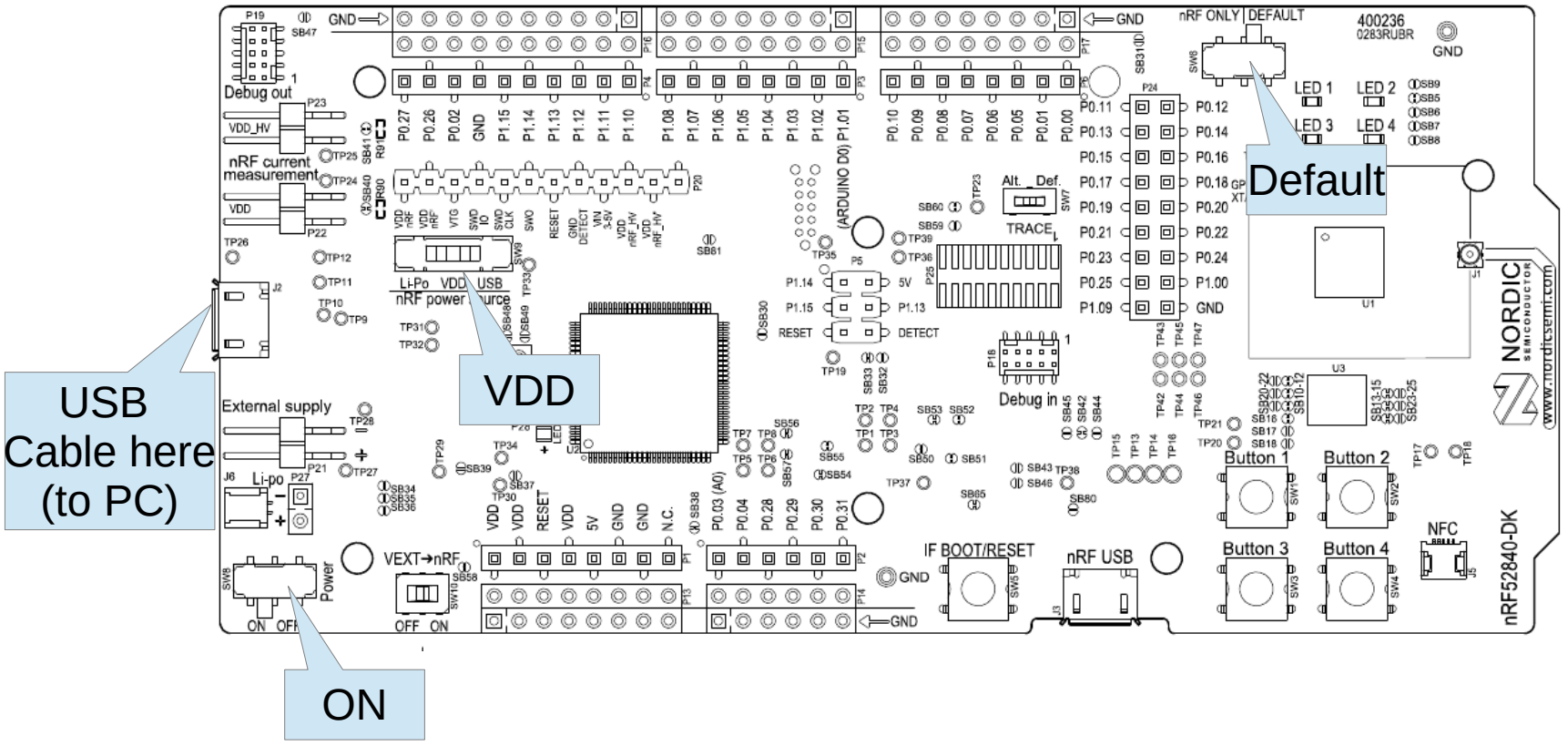
Creating an application with the nRF SDK

Paulo Pedreiras
SOTR
2022/10

Preliminaries

- Connect the nRF DK to the PC via the USB cable
- Check that the switches are on the correct position
  - Power – ON (left)
  - NRF Power Source – VDD (middle)
  - NRF Only – Default (right)
- If using the VM, pass the USB to it
  - On the VM, Devices → USB → Segger J-Link

Preliminaries



Starting from a Nordic Sample

- Welcome → Create a new application
- Check the SDK, toolchain, …
  - Everything should be fine, if the isnbtallation is OK
  - Ignore the “gn not found” message. Only necessary for Matter aplications
- Choose a location
- In the templates select a suitable template
  - E.g. “Hello World”
- Assign a name
- Hit Cretae Applicaion button

Starting from an existing application

- Copy project folder to the intended path. It must include
  - CmakeLists.txt
  - Prj.conf
  - Eventually DTS Overlay files
  - Source code (e.g. in ./src folder)
  - Remove any build folder, e.g. “build”, or “build_nrf52840dk_nrf52840”. This step is not strictly necessary but can save you from trouble
- Go to VScode
  - Switch to nRF Connect view
  - Application → “+” / Add folder as application
  - Select the folder that has the project files

Add a build configuration

- The build configuration identifies the target HW and allows setting several parameters
  - Applications → Action button after application entry
  - Select “nrf52840dk_nrf52840” (in our case)
  - Confirm the name of the “prj.conf” file
  - Set any Cmake arguments
  - Set the build folder name (if not pretend to use the default one)
  - Click on “Build Configuration”

Linking the app and the HW and building + flashing

- On the Applications window, open the “App_Name” drop-down menu to show the “build” entry
  - Hit the button “Link Build Configuration and Device”
  - Any other apps linked must be removed
- Then you can hit button “Flash all linked devices” icon after “Applications” entry
  - There are several other options
    - E.g. Actions → Flash

Editing the files and configs and utilities

- The active application has a matching window just below the “Applications” window that allows to navigate through the files
  - Source files (e.g. user code)
  - Input files (e.g. CmakeLists.txt, prj.conf, …)
- There are a few windows below the code that are useful
  - Problems: report issues in user code, config files, …
  - Terminal (compilation results)
  - NRF Terminal – an integrated UART terminal, useful for debugging

The UART interface

- Default terminal configuration
  - 115200 bn1 rtscts:off
- Can use the internal terminal or any other one
  - E.g. minicom, putty, ...