

应用笔记

N32H47x_48x系列BOOT跳转应用笔记

简介

N32H47X_48X 系列 MCU 内嵌有自举程序 (BOOT)，存放在 System Memory 内，用于通过 USART3 或 USB-FS 接口 (全速 USB 设备，DFU 协议) 对用户程序 (Main FLASH) 进行重新编程。

国民技术 MCU 系列产品提供多种启动模式，可通过 BOOT0 引脚和选项字节配置来选择。在实际应用中，MCU 通常设置为 Main Flash 启动模式。如果要使用内嵌的自举程序，必须将 MCU 修改为 System Memory 启动模式后重新上电。有关启动模式的详细说明请参照对应的用户手册。

本文档介绍了一种 BOOT 跳转方法，便于用户在正常使用中不修改启动模式也能使用内嵌的自举模式。

本文档适用于国民技术的 N32H47X_48X 系列产品。

国民技术 版权所有

目录

| | |
|--------------------|----|
| 目录 | II |
| 1. 硬件需求 | 1 |
| 2. 操作方法 | 1 |
| 2.1 参数定义..... | 1 |
| 2.1.1 函数指针..... | 1 |
| 2.1.2 必要参数..... | 1 |
| 2.2 使用方法..... | 1 |
| 2.2.1 系统时钟设置..... | 1 |
| 2.2.2 API 函数 | 3 |
| 2.3 应用示例..... | 5 |
| 2.3.1 BOOT 测试..... | 5 |
| 3. 历史版本 | 7 |
| 4. 声 明 | 8 |

1. 硬件需求

目前 MCU 内嵌的自举程序仅支持 USART3 或 USB-FS 接口接口，对应的 IO 端口分别为 PA9/PA10（USART3）、PA11/PA12（USB），使用前必须确保端口连接可用。

2. 操作方法

2.1 参数定义

2.1.1 函数指针

必须预先定义一个函数指针类型：typedef void (*pFunction)(void);

2.1.2 必要参数

必须预先定义以下几个参数：

```
/* Internal SRAM config fo MCU */
```

```
#define SRAM_BASE_ADDR          (0x20000000)
```

```
#define SRAM_SIZE                (0x24000)
```

```
/* Constant for BOOT */
```

```
uint32_t BootAddr = 0x1FFF01a1,SPAddr = 0x200024a8,_mainAddr = 0x1FFF0189;    // BOOT  
V1.0 Version
```

```
#define SRAM_VECTOR_WORD_SIZE   (128)
```

```
#define SRAM_VECTOR_ADDR        (SRAM_BASE_ADDR+SRAM_SIZE-0x200)
```

注意：

- 1) SRAM_BASE_ADDR 是芯片 SRAM 的起始地址，SRAM_SIZE 为 SRAM 大小，需要根据具体使用的芯片 SRAM 资源修改。用户必须预留 SRAM 最后大小为 0x200 字节的区域用于 BOOT 跳转；
- 2) 其他参数不能修改
- 3) 默认的参数值适用于大部分应用情况，不需要修改。

2.2 使用方法

2.2.1 系统时钟设置

参照下面的函数，将系统时钟设置为 HSI，不建议自己撰写此函数，请直接调用。

```
ErrorStatus SetSysClockToHSI(void)
```

```
{
```

```
uint32_t timeout_value = 0xFFFFFFFF;

ErrorStatus ClockStatus;

RCC_DeInit();

RCC_EnableHsi(ENABLE);

/* Wait till HSI is ready */
ClockStatus = RCC_WaitHsiStable();

if (ClockStatus == SUCCESS)
{
    /* Enable Prefetch Buffer */
    FLASH_PrefetchBufSet(FLASH_PrefetchBuf_EN);

    /* Flash 0 wait state */
    FLASH_SetLatency(FLASH_LATENCY_0);

    /* HCLK = SYSCLK */
    RCC_ConfigHclk(RCC_SYSCLK_DIV1);

    /* PCLK2 = HCLK */
    RCC_ConfigPclk2(RCC_HCLK_DIV1);

    /* PCLK1 = HCLK */
    RCC_ConfigPclk1(RCC_HCLK_DIV1);

    /* Restore HSI to PLL frequency division as default value */
    RCC->PLLCTRL |= 0x00040000;

    /* Select HSI as system clock source */
    RCC_ConfigSysclk(RCC_SYSCLK_SRC_HSI);
}
```

```

    /* Wait till HSI is used as system clock source */
    while (RCC_GetSysclkSrc() != RCC_CFG_SCLKSTS_HSI)
    {
        if ((timeout_value-- == 0)
        {
            return ERROR;
        }
    }
}
else
{
    /* HSI fails */
    return ERROR;
}
return SUCCESS;
}

```

2.2.2 API 函数

调用下面的 API (Jump_To_BOOT), MCU 直接跳转到自举程序 (BOOT)

```

void Jump_To_BOOT(void)
{
    uint8_t value1[8] = {0x00}, value2 = 8;
    uint32_t i, *pVec;

    /* Init vector */
    pVec = (uint32_t *)SRAM_VECTOR_ADDR;
    for(i=0; i<SRAM_VECTOR_WORD_SIZE; i++)
        pVec[i] = 0;

    /* clean up ICache*/
    FLASH_iCacheCmd(DISABLE);
    FLASH_iCacheRST();
}

```

```

/* Get BOOT Version */
Get_NVR(0x1FFF37A8,value1,value2);

if((value1[0] & 0xFF) == 0x11)    //BOOT V1.1 Version
{
    SPAddr = 0x200024C0;
    pVec[SysTick_IRQn+16]          = 0x1FFF2F15;
    pVec[USART3_IRQn+16]           = 0x1FFF2F95;
    pVec[USB_FS_LP_IRQn+16]        = 0x1FFF3095;
}
else                                //BOOT V1.0 Version
{
    pVec[SysTick_IRQn+16]          = 0x1FFF2d31;
    pVec[USART3_IRQn+16]           = 0x1FFF2db1;
    pVec[USB_FS_LP_IRQn+16]        = 0x1FFF2eb1;
}

/* Disable all interrupt */
__disable_irq();

/* Config IWDG */
IWDG_ReloadKey();
IWDG_WriteConfig(IWDG_WRITE_DISABLE);
IWDG_SetPrescalerDiv(IWDG_PRESCALER_DIV256);

/* Config system clock with HSI */
SetSysClockToHSI();

/* Set JumpBoot addr */
pFunction JumpBoot = (pFunction)BootAddr;

/* Initialize Stack Pointer */
__set_MSP(SPAddr);

```

```

/* Enable interrupt */
__enable_irq();

/* Initialize vector table */
SCB->VTOR = SRAM_VECTOR_ADDR;

/* Jump to BOOT */
JumpBoot();
}

```

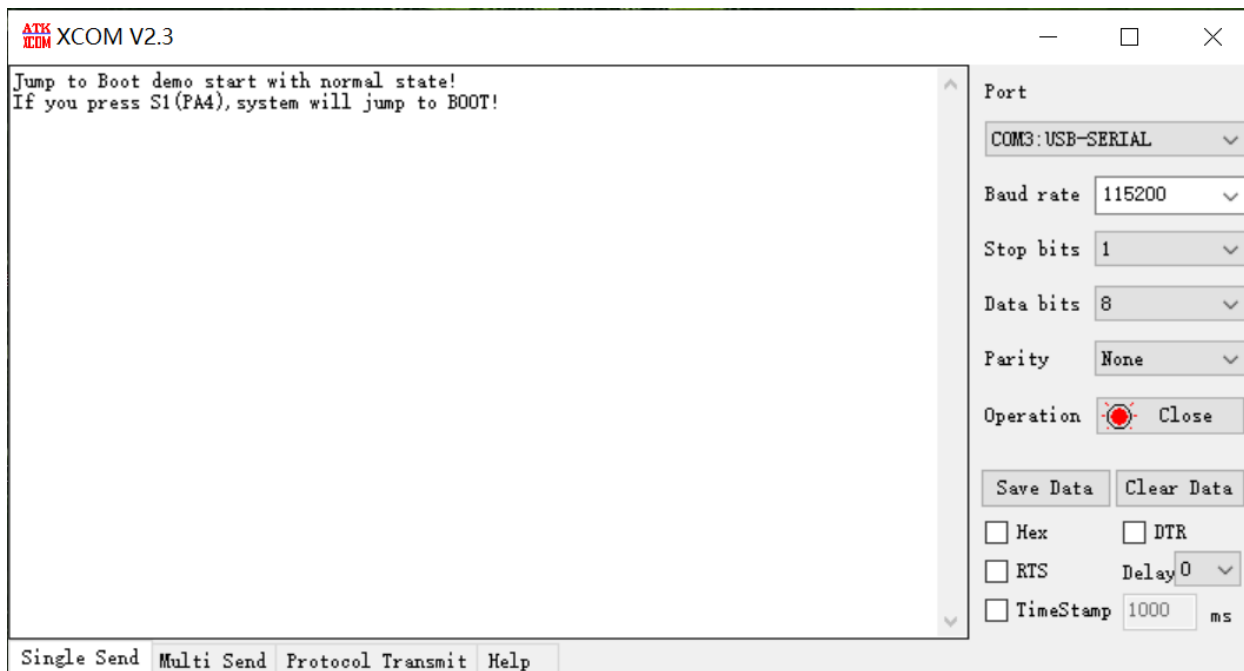
2.3 应用示例

参照示例软件包 Nations.N32H47x_48x_bootjump，演示了如何跳转到 BOOT，跳转成功后可通过 USART3 接口更新程序。

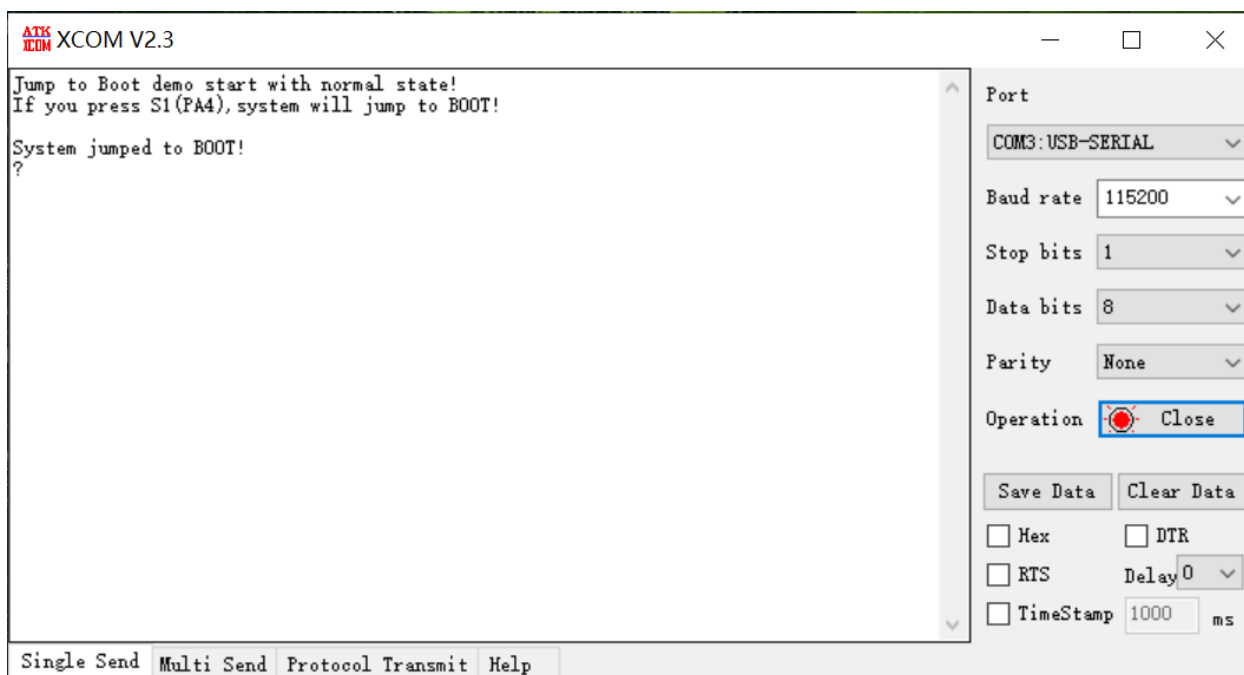
2.3.1 BOOT 测试

基于 N32H487ZEL7-STB V1.0，演示测试流程。

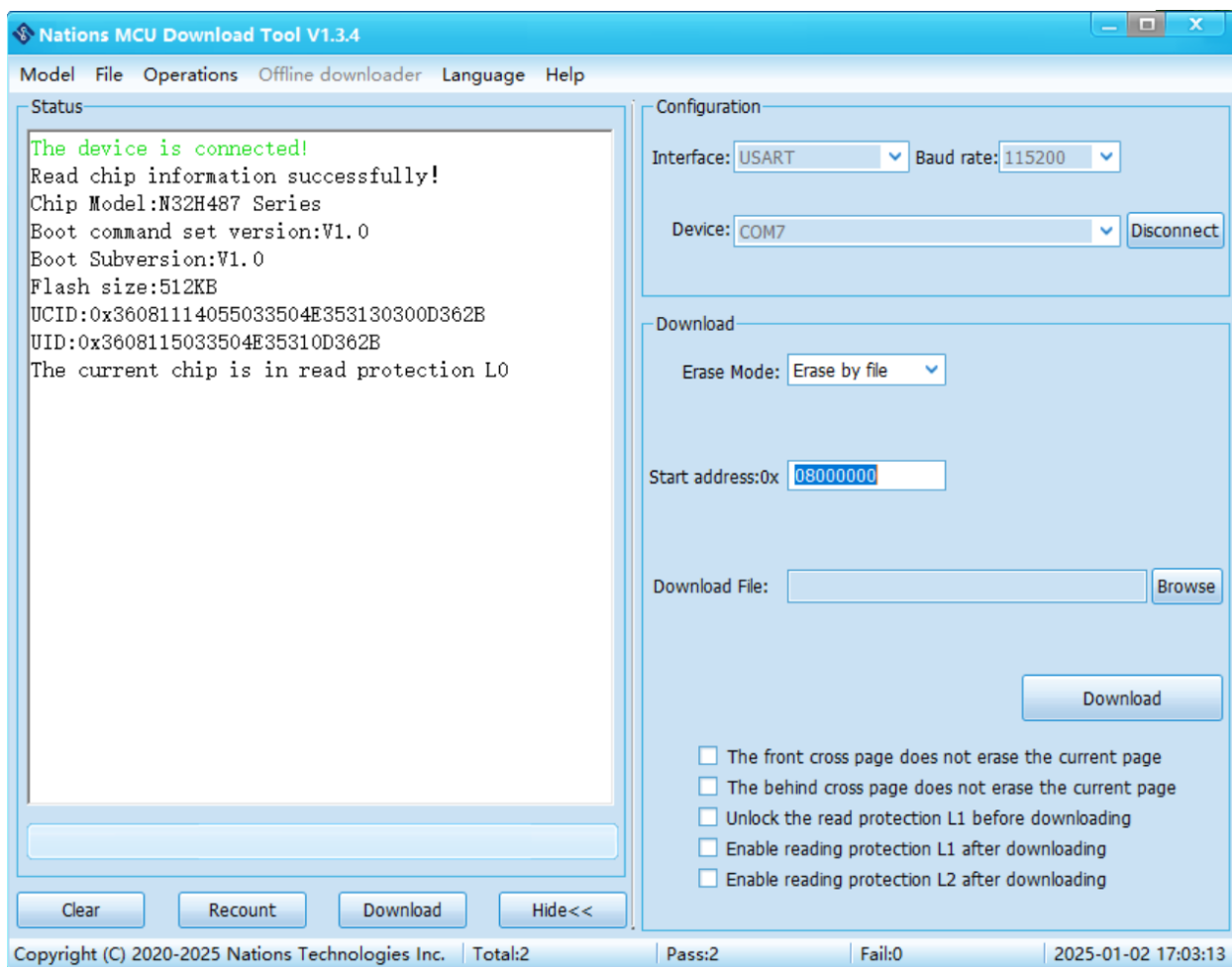
1. 在 KEIL 下将芯片型号改为 N32H487ZEL7，编译后烧录到开发板，通过 USB 线连接 PC，接通电源，可在 PC 上通过串口工具查看提示信息。



2. 在串口工具中打开串口，按下按键 KEY1，系统跳转至 BOOT。



3. 再次在串口工具中关闭串口(若不关闭,下载工具会显示打开串口失败), 通过 BOOT 下载工具连接成功, 如下图所示。



3. 历史版本

| 版本 | 日期 | 备注 |
|--------|------------|------|
| V1.0.0 | 2025-01-02 | 创建文档 |
| | | |

4. 声 明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证责任，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。