User Manual

DA16200 DA16600 Getting Started with Azure® IoT

UM-WI-058

Abstract

The focus of this document is to introduce a door lock example for Azure® IoT. This is a reference design which includes the Azure® IoT Application. This reference design provides the user an experience using the Azure® IoT Android/iOS Application.

Contents

Abstract **1**

Contents **2**

Figures **4**

1 Terms and Definitions **6**

2 References **6**

3 Azure IoT Hub **7**

3.1 Azure Services 7

3.2 Azure IoT Hub Setup 7

3.2.1 Create Azure Account 7

3.2.2 Create Resource Group 8

3.2.3 Create IoT Hub 11

3.2.4 Register Devices 15

3.3 Configure Azure IoT Hub 17

3.4 Verify Connection Between Azure IoT Hub and DA16200/DA16600 20

4 Door Lock Reference Application **25**

4.1 Reference Application in DA16200/DA16600 25

4.1.1 Open Door 26

4.1.2 Close Door 28

4.2 Reference Application in Host MCU 30

4.2.1 Download Package for Door Lock Reference Application in Host MCU 30

4.2.2 Hardware Connections between DA16200/DA16600 and Host MCU 31

4.2.3 Programming Firmware Images for DA16200/DA16600 35

4.2.4 Configure Components for Testing 36

4.2.5 Test without Host MCU 38

4.2.6 Test with Host MCU 38

4.3 Mobile App Demo 44

4.3.1 Open Door 45

4.3.2 Close Door 48

5 OTA Update **51**

5.1 Create Storage Account 51

5.2 Create Container 53

5.3 Upload Firmware Image File 55

5.4 Add OTA URL to Device Twin 58

5.5 OTA Update of MCU Firmware 63

Appendix A Azure IoT Hub Device Provisioning Service **65**

A.1 Create Azure IoT Hub 65

A.2 Manage Enrollments 69

A.3 Configure Registration ID Parameters in DA16200/DA16600 SDK 70

Appendix B Azure IoT AT Command List **73**

B.1 Operating Modes 73

B.1.1 Setting Mode 73

B.1.2 Provisioning Mode 73

B.1.3 Communication Mode 74

 © 2024 Renesas Electronics

B.2 How to Report Topic and Update Device Twin 74
B.2.1 Configure Device Twin Topic 74
B.3 AT Command List 75
B.3.1 Basic Set 75
B.3.2 TLS Certificate 76
B.3.3 PIN MUX 76
B.3.4 Configure Data as Topics 77
B.3.5 Command – MCU to DA16200/DA16600 78
B.3.6 Command – DA16200/DA16600 to MCU 78
B.3.7 DA16200/DA16600 Status – DA16200/DA16600 to MCU 78
Appendix C Troubleshooting 80
C.1 Operational Issue 80
Revision History 81

 © 2024 Renesas Electronics

Figures

Figure 1: Azure Services 7
Figure 2: Azure IoT Service Configuration 7
Figure 3: Create a Resource 8
Figure 4: Resource Group 8
Figure 5: New Resource Group 9
Figure 6: Adding Data for Resource Group 9
Figure 7: Create Resource Group 10
Figure 8: Go to Resource Group 10
Figure 9: Created Resource Group 11
Figure 10: Create Resources 11
Figure 11: Select IoT Hub 12
Figure 12: Create IoT Hub 12
Figure 13: Next Networking 13
Figure 14: Next Management 13
Figure 15: IoT Hub Management 14
Figure 16: Review IoT Hub Configuration 14
Figure 17: Completed IoT Hub Deployment 15
Figure 18: Generated IoT Hub Web Page 15
Figure 19: Add Devices 16
Figure 20: Add Devices (Continued) 16
Figure 21: Save Devices 17
Figure 22: Device Registered in Azure IoT Server 17
Figure 23: Device ID Registered in Azure IoT Hub 18
Figure 24: Primary Key with Device ID 18
Figure 25: Hostname in Azure IoT Hub 19
Figure 26: Connection String for Azure IoT Hub 19
Figure 27: Azure IoT Hub 22
Figure 28: Select Device ID 23
Figure 29: Device Twin in Azure IoT Hub 23
Figure 30: Items in Azure Device Twin 24
Figure 31: Structure of Azure IoT 25
Figure 32: Open Azure IoT on Mobile Device 26
Figure 33: Azure IoT on Mobile Device 26
Figure 34: Message Flows of Opening Door 27
Figure 35: Opening Door on Mobile App 27
Figure 36: Device Twin When Door Is Open 28
Figure 37: Message Flows of Closing Door 28
Figure 38: Closing Door on Mobile App 29
Figure 39: Device Twin When Door Is Closed 29
Figure 40: Azure IoT Using Firmware Images for AT Commands and Host MCU 30
Figure 41: Hardware Configuration 31
Figure 42: Default UART HW Connection 32
Figure 43: Sample Example of UART1 Connection 33
Figure 44: HW Connection for Waking up DA16200/DA16600 33
Figure 45: Default Pin Configuration for Waking Up Host MCU 34
Figure 46: Another Pin Configuration for Waking up Host MCU 34
Figure 47: Factory Reset Button on DA16200 EVB 35
Figure 48: Factory Reset Button on DA16600 EVB 35
Figure 49: e²studio Project File 39
Figure 50: FSP Configuration 39
Figure 51: Thingname/Primarykey/Hostname/IotHub Connection String 40
Figure 52: Build Project 40
Figure 53: Debug Configurations 41
Figure 54: Set Debug Configurations 41
Figure 55: Message Flow of Opening Door 45
Figure 56: Operation of Opening Door in App 45
Figure 57: Device Twin When Door is Opened 46

© 2024 Renesas Electronics

Figure 58: Message Flows of Closing Door 48
Figure 59: Operation of Closing Door in App 48
Figure 60: Device Twin When Door is Closed 49
Figure 61: Storage Accounts 51
Figure 62: Create Storage Account 51
Figure 63: Configure Storage Account 52
Figure 64: Storage Account Validation Done 52
Figure 65: Storage Account Deployment Done 53
Figure 66: Created Storage Account 53
Figure 67: Containers in Storage Account 54
Figure 68: Add Container 54
Figure 69: Container Details 55
Figure 70: Created Container 55
Figure 71: Container for Uploading Images 56
Figure 72: Upload Image 56
Figure 73: Navigate to Image 57
Figure 74: Uploaded Image 57
Figure 75: Copy Container URL 58
Figure 76: Add Device Configuration 58
Figure 77: Name and Label Section 59
Figure 78: Create Configuration 60
Figure 79: Target Devices Section 61
Figure 80: Review + Create Section 61
Figure 81: Created Device Twin Configuration 62
Figure 82: OTA Update Operation 62
Figure 83: Message Flows of OTA Update 63
Figure 84: OTA Process in DA16200/DA16600 63
Figure 85: Block Diagram of MCU OTA 64
Figure 86: Azure IoT Hub Device Provisioning Service 65
Figure 87: Create DPS 65
Figure 88: Configure Resource Group, Name, and Region 66
Figure 89: Create Button 66
Figure 90: Go to Resource 67
Figure 91: Linked IoT Hubs 67
Figure 92: Add Link to Azure IoT Hub 68
Figure 93: Configure Link 68
Figure 94: IoT Hub Linked to DPS 69
Figure 95: Manage Enrollments 69
Figure 96: Add Enrollment 70
Figure 97: Individual Enrollments 70
Figure 98: DPS Registration ID 71
Figure 99: Primary Key of DPS Registration ID 71
Figure 100: ID Scope of DPS 72
Figure 101: Registered DPS Device 72
Figure 102: Setting Mode 73
Figure 103: Provisioning Mode 74
Figure 104: Communication Mode 74
Figure 105: Communication from MCU to Phone 75

 © 2024 Renesas Electronics

1 Terms and Definitions

AP	Access Point
ADB	Android Debug Bridge
API	Application Programming Interface
Azure	Microsoft Web Services
CFG	Configure
CMD	Command
DPM	Dynamic Power Management
DTIM	Delivery Traffic Indication Map
EVB	Evaluation Board
IoT	Internet of Things
MCU	Micro-Controller Unit
OTA	Over-The-Air
RTC	Real-Time Clock
SDK	Software Development Kit
TIM	Traffic Indication Map
TSL	Transport Layer Security
NVRAM	Non-Volatile Random Access Memory
DPS	Device Provisioning Service

2 References

[1] UM-WI-056, DA16200 DA16600 FreeRTOS Getting Started Guide, User Manual, Renesas Electronics

[2] UM-WI-042, DA16200 DA16600 Provisioning Mobile App, User Manual, Renesas Electronics

© 2024 Renesas Electronics

3 Azure IoT Hub

A Microsoft Azure account is required to run the door lock reference application. This section provides how to configure Azure IoT Hub for communicating with the DA16200/DA16600 IoT device.

3.1 Azure Services

The Azure Portal provides various services including IoT as shown in Figure 1.

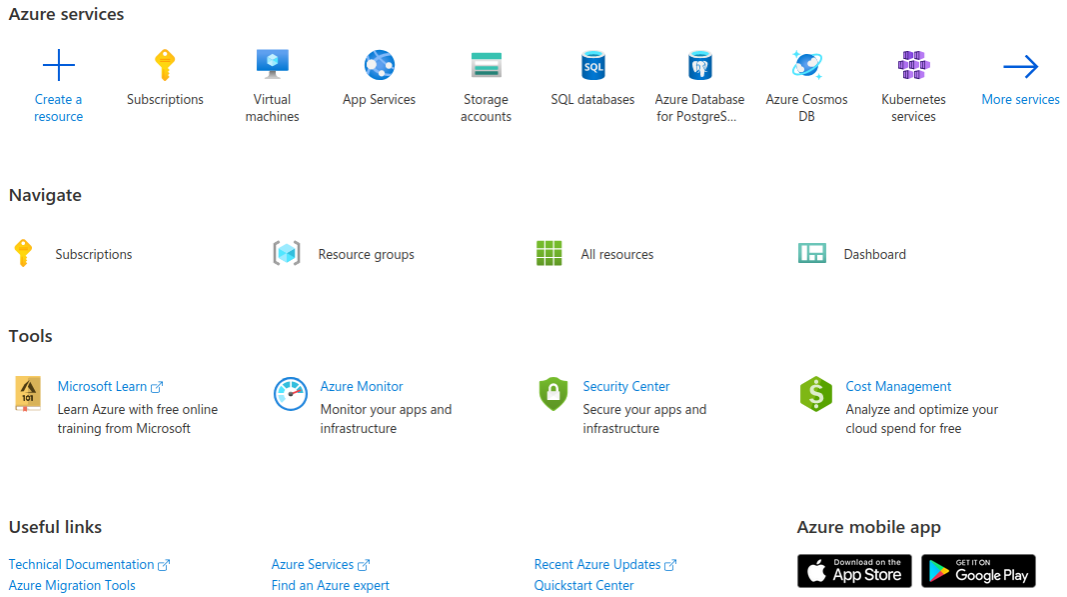

Figure 1: Azure Services

3.2 Azure IoT Hub Setup

To connect a device to the Azure IoT server, the following components are required. This section describes how to set up requirements before using Azure IoT Hub.

- Azure account
- Resource groups
- IoT Hub
- Devices

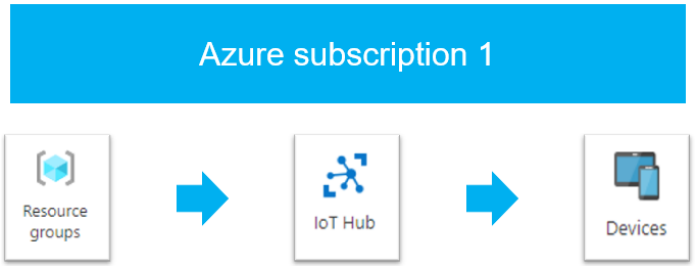

Figure 2: Azure IoT Service Configuration

3.2.1 Create Azure Account

To create an Azure account, follow the steps below:

© 2024 Renesas Electronics

1. Go to the https://azure.microsoft.com.
2. Create an Azure account and sign in.
3. Go to Microsoft Azure Portal to use Azure services (https://azure.microsoft.com/en-us/features/azure-portal).

3.2.2 Create Resource Group

To register an IoT Hub resource, a resource group should be created.

1. In Azure services, click **Create a resource**.

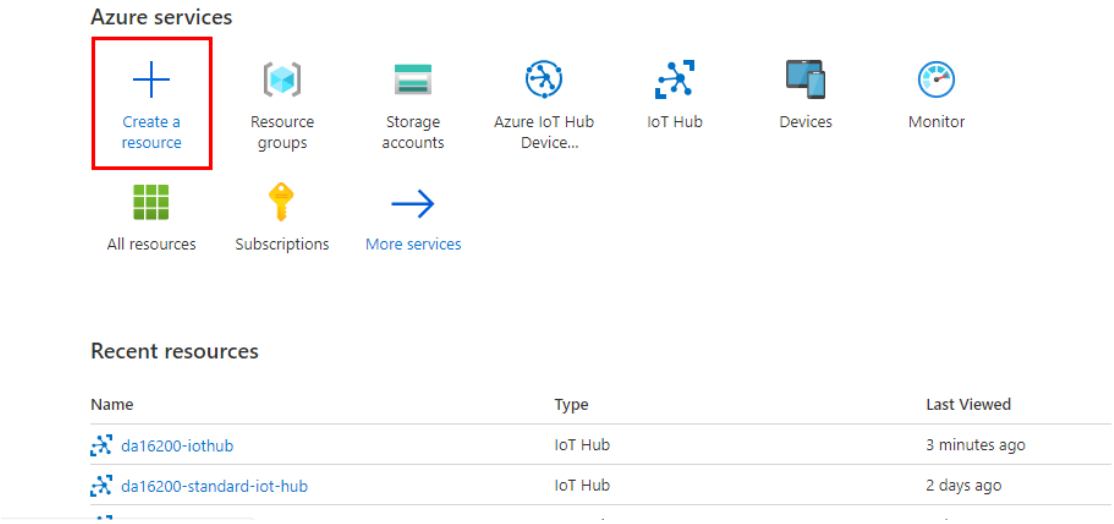

Figure 3: Create a Resource

2. Search and click **Resource group**.

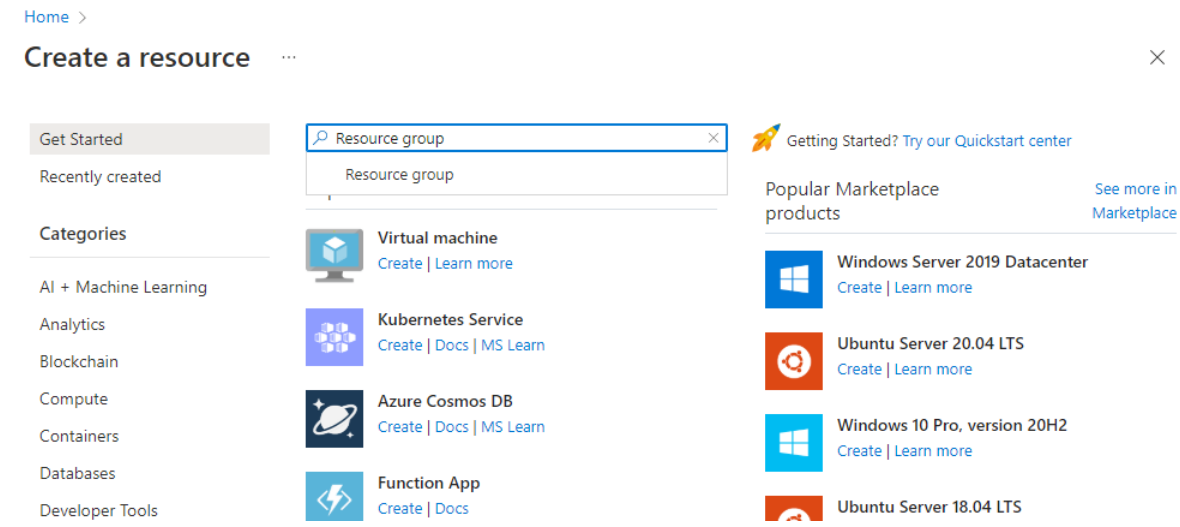

Figure 4: Resource Group

3. Click **Create** to create a new resource group.

© 2024 Renesas Electronics

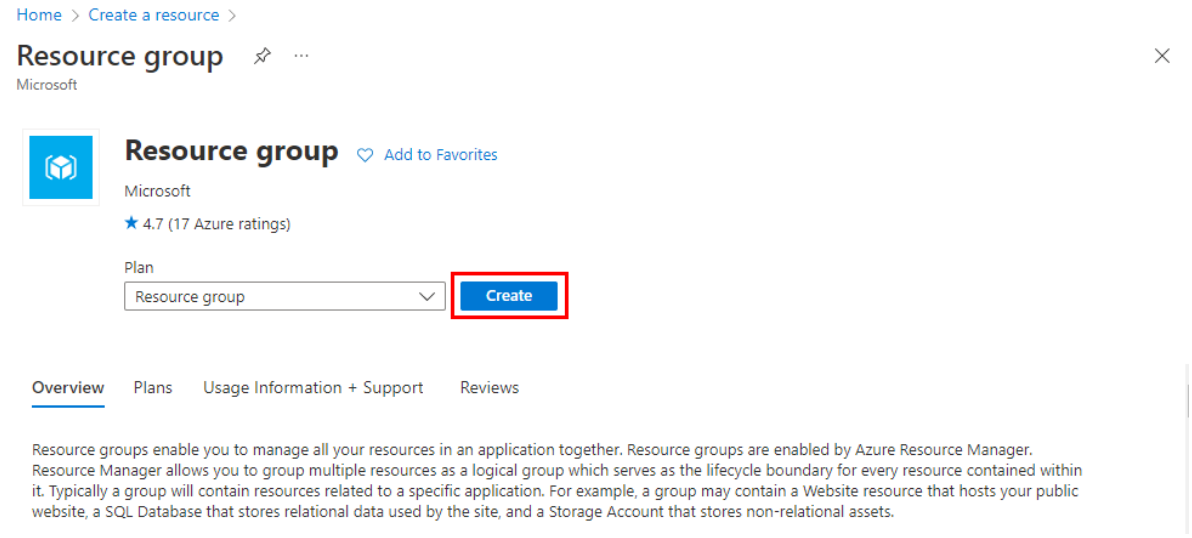

Figure 5: New Resource Group

4. Add a resource group name and select a region, and then click **Review + create** to accept the configuration.

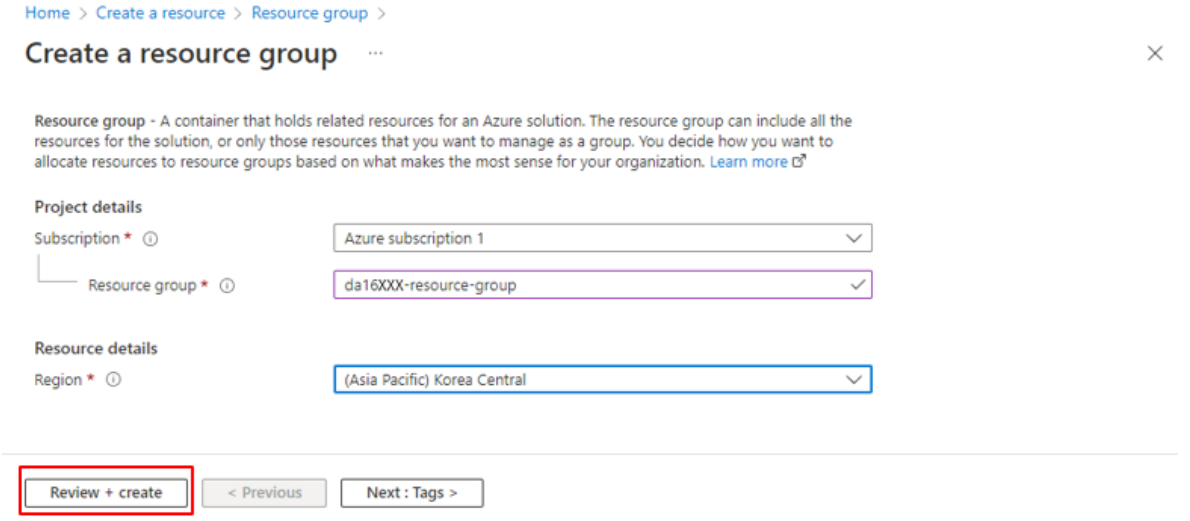

Figure 6: Adding Data for Resource Group

5. Click **Create**.

© 2024 Renesas Electronics

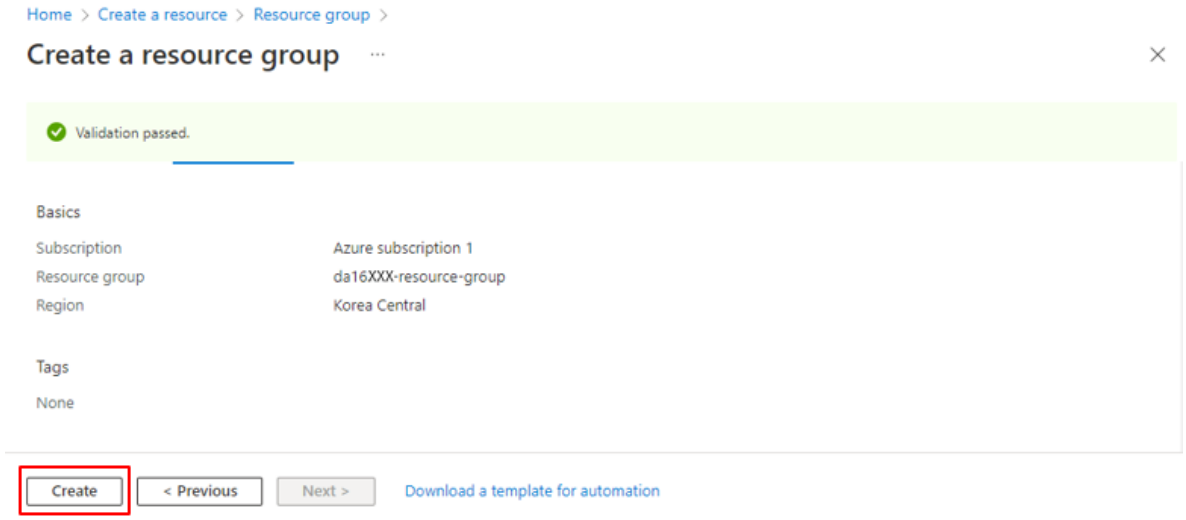

Figure 7: Create Resource Group

6. Click **Go to resource group**.

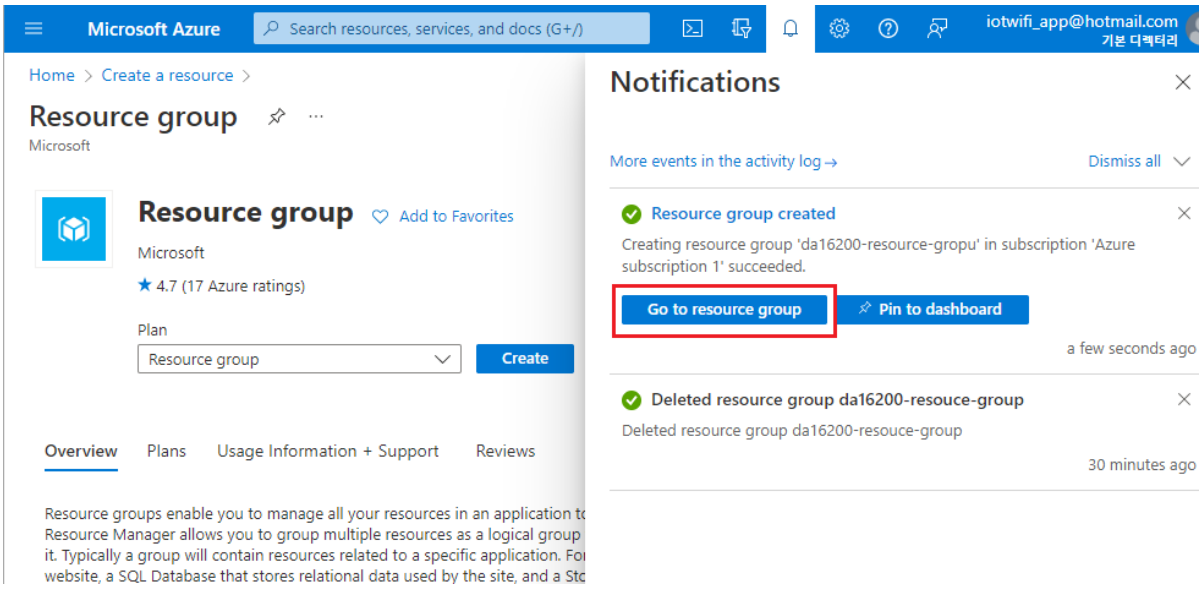

Figure 8: Go to Resource Group

The newly created resource group is displayed.

© 2024 Renesas Electronics

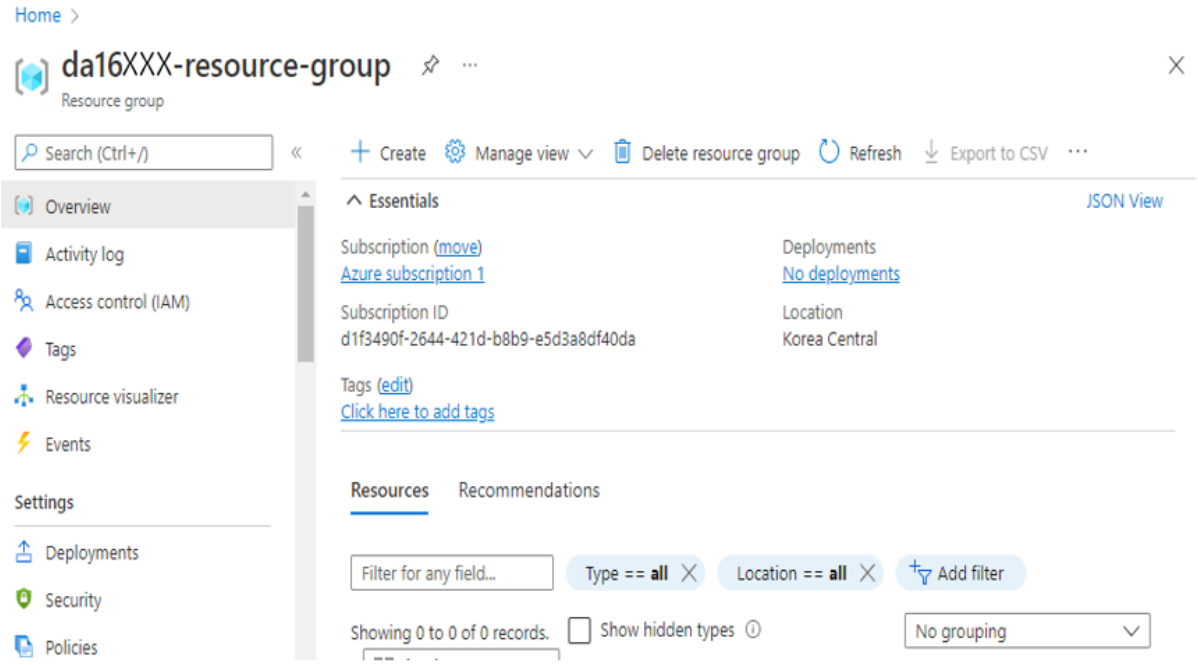

Figure 9: Created Resource Group

3.2.3 Create IoT Hub

Azure IoT Hub is a managed service that acts as a central message hub for communication between IoT applications and IoT devices that are connected to the DA16200/DA16600. IoT Hub should be created before creating IoT devices.

To create IoT Hub:

1. In the created resource group, click **Create resources**.

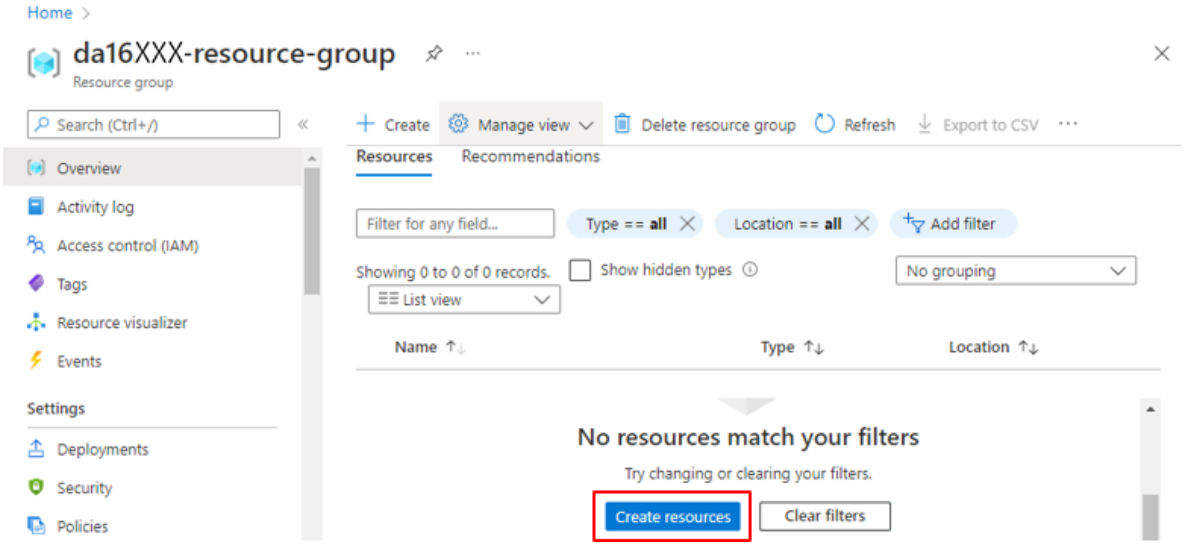

Figure 10: Create Resources

2. In **Create a resource** menu, search and click **IoT Hub**.

© 2024 Renesas Electronics

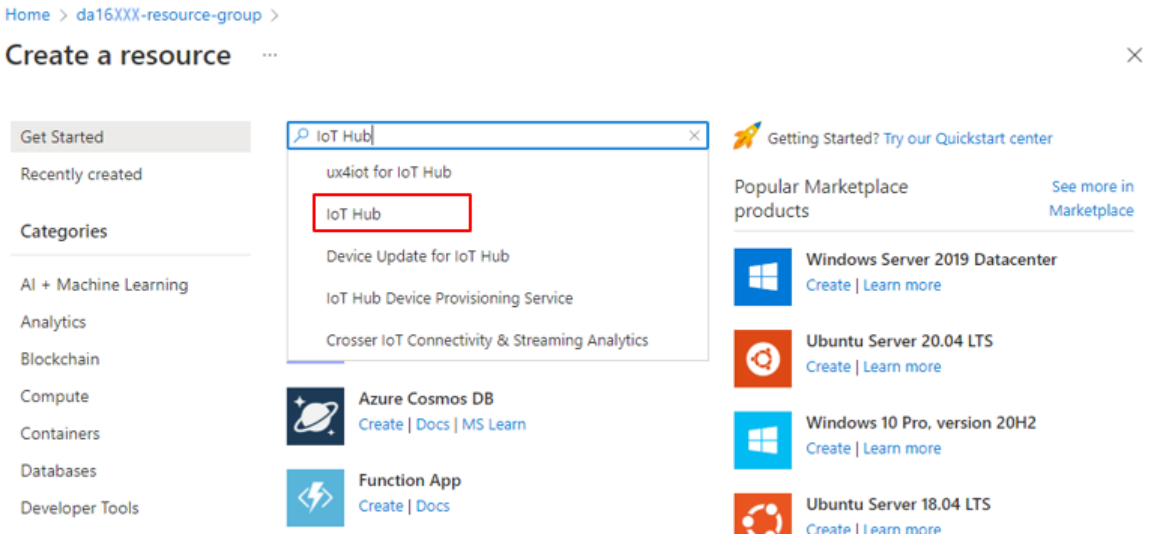

Figure 11: Select IoT Hub

3. In **IoT Hub** menu, click **Create**.

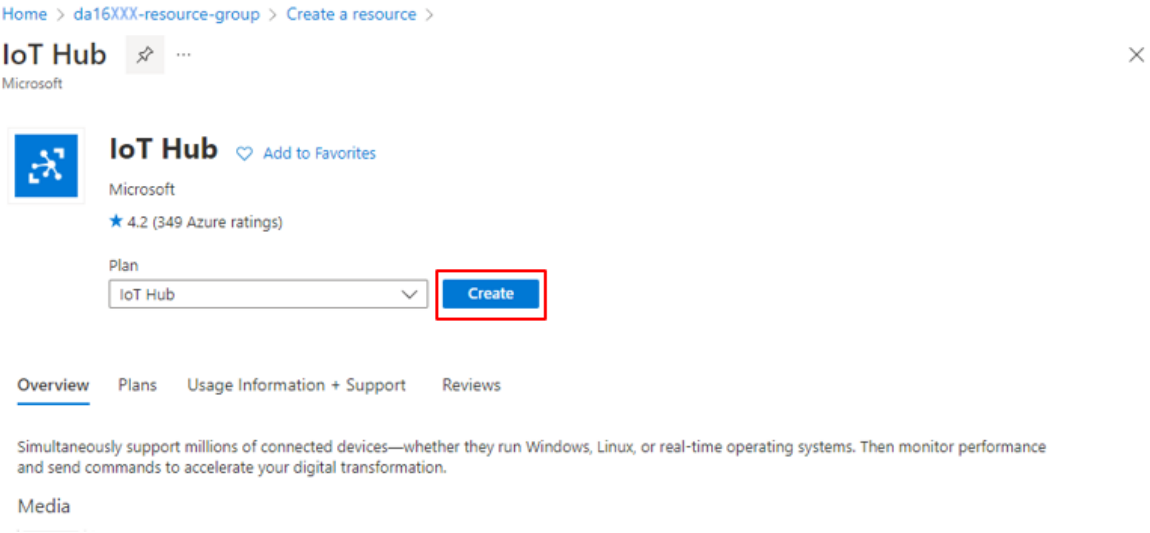

Figure 12: Create IoT Hub

4. In the **Basics** menu, add the IoT hub name and select a region, and then click **Next: Networking**.

© 2024 Renesas Electronics

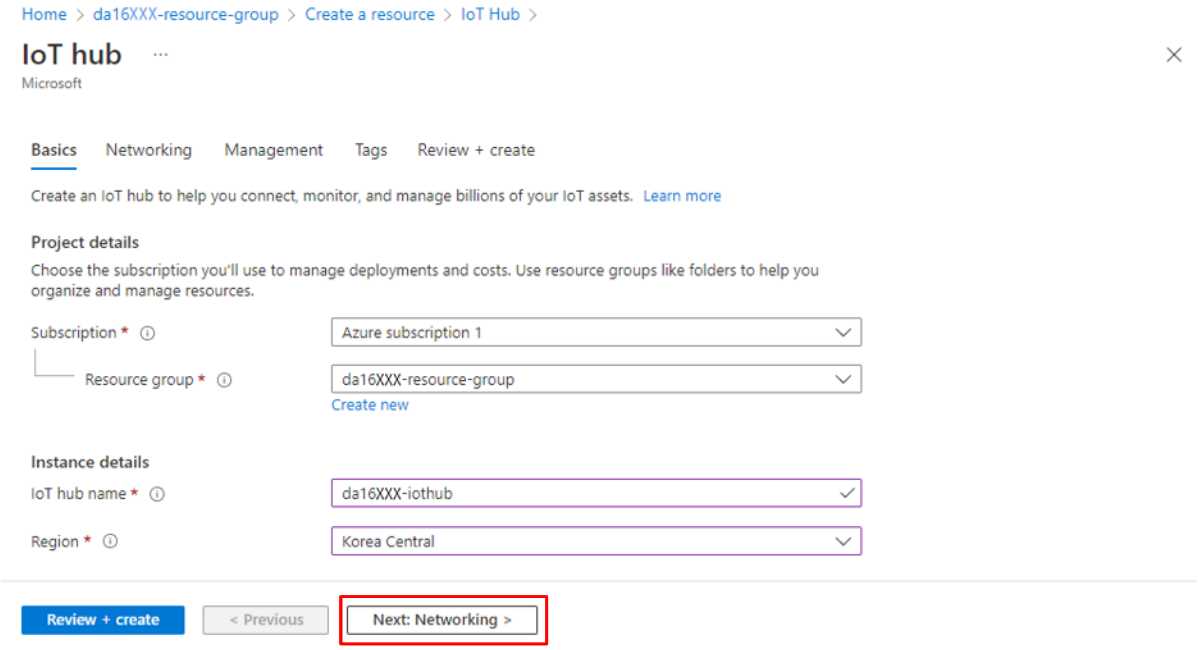

Figure 13: Next Networking

5. In **IoT Hub Networking** menu, click **Next: Management**.

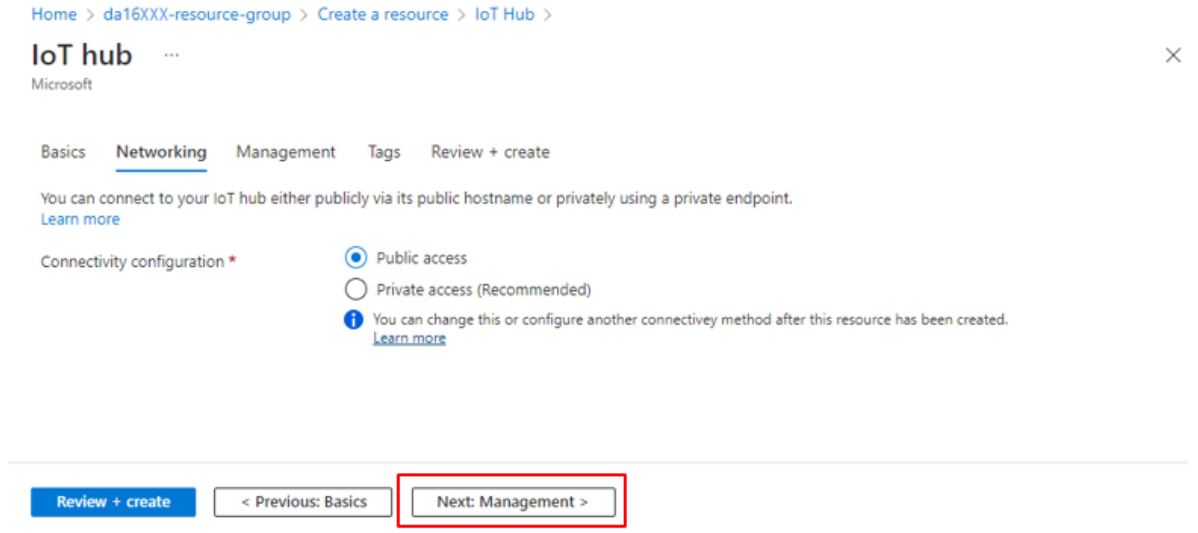

Figure 14: Next Management

6. In the **IoT Hub Management** menu, select **Pricing and scale tier**, and then click **Review+create**.

© 2024 Renesas Electronics

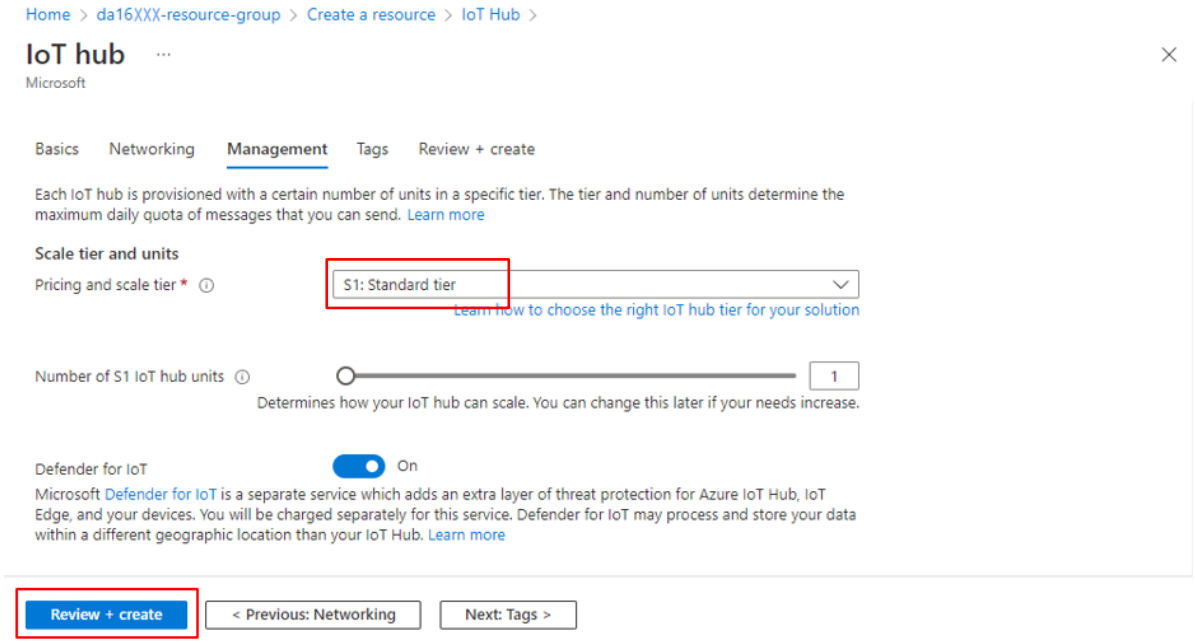

Figure 15: IoT Hub Management

7. Review the items displayed in the **Review + Create** menu, and click **Create**.

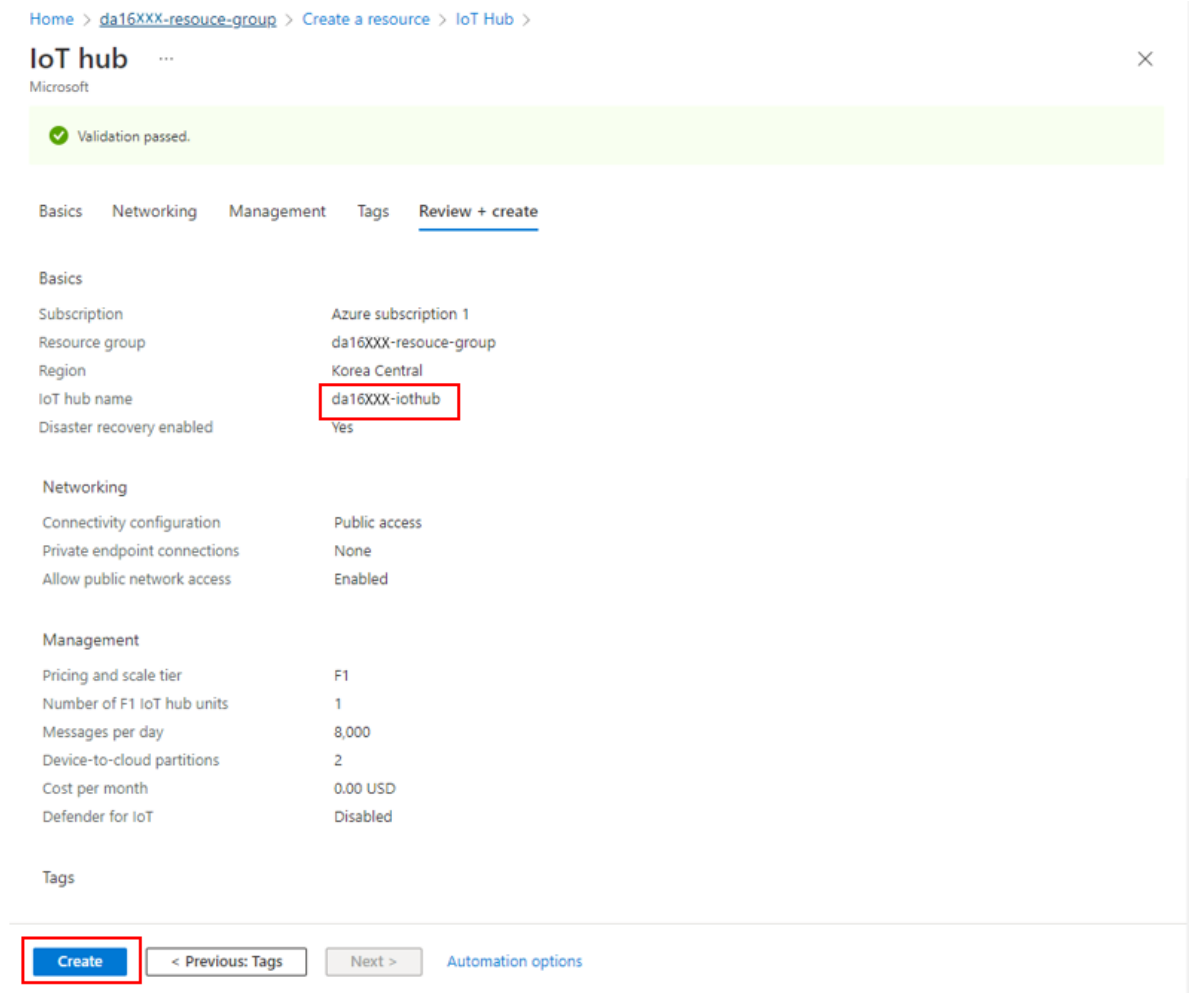

Figure 16: Review IoT Hub Configuration

© 2024 Renesas Electronics

8. After IoT hub deployment is completed, click **Go to resource**.

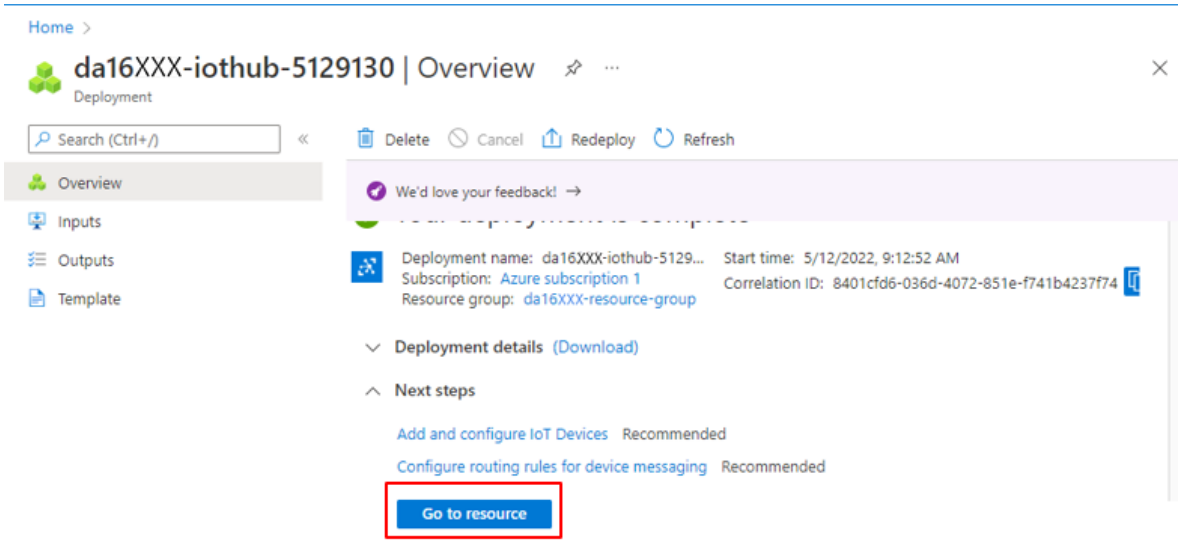

Figure 17: Completed IoT Hub Deployment

The generated IoT hub web page appears.

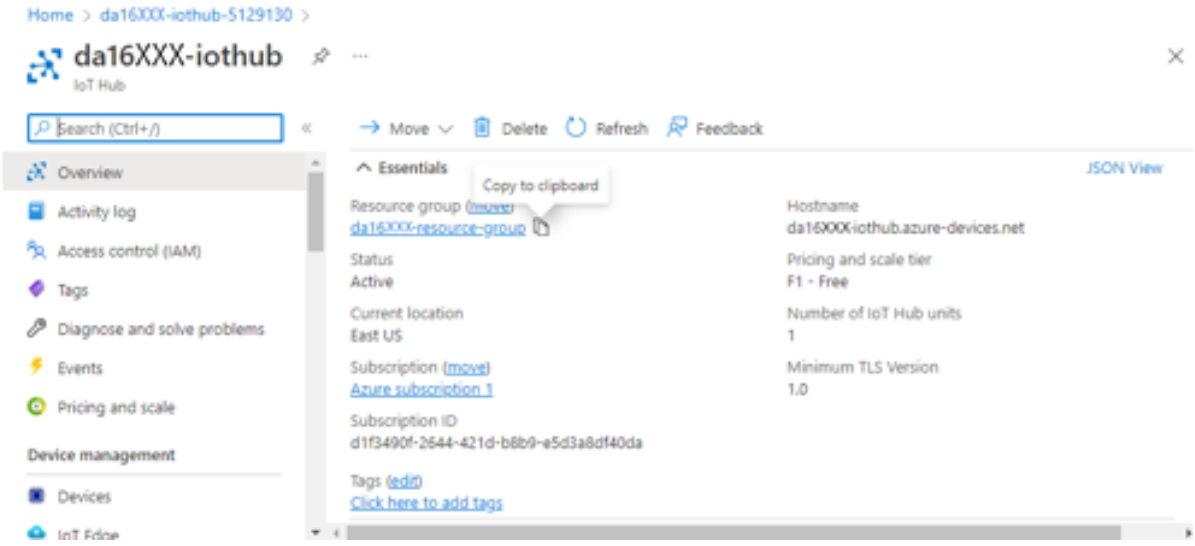

Figure 18: Generated IoT Hub Web Page

3.2.4 Register Devices

IoT devices should be added to Azure IoT Hub.

To add devices:

1. In the created IoT Hub, click **Devices**.

© 2024 Renesas Electronics

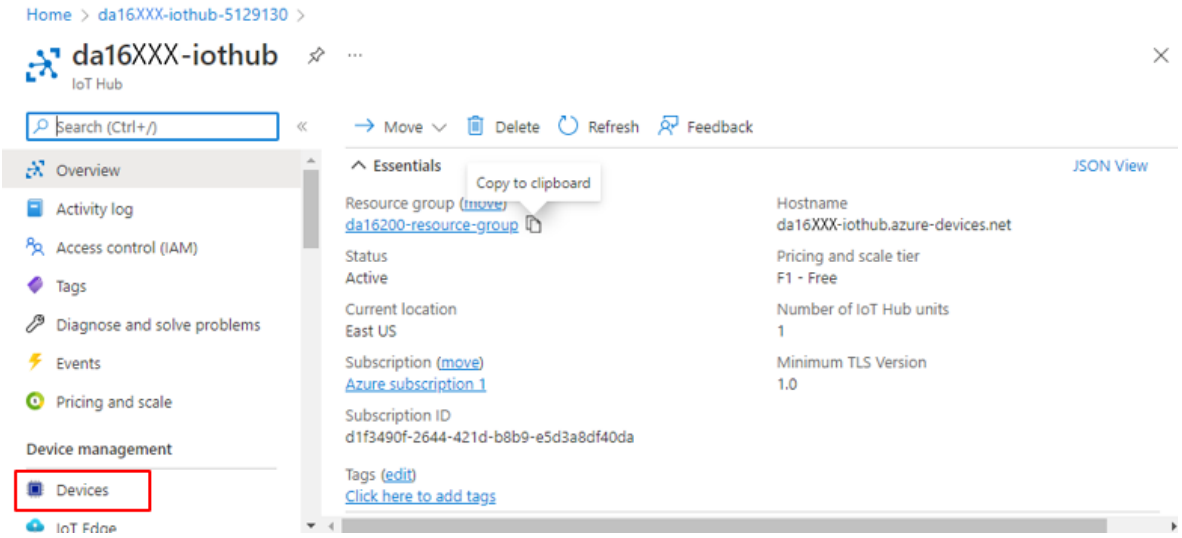

Figure 19: Add Devices

2. In **Devices** menu, click **Add Device**.

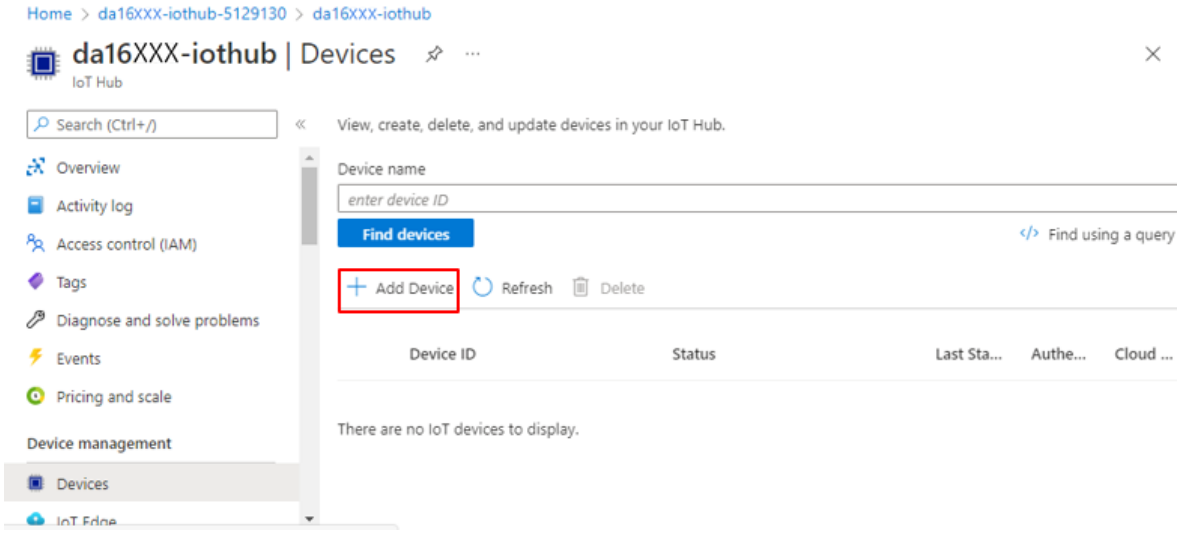

Figure 20: Add Devices (Continued)

© 2024 Renesas Electronics

3. Add the desired Device ID, and then click **Save**.

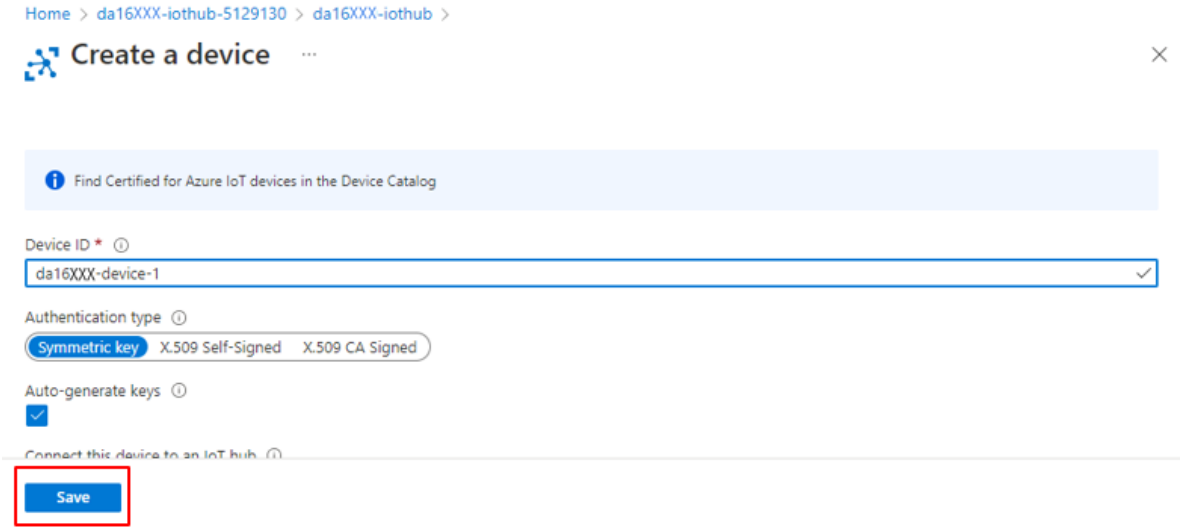

Figure 21: Save Devices

4. The device is registered in the Azure IoT server.

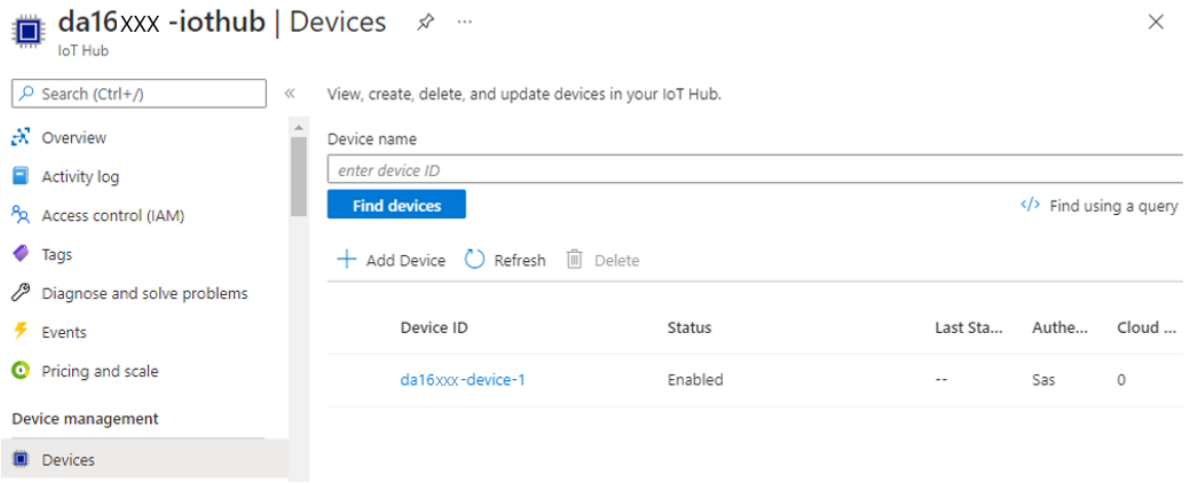

Figure 22: Device Registered in Azure IoT Server

3.3 Configure Azure IoT Hub

The Azure IoT Hub requires several parameters to be configured. These parameters are used for connecting DA16200/DA16600 device and must be set in the parameter of DA16200/DA16600. The Azure IoT Hub and DA16200/DA16600 need to be connected for data communication.

Configure parameters of Azure IoT Hub:

- Device ID: ID registered in Azure IoT Hub

© 2024 Renesas Electronics

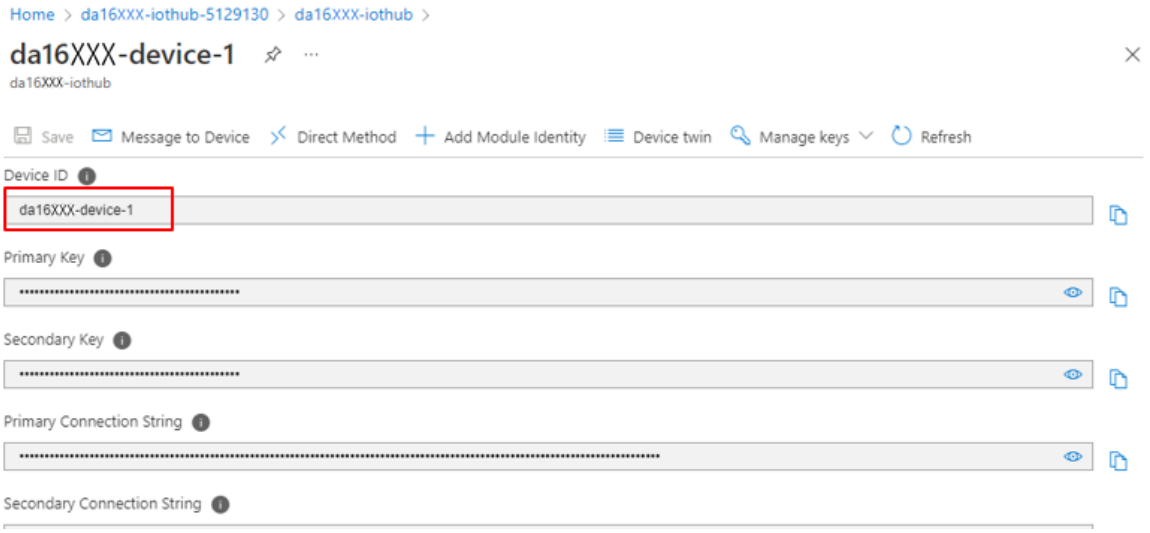

Figure 23: Device ID Registered in Azure IoT Hub

- Device-Primary-Key: Primary key with device ID registered in Azure IoT Hub

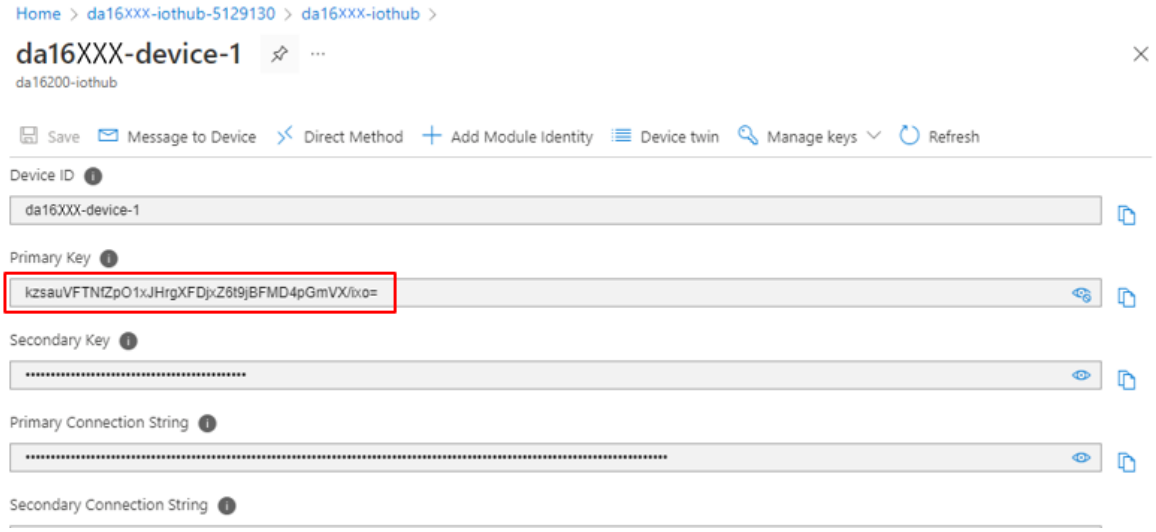

Figure 24: Primary Key with Device ID

© 2024 Renesas Electronics

- Hostname: Azure IoT Hub address

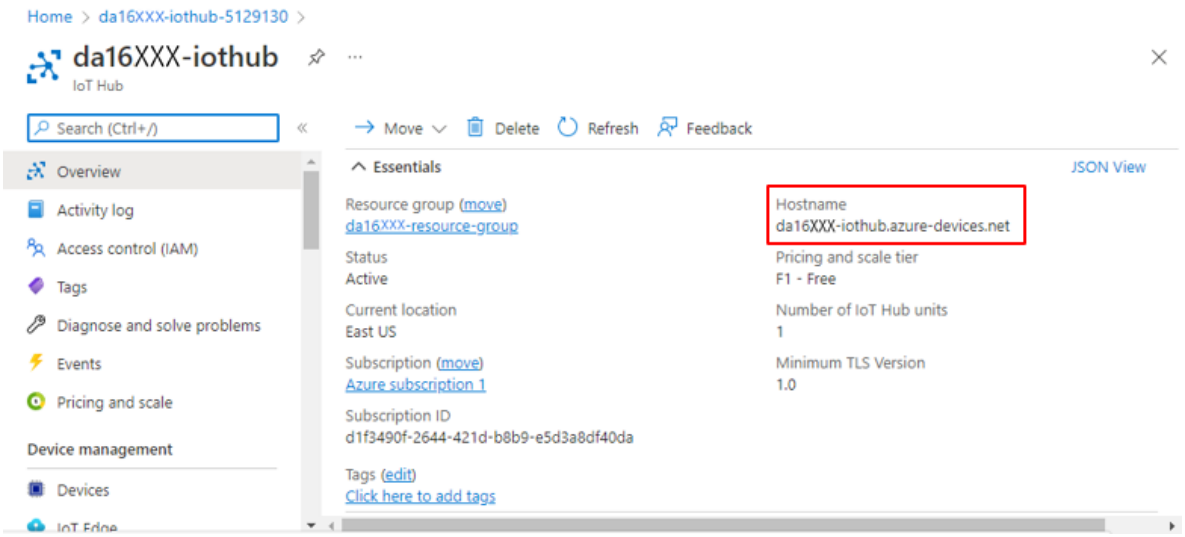

Figure 25: Hostname in Azure IoT Hub

- IOT-HUB-Connection-String: Connection string to communicate with Azure IoT Hub

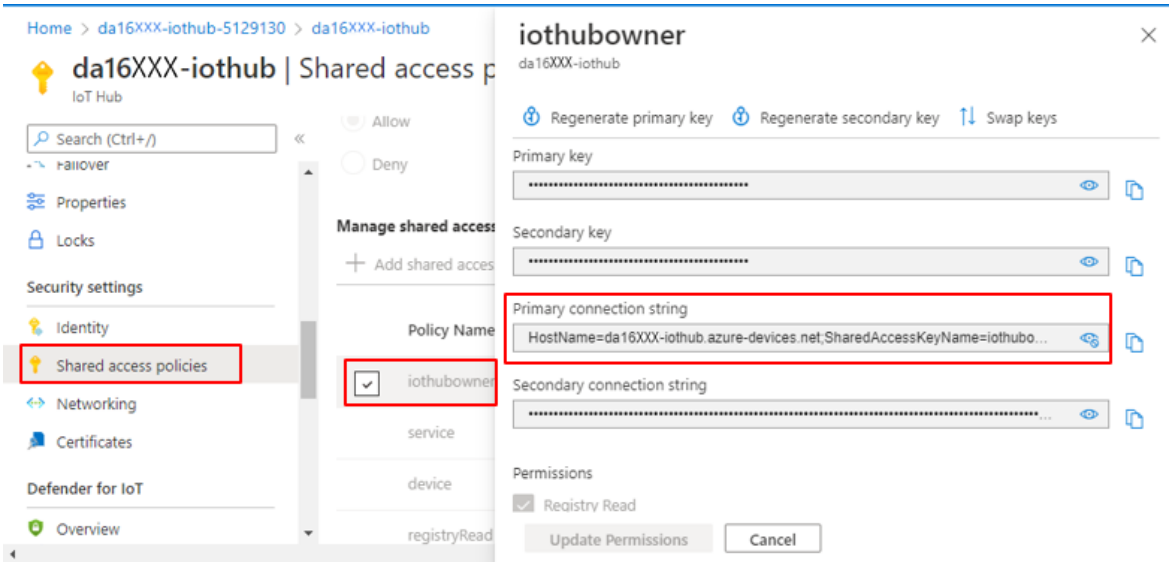

Figure 26: Connection String for Azure IoT Hub

There are two ways to configure these parameters in the DA16200/DA16600:

- Set the default values in the SDK:

  Edit the above settings in app_thing_manager.h, and then compile (for instruction on how to compile the DA16200/DA16600 SDK, see Ref. [1]) and download the firmware image.

```
#define APP_USER_MYHING_NAME              "da16XXX-device-1"  //Device ID
#define APP_USER_MY_DEV_PRIMARY_KEY       "kzsauVFTNfZpO1xJHrgXFDjxZ6t9Jbfmd4PgMvx/ixo="
#define APP_USER_MY_HOST_NAME             "da16XXX-iothub.azure-devices.net"
#define APP_USER_MY_IOTHUB_CONN_STRIN     "HostName= da16XXX-iothub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=a5UBDdFQJ92jVMhVjwJRzY/LR/GfnrEu289Aa806nUw="
```

- Program the values in the DA16200/DA16600 NVRAM:

  In the DA16200/DA16600 debug console, enter factory mode and set the 4 NVRAM parameters as shown below.

© 2024 Renesas Electronics

```
[/DA16200 or /DA16600] # nvram
    Command-List is changed, "NVRAM"
[/DA16200 or /DA16600/NVRAM] # setenv APP_THINGNAME da16XXX-device-1
[/DA16200 or /DA16600/NVRAM] # getenv
……
APP_THINGNAME (STR,17) ....... da16XXX-device-1
```

```
[/DA16200 or /DA16600] # nvram
    Command-List is changed, "NVRAM"
[/DA16200 or /DA16600/NVRAM] # setenv APP_DEV_PRIMARY_KEY kzsauVFTNfZpO1xJHrgXFDjxZ6t9Jbfmd4PgMvx/ixo=
[/DA16200 or /DA16600/NVRAM] # getenv
……
APP_THINGNAME (STR,17) ....... da16XXX-device-1
APP_DEV_PRIMARY_KEY (STR,45) . kzsauVFTNfZpO1xJHrgXFDjxZ6t9Jbfmd4PgMvx/ixo=
```

```
[/DA16200 or /DA16600] # nvram
    Command-List is changed, "NVRAM"
[/DA16200 or /DA16600/NVRAM] # setenv APP_HOSTNAME da16XXX-iothub.azure-devices.net
 [/DA16200 or /DA16600/NVRAM] # getenv
……
APP_THINGNAME (STR,17) ....... da16XXX-device-1
APP_DEV_PRIMARY_KEY (STR,45) . kzsauVFTNfZpO1xJHrgXFDjxZ6t9Jbfmd4PgMvx/ixo=
APP_HOSTNAME (STR,43) ........ da16XXX-iothub.azure-devices.net
```

```
[/DA16200 or /DA16600/NVRAM] # nvram
    Command-List is changed, "NVRAM"
[/DA16200 or /DA16600/NVRAM] # setenv APP_IOTHUB_CONN_STRING
Typing data: (setenv value)
        Cancel - CTRL+D, End of Input - CTRL+C or CTRL+Z
HostName= da16XXX-iothub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=Vmt7BP8BsSwNJRm1tSLJUCliblS1ApXjpd67HqF91xQ=
[/DA16200 or /DA16600/NVRAM] # getenv
……
APP_THINGNAME (STR,17) ....... da16XXX-device-1
APP_DEV_PRIMARY_KEY (STR,45) . kzsauVFTNfZpO1xJHrgXFDjxZ6t9Jbfmd4PgMvx/ixo=
APP_HOSTNAME (STR,43) ........ da16XXX-iothub.azure-devices.net
APP_IOTHUB_CONN_STRING (STR,147)  HostName= da16XXX-iothub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=Vmt7BP8BsSwNJRm1tSLJUCliblS1ApXjpd67HqF91xQ=
```

3.4 Verify Connection Between Azure IoT Hub and DA16200/DA16600

When parameters listed above are configured and firmware is successfully downloaded, boot the DA16200/DA16600 EVK and connect to the Azure IoT Hub. Once connected, the DA16200/DA16600 EVK reports to the Azure IoT Hub and the Azure device twin is updated with the reported items.

The log will appear in the debug console as shown below.

- Logs of the connection between the DA16200 EVK and Azure IoT Hub

```
Wakeup source is 0x4
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)


        ****************************************************
        *             DA16200 SDK Information
        * -------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.2.1 Azure Doorlock Ref.
        * - F/W Version     : FRTOS-GEN01-01-507cc85bd-003442
        * - F/W Build Time  : May  7 2022 17:04:14
        * - Boot Index      : 0
        *
        ****************************************************
```


© 2024 Renesas Electronics

```
……..

[ azure_twin_dpm_auto_start]


AZURE_IOT on Station Mode for "da16200-device-1"

……

[dpmAPPManager] statusFlag : 23
[dpmAPPManager] DM_NEED_INIT
DM_INIT

Info: [azure_dpm_app_init:544] : AZURE_UPDATE_OTA_UNKNOWN
Info: [azure_dpm_app_init:557] : read AZURE_NVRAM_CONFIG_OTA_STATE init status
[dpmAPPManager] DM_NEED_CONNECTION
DM_NEED_CONNECTION

Info: [create_connection_string:740] : connection string : HostName=da16200-standard-iot-hub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=Vmt7BP8BsSwNJRm1tSLJUCliblslApXjpd67HqF91xQ=
Info: [IoTHub_Init:22] : Platform intitializeion OK
Info: [ThreadAPI_Create:55] : thread "azrSubT_0" creation OK (pri=1)
IoTHubDeviceClient_SetMessageCallback()...
Info: [dns_resolver_create:70] : hostName = "da16200-standard-iot-hub.azure-devices.net"
Info: [dns_resolver_create:90] : host IP = "20.194.67.96"
Info: [azure_dpm_app_connect:614] : Root CA registered
[dpmAPPManager] DM_BOOT_WAKEUP
DM_WAKEUP_BOOT

IoTHubDeviceClient_GetTwinAsync()...
IoTHubDeviceClient_SetDeviceMethodCallback()...
IoTHubDeviceClient_SetDeviceTwinCallback()...

Sleep mode 3

DM_FINISH_DEVICE

……
```

- Logs of the connection between the DA16600 EVK and Azure IoT Hub

```
Wakeup source is 0x4
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)


        ******************************************************
        *             DA16600 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.4.0 Azure Doorlock Ref.
        * - F/W Version     : FRTOS-GEN01-01-48589f13c-004594
        * - F/W Build Time  : Aug 29 2022 19:03:29
        * - Boot Index      : 0
        *
        ******************************************************
gpio wakeup enable 00000402
[combo] dpm_boot_type = 1

>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
[combo] BLE_BOOT_MODE_0
[combo] BLE FW VER to transfer ....
   >>> v_6.0.14.1114.2 (id=1) at bank_1
[combo] BLE FW transfer done
……..

[ azure_twin_dpm_auto_start]


AZURE_IOT on Station Mode for "da16XXX-device-1"
```


© 2024 Renesas Electronics

```
.....

[dpmAPPManager] statusFlag : 23
[dpmAPPManager] DM_NEED_INIT
DM_INIT

Info: [azure_dpm_app_init:544] : AZURE_UPDATE_OTA_UNKNOWN
Info: [azure_dpm_app_init:557] : read AZURE_NVRAM_CONFIG_OTA_STATE init status
[dpmAPPManager] DM_NEED_CONNECTION
DM_NEED_CONNECTION

Info: [create_connection_string:740] : connection string : HostName= da16XXX-iothub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=Vmt7BP8BsSwNJRm1tSLJUCliblslApXjpd67HqF91xQ=
Info: [IoTHub_Init:22] : Platform intitializeion OK
Info: [ThreadAPI_Create:55] : thread "azrSubT_0" creation OK (pri=1)
IoTHubDeviceClient_SetMessageCallback()...
Info: [dns_resolver_create:70] : hostName = "da16XXX-iothub.azure-devices.net"
Info: [dns_resolver_create:90] : host IP = "20.194.67.96"
Info: [azure_dpm_app_connect:614] : Root CA registered
[dpmAPPManager] DM_BOOT_WAKEUP
DM_WAKEUP_BOOT

IoTHubDeviceClient_GetTwinAsync()...
IoTHubDeviceClient_SetDeviceMethodCallback()...
IoTHubDeviceClient_SetDeviceTwinCallback()...

Sleep mode 3

DM_FINISH_DEVICE

.....
```

- Logs of items reported to the Azure IoT Hub

```
Info: [deviceTwinCallback:463] : Report =>
{"temperature":23,"battery":100,"doorStateChange":0,"openMethod":"none","doorState":false,"doorBell":false,"doorOpenMode":0,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN","OTAversion":"none"}
```

- Updated items in the Azure Device Twin

To check updated items in the Azure Device Twin:

a. Go to Azure IoT Hub.

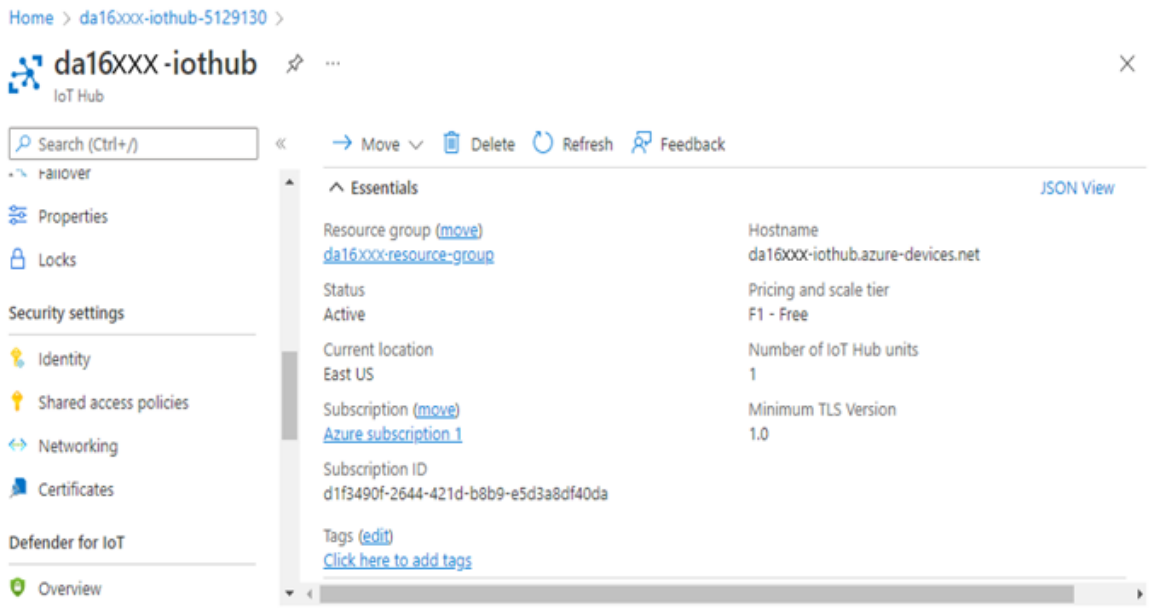

Figure 27: Azure IoT Hub

© 2024 Renesas Electronics

b. Click **Devices**, and then select the **Device ID**.

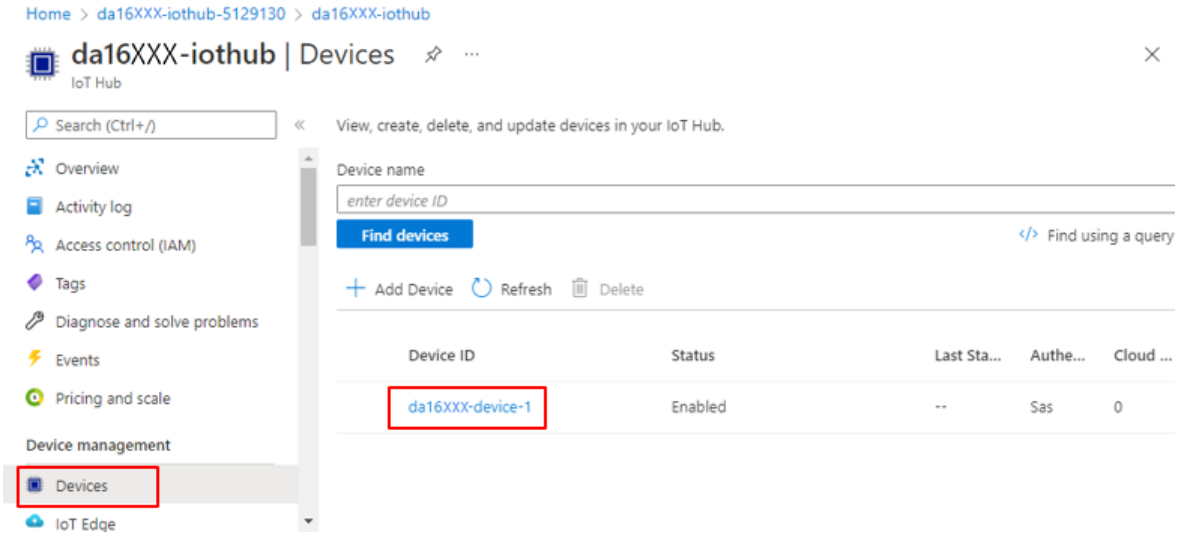

Figure 28: Select Device ID

c. Click Device twin.

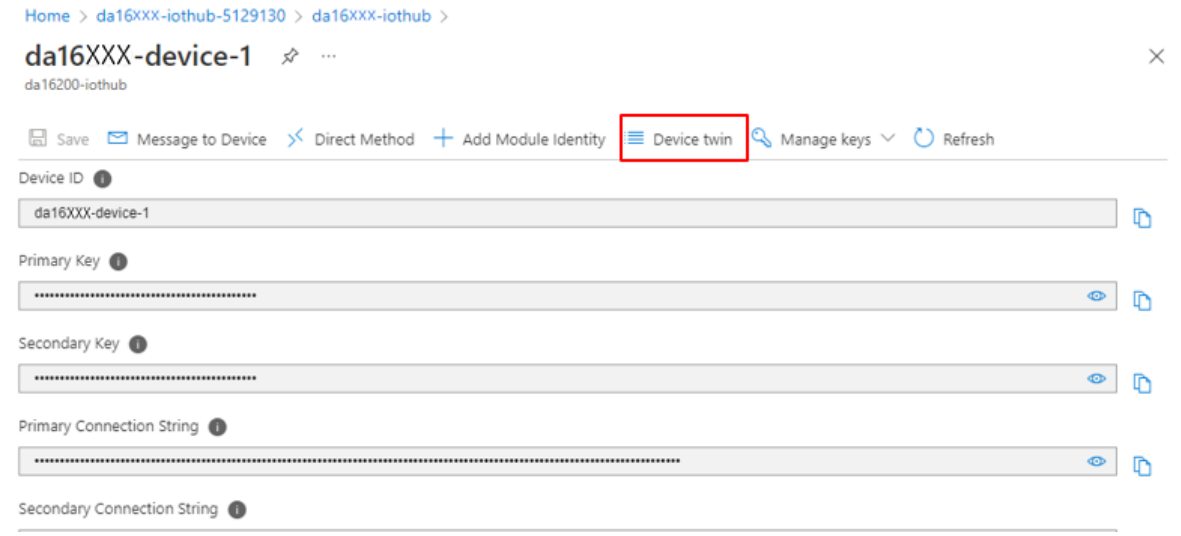

Figure 29: Device Twin in Azure IoT Hub

Figure 30 shows the property of Device Twin. The information shown in the device twin should match with the information shown in the DA16200/DA16600 console log.

© 2024 Renesas Electronics

```
"reported": {
    "temperature": 23,
    "battery": 100,
    "doorStateChange": 0,
    "openMethod": "none",
    "doorState": false,
    "doorBell": false,
    "doorOpenMode": 0,
    "OTAupdate": 0,
    "OTAresult": "OTA_UNKNOWN",
    "OTAversion": "none",
    "MCUOTAversion": "none",
    "$metadata": {
        "$lastUpdated": "2022-08-26T05:37:58.2242565Z",
        "temperature": {
            "$lastUpdated": "2022-08-26T05:37:58.2242565Z"
        },
        "battery": {
            "$lastUpdated": "2022-08-26T05:37:58.2242565Z"
        },
        "doorStateChange": {
            "$lastUpdated": "2022-08-26T05:37:58.2242565Z"
        },
        "openMethod": {
            "$lastUpdated": "2022-08-26T05:37:58.2242565Z"
        },
        "doorState": {
            "$lastUpdated": "2022-08-26T05:37:58.2242565Z"
        },
        "doorBell": {
```


Figure 30: Items in Azure Device Twin

© 2024 Renesas Electronics

4 Door Lock Reference Application

Door lock reference application is available in the official website.

NOTE
Go to the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi) and scroll down to the Software Downloads section. Find “Azure IoT Reference” or type it in the search box, and then select the reference package and download.

4.1 Reference Application in DA16200/DA16600

The following components shown in Figure 31 are required to run the application in DA16200/DA16600 via an Internet connection and Microsoft Azure IoT server.

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

Figure 31: Structure of Azure IoT

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

Provisioning is required for connection between DA16200/DA16600 and Router before connecting DA16200/DA16600 with Azure IoT hub. The provisioning can be done with the Renesas Wi-Fi Provisioning app on either an Android or iOS device. See the user manual Ref. [2] for details on how to install and provision the mobile app. Once provisioning is completed, select **Azure IoT** to open Azure application on mobile device as shown in Figure 32.

© 2024 Renesas Electronics

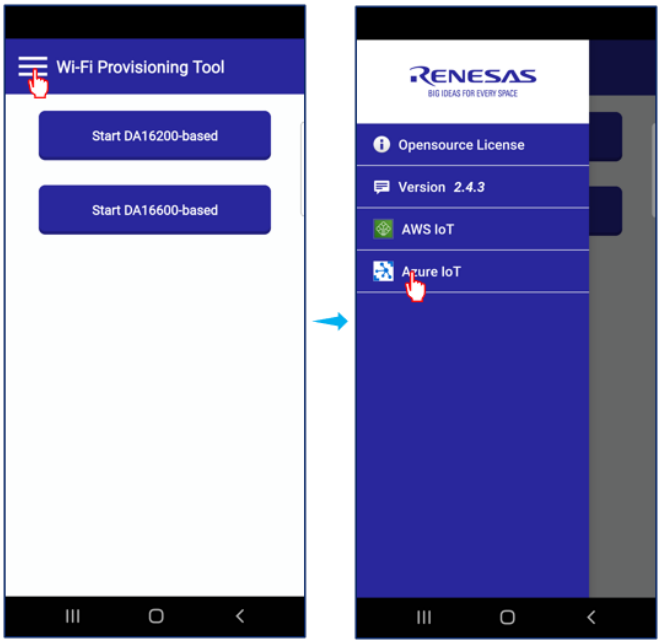

Figure 32: Open Azure IoT on Mobile Device

Figure 33 shows key features of the Azure IoT on the Android phone.

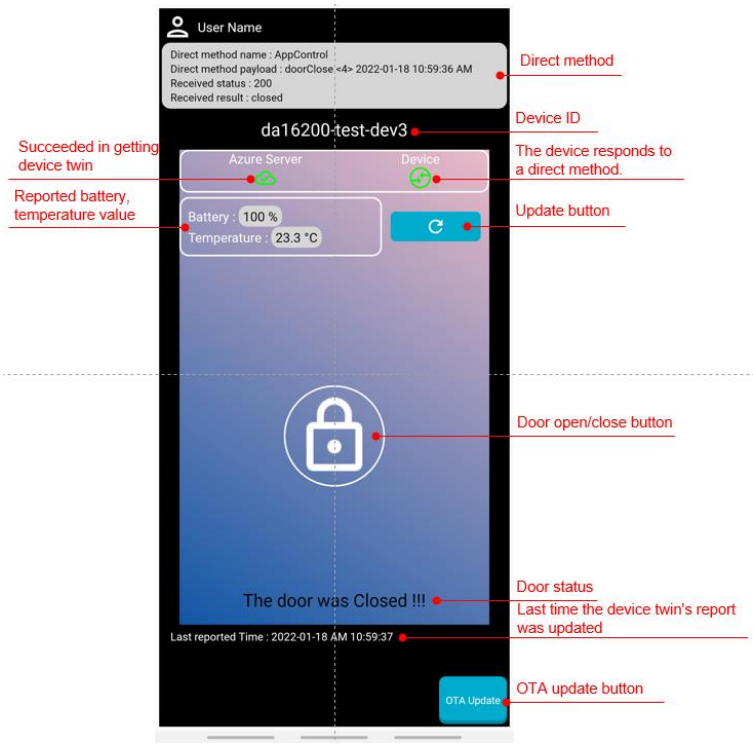

Figure 33: Azure IoT on Mobile Device

4.1.1 Open Door

Figure 34 shows message flows of opening the door.

© 2024 Renesas Electronics

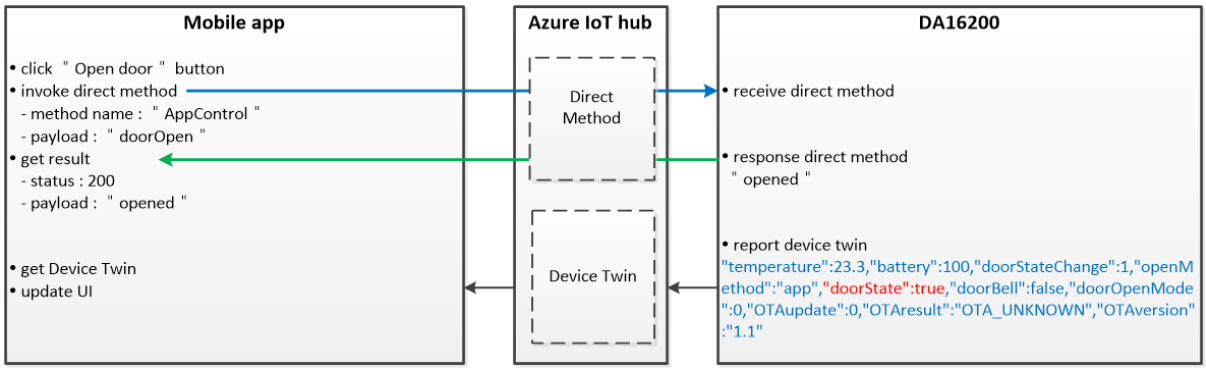

Figure 34: Message Flows of Opening Door

The operation of **opening door** in Android app is shown in Figure 35.

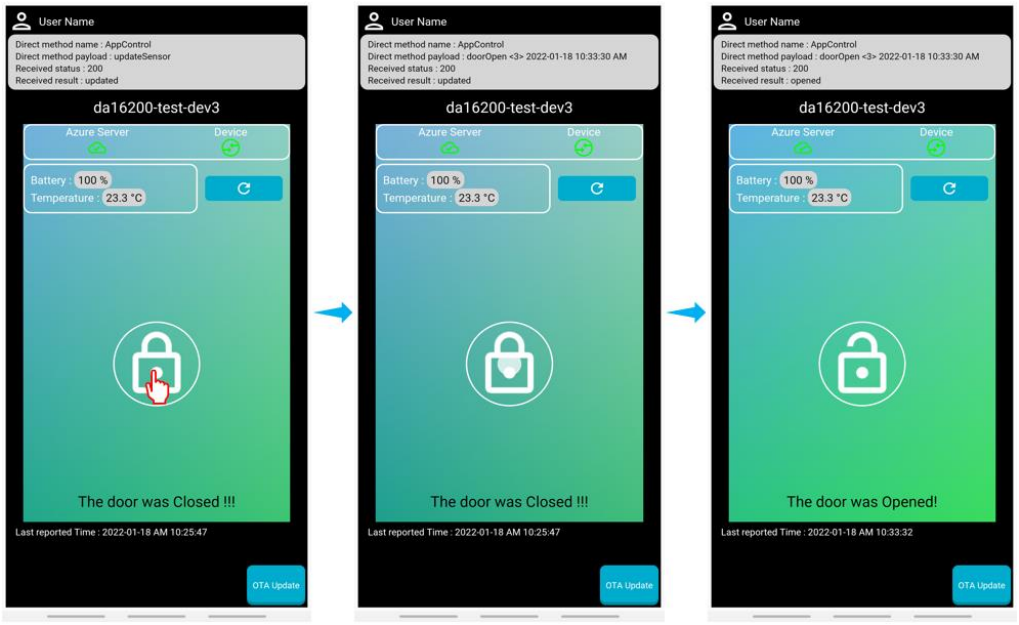

Figure 35: Opening Door on Mobile App

Figure 36 shows the device twin of Azure IoT Hub when the operation of opening door is completed.

© 2024 Renesas Electronics

```
"reported": {
    "temperature": 23,
    "battery": 100,
    "doorStateChange": 1,
    "openMethod": "app",
    "doorState": true,
    "doorBell": false,
    "doorOpenMode": 0,
    "OTAupdate": 0,
    "OTAresult": "OTA_UNKNOWN",
    "OTAversion": "none",
    "$metadata": {
        "$lastUpdated": "2022-05-12T03:20:50.1095307Z",
        "temperature": {
            "$lastUpdated": "2022-05-12T03:20:50.1095307Z"
        },
        "battery": {
            "$lastUpdated": "2022-05-12T03:20:50.1095307Z"
        },
        "doorStateChange": {
            "$lastUpdated": "2022-05-12T03:20:50.1095307Z"
        },
        "openMethod": {
            "$lastUpdated": "2022-05-12T03:20:50.1095307Z"
        },
        "doorState": {
```


Figure 36: Device Twin When Door Is Open

When the operation of opening door is completed, the console logs of DA16200/DA16600 are displayed as below.

```
[mqttOperationCompleteCallback:2102] MQTT_CLIENT_ON_CONNACK & CONNECTION_ACCEPTED: subscribing topics=0x000d
==> elapsed time: 2 ms,  total time: 207 ms
Info: [app_socket_set_recv_timeout:146] : recv timeout(=120) set OK (socket=0)

[SubscribeToMqttProtocol:2441] mqtt_client_subscribe() bypassed ==> elapsed time: 10 ms,  total time: 217 ms

[app_dpm_set_rcv_ready:633] DPM rcv ready & suscription completed ==> elapsed time: 0 ms,  total time: 217 ms
Info: [deviceMethodCallback:347] : method name: AppControl, Command payload : 10,
"doorOpen"thods/POST/AppControl/?$rid=3
Info: [deviceMethodCallback:351] : open comm
Info: [deviceMethodCallback:430] : Command response : "opened"

Info: [deviceMethodCallback:435] : Report =>
{"temperature":23,"battery":100,"doorStateChange":1,"openMethod":"app","doorState":true,"doorBell":false,"door
OpenMode":0,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN","OTAversion":"none"}

Info: [publishDeviceTwinGetMsg:1136] : publish reported data OK
Info: [removeExpiredTwinRequests:1260] : ack_waiting_queue timeout & removed

[app_dpm_set_recv_timeout_flag:556] DPM_RCV_OK_SLEEP set ==> elapsed time: 1162 ms,  total time: 1379 ms
Device Twin reported properties update completed with result: 204
```

4.1.2 Close Door

Figure 37 shows message flows of closing door.

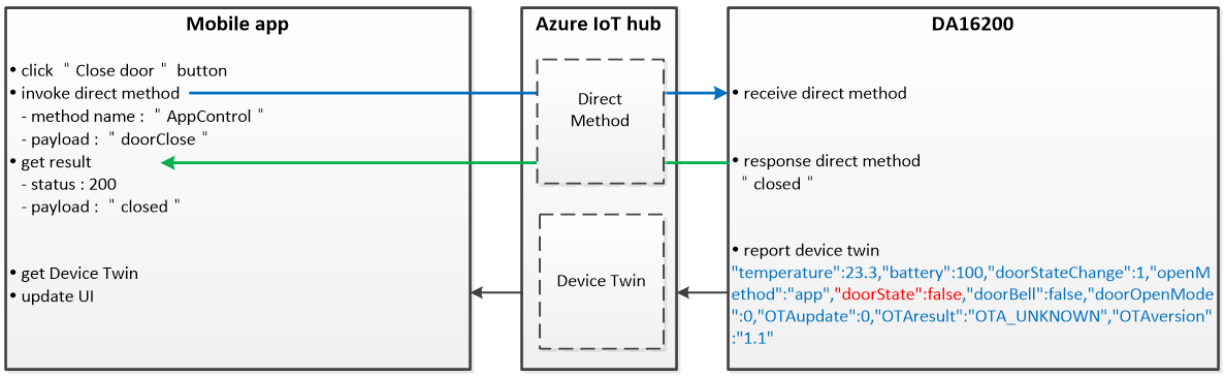

Figure 37: Message Flows of Closing Door

© 2024 Renesas Electronics

The operation of **closing door** in Android app is shown in Figure 38.

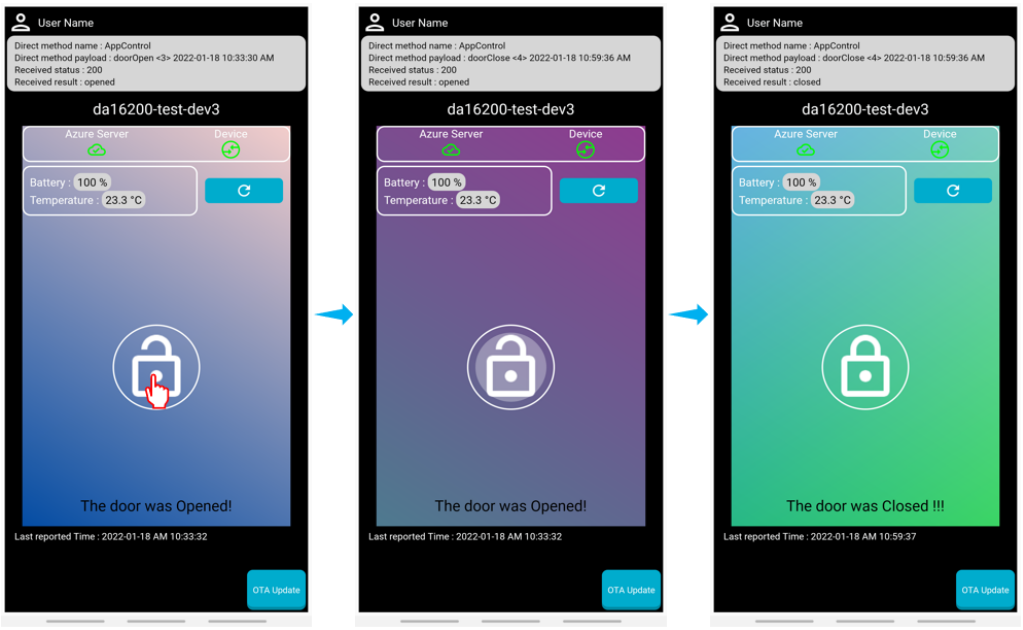

Figure 38: Closing Door on Mobile App

Figure 40 shows the device twin of Azure IoT Hub when the operation for closing door is completed.


```
24          "reported": {
25              "temperature": 23,
26              "battery": 100,
27              "doorStateChange": 1,
28              "openMethod": "app",
29              "doorState": false,
30              "doorBell": false,
31              "doorOpenMode": 0,
32              "OTAupdate": 0,
33              "OTAresult": "OTA_UNKNOWN",
34              "OTAversion": "none",
35              "$metadata": {
36                  "$lastUpdated": "2022-05-12T03:24:05.6886591Z",
37                  "temperature": {
38                      "$lastUpdated": "2022-05-12T03:24:05.6886591Z"
39                  },
40                  "battery": {
41                      "$lastUpdated": "2022-05-12T03:24:05.6886591Z"
42                  },
43                  "doorStateChange": {
44                      "$lastUpdated": "2022-05-12T03:24:05.6886591Z"
45                  },
46                  "openMethod": {
47                      "$lastUpdated": "2022-05-12T03:24:05.6886591Z"
48                  },
49                  "doorState": {
```


Figure 39: Device Twin When Door Is Closed

When the operation for closing door is completed, the console logs of DA16200/DA16600 are displayed as below.

```
[mqttOperationCompleteCallback:2102] MQTT_CLIENT_ON_CONNACK & CONNECTION_ACCEPTED: subscribing topics=0x000d
==> elapsed time: 0 ms,  total time: 204 ms
Info: [app_socket_set_recv_timeout:146] : recv timeout(=120) set OK (socket=0)

[SubscribeToMqttProtocol:2441] mqtt_client_subscribe() bypassed ==> elapsed time: 13 ms,  total time: 217 ms

[app_dpm_set_rcv_ready:633] DPM rcv ready & suscription completed ==> elapsed time: 1 ms,  total time: 218 ms
Info: [deviceMethodCallback:347] : method_name: AppControl, Command payload : 11, "doorClose"hods/
Info: [deviceMethodCallback:367] : close comm
Info: [deviceMethodCallback:430] : Command response : "closed"
```


© 2024 Renesas Electronics

```
Info: [deviceMethodCallback:435] : Report =>
{"temperature":23,"battery":100,"doorStateChange":1,"openMethod":"app","doorState":false,"doorBell":false,"doo
rOpenMode":0,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN","OTAversion":"none"}

Info: [publishDeviceTwinGetMsg:1136] : publish reported data OK
Info: [removeExpiredTwinRequests:1260] : ack_waiting_queue timeout & removed

[app_dpm_set_recv_timeout_flag:556] DPM_RCV_OK_SLEEP set ==> elapsed time: 1141 ms,  total time: 1359 ms
Device Twin reported properties update completed with result: 204
```

4.2 Reference Application in Host MCU

Application in host MCU can control DA16200/DA16600 and connection between the host MCU and mobile phone through Azure IoT server using AT commands. Figure 40 shows the Azure IoT using firmware images for AT commands and host MCU.

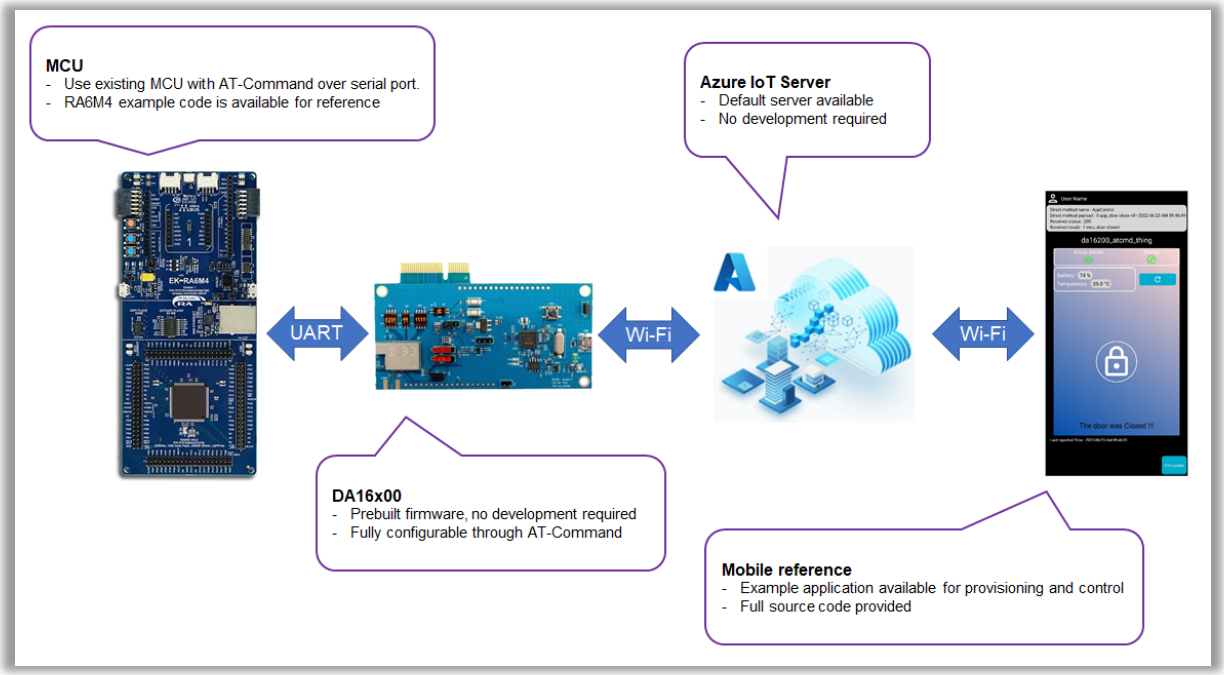

Figure 40: Azure IoT Using Firmware Images for AT Commands and Host MCU

4.2.1 Download Package for Door Lock Reference Application in Host MCU

A firmware image for AT command and application in MCU are available in 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 as follows:

- \DA16200 or \DA16600
  - Firmware images for the DA16200/DA16600 Wi-Fi devices
  - Tera Term script for downloading the firmware images to a DA16200/DA16600 Wi-Fi device
- \DA16200\script (\DA16600\script)
  - Tera Term script which demonstrates how to use AT commands for Azure IoT on a personal computer and the DA16200/DA16600 EVK\doc
  - Getting Stared with AT commands for Azure IoT:
    - Introduces the DA16200/DA16600 AT command for Azure IoT and describes how to set up the development environment and test the examples

© 2024 Renesas Electronics

  - Describes how to connect an external host to the DA16200/DA16600 EVB for using AT commands for Azure IoT
  - Describes AT commands for Azure IoT command list
- \MCU
  - Sample application project based on the RA6M4 development environment which demonstrates how to use AT commands for Azure IoT

4.2.2 Hardware Connections between DA16200/DA16600 and Host MCU

The hardware components listed in Figure 41 are required to run door lock reference application using AT commands and host MCU.

- DA16200/DA16600 EVB
- EK-RA6M4 board
- Windows laptop or personal computer

In addition, hardware connections listed below are required for each operation.

- UART0: Programming firmware images and monitoring logs from DA16200/DA16600
- UART1 or UART2: AT command interface between DA16200/DA16600 and MCU
- GPIO from host MCU to the DA16200/DA16600 to wake up the DA16200/DA16600 from DPM low power mode (DPM LPM)
- GPIO from the DA16200/DA16600 to host MCU to wake up the MCU in sleep mode

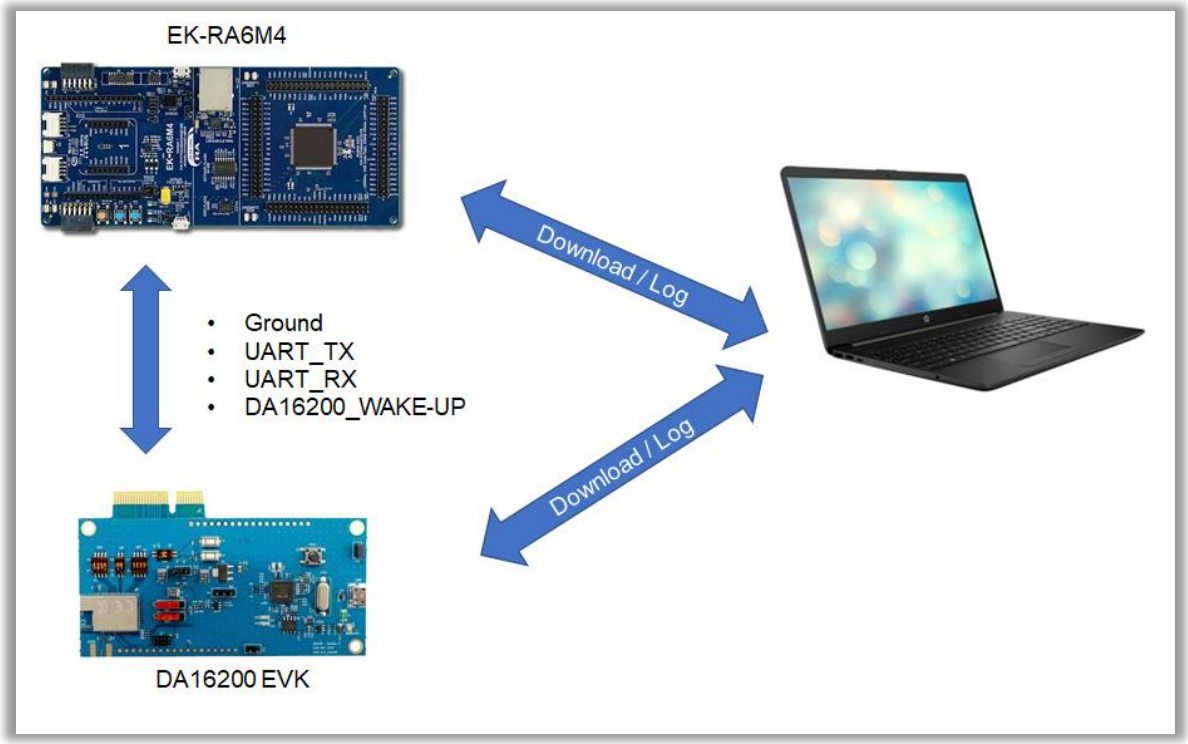

Figure 41: Hardware Configuration

The pin connections between the DA16200/DA16600 EVB and the EK-RA6M4 board are shown in Table 1.

© 2024 Renesas Electronics

Table 1: Pin Connection

Function	DA16200 EVB		DA16600 EVB		EK-RA6M4 Board	
	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	PA10
UART_RX	J4.12	RX1/GPIOA_5	J2.4	RX2/GPIOC_7	J23-1	PA9
DA16200_WAKE_UP	J3.11	RTC_WAKE_UP2	SW1	RTC_WAKE_UP2	J23-6	PB8

4.2.2.1 UART Connection for AT Commands

The default configuration of UART1 (DA16200 EVB) or UART2 (DA16600 EVB) for AT commansd is shown in Table 22.

Table 2: Default Configuration for UART1 or UART2

Settings	Value
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control (HW/SW)	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 42.

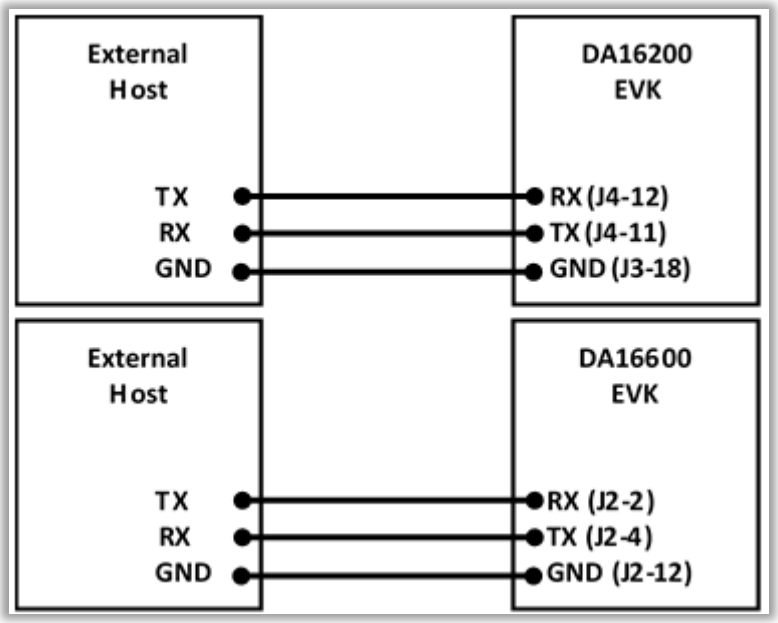

Figure 42: Default UART HW Connection

If GPIO pin configuration is changed using AT commands, other connections for UART1 can be used as shown in Figure 43. The below AT command is to use GPIOA_2 for UART1 TX and GPIOA_3 for UART1 RX. Table 3 shows the pin combination for UART1.

© 2024 Renesas Electronics

```
AT+AZU=SET NV_PIN_BMUX BMUX_UART1d          // GPIOA_2 and GPIOA_3 for UART1
```

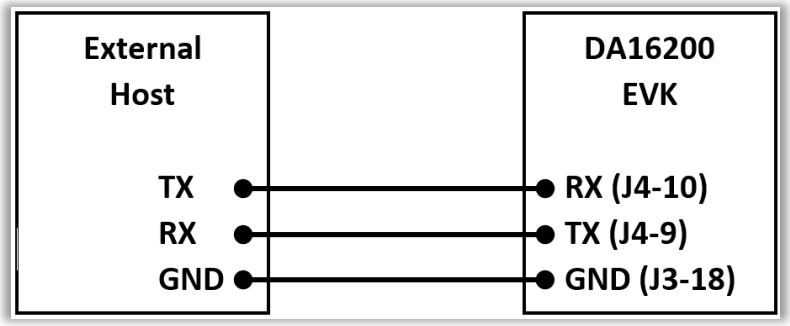

Figure 43: Sample Example of UART1 Connection

Table 3: UART1 Pin Configuration

PIN Mux	GPIO	Signal Name
PIN_AMUX	GPIOA_0	TX
	GPIOA_1	RX
PIN_BMUX	GPIOA_2	TX
	GPIOA_3	RX
PIN_CMUX	GPIOA_4	TX
	GPIOA_5	RX
PIN_DMUX	GPIOA_6	TX
	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 wakeup event is triggered when the GPIO pin of the host MCU changes from Low to High and then back to Low.

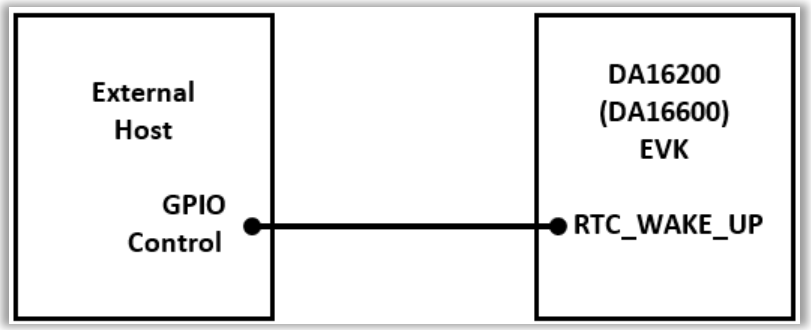

Figure 44: HW Connection for Waking up DA16200/DA16600

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 44. 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 host MCU by default (see Figure 45) and it can be configured using the following AT commands.

© 2024 Renesas Electronics

```
AT+AZU SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A          // GPIO_A port
AT+AZU SET APP_MCU_WKAEUP_PIN GPIO_PIN11            // GPIO_11
```

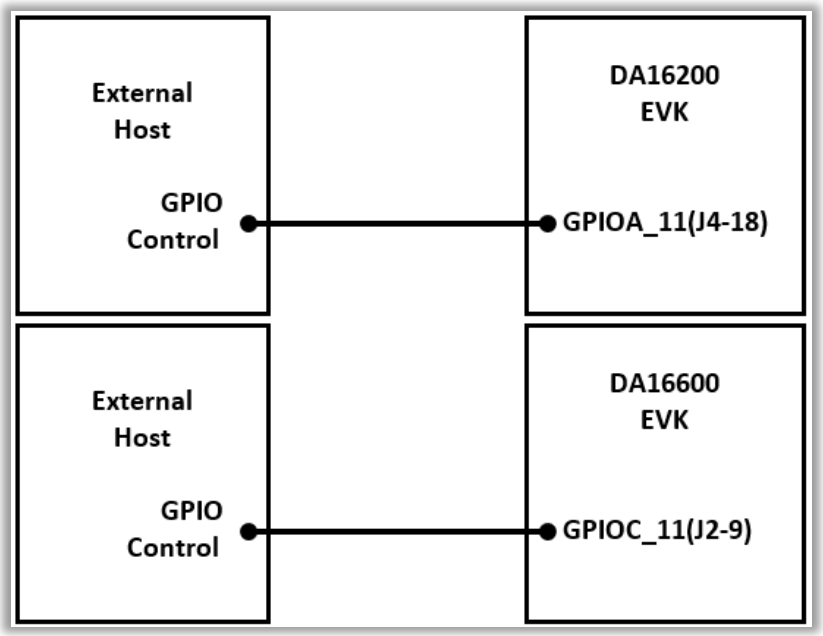

Figure 45: Default Pin Configuration for Waking Up Host MCU

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 host MCU using the below AT commands (see Figure 46).

```
AT+AZU SET APP_MCU_WKAEUP_PORT GPIO_UNIT_C          // GPIOC port
AT+AZU SET APP_MCU_WKAEUP_PIN GPIO_PIN6             // GPIOC_6
```

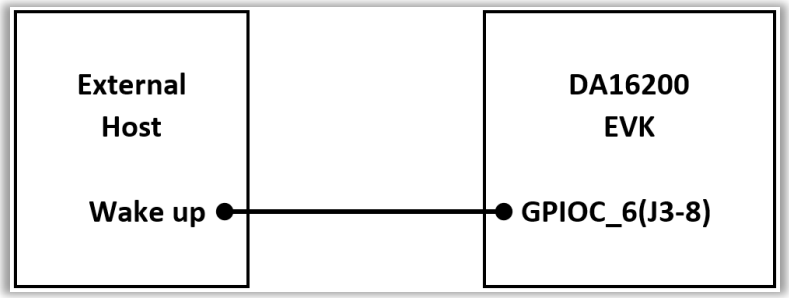

Figure 46: Another Pin Configuration for Waking up Host MCU

Table 4: GPIO Pin Configuration

Port	PIN Mux	GPIO(Port)_Pin
GPIO_UNIT_A	PIN_AMUX	GPIOA_0
		GPIOA_1
	PIN_BMUX	GPIOA_2
		GPIOA_3

© 2024 Renesas Electronics

Port	PIN Mux	GPIO(Port)_Pin
	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

4.2.3 Programming Firmware Images for DA16200/DA16600

When using an EVB for the first time, the firmware must be updated to the latest version. See Ref [1] for details. After programming the firmware image, factory reset is required to enter the Azure-IoT configuration setting mode. This can be done by pushing the “Factory_RST” button for 5 seconds as shown in Figure 47 and Figure 48, and the logs from DA16200/DA16600 are shown in box below.

Figure 47: Factory Reset Button on DA16200 EVB

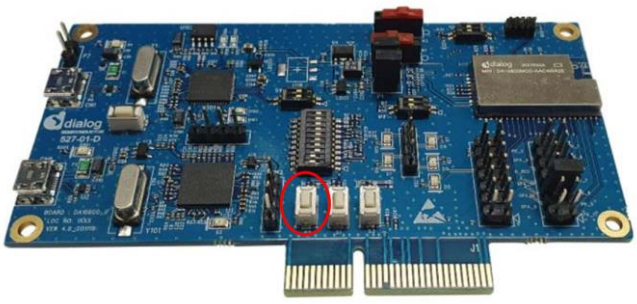

Figure 48: Factory Reset Button on DA16600 EVB

© 2024 Renesas Electronics

```
Soft-AP is Ready (d4:3d:39:11:5e:73)
>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
[reg_user_atcmd_help_cb] Start AZURE AT Command
[UART ready notification]
[http_server_task] HTTP-Server Start!!
=======================================
[ AZURE-IOT AT COMMAND ]
[ azure_twin_dpm_auto_start]

AZURE_IOT on Station Mode for "DA16200" \\For DA16600, AZURE_IOT on Station Mode for "DA16600"
=======================================

[azure_twin_dpm_auto_start] mcu_wakeup_port=-1, mcu_wakeup_pin=0x0
default set to mcu_wakeup_port=0, mcu_wakeup_pin=0x800

Root CA: X
Not ready Cert for Azure service ...

invalid AZURE feature...can't start AZURE thread...check again
.. UART ready
```

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

4.2.4 Configure Components for Testing

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

- Unique device id
- Primary key
- Primary connection string
- Hostname

The information can be set in the source code for host MCU or using the provided scripts in the downloaded package. See Programming Firmware Images section in Ref [1] for how to run the macro script. The scripts are located in the following location.

```
\DA16200\script\doorlock.ttl
```


© 2024 Renesas Electronics

```
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 da16200_atcmd_thing"

mpause 400

;set primary key of device

sendln "SET APP_DEV_PRIMARY_KEY dWKJjGHX9TiCOPheiKyRjjcmmUtBBZsuASBB+K6OoGs="

mpause 400

;set hostname of device

sendln "SET APP_HOSTNAME azure-atcmd-iothub.azure-devices.net"

mpause 400

;set connection string of device

sendln "SET APP_IOTHUB_CONN_STRING HostName=azure-atcmd-iothub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=Xq970WfOFvEmVU9Ixcm96zPGTKh1ztcoxF64QBmyvU8="

mpause 400
```

 © 2024 Renesas Electronics

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

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

```
const char* cmd_set_azure_cfg[MAX_CFG_NUM] =
{
  "\r\nAT+"PLATFORM" SET APP_BOARD_FEATURE EVK\r\n",
  "\r\nAT+"PLATFORM" SET APP_THINGNAME da16200_atcmd_thing\r\n",
  "\r\nAT+"PLATFORM" SET APP_DEV_PRIMARY_KEY dWKJjGHX9TiCOPheiKyRjjcmmUtBBZsuASBB+K6OoGs=\r\n",
  "\r\nAT+"PLATFORM" SET APP_HOSTNAME azure-atcmd-iothub.azure-devices.net\r\n",
  "\r\nAT+"PLATFORM" SET APP_IOTHUB_CONN_STRING HostName=azure-atcmd-iothub.azure-
devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=Xq970WfOFvEmVU9Ixcm96zPGTKh1ztcoxF64QBmyvU8=\r\n"
,
  "\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 battery 0 1\r\n",             /* shadow   int */
  "\r\nAT+"PLATFORM" CFG 3 temperature 2 1\r\n",         /* shadow   float */
  "\r\nAT+"PLATFORM" CFG 4 doorState 1 1\r\n",           /* shadow   str */
  "\r\nAT+"PLATFORM" CFG 5 app_shadow 1 2\r\n",          /* mcu sub. str */
  "\r\nAT+"PLATFORM" CFG 6 mcu_shadow 1 0\r\n",          /* mcu pub. str */
"\r\nAT+"PLATFORM" CFG 7 openMethod 1 1\r\n",          /* shadow   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",             /* 1740 */
  "\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" SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A\r\n",    /* GPIO_UNIT_A or GPIO_UNIT_C */
  "\r\nAT+"PLATFORM" SET APP_MCU_WKAEUP_PIN GPIO_PIN11\r\n"      /* GPIO_PIN0 ~ GPIO_PIN11 or
GPIO_PIN6~GPIO_PIN8 */
};
```

4.2.5 Test without Host MCU

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

Door lock for two-way communication: \DA16200\script\doorlock.ttl.

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

4.2.6 Test with Host MCU

E²studio is required for building source code for host MCU and programing the images to host MCU. Visit the Renesas website (https://www.renesas.com/us/en/software-tool/e-studio) for downloading and installing e²studio. After installing e²stuido, follow the steps for building and programming.

1. Import the project file at \MCU\RA6M4\ (see Figure 49).

© 2024 Renesas Electronics

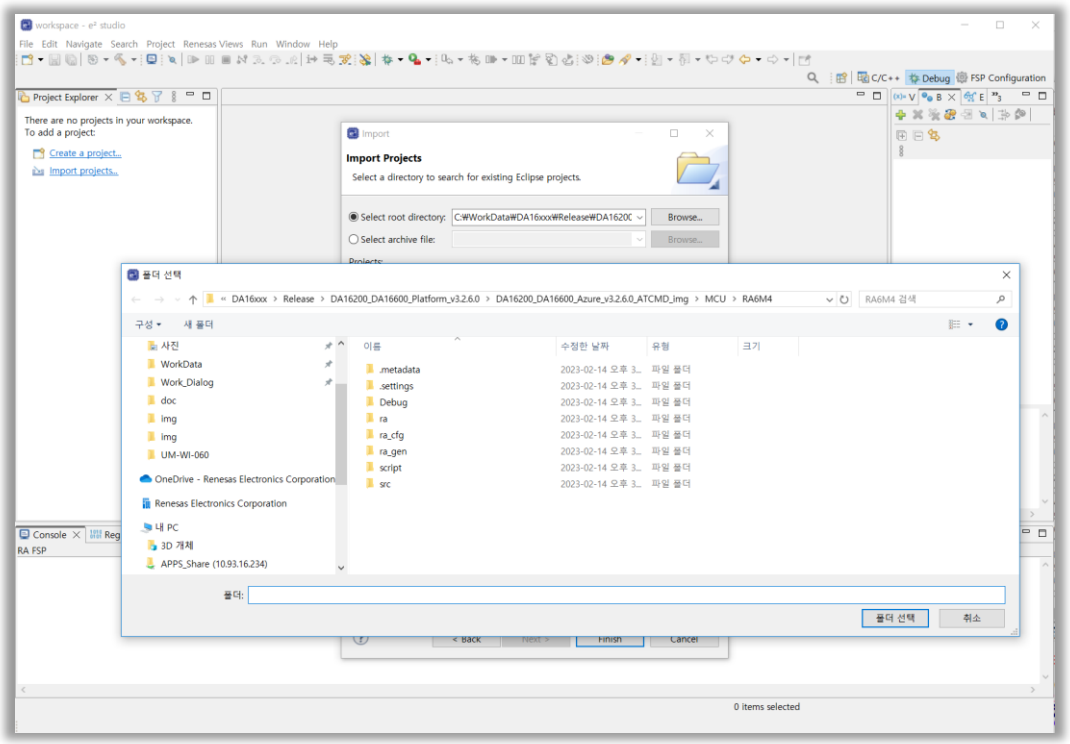

Figure 49: e²studio Project File

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

2. Select Configurations.xml to set FSP configuration of the RA6M4 MCU (see Figure 50).

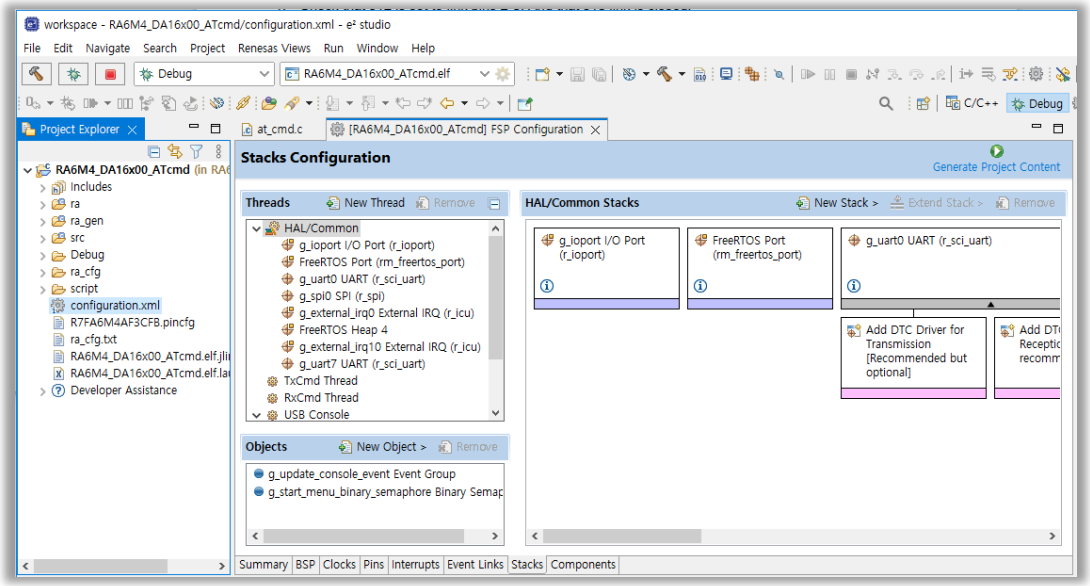

Figure 50: FSP Configuration

3. Use the “device id" (thingname), “primary key”, “hostname”, and “primary connection string” received from the FAE to test without setting up a server.
4. Change the configuration to the received configuration (see Figure 51).

© 2024 Renesas Electronics

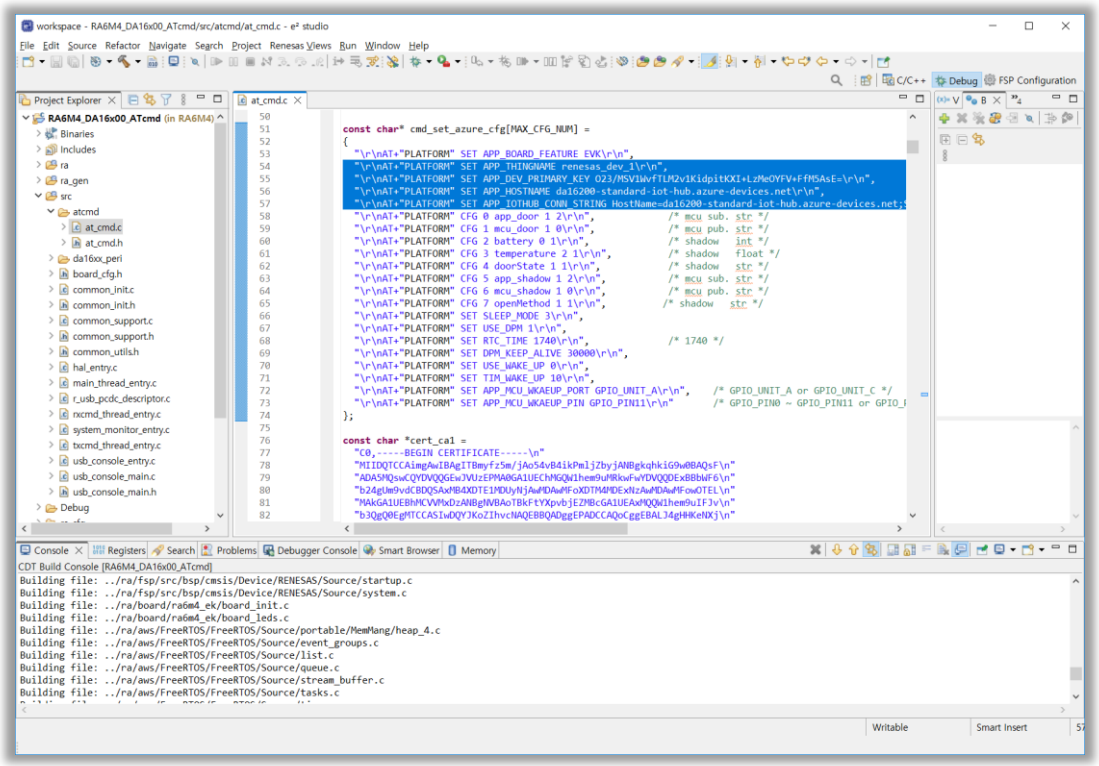

Figure 51: Thingname/Primarykey/Hostname/IotHub Connection String

5. Select `project > Build all` to build (see Figure 52).

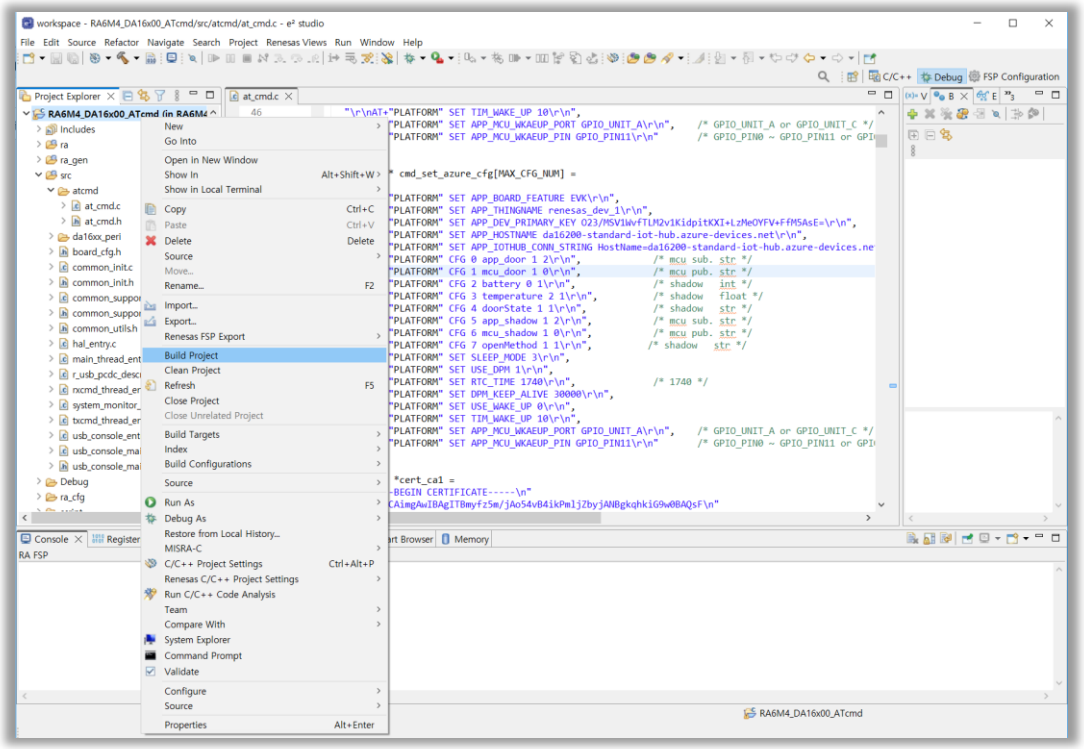

Figure 52: Build Project

6. Select `Debug Configurations` to set the connection to the RA6M4 MCU (see Figure 53).

 © 2024 Renesas Electronics

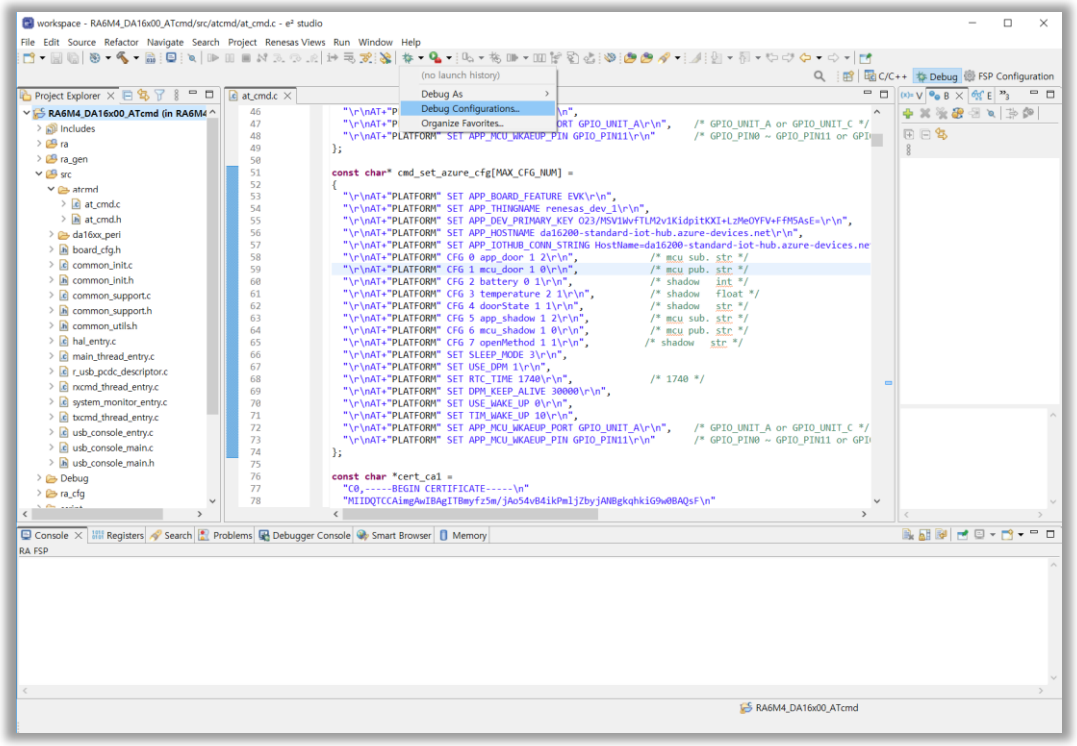

Figure 53: Debug Configurations

7. Change the configuration as shown in Figure 54 and then select the **Apply** and **Debug** buttons.

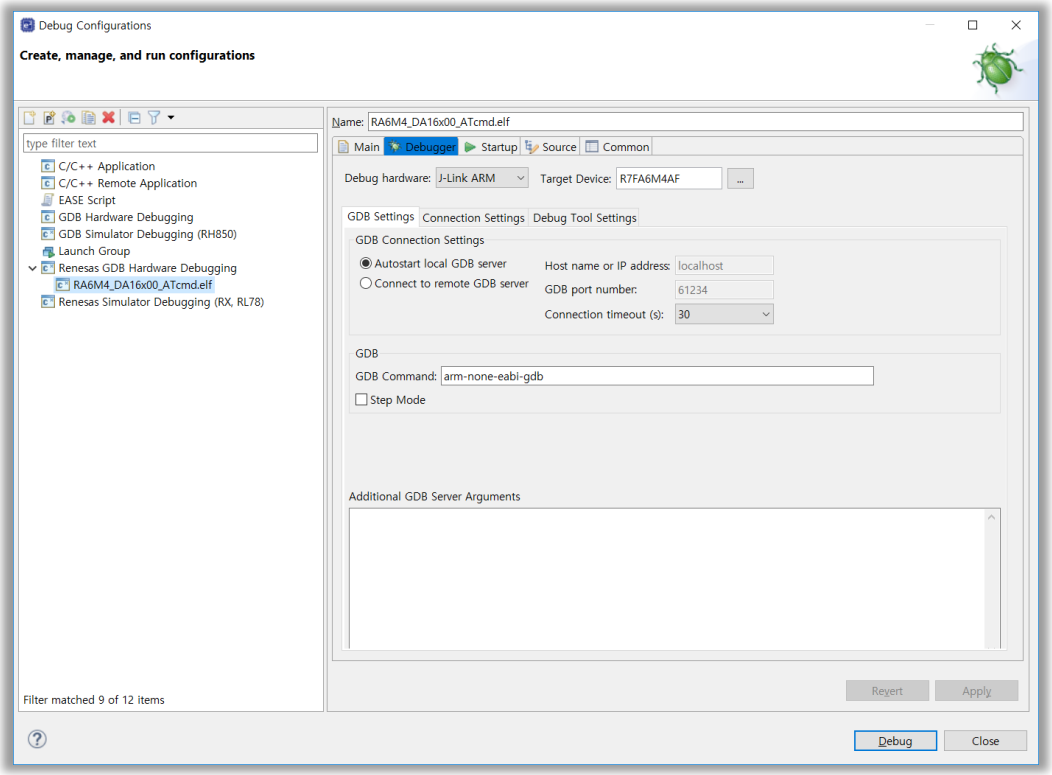

Figure 54: Set Debug Configurations

8. The image for host MCU is downloaded to the device.

© 2024 Renesas Electronics

9. The following shows the console output of the DA16200/DA16600 after a factory mode reset.

```
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:39:10:d5:07
>>> softap1 interface add OK
>>> AP Operating Channel: 1(2412)
>>> Network Interface (wlan1) : UP
BSS Isolate Disabled
Soft-AP is Ready (d4:3d:39:10:d5:07)
>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
[reg_user_atcmd_help_cb] Start AZURE AT Command
[UART ready notification]
[http_server_task] HTTP-Server Start!!
=======================================
[ AZURE-IOT AT COMMAND ]
[ azure_twin_dpm_auto_start]
AZURE_IOT on Station Mode for "DA16200" \\For DA16600, AZURE_IOT on Station Mode for "DA16600"
=======================================
[azure_twin_dpm_auto_start] mcu_wakeup_port=-1, mcu_wakeup_pin=0x0
default set to mcu_wakeup_port=0, mcu_wakeup_pin=0x800
Root CA: X
Not ready Cert for Azure service ...
invalid AZURE feature...can't start AZURE thread...check again
.. UART ready
```


© 2024 Renesas Electronics

10. The following shows the console output of the DA16200/DA16600 when setting the Azure IoT configuration through AT commands from an MCU.

```
================================================

argc num = 2

argv[0]: AT+AZU

argv[1]: SET APP_BOARD_FEATURE EVK

================================================

APP SET EVK

================================================

argc num = 2

argv[0]: AT+AZU

argv[1]: SET APP_THINGNAME da16200_atcmd_thing

================================================

APP SET da16200_atcmd_thing

================================================

argc num = 2

argv[0]: AT+AZU

argv[1]: SET APP_DEV_PRIMARY_KEY dWKJjGHX9TiCOPheiKyRjjcmmUtBBZsuASBB+K6OoGs=

================================================

APP SET dWKJjGHX9TiCOPheiKyRjjcmmUtBBZsuASBB+K6OoGs=

================================================

argc num = 2

argv[0]: AT+AZU

argv[1]: SET APP_HOSTNAME azure-atcmd-iothub.azure-devices.net

================================================

APP SET azure-atcmd-iothub.azure-devices.net

================================================

argc num = 2

argv[0]: AT+AZU

argv[1]: SET APP_IOTHUB_CONN_STRING HostName=azure-atcmd-iothub.azure-devices.net;SharedAccess

KeyName=iothubowner;SharedAccessKey=Xq970WfOFvEmVU9Ixcm96zPGTKh1ztcoxF64QBmyvU8=

================================================

APP SET HostName=azure-atcmd-iothub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAc

cessKey=Xq970WfOFvEmVU9Ixcm96zPGTKh1ztcoxF64QBmyvU8=

================================================

argc num = 2

argv[0]: AT+AZU

argv[1]: CFG 0 app_door 1 2

================================================

================================================

Att[0] number   : 0

Att[0] name       : app_door

Att[0] data type: 1

Att[0] MQTT type: 2

================================================
```

 © 2024 Renesas Electronics

11. The following shows the console output of the DA16200/DA16600 after the Soft AP has been configured and it is waiting to be provisioned by the mobile application.

```
Soft-AP is Ready (d4:3d:39:11:5e:73)
[http_server_task] HTTP-Server Start!!
=========================================
[ AZURE-IOT AT COMMAND ]
[ azure_twin_dpm_auto_start]
AZURE_IOT on Station Mode for "da16200_atcmd_thing"
=========================================
[azure_twin_dpm_auto_start] mcu_wakeup_port=0, mcu_wakeup_pin=0x800
Root CA: 0
subscribe index=0, name=app_door
AZURE_IOT AP Mode  da16200_atcmd_thing
+ATPROV=STATUS 1
=================================================
[Start Provisioning with TCP/TLS] .. Soft AP Mode
=================================================
[app_provision_TCP_server_thread] Create ...
[app_provision_switch_client_thread] Create...(status=0) [21]
[app_provision_TLS_server_thread] Create TLS...
>>> Start Provisioning Server (TLS) ...
Wait Accept (TLS)...
> HTTPS-Server Start !!!
> Wi-Fi Scan request success.
(0) Xiaomi_AX9000 / 3 / -26 / 2412
(1) ZIO-FREEZIO / 3 / -26 / 2437
(2) KAON_GAPK_7600 / 3 / -28 / 2452
(3) HiWiFi_5B2310 / 3 / -31 / 2417
(4) N_TPLINK_EAP620HD_WPA2-ENT / 3 / -32 / 2462
(5) N_TPLINK_EAP620HD_WPA23-PSK / 3 / -32 / 2462
(27) SAT_Synology_MR2200AC_OWE / 3 / -54 / 2437
(28) JMC_DIR-615_WPA1_TKIP / 2 / -55 / 2462
(29) N_A3004_ENT_Synology / 3 / -56 / 2422
[app_provision_TCP_server_thread] socket().. status=1
 Wait Accept...
```

4.3 Mobile App Demo

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

© 2024 Renesas Electronics

4.3.1 Open Door

A message flow of **opening door** is shown in Figure 55.

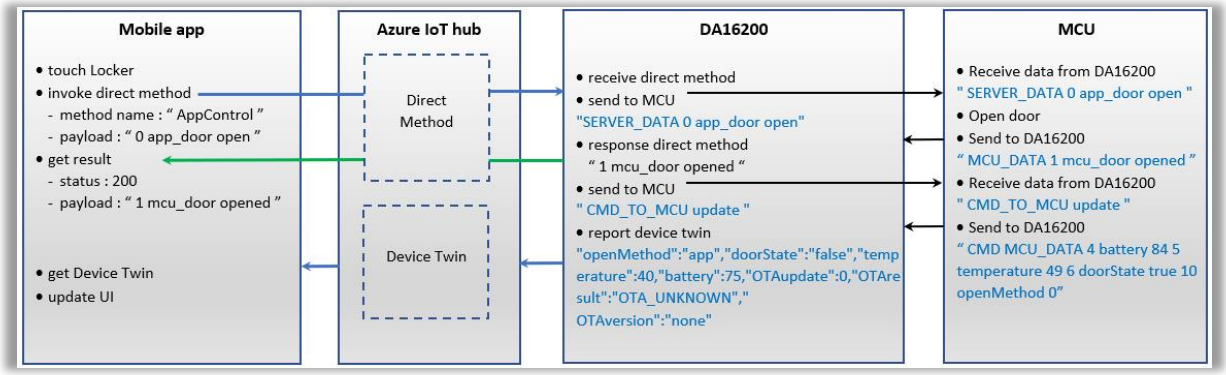

Figure 55: Message Flow of Opening Door

The operation for opening door on android app along with the Azure IoT console output displaying the device twin state is shown in Figure 56.

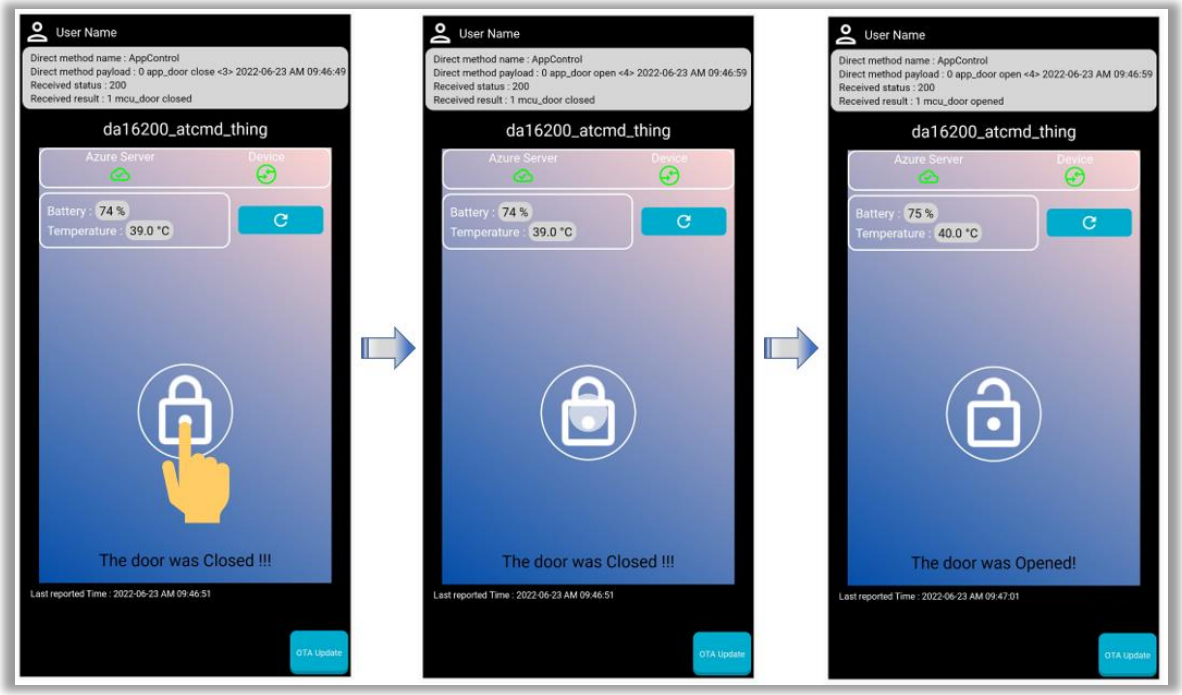

Figure 56: Operation of Opening Door in App

© 2024 Renesas Electronics

Figure 57 shows the state of the device twin on the Azure IoT Hub once the operation of opening door has been completed.

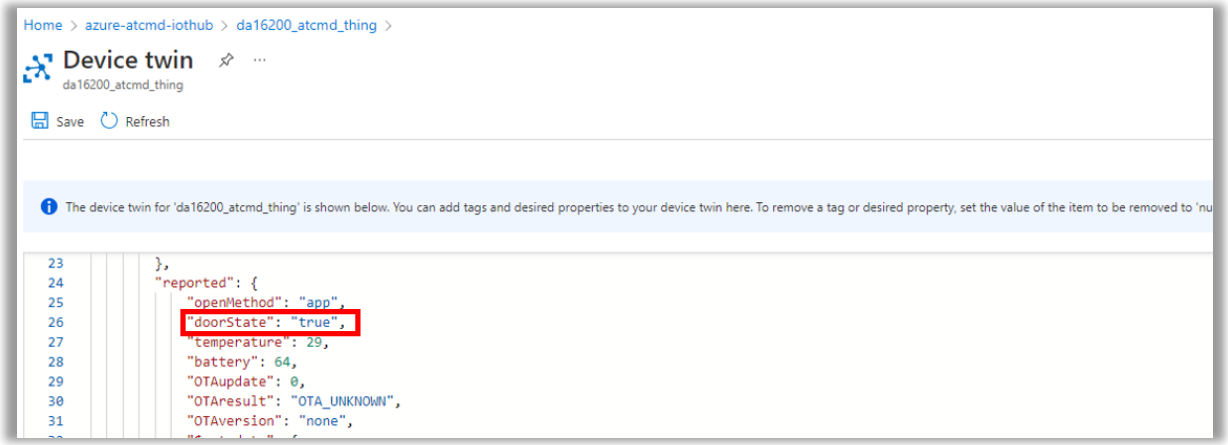

Figure 57: Device Twin When Door is Opened

The console log of the DA16200/DA16600 after the door is opened as below.

© 2024 Renesas Electronics

```
Info: [deviceMethodCallback:565] : method_name: AppControl, Command payload : 17, "0 app_door open"
dynamic subscription command ("app_door")
send "SERVER_DATA 0 app_door open" to MCU
================================================
argc num = 2
argv[0]: AT+AZU
argv[1]: CMD MCU_DATA 1 mcu_door opened
================================================
send "CMD_TO_MCU update" to MCU
================================================
argc num = 2
argv[0]: AT+AZU
argv[1]: CMD MCU_DATA 2 battery 75 3 temperature 40 4 doorState true 7 openMethod app
================================================
board feature : EVK
topicCount : 4
Count : 0, cmdNum = 2
mqtttype = 1
index(=3) matched
data type(shadow) = 0
call update sensor(need to be set variable): battery = 75
Count : 1, cmdNum = 3
mqtttype = 1
index(=2) matched
data type(shadow) = 2
call update sensor(need to be set variable): temperature = 35.000000
Count : 2, cmdNum = 4
mqtttype = 1
index(=1) matched
data type(shadow) = 1
call update sensor(need to be set variable): doorState = true
Count : 3, cmdNum = 7
mqtttype = 1
index(=0) matched
data type(shadow) = 1
call update sensor(need to be set variable): openMethod = app
release response
================================================================================
Info: [deviceMethodCallback:644] : Command response : "1 mcu_door opened"
registerShadow String data (openMethod) (app),
registerShadow String data (doorState) (true),
Info: [app_send_twins:305] : Report =>(method callback) {"openMethod":"app","doorState":"true",
"temperature":40,"battery":75,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN","OTAversion":"none","MCUOTAversion":"no
ne" }
```


© 2024 Renesas Electronics

4.3.2 Close Door

Message flows of **closing door** is shown in Figure 58.

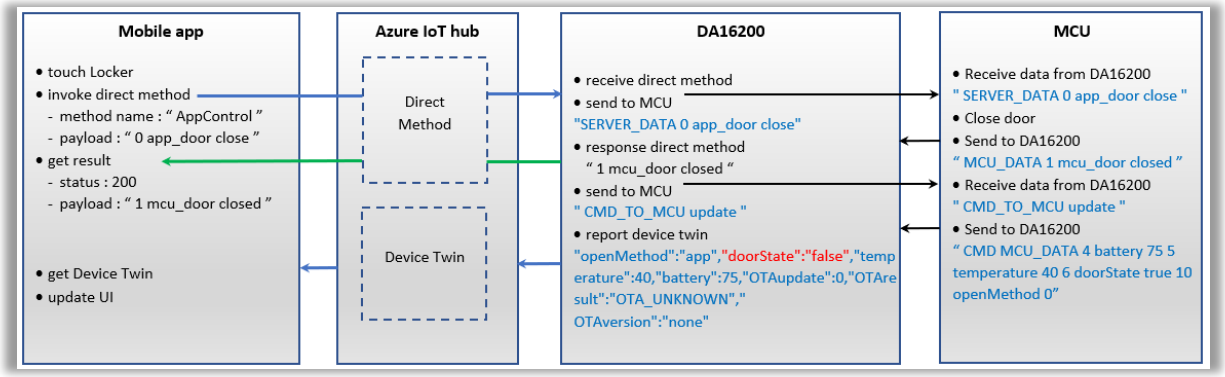

Figure 58: Message Flows of Closing Door

The operation for closing door on android app along with the Azure IoT console output displaying the device twin state is shown in Figure 59.

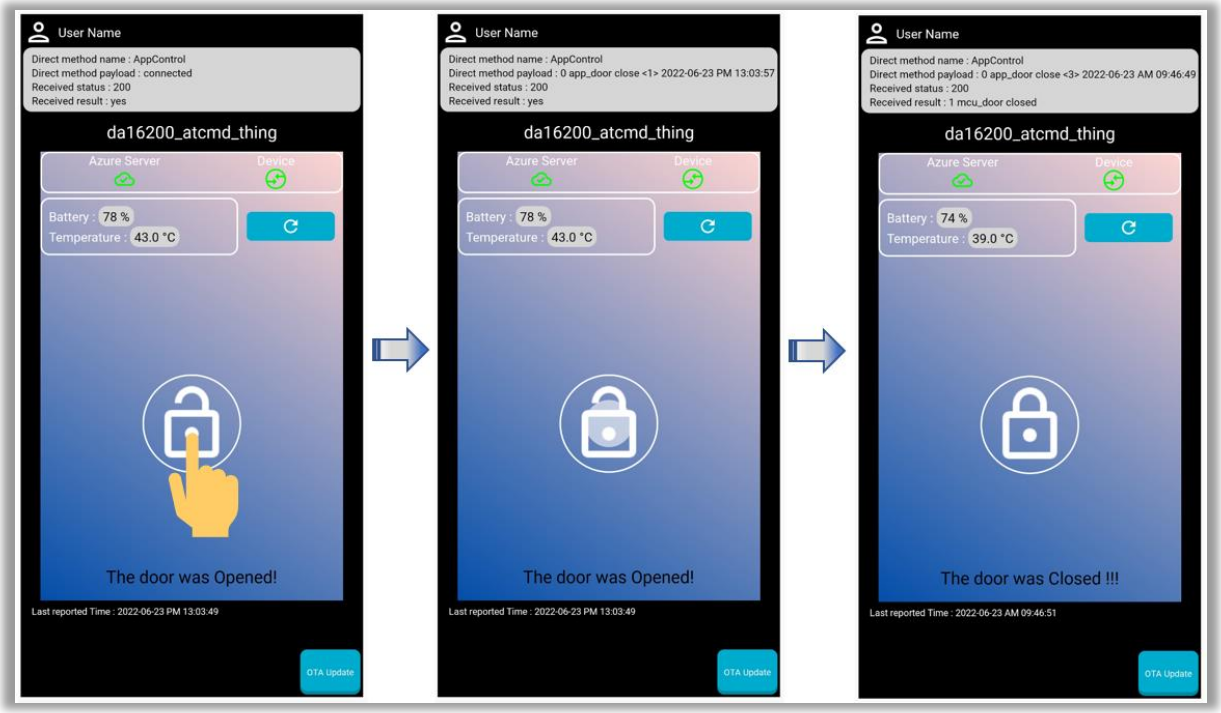

Figure 59: Operation of Closing Door in App

Figure 60 shows the device twin of Azure IoT Hub when the opening door operation is completed.

© 2024 Renesas Electronics

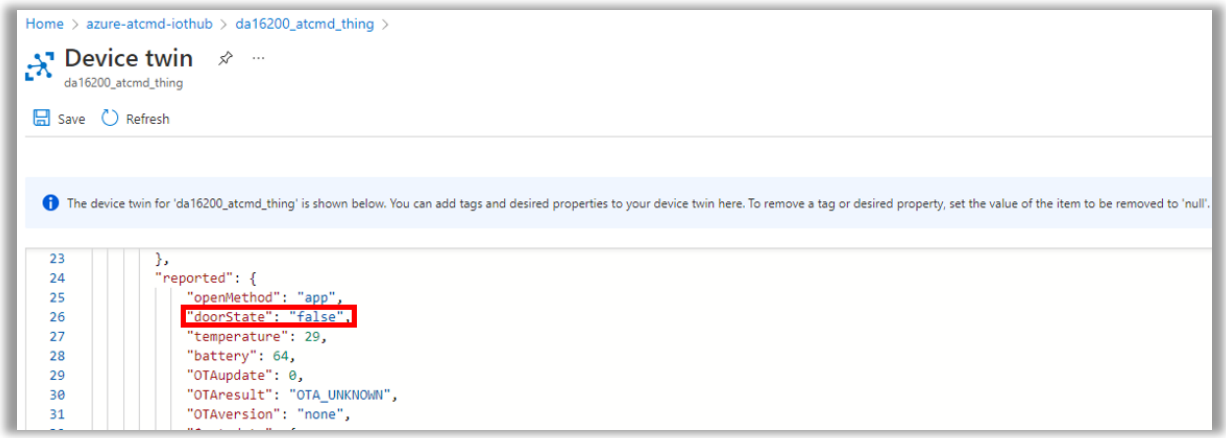

Figure 60: Device Twin When Door is Closed

The console output of the DA16200/DA16600 after the door is closed is shown below:

© 2024 Renesas Electronics

```
Info: [deviceMethodCallback:565] : method_name: AppControl, Command payload : 18, "0 app_door close"
dynamic subscription command ("app_door")
send "SERVER_DATA 0 app_door close" to MCU
================================================
argc num = 2
argv[0]: AT+AZU
argv[1]: CMD MCU_DATA 1 mcu_door closed
================================================
send "CMD_TO_MCU update" to MCU
================================================
argc num = 2
argv[0]: AT+AZU
argv[1]: CMD MCU_DATA 4 battery 79 5 temperature 44 6 doorState false 10 openMethod app
================================================
board feature : EVK
topicCount : 4
Count : 0, cmdNum = 2
mqtttype = 1
index(=3) matched
data type(shadow) = 0
call update sensor(need to be set variable): battery = 79
Count : 1, cmdNum = 3
mqtttype = 1
index(=2) matched
data type(shadow) = 2
call update sensor(need to be set variable): temperature = 44.000000
Count : 2, cmdNum = 4
mqtttype = 1
index(=1) matched
data type(shadow) = 1
call update sensor(need to be set variable): doorState = false
Count : 3, cmdNum = 7
mqtttype = 1
index(=0) matched
data type(shadow) = 1
call update sensor(need to be set variable): openMethod = app
release response
================================================================================
Info: [deviceMethodCallback:644] : Command response : "1 mcu_door closed"
registerShadow String data (openMethod) (app),
registerShadow String data (doorState) (false),
Info: [app_send_twins:305] : Report =>(method callback) {"openMethod":"app","doorState":"false",
"temperature":44,"battery":79,"OTAupdate":0,"OTAresult":"OTA_UNKNOWN","OTAversion":"none","MCUOTAversion":"no
ne"}
```


© 2024 Renesas Electronics

5 OTA Update

Over the Air (OTA) is the process of updating the DA16200/DA16600 firmware image through Wi-Fi using an Azure services storage account.

To set up OTA update:

1. Create a storage account.
2. Create a container.
3. Upload a DA16200/DA16600 image file to the container.
4. Add the OTA URL to the device twin.
5. Execute the OTA update.

5.1 Create Storage Account

For OTA update, an Azure services storage account is required.

1. Click **Storage Accounts**.

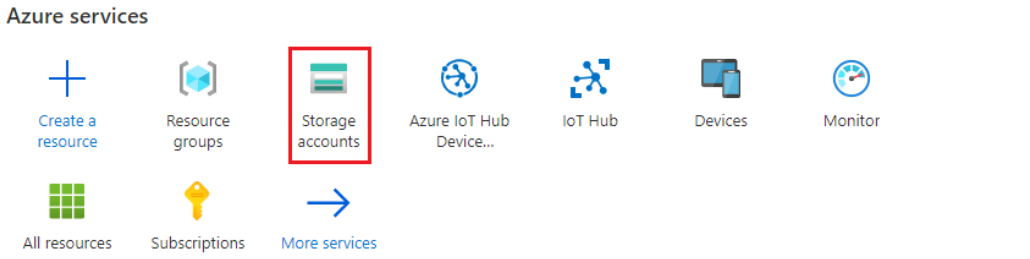

Figure 61: Storage Accounts

2. Click **Create**.

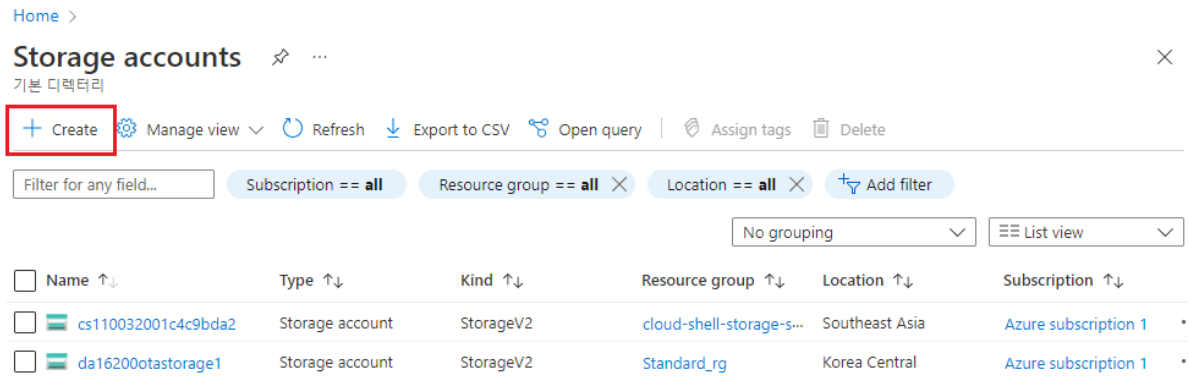

Figure 62: Create Storage Account

© 2024 Renesas Electronics

3. Configure the **Resource group**, **Storage account name**, and **Region**, and then click **Review + create**.

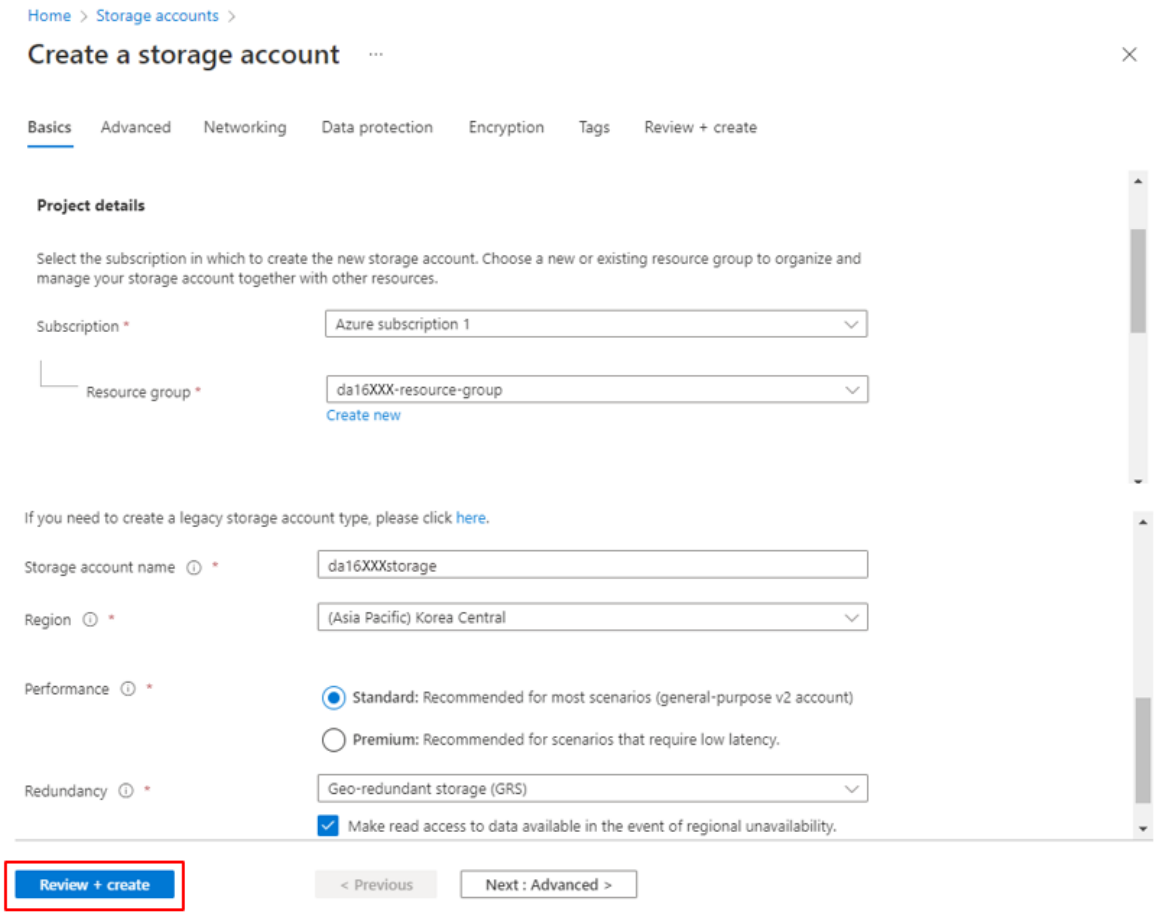

Figure 63: Configure Storage Account

4. Once the storage account validation passes, click **Create**.

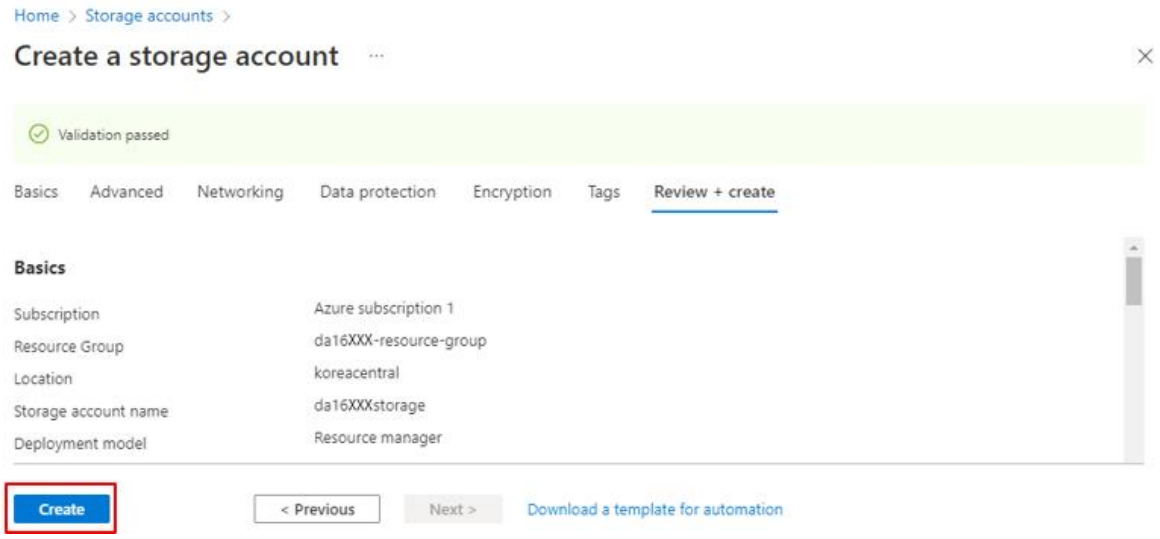

Figure 64: Storage Account Validation Done

5. Once the storage account deployment is complete, click **Go to resource**.

© 2024 Renesas Electronics

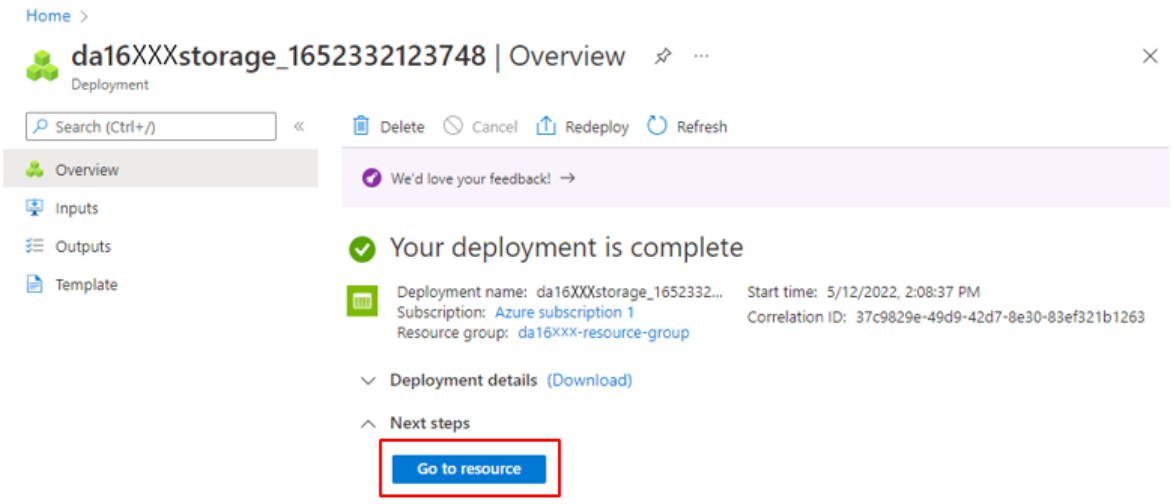

Figure 65: Storage Account Deployment Done

6. The created storage account is as shown in Figure 66.

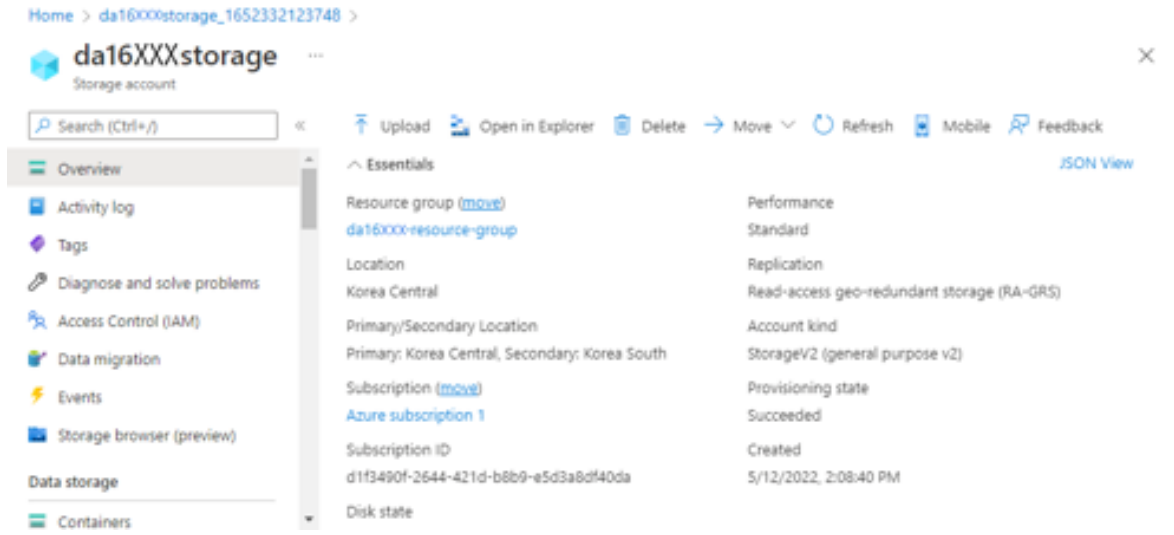

Figure 66: Created Storage Account

5.2 Create Container

A container must be created in the Azure services storage account to store the DA16200/DA16600 image file for OTA update.

To create a container:

1. In the created storage account, click **Containers**.

© 2024 Renesas Electronics

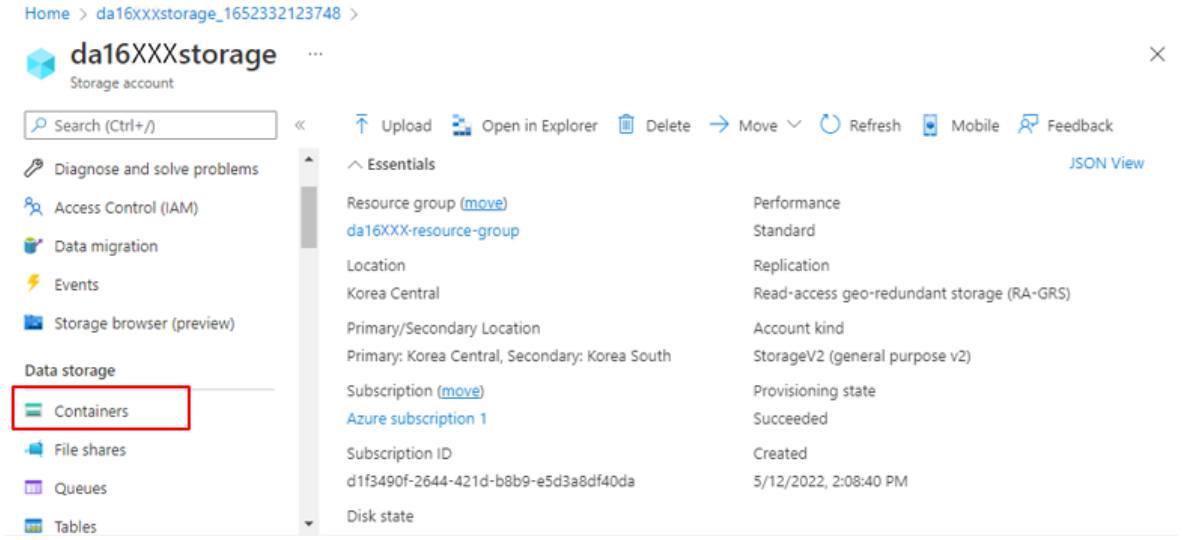

Figure 67: Containers in Storage Account

2. Click **Container** to add container.

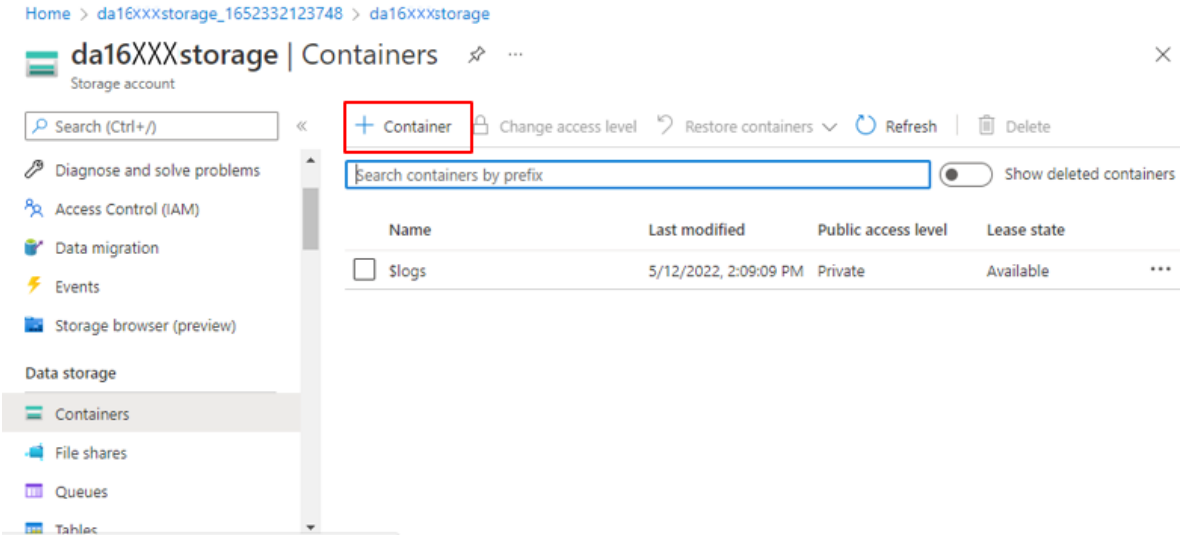

Figure 68: Add Container

© 2024 Renesas Electronics

3. Under the **New container** section, enter a name (for example, *"da16XXX-img-container"*) and select **Container (anonymous read access for containers and blobs)** as the Public access level, and then click **Create**.

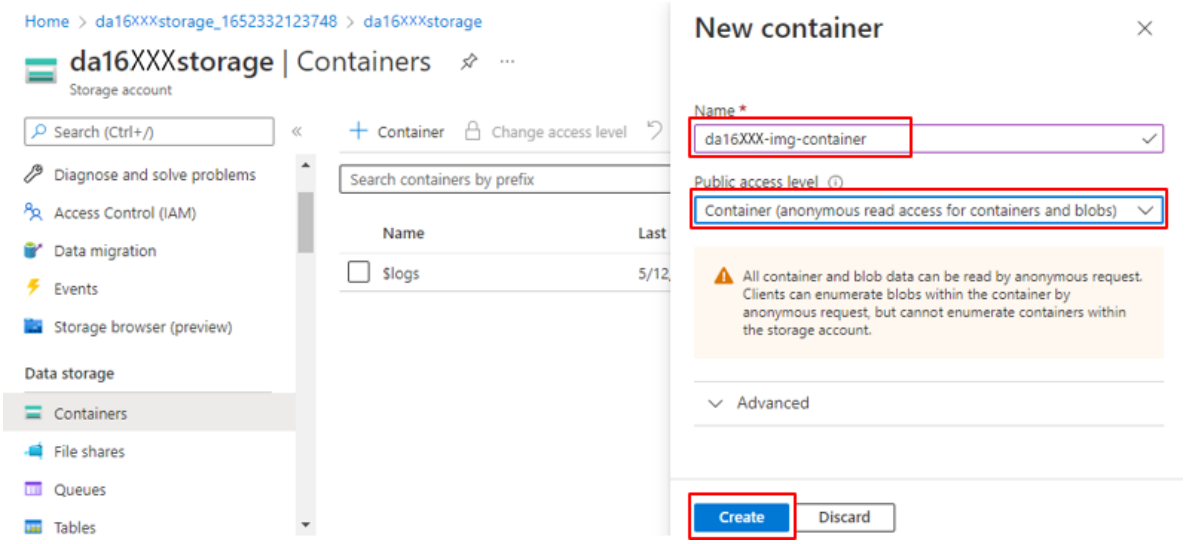

Figure 69: Container Details

4. The created container appears in the containers list.

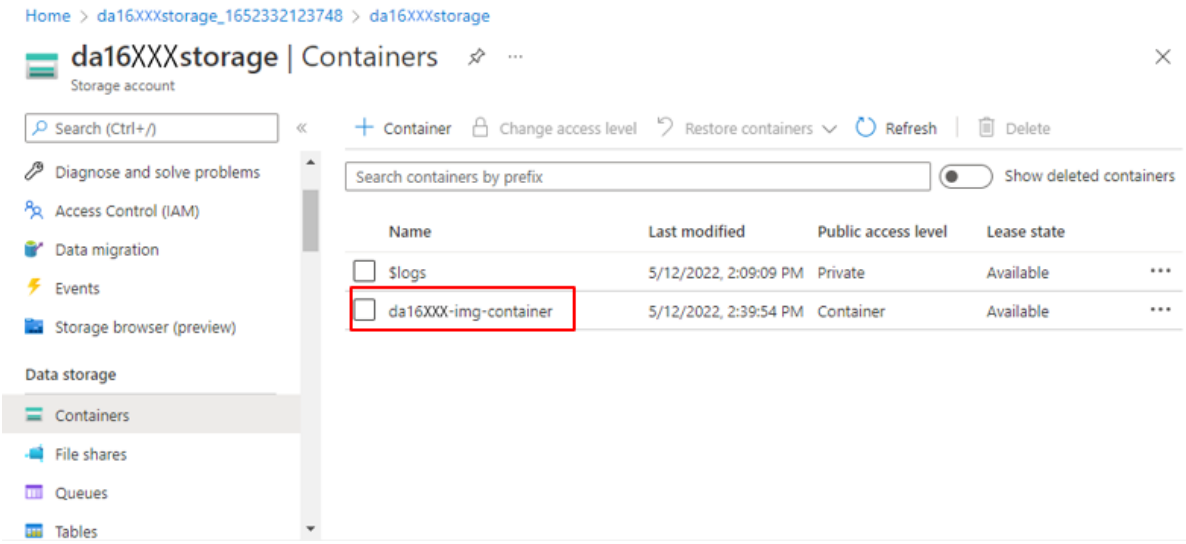

Figure 70: Created Container

5.3 Upload Firmware Image File

To upload the DA16200/DA16600 firmware image file to the container:

1. Click the created container in the containers list.

© 2024 Renesas Electronics

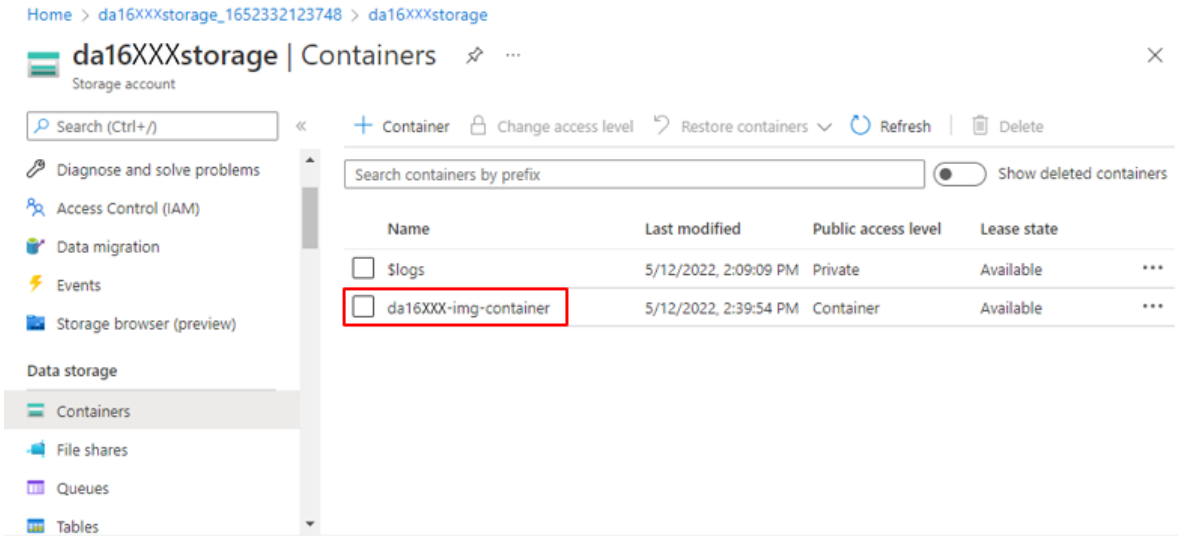

Figure 71: Container for Uploading Images

2. Click **Upload** to upload the DA16200/DA16600 image file.

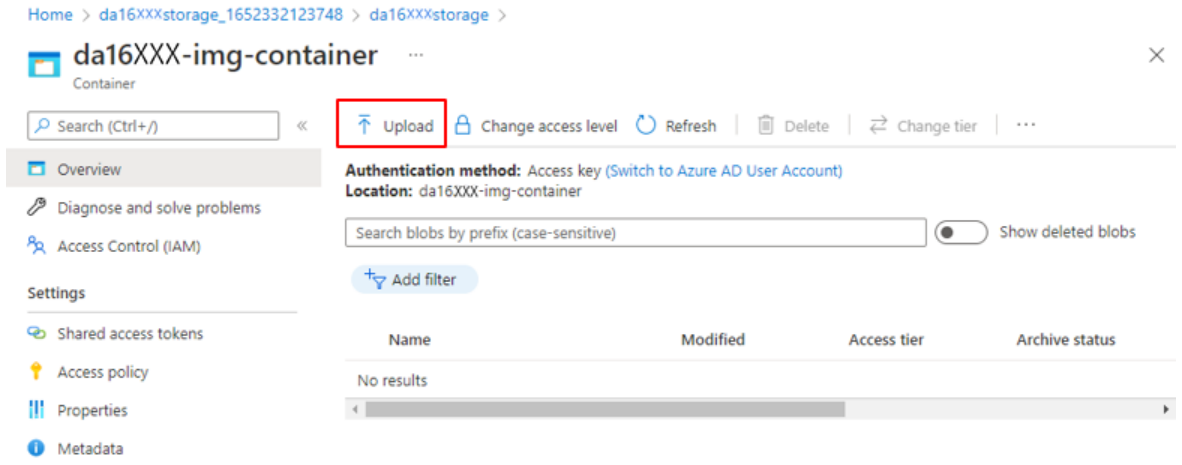

Figure 72: Upload Image

3. Under the **Upload blob** section, in Files navigate to the DA16200/DA16600 image file saved in local directory:
   - In case of DA16200
     - **DA16200_FRTOS-GEN01.img**: (must be uploaded as this name)
   - In case of DA16600
     - **DA16600_FRTOS-GEN01.img**: (must be uploaded as this name)
     - **DA16600_BLE_OTA.img**: (must be uploaded as this name)
4. Click **Upload** and then uploaded image files are appeared as shown in Figure 74.

© 2024 Renesas Electronics

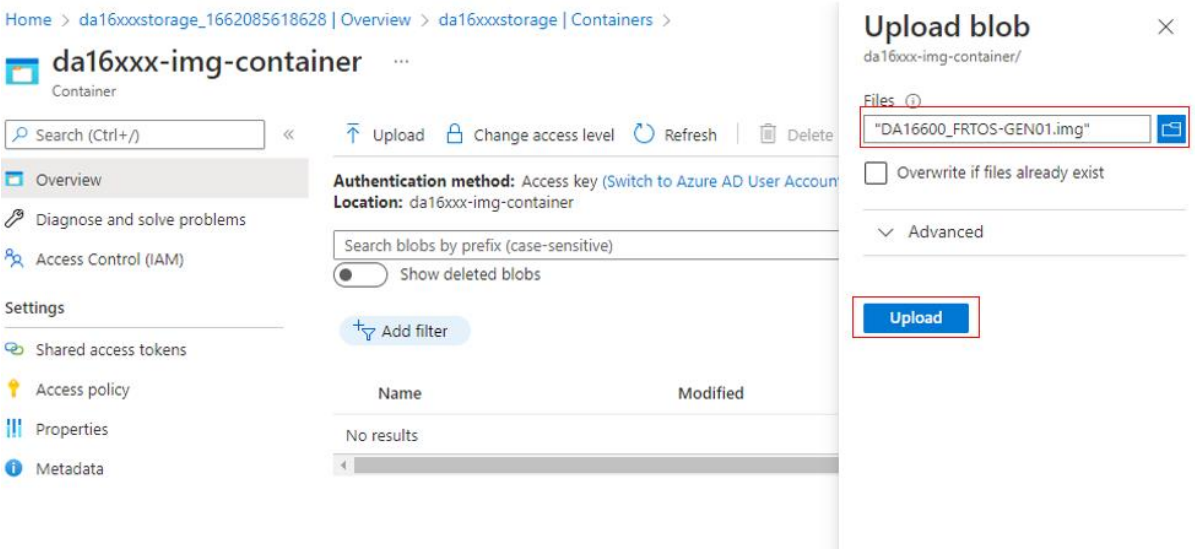

Figure 73: Navigate to Image

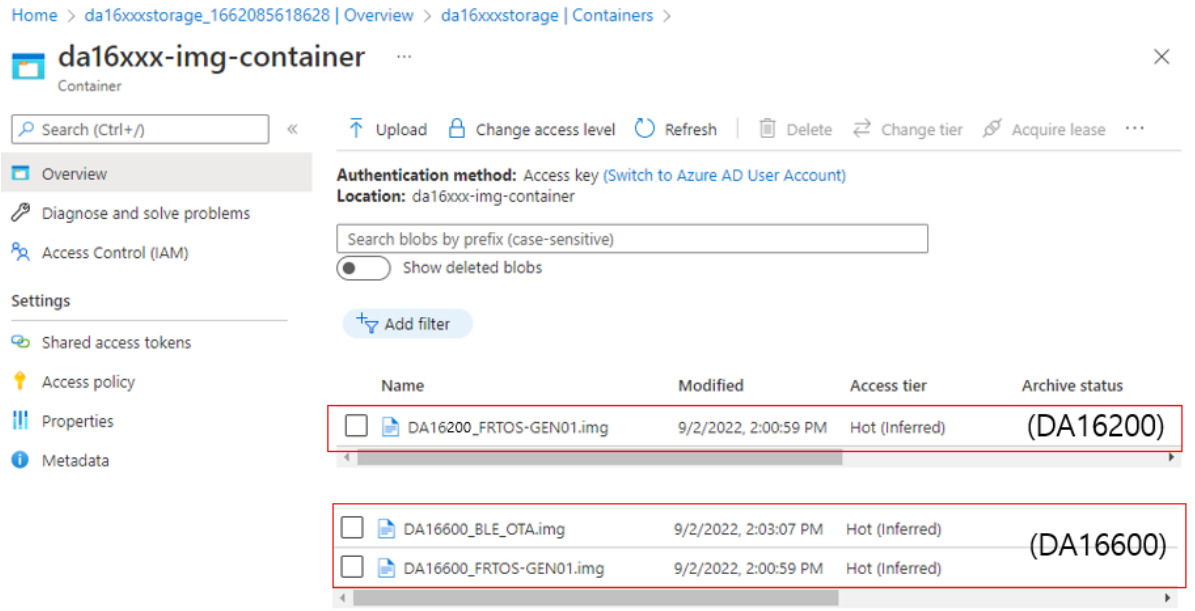

Figure 74: Uploaded Image

5. Choose **Properties**, and then click the Copy icon as shown in Figure 75 after the file is uploaded. Save the **URL** to configure the Azure Device Twin later.

© 2024 Renesas Electronics

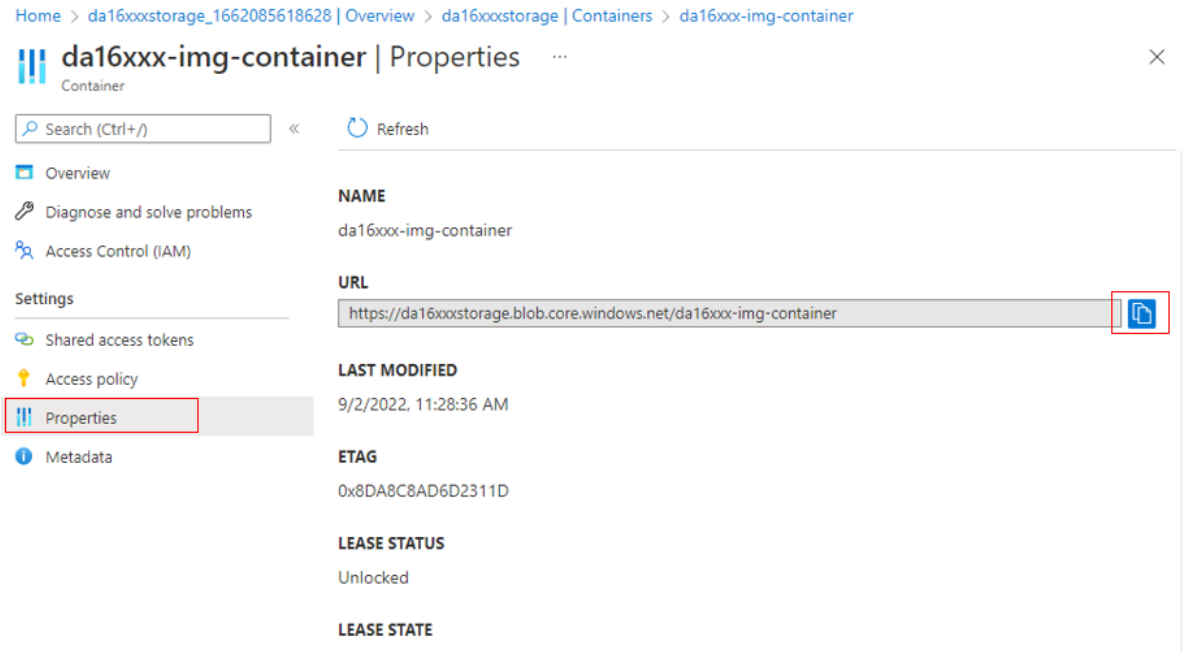

Figure 75: Copy Container URL

5.4 Add OTA URL to Device Twin

For OTA update, a configuration must be added to the IoT Hub that sets the OTA URL in the Device Twin.

To add configuration in Device Twin for the OTA URL:

1. In the IoT hub, for example *da16XXX-iothub*, click **Configurations**, and then select **Add Device Configuration**.

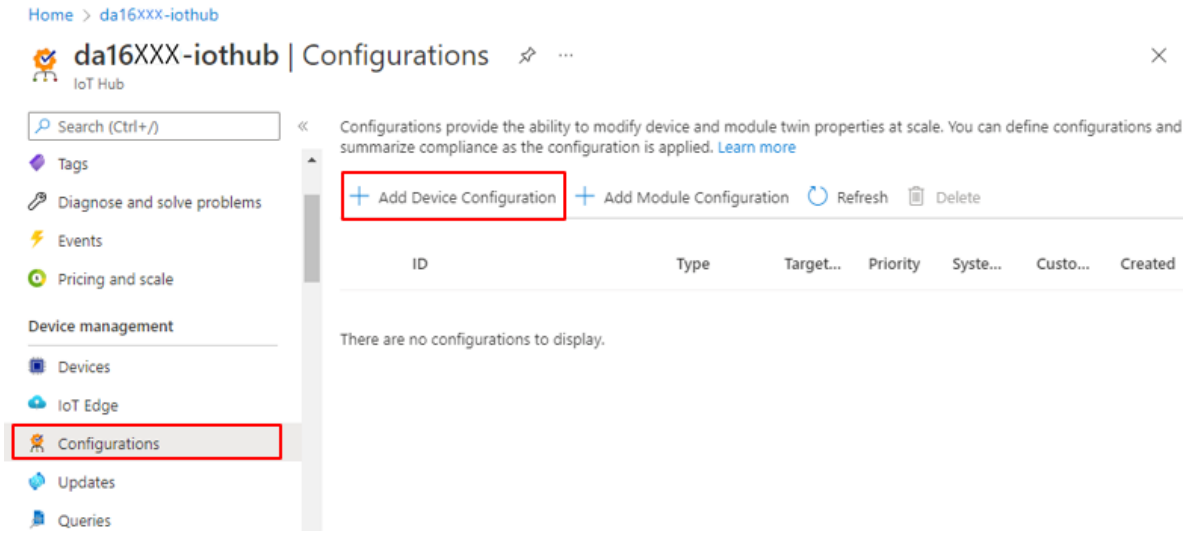

Figure 76: Add Device Configuration

© 2024 Renesas Electronics

2. Under the **Name and Label** section, follow the steps below:
   a. In **Name**, add a name, for example "*da16XXX-ota-version1*".
   b. Under Label, create a label by adding name and value, for example "*da16XXX-ota-label*", "*1.0*".

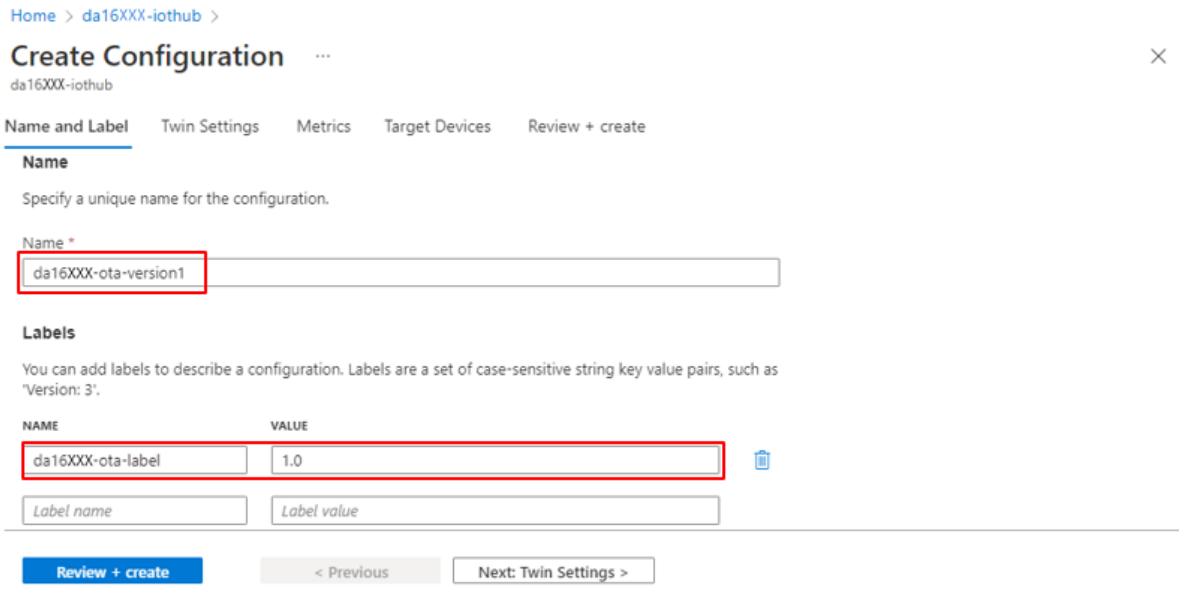

Figure 77: Name and Label Section

3. Under the Twin Settings section, follow the steps below:
   a. Set Device Twin Property to: "properties.desired.fwupdate".
   b. In the Device Twin Property Content field, which defines the version and URL, add a JSON formatted string.

      For example: {"version":"1.0","url": "https://da16XXXotastorage1.blob.core.windows.net/da16XXX-img-ontainer"}

© 2024 Renesas Electronics

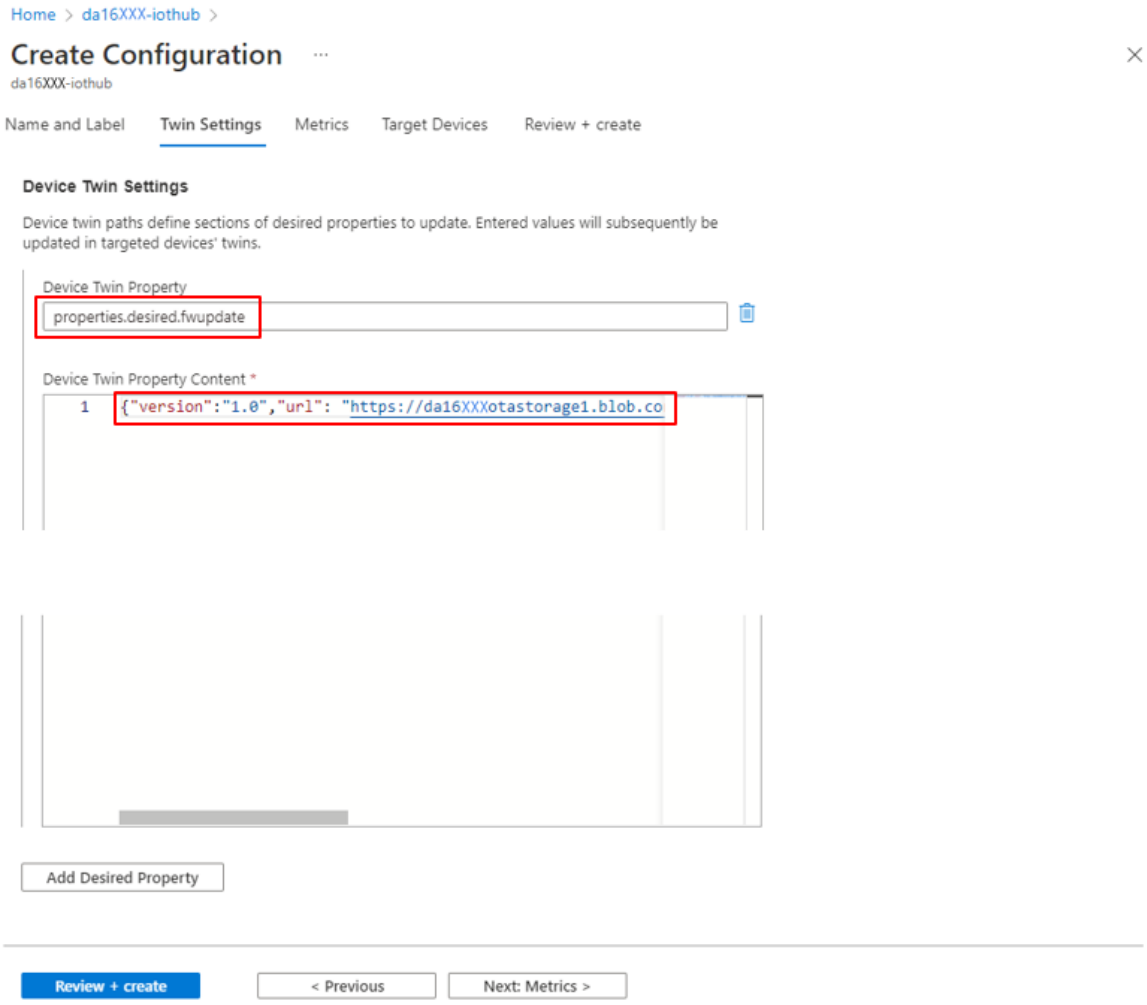

Figure 78: Create Configuration

4. Under the **Target Devices** section, follow the step below:
   a. Set **Priority**, for example "*10*".
   b. Set **Target Condition** to match the **Device ID** in the devices of Azure IoT Hub. For example, *"deviceid='da16200-device-1'"*.

© 2024 Renesas Electronics

c. Click **View Devices** to verify the device Id.

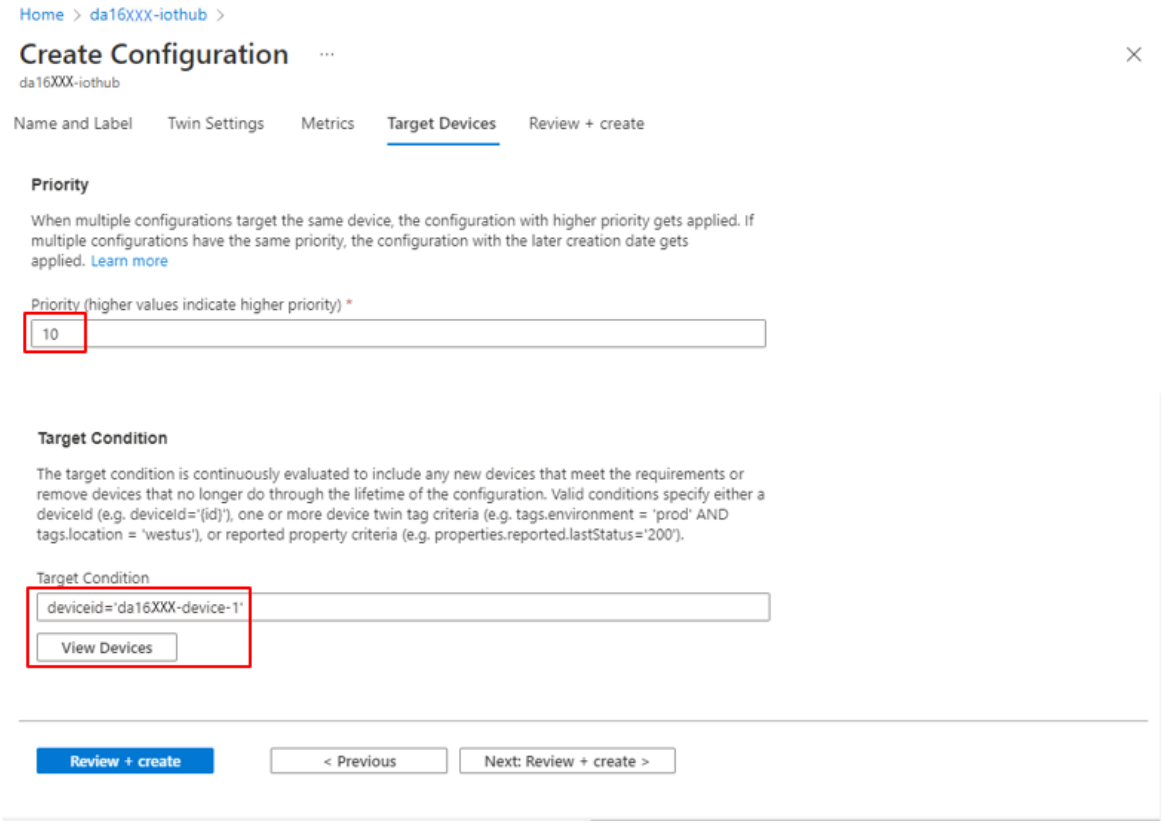

Figure 79: Target Devices Section

5. Under the **Review + create** section, follow the steps below:
   a. Verify the validation is passed.
   b. Review the configuration that is shown in the text box.
   c. Click **Create** to create the configuration.

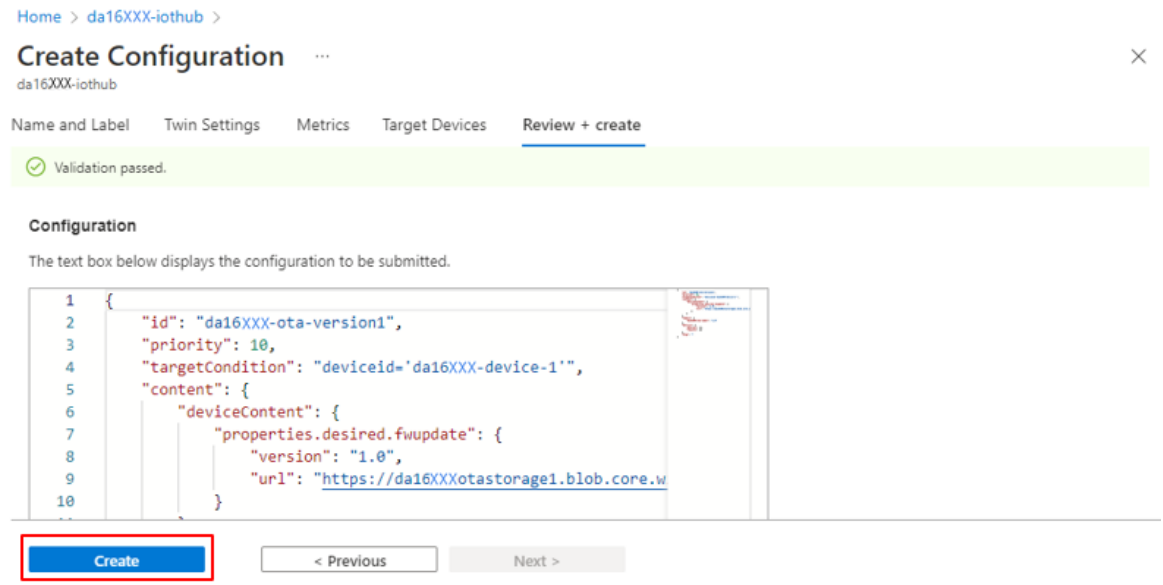

Figure 80: Review + Create Section

6. The created configuration is shown in Figure 81.

© 2024 Renesas Electronics

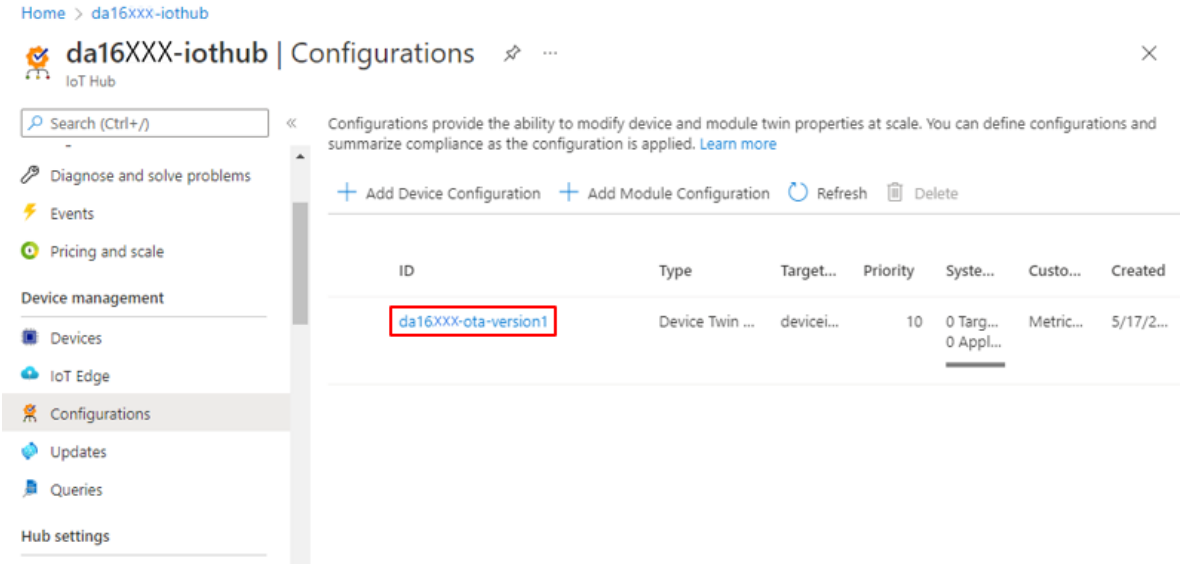

Figure 81: Created Device Twin Configuration

7. The operation of the Android app is shown in Figure 82.

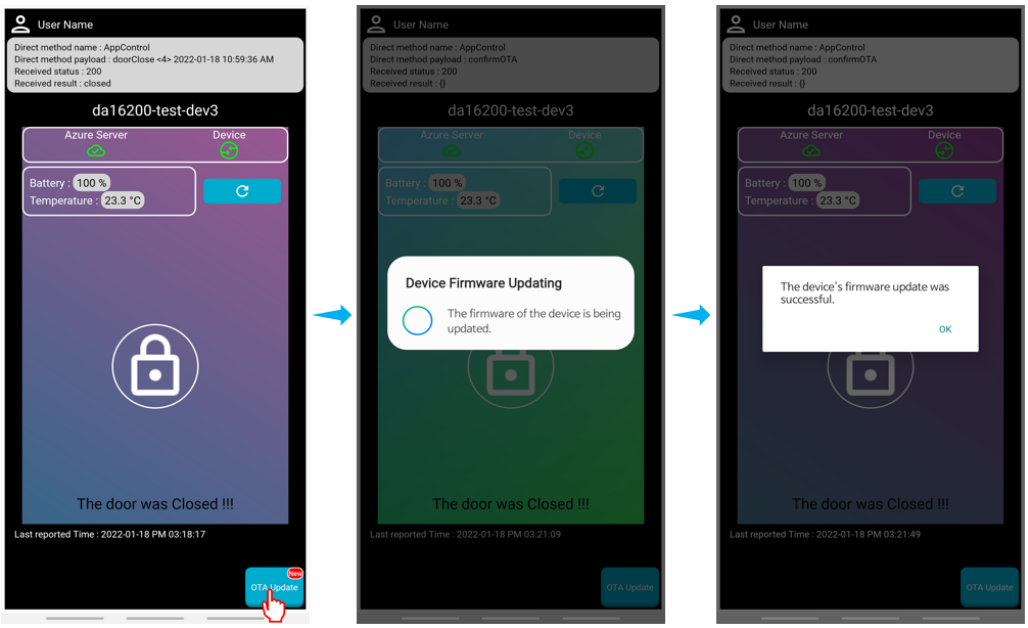

Figure 82: OTA Update Operation

8. The message flow of OTA update is shown in Figure 83.

© 2024 Renesas Electronics

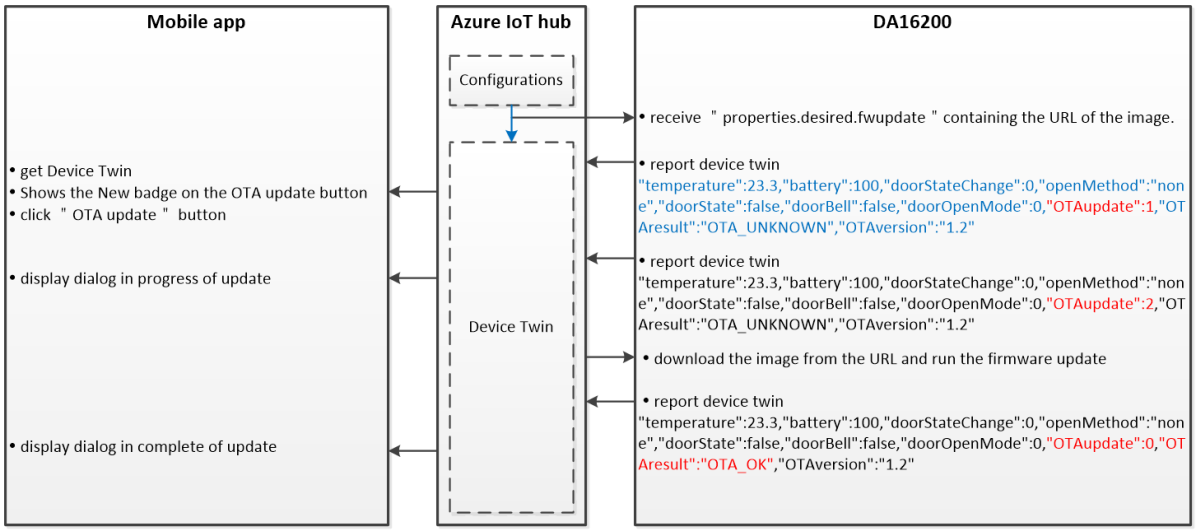

Figure 83: Message Flows of OTA Update

9. The log from DA16200/DA16600 during OTA process as shown in Figure 84.

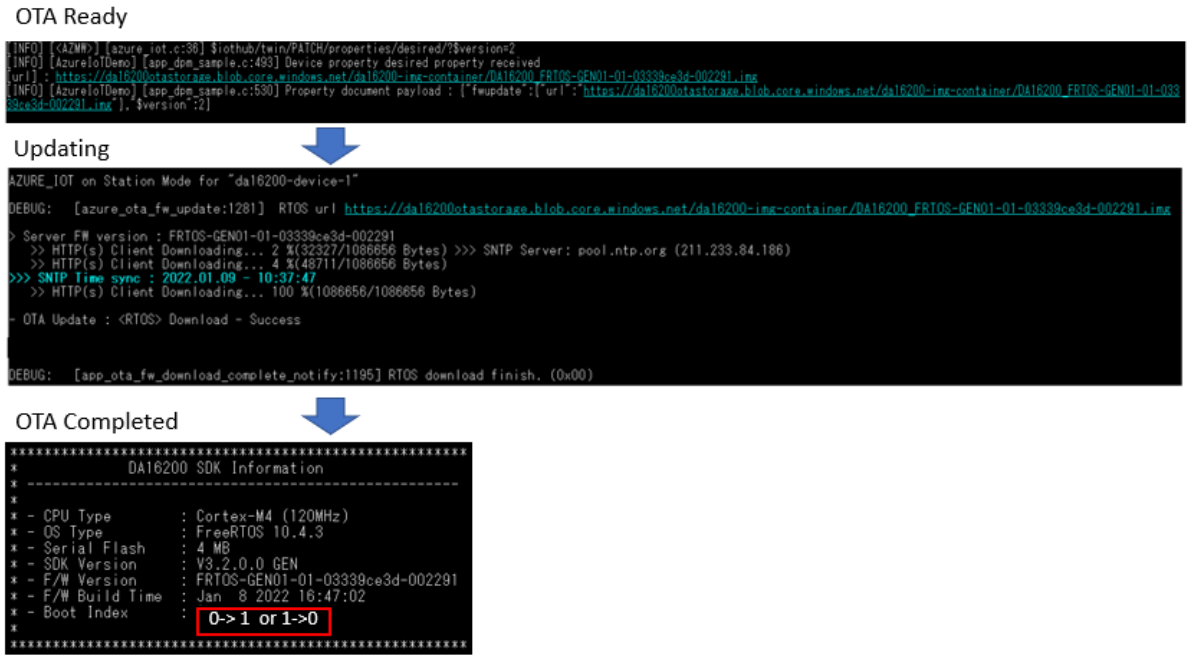

Figure 84: OTA Process in DA16200/DA16600

5.5 OTA Update of MCU Firmware

The Azure AT command package provides the ability to perform a safe and secure OTA update of the MCU firmware. Figure 85 shows the process when using the mobile app and the Azure server to update the MCU firmware through the DA16200/DA16600.

Note that the DA16200/DA16600 detects whether the firmware image is for the MCU or for the DA16200/DA16600 depending on the prefix of the firmware image name. If the firmware image is for the MCU, then the firmware image name should be prefixed with “mcu” for example, **mcu**_firmware_20220101.img.

© 2024 Renesas Electronics

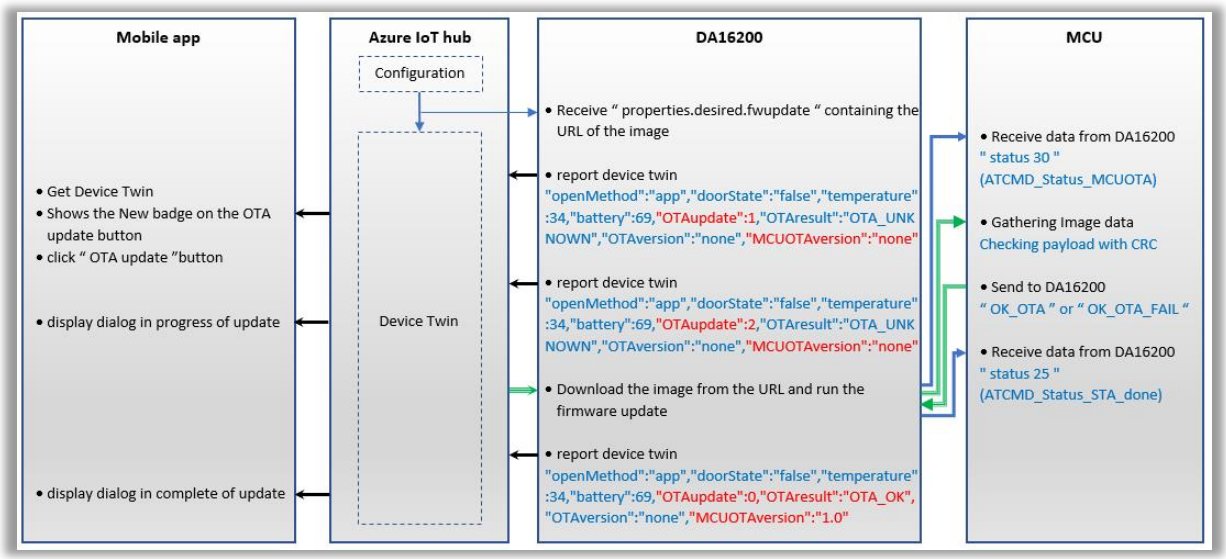

Figure 85: Block Diagram of MCU OTA

An example MCU firmware image for testing OTA is available in the Azure AT command package:

\MCU\bin\mcu_azure.bin.

© 2024 Renesas Electronics

Appendix A Azure IoT Hub Device Provisioning Service

Device Provisioning Service (DPS) is a process for registering the DA16200/DA16600 device on the Azure IoT Hub.

To set up DPS, follow the steps below:

1. Create an IoT Hub .
2. Manage Enrollments.
3. Configure Registration ID Parameters in DA16200/DA16600 SDK.

A.1 Create Azure IoT Hub

DPS is required to register devices on the Azure IoT Hub. The following steps show how to set up DPS and link it with the Azure IoT Hub.

To create DPS:

1. Click **Azure IoT Hub Device Provisioning Service**.

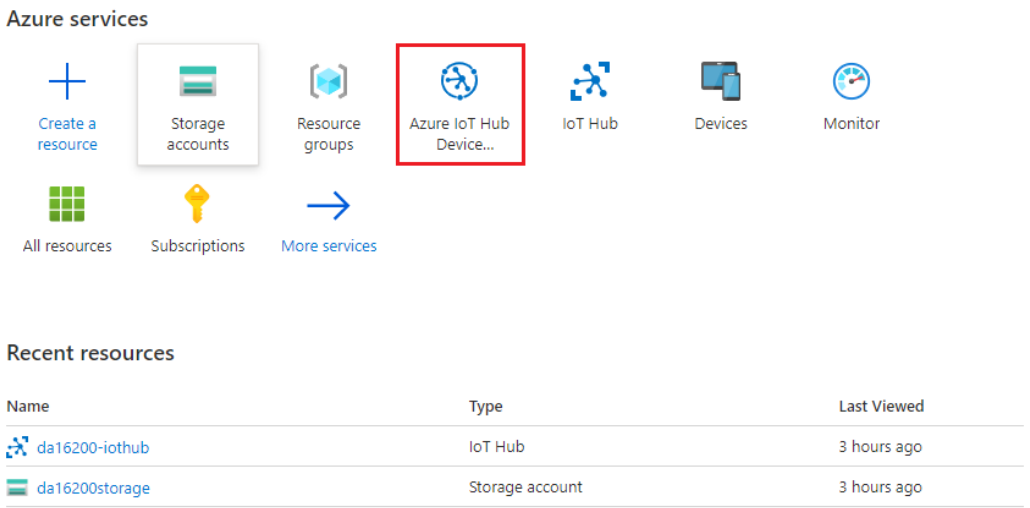

Figure 86: Azure IoT Hub Device Provisioning Service

2. Click **Create**.

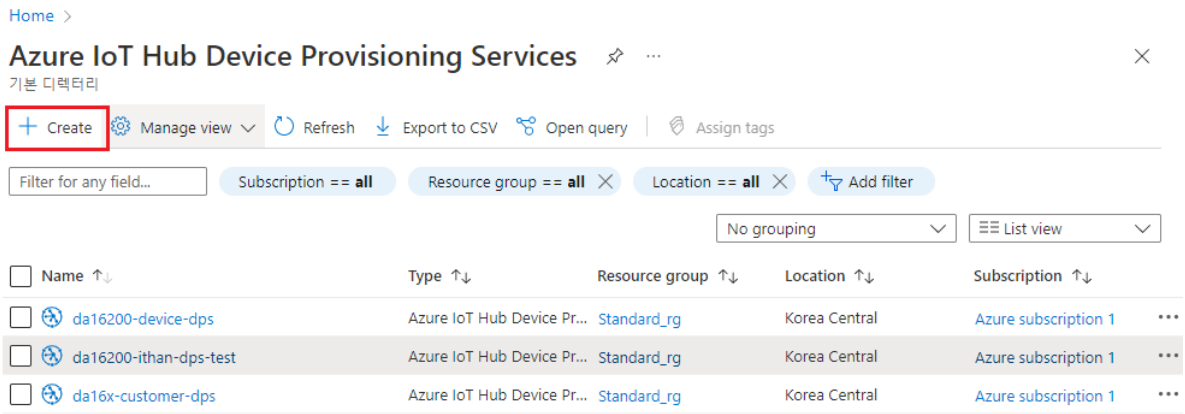

Figure 87: Create DPS

© 2024 Renesas Electronics

3. Configure the **Resource group**, **Name** and **Region** as shown in Figure 88, and then click **Review + create**.

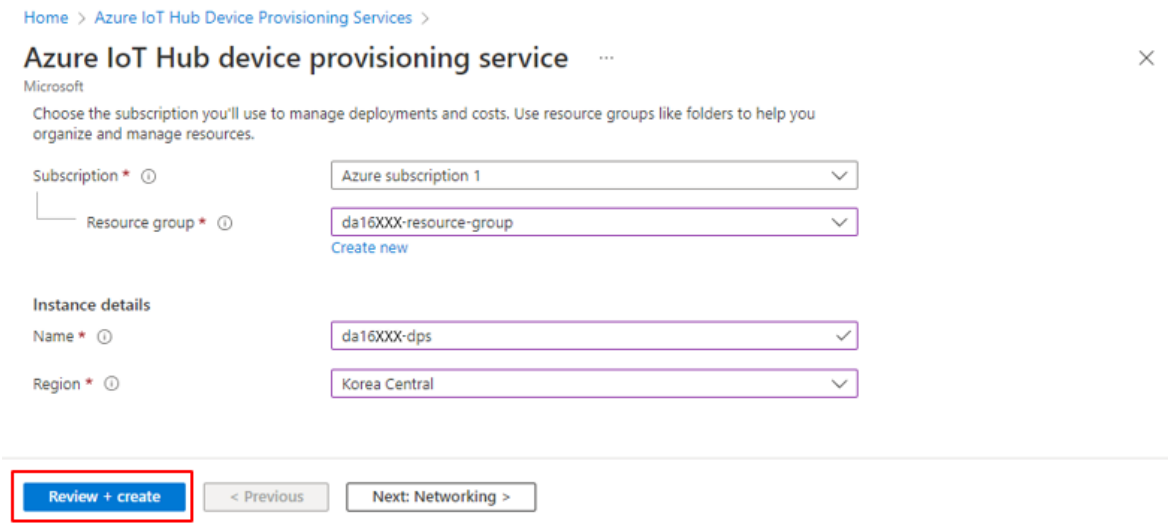

Figure 88: Configure Resource Group, Name, and Region

4. After the validation is passed, click **Create**.

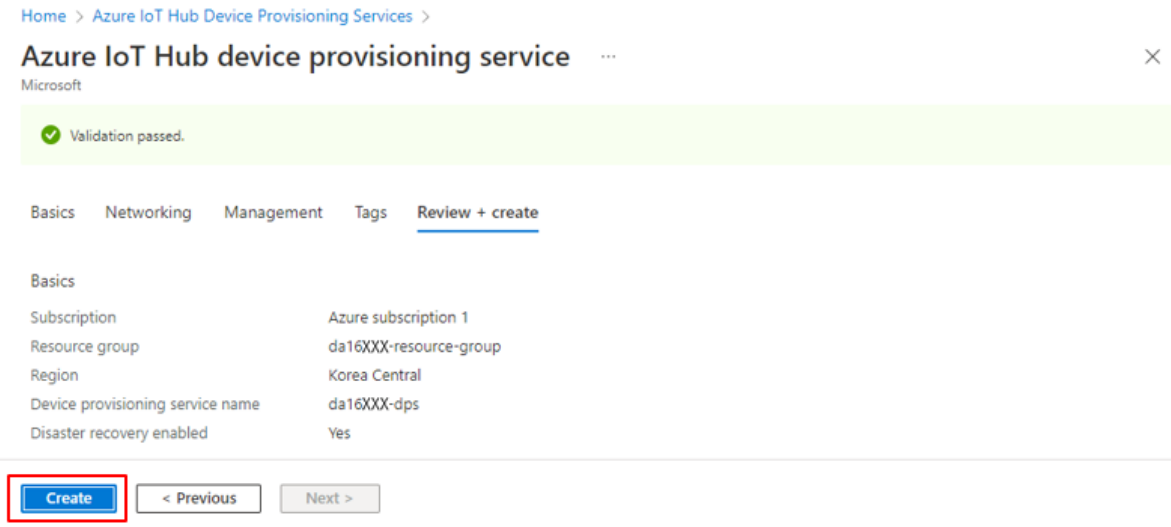

Figure 89: Create Button

© 2024 Renesas Electronics

5. After the deployment of DPS is complete, click **Go to resource**.

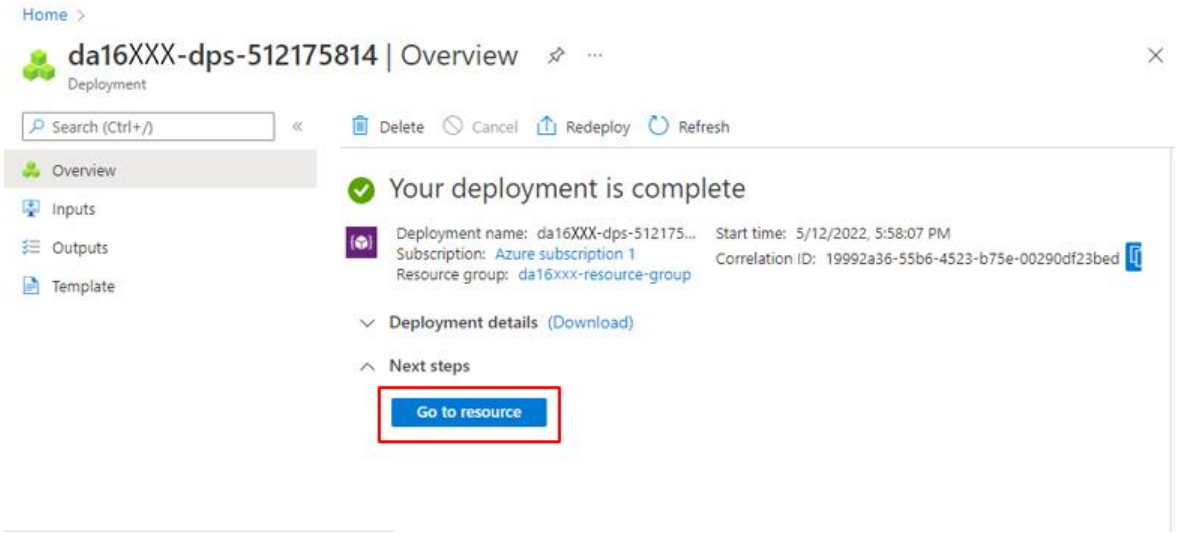

Figure 90: Go to Resource

6. Select **Linked IoT hubs** to link the created DPS resource to the Azure IoT Hub.

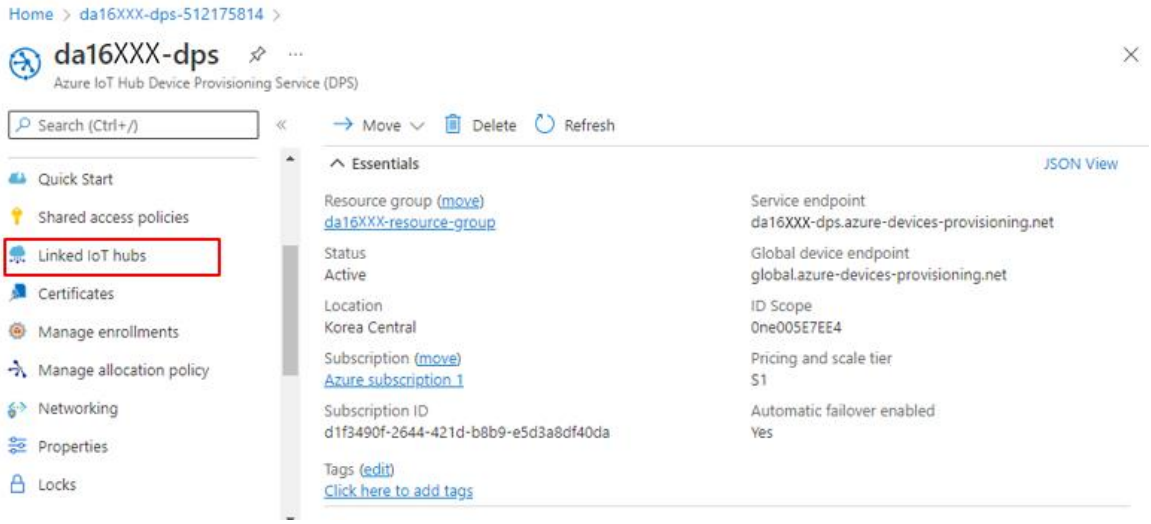

Figure 91: Linked IoT Hubs

7. Click **Add** to link the Azure IoT Hub.

© 2024 Renesas Electronics

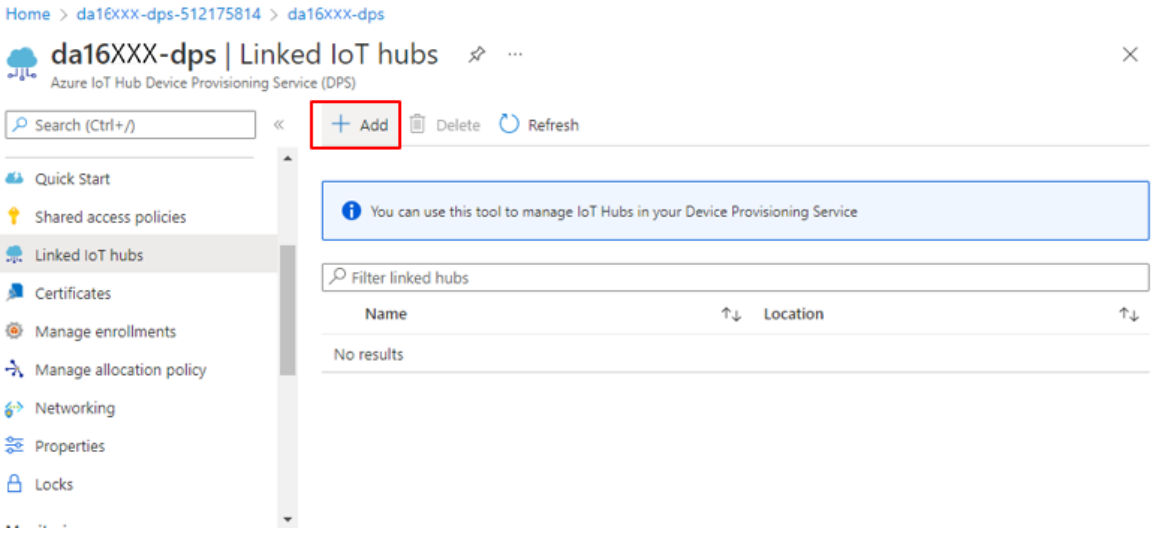

Figure 92: Add Link to Azure IoT Hub

8. Under **Add link to IoT hub** section, select the **IoT hub** and **Access Policy** as shown in Figure 93, and click **Save**.

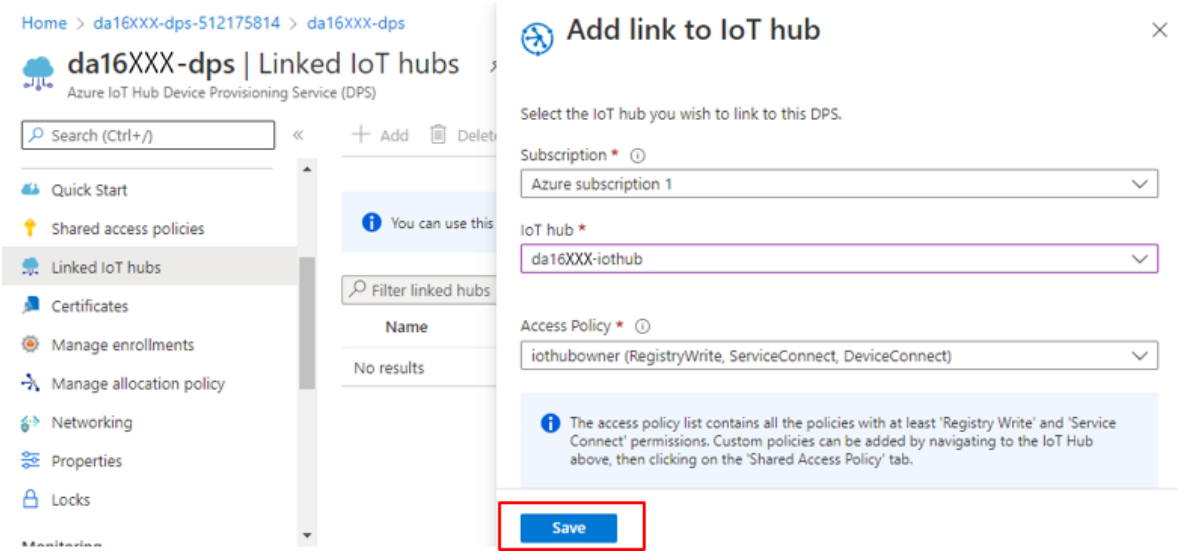

Figure 93: Configure Link

The IoT Hub that is linked to the DPS appears in the list.

© 2024 Renesas Electronics

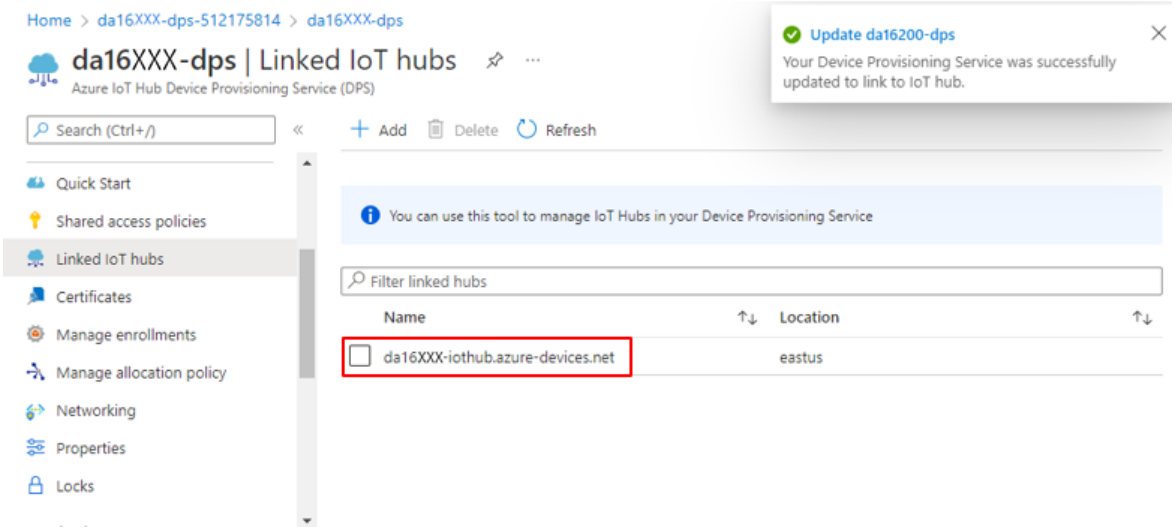

Figure 94: IoT Hub Linked to DPS

A.2 Manage Enrollments

Manage Enrollments is the process of registering a device with the Azure IoT Hub.

To manage enrollments on the Azure IoT Hub:

1. Go to the created DPS and click **Manage enrollments**, and then select **Add individual enrollment**.

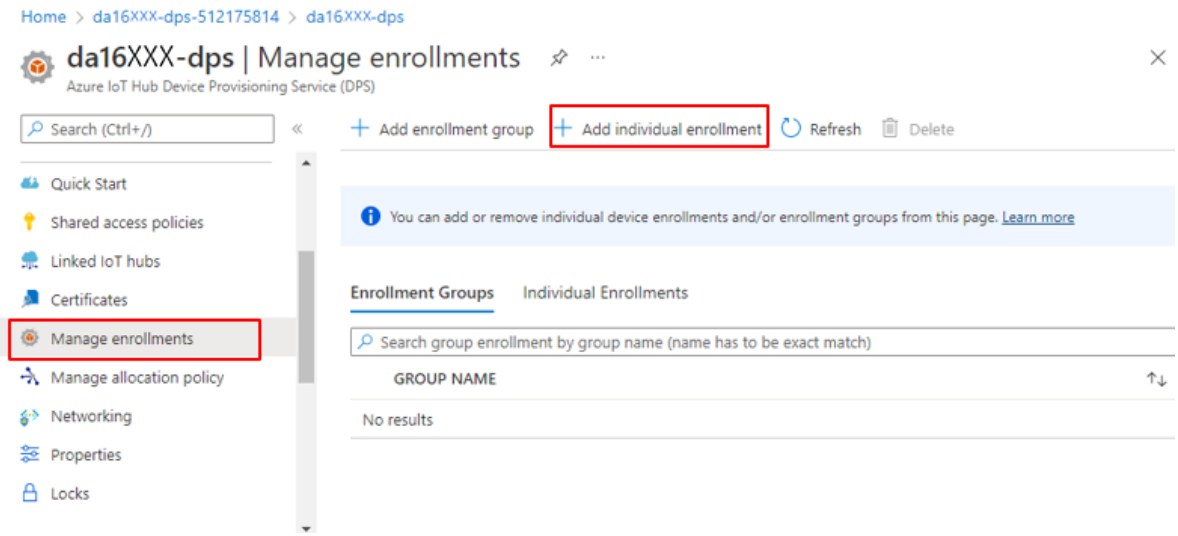

Figure 95: Manage Enrollments

2. In the **Add enrollment**, follow the steps below, and then click **Save**:
   a. In **Mechanism**, select **Symmetric Key**.
   b. In **Registration ID**, enter value for the device, for example *da16200-dps-dev1*.

© 2024 Renesas Electronics

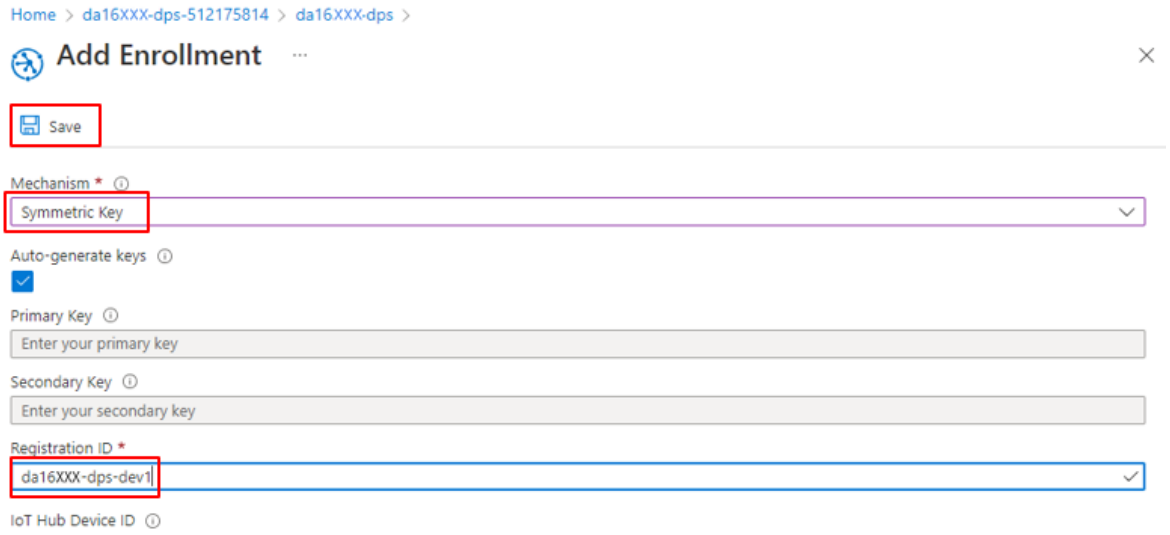

Figure 96: Add Enrollment

3. Click **Individual Enrollments** and the created registration ID appears in the list.

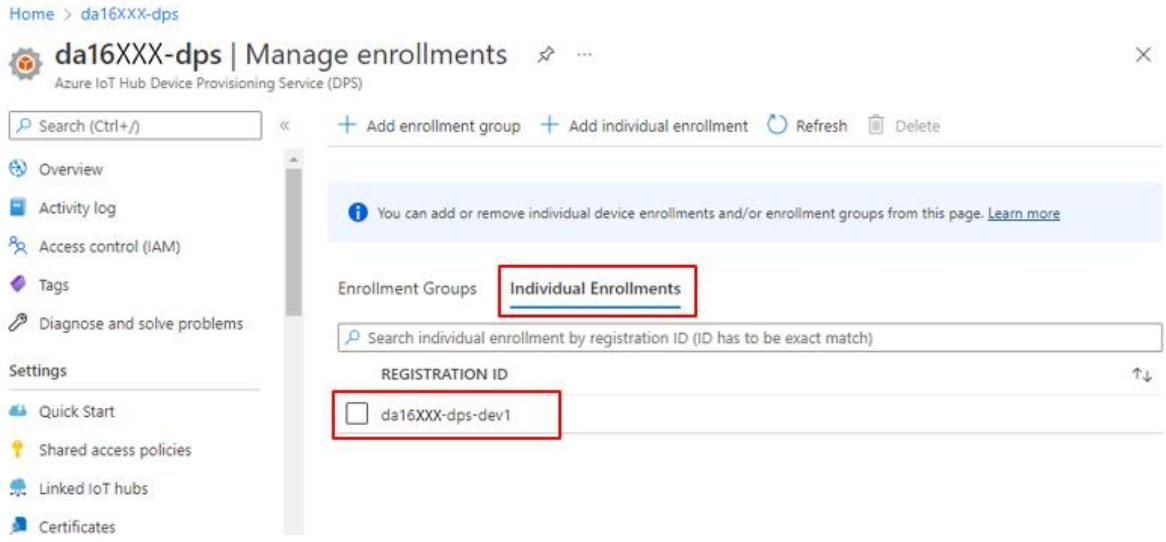

Figure 97: Individual Enrollments

A.3 Configure Registration ID Parameters in DA16200/DA16600 SDK

To enable the Device Provisioning Service, define `ENABLE_AZURE_DPS_EXAMPLE` as `1` in the e²studio.

To configure DPS parameters in the DA16200/DA16600 SDK:

1. In the DA16200/DA16600 SDK, edit app_thing_manager.h and define `APP_USER_MY_THING_NAME` as the **DPS Registration ID**.

   For example: `#define APP_USER_MY_THING_NAME "da16XXX-dps-dev1"`

© 2024 Renesas Electronics

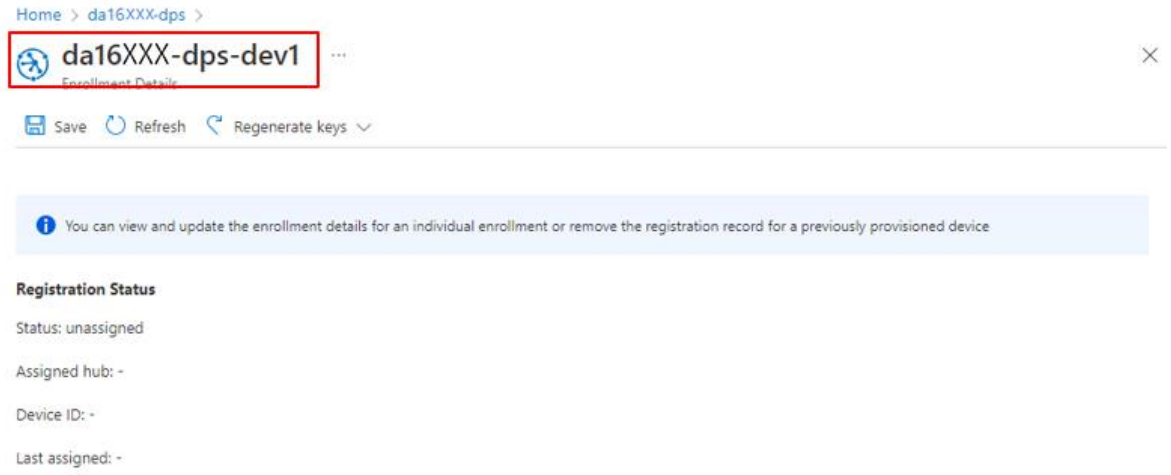

Figure 98: DPS Registration ID

2. In the DA16200/DA16600 SDK, edit app_thing_manager.h and define `APP_USER_MY_SYMMETRIC_KEY` as the **Primary Key** of the DPS Registration ID.

   For example: `#define APP_USER_MY_SYMMETRIC_KEY "ddl6VQRQymWXx30Hxv0TqikjQYzUSl5gmxC9JIpB5Vdi71UJtGRJUwbBxd/7tMOG1CuU43q0RFEodfCIQtMZCw=="`

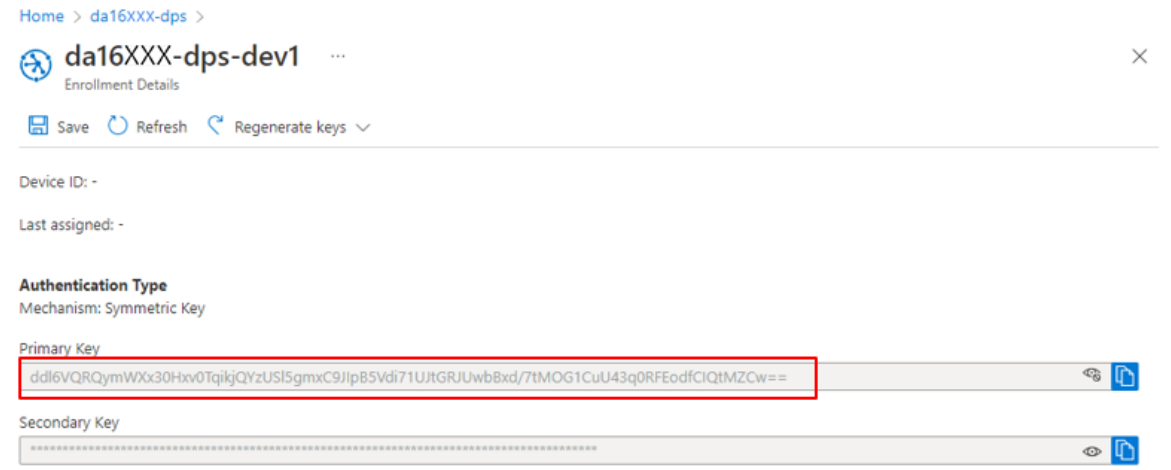

Figure 99: Primary Key of DPS Registration ID

3. In the DA16200/DA16600 SDK, edit app_azure_user_conf.h and define `ID_SCOPE` as the **ID Scope** of the **DPS overview** window.

   For example: `#define ID_SCOPE "0ne005E7EE4"`

© 2024 Renesas Electronics

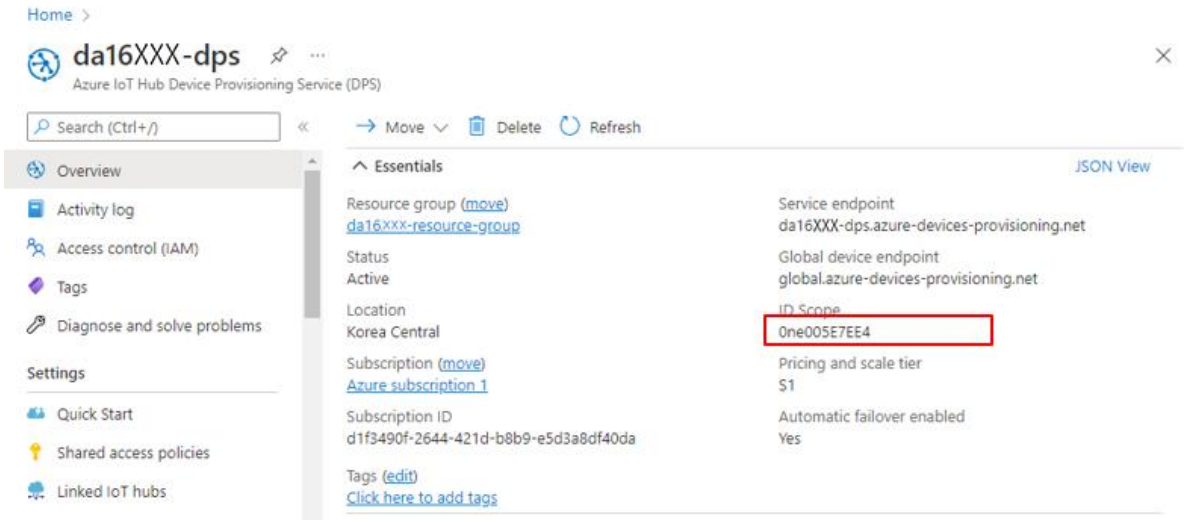

Figure 100: ID Scope of DPS

4. After enabled DPS feature is defined in app_azure_user_conf.h of the DA16200/DA16600 SDK as follows, compile it and download the image file to the device, and then reset the device.

```
#define ENABLE_AZURE_DPS_EXAMPLE                1
```

The DPS device is registered as shown in Figure 101 in the linked IoT Hub.

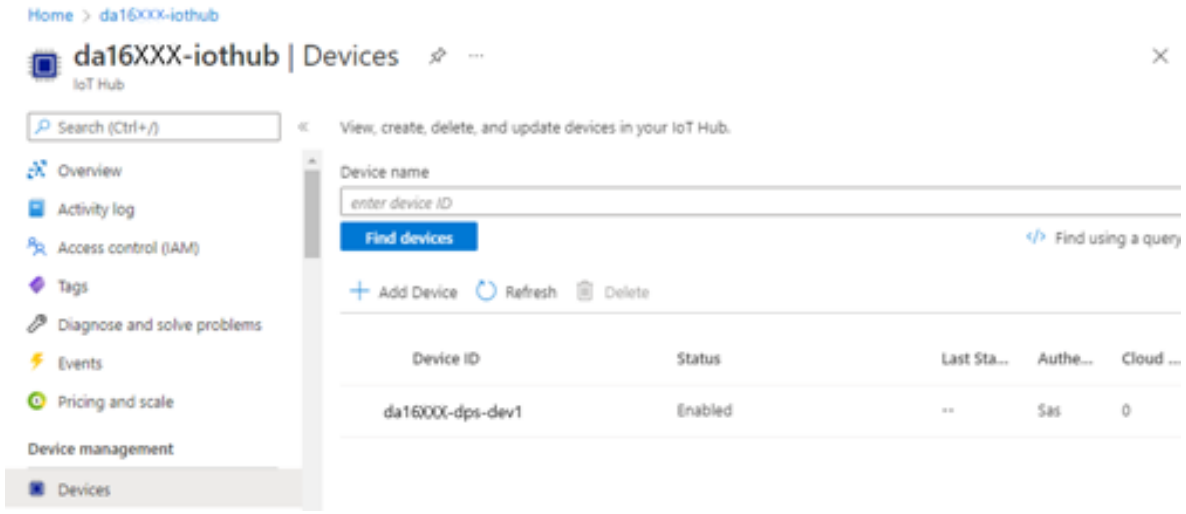

Figure 101: Registered DPS Device

© 2024 Renesas Electronics

Appendix B Azure IoT AT Command List

B.1 Operating Modes

There are three operating modes:

- Setting Mode for features configuration
- Provisioning Mode for network connection
- Communicating Mode for running

B.1.1 Setting Mode

After uploading a firmware image and rebooting, the DA16200/DA16600 enters setting mode. In this mode, all Azure IoT settings can be configured using the **SET** command and a specific topic can be configured using the **CFG** command. For proper operation of Azure IoT, the TLS certificate keys should be set. All configuration data is stored before calling the factory reset command (see Figure 102).

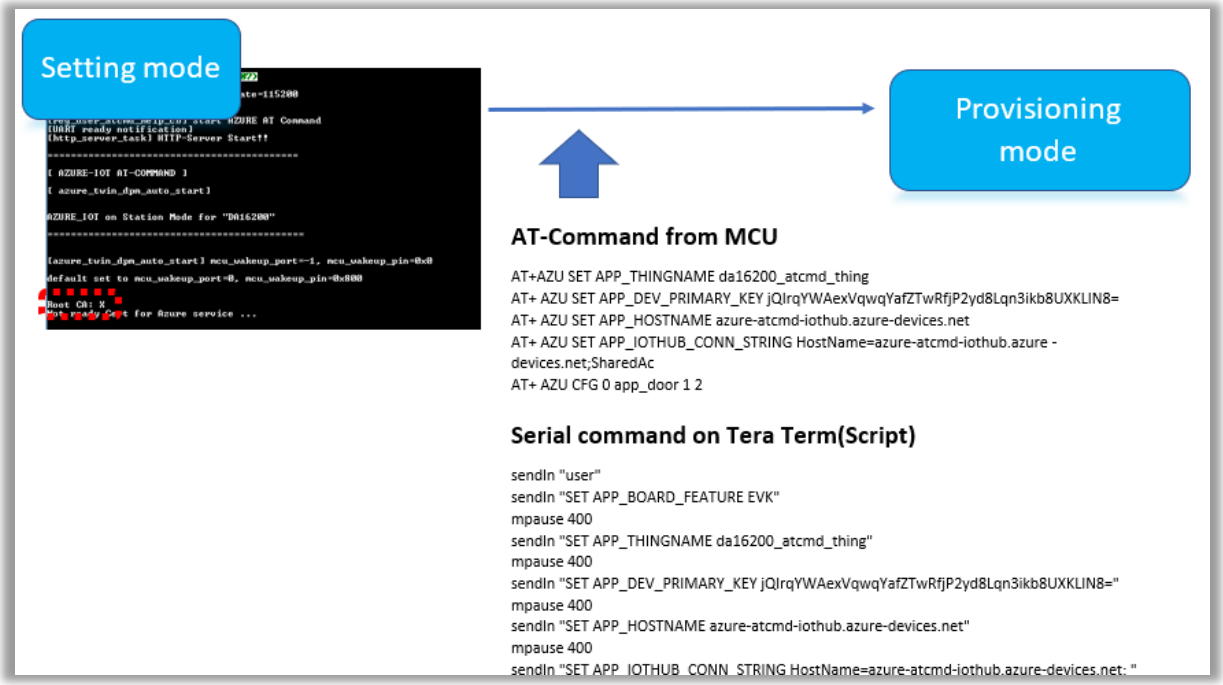

Figure 102: Setting Mode

B.1.2 Provisioning Mode

When 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 (see Figure 103).

© 2024 Renesas Electronics

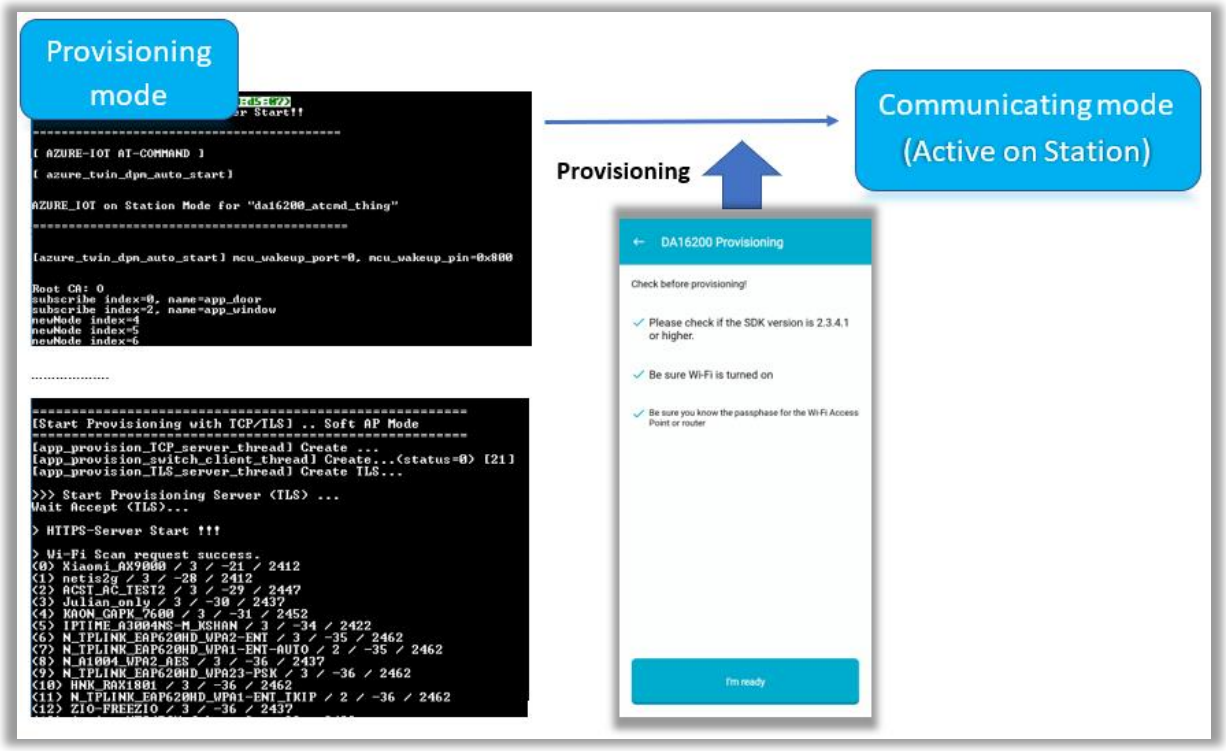

Figure 103: Provisioning Mode

B.1.3 Communication Mode

The DA16200/DA16600 Communicating Mode is used by the MCU to communicate (send and receive) topic values with an Azure server (see Figure 104).

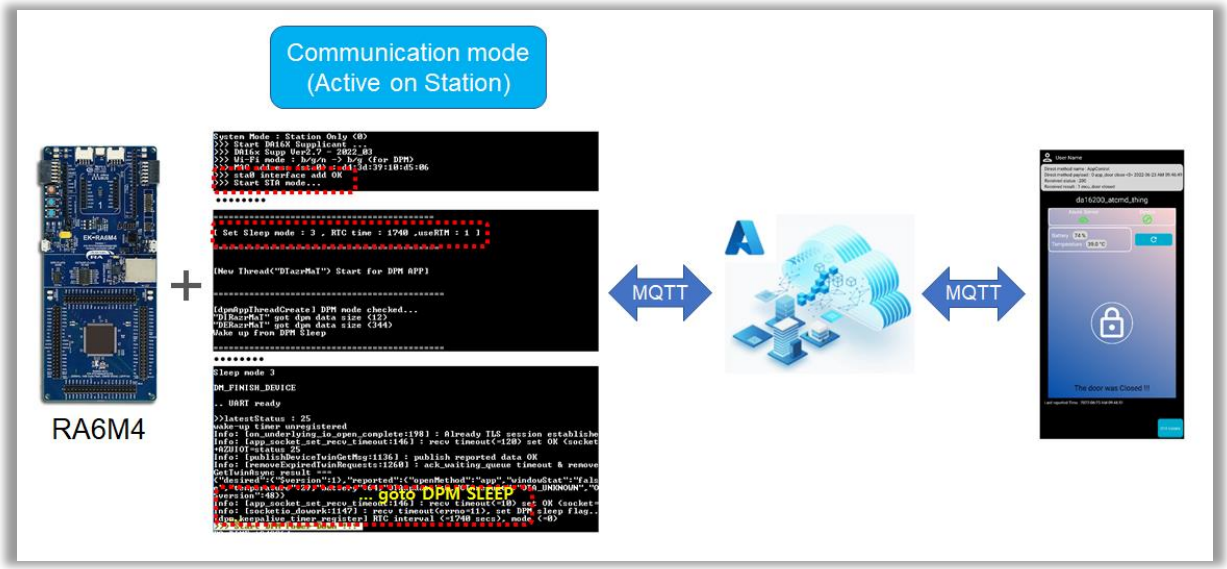

Figure 104: Communication Mode

B.2 How to Report Topic and Update Device Twin

B.2.1 Configure Device Twin Topic

- Topics are configured as shown in Table 5

© 2024 Renesas Electronics

- The MCU and Mobile App should be designed based on the topics shown in Table 5
- The MCU pushes the topics in Table 5 to the DA16200/DA16600 using AT commands

The DA16200/DA16600 facilitates the communication between the MCU and the phone as shown in Figure 105.

Table 5: Configuration of Topics

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”
4	battery	0: Integer	1: shadow	Battery value (0~100)
5	temperature	2: Float	1: shadow	Temperature value
6	doorState	1: String	1: shadow	“true”/”false”

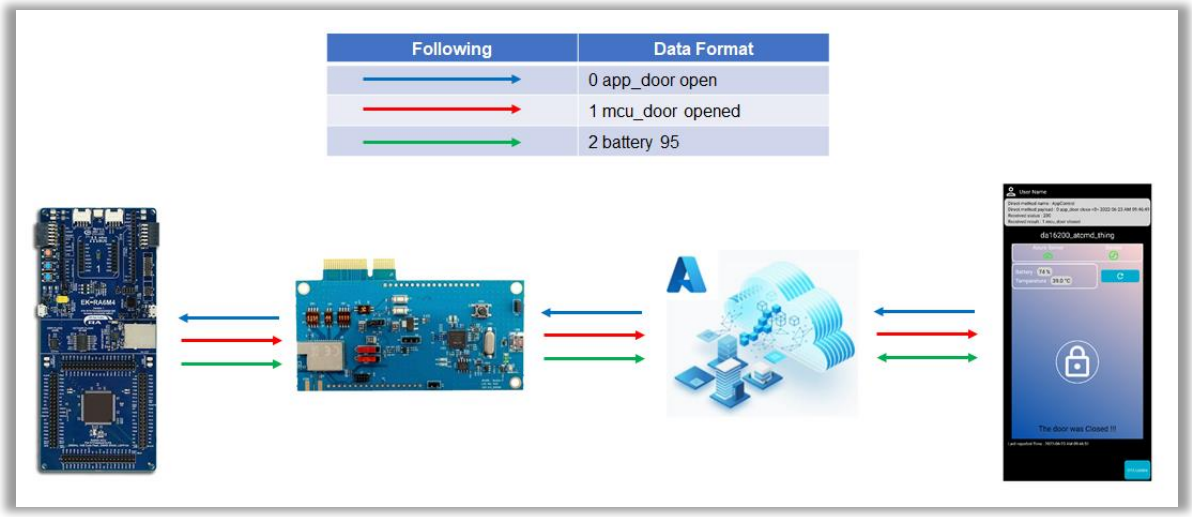

Figure 105: Communication from MCU to Phone

B.3 AT Command List

B.3.1 Basic Set

Table 6: Basic Set of MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AZU=	SET	APP_THINGNAME	Set the device thing name. Used to choose a device by its thing name during provisioning
		APP_DEV_PRIMARY_KEY	Set the device primary key.
		APP_HOSTNAME	Set the Azure server hostname .
		APP_IOTHUB_CONN_STRING	Set the IoT Hub connection string.

© 2024 Renesas Electronics

Head	Main	Sub	Parameters
		SLEEP_MODE	Set the DPM sleep mode 1 - not used 2 - not used 3 - connected sleep. The connection is retained even during DPM.
		USE_DPM	Defines the operation of sleep mode 3 0 - no DPM. Used during debug 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
EX) AT+AZU=SET APP_THINGNAME *AssignedDeviceID* AT+AZU=SET APP_HOSTNAME *azure-atcmd-iothub.azure-devices.net*			

B.3.2 TLS Certificate

Table 7: TLS Certificate from MCU to DA16200/DA16600

Start Code	Sub Code	Type	End Code
\x1b	C0,	Root CA Self-Signed, well known Has root Certificate public key Signed by root certificate private key	\x03
EX) send "\x1b“ over UART send "C0,-----BEGIN CERTIFICATE-----\n“ "MIIDQTCCAimgAwIBAgITBmyfz5m/jAo ….. over UART send “\x03”			

B.3.3 PIN MUX

Table 8: PIN MUX from MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AZU=	SET	NV_PIN_AMUX	AMUX_UART1d 4 /* UART1(RXD, TXD) */ AMUX_GPIO 9 /* GPIOA [1:0] */
		NV_PIN_BMUX	BMUX_UART1d 4 /* UART1(RXD, TXD) */ BMUX_GPIO 8 /* GPIOA [3:2] */

© 2024 Renesas Electronics

		NV_PIN_CMUX	CMUX_UART1d CMUX_GPIO	6 8	/* UART1(RXD, TXD) */ /* GPIOA [5:4] */
		NV_PIN_DMUX	DMUX_UART1d DMUX_GPIO	4 8	/* UART1(RXD, TXD) */ /* 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 GPIO_UNIT_C	0 2	 /*Support only GPIO 6,7,8 */
		APP_MCU_WKAEUP_PIN	GPIO_PIN0 ~ GPIO_PIN11		
		UART_CFG	[baud-rate]		

Note: Default pin mux is BMUX, and uses GPIOA2/GPIOA3 for UART1 and GPIOA9 for MCU wakeup.
AT+AZU=SET NV_PIN_BMUX BMUX_UART1d
AT+AZU=SET NV_PIN_EMUX EMUX_GPIO
AT+AZU=SET APP_MCU_WKAEUP_PORT GPIO_UNIT_A
AT+AZU=SET APP_MCU_WKAEUP_PIN GPIO_PIN9

B.3.4 Configure Data as Topics

Table 9: Configuration Data from MCU to DA16200/DA16600

Head	Main	Sub	Parameters
AT+AZU=	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. This means, door state = true/false 1 - Shadow: The value is sent to the device twin and will be 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. This means, door open command.

Ex) AT+AZU=CFG 0 app_door 1 2
AT+AZU=CFG 4 battey 0 1
AT+AZU=CFG 1 mcu_door 1 0

© 2024 Renesas Electronics

B.3.5 Command – MCU to DA16200/DA16600

Table 10: Command from MCU to DA16200/DA16600

Head	Main	Sub	Description
AT+AZU=	CMD	FACTORY_RESET	Resets the Azure IoT configuration to the factory default. All values stored in NVRAM are cleared. Use the “SET” and “CFG” commands to set the Azure IoT configuration.
		RESET_TO_AP	Switches to AP mode keeping the values set in NVRAM. The previous values in NVRAM will be kept.
		GET_STATUS	Gets the current Azure IoT status. The MCU can read the current status from the DA16200/DA16600 at any time.
		RESTART	Reboots 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]
Ex) AT+AZU=CMD FACTORY_RESET AT+AZU=CMD MCU_DATA 1 mcu_door opened			

B.3.6 Command – DA16200/DA16600 to MCU

Table 11: Command from DA16200/DA16600 to MCU

Head	Main	Parameters	Description
+AZUIOT	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.
+AZUIOT	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.
Ex) +AZUIOT SERVER_DATA 0 app_door open +AZUIOT CMD_TO_MCU update			

B.3.7 DA16200/DA16600 Status – DA16200/DA16600 to MCU

Table 12: Status from DA16200/DA16600 to MCU

Status	Value	Parameters
IDLE	-1	Initial state of AZU-IoT application Sent when a system error occurs. This means, network connection failure
Done factory reset	0	Sent after completes factory reset by “CMD FACTORY_RESET”
Boot Ready	1	Sent when entering AZU-IoT application mode

© 2024 Renesas Electronics

Status	Value	Parameters
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 It should 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 will happen during provisioning failure by the wrong SSID or PW It needs 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 start MCU OTA process.
EX) +AZUIOT STATUS 15		

© 2024 Renesas Electronics

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 it 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 [2].

© 2024 Renesas Electronics

Revision History

Revision	Date	Description
1.4	Jan. 26, 2024	Added Troubleshooting section
1.3	Sep. 27, 2023	Merged documents: ● UM-WI-058, DA16200 DA16600 Getting Started with Azure-IoT ● UM-WI-060, DA16200 DA16600 Getting Started with AT-Command for Azure-IoT
1.2	July 31, 2023	● Updated the reference section and images ● Changed the name of Dialog Semiconductor to Renesas Electronics
1.1	Sep. 01, 2022	Added description of DA16600
1.0	May 27, 2022	Initial release

© 2024 Renesas Electronics

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.

© 2024 Renesas Electronics

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 skilled in the art 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 for development of 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 out of 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.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu

Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

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:

https://www.renesas.com/contact/

Trademarks

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