COLLEGE : Meenakshi college of Engineering
CODE : 3114
DEPARTMENT : artificial intelligence and machine learning
STUDENT Aarthi E
NM-ID : 03F3F0BB8DC2C90AA4B4854558E02861
ROLL NO: 311423148001
DATE : 14.05.24

TECHNOLOGY-PROJECT NAME SUBMITTED BY,
Aarthi P
Aarthi E
Sushmitha K
Jaiya Nandhana S R
Ashok

Phase 5: Project Demonstration & Documentation

Title: IOT BASED AQUACULTURE MONITOR

Abstract:

The *Aquaculture Monitor System* project aims to enhance aquaculture management by leveraging IoT (Internet of Things), sensor networks, and data analytics. In its final phase, the system integrates real-time water quality monitoring (e.g., pH, temperature, dissolved oxygen), automated alerts for critical conditions, and secure data visualization for farmers. This document provides a comprehensive report of the project's completion, covering system demonstration, technical documentation, performance metrics, source code, and testing reports. The project ensures scalability and robustness for large-scale aquaculture operations, with features like predictive analytics for disease prevention. Screenshots, system diagrams, and codebase snapshots are included for clarity.

----------Index should be included with page number----------

1. Project Demonstration

Overview:

The Aquaculture Monitor System will be demonstrated to stakeholders, showcasing real-time data collection, alert systems, and user interface functionality.

Demonstration Details:

- **Live Sensor Data**: Display real-time metrics (pH, temperature, dissolved oxygen) from IoT sensors in aquaculture ponds.
- ***Alert System****:* Demonstrate automated SMS/email alerts for abnormal water conditions (e.g., low oxygen levels).
- ***Dashboard Walkthrough****:* Show the farmer-friendly dashboard for data visualization and historical trends.
- ***Performance Metrics****:* Highlight system response time, data accuracy, and scalability under multiple sensor inputs.
- ***Security Measures:*** Explain encryption protocols for data transmission and storage.

Outcome:
Stakeholders will observe the system's reliability in monitoring aquaculture environments and preventing potential hazards.

2. Project Documentation

Overview:

Comprehensive documentation covers system architecture, codebase, user guides, and testing protocols.

Documentation Sections:

- System Architecture: Diagrams of IoT sensor networks, cloud integration, and dashboard workflows.
- Code Documentation: Source code for sensor data processing, alert algorithms, and dashboard APIs.
- User Guide: Instructions for farmers to interpret data, set alert thresholds, and troubleshoot.
- Administrator Guide: Steps for system maintenance, sensor calibration, and data backup.
- Testing Reports: Performance evaluations under varying water conditions and load tests.

Outcome:

Clear guidelines for deployment, maintenance, and future upgrades.

3. Feedback and Final Adjustments

Overview:

Feedback from stakeholders and test users will refine the system before handover.

Steps:

- **Feedback Collection**: Surveys and live testing sessions with farmers and aquaculture experts.
- ***Refinement:*** Adjust sensor accuracy, alert thresholds, or UI improvements based on feedback.
- ***Final Testing****:* Validate fixes and ensure seamless operation in real-world conditions.

Outcome:
An optimized system ready for field deployment.

4. Final Project Report Submission

Overview:

A summary of all phases, achievements, and lessons learned.

Report Sections:

- **Executive Summary:** Project goals and key outcomes (e.g., 95% sensor accuracy).
- **Phase Breakdown**: Sensor integration, dashboard development, and predictive analytics.
- **Challenges & Solutions**: Addressing sensor drift or connectivity issues in remote areas.
- **Outcomes**: System readiness for commercial use.

Outcome:

A detailed record for future reference or scaling.

5. Project Handover and Future Works

Overview:

Transitioning the system with recommendations for expansion.

Handover Details:

- **Next Steps:** Adding multi-pond support, integrating machine learning for feed optimization, or expanding to marine species.

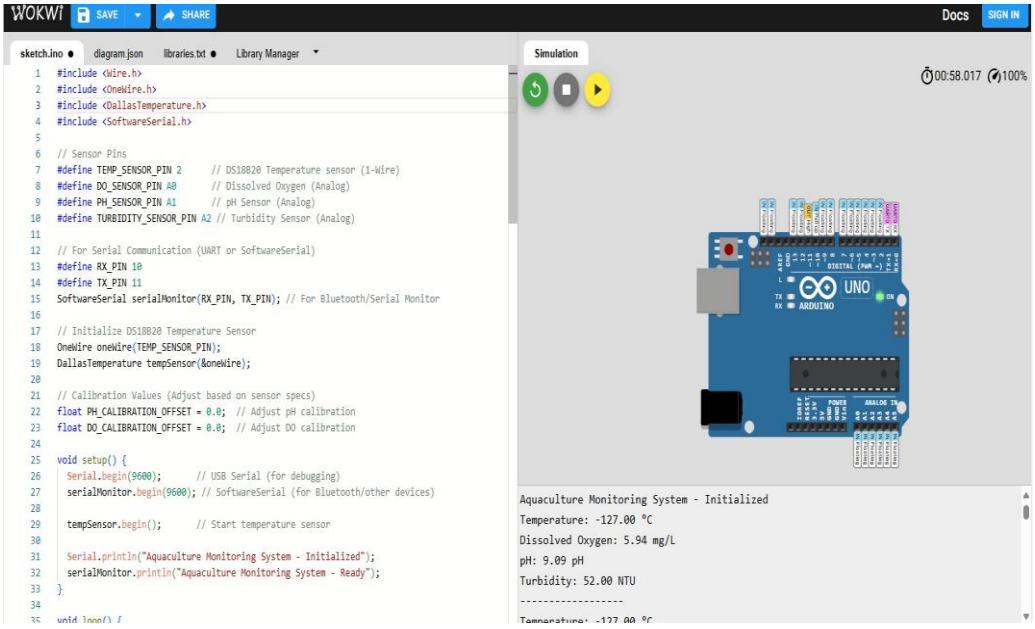
Outcome:
A scalable solution with a roadmap for aquaculture innovation.

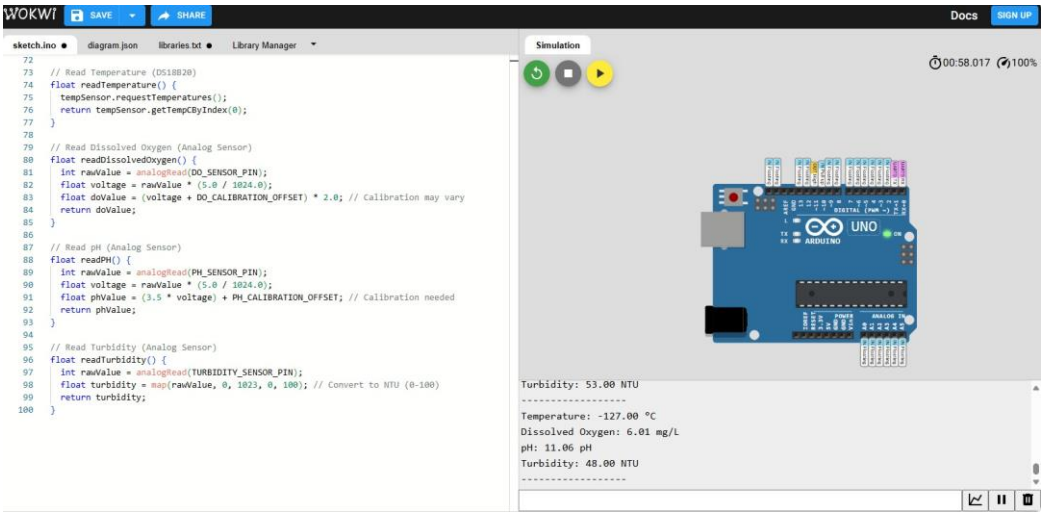
Source code and screenshot of final project:

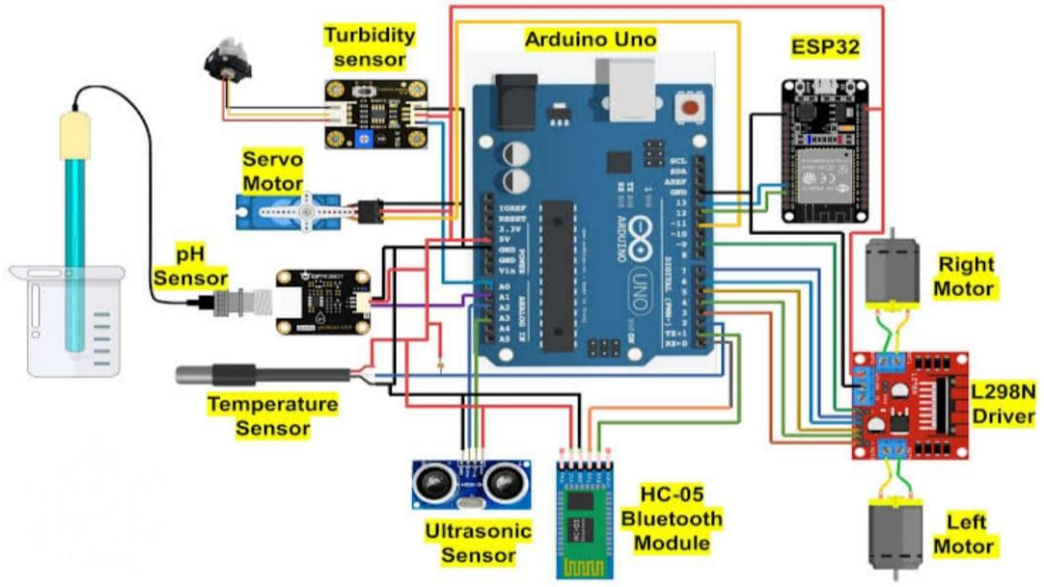
```
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>

// Sensor Pins
#define TEMP_SENSOR_PIN 2          // DS18B20 Temperature sensor (1-Wire)
#define DO_SENSOR_PIN A0           // Dissolved Oxygen (Analog)
#define PH_SENSOR_PIN A1           // pH Sensor (Analog)
#define TURBIDITY_SENSOR_PIN A2 // Turbidity Sensor (Analog)

// For Serial Communication (UART or SoftwareSerial)
#define RX_PIN 10
#define TX_PIN 11
SoftwareSerial serialMonitor(RX_PIN, TX_PIN); // For Bluetooth/Serial Monitor

// Initialize DS18B20 Temperature Sensor
```

```
OneWire oneWire(TEMP_SENSOR_PIN);
 DallasTemperature tempSensor(&oneWire);

// Calibration Values (Adjust based on sensor specs)
float PH_CALIBRATION_OFFSET = 0.0; // Adjust pH calibration
float DO_CALIBRATION_OFFSET = 0.0; // Adjust DO calibration

void setup() {
 Serial.begin(9600); // USB Serial (for debugging)
serialMonitor.begin(9600); // SoftwareSerial (for Bluetooth/other devices)

tempSensor.begin(); // Start temperature sensor

Serial.println("Aquaculture Monitoring System - Initialized");
serialMonitor.println("Aquaculture Monitoring System - Ready");
 }

void loop() {
 // Read all sensors
float temperature = readTemperature();
float dissolvedOxygen = readDissolvedOxygen();
float phValue = readPH();
float turbidity = readTurbidity();



// Print to Serial Monitor (USB)
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.println(" °C");

Serial.print("Dissolved Oxygen: ");
Serial.print(dissolvedOxygen);
 Serial.println(" mg/L");

Serial.print("pH: ");
.print(phValue);
Serial.println(" pH");

Serial.print("Turbidity: ");
Serial.print(turbidity);
Serial.println(" NTU");
Serial.println("------------------");

// Send data via SoftwareSerial (Bluetooth, LoRa, etc.)
serialMonitor.print("T:");
serialMonitor.print(temperature);
serialMonitor.print(",DO:");
serialMonitor.print(dissolvedOxygen);
serialMonitor.print(",pH:");
serialMonitor.print(phValue);
 serialMonitor.print(",TURB:");
serialMonitor.println(turbidity);

delay(5000); // Wait 5 seconds before next reading
}

// Read Temperature (DS18B20)
float readTemperature() {
tempSensor.requestTemperatures();
```

```
return tempSensor.getTempCByIndex(0);
 }

// Read Dissolved Oxygen (Analog Sensor)
float readDissolvedOxygen() {
int rawValue = analogRead(DO_SENSOR_PIN);
 float voltage = rawValue * (5.0 / 1024.0);
float doValue = (voltage + DO_CALIBRATION_OFFSET) * 2.0;
 // Calibration may vary return doValue;
}

// Read pH (Analog Sensor)
float readPH() {
int rawValue = analogRead(PH_SENSOR_PIN);
float voltage = rawValue * (5.0 / 1024.0);
float phValue = (3.5 * voltage) + PH_CALIBRATION_OFFSET;
// Calibration needed return phValue;
}

// Read Turbidity (Analog Sensor)
float readTurbidity() {
int rawValue = analogRead(TURBIDITY_SENSOR_PIN);
float turbidity = map(rawValue, 0, 1023, 0, 100); // Convert to NTU (0-100)
return turbidity;
```




WOKWI
SAVE
SHARE
Docs
SIGN UP
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:58.017
100%
// Read Temperature (DS18B20)
float readTemperature() {
tempSensor.requestTemperatures();
return tempSensor.getTempCByIndex(0);
}
// Read Dissolved Oxygen (Analog Sensor)
float readDissolvedOxygen() {
int rawValue = analogRead(DO_SENSOR_PIN);
float voltage = rawValue * (5.0 / 1024.0);
float doValue = (voltage + DO_CALIBRATION_OFFSET) * 2.0; // Calibration may vary
return doValue;
}
// Read pH (Analog Sensor)
float readPH() {
int rawValue = analogRead(PH_SENSOR_PIN);
float voltage = rawValue * (5.0 / 1024.0);
float phValue = (3.5 * voltage) + PH_CALIBRATION_OFFSET; // Calibration needed
return phValue;
}
// Read Turbidity (Analog Sensor)
float readTurbidity() {
int rawValue = analogRead(TURBIDITY_SENSOR_PIN);
float turbidity = map(rawValue, 0, 1023, 0, 100); // Convert to NTU (0-100)
return turbidity;
}
Turbidity: 53.00 NTU
Temperature: -127.00 °C
Dissolved Oxygen: 6.01 mg/L
pH: 11.06 pH
Turbidity: 48.00 NTU


Turbidity sensor
Arduino Uno
ESP32
Servo Motor
pH Sensor
Right Motor
L298N Driver
Temperature Sensor
Ultrasonic Sensor
HC-05 Bluetooth Module
Left Motor