Firmware-Over-the-Air (FOTA) allows for wireless delivery of firmware updates and/or configurations to embedded devices.

This section describes the FOTA process for the Talaria TWO EVB using the Talaria TWO SDK with details on how to implement or trigger FOTA in a customer provided application.

This implementation of FOTA provides the following facilities:

1. Check for the availability of new upgrades
2. Securely download the image into flash
3. Check the validity of the downloaded image
4. Set the new image as the boot image

In conjunction with SSBL, it enables booting the latest image downloaded. The firmware is downloaded into the application image partition in Flash.

Refer FOTA application for various configuration files and modules on which FOTA is dependent, its design and functional flow.

# Features and Limitations

Following are the FOTA application features:

1. FOTA over HTTP/HTTPS.
2. Image download from Cloud or any HTTP/web server.
3. Two copy solution. Backup copy of the correct firmware always exists.
4. Image integrity check using sha256 hash.
5. Error handling and recovery
   1. If any error occurs during downloading the image or updating the configuration files (part.json/boot.json/fota\_config.json), the device will remain in the current image.
   2. If a reboot occurs (due to issues like power failure) during image download or configuration files upgrade, the device will boot with the current image.
6. JSON based configuration

Limitations: Upgrading the Certificates is not supported as of now.

# Header file/s

Components/fota/inc/fota.h.

# API Reference

## fota\_init

### Overview

This API initializes the FOTA module.

### Definition

|  |
| --- |
| fota\_handle\_t \*  fota\_init(void) |

### Return

Success: Pointer to FOTA handle.

Error: NULL.

## fota\_perform

### Overview

Performs the FOTA update. This API checks for the update. If an update is available, it downloads the firmware specified in the fota.config file.

### Definition

|  |
| --- |
| int  fota\_perform(fota\_handle\_t \*f\_handle, int check\_for\_update, int flags) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *handle* | Handle returned by fota\_init() |
| *check\_for\_update* | This can be FOTA\_NO\_CHECK\_FOR\_UPDATE / FOTA\_CHECK\_FOR\_UPDATE / FOTA\_FETCH\_CFG\_NO\_CHECK\_FOR\_UPDATE   1. FOTA\_NO\_CHECK\_FOR\_UPDATE - No check done. FOTA is done using the currently available fota.config file in the data fs. 2. FOTA\_CHECK\_FOR\_UPDATE - Check if an update is available before executing FOTA. foat.config file will be fetched from cloud and the package version of the latest file is compared against the fota.config file present in data fs. FOTA is done only if the latest version is greater than the one preset on the device. 3. FOTA\_FETCH\_CFG\_NO\_CHECK\_FOR\_UPDATE - foat.config file will be fetched from the cloud. No check is done using package version. Newly downloaded fota.config file will be used for FOTA. This option can be used only if availability of newer package is confirmed using some other mechanism (like, MQTT for example) and FOTA needs to be done as per the latest fota.config in the cloud. |
| *Flags* | Not currently used. Always set to 0. |

Table : fota\_perform - parameters

### Return

Success: 0

Error: <0 (Refer fota\_error\_t)

## fota\_commit

### Overview

After the FOTA update is done, call this function to set the newly updated firmware as the default. This needs to be called after fota\_perform() is a success.

### Definition

|  |
| --- |
| int  fota\_commit(fota\_handle\_t \*f\_handle, int do\_reset) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *f\_handle* | Pointer to the FOTA handle that was returned by fota\_init() |
| *do\_reset* | If set to 1, perform reset |

Table : fota\_commit - parameters

### Return

Success: 0

Error: -1

## fota\_deinit

### Overview

This API will release all the resources allocated during fota\_init() and other FOTA API calls.

### Definition

|  |
| --- |
| void fota\_deinit(fota\_handle\_t \* f\_handle) |

### Parameters

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| *f\_handle* | Pointer to FOTA handle that was returned by fota\_init() |

Table : fota\_deinit - parameters

### Return

None.

# Application Example

For the example code, refer: *apps/fota application*.