**Contents**

(i) Setting up SDK to compile and generate the image file

(ii) Uploading Boot Image

(iii) AT Commands to set up DA16200

(iv) Wiring Diagaram to drive the DA16200 via nRF MCU UART for test configurations

(v) DA16200MOD Evaluation Board and UART Setup

(vi) Final Wiring Diagram for connecting nRF9160 to DA16200 for AWS and to nRF52480 for BLE configurations

**(i) Setting up SDK to compile and generate the image file**

1. Download the FreeRTOS SDK v3.2..8.1 from Renesas - <https://www.renesas.com/us/en/document/rln/da16200-da16600-freertos-sdk-release-note-v3281?language=en>

2. Download the AWS IoT Reference for FreeRTOS SDK- <https://www.renesas.com/us/en/document/sws/aws-iot-reference-freertos-sdk>

3. Install the AWS IoT package in the '~SDK\apps\' directory of the DA16200\_DA16600 Generic SDK as below:

apps \ app\_common

<br><dd>\ \*\*AWS\*\*

<br>\ common

<br>\ da16200

<br>\ da16600</dd>

4. Import the following folder into the Eclipse environment for access to the AWS IoT reference application

+ DA16200 : `~\SDK\apps\AWS\get\_started\_da16200\projects\da16200\_aws\`

**Or Use the SDK attached- DA16200\_DA16600\_SDK\_FreeRTOS\_v3.2.8.0\_1\_AWS\_ok1.rar**

i. Under the File menu, select Import to open the Import dialog box and then select General >Dialog SDK Project and click Next

ii. In the Import dialog box, select the Select SDK root directory and click the Browse button.

iii. Use the file manager to navigate to the <sdk\_root\_directory>/ directory and then click Select Folder.

iv. Various projects will appear in the Projects list. Select the **da16200\_aws project**. The second project to select is the **SDKJFlash** project which provides an absolute path for certain scripts to use and click Next.

v. Dialog SDK Project window will appear as shown below. Select Target device **DA16200** by selecting the button and then click Finish.

5. The basic setup for importing the SDK project is now complete. The next step is to build that project.

i. If this build process is performed for the first time, the Generate Configuration pop-up window is displayed automatically so that the flash memory type can be selected.

ii. Select the flash type - W25Q32JW , SIZES 4MB AND 4MB . click Generate to create the appropriate flash configuration file required to build the firmware.

**(ii) Uploading Boot Image**

6. Select the Run External Tools button and select the **uart\_program\_boot\_rtos\_win** launcher.

After that, specify the COM on the console and make sure that the device is not in DPM sleep (Reset it) after that the firmware should be downloaded automatically. Then please try to test the UART1 via AT commands.

7. The image can also be uploaded using the Macro available in teraterm.

Open the lower port in teraterm and set baudrate as 230400 in Setup->Serial port

using Control->Macro tab, and select Download\_W25Q32JW\_FreeRTOS.ttl to upload the Boot image..

8. The AT commands are run in the higher port with a baudrate of 115200

**(iii) AT Commands to set up DA16200**

1. **ATZ**  //to wake up DA if required

2. **ATE** //echo

3. **AT+WFMODE=0** //turn on station mode

4. **AT+WFJAPA=<ssid,password>** ex:AT+WFJAPA=Alcodex,Adx@2013 //wifi ssid and password

5. **AT+AWS=SET APP\_THINGNAME test\_aws\_iot** //set up AWS thing, test\_aws\_iot

6. **AT+AWS SET AWS\_BROKER a3kbzziaf9hg4n-ats.iot.us-east-1.amazonaws.com** //broker

7. **AT+AWS=SET APP\_PUBTOPIC test\_aws\_iot/testtopic** //Publish topic

8. //Set certificates using teraterm- Press Esc key, type C0, and copy paste the contents of the first certificate. Press CtrlC. proceed to certificate 2 as mentioned below

**<ESC Key>C0,CA1<CTRL+C>** //copy contects of amazonRootCA1 cert in ca1 space

**<ESC Key>C1,certificatekey<CTRL+C>** //cert.perm

**<ESC Key>C2,private key<CTRL+C>** //private

9. **AT+AWS=CFG 0 latshad 1 1** //configure shadow for publish topic

10. **AT+AWS=CMD MCU\_DATA 0 latshad seninit** //establish connection by sending a token value to the shadow

11. **AT+AWS=CFG 1 latlon 1 0** //set cofiguration to publish mode

12. **AT+AWS=CMD MCU\_DATA 1 latlon 28.123456789** //send values to publish to AWS

//location coordinates can be sent together or in 2 separate variables

13. AT+AWS=CMD MCU\_DATA 1 latlon 10.059065067392345W76.34034918061742N

**(iv) Wiring Diagram to drive the DA16200 via nRF MCU UART**

via nRF52480



or via nRF9160 which will be our main MCU going further



**(v) DA16200MOD Evaluation Board and UART Setup**





**(iv) Wiring Diagram to connect between nRF9160 and nRF52840 via UART 2 and nRF9160 to DA16200 via UART1**



The nRF9160 will be the central MCU. It drives the gnss module to get location information and also connects to the AWS to share the current location. It also sends this info to the nRF52840 to enable BLE data connections.