Application note AN250502

Use of the ISP2454-LX Development Kit · Application Note

ISP2454-LX

Introduction

Scope

This document gives details on hardware and software using and testing the Insight SiP Bluetooth Low Energy module ISP2454-LX.

Contents

1. Recommended Documentation

The following documents are required to understand the complete setup and programming methods:

- The nRF54l15 datasheets
- The introduction to nRF Connect SDK
- Developing with nRF54L
- The ISP2454-LX informations

2. Hardware content

2.1. Global Description

The Evaluation Board hardware includes:

- An Interface Board named ISP130603 with integrated J-Link OB JTAG/SWD Emulator
- One ISP2454-LX Test Board

2.2. About Insight SiP BT 6.0 Module

The ISP2454 module is based on nRF54L15 Nordic Semiconductor 2.4GHz wireless System on Chip (SoC) integrating a 2.4GHz transceiver, an ARM Cortex -M33 with TrustZone technology, RRAM and RAM memory adding digital peripherals.

It can support BLE, Mesh and a range of proprietary 2.4GHz protocols. Fully qualified stacks for nRF54L15 are freely for operating in Central role, Peripheral role, Scanning and Advertising.

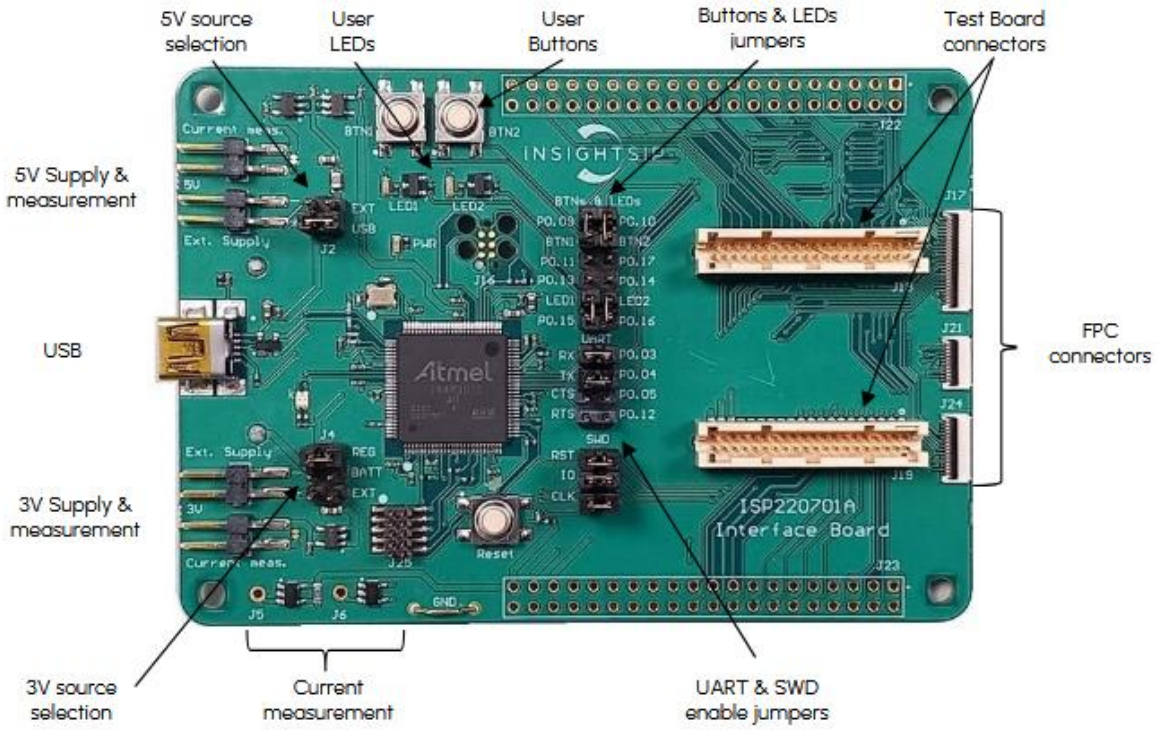
The Cyber Resilience Act (CRA) is embodied in the EU regulation 2024/2847. ISP2454 rely on microprocessors that have been designed with Platform Security Architecture (PSA) certification in mind. It is certified to PSA level 3. This level of platform security reduces the risk of vulnerabilities occurring.

The module measures 8x8x1 mm, the module integrates all required decoupling and load capacitors, DC-DC converter, 32Mz crystal, 32.768KHz crystal, RF matching circuit and internal antenna.

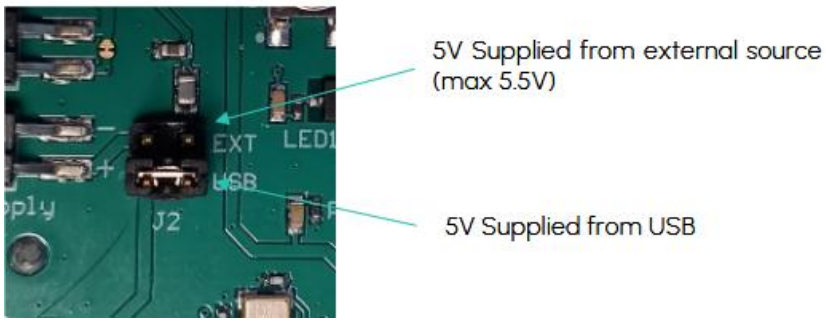
The module has ultra-low power consumption and advanced power management.

2.3. ISP130603G Interface Board

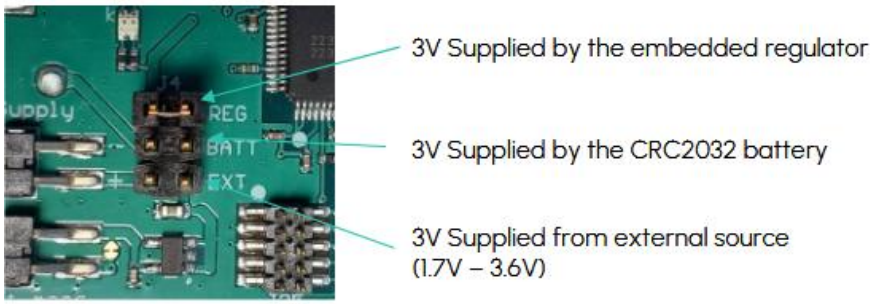
Board overview:



- J2 jumpers selects the 5V power supply source:



- J4 jumpers select the 3V power supply source:



- SWD & UART jumpers can connect or disconnect Debug and UART lines from the test board. It is recommended when powering up the board with a CR2032 battery or external 3V. With such supply sources the debugger will not be supplied

- BTNs & LEDs jumpers give the possibility to connect up 2 buttons & 2 LEDs to the test board:

2.4. ISP2454-LX-TB Test Board

Board dimensions are 47 x 42 mm². It includes:

- ISP2454-LX BLE module
- 2 2x20 pins connectors at the bottom for Interface Board connection
- 2x5 pin header for programming and debug when connected to a Nordic Evaluation Board
- JTAG footprint for programming and debug when connected to a Segger J-Link
- Printed NFC antenna

2.5. Hardware Setup

- Connect the USB cable from the Interface Board ISP130603 to your computer.
- Plug the BLE Test Board to the Interface Boad using the 2x20 connectors.

2.6. Pin mapping with the interface board

To use correctly the GPIO of the ISP2454 test board with the ISP130603 interface board, the following tab shows the matching between the GPIO pins of the interface board and the GPIO pins of the ISP2454 test board:

ISP130603 pins	ISP2454-TB pins
P02	P0.02
P03	P0.03
P04	P0.04
P05	-
P06	-
P07	-
P08	-
P09	-
P10	P2.00
P11	P2.01
P12	P2.02
P13	P2.03
P14	P2.04
P15	P2.05
P16	P2.06
P17	P2.07
P18	P2.08
P19	P2.09
P20	P2.10
P21	-
P22	-
P23	-
P24	-
P25	-
P26	-
P27	-
P28	-

ISP130603 pins	ISP2454-TB pins
P29	-
P30	-
P31	-
P32	-
P33	-
P34	P1.02
P35	P1.03
P36	P1.04
P37	P1.05
P38	P1.06
P39	P1.07
P40	P1.08
P41	P1.09
P42	P1.10
P43	P1.11
P44	P1.12
P45	P1.13
P46	P1.14
P47	P1.15
P48	-
P49	-
P50	-
P51	-
P52	-
P53	-
P54	-
P55	-

3. Software Content

3.1. Zephyr RTOS

The ISP2454 uses the nRF Connect SDK from Nordic Semiconductor running zephyr RTOS.
Zephyr RTOS is a small, scalable, real-time operating system (RTOS) designed for resource-constrained embedded systems.

The main key features are described below:

- Preemptive real-time kernel with priority-based scheduling
- Multi-architecture support
- Integrated security features
- Built-in connectivity stacks
- Multithreading support with synchronization primitives
- Power management framework for ultra-low power applications
- Modern development tools: CMake, West, native SDK.
- Highly modular and configurable architecture with Device Tree and Kconfig files

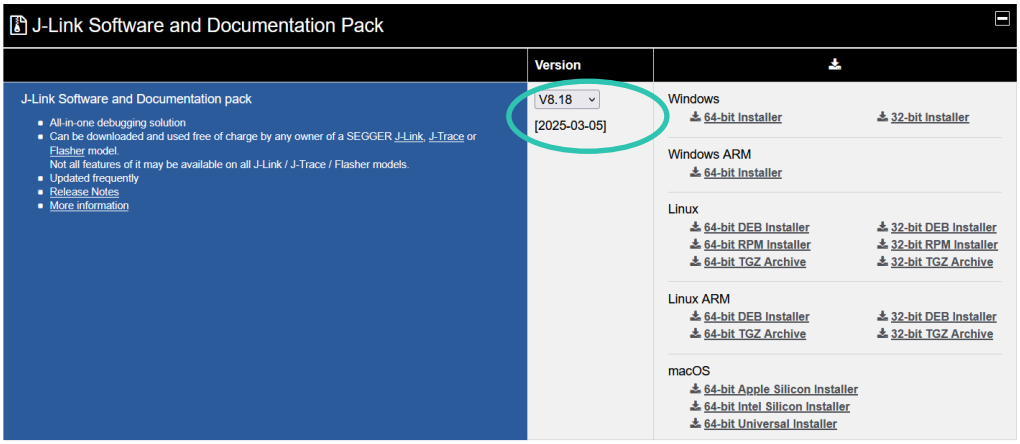
For the last point, many things can be done in the two files:

- The prj.conf file to enable or disable services (example: CONFIG_LOG_BACKEND_RTT=y allowing to enable the LOG on RTT)
- And the creation of an overlay file in the Device Tree for hardware configuration

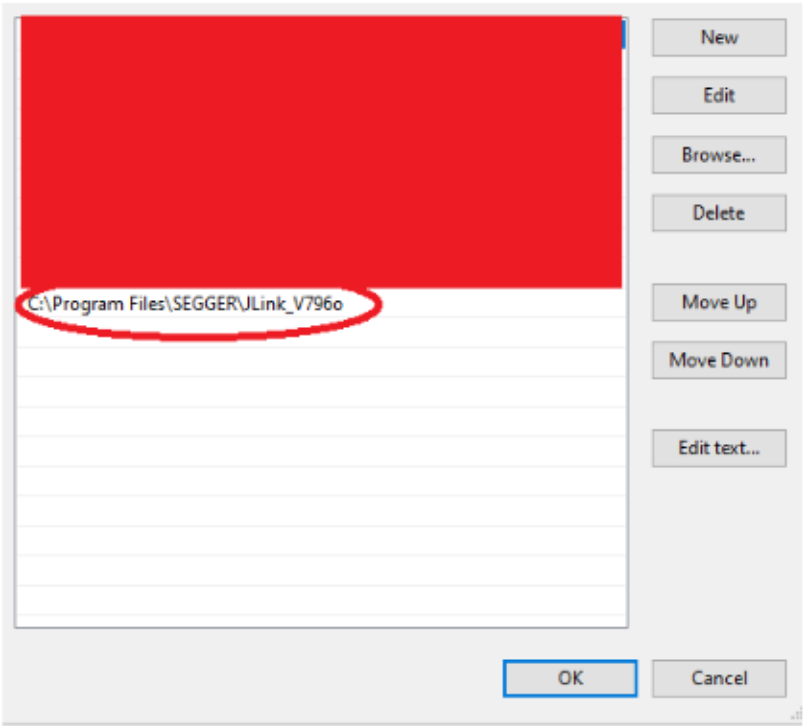
3.2. PC Software Installation

3.2.1. JLink installation

- Flashing & debugging the Insight SiP module is done using a J-Link probe
- Download and install the latest version of J-Link software at:
- https://www.segger.com/downloads/jlink



- Once the J-Link software has been installed set a Path in the Environment variables to the location of JLink.exe for the latest version:

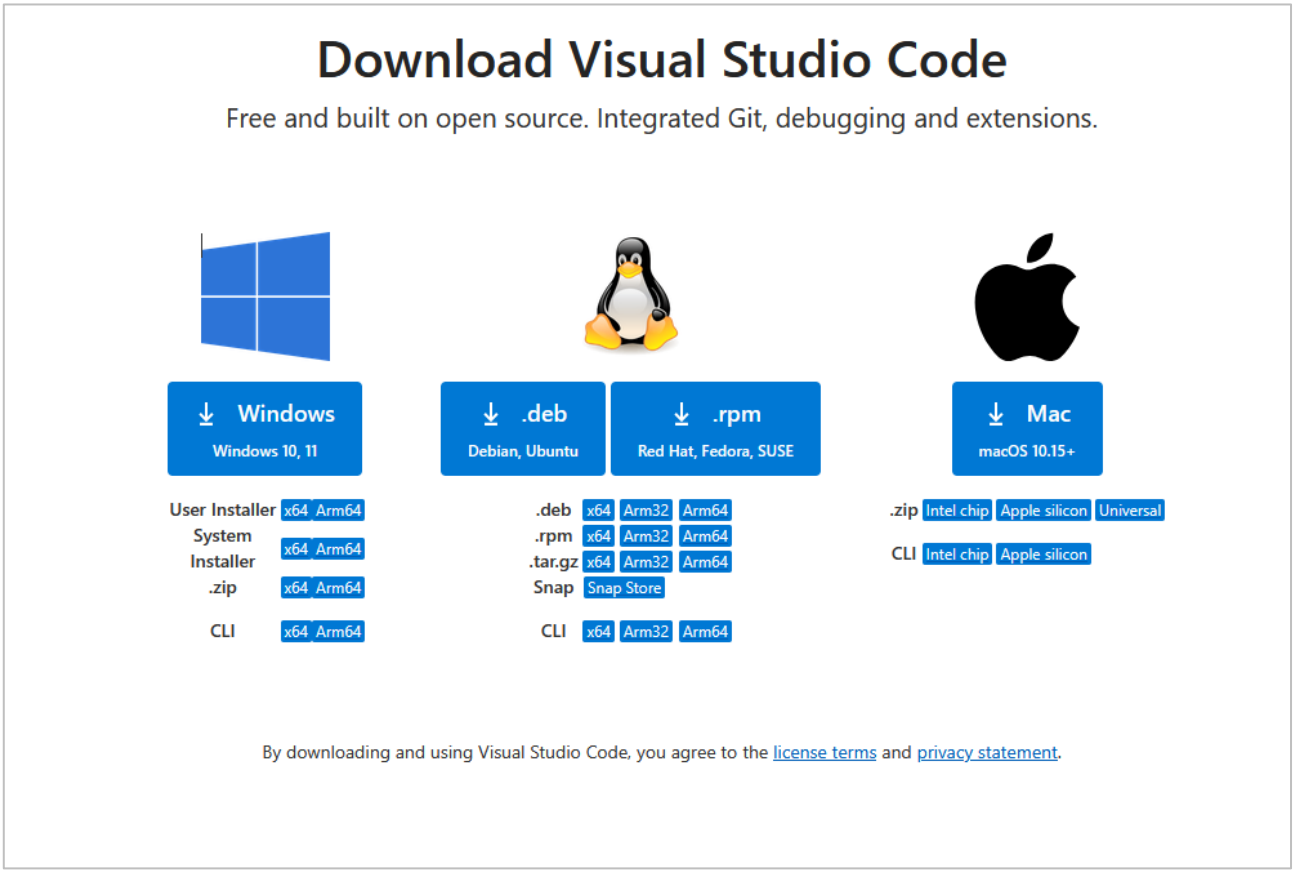


3.2.2. nRF Util installation

- Install nRF Util from Nordic website: https://www.nordicsemi.com/Products/Development-tools/nRF-Util
- Copy the executable in C:/ncs
- As same that the previously step, add path to nRF Util to Windows Environment Variables
- Open Windows Power Shell
- Type the command "nrfutil install device"
- Type the command "nrfutil install toolchain-manager"

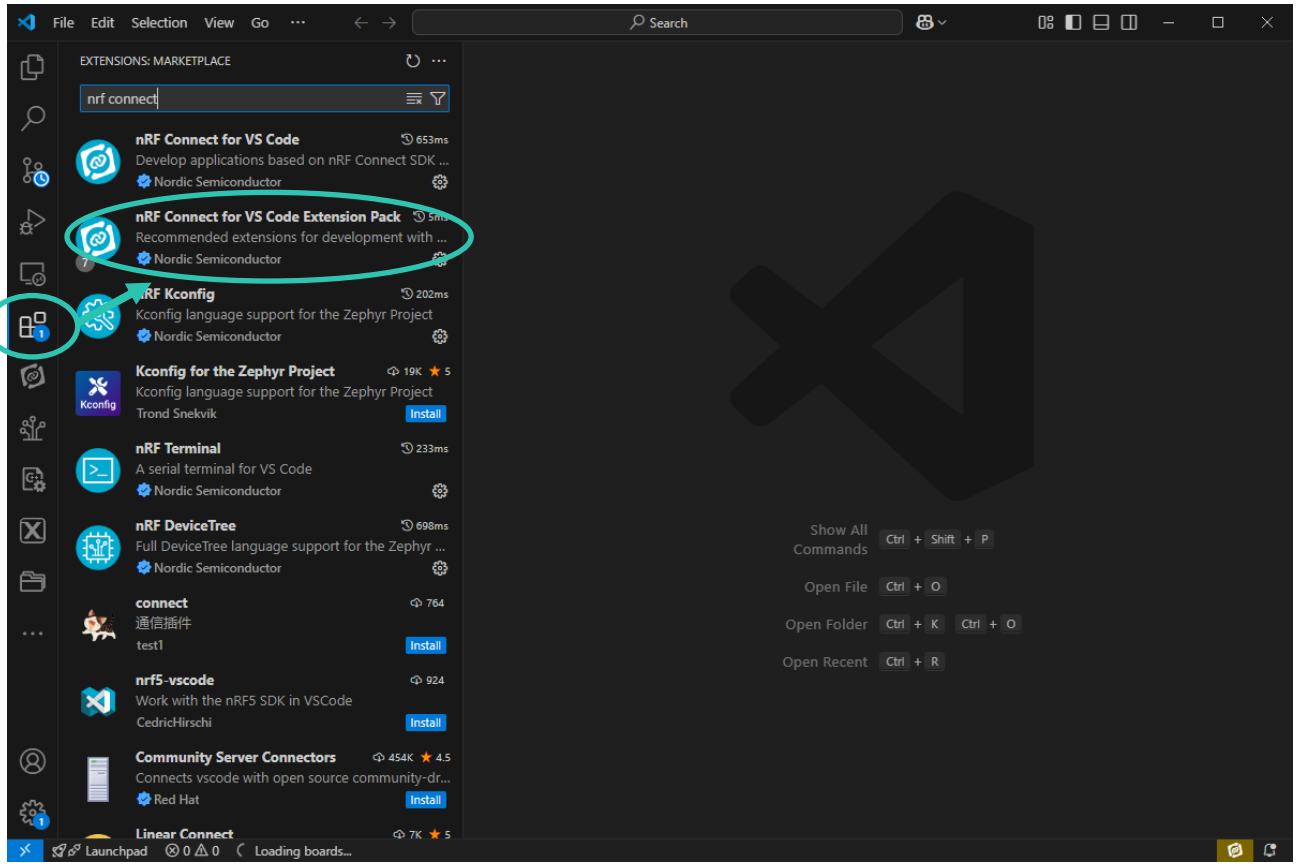
3.2.3. Visual Studio installation

- Download the latest version for visual Studio Code at: https://code.visualstudio.com/download



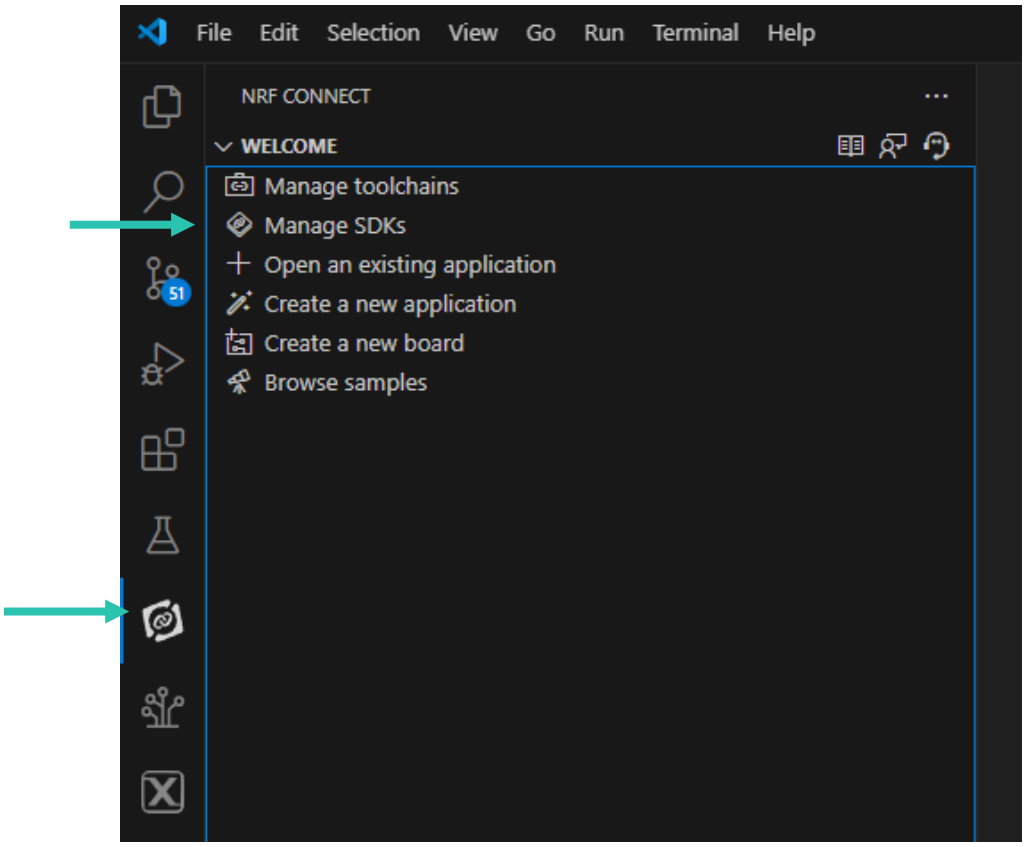
3.2.4. nRF Connect for VSCode installation

- Start VS Code, in extension menu search for nRF Connect and install the extension pack:

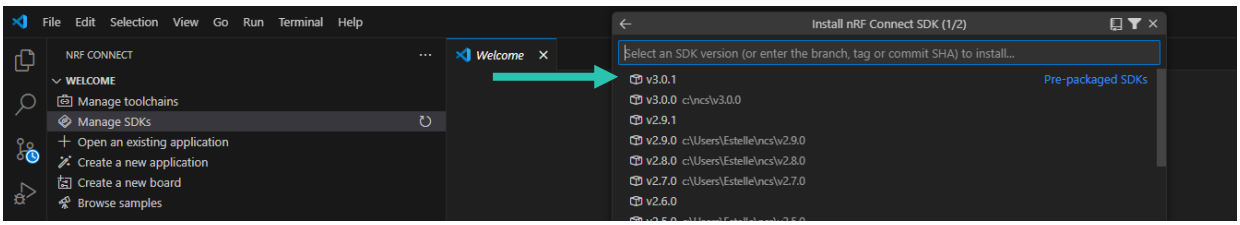


3.2.5. Nordic SDK and toolchain Installation

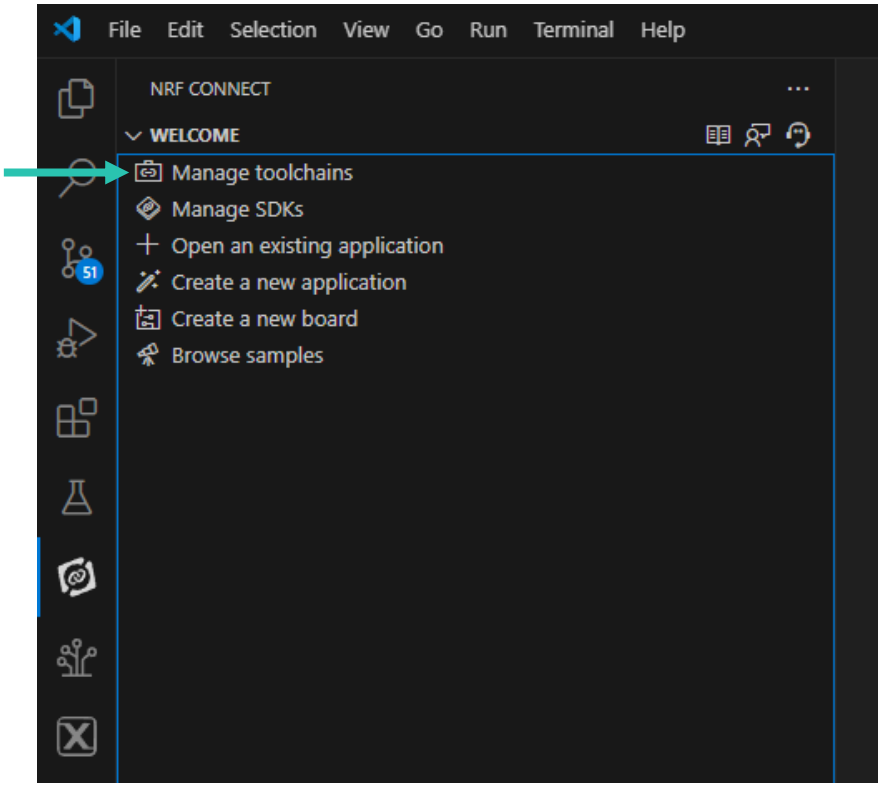
- Open VS Code
- Click on the nRF Connect icon and in the nRF Connect area click on "Manage SDK":



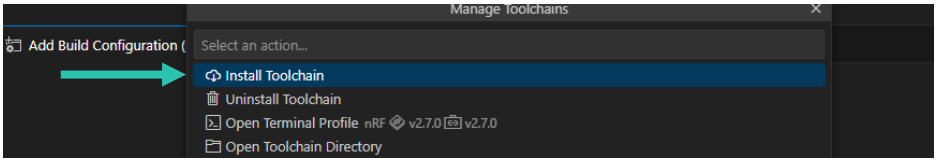
- In the top bar, click on "Install SDK" and select the latest SDK version:



- When SDK is installed, click on "Manage Toolchain":



- In the top bar, click on "Install Toolchain" and select the version that matching with the SDK version installed (for example 3.0.0 with a 3.0.0 SDK version)



3.3. Smartphone Software installation

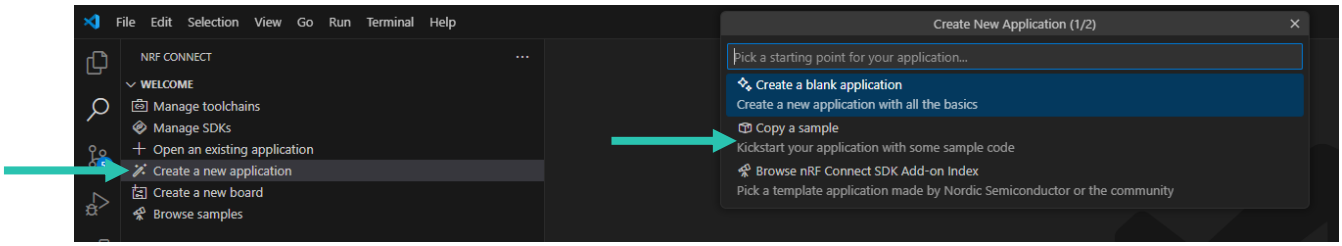
- Install the application "nRF Connect" available on App Store or on Google Play.

3.4. nRF Connect for Desktop

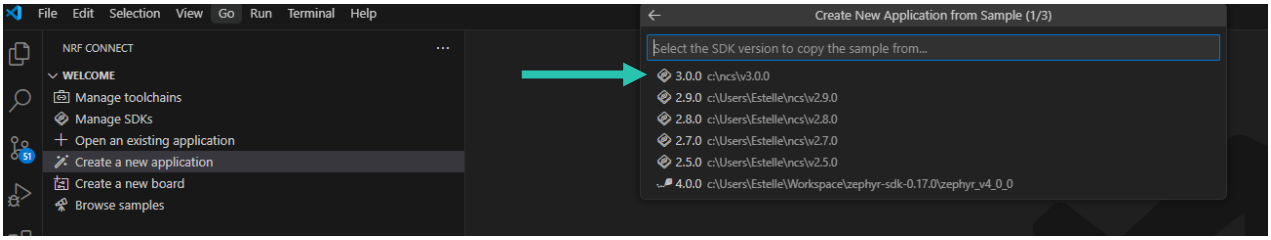
- Install the application "nRF Connect" for Desktop here: https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-Desktop
- Download the "Direct Test Mode" application from "nRF Connect for Desktop"

4. HR Peripheral Example

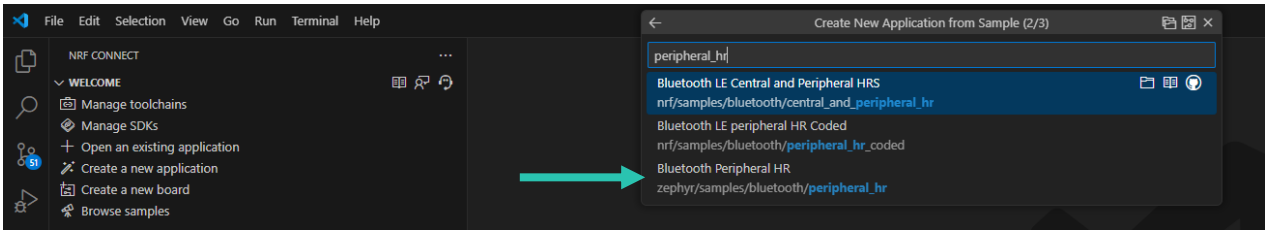
- Start VS Code
- In nRF Connect menu, ***create a new application*** and in search bar select ***Copy a sample***



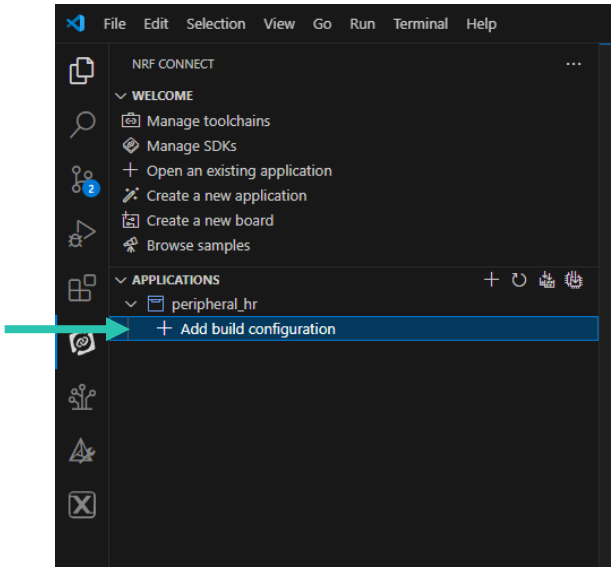
- Select the nRF Connect SDK installed (v3.0.0 minimum):



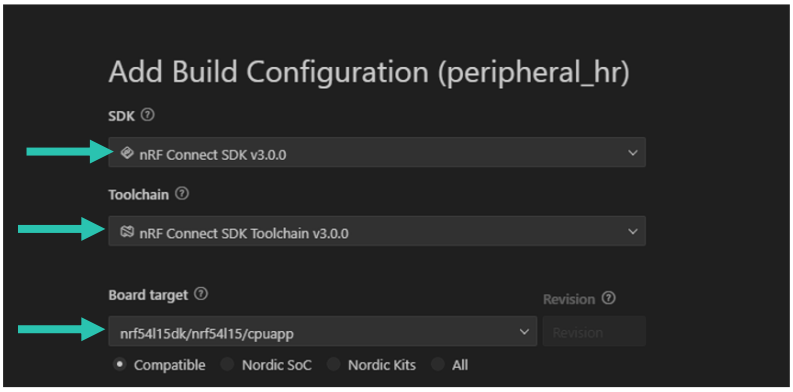
- Load the ***Bluetooth Peripheral HR*** example and press Enter:



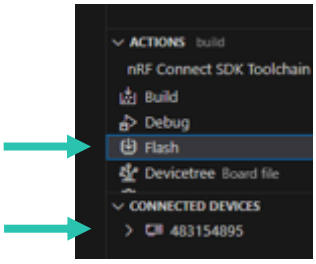
- In APPLICATIONS section of the project, click on ***Add build configuration***:



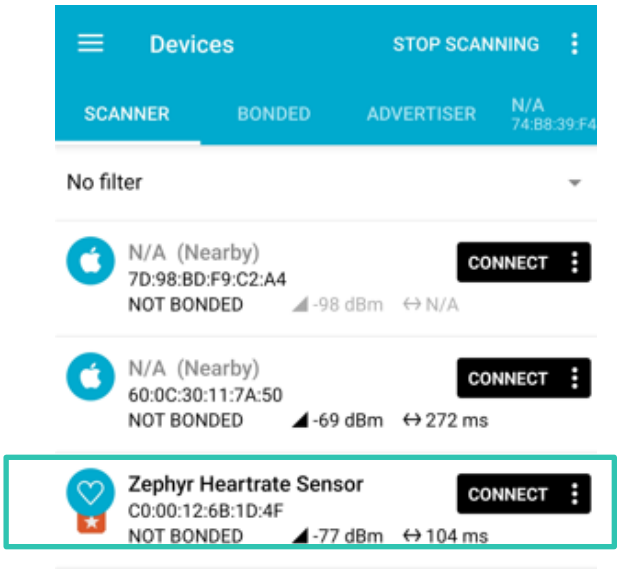
- A build configuration window opens on VS Code
- In this window, select the SDK and Toolchain version
- Choose the board target named "nrf54l15dk/nrf54l15/cpuapp"



- Click on Build
- In the ACTIONS section, after build successful, make sure the device is connected and flash it (proceed to a recover before flashing):

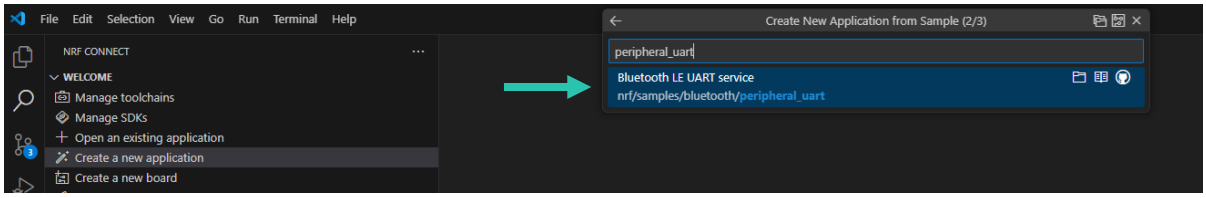


- Using "nRF Connect App" on your smartphone, scan for the devices and search the "Zephyr Heartrate Sensor":

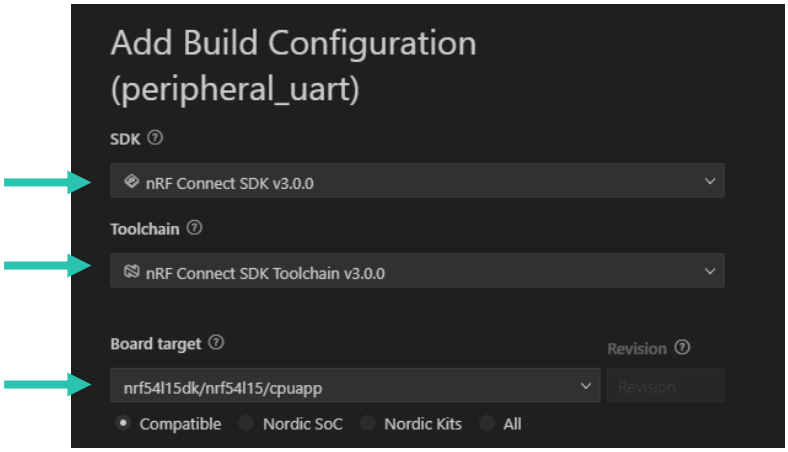


5. Peripheral UART Example

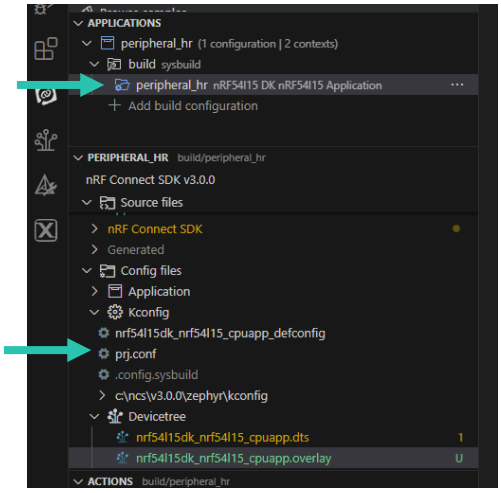
- As the same previously steps, create a new application by coping a sample
- Select the ***Bluetooth LE UART service*** application example



- As the same previously steps, add a build configuration with the last SDK, last toolchain and the board target named "nrf54l15dk/nrf54l15/cpuapp":



- Open the prj.conf file in the Kconfig area in your build project:

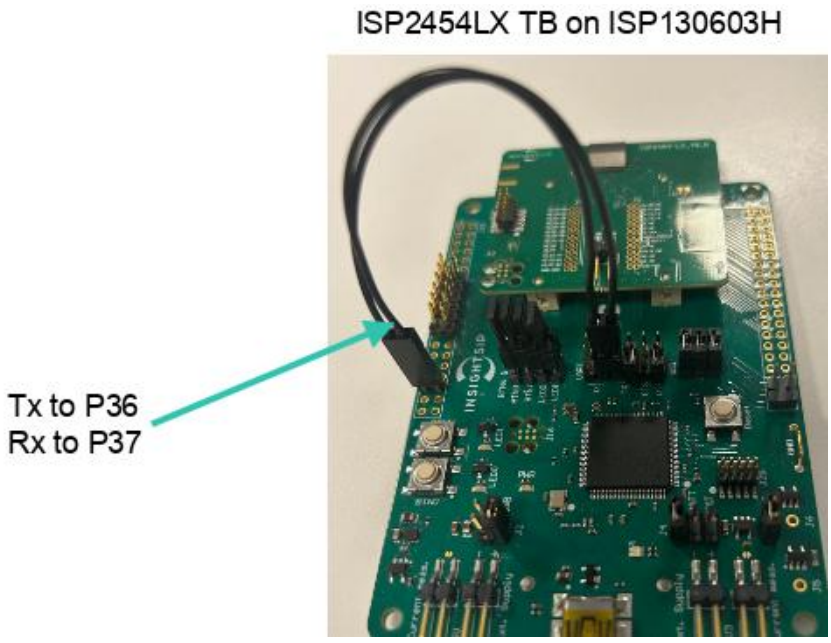


- To see the logs on the UART console, you need to enable the logs backend on UART and disabled the RTT* in the prj.conf file:

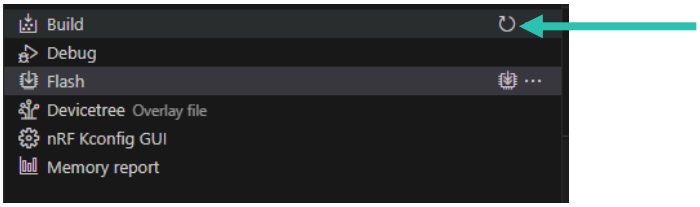
```
# Config logger
CONFIG_LOG=y
CONFIG_USE_SEGGER_RTT=n
CONFIG_LOG_BACKEND_RTT=n
CONFIG_LOG_BACKEND_UART=y
CONFIG_LOG_PRINTK=y

CONFIG_ASSERT=y
```

**RTT and UART can run in parallel on the nRF54l15 but there is a limitation of the software: the logger can generally use only one backend at a time for the same messages, except to write a custom solution.*

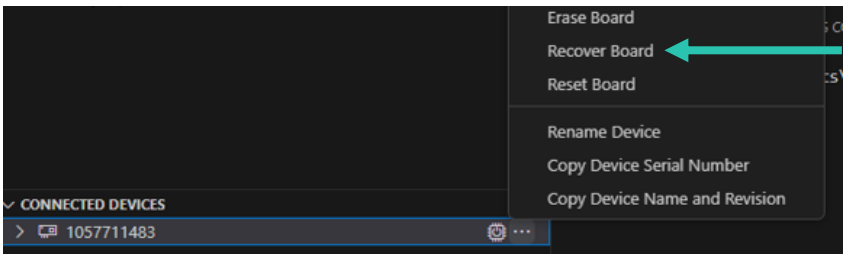
- Otherwise, if you want to see the logs over RTT, enable segger RTT and the log backend on RTT and disable the backend on UART
- Initially, the UART uses to log is the UART20 which uses the peripheral P1. TX is on the P1.04 GPIO pin and RX is on P1.05 GPIO pin
- If you want to use the initial UART, you need to connect the TX and RX pin of the interface boards to the correct GPIO pin:
  -Connect TX on pin P36 on the interface board
  - Connect RX to pin P37 on the interface board



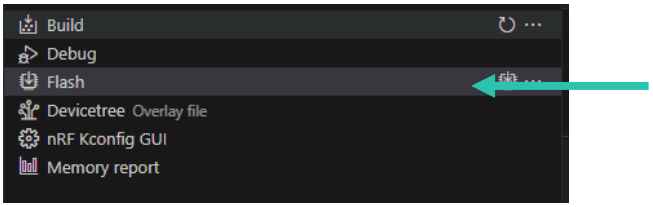
- If you want to modify the UART to log, you need to modify the overlay file to choose the nus-uart node and to map the pins that you want to use. Don't forget to connect the UART pins that match with the correct pins of the interface board (see the previous tab).
- Pristine build the project in the ACTION section:



- Recover the board before flashing:



- Then flash:

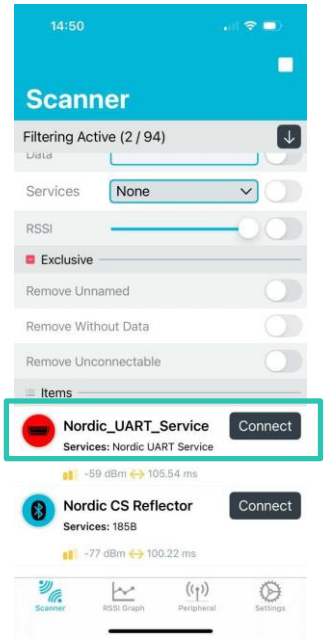


- Open a terminal emulator
- Connect the port COM at 115200 bauds
- Reset the development kit with the "Reset" bouton
- See the following logs on your terminal

```
*** Booting My Application v2.8.0-bc9e142c2395 ***
*** Using nRF Connect SDK v2.8.0-a2386bfc8401 ***
*** Using Zephyr OS v3.7.99-0bc3393fb112 ***
[00:00:00.003,423] <inf> fs_zms: 2 Sectors of 4096 bytes
[00:00:00.003,435] <inf> fs_zms: alloc wra: 0, f90
[00:00:00.003,443] <inf> fs_zms: data wra: 0, 10
[00:00:00.003,526] <inf> bt_sdc_hci_driver: SoftDevice Controller build revision:
                                            fe 2c f3 6a 7f 36 22 2e  a0 79 c0 40 be 2c 03 20 |.,.j.6". .y.@.,.
                                            40 c2 f3 32                                     |@..2
[00:00:00.005,513] <inf> bt_hci_core: HW Platform: Nordic Semiconductor (0x0002)
[00:00:00.005,529] <inf> bt_hci_core: HW Variant: nRF54Lx (0x0005)
[00:00:00.005,555] <inf> bt_hci_core: Firmware: Standard Bluetooth controller (0x00) Version 254.63788 Build 573996906
[00:00:00.005,878] <inf> bt_hci_core: No ID address. App must call settings_load()
[00:00:00.005,887] <inf> peripheral_uart: Bluetooth initialized
[00:00:00.006,295] <inf> bt_hci_core: Identity: DA:F8:C5:C5:CD:C9 (random)
[00:00:00.006,313] <inf> bt_hci_core: HCI: version 6.0 (0x0e) revision 0x304e, manufacturer 0x0059
[00:00:00.006,327] <inf> bt_hci_core: LMP: version 6.0 (0x0e) subver 0x304e
```

- Open the nRF Connect application on your smartphone

- Connect to the "Nordic_UART_Service":

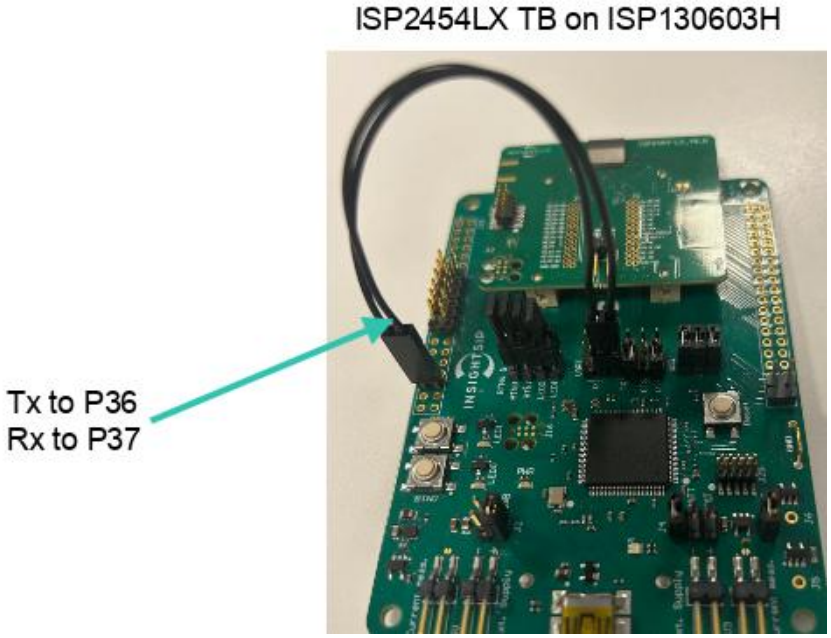


- See the log on emulator terminal that notifies that there is a device connected:

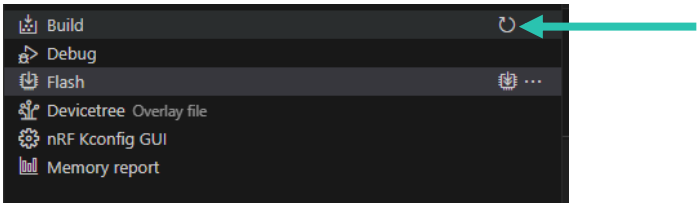
```
[00:01:58.800,078] <inf> peripheral_uart: Connected 54:9D:84:65:60:17 (random)
[00:01:59.276,439] <wrn> bt_l2cap: Ignoring data for unknown channel ID 0x003a
```

6. Direct Test Mode Example

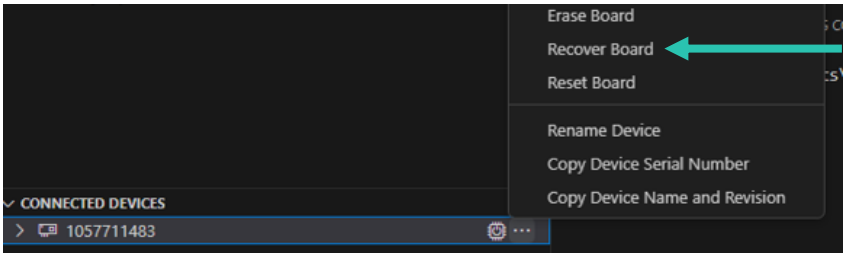
- As the same previously steps, create a new application by coping a sample
- Select the ***Direct Test Mode*** application example
- Initially, the UART uses for DTM is the UART20 which uses the peripheral P1. TX is on the P1.04 GPIO pin and RX is on P1.05 GPIO pin
- If you want to use the initial UART, you need to connect the TX and RX pin of the interface boards to the correct GPIO pin:
  - Connect TX on pin P36 on the interface board
  - Connect RX to pin P37 on the interface board



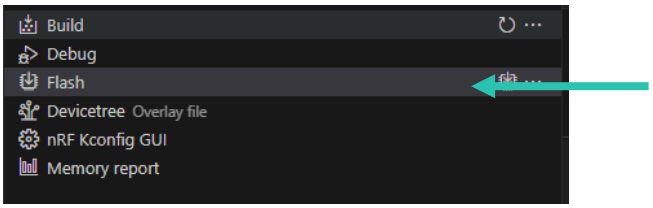
- If you want to modify the UART to use the dtm, you need to modify the overlay file to choose the dtm-uart node and to map the pins that you want to use. Don't forget to connect the UART pins that match with the correct pins of the interface board (see the previous tab).
- Pristine build the project in the ACTION section:



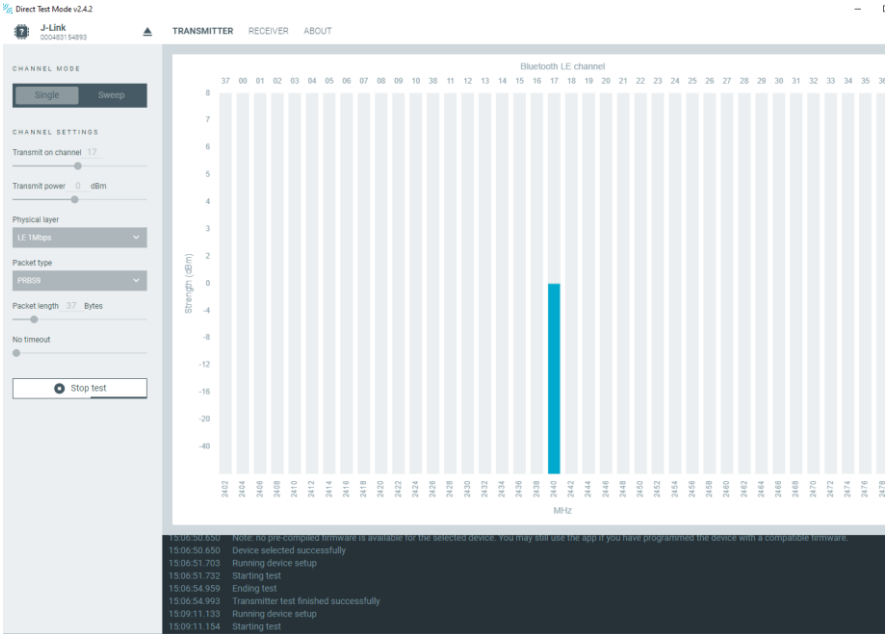
- Recover the board before flashing:



- Then flash:

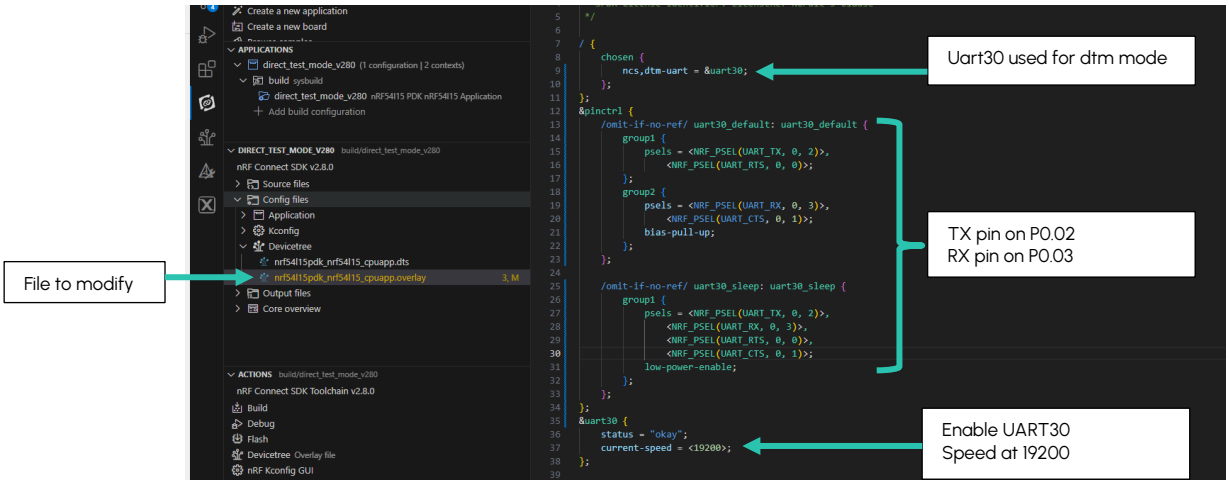


- Open the Direct Test Mode application in nRF Connect on your computer
- Connect to the JLink
- Then test the transmitter or receiver mode. For example, in transmitter mode on canal 17 with 0dBm power:



Annexes

- Example of modifying the overlay file:



- Hardware connection on ISP130603 interface board: