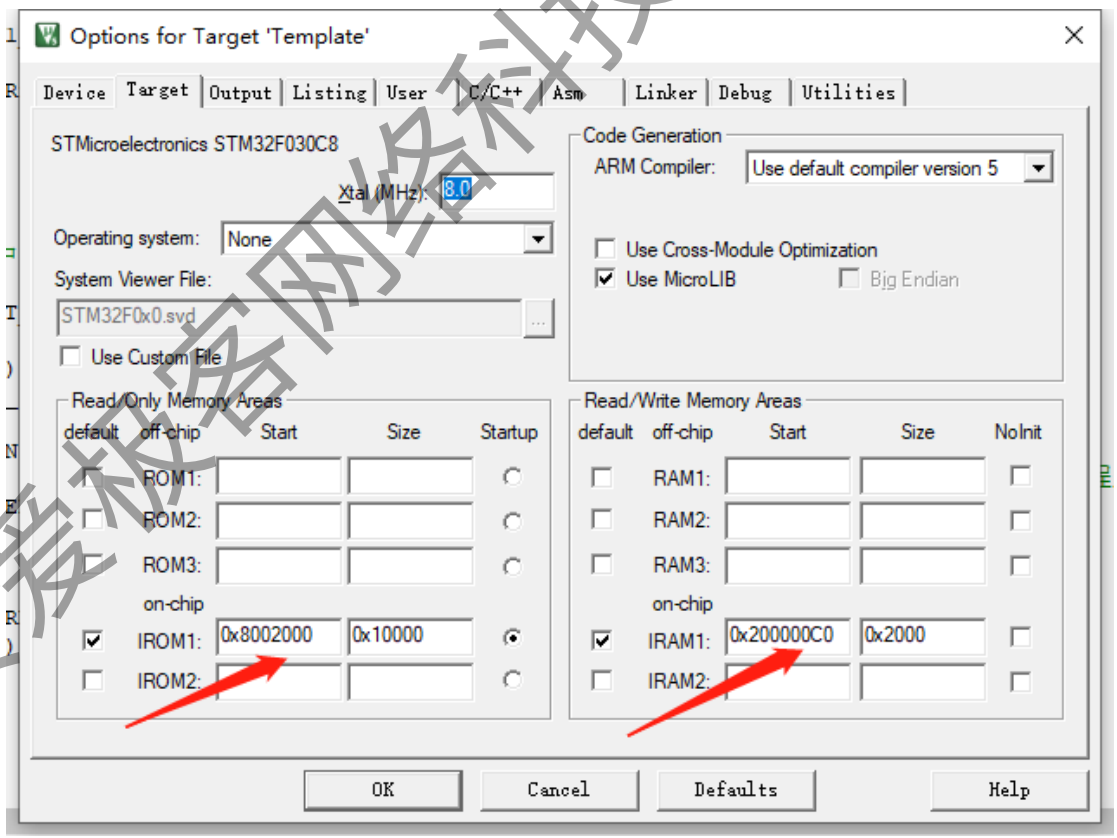
1.移植说明

我们将 STM32F030 的 FLASH 划分为两个区域，分为 Bootloader 区和 APP 区，我们规定 Bootloader 占用 8KB 字节的空间，剩下的空间分给 APP 程序。

2.APP 程序移植说明

1. 设置相关地址



如图中所示，将 IROM1 的起始地址设为 0X8002000，将 IRAM1 起始地址设为 0X200000C0.

2. 在 APP 程序中加入中断向量的重定位函数

```c
int main(void)
{
    IAP_Set();
    delay_init();////定时器初始化
    led_init();///LED灯初始代
    uart1_init(9600);
    DMA1_Init();
    //生成CRC表
    CRC32TableCreate();
    while (1)
    {
        //串口1发送数据，点灯，用于测试升级APP是否成功，实际移植中不需要
        Uart1_Tx[0] = 0XAA;Uart1_Tx[1] = 0X01;Uart1_Tx[2] = 0X02;Uart1_Tx[3] = 0X03;Uart1_Tx[4] = 0X88;
        Uart1_Start_DMA_Tx(5);
        GPIO_ResetBits(GPIOB, GPIO_Pin_4);
        delay_ms(100);
        GPIO_SetBits(GPIOB, GPIO_Pin_4);
        delay_ms(100);
    }
}
//将APP的中断向量表拷贝到SRAM里面去
void IAP_Set(void)
{
  uint32_t i = 0;
/* Relocate by software the vector table to the internal SRAM at 0x20000000 ***/
  /* Copy the vector table from the Flash (mapped at the base of the application
     load address 0x08003000) to the base address of the SRAM at 0x20000000. */
  for(i = 0; i < 48; i++)
  {
    *((uint32_t*)(0x20000000 + (i << 2)))=*(__IO uint32_t*)(APPLICATION_ADDRESS + (i<<2));
  }
  /* Enable the SYSCFG peripheral clock*/
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);
/* Remap SRAM at 0x00000000 */
  SYSCFG_MemoryRemapConfig(SYSCFG_MemoryRemap_SRAM);
}
```

3. 在串口中断中加入上位机更新固件的判断函数

```c
//判断是否接收到上位机的更新固件要求
void IAP_BootLoad_UpData(void)
{
    //上位机请求刷固件
    if((Uart1_Rx[0] == 0xF1) && (Uart1_Rx[1] == 0x01))
    {
        if(check_data(Uart1_Rx,Uart1_Rx_length))//crc校验成功
        {
            systeam_ReStart();
        }
    }
}
void USART1_IRQHandler(void) //串口1中断服务程序
{
    if(USART_GetITStatus(USART1, USART_IT_IDLE) != RESET)
    {
        DMA_Cmd(DMA1_Channel3,DISABLE);
        USART_ClearFlag(USART1, USART_FLAG_IDLE);
        USART1->RDR;
        Uart1_Rx_length = UART1_RX_LEN - DMA_GetCurrDataCounter(DMA1_Channel3);
        //串口空闲中断中调用分析函数，当在APP中接收到上位机发来的更新固件请求时，复位单片机，使其进入Bootloader更新程序
        if(Uart1_Rx_length<UART1_RX_LEN)
        {
            IAP_BootLoad_UpData();
        }
        DMA1_Channel3->CNDTR = UART1_RX_LEN;
        DMA_Cmd(DMA1_Channel3, ENABLE);
    }
}
```

3. Bootloader 更新程序协议说明

IAP 固件升级【串口】

通信协议

帧头：0xF1

1. 请求更新固件

上位机发：

0x01：

帧头【1】+ 功能码【1】+ 固件长度【4】+ 分包数【2】+ CRC32【4】

单片机响应：

0x81:

帧头【1】+ 功能码【1】+ 固件长度【4】+ 分包数【2】+ CRC32【4】

2. 发送数据包

上位机发：

0x02:

帧头【1】+ 功能码【1】+ 包索引【2】+数据【1024】+ CRC32【4】

单片机响应：

0x82:

帧头【1】+ 功能码【1】+ 包索引【2】+ 结果【1】+ CRC32【4】

结果说明：

0x01:正常；

0x02:CRC 校验错误；

0x03:长度不够，丢失数据；

3. 异常处理

1. 请求更新固件无响应：重新请求，直到用户取消或超时；
2. 发送数据包响应超时：重发；
3. 发送数据包响应 CRC 错误或长度错误：重发；

4. 备注

1. 数据高位在前，低位在后；

5. 单片机流程

1. 判断是否有刷固件请求【等待 150ms】；
2. 如果没有请求，跳转到主程序；

3. 如果有请求，则响应，接收固件，完成后跳转至主程序；

6. 上位机机流程

1. 加载固件；

2. 点击下载，间隔 50ms 发送一次请求；

3. 如果响应正确，开发分包发送固件，发送失败或响应报错则自动重发【不限次数】，全部发完后结束，刷固成功；

4.上位机更新程序说明

1. 打开串口，波特率选择 9600

2. 点到升级固件界面，点击选择固件，找到准备升级的固件，选择固件

3. 点击下载即可