

PROJECT TITLE: SMART WATER MANAGEMENT SYSTEM

PHASE 3: DEVELOPMENT PART- I

OBJECTIVE:

The objective of this project is to create a basic monitoring system for aquatic-related parameters, such as temperature, pH, and NH₃ levels. By interfacing with an I2C LCD and simulating sensor data, the project aims to demonstrate how to visualize and display these parameters on the LCD screen. The project serves as a starting point for those interested in building more advanced aquatic monitoring systems, where the real objective would be to collect and display accurate and meaningful data from physical sensors in order to monitor and maintain optimal conditions in an aquatic environment, such as water quality control system.

WOKWI:

Wokwi is a versatile online platform that allows you to design, simulate, and test electronic circuits in a virtual environment.



Website: (<https://wokwi.com/>)

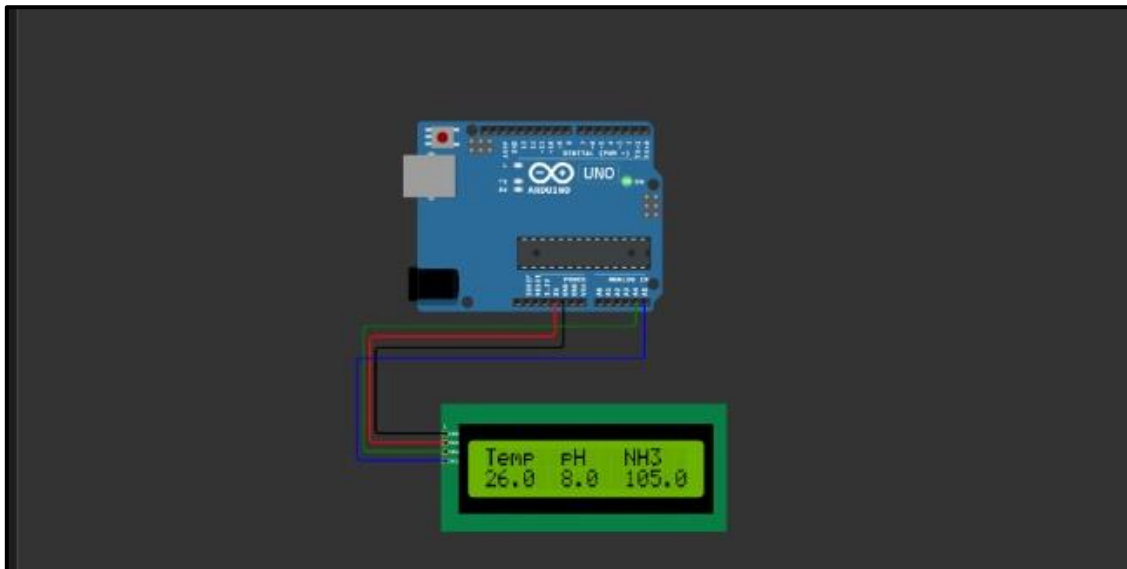
COMPONENTS REQUIRED:

- Arduino Board
- I2C LCD Display
- LiquidCrystal_I2C Library

WIRING INSTRUCTIONS:

1. Arduino Board Connections:

- **VCC (5V):** Connect the 5V output of the Arduino to the VCC (Power) pin on the I2C LCD.
- **GND (Ground):** Connect the GND (Ground) pin of the Arduino to the GND (Ground) pin on the I2C LCD.
- **SDA (Serial Data):** Connect the SDA (Serial Data) pin of the Arduino to the SDA (Data) pin on the I2C LCD.
- **SCL (Serial Clock):** Connect the SCL (Serial Clock) pin of the Arduino to the SCL (Clock) pin on the I2C LCD.



These connections establish the communication between the Arduino board and the I2C LCD, allowing the Arduino to send data and instructions to the display for visualization.

CODE DESCRIPTION:

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,20,4);
```

```
float t;
```

```
float pH;
```

```
float nh3;
```

```
long readT()
```

```
{
```

```
long t = random(1700, 2900)*0.01;
```

```
return t;
```

```
}
```

```
long readNh3()
```

```
{
```

```
long nh3 = random(9000, 14000)*0.01;
```

```
return nh3;
```

```
}
```

```
long readPh()
```

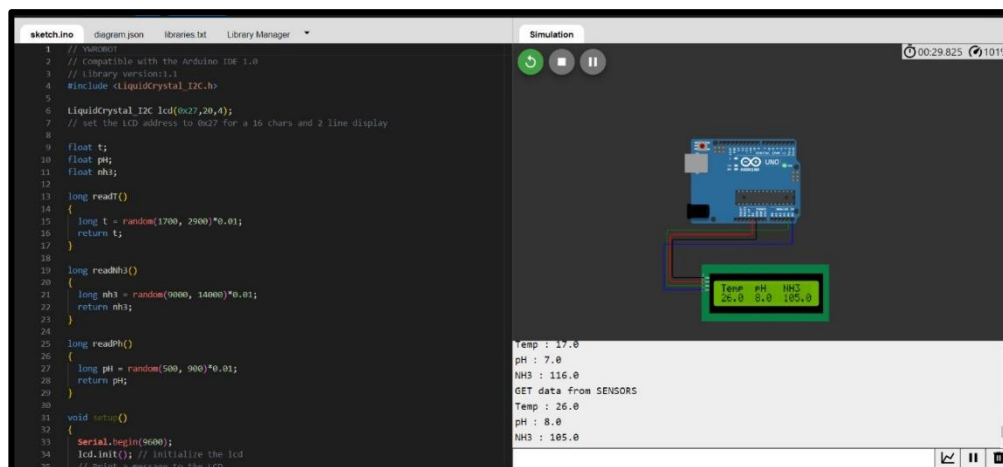
```
{  
  
  long pH = random(500, 900)*0.01;  
  
  return pH;  
  
}  
  
void setup()  
  
{  
  
  Serial.begin(9600);  
  
  lcd.init();  
  
  lcd.backlight();  
  
  lcd.print("Aqua Sensor");  
  
  delay(3000);  
  
  lcd.clear();  
  
}  
  
void loop()  
  
{  
  
  Serial.println("GET data from SENSORS");  
  
  t = readT();  
  
  pH = readPh();  
  
  nh3 = readNh3();  
  
  Serial.print("Temp : ");  
  
  Serial.println(t, 1);  
  
}
```

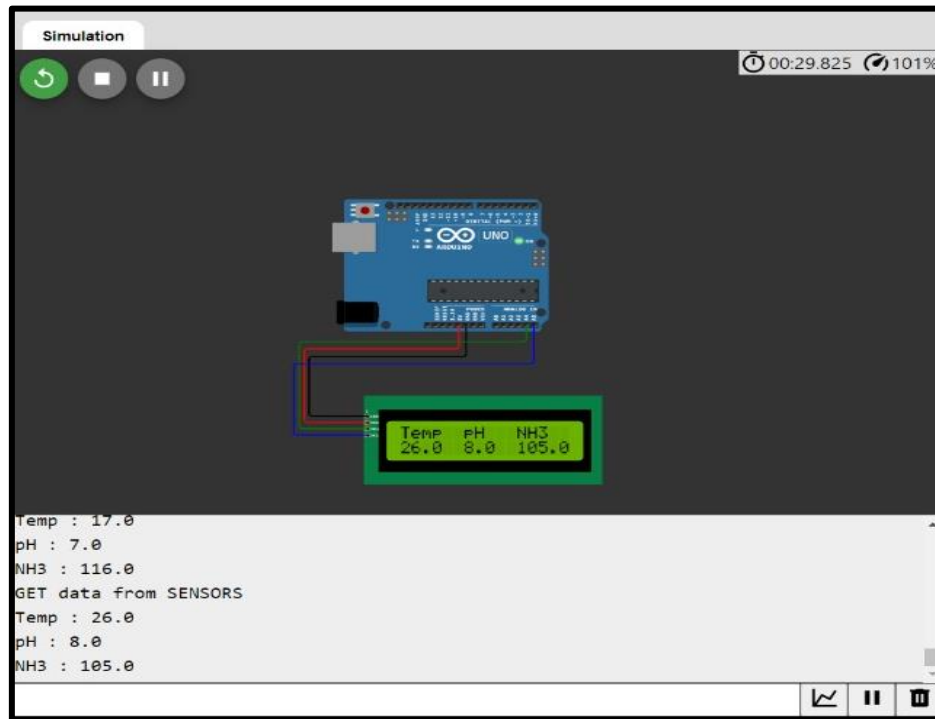
```
Serial.print("pH : ");  
  
Serial.println(pH, 1);  
  
Serial.print("NH3 : ");  
  
Serial.println(nh3, 1);  
  
lcd.setCursor(0,0);  
  
lcd.print("Temp");  
  
lcd.setCursor(0,1);  
  
lcd.print(t, 1);  
  
lcd.setCursor(6,0);  
  
lcd.print("pH");  
  
lcd.setCursor(6,1);  
  
lcd.print(pH, 1);  
  
lcd.setCursor(11,0);  
  
lcd.print("NH3");  
  
lcd.setCursor(11,1);  
  
lcd.print(nh3, 1);  
  
delay(2000);  
  
lcd.clear();  
  
}
```

This code employs the LiquidCrystal_I2C library to interface with a 20x4 I2C-based LCD, demonstrating the visualization of simulated sensor data. Three float variables, representing temperature, pH, and NH3 levels, are declared, and functions generate random data within predefined ranges to mimic sensor readings. In the setup phase, serial communication is initiated, the LCD is initialized, and a brief "Aqua Sensor" message is displayed before clearing the screen. The main loop iteratively generates simulated sensor data, displaying it on both the LCD and the Serial Monitor with corresponding labels, followed by a 2-second delay and screen clearing for the next data set. To apply this code practically, replace the simulated data with actual sensor readings from physical sensors connected to the Arduino.

RESULT ANALYSIS:

The Smart Water Management System project successfully displayed simulated sensor data, including temperature, pH, and NH3 levels, on a 20x4 I2C-based LCD. Upon startup, the LCD showed an "Aqua Sensor" message for 3 seconds, followed by clear presentation of the sensor data with labels, separated by 2-second intervals. While the project achieved its goal of showcasing data visualization, it's important to note that the data generated was simulated rather than obtained from actual sensors.





CONCLUSION:

The Smart Water Management System successfully demonstrated the visualization of simulated sensor data on the I2C LCD display. This project serves as a foundational example for those interested in creating water monitoring systems. To make this project practical for real-world applications, the next steps would involve connecting actual sensors to the Arduino and adapting the code to read and display real data.