

# Technical Note

TecStar

## Silicon Labs 社 Bluetooth Xpress アドバンスガイド(上級編)

---

2018 年 12 月

株式会社 **マクニカ**  
テクスター カンパニー

#### 目次

|                                     |    |
|-------------------------------------|----|
| 1 はじめに .....                        | 3  |
| 2 設定変更する .....                      | 4  |
| 2-1 アドバタイズ .....                    | 4  |
| 2-2 コネクション .....                    | 8  |
| 2-3 スキャン .....                      | 10 |
| 2-4 セキュリティ .....                    | 14 |
| 2-5 出力レベル .....                     | 17 |
| 2-6 UART ボーレート .....                | 19 |
| 2-7 デバイス名 (Device Name) .....       | 21 |
| 3 動作制御する .....                      | 24 |
| 3-1 エラーコード .....                    | 24 |
| 3-2 GPIO (ステータス信号) .....            | 25 |
| 4 消費電流を最適化する .....                  | 28 |
| 4-1 消費電流の簡易測定 .....                 | 28 |
| 4-2 UART ボーレート変更 .....              | 32 |
| 4-3 スリープモード (sleep コマンド) .....      | 34 |
| 4-4 スリープモード (sleep_select ピン) ..... | 35 |
| 4-5 出力レベルの変更 .....                  | 36 |
| 5 FAQ .....                         | 39 |
| 5-1 仕様・使用方法 .....                   | 39 |
| 5-2 トラブルシューティング .....               | 41 |
| 参考文献 .....                          | 42 |

## 1 はじめに

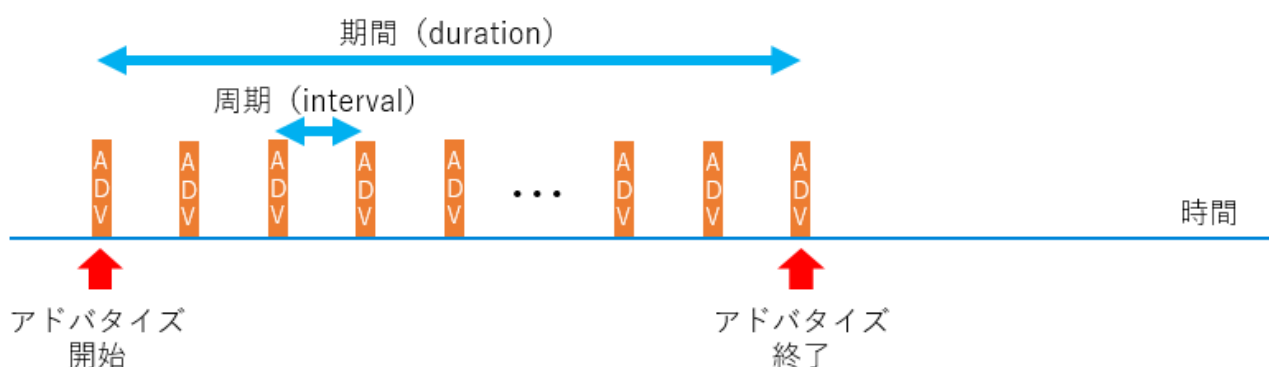
この資料は、Silicon Laboratories（以下、Silicon Labs）社製 Bluetooth®モジュール Bluetooth Xpress（以下、BGX）の使用方法について簡易にまとめたものです。内容に誤りがないよう注意は払っておりますが、もし Silicon Labs 社が提供するドキュメント等と差異がございましたら、メーカー提供のものを優先してご参照ください。

本資料は、基本的な使用方法是理解頂いている方を対象とした**アドバンスガイド(上級編)**です。初めて BGX をご使用になる方は、まず**クイックスタートガイド(初級編)**をご参照ください。

## 2 設定変更する

### 2-1 アドバタイズ

BLE では、セントラル側とペリフェラル側を接続し、それからデータ通信を行います。接続するにあたり、ペリフェラル側はアドバタイズという信号を一定間隔で送信し、セントラル側はアドバタイズ信号をスキャンすることでペリフェラル側を発見します。



アドバタイズの周期が短ければ、セントラル側に見つけて貰うのが早くなりますし、またアドバタイズの期間が長ければ、セントラル側に見つけて貰える可能性が高まります。しかしながら、アドバタイズの周期が短くしたり、期間を長くしたりすると、その分 消費電力も増えることになります。

BGX では、アドバタイズの周期 (interval) と期間 (duration) を変更できるようになっています。周期・期間の変更は、**bl v h d**, **bl v h i**, **bl v l d**, **bl v l i** という variable を変更することで実現できます

アドバタイズの周期と期間の組み合わせを 2 つ保持できるようになっていますので、計 4 個の variable があります。(2 つのセットは、High と Low という名称になっています)

| Variable 名 | 説明                                                |
|------------|---------------------------------------------------|
| bl v h d   | BLE adVertising High Duration (high 設定時のアドバタイズ期間) |
| bl v h i   | BLE adVertising High Interval (high 設定時のアドバタイズ周期) |
| bl v l d   | BLE adVertising Low Duration (low 設定時のアドバタイズ期間)   |
| bl v l i   | BLE adVertising Low Interval (low 設定時のアドバタイズ周期)   |

<advertising High の variable>

## bl v h d

BLE adVertising High Duration

### Description

Configures high mode advertising duration. This is the duration in seconds for which advertising continues after issuing the `adv high` command. See [BLE Advertising](#).

| Attribute | Description           |
|-----------|-----------------------|
| format    | integer               |
| units     | seconds               |
| min       | 0 (advertise forever) |
| max       | 1000                  |
| default   | 30                    |
| access    | get/set               |
| save      | yes                   |

## bl v h i

BLE adVertising High Interval

### Description

Configures high mode advertising interval, used for the `adv` command `high` option. The interval is measured in slots. For example, an interval of 32 means advertise in slot 0, then 32, then 64 and so on. Valid range: 32 - 16384 slots inclusive.

| Attribute | Description |
|-----------|-------------|
| format    | integer     |
| units     | slots       |
| min       | 32          |
| max       | 16384       |
| default   | 32          |
| access    | get/set     |
| save      | yes         |

<advertising Low の variable>

## blvld

BLE adVertising Low Duration

### Description

Configures low mode advertising duration. This is the duration in seconds for which advertising continues after issuing the `adv low` command. See [BLE Advertising](#).

| Attribute | Description           |
|-----------|-----------------------|
| format    | integer               |
| units     | seconds               |
| min       | 0 (advertise forever) |
| max       | 1000                  |
| default   | 0 (forever)           |
| access    | get/set               |
| save      | yes                   |

## blvli

BLE adVertising Low Interval

### Description

Configures low mode advertising interval, used for the `adv` command `low` option. The interval is measured in slots. For example, an interval of 32 means advertise in slot 0, then 32, then 64 and so on. Valid range: 32 - 16384 slots inclusive.

| Attribute | Description |
|-----------|-------------|
| format    | integer     |
| units     | slots       |
| min       | 32          |
| max       | 16384       |
| default   | 874         |
| access    | get/set     |
| save      | yes         |

書式は、

**set bl v h d 設定値**

**set bl v h i 設定値**

です。期間(duration)の設定値の単位は秒です。周期(interval)の設定値の単位は slot で、1 slot = 0.625 秒で換算ください。

```
> set bl v h d 40
Success
> set bl v h i 50
Success
>
```

アドバタイズを開始するには **adv** コマンドを使用します。このコマンドの引数で、high 設定を使ってアドバタイズするのか、low 設定を使ってアドバタイズするのか、を指定できます。書式は、

**adv 設定値**

です。設定値には、**high**, **low**, **off** が入ります。high 設定を使う場合は **adv high** とします。

```
> adv high
Success
>
```

adv

Advertise as a peripheral

Description

Turn on advertising as a peripheral at the specified rate. The command `adv off` turns advertising off. If no argument is supplied, the default is `adv high`.

On reset, advertising defaults to high for a duration specified by `bl v h d` (default: 30 seconds), then switches to low for a duration specified by `bl v l d` (default: always on), then turns off.

The advertising settings correspond to the following advertising modes.

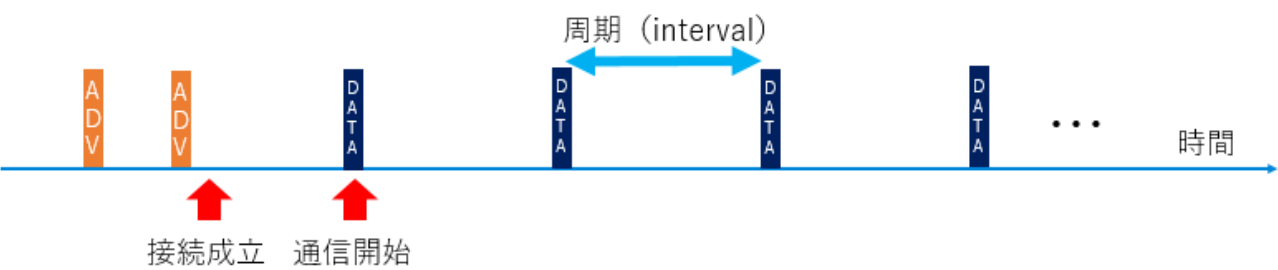
- `high` - High Duty Cycle Undirected Advertising
- `low` - Low Duty Cycle Undirected Advertising
- `off` - No Advertising

Syntax

```
> adv [low | high | off]
```

2-2 コネクション

接続が成立すると、一定間隔ごとにデータの送受を行います。この周期 (interval) をコネクション・インターバルといいます。



コネクション・インターバルが短ければ、多くのパケットを送ることができ、スループットが高くなります。しかしながら、送るデータがない場合にも通信を維持するために空パケットを送りますので、スリープする期間もその分短くなり、消費電力も増えることになります。

BGX では、コネクション・インターバルを変更できるようになっています。変更は `bl c i` という variable を変更することで実現できます

`bl c i`

BLE Connection Interval

Description

BLE connection interval. See [Bluetooth specification : Important Connection Parameters : Connection Interval, Supervision Timeout](#)

| Attribute | Description    |
|-----------|----------------|
| format    | integer        |
| units     | 1.25 ms        |
| min       | 6 (7.5 ms)     |
| max       | 3200 (4000 ms) |
| default   | 12 (15 ms)     |
| access    | get/set        |
| save      | yes            |



書式は、

## set bl c i 設定値

です。設定値の単位は 1.25ms です。例えば設定値=12 の場合、インターバルは  $12 * 1.25\text{ms} = 15\text{ms}$  となります。

```
> set bl c i 12
Success
>
```

BLE では、コネクション・インターバルに従ってデータ通信を行います。通信に失敗するとすぐに接続が解除される、という仕様ではありません。しかしながら、一定期間、通信が行えなかった場合には、タイムアウト (Supervision Timeout) が発生して、接続が解除されます。タイムアウトまでの期間は、**bl c t** という variable で変更することが可能です。

デフォルトでは、タイムアウトは 4 秒に設定されています。

### bl c t

BLE Connection Timeout

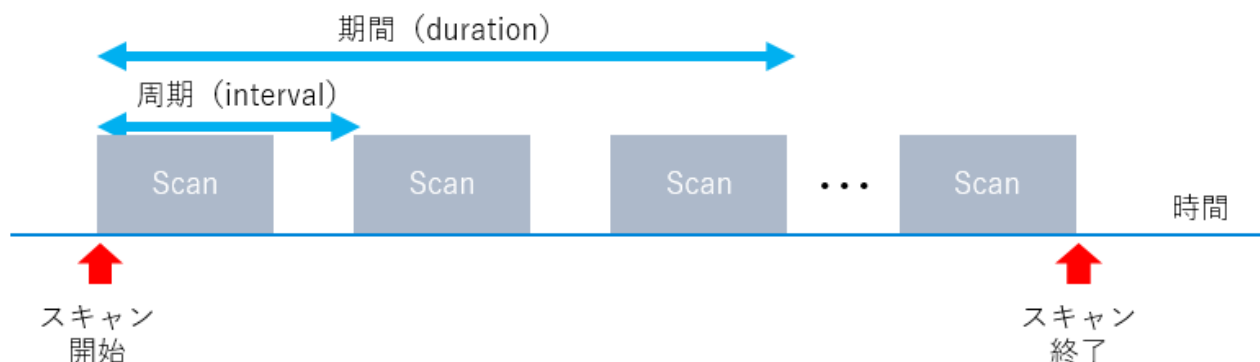
#### Description

BLE connection supervision timeout. See [Bluetooth specification : Important Connection Parameters : Connection Interval, Supervision Timeout](#).

| Attribute | Description     |
|-----------|-----------------|
| format    | integer         |
| units     | 10 ms           |
| min       | 10 (100 ms)     |
| max       | 3200 (32000 ms) |
| default   | 400 (4000 ms)   |
| access    | get/set         |
| save      | yes             |

## 2-3 スキャン

BLE では、セントラル側とペリフェラル側を接続し、それからデータ通信を行います。接続するにあたり、ペリフェラル側はアドバタイズという信号を一定間隔で送信し、セントラル側はアドバタイズ信号をスキャンすることでペリフェラル側を発見します。



スキャンの周期が短ければ、ペリフェラルを見つけるのが早くなりますし、またスキャンの期間が長ければ、ペリフェラルを見つける可能性が高まります。しかしながら、スキャンの周期を短くしたり、期間を長くしたりすると、その分 消費電力も増えることになります。

BGX では、スキャンの周期 (interval) と期間 (duration) を変更できるようになっています。周期・期間の変更は、`ce s h d`, `ce s h i`, `ce s l d`, `ce s l i` という variable を変更することで実現できます

スキャンの周期と期間の組み合わせを 2 つ保持できるようになっていますので、計 4 個の variable があります。(2 つのセットは、High と Low という名称になっています)

| Variable 名            | 説明                                           |
|-----------------------|----------------------------------------------|
| <code>ce s h d</code> | Central Scan High Duration (high 設定時のスキャン期間) |
| <code>ce s h i</code> | Central Scan High Interval (high 設定時のスキャン周期) |
| <code>ce s l d</code> | Central Scan Low Duration (low 設定時のスキャン期間)   |
| <code>ce s l i</code> | Central Scan Low Interval (low 設定時のスキャン周期)   |

<scan High の variable>

## ce s h d

Central Scan High Duration

### Description

Configures central scan high mode duration. This is the duration in seconds for which scan continues after issuing the [scan high](#) command.

| Attribute | Description      |
|-----------|------------------|
| format    | integer          |
| units     | seconds          |
| min       | 0 (scan forever) |
| max       | 1000             |
| default   | 30               |
| access    | get/set          |
| save      | yes              |

## ce s h i

Central Scan High Interval

### Description

Configures central high mode scanning interval, used for the [scan](#) command [high](#) option.

| Attribute | Description |
|-----------|-------------|
| format    | integer     |
| units     | slots       |
| min       | 4           |
| max       | 16384       |
| default   | 96          |
| access    | get/set     |
| save      | yes         |

<scan Low の variable>

## ce s l d

Central Scan Low Duration

### Description

Configures central low mode scanning duration. This is the duration in seconds for which scanning continues after issuing the `scan low` command.

| Attribute | Description      |
|-----------|------------------|
| format    | integer          |
| units     | seconds          |
| min       | 0 (scan forever) |
| max       | 1000             |
| default   | 300              |
| access    | get/set          |
| save      | yes              |

## ce s l i

Central Scan Low Interval

### Description

Configures low mode scanning interval, used for the `scan` command `low` option.

| Attribute | Description |
|-----------|-------------|
| format    | integer     |
| units     | slots       |
| min       | 4           |
| max       | 16384       |
| default   | 2048        |
| access    | get/set     |
| save      | yes         |

書式は、

**set ce s h d 設定値**

**set ce s h i 設定値**

です。期間(duration)の設定値の単位は秒です。周期(interval)の設定値の単位は slot で、1 slot = 0.625 秒で換算ください。

```
> set ce s h d 40
Success
> set ce s h i 1000
Success
>
```

スキャンを開始するには **scan** コマンドを使用します。scan コマンドについては、クイックスタートガイドを参照ください。このコマンドの引数で、high 設定を使ってスキャンするのか、low 設定を使ってスキャンするのか、を指定できます。high 設定を使う場合は **scan high** とします。

```
> scan high
Success
>
```

## 2-4 セキュリティ

LE security mode 1 では 4 段階のセキュリティレベルが定義されており、BGX は Level 2,3,4 のセキュリティをサポートしています。

| Level | LE security mode 1<br>(Bluetooth 仕様書での定義)                   | 補足                                                                                                  |
|-------|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| 1     | No security                                                 | セキュリティなし                                                                                            |
| 2     | Unauthenticated pairing with encryption                     | 暗号あり・認証なしのペアリング<br>(BGX の説明ページでは、Unauthenticated "Just works" encryption with no passkey と説明されています) |
| 3     | Authenticated pairing with encryption                       | 暗号あり・認証ありのペアリング<br>(BGX の説明ページでは、Authenticated encryption with a passkey と説明されています)                 |
| 4     | Authenticated LE Secure Connections pairing with encryption | LE Secure Connections を使ったペアリング                                                                     |

BGX は以下の 2 つの鍵タイプを使用した encryption をサポートしています。Numeric comparison に対応していません。

- Just Works (keyless)
- 6 digit pin code (passkey)

セキュリティレベルの変更は、blek, blep という variable を変更することで実現できます。デフォルトでは Level 4 のセキュリティが有効になっています。

| Level | blek 設定                | blep 設定 |
|-------|------------------------|---------|
| 2     | none                   | any     |
| 3     | ピンコード(6 digit)         | any     |
| 4     | none or ピンコード(6 digit) | secure  |

any を選択した場合、まずは LE Secure Connection (Level.4) での接続を試み、接続相手に対応していなければ Legacy Pairing (Level.3 or 2) での接続を試みます。

secure を選択した場合、LE Secure Connection (Level.4) での接続を試み、接続相手に対応していなければ、接続エラー (Security mismatch) を返します。

## ble k

BLE Encryption Key

### Description

The BLE encryption key can be set to **none** or to a 6 digit pin. Entering more or less than 6 digits, or entering a pin with non-numeric characters other than 'none' will cause the command to fail. Encryption is always enabled on the Bluetooth Xpress module.

The key type determines the pairing procedure and security mode and level. See [Security](#).

**Note:** if the key is set to **none**, and an Android phone requests a pin, use **000000**.

| Attribute | Description               |
|-----------|---------------------------|
| format    | 6-digit integer or 'none' |
| units     | n/a                       |
| min       | 000000                    |
| max       | 999999                    |
| default   | 'none'                    |
| access    | get/set                   |
| save      | yes                       |

## ble p

BLE Encryption Pairing (mode)

### Description

Selects whether to allow LE Legacy Pairing. If the connecting device supports LE Secure Connections (BLE 4.2 or newer) it will be used by default. Setting to **any** will allow pairing with either secure or legacy methods while **secure** requires secure connections pairing. Encryption is always enabled on Bluetooth Xpress modules.

| Attribute | Description                                    |
|-----------|------------------------------------------------|
| format    | string enumeration                             |
| units     | 'secure'=secure only<br>'any'=secure or legacy |
| default   | secure                                         |
| access    | get/set                                        |
| save      | yes                                            |

ボンディング(交換した鍵情報を保存し、次回接続を簡単にする機能)の有効／無効は、**ble b**という variable で行います。

## ble b

BLE Encryption Bondable (mode)

### Description

Enables or disable bonding. Encryption is always enabled for the Bluetooth Xpress module.

| Attribute | Description                      |
|-----------|----------------------------------|
| format    | integer                          |
| units     | 1=true/enabled, 0=false/disabled |
| min       | 0                                |
| max       | 1                                |
| default   | 1                                |
| access    | get/set                          |
| save      | yes                              |

なお、セキュリティについては、専用 WEB サイトで詳しい説明があります。

<https://docs.silabs.com/bgx/latest/security>



## 2-5 出力レベル

BGX の出力レベルは+8dBm(最大値)に設定されています。出力レベルが高ければ通信距離が長くなりますが、その代わりに消費電流も大きくなります。そのため、通信距離が短くて良い用途では、出力レベルを下げて使用することも選択肢になります。

出力レベルの変更は、**bl t c** という variable を変更することで実現できます。

**bl t c**

BLE Transmit Connected (power)

### Description

Configures the RF transmit power when connected.

| Attribute | Description |
|-----------|-------------|
| format    | integer     |
| units     | dBm         |
| min       | -25         |
| max       | 8           |
| default   | 8           |
| access    | get/set     |
| save      | yes         |

### Get example

```
> get bl t c
8
```

### Set example

```
> set bl t c 4
Success
```

書式は、

## set bl t c 出力レベル

です。出力レベルには、8（最大）から -25（最少）までの整数が入ります。



```
[COMMAND_MODE]
> get bl t c
8
> set bl t c -16
Success
> get bl t c
-16
>
```

設定した値が通信距離に与える影響については、非常にざっくりとしたイメージですが、値が 8 減るごとに通信距離が半分になる、程度で考えて頂くのが判りやすいかと思います。（8 を基準として、0 だと距離 1/2、-8 だと距離 1/4、-16 だと距離 1/8）

厳密なところは、実測するなどしてお確かめください。

## 2-6 UART ボーレート

BGX の制御インタフェースとして UART が使用されますが、ボーレートはデフォルトで 115200bps に設定されています。

ボーレートの変更は、**ua b** という variable を使用します。

**ua b**

UART Baud

Description

Sets the UART baud rate.

| Attribute | Description     |
|-----------|-----------------|
| format    | integer         |
| units     | bits per second |
| min       | 0               |
| max       | 2000000         |
| default   | 115200          |
| access    | get/set         |
| save      | yes             |

Get example

```
> get ua b
115200
```

Set example

```
> set ua b 115200
Success
```

書式は、

**set ua b ボーレート**

です。ボーレートには、0 から 2000000 までの整数が入ります。

```
> get ua b
115200
> set ua b 230400
Success
```

設定値を変更しても実際のボーレートはまだ変わっておらず、旧設定のまま動作しています。新しい設定値を使用して UART をリスタートするには、**uartu** コマンドを使用します。書式は、

**uartu**

です。

## uartu

UART initialization

### Description

Initializes UART with new UART-related settings. This command can be used to change UART settings at runtime without a device reset. When this command is executed, UART-settings stored in the Bluetooth Xpress module will take effect.

### Syntax

```
> uartu
```

### Example

```
> uartu
Success
```

リスタートすると、ボーレートが変更されます。BGX とターミナルソフトのボーレート設定が合致していないので、ターミナルソフト上の文字が文字化けするようになります。

```
> uartu
Success
> 艱紆■缺■
```

ターミナルソフトのボーレート設定を変更すると、また BGX と正常に通信が行えるようになります。

## 2-7 デバイス名 (Device Name)

スマートフォンなどで Bluetooth デバイスを検索した際に表示される文字列がデバイス名です。このデバイス名は、変更可能になっており、変更には **sy d n** という variable を使用します。

### sy d n

System Device Name

#### Description

Bluetooth device name, up to 16 characters in length. The last 2 to 6 characters from the **BD\_ADDR** may be substituted for # wildcards supplied in the final characters of the name. See the examples below.

| Attribute | Description          |
|-----------|----------------------|
| format    | string               |
| min       | --                   |
| max       | 16 characters length |
| default   | 'BGX-####'           |
| access    | get/set              |
| save      | yes                  |

#### Get example

```
> get bl d
4C55CCABCDEF
> get sy d n
BGX-CDEF
```

#### Set example

In the following examples the BD\_ADDR address is **4C55CCABCDEF** (see [bl a](#))

```
> set sy d n ACK-##
Success
> get sy d n
ACK-EF
> set sy d n BGX###
Success
> get sy d n
BGXDEF
> set sy d n my#####
Success
> get sy d n
myABCDEF
```

デフォルトでは BGX-#### になっており、####には BLE アドレスの下位 4 キャラクタが入ります。同時に複数の Bluetooth デバイスを使用した場合に、デバイス名が全く同じだと見分ける術がありませんので、異なる名称になるようにユニークな BLE アドレスをデバイス名に利用しています。下図の例では、BLE アドレスの下位 4 キャラクタは“8D86”ですので、デバイス名は“BGX-8D86”になっています。

```
[COMMAND_MODE]
> get bl a
D0CF5E828D86
> get sy d n
BGX-8D86
>
```

変更する際の書式は、

**set sy d n 新デバイス名**

です。

デバイス名には#(ワイルドカード)が使用できます。#は 0～12 個の範囲で使用でき、BLE アドレスの下位のキャラクタが割り当てられます。# が 5 個であれば BLE アドレスの下位 5 キャラクタを、#が 12 個であれば BLE の全 12 キャラクタを使用します。# を未使用とすることもできます。

```
> get bl a
D0CF5E828D86
> set sy d n BgxWirelessGecko
Success
> get sy d n
BgxWirelessGecko
> set sy d n #
Success
> get sy d n
6
> set sy d n #####
Success
> get sy d n
D0CF5E828D86
> set sy d n #####-Bgx
Success
> get sy d n
828D86-Bgx
>
```

#未使用

#1 個使用

#12 個使用

#を先頭に使用

ただし注意点として、アドバタイズに使用されるのは上位 8 キャラクタのみです。つまり、スマートフォンなどで表示されるデバイス名も上位 8 キャラクタとなります。例えば、デバイス名として“BGX-12345”(計 9 キャラクタ)を設定したとすると、上位 8 キャラクタ“BGX-1234”のみが表示されます。

この点を踏まえて、デバイス名を決定ください。

WEBドキュメントにも以下のように記載されています。

<https://docs.silabs.com/bgx/latest/ble-advertising>

**Note:** The maximum length of the name field in the BGX advertising packet is 8 characters. If the name of an advertising BGX (sy d n) exceeds 8 characters in length, it will be shortened in the advertising packet to its first 8 characters.

For example, the name:

BGX-12345

will appear in the advertising packet as:

BGX-1234

### 3 動作制御する

#### 3-1 エラーコード

BGX にコマンドを送ると、何かしらのレスポンスが返ってきます。もっとも多く目にするレスポンスは、コマンドが正常に受領されたことを知らせる **Success** ですが、それ以外にもエラーコードが規定されています。

エラーコードは、WEB ドキュメントにも記載されています。

<https://docs.silabs.com/bgx/latest/serial-interface>

#### Response Format

Responses from Bluetooth Xpress module follow the format shown in the text below.

```
RXXXXXX\r\n
<response data>
```

where:

- **R** denotes a Response header.
- **X** is the error code (response error codes are listed below).
- **XXXXXX** is the response data length in bytes including a trailing `\r\n`. The response data length is zero if no data is available, or >2 (including `\r\n`) if data is available.
- **<response data>**. If the response data length is >0, the **<response data>** is the data returned from a Bluetooth Xpress module in response to the command.

#### Response Error Codes

| Code | Description                | From Commands |
|------|----------------------------|---------------|
| 0    | Success                    | all           |
| 1    | Command failed             | all           |
| 2    | Parse error                | all           |
| 3    | Unknown command            | all           |
| 4    | Too few arguments          | all           |
| 5    | Too many arguments         | all           |
| 6    | Unknown variable or option | all           |
| 7    | Invalid argument           | all           |
| 8    | Timeout                    | con           |
| 9    | Security mismatch          | con           |
| 10   | GATT Procedure Failed      | con           |



### 3-2 GPIO（ステータス信号）

GPIO にステータス信号を割り当てることで、BGX の状態を GPIO 出力することができます。設定には **gfu** コマンドを使用します。

**gfu**

Select GPIO function

**Description**

Configure a GPIO with the specified function. A function may only be assigned to a pin that has a function set to **none** i.e. the pin is not already assigned.

A list of available functions is shown in the following table.

| I/O | Function         | Description                                                                                               |
|-----|------------------|-----------------------------------------------------------------------------------------------------------|
| 0   | con_active       | Asserts when BLE is connected and encrypted. (active high)                                                |
| 0   | con_active_n     | Asserts when BLE is connected and encrypted. (active low)                                                 |
| 0   | con_status_led   | Blinks when BLE is connected and encrypted. The blink pattern is controlled with the <b>sys</b> variable. |
| 0   | str_active       | Asserts when BLE is connected, encrypted, and stream mode is active. (active high)                        |
| 0   | str_active_n     | Asserts when BLE is connected, encrypted, and stream mode is active. (active low)                         |
| 0   | str_activity_led | Asserts when BLE is connected, encrypted, and stream mode data is being transmitted or received.          |

BGX Starter Kit にはボタンや LED が実装されています。どの GPIO に、どのボタン／LED が割り当てられているかは、ユーザガイド (BGX13P starter kit の場合は UG369) に記載されています。

| Table 5.1. LEDs and Buttons Connected to BGX13P |            |
|-------------------------------------------------|------------|
| BGX13P Pin                                      | LED/Button |
| GPIO0                                           | LED0       |
| GPIO1                                           | LED1       |
| GPIO2                                           | BTN0       |
| GPIO3                                           | BTN1       |

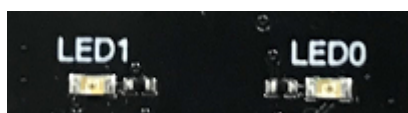
LED0 に **str\_active\_n** を、LED1 に **con\_active\_n** を割り当ててみます。LED0／LED1 は Low Active になっているので、ここでは **\*\*\*\_n** を使用しています。connected, encrypted, stream mode の条件を満たした時に LED0 が、connected, encrypted の条件を満たした時に LED1 が点灯するのが期待値です。

また、モード遷移を制御するために str\_select があると便利なので、ボタン 0 に str\_select を割り当てておきます。

```
> gfu 0 none
Success
> gfu 0 str_active_n
Success
> gfu 1 none
Success
> gfu 1 con_active_n
Success
> set bu s c edge
Success
> gfu 2 none
Success
> gfu 2 str_select
Success
>
```

まだBGXはBLE接続しておらず、Commandモードで動作していますので、LED0／LED1は消灯したままです。

消灯



消灯

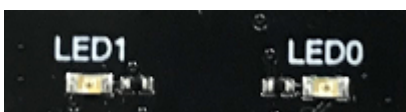
|             |   |
|-------------|---|
| Connected   | × |
| Encrypted   | × |
| Stream mode | × |

ここで、ボタン0を押して、Streamモードに移行します。

```
> STREAM_MODE
```

stream modeの条件を満たしましたが、BLE接続(connected, encrypted)の条件を満たしていないので、となっているので、LED0／LED1は消灯したままです。

消灯

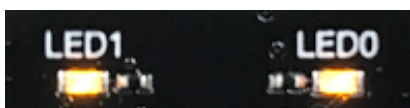


消灯

|             |   |
|-------------|---|
| Connected   | × |
| Encrypted   | × |
| Stream mode | ○ |

スマホアプリを使用して、BGXと接続をします。connected, encrypted, stream modeの条件を満たしたので、LED0／LED1はどちらも点灯します。

点灯



点灯

|             |   |
|-------------|---|
| Connected   | ○ |
| Encrypted   | ○ |
| Stream mode | ○ |

ここで、ボタン0を押して、Commandモードに移行します。

```
> STREAM_MODE
COMMAND_MODE
```

stream mode の条件を満たさなくなったため、LED0 が消灯します。connected, encrypted の条件は満たしたままなので、LED1 は点灯したままです。

点灯

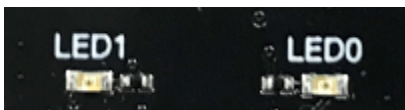


消灯

|             |   |
|-------------|---|
| Connected   | ○ |
| Encrypted   | ○ |
| Stream mode | × |

スマホアプリから切断します。connected, encrypted の条件を満たさなくなったため、LED1 も消灯します。

消灯



消灯

|             |   |
|-------------|---|
| Connected   | × |
| Encrypted   | × |
| Stream mode | × |

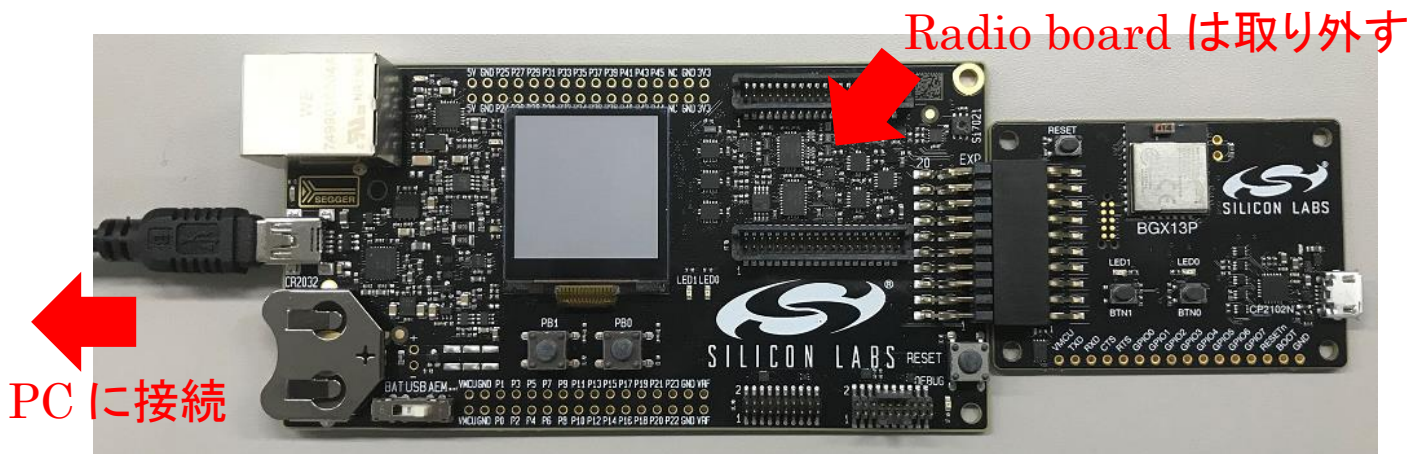
## 4 消費電流を最適化する

### 4-1 消費電流の簡易測定

Wireless Starter Kit main board（以後、WSTK）と Simplicity Studio を使用することで、簡単に消費電流を測定することができます。

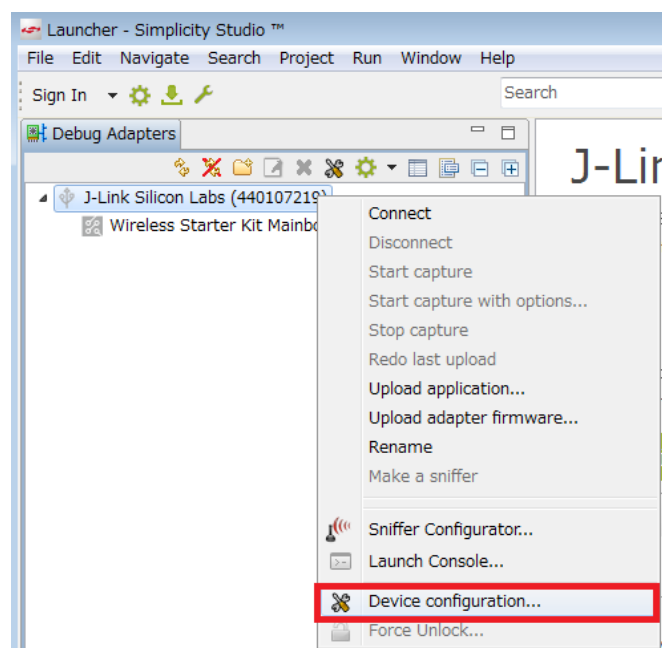
手順は以下の通りです。

- ① WSTK と BGX13P Starter Kit を下図のように接続します。そして WSTK を PC に接続します。

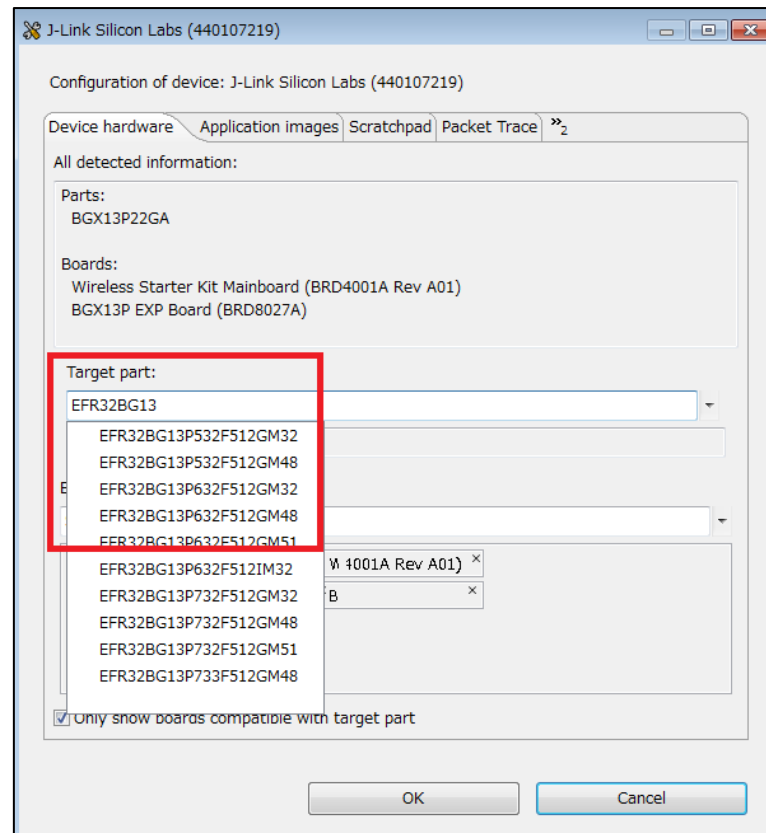


- ② Simplicity Studio を起動します。（Simplicity Studio および Bluetooth SDK のインストール方法は、BGM1xx クイックスタートガイドをご参照ください。）

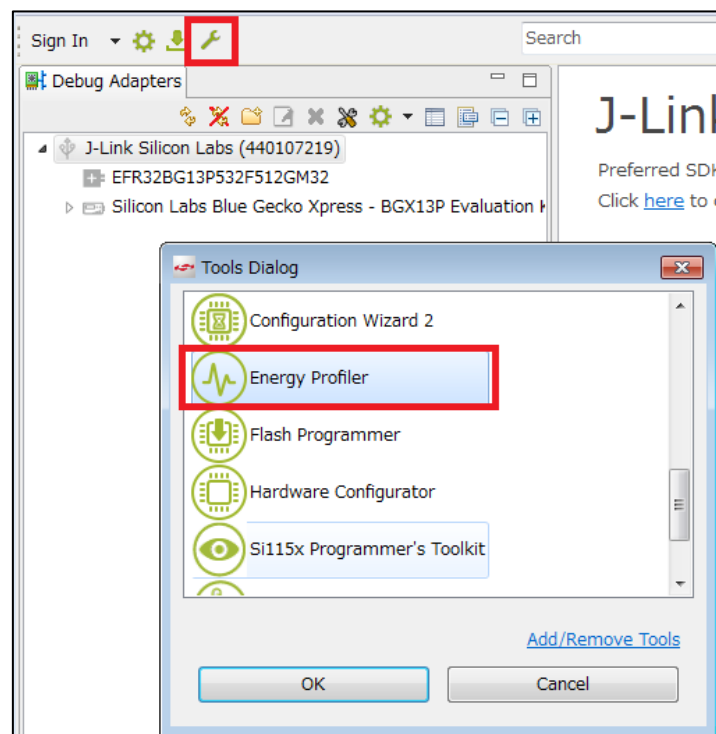
Debug Adapters に J-Link が表示されるので、選択して右クリックし、Device Configuration を選択します。



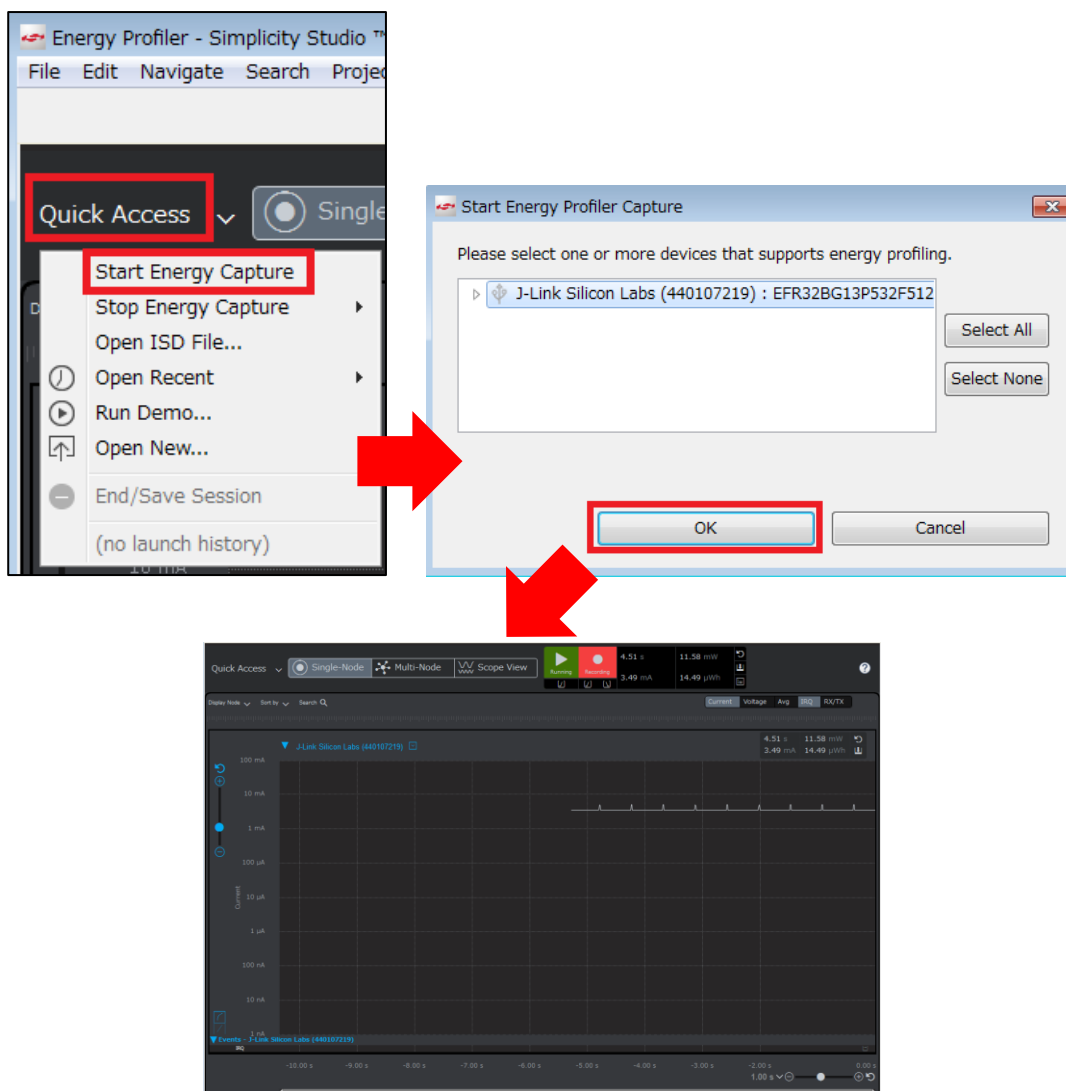
③ Simplicity Studio は BGX13P の電流測定には対応していないので、電流測定に対応している別の型番 (BGM1xx や EFR32) を選択します。下図では EFR32BG13... を選択しています。OK を押します。



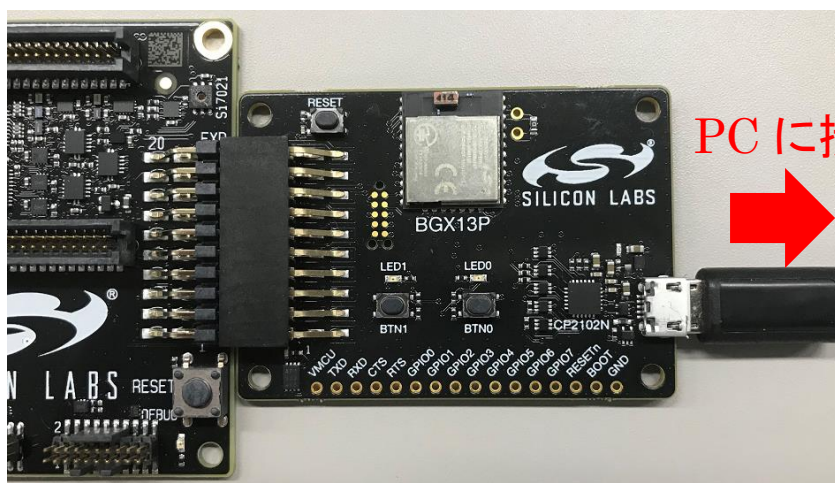
④ 再度 J-Link を選択し、Tool メニューから Energy Profiler を起動します。



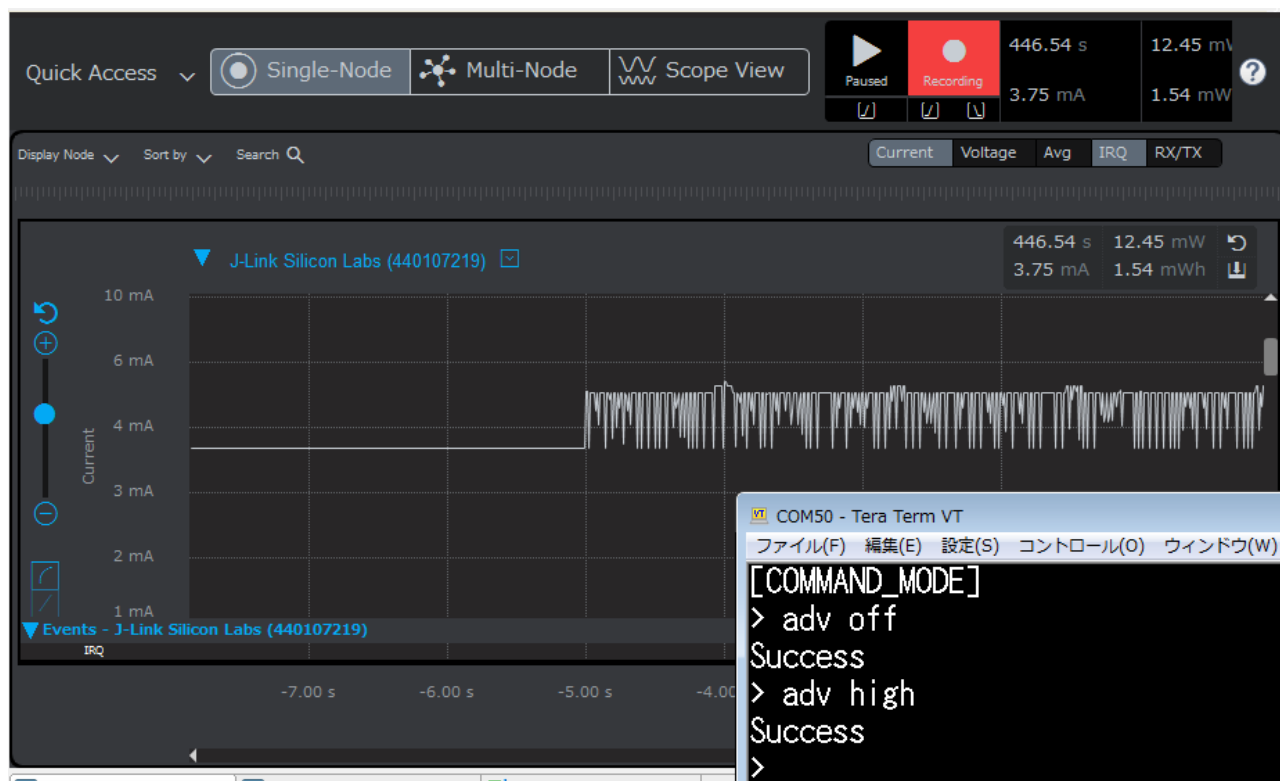
⑤ Energy Profiler が起動したら、Quick Access から Start Energy Capture を選択し、J-Link を選んで OK します。すると、電流測定がスタートします。



⑥ このままでは、BGX13P を制御できませんので、BGX13P Starter Kit 側の USB を PC に接続します。



⑦ ターミナルソフトからBGX13Pを制御します。制御に連動して消費電流値が変化することが確認できます。下図はアドバタイズを OFF し、続いて ON した場合の消費電流波形です。



Simplicity Studio および Energy Profiler の使用方法については、Silicon Labs 社ドキュメント および 弊社 BGM1xx クイックスタートガイドをご参照ください。

なお、本資料で電流波形を掲載する際には、動作を理解しやすくするために、サンプリング期間を長くして取得した波形を使用している場合があります。サンプリングした値が平均化されるため、ピーク電流が見えづらくなっていますので、より正確な値が知りたい場合には拡大してご確認ください。



## 4-2 UART ボーレート変更

Active モード時の消費電流は、9600bps を境に以下のように規定されています。大きな差があります。

| Parameter                                 | Symbol                   | Test Condition       | Min | Typ  | Max | Unit |
|-------------------------------------------|--------------------------|----------------------|-----|------|-----|------|
| Active supply current, Un-connected, Idle | I <sub>ACTIVE_IDLE</sub> | Baud rate ≤ 9600 bps | —   | 3    | —   | μA   |
|                                           |                          | Baud rate > 9600 bps | —   | 3.25 | —   | mA   |

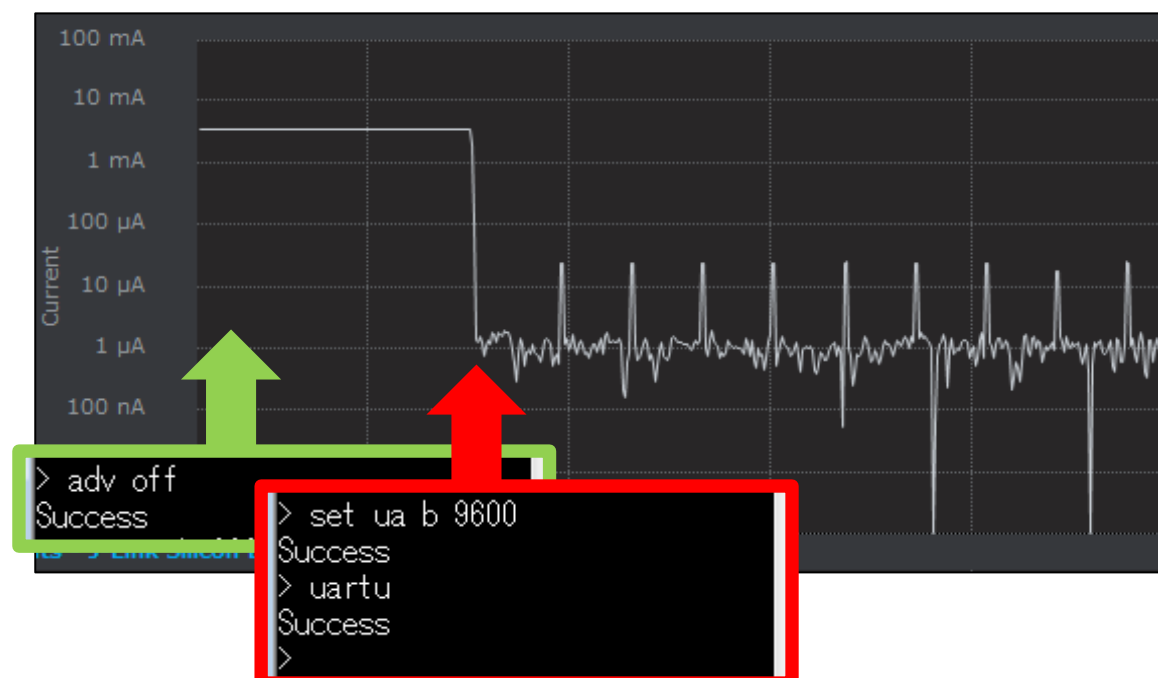
低速ボーレート(9600bps 以下)に設定した場合、BGX は Active モードと Sleep モードの間を自動的に行き来しながら、消費電流を下げるような振る舞いをします。これは BGX の構造に起因するものですので、9600bps を超える高速ボーレートに設定した場合には、このような振る舞いは行いません。

ですから、ボーレートを 9600bps 以下に下げること、消費電流の削減が期待できます。

ボーレートを下げることで、消費電流が削減できる実例をご紹介します。理解を助けるために、敢えて UART ボーレートを動的に変更しています。

### ■ 例1

アイドル(アドバタイズOFF)の状態、ボーレートを 115200 から 9600 に変更した場合の電流波形です。自動的に Sleep に移行するため、消費電流が大きく下がります。この状態でも UART 通信は生きています。





本当に UART 通信が活きているかを確認するために、UART にデータを送ってみます。(ターミナルソフト上でリターンキーを押します) BGX は瞬間的に Active モードに移行し(一時的に消費電流が大きくなる)、必要な処理を行い、再び Sleep に移行したことが確認できます。この動作を自動的に行ってくれます。



## ■ 例 2

アドバタイズ中に、ボーレートを 115200 から 9600 に変更した場合の電流カーブです。アドバタイズは維持しつつ、自動的に Sleep に移行する動作も行ってくれるため、同様に消費電流が大きく下がります。



## 4-3 スリープモード（sleep コマンド）

UART ボーレートを低速に設定している場合、UART 通信は維持しつつ、Active モードと Sleep モードの間を自動的に行き来しながら、消費電流を下げるように振る舞ってくれます。しかしながら、高速ボーレートで使用している場合には、同様の振る舞いを行いません。

このような場合、**sleep** コマンドを使用することで、BGX をスリープ状態にすることができます。書式は、

**sleep**

です。

### sleep

Enter sleep mode

**Description**

Put the module into the lowest-power sleep state. The module sleeps until a wakeup event occurs such as an interrupt on the `str_select` GPIO. See [Power Management](#).

**Syntax**

```
> sleep
```

**Example**

```
> sleep
Success
```

sleep コマンドを実行すると、消費電流が下がることが確認できます。なお、Sleep モードに移行すると UART からのコマンドを受け付けなくなるので、復帰方法を考えておく必要があります。



## 4-4 スリープモード（sleep\_select ピン）

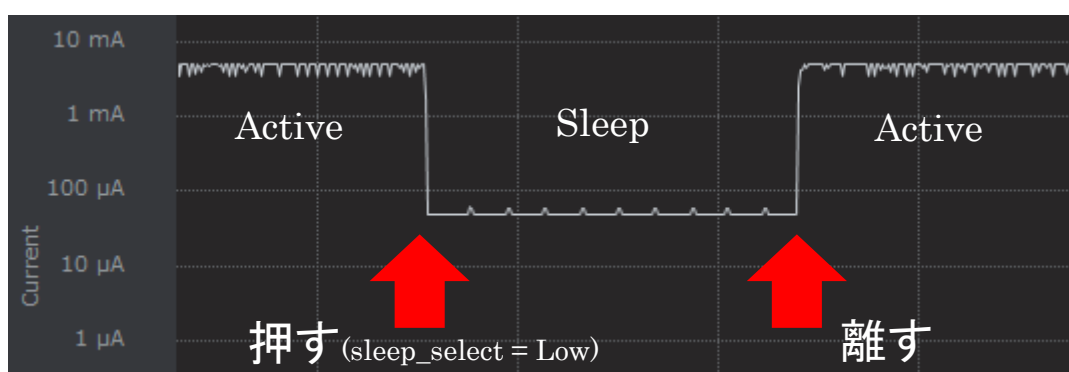
GPIO に sleep\_select 機能を割り当てることで、Sleep モード／Active モードの遷移を GPIO 制御することができます。設定には **gfu** コマンドを使用します。

| gfu                                                                                                                                                                 |              |                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------|
| Select GPIO function                                                                                                                                                |              |                                                                                                                         |
| <b>Description</b>                                                                                                                                                  |              |                                                                                                                         |
| Configure a GPIO with the specified function. A function may only be assigned to a pin that has a function set to <b>none</b> i.e. the pin is not already assigned. |              |                                                                                                                         |
| A list of available functions is shown in the following table.                                                                                                      |              |                                                                                                                         |
|                                                                                                                                                                     | sleep_select | Active low input that forces the module into sleep mode when asserted and wakes the module from sleep when de-asserted. |

BGX Starter Kit にはボタンが実装されていますが、ボタン 0 (BTN0) は BGX の GPIO2 と繋がっています。このボタン 0 に sleep\_select 機能を割り当ててみます。

```
> gfu 2 none
Success
> gfu 2 sleep_select
Success
>
```

ボタン 0 を押せば Sleep モードに、ボタン 0 を離せば Active モードに移行します。



sleep コマンド使用時と比べて、Sleep モード時の電流値が若干高くなっているように見えますが、これはボタンを押したことによる電流増加分です。BGX そのものは同じ電流値となっています。

## 4-5 出力レベルの変更

BGX で最も消費電流が高くなるのは BLE 通信中で、特に送信時に高くなります。下図はアドバタイズ時の電流波形です。スケールを拡大しないとピーク電流が読み取れないので、拡大しています。

デフォルト設定(+8dBm)では、ピークで 27mA 程度流れていることが確認できます。



平均電流で考えれば、送信時の電流値の影響は大きなものではありません。ただし、電池やバッテリーで動作する機器ではピーク電流が気になる場合もあるかと思います。その際には、出力レベルを下げるのが効果的です。

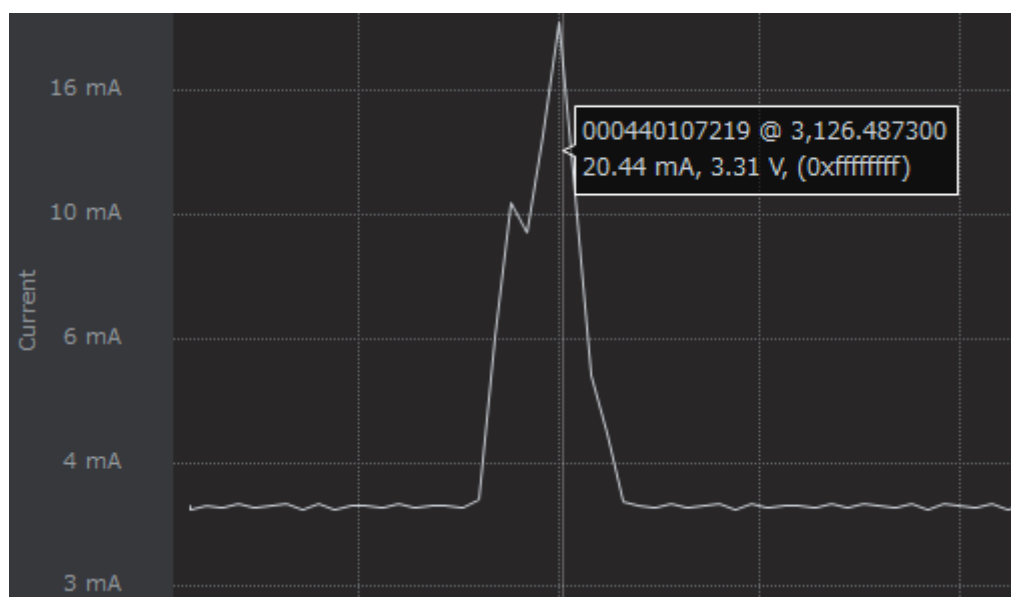
出力レベルを下げることで、消費電流が削減できる実例をご紹介します。ただし、出力レベル低減による効果が望めるのは 0 dBm 程度までで、それ以下に出力レベルを下げてても効果は期待できません。

## ■ 例1

出力レベルを+4 dBm に下げ、アドバタイズ時の消費電流を見ます。ピーク電流が 21mA 程度まで下がりました。

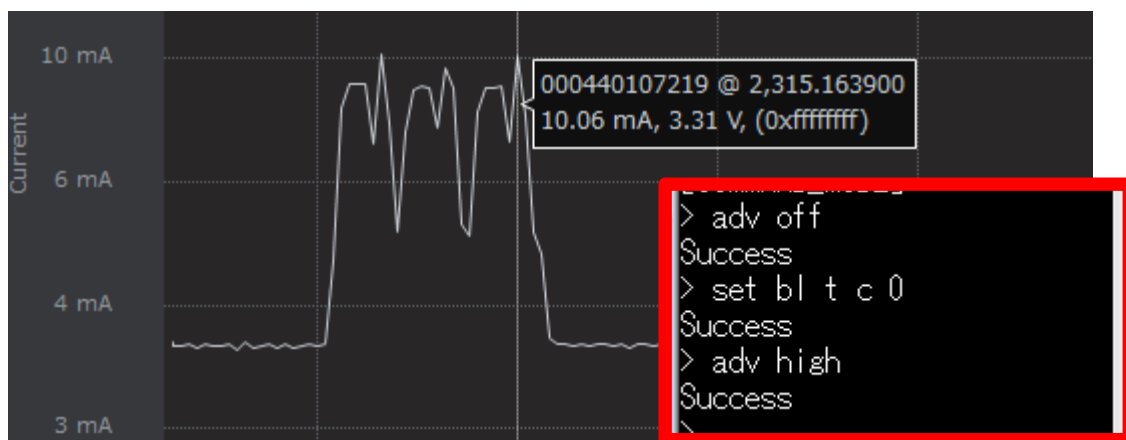


その後、スマホから接続して通信時の消費電流を見ますが、同程度のピーク電流になっています。



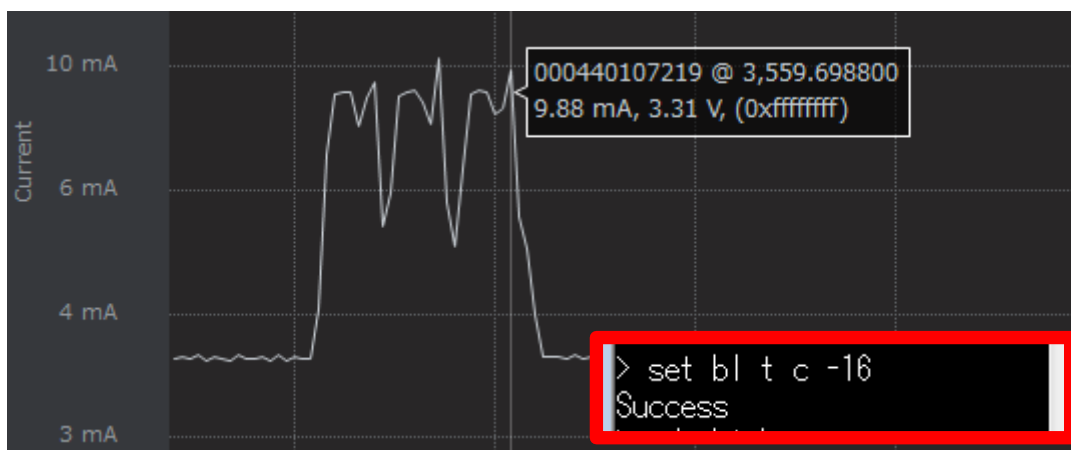
## ■ 例 2

出力レベルを 0 dBm に落としてみます。ピーク電流が 10mA 程度に下がりました。



## ■ 例 3

出力レベルを-8 dBm および-16 dBm に落としてみます。0 dBm 時と同様に、ピーク電流は 10mA 程度となりました。



## 5 FAQ

Bluetooth Xpress について、よく頂く質問をまとめました。

Silicon Labs 社の WEB ドキュメントにも FAQ が用意されています。

<https://docs.silabs.com/bgx/latest/faq>

### 5-1 仕様・使用方法

- Bluetooth 5.0 の新機能 2M PHY に対応していますか？
  - 対応しています。Variable (**bl p p**) で設定変更できます。
- 使用する PHY を 2M PHY (或いは 1M PHY) に固定できますか？
  - PHY として 2M を使用するか 1M を使用するかは、セントラル側の判断になります。スマホ (セントラル) と接続する場合、どの PHY を使用したいかを **bl p p** でリクエストすることはできますが、最終的にセントラル側の決定に従います。BGX 同士の接続の場合には、セントラル側 BGX の **bl p p** 設定に従います。
- Bluetooth 5.0 の新機能 LE Long Range (Coded PHY) に対応していますか？
  - 対応していません。
- LE Data Packet Length Extension に対応していますか？
  - 対応しています。パケットの最大長は 251 byte です。使用／未使用については、ユーザは介入できず、BGX 内蔵ソフトウェアが自動的に判断します。設定・制御コマンドも用意されておりません。
- 何台まで同時に接続できますか？
  - 同時接続は 1 台までです。(セントラル時、ペリフェラル時いずれにおいても)
- ボンディング情報は何件まで保持できますか？
  - 14 件のボンディング情報を保持できます。15 件目以降のボンディングについては、長い間使われていないボンディング情報を上書きします。
- 起動すると自動でアドバタイズ送信を開始しますが、設定値は High 設定と Low 設定のどちらを使用していますか？
  - High 設定 (High duty) を使用してアドバタイズを行います。
- 起動時のアドバタイズ送信開始を無効にすることはできますか？ (**adv off** の状態で起動することはできますか？)
  - 変更できません。起動後、**adv off** コマンドでアドバタイズを停止してください。

- セキュリティの Numeric comparison に対応していますか？
  - 対応していません。Keyless および Passkey がご使用になれます。
- ラズパイから BGX に接続することはできますか？
  - ラズパイから BGX に接続することは可能です。その場合、ラズパイはセントラルとして、BGX はペリフェラルとして動作することになります。現時点ではラズパイ向けのライブラリやサンプルコードは提供しておりません。しかしながら、使用する BLE サービスについては WEB ドキュメント([Xpress Streaming Service](#))にて情報提供しております。また、モバイル用のライブラリについても WEB ドキュメント([Mobile Library documentation](#))にて情報提供されておりますので、そちらをご覧頂くことで、ラズパイへの実装のご参考になるかと思えます。
- BOOT ピンの用途は何ですか？
  - BOOT ピンを Low にすると、BGX はシリアルポート経由でのアップデートを行えるようになります。アップデート時に使用するプログラムは、今後提供される予定です。それまでは、BGX Commander を使用した OTA update をご利用ください。



## 5-2 トラブルシューティング

- スマートフォンの特定機種と接続できません
  - LE Secureに対応していないスマートフォンと接続する場合には、variable(**ble p**)を**any**に設定してください。
- 接続に失敗します(タイムアウトが発生します)
  - ボンディング情報が合致していない可能性があります。例えば、一方にはボンディング情報が記憶されており、他方にはボンディング情報が残っていないような場合です。双方のボンディング情報を削除してから、再度接続をお試しください。
- BGX 同士の接続に失敗します
  - セキュリティに関する variable(**ble b**, **ble k**, **ble p**)が合致していない可能性があります。双方の設定が一致しているかお確かめください。
- ビーコンが検出(scan)できません
  - BGX をセントラルとして使用する場合には、BGX ペリフェラルのみ検出することができます。それ以外のペリフェラル(ビーコンなど)の検出が必要な場合には、BLE モジュール BGM ファミリをご使用ください。
- BGX Commander(iOS 版)をビルドすると、MMDrawer に関するリンクエラーが出ます
  - ビルドした際に  
**ld: library not found for -IMMDrawerController**  
というリンクエラーが生じます。BGX Commander では依存管理に CocoaPods を使用しており、それをインストール頂くことでビルドできるようになります。詳しくは WEB ドキュメントの [FAQ](#) をご覧ください。

## 改版履歴

| Version | 改定日         | 改定内容                 |
|---------|-------------|----------------------|
| 1.0     | 2018 年 12 月 | ・新規作成。マクニカオンラインで公開   |
| 1.1     | 2018 年 12 月 | ・コマンド制御、消費電流、FAQ を追加 |
|         |             |                      |
|         |             |                      |
|         |             |                      |
|         |             |                      |
|         |             |                      |
|         |             |                      |
|         |             |                      |
|         |             |                      |
|         |             |                      |

## 参考文献

- Silicon Labs 社 各種ドキュメント
- Silicon Labs 社 ナレッジベース、コミュニティフォーラム

## 免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不審な点や誤り、記載漏れなどお気づきの点がありましたら、弊社までご一報いただければ幸いです。
4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的なものとしてかかれたものです。製品をご使用になる場合は、メーカーリリースの資料もあわせてご利用ください。

### 本社

〒222-8561 横浜市港北区新横浜 1-6-3 TEL 045-470-9841 FAX 045-470-9844