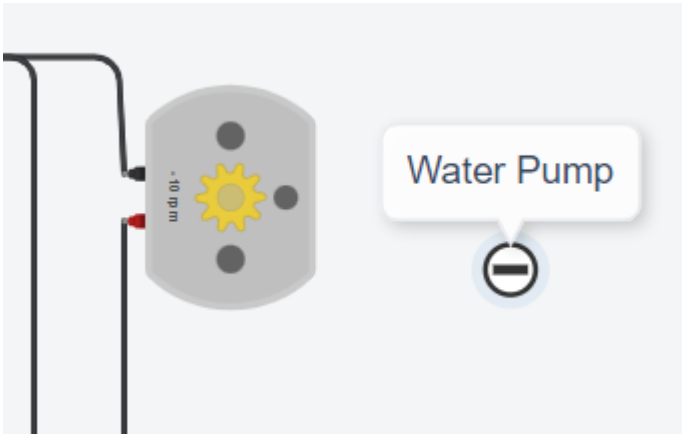
2. Switch off:


19BBS0099
Divyabrata
Das Gupta
DIP Switch
Name 1
Water Pump
9V
Switch
Water Level Depiction LCD Screen
Tank Level
16 %
Water Tank Level
Alarm
(Super Mario)

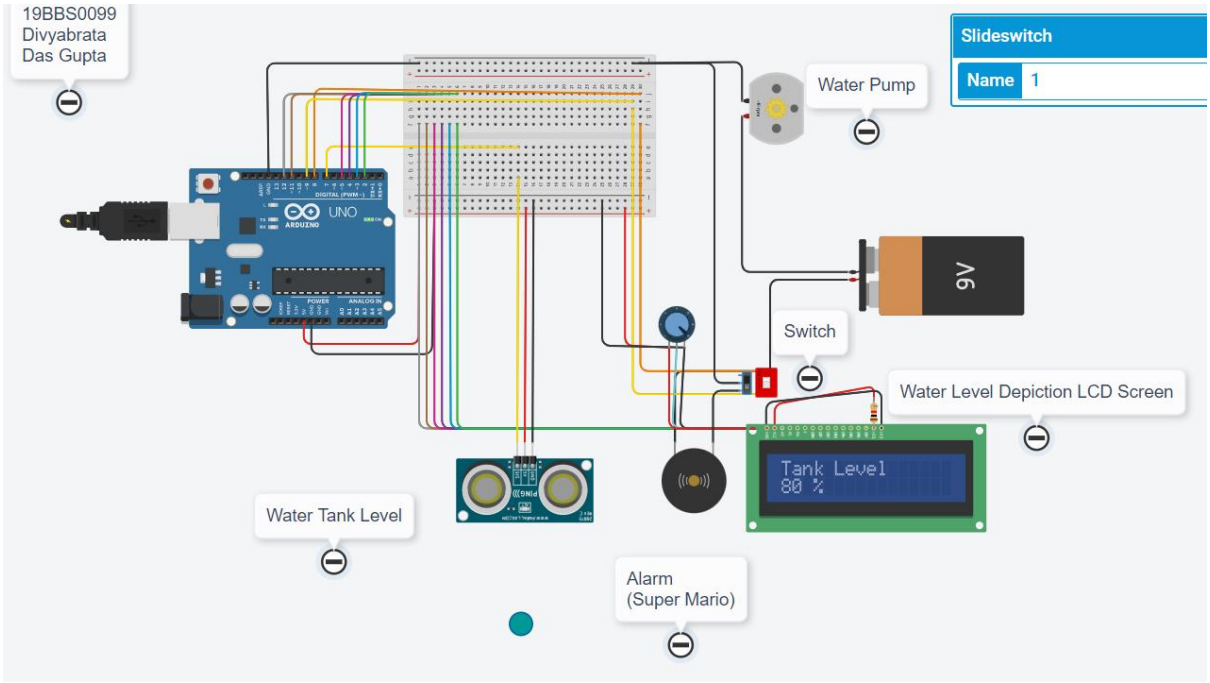

Water Pump
9V
Switch

3. Switch on:

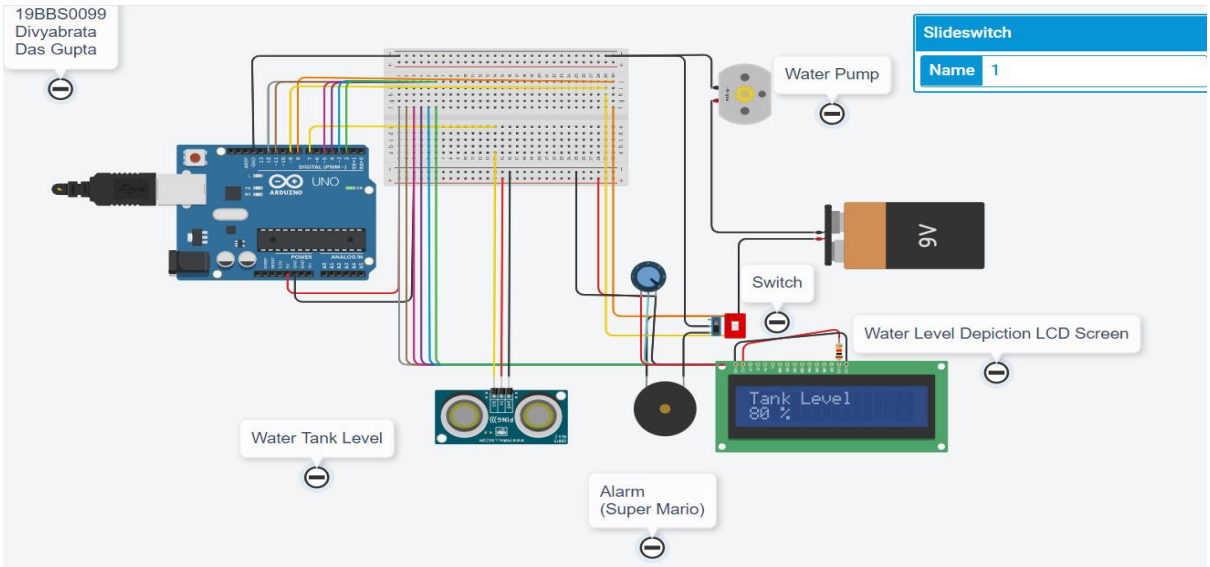




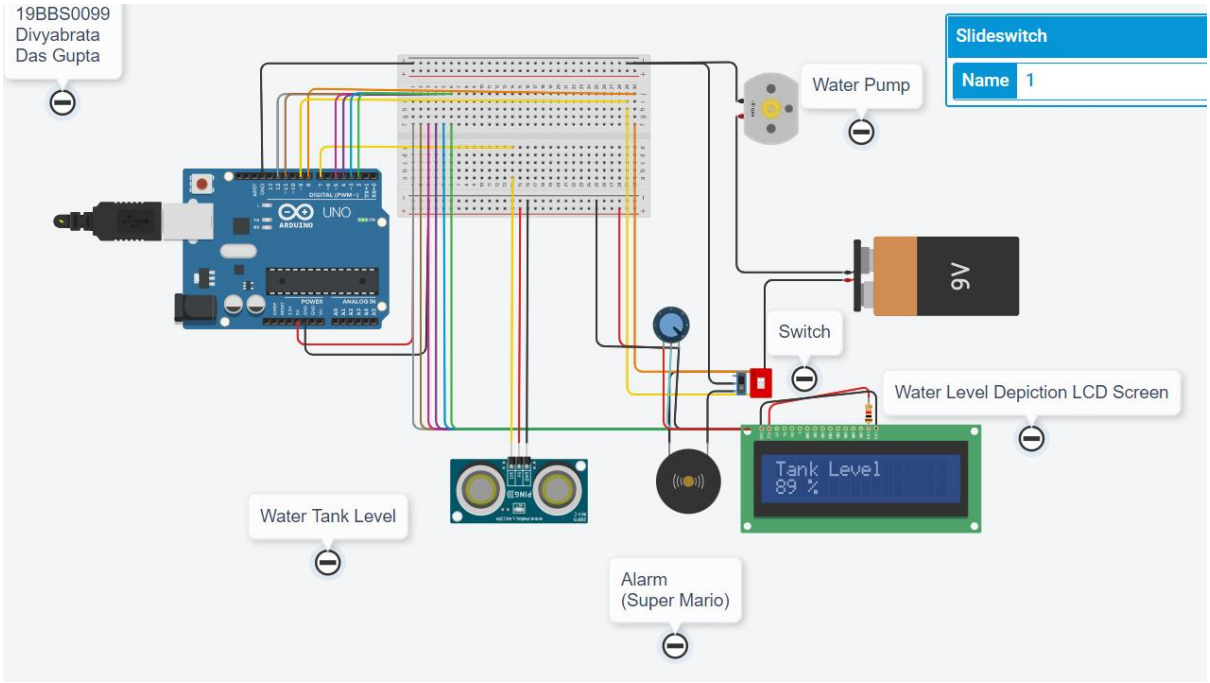
4. Alarm at 80% full:
   Buzzer switch on:

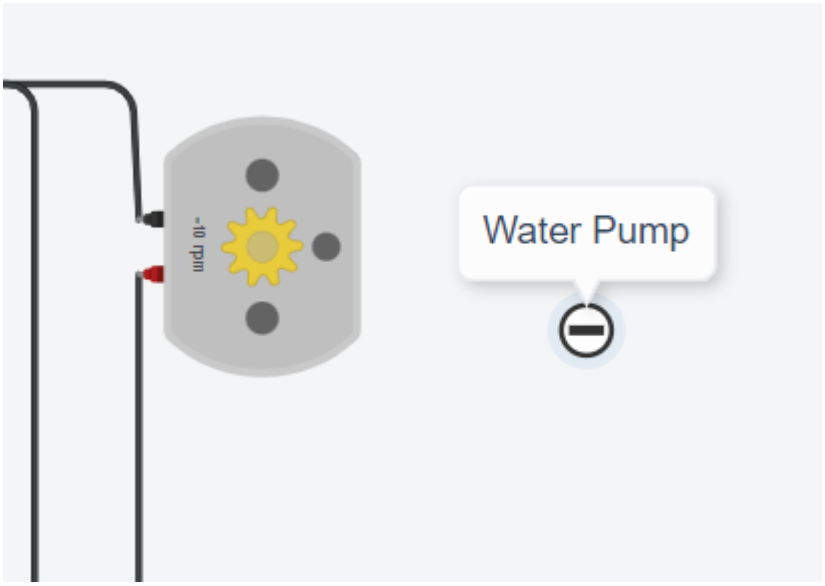


Buzzer switch off:

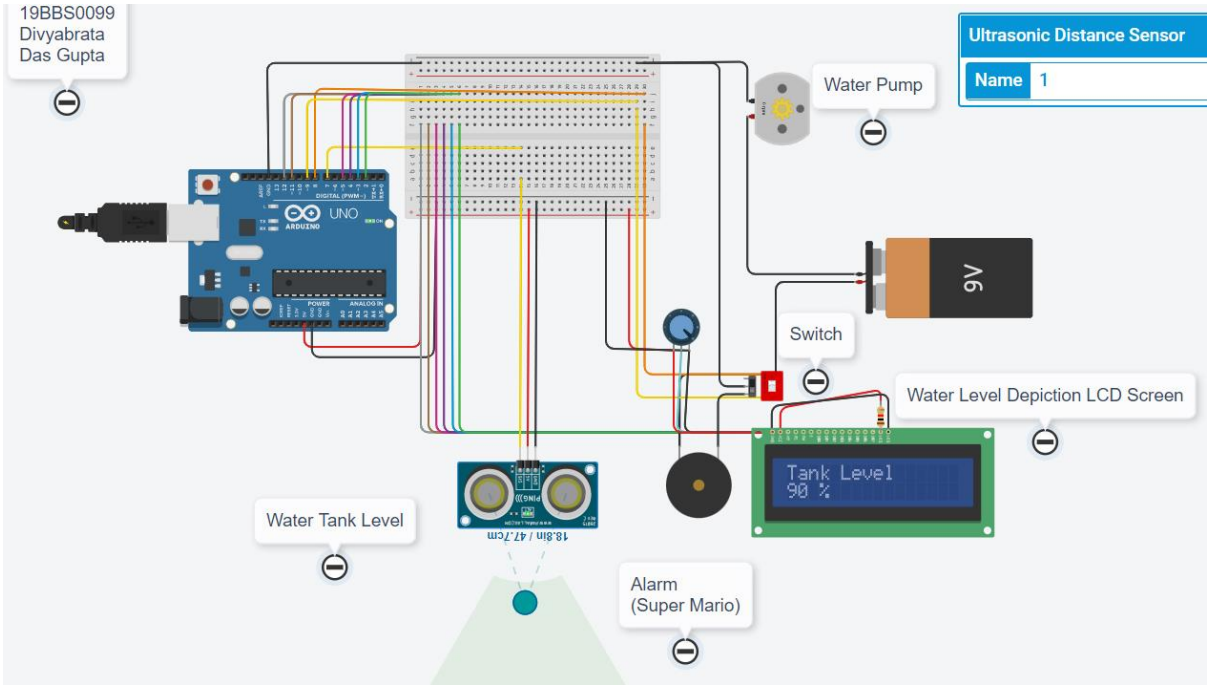


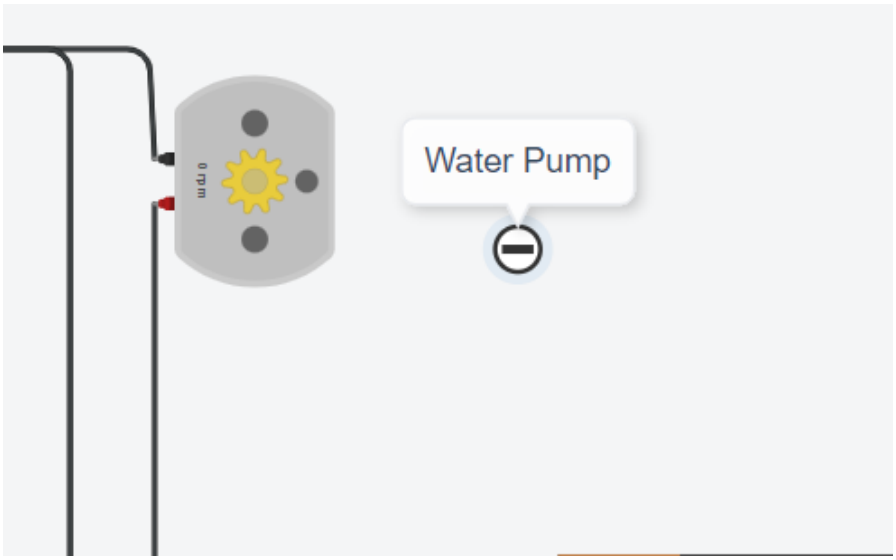
5. Tank at 89% full:





6. Tank at 90% full:





Limitations of this model: There are a few limitations to this model like the alarm system can be a bit irritating after prolonged times of hearing it, though there is a separate switch for it. Also, above a certain limit, about 90%, this system automatically switches off the pump assuming that the water-filled is

enough. Although this limit can be changed as required, user requirements can never be assumed and guessed correctly. Hence the amount of water also is not always enough even if the tank is 90% full. Although this model can be used for the main purpose of switching off the pump at the right time, the percentage of the tank that is full is correctly calculated for only a 400 cm tank, not for tanks of different sizes. Hence the user cannot use the percentage of tank full depicted in the LCD screen to determine correctly when is the right time to start the pump to fill that tank.

Conclusion: Thus, this model is able to solve the problem related to underground water tanks in my house and other houses in my locality. It is able to tell the amount of water present inside the tank and we can easily decide when to start the pump for filling water, say below 50% capacity we decided to switch it on. The water that is filled will gradually raise the level of water inside the tank which will be automatically reflected in the LCD screen using the ultrasonic distance sensor readings. We can switch the pump off at any time we desire, say at 70% capacity. Even if we are careless and exceed that pre-decided capacity, the alarm system will let us know we need to stop the pump. If we want to switch the alarm off and not the pump, then that is also possible and it will not result in overfilling of the tank as the pump is automatically switched off at 90%. We can also switch both the alarm and tank off directly. Lastly, keeping the pump switch on and the alarm switch off, we can use this system for maintaining water at 90% capacity always.

Thank You...!