

Bluetooth SDK 2.4.01 Gecko SDK Suite 1.1.0 MCU 5.2.1.0

For the OTA DFU we need the Internal Storage Bootloader (single image) (EFR32BG12 parts) or the Bluetooth in-place OTA DFU Bootloader (EFR32BG1 parts) configuration.

1. Open Simplicity Studio, and select your device in the Devices tab
2. Check the Preferred SDK at the top of the main window
3. Click New Project button
4. Select Gecko Bootloader application type, click Next
5. Select the latest installed SDK, click Next
6. Select Internal Storage Bootloader (single image) or Bluetooth in-place OTA DFU Bootloader sample application, click Next
7. Name your project, click Next
8. Check your device and choose toolchain. Click Finish
9. Open the Plugins tab
10. Click on Bootloader Core, On the right side tick the checkboxes Require signed firmware upgrade files
Require encrypted firmware upgrade files Enable secure boot
11. Click Generate in the upper right corner
12. Build your bootloader project.

Generate and flash security keys. The private key is used to generate the public key. All applications developed to be used with this bootloader must be signed using the private key. Before going further add `\SiliconLabs\SimplicityStudio\v4\developer\adapterpacks\commander\commander.exe` to system environment variables

Run the following commands

```
$ commander gbl keygen --type ecc-p256 --outfile app-sign-key.pem
```

```
$ commander gbl keygen --type aes-ccm --outfile app-encrypt-key.txt
```

```
$ commander flash --tokengroup znet --tokenfile app-encrypt-key.txt --tokenfile app-sign-key.pem-tokens.txt -  
-device EFR32BG1B232F256GM56
```

Create Bluetooth app with secure OTA DFU capability

Since OTA DFU is not fully implemented in the Bootloader, a Bluetooth application has to be created and flashed to the device first along with the Bootloader to support the upgrade. This can be any app that supports restarting the device in DFU mode. The easiest is to use the SoC-Empty software example.

1. Open Simplicity Studio, and select your device in the Devices tab
2. Check the Preferred SDK at the top of the main window
3. Click New Project button
4. Select Bluetooth SDK, click Next
5. Select the latest installed SDK, click Next
6. Select SOC - Empty sample application, click Next
7. Name your project, click Next
8. Check your device and choose toolchain. Click Finish
9. Build you project
10. Copy app-sign-key.pem and app-encrypt-key.txt into the Bluetooth project directory.
11. Run create_bl_files.bat found in your Bluetooth project
12. Copy the bootloader image that ends with `combined.s37` from the output folder(GNU ARM) of your bootloader project to the output_gbl folder of your Bluetooth project

13. Run the following command in the output_gbl folder:

```
$ commander convert bootloader-storage-internal-ble-combined.s37 stack-signed.gbl app-signed.gbl --outfile  
bootloader+stack+app.hex
```

14. Open Commander and flash the hex file to the device. ****Do not erase the flash before flashing the hex file as it will erase the tokens flashed previously

Create Bluetooth app to be uploaded via OTA

1. Build your project (iBeacon or Thermometer Example)
2. Copy app-sign-key.pem and app-encrypt-key.txt into the Bluetooth project folder of this project
3. Run create_bl_files.bat found in your Bluetooth project folder

This will create stack-signed-encrypted.gbl and app-signed-encrypted.gbl files into the output_gbl folder. These are the signed and encrypted upgrade files, which can be sent OTA to the target device

Performing the OTA

1. **(Android)** Copy the .gbl files to your phone and store it in /SiliconLabs_BGApp/OTAFiles/(ProjectFolder)/
(iPhone) Copy the .gbl file to your Google Drive
2. Download the Blue Gecko App on your phone
3. Go to Bluetooth Browser and Connect to your device
4. After connecting select OTA in the menu in the top right corner.
5. Select the OTA files and do a full or app only OTA depending on the changes in the project