

EXPERIMENT – 8

INTERFACING OF FLASH MEMORY WITH DEV BOARD

What will you learn from this module:

- Interfacing with the help of SPI protocol.
- Configuration of overlay file, device tree and prj file for enabling hardware device.

Requirements:

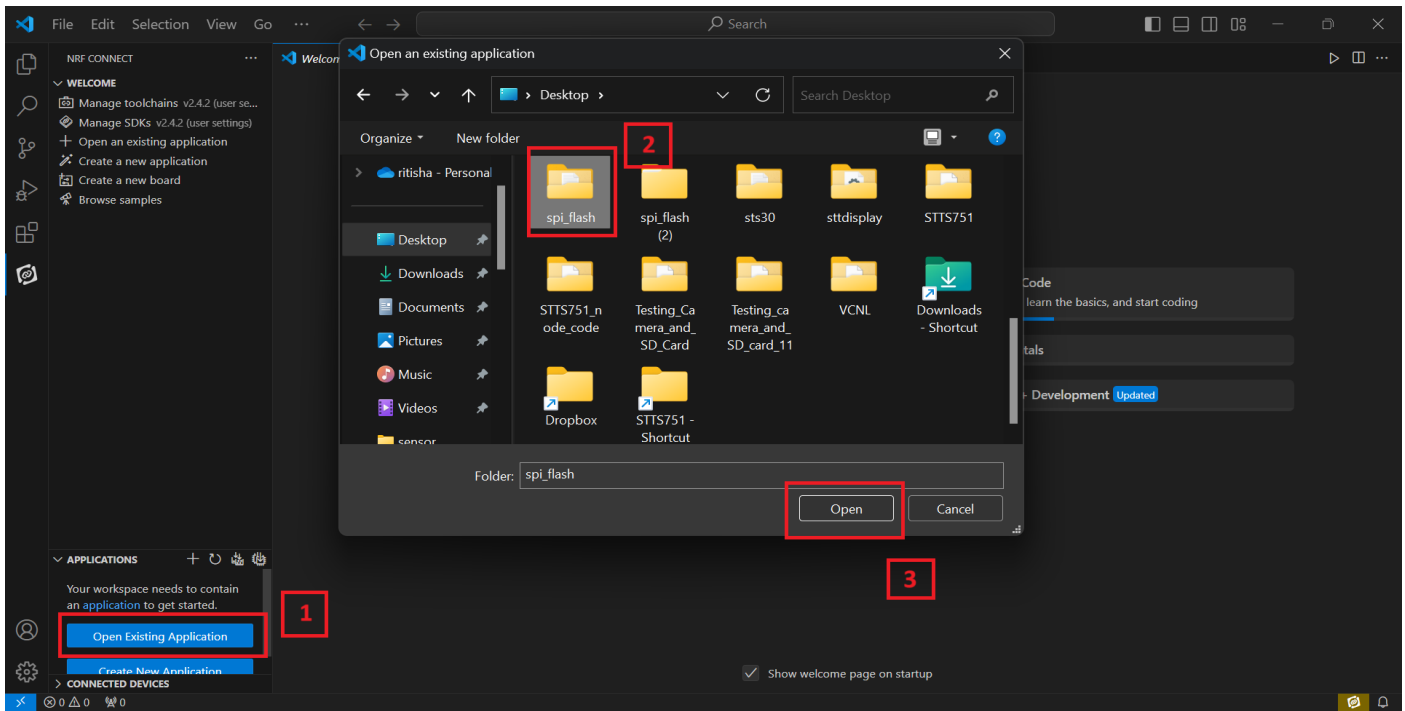
- nRF connect desktop software.
- nRF Command line tools.
- Visual studio code.
- USB cable.
- nRF52832 Development Board/Node.
- WS25Q16DV.

Prerequisites:

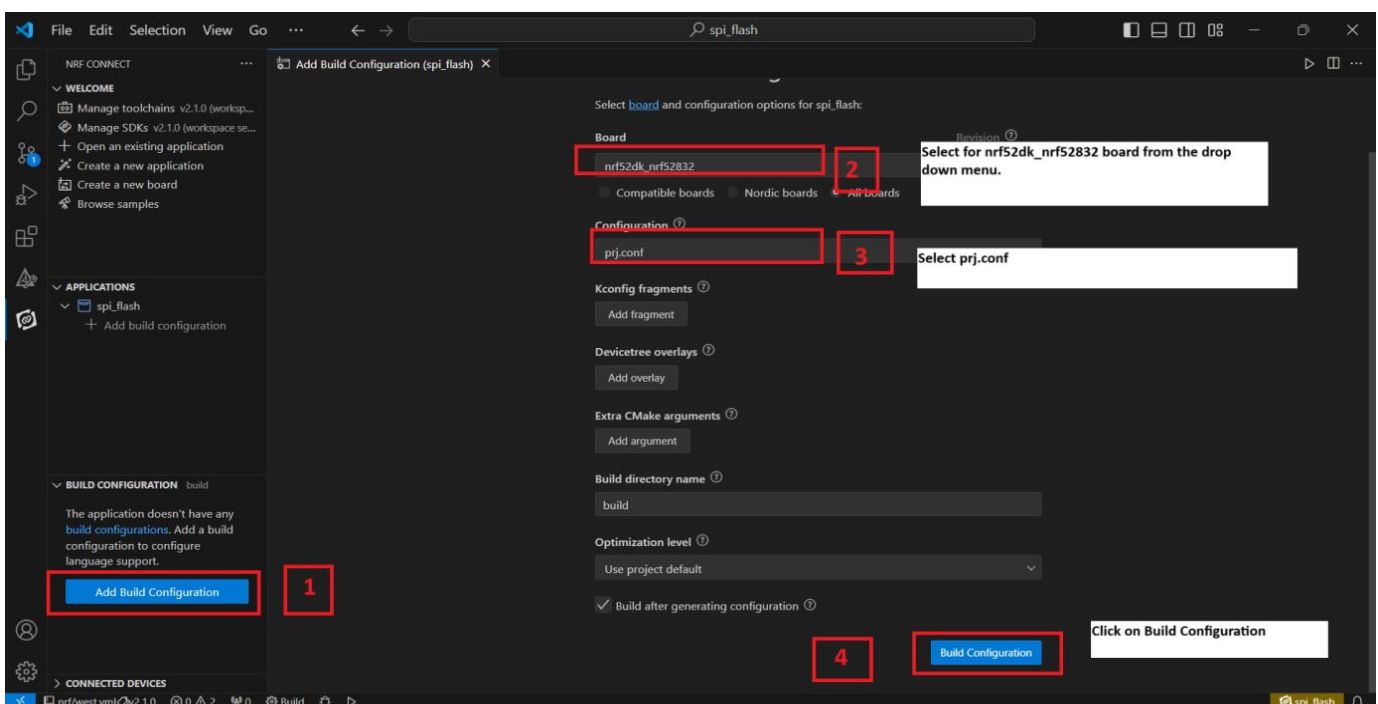
- Basic knowledge of C/C++.
- Basic knowledge of communication protocol.
- Basic project setup.

Setup and Configuration:

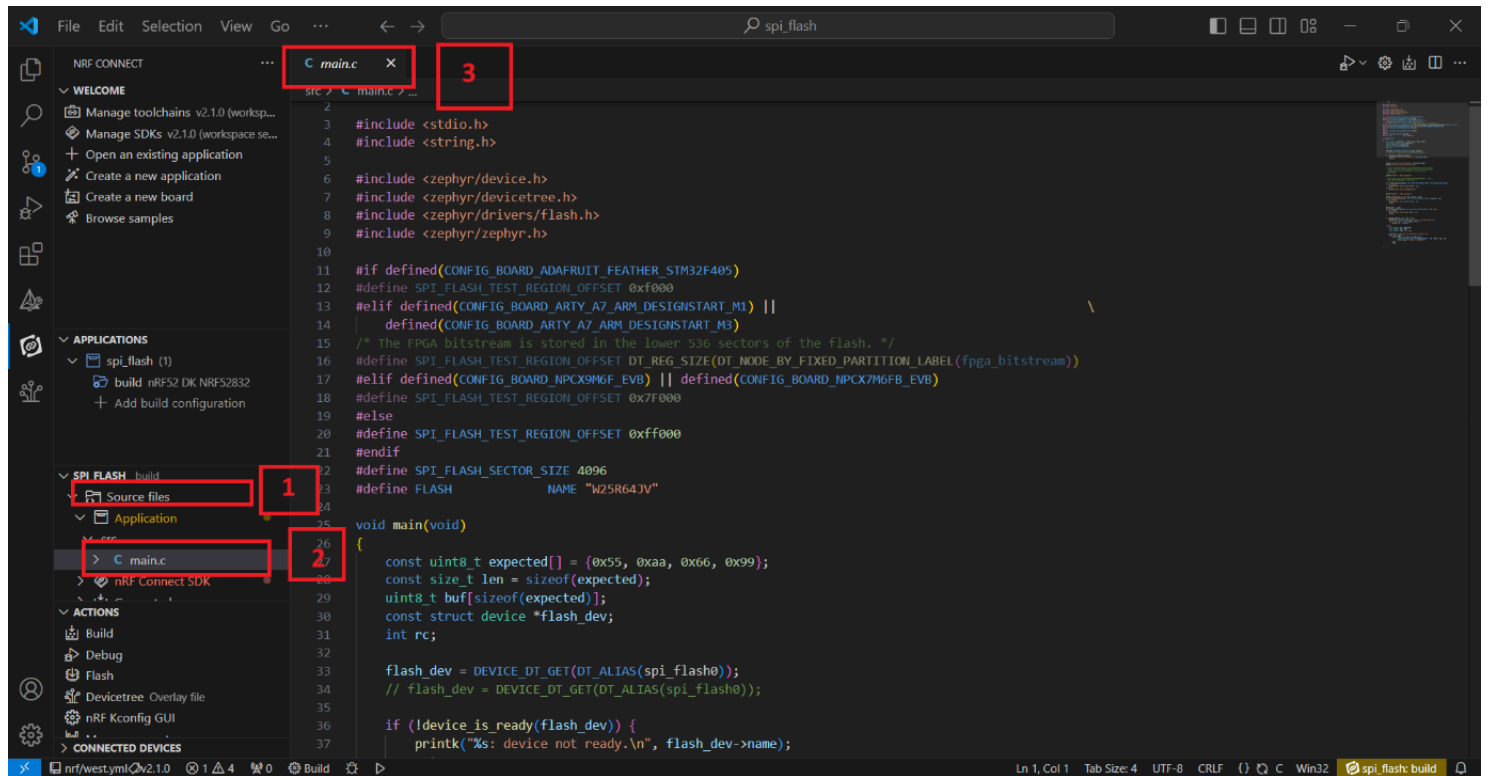
- Open VS Code and click on **Open Existing Application** [1] > click on **spi_flash** [2] > **Open** [3] as shown in the picture below.



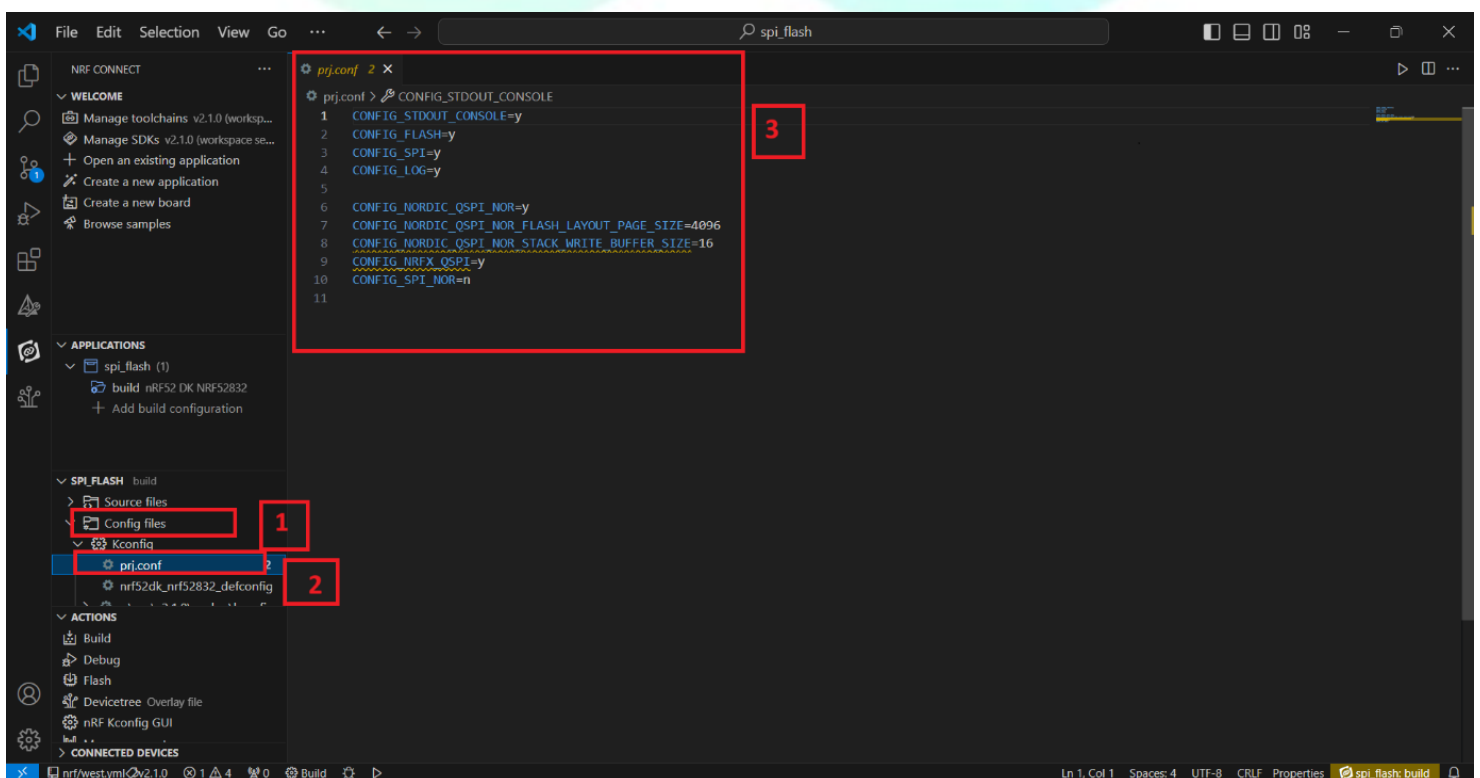
- Click on **Create new build configuration** [1]. Here you can change the board version, if you are using nRF52832, then select **nrf52dk_nrf52832** [2] or you can change from dropdown menu for another version like nRF52833 etc.
- Click on the Configuration and select **prj.conf** [3] from dropdown menu and then click on the **Build Configuration** [4] as shown below in the picture.



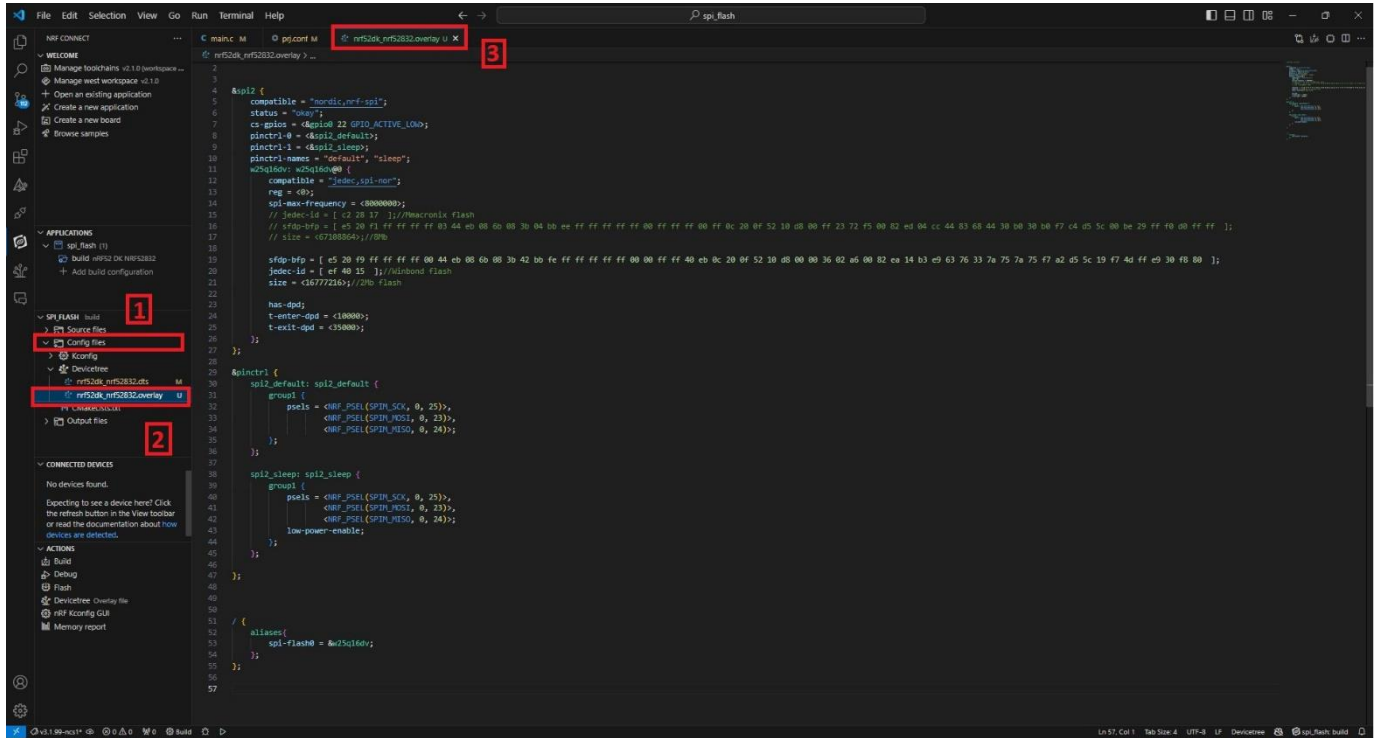
- Go to source file, click **source file [1]** > click on **Application** > click on **src** > click on **main.c [2]**.
- By Clicking on **main.c** file and you will see the code on your screen [3].



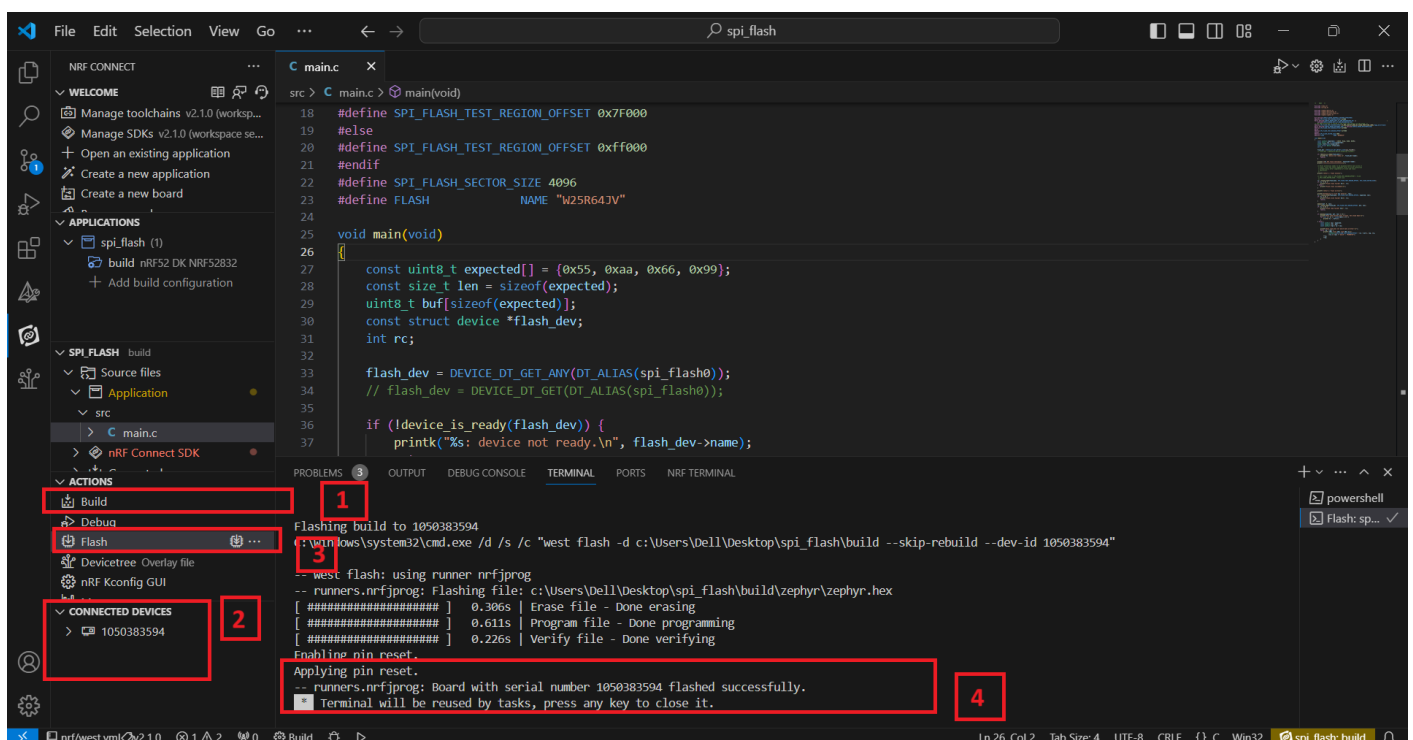
- To configure the prj configuration, click on **Config files [1]** > click on **Kconfig** > click on **prj.conf [2]**.
- The prj configuration will appear on your screen [3] as shown in the picture below.



- To configure the i2c protocol, you have to enable it in the **.overlay** file.
- Click on the **Config files [1]** > click on **Kconfig** > click on **Devicetree** > click on **nrf52dk_nrf52832.overlay [2]**.
- The .overlay file will appear on your screen and add the given code to the .overlay file as shown in the picture given below **[3]**.



- Click on **Build [1]** configuration again and check the **CONNECTED DEVICES [2]**.
- If device id is visible, then **Flash [3]** the code in Dev Kit.
- If **flashed successfully [4]** message is displayed on serial terminal, then flash process is complete.



❖ OUTPUT

- To see the output on the terminal click on the **device ID [1]** -> click on **VCOM0 COM1 [2]** -> click on **115200 8n 1 rtscetsoff [3]** as shown below in the picture.

The screenshot shows the NXP Connect IDE with the 'Serial Port Connection Device' dialog box open. The 'Select a device' dropdown is set to '04985974 w550000000'. The 'Connected devices' list shows '115200 8n 1 rtscetsoff'. The main.c code is visible in the background, showing the SPI flash testing logic. Red boxes are drawn around the device ID '04985974', the COM port 'VCOM0 COM1', and the baud rate '115200 8n 1' in the dialog box.

The screenshot shows the NXP Connect IDE with the 'Terminal' tab selected. The terminal output displays the results of the SPI flash testing process. Red boxes are drawn around the terminal output area and the 'VCOM0 COM1' device in the 'CONNECTED DEVICES' list. The terminal output shows the following steps:

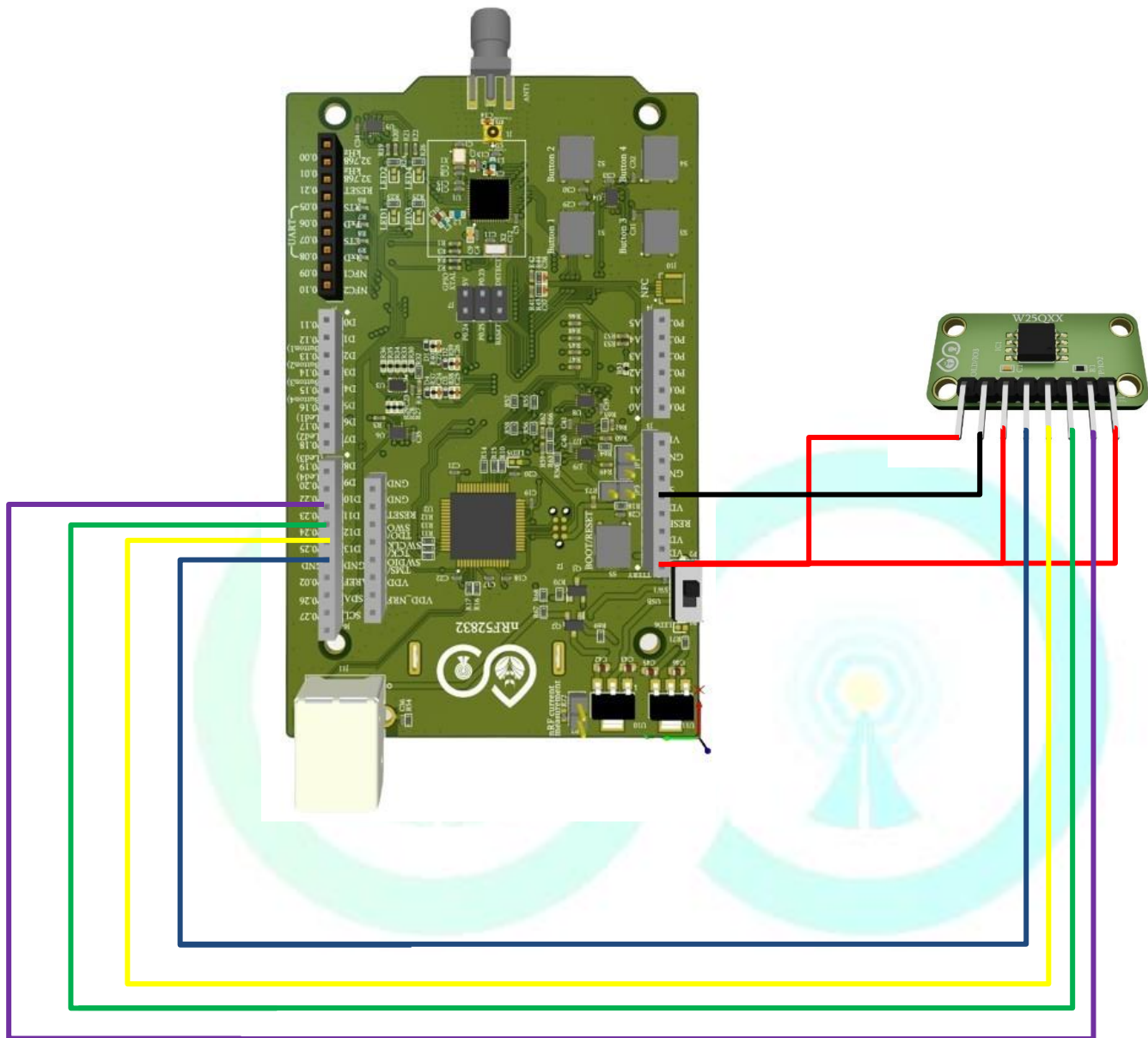
```

w550000000 SPI flash testing
=====

Test 1: Flash erase
Flash erase succeeded!

Test 2: Flash write
Attempting to write 1 bytes
Data read matches data written. Very Good!
  
```


❖ PIN CONFIGURATION



BOARD PIN -> MEMORY PIN

VDD -> VDD, HOLD, WP

GND -> GND

P0.24 -> MISO

P0.23 -> MOSI

P0.25 -> SCK

P0.22 -> CS