Internet of Things workshop-Experiments list

Dr. Mohammad Salah Uddin

December 22, 2024

Introduction

This document details four experiments that involve integrating a NodeMCU microcontroller with web and Java server-based interfaces to achieve real-time control and monitoring of relays and sensors. These experiments aim to provide hands-on experience in IoT application development and server communication.

Experiments

Experiment 1: Relay Control via Web Interface

Objective

To control relays connected to a NodeMCU microcontroller via a user-friendly web interface.

Equipment Required

- NodeMCU (ESP8266) microcontroller
- Relay module (3 relays)
- Jumper wires and breadboard
- 5V power source
- Laptop/PC with Arduino IDE

Steps

1. Connect the relay module to the NodeMCU, mapping each relay to the following GPIO pins:
    - Relay 1: GPIO5
    - Relay 2: GPIO4
    - Relay 3: GPIO16
2. Write a program to control the relays via a web-based interface using HTML and JavaScript hosted on the NodeMCU.
3. Implement toggling buttons for each relay to send ON/OFF commands.
4. Test the interface by toggling relays and observing the hardware response.

Outcome

Participants will be able to interact with relays through a web interface and understand the basics of IoT-based control systems.

Experiment 2: Sending Real-Time Relay Status to a Java Server (GUI)

Objective

To send the status of relays connected to a NodeMCU to a Java server in real-time and visualize their states using a graphical user interface.

Equipment Required

- NodeMCU (ESP8266) microcontroller
- Relay module (3 relays)
- Laptop/PC with Arduino IDE
- Pre-configured Java server

Steps

1. Extend the code from Experiment 1 to include HTTP POST requests that send relay statuses to a Java server.
2. Create a Java program to accept and parse relay status data using JSON format.

3. Display relay statuses in real-time using a graphical interface in the Java application.
4. Test the system by toggling relay states through the web interface and observing the real-time updates on the Java GUI.

Outcome

Participants will gain experience in sending IoT device data to a server and visualizing it in real-time.

Experiment 3: Real-Time Analog Data Visualization (Webpage)

Objective

To read analog data from a sensor connected to the NodeMCU and visualize the readings on a web-based interface.

Equipment Required

- NodeMCU (ESP8266) microcontroller
- Analog sensor (e.g., potentiometer, temperature sensor)
- Jumper wires and breadboard
- Laptop/PC with Arduino IDE

Steps

1. Connect the analog sensor to the NodeMCU's A0 pin.
2. Write a program to read analog values and update them on a web-based interface in real-time.
3. Use JavaScript and HTML to display the data dynamically on a webpage.
4. Test the setup by varying the sensor input and observing the live updates.

Outcome

Participants will learn to build real-time data visualization systems using NodeMCU and web technologies.

Experiment 4: Sending Real-Time Analog Pin Data Value to a Java Server

Objective

To send analog data from sensors connected to the NodeMCU to a Java server and visualize the data dynamically.

Equipment Required

- NodeMCU (ESP8266) microcontroller
- Analog sensor (e.g., potentiometer, temperature sensor)
- Laptop/PC with Arduino IDE
- Pre-configured Java server

Steps

1. Extend the code from Experiment 3 to send analog sensor data to a Java server using HTTP POST requests.
2. Configure the Java server to accept and parse the data.
3. Visualize the sensor readings dynamically in a graphical interface on the Java server.
4. Test the system by varying sensor input and verifying real-time updates on the server.

Outcome

Participants will understand how to send and visualize IoT sensor data on a server, enabling advanced real-time applications.