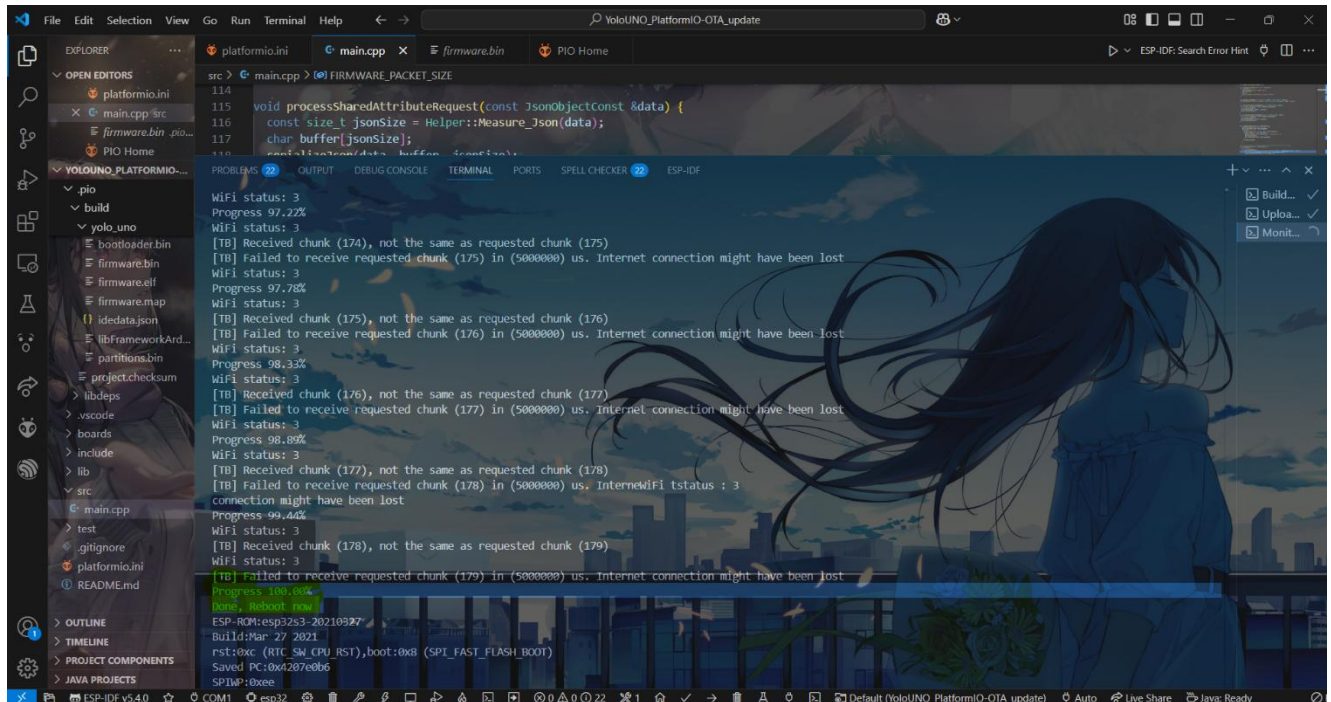


IoT Lab 3: OTA Firmware Update

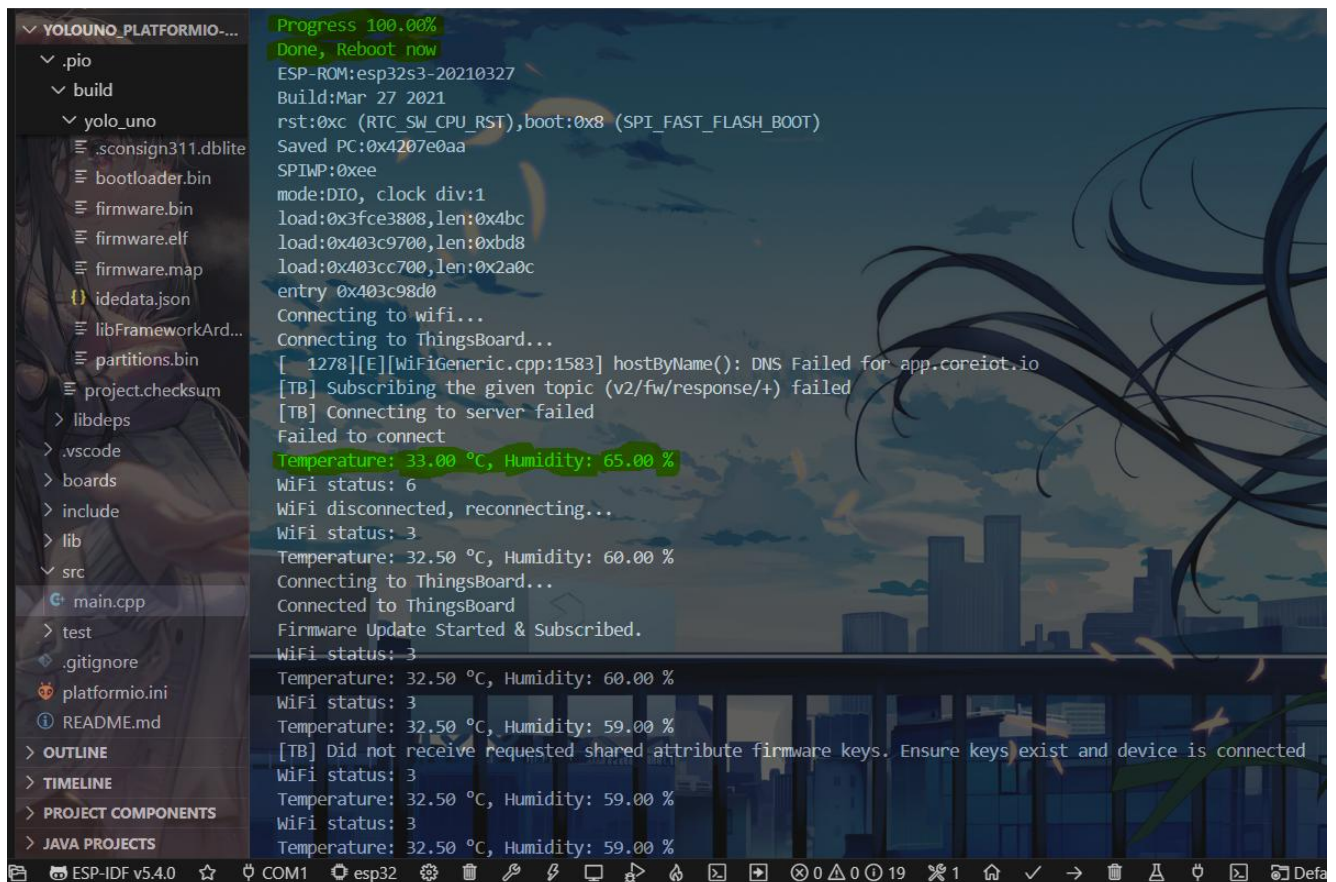
Tóm tắt quá trình thực hiện:

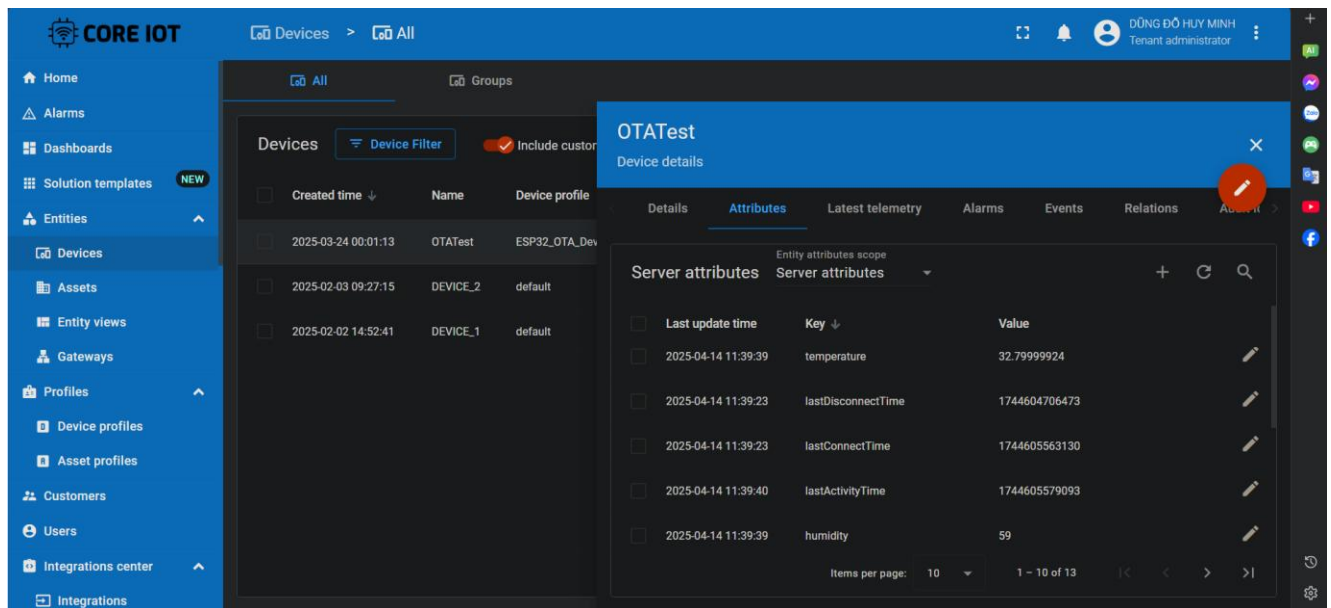
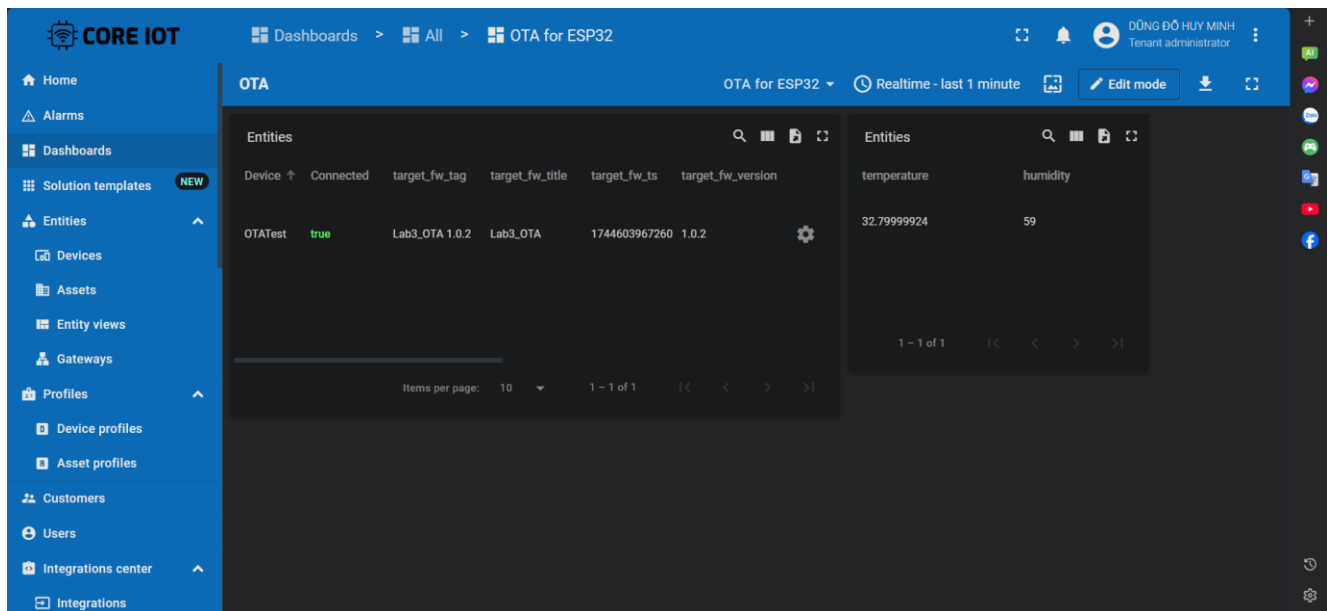
- Hiện thực mã nguồn:
 - + Hiện thực OTA update theo chuẩn RTOS (Source code: [IoT Lab 3](#)).
 - + Hiện thực mã nguồn mà OTA update sẽ tải về và compile để lấy file firmware.bin (tại .pio > build > yolo_uno > firmware.bin, nếu nạp code vào board khác yolo_uno thì đi vào thư mục tương ứng với board đó). File binary này sẽ dùng để cấu hình firmware thiết bị sẽ tải về trên core IoT.
- Rule chain:
 - + Tạo rule chain “Check is ESP32 firmware synced” để kiểm tra và theo dõi đồng bộ firmware.
 - + Rule chain này sẽ thực hiện thêm và cập nhật các attribute về firmware version vào metadata.
 - + Tiến hành thêm rule chain này vào root rule chain (Optional).
- Tạo Device Profile mới:
 - + Điền các trường cần thiết (Name, Default rule chain, Mobile Dashboard,...) trong đó chọn rule chain “Check is ESP32 firmware synced” hoặc rule chain khác chứa rule chain này.
 - + Tại Transport configuration, chọn MQTT làm phương thức chính.
- Thêm firmware mà thiết bị sẽ lấy về để cập nhật:
 - + Đi đến tác vụ Core IoT > Advanced features > OTA updates > Add package (biểu tượng dấu ‘+’) để upload firmware.
 - + Điền các trường cần thiết cho firmware này (Title, Version, Device Profile,...).
 - + Upload file firmware.bin (được đề cập ở bước hiện thực mã nguồn) và tiến hành add để hoàn thành cấu hình một packages repository về firmware mới.
 - + Quay lại Device Profile để chỉnh sửa, tại mục Assigned firmware chọn phiên bản vừa mới cấu hình.
- Thêm version firmware mới vào device:
 - + Tạo device mới trên core IoT hoặc chỉnh sửa cấu hình device đang có.
 - + Tại mục Device Profile, chọn Device Profile vừa cấu hình.
- Tạo dashboard mới và/hoặc thêm các widget cần thiết, sau đó nạp code và chạy chương trình để theo dõi và kiểm tra quá trình cập nhật firmware.
- Kiểm tra kết quả trên terminal, dashboard và các attribute được cập nhật trên Device:

+ Chương trình ban đầu chỉ có chức năng cập nhật OTA, không thực hiện đọc cảm biến, sau khi tải xong firmware mới từ trên core IoT, thiết bị reboot và sau đó mới có chức năng đọc cảm biến.



+ Sau khi firmware được cập nhật, thiết bị đã có thể đọc cảm biến và gửi data lên core IoT:





Question:

- What are some security measures you can implement to prevent unauthorized OTA updates?

Để ngăn chặn cập nhật OTA trái phép, có thể sử dụng một số biện pháp bảo mật sau:

- Xác thực và mã hóa kết nối: Sử dụng giao thức TLS/SSL để mã hóa kết nối giữa thiết bị và máy chủ OTA, đảm bảo thông tin và tệp firmware được truyền qua mạng được bảo vệ và không bị thay đổi bởi bên thứ ba.
- Sử dụng Digital Signature (Chữ ký số): Trước khi phát hành firmware mới, có thể tiến hành ký số tệp firmware bằng một khóa riêng. Trên thiết bị, khi nhận được tệp cập nhật, phải dùng khóa công khai để xác thực chữ ký số.

- Xác thực thiết bị: Đảm bảo rằng chỉ những thiết bị được ủy quyền mới có thể nhận và áp dụng OTA update. Điều này có thể được thực hiện thông qua việc cấp token, chứng chỉ số hoặc các phương thức xác thực an toàn khác.

- Kiểm tra toàn vẹn tệp (Integrity Check): Áp dụng các thuật toán băm (ví dụ như SHA-256,...) để tính toán giá trị băm của tệp firmware. Trước khi cập nhật, thiết bị sẽ so sánh giá trị băm nhận được từ máy chủ với giá trị băm tính toán nội bộ để đảm bảo tệp không bị thay đổi hoặc bị lỗi.

- How does your OTA update mechanism handle network interruptions?

Hành vi cập nhật OTA được hiện thực để xử lý tình trạng gián đoạn mạng bằng các cách:

- Sử dụng thư viện OTA_Firmware_Update và các callback xử lý tiến trình cập nhật.

- Tái kết nối WiFi tự động:

- + Theo dõi tình trạng kết nối WiFi thường xuyên (mỗi 3 giây một lần), khi mất kết nối, thiết bị sẽ tự động cố gắng kết nối lại.

- + Khi WiFi khôi phục, thực hiện kiểm tra lại kết nối với Core IoT và tiếp tục tiến trình OTA nếu trước đó đã bị gián đoạn.

- Cơ chế retry:

- + Đối với mỗi gói dữ liệu firmware, sẽ có logic để thử gửi lại (retry) khi không nhận được phản hồi từ server (được core IoT/Things board hỗ trợ), mỗi chunk nhận thất bại do lỗi truyền tải (mất kết nối mạng, băng thông yếu, nhiễu,...) sẽ được yêu cầu lại.

- + Cho phép một số lần thất bại tối đa trước khi thông báo lỗi. Nếu việc tải firmware về mất quá nhiều chi phí thời gian và năng lượng do nhận dữ liệu thất bại nhiều lần, có thể đã xảy ra vấn đề về đường truyền hoặc server, từ bỏ việc tải firmware và thông báo lỗi.

- What methods can be used to verify the integrity of the new firmware before applying it?

Để xác minh tính toàn vẹn của firmware mới trước khi áp dụng, có thể áp dụng một hoặc nhiều các phương pháp sau:

- Tính toán và so sánh hàm băm (checksum, hash):

- + Sử dụng các thuật toán băm mật mã như MD5, SHA-1 hoặc SHA-256 để tạo ra giá trị băm của tệp firmware đã tải về (core IoT có hỗ trợ Auto-generate checksum để kiểm tra và theo dõi).

- + So sánh giá trị băm tính được với giá trị băm được cung cấp bởi máy chủ cập nhật hoặc nguồn tin cậy. Nếu hai giá trị khớp nhau, điều đó cho thấy tệp firmware không bị thay đổi trong quá trình truyền tải.

- Xác minh chữ ký số (Digital Signature):

- + Firmware được ký số bằng khóa riêng của nhà sản xuất. Trên thiết bị, quá trình xác minh sử dụng khóa công khai đã được cài đặt sẵn để kiểm tra chữ ký.
- + Nếu chữ ký số hợp lệ, có nghĩa là firmware đến từ nguồn tin cậy và chưa bị can thiệp.

- So sánh kích thước và số liệu bổ sung:

- + Ngoài giá trị băm, có thể kiểm tra kích thước của tệp firmware hoặc các chỉ số đi kèm để phát hiện sự bất thường (đây là phương pháp bổ sung, không đủ độ tin cậy nếu sử dụng một mình).