PTX2xxR RUL SDK and Evaluation Board Usage

Introduction

The Renesas Near Field Communication (NFC) transceiver uses the Software Development Kit (SDK) library – a software toolset that interacts with the PTX2xxR hardware for implementing NFC-based solutions. This application note focuses on series PTX2xxR RUL SDK usage with PTX200R/PTX205R variants and provides the following guidelines:

- Installing/setting up the NFC evaluation board (EVB) and using the SDK
- Adapting the latest SDK to work on the EVB
- Flashing the SDK and running the examples on the EVB
- Flashing the EVK bootloader to use the EVB with the Config Tool
- The necessary steps for testing the PTX200R hardware after a successful flashing firmware process.

Contents

1. Abbreviations and Terminology

Abbreviations	Terminology
NFC	Near Field Communication
HW	Hardware
SW	Software
FW	Firmware
EVB	Evaluation Board
SDK	Software Development Kit
RUL	Reader Universal Library
IDE	Integrated Development Environment
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
RX	Receive
TX	Transmit
GPIO	General Purpose Input Output
HAL	Hardware Abstraction Layer
MCU	Microcontroller Unit
API	Application Programming Interface
NDEF	NFC Data Exchange Format
MFCC	MIFARE Classic® Compatibility
NSC	NFC Soft Controller
I2C	Inter IC Communication
RFID	Radio Frequency Identification
POS	Point of Sales
IOT	Internet of Things
ATQA	Answer To reQuest code A
RATS	Request for Answer To Select

2. NFC Basics and Overview

The NFC (Near Field Communication) is a short-range wireless communication technology operating in 13.56MHz band of unlicensed spectrum that allows two devices to exchange data when they are brought close together, typically within a few centimeters.

It operates on the principles of electromagnetic induction and transaction mainly relies on a combination of RFID and NFC using radio frequencies to establish a connection between an NFC-enabled device and a tag. RFID enables the reader or writer to detect the other device within its range and NFC refers to the protocol standards developed by the NFC Forum that ensure the devices can communicate with each other.

2.1 Key Characteristics

- NFC controller design employs a split-stack solution where time-critical operations are running on the on-chip HW accelerator and the NFC logic in the host controller carries out applications. This helps in minimizing the demands on the host MCU and simplifies the software integration.
- Compatible with existing ISO/IEC 14443-A / MIFARE® and FeliCa contactless card and reader infrastructure, ISO18092 and ISO/IEC 15693 reader/writer mode
- NFC Forum certification compliant
- Optimal RX sensitivity
- Data exchange rates supported are 106 / 212 / 424 / 848 kbit/s, depending on tag types
- Fast connection: NFC establishes connections quickly
- Low power: Using features like Low Power Card Detection (LPCD) and Low Power Field Detection (LPFD), it consumes minimal power and makes it suitable for passive devices
- Reader Universal Library (RUL) SDK for easy integration into host controller

For more information, refer to the PTX205R Datasheet.

3. Software Architecture of the PTX2xxR RUL SDK

The software architecture of the RUL Stack is comprised of multiple layers where each layer contains one or more component(s). See Figure 1 for a high-level overview of the RUL software architecture.

For additional documentation, refer to the file at location `../DOCS/html/index.html`.

3.1 Layer and Component Description

- **Platform Abstraction Layer/HAL (Hardware Abstraction Layer) Component**
  - This layer implements access to hardware resources such as host-interfaces (for example, SPI, I^2C, UART), timers, GPIOs, interrupt handling, etc.
  - The RUL SDK includes HAL reference implementations for the Renesas RA MCU Family, Windows and Linux hardware platforms.
- **Core Component Layer/HIP (Host Interface Protocol) Component**
  - The HIP component implements the PTX2xxR/W NFC hardware specific transport mappings for SPI, I^2C and UART.
  - Low-level commands to download the FW image into the code memory, accessing command buffers, reading / writing from/to memory and registers.
- **NSC Core Stack**
  - Command Set Library where the PTX2xxR/W NFC hardware implements a dedicated on-chip controller which hosts a Firmware/FW image.
  - The FW interacts with the RUL Stack using a Command/Responses/Data/Notifications as message-types and implements the following functionalities:
    - NFC RF-Discovery/Polling-loop for NFC Forum-/EMV-/ISO-mode including collision-resolution and device activation and deactivation.
    - Power Management (Stand-by, Low Power Card Detection (LPCD))
- **Application Layer/RUL Reader Universal Library API**
  - The Reader Universal Library API (RUL) is the main SW interface towards the application. It implements different core functionalities for Reader/Card-Emulation, P2P and Wireless Charging used in sample applications for IoT and POS.

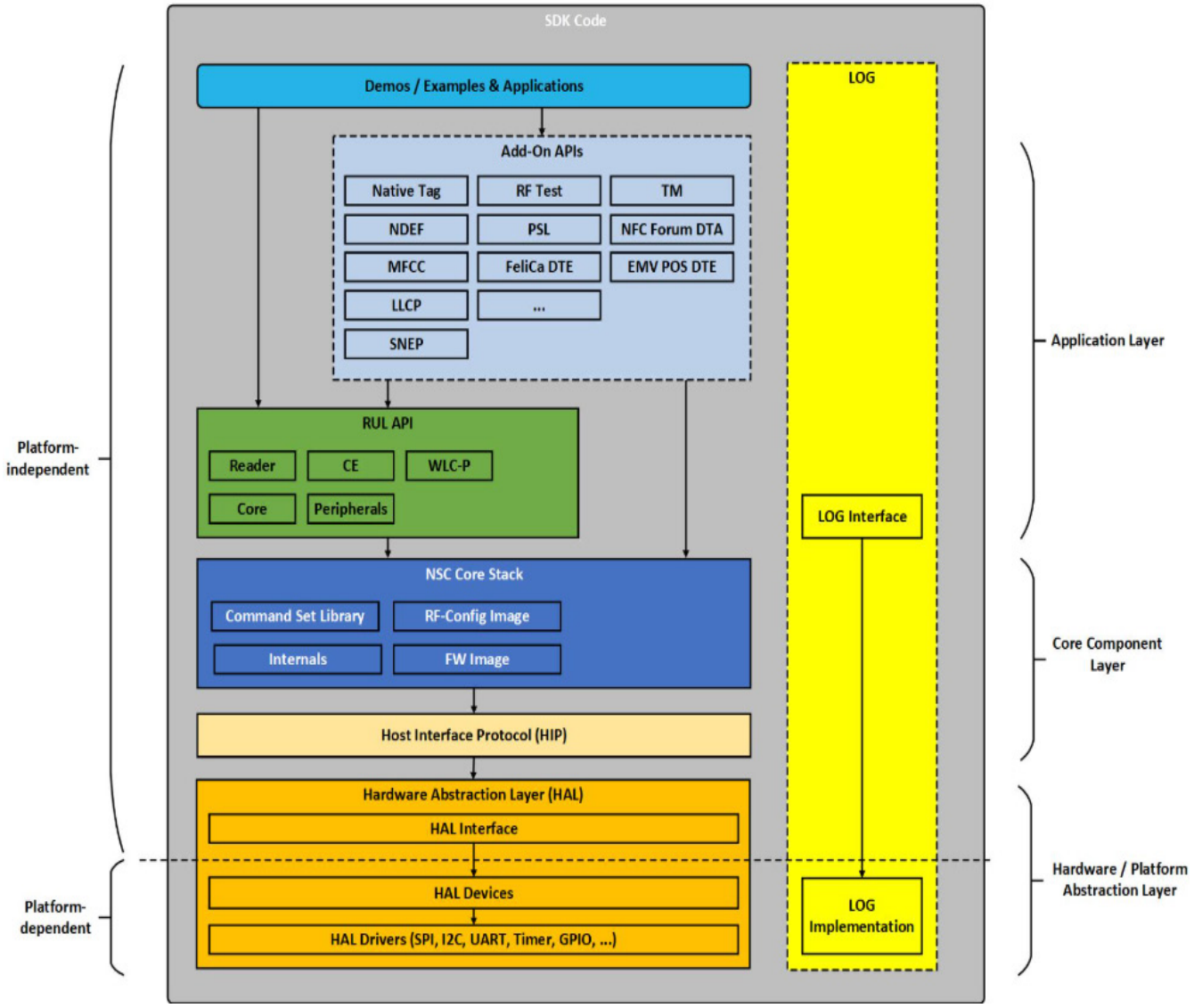


Figure 1. RUL API SW Architecture Overview

3.2 NFC Discovery Example

The PTX2xxR RUL stack handles command, response synchronously and asynchronous notifications, and other events showing the NFC discovery process (see Figure 2).

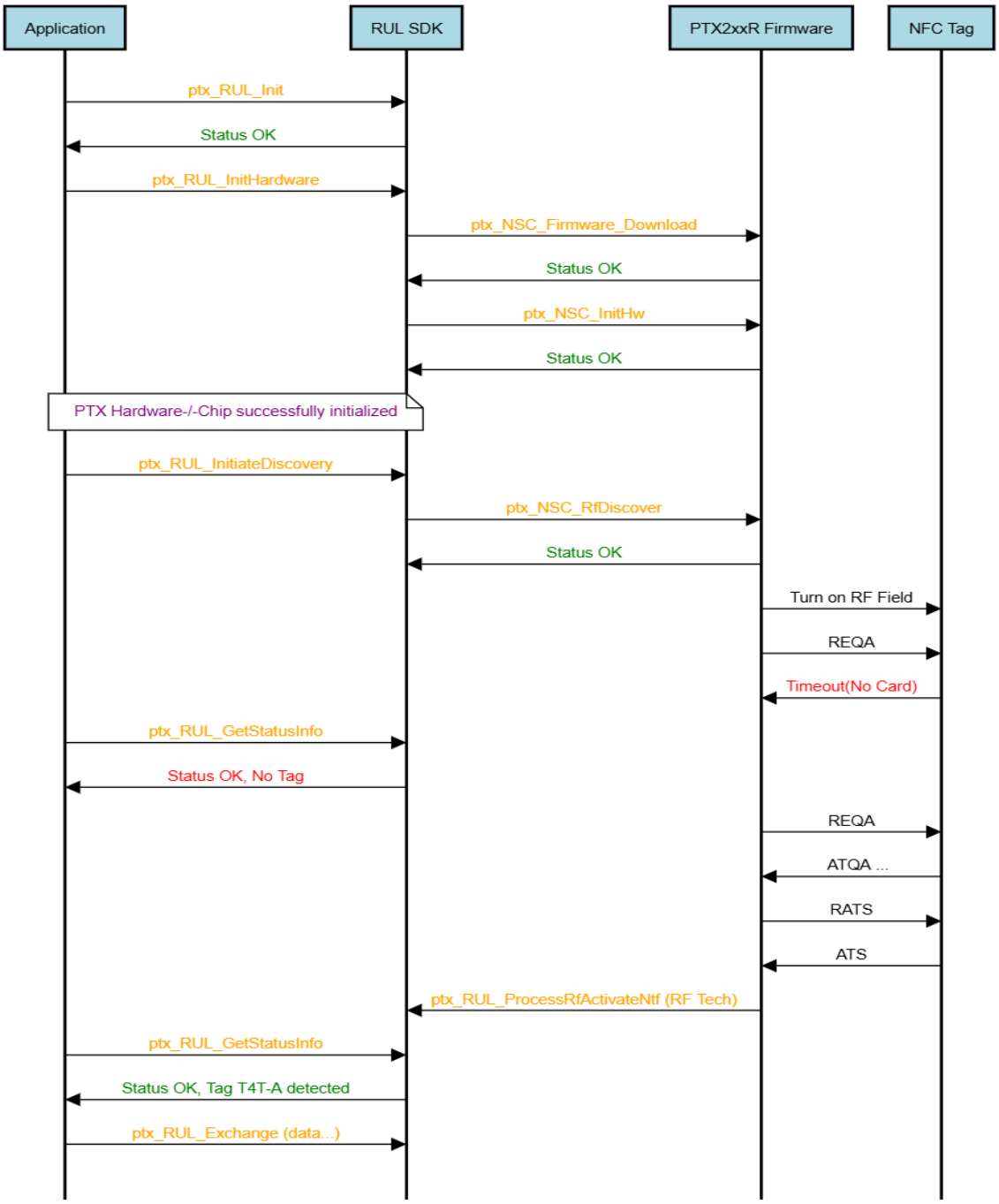


Figure 2. NFC Discovery

4. Software Installation for PTX2xxR

Access is required to the following software and hardware installation tools. Developers are encouraged to use either Windows, Linux or Mac OS supported on a PC. This section focuses on the primary development environment of Windows 10/11 64-bit only:

1. e2 studio IDE setup can be referred from e² studio - information for RA family
2. Flash programmer windows installation path is at Renesas Flash Programmer V3.19.00 Windows | Renesas, and for other OS (Linux, Mac) is at Renesas Flash Programmer (Programming GUI) | Renesas
3. Download and install SEGGER J-link Software.
4. Download and install Python.
5. Install PTX2xxR RUL Config Tool v1.5.0 (or higher version) application from the provided **.zip** package by double-clicking on the **.exe** windows specific installer.
6. Install SDK from the provided archive **ptx_RUL_SDK_v1.2.0.zip**

For more details, check with Renesas technical support.

4.1 PTX200R RUL SDK v1.2.0 Package Overview

The SDK content download includes a **.zip** package containing the `README.md` file.

- For SDK architecture and content, refer to **ptx_RUL_SDK_v1.2.0/DOCS/html/index.html**
  - System overview
  - HW/SW Architecture
  - Functional components
  - API explanation
- SDK source code (in **.zip** archive)
  - Example projects in folder '**PROJECTS\PTX2xxR**'
  - Chosen example: "**NFC_FORUM_NDEF**"
- PTX2xxR RUL Config Tool
  - Tool to configure, fine-tune RF configuration or run demo applications
  - User manual: PTX2xxR_RUL_Config_Tool/v1.5.0/user_manual

5. Hardware Installation for PTX2xxR

Connect the Tag-connect Spring-Pin cable to the MCU board's SWD connector and to the PC using the USB-C cable with J-link HW (see Figure 3), as follows:

1. J-Link BASE Debug Probe.
2. Tag-connect Adapter ARM20-CTX 20-Pin to TC2030-IDC Adapter for Cortex | Tag-Connect.
3. Tag-connect Spring-Pin Cable 6-pin Plug-Of-Nails™ no-legs to IDC cable 10"/254mm | Tag-Connect.
4. To flash and debug the device, connect the PTX2xxR using J-Link connect cable and to a PC using a USB to microUSB cable.

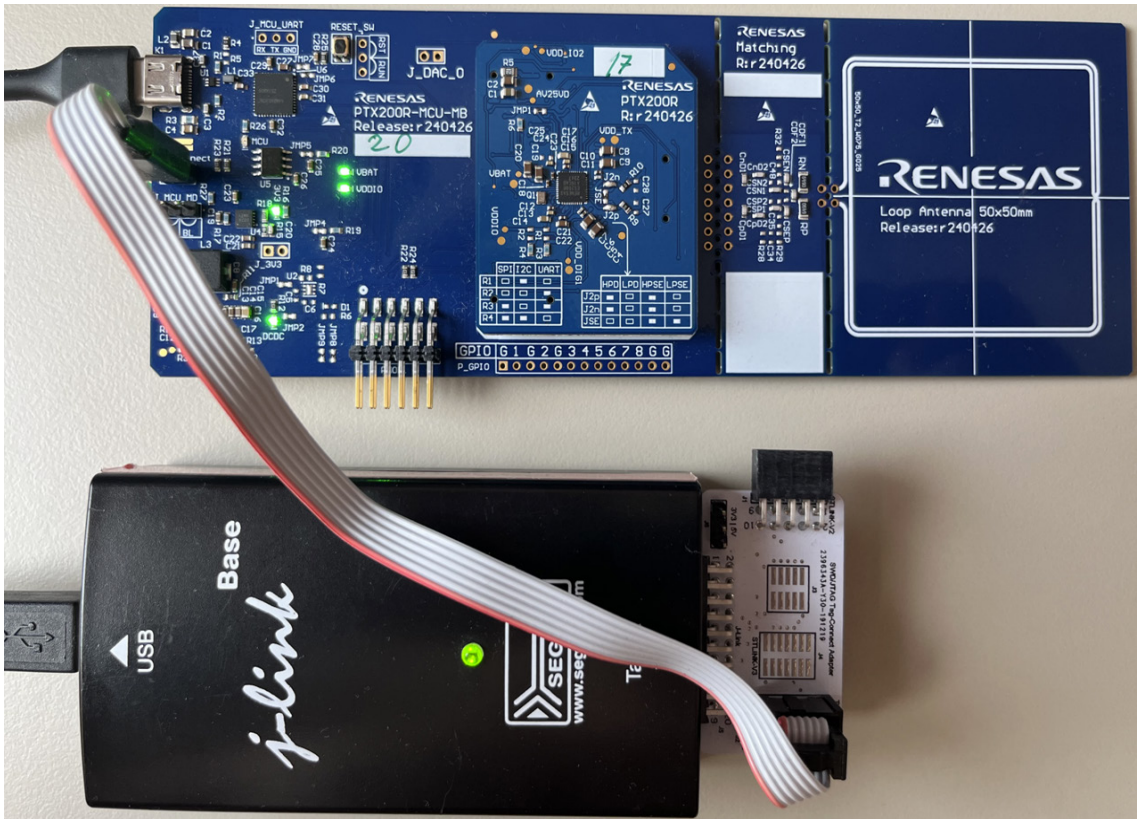


Figure 3. PTX200R Hardware with J-Link Connection

5.1 PTX2xxR High-Level Application Block Diagram and Schematic

The PTX2xxR-EVB contains the following three parts. It operates using four distinct power rails and internal controller/FW autonomously handles most NFC-related tasks.

1. PTX2xxR IC board: IC module together with blocking caps, additional component (for example, EEPROM) to identify the IC module board version
2. MCU board with PTX RUL SDK running on it, power supply, other interface components, and USB/PMOD connector.
3. Antenna board.

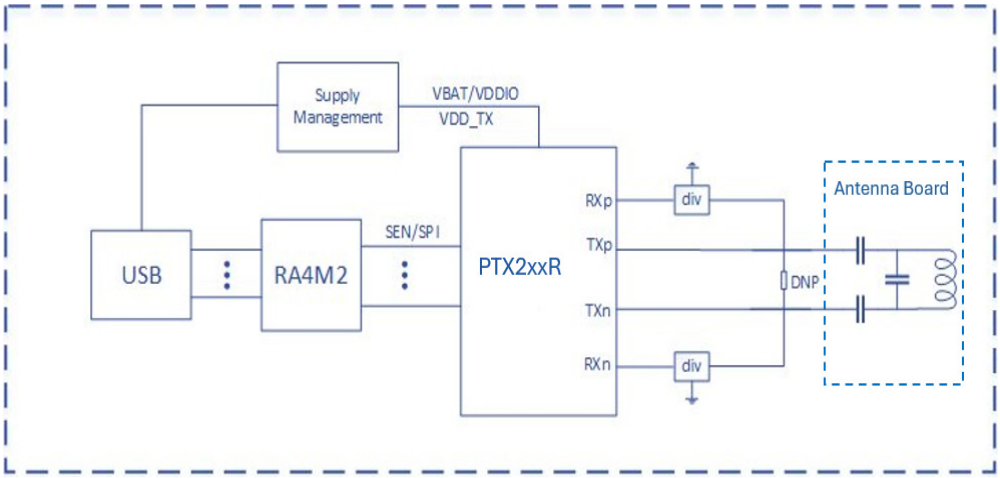


Figure 4. PTX2xxR High-Level Application Schematic

The PTX2xxR offers three different communication interfaces:

- SPI, I3C/I²C, or UART (as in sample PTX200R HW)

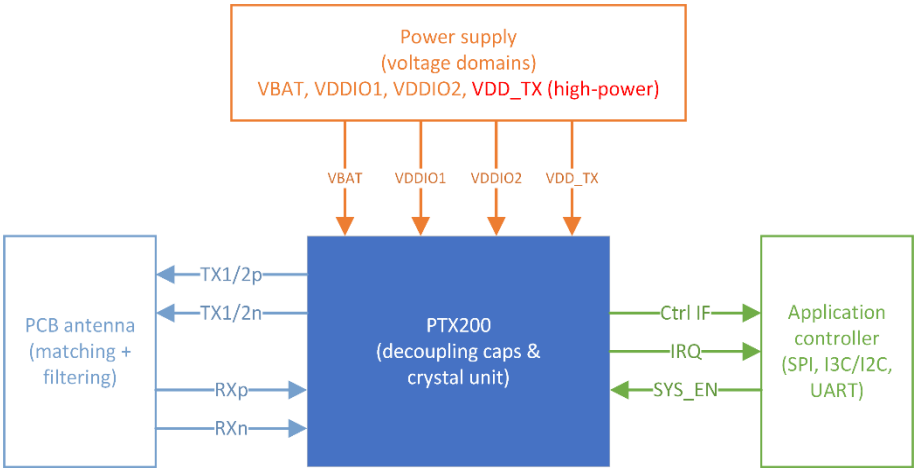
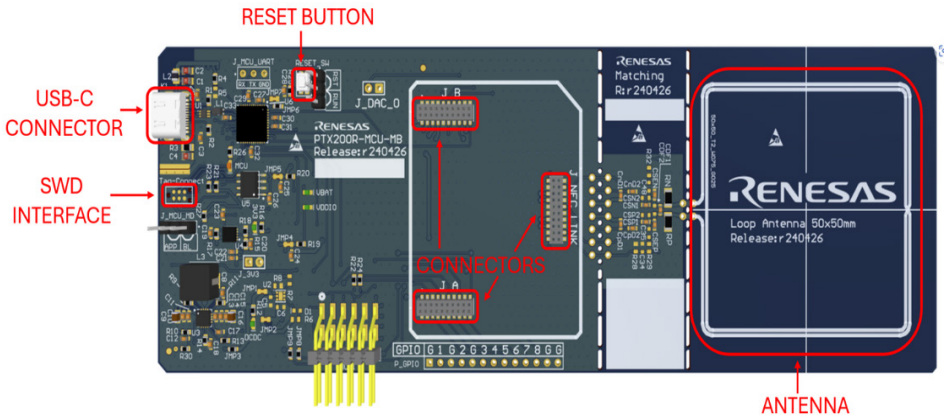


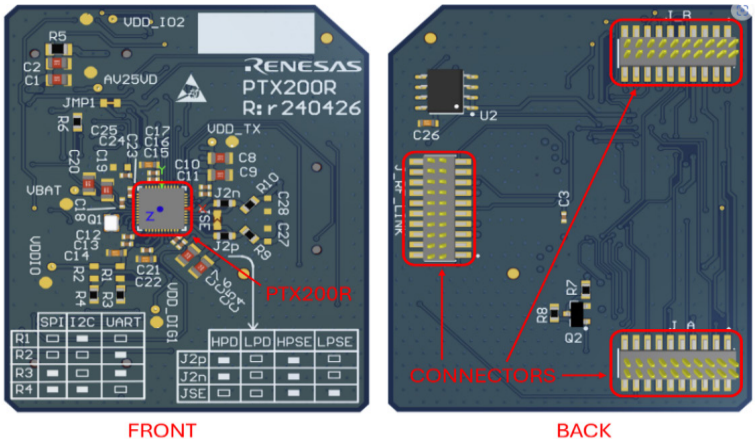
Figure 5. PTX2xxR High-Level Application Block Diagram

5.2 PTX200R Hardware Sample

- MCU board PTX200R-MCU-MB sample:

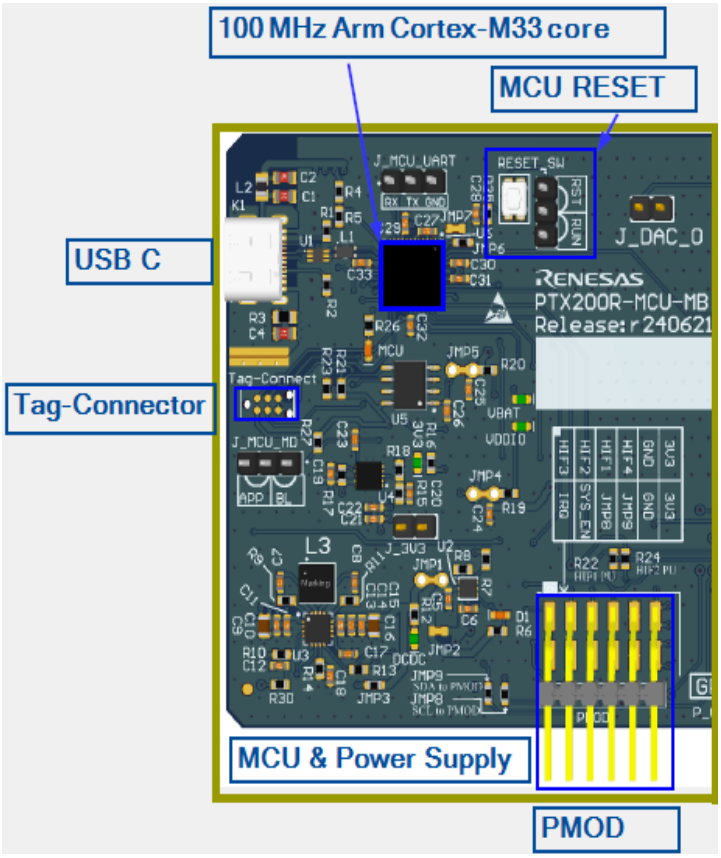


- IC Board PTX200R QFN44 sample:



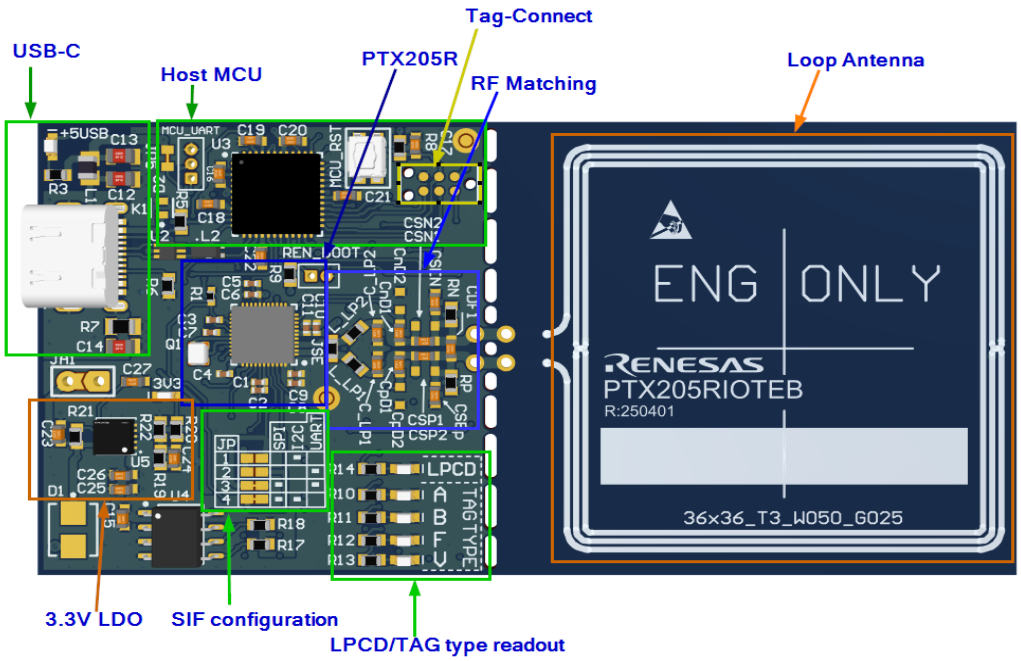
5.3 PTX200R PMOD Setup

PMOD connector facilitates SW development using external MCU where the local MCU is kept in reset (inserting near RST jumper as shown in diagram), while the PTX2xxR is driven externally with a MCU eval board via SPI.

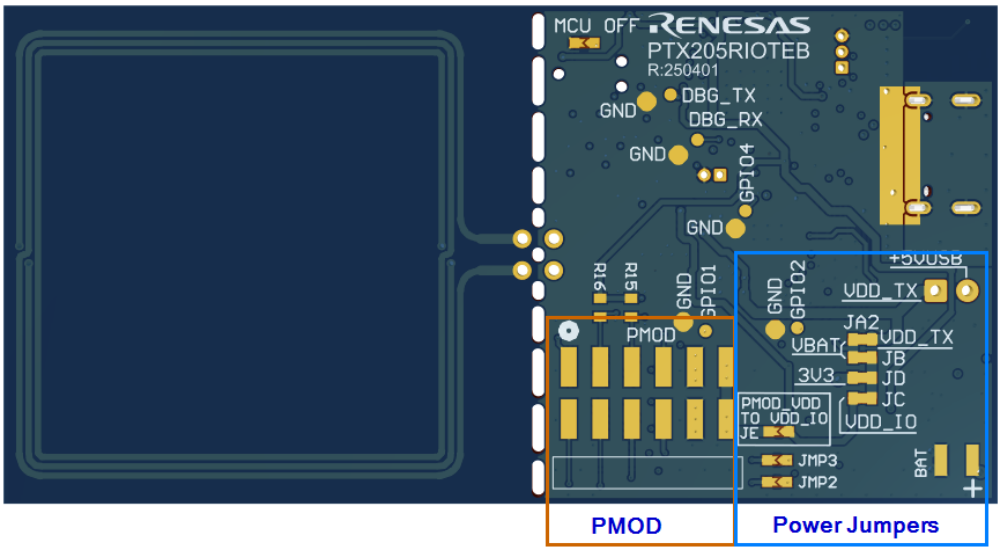


5.4 PTX205R Hardware Sample

- MCU board PTX205R-EB sample:



- PMOD connector PTX205R QFN44 sample:



6. e2 Studio IDE Setup for PTX2xxR RUL SDK

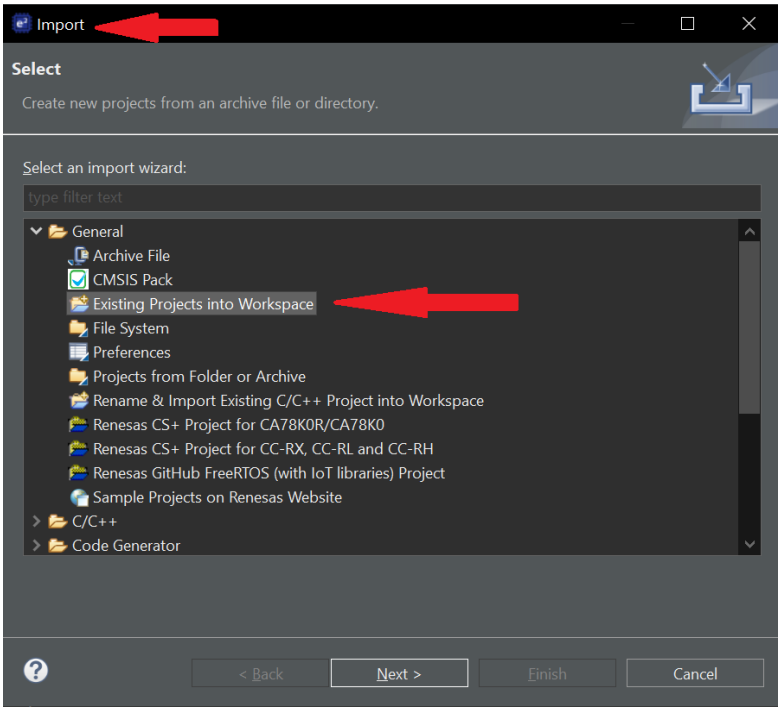
Open the project in the Renesas provided IDE (integrated development environment), specifically e2 Studio. The IDE is downloadable from e2 Studio as mentioned in section 4.

Download the recommended Flexible Software Package (FSP) version 6.1 and continue with installation.

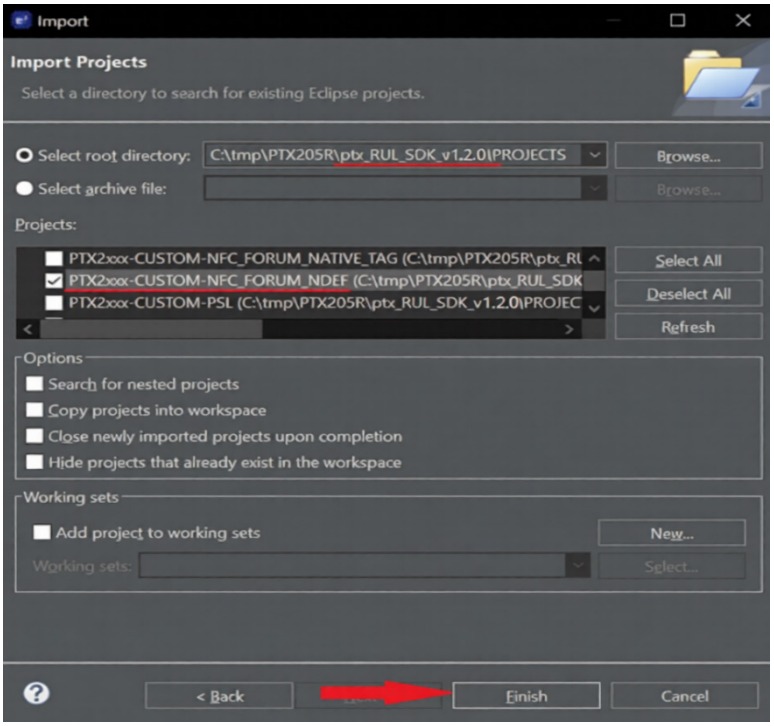
Once the required software and hardware pre-setup is done, load the SDK project using the e2 studio IDE and build a custom configuration (as explained in the following sections).

6.1 Importing and Running the Project

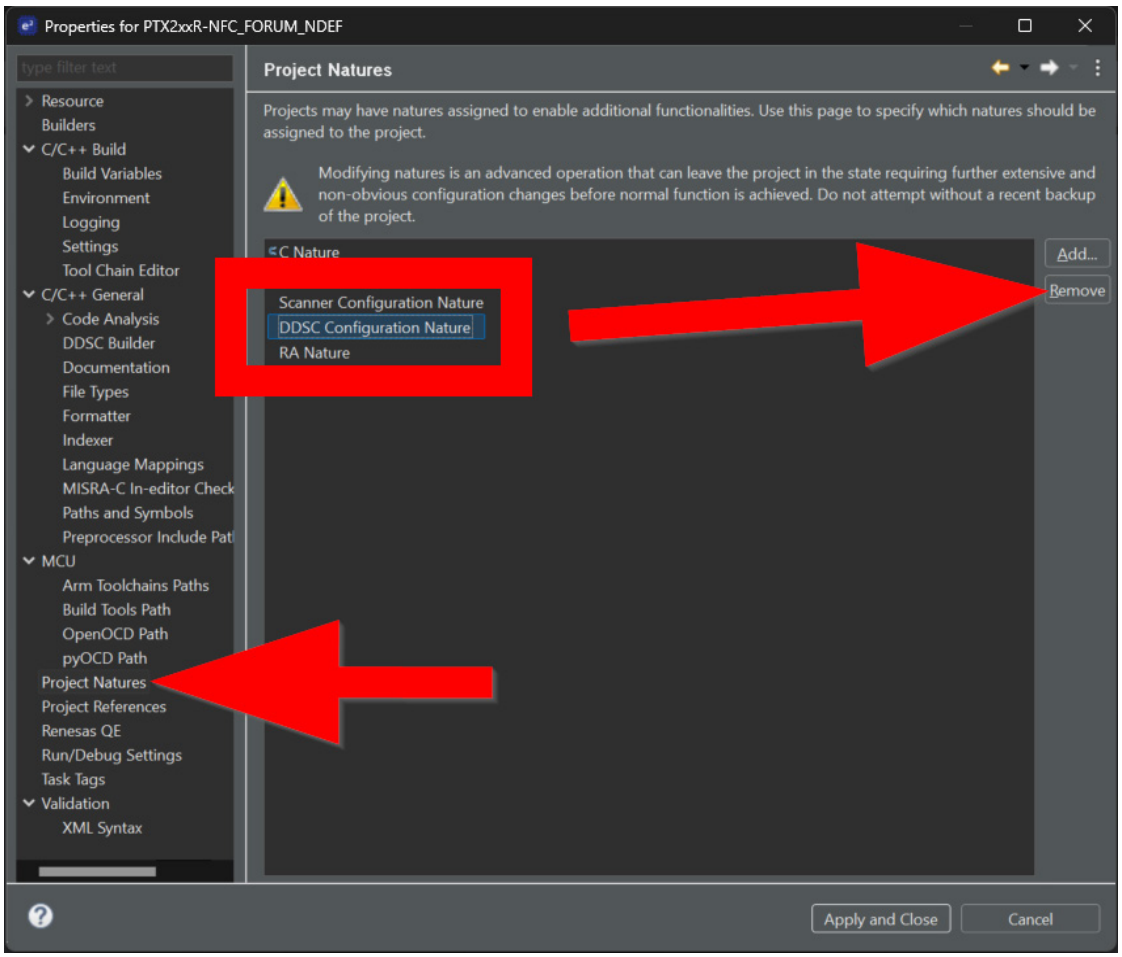
1. Unzip the SDK archive.
2. In e2studio, import existing projects.
3. Use option File →Import→Existing Project into Workspace from the drop-down menu.



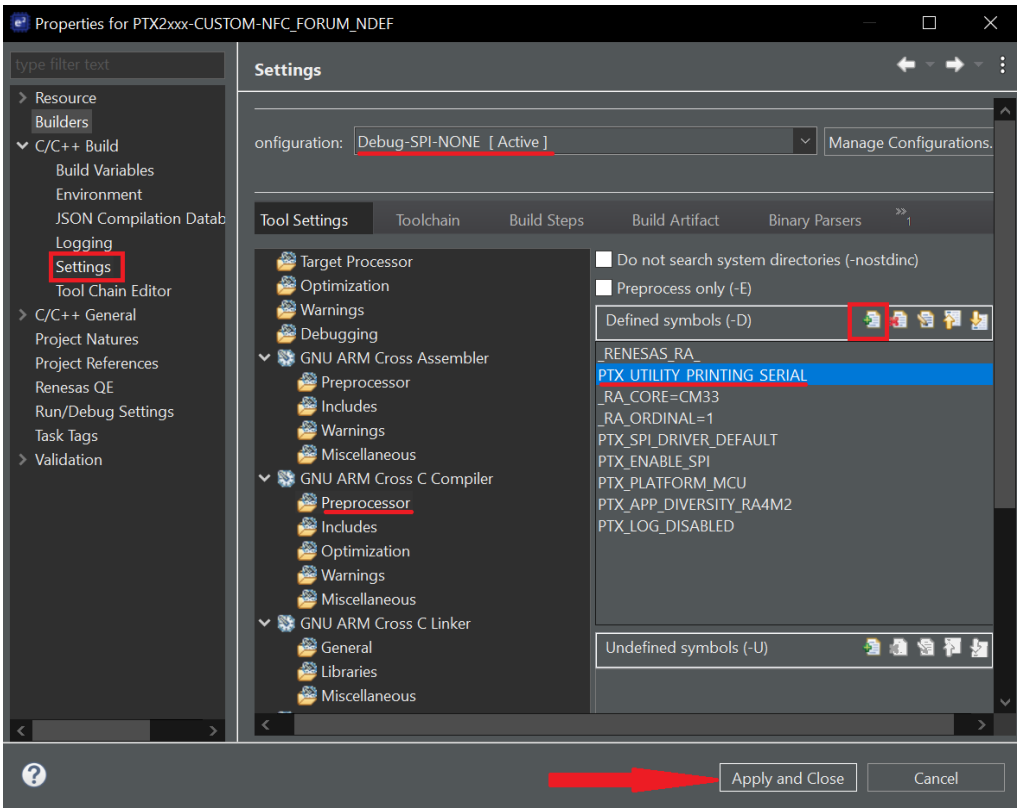
4. Choose example application **NFC_FORUM_NDEF** to complete the import procedure.



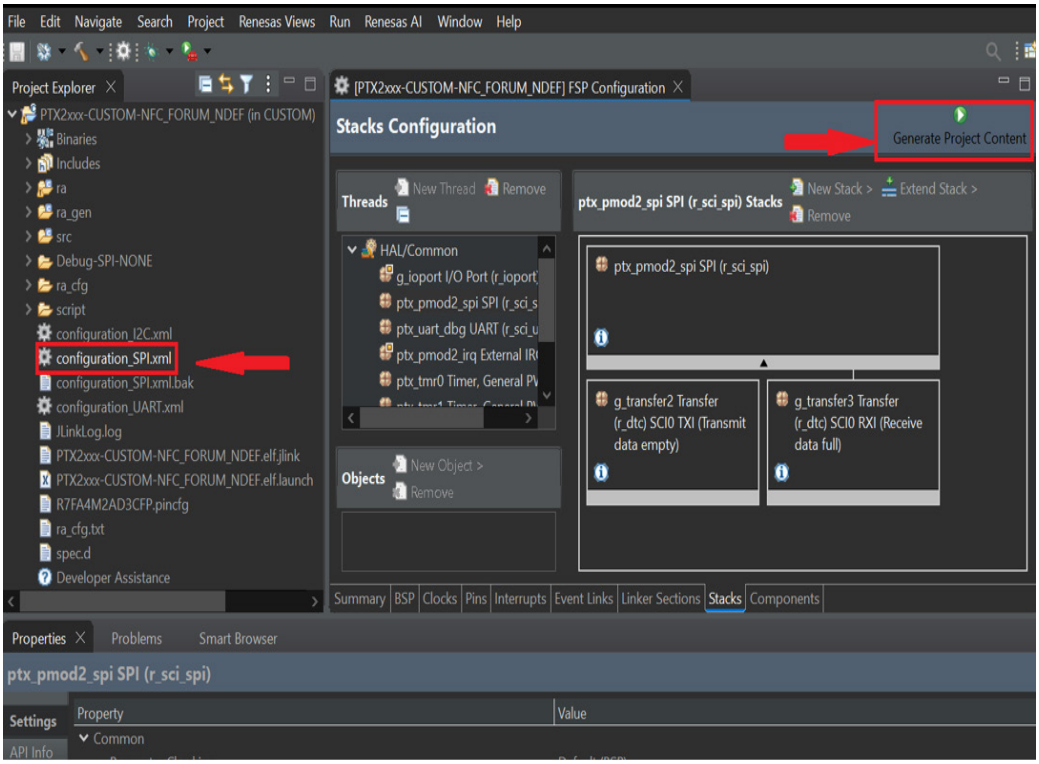
5. Within the Project Natures menu, go to C/C++ Project Settings. Remove DDSC Configuration Nature.



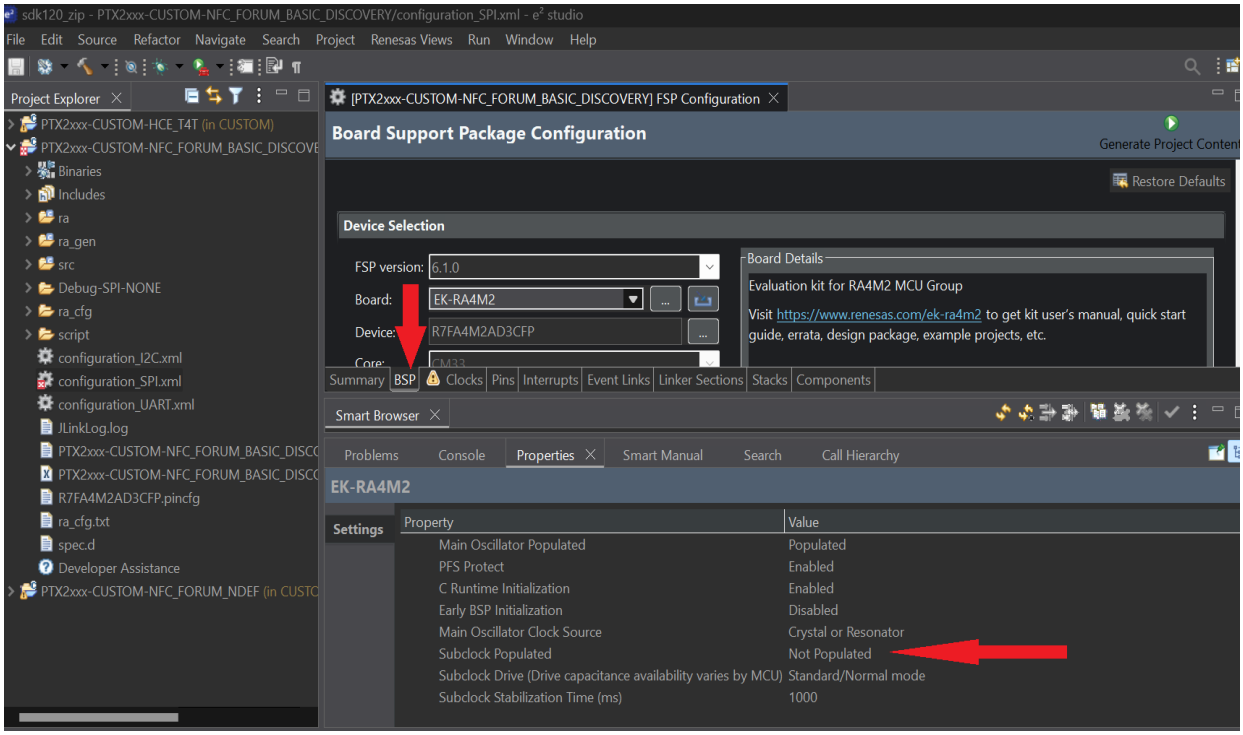
6. A new preprocessor macro **PTX_UTILITY_PRINTING_SERIAL** needs to be added to the default target firmware to assist in enabling serial logging.



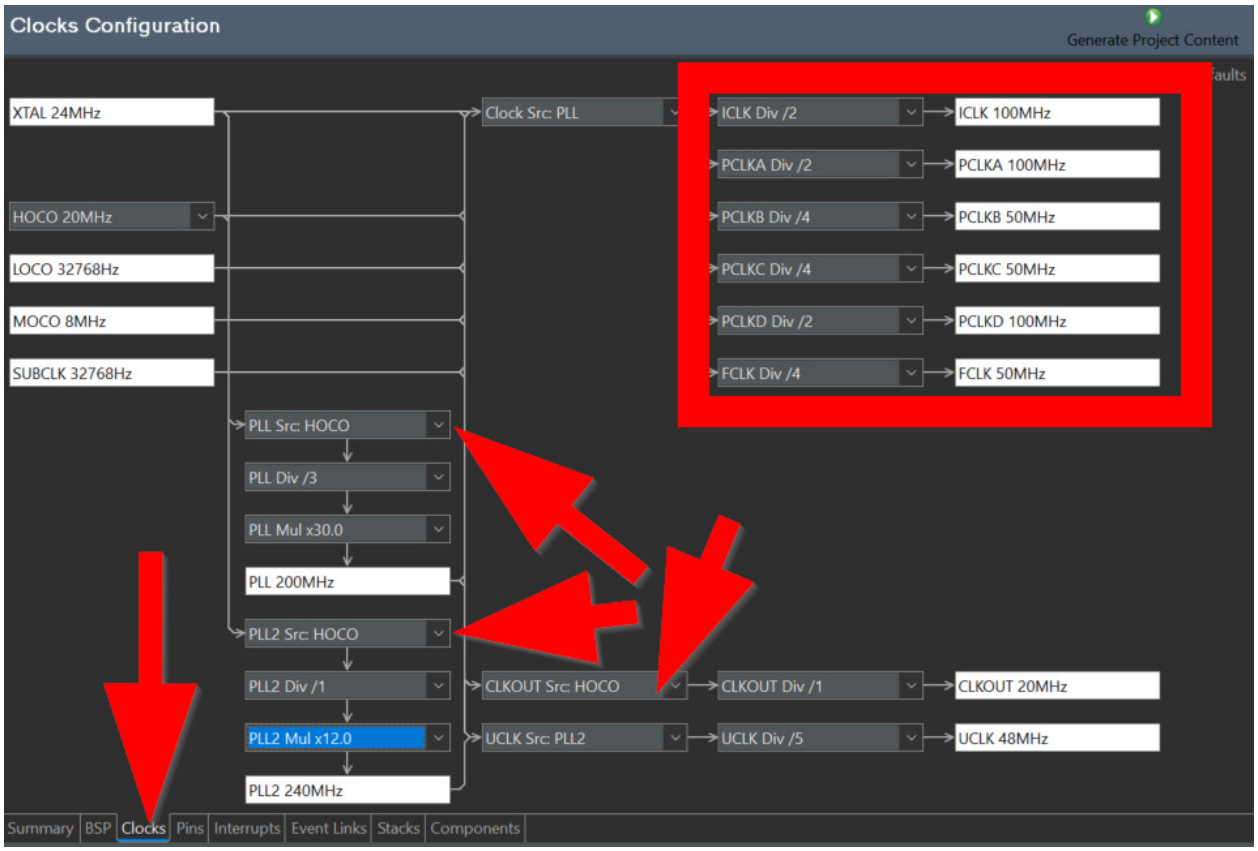
7. **Generate the source code**. To open, double-click on SPI configuration and generate project content by pressing the button.



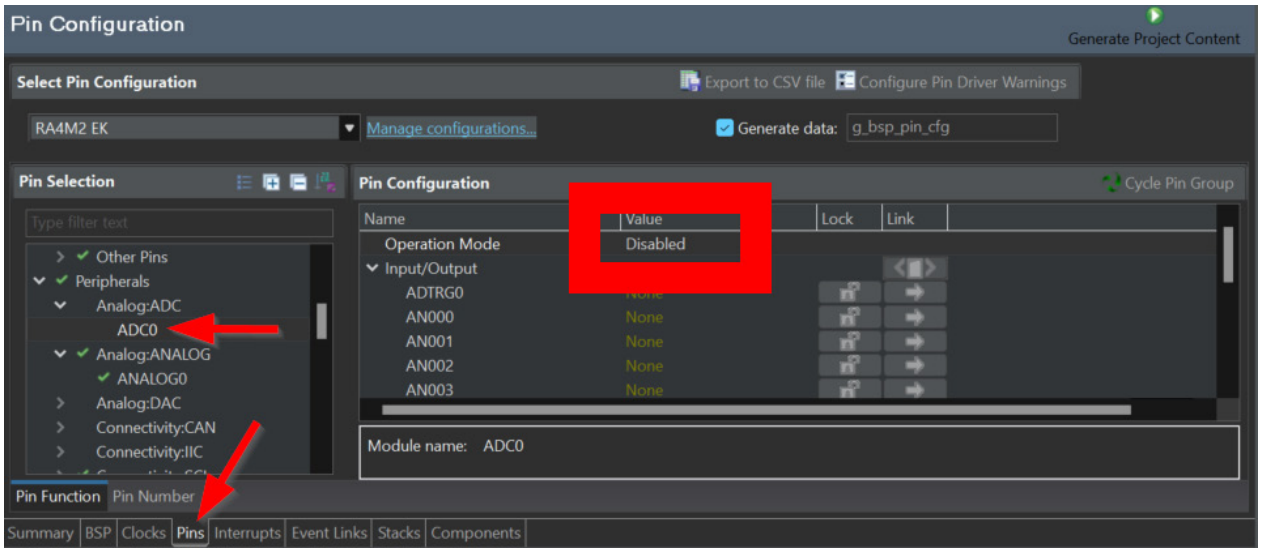
8. **Adapt BSP configuration**. Since there is no sub-clock source populated on the board, it needs to be configured in the **BSP** tab.



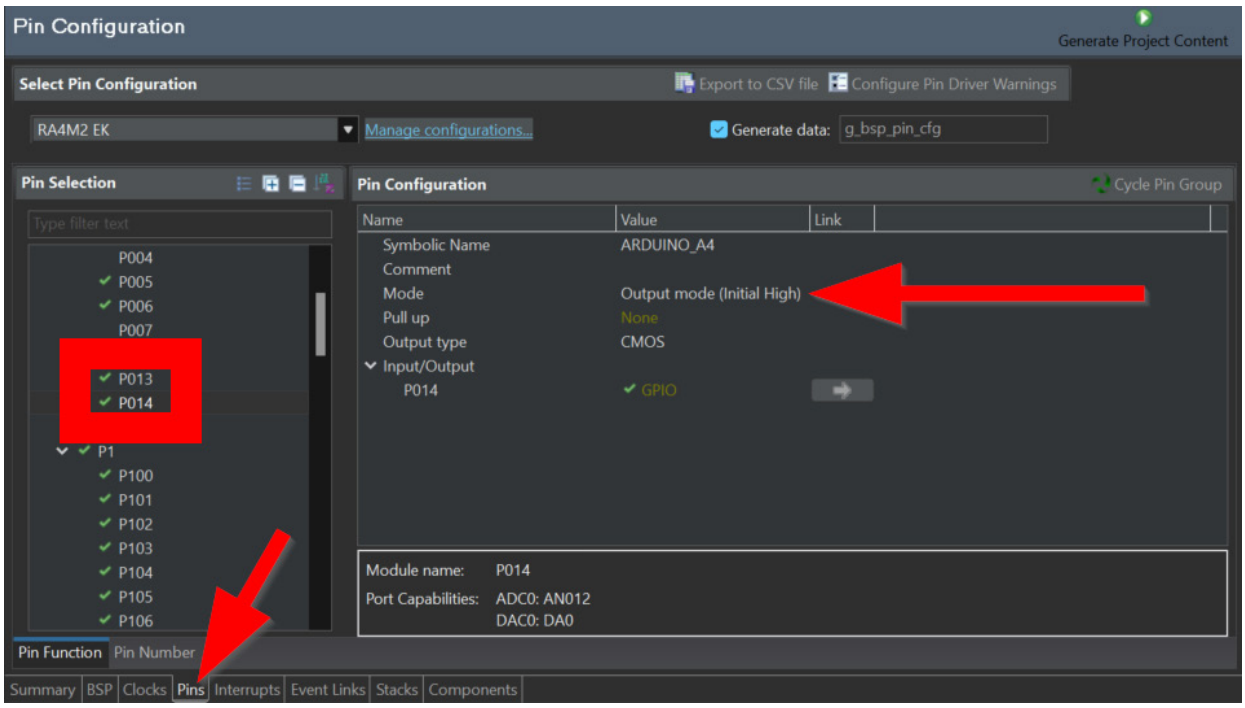
9. **Adapt clock configuration**. Due to lack of XTAL, set the clock source to HOCO and PLLs adjusted to achieve an optimal MCU clocking.



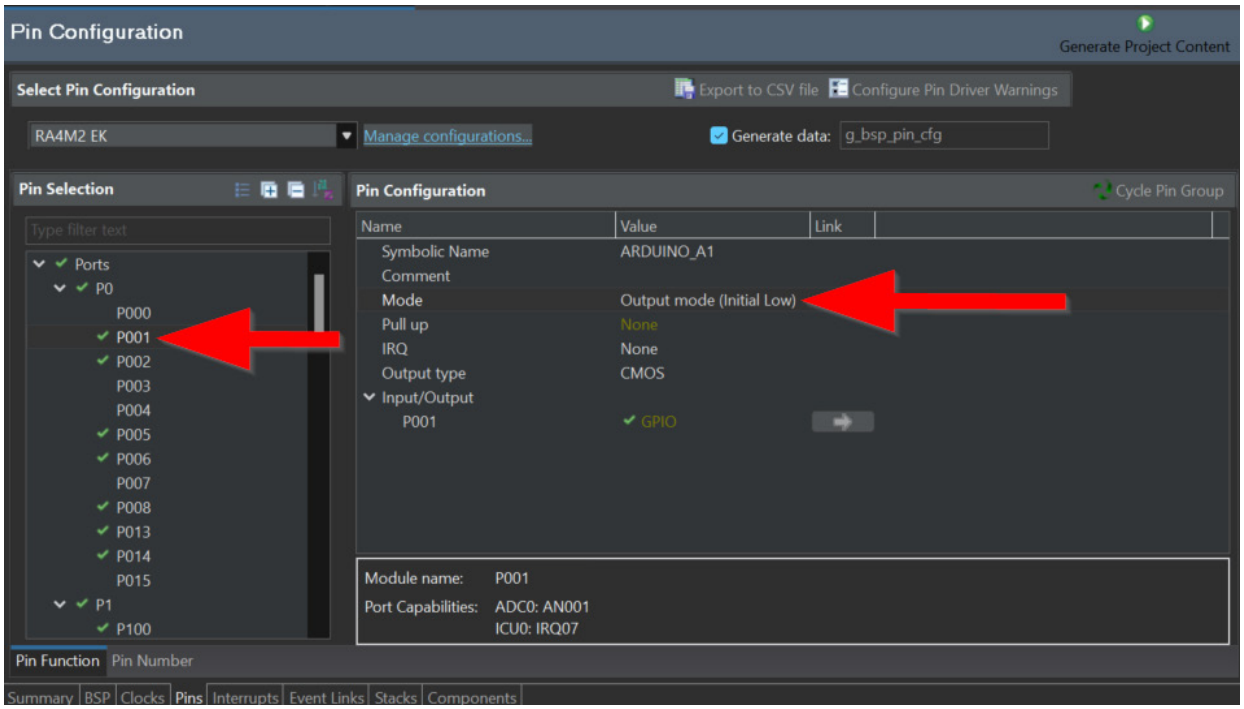
10. Existing ADC configuration may cause a conflict and needs to be disabled.



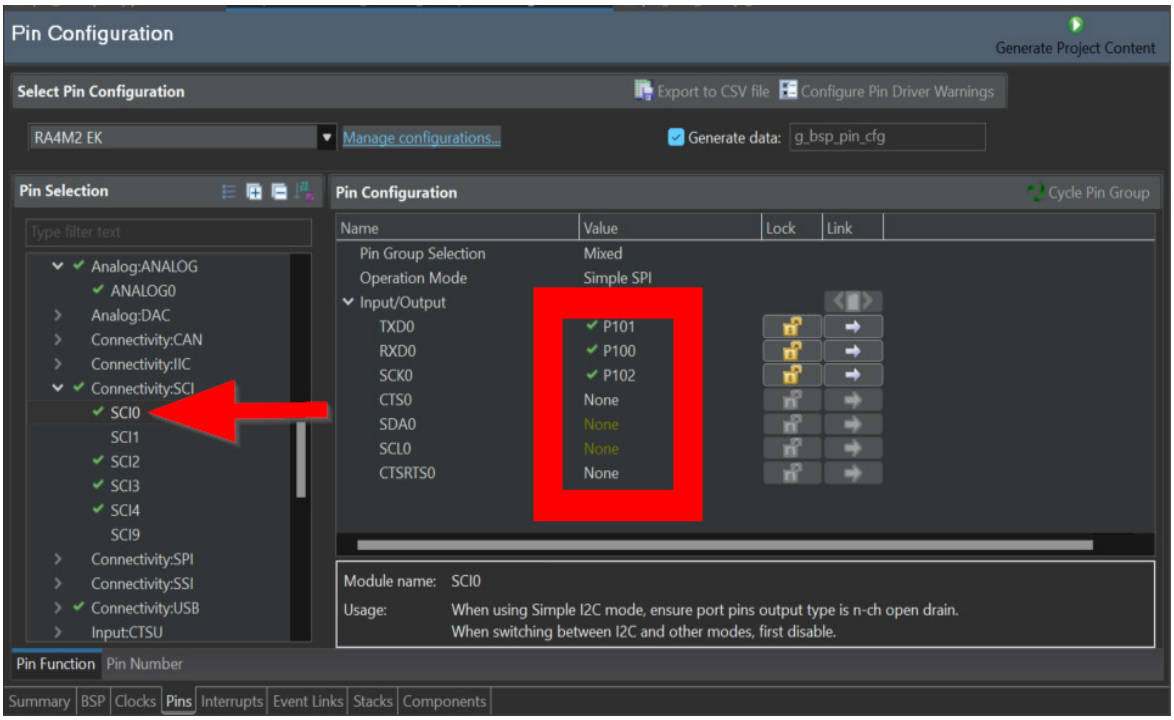
11. Configure the power supply (P013) and PTX enable (P014) pins to output **High** by default.

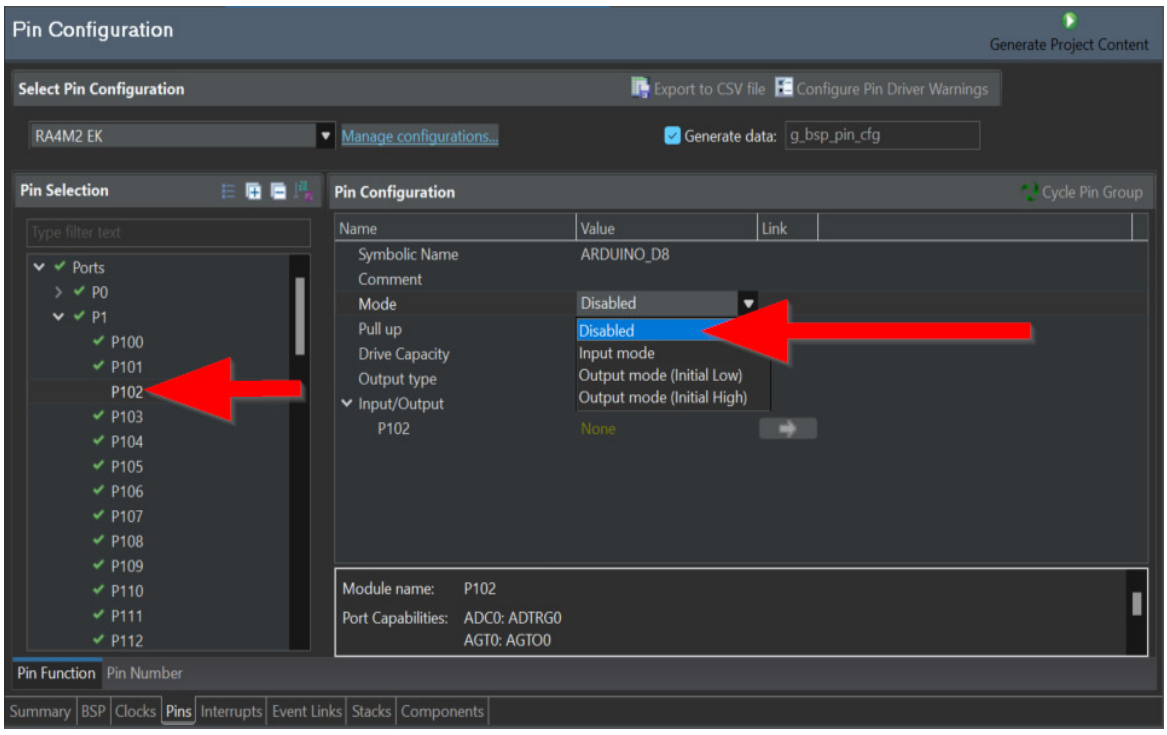


12. Configure PTX200R Reset (P001) to output **Low** to enable operation.

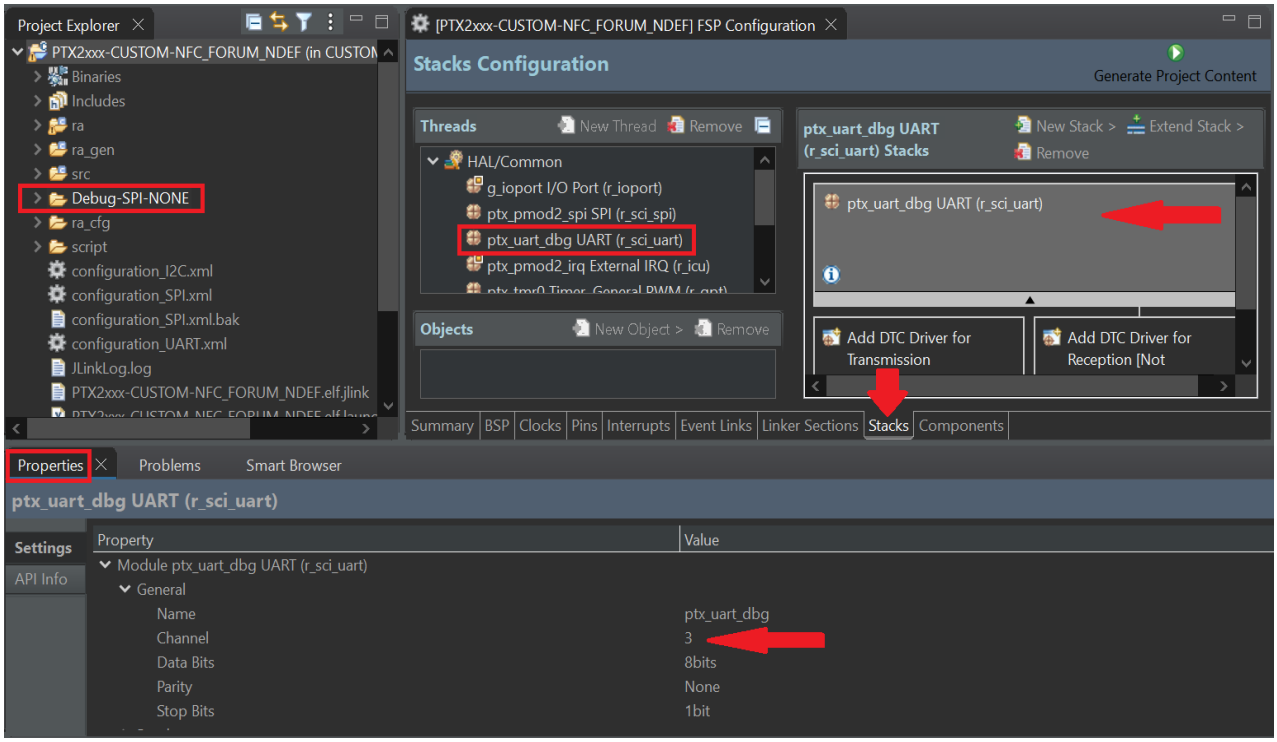


13. **Adapt configuration – SPI**. Configure SPI “SCI0” to use pins P100-P102. If a pin conflict occurs, the pin needs to be disabled in the individual pin configuration section

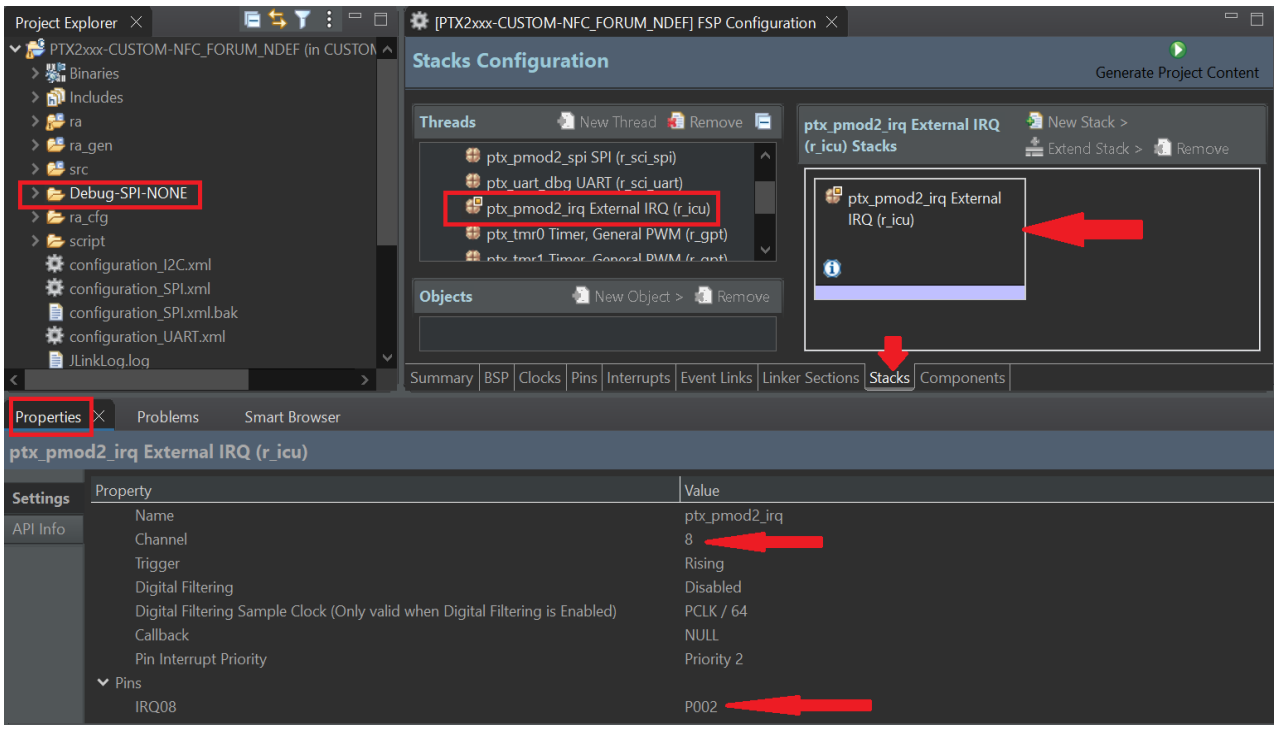


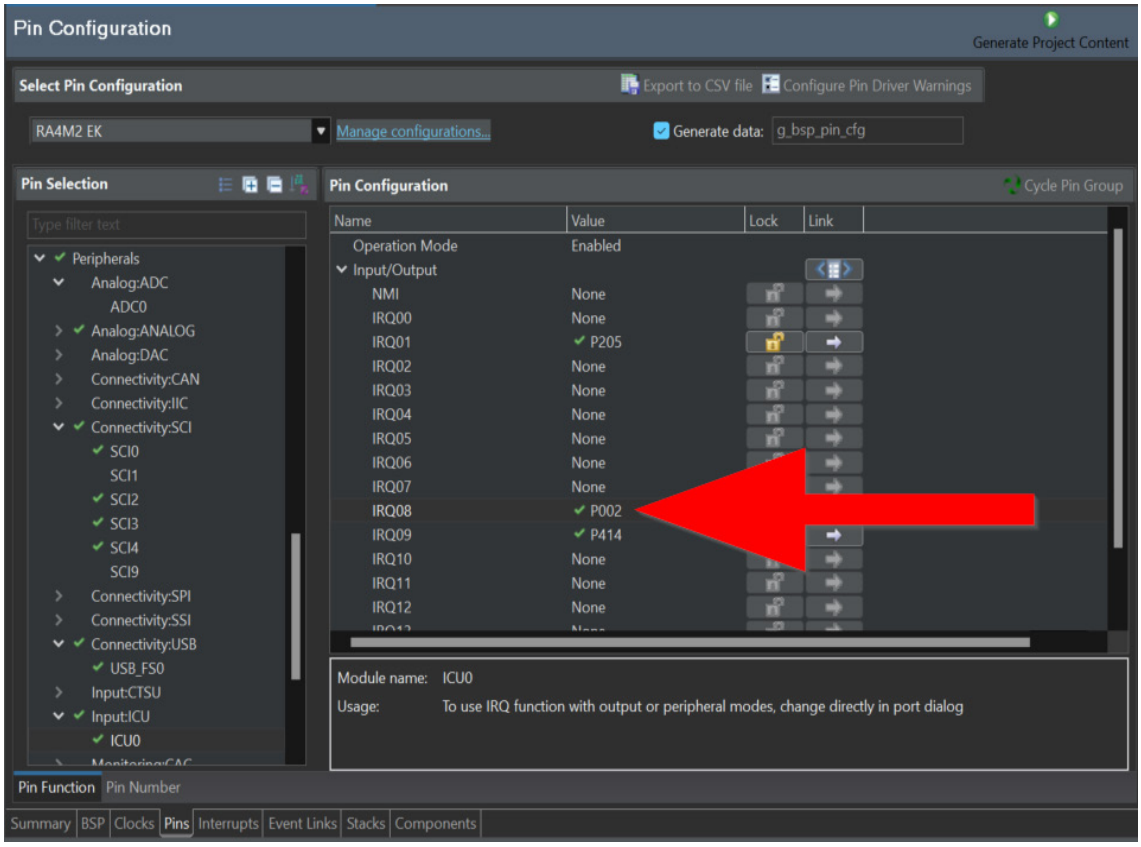


14. The default build target **Debug-SPI-NONE** without the previously applied **PTX_UTILITY_PRINTING_SERIAL** preprocessor macro handles message prints via semi-hosting, which is slow and cumbersome to view in e2studio.
15. **Additional configuration of the debug UART**. Changing the channel to '3' enables the log messages to be transmitted through the MCU_UART board terminals (P408-RX and P409-TX) using a USB/UART converter and a terminal application (for example, PuTTY or Tera term set at a default 256000 baud rate).

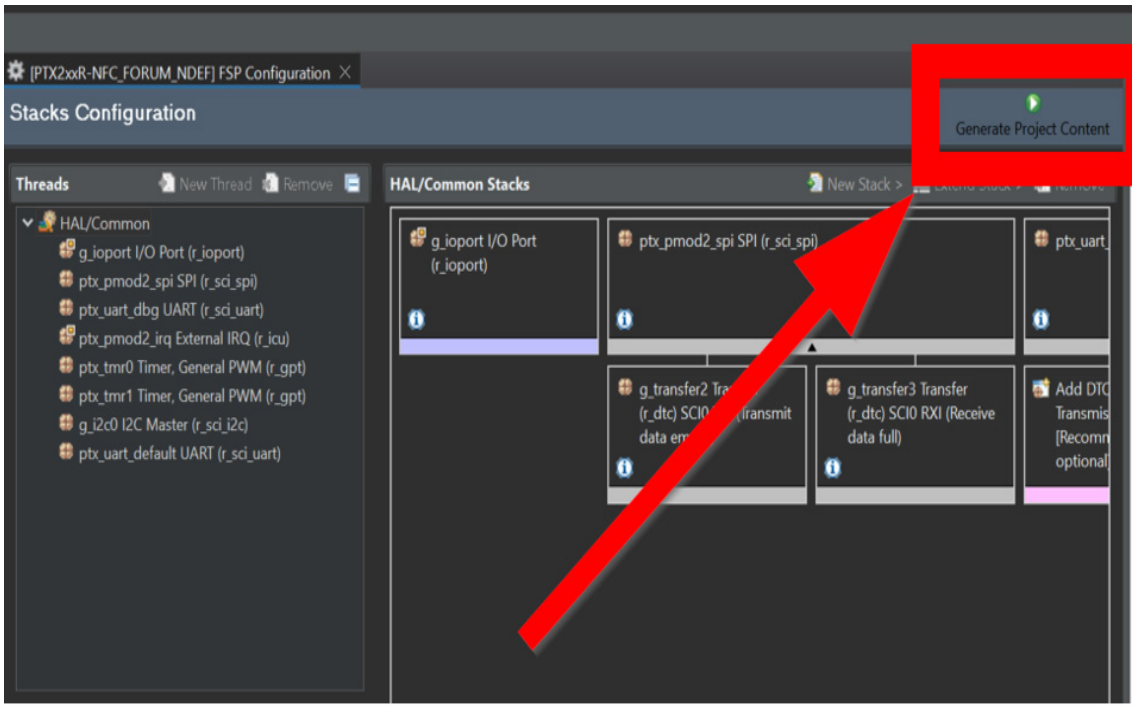


16. **Adapt configuration for IRQ**. IRQ channel (8) and pin (002) configuration needs to be changed. Set IRQ channel '8' input pin to P002 in stack configuration section.

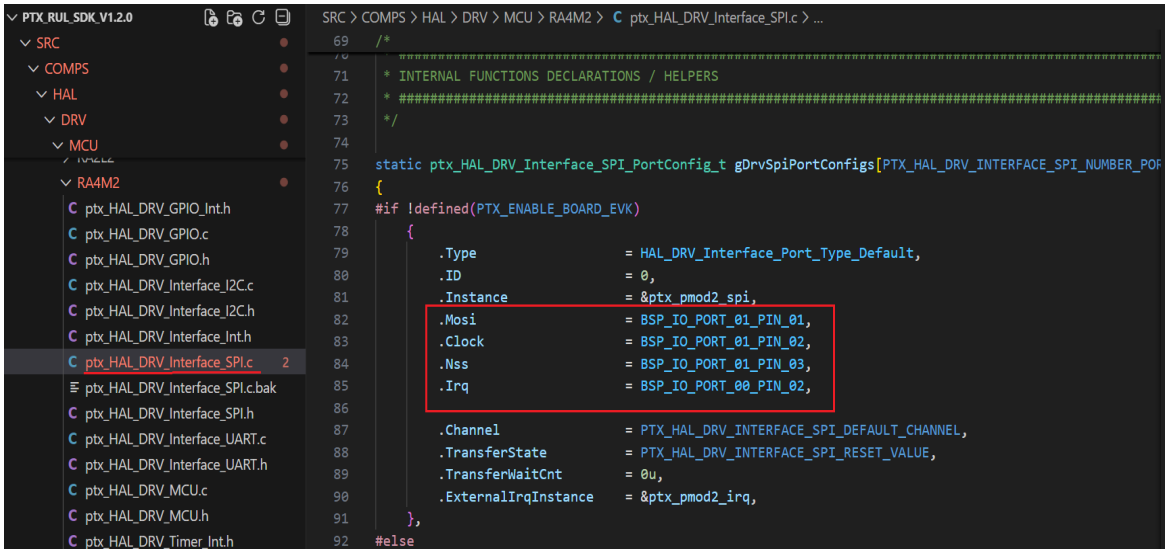




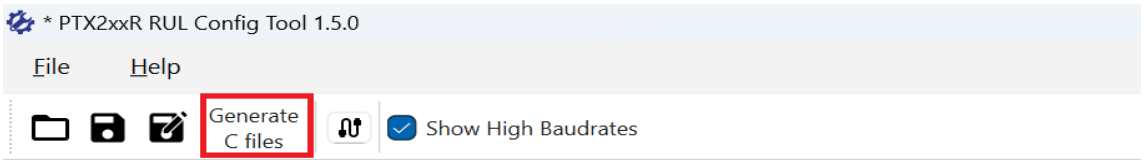
17. **Regenerate project content**. Due to the many changed settings, it is highly recommended to allow e2studio to regenerate the altered configuration files.

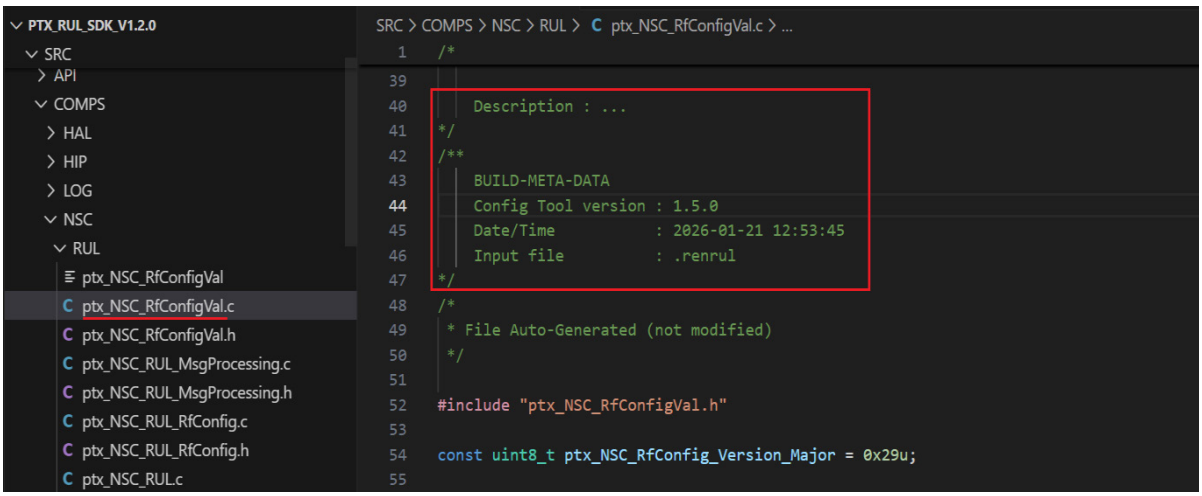


18. **Adapt HAL – SPI**. SPI pin configuration update as required in **SRC\COMPS\HAL\DRV\MCU\RA4M2\ptx_HAL_DRV_Interface_SPI.c**

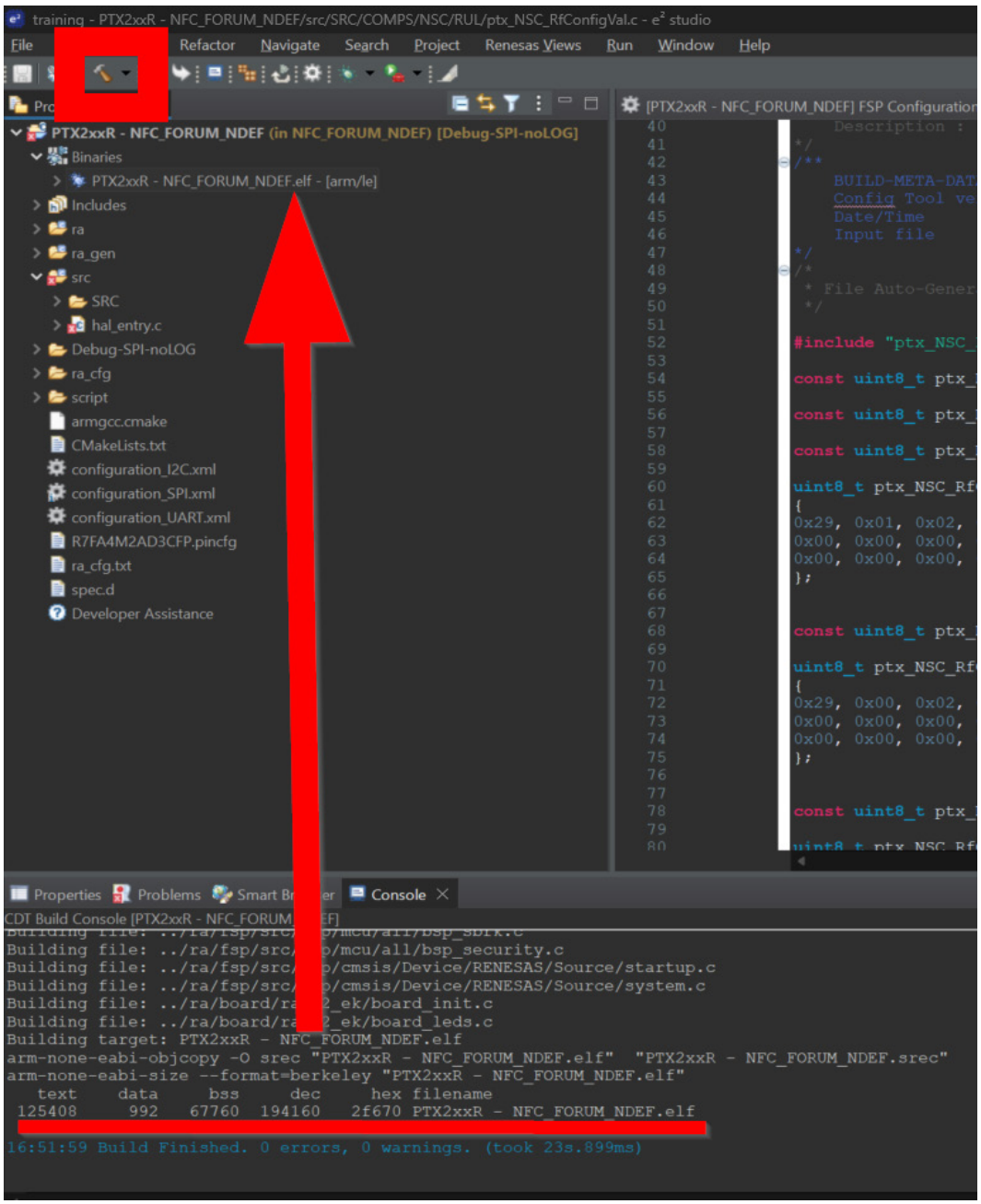


19. **Adapt RF-configuration**. Use the provided **.zip** file **PTX2xxR RUL Config Tool v1.5.0** to export an RF-Config file and overwrite the existing **ptx_NSC_RfConfigVal.c** and **ptx_NSC_RfConfigVal.h** files in SDK folder "**SRC\COMPS\NSC\RUL**" as required for setting up sample projects.

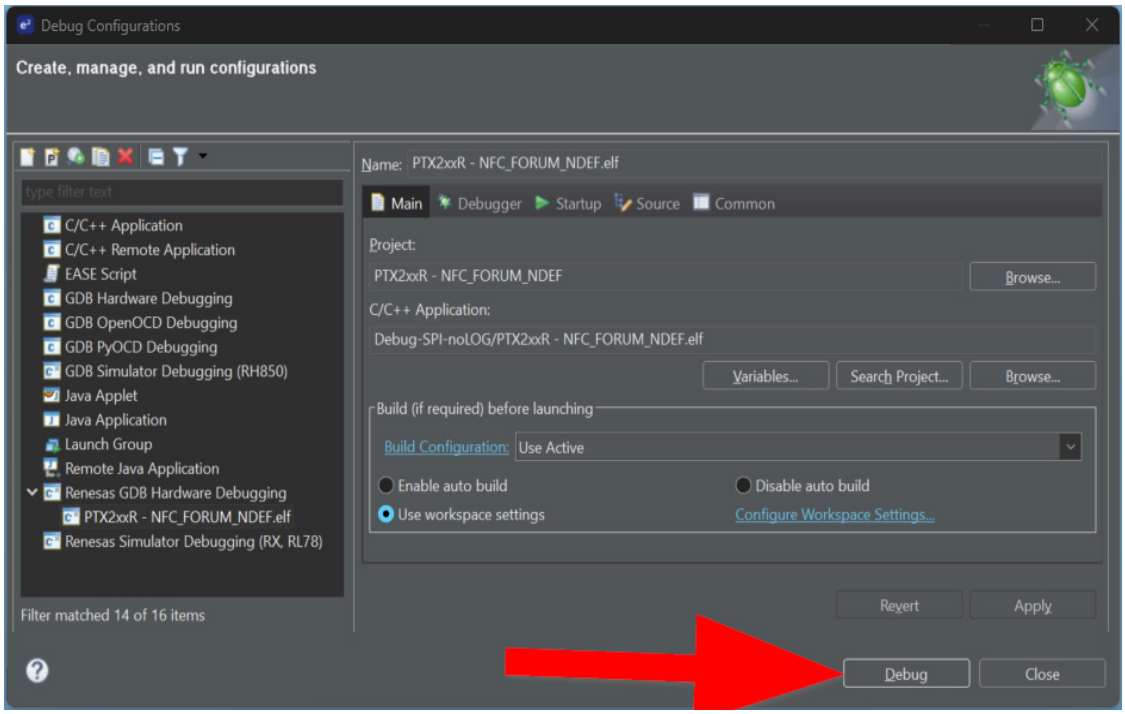




20. Build the firmware.
    a. Clicking on the hammer icon builds the firmware

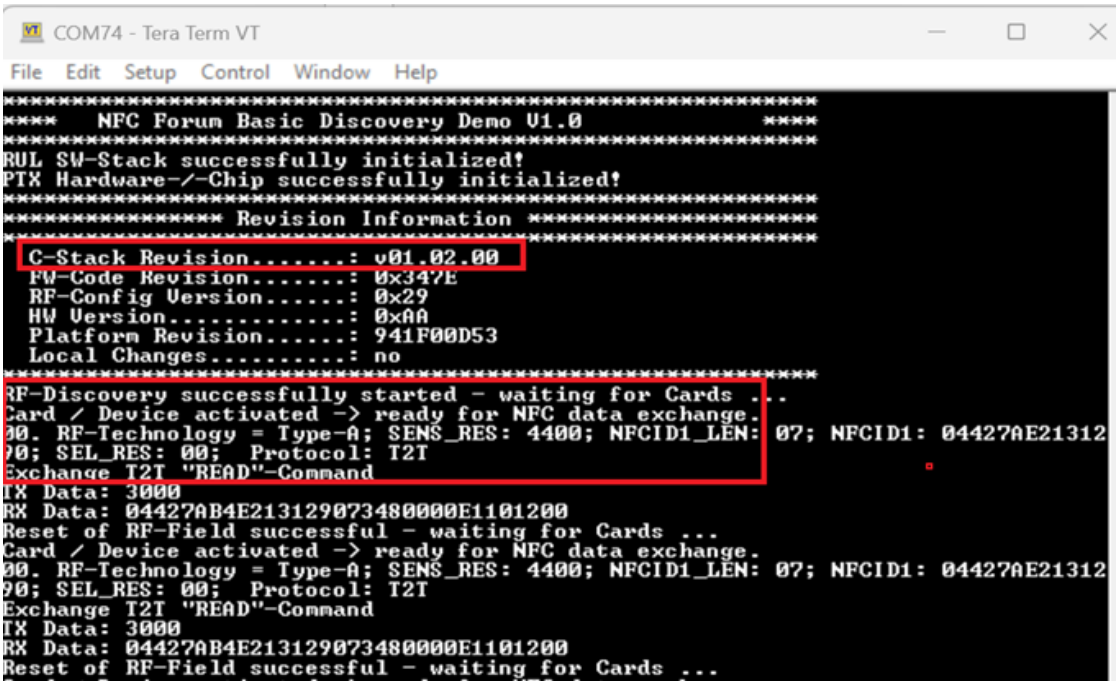


b. Right-click the **.elf** file and select **Debug As...** and **Renesas GDB Hardware Debugging**.
c. Configure debugger (for example, J-Link) connected to the board via the 6-pin TagConnect interface.
d. Start debugging with **Debug** button by choosing Target device as **R7FA4M2AD** for PTX2xxR.



21. Run the firmware.
    a. Run a terminal application and open the communication port (COMx) to see the messages printed by the firmware. You can set up any USB to TTL converter to capture these serial console logs by using, for example, the Tera Term terminal emulator operating at default 256000 baud rate. Referring to the hardware example in section 5.4, simply connect HOST_MCU UART_TX pin to RX of USB to TTL converter accordingly.
    b. After starting the debug session, press F8 twice to run the firmware.
    c. The NFC polling starts and messages are displayed in the terminal window as soon as the Poller HW is brought in proximity to the antenna of PTX200R/ PTX205R EVB.

22. Test using the sample NDEF application.
    a. NDEF (NFC Data Exchange Format) is a standardized message format used in NFC for exchanging data between devices. NDEF messages consist of one or more NDEF records, each containing a specific type of data, like a URL, text, or contact information for different NFC devices to understand and interpret the information.
    b. Once the sample NDEF app loaded in e2 studio is built and run when moving different cards into the RF-field, the inquired information and the potentially read NDEF message present on the card supporting this type will be displayed on the terminal as shown in the following image.



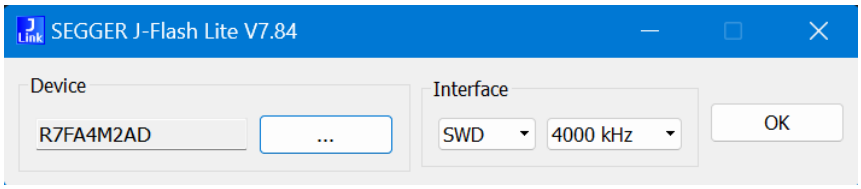
7. Setup and Testing using the PTX2xxR RUL Config Tool

Additionally, the *PTX2xxR RUL Config Tool* can be used to generate configuration data for the PTX2xxR SDK and the user can also perform RF-activities on the board using this PC application. The following sections explain how the default bootloader and sample NDEF application programming can be run on PTX2xxR EVB. The MCU has two pieces of firmware:

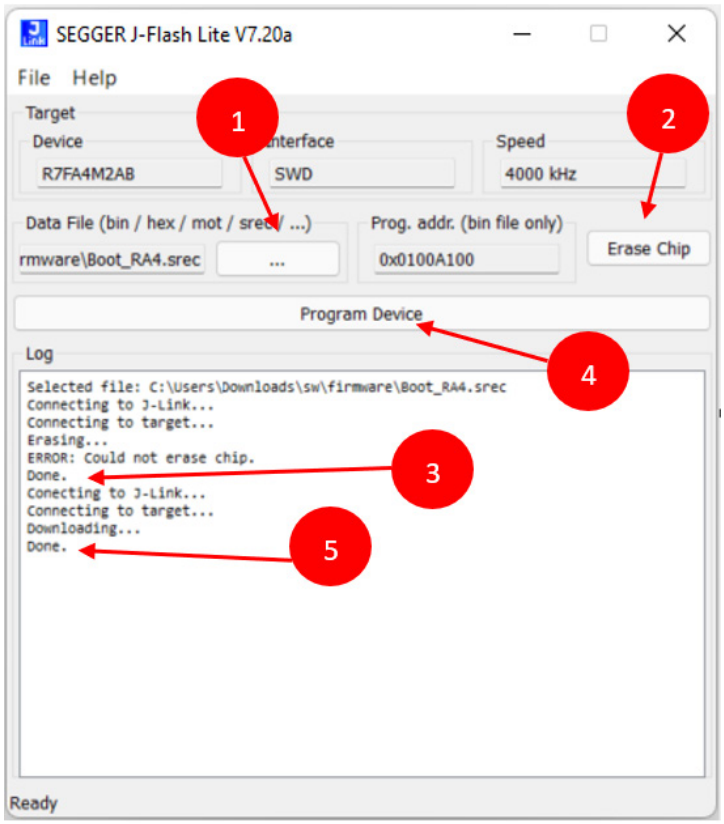
- **Bootloader**: Always present, it is used to perform FW upgrade and start the actual application FW.
- **Application (Reader) FW**: Main firmware that executes the commands.

7.1 Flashing the Bootloader to the MCU Board

1. Extract the content of the archive **PTX2xxR_RUL_Config_Tool__v1.5.0.zip**.
2. Find Boot_RA4.srec under PTX2xxR_RUL_Config_Tool__v1.5.0\firmware.
3. Open **JFlashLite.exe** (installed as part of J-LINK Software bundle).
4. If a message box with title **J-Flash Lite Info** pops up, click the **OK** button.
5. Configure the initial dialog as shown in the following image and click **OK**.



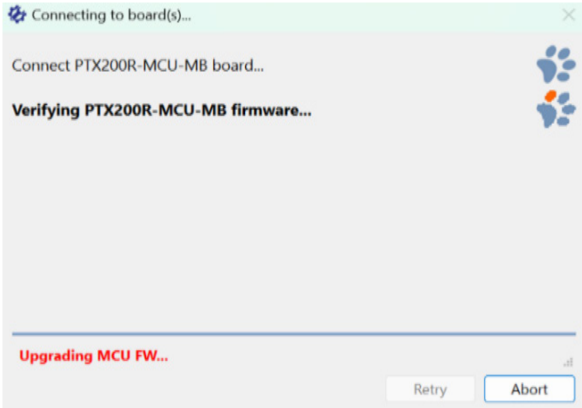
6. Connect the Tag-connect Spring-Pin cable to the MCU board's SWD connector.
7. Connect the MCU board to the PC using the USB-C cable (refer to section 5).
8. Load **Boot_RA4.srec** file mentioned above by pressing the **…** button (see item 1 below).
9. Click on the **Erase Chip** button and wait until the status is **Done** (see items 2 and 3 below).
10. Click on the **Program Device** button and wait until the status is **Done** (see items 4 and 5 below).
11. Disconnect the Tag-connect Spring-Pin cable from the MCU board's SWD connector.
12. Press the **Reset** button on the board.



8. Launching Tool and Testing

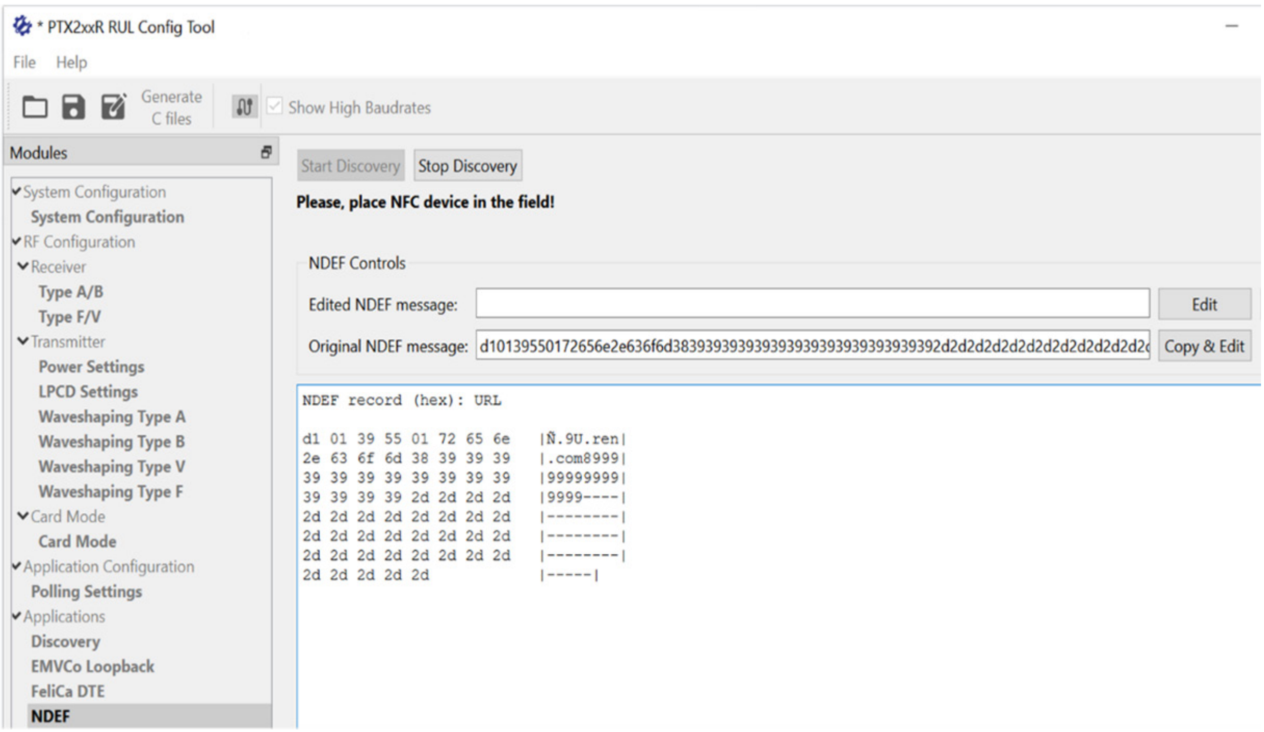
1. Start the *PTX2xxR RUL Config Tool* by double-clicking the following icon on the desktop.
2. Press the **Connection Wizard** toolbar button.

3. The following window appears and the board's firmware is updating:



8.1 NDEF Sample Application Running using Config Tool

1. The primary purpose of the NDEF application is to allow the use of the Config Tool for reading and writing NDEF messages to and from the tag. Tag must be pre-formatted and capable of supporting NDEF messages to use with the NDEF application.
2. After connecting to the config tool, select the **Applications→NDEF** entry.
3. Press **Start Discovery** to start NFC discovery. If a tag is detected in field, this view displays various details for the tag as shown the in the following example:



4. **Copy & Edit** button: Allows the user to edit the NDEF message that's read from the NFC tag.
5. **Edit** button: Allows the user to modify existing or create new NDEF message.
6. **Write** button: Triggers NDEF message write to NFC tag the next time the tag enters the field. Press **Cancel** to cancel a write operation.

For more information on supported options, refer to the PTX2xxR_RUL_Config_Tool/v1.5.0/user_manual.

9. References

SW-NFC2K-RUL-001-SDK-Release-Notes

PTX2xxR/W RUL (Reader Universal Library) SDK (v1.2.0) User Manual

PTX205R Datasheet and manual from PTX205R | Renesas

NFC Data Exchange Format (NDEF), Technical specification, version: 1.0

Renesas Innovative NFC for Security, Power, and IoT

PTX2xxR_RUL_Config_Tool/v1.5.0/user_manual

10. Revision History

Revision	Date	Description
2.00	Mar 20, 2026	Updated for SDK v1.2.0.
1.00	Aug 14, 2025	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.