

WS2116 MODULE/EVB

FW DEVELOPMENT GUIDE

Version: 2.0.2

CHANGE HISTORY

Version	Date	Description	Author
WS2116_FW_Guide_1.0.4	2019.2.1	Updated descriptions for Jorjin SDK v.1.0.0	Joshua Guo
WS2116_FW_Guide_2.0.0	2021.7.23	Updated descriptions for Jorjin SDK v2.1.0+	Louis Lee
WS2116_FW_Guide_2.0.1	2021.12.17	Added notation for Sigfox transmissions	Jack Tseng
WS2116_FW_Guide_2.0.2	2022.1.19	Updated descriptions for Jorjin SDK v2.1.7	Jack Tseng

CATALOG

1	Introduce.....	5
1.1	Development SDK from STMicro	5
1.2	Debug Tool - STLINK	6
1.3	Sigfox Simulator - SDR Dongle	6
1.4	Jorjin Sigfox and BLE Dual Mode EVB	8
1.5	Jorjin Dual Mode Module	8
1.6	Development Tool.....	9
2	Flash Mapping in BLUENRG-2	10
2.1	Flash Address Mapping	10
2.1.1	Enable OTA service manager	10
2.1.2	Disable OTA service manager.....	10
2.2	Flash Control.....	11
2.2.1	Flash functions	11
3	EEPROM mapping in WS2116 EVB	12
4	Jorjin SDK.....	13
4.1	SDK Information.....	13
4.2	IAR Embedded Workbench IDE	13
4.2.1	Open project file	13
4.2.2	Defined symbols	13
4.2.3	Switch your module/EVB type	24
4.2.4	Sigfox/BLE functions.....	25
4.2.5	Switch SMA connector.....	31
4.2.6	Build code and file location	32
4.2.7	Function Testing	35
5	FW OTA.....	38

- 5.1 ST-Link.....38
- 5.2 OTA.....43
- 6 How to Evaluate with Sigfox funciotn.....52
 - 6.1 Use PUBLIC KEY (Use SDR dongle).....52
 - 6.2 Use PRIVATE KEY53
 - 6.3 Read Sigfox ID AND PAC.....55
 - 6.4 Scan mode.....56

1 INTRODUCE

This document is for Sigfox and BLE dual mode module FW development guide, which is suitable for WS2116 EVB and Module. The WS2116 is designed for Sigfox Monarch which feature makes IoT devices aware of the local radio configuration (RC) to use, WS2116 support RC 1, 2, 3, 4, 5 and 6.

1.1 DEVELOPMENT SDK FROM STMICRO

2 items need to download from STMicro.

1. STSW-BNRG-S2LP evaluation software package based on BlueNRG-2 and S2-LP

http://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/stsw-bnrg-s2lp.html

2. BlueNRG-1 ST-LINK utility for BlueNRG-1, BlueNRG-2 MCU

http://www.st.com/content/st_com/en/products/embedded-software/wireless-connectivity-software/stsw-bnrg1stlink.html

NOTE1: Default BlueNRG-1_ST-LINK_CLI.exe in NRG ST-LINK installation directory has issue which is section erase.

Please unzip **BlueNRG-1_ST-LINK_CLI.7z** and copy it inside the folder C:\Program Files (x86)\STMicroelectronics\BlueNRG-1_2 ST-Link Utility V 2.0.0\ST-LINK_Utility to replace the original file.

NOTE2: DON'T erase Page 124 - 128 of flash by STLINK CLI or GUI

More BlueNRG-2 SW package for from ST:

<http://www.st.com/en/wireless-connectivity/bluenrg-2.html>

More information from Sigfox:

<https://resources.sigfox.com/>

1.2 DEBUG TOOL - STLINK

Buy it from ST or contact ST distributor.

<http://www.st.com/en/development-tools/st-link-v2.html>



1.3 SIGFOX SIMULATOR - SDR DONGLE

You must use antenna with correct frequency domain for your development sigfox RC zone.



Sigfox RC zones have different frequency as below:

Zone 1: Europe, Oman, South Africa

Tx Frequency: 868.13MHz

Rx Frequency: 869.525MHz

Zone 2: USA, Mexico, Brazil

Tx Frequency: 902.2MHz

Rx Frequency: 905.2MHz

Zone 3: Japan

Tx Frequency: 923.2MHz

Rx Frequency: 922.2MHz

Zone 4: Australia, New Zealand, Singapore, Taiwan, Hong Kong, Colombia, Argentina

Tx Frequency: 920.8MHz

Rx Frequency: 922.3MHz

Zone 5: South Korea

Tx Frequency: 923.25MHz

Rx Frequency: 922.25MHz

Zone 6: India

Tx Frequency: 865.2MHz

Rx Frequency: 866.3MHz

More information

<https://resources.sigfox.com/document/sigfox-sdr-dongle>

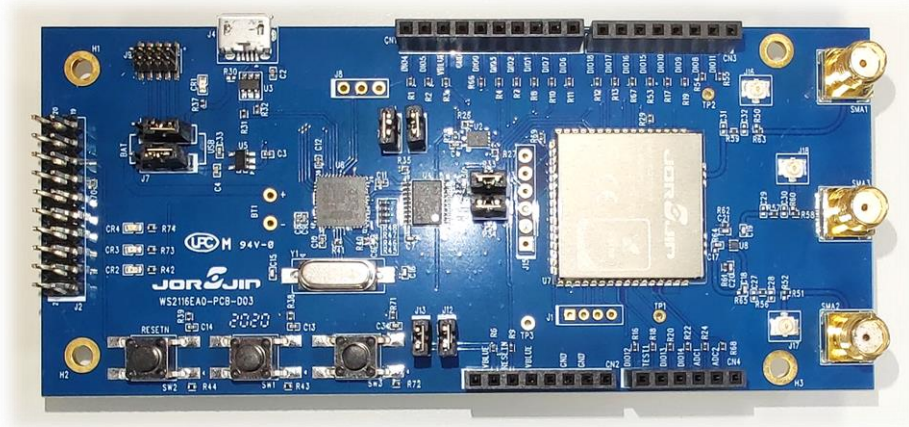
https://storage.sbg1.cloud.ovh.net/v1/AUTH_669d7dfced0b44518cb186841d7cbd75/staging_docs/att19630513-1709-SIGFOX-DATASHEET-SDR_dongle.pdf

Where to buy SDR Dongle:

<https://www.digikey.tw/product-detail/zh/sigfox/SDR-DONGLE/1895-1000-ND/7930762>

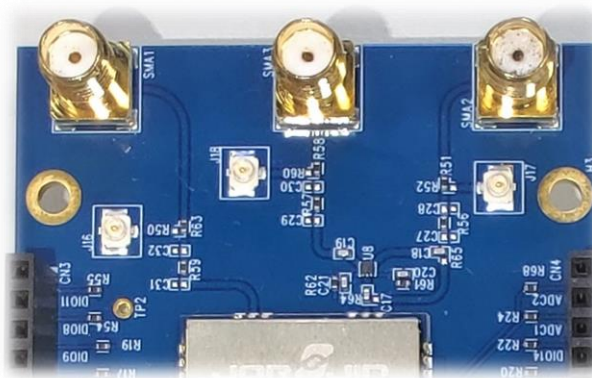
1.4 JORJIN SIGFOX AND BLE DUAL MODE EVB

- WS2116 EVB



The WS2116 EVB user guide you can download from Jorjin website or contact with Jorjin Sales.

WS2116 EVB has three SMA connector, left (SMA1) is for BT and the others (SMA2, SMA3) is for Sigfox.



1.5 JORJIN DUAL MODE MODULE



1.6 DEVELOPMENT TOOL

- IAR embedded workbench 8.32+ (*Note: 8.50.9 is required since SDK v2.1.7)
- Jorjin SDK version 2.1.0+
- Standard AT command FW developed by Jorjin

2 FLASH MAPPING IN BLUENRG-2

2.1 FLASH ADDRESS MAPPING

NOTE: DON'T modify page 124-128, there have module information and calibration parameter and SIGFOX sequence number.

2.1.1 Enable OTA service manager

	0x10080000
Jorjin Reserved (6K)	
	0x1007E800 (Page. 125)
Sigfox Reserved (2K)	
	0x1007E000 (Page. 124)
User app (248K)	
	0x10040000

2.1.2 Disable OTA service manager

	0x10080000
Jorjin Reserved (6K)	
	0x1007E800 (Page. 125)
Sigfox Reserved (2K)	
	0x1007E000 (Page. 124)
User app (178K)	
	0x10051800
OTA Service Manager (70K)	
	0x10040000

2.2 FLASH CONTROL

1. Reading Flash memory

To read one single word of the flash, just read it as if RAM memory: read the desired flash address and get read data on the bus.

2. Erasing Flash

The Flash controller allows erasing **one page** or the **full** main Flash.

3. Write function

The Flash Controller allows writing one word (4 bytes), up to 4 words or the full main Flash memory (with a single fixed word).

4. Basic Flash operations:

Erase a page

Write a page word by word

Verify write operation word by word

You have to erase flash (page) first then program flash. Otherwise you can't program data successfully.

2.2.1 Flash functions

Erase page (BlueNRG1_flash.c)

➤ `Void FLASH_ErasePage(uint16_t PageNumber);`

Erase all flash (BlueNRG1_flash.c)

➤ `void FLASH_EraseAllFlash(void);`

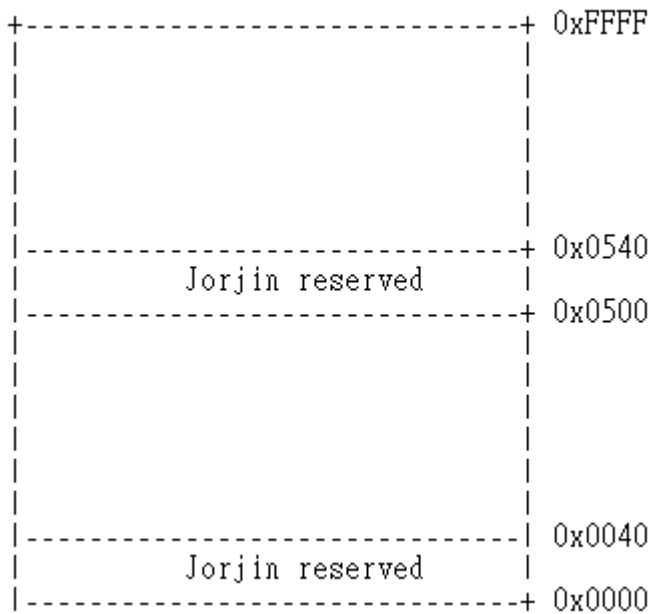
Read flash (uint32) (BlueNRG1_flash.c)

➤ `uint32_t FLASH_ReadWord(uint32_t Address);`

Read flash (uint8) (BlueNRG1_flash.c)

➤ `uint8_t FLASH_ReadByte(uint32_t Address);`

3 EEPROM MAPPING IN WS2116 EVB



DON'T modify 0x0000~ 0x0032 and 0x0500 ~ 0x0540, there have Sigfox information and calibration parameter.

If module flash has been erased, please read EVB information to develop. Reference [4.2.3](#) to switch setting.

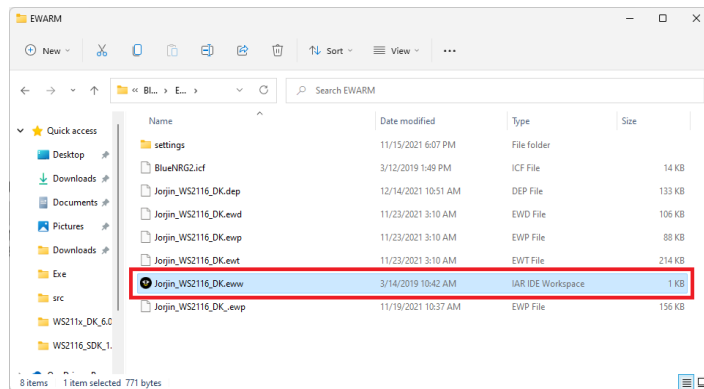
4 JORJIN SDK

4.1 SDK INFORMATION

Jorjin SDK is customized based on ST official released, which include latest Sigfox library and PIN configuration base on WS2116 PIN definition.

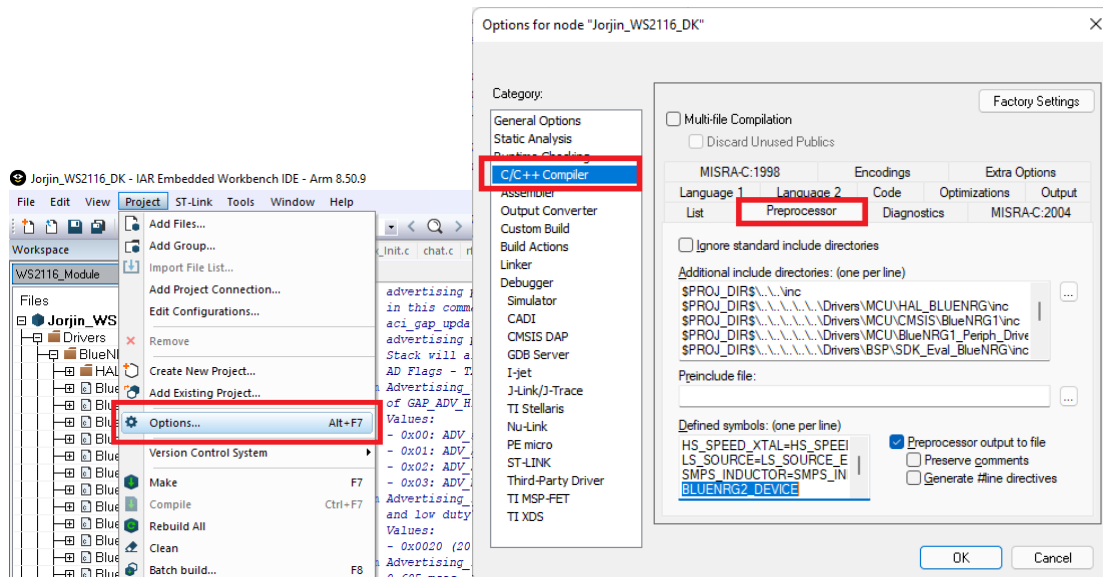
4.2 IAR EMBEDDED WORKBENCH IDE

4.2.1 Open project file



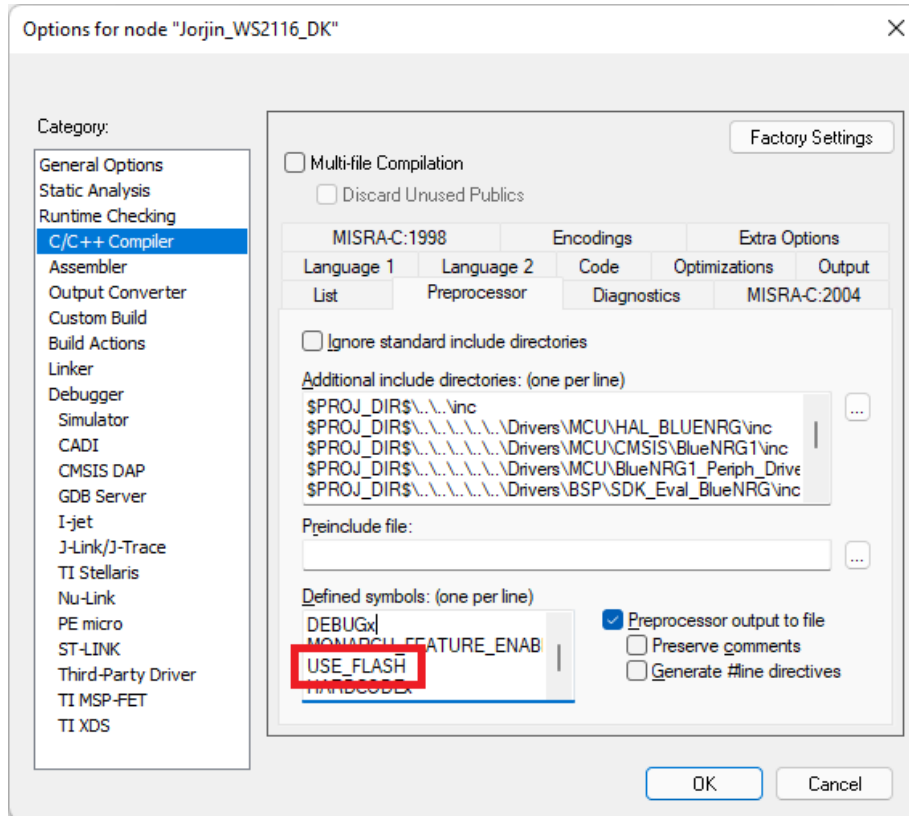
4.2.2 Defined symbols

Project→option→C/C++ compiler→preprocessor



1. Memory type

There have two types of memories, internal FLASH and external EEPROM. To use FLASH for data storing, the “USE_FLASH” symbol must be defined in the “Defined symbols” in IAR’s preprocessor settings (as shown in the following picture). If the “USE_FLASH” symbol is not defined, EEPROM will be used during the initialization.



The “HARDCODE” symbol controls the active ID, PAC, key and rcz at “ST_Sigfox_Init.c”. If it’s not defined, the Sigfox information stored in the memory will be used for the Sigfox operations.

If “HARDCODE” is defined, the hardcoded Sigfox information in “ST_Sigfox_Init.c” will be used, and the original information in the memory will also be overwritten by the hardcoded ones.

```

main.c ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h
ST_Sigfox_Init(NVM_BoardDataType *, uint8_t)
{
    ST_SFX_ERR ST_Sigfox_Init(NVM_BoardDataType *sfxConfig, uint8_t openAfterInit)
    {
        ST_SFX_ERR ret_err = ST_SFX_ERR_NONE;
        #ifdef HARDCODE
            uint32_t id = 0xFEDCBA98;
            uint8_t pac[8] = {0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88};
            uint8_t key[16] = {0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xAA,0xBB,0xCC,0xDD,0xEE,0xFF};
            uint8_t rcz = 1;
        #endif

        /* Configure XTAL frequency and offset for the RF Library */
        ST_RF_API_set_xtal_freq(S2LPManagementGetXtalFrequency());

        /* Macro that defines and initializes the nvmmconfig structure */
        INIT_NVM_CONFIG(nvmConfig);

        /* Sigfox Credentials Management */
        #ifndef USE_FLASH
            nvmConfig.nvmType = NVM_TYPE_FLASH;
            nvmConfig.sfxDataAddress = (uint32_t)FLASH_USER_START_ADDR; /* Set here the address for 'NVM sigfox data' management */
        #endif

        /* Configure the NVM_API */
        SetNVMInitial(&nvmConfig);
        ST_MCU_API_Shutdown(1);

        #ifdef HARDCODE
            sfxConfig->id = id;
            sfxConfig->rcz = rcz;
            memcpy(sfxConfig->pac, pac, 8);

            enc_utils_set_rcz(sfxConfig->rcz);
            enc_utils_set_id(sfxConfig->id);
            enc_utils_set_key(key);
        #endif
        #ifndef USE_FLASH
            ret_err = (ST_SFX_ERR)ST_RF_API_set_rssi_offset(sfxConfig->rssiOffset); /* Override RSSI offset */
        #endif //USE_FLASH
    }
}

```

Sigfox sequence number store located will change depend on “USE_FLASH” define.

```

main.c ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h nvm_api.c x
_setBlockState(uint32_t, uint64_t)
{
    #ifndef USE_FLASH
        #define _nvmReadOperation(addr, nbytes, buff) FlashRead(addr, nbytes, buff)
        #define _nvmWriteOperation(addr, nbytes, buff, mode) FlashWrite(addr, nbytes, buff, mode)
    #else
        #include "S2LP_AUX_EEPROM.h"

        #define _nvmReadOperation(addr, nbytes, buff) EepromRead(addr, nbytes, buff)
        #define _nvmWriteOperation(addr, nbytes, buff, mode) EepromWrite(addr, nbytes, buff)
    #endif
}

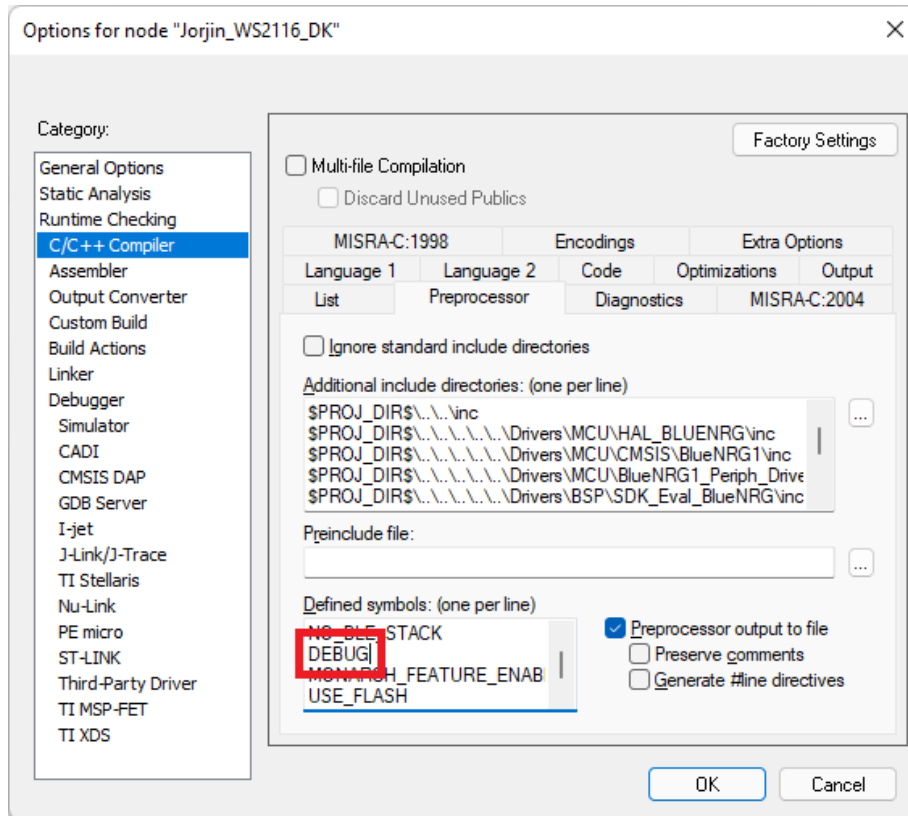
```

Use in main.c, ST_Sigfox_Init.c, nvm_api.c and S2LP_MON_REF_DES.h

```
main.c ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h nvm_api.c S2LP_MON_REF_DES.h >
/**
 * @brief Definitions for EEPROM
 */
#ifdef USE_FLASH
#define EEPROM_PRESENT          EEPROM_NO
#else
#define EEPROM_PRESENT          EEPROM_YES
#endif
```


2. Debug mode

Set “DEBUG=1” to open debug mode, will show PRINTF(“”). Defined symbols “DEBUG=0” to disable debug mode. If use printf(“”) debug message always show context(PRINTF ≠ printf).



It is used in chat.h, gatt.c, rf_api.c and mcu_api_bluewrg1.c.

```
main.c ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h nvm_api.c gatt_db.c x S2LP_MO
[ ] #if DEBUG
    #define PRINTF(...) printf(__VA_ARGS__)
    #else
    #define PRINTF(...)
    #endif
```

```
main.c ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h nvm_api.c chat.h x
```

```
#ifndef _CHAT_H_
#define _CHAT_H_

#ifdef DEBUG
#include <stdio.h>
#define PRINTF(...) printf(__VA_ARGS__)
#define BLE_CHAT_VERSION_STRING "1.0.0"
#else
#define PRINTF(...)
#endif
```

```
main.c ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h nvm_api.c mcu_api_bluenrg1.c x
```

```
#ifdef DEBUG
// #include <stdio.h>
// void ST_dbg_CB(const char *vectcStr,...);
// #define PRINTF(...) { ST_dbg_CB(__VA_ARGS__);}
#elif DEBUG_BLE
#include <stdio.h>
#include "SDK_EVAL_Com.h"
#define PRINTF(...) printf(__VA_ARGS__)
#else
#define PRINTF(...)
#endif
```

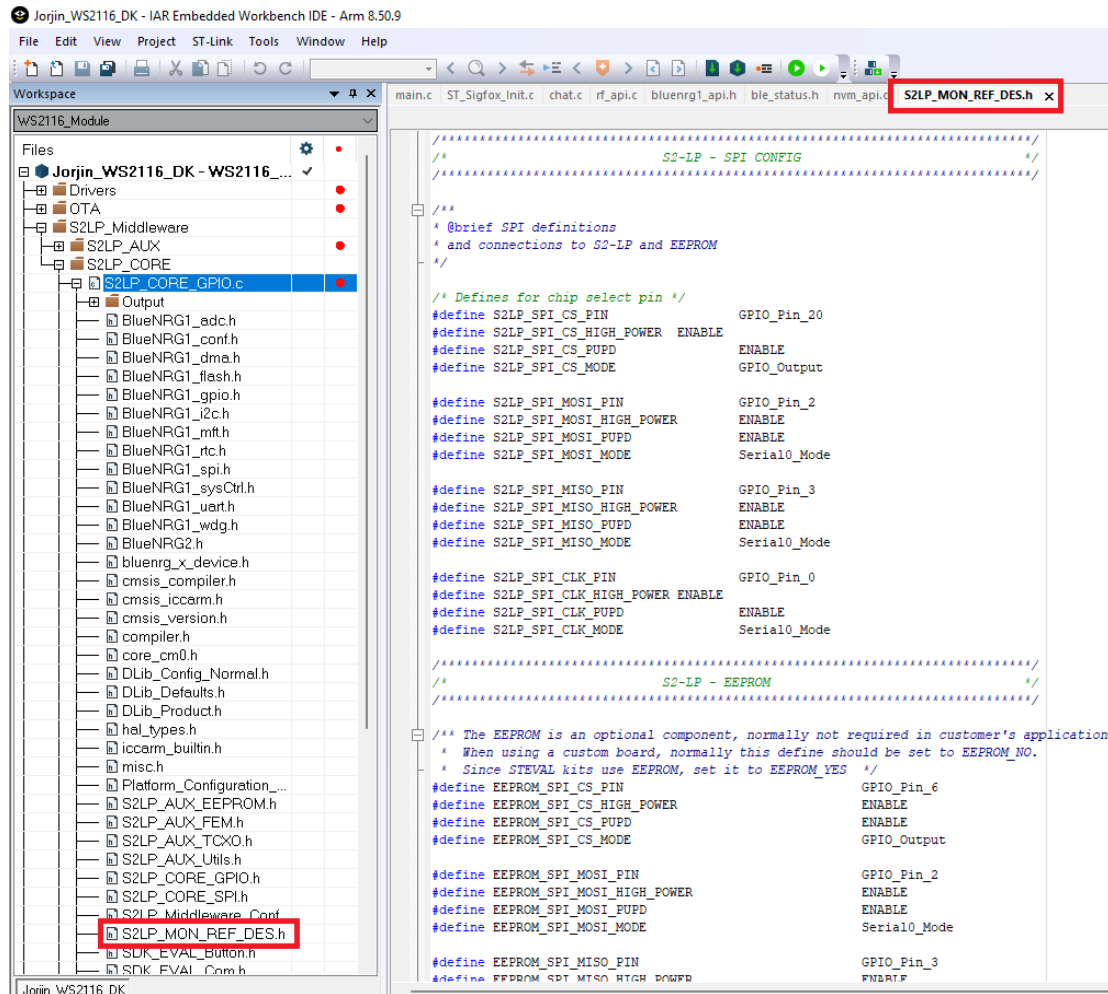
```
main.c ST_Sigfox_Init.c chat.c rf_api.c x bluenrg1_api.h ble_status.h nvm_api.c mcu_api_bluenrg1.c
```

```
/* The DEBUG symbol is used to print the names of the invoked functions */
#define DEBUG

#ifdef DEBUG
//void ST_dbg_CB(const char *vectcStr,...);
// #define PRINTF(...) { ST_dbg_CB(__VA_ARGS__);}
#elif DEBUG_BLE
#include "SDK_EVAL_Com.h"
#define PRINTF(...) printf(__VA_ARGS__)
#else
#define PRINTF(...)
#endif
```

3. GPIO mapping

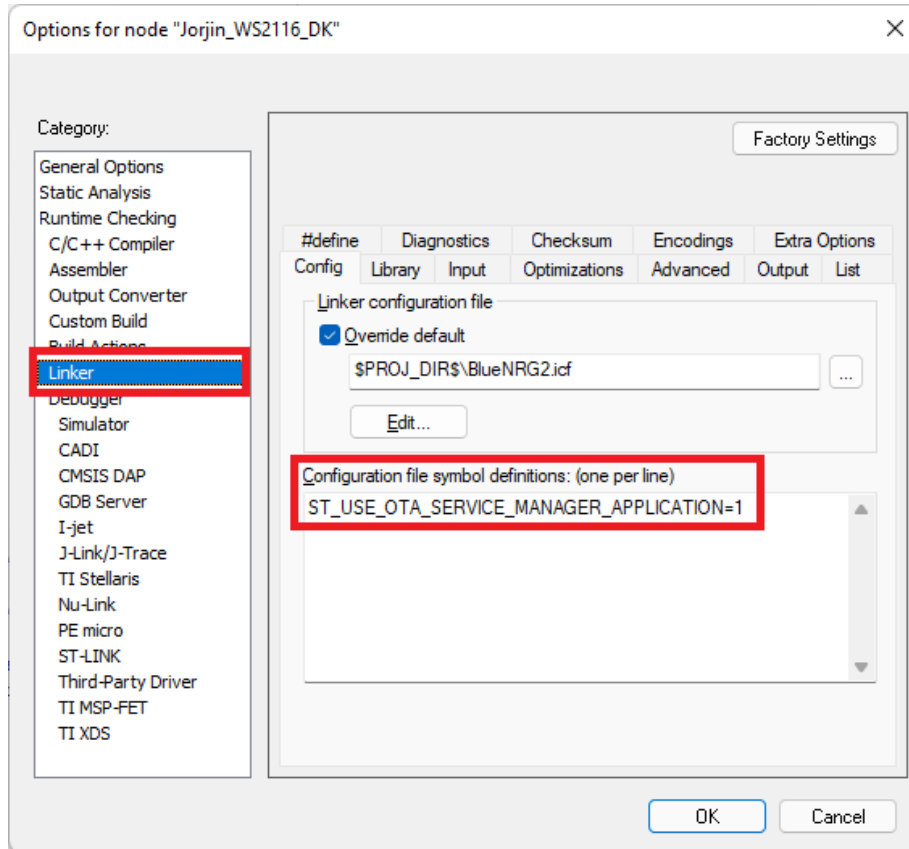
It is used in S2LP_MON_REF_DES.h.

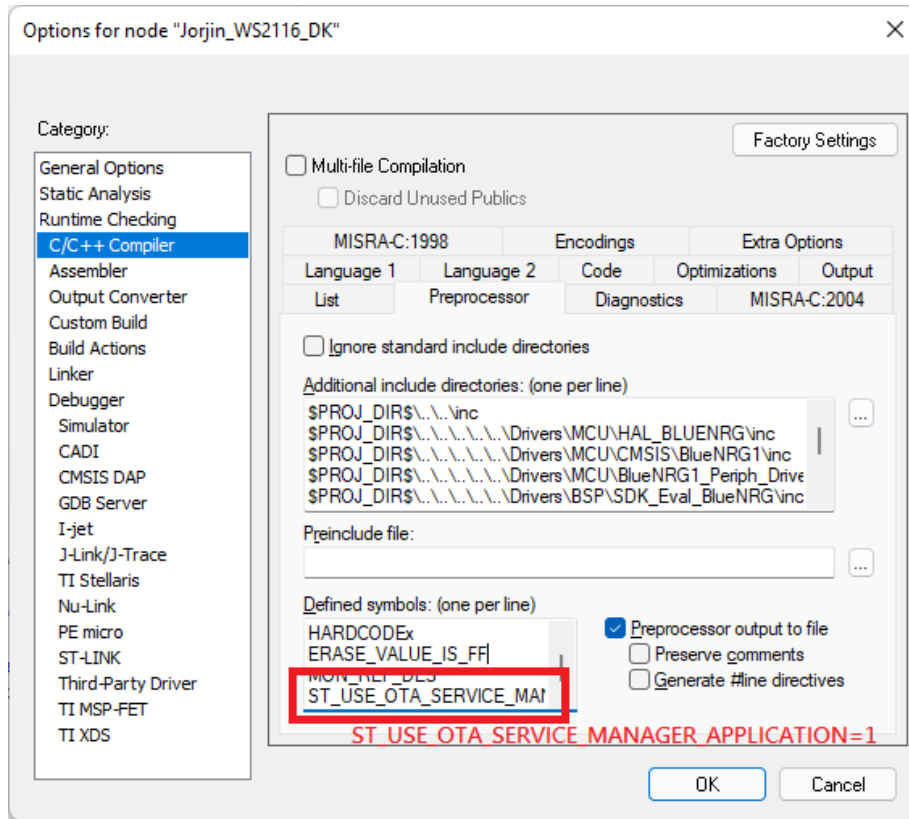


4. OTA mode

- i. Enable OTA service manager

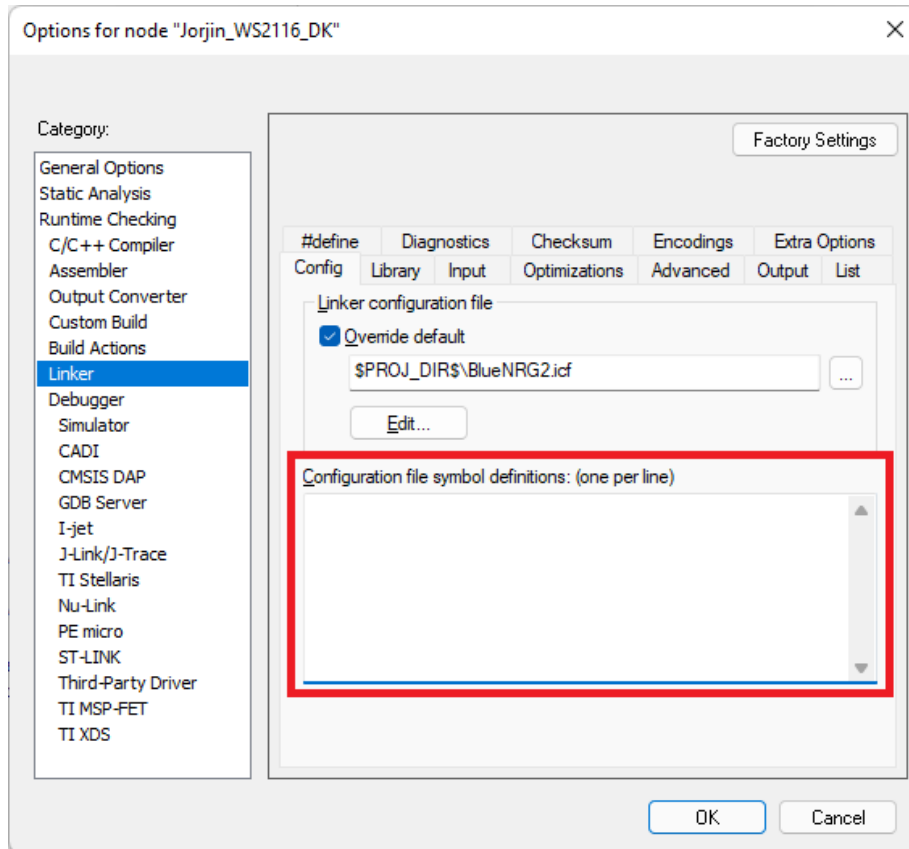
Define “ST_USE_OTA_SERVICE_MANAGER_APPLICATION=1” at **Linker** → **config** → **configuration file symbol definitions** and **C/C++ compiler** → **preprocessor** → **defined symbol** to enable OTA mode will change application FW start address to 0x10051800.

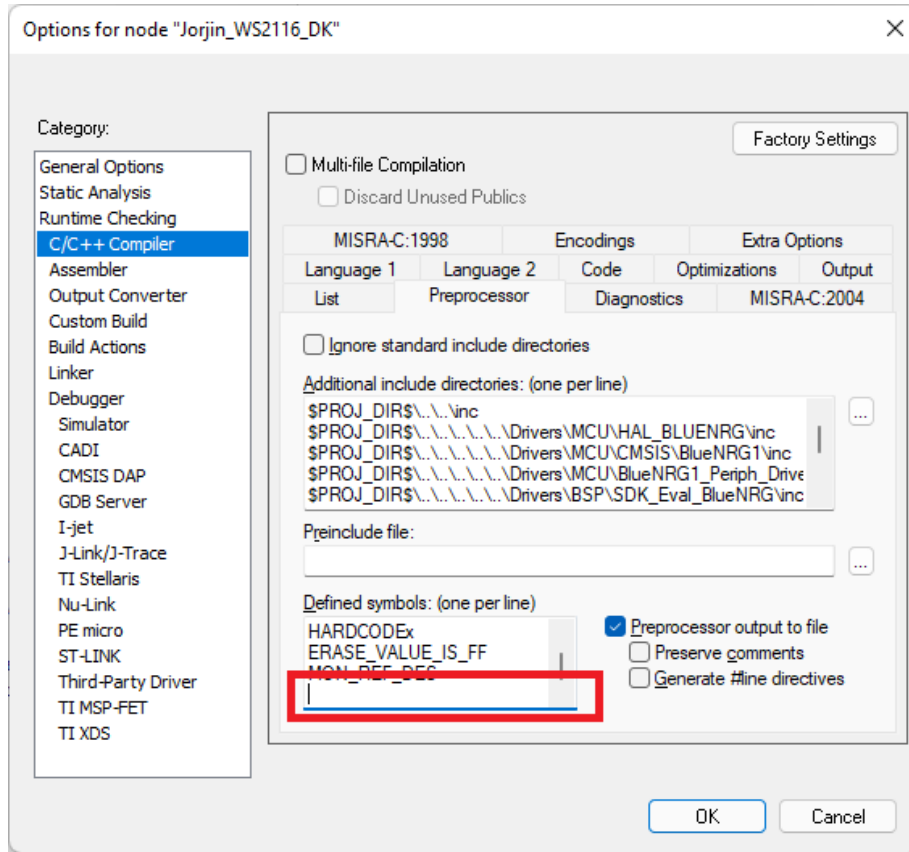




ii. Disable OTA service manager

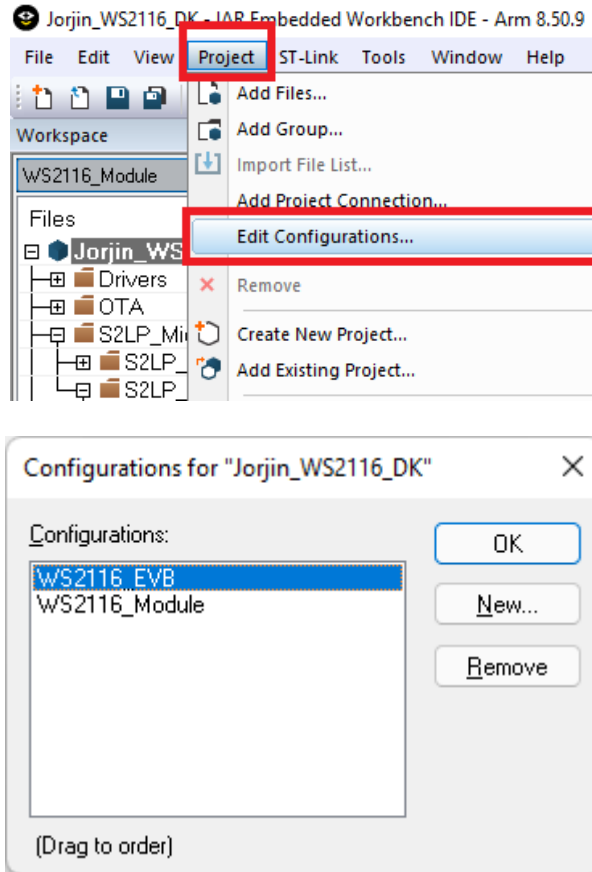
Remove define “ST_USE_OTA_SERVICE_MANAGER_APPLICATION=1” at **Linker** → **config** → **configuration file symbol definitions** and **C/C++ compiler** → **preprocessor** → **defined symbol** to disable OTA mode will change application FW start address to 0x10040000.



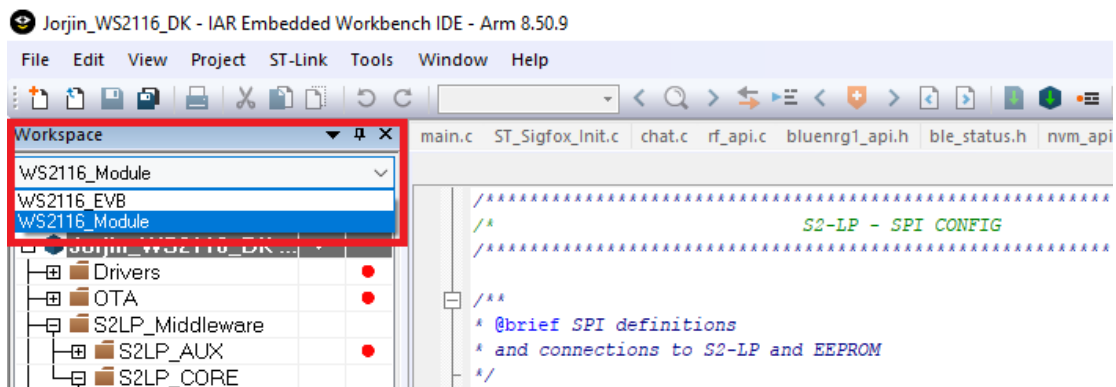


4.2.3 Switch your module/EVB type

Select edit configurations to switch type. Each configuration has different defined symbol.



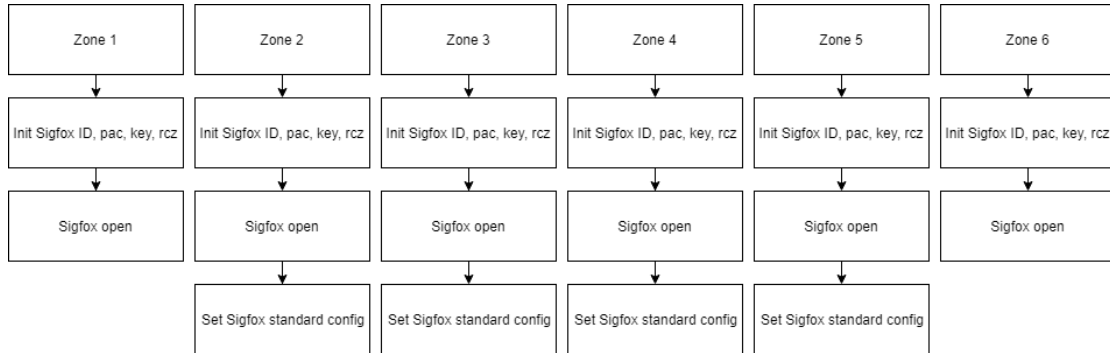
Quick switch configuration



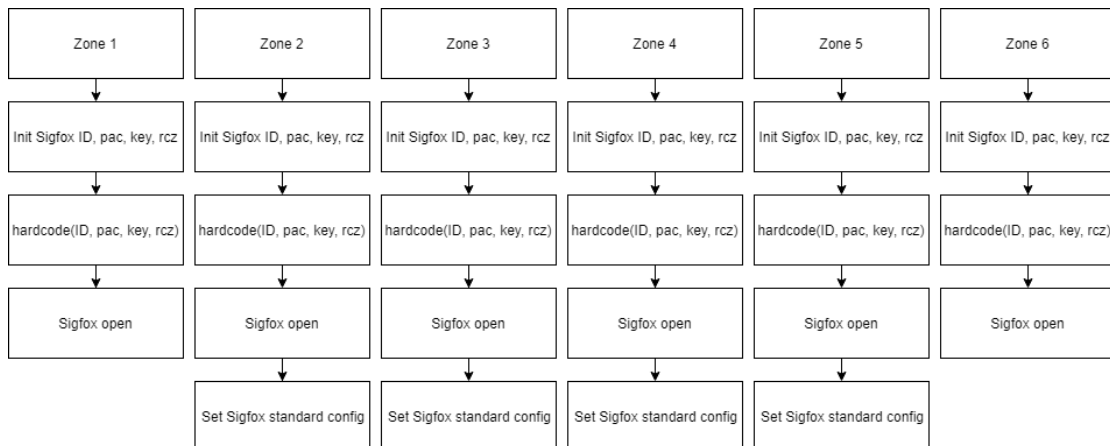
4.2.4 Sigfox/BLE functions

- Sigfox functions

Initialize flow (Read EEPROM/flash data)



Initialize flow (Hard code)



Sigfox Tx. (12 bytes limit) (main.c) Must initialize Sigfox

```
➤ SIGFOX_API_send_frame(sfx_u8 *customer_data,
                        sfx_u8 customer_data_length,
                        sfx_u8 *customer_response,
                        sfx_u8 tx_repeat,
                        sfx_bool initiate_downlink_flag);
```

Set Sigfox private key (main.c) (Must register ID, key from Sigfox)

```
➤ static uint8_t key[16]=\
  { 0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xAA,0xBB,0xCC,0xDD,0xEE,0xFF};

➤ enc_utils_set_key(key);
```

Set Sigfox reduction power (main.c)

```
➤ ST_RF_API_reduce_output_power(sfx_s16 reduction);
```

Set Sigfox private id (main.c)

```
➤ static uint32_t id=0xfedcba98;

➤ enc_utils_set_id(id);
```

Switch the private (0)/public (1) key.

```
➤ enc_utils_set_public_key(1);
```

Set PA. Set 1 if a PA, 0 if not. (st_rf_api.h)

```
➤ ST_RF_API_set_pa (1);
```

Set RSSI offset. (st_rf_api.h)

```
ST_RF_API_set_rssi_offset(sfx_s8 rssi_off);
```

Sigfox monarch scan mode. (sigfox_monarch_api.h)

```
➤ SIGFOX_MONARCH_API_execute_rc_scan (sfx_u8
rc_capabilities_bit_mask, sfx_u16 timer, sfx_timer_unit_enum_t unit,
sfx_u8 (* app_callback_handler ) ( sfx_u8 rc_bit_mask, sfx_s16 rssi ) );
```

NOTE: In WS2116 SDK 2.1.6 or earlier, the following operations are needed for Sigfox compliant transmissions.

1. In `main.c` & `ST_Sigfox_Init.c`, adjust the reduction parameters passed to the `ST_RF_API_reduce_output_power()` function to limit the Sigfox Tx power within the spec at the selected RC.

```
main.c ST_Sigfox_Init.c x chat.c rf_api.c bluenrg1_api.h
ST_Sigfox_Init(NVM_BoardDataType *, uint8_t)
ST_SFX_ERR St_Sigfox_Open_RCZ(uint8_t rcz)
{
    ST_SFX_ERR open_err = ST_SFX_ERR_NONE;

    switch(rcz)
    {
        case 1:
            ST_RF_API_reduce_output_power(2);
    }
}
In St_Sigfox_Open_RCZ() at ST_Sigfox_Init.c
```

```
main.c x ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h nvm_ap
sfx_u8 callback_for_found(sfx_u8 rc_bit_mask, sfx_s16 rssi)
{
    printf("return rc_bit_mask %d\r\n", rc_bit_mask);
    printf("return rssi %d\r\n", rssi);

    switch (rc_bit_mask)
    {
        case 0x01: //RC1
        {
            printf("Detected RC1!!!:\r\n");
            ST_RF_API_reduce_output_power(2);
        }
    }
}
In callback_for_found() at main.c
```

```
main.c x ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.h
main()
{
    sfx_u32 config_words[3]=RC5_CONFIG;
    SIGFOX_API_set_std_config(config_words,0);
    //For SPIE maximum power is 12dBm
    ST_RF_API_reduce_output_power(2);
}
In main() at main.c (for RC5 only)
```

Suggested reduction values for RC1-6 are listed as follows:

TX	RC1	RC2	RC3	RC4	RC5	RC6
(Uplink)	868.13MHz	902.2MHz	923.2MHz	902.8MHz	923.3MHz	865.2MHz
Reduction	2	6	2	6	2	2

- In `main.c` & `ST_Sigfox_Init.c`, set the offset parameters passed to the `ST_RF_API_set_lbt_thr_offset()` function to 0.

```

main.c ST_Sigfox_Init.c x chat.c rf_api.c bluenrg1_api.h bl
St_Sigfox_Open_RCZ(uint8_t)
    case 3:
    {
        volatile uint8_t ret;
        ST_RF_API_set_lbt_thr_offset(0);
        ST_RF_API_reduce_output_power(2);
        /* Turn PA off in RC1/3/5/6/7 */
        ST_RF_API_set_pa(0);
        ret=SIGFOX_API_open(&(sfx_rc_t)RC3C);
    }
    In St_Sigfox_Open_RCZ() at ST_Sigfox_Init.c

```

```

main.c x ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_sta
    case 0x04: //RC3c
    {
        printf("Detected RC3!!!:\r\n");
        /* Turn Power Amplifier Off - 14 dBm out */
        ST_RF_API_set_lbt_thr_offset(0);
        ST_RF_API_reduce_output_power(2);
    }
    In callback_for_found() at main.c

```

- In `rf_api.c`, change the local variable `fifo_buff []` in the `priv_ST_MANUF_SpiRaw_Ramp()` function to a static one to avoid system RAM glitch.

```

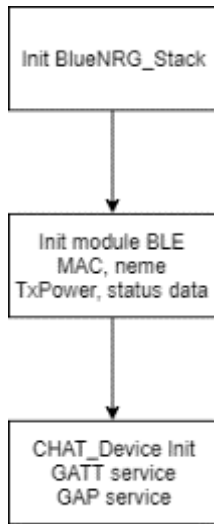
main.c ST_Sigfox_Init.c chat.c rf_api.c x bluenrg1_api.h ble_status.h nvm_api.c S2LP_MON_REF_DES.h
priv_ST_MANUF_SpiRaw_Ramp(uint8_t, uint8_t*, uint8_t*, uint8_t)
/* SPI functions - these functions are implemented using the priv_ST_MANUF_SpiRa
static void priv_ST_MANUF_SpiRaw_Ramp(uint8_t n_bytes, uint8_t* buff_in, uint8_t*
{
    if(st_manuf_context->power_reduction!=0 && buff_in!=zeroes)
    {
        uint32_t i;
        static uint8_t fifo_buff[82];
        fifo_buff[0]=buff_in[0];
        fifo_buff[1]=buff_in[1];
    }
}

```

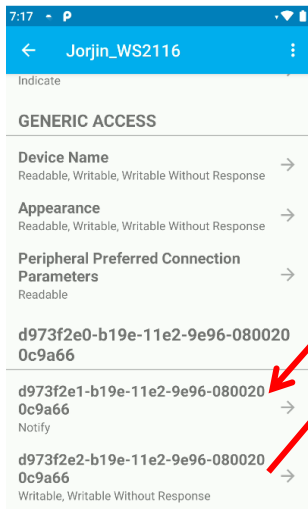
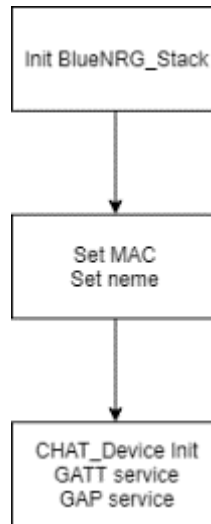
● BLE functions

Initialize flow

(Read EEPROM/flash data)

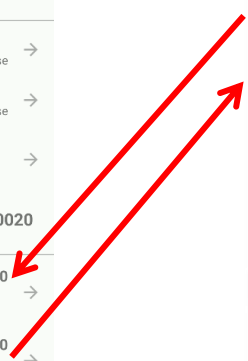


(Hard code)



Transmit from JORJIN module to device.

Receive form device to JORJIN module.



BT MAC (chat.c)

➤ `static uint8_t bdaddr[6]={0x00, 0x19, 0x94, 0xFF, 0xFF, 0xFF};`

BLE name (8 bytes limit) (chat.c)

➤ `static uint8_t name[8]={'W', 'S', '2', '1', '1', '6', 0x00, 0x00};`

BLE local_name (chat.c)

➤ `uint8_t local_name[] =
{AD_TYPE_COMPLETE_LOCAL_NAME, 'J', 'o', 'r', 'j', 'i', 'n', '_', 'W', 'S', '2', '1', '1', '6'};`

(Example in 4.2.7.2)

BLE Tx power setting(chat.c)

➤ `aci_hal_set_tx_power_level(1, 4);`

(Example in 4.2.7.3)

BLE Tx. (chat.c)

➤ `SendRczData(uint8_t rczValue, uint8_t keyValue);`

UART Tx via BT. (chat.c)

➤ `SdkEvalComIOConfig(Process_InputData);`

UART Rx via BT. (gatt_db.c)

➤ `Attribute_Modified_CB(uint16_t handle, uint16_t data_length, uint8_t
*att_data)`

(Example in 4.2.7.1)

4.2.5 Switch SMA connector

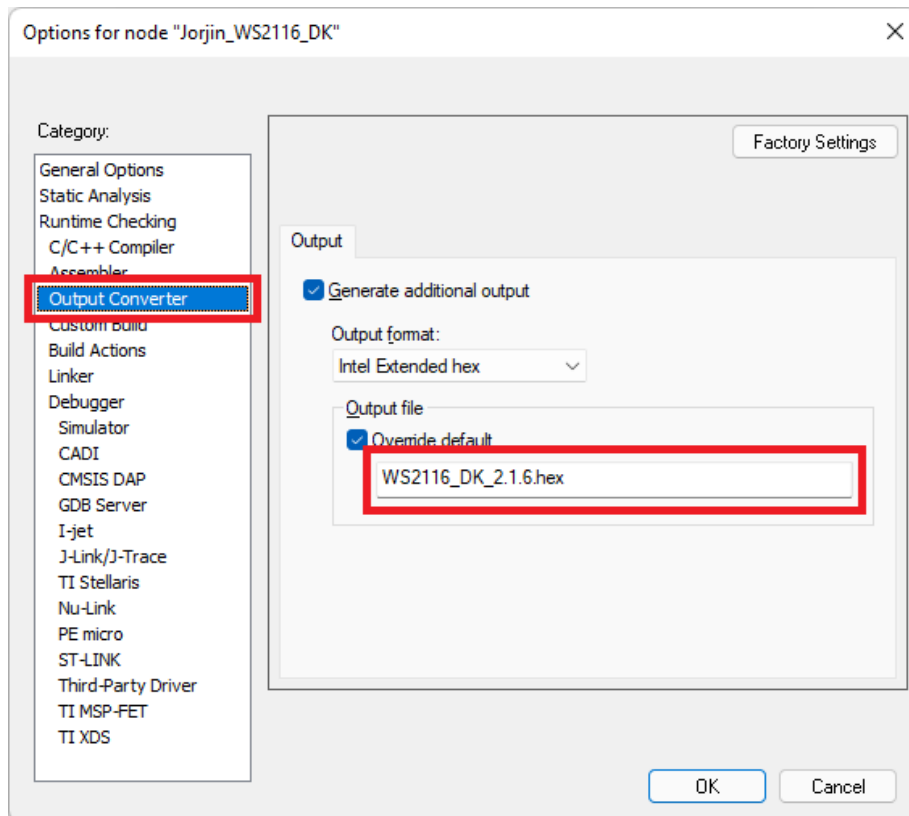
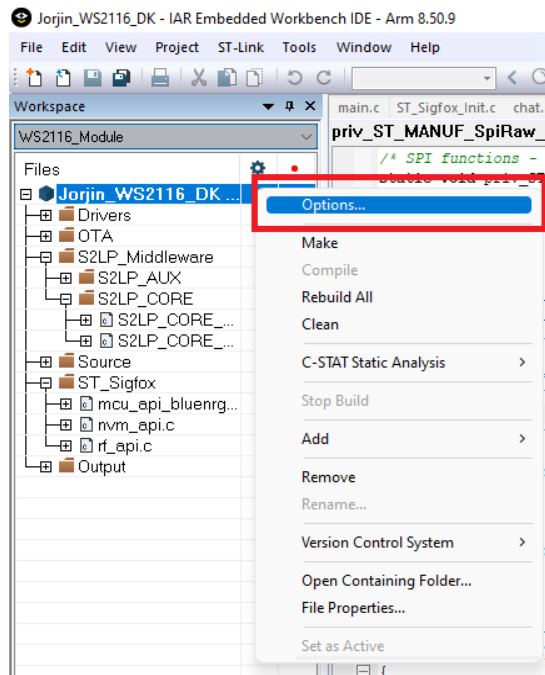
WS2116 has two Sigfox antenna connector, default antenna is SMA2.

If want to use SMA3, please use code below.

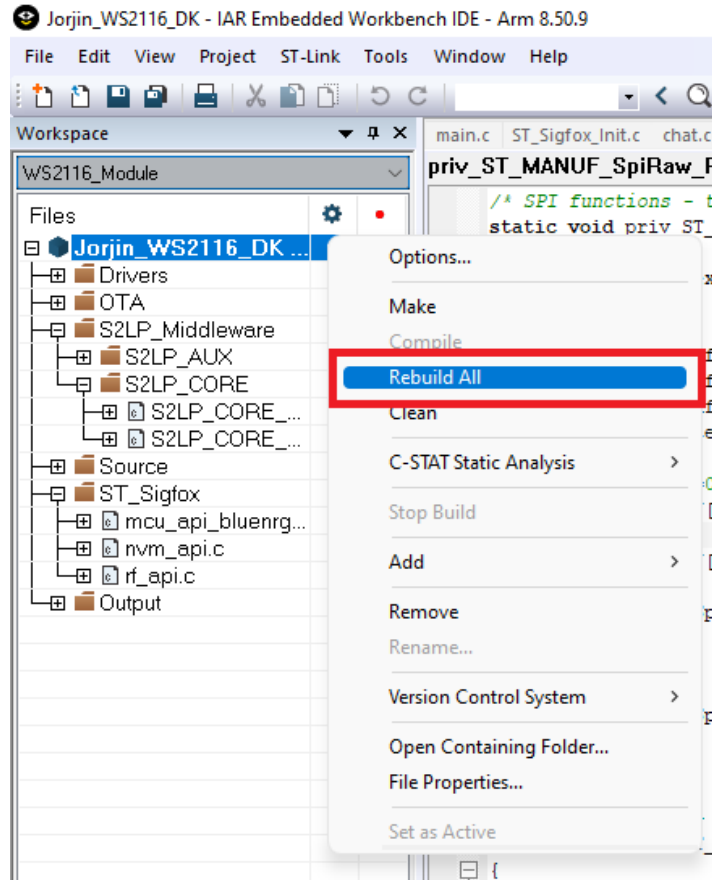
```
GPIO_WriteBit(SMA2_GPIO_PIN, Bit_RESET);  
  
GPIO_WriteBit(SMA3_GPIO_PIN, Bit_SET);;
```

4.2.6 Build code and file location

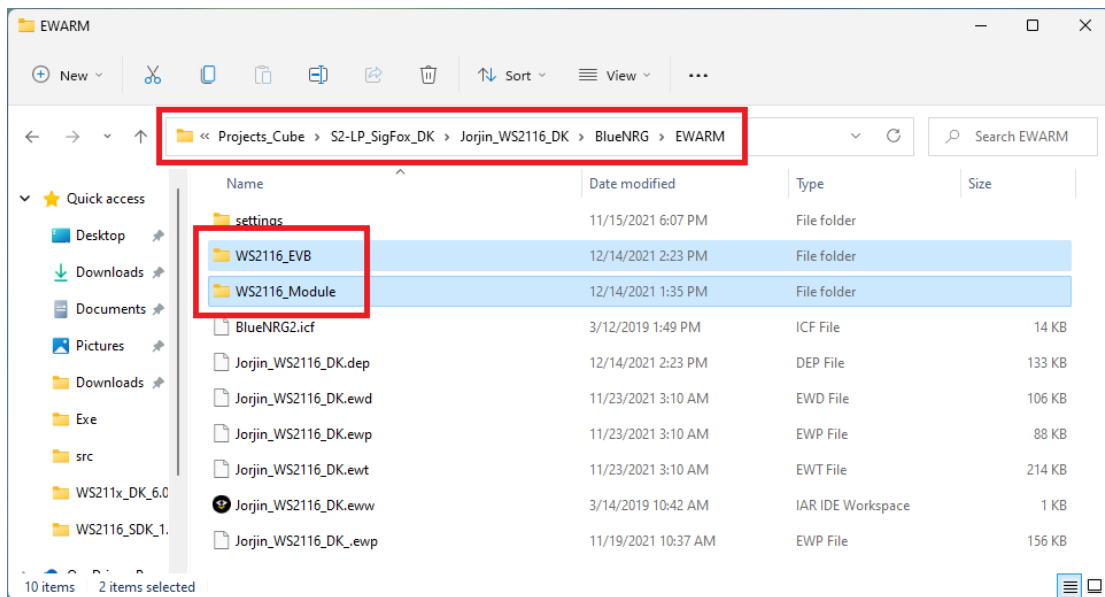
1. Change output file name. “Option→ Output Converter→Output file”

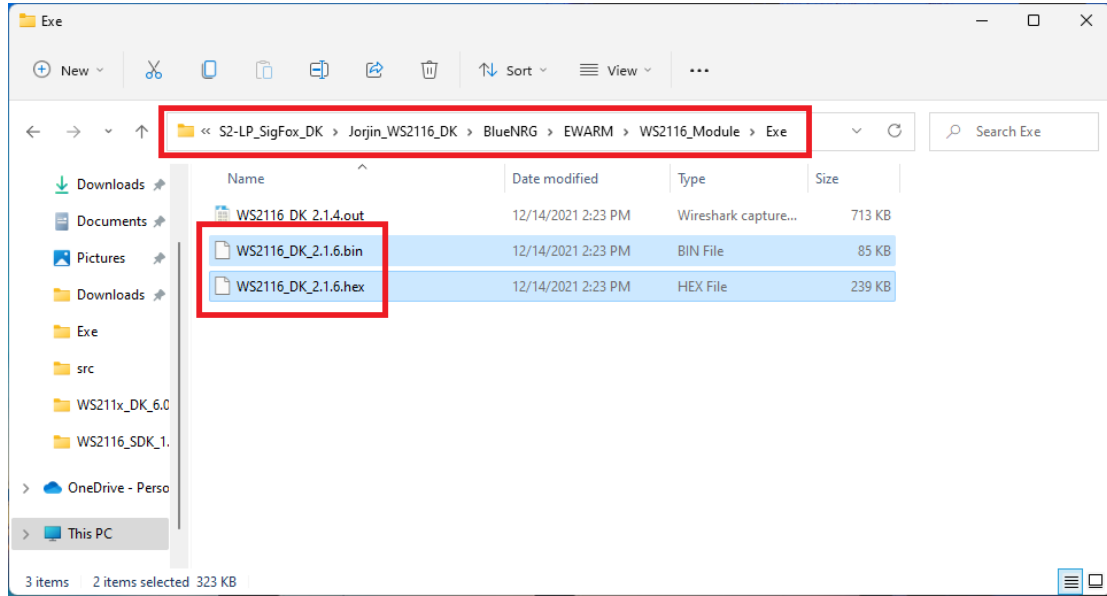


2. Build code



3. File location

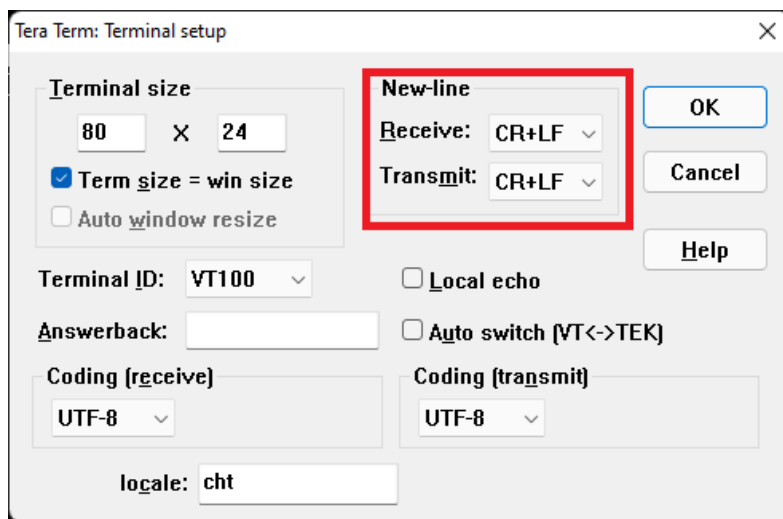
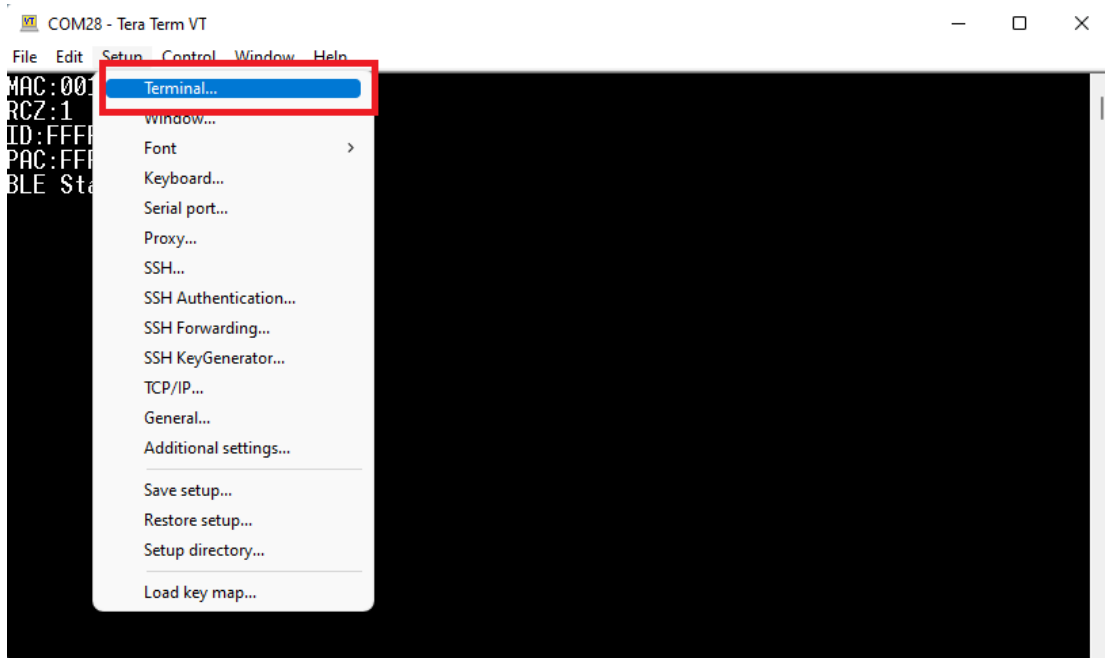




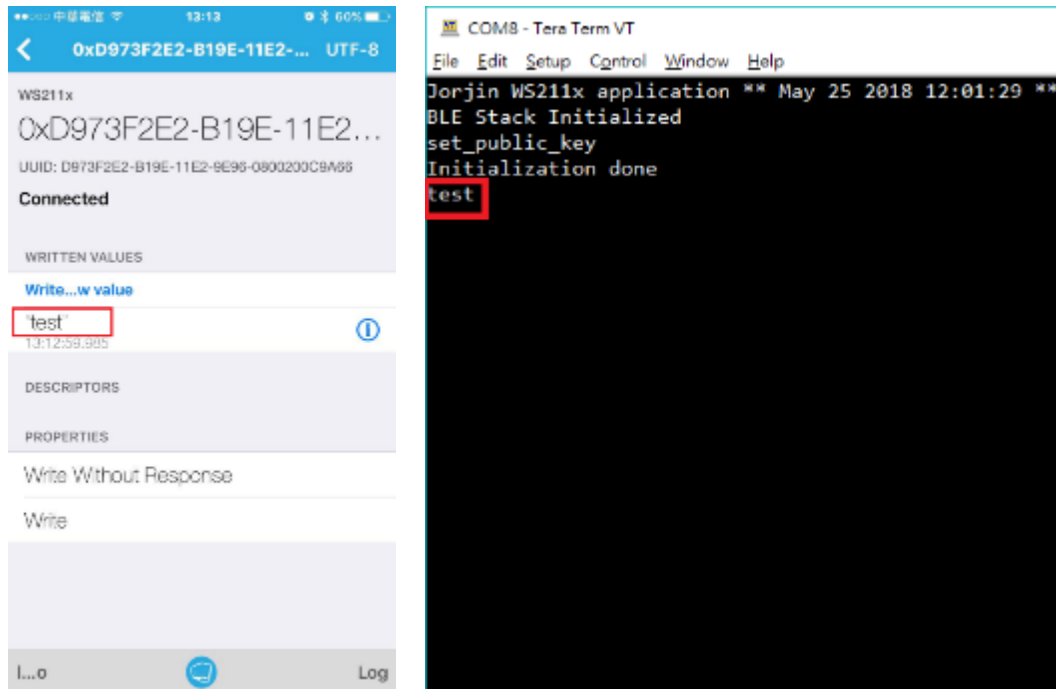
4.2.7 Function Testing

1. Written value via BLE to module.

Please set new-line both transmit and receive to CR+LF.

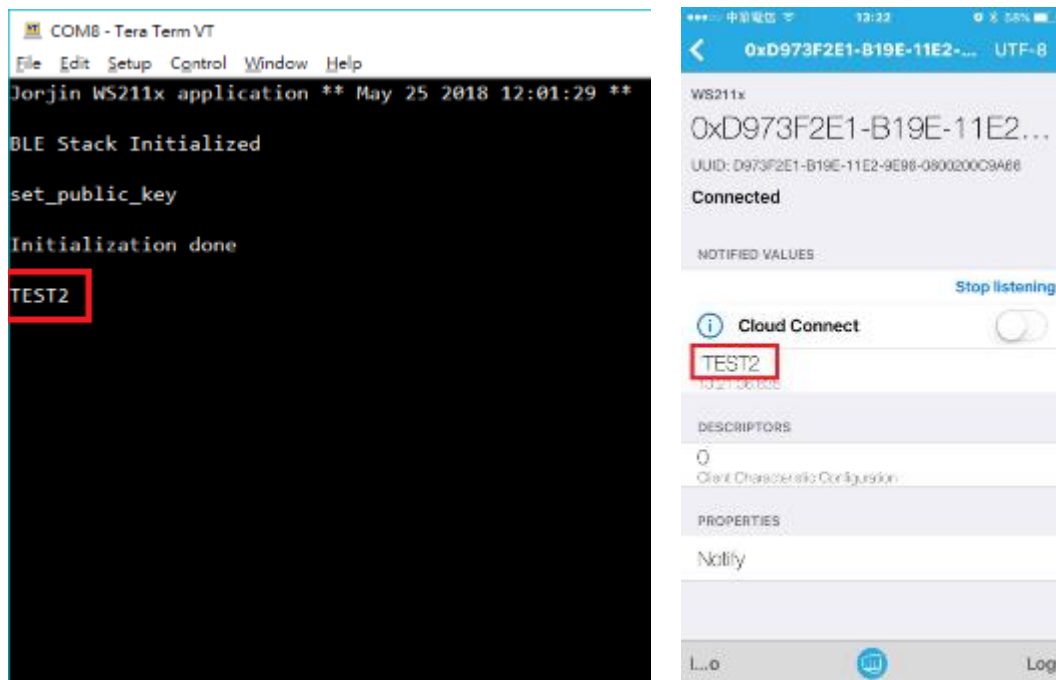


Application → module.

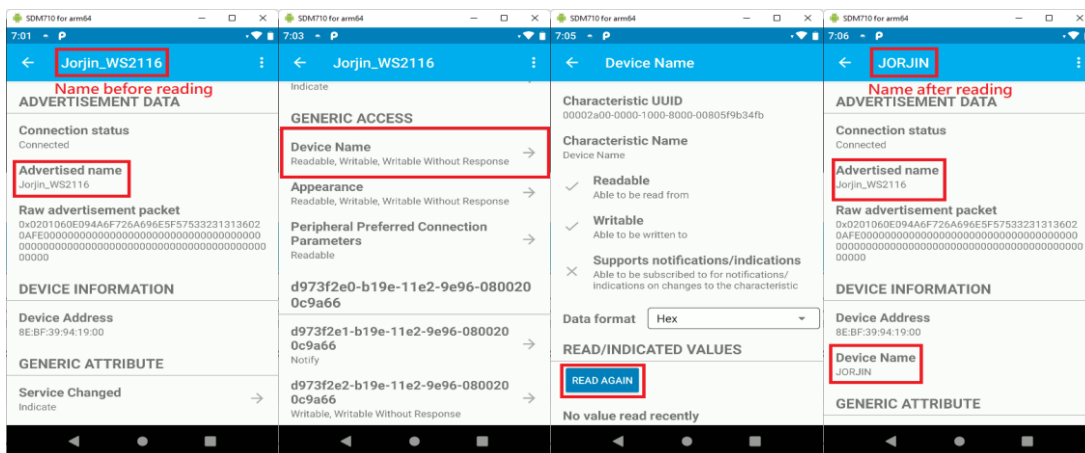
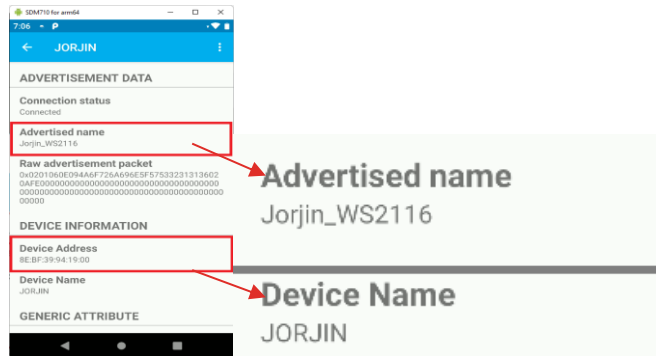


Module → application.

Push button or enable timer (Implement in while loop)



2. BLE name (device name) and local name (advertised name)



NOTE: The local name could only be changed by modifying the **local_name** variable in **Make_Connection()** at **chat.c** in the SDK. But, in the new AT command FW, it can also be changed through the **AT+NAME** command.

3. BLE Tx power parameter (bluenrg1_api.h)

```

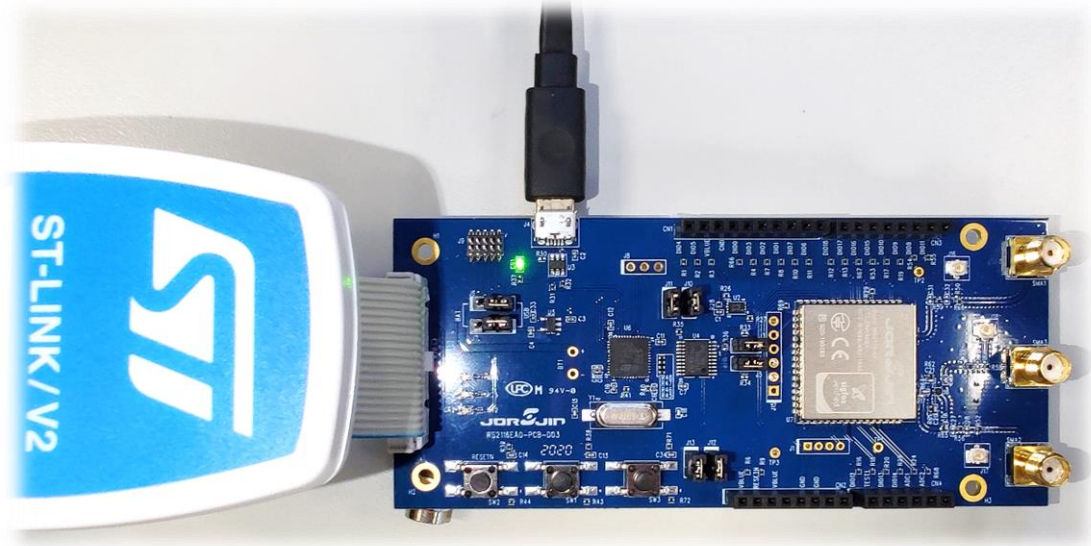
/**
 * @brief This command sets the TX power level of the device. By controlling the
 *        EN_HIGH_POWER and the PA_LEVEL, the combination of the 2 determines
 *        the output power level (dBm). When the system starts up or reboots,
 *        the default TX power level will be used, which is the maximum value of
 *        8 dBm. Once this command is given, the output power will be changed
 *        instantly, regardless if there is Bluetooth communication going on or
 *        not. For example, for debugging purpose, the device can be set to
 *        advertise all the time. And use this command to observe the signal
 *        strength changing. The system will keep the last received TX power
 *        level from the command, i.e. the 2nd command overwrites the previous
 *        TX power level. The new TX power level remains until another Set TX
 *        Power command, or the system reboots.
 * @param En_High_Power Enable High Power mode. High power mode should be
 *        enabled only to reach the maximum output power.
 *        Values:
 *        - 0x00: Normal Power
 *        - 0x01: High Power
 * @param PA_Level Power amplifier output level. The allowed PA levels depends
 *        on the device (see user manual to know which output power is expected
 *        at a given PA level). The obtained output power can also depend on PCB
 *        layout and associated components.
 *        Values:
 *        - 0x00 ... 0x31
 * @retval Value indicating success or error code.
 */
tBleStatus aci_hal_set_tx_power_level(uint8_t En_High_Power,
                                     uint8_t PA_Level);

```

5 FW OTA

5.1 ST-LINK

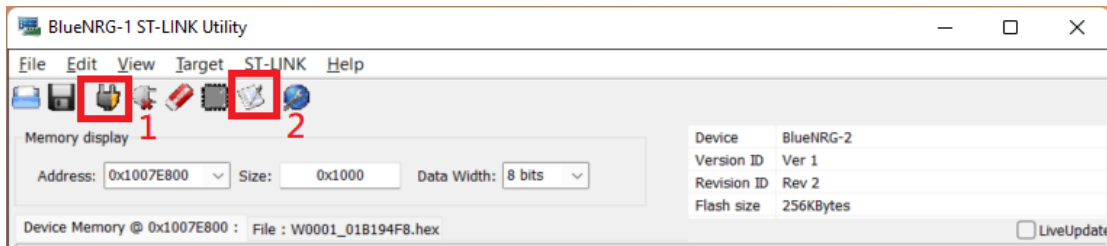
WS2116 setup, connect power and ST-link then open BlueNRG-1 ST-LINK Utility.



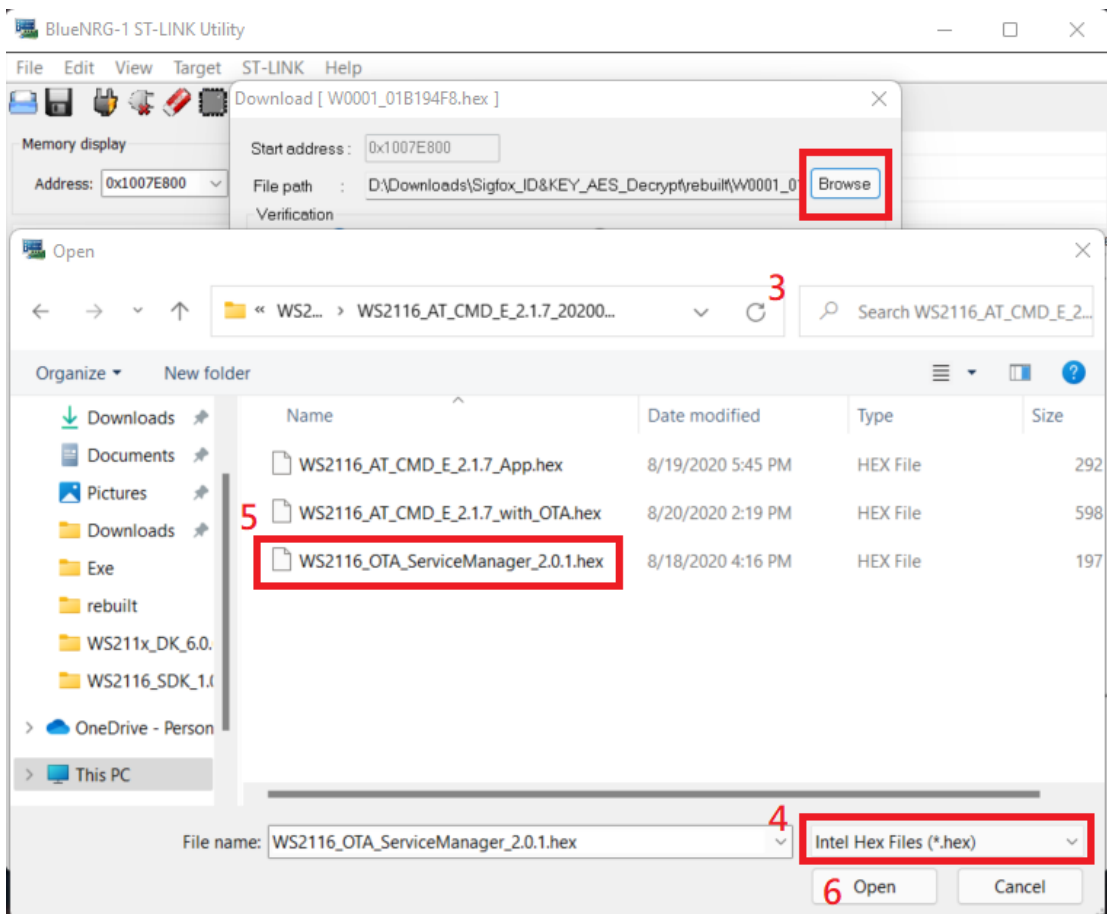
(EVB demo, module has to connect correct pin yourself)



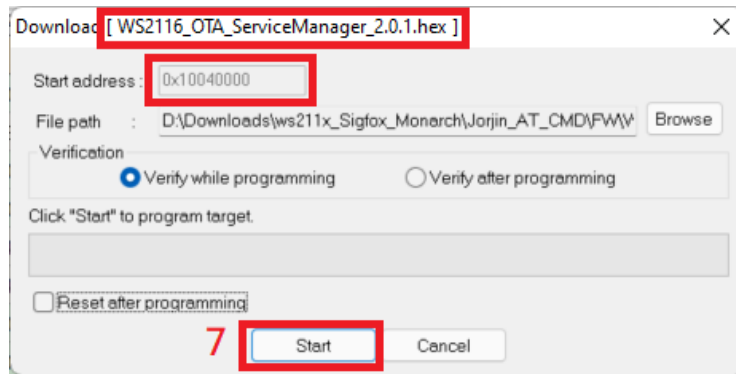
1. Connect to the target
2. Program verify (WS2116_OTA_ServiceManager_2.0.1.hex)



3. Browse and select file
4. Setting filter "*.hex" files
5. Select "WS2116_OTA_ServiceManager_2.0.1.hex"
6. Open file.



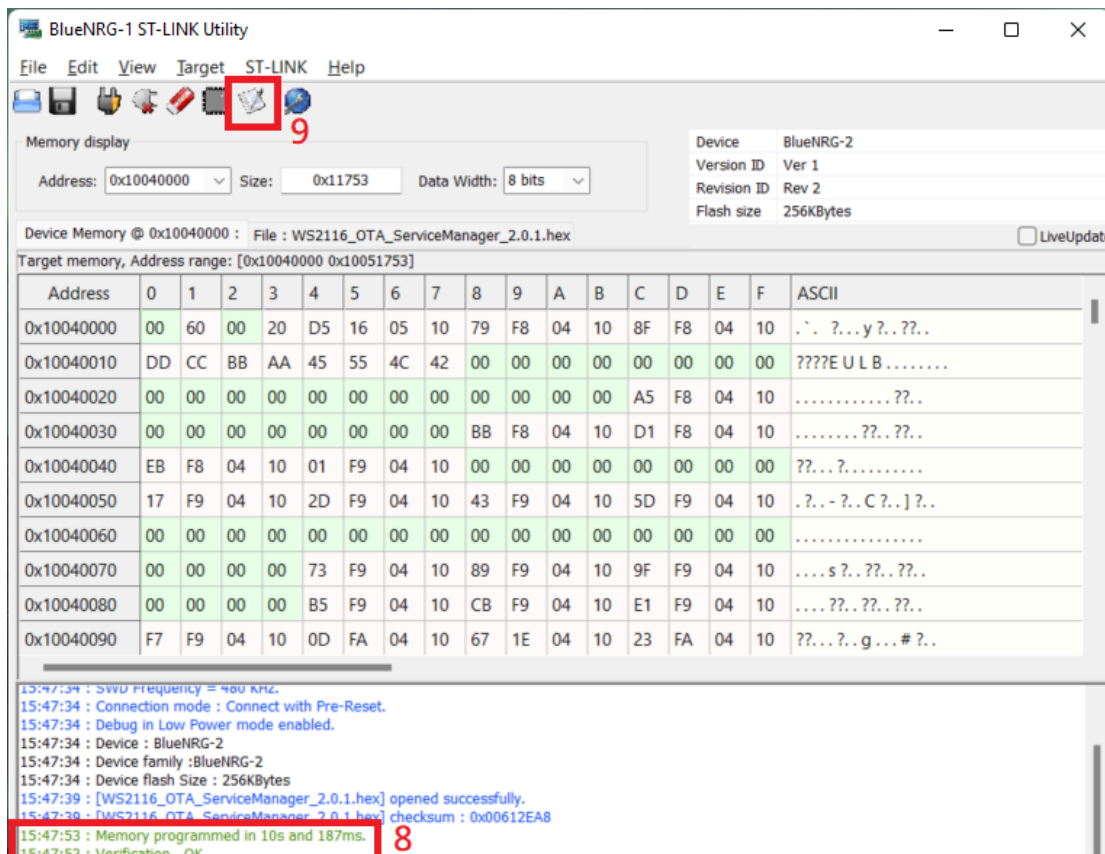
7. Check file \ start address and start programming



8. Check success message

9. Program verify (WS2116 firmware)

In OTA_ServiceManager FW also can use cell phone to program FW.

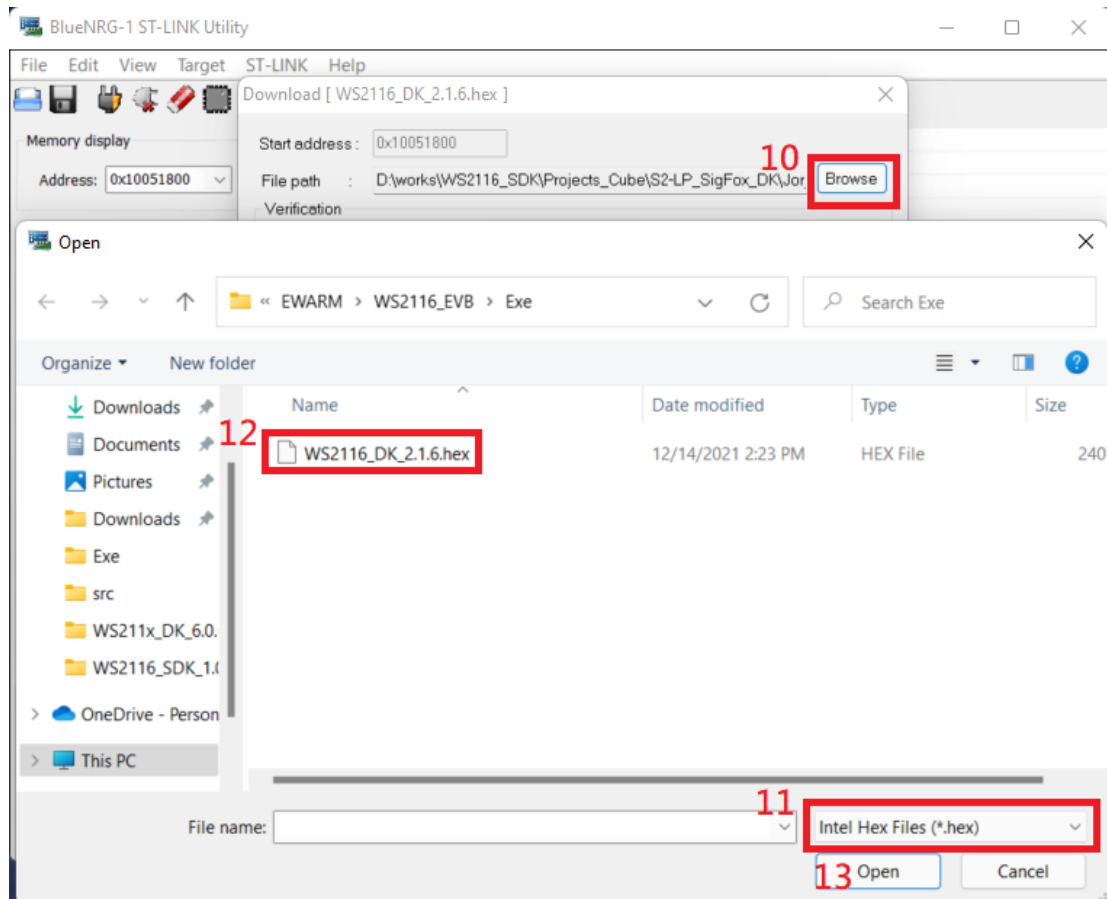


10. Browse and select file

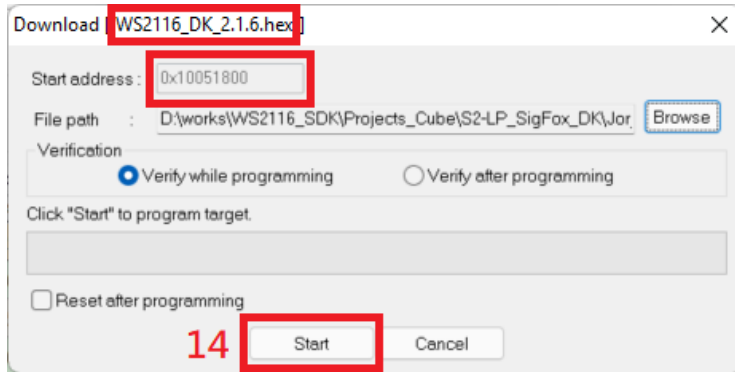
11. Setting filter "*.hex" files

12. Select "(WS2116 firmware).hex" (Must be *.hex file)

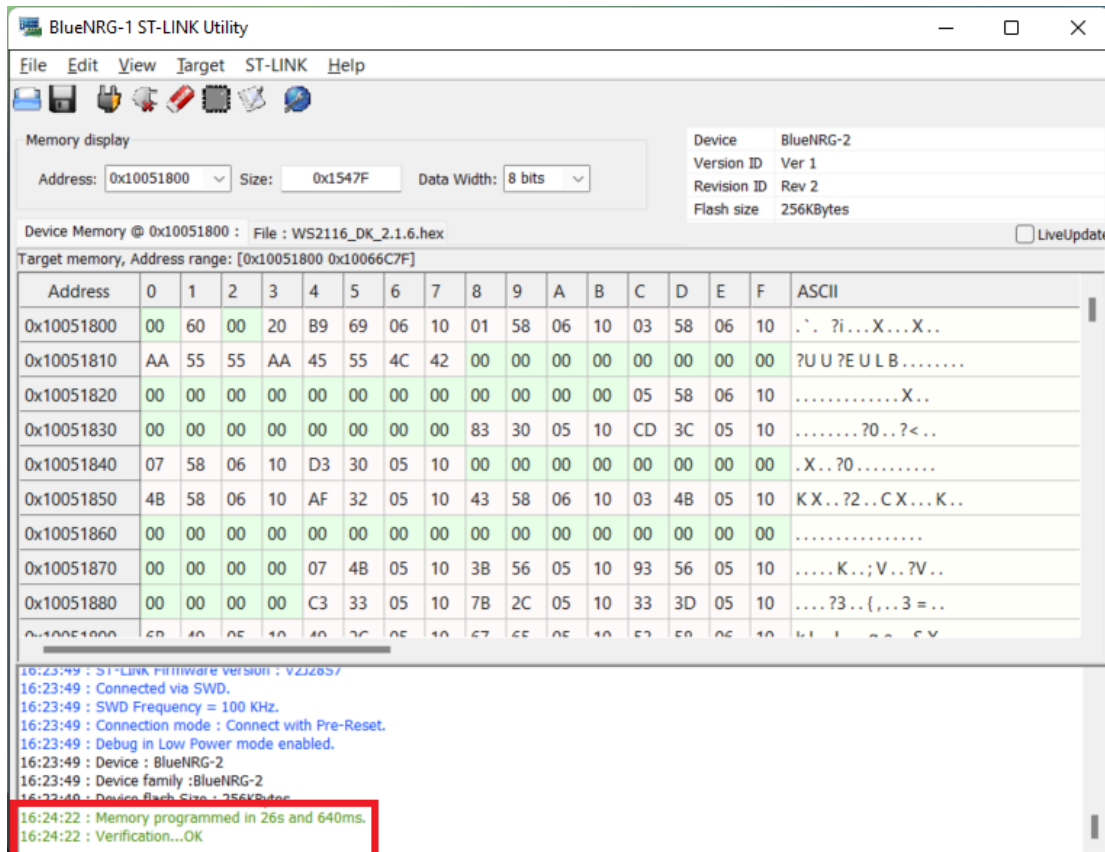
13. Open file.



14. Check file \ start address and start programming



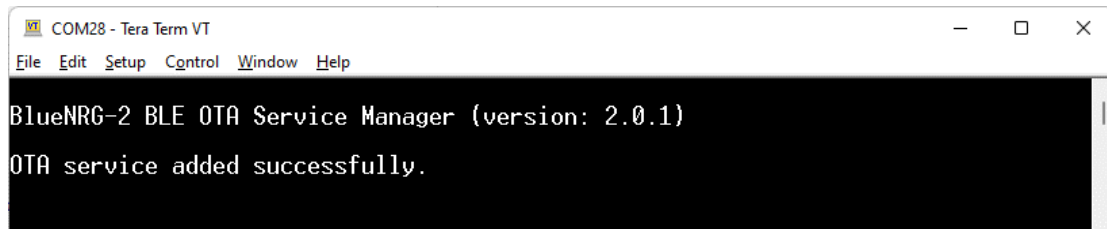
15. Check success message



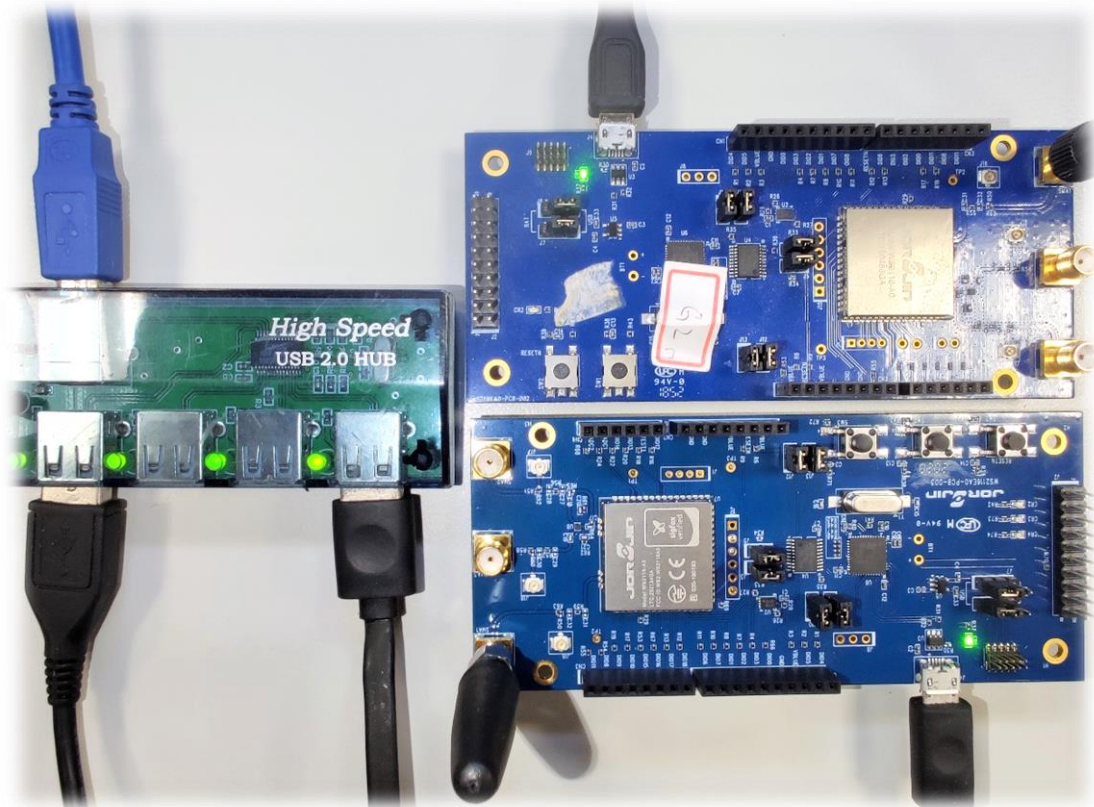
5.2 OTA

1. Press SW1 to return OTA service manager. Please ensure SW1 is setting “OTA_Jump_To_Service_Manager_Application”.

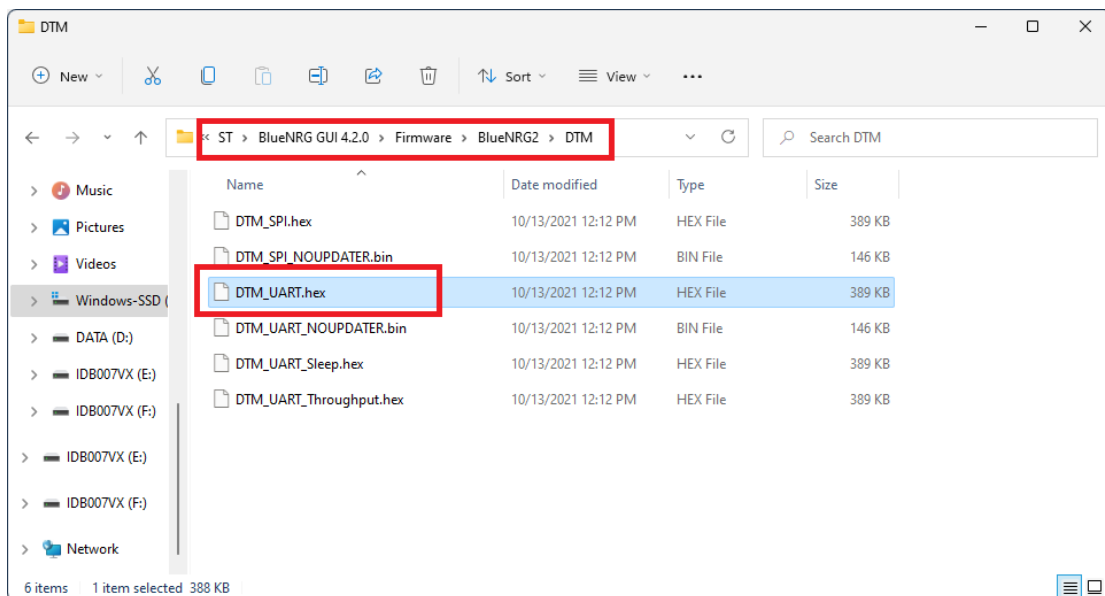
```
main.c x ST_Sigfox_Init.c chat.c rf_api.c bluenrg1_api.h ble_status.  
  
if (but_pressed)  
{  
#if ST_USE_OTA_SERVICE_MANAGER_APPLICATION  
printf("Enter OTA\r\n");  
SdkDelayMs(100);  
OTA_Jump_To_Service_Manager_Application();  
#endif /* ST_USE_OTA_SERVICE_MANAGER_APPLICATION */  
LedBlink(LED2, 5);  
}
```



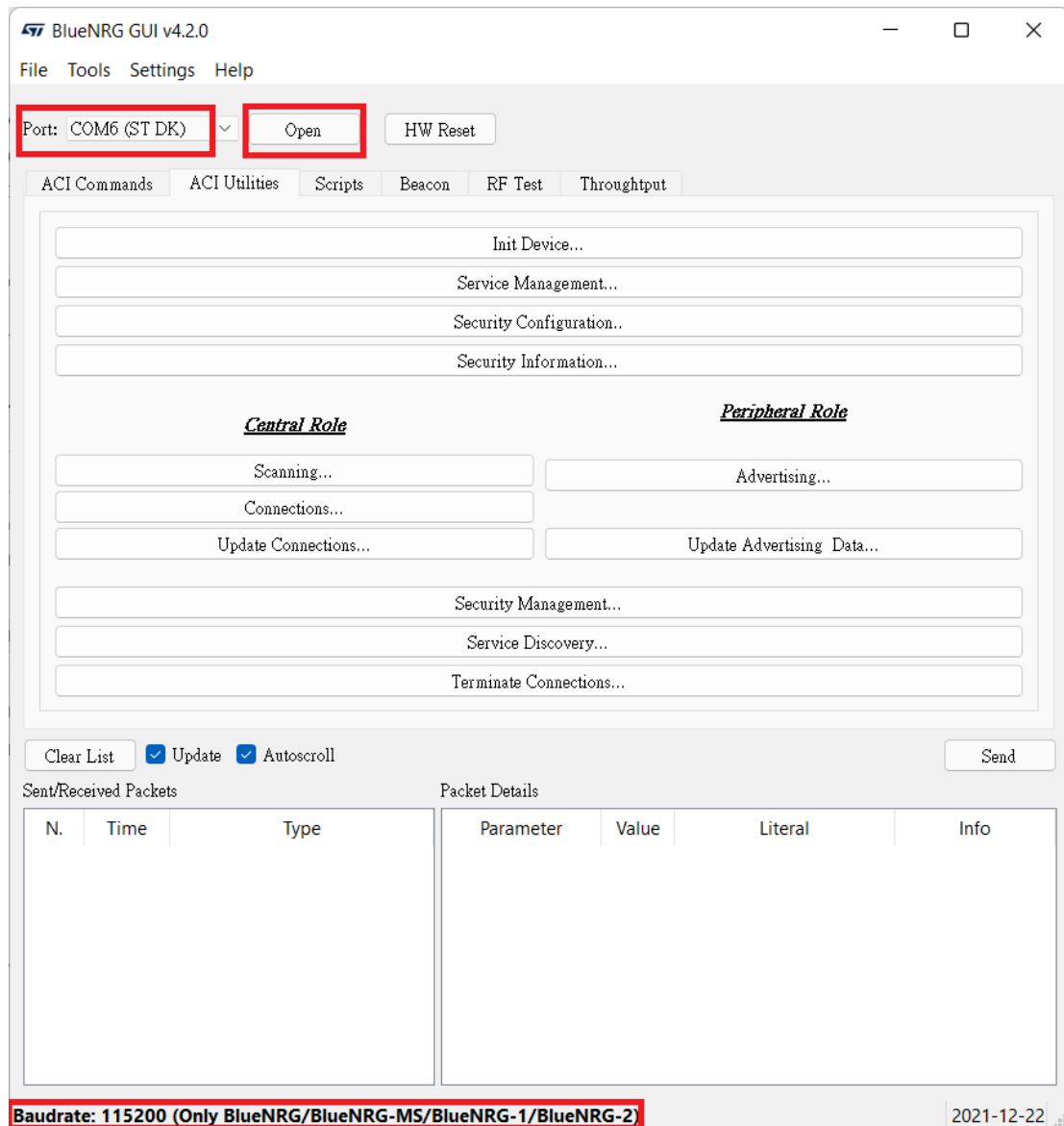
2. Prepare another WS2116 EVB to the USB connectors on a PC.



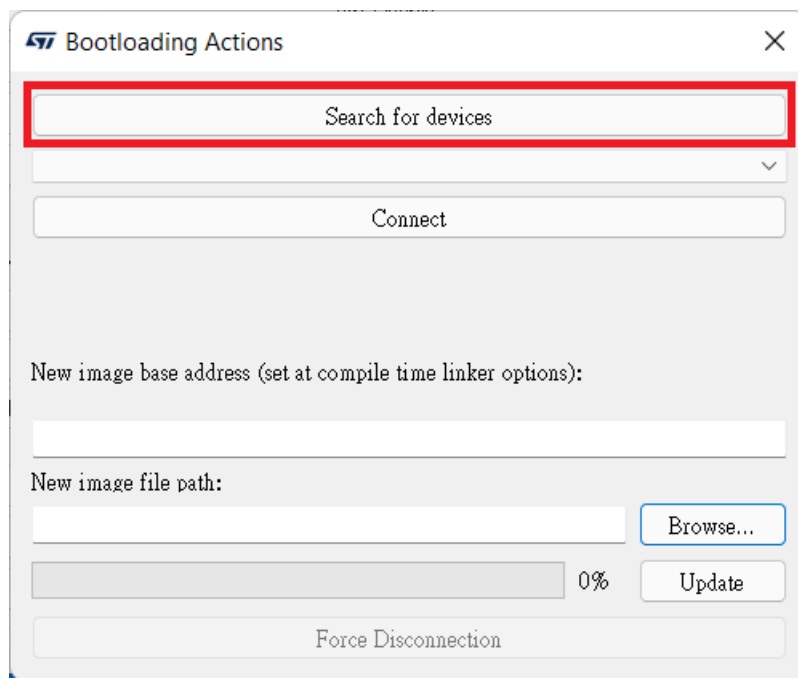
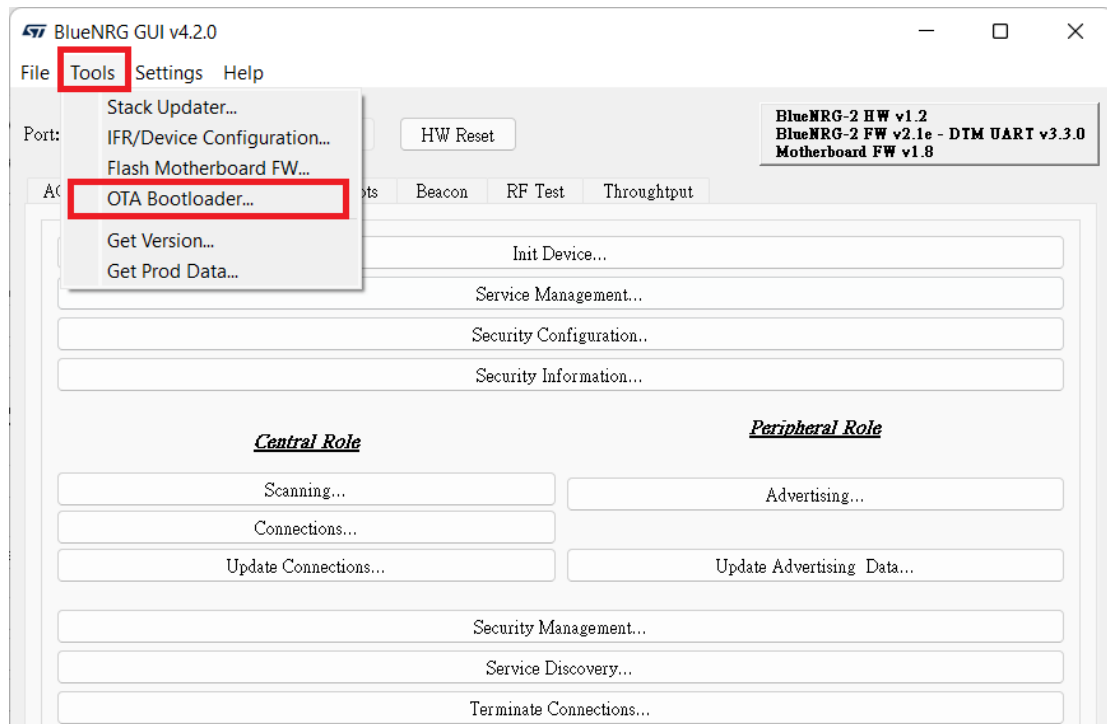
3. Program the second WS2116 EVB with the “DTM_UART.hex” to be used with the BlueNRG GUI



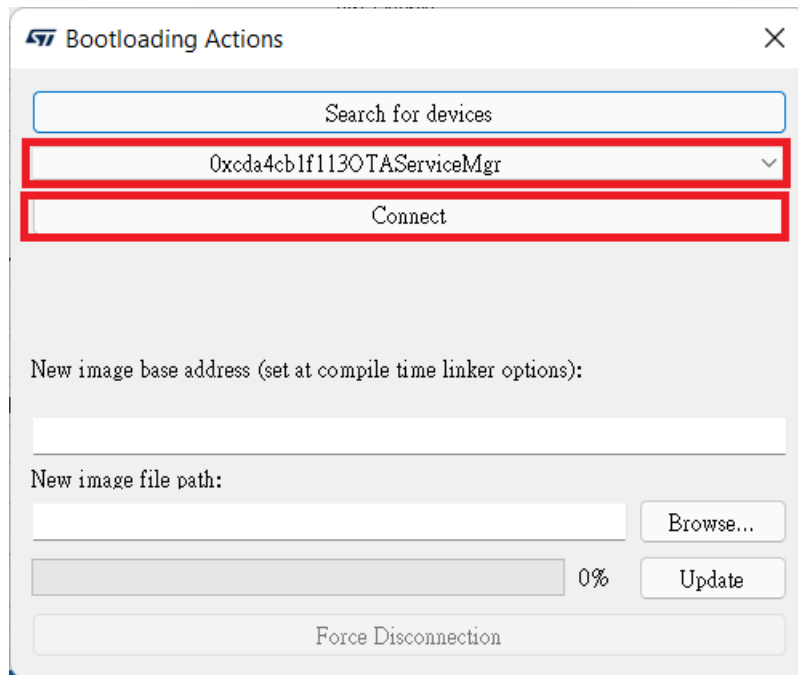
- Open the BlueNRG GUI on the PC and select the COM port related to the second WS2116 EVB configured on step 9, through the drop down “Port” and press ‘Open’. And make sure the baudrate is set to 115200.



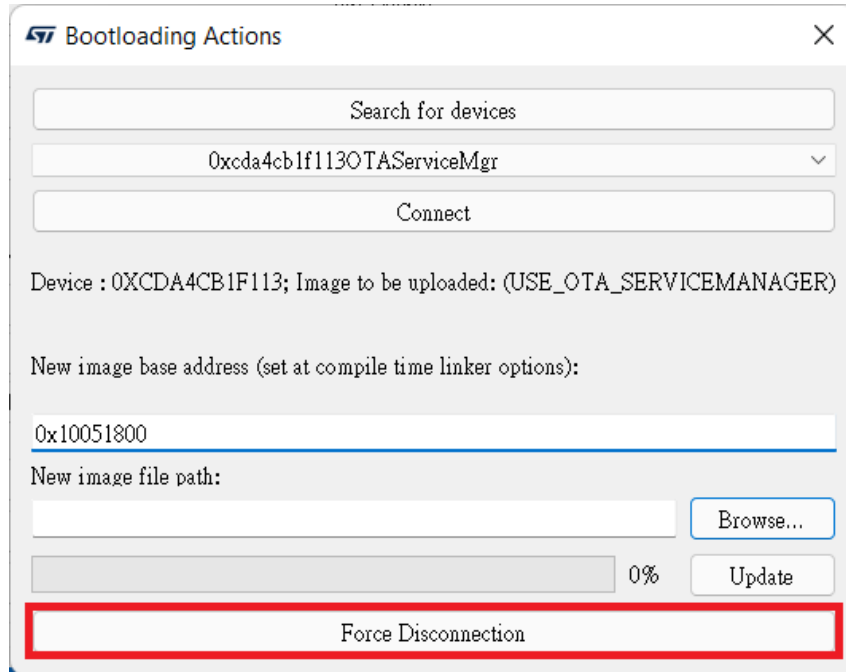
- On BlueNRG GUI select 'Tools' -'OTA bootloader' to open up the dialog containing OTA FW upgrade actions and press 'Search for devices'.



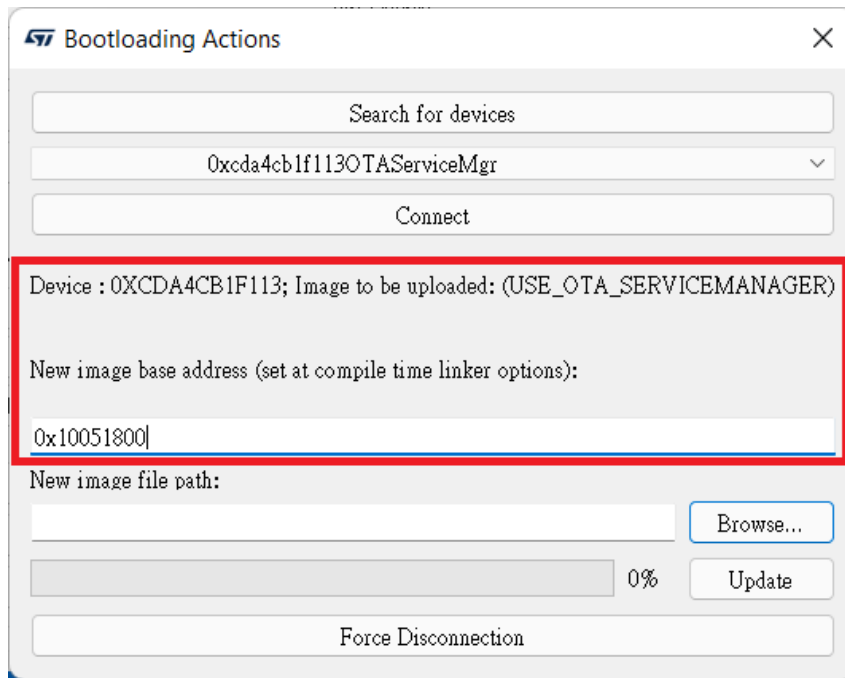
6. After 'Search for devices', the GUI starts the discovery process and gets back with some information about the address and application names of the devices running OTA FW upgrade service within the radio range.
7. Once the previous process ends, the device list can be opened up through the combo box arrow below the 'Search for devices' button and the user can find the device running the OTA "Service Manager" and press 'Connect'.



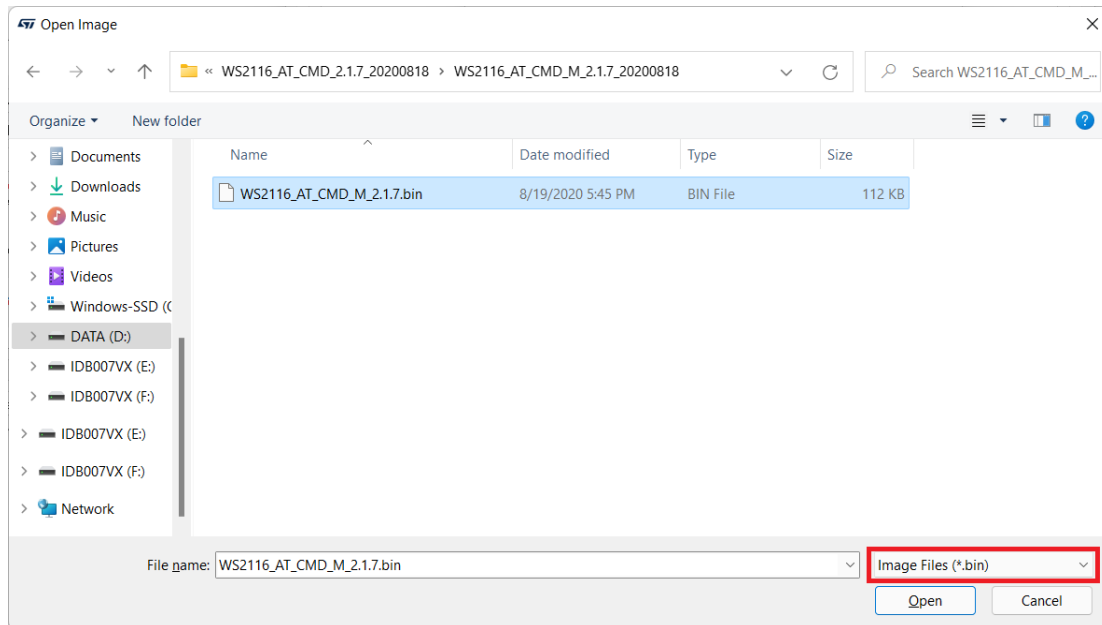
- If user realizes he has connected the wrong device, he can just press the 'Force Disconnection' button and get back to the device selection within the combo box.

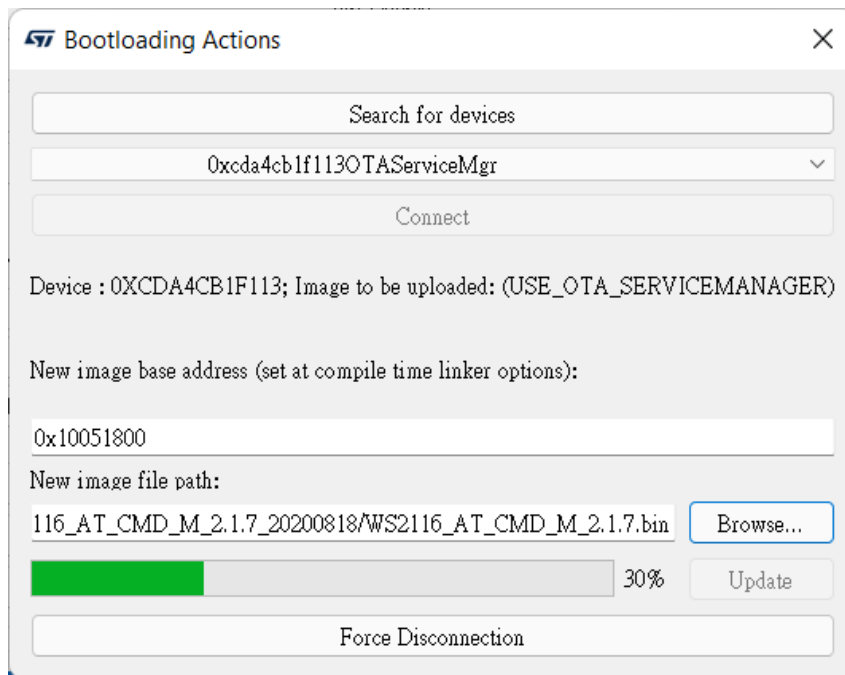
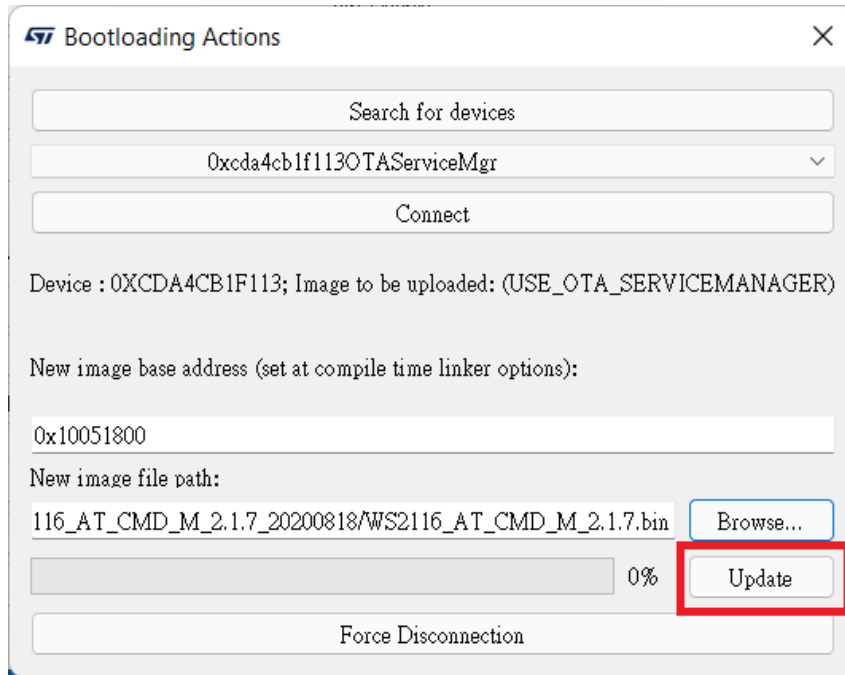


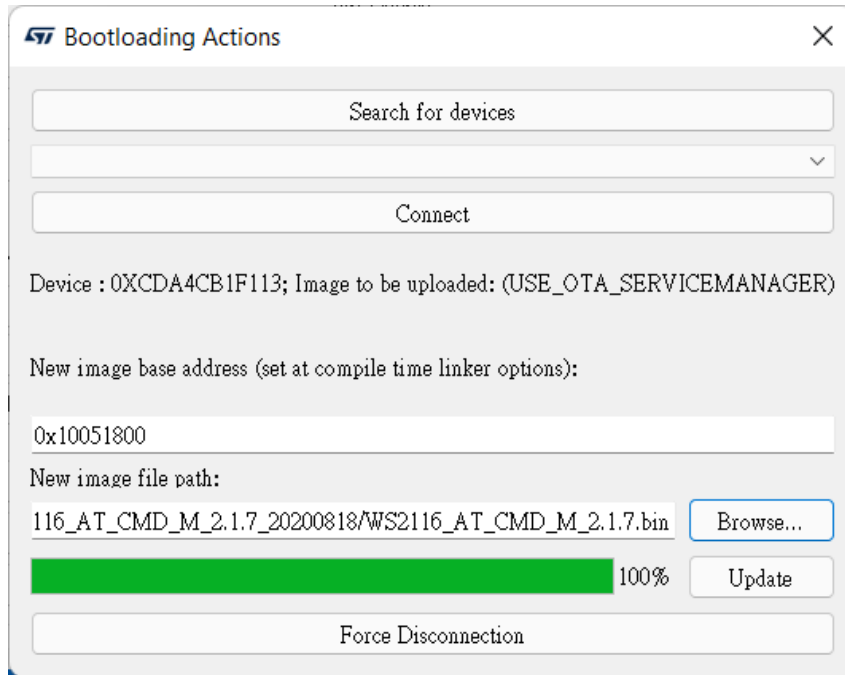
- After the device selection, connection through 'Connect' button and reading of the related free memory range. Then press 'Browse' to select image.



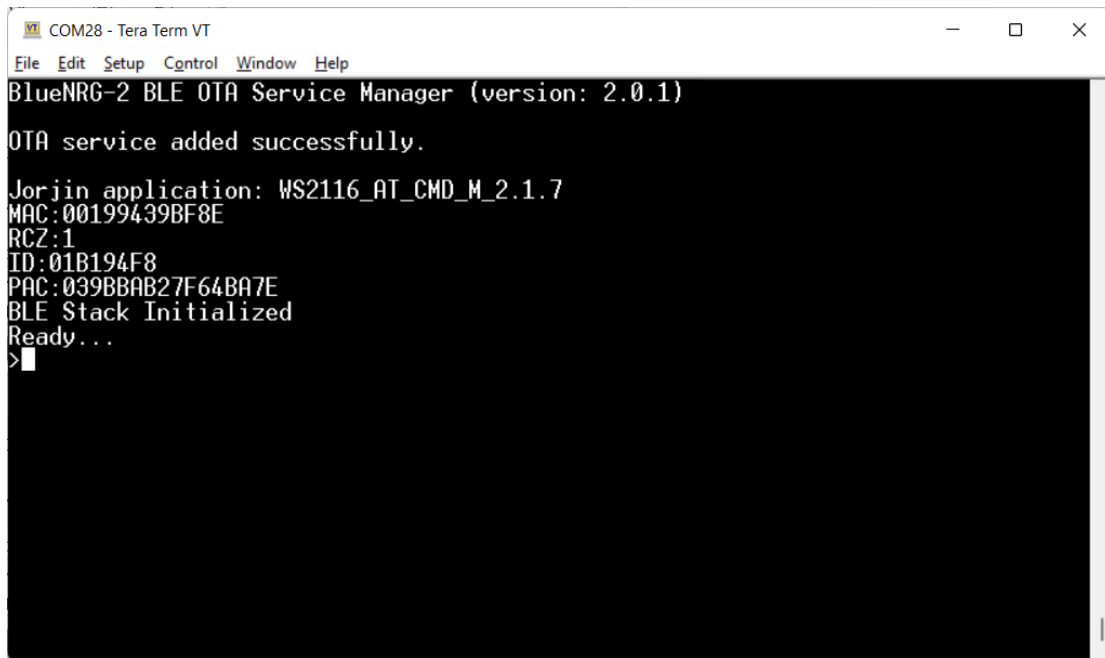
- Choose *.bin and select what you want to program. Make sure the image file and press "update".







11. OTA success will show firmware booting log.



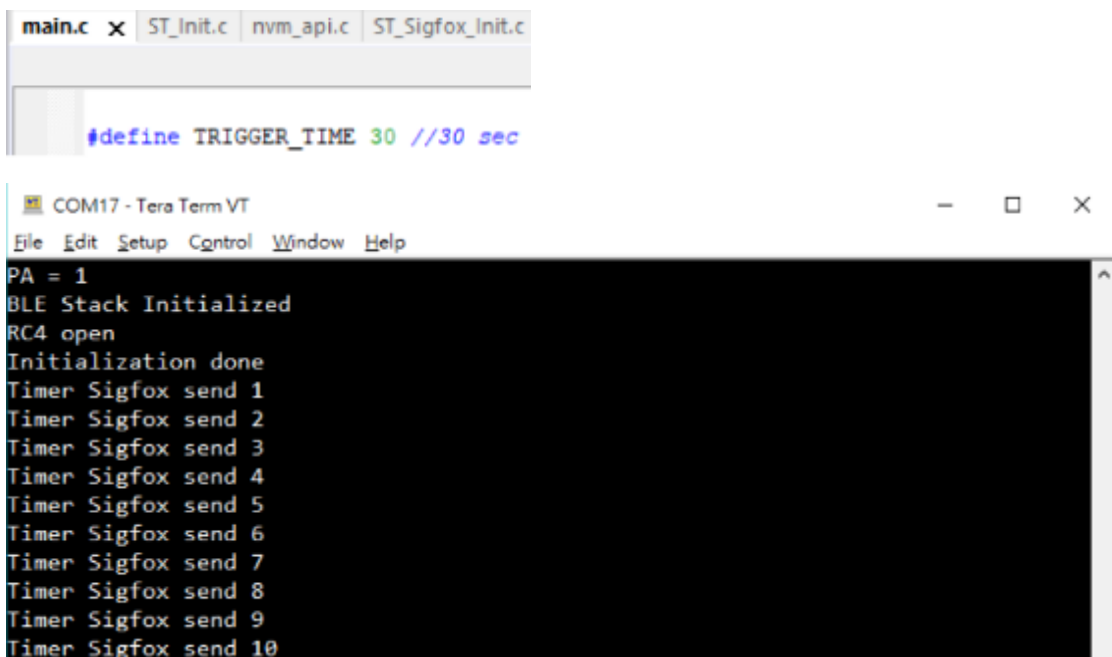
6 HOW TO EVALUATE WITH SIGFOX FUNCIONTN

6.1 USE PUBLIC KEY (USE SDR DONGLE)

1. Hold SW1
2. Press and release SW2
3. Release SW1
4. Check the log as below



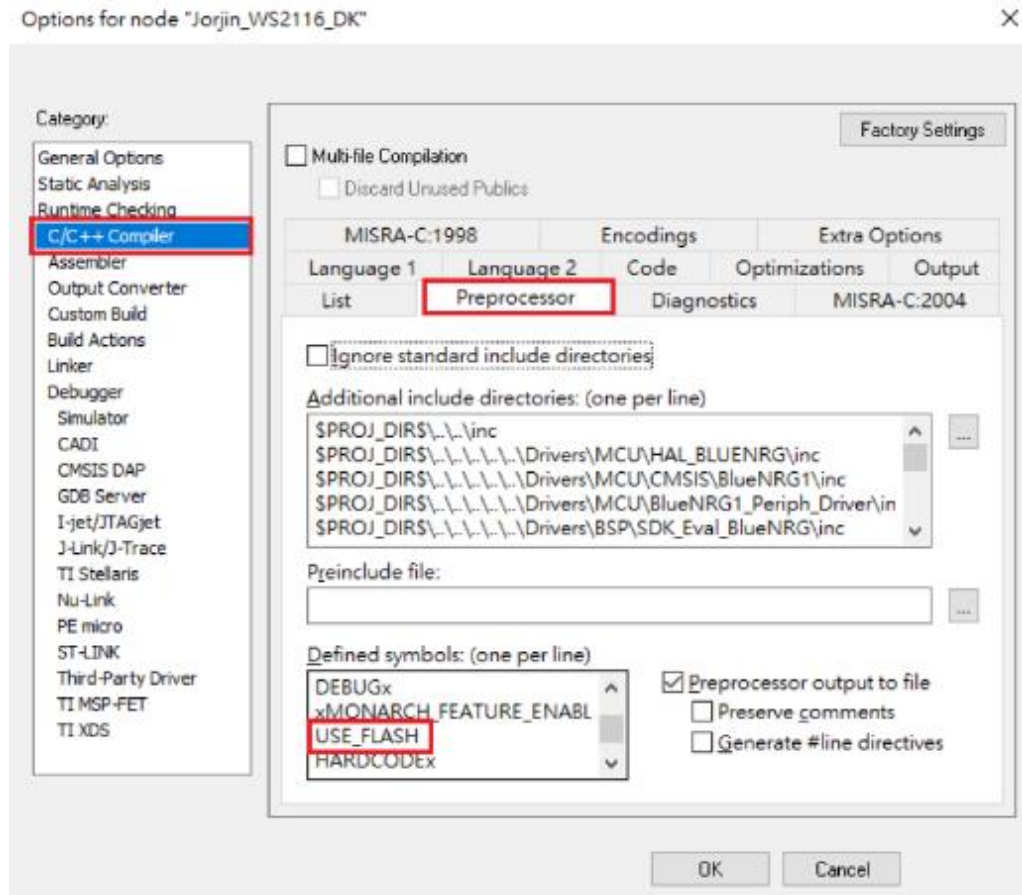
5. Set TRIGGER_TIMER as a trigger unit to send frame to sigfox backend.



Device ID	Time	Sequence number	Data / Decoding	LQI	Callbacks
FEDCBA98	2018年12月12日 16:26:42	162	1300	📶	0
FEDCBA98	2018年12月12日 16:25:42	161	1200	📶	0
FEDCBA98	2018年12月12日 16:24:42	160	1100	📶	0
FEDCBA98	2018年12月12日 16:23:42	159	1000	📶	0
FEDCBA98	2018年12月12日 16:22:42	158	0000	📶	0
FEDCBA98	2018年12月12日 16:21:42	157	0e00	📶	0
FEDCBA98	2018年12月12日 16:20:42	156	0e00	📶	0
FEDCBA98	2018年12月12日 16:19:42	155	0e00	📶	0
FEDCBA98	2018年12月12日 16:18:42	154	0e00	📶	0
FEDCBA98	2018年12月12日 16:17:42	153	0e00	📶	0

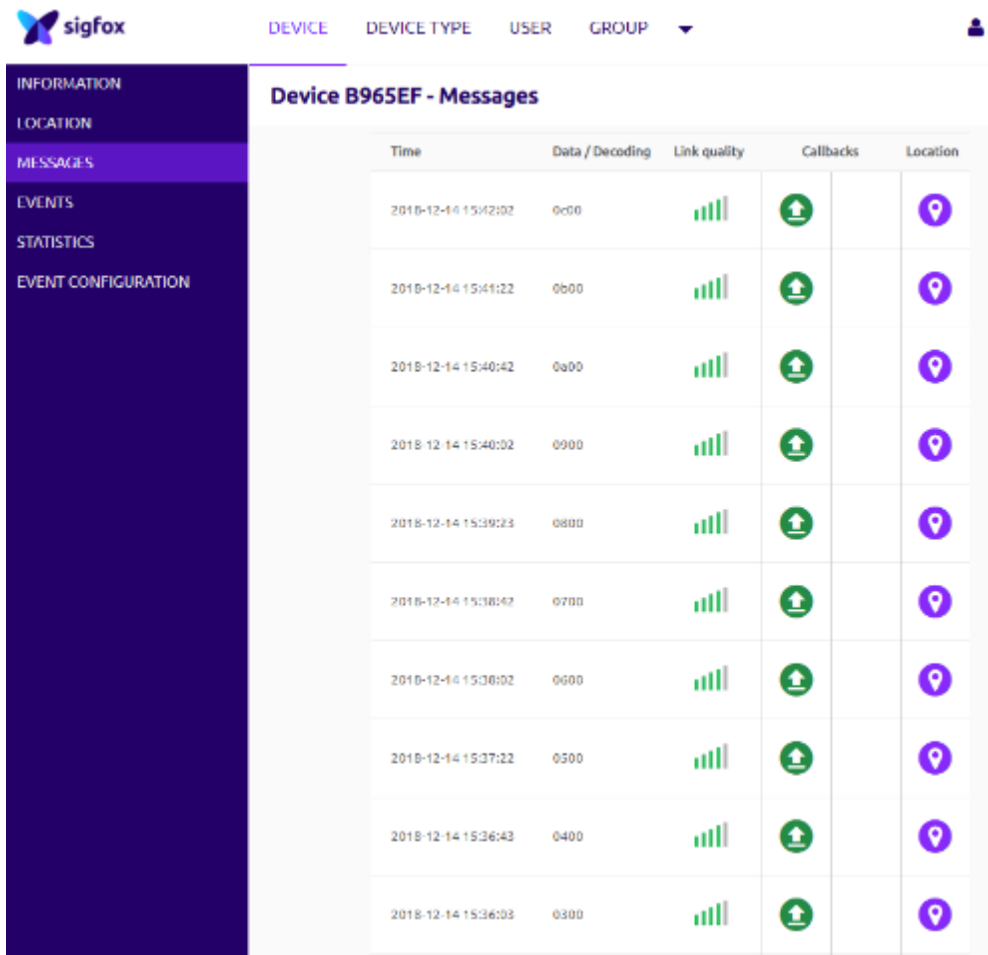
6.2 USE PRIVATE KEY

Both EEPROM/flash have stored SIGFOX information (id, pac and key etc.), depend on module or EVB you use to define symbol to read SIGFOX information (id, pac and key etc.) which Jorjin provide to simulate sigfox function.



```

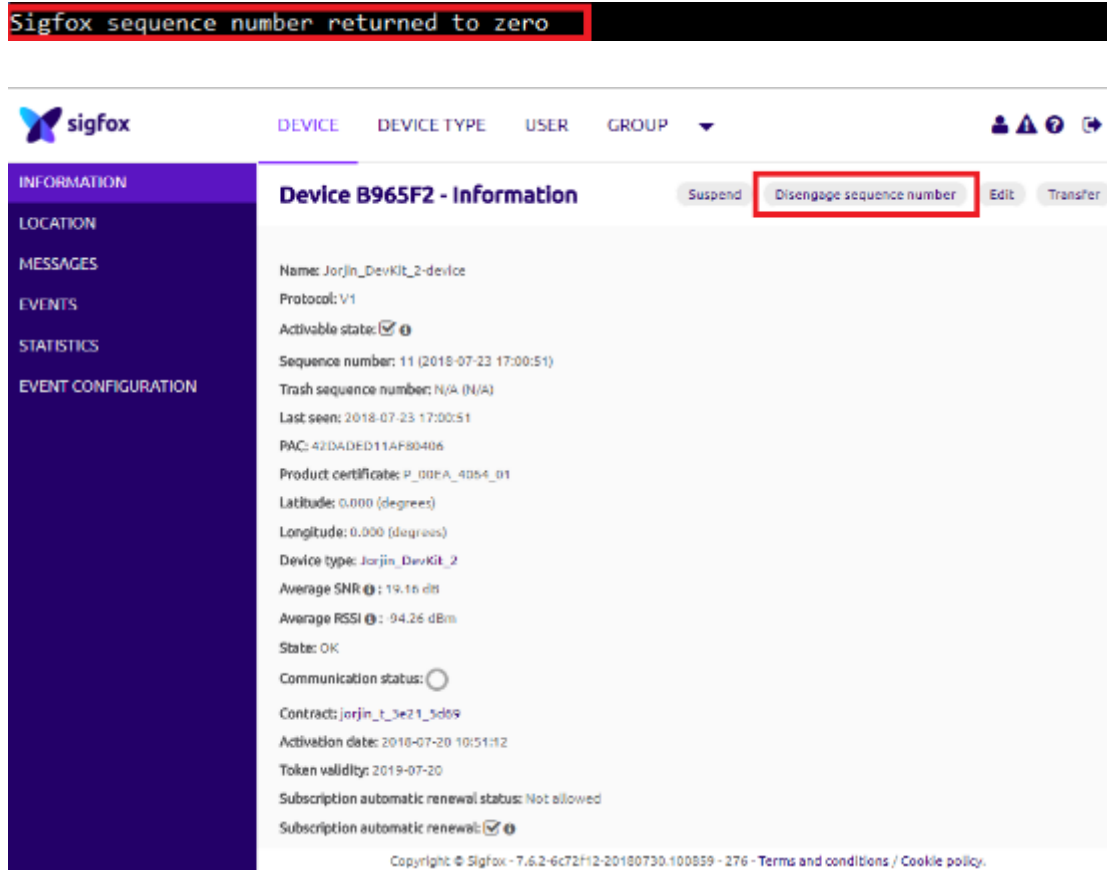
COM39 - Tera Term VT
File Edit Setup Control Window Help
Jorjin WS2116 application ** Jan 30 2019 16:56:46 **
BLE Stack Initialized
ID = 00B965EF
PAC = 0000000051A4A31B
RC4 open
Sigfox send 1
Sigfox send 2
Sigfox send 3
Sigfox send 4
Sigfox send 5
Sigfox send 6
Sigfox send 7
Sigfox send 8
Sigfox send 9
Sigfox send 10
Sigfox send 11
Sigfox send 12
Sigfox send 13
Sigfox send 14
Sigfox send 15
  
```



The screenshot shows the Sigfox web interface for a device named B965EF. The left sidebar contains navigation options: INFORMATION, LOCATION, MESSAGES, EVENTS, STATISTICS, and EVENT CONFIGURATION. The main content area is titled "Device B965EF - Messages" and displays a table of message logs.

Time	Data / Decoding	Link quality	Callbacks	Location
2018-12-14 15:42:02	0c00			
2018-12-14 15:41:22	0b00			
2018-12-14 15:40:42	0a00			
2018-12-14 15:40:02	0900			
2018-12-14 15:39:22	0800			
2018-12-14 15:38:42	0700			
2018-12-14 15:38:02	0600			
2018-12-14 15:37:22	0500			
2018-12-14 15:36:43	0400			
2018-12-14 15:36:03	0300			

If show information below, please click “ Disengage sequence number” on Sigfox backend. Then the sequence number will match.



6.3 READ SIGFOX ID AND PAC

This section is based on SDK 3.32+ version. If you are using old SDK. Please upgrade.

Don't define “HARDCODE” to read module data SIGFOX information, UART will show ID and PAC.



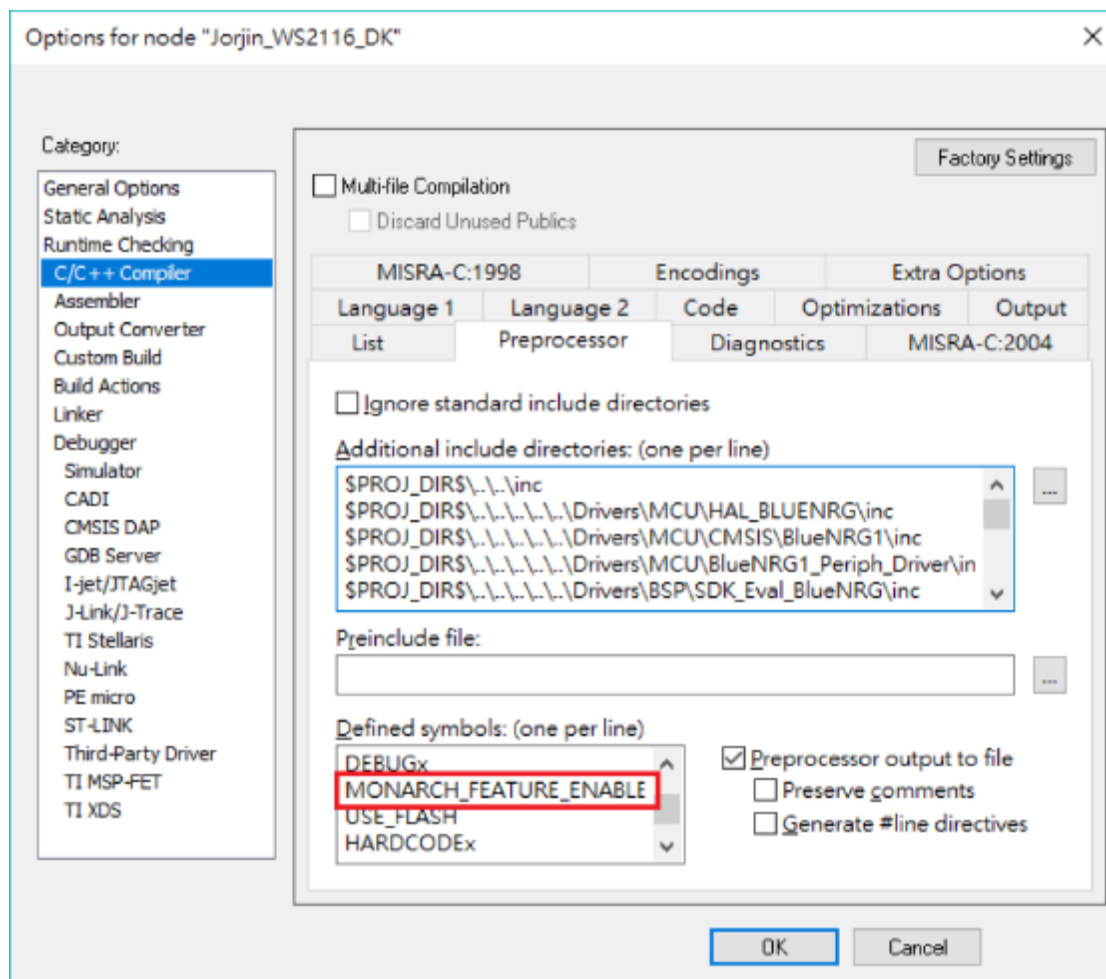
6.4 SCAN MODE

WS2116 supply a function which scan of the air to detect a Sigfox Beacon to use in different zone. To enable function please define “MONARCH_SCAN”.

This SDK is designed that device always in sleep mode when specific timer / unit time wakeup to scan where zone is it.

Wakeup source have IO-11 (UART), IO-12 (Button) and timer. Timer interval setting “WAKEUP_TIMEOUT”.

```
#define WAKEUP_TIMEOUT 30000 //30 sec, MAX = 5242879(5242 sec)
```

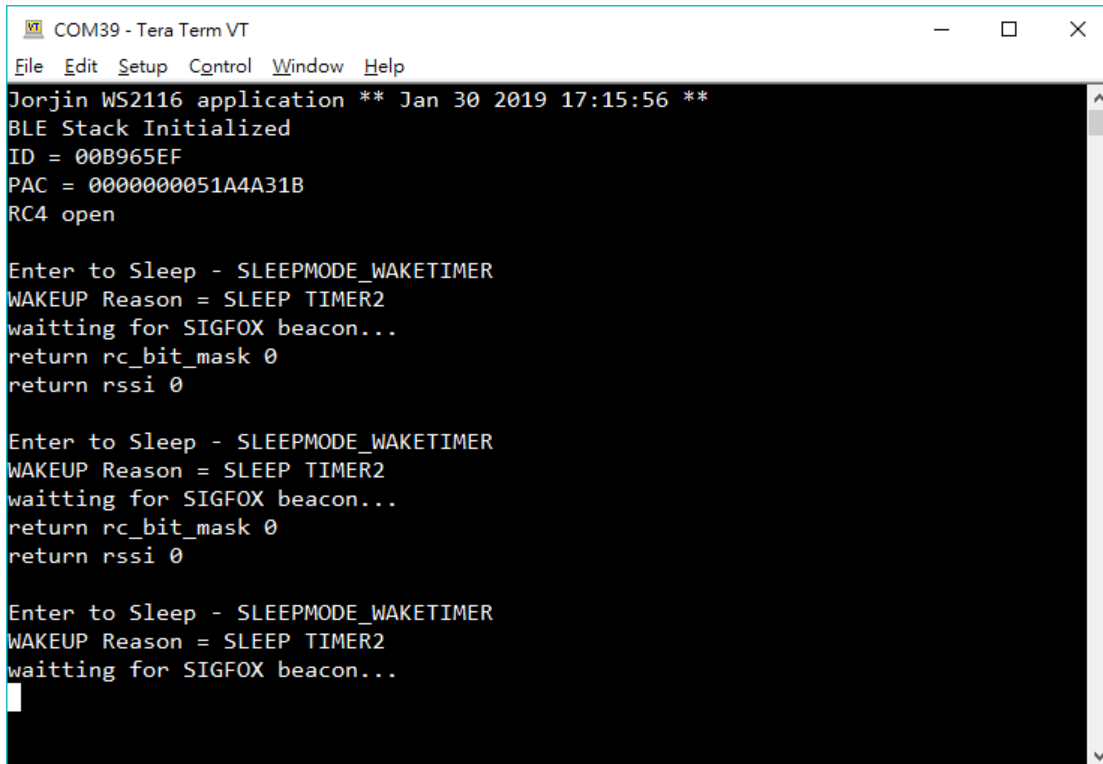



```
if(callbackFlag)
{
    callbackFlag = 0;
    sleep_timer();
}

if(wakeupFlag==1)
{
    wakeupFlag=0;

    SIGFOX_API_close();
    resetRcz=1;
    //Begin Monarch Scan session
    SIGFOX_MONARCH_API_execute_rc_scan (63, 5, SFX_TIME_M, callback_for_found);

    printf("waitting for SIGFOX beacon...\r\n");
}
}
```



```
COM39 - Tera Term VT
File Edit Setup Control Window Help
Jorjin WS2116 application ** Jan 30 2019 17:15:56 **
BLE Stack Initialized
ID = 00B965EF
PAC = 0000000051A4A31B
RC4 open

Enter to Sleep - SLEEPMODE_WAKETIMER
WAKEUP Reason = SLEEP TIMER2
waitting for SIGFOX beacon...
return rc_bit_mask 0
return rssi 0

Enter to Sleep - SLEEPMODE_WAKETIMER
WAKEUP Reason = SLEEP TIMER2
waitting for SIGFOX beacon...
return rc_bit_mask 0
return rssi 0

Enter to Sleep - SLEEPMODE_WAKETIMER
WAKEUP Reason = SLEEP TIMER2
waitting for SIGFOX beacon...

```