User's Manual

32

Bluetooth Low Energy プロトコルスタック 基本パッケージ

ユーザーズマニュアル

ルネサスマイクロコンピュータ
RX ファミリ／RX200 シリーズ／RX23W グループ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、
予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ルネサス エレクトロニクス
www.renesas.com

Rev.2.61　2025.03

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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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。
6. 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、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、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想定していません。たとえ、当社が想定してい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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、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。
10. 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。
11. 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。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。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 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。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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直接的、間接的に支配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.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）
www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。
www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。
Bluetooth® ワードマークおよびロゴは登録商標であり、Bluetooth SIG, Inc. が所有権を有します。ルネサス エレクトロニクス株式会社は使用許諾の下でこれらのマークおよびロゴを使用しています。
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。

© 2025 Renesas Electronics Corporation. All rights reserved.

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. 静電気対策

   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

   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. 電源オフ時における入力信号

   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

   未使用端子は、「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。

5. クロックについて

   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。

6. 入力端子の印加波形

   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

   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、将来の拡張機能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。

8. 製品間の相違について

   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、フラッシュメモリ、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。

このマニュアルの使い方

1.　目的と対象者

このマニュアルは、本ソフトウェアの使用方法と提供機能をユーザに理解していただくためのマニュアルです。本ソフトウェアを用いた Bluetooth® Low Energy アプリケーションを設計するユーザを対象にしています。このマニュアルを使用するには、マイクロコンピュータやそのソフトウェア開発環境および Bluetooth® Low Energy に関する基本的な知識が必要です。

このマニュアルは、大きく分類すると、製品の概要、インストール方法、ビルド方法、提供機能の使用方法で構成されています。

本ソフトウェアは、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中や各章の最後などに記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

RX23W グループでは次のドキュメントを用意しています。ドキュメントは最新版を使用してください。最新版はルネサス エレクトロニクスのホームページに掲載されています。

ドキュメントの種類	記載内容	資料名	資料番号
データシート	ハードウェアの概要と電気的特性	RX23W グループ データシート	R01DS0342JJ
ユーザーズマニュアル ハードウェア編	ハードウェアの仕様（ピン配置、メモリマップ、周辺機能の仕様、電気的特性、タイミング）と動作説明	RX23W グループ ユーザーズマニュアル ハードウェア編	R01UH0823JJ
ユーザーズマニュアル ソフトウェア編	CPU・命令セットの説明	RX ファミリ RXv2 命令セットアーキテクチャ ユーザーズマニュアル ソフトウェア編	R01US0071JJ
ユーザーズマニュアル ミドルウェア編	製品の概要、インストール方法、ビルド方法、提供機能の使用方法	Bluetooth Low Energy プロトコルスタック 基本パッケージ ユーザーズマニュアル	本ユーザーズマニュアル
アプリケーションノート	基板設計上の注意事項	RX ファミリ ハードウェアデザインガイド	R01AN1411JJ
		RX23W グループ Bluetooth 専用クロック周波数の調整手順	R01AN4762JJ
	応用例参考プログラムなど	RX23W グループ BLE モジュール Firmware Integration Technology	R01AN4860JJ
		Bluetooth Low Energy プロファイル開発者ガイド	R01AN6459JJ
		RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド	R01AN5504JJ
Renesas Technical Update	製品の仕様、ドキュメント等に関する速報	—	—

2.　略語および略称の説明

略語／略称	英語名
API	Application Programming Interface
ATT	Attribute Protocol
BD	Bluetooth Device
BD_ADDR	Bluetooth Device Address
BSP	Board Support Package
BTTS	Bluetooth Test Tool Suite
CLI	Command Line Interface
CMT	Compare Match Timer
CSRK	Connection Signature Resolving Key
DFU	Device Firmware Update
ECDH	Elliptic curve Diffie–Hellman key exchange
EDIV	Encrypted Diversifier
FIT	Firmware Integration Technology
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface
L2CAP	Logical Link Control and Adaptation Protocol
IRK	Identity Resolving Key
LE	Low Energy
LL	Link Layer
LTK	Long Term Key
MCU	Micro Controller Unit
MITM	Man-in-the-middle
OOB	Out of Band
OS	Operating System
OTA	Over The Air
PHY	Physical layer
QE	Quick and Effective tool solution
RF	Radio Frequency
RFP	Renesas Flash Programmer
RPA	Resolvable Private Address
RSK	Renesas Starter Kit
RSSI	Received Signal Strength Indication
RSSK	Renesas Solution Starter Kit
SC	Smart Configurator
SCI	Serial Communication Interface
SM	Security Manager

略語／略称	英語名
SMP	Security Manager Protocol
STK	Short Term Key
TB	Target Board
TK	Temporary Key
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UUID	Universal Unique Identifier
VS	Vendor Specific
WDT	Watchdog Timer

目次

1. 概要 1
1.1 特長 1
1.2 提供機能 2
2. インストールガイド 3
2.1 パッケージ構成 3
2.2 プロジェクト構成 5
2.3 ビルド環境 6
2.4 インストール 7
2.4.1 BLE FIT モジュール 7
2.4.2 デモプロジェクト 8
3. ソフトウェア構成 9
3.1 Bluetooth LE Protocol Stack 10
3.1.1 Bluetooth LE Protocol Stack の構成 10
3.1.2 Bluetooth LE Protocol Stack ライブラリ 13
3.2 app_lib 16
3.2.1 抽象 API 16
3.2.2 ソフトウェアタイマ 16
3.2.3 セキュリティデータ管理 16
3.2.4 プロファイル共通部 17
3.2.5 ロガー 17
3.2.6 コマンドライン 17
3.2.7 LED and Switch 制御 17
3.2.8 BLE タスク制御 17
4. 設定 18
4.1 コンフィギュレーションオプション 18
4.2 セクション 26
4.2.1 アプリケーションのリンク・オプションの変更 27
4.3 Bluetoothデバイスアドレス 30
4.4 Bluetoothデバイス名 30
4.4.1 Advertising による通知 30
4.4.2 Device Name Characteristic による通知 32

5. 独自機能 33
5.1 コマンドライン 33
5.1.1 GAP コマンド 35
5.1.2 Vendor Specific (VS)コマンド 48
5.1.3 SYS コマンド 53
5.1.4 BLE コマンド 54
5.2 セキュリティデータ管理 55
5.2.1 セキュリティデータ管理情報 56
5.2.2 ローカルデバイスのセキュリティデータ 57
5.2.3 リモートデバイスのセキュリティデータ 59
5.3 RF通信タイミング通知機能 61
5.3.1 Connection イベントの通知タイミング 62
5.3.2 Advertising イベント通知タイミング 63
5.3.3 Scan/Initiator イベント通知タイミング 63
5.3.4 RF スリープモードイベント通知タイミング 64
5.3.5 RF 通信タイミング通知仕様 65
5.4 デバイス固有データ 66
5.4.1 デバイス固有データの配置ブロック指定 66
5.4.2 デバイス固有データのフォーマット 67
5.4.3 ユーザ領域(ROM)への書き込み 67
5.4.4 データ領域(E2 データフラッシュ)への書き込み 68
5.4.5 RX23W のフラッシュメモリプロテクト機能 70
5.4.6 BD アドレスの決定方法 71

6. HCI モード 72
6.1 ソフトウェア構成 72
6.2 デモプロジェクト 73
6.2.1 プロジェクトのデバイス種別変更 75
6.3 UARTドライバ 79
6.3.1 UART ドライバのコンフィギュレーション 79

7. 付録 81
7.1 Target Board for RX23Wのファームウェア書き込み手順(デバイス固有データ保持) 81

1. 概要

1.1 特長

Bluetooth Low Energy プロトコルスタック 基本パッケージは、RX23W グループ向けの Bluetooth® Core Specification version 5.0 の Bluetooth Low Energy (以下、Bluetooth LE)仕様に準拠した BLE FIT モジュールおよび Bluetooth 機能を利用するアプリケーション開発のための基本的なソフトウェアを提供します。

1.2 提供機能

表 1.1 に Bluetooth Low Energy プロトコルスタック 基本パッケージが提供するソフトウェアの機能概要を示します。

表 1.1 提供機能概要

分類	機能一覧
BLE FIT モジュール	RX23W グループ向け BLE FIT モジュールでは、以下の Bluetooth 機能を R_BLE API として提供します。 Bluetooth 5.0 機能 • LE 2M PHY • LE Coded PHY • LE Advertising Extensions • LE Channel Selection Algorithm #2 • High Duty Cycle Non-Connectable Advertising Bluetooth 4.2 機能 • LE Secure Connections • Link Layer Privacy • LE Data Packet Length Extension • Link Layer Extended Scanner Filter Policies Bluetooth 4.1 機能 • LE L2CAP Connection Oriented Channel Support • Low Duty Cycle Directed Advertising • 32-bit UUID Support in LE • LE Link Layer Topology • LE Ping また、Bluetooth LE アプリケーション向けの以下の独自機能を提供します。 [app_lib] • 抽象 API • ソフトウェアタイマ • セキュリティデータ管理 • ロガー • コマンドライン • LED and Switch 制御 • BLE タスク制御 BLE FIT v2.50 以降はプロファイル共通部を含みません。QE for BLE(V1.6.0 以降)で提供されます。 [Bluetooth LE Protocol Stack] • RF 通信タイミング通知 • デバイス固有データ
デモプロジェクト	以下のデモプロジェクトを Bluetooth Low Energy 動作チェック用に用意しています。 • カスタムプロファイル GATT サーバアプリケーション • カスタムプロファイル GATT クライアントアプリケーション • HCI mode
ユーティリティ	HCI mode 用に以下のツールを用意しています。 • パブリック BD アドレス設定ツール(BDAddrWriter) • キャリブレーション操作プログラム(CLVALTune)

2. インストールガイド

2.1 パッケージ構成

Bluetooth Low Energy プロトコルスタック 基本パッケージは Renesas の RX ファミリ用 Bluetooth Low Energy プロトコルスタックの Web ページ (https://www.renesas.com/software-tool/bluetooth-low-energy-protocol-stack-rx-family) 、あるいは e²studio のスマート・ブラウザー(図 2.1)から「RX23W グループ BLE Module Firmware Integration Technology アプリケーションノート」をダウンロードすることで入手可能です。BLE FIT モジュールのみダウンロードする場合は「2.4 インストール」を参照してください。

図 2.1　e²studio のスマート・ブラウザー

パッケージ構成を表 2.1 に示します。

表 2.1　BLE FIT のパッケージ構成

r01an4860xxXXXX-rx23w-ble-fit.zip	XXXX: リビジョン番号
├─ FITDemos¥	アプリケーションデモプロジェクトフォルダ
│ ├─ ble_demo_rsskrx23w_profile_client.zip	RSSK 向け GATT クライアントプロジェクト一式
│ ├─ ble_demo_rsskrx23w_profile_server.zip	RSSK 向け GATT サーバプロジェクト一式
│ ├─ ble_demo_rsskrx23w_uart_hci.zip	RSSK 向け HCI モードプロジェクト一式
│ ├─ ble_demo_tbrx23w_FreeRTOS_multi_services.zip	Target Board for RX23W 向け FreeRTOS GATT サーバプロジェクト一式
│ ├─ ble_demo_tbrx23w_profile_client.zip	Target Board for RX23W 向け GATT クライアントプロジェクト一式
│ ├─ ble_demo_tbrx23w_profile_server.zip	Target Board for RX23W 向け GATT サーバプロジェクト一式
│ ├─ ble_demo_tbrx23w_uart_hci.zip	Target Board for RX23W 向け HCI モードプロジェクト一式
│ ├─ ble_demo_tbrx23wmodule_profile_client.zip	Target Board for RX23W module 向け GATT クライアントプロジェクト一式
│ ├─ ble_demo_tbrx23wmodule_profile_server.zip	Target Board for RX23W module 向け GATT サーバプロジェクト一式
│ ├─ ble_demo_tbrx23wmodule_uart_hci.zip	Target Board for RX23W module 向け HCI モードプロジェクト一式
│ └─ ROM_Files¥	mot ファイルフォルダ
│ │ ├─ ble_demo_rsskrx23w_profile_client.mot	RSSK 向け GATT クライアントの mot ファイル
│ │ ├─ ble_demo_rsskrx23w_profile_server.mot	RSSK 向け GATT サーバの mot ファイル
│ │ ├─ ble_demo_rsskrx23w_uart_hci.mot	RSSK 向け HCI モードの mot ファイル
│ │ ├─ ble_demo_tbrx23w_FreeRTOS_multi_services.mot	Target Board for RX23W 向け FreeRTOS GATT サーバの mot ファイル
│ │ ├─ ble_demo_tbrx23w_profile_client.mot	Target Board for RX23W 向け GATT クライアントの mot ファイル
│ │ ├─ ble_demo_tbrx23w_profile_server.mot	Target Board for RX23W 向け GATT サーバの mot ファイル
│ │ ├─ ble_demo_tbrx23w_uart_hci.mot	Target Board for RX23W 向け HCI モードの mot ファイル
│ │ ├─ ble_demo_tbrx23wmodule_profile_client.mot	Target Board for RX23W module 向け GATT クライアントの mot ファイル
│ │ ├─ ble_demo_tbrx23wmodule_profile_server.mot	Target Board for RX23W module 向け GATT サーバの mot ファイル
│ │ └─ ble_demo_tbrx23wmodule_uart_hci.mot	Target Board for RX23W module 向け HCI モードの mot ファイル
├─ FITModules¥	FIT モジュールフォルダ
│ ├─ r_ble_rx23w_vX.XX.xml	BLE FIT モジュール xml ファイル
│ ├─ r_ble_rx23w_vX.XX.zip	BLE FIT モジュール本体
│ └─ r_ble_rx23w_vX.XX_extend.mdf	BLE FIT モジュール mdf ファイル
├─ utilities¥	ユーティリティフォルダ
│ ├─ BDAddrWriter.zip	HCI モード向けパブリック BD アドレス設定ツール
│ └─ CLVALTune.zip	HCI モード向けキャリブレーション操作プログラム
├─ r01an4860ejXXXX-rx23w-ble.pdf	BLE FIT モジュール アプリケーションノート(英文)
├─ r01an4860jjXXXX-rx23w-ble.pdf	BLE FIT モジュール アプリケーションノート(和文)
└─ r_ble_api_spec.chm	R_BLE API ドキュメント(英文)

2.2　プロジェクト構成

e²studio 上のスマート・コンフィグレータから BLE FIT モジュール、BLE QE Utility モジュール、QE for BLE、および、その他の FIT モジュールの各種設定を行い、コード生成して出力される、Bluetooth LE アプリケーションのテンプレートのプロジェクト構成を表 2.2 に示します。

表 2.2　Bluetooth LE アプリケーションのテンプレートのプロジェクト構成

[project name]¥	プロジェクトフォルダ
qe_gen¥	
ble¥	QE for BLE が生成したコード一式
app_main.c	ユーザアプリケーションとプロファイルのフレームワーク
gatt_db.c	GATT データベースのプログラム
gatt_db.h	
r_ble_[profile name]s.c	プロファイルの API プログラム
r_ble_[profile name]s.h	
r_ble_[profile name]c.c	
r_ble_[profile name]c.h	
src¥	
smc_gen¥	スマート・コンフィグレータが生成したコード一式
r_ble_rx23w¥	BLE FIT モジュール
lib¥	Bluetooth LE Protocol Stack ライブラリ
src¥	
app_lib¥	Bluetooth LE アプリケーション補助機能
abs¥	抽象 API
board¥	LED and Switch 制御
cli¥	コマンドライン(ターミナルソフトへの入出力処理)
cmd¥	コマンドライン(コマンド実装部分)
discovery¥	プロファイル共通部(discovery 機能)
logger¥	ロガー
profile_cmn¥	プロファイル共通部
rtos¥	BLE タスク制御
sec_data¥	セキュリティ管理データ
timer¥	ソフトウェアタイマ
platform¥	プラットフォーム依存部
driver¥	BLE 用 Data Flash 制御ドライバ
r_ble_pf_config_private.h	BLE コンフィギュレーション制御部
r_ble_pf_configs.c	
r_ble_pf_functions.c	RF ドライバプラットフォーム依存部
r_ble_pf_lowpower.c	MCU 消費電力低減機能プログラム
r_ble_rx23w_if.h	BLE インタフェース定義ファイル
BLE 以外の FIT モジュール	
[project].c	アプリケーションのテンプレート

【注】BLE FIT v2.50 以降はプロファイル共通部(profile_cmn, discovery)を含みません。QE for BLE(V1.6.0 以降)で qe_gen¥ble 配下に提供されます。

2.3　ビルド環境

Bluetooth LE アプリケーションのビルドおよびデバッグに必要なハードウェア条件を表 2.3、ソフトウェア条件を表 2.4 に示します。

表 2.3　ハードウェア条件

ハードウェア	説明
ホスト PC	USB インタフェースを搭載した Windows 10 PC
RX23W ボード	Target Board for RX23W [RTK5RX23W0C00000BJ] Target Board for RX23W module [RTK5RX23W0C01000BJ] RSSK RX23W [RTK5523W8AC00001BJ]
オンチップデバッギングエミュレータ	E2 エミュレータ Lite [RTE0T0002LKCE00000R] または E1 エミュレータ [R0E000010KCE00] RSSK を使用する場合、どちらかのエミュレータが必要となります。 Target Board には E2 エミュレータ Lite 相当のオンボードデバッガが実装されているため、エミュレータを用意する必要はありません。
USB ケーブル	エミュレータおよび対象ボードとの接続に使用します。 E2 または E1 エミュレータ：USB A-miniB 1 本 Target Board：USB A-microB 2 本 RSSK： USB A-microB 1 本

表 2.4　ソフトウェア条件

ソフトウェア		バージョン	説明
CC-RX 環境	統合開発環境 e² studio	2024-01.1 (64bit 版) 以降	ルネサス製デバイス用の統合開発環境
	CC-RX コンパイラ	v2.08.00	RX ファミリ用 C/C++コンパイラ
	Renesas Flash Programmer	v3.06.00 以降	ルネサス製フラッシュ内蔵マイコンのフラッシュメモリ書き込みツール
IAR 環境	統合開発環境 IAR Embedded Workbench for Renesas RX	v4.20.1	IAR 製の RX ファミリ用の統合開発環境 【注】BLE FIT モジュール v1.10 以降

2.4 インストール

2.4.1 BLE FIT モジュール

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、スマート・コンフィグレータを使用した(1)、(3)、(5)の追加方法を推奨しています。ただし、スマート・コンフィグレータは、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)、(4)の方法を使用してください。

(1) e² studio 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
e² studio のスマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: e² studio 編 (R20AN0451)」を参照してください。

(2) e² studio 上で FIT コンフィグレータを使用して FIT モジュールを追加する場合
e² studio の FIT コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。

(3) CS+上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
CS+上で、スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: CS+編 (R20AN0470)」を参照してください。

(4) CS+上で FIT モジュールを追加する場合
CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。

(5) IAREW 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: IAREW 編 (R20AN0535)」を参照してください。

2.4.2 デモプロジェクト

2.4.2.1 e^2 studio プロジェクト

FITDemos フォルダ配下の使用したいデモプロジェクトの Zip ファイルを、なるべく浅いフォルダ階層にインポートしてください。(例： C:¥RenesasBLE) フォルダ階層が深い場合、e^2 studio でのビルド時に中間ファイルの出力が行えず、ビルドが失敗する場合があります。

また、インポート先はブランク、マルチバイト文字、半角のアンパサンド'&'が含まれない場所を選択してください。

デモプロジェクトの Zip ファイルを e^2 studio のワークスペースにインポートする手順の詳細は"BLE モジュール Firmware Integration Technology アプリケーションノート(R01AN4860)"を参照してください。

【注】デモプロジェクトのインポート先が適切でない場合、e^2 studio でのビルドに失敗する場合があります。

2.4.2.2 IAR Embedded Workbench for Renesas RX プロジェクト

FITDemos フォルダ配下の使用したいデモプロジェクトの Zip ファイルを解凍後、eww ファイルをダブルクリックします。IAR Embedded Workbench for Renesas RX 起動後、[プロジェクト]→[すべてを再ビルド]により、ビルドを実施します。ビルド成功後、[プロジェクト]→[ダウンロードしてデバッグ]により、ボード上にファームウェアをダウンロードします。

3. ソフトウェア構成

Bluetooth LE アプリケーションのソフトウェア構成について図 3.1 に示します。

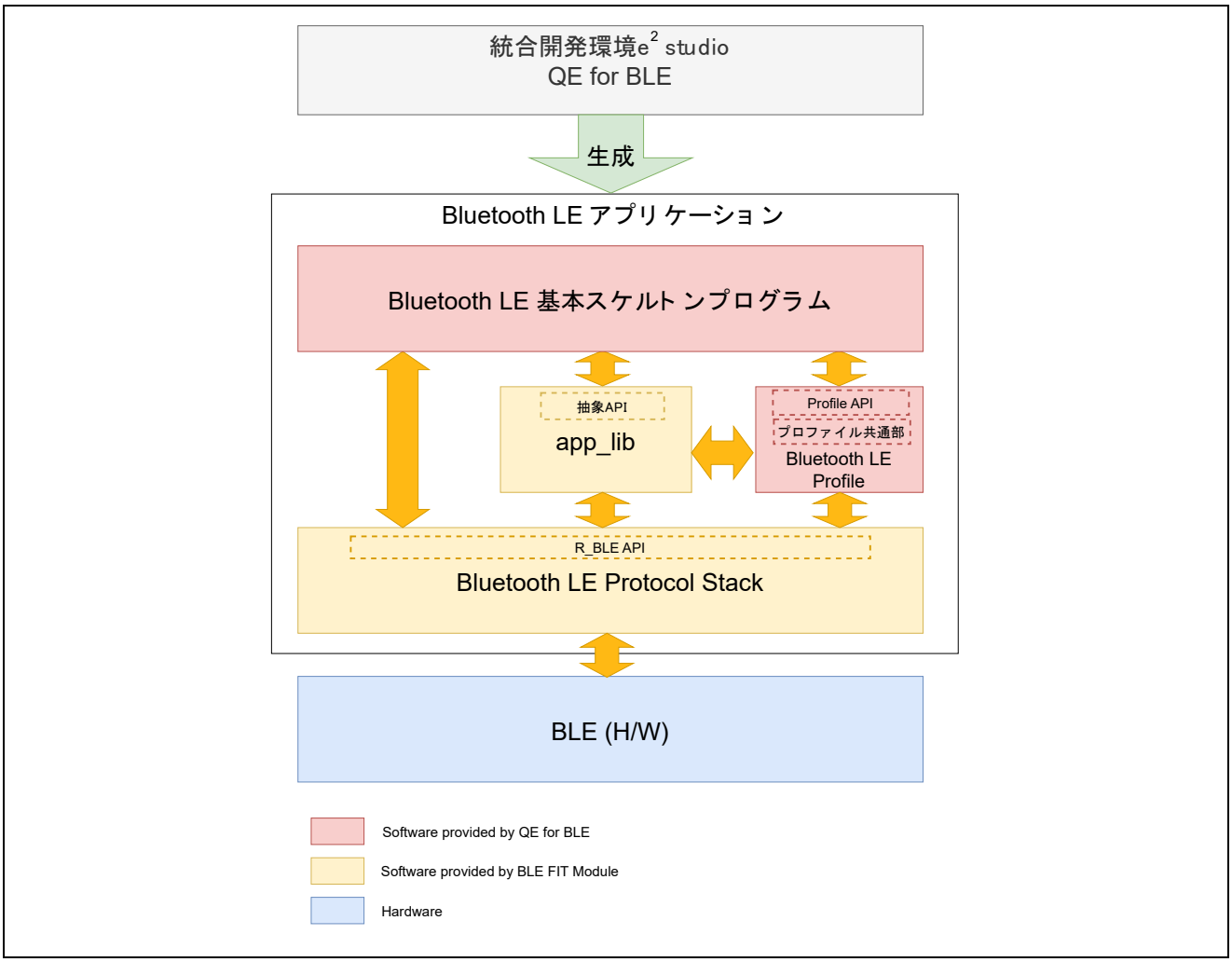


図 3.1　Bluetooth LE アプリケーションのソフトウェア構成

Bluetooth LE アプリケーションは、BLE FIT モジュールで提供される Bluetooth LE Protocol Stack と app_lib、および QE for BLE が生成するスケルトンプログラムと Bluetooth LE Profiles で構成されています。

Bluetooth LE Protocol Stack が提供する R_BLE API 関数をコールすることにより、Bluetooth LE アプリケーションは Bluetooth 機能の使用が可能となります。

app_lib は Bluetooth LE アプリケーションが利用可能な補助機能を提供します。app_lib に含まれる R_BLE API を抽象化した抽象 API(Abstraction API)を使用することによって容易に Bluetooth 機能を使用することが可能です。BLE FIT v2.50 以降はプロファイル共通部を含みません。QE for BLE V1.6.0 以降で提供されます。

e² studio 上で動作するソリューション・ツールキットの QE for BLE は Bluetooth LE Protocol Stack および抽象 API を使用したアプリケーションとプロファイル開発用のスケルトンプログラムを生成します。

ルネサスでは、QE for BLE を使用した、Bluetooth LE アプリケーション開発を推奨しています。

QE for BLE を使用したプロファイルの設計方法は「Bluetooth Low Energy プロファイル開発者ガイド (R01AN6459)」を参照してください。アプリケーションの開発については「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド(R01AN5504)」を参照してください。

3.1 Bluetooth LE Protocol Stack

3.1.1 Bluetooth LE Protocol Stack の構成

Bluetooth LE Protocol Stack の構成について図 3.2 に示します。

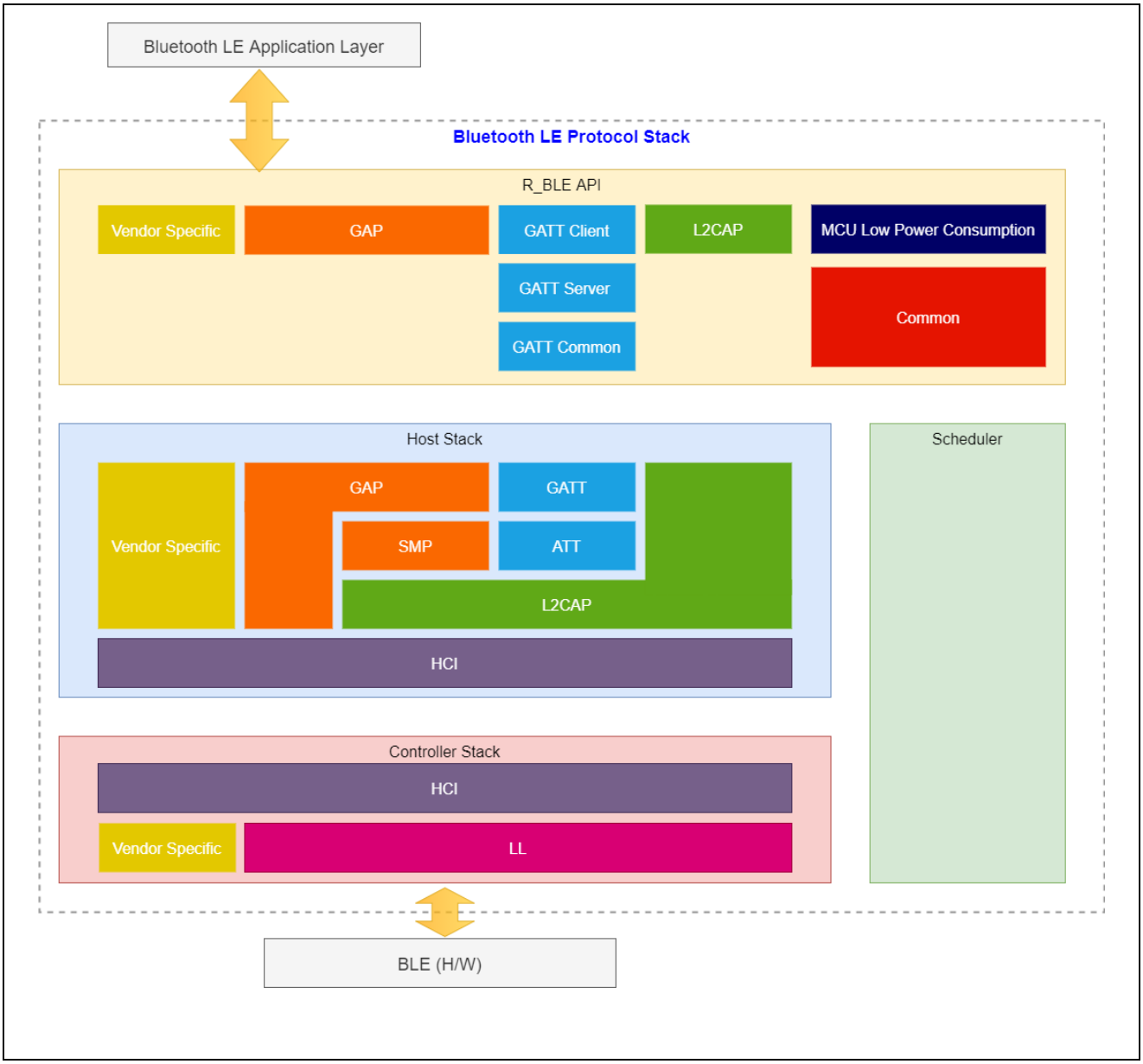


図 3.2　Bluetooth LE Protocol Stack の構成

Bluetooth LE Protocol Stack は R_BLE API, Host Stack, Link Layer, Scheduler から構成されます。

- R_BLE API

R_BLE API は Bluetooth LE アプリケーションに BLE 機能を提供するために表 3.1 の API を提供します。各 API の詳細な仕様については R_BLE API ドキュメント(r_ble_api_spec.chm)を参照してください。

表 3.1　R_BLE API 概要

R_BLE API	プロトコル/ プロファイル	説明
Common API	-	BLE FIT の起動／終了処理や Scheduler を制御する API です。 主な機能 • MCU に搭載されている BLE の開始／終了処理 • Bluetooth LE Protocol Stack 内の task の処理 • Scheduler へのイベントの登録
GAP API	GAP SMP	GAP, SMP に規定されている手続きをサポートする API です。 主な機能 • GAP Advertising, Scan, 接続, セキュリティ • SMP ペアリング
GATT Server API	ATT GATT	サービスに関連した Attribute とデータのセット(GATT Database)を公開する GATT Server 用の API です。 主な機能 • GATT Database へのアクセス • Notification/Indication
GATT Client API	ATT GATT	GATT Server へ要求を発行する GATT Client 用の API です。 主な機能 • Service/Characteristic の Discovery • Characteristic の Read/Write
GATT Common API	ATT GATT	GATT Server/Client 共通で使用する機能についての API です。
L2CAP API	L2CAP	クレジットベースのフロー制御を行うチャネル上のデータ転送用の API です。
Vendor Specific API	-	ルネサス独自の拡張機能を提供する API です。 主な機能 • 拡張版 Direct Test Mode • BD アドレスの設定／取得
MCU Low Power Consumption API	-	MCU の消費電力低減のための API です。

Bluetooth LE Protocol Stack のタイプにより、サポートする API が異なります。各ライブラリがサポートする API については表 3.4 を参照してください。

- Host Stack

Host Stack は Bluetooth SIG が規定したプロトコルおよびプロファイルの機能を提供します。各プロトコル、プロファイルで規定された手続に応じて、R_BLE API より受信したデータを Link Layer に送信し、Link Layer より受信したデータを R_BLE API のイベント、またはデータとして通知します。

- Link Layer

Link Layer は MCU に搭載された BLE ハードウェアを制御し、Advertising, Scan, 接続、データ通信などの Bluetooth LE 機能を HCI(Host Controller Interface)経由で、Host Stack に提供します。Bluetooth LE のコマンド、送信データは Host Stack から Link Layer に送信されます。Bluetooth LE コマンドの結果やリモートデバイスからの受信データは Link Layer から Host Stack に送信されます。

- Scheduler

Scheduler は Common API の R_BLE_Execute()により、Bluetooth LE Protocol Stack の各レイヤのタスク宛に送信されたメッセージキューに従ってタスクを処理します。図 3.3 に Bluetooth LE Protocol Stack の基本シーケンスチャートを示します

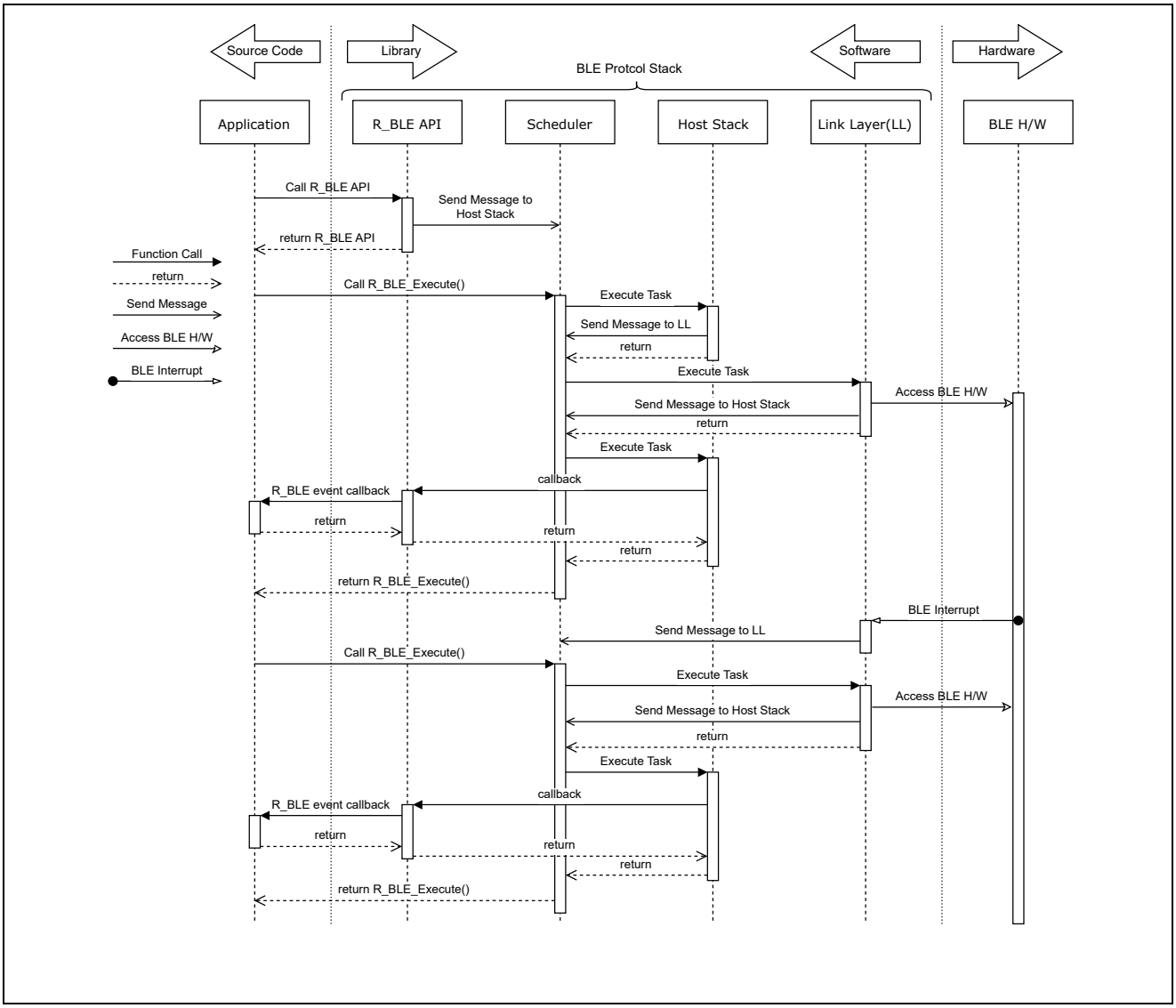


図 3.3　Bluetooth LE Protocol Stack の基本シーケンスチャート

3.1.2 Bluetooth LE Protocol Stack ライブラリ

BLE FIT モジュールはサポートする機能に応じて、表 3.2 に示す 3 タイプの Bluetooth LE Protocol Stack をスタティックライブラリとして提供します。Bluetooth LE アプリケーションで使用する機能に応じたタイプを選択することで、プログラムの ROM/RAM コードサイズを削減することが可能です。

各タイプのコードサイズは「BLE モジュール Firmware Integration Technology (R01AN4860) 2.10 コードサイズ」を参照してください。

表 3.2　Bluetooth LE Protocol Stack のライブラリ

Bluetooth LE Protocol Stack のタイプ	ライブラリファイル名	概要
All features	lib_ble_ps_ccrx_a.lib	Bluetooth LE Protocol Stack がサポートするすべての機能を使用できます。
Balance	lib_ble_ps_ccrx_b.lib	ROM/RAM サイズの使用率が大きい LE Advertising Extensions を無効にしていますが、LE 2M PHY や LE Coded PHY の変更は接続後に行うことが可能です。
Compact	lib_ble_ps_ccrx_c.lib	Bluetooth LE の Peripheral 動作専用のタイプで、センサデバイスなど Central 動作が不要なアプリケーションで使用可能です。

表 3.3 に Bluetooth LE Protocol Stack の各タイプがサポートする機能を示します。

表 3.3　Bluetooth LE Protocol Stack の各タイプがサポートする機能

Bluetooth LE Feature	Bluetooth LE Protocol Stack のタイプ		
	All features	Balance	Compact
LE 2M PHY	Yes	Yes	No
LE Coded PHY	Yes	Yes	No
LE Advertising Extensions	Yes	No	No
LE Channel Selection Algorithm #2	Yes	Yes	No
High Duty Cycle Non-Connectable Advertising	Yes	Yes	Yes
LE Secure Connections	Yes	Yes	Yes
Link Layer privacy	Yes	Yes	Yes
Link Layer Extended Scanner Filter policies	Yes	Yes	No
LE Data Packet Length Extension	Yes	Yes	Yes
LE L2CAP Connection Oriented Channel Support	Yes	No	No
Low Duty Cycle Directed Advertising	Yes	Yes	Yes
LE Link Layer Topology	Yes	Yes	No
LE Ping	Yes	Yes	Yes
GAP Role	Central Peripheral Observer Broadcaster	Central Peripheral Observer Broadcaster	Peripheral Broadcaster
GATT Role	Sever Client	Sever Client	Sever Client
32-bit UUID Support in LE	Yes	Yes	Yes

- LE 2M PHY
  2Msym/s PHY での Bluetooth LE 通信をサポートします。

- LE Coded PHY
  Coded PHY での Bluetooth LE 通信をサポートします。
  1M PHY, 2M PHY より長い距離での通信が可能となります。

- LE Advertising Extensions
  Advertising の拡張機能です。本機能の特徴は以下の通りです。

  - 4 つまでの独立した Advertising の同時実行が可能。
    (コンフィギュレーションオプションの BLE_CFG_RF_ADV_SET_MAX で同時実行する Advertising の数を設定します。)
  - Advertising Data/Scan Response Data のサイズを最大 1650 バイトまで拡張。
    (コンフィギュレーションオプションの BLE_CFG_RF_ADV_DATA_MAX で最大サイズ(バイト)を設定します。)
  - Periodic Advertising が可能。

- LE Channel Selection Algorithm #2
  Version 5.0 で追加されたホッピングチャネルを選択するアルゴリズムにより、チャネルを選択する機能です。

- High Duty Cycle Non-Connectable Advertising
  最小のインターバルが 20 msec までの non-connectable の Advertising をサポートする機能です。

- LE Secure Connections
  Elliptic curve Diffie-Hellman 鍵共有方式(ECDH)により、passive eavesdropping に対応したペアリングをサポートします。

- Link Layer privacy
  定期的に Bluetooth デバイスアドレスを変更することにより、他の Bluetooth LE デバイスからの追跡を回避する機能です。

- LE Data Packet Length Extension
  Bluetooth LE データ通信のパケットサイズを拡張する機能です。
  251 バイトまで拡張することが可能です。

- LE L2CAP Connection Oriented Channel Support
  L2CAP の credit based flow control チャネルを使った通信をサポートする機能です。

- Low Duty Cycle Directed Advertising
  既知のデバイスとの再接続用に Low Duty Cycle の advertising をサポートする機能です。

- LE Link Layer Topology
  Central, Peripheral の両方のロールをサポートし、あるリモートデバイスとの接続では Central として、別のリモートデバイスとの接続では Peripheral として動作できる機能です。

- LE Ping
  接続の暗号化後、MIC を含むパケットの送信要求により、接続が維持されているかどうかをチェックする機能です。

- GAP Role
  GAP Role として、以下をサポートします。

  - Central : 接続要求を Peripheral デバイスに送信するデバイスです。
  - Peripheral : Central からの接続要求を受け入れ、接続を確立するデバイスです。
  - Observer : Advertising をスキャンするデバイスです。
  - Broadcaster : Advertising を送信するデバイスです。

- GATT Role
  GATT Role として、以下をサポートします。

  - Server : GATT Database にサービスが提供する Characteristic を用意し、Client からの要求に応答するデバイスです。
  - Client　: Server が提供するサービスに対して、要求を発行するデバイスです。

- 32-bit UUID Support in LE
  32-bit の UUID をサポートします。GATT で使用する場合は、128-bit に拡張して使用します。

各 Bluetooth LE Protocol Stack ライブラリの R_BLE API サポートを表 3.4 に示します。

表 3.4　各 Bluetooth LE Protocol Stack ライブラリの R_BLE API サポート

R_BLE API	Bluetooth LE Protocol Stack のタイプ		
	All features	Balance	Compact
Common API	Yes	Yes	Yes
GAP API	Yes	C.1	C.1
GATT Common API	Yes	Yes	Yes
GATT Server API	Yes	Yes	Yes
GATT Client API	Yes	Yes	Yes
L2CAP API	Yes	No	No
Vendor Specific API	Yes	Yes	Yes
MCU Low Power Consumption API	Yes	Yes	Yes

C.1 : Bluetooth LE Protocol Stack ライブラリのタイプによって、各 GAP API のサポートは異なります。
詳細については R_BLE API ドキュメント(r_ble_api_spec.chm)を参照してください。

Bluetooth LE Protocol Stack のタイプはコンフィギュレーションファイル(r_ble_rx23w_config.h)の BLE_CFG_LIB_TYPE の定義値により決定します。ビルド時の BLE_CFG_LIB_TYPE により、どのタイプが使用されるのかを判断し、lib_ble_ps_ccrx_x.lib(x: a or b or c)を lib_ble_ps_ccrx.lib にリネームしてリンクします。

3.2 app_lib

app_libの構成について図 3.4 に示します。それぞれの詳細については「5 独自機能」で説明します。

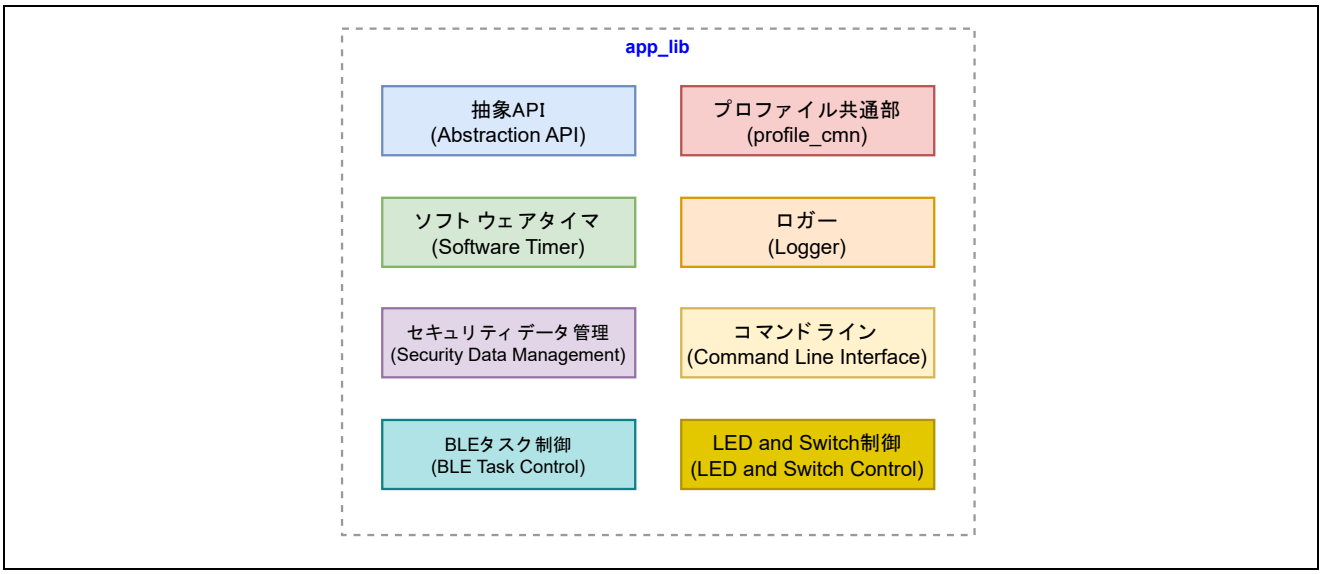


図 3.4　app_lib の構成

【注】BLE FIT v2.50 以降の app_lib はプロファイル共通部を含みません。QE for BLE(V1.6.0 以降)により提供されます。

app_lib が提供する機能について以下に説明します。

3.2.1 抽象 API

抽象 API は Bluetooth LE Protocol Stack でよく使う機能をより簡単に使うための API です。抽象 API の詳細な仕様については R_BLE API ドキュメント(r_ble_api_spec.chm)の「Abstraction API」を参照してください。抽象 API のコードは変更しないでください。

3.2.2 ソフトウェアタイマ

ソフトウェアタイマは MCU のコンペアマッチタイマ(CMT)を使用します。
ソフトウェアタイマを使用する場合、CMT FIT モジュールをアプリケーションに組み込んでください。使用する CMT のチャネルは FIT により、動的に割り当てられます。ソフトウェアタイマについては「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド(R01AN5504) 4.1 ソフトウェアタイマ」を参照してください。

3.2.3 セキュリティデータ管理

ペアリング成功時に Data Flash にボンディング情報を自動的に保存するためのインタフェースを提供します。この機能を使用する場合、BLE FIT モジュールのコンポーネント設定から BLE_CFG_EN_SEC_DATA を”1”に設定してください。セキュリティデータ管理の使用手順については「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド(R01AN5504) 4.4 セキュリティデータ管理」を参照してください。セキュリティデータ管理のデータ構成の詳細については「5.2 セキュリティデータ管理」を参照してください。

3.2.4 プロファイル共通部

プロファイル共通部は Bluetooth LE Profiles で共通する処理です。QE for BLE が生成したプロファイルのソースコードからコールされます。BLE FIT v2.50 以降はプロファイル共通部を含みません。QE for BLE V1.6.0 以降で提供されます。

3.2.5 ロガー

ログ用のメッセージを出力します。出力レベルはコンフィギュレーションオプションの BLE_CFG_LOG_LEVEL で設定します。ロガーについては「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド(R01AN5504) 4.3 ロガー」を参照してください。

3.2.6 コマンドライン

シリアルから入力したコマンドを介して Bluetooth LE 機能を使用する機能です。
コマンドラインは MCU のシリアルコミュニケーションインタフェース(SCI)を使用します。コマンドにはパッケージが用意している標準コマンドとユーザが作成するユーザコマンドがあります。標準コマンドとユーザコマンドについては表 3.5 のドキュメントを参照してください。

表 3.5　コマンドラインの参照ドキュメント

コマンド		参照
標準コマンド	仕様	本ドキュメントの 「5.1 コマンドライン」
	使用手順	「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド (R01AN5504) 4.2.1 標準コマンドの使用手順」
ユーザコマンド	作成方法	「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド (R01AN5504) 4.2.2 ユーザコマンドの作成手順」

3.2.7 LED and Switch 制御

ボード上の LED, SW の制御を行う機能です。LED and Switch 制御を行うための BLE, IRQ FIT のコンフィギュレーションオプションの設定については「RX23W グループ BLE モジュール Firmware Integration Technology (R01AN4860)　4.5.4 LED and Switch 制御機能」を参照してください。LED and Switch 制御の使用方法については「RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド(R01AN5504) 4.5 ボード LED & Switch」を参照してください。

3.2.8 BLE タスク制御

プロジェクト生成時に RTOS に[FreeRTOS]を選択した場合に有効となります。他のタスクから BLE タスク制御を行うときに使用します。BLE タスク制御については「RX23W グループ BLE モジュール Firmware Integration Technology (R01AN4860) 6.4 他のタスクからの BLE タスクの起動」を参照してください。

4. 設定

Bluetooth LE アプリケーションは、BLE FIT モジュールにより用途に合わせて最適な設定が可能です。

4.1 コンフィギュレーションオプション

BLE FIT モジュールのコンフィギュレーションオプションを表 4.1 に示します。コンフィギュレーションオプションの設定は r_ble_rx23w_config.h 内のマクロの定義により決定します。Smart Configurator を使用する場合は、ソフトウェアコンポーネント設定画面でコンフィギュレーションオプションを設定できます。設定値はモジュールを追加する際に、自動的に、r_ble_rx23w_config.h に反映されます。

表 4.1 コンフィギュレーションオプション一覧(1/8)

コンフィギュレーションオプション	説明
BLE_CFG_LIB_TYPE ※デフォルト値は"0"	Bluetooth LE Protocol Stack のタイプを設定します。 0: All features 1: Balance 2: Compact 詳細は「3.1.2 Bluetooth LE Protocol Stack ライブラリ」を参照してください。
BLE_CFG_RF_DBG_PUB_ADDR ※デフォルト値は"{0xFF,0xFF,0xFF,0x50,0x90,0x74}"	BLE FIT モジュールに設定されるパブリックアドレスの初期値を設定します。このオプションのパブリックアドレスはデータフラッシュ、コードフラッシュのパブリックアドレスの領域が ALL 0x00 or 0xFF の場合に採用されます。設定値が ALL 0x00 or 0xFF の場合、74:90:50:FF:FF:FF がパブリックアドレスとして採用されます。
BLE_CFG_RF_DBG_RAND_ADDR ※デフォルト値は"{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF}"	BLE FIT モジュールに設定されるスタティックアドレスの初期値を設定します。このオプションのスタティックアドレスはデータフラッシュ、コードフラッシュのスタティックアドレスの領域が ALL 0x00 or 0xFF の場合に採用されます。設定値が ALL 0x00 or 0xFF の場合、デバイス固有の値から生成したスタティックアドレスを使用します。
BLE_CFG_RF_CONN_MAX ※デフォルト値は"7"	リモートデバイス接続情報管理テーブルの最大エントリ数を設定します。具体的には同時に接続する最大デバイス数、または、保持するボンディング情報の最大数のいずれか大きい方の値を設定します。なお、この値の増減に伴い、Bluetooth LE Protocol Stack が使用するヒープメモリが増減します。"1"～"7"の範囲で設定してください。
BLE_CFG_RF_CONN_DATA_MAX ※デフォルト値は"251"	最大パケットデータサイズ(バイト)を設定します。 "27"～"251"の範囲で設定してください。
BLE_CFG_RF_ADV_DATA_MAX ※デフォルト値は"1650"	最大アドバータイジングデータサイズ(バイト)を設定します。 "31"～"1650"の範囲で設定してください。 Bluetooth LE Protocol Stack のタイプが"0: All features"以外の場合は"31"固定となります。
BLE_CFG_RF_ADV_SET_MAX ※デフォルト値は"4"	最大アドバータイジングセット数を設定します。 "1"～"4"の範囲で設定してください。 Bluetooth LE Protocol Stack のタイプが"0: All features"以外の場合は"1"固定となります。

表 4.1 コンフィギュレーションオプション一覧(2/8)

コンフィギュレーションオプション	説明
BLE_CFG_RF_SYNC_SET_MAX ※デフォルト値は"2"	最大ピリオディックシンクセット数を設定します。 "1"～"2"の範囲で設定してください。 Bluetooth LE Protocol Stack のタイプが"0: All features"以外の場合はこの値は使用しません。
BLE_CFG_EVENT_NOTIFY_CONN_START ※デフォルト値は"0"	接続イベントの開始割り込み発生通知機能の有効／無効を設定します。 0: Disable 1: Enable 開始通知は割り込みを契機としていることから、実際の RF イベントの事後通知となります。
BLE_CFG_EVENT_NOTIFY_CONN_CLOSE ※デフォルト値は"0"	接続イベントの完了割り込み発生通知機能の有効／無効を設定します。 0: Disable 1: Enable Connection の停止コマンドを実行した場合、完了イベントは通知されません。
BLE_CFG_EVENT_NOTIFY_ADV_START ※デフォルト値は"0"	Advertising イベントの開始割り込み発生時通知機能の有効／無効を設定します。 0: Disable 1: Enable 下記のタイミングで通知します。 Primary Adv Ch の開始 Secondary Adv Ch(AdvExt)の開始 Periodic Adv の開始 ※対応する AdvExt が Enable であることが前提 開始通知は割り込みを契機としていることから、実際の RF イベントの事後通知となります。
BLE_CFG_EVENT_NOTIFY_ADV_CLOSE ※デフォルト値は"0"	Advertising イベントの完了割り込み発生時通知機能の有効／無効を設定します。 0: Disable 1: Enable 下記のタイミングで通知します。 Primary Adv Ch の完了 Secondary Adv Ch(AdvExt)の完了 Periodic Adv の完了 ※対応する AdvExt が Enable であることが前提 Advertising の停止コマンドを実行した場合、完了イベントは通知されません。

表 4.1 コンフィギュレーションオプション一覧(3/8)

コンフィギュレーションオプション	説明
BLE_CFG_EVENT_NOTIFY_SCAN_START ※デフォルト値は"0"	Scan イベントの開始割り込み発生通知機能の有効／無効を設定します。 0: Disable 1: Enable Scan Interval=Window の場合、通知されません。 開始通知は割り込みを契機としていることから、実際の RF イベントの事後通知となります。
BLE_CFG_EVENT_NOTIFY_SCAN_CLOSE ※デフォルト値は"0"	Scan イベントの完了割り込み発生通知機能の有効／無効を設定します。 0: Disable 1: Enable Scan Interval=Window の場合、通知されません。 Scan の停止コマンドを実行した場合、完了イベントは通知されません。
BLE_CFG_EVENT_NOTIFY_INIT_START ※デフォルト値は"0"	Initiator イベントの Scan 開始割り込み発生通知機能の有効／無効を設定します。 0: Disable 1: Enable Scan Interval=Window の場合、通知されません。 開始通知は割り込みを契機としていることから、実際の RF イベントの事後通知となります。
BLE_CFG_EVENT_NOTIFY_INIT_CLOSE ※デフォルト値は"0"	Initiator イベントの Scan 完了割り込み発生通知機能の有効／無効を設定します。 0: Disable 1: Enable Scan Interval=Window の場合、通知されません。 接続要求の停止コマンドを実行した場合、完了イベントは通知されません。
BLE_CFG_EVENT_NOTIFY_DS_START ※デフォルト値は"0"	RF_DEEP_SLEEP start 通知機能の有効／無効を設定します。 0: Disable 1: Enable
BLE_CFG_EVENT_NOTIFY_DS_WAKEUP ※デフォルト値は"0"	RF_DEEP_SLEEP wakeup 通知機能の有効／無効を設定します。 0: Disable 1: Enable

表 4.1 コンフィギュレーションオプション一覧(4/8)

コンフィギュレーションオプション	説明
BLE_CFG_RF_CLVAL ※デフォルト値は"6"	32MHz 水晶振動子の調整値をボード環境に応じて設定します。 "0"～"15"の範囲で設定してください。 関連ドキュメント「RX23W グループ Bluetooth 専用クロック周波数の調整手順(R01AN4762)」を参照してください。 RX23W モジュールの型番 R5F523W8CxLN, R5F523W8DxLN を選択した場合、7 に設定してください。
BLE_CFG_RF_DDC_EN ※デフォルト値は"0"	RF 部の DC―DC の有効／無効を設定します。 0: Disable 1: Enable Target Board をご使用の場合は、0 を設定してください。
BLE_CFG_RF_EXT32K_EN ※デフォルト値は"0"	RF 部への slow clock source を設定します。 "0"～"1"の範囲で設定してください。 0: RF_LOCO 使用 1: 外部 32.768kHz 使用 1 にする場合、クロック設定でサブクロックを有効にする必要があります。
BLE_CFG_RF_MCU_CLKOUT_PORT ※デフォルト値は"0"	MCU CLKOUT のポートを設定します。 "0"～"1"の範囲で設定してください。 0: PE3 1: PE4 BLE_CFG_RF_EXT32K_EN が 0 の場合、この値は無視されます。
BLE_CFG_RF_MCU_CLKOUT_FREQ ※デフォルト値は"0"	MCU の CLKOUT からの出力周波数を設定します。 "0"～"1"の範囲で設定してください。 0: MCU CLKOUT frequency 32.768kHz 1: MCU CLKOUT frequency 16.384kHz BLE_CFG_RF_EXT32K_EN が 0 の場合、この値は無視されます。
BLE_CFG_RF_SCA ※デフォルト値は"250"	RF slow clock 用の Sleep Clock Accuracy(SCA)を設定します。 "0"～"500"の範囲で設定してください。 BLE_CFG_RF_EXT32K_EN が 0 の場合、SCA は 250ppm 以上に固定され、この値は無視されます。
BLE_CFG_RF_MAX_TX_POW ※デフォルト値は"1"	最大送信パワーを設定します。 "0"～"1"の範囲で設定してください。 0: max +0dBm 1: max +4dBm

表 4.1 コンフィギュレーションオプション一覧(5/8)

コンフィギュレーションオプション	説明
BLE_CFG_RF_DEF_TX_POW ※デフォルト値は"0"	デフォルトの送信パワーを設定します。 "0"～"2"の範囲で設定してください。 デフォルト送信パワーは BLE_CFG_RF_MAX_TX_POW の設定に依存します。 BLE_CFG_RF_MAX_TX_POW が 0(0dBm)の場合、BLE_CFG_RF_DEF_TX_POW は以下のようになります。 0(High) : 0 dBm 1(Mid) : 0 dBm 2(Low) : -18 dBm BLE_CFG_RF_MAX_TX_POW が 1(+4dBm)の場合、BLE_CFG_RF_DEF_TX_POW は以下のようになります。 0(High) : +4 dBm 1(Mid) : 0 dBm 2(Low) : -20 dBm
BLE_CFG_RF_CLKOUT_EN ※デフォルト値は"0"	CLKOUT_RF 出力を設定します。 以下のいずれかの値を選択してください。 0: No output 5: 4MHz output 6: 2MHz output 7: 1MHz output
BLE_CFG_RF_DEEP_SLEEP_EN ※デフォルト値は"1"	RF Deep Sleep 機能の有効／無効を設定します。 0: Disable 1: Enable
BLE_CFG_MCU_MAIN_CLK_KHZ ※デフォルト値は"4000"	MCU メインクロック周波数(kHz)を設定します。 ボード環境に合わせて設定してください。 HOCO の場合、この値による設定は無視されます。 メインクロックの場合、"1000"から"20000"の範囲で設定してください。 PLL Circuit の場合、"4000"から"20000"の範囲で設定してください。 Smart Configurator の”クロック”画面で入力した周波数を設定してください。
BLE_CFG_DEV_DATA_CF_BLOCK ※デフォルト値は"16"	デバイス固有データを格納する Code Flash(ROM)の Block を設定します。 "-1"～"255"の範囲で設定してください。 "-1"が設定された場合、Code Flash のデバイス固有データを使用しません。 "0"から"15"は Start-Up Program Protection block となっていますので、Start-Up Program Protection 機能を使用する場合は、”0”から”15”を指定しないでください。

表 4.1 コンフィギュレーションオプション一覧(6/8)

コンフィギュレーションオプション	説明
BLE_CFG_DEV_DATA_DF_BLOCK ※デフォルト値は"-1"	デバイス固有データを格納する Data Flash の Block を設定します。 "-1"～"7"の範囲で設定してください。 "-1"が設定された場合、E2 Data Flash のデバイス固有データを使用しません。 BLE_CFG_SECD_DATA_DF_BLOCK で指定した Block と別の Block を指定してください。
BLE_CFG_GATT_MTU_SIZE ※デフォルト値は"247"	GATT 通信で使用する MTU サイズ(バイト)を設定します。 "23"～"247"の範囲で設定してください。
BLE_CFG_NUM_BOND ※デフォルト値は"7"	セキュリティデータ管理機能で保存するボンディング情報の数を設定します。"1"～"7"の範囲で設定してください。 この値を一度決定してコード生成した後、デバッグ等で値を変更すると、変更前にセキュリティデータ管理機能によって管理されていたボンディング情報が Data Flash に残り、再ペアリングができない場合があります。これを避けるため、ファームウェア書き込み後、R_BLE_GAP_DeleteBondInfo()、または"gap auth del remote all" コマンドにより、ボンディング情報を削除してください。
BLE_CFG_EN_SEC_DATA ※デフォルト値は"0"	セキュリティデータ管理機能を有効にします。この機能はペアリング成功時に BLE_CFG_SECD_DATA_DF_BLOCK で示す Block にボンディング情報を保存します。 0: Disable 1: Enable この機能を有効にする場合、Data Flash モジュールのコンポーネントを有効にしてください。
BLE_CFG_SECD_DATA_DF_BLOCK ※デフォルト値は"0"	セキュリティデータ管理機能で Data Flash へのボンディング情報の保存機能で使用する Data Flash Block を設定します。 "0"～"7"の範囲で設定してください。 BLE_CFG_DEV_DATA_DF_BLOCK で指定した Block とは別の Block を指定してください。
BLE_CFG_CMD_LINE_EN ※デフォルト値は"0"	コマンドライン機能の有効／無効を設定します。 0: Disable 1: Enable この機能を有効にする場合、SCI FIT モジュールのコンポーネントを有効にしてください。

表 4.1 コンフィギュレーションオプション一覧(7/8)

コンフィギュレーションオプション	説明
BLE_CFG_CMD_LINE_CH ※デフォルト値は"1"	コマンドライン機能で使用する SCI のチャネルを設定します。以下のいずれかの値を設定してください。 1: SCI1 5: SCI5 8: SCI8 12: SCI12 (BGA 85pin のみ) SCI FIT モジュールの設定で、コマンドライン機能で使用する SCI のチャネルを有効にしてください。 BLE_CFG_CMD_LINE_EN が 0 の場合、この値は無視されます。 HCI モードで使用する SCI は本マクロではなく「6.3.1 UART ドライバのコンフィギュレーション」で設定する必要があります。
BLE_CFG_BOARD_LED_SW_EN ※デフォルト値は"0"	LED and Switch 制御の有効／無効を設定します。 0: Disable 1: Enable この機能を有効にする場合、IRQ FIT モジュールと GPIO FIT モジュールのコンポーネントを有効にしてください。
BLE_CFG_BOARD_TYPE ※デフォルト値は”0”	ボードのタイプを設定します。 "0"～"3"の範囲で設定してください。 0 : Customer board 1 : Target Board 2 : RSSK 3 : Evaluation Board
BLE_CFG_LOG_LEVEL ※デフォルト値は"3"	ログレベルを設定します。 "0"～"3"の範囲で設定してください。 0 : disable 1 : Error 2 : Error & Warning 3 : Error & Warning & Debug

表 4.1 コンフィギュレーションオプション一覧(8/8)

コンフィギュレーションオプション	説明
BLE_CFG_ABS_API_EN ※デフォルト値は"1"	抽象 API の有効／無効を設定します。 0: Disable 1: Enable
BLE_CFG_SOFT_TIMER_EN ※デフォルト値は"1"	app_lib が提供するソフトウェアタイマの有効／無効を設定します。 0: Disable 1: Enable 抽象 API を使用する場合は、この機能を有効に設定してください。
BLE_CFG_MCU_LPC_EN ※デフォルト値は"1"	MCU の低消費電力機能の有効／無効を設定します。 0: Disable 1: Enable
BLE_CFG_HCI_MODE_EN ※デフォルト値は"0"	HCI モードでの起動を設定します。 0: 通常起動 1: HCI モードでの起動

4.2 セクション

Bluetooth LE Protocol Stack ライブラリは Bluetooth LE 通信を使用した Over The Air(OTA)による、Device Firmware Update(DFU)を行うためのしくみを提供します。Bluetooth LE Protocol Stack ライブラリのセクション配置指定を行えるように、Bluetooth LE Protocol Stack ライブラリのコードに対して、表 4.2 に示す、標準のセクション名とは別のセクション名(標準セクション名の先頭に”BLE_”を付与した名称)を指定しています。

表 4.2 Bluetooth LE Protocol Stack ライブラリのセクション名

#	名称	セクション名
1	プログラム領域	BLE_P
2	定数領域	BLE_C
		BLE_C_2
		BLE_C_1
3	初期化データ領域(ROM)	BLE_D
		BLE_D_2
		BLE_D_1
4	初期化データ領域(RAM)	BLE_R
		BLE_R_2
		BLE_R_1
5	未初期化データ領域	BLE_B
		BLE_B_2
		BLE_B_1
6	switch 文分岐テーブル領域	BLE_W
		BLE_W_2
		BLE_W_1
7	リテラル領域	BLE_L

上記の Bluetooth LE Protocol Stack ライブラリのセクション配置指定に伴い、「4.2.1 アプリケーションのリンク・オプションの変更」の設定を行う必要があります。

BLE 初期化を行う、R_BLE_Open() API で BLE セクションは初期化されるため、セクション初期化用テーブルである、DTBL/BTBL テーブルについては変更する必要はありません。

4.2.1 アプリケーションのリンク・オプションの変更

アプリケーションのセクション配置の設定で、表 4.2 の Bluetooth LE Protocol Stack ライブラリのセクション名を追加します。CC-RX では、セクション名にワイルドカード”*”による指定が可能です。e² studio プロジェクトのセクション配置は”Linker”の”Section”で設定します。

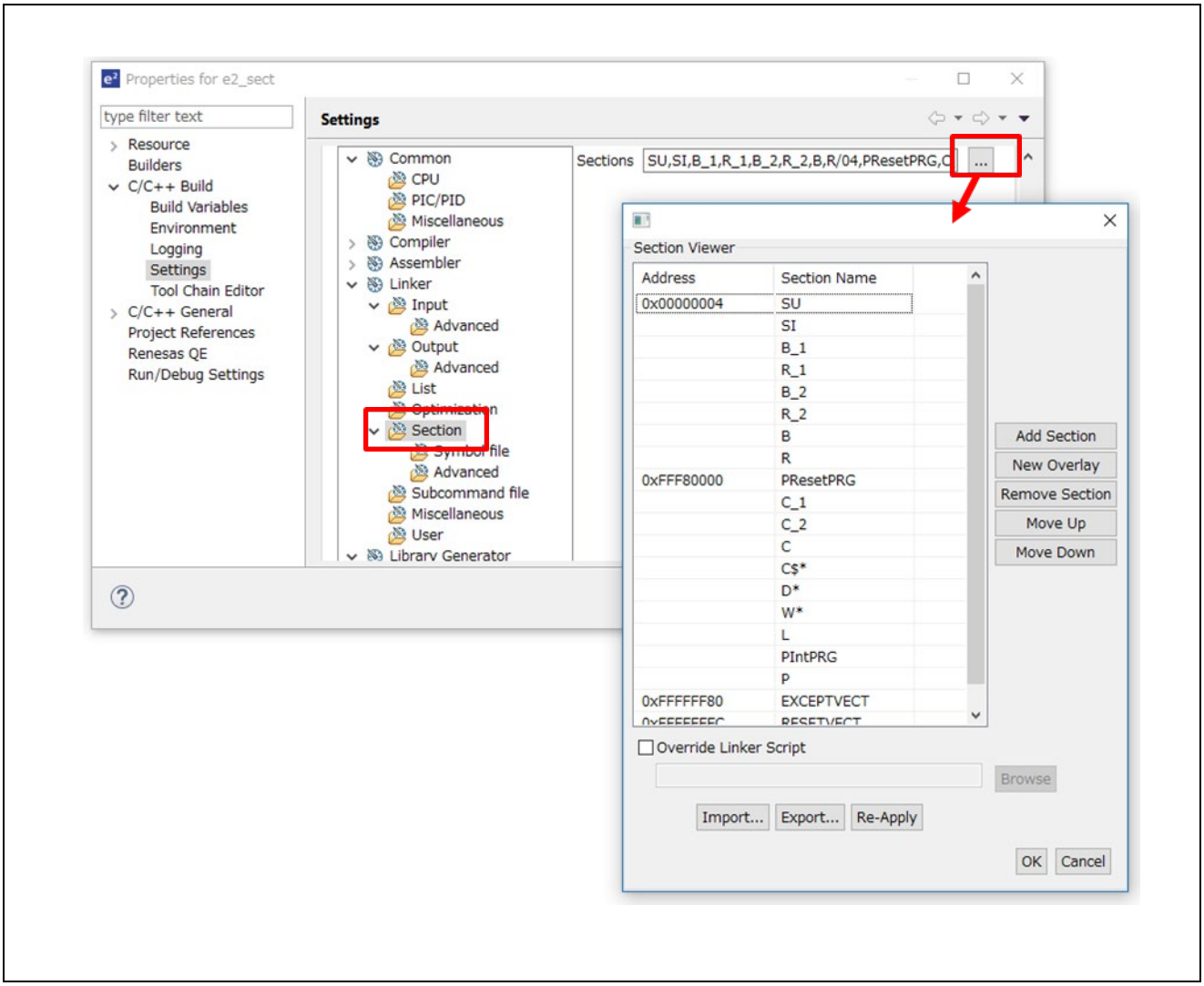


図 4.1 e² studio プロジェクトのセクション配置

OTA を使用しない場合のセクション配置設定を表 4.3 に記します。セクション指定が複雑化しないように、ワイルドカードを使用しています。

表 4.3 Bluetooth LE ライブラリセクションのセクション配置(OTA 未使用時)

アドレス	セクション名
0x00000004 (RAM)	SU
	SI
	B_1
	R_1
	B_2
	R_2
	B
	R
	BLE_B*
	BLE_R*
0xFFF80000 (ROM)	PResetPRG
	C_1
	C_2
	C
	C$*
	D*
	W*
	L
	PIntPRG
	P
	BLE_C*
	BLE_D*
	BLE_W*
	BLE_L
	BLE_P
0xFFFFFF80	EXCEPTVECT
0xFFFFFFFC	RESETVECT

また、ROM から RAM へマップするセクションに表 4.2 の Bluetooth LE Protocol Stack ライブラリのセクションを追加します。この設定はワイルドカードの使用が不可となります。割り当てが存在するセクションのみ追加する必要があります。

e² studio では、”Linker”→”Section”→”Symbol file”で設定します。

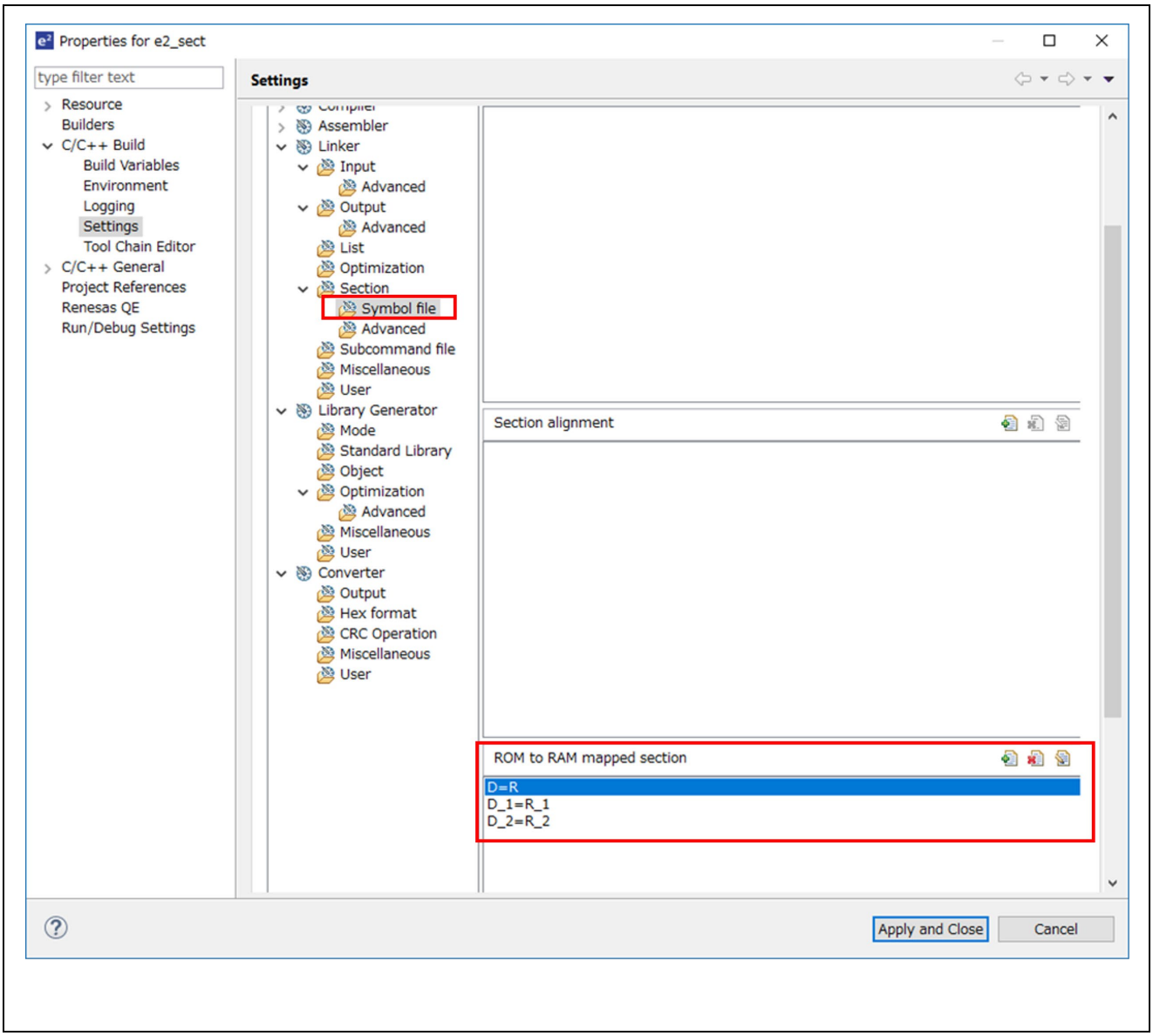


図 4.2　e² studio プロジェクトの ROM から RAM へマップするセクション設定

Bluetooth LE Protocol Stack ライブラリの BLE_D/BLE_R セクションを下記のように追加します。

```
D=R
D_1=R_1
D_2=R_2
BLE_D=BLE_R
BLE_D_1=BLE_R_1
BLE_D_2=BLE_R_2
```

【注】　該当するセクションへの割当が存在しない場合、リンクエラーとなります。”BLE_D_8” セクションは使用しないため、指定しません。

4.3 Bluetooth デバイスアドレス

Bluetooth LE Protocol Stack が使用する Bluetooth デバイスアドレスについては、「5.4 デバイス固有データ」を参照してください。

4.4 Bluetooth デバイス名

Bluetooth デバイス名は通知方法に応じて、以下の設定が必要となります。

4.4.1 Advertising による通知

表 4.4 の R_BLE API により Advertising Data または、Scan Response Data 内に Bluetooth デバイス名を含めた Advertising を行うことで Scanner デバイスに Bluetooth デバイス名を通知することが可能です。

表 4.4 Advertising Data を設定する API

R_BLE API	パラメータ構造体	Advertising Data or Scan Response Data を設定するフィールド
R_BLE_GAP_SetAdvSresData()	st_ble_gap_adv_data_t	p_data
R_BLE_ABS_StartLegacyAdv()	st_ble_abs_legacy_adv_param_t	p_adv_data
		p_sres_data
R_BLE_ABS_StartExtAdv()	st_ble_abs_ext_adv_param_t	p_adv_data
R_BLE_ABS_StartNonConnAdv()	st_ble_abs_non_conn_adv_param_t	p_adv_data

図 4.3 のフォーマットのデータを Advertising Data、または、Scan Response Data に設定してください。

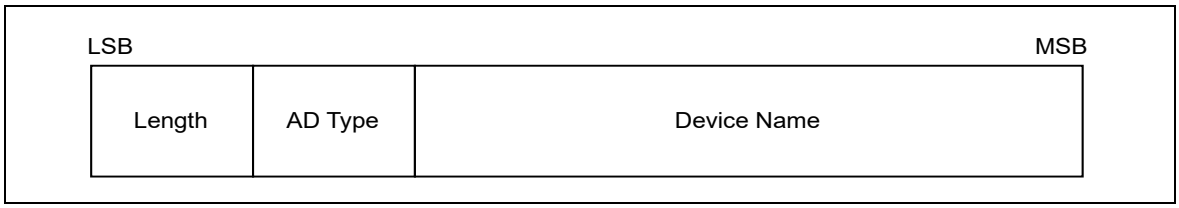


図 4.3 Advertising における Bluetooth デバイス名のフォーマット

Bluetooth デバイス名のフォーマット内の各フィールドについて表 4.5 に説明します。

表 4.5 Bluetooth デバイス名のフィールド

フィールド	説明
Length	“Device Name”と“AD Type”の合計のバイト数です。
AD Type *1	“Device Name”で設定した Bluetooth デバイス名のタイプを示します。 Bluetooth デバイスの一部を示す場合は”0x08"。 全体を示す場合、”0x09”。
Device Name	Bluetooth デバイス名を示す文字列です。

*1: その他の AD Type は Bluetooth SIG の”Assigned numbers”の”Generic Access Profile”を参照してください。

図 4.4 に R_BLE_GAP_SetAdvSresData()を使用した Bluetooth デバイス名の設定例について示します。

```
/* Advertising Data */
static uint8_t gs_adv_data[] =
{
    /* 省略 */
    /* Complete Local Name */
    11,   /* Length */
    0x09, /* AD Type */
    'B', 'L', 'E', '-', 'S', 'E', 'R', 'V', 'E', 'R', /* Device Name */
    /* 省略 */
};

/* Advertising Parameters */
static st_ble_gap_adv_data_t gs_adv_param =
{
    /* 省略 */
    .p_data      = gs_adv_data,
    /* 省略 */
};

/* GAP callback function */
static void ble_app_gapcb(uint16_t type, ble_status_t result,
st_ble_evt_data_t *p_data)
{
    switch (type)
    {
        /* 省略 */
        case BLE_GAP_EVENT_ADV_PARAM_SET_COMP :
        {
            R_BLE_GAP_SetAdvSresData (&gs_adv_param);
        } break;
        /* 省略 */
}
```

図 4.4　R_BLE_GAP_SetAdvSresData()を使用した Bluetooth デバイス名の設定例

4.4.2 Device Name Characteristic による通知

GAP Service の Device Name Characteristic を設定することで、GATT Client から Device Name Characteristic のリード要求受信時に、Bluetooth デバイス名を応答することが可能となります。

Device Name Characteristic の設定は QE for BLE から設定することが可能です。QE for BLE による設定の詳細については関連ドキュメント「Bluetooth Low Energy プロファイル開発者ガイド(R01AN6459) 3.2.3 キャラクタリスティックの追加と設定」を参照してください。

5. 独自機能

本章では BLE FIT モジュールが提供する独自機能について説明します。

独自機能には「3.2 app_lib」が提供する機能に加え、Bluetooth LE Protocol Stack が提供する機能があります。表 5.1 に Bluetooth LE Protocol Stack が提供する独自機能を示します。

表 5.1 Bluetooth LE Protocol Stack が提供する独自機能

機能	説明
RF 通信タイミング通知	RF 通信タイミング通知機能は、Bluetooth LE の RF 通信が行われる前後のタイミングをユーザアプリケーション層にコールバック通知する機能です。 詳細は「5.3 RF 通信タイミング通知機能」を参照してください。
デバイス固有データ	Bluetooth LE Protocol Stack が使用する Bluetooth デバイスアドレスをデバイス固有データとしてフラッシュメモリのユーザ領域(ROM)またはデータ領域(E2 データフラッシュ)に書き込むことができます。 詳細は「5.4 デバイス固有データ」を参照してください。

5.1 コマンドライン

コマンドラインは、VT100 エミュレーションに対応した端末エミュレータを通じて、任意のコマンド処理を実行する機能を提供します。VT100 エミュレーションに対応したエミュレータには、Tera Term などがあります。e² studio が提供するデバッグ・コンソールは VT100 エミュレーションに対応していません。コマンドラインが提供する機能を表 5.2 に示します。

表 5.2 コマンドラインが提供する機能

機能	説明
プロンプトの表示	プロンプトとして「$」を表示し、コマンドが実行できる状態であることを示します。
左右矢印キーによる編集位置の指定	左右の矢印キーにより、行内の文字を編集する位置を指定可能です。
バックスペースキー、デリートキーによる文字の編集	BS および DEL キーにより、行内の文字を削除します。
コマンドヘルプ	コマンドに続いて「help」を入力して実行すると、各コマンドのヘルプを表示します。
サブコマンドによる階層的なコマンド体系	階層的なコマンド体系を提供します。
複数のコマンドの登録	複数のコマンドを登録することが可能です。
コマンド実行の中断	Ctrl+C または Ctrl+D キーにより、実行中のコマンドを中断します。
コマンド補完	TAB キーにより、コマンド入力の補完を行います。

コマンドライン機能を使用する場合、コンフィギュレーションオプションの BLE_CFG_CMD_LINE_EN を 1 に設定してください。また、BLE_CFG_CMD_LINE_CH オプションに SCI のチャネルを指定します。

ボードと接続するコンピュータの端末エミュレータに以下の項目を設定します。

表 5.3 端末エミュレータの設定

項目	設定
New-line (Receive)	LF
New-line (Transmit)	CR
Terminal Mode	VT100
Baud rate	115200
Data	8bit
Parity	none
Stop bits	1bit
Flow Control	none

コマンドライン機能を使用して、BLE FIT モジュールが提供する Bluetooth LE 制御コマンドを以下に示します。

5.1.1 GAP コマンド

(1) Advertising コマンド

adv コマンド		
書式：	gap adv [adv_type] [operation] (addr_type) (-wl)	
	advertising の開始、停止、Advertising Set の削除を行います。	
パラメータ：	[adv_type]	Advertising のタイプとして、以下のいずれかを指定します。 legacy : legacy advertising を行います。 ext : extended advertising を行います。 non-conn : non-connectable advertising を行います。 periodic : periodic advertising を行います。
	[operation]	Advertising 開始／停止／Advertising Set の削除を選択します。 start : advertising を開始します。 stop : advertising を停止します。 remove : adv_type で指定した Advertising Set を削除します。
	(addr_type)	Advertising のアドレスのタイプを指定します。 省略するとスタティックアドレスになります。 pub : パブリックアドレス rnd : スタティックアドレス rpa_pub : gap priv set コマンド、またはアプリケーションでアイデンティティアドレスを登録している場合は RPA を使用します。上記以外の場合、パブリックアドレスとなります。 rpa_rnd : gap priv set コマンド、またはアプリケーションでアイデンティティアドレスを登録している場合は RPA を使用します。上記以外の場合、スタティックアドレスとなります。
	(wl)	ホワイトリストを使用する場合に指定します。 ホワイトリストを使用しない場合は省略可能です。
使用例：	gap adv legacy start legacy advertising を開始します。 ローカルデバイスのアドレスはスタティックアドレスとなります。 gap adv legacy start pub legacy advertising を開始します。 ローカルデバイスのアドレスはパブリックアドレスとなります。 gap adv ext stop extended advertising を停止します。	

本コマンドから設定できないその他のAdvertisingに関するパラメータはapp_lib/cmd/r_ble_cmd_abs.cのgs_legacy_adv_param, gs_ext_adv_param, gs_non_conn_adv_param, gs_periodic_adv_paramのAdvertisingパラメータの変数に設定しています。これらの変数を変更することで、Advertisingパラメータの設定が変更されます。パラメータの詳細はAPIドキュメント(r_ble_api_spec.chm)のModules >> Application Library >> Abstraction API >> Structures >> st_ble_abs_legacy_adv_param_t、st_ble_abs_ext_adv_param_t、st_ble_abs_non_conn_adv_param_t、st_ble_abs_perd_adv_param_t を参照してください。

(2) Scan コマンド

scan コマンド		
書式：	gap scan (operation) (filter_ad_type) (filter_data) (addr_type) (-wl)	
	scan の操作を行います。 scan を開始する場合、(operation)の指定は不要です。 scan を停止する場合、[ctrl] + [c]を入力してください。	
パラメータ：	(operation)	scan の操作として、以下を指定します。 stop : scan を停止します。
	(filter_ad_type)	scan 開始時に使用します。フィルターする AD type を指定します。 AD type の定義については"Assigned numbers"の"Generic Access Profile"を参照してください。フィルターしない場合、省略可能です。
	(filter_data)	scan 開始時に使用します。フィルターする Data を指定します。 filter_ad_type で指定した AD type のデータを指定します。 フィルターしない場合、省略可能です。filter_ad_type を使用しない場合、このパラメータは使用できません。
	(addr_type)	Scan request のアドレスのタイプを指定します。 省略するとスタティックアドレスになります。 pub : パブリックアドレス rnd : スタティックアドレス rpa_pub : gap priv set コマンド、またはアプリケーションでアイデンティティアドレスを登録している場合は RPA を使用します。 上記以外の場合、パブリックアドレスとなります。 rpa_rnd : gap priv set コマンド、またはアプリケーションでアイデンティティアドレスを登録している場合は RPA を使用します。 上記以外の場合、スタティックアドレスとなります。
	(wl)	ホワイトリストを使用する場合に指定します。 ホワイトリストを使用しない場合は省略可能です。
使用例：	gap scan スキャンを開始します。 gap scan 2 0x01,0x29 AD Type : Incomplete List of 16-bit Service Class UUIDs(0x02)かつ、 UUID : 0x2901 のサービスの advertising report を検索します。	

本コマンドから設定できないその他の scan に関するパラメータは app_lib/cmd/r_ble_cmd_abs.c の gs_phy_param_1m と gs_scan_param の Scan パラメータの変数に設定しています。これらの変数を変更することで、Scan パラメータの設定が変更されます。パラメータの詳細は API ドキュメント (r_ble_api_spec.chm)の Modules >> Application Library >> Abstraction API >> Structures >> st_ble_abs_scan_param_t、st_ble_abs_scan_phy_param_t を参照してください。

(3) 接続コマンド

conn コマンド		
書式：	gap conn [addr] [addr_type] (own_addr_type) (dummy_irk) (-wl)	
	接続要求を送信します。 接続要求をキャンセルしたい場合、[ctrl] + [c]を入力してください。	
パラメータ：	[addr]	リモートデバイスのアドレス(6 バイト)を指定します。 ビックエンディアンで 1 バイト毎にコロン(:)で区切って指定します。 例 ：74:90:50:00:95:a8
	[addr_type]	リモートデバイスの address type として、以下のいずれかを指定します。 pub： パブリックアドレス rnd： ランダムアドレス
	(own_addr_type)	接続要求のアドレスのタイプを指定します。 省略するとスタティックアドレスになります。 pub： パブリックアドレス rnd： スタティックアドレス rpa_pub : gap priv set コマンド、またはアプリケーションでアイデンティティアドレスを登録している場合は RPA を使用します。 上記以外の場合、パブリックアドレスとなります。 rpa_rnd : gap priv set コマンド、またはアプリケーションでアイデンティティアドレスを登録している場合は RPA を使用します。 上記以外の場合、スタティックアドレスとなります。
	(dummy_irk)	own_addr_type で RPA を指定した場合に Resolving List に登録するリモートデバイスの dummy の IRK を使用するかどうかを選択します。 dummy_irk： リモートデバイスの dummy の IRK を使用します。
	(wl)	ホワイトリストを使用する場合に指定します。 ホワイトリストを使用しない場合は省略可能です。
使用例：	gap conn 74:90:50:00:95:a8 pub 74:90:50:00:95:a8 のパブリックアドレスのリモートデバイスに接続要求を送信します。 ローカルデバイスはスタティックアドレスを使用します。 gap priv set net gap conn d8:19:e3:30:92:21 rnd rpa_rnd dummy_irk d8:19:e3:30:92:21 のランダムアドレスのリモートデバイスに接続要求を送信します。 ローカルデバイスは RPA を使用します。リモートデバイスの IRK として、dummy の IRK を使用します。 【注】ローカルデバイスが RPA を使用する場合、gap priv コマンドを事前に実行する必要があります。	

本コマンドから設定できないその他の接続に関するパラメータは app_lib/cmd/r_ble_cmd_abs.c の gs_conn_phy_1m と gs_conn_param の接続パラメータの変数に設定しています。これらの変数を変更することで、接続パラメータの設定が変更されます。パラメータの詳細は API ドキュメント(r_ble_api_spec.chm) の Modules >> Application Library >> Abstraction API >> Structures >> st_ble_abs_conn_phy_param_t、st_ble_abs_conn_param_t を参照してください。

(4) 切断コマンド

disconn コマンド		
書式：	gap disconn [conn_hdl]	
	接続を切断します。	
パラメータ：	[conn_hdl]	切断する接続の connection handle を指定します。
使用例：	gap disconn 0x0020 connection handle : 0x0020 の接続を切断します。	

(5) デバイスコマンド

device コマンド	
書式：	gap device
	接続中のデバイスのアドレスを表示します。
パラメータ：	なし
使用例：	gap device 接続中のデバイスのアドレスを表示します。

(6) プライバシーコマンド

<table>
<tr><th colspan="3">privコマンド</th></tr>
<tr><td rowspan="2">書式 :</td><td colspan="2">gap priv [operation] {params, ...}
gap priv set (IRK) [priv_mode]
gap priv remove [addr] [addr_type]
gap priv get lrpa ([addr] [addr_type])
gap priv off</td></tr>
<tr><td colspan="2">ローカルデバイスのプライバシーの操作を行います。</td></tr>
<tr><td rowspan="7">パラメータ :</td><td>[operation]</td><td>プライバシーの操作として、以下のいずれかを指定します。
set : ローカルデバイスの IRK を Resolving list に登録し、アドレス生成機能を ON にします。Advertising コマンド、接続コマンドでローカルデバイスが RPA を使用する場合に使用します。
remove : Resolving list に登録したリモートデバイスを削除します。
get : Resolving list に登録したリモートデバイスに送信するローカルデバイスの RPA を取得します。
off : アドレス生成機能を OFF にします。</td></tr>
<tr><td rowspan="6">{params, ...}</td><td>[operation] : set 指定時</td></tr>
<tr><td>(IRK) : resolving list に登録するローカルデバイスの IRK を指定します。乱数生成機能で IRK を生成する場合、このパラメータを省略します。
[priv_mode] :プライバシーモードとローカルデバイスのアイデンティティアドレスのタイプとして、以下のいずれかを指定します。
net : network プライバシーモードを設定します。
アイデンティティアドレスはスタティックアドレスを使用します。
dev : device プライバシーモードを設定します。
アイデンティティアドレスはスタティックアドレスを使用します。
net_pub : network プライバシーモードを設定します。
アイデンティティアドレスはパブリックアドレスを使用します。
dev_pub : device プライバシーモードを設定します。
アイデンティティアドレスはパブリックアドレスを使用します。
net_rnd : network プライバシーモードを設定します。
アイデンティティアドレスはスタティックアドレスを使用します。
dev_rnd : device プライバシーモードを設定します。
アイデンティティアドレスはスタティックアドレスを使用します。</td></tr>
<tr><td>[operation] : remove 指定時</td></tr>
<tr><td>[addr] : Resolving list に登録したリモートデバイスのアドレス(6 バイト)を指定します。
[addr_type] : Resolving list に登録したリモートデバイスのアドレスタイプを指定します。</td></tr>
<tr><td>[operation] : get 指定時</td></tr>
<tr><td>lrpa : Advertising 用のローカルデバイスの RPA を取得します。
[addr] : Resolving list に登録したリモートデバイスのアドレス(6 バイト)を指定します。
[addr_type] : Resolving list に登録したリモートデバイスのアドレスタイプを指定します。</td></tr>
</table>

使用例：	gap priv set 000102030405060708090a0b0c0d0e0f dev 0x0f0e0d0c0b0a09080706050403020100 の IRK を device プライバシーモードで設定します。 アイデンティティアドレスはスタティックアドレスを使用します。 gap priv set net network プライバシーモードを設定します。IRK は乱数生成で生成したものを使用します。 アイデンティティアドレスはスタティックアドレスを使用します。 gap priv set net_pub network プライバシーモードを設定します。IRK は乱数生成で生成したものを使用します。 アイデンティティアドレスはパブリックアドレスを使用します。 gap priv remove 12:34:56:78:9a:bc pub Resolving list に登録した 12:34:56:78:9a:bc(パブリック)のリモートデバイスを削除します。

(7) 接続コンフィグコマンド

conn_cfg コマンド		
書式 :	gap conn_cfg [operation] {params, …}	
	接続のコンフィグに関する操作を行います。	
パラメータ :	[operation]	接続コンフィグの操作として、以下のいずれかを指定します。 update : 接続パラメータのアップデートを行います。 phy : PHY の更新を行います。 def_phy : リモートデバイスからの PHY 更新要求を受け入れる内容を指定します。 data_len : データ送信時のパケットサイズ、送信時間を指定します。
	{params, ...}	[operation] : update 指定時 パラメータ 1 : connection handle を指定します。 パラメータ 2 : connection interval を指定します。範囲は 0x0006 – 0x0C80。 Time(ms) = パラメータ 2 x 1.25 パラメータ 3 : peripheral latency を指定します。範囲は 0x0000 – 0x01F3。 パラメータ 4 : supervision timeout を指定します。範囲は 0x000A – 0x0C80。 Time(ms) = パラメータ 4 x 10 パラメータ 2-4 は以下の条件を満たすように設定します。 パラメータ 4 x 10 >= (1 + パラメータ 3) x パラメータ 2 x 1.25
		[operation] : phy 指定時 パラメータ 1 : connection handle を指定します。 パラメータ 2 : 送信時の PHY を指定します。 パラメータ 2 は以下の値の論理和を設定します。 bit0 : 1M PHY bit1 : 2M PHY bit2 : Coded PHY パラメータ 3 : 受信時の PHY を指定します。 パラメータ 3 は以下の値の論理和を設定します。 bit0 : 1M PHY bit1 : 2M PHY bit2 : Coded PHY パラメータ 4 : Coded PHY の coding scheme を指定します。 パラメータ 4 以下のいずれかを設定します。 0x00 : コントローラの preferred value を設定します。 0x01 : S=2 Coding scheme 0x02 : S=8 Coding scheme
		[operation] : def_phy 指定時 パラメータ 1 : 変更を受け入れる送信時の PHY を指定します。 パラメータ 1 は以下の値の論理和を設定します。 bit0 : 1M PHY bit1 : 2M PHY bit2 : Coded PHY パラメータ 2 : 変更を受け入れる受信時の PHY を指定します。 パラメータ 2 は以下の値の論理和を設定します。 bit0 : 1M PHY bit1 : 2M PHY bit2 : Coded PHY
		[operation] : data_len 指定時 パラメータ 1 : connection handle を指定します。 パラメータ 2 : 変更する最大送信パケットサイズを指定します。 パラメータ 3 : 変更する最大送信時間を指定します。

使用例 :	gap conn_cfg update 0x0020 0x0100 0 0x0100 connection handle : 0x0020 の接続パラメータを connection interval : 0x0100 peripheral latency : 0 supervision timeout : 0x0100 に変更します。 gap conn_cfg phy 0x0020 2 2 0 connection handle : 0x0020 の接続 PHY を 送信 PHY : 2M 受信 PHY : 2M に変更します。 gap conn_cfg def_phy 7 7 送信 PHY : 1M, 2M, Coded PHY への変更をすべて受け入れる 受信 PHY : 1M, 2M, Coded PHY への変更をすべて受け入れる に変更します。 gap conn_cfg data_len 0x0020 0x00FB 0x4290 最大送信パケットサイズを 251 バイト、 最大パケット転送時間を 0x4290 us に変更します。

(8) ホワイトリストコマンド

wl コマンド		
書式 :	gap wl [operation] {params, …}	
	White List の操作を行います。	
パラメータ :	[operation]	White List の操作として、以下のいずれかを指定します。 reg : White List に{params, …}で指定したデバイスを登録します。 del : White List から{params, …}で指定したデバイスを削除します。 clear : White List をクリアします。
	{params, …}	[operation] : reg 指定時 パラメータ 1 : White List に登録するデバイスのアドレスを指定します。 パラメータ 2 : White List に登録するデバイスのアドレスタイプを指定します。 pub : パブリックアドレス rnd : ランダムアドレス
		[operation] : del 指定時 パラメータ 1 : White List から削除するデバイスのアドレスを指定します。 パラメータ 2 : White List から削除するデバイスのアドレスタイプを指定します。 pub : パブリックアドレス rnd : ランダムアドレス
		[operation] : clear 指定時 使用しません。
使用例 :	gap wl reg 74:90:50:00:95:a8 pub 74:90:50:00:95:a8 のパブリックアドレスのデバイスを White List に登録します。 gap wl del 74:90:50:00:95:a8 pub 74:90:50:00:95:a8 のパブリックアドレスのデバイスを White List から削除します。 gap wl clear White List をクリアします。	

(9) セキュリティコマンド

auth コマンド		
書式：	gap auth [operation] {params, …}	
	セキュリティの操作を行います。	
パラメータ：	[operation]	セキュリティの操作として、以下を指定します。 start : pairing or encryption を開始します。 passkey : passkey entry でのペアリング時に 10 進数の 6 桁の数値を入力します。 numcmp : numeric comparison でのペアリング時に表示されている数値が正しければ、"yes"、違っていれば、"no"を{params,…}に指定します。 del : ペアリングに関連する鍵を削除します。
	{params,…}	[operation] : start 指定時 パラメータ 1 : pairing or encryption を開始する接続を示す、connection handle を指定します。
		[operation] : passkey 時 パラメータ 1 : passkey entry で入力する、10 進数の 6 桁の数値を入力します。
		[operation] : numcmp 時 パラメータ 1 : numeric comparison の結果を"yes" or "no"で入力します。 両デバイスが同じ数値を示しているときは”yes”、それ以外は”no”を入力します。
		[operation] : del 時 パラメータ 1 : 鍵を削除する対象として、以下から指定します。 local : ローカルデバイスの鍵を削除します。 remote : リモートデバイスの鍵を削除します。 all : ローカルデバイスとリモートデバイスの鍵を削除します。 パラメータ 2: 鍵を削除するリモートデバイスを以下から指定します。 addr : パラメータ 3, 4 で指定したリモートアドレスの鍵を削除します。 all : すべてのリモートデバイスの鍵を削除します。 not_conn : 非接続状態のリモートデバイスの鍵を削除します。 パラメータ 3 : 鍵を削除するリモートデバイスのアドレスを指定します。 パラメータ 4 : 鍵を削除するリモートデバイスのアドレスタイプを指定します。 pub : パブリックアドレス rnd : ランダムアドレス

使用例 ：	gap auth start 0x0020 connection handle : 0x0020 の接続のペアリング、または、暗号化を開始します。 gap auth passkey 123456 Passkey Entry でのペアリング時に passkey として、123456 を入力します。 gap auth numcmp yes Numeric Comparison でのペアリング時にローカル、リモートデバイスが表示している数値が同じであることを回答します。 gap auth del remote all すべてのリモートデバイスの鍵を削除します。

(10) Periodic Advertising 同期コマンド

sync コマンド		
書式 :	gap sync [operation] {params…}	
	Periodic sync に関する操作を行います。	
パラメータ :	[operation]	sync の操作として、以下のいずれかを指定します。 create : {params...}でアドレスを指定したデバイスと periodic sync を確立します。 periodic sync が確立されるまで、scan を実施します。 periodic sync の確立をキャンセルしたい場合、[ctrl] + [c]を入力してください。 term : {params...}で指定した sync_hdl の periodic sync を終了します。
	{params,...}	[operation] : create 指定時 パラメータ 1 : periodic sync を確立する advertiser のアドレスを指定します。 パラメータ 2 : periodic sync を確立する advertiser のアドレスタイプを指定します。
		[operation] : term 指定時 パラメータ 1 : 終了する periodic sync の sync handle を指定します。
使用例 :	gap sync create 74:90:50:00:95:a8 pub 74:90:50:00:95:a8 のパブリックアドレスの advertiser に periodic sync を確立します。 gap sync term 0x01 sync handle : 0x01 の periodic sync を終了します。	

(11) バージョンコマンド

ver コマンド	
書式：	gap ver バージョン情報を取得します。 実行すると、Bluetooth LE Protocol Stack の以下のバージョン情報を表示します。 - Link Layer - HCI - Host Stack - 製造者 ID - BLE FIT モジュール - ライブラリのタイプ
パラメータ：	なし
使用例：	gap ver バージョン情報を取得します。 <u>結果例:</u> Link Layer / HCI Version HCI version : 0x09 *1 HCI revision : 0x000b Link Layer version : 0x09 *1 Link Layer subversion : 0x1908 Manufacturor ID : 0x0036 Host stack Version major version : 0x0d minor version : 0x19 subminor version : 0x08 BLE FIT module Version major | minor version : 0x00010000 *2 lib type : 0x00000001 *3

*1： Bluetooth SIG で規定されているバージョン(https://www.bluetooth.com/specifications/assigned-numbers)となります。0x09 は Bluetooth 5.0 を示します。

*2： 上位 2 バイトが major version、下位 2 バイトが minor version を示します。

*3： 0: All features, 1: Balance, 2: Compact となります。

5.1.2 Vendor Specific (VS)コマンド

(1) Tx Power 設定コマンド

txp コマンド		
書式：	vs txp [operation] [conn_hdl] [tx_power]	
	tx power の操作を行います。	
パラメータ：	[operation]	tx power の操作として、以下のいずれかを指定します。 set : tx power を設定します。 get : tx power を取得します。
	[conn_hdl]	接続中の tx power の設定／取得を行う場合は、 有効な connection handle を指定してください。 非接続状態の tx power の設定／取得を行う場合は、0xFFFF を 指定してください。
	{params, …}	set： 設定する tx power のレベルを指定します。 0 : High 1 : Middle 2 : Low
		get： 使用しません。
使用例：	vs txp set 0xFFFF 0 非接続状態の tx power を High に設定します。 vs txp get 0x0020 connection handle : 0x0020 の接続 tx power を取得します。	

(2) Coded スキーマ設定コマンド

scheme コマンド		
書式：	vs scheme [type]	
	Coded PHY の coding scheme を指定します。	
パラメータ：	[type]	coding scheme として、1 バイトの値を指定します。 各 bit が示す内容は以下の通りです。 デフォルトでは、S=8 の coding scheme が有効になっています。 bit0 : Primary Advertising の Coded PHY の Coding scheme (0:S=8/1:S=2) bit1 : Secondary Advertising の Coded PHY の Coding scheme (0:S=8/1:S=2) bit2： 接続時の Coded PHY の Coding scheme(0:S=8/1:S=2)
使用例：	vs scheme 7 Primacy Advertising, Secondary Advertising, 接続時の Coded PHY の coding scheme を S=2 に設定します。	

(3) 拡張 Direct Test Mode(DTM)コマンド

<table>
<tr><th colspan="3">test コマンド</th></tr>
<tr><td rowspan="2">書式 ：</td><td colspan="2">vs test [operation] {params, ...}</td></tr>
<tr><td colspan="2">DTM の操作を行います。</td></tr>
<tr><td rowspan="2">パラメータ ：</td><td>[operation]</td><td>DTM の操作として、以下を指定します。
tx : DTM の transmitter 側のテストを行います。
{params, ...}には
ch, data_len, payload, phy, tx_power, option, number of packet を指定してください。
rx : DTM の receiver 側のテストを行います。
{params, …}には、
ch, phy を指定してください。
end : DTM を終了します。
{params, …}で指定するパラメータはありません。</td></tr>
<tr><td>{params, …}</td><td>[operation] : tx 指定時
パラメータ 1 : 使用するチャネルを 0-39 の範囲で指定します。
周波数範囲は 0=2402MHz、39=2480MHz です。
パラメータ 2 : 送信するパケットサイズを 0-255 の範囲で指定します。
パラメータ 3 : payload のタイプを 0-7 までの範囲で指定します。
0x00 : PRBS9 '11111111100000111101..'.
0x01 : '11110000'
0x02 : '10101010'
0x03 : PRBS15
0x04 : '11111111'
0x05 : '00000000'
0x06 : '00001111'
0x07 : '01010101'

パラメータ 6 で”無変調”を指定した場合、このパラメータは無視されます。
パラメータ 4 : 使用する phy のタイプを 1-4 の範囲で指定します。
0x01 : 1M PHY
0x02 : 2M PHY
0x03 : Coded PHY (S=8)
0x04 : Coded PHY (S=2)

パラメータ 6 で”無変調”を指定した場合、このパラメータは無視されます。
パラメータ 6 で”変調”+”連続送信”を指定した場合、このパラメータに
0x03:Coded PHY(S=8)および 0x04:Coded PHY(S=2)を指定できません。
パラメータ 5 : tx_power のレベルを 0-2 の範囲で指定します。
0x00 : High
0x01 : Middle
0x02 : Low</td></tr>
</table>

		パラメータ 6 : option を 0-3 の範囲で指定します。 このパラメータは、次のビットの論理和で指定します。 bit0 : 0:変調、1:無変調 bit1 : 0:パケット送信、1:連続送信 パラメータ 7 : 送信するパケットの数を 0-0xFFFF の範囲で指定します。 パラメータ 6 で"連続送信"を指定した場合、このパラメータは無視されます。 このパラメータに 0 を指定した場合、テスト終了コマンドが実行されるまで継続して送信されます。
		[operation] : rx 指定時 パラメータ 1 : 使用するチャネルを 0-39 の範囲で指定します。 周波数範囲は 0=2402MHz、39=2480MHz です。 パラメータ 2 : 使用する phy のタイプを 1-3 の範囲で指定します。 0x01 : 1M PHY 0x02 : 2M PHY 0x03 : Coded PHY 受信では Coded PHY の coding scheme(S=8/S=2)は指定の必要がありません。
		[operation] : end 指定時 使用しません。
使用例 :	vs test tx 39 251 1 3 1 0 1 CH : 39ch パケットサイズ : 251 バイト payload : '11110000'の繰り返し phy : Coded PHY(S=8)を使用 tx_power : Middle option : 変調のパケット送信 num_of_packet : 1 個 で DTM 送信を行います。 vs test rx 39 2 CH : 39ch phy : 2M を使用 で DTM 受信を行います。 vs test end DTM を終了します。	

(4) BD アドレス操作コマンド

addr コマンド		
書式：	vs addr [operation] [area] {params,...}	
	ローカルデバイスのアドレスの操作を行います。	
パラメータ：	[operation]	ローカルデバイスのアドレスの操作として、以下のいずれかを指定します。 set : ローカルデバイスのアドレスを設定します。 {params,...}に address type, address を指定します。 [area]に"df"を指定した場合、リセット後に設定したアドレスが有効となります。 get : ローカルデバイスのアドレスを取得します。 {params,...}に address type を指定します。
	[area]	アドレスを格納している場所を指定します。 curr : 現在使用している一時的なアドレスの保管場所です。 df : Data Flash 内のアドレスを保管している場所です。
	{params,...}	[operation] : set 指定時 パラメータ 1 : 設定するアドレスのタイプを指定します。 pub : パブリックアドレス rnd : ランダムアドレス パラメータ 2 : 設定するアドレスを指定します。
		[operation] : get 指定時 パラメータ 1 : 取得するアドレスのタイプを指定します。 pub : パブリックアドレス rnd : ランダムアドレス
使用例：	vs addr set df pub 78:90:50:00:95:a8 Data Flash にパブリックアドレス、78:90:50:00:95:a8 を設定します。 設定を反映するには MCU のリセットが必要です。 vs addr get curr pub 現在使用しているパブリックアドレスを取得します。	

(5) 乱数生成コマンド

rand コマンド		
書式 :	vs rand [rand_size]	
	乱数を生成します。	
パラメータ :	[rand_size]	生成する乱数のサイズ(バイト)を指定します。 4-16 の範囲の値を指定してください。
使用例 :	vs rand 16 16 バイトの乱数を生成します。	

(6) Scan Channel 操作コマンド

scan_ch_map コマンド		
書式 :	vs scan_ch_map [operation] {params,...}	
	Scan Channel の操作を行います。	
パラメータ :	[operation]	Scan Channel の操作として、以下のいずれかを指定します。 set : Scan Channel を設定します。 {params,...}にて Channel Map を指定します。 get : Scan Channel を取得します。
	{params,...}	[operation] : set 指定時 パラメータ 1 : 設定する Channel Map を指定します。 論理和で指定することが可能です。 bit 0 : 37 ch bit 1 : 38 ch bit 2 : 39 ch 上記以外 : reserved
		[operation] : get 指定時 指定するパラメータはありません。
使用例 :	vs scan_ch_map set 7 Scan Channel として、37, 38, 39ch を設定します。 vs scan_ch_map get 現在使用している Scan Channel を取得します。	

5.1.3 SYS コマンド

(1) MCU ソフトウエアスタンバイコマンド

stby コマンド		
書式：	sys stby [operation]	
	ソフトウェアスタンバイモードの操作を行います。	
パラメータ：	[operation]	ソフトウェアスタンバイモードの操作として、以下のいずれかを指定します。 on： ソフトウェアスタンバイモードに移行します。 off： ソフトウェアスタンバイモードから復帰します。 get： 現在のソフトウェアスタンバイモードの状態を取得します。
使用例：	sys stby on ソフトウェアスタンバイモードに移行します。	

5.1.4 BLE コマンド

(1) Bluetooth LE Protocol Stack リセットコマンド

stby コマンド	
書式：	ble reset
	Bluetooth LE Protocol Stack をリセットします。
パラメータ：	なし
使用例：	ble reset

(2) Bluetooth LE Protocol Stack 終了コマンド

stby コマンド	
書式：	ble close
	Bluetooth LE Protocol Stack を終了します。 Bluetooth LE Protocol Stack を起動する場合は、ble reset コマンドにより行ってください。
パラメータ：	なし
使用例：	ble close

5.2 セキュリティデータ管理

セキュリティデータ管理機能はデータフラッシュ内で以下を管理します。

- ペアリング時に配布するローカルデバイスの鍵
- ペアリング時にリモートデバイスから取得した鍵とその情報

データフラッシュに保存されたローカルデバイスの鍵とリモートデバイスの鍵はセキュリティデータ管理 API により、Bluetooth LE Protocol Stack に再設定することが可能です。
なお、抽象 API では、セキュリティデータ管理 API を使用して、ローカルデバイスとリモートデバイスのセキュリティデータを管理しています。

セキュリティデータ管理機能は表 5.4 のコンフィギュレーションオプションにより設定を行います。

表 5.4 セキュリティデータ管理機能のコンフィギュレーションオプション

コンフィギュレーションオプション	説明
BLE_CFG_EN_SEC_DATA ※デフォルト値は"1"	セキュリティデータ管理機能を有効にします。この機能はペアリング成功時に"BLE_CFG_SECD_DATA_DF_BLOCK"で示す Block にボンディング情報を保存します。 0: Disable 1: Enable この機能を有効にする場合、Data Flash モジュールのコンポーネントを有効にしてください。
BLE_CFG_SECD_DATA_DF_BLOCK ※デフォルト値は"0"	セキュリティデータ管理機能で Data Flash へのボンディング情報の保存機能で使用する Data Flash Block を設定します。 "0"～"7"の範囲で設定してください。 BLE_CFG_DEV_DATA_DF_BLOCK で指定した Block と別の Block を指定してください。
BLE_CFG_NUM_BOND ※デフォルト値は"7"	保存するボンディング情報の数を設定します。 "1"～"7"の範囲で設定してください。

セキュリティデータ管理機能はセキュリティデータ管理情報とローカルデバイスのセキュリティデータとリモートデバイスのセキュリティデータを管理します。データフラッシュ内での配置は図 5.1 のようになります。

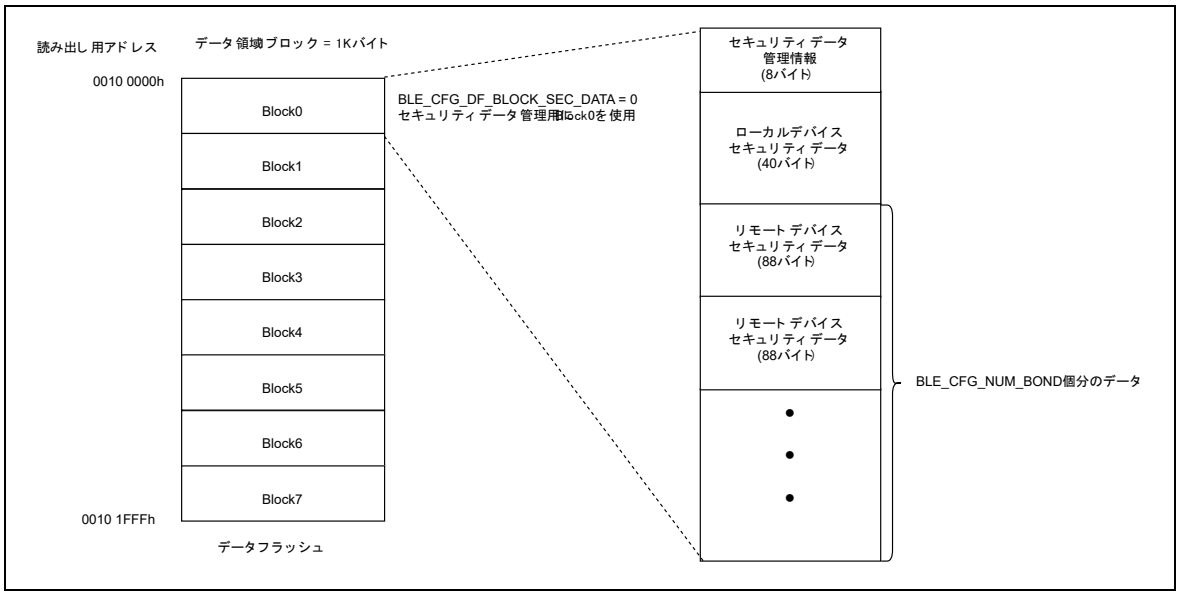


図 5.1 データフラッシュ内のセキュリティデータの配置

各データについて以下に説明します。

5.2.1 セキュリティデータ管理情報

セキュリティデータに関する情報を格納する領域です。セキュリティデータ管理情報の構成と構成要素を図 5.2、表 5.5 に示します。このデータはセキュリティデータ管理機能が内部的に処理するため、ユーザが更新する必要はありません。

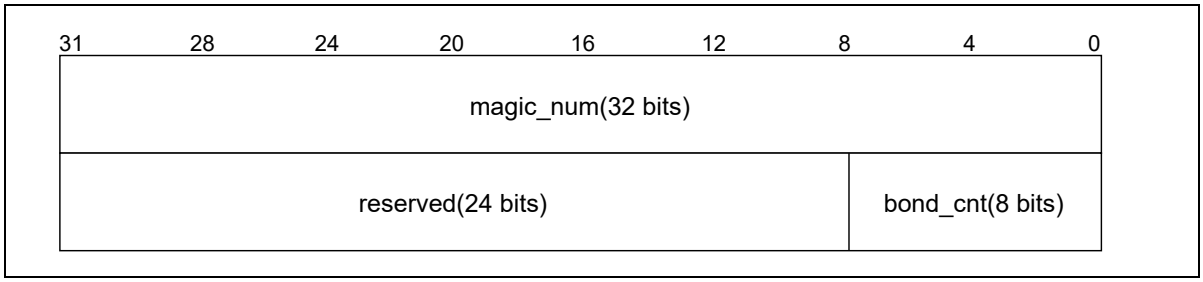


図 5.2 セキュリティデータ管理情報の構成

表 5.5 セキュリティ管理情報の構成要素

セキュリティデータ管理情報	Size (バイト)	説明
magic_num	4	セキュリティデータのマジックナンバー。 セキュリティデータの書き込み有無をチェック。 0x12345678 固定。未書き込み時は 0xFFFFFFFF。
bond_cnt	1	保管している bonding 情報の数
reserved	3	予約領域

5.2.2 ローカルデバイスのセキュリティデータ

ローカルデバイスのセキュリティデータの構成と構成要素を図 5.3、表 5.6 に示します。

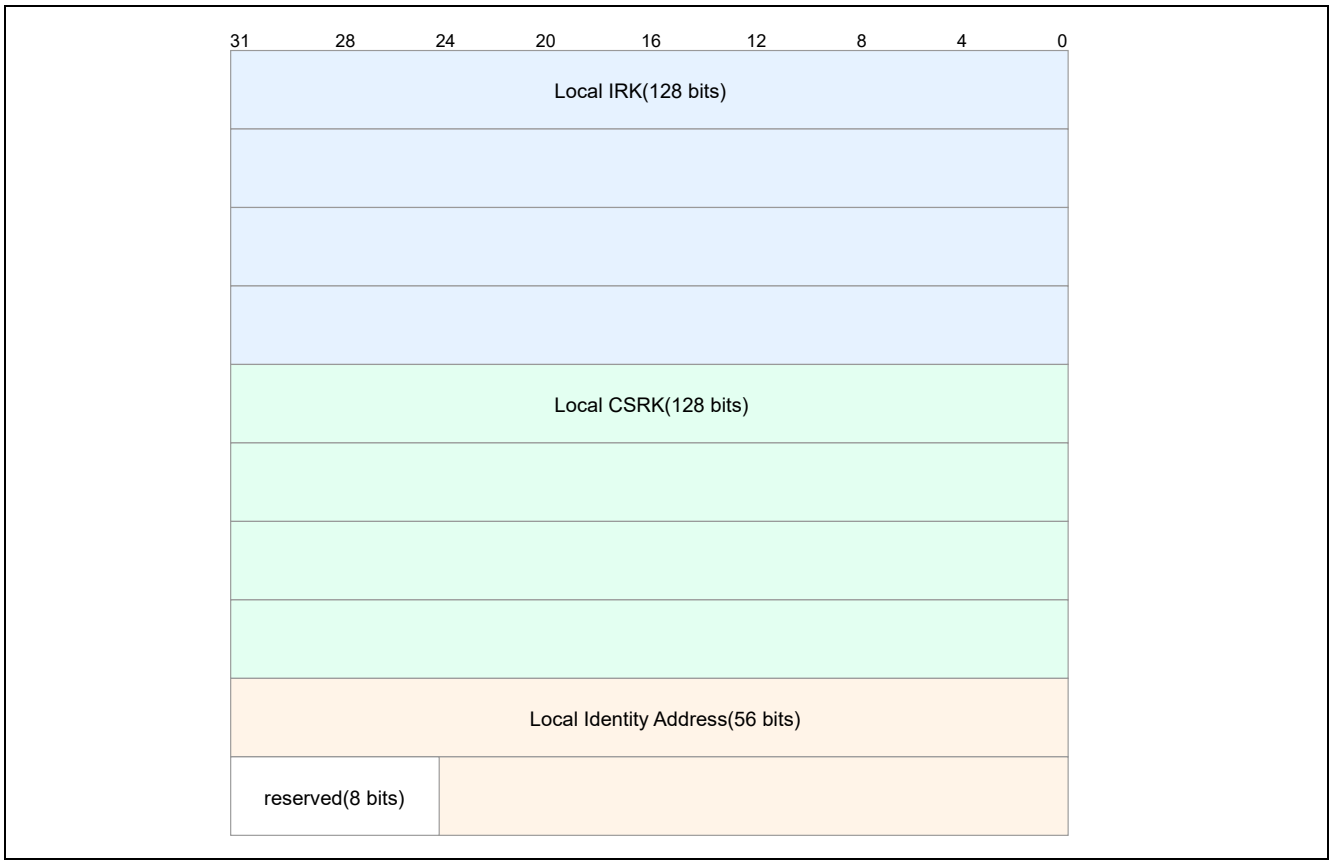


図5.3 ローカルデバイスのセキュリティデータの構成

表5.6 ローカルデバイスのセキュリティデータの構成要素

セキュリティデータ	サイズ (バイト)	説明
Local Identity Resolving Key (IRK)	16	ペアリング時にリモートデバイスに配布する IRK です。Privacy 機能により、Resolvable Private Address(RPA)を生成時に使用します。
Local Connection Signature Resolving Key (CSRK)	16	ペアリング時にリモートデバイスに配布する CSRK です。署名付きデータの送信時に使用します。
Local Identity Address	7	ペアリング時にリモートデバイスに通知するローカルデバイスのアイデンティティアドレスです。
reserved	1	予約領域

ローカルデバイスのセキュリティデータの設定について以下に記します。

- IRK と CSRK は 16 バイトの乱数を生成し、設定します。
- Bluetooth LE Protocol Stack への設定は、R_BLE_GAP_SetLocIdInfo() (IRK, アイデンティティアドレス) と R_BLE_GAP_SetLocCsrk()(CSRK)を使用します。
- データフラッシュへの書き込みは R_BLE_SECD_WriteLocInfo()により行います。
- データフラッシュからの読み出しは R_BLE_SECD_ReadLocInfo()により行います。
- データフラッシュからの削除は R_BLE_SECD_DelLocInfo()により行います。

セキュリティデータ管理機能の API を使用することで、生成したセキュリティデータをデータフラッシュに書き込み、デバイスの再起動時に Bluetooth LE Protocol Stack に再設定することが可能です。Bluetooth LE Protocol Stack 起動時などで実施する、ローカルデバイスのセキュリティデータの設定処理の例について図 5.4 に示します。

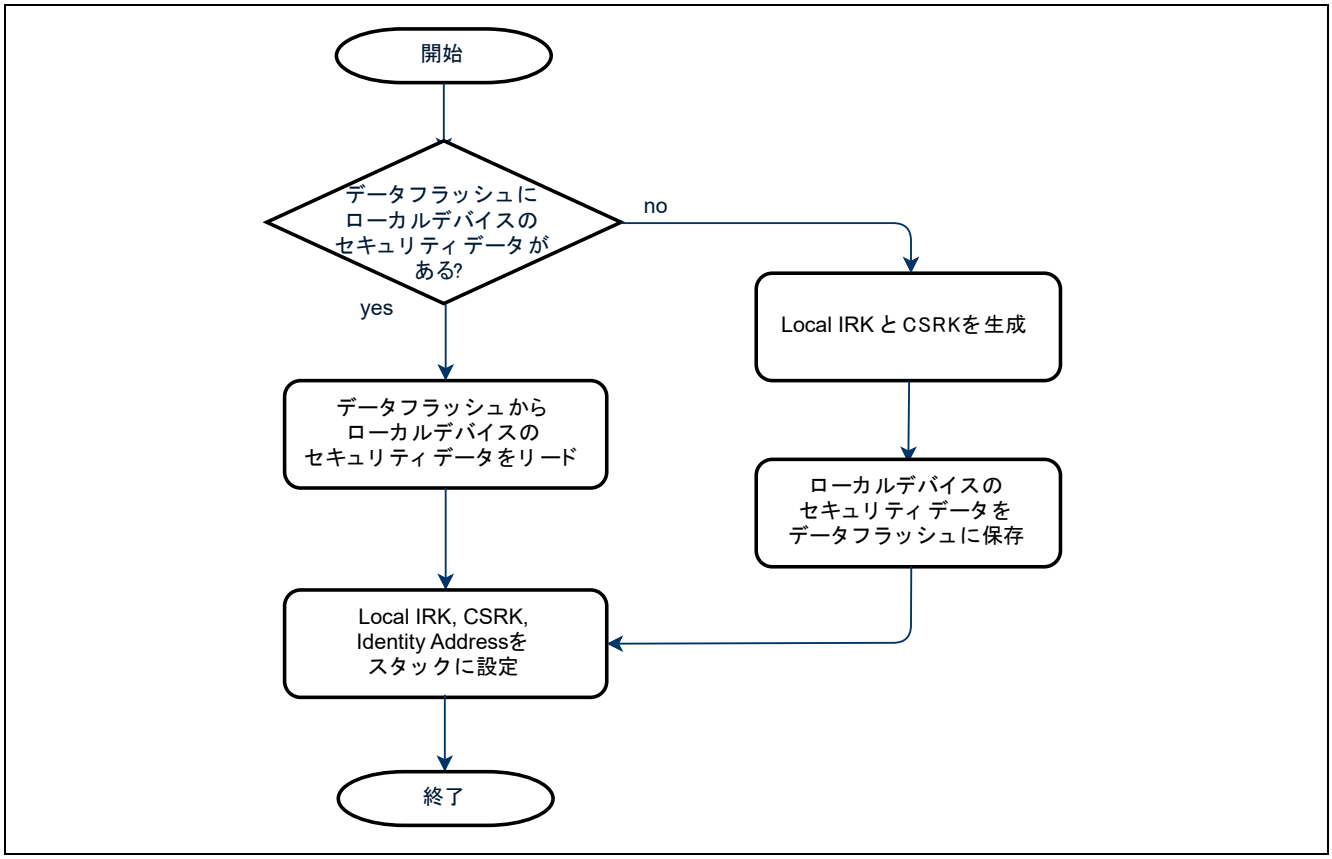


図5.4 ローカルデバイスのセキュリティデータの設定処理の例

5.2.3 リモートデバイスのセキュリティデータ

リモートデバイスのセキュリティデータの構成と構成要素を図 5.5、表 5.7 に示します。

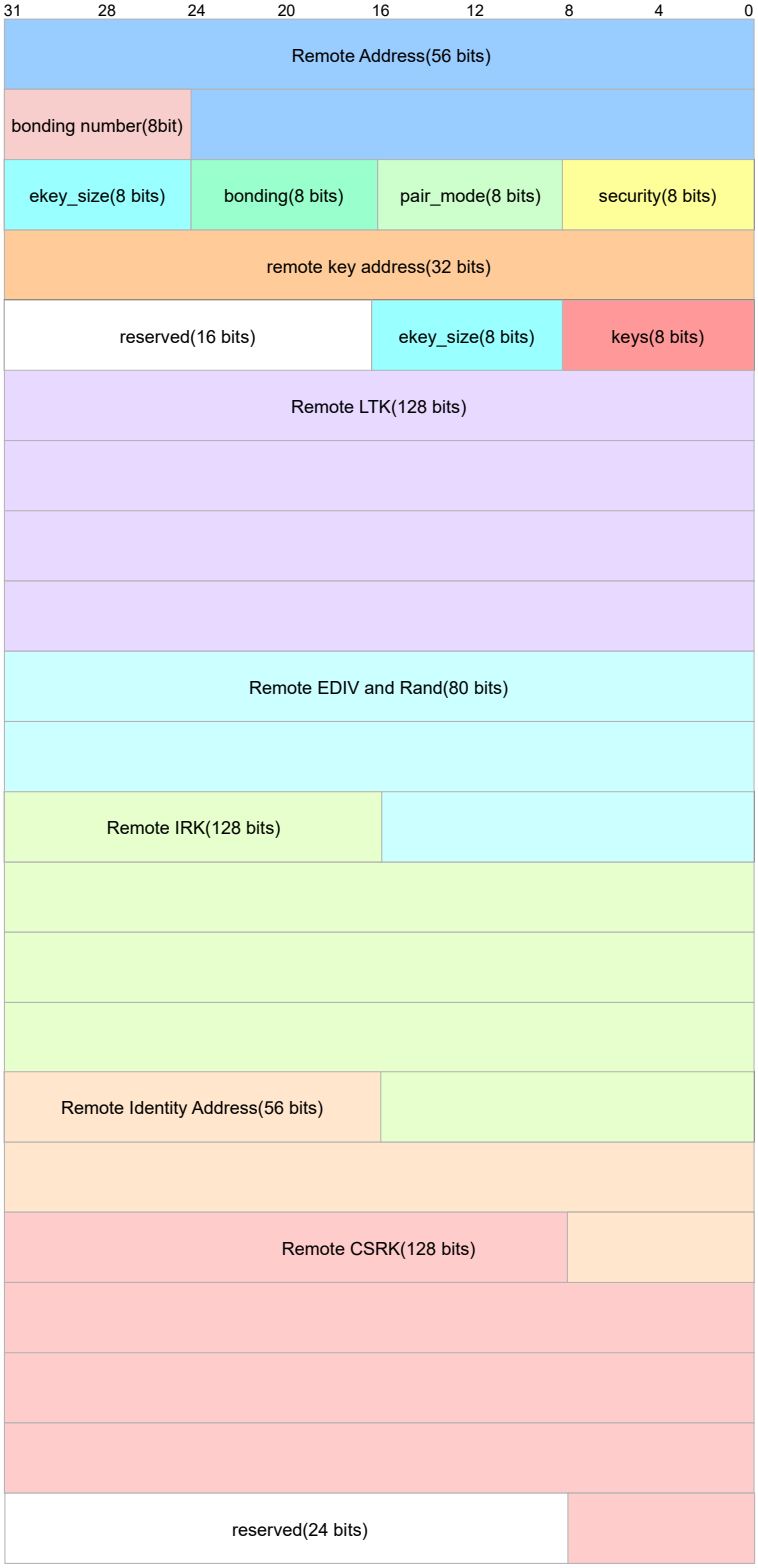


図5.5 リモートデバイスのセキュリティデータの構成

表 5.7 リモートデバイスのセキュリティデータの構成要素

セキュリティデータ	サイズ (バイト)	説明
Remote Address	7	ペアリング時にリモートデバイスが使用したアドレスです。
bonding number	1	Bonding の順番を示す値です。
security	1	実施したペアリングのセキュリティレベルです。 0x01 : Unauthenticated pairing でペアリングを実施。 0x02 : Authenticated pairing でペアリングを実施。
pair_mode	1	実施したペアリングのタイプです。 0x01 : Legacy pairing でペアリングを実施。 0x02 : Secure Connections でペアリングを実施。
bonding	1	リモートデバイスの bonding policy です。 0x00 : リモートデバイスが bonding を行わないことを示します。 0x01 : リモートデバイスが bonding を行うことを示します。
ekey_size	1	LTK のサイズです。
remote key address	4	リモートデバイスの鍵(Remote LTK から Remote CSRK)を保存する データフラッシュの開始アドレスです。
keys	1	リモートデバイスが配布した鍵の種類です。
Remote LTK	16	リモートデバイスが配布した LTK です。 接続の暗号化に使用します。
Remote EDIV and Rand	10	リモートデバイスが配布した EDIV と Rand です。 接続の暗号化に使用します。
Remote IRK	16	リモートデバイスが配布した IRK です。 リモートデバイスが privacy 機能を使用している場合、 アドレス解決に使用します。
Remote Identity Address	7	リモートデバイスのアイデンティティアドレスです。 リモートデバイスが privacy 機能を使用している場合、 アドレス解決に使用します。
Remote CSRK	16	リモートデバイスが配布した CSRK です。 署名付きデータの受信時に使用します。

リモートデバイスのセキュリティデータの設定について以下に記します。

- リモートデバイスのセキュリティデータはペアリング時に取得します。
- 表 5.7 の security, pair_mode, bonding, ekey_size は BLE_GAP_EVENT_PAIRING_COMP イベント時に R_BLE_SECD_WriteRemKeys ()により、その他のデータは、BLE_GAP_EVENT_PEER_KEY_INFO イベント時に R_BLE_SECD_WriteRemKeys ()により、データフラッシュに書き込みます。
- R_BLE_SECD_Init()はデータフラッシュからリモートデバイスのセキュリティデータをリードし、R_BLE_GAP_SetBondInfo ()をコールして、Bluetooth LE Protocol Stack へ設定します。
- データフラッシュからの削除は R_BLE_SECD_DelRemKeys ()により行います。
- BLE_CFG_NUM_BOND で指定した数を超えてデータフラッシュに書き込まれた場合、最も古いセキュリティデータのエントリに上書を行います。

セキュリティデータ管理機能の API を使用することで、取得したセキュリティデータをデータフラッシュに書き込み、デバイスの再起動時に Bluetooth LE Protocol Stack に再設定することが可能です。

5.3 RF 通信タイミング通知機能

RF 通信タイミング通知機能は、Bluetooth LE の RF 通信が行われる前後のタイミングをユーザアプリケーション層にコールバック通知する機能です。また、RF スリープモードの遷移前後でのコールバック通知も行えます。

RF 通信タイミング通知は実行する Bluetooth LE のイベント種別およびイベントの開始・完了それぞれのタイミング通知の有効・無効を選択できます。

【注】 本機能は Bluetooth LE Protocol Stack のソフトウェアタスク処理内でコールバックを行う機能のため、実際の RF 通信のタイミングからは遅延して通知されます。また、MCU の動作周波数、他周辺機能の割り込み、ユーザアプリケーションの処理内容などによって遅延時間は一定でない場合があります。

通知可能なイベント種別を表 5.8 に示します。

表 5.8 RF 通信タイミング通知のイベント種別

イベント種別	説明
Connection イベント開始・完了	Connection イベントの開始・完了のタイミングでコールバック通知します。 コンフィギュレーションオプション • BLE_CFG_EVENT_NOTIFY_CONN_START • BLE_CFG_EVENT_NOTIFY_CONN_CLOSE
Advertising イベント開始・完了	Advertising イベントの開始・完了のタイミングでコールバック通知します。 コンフィギュレーションオプション • BLE_CFG_EVENT_NOTIFY_ADV_START • BLE_CFG_EVENT_NOTIFY_ADV_CLOSE
Scan イベント開始・完了	Scan イベントの Window 開始・完了のタイミングでコールバック通知します。Scan の Interval が Window と一致する設定の場合にはコールバック通知は行いません。 コンフィギュレーションオプション • BLE_CFG_EVENT_NOTIFY_SCAN_START • BLE_CFG_EVENT_NOTIFY_SCAN_CLOSE
Initiator イベント開始・完了	Initiator 時の Scan イベントの Window 開始・完了のタイミングでコールバック通知します。Initiator 時の Scan の Interval が Window と一致する設定の場合にはコールバック通知は行いません。 コンフィギュレーションオプション • BLE_CFG_EVENT_NOTIFY_INIT_START • BLE_CFG_EVENT_NOTIFY_INIT_CLOSE
RF スリープモード開始・復帰	RF スリープモードの開始・復帰のタイミングでコールバック通知します。 コンフィギュレーションオプション • BLE_CFG_EVENT_NOTIFY_DS_START • BLE_CFG_EVENT_NOTIFY_DS_WAKEUP

5.3.1 Connection イベントの通知タイミング

Connection イベントは Connection Interval ごとの RF 通信タイミングでコールバックを行います。
Connection イベントの通知タイミングを図 5.6 に示します。

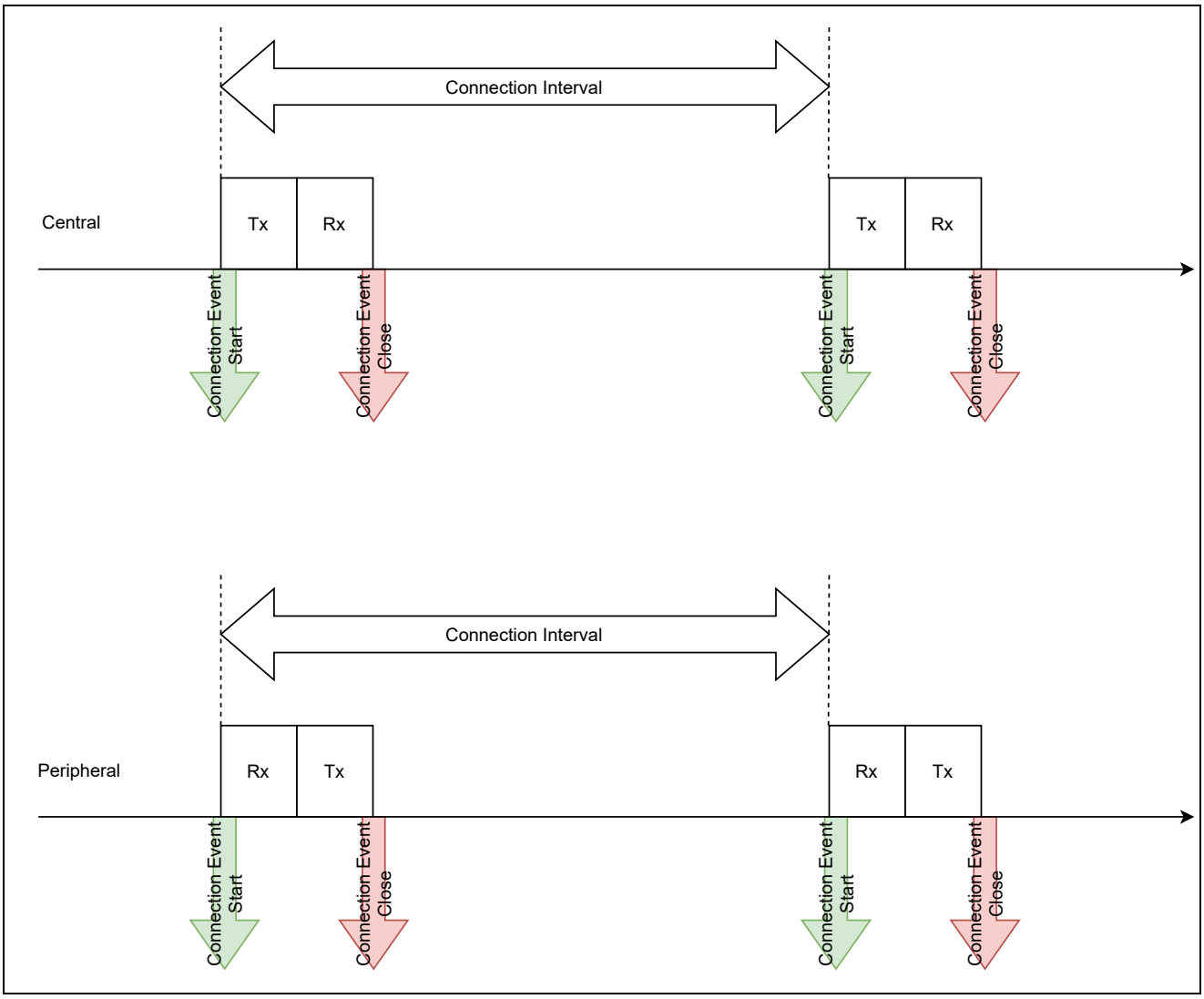


図 5.6 Connection イベントの通知タイミング

5.3.2 Advertising イベント通知タイミング

Advertising イベントは Advertising Interval ごとの RF 通信タイミングでコールバックを行います。
Advertising イベントの通知タイミングを図 5.7 に示します。

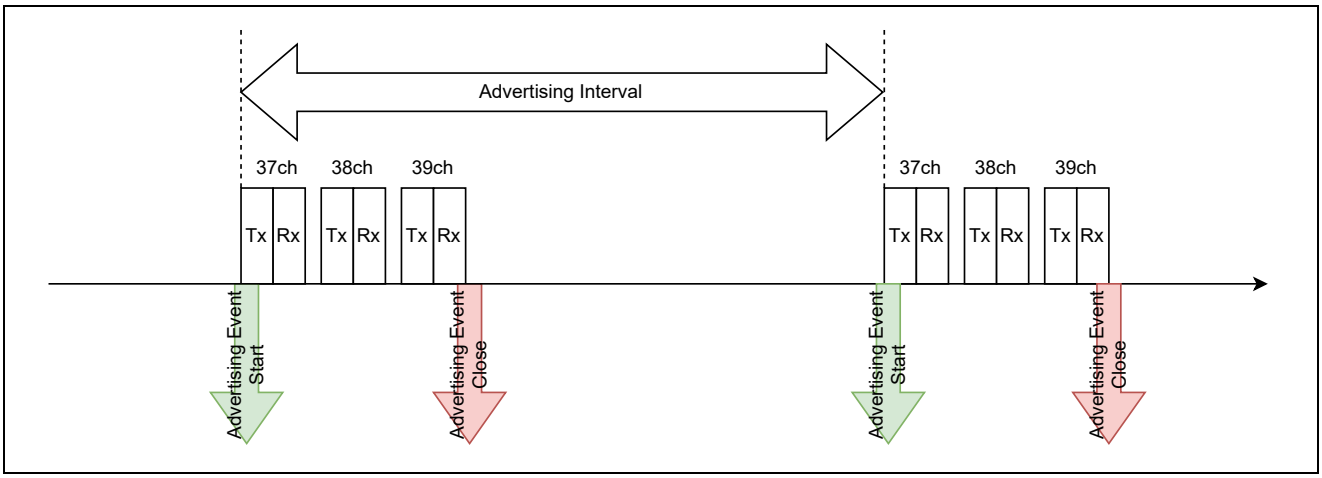


図 5.7 Advertising イベントの通知タイミング

5.3.3 Scan/Initiator イベント通知タイミング

Scan および Initiator イベントは Scan Interval ごとの RF 通信タイミングでコールバックを行います。
Scan/Initiator イベントの通知タイミングを図 5.8 に示します。
Scan Interval = Scan Window を設定した場合、各開始・完了の通知は行いません。

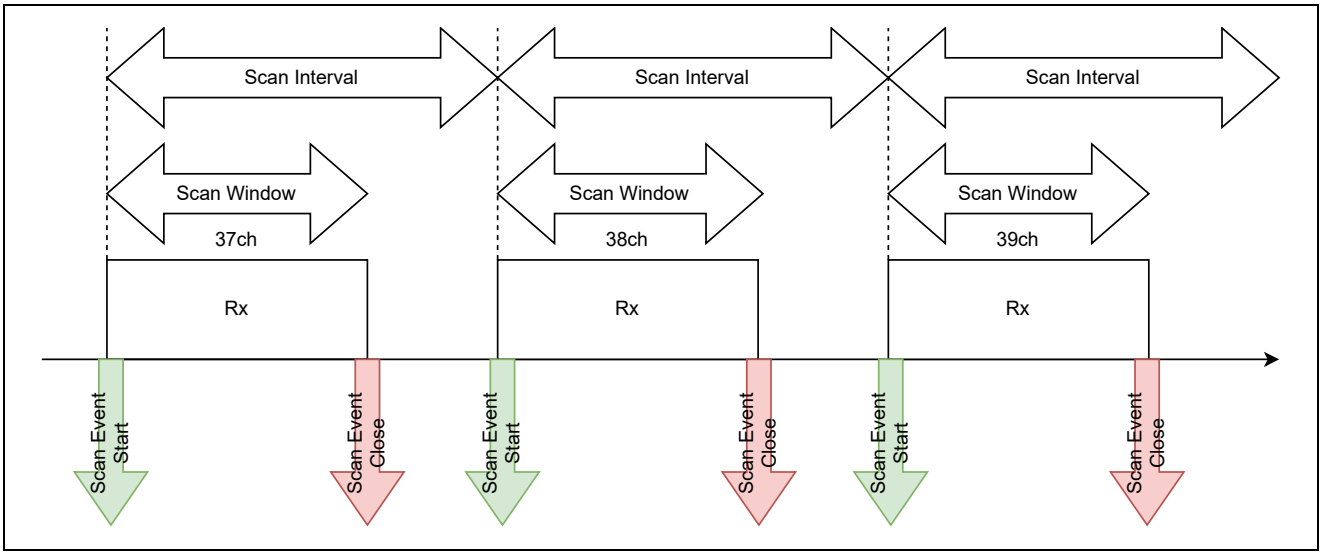


図 5.8 Scan/Initiating イベントの通知タイミング

5.3.4 RF スリープモードイベント通知タイミング

BLE_CFG_RF_DEEP_SLEEP_EN が 1 (Enable)に設定されており、かつ次の RF イベントまでの時間が一定時間以上ある場合に Bluetooth LE Protocol Stack のスケジューラ内で RF スリープモード遷移を行います。

RF スリープモードイベントの通知は、各種イベント間で RF スリープモードの状態遷移があった場合にコールバックします。

Advertising 実行中の RF スリープモードイベントの通知タイミングを図 5.9 に示します。

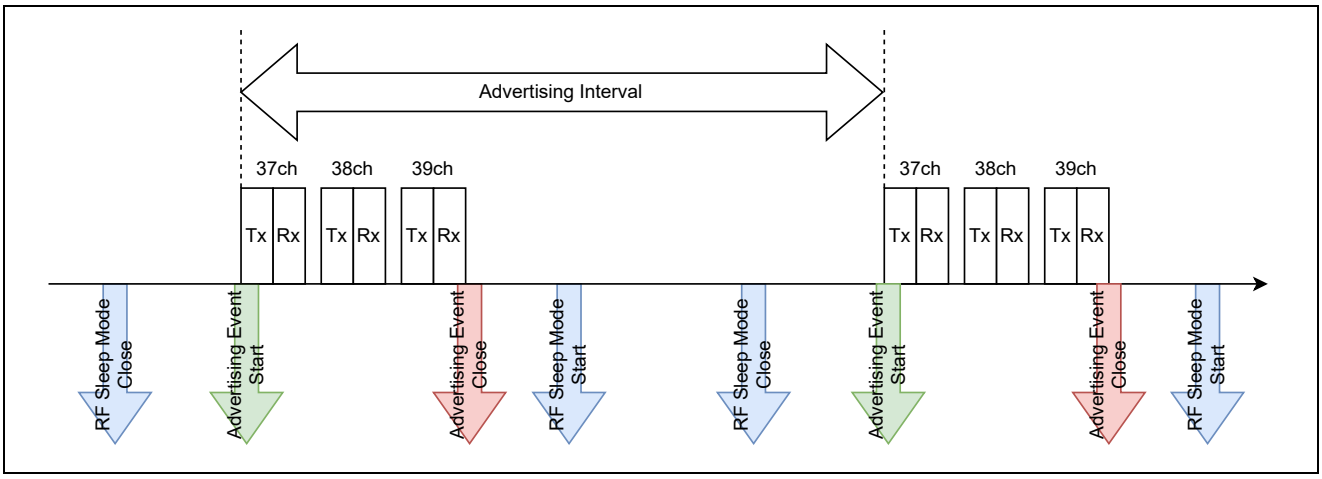


図 5.9 RF スリープモードイベントの通知タイミング(Advertising の場合)

5.3.5 RF 通信タイミング通知仕様

RF 通信タイミング通知は表 5.8 に示したコンフィギュレーションオプションで有効・無効を切り替えます。各オプションの概要は「4.1 コンフィギュレーションオプション」を参照してください。

タイミング通知のコールバック関数は r_ble_pf_functions.c に登録されています。必要に応じて各コールバック関数内に処理を追加してください。

【注】負荷の大きい処理を追加した場合、スループット性能低下などの影響が発生する場合があります。

RF 通信タイミング通知コールバック関数の概要を表 5.9 に示します。

表 5.9 RF 通信タイミング通知コールバック関数

コールバック関数名	概要
void r_ble_rf_notify_event_start(uint32_t param)	各 RF 通信イベント開始の通知コールバック関数
	引数 uint32_t param [31:16] イベント種別(表 5.10 参照) [15:0] イベント識別コード(表 5.11 参照)
void r_ble_rf_notify_event_close(uint32_t param)	各 RF 通信イベント完了の通知コールバック関数
	引数 uint32_t param [31:16] イベント種別(表 5.10 参照) [15:0] イベント識別コード(表 5.11 参照)
void r_ble_rf_notify_deep_sleep(uint32_t param)	RF スリープモードの状態変更通知コールバック関数
	引数 uint32_t param 0x1：RF スリープモード完了(RF スリープモードから復帰) 0x0：RF スリープモード開始(RF スリープモードに遷移)

表 5.10 RF 通信イベント種別 定義マクロ

イベント種別	マクロ名	値
Connection イベント	BLE_EVENT_TYPE_CONN	0x0000
Advertising イベント	BLE_EVENT_TYPE_ADV	0x0001
Scan イベント	BLE_EVENT_TYPE_SCAN	0x0002
Initiator イベント	BLE_EVENT_TYPE_INITIATOR	0x0003

表 5.11 RF 通信イベント識別コード

イベント種別	イベント識別コード
Connection イベント	該当する conn_hdl を設定します。
Advertising イベント	該当する adv_hdl を設定します。
Scan イベント	0x0000 を設定します。
Initiating イベント	0x0000 を設定します。

5.4 デバイス固有データ

Bluetooth LE Protocol Stack が使用する Bluetooth デバイスアドレス(以降は BD アドレスと記載)をデバイス固有データとしてフラッシュメモリのユーザ領域(ROM)またはデータ領域(E2 データフラッシュ)に書き込むことができます。これにより、同じファームウェアを使用して複数の RX23W デバイスに対して異なる BD アドレスを設定することができます。

デバイス固有データはファームウェアのプログラム領域とは異なる領域に配置され、ファームウェア書き換え時にデバイス固有データが消去されなければ継続して同じ BD アドレスを使用することができます。デバイス固有データが消去された場合は「5.4.6 BD アドレスの決定方法」に従い BD アドレスを決定します。

5.4.1 デバイス固有データの配置ブロック指定

デバイス固有データを配置するユーザ領域(ROM)およびデータ領域(E2 データフラッシュ)のブロック番号はコンフィギュレーションオプション（r_ble_rx23w_config.h）の BLE_CFG_DEV_DATA_CF_BLOCK および BLE_CFG_DEV_DATA_DF_BLOCK で指定できます。

ユーザ領域のブロック番号は、アドレス終端(0xFFFFF800)がブロック 0、アドレス先頭(0xFFF80000)がブロック 255 となります。データ領域のブロック番号は、アドレス先頭(0x00100000)がブロック 0、アドレス終端(0x00101C00)がブロック 7 となります。

図 5.10 に RX23W のブロック構成を示します。

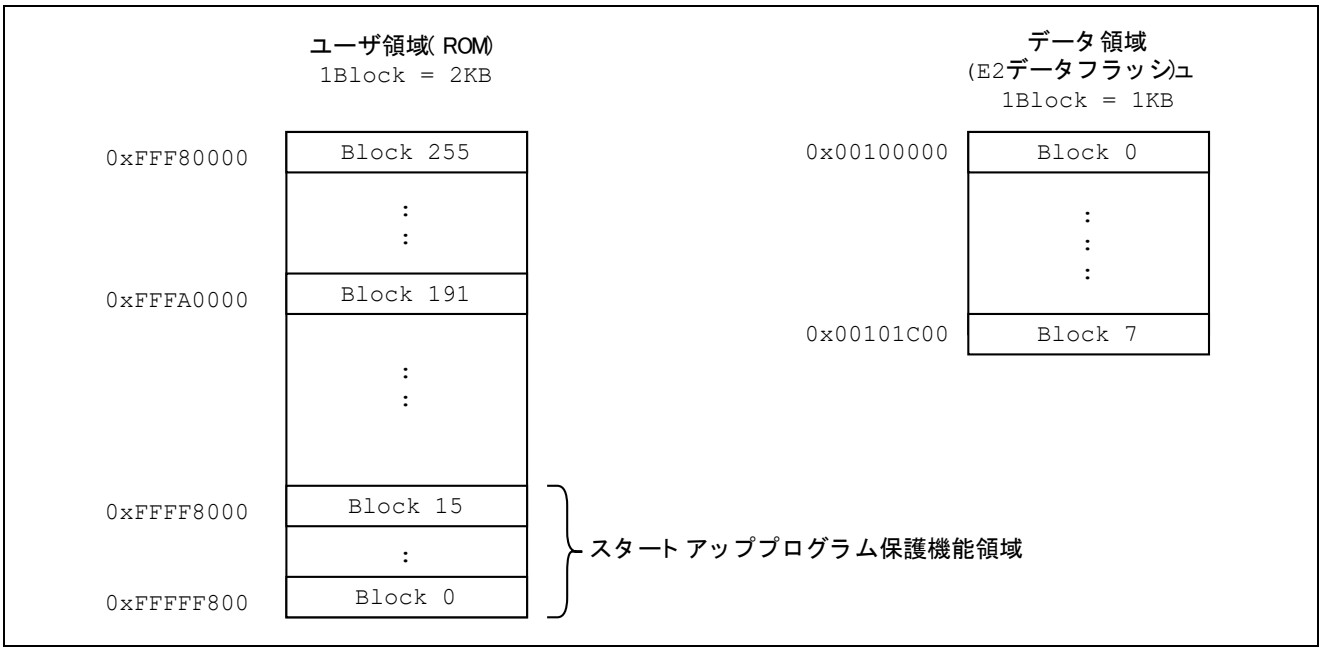


図 5.10 RX23W のフラッシュメモリブロック構成

ユーザ領域にデバイス固有データを配置する場合、プログラムコードでは使用しないブロックを指定する必要があります。また、指定したブロックの先頭アドレスにデバイス固有データを書き込む必要があります。

RX23W のスタートアッププログラム保護機能を使用する場合、ユーザ領域のブロック 0～15 にはデバイス固有データを配置しないようにしてください。

データ領域にデバイス固有データを配置する場合、指定したブロックの先頭アドレスにデバイス固有データを書き込みます。デバイス固有データを配置したブロックには他のデータを書き込まないようにしてください。

5.4.2 デバイス固有データのフォーマット

デバイス固有データのフォーマットを表 5.12 に示します。

表 5.12 デバイス固有データのフォーマット

オフセット	サイズ[byte]	型	概要
0	4	uint32_t	マジックナンバー以降のデータ長(0x00000010 固定)
4	4	uint32_t	マジックナンバー(0x12345678 固定)
8	6	uint8_t [6]	パブリック BD アドレス
14	6	uint8_t [6]	ランダム BD アドレス

データはブロックごとにリトルエンディアンで書き込む必要があります。

例として、BD アドレスが“01:02:03:04:05:06”の場合、フラッシュメモリには 0x06,0x05,0x04,0x03,0x02,0x01 の順に書き込みます。

図 5.11 にデバイス固有データのフラッシュメモリ配置例を示します。

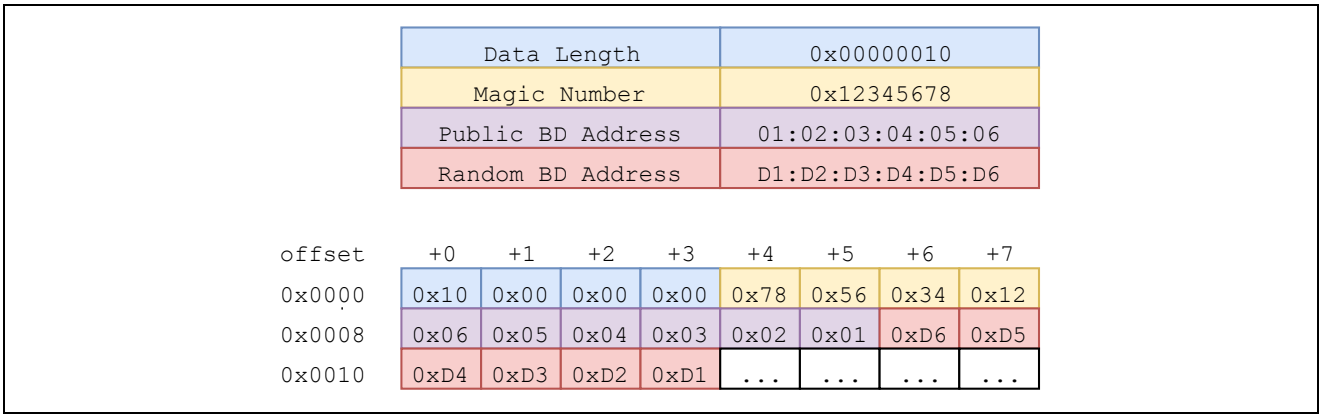


図 5.11 デバイス固有データのフラッシュメモリ領域配置例

5.4.3 ユーザ領域(ROM)への書き込み

ユーザ領域(ROM)への書き込みは Renesas Flash Programmer(RFP)のユニークコード機能を使用してファームウェアと共にユーザ領域へ書き込みます。

RFP を使用したデバイス固有データの書き込み概要を図 5.12 に示します。

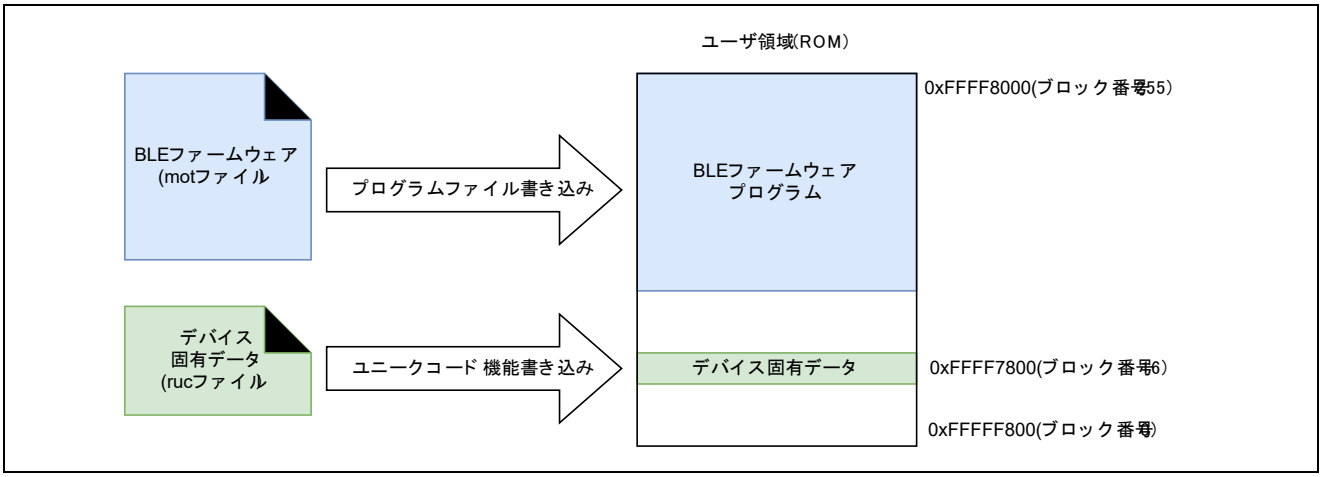


図 5.12 RFP を使用したデバイス固有データ書き込み概要

図 5.13 にデバイス固有データの RFP ユニークコードファイル(ruc)の設定例を示します。

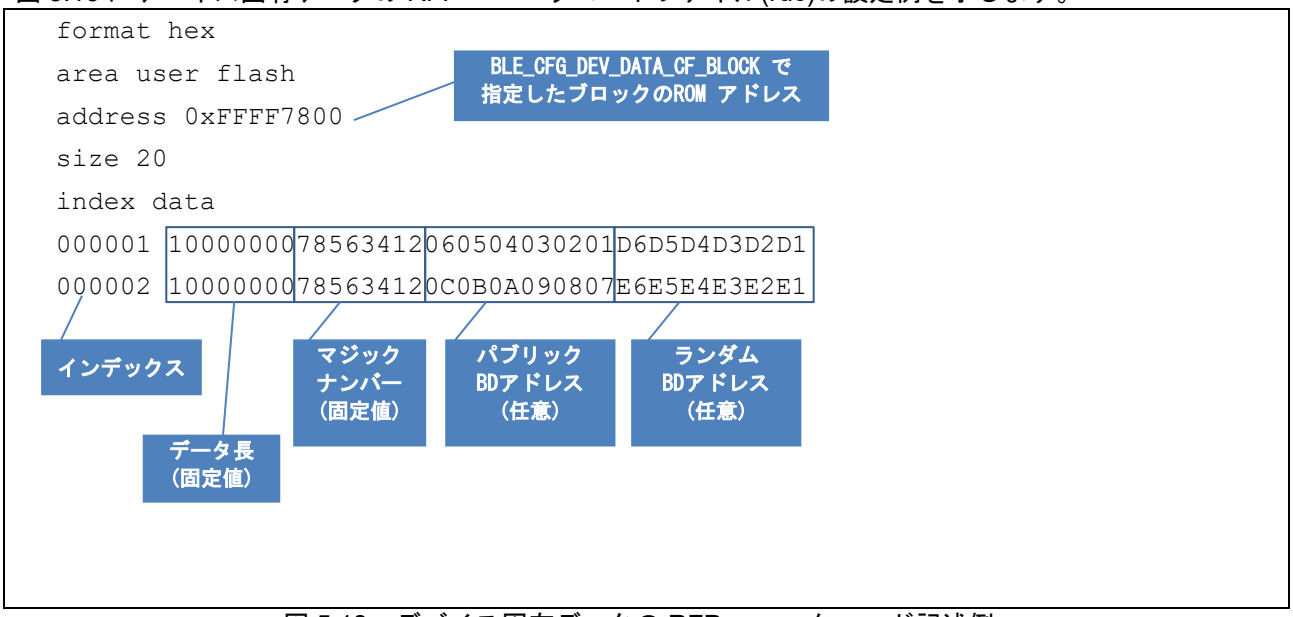


図 5.13 デバイス固有データの RFP ユニークコード記述例

5.4.4 データ領域(E2 データフラッシュ)への書き込み

データ領域への書き込みは R_BLE API の R_BLE_VS_SetBdAddr()を使用して書き込みます。

HCI モードファームウェアの場合はパブリック BD アドレス書き込みツール(BDAddrWriter)を使用して書き込みます。

データ領域にデバイス固有データを書き込んだ場合、RX23W を一度リセットすることで書き込まれた BD アドレスを使用します。

5.4.4.1 R_BLE API による書き込み

R_BLE_VS_SetBdAddr() API を使用してデータ領域に書き込みを行うことが可能です。

API の詳細は R_BLE API ドキュメント(r_ble_api_spec.chm)を参照してください。

デモプロジェクトではコマンドラインから BD アドレス操作コマンドで BD アドレスの書き込みを行えます。

5.4.4.2 BDAddrWriter による書き込み

HCI モードのファームウェアが書き込まれた RX23W デバイスに対して、パブリック BD アドレス書き込みツール(BDAddrWriter)を使用することでデータ領域にパブリック BD アドレスを書き込むことができます。

【注】BDAddrWriter はランダム BD アドレスの書き込みには対応していません。

以下に BDAddrWriter を使用したパブリック BD アドレス書き込み手順を示します。

1. RX23W と接続している COM ポートを選択します。
2. UART のボーレートを設定します。リストにない場合は手動で入力してください。
3. 書き込みたいパブリック BD アドレスをテキストボックスに入力します。BD アドレスの入力は 16 進文字列(0-9, a-f, A-Z)を 12 桁(6 バイト分)入力してください。16 進文字列以外は書き込み実行時に破棄します。
4. [Write]ボタンをクリックします。
5. 書き込みが成功した場合、”Success!!” のメッセージが表示されます。RX23W をリセットすることで書き込んだパブリック BD アドレスが有効になります。

   書き込みに失敗した場合は以下を確認してください。
   - RX23W デバイスに HCI モードのファームウェアが書き込まれているか
   - RX23W デバイスと接続している COM ポートを選択しているか
   - HCI モードの UART ボーレートは一致しているか

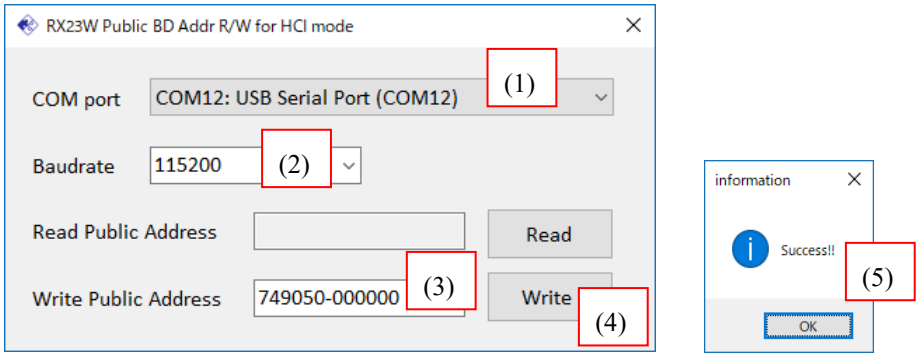


図 5.14 BD アドレス書き込み手順

5.4.5 RX23W のフラッシュメモリプロテクト機能

RX23W にはフラッシュメモリプロテクト機能があります。フラッシュメモリプロテクト機能は、第三者によるフラッシュメモリの読み出し、書き換えから保護する機能です。

フラッシュメモリプロテクト機能の詳細は「RX23W グループ ユーザーズマニュアル ハードウェア編 (R01UH0823) 50.9 フラッシュメモリプロテクト機能」を参照してください。

RFP のようなシリアルプログラマを使用してデバイス固有データを消去せずに別のファームウェアを書き込む場合には、RX23W のブートモード ID コードプロテクトを有効にし、フラッシュメモリ消去のブロック選択を行えるようにする必要があります。

ブートモード ID コードプロテクトを有効にするには、ID コードプロテクトの制御コードに 0x45 または 0x52 を設定し、任意の ID コード 1～15 を設定します。

【注】 制御コードに 0x52 を設定する場合、ID コード 1～15 を失念するとシリアルプログラマによるファームウェアの書き換えが行えなくなるのでご注意ください。

ID コードプロテクトの制御コードおよび ID コードの設定は、Smart Configurator の[コンポーネント]タブから[r_bsp]を選択し、[ID code 1]～[ID code4]で変更することができます。

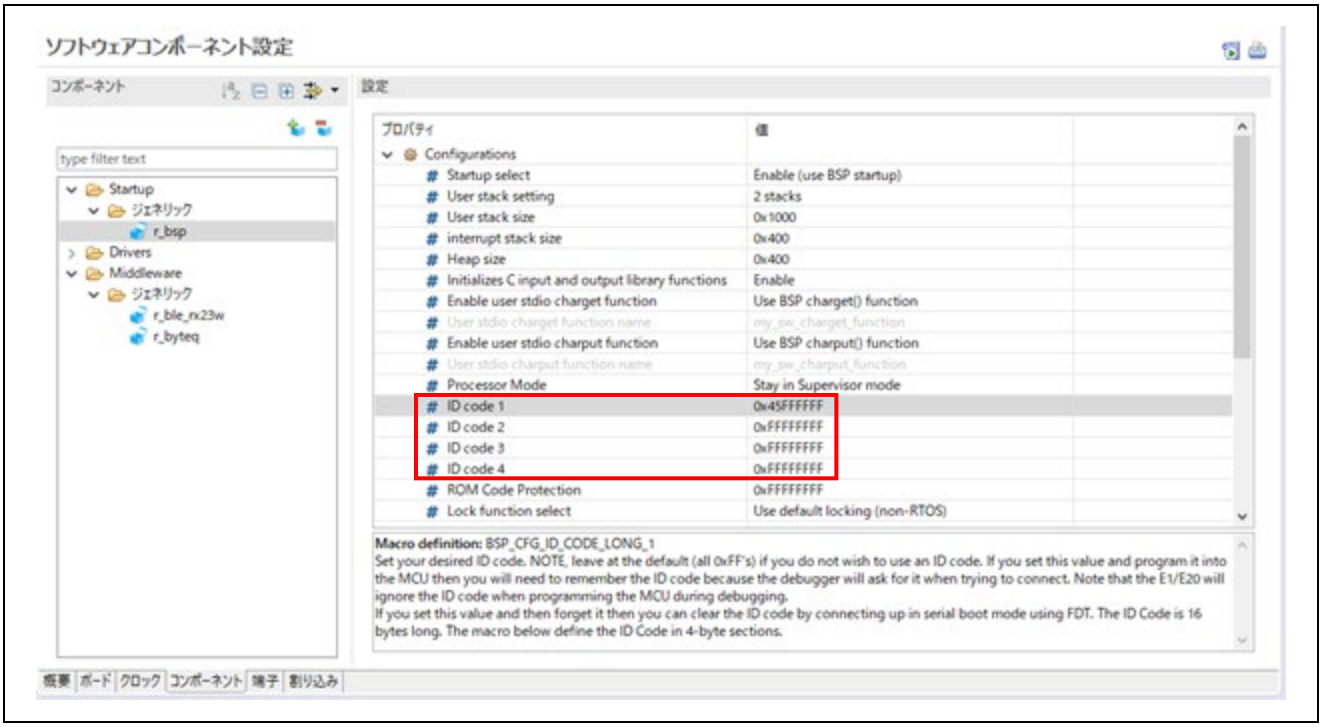


図 5.15 BSP のコンフィギュレーションオプションの設定画面

ID コードプロテクトはファームウェアの書き込み時にユーザ領域(ROM)の ID コードプロテクト領域(0xFFFFFFA0 番地)に書き込まれます。ただし、e² studio から E1 エミュレータなどを使用してデバッグ接続する場合、ファームウェアで ID コードプロテクトを設定していても、ID コードプロテクト領域には ALL 0xFF が設定され、ID コードプロテクトは無効化されます。

ID コードプロテクトが無効の場合、RFP でファームウェアを書き込む際にユーザ領域(ROM)およびデータ領域(E2 データフラッシュ)のすべてのブロックは一旦消去されます。

そのため、統合開発環境でデバッグ接続する場合は、ユーザ領域またはデータ領域に書き込まれている BD アドレスを控えておき、ブロック消去されたあとに再度デバイス固有データを書き込んでください。

5.4.6 BD アドレスの決定方法

Bluetooth LE Protocol Stack は R_BLE_Open() API 内で以下の優先順で BD アドレスの初期値を決定します。

(1) データ領域(E2 データフラッシュ)の指定ブロック
(2) ユーザ領域(ROM)の指定ブロック
(3) ファームウェア初期値
(BLE_CFG_RF_DBG_PUB_ADDR または BLE_CFG_RF_DBG_RAND_ADDR)

ランダム BD アドレスに関しては、全ての領域の BD アドレスが無効な場合、MCU のユニーク ID から MCU 固有のスタティックアドレスを生成します。生成したスタティックアドレスは R_BLE_VS_GetBdAddr() API で取得できます。

【注】生成するスタティックアドレスは MCU の電源 OFF やリセットなどで変化せず、固定値となります。
【注】スタティックアドレスは乱数値で構成されるため、他デバイスと値が重複する可能性はゼロではありません。

BD アドレス決定後も、R_BLE_VS_SetBdAddr() API で再度 BD アドレスを変更することができます。
Bluetooth LE Protocol Stack の BD アドレス決定方法について図 5.16 に示します。

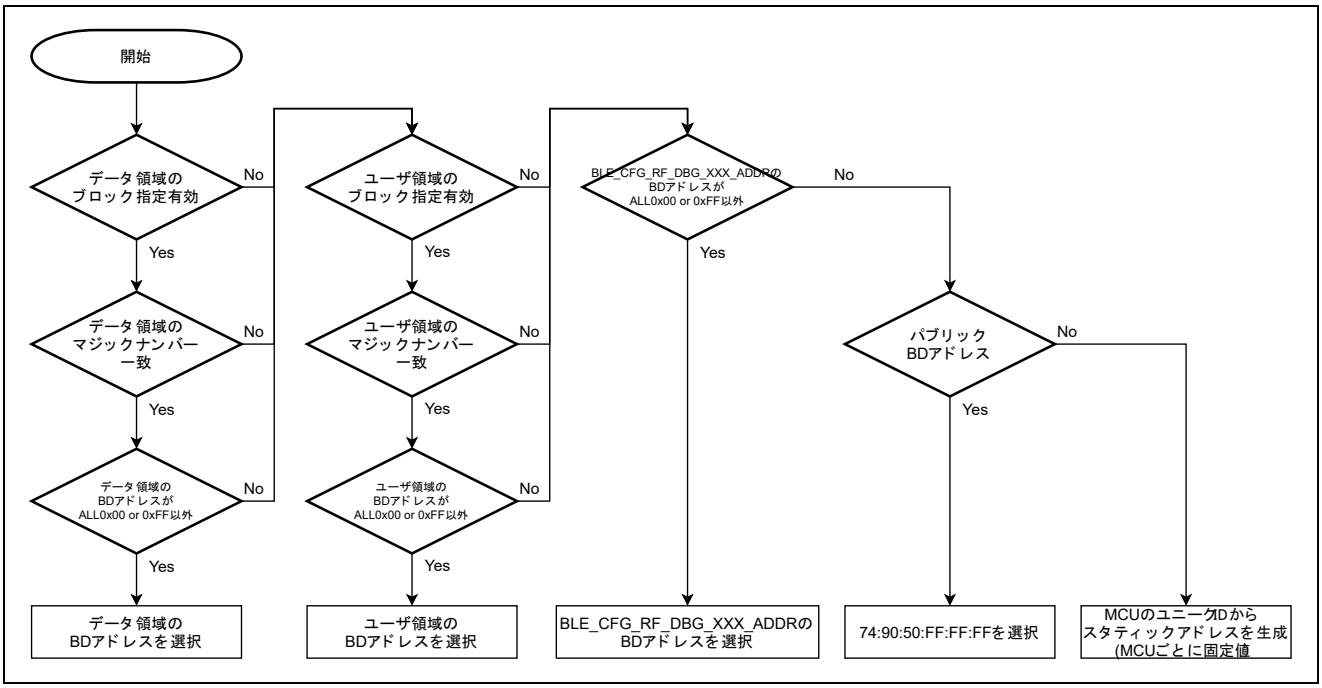


図 5.16 BD アドレスの決定方法

(1)－(3)の BD アドレスの設定では BD アドレスのフォーマットチェックを行わないため、スタティックアドレスを設定する場合は、図 5.17 のフォーマットに合う値を設定してください。

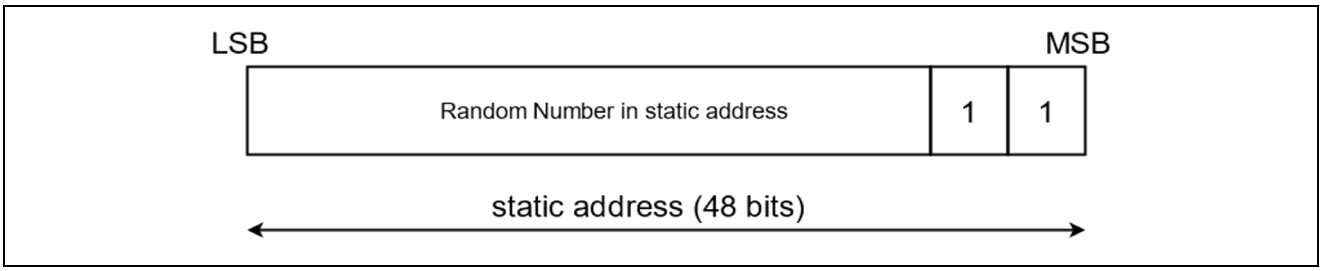


図 5.17 スタティックアドレスのフォーマット

6. HCI モード

HCI(Host Controller Interface)モードは RF 特性評価および BTTS(Bluetooth Test Tool Suite : R01AN4554)用のファームウェアです。PC などのシリアルインタフェース接続されたホストデバイスから HCI コマンドを RX23W マイコンに送信し、Bluetooth LE 通信を行うことが可能です。Bluetooth LE 通信に応じた HCI イベントは RX23W からホストデバイスに送信されます。

HCI モードでは Bluetooth Core Spec ver5.0 に準拠しています。

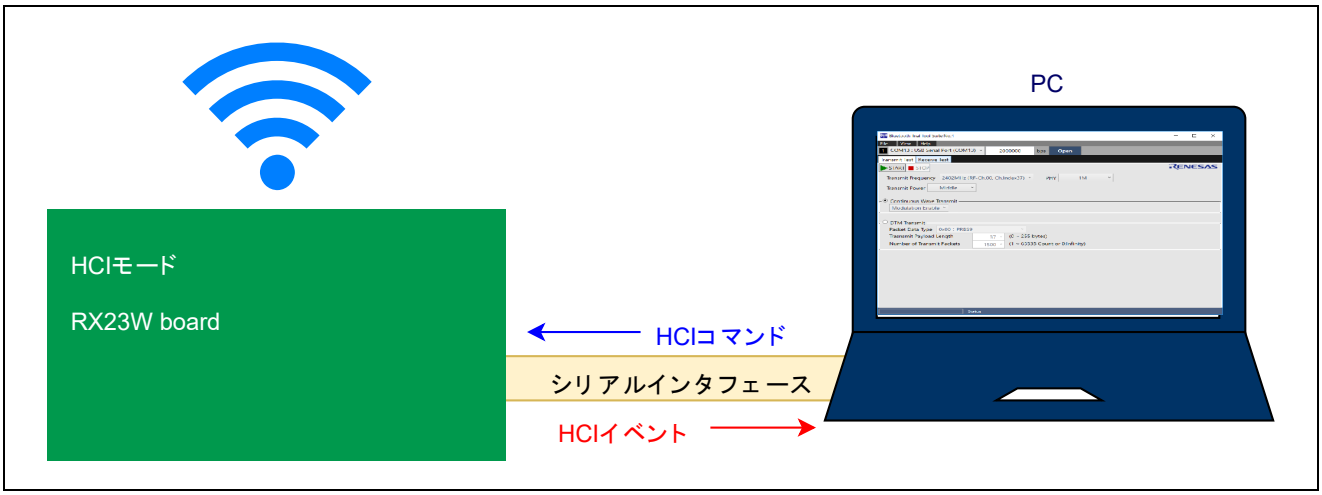


図 6.1 HCI モードの評価ボードと PC の接続

6.1 ソフトウェア構成

HCI モードのソフトウェア構成について図 6.2 に示します。

HCI モードではユーザアプリケーションは搭載できません。

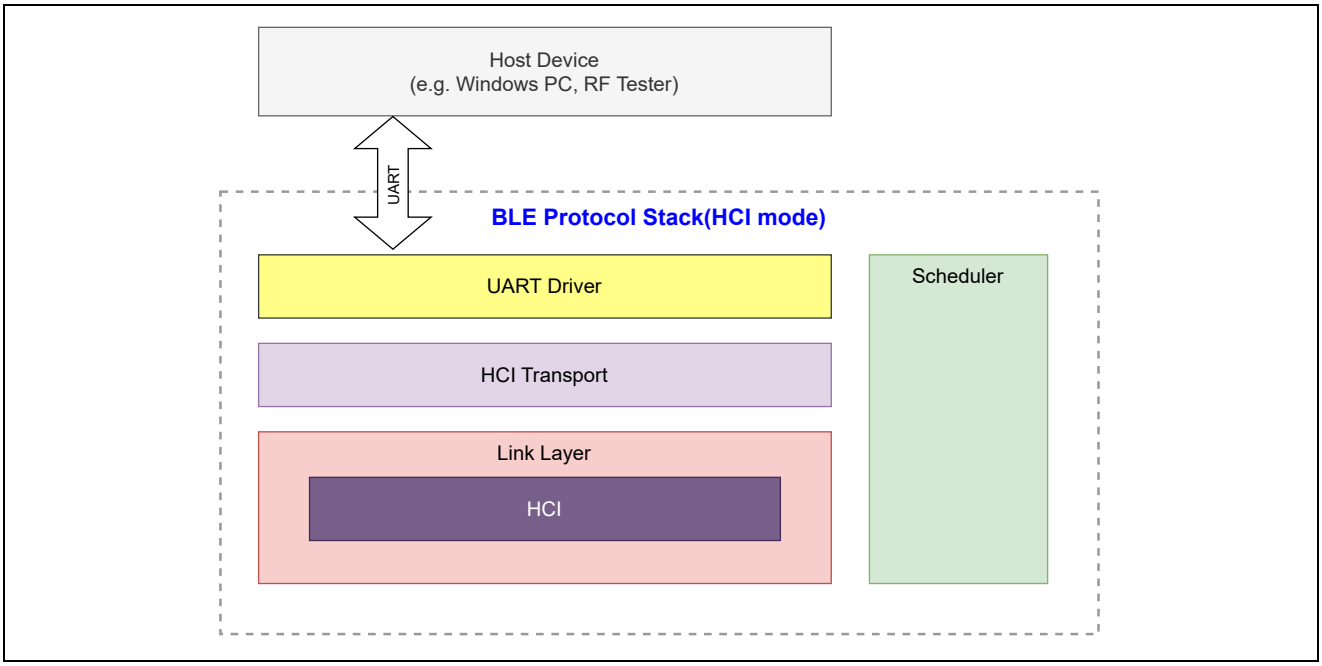


図 6.2 HCI モードのソフトウェア構成

6.2 デモプロジェクト

HCI モードのデモプロジェクトは Bluetooth Low Energy プロトコルスタック 基本パッケージの FITDemos フォルダに含まれます。このプロジェクトは HCI モードファームウェアのベースプロジェクトとしても使用できます。

表 6.1 に HCI モードのデモプロジェクトを示します。

表 6.1 HCI モードのデモプロジェクト

ファイル名	概要
ble_demo_rsskrx23w_uart_hci.zip	RSSK (RX23W 85 ピン BGA)向け HCI モードプロジェクト一式
ble_demo_tbrx23w_uart_hci.zip	Target Board for RX23W (RX23W 56 ピン QFN)向け HCI モードプロジェクト一式
ble_demo_tbrx23wmodule_uart_hci.zip	Target Board for RX23W module(RX23W module 83 ピン LGA)向け HCI モードプロジェクト一式

表 6.2 に HCI モードのデモプロジェクトのファイル構成を示します。

表 6.2 HCI モードのデモプロジェクトファイル構成

```
ble_demo_XXrx23w(module)_uart_hci.zip          XX: rssk または tb
├── .cproject                                  e2 studio プロジェクトファイル
├── .project                                   e2 studio プロジェクトファイル
├── ble_demo_section_rom384kb.esi              ROM384KB 向けセクション情報ファイル
├── ble_demo_section_rom512kb.esi              ROM512KB 向けセクション情報ファイル
├── ble_demo_XXrx23w_uart_hci.launch           デバッグ情報ファイル
├── ble_demo_XXrx23w_uart_hci.scfg             スマート・コンフィグレータ設定ファイル
├── ble_demo_XXrx23w_profile_client.ewd        IAR デバッガ設定ファイル
├── ble_demo_XXrx23w_profile_client.ewp        IAR プロジェクトファイル
├── ble_demo_XXrx23w_profile_client.eww        IAR ワークスペースファイル
├── lnkr5f523w7.icf                            IAR R5F23W7xxxx 用リンカ設定ファイル
├── lnkr5f523w8.icf                            IAR R5F23W8xxxx 用リンカ設定ファイル
├── ble_demo_uart_hci.rcpc                     Renesas 共通プロジェクトファイル
└── src¥
    ├── app_main.c                             HCI モードメインコード
    ├── smc_gen¥                               コード生成フォルダ(内容省略)
    └── uart_hci¥                              UART ドライバフォルダ
        ├── r_ble_dtc.c                        DTC 制御ドライバソースファイル
        ├── r_ble_dtc.h                        DTC 制御ドライバヘッダファイル
        ├── r_ble_uart_hci.c                   SCI 制御ドライバソースファイル
        └── r_ble_uart_hci.h                   SCI 制御ドライバヘッダファイル
```

HCI モードでは、BLE FIT モジュールの下記ファイルおよびフォルダをビルド対象外に設定する必要があります。新規プロジェクトで HCI モードを作成する場合、下記ファイルとフォルダをビルド対象外に設定してください。

src¥smc_gen¥r_ble_rx23w¥src¥platform¥r_ble_pf_lowpower.c
src¥smc_gen¥r_ble_rx23w¥src¥app_lib　※フォルダ全体をビルド対象外

- e² studio

図 6.3 に示すように該当ファイル(またはフォルダ)を右クリック→[プロパティ]を選択し、「ビルドからリソースを除外」をチェック ON にして[Apply and Close]ボタンをクリックします。

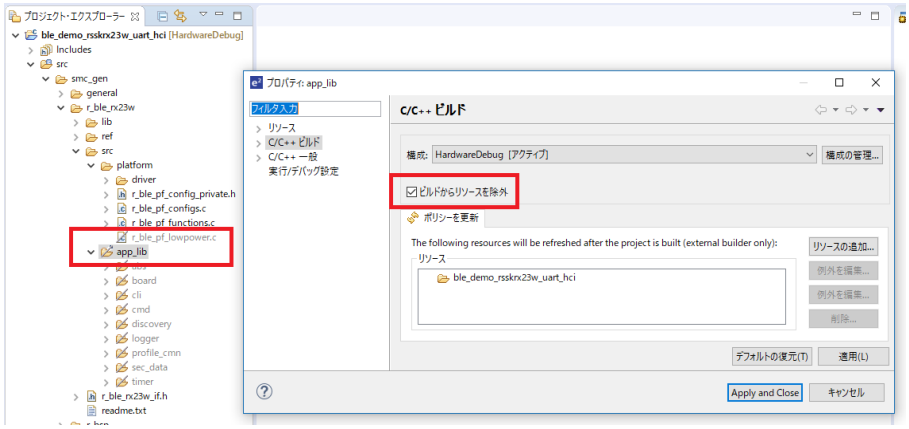


図 6.3　e² stuido のビルド対象外設定

- IAR Embedded Workbench for Renesas RX

図 6.4 に示すように該当ファイル(またはフォルダ)を右クリック→[オプション]を選択し、「ビルドから除外」をチェック ON にして[OK]ボタンをクリックします。

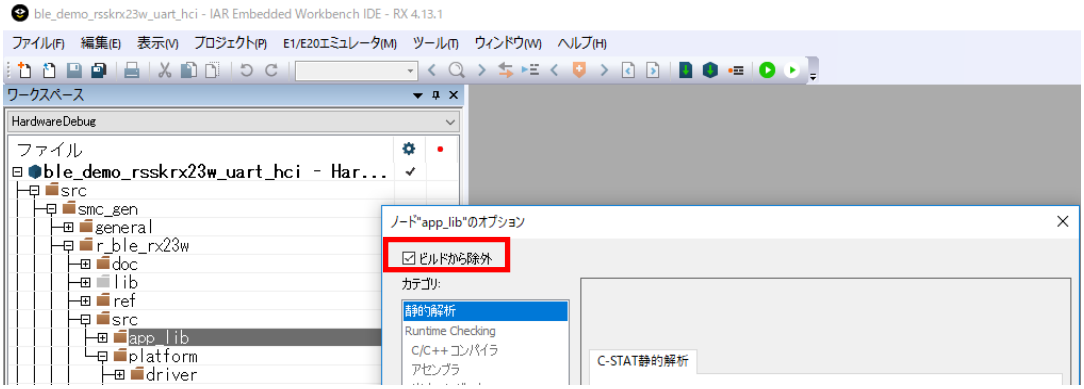


図 6.4　IAR Embedded Workbench for Renesas RX のビルド対象外設定

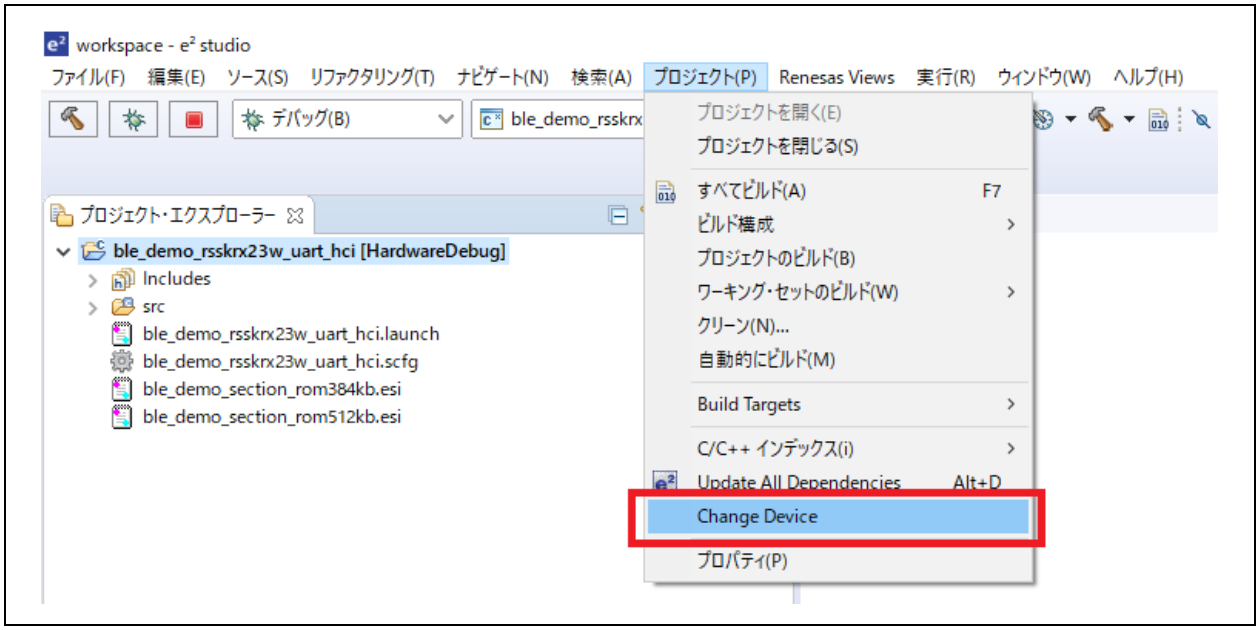
HCI モードではオプション機能としてデータフラッシュの FIT モジュール(r_flash_rx v4.10 以降)を使用します。

6.2.1 プロジェクトのデバイス種別変更

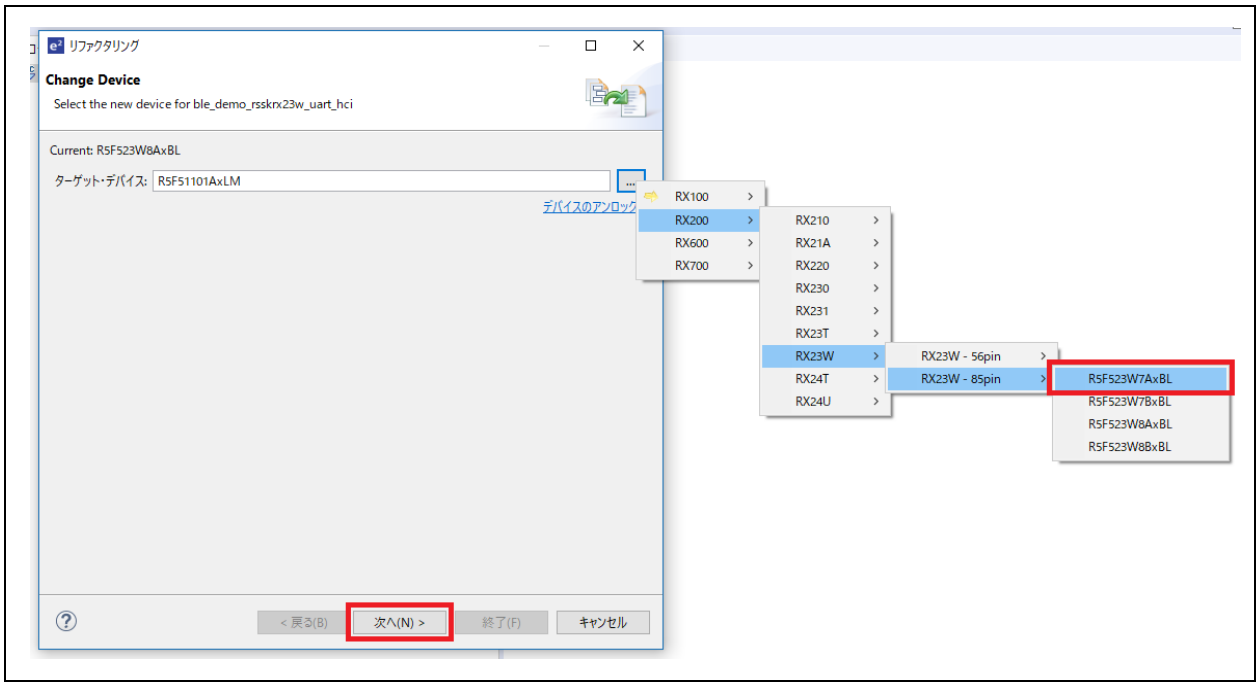
デモプロジェクトでは RX23W の 512KB デバイス(R5F523W**8**xxxx)が選択されています。

本節では ble_demo_rsskrx23w_uart_hci.zip (R5F523W8AxBL)を BGA384KB デバイス(R5F523W7AxBL)への変更方法を説明します。

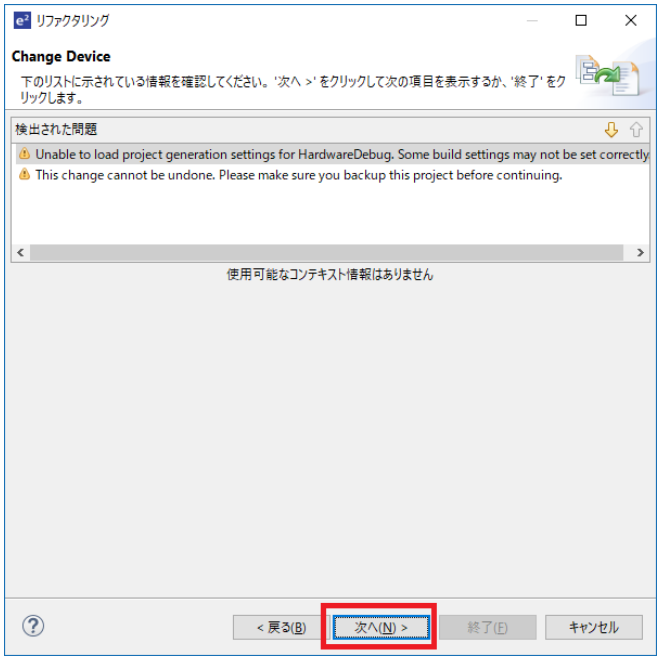
1. メニューの[プロジェクト]→[Change Device]を選択します。



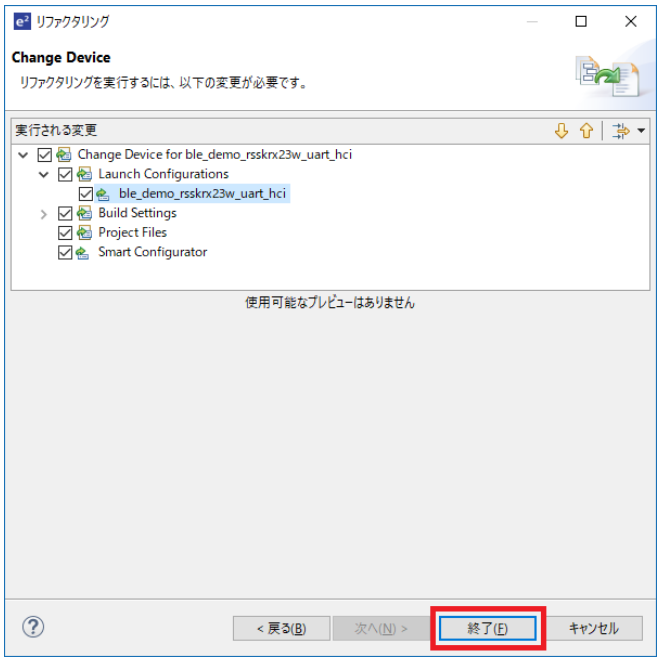
2. 変更先の RX23W デバイス(ここでは 85pin R5F523W7AxBL)を選択し、「次へ」をクリックします。



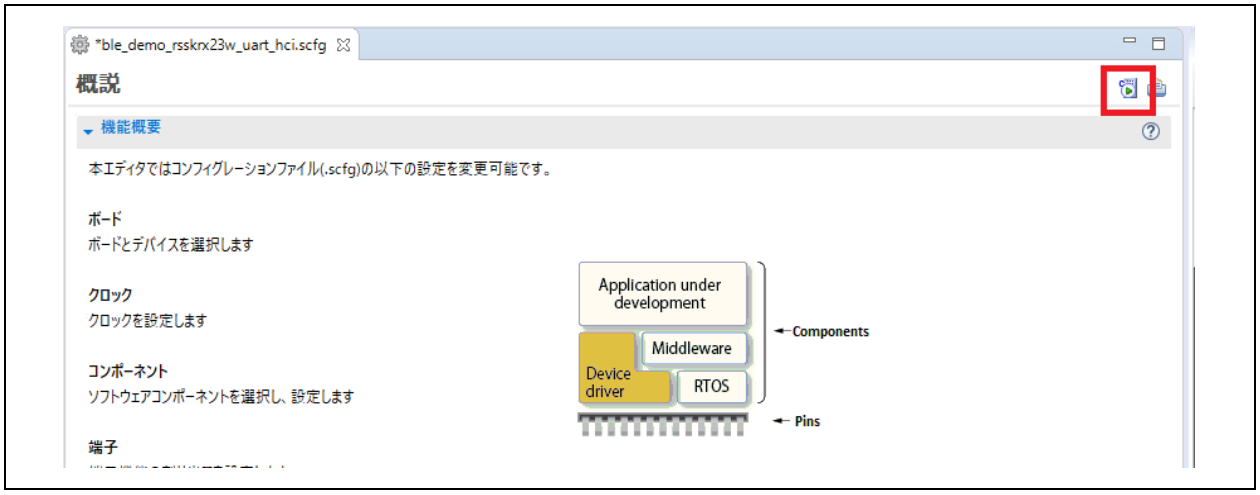
3. 「次へ」をクリックします。



4. 全項目チェック ON であることを確認し、「終了」をクリックします。

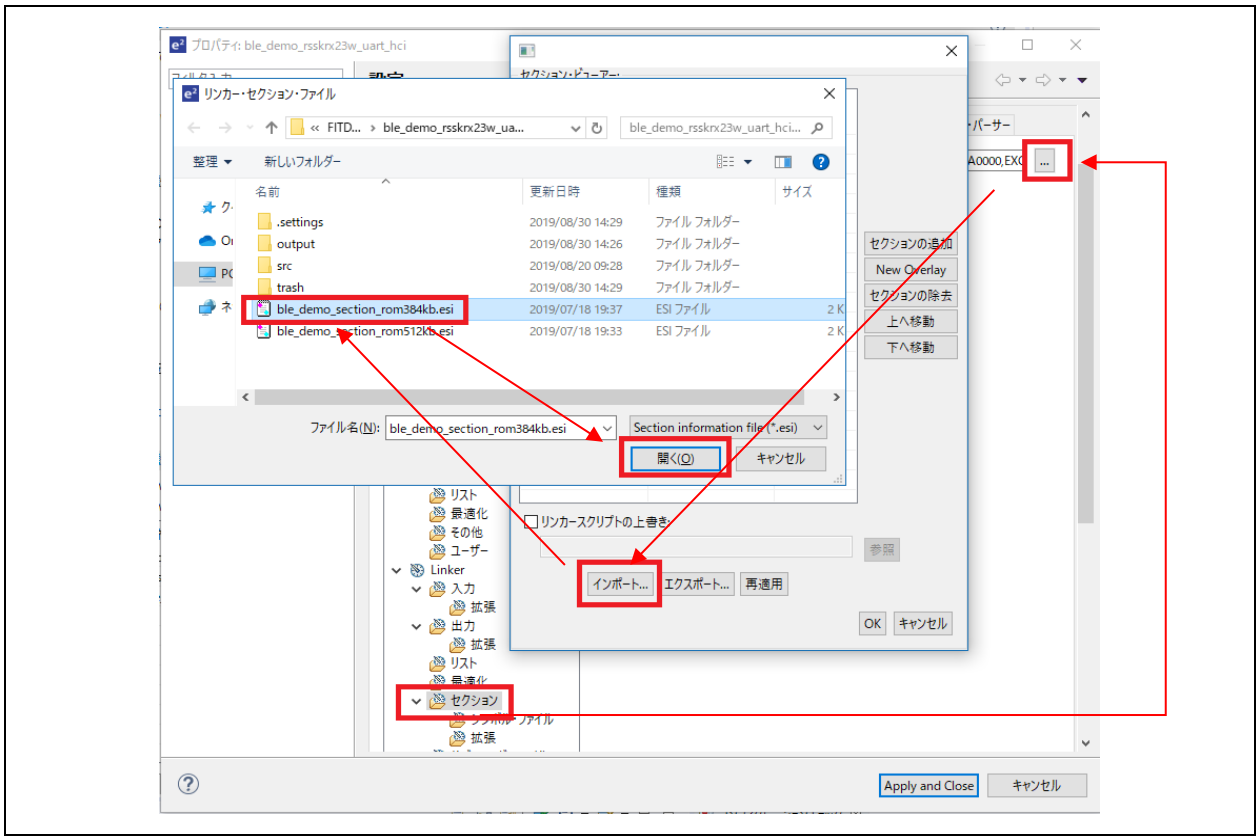


5. スマート・コンフィグレータの「コード生成」ボタンをクリックします。変更したデバイス向けのコード生成が行われます。



6. [Linker]→[セクション]の[…]ボタンをクリックし、[インポート]ボタンをクリックします。“ble_demo_section_rom384kb.esi” ファイルを選択し、[開く]ボタンをクリックします。

【注】512KB デバイスの場合は“ble_demo_section_rom512kb.esi” ファイルを選択してください。



7. 使用するボードの SCI チャネル、ポート番号の設定を行います。設定方法は「6.3.1 UART ドライバのコンフィギュレーション」を参照してください。

8. プロジェクトのビルドを行い、ビルドエラーが発生する場合は設定を見直してください。

6.3 UART ドライバ

HCI モードでは RX23W のシリアルコミュニケーションインタフェース(SCIg, SCIh)およびデータトランスファコントローラ(DTCa)を使用して UART (Universal Asynchronous Receiver/Transmitter)通信を行います。

HCI モードでは専用の UART ドライバを用意し、Bluetooth LE Protocol Stack から UART ドライバの API を使用します。

HCI モードの UART ドライバファイル構成を表 6.3 に示します。

表 6.3 UART ドライバ ファイル構成

ファイル名	概要
r_ble_dtc.c	DTC 制御ドライバソースファイル
r_ble_dtc.h	DTC 制御ドライバヘッダファイル
r_ble_uart_hci.c	SCI 制御ドライバソースファイル
r_ble_uart_hci.h	SCI 制御ドライバヘッダファイル (※1)

※1：SCI チャネルを変更する場合、本ファイルを編集する必要があります。

6.3.1 UART ドライバのコンフィギュレーション

UART ドライバは r_ble_uart_hci.h ファイルの各マクロによって SCI チャネル、UART ボーレートなどを変更することが可能です。変更する場合は直接ファイルを編集してください。

各マクロは r_ble_rx23w_config.h の BLE_CFG_BOARD_TYPE マクロの値によって設定箇所が分岐しています。ユーザのボード環境で設定する場合、BLE_CFG_BOARD_TYPE マクロに(0)を設定してください。

表 6.4 に UART ドライバのコンフィギュレーション概要を示します。

表 6.4 UART ドライバ コンフィギュレーション (BLE_CFG_BOARD_TYPE == 0 の場合)

No.	マクロ名	初期値	概要
1	SCI_CHANNEL	1	SCI チャネル番号を設定します。 1: SCI1 5: SCI5 8: SCI8 12: SCI12
2	SCI_INTR_PRIORITY	14	SCI 割り込み優先レベルを設定します。 範囲： 1(低優先) – 15(高優先)
3	SCI_CTS_RTS_EN	1	SCI の CTS/RTS 機能を設定します。 0: CTS/RTS 無効 1: RTS 有効 2: CTS 有効
4	SCI_RXD_PIN_X SCI_RXD_PIN_Y	3 0	SCI の RXD で使用するポート番号を設定します。 例： P30 の場合、X=3, Y=0 PC6 の場合、X=C, Y=6
5	SCI_TXD_PIN_X SCI_TXD_PIN_Y	2 6	SCI の TXD で使用するポート番号を設定します。 例： P26 の場合、X=2, Y=6 PC7 の場合、X=C, Y=7
6	SCI_CTS_RTS_PIN_X SCI_CTS_RTS_PIN_Y	3 1	SCI_CTS_RTS_EN が 1 の場合に SCI の CTS/RTS で使用するポート番号を設定します。 例： P31 の場合、X=3, Y=1 PC4 の場合、X=C, Y=4
7	DBG_CALC_BAUDRATE	有効	本マクロが有効な場合、UART ボーレートに対応したレジスタ設定値を自動計算します。 本マクロを無効にした場合、SCI のビットレートレジスタ(BRR)およびモジュレーションデューティレジスタ(MDDR)などを手動で設定する必要があります。
8	DBG_BAUDRATE_SWITCH	有効	本マクロが有効な場合、UART ドライバ初期化時に SCI_BR_SW_PIN_X および SCI_BR_SW_PIN_Y で指定されるポートの入力状態をチェックし、入力レベルに応じてボーレートを切り替えます。 入力 Low： D_UART_BR_SWITCH を選択 入力 High： D_UART_BR を選択
9	D_UART_BR	115200	DBG_CALC_BAUDRATE マクロが有効な場合の、ボーレートの初期値を設定します。
10	D_UART_BR_SWITCH	2000000	DBG_CALC_BAUDRATE および DBG_BAUDRATE_SWITCH マクロが有効な場合に、切り替え用のボーレートを設定します。
11	SCI_BR_SW_PIN_X SCI_BR_SW_PIN_Y	1 4	DBG_CALC_BAUDRATE および DBG_BAUDRATE_SWITCH マクロが有効な場合に、入力レベルをチェックするポート番号を設定します。 例： P14 の場合、X=1, Y=4

7. 付録

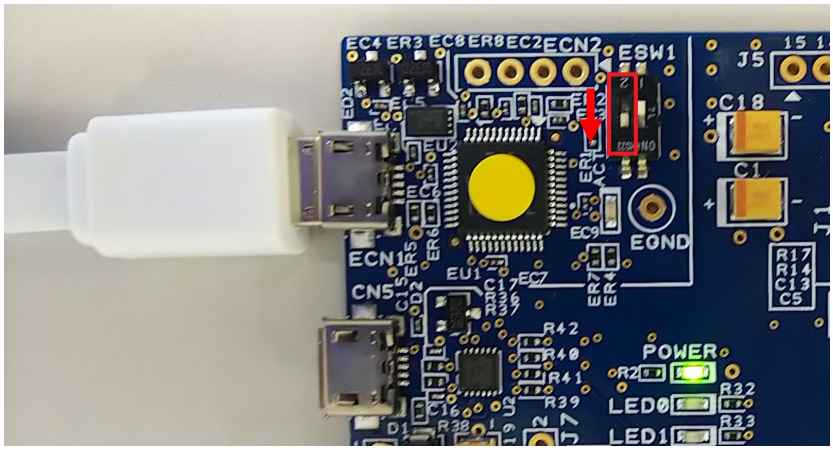
7.1 Target Board for RX23W のファームウェア書き込み手順(デバイス固有データ保持)

ここでは、ビルドして作成したファームウェアのプログラムファイル(mot ファイルなど)を、Renesas Flash Programmer(以降は RFP と記載)を使用して RX23W デバイスに書き込む方法について説明します。

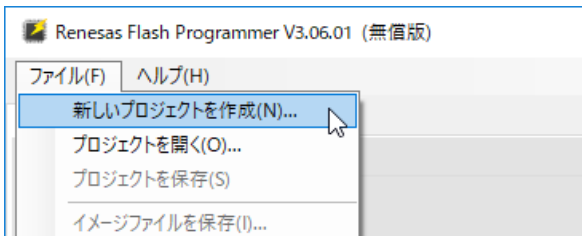
【注】RFP は v3.06.00 以降のバージョンを使用してください。

以降に Target Board for RX23W (以降 TB と記載)向けの RFP によるデバイス固有データを保持しながらファームウェアを書き込む手順を示します。なお、デバイス固有データを保持するためには、ID コードプロテクトが有効なファームウェアがすでに書き込まれていることが必須条件となります。

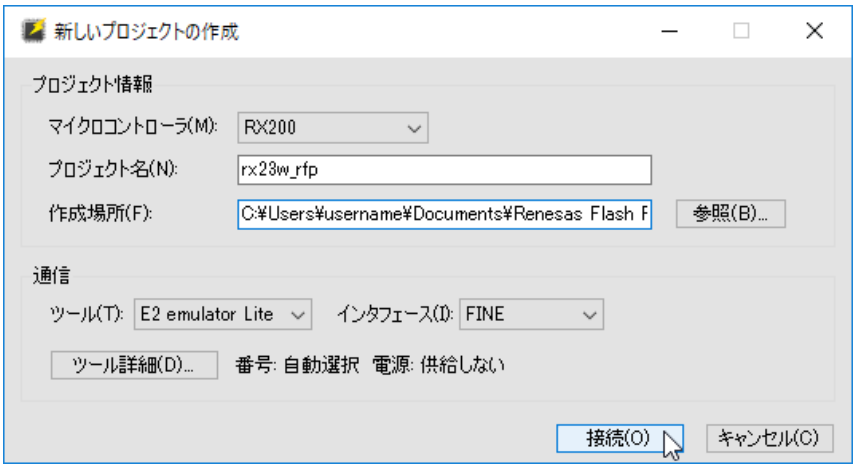
1. TBのESW1-2をONに変更し、PCとECN1コネクタとをUSB A-microBケーブルで接続してください。



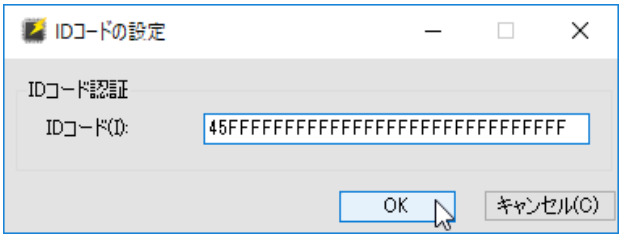
2. RFPを起動し、[ファイル]→[新しいプロジェクトを作成]を選択します。



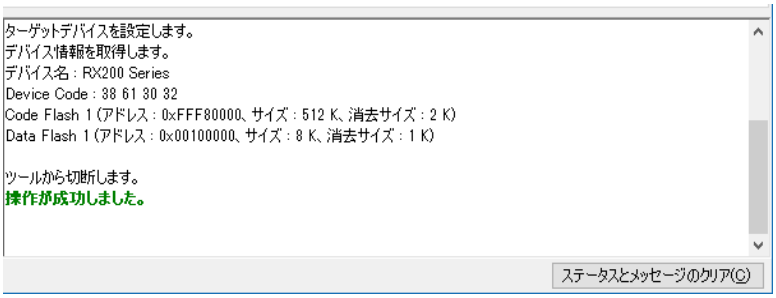
3. [新しいプロジェクトの作成]ウインドウで、以下の設定を行い、[接続]ボタンをクリックします。
   - マイクロコントローラ： RX200 を選択
   - プロジェクト名： 任意のプロジェクト名を入力
   - 作成場所：任意のフォルダを選択
   - 通信 ツール： “E2 emulator Lite” を選択、インタフェース： “FINE” を選択
   - 電源：”供給しない” を選択 (デフォルト)



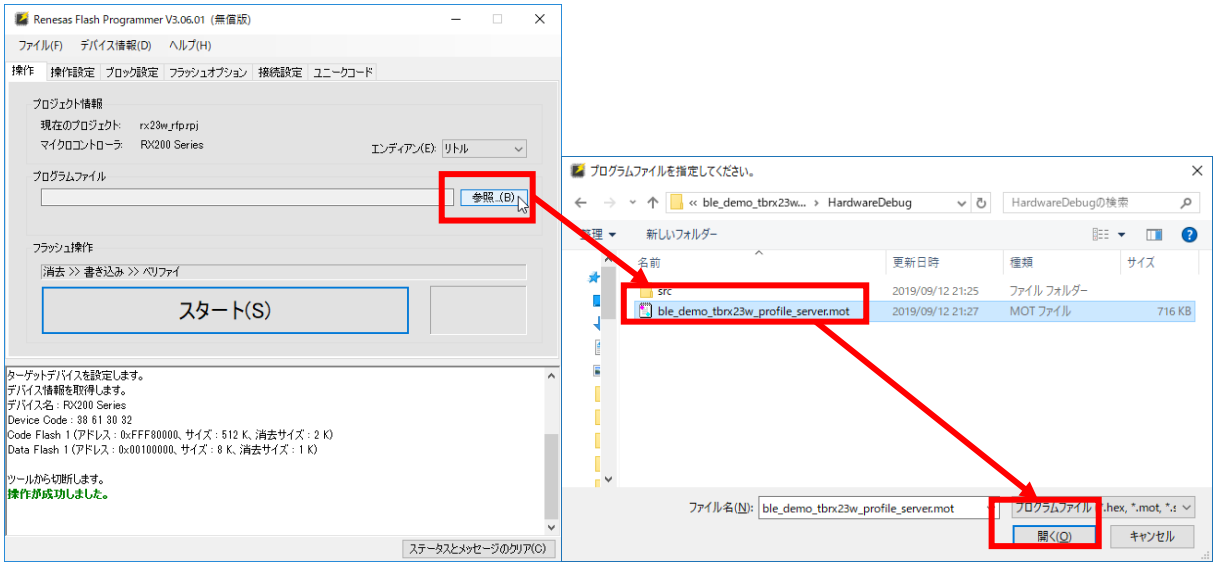
4. IDプロテクトコードが有効なファームウェアが書き込まれている場合、[IDコードの設定]の入力を求められます。その場合、書き込まれているIDコードを入力し、[OK]ボタンをクリックしてください。ここでは、IDコード”**45FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF**”を入力例として記載します。IDコードを変更している場合は、適切な値を入力してください。



5. 接続に成功すると、「**操作が成功しました。**」と表示されます。

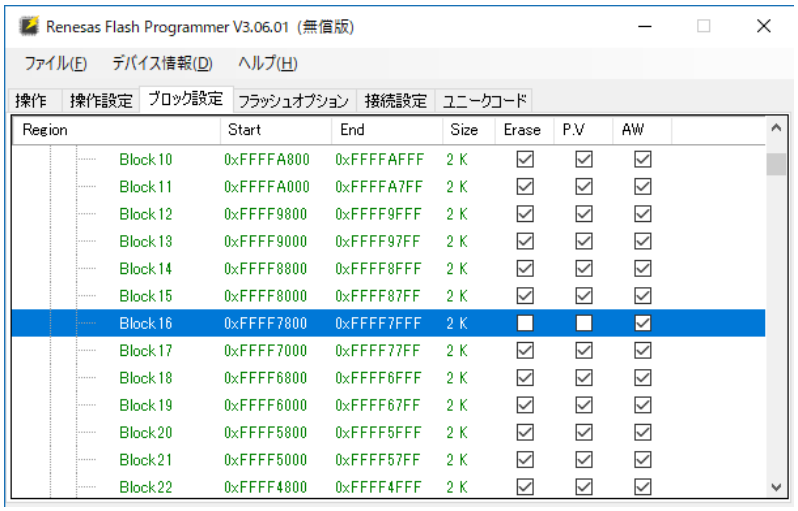


6. [参照]ボタンをクリックし、書き込みたいプログラムファイルを選択し、[開く]ボタンをクリックします。



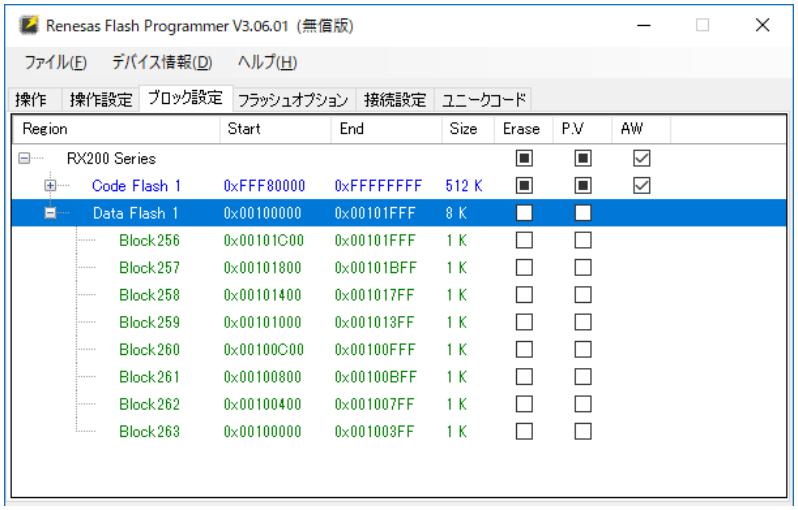
7. すでに書き込み済みのデバイス固有データを消去しないように、[ブロック設定]タブで以下を設定します。

- ユーザ領域(ROM)： BLE_CFG_DEV_DATA_CF_BLOCK で指定したブロック(例ではブロック 16)の[Erase] [P.V]のチェックを OFF

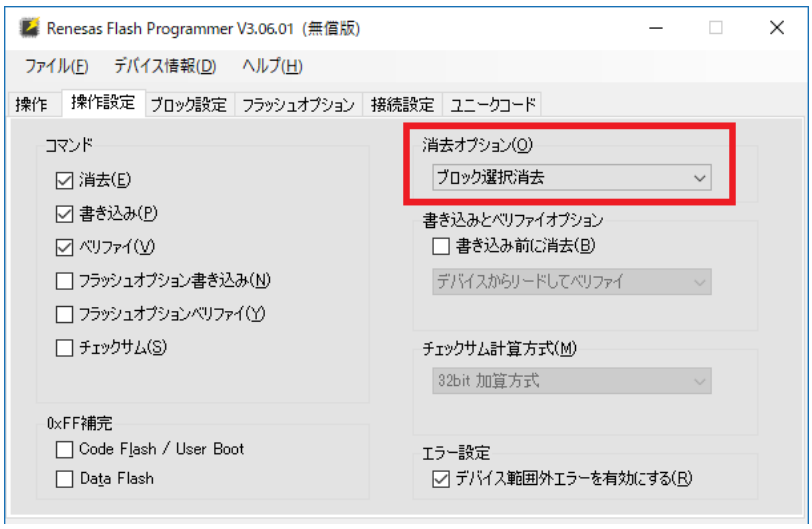


【注】 本チェックを OFF にした場合、ユニークコード書き込みも行えなくなるため、該当ブロックにユニークコード書き込みを行う場合はチェック ON にしてください。

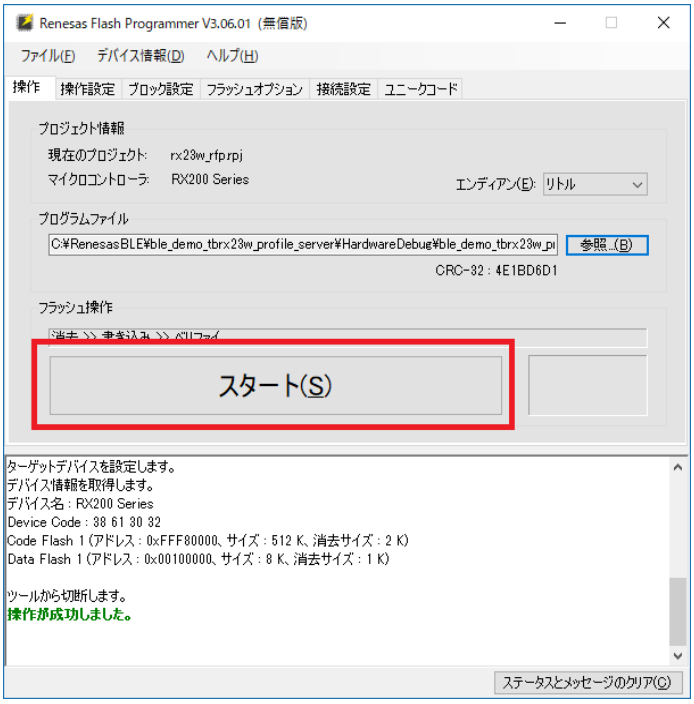
- データ領域(E2 データフラッシュ)の場合：全ての Data Flash ブロックの[Erase] [P.V]のチェックを OFF



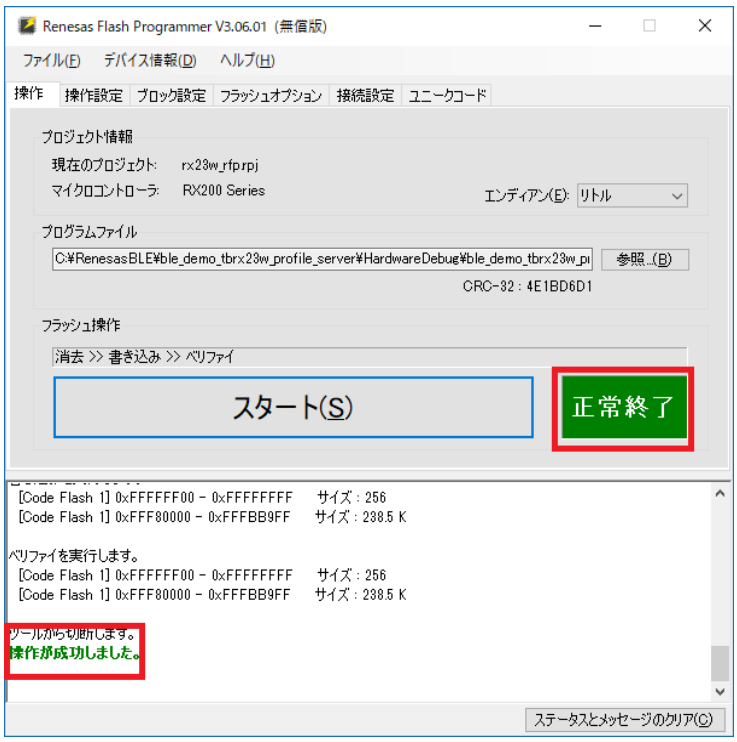
8. [操作設定]タブの[消去オプション]で”ブロック選択消去”を選択します。(デフォルトで選択されています)



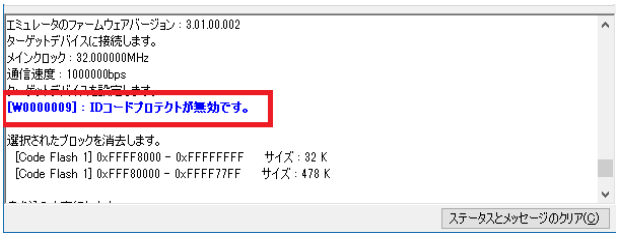
9. [操作]タブの[スタート]ボタンをクリックし、ファームウェア書き込みを開始します。



10. 書き込みが正常に終了すると、「**操作が成功しました。**」および「**正常終了**」と表示されます。



【注】 書き込み時にメッセージ部で「**[W0000009] : ID コードプロテクトが無効です。**」と表示された場合、フラッシュメモリのすべてのブロックは一旦消去され、デバイス固有データも無効になります。この場合、再度デバイス固有データを書き込む必要があります。



改訂記録	Bluetooth Low Energy プロトコルスタック 基本パッケージ ユーザーズマニュアル

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.10.31	ー	初版発行
1.01	2020.1.27	3	表 2.1 に ROM_Files を追加。
		4	表 2.3 に IAR 環境のソフトウェア条件を追加。
		5	IAR Embedded Workbench for Renesas RX デモプロジェクトの説明を追加。
		15-22	表 4.1 コンフィギュレーションオプションを BLE FIT モジュール v1.10 の変更に合わせて更新。
		67	BLE コマンド追加。
		93	表 6.2 に IAR Embedded Workbench for Renesas RX 用のファイルを追加。
		94	IAR Embedded Workbench for Renesas RX でのビルド対象外設定を追加。
		107	7.2 接続パラメータ更新リクエストへの応答の節追加。
1.02	2020.9.30	3	2.1 パッケージ構成にパッケージの入手方法について追記。
		5	2.2 プロジェクト構成を追加。
		6	表 2.4 の IAR 環境の説明の【注】を修正。
		16	3.2.7 LED and Switch 制御を追加。 3.2.8 BLE タスク制御を追加。
		25	DTBL/BTBL について補足を追記。
		60	5.1.2 Vendor Specific (VS)コマンドに(6)Scan Channel 操作コマンドを追加。
		73	表 5.17 の RF スリープモードの状態変更通知コールバック関数パラメータ値誤りを修正
		79	生成するランダム BD アドレスについて注記を追記。
		ー	以下の項目を RX23W グループ Bluetooth Low Energy アプリケーション開発者ガイド (R01AN5504)に移行。 4.3 ボードの LED, プッシュスイッチの設定 4.4 MCU 低消費電力設定 5.1.2 コマンド作成手順 5.2 ロガー 5.3 ソフトウェアタイマ 7.2 接続パラメータ更新リクエストへの応答

Rev.	発行日	改訂内容	
		ページ	ポイント
1.03	2021.3.30	-	RX23W モジュール(R5F523W8CxLN, R5F523W8DxLN)をサポート。
		4	表 2.1 に RX23W module 向けの FITDemo を追加。
		6	- 表 2.3 に Target Board for RX23W module を追加。 - 表 2.4 に e^{2}studio 2021_01 を追加。
		20	- RX23W モジュール使用時の BLE_CFG_RF_CLVAL の設定値追加。 - Target Board 使用時の BLE_CFG_RF_DDC_EN の設定値を追加。
		34	- Advertising コマンドに(addr_type)と(-wl)を追加 - API パラメータの説明を削除して API ドキュメントへの参照に変更
		35	- Scan コマンドに(addr_type)と(-wl)を追加 - API パラメータの説明を削除して API ドキュメントへの参照に変更
		36	- 接続コマンドに(own_addr_type)と(dummy_irk)と(-wl)を追加 - API パラメータの説明を削除して API ドキュメントへの参照に変更
		38	プライバシーコマンドに remove と get と off を追加
		63	RF スリープモードへの遷移の説明を修正。
		72	表 6.1 に Target Board for RX23W module 用の HCI モードプロジェクト一式を追加。
1.04	2021.6.30	4	表 2.1 の RX23W module 用 FITDemo のプロジェクト一式の名称を修正。
		6	表 2.4 の e^2 studio の 64bit 版のバージョンを 2021_04 に変更。
		72	表 6.2 に rcpc ファイルを追加。
2.30	2021.10.15	-	ユーザマニュアルのバージョンを BLE FIT のバージョンに合わせて変更。
		21	表 4.1 の BLE_CFG_MCU_MAIN_CLK_KHZ の PLL Circuit 使用時の設定範囲を修正。
2.31	2022.4.15	3	Bluetooth Low Energy プロトコルスタック 基本パッケージの URL を修正。
		5	表 2.2 から ref ディレクトリを削除。 表 2.2 のプロジェクト構成を QE for BLE v1.40 以降に変更。
		6	e^{2}studio v7.8 (32bit)を表 2.4 から削除。 e^{2}studio のインストーラから CCRX v2.08 ダウンロードの記載を削除。
		17	BLE_CFG_RF_CONN_MAX の説明を修正。
		22	BLE_CFG_NUM_BOND の説明を修正。
		47	Tx Power 設定コマンドのパラメータの説明を追記。
		76	HCI プロジェクトのライブラリ名再設定についての記載を削除。
2.40	2022.6.30	15	抽象 API のコードを変更しないよう追記。
2.50	2022.12.27	-	BLE FIT v2.50 以降はプロファイル共通部を含まず、QE for BLE(V1.6.0 以降)で提供されることを追記。
		56-60	図 5.2, 5.3, 5.5 と表 5.5-5.7 を修正。
2.60	2022.05.17	6	BLE FIT v2.60 の改版に伴い e^2 studio のバージョンを 2024.01-1 に更新。
		7	FIT モジュールの追加方法について記載内容を更新。
2.61	2025.03.31	-	ユーザマニュアルのバージョンを BLE FIT のバージョンに合わせて変更。

Bluetooth Low Energyプロトコルスタック 基本パッケージ
ユーザーズマニュアル

発行年月日　2025.03.31　　Rev.2.61

発行　　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）

Bluetooth Low Energy プロトコルスタック 基本パッケージ

ルネサス エレクトロニクス株式会社

R01UW0205JJ0261