

0. Set-up virtual machine
 - a) Download Ubuntu 20.04 LTS
 - a) https://releases.ubuntu.com/20.04.4/?_ga=2.48235494.1582537848.1657801855-274940221.1657106669
 - b) Run Virtualbox
 - a) Create Virtual machine, add Ubuntu iso as optical disk and run
 - b) Install Ubuntu 22.04 LTS on virtual box
 1. Minimal installation, no third-party SW, update while installation
 2. Created user "rt", with password q1w2e3r4
 3. Install AppImageLauncher (see: <https://github.com/TheAssassin/AppImageLauncher>)
 1. `sudo add-apt-repository ppa:appimagelauncher-team/stable`
 2. `sudo apt update`
 3. `sudo apt install appimagelauncher`
 4. `sudo apt install gcc make perl` (required by Guest Additions))
 5. Install Guest Additions
 6. Restart Ubuntu to load recompiled modules
1. Install the required tools
 - a) Check indications at:
 - a) https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/introduction.html
 1. Version 2.0.99 at the time of the writing of this doc
 2. Overview of toolchain, links to documentations, etc.
 - b) I tried the "Getting started" section and follow the "Installing automatically" instructions. Did not work. VSCode could not find the toolchain. Cmake and West missing, ...
 - c) Next I followed "Getting started" → "Install manually"
 1. Required tools
 2. Steps 1,2,3 OK. Did not install GN (no support to Matter is needed)
2. Install West
 - a) If it fails complaining about local config file not existing (happened with west 0.13.1) the fix is install previous version:
 1. `pip3 uninstall west`
 2. `pip3 install --user west==0.12.0`
 3. Then follow the indications to update path and test west
3. Get nRF Connect SDK code
 - a) Create folder \$HOME/ncs
 - b) move to folder \$HOME/ncs
 - c) `west init -m https://github.com/nrfconnect/sdk-nrf --mr v2.0.1`
 - a) To check out the latest version instead of a specific one
`west init -m https://github.com/nrfconnect/sdk-nrf --mr main`

- d) west update
- e) west zephyr-export
- 4. Install additional python dependencies
 - a) Ok. Just follow instructions
- 5. Install toolchain
 - a) Ok. Just follow instructions
- 6. nRF Connect for VSCode
 - a) Use Ubuntu Software app, look for “code” and install
 - b) Run VSCode
 - c) Then install the “nRF Connect Extension pack “ (it is a bundle of several packages)
 - d) In the quick set-up set the paths to SDK and Toolchain to the default values. Ignore the error regarding the nrfjprog. The issue is solved in step 8, below (Install Jlink SW)
- 7. Command line environment
 - a) Check instructions instructions to set env variables
 - b) install nRF command line tools
 - a) Download .deb from:
<https://www.nordicsemi.com/Products/Development-tools/nRF-Command-Line-Tools/Download?lang=en#infotabs>
 - b) sudo apt install nrf-....
- 8. **Now it is necessary to install the J-Link software (Not in the Nordic instructions. If it is not installed VSCode complains of not finding the nrfjprog, despite it being installed)**
 - 1. Download the rpm/deb package that matches your system from:
 - 1. <https://www.segger.com/downloads/jlink/>
 - 2. Accept the license agreement
 - 3. Wait for the download to complete
 - 4. Move to the folder in which you have downloaded the package
 - 5. Install the package using the package manager of your system
 - 1. e.g. “sudo apt install ./Jlink_Linux_Vxxx.deb”, in which “xxx” matches the version and architecture