IoTCPP Smart Lab Automation Project Overview

The **IoTCPP Smart Lab Automation** project is designed to automate a laboratory environment using IoT technologies. By integrating sensors, actuators, and cloud-based solutions, the system monitors and controls various aspects of the lab, including environmental conditions, access control, and safety measures. The project is built on **ThingsBoard**, an IoT platform for real-time monitoring and remote control.

Project Objectives:

The main goals of the project include:

- **Temperature and Humidity Monitoring**: Ensuring the lab environment is optimal for work.
- **RFID-based Access Control**: Restricting entry to authorized personnel only.
- **Gas Detection**: Using sensors to detect hazardous gas levels, ensuring safety.
- **Door and Light Control**: Monitoring window status and controlling light intensity.
- **Cloud Integration**: Real-time data visualization and remote control of devices using **ThingsBoard**.

Team Members and Task Allocation:

The project was developed collaboratively, with each team member focusing on specific tasks:

- **Pouya Soleimanifar**: Managed relay control and the integration of Remote Procedure Calls (RPC) for remote device control via **ThingsBoard**.
- **Mahdi Gheshmi**: Handled temperature and humidity sensors (DHT11) and its documentation.
- **Alireza Naderinasab**: Focused on magnetic sensors, LDR, and gas sensors, and assisted with LCD integration.
- **Aida Alimohammadi**: Supported relay control and the documentation for the same, helping with LDR sensor tasks.
- **Mobina Fakhimi**: Managed the **ThingsBoard** cloud platform, dashboard design, and widget integration.
- **Asal Etaati**: Created primary documentation and contributed to the LCD setup and data visualization.
- **Amirhossein Mohammadi**: Managed RFID and LCD modules.

Project Components and Functionalities:

1. Magnetic and Light Sensors (LDR):

- **Purpose**: Monitor Door status and ambient light intensity.
- **Hardware**: **ESP32**, **HW-017 Magnetic Sensor**, **LDR**.
- **Actions**:
  - If the door is open, a **red LED** lights up, and a **buzzer** sounds.
  - If closed, the **green LED** activates.
  - LDR sensor sends light intensity data to **ThingsBoard** for visualization.

2. Temperature and Humidity Monitoring (DHT11):

- **Purpose**: Maintain optimal lab conditions.
- **Hardware**: **DHT11** with **ESP32**.
- **Actions**:
  - Regular data transmission to **ThingsBoard** via MQTT.
  - High humidity triggers warnings and climate control systems.

3. RFID Access Control:

- **Purpose**: Ensure that only authorized personnel can access the lab.
- **Hardware**: **RFID Module (MFRC522)** with **ESP32**.
- **Implementation**:
  - Authorized personnel have RFID cards with unique **UID**.
  - When scanned, the UID is verified against an authorized list.
  - If valid, **ThingsBoard** displays the user's name and access status (Entered/Exited).
  - Unauthorized access triggers a buzzer and "Not Allowed" message.

4. Gas Detection (MQ2 Sensor):

- **Purpose**: Monitor hazardous gas levels in the lab for safety.
- **Hardware**: **MQ2 Gas Sensor** with **ESP32**.
- **Actions**:
  - If gas concentration exceeds a safe threshold, a **red LED** blinks, and a **buzzer** sounds.
  - Data is sent to **ThingsBoard** for real-time monitoring.

5.RPC:

- **Relay Control via RPC**: We used **ESP8266** with a relay to control devices via **ThingsBoard** using RPC. The device received commands from a switch widget on **ThingsBoard** through MQTT, toggling the relay state based on a boolean value.

6. ThingsBoard Dashboard Integration:

- **Platform**: **ThingsBoard Cloud**.
- **Dashboard Elements**:
  - Real-time visualization of **Temperature & Humidity**.
  - **RFID Access Table** showing personnel access logs.
  - **Gas Monitoring** with CO2 levels and warning indicators.
  - **Light Control** via LDR sensors' intensity data.
  - **Power Control Widgets** to remotely toggle LEDs and relays.

Protocols and Implementation:

- **MQTT**: Used for lightweight, real-time data transmission between **ESP32** and the cloud.
- **HTTP**: Used when reliability is prioritized over speed.
- **Arduino IDE & C++**: The programming environment and language for **ESP32**.

Conclusion and Recommendations:

The project achieved its goal of automating a smart laboratory environment by integrating IoT-based sensors, remote control, and real-time monitoring via **ThingsBoard**. This enhanced lab safety, optimized working conditions, and provided a scalable model for other smart spaces.

Future Recommendations:

- **Data Encryption**: To enhance data security during transmission.
- **Energy Optimization**: Improve energy efficiency by optimizing MQTT usage.
- **Predictive Maintenance**: Use AI models for early detection of sensor failures.