
N32H7xx系列软件开发使用指南

简介

此文档的目的在于让使用者能够快速熟悉 N32H7xx 系列微控制器（MCU）的软件使用方法，用于指导用户更容易的了解和使用本产品。

目录

1. 概述	2
2. 系统架构	3
3. 软件功能开发	4
3.1. BOOT 启动	4
3.2. 下载与调试.....	5
3.3. RCC 使用说明	6
3.4. PWR (电源管理模块) 软件配置	7
3.5. SMU (系统管理单元)	8
3.6. 功耗模式使用注意事项.....	11
3.6.1. Standby 模式.....	11
3.6.2. BKP SRAM 在 Standby 模式下保持.....	11
3.6.3. SLEEP 模式下, ITCM 执行代码注意事项.....	11
3.7. FLASH 访问与 CACHE.....	11
3.8. 功能外设的配置注意事项.....	12
3.8.1. ADC.....	12
3.8.2. OTP	12
3.8.3. ECCMON	12
3.8.4. WWDGI.....	12
3.8.5. LPTIM.....	12
3.8.6. XSPI.....	12
3.8.7. SPI.....	13
3.8.8. FDCAN.....	13
3.8.9. RCC.....	13
3.8.10. LI 读保护.....	13
4. 版本历史	14
5. 声明	15

1. 概述

欢迎使用国民技术 N32H7xx 系列芯片，本文介绍了 N32H7xx 系列微控制器（MCU）的软件使用方法以及如何使用各个模块，用于指导用户更容易的了解和使用本产品。

2. 系统架构

N32H7xx 系列高性能 MCU 采用 ARM cortex-M7 + Cortex-M4F 双核架构，内置 2MB/4MB sip flash，内置 1504KB SRAM，其中有 1024KB SRAM 可以配置为 TCM。

支持主流的开发和调试工具，Keil、IAR、Eclips、GCC、GDB、uLink、Jlink、nsLink 等，通过 JTAG/SWD 接口进行调试，支持双核同时 debug（需打开 2 个 IDE，选择不同的 AP 节点进行调试：CM7 选 AP0，CM4 选 AP2）。详情请见 3.2 下载与调试章节。

Cortex-M7 与 Cortex-M4 通过 AXI-AHB 桥进行总线互通，都可以访问所有的 Flash、SRAM 以及所有外设资源（特别注意：M4 不能访问 TCM）。

AHB 上 SRAM5 既是普通 AHB SRAM，也可以配置为 CAN 的专用 RAM，使用时要注意。

为了更好的发挥 N32H7xx 系列芯片的性能，推荐将用户应用代码放到 TCM 中执行。（如何使用 TCM 加速，请参考文档《CN_UG_N32H7xx_TCM_User_Guide.pdf》）

N32H7xx 系列提供了一组 boot API 函数，实现包括选项字节配置、flash 擦写、内存保护等功能。

3. 软件功能开发

3.1. Boot 启动

N32H7xx 系列产品使用了内置的系统引导程序（Boot），在系统启动时进行权限保护管理和用户启动模式配置。

系统启动时始终从 Boot 程序开始执行，之后根据 OTP 和选项字节（Option bytes）参数完成权限配置，再根据 boot0 引脚电平、选项字节和 BTM 字段（详见用户手册 OTPC 章节）完成用户启动。

权限管理相关的部分请参考《CN_UG_N32H7xx_Series_Security_Feature_User_Guide.pdf》。

N32H7xx 分为标准 BOOT 模式和串行下载模式（Bootloader）：

标准 BOOT 模式支持三种启动方式：从 Flash 启动、从 SRAM 启动和安全启动。

1. 当 boot0 引脚为低电平时，启动模式为标准 BOOT 模式，再根据选项字节中的 BOOT_ADDR 字段的地址，确定是从 Flash 或是 SRAM 启动。BOOT_ADDR 字段的值可以是主存储区或 SRAM/ITCM 区的任意地址。
2. 当使能了选项字节中的安全用户模式，并且正确配置了安全用户区之后，每次从标准模式的启动都将强制从安全用户区启动，安全用户区应当预先下载好正确的安全用户代码，否则系统将无法启动。关于安全用户区的有关详情请参考安全功能使用指南。

串行下载模式即系统 bootloader，当 boot0 引脚为高电平时进入串行下载模式。

N32H7xx 还支持启动模式锁定功能，当 BTM 字段为 0x5AA5 时，强制从标准模式启动。当 BTM 字段为 0x4884 时，强制从串行下载模式启动。除此之外，启动模式跟随 boot0 引脚电平。

注意：请谨慎使用此功能，错误操作可能导致芯片锁死。

启动模式选择：

BTM	boot0	安全模式	BOOT_ADDR	启动位置
x (不开锁定)	0	0	0x15000000~ max user flash	Flash
			0x24000000~ max AXI SRAM shared (ITCM/DTCM)	AXI SRAM
			0x30000000~ max AHB SRAM	AHB SRAM
			0x00000000~ max ITCM	ITCM
	1	1	安全用户区	Flash
	1	x	x	串行下载模式
0x5AA5	x	0	0x15000000~ max user flash	Flash
			0x24000000~ max AXI SRAM shared (ITCM/DTCM)	AXI SRAM
			0x30000000~ max AHB SRAM	AHB SRAM
			0x00000000~ max ITCM	ITCM
	1	1	安全用户区	Flash
0x4884		x	x	串行下载模式

3.2. 下载与调试

■ 关闭调试接口

1. 调试接口 JTAG/SWD 默认打开，客户可以根据需要选择是否关闭调试接口，当调试接口被关闭后，也可以再次开启，此配置仅可修改 16 次。
2. 此配置通过修改 OTP 字段（SEC_JTAG_CFG_OTP）实现，修改时需要 BOOT0 拉高，进入串行下载模式，借助国民上位机（Nations MCU Download Tool）进行修改。

■ 调试器连接注意事项

3. 芯片上电后，默认运行 M7，M4 则处于 STANDBY 状态。
4. 调试器连接 M4 前，需要令 M7 先使能 M4(RCC.M4RSTREL=1)，否则调试器找不到 M4 设备。
5. 在任何模式下，当安全 BOOTROM 程序正在执行时，调试器接口被关闭，退出安全 BOOTROM 程序时调试器接口才被恢复。

■ 下载调试注意事项

1. M4 下载调试时需要选择内核（VECEREST/CORE）复位方式，如果选择 SYSRESETREQ 复位方式会复位整个系统，RCC.M4RSTREL 被复位，导致调试器连接不上；
2. KEIL 双核调试时，M7 需选择 SYSRESETREQ 复位方式，此时复位会复位整个系统(包括 M7 和 M4)，复位后由 M7 会启动 M4，M4 启动后，方可正常调试。

3.3. RCC 使用说明

有些外设的时钟是可配置的，用户可以根据需要选择指定的时钟进行配置。但在配置外设时钟之前，应保证相应的时钟已正确配置且稳定运行。时钟可配置的外设如下，详细内容可参考用户手册：

1. SDMMC1 内核时钟源可选 AXI 分频时钟、peri_clk、PLL2A、PLL3A 或 PLL1B；SDMMC2 内核时钟源可选 sys_bus_div_clk 分频时钟、peri_clk、PLL2A、PLL3A 或 PLL1B
2. ADC 的 pll_div_clk 时钟源可选 PLL2B、PLL1B、PLL3B 或 PLL3C
3. Ethercat 内核时钟源可选 sys_bus_div_clk 分频时钟、PLL2B、PLL3A、PLL3C 或 PLL1B
4. Ethernet1 的 GMII_TX_clk 时钟源可选 ETH1_125M_clk、PLL1C、PLL2B 或 PLL3A；Ethernet1 和 Ethernet2 的 ptp_clk 时钟源可选 sys_bus_div_clk 分频时钟、peri_clk、PLL3A 或 PLL2C
5. I2S 的内核时钟源可选 sys_bus_div_clk 分频时钟、PLL3B、HSI 或 CK 引脚
6. I2C 的内核时钟源可选 sys_bus_div_clk 分频时钟、PLL3C、HSI 或 MSI
7. FDCAN 的内核时钟源可选 sys_bus_div_clk 分频时钟、PLL1C、PLL2C、PLL3B 或 peri_clk
8. DSMU 内核 A 的时钟源可选 APB2 时钟、PLL1B、PLL2B、PLL3A、i2s_ckin_clk 或 peri_clk，其中 i2s_ckin_clk 还可进一步选择 I2S1、I2S2、I2S3 或 I2S4 的引脚；内核时钟源可选 sys_bus_div_clk 分频时钟或 APB2 时钟
9. FEMC 的 M0 和 M1 时钟源可选 AXI 分频时钟、peri_clk、PLL2C、PLL3B 或 PLL1B
10. LCD 的 pixel_clk 时钟源可选 AXI 分频时钟、peri_clk、PLL2C 或 PLL3B
11. XSPI 的 ssi_clk 时钟源可选 AXI、PLL3C、PLL1B、PLL2A 或 PLL2C
12. SDRAM 的 mem_clk 时钟源可选 AXI 分频时钟、peri_clk、PLL2A、PLL3A 或 PLL1B
13. RTC 时钟源可选 LSE、LSI 或 HSE 分频时钟
14. DVP 时钟源可选 AXI 分频时钟、peri_clk、PLL2C 或 PLL3A
15. TRNG 时钟源可选 sys_bus_div_clk 分频时钟或 HIS
16. LPTIM 内核时钟源可选 APB5、LSI、LSE、HSE、HSI、MSI、comp1_out、comp2_out、comp3_out 或 comp4_out，当时钟源选择 comp_out 时可选是否加滤波器
17. LPUART 内核时钟源可选 sys_bus_div_clk 分频时钟、HSI、LSE、HSE 或 MSI
18. COMP 低速时钟源可选 LSE 或 LSI

3.4. PWR（电源管理模块）软件配置

为了正确配置内核电压 V_{CAP} 的供电模式，PWR 共有四种供电方式：

- **模式 0：**这是 V_{CAP} 上电时的瞬态电源模式，此时 SMPS 被启用。CM7 启动后，供电模式应配置为以下有效供电配置之一（模式 1/2/3）；
- **模式 1：**SMPS 供电 V_{CAP} ；
- **模式 2：**外部 LDO 供电 V_{CAP} ；
- **模式 3：**通过外部电源给 V_{CAP} 供电， SMPS 关闭；

注意：

1. 模式 0 属于电源的瞬态模式，不能选作为芯片的持久供电模式；
2. 如果软件尝试写入模式 1、2、3 之外的无效配置，则模式配置的寄存器位将保持锁定在模式 0，直到系统重置清除该锁定；
3. 芯片上电后，模式 1/2/3 配置只能操作一次，再次操作将保持锁定状态，系统复位才会清除该锁定；

用户必须在固件库 `system_n32h7xx.c` 中选择正确的 PWR 模式用来匹配外部电路。

例如，如果用户的外部电路设计为使用 SMPS 为芯片供电，为了正确的配置 PWR，用户需要取消 85 行的代码注释 “`#define PWR_SUPPLY_SELECTION (PWR_DIRECT_SMPS_SUPPLY)`”。如下图所示

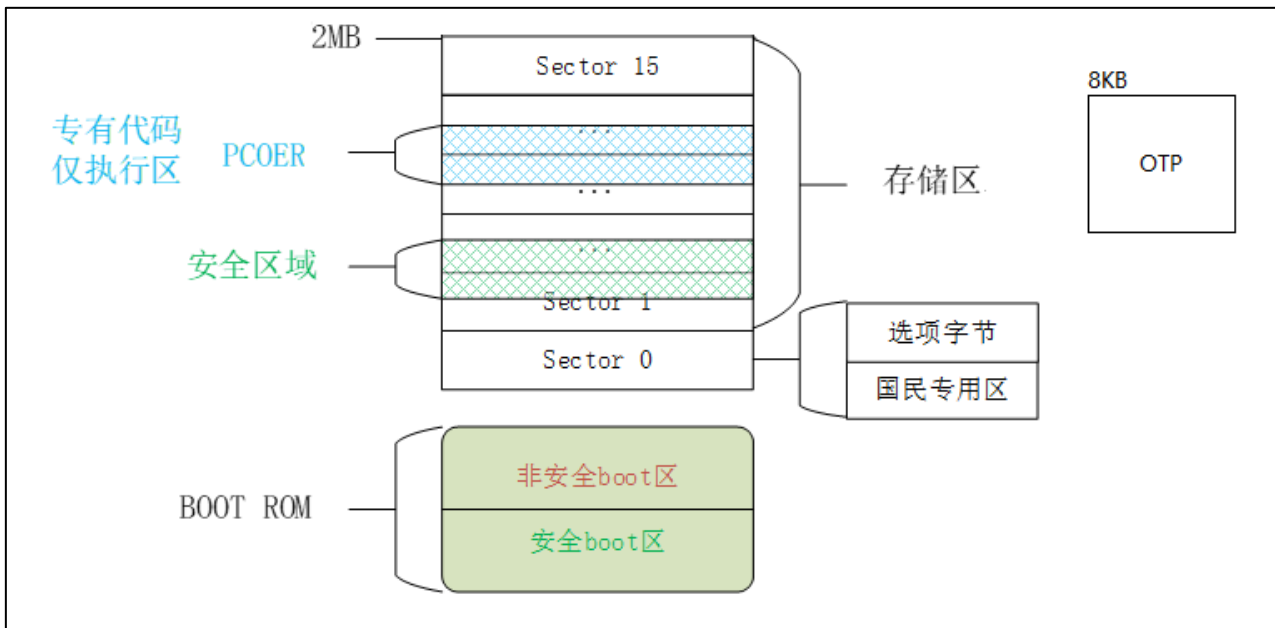
```
79 #define TCM_SIZE_VALUE (0x20) /*TCM_SIZE=0x20:256K·ITCM;256K·DTCM;512K·AXI_SRAM*/
80
81 /**·Private·Functions·**/
82
83 /*·defaulce·power·supply·is·extern·LDO··User·can·change·the·way·of·power·supply*/
84 // #define PWR_SUPPLY_SELECTION (PWR_LDO_SUPPLY) /*·External·LDO·Supply·*/
85 #define PWR_SUPPLY_SELECTION (PWR_DIRECT_SMPS_SUPPLY) /*·DCDC·Supply·*/
86 // #define PWR_SUPPLY_SELECTION (PWR_EXTERNAL_SOURCE_SUPPLY) /*·VCAP·Supply·*/
87
```

类似的， 如果选择选择外部电源通过 VCAP 给芯片供电， 需要上图中注释掉 85 行， 取消 86 行的注释。

3.5. SMU（系统管理单元）

SMU 提供了一套 API 接口用于实现一些列功能，包括 flash 读写、安全存储管理、选项字节（RDP/WRP/安全模式等）、安全区域、仅执行区域、接口配置等等。

存储空间示意图如下，片内有一块 32KB ROM 区域，其分为安全和非安全两部分，安全 boot 区是国民官方的系统启动固件（bootloader），非安全 boot 区中有一系列用户 API。片内集成了 2MB QSPI Flash，其中开头 128KB 是国民内部专用区域，包含国民专有配置和选项字节区。剩余空间是用户主存储区（mainflash）。除此之外，片内还有一块 8KB 的 OTP 区，用于系统配置，其中有 1KB 是用户区。



用户 API 提供了对上述所有区域的操作接口，并提供相应的权限管理和安全功能服务，防止非法访问和篡改，避免敏感信息泄露。（关于系统安全与权限管控策略请参考用户手册相关章节）

用户 API 有两类，一类为 Get 类型（如 GetM4StoreAddrApi），用于获取当前配置参数，另一类是 Set 类型（如 SetM4StoreAddrApi）用于对该参数进行配置。在使用时，应当先通过 Get API 获取当前状态，并对当前状态进行识别，确认需要修改再使用相应的 Set API 进行配置。

Get 类型的 API 在调用时会直接返回当前配置参数，Set 类型 API 将配置参数写入暂存寄存器，然后产生系统复位以生效该配置，（个别 API 不会产生复位，而是会直接生效，视具体 API 而定），产生复位的 API 在调用时不会有返回值。

用户在使用 API 时要考虑启动流程，避免系统反复复位。一般情况下，将只需配置一次之后一直使用的配置写入选项字节或 OTP 中，保证每次上电默认生效。

用户 APIs 列表如下：

函数	功能
uint32_t GetCryptLvlApi(void)	获取当前加密等级
uint32_t SetCryptLvlApi(uint32_t crypt)	设置当前加密等级
uint32_t GetDcDcHvEnApi(void)	获取 DCDC High Core 电压
uint32_t SetDcDcHvEnApi(uint32_t state)	设置 DCDC High Core 电压

uint32_t GetJtagModeApi(void)	获取 JTAG 模式
uint32_t SetJtagModeApi(uint32_t jtag_state,uint32_t* check_key)	设置 JTAG 模式
uint32_t GetM4BootAddrApi(void)	获取 M4 启动地址
uint32_t SetM4BootAddrApi(uint32_t addr)	设置 M4 启动地址
uint32_t GetM7BootAddrApi(void)	获取 M7 启动地址
uint32_t SetM7BootAddrApi(uint32_t addr)	设置 M7 启动地址
uint32_t GetPfoerCfgApi(uint32_t idx)	获取 PFOER 区域
uint32_t GetRdpLvlApi(void)	获取 RDP 等级
uint32_t SetRdpLvlApi(uint32_t rdplvl)	设置 RDP 等级
uint32_t GetSecCfgApi(uint32_t idx)	获取安全区域
uint32_t GetSysCfgBorApi(void)	获取 BORLS
uint32_t SetSysCfgBorApi(uint32_t borcfg)	设置 BORLS
uint32_t GetSysCfgBtmApi(void)	获取 BTM
uint32_t SetSysCfgBtmApi(uint32_t mode)	设置 BTM
uint32_t GetSysCfgNrstIwdgApi(void)	获取 IWDG
uint32_t SetSysCfgNrstIwdgApi(uint32_t nrst_iwdg_cfg)	设置 IWDG
uint32_t GetTcmSzCfgApi(void)	获取 TCM 大小
uint32_t SetTcmSzCfgApi(uint32_t tcm_sz_cfg)	设置 TCM 大小
uint32_t GetWrpApi(void)	获取 WRP
uint32_t SetWrpApi(uint32_t wrp_sector)	设置 WRP

注意：APIs 函数的详细说明请参考安全用户手册相关章节。

除此之外，还有一类**根安全服务函数**，根安全服务是由 Nations 提供的用于配置安全区域功能的代码，只在系统复位后先于其它任何软件被执行，执行完退出后，到下一次复位前无法再被访问。

要使用根安全服务，需要先使能安全模式，安全模式可以通过 PC 工具修改选项字节或者通过调用相应的 API 来使能。

用户通过调用 RSS_xxx api 来执行根安全服务，执行过程中可能会触发多次系统复位。

RSS_resetAndInitializeSecureAreas

原型 uint32_t RSS_resetAndInitializeSecureAreas(RSS_SecureArea_t area)

参数 安全用户区域起始地址和结束地址。可设置安全用户区域。

说明 此服务按照选择字节寄存器中存储的值来设置安全用户区域边界：

- 存储区 1 的 SEC_AREA_START1 和 SEC_AREA_END1

此服务仅在第一次设置安全区域时使用，服务完成后将触发系统复位。

RSS_SecureArea_t 结构体：

```
typedef union
{
    struct
    {
        __IO uint16_t SEC_BEGIN : 13;
        uint16_t : 3;
        __IO uint16_t SEC_END : 13;
        uint16_t : 2;
        __IO uint16_t DMES : 1;
    }
};
```

```
}Bit;  
uint32_t Dword;  
}RSS_SecureArea_t;
```

RSS_exitSecureArea

原型 void RSS_exitSecureArea(unsigned int vectors)

参数 退出后的应用向量跳转的地址

说明 此服务用于退出安全用户软件并跳转至用户主应用，此服务不会触发系统复位。

3.6. 功耗模式使用注意事项

3.6.1. Standby 模式

1. 在进入 Standby 模式前先禁用 BOR 作为系统复位源(RCC_BDCTRL.BORRSTEN=0)，退出后重新使能 BOR 作为复用源；或者在进入 Standby 模式前将 PWR_SYSCTRL2[10]清 0，否则系统进入 Standby 模式后会立即退出 Standby 模式。
2. 在双核产品中，M7 进入 Standby 模式前，确保 M4 也处在 Standby 模式下，M7 进入 Standby 模式后系统才处于 Standby 模式 (SSTBY_C1STBY_C2STBY)，M7 发生唤醒事件才能正常唤醒。

3.6.2. BKP SRAM 在 Standby 模式下保持

如果需要禁用备份 SRAM 在 Standby 模式下保持功能，不管是单核产品还是双核产品，都需要同时设置 PWR_M7CTRL2.BSRSTBRET=0 和 PWR_M4CTRL2.BSRSTBRET=0,否则备份 SRAM 在 Standby 模式保持功能为使能状态。

3.6.3. SLEEP 模式下，ITCM 执行代码注意事项

代码在 ITCM 中执行时，进入 Sleep 模式前必须设置 RCC_AXIEN3.M7TCMLPEN = 1，确保在低功耗模式下 TCM 的时钟不关闭。

3.7. Flash 访问与 Cache

当程序需要访问 Flash 或者程序在 Flash 中执行，不能开启 Cache。

3.8. 功能外设的配置注意事项

3.8.1. ADC

1. ADC_CLK 最大支持配置 20M，超过 20M 会导致 ADC 采样精度受影响。
2. 当需要采集 ADC 的内部通道（Temp Sensor /Vrefint /Vbat）数据时，需要使能 ADC1 的内部通道控制位寄存器，配置 ADC3 来完成内部通道的采样转换。具体可以参考 SDK 中 VREFBUF 的例程。
3. 当一个 ADC 使能多个通道，并且数据的管理方式选择 DSMU 模式，这个 ADC 所有通道数据都被发送到 DSMU，但 DSMU 不区分 ADC 的通道，所有 ADC 通道的数据都被 DSMU 处理

3.8.2. OTP

1. OTP 编程期间，不能产生 NRST 复位，否则 OTP 可能写入错误的值。

3.8.3. ECCMON

1. 使用任何一个或多个 ECCMON 时，都需要启动三个 ECCMON 的时钟，建议使用 ECCMON_EnableClk(void)函数使能 ECCMON 时钟。
2. 如果使能了 SRAM（AXI-SRAM 或者 AHB-SRAM）的 ECC，需要对相应区域的 SRAM 做初始化，即对相应的 SRAM 区域做写 0 操作。

3.8.4. WWDG1

1. 如果 M7 使能了 WWDG1，需要设置 RCC_CFG1.WWDG1RSTEN=1，以确保 WWDG1 的超时复位触发系统复位。

3.8.5. LPTIM

1. 在 DBG_STOP0/DBG_STOP2 模式下，RCC 仅能关闭 LPTIM 内部时钟，外部时钟模式下的 LPTIM 不受影响。

3.8.6. XSPI

1. XSPI 在读写 Memory 的时候配置注意事项：
 - 当传输模式是 Rx-Only 的时候须把 Tx FIFO 的 transfer start fifo 写 1
 - 当传输模式是 Tx-Only 时时时候须把 Tx FIFO 的 transfer start fifo 写 2
 - 当传输模式是 Tx and Rx 的时候须把 Tx FIFO 的 transfer start fifo 写 2

2. XSPI DMA 注意事项

配置 XSPI_TXFT.TXFTST 和 XSPI_DMATDL_CTRL.DMATDL 时需满足以下条件:

- DMATDL \geq TXFTST, 否则 XSPI 会发送失败, 也不会生成 DMA 请求

3.8.7. SPI

1. SPI 在使用的时候需根据主机时钟空闲电平来配置主机 CLK 引脚的内部上/下拉, CLKPOL 为 1 配置为上拉, CLKPOL 为 0 配置为下拉。

3.8.8. FDCAN

1. FDCAN 内核时钟建议配置成 20M、40M、80M, 具体可以参考 SDK 中的 FDCAN_Classic 的 demo。

3.8.9. RCC

1. 如果需要关闭 LSI (RCC_BDCTRL.LSIEN = 0), 需要同时关闭 LSI 时钟安全检测功能 (即将 RCC_BDCTRL.LSICSSSEN 位设置为 0)。
- 2.

3.8.10. L1 读保护

1. M7 在 L1 都保护下, FLASH 受到读保护, 非保护的 SRAM (或 ITCM) 不能访问 FLASH。需要调用 SMU_SetSRAMProtection (或 SMU_SetITCMProtection) 函数将指定的 SRAM (或 TCM) 区设置为受保护区域, 才能正常访问;

4. 版本历史

版本	日期	备注
V1.0.0	2025-04-29	创建文档
V1.1.0	2025-08-20	1.增加 3.5 章节 SMU 描述 2.删除 3.8 章节下的 COMP 的注意事项 3.完善 3.2 章节下载与调试的描述 4.修改 ADC 外设的注意事项描述 5.删除 Flash 执行的程序不能进入 STOP2 模式的章节描述 6.3.8 章节增加 FDCAN 的注意事项 7.3.7 章节增加 RCC 的注意事项 8.修改页眉的 logo
V1.2.0	2025-10-21	1.修改 3.8.7 章节 SPI 的描述 2.新增 3.8.10 章节 L1 读保护的描述

5. 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，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。