

User Manual

DA16200 OTA Update

UM-WI-036

Abstract

This OTA update User Manual intends to assist software developers that implement applications with the DA16200 SDK. A certain degree of reader familiarity to programming environments, debugging tools and software engineering process in general is assumed.

Contents

Abstract	1
Contents	2
Figures	2
Tables	2
Terms and Definitions	4
References	4
1 Introduction	5
2 SFLASH Memory Area	5
3 HTTP Protocol	7
4 OTA Update Function	7
4.1 Header	8
4.2 Version	8
4.3 Result Code	8
4.4 Download	9
4.5 Renew	10
4.5.1 Boot Index	11
5 Application Programming	12
5.1 Type	12
5.2 Structure	13
5.3 APIs	14
5.4 Example	15
5.4.1 Sample Code	15
5.4.2 Test Command	15
6 OTA Update Extension	15
6.1 Certificates	15
6.2 MCU Firmware	16
Appendix A OTA Test Server	16
Revision History	18

Figures

Figure 1 OTA Update Layer	5
Figure 2 Firmware Header Information	8
Figure 3 Firmware DOWNLOAD	10
Figure 4 Firmware RENEW	11
Figure 5 Boot Index Operation	12

Tables

Table 1: 2 MByte SFLASH Memory Map	6
Table 2: 4 MByte SFLASH Memory Map	7
Table 3: Result Code	8

DA16200 OTA Update

Table 4 OTA Update Type.....	12
Table 5 OTA_UPDATE_CONFIG	13
Table 6 Lists for OTA APIs	14
Table 7 OTA Test Command.....	15

Terms and Definitions

OTA	Over The Air
API	Application Programming Interface
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol over Secure Socket Layer
MQTT	Message Queuing Telemetry Transport
TLS	Transport Layer Security
AWS	Amazon Web Services
CLI	Command Line Interface

References

- [1] <DA16200>, Datasheet, Dialog Semiconductor
- [2] <DA16200>, SDK Programmer Guide, Dialog Semiconductor
- [3] <lwIP>, Lightweight IP stack, https://www.nongnu.org/lwip/2_1_x/group__httpc.html

DA16200 OTA Update

1 Introduction

The DA16200 can update the firmware over the air using the HTTP protocol. The DA16200 operates as an HTTP client, it can download and update new firmware from the HTTP server.

Users can easily develop these functions using the API provided by the DA16200 SDK.

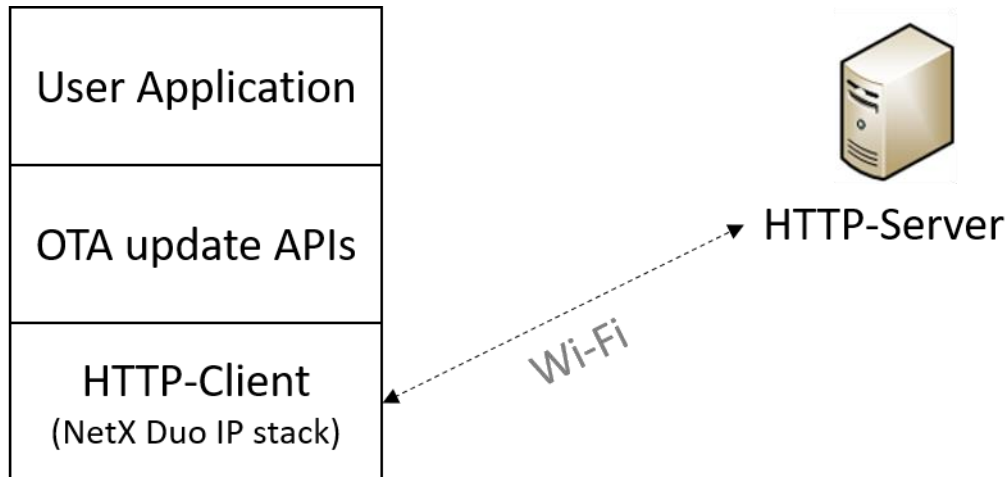


Figure 1 OTA Update Layer

2 SFLASH Memory Area

The DA16200 does not support file systems, so that the firmware is stored in the SFLASH memory area. SFLASH is divided into several areas as shown in the [Table 1](#). Among them, the areas that users can directly access are as follows:

- User accessible SFLASH area
 - RTOS #0
 - RTOS #1
 - User Area #0
 - User Area #1
 - TLS Certificate #0
 - TLS Certificate #1

NOTE: If other areas are accessed incorrectly, serious failure may occur in the system.

DA16200 OTA Update

Table 1: 2 MByte SFLASH Memory Map

Address	Name		Size (byte)
0x0000_0000	ADDRESS START		-
0x0000_0000	Bootloader		36,864
0x0000_9000	Boot Index		4,096
0x0000_A000	RTOS #0		946,176
0x000F_1000	SLIB #0		53,248
0x000F_E000	RTOS #1		946,176
0x001E_5000	SLIB #1		53,248
0x001F_2000	User Area		12,288
0x001F_5000	Debug / RMA Certificate		4,096
0x001F_6000	TLS Certificate #0 (MQTT)	CA	16,384
0x001F_7000		Cert	
0x001F_8000		Private key	
0x001F_9000		DH	
0x001F_A000	TLS Certificate #1 (HTTPS / OTA)	CA	16,384
0x001F_B000		Cert	
0x001F_C000		Private key	
0x001F_D000		DH	
0x001F_E000	NVRAM #0		4,096
0x001F_F000	NVRAM #1		4,096
0x0020_0000	ADDRESS END		-

DA16200 OTA Update

Table 2: 4 MByte SFLASH Memory Map

Address	Name	Size (byte)
0x0000_0000	ADDRESS START	-
0x0000_0000	Bootloader	36,864
0x0000_9000	Boot Index	4,096
0x0000_A000	RTOS #0	1,572,864
0x0018_A000	SLIB #0	65,536
0x0019_A000	User Area #0	372,736
0x001F_5000	Debug / RMA Certificate	4,096
0x001F_6000	TLS Certificate #0 (MQTT)	CA
0x001F_7000		Cert
0x001F_8000		Private key
0x001F_9000		DH
0x001F_A000	TLS Certificate #1 (HTTPS / OTA)	CA
0x001F_B000		Cert
0x001F_C000		Private key
0x001F_D000		DH
0x001F_E000	NVRAM #0	4,096
0x001F_F000	NVRAM #1	4,096
0x0020_0000	RTOS #1	1,572,864
0x0038_0000	SLIB #1	65,536
0x0039_0000	User Area #1	458,752
0x0040_0000	ADDRESS END	-

3 HTTP Protocol

The DA16200 supports HTTP/HTTPS 1.1. DA16200 requests firmware download to the HTTP-server by using the GET method of the HTTP-client.

The OTA update application must know the URL of the HTTP-server in advance before requesting a download. How to obtain the URL depends on the user's preference. Therefore, it is not mentioned in this manual.

There should be at least 50 kbyte of free heap memory for HTTPs (TLS)

4 OTA Update Function

DA16200 firmware is divided into two firmwares: SLIB and RTOS. To update the firmware, SLIB and RTOS firmware must be downloaded.

The OTA update function is divided into two stages: DOWNLOAD and RENEW.

1. DOWNLOAD refers to the process of downloading new firmware from the OTA server. In this case, the new firmware is not yet applied.
2. RENEW is the process of applying to operate with the successfully downloaded firmware. To do this, some rules and information are required.

DA16200 OTA Update

4.1 Header

Figure 2 shows DA16200 header information as an example. Header information is 96 bytes and automatically inserted when firmware is built. Users do not need to understand all the contents of the header. The red box in Figure 2 is the magic number and version information. The yellow box is information for checking firmware CRC. Users only need to understand the version information in the red box.

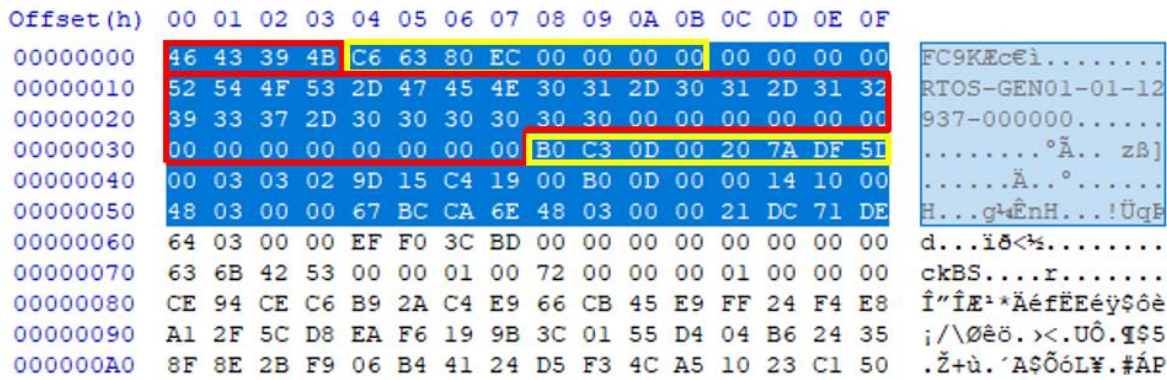


Figure 2 Firmware Header Information

4.2 Version

DA16200's RTOS has unique version rules for system protection. The version name is inserted as a 26-character string in the header part of the firmware image at the time of build. (Maximum 32 bytes).

There are five elements in the version string, separated by "-". These are: Type, Vendor, Major, Minor, and Customer.

1. Type (4 bytes) : Identify the type of firmware. (SLIB, RTOS)
2. Vendor (5 bytes) : Vendor classification
3. Major (2 bytes) : Major number to check compatibility
4. Minor (4~5 bytes) : SDK patch number
5. Customer (6 bytes) : User configurable version

Users can change the customer version by editing ..\version\3rd_customer_build_num.h. If users change the customer version and build the SDK, the customer version is applied to the image.

4.3 Result Code

All APIs, provided by OTA update return the result codes shown in Table 3. It is delivered through the callback function connected with DOWNLOAD and RENEW API.

Table 3: Result Code

Result Code	Value	Description
OTA_SUCCESS	0x00	Return success.
OTA_FAILED	0x01	Return failed.
OTA_ERROR_SFLASH_ADDR	0x02	SFLASH address is wrong.
OTA_ERROR_TYPE	0x03	FW type is unknown.
OTA_ERROR_URL	0x04	Server URL is unknown.
OTA_ERROR_SIZE	0x05	FW size is too big.
OTA_ERROR_CRC	0x06	CRC is not correct.

DA16200 OTA Update

Result Code	Value	Description
OTA_VERSION_UNKNOWN	0x07	FW version is unknown.
OTA_VERSION_INCOMPATI	0x08	FW version is incompatible.
OTA_NOT_FOUND	0x09	Fw not found on the server.
OTA_NOT_CONNECTED	0x0A	Failed to connect to server.
OTA_NOT_ALL_DOWNLOAD	0x0B	All new FWs have not been downloaded.
OTA_MEM_ALLOC_FAILED	0x0C	Failed to alloc memory.

4.4 Download

The download step is the process of downloading firmware from the OTA server and saving it into the SFLASH area.

The communication protocol with the OTA server uses HTTP and can be implemented using the HTTP API supported by lwIP. Therefore, the process of communicating with HTTP-server works the same as lwIP's HTTP-client.

The download sequence proceeds as follows, and both success and failure results can be delivered through the callback function (see [Table 3](#) for results).

1. Request a query from the HTTP-server.
2. Confirm that the response was successfully received from the HTTP server. If the server connection fails or receives a failure response, the download will be terminated and the result will be transferred to the callback function. See table for result values.
3. Check the magic number and version name in the firmware header, and if they do not match, the download is terminated and the result is transferred to the callback function (Compatibility of SLIB and RTOS is checked).
4. If the magic number and version name are normal, the downloaded data is written to SFLASH. When the download is completed successfully, the entire firmware stored in SFLASH will have CRC check.
5. When the CRC check is successfully completed, the result value of 0x00 is transferred to the callback function and the download is terminated.

DA16200 OTA Update

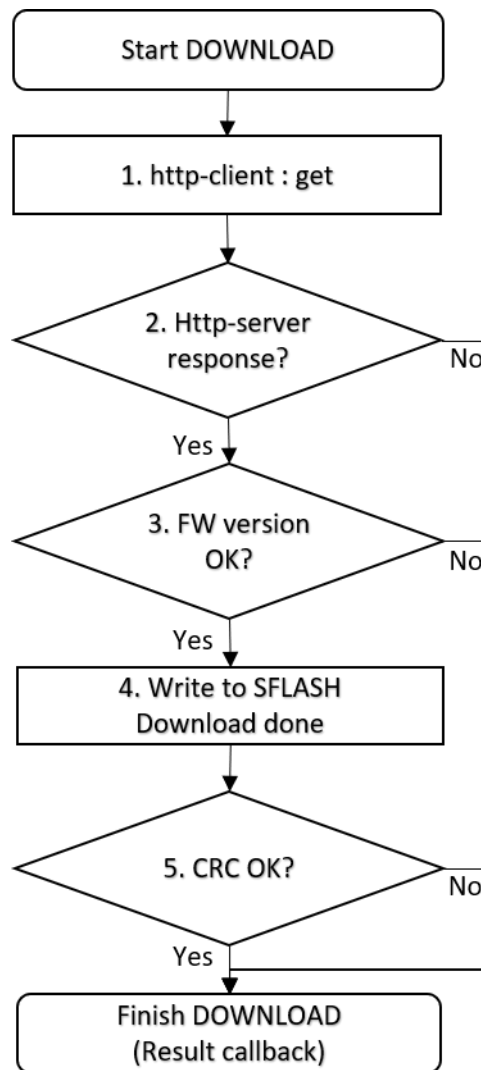


Figure 3 Firmware DOWNLOAD

4.5 Renew

RENEW only operates when the firmware download is successful. Additionally, the download success can be checked, based on the power ON/OFF standard of the DA16200, DA16200 should have the download history after power ON.

1. Check whether the download of both SLIB and RTOS firmware is successful. After turning on the power, check the download history.
2. Check the CRC of the firmware stored in the SFLASH. In case of failure, RENEW ends and the result is transmitted to the callback function.
3. Check the firmware version stored in the flash. In case of failure, RENEW ends and the result is transmitted to the callback function.
4. It is determined that the new firmware is normal and the boot index is changed to the new firmware location.

DA16200 OTA Update

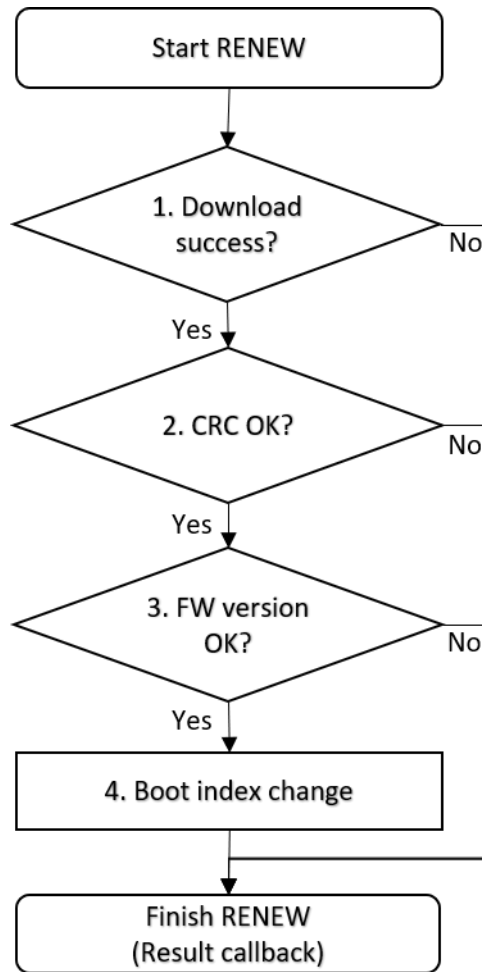


Figure 4 Firmware RENEW

4.5.1 Boot Index

The DA16200 is divided into firmware download area and current area for OTA update function. The two areas are toggled on each other by the boot index. For example, if the boot index value is 0, it operates as the firmware stored in the SFLASH RTOS #0 area upon booting, and the newly downloaded firmware is stored in RTOS #1. After that, if RENEW is operated successfully, the boot index value is changed to 1, rebooted, and the firmware stored in the SFLASH RTOS#1 area is operated.

DA16200 OTA Update

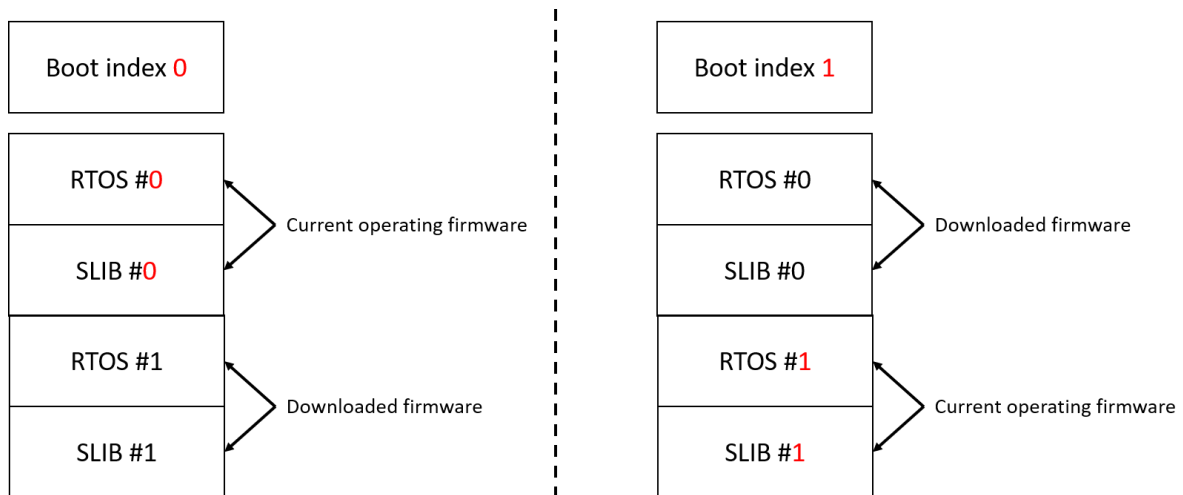


Figure 5 Boot Index Operation

5 Application Programming

Describes the structures and APIs required for the OTA firmware update application.

5.1 Type

OTA update task is operated based on the type defined in OTA update type. The operation sequence is tailored to the specified type.

Table 4 OTA Update Type

Name	ota_update_type
Description	Identify and specify targets for OTA updates.
	<pre> /// Operation step of process enum ota_update_fw_type { /// Init value OTA_TYPE_INIT, /// SLIB OTA_TYPE_SLIB, /// RTOS OTA_TYPE_RTOS, /// Other firmware - ex) user mcu firmware OTA_TYPE_OTHER_FW, /// Certificate for first OTA_TYPE_CERT_1, /// Certificate for second OTA_TYPE_CERT_2, /// Unknown value OTA_TYPE_UNKNOWN }; </pre>

DA16200 OTA Update

5.2 Structure

OTA UPDATE CONFIG sets the necessary parameters when calling OTA firmware update API.

Table 5 OTA_UPDATE_CONFIG

Name	OTA_UPDATE_CONFIG
Description	Contains information to be passed as argument values to OTA update APIs.
	<pre> /// OTA update configuration structure typedef struct { /// Pointer variable to the server address where the SLIB exists. char *uri_slib; /// Pointer variable to the server address where the RTOS exists. char *uri_rtos; /// Pointer variable to the server address where the other_fw exists. char *uri_other_fw; #ifdef (__FOR_4MB_SFLASH__) /// Pointer variable to the server address where the first certificate exists. char *uri_1st_cert; /// Pointer variable to the server address where the second certificate exists. char *uri_2nd_cert; #endif // (__FOR_4MB_SFLASH__) /// Callback function pointer to call HTTP-client API. void (*http_configure)(NX_HTTP_CLIENT *http); /// Callback function pointer to check the download status. void (*download_complete_notify)(UINT fw_type, UINT ret_status); /// Callback function pointer to check the renew state. void (*renew_notify)(UINT ret_status); /// If the value is true, if the download of a new FW succeeds, the new FW is rebooted. UINT auto_renew; /// Address of SFLASH where other_fw is stored. UINT other_fw_addr; /// If the value is true, firmware download does not check compatibility by version. (Use only in debugging or special situations.) UINT disable_version_check; } OTA_UPDATE_CONFIG; </pre>

DA16200 OTA Update

5.3 APIs

Describes the API required for the OTA firmware update application.

Table 6 Lists for OTA APIs

UINT ota_update_start_download(OTA_UPDATE_CONFIG *fw_conf)		
Parameter	ota_update_conf	The pointer of OTA_UPDATE_CONFIG structure.
Return		Returns 0x00 on success. See Table 3.
Description		HTTP-client task is created and sent query to HTTP-server. It checks the version compatibility of the firmware received from the server and writes it to the download area of SFLASH.

UINT ota_update_stop_download(void)		
Parameter	void	None.
Return		Returns 0x00 on success. See Table 3.
Description		Download can be stopped while downloading from HTTP-server.

UINT ota_update_get_download_progress(UINT fw_type)		
Parameter	update_type	Specifies the type to be updated.
Return		Returns a value between 0 and 100. If the download was successful, it returns 100.
Description		Check the progress while downloading or after completion.

UINT ota_update_start_renew(OTA_UPDATE_CONFIG *fw_conf)		
Parameter	ota_update_conf	The pointer of OTA_UPDATE_CONFIG structure.
Return		Returns 0x00 on success. See Table 3.
Description		Check the version compatibility and CRC, change the boot index to the new firmware location, and then reboot automatically.

DA16200 OTA Update

5.4 Example

5.4.1 Sample Code

The DA16200 SDK provides sample code. See the sample code located below.

```
..\sample\Network\OTA_Update\src\ota_update_sample.c
```

5.4.2 Test Command

OTA updates support the CLI command. It can be helpful for testing when developing an application.

Table 7 OTA Test Command

Command	Option	Description
ota_update	[update_type] [url] <[sflash_addr]>	Start to FW download. * update_type slib : update_type of SLIB rtos : update_type of RTOS cert_1 : update_type of cert_key.(only for 4MB) cert_2 : update_type of cert_key.(only for 4MB) * url : Server URL where FW exists ex) ota_update rtos http://192.168.0.1/rtos.img
	stop	Stop to firmware download. ex) ota_update stop
	renew	Change current firmware to new firmware. ex) ota_update renew
	info	Show FW information. ex) ota_update info
	crc	[addr] Check CRC of firmware. ex) ota_update crc 0x23000
	read	[addr] [size] Read SFLASH data. ex) ota_update read 0x23000 128

6 OTA Update Extension

The OTA update function supports updating not only the DA16200's firmware, but also the firmware of the MCU chip or the certificate for TLS protocol.

6.1 Certificates

Certificate update is only supported by 4 MByte SFLASH memo.

The certificate update types are OTA_TYPE_CERT_KEY_1, OTA_TYPE_CERT_2, which are defined in the enum ota_update_type.

Update the TLS Certificate #0 and TLS Certificate #1 areas stored in SFLASH.

DA16200 OTA Update

6.2 MCU Firmware

MCU firmware update type is `OTA_TYPE_OTHER_FW` and defined in `enum ota_update_type`.

If the `other_fw_addr` value of `OTA_UPDATE_CONFIG` is not set, it is downloaded to the User Area area(In case of 4 MByte SFLASH , it is User Area #1). The MCU firmware downloaded to the User Area area(In case of 4 MByte SFLASH , it is User Area #1.) can be transmitted to the MCU using UART1. If the user needs hardware configuration, please contact our customer service team.

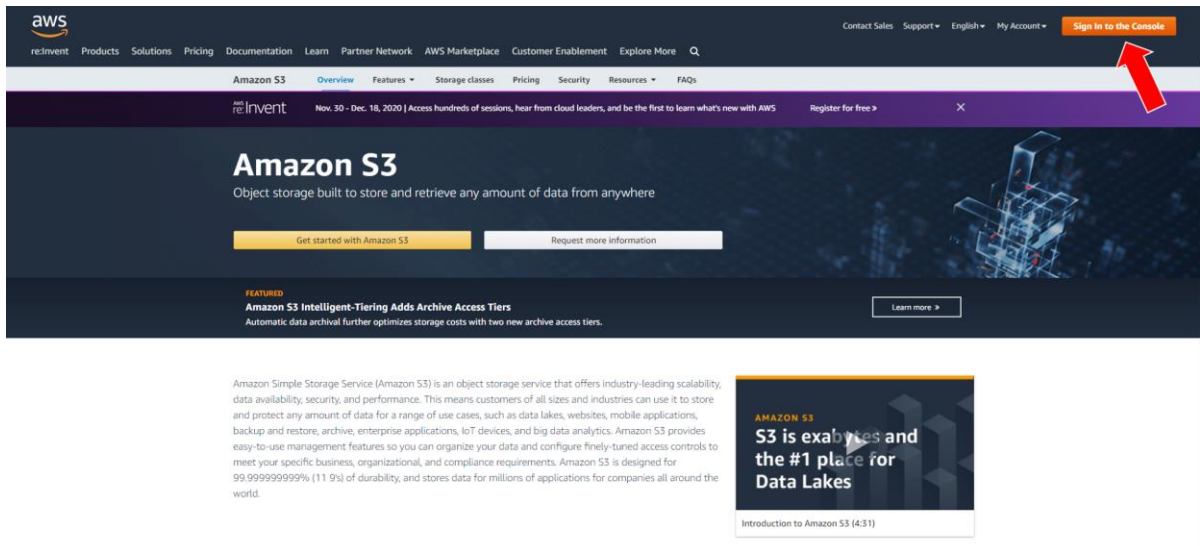
Appendix A OTA Test Server

OTA update complies with HTTP protocol to download firmware. Therefore, users can easily implement OTA server using HTTP-server. This manual does not provide a guide on configuring OTA servers. However, it explains how to configure a simple test environment for functional testing in the application development stage.

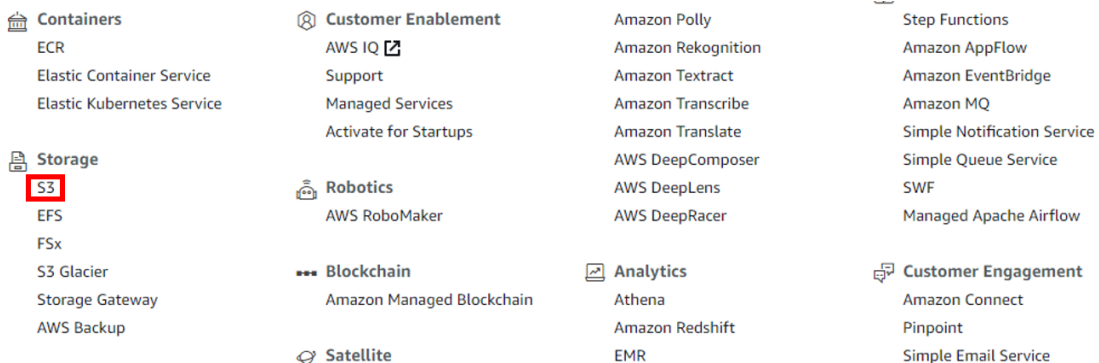
AWS S3 is recommended as the OTA test server.

Procedure:

1. Sign up for an AWS account for free and log in to the console.

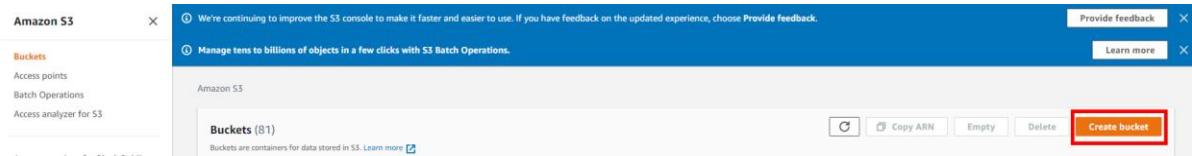


2. Select S3 in the storage category.

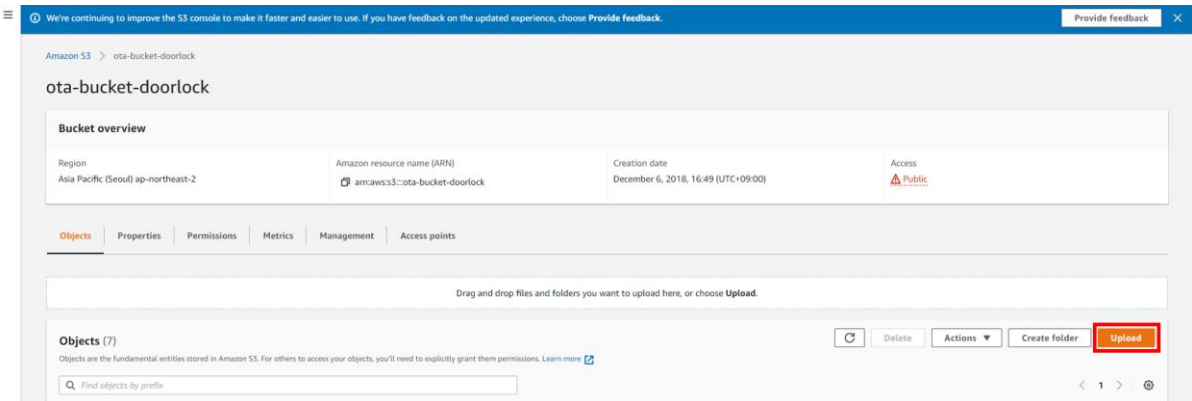


DA16200 OTA Update

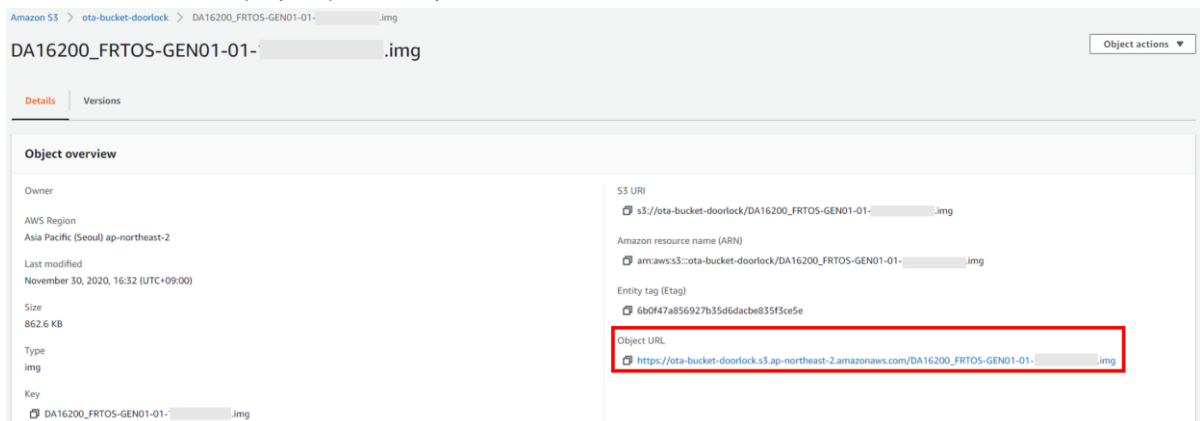
3. Click on "Create Bucket" to create a bucket with default settings.



4. Upload the firmware to the created bucket.



5. Check the URL (https://) of the uploaded firmware.



6. Set the URL as the OTA update API parameter value and proceed with the test.

Revision History

Revision	Date	Description
1.0	02-Dec-2020	First Release

DA16200 OTA Update

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.

Disclaimer

Unless otherwise agreed in writing, the Dialog Semiconductor products (and any associated software) referred to in this document are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of a Dialog Semiconductor product (or associated software) can reasonably be expected to result in personal injury, death or severe property or environmental damage. Dialog Semiconductor and its suppliers accept no liability for inclusion and/or use of Dialog Semiconductor products (and any associated software) in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, express or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications. Notwithstanding the foregoing, for any automotive grade version of the device, Dialog Semiconductor reserves the right to change the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications, in accordance with its standard automotive change notification process.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document is subject to Dialog Semiconductor's [Standard Terms and Conditions of Sale](http://www.dialog-semiconductor.com), available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog, Dialog Semiconductor and the Dialog logo are trademarks of Dialog Semiconductor Plc or its subsidiaries. All other product or service names and marks are the property of their respective owners.

© 2020 Dialog Semiconductor. All rights reserved.

RoHS Compliance

Dialog Semiconductor's 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.

Contacting Dialog Semiconductor

United Kingdom (Headquarters)

Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany

Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands

Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:

enquiry@diasemi.com

North America

Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan

Dialog Semiconductor K. K.
Phone: +81 3 5769 5100

Taiwan

Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:

www.dialog-semiconductor.com

Hong Kong

Dialog Semiconductor Hong Kong
Phone: +852 2607 4271

Korea

Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)

Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)

Dialog Semiconductor China
Phone: +86 21 5424 9058