

ランニングエレクトロニクス  
SBDBT シリーズ用  
BLE フィジカルコンピューティング用サンプルプログラム

# ***FirmaSBBLE***(浜サブレー) **ユーザーズマニュアル**



2015/05/29 版

**RE** ***Running Electronics***

## 目次

### 内容

改訂履歴 .....	3
1. はじめに .....	4
2. 本プログラムの概要 .....	5
3. ピンアサイン .....	6
4. GATT プロファイル .....	7
サービス UUID GAP_SERVICE .....	7
サービス UUID GATT_SERVICE .....	7
サービス UUID 0x180A .....	7
サービス UUID 0x180F .....	7
サービス UUID 0xFF00 .....	8
5. 接続試験 .....	11
6. ライセンス .....	12
7. サポート .....	12

## 改訂履歴

日付	内容
13/12/24	初版 サンプルプログラム「firmasbble_131224.zip」をリリース。
15/05/29	動作確認済み Bluetooth アダプタを更新 「firmasbble_150529.zip」をリリース 変更点 ・ iOS の更新により接続時に長いパケットが送られてくるようになりバッファオーバーフローが発生していた問題を修正。 ・ Broadcom Corp. BCM20702A0 (I/O DATA USB-BT40LE)に対応

## 1. はじめに

本書ではランニングエレクトロニクス SBDBT/SBDBT5V（以下本装置）用、「BLE フィジカルコンピューティング用サンプルプログラム FirmaSBBLE（浜サプレー）」（以下本プログラム）の使用方法を説明します。

本装置では SBBLE(サプレー)等の BLE フィジカルコンピューティング用のサンプルプログラムを提供していますが、SBDBT や SBDBT5V はピン数が少なくもっと多くの I/O を扱いたいというご要望があります。

Arduino には Standard Firmata というサンプルプログラムがあり、それを書き込んだ Arduino は操り人形のように UART 経由で I/O を操作することが出来ます。

本プログラムは、BLE 側は SBBLE(サプレー)のインタフェースを持ったまま、UART で Arduino を操る Firmata プロトコルを喋ります。

SBBLE(サプレー)はユカイ工学社さんの konashi のプロファイルに似せて作られていますので、本プログラムでも konashi のプロファイルに近い形で使用可能です。

本プログラムは様々な事に応用が可能だと思いますので、ユーザー様のご利用の目的に応じて追加・変更してご利用いただきますようお願い致します。

本プログラムには欠陥が含まれている可能性がありますので、信頼性や正確性を保証することは出来ません。またその欠陥を修正することを保証できません。

本プログラムご利用の結果についてランニングエレクトロニクスはいかなる責任も負えません。

また、本プログラムの仕様は予告なく変更する場合がありますので、ランニングエレクトロニクスのサイトを確認して最新の情報を得るようにして下さい。

Bluetooth のプロトコルスタックには Matthias Ringwald 氏の btstack を使用していません。

また、内部にて ChaN 氏の xprintf を利用させていただいています。

素晴らしいソフトウェアを公開されている両氏には深く御礼申し上げます。

## 2. 本プログラムの概要

本プログラムは、本装置の他に Arduino※と USB 接続の Bluetooth4.0 に対応したアダプタが必要です。

※現在のバージョンでは Arduino は UNO 又は UNO とピンコンパチの Arduino 専用となっています。

現在動作確認できている Bluetooth アダプタは以下のものです。他のアダプタでも動作する可能性があります。もしこの表にないアダプタが動作いたしましたらランニングエレクトロニクスまでお知らせいただくと助かります。

新たに動作確認がとれたアダプタが見つかりましたら本書を更新していきます。

メーカー	型番	参考価格
iBUFFALO	BSHSBD08BK	販売終了
iBUFFALO	BSBT4D09BK※	1,273 円
プラネックス	BT-Micro4※	1,109 円
Logitech	LBT-UAN04C2BK※	1,208 円
サンワサプライ	MM-BTUD40※	1,299 円
Logitech	LBT-UAN04C1BK	1,820 円
Logitech	LBT-UAN04C2BK	1,236 円
GREEN HOUSE	GH-BHDA42	1,280 円
Princeton	PTM-UBT7	1,445 円
I/O DATA	USB-BT40LE	1,530 円

※ユーザー様からのご報告により動作確認済みとさせていただきます。

本サンプルプログラムでは以下の開発環境で開発しました。

開発ツール	Microchip MPLAB IDE v8.88	フリー版
コンパイラ	Microchip MPLAB C30 コンパイラ v3.31	フリー版

テストで使った相手装置は Apple 社製 iPhone5(iOS7.0.4)を使用しました。iPhone4S以降、iPad3 以降が Bluetooth4.0 対応機種となります。

### 3. ピンアサイン

ピン番	種類	機能
1	入力	NMCLR (リセット入力)(内部 pull-up) 電源 ON のリセットは CPU 内部でかかりますので、それ以外のリセットが必要なければ未接続で構いません。
2	電源	VDD
3	電源	GND
4		未使用(書き込み・デバッグで使用)
5		未使用(書き込み・デバッグで使用)
6		未使用
7	出力	UART TX (ArduinoUNO0 番ピンに接続)※1
8	入力	UART RX (ArduinoUNO1 番ピンに接続)※2
9		未使用
10		未使用

※1 SBDBT では 3.3V 出力となるため、Arduino 側で安定して受信できない事があります。

※2 SBDBT/SBDBT5V では 5V トレランスのポートになるため、5V の入力が可能です。

## 4. GATT プロファイル

本プログラムでは以下の様な GATT プロファイルを含んでいます。

ユーザー様の方で変更される場合、profile.gatt ファイルを編集した後に

```
./compile-gatt.py profile.gatt profile.h attribute.h
```

又は

```
python compile-gatt.py profile.gatt profile.h attribute.h
```

などのように python を使用して profile.h と attribute.h を生成してください。

### サービス UUID GAP\_SERVICE

UUID	属性	値
GAP_DEVICE_NAME	READ	“SBBLE”
GAP_APPEARANCE	READ	0x00,0x00

### サービス UUID GATT\_SERVICE

UUID	属性
GATT_SERVICE_CHANGED	READ

### サービス UUID 0x180A

UUID	属性	値	内容
0x2A24	READ	“SBBLE”	Model Number String
0x2A26	READ	“1.1”	Firmware Revision String
0x2A27	READ	“1.0”	Hardware Revision String
0x2A28	READ	“1.0”	Software Revision String
0x2A29	READ	“Running Electronics”	Manufacturer Name String

### サービス UUID 0x180F

UUID	属性	値	長さ	備考
0x2A19	READ	0x64	1	バッテリー残量(%) 0x64=100%

サービス UUID 0xFF00

UUID	属性	長さ	内容
0x3000	READ   WRITEWOR	1	PIO 入出力設定 ビット 0:PIO0～ビット 5:PIO5 値 0:PIO 入力設定 値 1:PIO 出力設定
0x3001	READ   WRITEWOR	1	PIO プルアップ設定 ビット 0:PIO0～ビット 5:PIO5 値 0:プルアップなし 値 1:プルアップ
0x3002	WRITEWOR	1	PIO 出力値設定 ビット 0:PIO0～ビット 5:PIO5 値 0:L 出力 値 1:H 出力
0x3003	READ   NOTIFY	1	PIO 入力値 ビット 0:PIO0～ビット 5:PIO5 値 0:L 入力 値 1:H 入力
0x3004	READ   WRITEWOR	1	PWM 設定 ビット 0:PIO0～ビット 5:PIO5 値 0:PWM 使用しない 1:PWM 使用する
0x3005	READ   WRITEWOR	5	PWM パラメータ設定 1 バイト目:PIO チャンネル 2 バイト～5 バイト目:PWM period 指定 (単位:マイクロ秒) バイトオーダー:ビッグエンディアン
0x3006	READ   WRITEWOR	5	PWM デューティ設定 1 バイト目:PIO チャンネル 2 バイト～5 バイト目:PWM duty 指定 (単位:マイクロ秒) バイトオーダー:ビッグエンディアン
0x3008	READ	2	アナログ値 0 (単位:ミリボルト) バイトオーダー:ビッグエンディアン
0x3009	READ	2	アナログ値 1 (単位:ミリボルト) バイトオーダー:ビッグエンディアン



0x300B	READ   WRITEWOR	1	I2C 設定 0:I2C を使用しない 1:I2C を使用する(100kHz) 2:I2C を使用する(400kHz) 現在のバージョンでは一度使用すると、リセットするまで PIO には切り替えできません。
0x300C	READ   WRITEWOR	1	I2C START/STOP 0:STOP CONDITION 発行 1:START CONDITION 発行 2:REPEATED START CONDITION 発行
0x300D	WRITEWOR	MIN3 MAX20	I2C 書き込み 0 バイト目:送信長(2 バイト目よりの長さ) 1 バイト目:ビット 0:0 のこと, ビット 1~ビット 7:スレーブアドレス 2 バイト目~:送信データ
0x300E	WRITEWOR	2	I2C 読み込み指示 0 バイト目:受信する長さ 1 バイト目:ビット 0:1 のこと ビット 1~ビット 7:スレーブアドレス (実際に読み込まれたデータは 0x300F で取得する)
0x300F	READ	MAX20	I2C 読み込みデータ取得 0x300E の読み込み指示で読み込まれたデータを返す
0x3010	READ   WRITEWOR	1	UART 設定 0:UART 使用しない 1:UART 使用する (この指定を行う前に予めビットレートを指定しておくこと)
0x3011	READ   WRITEWOR	2	UART ビットレート設定 (単位 ビットレートを 240 で割った値) 例:2400bps:0x000A 例:9600bps:0x0028 例:115200bps:0x01E0

0x3012	READ   WRITEWOR	MAX20	UART 送信 UART に送信する文字列
0x3013	READ   NOTIFY	MAX20	UART 受信 UART から受信された文字列
0x3014	WRITEWOR	1	リセット 0 以外の値を書き込むとリセット
0x3015	READ   NOTIFY	1	ローバッテリー 常に 0
0xFF10	READ	4	SBDBT: 0x33000001 SBDBT5V: 0x50000002 SBXBT: 0x33000003※ SBRBT: 0x50000004※ SBGRBT: 0x50000005※ バイトオーダー:ビッグエンディアン ※将来拡張用
0xFF20	READ   WRITEWOR	1	UART 極性反転設定 ビット 0:RX 信号反転指定 ビット 1:TX 信号反転指定 ここで値を設定した後に UART 設定で使用しないから使用するに变化したときに反映されます。

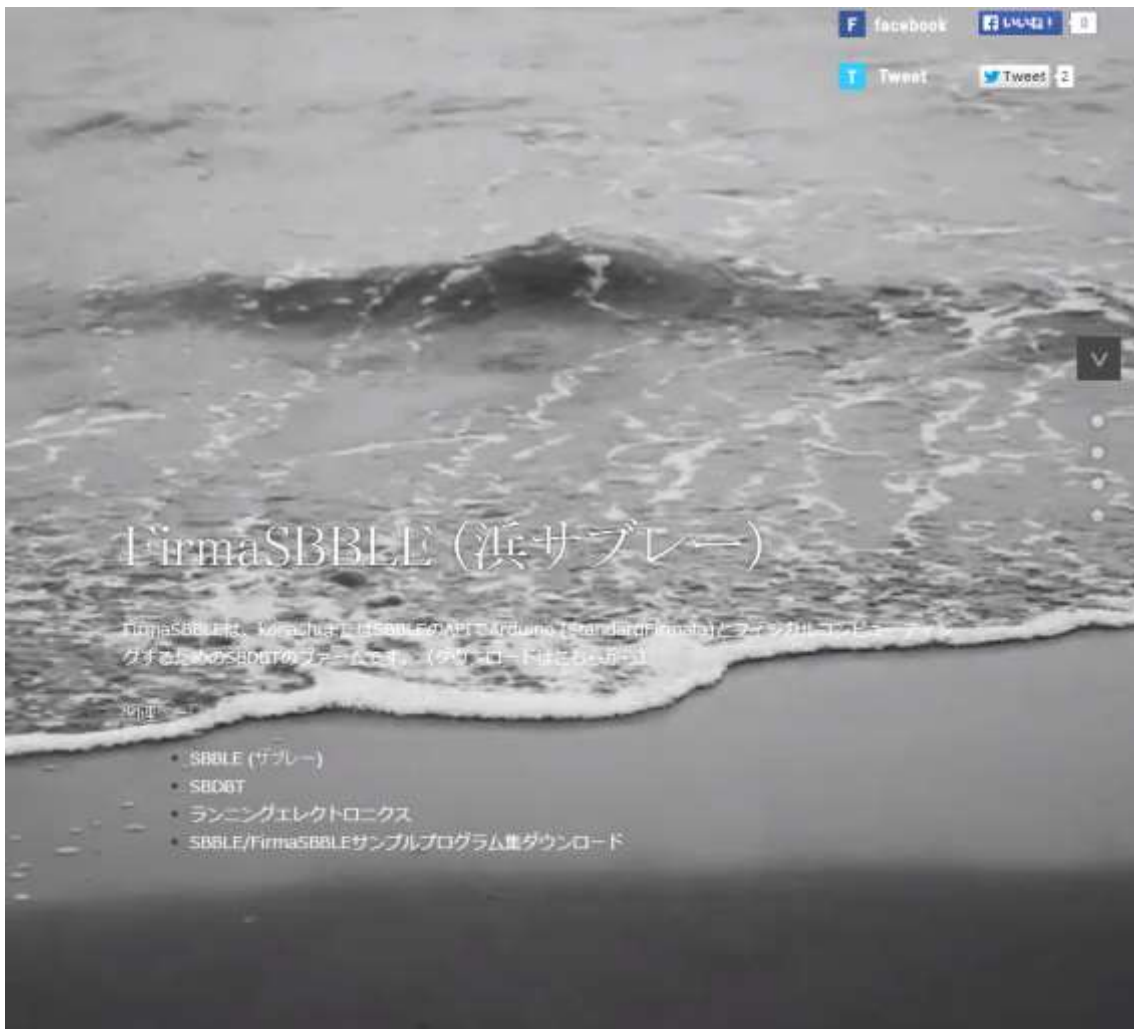
WRITEWOR:write without response

## 5. 接続試験

iOS 側のサンプルプログラムは、今回コラボしていただいている、[Micono Utilities さんのウェブサイト](http://firmasbble.micutil.com/)(<http://firmasbble.micutil.com/>)にて公開されています。

サンプルプログラムの他に、動作させている動画や、使い方等とても詳しく解説していただいています。

そちらをご覧ください。お試し下さい。



## 6. ライセンス

本ファームウェアには、ランニングエレクトロニクスによるプログラムの他に、Matthias Ringwald 氏による Bluetooth プロトコルスタック(btstack)が含まれています。

btstack のウェブサイトで作者 Matthias Ringwald 氏は btstack を商用利用する際は連絡してほしいと記載されています。

ランニングエレクトロニクス製の本サンプルプログラム対応基板は Matthias Ringwald 氏とライセンス契約を締結していますので、本ファームウェアを商用利用する際にもご連絡や追加ロイヤリティは必要ありません。

またランニングエレクトロニクスにて作成した部分、btstack に変更を加えている部分に関しても btstack と同じライセンスとさせていただきます。

本ファームウェアのソースコードを非商用でご利用になる場合、Copyright 表記を消さずに本ファームウェアを使用していることを明記すれば使用できます。ランニングエレクトロニクス製マイコン基板上でご利用になる場合は、非商用・商用にかかわらずご利用いただけます。他の装置に流用して商用利用する場合、ランニングエレクトロニクスにご相談ください。

iOS 側の Micono Utilities さんのウェブサイトにて公開されているソースについては、Micono Utilities さんのライセンスに従って下さい。

## 7. サポート

当店のウェブサイトにはサポート掲示板が設置してあります。  
ご質問やご要望等ありましたらお気軽に書き込みください。