Reference guide

N32WB452 Series Bluetooth Components Reference Guide

Introduction

The purpose of this document is to allow users to quickly familiarize themselves with the use of N32WB452 series Bluetooth components, so as to reduce the preparation time in the early stage of development and reduce the difficulty of development.

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

CONTENTS

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

1. Introduction

1.1. Overview

Welcome to National Technology N32WB452 series Bluetooth component reference manual, the document mainly introduces the interface description and usage guide of N32WB452 series MCU Bluetooth communication.

The N32WB452 series chips support the Bluetooth BLE5.0 specification and support Profile configuration. Customers can modify the Profile item through the interface parameters according to their own needs.

Developers can cooperate with on-chip related resources and SDK kits to develop their own Bluetooth products.

This document will describe in detail the resource requirements of the Bluetooth communication API, function descriptions and reference examples of each API function.

1.2. Applicability

- In the N32WB452 series, Bluetooth BLE5.0 runs on an independent high-performance 32-bit processor on-chip; the on-chip embeded ARM Cotex-M4 core controls and transmits data with the Bluetooth BLE5.0 core through the API interface provided; The Bluetooth components in the documentation are only for N32WB452 series chips.
- The Bluetooth component provided in this document shields the underlying hardware and complex protocol stack processing, and abstracts a simple and easy-to-use user interface. Provide high integration, convenience, Bluetooth communication function requirements for MCU platform developers.
- The SDK only supports the KEIL5 platform for the time being, and other compilation platforms are being improved.

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

2. SDK application architecture

API supports system or none-system Bluetooth products.

When a Bluetooth-related event occurs, the application layer is notified through the user layer callback function.

Figure 1 Application layering of bluetooth components

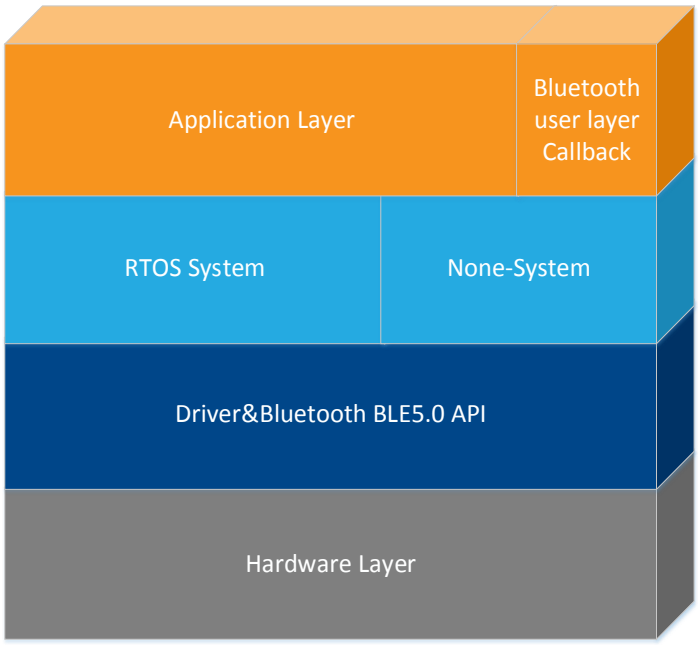
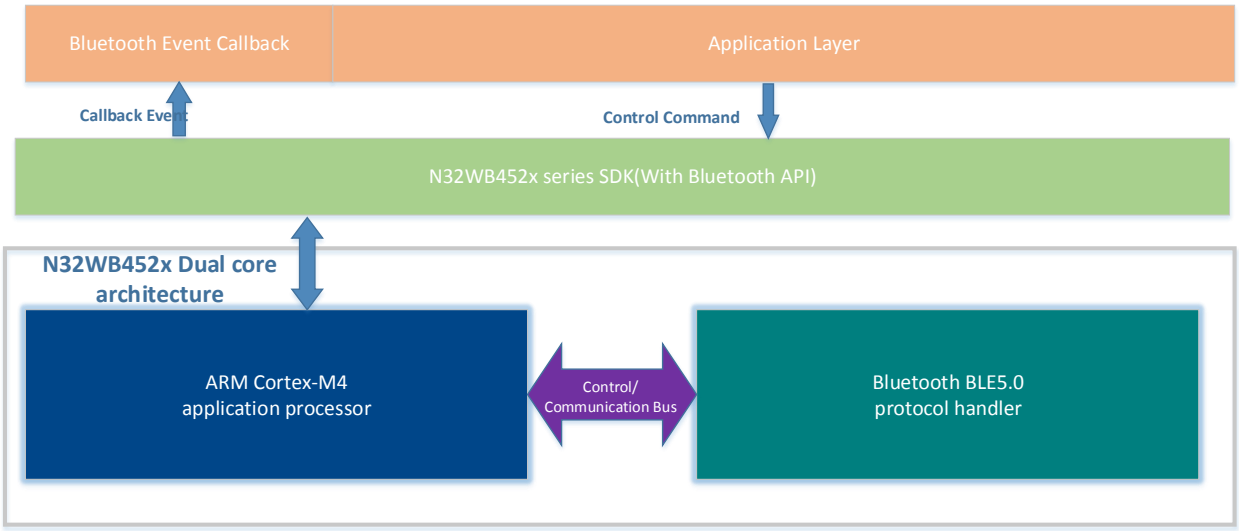


Figure 2 Bluetooth control and callback event box



Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

3. Bluetooth component resource requirements

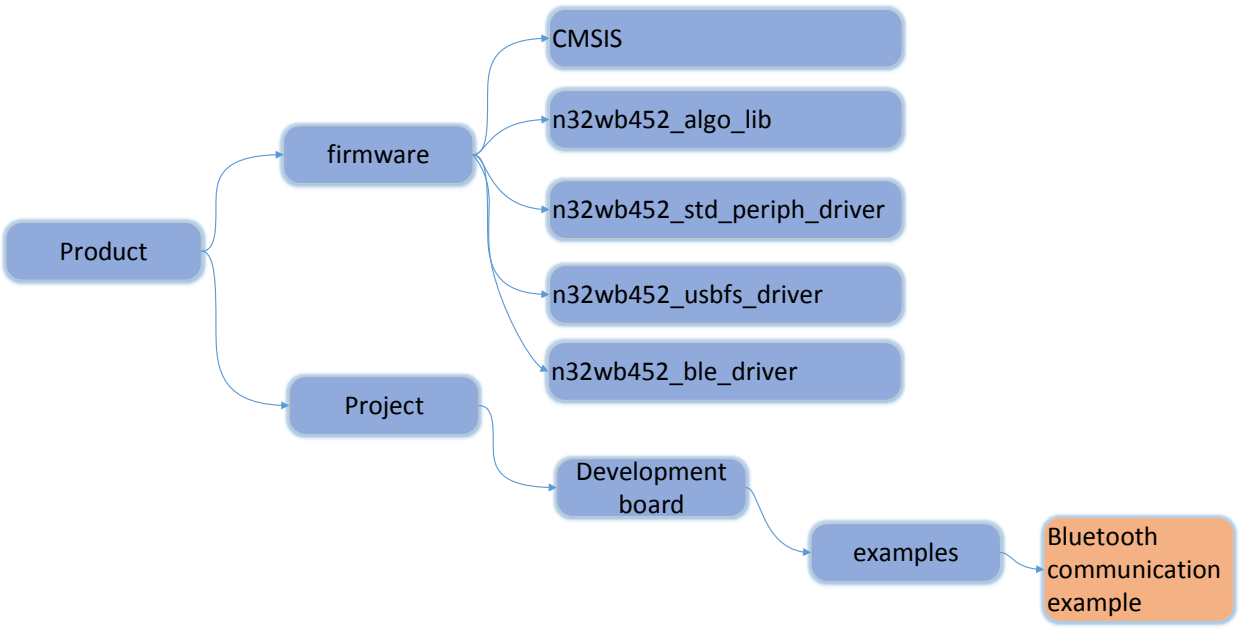
The steps of using the SDK are roughly divided into: initialization, running the main bluetooth thread, and callback (when the bluetooth event is triggered).

3.1. Resources occupied by bluetooth components

No.	Resource	Description	Notes
1	FLASH 70KB	About 70KBytes FLASH area	-
2	SRAM	About 6KB	-
3	Stack	About 5KB	-
4	Heap	About 2KB	-
5	EXTI6	BLE status monitor interrupt	
6	EXTI14	BLE common interrupt	
7	TIM3	BLE kernel timeout event	

3.2. Project architecture

Figure 3 Code project directory structure



Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

4. Bluetooth component parameters and interface description

4.1. Bluetooth initialization function

Initialize function: bt_attr_parambt_init; // Used to define Bluetooth configuration information, such as broadcast name, address, UUID, signature, etc. Details are as follows：

Variable	Description
ctrl_param	control flag
device_name[32]	The name of the bluetooth broadcast. *If it is 0, use the default bluetooth name.
scan_rsp_data[31]	The response data of the slave when the master scans. * Valid length does not exceed 30 characters.
scan_rsp_data_len	response data length
device_addr[20]	Bluetooth device address *format as："FE:11:22:FA:33:44".If it is 0, it will be automatically configured as the default address.
service[1]	Bluetooth service definition, including features, permissions, etc.

4.2. Callback functions

The callback function is implemented by the user layer and is mainly used to notify the upper layer for processing when a Bluetooth event occurs. Such as connection, disconnection, data reception, etc.

Note: The execution time of the callback function should be as short as possible. If the time is too long, the sending and receiving efficiency may be affected!

Callback function type definition：

```
typedef void (*bt_event_callback_handler_t)( bt_event_enum event,
                                             const uint8_t * data,
                                             uint32_t size,
                                             uint32_t character_uuid);
```

Parameter Description：

[IN] bt_event_enum	event	Bluetooth component notification events.
[IN] const uint8_t	*data	Data address (when there is a data event notification).
[IN]uint32_t	size	Data size (unit number of bytes)
[IN]uint32_t	character_uuid	Feature sub id, don't pay attention.

Return Value：

none

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

4.3. Initialization bt_ware_init()

Definition：int32_t bt_ware_init(bt_attr_param *pinit, bt_event_callback_handler_tpcallback);

Bluetooth component initialization.

Parameter Description：

[IN] bt_attr_param *	pinit	Bluetooth initialization parameters
[IN] bt_event_callback_handler_tpcallback		User callback interface.

Return Value：

BT_RET_SUCCESS:	Initialize success
Other:	Initialize failure

4.4. Interrupt handler bt_handler()

Definition：void bt_handler(void)

Bluetooth component interrupt handler.

Parameter Description：

void: none.

Return Value：

void: none.

4.5. Main thread bt_run_thread()

Definition：void bt_run_thread(void);

The main thread of the Bluetooth component running, used for Bluetooth reception, transmission, broadcast, notification and other event processing. It is recommended that the running time interval of the main thread should be less than 1ms. If a large amount of data needs to be sent or received, centralized time scheduling can be used.

Parameter Description：

void: none.

Return Value：

void: none.

4.6. Receive data bt_rcv_data()

Definition：uint32_t bt_rcv_data(uint8_t *data, uint32_t size, uint32_t character);

According to the BT_EVENT_RCV_DATA event, read the data of the corresponding service feature word.

Parameter Description：

[OUT] uint8_t *	data	Data address

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

[IN] uint32_t size Data size

[IN] uint32_t character Characteristic words of occurrence data, do not pay attention

Return Value：

Returns the actual read size (unit: number of bytes).

4.7. Send data bt_snd_data()

uint32_t bt_snd_data(const uint8_t *data, uint32_t size, uint32_t character);

Send data.

Parameter Description：

[IN] uint8_t * data Send data address

[IN] uint32_t size Send data size

[IN] uint32_t character Characteristic words of sending data, do not pay attention

Return Value：

Return 0.

4.8. Disconnect bt_disconnect()

Void bt_disconnect(void)

Slave actively disconnect the bluetooth connection.

Parameter Description：

void: none.

Return Value：

void none.

4.9. Status monitoring ble_status_monitor ()

void ble_status_monitor (void);

Real-time monitoring of bluetooth working status, automatic reset operation of bluetooth when necessary.

Parameter Description：

void: none.

Return Value：

void none.

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

4.10. Waiting for BLE status feedback ble_monitor_wait ()

uint8_t ble_monitor_wait (uint32_t timeout);

After calling the status monitoring function,you need to call this function to wait for the real-time status of BLE feedback.

Parameter Description：

timeout: wait timeout(32bits unsigned integer),simple timeing with a while loop..

Return Value(8bits unsigned integer)：

0: Received status feedback, BLE is working.

1: No status feedback received, wait time out.

4.11. Status monitoring callback function ble_monitor_callback ()

void ble_monitor_callback (void);

Bluetooth status monitoring callback function.

Parameter Description：

void: none.

Return Value：

void none.

4.12. Bluetooth protocol stack initialization BT_init()

void BT_init(void)

Initialize the Bluetooth protocol stack.

Parameter Description：

void: none.

Return Value：

void: none.

4.13. Bluetooth kernel busy state detection is_bt_busy()

bool is_bt_busy(void)

Detect bluetooth core busy state. Low power states are not allowed when the core is busy.

Parameter Description：

void: none.

Return Value：

1: busy

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

0: idle

4.14. Get the current Bluetooth API version BT_get_version()

void BT_get_version(uint8_t * version)

Get current Bluetooth API version information.

Parameter Description：

version: Version information string cache, maximum 10byte.

Return Value：

void: none.

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

5. User guide

5.1. Operation flow

- **System clock：**

Please ensure that the system clock SYSCLK_FREQ is an integer multiple of 8, the recommended values are 144MHz, 96MHz, 48MHz.

- **Define the callback function：**

The callback type: bt_event_callback_handler_t

- **Configuration parameters and initialization：**

Fill in the content of the bt_ware_init parameter and call the initialization interface.

Note: Judging the return value of the interface, such as BT_RET_SUCCESS, continue after success, otherwise check the error type.

- **Define relevant interrupts：**

Used for Bluetooth component interrupt handling bt_handler().

- **Run main thread：**

Run bt_run_thread() thread.

Note: According to the actual environment of the user, such as in a none-system/system environment, ensure that the running cycle of the main thread of the Bluetooth component is 1ms. If other threads schedule relationships or frequently erase and write FLASH, the main bluetooth thread does not have permission to execute, which will result in abnormal data reception or error at that moment.

- **Status monitoring：**

The Bluetooth status monitoring function is used to monitor the status of the Bluetooth controller and the kernel. It is used for status monitoring and control, and automatic power consumption management. It requires the user to periodically call in the program. The recommended calling period is 1 second.

Note: Once status monitoring is enabled, the status monitoring callback function ble_monitor_callback() must be called in the external interrupt function EXTI9_5_IRQHandler().

- **Low power consumption mechanism：**

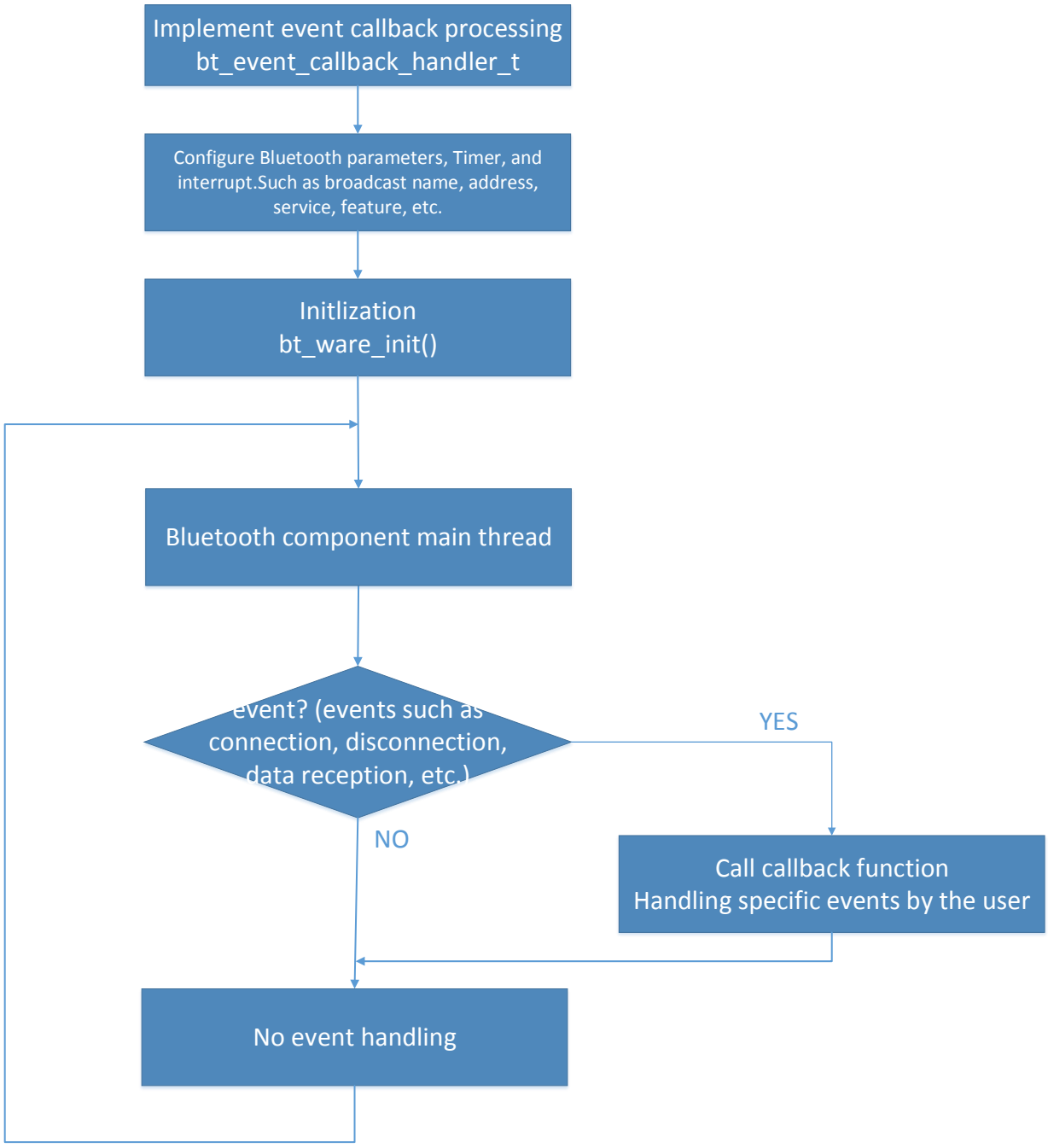
The MCU will try to enter the STOP2 mode 10 seconds after it is powered on. At this time, if the Bluetooth is not connected, the system will enter the STOP2 mode to save power consumption. When there is a Bluetooth event or other external interrupt of the MCU, the system will be woken up, and the initialization process will be re-executed after waking up.

After being woken up for 5 seconds, the system will try to enter low power mode again.

If the system needs to process other application tasks for a long time and cannot enter the low-power mode when the Bluetooth is not connected, please perform the required operations before entering the low-power mode.

Figure 4 Bluetooth main component processing flow

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China



5.2. Application examples

```
uint8_t snd_buf[] = {"0-1234567890."};
bt_attr_parambt_init = {0};


/**
  * @brief    Bluetooth interrupt handling.
  * @param count specifies the delay time length.
  */
void EXTI15_10_IRQHandler(void)
```


Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

```
{
    if (EXTI_GetITStatus(EXTI_LINE14) != RESET)
    {
        bt_handler();
        EXTI_ClrITPendBit(EXTI_LINE14);
    }
}

Void bt_event_callback_func(bt_event_enum event, uint8_t * data, uint32_t size, uint32_t character_uuid)
{
    uint16_t length;

    switch (event)
    {
    case BT_EVENT_VERSION:
        printf("cb version:\r\n");
        break;

    case BT_EVENT_CONNECTED:// connection event
        printf("cb connect:\r\n");
        break;

    case BT_EVENT_DISCONNECTD: // disconnection event
        printf("cb disconnect:\r\n");
        break;

    case BT_EVENT_RCV_DATA:// data receive event
        length = bt_rcv_data(bt_data_buf, sizeof(bt_data_buf), character_uuid);
        if (length)
        {
            printf("rcv data [%d] ok,\r\n", length);
            //bt_snd_data(bt_data_buf, length, USER_IDX_WRITE_NOTIFY_CHAR);
        }
        else
        {
            printf("rcv data err,\r\n");
        }
        break;

    default:
        break;
    }
}
```


Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

```
int main(void)
{
    const char *name = "WB452_BLE";
    const char *addr = "10:20:30:a0:b0:c2";
    //const char *response = "bt salve say hello";
    int32_t ret;

    /* System Clocks Configuration */
    RCC_Configuration();

    /* USART ForPrintf */
    USART_Config(256000);
    printf("system start...\r\n");

    memcpy(bt_init.device_name, name, MIN(strlen(name), sizeof(bt_init.device_name)));
    memcpy(bt_init.device_addr, addr, MIN(strlen(addr), sizeof(bt_init.device_addr)));

    bt_init.service[0].svc_uuid                 = SERVICE_UUID;
    bt_init.service[0].character[0].uuid        = USER_IDX_WRITE_NOTIFY_CHAR;
    bt_init.service[0].character[0].permission  = BT_WRITE_PERM | BT_NTF_PERM;

    ret = bt_ware_init(&bt_init, (bt_event_callback_handler_t)bt_event_callback_func);
    if (ret != BT_RET_SUCCESS)
    {
        printf("btinit failed.\r\n");
    }
    else
    {
        printf("btinit ok.\r\n");
    }

   smartlock_touch_tim6_init();//Timing period 5ms

    while(1)
    {
        bt_run_thread();
    }
}
```


Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

5.3. BLE Low power configuration

5.3.1. Bluetooth broadcast parameter setting

During the BLE development process, the Bluetooth broadcast time interval can be configured according to the application scenario and power consumption requirements；

app_env.adv_para.adv_int_min = 0x320; // Minimum broadcast interval time: 0.5S = 0x320*0.625 ms

app_env.adv_para.adv_int_max = 0x640; // Maximum broadcast interval time: 0.5S = 0x320*0.625 ms

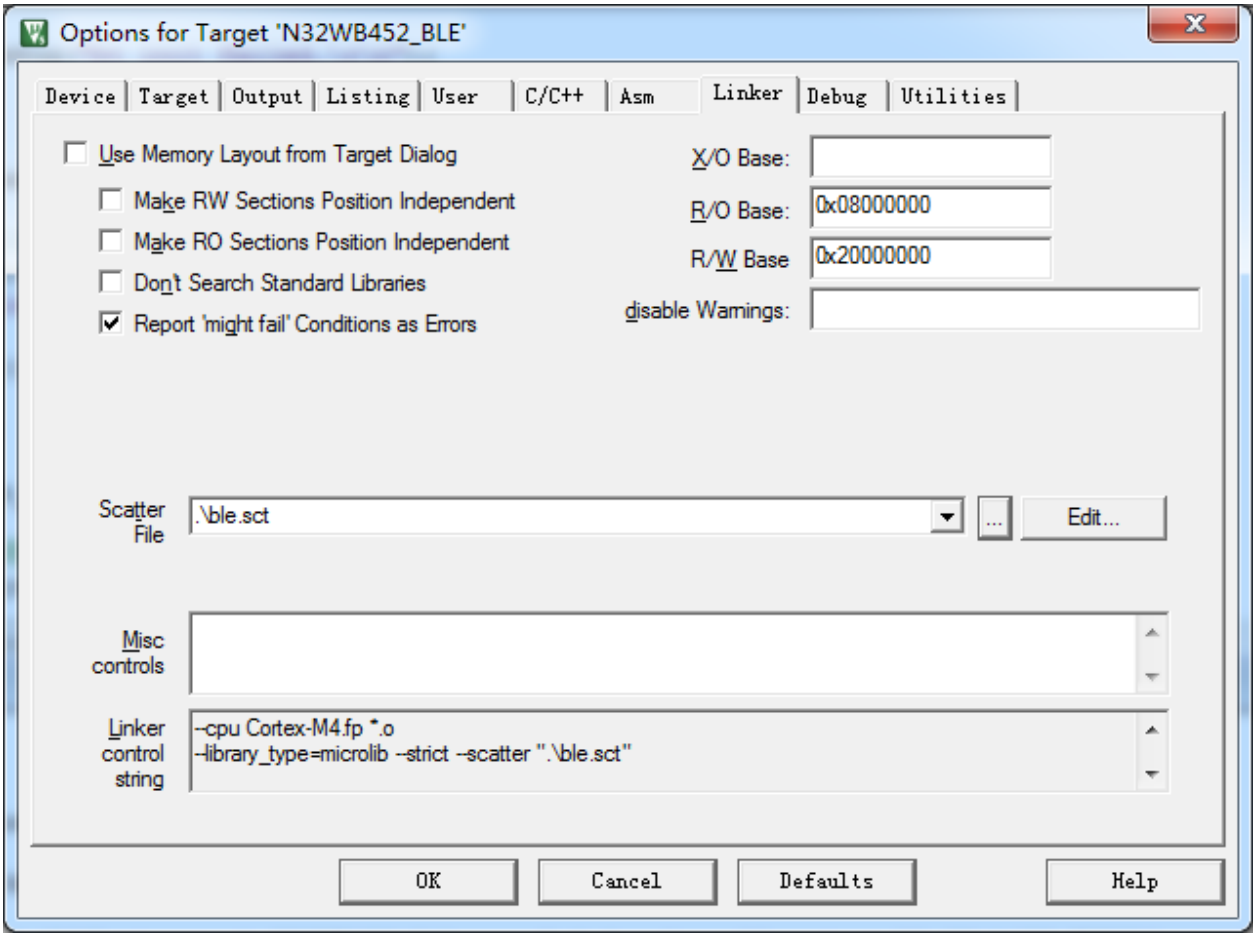
5.3.2. Bluetooth Low Energy Profile

When the MCU enters the low-power STOP2 mode, it needs to put the Bluetooth-related resources in the SRAM of the 0x20000~0x24000 address segment, so that when it wakes up from Bluetooth, it can quickly obtain the Bluetooth protocol stack resources, as shown in the figure below:

```
RW_IRAM2 0x20020000 0x00004000  {; powerdown variables
 n32wb452_ble_api.o (+RW +ZI)
 user_task.o ;(+RW +ZI)
 main.o (+RW +ZI)
 n32wb452_data_fifo.o (+RW +ZI)
 app_sec.o (+RW +ZI)
 app_task.o (+RW +ZI)
 interface.o (+RW +ZI)
 gapc.o (+RW +ZI)
 gattc.o (+RW +ZI)
 gapc_task.o (+RW +ZI)
 gattc_task.o (+RW +ZI)
 gattm_task.o (+RW +ZI)
 l2cc_task.o (+RW +ZI)
 gapm_task.o (+RW +ZI)
 l2cc.o (+RW +ZI)
 app.o (+RW +ZI)
 app_user.o (+RW +ZI)
 prf.o (+RW +ZI)
 ke_task.o (+RW +ZI)
 eif_spi.o (+RW +ZI)
 ke.o (+RW +ZI)
 ke_event.o (+RW +ZI)
 gattm.o (+RW +ZI)
 rwip.o (+RW +ZI)
 gapm.o (+RW +ZI)
 h4tl.o (+RW +ZI)
 hci.o (+RW +ZI)
 hci_tl.o (+RW +ZI)
 l2cm.o (+RW +ZI)
 stdout.o (+RW +ZI)
 rand.o (+RW +ZI)
 startup_n32wb452.o (+RW +ZI)
}
```

In the project configuration, add the load file：

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China



5.3.3. Low power wakeup

When the Bluetooth wakes up from low power mode, it first enters the IRQ interrupt service function EXTI15_10_IRQHandler(). After the Bluetooth wakes up, you need to configure the Bluetooth driver related resources immediately, so that the Cortex-M4 core can communicate with the Bluetooth coprocessor normally.

```
/**
 * @brief  Inserts a delay time.
 * @param count specifies the delay time length.
 */
void EXTI15_10_IRQHandler(void)
{
    if (EXTI_GetITStatus(EXTI_LINE14) != RESET)
    {
        //When the MCU is in STOP2 mode, it is woken up by Bluetooth, and the peripherals used need to be reconfigured
        if (flag_bt_enter_sleep == 1)
        {
            flag_bt_enter_sleep = 0;
            mainboard_exit_stop2();
        }
        //log_debug("exti 14 irq %d.\r\n", flag_bt_enter_sleep);
        bt_handler();
        /* Clears the SEL Button EXTI line pending bits. */
        EXTI_ClrITPendBit(EXTI_LINE14);
    }
}
```


Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

```c
//After exiting from STOP2 mode, all MCU resources used need to be reconfigured
void mainboard_exit_stop2(void)
{
    SystemInit();
    SetSysClock_HSE_PLL(RCC_PLL_MUL_18);
    RCC_Configuration();
    log_init();
    log_debug("system restart...\r\n");
    led_gpio_init();
    LED3_ON();
    ble_hardware_reinit();
    TIM6_init();
    flag_bt_enter_sleep = 0;
    system_10s_flag = 0;
}
```


Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

6. Version History

VERSION	DATE	NOTE
V1.0	2020-05-29	Create documentation
V1.1.5	2021-3-23	Delete the heartbeat guard mechanism and related APIs, and add the Bluetooth status monitoring mechanism, status monitoring, and callback function descriptions
V1.2	2022-3-29	Remove unused timing functions,added the description of waiting for BLE status feedback function.
V1.3	2023-6-20	Add some resources used by BLE:EXTI6/14 and TIM3

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China

7. NOTICE

This document is the exclusive property of Nations Technologies Inc. (Hereinafter referred to as NATIONS). This document, and the product of NATIONS described herein (Hereinafter referred to as the Product) are owned by NATIONS under the laws and treaties of the People's Republic of China and other applicable jurisdictions worldwide.

NATIONS does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only.

NATIONS reserves the right to make changes, corrections, enhancements, modifications, and improvements to this document at any time without notice. Please contact NATIONS and obtain the latest version of this document before placing orders.

Although NATIONS has attempted to provide accurate and reliable information, NATIONS assumes no responsibility for the accuracy and reliability of this document.

It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NATIONS be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product.

NATIONS Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at user's risk. User shall indemnify NATIONS and hold NATIONS harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage.

Any express or implied warranty with regard to this document or the Product, including, but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law.

Unless otherwise explicitly permitted by NATIONS, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.

Nations Technologies Inc.
Tel：+86-755-86309900
Email：info@nationstech.com
Address: Nations Tower, #109 Baoshen Road, Hi-tech Park North.
Nanshan District, Shenzhen, 518057, P.R.China