DA16200/DA16600 Getting Started with AWS IoT Core

The DA16200/DA16600 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 document is a DA16200/DA16600 manual intended to help new or existing developers quickly get started using AWS IoT Core.

Contents

Со	ntents	\$		1
Fig	ures.			2
Tal	oles			5
1.	Term	ns and D	efinitions	6
2.	Refe	rences .		6
3.	Over	view		7
	3.1	Benefits	5	7
	3.2	Feature	S	7
	3.3	Applicat	ions	7
4.	Hard	lware De	scription	8
	4.1	Datashe	eet	8
		4.1.1	DA16200MOD	8
		4.1.2	DA16600MOD	8
5.	Set l	Jp Devel	opment Environment	8
6.	AWS	6 IoT		9
	6.1	Configu	re AWS IoT	9
		6.1.1	Sign Up for AWS Account	9
		6.1.2	Connect Devices to AWS IoT	9
		6.1.3	Configure Amazon Cognito	
		6.1.4	Set Up AWS IAM	35
		6.1.5	Create Amazon S3 Bucket	37
7.	Build	d and Ru	In Reference Application	38
	7.1	Referen	ce DA16200/DA16600 SDK Setting	38
		7.1.1	Edit Endpoint	38
		7.1.2	Edit Thing Name	38
		7.1.3	Edit Image File Name for OTA	39
		7.1.4	Connect Certificates to Thing	39
	7.2	Referen	ce Application in DA16200/DA16600	40
		7.2.1	Open Door	41
		7.2.2	Close Door	43
	7.3	Referen	ce Application in Host MCU	46
		7.3.1	Download Package for Door Lock Reference Application in Host MCU	46
		7.3.2	Hardware Connections between DA16200/DA16600 and Host MCU	46
		7.3.3	Programming Firmware Images for DA16200/DA16600	51
		7.3.4	Configure Components for Testing	54

		7.3.5	Test without Host MCU	
		7.3.6	Test with Host MCU	
	7.4	Mobile A	pp Demo	61
		7.4.1	Open Door	61
		7.4.2	Close Door	
8.	ΟΤΑ	Update		
	8.1	Create S	3 Bucket	
	8.2	Upload II	mage File and JSON File	
	8.3	Create J	ob	
	8.4	Execute	OTA Update	
9.	Priva	ate S3 Do	wnload Demo	
	9.1	Sign Up	for AWS Account	
	9.2	Create S	3 Bucket	
	9.3	Create A	WS IoT Thing and Certificate	83
	9.4	Create P	Policies	
	9.5	Create a	nd Configure IAM Role	
	9.6	Create a	nd Configure IAM User	
	9.7	Create R	cole Alias	
	9.8	Attach R	ole Alias Policy to Certificate	89
	9.9	Request	Security Token	
	9.10	Code Co	nfiguration	
	9.11	Testing [Demo	
Ap	pendi	x A Provi	sioning	
	A.1	Android /	Application	
Ap	pendi	x B AT Co	ommands for AWS IoT	
	B.1	Operatin	g Modes	
		B.1.1	Setting Mode	
	B.2	Provisior	ning Mode	
		B.2.1	Communication Mode	
	B.3	Configur	ing Topic to Publish, Subscribe, and Shadow	
		B.3.1	Configure Topics	
	B.4	AT Com	mand List	
		B.4.1	Basic Set	
		B.4.2	TLS Certificate	
		B.4.3	PIN MUX	
		B.4.4	Configure Data as Topics	100
		B.4.5	Command – MCU to DA16200/DA16600	100
		B.4.6	Command – DA16200/DA16600 to MCU	101
		B.4.7	DA16200/DA16600 Status - DA16200/DA16600 to MCU	101
Ap	pendi	x C Troub	bleshooting	102
	C.1	Operatio	nal Issue	102
10.	Revi	sion Hist	ory	103

Figures

Figure 1.	Sign up for AWS	S accountS)
-----------	-----------------	------------	---

Figure 2. I	Register things	10
Figure 3. (Create single thing	10
Figure 4.	Ching name	11
Figure 5.	hing without certificate	12
Figure 6. 0	Created thing	12
Figure 7. (Classic shadow	13
Figure 8. I	Device shadow document	14
Figure 9. 0		15
Figure 10.	Create certificates (continued)	15
Figure 11.	Download certificates and keys	16
Figure 12.	Activate certificate	16
Figure 13.	Create policy	17
Figure 14.	Add policy name	17
Figure 15.	Enter JSON policy statement	18
Figure 16.	Created policy	19
Figure 17.	Check created policy	19
Figure 18.		20
Figure 19.	Attach policy	20
Figure 20.	Attach things to certificate	21
Figure 21.	Attach to thing	21
Figure 22.		22
Figure 23.	Specify rule name	22
Figure 24.	Configure SQL statement	23
Figure 25.	Attach rule actions	23
Figure 26.	Attach rule actions (continued)	24
Figure 27.		24
Figure 28.	Crooted rule	25
Figure 29.		20
Figure 30.		20
Figure 31.		20
Figure 32.	Attached policies	21 27
Figure 33.	Create year peal	21 20
Figure 34.	Create user pool	20 20
Figure 35.	Configure sign-in options	20 20
Figure 30.	Configure security requirements	29 20
Figure 37.	Configure sign-up experience	3U 21
Figure 30.	Lotarate app client	31 22
Figure 39.	Created user peel	32 22
Figure 40.	Create identity pool	22
Figure 41.	Create identity pool trust	33 22
Figure 42.	Configure permissions	37 27
Figure 43.	Configure properties	24
Figure 44.	Configure properties	25
Figure 45.		35
Figure 40.	Attach policies	36
Figure 47.	AWSIATEUILAccess policy	36
Figure 40.	Amazon S3 Full Access policy	36
Figure 50	Attached policies	37
Figure 51	Architecture of AWS InT	<u>4</u> 1
Figure 52	Message flows of opening door	41
Figure 52	Open dooring on mobile app	42
Figure 54	Shadow state when door is open	42
Figure 55	Message flows of closing door	בי 42
Figure 56	Closing door on mobile app	44
Figure 57	Shadow state when door is closed	44
Figure 58	AWS IoT using firmware images for AT commands and host MCU	46
Figure 59	Hardware configuration	47
Figure 60	Default UART hardware connection	48

Figure 61. Example of UART1 connection	48
Figure 62. Hardware connection for waking up DA16200/DA16600	
Figure 63. Default pin configuration for waking up host MCU	
Figure 64. Another pin configuration for waking up host MCU	
Figure 65. Factory reset button on DA16200 EVB	51
Figure 66. Factory reset button on DA16600 EVB	51
Figure 67. e ² studio project file	
Figure 68. FSP configuration	
Figure 69. Thing name in MCU source code	57 57
Figure 70. Build project	
Figure 71. Debug configurations	
Figure 72. Set debug configurations	00 61
Figure 73. Opened status on application	01
Figure 74. Opened status on AWS IOT COnsole	
Figure 75. Closed status on AW/S IoT console	02 62
Figure 77, OTA undate	
Figure 78. Create bucket for OTA undate	04 64
Figure 79. Bucket configuration – general and object ownership	04 65
Figure 80. Bucket configuration – public access and versioning	60 66
Figure 81. Bucket configuration – bucket key	66
Figure 82 Created buckets for OTA	
Figure 83. Edit bucket for public access	67 67
Figure 84. Public access settings for bucket	68
Figure 85. Confirm settings	
Figure 86. Settings updated	
Figure 87. Public access for everyone	
Figure 88. Bucket policy editor	
Figure 89. Upload files	73
Figure 90. Ready to upload	73
Figure 91. URL of source	74
Figure 92. Uploaded files	74
Figure 93. Completed setup for OTA update	74
Figure 94. Create job	75
Figure 95. Create custom job	75
Figure 96. Enter job name	76
Figure 97. Select thing for OTA update	76
Figure 98. Select JSON for OTA update	77
Figure 99. Job run type	77
Figure 100. Job being created	78
Figure 101. Successfully created job	78
Figure 102. Successful job for OTA update in mobile app	80
Figure 103. Execute OTA update in Android app	80
Figure 104. Create private bucket	82
Figure 105. Upload files to bucket	
Figure 106. Device shadow in thing	
Figure 107. Create private policy	
Figure 108. Create private role	
Figure 109. I rusted entity	85
Figure 110. Adding policy to role	85
Figure 111. Create private user	
Figure 112. Permission setting	86 50
Figure 113. Selecting policy for user	
Figure 114. Review and Create user	
Figure 116. Retrieve access key	.00 مە
Figure 117 Create role alias	00 00
Figure 118 Role alias properties	 ספ
Figure 119 Provisioning flow	ອງ ເດງ



Figure 120. Provisioning from mobile app	
Figure 121. Running AWS IoT application from mobile app	
Figure 122. Setting mode	
Figure 123. Provisioning mode	
Figure 124. Communication mode	
Figure 125. Communication between MCU and phone	

Tables

Table 1. Pin connection	47
Table 2. Default configuration for UART1 or UART2	47
Table 3. UART1 pin configuration	48
Table 4. GPIO pin configuration	50
Table 5. Bucket policy in JSON format	71
Table 6. Configuration of topics	97
Table 7. Basic set of MCU to DA16200/DA16600	
Table 8. TLS from MCU to DA16200/DA16600	
Table 9. PIN MUX from MCU to DA16200/DA16600	
Table 10. Configuration data from MCU to DA16200/DA16600	
Table 11. Command of MCU to DA16200/DA16600	
Table 12. Command of DA16200/DA16600 to MCU	
Table 13. Status from DA16200/DA16600 to MCU	



1. Terms and Definitions

AP	Access Point
API	Application Programming Interface
AWS	Amazon Web Services
DEVKT	Development Kits
DPM	Dynamic Power Management
DTIM	Delivery Traffic Indication Map
IDE	Integrated Development Environment
loT	Internet of Things
LE	Low Energy
MCU	Micro-Controller Unit
ΟΤΑ	Over the Air
SDK	Software Development Kit
TIM	Traffic Indication Map

2. References

- [1] DA16200MOD, Datasheet, Renesas Electronics.
 (DA16200MOD Ultra-Low Power Wi-Fi Modules for Battery Powered IoT Devices | Renesas)
- [2] DA16600MOD, Datasheet, Renesas Electronics.
 (DA16600MOD Ultra-Low Power Wi-Fi + Bluetooth[®] Low Energy Combo Modules for Battery Powered IoT Devices | Renesas)
- [3] UM-WI-056, DA16200 DA16600 FreeRTOS Getting Started Guide, User Manual, Renesas Electronics. (UM-WI-056 DA16200 DA16600 FreeRTOS Getting Started Guide (renesas.com))
- [4] UM-WI-042, DA16200 DA16600 Provisioning Mobile App for Android/iOS, User Manual, Renesas Electronics.

(DA16200 DA16600 Provisioning Mobile App (renesas.com))

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



3. Overview

The DA16200 Wi-Fi Development Kit (DEVKIT) provides a quick and easy method to start developing battery powered applications and products using ultra-low power Wi-Fi. The ultra-low power DA16200 chipset is the world's lowest power Wi-Fi device specifically designed to meet the requirements for power sensitive wireless applications.

The DA16600MOD-DEVKT provides a host board based on the DA16600 Wi-Fi + Bluetooth® Low Energy (LE) module supporting a USB connection to a personal computer for evaluation and development of low power Wi-Fi and Bluetooth® LE applications.

3.1 Benefits

- Ultra-low power Wi-Fi technology
- More than 1-year battery life for most applications
- Industry leading wireless range
- Fully integrated Wi-Fi system on chip (SoC)
- Comprehensive security capabilities
- Easy to use development tools means shorter time to market.

3.2 Features

- Highly integrated ultra-low power DA16200 Wi-Fi system module
- Best Radio Frequency performance
- SoC runs full networking OS and TCP/IP stack
- Built-in 4-channel auxiliary ADC for sensor interfaces
- Built-in hardware crypto engines for advanced security features
- Complete software stack
- eMMC/SD expanded memory.

3.3 Applications

DA16200MOD/DA16600MOD is a full offload SoC for IoT applications, such as:

- Security systems
- Door locks
- Thermostats
- Garage door openers
- Blinds
- Lighting control
- Sprinkler systems
- Video camera security systems
- Smart appliances
- Video doorbell.



4. Hardware Description

4.1 Datasheet

4.1.1 DA16200MOD

The DA16200MOD is a fully integrated Wi-Fi® module with ultra-low power consumption, best RF performance, and easy development environment. For more information, see Ref. [1].

4.1.2 DA16600MOD

The DA16600 modules provide a convenient way to add both low power Wi-Fi and low power Bluetooth[®] LE functionality to your device. For more information, see Ref. [2].

NOTE

To purchase the development kits, click the following links:

- DA16200MOD-DEVKT Renesas Electronics Corporation | Development Boards, Kits, Programmers | DigiKey
- DA16600MOD-DEVKT Renesas Electronics Corporation | Development Boards, Kits, Programmers | DigiKey

5. Set Up Development Environment

You can develop Wi-Fi applications for the DA16200 using the DA16200 FreeRTOS Software Development Kit

(SDK) and the Renesas e²studio IDE on either a Windows 10 or Linux based development system.

To set up SDK, complete the following steps:

- Install and configure the e²studio IDE
- Import the DA16200 SDK into the e²studio, and build an application
- Setting up the evaluation board
- Download and test the application
- Use J-Link debugger to debug the application.

For more information, see DA16200 DA16600 FreeRTOS Getting Started Guide, Ref. [3].



6. AWS IoT

The DA16200MOD/DA16600MOD is a full offload SoC for IoT applications such as security systems, door locks, and smart applications. This section provides procedures on how to configure AWS IoT for communicating with DA16200/DA16600 IoT devices.

6.1 Configure AWS IoT

To connect a device to the AWS IoT server, complete the following steps:

- 1. Sign up AWS account and permissions.
- 2. Connect devices to AWS IoT.
- 3. Configure Amazon Cognito user pools and identity pools.
- 4. Set up Amazon IAM.
- 5. Create S3 bucket.

6.1.1 Sign Up for AWS Account

To create an AWS account and grant permissions:

- 1. Go to AWS website and create a free account (https://portal.aws.amazon.com/).
- 2. Create an administrative user for performing daily administrative tasks.
- 3. Open the AWS IoT console to get started with AWS IoT.



Figure 1. Sign up for AWS account

NOTE

If you do not have an AWS account, Renesas Electronics can provide a Thing name that has already been created for testing.

6.1.2 Connect Devices to AWS IoT

You can configure and manage the thing objects, certificates, rules, jobs, policies, and other elements of IoT solutions through AWS IoT console. Prior to sending data to and receiving data from AWS IoT server, you need to register a device first.

6.1.2.1 Register a Device in Thing Registry

In the Thing Registry, the devices connected to the AWS IoT server are represented by Things. The Thing Registry allows keeping records of all devices that are connected to an AWS IoT account.

To register a device in the Thing Registry:

- 1. On the AWS IoT console, on the navigation pane, expand Registry.
- 2. Expand **All devices** and click **Things** > **Create things**.

AWS IoT ×	AWS IoT > Manage > Things	
Monitor	Things (0) Info C Advanced search	Run aggregations Edit Delete Create things
Connect Connect one device Connect many devices	An IoT thing is a representation and rec needs a thing record in order to work w	rd of your physical device in the cloud. A physical device the AWS toT. roup, billing, or searchable attribute.
	Name	Thing type
est MQTT test client Device Location <u>New</u>		No things No things to display in this Region Create things
Manage All devices Things Thing groups		

Figure 2. Register things

3. Select Create single thing.

ource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a think
the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.
er of things to create
reate single thing
eate a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS T.
reate many things
reate a task that creates multiple thing resources to register devices and provision the resources those devices require to printed to AWS IoT.

Figure 3. Create single thing

4. To add the device to the Thing Registry, in the **Thing name** field, enter a device name, for example, "MyTestDoorLock", and under **Device Shadow**, select **Unnamed shadow** (classic) and click **Next**.

AWS IOT > Manage > Things > Crea	ate things > Create single thing
Step 1 Specify thing properties Step 2 - optional Configure device certificate	Specify thing properties Info A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.
Step 3 - <i>optional</i> Attach policies to certificate	Thing properties Info
	Thing name MyTestDoorLock Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.
	Additional configurations You can use these configurations to add detail that can help you to organize, manage, and search your things.
	► Thing type - optional
	Searchable thing attributes - optional
	Thing groups - optional Billing group - optional
	Packages and versions - optional
	Device Shadow Info
	Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.
	 No shadow Named shadow Create multiple shadows with different names to manage access to properties, and logically group your devices properties. Unnamed shadow (classic) K tring can have only one unmanted shadow.
	Edit shadow statement - optional
	Cancel

Figure 4. Thing name



5. Select Skip creating a certificate at this time and click Create thing.

Step 1 Specify thing properties Step 2 - optional Configure device certificate	Configure device certificate – <i>optional</i> Info A device requires a certificate to connect to AWS IoT. You can choose how to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.
	Device certificate
	O Auto-generate a new certificate (recommended) Generate a certificate, public key, and private key using AWS IoT's certificate authority.
	Use my certificate Use a certificate signed by your own certificate authority.
	O Upload CSR Register your CA and use your own certificates on one or many devices.
	• Skip creating a certificate at this time You can create a certificate for this thing and attach a policy to the certificate at a later time.
	Cancel Previous Create thing

Figure 5. Thing without certificate

Now you have the thing created to perform the test and it is named **MyTestDoorLock**.

6. In the **Things** list, click the created thing.

⊘ You successfully created thing MyTestDoolock.	View thing	×
AWS IoT > Manage > Things		
Things (1) Info C Advanced search Run aggregations Edit Delete Create thin	igs	
An loT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS loT. Q Filter things by: name, type, group, billing, or searchable attribute.	1 > @	
Name MyTestDoolock		
		,

Figure 6. Created thing

Then, the thing details appear.



7. For the shadow function of the thing, select the **Device Shadows** and click **Classic shadow**.

AWS IoT \times	AWS IoT > Manage > Things > MyTestDoorLock
Monitor	MyTestDoorLock Info Create secure tunnel Edit Delete
Connect	Thing details
Connect one device Connect many devices	Name Type MyTestDoorLock -
Test MQTT test client	ARN Billing group arn:aws:iot:ap-northeast-2:432073875051:thing/MyTestDoorLock
Manage	Attributes Certificates Thing groups Device Shadows Activity Packages and versions Jobs Al
 All devices Things Thing groups Thing types 	Device Shadows (1) Info Create Shadows Device Shadows allow connected devices to sync their state with AWS. You can also get, update or delete the state information about this thing's Device Shadows by using HTTPS and MQTT topics.
Fleet metrics Greengrass devices 	Q. Filter Device Shadows
Software packages New	Name ▲ MQTT topic prefix Fleet indexing status Last updated date
Remote actionsMessage routing	Classic Sh 🗇 \$aws/things/MyTestDoorLock/shadow 🥥 Selected October 19, 2023, 16:36:

Figure 7. Classic shadow



You can select named shadows to add to your fleet index	ding settings. <u>Learn more</u>
lassic Shadow	C Del
Device Shadow details	
ARN arn:aws:iot:ap-northeast- 2:432073875051:thing/MyTestDoorLock MQTT topic prefix \$ \$aws/things/MyTestDoorLock/shadow Device Shadow URL https://a1kzdt4nun8bnh-ats.iot.ap-northeast- 2.amazonaws.com/things/MyTestDoorLock/shadow Device Shadow document MQTT topics	Last updated October 19, 2023, 16:36:37 (UTC+09:00) Version 1 Prefix for Fleet indexing query shadow.name.Classic Shadow. Fleet indexing status Selected
<pre>Device Shadow document Info The Device Shadow document contains the reported, desired, and delta va device can sync its state while it's connected to AWS IoT. Device Shadow state { "state": { "desired": { "welcome": "aws-iot" }, "reported": { "welcome": "aws-iot" } } }</pre>	Edit

Figure 8. Device shadow document

For more information on device shadows for AWS IoT, visit AWS IoT Device Shadow service (https://docs.aws.amazon.com/iot/latest/developerguide/iot-device-shadows.html).



6.1.2.2 Create and Activate Device Certificate

The communication between the device and the AWS IoT is protected by X.509 certificates. You can let the AWS IoT generates a certificate or you can use your own X.509 certificate. This section shows that AWS IoT generates the X.509 certificate.

You should activate the certificates before use. To create and activate a device certificate:

1. On the navigation pane, expand **Security** and click **Certificates**, and then click **Add certificate** > **Create certificate**.

AWS IoT \times	AWS IoT > Security > Certificates	
Monitor	Certificates Info X.509 certificates authenticate device and client connections.	Certificates must be registered with AWS IoT and activated before a
Connect Connect one device	Certificates Certificates you've transferred	
	Certificates (0)	C Actions V Add certificate
Test	Q Find certificates	Create certificate
Device Location New	Certificate ID	▼ Status ▼ Created
Manage	No You don't have any	o certificates
All devices		
 Greengrass devices 	Crea	
LPWAN devices		
Software packages New		
Remote actions		
Message routing		
Retained messages		
Security		
Intro		
Certificates		

Figure 9. Create certificates

2. Select Auto-generate new certificate (recommended) > Activate and click Create.

T with	ate certificate Info ates authenticate devices and clients so that they can co hout authentication and an appropriate policy.	nnect to AWS IoT. Your device won't be able to connect to AW
•	Auto-generate new certificate (recommended) Generate a new certificate, public key, and private key using AWS IoT's certificate authority and register it with AWS IoT.	Create certificate with certificate signing request (CSR) Upload your own certificate signing request (CSR) file to create and register a certificate that's based on a private key you own.
Cer Assign status	tificate status In the initial state of the new certificate. The certificate must be a Is later in the certificate's detail page.	ctive before it can be used to connect to AWS loT. You can change its
 Ir A A 	nactive device won't be able to connect to AWS using this certificate unt Active A device will be able to connect to AWS using this certificate imme	il it's activated. idiately after you create it.
		Cancel

Figure 10. Create certificates (continued)

- 3. There are three required certificates to download.
 - On the **Certificate Created** page, download the device certificate, private key, and root CA certificates for AWS IoT, and then save the downloads to your computer, click **Download**.

NOTE

You must save the certificate files before leaving this page. If you leave the page without saving, you no longer have access to the certificate files. Renesas recommends that Device certificate, Private key file, and Root CA should be downloaded in sequential order.

Download certificates and keys	×
Download certificates and keys Download and install the certificate and key files to your device so that it can connect se IoT. You can download the certificate now, or later, but the key files can only be downloa	curely to AWS ided now.
Device certificate 9e43e7e6594te.pem.crt	ownload
Key files	
The key files are unique to this certificate and can't be downloaded after you leave this p Download them now and save them in a secure place.	age.
▲ This is the only time you can download the key files for this certificat	ie.
Public key file	ownload
9e43e7e659461f0b979c2179c9432a-public.pem.key	
Private key file	bound
9e43e7e659461f0b979c217c9432a-private.pem.key	owntoau
Root CA certificates	
Download the root CA certificate file that corresponds to the type of data endpoint and you're using. You can also download the root CA certificates later.	cipher suite
Amazon trust services endpoint	ownload
RSA 2048 bit key: Amazon Root CA 1	owntoau
Amazon trust services endpoint	heolawo
ECC 256 bit key: Amazon Root CA 3	ownoad
If you don't see the root CA certificate that you need here, AWS IoT support root CA certificates. These root CA certificates and others are available from developer guides.	s additional 1 our
C	
	Continue

Figure 11. Download certificates and keys

For Root CA, visit the AWS Docs site (https://docs.aws.amazon.com/iot/latest/developerguide/serverauthentication.html#server-authentication-certs). Root CA certificates are subjected to expiration and/or revocation.

The certificate status should be Active in the list of certificates.

⊘ You successfully created certificate 9e43e7e659461f0b979c2170d2953377e1a242a61db5ffff71576ac7a9c9432a.	View certificate	×
AWS IOT > Security > Certificates		
Certificates Info X.509 certificates authenticate device and client connections. Certificates must be registered with AWS IoT and activated before a device or client can comm Certificates Certificates you've transferred	unicate with AWS IoT.	
Certificates (58)	Add certificate 🔻	
Q Find certificates	< 1 2 > ©	
Certificate ID V Status V Created	•	
□ <u>9e43e7e659461f0b979c2170d2953377e1a242a61db5ffff71576ac7a9c9432a</u> ⊘ Active October 19, 2023, 1	17:12:54 (UTC+09:00)	

Figure 12. Activate certificate

6.1.2.3 Create Policy

The X.509 certificates are used to authenticate the device with the AWS IoT. The AWS IoT policies are used to authorize the device for AWS IoT operations, such as subscribing or publishing to MQTT topics. The device displays its certificate only while connecting to the AWS IoT.

To allow the device for AWS IoT operations, you should create an AWS IoT policy and attach that policy to the device certificate.

To create an AWS IoT policy:

1. On the navigation pane, expand **Security** and click **Policies** > **Create policy**.

AWS IOT ×	AWS IoT > Security > Policies
Monitor	AWS IoT policies (0) Info
Connect Connect one device	Q. Find policies < 1 > (2)
Test MQTT test client Device Location <u>New</u>	No policies You don't have any AWS IoT policies in ap-southeast-2. Create
Manage All devices Greengrass devices UPWAN devices Software packages New	
Message routing Retained messages Security Intro Certificates	
Policies	

Figure 13. Create policy

- 2. On the **Create policy** page:
 - a. In the **Policy name** field, under **Policy properties**, enter a name for the policy (for example, MyTestPolicy). Renesas strongly recommends not using personally identifiable information in policy names.



Figure 14. Add policy name

b. Under Policy document, select JSON, and then copy and paste the following JSON statement:

After entering the required information, click Create.

C. NOTE

The examples in this document are intended only for development environments. All devices in your production fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies may vary depending on use cases. Identify the permission policies that best meet the business and security requirements. For more information, see Example Policies and Security Best practices in AWS IoT.

IoT policy contains on					
ources.	or more policy statements. E	Each policy statement of	contains actions, resour	ces, and an effect tha	it grants or denies the action
document					
.▼ {					
"Version":	2012-10-17",				
v Statement V {	E L				
Effect	': "Allow",				
"Action	': "iot:*",				
<pre>kesour</pre>	.e : ••				
•					
"Effect	': "Allow",				
"Action	1 53:*", p": "*"				
})		
: <u>1</u>					
3					
				le	
				li	
				li	

Figure 15. Enter JSON policy statement

3. To view the created policies, expand Security and click Policies.

⊘ Successfully created policy MyTestPolicy.	View policy	×
AWS IoT > Security > Policies		
AWS IOT policies (1) Info C Delete AWS IOT policies allow you to control access to the AWS IoT Core data plane operations. AWS IoT policies are separa IAM policies. AWS IoT policies apply only to AWS IoT data plane operations.	Create policy ate and different from	
Q Find policies	< 1 > 🔘	
Policy name	•	
<u>MyTestPolicy</u>		

Figure 16. Created policy

4. Click the policy to view the details. Figure 17 shows an example of the selected policy content.

lyTestPolicy Info		Edit	t active version Delete
Details			
Policy ARN arn:aws:iot:ap-south east-2:649620604383:p olicy/MyTestPolicy	Active version 1	Created October 20, 2023, 11:48:39 (UTC+09:00)	Last updated October 20, 2023, 11:48:39 (UTC+09:00)
Versions Targets N	loncompliance Tags		
Active version: 1 Info			Builder JSON
Active version: 1 Info Policy effect	Policy action	Policy re	Builder JSON
Active version: 1 Info Policy effect Allow	Policy action	Policy re *	Builder JSON
Active version: 1 Info Policy effect Allow Allow	Policy action iot:* s3:*	Policy re * *	Builder JSON
Active version: 1 Info Policy effect Allow Allow Allow All versions (1) Info The active and previous versions of t with 5 versions, you must first delete	Policy action iot:* s3:*	Policy re	Builder JSON esource version View JSON e than 5 versions. To update a policy
Active version: 1 Info Policy effect Allow Allow Allow The active and previous versions of t with 5 versions, you must first deleted Version number	Policy action iot:* s3:* C Delete this policy. Only one version can be e one. Status	Policy re	Builder JSON esource

Figure 17. Check created policy

6.1.2.4 Attach Certificate to Thing and Policy

After an AWS IoT policy is created, you must attach that policy to the device certificate. The attachment of an AWS IoT policy to a certificate gives the device the permissions that are specified in the policy.

To attach the AWS IoT policy to a device certificate:

1. Go to the certificate you created, select **Policies** and click **Attach policies**.

SIOT > Security > Certificates > 011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227			
011b436fee088be5999c79759 de7184557c3227c Info Actions 💌	ae308ef71c97a478eba159b3		
Details			
Certificate ID 011b436fee088be5999c79759ae308ef71c97a478eba1 59b3de7184557c3227c Certificate ARN am:aws:iot:ap-southeast-2:649620604383:cert/01 1b436fee088be5999c79759ae308ef71c97a478eba159 b3de7184557c3227c Subject CN=AWS IoT Certificate Issuer OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US	Status		
Policies Things Noncompliance	Policies Things Noncompliance		
Policies (0) Info C Detach policies Attach policies Attach policies Attach policies allow you to control access to the AWS IoT Core data plane operations.			
Name	v		
No po You don't have any policies	blicies attached to this certificate.		

Figure 18. Policies

2. Select the created policy and click Attach policies.

Attach policies to the certificate 011b436fee088be5999c79759ae308ef71c97a478eba159b3de 7184557c3227c.	X
Policies Choose policies to attach to this certificate. The certificate can have up to 10 policies attached to it. <i>Choose AWS IoT policy</i>	9
Cancel Attach policies	

Figure 19. Attach policy

NOTE

A device should have a certificate, private key, and root CA certificate to authenticate with the AWS IoT. Renesas recommends that you attach the device certificate to the thing that represents the device in AWS IoT. This allows you to create AWS IoT policies that grant permissions based on certificates attached to things.

3. Go to the certificate created by you, select **Things** and click **Attach to things**.

AWS IoT > Certificates > 011b436fee088be5999c79759ae308ef71c97a478eba159b3de7184557c3227c			
011b436fee088be5999c79759ae308ef71c97a478eba159b3 de7184557c3227c Info			
Details			
Certificate ID St. 011b436fee088be5999c79759ae308ef71c97a478eba1 St. 59b3de7184557c3227c Gr. Certificate ARN Gr. Gram:aws:iot:ap-southeast-2:649620604383:cert/01 1b436fee088be5999c79759ae308ef71c97a478eba159 b3de7184557c3227c Subject Subject Ex. CN=AWS IoT Certificate Ja Issuer OU-Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US Policies Things (0) Info C An AWS IoT thing is a representation and record of your physical device in the device using the certificate to the thing resource.	atus) Active eated ctober 20, 2023, 10:53:30 (UTC+09:00) did tcober 20, 2023, 10:51:30 (UTC+09:00) pires nuary 01, 2050, 08:59:59 (UTC+09:00) Detach from things Attach to things he cloud. Attaching a certificate to an AWS Io1 thing relates		
Name 🔻			
No things This certificate is not attached to any things.			

Figure 20. Attach things to certificate

4. Select the box of the thing that was created and click **Attach to thing**.

Tincate to.	
cer	certificate to. ▼ C

Figure 21. Attach to thing

6.1.2.5 Store Events in S3 Bucket

To store log files for the Door lock:

NOTE

To create Amazon S3 bucket, see Section 6.1.5.

1. Select AWS console > AWS IoT Core, expand Message routing and click Rules > Create rule.

AWS IoT ×	AWS IOT > Message routing > Rules
Monitor	Rules (0) Info C Activate Deactivate Edit Delete Create rule Rules allow your things to interact with other services. Rules are analyzed and perform specific actions based on messages puonsned by Second
Connect Connect one device	your devices. Q. Find rules < 1 > @
Connect many devices	Name ▲ Status ▼ Rule topic ▼ Cre
Test MQTT test client Device Location New	No rules You don't have any rules in ap-southeast-2.
Manage All devices Greengrass devices	
 LPWAN devices Software packages New Remote actions 	
Message routing Rules Destinations	

Figure 22. Create rule

2. Enter a rule name and click Next.

ecify rule properties	Specify rule properties Info	
	A rule resource contains a list of actions based on the MQTT topic stream.	
ep 2 onfigure SQL statement	Rule properties	
ep 3 tach rule actions	Rule name	
ep 4 iview and create	MyTestAct Enter an alphanumeric string that can also contain underscore (_) characters, but no spaces. Rule description - optional. Enter a description to consider additional details about the sub to others.	
	A description of your new rule	
	Tags - optional No tags are associated with the resource.	
	Add tag You can add 1 more tag.	

Figure 23. Specify rule name

3. Copy and paste the following SQL statement in the SQL statement box and click Next.

SELECT * FROM '\$aws/things/Yourthingname/shadow/update'

WHERE state.reported.doorStateChange > 0 OR state.reported.temperature > 70 OR

state.reported.doorBell > 0

Note that the thing name is now **MyTestDoorLock**.

AWS IOT > Message routing > Ru	Jes > Create rule
Step 1 Specify rule properties	Configure SQL statement Info Add a simplified SQL syntax to filter messages received on an MQTT topic and push the data elsewhere.
Step 2 Configure SQL statement	SQL statement
Step 3	
Attach rule actions	SQL version The version of the SQL rules engine to use when evaluating the rule.
Step 4	2016-03-23
Review and create	SQL statement Enter a SQL statement using the following: SELECT <attribute> FROM <topic filter=""> WHERE <condition>. For example: SELECT</condition></topic></attribute>
	1 SELECT * FROM 'Saws/things/MyTestDoorLock/shadow/update' 2 WHERE state.reported.doorStateChange > 0 OR state.reporte .temperature > 70 OR state.reported.doorBell > 0
	SQL Line 2, Column 106

Figure 24. Configure SQL statement

4. Under Rule actions, in the Action 1 list, select S3 bucket.

tep 1 ipecify rule properties	Attach rule actions Info An action routes data to a specific AWS service.
tep 2 Configure SOL statement	SQL statement Back
Step 3 Attach rule actions Step 4 Review and create	<pre>SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' WHERE state.reported.doorStateChange > 0 OR state.reported. temperature > 70 OR state.reported.doorBell > 0</pre>
	Rule actions Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.
	Action 1
	Choose an action Remove

Figure 25. Attach rule actions

5. Click Browse S3 and in the Key field, enter \${timestamp()}. Then, click Create new role.

pecify rule properties	Attach rule actions into An action routes data to a specific AWS service.
tep 2 onfigure SQL statement	SQL statement Back
tep 3 Attach rule actions	<pre>SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' WHERE state.reported.doorStateChange > 0 OR state.reported.temperature > 70 OR state.reported.doorBell > (</pre>
tep 4 eview and create	Rule actions Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive. Bie sching them in a database. Inveking cloud functions, or sending notifications. You can add up to 10 actions.
	Action 1
	S3 bucket Remove Store a message in an Amazon S3 bucket Remove
	Bucket name Info S3 URL
	C s3:// X View Browse 53
	s3key
	Canned ACL The Amazon S3 canned ACL that controls access to the object identified by the object key. private
	IAM role Chose a role to grant AWS IoT access to your endpoint.
	Choose an IAM role View C Create new role W5 IoT will automatically create a policy with a prefix of "avs-lot-rule" under your IAM role select.
	Add rule action
	Error action - optional You can optionally set an action that will be executed when something goes wrong with processing your rule. If two rule actions in the same rule fail, the error action receives one message that contains both errors.
	Add error action

Figure 26. Attach rule actions (continued)

6. In the **Role name** field, enter an IAM role name and click **Create**.

Create role	×
Role name Log-MyAct-IAM-Role	
Enter a unique role name that contains alphanumeric name can't contain any spaces.	characters, hyphens, and underscores. A role
	Cancel

Figure 27. Create IAM role to save log files

7. Review the entered information and click **Create**.

itep 1 Specify rule properties	Review and create Info		
tep 2	Step 1: Rule properties	[Edit
Configure SQL statement	Rule properties		
tep 4	Name MyTestAct		
arep 4 Review and create	Description -		
	Step 2: SQL statement	[Edit
	SQL statement		
	SQL version 2016-03-23		
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' W	HERE state.reported.doorStateChange	> 0
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' i OR state.reported.temperature > 70 OR state.reported.doord Step 3: Rule actions	WHERE state.reported.doorStateChange Bell > 0	e > 0 Edit
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' I OR state.reported.temperature > 70 OR state.reported.doorl Step 3: Rule actions Actions	#HERE state.reported.doorStateChange Bell > θ [e > 0 Edit
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' I OR state.reported.temperature > 70 OR state.reported.doorl Step 3: Rule actions Actions S3 bucket Store a message in an Amazon S3 bucket	WHERE state.reported.doorStateChange Bell > 0	e > 0 Edit
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' I OR state.reported.temperature > 70 OR state.reported.doorl Step 3: Rule actions Actions S3 bucket Store a message in an Amazon S3 bucket Bucket name Key mytestdoorlock-log \$(timestamp()) IAM role amawsiam:549520504383:role/s ervice-role/Log-MyAct-IAM-Role [2]	dHERE state.reported.doorStateChange Bell > θ Canned ACL private	: > 0 Edit
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' I OR state.reported.temperature > 70 OR state.reported.doorl Step 3: Rule actions Actions S3 bucket Store a message in an Amazon S3 bucket Bucket name Key mytestdoorlock-log \$(timestamp()) IAM role amaxwsiam:649620604583:role/s ervice-role/Log-MyAct-IAM-Role [2]	dHERE state.reported.doorStateChange Bell > θ Canned ACL private	Edit
	SQL query SELECT * FROM '\$aws/things/MyTestDoorLock/shadow/update' I OR state.reported.temperature > 70 OR state.reported.doorl Step 3: Rule actions Actions S3 bucket Store a message in an Amazon S3 bucket Bucket name Key mytestdoorlock-log \$(timestamp0)) IAM role arm:awsiam:549620504383:role/s ervice-role/Log-MyAct-IAM-Role [] Error action No error action	dHERE state.reported.doorStateChange Bell > θ Canned ACL private	Edit

Figure 28. Review rules



8. The created rules should appear in the list of policies.

⊘ Successfully created rule MyTestAct.	View rule	×
AWS IoT > Message routing > Rules		
Rules (1) Info C Activate Deactivate Edit Delete C Rules allow your things to interact with other services. Rules are analyzed and perform specific actions based on messages published by your device C	reate rule	
Q, Find rules □ Name ▲ Status ▼ Rule topic ▼ Cree	1 > (2) eated date	
MyTestAct ⊘ Active \$aws/things/Yourthingname/ Oct	tober 20, 2023, *	1

- Figure 29. Created rule
- 9. Go to the IAM console and select **Roles** and review the created roles.

Identity and Access ×	IAM > Roles	
Q Search IAM	Roles (3) Info An IAM role is an identity you can create that has specific permissions with credentials rentities that you trust.	that are valid for short durations. Roles can be assumed by
Dashboard	Q Search	< 1 > ©
 Access management 		
User groups		 Trusted entities
Users	AWSServiceRoleForSupport	AWS Service: support (Service-Lin
Roles	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Serv
Policies		
Identity providers	Log-MyAct-IAM-Role	AWS Service: iot
Account settings		

Figure 30. Created role

10. Choose the created role name and click **Attach policies**.

Permissions	Trust relationships	Tags Access Advisor Re	voke sessions	
Permissions You can attach up	s policies (1) Info to 10 managed policies.	C Simulate Z	Remove Add permissions	
Q Search		Filter by Type All types	Create inline policy	0
Policy	name 🛛		▲ Type	▽
. •	aws-iot-rule-MyTestAct-act	tion-1-role-Log-MyAct-IAM-Role	Customer managed	

Figure 31. Attach policy to role

11. Search for the policy name of **AWSIoTFullAccess** and click **Add permissions**. Do the same thing for **AmazonS3FullAccess**.

IAM > Roles > Log-MyAct-IAM-Role	> Add permissions	
Attach policy to Log-MyAct-IAM	I-Role	
Current permissions policie	es (1)	
Other permissions policies (1,	/882)	C
	Filter by Type	
		· match
Policy name	▲ Туре	▼ Description
✓	AWS managed	This policy gives full access to the AWS I
		Cancel Add permissions

Figure 32. AWSIoTFullAccess policy

NOTE

AWSIoTFullAccess and AmazonS3FullAccess policies are not recommended for production. When the policies are added, the execution roles should look like Figure 33.

Policies have been successfully attached to role.		×
IAM > Roles > Log-MyAct-IAM-Role Log-MyAct-IAM-Role Info	Delete]
Summary	Edit	
Creation date October 20, 2023, 15:40 (UTC+09:00)	ARN Arn:aws:iam::649620604383:role/service-role/Log- MyAct-IAM-Role	
Last activity -	Maximum session duration 1 hour	
Permissions Trust relationships Tags Acce	ss Advisor Revoke sessions	
Permissions policies (3) Info You can attach up to 10 managed policies.	Simulate 🖾 Remove Add permissions 🔻	
Q Search Filter by All type	Type is	
Policy name 🔀	▲ Type	
AmazonS3FullAccess	AWS managed 1	
	Customer managed 1	
AWSIoTFullAccess	AWS managed 1	

Figure 33. Attached policies

6.1.3 Configure Amazon Cognito

Amazon Cognito provides authentication, authorization, and user management for web and mobile apps. You can sign in directly with a username and password or through a third party such as Facebook, Amazon, or Google.

The two main components of Amazon Cognito are **user pools** and **identity pools**. User pools are directory of users that provide sign-up and sign-in options for app users. Identity pools provide AWS credentials to grant users access to other AWS services. Identity pools and user pools can be used separately or together. For more information, visit AWS Docs site (https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html).

6.1.3.1 Create User Pools

1. Go to the Amazon Cognito console and click **Create user pool**.



Figure 34. Create user pool

2. On the **Configure sign-in experience** page, select the **Email** checkbox, and click **Next**.



Figure 35. Configure sign-in options

3. Select Cognito defaults as password policy mode. Then, select No MFA and Email only, and click Next.



Figure 36. Configure security requirements

4. Configure sign-up as shown in Figure 37.



Figure 37. Configure sign-up experience

5. Select Send email with Cognito.



Figure 38. Configure message delivery



6. On the Integrate your app page, enter required items as shown in Figure 39 and click Next.

Amazon Cognito > User pools >	Lreate user pool
Step 1 Configure sign-in experience	Integrate your app Info
Step 2	Set up app integration for your user pool with Cognito's built-in authentication and authorization flows.
Configure security requirements	User pool name Create a friendly name for your user pool.
Step 3	User pool name
COLLI GALLE A GILLARE SORELLE LESS	MyUserPool_DoorLock
Step 4 Configure message delivery	User pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , . @ -
Step 5	Your user pool name can't be changed once this user pool is created.
ntegrate your app	
Step 6 Review and create	Hosted authentication pages Choose whether to use Cognito's Hosted UI and OAuth 2.0 server for user sign-up and sign-in flows.
	Use the Cognito Hosted UI Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.
	Initial app client Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.
	App type Info Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is
	Public client A native, browser or mobile- device app. Cognito API requests are made from user systems that secret. Cognito API requests are made from a central server. Cognito API requests are made
	App client name Info
	MyAppClient App client names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: +====================================
	Client secret Info
	Choose whether your app client will have a client secret. Client secrets are used by the server-side component of an app to authorize API Generate a client secret Don't generate a client secret
	You cannot change or remove a client secret after you allow Amazon Cognito to generate it for your app client.
	 Advanced app client settings We have populated suggested authentication flows, OAuth 2.0 grant types, and OIDC scopes based on the selections you made earlier.
	Attribute read and write permissions Info Choose the standard and custom attributes this app can read and write. Required attributes are locked as writable. We recommend that you set immutable custom attributes as writable to allow the app client to set initial values during sign-up.
	Tags (0) - optional You can add tags to your user pool for cost management and access control.
	No tags associated with the resource. Add new tag
	You can add up to 50 tags.
	Cancel Previous Next

Figure 39. Integrate app client

7. On the **Review and create** page, review the entered information, and click **Create user pool**. Then, the created user pool should appear in the list.

ser pool "MyUserPool_DoorLock" has been created successfully.	View details
mazon Cognito > User pools	
Integrate Amazon Cognito with Amazon Verified Permissions Amazon Verified Permissions is a fine-grained authorization service for role- and attribute-based access control in apps that authenticate with Amazo From a user's identity or access token, Verified Permissions compares their attributes to access rules for your resource. The response is an authorizati allow or dery. Verified Permissions can consolidate authorization for all of your apps and resources into a central policy store. Your policies are writte Cedat P, an open-source language built for access control. Learn more P Go to Amazon Verified Permissions	X on Cognito. on decision: en in
User pools (1) Info Cre View and configure your user pools. User pools are directories of federated and local user profiles. They provide authentication options for your users.	ate user pool
Q Search user pools by name or ID	(1) ()
User pool name 🔺 🛛 User pool ID 🛛 🔻 Created time 🔍 Last updated time	⊽
MyUserPool DoorLock ap-northeast-2_PWpT7eGQ3 1 second ago 1 second ago	

Figure 40. Created user pool

6.1.3.2 Create Identity Pools

1. Go to the Amazon Cognito console. Choose Identity pools and click Create identity pool.

User pools Identity pools (0) Info C Delete Create identity pool View and configure your identity pools. Identity pools enable you to create unique identities and assign permissions for users. C 1 > 0 Q Search identity pools by name or ID C Created time T Identity pool name Identity pool ID Created time T No identity pools You don't have any identity pools yet. Create identity pool	Amazon Cognito	×	Amazon Cognito > Identity pools	
Identity pool name ▲ Identity pool ID ▼ Created time ▼ No identity pools You don't have any identity pools yet. Create identity pool F F	User pools Identity pools		Identity pools (0) Info View and configure your identity pools. Identity pools enable you to create unique identities and as Q. Search identity pools by name or ID	C Delete Create identity pool ssign permissions for users.
No identity pools You don't have any identity pools yet. Create identity pool			Identity pool name Identity pool ID	▼ Created time ▼
			No identity pools You don't have any identity pools ye Create identity pool	et.

Figure 41. Create identity pool

2. On the **Configure identity pool trust** page, select **Guest access** and click **Next**.

Amazon Cognito > Identity pools	> Create identity pool
Step 1 Configure identity pool trust	Configure identity pool trust Info
Step 2 Configure permissions	Authentication Choose the sources that your identity pool trusts to generate identities and issue credentials.
Step 3 Configure properties	User access Info Configure your identity pool to generate credentials for users authenticated to third existence and existence to existence of existence of the second exist
Step 4 Review and create	 ☐ Authenticated access Issue credentials to authenticated users from trusted Identity providers. ☑ Guest access Use guest-access credentials to anyone with internet access. Use guest access with AWS resources such as public APIs and graphics assets.
	▲ An identity pool with guest access distributes AWS credentials that authorize access to resources in your AWS account. Your IAM policy for guest users must permit access only to resources that you want to be available to anyone on the internet. Learn more Z
	Cancel

Figure 42. Create identity pool trust

3. On the **Configure permissions** page, select **Create a new IAM role,** enter an **IAM role name**, and then click **Next**.

onfigure identity pool trust	Configure permissions Info				
ep 2 onfigure permissions	Each Amazon Cognito identity pool can have a default AWS identity and Access Management (IAM) role for authenticated and unauthenticated identities. If you don't change the default authenticated role for an identity provider, your users will receive credentials for the authenticated identities role.				
tep 3 onfigure properties	Guest role Info Configure the default IAM role that your anonymous guest users will assume through your identity pool.				
eview and create	IAM role Cheese a vota la voue accesse, actives des voo Consiste contra a poer one for you.				
	Create a new IAM role Create an IAM role with basic permissions and a trust relationship with your identity pool. Use an existing IAM role Choose a role in your account that you have configured to trust cognito-identity.amazonaws.com.				
	IAM role name Enter a name for your new IAM role. Cognito_MyTestDoorLockG Role names may be up to 64 characters long and can use alphanumeric characters, as well as the following special characters: +=_@				
	You're creating an IAM role with initial minimum permissions and a trust relationship with your identity pool. After you create your identity pool, add permissions in the IAM console.				
	View policy document View the policy document that Amazon Cognito has generated.				

Figure 43. Configure permissions

4. Enter an Identity pool name and click **Next**.

Amazon Cognito > Identity pools > 0	Create identity pool
Step 1 Configure identity pool trust	Configure properties Info
Step 2 <u>Configure permissions</u>	Identity pool name Create a friendly name for your identity pool.
Step 3 Configure properties Step 4 Review and create	Name MyldentityPool_DoorLockG Image: Special characters: + = , . @ -
	Basic (classic) authentication info Activate the classic authentication flow if your app relies on separate API requests to retrieve an identity token, and then to assume a role using that token. When you activate the classic flow, you can still use the recommended enhanced flow. Basic authentication Activate basic flow
	Tags (0) - optional info Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this identity pool.
	No tags associated with the resource. Add new tag You can add up to 50 tags.
	Cancel Previous Next

Figure 44. Configure properties

5. Review the selected items and click **Create identity pool**. Then, the created identity pools appear in the list.

amazon Cognito ×	Amazon Cognito > Identity pools		
lser pools dentity pools	Identity pools (2) Info View and configure your identity pools. Identity	pools enable you to create unique identities and assign permissions for	Create identity pool users.
	Q. Search identity pools by name or ID		< 1 > @
	Identity pool name	Identity pool ID 🗢	Created time v
	O MyldentityPool_DoorLock	ap-northeast-2:374beffc-b245-4bc8-972a- d5a441f2f271	4 hours ago
	O MyldentityPool DoorLockG	ap-northeast-2:2cbc2697-8f4a-486d-9aef- 0c9c613c3825	4 hours ago
	4		

Figure 45. Created identity pools

6.1.4 Set Up AWS IAM

While creating an identity pool, you should update the IAM roles that the users assume. When a user logs in to the app, Amazon Cognito generates temporary AWS credentials for the user. These temporary credentials are associated with a specific IAM role. The IAM role lets users define a set of permissions to access AWS resources. For more information, visit https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html.

- The roles in Cognito_MyTestDoorlockG are created automatically when the federation identity is created through Cognito Identity Pool.
- The device only needs an unauthorized role.

To set up AWS IAM:

1. Go to the IAM console and select Cognito_MyTestDoorLockG.

Identity and Access × Management (IAM)	IAM > Roles	
Q Search IAM	Roles (5) Info An IAM role is an identity you can create that has specific permissions with credentials assumed by entities that you trust.	that are valid for short durations. Roles can be
Dashboard	Q. Search	< 1 > ©
 Access management 	Role name	Trusted entities
User groups Users	AWSServiceRoleForSupport	AWS Service: support (Servic
Roles	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor
Policies	Cognito MyTestDoorLock	Identity Provider: cognito-id
Account settings	Cognito_MyTestDoorLockG	Identity Provider: cognito-id
Access reports	Log-MyAct-IAM-Role	AWS Service: iot
Access analyzer	4	

Figure 46. IAM role



2. Expand Add permissions and click Attach policies.

Permissions	to 10 managed) Info					
C Sin	nulate 🖸	Remove	Add permissions				
			Attach policies				
Q Search			Create inline polic	У			
					<	1 >	0

Figure 47. Attach policies

3. Search for the policy name of AWSIoTFullAccess and click Add permissions.

Current permissions policie	es (1)		
Other permissions policies (1	Filter by Type X All types	▼ 2 matches	< 1 > ©
Policy name	▲ Туре	▼ Description	
AWSIoTFleetHubFede	aration AWS managed	aged Federation access for IoT Fleet Hub app	
AWSIoTFullAccess	AWS managed	This policy gives	full access to the AWS Io

Figure 48. AWSIoTFullAccess policy

NOTE

AWSIoTFullAccess policy is not recommended for production.

4. Search for the policy name of AmazonS3FullAccess and click Add permissions.

Add permissions to Cognito_DoorlockUnauth_Role Attach Permissions						
Create policy						
Filter policies V Q. AmazonS3FullAccess						
Policy name 👻	Туре	Used as	Description			
AmazonS3FullAccess	AWS managed	Permissions policy (1)	Provides full access to all buckets via the AWS Management Console.			

Figure 49. AmazonS3FullAccess policy

NOTE

- AmazonS3FullAccess policy is not recommended for production.
- 5. The attached policies appear in the list.
| Permissions Trust relationships | Tags Access Advi | sor Revoke sessions |
|--|--|-----------------------|
| Permissions policies (3) Info You can attach up to 10 managed policies. C Simulate 2 Remove Q Search | Add permissions
Filter by Type
All types | ▼
▼
< 1 > ⊗ |
| Policy name | Туре | ▼ Attached entities ▼ |
| □ | AWS managed | 2 |
| □ | AWS managed | 2 |
| | Customer managed | 1 |

Figure 50. Attached policies

6.1.5 Create Amazon S3 Bucket

Every object in Amazon S3 is stored in a bucket. Before storing data in Amazon S3, you need to create a bucket. To create S3 bucket:

- 1. In the Amazon S3 console, in the left navigation pane, select **Buckets**, and click **Create bucket**.
- 2. On the **Create bucket** page, in the **Bucket name** field, type a bucket name.
- For Region, choose the AWS region where you want the bucket to reside, and click Create bucket.
 When Amazon S3 successfully creates the bucket, the console displays an empty bucket in the Buckets pane.



7. Build and Run Reference Application

Door lock reference application is available on the Renesas official website.

NOTE

Go to the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi), click Product Selector, check one of MOD devices, and scroll down to the Software Downloads section. Find "AWS IoT Reference" or type it in the search box, and then select the reference package and download.

For more detail, see DA16200 DA16600 FreeRTOS Getting Started Guide, Ref. [3].

7.1 Reference DA16200/DA16600 SDK Setting

7.1.1 Edit Endpoint

- 1. Change the AWS_USER_MQTT_HOST in the **app_aws_user_conf.h** file as follows:
 - AWS IoT > Settings

```
#define AWS_USER_MQTT_HOST "(account-specific-prefix).iot.(aws-region).amazonaws.com"
```

2. Build the SDK and then update the image.

7.1.2 Edit Thing Name

7.1.2.1 Edit Thing Name with Console Command

You can directly rename things using the console commands without the need to build an SDK. Renesas recommends using console commands over other methods for changing a thing name.

If the test board is running, run the factory command first, and then proceed to NVRAM as follows.

[/DA16200] #	
[/DA16200] # nvram	
Command-List is changed, "NVRAM"	
[/DA16200/NVRAM] #	
[/DA16200/NVRAM] # setenv APP_THINGNAME USER_THING_NAME	// write user thing name
[/DA16200/NVRAM] # getenv	// read user thing name
[/DA16200/NVRAM] # unsetenv APP_THINGNAME	// remove user thing name
[/DA16200/NVRAM] #	

Then, complete the provisioning process.

```
[/DA16200] # nvram
Command-List is changed, "NVRAM"
[/DA16200/NVRAM] # setenv APP_THINGNAME APP-DOORLOCK-1
[/DA16200/NVRAM] # getenv
Total length (411)
APP_THINGNAME (STR,15) ..... APP-DOORLOCK-1
N1_Profile (STR,02) ..... 1
N1_mode (STR,02) ..... 1
N1_mode (STR,02) ..... 2
SYSMODE (STR,02) ..... 1
N1_ssid (STR,17) ..... "Renesas_DA16200"
N1_psk (STR,13) ..... "1234567890"
N1_proto (STR,04) ..... RSN
```



N1_key_mgmt (STR,08) WPA-PSK
country_code (STR,03) KR
1:IPADDR (STR,09) 10.0.0.1
1:NETMASK (STR,14) 255.255.255.0
1:GATEWAY (STR,09) 10.0.0.1
1:DNSSVR (STR,08) 8.8.8.8
USEDHCPD (STR,02) 1
DHCPD_IPCNT (STR,03) 10
DHCPD_TIME (STR,05) 3600
DHCPD_S_IP (STR,09) 10.0.0.2
DHCPD_E_IP (STR,10) 10.0.0.11
DHCPD_DNS (STR,08) 8.8.8.8
[/DA16200/NVRAM] #

7.1.2.2 Edit Thing Name in Configuration File

If the thing name does not exist in NVRAM, the predefined name located in the first header is stored in NVRAM. Change AWS_USER_MY_THING_NAME in the **app_thing_manager.h** file, then build the SDK and update the image:

```
/*
 * USER Thing name define
 * Generic SDK default : "DA16200"
 * AWS IOT default : "IOT-SENSOR-46" or "FAE-DOORLOCK-4" or "assigned_thing_name"
 */
#define APP_USER_MY_THING_NAME "FAE-DOORLOCK-4"
```

7.1.3 Edit Image File Name for OTA

To test the OTA update, edit the **app_aws_user_conf.h** file in the DA16200 SDK and modify the file names to match the image file names that are uploaded to the Amazon S3 bucket.

```
#if defined(__BLE_COMBO_REF__)
#define RTOS_NAME "DA16600_FRTOS-GEN01.img"
#define BLE_NAME "DA16600_BLE_OTA.img"
#else
#define RTOS_NAME "DA16200_FRTOS-GEN01.img"
#endif
```

7.1.4 Connect Certificates to Thing

To authenticate the device with AWS IoT, the device must contain the **Root CA**, **Client Certificate**, and **Client Private Key**. For more information, see https://docs.aws.amazon.com/iot/latest/developerguide/iot-security-identity.html.

To add these certificates to the device, edit **app_aws_certi.h** and insert the certificates downloaded from AWS as follows:



<pre>"MITICPUTCALINGAWTBAGITTERNYESTM/jA054VB41kBn1j2kyjANBgkghkiG9w0BAQSF\n" \ "ADASMQswQQTDWQQGewJVUZEHMA0GA1UEChMSQM1hem9uARdvaFWTDVQQExBBbWF6\n" \ "DASMQswQQTDWQQGewJVUZEHMA0GA1UEChMSQM1hem9uARdvaFWTDVQQExBBbWF6\n" \ "DASMQswQQTDWQQGewJVUZEHMA0GA1UEChMSQM1hem9uARdvaFWTDVQQExBBbWF6\n" \ "DASMQswQQTDWQQGewJVUZEHMA0GA1UEChMSQW1hem9uARdvaFWTDVQQExBBbWF6\n" \ "DASMQswQQTDWQQGewJVUZEHMA0GA1UEChMSQW1hem9uARdvaFWTDVQQExBBbWF6\n" \ ""O/LIGUTUAMTBQtFHRBhBjrdkPSHCa2XV4cdFyQzR1b1dZwgJcJmApzyMZF06IQ6XU\n" \ "SMs1+yMRQ+hDK0JioaldXgjUKK642M4UWEU8062xJNDd2ZhwLnQdeXeGADDkpy\n" \ "SMs1+yMRQ+hDK0JioaldXgjUKK642M4UWEU8062xJNDd2ZhwLnQdeXeGADDkpy\n" \ "TKQREhCQDa2sG4q5WTF468SQrvG5\n" \ ""END CERTIFICATE\n" \ #define democonfigCLIENT_CERTIFICATE_FEM "BEGIN CERTIFICATE\n" \ "MITIDMjCQAKKgAwIBAgIVAIqSKvd/Qq2E92LeQNZGK/iPw2Q4A0GCSqGSID3DQEB\n" \ "WUAME0XS2BJBgNVBAsdQkFtYQevbiBXZWIgUZVydn1j2Q4gTz1BbWF6b24u29t\n" \ "TGCAbJQy2XprqPpBo3zuMqmSi55usIXj+2B4Xg9Zutim+8J7DHQbfHAGZwiAEN\n" \ "TGCAbJQy2XprqPpBo3zuMqmSi55usIXj+2B4Xg9Zutim+8J7DHQbfHAGZwiAEN\n" \ ""TGCAbJQy2XprqPpBo3zuMqmSi55usIXj+2B4Xg9Zutim+8J7DHQbfHAGZwiAEN\n" \ "DOIExCQE2DAnmfvAR8qSuLmEBKz2ILxQX5+CO4wes8R5E5NICK2QrKgr9NJKA=\n" \ ""END CERTIFICATE\n" \ #define democonfigCLIENT_FRIVATE_KEY_FEM "BEGIN RSA PRIVATE KEY\n" \ #define democonfigCLIENT_FRIVATE_KEY_FEM "BEGIN RSA PRIVATE KEY\n" \ #UTIBpAIRAAKCQQEA2FAGze8c4vALLogdeGK1fzD1166vD0p62x3C83TqSiCG84iz\n" \ ""UITBpAIRAAKCQQEA2FAGze8c4vALLogdeGK1fzD1166vD0p62x3C83TqSiC684iz\n" \ ""UITBpAIRAAKCQQEA2FAGze8c4vALLogdeGK1fzD1166vD0p62x3C83TqSiC684iz\n" \ ""UITBpAIRAAKCQQEA2FAGze8c4vALLogdeGK1fzD1166vD0p62x3C83TqSiC684iz\n" \ ""UITBpAIRAAKCQQEA2FAGZ8c4vALLogdeGK1fzD1166vD0p62x3C83TqSiC684iz\n" \ ""UITBpAIRAAKCQQEA2FAGZ8c4vALLogdeGK1fzD116fvD0p62x3C8TTqSiC684iz\n" \ ""UITBpAIRAAKCQQEA2FAGZ8c4ALLogdeGK1fzD1166vD0p62x3C8TTqSiC684iz\n" \ ""UITBpAIRAAKCQQEA2FASGE6c4ALLogdeGK1fzD116fvDAfz8c3GEVNTrR/Mo8\n" \ ""IITBpTFRISAbbUIJV-20XZGSC/HASJMWGKkeAZJWANDKkeAgV3NBBZEBKGZLXVY790\n" \ ""IITBFAIRAA</pre>	#define democonfigROOT_CA_PEM "BEGIN CERTIFICATE\n" \
<pre>"ADASMQ:swCQYIDQQGewJUZEEMA0GALUECHMCQWIhem9uMRkwFwYDQQDExeBohF6\n" \ "b24gUm9vdGEDQSAv6#4XDTELMDUyAjAwEAwAFGXDTM4MDEx0ZwMDAwFGxOTEL\n" \ "b24gUm9vdGEDQSAv6#4XDTELMDUyAjAwEAwAFGXDTM4MDEx0ZwMDAwFGxOTEL\n" \ "b24gUm9vdGEDQSAv6#4XDTELMDUyAjAwEAwAFGXDTM4MDEx0ZwMDAwFGxOTEL\n" \ "b4dsinsvtremeter the the theory of theory of the theory of the theory of the theory of theory of theory of theory of theory of theory of the theory of theory of theory of the theory of the theory of theory of theory of theory of theory of t</pre>	"MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF\n" \
<pre>"b24gUn9vdCBDQSAxMB4KDTE1MDUyAjAwADAwAFeXDTM4MDExN2AwADBwAFeoXOTEL\n" \ "o/ufQJVEMVT8QEPHEN8jrdkFSHCa2XV4odFyQzRlbldzwgJcJmApzyMZFo6IQ6XU\n" \ "SMs1+yARQ+hDKXJialdXgjUkK642M4W+EV8ob2xJNDd2ZmkInoQdeXeGADbkpy\n" \ "rcfKEDoonc2sG4qSMTP468Sgvvc5\n" \ "MIIIDWjCcAkKgAwIBadVLENVGvc2pkEd0AU2Gk/iEv204Au30CSqSID3DQEB\n" \ "CvUAME0xsEBJBqNVBasWQFrYXpvbiBXZBIgU2VydnjjZvvdfIzBWF6b24U29t\n" \ "TCaObDU22XprqPpBo3ZuMgnSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "TCaObDU22XprqPpBo3ZuMgnSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "TCaObDU22XprqPpBo3ZuMgnSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "TCaObDU22XprqPpBo3ZuMgnSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "TCaObDU22XprqPpBo3ZuMgnSi55uslXj+2B4XgPZUTM+VC2rKgrNXHA=\n" \ "TCaObDU22XprqPpBo3ZuMgnSi55uslXj+2B4XgPZUTM+\ "TCaObDU22KprqPBBo3ZuMgnSi55uslXj+2B4XgPZUMTOvX0+i02Be7w30\n" \ "TCaObDU22XprqPBBo3ZuMgnSi55uslXj+2D16EYDDp62x3CRTFZNba\n" \ "MIIIPAJIBAKCAQEA2fwZc4wLargdeGR1fzD116EYDDp62x3CRTFZNba\n" \ "TD1TEfSBobKsD1jV+2HzZVSXtgC5/fzsFJMWpeNRX+9vEFICCfxoygGM0VoVy\n" \ "TD1TEfSBbksU1jV+2HzZVSXtgC5/fzsFJMWpeNRX+9vEFICCfxoygGM0VoVy\n" \ "TCTGFBbksU1jV+2HzZVSXtgC5/fzsFJMWpeNRX+9vEFICCfxoygGM0VoVy\n" \ "TCTGFBbksU1jV+2HzZVSXtgC6/fzsFJMWpeNR</pre>	"ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6\n" \
<pre> "o/utQJVtMT8QtPHRh8jrdkPSHCa2XV4cdFyQzRlbldZwgJcJmApzyMZFo6IQ6XU\n" \ "%MsI+yMRQ+hDKXJioaldXgjUKK642M4UwtEV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n" \ "%qKRfboQno2sG4q5WTP468SQvvG5\n" \ "~~~~END CERTIFICATE~~~~~\n" #define democonfigCLIENT_CERTIFICATE_PEM "~~~~BEGIN CERTIFICATE~~~~~\n" \ "MIILUMJCCAKKgAwIEAgIVALqSKvd/Qq2E92LeQMN2Gk/iPw2GA4D0CSqGSID3DDEN\n" \ "~CMUMMEOXS2BJBgNVBASMQXFtYXpvb1EXZWIgUZVydmljZXdyTz1EbWF6b24UY29t\n" \ "MIILUMJCCAKKgAwIEAgIVALqSKvd/Qq2E92LeQMN2Gk/iPw2GA4D0CSqGSID3DDEN\n" \ "MUAMEOXS2BJBgNVBASMQXFtYXpvb1EXZWIgUZVydmljZXdyTz1EbWF6b24UY29t\n" \ "MIILUMJCCAKKgAwIEAgIVALqSKvd/Qq2E92LeQMN2Gk/iPw2GA4D0CSqGSID3DDEN\n" \ "MUILUMJCCAKKgAwIEAgIVALqSKvd/Qq2E92LeQMN2Gk/iPw2GA4D0CSqGSID3DDEN\n" \ "MUILUMFOGAKIIFNUPVdhc2hpbmd0b24g21VU2AeFv0xODEyMDYwNjQw\n" \ "TCaCMbJQy2XprqPpBo32UMqmSi55uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ "%0GKbJQ2XprqPpBo32UMqmSi55uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ "%0G62CQB2ADnmfvAR8gSuImIBfKz2IIxqX5+CO4wes8RH5pNICK2pKgr9NJkA=\n" \ "%0662CQB2ADnmfvAR8gSuImIBfKz2IIxqX5+CO4wes8RH5pNICK2pKgr9NJkA=\n" \ "%1HIpAITBAAKCAQEA2fwGze8cV4ALJcgdeGRLfzD1166YD0p62x3C8TqSiCGEMiz\" \ "%Uqk6n17/6CXFBodFAh6adTxet5tL5mGIgLnKYFtt7Iyj10T8hpxT1Xxp7YIZEDIw\n" \ "F19fptFRi5KncVhs9sICqJBmwKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "F19fptFRi5KncVhs9sICqJBmwKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "%2TTGFSAbksUIJV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2TTGFSAbksUIJV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2TTGFSAbksUIJV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2TTGFSAbksUIJV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2TTGFSAbksUIJV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2G1geirHUDYgog9XzGKATXc3K/m7JdyOcWkbf54nhzcEqjRv1DhCA=\n" \ "%2TTGFSAbksUIJV+2HZZVSTgsCS/fzsFjMWYpeNRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2G1geirHUDYgog9XzGKATXc3K/m7JdyOcWkbf54nhzcEqjRv1DhCA=\n" \ "%2TTGFSAbksUIJV+2HZZVSTgsCS/fzsF/MYPENRX3+9wtFfGCfxogqGM0JvOyY\n" \ "%2G1geirHUDYgog9XzGKATXc3K/m7JdyOcWkb654nhzcEqjRv1DhCA=\n" \ "%2TTGFSAbksUIJV+2</pre>	"b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL\n" \
<pre>"o/ufQJVLMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzRlbldZwgJcJmApzyMZFo6IQ6XU\n" \ "SMsI+yMEQ+hDKXJioaldXgjUKK642M4UwtEV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n" \ "rqKrfboQnoZsG4q5WTP468SQvvG5\n" \ "r====END CERTIFICATE=====\n" fdefine democonfigCLIENT_CERTIFICATE_FEM "=====BEGIN CERTIFICATE=====\n" \ "MIIDNjCCAKKgAwIBAgIVAIq5Kvd/Qq2E92LeQMN2Gk/iFw2G4A0GCSqGSTb3DQEB\n" \ "MUIDNjCCAKKgAwIBAgIVAIq5Kvd/Qq2E92LeQMN2Gk/iFw2G4A0GCSqGSTb3DQEB\n" \ "MUIDNjCCAKKgAwIBAgIVAIq5Kvd/Qq2E92LeQMN2Gk/iFw2G4A0GCSqGSTb3DQEB\n" \ "MUIDNjCCAKKgAwIBAgIVAIq5Kvd/Qq2E92LeQMN2Gk/iFw2G4A0GCSqGSTb3DQEB\n" \ "MUIDNjCCAKKgAwIBAgIVAIq5Kvd/Qq2E92LeQMN2Gk/iFw2G4A0GCSqGSTb3DQEB\n" \ "MUIDNGCASgBJBgNVBAsMQXFFYXpvbiBXZWIgUZVydnljZXMgTz1BbWF6b24UY29t\n" \ "IELUYY4gTDITZWF0dGxlIFNUFVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYvNjQw\n" \ "IELUYY4gTDITZWF0dGxlIFNUFVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYvNjQw\n" \ "TGOKbJQy2XprqPpBo3ZuWgmSi55uslXj+2B4XgPzutim++8J7DHQbfHAGZwiAFN\n" \ "90TNIhZBd187Ga07p0db03KcBqz8dEMaABCORK39LqJ52dQMT/owx0+i02Be7w30\n" \ "90T01hZBd187Ga07p0db03KcBqz8dEMaABCORK39LqJ52dQMT/owx0+i02Be7w30\n" \ "90T062CQE2A0nmfvAR8gSuImIBFKz2I1xQX5+C04wes8RH5pNICK2QrKgr9NJNA=\n" \ "90TMIEpAIBAAKCQQEA2fwGze8cV4ALJcgdeGRIfzDI166YD0p62x3CBITqSiC6B4iz\n" \ "WIIEpAIBAAKCQQEA2fwGze8cV4ALJcgdeGRIfzDI166YD0p62x3CBITqSiC6B4iz\n" \ "WIIEpAIBAAKCQQEA2fwGze8cV4ALJcgdeGRIfzDI166YD0p62x3CBITqSiC6B4iz\n" \ "Ugk6n17/6cXP8odFh6adfIxet5L5mGLgLnKYFtt7Iyj10T8hpxTI1xp7TYZRDlw\n" \ "FI9fptPRi5KncVhs9sICqJEmwKDv6LUwIlefrofM+6uX7gBssGUeVnrrR/Mo8\n" \ "EOP11QKBqQCDnAVbfrXC+455UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "PTTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWpeNRX3+9wtFfGCfxoyqGW0VOYYN" \ "Kg61geirHUDYgog9XzGKATXC3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "Kg61geirHUDYgog9XzGKATXC3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "</pre>	
<pre>"MsI+yMQ+hEXJioaldXgjUkK642M4UwtEV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n" \ "rqXRfboQnoZsG4q5WTP468SQvvG5\n" \ "rqXRfboQnoZsG4q5WTP468SQvvG5\n" \ "r====END CERTIFICATE=====-\n" #define democonfigCLIENT_CERTIFICATE_FEM "=====BEGIN CERTIFICATE=====-\n" \ "MIIDWjCCAkkgawIBAgIVAIqSKvd/Qq2E92leQMNZGk/iPw2G4A0GCSqGSIb3DQEE\n" \ "MUIDWjCCAkkgawIBAgIVAIqSKvd/Qq2E92leQMNZGk/iPw2G4A0GCSqGSIb3DQEE\n" \ "MUIDWjCCAkkgawIBAgIVAIqSKvd/Qq2E92leQMNZGk/iPw2G4A0GCSqGSIb3DQEE\n" \ "MUIDWjCCAkkgawIBAgIVAIqSKvd/Qq2E92leQMNZGk/iPw2G4A0GCSqGSIb3DQEE\n" \ "MUIDWjCCAkkgawIBAgIVAIqSKvd/Qq2E92leQMNZGk/iPw2G4A0GCSqGSIb3DQEE\n" \ "MUIDWjCCAkkgawIBAgIVEDAsMQkFtYXpvbiBXZWIgU2VydmjjZMgT21BbWF6b24UY29t\n" \ "MUIDWjCQAkgawIBAgNQkFtYXpvbiBXZWIgU2VydmjjZMgT21BbWF6b24UY29t\n" \ "Telavy4gTD1TZWF0dGx1IFNUPVdhc2hpbmd0b24gQ21VUzAeFw0xODEyMDYwNjQw\n" \ "TCaCwbJQy2XprqPpBo3ZuWqmSi55usIXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "90TNIhZBd187Ga07p0db03KcBQs8dEMaABCORK39LqJ5ZdQMT/Owx0+i02Be7w30\n" \ "70062CQB2A0nmfvAR8gSuImIBfKz2I1xQX5+C04wes8RH5pNIOK2QrKgr9NJkA==\n" \ "70062CQB2A0nmfvAR8gSuImIBfKz2I1xQX5+C04wes8RH5pNIOK2QrKgr9NJkA==\n" \ "*Gefine democonfigCLIENT_FRIVATE_KEY_FEM "=====BEGIN RSA PRIVATE KEY=====\n" \ #define democonfigCLIENT_FRIVATE_KEY_FEM "======BEGIN RSA PRIVATE KEY=====\n" \ "MIIEpAIEAAKCAQA2AfwGza6v4ALJcgdeGRIfzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIEAAKCAQA2fwGze6v4ALJcgdeGRIfzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "Ugk6n17/6cXF8odFAh6adTxet5L5mGLJnkYFt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "FI9fptPRi5KncVhs9sICqJemwTDv6LDwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EDPI1QKBqQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WIBSZFbbGzIxrVy790\n" \ "FITGFSAbksUIjV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGNUJVOYY\n" \ "Kg61geirHUDYgog9XZGKATXC3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "</pre>	"o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1b1dZwgJcJmApzyMZFo6IQ6XU\n" \
<pre>"rqxRfboQno2sG4q5WTP468SQvvG5\n" \ "END CERTIFICATE\n" #define democonfigCLIENT_CERTIFICATE_PEM "BEGIN CERTIFICATE\n" \ "MIIDWjCCAkKgAwIBAgIVAIqSKvd/Qq2E92leQWN2Gk/iPw2GMA0GCSqGSIb3DQEB\n" \ "WUAME0xS2BJBqNVBAsMQkFtYXpvbiBXZWIgU2Vydnlj2XMgTz1BbWF6b24uY29t\n" \ "WUAME0xS2BJBqNVBAsMQkFtYXpvbiBXZWIgU2Vydnlj2XMgTz1BbWF6b24uY29t\n" \ "WUAME0xS2BJBqNVBASMQkFtYXpvbiBXZWIgU2Vydnlj2XMgTz1BbWF6b24uY29t\n" \ "WUAME0xS2BJBqNVBASMQkFtYXpvbiBXZWIgU2Vydnlj2XMgTz1BbWF6b24uY29t\n" \ "UGUAME0xS2BJBqNVBASMQkFtYXpvbiBXZWIgU2Vydnlj2XMgTz1BbWF6b24uY29t\n" \ "WUAME0xS2BJBqNVBASMQkFtYXpvbiBXZWIgU2Vydnlj2XMgTz1BbWF6b24uY29t\n" \ "UELuYY4gTD1TZWF0dGx1IFNUFVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \ "TcaCwbJQy2XprqPpBo3zuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "90TNIhZBd187Ga07p0db03KcBQs8dEMaABCORK39LgJ5ZdQMf/Owx0+i02Be7w30\n" \ "7006zCQB2A0nmfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0nmfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0nmfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0nmfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0nmfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0nmfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0mfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0mfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0mfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr9NJkA=\n" \ "7006zCQB2A0mfvAR8gsuImIBfKz211xQX5+C04wes8RH5pNICK2QrKgr7YZRblw\n" \ "711EpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166tD0p62x3C8ITqS1C6B4iz\n" \ "719fptFR15KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrr/Mo8\n" \ "7EOP11QKBgQCDnAVbfrXC+455UNwxGHw4cZJwAvOkkeApV3WIBSZFbbGzIxrVy79O\n" \ "7ETTGfSAbksU1jV+2HZZVSXtgsCS/fzsFjMWpeNRX3+9wtFfGCfxoygGM0JvOyY\n" \ "72FTGfSAbksU1jV+2HZZVSXtgsCS/fzsFjMWpeNRX3+9wtFfGCfxoygGM0JvOyY\n" \ "%g61geirHUDYgog9XzGKATXc3K/m7JdyoC%dbf54hnzcEqjRv1DhCa=\n" \ "</pre>	"5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n" \
<pre>"END CERTIFICATE\n" #define democonfigCLIENT_CERTIFICATE_FEM "BEGIN CERTIFICATE\n" \ "MIIDMJCCAKKgAwIBAgIVAIgSKvd/Qq2E92leQMN2GK/iPw2GMA0GCSqGSIb3DQEE\n" \ "GwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2Vydnjj2XMgTz1BbwF6b24uY29t\n" \ "GwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2Vydnjj2XMgTz1BbwF6b24uY29t\n" \ "GwUAME0xSzBJBgNVBASMQkFtYXpvbiBXZWIgU2Vydnjj2XMgTz1BbwF6b24uY29t\n" \ "GwUAME0xSzBJBgNVBASMQkFtYXpvbiBXZWIgU2Vydnjj2XMgTz1BbwF6b24uY29t\n" \ "TELuYy4gTD1T2WF0dGx1IFNUFVdho2hpbmd0b24gQ21VU2AeFw0xODEyADYwNjgw\n" \ "TcaCwbJQy2XprqPpBo3ZuWgmS155uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ "TcaCwbJQy2XprqPpBo3ZuWgmS155uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ "TcaCwbJQ92XprqPpBo3ZuWgmS155uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ "ToaCwbJQ92XprqPpBo3ZuWgmS155uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ "ToaCwbJQ92XgMarKaZeScV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "FI9fptPrI5KncVhs9sICqJEmvKTDv6UJWIIEfrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EDP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WIBSZFbbGzIxrVY790\n" \ "FITGfSAbksU1jV+2HZZVSxtgsCS/fzsFjMWYpeNRX3+9wtFfCCfxoygGW0JvOyY\n" \ "Kg6lgeirHUDYg0g9XzGKATXC3K/m7Jdy0cWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA FRIVATE KEY\n" \ "</pre>	"rqXRfboQnoZsG4q5WTP468SQvvG5\n" \
<pre>#define democonfigCLIENT_CERTIFICATE_PEM "BEGIN CERTIFICATE\n" \ "MIIDMjCCAkkgAwIBAgIVALqSKvd/Qq2E921eQMN2Gk/iFw2GMA0GCSqGSIb3DQEB\n" \ "GuUAME0xSzBJBgNVBASMQkFtYXpvbiBXZWIgU2VydnljZXMgTz1BbWF6b24uY29t\n" \ "GUUAME0xSzBJBgNVBASMQkFtYXpvbiBXZWIgU2VydnljZXMgTz1BbWF6b24uY29t\n" \ "IELuYy4gTDITZWF0dGx1IFNUFVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \ "TacAvbJQy2XprqPpBo3ZuMgmSi55uslXj+2B4XgPZutim++&J7DHQbfHAGZwiAFN\n" \ "90TN1hZBdI87Ga07p0db03KcBQs8dBMaABCORK39LqJ5ZdQMT/Owx0+i02Be7w30\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "WIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "WIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "WIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD116fYDvF6UX7gEhssGUeVnrrR/M08\n" \ "EDP11QKBqQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOKkeApV3W1BSZFbbGzIxrVy790\n" \ "PETTGfSAbksU1jV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "Kg61geirHUDYg0g9XzGKATXC3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA FRIVATE KEY\n" \ </pre>	"END CERTIFICATE\n"
<pre>#define democonfigCLIENT_CERTIFICATE_PEM "BEGIN CERTIFICATE\n" \ "MIIDWjCCAkKyAwIBAGIVAIqSKvd/Qq2E92leQMN2GK/iPw2CMAOGCSqGSIb3DQEB\n" \ "CwUAMEOxS2BJBgNVBASMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1EbWF6b24uY29t\n" \ "CwUAMEOxS2BJBgNVBASMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1EbWF6b24uY29t\n" \ "TELuYy4gTD1TZWF0dGxlIFNUFVdhc2hptmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \ "TCaCMbJQy2XprqPpBo3ZuWqmSi55us1Xj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "90TN1hZBd187Ga07p0db03KcBQs8dEMAABCORK39IqJ5ZdQMT/Owx0+i02Be7w30\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "FI9fptPR15KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXc+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "%g61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "%g61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \ </pre>	
<pre>"MIIDWjCCAkKgAwIBAgIVAIqSKvd/Qq2E92LeQMN2Gk/iFw2GMAOGCSqGSIb3DQEB\n" \ "CwUAME0xSzBJBgNVBASMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t\n" \ "TEluYy4gTDITZWF0dGxlIFNUFVdhc2hpbmd0b24gQzIVUzAeFw0xODEyMDYwNjQw\n" \ "TcaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ ""ToaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgP2utim++8J7DHQbfHAGZwiAFN\n" \ ""OofacQB2A0nmfvAR8gSuImIBfKz2I1xQX5+cO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ ""END CERTIFICATE\n" \ #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "F19fptPRi5KncVhs9sICqJEmwKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXc+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "%g61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhczEqjRv1DhCA=\n" \ "</pre>	#define democonfigCLIENT_CERTIFICATE_PEM "BEGIN CERTIFICATE\n" \
<pre>"CwUAME0xSzBJBGNVBAsMQkFtYXpvbiBXZWIgU2VydnljZXMgTz1BbWF6b24uY29t\n" \ "TEluYy4gTD1TZWF0dGx1IFNUFVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \ "TCaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "ToaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "To06zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "MIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqY77YZRDlw\n" \ "FI9fptPrFi5KncVhs9sICQJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "FEOFIIQKBgQCDnAVbfrXC3KgGHv4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "Kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA FRIVATE KEY\n" \ "END RSA FRIVATE KEY</pre>	"MIIDWjCCAkKgAwIBAgIVAIqSKvd/Qq2E9ZleQWN2Gk/iPw2GMA0GCSqGSIb3DQEB\n" \
<pre>"IELuYy4gTD1TZWF0dGx11FNUFVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \ "TcaCwbJQy2XprqPpBo3ZuWqmSi55us1Xj+2B4XgPZutim++&J7DHQbfHAGZwiAFN\n" \ ""00TN1hZBd187Ga07p0db03KcBQs&dBMaABC0RK39LqJ5ZdQMT/Owx0+i02Be7w30\n" \ ""00f2CQB2A0nmfvAR8gSuImIBfKz2I1xQX5+C04wes8RH5pN10K2QrKgr9NJkA=\n" \ "7006zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+C04wes8RH5pN10K2QrKgr9NJkA=\n" \ "To06zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+C04wes8RH5pN10K2QrKgr9NJkA=\n" \ "TeNTIFLATE\n" #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "Ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYftt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5Kncvhs9sICqJEmvKTDv6IUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "F2TTGfSAbksU1jV+2HZZVSXtgsCS/fzsFjMWpeNRX3+9wtFfGCfxoygGWUJvOyY\n" \ "Kg61geirHUDYgog9XzGKATXC3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	"CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t\n" \
<pre> "TcaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ ""90TNlhZBd187Ga07p0db03KcBQs8dBMaABCORK39LqJ5ZdQMT/Owx0+i02Be7w30\n" \ ""00facCQB2A0nmfvAR8gSuImIBfKZ2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ ""00facCQB2A0nmfvAR8gSuImIBfKZ2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ ""00facCQB2A0nmfvAR8gSuImIBfKZ2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ ""00facCQB2A0nmfvAR8gSuImIBfKZ2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "To06zCQB2A0nmfvAR8gSuImIBfKZ2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ ""00facCQB2A0nmfvAR8gSuImIBfKZ2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ ""END CERTIFICATE\n" \ #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGRlfzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGRlfzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "Ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "ReoF11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "Kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcBqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \ </pre>	"IEluYy4gTD1TZWF0dGx1IFNUPVdhc2hpbmd0b24gQz1VUzAeFw0xODEyMDYwNjQw\n" \
<pre>"TcaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \ "90TNlhZBdI87Ga07p0db03KcBQs8dBMaABCORK39LqJ5ZdQMT/Owx0+iO2Be7w30\n" \ "7o06zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+cO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "7o06zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+cO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "7o06zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+cO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "END CERTIFICATE\n" #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGRlfzDl166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "Ref1geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "wg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	
<pre>"90TN1hZBd187Ga07p0db03KcBQs8dEMaABCORK39LqJ5ZdQMT/Owx0+i02Be7w30\n" \ "7o06zCQB2A0nmfvAR8gSuImIBfKz211xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "END CERTIFICATE\n" #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WIBSZFbbGzIxrVy790\n" \ "TETTGfSAbksU1jV+2HzZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	"TcaCwbJQy2XprqPpBo3ZuWqmSi55uslXj+2B4XgPZutim++8J7DHQbfHAGZwiAFN\n" \
<pre>"7006zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \ "END CERTIFICATE\n" #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WIBSZFbbGzIxrVy79O\n" \ "TeTTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	"90TN1hZBdI87Ga07p0db03KcBQs8dBMaABC0RK39LqJ5ZdQMT/Owx0+i02Be7w30\n" \
<pre>"END CERTIFICATE\n" #define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGRlfzDl166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxTlYxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \</pre>	"7006zCQB2A0nmfvAR8gSuImIBfKz2I1xQX5+CO4wes8RH5pNIOK2QrKgr9NJkA=\n" \
<pre>#define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGRlfzDl166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxTlYxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	"END CERTIFICATE\n"
<pre>#define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \ "MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGRlfzDl166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \</pre>	
<pre>"MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1166YD0p62x3C8ITqSiC6B4iz\n" \ "ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	#define democonfigCLIENT_PRIVATE_KEY_PEM "BEGIN RSA PRIVATE KEY\n" \
<pre>"ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \ "F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA=\n" \ "END RSA PRIVATE KEY\n" \</pre>	"MIIEpAIBAAKCAQEA2fwGze8cV4ALJcgdeGR1fzD1I66YD0p62x3C8ITqSiC6B4iz\n" \
<pre>"F19fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \ "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \ "END RSA PRIVATE KEY\n" \</pre>	"ugk6n17/6cXF8odFAh6adTxet5tL5mGLgLnkYFtt7Iyj10T8hpxT1Yxp7TYZRblw\n" \
<pre> "EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy79O\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \ "END RSA PRIVATE KEY\n" \</pre>	"Fl9fptPRi5KncVhs9sICqJEmvKTDv6LUwIlefrofMv+6uX7gEhssGUeVnrrR/Mo8\n" \
"EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy79O\n" \ "7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \ "END RSA PRIVATE KEY\n" \	
"7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \ "kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \ "END RSA PRIVATE KEY\n" \	"EOP11QKBgQCDnAVbfrXC+4S5UNwxGHw4cZJwAvOkkeApV3WlBSZFbbGzIxrVy790\n" \
"kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \ "END RSA PRIVATE KEY\n" \	"7ETTGfSAbksUljV+2HZZVSXtgsCS/fzsFjMWYpeNRX3+9wtFfGCfxoygGW0JvOyY\n" \
"END RSA PRIVATE KEY\n" \	"kg61geirHUDYgog9XzGKATXc3K/m7JdyOcWdbf54nhzcEqjRv1DhCA==\n" \
	"END RSA PRIVATE KEY\n" \

7.2 Reference Application in DA16200/DA16600

The following components shown in Figure 51 are required to run the application in DA16200/DA16600 through an Internet connection and AWS IoT server:

- AWS IoT reference application package
- DA16200/DA1660 EVB
- Router: Connection to internet
- Mobile device: Android/iOS application
- AWS account.





Figure 51. Architecture of AWS IoT

Install the mobile application by searching for **DA16200** or **DA16600** in the Google Pay Store or the Apple App Store on the mobile devices.

Provisioning is required for connection between DA16200/DA16600 and Router before connecting DA16200/DA16600 with AWS IoT hub. The provisioning can be done with the Renesas Wi-Fi Provisioning app on either an Android or iOS device. For details on how to install and provision the mobile app, see Ref. [4]. When provisioning is completed, select AWS IoT to open AWS application on mobile device.

7.2.1 Open Door

Figure 52 shows message flows of opening the door.



Figure 52. Message flows of opening door

The operation of **opening door** in Android app is shown in Figure 54.



PST-DOORLOCK-3		PST-DOORLO	оск-з		PST-DOORLOCK-3		
AWS Server Device		AWS Server	AWS Serve	er Device			
AWS IOT MQTT client		AWS IOT MQTT client		AWS IOT MQTT client		A	AWS IOT MQTT client
Publish topic: IPST000R.DOK.3/AppCentel Publish message: 3 pag.dore folios Subunche topic: (DOK.DOK.3).dok-RoberControl Last subinthe message: 1 mou_door closed		Publish topic : PST/DO0RLOCK-3/AppControl Publish message : 0 app, door open Subscribe topic : PST/DO0RLOCK-3/DeviceControl Last subcribe message : 1 mcu_door closed		Publish topic : PST-DOORLO Publish message : 0 app, do Subscribe topic : PST-DOORL Last subcribe message : 1 m	IX-31/AppCentrol or open OCK-3/Device-Control ixu_door opened		
AWS IoT Shadow		AWS IOT Sh		AWS IoT Shadow			
(densed') { betacome' have sor' } 'monoff' { 'monoff' { 'mo		factor The door is being on the factor is being on t	('desired': { 'welcome': 'aws-iot' 'percented': { 'avstored': 'aws-iot', 'organized': 'organized', 'barrey': 44, 'barrey': 43,	nr,			
Last reported Time : 2022-09-16 08:07:15 ± 건	,	Last reported Time : 2022-09-16 08:06:56 표전		Last reported Time : 202	2-09-16 08:07:03 초전		
Request Update Shadow		Request Update Shadow			Request Update Shadow		
Open door		Open door The door is clo	sed	Close door	The door is opened		
Open window The window is closed		Open window The window is	closed	Open window	The window is closed		
OTA Upda			OTA Update		OTA Update		

Figure 53. Open dooring on mobile app

Device Shadow state
{ "state": { "desired": { "welcome": "aws-iot" "welcome": "aws-iot" "welcome": "aws-iot"
<pre>""""""""""""""""""""""""""""""""""""</pre>
<pre>"windowStat": "closed", "doorStat": "opened", "temp": 44, "battery": 75, "doorState": false, "coorMathed", "coo"</pre>
"doorStateChange": 1, "DoorOpenMode": 0, "temperature": 40,

Figure 54. Shadow state when door is open

When the operation of opening door is completed, the console logs of the DA16200 appear as follows:

```
Count : 0, cmdNum = 4

mqtttype = 1

index(=3) matched

data type(shadow) = 0

call update sensor(need to be set variable): battery = 63

Count : 1, cmdNum = 5

mqtttype = 1

index(=2) matched

data type(shadow) = 2

call update sensor(need to be set variable): temperature = 28.000000
```



```
Count : 2, cmdNum = 6
mqtttype = 1
index(=1) matched
data type(shadow) = 1
call update sensor(need to be set variable): doorStat = opened
Count : 3, cmdNum = 7
mqtttype = 1
index(=0) matched
data type(shadow) = 1
call update sensor (need to be set variable): windowStat = closed
release response
publish (shadow sensor update) OK -
                                 payload:
"{"state":{"reported":{"windowStat":"closed","doorStat":"opened","temperature":28.0
00000,"battery":63,"OTAupdate":0,"OTAresult":"OTA UNKNOWN"}},"clientToken":"PST-DOORLOCK-3-0"}"
```

7.2.2 Close Door

Figure 55 shows message flows of closing door.



Figure 55. Message flows of closing door



The operation of **closing door** in Android app is shown in Figure 56.

PST-DOORLOCK-3			PST-DOORLOCK-3				PST-DOORLOCK-3		
AWS Server I	Device		AWS Serve	er	Device		AWS Serve	r D	evice
AWS IOT MQTT client			A	WS IOT MQTT client			A	WS IOT MQTT client	
Publish hopic: PST-DODR,OCK:3/AppControl Publish message: 0 app_Goor open Subscribe topic: PSTDODR,OCK://pvviceControl Last subcribe message: 1 mot_door opened			Publish topic : PST-DOORLOC Publish message : 0 app_doo Subscribe topic : PST-DOORLS Last subcribe message : 1 me	X-3/AppControl or close OCK-3/DeviceControl Cu_door opened			Publish topic : PST-DOORLOC Publish message : 0 app_doo Subscribe topic : PST-DOORLI Last subcribe message : 1 me	K-3/AppControl r close DCK-3/DeviceControl DCK-3/DeviceControl su_door closed	
AWS IoT Shadow				AWS IoT Shadow				AWS IoT Shadow	
{ designed" (designedsigned" (designed" (designed" (designed" (de			The door is being closed Fond To any of the door is being closed To any of the door is			('desired': { 'welcome': 'aws-iot' 'bwelcome': 'aws-iot', '0'TAupdate': 0, '0'TAupdate': 0, '0'TAupdate': 0, 'welcome': 'aws-iot', 'otoritat': 'closed', 'don'tat': 'closed', 'tamp': 44, 'battery': 64,	ne,		
Last reported Time : 2022-09-16 08:07:03 ± だ		·	Last reported Time : 2022	2-09-16 08:07:03 エゼ			Last reported Time : 2022	2-09-16 08:07:15 ±?!	
Request Update Shadow				Request Update Shadow				Request Update Shadow	
Close door The door is opened			Close door	The door is opened			Open door	The door is closed	
Open window The window is closed			Open window	The window is closed			Open window	The window is closed	
	OTA Update				OTA Update				OTA Update

Figure 56. Closing door on mobile app

Figure 57 shows the state of Shadow on the AWS IoT Hub when the operation for closing door is completed.

Device Shadow state
<pre>{ "state": { "desired": { "welcome": "aws-iot" }, "reported": { "welcome": "aws-iot", "OIAundate": 0 } } } </pre>
"OTAresult": "OTA_UNKNOWN", "windowStat": "closed", "doorStat": "closed",
"temp": 44, "battery": 76, "doorState": false, "openMethod": "app", "doorStateChange": 1, "DoorOpenMode": 0, "temperature": 41,

Figure 57. Shadow state when door is closed

When the operation of closing door is completed, the console logs of the DA16200/DA16600 appears as follows:



```
Count : 1, cmdNum = 5
mqtttype = 1
index(=2) matched
data type(shadow) = 2
call update sensor(need to be set variable): temperature = 41.000000
Count : 2, cmdNum = 6
mqtttype = 1
index(=1) matched
data type(shadow) = 1
call update sensor(need to be set variable): doorStat = closed
Count : 3, cmdNum = 7
mqtttype = 1
index(=0) matched
data type(shadow) = 1
call update sensor(need to be set variable): windowStat = closed
release response
publish (shadow sensor update) OK - payload:
"{"state": {"reported": {"windowStat":"closed", "doorStat":"closed", "temperature": 41.0
00000, "battery":76, "OTAupdate":0, "OTAresult":"OTA UNKNOWN"}}, "clientToken":"PST-DOORLOCK-3-0"}"
```



7.3 Reference Application in Host MCU

Application in the host MCU can control DA16200/DA16600 and connection between the host MCU and mobile phone through AWS IoT server using AT commands. Figure 58 shows the AWS IoT using firmware images for AT commands and the host MCU.



Figure 58. AWS IoT using firmware images for AT commands and host MCU

7.3.1 Download Package for Door Lock Reference Application in Host MCU

A firmware image for AT command and application in MCU are available on the official Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi).

The contents of the package are the following:

- DA16200 or \DA16600
 - Firmware images for the DA16200/DA16600 Wi-Fi devices.
 - Tera Term script for downloading the firmware images to the DA16200/DA16600 Wi-Fi device.
- DA16200\Script (\DA16600\Script)
 - Tera Term script that demonstrates how to use AT commands for AWS IoT using a personal computer and the DA16200/DA16600.
 - · Getting Stared with AT commands for AWS IoT
 - Introduces the DA16200/DA16600 AT commands for AWS IoT and describes how to set up the development environment and test the examples.
 - Describes how to connect an external host to the DA16200/DA16600 EVK for using the AT commands for AWS IoT.
 - $_{\odot}\,$ Describes the AT commands for AWS IoT command list.
- MCU
 - Sample project based on the RA6M4 development environment which demonstrates how to use AT commands for AWS IoT.

7.3.2 Hardware Connections between DA16200/DA16600 and Host MCU

The hardware components shown in Figure 59 are required to run door lock reference application using AT commands and the host MCU:

- DA16200/DA16600 EVK
- EK-RA6M4 board

- Windows laptop or personal computer.
- In addition, the following hardware connections are required for each operation:
- UART0: Programming firmware images and monitoring logs from DA16200/DA16600.
- UART1 or UART2: AT command interface between MCU and DA16200/DA16600.
- GPIO from the MCU to the DA16200/DA16600 to wake up the DA16200/DA16600 from DPM Low-power mode (DPM LPM).
- GPIO from the DA16200/DA16600 to the host MCU to wake up the MCU in Sleep mode.



Figure 59. Hardware configuration

Table 1 shows the pin connections between the DA16200/DA16600 EVK and the EK-RA6M4 board.

	Table	1.	Pin	connection
--	-------	----	-----	------------

	DA16200 EVK		DA	16600 EVK	EK-RA6M4 board	
Function	Pin number	Pin name	Pin number	Pin name	Pin number	Pin name
Ground	J3.18	GND	J2.12	GND	J24-7	GND
UART_TX	J4.11	TX1/GPIOA_4	J2.2	TX2/GPIOC_6	J23-2	D1/TXD
UART_RX	J4.12	RX1/GPIOA_5	J2.4	RX2/GPIOC_7	J23-1	D0/RXD
DA16200_WAKE_UP	J3.11	RTC_WAKE_UP2	SW1	RTC_WAKE_UP2	J23-6	D5/PWM
MCU_WAKE_UP	J4.18	GPIOA_11	J2.9	GPIOA_11	None	None

7.3.2.1 UART Connection for AT Commands

Table 2 shows the default configuration of UART1 (DA16200 EVB) or UART2 (DA16600 EVB) for AT commands.

Table 2. Default configuration for UART1 or UART2

Settings	Value
Baud Rate	115200
Data Bits	8
Parity	None

Settings	Value
Stop Bits	1
Flow Control (Hardware/Software)	None

The DA16200 EVB uses GPIOA_4 and GPIOA_5 for UART1 TX and UART1 RX, and the DA16600 EVB uses GPIOC_6 and GPIOC_7 for UART2 TX and UART2 RX by default. In addition, GND needs to be connected to the host MCU as shown in Figure 60.



Figure 60. Default UART hardware connection

If the GPIO pin configuration is changed using AT commands, other connections for UART1 can be used as shown in Figure 61. The following AT command is used for GPIOA_2 for UART1 TX and GPIOA_3 for UART1 RX. Table 3 shows the pin combination for UART1.

AT+AWS=SET NV_PIN_BMUX BMUX_UART1d

// GPIOA 2 and GPIOA 3 for UART1



Figure 61. Example of UART1 connection

Table 3. UART1 pin configuration

PIN MUX	GPIO	Signal name
PIN_AMUX	GPIOA_0	ТХ
	GPIOA_1	RX
PIN_BMUX	GPIOA_2	ТХ
	GPIOA_3	RX
PIN_CMUX	GPIOA_4	ТХ

PIN MUX	GPIO	Signal name		
	GPIOA_5	RX		
PIN_DMUX	GPIOA_6	ТХ		
	GPIOA_7	RX		

When Dynamic Power Management (DPM) mode is enabled and DA16200/DA16600 is in DPM LPM, the host MCU must wake up the DA16200/DA16600 from DPM LPM using RTC_WAKE_UP. Then, the host MCU can send or receive data over the network in DPM Fully Functional Mode (FFM). The wake-up event is triggered when the GPIO pin of the host MCU changes from Low to High and then back to Low.



Figure 62. Hardware connection for waking up DA16200/DA16600

The host MCU may be in Sleep mode when DA16200/DA16600 wakes up from DPM LPM and needs to send responses to the host MCU. In this scenario, the DA16200/DA16600 needs to wake up the host MCU from sleep using GPIO as shown in Figure 62. This connection is not required if the host MCU does not use sleep mode. GPIOA_11 is available on DA16200/DA16600 EVB for waking up the host MCU by default (see Figure 63) and it can be configured using the following AT commands:

AT+AWS SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A	// GPIO_A port
AT+AWS SET APP MCU WKAEUP PIN GPIO PIN11	// GPIO 11



Figure 63. Default pin configuration for waking up host MCU

Other GPIOs in the DA16200 EVB can be used for waking up the host MCU as shown in Table 4. For example, GPIOC_6 can be configured for waking up the host MCU using the following AT commands (see Figure 64):

AT+AWS SET APP_MCU_WKAEUP_PORT GPIO_UNIT_C	// GPIO_C port
AT+AWS SET APP_MCU_WKAEUP_PIN GPIO_PIN6	// GPIO_6



Figure 64. Another pin configuration for waking up host MCU

Table 4. GPIO pin configuration

Port	PIN MUX	GPIO
GPIO_UNIT_A	PIN_AMUX	GPIOA_0
		GPIOA_1
	PIN_BMUX	GPIOA_2
		GPIOA_3
	PIN_CMUX	GPIOA_4
		GPIOA_5
	PIN_DMUX	GPIOA_6
		GPIOA_7
	PIN_EMUX	GPIOA_8
		GPIOA_9
	PIN_FMUX	GPIOA_10
		GPIOA_11
GPIO_UNIT_C	PIN_UMUX	GPIOC_6
		GPIOC_7
		GPIOC_8



7.3.3 Programming Firmware Images for DA16200/DA16600

When using an EVB for the first time, the firmware must be updated to the latest version. For more details, see DA16200 DA16600 FreeRTOS Getting Started Guide, Ref. [3]. After programming the firmware image, factory reset is required to enter the AWS IoT configuration setting mode. This can be done by pushing the "Factory_RST" button for 5 seconds as shown in Figure 65 and Figure 66.



Figure 65. Factory reset button on DA16200 EVB



Figure 66. Factory reset button on DA16600 EVB



The logs from DA16200 are as follows:

```
[/bal6200]#
Factory reset ready.
Factory Reseting....
DA16200 concurrent factory reset AP mode = 1 ("AP_ONLY")....
DA16200 concurrent factory reset AP mode = 1 ("AP_ONLY")....
....
(app_set_customer_ap_configure] set AP config mode = 0
apps_reboot_ap_mode Customer configuration ...
...
default_ssid = "Dialog_DA16200" ..., ap_config_param->ssid_name
FW = 1234567890
FW = 1234567890 completed
...
apps_reboot_ap_mode IPADDR_CUSTOMER...
....
apps_reboot_ap_mode customer_dhcpd_flag == DHCPD_CUSTOMER..
.....
OK
```

The logs from DA16600 are as follows:



* - OS Type : FreeRTOS 10.4.3 * - Serial Flash : 4 MB * - SDK Version : V3.2.8.0 AWS-ATCMD Doorlock Ref. QFN GEN * - F/W Version : FRTOS-GEN01-01-f017bfdf51-006558 * - F/W Build Time : Sep 5 2023 17:17:05 * – Boot Index : 0 gpio wakeup enable 00000402 [combo] dpm_boot_type = 0 >>> UART1 : Clock=80000000, BaudRate=115200 >>> UART1 : DMA Enabled ... [combo] BLE BOOT MODE 0 [combo] BLE FW VER to transfer >>> v 6.0.14.1114.3 (id=1) at bank 1 [combo] BLE FW transfer done System Mode : Station Only (0) >>> Start DA16X Supplicant ... >>> DA16x Supp Ver2.7 - 2022 03 >>> MAC address (sta0) : d4:3d:39:40:72:16 >>> sta0 interface add OK >>> Start STA mode... by default, rf meas btcoex(1, 0, 0) >>> UART2 : Clock=80000000, BaudRate=115200 >>> UART2 : DMA Enabled ... [UART ready notification] <<< GAPM_DEVICE_READY_IND AWS IOT dev name="DA16200", len=7 IoT dev_name="DA16200", len=7 [combo] Advertising... [/DA16600] #

After the factory reset, the DA16200/DA16600 is now ready to enter the AWS IoT Configuration Settings.

7.3.4 Configure Components for Testing

The following information are required for testing the application with AWS IoT server:

Unique thing name

The information can be set in the source code for the host MCU or using the provided scripts in the downloaded package. For how to run the macro script, see Ref. [3]. The scripts are in the following location:

\DA16x00_img\script\doorlock.ttl

;In order to use this script on DA16200, the console should be prompt ;after setting the DA16200 to STA mode, SNTP client enable, and no DPM mode in easy setup through the console. ;set configurations with DA16200's console ;set features sendln "user" ;set board type sendln "SET APP_BOARD_FEATURE EVK" mpause 400 ;set your thingname ;sendln "SET APP_THINGNAME FAE-DOORLOCK-4" mpause 400 ;set broker address sendln "SET AMS_BROKER alkzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com" mpause 400

The MCU source code can be found in the following file:

\MCU\RA6M4\Src\atcmd\at_cmd.c.

#define MAX_RETRY_SEND_COUNT 10		
/* AWS features, configurations, and certification keys	*/	
const char* cmd_set_cfg[MAX_CFG_NUM] =		
{		
"\r\nAT+"PLATFORM" SET AWS_USE_FP 0\r\n",		
"\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",		
"\r\nAT+"PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\	\n",	
"\r\nAT+"PLATFORM" SET AWS_BROKER a1kzdt4nun8bnh-ats.i	iot.ap-northeast-2.	amazonaws.com\r\n",
"\r\nAT+"PLATFORM" SET APP_LPORT 1883\r\n",		
"\r\nAT+"PLATFORM" SET APP_SUBTOPIC /AppControl\r\n",		
"\r\nAT+"PLATFORM" SET APP_PUBTOPIC /DeviceControl\r\r	ı",	
"\r\nAT+"PLATFORM" CFG 0 app_door 1 2\r\n",	/* mcu sub.	str */
"\r\nAT+"PLATFORM" CFG 1 mcu_door 1 0\r\n",	/* mcu pub.	str */
"\r\nAT+"PLATFORM" CFG 2 app_window 1 2\r\n",	/* mcu sub.	str */
"\r\nAT+"PLATFORM" CFG 3 mcu_window 1 0\r\n",	/* mcu pub.	str */
"\r\nAT+"PLATFORM" CFG 4 battery 0 1\r\n",	/* shadow int *	/
"\r\nAT+"PLATFORM" CFG 5 temperature 2 1\r\n",	/* shadow float	*/



"\r\nAT+"PLATFORM"	CFG 6 doorStat 1 $1\r\n''$,	/* shadow	str */		
"\r\nAT+"PLATFORM"	CFG 7 windowStat 1 $1\r\n''$,	/* shadow	str */		
"\r\nAT+"PLATFORM"	CFG 8 app_shadow 1 $2\r\n''$,	/* mcu sub	- :	str */	
"\r\nAT+"PLATFORM"	CFG 9 mcu_shadow 1 $0\r\n"$,	/* mcu pub	- :	str */	
"\r\nAT+"PLATFORM"	SET SLEEP_MODE 3\r\n",				
"\r\nAT+"PLATFORM"	SET USE_DPM 1\r\n",				
"\r\nAT+"PLATFORM"	SET RTC_TIME 1740\r\n",				
"\r\nAT+"PLATFORM"	SET DPM_KEEP_ALIVE 30000\r\n",				
"\r\nAT+"PLATFORM"	SET USE_WAKE_UP 0\r\n",				
"\r\nAT+"PLATFORM"	SET TIM_WAKE_UP 10\r\n",				
"\r\nAT+"PLATFORM" GPIO_UNIT_C */	SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A	r\n", /*	GPIO_UN	IT_A or	
"\r\nAT+"PLATFORM" or GPIO_PIN6~GPIO_PI	SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\ N8 */	n" /*	GPIO_PIN	NO ~ GPIO_PIN1	1
1.					

57

7.3.5 Test without Host MCU

If the host MCU is not available, the AWS IoT commands can be tested with the script provided in the downloaded package.

Door lock for two-way communication:

\DA16x00 img\script\doorlock.ttl.

NOTE

The example script only supports initial value setting. To fully verify the operation of the AT commands, use the host MCU for interacting with the server and application.

7.3.6 Test with Host MCU

The e²studio is required for building source code for the host MCU and programing the images to the host MCU. Visit the Renesas website (https://www.renesas.com/us/en/software-tool/e-studio) for downloading and installing the e²studio. After installing the e²studio, complete the following steps for building and programming.

1. Import the project file to \MCU\RA6M4\.





Figure 67. e²studio project file

NOTE

When connecting to the RA6M4 MCU for the first time or changing the configuration, complete the step 2 to set up the FSP configuration.

2. To set FSP configuration of the RA6M4 MCU, select **configurations.xml**.

😨 workspace - RA6M4_DA16x00_ATcmd/configuration.xml - e* studio						
File Edit Navigate Search Project	Renesas Views Run Window Help					
🔦 🗱 🔳 🔯 Debug	🗸 💽 RA6M4_DA16x00_ATcmd.elf 💿 🗸 🏟 🗄 😭 🕶 🔚 🐚 🥸 🕶 🗞 🕶 🖬 ! 🖳 🐘 🕪 💷 📾 🕺 🐟 🕼 ! 🔛 💷 🖓 🐟 🕼 ! 🔅					
10a • 8a m • m 🗑 🖏 💩 🚳	Ø : 参 タ ▼ : 월 ▼ 初 ▼ や ヴ ゆ ▼ ウ ▼ ┏ C/C++ ☆ Debug					
Noject Explorer 🗙 🗖 🗖	🖻 at_cmd.c 🛛 🕸 [RA6M4_DA16x00_ATcmd] FSP Configuration X					
	Stacks Configuration O Generate Project Content					
> 📴 ra	Threads 🔄 New Thread 🔬 Remove 😑 HAL/Common Stacks 💿 New Stack > 🚊 Extend Stack > 🔬 Remove					
> 23 ra_gen > 29 src > 29 Debug	✓					
> 🍃 ra_cfg > 🍃 script 👹 configuration.xml 📄 R7FA6M4AF3CFB.pincfg	⊕ g_uart0 UART (r_sci_uart) ⊕ g_spi0 SPi (r_spi) ⊕ g_external_irq0 External IRQ (r_jcu) ⊕ freeRTOS Heap 4 □					
 ra_ctg.txt RA6M4_DA16x00_ATcmd.elf.jlit RA6M4_DA16x00_ATcmd.elf.lat O Developer Assistance 						
	v in Console v v v v v v v v v v v v v v v v v v v					
	Objects 💿 New Object > 🎪 Remove					
	 g_update_console_event Event Group g_start_menu_binary_semaphore Binary Semap 					
< >	Summary BSP Clocks Pins Interrupts Event Links Stacks Components					

Figure 68. FSP configuration

- 3. Use the thing name received from the FAE to test without setting up a server.
- 4. Change the thing name to the received name.

🕲 workspace - RA6_DA16x00_Interface/src/atcmd/at_cmd.c - e ^a studio								
File Edit Source Refactor Navigate Search Pr	rojec	t Renesas Vie	ws Ru	tun Renesas Al Window Help				
📑 • 🔛 🐚 🛞 • 🐔 • 🛗 🖻 🔌 D> 00		N 3. 9.1	e 🗟	5 元: 株 + 💁 - 1월 : 않 + 않 + 🕑 + 🕑 - 1월 🥒 - 1월 🗉 - 1 🍂 한 - 1월 - 4 - 4 - 1 🛃	Q	: 😭	0 C/C++	to the De
🍋 Project Explorer 🗙 📄 😫 🍟 🗖		at_cmd.c \times	🖻 at_	L_cmd.h				
V 💕 RA6_DA16x00_Interface (in RA6M4_Platfo		195		"jkzSswF07r51XgdIGn9w/xZchMB5hbgF/X++ZRGjD8ACtPhSNzkE1akxehi/oCr0\r\n"				^
> 💒 Binaries		196		<pre>"Epn3o0WC4zxe9Z2etciefC7IpJ50CBRLbf1wbWsaY71k5h+3zvDyny67G7fyUIhz\r\n"</pre>				
> 🔊 Includes		197		"ksLi4xaNmjICq44Y3ekQEe5+NauQrz4wlHrQMz2nZQ/1/I6eYs9HRCwBXbsdtTLS\r\n"				
> 😫 ra		198		"R914LtD+gdwyah61/j2V/OeBHRDJELqYzmp\r\n"				
> 😝 ra gen		200		END_CER(1F1CA(E{P\/n})				
Y CA STC		201	6	⊖ #else				100
X (a atomd		202		<pre>const char *cmd_set_aws_cfg[MAX_CFG_NUM] = {</pre>				
> P at cmd c		203		"\r\nAT+"PLATFORM" SET AWS_USE_FP 0\r\n",				
		204		"\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",				
> 🖪 at_cmd.h		205		"\r\nAT+PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\n",				
> 🗁 da16xx_peri		200		<pre>\\r\nAl+ PLAIFORM SEI AWS_BROKEK alkZotAnunsonn-ats.lot.ap-nortneast-z.amaZonaWs.com/r\n, "\\p\At."DLATEOM" SET AND LOOPT 1923\p\s". "\\p\at."DLATEOM" SET ADD SUBTORIC (AppCostnol)p\s"</pre>				
> board_cfg.h		207		("INAT PLATFORM SET APP_LFORT 1005/("), ("INIT PLATFORM SET APP_SOFORT(), AppContol("), "\chat_"Distion" SET ADD DIBTODIT (Davis control(")," "\chat_"Distion" (C.G.G.app door 1.2);)" //	*	sub i	ctn */	
> 🖻 common_init.c		209		"\r\nAT+"PLATEORM" CF6 1 mcu door 1 0\r\n", " mcu dub, str #/	me of .	300. 2	8.60. /	
> b common_init.h		210		"\r\nAT+"PLATFORM" CFG 2 app window 1 2\r\n", /* mcu sub. str */				
> common_support.c		211		"\r\nAT+"PLATFORM" CFG 3 mcu_window 1 0\r\n", /* mcu pub. str */				
> b common support.h		212		"\r\nAT+"PLATFORM" CFG 4 battery 0 1\r\n", /* shadow int */				
> Common utils h		213		"\r\nAT+"PLATFORM" CFG 5 temperature 2 1\r\n", /* shadow float */				
> Common_duish		214		"\r\nAT+"PLATFORM" CF6 6 doorStat 1 1\r\n", /* shadow str */				
> in nai_entry.c		215		"\\\AI+PLAIFONN CFG / WINDOWSTAT 1 1\\\\", Shadow Str -/				
> [c] main_thread_entry.c		210		<pre>\r\nAl+PLAIFORM CFG 9 app_shadow 1 2(r\n , / mcu sub.str */ "\r\nAl+PLAIFORM CFG 9 mcu shadow 1 2(r\n , / mcu sub.str */</pre>				
Image: Contract of the second seco		218		<pre>('()naT+*PLATFORM' CFG 9 mcd_sindow 1 0'('), / mcd publ 201 / for path 1 / platform' SFT SIFEP MODE 3 / platform' SFT SIFEP MOD</pre>	T RTC	TIME :	1740\c\n"	
> ic rxcmd_thread_entry.c		219		"\\nat+"PLATFORM" SET DPM KEEP ALIVE 30000\\n". "\\nat+"PLATFORM" SET USE WAKE UP 0\\n".	inte_		1140 (1 (1)	·
> is system_monitor_entry.c		220		"\r\nAT+"PLATFORM" SET TIM WAKE_UP 10\r\n", "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\r\n	. /* /	GPIO_	UNIT_A or	
> C txcmd_thread_entry.c		221		"\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\n", /* GPIO_PIN0 ~ GPIO_PIN11 or GPIO_PIN6~GPIO	PIN8	*/		
> c usb_console_entry.c		222 223);				

Figure 69. Thing name in MCU source code

5. To build a new project, select **project > Build Project**.

Sworkspace - RA6_DA16	x00_Interface/src/atcmd/at	_cmd.c - e ² studio		
File Edit Source Refacto	or Navigate Search Pro	ject Renesas Views Run R	enesas Al Window Help	
	- 🗟 🕒 🔪 🕪 💷	= M 3. ⊕ .e ≂ 72	🎄 = 🎭 = 🕸 🕍 = 🛍 = 🖻 = 🧭 = 🤔 🔗 = 🗟 🗉 💷 🖉 = 🖗 = 🖗 = 🏷 = 🏷 = 🖓 = 🖓 = 🖓 = 🎼 👘 🖉	C++
Project Explorer ×	🖻 😫 🍸 🕴 🗖 🚺	💽 at_cmd.c × 💽 at_cmd.	h l	
~ 😂 RA6_DA16x00_In**	rface (in DAGMA Distfo	105	"jkz5swF07r51XgdIGn9w/xZchMB5hbgF/X++ZRGjDBACtPhSNzkE1akxeh1/ocr0\r\n"	^
> 💸 Binaries	New	>	<pre>"Epn3o0WC4zxe9Z2etciefC7IpJ50CBRLbf1wbWsaY71k5h+3zvDyny67G7fyUIhz\r\n"</pre>	
> 🔊 Includes	Go Into		"ksLi4xaNmjICq44Y3ekQEe5+NauQrz4wlHrQMz2nZQ/1/I6eYs9HRCwBXbsdtTLS\r\n"	
> 🥵 ra	Open in New Window		<pre>wsi4tLorgawyanoi/jav/oeonkniozatyzmp/r/n "FND CERTFICATE/r/n":</pre>	
> 🥵 ra_gen	Show In	Alt+Shift+W >		
Y 😕 src	Show in Local Terminal	>	se	
Y 🗁 atcmd 🛛 👔	Сору	Ctrl+C	st char "cmd_set_aws_ctg[MAX_C+G_NUM] = {	
> 💽 at_cmd.c 🍵	Paste	Ctrl+V	"\r\nt+"PLATFORM"SET APP BOARD FEATURE EVK\r\n",	
> 🖻 at_cmd.h 🔰	Contraction Contractic Contr	Delete	"\r\nAT+"PLATFORM" SET APP_THINGNAME FAE-DOORLOCK-4\r\n",	
> 🗁 da16xx_peri	Source	>	"\r\nAT+"PLATFORM" SET AWS_BROKER alkzdtAnun8bnh-ats.iot.ap-northeast-2.amazonaws.com\r\n",	
> h board_cfg.h	Move		<pre>`\r\nAi+ PLAIFORM' SEI APP_LPOKI 1883/r\n', '\r\nAi+ PLAIFORM' SEI APP_SUBIDIL /AppControllr\n', '\r\nAi+"PLAFEORM' SET APP PURTOPIC (DeviceControllr\n', '\r\nAi+"PLAIFORM' SEI APP_SUBIDIL /CEE and and the set '\r\nAi+"PLAFEORM' SET APP_URITOPIC (DeviceControllr\n', '\r\nAi+"PLAIFORM' SEI APP_SUBIDIL /CEE and and the set '\r\nAi+"PLAFEORM' SET APP_LPOKI 1883/r\n', '\r\nAi+"PLAIFORM' SEI APP_SUBIDIL /AppControllr\n', '\r\n', '\r</pre>	6
> 🖻 common_ini	Rename	F2	"\r\nAT+"PLATFORM" CFG 1 mcu door 1 0\r\n", /* mcu pub. str */	
> 🗈 common_ini 📐	Import		"\r\nAT+"PLATFORM" CFG 2 app_window 1 2\r\n", /* mcu sub. str */	
> 🖻 common_su	Export		"\r\nAT+"PLATFORM" CF6 3 mcu_window 1 0\r\n", /* mcu pub. str */	
> h common_su	Renesas FSP Export	>	"In that "PLATFORM" CFG 5 temperature 2 1/r/n", /* shadow float */	
> 🖻 common_uti	Build Project		"\r\nAT+"PLATFORM" CFG 6 doorStat 1 1\r\n", /* shadow str */	
> C hal_entry.c	Clean Project		"\r\nAT+"PLATFORM" CFG 7 windowStat 1 1\r\n", /* shadow str */	
> C main_thread	Refresh	F5	<pre>\r\nAi+ PLAIFORMT CF0 & app_shadow 1 2\r\n , / mcu sub str */ ''\r\nAi+PLAIFORMT CF0 & mcu shadow 1 2\r\n , / mcu sub str */</pre>	
> c r_usb_pcdc_(Close Project		"\r\nAT+"PLATFORM" SET SLEEP_MODE 3\r\n", "\r\nAT+"PLATFORM" SET USE_DPM 1\r\n", "\r\nAT+"PLATFORM" SET RTC_TIME 1740\r	·\n",
> ic rxcmd_threa	Close Unrelated Project		"\r\nAT+"PLATFORM" SET DPM_KEEP_ALIVE 30000\r\n", "\r\nAT+"PLATFORM" SET USE_WAKE_UP 0\r\n",	
> system_mon	Build Targets	,	"\r\nai+"PLATFORM" SET IIM WAKE UP 10/r\n", "\r\nai+"PLATFORM" SET APP_MCU WKAEUP_10RT GPIO_UNIT_A\r\n", /" GPIO_UNIT_A	or =
b txcmd_threa	Index	>	CONTRACTOR STATES AND	

Figure 70. Build project

6. To set the connection to the RA6M4 MCU, select **Debug Configurations**.

workspace - RA6_DA16x00_Interface/src/atcmd/a	t_cmd.c - e ² studio			
File Edit Source Refactor Navigate Search Pr	oiect Renesas View	s Run Renesas Al Window Help		
	H N 3. 5. 8	= ::: * • • • • : > :: = : · · · · · · · · · · · · · · · ·	- (a + (a 	Q 😭 🖬 C/C++ 🎄 De
Project Explorer × E 😫 🍸 🛔 📟 🗖	at_cmd.c ×	at_cmd.r C 1 RA6_DA16x00_Interface Debug		
V SRA6 DA16x00 Interface (in RA6M4 Platfo	195	Debug As >>	(++7RGiD8ACtPhSNzkF1akxehi/oCrA\r\n"	^
> Rinaries	196	Debug Configurations	1wbWsaY71k5h+3zvDvnv67G7fvUIhz\r\n"	
) Shares	197	Organize Favorites	rQMz2nZQ/1/I6eYs9HRCwBXbsdtTLS\r\n"	
> Ap includes	198	-usteconEnvirontilitationumpiced	-zmp\r\n"	
> 🛃 ra	199	"END CERTIFICATE\r\n";		
> 🥵 ra_gen	200			
Y 🐸 src	201	e #else		
Y 🕞 atcmd	202	<pre>const char *cmd_set_aws_cfg[MAX_CFG_N</pre>	JM] = {	
) B at ends	203	"\r\nAT+"PLATFORM" SET AWS_USE_FP	0\r\n",	
> D at with	204	"\r\nAT+"PLATFORM" SET APP_BOARD_	EATURE EVK\r\n",	
2 🔤 at_cmd.n	205	"\r\nAT+"PLATFORM" SET APP_THINGN	VME FAE-DOORLOCK-4\r\n",	
> 🗁 da16xx_peri	206	"\r\nAT+"PLATFORM" SET AWS_BROKER	alkzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com\r\n",	
> h board_cfg.h	207	"\r\nAI+"PLATFORM" SET APP_LPORT	1883\r\n", "\r\nAI+"PLAIFORM" SET APP_SUBTOPIC /AppControl\r\n",	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> common init.c	208	"\r\nAI+"PLATFORM" SET APP_PUBTOP	IC /DeviceControl\r\n", "\r\nAl+"PLAIFORM" CFG 0 app_door 1 2\r\n"	, /* mcu sub. str */
> Common init h	209	(r(nAI+ PLATFORM CFG 1 mcu_door	1 O(r\n , /- mcu pub. str -/	
	210	(r\nAI+ PLATFORM CFG 2 app_wind	ow 1 2\r\n , /* mcu sub. str */	
> [c] common_support.c	211	"\n\nAT+ PLATFORM CFG 4 bottom	a laar / chadey int #/	
> h common_support.h	212	"\s\sAT+"PLATFORM" CFG 5 tomograf	ine 2 the " /* chades float */	
> A common_utils.h	213	"\c\nAT+"PLATFORM" CFG 6 doorStat	1 1\n\n" /* shadow stn */	
> c hal entry.c	215	"\r\nAT+"PLATFORM" CFG 7 windowSt	at 1 1\r\n" /* shadow str */	
> R main thread entruc	216	"\r\nAT+"PLATEORM" (EG 8 ann shad	aw 1 2\r\n", /* mcu sub, str */	
in an curead endyc	217	"\r\nAT+"PLATFORM" CEG 9 mcu shad	1 0\r\n", /* mcu pub, str */	
> [c] r_usb_pcdc_descriptor.c	218	"\r\nAT+"PLATFORM" SET SLEEP MODE	3\r\n", "\r\nAT+"PLATFORM" SET USE DPM 1\r\n", "\r\nAT+"PLATFORM"	SET RTC TIME 1740\r\n".
ic rxcmd_thread_entry.c	219	"\r\nAT+"PLATFORM" SET DPM KEEP A	IVE 30000\r\n", "\r\nAT+"PLATFORM" SET USE WAKE UP 0\r\n".	-
> is system_monitor_entry.c	220	"\r\nAT+"PLATFORM" SET TIM WAKE U	P 10\r\n", "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\	r\n", /* GPIO_UNIT_A or 💼
> is tecminate thread entry.c	221	"\r\nAT+"PLATFORM" SET APP_MCU_WK	AEUP_PIN GPIO_PIN11\r\n", /* GPIO_PIN0 ~ GPIO_PIN11 or GPIO_PIN6~G	PIO_PIN8 */
) I ush console entruc	222	};		
abl_consore_enuy.c	223			
> [C] usb_console_main.c	224	const char *cmd set arure cfg[MAX (FG	NIM] = {"\r\rAT+"PLATFORM" SET APP BOARD FEATURE EVK\r\n"	

Figure 71. Debug configurations

7. On the **Debugger** tab, change the configuration as shown in Figure 72, and then click **Apply** > **Debug**.

Bebug Configurations				- D X
Create, manage, and run configurations				to-
🖺 🕼 🐌 🔚 🗶 🖻 🏷 🗕	Name: RA6M4_DA16x00_ATcmd.elf			
type filter text	📄 Main 🎋 Debugger 🕨 Startup 🎙	Source Common		
C/C++ Application C/C++ Remote Application FASE Script	Debug hardware: J-Link ARM	Target Device: R7FA6M44	AF	
GDB Hardware Debugging	GDB Settings Connection Settings	Debug Tool Settings		
C GDB Simulator Debugging (RH850)	GDB Connection Settings			
✓	Autostart local GDB server	Host name or IP address:	localhost	
RA6M4_DA16x00_A1cmd.elf Renesas Simulator Debugging (RX, RL78)	Connect to remote GDB server	GDB port number:	61234	
		Connection timeout (s):	30 ~	
	GDB			
	GDB Command: arm-none-eabi-g	db		
	Step Mode			
	Additional GDB Server Arguments			
				^
Filter matched 9 of 12 items				Re <u>v</u> ert Apply
?				Debug Close

Figure 72. Set debug configurations

The following shows the console output of the DA16200 after a factory mode reset.





nvram read string(thingname) error

invalid APP feature...can't start APP Platform thread...check again

.. UART ready

The following shows the console output of the DA16200 when setting the AWS IoT configuration with AT commands from an MCU.

argc num = 2 argv[0]: AT+AWS argv[1]: CFG 3 mcu window 1 0 Att[3] number : 3 Att[3] name : mcu_window Att[3] data type: 1 Att[3] MQTT type: 0 argc num = 2 argv[0]: AT+AWS argv[1]: CFG 4 battery 0 1 Att[4] number : 4 Att[4] name : battery Att[4] data type: 0 Att[4] MQTT type: 1 argc num = 2 argv[0]: AT+AWS argv[1]: CFG 5 temperature 2 1 Att[5] number : 5 Att[5] name : temperature Att[5] data type: 2



Att[5] MQTT type: 1

The following shows the console output of the DA16200 after the Soft AP was configured and it is waiting to be provisioned by the mobile application.

Soft AP is Ready (d4:3d:39:10:d5:07) >>> UART1 : Clock=80000000, BaudRate=115200 >>> UART1 : DMA Enabled ... [UART ready notification] [http server task] HTTP-Server Start!! [AWS-IOT AT COMMAND] [aws shadow dpm auto start] AWS_IOT on Station Mode for "FAE-DOORLOCK-4" [pal app dpm auto start] mcu wakeup port=0, mcu wakeup pin=0x800 Root CA: O Certificate: 0 Private Key: O subscribe index=0, name=app_door subscribe index=2, name=app window newNode index=4 newNode index=5 newNode index=6 newNode index=7 subscribe index=8, name=app_shadow shadow item count = 4, (integer#=1, string#=2, float#=1) current shadowConut = 4pkey=windowStat, pdata=test current shadowConut = 3pkey=doorStat, pdata=test current shadowConut = 2pkey=temperature, pdata=16.500000 current shadowConut = 1pkey=battery, pdata=2700 AWS IOT AP Mode FAE-DOORLOCK-4 +ATPROV=STATUS 1 [Start Provisioning with TCP/TLS] .. Soft AP Mode [app provision switch client thread] Create...(status=0) [10] [app provision TCP server thread] Create ... [app provision TLS server thread] Create TLS... >>> Start Provisioning Server (TLS) ... Wait Accept (TLS) ... [app find home ap] Wi-Fi Scan request success. [app_find_home_ap:518] (0) iptime_justin / 3 / -34 / 2447 [app find home ap:518] (1) AP-101-201 / 3 / -66 / 2432 [app find home ap:518] (2) SK WiFiGIGA551A 2.4G / 3 / -78 / 2422 [app_find_home_ap:518] (3) SK_WiFiGIGA551A / 3 / -79 / 2422 [app find home ap:518] (4) SK WiFi3801 / 3 / -94 / 2412 [app_find_home_ap:518] (5) NIS-HomeAP11N / 0 / -74 / 2447 [app provision_TCP_server_thread] socket().. status=1 Wait Accept...

CFR0012

7.4 Mobile App Demo

Install the mobile application by searching for **DA16200** or **DA16600** in the Google Pay Store or the Apple App Store on the mobile devices.

7.4.1 Open Door

Publish Publish Subscri Last sub	topic : APP-DOO message : door be topic : APP-DO boribe message :	RLOCK-1/AppCo Open <3> 2022-1 OORLOCK-1/Dev opened	ontrol 0-27 04:03:4 iceControl	5 오후	
	AP	P-DOORL	OCK-1		
	AWS Serv	er	Dev	ice	
Bat Ter	tery : <mark>%</mark> nperature : <mark></mark>	°C		c	
	(
Last rep	The c	100r was (22-10-27 04:03:	Opened! 49 २३		
Acce	Â ss∕Notify	+ O Add User		رژئ Setting	

Figure 73. Opened status on application

The con	Device Shadow document contains the reported, desired, and delta values nected to AWS IoT.
	Edit
De	evice Shadow state
{	
	"state": {
	"desired": {
	"welcome": "aws-iot"
	},
	"reported": {
	"welcome": "aws-iot",
	"doorState": true,
	"openMethod": "app",
	"doorStateChange": 1,
	"doorOpenMode": 0,
	"OTAupdate": 0,
	"OTAresult": "OTA_UNKNOWN",
	"temperature": 4294967296,
	"battery": 4294967296
	}
	1

Figure 74. Opened status on AWS IoT console

- [Current Status]
 - Opened, Battery: __%, Temperature: __ °C (Real values are displayed on door lock ref. board)
 - Mobile APP (User): **Opened image button**
 - AWS (Server)
 - "doorState": true
 - o "temperature": 4294967296
 - o "battery": 4294967296

NOTE

A value of 4294967296 for the temperature or battery fields indicates the function is not available.

• DA16200 (Thing): The status of the device is displayed as shown in the red text.

INFO] [DoorLockDemo] [prvEventCallback:728]

Incoming Publish Topic Name: (Command) APP-DOORLOCK-1/AppControl matches subscribed topic.

Incoming Publish Message : doorOpen

open comm

[openControl]

[INFO]	[DoorLockDemo]	[controlDoorLoc]	k:1555] publish	ı (command r	response) O	к –	payload: '	'opened''	
DEBUG:	[aws_dpm_app_	door_work:1974]	previous MQTT	result = 0 ,	doorLock	CMD	(=1: 0-idle	e, 1-open,	2-
close,	3-auto close)								

CFR0012

[INFO] [DoorLockDemo] [aws_dpm_app_door_work:2030] publish (shadow doorlock update) OK - payload:
"{"state":{"reported":{"doorState":true,"openMethod":"app","doorStateChange":1,"doorOpenMode":0,"OTAupdat
e":0,"OTAresult":"OTA_UNKNOWN"}}"
terreturn ID = 5
last user Timer ID = 5
last doorOpenFlag state: "true"
last FOTA Stat: 0
last FOTA Url: ""

7.4.2 Close Door



Figure 75. Closed status on application

Figure 76. Closed status on AWS IoT console

- [Current Status]
 - Closed, Battery: __%, Temperature: __°C (Real values are displayed on door lock ref. board)
 - Mobile APP (User): Closed image button
 - AWS (Server)
 - "doorState": false
 - o "temperature": 4294967296
 - "battery": 4294967296

NOTE

A value of 4294967296 for the temperature or battery fields indicates the function is not available.

• DA16200 (Thing): The status of the device is displayed as shown in the red text.

[INFO] [DoorLockDemo] [prvEventCallback:728]

Incoming Publish Topic Name: (Command) APP-DOORLOCK-1/AppControl matches subscribed topic.

Incoming Publish Message : doorClose





8. OTA Update

Over the Air (OTA) is the process of updating the DA16200/DA16600 firmware image through Wi-Fi using an AWS S3 bucket.

Figure 77 shows the setting up process of the OTA update.



Figure 77. OTA update

8.1 Create S3 Bucket

For OTA update, create a new bucket in S3:

1. In the Amazon S3 console, click Create bucket.

Amazon S3 X	Amazon S3
Buckets Access Points	Account snapshot Storage lens provides visibility into storage usage and activity trends. Learn more 🖓
Object Lambda Access Points Multi-Region Access Points Batch Operations IAM Access Analyzer for S3	Buckets (1) Info Buckets are containers for data stored in S3. Learn more 2 C Copy ARN Empty Delete Create bucket
Block Public Access settings for this account	Q. Find buckets by name < 1 > ⊗ Name ▲ AWS Region ▼ Access ▼ Creation date ▼
Storage Lens Dashboards AWS Organizations settings	mytestdoorlock-log Asia Pacific (Sydney) ap- southeast-2 Bucket and objects not public October 20, 2023, 15:00:50 (UTC+09:00)

Figure 78. Create bucket for OTA update



2. Enter a Bucket name, apply the settings as shown in Figure 79–Figure 81, and click Create bucket.

Sellerat Co	nfiguration
Bucket name	
mytest-ota	
Bucket name mu	ist be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming 🕻
AWS Region	
Asia Pacific (Seoul) ap-northeast-2
Object Ow Control ownersh determines who	nership Info ip of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership can specify access to objects.
Object Ow Control ownersh Jetermines who O ACLs dis All object Access to only poli	nership Info ip of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership can specify access to objects. iabled (recommended) is in this bucket are owned by this account. this bucket and its objects is specified using dees.

Figure 79. Bucket configuration – general and object ownership









Figure 81. Bucket configuration – bucket key

3. Select the created bucket in the Buckets list.

o upload	ully created bucket "mytes d files and folders, or to cor	s t-ota" nfigure additional bucket settings, c	choose View details .		View details
mazon	S3 > Buckets				
► A St	account snapshot	o storage usage and activity trends. <u>Lear</u>	n more 🔀	View Stora	ge Lens dashboard
Buck	kets (2) Info	l in 53. Learn more 🖸	Copy ARN Emp	ty Delete	Create bucket
Q	Find buckets by name				< 1 > 🕲
٩	Find buckets by name	AWS Region	▼ Access	▽ (< 1 > 🕲
م 0	Find buckets by name Name Mytest-ota	AWS Region Asia Pacific (Seoul) ap-northeas	Access Bucket and objects no	▼ c	< 1 > Creation date ▼ Dectober 23, 2023, 16:18:57 (UTC+09:00)

Figure 82. Created buckets for OTA

4. Click the Permissions tab, and then click Edit.

This bucket must be modified for public access in the next step.

NOTE

Use public buckets for development environments only due to security concerns.

ytest	-Ota Info
Objects	Properties Permissions Metrics Management Access Points
Permissi	ons overview
Access	
Bucket and	objects not public
Block pu Public access to all your S3 you turn on E some level of more 2 Edit	blic access (bucket settings) is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that lock all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn
Block pu Public access to all your S3 you turn on B some level of more Edit Block all pu	blic access (bucket settings) is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that lock all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn buckets access

Figure 83. Edit bucket for public access

5. Clear all checkboxes, and then click Save changes.



Figure 84. Public access settings for bucket

6. To save the settings, type "confirm", and then click **Confirm**.

Lait	block public access (backet settings)	
▲	Updating the Block Public Access settings for this bucket will affect this bucket and all objects within. This may result in some objects becoming public.	
To con	firm the settings, enter confirm in the field.	
confi	irm	
		_

Figure 85. Confirm settings

7. On the Permissions tab, verify that all block options of public access are off.

Objects	Properties Permissions Metrics Management Access Points
Permissi	ons overview
Access	
Bucket and	objects not public
Block pu	blic access (bucket settings)
Public access	is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that
to all your S3	buckets and objects is bounded, and of book an public access. These sectings apply only to this bucket and its access points. Any recommends and
to all your S3 you turn on E	lock all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you requir
to all your S3 you turn on E some level of more.	lock all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you requir public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn
to all your S3 you turn on E some level of more C Edit	lock all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you requir public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn

Figure 86. Settings updated

8. Click Access Control List (ACL) > Edit and next to Everyone, select Read Bucket ACL, and then click Save changes.

NOTE	
To avoid ACLs, visit https://docs.aws.amazon.com/AmazonS3/latest/userguide/about-object-ownership.html.	



Access control list (AC Grant basic read/write permission	:L) is to other AWS accounts.	Learn more 🛛	
Grantee	Objects	Bucket ACL	
Bucket owner (your AWS account) Canonical ID: D a6eb71d66033da0db971 067ac0e717331d427a9098625 cc8c21f4e8c623db400	✓ List✓ Write	✓ Read✓ Write	
Everyone (public access) Group: D http://acs.amazonaws.co m/groups/global/AllUsers	List	Read	
Authenticated users group (anyone with an AWS account) Group: D http://acs.amazonaws.co m/groups/global/Authenticated Users	List Write	Read Write	
S3 log delivery group Group: D http://acs.amazonaws.co m/groups/s3/LogDelivery	List	Read Write	
When you grant acce access the objects in Learn more [2] I understand the Access for other AWS acco No other AWS accounts associated Add grantee	ss to the Everyone or a this bucket. effects of these chang unts clated with the resour	Authenticated users group grantees, anyone in the world can ges on my objects and buckets.	

Figure 87. Public access for everyone



9. The bucket policy must be added as shown in Figure 88 and Table 5.

"User Bucket Name" in Table 5 is the name of the S3 bucket created for an OTA update.

Bucket policy The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies Policy examples Policy generator	don't apply to objects owned by other accounts. Learn more 🔀
Bucket ARN anniawsis3:::mytest-ota Policy	
1 ▼ [] 2 "Version": "2012-10-17", 3 ▼ "Statement": [Edit statement
<pre>9 5 "sid": "", 6 "Effect": "Allew", 7 "Principal1: "", 8 " "Action": [9 "slicetObject", 10 "slicitBucketCetocation", 11 "slicitBucket" 12], 13 " "Resource": [14 "arniawsis1::mytest-ota/*", 15 "arniawsis1::mytest-ota" 16] 17 } 18] 19 ************************************</pre>	Select a statement Select an existing statement in the policy or add a new statement.
JSON Ln 19, Col 1	

Figure 88. Bucket policy editor

Table 5. Bucket policy in JSON format



] } }


8.2 Upload Image File and JSON File

- 1. Rename the image files as follows:
- RTOS Image: DA16200_FRTOS-GEN01.img
- 2. To upload the image files and JSON file for the OTA update, click Upload.

azon 53 > diasemi-ota-test3	
Objects Properties Permissions Metrics Management Access Points	
Objects (0) Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory [2] to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant permissions. Learn more [2] C C copy S3 URI C copy URL Download Open [2] Delete Actions V Create folder [1] Upload Q. Find objects by prefix V V V V V V V	them
Name ▲ Type ▼ Last modified ▼ Size ▼ Storage class	~
No objects You don't have any objects in this bucket.	

Figure 89. Upload files

- 3. Drag and drop or add files to upload.
- 4. There is one IMG file for DA16200 OTA update, and the JSON file is a path setting file for the update. The important thing is that the names of the two files for the update should be the same as in Figure 90.

mazon 53	3 > diasemi-ota-test3 > Uploa	ıd				
Inlo						
Jpius	ad					
dd the fil 3 REST AI	es and folders you want to upload to PI. Learn more 🔀	o S3. To upload a file	larger than 160	5B, use the AWS C	LI, AWS SDK	or Amazoi
	Drag and drop files and folder	rs you want to upload	l here, or choose	Add files, or Add	folders.	
Files a	and folders (2 Total, 963.1 k	(B)	Remove	Add files	Add	folder
All files a	and folders in this table will be uploaded.					
0.5						
Q FI	nd by name	_			<	1 2
✓	Name 🔺	Folder	~	Гуре	⊽ Si	ze
~	DA16200_FRTOS-GEN01.img	-	-		90	53.0 KB
~	find_path.json			polication (icon	11	
				ipplication/json		J7.0 B
_				plication/json	1	J7.0 В
				ppication/json		J7.0 В
Desti	nation			ppication/json		J7.0 B
Destin	nation			ppication/json		J7.0 B
Destina Destina s3://dia	nation tion ssemi-ota-test3			ppication/json		J7.0 B
Destina s3://dia	nation tion ssemi-ota-test3			ppication		J7.0 B
Destina Destina s3://dia	nation tion semi-ota-test3 stination details		tion	ppication/json		J7.0 B
Destina s3://dia Dessina	nation tion ssemi-ota-test3 stination details ket settings that impact new objects store	d in the specified destina	, tion.	ppication		
Destina Destina s3://dia b Des Buch	nation tion ssemi-ota-test3 stination details et settings that impact new objects store	d in the specified destina	tion.	ppincation/json		
Destina s3://dia b Des Buck	nation tion issemi-ota-test3 stination details ket settings that impact new objects store iissions	d in the specified destina	ation.	ppincarton/json		
Destina s3://dia b Des Buck	nation tion ssemi-ota-test3 stination details ket settings that impact new objects store iissions ublic access and access to other AWS acco	d in the specified destina	tion.	ppincarton/json		7.08
Destina s3://dia b Des Bud	nation tion ssemi-ota-test3 stination details tet settings that impact new objects store issions ubble access and access to other AWS acco erties	d in the specified destina	ation.	ppincarton/json		7.08
Destina Destina s3://dia b Des Bucl b Perm Grant p b Propu Specify	nation tion issemi-ota-test3 stination details let settings that impact new objects store lissions ublic access and access to other AWS acco erties	d in the specified destina unts. nd more.	, stion.	ppinearton/json		7.08
Destina s3://dia b Des Buck Grant p Specify	nation tion semi-ota-test3 stination details ket settings that impact new objects store iissions ublic access and access to other AWS acco erties storage class, encryption settings, tags, ar	d in the specified destination of the specifi	stion.	ppincarcon/json	ancel	Upload

Figure 90. Ready to upload

The JSON information for the OTA update is as follows:



"User Bucket Name" is the name of the S3 bucket created for the OTA update. The URL policy of the "Source" can be changed by AWS. 5. Click the uploaded file name to check it.

Amazon S3 > diasemi-ota-test3 > DA16200_FRTOS-GEN01.img	
DA16200_FRTOS-GEN01.img	Copy S3 URI 🗄 Download Open 🖾 Object actions 🔻
Properties Permissions Versions	
Object overview	
Owner	S3 URI
AWS Region	s3://diasemi-ota-test3/DA16200_FRTOS-GEN01.img
Asia Pacific (Seoul) ap-northeast-2	Amazon Resource Name (ARN)
Last modified	arr:aws:s3:::diasemi-ota-test3/DA16200_FRTOS-GEN01.img
June 9, 2021, 18:08:56 (UTC+09:00)	Entity tag (Etag)
Size	☐ d9cf9cb27b1bf20c7b36cb96619c7595
963.0 KB	
Туре	Object URL
img	Inttps://diasemi-ota-tests.ss.ap-nortneast-z.amazonaws.com/DA16200_FRTOS-GEN01.img
Key	
D DA16200_FRTOS-GEN01.img	

Figure 91. URL of source

6. Check if the files are uploaded correctly. You can delete and/or reupload files to the bucket on the **Actions** tab.

Amazo	n S3 > diasemi-ota-test3				
dia	semi-ota-test3				
Public	y accessible				
Ob	ects Properties Permissions Metrics Mana	agement Acces	is Points		
Ob	jects (2) cts are the fundamental entities stored in Amazon S3. You can use Amazon S	3 inventory 🔽 to get a	list of all objects in your bucket. For others to access your objects	, you'll need to explicitly grant them permissions. Learn more 🏹	
	🖞 🗇 Copy S3 URI 🗇 Copy URL 😢 Downloa	ad Open 🖄	Delete Actions Create folder	ি Upload	
Q	Find objects by prefix			<	1 > 💿
	Name 🔺	Type ⊽	Last modified		∇
	DA16200_FRTOS-GEN01.img	img	June 9, 2021, 18:08:56 (UTC+09:00)	963.0 KB Standard	

Figure 92. Uploaded files

As a result, a publicly accessible bucket is created.



Figure 93. Completed setup for OTA update



8.3 Create Job

AWS IoT Jobs is a service that allows you to define a set of remote operations that are sent to and executed on one or more devices connected to AWS IoT.

For an OTA update, go to the **IoT Core** service in AWS Management Console. OTA is the process of replacing a product with a newer version of the same product. A Job must be created and registered to perform an OTA update. It is a task to access the file uploaded to the bucket of the S3 service. If the server operator registers this Job at the desired time, the test thing proceeds with the OTA update.

1. In the AWS Management Console, go to IoT core > Manage > Remote Actions > Jobs, and click Create job. Figure 94. Create job

Jobs Jobs de or insta	(4) Info fine a set of remote operations to ser lling new applications, use job templa	nd to and run o ates to create re	n one or more de eusable jobs.	vices that	t are connected to AWS	IoT. If you ha	ve remote oper	ations that are frequently pe	rformed, such as rebooting
C	Edit Cancel	Delete	Create job						
Q F	ilter jobs			All	status values	•	All types	•	< 1 > 💿
	Name	\bigtriangledown	Туре	∇	Status		∇	Created date	•
	kj_002		Snapshot		⊘ Completed			October 26, 2022, 17:5	56:44 (UTC+0900)
	kj_001		Snapshot		⊘ Completed			October 26, 2022, 17:5	54:15 (UTC+0900)
	800p3y9ggnnr		Snapshot		O Completed			September 05, 2022, 1	4:13:19 (UTC+0900)
	9h4b3b0aztp4		Snapshot		⊘ Completed			September 05, 2022, 1	3:59:51 (UTC+0900)

Figure 94. Create job

2. Select Create custom job and click Next.

Create job Info Jobs define remote operations to send to and run on devices that are connected to AWS IoT. Create a custom job, a FreeRTOS over-the-air (OTA) update job, or a Greengrass V1 Core update job.
Job type
• Create custom job Create a job to send an executable job file to one or more devices connected to AWS IoT.
Create FreeRTOS OTA update job Send a request to acquire an executable job file from one of your S3 buckets to one or more devices connected to AWS IoT.
Create Greengrass V1 Core update job Create a snapshot job to update one or more Greengrass V1 Core devices with the latest Greengrass V1 Core or OTA agent version.
Cancel

Figure 95. Create custom job

3. In the **Name** field, enter the job name and click **Next**.

Step 1 Custom job properties	Custom job properties Info
Step 2 File configuration	Job properties
Step 3 Job configuration	Name ota_job1 Enter a unique name that contains only alphanumeric characters, hyphens, or underscores. Job names can't contain any spaces. Description - optional
	Description The description can have up to 2,028 characters.
	► Tags - optional
	Cancel

Figure 96. Enter job name

4. Select the devices to update. The thing to select is available in the list of options.

Step 1 Custom job properties	File configuration
Step 2 File configuration	Job targets info A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.
Step 3 Job configuration	Things to run this job Choose existing things APP-DOORLOCK-1 × Thing groups to run this job
	Job document - new info Job documents specify the remote action to send to and run on devices that are connected to AWS IoT. Jobs that are used often can be converted to a Job template for quicker deployment. AWS provides some public templates under job templates to help accelerate implementation.
	• From file Specify a job file located in S3. This job can be converted to a job template later allowing it to be reused.
	Job file A JSON file to upload to 53. S3 URL
	Q s3://bucket/object.json View 🖾 Browse S3
	Cancel Previous Next

Figure 97. Select thing for OTA update

5. Under Job file, click Browse S3 and select the S3 URL and click Next.

Step 1 Custom job properties	File configuration
Step 2 File configuration	Job targets info A custom job is a remote operation that is sent to and runs on one or more devices connected to AWS IoT. Job targets are the things and thing groups that represent the devices that should run this job.
Step 3 Job configuration	Things to run this job Choose existing things APP-DOORLOCK-1 X Thing groups to run this job
	Choose existing thing groups
	Job document - new infe Job documents specify the remote action to send to and run on devices that are connected to AWS IoT. Jobs that are used often can be converted to a job template for quicker deployment. AWS provides some public templates under job templates to help accelerate implementation. • From file Specify a job file located in S3. This job can be converted to a job template to reuse a job document and job configurations. You can customize the file and its configurations. You can customize the file and its configurations.
	Job file A JSON file to upload to \$3. S3 URL Q s3://diasemi-ota-test3/find_path.json X View I Browse \$3 No pre-signing URL found To use pre-signing URL, a placeholder snippet is required in the job file.
	Cancel Previous Next

Figure 98. Select JSON for OTA update

6. Under Job run type, select Snapshot and click Submit.

Step 1 Custom job properties	Job configuration Info
Step 2 File configuration	Job configuration
Step 3 Job configuration	Job run type Configure how your job will deploy to the job targets. Snapshot Your job deploys and completes execution for current devices added to thing groups in current job target list. Continuous Your job continues to deploy to devices that are added to the thing groups in the job target list.
	Additional configurations - optional Rollout configuration Job executions timeout configuration
	Job executions retry configuration - new Abort configuration Cancel Previous Submit

Figure 99. Job run type

		Contraction and the second second	100 C						
C	Edit Cancel Dele	te Create jo	b						
QI	Filter jobs		A	ll status values		All types	•	< 1 >	¢
	Name	⊽ Туре	▽	Status		▽	Created date		
	ota_job1	Snapshot		Θ In progress - Ro	ollout in pro	ogress	October 27, 2022, 13:19:	29 (UTC+0900)]
	kj_002	Snapshot		O Completed			October 26, 2022, 17:56:	44 (UTC+0900)	
	kj_001	Snapshot		⊘ Completed			October 26, 2022, 17:54:	15 (UTC+0900)	
	800p3y9ggnnr	Snapshot		⊘ Completed			September 05, 2022, 14:	13:19 (UTC+0900)	
	9h4b3b0aztp4	Snapshot		O Completed			September 05, 2022, 13:	59:51 (UTC+0900)	

Figure 100. Job being created

ota_job1 Info	C Edit S	ave as a job template Cancel Delete
Details Job executions Job document Job targets Tags		
Job details		
Job name ota_job1	Last updated October 27, 2022, 13:19:34 (UTC+0900)	Devices to update 1 thing
ARN 🗇 arn:aws:iot:ap-northeast-1:432073875051:job/ota_job1	Created October 27, 2022, 13:19:29 (UTC+0900)	Job run type SNAPSHOT
Description -	Status Ø Completed	Timeout configuration
		Execution failure

Figure 101. Successfully created job



8.4 Execute OTA Update

When a job is created successfully, the device receives the job details as follows:

```
[dpmAPPManager] DM NEED CONNECTION
DM NEED CONNECTION
[INFO] [DoorLockDemo] [aws dpm app connect:2267] Establishing MQTT session with provisioned
certificate...
recv timeout (=2000 ms) set OK (socket=0)
hostName = "alkzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com", flag to re-query (=0)
host IP from RTM = "54.178.218.11"
TCP connection OK to "alkzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com"
[INFO] [DoorLockDemo] [aws dpm app connect:2317] Sucessfully established connection with provisioned
credentials.
[Make AWS-Thing-Name]
[NVRAM] AWS Thing name : [APP-DOORLOCK-1] (len=14)
[NVRAM] [APP-DOORLOCK-1/DeviceConnect] [APP-DOORLOCK-1/AppControl] [APP-DOORLOCK-1/DeviceControl]
[INFO] [DoorLockDemo] [aws dpm app subscription:1939] subscription info: total(default:4, tried:4), OK(4)
current RTM user Timer ID = 5
current RTM temperature(str): 0.000000
current RTM battery(str): 0.000000
current RTM doorOpen state: "false"
current RTM doorOpenMode : 0
current RTM FOTAFlag: 1
current RTM FOTA url : "https://diasemi-ota-test3.s3.ap-northeast-2.amazonaws.com/"
[dpmAPPManager] DM RTC WAKEUP
DM WAKEUP TIMER (tid=5)
DEBUG:
        [aws dpm app sensor work:2104] read values from sensor if available
recv timeout (=120 ms) set OK (socket=0)
[INFO] [DoorLockDemo] [aws dpm app sensor work:2162] publish (shadow sensor update) OK -
                                                                                            payload:
"{"state":{"reported":{"doorState":false,"temperature":4294967296.000000,"battery":4294967296.000000}}}"
last temperature: Not available
last battery: Not available
Sleep mode 3: KA timer interval (=1800 sec)
DM FINISH DEVICE
recv timeout (=20 ms) set OK (socket=0)
[dpm_keepalive_timer_register] RTC interval (=1780 secs), mode (=0)
>>> Start DPM Power-Down !!!
```



- NOTE
- When a Job for an OTA update is created, you can see the URL of the S3 bucket accessed through JSON in the console. Also, the setting icon changes in the Mobile application. See Figure 102 and the console message.
- The temperature and battery value displayed as 4294967296 indicates that it is not available.



Figure 102. Successful job for OTA update in mobile app

The update is executed when you click the **Update** button on the Setting screen. The console and the Android application show the progress status during the OTA update. When the update is completed, the thing restarts, and in the Android device, the update notification disappears (Figure 103).



Figure 103. Execute OTA update in Android app

The following example shows the console message when an update is being performed.



```
save URL info & reboot for OTA
Wakeup source is 0x0
. . .
. . .
. . .
DEBUG:
        [aws ota fw update:3532] RTOS url https://diasemi-ota-test2.s3.ap-northeast-
2.amazonaws.com/DA16200 FRTOS-GEN01.img
>>> SNTP Server: pool.ntp.org (106.247.248.106)
>>> SNTP Time sync : 2022.10.26 - 08:56:58
> Server FW version : FRTOS-GEN01-01-56c232799-004457
  >> HTTP(s) Client Downloading... 100%(1202848/1202848 Bytes)
- OTA Update : <RTOS> Download - Success
DEBUG:
         [app_ota_fw_download_complete_notify:3375] RTOS download finish. (0x00)
- OTA: Renewing with new F/W
- OTA: RTOS
        > Same Version : FRTOS-GEN01-01-56c232799-004457
>>> RTOS is updated and system reboots. (New boot idx=0) !!!
DEBUG: [app_ota_fw_renew_notify:3497] Succeeded to replace with new FW.
- OTA: Reboot after 0 secs ...
Wakeup source is 0x0
[dpm init retmemory] DPM INIT CONFIGURATION(1)
```



9. Private S3 Download Demo

This section explains how to download a private S3 bucket object content using DA16200. For the demonstration, AWS Signature Version 4 (Authenticating Requests (AWS Signature Version 4) - Amazon Simple Storage Service) is used for authentication.

9.1 Sign Up for AWS Account

For the Demo, you can create a new AWS account by following steps in Section 6.1.1 or use the AWS account created for this demonstration.

Use the following credential for signing in.

Username: wifiapps0@gmail.com Password: Wifiapp@123

9.2 Create S3 Bucket

For private S3 bucket file download demo, you need to create a S3 bucket.

1. In the S3 console, select Create bucket.

Amazon S3 ×	Amazon S3 > Buckets
Buckets Access Grants	Account snapshot - updated every 24 hours All AWS Regions View Storage Lens dashboard Storage lens provides visibility into storage usage and activity trends. Learn more [2]
Access Points Object Lambda Access Points Multi-Region Access Points	General purpose buckets Directory buckets
Batch Operations IAM Access Analyzer for S3	General purpose buckets Info All AWS Regions C D Empty Delete Create bucket Buckets are containers for data stored in 53. S S S S S S
Block Public Access settings for this account	Q. Find buckets by name < 1 > ③ Name AWS Region ▼ IAM Access Analyzer Creation date ▼
 Storage Lens Dashboards Storage Lens groups AWS Organizations settings 	No buckets You don't have any buckets. Create bucket
Feature spotlight 7	

Figure 104. Create private bucket

- 2. Enter a Bucket name and select Create bucket.
- 3. In the S3 bucket, you need to upload the files you want to download. Choose **Upload** after selecting the bucket.

NOTE	
While creating buckets, all S3 buckets are private by default.	



Objects Prop	erties Permissions	Metrics Managen	nent Access Poi	nts		
Objects (0) Info					-] [
Objects are the fundam	nental entities stored in Amazon	S3. You can use Amazon S3 inve	ntory 🔀 to get a list of a	Il objects in your bucket. F	or others to access your objects,	you'll need to explicitly
grant them permission	s. Learn more 🔀					
Q Find objects by	prefix					< 1 > @
				12 - 20		

Figure 105. Upload files to bucket

- 4. Drag and drop or add files to upload.
- 5. Select Upload.

9.3 Create AWS IoT Thing and Certificate

1. To create a thing, follow the steps 1 to 5 in Section 6.1.2.1 and select **No shadow**.

Dev	VICE ShadOW Info ice Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's dow using either HTTPs or MQTT topics.
0	No shadow
0	Named shadow Create multiple shadows with different names to manage access to properties, and logically group your devices properties.
0	Unnamed shadow (classic) A thing can have only one unnamed shadow.

Figure 106. Device shadow in thing

2. To create and configure certificate to thing, follow steps in Sections 6.1.2.2, 6.1.2.3, and 6.1.2.4.

NOTE	
Names for thing and policy can be modified to intended use.	

9.4 Create Policies

You can manage access in AWS by creating policies and attaching them to IAM identities.

To create policy:

{

- 1. Go to the IAM console in dropdown and select Policies.
- 2. Select Create policy.
- 3. Copy the following JSON which is used for performing actions on S3.

```
"Version": "2012-10-17",
"Statement": {
```



```
"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": "arn:aws:s3:::<BUCKET_NAME>/*"

}
```

NOTE

<BUCKET_NAME>: Use the bucket name created previously.

4. Select Next.

5. Give policy a name.

IAM > Policies > Create policy	
Step 1 Specify permissions	Review and create Info Review the permissions, specify details, and tags.
Review and create	Policy details
	Policy name Enter a meaningful name to identify this policy.
	Maximum 128 characters. Use alphanumeric and '+=,.@' characters. Description - optional Add a short explanation for this policy.
	مر Maximum 1,000 characters. Use alphanumeric and '+=,@+_' characters.

Figure 107. Create private policy

6. Select Create.

9.5 Create and Configure IAM Role

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role and an IAM user are similar, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS.

1. Go to the IAM console and select Roles.

Identity and Access $ imes$ Management (IAM)	IAM > Roles	
Q Search IAM	Roles (5) Into An IAM role is an identity you can create that has specific permissions wit trust.	C Delete Create role
ashboard	Q search	< 1 > @
ccess management	Role name	Trusted entities Last activit
ser groups	AWSServiceRoleForAccessAnalyzer	AWS Service: access-analyzer (Servic 1 hour ago
sers	AWSServiceRoleForSupport	AWS Service: support (Service-Linker -
les	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service -

Figure 108. Create private role

2. Select Create role.

3. Select custom trust policy and copy the following trust policy (JSON) for grants the credentials provider permission to assume the role.

```
{
    "Version": "2012-10-17",
    "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": "credentials.iot.amazonaws.com"},
    "Action": "sts:AssumeRole"
    }
}
```

4. Select Next

lect trusted entity	Select trusted entity un	
y a Id permissions	Trusted entity type	
1 ne, review, and create	Allow ANS service Allow ANS services like EC2, Landods, or others to perform actions in this account.	esternal web
	SAME 20 federation Allow users holensed units SAME, 20 from a surgranter denting to genderat actions on the account. Contrast counters hole product actions on the account.	
	Custom trust policy Create a custom trust policy to evaluate attems to porture actions in this account.	
	1v (2 "westaar") "2013-30-37", 3v "thatmant") (Edit statement Remove Statement1
	1* (2 *	Edit statement Remove Statement Add actions for STS
	1* (2* (3* 1/2010) 3* 1/2010 3* 1/2010 5* 1/2010	Edit statement Bamove Statement Bamove Add actions for STS Q. Filter actions
	1 • (2 • '''''''''''''''''''''''''''''''''''	Edit statement Bamove Statement Bamove Add actions for STS Q. Filter actions All actions (bt:")

Figure 109. Trusted entity

5. In the **Filter by type** drop down, choose **Customer managed** and select the previously created policy in Section 9.4.

IAM > Roles > Create role			
Step 1 Select trusted entity	Add permissions into		
Step 2 Add permissions	Permissions policies (1/953) Info Choose one or more policies to attach to your new role.		C
Step 3 Name, review, and create	Q, Search	Filter by Type Customer managed	< 1 >
	Policy name [2]	Туре	v Description
	aws-iot-rule-DemoRule-action-1-role-Demo_IAM_Role	Customer managed	
	<u>Getobjs3</u>	Customer managed	Get s3 object
	passrolepermission	Customer managed	
	policy_aws_cli_permission	Customer managed	
	S3Getobjectpolicy	Customer managed	
	sts_assumerole_policy	Customer managed	· ·
	 Set permissions boundary - optional 		
			Cancel Previous Next

Figure 110. Adding policy to role

- 6. Also, select **AWSIotFullAccess** policy in **All type** filter.
- 7. Select Next.
- 8. Enter a name for the role and select Create role.

9.6 Create and Configure IAM User

1. In the IAM console, select **Users > Create User**.

Identity and Access × Management (IAM)	<u>iam</u> >	Users																		
Q. Search IAM	User An IAN	rs (2) Info 4 user is an identity with long-term	credentials	that is used to interact	with J	AWS in an account										С	Delete	Cr	nate (iser
Dashboard	Q	Search			_													< 1	\rightarrow	۲
Access management		User name	▲	Path	7	Group: v	Last activity	~	MFA	₹	Password age v	0	onsole last sign-in	⊽	Access key ID	⊽	Active key age	~	A	ccess key
User groups		door_lock_user		1		0	-					-			Active - AKIAQFC27LN		⊘ 15 days			
Users		s3 download user1		1		0	Q 4 days app								Active - AKIAOFC27LN		Q 7 days		6	4 days a
Roles				·		•			-	_	-	_					O / only			,
Policies																				
Identity providers																				
Account settings																				
Access reports																				
Access Analyzer																				
External access																				

Figure 111. Create private user

- 2. Enter a User name and select Next.
- 3. Select Attach policies directly.

IAM > Users > Create user		0
Step 1 Specify user details	Set permissions Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more 🛛	6
Step 2 Set permissions	Permissions options	
Step 3 Review and create	Add user to group Add user to an ending group, or create a new group, We recommend using groups to message user permanents by als function.	
	Permissions policies (1230) Create policy (2) Cr	
	Fitter by Type Q. Search Altypes < 1 2 3 4 5 6 7 - 62 5	
	Policy name [2] Type V Attached entities V	
	O O O O O O	

Figure 112. Permission setting

4. Select Create policy.

5. The IAM role that you have created must be passed to AWS IoT to create a role alias. For that, follow Section 9.4 with the given JSON.



NOTE

<IAM_ROLE_NAME>: IAM name created previously.

<your_aws_account_id>: It can be found in the top-right side of AWS console while selecting account name.

6. In the Filter by type drop down select Customer managed and select the previously created policy.

Pern	nissio e one or r	ns policies (1/1232) more policies to attach to your new user.					C Create policy 🖸
				Filter by Type			
Q, I	Search			Customer managed	•	7 matches	< 1 > (9)
	Pollo	ty name [2] ▲	Туре			Attached entities	4
	۲	aws-iot-rule-DemoRule-action-1-role-Demo_IAM_Role	Customer manag	ed		1	
	۲	GET_N_PASS_ROLE	Customer manag	ed		0	
	۲	Getobjs3	Customer manag	ed		3	
	۲	passrolepermission	Customer manag	ed		2	
	۲	policy_aws_cli_permission	Customer manag	ed		1	
	۲	S3Getobjectpolicy	Customer manag	ed		1	

Figure 113. Selecting policy for user

- 7. Also select AWSIotFullAccess policy in AWS managed filter.
- 8. Select Next.
- 9. Select Create User.

IAM Jump 1 Smale user Step 1 Secoly user detably Step 2 Secoly user detably Step 3 Secoly user detably Step 3 Review and create	Review and create Review your choices. After you create the user, you can vie	w and download the autogenerated password, if enabled.				
	User details					
	User name s3download_user_new	Console password type None	Require password reset No			
	Permissions summary			< 1.5		
	Name (2	▲ Туре	♥ Used as	Ÿ		
	GET_N_PASS_ROLE	Customer managed	Permissions policy			
	Tags - optional Tags are key value pairs you can add to anti-resources to help of No tags associated with the resource. <u>Add new tag</u> 	writhy, organizat, or wanch for resources. Choose any tags you want to associate with this case.				
				Cancel Previous Create user		

Figure 114. Review and create user

- 10. Select created user.
- 11. Select Create access key.

IAM > Users > s3download_user					
s3download_user Info					
Summary					
ARN D arn:aws:iam::509399597347:user/s3download_user	Console access Disabled	Access key 1 AKIAXNGUVAER7EGJMF4F - Active @ Never used. Created today.			
Created August 29, 2024, 14:27 (UTC+05:30)	Last console sign-in -	Access key 2 Create access key			

Figure 115. Create access key

- 12. Select **CLI**, tick confirmation and choose **Next**.
- 13. (Optional) give description tag and select Create access key.
- 14. Download .csv file for storing Access key and Secret access key.

Retrieve access keys Info				
Access key If you lose or forget your secret access key, y	cannot retrieve it. Instead, create a new access key and make the old key inactive.			
Access key	Secret access key			
AKIAXNGUVAER7YWPINP4	D Show			
Access key best practices Never store your access key in plain text, in a code repository, or in code. Disable or delete access key when no longer needed. Enable least-privilege permissions.				
For more details about managing access keys, see the best practices for managing AWS access keys.				
	Download .csv file Done			

Figure 116. Retrieve access key

15. Select Done.

9.7 Create Role Alias

Now that you have configured the IAM role, you can create a role alias with AWS IoT. You must provide the following pieces of information when creating a role alias.

1. In the IoT console select **Security** > **Role aliases**.

MQTT test client	▲ <u>ANS IOT</u> > <u>Security</u> > Role allases	
Device Location New	Role allases (1) into	C Delete Create role alias
Manage	An AWS lot role allas gives an authenticated device temporary access to AWS services that are not included in the policies attached to the device's certificates.	
All devices	Q, Find role aliases	< 1 > 🛞
Greengrass devices		-
 LPWAN devices 	□ Name	Ŷ
Software packages New	name-SJdownload1-alias	
Remote actions		
Message routing		
Retained messages		
₩ Security		
Intro		
Certificates		
Policies		
Certificate authorities		
Certificate signing New		
Role aliases		
Authorizers		
Audit		
Detect		
Fleet Hub		



2. Select Create role alias.

3. Type alias name, credential duration and select Role created in Section 9.5.

NOTE

CredentialDurationSeconds: This is an optional attribute specifying the validity (in seconds) of the security token. The minimum value is 900 seconds (15 minutes), and the maximum value is 3,600 seconds (60 minutes).

RENESAS
CFR0012

Role alias properties			
Role alias name Role_alias_name			
The name can't contain spaces.	/alid characters: A-Z, a-z, 0-9, and	_ @ = , - (hyphen)	
Role	ou can change this at any time by	editing the role alias.	
Choose an IAM role	▼	C View 🖸	
Credential duration How long the credential is valid	in seconds.		
Credential duration How long the credential is valid 3600	in seconds.		7
Tredential duration How long the credential is valid 3600 The credential duration can be a	in seconds. t short as 900 seconds and as ion	g as 43200 seconds.	
edential duration w long the credential is valid 3600	in seconds.		

Figure 118. Role alias properties

4. Select Create.

9.8 Attach Role Alias Policy to Certificate

1. Create custom policy using the given JSON and follow the steps in 6.1.2.3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iot:AssumeRoleWithCertificate",
            "Resource": "arn:aws:iot:<Region>:<your_aws_account_id>:rolealias/<name_of_role_alias>"
        }
    ]
}
```

NOTE

< name_of_role_alias>: Name of the role alias created.

<your_aws_account_id>: It can be found in the top-right side of AWS console while selecting account name. <Region>: It can be selected in AWS console.

2. Attach the above created policy to certificate by following the steps in Section 6.1.2.4.

9.9 Request Security Token

- 1. To create or describe Endpoint, you need to install AWS CLI. To install it, follow the steps in the given link. https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
- 2. After installing AWS CLI, you need to run the following command.

aws configure

- 3. Provide **aws_access_key_id** and **aws_secret_access_key** which generated in Section 9.6. Also, the region and output format (JSON).
- 4. Run the following command for getting security token after setting AWS CLI.

aws iot describe-endpoint --endpoint-type iot:CredentialProvider

5. The following **JSON** object is sample output of the describe-endpoint command. It contains the endpointAddress that you use to request a security token.

```
"endpoint Address": "your_aws_account_specific_prefix.credentials.iot.your
region.amazonaws.com"
}
```

9.10 Code Configuration

{

Use the certificate, key, bucket created during the previous process to the macros in file demo_config.h.

```
#define democonfigIOT CREDENTIAL PROVIDER ENDPOINT
"<your aws account specific prefix>.credentials.iot.us-east-1.amazonaws.com"
#define democonfigROOT CA PEM "root certificate downloaded when setting up the device
certificate in AWS Account Setup"
#define democonfigCLIENT CERTIFICATE PEM "client certificate downloaded when setting up the
device certificate in AWS IoT Account Setup"
#define democonfigCLIENT PRIVATE KEY PEM "private key downloaded when setting up the device
certificate in AWS IoT Account Setup"
#define democonfigTHING NAME
                                                  "Name of IOT Thing that you provided in STEP
1"
#define democonfigIOT CREDENTIAL PROVIDER ROLE
                                                      "Name of ROLE ALIAS that you provided in
section 6.7"
#define democonfigS3 BUCKET NAME
                                                      "Name of Bucket that contains the object
that needs to be downloaded"
#define democonfigS3 BUCKET REGION
```

"Region where Bucket is located" "Name of object that needs to be

9.11 Testing Demo

downloaded from AWS S3"

#define democonfigS3 OBJECT NAME

- 1. DA16x00 should be enabled with SNTP for using certificates and connected to AP with internet access.
- 2. After building the code, load the image in DA16x00.
- 3. The demo automatically downloads the s3 private bucket file content and displays it in the console.

```
DEBUG] [Demos] [prvSendS3HttpEmptyGet:1068] Received HTTP response from
s3privatebucketdemo.s3.us-east-1.amazonaws.com/test s3 bucket.txt...
[DEBUG] [Demos] [prvSendS3HttpEmptyGet:1070] Response Headers:
x-amz-id-2: Py3YyB+XPyTCFH6dWE1WcyjY7q3Q/WBOZRxcqaybCSIumkbVv2oZ+LrpbyIhHK2w6443WuJt/D8=
x-amz-request-id: RVWBXMZQAKJ8FRWV
Date: Thu, 22 Aug 2024 19:02:12 GMT
Last-Modified: Thu, 22 Aug 2024 17:40:55 GMT
ETaq: "2e5148afee6432823aee10a27a3a52ff"
x-amz-server-side-encryption: AES256
Accept-Ranges: bytes
Content-Range: bytes 0-1024/1025
Content-Type: text/plain
Server: AmazonS3
Content-Length: 1025
[DEBUG] [Demos] [prvSendS3HttpEmptyGet:1073] Response Body:
6612564665543818482403435750349017663609926499163346576220149801451912389185926873398365303938
8726432642995143358504569007771
5859869340249686694340283504163457022411806633040456823648322149407649291709884486624991429087
9929866424562331479470484929530
4798107198075017717708753814435626352262734959756725609267280962722018526857388403754623314994
1048425721886017397002493771038
```



5978949352294638874287215930948390792479864689759029679908713843203529304159229725861615620844 3607672462374144231313952523825 4121472243678952135750691080678438523913121266791528606569722357719234953663106981929185242016 1751071280762096700317526464632 9092876562122951842146119916941895931718937037709622303904807519784876983985859485514354675809 3458201630388955491473164903161 19029733685356457419092050823362333977133993758927393621966880365414110809808625711116204972494708604941468381375412202718800 3075727614346439528964487690991586649321220625005355040038529367337670153746836096076465791378 6708380781323834871961191069325 5294339716425075 [Sockets Disconnect:449] [INFO] [Demos] [prvHTTPDemoTask:706] Demo iteration 1 was successful. [INFO] [Demos] [prvHTTPDemoTask:724] prvHTTPDemoTask() completed successfully. Total free heap is 168608. [INFO] [Demos] [prvHTTPDemoTask:727] Demo completed successfully. [INFO] [Demos] [prvHTTPDemoTask:728] -----DEMO FINISHED-----



Appendix A Provisioning

The DA16200 supports a provisioning feature called Soft AP mode for an easy network configuration. Provisioning with the **mobile network data off** on your mobile phone and Wi-Fi turned on. When provisioning is complete, turn on your mobile data again. Figure 119 shows the workflow of the provisioning process.

Press the Factory Reset button for about 5 seconds. Start the Android application and touch the START button to find the wanted AP.





A.1 Android Application

```
System Mode : Soft AP (1)
>>> DHCP Server Started
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> Add SoftAP Inteface (softap1) ...
>>> MAC address (softap1) : d4:3d:39:11:5e:73
>>> softap1 interface add OK
>>> AP Operating Channel: 1(2412)
>>> Network Interface (wlan1) : UP
BSS Isolate Disabled
Soft AP is Ready (d4:3d:39:11:5e:73)
[ APP-IOT Doorlock ]
[ aws_shadow_dpm_auto_start]
AWS IOT on Station Mode for "APP-DOORLOCK-1"
AWS IOT AP Mode APP-DOORLOCK-1
[Start Provisioning with TCP/TLS] .. Soft AP Mode
[app_provision_switch_client_thread] Create...(status=0) [10]
[app_provision_TCP_server_thread] Create ...
[app provision TLS server thread] Create TLS...
>>> Start Provisioning Server (TLS) ...
Wait Accept (TLS) ...
> Wi-Fi Scan request success.
(0) KT_GIGA_2G_505 / 3 / -25 / 2412
(1) TP-LINK AECC / 3 / -40 / 2412
[app provision TCP server thread] socket().. status=1
Wait Accept...
```



Figure 120. Provisioning from mobile app

```
[dpmAPPManager] DM NEED CONNECTION
DM NEED CONNECTION
[INFO] [DoorLockDemo] [aws_dpm_app_connect:2267] Establishing MQTT session with provisioned
certificate...
recv timeout (=2000 ms) set OK (socket=0)
hostName = "alkzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com", flag to re-query (=0)
host IP = "52.69.14.255"
TCP connection OK to "alkzdt4nun8bnh-ats.iot.ap-northeast-1.amazonaws.com"
recv timeout (=120 ms) set OK (socket=0)
[INFO] [DoorLockDemo] [aws_dpm_app_connect:2317] Sucessfully established connection with provisioned
credentials.
[Make AWS-Thing-Name]
[NVRAM] AWS Thing name : [APP-DOORLOCK-1] (len=14)
[NVRAM] [APP-DOORLOCK-1/DeviceConnect] [APP-DOORLOCK-1/AppControl] [APP-DOORLOCK-1/DeviceControl]
[INFO] [DoorLockDemo] [aws dpm app subscription:1939] subscription info: total(default:4, tried:4), OK(4)
current RTM user Timer ID = 0
current RTM temperature(str): 0.000000
current RTM battery(str): 0.000000
current RTM doorOpen state: "false"
current RTM doorOpenMode : 0
current RTM FOTAFlag: 0
current RTM FOTA url : ""
[dpmAPPManager] DM BOOT WAKEUP
DM WAKEUP BOOT
[INFO] [DoorLockDemo] [connectionReadyInform:1598] publish (command response) OK - payload: "yes"
[closeControl]
```

CFR0012







Appendix B AT Commands for AWS IoT

B.1 Operating Modes

There are three operating modes:

- Setting Mode for features configuration.
- Provisioning Mode for network connection.
- Communication Mode for running.

B.1.1 Setting Mode

After uploading the image and rebooting, the DA16200/DA16600 enters Setting mode. In this mode, all AWS IoT settings can be configured using the SET command and a specific topic can be configured using the CFG command. For proper operation of AWS IoT, the TLS certificate keys must be set. All configuration data is stored before calling the factory reset command (Figure 122).



Figure 122. Setting mode

B.2 Provisioning Mode

In provisioning mode, the DA16200/DA16600 can be provisioned using an Android or iOS device. During provisioning, the MCU only receives a report on the provisioning status. When provisioning is complete, the DA16200/DA16600 enters Communication mode automatically after rebooting (Figure 123).





Figure 123. Provisioning mode

B.2.1 Communication Mode

The DA16200/DA16600 Communication Mode is used by the MCU to communicate (send and receive) topic values with an AWS server (Figure 124).



Figure 124. Communication mode

B.3 Configuring Topic to Publish, Subscribe, and Shadow

B.3.1 Configure Topics

- Topics are configured as shown in Table 6.
- The MCU and Mobile App should be configured based on the topics shown in Table 6.
- The MCU pushes the topics in Table 6 to the DA16200/DA16600 using AT command.
- The DA16200 facilitates the communication between the MCU and phone as shown in Figure 125.

Number	Name	Value type	CMD type	Value
0	app_door	1: String	2: Subscribe	"open"/"close"
1	mcu_door	1: String	0: Publish	"opened"/"closed"
2	battery	0: Integer	1: Shadow	Battery value (0~100)
3	temperature	2: Float	1: Shadow	Temperature value
4	doorStat	1: String	1: Shadow	"opened"/"closed"
5	windowStat	1: String	1: Shadow	"opened"/"closed"

Table 6. Configuration of topics





Figure 125. Communication between MCU and phone

B.4 AT Command List

B.4.1 Basic Set

Table 7. Basic set of MCU to DA16200/DA16600

Head	Main	Sub	Parameters	
			Set the device thing name.	
			Used to choose a device by its thing name during provisioning.	
		AWS_BROKER	Set the broker address.	
		APP_LPORT	Set the local port.	
		APP_SUBTOPIC	Set subscriber topic name, and the default is "/AppControl".	
		APP_PUBTOPIC	Set subs topic name, and the default is "/DeviceControl".	
			Set sleep mode.	
		SLEEP_MODE	1 – not connected sleep. The DA16200/DA16600 wakes up only by RTC_PWR_KEY.	
			2 – not connected sleep. The DA16200/DA16600 wakes up by RTC.	
			3 – connected sleep. The connection is retained even during DPM.	
AT . AVA/C_	Define the operation of sleep mode 3.	Define the operation of sleep mode 3.		
AT+AW3=	SEI	USE_DPM	0 – no DPM. Used during debugging.	
			1 – DPM mode.	
		RTC_TIME	Set the wake-up time for Sleep mode 2.	
		DPM KEEP ALIVE	Set the keep-alive time between the IoT device and the AP.	
			Default value is 30*1000 microseconds.	
		USE WAKE UP	Set the wake-up time for full-boot mode.	
			Default value is set to 0 (0 = unused).	
		TIM_WAKE_UP	Set the period to check a beacon frame from the AP.	
			Default value is set to10.	
			Not used command.	
		AWS_USE_FP	0 – Default value.	
			1 – Not in use.	
For example	e:			
AT+AWS=S	SET APP_	THINGNAME Assigned	ThingName	

AT+AWS=SET AWS_BROKER a1kzdt4nun8bnh-ats.iot.ap-northeast-2.amazonaws.com

B.4.2 TLS Certificate

Table 8. TLS from MCU to DA16200/DA16600

Start code	Sub code	Туре	End code
	C0,	Root CA. Self-Signed, well known. Has root certificate public key.	
\x1b	C1,	Certificate key. Has own public key. Signed by root certificate private key. Use root certificate public key to prove authenticity.	\x03
	C2,	Private key. Has own public key. Signed by certificate private key. Use certificate 1 public key to prove authenticity.	
For example: send "\x1b" over UART send "C0,BEGIN CERTI send "\x03"	FICATE\n" '	'MIIDQTCCAimgAwIBAgITBmyfz5m/jAo over UART	

B.4.3 PIN MUX

Table 9. PIN MUX from MCU to DA16200/DA16600

Head	Main	Sub			Parameters
		NV_PIN_AMUX	AMUX_UART1d	4	/* UART1(RXD, TXD) */
			AMUX_GPIO	9	/* GPIOA [1:0] */
			BMUX_UART1d	4	/* UART1(RXD, TXD) */
			BMUX_GPIO	8	/* GPIOA [3:2] */
			CMUX_UART1d	6	/* UART1(RXD, TXD) */
			CMUX_GPIO	8	/* GPIOA [5:4] */
		NV_PIN_DMUX	DMUX_UART1d	4	/* UART1(RXD, TXD) */
AT+AWS=	SET		DMUX_GPIO	8	/* GPIOA [7:6] */
		NV_PIN_EMUX	EMUX_GPIO	8	/* GPIOA [9:8] */
		NV_PIN_FMUX	FMUX_GPIO	6	/* GPIOA [11:10] */
		NV_PIN_UMUX	UMUX_GPIO	2	/* GPIOC [8:6] */
		APP_MCU_WKAEUP_PORT	GPIO_UNIT_A	0	
			GPIO_UNIT_C	2	/*Support only GPIO 6,7,8 */
		APP_MCU_WKAEUP_PIN	GPIO_PIN0 ~	GPIO_	PIN11
		UART_CFG	[baud-rate]		
Note: Default	t nin m	ux is BMUX	•		

Note: Default pin mux is BMUX

For example: use GPIOA2 and GPIOA3 for UART1, and GPIOA9 for MCU wakeup

AT+AWS=SET NV_PIN_BMUX BMUX_UART1d

AT+AWS=SET NV_PIN_EMUX EMUX_GPIO

AT+AWS=SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A

AT+AWS=SET APP_MCU_WKAEUP_PIN GPIO_PIN9

B.4.4 Configure Data as Topics

Table 10. Configuration data from MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AWS=	CFG	[number] [name] [value-type] [MQTT-type]	 number: Index to identify the saved topic. Increase by 1 when setting a new topic. Max value is 10 (total supported topics is 10). name: String specifying the topic name. value-type 0 – Integer type. 1 – String type. 2 – Float type. MQTT-type 0 – Publish: The prompt command is used to send a value from the MCU to the phone. For example, door state = true/false. 1 – Shadow: The value is sent to the device twin and is updated on the phone the next time it is connected. 2 – Subscribe: The prompt command is used to send a value from the phone to the MCU. For example, door open command.
For example:	G 0 door	Stat 1 1	
AT+AWS=CF	G 1 batte	ry 2 1	
AT+AWS=CF	G 2 door <u>.</u>	_open 0 2	

B.4.5 Command – MCU to DA16200/DA16600

Table 11. Command of MCU to DA16200/DA16600

Head	Main	Sub	Description
		FACTORY_RESET	Reset the AWS IoT configuration to the factory default. All values stored in NVRAM are cleared.
			Use the "SET" and "CFG" commands to set the AWS IoT configuration.
			Switch to AP mode keeping the values set in NVRAM.
		RESET_TO_AF	The previous values in NVRAM are kept.
		GET_STATUS	Get the current AWS IoT status.
AT+AWS=	CMD		The MCU can read the current status from the DA16200/DA16600 at any time.
		RESTART	Reboot the device keeping the current mode and status.
		MCU_DATA	Used by the MCU to set a CFG parameter in the DA16200/DA16600.The value must be the same format as defined by the CFG setting. Parameters:
			[number] [name] [value]
For example:			

AT+AZU=CMD FACTORY_RESET

AT+AZU=CMD MCU_DATA 1 mcu_door opened

B.4.6 Command – DA16200/DA16600 to MCU

Table 12. Command of DA16200/DA16600 to MCU

Head	Main	Parameters	Description
+AWSIOT	SERVER_DATA	[number] [name] [value]	Used by the DA16200/DA16600 to set a CFG parameter in the MCU. The value must be the same format as defined by the CFG setting.
+AWSIOT	CMD_TO_MCU	update	Used by the DA16200/DA16600 to request the status of devices such as sensors, batteries, and doors from the MCU. The DA16200/DA16600 maintains the values obtained from the MCU and forwards them when requested by an external phone app or by an MQTT ping-pong wake-up event.
For example +AWSIOT S +AWSIOT C	: ERVER_DATA 0 doc MD_TO_MCU updat	or_control open e	

DA16200/DA16600 Status - DA16200/DA16600 to MCU **B.4.7**

Table 13. Status from DA16200/DA16600 to MCU

Status	Value	Parameters	
IDLE	-1	Initial state of AWS-IoT application.	
		Sent when a system error occurs.	
		For example, network connection failure.	
Done factory reset	0	Sent after completes factory reset by "CMD FACTORY_RESET".	
Boot Ready	Ready 1 Sent when entering AWS-IoT application mode.		
Need configuration	5	Sent if there is no setting.	
		MCU should set and configure with the SET and CFG command.	
Start AP mode	10	Sent when being started to AP mode.	
		Need to process provisioning with Phone.	
Network OK	15	Sent when it is OK to connect AP without problem.	
Network fail	16	Sent when it fails to connect AP with any problem.	
		Normally, it happens during provisioning failure by the wrong SSID or PW.	
		Need to go to AP mode by MCU send "RESET_TO_AP" command.	
Start STA	20	Not defined yet.	
Done STA	25	Sent when entering Sleep mode for DPM.	
MCUOTA	30	Sent when MCU OTA starts processing.	
For example:			
+AWSIOT STATUS 15			



Appendix C Troubleshooting

C.1 Operational Issue

When UI buttons are not visible or not showing up properly while using the mobile app, try to uninstall and install the app again. The first time running the mobile app after reinstalling it, make sure that the app can access the location of the device as described in Test Provisioning on Android/iPhone Sections of Ref. [4].



10. Revision History

Revision	Date	Description
1.7	Oct 7, 2024	 Added Section 3, 4, 5.
		 Added URL link in Section 2.
		 Added Section 9.
1.6	July 22, 2024	 Modified Note to provide customer with thing name for testing instead of providing the AWS login credentials.
		Added Section 4.1.
		 Editorial changes.
1.5	Jan 26, 2024	Added Troubleshooting Section.
1.4	Nov 30, 2023	Merged documents:
		UM-WI-016 DA16200 Door Lock Application Using AWS IoT.
		UM-WI-017 DA16200 AWS IoT Server Setup.
		UM-WI-038 DA16200 DA16600 Getting Started with AWS IoT Using AT
		Commands.
1.3	Aug 18, 2023	Changed IDE to e2studio.
		 Editorial update.
1.2	Dec 1, 2022	Edited as direct link of documents.
1.1	Nov 4, 2022	Modify hyperlink of the documents.
1.0	Oct 13, 2022	Initial version.



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.1.01 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 <u>www.renesas.com/contact-us/</u>.