User Guide

N32WB03x FLASH User Guide

Introduction

The purpose of this document is to enable users to quickly get familiar with the FLASH usage of the N32WB03x series Bluetooth SOC chips, so as to reduce the preparation time before development and lower the development difficulty.

Table of Contents

1 FLASH Introduction

The FLASH memory of N32WB03x contains the FLASH area and the TRAM storage area as follows:

- FLASH: 256KBytes/512KBytes, used to store user code and data
- TRAM storage area: up to 512 bytes maximum, used to store chip calibration values

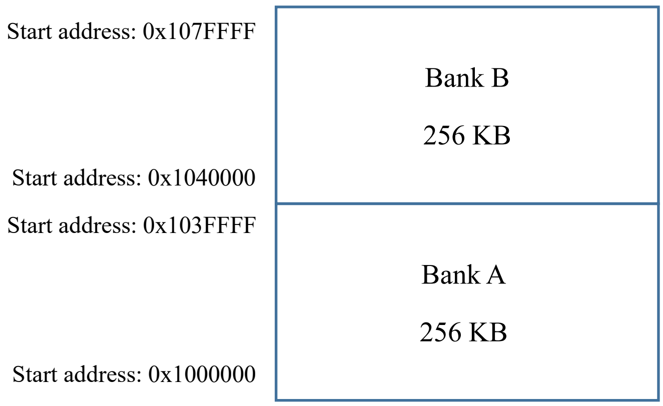
2 Flash

The N32WB03x chip has a 256KB or 512KB FLASH. The user code will be programmed into the FLASH and the kernel can directly address and run from the FLASH at runtime, at this time the FLASH runs in XIP mode. Reading the Bank A area can directly access the address, that is, the FLASH keeps running in XIP mode. Erasing and writing Bank A or B, reading Bank B operations must exit XIP mode, at this time we encapsulate the library function to run in RAM, execute the FLASH operation and then return to the FLASH area to continue running the user code.

2.1 Address Range

The FLASH address range is 0x01000000 - 0x0107FFFF, with a total space of 512K bytes, divided into Bank A and Bank B, with spaces of 256K bytes respectively. Bank A address range is 0x01000000 - 0x0103FFFF, Bank B address range is 0x01040000 - 0x0107FFFF.

The user code can only run in one of the Banks, BankA is used by default.



Start address: 0x107FFFF

Bank B

256 KB

Start address: 0x1040000

Start address: 0x103FFFF

Bank A

256 KB

Start address: 0x1000000

2.2 Erase and Write Operation Time

FLASH write operations must be written page by page, that is, the address must be a multiple of 0x100, otherwise data correctness cannot be guaranteed. The erase operation can only be erased by sector, that is, the address must be a multiple of 0x1000. The erase and write operation times are shown in the table below.

Operation Item	**Typical Value**	**Maximum Value**	**Unit**
Page Write Operation Time (256 bytes)	2	3	ms
Sector Erase (4Kbytes)（4Kbytes）	16	30	ms

2.3 Read Operations

The read, write and erase operation function interfaces of FLASH are as follows. When calling read, write and erase functions, it will jump into RAM and exit XIP mode to operate on FLASH. Therefore, adding compilation of FLASH library n32wb03x_qflash.c will occupy an additional 804 bytes of RAM to store the code.

2.4 Interface Functions

The read, write and erase operation function interfaces of FLASH are as follows. When calling read, write and erase functions, it will jump into RAM and exit XIP mode to operate on FLASH. Therefore, adding compilation of FLASH library n32wb03x_qflash.c will occupy an additional 804 bytes of RAM to store the code.

```
void Qflash_Init(void);                                  //Must init before use
void Qflash_Erase_Sector(uint32_t address);                      //Erase    sector
void Qflash_Write(uint32_t address, uint8_t* p_data, uint32_t len); //Write
void Qflash_Read(uint32_t address, uint8_t* p_data, uint32_t len);  //Read
```

2.4.1 Qflash_Init

Function: Initialize the FLASH library. This function must be called before calling the FLASH read/write/erase function.

Syntax:

```
void Qflash_Init(void);
```

Parameters: None.

Return: None.

Example:

```
Qflash_Init();
```

2.4.2 Qflash_Erase_Sector

Function: Erase a sector of FLASH at the specified address.

Syntax:

```
void Qflash_Erase_Sector(uint32_t address);
```

Parameters:

[in] address: FLASH address to be erased, must be sector starting address.

Return: None.

Example:

```
Qflash_Erase_Sector(0x1040000);
```

2.4.3 Qflash_Write

Function: Write data to the specified FLASH address.

Syntax:

```
void Qflash_Write(uint32_t address, uint8_t* p_data, uint32_t len);
```

Parameters:

[in] address: FLASH address to write, must be page starting address.

[in] p_data: Pointer to the data block to be written to FLASH.

[in] len: Length of data to write to FLASH.

Return: None.

Example:：

```
uint8_t data[] = {"12345"};
Qflash_Write (0x1040000, (uint8_t*)data, 5);
```

2.4.4 Qflash_Read

Function: Read data from specified FLASH address.

Syntax:

```
void Qflash_Read(uint32_t address, uint8_t* p_data, uint32_t len);
```

Parameters:

[in] address: FLASH address to read, must be 4 byte aligned.

[out] p_data: Pointer to data block for storing data read from FLASH.

[in] len: Length of data to read from FLASH.

Return: None.

Example:

```
uint8_t data[5];
Qflash_Read(0x1040000, (uint8_t*)data, 5);
```

2.5 Precautions

- Pay attention to the minimum unit of FLASH operations, reading is 4 bytes, writing is 256 bytes per page, and erasing is 4K bytes per sector.
- Pay attention to the blocking behavior of FLASH write and erase operations, and consider whether the operation time affects other code logic.

- Note that the FLASH erase operation takes 16~30ms. If an erase operation needs to be performed while Bluetooth is connected, it is recommended that the interval between connections exceeds 30ms, otherwise it may cause abnormal Bluetooth disconnection.
- Note that interrupts should be masked to avoid exceptions when performing FLASH operations, this step is included in the driver function.

3 TRAM Storage Area

There is a 512 byte TRAM value storage area on the chip.

3.1 TRAMvalue is declared as the following structure

```
typedef struct{
    uint32_t stote_bg_vtrim_value;
        uint32_t stote_rc28800_trim_value;
        uint32_t stote_rc32000_trim_value;
        uint32_t stote_rc32768_trim_value;
        uint32_t stote_rc64m_trim_value;
        uint32_t stote_rc96m_trim_value;
        uint32_t rc_adc_ts_25c;
        uint32_t rc_gpadc_value_3400mv;
        uint32_t rc_gpadc_value_600mv;
        uint8_t    flash_uuid[16];
}trim_stored_t;
```

The TRAM storage area is read-only, and data must be read using the dedicated function SystemTrimValueGet to return a pointer to the structure.

3.2 Interface Functions

3.2.1 SystemTrimValueRead

Function: Read TRIM storage area data and return pointer to structure.

Syntax:

```
void SystemTrimValueRead(uint8_t* p_data,uint32_t byte_length);
```

Parameters: None

Return:

If non-NULL pointer is returned, it is a pointer to the trim_stored_t structure that has successfully read the TRIM storage data. NULL is returned indicating no TRIM values have been read.

Example:：

```
trim_stored_t trim_stored;
SystemTrimValueRead((uint8_t*)&trim_stored,sizeof(trim_stored));
```

3.2.2 SystemTrimValueGet

Function: Return pointer to chip UUID array, chip UUID length is 16 bytes.

Syntax:

```
trim_stored_t* SystemTrimValueGet(void);
```

Parameters: None

Return:

If non-NULL pointer is returned, it is a pointer to the successfully read UUID array. NULL is returned indicating no UUID value has been read.

Example:

```
trim_stored_t *p_trim;
p_trim = SystemTrimValueGet();
```

3.2.3 SystemGetUUID

Function: Return pointer to chip UUID array, chip UUID length is 16 bytes.

Syntax:

```
uint8_t* SystemGetUUID(void);
```

Parameters: None

Return:

If non-NULL pointer is returned, it is a pointer to the successfully read UUID array. NULL is returned indicating no UUID value has been read.

Example:

```
uint8_t chip_uuid[16];
memcpy(chip_uuid, SystemGetUUID(), 16);
```

3.2.4 SystemGetMacAddr

Function: Return pointer to chip MAC address array, chip MAC address length is 6 bytes. The chip's MAC address consists of 6 bytes (6-11) of the chip UUID.

Syntax:

```
uint8_t* SystemGetMacAddr(void);
```

Parameters: None

Return:

If non-NULL pointer is returned, it is a pointer to the successfully read MAC address array. NULL is returned indicating no MAC address value has been read.

Example:

```
uint8_t chip_mac[6];
memcpy(chip_mac, SystemGetMacAddr(), 6);
```

4 Version History

Version	**Date**	**Changes**
V1.0	2021.10.26	Initial vesion
V1.2	2021.12.23	Add API function description

5 Disclaimer

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