

# E-Kit EnOcean IoT ゲートウェイ開発キット 取扱説明書



株式会社デバイスドライバーズ

目次

1. はじめに.....	3
1.1. 製品概要.....	3
2. 準備.....	4
2.1. マイクロ SD カードの交換.....	4
3. ログインとコンパイル.....	5
3.1. ログイン.....	5
3.2. ブートローダー.....	5
3.2.1. ブートローダーの場所と設定.....	6
3.3. EnOcean 無線通信.....	6
3.3.1. ソースコードの場所.....	6
3.3.2. コンパイル.....	6
3.3.3. インストール.....	6
3.4. Azure 通信.....	7
3.4.1. ソースコードの場所.....	7
3.4.2. 準備.....	7
3.4.3. コンパイル.....	7
3.4.4. インストール.....	8
3.5. Azure IoT Hub サポート.....	8
3.5.1. ソースコードの場所.....	8
3.5.2. コンパイル.....	8
3.5.3. インストール.....	9
3.6. OPC UA 通信.....	9
3.6.1. ソースコードの場所.....	9
3.6.2. コンパイル.....	9
3.6.3. インストール.....	9
3.7. Web GUI.....	10
3.7.1. ソースコードの場所.....	10
3.7.2. 実行手順.....	10
4. サポートと保証.....	10

## ***Device Drivers***

### **1. はじめに**

このたびは、**E-Kit EnOcean** ゲートウェイ開発キットをご購入頂きまして、ありがとうございます。本製品の取り扱いについては、別紙「**E-Kit EnOcean Gateway 取扱説明書**」をご参照ください。

#### **1.1. 製品概要**

本製品は、**E-Kit** ゲートウェイソースコードと社内利用ライセンスのセット販売です。ビルド手順書と、そのまま運用システムをビルドできるビルドシステムを収めたマイクロ **SD** カードが付属します。**E-Kit** ゲートウェイのカスタマイズや、同様の製品を購入者が開発して社内ですることができる。次のソースコードの開示に際しては、各ライセンス提供元の許諾が必要です。

- **iBRESS Cloud / SKKYNET ETK**

iBRESS Cloud / SKKYNET ETK 部分の開示には株式会社ベルチャイルドの許諾が必要です。

- **EnOcean GP**

EnOcean GP 部分の開示には **EnOcean Alliance** への参加と許諾が必要です。

- **ファームウェア更新機能**

ファームウェア更新部分の開示には株式会社デバイスドライバーズとの **OEM** ライセンスの締結（有償）が必要です。

その他の **EnOcean** 通信、**Web GUI**、**OPC UA** 通信等の部分は、本開発キットにソースコードが全て含まれています。

開発キットをご購入後3年間、最新ソースコードを受け取ることができます。購入者の社内利用に関しては、上記条件のほかには制約がありませんので、自由に改造して利用することが可能です。

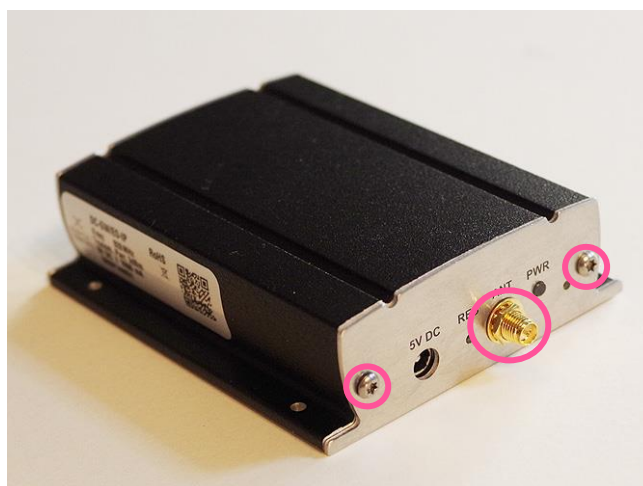
このソースコードに基づいた製品を社外向けに提供または販売される場合には、別途株式会社デバイスドライバーズとの **OEM** ライセンスの締結（有償）が必要です。

### 2. 準備

開発キットでは、E-Kit EnOcean ゲートウェイ本体を使用して開発を行います。

#### 2.1. マイクロ SD カードの交換

開発キットを利用する際は、最初に本装置の電源を停止した状態で、電源ケーブルを外して、内蔵のマイクロ SD カードを開発キットと交換する必要があります。交換は次の手順で行います。



① アンテナ側パネルの取り外し

アンテナを取り外し、同じ面にあるアンテナワッシャーと 2 か所あるトルクス「T10H」のネジを取り外して、パネルを外します。

② マイクロ SD カードの交換

中に緑色の基板が上下 2 段重なった状態であるので、その上側の基板に付いているマイクロ SD カードを開発キットのものと交換します。

③ アンテナ側パネルの取り付け

マイクロ SD カード交換後は、パネルを元通りに取付けて、アンテナワッシャーとトルクスネジを締めた後、アンテナを取り付けます。

### 3. ログインとコンパイル

開発に使用するためのログインとコンパイル等の手順を示します。開発環境は Raspberry-Pi 標準の Debian 系 OS である Raspbian の環境をそのまま利用しています。したがって、システムモジュールの追加や更新などは、一般的な Raspbian の利用方法がそのまま踏襲できます。

#### 3.1. ログイン

E-Kit EnOcean ゲートウェイで開発を行うためには、Windows や Linux 等の他のマシンから、LAN 経由で E-Kit EnOcean ゲートウェイリモートログインして作業します。

ログインは、TeraTerm や Linux コンソール等の端末エミュレータや端末ソフトウェアから、IP アドレスを指定して行います。IP アドレスは UPnP により取得可能ですので、詳細手順は「E-Kit EnOcean Gateway 取扱説明書」を参照してください。

ユーザー名は、Raspberry Pi 標準の「pi」、パスワードも Raspberry Pi 標準の「raspberrypi」に設定してありますので、必要に応じて変更してください。これらは、Web ユーザーインターフェースとは全く異なる設定となっています。一般配布のマイクロ SD カードのログインユーザー名とパスワード、公開していません。

以降は次の各モジュールの設定、またはコンパイルとインストール手順を示します。

- ブートローダー
- EnOcean 無線通信
- Microsoft Azure 通信
- OPC UA 通信
- Web UI

#### 3.2. ブートローダー

E-Kit EnOcean Gateway では、起動時に各ソフトウェアを自動的に立ち上げるため、次のローカルブートローダーの Shell スクリプトで保持しています。したがって、モジュールの入れ替え等、開発を行う場合には次の手順で、この各ソフトウェアの自動実行機能を無効にする必要があります。

## *Device Drivers*

### 3.2.1. ブートローダーの場所と設定

- ブートローダーの場所

```
/etc/rc.d/egw.sh
```

- ブートローダー無効化（自動実行しない）

```
$ sudo mv /etc/rc.d/egw.sh /etc/rc.d/egw-back.sh
```

システムファイルのため管理者権限で実行します。設定後、システムを再起動します。

- ブートローダー有効化（上記設定後、自動実行する様に再設定）

```
$ sudo mv /etc/rc.d/egw-back.sh /etc/rc.d/egw.sh
```

システムファイルのため管理者権限で実行します。設定後、システムを再起動します。

### 3.3. EnOcean 無線通信

E-Kit EnOcean Gateway は、次の EnOcean 動作モードに対応した、EnOcean センサーの受信機能を提供します。

#### 3.3.1. ソースコードの場所

```
/home/pi/devgw/dpride
```

#### 3.3.2. コンパイル

Makefile を使用して次のコマンドでコンパイルします。

```
$ cd
```

```
$ cd devgw/dpride
```

```
$ make
```

オブジェクトを一掃する場合は `make clean` コマンドを実行します。

```
$ make clean
```

#### 3.3.3. インストール

Makefile を使用して次のコマンドでインストールします。コンパイルした各モジュールを「/usr/local/bin」ディレクトリコピーすることで、実行可能とします。

```
$ sudo make install
```

管理者権限で実行します。実行の際も、次の例の様に管理者権限が必要です。

```
$ sudo dpride -o
```

## *Device Drivers*

### 3.4. Azure 通信

Microsoft Azure 通信は、Azure iot-edge (V1)のサンプルモジュールである「simulated\_device\_cloud\_upload」を改変して実現しているため、基本的な使い方は次のURLを参照してください。

[https://github.com/Azure/iot-edge-v1/tree/master/v1/samples/simulated\\_device\\_cloud\\_upload](https://github.com/Azure/iot-edge-v1/tree/master/v1/samples/simulated_device_cloud_upload)

#### 3.4.1. ソースコードの場所

`/home/pi/github/iot-edge`

#### 3.4.2. 準備

「simulated\_device\_cloud\_upload」モジュールの実行では、このモジュールの構造がGitHubからダウンロードしたディレクトリをそのまま絶対パスで使用するため、実行ファイルの場所ソースコードの場所を開発時に切り替えています。

`/home/pi/github/iot-edge/v1`

が実行モジュールの場所のため、現在の実行モジュールをそのまま保持して開発を行うためには、開発時に一時的にディレクトリ名を変更します。

`simulated_device_cloud_upload_sample` を実行中の場合は、`kill` コマンドで停止します。

- 実行モジュールの一時保存

```
$ cd
```

```
$ cd github/iot-edge
```

```
$ mv v1 v1-bin
```

- ソースコードモジュールのディレクトリ設定（前項実施後）

```
$ mv v1-src v1
```

#### 3.4.3. コンパイル

shell スクリプトを使用して次のコマンドでコンパイルします。

```
$ cd
```

```
$ cd github/iot-edge/v1
```

```
$ sh ./mk
```

## *Device Drivers*

### 3.4.4. インストール

コンパイルしたモジュールをインストールする場合は、前項のコマンド実行後、`shell` スクリプトを使用して次のコマンドでインストール先 (`github/iot-edge/v1-inst`) にオブジェクトをコピーします。

```
$ mkdir ../v1-inst
$ sh ./inst
```

この手順でコンパイル・ビルドした動作に必要なバイナリーとシェアライブラリを、「`/home/pi/github/iot-edge/v1-inst`」に抽出したので、次の手順でディレクトリを入れ替えます。

```
$ cd
$ cd github/iot-edge/
$ mv v1 v1-src
  (コンパイルで使用した v1 ディレクトリを元の名前に戻します)
$ mv v1-inst v1
  (コンパイル・ビルドしたモジュールを v1 ディレクトリとして設定します)
```

### 3.5. Azure IoT Hub サポート

Microsoft Azure IoT Hub のデバイス管理に対して、EnOcean の LEARN ボタンと Teach-In テレグラムによる自動メンテナンス機能 (デバイス側で LEARN ボタンを押下した際に、自動的に Azure 上の IoT Hub 上に必要な全 Device ID を生成) を実現するために、サポートモジュールを開発して使用しています

#### 3.5.1. ソースコードの場所

```
/home/pi/devgw/iothub
```

#### 3.5.2. コンパイル

Makefile を使用して次のコマンドでコンパイルします。

```
$ cd
$ cd devgw/ iothub
$ make
```

オブジェクトを一掃する場合は `make clean` コマンドを実行します。

```
$ make clean
```



## *Device Drivers*

### 3.5.3. インストール

Makefile を使用して次のコマンドでインストールします。コンパイルした各モジュールを「/usr/local/bin」ディレクトリコピーすることで、実行可能とします。

```
$ sudo make install
```

### 3.6. OPC UA 通信

OPC UA 通信は、Open62541 のサンプルモジュールである「open62541」を改変して実現しているため、基本的な使い方は次の URL を参照してください。ただし弊社でこのモジュールを E-Kit EnOcean Gateway 様に移植してから、このサンプルやライブラリの構成が大幅に変更になりました。現在の E-Kit EnOcean ゲートウェイでは、open62541 の古いバージョンを使用しているため、改変の際はご注意ください。

<https://github.com/open62541/open62541>

#### 3.6.1. ソースコードの場所

```
/home/pi/devgw/open62541
```

#### 3.6.2. コンパイル

Makefile を使用して次のコマンドでコンパイルします。

```
$ cd  
$ cd devgw/open62541  
$ make
```

オブジェクトを一掃する場合は `make clean` コマンドを実行します。

```
$ make clean
```

#### 3.6.3. インストール

インストール用 Shell スクリプトを実行してインストールします。コンパイルした各モジュールを「/usr/local/bin」ディレクトリコピーすることで、実行可能とします。

```
$ sudo sh ./inst
```

システムファイルのため管理者権限で実行します。

管理者権限で実行します。実行の際も、次の例の様に管理者権限が必要です。

```
$ sudo EnOceanJob
```

## *Device Drivers*

### 3.7. Web GUI

E-Kit EnOcean Gateway 搭載の Web GUI は、「Node.js」を使用して開発しています。Node.js はインタプリタ動作のため、ソースコードがそのまま実行ファイルとなりますので、注意してください。従って改変の際は、事前に古い構成ソースコードファイルを「tar」等の手段でバックアップ、保管することをお勧めします。

#### 3.7.1. ソースコードの場所

```
/home/pi/newgw/node/modules
```

#### 3.7.2. 実行手順

手作業で実行する場合は、次のコマンドを入力します。

```
$ cd  
$ cd newgw/node/modules  
$ sudo node app.js
```

システムファイルにアクセスするため、管理者権限で実行します。

## 4. サポートと保証

本製品は、開発者用のキットのため、本説明書の通りに実行したが動作しないなどの、不具合以外のサポートは、全て有償でのサポートとなります。最新版のソースコードは一部 GitHub でも公開していますが、ご確認が必要の際は、株式会社デバイスドライバーズ E-Kit 事業部「[e-kit@devdrv.co.jp](mailto:e-kit@devdrv.co.jp)」宛にメールでお問い合わせください。

株式会社デバイスドライバーズ

E-Kit 事業部

© 2019 Device Drivers, Ltd.