

README__nordic.md

The software has been tested against Nordic SDK 15.3.0 and GCC ARM Embedded 9-2020-q2-update using Ubuntu 20.04.

To view release notes use the following link: <https://developer.acconeer.com/sw-release-notes/>

Hardware requirements

XM122 & JLink Base

Install J-Link Software and Documentation pack

Install J-Link Software and Documentation pack from <https://www.segger.com/downloads/jlink/>

Install Nordic SDK

Download sdk from <https://www.nordicsemi.com/Software-and-Tools/Software/nRF5-SDK> and extract in a folder, e.g. “~/bin/”

Export the location of the SDK:

```
$ export NRF_SDK_ROOT=~/bin/nRF5_SDK_15.3.0_59ac345/
```

Install nRF5 Command Line Tools

Download the command line tools from <https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF5-Command-Line-Tools>

Extract the archive to a folder, e.g. “~/Downloads/nRF-Command-Line-Tools_10_5_0_Linux-amd64”

Install the .deb file:

```
$ sudo dpkg -i ~/Downloads/nRF-Command-Line-Tools_10_5_0_Linux-amd64/nRF-Command-Line-Tools_10_5_0_Linux-amd64.deb
```

Install Python packages *nrfutil*

nrfutil is needed to build DFU packages compatible with the XM122 bootloader.

```
$ pip3 install nrfutil
```

Note: If pip complains about missing versions, do a pip upgrade first:

```
$ pip3 install --upgrade pip
```

Install ARM Compiler

Download GCC ARM Embedded 9-2020-q2-update from <https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads> Extract the archive into a folder, e.g. `~/compilers/`

Export the location of the compiler:

```
$ export GNU_INSTALL_ROOT=~/compilers/gcc-arm-none-eabi-9-2020-q2-update/bin/
```

Install the Acconeer Software

Download the Acconeer zip file needed to run the Nordic device. Extract it into a folder, e.g. `~/src/`

Build the Software

There exists several build targets. Get the supported build and flash targets using:

```
$ cd ~/src/xm122_single_thread
$ make help
```

Build all targets using:

```
$ make
```

Run the Software

Connect XM122 via SWD to a JLink Base, then flash using:

```
$ make flash_<target>
$ make flash_softdevice
```

See the Logs

Logs are written both to the UART and over RTT. To see RTT logs use:

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

And then in another terminal:

```
$ tail -f log.txt
```

In order to see UART logs connect a USB to UART converter, e.g. TTL-232RG-VREG1V8-WE to the Tx Pin on the XM122 (TestPoint P1). Then start a terminal program with:

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

ETM tracing

Apart from debugging via SWD, XM122 also has trace pins routed to XB122 for supporting instruction tracing. There is support for both the Segger Ozone debugger as well as Lauterbach Trace32.

Segger Ozone

A project file for Seggers Ozone debugger is delivered with the Acconeer SW. Install Ozone and open the project file, xm122.jdebug. If needed change the application that is to be instrumented:

```
File.Open("${ProjectDir}/_build/detector_distance.out");
```

Lauterbach Trace32

Install Trace32 by downloading the latest release from https://www.lauterbach.com/download_request.html?ctyp. Extract and install Trace32:

```
tar xf TRACE32_R_2019_09_000114442.tar.xz
./setup_linux.sh ~/t32
```

Connect the debugger to the DUT and the host and start the Trace32 application with:

```
~/t32/bin/pc_linux64/t32marm-qt -c ~/t32/config_usb.t32 &
```

An example Trace32 trace script for XM122 is available in sw-main/scripts/trace32/nrf52840-acconeer_xm122_trace.cmm. Type the following command in the command window in Trace32 to start debugging XM122:

```
do sw-main/scripts/trace32/nrf52840-acconeer_xm122_trace.cmm
```

Please note that the trace script will not flash the application, only load the symbols. Flashing needs to be done prior to starting debugging.