



NORDIC[®]
SEMICONDUCTOR

SECURE-DFU

Device Firmware Update

CONTENTS

Secure-DFU

Contents

소개

Secure-DFU란?

빠른 Secure DFU 사용 가이드

Secure DFU

첫번째 : Secure-DFU 준비하기

두번째 : Security Key 생성

세번째 : Bootloader 컴파일

업데이트할 Firmware Package 생성하기

준비 사항

Package 생성

부가사항

양산을 위한 통합된 펌웨어 만들기

첫번째 : Bootloader Settings File 생성

두번째 : 하나의 파일로 통합 생성

Firmware Package UPDATE

Target Board에 펌웨어 다운로드

Mobile 또는 nRF Connect 를 통해 무선 펌웨어 업데이트

nRF Connect Desktop 이용한 방법

nRF Connect Mobile 을 이용한 방법

nrfutil.exe 를 이용하여 **Command Line**으로 업데이트하는 방법

Serial을 통한 유선 업데이트

참조 문서

소개

Secure-DFU란?



Nordic SDK 에서 제공되는 Firmware Update 기능으로 Bluetooth Low Energy/Bluetooth Mesh/Thread/ZigBee와 같은 무선 프로토콜을 이용하는 OTA Upgrade 기능 및 UART/SPI와 같은 wired Serial Interface를 통해 생산 제품에 펌웨어를 업데이트 하는 기능

Secure DFU는 기존의 Device Firmware Upgrade 기능과는 다르게 Firmware 를 Signing하여 양단의 키 값이 맞지 않을 경우 사용하지 못하도록 하는 기능이 추가됨

- PC 버전과 Mobile App 버전에 대한 프로그램 및 소스코드 제공
 - [Android Google Play](#)
 - [Apple App Store](#)
 - [PC version \(nRF Connect\)](#)

빠른 SECURE DFU 사용 가이드

1. script를 이용하여 micro-ecc compile
2. nrfutil 설치
3. nrfutil를 이용하여 Private/Public key 생성
4. 생성된 Public Key를 DFU(Bootloader) firmware code에 삽입하여 DFU 예제 컴파일
5. nrfutil을 이용하여 bootloader settings 파일 생성
6. mergehex를 통해 bootloader settings, softdevice, application, bootloader 를 하나의 파일로 통합
7. 통합된 펌웨어를 Target Device에 다운로드
8. nrfutil을 이용하여 새롭게 업데이트 할 Firmware package 생성
9. 생성된 Firmware Package를 nRF Connect for Mobile, nRF Connect for Desktop 또는 nRFToolBox App을 이용하여 Target Device에 무선으로 업데이트

SECURE DFU

첫번째 : Secure-DFU 준비하기

- Python PIP (PIP 가 설치되지 않았을 경우 참조 [here](#)) * Python 3.0 지원안됨
- ARM 용 **GCC 컴파일러** 설치 (4.9-2015-q3-update 이상)
- Make 설치 (use **MinGW**, **GNU Make**, or **Xcode**) - 쉬운 설정을 위해 **GNU Make**를 설치할 권장
- nrfutil

Python (2.7) 이 설치되어 있을 경우 (Windows/Linux 동일)

```
$ pip install nrfutil
```

```
(base) C:\Users\Eugene>pip install nrfutil
Installing collected packages: nrfutil
Successfully installed nrfutil-5.1.0
```

실행 파일 직접 [다운로드](#)

- micro-ecc library
 - SDK 에서 제공되는 Script File을 이용하여 Library Download 및 Build
 - 위치 : nRF5_SDK_x.x.x_xxxxxx\external\micro-ecc\
 - build_all.bat (Windows) 또는 build_all.sh (Linux/macOS) 실행
 - 빌드시 아래와 같은 에러가 날 경우 SDK에서 설정된 Compiler와 설치된 gcc 버전의 차이가 있으므로 SDK 의 Makefile 관련 내용의 수정이 필요

문제현상

```
D:\13_SDK_NRF5\NRF5_SDK_15.3.0_59ac345\external\micro-ecc>build_all.bat
make: Entering directory 'D:/13_SDK_NRF5/NRF5_SDK_15.3.0_59ac345/external/micro-ecc/nrf51_armgcc/armgcc'
process_begin: CreateProcess(NULL, "C:/Program Files (x86)/GNU Tools ARM Embedded/7 2018-q2-update/bin/arm-none-eabi-gcc" --version, ...) failed.
make: ../././././components/toolchain/gcc/Makefile.common:129: pipe: No error
Cannot find: 'C:/Program Files (x86)/GNU Tools ARM Embedded/7 2018-q2-update/bin/arm-none-eabi-gcc'.
Please set values in: "D:/13_SDK_NRF5/NRF5_SDK_15.3.0_59ac345/components/toolchain/gcc/Makefile.windows"
according to the actual configuration of your system.
../././././components/toolchain/gcc/Makefile.common:129: *** Cannot continue. Stop.
make: Leaving directory 'D:/13_SDK_NRF5/NRF5_SDK_15.3.0_59ac345/external/micro-ecc/nrf51_armgcc/armgcc'
```

해결방안

위의 그림에서 가이드하는 바와 같이 nRF5_SDK\components\toolchain\gcc\Makefile.windows 의 내용중 Compiler 관련 내용을 사용자 환경에 맞게 변경하거나 SDK 맞는 버전을 다운로드 하여 설치

```
GNU_INSTALL_ROOT := C:/Program Files (x86)/GNU Tools
ARM Embedded/6 2017-q2-update/bin/
GNU_VERSION := 7.3.1
GNU_PREFIX := arm-none-eabi
```

두번째 : Security Key 생성

nrfutil 을 통한 Private Key 생성하기

```
nrfutil keys generate priv.pem
```

Private Key로 부터 Public Key 생성하기

```
nrfutil keys display --key pk --format code priv.pem --out_file  
public_key.c  
또는  
nrfutil keys display --key pk --format code priv.pem
```

```
(base) C:\Users\Eugene>nrfutil keys display --key pk --format code key  
/* This file was automatically generated by nrfutil on 2019-03-19 (YY-MM-DD) at 12:56:28 */  
#include "stdint.h"  
#include "compiler_abstraction.h"  
/** @brief Public key used to verify DFU images */  
__ALIGN(4) const uint8_t pk[64] =  
{  
    0xa3, 0x69, 0xaa, 0x07, 0xbc, 0x83, 0xf5, 0x94, 0xac, 0x2b, 0x58, 0x8a, 0x08, 0xa9, 0x0e, 0x85, 0x46, 0x51, 0x0e, 0x17, 0x8e, 0x88, 0x5f, 0xba, 0xdf, 0x59, 0x44, 0x84, 0x43, 0xc6, 0xf9, 0xd6,  
    0x05, 0x3b, 0x19, 0xf0, 0xa4, 0xf3, 0xe1, 0xed, 0xab, 0x77, 0x98, 0x76, 0xf8, 0x00, 0xfa, 0xc8, 0xa2, 0xac, 0x44, 0xc7, 0x68, 0x7a, 0x80, 0xf3, 0xf5, 0x8e, 0xbf, 0xec, 0xb6, 0xb0, 0xef, 0xe3  
};
```

세번째 : Bootloader 컴파일

위 단계에서 생성된 Public Key 를 코드형태로 Bootloader Firmware (SDK\examples\dfu\bootloader_secure_ble)에 삽입(dfu_public_key.c)하여 컴파일

```
#include "sdk_config.h"  
#include "stdint.h"  
#include "compiler_abstraction.h"  
  
#if NRF_CRYPTO_BACKEND_OBERON_ENABLED  
/* Oberon backend is changing endianness thus public key must be kept  
in RAM. */  
#define _PK_CONST  
#else  
#define _PK_CONST const  
#endif  
  
/** @brief Public key used to verify DFU images */  
__ALIGN(4) const uint8_t pk[64] =  
{  
    0xa3, 0x69, 0xaa, 0x07, 0xbc, 0x83, 0xf5, 0x94, 0xac, 0x2b, 0x58,  
    0x8a, 0x08, 0xa9, 0x0e, 0x85, 0x46, 0x51, 0x0e, 0x17, 0x8e, 0x88,  
    0x5f, 0xba, 0xdf, 0x59, 0x44, 0x84, 0x43, 0xc6, 0xf9, 0xd6,  
    0x05, 0x3b, 0x19, 0xf0, 0xa4, 0xf3, 0xe1, 0xed, 0xab, 0x77, 0x98,  
    0x76, 0xf8, 0x00, 0xfa, 0xc8, 0xa2, 0xac, 0x44, 0xc7, 0x68, 0x7a,  
    0x80, 0xf3, 0xf5, 0x8e, 0xbf, 0xec, 0xb6, 0xb0, 0xef, 0xe3  
};
```

업데이트할 FIRMWARE PACKAGE 생성하기

준비 사항

- 업데이트 펌웨어 및 Private Key file
- Softdevice version information (예, S132 v6.1.1 == 0xB7)
 - <https://github.com/NordicSemiconductor/pc-nrfutil#usage>
 - nrfutil를 통해서 확인

```
nrfutil pkg generate --help
```

Package 생성

```
nrfutil pkg generate --hw-version 52 --sd-req 0xB7 --application-version 1 --application nrf52832_xxaa.hex --key-file priv.pem app_dfu_package.zip
```

부가사항

양산을 위한 통합된 펌웨어 만들기

첫번째 : Bootloader Settings File 생성

Application Firmware가 정상적으로 Flash 된 것인지를 확인하기 위해 사용되는 Setting 값들에 대한 파일을 생성

Firmware Download 시 Bootloader Settings 파일이 추가되지 않았을 경우 Bootloader에서 Application 영역으로 실행주소를 옮기지 못하는 문제가 발생되므로 반드시 생성해서 같이 Download 해야 함

```
nrfutil settings generate --family NRF52 --application <application.hex> --application-version 1 --bootloader-version 1 --bl-settings-version 1 bootloader_settings.hex
```

두번째 : 하나의 파일로 통합 생성

생성된 Bootloader Settings 파일은 Application Firmware 와 함께 통합하여 nRF5x 에 다운로드

```
mergehex -m <application.hex> <bootloader_settings.hex> -o <output file.hex>
```

Softdevice 및 Bootloader를 하나로 통합할 경우

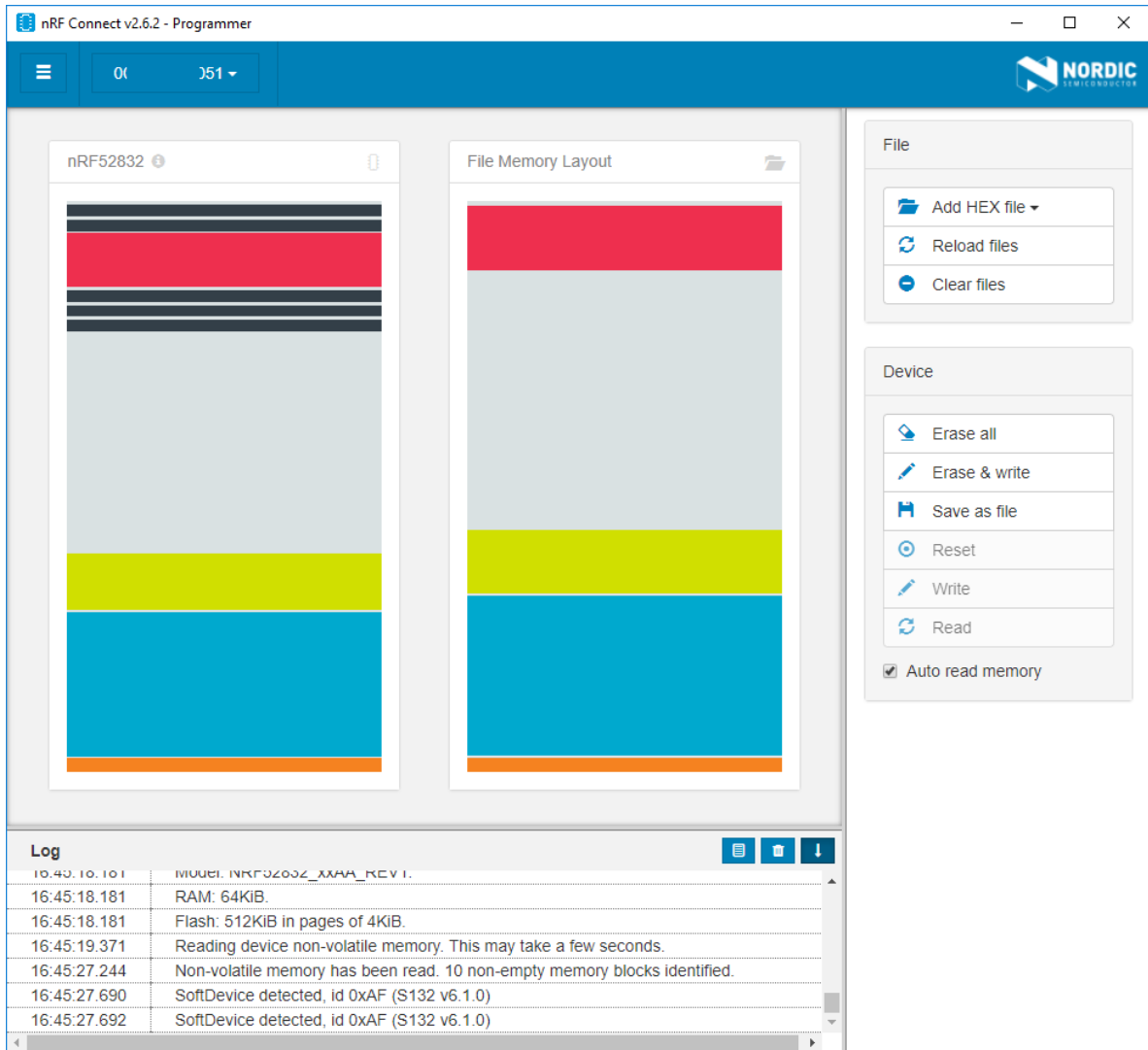
```
mergehex -m <application.hex> <bootloader_settings.hex> <bootloader.hex> <softdevice.hex> -o <output file.hex>
```

FIRMWARE PACKAGE UPDATE

위에서 만든 Firmware Package와 Mergehex를 통해 통합된 펌웨어를 어떻게 업데이트 하는지에 대한 설명

Target Board에 펌웨어 다운로드

Bootloader Settings 와 함께 Mergehex 를 통해 통합된 펌웨어를 Programmer 또는 nrfjprog 를 통해 다운로드



또는

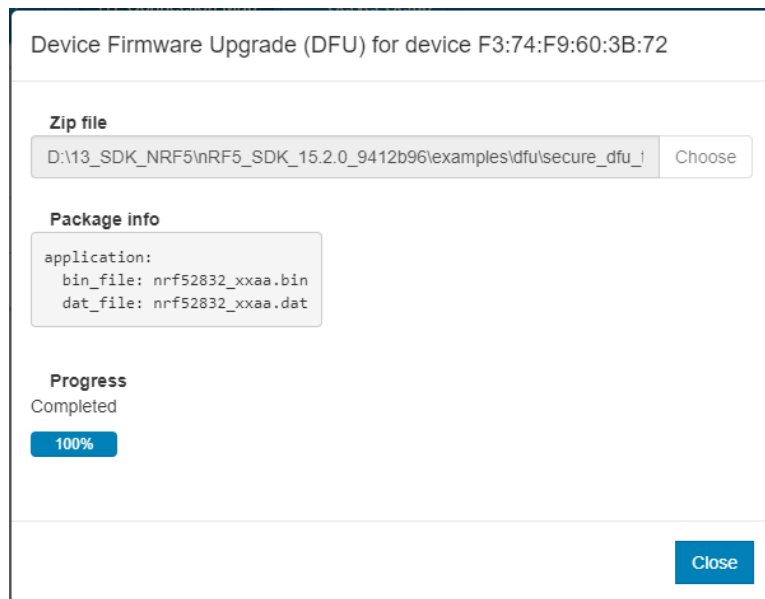
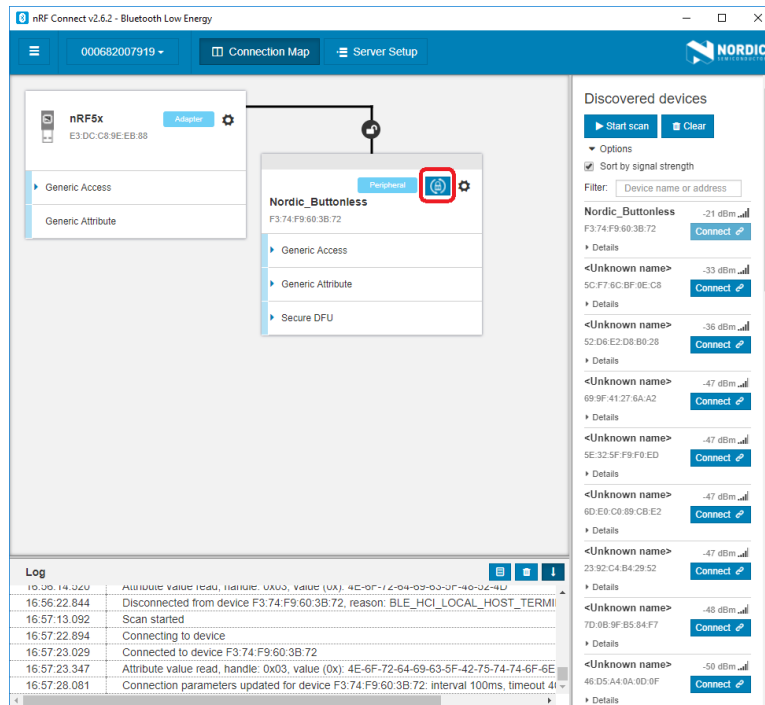
```
nrfjprog -f nrf52 --chiperase --reset --program <target firmware hex>
```

Mobile 또는 nRF Connect 를 통해 무선 펌웨어 업데이트

nRF Connect Desktop 이용한 방법

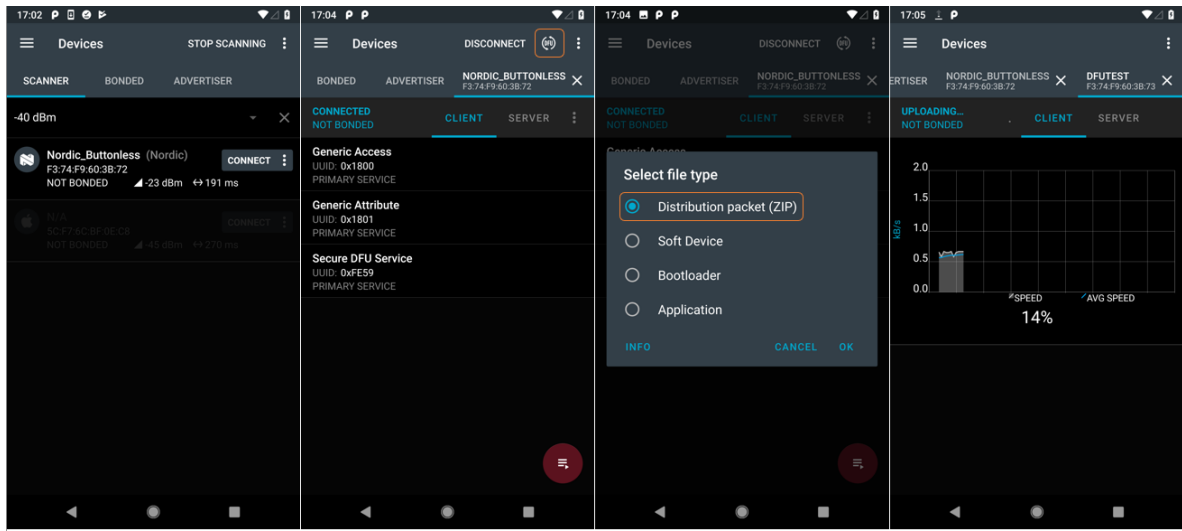
1. nRF Connect for Desktop을 설치

2. In-App 인 Bluetooth Low Energy 를 설치
3. nRF52-DK 또는 nRF52840-DK를 PC에 연결
4. Port Menu에서 연결된 DK의 Serial 을 선택한 후 Start Scan 하여 Target Device를 연결
5. 연결된 Target Device 항목에 DFU Icon을 눌러 업데이트할 펌웨어를 선택 후 업데이트



nRF Connect Mobile 을 이용한 방법

1. nRF Connect for Mobile을 App store 또는 Play Market을 통해 다운로드
2. nRF Connect for Mobile을 실행하여 연결하고자 하는 디바이스를 검색하여 연결
3. DFU Menu를 선택하여 Distribution packet(ZIP) 을 선택하여 펌웨어 업데이트



nrfutil.exe 를 이용하여 Command Line으로 업데이트하는 방법

PC에 nRF52-DK, nRF52840-DK 또는 nRF52840-Dongle을 연결하여 디바이스를 업데이트 (COM port 3를 이용할 경우)

```
nrfutil dfu ble -ic NRF52 -pkg app_dfu_package.zip -p COM3 -n "MyDevice" -f
```

Serial을 통한 유선 업데이트

COM port를 통해 연결된 디바이스를 nrfutil.exe를 통해서 아래의 커맨드로 업데이트 (COM port 3를 이용할 경우)

```
nrfutil dfu serial -pkg app_dfu_package.zip -p COM3
```

참조 문서

- [BLE Secure DFU Bootloader Guide](#)
- [Getting started with Nordic's Secure DFU bootloader, a step by step guide](#)
- <https://github.com/NordicSemiconductor/pc-nrfutil#commands>