HY0023
Software Development Guide

- The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by FDK CORPORATION is under license. Other trademarks and trade names are those of their respective owners.
- FDK Corporation is granted a license to use Slot Antenna on Shielded Package Technology patented by Toshiba Corporation.

Version	Date	Description
1.0	Dec.12, 2024	First edition issued

1 Overview

HY0023 is an evaluation board for software (firmware) development and evaluation with the ultra-compact Bluetooth® Low Energy module HY0021 with integrated nRF52805 from Nordic Semiconductor Corporation and Toshiba Corporation's Slot Antenna on Shielded Package technology. This document describes how to set up the software development environment using HY0023, and how to build, write, and debug sample source code.

2 Hardware Requirement

Hardware requirements (recommended environment) are listed below. Devices other than those listed below that meet equivalent performance requirements can also be used.

- PC with Windows® 11
- J-Link debugger (J-Link Base, etc.) manufactured by SEGGER Corporation and USB cable or flat cable for connection
- HY0023 evaluation board and USB cable for connection
- Smartphone (for Bluetooth® Low Energy operation check)

3 Development Environment

As mentioned above, the HY0021 mounted on the HY0023 has a built-in Bluetooth® IC (nRF52805) manufactured by Nordic Semiconductor, so the following two software development environments are available (free of charge).

- nRF5 SDK
- nRF Connect SDK

Please review the license of each SDK and choose one based on your current holding environment and usage history.

3.1 Building the nRF5 SDK Development Environment

The nRF5 SDK has not been maintained since the last v17.1.0 release in 2021. Since there was no support for nRF52805 at that time, it was later released by Nordic Semiconductor as a patched v17.1.0 from the link below. In addition, Softdevice, the Bluetooth® Low Energy core stack used in the nRF5 SDK, is Bluetooth® Ver.5.2 compliant.

1. Installing the nRF5 SDK
   Download and unzip the following and extract it to an arbitrary location.
   https://devzone.nordicsemi.com/cfs-file/__key/communityserver-discussions-components-files/4/5751.nRF5_5F00_SDK_5F00_17.1.0_5F00_patched.zip

   Note: The link to the above patch environment is based on Nordic's comment in the following URL.
   https://devzone.nordicsemi.com/f/nordic-q-a/101411/secured-bootloader-uart-mbr-pca10040-for-nrf52805?ReplySortBy=CreatedDate&ReplySortOrder=Descending

2. Installing the SEGGER Embedded Studio
   Download and install v5.66 from the Windows 64bit "Embedded Studio for ARM (Legacy)" installer at the following URL.
   https://www.segger.com/downloads/embedded-studio/

3.2 Building the nRF Connect SDK Development Environment

The nRF Connect SDK environment also supports Bluetooth® Ver.5.4 and new features such as Mesh. If you are planning to build a new development environment, we recommend that you use this environment.

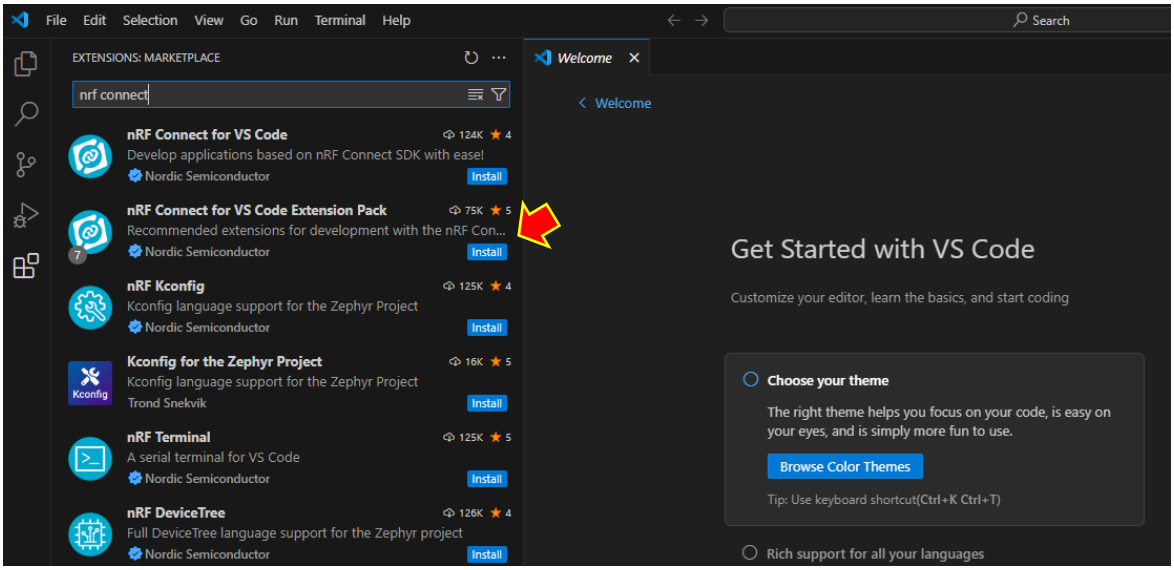
1. nRF Command Line Tools
   Download and install the latest version of nRF Command Line Tools from the following URL.
   nRF Command Line Tools - Downloads - nordicsemi.com

2. Visual Studio Code
   Download and install the latest version of Visual Studio Code (VS Code) from the following URL.
   Download Visual Studio Code - Mac, Linux, Windows

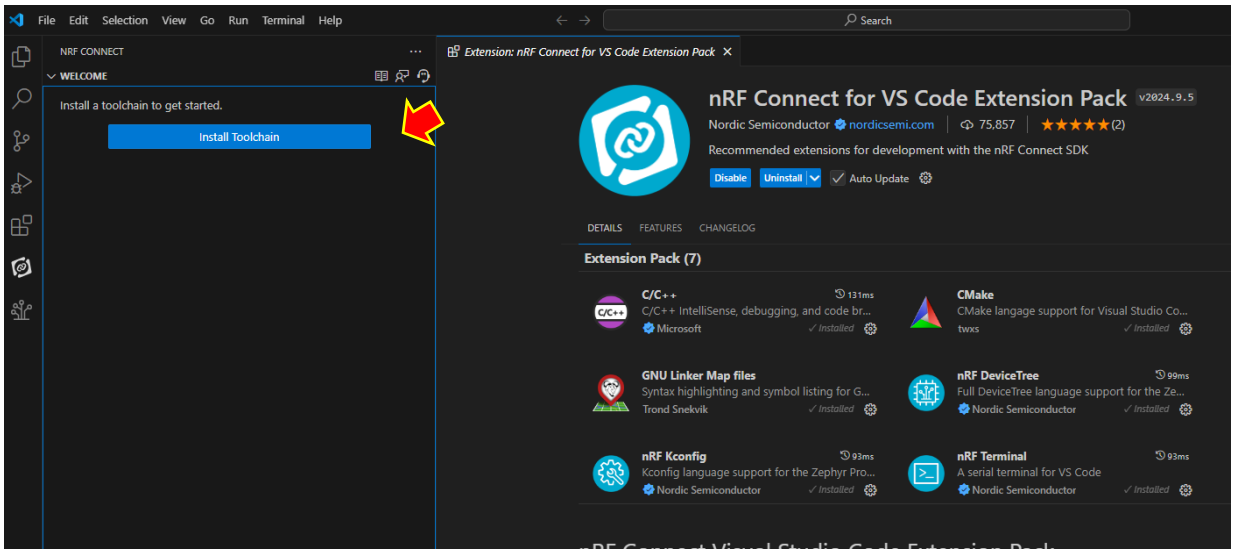
3. nRF Connect SDK

   Note: The following installation procedure can be referred to the installation video (Installing nRF Connect SDK and nRF Connect for VS Code : Get Started with nRF Connect SDK - nordicsemi.com) by Nordic.

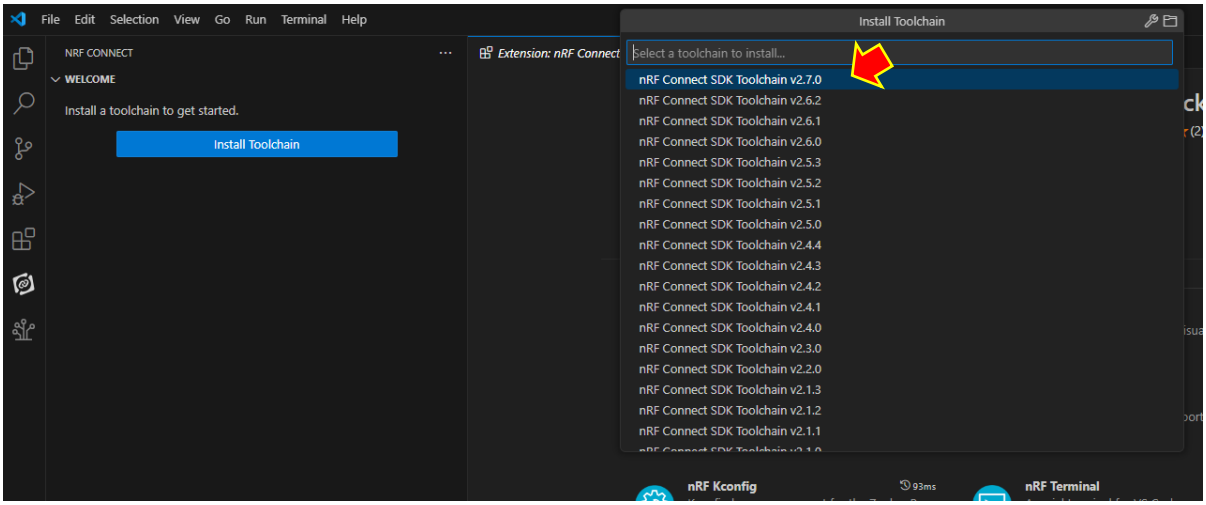
(1) Launch VS Code and click on Extension in the left pane. Type “nrf connect” in the search box and click the Install button for the nRF Connect for VS Code Extension Pack.



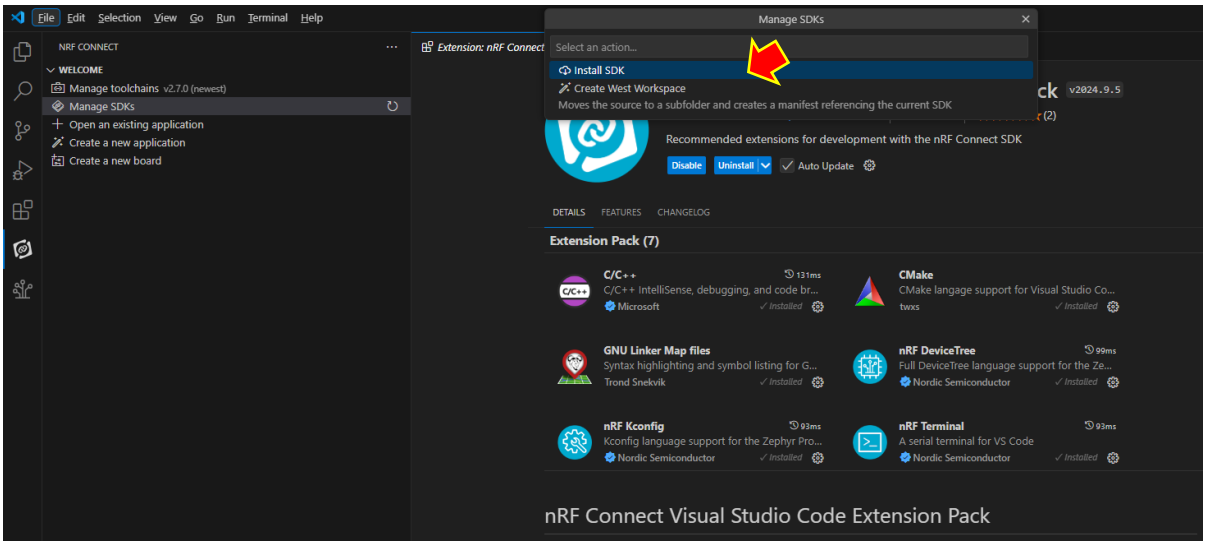
(2) After the installation is complete, click the nRF Connect icon in the left pane, and then click the Install Toolchain button.



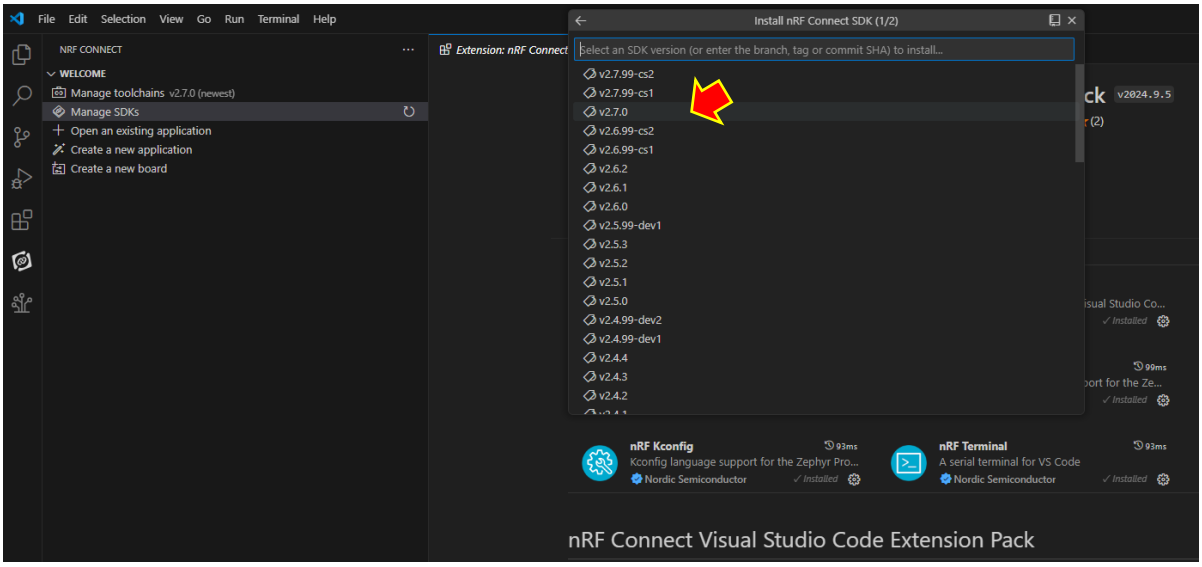
(3) Click the latest version (e.g., nRF Connect SDK Toolchain v2.7.0) when the menu appears.



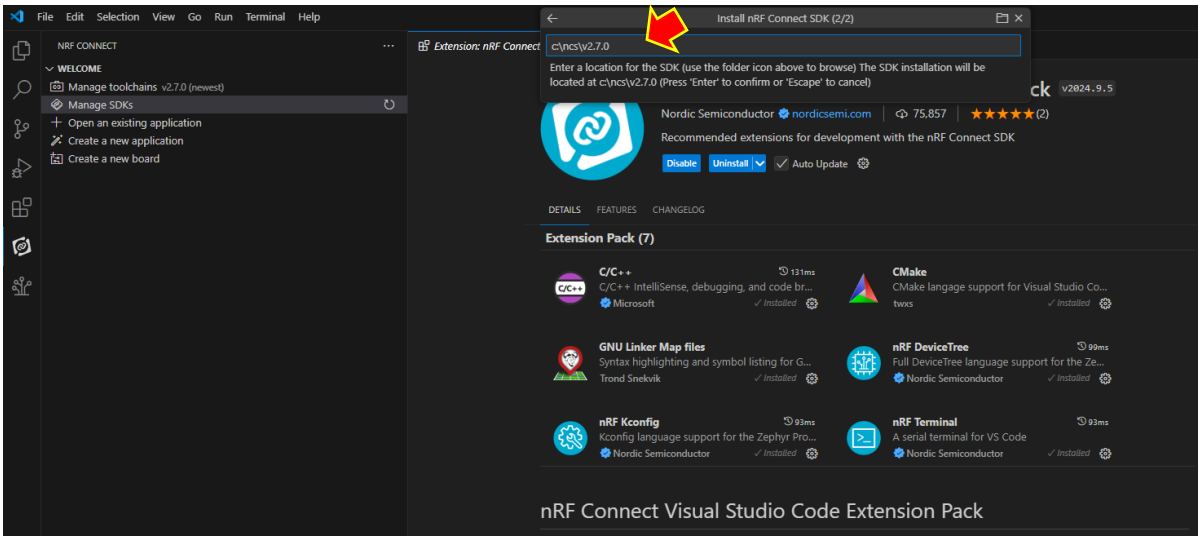
(4) After the installation is complete, click “Manage SDKs”, and then click “Install SDK”, which is displayed in the center of the screen.



(5) When the version is displayed, select the latest tag release version. In this case, select “v2.7.0”.



(6) Select the installation location. The default installation location is left as it is here.



3.3 Other necessary software

HY0023 is equipped with a USB-UART converter IC (FT232RNQ) manufactured by FTDI for UART communication, so the following USB driver must be installed beforehand.

・Driver for FT232R
https://ftdichip.com/drivers/d2xx-drivers
Please download the driver from the setup executable link in the URL above.

4 Project Example

4.1 Implementation – nRF5 SDK

Note: The following description assumes that the SDK installation folder has been extracted and placed in “c:¥”.

4.1.1 Sample code

This document describes the sample code “ble_app_uart” for checking UART communication via Bluetooth® Low Energy as an example.

◆ Open IDE-Project
Start SEGGER Embedded Studio, select the following file from the menu “File” -> “Open Solution” and open it.

C:¥5751.nRF5_SDK_17.1.0_patched¥examples¥ble_peripheral¥ble_app_uart¥pca10040e¥s112¥ses¥ble_app_uart_pca10040e_s112.emProject

4.1.2 SoftDevice

When using SoftDevice as Bluetooth® protocol stack, “S112 SoftDevice v7.2.0” for HY0021 (nRF52805) is stored in the SDK path below.

C:¥5751.nRF5_SDK_17.1.0_patched¥components¥softdevice¥s112¥hex

Versions of SoftDevice other than v7.2.0, such as “S112 SoftDevice v7.3.0” can be obtained from Nordic's URL below. (In this document, “v7.2.0” is used as is.)

S112 Downloads - nordicsemi.com

Modify the description in the file “*ble_app_uart_pca10040e_s112.emProject*” based on the version of SoftDevice to be used.

```
                .
                .
macros="CMSIS_CONFIG_TOOL=../../../../../../external_tools/cmsisconfig/CMSIS_Configuration_Wizard.jar"
debug_register_definition_file="../../../../../../modules/nrfx/mdk/nrf52810.svd"
debug_additional_load_file="../../../../../../components/softdevice/s112/hex/s112_nrf52_7.2.0_softdevice.hex"
debug_start_from_entry_point_symbol="No"
gcc_debugging_level="Level 3"          linker_output_format="hex"
                .
                .
                .
```

4.1.3 Source code modification

Modify each file with the following contents.

・ pca10040.h （*C:¥5751.nRF5_SDK_17.1.0_patched¥components¥boards*）

Change the pin number for UART sending/receiving to match the specification of the HY0023 evaluation board.

```
//#define RX_PIN_NUMBER  8
#define RX_PIN_NUMBER  16            // for HY0021
//#define TX_PIN_NUMBER  6
#define TX_PIN_NUMBER  14            // for HY0021
```

・ nrf_radio.h （*C:¥5751.nRF5_SDK_17.1.0_patched¥modules¥nrfx¥hal*）

Fix the Nordic code bug based on the comments in the following URL thread.
https://devzone.nordicsemi.com/guides/short-range-guides/b/getting-started/posts/developing-for-the-nrf52805-with-nrf5-sdk

(Before modify)

```
#if defined(RADIO_PDUSTAT_PDUSTAT_Msk)
__STATIC_INLINE uint8_t nrf_radio_pdustat_get(void)
{
    return (uint8_t)(NRF_RADIO->PDUSTAT & RADIO_PDUSTAT_PDUSTAT_Msk);
}

__STATIC_INLINE uint8_t nrf_radio_cistat_get(void)
{
    return (uint8_t)((NRF_RADIO->PDUSTAT & RADIO_PDUSTAT_CISTAT_Msk) >>
RADIO_PDUSTAT_CISTAT_Pos);
}
#endif // defined(RADIO_PDUSTAT_PDUSTAT_Msk)
```

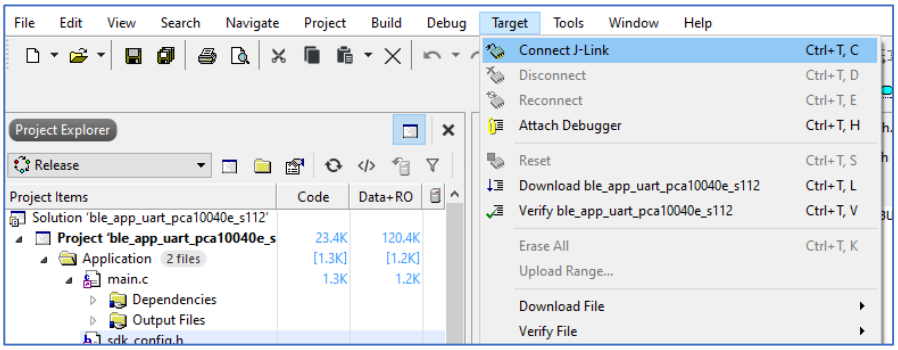
(After modify)

```
#if defined(RADIO_PDUSTAT_PDUSTAT_Msk)
__STATIC_INLINE uint8_t nrf_radio_pdustat_get(void)
{
    return (uint8_t)(NRF_RADIO->PDUSTAT & RADIO_PDUSTAT_PDUSTAT_Msk);
}
#endif // defined(RADIO_PDUSTAT_PDUSTAT_Msk)

#if defined(RADIO_PDUSTAT_CISTAT_Msk)
__STATIC_INLINE uint8_t nrf_radio_cistat_get(void)
{
    return (uint8_t)((NRF_RADIO->PDUSTAT & RADIO_PDUSTAT_CISTAT_Msk) >>
RADIO_PDUSTAT_CISTAT_Pos);
}
#endif // defined(RADIO_PDUSTAT_CISTAT_Msk)
```

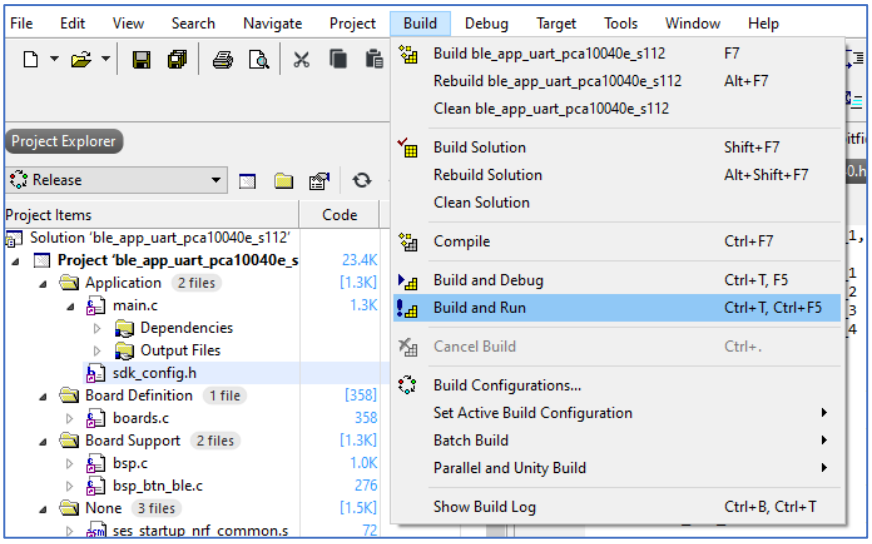
4.1.4 Connect Debugger

Connect the evaluation board to the J-Link debugger and perform “Connect J-Link” in the IDE.



4.1.5 Build and Run

Correct/update the files up to the previous section, confirm the connection, and execute “Build and Run” from the Build menu.



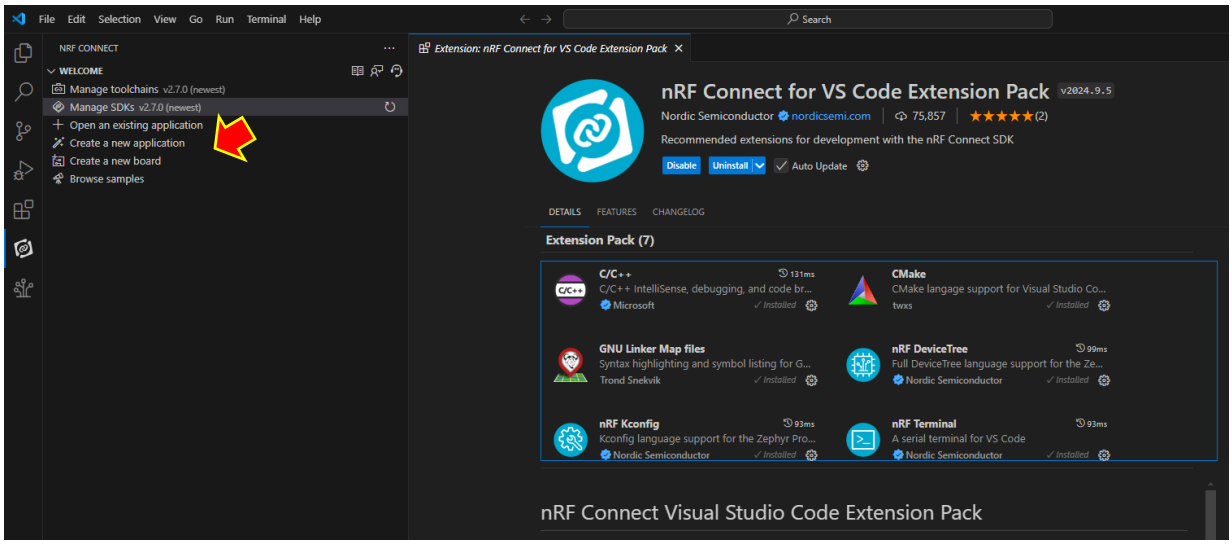
It is also possible to write the FW (.hex file) generated by the above build to the evaluation board by using the “Programmer” in the “nRF Connect for Desktop Tool”, which can be downloaded from nRF Connect for Desktop - Downloads - nordicsemi.com.

4.2 Implementation – nRF Connect SDK

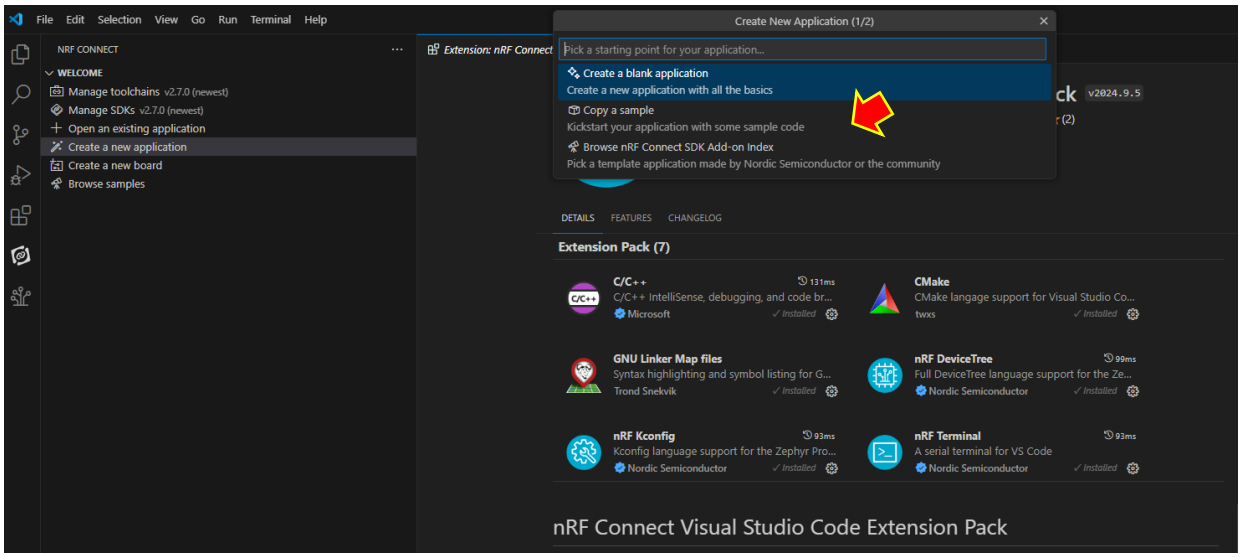
4.2.1 Sample code Build

The following shows the build to write procedure using “Bluetooth Peripheral UART” as an example to test the Bluetooth® Low Energy UART communication provided by the nRF Connect SDK.

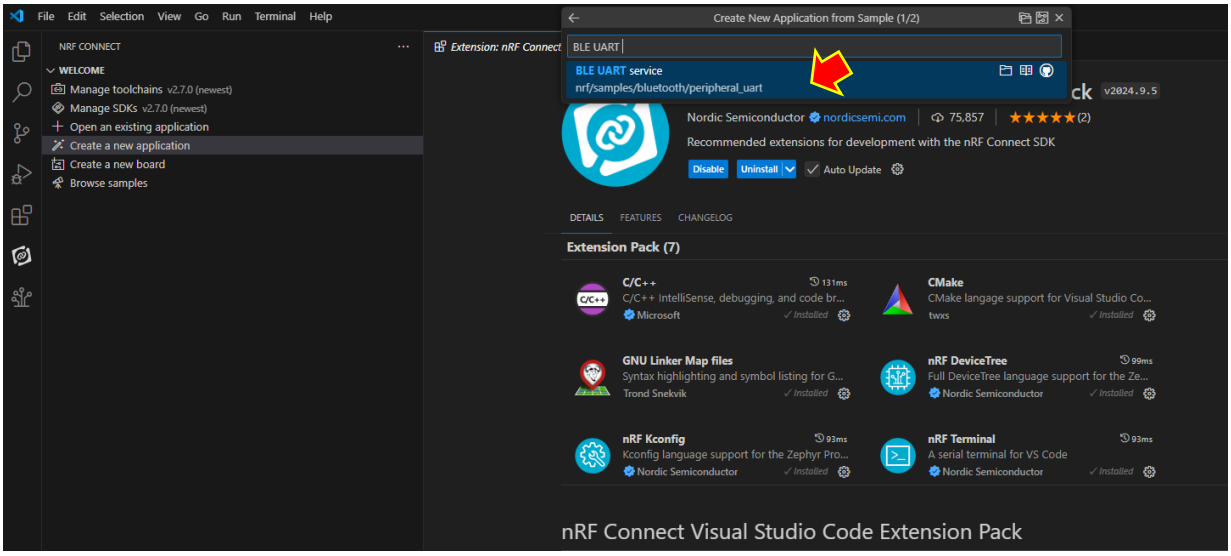
(1) Click “Create a new application” in the left pane of VS Code.



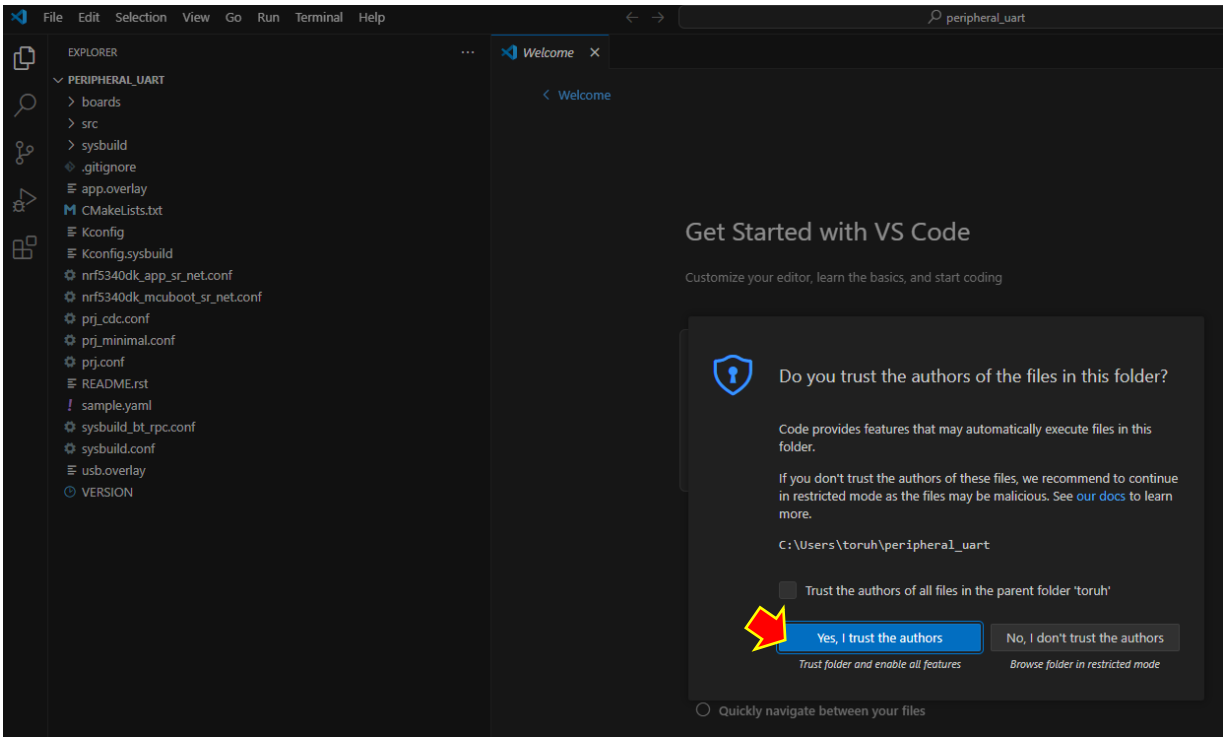
(2) Select “Copy a sample”.



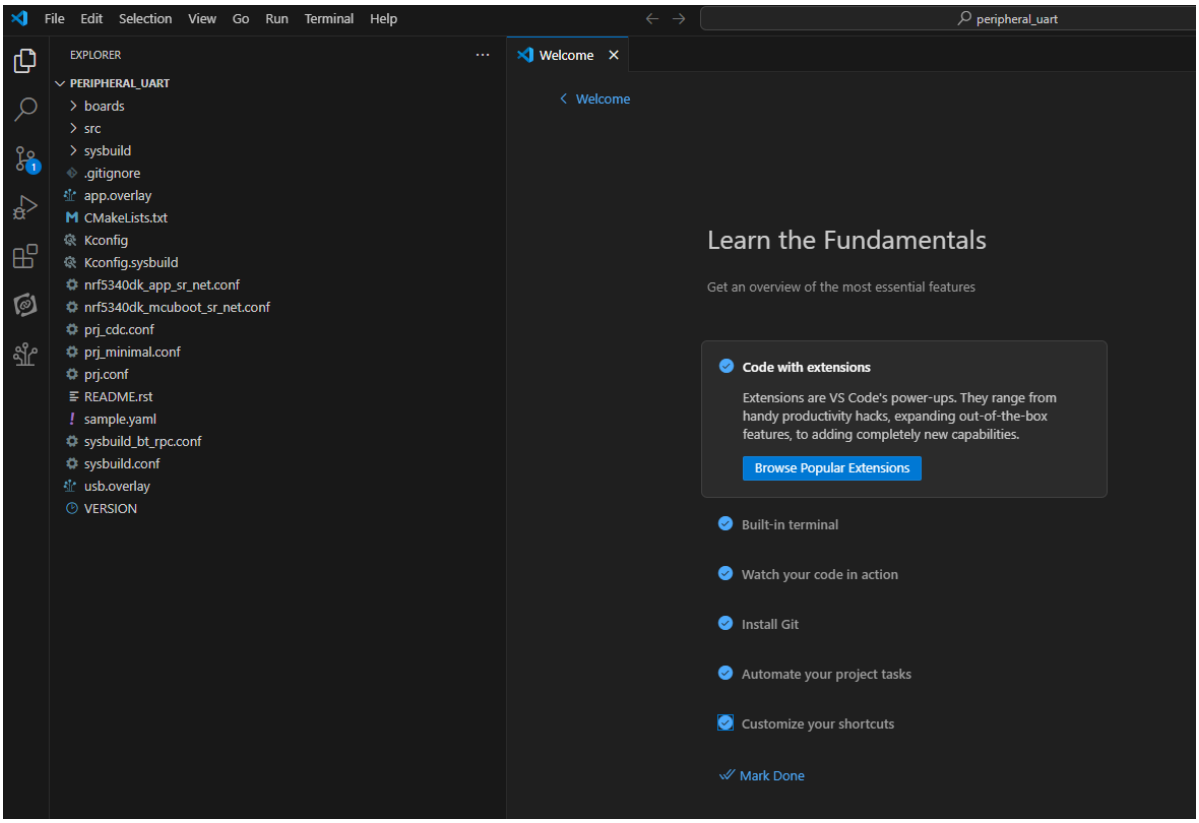
(3) Type “BLE UART” in the input text box and select “BLE UART service” as a candidate.



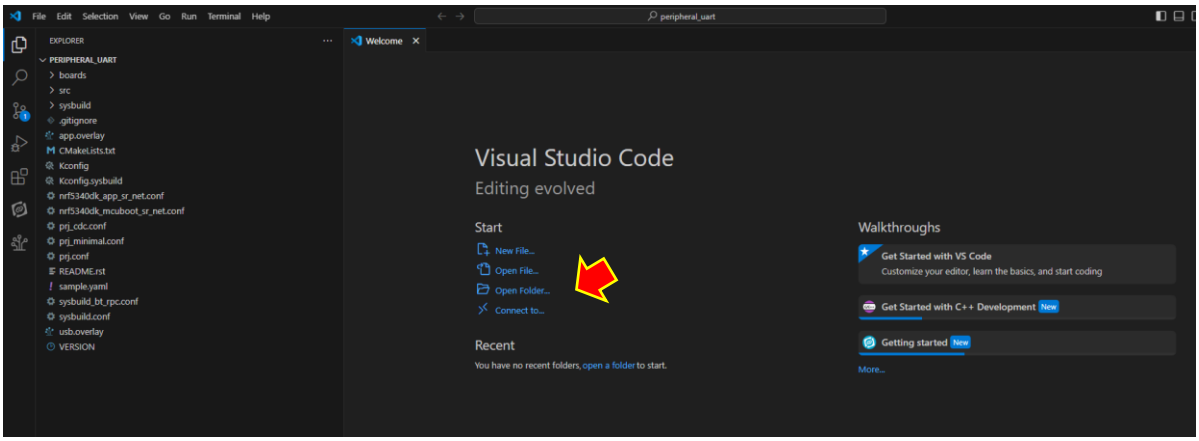
(4) Select "Yes" when asked "Do you trust the authors of the files in this folder?".



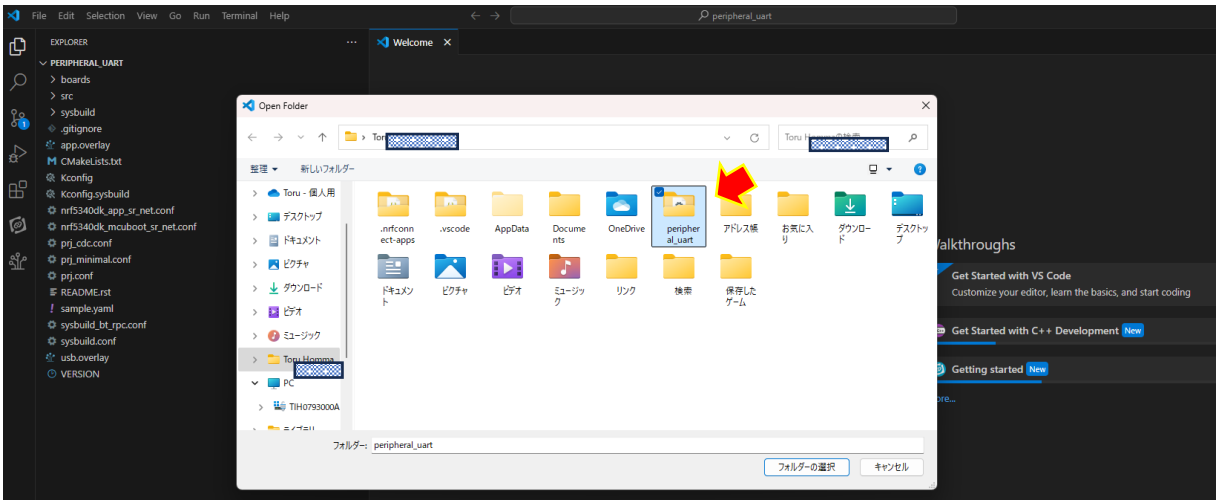
(5) Mark as appropriate and proceed.



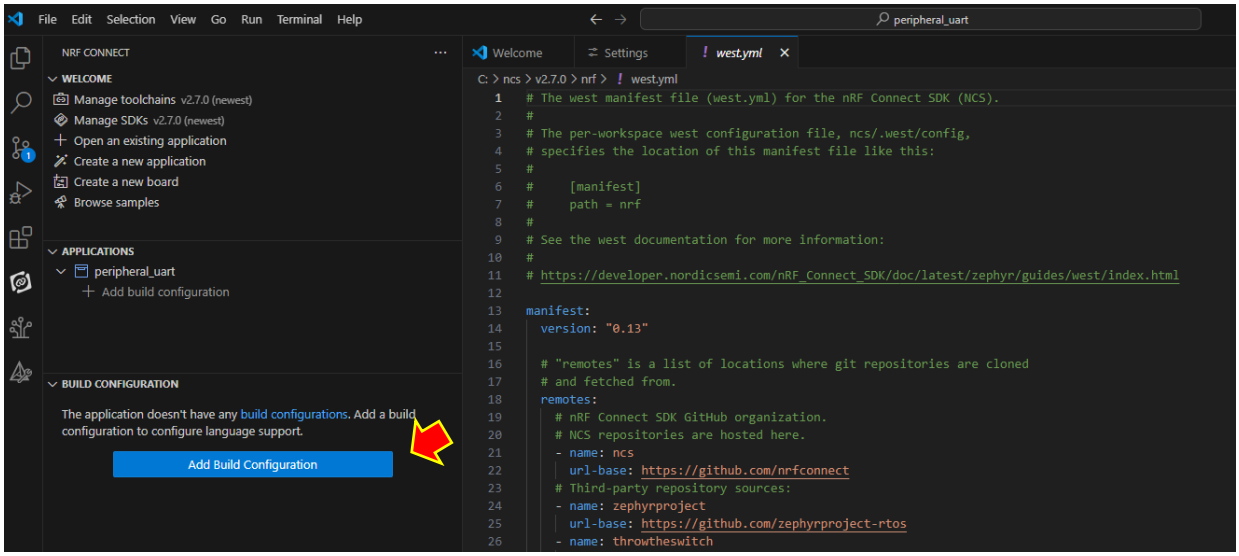
(6) Select “Open Folder...”.



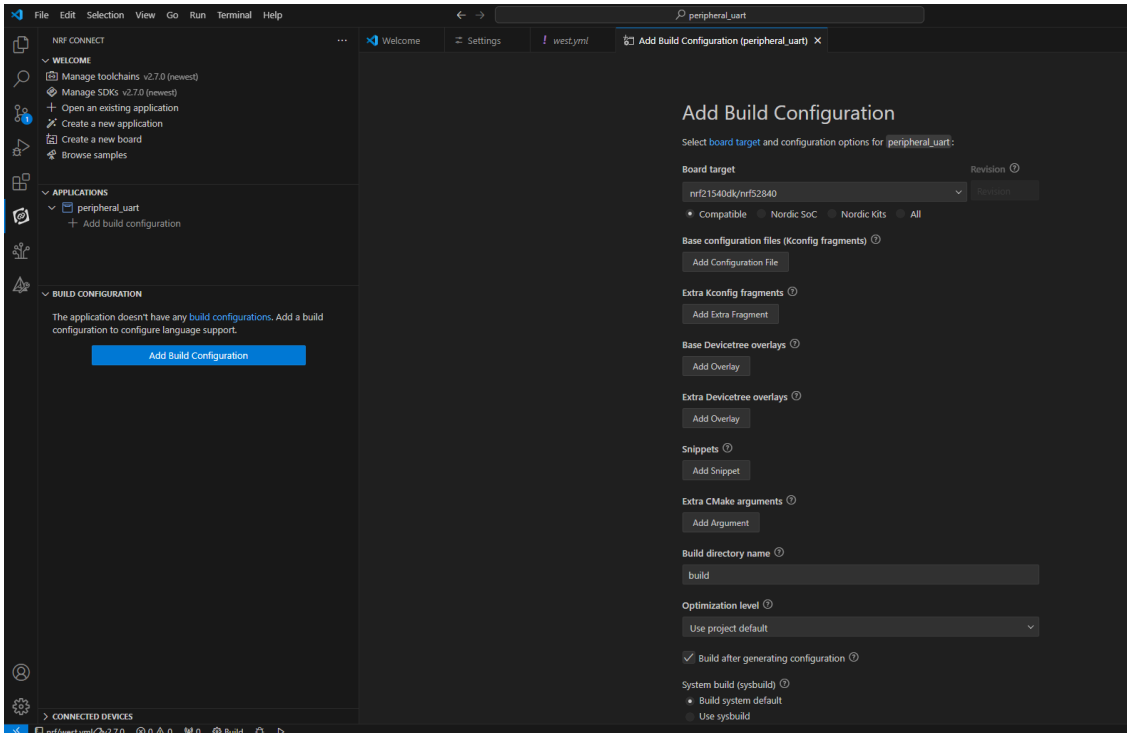
(7) Select the “Peripheral_uart” folder.



(8) Click “Add Build Configuration” in the left pane.

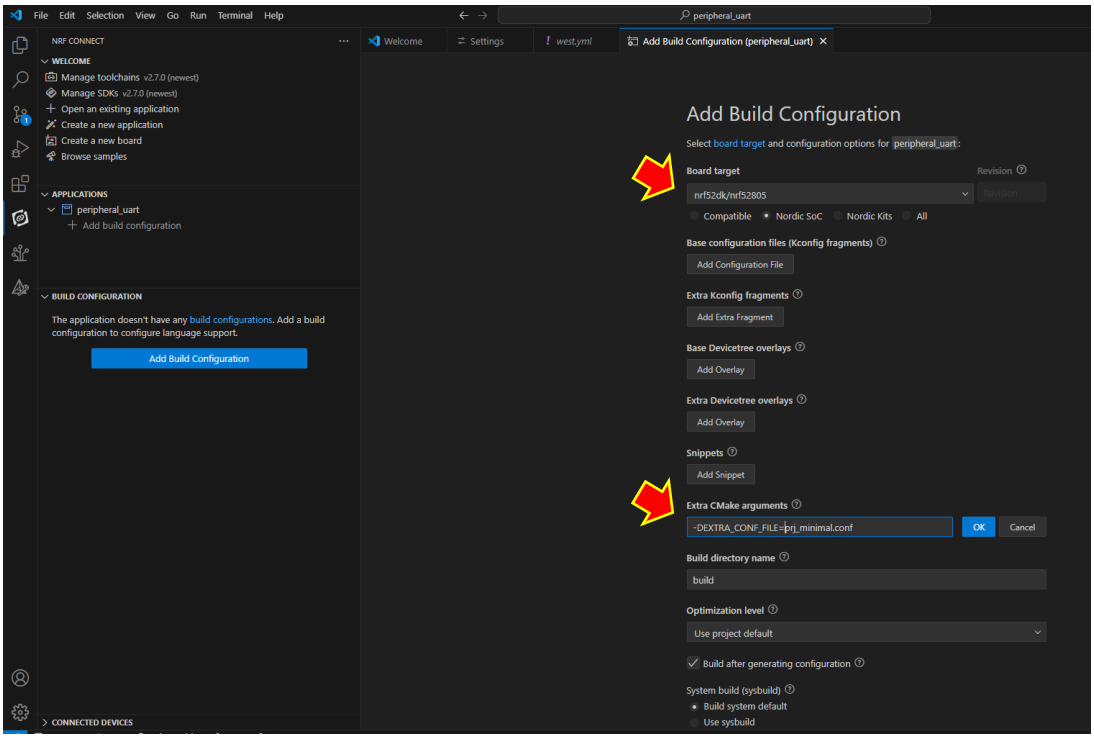


(9) The following will be displayed.



(10) Select Nordic SoC in the board target and choose “nRF52dk/nrf52805” from the pull-down menu. In addition, enter the following in the Extra CMake arguments field and press the OK button.

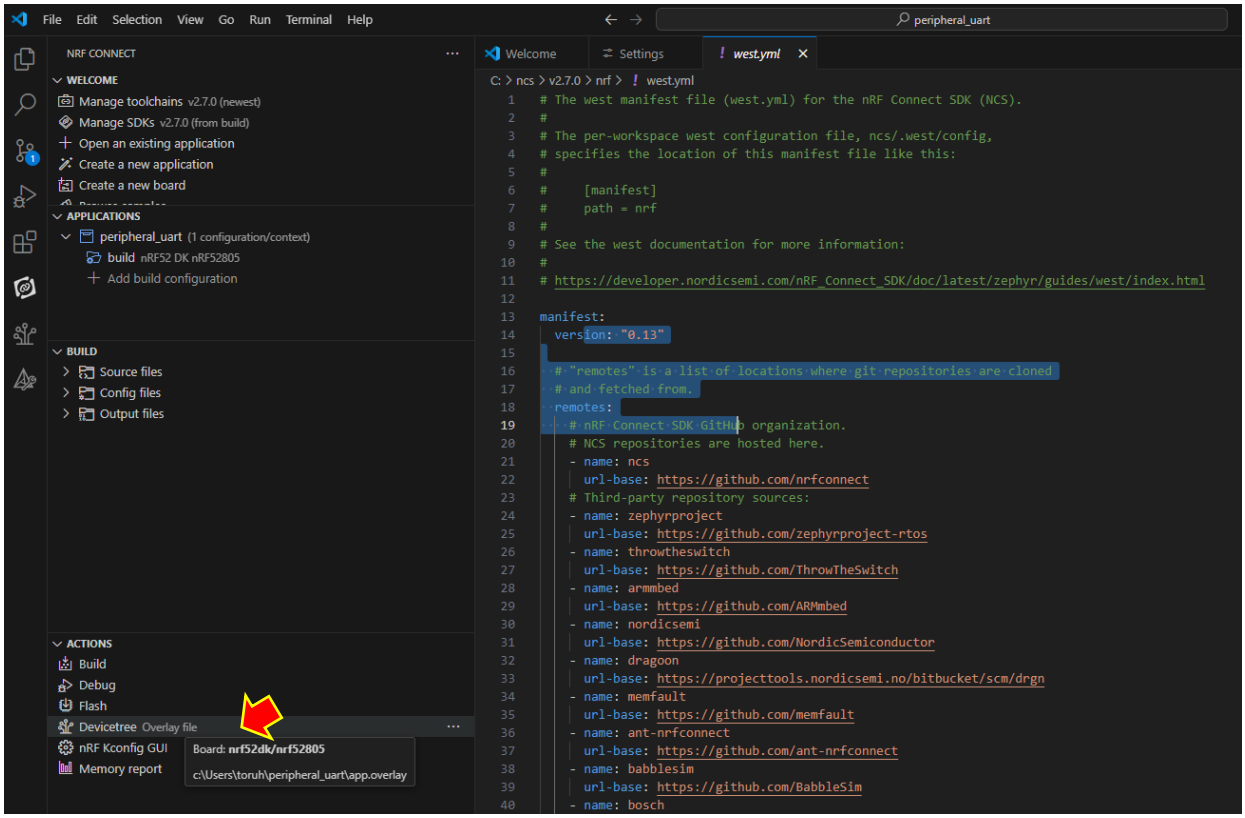
```
-DEXTRA_CONF_FILE=”prj_minimal.conf”
```



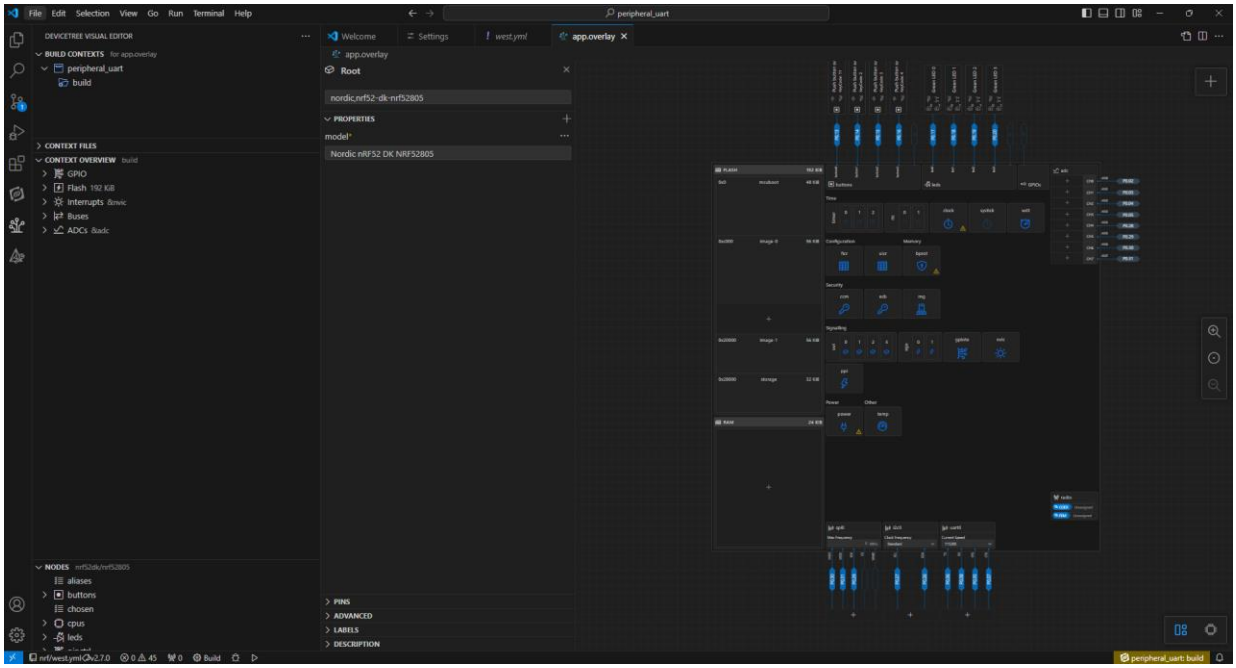
Note: The configuration of the Extra CMake arguments is based on the following. Refer to the Minimal sample variant section of the URL:https://docs.nordicsemi.com/bundle/ncs-2.7.0/page/nrf/samples/bluetooth/peripheral_uart/README.html to configure CMake settings for resource-limited cases.

To activate the optional extensions supported by this sample, set EXTRA_CONF_FILE using the respective CMake option in the following manner: For the minimal build variant, set it to prj_minimal.conf.

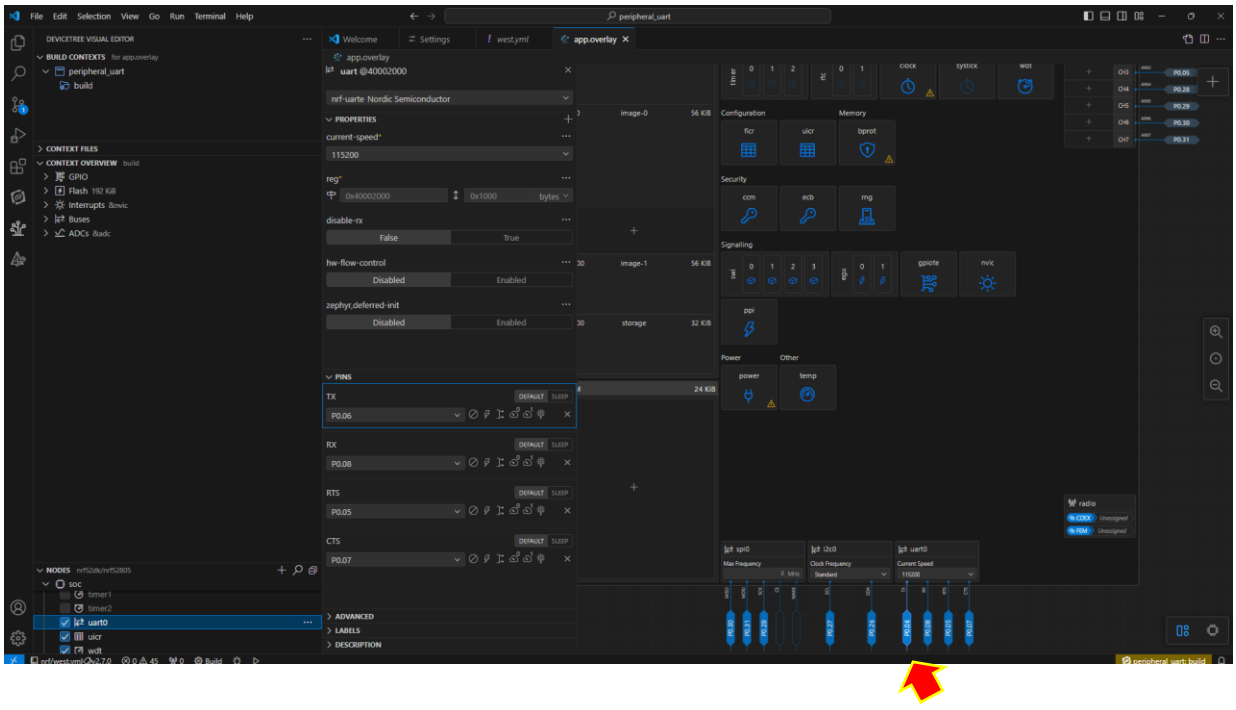
(11) Scroll down and click on Build Configuration. Then click on “Devicetree Overlay File” in the left pane to configure the UART Tx/Rx pins of the HY0023 Evaluation Board.



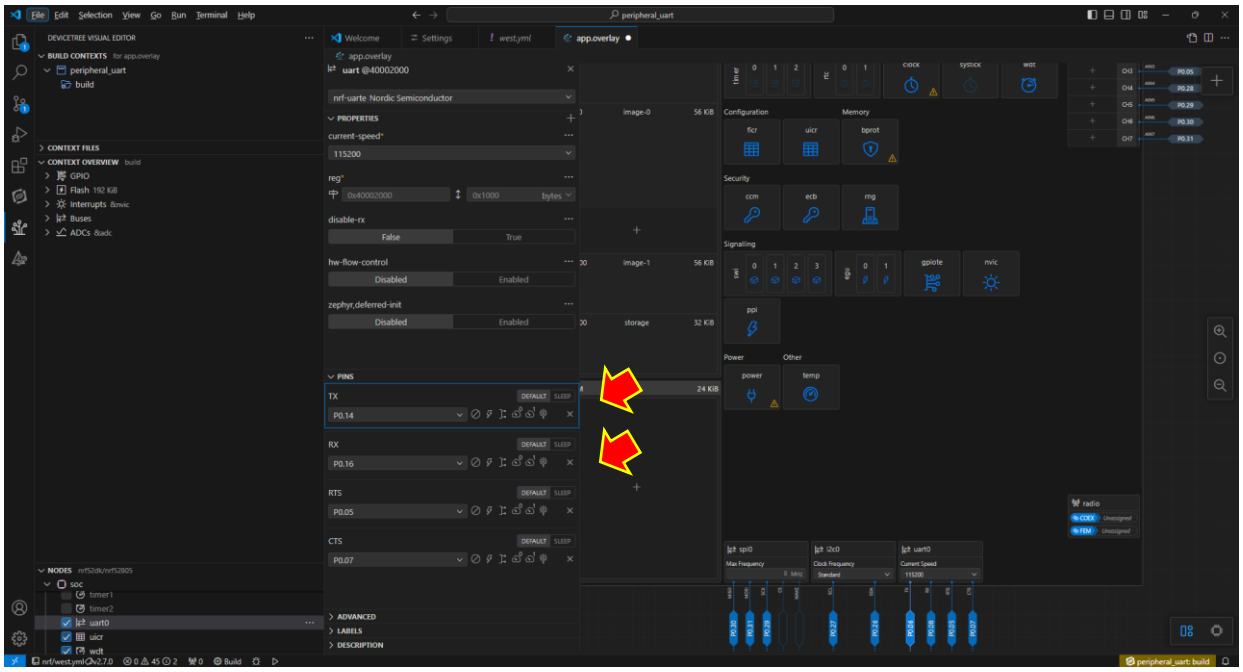
(12) The following will be displayed.



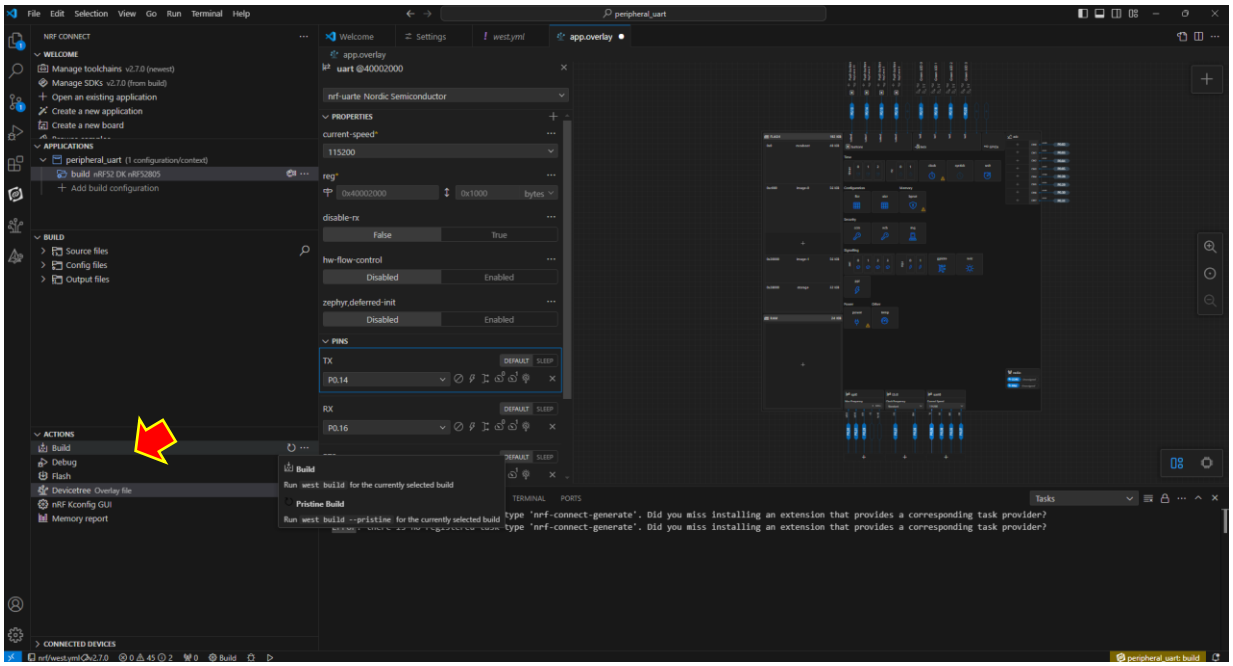
(13) Click on the uart0 pin (e.g.P0.06) in the lower right pane to display the menu for each pin in the center.



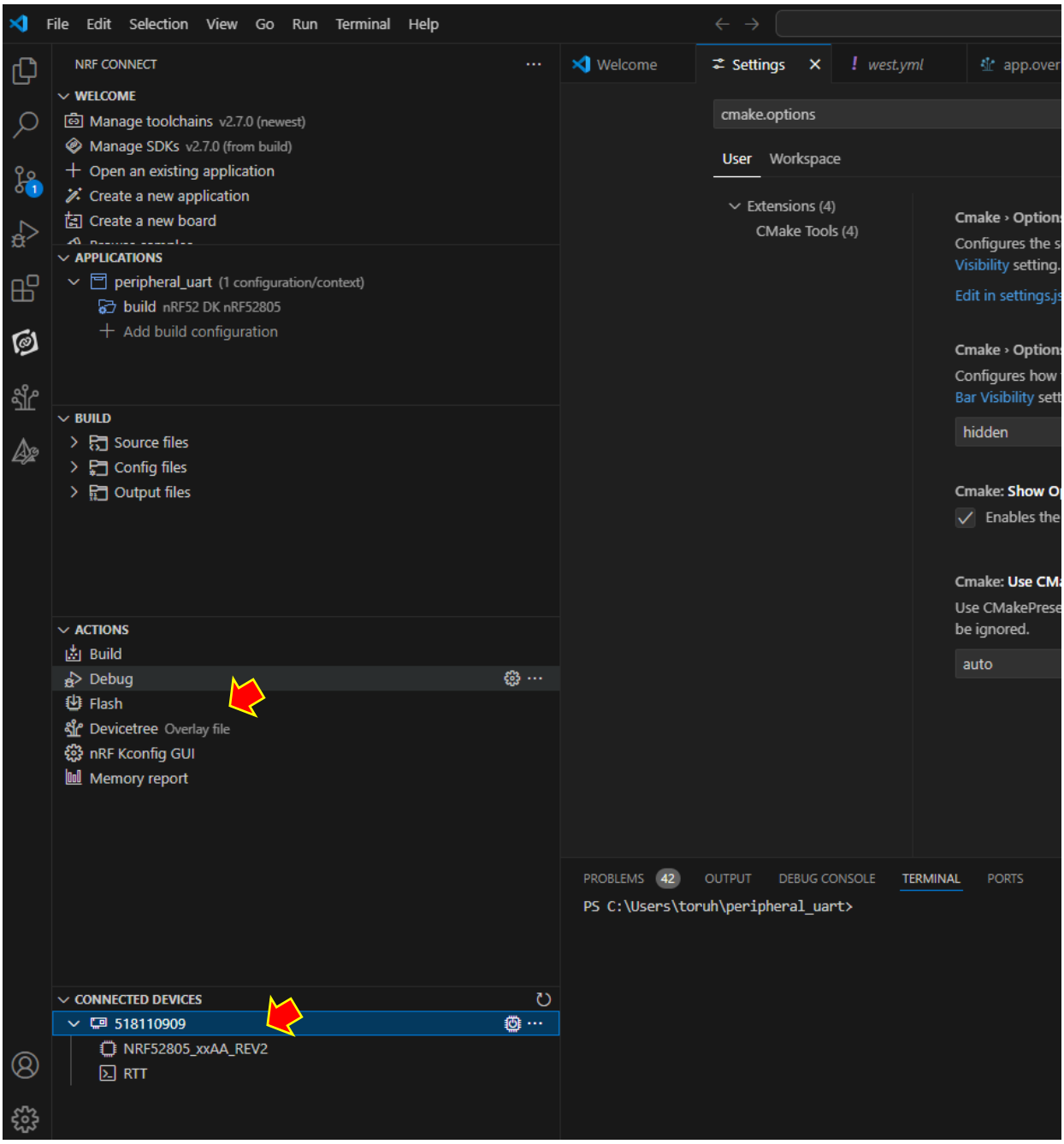
(14) Set TX and RX to P0.14 and P0.16, respectively.



(15) Click “Build” under Actions to build.



(16) Connect the J-Link Debugger and the HY0023 Evaluation Board to the PC via USB. Click on the node under CONNECTED DEVICES in the lower left corner to automatically detect the devices. Next, click on “Flash” under ACTIONS in the left pane and write FW to the evaluation board.



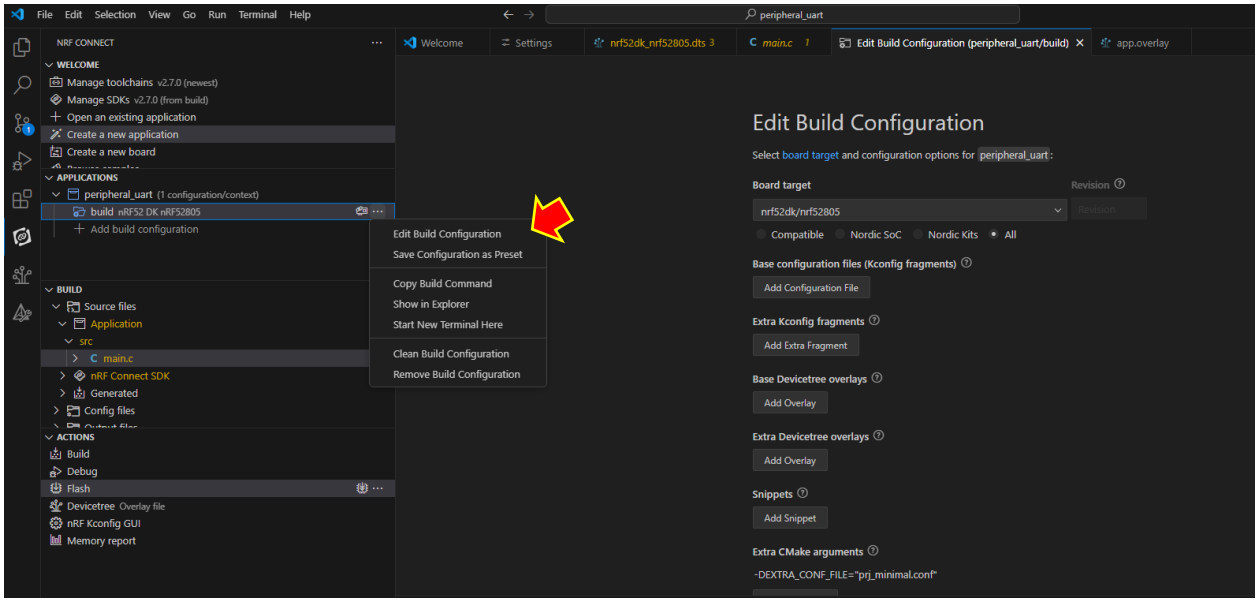
4.2.2 Debugging

Following the previous section, this section describes debugging operations using the “Bluetooth Peripheral UART” project to test the Bluetooth® Low Energy UART communication.

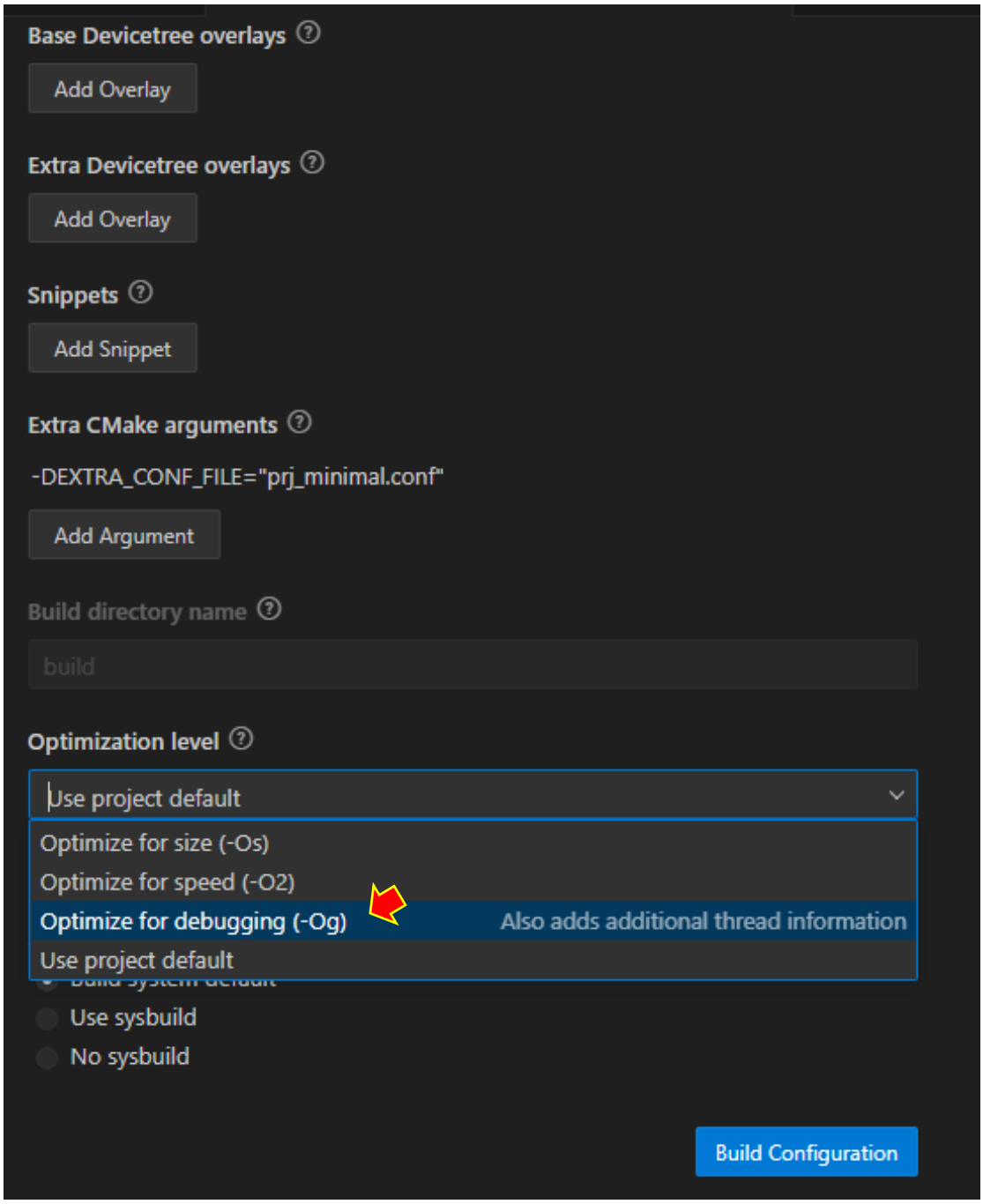
(1) To avoid problems during debugging, comment out and edit the following lines in the "prj_minimal.conf" file in the root folder of the sample project and save it.

```
CONFIG_BT_NUS_THREAD_STACK_SIZE=512
                ↓
#CONFIG_BT_NUS_THREAD_STACK_SIZE=512
```

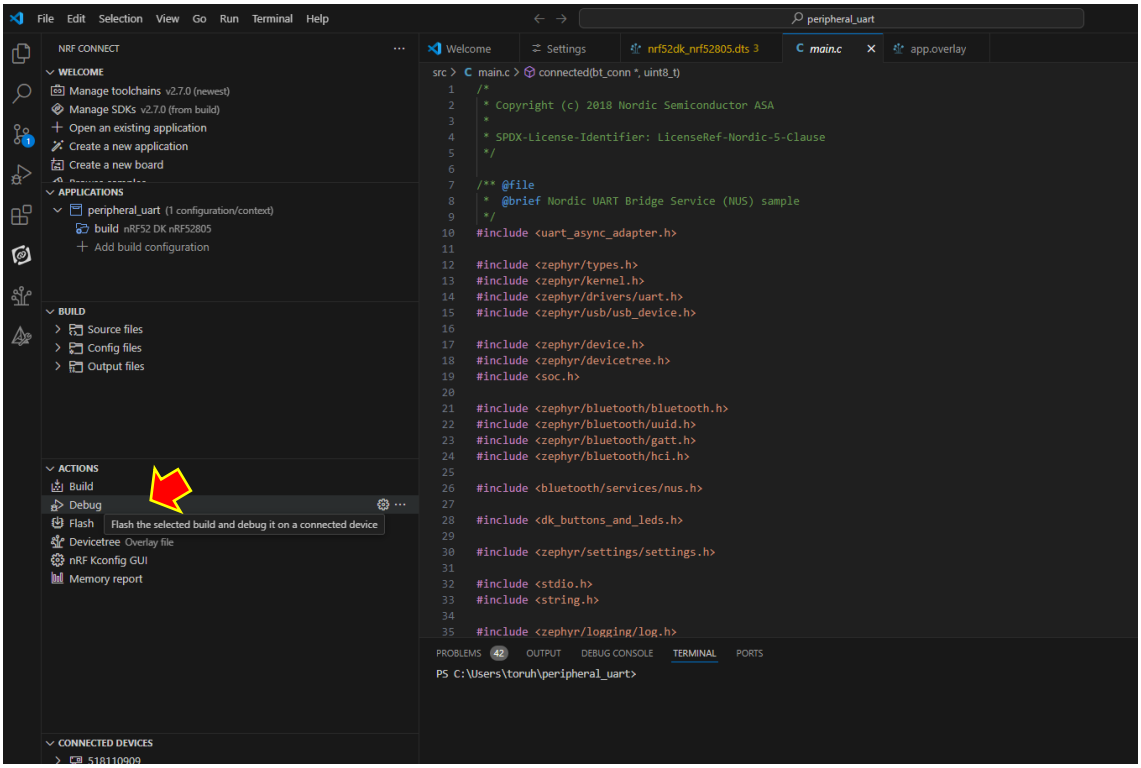
(2) Start VS Code, click “...” under build in the left pane under APPLICATIONS - peripheral_uart, and select “Edit Build Configuration”.



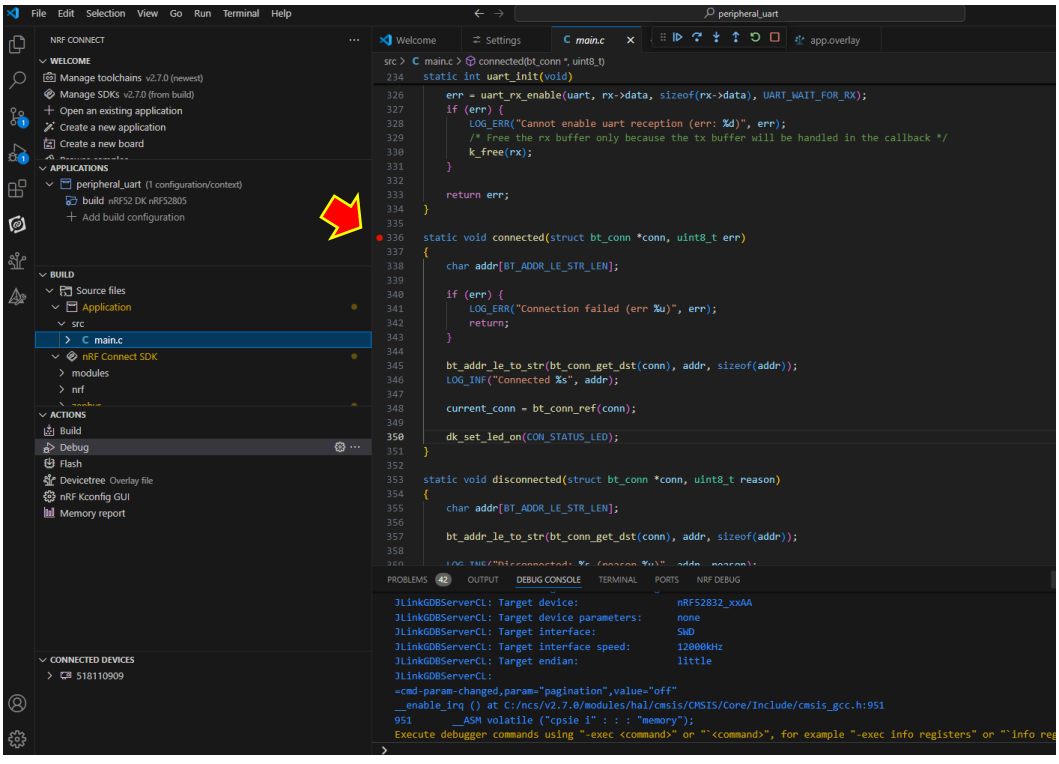
(3) Change the Optimization level to “Optimize for debugging (-Og)”.



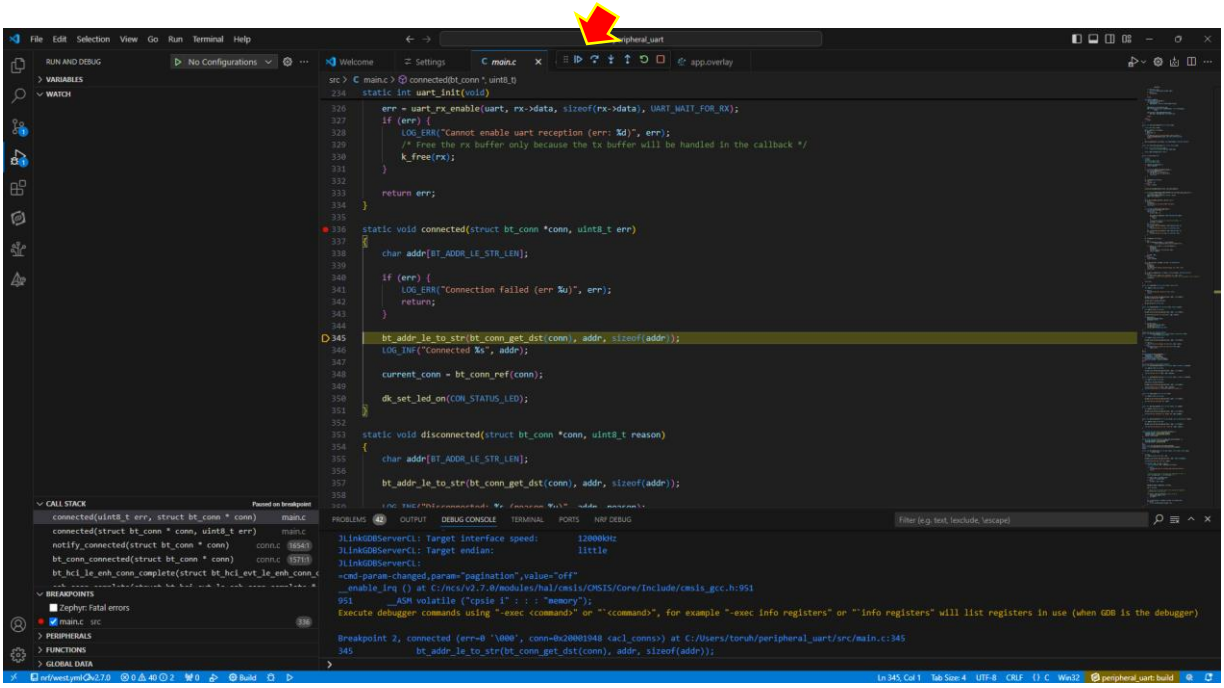
(4) Click “Debug” under ACTIONS in the left pane to start debugging.



(5) Set a temporary breakpoint. (e.g., click *static void connected(struct bt_conn *conn, uint8_t err)* in main.c to set a breakpoint)



(6) Press the play button in the upper center or F5 key to execute. When you connect to the “Nordic_UART_Service” device from your smartphone using the Nordic application (nRF Toolbox) described below, you will see that the device stops at the breakpoint as shown below.



5 Verify action

5.1 Sample Verify

The following is a confirmation of the FW operation created in each of the above nRF5 SDK/nRF Connect SDK development procedures, confirming that UART communication via Bluetooth® Low Energy can be performed between a PC and a smartphone.

5.1.1 Setup of PC

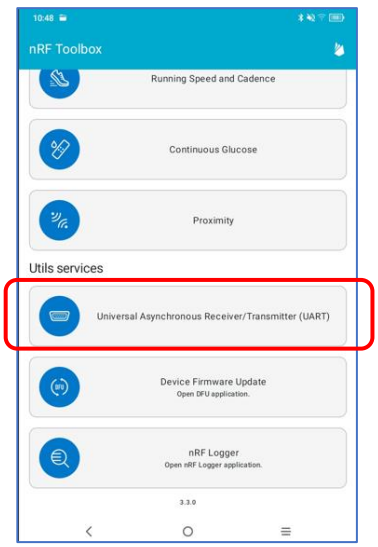
(1) Install the software for UART terminal “TeraTerm” from below.
https://teratermproject.github.io/index-en.html

(2) Start “TeraTerm” and set the COM port and baud rate of the evaluation board to 115200bps. Open the setup menu by clicking “Setup” --> “Terminal” and mark “Local Echo”.

5.1.2 Setup of Smartphone

(1) Install “nRF Toolbox” application on smartphone.

(2) Start the “nRF Toolbox” application and tap the following “UART service”.



(3) Connect to the device named “Nordic_UART_Service”.

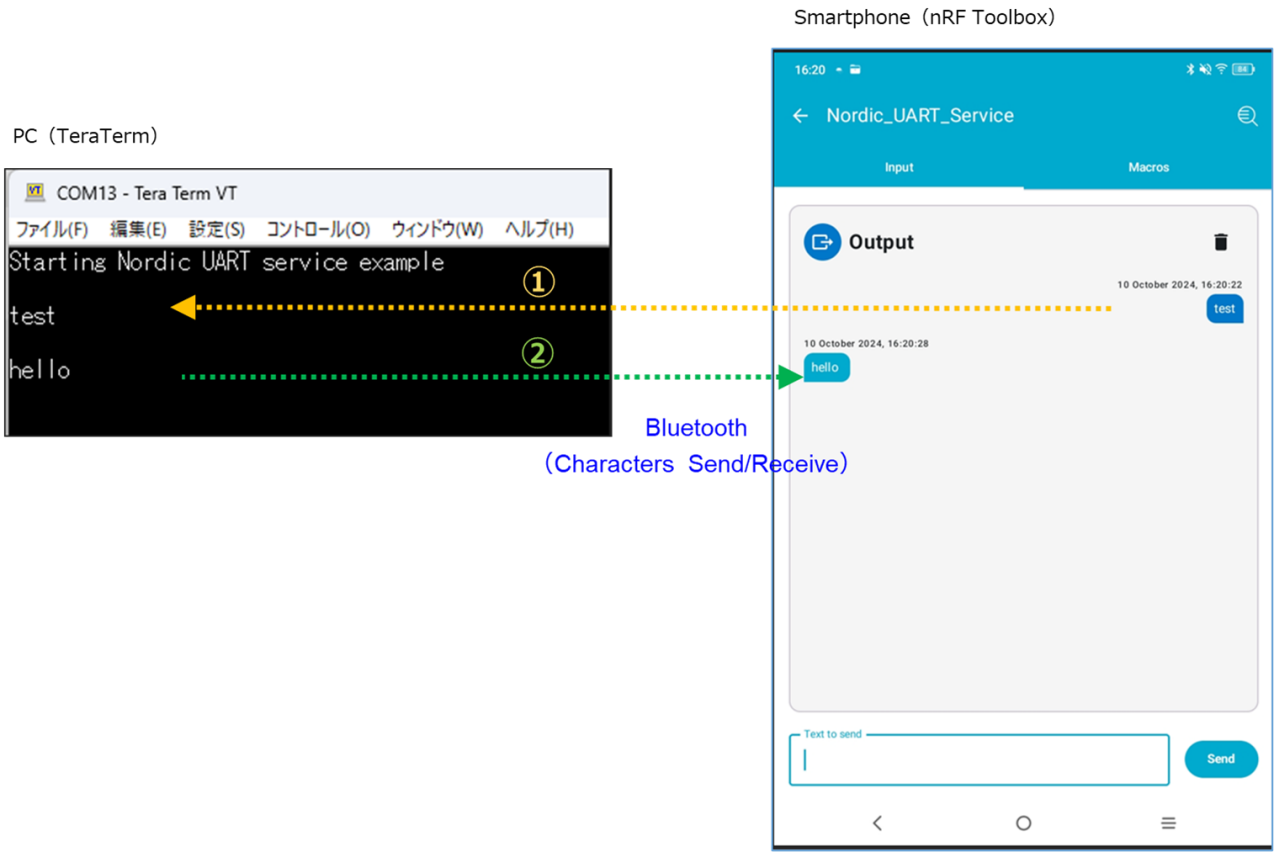
5.1.3 Bluetooth® Low Energy (UART) communication check

(1) Input text (e.g. ”test”) from smartphone and tap “Send”.

→ The same characters will be displayed on the Tera Term.

(2) Enter any character in the Tera term and press Return key. (e.g. ”hello”)

→ The same characters will be displayed on smartphone.



Note: The following describes the flow up to the start of development using the sample code. For more detailed information, please refer to the technical information provided by Nordic Semiconductor.

6 Reference

1 Nordic Semiconductor ASA, “nRF Connect SDK”, nRF Connect SDK - nordicsemi.com

2 Nordic Semiconductor ASA, “Bluetooth: Peripheral UART — nRF Connect SDK 2.7.0 documentation (nordicsemi.com)”, https://docs.nordicsemi.com/bundle/ncs-2.7.0/page/nrf/samples/bluetooth/peripheral_uart/README.html

3 Nordic Semiconductor ASA, “Debugging in nRF Connect for VS Code - Nordic Developer Academy (nordicsemi.com)”, https://academy.nordicsemi.com/courses/nrf-connect-sdk-intermediate/lessons/lesson-2-debugging/topic/debugging-in-vs-code/

4 Nordic Semiconductor ASA, “nRF52805 Product Specification”, nRF52805 Product Specification (nordicsemi.com)

7 Precautions

- The information in this document is current at the time of writing and is subject to change.
- We assume no responsibility whatsoever for any damages, incidents, or accidents caused by this material.
- Please confirm the applicable license terms and conditions when using any of the software listed in this document.
- The names of companies, products, and services mentioned in this document may be used as trademarks of their respective owners.