# Add Nordic Semiconductor DFU to SDK example

### **Bluetooth low energy**

Application note







### Abstract

A device firmware update bootloader (DFU) provides a means of updating application firmware over the air (OTA) in the field. This application note describes the procedure to add a buttonless DFU bootloader to an existing application and perform an OTA update.



UBX-19050198 - R03 C1-Public

www.u-blox.com



## **Document information**

Title	Add Nordic Semiconductor DFU to SDK example		
Subtitle	Bluetooth low energy		
Document type	Application note		
Document number	UBX-19050198		
Revision and date	R03	8-Jan-2021	
<b>Disclosure restriction</b>	C1-Public		

#### This document applies to the following products:

Product name	EVK
ANNA-B112	EVK-ANNA-B112
BMD-300	BMD-300-EVAL
BMD-301	BMD-301-EVAL
BMD-330	BMD-330-EVAL
BMD-340	BMD-340-EVAL
BMD-341	BMD-341-EVAL
BMD-345	BMD-345-EVAL
BMD-350	BMD-350-EVAL
BMD-360	BMD-360-EVAL
BMD-380	BMD-380-EVAL
NINA-B111	EVK-NINA-B111
NINA-B112	EVK-NINA-B112
NINA-B301	EVK-NINA-B301
NINA-B302	EVK-NINA-B302
NINA-B306	EVK-NINA-B306
NINA-B400	EVK-NINA-B400
NINA-B406	EVK-NINA-B406

u-blox or third parties may hold intellectual property rights in the products, names, logos and designs included in this document. Copying, reproduction, modification or disclosure to third parties of this document or any part thereof is only permitted with the express written permission of u-blox.

The information contained herein is provided "as is" and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.

Copyright © u-blox AG.



## Contents

Document information	2
Contents	3
1 Software preparation	4
1.1 Nordic Semiconductor nRF5 SDK	4
1.2 SEGGER Embedded Studio	4
1.3 Nordic utilities	4
1.3.1 nRF Connect for Desktop	4
1.3.2 nRF Command Line Tools	4
1.3.3 nRF Util	4
1.4 Encryption libraries	5
1.4.1 Git	5
1.4.2 GCC for ARM	5
1.4.3 GNU make	5
1.4.4 Fetch and compile the encryption libraries	6
2 Hardware preparation	7
2.1 Bluetooth address	7
3 Application firmware	8
4 Button DFU	12
4.1 DFU bootloader	.12
4.2 Key generation	.12
4.3 Firmware update package generation	.13
5 Buttonless DFU	19
5.1 Preprocessor definitions	.19
5.2 sdk_config.h	.22
5.3 Libraries	.22
5.4 main.c	.22
5.5 Hex file generation	.24
6 Test DFU	27
6.1 Program and test application	.27
6.2 Prepare and test update	.28
Appendix	30
A Glossary	30
Related documents	31
Revision history	31
Contact	32



## **1 Software preparation**

### 1.1 Nordic Semiconductor nRF5 SDK

A fresh installation of the nRF5 SDK is assumed. The SDK may be downloaded from reference [6]. For this application note, the installation is assumed to be located in:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099

### **1.2 SEGGER Embedded Studio**

Nordic Semiconductor have partnered with SEGGER to provide a no-cost, no-size-limit, commercial license for SEGGER Embedded Studio (SES) for use with Nordic ICs, including the nRF5x series on which many u-blox modules are based. Install SES from SEGGER and a license from Nordic:

SES: https://www.segger.com/downloads/embedded-studio/

License: https://license.segger.com/Nordic.cgi

### 1.3 Nordic utilities

Several utilities from Nordic Semiconductor are used throughout the process. Install these utilities:

### 1.3.1 nRF Connect for Desktop

nRF Connect for Desktop provides several tools for working with the Nordic Semiconductor-based u-blox modules. nRF Connect for Desktop may be downloaded from reference [7].

### 1.3.2 nRF Command Line Tools

nRF Command Line Tools provides nrfjprog and mergehex, used for development, programming, and debugging. nRF Command Line Tools may be downloaded from reference [8].

### 1.3.3 nRF Util

nRF Util is used for generating the DFU update packages, bootloader settings generation, key generation, and performing updates. nRF Util may be downloaded from the Nordic Semiconductor GitHub repository [9].

nRF Util depends on Python v3.7 or later. Python may be downloaded from reference [10].



It is suggested to install the python component wheel before installing nrfutil:

pip install wheel



### 1.4 Encryption libraries

Encryption libraries are required to create the signed update files use with DFU. Follow the steps outlined in the Nordic instructions to add the encryption libraries to the SDK. For installation instructions, see reference [11].

The following installations are required to install the encryption libraries.

### 1.4.1 Git

3

Git is a distributed version control system used by many development projects. macOS supports git directly through installing X-Code. On Linux, install git with the package manager for the operating system. On Windows, install it from reference [12].

Any other form of git may be used, such as through Linux installed on Windows through WSL or a virtual machine.

The SDK uses git to fetch the source code for micro-ecc.

### 1.4.2 GCC for ARM

The GCC for ARM compiler is used to compile the micro-ecc libraries. GCC for ARM may be downloaded from reference [13].

The nRF5 SDK v17 uses a newer version of the ARM GCC compiler than the instructions at the InfoCenter link indicates. Download this version:

#### GNU Arm Embedded Toolchain: 7-2018-q2-update June 27, 2018

T When executing the installer, be sure to select the option to "Add path to environment variable".

### 1.4.3 GNU make

The make utility is used to manage the build process. There are several ways to install it. macOS and Linux include make as part of the usual development tools. For Windows, the installer is found at reference [14].

On Windows, update the PATH environment variable to include the binary folder for make.exe:

C:\Program Files (x86)\GnuWin32\bin



### 1.4.4 Fetch and compile the encryption libraries

Once the GCC compiler, git, and make are in place, a shortcut provided within the SDK is used to do compile and install the encryption libraries. Open a Command window and navigate to the directory

```
C:\u-blox\nRF5_SDK_17.0.0_9d13099\external\micro-ecc
```

A batch file, and a script for macOS and Linux, called <code>build\_all</code> is used to compile and install microecc.

```
C:\u-blox\nRF5 SDK 17.0.0 9d13099\external\micro-ecc>build all.bat
"micro-ecc not found! Let's pull it from HEAD."
Cloning into 'micro-ecc'...
remote: Enumerating objects: 1086, done.
remote: Total 1086 (delta 0), reused 0 (delta 0), pack-reused 1086
00 KiB/s
Receiving objects: 100% (1086/1086), 647.94 KiB | 921.00 KiB/s, done.
Resolving deltas: 100% (637/637), done.
make: Entering directory `C:\u-blox\nRF5 SDK 17.0.0 9d13099/external/micro-
ecc/nrf51_armgcc/armgcc'
mkdir build
cd build && mkdir micro ecc lib
Compiling file: uECC.c
. . .
. . .
ecc/nrf52nf_iar/armgcc'
make: Entering directory `C:\u-blox\nRF5 SDK 17.0.0 9d13099/external/micro-
ecc/nrf52nf keil/armgcc'
mkdir build
cd build && mkdir micro ecc lib
Compiling file: uECC.c
Creating library: ../../nrf52nf_keil/armgcc/micro ecc lib nrf52.lib
C:/Program Files (x86)/GNU Tools ARM Embedded/7 2018-q2-update/bin/arm-none-eabi-ar:
creating ../../nrf52nf_keil/armgcc/micro_ecc_lib_nrf52.lib
Done
make: Leaving directory `C:\u-blox\nRF5 SDK 17.0.0 9d13099/external/micro-
ecc/nrf52nf keil/armgcc'
```

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\external\micro-ecc>

At this point, the SDK has the necessary encryption libraries not only for DFU use, but also for establishing secure connections through pairing and bonding.

The preparation to this point only needs done once after the SDK zip file is extracted.



## 2 Hardware preparation

For this application note, the BMD-340-EVAL board and a nRF52840 USB dongle are used to demonstrate the process.

Any of the nRF52840 or nRF52832 boards may be used: EVK-NINA-B3, nRF52840 Dongle, nRF52840 DK, EVK-ANNA-B1, EVK-NINA-B1, BMD-300-EVAL, BMD-301-EVAL, BMD-350-EVAL, or Nordic Semiconductor nRF52 DK.

If an alternative board is being used, adjust the directories for the project:

nRF52840 / BMD-340-EVAL, BMD-341-EVAL, BMD-345-EVAL, BMD-380-EVAL, EVK-NINA-B3

- S140 SoftDevice
- PCA10056 hardware directory
- custom\_board.h for NINA-B3, BMD-345
- When working with the BMD-345, be sure to properly configure the PA/LNA in both the bootloader and application. This information is found in the BMD-345 data sheet [4].

nRF52832 / BMD-300-EVAL, BMD-301-EVAL, BMD-350-EVAL, EVK-ANNA-B1, EVK-NINA-B1

- S132 SoftDevice
- PCA10040 hardware directory
- custom\_board.h for ANNA-B1, NINA-B1

nRF52810/BMD-330-EVAL

- S112 SoftDevice
- PCA10040e hardware directory

nRF52811/BMD-360-EVAL

- S113 SoftDevice
- PCA10056e hardware directory

### 2.1 Bluetooth address

u-blox modules are programmed with a unique, public Bluetooth address (MAC address).

Some of the activities outlined in this application note require fully erasing the module, including the Bluetooth address.

To save this address for later reference, use nrfjprog to read the User Information Configuration Register (UICR) area. This example uses the file name evk uicr.hex to store the UICR contents:

nrfjprog --readuicr evk\_uicr.hex

Save the hex file in a convenient location.

The UICR may be restored with:

nrfjprog --program evk uicr.hex

The example does not use the public Bluetooth address. Instead, it uses a static random Bluetooth address that is found in the Factory Information Configuration Registers (FICR). Application note [5] describes how to add code to read and assign the public Bluetooth address held in the UICR.

The BMD-330-EVAL and BMD-360-EVAL have limited DFU functionality. For example, updating the SoftDevice (Bluetooth stack) is not supported due to the flash and RAM memory sizes.



## **3 Application firmware**

Navigate to the example application called ble\_app\_uart:

C:\u-blox\nRF5 SDK 17.0.0 9d13099\examples\ble peripheral\ble app uart>

To maintain the integrity of the original example code, make a copy of the project and all its subdirectories. Only the content of the working directory will be modified:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\examples\ble\_peripheral\ble\_app\_uart\_working>

The BMD-340-EVAL board is functionally equivalent to the nRF52840 DK. Throughout the SDK, this board is referred to by its board number, "pca10056".

Navigate to the SES folder and open the project:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\examples\ble\_peripheral\ble\_app\_uart\_working\pca10056\s1 40\ses\ble\_app\_uart\_pca10056\_s140.emProject

Compile and load the application to the BMD-340-EVAL to confirm operation before any changes are mode. Details for testing the functionality are located at reference [15].

Open the file in nRF Connect to have a look at the memory layout of the hex file. The hex file is located in the Output directory for the example:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\examples\ble\_peripheral\ble\_app\_uart\_working\pca10056\s1 40\ses\Output\Release\Exe\ble\_app\_uart\_pca10056\_s140.hex

The Programmer app of nRF Connect is used to view the memory layout.

Ø	nRF Connect v3	-		$\times$	
	APPS	SETTINGS			
	≇ Filter	Search			
	8	Bluetooth Low Energy General tool for development and testing with Bluetooth Low Energy official, v2.4.0	Open	•	
	<b>IJ</b> .	Direct Test Mode RF PHY testing of Bluetooth Low Energy devices official, v1.1.4	Open	•	
I.	-Ø	Power Profiler Tool to measure current for nRF5x applications official, v2.3.1	Open	•	
		Programmer Tool for flash programming of nRF SoCs official, v1.4.4	Open	•	
	.11	RSSI Viewer Live visualization of RSSI per frequency for nRF52832 official, v1.2.0	Open	•	
	ම	Toolchain Manager Install and manage tools to develop with the nRF Connect SDK (NCS) official, v0.7.1	Open	•	

Figure 1: nRF Connect main window (programmer)



Add the hex file, or just drag it into the "File Memory Layout" section:

nRF Connect v3.4.2	- Programmer				- 🗆
≡ Selec	t device 👻 🔍				
_				File	
Device mer	mory layout	File memory layout 🗁			
				🖨 Add HEX	K file
				C Reload f	îles
				😑 Clear file	es
	11			Device	
		F	Region name		
Connect a	device to display memory contents	F	file name ble_app_uart_p	ca10056_s140.hex	write
		A O	Address range 0x00027000 —	0x0002E502	file
		s 2	Size 19955 bytes		
a			1 4	/ Write	
-22.50 277	Application data folder: C:\Llocro\brea	\AppData\Reaming\prfcoppact\po.prfcoppact progra	mmor	C Read	
22:50.277	Using prfiprog library 10.9.0 pc-prfipro	on-is 1.7.3			
:23:04.308	Parsing HEX file: C:\u-blox\nRF5_SDK	17.0.0 9d13099\examples\ble_peripheral\ble_app_u	Jart\pca1	Auto rea	d memory
:23:04.331	File was last modified at 8/26/2020, 2	:22:43 PM		Enable N	//CUboot
			•		

#### Figure 2: nRF Connect showing application only

Only the application itself is in the file. Notice the gap below. This is where the SoftDevice gets loaded. Add or drag the SoftDevice hex file to nRF Connect. The SoftDevice is located here:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\components\softdevice\s140\hex\s140\_nrf52\_7.0.1\_softdevice.hex



#### Now both are shown, along with the MBR:



Figure 3: nRF Connect showing application with SoftDevice

While nRF Connect can be used to program the BMD-340, switch back to SES and load the SoftDevice and application from there:

File Edit View Search Na	avigate P	roject Build	d Debug	Targ	get Tools Window Help	
D 🕶 🛩 🕶 🖪 🕼 🗔	× 🔳	$\mathbf{f} \cdot \mathbf{v} \times \mathbf{v}$	$\Sigma + \alpha$	1	Connect J-Link	Ctrl+T, C
			main c	×.	Disconnect	Ctrl+T, D
Project Explorer			main.c	3	Reconnect	Ctrl+T, E
🕄 Release 🔹 🗖 🧧	🗋 😭 🖸	+ <> *	$\leftarrow \rightarrow$	<b>i</b> j≣	Attach Debugger	Ctrl+T, H
Project Items	Code	Data	*		Poset	Ctrl+T_S
Solution 'ble_app_uart_pca10056_s			*	<b></b>		curri, 5
Project 'ble_app_uart_pca1005	25.5K	191.6K	*	t≣.	Download ble_app_uart_pca10056_s1	40 Ctrl+T, L y
Application 2 files	[1.3K]	[2.0K]	*	7	Verify ble_app_uart_pca10056_s140	Ctrl+T, V CO
▷ 着 main.c	1.3K	2.0K	*		France All	Challe T. IV
👌 sdk_config.h			*		Erase All	ctn+1, K tai

Figure 4: SES download application to target



Output	🖸 🏘 🗙
Show: Target 🔹 🍢 Tasks 🔹	
Connecting 'J-Link' using 'USB' > Completed	
Preparing target for download Completed	
Downloading 's140_nrf52_6.1.1_softdevice.hex' to J-Link Download successful	150.2 KB in 5.4: 27.6 KB/s
Downloading 'ble_app_uart_pca10056_s140.elf' to J-Link Download successful	28.6 KB in 0.9s 30.9 KB/s

#### Figure 5: SES download progress window

LED1 should start flashing, indicating that the BMD-340 is advertising for a connection. Go ahead and test the application according to the instructions at the InfoCenter link above.





## 4 Button DFU

### 4.1 DFU bootloader

After confirming operation, the bootloader can now be added. Nordic Semiconductor provides several DFU examples. The Secure Bootloader example is used here. See the Nordic Semiconductor InfoCenter for details about this secure bootloader [16].

On the local drive, navigate to the <code>secure\_bootloader</code> DFU directory of the examples and make a copy of the directory. Copy

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\examples\dfu\secure\_bootloader\pca10056\_s140\_ble

to

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\examples\dfu\secure\_bootloader\pca10056\_s140\_ble\_working

#### Open the SES project for the pca10056 Bluetooth low energy bootloader:

```
C:\u-blox\nRF5_SDK_17.0.0_9d13099\examples\dfu\secure_bootloader\pca10056_s140_ble_working \ses\secure_bootloader_ble_s140_pca10056.emProject
```

#### Compiling the bootloader "as is" will generate an error:

```
#error "Debug public key not valid for production. Please see
https://github.com/NordicSemiconductor/pc-nrfutil/blob/master/README.md to generate it"
```

Nordic Semiconductor supplies an example keys along with test files. These are only made available so the process of working with updating a file on a device can be tested; however, they cannot be used for production.

### 4.2 Key generation

nrfutil is used to generate a custom set of keys. Navigate to the directory where the default public key is located and change the name of the original key file, dfu public key.c:

move dfu\_public\_key.c dfu\_public\_key.c.NORDIC

Generate new keys. First, start with the private key:

nrfutil keys generate dfu\_private\_key.pem
Generated private key and stored it in: dfu\_private\_key.pem

Then create the public key in C language:

These are example values. The actual output will vary.

3



The DFU project in SES can now be compiled without errors.

Fully erase the BMD-340-EVAL before proceeding:

```
nrfjprog -f nrf52 --recover
Recovering device. This operation might take 30s.
Erasing user code and UICR flash areas.
```

From the SES project, load the DFU and SoftDevice using the same procedure as the UART project.

Both LED1 and LED2 will light up solid. This indicates the BMD-340 is in bootloader mode. The module will be advertising with the name "DfuTarg".

### 4.3 Firmware update package generation

The firmware update package consists of a signed zip file containing the application, SoftDevice, bootloader, and bootloader settings.

Navigate to the output directory of the ble\_app\_uart example that was compiled earlier.

```
C:\u-
blox\nRF5_SDK_17.0.0_9d13099\examples\ble_peripheral\ble_app_uart_working\pca10056\s140\se
s\Output\Release\Exe
```

## nrfutil will be used with the package generate function. The help output will show all of the available optoins.

#### nrfutil pkg generate --help

For this example, the following options are used:

Option	Description
application ble_app_uart_pca10056_s140.hex	Compiler output hex file
application-version-string "1.2.3"	Text version string
	The string "1.2.3" is converted to an integer: 10203
hw-version 52	The default number "52" is used for a nRF52xxx device. This can actually be any integer as well, for example to reflect the host board version. If new code cannot be loaded onto an older version board, then this is the value for that check.
sd-req 0xCA	If your application requires a specific version of the SoftDevice, this value is used for that check.
	nrfutil pkg generate -help will list the available SoftDevice versions and their corresponding firmware IDs. More than one may be included here, comma separated.
	The following SoftDevice v7.x.x codes are not listed in the help output, but can be used with nrfutil:
	s112_nrf52_7.0.0 0xC4
	s112_nrf52_7.0.1 0xCD
	s113_nrf52_7.0.0 0xC3
	s113_nrf52_7.0.1 0xCC
	s132_nrf52_7.0.0 0xC2
	s132_nrf52_7.0.1 0xCB
	s140_nrf52_7.0.0 0xC1
	s140_nrf52_7.0.1 0xCA
key-file C:\u-blox\nRF5_SDK_17.0.0_9d13099 \examples\dfu\dfu_private_key.pem	Path and file name of the new private .pem key
app_vl.zip	Output file name

#### Table 1: nrfutil package generation options



Putting it all together:

```
nrfutil pkg generate --application ble_app_uart_pca10056_s140.hex --application-version-
string "1.2.3" --hw-version 52 --sd-req 0xCA --key-file C:\u-
blox\nRF5_SDK_17.0.0_9d13099\examples\dfu\dfu_private_key.pem app_v1.zip
Zip created at app_v1.zip
```

To test this package, connect a second EVK or nRF52840 Dongle to your computer. Open nRF Connect and select the "Bluetooth Low Energy" option:

nRF Connect v3	.4.2	-		$\times$
APPS	SETTINGS			
≇ Filter	Search			
8	Bluetooth Low Energy General tool for development and testing with Bluetooth Low Energy official, v2.4.0	Open	•	
2.	Direct Test Mode RF PHY testing of Bluetooth Low Energy devices official, v1.1.4	Open	•	
Ø	Power Profiler Tool to measure current for nRF5x applications official, v2.3.1	Open	•	
	Programmer Tool for flash programming of nRF SoCs official, v1.4.4	Open	•	
.11	RSSI Viewer Live visualization of RSSI per frequency for nRF52832 official, v1.2.0	Open	•	
ම	Toolchain Manager Install and manage tools to develop with the nRF Connect SDK (NCS) official, v0.7.1	Open	•	

Figure 6: nRF Connect main window (Bluetooth Low Energy)



In this example the nRF52840 dongle is the second device. Select this device.

If a notice to program the connectivity firmware is displayed, select yes.

	.2 - Bluetooth Low Energy					- 🗆
Sele	ct device 👻 🔍	📥 Connection Map	亘 Server Setup			
EF7	905FEC31E Serial port: COM9				Discovered de	evices
•	USB: Nordic Semiconductor	r nRF52 Connectivity			Start scan	Clear
000	0683592301 PCA10 Serial port: COM16	056			<ul> <li>Options</li> </ul>	
				■ ■ ↓		
g						
<b>g</b> :45:48.040	Device setup cor	πριετεα		<b>A</b>		
<b>g</b> :45:48.646 :45:48.647	Device setup cor Connectivity firm	mpietea nware version: ble-conne	ctivity 4.1.2+Jul-14-2020-05-48-48. SoftDe	evice API versio		
<b>g</b> :45:48.646 :45:48.647 :45:48.652	Device setup cor Connectivity firm Opening adapter	mpieted ware version: ble-connec connected to COM9	ctivity 4.1.2+Jul-14-2020-05-48-48. SoftDe	evice API versio		
<b>g</b> :45:48.646 :45:48.652 :45:49.595	Device setup cor Connectivity firm Opening adapter Successfully ope	mpleted nware version: ble-connel connected to COM9 ened COM9. Baud rate: 1	ctivity 4.1.2+Jul-14-2020-05-48-48. SoftDe 000000. Flow control: none. Parity: none.	evice API versio		
<b>g</b> :45:48.640 :45:48.652 :45:49.595 :45:49.600	Device setup cor Connectivity firm Opening adapter Successfully ope Reset performed	mpleted ware version: ble-conner connected to COM9 ened COM9. Baud rate: 1 d on adapter COM9	ctivity 4.1.2+Jul-14-2020-05-48-48. SoftDe 000000. Flow control: none. Parity: none.	evice API versio		
<b>g</b> :45:48.640 :45:48.647 :45:48.652 :45:49.595 :45:49.600 :45:50.692	Device setup cor Connectivity firm Opening adapter Successfully ope Reset performed Adapter connect	mpietea ware version: ble-conner connected to COM9 ened COM9. Baud rate: 1 don adapter COM9 ted to COM9 opened	ctivity 4.1.2+Jul-14-2020-05-48-48. SoftDe 000000. Flow control: none. Parity: none.	evice API versio		

Figure 7: nRF Connect Bluetooth Low Energy window



Start the scan and look for the Eval board advertising "DfuTarg":

8 nRF Connect v3.4.2 -	Bluetooth Low Energy	- 🗆 ×
≡ EF7905	FEC31E 👻 🍳 📥 Connection Map 🗉 Server Setup	
RRF5x	Adapter 💠	Discovered devices  Stop scan  Clear  Options  Sort by signal strength  Eliters  of a
Generic Access     Generic Attribute	2	DfuTarg     -41 dBm .ull       F1:D7:E9:CF:E4:48     Connect Ø       • Details
Log	8 B 4	-
15:49:01.320	Device setup completed	
15:49:01.320	Connectivity Tirmware version: ble-connectivity 4.1.2+Jul-14-2020-05-48-48. SoftDevice API versio	
15:49:01.321	Upening adapter connected to CUM9 Successfully opened COM0, Reud rate: 1000000, Elew control: pone, Darit::::::::::::::::::::::::::::::::::::	
15:49:02:203	Succession opened COM9. Bauditate. TO00000. Flow Control. none. Parity: none.	
15:49.02.200	Adapter connected to COM9	
15:49:05:371	Scan started	
4		

Figure 8: nRF Connect showing scanned bootloader DfuTarg



8 nRF Connect v3.4.2	Bluetooth Low Energy	- 🗆 X
≡ EF790	FEC31E 👻 🔹 📥 Connection Map 🗉 Server Setup	
<ul> <li>RRF5x</li> <li>EF:79:05:F</li> <li>Generic Access</li> <li>Generic Attribut</li> </ul>	CC3.1E DfuTarg Perphanal () C () () () () () () () () () () () () ()	Discovered devices
<b>Log</b> 15:49:03.371	Adapter connected to CUM9 opened	
15:49:05.357	Scan started	
15:50:05.356	Scanning timed out on adapter COM9	
15:51:38.173	Connecting to device	
15:51:38.221	Connected to device F1:D7:E9:CF:E4:4B	
15:51:38.317	Connection parameters updated for device F1:D7:E9:CF:E4:4B: interval 15ms, timeout 6000ms, lat	
15:51:38.884	Attribute value read, handle: 0x03, value (0x): 44-66-75-54-61-72-67	
4		

Select "Connect". Observe the "Secure DFU" service and update icon:

Figure 9: nRF Connect showing secure DFU icon



Select the icon to start the DFU process. Navigate to the zip file, then select "Start DFU".

8 nRF Connect v3.4.2 -	Bluetooth Low Energy		- 🗆 ×
≡ EF7905F	FEC31F 🗸 🍙	& Connection Man	
	Device Firm	ware Upgrade (DFU) for device F1:D7:E9:CF:E4:4B	
	-		d devices
nRF5x     EF:79:05:FE:	Zip file C3:1E	C:\u-blox\nRF5_SDK_17.0.0_9d13099\examples\ble_periph	ose an 🔟 Clear
	Package	application:	nal strength
Conorio Accesso	info	<pre>bin_file: ble_app_uart_pca10056_s140.bin</pre>	arg
Generic Access		<pre>dat_file: ble_app_uart_pca10056_s140.dat</pre>	-39 dBmI
Generic Attribute	8		4B Connect Ø
		► Start D	FU
			-
		Cl	se
Log			
15:51:38.317	Connection parame	eters updated for device FT:D7:E9:CF:E4:48: Interval T5ms, tim	
15:51:38.884	Attribute value read	l, handle: 0x03, value (0x): 44-66-75-54-61-72-67	
15:53:25.773	Disconnected from	device F1:D7:E9:CF:E4:4B, reason: BLE_HCI_CONNECTION_T	
15:54:23.919	Connecting to device	ce	
15:54:23.965	Connected to devic	e F1:D7:E9:CF:E4:4B	
15:54:24.096	Connection parame	eters updated for device F1:D7:E9:CF:E4:4B: interval 15ms, tim	
15:54:24.625	Attribute value read	l, handle: 0x03, value (0x): 44-66-75-54-61-72-67	
4		<b>•</b>	

Figure 10: nRF Connect ready for DFU upgrade

A status bar will show up, first with a progress bar, then "Complete".

At this point, the Eval board will start advertising "Nordic\_UART" – the example from Section 3 – though now it includes the bootloader.

If power to the Eval board is reset or power-cycled while holding Button 4, it will re-enter bootloader mode indicated by both LED1 and LED2 being lit.

Up to this point, the following is completed:

- Confirmed that the application code works as expected
- Generated private and public keys for the bootloader and update files
- Compiled the bootloader with the new public key
- Generated a DFU update zip file with the new private key
- Performed a DFU update and observed the application running
- Return to bootloader mode through a button press.
- Up to this point, the application did not require any modification. The DFU zip file is generated from the existing application hex and generated public key files.





## **5 Buttonless DFU**

If there are no available buttons available to enter bootloader mode through hardware, or to create a more seamless experience for the end user, the process may be started automatically through the "buttonless DFU" feature. This requires modification of the application to add a new service which enables this feature.

A Bluetooth low energy service is a collection of information and behaviors to perform a particular function or feature. In this case, the service will accept a data value (a characteristic) and write it to a register that is persistent across a device reset. The bootloader then reads this register on startup to determine if it should continue with the DFU function, or pass control back to the application. Additional details can be found in this tutorial [17] from Nordic Semiconductor.

In short, this service performs what was originally enabled by pressing button 4 while resetting the Eval board.

The ble\_app\_buttonless\_dfu example will be added to the ble\_app\_uart example.

To save the work completed so far, copy the modified project a new example folder:

 $\label{eq:copy_c:u-blox} copy_C:\u-blox\nRF5_SDK_17.0.0_9d13099\examples\ble_peripheral\ble_app_uart_working_C:\u-blox\nRF5_SDK_17.0.0_9d13099\examples\ble_peripheral\ble_app_uart_working_dfu$ 

Change directory into the new folder:

 $cd \ C: \ u-blox \ nRF5\_SDK\_17.0.0\_9d13099 \ examples \ ble\_peripheral \ ble\_app\_uart\_working\_dfu$ 

Open the project file from here:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\examples\ble\_peripheral\ble\_app\_uart\_working\_dfu \pca10056\s140\ses\ble\_app\_uart\_pca10056\_s140.emProject

### 5.1 Preprocessor definitions

The project options need modified to use DFU.

Right click on the project and select "Options". Once the options window is open.

Be sure to select the "Common" configuration before editing the values:



SEGGER Embedded	SEGGER Embedded Studio for ARM V5.10 - Options			
Project 'ble_app	_uart_pca10056_s140' Options			
↑ ↓ 🕄 Release	Search Options	Show Modified Options Only		
As Release Bui Co Common	figurations embler	gcc		
External Buil File Library	Additional Assembler Options     Additional Assembler Options From File     Generate Assembler Listing File     Run Preprocessor	None No Yes		
Linker Preprocessc Printf/Scanf	Build     Exclude From Build	No		
Runtime Me Section Source Code	Project Type     Project Directory     Output Directory     Intermediate Directory	Executable inherits None modified Output/\$(Configuration)/Exe Output/\$(Configuration)/Obi/\$		
User Build S Debug Debugger GDB Server				
J-Link Loader Simulator 🗸				
		OK Cancel		

Figure 11: SES project pre-processor definitions

Select the Preprocessor topic under "Code" and add the following to "Preprocessor Definitions":

```
BL_SETTINGS_ACCESS_ONLY
NRF_DFU_TRANSPORT_BLE=1
```

F

Add the following to "User Include Directories":

```
../../../../../components/libraries/bootloader
../../../../components/libraries/bootloader/dfu
../../../../components/libraries/bootloader/ble_dfu
```

The last item above may already be in the list. If so, do not add it again.



The final values to adjust are the RAM\_START address and RAM\_SIZE. These are found in the Linker options, within the Section Placement Macros line.

Code     Option     Value       Assembler     Build	Code       Option       Value         Assembler       Build       - I Linker         Build       - Executable File Name       \$(OutDir)/\$(ProjectName)\$(EXE)         Compiler       - Additional Input Files       -         External Build       - Linker       Yes         File       Use Manual Linker Script       No         Library       - Section Placement File       flash_placement.xml modified         Linker       - Section Placement Macros       FLASH_PH_START=0x0;FLASH_PH_SIZE=0x10000;Ru         Preprocessor       - DebugIO Implementation       Default         Printf/Scanf       - Additional Output File Gap Fill Value       None         Source Code       - Entry Point       Yes       - Additional Output File Gap Fill Value         Debug       - Keep Symbol Definitions       - Entry Point       - Eset Handler modified         Debug       - Keep Symbols       - Server       - Strip Debug Information       No         J-Link       - Strip Symbols       No       - Server       - Strip Symbol Definition         J-Link       - No Fund Size Warning       No       No       - Server         J-Link       - No Fund Size Warning       No       - No Wide Char Size Warning       No         Simulator	Code	1 Prime Part International Control of Contro		
Assembler         Build         Code Generation         Compiler         Executable File Name         External Build         File         Linker         Section Placement File         Linker         Preprocessor         Printf/Scanf         Runtime Memory Are         Section         Section all Upt File         Use Manual Linker Script         No         Preprocessor         Printf/Scanf         Runtime Memory Are         Section         Section 2         Section 3         Section 4         Section 4         Section 5         Section 4         Section 6         Section 8         Section 9         Section 9         Section 9         Section 10         Section 10         Section 2         Section 10         Section 2         Section 11         Section 12         Section 12         Section 12         Section 12         Section 12         Section 12         Secti	Assembler         Build       • ILinker         Code Generation       • Executable File Name         Compiler       • Additional Input Files         External Build       • Link Dependent Projects       Yes         File       • Use Manual Linker Script       No         Library       • Section Placement File       flash_placement.xml modified         Linker       • Section Placement Macros       FLASH_PH_START=0x0;FLASH_PH_SIZE=0x10000;Ri         Vinker       • Section Placement Macros       Preprocessor         Preprocessor       • Default Fill Pattern       None         Printf/Scanf       • Additional Output Format       hex modified         Runtime Memory Are       • Additional Output File Gap Fill Value       None         Source Code       • Entry Point       Reset_Handler modified         User Build Step       • Linker Symbol Definitions	code	Option	Value	
Code Generation	Joind       Inker         Code Generation       • Executable File Name       \$(OutDir)/\$(ProjectName)\$(EXE)         Compiler       • Additional Input Files       •         External Build       • Link Dependent Projects       Yes         File       • Use Manual Linker Script       No         Linker       • Section Placement File       flash_placement.xml modified         Linker       • Section Placement Macros       FLASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;Ru         Preprocessor       • Default Fill Pattern       None         Printf/Scanf       • Additional Output Format       hex modified         Runtime Memory Are       • Additional Output File Gap Fill Value       None         Source Code       • Entry Point       Reset_Handler modified         User Build Step       • Linker Symbol Definitions       Debug         Debug       • Strip Debug Information       No         GDB Server       • Strip Symbols       No         J-Link       • No Wide Char Size Warning       No         Joader       • No Wide Char Size Warning       No         Simulator       • Suppress Warning on Mismatch       No         Target Control       • Treat Linker Warning as Errors       No modified	Assembler			
Code GenerationExecutable File NameS(OutDir)/S(ProjectName)S(EXE)Compiler <ul><li>Additional Input Files</li><li>External Build</li><li>Link Dependent Projects</li><li>Yes</li><li>Section Placement File</li><li>Default Fill Pattern</li><li>Debug/Orege</li><li>Source Code</li><li>User Build Step</li><li>Debug</li><li>Betury Debug Information</li><li>Strip Symbols</li><li>Strip Symbols</li><li>No</li><li>Additional Definition</li><li>No</li><li>Strip Symbols</li><li>No</li><li>No</li><li>Strip Symbols</li><li>No</li><li>Strip Symbols</li><li>No</li><li>Simulator</li><li>Target Control</li><li>Target Trace</li></ul>	Code Generation• Executable File Name\$ (OutDir)/\$(ProjectName)\$(EXE)Compiler• Additional Input FilesExternal Build• Link Dependent ProjectsYesFile• Use Manual Linker ScriptNoLibrary• Section Placement Fileflash_placement.xml modifiedLinker• Section Placement MacrosFLASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;RuPreprocessor• Default Fill PatternNonePrintf/Scanf• Additional Output File Gap Fill ValueNoneRuntime Memory Are• Generate Map FileYesSource Code• Entry PointReset_Handler modifiedUser Build Step• Linker Symbol DefinitionsNoDebugger• Strip Debug InformationNoGDB Server• Strip Debug InformationNoJ-Link• Allow Multiple Symbol DefinitionNoLoader• No Wide Char Size WarningNoSimulator• No Wide Char Size WarningNoTarget Control• Treat Linker Warnings as ErrorsNo modified	Code Constation	4 Linker		
CompilerAdditional Input FilesExternal Build• Link Dependent ProjectsYesFile• Use Manual Linker ScriptNoLibrary• Section Placement Fileflash_placement.xml modifiedLinker• Section Placement MacrosFLASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;RA····Preprocessor• Default Fill PatternNonePrintf/Scanf• DebuglO ImplementationDefaultRuntime Memory Are• Additional Output Formathex modifiedSection• Generate Map FileYesUser Build Step• Linker Symbol DefinitionsDebugger• Strip DebugNoGDB Server• Strip Debug InformationNoJ-Link• No Wide Char Size WarningNoLoader• No Wide Char Size WarningNoSimulator• Suppress Warning on MismatchNoTarget Control• Treat Linker Warnings as ErrorsNo modifiedTarget TraceSection Placement MacrosNo modified	Compiler• Additional Input FilesExternal Build• Link Dependent ProjectsYesFile• Uise Manual Linker ScriptNoLibrary• Section Placement Fileflash_placement.xml modifiedLinker• Section Placement MacrosFLASH_PH_START=0x0;FLASH_PH_SIZE=0x10000;RiPreprocessor• Default Fill PatternNonePrintf/Scanf• Default Fill PatternNoneRuntime Memory Are• Additional Output File Gap Fill ValueNoneSection• Generate Map FileYesSource Code• Entry PointReset_Handler modifiedUser Build Step• Linker Symbol DefinitionsNoDebugger• Strip Debug InformationNoGDB Server• Strip Debug InformationNoJ-Link• No Enum Size WarningNoLoader• No Wide Char Size WarningNoSimulator• No Wide Char Size WarningNoTarget Control• Treat Linker Warnings as ErrorsNo modified	Code Generation	Executable File Name	\$(OutDir)/\$(ProjectName)\$(EXE)	
External Build     • Link Dependent Projects     Yes       File     • Use Manual Linker Script     No       Library     • Section Placement File     flash_placement.xml modified       Linker     • Section Placement Macros     FLASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;RA***       Preprocessor     • Default Fill Pattern     None       Printf/Scanf     • Additional Output Format     hex modified       Runtime Memory Are     • Additional Output File Gap Fill Value     None       Source Code     • Entry Point     Reset_Handler (modified)       User Build Step     • Linker Symbol Definitions        Debugger     • Strip Debug Information     No       GDB Server     • Strip Symbol Definition     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	External Build• Link Dependent ProjectsYesFile• Use Manual Linker ScriptNoLibrary• Section Placement Fileflash_placement.xml modifiedLinker• Section Placement Fileflash_placement.xml modifiedPreprocessor• Default Fill PatternNonePrintf/Scanf• Additional Output Formathex modifiedRuntime Memory Are• Additional Output File Gap Fill ValueNoneSource Code• Entry PointYesUser Build Step• Linker Symbol DefinitionsPebuggerDebugger• Strip Debug InformationNoGDB Server• Strip Debug InformationNoJ-Link• No Wide Char Size WarningNoLoader• No Wide Char Size WarningNoSimulator• No Wide Char Size WarningNoTarget Control• Treat Linker Warnings as ErrorsNo modified	Compiler	Additional Input Files		
File     Use Manual Linker Script     No       Library        • Section Placement File       • Section Placement File       • Section Placement Macros       • LiASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;RA····       • Default Fill Pattern       • Default Fill Pattern       • Default Fill Pattern       • Default Output Format       • Additional Output Format       • Additional Output File Gap Fill Value       • Generate Map File       • Entry Point       • Strip Debug Information       • Strip Debug Information       • No       • Strip Debug Information       • No       • No Enum Size Warning       • No Enum Size Warning       • No       • No Wide Char Size Warning       • No       • Suppress Warning on Mismatch       • Traget Trace	FileUse Manual Linker ScriptNoLibrary• Section Placement Fileflash_placement.xml modifiedLinker• Section Placement MacrosFLASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;RuPreprocessor• Default fill PatternNonePrintf/Scanf• Additional Output Formathex modifiedRuntime Memory Are• Additional Output File Gap Fill ValueNoneSection• Generate Map FileYesSource Code• Linker Symbol DefinitionsPetsut File Gap FillUser Build Step• Linker Symbol DefinitionsNoDebugger• Strip Debug InformationNoGDB Server• Strip Debug InformationNoJ-Link• No Kuld Char Size WarningNoLoader• No Wide Char Size WarningNoSimulator• Suppress Warning on MismatchNoTarget Control• Treat Linker Warnings as ErrorsNo modified	External Build	Link Dependent Projects	Yes	
Library       * Section Placement File       Trash_placement.xml moduled         Linker       * Section Placement Macros       FLASH_PH_START=0x0;FLASH_PH_SIZE=0x10000;R4····         Preprocessor       • Default fill Pattern       None         Printf/Scanf       • Additional Output Format       hex modified         Runtime Memory Are       • Additional Output File Gap Fill Value       None         Source Code       • Entry Point       Reset_Handler modified         User Build Step       • Linker Symbol Definitions       •         Debugger       • Strip Debug Information       No         GDB Server       • Allow Multiple Symbol Definition       No         J-Link       • No Enum Size Warning       No         Loader       • No Wide Char Size Warning       No         Simulator       • Suppress Warning on Mismatch       No         Target Control       • Treat Linker Warnings as Errors       No modified	Library       * Section Placement Hie       Tlash_placement.xml moduled         Linker       * Section Placement Macros       FLASH_PH_START=0x0;FLASH_PH_SIZE=0x100000;Ri         Preprocessor       • Default Fill Pattern       None         Printf/Scanf       • Additional Output Format       hex modified         Runtime Memory Are       • Additional Output File Gap Fill Value       None         Section       • Generate Map File       Yes         Source Code       • Entry Point       Reset_Handler modified         User Build Step       • Linker Symbol Definitions       Eester Handler modified         Debugger       • Strip Debug Information       No         GDB Server       • Strip Symbols       No         J-Link       No Wide Char Size Warning       No         Simulator       • No Wide Char Size Warning       No         Target Control       • Treat Linker Warnings as Errors       No modified	File	Use Manual Linker Script	No Bask alexandered and find	
Linker     * Section Pracement Macros     PLASH_PH_START=UX07LASH_P	Linker     • Section Pracement Macros     FLASH_PH_START=UX07LASH_PH_SIZE=UX100000/K       Preprocessor     • Default Fill Pattern     None       Printf/Scanf     • DebugIO Implementation     Default       Runtime Memory Are     • Additional Output Format     hex_modified       Section     • Additional Output File Gap Fill Value     None       Section     • Generate Map File     Yes       Source Code     • Entry Point     Reset_Handler modified       User Build Step     • Linker Symbol Definitions     Inter Symbol S       Debugger     • Strip Debug Information     No       GDB Server     • Strip Symbols     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	Library	Section Placement File	flash_placement.xml modified	
Preprocessor     Preprocessor     None       Printf/Scanf     DebuglO Implementation     Default       Runtime Memory Are     Additional Output File Gap Fill Value     None       Section     • Additional Output File Gap Fill Value     None       Source Code     • Entry Point     Reset_Handler modified       User Build Step     • Linker Symbol Definitions     -       Debugger     • Strip Debug Information     No       GDB Server     • Strip Symbols     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	Preprocessor     Default Fill Pattern     None       Printf/Scanf <ul> <li>Additional Output Format</li> <li>Additional Output File Gap Fill Value</li> <li>Source Code</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Keep Symbol Definitions</li> <li>Entry Point</li> <li>Keep Symbol Definitions</li> <li>Strip Debug Information</li> <li>Strip Debug Information</li> <li>Strip Symbols</li> <li>Strip Symbols</li> <li>No</li> <li>Allow Multiple Symbol Definition</li> <li>No</li> <li>No Enum Size Warning</li> <li>No Mode</li> <li>Suppress Warning on Mismatch</li> <li>Target Control</li> <li>Treat Linker Warnings as Errors</li> <li>No</li> <li>Mode</li> <li>Mone</li> <li>Mon</li></ul>	Linker	Section Placement Macros     Default fill Pattern	FLASH_PH_START=0X0;FLASH_PH_SIZE=0X100000;R	- · ·
Printf/Scanf     Debugio Implementation     Default       Runtime Memory Are     Additional Output Format     hex modified       Section     Generate Map File     Yes       Source Code     Entry Point     Reset_Handler modified       User Build Step     Linker Symbol Definitions     Pebugger       Debugger     Strip Debug Information     No       GDB Server     Strip Symbols     No       J-Link     No Enum Size Warning     No       Loader     No Wide Char Size Warning     No       Simulator     Suppress Warning on Mismatch     No       Target Control     Treat Linker Warnings as Errors     No modified	Printf/Scanf     • Debugio Implementation     Debugio       Runtime Memory Are     • Additional Output File Gap Fill Value     None       Section     • Additional Output File Gap Fill Value     None       Source Code     • Entry Point     Reset_Handler modified       User Build Step     • Linker Symbol Definitions     Pebug       Debug     • Keep Symbols     No       Debugger     • Strip Debug Information     No       GDB Server     • Strip Symbols     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	Preprocessor	Default Fill Pattern     SchudlQ Implementation	Default	
Runtime Memory Are Section     Additional Output File Gap Fill Value     None       Source Code     • Entry Point     Reset_Handler modified       User Build Step     • Linker Symbol Definitions     •       Debug     • Keep Symbols     •       Debugger     • Strip Debug Information     No       J-Link     • Allow Multiple Symbol Definition     No       J-Link     • No Enum Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	Runtime Memory Are     Additional Output Format     Inter modified       Section <ul> <li>Additional Output File Gap Fill Value</li> <li>None</li> <li>Generate Map File</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Linker Symbol Definitions</li> <li>Etry Point</li> <li>Strip Debug Information</li> <li>Strip Debug Information</li> <li>Strip Symbols</li> <li>Strip Symbol Definition</li> <li>Allow Multiple Symbol Definition</li> <li>No</li> <li>No Endote Control</li> <li>Target Control</li> <li>Treat Linker Warning as Errors</li> <li>No modified</li> </ul> <li>No modified</li>	Printf/Scanf	Additional Output Format	bey modified	
Section     • Additional Output Pile Gap Pile     None       Source Code     • Entry Point     Reset_Handler modified       User Build Step     • Linker Symbol Definitions     •       Debug     • Keep Symbols     •       Debugger     • Strip Debug Information     No       GDB Server     • Strip Symbols Definitions     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	Section     • Additional Output Pile Gap Pill Value     None       Source Code     • Entry Point     Reset_Handler modified       User Build Step     • Linker Symbol Definitions     Pebug       Debug     • Keep Symbols     No       Debugger     • Strip Debug Information     No       GDB Server     • Strip Symbol Definitions     No       J-Link     • Allow Multiple Symbol Definition     No       Loader     • No Finum Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No	Runtime Memory Are	Additional Output Format	None	
Source Code     Contract Map The     Reset_Handler modified       User Build Step <ul> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Reset_Handler modified&lt;</li></ul>	Source Code     Entry Point     Reset_Handler modified       User Build Step <ul> <li>Entry Point</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li>Entry Point</li> <li>Reset_Handler modified</li> <li< td=""><td>Section</td><td>* Generate Man File</td><td>Vac</td><td></td></li<></ul>	Section	* Generate Man File	Vac	
User Build Step <ul> <li>Linker Symbol Definitions</li> <li>Ether Symbol Definitions</li> <li>Strip Debug Information</li> <li>Strip Debug Information</li> <li>Strip Symbols</li> <li>Strip Symbol Definitions</li> <li>Strip Symbol Definition</li> <li>No</li> <li>Allow Multiple Symbol Definition</li> <li>No Enum Size Warning</li> <li>No Wide Char Size Warning on Mismatch</li> <li>No Woide Char Size Warning as Errors</li> <li>Target Script</li> <li>Target Trace</li> <li>Section Placement Macros</li> <li>Control Target Trace</li> <li>Control T</li></ul>	User Build Step     Linker Symbol Definitions       Debug     • Keep Symbols       Debugger     • Strip Debug Information       GDB Server     • Strip Symbol Definitions       J-Link     • Allow Multiple Symbol Definition       Loader     • No Wide Char Size Warning       Simulator     • Suppress Warning on Mismatch       Target Control     • Treat Linker Warnings as Errors	Source Code	* Entry Point	Reset Handler modified	
Debug     • Keep Symbols       Debugger     • Strip Debug Information       GDB Server     • Strip Symbols       J-Link     • Allow Multiple Symbol Definition       Loader     • No Enum Size Warning       Simulator     • Suppress Warning on Mismatch       Target Control     • Treat Linker Warnings as Errors       Target Script     • Section Placement Macros	Debug     • Keep Symbols       Debugger     • Strip Debug Information       GDB Server     • Strip Symbols       J-Link     • Allow Multiple Symbol Definition       Loader     • No Find Size Warning       Simulator     • Suppress Warning on Mismatch       Target Control     • Treat Linker Warnings as Errors	User Build Step	Linker Symbol Definitions	hese handler mountes	
Debugger     • Strip Debug Information     No       GDB Server     • Strip Debug Information     No       J-Link     • Allow Multiple Symbol Definition     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified       Target Trace     Section Placement Macros	Debugger         Strip Debug Information         No           GDB Server         • Strip Symbols         No           J-Link         • Allow Multiple Symbol Definition         No           Loader         • No Wide Char Size Warning         No           Simulator         • Suppress Warning on Mismatch         No           Target Control         • Treat Linker Warnings as Errors         No	Debug	Keep Symbols		
Strip Symbols     No       GDB Server        • Strip Symbols     No       J-Link        • Allow Multiple Symbol Definition     No       Loader        • No Wide Char Size Warning     No       Simulator        • Suppress Warning on Mismatch     No       Target Control        • Treat Linker Warnings as Errors     No modified       Target Trace        Section Placement Macros	GDB Server     • Strip Symbols     No       GDB Server     • Allow Multiple Symbol Definition     No       J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No	Debugger	Strip Debug Information	No	
GDB Server        • Allow Multiple Symbol Definition      No       J-Link        • No Enum Size Warning      No       Loader        • No Wide Char Size Warning      No       Simulator        • Suppress Warning on Mismatch      No       Target Control        • Treat Linker Warnings as Errors      No       Target Script        • Section Placement Macros	GDB Server <ul> <li>Allow Multiple Symbol Definition</li> <li>No</li> </ul> No     No <td>Debugger</td> <td>Strip Symbols</td> <td>Ne</td> <td></td>	Debugger	Strip Symbols	Ne	
J-Link <ul> <li>No Enum Size Warning</li> <li>No Wide Char Size Warning</li> <li>No Wide Char Size Warning</li> <li>No Wide Char Size Warning</li> <li>Suppress Warning on Mismatch</li> <li>No</li> <li>Traget Control</li> <li>Treat Linker Warnings as Errors</li> <li>No modified</li> <li>Section Placement Macros</li> <li>Section Placement Macros</li> <li>No</li> <l< td=""><td>J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified</td><td>GDB Server</td><td>Allow Multiple Symbol Definition</td><td>No</td><td></td></l<></ul>	J-Link     • No Enum Size Warning     No       Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	GDB Server	Allow Multiple Symbol Definition	No	
Loader        • No Wide Char Size Warning     No       Simulator        • Suppress Warning on Mismatch     No       Target Control        • Treat Linker Warnings as Errors     No       Target Script        • Section Placement Macros	Loader     • No Wide Char Size Warning     No       Simulator     • Suppress Warning on Mismatch     No       Target Control     • Treat Linker Warnings as Errors     No modified	J-Link	No Enum Size Warning	No	
Simulator     * Suppress Warning on Mismatch     No       Target Control     * Treat Linker Warnings as Errors     No modified       Target Script     Section Placement Macros	Simulator    Suppress Warning on Mismatch  Target Control  Treat Linker Warnings as Errors  No modified	Loader	No Wide Char Size Warning	No	
Target Control        • Treat Linker Warnings as Errors      No     modified       Target Script        • Section Placement Macros	Target Control	Simulator Target Control	<ul> <li>Suppress Warning on Mismatch</li> </ul>	No	
Target Script Target Trace Section Placement Macros			<ul> <li>Treat Linker Warnings as Errors</li> </ul>	No modified	
Target Trace Section Placement Macros	Target Script	Target Script	· · · · · · · ·		
Marro values to substitue in section placement redex - MACRO1-value1/MACRO2-value2	Target Trace Section Placement Macros	Target Trace	Section Placement Macros	- MACRO1-uslus1MACRO2-uslus2	

#### Figure 12: SES project RAM settings

Since the DFU service is being added alongside the Nordic UART Service (NUS), additional memory needs allocated to flash. This is accomplished by increasing the start address of RAM and decreasing RAM size. The new characteristics are 16-bytes each; increase the RAM\_START by 0x10 and decrease the RAM\_SIZE by 0x10.

For nRF5 SDK v16.0.0, the original values are:

```
RAM_START=0x20002ae8
RAM_SIZE=0x3d518
```

The new values will be:

RAM\_START=0x20002af8 RAM\_SIZE=0x3d508

Select "OK" for each open dialog window to return to the main SES window.

⚠ Forgetting to change the RAM start and size is a common error that will cause a hard fault stop.



### 5.2 sdk\_config.h

Open the project sdk\_config.h file.

There are several settings that need modified to enable DFU. Each line in this list are in separate locations throughout the file. Perform a search for each of these lines and change its value as noted:

```
#define BLE_DFU_ENABLED 1 // was 0
#define NRF_PWR_MGMT_CONFIG_AUTO_SHUTDOWN_RETRY 1 // was 0
#define NRF_SDH_BLE_VS_UUID_COUNT 2 // was 1
#define NRF_SDH_BLE_SERVICE_CHANGED 1 // was 0
```

### 5.3 Libraries

The DFU library and driver files need added to the project.

Add a new folder in the project called nRF\_DFU and add these existing files from the DFU directory of the Bluetooth low energy components:

#### From the directory:

C:\u-blox\nRF5 SDK 17.0.0 9d13099\components\ble\ble services\ble dfu

Add these files:

ble\_dfu.c
ble\_dfu\_bonded.c
ble\_dfu\_unbonded.c

#### And from the directory:

C:\u-blox\nRF5\_SDK\_17.0.0\_9d13099\components\libraries\bootloader\dfu

#### Add this file:

nrf\_dfu\_svci.c

After adding the folder and files, the project will have this section:

```
    nRF_DFU 4 files
    ble_dfu.c
    ble_dfu_bonded.c
    ble_dfu_unbonded.c
    ble_dfu_unbonded.c
    anf_dfu_svci.c
```

#### Figure 13: Buttonless UART DFU project - added files

The project still has the same ble\_app\_uart functionality as the original example application, though now it also includes everything needed to start adding the buttonless DFU function.

### 5.4 main.c

With the "behind the scenes" items configured, the DFU code can be added to main.c.

Several header files are needed. Add these to the other header files near the top of main.c:

// BEGIN Block Added for DFU
#include "nrf\_dfu\_ble\_svci\_bond\_sharing.h"
#include "nrf\_svci\_async\_function.h"
#include "nrf\_svci\_async\_handler.h"
#include "ble\_dfu.h"
#include "nrf bootloader info.h"

// END Block Added for DFU



The advertising name will be changed to confirm the DFU version is running. Search for "Nordic\_UART" and change it to "UART\_DFU":

#define DEVICE\_NAME "UART\_DFU" /\*\*< Name of device. Will be included in the advertising
data. Changed for DFU \*/</pre>

A new event handler function is added to act on the data being sent to the DFU service. Place this code above the "assert nrf callback" function:

```
/**@brief Function for handling DFU events
 * @details This function is called when entering buttonless DFU
 * @param[in] event Buttonless DFU event.
 */
static void ble_dfu_buttonless_evt_handler(ble_dfu_buttonless_evt_type_t event)
{
    switch (event)
    {
        case BLE DFU EVT BOOTLOADER ENTER PREPARE:
            NRF LOG INFO("Device is preparing to enter bootloader mode\r\n");
            break:
        case BLE DFU EVT BOOTLOADER ENTER:
            NRF LOG INFO("Device will enter bootloader mode\r\n");
            break;
        case BLE DFU EVT BOOTLOADER ENTER FAILED:
            NRF LOG ERROR("Device failed to enter bootloader mode\r\n");
           break;
        default:
           NRF LOG INFO("Unknown event from ble dfu.\r\n");
           break;
    }
}
```

The next added function handles the power management. The bootloader service writes a value to a persistent register, then issues a system reset. On restart, the bootloader reads this value to determine whether it should continue, or pass control to the application.

Code can be added here to prevent the DFU from starting if something critical is going on within the application.

Add this function code below the DFU event handler that was just added:

```
/**@brief Function for handling bootloader power management events
 * @details This function is called to set a persistent register which informs the
   bootloader it should continue or pass control back to the application
 * @param[in] event Power management event.
 */
static bool app shutdown handler (nrf pwr mgmt evt t event)
{
    switch (event)
    {
        case NRF PWR MGMT EVT PREPARE DFU:
            NRF LOG INFO("Power management wants to reset to DFU mode\r\n");
            // \bar{\mathrm{C}}\mathrm{hange} this code to tailor to your reset strategy.
            // Returning false here means that the device is not ready
            // to jump to DFU mode yet.
            11
            // Here is an example using a variable to delay resetting the device:
            11
            /* if (!im ready for reset)
```





```
{
                     return false;
                 }
             * /
             break;
         default:
             // Implement any of the other events available
             // from the power management module:
             // -NRF_PWR_MGMT_EVT_PREPARE_SYSOFF
// -NRF_PWR_MGMT_EVT_PREPARE_WAKEUP
             // -NRF_PWR_MGMT_EVT_PREPARE_RESET
             return true;
    }
    NRF LOG INFO("Power management allowed to reset to DFU mode\r\n");
    return true;
}
NRF PWR MGMT HANDLER REGISTER (app shutdown handler, 0); // persistent register for \
                                                                   determining DFU status \
```

on startup

The DFU service is initialized by adding it to the services init() function, in **bold**, italic below:

```
static void services init(void)
{
   uint32_t err_code;
ble_nus_init_t nus_init;
    nrf_ble_qwr_init_t qwr_init = {0};
// Initialize Queued Write Module.
    qwr_init.error_handler = nrf_qwr_error_handler;
    err_code = nrf_ble_qwr_init(&m_qwr, &qwr_init);
   APP ERROR_CHECK (err_code);
// Initialize NUS (Nordic UART Service)
    memset(&nus_init, 0, sizeof(nus_init));
    nus init.data handler = nus data handler;
    err code = ble_nus_init(&m_nus, &nus_init);
    APP ERROR CHECK (err code);
// BEGIN Block Added for DFU
// ONLY ADD THIS BLOCK TO THE EXISTING FUNCTION
    // Initialize the DFU service
    ble_dfu_buttonless_init_t dfus_init =
    ſ
        .evt_handler = ble_dfu_buttonless_evt_handler
    ł;
    err code = ble dfu buttonless init(&dfus init);
    APP_ERROR_CHECK(err_code);
// END Block Added for DFU
```

Check that the application compiles without errors.

J

Do not load the application to the BMD-340-EVAL at this time. A different process will be used to load it to at a later step.

### 5.5 Hex file generation

Three of the four required components of the project are completed:

- Bootloader
- Application with buttonless DFU code included
- SoftDevice



The last crucial part – bootloader settings – needs generated. The bootloader settings values tell the bootloader about the application:

- Is a valid application present?
- What is the start address of the application?
- What are the version numbers of the bootloader, SoftDevice, and application?

This file is generated by nrfutil with the new application file as input. The output is a new hex file which is then combined with the bootloader, SoftDevice and application hex files. A single hex file is then generated from the four files and used to program a "blank-part" with everything in one pass over the SWD port.

To avoid long file paths when using the nrfutil command, copy the hex files generated up to this point to a common directory:

md c:\u-blox\hexfiles

Copy the respective hex files to this new location:

```
\label{eq:copy_c:lu-blox/nRF5_SDK_17.0.0_9d13099\components\softdevice\s140\hex\s140_nrf52_7.0.1\_softdevice\hex\c:\u-blox\hexfiles
```

```
copy C:\u-
blox\nRF5_SDK_17.0.0_9d13099\examples\dfu\secure_bootloader\pca10056_s140_ble_working\ses\
Output\Release\Exe\secure_bootloader_ble_s140_pca10056.hex c:\u-blox\hexfiles
```

```
copy C:\u-
blox\nRF5_SDK_17.0.0_9d13099\examples\ble_peripheral\ble_app_uart_working_dfu\pca10056\s14
0\ses\Output\Release\Exe\ble_app_uart_pca10056_s140.hex c:\u-blox\hexfiles
```

cd c:\u-blox\hexfiles

Run nrfutil to generate the bootloader settings file:

```
nrfutil settings generate --family NRF52840 --application ble_app_uart_pca10056_s140.hex -
-application-version-string "1.2.3" --bootloader-version 0 --bl-settings-version 2
bl_setting.hex
```

Note: Generating a DFU settings page with backup page included. This is only required for bootloaders from nRF5 SDK 15.1 and newer. If you want to skip backup page generation, use --no-backup option. Generated Bootloader DFU settings .hex file and stored it in: bl\_setting.hex Bootloader DFU Settings: \* File: bl\_setting.hex \* Family: NRF52840

- \* Start Address: 0x000FF000
- \* CRC: 0x9DA9D5DF
- \* Settings Version: 0x0000002 (2)
- \* App Version: 0x000027DB (10203)
- \* Bootloader Version: 0x0000000 (0)
- \* Bank Layout: 0x0000000
- \* Current Bank: 0x0000000
- \* Application Size: 0x0000BC28 (48168 bytes)
- \* Application CRC: 0xD5A02BF3
- \* Bank0 Bank Code: 0x00000001
- \* Softdevice Size: 0x0000000 (0 bytes)
- \* Boot Validation CRC: 0xF906A7EC
- \* SD Boot Validation Type: 0x0000000 (0)
- \* App Boot Validation Type: 0x0000001 (1)

c:\u-blox\hexfiles>

The family of NRF52840 is selected for the BMD-340. See the --help output for other nRF5x versions.



All components are now ready:

- Bootloader
- Application with buttonless DFU code included
- SoftDevice
- Bootloader Settings

The Nordic Semiconductor mergehex is used to create the resulting hex file. mergehex will allow a maximum of three input files, so two steps are required:

mergehex --merge bl\_setting.hex secure\_bootloader\_ble\_s140\_pca10056.hex
s140\_nrf52\_7.0.1\_softdevice.hex --output bl\_set\_s140.hex

Parsing input hex files. Merging files. Storing merged file.

mergehex --merge bl\_set\_s140.hex ble\_app\_uart\_pca10056\_s140.hex --output bl\_set\_s140\_app.hex

Parsing input hex files. Merging files. Storing merged file.

This final hex file – bl\_set\_s140\_app.hex – with all the parts included is what is typically used on a production line, or for pre-programming the module ahead of assembly into the end-product.

Other details may be added to the hex file, including the public Bluetooth address noted in section
 2.1 above.



## 6 Test DFU

### 6.1 Program and test application

Everything is now in place to test the new application.

Erase the BMD-340 to ensure nothing is left-over from previous work:

```
nrfjprog -f nrf52 --recover
Recovering device. This operation might take 30s.
Erasing user code and UICR flash areas.
```

The final step is to load the hex file to the BMD-340:

#### nrfjprog --program bl\_set\_s140\_app.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.
```

Power-cycle the BMD-340-EVAL. The application will start advertising and blink LED1 as before.

#### Connect to it with nRF Connect:

8 nRF Connect v3.4.2	Bluetooth Low Energy		-	
≡ EF7905	FEC31E 🔻 🔍 📥 Connection Map 🗉 Server	Setup		
■ nRF5x ■ EF:79:05:FI	CC3:1E		Discovered device ► Start scan ► Options Sort by signal strengt ilter: Device name or	es I Clear h address
Generic Access	F1:D7:E9:CF:E4:4A		:Unknown name> 39:86:D0:00:FC:C6 > Details	-44 dBmI Connect &
	Generic Attribute		JART_DFU 1:D7:E9:CF:E4:4A • Details	-35 dBm 📶 Connect 🔗
	VART over BLE     Secure DFU	E	<b>1B-00022885</b> 30:38:29:9D:60:24 ▸ Details	-74 dBm 🔐
		<	: <b>Unknown name&gt;</b> 57:FA:EE:19:08:28 • Details	-69 dBm 🔐
		1	:Unknown name> IA:ED:53:6E:BA:04 > Details	-66 dBm 📶
Log 17:40:32.449 17:40:34.169	Scan started Connecting to device			
17:40:34.236 17:40:34.297 17:40:34.344 17:40:34.948 17:40:39.325	Connected to device F1:D7:E9:CF:E4:4A ATT MTU updated for device F1:D7:E9:CF:E4:4A, new Data length updated for device F1:D7:E9:CF:E4:4A, ne Attribute value read, handle: 0x03, value (0x): 55-41-5: Connection parameters updated for device F1:D7:E9:C	value is 247 w value is 251 2-54-5F-44-46-55 CF:E4:4A: interval 20ms. tim •		
4	· F ··································	•		

Figure 14: nRF Connect with application showing secure DFU icon

The bootloader icon is shown as with the button DFU example. The new name, UART\_DFU, and the Nordic UART Service, UART over BLE, are also present.



### 6.2 Prepare and test update

Since the application is already compiled to a hex file, it can be used as the source application file. The application version is stepped to inform the bootloader that it is a new, valid update is available.

In the hexfile directory, nrfutil is used to generate the new zip file:

```
nrfutil pkg generate --application ble_app_uart_pca10056_s140.hex --application-version-
string "2.0.0" --hw-version 52 --sd-req 0xCA --key-file C:\u-
blox\nRF5_SDK_17.0.0_9d13099\examples\dfu\dfu_private_key.pem app_v2.zip
```

From nRF Util, start the DFU process and send the update:

8 nRF Connect v3.4.2 - Blu	ietooth Low Energy			- 🗆 X
EF7905FE0	Device Firm	& Connection Man IE Server Setun	-40	N
	Device I III		d de	vices
<b>nRF5x</b> • EF:79:05:FE:C3:	Zip file	C:\u-blox\hexfiles\app_v2.zip	Choose an	T Clear
	Package	1:+:	nal sti	rength
	info	bin file: ble app uart pca10056 s140.bin	e nar	ne or address
Generic Access		dat_file: ble_app_uart_pca10056_s140.dat	me>	-44 dBm
Connection Attacheday			C6	Connect C
Generic Attribute				Connect o
			Start DFU	-25 dBm u
9			4A	
				Connect &
			Close	
			B0:38:20:00:60:24	-74 dBm
		Secure DFU	» Dotaile	Connect Ø
			<unknown name=""></unknown>	-69 dBm 🔐
			Dataila	Connect Ø
			, Details	
			<unknown name=""></unknown>	-66 dBm 🔐
			TA:ED:53:6E:BA:04	Connect Ø
			Details	
Log		8 8	<b>V</b>	
17:40:48.001	DFU completed suc	cesstully.	-	
17:40:58.184	Connecting to devic			
17:40:30:321	ATT MTI Lundated f	or device E1:D7:E9:CE:E4:4A		
17:46:58 473	Data length updated	d for device F1:D7:E9:CF:E4:4A, new value is 251		
17:46:59.091	Attribute value read	, handle: 0x03, value (0x): 55-41-52-54-5F-44-46-55		
17:47:03.401	Connection parame	ters updated for device F1:D7:E9:CF:E4:4A: interval 20ms, t	im 👻	
4			•	

#### Figure 15: SES ready for update from application

To observe version checking, create another zip file, with a version string of "1.0.0".

```
nrfutil pkg generate --application ble_app_uart_pca10056_s140.hex --application-version-
string "1.0.0" --hw-version 52 --sd-req 0xCA --key-file C:\u-
blox\nRF5_SDK_17.0.0_9d13099\examples\dfu\dfu_private_key.pem app_v1.zip
```

If the new file is selected with nRF Connect, an error will be generated since a newer version is present on the BMD-340.



8 nRF Connect v3.4.2 - Bluetoo	th Low Energy	-		$\times$
≡ EF7905FEC315	Error	×		
»	EIIOI	d device	es	
<b>3 nRF5x</b> - EF:79:05:FE:C3:1E	When writing 'EXECUTE' command to Control Point Characteristic of DFU Target: Operation code 4 (EXECUTE) failed on DFU Target. Result code 11 (EXTENDED_ERROR) Extended error code 5 (FW_VERSION_FAILURE)	n an mainteanaite nai strengti e name or	l Clear	
Generic Access     Generic Attribute	Close	<b>3</b> 24	-75 di Connec	Bm ail st Ø
	► Stalt DFU	<b>me&gt;</b>	-50 di	Bm .al
	Close	me>	-73 di	Bm <sub>at</sub> i
	⇒ Details	99:EB:F2	Connec	rt Ø
	<ul> <li><unknow< li=""> <li>44:82:85:0</li> <li>→ Details</li> </unknow<></li></ul>	/n name> :A:B9:DD s	-34 di	Bm <b>all</b> st <i>Ø</i>
	<unknow 50:01:92:6 • Details</unknow 	/n name> 4:58;40	-70 de	Bm ail st <i>©</i>
Log	F1:D7:E9:C	F <b>U</b> SF:E4:4A	-36 de	Bm <b>l</b> st Ø
I/:50:53.075         Attri           17:50:53.092         Attri           17:50:53.216         Attri           17:50:53.232         Attri           17:50:53.279         Attri           17:50:53.292         Attri           17:50:53.292         Attri	> Details         > Dute value changed, handle: 0x0F, value (0x): 01-01-8E-00-00-00         bute value changed, handle: 0x0F, value (0x): 03         bute value changed, handle: 0x0F, value (0x): 03         bute value changed, handle: 0x0F, value (0x): 03         bute value changed, handle: 0x0F, value (0x): 00-03-01-8E-00-00-00-50-40-BA         bute value changed, handle: 0x0F, value (0x): 60-03-01-8E-00-00-00-50-40-BA         bute value changed, handle: 0x0F, value (0x): 60-03-01-8E-00-00-00-50-40-BA         bute value changed, handle: 0x0F, value (0x): 60-04-0B-05         failed with error: When writing 'EXECUTE' command to Control Point Charact -			

Figure 16: nRF Connect showing error when attempting an "old" update

Version checking is performed for all three components: bootloader, SoftDevice, and application.

There are many more aspects of the Nordic DFU service, such as activating additional transports (UART, USB, Bluetooth Mesh, Zigbee, and Thread), additional security with bond forwarding, and so on. For further information, see reference [18].



## Appendix

## A Glossary

Abbreviation	Definition		
ASCII	American Standard Code for Information Interchange		
ARM	Arm (Advanced RISC Machines) Holdings		
CPU	Central Processing Unit		
DFU	Device Firmware Update		
DK	Development Kit (see EVK)		
EVK	EValuation Kit		
FICR	Factory Information Control Registers		
GCC	GNU Compiler Collection		
GNU	Recursive acronym "GNU Not Unix"		
IDE	Integrated Development Environment		
LED	Light Emitting Diode		
MAC address	Media Access Control address: a unique identifier assigned to a network interface controller		
NUS	Nordic UART Service		
RAM	Random Access Memory		
SDK	Software Development Kit		
SES	SEGGER Embedded Studio		
SoftDevice	Bluetooth low energy stack provided by Nordic Semiconductor		
TLA	Three Letter Acronym		
UART	Universal Asynchronous Receiver Transmitter		
UICR	User Information Control Registers		
USB	Universal Serial Bus		
WSL	Windows Subsystem for Linux		

Table 2: Explanation of the abbreviations and terms used



## **Related documents**

- [1] ANNA-B112 system integration manual, UBX-18009821
- [2] NINA-B1 system integration manual, UBX-15026175
- [3] NINA-B3 system integration manual, UBX-17056748
- [4] BMD-345 data sheet, UBX-19039908
- [5] u-blox package information guide, UBX-14001652
- [6] Using the public IEEE address from UICR application note, UBX-19050198
- [7] Nordic Semiconductor nRF5 SDK
- [8] Nordic Semiconductor nRF Connect for Desktop
- [9] Nordic Semiconductor nRF Command Line Tools
- [10] Nordic Semiconductor nRF Util GitHub repository
- [11] Python scripting language download
- [12] Micro\_ecc encryption libraries backend instructions
- [13] Git version control download
- [14] GNU Arm Embedded Toolchain (GCC for Arm) download
- [15] GNU make download
- [16] UART example test instructions
- [17] Nordic Semiconductor bootloader details
- [18] Nordic Semiconductor Bluetooth services tutorial
- [19] Nordic Semiconductor Bootloader and DFU modules

For product change notifications and regular updates of u-blox documentation, register on our website, www.u-blox.com.

## **Revision history**

Revision	Date	Name	Comments
R01	16-Dec-2019	brec	Initial release
R02	02-May-2020	brec	Added reference to UBX-19050198, corrected minor typographical errors
R03	08-Jan-2021	brec	Changed style, updated for nRF5 SDK v17.0.0, nRF Connect v3.4.2



## Contact

For complete contact information, visit us at www.u-blox.com.

#### u-blox Offices

#### North, Central and South America

#### u-blox America, Inc.

Phone: +1 703 483 3180 E-mail: info\_us@u-blox.com

#### **Regional Office West Coast:**

Phone: +1 408 573 3640 E-mail: info\_us@u-blox.com

#### **Technical Support:**

Phone: +1 703 483 3185 E-mail: support@u-blox.com

#### Headquarters Europe, Middle East, Africa

#### u-blox AG

Phone: +41 44 722 74 44 E-mail: info@u-blox.com Support: support@u-blox.com

#### Asia, Australia, Pacific

#### u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811 E-mail: info\_ap@u-blox.com Support: support\_ap@u-blox.com

#### **Regional Office Australia:**

Phone: +61 2 8448 2016 E-mail: info\_anz@u-blox.com Support: support\_ap@u-blox.com

#### **Regional Office China (Beijing):**

Phone: +86 10 68 133 545 E-mail: info\_cn@u-blox.com Support: support\_cn@u-blox.com

#### **Regional Office China (Chongqing):**

Phone: +86 23 6815 1588 E-mail: info\_cn@u-blox.com Support: support\_cn@u-blox.com

#### **Regional Office China (Shanghai):**

Phone: +86 21 6090 4832 E-mail: info\_cn@u-blox.com Support: support\_cn@u-blox.com

#### Regional Office China (Shenzhen):

Phone: +86 755 8627 1083 E-mail: info\_cn@u-blox.com Support: support\_cn@u-blox.com

#### **Regional Office India:**

Phone: +91 80 405 092 00 E-mail: info\_in@u-blox.com Support: support\_in@u-blox.com

#### Regional Office Japan (Osaka):

Phone: +81 6 6941 3660 E-mail: info\_jp@u-blox.com Support: support\_jp@u-blox.com

#### Regional Office Japan (Tokyo):

Phone: +81 3 5775 3850 E-mail: info\_jp@u-blox.com Support: support\_jp@u-blox.com

#### **Regional Office Korea:**

Phone: +82 2 542 0861 E-mail: info\_kr@u-blox.com Support: support\_kr@u-blox.com

#### Regional Office Taiwan:

Phone: +886 2 2657 1090 E-mail: info\_tw@u-blox.com Support: support\_tw@u-blox.com