



IoT Lab 2: Core-IoT Platform

Overview

This lab introduces students to Core IoT, a platform used for device management, data visualization, and real-time analytics. By the end of this lab, students will understand how to connect IoT devices to ThingsBoard, send and receive telemetry data, and create interactive dashboards for monitoring IoT systems.

Learning Outcomes

- Publish telemetry data and visualize it using ThingsBoard dashboards.
- Implement basic rule chains for event-driven automation.

Requirements

- Create a project with PlatformIO in VSCode and integrate the ESP32 board to the project (you can clone available projects at https://github.com/ACLAB-HCMUT/PlatformIO_Arduino-Framework_ESP32_Template).
- Get the temperature and humidity data from DHT20 and display these data in CoreIoT server with template dashboard.
- Complete all tasks in this lab with RTOS standard.
- Create a scheduler on the dashboard to set up scheduled turn-on and turn-off times accurately (**Extra**).
- Automating actions using ThingsBoard's rule engine (**Extra**).

Hardware

- ESP32 or ESP32-S3 Development board.
- DHT20 or DHT11 temperature and humidity sensors.
- Connectors.

Instructions

- Initial Setup:
 - ☐ Download VSCode and install the PlatformIO plugin.
 - ☐ Build the project with available source code at github link and upload the code to ESP32-S3 board.
- Get data from DHT20:



- ☐ Connect DHT20 to the ESP32-S3 board via I2C port.
- ☐ Measure temperature and humidity data from DHT20 every 5s.
- Display data in CoreIoT server:
 - ☐ Create a dashboard in CoreIoT platform.
 - ☐ Display these data in the dashboard.
- Implement RPC callback to make sure the data is correct(The device will send data to the CoreIoT server, and the server will process it using the Rule Engine, then send back a response to confirm if the data is valid).
- Input the on/off schedule via the dashboard, retrieve it through shared attributes, store it on the device.

Questions

- What protocols can be used to connect devices to ThingsBoard, and what are their advantages?
- What are the differences between shared attributes and client attributes, and when should each be used?

Resources