

ESP Zigbee Gateway

本チュートリアルでは、Module Gateway H2 と CoreS3 主制御を組み合わせる ESP Zigbee Gateway サンプルプログラムを実行する方法を紹介します。ESP Zigbee Gateway は、ESP32 シリーズの Wi-Fi SoC と ESP32-H2 802.15.4 SoC に基づくゲートウェイデバイスであり、Zigbee ネットワークと Wi-Fi ネットワークを接続してスマートホームデバイスの相互接続を実現します。

1. 準備作業

- 1.環境設定: [ESP-IDF - ESP32S3 入門チュートリアル](#) を参照して、基本的なコンパイル環境を整えてください。

ESP-IDF

このサンプルのコンパイルには ESP-IDF バージョン **v5.3.1** の使用を推奨します。

```
git clone -b v5.3.1 --recursive https://github.com/espressif/esp-idf.git
cd esp-idf
./install.sh
. ./export.sh
```

- 2.Git コマンド `clone --recursive` を使用して ESP-Zigbee-SDK リポジトリを再帰的にクローンします

```
git clone --recursive https://github.com/espressif/esp-zigbee-sdk.git
cd esp-zigbee-sdk
```

- 3.使用するハードウェア製品:

- [ESP32 Downloader](#)
- [Module Gateway H2](#)
- [CoreS3](#)



- 4. 後続のチュートリアルで使用する `idf.py` コマンドはすべて ESP-IDF に依存します。コマンドを実行する前に ESP-IDF 内で `./export.sh` を実行して関連の環境変数をアクティベートしてください。詳細については [ESP-IDF - ESP32S3 入門チュートリアル](#) をご参照ください。

2. RCP ファームウェアのコンパイル

- 1.Gateway ファームウェアをコンパイルする前に、まず RCP ファームウェアを生成する必要があります。以下のコマンドを参考に、対応する RCP ファームウェアのディレクトリに移動し、ビルドターゲットを `esp32h2` に設定してください。

```
cd $IDF_PATH/examples/openthread/ot_rcp
idf.py set-target esp32h2
idf.py menuconfig
```

- 2.idf.py menuconfig を使用して設定画面に入り、menuconfig 内で Component config -> OpenThread RCP Example - Enable OPENTHREAD_NCP_VENDOR_HOOK を設定してください。
- 3. 設定が完了したら、以下のコマンドを実行して RCP ファームウェアをコンパイルします。

```
idf.py build
```

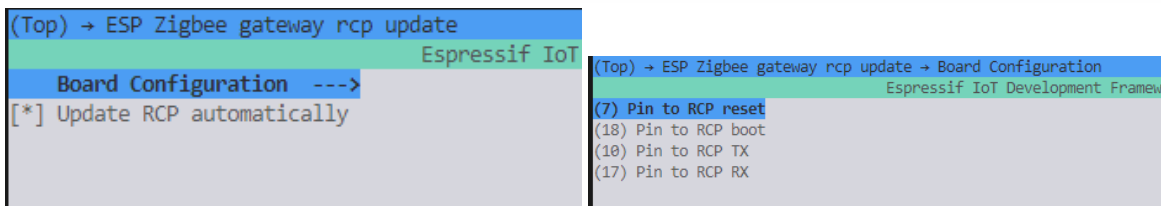
3. Gateway ファームウェアのコンパイル

- 1. esp_zigbee_gateway サンプルプログラムのディレクトリに移動し、ビルドターゲットを設定してください。

```
cd esp-zigbee-sdk/examples/esp_zigbee_gateway
idf.py set-target esp32s3
idf.py menuconfig
```

- 2.menuconfig 内で、ESP Zigbee gateway rcp update -> Update RCP automatically を有効にしてください。同時に正しい通信ピンを設定します。このピン設定は CoreS3 主制御用です。その他の主制御デバイスを使用する場合は、実際の状況に応じて変更してください。

- Board Configuration
 - Pin to RCP reset: 7
 - Pin to RCP boot: 18
 - Pin to RCP TX: 10
 - Pin to RCP RX: 17



- 3.Gateway Wi-Fi 接続設定

- Example Connection Configuration
 - WiFi SSID
 - WiFi Password

```
(Top) → Example Connection Configuration
[*] connect using WiFi interface
[ ] Get ssid and password from stdin
[*] Provide wifi connect commands
(xxxxxx) WiFi SSID
(xxxxxx) WiFi Password
(6) Maximum retry
    WiFi Scan Method (All Channel) --->
    WiFi Scan threshold --->
    WiFi Connect AP Sort Method (Signal) --->
[ ] connect using Ethernet interface
[ ] connect using Point to Point interface
[*] Obtain IPv6 address
    Preferred IPv6 Type (Local Link Address) --->
```

4. コンパイルとフラッシュ

```
idf.py build
idf.py erase_flash
idf.py flash
```

5. 実行開始

- CoreS3 と Module Gateway H2 を接続します。
- CoreS3 をパソコンに接続します。
- `idf.py monitor` またはその他のシリアルデバッグツールを使用して、115200bps の設定で実行ログを確認してください。

```
I (3551) wifi:set rx beacon pti, rx_bcn_pti: 0, bcn_timeout: 25000, mt_pti: 0, mt_time: 10000
I (3561) wifi:AP's beacon interval = 163840 us, DTIM period = 1
I (3561) wifi:<ba-add>idx:0 (ifx:0, c4:5e:5c:9b:63:f0), tid:6, ssn:2, winSize:64
I (3611) wifi:<ba-add>idx:1 (ifx:0, c4:5e:5c:9b:63:f0), tid:0, ssn:0, winSize:64
I (4651) esp_netif_handlers: example_netif_sta ip: 192.168.52.21, mask: 255.255.255.0, gw: 192.168.52.1
I (4651) example_connect: Got IPv4 event: Interface "example_netif_sta" address: 192.168.52.21
I (5461) example_connect: Got IPv6 event: Interface "example_netif_sta" address: fe80:0000:0000:0000:4a27:e2ff:fe78:fd74, type: ESP_IP6_ADDR_IS_LINK_LOCAL
I (5461) example_common: Connected to example_netif_sta
I (5461) example_common: - IPv4 address: 192.168.52.21
I (5471) example_common: - IPv6 address: fe80:0000:0000:0000:4a27:e2ff:fe78:fd74, type: ESP_IP6_ADDR_IS_LINK_LOCAL
I (5481) wifi:Set ps type: 0, coexist: 0

I (5481) ESP_ZB_GATEWAY: Initialize Zigbee stack
W (5511) ESP_ZB_GATEWAY: Network(0xb3f7) closed, devices joining not allowed.
I (5511) ESP_ZB_GATEWAY: Device started up in non factory-reset mode
I (5511) ESP_ZB_GATEWAY: Device rebooted
I (6121) ESP_ZB_GATEWAY: Network(0xb3f7) is open for 180 seconds
```

正常な実行ログの内容:

- RCP ファームウェアのバージョンチェック
- Wi-Fi 接続成功
- Zigbee ネットワークの作成成功
- ネットワークがオープンになり、デバイスの参加が許可される