RX23W Group Bluetooth Mesh Stack

Startup Guide

Introduction

Bluetooth Mesh Stack is the software library to build a mesh network that is compliant with Bluetooth Mesh Networking Specification and to perform many-to-many wireless communication.

This document describes how to get started with Bluetooth Mesh Stack Package. For more information on the software architecture and its layers of the Bluetooth Mesh Stack Package and how to develop a mesh application, refer to "RX23W Group Bluetooth Mesh Stack Development Guide" (R01AN4875).

Target Device

RX23W Group

Related Documents

The following documents are published on Renesas website.

Document Title	Document No.
RX23W Group User’s Manual: Hardware	R01UH0823
CC-RX Compiler User's Manual	R20UT3248
Bluetooth Low Energy Protocol Stack Basic Package User's Manual	R01UW0205
RX23W Group Bluetooth Mesh Stack Startup Guide	R01AN4874 This document
RX23W Group Bluetooth Mesh Stack Development Guide	R01AN4875
RX23W Group Target Board for RX23W User's Manual	R20UT4634
RX23W Group Target Board for RX23W module User's Manual	R20UT4890
RX23W Group Renesas Solution Stater Kit for RX23W CPU Board User's Manual	R20UT4446

Contents

1. Features

This chapter introduces the features of Bluetooth Mesh Network implemented by the Bluetooth Mesh Stack software.

1.1 Communication Topology

Communication topologies provided by Bluetooth Low Energy technology are Point-to-Point and Broadcast. Bluetooth Mesh Stack software uses the Bluetooth Low Energy technology and build a Mesh network. Figure 1-1 shows the communication topologies.

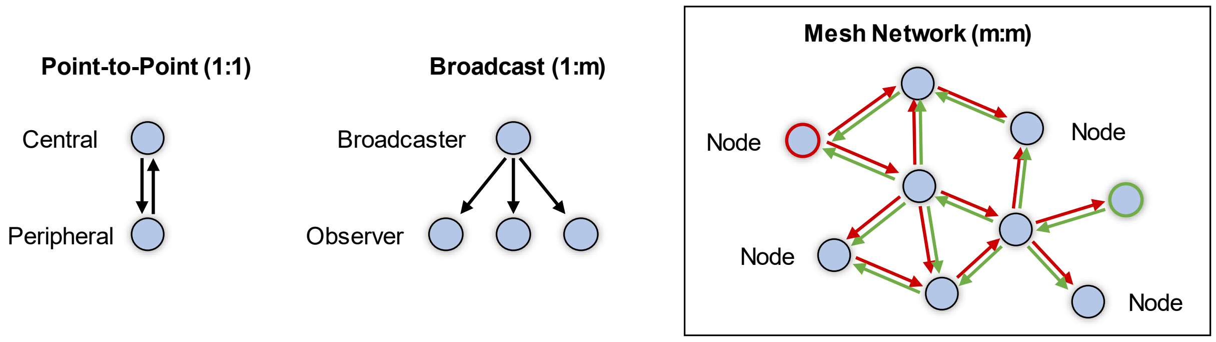


Figure 1-1 Communication Topologies

- **Point-to-Point**

  In a Point-to-Point topology, Central communicates with Peripheral bidirectionally. For example, central device is a smartphone and peripheral device is a fitness tracker respectively. Smartphone informs the tracker that a user starts exercise and then the tracker sends measurement data such as user's exercise intensity.

- **Broadcast**

  In a Broadcast topology, Broadcaster broadcasts data to unspecified Observers, while each Observer decides whether to use received data. For example, Broadcaster device is a beacon for shopping coupon and Observer device is a smartphone respectively. Beacon sends a coupon ID and then the smartphone gets a coupon corresponding the coupon ID via internet.

- **Mesh Network**

  In a Mesh Network topology, many Nodes communicate with each other bidirectionally. Each node can send messages in any timing. Moreover, a node can communicate with the nodes in a place where the message originator node cannot communicate directly, because neighbor nodes relay the message. An example of use-case is a building automation system composed of lightings, air-conditioners, control panel, and smartphones. Building administrator controls whole lightings and air-conditioners of a floor, and building users adjust operations of the lightings and air-conditioner in a room.

1.2 Mesh Network Operation

Operation of Bluetooth Mesh Network is shown below. Also, a network implementing this mesh operation is defined as a **Managed-Flood-based Mesh Network**.

Figure 1-2 shows the operation of nodes making a mesh network. One node (SRC in the figure) sends a message, and neighbor nodes relay the message one after another. The message spreads in whole network and finally arrives at destination node (DST in the figure).

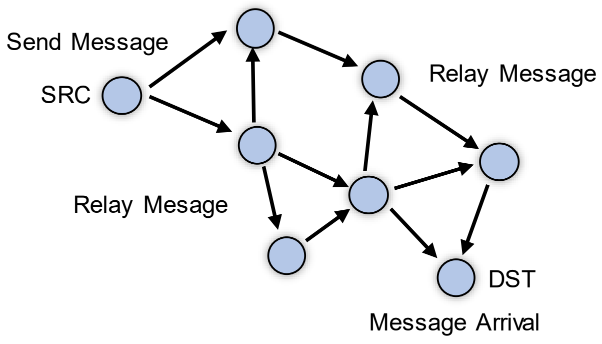


Figure 1-2 Mesh Network Operation

Moreover, nodes use the two methods not to relay each message unlimitedly.

- **Message Cache method**

  When receiving a message, each node relays it and caches it in own list. After that, when receiving the same message again, the node does not relay it.

- **Time to Live method**

  Before sending a message, each node sets limitation of the number of relaying times to TTL (Time to Live) value in the message. Maximum number that can be set is 127. Other nodes decrement the TTL value by 1 and relays the message. Message which TTL value is 0 is not relayed.

1.3 Lifecycle of a Mesh Device

Figure 1-3 shows typical lifecycle of a device supporting Bluetooth Mesh.

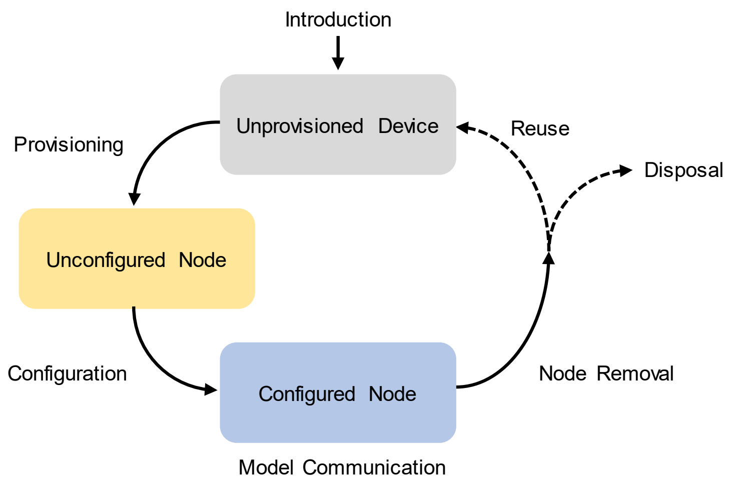


Figure 1-3 Mesh Network Operation

A device supporting Bluetooth Mesh is authenticated and receives Provisioning Data such as a Network Key and Unicast Addresses of each element by Provisioning process.

After Provisioning, device referred to as a node receives configuration required for model operations such as Application Keys, Publish Address, and Subscription Addresses by Configuration process.

After Configuration, the node can communicate with other nodes with models. Also, configuration of the node may be changed by performing Configuration process again.

When there can be a security risk related to Network Keys or/and Application Keys, Configuration Client refreshes these encryption keys. Moreover, unnecessary nodes have risks such as leakage of currently used encryption keys, so the Configuration adds unnecessary nodes into blacklist. Blacklisted nodes cannot receive new encryption keys.

Configuration Client can remove unnecessary node to from a network by Node Removal Procedure. Device removed from a network may be reused or discarded.

2. Overview

2.1 Bluetooth Mesh Demonstration

The sample program included in Bluetooth Mesh Stack Package performs the following four demonstration phases.

- Demo Phase1: Provisioning
- Demo Phase2: Configuration
- Demo Phase3: Model Communication
- Demo Phase4: Node Removal

To perform the demonstration, Mesh Mobile Application which works on a smart phone is also included in the package.

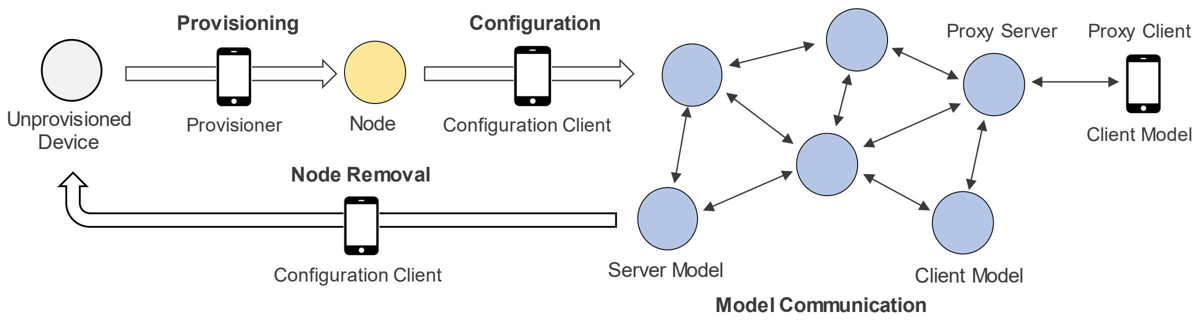


Figure 2-1 Overview of Mesh Demonstration

2.1.1 Demo-Phase1: Provisioning

Each Unprovisioned Device needs to be provisioned by a Provisioner. Provisioning is the process of authenticating and providing basic information such as Unicast Addresses and a Network Key to transmit or receive messages in a mesh network as a node. Also, Provisioner is typically mobile computing device such as smart phone.

The sample program works as a Unprovisioned Device (Provisioning Server). After power on, the sample program starts transmitting Unprovisioned Device beacon and connectable advertising alternately to be discovered by a Provisioner.

On the other hand, Mesh Mobile Application works as a Provisioner (Provisioning Client) and performs provisioning by receiving a connectable advertising.

2.1.2 Phase2: Configuration

After provisioned, provisioned device is called a Node. Configuration Client distributes configuration for mesh model communication such as Application Keys, Publish Address and Subscription List to each Node which works as Configuration Server. Also, Configuration Client is typically a smart phone or other mobile computing device.

The sample program works as a Configuration Server. After Provisioning, the sample program transmits Connectable Advertising to connect to Configuration Client.

On the other hand, Mesh Mobile Application works as a Configuration Client and performs configuration by receiving a connectable advertising.

2.1.3 Phase3: Model Communication

After Configuration, each node encrypts messages with the Application Key and sends them to the Publish Address. Also, it receives the message sent to the address in the Subscription List and decrypts it with the Application Key.

Original Vendor Server and Vendor Client as well as Generic OnOff Server and Generic OnOff Client are implemented in the sample program.

When a switch on development board is pushed, Generic OnOff Client sends Generic OnOff Set Unacknowledged message that includes value representing either On or Off. When receiving the message, Generic OnOff Server controls a LED on development board in accordance with the value included in the message.

When character string is input to console on PC connected to development board, Vendor Client sends Vendor Set Unacknowledged message that includes the character string. When receiving the message, Vendor Server Vendor Client outputs the character string included in the message to console on PC.

Mesh Mobile Application can work as Generic OnOff Client and Vendor Client and send messages. It connects to a Proxy node to transmit and receive messages. Proxy node forwards massages to neighbor nodes, and each node relays messages, then messages spread in the mesh network.

2.1.4 Phase4: Node Removal

Configuration Clien can remove a node from a mesh network by sending Config Node Reset message.

When receiving a Config Node Reset message, the sample program reset its configuration information and perform Provisioning again.

3. Hardware Requirement

Bluetooth Mesh Stack Package includes the sample program that works on the following development boards for RX23W. It is recommended to use two or more boards for developing applications.

- 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 (hereinafter referred to as "TB") can be used for developing applications using Bluetooth Mesh Stack. It has on-board debugging emulator that is equivalent to E2 emulator Lite, so you can develop software without external emulator.

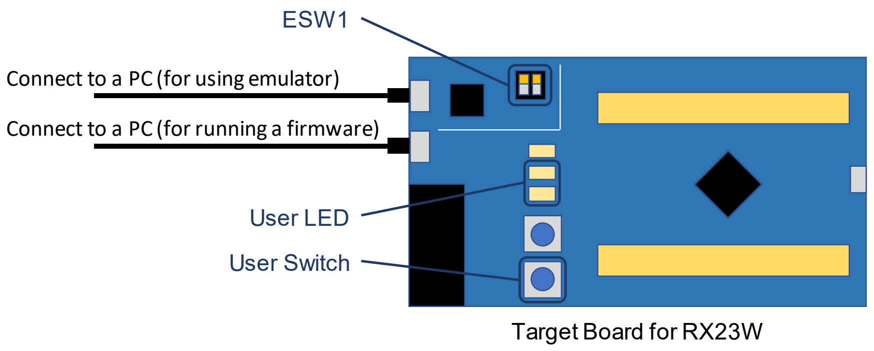

ESW1
Connect to a PC (for using emulator)
Connect to a PC (for running a firmware)
User LED
User Switch
Target Board for RX23W

Figure 3-1 Target Board for RX23W

NOTE: Keep in mind that it is necessary to configure the switch ESW1 appropriately as follows:

- Before you use on-board emulator for writing a firmware or debugging, switch on the 2-4 of ESW1.
- Before you run a firmware without on-board emulator, switch off the 2-4 of ESW1.

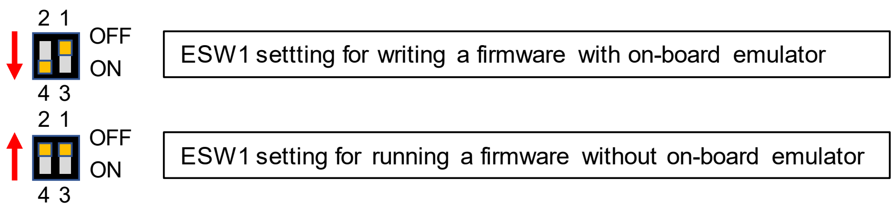

2 1
OFF
ON
4 3
ESW1 settting for writing a firmware with on-board emulator
2 1
OFF
ON
4 3
ESW1 setting for running a firmware without on-board emulator

Figure 3-2 Switch ESW1 settings on Target Board for RX23W

For more information regarding TB, refer to "RX23W Group Target Board for RX23W User's Manual" (R20UT4634).

3.2 Target Board for RX23W module

Similar to Target Board for RX23W, Target Board for RX23W module (hereinafter referred to as "TB module") can be used for developing applications using Bluetooth Mesh Stack. For more information regarding Target Board for RX23W module, refer to "RX23W Group Target Board for RX23W module User's Manual" (R20UT4890).

3.3 Renesas Solution Starter Kit for RX23W

Renesas Solution Starter Kit for RX23W (hereinafter referred to as "RSSK") can also be used for developing applications using Bluetooth Mesh Stack.

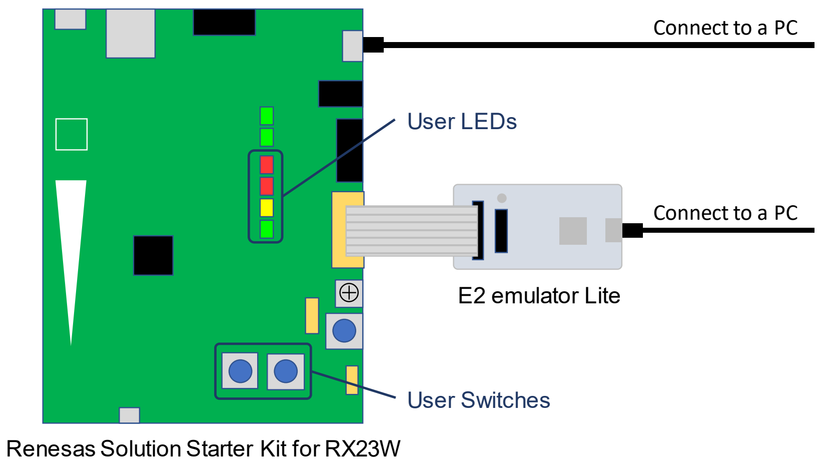


Figure 3-3 Renesas Solution Starter Kit for RX23W

To supply power from the USB Serial Connector (USBCN0), refer to the jumper setting shown in Table 3-1.

Table 3-1 Jumper Setting of Renesas Solution Starter Kit for RX23W

Jumper	Setting
J3	1-2 Short
J4	1-2 Short
J5	1-2 Short
J10	Short
J13	Short
J7	1-2

For more information regarding RSSK, refer to "RX23W Group Renesas Solution Stater Kit for RX23W CPU Board User's Manual" (R20UT4446).

3.4 E2 emulator Lite

E2 emulator Lite is an emulator that supports real-time debugging on the actual chip and a flash programmer for MCUs of the RX and RL78 families. If you use another emulator, check if it supports RX23W.

3.5 Smart Phone or Mobile Computing Device

Typically, Mobile Computing Device such as Smart Phone is used as A Provisioner and Configuration Client. Also, the sample program included in Bluetooth Mesh Stack Package needs either Android device or iOS device which can perform Bluetooth Low Energy communication to perform demonstration.

4. Software Requirement

4.1 Bluetooth Mesh Stack

Bluetooth Mesh Stack (hereinafter referred to as "Mesh Stack") provides applications with the Bluetooth Mesh Networking features for many-to-many wireless communication in a mesh network. Mesh Stack is provided as a FIT Module. Also, Mesh Stack and sample programs are included in Bluetooth Mesh Stack Package (hereinafter referred to as "MESH Package").

4.2 Bluetooth Low Energy Protocol Stack

Mesh Stack requires the Bluetooth Protocol Stack to use Bluetooth Low Energy technology. Bluetooth Low Energy Protocol Stack is provided as a FIT Module. Also, Bluetooth Low Energy Protocol Stack is included in the MESH Package.

4.3 Mesh Mobile Application

Bluetooth Mesh Mobile Application (hereinafter referred to as "Mesh Mobile") is a sample application for demonstration and works on Android and iOS. You can get Mesh Mobile application from the following link. Install the application to your device.

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

For iOS: https://apps.apple.com/app/renesas-meshmobile/id1641877699

4.4 e² studio IDE and CC-RX Compiler

The demo projects included in MESH Package were generated by the following IDE.

IDE	: Renesas Electronics e^2 studio 2022-10

Mesh Stack library was built by the following compiler.

Compiler	: Renesas Electronics CC-RX V2.08.01

e² studio is a development environment based on the popular Eclipse CDT (C/C++ Development Tooling), covers build (editor, compiler, and linker control) and debug. You can get the installer from the following web site.

e^2 studio:
https://www.renesas.com/products/software-tools/tools/ide/e2studio.html#downloads

Moreover, C/C++ Compiler Package for RX Family (CC-RX) is required for building the RX23W firmware. It can be installed during above e^2 studio installation process. To use this compiler, compiler license is required. For more information, refer to the following website.

Compiler Licenses:
https://www.renesas.com/products/software-tools/tools/compiler-assembler/compiler-licenses.html

4.5 Renesas Flash Programmer

Renesas Flash Programmer (hereinafter referred to as "RFP") is a tool for writing firmware to the on-chip flash memory of Renesas MCUs in each phase of development and mass production. You can get the installer from the following website.

Renesas Flash Programmer (Programming GUI):
https://www.renesas.com/products/software-tools/tools/programmer/renesas-flash-programmer-programming-gui.html#downloads

4.6 Third-party Software

4.6.1 Serial Terminal Emulator

The sample program sends log messages over UART. Serial terminal emulator that supports CSI (Control Sequence Indicator) of ANSI escape sequence can be used for checking log message.

Tera Term

For example of log messages sent from the sample program, refer to Section 8.1.

4.6.2 Development Tools required for building the Mesh Mobile

The following development tools are required for building Mesh Mobile. You can get them from the internet. For more details on how to build the Mesh Mobile, refer to Section 6.3.

npm (Node Package Manager) included in Node.js
Python
Capacitor
Ionic framework
Android Studio, for building the Mesh Mobile for Android
Xcode, for building the Mesh Mobile for iOS

NOTE: Xcode works on Mac PC only. To download this, Apple ID is required. Moreover, to develop the iOS App products, paid license is required.

Apple Developer Program, for distributing Apps via App Store
Apple Developer Enterprise Program, for In-house Apps

5. Bluetooth Mesh Stack Package

5.1 Contents

Contents of the MESH Package "r01an4930xx0131-rx23w-blemesh-fit.zip" are shown as follows:

```
r01an4930xx0131-rx23w-blemesh-fit.zip
    |   blemesh_api.chm                         :   Mesh Stack API Manual
    |   r01an4874ej0131-blemesh.pdf             :   Mesh Startup Guide (en)
    |   r01an4874jj0131-blemesh.pdf             :   Mesh Startup Guide (ja)
    |   r01an4875ej0131-blemesh.pdf             :   Mesh Development Guide (en)
    |   r01an4875jj0131-blemesh.pdf             :   Mesh Development Guide (jp)
    |   r01an4930ej0131-rx23w-blemesh.pdf       :   Mesh FIT Module Document (en)
    |   r01an4930jj0131-rx23w-blemesh.pdf       :   Mesh FIT Module Document (ja)
    |   readme.txt                              :   Mesh Stack Package Information File
    |                                           :
    +---FITDemos\                               :   ---- Mesh FIT Demonstration Projects ----
    |   |   make_workspace_rsskrx23w.bat        :   Batch File to make workspace for RSSK
    |   |   make_workspace_tbrx23w.bat          :   Batch File to make workspace for TB
    |   |   rsskrx23w_mesh_cli.zip              :   Command Line Interface Program for RSSK
    |   |   rsskrx23w_mesh_client.zip           :   Client Models Sample Program for RSSK
    |   |   rsskrx23w_mesh_server.zip           :   Server Models Sample Program for RSSK
    |   |   tbrx23w_mesh_cli.zip                :   Command Line Interface Program for TB
    |   |   tbrx23w_mesh_client.zip             :   Client Models Sample Program for TB
    |   |   tbrx23w_mesh_server.zip             :   Server Models Sample Program for TB
    |   |   tbrx23wmodule_mesh_cli.zip          :   Command Line Interface Program for TB module
    |   |   tbrx23wmodule_mesh_client.zip       :   Client Models Sample Program for TB module
    |   |   tbrx23wmodule_mesh_server.zip       :   Server Models Sample Program for TB module
    |   |                                       :
    |   +---ROM_Files\                          :   ---- Pre-built Firmware files ----
    |           rsskrx23w_mesh_cli.mot          :   Command Line Interface Program for RSSK
    |           rsskrx23w_mesh_client.mot       :   Client Models Sample Program for RSSK
    |           rsskrx23w_mesh_server.mot       :   Server Models Sample Program for RSSK
    |           tbrx23w_mesh_cli.mot            :   Command Line Interface Program for TB
    |           tbrx23w_mesh_client.mot         :   Client Models Sample Program for TB
    |           tbrx23w_mesh_server.mot         :   Server Models Sample Program for TB
    |           tbrx23wmodule_mesh_cli.mot      :   Command Line Interface Program for TB module
    |           tbrx23wmodule_mesh_client.mot   :   Client Models Sample Program for TB module
    |           tbrx23wmodule_mesh_server.mot   :   Server Models Sample Program for TB module
    |                                           :
    +---FITModules\                             :   ---- Mesh FIT Module ----
    |       r_mesh_rx23w_v1.31.xml              :   Plug-in File
    |       r_mesh_rx23w_v1.31.zip              :   Mesh FIT Module
    |       r_mesh_rx23w_v1.31_extend.mdf       :   Configuration Description File
    |                                           :
    +---mesh_mobile\                            :   ---- Mesh Mobile Application ----
```

In this document, Client Models Sample Program Server Models Sample Program are used. Regarding the Command Line Interface Program, refer to Section 4.1 in "RX23W Group Bluetooth Mesh Stack Development Guide" (R01AN4875).

Contents of the Mesh FIT Module "r_mesh_rx23w_v1.31.zip" are shown as follows:

```
r_mesh_rx23w_v1.31.zip
   +---r_config\                                    :  ---- FIT Configurations ----
   |       r_mesh_rx23w_config.h                    :  Mesh FIT Configuration File
   |                                                :
   +---r_mesh_rx23w\                                :  ---- Mesh FIT Module ----
       |   r_mesh_rx23w_if.h                        :  Mesh Stack API Header File
       |   readme.txt                               :  Mesh FIT Module Information
       |                                            :
       +---doc\                                     :
       |   |   blemesh_api.chm                      :  Mesh Stack API Manual
       |   +---en\                                  :
       |   |       r01an4930ej0131-rx23w-blemesh.pdf   :  Mesh FIT Module Application Note (en)
       |   +---ja\                                  :
       |           r01an4930jj0131-rx23w-blemesh.pdf   :  Mesh FIT Module Application Note (ja)
       |                                            :
       +---json                                     :  Service Definition Files for the QE for BLE
       |       mesh_provisioning.service.json       :
       |       mesh_proxy.service.json              :
       |                                            :
       +---lib\                                     :
       |       lib_ble_ms_ccrx.lib                  :  Mesh Stack Library
       |                                            :
       +---src\                                     :
           +---bearer\                              :  ---- Bluetooth Bearer ----
           +---drivers\                             :  ---- Mesh Drivers ----
           +---include\                             :  ---- Mesh Stack Header Files ----
```

Regarding the Mesh FIT Module Configuration, refer to Section 3.1 in "RX23W Group Bluetooth Mesh Module Using Firmware Integration Technology" (R01AN4930).

6. Development Environment Setup

This chapter describes how to set up a development environment for developing applications using Bluetooth Mesh Stack.

In the case that you need to perform demonstration as soon as possible, skip this chapter and go on to the Chapter 7; you can perform demonstration by using prebuilt firmware that is included in MESH Package.

6.1 e^2 studio IDE and CC-RX Compiler

Launch the e^2 studio installer and follow the installation wizard. In the installation process, select RX Devices as a Device Family to be supported. Next, CC-RX is installed by default, so it is not necessary to change configuration of Optional Components to be installed.

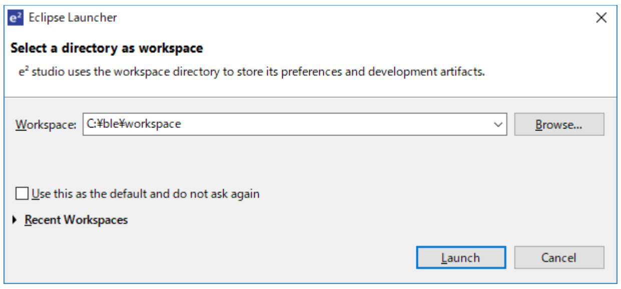
NOTE: Installation directory name and its path should not include space codes, multi-byte characters, and some symbols such as '$','#' and '%'. That is because those characters may prevent installation or build/debug process. Similarly, workspace name and its path should not include those characters.

NOTE: Workspace directory name and its path should be as short as possible. That is because long workspace path may prevent compilation and/or linkage of any source code placed in the workspace.

6.2 Demo Project

Follow the steps bellow:

1. Launch the e^2 studio and select any directory as a workspace.


Eclipse Launcher

Select a directory as workspace

e² studio uses the workspace directory to store its preferences and development artifacts.

Workspace: C:¥ble¥workspace

☐ Use this as the default and do not ask again

Recent Workspaces

2. Import projects in the workspace.

If you use TB, import the archive files below included in the "FITDemos" directory:

```
FITDemos\
   +---tbrx23w_mesh_client.zip          :  Project for Client Models
   +---tbrx23w_mesh_server.zip          :  Project for Server Models
   +---tbrx23w_mesh_cli.zip             :  Project for Command Line Interface (CLI)
```

If you use TB module, import the archive files below included in the "FITDemos" directory.

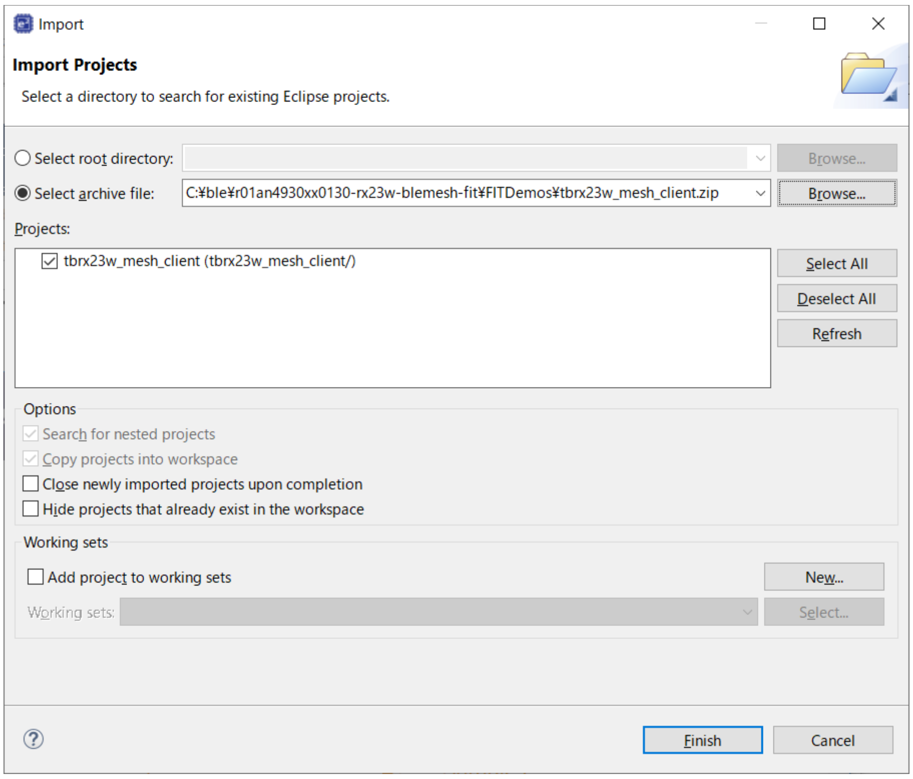
```
FITDemos\
   +---tbrx23wmodule_mesh_client.zip    :  Project for Client Models
   +---tbrx23wmodule_mesh_server.zip    :  Project for Server Models
   +---tbrx23wmodule_mesh_cli.zip       :  Project for Command Line Interface (CLI)
```

If you use RSSK, import the archive files below included in the "FITDemos" directory.

```
FITDemos\
   +---rsskrx23w_mesh_client.zip        :  Project for Client Models
   +---rsskrx23w_mesh_server.zip        :  Project for Server Models
   +---rsskrx23w_mesh_cli.zip           :  Project for Command Line Interface (CLI)
```

Select [FIle]→[Import…] in the menu to open the Import dialog.

In the Import dialog, select the project for Client Models (mesh_client) in the FITDemos directory and click the Finish button. Similarly, import the project for Server Models (mesh_server) too.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

- ○ Select root directory: [Browse...]
- ● Select archive file: C:¥ble¥r01an4930xx0130-rx23w-blemesh-fit¥FITDemos¥tbrx23w_mesh_client.zip [Browse...]

Projects:

- ☑ tbrx23w_mesh_client (tbrx23w_mesh_client/)

[Select All] [Deselect All] [Refresh]

Options

- ☑ Search for nested projects
- ☑ Copy projects into workspace
- ☐ Close newly imported projects upon completion
- ☐ Hide projects that already exist in the workspace

Working sets

- ☐ Add project to working sets [New...]
- Working sets: [Select...]

[Finish] [Cancel]

3. Click Build icon to build the project.

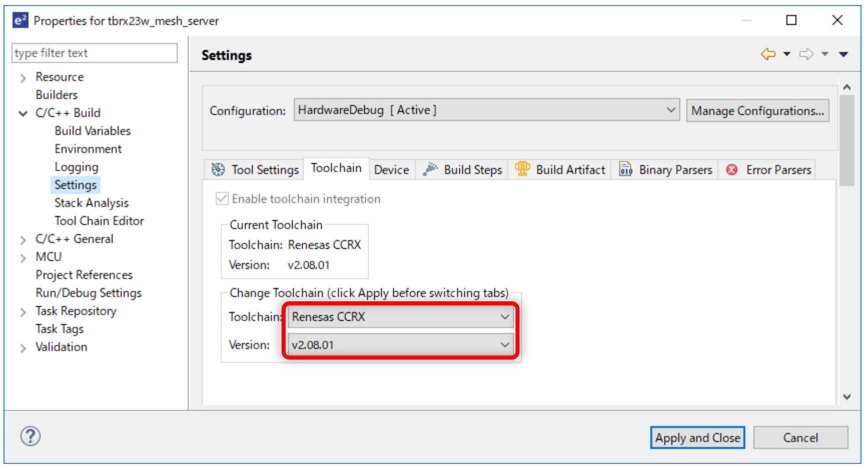
   You can see and change current project with Launch Configuration tbrx23w_mesh_server HardwareDel .
   After building, firmware (.mot file) is generated in the "HardwareDebug\" in the project directory.

4. Connect development board and E2 emulator Lite to a PC and connect the emulator to the board.
5. Click Debug icon to launch the project in a debug mode.
   After launching the project, the firmware is downloaded to development board.

6. Click Resume icon on Debug Perspective to run the project.
7. After debugging the project, click Terminate icon on Debug Perspective.
   Firmware of the project remains on the flash memory of RX23W even after termination and power off.

To perform the demonstration described in Chapter 7, it is recommended to use at least two development boards; at least one board works as a Client and the other boards work as a Server. Before going to Chapter 7, write the firmware of Server to some boards and then write the firmware of Client to the other boards by either repeating Step 5 and Step 7 or using the RFP.

NOTE: When the error indicating "No toolchain set or toolchain not integrated." occurs and building fails, open the project properties dialog and move to [C/C++ Build]→ [Settings]→[Toolchain] tag, then set the toolchain.


Properties for tbrx23w_mesh_server

Settings

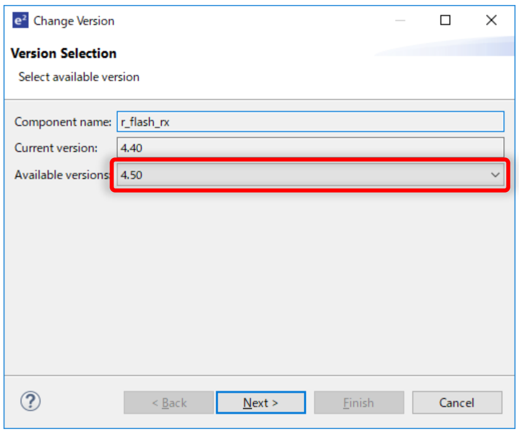
Configuration: HardwareDebug [Active]

Enable toolchain integration

Current Toolchain
Toolchain: Renesas CCRX
Version: v2.08.01

Change Toolchain (click Apply before switching tabs)
Toolchain: Renesas CCRX
Version: v2.08.01

NOTE: The project includes source codes generated by FIT Modules. You can perform code generation by using Smart Configurator. When the error indicating "W04020001: The configuration r_flash_rx has been unloaded because the module could not be found" occurs, download the latest FIT Modules and select [Change version] in right-click menu of each component in Smart Configurator and then select the downloaded version.


Change Version

Version Selection

Select available version

Component name: r_flash_rx
Current version: 4.40
Available versions: 4.50

6.3 Mesh Mobile

Regarding how to build and install Mesh Mobile, refer to "mesh_mobile\mesh_mobile_guide.pdf".

7. Demonstration

This chapter describes how to operate the sample program and Mesh Mobile for mesh demonstration. It is recommended to use at least two development boards for this demonstration; one works as a Client and the other works as a Server.

If you skip the Chapter 6, write firmware to development boards by using the RFP. Firmware for demonstration is included in Mesh Package as follows:

```
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
```

If the Mesh Mobile is not installed yet, follow the steps bellow:

NOTE: Application for Android only is included in the Package. If you will use iOS device, build and install the application in accordance with Section 6.3.

1. Copy the following package file from PC to Android via USB.

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

2. Execute the package file using any file manager application on Android.

Also, you can observe the log message sent from each development board by using serial terminal emulator with the following settings:

Table 7-1 Serial Port Setting

Item	Setting
Baud rate	115200 bps
Data	8 bits
Parity	none
Stop	1 bit
Flow Control	none

7.1 Phase1: Provisioning

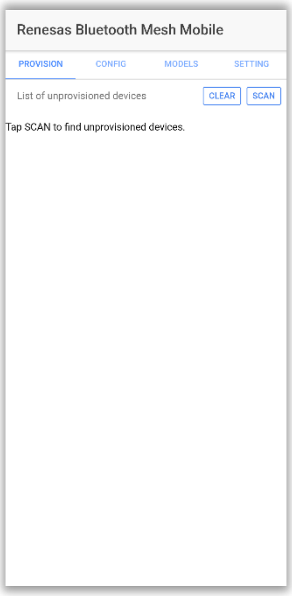
To start demonstration, power on all development boards and launch the Mesh Mobile.

NOTE: Mesh Mobile for Android requires some permissions; Location, Nearby devices, and Storage.
NOTE: If SCAN operation fails on Mesh Mobile for Android, remove the permissions above once and then set the permissions again.

Repeat the following steps to provision all development boards.

1. To start searching Unprovisioned Devices, move to PROVISION tab and tap SCAN button.
2. Select any device in result of searching to provision it.
3. Establishing a connection and provisioning are performed.
4. When the device supports OOB Public Key and/or any of the Authentication method, the Mesh Mobile shows a screen to select whether to use OOB Public Key and select any of Authentication method. For simplicity of procedure in this demo, check "No OOB Public Key is used" and "No OOB Authentication is used", and tap PROVISIONING START button.

Step1

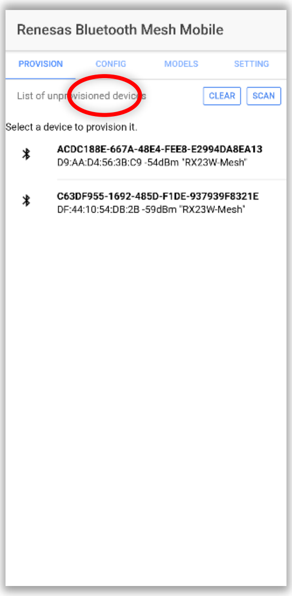


Renesas Bluetooth Mesh Mobile

List of unprovisioned devices

Tap SCAN to find unprovisioned devices.

Step2



Renesas Bluetooth Mesh Mobile

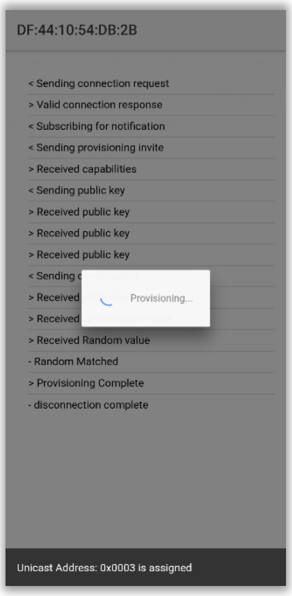
List of unprovisioned devices

Select a device to provision it.

ACDC188E-667A-48E4-FEE8-E2994DA8EA13
D9:AA:D4:56:3B:C9 -54dBm "RX23W-Mesh"

C63DF955-1692-485D-F1DE-937939F8321E
DF:44:10:54:DB:2B -59dBm "RX23W-Mesh"

Step3



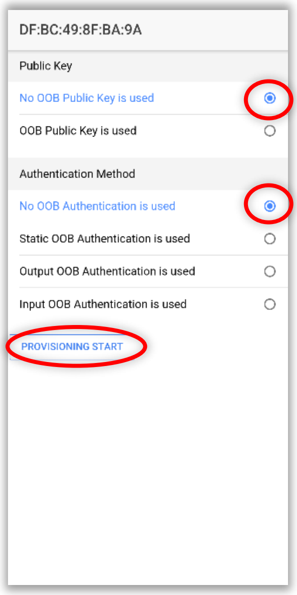
DF:44:10:54:DB:2B

```
< Sending connection request
> Valid connection response
< Subscribing for notification
< Sending provisioning invite
> Received capabilities
< Sending public key
> Received public key
> Received public key
> Received public key
< Sending c
> Received
> Received
> Received Random value
- Random Matched
> Provisioning Complete
- disconnection complete
```

Provisioning...

Unicast Address: 0x0003 is assigned

Step4



DF:BC:49:8F:BA:9A

Public Key
[✓ No OOB Public Key is used]
OOB Public Key is used

Authentication Method
[✓ No OOB Authentication is used]
Static OOB Authentication is used
Output OOB Authentication is used
Input OOB Authentication is used

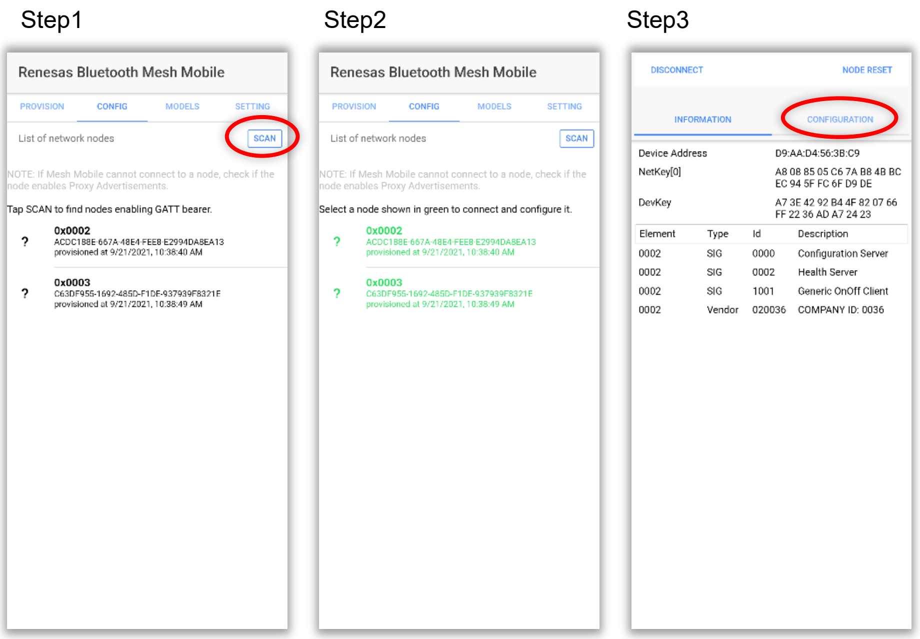
Figure 7-1 Provisioning

7.2 Phase2: Configuration

After all development boards are provisioned, repeat the following steps to configure all development boards.

1. To start searching nodes, move to CONFIG tab and tap SCAN button.
2. Connectable nodes are displayed in green. To connect and perform Configuration, select any node displayed in green.

   NOTE: Node suspends Connectable Advertising transmission after 60seconds. If device to be configured is not displayed in green, long-press the SW1 of development board longer than 2seconds to resume Connectable Advertising transmission.
3. After Composition Data is displayed, move to CONFIGURATION tab.



List of network nodes

NOTE: If Mesh Mobile cannot connect to a node, check if the node enables Proxy Advertisements.

Tap SCAN to find nodes enabling GATT bearer.

? 0x0002
ACDC188E-667A-48E4-FEE8-E2994DA8EA13
provisioned at 9/21/2021, 10:38:40 AM

? 0x0003
C63DF955-1692-485D-F1DE-937939F8321E
provisioned at 9/21/2021, 10:38:49 AM

List of network nodes

NOTE: If Mesh Mobile cannot connect to a node, check if the node enables Proxy Advertisements.

Select a node shown in green to connect and configure it.

? 0x0002
ACDC188E-667A-48E4-FEE8-E2994DA8EA13
provisioned at 9/21/2021, 10:38:40 AM

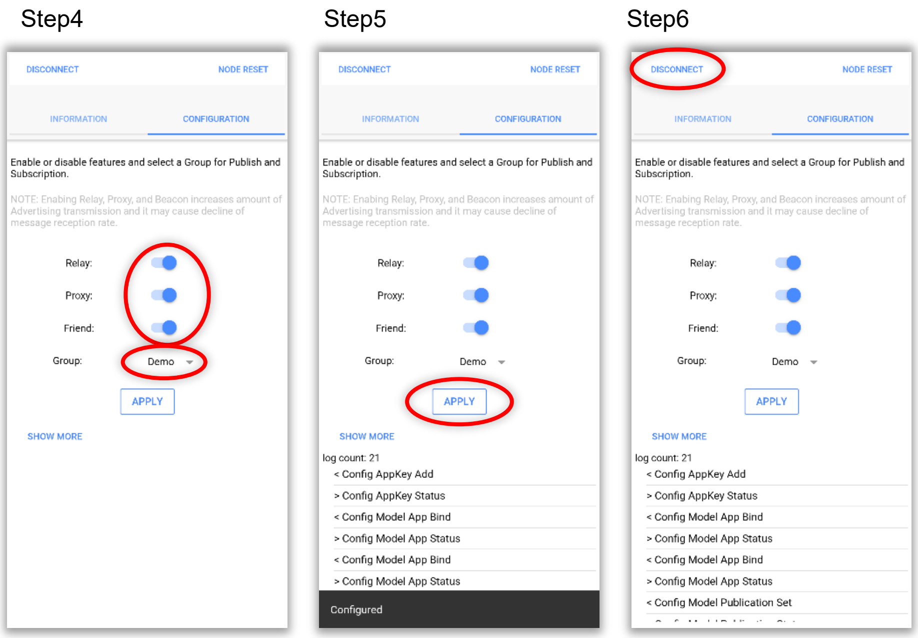
? 0x0003
C63DF955-1692-485D-F1DE-937939F8321E
provisioned at 9/21/2021, 10:38:49 AM

Device Address	D9:AA:D4:56:3B:C9
NetKey[0]	A8 08 85 05 C6 7A B8 4B BC EC 94 5F FC 6F D9 DE
DevKey	A7 3E 42 92 B4 4F 82 07 66 FF 22 36 AD A7 24 23

Element	Type	Id	Description
0002	SIG	0000	Configuration Server
0002	SIG	0002	Health Server
0002	SIG	1001	Generic OnOff Client
0002	Vendor	020036	COMPANY ID: 0036

Figure 7-2 Configuration (1/2)

4. In the CONFIGURATION tab, configure to enable Relay, Proxy, and Friend and register "Demo" Group.
5. Tap APPLY button.
6. After Configured, tap DISCONNECT button.



Enable or disable features and select a Group for Publish and Subscription.

NOTE: Enabing Relay, Proxy, and Beacon increases amount of Advertising transmission and it may cause decline of message reception rate.

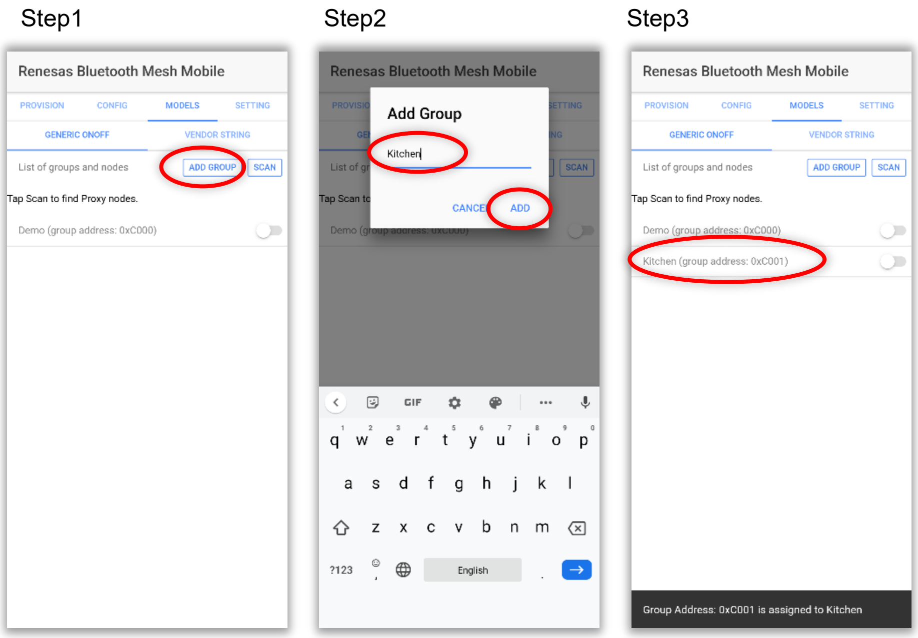
```
log count: 21
< Config AppKey Add
> Config AppKey Status
< Config Model App Bind
> Config Model App Status
< Config Model App Bind
> Config Model App Status
< Config Model Publication Set
```

Configured

Figure 7-3 Configuration (2/2)

Group selected by Configuration is used when the application makes a group to operate multiple nodes. Any group can be added by the following steps.

1. Move to MODELS tab and tap ADDGROUP button.
2. In the Add Group dialog, enter any group name like a "Kitchen".
3. Check the group is added.



List of groups and nodes

Tap Scan to find Proxy nodes.

Demo (group address: 0xC000)

Kitchen (group address: 0xC001)

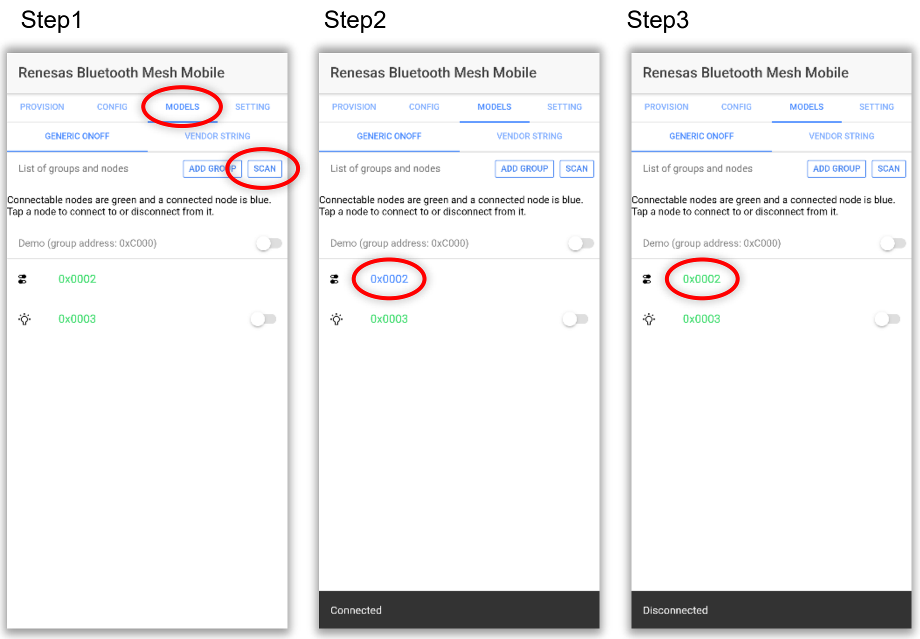
Group Address: 0xC001 is assigned to Kitchen

Figure 7-4 Add Group

7.3 Phase3: Model Communication

After development boards are configured, Mesh Mobile and development boards can send model messages. First, follow the steps below to establish a Proxy connection between Mesh Mobile and one of development boards. Mesh Mobile sends messages to any nodes over the connected Proxy node.

1. To start searching Proxy Nodes, move to MODELS tab and tap SCAN button.
2. To connect a Proxy node, select any node displayed in green. After a connection is established, node is displayed in blue.
3. Tapping the node displayed in blue can disconnect a connection.



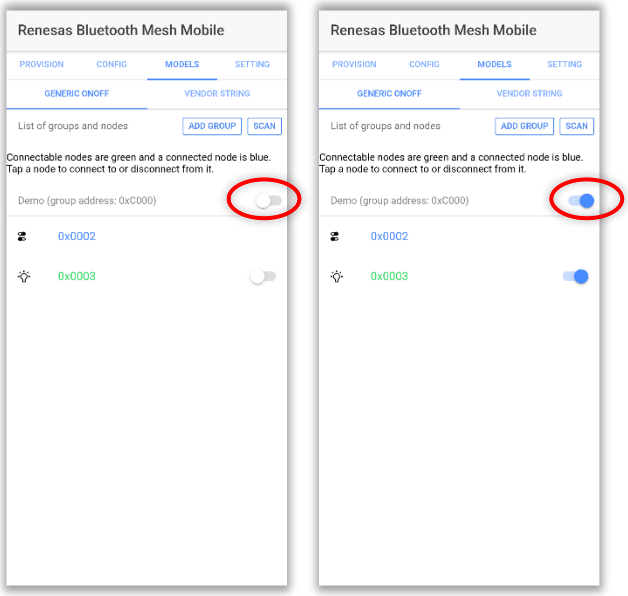
Connectable nodes are green and a connected node is blue.
Tap a node to connect to or disconnect from it.

Figure 7-5 Proxy Connection

7.3.1 Generic OnOff State Control by Mesh Mobile

1. In the GENERIC ONOFF of the MODELS tab, switch a toggle button of the group repeatedly.

   You can see that LEDs of all development boards working as Generic OnOff Server follow the Generic OnOff SET messages received.

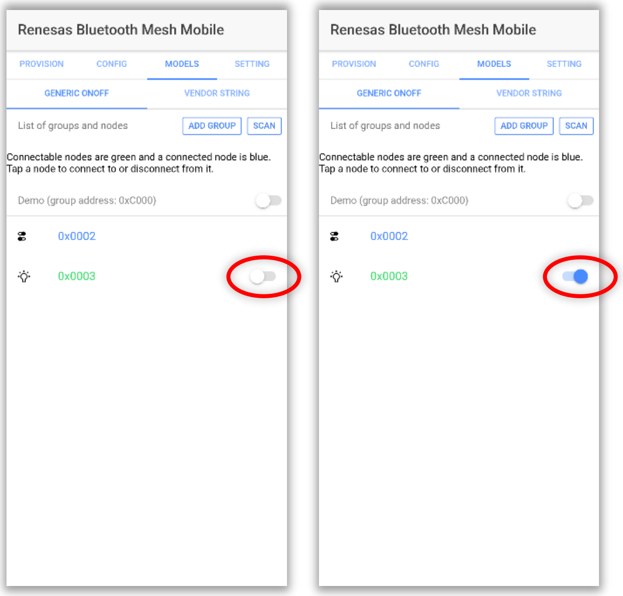


Connectable nodes are green and a connected node is blue.
Tap a node to connect to or disconnect from it.

Figure 7-6 Publication of Generic OnOff SET Messages to Unicast Address

2. In the GENERIC ONOFF tab of the MODELS tab, switch a toggle button of any development board repeatedly.

   You can see that a LED of the development board working as Generic OnOff Server follows the Generic OnOff SET messages received.



Connectable nodes are green and a connected node is blue.
Tap a node to connect to or disconnect from it.

Figure 7-7 Publication of Generic OnOff SET Messages to Group Address

7.3.2 Generic OnOff State Control by Development Board

1. Push the SW1 on each development board working as Generic OnOff Client repeatedly.

   You can see that LEDs of all development boards working as Generic OnOff Server follow the Generic OnOff SET messages received.

7.3.3 Vendor State Control by Mesh Mobile

1. In the VENDOR STRING of the MODELS tab, tap the STRING button of the group. Next, enter any character string and tap SEND button in the dialog.

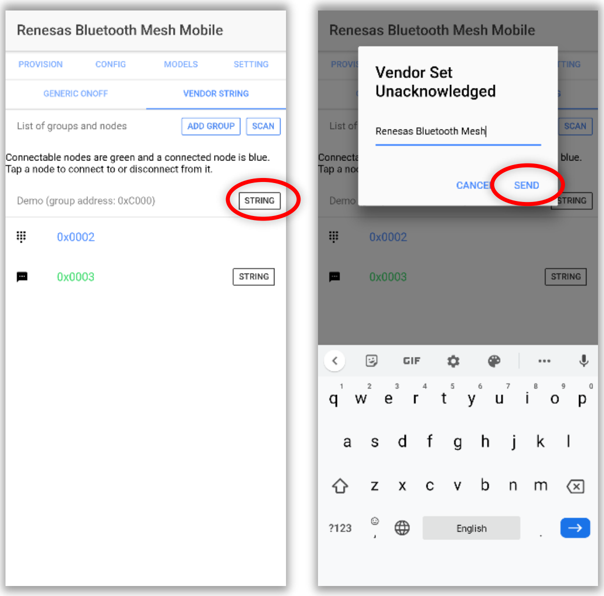

Renesas Bluetooth Mesh Mobile
PROVISION
CONFIG
MODELS
SETTING
GENERIC ONOFF
VENDOR STRING
List of groups and nodes
ADD GROUP
SCAN
Connectable nodes are green and a connected node is blue.
Tap a node to connect to or disconnect from it.
Demo (group address: 0xC000)
STRING
0x0002
0x0003
STRING
Vendor Set Unacknowledged
Renesas Bluetooth Mesh
CANCEL
SEND

Figure 7-8 Publication of Vendor SET Messages to Group Address

You can see that all development boards working as Vendor Server output the character string included in the Vendor SET messages received to console.

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

Figure 7-9 Console Log of Vendor Server Nodes

2. In the VENDOR STRING of the MODELS tab, tap the STRING button of any development node. Next, enter any character string and tap SEND button in the dialog.

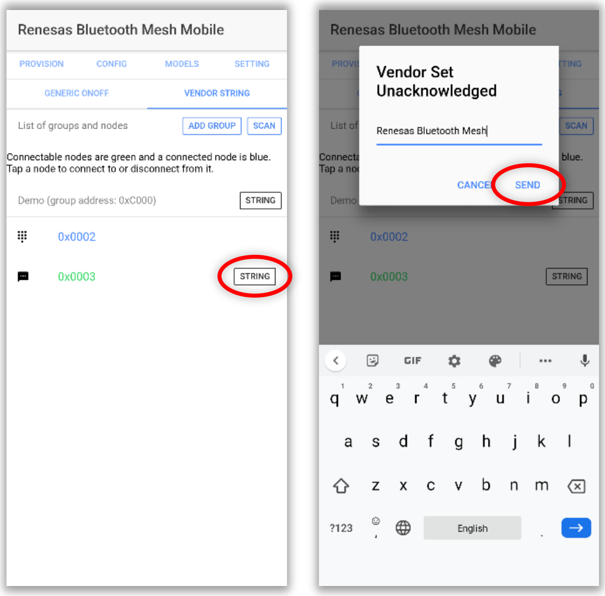

Renesas Bluetooth Mesh Mobile
PROVISION
CONFIG
MODELS
SETTING
GENERIC ONOFF
VENDOR STRING
List of groups and nodes
ADD GROUP
SCAN
Connectable nodes are green and a connected node is blue.
Tap a node to connect to or disconnect from it.
Demo (group address: 0xC000)
STRING
0x0002
0x0003
STRING
Vendor Set Unacknowledged
Renesas Bluetooth Mesh
CANCEL
SEND
English

Figure 7-10 Publication of Vendor SET Messages to Unicast Address

You can see that the development board working as Vendor Server outputs the character string included in the Vendor SET messages received to console.

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

Figure 7-11 Console Log of a Vendor Server Node

7.3.4 Vendor State Control by Development Board

1. Enter any character string and press enter key in the console connected to the development board which works as Vendor Client.

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

Figure 7-12 Entering Character String in Console of a Vendor Client Node

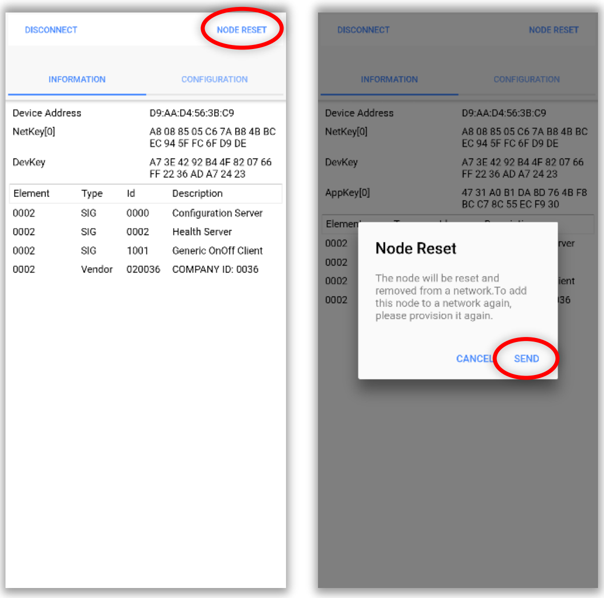
You can see that the development board working as Vendor Server outputs the character string included in the Vendor SET messages received to console.

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

Figure 7-13 Console Log of a Vendor Server Node

7.3.5 Node Reset by Mesh Mobile

1. Connect a node in the CONFIG tab and tap the RESET button. Next, tap the SEND button in the dialog.



Device Address	D9:AA:D4:56:3B:C9
NetKey[0]	A8 08 85 05 C6 7A B8 4B BC EC 94 5F FC 6F D9 DE
DevKey	A7 3E 42 92 B4 4F 82 07 66 FF 22 36 AD A7 24 23

Element	Type	Id	Description
0002	SIG	0000	Configuration Server
0002	SIG	0002	Health Server
0002	SIG	1001	Generic OnOff Client
0002	Vendor	020036	COMPANY ID: 0036

Device Address	D9:AA:D4:56:3B:C9
NetKey[0]	A8 08 85 05 C6 7A B8 4B BC EC 94 5F FC 6F D9 DE
DevKey	A7 3E 42 92 B4 4F 82 07 66 FF 22 36 AD A7 24 23
AppKey[0]	47 31 A0 B1 DA 8D 76 4B F8 BC C7 8C 55 EC F9 30

Node Reset

The node will be reset and removed from a network.To add this node to a network again, please provision it again.

Figure 7-14 Publication of Config Node Reset Message to a Node

You can see that the development board connecting to Mesh Mobile performs Provisioning again by receiving the Config Node Reset message.

```
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
```

Figure 7-15 Console Log of the Node that received Config Node Reset message

8. Appendix

8.1 Log Message

An example of log message sent from the sample program is shown below. Regarding the serial port setting for terminal tool, refer to Table 7-1 in Chapter 7.

```
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                       Start of Provisioning
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                    End of Provisioning
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                                               Establishment of Proxy Connection
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                              Start of Configuration
         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
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
```

End of Configuration

```
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
```

Generic OnOff Server Model

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

Vendor Server Model

Trademark and Copyright

The *Bluetooth*® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Renesas Electronics Corporation is under license. Other trademarks and registered trademarks are the property of their respective owners.

RX23W Group Bluetooth Mesh Stack uses the following open source software.

- crackle; AES-CCM, AES-128bit functionality
  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.

Revision History

Rev.	Date	Description	
1.00	Sep. 30, 2019	-	First edition
1.01	Nov. 29, 2019	P.4	Added Section 1 "Features"
		P.27	Added Section 8 "Appendix"
1.10	Sep. 30, 2020	P.6	Added Section 1.3 "Lifecycle of a Mesh Device"
		P.24	Added Subsection 7.3.3 "Vendor State Control by Mesh Mobile"
		P.25	Added Subsection 7.3.4 "Vendor State Control by Development Board"
		P.26	Added Subsection 7.3.5 "Node Reset by Mesh Mobile"
		P.27	Updated log in Section 8.1 "Log Message "
		Overall	Updated some description, figures, and screenshots
1.20	Sep. 30, 2021	P.13	Updated Mesh Package contents and Mesh FIT Module contents described in Section 5.1 "Contents"
		P.18	Updated Demo procedures described in Chapter 7 "Demonstration"
		Overall	Updated some description, figures, and screenshots
1.30	Dec. 22, 2022	P.1	Added "Target Board for RX23W module User's Manual" in the table of Related Documents
		P.9	Added Target Board for RX23W module in Chapter 3 "Hardware Requirement"
		P.9	Added Section 3.2 "Target Board for RX23W module"
		P.10	Added Table 3-1 "Jumper Setting of Renesas Solution Starter Kit for RX23W" in Section 3.3 "Renesas Solution Starter Kit for RX23W"
		P.12	4.4e^2 studio IDE and CC-RX Compiler
		P.14	Updated MESH Package contents and FIT Module contents described in Section 5.1 "Contents"
		P.16	Updated the step 1 and step 2 described in Section 6.2 "Demo Project"
		P.19	Updated console logs in Chapter 7 "Demonstration"
		P.28	Updated console logs in Section 8.1 "Log Message"
1.31	Mar. 31, 2025	-	Revised to be consistent with Bluetooth Mesh FIT version

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.

© 2025 Renesas Electronics Corporation. All rights reserved.