本研究開発の成果の一部は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）委託事業「IoT推進のための横断技術開発プロジェクト」の結果得られたものです。

リーフ、基板、冶具等の配布物、及び次頁以降のドキュメントは、研究・評価の目的以外での使用は、行わないでください。

## AC02A BLE Sugar

# 概要

Silicon Labsの技的認証済みBLEモジュールBGM11S22F256GA-V2を搭載したリーフである。MCUリーフとはUARTで接続される。

# リーフ仕様

## ブロック図



## 電源仕様

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Condition** | **Min.** | **Typ.** | **Max.** |
| Vdd | Power Supply Voltage | － | 2.4V | 3.3V | 3.8V |
| Idd | Operating current | Active | - | 3.8mA | - |
| Sleep | - | 2.8uA | - |

## 主要部品

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **部品番号** | **部品名** | **型番** | **ベンダー名** | **備考** |
| IC200 | BLEモジュール | BGM11S22F256GA-V2 | Silicon Labs | － |

## 外観

|  |  |
| --- | --- |
| **表面** | **裏面** |
| 内蔵アンテナ | C:\Users\k\Desktop\Basic Edu写真20180906\20181009\BLE Sugar裏.JPG |

## ピンアウト

|  |  |
| --- | --- |
| **Name** | **Function** |
| A2 | TXD：UART送信 チップ抵抗の付け替えでD9に変更可 |
| A1 | RXD：UART受信 チップ抵抗の付け替えでD8に変更可 |
| D7 | WAKEUP：ウエイクアップ　H：ウエイクアップ |
| RESET | RST：リセット |
| SWCLK | デバッグI/F クロック |
| SWDIO | デバッグI/F データ入出力 |
| 3V3 | 3.3V入力 |
| GND | GND |

# BLEモジュール(BGM11S22F256GA-V2)仕様

## 概要

|  |  |
| --- | --- |
| **項目** | **内容** |
| SoC | EFR32BG1 (ARM Cortex-M4) |
| Bluetooth version | 4.2 |
| Frequency range | 2400M ~ 2483.5MHz |
| Internet Security | • General Purpose CRC  • Random Number Generator  • Hardware Cryptographic Acceleration for AES 128/256,SHA-1, SHA-2 (SHA-224 and SHA-256) and ECC |
| RX sensitivity | -90 dBm @ 1 Mbit/s GFSK |
| TX power | +8dBm |
| RF certification | CE, full FCC, ISED Canada, Japan and South-Korea |
| Flash | 256KB |
| RAM | 32KB |
| Interfaces | UART |

## 電気的特性

### 最大定格

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Operating Temperature | -40℃ to +85℃ |
| Maximum Operation Voltage | 3.8V |

### 定格

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Symbol** | **Parameter** | **Condition** | **Min.** | **Typ.** | **Max.** |
| Vdd | Power Supply Voltage | － | 2.4V | 3.3V | 3.8V |
| Idd | EM0 Active mode | 38 MHz HFRCO  all peripherals disabled | - | 3.8mA | 3.99mA |
| EM1 Sleep mode | 38 MHz HFRCO  all peripherals disabled | - | 1.33mA | 1.44mA |
| EM2 Deep Sleep mode | Full RAM retention and RTCC running from LFXO | - | 33uA | - |
| EM3 Stop mode | Full RAM retention and CRYOTIMER running from ULFRCO | - | 2.8uA | 6uA |
| EM4H Hibernate mode | 128 byte RAM retention, RTCC running from LFXO | - | 1.1uA | - |
| EM4S Shutoff mode | no RAM retention, no RTCC | - | 0.04uA | 0.20uA |
| Receive mode, active packet reception (MCU in EM1 @38.4 MHz, peripheral clocks disabled) | 1 Mbit/s, 2GFSK, F = 2.4 GHz,Radio clock prescaled by 4 | - | 9.0mA | - |
| Transmit mode (MCU in EM1@ 38.4 MHz, peripheral clocks disabled) | 0 dBm output power, Radio clock prescaled by 3 | - | 8.2mA | - |
| 2 dBm output power | - | 16.5mA | - |
| 8 dBm output power | - | 24.6mA | - |

## データシートリンク先

<https://jp.silabs.com/products/wireless/bluetooth/bluetooth-low-energy-modules/bgm11s-bluetooth-sip-module>

## 主な関数とライブラリ

### BLEの制御

include file：BGLib.h(Leaf Libraies)

|  |  |
| --- | --- |
| **関数** | **概要** |
| BGLib ble112( HardwareSerial \*module, HardwareSerial \*output, uint8\_t pMode ) | BGLibのインスタンスを作成します 【構文】 BGLib ble112(HardwareSerial \*module, HardwareSerial \*output, pMode) 【パラメータ】 ble112: インスタンス名 module: BLEリーフと通信するシリアルポートののインスタンス output:　BLEリーフが出力するシリアルポートののインスタンス　Null固定 pMode: パケットモード　０固定 【戻り値】 なし |
| ble112.ble\_cmd\_le\_gap\_set\_adv\_parameters( interval\_min, interval\_max, channnel\_map ) | アドバタイズのパラメータ設定を行います 【構文】 ble\_cmd\_le\_gap\_set\_adv\_parameters( uint16 interval\_min, uint16 interval\_max, uint8 channnel\_map ) 【パラメータ】 ble112: インスタンス名 interval\_min: Minimum advertising interval. Value in units of 0.625 ms  • Range: 0x20 to 0xFFFF  • Time range: 20 ms to 40.96 s  Default value: 100 ms interval\_max:Maxmum advertising interval. Value in units of 0.625 ms  • Range: 0x20 to 0xFFFF  • Time range: 20 ms to 40.96 s  Default value: 200 ms channel\_map: Advertising channel map which determines which of the three channels will be used for advertising. This value is given as a bitmask.  • 1: Advertise on CH37  • 2: Advertise on CH38  • 3: Advertise on CH37 and CH38  • 4: Advertise on CH39  • 5: Advertise on CH37 and CH39  • 6: Advertise on CH38 and CH39  • 7: Advertise on all channels  Default value: 7 【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_discover( mode ) | Bluetooth discovery mode設定  【構文】 ble\_cmd\_le\_gap\_discover( uint8 mode ) 【パラメータ】 ble112: インスタンス名 mode: discovery mode  enum\_le\_gap\_discover\_mode参照 【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_set\_adv\_data( scan\_rsp, adv\_data\_len, adv\_data ); | アドバタイズデータの設定を行います 【構文】 ble\_cmd\_le\_gap\_set\_adv\_data( uint8 scan\_rsp, uint8 adv\_data\_len, const uint8 \*adv\_data\_data ) 【パラメータ】 ble112: インスタンス名 scan\_rsp: This value selects if the data is intended for advertising packets,scan response packets or advertising packet in OTA. Values: 　• 0: Advertising packets 　• 1: Scan response packets 　• 2: OTA advertising packets 　• 4: OTA scan response packets adv\_data\_len:設定するアドバタイズデータ長 　　　最大31バイト adv\_data\_data:アドバタイズデータ 【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_start\_advertising(handle, discover, connect) | アドバタイズを開始します 【構文】 ble\_cmd\_le\_gap\_start\_advertising( uint8 handle, uint8 discover, uint8 connect ) 【パラメータ】 ble112: インスタンス名 handle: BLE leaf handle discover:Discoverable mode  enum\_le\_gap\_discoverable\_mode参照 connect: Connectable mode 　　enum\_le\_gap\_connectable\_mode参照  【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_stop\_advertising( handle ) | アドバタイズを終了します 【構文】 ble\_cmd\_le\_gap\_stop\_advertising( uint8 handle ) 【パラメータ】 ble112: インスタンス名  【戻り値】  0 |
| ble112.checkActivity( timeout ) | 応答があるまで待ちます 【構文】 checkActivity( uint16\_t timeout ) 【パラメータ】 ble112: インスタンス名 timeout: タイムアウト値　ms  【戻り値】  0 :nobusy  1 :busy |
| ble112.ble\_cmd\_gatt\_set\_characteristic\_notification(connection, characteristic, flags) | GATT Serverにnotificationを設定します 【構文】 ble\_cmd\_gatt\_set\_characteristic\_notification( uint8 connection, uint16 characteristic, uint8 flags ) 【パラメータ】 ble112: インスタンス名 connection: Connection handle characteristic:GATT characteristic handle  flags: Characteristic client configuration flags  • 0: Disable notifications and indications  • 1: Notification  • 2: Indication  【戻り値】  0 |
| ble112.ble\_cmd\_gatt\_server\_send\_characteristic\_notification( connection, characteristic, value\_len, (const uint8 \*)value\_data ) | GATT clientsにnotificationを送信します 【構文】 ble\_cmd\_gatt\_server\_send\_characteristic\_notification( uint8 connection, uint16 characteristic, uint8 value\_len, const uint8 \*value\_data )  【パラメータ】 ble112: インスタンス名 connection: Connection handle  • 0xff: Sends notification or indication to all connected devices.  • Other: Connection handle characteristic:Characteristic handle  enum\_le\_gap\_discoverable\_mode参照 value\_len: value length value: Value to be notified or indicated  【戻り値】  0 |
| ble112.ble\_cmd\_gatt\_write\_characteristic\_value(connection, characteristic, value\_len, \*value\_data); | GATT Serverにnotificationを設定します 【構文】 ble\_cmd\_gatt\_write\_characteristic\_value( uint8 connection, uint16 characteristic, uint8 value\_len, const uint8 \*value\_data ) 【パラメータ】 ble112: インスタンス名 connection: Connection handle characteristic:GATT characteristic handle  value\_len:Characteristic value length  value\_data:Characteristic value  【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_set\_scan\_parameters(scan\_interval, scan\_window, active) | スキャンパラメータを設定します 【構文】 ble\_cmd\_le\_gap\_set\_scan\_parameters( uint16 scan\_interval, uint16 scan\_window, uint8 active ) 【パラメータ】 ble112: インスタンス名 scan\_interval: Scanner interval• Time = Value x 0.625 ms  • Range: 0x0004 to 0x4000  • Time Range: 2.5 ms to 10.24 s   Default value: 10 ms scan\_window: Scan window. The duration of the scan.  • Time = Value x 0.625 ms  • Range: 0x0004 to 0x4000  • Time Range: 2.5 ms to 10.24 s  Default value: 10 ms Note that packet reception is aborted if it has been started before scan window ends. active : Scan type indicated by a flag  • 0: Passive scanning  • 1: Active scanning  Default value: 0 【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_end\_procedure() | current GAP procedureの使用を停止します 【構文】 ble\_cmd\_le\_gap\_end\_procedure( void ) 【パラメータ】 ble112: インスタンス名 【戻り値】  0 |
| ble112.ble\_cmd\_le\_gap\_connect(address, address\_type, initiating\_phy) | デバイスと接続します 【構文】 ble\_cmd\_le\_gap\_connect( bd\_addr address, uint8 address\_type, uint8 initiating\_phy ) 【パラメータ】 ble112: インスタンス名 address: Address of the device to connect to address\_type: Address type of the device to connect to  enum\_le\_gap\_address\_types参照 initiating\_phy:The initiating PHY.  • 1: LE 1M PHY  • 4: LE Coded PHY 【戻り値】  0 |
| ble112.ble\_cmd\_le\_connection\_close(connection) | デバイスを切断します 【構文】 ble\_cmd\_le\_connection\_close(uint8 connection) 【パラメータ】 ble112: インスタンス名 connection: Handle of the connection  【戻り値】  0 |
| ble112.ble\_cmd\_system\_reset(boot\_in\_dfu) | システムリセットを実行します 【構文】 ble\_cmd\_system\_reset(uint8 boot\_in\_dfu) 【パラメータ】 ble112: インスタンス名 boot\_in\_dfu: Boot mode  • 0: Normal reset  • 1: Boot to UART DFU mode  • 2: Boot to OTA DFU mode  【戻り値】  0 |
| ble112.ble\_cmd\_system\_halt(halt) | SLEEPモードへ移行します 【構文】 ble\_cmd\_system\_halt(uint8 halt)  【パラメータ】 ble112: インスタンス名 halt: halt mode  • 1: halt  • 0: resume  【戻り値】  0 |
| ble112.getLastEvent() | 最後に受信したイベントを返します 【構文】 getLastEvent() 【パラメータ】 ble112: インスタンス名 【戻り値】  lastEvent[0] : Message class: System  lastEvent[1] :Message ID |

enum\_le\_gap\_connectable\_mode

|  |  |  |
| --- | --- | --- |
| **Value** | **Name** | **Description** |
| 0 | le\_gap\_non\_connectable | Non-connectable non-scannable. |
| 1 | le\_gap\_directed\_connectable | Directed connectable (RESERVED, DO NOT USE) |
| 2 | le\_gap\_undirected\_connectable | Undirected connectable scannable. Deprecated, replaced by enum le\_gap\_connectable\_ scannable. This mode can only be used in legacy advertising PDUs. |
| 2 | le\_gap\_connectable\_scannable | Undirected connectable scannable. This mode can only be used in legacy advertising PDUs. |
| 3 | le\_gap\_scannable\_non\_connectable | Undirected scannable (Non-connectable but responds to scan requests) |
| 4 | le\_gap\_connectable\_non\_scannable | Undirected connectable non-scannable. This mode can only be used in extended advertising PDUs. |

enum\_le\_gap\_discoverable\_mode

|  |  |  |
| --- | --- | --- |
| **Value** | **Name** | **Description** |
| 0 | le\_gap\_non\_discoverable | Not discoverable |
| 1 | le\_gap\_limited\_discoverable | Discoverable using both limited and general discovery procedures |
| 2 | le\_gap\_general\_discoverable | Discoverable using general discovery procedure |
| 3 | le\_gap\_broadcast | Device is not discoverable in either limited or generic discovery procedure, but may be discovered by using the Observation procedure |
| 4 | le\_gap\_user\_data | Send advertising and/or scan response data defined by the user using le\_gap\_bt5\_set\_adv\_data. The limited/general discoverable flags are defined by the user. |

enum\_le\_gap\_discover\_mode

|  |  |  |
| --- | --- | --- |
| **Value** | **Name** | **Description** |
| 0 | le\_gap\_discover\_limited | Discover only limited discoverable devices |
| 1 | le\_gap\_discover\_generic | Discover limited and generic discoverable devices |
| 2 | le\_gap\_discover\_observation | Discover all devices |

enum\_le\_gap\_address\_type

|  |  |  |
| --- | --- | --- |
| **Value** | **Name** | **Description** |
| 0 | le\_gap\_address\_type\_public | Public address |
| 1 | le\_gap\_address\_type\_random | Random address |
| 2 | le\_gap\_address\_type\_public\_identity | Public identity address resolved by stack |
| 3 | le\_gap\_address\_type\_random\_identity | Random identity address resolved by stack |

## イベントコールバック関数

|  |  |
| --- | --- |
| **イベントコールバック関数** | **概要** |
| ble\_evt\_gatt\_server\_attribute\_value | ローカルGATTデータベース内のアトリビュート値が、リモートGATTクライアントによって変更されたときに呼ばれるコールバック関数ポインタ 【構文】 name.ble\_evt\_gatt\_server\_attribute\_value = my\_evt\_gatt\_server\_attribute\_value; |
| ble\_evt\_le\_connection\_opend | セントラルと接続したときに呼ばれるコールバック関数ポインタ 【構文】 name..ble\_evt\_le\_connection\_opend = my\_evt\_le\_connection\_opend; |
| ble\_evt\_le\_connection\_closed | セントラルと接続が終了したときに呼ばれるコールバック関数ポインタ 【構文】 name.ble\_evt\_le\_connection\_closed = my\_evt\_le\_connection\_closed; |
| ble\_evt\_system\_boot | システムが起動したときに呼ばれるコールバック関数ポインタ 【構文】 name.ble\_evt\_system\_boot = my\_evt\_system\_boot; |
| ble\_evt\_system\_awake | システムがスリープモードから復帰したときに呼ばれるコールバック関数ポインタ 【構文】 name.ble\_evt\_system\_awake = my\_evt\_system\_awake; |
| ble\_evt\_le\_gap\_scan\_response | スキャン応答を受信した時に呼ばれるコールバック関数ポインタ 【構文】 name.ble\_evt\_le\_gap\_scan\_response = my\_evt\_le\_gap\_scan\_response; |

## 省電力

BLEの省電力は下記関数によりSleepモードに移行する。

関数：ble112.ble\_cmd\_system\_halt(1)

D7のWAKEUP信号をHighにすることによりWakeupする。