

RA6W1 Getting Started with Matter

RA6W1 is a highly integrated ultra-low-power Wi-Fi® System on Chip (SoC) that allows you to develop a complete Wi-Fi solution on a single chip. This FSP guide is intended for developers who want to develop a Matter Wi-Fi device which is compatible with Matter Specification v1.4 using the RA6W1 EVB and FSP.

Contents

Contents	1
Figures	2
Tables	2
1. Terms and Definitions	3
2. References	3
3. Introduction	4
4. Hardware Setup	4
5. Software Setup	5
5.1 Install e2 studio IDE	5
5.2 Download RAFW Matter FSP	5
5.3 Load Matter Application Project Template in e2 studio	5
5.4 Change Matter Application in Reference Project	7
5.5 Program Firmware Image	9
5.6 Debug with J-Link Debug Probe	10
6. Matter Application Setup	11
6.1 Apple Home App	11
6.1.1 Wi-Fi Provisioning and Commissioning for Registering Device to Apple Home	11
6.1.2 Bluetooth LE Provisioning and Commissioning for Registering Device to Apple Home	12
6.2 Google Home Application	12
6.2.1 Wi-Fi Provisioning and Commissioning for Registering Device to Google Home	12
6.2.2 Bluetooth Provisioning & Commissioning for Registering Device to Google Home	12
6.2.3 Google Home Control	12
6.3 Amazon Alexa App	12
6.3.1 Amazon Alexa Control	13
6.4 Customize Application Development	13
6.4.1 Describe Reference Lock Application	13
6.4.2 App Development Flow	16
6.4.3 Set Clusters	16
6.4.4 Zap Code Generation	17
6.4.5 Matter DPM	18
6.5 Security and Certificate	19
6.6 AT Commands for Matter from External MCU	20
6.6.1 Overview	20
6.6.2 AT Command List	20
6.6.3 AT Command Examples for Lock Application	22

7. Revision History 23

Figures

Figure 1. RA6W1 Matter structure 4

Figure 2. Matter hardware configuration 4

Figure 3. Rename and import 5

Figure 4. Build project 6

Figure 5. Console window 6

Figure 6. Build successfully 6

Figure 7. Image in Project Explorer tab 7

Figure 8. Application files to update 7

Figure 9. C/C++ Project Settings 8

Figure 10. Zap CLI installation path 8

Figure 11. Pre-build script 9

Figure 12. Generated files. 9

Figure 13. Quick start 11

Figure 14. Bluetooth LE configuration property in e2 studio 14

Figure 15. Lock application structure 15

Figure 16. App development flow 16

Figure 17. Cluster selection 16

Figure 18. Generated Clusters 17

Figure 19. Pre-build step 18

Figure 20. rm_matter_app_port_w module 18

Figure 21. Property: Matter PMGR 19

Figure 22. With DPM 19

Figure 23. Without DPM 19

Figure 24. AT command flow for lock control 20

Figure 25. AT command flow for lock application 22

Tables

Table 1. AT commands from MCU to RA6W2 20

Table 2. AT commands from RA6W2 to MCU 21

1. Terms and Definitions

AP	Access Point
ACL	Access Control List
BDX	Bulk Data Exchange
Bluetooth® LE	Bluetooth Low Energy
CASE	Certificate Authenticated Session Establishment
CD	Certification Declaration
COM	Communication Port
CSA	Connectivity Standard Alliance
DAS	Device Attestation Certificate
DPM	Dynamic Power Management
EVB	Evaluation Board
EVK	Evaluation Kit
FSP	Flexible Software Package
IDE	Integrated Development Environment
IP	Internet Protocol
OTA	Over the Air
PAA	Product Attestation Authority
PAI	Product Attestation Intermediate
PASE	Passcode-Authenticated Session Establishment
PMGR	Power Manger
RTOS	Real Time Operating System
SDK	Software Development Kit
SFDP	Serial Flash Discoverable Parameter
SoC	System on Chip
SSID	Service Set Identifier
UART	Universal Asynchronous Receiver/Transmitter
UTC	Universal Time Coordinated
TH	Test Harness

2. References

- [1] RA6W1, Datasheet, Renesas Electronics.
- [2] RA6W1, JTAG Manual, Renesas Electronics.
- [3] RA6W2, Getting Started Guide, Manual, Renesas Electronics.
- [4] RA6W2, AT Commands Document, Renesas Electronics.
- [5] RA6W1, Matter Certification Document, Renesas Electronics
- [6] Matter Specification Version 1.4
- [7] QE for Bluetooth LE, Renesas 1.8.0

Note 1 References are for the latest published version, unless otherwise indicated.

3. Introduction

Matter is a unified, open-source application-layer connectivity standard built to enable developers and device manufacturers to connect and build reliable, and secure ecosystems and increase compatibility among connected home devices. By building upon Internet Protocol (IP), Matter enables communication across smart home devices, mobile apps, and cloud services. For more information about Matter, go to the official site (<https://project-chip-github.io/connectedhomeip-doc/index.html>).

The Matter platform is ported on the top of RAFW generic Flexible Software Package (FSP), and the applications are built on the top of Matter platform. Using the RAFW Matter FSP, the RA6W1 can be configured as a Matter accessory device which connects to the standard Matter network and can be controlled by Matter-enabled controllers. Figure 1 shows the structure of RA6W1 Matter.

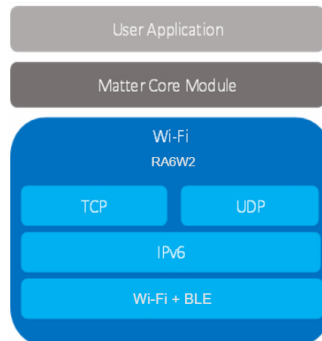


Figure 1. RA6W1 Matter structure

4. Hardware Setup

Figure 2 shows the basic element of Matter network (Hub, Device, and Controller) and how the network is linked together remotely. Matter accessory applications (lights and door locks) can be tested and developed on the device, RA6W1 Evaluation Board (EVB). To find out more about setting up the RA6W1 EVB for the Matter application, see the Setting Up EK-RA6W1 section of Ref. 3.



Figure 2. Matter hardware configuration

- **Hub:** Matter smart hubs such as Apple HomePod mini, Google Nest Hub, and Amazon Echo dot
- **Controller:** Mobile apps such as Apple Home, Google Home, and Amazon Alexa
- **Device:** RA6W1 EVB.

5. Software Setup

To develop Matter applications on the RA6W1 using the Matter module, install and set up the Renesas Electronics e² studio IDE.

To set up the development environments:

1. Install and configure the e² studio IDE.
2. Download the RAFW FSP from the Renesas Electronics website.
3. Load the Matter Application project template in e² studio.
4. Change the Matter Application in the reference project. Build the application project.
5. Program firmware image.
6. Debug with J-Link.

NOTE

For software setups, see [Ref. 2](#) and [Ref. 3](#).

5.1 Install e² studio IDE

To install the e² studio, download and run the e² studio installer on either Windows or Linux from the Renesas Electronics website (<https://www.renesas.com/us/en/software-tool/e-studio>). For more information, see the Install e² studio IDE for Windows and Linux sections of [Ref. 3](#).

5.2 Download RAFW Matter FSP

1. Download the following files from the Renesas Electronics website: RA FSP Pack - RAFW_FSP_Packs_<version>.zip.
2. Sample project template - <version>_rm_matter_wifi_lock_test_app.zip.
3. For using Bluetooth LE commissioning on RA6W2, use <version>_vndm_rm_matter_wifi_lock_test_app.zip.
4. For using AT commands for Matter, use <version>_rm_matter_wifi_atcmd_lock_test_app.zip.

5.3 Load Matter Application Project Template in e² studio

1. Extract the project template `rm_matter_wifi_lock_test_app.zip` and launch e² studio.
2. To import project template, go to **File > Import**.
3. Under **General** option, select **Rename & Import Existing C/C++ Project into Workspace**.

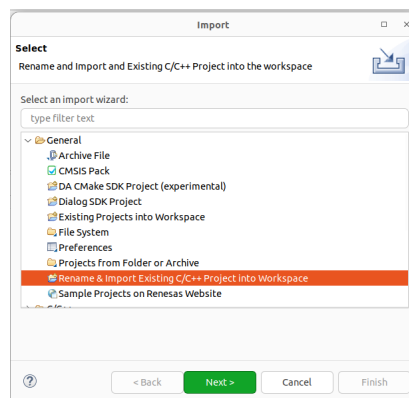


Figure 3. Rename and import

4. Browse to project template directory (unzip `rm_matter_wifi_lock_test_app.zip`) and enter **Project name**.
5. When the import is done, right-click on the project, and select **Build Project** to build the project.

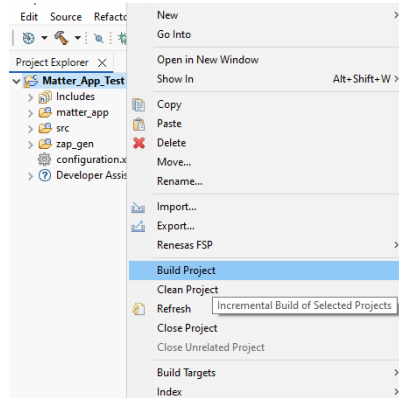


Figure 4. Build project

6. Check the console window to see if the build is proceeding.

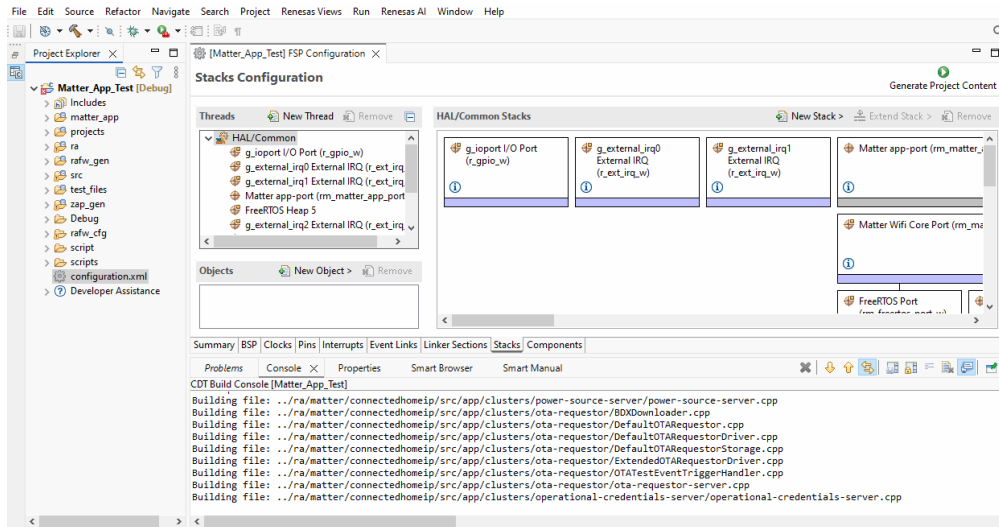


Figure 5. Console window

Figure 6 shows the build is completed successfully.

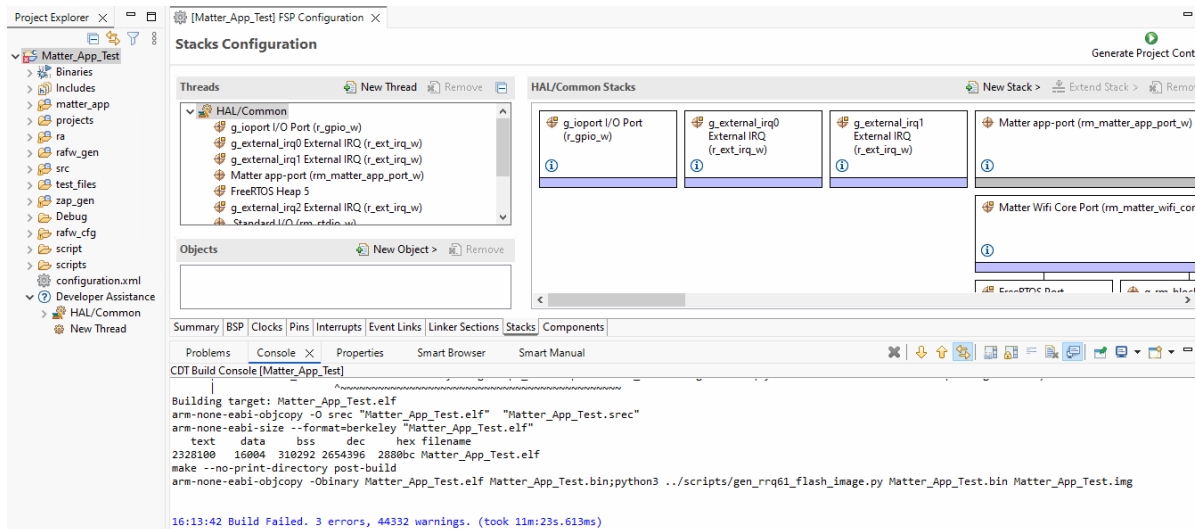


Figure 6. Build successfully

NOTE

The project **image** and **ELF** are generated correctly. After the build is completed, you may see three error messages in the output log. These are known **non-critical messages** and can be safely ignored, as they do **not** impact the final build output.

You can see the image in the project explorer as shown in [Figure 7](#).

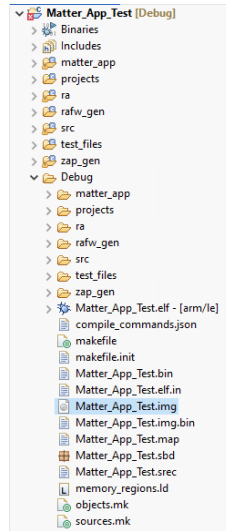


Figure 7. Image in Project Explorer tab

5.4 Change Matter Application in Reference Project

To change the Matter application, reconfigure the Matter application template project:

1. When the Matter application project is created, double click the `configuration.xml` file.
2. Choose the clusters needed for your application as described in [Section 6.4.4 Zap Code Generation](#).
3. Click **Generate Project Content**. If a dialog appears with an option to save the configuration change, click **Proceed**.

Instructions for updating the ZAP file using a reference configuration are provided in [Section 6 Matter Application Setup](#).

4. In `matter_app/src` and `matter_app/include`, update, add, or remove application files as needed to support your specific application requirements.

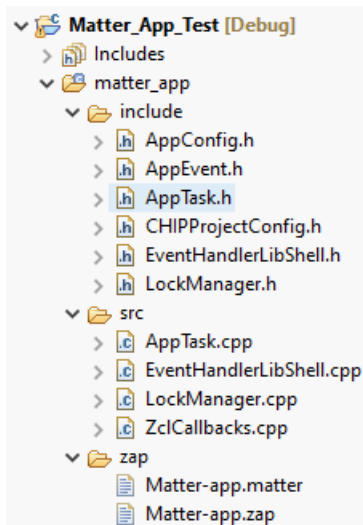


Figure 8. Application files to update

- In Project Explorer, right-click the project name and open **C/C++ Project Settings** as shown in Figure 9. A property window pops up.

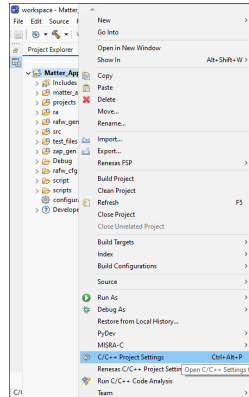


Figure 9. C/C++ Project Settings

- Go to C/C++ Build > Environment. Update the ZAP_INSTALL_PATH to point to the zap cli installation directory as shown in Figure 10.

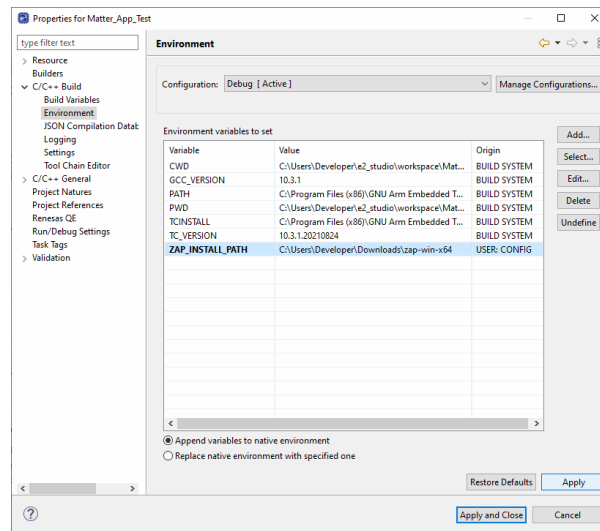


Figure 10. Zap CLI installation path

- Right-click the project name and select **Build Project** as shown in Figure 11. This regenerates `ra/matter_app/zap/Matter-app.matter` along with the ZAP-generated code in `zap-gen/app/` by running a pre-build script. See Figure 11 and Figure 12. The main build then proceeds and generates the `.img` file in the **Debug** folder — this is the normal flow.

NOTE

In e² studio, the refreshed files are applied only after the first build completes. Therefore, a **second** build is required for the newly generated files to be used in compilation.

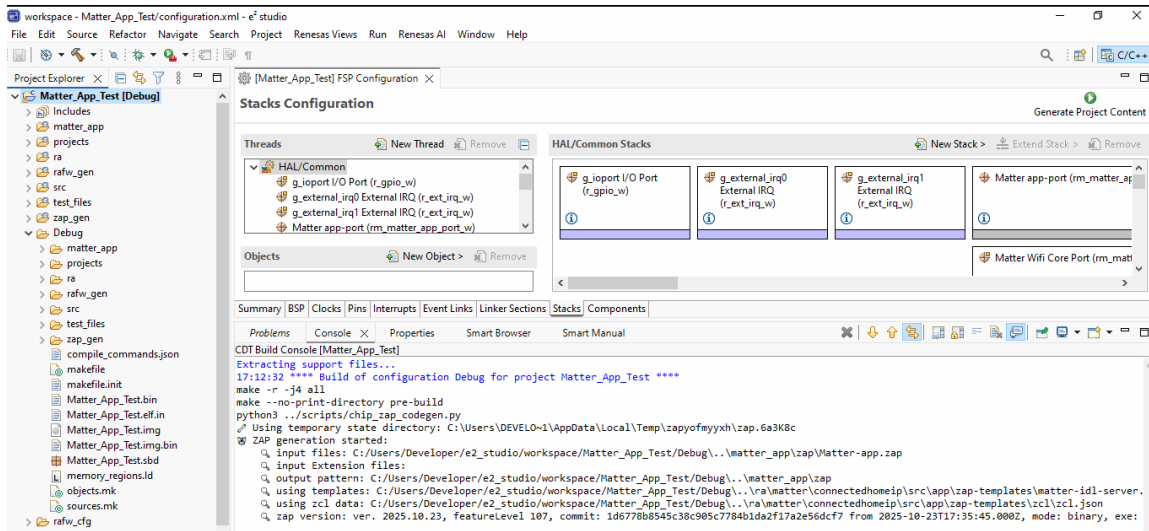


Figure 11. Pre-build script

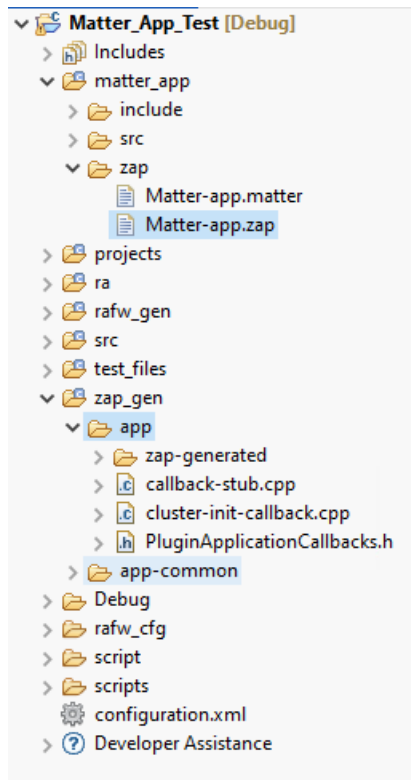


Figure 12. Generated files.

5.5 Program Firmware Image

The Matter lock application image is created in the e² studio project folder (e²_studio/Matter_App_Test /Debug/Matter_App_Test.img).

For more information about firmware downloading, see the Programming Firmware Images Using cli_programmer.exe section of Ref. 3.

5.6 Debug with J-Link Debug Probe

For more information about firmware downloading, see the e² studio Debug Mode with GDB of [Ref. 3](#).

To debug Matter application, complete the following steps after setting up the debugger:

1. Reset the target system by entering command: `monitor reset`
2. Set a breakpoint at the Matter entry function `MATTER_On` by entering command: `b MATTER_On`.
3. Start or continue execution by entering command: `c`

6. Matter Application Setup

Figure 13 shows how to start testing each application in the Matter environment.

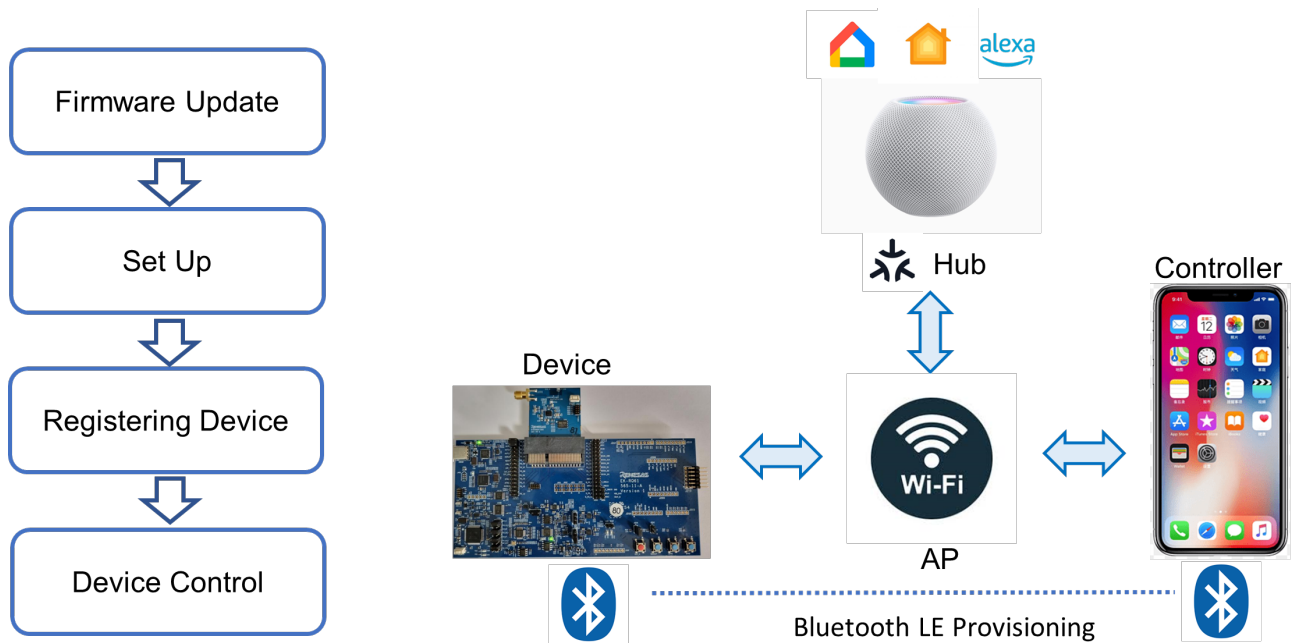


Figure 13. Quick start

- **Firmware update:** Matter application firmware image update for RA6W1.
- **Setup:** Smart Hub and mobile controller app configuration and connecting home AP.
- **Registering device:** RA6W1 pairing (commissioning) as Matter accessory.
- **Device control:** RA6W1 is controlled through mobile controller app or Hub.

6.1 Apple Home App

To set up HomePod mini to Apple Home app:

1. Get your iPhone or iPad ready to set up HomePod.
2. Plug HomePod into power and wait for a chime.
3. Hold your iPhone or iPad close to HomePod.
4. Hold your iPhone or iPad over HomePod.
5. Customize your HomePod settings.
6. Wait for setup and configuration to complete.

For more information on HomePod mini setup, go to the official website (<https://support.apple.com/en-us/111110>).

To pair and manage your Matter accessories:

1. Pair a Matter accessory.
2. Add a Matter accessory to the Home app.
3. View your paired Matter accessories.
4. Remove Matter accessories.

For more information on pair Matter accessories, go to the official website (<https://support.apple.com/en-us/102135>).

6.1.1 Wi-Fi Provisioning and Commissioning for Registering Device to Apple Home

For using Wi-Fi provisioning and commissioning with RA6W1, you need to flash RA6W1 with the image of `rm_matter_wifi_lock_test_app template`. To register an accessory device to Home App, connect the RA6W1 using manual code/scan the QR code which is shared with product. During this process, the RA6W1 is connected to AP. Then, the device is registered to a Home application.

6.1.2 Bluetooth LE Provisioning and Commissioning for Registering Device to Apple Home

For using Bluetooth LE provisioning and commissioning with RA6W2, you need to flash RA6W2 with the image of **vndm_rm_matter_wifi_lock_test_app template**. The Bluetooth LE provisioning and commissioning process starts automatically when you try to add matter device to the controller. During this process, AP information is transferred to the RA6W2 over Bluetooth LE network, and the RA6W2 can be connected to AP. Then, the device is registered to the Home application.

6.2 Google Home Application

Google Nest needs to create a developer project and Matter integration in advance:

- Create a developer project (<https://developers.home.google.com/matter/project/create>)
- Create a Matter integration (<https://developers.home.google.com/matter/integration/create>)

To register Google Nest to Google Home app:

- Set up your Google Nest or Home speaker or display (<https://support.google.com/googlenest/answer/7029485?hl=en&sjid=5926392987998707780-AP>)
- Pair a Matter Device (<https://developers.home.google.com/matter/integration/pair>)

6.2.1 Wi-Fi Provisioning and Commissioning for Registering Device to Google Home

Procedure is same as in [Section 6.1.1 Wi-Fi Provisioning and Commissioning for Registering Device to Apple Home](#).

6.2.2 Bluetooth Provisioning & Commissioning for Registering Device to Google Home

Procedure is same as in [Section 6.1.2 Bluetooth LE Provisioning and Commissioning for Registering Device to Apple Home](#).

6.2.3 Google Home Control

After setting up the device in the Google Home app, the Controller can control the accessory.

1. Open the Google Home app.
2. Tap Favourites or Devices and find the tile for your device.
3. Tap or drag the tile to take actions such as turning lights on and off or locking and unlocking a door.

6.3 Amazon Alexa App

To register Amazon smart speaker (Echo Dot 5th) to Amazon Alexa:

1. Search for Amazon Alexa in the Google Play Store and install the Alexa app.
2. Sign in with your Amazon account or create a new Amazon account if you do not have one.
3. To connect Matter device with Alexa:
 - a. Open the Alexa app.
 - b. Open **Devices**.
 - c. Select the plus icon.
 - d. Select **Add Device**.
 - e. Select the type of smart home device that you want to connect.
 - f. Select the brand and follow the on-screen instructions.

For more information, see the official documents about Amazon Alexa (<https://www.amazon.-com/gp/help/customer/display.html?nodeId=G3RKPNRKF33ECTW7>).

NOTE

For Wi-Fi/Bluetooth LE provisioning and scanning QR code of the RA6W1, see [Section 6.1.1 Wi-Fi Provisioning and Commissioning for Registering Device to Apple Home](#) and [Section 6.1.2 Bluetooth LE Provisioning and Commissioning for Registering Device to Apple Home](#).

6.3.1 Amazon Alexa Control

After setting up the device in the Amazon Alexa application², the Controller can control the accessory.

1. Open the Alexa application.
2. Tap the icon of the device.
3. Tap or drag the device to take actions such as turning the lights on and off or lock and unlock a door.

6.4 Customize Application Development

You can create Matter device type application.

6.4.1 Describe Reference Lock Application

The lock application consists of three parts as shown [Figure 15](#).

6.4.1.1 Wi-Fi Configuration

In the NVRAM, there is a section called **WIFI PROFILE** where Wi-Fi configuration parameters can be stored. Upon rebooting, the application checks to see if a saved profile exists. If a saved profile exists, it loads the configuration from the WIFI PROFILE; otherwise, it uses hardcoded default values. These values are then saved to the WIFI PROFILE for future use. Currently, the application supports only WPA2 Security mode.

6.4.1.2 Bluetooth® LE

During Bluetooth LE commissioning, the device advertises itself over Bluetooth LE and can be discovered by a Matter controller (for example, chip-tool or mobile app) using the Bluetooth LE advertisement. The controller connects to the device over Bluetooth LE and exchanges information, such as setup codes and network credentials. Once provisioning is complete, the device uses these credentials to join the network.

The Bluetooth LE transport is only used for the commissioning process. After successful provisioning, communication with the device continues over Wi-Fi.

To enable Bluetooth LE Commissioning:

1. Use the QE for Bluetooth LE tool to create the required Bluetooth LE profile.
2. Import the generated profile into the project.
3. In the Matter module (`rm_matter_app_port_w`) configuration, under **Properties** tab, enable the **CHIPOBLE** option to allow the Matter stack to use Bluetooth LE for commissioning and device communication.

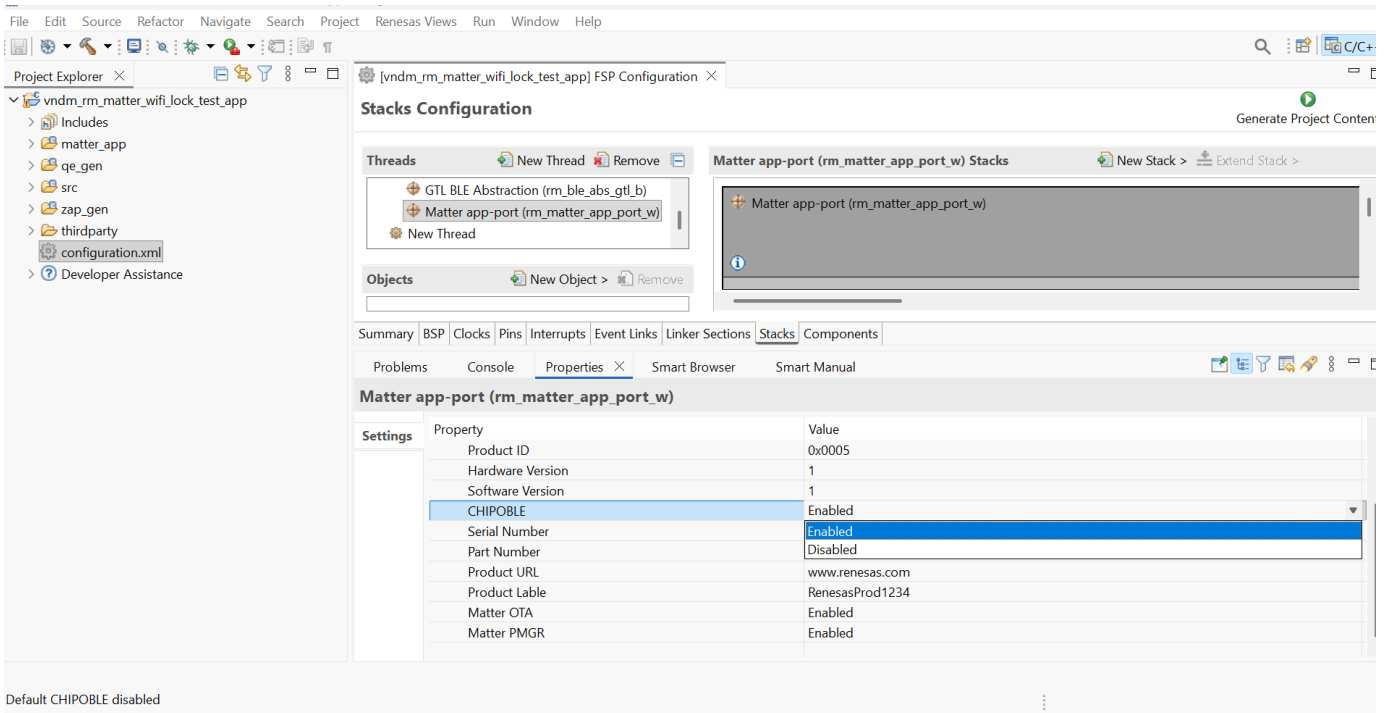


Figure 14. Bluetooth LE configuration property in e² studio

6.4.1.3 Application Main Task

This is the main task where the lock device type is initialized.

- Base Application:** the Base Application class provides functions to set and get basic information such as serial number, vendor ID, product ID, product name, hardware version, pin code, discriminator manufacturing date, device type, QR code, manual pairing code, and information for commissioning.
- Lock Manager:** the Lock Manager class provides functions for managing the lock device. This includes configuring, initializing, registering callback functions for receiving messages from the Matter core, and registering timers.
- Attribute Change Callback:** the Attribute Change Callback is called by the Matter application framework when one of the lock device attribute values changes. This is where the application performs an action based on the value of the attribute.

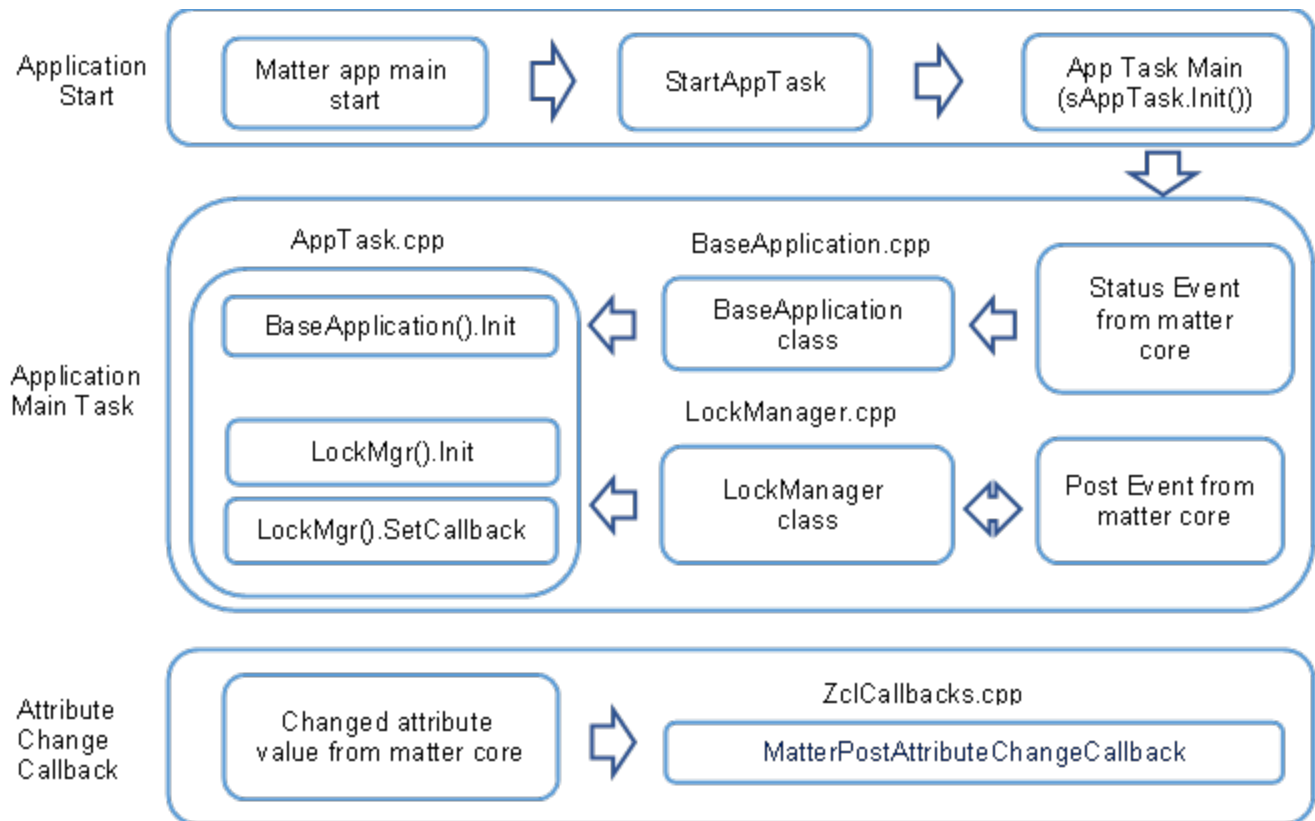


Figure 15. Lock application structure

6.4.2 App Development Flow

Figure 16 shows the development flow of the application.

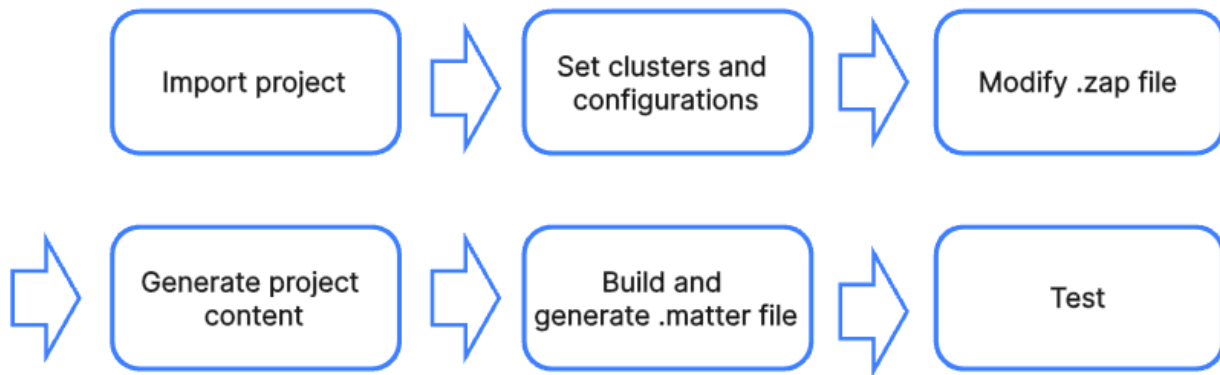


Figure 16. App development flow

6.4.3 Set Clusters

In case of Renesas FSP, modules are created for each cluster:

1. In e² studio, on the **Components Configuration** tab, select the required clusters components.

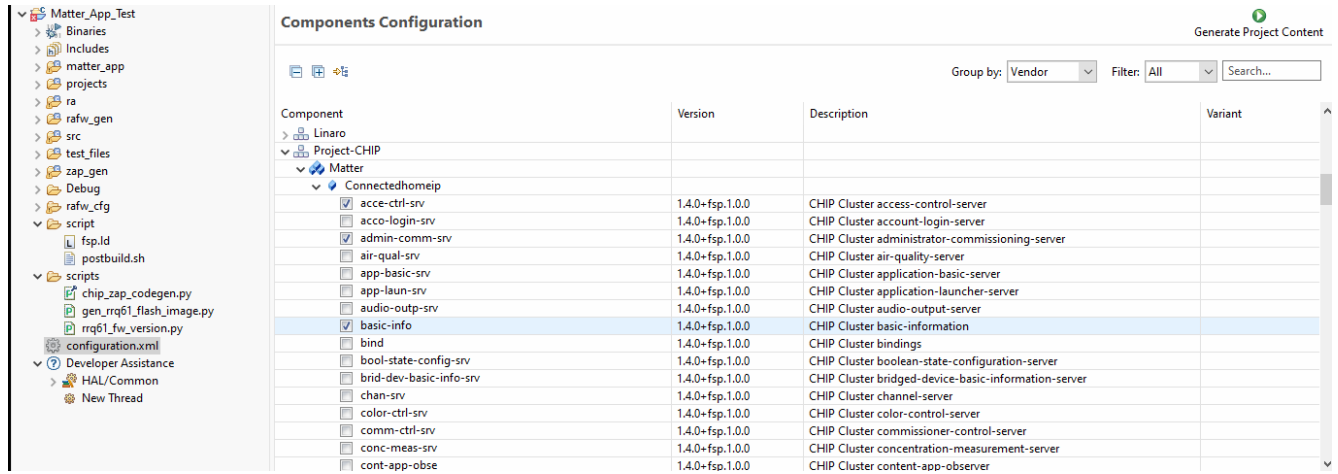


Figure 17. Cluster selection

2. Click **Generate Project Content**.

Corresponding cluster code is retrieved from Matter FSP pack and made available under ra/matter/connectedhomeip/src/app/clusters.

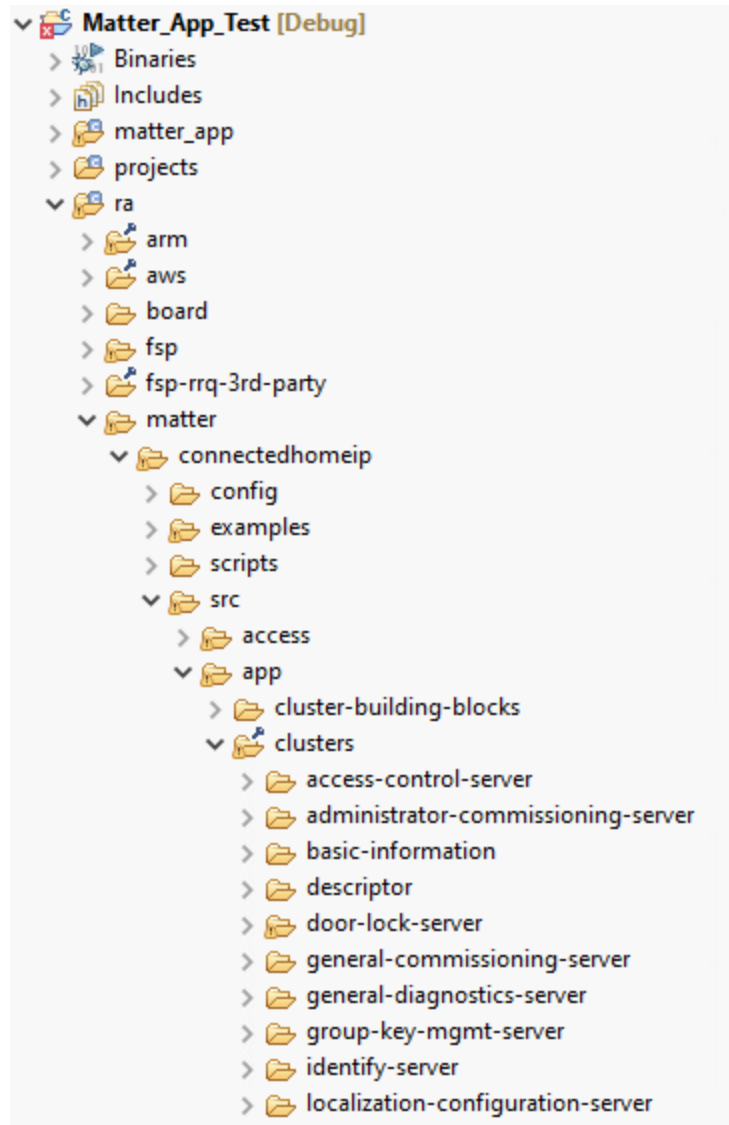


Figure 18. Generated Clusters

6.4.4 Zap Code Generation

1. Using the ZAP tool, modify the .zap file based on your application requirements.
A reference Matter-app.zap file is available under ra/matter_app/zap.
2. Run the ZAP tool from the connectedhomeip GitHub repository:
 - a. Follow the steps in the Documentation: https://project-chip.github.io/connectedhomeip-doc/zap_and_codegen/zap_intro.html.
 - b. To launch the tool enter the command

```
./scripts/tools/zap/run_zaptool.sh
```

3. After modification, copy the updated .zap file into ra/matter_app/zap/Matter-app.zap.
4. The Python script scripts/chip_zap_codegen.py runs automatically as a **pre-build step** in e² studio.
5. Right-click the project and select **Properties**.
6. In the **Properties for Matter_App_Test** dialog, go to **C/C++ Build > Settings**.
7. On the **Build Steps** tab, in **Pre-build steps**, check that the **Command** field has the needed command.

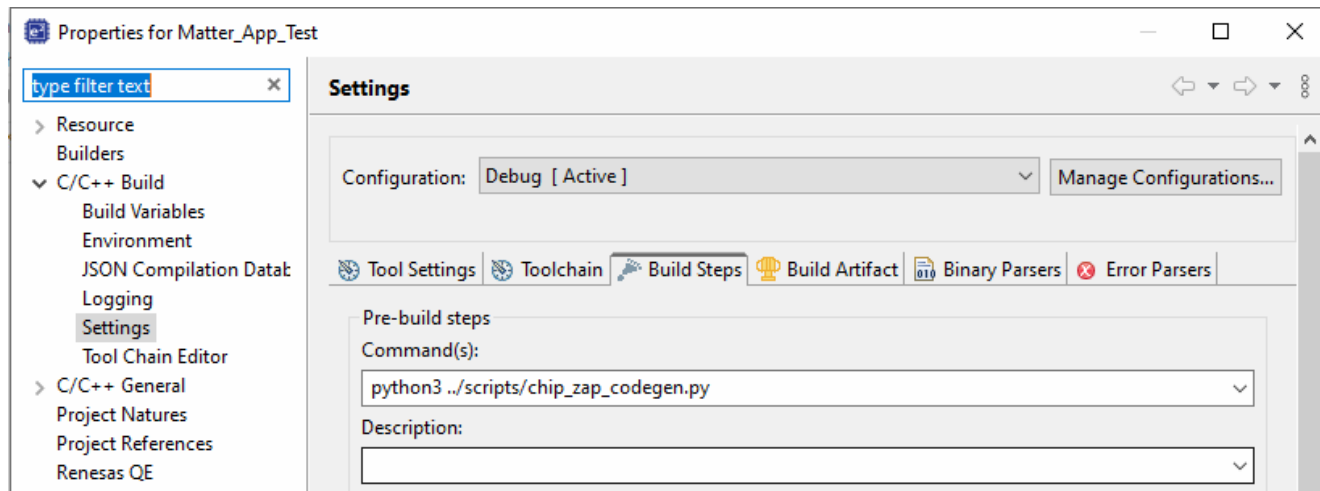


Figure 19. Pre-build step

8. In e2 studio, right-click the project name and select **Build Project**. The script generates these files from from the existing `matter_app/zap/Matter-app.zap` (see [Section 5.4 Change Matter Application in Reference Project](#)):

- `matter_app/zap/Matter-app.matter` – human-readable file.
- `zap_gen/app` – application-specific ZAP-generated source files.
- `zap_gen/app/ zap-generated` – auto-generated Matter data model and cluster implementation files.

6.4.5 Matter DPM

Power Manager (PMGR) is the power management module of RA6W1. It is used to configure Dynamic Power Management (DPM) mode for optimizing energy consumption for devices. Matter uses the configuration parameter **Matter PMGR** to enable or disable DPM for Matter. By default, Matter PMGR is enabled in the Matter Application template. To disable Matter PMGR:

1. Double-click `configuration.xml`.
2. Click **Matter app-port(rm_matter_app_port_w)**.

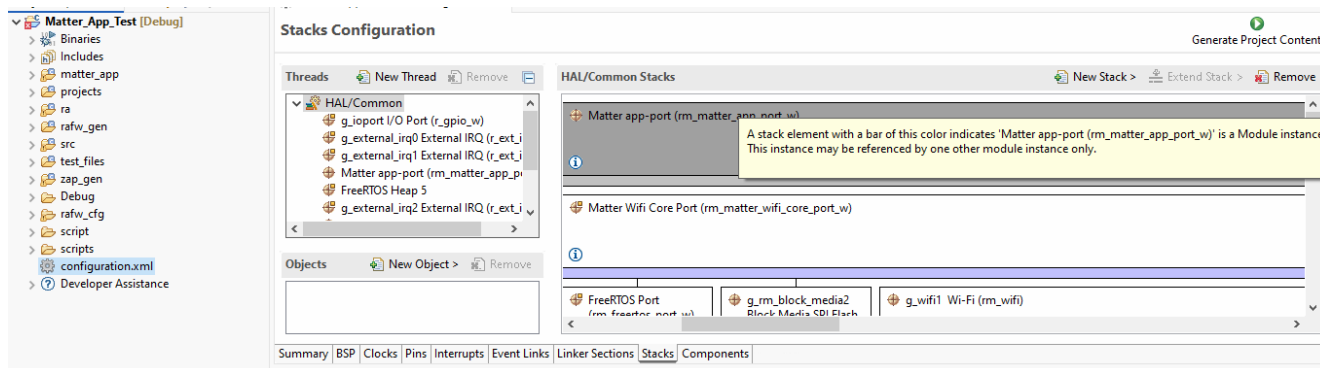


Figure 20. `rm_matter_app_port_w` module

3. In **Stacks > Properties** window, for **Matter PMGR** select **Disabled**.

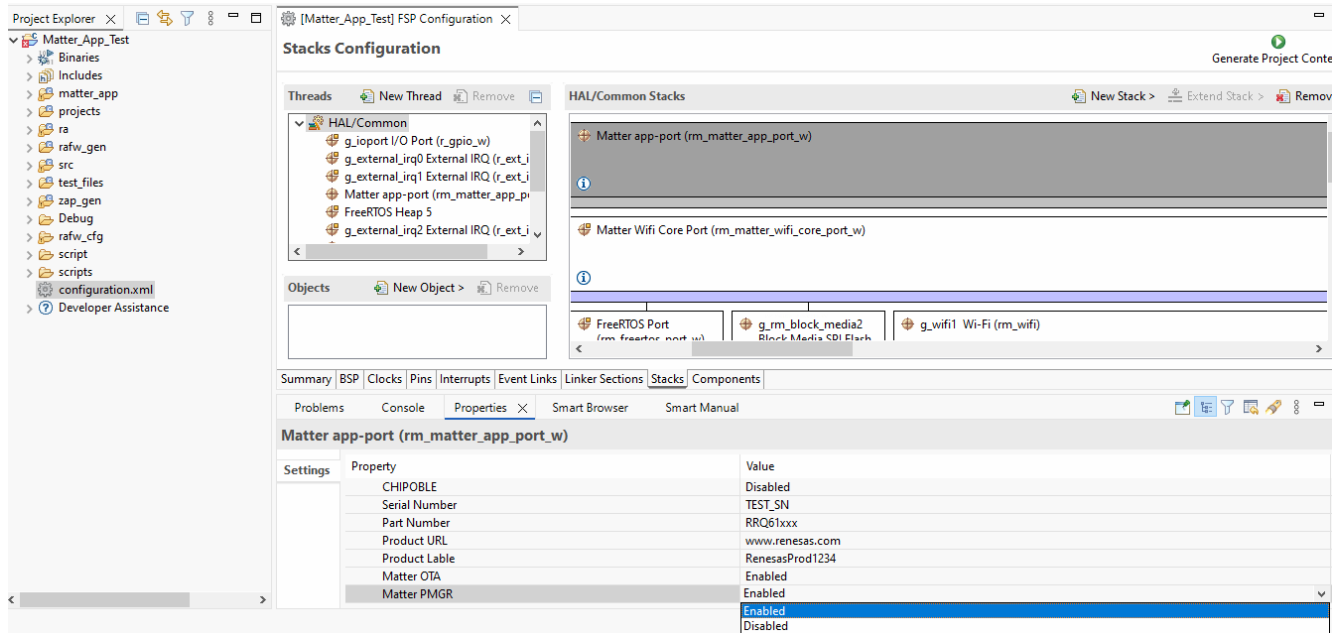


Figure 21. Property: Matter PMGR

4. Click **Generate Project content**.
5. Right-click the project name and select **Build Project**.

With Matter PMGR enabled, the RA6W1 Lock enters DPM mode after commissioning, wakes up upon receiving a unicast packet, and performs the lock or unlock operation.

From the application's perspective, the only difference when DPM is enabled is a change in the sequence flow. Figure 22 shows that DPM is enabled so that the Server Ready Event is sent only after the application started. When DPM is disabled, Server Ready Event is sent before the application starts, during server initialization as shown in Figure 23.



Figure 22. With DPM



Figure 23. Without DPM

6.5 Security and Certificate

Certificates and keys are important for secured communication in Matter. For details, see [Ref. 6](#).

6.6 AT Commands for Matter from External MCU

6.6.1 Overview

External MCU can control RA6W1 using AT commands. Basically, AT commands are available for all Device type of Matter using Device Type and Cluster ID and Attribute ID. For AT UART pin configuration, see Introduction of [Ref. 4](#). [Figure 24](#) shows the AT command flow for lock and unlock control.

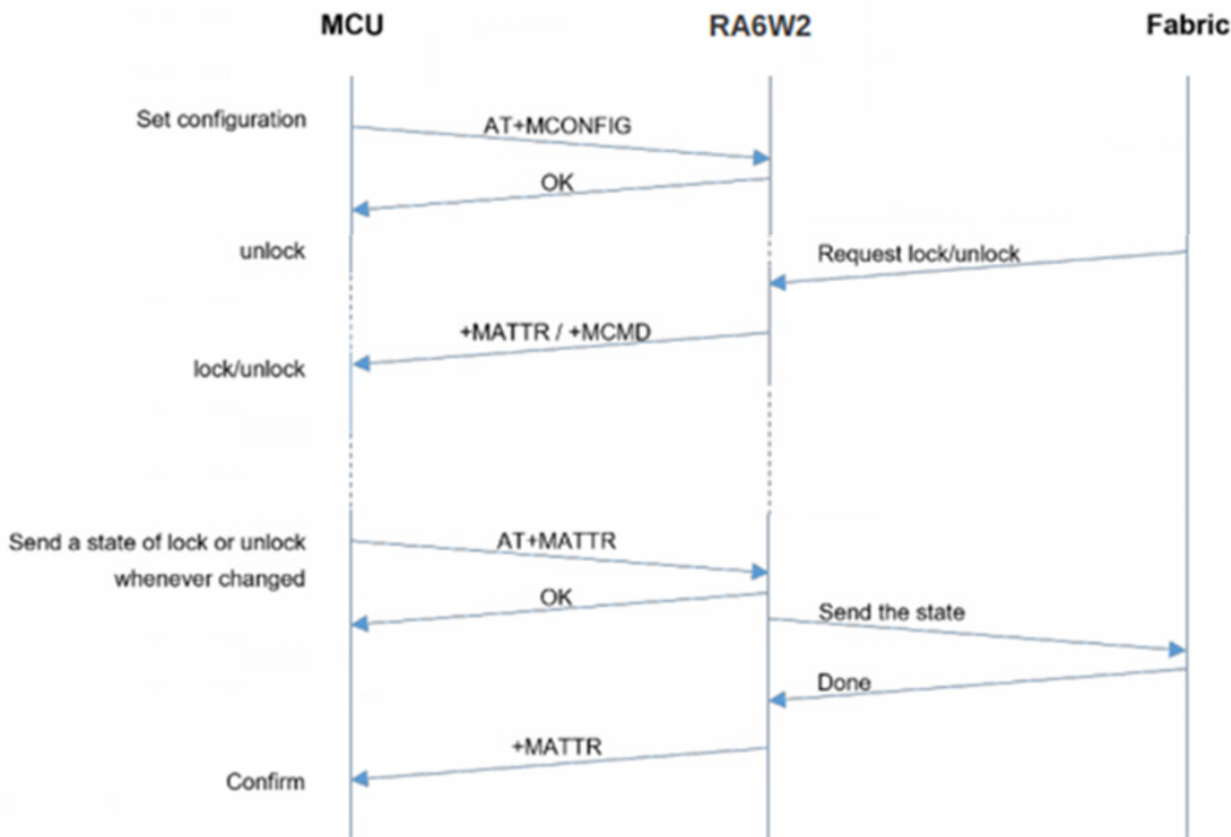


Figure 24. AT command flow for lock control

6.6.2 AT Command List

Table 1. AT commands from MCU to RA6W2

Command	Parameters	Description
AT+MCONFIG	<parameter>,<value>	Sets configurations for commissioning. <parameter>: configurable parameter <ul style="list-style-type: none"> ▪ DISC: discriminator ▪ VID: vendor ID ▪ PID: product ID ▪ HWVER: Hardware version ▪ SPKPCNT: spake2p iteration count ▪ SPKPSALT: spake2p salt ▪ SPKPVF: spake2p verifier ▪ PINCODE: pin code ▪ DEVTYPE: matter device type <value>: data of string type <ul style="list-style-type: none"> ▪ '?' for <value>: is responded current value of the parameter with "+MSTATUS-S=CFG"

Command	Parameters	Description
		Response: OK or ERROR
	<p>Example</p> <pre>AT+MCONFIG=DISC,3840 OK AT+MCONFIG=VID,FFF1 OK AT+MCONFIG=DISC,? +MSTATUS=CFG,DISC,3840 OK</pre>	
AT+MATTR	<p><endpoint-id>, <cluster-id>, <attribute-id>, <attribute-data>, <data-type(w),data-length (r)>, <write/read></p>	<p><endpoint-id>: device endpoint id <cluster-id>: cluster id <attribute-id>: attribute id <attribute-data>: attribute data <data-type(w), data-length(r)>: data type for write, data length for read <write/read>: write or read to attribute</p>
	<p>Example</p> <pre>//write lock to lockstate_attribute for lock cluster AT+MATTR=1,256,0,1,36,1 //read 1 byte from lockstate_attribute for lock cluster AT+MATTR=1,256,0,0,1,0</pre>	

Table 2. AT commands from RA6W2 to MCU

Command	Parameters	Description
AT+MCONFIG	<parameter>,<value>	<p>Sets configurations for commissioning.</p> <p><parameter>: configurable parameter</p> <ul style="list-style-type: none"> ▪ DISC: discriminator ▪ VID: vendor ID ▪ PID: product ID ▪ HWVER: Hardware version ▪ SPKPCNT: spake2p iteration count ▪ SPKPSALT: spake2p salt ▪ SPKPVF: spake2p verifier ▪ PINCODE: pin code ▪ DEVTYPE: matter device type <p><value>: data of string type</p> <ul style="list-style-type: none"> ▪ '?' for <value>: is responded current value of the parameter with "+MSTATUS-S=CFG" <p>Response: OK or ERROR</p>
	<p>Example</p> <pre>AT+MCONFIG=DISC,3840 OK AT+MCONFIG=VID,FFF1 OK AT+MCONFIG=DISC,? +MSTATUS=CFG,DISC,3840 OK</pre>	
AT+MATTR	<p><endpoint-id>, <cluster-id>, <attribute-id>,</p>	<p><endpoint-id>: device endpoint id <cluster-id>: cluster id <attribute-id>: attribute id</p>

Command	Parameters	Description
	<attribute-data>, <data-type(w),data-length (r)>, <write/read>	<attribute-data>: attribute data <data-type(w), data-length(r)>: data type for write, data length for read <write/read>: write or read to attribute
	Example //write lock to lockstate_attribute for lock cluster AT+MATTR=1,256,0,1,36,1 //read 1 byte from lockstate_attribute for lock cluster AT+MATTR=1,256,0,0,1,0	

6.6.3 AT Command Examples for Lock Application

Figure 25 shows the actual flow detailed for lock application.

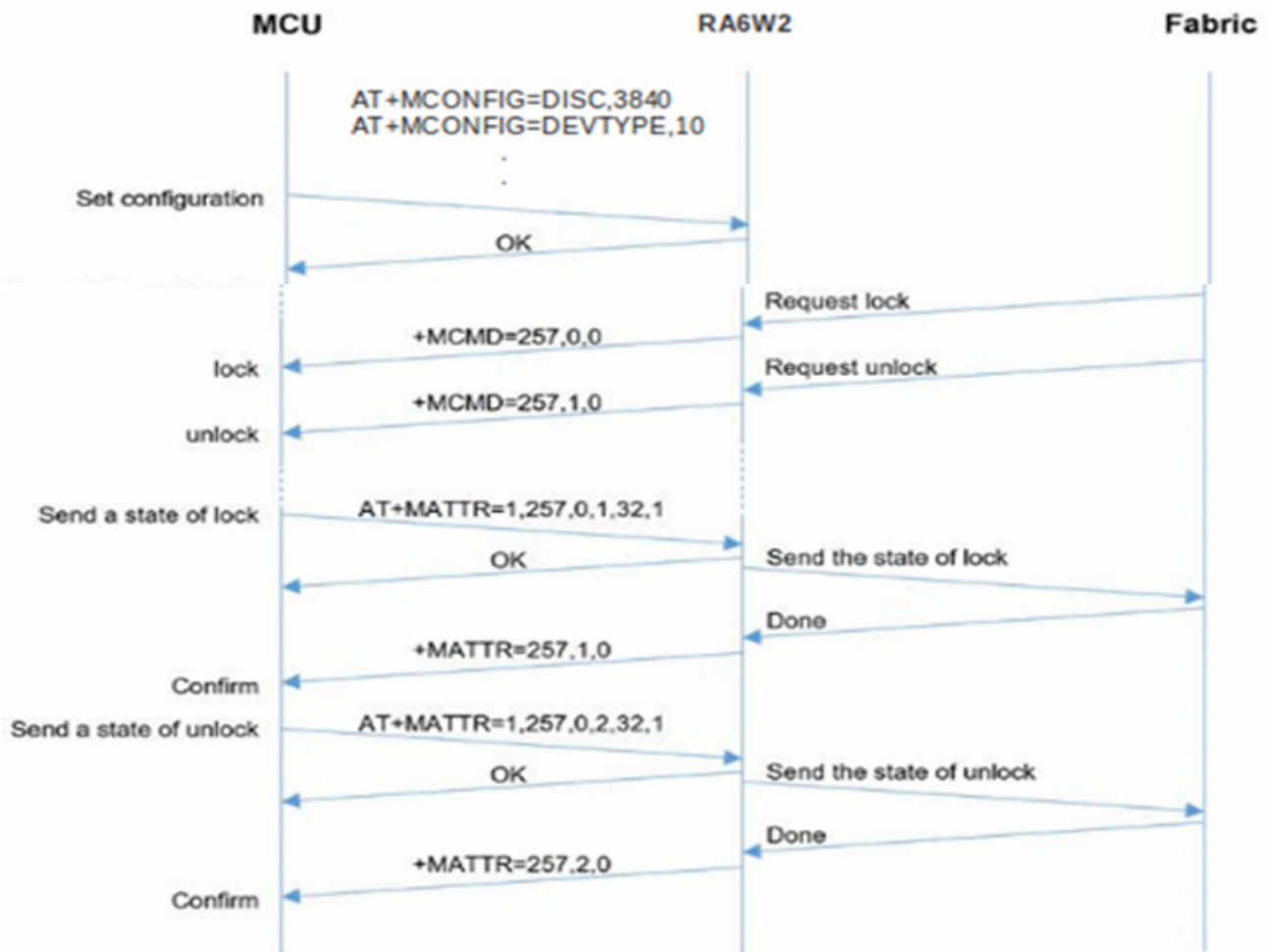


Figure 25. AT command flow for lock application

7. Revision History

Revision	Date	Description
1.03	Nov 30, 2025	Updated the new matter module structure in sections about software setup and customizing application development.
1.02	Aug 31, 2025	Updated Device name, M R19US0027EK0102 after version and added Bluetooth LE commissioning option.
1.01	Mar 13, 2025	Added the section of Matter DPM.
1.00	Dec 30, 2024	First release.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

RoHS Compliance

Renesas Electronics' suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.1 Jan 2024)

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-us/

© 2025 Renesas Electronics Corporation. All rights reserved.