



Device Firmware Update Sample Application Guide

Rev. 1.0 2023/3/31

内容

1	はじめに	5
2	概要	5
2.1	使用するツール	5
2.2	Bootloader	6
2.3	Bootloader の流れについて	7
3	DFU 構造	8
3.1	DFU の動作	8
3.2	書き込み FLASH 領域	8
3.3	DFU のパッケージ	9
3.4	秘密鍵ファイル	10
3.4.1	秘密鍵ファイル作成のツールインストール	10
3.4.2	秘密鍵ファイル作成手順	10
4	コンパイル	11
4.1	Nordic SDK への組み込み	11
4.2	UART 通信	12
4.2.1	UART 通信 Pin	13
4.2.2	UART 通信 Baud rate	14
4.2.3	通知 Pin 変更	15
4.3	micro-ecc について	15
4.4	モジュールの選択	16
5	ファームウェアの書き込み	17

5.1	J-link を使った書き込み.....	17
5.1.1	PC と評価ボードの接続.....	17
5.1.2	nRF Command Line Tools での書き込み	18
5.1.3	nRF Connect Programmer での書き込み	19
5.1.4	Segger Embedded Studio での書き込み	20

改版履歴

改訂	日付	コメント
1.0	2023/3/31	初版

1 はじめに

このドキュメントでは、加賀 FEI の BLE モジュールに適応する DFU(Device Firmware Update) sample Application について記載しています。

2 概要

2.1 使用するツール

本アプリケーションでは下記を使用することを前提としています。

- Nordic SDK
- nRF Util ※install する過程で Python3.7 ・ Pip を使用します
- SES(Segger Embedded Studio)および NRF ボードサポートパッケージ（インストールおよびセットアップ手順については、Segger Embedded Studio の NRF のクイックスタートガイド『SES_NRF52xxx_Quickstart_Guide_X_XX_JP.pdf』をこちらからログインしてダウンロードしてください）
- 書き込みツール(nRF Command Line Tools/ nRF Programmer)
- サポートされている加賀 FEI の BLE モジュール

Module	Chip
EC2820AA2	nRF52820
ES2820AA2	nRF52820
EC2832AAx	nRF52832
EJ2832AAx	nRF52832
ES2832AAx	nRF52832
EB2833AAx	nRF52833
ED2833AA2	nRF52833
EJ2833AA2	nRF52833
EB2840AAx	nRF52840
ED2840AAx	nRF52840
EJ2840AAx	nRF52840

- J-link Lite

評価キットで EVB とセットで販売しております。必要に応じてお求めください。



2.2 Bootloader

DFU Sample Application の Bootloader には DFU 機能を含んでおり、特定の操作を行うことで DFU の処理に移行することが可能です。構成の詳細については下記 Nordic URL を参照下さい。

参考 URL :

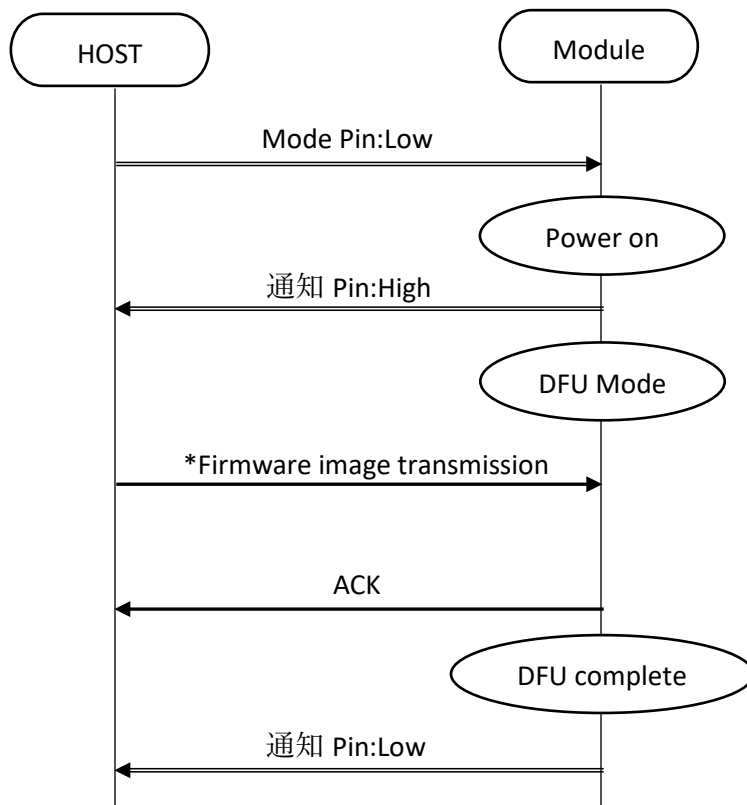
https://infocenter.nordicsemi.com/topic/sdk_nrf5_v17.1.0/lib_bootloader_modules.html

※このサンプルアプリケーションでは SDK17.1.0 をベースとしているため mbr を書き込んでおりますが、使用される softdevice はお客様の責任において書き込みをお願い致します。

2.3 Bootloader の流れについて

Bootloader は下記フローチャートで処理を行います。各モジュールの通知 Pin、DFU Mode に入る Pin については表をご確認ください。使用する Pin の変更方法については、「4.2.3 通知 Pin 変更」を参照してください。

[フローチャート]



Module	Mode Pin	通知 Pin
EC2820AA2	P0.14	P0.03
ES2820AA2	P0.14	P0.03
EC2832AAx	P0.18	P0.20
EJ2832AAx	P0.18	P0.20
ES2832AAx	P0.18	P0.20
EB2833AAx	P0.28	P0.30
ED2833AA2	P0.28	P0.30
EJ2833AA2	P0.28	P0.30
EB2840AAx	P0.25	P0.20
ED2840AAx	P0.28	P0.29
EJ2840AAx	P0.28	P0.29

3 DFU 構造

3.1 DFU の動作

DFU Sample Application では MBR、Bootloader 及びその設定値の領域を FLASH に書き込む設定になっています。

URL : https://infocenter.nordicsemi.com/topic/sdk_nrf5_v17.1.0/lib_bootloader.html

3.2 書き込み FLASH 領域

DFU Sample Application の書き込み動作について、通常書き込みは Dual Bank で行われますが、領域が不足する場合は SingleBank に変更して動作を行います。詳細については下記 Nordic InfoCenter をご参照下さい。

URL : https://infocenter.nordicsemi.com/topic/sdk_nrf5_v17.1.0/lib_bootloader_dfu_banks.html

3.3 DFU のパッケージ

DFU で使用する通信パッケージの詳細については下記 Nordic InfoCenter をご参照下さい。

URL : https://infocenter.nordicsemi.com/topic/sdk_nrf5_v17.1.0/lib_dfu_transport.html

※上記 URL は SDK17.1.0 になります。ご使用のバージョン合わせてご確認ください。

3.4 秘密鍵ファイル

DFU では秘密鍵ファイルを使用して第 3 者による書き込みを防止しています。

3.4.1 秘密鍵ファイル作成のツールインストール

DFU で使用する秘密鍵ファイルは、Nordic の”nRF Util”を使用して作成を行います。インストールの詳細については下記 URL を参照してください。

URL : https://infocenter.nordicsemi.com/topic/ug_nrfutil/UG/nrfutil/nrfutil_installing.html

3.4.2 秘密鍵ファイル作成手順

下記手順で秘密鍵ファイルを作成し、秘密鍵ファイルから DFU Sample Application に組み込みソースファイルを生成します。作成したファイル(ex: `dfu_public_key.c`)は解凍したフォルダの直下にコピーしてください。

[秘密鍵作成]

コマンド : `nrfutil keys generate XXXXXXXX.pem`

※作成された秘密鍵ファイルはファームウェアの書き換えに使用します。管理方法にはご注意ください。

[Bootloader 組み込みファイル作成]

コマンド : `nrfutil keys display --key pk --format code XXXXXXXX.pem --out_file dfu_public_key.c`

※出力ファイル名を変更される場合、プロジェクトに設定されているファイルも変更してご使用ください

4 コンパイル

4.1 Nordic SDK への組み込み

DFU Sample Application では、Nordic SDK を使用しています。下記バージョンの Nordic SDK をダウンロード・解凍して、DFU Sample Application のフォルダを「nRF5_SDK_17.1.0_ddde560\examples」の下にコピーしてご使用ください。

DFU Sample Application では、下記 SDK バージョンを使用していますが、開発に使用する際には最新のバージョンでのご検討をお願い致します。

また『3.4.2 秘密鍵ファイル作成手順』で作成したソースファイルを DFU Sample Application のフォルダにコピーしてください。

SDK Version : 17.1.0 nRF5 SDK

ダウンロード URL : <https://www.nordicsemi.com/Software-and-tools/Software/nRF5-SDK/Download#infotabs>

※Nordic のソースコードの著作権は Nordic に帰属します

4.2 UART 通信

DFU Sample Application では、EVB で PC と接続して評価可能なように下記仕様で UART の Pin を設定しています。また UART の通信仕様については Nordic SDK の仕様に沿って下記設定になっています。

[モジュールの UART Pin 配置]

Module	RX	TX	CTS	RTS
EC2820AA2	P0.04	P0.05	P0.06	P0.07
ES2820AA2	P0.08	P0.06	P0.07	P0.05
EC2832AAx	P0.08	P0.06	P0.07	P0.05
EJ2832AAx	P0.08	P0.06	P0.07	P0.05
ES2832AAx	P0.08	P0.06	P0.07	P0.05
EB2833AAx	P0.08	P0.06	P0.07	P0.05
ED2833AA2	P0.08	P0.06	P0.07	P0.05
EJ2833AA2	P0.08	P0.06	P0.07	P0.05
EB2840AAx	P0.08	P0.06	P0.07	P0.05
ED2840AAx	P0.08	P0.06	P0.07	P0.05
EJ2840AAx	P0.08	P0.06	P0.07	P0.05

[UART 通信仕様]

Baud rate	115200bps
Data	8bit
Parity	None
Stop	1bit
Hardware flow control	Enable

4.2.1 UART 通信 Pin

UART 通信で使用する Pin を変更する場合、下記定義を変更することで可能になります。

[UART Pin 設定箇所(nrf_dfu_serial_uart.c)]

```

192     nrf_drv_uart_config_t uart_config = NRF_DRV_UART_DEFAULT_CONFIG;
193
194     uart_config.pseltxd  = TX_PIN_NUMBER;
195     uart_config.pselrxd  = RX_PIN_NUMBER;
196     uart_config.pselcts  = CTS_PIN_NUMBER;
197     uart_config.pselrts  = RTS_PIN_NUMBER;
198     uart_config.hwfc     = NRF_DFU_SERIAL_UART_USES_HWFC ?
199                         NRF_UART_HWFC_ENABLED : NRF_UART_HWFC_DISABLED;
200     uart_config.p_context = &m_serial;
201
202     err_code = nrf_drv_uart_init(&m_uart, &uart_config, uart_event_handler);

```

[UART Pin 定義箇所(pca10040.h/pca10056.h/pca10100.h/module_config.h)]

```

89 #define RX_PIN_NUMBER 8
90 #define TX_PIN_NUMBER 6
91 #define CTS_PIN_NUMBER 7
92 #define RTS_PIN_NUMBER 5
93 #define HWFC true

```

※その他の example プロジェクトでも参照しているためご注意ください。

Module	ヘッダーファイル
EC2820AA2	module_config.h
ES2820AA2	pca10100.h
EC2832AAx	pca10040.h
EJ2832AAx	pca10040.h
ES2832AAx	pca10040.h
EB2833AAx	pca10100.h
ED2833AA2	pca10100.h
EJ2833AA2	pca10100.h
EB2840AAx	pca10056.h
ED2840AAx	pca10056.h
EJ2840AAx	pca10056.h

※EC2820AA2 のみ module_config.h で UART Pin の定義を行っています。

4.2.2 UART 通信 Baud rate

UART 通信で使用する Baud rate を変更する場合、下記定義を変更することで可能になります。

[UART Baudrate 設定箇所(nrf_drv_uart.h)]

```

209 /**@brief UART default configuration. */
210 #define NRF_DRV_UART_DEFAULT_CONFIG
211 {
212     .pseltxd          = NRF_UART_PSEL_DISCONNECTED,
213     .pselrxd          = NRF_UART_PSEL_DISCONNECTED,
214     .pselcts          = NRF_UART_PSEL_DISCONNECTED,
215     .pselrts          = NRF_UART_PSEL_DISCONNECTED,
216     .p_context        = NULL,
217     .hwfc              = (nrf_uart_hwfc_t)UART_DEFAULT_CONFIG_HWFC,
218     .parity            = (nrf_uart_parity_t)UART_DEFAULT_CONFIG_PARITY,
219     .baudrate          = (nrf_uart_baudrate_t)UART_DEFAULT_CONFIG_BAUDRATE,
220     .interrupt_priority = UART_DEFAULT_CONFIG_IRQ_PRIORITY,
221     NRF_DRV_UART_DEFAULT_CONFIG_USE_EASY_DMA
222 }

```

[UART Baudrate 定義箇所("config_32" or "config_40" or "config_20" or "config_33"
/sdk_config.h)]

```

1723 // <o> UART_DEFAULT_CONFIG_BAUDRATE - Default Baudrate
1724
1725 // <323584=> 1200 baud
1726 // <643072=> 2400 baud
1727 // <1290240=> 4800 baud
1728 // <2576384=> 9600 baud
1729 // <3862528=> 14400 baud
1730 // <5152768=> 19200 baud
1731 // <7716864=> 28800 baud
1732 // <10289152=> 38400 baud
1733 // <15400960=> 57600 baud
1734 // <20615168=> 76800 baud
1735 // <30801920=> 115200 baud
1736 // <61865984=> 230400 baud
1737 // <67108864=> 250000 baud
1738 // <121634816=> 460800 baud
1739 // <251658240=> 921600 baud
1740 // <268435456=> 1000000 baud
1741
1742 #ifndef UART_DEFAULT_CONFIG_BAUDRATE
1743 #define UART_DEFAULT_CONFIG_BAUDRATE 30801920
1744 #endif

```

※変更する値についてコメント欄に記載されている値を参考に变更してください。定義のファイルは使用するモジュールによって異なります。

4.2.3 通知 Pin 変更

DFU モードに入る Pin、DFU もの状態を通知する Pin は下記にて定義されています。変更する場合は下記定義を変更してください。

[Pin 定義箇所(module_config.h)]

```
13 #define DFU_ENTER_PIN      18
14 #define DFU_NOTICE_PIN    20
```

4.3 micro-ecc について

Sample ソフトをコンパイルするには micro-ecc が必要となります。下記 Nordic の手順に従ってライブラリを作成してください

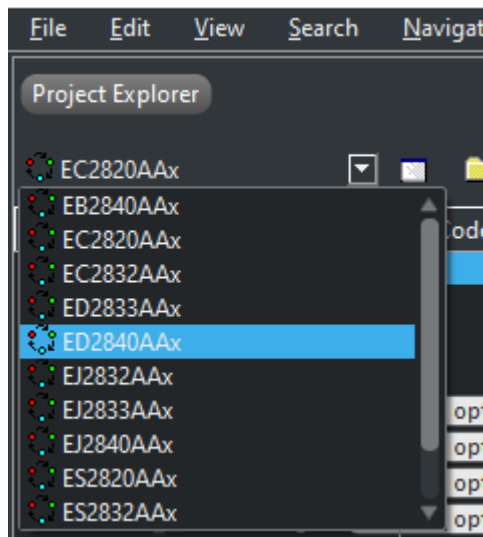
URL :

https://infocenter.nordicsemi.com/topic/sdk_nrf5_v17.1.0/lib_crypto_backend_micro_ecc.html

※micro-ecc は nRF52832、nRF52820、nRF52833 で使用されます

4.4 モジュールの選択

サポートされているモジュールには、プロジェクト毎に固有の構成があります。Segger Embedded Studio のドロップダウンリストを使用して、アプリケーションをコンパイルするターゲットを選択することで、アプリケーションに必要な関連する設定がプロジェクトにロードされます。



5.1.2 nRF Command Line Tools での書き込み

下記手順でコマンドを実施して、評価ボードにファームウェアを書き込みます。

1. FLASH 領域の削除

- ・ コマンド : `nrfjprog --family NRF52 --recover`

※FLASH 領域は書き込み前に削除する必要があります

2. SoftDevice/Master Boot Record(MBR)の書き込み

- ・ コマンド : `nrfjprog --family NRF52 --program mbr_nrf52_2.4.1_mbr.hex`

3. Bootloader ファームウェアの書き込み

- ・ コマンド : `nrfjprog --family NRF52 --program XXXXXXXX.hex`

※書き込む HEX ファイルを指定してください。

4. 評価ボードのリセット SW を押下します。

[実行画面]

```

C:\WINDOWS\system32\cmd.exe
C:\work\BT\Nordic\Firmware_update\FU\sample\EYSKBNZWB>nrfjprog --family NRF52 --recover
Erasing user available code and UICR flash areas.
Applying system reset.

C:\work\BT\Nordic\Firmware_update\FU\sample\EYSKBNZWB>nrfjprog --family NRF52 --program ..\mbr_nrf52_2.4.1_mbr.hex
Parsing hex file.
Reading flash area to program to guarantee it is erased.
Checking that the area to write is not protected.
Programming device.

C:\work\BT\Nordic\Firmware_update\FU\sample\EYSKBNZWB>nrfjprog --family NRF52 --program secure_bootloader_uart_mbr.hex
Parsing hex file.
Reading flash area to program to guarantee it is erased.
Checking that the area to write is not protected.
Programming device.

C:\work\BT\Nordic\Firmware_update\FU\sample\EYSKBNZWB>nrfjprog --family NRF52 --reset
Applying system reset.
Run.

C:\work\BT\Nordic\Firmware_update\FU\sample\EYSKBNZWB>pause
続行するには何かキーを押してください . . .
  
```

5.1.3 nRF Connect Programmer での書き込み

Nordic の下記サイトを参考に書き込みを実施します。

URL :

https://infocenter.nordicsemi.com/topic/ug_nc_programmer/UG/nrf_connect_programmer/ncp_application_overview.html

1. 左上の「Select Device」から接続した j-link Lite を選択します。
2. 右側のメニューDevice から「Erase all」を実施します
3. 右上のメニューFile の「Add HEX file」をクリックし、書き込む mbr の HEX ファイルを選択します。
4. 右側のメニューDevice から「Write」を実行し書き込みを行います
5. 再度右上のメニューFile の「Add HEX file」をクリックし、書き込む Bootloader の HEX ファイルを選択します。
6. 再度右側のメニューDevice から「Write」を実行し書き込みを行います
7. 評価ボードのリセット SW を押下します。

5.1.4 Segger Embedded Studio での書き込み

SES(Segger Embedded Studio)を使って書き込みを行う場合、下記手順を実施して書き込みを行います。

DFU Sample Application では *mbr_nrf52_2.4.1_mbr.hex* を書き込むように設定しているため、下記手順を実施すると一緒に書き込みが行われます。Softdevice 等に変更する場合は、下記”Edit Options...”で表示されるメニューから、Debug→Loader を選択し「Additional Load File[0]」のファイルを変更してください。

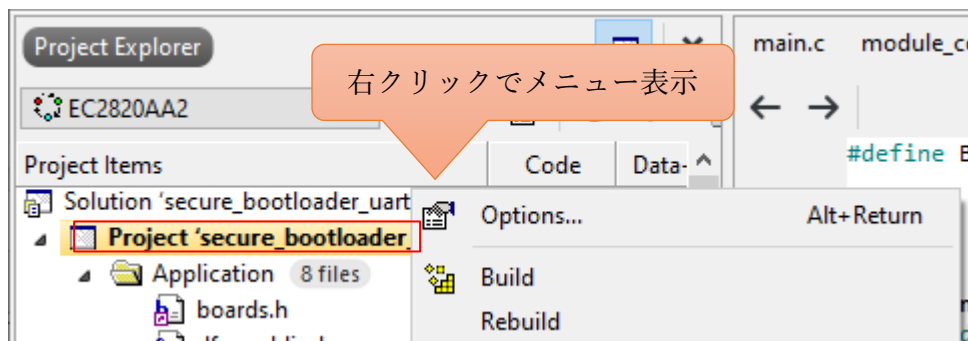
[書き込み手順]

[FLASH 領域の erase]

1. Menu から「Target」をクリックし、「Connect J-Link」を実施します
2. Menu から「Target」をクリックし、「Erase All」を実施します

[Bootloader サンプルの書き込み]

1. Menu から「Target」をクリックし、「Connect J-Link」を実施します
2. Menu から「Target」をクリックし、「Download secure_bootloader_uart_mbr」を実施します



本文中に記載されている会社名、製品名等は、各社の登録商標または商標です。本文中では TM、(R)マーク等は明記していません。

当社のソースコードには一部 Nordic のソースコードが使用されており、その部分の著作権は Nordic に帰属します。