RA4W1

Bluetooth Mesh Introduction

Introduction

Flexible Software Package (FSP) provides a Bluetooth Mesh module compliant with Bluetooth Mesh Profile 1.0.1 and Bluetooth Mesh Model 1.0.1 specification. This document introduces the Bluetooth mesh module, application notes published on the Renesas RA4W1 web site, and describes an application development method using the Bluetooth mesh module and the application notes.

Target Device

RA4W1 Group

Related Documents

The following documents are published on Renesas website.

Document Title	Document No.
RA4W1 Group User's Manual: Hardware	R01UH0883
Renesas Flexible Software Package User's Manual	Link
RA4W1 Group Bluetooth Mesh Startup Guide	R01AN5847
RA4W1 Group Bluetooth Mesh sample application Application Note	R01AN5848
RA4W1 Group Bluetooth Mesh Development Guide	R01AN5849
RA4W1 Group EK-RA4W1 Quick Start Guide	R20QS0015
RA4W1 Group EK-RA4W1 User's Manual	R20UT4683
Renesas RA4W1 Web site	Link
Bluetooth SIG	Link

Contents

1. Bluetootn Mesh Structure in the Flexible Software Package

1.1 Bluetooth Mesh module

Flexible Software Package (FSP) provides Bluetooth Mesh module compliant with Bluetooth Mesh Profile 1.0.1 and Bluetooth Mesh Model 1.0.1 specification. The structure of the Bluetooth Mesh module is shown below.

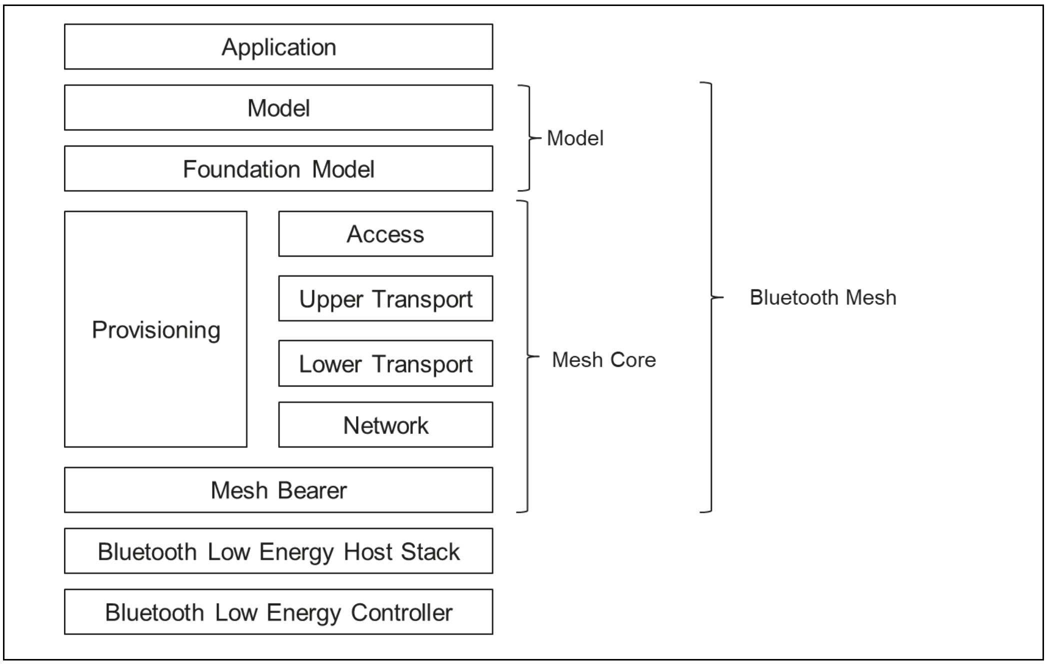


Figure 1 Bluetooth mesh module structure

- **Model Layer**

  The Model layer standardize the primitive functionality for user applications. Refer to section 1.2 for the models which are provided by FSP.

- **Foundation model layer**

  The Foundation Model layer defines the states, messages, and models required to configure and manage a mesh network.

- **Access Layer**

  The Access Layer defines how higher layer applications can use the upper transport layer.

- **Upper Transport Layer**

  The Upper Transport Layer encrypts, decrypts, and authenticates application data and is designed to provide confidentiality for the access messages.

- **Lower Transport Layer**

  The lower transport layer defines how upper transport layer messages are segmented and reassembled into multiple Lower Transport PDUs to deliver large upper transport layer messages to other nodes.

- **Network Layer**

  The Network Layer processes the messages of the Lower Transport layer into a Network PDU that can be transferred by the Bearer layer. It also decrypts the received message, authenticates it, and forwards it to the upper layer.

- **Bearer Layer**

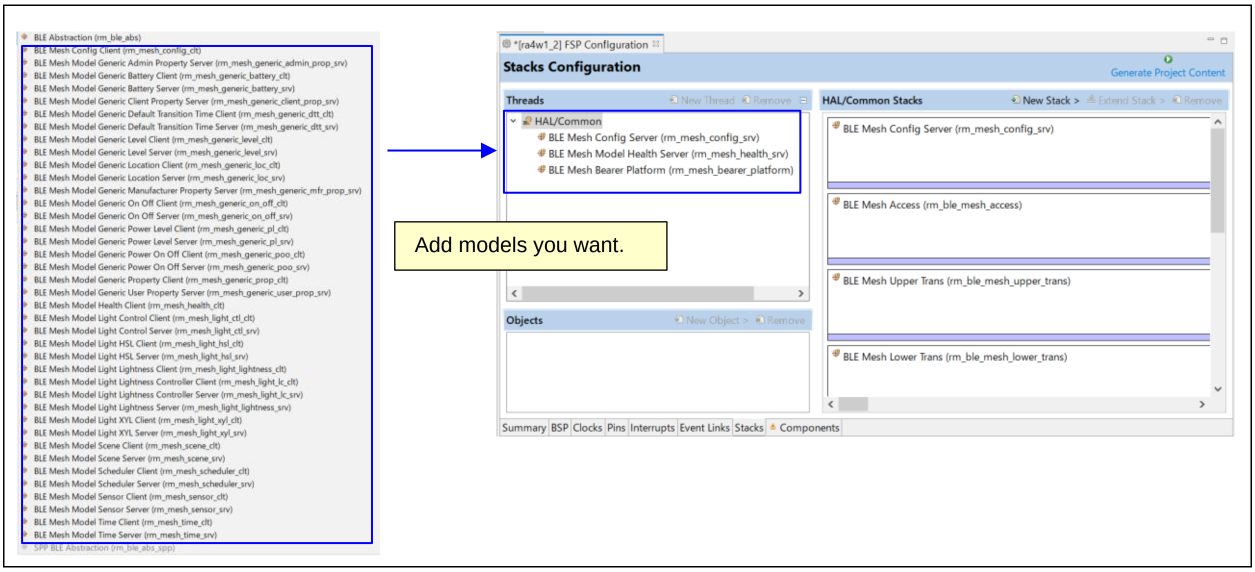
  The Bearer Layer defines how network messages are transported between nodes.

1.2 Models

The Bluetooth Mesh module provides following models which are defined by Bluetooth SIG.

- **Foundation Model**
  - Configuration
  - Health
- **Generic Models**
  - OnOff, Power OnOff, Power OnOff Setup
  - Level, Power Level, Power Level Setup
  - Default Transition Time
  - Battery
  - Location, Location Setup
  - Manufacturer Property, Admin Property, User Property, Client Property
- **Sensor Model**
  - Sensor, Sensor Setup
- **Time Model**
- **Scene Model**
  - Scene, Scene Setup
- **Scheduler Model**
  - Scheduler, Scheduler Setup
- **Light Models**
  - Light Lightness, Light Lightness Setup
  - Light CTL, Light CTL Setup
  - Light HSL, Light HSL Setup
  - Light XYL, Light XYL Setup
  - Light Control

You can add the required models in the FSP configuration. The health server model and configuration server model are defined as the Foundation models. Therefore, be sure to add the health server model and configuration server model.


BLE Abstraction (rm_ble_abs)
BLE Mesh Config Client (rm_mesh_config_clt)
BLE Mesh Model Generic Admin Property Server (rm_mesh_generic_admin_prop_srv)
BLE Mesh Model Generic Battery Client (rm_mesh_generic_battery_clt)
BLE Mesh Model Generic Battery Server (rm_mesh_generic_battery_srv)
BLE Mesh Model Generic Client Property Server (rm_mesh_generic_client_prop_srv)
BLE Mesh Model Generic Default Transition Time Client (rm_mesh_generic_dtt_clt)
BLE Mesh Model Generic Default Transition Time Server (rm_mesh_generic_dtt_srv)
BLE Mesh Model Generic Level Client (rm_mesh_generic_level_clt)
BLE Mesh Model Generic Level Server (rm_mesh_generic_level_srv)
BLE Mesh Model Generic Location Client (rm_mesh_generic_loc_clt)
BLE Mesh Model Generic Location Server (rm_mesh_generic_loc_srv)
BLE Mesh Model Generic Manufacturer Property Server (rm_mesh_generic_mfr_prop_srv)
BLE Mesh Model Generic On Off Client (rm_mesh_generic_on_off_clt)
BLE Mesh Model Generic On Off Server (rm_mesh_generic_on_off_srv)
BLE Mesh Model Generic Power Level Client (rm_mesh_generic_pl_clt)
BLE Mesh Model Generic Power Level Server (rm_mesh_generic_pl_srv)
BLE Mesh Model Generic Power On Off Client (rm_mesh_generic_poo_clt)
BLE Mesh Model Generic Power On Off Server (rm_mesh_generic_poo_srv)
BLE Mesh Model Generic Property Client (rm_mesh_generic_prop_clt)
BLE Mesh Model Generic User Property Server (rm_mesh_generic_user_prop_srv)
BLE Mesh Model Health Client (rm_mesh_health_clt)
BLE Mesh Model Light Control Client (rm_mesh_light_ctl_clt)
BLE Mesh Model Light Control Server (rm_mesh_light_ctl_srv)
BLE Mesh Model Light HSL Client (rm_mesh_light_hsl_clt)
BLE Mesh Model Light HSL Server (rm_mesh_light_hsl_srv)
BLE Mesh Model Light Lightness Client (rm_mesh_light_lightness_clt)
BLE Mesh Model Light Lightness Controller Client (rm_mesh_light_lc_clt)
BLE Mesh Model Light Lightness Controller Server (rm_mesh_light_lc_srv)
BLE Mesh Model Light Lightness Server (rm_mesh_light_lightness_srv)
BLE Mesh Model Light XYL Client (rm_mesh_light_xyl_clt)
BLE Mesh Model Light XYL Server (rm_mesh_light_xyl_srv)
BLE Mesh Model Scene Client (rm_mesh_scene_clt)
BLE Mesh Model Scene Server (rm_mesh_scene_srv)
BLE Mesh Model Scheduler Client (rm_mesh_scheduler_clt)
BLE Mesh Model Scheduler Server (rm_mesh_scheduler_srv)
BLE Mesh Model Sensor Client (rm_mesh_sensor_clt)
BLE Mesh Model Sensor Server (rm_mesh_sensor_srv)
BLE Mesh Model Time Client (rm_mesh_time_clt)
BLE Mesh Model Time Server (rm_mesh_time_srv)
SPP BLE Abstraction (rm_ble_abs_spp)
*[ra4w1_2] FSP Configuration
Stacks Configuration
Generate Project Content
Threads
New Thread
Remove
HAL/Common
BLE Mesh Config Server (rm_mesh_config_srv)
BLE Mesh Model Health Server (rm_mesh_health_srv)
BLE Mesh Bearer Platform (rm_mesh_bearer_platform)
Objects
New Object >
Remove
HAL/Common Stacks
New Stack >
Extend Stack >
Remove
BLE Mesh Config Server (rm_mesh_config_srv)
BLE Mesh Access (rm_ble_mesh_access)
BLE Mesh Upper Trans (rm_ble_mesh_upper_trans)
BLE Mesh Lower Trans (rm_ble_mesh_lower_trans)
Summary
BSP
Clocks
Pins
Interrupts
Event Links
Stacks
Components

Add models you want.

Figure 2 Add models

1.3 APIs

The Bluetooth mesh module provides APIs that you can use in your own application. For details of each API, refer to *Renesas Flexible Software Package User’s Manual*.

Table 1 Mesh API

Category	Module	Number of APIs
Mesh Core	rm_ble_mesh	8
	rm_ble_mesh_access	76
	rm_ble_mesh_upper_trans	19
	rm_ble_mesh_lower_trans	7
	rm_ble_mesh_network	14
	rm_ble_mesh_bearer	15
	rm_ble_mesh_provisioning	9
	rm_ble_mesh_bearer_platform	12
Mesh Model	rm_mesh_config_client/server	54
	rm_mesh_health_client/server	15
	rm_mesh_generic_*	122
	rm_mesh_light_*_client/server	107
	rm_mesh_scene_client/server	16
	rm_mesh_scheduler_client/server	12
	rm_mesh_sensor_client/server	20
	rm_mesh_time_client/sever	16

2. Application note

The following application notes for the Bluetooth Mesh are published in RA4W1 pages on the Renesas Web site.

Table 2 Application Notes for Bluetooth Mesh

Name of APN	APN #	Usage
Bluetooth Mesh Startup Guide	R01AN5847	• Describes how to work Bluetooth mesh sample application provided with R01AN5848. • Mesh mobile application.
Bluetooth Mesh sample application	R01AN5848	• Mesh sample application project for baremetal and FreeRTOS.
Bluetooth Mesh Development Guide	R01AN5849	• Briefly Explains Bluetooth mesh specification. • Describes how to make Bluetooth mesh application.

The *Bluetooth Mesh sample application* (R01AN5848) application note includes the following sample projects that can be used with the EK-RA4W1.

Table 3 Sample Projects included with the *Bluetooth Mesh sample application* (R01AN5848)

Name	Purpose
ekra4w1_mesh_server_baremetal	Generic ON/OFF server model sample application for baremetal environment.
ekra4w1_mesh_server_freertos	Generic ON/OFF server model sample application for FreeRTOS environment.
ekra4w1_mesh_client_baremetal	Generic ON/OFF client model sample application for baremetal environment.
ekra4w1_mesh_client_freertos	Generic ON/OFF server model sample application for FreeRTOS environment.
Mesh mobile application	• This is a smart phone application which performs as Provisioner, Configuration Client, Client for Generic ON/OFF model. • Includes *.apk file for Android environment. • Includes source codes which can be built for iOS environment. Refer to *Bluetooth Mesh Startup Guide* (R01AN5847) about required build environment.
ekra4w1_mesh_cli_server_baremetal	This sample application has command line interface. You can input the server side of commands for creating a mesh network, message exchange procedure via terminal emulator. This sample application can be used in a baremetal environment.
ekra4w1_mesh_cli_client_baremetal	This sample application has command line interface. You can input the client side of commands for creating a mesh network, message exchange procedure via terminal emulator. This sample application can be worked on baremetal environment

You can check the life cycle of a device in Bluetooth mesh network in several ways, such as joining Bluetooth mesh network, sending and receiving messages, and remove a device from Bluetooth mesh network by a using server, client and the mesh mobile application included with *Bluetooth Mesh sample application* (R01AN5848).

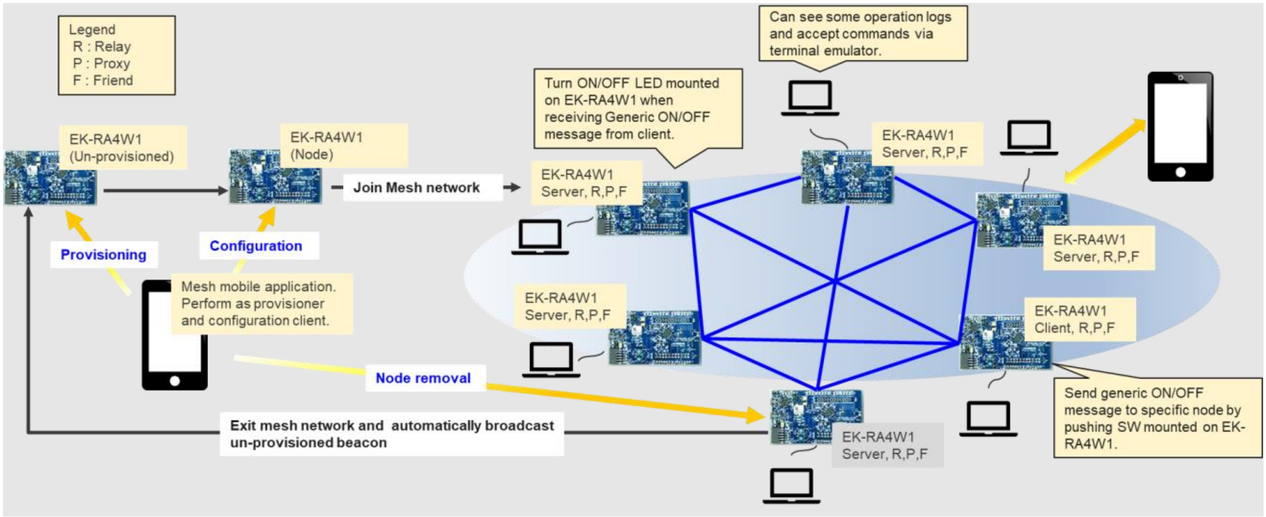

Legend
R : Relay
P : Proxy
F : Friend
EK-RA4W1 (Un-provisioned)
EK-RA4W1 (Node)
Join Mesh network
Provisioning
Configuration
Mesh mobile application. Perform as provisioner and configuration client.
Node removal
Exit mesh network and automatically broadcast un-provisioned beacon
Turn ON/OFF LED mounted on EK-RA4W1 when receiving Generic ON/OFF message from client.
Can see some operation logs and accept commands via terminal emulator.
EK-RA4W1 Server, R,P,F
EK-RA4W1 Server, R,P,F
EK-RA4W1 Server, R,P,F
EK-RA4W1 Server, R,P,F
EK-RA4W1 Client, R,P,F
EK-RA4W1 Server, R,P,F
Send generic ON/OFF message to specific node by pushing SW mounted on EK-RA4W1.

Figure 3 Mesh network that can be made using a Bluetooth mesh sample application

3. How to make a Bluetooth mesh user application

When creating a Bluetooth Mesh application, it is recommended that you import the sample project included with the *Bluetooth Mesh sample application* (R01AN5848) into your development environment based on the *Bluetooth Mesh Startup Guide* (R01AN5847) and add the necessary functions. In this section, the Generic Battery Model is added to the sample project included in the *Bluetooth Mesh sample application* (R01AN5848). Then, this new function can be used to acquire the Battery Status of the server by the client at any time.

- **Features to add**
  - **Server side**
    - Add the Generic Battery Server model.
    - Returns battery state containing dummy value in response to GET message from client.
  - **Client side**
    - Add the Generic Battery Client model.
    - Send GET message when pressing SW1 mounted on EK-RA4W1.
    - Display the acquired battery state from server on the terminal emulator.

3.1 Import sample project

Methods on how to import the following sample projects into your development environment is included with the *Bluetooth Mesh sample application* (R01AN5848) based on the *Bluetooth Mesh Startup Guide* (R01AN5847).

- **Baremetal environment**
  - ekra4w1_mesh_server_baremetal (Server side)
  - ekra4w1_mesh_client_baremetal (Client side)
- **FreeRTOS environment**
  - ekra4w1_mesh_server_freertos (Server side)
  - ekra4w1_mesh_client_freertos (Client side)

3.2 Open Generic Battery model

The Generic Battery model has been added to the sample project included with the *Bluetooth Mesh sample application* (R01AN5848). Open the Generic Battery model as shown below with the "*./src/mesh_model.c/mesh_application_model_register()*" function.

```
static API_RESULT mesh_application_model_register(void)
{
.........
    g_rm_mesh_generic_battery_srv0.p_api->open(g_rm_mesh_generic_battery_srv0.p_ctrl,
                                               g_rm_mesh_generic_battery_srv0.p_cfg);
   return retval;
}
```

Code 1 Example for server side

```
static API_RESULT mesh_application_model_register(void)
{
.........
    g_rm_mesh_generic_battery_clt0.p_api->open(g_rm_mesh_generic_battery_clt0.p_ctrl,
                                               g_rm_mesh_generic_battery_clt0.p_cfg);
   return retval;
}
```

Code 2 Example for client side

3.3 Implement server side of callback function for Generic Battery model

Add behavior when receiving GET message from client to server side of sample application. In the sample project included in the *Bluetooth Mesh sample application* (R01AN5848), the Callback function of the Generic Battery Model is specified as *appl_generic_battery_server_cb* function in the properties of the *rm_mesh_generic_battery_srv* module. The process of returning fixed value 0xAA as Battery status to the GET message from the client is added to the server side as follows.

```
void appl_generic_battery_server_cb(rm_ble_mesh_model_server_callback_args_t * p_args)
{
………
    /* Check message type */
    if (RM_BLE_MESH_ACCESS_MODEL_REQ_MSG_TYPE_GET == p_args->p_req_type->type)
    {
        CONSOLE_OUT(
        "[GENERIC_BATTERY] GET Request. ");
        appl_model_state_get(p_args->p_state_parameter->state_type, (uint16_t)elem_handle,
                                         &param, 0);
        param.generic_battery_level = 0xAA;


        current_state_params.state_type = p_args->p_state_parameter->state_type;
        current_state_params.state = &param;
………
    return retval;
}
```

Always respond “0xAA” as current battery level against GET message from client.

Code 3 Example of implementation

3.4 Send GET message

Add a function to send a GET message by pressing SW1 mounted on EK-RA4W1 to client side of sample application. In the sample project included in the *Bluetooth Mesh sample application* (R01AN5848), callback function for pressing SW1 has been already implemented as *Callback _ble_sw_irq* function. Add the following code to the callback function on the client side.

```
static void board_sw1_cb(void)
{
   .........
   gs_sw_ignore = false;
   RM_MESH_GENERIC_BATTERY_CLT_Get(&g_rm_mesh_generic_battery_clt0_ctrl);
}

void Callback_ble_sw_irq(external_irq_callback_args_t *p_args)
{
    FSP_PARAMETER_NOT_USED(p_args);
     .........
    R_BLE_SetEvent(board_sw1_cb);
}
```

Code 4 Example of implementation

3.5 Implement client side of callback function for Generic Battery model

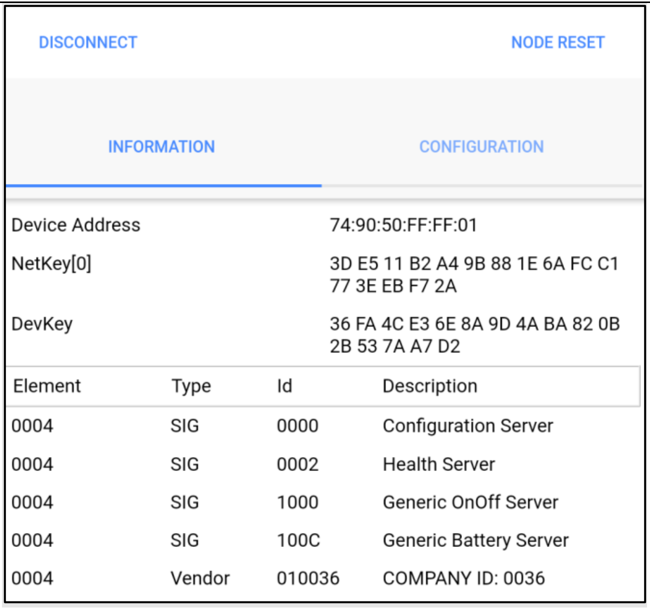
Add behavior to client side of sample application when receiving the response to GET message from server side. In the sample project included in the *Bluetooth Mesh sample application* (R01AN5848), the callback function is specified in *cli_generic_battery_client_cb* function in the property of *rm_mesh_generic_battery* module. Add a process to display the Battery status received from the server.

```
void cli_generic_battery_client_cb(rm_ble_mesh_model_client_callback_args_t * p_args)
{
     .........
    switch(p_args->p_msg_raw->opcode)
    {
        case RM_BLE_MESH_ACCESS_MESSAGE_OPCODE_GENERIC_BATTERY_STATUS:
            CONSOLE_OUT( "RM_BLE_MESH_ACCESS_MESSAGE_OPCODE_GENERIC_BATTERY_STATUS n");
            appl_dump_data(p_args->p_msg_raw->data_param, sizeof(uint8_t));
            break;
    }
.........
}
```

Code 5 Example of implementation

3.6 Behavior

If you add the implementations in the previous section and perform provisioning according to *Mesh Startup Guide* (R01AN5847), you can see the Generic Battery Model added to the *INFORMATION* tab of the Mesh mobile application.

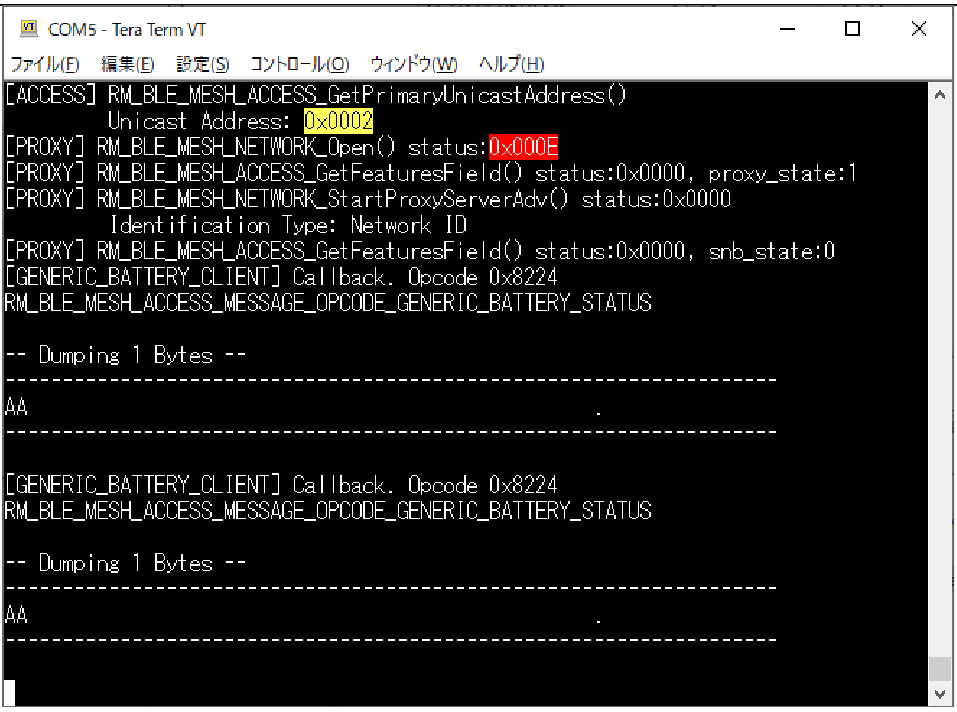


Device Address	74:90:50:FF:FF:01
NetKey[0]	3D E5 11 B2 A4 9B 88 1E 6A FC C1 77 3E EB F7 2A
DevKey	36 FA 4C E3 6E 8A 9D 4A BA 82 0B 2B 53 7A A7 D2

Element	Type	Id	Description
0004	SIG	0000	Configuration Server
0004	SIG	0002	Health Server
0004	SIG	1000	Generic OnOff Server
0004	SIG	100C	Generic Battery Server
0004	Vendor	010036	COMPANY ID: 0036

Figure 4 Mesh mobile application INFORMATION tab

After you have completed the configuration process according to *Mesh Startup Guide* (R01AN5847), you can get battery status from the server side by pressing SW1 mounted on Client side of EK-RA4W1 as shown below.


COM5 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)

```
[ACCESS] RM_BLE_MESH_ACCESS_GetPrimaryUnicastAddress()
         Unicast Address: 0x0002
[PROXY] RM_BLE_MESH_NETWORK_Open() status:0x000E
[PROXY] RM_BLE_MESH_ACCESS_GetFeaturesField() status:0x0000, proxy_state:1
[PROXY] RM_BLE_MESH_NETWORK_StartProxyServerAdv() status:0x0000
         Identification Type: Network ID
[PROXY] RM_BLE_MESH_ACCESS_GetFeaturesField() status:0x0000, snb_state:0
[GENERIC_BATTERY_CLIENT] Callback. Opcode 0x8224
RM_BLE_MESH_ACCESS_MESSAGE_OPCODE_GENERIC_BATTERY_STATUS

-- Dumping 1 Bytes --
-------------------------------------------------------------------
AA                                                   .
-------------------------------------------------------------------

[GENERIC_BATTERY_CLIENT] Callback. Opcode 0x8224
RM_BLE_MESH_ACCESS_MESSAGE_OPCODE_GENERIC_BATTERY_STATUS

-- Dumping 1 Bytes --
-------------------------------------------------------------------
AA                                                   .
-------------------------------------------------------------------
```

Figure 5 Terminal emulator of client side

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.

RA4W1 Group Bluetooth Mesh Stack uses the following open source software.
crackle; AES-CCM, AES-128bit functionality

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.0.0	Mar. 25. 2022	---	First edition.

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/.