

摘要

本技术文档旨在帮助客户实现 IAP 升级提供一个参考方案和模板。本文档提供基于 UART 升级的 MCU 参考代码和 PC 端的升级工具软件。

适用 MCU 型号：CS32F030、CS32F031、CS32F035、CS32F036、CS32F03RA

关键词：IAP、Bootloader、OTA、在线升级、空中升级、串口升级

版本

历史版本	修改内容	日期
V1.0	初版生成	2022-07-07

目录

1 MCU FLASH MEMORY 空间分配.....	3
2 硬件介绍.....	4
3 代码介绍.....	5
4 配合 PC IAP 上位机实际测试.....	7
5 总结.....	10

1 MCU Flash Memory 空间分配

图 1 IAP Flash 空间分配说明

Boot 区域占用 3KB 空间: 0x0800 0000 - 0x0800 0BFF

用户程序占用 29KB 空间: 0x0800 0C00 - 0x0800 7FFF (Flash 大小为 32KB 的 MCU)

用户程序占用 61KB 空间: 0x0800 0C00 - 0x0800 FFFF (Flash 大小为 64KB 的 MCU)

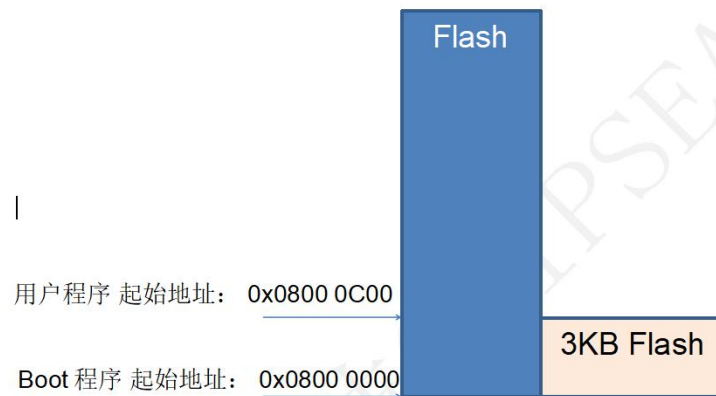


图 1

Flash 地址范围	大小	区域说明
0x0800 0000 - 0x0800 0BFF	3KB	Boot 区域占用空间
0x0800 0C00 - 0x0800 7FFF	29KB	用户程序占用空间
0x0800 8000 - 0x0800 FFFF	32KB	用户程序占用空间

表 1

```
Build Output
compiling cs32f0xx_syscfg.c...
compiling cs32f0xx_usart.c...
compiling system_cs32f0xx.c...
linking...
Program Size: Code=2180 RO-data=288 RW-data=4 ZI-data=1324
FromELF: creating hex file...
".\Objects\project.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:04
```

图 2 工程编译后，大概占用 2.5KB 的空间

2 硬件介绍

- ◆ CS32F03x 开发板
 - ◆ 串口工具
 - ◆ IAP 上位机
 - ◆ Jlink 调试器
- 说明：IAP 例程基于 KEIL 工程

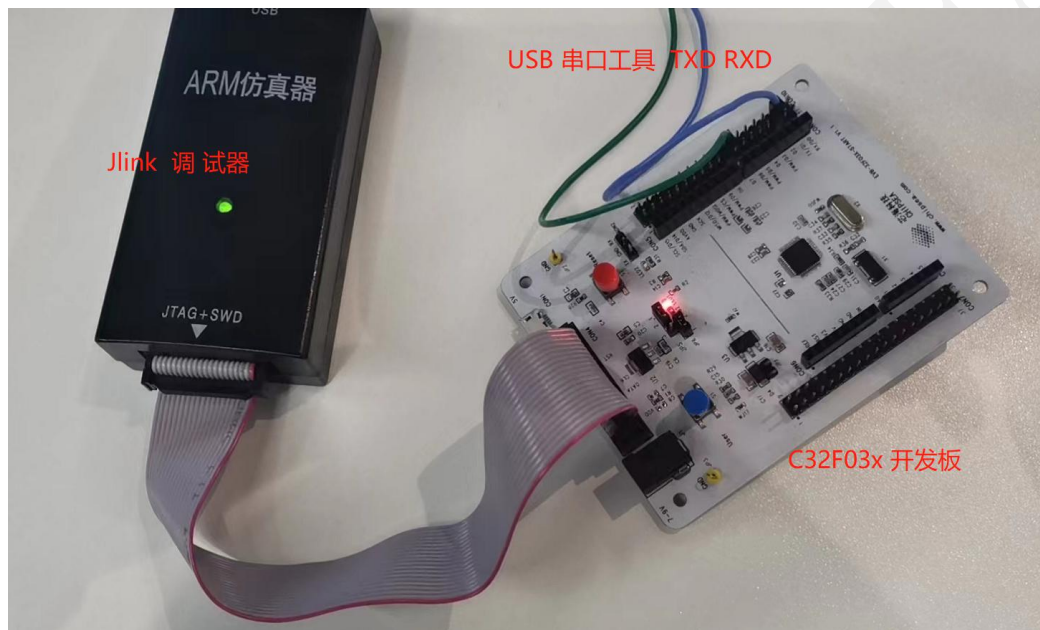


图 3 硬件工具

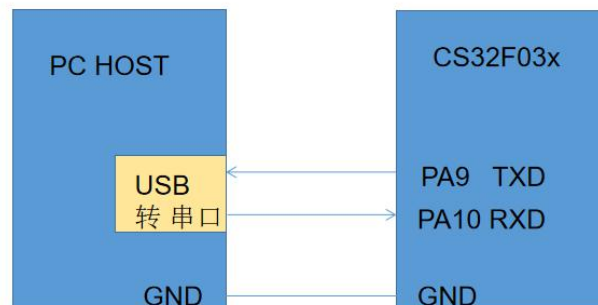


图 4 硬件连接说明

3 代码介绍

CS32F03X SDK 2.0.4 > Boards > EVB_32F03x_START > Applications > IAP_Demo >

名称	修改日期	类型	大小
IAP_Binary_Template	2022/6/13 14:14	文件夹	
IAP_Host	2022/6/13 14:14	文件夹	
IAP_Main	2022/6/13 14:14	文件夹	

图 5 SDK 2.0.4 中的 IAP 参考 DEMO

F030 IAP Boot App > Src > MDK-ARM

名称
Listings
Objects
RTE
JLinkLog.txt
JLinkSettings.ini
Project.uvguix.yuqp
Project.uvoptx
Project.uvprojx

图 6 Boot App 工程文件路径

F030 IAP User App > Src > MDK-ARM

名称
Binary
Listings
Objects
RTE
JLinkLog.txt
JLinkSettings.ini
Project.uvguix.yuqp
Project.uvoptx
Project.uvprojx

图 7 User App 工程文件路径

IAP HOST (Boot) 工程文件代码介绍。

Main 函数中，首先初始化 硬件外设，然后初始化串口。如果 MCU Flash 中已经有了用户程序，则调用延时函数，超时就直接跳到用户程序执行。如果没有用户程序，或者用户程序不完整，则执行用户程序升级。如果超时或者接收到跳转命令，则直接跳到 用户程序执行。

```
int main(void)
{
    /* Hardwrae init */
    hal_init();
    comm_init();

    /* App validate, good app - enable timeout, bad app - always loop */
    if(csboot_app_validate() != false)
    {
        /* Enable timeout */
        hal_timeout_enable();
    }

    while(1)
    {
        /* Polls and process command */
        ret = comm_process();

        /* Jump to app if timeout or received jump command */
        if((ret == RET_JUMP_TO_APP) || (hal_is_timeout() == 1))
        {
            csboot_app_jump_to_app();
        }

        /* Disable timeout if received validated command */
        if(ret == RET_OK)
        {
        }
    }
}
```

```

        hal_timeout_disable();
    }
}
    
```

以下是硬件初始化函数，配置时钟，配置 Flash 参数，配置 UART PIN 脚和波特率，开始超时控制等。

```

void hal_init(void)
{
    /* RCU Connfig */
    RCU->AHBEN |= RCU_AHB_PERI_GPIOA | RCU_AHB_PERI_CRC | RCU_AHB_PERI_FMC;
    RCU->APB2EN |= RCU_APB2_PERI_SYSCFG;

    /* FLASH Connfig */
    FLASH->CCODE = FMC_CCOD0_VALUE1;
    FLASH->CCODE = FMC_CCOD0_VALUE2;
    FLASH->OBCCODE = FMC_OBCCOD0_VALUE1;
    FLASH->OBCCODE = FMC_OBCCOD0_VALUE2;
    FLASH->STS = FLASH_FLAG_ENDF|FLASH_FLAG_PGERR|FLASH_FLAG_WPERR;

    /* USART Config PA9/TX, PA10*/
    RCU->APB2EN |= RCU_APB2_PERI_USART1;

    GPIOA->MSEL[1] |= (GPIO_MF_SEL1 << (((uint32_t)9 & 0x7) * 4)) | (GPIO_MF_SEL1 << (((uint32_t)10 & 0x7) * 4));
    GPIOA->PFR |= ((uint32_t)0x02 << (9 * 2)) | ((uint32_t)0x02 << (10 * 2));
    GPIOA->OSR |= ((uint32_t)GPIO_SPEED_MEDIUM << (9 * 2)) | ((uint32_t)GPIO_SPEED_MEDIUM << (10 * 2));

    USART1->CTR1 = 0x0000000C;
    USART1->BRT = 0x01A1; // 115200bps
    USART1->CTR1 |= USART_CTR1_RXNEIE;
    USART1->CTR1 |= USART_CTR1_UEN;

    NVIC_EnableIRQ(IRQn_USART1);

    /* SysTick Config, 1ms interval */
    SysTick_Config(SystemCoreClock / 1000);
    NVIC_SetPriority(SysTick_IRQn, 0);
    hal_timeout_disable();
}
    
```

以下跳转函数。

```

#define CSBOOT_APP_BASE ((uint32_t)0x0800C00)

void csboot_app_jump_to_app(void)
{
    memcpy((void*)Vectors, (void*)CSBOOT_VECTOR_BASE, CSBOOT_VECTOR_SIZE); //copy interrupt vector table to
sram.

    if((*(__IO uint32_t*)CSBOOT_APP_BASE) & 0xFFFF0000) == 0x20000000)
    {
        SYSCFG->RMAPCFG |= SYSCFG_MEM_REMAP_SRAM;
        hal_deinit(); //close interrupt and clock and so on.
        JumpAddress = *(__IO uint32_t*) (CSBOOT_APP_BASE + 4); // Jump to user application
        Jump_To_Application = (pFunction) JumpAddress;
        set_MSP(*(__IO uint32_t*) CSBOOT_APP_BASE); // Initialize user application's
Stack Pointer
        Jump_To_Application();
    }
}
    
```

IAP Binary Template (UserApp) 用户程序 Keil 工程文件配置.

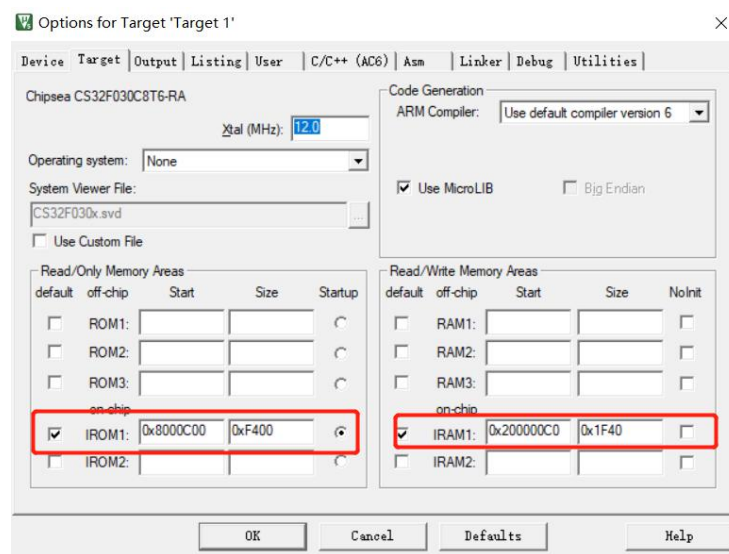


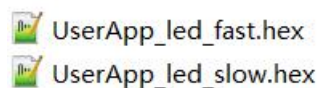
图 用户程序 KEIL 工程配置说明

用户程序 Mian 函数，PA5 电平翻转，PA5 在开发板上连接到一个 LED。

```
int main(void)
{
    gpio_config();
    SysTick_Config(SystemCoreClock / 1000);

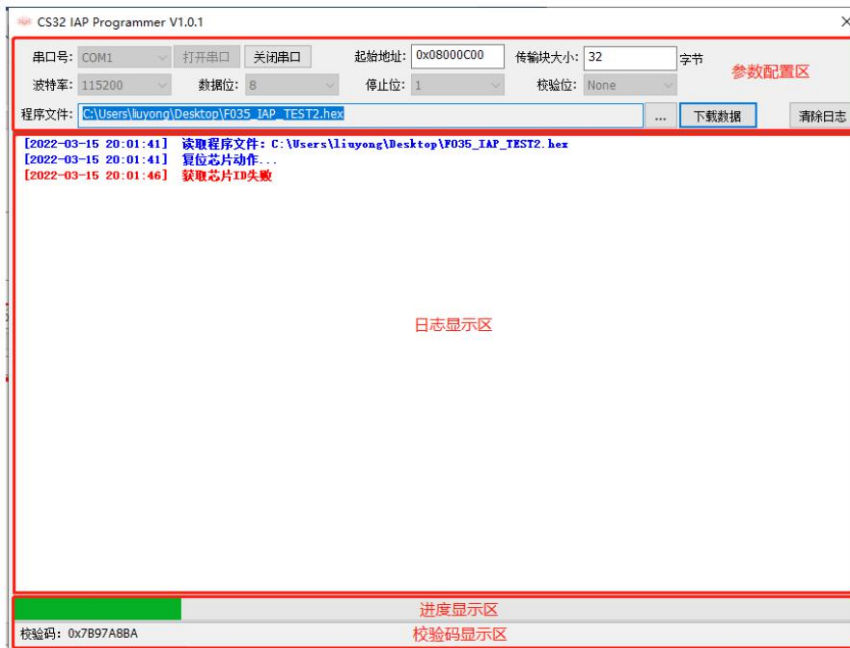
    while (1)
    {
        _GPIO_PIN_SET(GPIOA, GPIO_PIN_5);
        delay(200);
        _GPIO_PIN_RESET(GPIOA, GPIO_PIN_5);
        delay(200);
    }
}
```

用户程序 Mian 函数，修改延时函数的参数，使得 LED 快速闪动和慢速闪动，并生成两个 hex 文件，用于后面做实测测试。

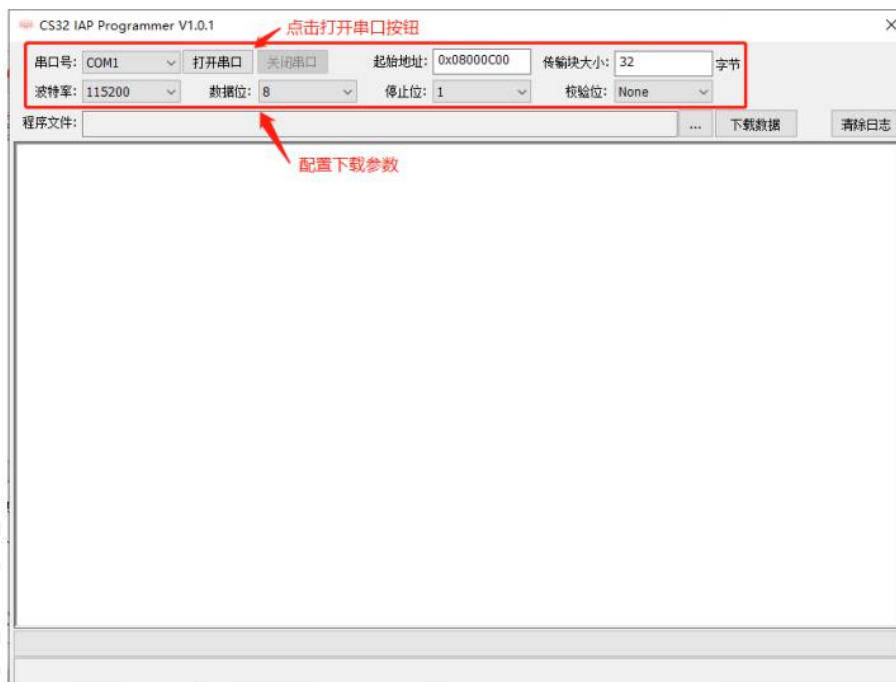


4 配合 PC IAP 上位机实际测试

打开 IAP 上位机升级工具 CS32 IAP Programmer。



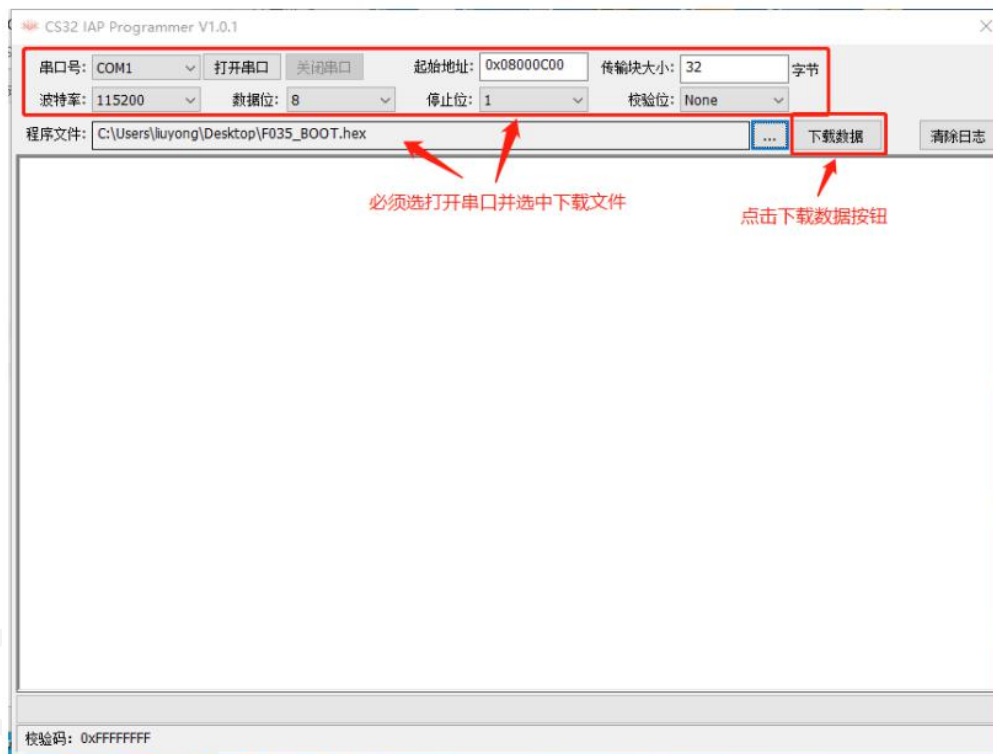
打开串口，配置波特率



选择上一步生成的 LED 闪灯 Hex 文档

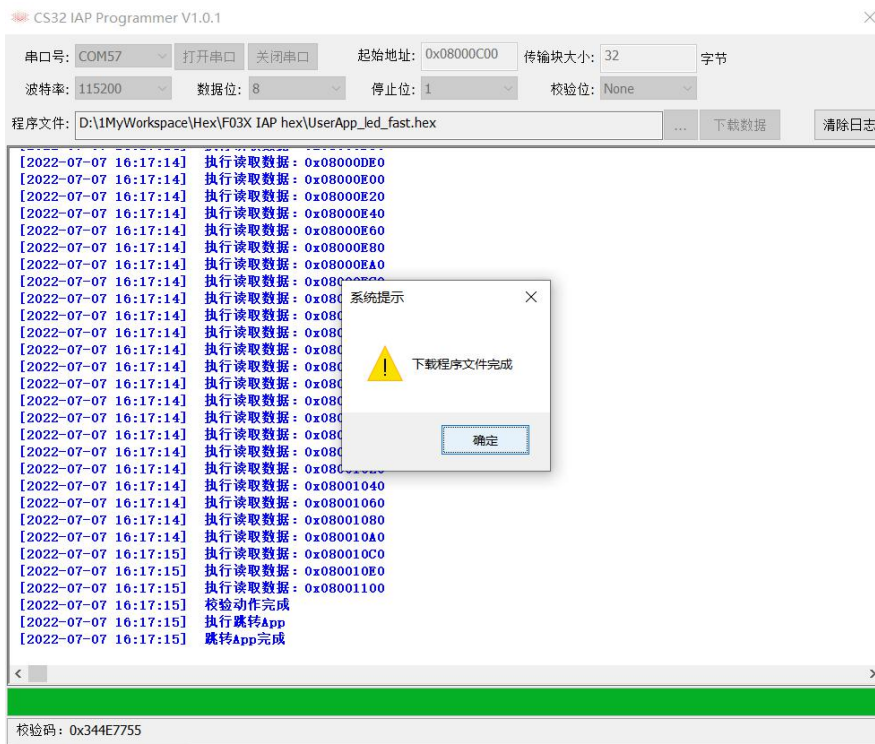


选择下载数据



注：在下载数据前，必须先打开串口并配置好下载文件

IAP 程序升级成功，开发板上 LED 灯会闪动起来。



5 总结

本文档介绍了 CS32F03x MCU 通过 UART 接口来升级用户程序的基本功能，并且提供了参考的下位机 Bootloader 工程文件，用户程序工程文件，上位机 IAP 升级工程文件。用户可以根据自己的实际需要，增加相应功能来实现不同实际需求的应用。

免责声明和版权公告

本文档中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

本文档可能引用了第三方的信息，所有引用的信息均为“按现状”提供，芯海科技不对信息的准确性、真实性做任何保证。

芯海科技不对本文档的内容做任何保证，包括内容的适销性、是否适用于特定用途，也不提供任何其他芯海科技提案、规格书或样品在他处提到的任何保证。

芯海科技不对本文档是否侵犯第三方权利做任何保证，也不对使用本文档内信息导致的任何侵犯知识产权的行为负责。本文档在此未以禁止反言或其他方式授予任何知识产权许可，不管是明示许可还是暗示许可。

Wi-Fi 联盟成员标志归 Wi-Fi 联盟所有。蓝牙标志是 Bluetooth SIG 的注册商标。

文档中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

版权归 © 2022 芯海科技（深圳）股份有限公司，保留所有权利。



芯海科技
CHIPSEA

股票代码:688595