GOVERNMENT OF TAMILNADU

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

IoT AND ITS APPLICATIONS

HOST INSTITUTION

xxxxx

COIMBATORE – 04

TRAINING PARTNER

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

Name	Roll No

Table Of Contents

ABSTRACT

This project focuses on the development of a water quality monitoring system using the ESP32 microcontroller and a DHT11 sensor. The system is designed to measure and report environmental conditions such as temperature and humidity, which are crucial factors in assessing water quality. Additionally, in the absence of a pH sensor, the system generates a simulated pH value using a randomization technique. This approach allows for the testing and demonstration of the system's functionality without the need for a physical pH sensor. The data collected, including temperature, humidity, and simulated pH values, is displayed via a serial monitor, providing a comprehensive overview of the water quality. This project demonstrates an efficient and cost-effective method for environmental monitoring, with potential applications in various water quality management systems.

INTRODUCTION

This project develops a water quality monitoring system with the ESP32 microcontroller, known for its wireless capabilities and low power consumption. It integrates a DHT11 sensor to measure temperature and humidity, which affect water quality by influencing oxygen solubility and evaporation rates. In the absence of a pH sensor, the system generates simulated pH values for testing and demonstration. Real-time data is displayed via a serial monitor, providing an overview of environmental conditions impacting water quality. This approach is beneficial for researchers, educators, and hobbyists, showcasing how affordable technology can advance water quality monitoring and environmental conservation. Overall, this project highlights the potential of using affordable microcontrollers and sensors to develop practical solutions for water quality monitoring, contributing to broader efforts in environmental conservation and resource management.

Hardware and Software Requirements

Hardware Requirements

1. ESP32 Microcontroller
2. DHT 11
3. USB Cable
4. Bread Board
5. Jumper Wire

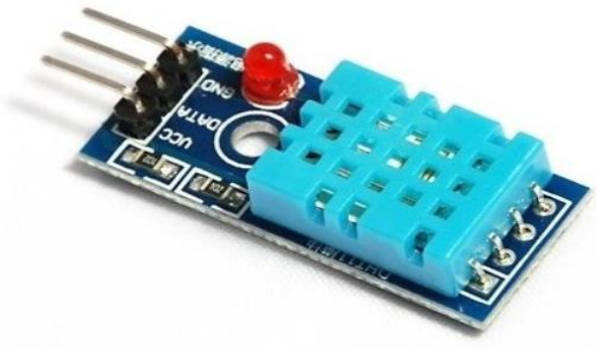
Software Requirements

1. Wok-Wi Simulator
2. Arduino IDE
3. Thingzmate Cloud

ESP32 Microcontroller

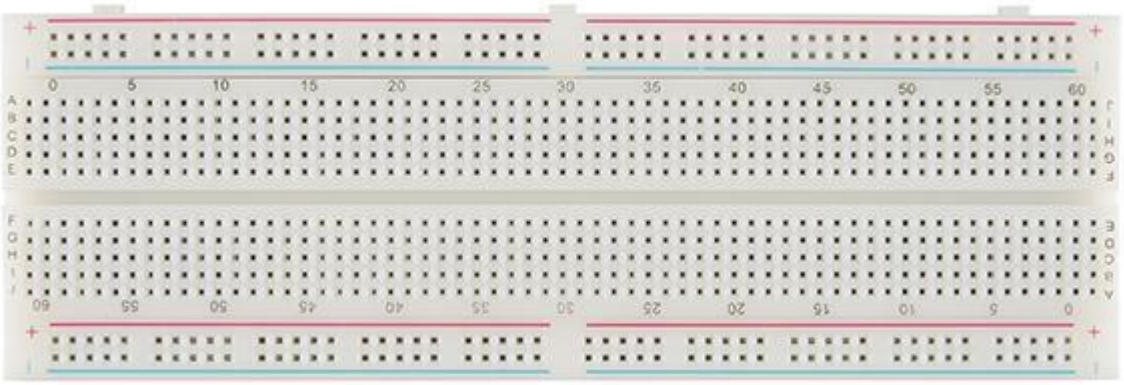
The ESP32 microcontroller is a powerful and versatile choice for IoT-based home automation projects, offering built-in Wi-Fi and Bluetooth capabilities for seamless wireless communication. Its dual-core processor, combined with a rich set of peripherals, allows for efficient control and monitoring of multiple household appliances. Additionally, the ESP32's low power consumption and robust security features make it an ideal solution for maintaining a reliable and secure smart home ecosystem.

DHT 11 Sensor



The relay module in this IoT-based home automation project is used to control high-voltage appliances, such as fans and heaters, by interfacing with the low-voltage ESP32 microcontroller. It acts as a switch that opens or closes electrical circuits based on control signals from the microcontroller. This allows for safe and reliable automation of home appliances, enabling their activation or deactivation remotely.

Bread Board



The breadboard in this IoT-based home automation project is used as a prototyping platform to easily assemble and test the connections between the ESP32 microcontroller, LEDs, and other components. It allows for quick adjustments and modifications without soldering, facilitating the design and troubleshooting process. This setup helps in efficiently demonstrating and validating the circuit before moving on to a more permanent solution.

USB-Cable

The USB cable is a critical tool in this project, used to connect the ESP32 microcontroller to a computer for power supply, programming, and debugging. It enables the transfer of code and data between the development environment and the microcontroller, facilitating the upload of firmware and real-time communication during the development process. The USB connection also allows for serial monitoring, providing valuable insights into the system's performance and behavior.

Jumper Wires

Jumper wires are essential in this project, used to connect the ESP32 microcontroller to the components on the breadboard. These wires provide a flexible and reliable way to link the microcontroller's GPIO pins to the LEDs, resistors, and other circuit elements, enabling proper signal and power flow. Their ease of use allows for quick modifications and testing during the prototyping stage.

Wokwi Simulator

The Wok-Wi simulator is utilized in this IoT-based home automation project to virtually prototype and test the circuit, including the ESP32 microcontroller, LEDs, and other components, without the need for physical hardware. It provides an interactive environment where code and connections can be simulated and debugged, ensuring that the system functions as expected before real-world implementation. This tool accelerates development by allowing iterative testing and refining of the design cost-effectively.

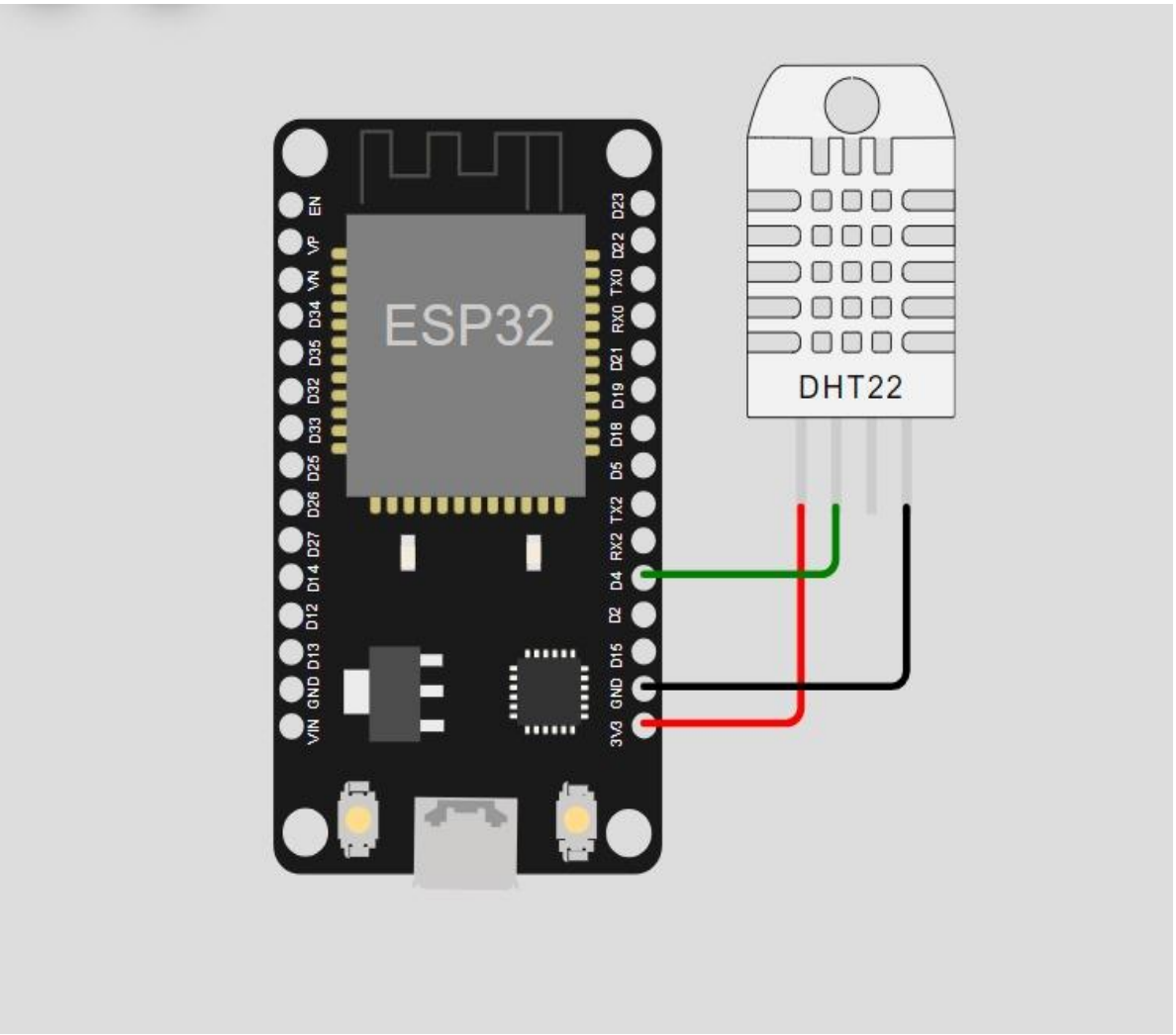
Arduino IDE

The Arduino IDE is employed in this IoT-based home automation project to write, compile, and upload code to the ESP32 microcontroller. Its user-friendly interface and extensive library support streamline the development process, making it easier to program and integrate various components. The IDE also facilitates debugging and real-time monitoring, enhancing the overall efficiency of the project development.

Thingzmate Cloud

Thingzmate Cloud is utilized in this IoT-based home automation project to provide a robust platform for remotely managing and controlling household appliances. It enables secure communication between the ESP32 microcontroller and user interfaces, facilitating real-time updates and control through web and mobile applications. The cloud service also offers data storage and analytics, enhancing the system's functionality and user experience.

Block Diagram



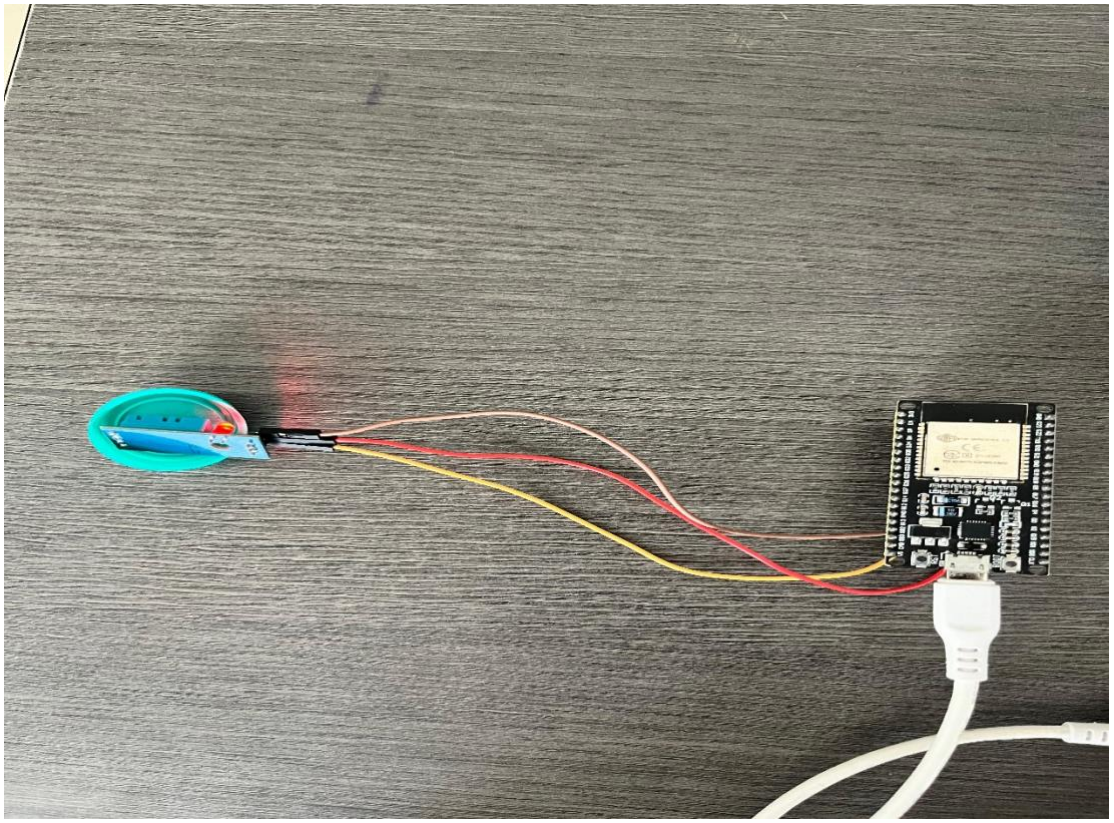
Code

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <DHT.h>

#define WIFI_SSID "USERNAME"
#define WIFI_PASSWORD "PASSWORD"

#define DHTPIN 4      // Define the DHT11 data pin
#define DHTTYPE DHT11 // Define the type of DHT sensor (DHT11)

DHT dht(DHTPIN, DHTTYPE);

const char *serverUrl = "https://console.thingzmate.com/api/v1/device-types/esp31/devices/esp31/uplink"; // Replace with your server endpoint
String AuthorizationToken = "Bearer 1d3df63191d1438ced97e1a0909fc623";

void setup () {
  Serial.begin(115200);
  delay(4000); // Delay to let serial settle

  // Connect to WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");

  // Initialize the DHT sensor
  dht.begin();
}
```

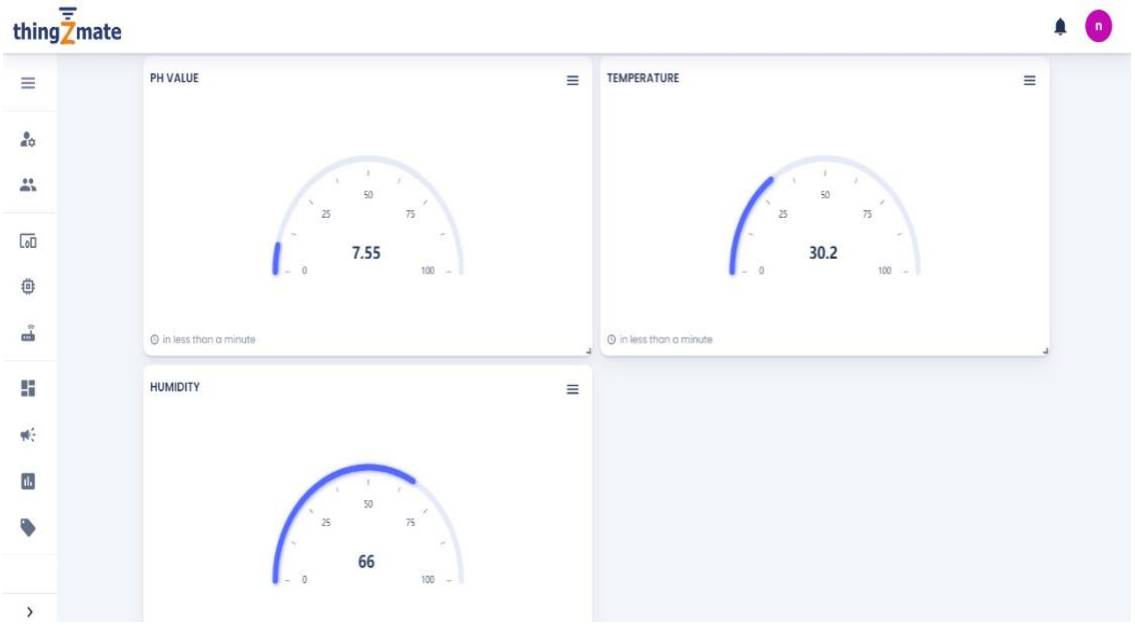
```
void loop() {
  float Temperature = dht.readTemperature();
  float Humidity = dht.readHumidity();
  int Raw_Value = 100; // Example value, replace with your sensor reading if needed

  // Generate a random pH value between 0 and 14
  float pH = random(0, 1400) / 100.0;

  // Check if any reads failed and exit early (to try again).
  if (isnan(Temperature) || isnan(Humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  HTTPClient http;
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/json");
  http.addHeader("Authorization", AuthorizationToken); // Authorization token

  // Create JSON payload
  String payload = "{\"temperature\":" + String(Temperature) + ",\"humidity\":" + String(Humidity) + ",\"Raw_Value\":" + String(Raw_Value) + ",\"pH\":" + String(pH) + "}";

  // Send POST request
  int httpResponseCode = http.POST(payload);

  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println("HTTP Response code: " + String(httpResponseCode));
    Serial.println(response);
  } else
```

```
 {
    Serial.print("Error code: ");
Serial.println (http ResponseCode);
  }
  http.end(); // Free resources
  delay(10000); // Wait for 10 seconds before sending next
request
}
```

Output Results



Cloud Output



Conclusion

This project demonstrates the successful implementation of a water quality monitoring system using the ESP32 microcontroller and DHT11 sensor, with data transmission handled through Thingzmate. By integrating Thingzmate, the system not only measures temperature and humidity but also simulates pH values, providing a comprehensive overview of water quality in real-time. The use of Thingzmate enables efficient and seamless data transmission to remote servers, allowing for easy monitoring and analysis from anywhere. This approach underscores the effectiveness of combining low-cost hardware with cloud-based platforms like Thingzmate, offering a scalable and adaptable solution for water quality management and environmental monitoring.