

# Renesas Synergy<sup>TM</sup>

# RL78/G1D を使った簡易 BLE 通信

R30AN0305JJ0100 Rev.1.00 2017.4.27

#### 要旨

本書では、SK-S7G2 上の RL78/G1D を用いた Bluetooth Low Energy (BLE)通信(Master/Slave 動作)に対する、サンプルプログラムの仕様、動作確認の手順、ハードウェア構成およびソフトウェア構成について記載しています。

サンプルプログラムの動作確認を行うためには、SK-S7G2 を 2 台用意するか、SK-S7G2 1 台と Android/iOS デバイス 1 台を用意する必要があります(6. Appendix 参照)。

#### 表 1-1 ソフトウェア環境

Renesas Synergy™ Software Package (SSP)	v1.2.0
Renesas Synergy™ Standalone Configurator (SSC)	v5.3.1.002
e <sup>2</sup> studio ISDE	v5.3.1.002
IAR Embedded Workbench® for Renesas Synergy™ (IAR EW for Synergy)	v7.71.1

### 動作確認デバイス

SK-S7G2

# 目次

3
3
3
4
_
5
6
6
6
6
7
7
8
g
10
12
12
13
14
14
15
15
18
19
20
20
20 20
20

#### 1. はじめに

#### 1.1 概要

本書では、SK-S7G2 上の RL78/G1D を用いた BLE 通信 (Master/Slave 動作) に対する、サンプルプログラムの仕様と動作確認の手順、ハードウェア構成およびソフトウェア構成について記載しています。

本サンプルプログラムは、RL78/G1D 上で動作する BLE プロトコルスタック (Modem 構成) を制御する ためのプログラムです。 BLE プロトコルスタックの API の詳細につきましては、Bluetooth Low Energy プロトコルスタック API リファレンスマニュアル (R01UW0088J/R01UW0088E) を参照してください。

サンプルプログラムの動作を確認するためには、SK-S7G2 を 2 台用意するか、SK-S7G2 を 1 台と Android/iOS デバイスを 1 台用意する必要があります。

#### 1.2 RL78/G1D

RL78/G1D は BLE マイコンであり、図 1-1 に示すように RL78/G1D 単独で動作する Embedded 構成と他の MCU によって動作する Modem 構成を用意しています。また、Modem 構成における APP MCU や Embedded 構成におけるアプリケーションは rBLE API によって BLE MCU 上の BLE スタックのサービスを制御します。

SK-S7G2 上の RL78/G1D に書き込まれているファームウェア(BLE プロトコルスタックのバージョン: V1.01)は Modem 構成として動作し、プロファイルは CSCP (Cycling Speed and Cadence Profile)と CPP (Cycling Power Profile)に対応しています。S7G2 は図 1-1 における APP MCU として動作し、RL78/G1D (BLE MCU)と UART を介してシリアル通信を行います。表 1-2 に UART の通信仕様を示します。また、RL78/G1D を利用するには、RL78/G1D のリセットを解除する必要があります。ここでは、S7G2 の汎用ポート P309 から High レベルを出力し、RL78/G1D のリセットを解除します。

RL78/G1D についての詳細は、参考文献 [1][2][3][4][5][6]を参照してください。

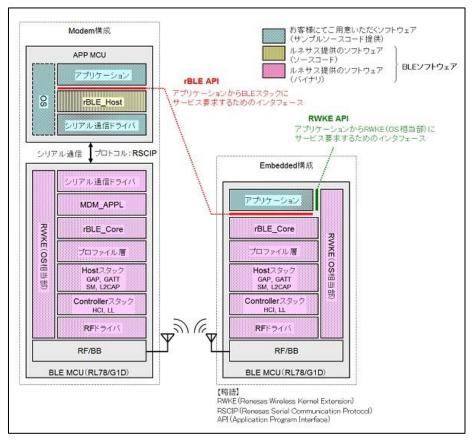


図 1-1 Modem 構成と Embedded 構成

表 1-2 UART 通信仕様

項目	設定値
ボーレート	4800
データ	8 bits
パリティ	なし
ストップビット	1

# 1.3 参考文献

No 資料名		資料番号	
140	良竹石	和文	英文
	Bluetooth Low Energy プロトコルスタック		
1	ユーザーズマニュアル	R01UW0095J	R01UW0095E
2	API リファレンスマニュアル 基本編	R01UW0088J	R01UW0088E
3	API リファレンスマニュアル CSCP 編	R01UW0098J	R01UW0098E
4	rBLE コマンド仕様書	R01AN1376J	R01AN1376E
5	RX113 向けホストサンプルアプリケーション	R01AN3155J	R01AN3155E
	RL78/G1D		
6	ユーザーズマニュアル ハードウェア編	R01UH0515J	R01UH0515E
	Renesas Synergy		
7	Renesas Synergy™ Platform Renesas Bluetooth MCU (RL78/G1D)	-	R12AN0056E
	Bluetooth Low Energy		
8	CSC Meaurement	https://www.bluetooth.com/ja- jp/specifications/gatt/viewer?attributeXmlFile=org.bluetoo th.characteristic.csc_measurement.xml	

#### 2. 仕様

図 2-1 にデバイス構成を示します。SK-S7G2 上で S7G2 と RL78/G1D は UART で接続されています。2 台の SK-S7G2 のうち 1 台を Master、もう 1 台を Slave とします。また、Slave の動作確認用としてスマートフォンを Master とすることも可能です(6. Appendix 参照)。

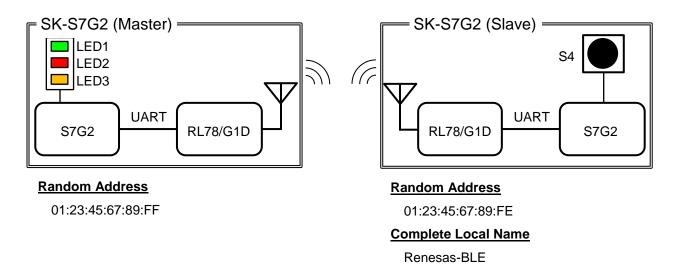


図 2-1 デバイス構成

サンプルプログラムの概要は以下の通りです。

#### • Master (SK-S7G2)

- ▶ 電源投入後、自身のランダムアドレスを「01:23:45:67:89:FF」に変更する。
- ▶ ランダムアドレス変更後、Device 検索を行い、Complete Local Name が「Renesas-BLE」のデバイス (Slave)を発見した場合、接続を開始する。
- ➤ 接続完了後、プロファイル (CSCP) の有効化と Notification の許可を行い、LED1 を点灯する。
- ▶ Slave を発見できなかった場合、もしくは接続が失敗した場合は、LED2 を点灯する。
- ▶ Slave からの Notification を受信する度に LED3 の点灯/消灯を切り替える。

#### Slave

- ▶ 電源投入後、自身のランダムアドレスを「01:23:45:67:89:FE」に変更する。
- ▶ ランダムアドレス変更後、Advertising Packet 中に Complete Local Name として「Renesas-BLE」を設定し、Broadcast を開始する。
- ▶ 接続完了後、プロファイル (CSCP) を有効化する。
- ▶ Master からの Notification 許可後、S4 が押下される度に Notification を送信する。

#### 3. 動作確認

#### 3.1 ソフトウェアのインストール

付属の"Synergy Project Import Guide" (R11AN0023EU0114 Rev1.14)に従い、Master と Slave のプロジェクトをインポートしてください。対応するプロジェクトファイルは表 3-1 の通りです。

表 3-1 プロジェクトファイル

Master	RL78_G1D_master.zip
Slave	RL78_G1D_slave.zip

#### 3.2 RL78/G1D ファームウェアの準備

参考文献 [7] の 13.1 Appendix 1 を参考に、付属の RL78\_G1D\_IM(CSCP,CPP).hex を SK-S7G2 上の RL78/G1D へ書き込んでください。

#### 3.3 確認手順

- 1. Master と Slave それぞれのプログラムを書き込んだ 2 台の SK-S7G2 に電源を投入します。
- 2. 電源投入後、Master と Slave は自動で接続を開始するため、Master 側の LED1 が点灯して接続が完了したことを確認します。
  - ※電源投入から LED1 点灯までに約 10 秒かかります。(5.3.3 Connection シーケンス参照)また、もし LED1 が点灯せずに LED2 が点灯した場合は接続が失敗しているため、再度 Master と Slave の電源を入れ直して、本手順をやり直してください。
- 3. 接続完了後、Slave 側の S4 を押す毎に、Master 側の LED3 が点灯/消灯することを確認します。

## 4. ハードウェア構成

#### 4.1 ブロック図

図 4-1 に Master/Slave のブロック図を示します。1.2 RL78/G1D に記載の通り、S7G2 は RL78/G1D と UART で通信します。使用する機能は、UART(SCI6)とタイムアウト検知のためのタイマ(GPT0)、RL78/G1D のリセット解除や LED 制御のための GPIO です。また Slave のみ S4 が押されたことを検出する ための外部割り込み(IRQ11)も使用します。

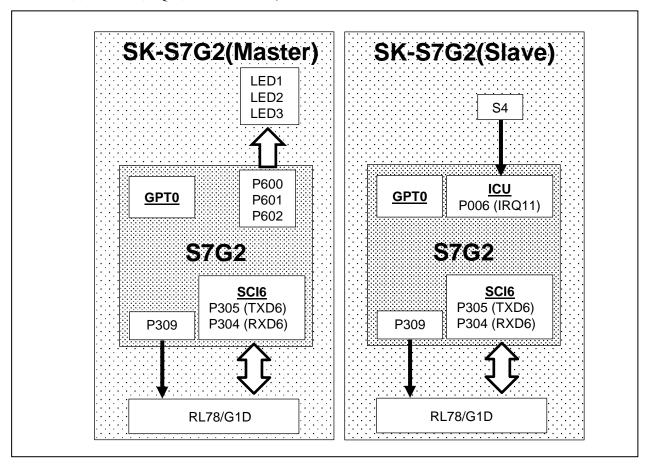


図 4-1 SK-S7G2 (Master/Slave)のブロック図

### 4.2 クロック設定

コンフィグレータのクロックで行う設定を表 4-1 に示します。また、コンフィギュレータ画面での実例を 図 4-2 に示します。

クロック	周波数	算出式
XTAL	24MHz	-
HOCO	20MHz	-
LOCO	32768Hz	-
MOCO	8MHz	-
SUBCLK	32768Hz	-
PLL	240MHz	= XTAL ÷ 2 × 20
Clock Src	240MHz	= PLL
ICLK	240MHz	= Clock Src / 1
PCLKA	120MHz	= Clock Src / 2
PCLKB	3750kHz	= Clock Src / 64
PCLKC	60MHz	= Clock Src / 4
PCLKD	120MHz	= Clock Src / 2
BCLK	120MHz	= Clock Src / 2
SDCLKout	120MHz	= BCLK
BCLKout	60MHz	= BCLK / 2
UCLK	48MHz	= Clock Src / 5
FCLK	60MHz	= Clock Src / 4

表 4-1 クロック設定 (Master/Slave)

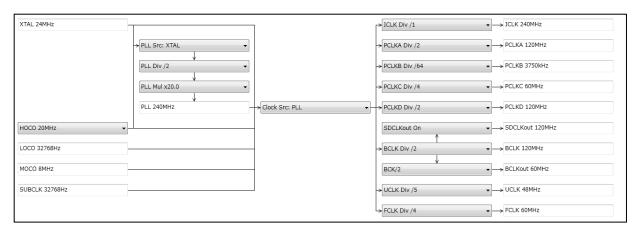


図 4-2 クロック設定画面

# 4.3 ピン設定

コンフィグレータを使って図 4-1 で示す機能の各ピンを表 4-2、表 4-3 のように設定します。

表 4-2 ピン設定一覧 (Master)

カテゴリ	分類	項目	設定値
Ports	P309 (P3)	Mode	Output mode (Initial
			Low)
		Pull up	None
		Drive Capacity	Medium
		Output type	CMOS
	P600 (P6)	Mode	Output mode (Initial
	P601 (P6)		High)
	P602 (P6)	Pull up	None
		Drive Capacity	Medium
		Output type	CMOS
Peripherals	SCI6 (Connectivity:SCI)	Pin Group Selection	Mixed
		Operation Mode	Asynchronous UART
		TXD_MOSI	P305
		RXD_MISO	P304

表 4-3 ピン設定一覧 (Slave)

カテゴリ	分類	項目	設定値
Ports	P006 (P0)	Mode	Input mode
		IRQ	IRQ11_DS
	P309 (P3)	Mode	Output mode (Initial Low)
		Pull up	None
		Drive Capacity	Medium
		Output type	CMOS
Peripherals	SCI6 (Connectivity:SCI)	Pin Group Selection	Mixed
		Operation Mode	Asynchronous UART
		TXD_MOSI	P305
		RXD_MISO	P304

#### 4.4 SSP モジュール設定

本節では本サンプルプログラムが使用している SSP モジュールとその設定について説明します。ただし、プロジェクトの新規作成時から登録されている、 $g\_cgc$  (CGC Driver),  $g\_elc$  (ELC Driver),  $g\_ioport$  (I/O Port Driver)はデフォルトの設定値を利用しているため、説明は割愛します。

図 4-3 に Master/Slave 共通の SSP モジュールの構成、表 4-4、表 4-5 に各モジュールの設定値を示します。 図 4-1 のように RL78/G1D と UART 通信するための UART ドライバと、タイムアウトを検出し、データの 再送を行うためのタイマドライバから構成されています。

UART ドライバの設定値は表 1-1 に従って設定します。タイマ割り込みの周期は 10ms としています。

g\_uart\_ble UART Driver on r\_sci\_uart

g\_timer\_ble Timer Driver on r\_gpt

図 4-3 Master/Slave 共通で使用している SSP モジュール

表 4-4 g\_uart\_ble (Master/Slave 共通)

	プロパティ	値
Common	External RTS Operation	Disable
	Reception	Enable
	Transmission	Enable
	Parameter Checking	Default (BSP)
Module	Name	g_uart_ble
	Channel	6
	Baud Rate	4800
	Data Bits	8bits
	Parity	None
	Stop Bits	1bit
	CTS/RTS Selection	RTS (CTS is
		disabled)
	Name of UART callback	uart_callback_ble
	function to be defined by user	
	Name of UART callback	NULL
	function for the RTS external	
	pin control to be defined by	
	user Clock Source	Internal Clock
	Baudrate Clock Output from SCK pin	Disable
	Start bit detection	Falling Edge
	Noise Cancel	Disable
	Bit Rate Modulation Enable	Enable
	Receive Interrupt Priority	Priority 3
	Transmit Interrupt Priority	Priority 3
	Transmit End Interrupt Priority	Priority 3
	Error Interrupt Priority	Priority 3
	=torrapt r rionty	

プロパティ 値 Parameter Checking Default (BSP) Common Module Name g\_timer\_ble Channel 0 Periodic Mode Period Value 10 Period Unit Milliseconds **Duty Cycle Value** 50 **Duty Cycle Unit Unit Raw Counts** Auto Start True **GTIOCA Output Enabled** False Pin Level Low GTIOCA Stop Level **GTIOCB** Output Enabled False GTIOCB Stop Level Pin Level Low Callback timer\_callback\_ble Priority 3 Interrupt Priority

表 4-5 g\_timer\_ble (Master/Slave 共通)

また、図 4-4 に Slave のみ使用している SSP モジュール、表 4-6 にその設定値を示します。図 4-1 のように、Slave のみ S4 が押されたことを検出するために外部割り込みの SSP モジュールを使用しています。

g\_external\_irq\_pushbutton\_switch\_s4 External Driver on r\_icu

図 4-4 Slave のみ使用している SSP モジュール

表 4-6 g\_external\_irq\_pushbutton\_switch\_s4 (Slave)

	プロパティ	値
Common	Parameter Checking	Default (BSP)
Module	Name	g_external_irq_pushbutton_switch_s4
	Channel	11
	Trigger	Falling
	Digital Filtering	Enabled
	Digital Filtering Sample Clock	PCLK / 64
	Interrupt enabled after initialization	True
	Callback	external_irq_callback_pushbutton_switch_s4
	Interrupt Priority	Priority 4

#### 5. ソフトウェア構成

## 5.1 スレッド構成

図 5-1 にスレッド構成を示します。図 1-1 において、APP MCU は、アプリケーションと rBLE\_Host を 1 つのタスクとして動作させていますが(図 5-1 左図)、本サンプルプログラムでは図 5-1 右図のようにアプリケーション(APP)と rBLE\_Host をそれぞれスレッドとして動作させています。そのため、APP である rBLE App Thread が rBLE API を介して、rBLE\_Host である rBLE Host Thread とデータのやり取りを行います。また、ドライバ(UART や GPT)とのやり取りは rBLE Host Thread が RSCIP を通して行います。

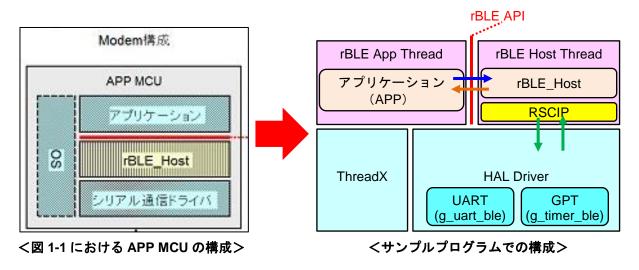


図 5-1 スレッド構成

表 5-1 と表 5-2 にスレッド一覧とそれらの設定を示します。主に動作するスレッドは rBLE Host Thread と rBLE App Thread ですが、これらスレッドの生成と UART やタイマの初期化は BLE Thread が実施します。 スレッドを生成後、BLE Thread は自身のスレッドを終了します。詳細は次節で述べます。

表	5-1	スレッ	ドー覧	(Master/Slave)	١

スレッド名	スレッドの機能
BLE Thread	RL78/G1D のリセット解除、UART やタイマの初期化、rBLE Host/App Thread の 生成を行う。生成後はスレッドを終了する。
rBLE Host Thread	BLE Thread より生成され、rBLE_Host として動作する。
rBLE App Thread	BLE Thread より生成され、rBLE の初期化後、APP として動作する。

表 5-2 スレッド設定値 (Master/Slave)

Name	Symbol	Stack size	Priority	Auto Start	Time slicing
		(bytes)			Intervals (ticks)
BLE Thread	ble_thread	1024	2	Enabled	1
rBLE Host Thread	rble_host_thread	1024	1	Enabled	1
rBLE App Thread	rble_app_thread	1024	1	Enabled	1

#### 5.2 各スレッドの動作シーケンス

図 5-2 に各スレッドの動作シーケンスを示します。

- 1. BLE Thread が UART とタイマを初期化し、RL78/G1D のリセットを解除します。
- 2. BLE Thread が rBLE Host Thread と rBLE App Thread を生成し、自身のスレッドを終了します。
- 3. rBLE App Thread が rBLE\_Init 関数をコールして、rBLE を初期化します。
- 4. rBLE APP Thread は、コマンド要求キューを持ち、コマンド要求キューに要求がセットされる毎に対応した rBLE API をコールします。
- 5. rBLE Host Thread は rBLE API がコールされると対応した rBLE コマンドを実行し、BLE MCU (RL78/G1D) と通信します。また、UART やタイマの割り込みが発生した場合は登録されている Callback 関数をコールすることで、rBLE App Thread に対応するイベントを通知します。
- 6. rBLE App Thread はイベントが通知されると、必要に応じて次のコマンド要求キューに要求をセットし、4.の処理を実行します。

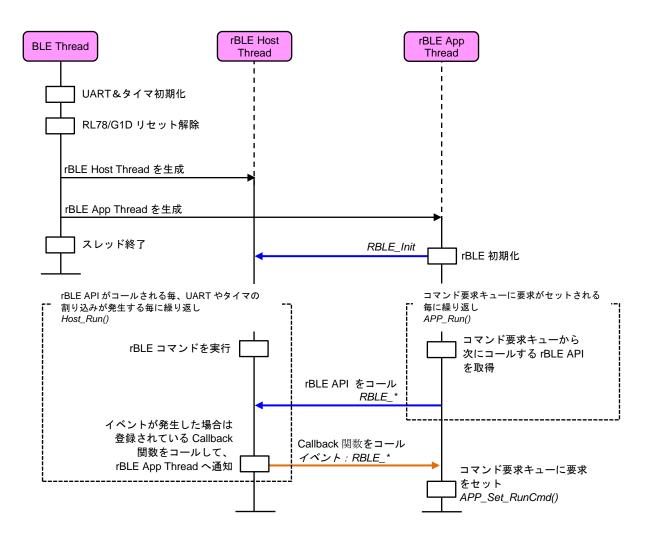


図 5-2 各スレッドの動作シーケンス

#### 5.3 rBLE API シーケンス

Master/Slave それぞれについて、Host MCU(S7G2)とBLE MCU(RL78/G1D)、また Host MCU 内のAPPとrBLE\_Host、BLE MCUのrBLE\_Coreのシーケンスを示します。

#### 5.3.1 メインシーケンス

処理ブロックとして  $Step1\sim5$  までを定義し、処理ブロックの順序と関連するデバイスまたはソフトウェアの範囲をメインシーケンス(図 5-3)に示します。処理ブロック  $Step1\sim10$  の詳細は次節以降に記載します。

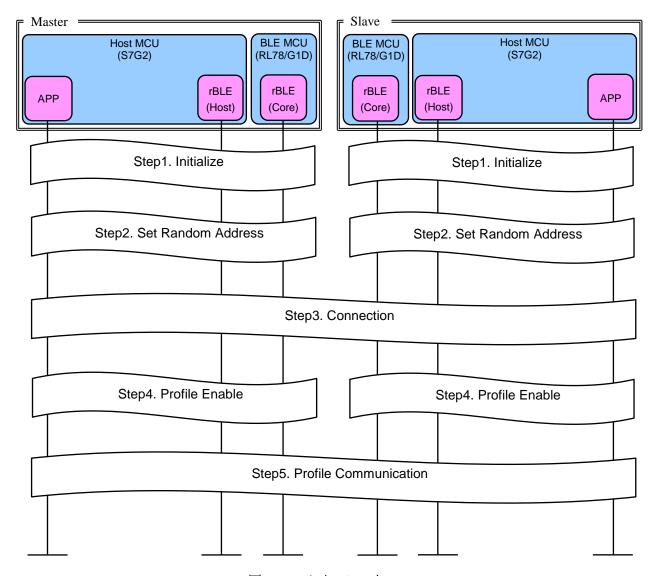


図 5-3 メインシーケンス

#### 5.3.2 Step1. Initialize シーケンス

APP は RBLE\_Init 関数をコールし、rBLE(rBLE\_Host/rBLE\_Core)を初期化します。rBLE の初期化が完了し BLE MCU とのシリアル通信が確立されると、rBLE から RBLE\_MODE\_ACTIVE イベントが通知されます。その後、APP は RBLE\_GAP\_Reset 関数をコールし、GAP を初期化します。初期化が完了すると、rBLE から RBLE GAP EVENT RESET RESULT イベントが通知されます。

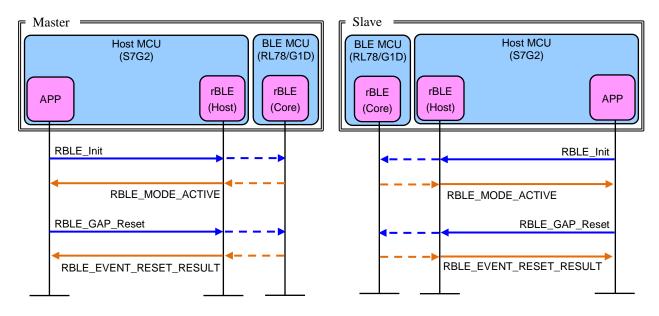


図 5-4 Initialize シーケンス

### 5.3.3 Step2. Set Random Address シーケンス

APP は RBLE\_GAP\_Set\_Random\_Address 関数をコールし、自身のランダムアドレスを変更します (Master: 01:23:45:67:89:FF、Slave: 01:23:45:67:89:FE) 。変更が完了すると、rBLE から RBLE\_GAP\_EVENT\_SET\_RANDOM\_ADDRESS\_COMP イベントが通知されます。

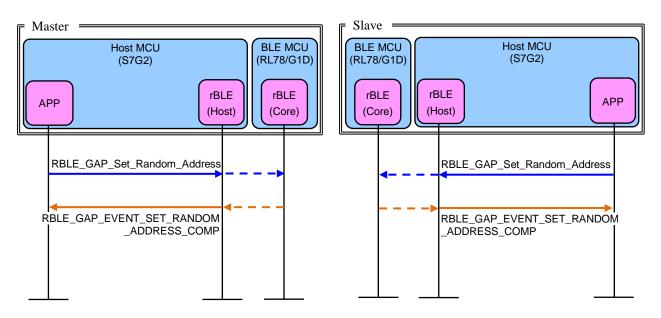


図 5-5 Set Random Address シーケンス

## 5.3.4 Step3. Connection シーケンス

Slave は、RBLE\_GAP\_Broadcast\_Enable 関数をコールし、Broadcast を開始します(この際、Complete Local Name に"Renesas-BLE"を設定)。Broadcast が完了すると rBLE から

RBLE\_GAP\_EVENT\_BROADCAST\_ENABLE\_COMP イベントが通知されます。 Advertising 仕様は表 5-3 の 通りです。

Master は Slave を検索するために RBLE\_GAP\_Device\_Search 関数をコールします。その後、デバイスが発見される毎に RBLE\_GAP\_EVENT\_DEVICE\_SEARCH\_RESULT\_IND イベントが通知されます。その際、Advertising Packet に Complete Local Name が含まれている場合、その値が"Renesas-BLE"であるか確認します。Complete Local Name が"Renesas-BLE"であった場合、そのデバイスが Slave であると判定し、そのデバイスアドレスを保持します(後の RBLE\_GAP\_Create\_Connection 関数用)。また、デバイス検索は 7.68 秒で自動終了し、RBLE\_GAP\_EVENT\_DEVICE\_SEARCH\_COMP イベントが通知されます。

デバイス検索終了後、Master は RBLE\_GAP\_Create\_Connection 関数をコールし、Slave に対して接続を要求します。接続が完了すると、Master/Slave ともに RBLE\_GAP\_EVENT\_CONNECTION\_COMP イベントが通知されます。接続仕様は表 5-4 の通りです。

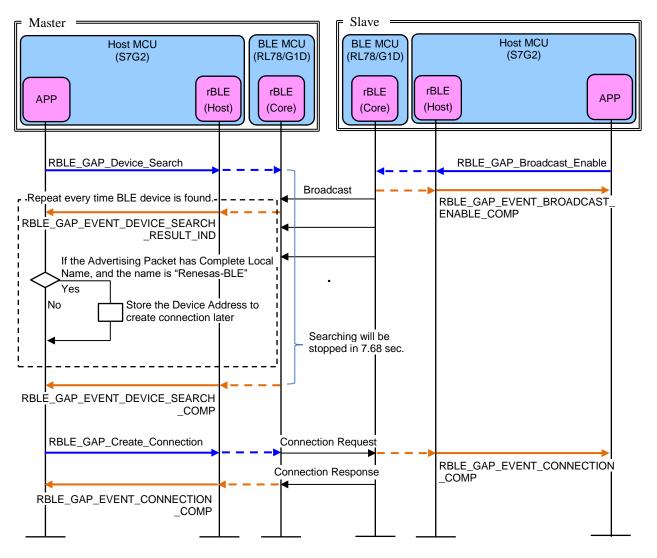


図 5-6 Connection シーケンス

# 表 5-3 Advertising 仕様

Advertisin	g Type	Connectable undirected advertising (ADV_IND)		
Advertising Interval Min		30 [ms]		
Advertising Interval Max		60 [ms]		
Advertisin	g Channel Map	All Channels (37, 38, 39 ch)		
Advertisin	g Data	-		
Length of this Data  Data Type  Flags  Length of this Data  Data Type		2 [bytes]		
		< <flags>&gt; (0x01)</flags>		
		LE General Discoverable Mode		
		BR/EDR Not Supported		
		12 [bytes]		
		< <complete local="" name="">&gt; (0x09)</complete>		
	Local Name	Renesas-BLE		
Scan Response Data		none		

# 表 5-4 接続仕様

Scan Interval	31.25 [ms]
Scan Window Size	30 [ms]
Initiator Filter Policy	Without using the White List
Peer Address Type	Random Address
Peer BD Address	01:23:45:67:89:FE
Own Address Type	Random Address
Minimum of Connection Interval	30 [ms]
Maximum of Connection Interval	30 [ms]
Connection Latency	0
Link Supervision Timeout	5 [s]
Minimum CE Length	0 [ms]
Maximum CE Length	50 [ms]

#### 5.3.5 Step4. Profile Enable シーケンス

データ送信に利用する CSCP (Cycling and Cadence Profile) を Master/Slave ともに有効化します。

Master 側の APP は RBLE\_CSCP\_Collector\_Enable 関数をコールし、CSCP(Collector)を有効化します。有効化が完了すると、RBLE\_CSCP\_EVENT\_COLLECTOR\_ENABLE\_COMP イベントが通知されます。その後、Slave の Notification(CSC Measurement)を許可するために、RBLE\_CSCP\_Collector\_Write\_Char 関数をコールします。有効化が完了すると、RBLE\_CSCP\_EVENT\_COLLECTOR\_WRITE\_CHAR\_RESPONSE イベントが通知されます。

また、Slave 側の APP は RBLE\_CSCP\_Sensor\_Enable 関数をコールし、CSCP(Sensor)を有効化します。 有効化が完了すると、RBLE\_CSCP\_EVENT\_SENSOR\_ENABLE\_COMP イベントが通知されます。Master からの Notification を許可する Client Write Characteristic が送信されると、 RBLE\_CSCP\_EVENT\_SENSOR\_CFG\_INDNTF\_IND イベントが通知されます。

以上でデータ送信の準備が整います。

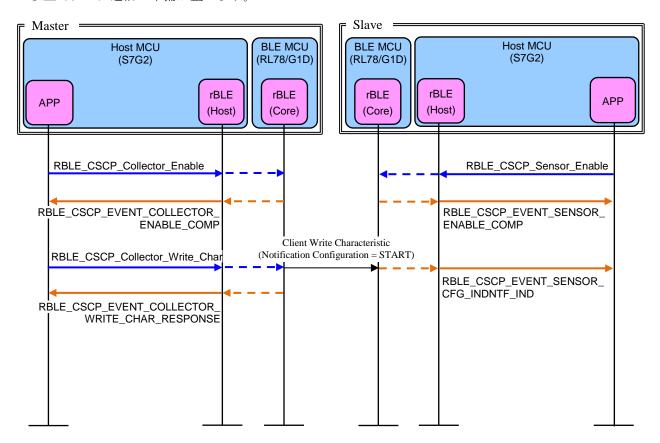


図 5-7 Profile Enable シーケンス

## 5.3.6 Step5. Profile Communication シーケンス

CSCP(Cycling and Cadence Profile)を利用し Notification によるデータ送信を開始します。

Slave 側の APP は External IRQ11 割り込み(S4 が押されることにより発生)が発生すると、RBLE\_CSCP\_Sensor\_Send\_Measurements 関数をコールし、データを Master へ送信します。データ送信が完了すると、RBLE\_CSCP\_EVENT\_SENSOR\_SEND\_MEASUREMENTS\_COMP イベントが通知されます。この際のデータフォーマットは表 5-5 の通りです(詳細は参考文献 [8]を参照してください)。Slave 側の APP は、S4 が押される毎に CSC Measurement 内の Cumulative Wheel Revolutions の値をインクリメント(初期値:0x00000000)して送信します。また、他の Last Wheel Event Time、Cumulative Crank Revolutions、Last Crank Event Time には、それぞれ 0x0001,0x0002,0x0003 を設定します。Flags は Wheel Revolution Data と Crank Revolution Data を有効にするために 0x03 を設定します。

Master 側の APP は Notification を受信すると、RBLE\_CSCP\_EVENT\_COLLECTOR\_MEASUREMENT\_NTF イベントが通知されます。

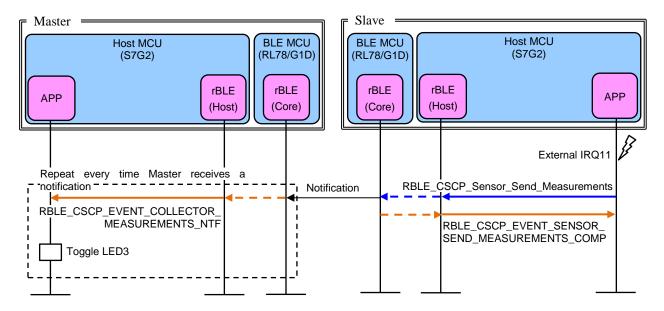


図 5-8 Profile Communication シーケンス

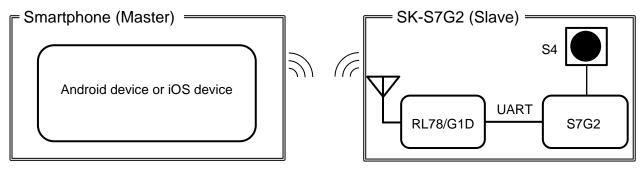
Fields	Value
Flags	0x03 (Wheel/Crank Revolution Data ともに有効化)
Cumulative Wheel Revolutions	S4 が押される毎にインクリメント (初期値:0x00000000)
Last Wheel Event Time	0x0001
Cumulative Crank Revolutions	0x0002
Last Crank Event Time	0x0003

表 5-5 CSC Measurement の内容

#### 6. Appendix

## 6.1 スマートフォンによる確認手順

本節では SK-S7G2 の代わりに Master としてスマートフォン(Android/iOS デバイス)を使用する際の確認 手順を示します。その際のデバイス構成図は図 6-1 の通りです。



## **Random Address**

01:23:45:67:89:FE

#### **Complete Local Name**

Renesas-BLE

図 6-1 スマートフォンを使用した際のデバイス構成図

#### 6.1.1 Android デバイス

SK-S7G2 の代わりに、Master として Android デバイスを使用する際の確認手順を示します。

- 1. 使用する Android デバイスに以下のアプリをインストールします。
  - "BLE Scanner:Read, Write, Notify" Pixel's Perception https://play.google.com/store/apps/details?id=com.macdom.ble.blescanner&hl=ja
- 2. [本体設定]→[Bluetooth]を選択し、BluetoothをONにします。
- 3. 1.でインストールしたBLE Scannerを起動します。
- 4. 起動後 SCAN を開始し、Renesas-BLE が表示されることを確認します。 (図 6-2 の (1))
- 5. Renesas-BLE の欄にある緑色の CONNECT ボタンをタップし、接続を開始します。 (図 6-2 の (2))
- 6. Service 一覧の CYCLING SPEED AND CADENCE を選択します。 (図 6-3)
- 7. Characteristic 一覧最上位の CSC MEASUREMENT 欄にある青色の N ボタンを押して Notification を許可します。(図 6-4)
- 8. Nボタンが緑色になり、Notification が許可されたことを確認します。(図 6-5)
- 9. SK-S7G2 の S4 を押すたびに Notification が送信され、16 進で表記されたデータの Cumulative Wheel Revolutions に該当する赤線の部分がインクリメントされることを確認します。(図 6-6, 図 6-7)



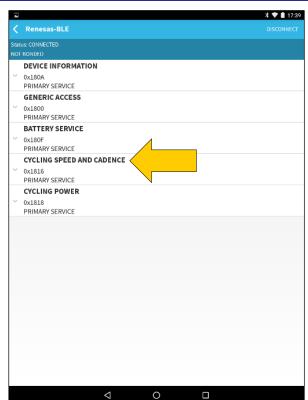
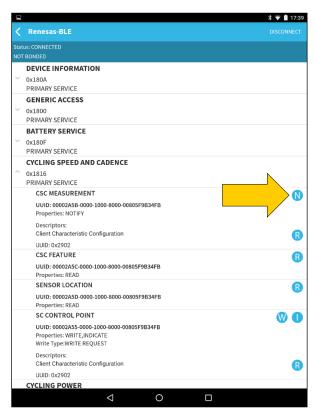


図 6-2



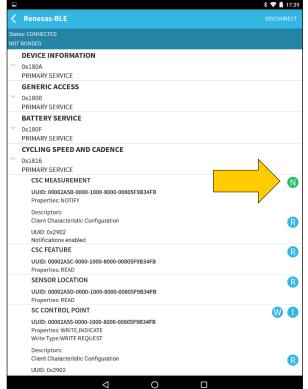
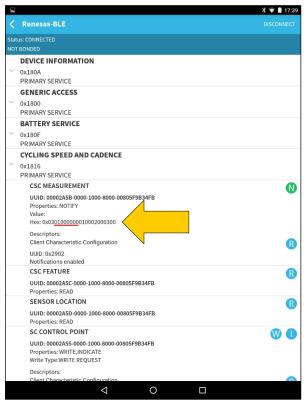


図 6-4



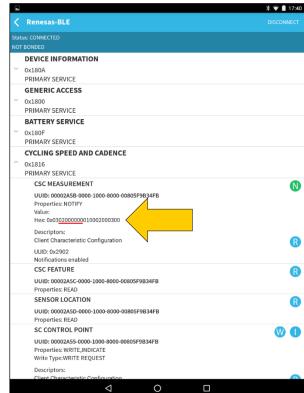


図 6-6

#### 6.1.2 iOS デバイス

SK-S7G2 の代わりに、Master として iOS デバイスを使用する際の確認手順を示します。

- 1. 使用する Android デバイスに以下のアプリをインストールします。
  - "LightBlue" Punch Through Design https://itunes.apple.com/jp/app/lightblue-explorer-bluetooth-low-energy/id557428110?mt=8
- 2. [本体設定]→[Bluetooth]を選択し、BluetoothをONにします。
- 3. 1.でインストールしたLightBlueを起動します。
- 4. 起動後 SCAN を開始し、Renesas-BLE が表示されることを確認します。選択することで接続を開始します。 (図 6-8 )
- Connected となっていることを確認し(図 6-9)、下へスクロールして「UUID: 1816」内の「0x2A5B」を選択します。(図 6-10)
- 6. Listen for notifications をタップして Notification を許可し(図 6-11)、Stop Listening と表示されることを確認します。(図 6-12)
- 7. SK-S7G2 の S4 を押すたびに Notification が送信され、16 進で表記されたデータの Cumulative Wheel Revolutions に該当する赤線の部分がインクリメントされることを確認します。 (図 6-13 (1), (2))



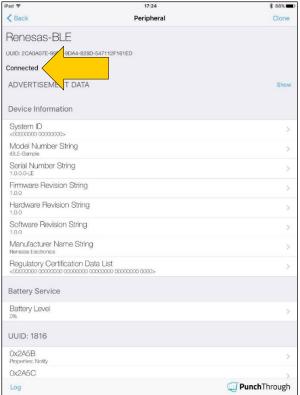
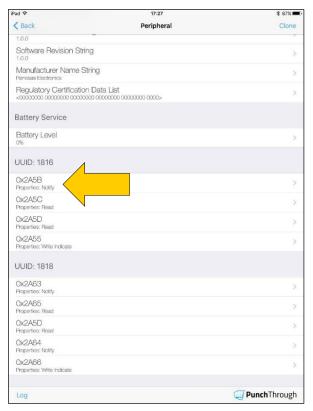


図 6-8



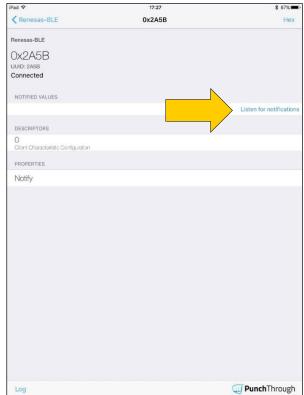


図 6-10

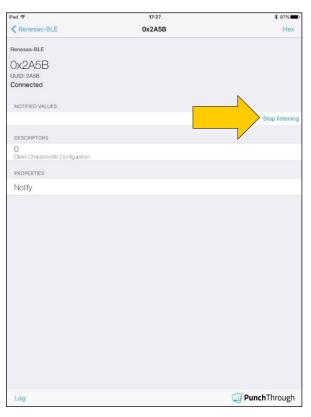


図 6-11



図 6-13

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ <a href="http://japan.renesas.com/">http://japan.renesas.com/</a>

お問合せ先

http://japan.renesas.com/contact/

すべての商標および登録商標は、それぞれの所有者に帰属します。

# 改訂記録

		改訂内容		
Rev.	発行日	ページ	ポイント	
1.0.0	2017.4.27	-	初版	

#### 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意 事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

#### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は,製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス(予約領域)のアクセス禁止

【注意】リザーブアドレス(予約領域)のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス(予約領域)があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。 プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子(または外部発振回路)を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子(または外部発振回路)を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

- 1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害(お客様または第三者いずれに生じた損害も含みます。以下同じです。)に関し、当社は、一切その責任を負いません。
- 2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の 知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
- 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 4. 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
- 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準: コンピュータ、OA機器、通信機器、計測機器、AV機器、

家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準: 輸送機器(自動車、電車、船舶等)、交通制御(信号)、大規模通信機器、

金融端末基幹システム、各種安全制御装置等

当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム(生命維持装置、人体に埋め込み使用するもの等)、もしくは多大な物的損害を発生させるおそれのある機器・システム(宇宙、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等)に使用されることを意図しておらず、これらの用途に使用することはできません。たとえ、意図しない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

- 6. 当社製品をご使用の際は、最新の製品情報(データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等)をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
- 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
- 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、 当社は、一切その責任を負いません。
- 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル(無人航空機を含みます。)の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
  - 当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
- 10. お客様の転売、貸与等により、本書(本ご注意書きを含みます。)記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
- 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
- 12. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を 直接または間接に保有する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.3.0-1 2016.11)



■営業お問合せ窓口

http://www.renesas.com

営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

■技術的なお問合せおよび資料のご請求は下記へどうぞ。 総合お問合せ窓口: https://www.renesas.com/contact/

© 2017 Renesas Electronics Corporation. All rights reserved.

Colophon 5.0