

OpenSource with Market Ready Enclosure

USB840 Series dongles support Bluetooth 5, Thread, or Zigbee radio protocols. They are in market ready enclosure and pre-certified. OpenSource codes are available to expedite your firmware development.

Dongles are preloaded with Nordic USB CDC ACM secure boot loader. It can be booted and firmware updated from a PC.

USB840F

USB840F embedded a long-range module BT840F. Range between 2 modules (not in enclosure) is 2300 meters at 125 Kbps. It has 2 user definable LED indicators and 2 buttons for reset or set to boot loader mode.



USB840X

Using BT840X with +21 dBm TX, BWG832X can penetrate multiple walls or be used in wide area Bluetooth mesh applications.

Size and features of USB840X are the same as that of USB840F.

USB840M

Using compact module BC840M, USB840M is a compact dongle. It has an LED indicator and a button to set to boot loader mode for application firmware update.

Nordic nRF52 Development Environment

Nordic nRF52 development environment should be used. Open source codes are available at:

<http://www.fanstel.com/download-opensource/>

USB DFU

The Bluetooth dongle is supported by an USB DFU Firmware Upgrade feature. This allows for in the field updates of application firmware and SoftDevice.

Module Bluetooth Range

Bluetooth ranges are measured between 2 modules not in enclosure. Enclosure will reduce Bluetooth range by about 20%. More obstruction in typical dongle application environment than our testing environment, Bluetooth range is shorter than listed below.

Bluetooth range between an USB840F/X and a smartphone is much shorter because smartphone Bluetooth interface is not long range.

USB Dongle Summaries

Dongle	USB840F	USB840M	USB840X
BLE module	BT840F	BC840M	BT840X
Size	52x21x21mm	26x15x6.0mm	52x21x21mm
BT range, low interference	2300 meters at 125 Kbps	400 meters at 125 Kbps	>4500 meters at 125 Kbps
BT range, antenna at 1.52M	640 meters at 125 Kbps		1200 meters at 125 Kbps
BT range, low interference	1000 meters at 1Mbps		1170 meters at 1Mbps
BT range, antenna at 1.52M	390 meters at 1Mbps	135 meters at 1Mbps	900 meters at 1Mbps
FCC	X8WBT840F	X8WBC840M	X8WBT840X
Industrial Canada	4100A-BT840F	4100A-BC840M	4100A-BT840X
CE	Certified	Certified	
Japan TELEC	Pending	201-190140	
Australia	Pending	RCM certified	
QDID		108621	108621

USB840F/M/X BLE 5, Thread, Zigbee Dongles

Table Of Contents

1. Hardware Description	3
USB840F/X Hardware Description	3
USB840F/X Schematics	3
USB840M Hardware Description.....	4
USB840M Schematics.....	4
2. Firmware Development and Programming	5
Preloaded Bootloader and Private Key	5
Procedures to Program USB Bootloader.....	5
BT840F/X Hardware Setup for Programming Bootloader	7
USB840M	7
Programming Bootloader.....	7
Steps to install Fanstel demo HEX codes	9
Firmware to Control Power Amplifier in USB840X	9
Nordic Development Environment.....	12
Revision History	13
Contact Us	14

USB840F/M/X BLE 5, Thread, Zigbee Dongles

1. Hardware Description

USB840F/X Hardware Description

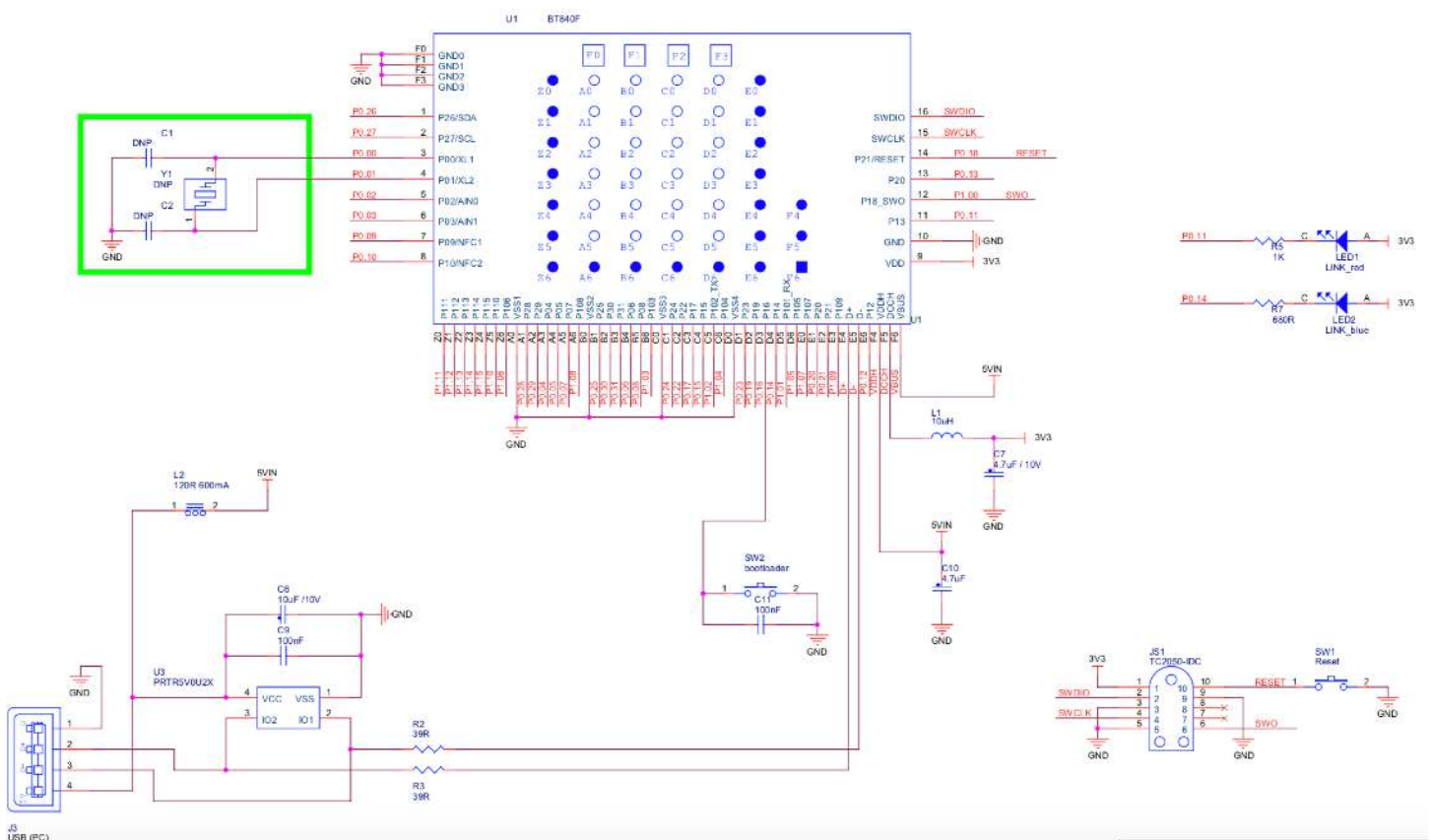
USB840F and USB840X use the same enclosure and host PCB. LED1 (red color) and LED2 (blue color) are not silkscreened on enclosure. Operation voltage of blue LED is higher than 1.8V DC output voltage of nRF52840 after **chiperase**. You must set DC output to 3V before using the blue LED.



You need to open USB840F/X enclosure to access bootloader button to load application program. Or, to reprogram dongle with your own bootloader and private key after **chiperase**.

USB840F/X Schematics

Schematics of USB840F and USB840X is below.



USB840M Hardware Description

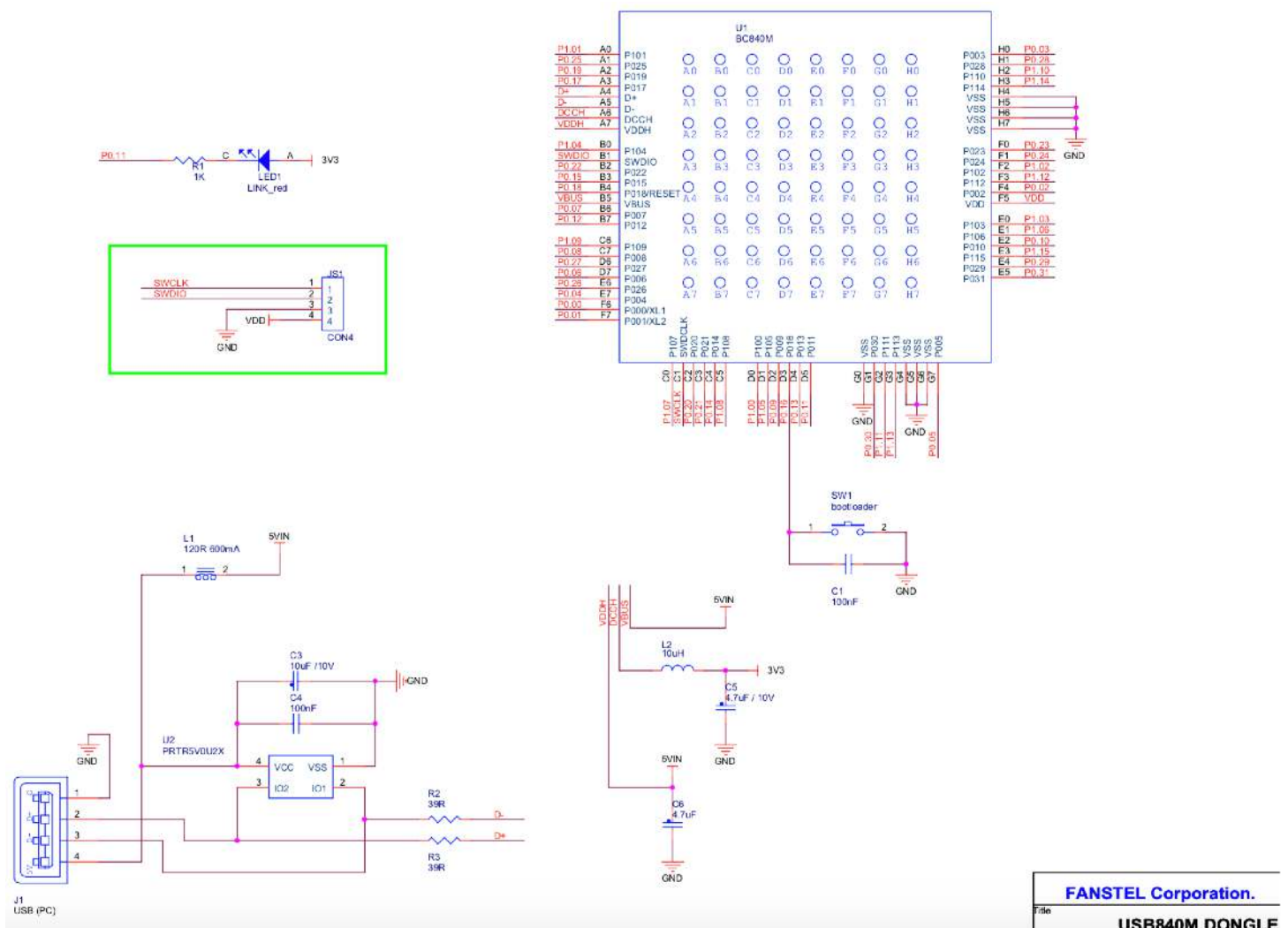
USB840M has a Bootloader button and an LED indicator. Bootloader button is at one side of enclosure. It is not necessary to open the enclosure to load application program.



USB840M is ultrasound sealed. Programming your own bootloader with private key is possible only in Fanstel factory.

We suggest using USB840F to develop your own bootloader codes.

USB840M Schematics



USB840F/M/X BLE 5, Thread, Zigbee Dongles

2. Firmware Development and Programming

Important: USB840F and USB840M don't include 32.768 KHz sleep crystal. The internal RC oscillator must be enabled.

USB840X includes 32.768 KHz sleep crystal.

Preloaded Bootloader and Private Key

USB840F dongle is pre-programmed with an USB bootloader. The project path is

```
~nRF5_SDK_15.2.0_9412b96\examples\dfu\open_bootloader
```

The Sign Key is:

```
__ALIGN(4) const uint8_t pk[64] =  
{  
    0x93, 0x49, 0x6f, 0x55, 0x97, 0x2e, 0x62, 0xcc, 0x05, 0x0c, 0x39, 0x8a, 0x46, 0x83, 0xbb, 0x06, 0xc4, 0x49,  
    0xc9, 0x5f, 0x82, 0x58, 0x42, 0x4f, 0xcd, 0x79, 0x44, 0x9d, 0x35, 0xc9, 0xc2, 0x3d,  
    0x68, 0xb9, 0x45, 0xe6, 0xdc, 0x5d, 0x38, 0xba, 0x6d, 0x9c, 0xca, 0x5e, 0x1f, 0x87, 0x0c, 0x9a, 0x6b, 0xb4,  
    0x08, 0xd1, 0x89, 0x7a, 0x32, 0x52, 0x8b, 0xf2, 0xb4, 0xa6, 0x8c, 0x28, 0xe9, 0x22  
};
```

This private key is published. We strongly recommend creating your own bootloader and private key.

Procedures to Program USB Bootloader

Go to Windows command mode and enter following commands.

```
//need nrf52 DK or JLink  
//erase all  
nrfjprog -f NRF52 --eraseall  
//programming softdevice  
nrfjprog --program s140v61.hex --family NRF52  
//programming bootloader  
nrfjprog --program 840usbBOOT.hex --family NRF52 --sectoranducirerase --reset
```

USB840F/M/X BLE 5, Thread, Zigbee Dongles

The following is screenshot for running these commands.

```
C:\Python27\Scripts>nrfjprog -f NRF52 --eraseall
Erasing user available code and UICR flash areas.
Applying system reset.

C:\Python27\Scripts>nrfjprog --program sl40v61.hex --family NRF52
Parsing hex file.
Reading flash area to program to guarantee it is erased.
Checking that the area to write is not protected.
Programming device.

C:\Python27\Scripts>nrfjprog --program 840usbBOOT.hex --family NRF52 --sectorand
uicrerase --reset
Parsing hex file.
Erasing page at address 0xF4000.
Erasing page at address 0xF5000.
Erasing page at address 0xF6000.
Erasing page at address 0xF7000.
Erasing page at address 0xF8000.
Erasing page at address 0xF9000.
Erasing page at address 0xFA000.
Erasing page at address 0xFB000.
Erasing page at address 0xFC000.
Erasing UICR flash area.
Applying system reset.
Checking that the area to write is not protected.
Programming device.
Applying system reset.
Run.
```

USB840F/M/X BLE 5, Thread, Zigbee Dongles

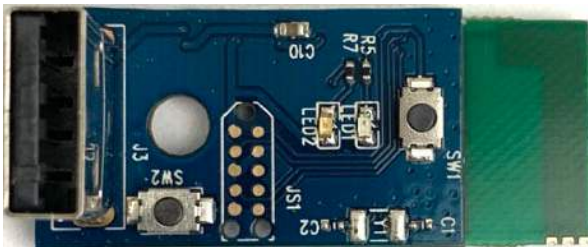
BT840F/X Hardware Setup for Programming Bootloader

We recommend creating your own bootloader with private key. To load your boot loader, you need to remove USB840F or USB840X PCBA from enclosure.

Procedures to open enclosure:



- Uses a Phillips head screw driver to unscrew.
- Uses a medium-large flat head screw driver on the notch to separate top and bottom enclosures.
- Press down USB connector to remove PCBA from enclosure.
- As shown in the following photo, SW1 is Reset and SW2 is Bootloader button.



USB840M

USB840M enclosure is ultrasound sealed. It can not be open without destroying enclosure. Programming new bootloader is possible only in factory.

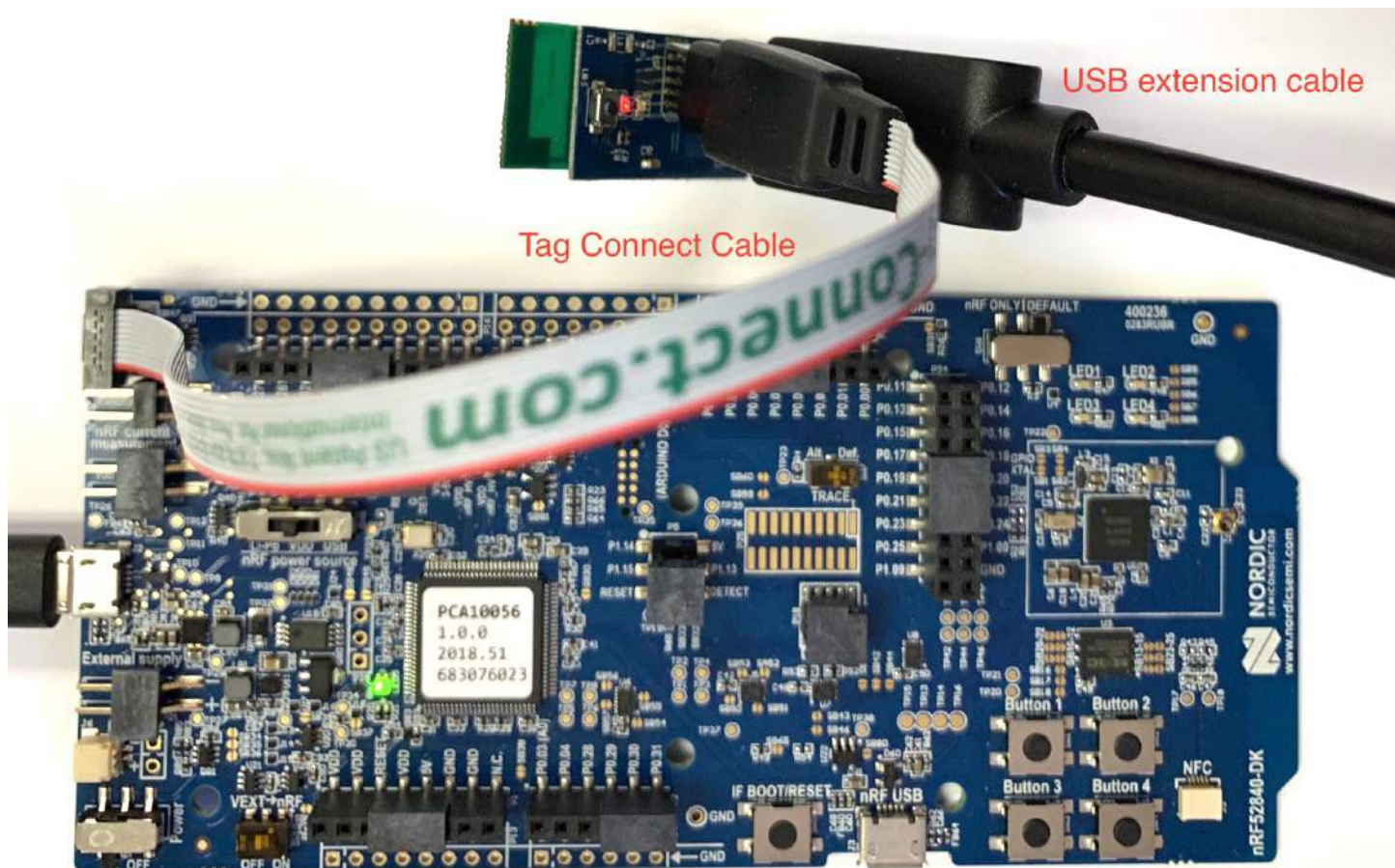
We recommend using USB840F for development. Programming service is available from Fanstel.

Or, you can continue to use Fanstel bootloader and “published” private key. It is not secure.

You can load your application program by pressing the Bootloader button at one side of enclosure during powering up.

Programming Bootloader

- Uses Nordic PCA10056 and a Tag Connect cable TC2050-IDC-NL-050-ALL to program bootloader.
- If you connect dongle to a PC USB port, Tag Connect cable can not make reliable contact with pads on dongle. You need to use an USB extension cable to connect dongle to an USB power source.



After Executing **chiperase**, VDD in dongle is set to 1.8V. To use 3.3V VDD, change setting in the UICR register. The command to write UICR->REGOUT0 to output 3.3V supply voltage is:

```
nrfjprog --memwr 0x10001304 --val fffff5
```

After programming this bootloader. Plug the dongle into the PC USB port. You should able to see new device name “nRF52 SDFU USB”



```
C:\Python27\Scripts>nrfutil dfu usb_serial -pkg BT840distance+4180828.zip -p COM45
[#####] 100%
Device programmed.
```


USB840F/M/X BLE 5, Thread, Zigbee Dongles

Steps to install Fanstel demo HEX codes

Plug the dongle into a PC USB port. You don't need the nRF52 DK when programming the application.

There are two codes in zip file **BT840_USBbootloader181018b.rar**.

```
//install distance test application  
nrfutil dfu usb_serial -pkg BT840distance+4180828.zip -p COM45  
Com port number, COMXX is set by your PC.
```

This command will install the Nordic BLE blink example.

If you want to program other applications, hold the **Bootloader** button during reset. Dongle is set into USB bootloader mode and red LED on.

Firmware to Control Power Amplifier in USB840X

The following codes are used to set up and control power amplifier SKY66112 in USB840X.

Header files

```
#ifndef PA_LNA_H_  
#define PA_LNA_H_
```

```
#include "ble.h"  
#include "app_error.h"  
#include "nrf_drv_gpiote.h"  
#include "nrf_drv_ppi.h"
```

```
void pa_lna_init(uint32_t gpio_pa_pin, uint32_t gpio_lna_pin);
```

```
#endif
```

Main program.

```
#include "pa_lna.h"
```

```
#define APP_PA_LAN
```

```
#ifdef APP_PA_LAN
```

```
#define APP_PA_PIN 17
```

```
#define APP_LNA_PIN 19
```

```
#define APP_CHL_PIN8
```

```
#define APP_CPS_PIN 6
```

```
int main(void)
{
....
#ifdef APP_PA_LAN
    nrf_gpio_cfg_output(APP_CPS_PIN);
    nrf_gpio_cfg_output(APP_CHL_PIN);
    nrf_gpio_pin_set(APP_CHL_PIN);
    nrf_gpio_pin_clear(APP_CPS_PIN); //enable
    pa_lna_init(APP_PA_PIN,APP_LNA_PIN);
#endif
....
}
```

SKY66112 control codes.

```
#include <stdint.h>
#include <string.h>

#include "pa_lna.h"
#include "ble.h"
#include "app_error.h"
#include "nrf_drv_gpiote.h"
#include "nrf_drv_ppi.h"

void pa_lna_init(uint32_t gpio_pa_pin, uint32_t gpio_lna_pin)
{
    ble_opt_t opt;
    uint32_t gpiote_ch = NULL;
    ret_code_t err_code;

    memset(&opt, 0, sizeof(ble_opt_t));

    err_code = nrf_drv_gpiote_init();
    if(err_code != NRF_ERROR_INVALID_STATE)
        APP_ERROR_CHECK(err_code);

    err_code = nrf_drv_ppi_init();
    //if(err_code != MODULE_ALREADY_INITIALIZED)
        APP_ERROR_CHECK(err_code);

    nrf_ppi_channel_t ppi_set_ch;
    nrf_ppi_channel_t ppi_clr_ch;

    err_code = nrf_drv_ppi_channel_alloc(&ppi_set_ch);
```

```
APP_ERROR_CHECK(err_code);

err_code = nrf_drv_ppi_channel_alloc(&ppi_clr_ch);
APP_ERROR_CHECK(err_code);

nrf_drv_gpiote_out_config_t config = GPIOTE_CONFIG_OUT_TASK_TOGGLE(false);

if((gpio_pa_pin == NULL) && (gpio_lna_pin == NULL))
{
    err_code = NRF_ERROR_INVALID_PARAM;
    APP_ERROR_CHECK(err_code);
}

if(gpio_pa_pin != NULL)
{
    if(gpiote_ch == NULL)
    {
        err_code = nrf_drv_gpiote_out_init(gpio_pa_pin, &config);
        APP_ERROR_CHECK(err_code);

        gpiote_ch = nrf_drv_gpiote_out_task_addr_get(gpio_pa_pin);
    }

    // PA config
    opt.common_opt.pa_lna.pa_cfg.active_high = 1; // Set the pin to be active high
    opt.common_opt.pa_lna.pa_cfg.enable     = 1; // Enable toggling
    opt.common_opt.pa_lna.pa_cfg.gpio_pin   = gpio_pa_pin; // The GPIO pin to toggle tx
}

if(gpio_lna_pin != NULL)
{
    if(gpiote_ch == NULL)
    {
        err_code = nrf_drv_gpiote_out_init(gpio_lna_pin, &config);
        APP_ERROR_CHECK(err_code);

        gpiote_ch = nrf_drv_gpiote_out_task_addr_get(gpio_lna_pin);
    }

    // LNA config
    opt.common_opt.pa_lna.lna_cfg.active_high = 1; // Set the pin to be active high
    opt.common_opt.pa_lna.lna_cfg.enable     = 1; // Enable toggling
    opt.common_opt.pa_lna.lna_cfg.gpio_pin   = gpio_lna_pin; // The GPIO pin to toggle rx
}

// Common PA/LNA config
opt.common_opt.pa_lna.gpiote_ch_id = (gpiote_ch - NRF_GPIOTE_BASE) >> 2; // GPIOTE channel used for
radio pin toggling
opt.common_opt.pa_lna.ppi_ch_id_clr = ppi_clr_ch; // PPI channel used for radio pin clearing
opt.common_opt.pa_lna.ppi_ch_id_set = ppi_set_ch; // PPI channel used for radio pin setting

err_code = sd_ble_opt_set(BLE_COMMON_OPT_PA_LNA, &opt);
```

USB840F/M/X BLE 5, Thread, Zigbee Dongles

```
    APP_ERROR_CHECK(err_code);  
}
```

Nordic Development Environment

Nordic Semiconductor provides a complete range of hardware and software development tools for the nRF52 Series devices. nRF52 DK board is recommended for firmware development. Document and Software development tools can be downloaded by the following links.

Get start with Nordic chip and all online documents.

http://infocenter.nordicsemi.com/index.jsp?topic=/com.nordic.infocenter.nrf52/dita/nrf52/development/nrf52_dev_kit.html&cp=1_1

Nordic SDK with many example projects.

https://developer.nordicsemi.com/nRF5_SDK/

Nordic development zone. You can search or ask a question there.

<https://devzone.nordicsemi.com/tutorials/b/getting-started/posts/development-with-gcc-and-eclipse>

Programming the Nordic chip

Download and install Nrf5x-Command-Line Tools

<https://www.nordicsemi.com/eng/nordic/Products/nRF52840/nRF5x-Command-Line-Tools-Win32/58850>

Download and install nRF Connect

https://www.nordicsemi.com/?sc_itemid={B935528E-8BFA-42D9-8BB5-83E2A5E1FF5C}

USB840F/M/X BLE 5, Thread, Zigbee Dongles

Revision History

- May 2019, Ver. 1.00: Initial release

USB840F/M/X BLE 5, Thread, Zigbee Dongles

Contact Us

United States:

Fanstel Corp.
7466 E. Monte Cristo Ave. Scottsdale AZ 85260
Tel. 1 480-948-4928
Fax. 1-480-948-5459
Email: info@fanstel.com
Website: www.fanstel.com

Taiwan:

Fanstel Corp.
10F-10, 79 Xintai Wu Road
Xizhu, New Taipei City, Taiwan 22101
泛世公司
臺灣省新北市汐止區新臺五路79號10樓之10, 22101
Tel. 886-2-2698-9328
Fax. 886-2-2698-4813
Email: info@fanstel.com
Website: www.fanstel.com

China:

Fanstel Technologies Corp.
11 Jiale Street
Ping-Dih, Long-Gang, Shen Zhen, GD 518117
泛世康科技(深圳)有限公司
廣東省深圳市龍崗區坪地鎮佳樂街11號
Tel. 86-755-8409-0928
Fax. 86-755-8409-0973
QQ. 3076221086
Email: info@fanstel.com
Website: www.fanstel.com