

MicroPythonで無線制御

ご購入はこちら

西本 卓也

無線制御1：BLE

ESP32の特徴でもあるBluetooth Low Energy (BLE) 通信機能に注目します。BLEは、低消費電力で低コストの近距離無線通信の1つです。第1章でビルドしたMicroPythonではBLE通信ができます。

ここでは紛失防止タグで用いられているアラート・サービスをESP32ボードに実装して、BLE接続されたPCなどからLEDのON/OFFを制御してみます(図1)。

● ESP32ボードはペリフェラルとして使う

BLEのGeneric Access Profile (GAP) では、ペリフェラル、セントラル、ブロードキャスト、オブザーバの4種類が定義されています。ここではESP32ボードをペリフェラルとして、iOSデバイスやLinuxマシンなどをセントラルとして使います。

● プロトコルはImmediate Alert

BLEのGeneric Attribute Protocol (GATT) として、ここでは紛失防止タグなどで使われるImmediate Alert (アラート・サービス) を実装します。セントラルからペリフェラルに対して1バイトの属性を書き込むと、アラートの開始や停止ができます。

アラート・サービスの仕様は次の通りです。

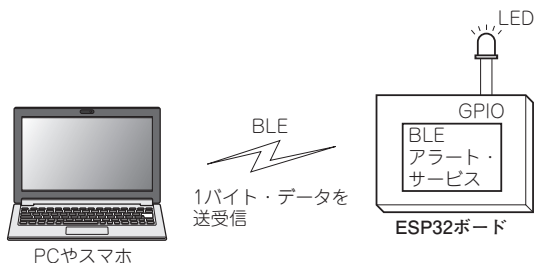


図1 BLE接続されたPCなどからESP32ボードのLEDのON/OFFを制御する

```
>>> import alert
>>> alert.main()
```

図2 アラート・サービスの実行

- 属性はサービスが1802, キャラクタリスティックが2a06
- 0以外の値を書き込むとLEDが点灯, 0を書き込むと消灯

リスト1のMicroPythonコードをalert.pyというファイル名でESP32ボードに書き込み、図2のように実行します。

データの送信

● iPhoneからデータを送る

Apple社のiPhone, iPod touch, iPadがある場合はLightBlue Explorerという無料アプリ(開発元PunchThrough社)をAppStoreからダウンロードして実行します(図3)。

起動するとPeripherals Nearby画面が表示されESP32が見つかります(図3①)。

ESP32を選択するとConnected画面になります(図3②)。Immediate Alertサービスが表示されるので、Alert Levelを選択します。

Alert Levelの設定画面(図3③)で、Write new valueを選択し、Edit Value画面(図3④)を使って送信するデータを入力します。1を入力し[Done]を押すとESP32ボードのLEDが点灯します。

LightBlueの最初の画面に移動すると、ESP32ボードはアドバタイズ状態に戻ります。

また、このalert.pyをESP32ボード起動時に自動実行させるには、main.py(リスト2)を使います。

● Linux PCからデータを送る

LinuxやRaspbianの環境では、Bluetooth関連ツールのBluezを使います(図4)。

あらかじめ、ESP32のアドレスを調べて記録してお