XM122 Software Development Guide

User Guide

XM122 Software Development Guide

User Guide

Author: Acconeer AB

Version:a111-v2.12.0

Acconeer AB June 20, 2022

Contents

1 Introduction

The Acconeer Software Developmen Kit (SDK) enables customers to develop their own software that can be executed on the module. This enables full control of all the peripherals and to maximize the performance and power consumption for a specific use case.

The SDK comes with a number of example applications that can be used as a starting point when developing your own application. These applications can be downloaded and executed using the methods described in “Installing Software Image” at page 4.

When developing your own application we recommend that you setup a development environment as described in “Setting up a Development Environment at page 7.

2 Installing Software Image

The XM122 comes with a preinstalled bootloader supporting the Device Firmware Upgrade (DFU) protocol, which allows the customer to program (”flash”) the XM122 with application software using only a USB cable connected to the XB122 board.

Another option is to flash the device over-the-air using its Bluetooth Low Energy (BLE) capabilities. This requires the aid of Nordic Semiconductor’s nRF Toolbox or nRF Connect app for Android or iPhone (freely available on Google Play and Apple’s App Store, respectively).

The third option is to program the device using a SWD debugger, this requires additional hardware which is suitable when developing your own applications.

A device without application software will enter DFU mode automatically during boot. It will continue to do so until the device has been flashed. After that, the device can be made to enter DFU mode again by holding down the DFU button on XB122 while pressing the Reset button.

2.1 Android DFU

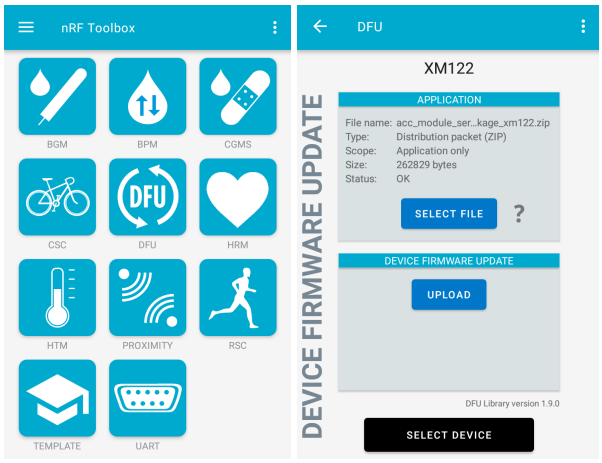
In order to install or update the software on the XM122 module we recommend using the ‘nRF Toolbox‘ application for Android which is available on Google Play Store.

1. Connect the XB122 together with XM122 to your PC with a micro USB cable to the USB connector
2. Press and hold the “DFU” button on the board
3. Press the “RESET” button (still holding the “DFU” button)
4. Release the “RESET” button
5. Release the “DFU” button

Your XM122 device is now in ‘DFU‘ mode waiting for a software uprade procedure to be started.

1. Start the ‘nRF Toolbox‘ application on your phone 

2. Transfer the zip-file, e.g. ‘example_detector_distance.zip‘ to your phone.
3. Press ‘DFU‘
4. Press ‘SELECT FILE‘
5. Make sure ‘Distribution packet (ZIP)‘ is selected and press ‘OK‘
6. Select the zip file, e.g. ‘example_detector_distance.zip‘
7. Make sure ‘Application only‘ scope is selected and press ‘OK‘
8. Press ‘SELECT DEVICE‘ and select XM122
9. Press ‘UPLOAD‘
10. Wait until the update have finished



2.2 UART DFU

When the device is in DFU mode, it can be programmed from the command line using nrfutil, which is a Python program provided by Nordic that can be installed using:

```
pip install nrfutil
```

1. Connect the XB122 together with XM122 to your PC with a micro USB cable to the USB connector
2. Press and hold the “DFU” button on the board
3. Press the “RESET” button (still holding the “DFU” button)
4. Release the “RESET” button
5. Release the “DFU” button

Your XM122 device is now in ‘DFU‘ mode waiting for a software upgrade procedure to be started.

Flash the device as follows (using “example_detector_distance.zip” as an example):

```
nrfutil dfu serial -pkg example_detector_distance.zip -p /dev/ttyUSB0
```

where “/dev/ttyUSB0” is the serial port designation assigned by the operating system to the XB122 board when plugging it in. On Windows this might be ’COM1’ or similar instead.

After a few seconds, nrfutil will print a message saying “No trigger interface found”. That’s because USB implementations of DFU may include a special interface for putting the connected device into DFU mode. No such interface is available here because we’re only using USB to emulate a UART connection.

Another few seconds later, the actual DFU process will begin. When finished, the device will reset and boot into the newly installed application.

2.3 Flash using J-Link

Installing the software image with a J-Link can be done with help of nrfjprog:

1. Download and install “nRF5x Command Line Tools” from www.nordicsemi.com
2. Download and install “J-Link Software and Documentation Pack” from www.segger.com
3. Download and extract ‘S140 SoftDevice version 6.1.1” from www.nordicsemi.com

```
nrfjprog -f nrf52 --eraseall
nrfjprog -f nrf52 --program acc_module_server.hex --sectorerase --verify
nrfjprog -f nrf52 --program s140_nrf52_6.1.1_softdevice.hex --sectorerase --
   verify
nrfjprog -f nrf52 --reset
```

Note that this will remove the bootloader which can be restored if needed.

2.4 Restore Bootloader

A backup copy of the bootloader is delivered as part of the "production_sw.hex" file, which also includes Nordic's softdevice (containing their Bluetooth stack). Bootloader and softdevice has been merged into a single hex file, because as of version 15.3.0 of the Nordic SDK they cannot be flashed separately.

To restore an XM122 device to its factory condition, do the following using a J-Link:

1. Download and install "nRF5x Command Line Tools" from www.nordicsemi.com
2. Download and install "J-Link Software and Documentation Pack" from www.segger.com

```
nrfjprog -f nrf52 --erasepage 0xFE000-0x100000
nrfjprog -f nrf52 --program production_sw.hex --sectorerase --verify
nrfjprog -f nrf52 --reset
```

3 Setting up a Development Environment

In order to develop your own applications you need to set up a development environment. The XM122 is based on a nRF52840 SoC by Nordic Semiconductor. Additional information and support can be found on their support site.

3.1 Using a Debugger

In order to debug your applications it is recommended to use a SWD debugger. We recommend that you use a SEGGER JLink debug probe e.g. J-Link BASE Compact.

Figure 1: J-Link Base Compact

The J-Link BASE Compact can be used to set breakpoints and single step the program in an easy way.

3.2 Building From the Command Line

All example applications can be built from the command line using “make”.

1. Install “nRF5x Command Line Tools” with “pip install nrfutil” or from www.nordicsemi.com
2. Download the nRF5-SDK (version 15.3) from www.nordicsemi.com.
3. Extract the archive into a folder, e.g. “/home/acconer/sdk/”
4. Download “GCC ARM Embedded 9-2020-q2-update” from developer.arm.com.
5. Extract the archive into a folder, e.g. “/home/acconer/compilers/”
6. Download and extract the Acconeer SDK zip file, e.g. “/home/acconer/acconeer_xm122/”

```
$ cd /home/acconer/acconeer_xm122
$ export GNU_INSTALL_ROOT=/home/acconer/compilers/gcc-arm-none-eabi-9-2020-
   q2-update/bin/
$ export NRF_SDK_ROOT="/home/acconer/sdk/nRF5_SDK_15.3.0_59ac345/"
$ make -j10
```

The above will compile all example applications. It will also generate DFU packages that can be used to install the example using any of the methods described in “Installing Software Image” at page 4

3.2.1 Download Software Using a J-Link

You can also flash using a J-Link from the command line. First install the “J-Link Software and Documentation Pack” from www.segger.com.

```
$ make flash_softdevice
$ make flash_detector_distance
```

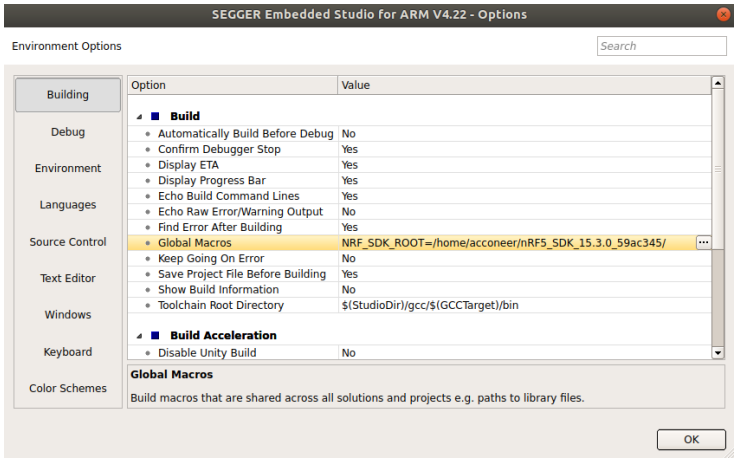
Note that the “make flash_softdevice” will disable the bootloader and its DFU functionality. Without this, the device would enter DFU mode during boot because previously generated “bootloader settings” (which are automatically generated when DFU is used) do not match the new application.

See “Restore Bootloader” for how to restore the bootloader again.

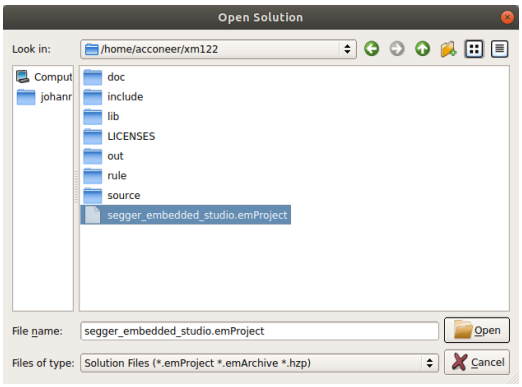
3.3 SEGGER Embedded Studio

If you prefer to use an integrated development environment we recommend that you use the SEGGER Embedded Studio together with a SEGGER J-Link debug probe. SEGGER Embedded Studio is free to use with Nordic Semiconductor Devices.

1. Download the nRF5-SDK (version 15.3) from www.nordicsemi.com.
2. Extract the archive into a folder, e.g. “/home/acconer/sdk/”
3. Download and install SEGGER Embedded Studio at www.SEGGER.com.
4. Download and extract the Acconeer SDK zip file, e.g. “/home/acconer/acconeer_xm122/”
5. Start SEGGER Embedded Studio, then select “Tools/Options...”.
6. Under “Building” add “NRF_SDK_ROOT=/home/acconeer/nRF5_SDK_15.3.0_59ac345/” to “Global Macros” (Make sure underscores are included if path is copied)



Select “File/Open Solution...” and browse to the folder where you unpacked the zip file, then select “segger_embedded_studio.emProject” and click on “Open”



3.3.1 Running the Program

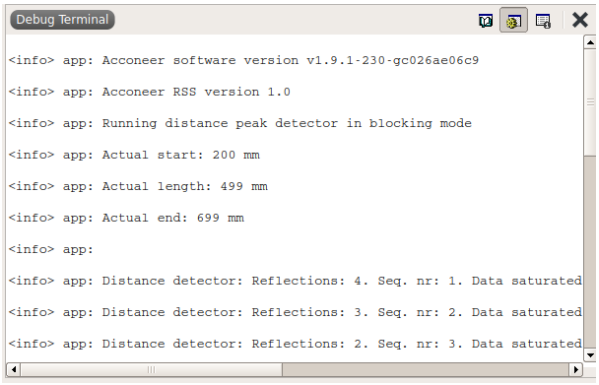
Build the software by pressing “F7” and then start debugging by pressing “F5”. This will automatically flash the XM122 and jump to the “main()” function.

Note that this method does not flash the softdevice which means that the bootloader is still present. This also means that the SoC will enter DFU mode if a hard reset is performed as there are no valid bootloader settings (automatically generated when DFU is used). An easy way around this is to flash the softdevice when developing your own applications. See devzone.nordicsemi.com for how to program the softdevice using SEGGER Embedded Studio. The softdevice for nRF52840 can be found under “$(NRF_SDK_ROOT)/components/softdevice/s140/hex/s140_nrf52_6.1.1_softdevice.hex”.

In order to restore the bootloader functionality see “Restore Bootloader”.

3.3.2 Debug Output

The debug output can be seen in the “Debug Terminal”, also see “Debug Logging Output” at page 10 for other ways to obtain the logs.



4 Debug Logging Output

RSS and module server logs can be retrieved either using a J-Link or over UART.

4.1 UART Logs

The easier way to see the logs is using a terminal program. E.g.

```
picocom --imap lfcrlf --baud 115200 /dev/ttyUSB0
```

In order to exit the “picocom” program press Ctrl-a and the Ctrl-x.

Baudrate	115200
Byte size	8
Parity	None
Stop bits	1

Table 1: Debug UART Settings

4.2 SEGGER’s Real Time Transfer (RTT)

If you are using a J-Link connected to the XB122 you can also access the debug logs over SWD interface. Make sure you have installed the “J-Link Software and Documentation Pack” from www.segger.com then start the logging using the following commands:

```
JLinkRTTLogger -if swd -device NRF52840_XXAA -speed 4000 -RTTChannel 0 /tmp/
   log.txt
```

Using above command the logs will be written to /tmp/log.txt. In another terminal then write

```
tail -f /tmp/log.txt
```

5 Bluetooth Advertisement Example

The SDK comes with example applications that combines low power mode with advertising the results over Bluetooth utilizing the nRF Beacon advertisement type. One example is using the envelope service and the other is using the sparse service. The output from the Acconeer service is advertised in the Major and Minor fields of the nRF Beacon advertisement type and can be visualized using the 'nRF Connect' application.

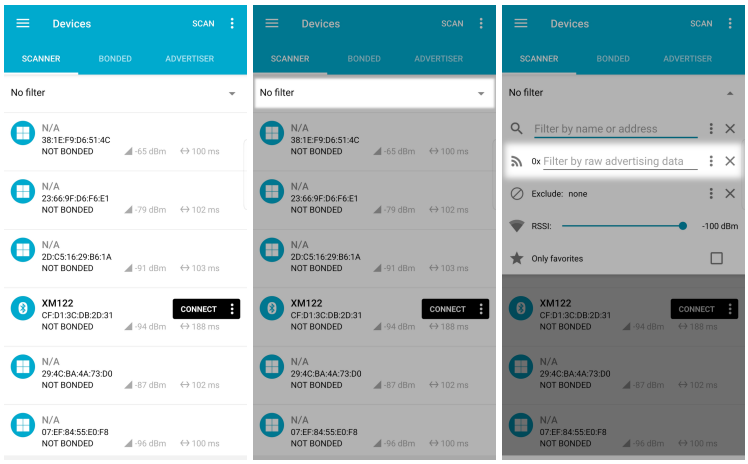
The 'nRF Connect' application for Android is available on Google Play Store:

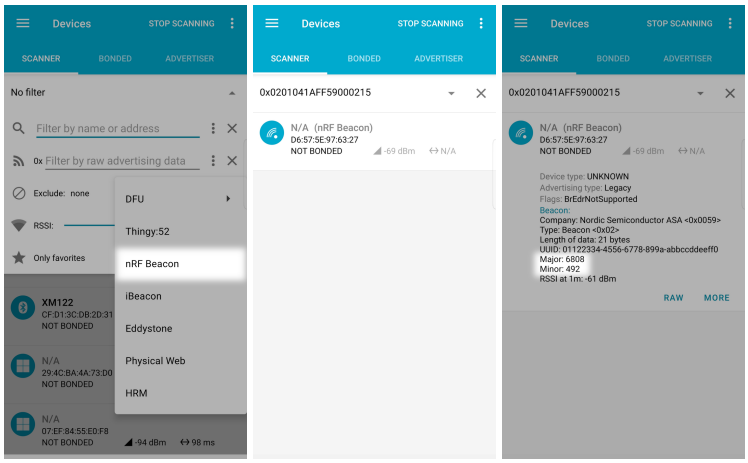
Follow the instruction in "Installing Software Image" and install either the envelope example application 'example_low_power_service_envelope.zip', or the sparse example application 'example_low_power_service_sparse.zip' to your module. This example will run the service every 10 seconds and advertise the result using the nRF Beacon type at the same interval. The data in the Major and Minor fields will be different depending on the service used:

	Major	**Minor**
Envelope	Max amplitude	Max amplitude index
Sparse	The first element in the sparse data array	Number of elements in the sparse data array

Table 2: Service output

1. Start the 'nRF Connect' application on your phone
2. Go to the 'Devices' screen
3. Press the down arrow in the filter text box
4. Press the three circles in the 'Filter by raw advertising data' text box and select nRF Beacon
5. Minimize the filter text box and wait for a device to show up inte list
6. Show more information about the deivce by pressing on it
7. The service data output is now visualized in the Major and Minor field





6 Disclaimer

The information herein is believed to be correct as of the date issued. Acconeer AB (“Acconeer”) will not be responsible for damages of any nature resulting from the use or reliance upon the information contained herein. Acconeer makes no warranties, expressed or implied, of merchantability or fitness for a particular purpose or course of performance or usage of trade. Therefore, it is the user’s responsibility to thoroughly test the product in their particular application to determine its performance, efficacy and safety. Users should obtain the latest relevant information before placing orders.

Unless Acconeer has explicitly designated an individual Acconeer product as meeting the requirement of a particular industry standard, Acconeer is not responsible for any failure to meet such industry standard requirements.

Unless explicitly stated herein this document Acconeer has not performed any regulatory conformity test. It is the user’s responsibility to assure that necessary regulatory conditions are met and approvals have been obtained when using the product. Regardless of whether the product has passed any conformity test, this document does not constitute any regulatory approval of the user’s product or application using Acconeer’s product.

Nothing contained herein is to be considered as permission or a recommendation to infringe any patent or any other intellectual property right. No license, express or implied, to any intellectual property right is granted by Acconeer herein.

Acconeer reserves the right to at any time correct, change, amend, enhance, modify, and improve this document and/or Acconeer products without notice.

This document supersedes and replaces all information supplied prior to the publication hereof.