

Pico W アプリの開発方法②

…Arduino IDE

関本 健太郎

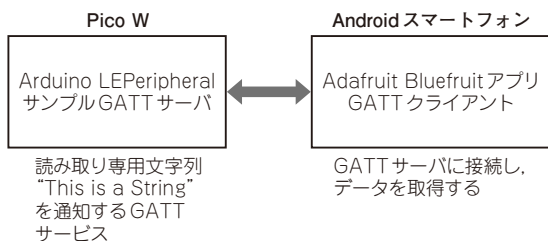
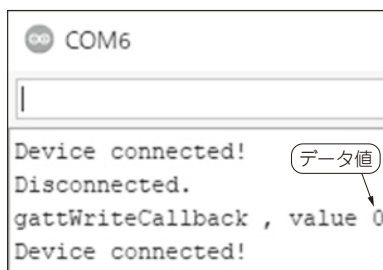


図1 LEPeripheral サンプルのシステム構成

リスト1 GATTサービス定義ファイル

```
// setup GATT Database
BTstack.addGATTService(new UUID("B8E06067-62AD-41BA-9231-206AE80AB551"));
BTstack.addGATTCharacteristic(new UUID("f897177b-ae8-4767-8ecc-cc694fd5fcef"), ATT_PROPERTY_READ,
                                "This is a String!");
BTstack.addGATTCharacteristicDynamic(new UUID("f897177b-ae8-4767-8ecc-cc694fd5fcef"), ATT_PROPERTY_READ |
                                        ATT_PROPERTY_WRITE | ATT_PROPERTY_NOTIFY, 0);
```

図2



Arduino シリアル・モニタ画面 LEPeripheral サンプルのスケッチにあるデータ値を通知するキャラクターリスティックで定義したデータ値を表示している

この章では、マイコン開発で定番環境として使われている Arduino IDE を使ってラズベリー・パイ Pico W の Bluetooth アプリケーション開発を行います。例として、①BLE のサンプルである LEPeripheral と、②マイコンと PC 間で通信するためのシリアル・プロトコル Firmata を取り上げます。(編集部)

サンプル①：文字列とデータ値のキャラクターリスティックを定義する

この章では、ラズベリー・パイ Pico/Pico W (以降、Pico/Pico W) 向けの Arduino IDE である Arduino-pico 環境⁽¹⁾に含まれる、BTstack ライブラリの LEPeripheral サンプルを取り上げます(なお、LEPeripheral サンプルの拡張例は、第1部第8章を参照)。本稿では、Arduino IDE は v2.1.1 を利用しました⁽²⁾。

● LEPeripheral サンプルの実行

Arduino-pico 環境付属の BTstack ライブラリのサンプル・スケッチとして、ANCS、iBeacon、iBeacon Scanner、LECentral、LEPeripheral が提供されています。サンプル・スケッチの中から、LEPeripheral を実行してみます。LEPeripheral スケッチ(図1)には、

BLE パリフェラルのサービスとして動作し、文字列 "This is a String!" を表示するキャラクターリスティックと、データ値('H')を通知するキャラクターリスティックが定義されています(リスト1)。データ値は Arduino IDE のシリアル・モニタにも表示されます(図2)

ビルドと実行の手順は次の通りです。

- (1) Arduino IDE のメニュー・バーから [ツール]-[ボード]-[ボードマネージャ] で [Raspberry Pi Pico/RP2040] を選択し、ボード・ライブラリをインストールする
- (2) [ツール]-[ボード]-[Raspberry Pi RP2040 Boards (3.2.0)]-[Raspberry Pi Pico W] を選択する
- (3) [ツール]-[IP/Bluetooth Stack: "IPv4 Only"] で [IPv4 Bluetooth] を選択する
- (4) [ファイル]-[スケッチ例]-[BTstack]-[LEPeripheral] を選択する。
- (5) [スケッチ]-[検証・コンパイル] でプログラムをコンパイルする

正常にコンパイルできたことを確認し、Pico W のリセット・ボタンを押しながら USB ケーブルをホスト PC につなぎ、ホスト PC からドライブとして認識できる状態にします。