GIAO THỨC GIAO TIẾP GIỮA ESP32 GATEWAY VỚI RTLS IOT CLOUD

BẢNG GHI NHẬN THAY ĐỔI

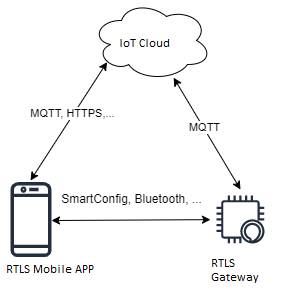
A – Tạo mới, M – Sửa đổi, D – Xóa bỏ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ngày  thay đổi | Vị trí  thay đổi | A\*  M, D | Người tạo | Phiên  bản cũ | Mô tả thay đổi | Phiên  bản mới |
| 20/04/2022 |  | A | Manhth |  | Tạo mới | 1.0 |
| 26/04/2022 |  | M | Manhth |  | Thêm chương 5 | 1.1 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Tính năng:

|  |  |  |
| --- | --- | --- |
| STT | Tính năng | Chi tiết |
| 1 | Kết nối IOT | Kết nối đến nền tảng RTLS IoT Cloud |
| 2 | Kết nối với các node trong mạng | Kết nối đến các UWB Nodes (Tag/anchor) |
| 3 | Giao tiếp giữa các node trong mạng | Truyền nhận dữ liệu giữa các Nodes. |
| 4 | Truyền nhận dữ liệu với Cloud | Gửi, nhận các dữ liệu về tọa độ, cấu hình cho Nodes |

# Mô hình kết nối.



* Thiết bị (Gateway) sử dụng giao thức MQTTS để kết nối đến Cloud.
* Các thông tin kết nối bao gồm:
  + Host: rtls.dinhvichinhxac.online
  + Port: 1883
  + User: unused
  + Password: ES256 JWT
  + Server Root CA.
  + ClientID:

***projects/{PROJECT\_ID} /locations/default/registries/{REGISTRY\_ID}/devices/{GATEWAY\_ID]***

* Các thông tin về Host, port, PROJECT\_ID, REGISTRY\_ID, GATEWAY\_ID được cấu hình qua local server trên Gateway hoặc qua RTLS mobile APP trong quá trình config.
* Password: sử dụng định dạng JSON Web Token, mã hóa ES256 hoặc RSA256 cùng với private key để gen.
* Gateway lưu trữ thông tin về Server CA Certificate và Private key trong bộ nhớ (format PEM). Các thông tin này có thể được config qua local server của gateway

## Cấu hình thiết bị sử dụng Smart config:

Diagram

Description automatically generated

* Device sau khi nhận SSID và PWD, kết nối thành công vào mạng WiFi sẽ mở http server tại port80, app sẽ kết nối vào port này để truyền nhận dữ liệu.
* APP gửi GET method vào API /api/info để nhận thông tin về gateway bao gồm gateway\_id, firmware\_version, hardware\_version, … APP sẽ check id của gateway để gửi đúng private key đã lưu trữ trên cloud cho gateway.
* APP gửi POST method vào các API sau để device lưu lại trong bộ nhớ.
  + /api/setup/project
  + /api/setup/cert
* Root ca và private\_key: mã hóa Base64.

|  |  |  |
| --- | --- | --- |
| API | App | Device |
| /api/info | GET | HTTP OK  {  "id": {id},  "model": "{model}",  "firmware": {fw},  "hardware": {hw},  "mac": "{mac}"  } |
| /api/setup/project | POST  {  "host": "{host}",  "port": {port},  "project\_id": "{project\_id}",  "location": "{location}",  "registry\_id": "{registry\_id}"  } | HTTP OK |
| /api/setup/cert | POST  {  "root\_ca":  {  "length": {ca\_length},  "data":"{root\_ca\_data}"  }  "private\_key":  {  "length": {private\_key\_length},  "data":"{private\_key\_data}"  }  } | HTTP OK |

## Cấu hình thiết bị sử dụng Smart config V2

## Cấu hình thiết bị sử dụng Bluetooth

# Device connectivity flowchart:

* Factory reset: Tín hiệu đầu đưa thiết bị về trạng thái Smartconfig, tín hiệu này có thể từ bất kỳ nguồn nào: Nút bấm, App, …

Diagram

Description automatically generated

# MQTT Protocol:

# MQTT Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Param name | Param type | Param value | Param description |
| mqttCliendId | String | ***projects/{PROJECT\_ID} /locations/default/registries/{REGISTRY\_ID}/devices/{GATEWAY\_ID]*** | **GATEWAY\_ID**: 4 byte MAC |
| mqttUser | String | ***unused*** | Don’t care |
| mqttPass | String | JSON Web Token (ES256/RS256 algorithm) |  |
| mqttHost | String |  | Chiều dài tối đa 64 bytes |
| mqttPort | Integer | Socket port | Port giao tiếp |
| mqttKeepAlive | Integer | 0 – 65535 (s) | Mặc định 120s |
| mqttSsl | Boolean | True/false | Kết nối có sử dụng ssl hoặc không |
| mqttCleanSession | Boolean | True/false |  |

## MQTT Topic

|  |  |  |
| --- | --- | --- |
| Topic name | Message direction | Description |
| /devices/{Gateway\_Id}/events | Device -> Cloud | Bản tin từ thiết bị gửi tới cloud |
| /devices/{Gateway\_Id}/state | Device -> Cloud | Bản tin từ thiết bị gửi tới cloud |
| /devices/{Gateway\_Id}/config | Cloud->Device | Bản tin từ cloud gửi tới thiết bị |
| /devices/{Gateway\_Id}/command/# | Cloud->Device | Bản tin từ cloud gửi tới thiết bị |
| /devices/{Node\_id}/events | Device -> Cloud | Bản tin từ thiết bị gửi tới cloud |
| /devices/{Node\_id}/state | Device -> Cloud | Bản tin từ thiết bị gửi tới cloud |

## MQTT Payload:

* Gói tin Gateway extend state

Graphical user interface, application

Description automatically generated with medium confidence

* Gói tin Gateway status

Background pattern

Description automatically generated with low confidence

* Gói tin gateway event:
* Gói tin gateway config:
* Gói tin gateway command:
* Gói tin Node attach:

Background pattern

Description automatically generated with low confidence

* Gói tin Node detach:

Background pattern

Description automatically generated with low confidence

* Gói tin Node events:

A picture containing background pattern

Description automatically generated

* Gói tin Node state:

Background pattern

Description automatically generated with low confidence

* Gói tin Node extend state:

A picture containing map

Description automatically generated

* Gói tin Node config:
* Gói tin Node command:

# Hoạt động của hệ thống.

* Lần đầu khởi động hệ thống, cần phải cấu hình cho **Node** + **Gateway**. Các **Node** ở đây bao gồm các **TAG** và **ANCHOR**
* Đối với **Gateway**: Sử dụng app trên điện thoại để kích hoạt cho **gateway**, kết nối **Gateway** tới mạng Wifi và cấu hình các thông số để **gateway** có thể kết nối đến Cloud. Gateway sẽ subcribes topic sau:

***/devices/{Gateway\_id}/config***

***/devices/{Gateway\_id}/command***

* Sau khi đã cấu hình cho **Gateway**, sử dụng Web Client (hoặc app trên điện thoại) để cấu hình cho các **node**.
* Tất cả các **Node** và **Gateway** cần cấu hình chung một mạng được đặc trưng bởi thông số networkID.
* Gateway->Cloud: các **node** cùng networkID với **gateway** mới được quyền gửi dữ liệu lên Cloud. Khi các **node** chưa được cấu hình networkID, chúng vẫn có thể kết nối với **Gateway** thông qua BLE nhưng sẽ không được phép gửi dữ liệu vào các topic của chúng trên Cloud.
* Cloud->Gateway: có thể gửi các gói tin cấu hình cho **Node** thông qua **Gateway**, **gateway** sẽ nhận, xử lí các gói tin và gửi các cấu hình đó đến Node tương ứng thông qua BLE.
* Các Node có 2 thông số đặc trưng:
  + Node Type: {TAG/ANCHOR}
  + Node ID: {1cdf/ab56…} ID mặc định được nsx cấu hình sẵn.
* Khi hoạt động, Gateway sẽ tạo một danh sách gọi là Device List gồm 32 slots (phụ thuộc vào tài nguyên của ESP32) chia làm 2 phần, phần đầu để dành cho ANCHOR, phần còn lại để dành cho TAG, đặc trưng bởi 2 thông số:
  + Slot Index: {1/2/3…32}
  + Slot Name: {tag1/tag2…tagN; anchor1/anchor2/…anchorN}
* Khi Gateway scan được bất kì kết nối BLE nào, nó sẽ kết nối đến Node tương ứng đó, căn cứ vào Node Type và Node ID, gateway sẽ kiểm tra và đưa Node đó vào 1 slot còn trống trong Device List để quản lí, đồng thời link giá trị Node ID với Slot Name tương ứng. Nếu Node kết nối với Gateway chưa được config thì Gateway sẽ gửi gói tin với method “new\_node” vào topic ***/devices/{Gateway\_id}/events***

{

“ts”: 123456789,

“gateway\_id”: “ESP32-987654321”,

“method”: “new\_node”,

“values”:

{

“node\_id”: “abcd”,

“node\_type”: “TAG”,

“slot\_name”: “tag-01”,

}

}

* Cấu hình Node qua Web Client, các thông tin cấu hình sẽ tùy thuộc vào Node Type và được gửi vào topic ***/devices/{Gateway\_id}/config.*** Gateway căn cứ vào node\_id trong topic để forward các gói tin cấu hình đến đúng Node yêu cầu. Trên Web Client, căn cứ vào slot\_name trong gói tin “new\_node” để lựa chọn device\_id cho đúng, thông số quan trọng nhất cần cấu hình ngay cho Node đó chính là network\_ID

{

“ts”: 123456789,

“gateway\_id”: “ESP32-987654321”,

“method”: “config”,

“values”:

{

“node\_id”: “abcd”,

“node\_type”: “TAG”,

“slot\_name”: “tag-01”,

“network\_id”: “net01”,

…….

}

}

* Gateway nhận, xử lí gói tin “config”, sau đó forward cấu hình đó sang Node tương ứng với Node\_id, đợi phản hồi từ Node, Nếu Node join vào network thành công, Gateway gửi gói tin “attach” vào topic ***/devices/{Gateway\_id}/events*** báo hiệu Node đó đã join được vào network.

{

“ts”: 123456789,

“gateway\_id”: “ESP32-987654321”,

“method”: “attach”,

“values”:

{

“node\_id”: “abcd”,

“node\_type”: “TAG”,

“slot\_name”: “tag-01”,

“network\_id”: “net01”,

…….

}

}

* Nếu Node đã được cấu hình “network\_ID” từ trước, khi join vào mạng thành công, Gateway cũng sẽ gửi gói tin attach tương ứng của node đó.
* Gateway gửi gói tin “state” vào topic ***/devices/{device\_id}/state*** thể hiện trạng thái online/offline của Node tương ứng.
* Các Node khi được add vào device\_list được kiểm soát thông qua thông số gọi là ***keep\_alive,*** nếu khoảng thời gian mất kết nối giữa Node và Gateway lớn hơn keep\_alive thì Node đó được coi là offline, Gateway sẽ gửi gói tin “detach” vào topic ***/devices/{Gateway\_id}/events*** báo hiệu Node đã out khỏi network, đồng thời gửi gói tin “state” vào topic ***/devices/{device\_id}/state*** thể hiện trạng thái offline của Node.
* Gateway gửi dữ liệu tọa độ của các TAG vào các topic tương ứng, ví dụ:

***/devices/tag-01/events*** hoặc ***/devices/tag-02/events***

* Gateway gửi các gói tin “extend\_state” vào topic ***/devices/{device\_id}/state*** khi nhận được gói tin “config” từ cloud.