Reference guide

N32WB452 Series Bluetooth OTA Upgrade Reference GuideV1.0

Introduction

This guide mainly provides Bluetooth OTA upgrade reference for N32WB452 series MCU, and provides technical support for developers.

Contents

1. Introduction

1.1 Overview

Welcome to the National Technology N32WB452 series Bluetooth OTA upgrade reference manual, the document mainly introduces the N32WB452 series MCU Bluetooth OTA upgrade process description and usage guide.

Developers can cooperate with on-chip related resources and SDK kit to develop their own Bluetooth OTA upgrade process.

1.2 Applicability

- This document is based on the N32WB452 series chips to develop the Bluetooth OTA upgrade function, so the Bluetooth components in this document are only applicable to the N32WB452x series chips.
- This project only supports the KEIL5 platform for the time being, and other compilation platforms are being improved.

2. OTA overview of upgraded features

2.1 OTA flow description:

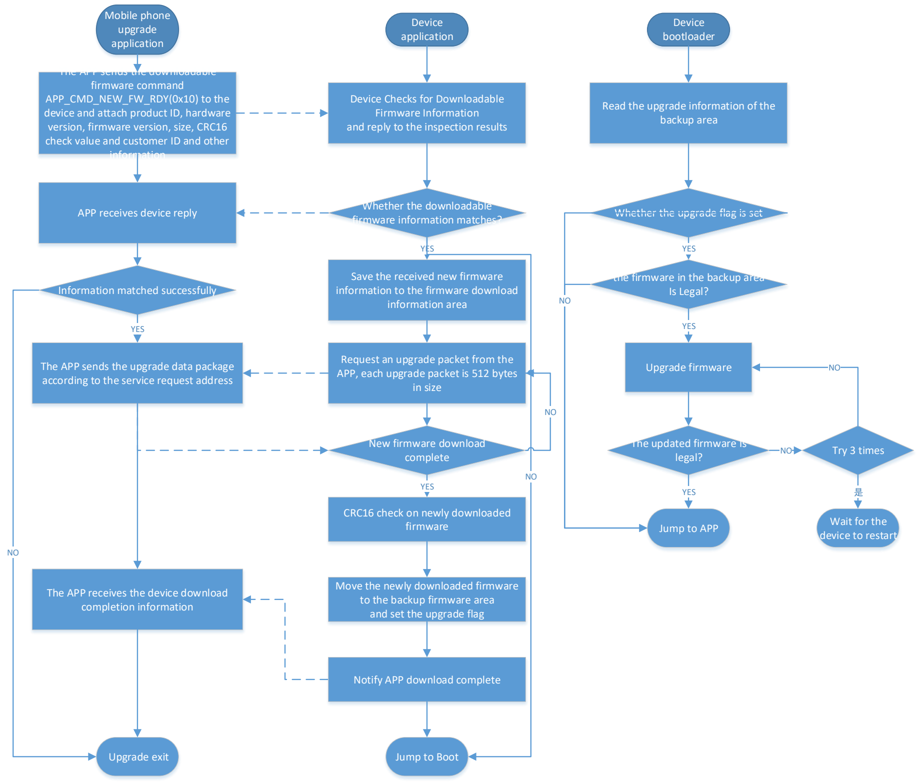
OTA flow description：

1. When the firmware needs to be upgraded, the mobile APP sends the firmware upgrade information to the N32WB452, including firmware version number, hardware version number, product ID, customer code, firmware size and CRC16 check code；

2. After N32WB452 receives the upgrade firmware information, it will judge whether it conforms to the current version information, if not, it will reply to the APP according to the actual situation；

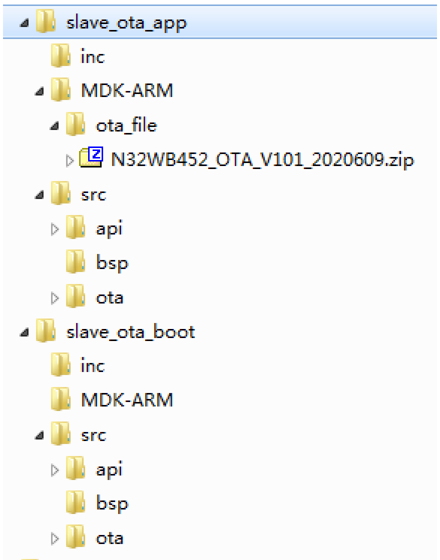
3. During the process of downloading and upgrading the firmware, the N32WB452 actively obtains the upgrade package according to the upgrade progress, and the APP delivers the upgrade package data according to the upgrade package download address reported by the N32WB452. Each upgrade package comes with CRC16 checksum.

The flowchart is as follows：


Mobile phone upgrade application
Device application
Device bootloader
The APP sends the downloadable firmware command APP_CMD_NEW_FW_RDY(0x10) to the device and attach product ID, hardware version, firmware version, size, CRC16 check value and customer ID and other information
Device Checks for Downloadable Firmware Information and reply to the inspection results
Read the upgrade information of the backup area
APP receives device reply
Whether the downloadable firmware information matches?
Whether the upgrade flag is set
YES
YES
Information matched successfully
Save the received new firmware information to the firmware download information area
the firmware in the backup area Is Legal?
NO
YES
YES
The APP sends the upgrade data package according to the service request address
Request an upgrade packet from the APP, each upgrade packet is 512 bytes in size
Upgrade firmware
NO
NO
New firmware download complete
The updated firmware is legal?
NO
Try 3 times
NO
YES
YES
是
CRC16 check on newly downloaded firmware
Jump to APP
Wait for the device to restart
NO
The APP receives the device download completion information
Move the newly downloaded firmware to the backup firmware area and set the upgrade flag
Notify APP download complete
Upgrade exit
Jump to Boot

2.2 OTA project description

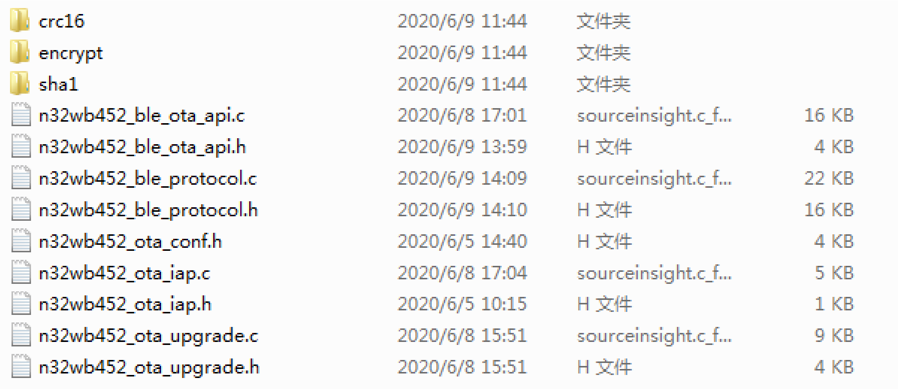
The OTA project includes the application program “Slave_ota_app” and the boot program “Slave_ota_boot”, and the OTA-related programs are placed in the OTA folder under the project. The N32WB452_OTA_V101_20200609.zip in the “Ota_file” is the upgrade package in the Demo project. As shown below：



```
slave_ota_app
    inc
    MDK-ARM
        ota_file
            N32WB452_OTA_V101_2020609.zip
    src
        api
        bsp
        ota
slave_ota_boot
    inc
    MDK-ARM
    src
        api
        bsp
        ota
```

2.3 OTA description of related documents

OTA the relevant files are shown in the figure below：



crc16	2020/6/9 11:44	文件夹	
encrypt	2020/6/9 11:44	文件夹	
sha1	2020/6/9 11:44	文件夹	
n32wb452_ble_ota_api.c	2020/6/8 17:01	sourceinsight.c_f...	16 KB
n32wb452_ble_ota_api.h	2020/6/9 13:59	H 文件	4 KB
n32wb452_ble_protocol.c	2020/6/9 14:09	sourceinsight.c_f...	22 KB
n32wb452_ble_protocol.h	2020/6/9 14:10	H 文件	16 KB
n32wb452_ota_conf.h	2020/6/5 14:40	H 文件	4 KB
n32wb452_ota_iap.c	2020/6/8 17:04	sourceinsight.c_f...	5 KB
n32wb452_ota_iap.h	2020/6/5 10:15	H 文件	1 KB
n32wb452_ota_upgrade.c	2020/6/8 15:51	sourceinsight.c_f...	9 KB
n32wb452_ota_upgrade.h	2020/6/8 15:51	H 文件	4 KB

Among：

crc16、encrypt、sha1 folders are the Bluetooth communication encryption algorithm file；

n32wb452_ble_ota_api.c and n32wb452_ble_ota_api.h are the API interface file for Bluetooth OTA upgrade；

n32wb452_ble_protocol.c and n32wb452_ble_protocol.h are the Bluetooth communication protocol processing file；

n32wb452_ota_conf.h is a configuration file that divides the flash address of each area；

n32wb452_ota_iap.c and n32wb452_ota_iap.h are jump implementation file between bootloader and APP；

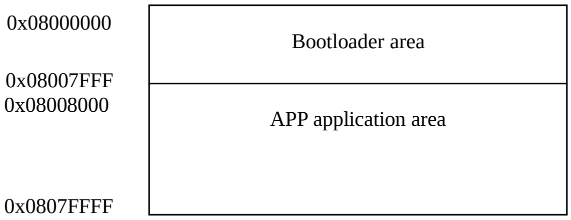
n32wb452_ota_upgrade.c and n32wb452_ota_upgrade.h are for Flash operation in OTA.

3. User guide

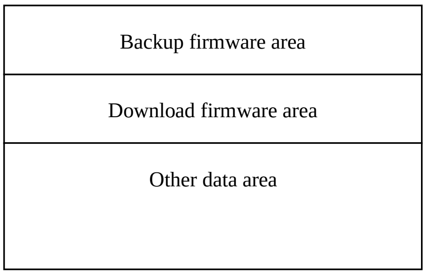
3.1 Establish FLASH storage partition

The Flash partition is the basis of the entire OTA upgrade function, so at the beginning of the project, it is necessary to plan the Flash partition and determine the size of the bootloader and application。

3.1.1 Internal Flash division of N32WB452 series



3.1.2 External SPI Flash Division



3.1.3 Description of each storage area

1. Bootloader area：

The main function of the system boot program area is to upgrade the APP application program area. When the upgrade flag in the backup firmware area is set, the bootloader moves the upgrade program from the backup firmware area to the APP application program area；

2. APP application area：

System application area；

3. Backup firmware area：

The backup area of the application program, generally, the backup firmware area is consistent with the firmware of the application program area; when the application program area needs to be upgraded, copy it from the backup area；

4. Download firmware area:

The upgrade firmware package received by the application is stored in the download firmware area in advance. After the download is completed, the downloaded firmware is moved to the backup firmware area；

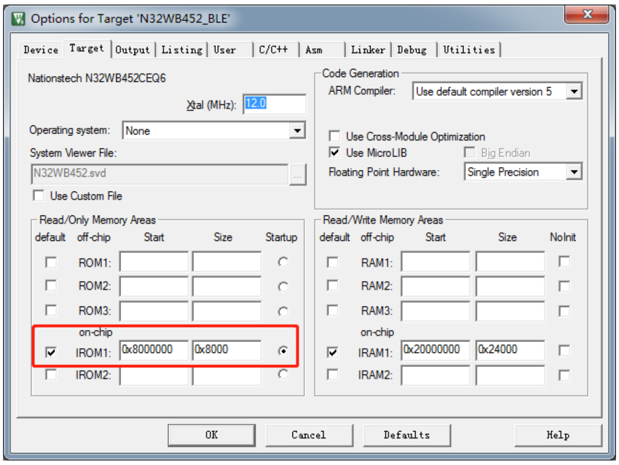
5. Other data areas:

This can be used according to the actual situation.

3.2 Project configuration of Flash

After the partition is determined, you need to modify the relevant configuration of the project：

Bootloader configuration



Nationstech N32WB452CEQ6

Xtal (MHz): 12.0

Operating system: None

System Viewer File: N32WB452.svd

□ Use Custom File

Code Generation

ARM Compiler: Use default compiler version 5

□ Use Cross-Module Optimization

☑ Use MicroLIB □ Big Endian

Floating Point Hardware: Single Precision

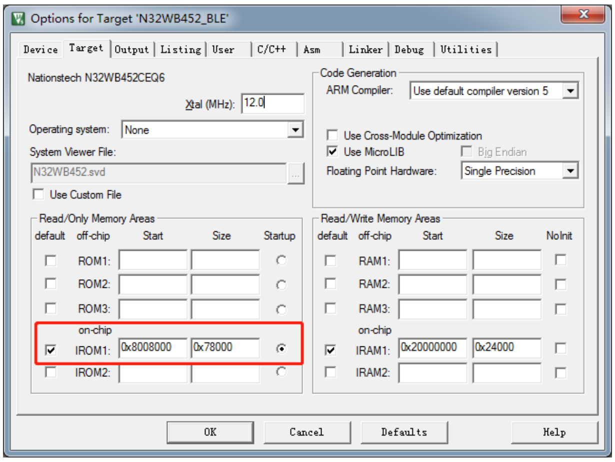
Read/Only Memory Areas

default	off-chip	Start	Size	Startup
□	ROM1:			○
□	ROM2:			○
□	ROM3:			○
	on-chip			
☑	IROM1:	0x8000000	0x8000	●
□	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
□	RAM1:			□
□	RAM2:			□
□	RAM3:			□
	on-chip			
☑	IRAM1:	0x20000000	0x24000	□
□	IRAM2:			□

APP Application configuration:



Nationstech N32WB452CEQ6

Xtal (MHz): 12.0

Operating system: None

System Viewer File: N32WB452.svd

□ Use Custom File

Code Generation

ARM Compiler: Use default compiler version 5

□ Use Cross-Module Optimization

☑ Use MicroLIB □ Big Endian

Floating Point Hardware: Single Precision

Read/Only Memory Areas

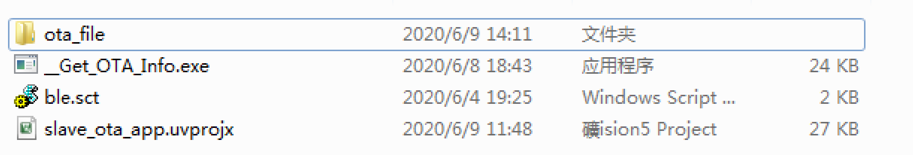
default	off-chip	Start	Size	Startup
□	ROM1:			○
□	ROM2:			○
□	ROM3:			○
	on-chip			
☑	IROM1:	0x8008000	0x78000	●
□	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
□	RAM1:			□
□	RAM2:			□
□	RAM3:			□
	on-chip			
☑	IRAM1:	0x20000000	0x24000	□
□	IRAM2:			□

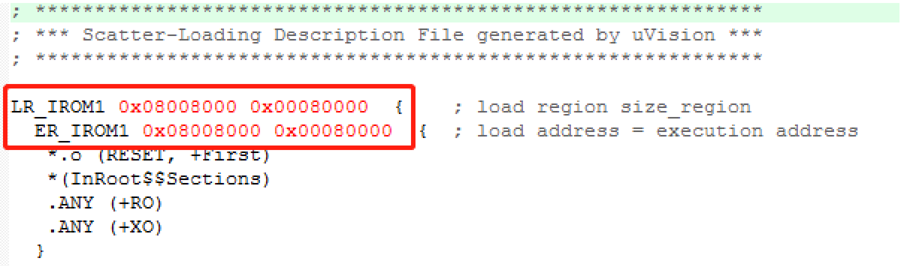
Since this project uses the loading file, it is also necessary to modify the configuration of the loading file：

As shown below, open the load file “ble.sct” under the “slave_ota_app” project file：



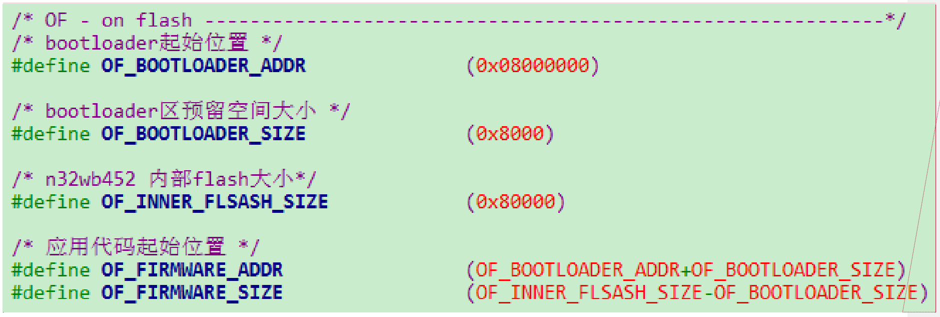
ota_file	2020/6/9 14:11	文件夹	
_Get_OTA_Info.exe	2020/6/8 18:43	应用程序	24 KB
ble.sct	2020/6/4 19:25	Windows Script ...	2 KB
slave_ota_app.uvprojx	2020/6/9 11:48	礦ision5 Project	27 KB

Modify the following address to the address planned by the project. The bootloader here is divided into 32K Flash. Therefore, the starting address of the APP application is configured as 0x08008000;



```
; *************************************************************
; *** Scatter-Loading Description File generated by uVision ***
; *************************************************************

LR_IROM1 0x08008000 0x00080000  {    ; load region size_region
  ER_IROM1 0x08008000 0x00080000  {  ; load address = execution address
   *.o (RESET, +First)
   *(InRoot$$Sections)
   .ANY (+RO)
   .ANY (+XO)
  }
```

At the same time, modify the Flash configuration in the program configuration file n32wb452_boot_conf.h: under the bootloader and app application project:

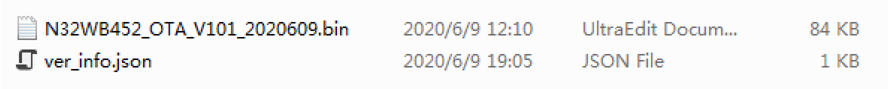


```
/* OF - on flash ------------------------------------------------------------*/
/* bootloader起始位置 */
#define OF_BOOTLOADER_ADDR              (0x08000000)

/* bootloader区预留空间大小 */
#define OF_BOOTLOADER_SIZE              (0x8000)

/* n32wb452 内部flash大小*/
#define OF_INNER_FLSASH_SIZE            (0x80000)

/* 应用代码起始位置 */
#define OF_FIRMWARE_ADDR                (OF_BOOTLOADER_ADDR+OF_BOOTLOADER_SIZE)
#define OF_FIRMWARE_SIZE                (OF_INNER_FLSASH_SIZE-OF_BOOTLOADER_SIZE)
```

批注 [n1]：有中文

3.3 Generation of upgrade package files

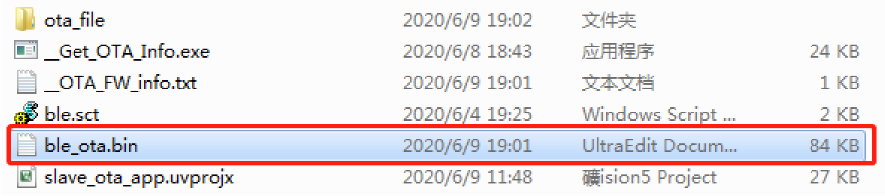
The upgrade package file is a compressed file, such as N32WB452_OTA_V101_2020609.zip, which contains the APP application upgrade firmware bin file and upgrade firmware information json file, as shown below.



N32WB452_OTA_V101_2020609.bin	2020/6/9 12:10	UltraEdit Docum...	84 KB
ver_info.json	2020/6/9 19:05	JSON File	1 KB

3.3.1 Generation of bin file

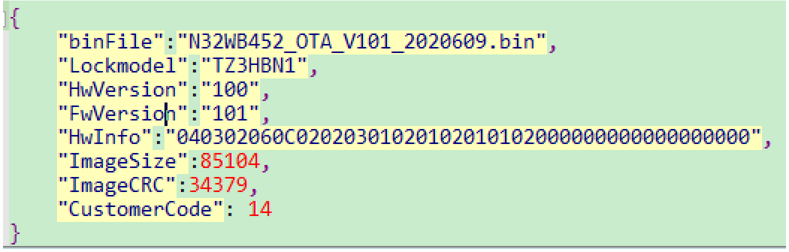
The upgrade firmware bin file can be obtained by compiling the slave_ota_app project, as shown below：



ota_file	2020/6/9 19:02	文件夹	
__Get_OTA_Info.exe	2020/6/8 18:43	应用程序	24 KB
__OTA_FW_info.txt	2020/6/9 19:01	文本文档	1 KB
ble.sct	2020/6/4 19:25	Windows Script ...	2 KB
ble_ota.bin	2020/6/9 19:01	UltraEdit Docum...	84 KB
slave_ota_app.uvprojx	2020/6/9 11:48	礦ision5 Project	27 KB

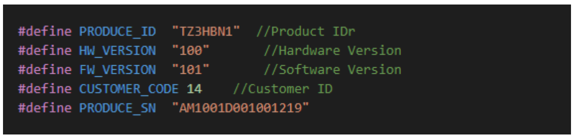
3.3.2 Generation of json file

The json file contains the bin file name (binFile), module name (Lockmode), hardware version (HwVersion), firmware version (FwVersion), firmware size (ImageSize), firmware CRC16 check value (ImageCRC) and customer code (CustomerCode), etc. information；



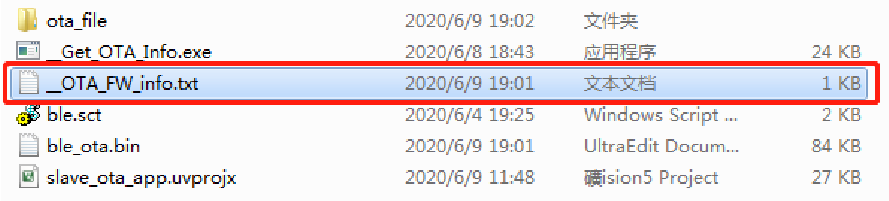
```
{
    "binFile":"N32WB452_OTA_V101_2020609.bin",
    "Lockmodel":"TZ3HBN1",
    "HwVersion":"100",
    "FwVersion":"101",
    "HwInfo":"040302060C02020301020102010102000000000000000000",
    "ImageSize":85104,
    "ImageCRC":34379,
    "CustomerCode": 14
}
```

The information in the json file needs to be synchronized with the firmware information in n32wb452_ble_ota_api.h, as shown below；



```
#define PRODUCE_ID  "TZ3HBN1"  //Product IDr
#define HW_VERSION  "100"       //Hardware Version
#define FW_VERSION  "101"       //Software Version
#define CUSTOMER_CODE 14    //Customer ID
#define PRODUCE_SN  "AM1001D001001219"
```

The firmware size (ImageSize) and firmware CRC16 check value (ImageCRC) can be obtained in the __OTA_FW_info.txt file after the project is compiled.



ota_file	2020/6/9 19:02	文件夹	
__Get_OTA_Info.exe	2020/6/8 18:43	应用程序	24 KB
__OTA_FW_info.txt	2020/6/9 19:01	文本文档	1 KB
ble.sct	2020/6/4 19:25	Windows Script ...	2 KB
ble_ota.bin	2020/6/9 19:01	UltraEdit Docum...	84 KB
slave_ota_app.uvprojx	2020/6/9 11:48	礦ision5 Project	27 KB

3.3.3 Generation of compressed files

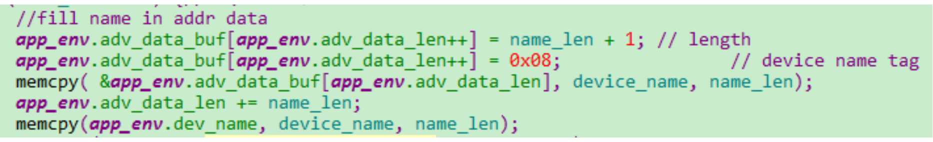
After preparing the bin file and the json file, save them to a folder with the same name as bin, and compress the entire folder into a zip file to get the upgrade package.

3.4 Bluetooth broadcast configuration

批注 [n2]: 页眉的那个标签不在了

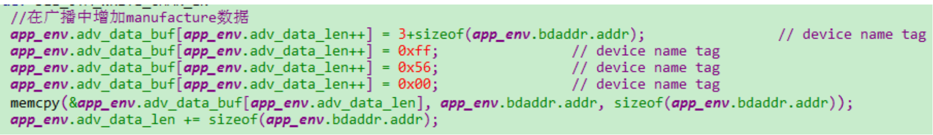
The OTA APP on the mobile phone is to identify the device through the Bluetooth broadcast data, and the Bluetooth-related configuration needs to be set in the program.

Populate Bluetooth name to Bluetooth broadcast data：



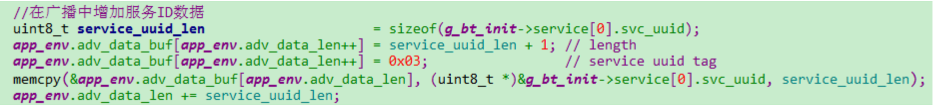
```
//fill name in addr data
app_env.adv_data_buf[app_env.adv_data_len++] = name_len + 1; // length
app_env.adv_data_buf[app_env.adv_data_len++] = 0x08;            // device name tag
memcpy( &app_env.adv_data_buf[app_env.adv_data_len], device_name, name_len);
app_env.adv_data_len += name_len;
memcpy(app_env.dev_name, device_name, name_len);
```

Populate Bluetooth address to Bluetooth broadcast data：



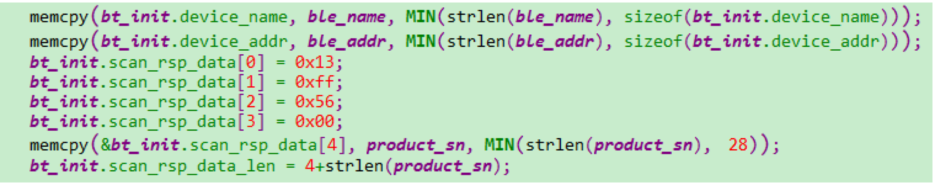
```
//在广播中增加manufacture数据
app_env.adv_data_buf[app_env.adv_data_len++] = 3+sizeof(app_env.bdaddr.addr);              // device name tag
app_env.adv_data_buf[app_env.adv_data_len++] = 0xff;                    // device name tag
app_env.adv_data_buf[app_env.adv_data_len++] = 0x56;                    // device name tag
app_env.adv_data_buf[app_env.adv_data_len++] = 0x00;                    // device name tag
memcpy(&app_env.adv_data_buf[app_env.adv_data_len], app_env.bdaddr.addr, sizeof(app_env.bdaddr.addr));
app_env.adv_data_len += sizeof(app_env.bdaddr.addr);
```

Populate Bluetooth service ID to Bluetooth broadcast data：



```
//在广播中增加服务ID数据
uint8_t service_uuid_len                        = sizeof(g_bt_init->service[0].svc_uuid);
app_env.adv_data_buf[app_env.adv_data_len++] = service_uuid_len + 1; // length
app_env.adv_data_buf[app_env.adv_data_len++] = 0x03;                 // service uuid tag
memcpy(&app_env.adv_data_buf[app_env.adv_data_len], (uint8_t *)&g_bt_init->service[0].svc_uuid, service_uuid_len);
app_env.adv_data_len += service_uuid_len;
```

Populate device SN to Bluetooth scan response data：



```
  memcpy(bt_init.device_name, ble_name, MIN(strlen(ble_name), sizeof(bt_init.device_name)));
  memcpy(bt_init.device_addr, ble_addr, MIN(strlen(ble_addr), sizeof(bt_init.device_addr)));
  bt_init.scan_rsp_data[0] = 0x13;
  bt_init.scan_rsp_data[1] = 0xff;
  bt_init.scan_rsp_data[2] = 0x56;
  bt_init.scan_rsp_data[3] = 0x00;
  memcpy(&bt_init.scan_rsp_data[4], product_sn, MIN(strlen(product_sn),  28));
  bt_init.scan_rsp_data_len = 4+strlen(product_sn);
```

4. Historical Versions

VERSION	DATE	REMARK
V1.0	2020-04-17	Create documentation

5. 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.