SILICON LABS (/community)

silabs.com (https://www.silabs.com/) | Log In | Register

简体中文 日本語

MCUs Wireless More Products Search silabs.com

Development Tools Expert's Corner

Community (/community) » Bluetooth (/community/wireless/bluetooth)

» Knowledge Base (/community/wireless/bluetooth/knowledge-base.html)

» KBA_BT_0804: Secure OTA DFU

Bluetooth Knowledge Base

Search Bluetooth Knowledge Base

KBA_BT_0804: Secure OTA DFU

06/22/2017 | 08:06 am

%20security%20features%3A%0A%0A%0A%09Authenticated%20(signed)%20upgrade%20file

ecked%20while%20upgrading.

%20stack%20%2B%20Supervisor%20(provided%20as%20precompiled%20binary%20code)
%20%2B%20user%20application
olvement%20from%20the%20user%20application.
already%20have%20OTA%20support%20built%20into%20the%20code.%20In%20these%20examples

0the%20Stack%20%2F%20Apploader%20is%20started%0A%09%0A%09The%20Stack%20%2F
rocess%20is%20done%20in%203%20steps%3A%0A%0A%09%0A%09
20%2F%20Application%20is%20overwritten!%0A%09%09%0A
ng%20this%20upload%20process.)%0A%09%0A%09%0A%0A
20EFR32xG1%20and%20BGM11x%20Series%20Products.
ferent%20bootloader%20configurations%20are%20recommended%20for%20different%20devices
C2%A0or
knowledge-base.entry.html%2F2017%2F09%2F20%2Fuploading_imagesto-DXxD%0A
nstalled%20SDK%2C%20click%20Next
%20security%20features%0A%0A%0A%09Open%20the%20Plugins%20tab
ated%20by%20the%20AppBuilder%2C%20build%20your%20project%3A%0A%0A%0A

Secure%20Boot.
A%09commander%20gbl%20keygen%20--type%20ecc-p256%20--outfile%20app-sign-key.pem
09%C2%A0%0A%09commander%20gbl%20keygen%20--type%20aes-ccm%20--outfile%20app-
%20app-sign-key.pem-tokens.txt%0A%0A%09%C2%A0%0A%09%0A
pports%20OTA%20DFU%3A%0A%0A%0A%09Open%20Simplicity%20Studio
%20your%20project%2C%20click%20Next
rt%20the%20device%20in%20DFU%20mode).%20To%20build%20the%20application%3A%0A
eparately
%20in%20your%20project

Merge%20the%20bootloader%20image%20with%20the%20application%0A%09%0A%09
e%20Application%20images%0A%09%09commander%20convert%20bootloader-storage-internal-
A%0A%C2%A0%0A%0A%C2%A0%0A
ur%20device%20in%20the%20Devices%20or%20Debug%20Adapters%20tab
20toolchain.%20Click%20Finish%0A%0A%0ABuild%20Application
2Fencrypt%20it%0A%0A%0A%09Copy%20app-sign-key.pem%20and%20app-encrypt-

op%20(%20http%3A%2F%2Fsilabs.com%2Fbluegeckoapp%20)%20to%20your%20smartphone

arkalvac (/community
/profile.html
/home/users
/community
/r/-VGez6laPzNQGv2
exNGX/profile)
Employee

1. Introduction

The following article shows how to securely upgrade Bluetooth application OTA (over-the-air) using signed+encrypted upgrade files. The process is tested with Bluetooth SDK v2.8.1.

Gecko Bootloader

The Gecko Bootloader is a common bootloader for all Silicon Labs protocol stacks. It can load applications from different sources (internal flash, external flash, UART, SPI, over-the-air) using different protocols (XMODEM, BGAPI, EZSP SPI, Bluetooth etc.). It can be configured in a number of ways and its capabilities depend on the current configuration. In this training we demonstrate, how you can use it for loading a new application into the device sent over a Bluetooth connection.

Security features

The Gecko Bootloader has three security features:

- Authenticated (signed) upgrade file
- Encrypted upgrade file
- Secure Boot

Upgrade files are in a custom GBL (Gecko Bootloader) format. An authenticated upgrade file means that an electronic signature is attached to the GBL file. The signature is produced with a public-private key pair. The public key is stored in the device, while the private key is kept secret by the manufacturer. The signature ensures that the upgrade file is from a trusted source.

An encrypted upgrade file means that the content of the GBL file is encrypted to protect against eavesdroppers.

Secure Boot means that a signature is attached to the firmware image (.s37) BEFORE it is packed into upgrade file formal (.gbl). Note that this differs from authenticated upgrade file, as authenticated upgrade file means, that a signature is attached to the upgrade file AFTER the image was packed into GBL format. A signed

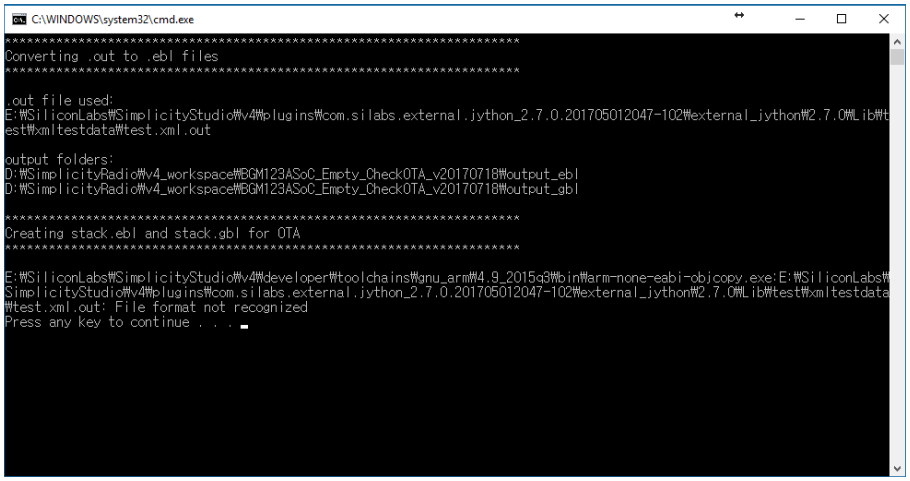
ps_eddy (/community/profile.html/home/users/community/k/KiqgruO7HwYI0oNoW_0R/profile)

Replied Jul 18 2017, 9:21 PM

0

Hello arkalvac,

I'm currently having some problem on replicating your tutorial into my BLE project. I have copied app-sign-key.pem and app-encrypt-key.txt into the Bluetooth project as shown in **Create Bluetooth app with secure OTA DFU capability**. However, instead of stack-signed.gbl and app-signed.gbl files, I got this message.



is there any step that I missed?

Thank you.

Eddy.

ps eddy
(/community
/profile.html
/home/users
/community
/k/Kiqgruo7Hw
YI0oNoW_0R
/profile)

Replied Jul 19 2017, 1:30 AM

0

Hello again,

Sorry about my stupidity. I double-clicked the .bat files from Simplicity studio, instead of manually going to the project folder and run it from there. Problem solved.

Mounika
(/community
/profile.html
/home/users
/_/_Tr6UjXg3h
EBaP671oJC
/profile)

Replied Aug 08 2017, 1:08 PM

0

I have an application which has 2 slots. 1 slot has BLE image, 2nd slot is a proprietary image.So can I update the image in 2nd slot while in Bluetooth(1st slot) with OTA?

arkalvac
(/community
/profile.html
/home/users
/community
/r/-VGez6laPz
NQGv2exNGX
/profile)
Employee

Replied Aug 14 2017, 10:16 AM

0

Hi,

In your case you have 3 slots in the flash: the running application, slot0, slot1.

This is described in this article:
http://community.silabs.com/t5/Bluetooth-Wi-Fi-Knowledge-Base/Switching-between-firmware-images-using-Internal-Storage/ta-p/204712 (http://community.silabs.com/t5/Bluetooth-Wi-Fi-Knowledge-Base/Switching-between-firmware-images-using-Internal-Storage/ta-p/204712)

When you are doing an OTA DFU, it will upgrade the running application, and it has nothing to do with the slots.

If you want to upgrade the slots with new images via Bluetooth, you have to implement an application level bootloader, i.e. you have to solve the receiving of .gbl files via Bluetooth in your application, and you have to store the received data in the flash on the proper address. (Also if you use encrypted/signed images, you have to decrypt them and check the signature.) Then you can command the bootloader to reboot from slot0 or slot1.

As a reference code for this you can use the "SOC - Switched Multiprotocol Joining Device" software example which implements an application level bootloader to populate the slots with upgrade images.

pa1329 (/community/profile.html/home/users/c/c8s88lG4vZnzc3T4GSIt/profile)

Replied Aug 21 2017, 12:40 PM

0

After copying the bootloader image that ends with -combined.s37 into my output folder, and running the command that you said, I get the following:

PS C:\SiliconLabs\SimplicityStudio\v4\developer \adapter_packs\commander> ./commander convert bootloader-storage-internal-single-512k-combined.s37 stack-signed.gbl app-signed.gbl --
outfile bootloader+stack+app.hex
Parsing file bootloader-storage-internal-single-512k-combined.s37...
ERROR: Could not open file: bootloader-storage-internal-single-512k-combined.s37.
DONE

Do you maybe know what the problem could be?

Thank you!

arkalvac (/community/profile.html/home/users/community/r/-VGez6laPzNQGv2exNGX/profile)
Employee

Replied Aug 22 2017, 4:36 AM

0

Hi,

Apparently you are running the Commander from the Commander directory and not from the output_gbl directory. In this case you have to give the full paths of the images. But it's easier to navigate to the output_gbl directory and run Commander from there. (You may have to add Commander path to the PATH environmental variable, if it is not yet added, to access the Commander command from any directory)

pa1329 (/community/profile.html/home/users/c/c8s88lG4vZnzc3T4GSIt/profile)

Replied Aug 22 2017, 5:06 AM

0

Ok yeah that was a stupid mistake. Thanks for that.

Is there a way of uploading a new upgrade .gbl file to the device without UART or BT? Just through Simplicity Studio?

arkalvac (/community/profile.html/home/users/community/r/-VGez6laPzNQGv2exNGX/profile)
Employee

Replied Aug 22 2017, 5:16 AM

0

Yes, there is. You can flash the images with Simplicity Commander to any address after renaming them to .bin.

Please walk through this article, I think you will find everything there: http://community.silabs.com/t5/Bluetooth-Wi-Fi-Knowledge-Base/Switching-between-firmware-images-using-Internal-Storage/ta-p/204712 (http://community.silabs.com/t5/Bluetooth-Wi-Fi-Knowledge-Base/Switching-between-firmware-images-using-Internal-Storage/ta-p/204712)

bhushan_patil (/community/profile.html/home/users/y/y5tNGKXi1DKR-QdILCZm/profile)

Replied Sep 12 2017, 7:36 AM

0

Hello Arkalvac,

Good Morning

Very useful details and very helpful in my project.

I followed the steps you had mentioned and I am able to perform the OTA from Blue Gecko app.

As I understood the bootloader used in above example is **Internal Storage Bootloader (single image)** with Slot 0 for storage. I have read the flash and could able to see Slot 0 holds the new OTA image.

I have tried using the bootloader **Internal Storage Bootloader (Multiple image)** with Slot 0 and Slot 1 for storage. But what I see is every time only Slot0 is updated with new OTA where as Slot 1 is intact. Looks like DFU updates the Slot 0 only when triggered an OTA from Blue Gecko. I would like to modify the code in such way that I could able to store the OTA image either in slot 0 or Slot 1.

Thanks for your help.

Board used is,

EFR32MG12P332F1024GL125 Mighty Gecko

Regards,

Bhushan

Replied Sep 14 2017, 12:41 PM

0

arkalvac (/community/profile.html/home/users/community/r/-VGez6laPzNQGv2exNGX/profile)
Employee

Hi Bhushan,

The OTA supervisor code is implemented in the stack and it uses only slot 0, so you cannot change it, to use slot 1 as well.

However you can upload images to any slot with user application using the same OTA protocol.

I'm just about writing an article about this, including example code. Please give me some days.

Arnold

1 2 3 Next

Copyright © 2019 Silicon Laboratories. All rights reserved.

粤ICP备15107361号-1 (http://www.miibeian.gov.cn)