BOOT 区域作为用户区使用说明

版本：V1.2

https://wch.cn

一、 使用场景

1. 当用户代码量过大时，可以通过该功能，将芯片 BOOT 区用于存放部分用户代码。
2. 适用芯片：CH32V003、CH641、CH32V002/4/5/6/7、CH32X035

二、 使用方法

方法一：直接替换官方 EVT 例程的 LD 链接文件和启动文件，具体修改步骤如下。下面以 CH32X035 系列芯片为例，将新工程的 LD 链接文件和启动文件替换为“BootAsUser”工程中 Ld 和 Startup 文件夹下的 LD 链接文件和启动文件，如图所示。

其次在修改后的工程中，在函数前添加 attribute 关键字，则可以将该函数存放在 BOOT 区域，attribute 关键字具体使用格式如下图所示。

```
*/
__attribute__((section(".Bcode")))
void GPIO_Toggle_INIT(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
}
```

方法二：
在现有工程中修改，具体修改步骤如下
1.修改 LD 链接文件

①在 LD 文件中添加新的内存分配区域，如图新增内存段 ‘BFLASH’，起始地址 0x1fff0000，大小 3328Byte；FLASH 和 RAM 的大小根据当前使用具体的系列芯片修改。

```
MEMORY
{
    BFLASH (rx) : ORIGIN = 0x1fff0000, LENGTH = 3328
    FLASH (rx) : ORIGIN = 0x00000000, LENGTH = 62K
    RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 20K
}
```

②在 SECTIONS 段描述中添加新增内存区域的内存分配定义

```
.bbinit :
{
  . = ALIGN(4);
  KEEP(*(SORT_NONE(.bxx)))
    . = ALIGN(4);
    *(.Bcode);
    *(.Bcode.*);
  . = ALIGN(4);
} >BFLASH AT>BFLASH
```

2. 修改启动文件，在启动文件起始处添加如下代码，新增“.bxx”代码段，属性为可执行可读；之后执行软复位，跳转到用户区。

```
.section    .bxx,"ax",@progbits
.align  2
lui a4,0x80000
lui a5,0xe000f
sw  a4,-752(a5)

.section    .init,"ax",@progbits
```

3. 同样在函数前添加 attribute 关键字，其中 section()中的名称“.Bcode”与②中定义的一致，则可以将该函数存放在 BOOT 区域。

```
*/
__attribute__((section(".Bcode")))
void GPIO_Toggle_INIT(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
}
```

三、 注意事项

1. BOOT 区域大小为 3328Byte，实际用户代码应不超过（FLASH+BFLASH）区域总的空间大小，比如当前应不超过 62K+3328Byte。

2. 在 BOOT 区域与用户 FLASH 区域之间的函数跳转理论上有 1us 的延时(48M1 等待条件下)。（V006\X035 系列芯片没有额外的跳转延时时间，忽略注意事项 2）

3. BOOT 区域不支持用户代码擦除，所以数据段不能被存储在 BOOT 区域。

4. 当前功能仅可使用 Link 工具下载 Hex 文件实现。

5. 使用该功能后原本的 BOOT 区域内的代码将会被覆盖，此时将不能使用 Boot 跳转相关功能（006、003、641 系列芯片则会出现程序上电不跑的情况，原因是芯片无 BOOT 程序，可以通过重新下载 BOOT 或者将上电启动位置切换为用户区可解决该问题）。

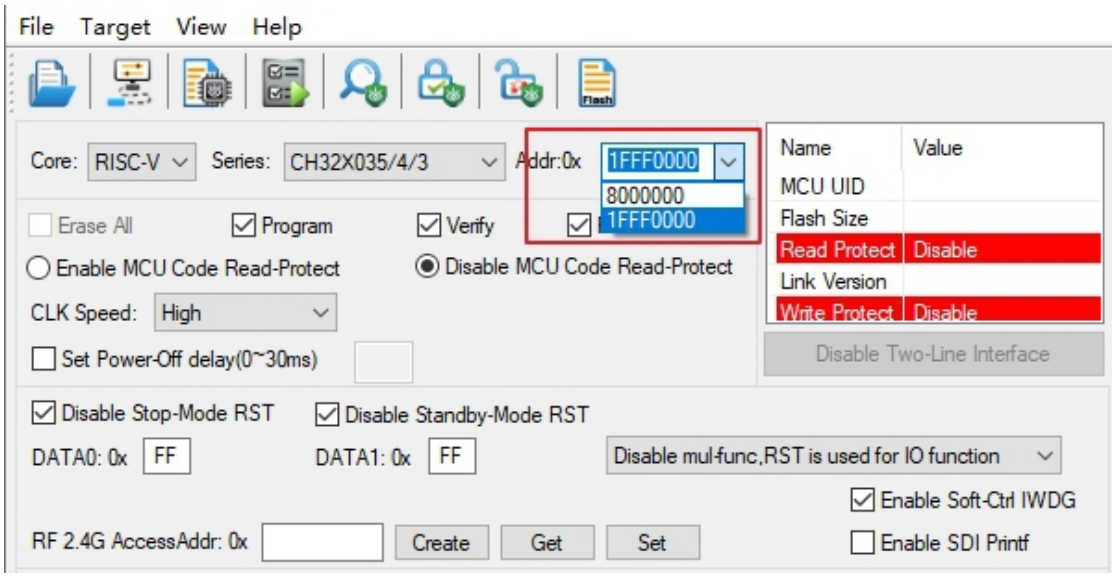
6. 在当前支持该功能的芯片中需要特别注意 BOOT 区域的起始地址和 BOOT 区域实际分配大小的差异性。

7. 对于有额外跳转时间的芯片（003、641），放在 BOOT 中的函数，每次取址都会增加固定的跳转时间，因此在使用该功能时，应当避免将频繁调用的函数放在 BOOT 中，而 BOOT 中适宜存放一次性执行的初始化函数等。

四、 下载方式

使用最新的 WCH-Link Utility 工具，版本号大于 2.60，并且想要将 BOOT

区域作为用户区使用需要将程序下载地址设置为 0x1fff0000(新增内存段 BFLASH 首地址)。以 CH32X035 系列芯片为例，下载工具如下图所示，在下载前设置下载地址为 0x1fff0000。



五、 相关链接

沁恒微电子社区：https://www.wch.cn/bbs/forum-106-1.html

沁恒官网：https://www.wch.cn/

WCH-Link 使用说明：https://www.wch.cn/products/WCH-Link.html