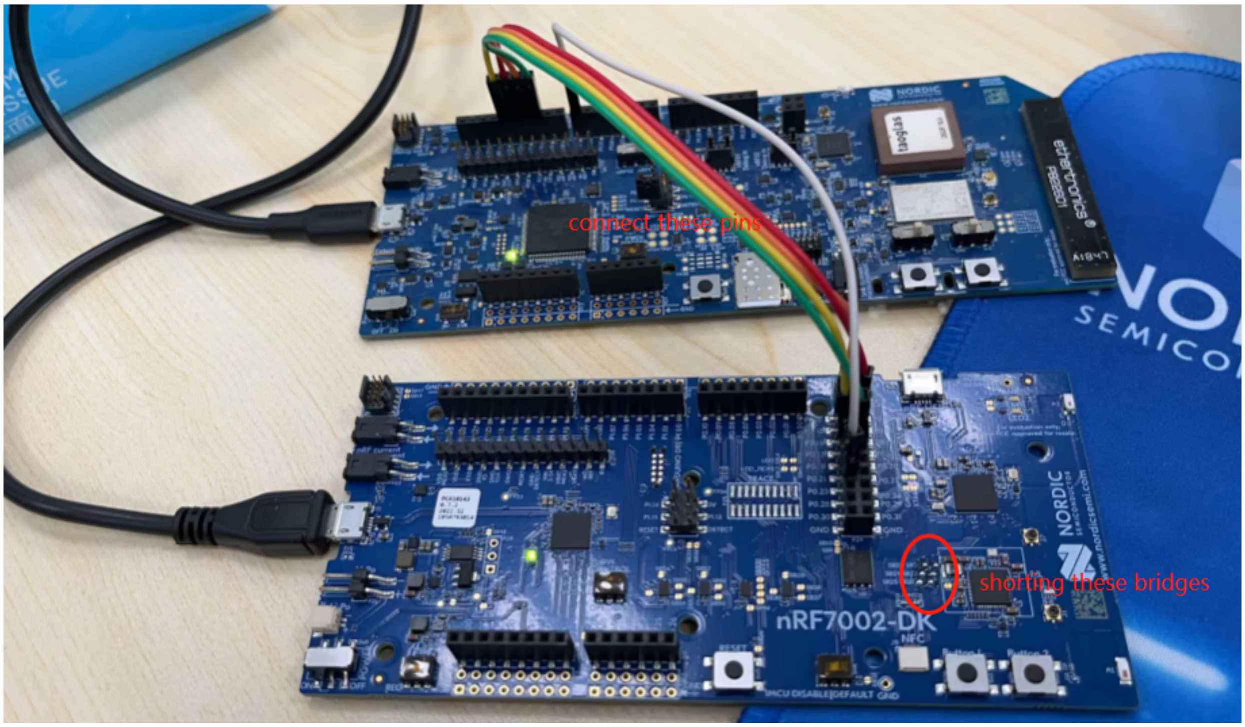
nRF9160dk & nRF7002dk WIFI positioning

How to connect nRF9160dk and nRF7002dk

1. nRF7002dk communicates with nRF340 through QSPI by default, now we want to change it communicate with nRF9160dk via SPI bus, so the first thing is connect two DK boards through jumpers, please connect them according to the following tables.

nRF9160 dk	nRF7002 dk	Pin define
P0.10	P0.18	CS
P0.13	P0.17	SPIM_SCK
P0.12	P0.14	SPIM_MISO
P0.11	P0.13	SPIM_MOSI
P0.07	P0.23	HOST_IRQ

2. shorting the solder bridges **SB20—SB25** on 7002DK



3. flash program for nRF7002dk and nRF9160dk

    - nRF7002 need to power on the wifi, so we must init the power pins and disabled the QSPI bus. It's a simple project only need control two pins. you can get the source code from below urls.

      https://github.com/Noy0908/nRF9160dk_nRF7002dk_WIFI_positioning.git

      branch: wifi_power_on for nRF7002dk

      branch: asset_tracker_wifi_position** for NRF9160dk

- **asset_tracker_v2** is an application which can support wifi positioning on nRF9160DK, but now it only supports nRF7002ek, so if we want to support nRF7002DK, we need to modify the wifi driver.
- first NCS SDK check to the tag **v2.2.99-dev3**.
- Then modify below driver file.

  wifi driver path: **C:\ncs\v2.0.2\nrf\drivers\wifi\nrf700x\zephyr\src\qspi\src\rpu_hw_if.c**

  comments the power pins config code and rpu_pwron() function:

  ```
  int rpu_gpio_config(void)

  {

    int ret;

    // if (!device_is_ready(iovdd_ctrl_spec.port)) {

    //  return -ENODEV;

    // }

    // if (!device_is_ready(bucken_spec.port)) {

    //  return -ENODEV;

    // }

    if (!device_is_ready(host_irq_spec.port)) {

  •    return -ENODEV;

    }

    // ret = gpio_pin_configure_dt(&bucken_spec, (GPIO_OUTPUT |
  NRF_GPIO_DRIVE_H0H1));

    // ret = gpio_pin_configure_dt(&iovdd_ctrl_spec, GPIO_OUTPUT);

    LOG_DBG("GPIO configuration done...\n\n");

    return 0;

  }
  ```

  ```
  comment rpu_pwron() function
  ```

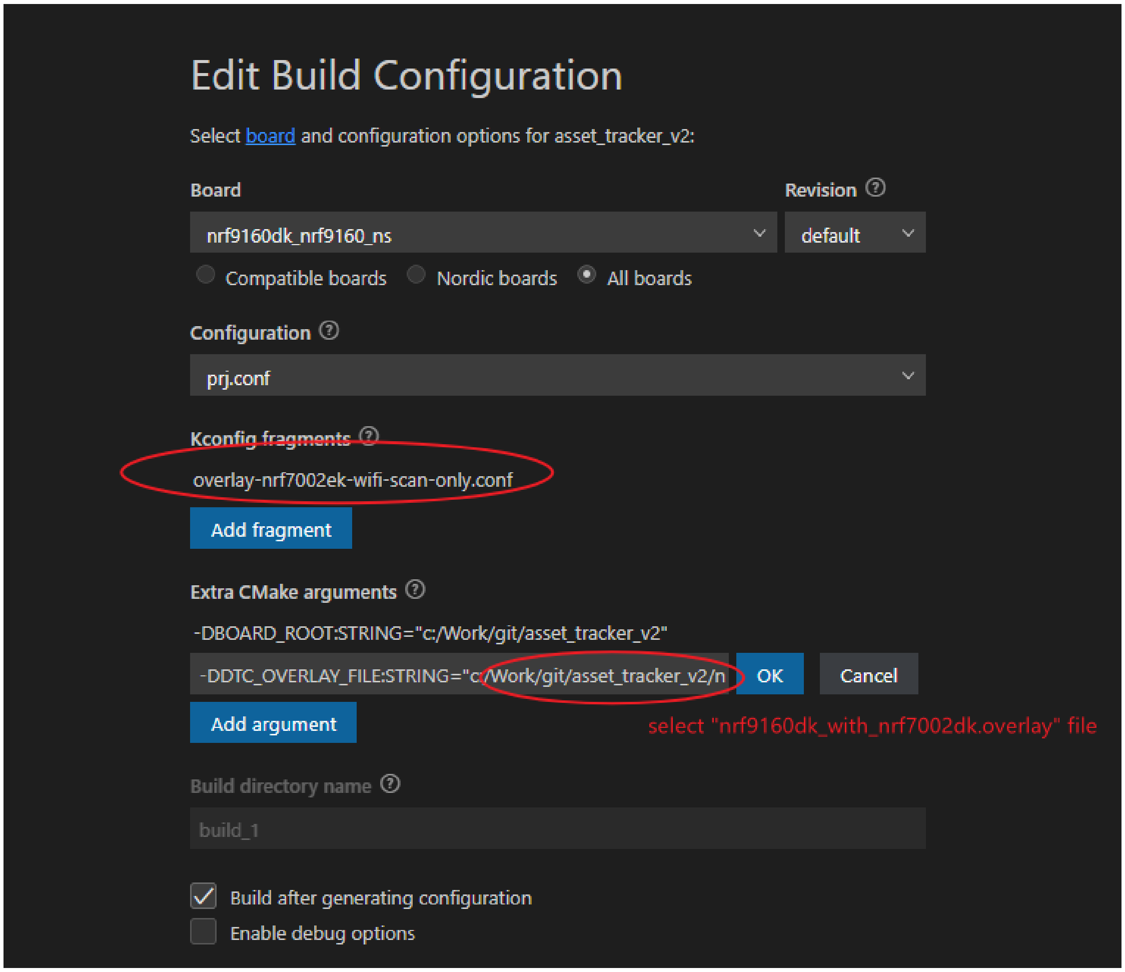
```
int rpu_enable(void)
{
    rpu_gpio_config();
    ble_gpio_config();
    // rpu_pwron();
    rpu_qspi_init();
    rpu_wakeup();
    rpu_clks_on();

    return 0;
}
```

- pull the **asset_tracker_v2** applications from github and flash it to the nRF9160dk, remember to modify below macro if you can't register LTE network.

  CONFIG_LTE_NETWORK_MODE_NBIOT_GPS=y

  Edit build configuration as below figure.



Edit Build Configuration

Select board and configuration options for asset_tracker_v2:

Board: nrf9160dk_nrf9160_ns

Revision: default

○ Compatible boards ○ Nordic boards ● All boards

Configuration: prj.conf

Kconfig fragments: overlay-nrf7002ek-wifi-scan-only.conf

Extra CMake arguments:

```
-DBOARD_ROOT:STRING="c:/Work/git/asset_tracker_v2"
-DDTC_OVERLAY_FILE:STRING="c:/Work/git/asset_tracker_v2/n
```

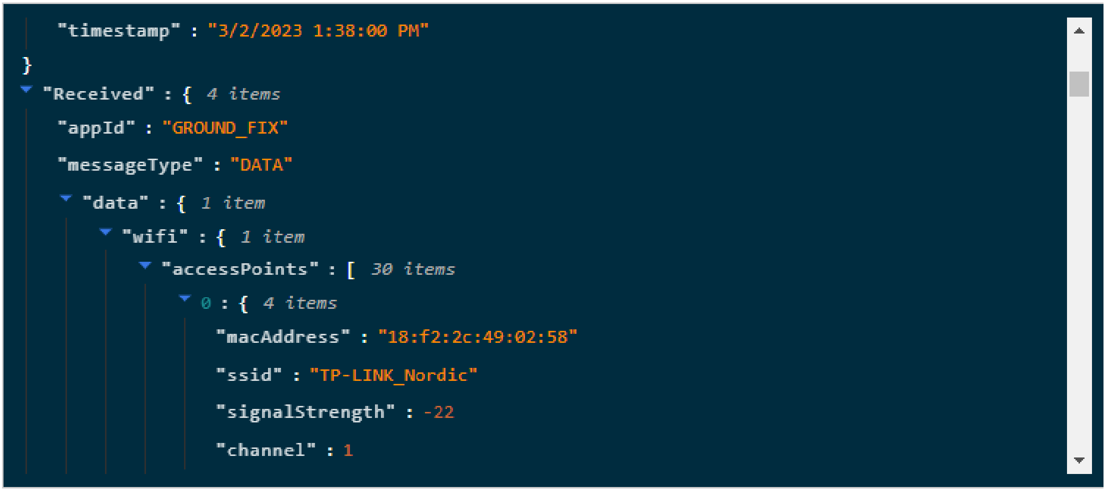
select "nrf9160dk_with_nrf7002dk.overlay" file

Build directory name: build_1

☑ Build after generating configuration

□ Enable debug options

nRF Cloud log

- 

```
   "timestamp" : "3/2/2023 1:38:00 PM"
}
▾ "Received" : { 4 items
   "appId" : "GROUND_FIX"
   "messageType" : "DATA"
   ▾ "data" : { 1 item
      ▾ "wifi" : { 1 item
         ▾ "accessPoints" : [ 30 items
            ▾ 0 : { 4 items
               "macAddress" : "18:f2:2c:49:02:58"
               "ssid" : "TP-LINK_Nordic"
               "signalStrength" : -22
               "channel" : 1
```

Notice :To start the test, you need to reset two dk boards. It is better to reset the 7002dk one second in advance.