

RX23W グループ Bluetooth Mesh スタック

スタートアップガイド

要旨

Bluetooth Mesh スタックは、Bluetooth Mesh Networking 仕様に準拠した Mesh ネットワークを構成して、多対多デバイス間で無線通信を実行するためのソフトウェアライブラリです。

本書は Bluetooth Mesh スタックパッケージの導入方法を示します。Bluetooth Mesh スタックパッケージのソフトウェア構成や各レイヤーの概要、Mesh アプリケーションの開発方法については「RX23W グループ Bluetooth Mesh スタック 開発ガイド」(R01AN4875)を参照してください。

対象デバイス

RX23W グループ

関連文書

下記の文書がルネサスのウェブサイトにて公開されています。

文書名	文書番号
RX23W グループ ユーザーズマニュアル ハードウェア編	R01UH0823
CC-RX コンパイラ ユーザーズマニュアル	R20UT3248
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	R01UW0205
RX23W グループ Bluetooth Mesh スタック スタートアップガイド	R01AN4874 本書
RX23W グループ Bluetooth Mesh スタック 開発ガイド	R01AN4875
RX23W グループ Target Board for RX23W ユーザーズマニュアル	R20UT4634
RX23W Group Target Board for RX23W module ユーザーズマニュアル	R20UT4890
RX23W グループ Renesas Solution Starter Kit for RX23W CPU ボード ユーザーズマニュアル	R20UT4446

目次

1. 特徴	4
1.1 通信トポロジー	4
1.2 Mesh ネットワーク動作	5
1.3 Mesh デバイスのライフサイクル	6
2. 概要	7
2.1 Bluetooth Mesh デモ	7
2.1.1 デモフェーズ 1: プロビジョニング	7
2.1.2 デモフェーズ 2: コンフィグレーション	7
2.1.3 デモフェーズ 3: モデル通信	8
2.1.4 デモフェーズ 4: ノードリムーバル	8
3. ハードウェア要件	9
3.1 Target Board for RX23W	9
3.2 Target Board for RX23W module	9
3.3 Renesas Solution Starter Kit for RX23W	10
3.4 E2 エミュレータ Lite	11
3.5 モバイル端末	11
4. ソフトウェア要件	12
4.1 Bluetooth Mesh スタック	12
4.2 Bluetooth Low Energy プロトコルスタック	12
4.3 Mesh モバイルアプリケーション	12
4.4 e ² studio IDE および CC-RX コンパイラ	12
4.5 Renesas Flash Programmer	12
4.6 サードパーティー ソフトウェア	13
4.6.1 シリアルターミナルツール	13
4.6.2 Mesh モバイルのビルドに必要な開発ツール	13
5. Bluetooth Mesh スタックパッケージ	14
5.1 構成	14
6. 開発環境の構築	16
6.1 e ² studio IDE および CC-RX コンパイラ	16
6.2 デモプロジェクト	16
6.3 Mesh モバイル	18
7. デモの実行	19
7.1 フェーズ 1: プロビジョニング	20
7.2 フェーズ 2: コンフィグレーション	21
7.3 フェーズ 3: モデル通信	23

7.3.1	Mesh モバイルによる Generic OnOff ステート制御	24
7.3.2	開発ボードによる Generic OnOff ステート制御	25
7.3.3	Mesh モバイルによる Vendor ステート制御	25
7.3.4	開発ボードによる Vendor ステート制御	26
7.3.5	Mesh モバイルによるノードリセット	27
8.	Appendix	28
8.1	ログメッセージ	28

1. 特徴

Bluetooth Mesh スタックソフトウェアが実現する Bluetooth Mesh ネットワークの特徴を紹介します。

1.1 通信トポロジー

Bluetooth Low Energy 技術が提供する通信トポロジーは、ポイントツーポイントとブロードキャストです。Bluetooth Mesh スタックソフトウェアは、Bluetooth Low Energy 技術を使用して Mesh ネットワークを構築します。図 1-1 に通信トポロジーを示します。

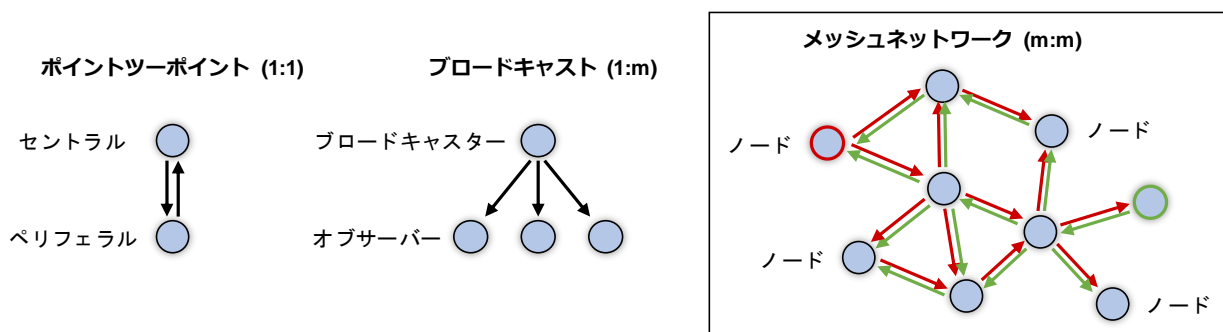


図 1-1 通信トポロジー

- **ポイントツーポイント(Point-to-Point)**

ポイントツーポイントトポロジーでは、セントラルはペリフェラルと双方向で通信します。ユースケースの例としては、セントラルデバイスがスマートフォン、ペリフェラルデバイスがフィットネストラッカーです。スマートフォンがユーザの運動開始を通知し、フィットネストラッカーは運動強度といった測定データを送信します。

- **ブロードキャスト(Broadcast)**

ブロードキャストトポロジーでは、ブロードキャスターが不特定多数のオブザーバーに対してデータを配信し、各オブザーバーは受信したデータを取捨選択して利用します。ユースケースの例として、ブロードキャスターデバイスがクーポン配信ビーコン、オブザーバーデバイスがスマートフォンです。ビーコンからクーポン ID を送信し、スマートフォンはインターネットを経由してクーポン ID に対応するクーポンを取得します。

- **Mesh ネットワーク(Mesh Network)**

Mesh ネットワークトポロジーでは、多数のノードが双方向で通信します。各ノードは任意のタイミングでメッセージを送信できます。さらに、直接通信できない距離にあるノードとも、周囲のノードがメッセージ中継することで通信できます。ユースケースの例としては、照明、空調、コントロールパネル、スマートフォンといったノードで構成されるビルオートメーションシステムです。管理者がコントロールパネルからフロア全体の照明や空調を制御し、さらにビルの利用者はスマートフォンから会議室の照明や空調のみ調整します。

1.2 Mesh ネットワーク動作

Bluetooth Mesh ネットワークの動作を以下に示します。また本 Mesh 動作を実現するネットワークは、マネージドフラッドベースド Mesh ネットワーク(Managed-Flood-based Mesh Network)として定義されません。

図 1-2 に Mesh ネットワークを構成するノードの動作を示します。1つのノード(図中の SRC)がメッセージを送信すると、周囲のノードが次々とメッセージを中継します。メッセージがネットワーク全体に広がり、最終的に宛先のノード(図中の DST)に到達します。

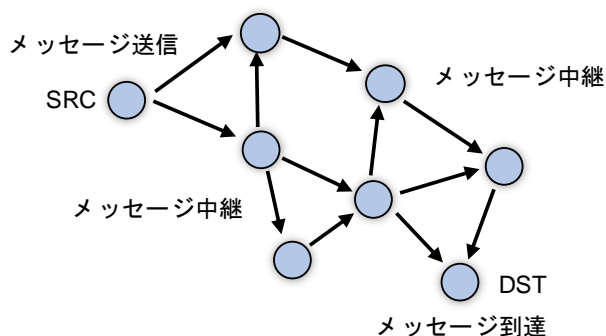


図 1-2 通信動作

さらに、メッセージを無制限に中継しないために、ノードは2つのメソッドを利用します。

- **メッセージキャッシュメソッド(Message Cache method)**

各ノードはメッセージを受信すると、メッセージを中継し、メッセージをキャッシュします。その後、ノードが同一メッセージを再受信した場合は中継しません。

- **タイムトゥリーブメソッド(Time to Live method)**

各ノードは発信するメッセージに、中継の限度回数を TTL(Time to Live)値に設定します。設定できる最大値は 127 です。その他のノードはメッセージの TTL 値を 1 減算して中継します。TTL 値が 0 であるメッセージは中継されません。

1.3 Mesh デバイスのライフサイクル

図 1-3 に Bluetooth Mesh に対応したデバイスの典型的なライフサイクルを示します。

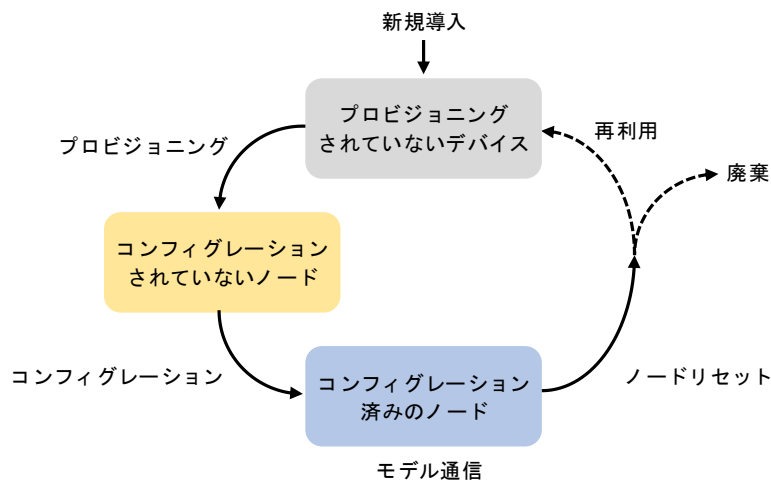


図 1-3 ノードのライフサイクル

Bluetooth Mesh に対応したデバイスは、まずプロビジョニング処理によって認証された後、ネットワークキーと各エレメントのユニキャストアドレスといったプロビジョニングデータを受け取ります。

プロビジョニングの完了後、ノードとなったデバイスは、コンフィグレーションによってアプリケーションキーやパブリッシュアドレス、サブスクリプションアドレスといったモデル動作に必要な設定を受け取ります。

コンフィグレーションの完了後、ノードはモデルによって他ノードと通信することができます。またコンフィグレーションの再実行により、ノードの設定が更新されることがあります。

コンフィグレーションクライアントは、ネットワークキーやアプリケーションキーに関するセキュリティリスクがある場合、これらの暗号鍵を更新します。さらに、不要となったノードには現在使用している暗号鍵の漏えいといったリスクがあるため、不要となったノードをブラックリストに追加します。ブラックリストに追加されたノードは、新しい暗号鍵を受け取ることができません。

コンフィグレーションクライアントは、ノードリムーバルプロシージャによって不要となったノードをネットワークから除外することができます。ネットワークから除外されたデバイスは、再利用または破棄されます。

2. 概要

2.1 Bluetooth Mesh デモ

Bluetooth Mesh スタックパッケージに含まれるサンプルプログラムは、下記 4 つのデモフェーズを実行します。

- デモフェーズ 1: プロビジョニング
- デモフェーズ 2: コンフィグレーション
- デモフェーズ 3: モデル通信
- デモフェーズ 4: ノードリムーバル

パッケージには本デモを実行するためのスマートフォン上で動作する Mesh モバイルアプリケーションも含まれます。

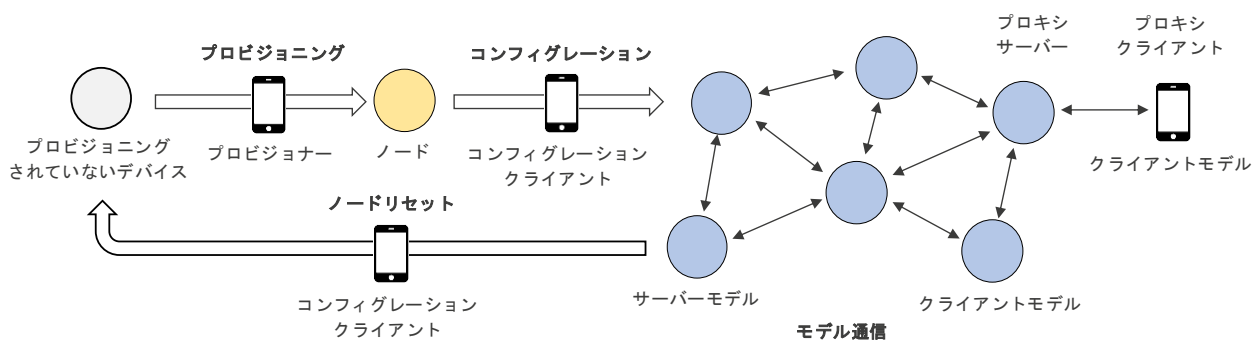


図 2-1 Mesh デモ概要

2.1.1 デモフェーズ 1: プロビジョニング

プロビジョニングされていないデバイス(Unprovisioned Device)は、プロビジョナー(Provisioner)によってプロビジョニングされる必要があります。プロビジョニングとはデバイスを認証し、ユニキャストアドレスやネットワークキーといった Mesh ネットワークでメッセージを送受信するために必要な基本情報を提供するための処理です。なおプロビジョナーは、基本的にはスマートフォンなどのモバイル端末です。

サンプルプログラムは、プロビジョニングされていないデバイス(プロビジョニングサーバー(Provisioning Server))として動作します。電源オン後はプロビジョニングに検知してもらうため、アンプロビジョンドデバイスビーコン(Unprovisioned Device beacon)とコネクタブルアドバタイジングを交互に送信します。

一方、Mesh モバイルアプリケーションはプロビジョナー(プロビジョニングクライアント(Provisioning Client))として動作し、コネクタブルアドバタイジングを受信するとプロビジョニングを実行します。

2.1.2 デモフェーズ 2: コンフィグレーション

プロビジョニング後、プロビジョニングされたデバイスは、ノード(Node)となります。コンフィグレーションクライアント(Configuration Client)はアプリケーションキー、パブリッシュアドレス、サブスクリプションリストといった Mesh モデル通信に必要な設定をコンフィグレーションサーバー(Configuration Server)である各ノードに配布します。なおコンフィグレーションクライアントは、基本的にはスマートフォンまたはモバイル端末です。

サンプルプログラムは、コンフィグレーションサーバーとして動作します。プロビジョニング後、コンフィグレーションクライアントと接続するため、コネクタブルアドバタイジングを送信します。

一方、Mesh モバイルアプリケーションはコンフィグレーションクライアントとして動作し、コネクタブルアドバタイジングを受信するとコンフィグレーションを実行します。

2.1.3 デモフェーズ 3: モデル通信

コンフィグレーション後、各ノードはメッセージをアプリケーションキーで暗号化してパブリッシュアドレス宛に送信します。またサブスクリプションリストに含まれるアドレス宛でのメッセージを受信してアプリケーションキーで復号化します。

サンプルプログラムには Generic OnOff Server モデル、Generic Client モデルに加え、独自の Vendor Server モデル、Vendor Client モデルが実装されています。

開発ボード上のスイッチを押下すると、Generic OnOff Client は On または Off の値を持つ Generic OnOff Set Unacknowledged メッセージを送信します。本メッセージを受信した Generic OnOff Server は、メッセージの値に従って開発ボード上の LED を制御します。

開発ボードと接続した PC 上のコンソールで文字列を入力すると、Vendor Client は文字列を持つ Vendor Set Unacknowledged メッセージを送信します。本メッセージを受信した Vendor Server は、メッセージを持つ文字列を PC 上のコンソールに出力します。

Mesh モバイルアプリケーションは Generic OnOff Client および Vendor Client として動作することができ、各ノードにメッセージを送信します。メッセージの送受信のため、プロキシ(Proxy)ノードと接続を確立します。プロキシノードはメッセージを周囲のノードに転送し、さらに各ノードが中継することで、メッセージは Mesh ネットワーク全体に広がります。

2.1.4 デモフェーズ 4: ノードリムーバル

コンフィグレーションクライアントはノードに Config Node Reset メッセージを送信することで、Mesh ネットワークからノードを除外することができます。

サンプルプログラムは Config Node Reset メッセージを Mesh モバイルアプリケーションから受信すると、コンフィグレーション情報をリセットして再度プロビジョニングを開始します。

3. ハードウェア要件

Bluetooth Mesh スタックパッケージには、次の RX23W 向け開発ボード上で動作するサンプルプログラムが同梱されます。アプリケーション開発の際には 2 台以上の使用を推奨します。

- Target Board for RX23W
- Target Board for RX23W module
- Renesas Solution Starter Kit for RX23W

3.1 Target Board for RX23W

Target Board for RX23W (以後、"TB"と表記)は Bluetooth Mesh スタックを利用したアプリケーション開発に使用できます。また E2 エミュレータ Lite 相当のオンボードデバッグエミュレーション機能を搭載しているため、別途エミュレータを使用することなくソフトウェアを開発することができます。

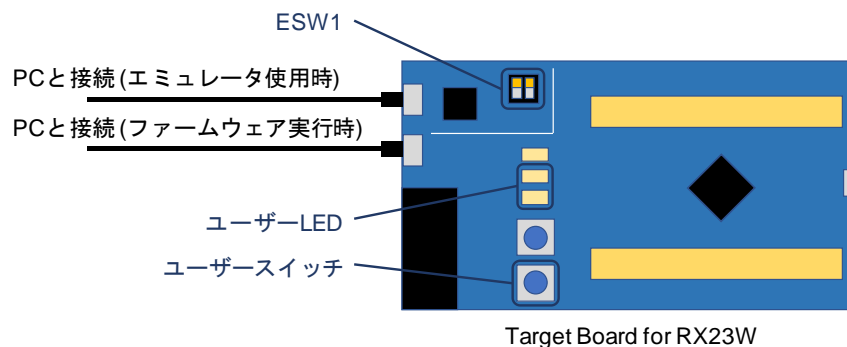


図 3-1 Target Board for RX23W

注: ESW1 スイッチを下記に従って適切に設定する必要があります。

- ファームウェアの書き込みやデバッグの場合: ESW1 の 2-4 を ON
- オンボードエミュレータを使用せずファームウェアを実行する場合: ESW1 の 2-4 を OFF

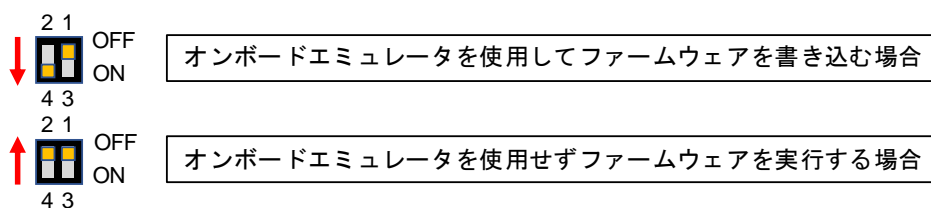


図 3-2 Target Board for RX23W の ESW1 スイッチ設定

TB の詳細は「RX23W グループ Target Board for RX23W ユーザーズマニュアル」(R20UT4634)を参照してください。

3.2 Target Board for RX23W module

Target Board for RX23W と同様、Target Board for RX23W module (以後、"TB module"と表記)は Bluetooth Mesh スタックを利用したアプリケーション開発に使用できます。TB module の詳細は「RX23W グループ Target Board for RX23W module ユーザーズマニュアル」(R20UT4890)を参照してください。

3.3 Renesas Solution Starter Kit for RX23W

Renesas Solution Starter Kit for RX23W (以後、"RSSK"と表記)も Bluetooth Mesh スタックを利用したアプリケーションの開発に使用できます。

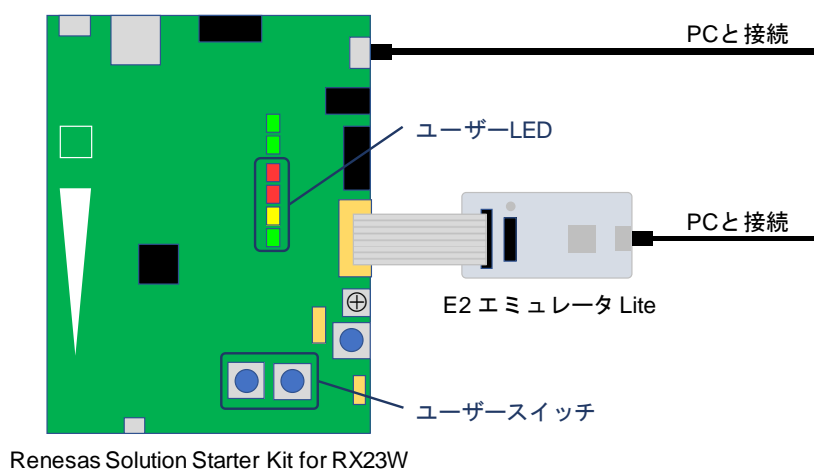


図 3-3 Renesas Solution Starter Kit for RX23W

RSSK のシリアルポートである USBCN0 から電源供給するため、表 3-1 に示したジャンパ設定に変更してください。

表 3-1 Renesas Solution Starter Kit for RX23W のジャンパ設定

ジャンパ	設定
J3	1-2 Short
J4	1-2 Short
J5	1-2 Short
J10	Short
J13	Short
J7	1-2

RSSK の詳細は「RX23W グループ Renesas Solution Starter Kit for RX23W CPU ボード ユーザーズマニュアル」(R20UT4446)を参照してください。

3.4 E2 エミュレータ Lite

[E2 エミュレータ Lite](#) は RX および RL78 ファミリの MCU 上でのリアルタイムデバッグとフラッシュ書き込みに対応したエミュレータです。その他のエミュレータを使用する場合は RX23W への対応を確認してください。

3.5 モバイル端末

基本的にはスマートフォンなどのモバイル端末が、プロビジョナーおよびコンフィグレーションクライアントとして使用されます。またパッケージに含まれるサンプルプログラムはデモ実行のため、Bluetooth Low Energy 通信可能な Android デバイスまたは iOS を必要とします。

4. ソフトウェア要件

4.1 Bluetooth Mesh スタック

Bluetooth Mesh スタック(以後、"Mesh スタック"と表記)は、Mesh ネットワーク上での多対多の無線通信に必要な Bluetooth Mesh 機能をアプリケーションに提供します。Mesh スタックは、[FIT モジュール](#)として提供されます。また Mesh スタックとサンプルプログラムは、Bluetooth Mesh スタックパッケージ(以後、"Mesh パッケージ"と表記)に同梱されています。

4.2 Bluetooth Low Energy プロトコルスタック

Mesh スタックは、Bluetooth Low Energy 技術を利用するために Bluetooth Low Energy プロトコルスタックが必要です。Bluetooth Low Energy スタックは、[FIT モジュール](#)として提供されます。また Mesh パッケージに同梱されています。

4.3 Mesh モバイルアプリケーション

Mesh モバイルアプリケーション(以後、"Mesh モバイル"と表記)は、デモのためのサンプルアプリケーションであり、Android または iOS 上で動作します。Mesh モバイルアプリケーションは下記リンクよりダウンロードしてください。

Android 版: <https://play.google.com/store/apps/details?id=com.renesas.meshmobile>

iOS 版: <https://apps.apple.com/jp/app/renesas-meshmobile/id1641877699>

4.4 e² studio IDE および CC-RX コンパイラ

Mesh パッケージに同梱されるデモプロジェクトは次の IDE で生成されました。

IDE : Renesas Electronics e² studio 2022-10

Mesh スタックライブラリは次のコンパイラでビルドされました。

コンパイラ : Renesas Electronics CC-RX V2.08.01

[e² studio](#) は広く普及している Eclipse CDT (C/C++ Development Tooling) をベースとした開発環境であり、ビルド(エディタ、コンパイラやリンクの制御)やデバッグまで対応します。インストーラーは下記のウェブサイトから入手できます。

e² studio:

<https://www.renesas.com/products/software-tools/tools/ide/e2studio.html#downloads>

さらに [RX ファミリー用 C/C++ コンパイラパッケージ\(CC-RX\)](#) が RX23W 向けファームウェアのビルドに必要です。本コンパイラは e² studio のインストール中にインストールされます。コンパイラの使用には、コンパイラライセンスが必要です。詳細は下記のウェブサイトを参照してください。

コンパイラライセンス:

<https://www.renesas.com/products/software-tools/tools/compiler-assembler/compiler-licenses.html>

4.5 Renesas Flash Programmer

[Renesas Flash Programmer](#)(以後、"RFP"と表記)は開発や量産においてルネサス MCU のフラッシュメモリにファームウェアを書き込むためのツールです。インストーラーは下記のウェブサイトから入手できます。

Renesas Flash Programmer (Programming GUI):

<https://www.renesas.com/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html#downloads>

4.6 サードパーティー ソフトウェア

4.6.1 シリアルターミナルツール

サンプルプログラムは、UART 経由でログメッセージを出力します。ログメッセージの確認には、ANSI エスケープシーケンスの CSI(Control Sequence Introducer)に対応したシリアルターミナルツールを使用してください。

[Tera Term](#)

なおサンプルプログラムが出力するログメッセージの例は 8.1 節を参照してください。

4.6.2 Mesh モバイルのビルドに必要な開発ツール

Mesh モバイルのビルドには次の開発ツールが必要です。これらはインターネット上で入手できます。Mesh モバイルのビルド方法の詳細は 6.3 節を参照してください。

npm (Node Package Manager), [Node.js](#) に同梱

[Python](#)

[Capacitor](#)

[Ionic framework](#)

[Android Studio](#), Android 向け Mesh モバイルのビルド

[Xcode](#), iOS 向け Mesh モバイルのビルド

注: Xcode は Mac PC 上でのみ動作します。Xcode をダウンロードするためには、Apple ID が必要です。また iOS アプリケーションを開発するには有償のライセンスが必要です。

[Apple Developer Program](#) : App Store で配布するアプリケーション向け

[Apple Developer Enterprise Program](#) : インハウスアプリケーション向け

5. Bluetooth Mesh スタックパッケージ

5.1 構成

Mesh パッケージ(r01an4930xx0130-rx23w-blemesh-fit.zip)の構成を以下に示します。

r01an4930xx0130-rx23w-blemesh-fit.zip	
blemesh_api.chm	: Mesh スタック API マニュアル
r01an4874ej0130-blemesh.pdf	: Mesh スタートアップガイド(英)
r01an4874jj0130-blemesh.pdf	: Mesh スタートアップガイド(日)
r01an4875ej0130-blemesh.pdf	: Mesh 開発ガイド(英)
r01an4875jj0130-blemesh.pdf	: Mesh 開発ガイド(日)
r01an4930ej0130-rx23w-blemesh.pdf	: Mesh FIT モジュール文書(英)
r01an4930jj0130-rx23w-blemesh.pdf	: Mesh FIT モジュール文書(日)
readme.txt	: Mesh スタックパッケージ情報ファイル
	:
+---FITDemos\	: ---- Mesh FIT デモプロジェクト ----
make_workspace_rsskrx23w.bat	: ワークスペース作成バッチファイル(RSSK)
make_workspace_tbrx23w.bat	: ワークスペース作成バッチファイル(TB)
rsskrx23w_mesh_cli.zip	: Command Line Interface プログラム(RSSK)
rsskrx23w_mesh_client.zip	: Client Models サンプルプログラム(RSSK)
rsskrx23w_mesh_server.zip	: Server Models サンプルプログラム(RSSK)
tbrx23w_mesh_cli.zip	: Command Line Interface プログラム(TB)
tbrx23w_mesh_client.zip	: Client Models サンプルプログラム(TB)
tbrx23w_mesh_server.zip	: Server Models サンプルプログラム(TB)
tbrx23wmodule_mesh_cli.zip	: Command Line Interface プログラム(TB module)
tbrx23wmodule_mesh_client.zip	: Client Models サンプルプログラム(TB module)
tbrx23wmodule_mesh_server.zip	: Server Models サンプルプログラム(TB module)
	:
+---ROM_Files\	: ---- ビルド済みファームウェア ----
rsskrx23w_mesh_cli.mot	: Command Line Interface プログラム(RSSK)
rsskrx23w_mesh_client.mot	: Client Models サンプルプログラム(RSSK)
rsskrx23w_mesh_server.mot	: Server Models サンプルプログラム(RSSK)
tbrx23w_mesh_cli.mot	: Command Line Interface プログラム(TB)
tbrx23w_mesh_client.mot	: Client Models サンプルプログラム(TB)
tbrx23w_mesh_server.mot	: Server Models サンプルプログラム(TB)
tbrx23wmodule_mesh_cli.mot	: Command Line Interface プログラム(TB module)
tbrx23wmodule_mesh_client.mot	: Client Models サンプルプログラム(TB module)
tbrx23wmodule_mesh_server.mot	: Server Models サンプルプログラム(TB module)
	:
+---FITModules\	: ---- Mesh FIT モジュール ----
r_mesh_rx23w_v1.30.xml	: プラグインファイル
r_mesh_rx23w_v1.30.zip	: Mesh FIT モジュール
r_mesh_rx23w_v1.30_extend.mdf	: 設定記述ファイル
	:
+---mesh_mobile\	: ---- メッシュモバイルアプリケーション ----

本書では Client Models サンプルプログラムと Server Models サンプルプログラムを使用します。Command Line Interface プログラムについては「RX23W グループ Bluetooth Mesh スタック 開発ガイド」(R01AN4875)の 4.1 節を参照してください。

Mesh FIT モジュール(r_mesh_rx23w_v1.30.zip)の構成を以下に示します。

r_mesh_rx23w_v1.30.zip	
+---r_config\	: ---- FIT 設定 ----
r_mesh_rx23w_config.h	: Mesh FIT 設定ファイル
	:
+---r_mesh_rx23w\	: ---- Mesh FIT モジュール ----
r_mesh_rx23w_if.h	: Mesh スタック API ヘッダファイル
readme.txt	: Mesh FIT モジュール情報ファイル
	:
+---doc\	:
blemesh_api.chm	: Mesh スタック API マニュアル
+---en\	:
r01an4930ej0130-rx23w-blemesh.pdf	: Mesh FIT モジュール説明書(英)
+---ja\	:
r01an4930jj0130-rx23w-blemesh.pdf	: Mesh FIT モジュール説明書(日)
	:
+---json	:
mesh_provisioning.service.json	: QE for BLE 向けサービス定義ファイル
mesh_proxy.service.json	:
	:
+---lib\	:
lib_ble_ms_ccrx.lib	: Mesh スタックライブラリ
	:
+---src\	:
+---bearer\	: Bluetooth ベアラー
+---drivers\	: Mesh ドライバ
+---include\	: Mesh スタックヘッダファイル

Mesh FIT モジュールの設定については「RX23W グループ BLE モジュール Firmware Integration Technology」(R01AN4930)の 3.1 節を参照してください。

6. 開発環境の構築

本章では Bluetooth Mesh スタックを使用したアプリケーション開発環境の構築方法を示します。

手早くデモを実行する必要がある場合、本章はスキップして7章に進んでください。パッケージに同梱されたファームウェアを使用することでデモを実行することができます。

6.1 e² studio IDE および CC-RX コンパイラ

e² studio のインストーラーを起動し、インストールウィザードに従って処理を進めます。インストールの処理中、サポートするデバイス・ファミリーとして RX デバイスを選択してください。次に CC-RX はデフォルトでインストールされるため、オプション・コンポーネントを変更する必要はありません。

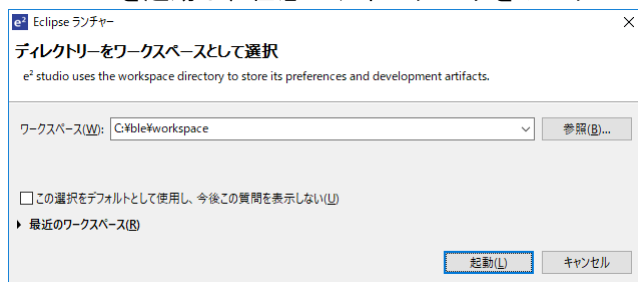
注: インストールディレクトリ名とそのパスにはスペースや多バイト文字、\$,#,%などの記号を含めるべきではありません。これらの文字はインストールおよびビルド、デバッグ処理を妨げる可能性があります。同様にワークスペース名やそのパスにもこれらの文字は含めるべきではありません。

注: ワークスペースディレクトリ名とそのパスはできる限り短くしてください。長いワークスペースパスは、そのワークスペースにあるソースコードのコンパイルやリンク処理を妨げる可能性があります。

6.2 デモプロジェクト

以下の手順を実行します。

1. e² studio を起動し、任意のディレクトリをワークスペースとして指定します。



2. ワークスペースにデモプロジェクトをインポートします。

TB を使用する場合は FITDemos ディレクトリの下記のアーカイブ・ファイルをインポートします。

```
FITDemos\
+---tbrx23w_mesh_client.zip      : Client Models プロジェクト
+---tbrx23w_mesh_server.zip      : Server Models プロジェクト
+---tbrx23w_mesh_cli.zip         : Command Line Interface(CLI)プロジェクト
```

TB module を使用する場合は FITDemos ディレクトリの下記のアーカイブ・ファイルをインポートします。

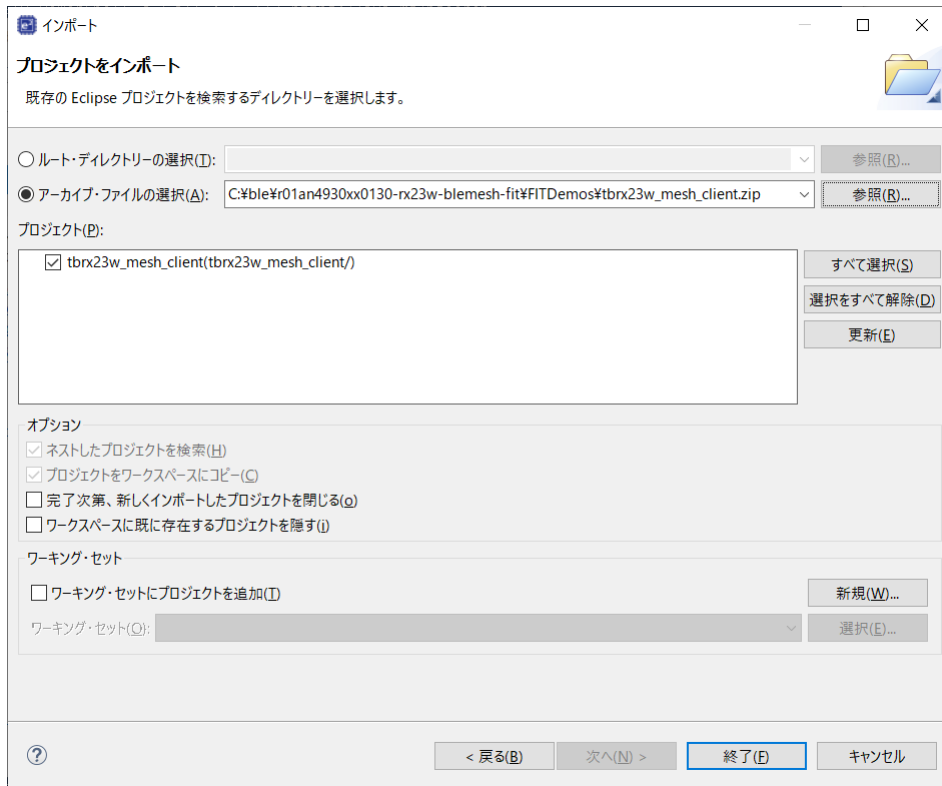
```
FITDemos\
+---tbrx23wmodule_mesh_client.zip : Client Models プロジェクト
+---tbrx23wmodule_mesh_server.zip : Server Models プロジェクト
+---tbrx23wmodule_mesh_cli.zip    : Command Line Interface(CLI)プロジェクト
```

RSSK を使用する場合は FITDemos ディレクトリの下記のアーカイブ・ファイルをインポートします。

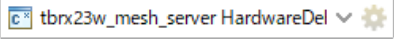
```
FITDemos\
+---rsskrx23w_mesh_client.zip     : Client Models プロジェクト
+---rsskrx23w_mesh_server.zip     : Server Models プロジェクト
+---rsskrx23w_mesh_cli.zip        : Command Line Interface(CLI)プロジェクト
```


メニューの[ファイル]→[インポート...]を選択してインポートダイアログを開きます。

インポートダイアログの[アーカイブ・ファイルの選択]で FITDemos ディレクトリの Client Models プロジェクト(mesh_client)を選択して、終了ボタンをクリックします。同様の手順で Server Models プロジェクト(mesh_server)もインポートします。




3. ビルドアイコン  をクリックしてプロジェクトをビルドします。


現在のプロジェクトは Launch Configuration  で確認と変更ができます。

ビルド後、ファームウェア(.mot ファイル)がプロジェクトディレクトリの HardwareDebug\内に生成されます。

4. 開発ボードと E2 エミュレータ Lite を PC に接続し、エミュレータとボードを接続します。

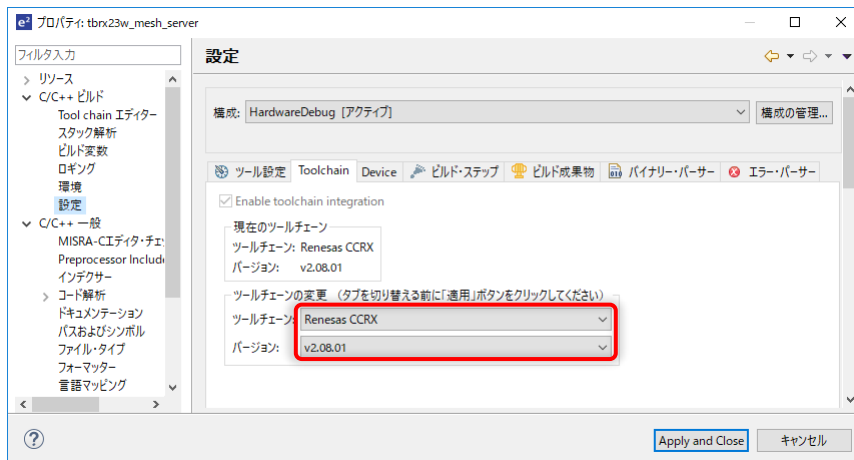
5. デバッグアイコン  をクリックしてデバッグモードでプロジェクトを起動します。
プロジェクトの起動後、ファームウェアが開発ボードにダウンロードされます。

6. デバッグパースペクティブの再開アイコン  をクリックしてプロジェクトを実行します。

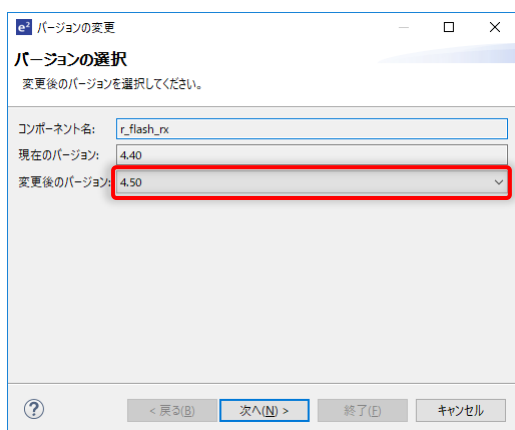
7. プロジェクトのデバッグ後は、デバッグパースペクティブの終了アイコン  をクリックします。
プロジェクトのファームウェアは終了と電源オフ後も RX23W のフラッシュメモリに保持されます。

7章に示されるデモを実行するため2台以上の開発ボードを使用することを推奨します（少なくとも1台が Client、その他が Server）。7章に進む前に Server のファームウェアと Client のファームウェアを、上記の手順5と7を実行または RFP を使用することで開発ボードに書き込んでください。

注: 「No toolchain set or toolchain not integrated.」とエラーが発生し、ビルドに失敗する場合は、プロジェクトのプロパティダイアログを開き、[C/C++ ビルド]→[設定]→[Toolchain]タブに移動して、ツールチェーンを選択してください。



注: プロジェクトには FIT モジュールから生成されたソースコードが含まれます。スマート・コンフィグレータを使用することでソースコードを再度生成できます。スマート・コンフィグレータを開いた際に「W04020001: モジュールが見つからないため、コンフィグレーション r_lpc_rx は削除されました。」といったエラーが発生する場合は、最新の FIT モジュールをダウンロード後、スマート・コンフィグレータの各コンポーネントの右クリックメニューで[バージョンの変更]を選択し、ダウンロードしたバージョンを選択してください。



6.3 Mesh モバイル

Mesh モバイルのビルドとインストール方法については mesh_mobile\mesh_mobile_guide.pdf を参照してください。

7. デモの実行

本章は Mesh デモを実行するためのサンプルプログラムと Mesh モバイルの操作方法を示します。本デモでは 2 台以上の開発ボードを使用することを推奨します（少なくとも 1 台が Client、その他が Server）。

もし 6 章をスキップした場合は RFP を使用して開発ボードにファームウェアを書き込みます。デモのためのファームウェアは Mesh パッケージに同梱されています。

```
FITDemos\ROM_Files\rsskrx23w_mesh_client.mot
FITDemos\ROM_Files\rsskrx23w_mesh_server.mot
FITDemos\ROM_Files\tbrx23w_mesh_client.mot
FITDemos\ROM_Files\tbrx23w_mesh_server.mot
FITDemos\ROM_Files\tbrx23wmodule_mesh_client.mot
FITDemos\ROM_Files\tbrx23wmodule_mesh_server.mot
```

Mesh モバイルをインストールしていない場合は以下の手順を実行します。

注: 本パッケージには Android デバイスのアプリケーションパッケージのみ同梱されます。iOS デバイスを使用する場合は、6.3 節に従って Mesh モバイルのビルドとインストールを行ってください。

1. 次のパッケージファイルを USB 経由で PC から Android にコピーします。

```
mesh_mobile\android-debug.apk
```

2. Android 上の任意のファイルマネージャアプリケーションでパッケージファイルを実行します。

また次の設定でターミナルツールを使用することで、開発ボードが出力するログメッセージを確認することができます。

表 7-1 シリアルポート設定

項目	設定
ボーレート	115200 bps
データ	8 bits
パリティ	なし
ストップ	1 bit
フロー制御	なし

7.1 フェーズ 1: プロビジョニング

デモを開始するため、全ての開発ボードに電源を供給し、Mesh モバイルを起動します。

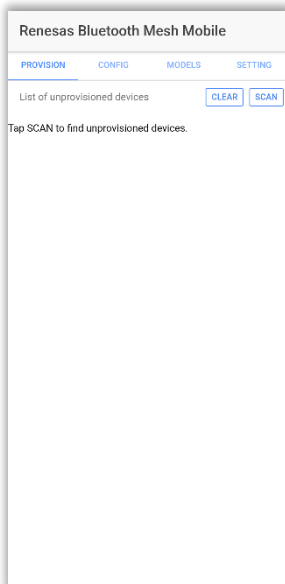
注: Android 向け Mesh モバイルは次のパーミッション設定が必要です。: 位置情報、付近のデバイス、ストレージ

注: Android 向け Mesh モバイルで SCAN 動作に失敗する場合、上記のパーミッションをいったん削除して再設定してください。

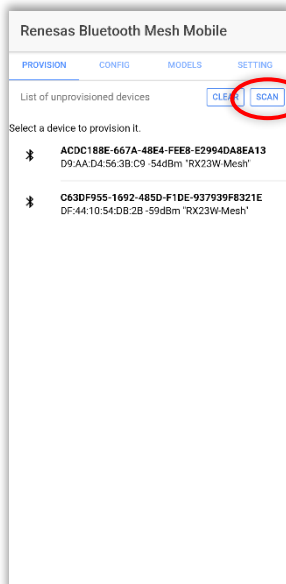
全ての開発ボードをプロビジョニングするため次の手順を繰り返します。

1. プロビジョニングされていないデバイスを探るため、PROVISION タブに移動して SCAN ボタンをタップします。
2. 探索結果からプロビジョニングする任意のデバイスを選択します。
3. 接続の確立後、プロビジョニングが実行されます。
4. デバイスが OOB Public Key や OOB Authentication に対応している場合、Mesh モバイルは OOB Public Key の使用有無と OOB Authentication のいずれかを選択する画面を表示します。デモでは手順の簡略化のため、「No OOB Public Key is used」と「No OOB Authentication is used」を選択し、PROVISIONING START ボタンをタップします。

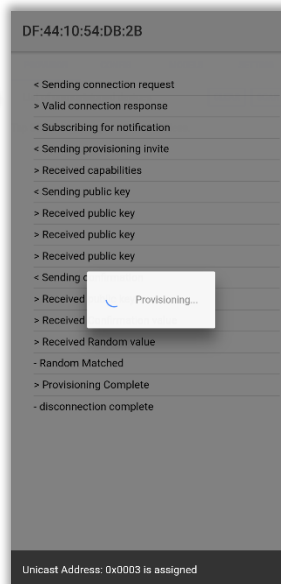
手順 1



手順 2



手順 3



手順 4

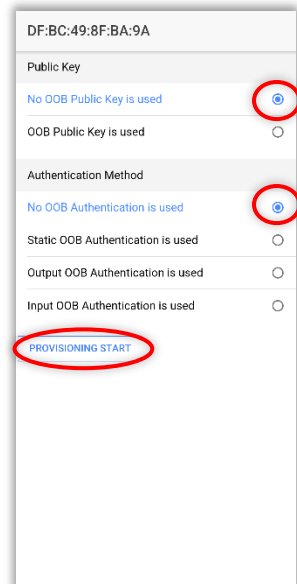


図 7-1 プロビジョニング

7.2 フェーズ 2: コンフィグレーション

全ボードのプロビジョニング後、各ボードをコンフィグレーションするため次の手順を繰り返します。

1. ノードを探索するため、CONFIG タブに移動し SCAN ボタンをタップします。
2. 接続可能なノードは緑色で表示されます。接続を確立してコンフィグレーションを実行するため、緑色で表示された任意のノードを選択します。

注: ノードのコネクタブルアドバタイジング送信は 60 秒で停止します。コンフィグレーションすべきノードが緑色表示とならない場合は、開発ボードの SW1 を 2 秒以上長押しして、コネクタブルアドバタイジング送信を再開してください。

3. 構成情報の表示後、CONFIGURATION タブに移動します。

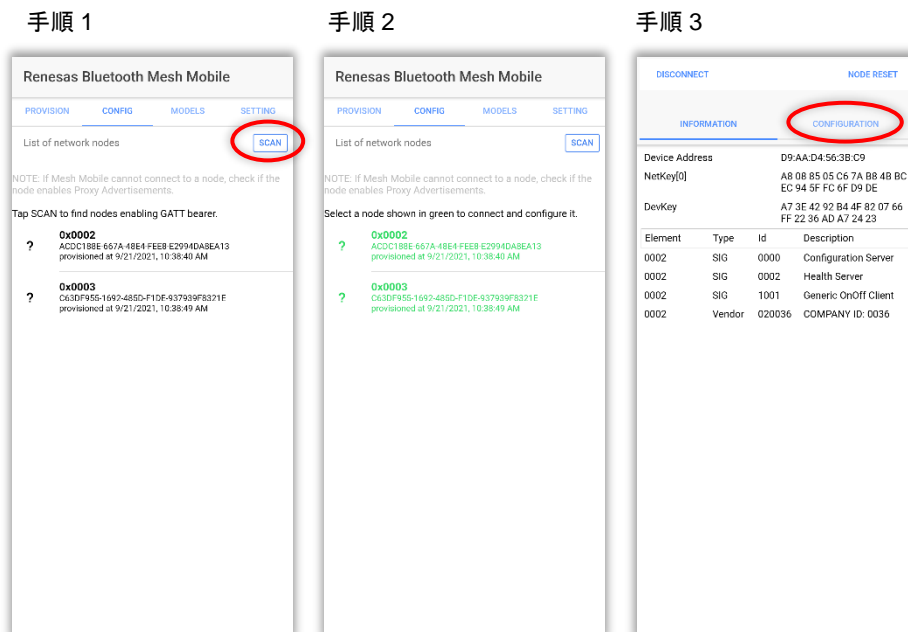


図 7-2 コンフィグレーション (1/2)

4. CONFIGURATION タブ上で Relay、Proxy、Friend を有効化し、"Demo"グループを選択します。
5. APPLY ボタンをタップします。
6. コンフィグレーションの完了後、DISCONNECT ボタンをタップします。

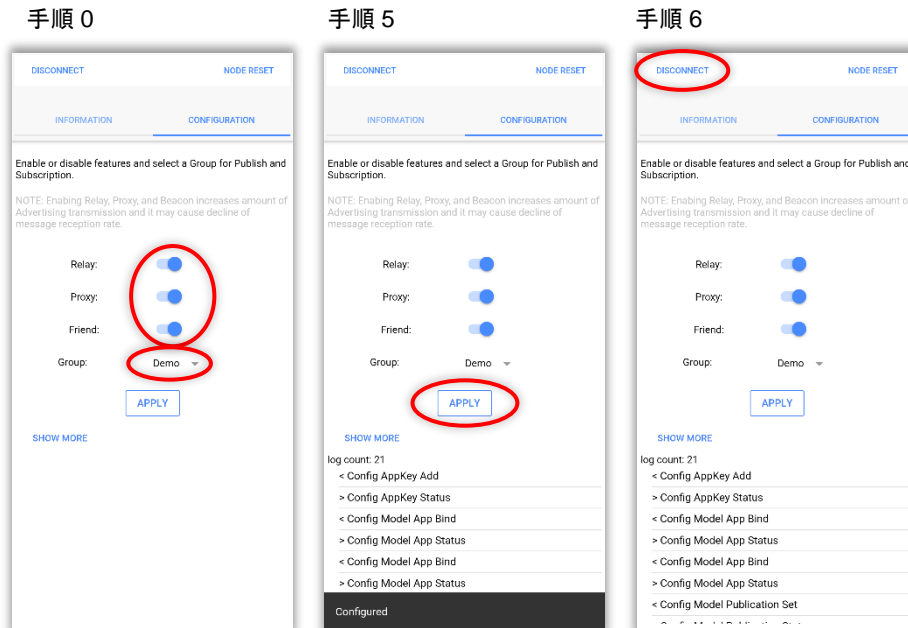


図 7-3 コンフィグレーション (2/2)

コンフィグレーションで選択したグループは複数のノードをグループ化して操作する際に使用されます。下記の手順により任意のグループを追加することができます。

1. MODELS タブに移動し ADDGROUP ボタンをタップします。
2. Add Group ダイアログ上で "Kitchen" のような任意のグループ名を入力します。
3. グループが追加されたことを確認します。

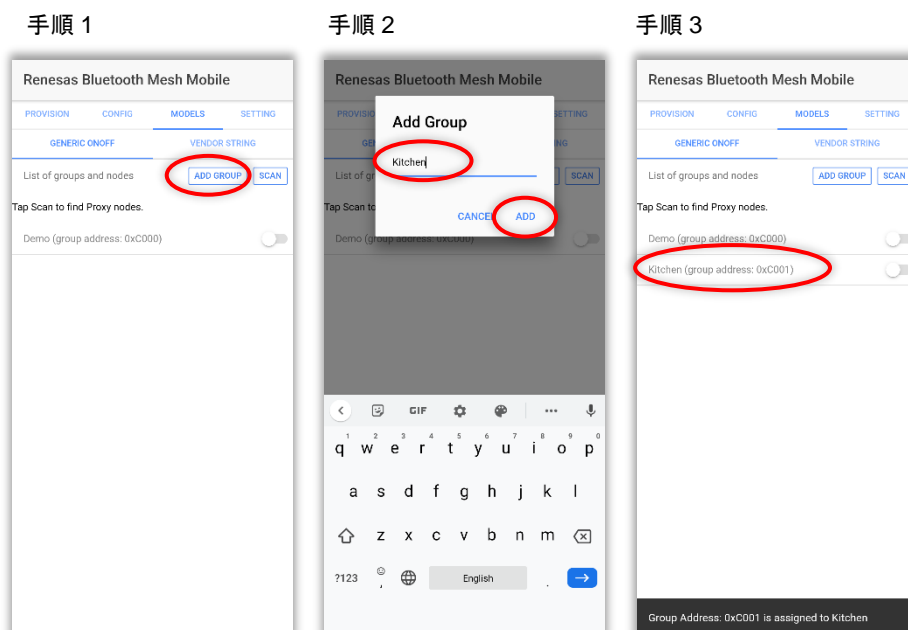


図 7-4 グループ追加

7.3 フェーズ 3: モデル通信

開発ボードのコンフィグレーション後、Mesh モバイルおよび開発ボードはモデルメッセージを送信できます。まずは Mesh モバイルが開発ボードの一つとプロキシ接続を確立するため、次の手順を実行してください。Mesh モバイルはプロキシ接続したノードを通して、任意のノードにメッセージを送信します。

1. プロキシノードを探索するため、MODELS タブに遷移し SCAN ボタンをタップします。
2. プロキシノードと接続を確立するため、緑色に表示された任意のノードを選択します。接続が完了すると、ノードは青色で表示されます。
3. 青色で表示されたノードを再度タップすると、ノードとの接続を切断することができます。

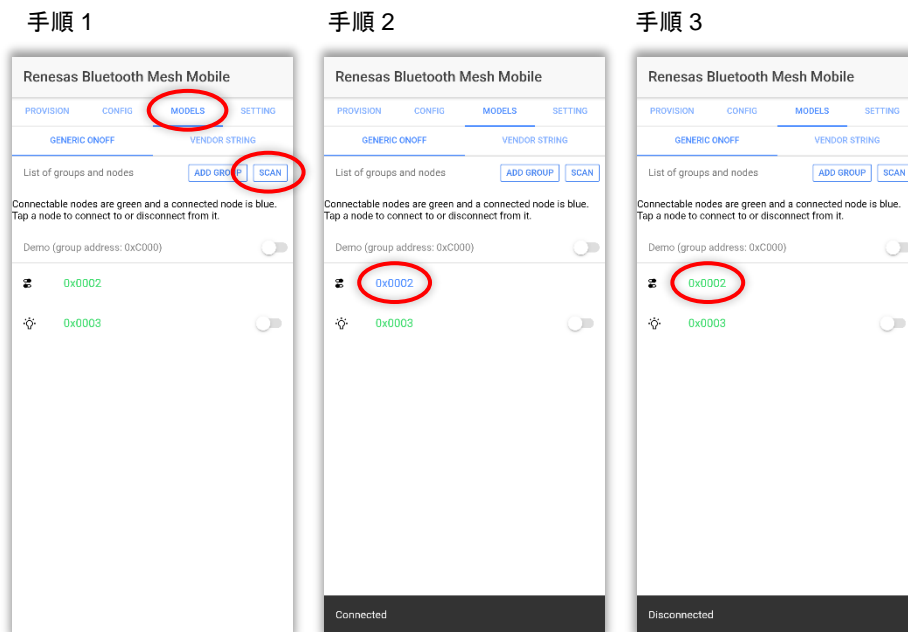


図 7-5 プロキシ接続

7.3.1 Mesh モバイルによる Generic OnOff ステート制御

1. MODELS タブの GENERIC ONOFF タブ上で、グループのトグルボタンを繰り返しタップします。Generic OnOff Server として動作する全ての開発ボードの LED が、受信した Generic OnOff SET メッセージに連動して変化することを確認できます。

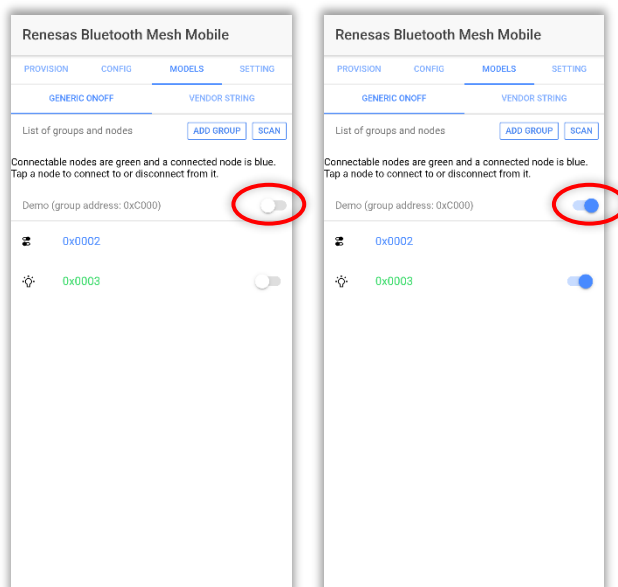


図 7-6 グループアドレスへの Generic OnOff SET メッセージ発行

2. MODELS タブの GENERIC ONOFF タブ上で、任意の開発ボードのトグルボタンを繰り返しタップします。Generic OnOff Server として動作する任意の開発ボードの LED が、受信した Generic OnOff SET メッセージに連動して変化することを確認できます。

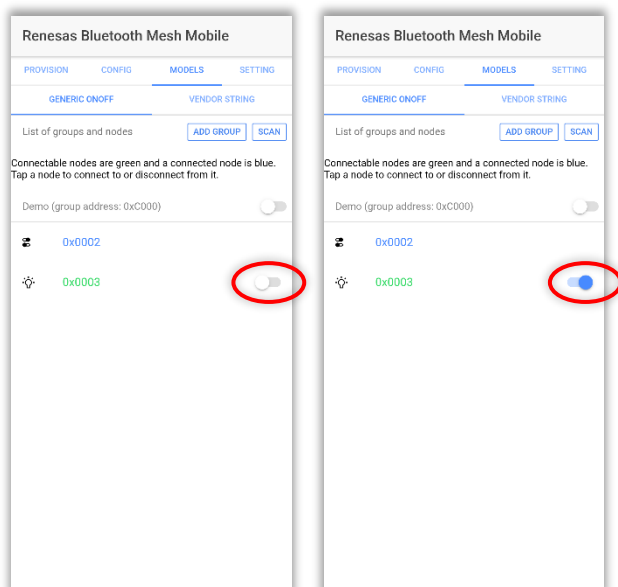


図 7-7 ユニキャストアドレスへの Generic OnOff SET メッセージ発行

7.3.2 開発ボードによる Generic OnOff ステート制御

1. Generic OnOff Client として動作する開発ボードの SW1 を繰り返し押し下します。
Generic OnOff Server として動作する全ての開発ボードの LED が、受信した Generic OnOff SET メッセージに連動して変化することを確認できます。

7.3.3 Mesh モバイルによる Vendor ステート制御

1. MODELS の VENDOR STRING タブ上でグループの STRING ボタンをタップします。表示されたダイアログに任意の文字列を入力後、SEND ボタンをタップしてください。

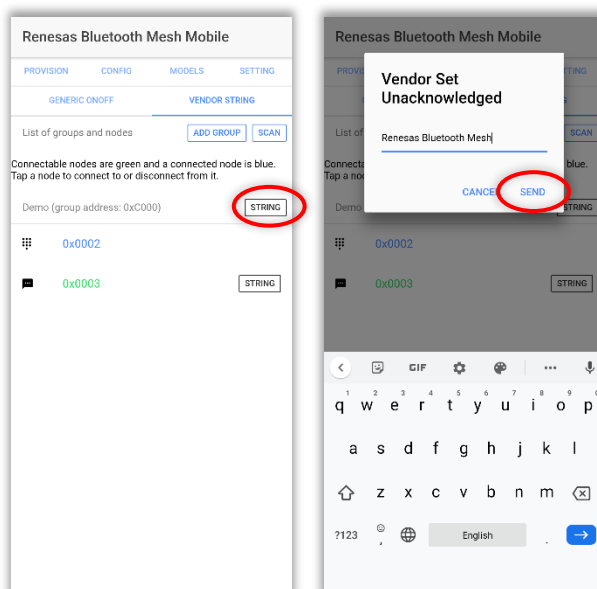


図 7-8 グループアドレスへの Vendor SET メッセージ発行

Vendor Server として動作する全ての開発ボードが、受信した Vendor SET メッセージに含まれる文字列をコンソールに出力することを確認できます。

```
Vendor Set src: 0x7F00 dst: 0xC000 len = 23 value: "Renesas bluetooth mesh"
```

図 7-9 Vendor Server ノードのコンソールログ

- MODELS の VENDOR STRING タブ上で、任意の開発ボードの STRING ボタンをタップします。表示されたダイアログに任意の文字列を入力後、SEND ボタンをタップしてください。

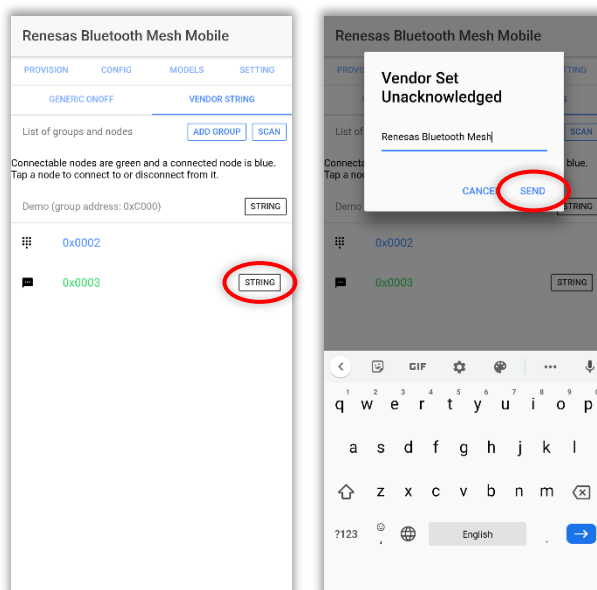


図 7-10 ユニキャストアドレスへの Vendor SET メッセージ発行

Vendor Server として動作する任意の開発ボードが、受信した Vendor SET メッセージに含まれる文字列をコンソールに出力することを確認できます。

```
Vendor Set src: 0x7F00 dst: 0x0003 len = 23 value: "Renesas bluetooth mesh"
```

図 7-11 Vendor Server ノードのコンソールログ

7.3.4 開発ボードによる Vendor ステート制御

- Vendor Client として動作する開発ボードのコンソールに文字列を入力後、Enter キーを押下します。

```
Renesas Bluetooth Mesh
MS_vendor_set_unack() status:0x0000
```

図 7-12 Vendor Client のコンソールへの文字列入力

Vendor Server として動作する任意の開発ボードが、受信した Vendor SET メッセージに含まれる文字列をコンソールに出力することを確認できます。

```
Vendor Set src: 0x0004 dst: 0xC000 len = 23 value: "Renesas bluetooth mesh"
```

図 7-13 Vendor Server ノードのコンソールログ

7.3.5 Mesh モバイルによるノードリセット

1. CONFIG タブでノードと接続後、RESET ボタンをタップします。表示されたダイアログ上の SEND ボタンをタップしてください。

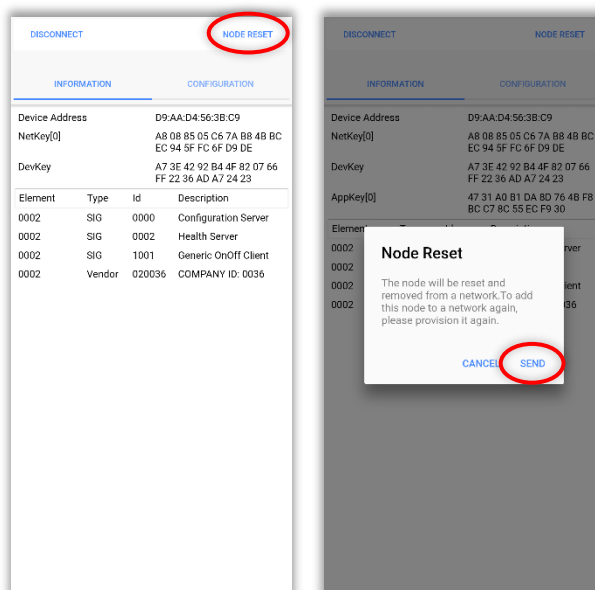


図 7-14 ノードへの Config Node Reset メッセージ発行

Mesh モバイルとプロキシ接続していた開発ボードが、Config Node Reset メッセージを受信することでプロビジョニングを再度実行することを確認できます。

```
MS_ACCESS_CONFIG_NODE_RESET_OPCODE
This node was removed from the mesh network
MS_PROXY_DOWN_EVENT
BLEBRR_GATT_IFACE_DOWN
    Device Address: 75:56:76:B5:7A:89 rnd
MS_prov_register() status:0x0000
MS_prov_setup() status:0x0000
    Role : UNPROVISIONED DEVICE
    Bearer: BOTH
MS_prov_bind() status:0x0000
    Device UUID: 5C6E7883-8FB0-DE1A-1941-212A2A4F9AF1
```

図 7-15 Config Node Reset メッセージを受信したノードのコンソールログ

8. Appendix

8.1 ログメッセージ

サンプルプログラムが出力するログメッセージの例を示します。ターミナルツールのシリアルポート設定については7章の表 7-1 を参照してください。

```

Bearer Initialization Completed
  Device Address: DF:BC:49:8F:BA:9A rnd
MS_init_ext() status:0x0000
MS_access_create_node() status:0x0000
MS_access_register_element_ext() status:0x0000
MS_config_server_init() status:0x0000
MS_health_server_init() status:0x0000
MS_generic_onoff_server_init() status:0x0000
MS_vendor_server_init() status:0x0000
MS_prov_register() status:0x0000
MS_prov_setup() status:0x0000
  Role : UNPROVISIONED DEVICE
  Bearer: BOTH
MS_prov_bind() status:0x0000
  Device UUID: 5C6E7883-8FB0-DE1A-1941-212A2A4F9AF1
BLEBRR_GATT_IFACE_UP
  Device Address: 53:8D:18:43:7F:4B rnd
BLEBRR_GATT_IFACE_ENABLE
  Device Address: 53:8D:18:43:7F:4B rnd
MS_prov_bind() status:0x0000
  Device UUID: 5C6E7883-8FB0-DE1A-1941-212A2A4F9AF1
PROV_EVT_PROVISIONING_SETUP status:0x0000
PROV_EVT_PROVDATA_INFO status:0x0000
  Unicast Address: 0x005C
  IV Index: 0x00000001
  Flags: 0x00
MS_access_cm_set_prov_data() status:0x0000
PROV_EVT_PROVISIONING_COMPLETE status:0x0000
MS_access_cm_get_netkey() status:0x0000
  NetKey: 16bytes:741AC0FCA1EBCD68476C66E3A8CA99A5
MS_access_cm_get_device_key() status:0x0000
  DevKey: 16bytes:2B951BEE7CEB5AEFEB5F2FE5E581CB15
BLEBRR_GATT_IFACE_DOWN
  Device Address: 53:8D:18:43:7F:4B rnd

MS_proxy_register() status:0x0000
MS_proxy_server_adv_start() status:0x0000
  Identification Type: Node Identity
Advertising with Node Identity is limited to 60 seconds.
To perform Advertising with Node Identity again, keep pressing SW1 for 2 seconds.
BLEBRR_GATT_IFACE_UP
  Device Address: 53:8D:18:43:7F:4B rnd
MS_PROXY_UP_EVENT
MS_net_broadcast_secure_beacon() status:0x0000
  subnet_handle: 0
BLEBRR_GATT_IFACE_ENABLE
  Device Address: 53:8D:18:43:7F:4B rnd
MS_ACCESS_CONFIG_APPKEY_GET_OPCODE
  2bytes:0000
MS_ACCESS_CONFIG_RELAY_GET_OPCODE
MS_ACCESS_CONFIG_FRIEND_GET_OPCODE
MS_ACCESS_CONFIG_GATT_PROXY_GET_OPCODE
MS_ACCESS_CONFIG_APPKEY_ADD_OPCODE
  19bytes:000000BB1985565CE4870CE78374E6D095277C
MS_ACCESS_CONFIG_MODEL_APP_BIND_OPCODE
  6bytes:5C0000000010

```

プロビジョニングの開始

プロビジョニングの完了

プロキシ接続の確立

コンフィグレーションの開始

```
MS_ACCESS_CONFIG_MODEL_APP_BIND_OPCODE
  8bytes:5C00000036000100
MS_ACCESS_CONFIG_MODEL_PUBLICATION_SET_OPCODE
  11bytes:5C0000C000007F00000010
MS_ACCESS_CONFIG_MODEL_PUBLICATION_SET_OPCODE
  13bytes:5C0000C000007F000036000100
MS_ACCESS_CONFIG_MODEL_SUBSCRIPTION_ADD_OPCODE
  6bytes:5C0000C00010
MS_ACCESS_CONFIG_MODEL_SUBSCRIPTION_ADD_OPCODE
  8bytes:5C0000C036000100
MS_ACCESS_CONFIG_RELAY_SET_OPCODE
  2bytes:0148
MS_ACCESS_CONFIG_FRIEND_SET_OPCODE
  1bytes:00
MS_ACCESS_CONFIG_GATT_PROXY_SET_OPCODE
  1bytes:01
MS_PROXY_DOWN_EVENT
MS_proxy_server_adv_start() status:0x0000
  Identification Type: Network ID
BLEBRR_GATT_IFACE_DOWN
  Device Address: 53:8D:18:43:7F:4B rnd

BLEBRR_GATT_IFACE_UP
  Device Address: 53:8D:18:43:7F:4B rnd
MS_PROXY_UP_EVENT
MS_net_broadcast_secure_beacon() status:0x0000
  subnet_handle: 0
BLEBRR_GATT_IFACE_ENABLE
  Device Address: 53:8D:18:43:7F:4B rnd
Generic OnOff Set src: 0x7F00 dst: 0xC000 tid: 0x00 state: ON
MS_generic_onoff_server_state_update() status:0x0000
Generic OnOff Set src: 0x7F00 dst: 0xC000 tid: 0x01 state: OFF
MS_generic_onoff_server_state_update() status:0x0000

Vendor Set src: 0x7F00 dst: 0xC000 len = 36 value: "Renesas bluetooth mesh vendor model"
MS_vendor_server_state_update() status:0x0000
```

コンフィギュレーションの完了

Generic OnOff Server モデル

Vendor Server モデル

商標権および著作権

Bluetooth® のワードマークおよびロゴは Bluetooth SIG, Inc が所有する登録商標であり、ルネサスエレクトロニクス株式会社はこれらのマークをライセンスに基づいて使用しています。その他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。

RX23W グループ Bluetooth Mesh スタックは次のオープンソースソフトウェアを使用します。

- [crackle](#); AES-CCM, AES-128bit 機能
BSD 2-Clause License

Copyright (c) 2013-2018, Mike Ryan
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

改訂記録

Rev.	発行日	改定内容	
1.00	2019.09.30	-	初版発行
1.01	2019.11.29	P.4	1章「特徴」を追加
		P.27	8章「Appendix」を追加
1.10	2020.09.30	P.6	1.3節「Mesh デバイスのライフサイクル」を追加
		P.24	7.3.3項「Mesh モバイルによる Vendor ステート制御」を追加
		P.25	7.3.4項「開発ボードによる Vendor ステート制御」を追加
		P.26	7.3.5項「Mesh モバイルによるノードリセット」を追加
		P.27	8.1節「ログメッセージ」のログを更新
		全体	一部の記述、図、スクリーンショットを更新
1.20	2021.09.30	P.13	5.1節「構成」の Mesh パッケージ構成と Mesh FIT モジュール構成を更新
		P.18	7章「デモの実行」に記載されたデモ手順を更新
		全体	一部の記述、図、スクリーンショットを更新
1.30	2022.12.22	P.1	関連文書に「Target Board for RX23W module ユーザーズマニュアル」を追加
		P.9	3章「ハードウェア要件」に Target Board for RX23W module を追加
		P.9	3.2節「Target Board for RX23W module」を追加
		P.10	3.3節「Renesas Solution Starter Kit for RX23W」に表 3-1「Renesas Solution Starter Kit for RX23W のジャンパ設定」を追加
		P.12	4.4節「e ² studio IDE および CC-RX コンパイラ」の IDE バージョンを更新
		P.14	5.1節「構成」の Mesh パッケージ構成と FIT モジュール構成を更新
		P.16	6.2節「デモプロジェクト」の手順 1 と手順 2 を更新
		P.19	7章「デモの実行」のコンソールログを更新
		P.28	8.1節「ログメッセージ」のコンソールログを更新

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、リセットを解除してください。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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ 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えば、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、ノイズ放射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。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まれます。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）、交通管制（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た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なります。
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（豊洲フォレストシア）

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。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。