SMART WATER FOUNTAIN

Problem Definition:

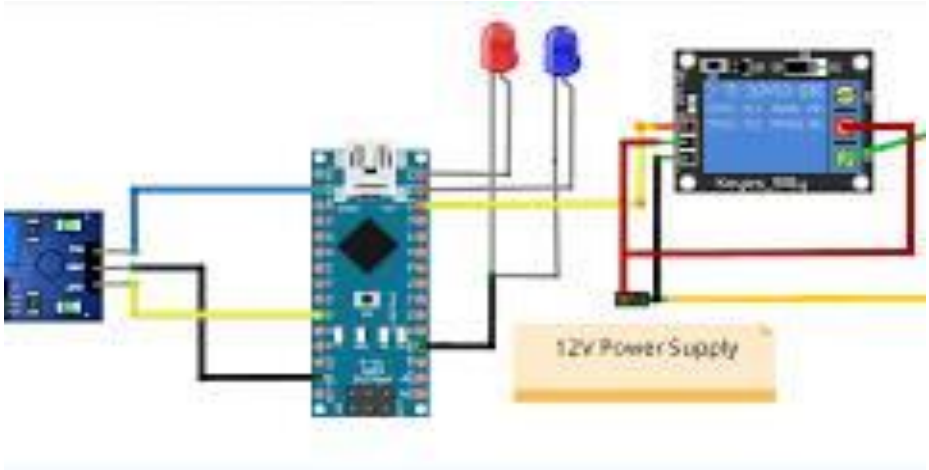
The project aims to enhance public restroom management by installing IoT sensors to monitor occupancy and maintenance needs. The goal is to provide real-time data on restroom availability and cleanliness to the public through a platform or mobile app. This project includes defining objectives, designing the IoT sensor system, developing the restroom information platform, and integrating them using IoT technology and Python.

Design Thinking:

1. **Project Objectives:** Define objectives such as real-time water fountain monitoring, efficient water usage, malfunction detection and resident awareness.
2. **IoT Sensor Design:** Plan the deployment of IoT sensors (e.g., flow rate sensors, pressure sensors) in public water fountains.
3. **Real-Time Transit Information Platform:** Design a mobile app interface that display real-time parking availability to users.

Integration Approach: Determine how IoT sensors will send data to the water fountain status platform.

CIRCUIT AND EXPLANATION:



SMART WATER FOUNTAIN

The **Arduino UNO** is a microcontroller board based on the ATmega328P. It is a popular choice for beginners and hobbyists as it is easy to use and has a wide range of available libraries and tutorials. The board has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button.

A **water pump** that operates at 12V is a type of pump that is designed to run on a 12-volt DC power supply.

The pump typically has two wires, one for power and one for ground. These wires can be connected to the **Arduino**‘s 12V and GND pins, respectively. Once the connections are made, you can use the Arduino’s digital output pins to control the pump. A **relay** is an electronic switch that can be controlled by an **Arduino** microcontroller to turn a **water pump** on and off.

To use a **relay** to control a **water pump** with an Arduino, you will need to connect the relay to the **Arduino**‘s digital output pins. The **relay** typically has three pins, VCC, GND, and IN. VCC should be connected to 5V of the **Arduino**, GND to GND, and IN to a digital pin of the **Arduino**. Once the connections are made, you can use the **Arduino**‘s digitalWrite function to turn the **relay** on and off, and thus control the **water pump**.

The relay is the intermediate component between the Arduino board and the water pump. It allows the Arduino board to control the water pump.

The role of the relay is to start or stop the pump responsible for filling the bottles.

- We connect the (-) terminal of the relay to the GND pin of the Arduino
- We connect the (+) terminal of the relay to the 3.3V pin of the Arduino
- We connect the terminal (S) of the relay to pin N° 2 of the Arduino

the role of the push button is the activation or deactivation of the relay module.

- We connect the first terminal of the button to the GND pin of the Arduino.
- We connect the second terminal of the button to pin N°1 of the Arduino.

SENSOR UNIT

This block contains the four sensors. The data acquired from the sensors will be transmitted to the control unit. Control unit will then have some logic designed to send corresponding signals to control other blocks of the water fountain. At the same time, the display screen on the water

fountain will display the readings along with the determined water quality level and remaining water quantity.

For the PH-value sensor, temperature sensor and conductivity sensor, values will be retrieved and calculated to determine the overall water quality level. When poor water quality is determined, the water replacement procedures will take place. The weight sensor readings will be used to determine the amount of fresh water left in the water tank.

TEMPERATURE SENSOR:

A water-proof temperature sensor is going to be used. Part number from sparkfun is: DS18B20 [6]. This temperature sensor is compatible with a relatively wide range of power supply from 3.0V to 5.5V. The measured temperature ranges from -55 to +125 celsius degrees. Between -10 to + 85 degrees, the accuracy is up to +-0.5 degrees. This sensor can fulfill all requirements needed for this project.

PH-SENSOR:

PH value is a valued indicator of water quality. This PH-sensor[7] works with 5V voltage, which is also compatible with the temperature sensor. It can 6measure the PH value from 0 to 14 with an accuracy of +- 0.1 at the temperature of 25 degrees.

CONDUCTIVITY SENSOR:

Conductivity sensor is also part of the water quality assessment. The input voltage is from

3.0 to 5.0V. The error is small, +-5%F.S. The measurement value ranges from 0 to 20 ms/cm which is enough for water quality monitoring.

LIQUID LEVEL SENSOR:

This sensor [9] is responsible for reflecting how much freshwater is left in the water tank. When the water level is low, fresh water will be pumped to the water tank to ensure the water fountain keeps running with freshwater. This sensor is 0.5 Watts. For water level from 0 to 9 inches, the corresponding sensor outputs readings from 0 to 1.6. From that, the quantity of freshwater left can be determined.

DISPLAY UNIT:

SMART WATER FOUNTAIN

SCREEN:

The screen will be used to display the readings from the sensors in a real-time manner.[10] In addition, other necessary information will also be displayed. As described in the sensor part, the water quality and remaining water quantity will be displayed. The screen will be programmed so that it makes it easy for users to read information.

This 20*4 LCD display screen is going to be used to display the relevant information. After programming the screen, a conclusion of water quality(Good, Average, Poor) will be displayed along with the remaining water level.

POWER SUPPLY UNIT:

ZN-MN BATTERY:

The Zn-Mn battery must be able to continuously support the functioning of the circuit, display unit, and the mechanical unit.

Requirement: Commercial batteries will be used to maintain a continuous 3.60V power supply for at least 24 hours. If the chosen battery is not powerful enough, 120V power outlets will be considered.

VOLTAGE REGULATOR:

The integrated circuit will regulate the power supply for each module to maintain their functionality. This chip must be able to handle the maximum voltage supplied by the battery (3.60V ± 0.5V) while ensuring the voltage at each module does not exceed their limit.

Requirement: Must maintain thermal stability below 100°C.

MECHANICAL UNIT:

FOUNTAIN PUMP:

The fountain pump [14] must maintain a continuous water supply through the fountain mechanism. The pump must work 24 hours a day, 7 days a week unless the user manually turns off the power supply.

Requirement 1: The fountain pump must lift a cylindrical water stream of diameter 6mm for a height of 400mm.

Requirement 2: The fountain pump must serve for a duration of 2 years without maintenance or replacement under heavy workload.

Requirement 3: The fountain pump should have an operational condition around 3V, 200mA.

SUPPLY PUMP:

The supply pump must function when a low water level alert is raised. While no water supply is requested, the pump must prevent water flow between the main supply and the fountain.

Requirement: The supply pump should have an operational condition around 3V, 200mA.

FILTER:

The filter must maintain the water quality through controlling the pH value and conductivity of the water.

Requirement 1: The filter must have a cost less than $5 each for frequent replacement. Each new filter must serve a duration no less than 3 month.

Requirement 2: The filter must be designed for easy removal and installation, while the connection mechanism must have a low degenerate rate when submerged in water.

DRAIN:

The drain [13] must be able to hold and release water in the fountain. When water in the fountain should be replaced, the faucet should automatically drain the fountain once instruction is received from the integrated circuit.

CONTROL UNIT

This unit contains the control unit which does the following things:

- When the weight sensor reports a weight less than the minimum weight setting, the control unit will send an alert signal to the user and then control the water supply unit to refill the water fountain with a certain amount of water.
- Computes the water quality with data transferred from the three

sensors in the water quality module and sends the result in terms of “Good”, “Average” or “Bad” to the user.

- If the water quality is “Bad”, the control unit will control the drain module to drain the water in the fountain and then control the water supply to refill.
- Water quality result is sent to the user with wireless connection and screen display as described above in the display unit.(unsure about keeping this function).

Program:

```
    #include <EEPROM.h>
#include <LiquidCrystal.h>
     LiquidCrystal lcd(2,3,4,5,6,7);
     long duration, inches;  int
     set_val,percentage;  bool
     state,pump;  void setup() {
     lcd.begin(16, 2);
     lcd.print("WATER LEVEL:");
lcd.setCursor(0, 1);
     lcd.print("PUMP:OFF
     MANUAL"); pinMode(8,
     OUTPUT);  pinMode(9, INPUT);
     pinMode(10, INPUT_PULLUP);
     pinMode(11, INPUT_PULLUP);
     pinMode(12, OUTPUT);
     set_val=EEPROM.read(0);
     if(set_val>150)set_val=150;
}
 void loop()
 {
 digitalWrite(3, LOW);
     delayMicroseconds(2);  digitalWrite(8,
     HIGH); delayMicroseconds(10);
     digitalWrite(8, LOW);  duration =
     pulseIn(9, HIGH);  inches
     =microsecondsToInches(duration);
     percentage=(set_val-
     inches)*100/set_val;
lcd.setCursor(12, 0);  if(percentage<0)percentage=0;
     lcd.print(percentage);
lcd.print("% ");
```

```
if(percentage<30&digitalRead(11))pump=1;
      if(percentage>99)pump=0;
      digitalWrite(12,!pump);  lcd.setCursor(5,
      1); if(pump==1)lcd.print("ON ");  else
      if(pump==0) lcd.print("OFF");
      lcd.setCursor(9, 1);
      if(!digitalRead(11))lcd.print("MANUAL");
      else lcd.print("AUTO ");
if(!digitalRead(10)&!state&digitalRead(11))
{
 state=1; set_val=inches;  EEPROM.write(0,
      set_val);
}
if(!digitalRead(10)&!state&!digitalRead(11)){ state=1;
pump=!pump;
}
if(digitalRead(10))state=0;  delay(500);
}
long microsecondsToInches(long microseconds) { return
microseconds / 74 / 2;
}
```

Output link:

HTTPS://WOKWI.COM/PROJECTS/379631344770984961