

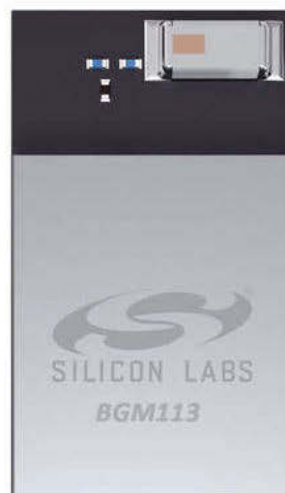
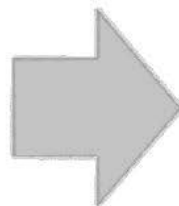
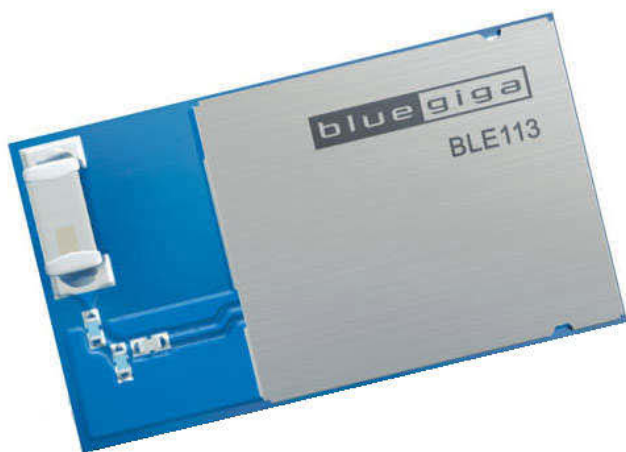
AN1036: BLE113 から BGM113 への移行ガイド



このドキュメントでは、BLE113 Bluetooth® モジュールから Blue Gecko BGM113 Bluetooth® モジュールに移行する方法について説明します。できるだけ簡単に移行を行うために、高レベルの機能、モジュール構成、ハードウェア設計要件、およびソフトウェア API に関する主な相違点を解説します。

要点

- ・ SDK とツール
- ・ BLE113 と BGM113 の相違点
 - ・ 電源
 - ・ PCB レイアウト
 - ・ 構成
 - ・ API
- ・ 移行の簡略化



第 1 章 はじめに

このドキュメントでは、BLE113 Bluetooth モジュールから Blue Gecko BGM113 Bluetooth モジュールに移行する方法について説明します。セクション [第 2 章 SDK とツール](#) では、BGM113 での作業時に使用する SDK およびツールについて説明します。セクション [第 3 章 機能の相違点](#) では、BLE113 と BGM113 の機能の相違点について簡単に説明します。ハードウェア関連の相違点については、セクション [第 4 章 ハードウェア関連の相違点](#) で説明します。

BGM113 は、BLE113 とのフットプリント互換性がありますが、GPIO 配置で BGM113 を BLE113 の直接ドロップイン代替として使用できないなど、いくつかの相違点があります。

セクション [第 5 章 ソフトウェア関連の相違点](#) では、BLE113 と BGM113 のプロジェクト構成オプションの相違点について説明します。セクション [第 6 章 BLE113 から BGM113 へのポーティング・アプリケーション・コード](#) では、アプリケーション・プログラミング・インターフェイス (BGAPI) の観点から BLE113 と BGM113 を比較します。

第 2 章 SDK とツール

2016 年 9 月時点で、最新の SDK for BLE113 のバージョンは 1.4.2.-130 です。

Blue Gecko ベースの製品 (BGM113 を含む) での最新の SDK のバージョンは 2.0.0 (ビルド 1391) です。この SDK は Simplicity Studio v4 に統合されています。最新バージョンの Simplicity Studio v4 をインストールすると、Studio で *Update Software* 機能を実行することによって、Bluetooth Smart SDK を追加できます。

Blue Gecko モジュールの開発ツールの詳細は、ドキュメント『QSG108: Blue Gecko Bluetooth Smart ソフトウェア・クイック・スタート・ガイド』に記載されています。

BLE113 SDK には、BLE GUI という名前のテスト・プログラミングが付属しています。これは BLExxx モジュールのさまざまな機能のテストに使用できます。BGM113 の開発の場合、対応するツールは BGTool です。

Note: BGM113 では BLEGUI を使用できません。

第3章 機能の相違点

このセクションでは、BLE113 と BGM113 の主な機能の相違点について説明します。

表 3.1. Feature Differences between BLE113 and BGM113

Feature	BLE113	BGM113
Bluetooth features	Bluetooth 4.0	Bluetooth 4.2 ¹
Max TX power	0 dBm	+3 dBm
Sensitivity	-93 dBm	-93 dBm
TX current (at 0 dBm)	18.2 mA (14.3 mA with DC-DC)	8.8 mA
RX current	14.3 mA	8.7 mA
Sleep current	0.4 μ A (Sleep mode 3)	1.4 μ A (EM2), 1.1 μ A (EM3 stop current)
MCU	Single-cycle 8051-compatible core	ARM [®] Cortex [®] -M4, 38.4 MHz
RAM	8 kB	32 kB
FLASH	128 kB / 256 kB	256 kB
Number of GPIO	19	14
DC-DC	No DC-DC	Integrated DC-DC converter
Supply voltage range	2V to 3.6V	1.85 V to 3.8 V
Operating temperature range	-40 ° C to 85 ° C	-40 ° C to 85 ° C
Dimensions	9.15 × 15.75 × 2.1 mm	9.15 × 15.73 × 1.9 mm
Note:		
1. Software upgradable to Bluetooth 4.2.		

第 4 章 ハードウェア関連の相違点

このセクションでは、BLE113 と BGM113 のハードウェア関連の相違点について説明します。

4.1 ピン配列の互換性

以下の図は BGM113 モジュールのピン配列を示しています。BGM113 は BLE113 とのフットプリント互換性があります。

グラウンドおよび電源ピンと、リセット、プログラミング、および GPIO ピンは、同じピンにマッピングされます。

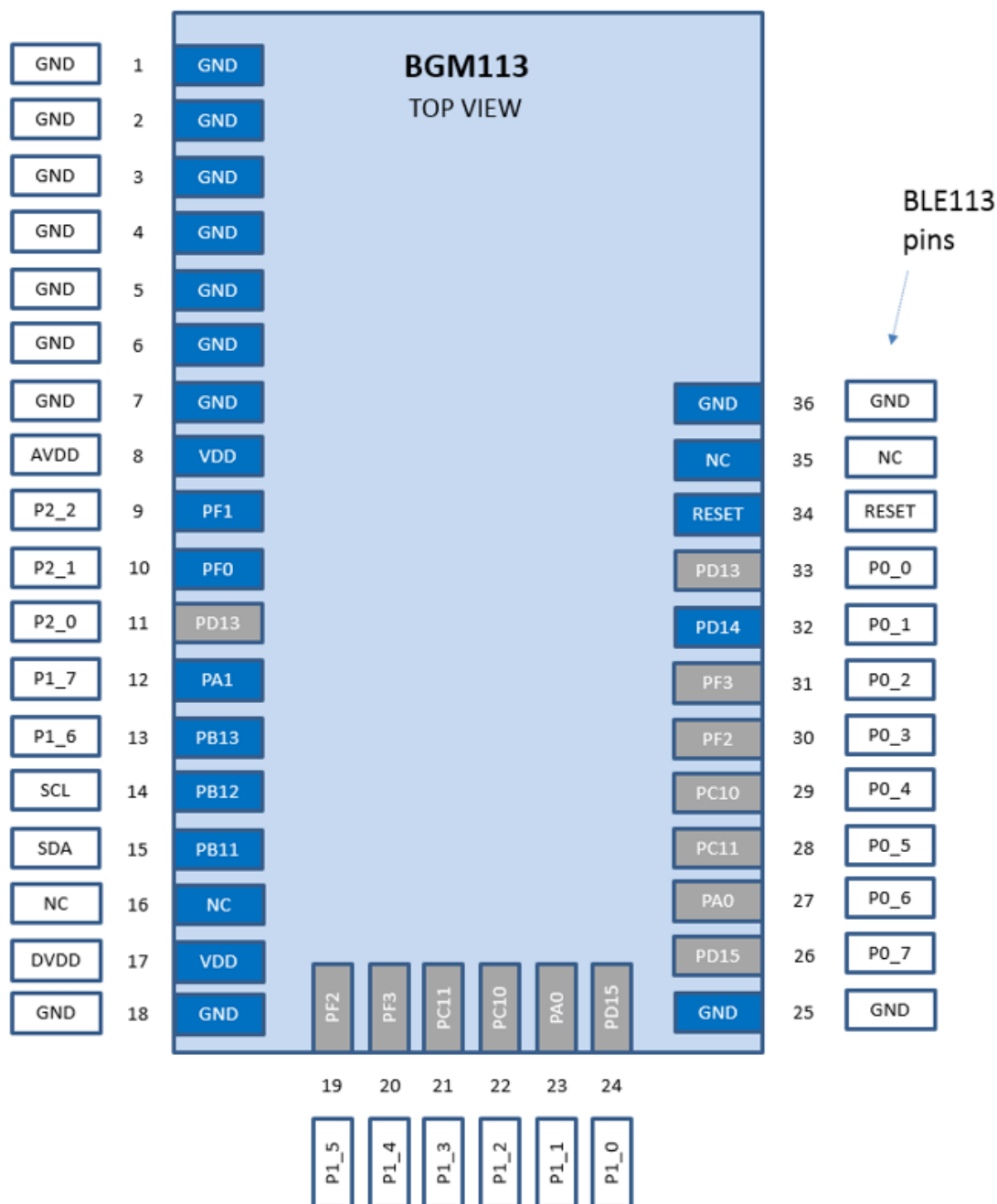


図 4.1. BLE113 と BGM113 のピン配列

Note: BGM113 の一部の GPIO 信号は、複数のピンに接続されます。つまり、個々の GPIO ラインの最大数は、BLE113 より BGM113 の方が少なくなります。上図で網掛けされているのがこれらのピンです。

4.1.1 デバッグおよびプログラミング・ピン

デバッグ・ポートは、BGM113 と BLE113 の両方とも同じピンにマッピングされます。そのため、両方のモジュールで同じデバッグ・コネクタを使用できます。ただし、プログラミング・インターフェイスはまったく異なることに注意してください。BGM113 は ARM SWD インターフェイスを使用しますが、BLE113 は TI CC デバッガ・インターフェイスを使用します。

BLE113 と BGM113 のデバッガ・ピンを以下の表に示します。

表 4.1. BLE113 and BGM113 Debugger Pins

Pin Number	BLE113 Pin Name / Function	BGM113 Pin Name / Function
9	P2_2 / PROG_DC	PF1 / SWDIO
10	P2_1 / PROG_DD	PF0 / SWCLK

4.1.2 電源とリセット・ピン

電源はピン 8 と 17 に接続されます。BLE113 は、アナログおよびデジタル電源 (AVDD、DVDD) 用の個別のピンを備えていますが、BGM113 は、アナログおよびデジタル領域で共通の単一電源電圧 VDD のみを備えています。

BGM113 と BLE113 の両方で、リセット・ピンはアクティブ・ローであり、内部プルアップ抵抗を備えています。

4.1.3 GPIO ピン

GPIO 信号は、BGM113 と BLE113 の同じピンにマッピングされます。ただし、[図 4.1 BLE113 と BGM113 のピン配列 \(5 ページ\)](#) に示すように、一部の GPIO 信号は BGM113 の 2 つのモジュール・ピンにマッピングされます。そのため、BLE113 を BGM113 に交換する際にコンフリクトが発生する可能性があります。

BLE113 と BGM113 の GPIO ピンの一覧はセクション [第 7 章 付録 - BLE113 と BGM113 のピンの比較](#) に記載されています。BLE113 では、ピン 14 と 15 は I²C 用に固定配線されていますが、BGM113 では、これらを I²C に使用するか、または GPIO 信号として使用することができます。

4.1.4 未使用のピン

未使用のピンをフローティング状態のままにしないでください。BLE113 では、定義された内部プルアップまたはプルダウン抵抗は、当該のポート全体に影響します。たとえば、ポート P2 のすべてのピンは high または low に設定できます。BGM113 では、プル抵抗をピンごとに個別にプログラムできます。

4.1.5 電流駆動能力の高いピン

BLE113 には、20 mA の駆動能力を持つ 2 つの I/O ピン、P1_0 と P1_1 があります。BGM113 には、駆動能力の高い専用ピンはありません。

4.1.6 特化した機能を持つピン

BLE113 では、I²C SCL と SDA 信号はピン 14 と 15 に固定されています。BGM113 では、I²C 信号を利用可能な GPIO に自由にマッピングできます。これらは、互換性のために BLE113 と同じピンにマッピングすることもできます。I²C が不要な場合は、これらのピンを汎用 I/O としてアプリケーションで使用できます。

4.2 パリフェラル

このセクションでは、BLE113 と BGM113 の周辺機器の相違点について説明します。

4.2.1 UART/SPI

BLE113 と BGM113 の両方に 2 つの USART ブロックがあり、UART または SPI モードに構成できます。BLE113 には、USART ピン用の 2 つの代替ピン・マッピングがあります。信号はグループとしてマッピングされます。これは、たとえば、すべての UART0 信号が alt1 または alt2 マッピングのいずれかを使用することを意味します。

BGM113 では、USART 信号を使用可能な GPIO に個別に配線できるため、より柔軟なピン割り当てが可能になります。

4.2.2 ADC

BLE113 には、ADC 入力として使用可能な 8 つのピンがあります。これらのピンは P0_0 ~ P0_7 です。BGM113 では、使用可能な GPIO 入力を ADC への入力として使用できます。

ADC リファレンス

BLE113 の ADC リファレンスは以下のとおりです。

- ・ 内部リファレンス (1.24 V)
- ・ AIN7 ピンの外部リファレンス
- ・ AVDD ピン
- ・ AIN6 の外部リファレンス - AIN7 差動入力

BGM113 の ADC リファレンスは以下のとおりです。

- ・ 内部 1.25 V リファレンス
- ・ 内部 2.5 V リファレンス
- ・ バッファ付き VDD
- ・ 内部差動 5 V リファレンス
- ・ ピン 6 からのシングルエンド外部リファレンス
- ・ 差動外部リファレンス、2x (ピン 6 ~ ピン 7)
- ・ バッファなし 2xVDD

詳細については、『*EFR32xG1 Wireless Gecko Reference Manual (EFR32xG1 ワイヤレス Gecko リファレンス・マニュアル)*』を参照してください。

電源電圧の測定

BLE113 には、ADC による電源電圧レベルの測定に使用できる特別な ADC チャンネル VDD/3 があります。

BGM113 では、ダウンスケールされた VDD を ADC 入力に接続することはありません。ただし、内部 5 V 差動で ADC リファレンスを設定し、AVDD チャンネルをサンプリングすることで、VDD 測定を行うことができます。詳細については、『*Bluetooth® Smart ソフトウェア API リファレンス・マニュアル*』のコマンド `cmd_hardware_read_adc_channel` を参照してください。

4.2.3 タイマ

BLE113 には、ユーザ・アプリケーションから使用できる 3 つのタイマ、timer1、timer3、timer4 があります。これらのタイマには、タイマ/カウンタ/PWM 機能があります。タイマ 1 は 16 ビットですが、タイマ 3 と 4 は 8 ビットです。

BLE113 データシートには、タイマ・チャンネルから出力ピンへのマッピングの一覧が記載されています。各タイマには、alt1 と alt2 の 2 つの代替マッピングがあります。モジュール・ピンにマッピングできるタイマ・チャンネルの総数は $5+2+2 = 9$ です。

BGM113 には、入力キャプチャおよびコンペア/パルス幅変調 (PWM) 用の 7 (3+4) 個のコンペア/キャプチャ・チャンネルを備えた 2 つの 16 ビット汎用タイマが用意されています。

BGM113 では、タイマ・チャンネルを使用可能な GPIO ピンにマッピングできます。これらをグループ内でマッピングする必要はありません (BLE113 での alt1/alt2 マッピングなど)。

一般に、BGM113 でのタイマ構成は少なくとも BLE113 と同等の柔軟性があります。タイマ・チャンネルは自由に使用可能なピンにマッピングできるため、GPIO ピンの割り当てにコンフリクトがなければ、BLE113 から BGM113 への設計の移植は問題なく行うことができます (前述のように、BGM113 では、一部の信号が複数のピンに配線されるため)。

一般的な 16 ビット・タイマのほかに、BGM113 には、CRYOTIMER と LETIMER の 2 つの特殊タイマ周辺機器が用意されています。

CRYOTIMER は、低周波発振器で動作する 32 ビットのカウンタです。これはすべてのエネルギー・モードで動作可能であり、最も深いスリープ・モードである EM4 からでも、モジュールをスリープ・モードから復帰させます。

LETIMER は、32.768 Hz のクロックで動作する 16 ビットのダウンカウンタです。ここでは、時間を記録し、構成可能な波形を出力することができます。LETIMER は、スリープ・モード EM2 まで使用できます。

Note: 書き込み時、BGM113 (基本タイマと LETIMER、CRYOTIMER) のタイマ機能は C ベースのアプリケーションでのみ使用でき、EMLIB ドライバを使用して MCU 周辺機器に直接アクセスできます。タイマ機能 (PWM 生成など) は、BGAPI または BGScript からは使用できません。

第 5 章 ソフトウェア関連の相違点

このセクションでは、BLE113 と BGM113 のソフトウェア関連の相違点について説明します。

5.1 プロジェクト構成ファイルの変換

このセクションでは、BGScript™ プロジェクト構成ファイルを BLE113 から BGM113 形式に変換する方法について説明します。

プロジェクト構造のセットは BLE113 と BGM113 で共通です。

- ・ *Project_name.bgproj* (デバイス・タイプと含まれる XML ファイルのリストを定義)
 - ・ *hardware.xml* (hw 構成)
 - ・ *gatt.xml* (GATT データベース定義)
 - ・ *script.bgs* (BGScript スクリプト)

以下の表は、2 つのサンプル・プロジェクト・ファイル (**bgproj*) を示しています。左側が BLE113、右側が BGM113 です。

表 5.1. BLE113 and BGM113 Debugger Pins

BLE113 Project File	BGM113 Project File
<pre><?xml version="1.0" encoding="UTF-8" ?> <project> <gatt in="gatt.xml" /> <hardware in="hardware.xml" /> <script in="keyboard.bgs" /> <image out="out-ble113.hex" /> <device type="ble113" /> <boot fw="bootuart" /> </project></pre>	<pre><?xml version="1.0" encoding="UTF-8" ?> <project device="bgm113"> <!-- GATT service database --> <gatt in="gatt.xml" /> <!-- Local hardware configuration file --> <hardware in="hardware.xml" /> <!-- BGScript source code --> <script in="keyboard.bgs" /> <!-- Firmware output file --> <image out="bgm113demo.bin" /> </project></pre>

プロジェクト・ファイルを BLE113 から BGM113 に移行する際に考慮すべき相違点は以下のとおりです。

- ・ BGM113 では、ターゲットのデバイス・タイプを <project> タグで定義します。
- ・ BGScript ソース・ファイルは、BGM113 プロジェクトの <scripting> タグの中に入れます。

Note: BGM113 では <boot> オプションを使用できません。

5.2 ハードウェア構成ファイル

以下のセクションでは、BLE113 と BGM113 のハードウェア構成関連の相違点について説明します。

5.2.1 GPIO 構成

BLE113 プロジェクトでは、<port> タグを使用して、GPIO 入力にのみ有効な GPIO 設定を構成します。以下は BLE113 プロジェクトからの例です。

例: <port> タグを使用して GPIO 設定を構成する

```
<port index="0" tristatemask="0" pull="down" />
```

上記の行は、バンク 0 のすべてのピンのプル方向を「down」に構成します。プル方向はバンク全体に対して定義されているため、同じバンク内でプルアップとプルダウンの両方を使用することはありません。バンク内の選択したピンのプル抵抗を無効にする場合は、トライステートマスクを使用します。

BGM113 の GPIO 構成はまったく異なります。以下の例は、2 つの出力ピン (PA6、PA3) と 1 つの入力ピン (PF6) の構成例を示しています。

例: 2 つの出力ピン (PA6、PA3) と 1 つの入力ピン (PF6) を構成する

```
<gpio port="A" pin="6" mode="pushpull" out="1" />
<gpio port="A" pin="3" mode="pushpull" out="0" />
<gpio port="F" pin="6" mode="input" out="0" interrupt="both" />
```

BGM113 プロジェクトでは、*hardware.xml* ファイルの入力ピンと出力ピンの両方の設定を構成できます。プルアップ抵抗はピンごとに個別に定義され、GPIO 出力については初期状態を定義することもできます。

入力および出力の GPIO の基本設定は、BGScript または BGAPI コマンドを使用して実行時に構成することもできます。この関数は `hardware_configure_gpio` です。詳細については、『*Bluetooth® Smart ソフトウェア API リファレンス・マニュアル*』を参照してください。

使用可能な GPIO ピン構成のオプションの詳細は、『*UG119: Blue Gecko Bluetooth® Smart デバイス構成ガイド*』に記載されています。

5.2.2 UART 構成

BLE113 と BGM113 での、BGAPI コマンドまたはアプリケーション固有の入出力の UART の基本的な使用法はほぼ同じです。ただし、UART 構成では、BLE113 プロジェクトから BGM113 に移行する際に考慮すべき相違点がいくつかあります。

以下は UART 構成の例です。最初の例は BLE113、2 番目の例は BGM113 です。

例: BLE113 での UART 構成

```
<usart channel="1" alternate="1" baud="115200" endpoint="none" />
```

例: BGM113 での UART 構成

```
<uart index="1" baud="115200" flowcontrol="false" bgapi="false" />
```

BLE113 と BGM113 の UART 構成の相違点は以下のとおりです。

- ・ タグ名は、BLE113 の場合は `<usart>`、BGM113 の場合は `<uart>` です。
- ・ UART インスタンスを選択するパラメータは、BLE113 の場合は「channel」、BGM113 の場合は「index」です。
- ・ BLE113 の「Flow」パラメータは、BGM113 では「flowcontrol」になります。

UART 構成オプションの詳細なリストについては、`CONFIG_GUIDE` を参照してください。

上記の BLE113 UART 例には、2 つの代替ピン・マッピングからの選択に使用するパラメータ「alternate」があることに注意してください。BGM113 には、このようなパラメータがありません。これは、UART ピンを使用可能な GPIO に個別にマッピングできるためです。

以下は、RTS および CTS ピンの位置を明示的に指定する BGM113 の UART 構成の例です。

例: RTS および CTS ピンが明示的に定義されている BGM113 での UART 構成

```
<uart index="1" baud="115200" flowcontrol="true" rts_pin="PA3" cts_pin="PA2" bgapi="true"/>
```

5.2.3 ウェイクアップ・ピン

ウェイクアップ・ピンの機能は、BLE113 と BGM113 でほぼ同じです。ウェイクアップ・ピンを定義する構文が多少違います。

例: BLE113 でのウェイクアップ・ピンの定義 (ピン P0_0)

```
<wake_up_pin enable="true" port="0" pin="0" state="up" />
```

例: BGM113 でのウェイクアップ・ピンの定義 (ピン PD13)

```
<wake_up port="d" pin="13" state="up" />
```

機能的に上記の例は両方とも同じことを行います。ウェイクアップ・ピンを P0_0/PD13 に構成して、極性をアクティブ・ハイとして設定します。構文のみが少し異なります。詳細については、『*UG119: Blue Gecko Bluetooth® Smart デバイス構成ガイド*』に記載されています。

5.2.4 ホスト・ウェイクアップ・ピン

ホスト・ウェイクアップ・ピンの機能は、BLE113 と BGM113 でほぼ同じです。このピンを構成するための構文は少し異なります。以下の例を参照してください。詳細については、『UG119: Blue Gecko Bluetooth® Smart デバイス構成ガイド』に記載されています。

例: BLE113 で P1_7 をホスト・ウェイクアップ・ピンとして構成する

```
<host_wakeup_pin enable="true" port="1" pin="7" state="up" />
```

例: BGM113 で PA1 をホスト・ウェイクアップ・ピンとして構成する

```
<host_wake_up port="a" pin="1" state="up" />
```

5.2.5 スリープの有効化

スリープの有効化の構成は BLE113 と BGM113 でほぼ同じです。BGM113 は属性 `max_mode` をまったく使用しない点が異なります。以下の例は、BLE113 と BGM113 でスリープを有効化する方法を示しています。

例: BLE113 でスリープを有効化する (max mode は 2、powermode 3 を有効化しない)

```
<sleep enable="true" max_mode="2" />
```

例: BGM113 でスリープを有効化する (max mode は暗黙的に EM2)

```
<sleep enable="true" />
```

5.2.6 TX 電力の設定

BLE113 プロジェクトでは、`<txpower>` タグを使用してハードウェア構成ファイルで TX 電力を設定できます。代わりに、関数 `hardware_set_txpower` を呼び出すことで、実行時に BLE113 TX 電力を動的に設定できます。

例: BLE113 で実行時に最大電力レベル (0dBm) を設定する

```
call hardware_set_txpower(14)
```

BGM113 には、構成ファイルで TX 電力を静的に構成するオプションがありません。そのため、関数 `system_set_tx_power` を使用して動的に TX 電力レベルを設定する必要があります。詳細については、『Bluetooth® Smart ソフトウェア API リファレンス・マニュアル』を参照してください。

例: BGM113 で実行時に最大電力レベル (+3 dBm) を設定する

```
call system_set_tx_power(30)
```

TX 電力を調整する際の単位の違いに注意してください。BLE113 は値 0 ~ 14 を使用します。ここで 0 は最低値 (約 -24 dBm)、14 は 0 dBm にほぼ等しい最高設定値です。

BGM113 では、パラメータを 0.1 dBm 単位で指定します。たとえば、設定 15 は 1.5 dBm に対応します。関数 `system_set_tx_power` は、スタックにより使用される実際の電力設定の値を返します。電力は 0.1 dBm ステップでは調整できません。スタックは、ユーザーにより要求された値に最も近い、事前定義された電力レベルを使用します。

5.2.7 廃止予定のハードウェア関連設定

次のハードウェア関連設定は BGM113 には適用されません。

- `<sleeposc>`
- `<slow_clock>`
- `<lock_debug>`
- `<pmux>`

第 6 章 BLE113 から BGM113 へのポーティング・アプリケーション・コード

この章では、BLE113 と BGM113 の BGAPI コマンドの相違点について説明します。BGAPI コマンドおよびイベントの中には、BLE113 と BGM113 でほとんど同じものもあります。一方、一部のパート（GATT サービスの検出など）は、BGM113 と BLE113 ではかなり異なります。

この章では以下について説明します。

- ・ System_boot イベント
- ・ アドバタイズの有効化とアドバタイズ・データ/パラメータの変更
- ・ 接続のオープン/クローズ
- ・ 通知および表示の送信
- ・ ソフト・タイマ
- ・ 検出の有効化
- ・ 周辺機器への接続
- ・ GATT サービスの検出
- ・ 通知のサブスクライブ

6.1 システムの起動

BLE113 と BGM113 の両方で、「システムの起動」はモジュールの再起動後に生じる最初のイベントです。BGScript プロトタイプは以下に示されています。

```
# BLE113:
event system_boot(major, minor, patch, build, ll_version, protocol_version, hw)
# BGM113:
event system_boot(major, minor, patch, build, bootloader, hw)
```

両方のケースで、スタック・バージョン（1.0.2-755 など）を識別する最初の 4 つのパラメータは類似しています。ハードウェアのバージョンなどに関連する最後のパラメータはわずかに異なります。

6.2 アドバタイズの有効化

アドバタイズの開始と終了は BLE113 と BGM113 で基本的に同一で、コマンド `gap_set_mode` を使用して実行されます：

```
# BLE113:
call gap_set_mode(discover, connect)(result)
# BGM113:
call le_gap_set_mode(discover, connect)(result)
```

ここでの唯一の違いは、BGM113 コマンドには名前に接頭辞「le_」が付くことです（「le」は「低エネルギー」を表します）。これと同じ命名規則は、ほとんどの BGM113 コマンドとイベントで使用されます。

アドバタイズ・パラメータとアドバタイズ・データの設定は BLE113 と BGM113 でほとんど同じです。関数とパラメータは同じです（BGM113 の接頭辞「le_」を除き）：

```
# BLE113:
call gap_set_adv_parameters(adv_interval_min, adv_interval_max, adv_channels)(result)
call gap_set_adv_data(set_scanrsp, adv_data_len, adv_data_data)(result)
# BGM113:
call le_gap_set_adv_parameters(interval_min, interval_max, channel_map)(result)
call le_gap_set_adv_data(scan_rsp, adv_data_len, adv_data_data)(result)
```

6.3 接続ステータスの変更

接続ステータスの変更を示す BGAPI イベントは、BLE113 と BGM113 でまったく異なります。このため、入力接続を受け入れるアプリケーションで、何らかの移植作業が基本的に必要になる可能性があります。

BLE113 では、接続ステータスの変更は、次のプロトタイプを持つ 1 つのイベント「connection_status」で示されます：

```
# BLE113:
event connection_status(connection, flags, address, address_type, conn_interval, timeout, latency, bonding)
```

アプリケーションは、flags パラメータからこのイベントが生じた原因をデコードできます。

BGM113 では、どのようなものでも connection_status イベントになります。その代わりに、接続ステータスの変更は、より具体的な使用を対象にする、いくつかの異なるイベントで示されます（入力接続の処理など）。BGM113 スタック関連イベントの一部を以下に示します。

```
# BGM113:
event le_connection_opened(address, address_type, master, connection, bonding)
event le_connection_closed(reason, connection)
event le_connection_parameters(connection, interval, latency, timeout, security_mode)
```

アプリケーションを BLE113 から BGM113 に移植する際は、connection_status イベント・ハンドラの機能を、BGM113 API により提供される最適なイベントに移動する必要があります。

6.4 通知または指示の送信

多数のアプリケーションでは通知または指示を使用して、データをクライアントに転送します。BLE113 と BGM113 で通知を送信するために使用される API 関数を以下に示します。

```
# BLE113:
call attributes_write(handle, offset, value_len, value_data)(result)

# BGM113:
call gatt_server_send_characteristic_notification(connection, characteristic, value_len, value_data)(result)
# Optional:
call gatt_server_write_attribute_value(attribute, offset, value_len, value_data)(result)
```

BLE113 と BGM113 では、通知の導入方法に基本的な違いがあります。BLE113 では、アプリケーションは、attributes_write コマンドを使用してローカル GATT データベースに書き込むだけです。この属性に関する通知を定期購読している接続済みのクライアントがある場合、スタックは通知を自動的に生成します。

BGM113 の場合、通知の自動トリガー機能はありません。このアプリケーションは、コマンド gatt_server_write_attribute_value を使用してローカル・データベースの値を更新できます。通知をトリガーするには、アプリケーションはコマンド gatt_server_send_characteristic_notification を明示的に使用する必要があります。

アプリケーションが通知の送信のみを行う必要があり、クライアントがデータベースの値を読み取る必要がなければ、大抵の場合は attribute_write の呼び出しを gatt_server_send_characteristic_notification で置換するだけで十分です。

上記の説明は BGM113 で通知がどのように生成されるかを説明するもので、これは指示についても同様です。

6.5 ソフト・タイマ

ソフト・タイマを構成するための API 呼び出しは、BLE113 と BGM113 の両方で同じです。

```
call hardware_set_soft_timer(time, handle, single_shot)(result)
```

両方のモジュールで、タイマ間隔は、1 秒が 32768 チックに等しくなるようにハードウェア・クロック・チックで定義されます。

BLE113 では、一度に 1 回のみ繰り返しタイマが存在できます。新しい繰り返しタイマを開始すると、既存の繰り返しタイマは削除されます（そのようなタイマが存在する場合）。BGM113 の場合、繰り返しソフト・タイマの数に制限はありません。

6.6 検出の有効化

BLE113 と BGM113 の両方について、スキャン・パラメータの設定、検出の開始、および検出の終了のための API 呼び出しを以下に示します。BGM113 で使用される接頭辞「le_」を除き、プロトタイプは同一です。

```
# BLE113:
call gap_set_scan_parameters(scan_interval, scan_window, active)(result)
call gap_discover(mode)(result)
call gap_end_procedure()(result)

# BGM113:
call le_gap_set_scan_parameters(scan_interval, scan_window, active)(result)
call le_gap_discover(mode)(result)
call le_gap_end_procedure()(result)
```

デフォルトのスキャン・パラメータは、BLE113 と BGM113 で異なります。アプリケーションが `gap_set_scan_parameters` を明示的に呼び出さないで、スタックのデフォルトを使用する場合は、このことを考慮する必要があります。BLE113 のスキャン間隔とスキャン・ウィンドウのデフォルトは値 (75, 50) ms で、BGM113 のデフォルト値は (10, 10) ms です。

各スキャン応答で生じるイベントは両方のモジュールで類似しています。

```
# BLE113: イベント gap_scan_response(rssi, packet_type, sender, address_type, bond, data_len, data_data) # BGM113: イベント le_gap_scan_response(rssi, packet_type, address, address_type, bonding, data_len, data_data)
```

パケット・タイプのエミュレーションは BLE113 と BGM113 でわずかに異なることに注意してください。

BLE113 :

- ・ 0 : 接続可能なアドバタイズ・パケット
- ・ 2 : 接続可能でないアドバタイズ・パケット
- ・ 4 : スキャン応答パケット
- ・ 6 : 検出可能なアドバタイズ・パケット

BGM113 :

- ・ 0 : 接続可能な不特定アドバタイズ
- ・ 2 : スキャン可能な不特定アドバタイズ
- ・ 3 : 接続可能でない不特定アドバタイズ
- ・ 4 : スキャン応答

6.7 接続を開く

アドバタイズ・デバイスへの接続を開くために使用する API 呼び出しは BLE113 と BGM113 でまったく異なります。関数プロトタイプを以下に示します：

```
# BLE113:
call gap_connect_direct(address, addr_type, conn_interval_min, conn_interval_max, timeout, latency)(result, connection_handle)

# BGM113:
call le_gap_open(address, address_type)(result, connection)
```

BLE113 では、接続パラメータは `gap_connect_direct` へのパラメータとして渡され、BGM113 では、関数 `le_gap_open` は送信先アドレスとアドレス・タイプのみをパラメータとして受け取ります。

BGM113 で接続パラメータを変更する場合、次の API 呼び出しを使用できます。

```
# BGM113:
call le_gap_set_conn_parameters(min_interval, max_interval, latency, timeout)(result)
call le_connection_set_parameters(connection, min_interval, max_interval, latency, timeout)(result)
```

これらの関数は、BLE113 で、`gap_connect_direct` コマンドの一部として定義する同じパラメータを設定するために使用されます。上にリストしたこの 2 つの API 呼び出しの違いは、`le_gap_set_conn_parameters` は、後続のすべての接続のために使用されるデフォルトのパラメータを設定するのに対して、`le_connection_set_parameters` は、既に開いている 1 つの接続のパラメータを修正するために使用されるということです。

接続を開いた後に生成されるイベントを以下に示します。

```
# BLE113:
event connection_status(connection, flags, address, address_type, conn_interval, timeout, latency, bonding)

# BGM113:
event le_connection_opened(address, address_type, master, connection, bonding)
event le_connection_parameters(connection, interval, latency, timeout, security_mode)
```

6.8 接続のクローズ

アプリケーションにより、接続が予想外に（監視のタイムアウトなどで）または意図的にクローズされる場合があります。

予想外の接続クローズの場合、BLE113 と BGM113 で次のイベントが発生します。

```
# BLE113:
event connection_disconnected(connection, reason)

#BGM113:
event le_connection_closed(reason, connection)
```

上でリストした両方のイベントには、構文のみが異なる、同じ情報が含まれます。

アプリケーションはアクティブな接続をクローズする必要がある、その後、BLE113 と BGM113 で次の API 呼び出しが使用されます。

```
# BLE113:
call connection_disconnect(connection)(connection, result)

# BGM113:
call endpoint_close(endpoint)(result, endpoint)
```

繰り返しますが、基本的な使用方法は同じですが、構文のみ異なります。

6.9 GATT サービスの検出

サービスの検出方法は、BLE113 と BGM113 で多くの違いがあります。両方のモジュールでのサービス検出に関連する代表的な API 呼び出しとイベントの概要を以下に示します。ここですべての違いを説明することは現実的ではありません。このリストの目的は、BLE113 から BGM113 にアプリケーションを移植する際に役立つように、BGAPI 資料の関連する部分を読者に示すことです。

BLE113 でのサービスの検索：

1. `attclient_read_by_group_type` を呼び出します。
2. これにより、タイプ `attclient_group_found` のイベントが生成されます（一次または二次のサービスそれぞれに 1 つ）。
3. 最後のイベントの後、スタックはイベント `attclient_procedure_completed` を発生させます。

BLE113 の特定のサービスに関する属性の検索：

1. `attclient_find_information` を呼び出します（`handle_range` はパラメータとして渡されます）。
2. これにより、タイプ `attclient_find_information_found` のイベントが生成されます。
3. 最後のイベントの後、スタックはイベント `attclient_procedure_completed` を発生させます。

BGM113 でのサービスの検索：

1. `gatt_discover_primary_services`（または `gatt_discover_primary_services_by_uuid`）を呼び出します。
2. これにより、タイプ `gatt_service` のイベントが生成されます（サービスそれぞれに 1 つ）。
3. 最後のイベントの後、スタックはイベント `gatt_procedure_completed` を発生させます。

BGM113 の特定のサービスに関する属性の検索：

1. `gatt_discover_characteristics` を呼び出します（`service_handle` はパラメータとして渡されます）。
2. これにより、タイプ `gatt_characteristic` のイベントが生成されます。
3. 最後のイベントの後、スタックはイベント `gatt_procedure_completed` を発生させます。

6.10 通知または指示の定期購読

通知（または指示）を定期購読するために、GATT クライアントは値 `0x0001`（または `0x0002`）を一定の特性のクライアント特性構成（CCC）に書き込む必要があります。一般的に、CCC ハンドル値は、特性自体に対して +1 または +2 ステップです。（クライアントはこれが当てはまると思い込まないように注意してください）

BLE113 では、これは `attclient_attribute_write` を呼び出して CCC 値を直接変更することにより実行できます。クライアントは CCC に関連付けられたハンドルが何かを知る必要があります、これはサービス検出の一部として検出されます（上記を参照）。

BGM113 では、クライアントは CCC ハンドルを知る必要はありません。その代わりに、クライアントは `gatt_set_characteristic_notification` を呼び出して、特性の値（CCC ではない）を関数に渡します。スタックは特性クライアント構成を自動的に検出します。

関数 `gatt_set_characteristic_notification` は、通知と指示の両方を有効化/無効化するために使用されます。これは関数に渡される「flags」パラメータで選択します。詳細については、『*Bluetooth® スマート・ソフトウェア API リファレンス・マニュアル*』を参照してください。

第 7 章 付録 – BLE113 と BGM113 のピンの比較

以下の表は、BLE113 と BGM113 のピンの一覧です。

表 7.1. BLE113 and BGM113 Pin Comparison

Pin Number	BLE113	BLE113 Notes	BGM113	BGM113 Notes
9	P2_2		PF1	
10	P2_1		PF0	
11	P2_0		PD13	Same as pin 33
12	P1_7		PA1	
13	P1_6		PB13	
14	SCL	I ² C only	PB12	
15	SDA	I ² C only	PB11	
19	P1_5		PF2	Same as pin 30
20	P1_4		PF3	Same as pin 31
21	P1_3		PC11	Same as pin 28
22	P1_2		PC10	Same as pin 29
23	P1_1	20 mA	PA0	Same as pin 27
24	P1_0	20 mA	PD15	Same as pin 26
26	P0_7		PD15	Same as pin 24
27	P0_6		PA0	Same as pin 23
28	P0_5		PC11	Same as pin 21
29	P0_4		PC10	Same as pin 22
30	P0_3		PF2	Same as pin 19
31	P0_2		PF3	Same as pin 20
32	P0_1		PD14	
33	P0_0		PD13	Same as pin 11

Notes:

1. BLE113 max number of GPIO lines is 19 (+ 2 dedicated pins for I2C).
2. BGM113 max number of GPIO lines is 14.

Silicon Labs

Simplicity Studio™4



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



IoT Portfolio
www.silabs.com/loT



SW/HW
www.silabs.com/simplicity



Quality
www.silabs.com/quality



Support and Community
community.silabs.com

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOModem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

<http://www.silabs.com>