Smart Water Quality Monitoring System Using Iot

Annu Kumari[1], Dr Amit Shrivastava[2], Ms Tanvi Khandelwal[3]
SAM Global University, Bhopal

***Abstract*—Clean water is essential for life, yet increasing industrialization and pollution have made access to safe drinking water a global concern. This paper presents the design and implementation of a Smart Water Quality Monitoring System using Internet of Things (IoT) technologies. The system uses sensors to measure water parameters such as pH, temperature, and turbidity, and transmits data via an ESP8266 Wi-Fi module to the ThingSpeak cloud for real-time monitoring. This solution provides an affordable, scalable, and efficient alternative to traditional water testing methods, allowing for prompt detection of contamination and enhancing public health response.**

1. INTRODUCTION

The quality of water is crucial to human health, agriculture, and the environment. Traditional water testing methods are time-consuming, expensive, and often fail to provide real-time insights. With the advent of IoT, it is now possible to develop real-time monitoring systems that can remotely analyze various parameters of water and alert authorities when values exceed safe limits. This paper describes an IoT-based Smart Water Quality Monitoring System that utilizes pH, turbidity, and temperature sensors connected to an Arduino microcontroller. Sensor data is uploaded to a cloud platform, providing accessible, live updates on water quality.

2. LITERATURE REVIEW

Ref No	Title	Technology Used	Parameters Measured	Communication
[1]	Zigbee-based Water Quality Monitoring	Zigbee, WSN	pH, Temperature, Turbidity	IEEE 802.15.4
[2]	GSM-based Autonomous Water Monitor	GSM, Biosensors	Salinity, DO, EC, Light Intensity, pH	GSM SMS
[3]	Image Processing for Water Quality Monitoring	Fuzzy Inference, Camera	Fish Motion Response	Wireless
[4]	ZigBee Smart Sensors for Water Quality	ZigBee, Smart Sensors	Conductivity, pH, Temperature, Flow Rate	Zigbee
[5]	WSN for Water Management	GUI, Wireless Network	pH, Turbidity, Temperature	Wireless
[6]	Zigbee + GPRS in WSN-based Monitoring	Zigbee, GPRS	pH, Conductivity, Temperature	GPRS
[7]	Cloud-Based Real-Time Monitoring System	Embedded System, Cloud	pH, Temperature, Turbidity, Flow	Cloud + GSM

This system builds on the existing literature by integrating multiple sensors and using the ThingSpeak platform for better visualization and remote access.

3. METHODOLOGY

Components Used:

- Arduino Mega 2560
- DS18B20 Temperature Sensor
- DFRobot pH Sensor
- DFRobot Turbidity Sensor
- ESP8266 Wi-Fi Module
- GSM Module (for SMS alerts)
- LCD Display
- Power Supply and Regulator

System Architecture: The sensors collect data and pass it to the Arduino Mega 2560. The microcontroller processes the data and sends it to the ThingSpeak server via the ESP8266 module. If any value breaches the safe threshold, an alert is also sent via the GSM module.

Flowchart:

- Start
- Initialize sensors and modules
- Read data every 15 minutes
- If data > threshold:
  - Take 12 readings at 3-sec intervals
  - If average still unsafe, send alert
- Upload safe/unsafe data to ThingSpeak and SD card
- Repeat

4. SAMPLE CIRCUIT DIAGRAM

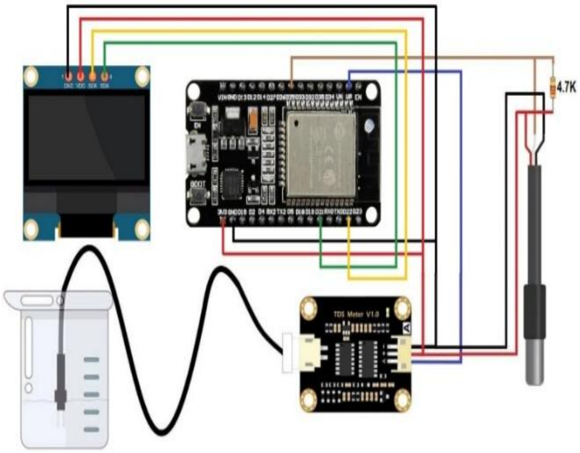

Fig.2: Circuit diagram showing connection of sensors to Arduino and ESP8266.

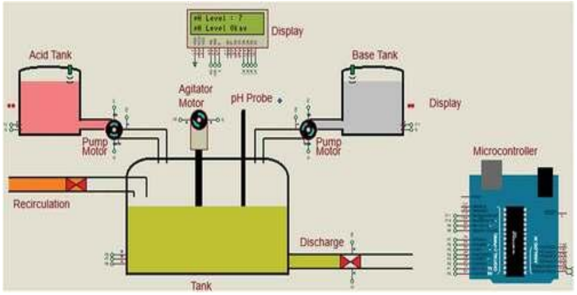

Fig.2: Circuit diagram showing connection of sensors to Arduino and ESP8266.

5. SAMPLE PICTURES

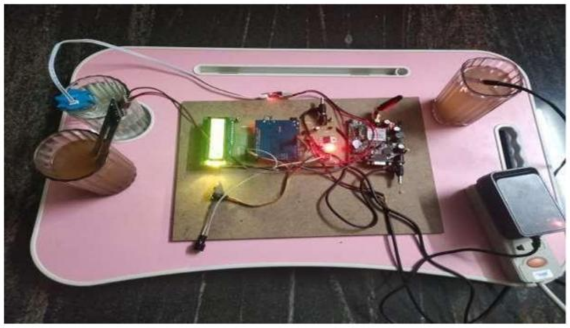

Fig.3: Final hardware prototype of the water quality monitoring system.

6. SAMPLE CODE

```
#include <ESP8266WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS D4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

const char* ssid = "YourSSID"
const char* password = "YourPassword";
const char* host = "api.thingspeak.com";
const char* apiKey = "YOUR_API_KEY";

void setup() {
  Serial.begin(115200);
  sensors.begin();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}

void loop() {
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);

  WiFiClient client;
  if (!client.connect(host, 80)) return;

  String url = "/update?api_key=" + String(apiKey) +
"&field1=" + String(temperature);
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(15000);
}
```

REFERENCES

[1] Prathibha, S.R., et al., "Zigbee based remote monitoring of water quality," IJAREEIE, 2016.

[2] Singh, R., et al., "GSM based water pollution monitoring system," IJSRET, 2015.

[3] Gao, Y., et al., "Water quality detection using image processing," Procedia Engineering, 2014.

[4] Islam, M.A., et al., "Zigbee enabled smart sensors for water quality," Sensors, 2017.

[5] Kumar, M., et al., "Wireless sensor-based water management," IJCA, 2016.

[6] Mahajan, P., et al., "WSN with GPRS for water quality," IJETAE, 2017.

[7] Lee, S., et al., "Real-time monitoring of water using cloud," IEEE Access, 2018.