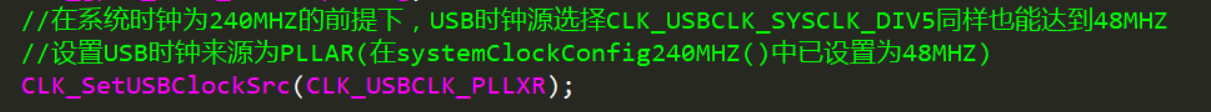
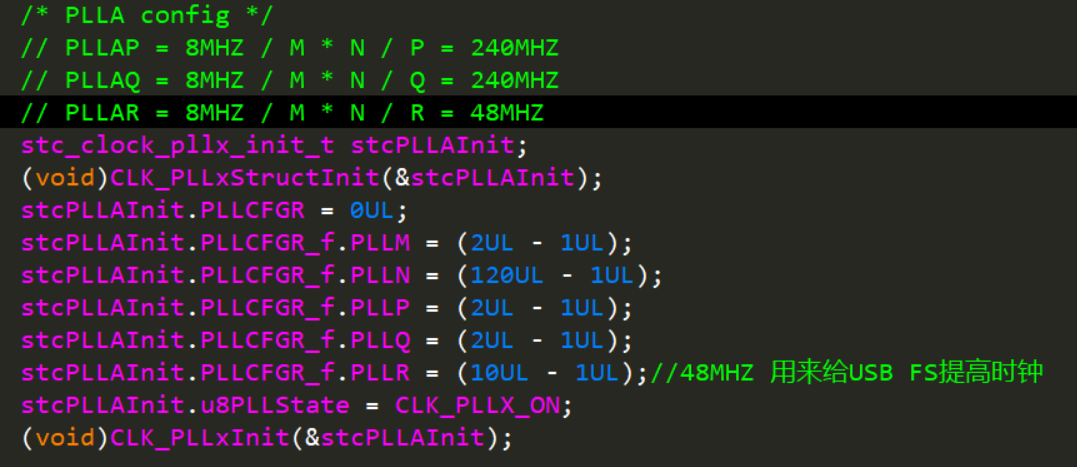
实现MSC功能，作为device枚举出两个磁盘，一个基于SD卡，一个基于NOR Flash(W25Q32)。

CherryUSB V1.4.0

1. 准备一个移植好FatFS文件系统的工程，能实现在SD卡和NOR Flash上挂载文件系统。
2. 配置好合适的系统时钟，如果系统时钟为240MHZ，USB时钟源选择CLK\_USBCLK\_SYSCLK\_DIV5即系统时钟5分频，同样也能达到48MHZ。从芯片参考手册4.4的表4-2可以看出，USBFS/USBHS的时钟来源为UCLK，而UCLK的时钟来源为PLLAP,PLLAQ,PLLAR,PLLHQ,PLLHR从4.2.1的时钟树也可以看出这一点，本次工程不用系统时钟做为USB的时钟源，用PLLAR做USB的时钟源，用如下函数设置

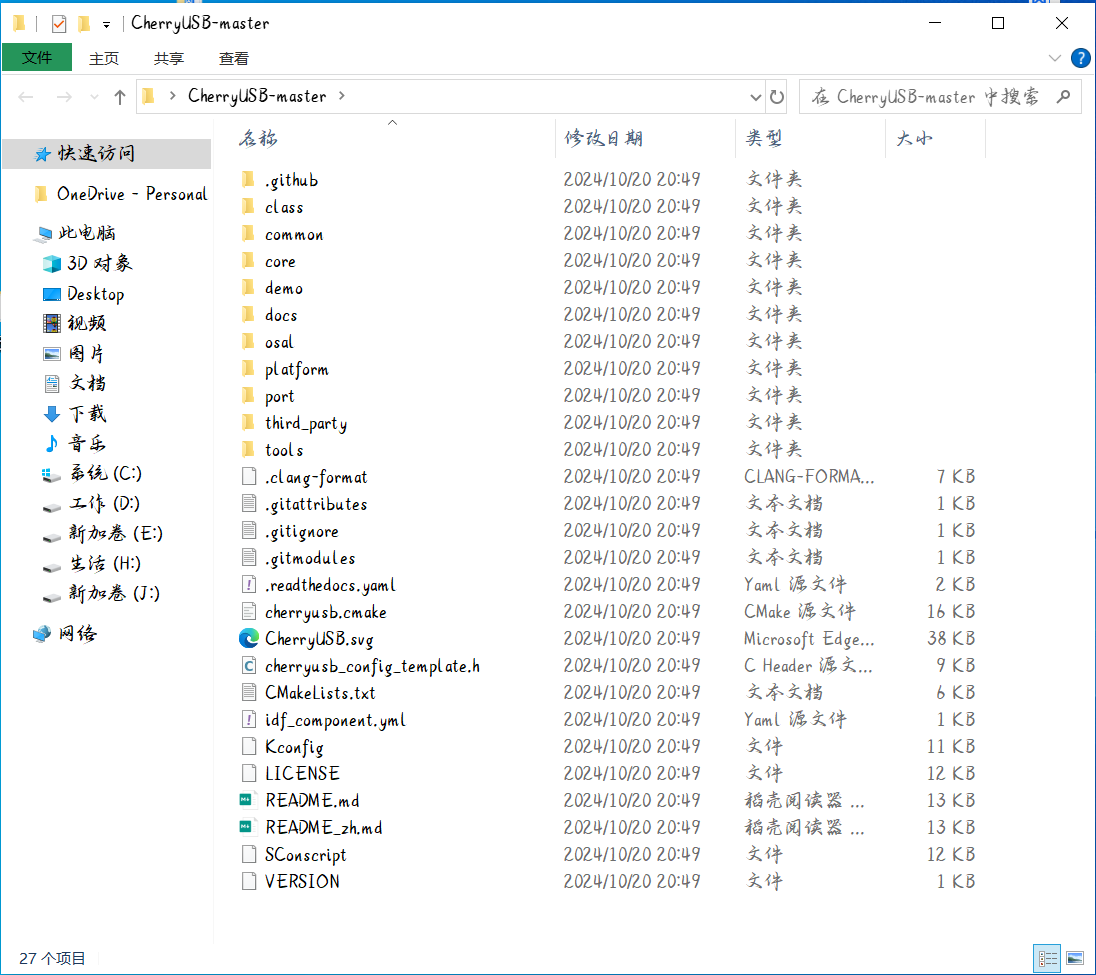


这就需要在系统时钟初始化时将PLLAR设置为48MHZ，如下图

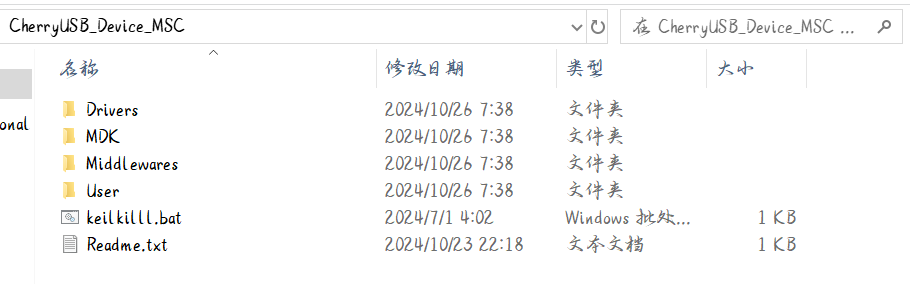


一定要注意配置好USB时钟为48MHZ！

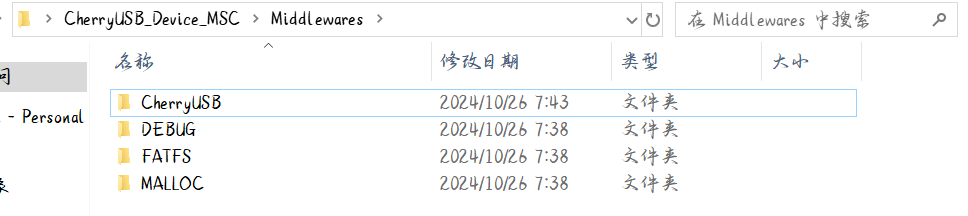
1. 准备好Cherry USB源码（<https://github.com/sakumisu/CherryUSB>）,24/10/26获取的V1.4.1版本



1. 将准备好的工程名字改为CherryUSB\_Device\_MSC



1. 在Middlewares中间件文件夹中新建CherryUSB文件夹



在CherryUSB文件夹中新建app、core、class、port、common五个文件夹



其中

app存放usb应用层文件

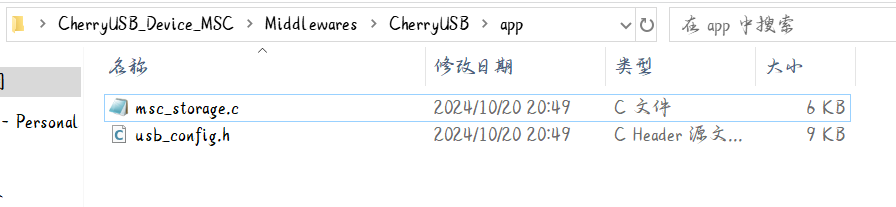
core文件夹中存放CherryUSB主从协议栈核心实现

class文件夹用来存放各种USB设备和主机类

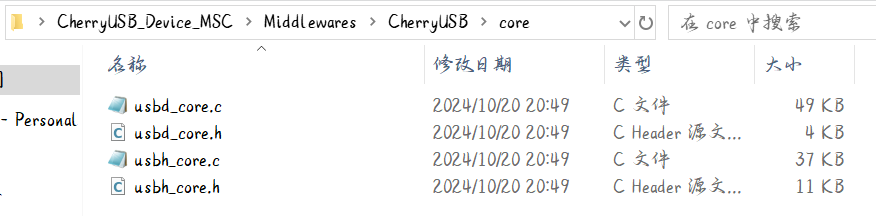
port文件夹存放接口文件，用来对接CherryUSB的core代码

common文件夹存放usb spec 定义、常用宏、标准接口定义

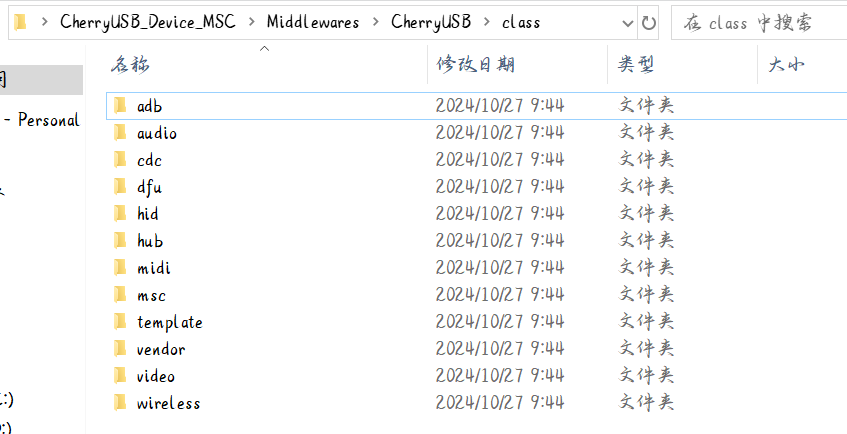
1. 从CherryUSB源码文件夹中拷贝需要的文件到工程文件夹中Middlewares下的Cherry USB中
2. 复制CherryUSB源码文件夹下的cherryusb\_config\_template.h到Middlewares/CherryUSB/app文件夹中并改名为usb\_config.h ，此文件用来配置USB相关功能；同时复制CHerryUSB/demo文件夹下的msc\_storage\_template.c到app文件夹中，并改名为msc\_storage.c ，此文件是CHerryUSB提供的实现MSC功能的模板，我们就参考这个文件来实现MSC功能，此文件中主要是实现对存储介质的读写。



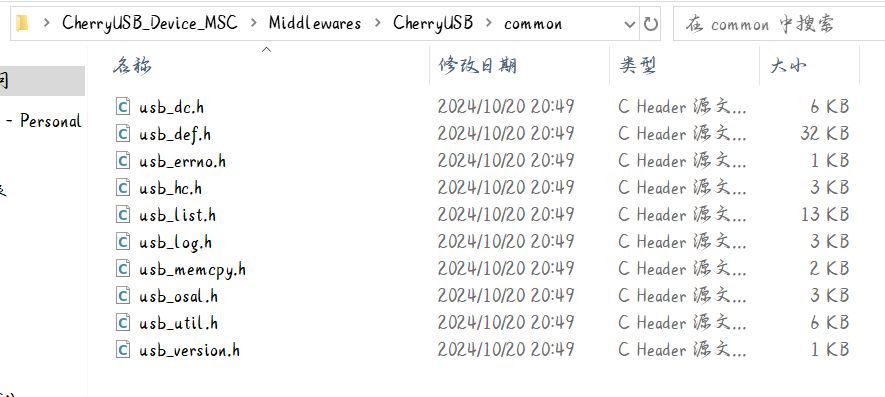
1. 复制CherryUSB源码文件夹下的core文件夹中的所有文件到Middlewares/CherryUSB/core文件夹中，其中usbd\_core.c和.h为Device core文件；usbh\_core.c和.h为Host core文件，我们这里是实现Device MSC功能，所以稍后只需要将usbd\_core.c和.h添加到我们的工程中。



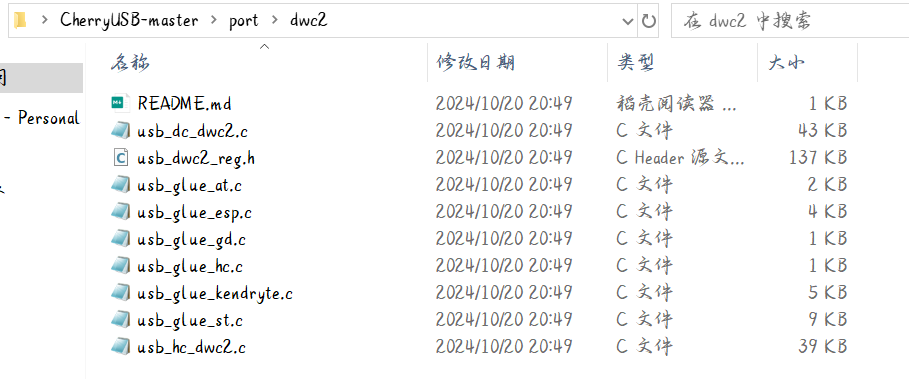
1. 复制CherryUSB源码文件夹下的class文件夹中的所有文件夹到Middlewares/CherryUSB/class文件夹中，我们这里只实现MSC功能，所以稍后只需要将class文件夹下MSC文件夹中的文件添加到工程中。



1. 复制CherryUSB源码文件夹下的common文件夹中的所有文件到Middlewares/CherryUSB/common文件夹中



1. 我们使用的是HC32F4A0PITB单片机，此单片机对接CherryUSB的接口文件已经由CherryUSB的作者帮我们实现，在CherryUSB源码文件夹中port文件夹下，是各种不同USB IP对应的接口文件夹，每个文件夹中都有说明此IP文件夹支持的单片机。HC32F4A0PITB的接口文件夹夹为dwc2 ，dwc2文件夹下的文件如下图



其中

README.md为说明文件，说明使用dwc2 IP核的单片机有哪些

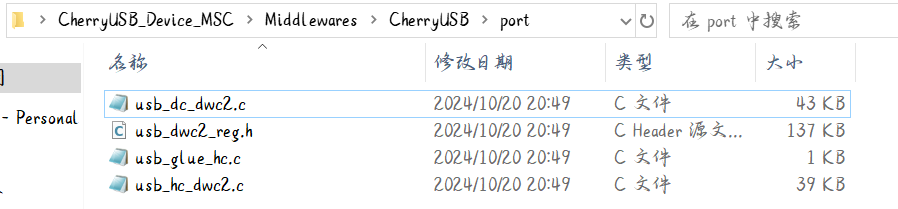
usb\_dc\_dwc2.c为Device dwc2接口源码

usb\_hc\_dwc2.c为Host dwc2接口源码

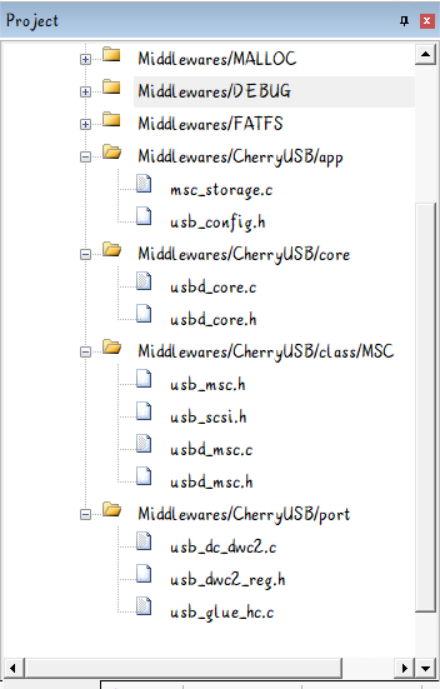
usb\_dwc2\_reg.h为dwc2 IP核的USB相关寄存器描述文件（会发现其与hc32\_ll\_usb.h中描述的一样）

usb\_glue\_xxx.c为胶水文件，用来连接 dwc2 IP核的不同单片机 与 usb\_dc\_dwc2.c或usb\_hc\_dwc2.c ，我们这里用的hc单片机所以需要的是usb\_glue\_hc.c ，我们需要在里面实现usbd\_dwc2\_delay\_ms()这个函数

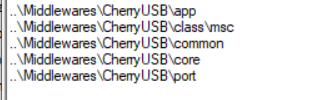
我们将需要的文件复制到Middlewares/CherryUSB/port文件夹中



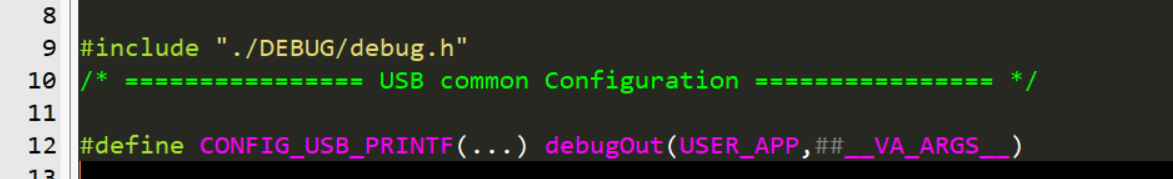
1. 打开MDK工程，将需要的文件添加到工程中



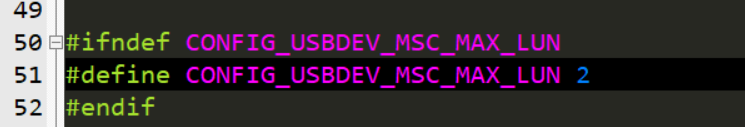
添加相关的头文件路径



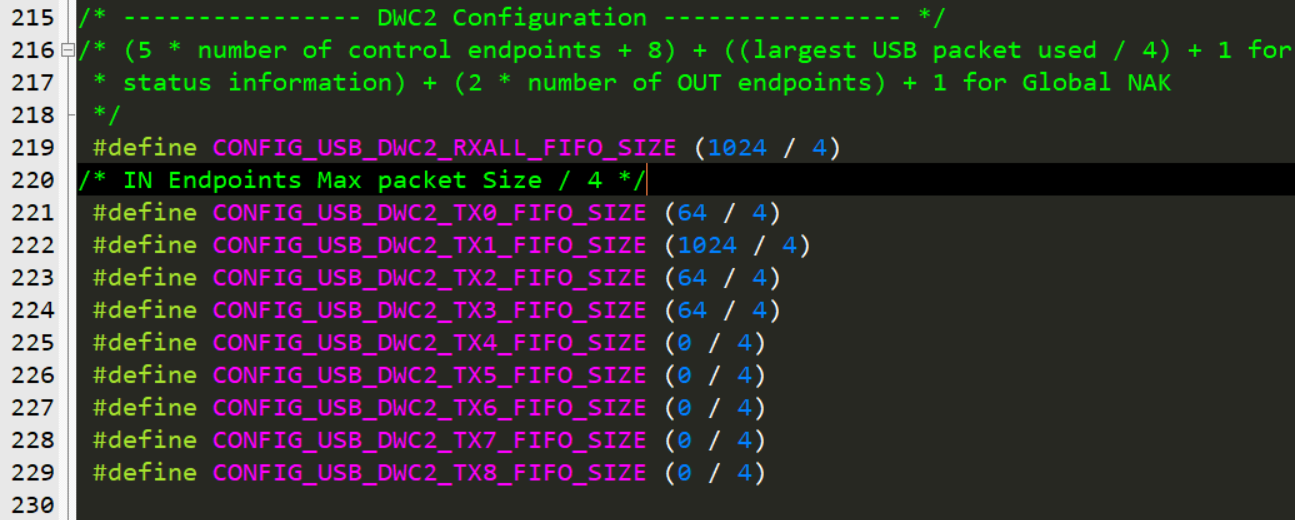
1. 修改usb\_config.h中调试输出宏定义



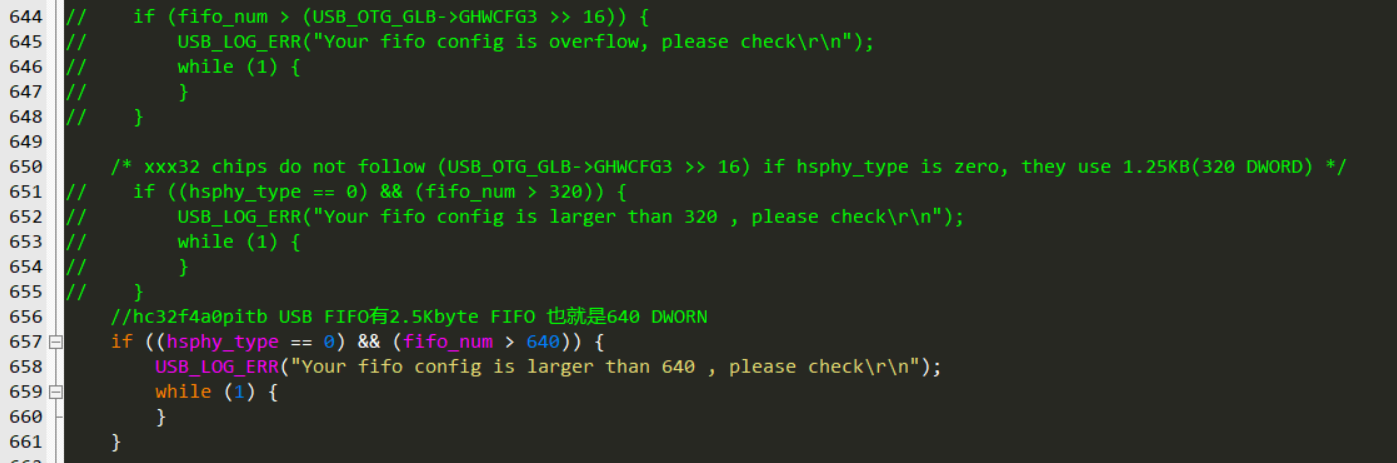
因为我们要实现枚举出两个U盘，一个是SD一个是W25Qxx，所以我们需要修改CONFIG\_USBDEV\_MSC\_MAX\_LUN为2



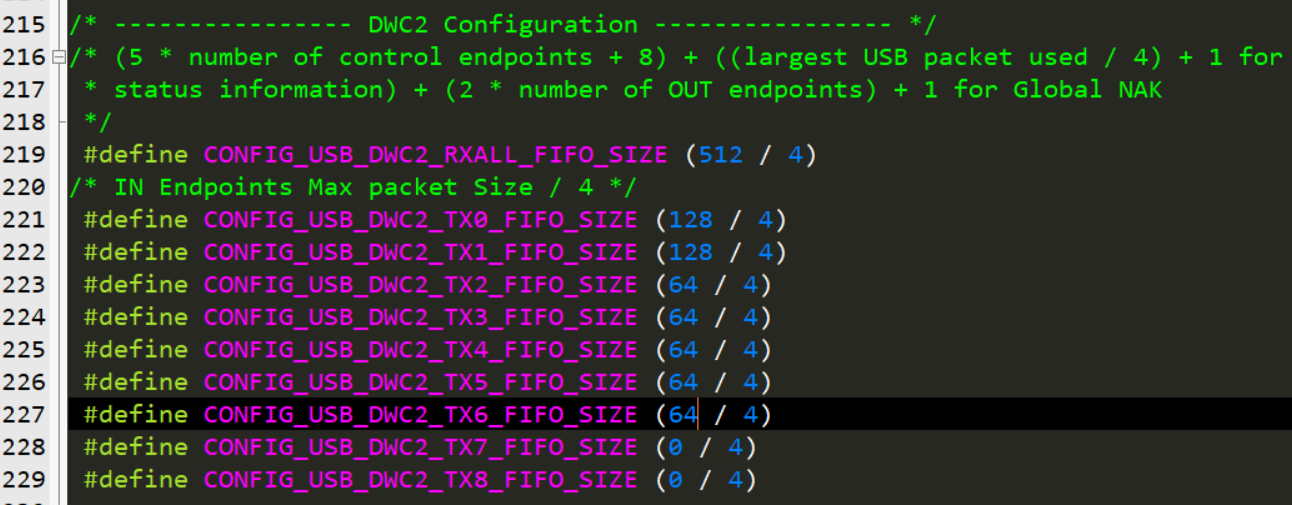
取消下面这些宏定义，这样就不要在usb\_dc\_dwc2.c中修改这些宏了



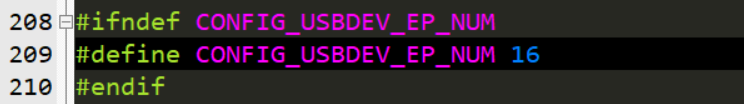
关于这些宏，是用来分配USB FIFO的，HC32F4A0PITB共2.5KByte的USB FIFO，但是此工程中如果配置使用的FIFO超过1.25KByte也就是320 DWORN CherryUSB就会报错，可以通过修改usb\_dc\_dwc2.c中的代码来避免这个限制，如下图



经测试发现，在MSC功能中，FIFO容量增加对读写速度影响不大，所以本次就不修改CherryUSB源码，修改FIFO大小为320 DWORN,如下图

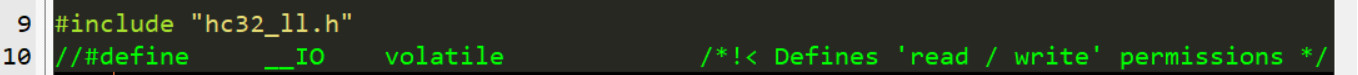


HC32F4A0PITB共16个ep端点，修改如下宏为16



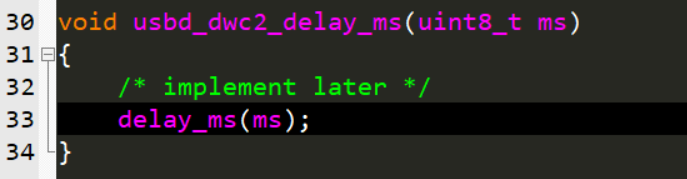
至此usb\_config.h就修改完成

1. 在usb\_dwc2\_reg.h中加入头文件#include ”hc32\_ll.h” ，注释掉\_\_IO的定义

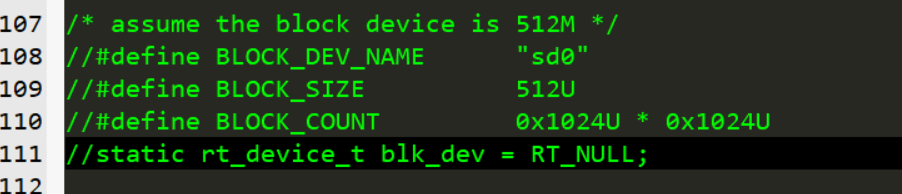


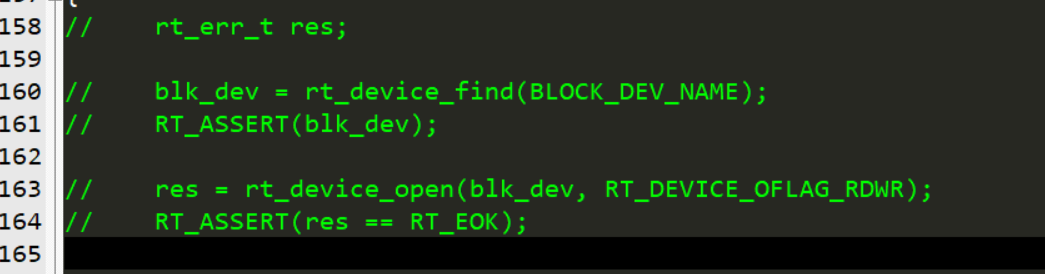
1. 在usb\_glue\_hc.c中包含#include "delay\_systick.h"，并在usbd\_dwc2\_delay\_ms()函数中实现描述ms延时





1. 删除usb\_storage.c中的宏定义判断 #ifdef \_\_RT\_THREAD\_H\_\_，同时注释以下内容



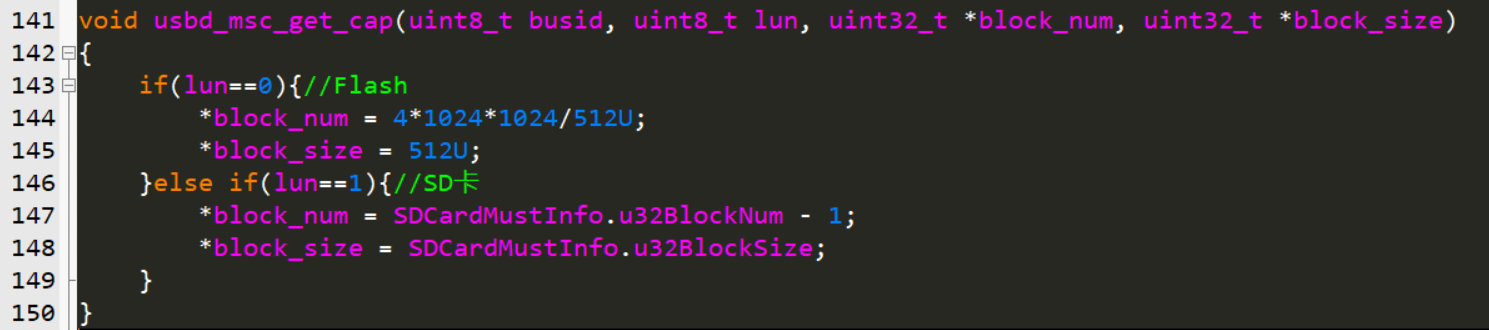


在usb\_storage.c中包含这两个头文件

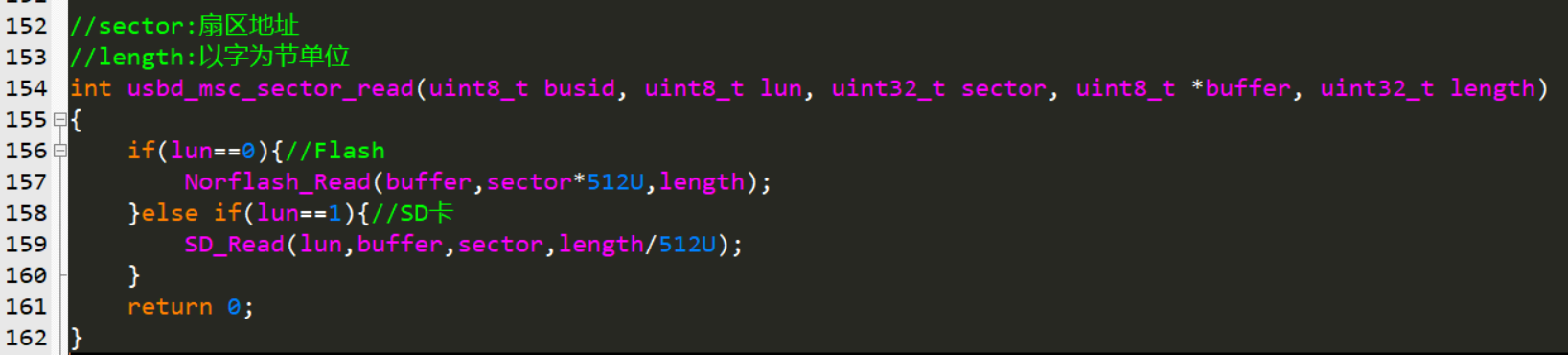
#include "w25qxx.h"

#include "sd\_diskio.h"

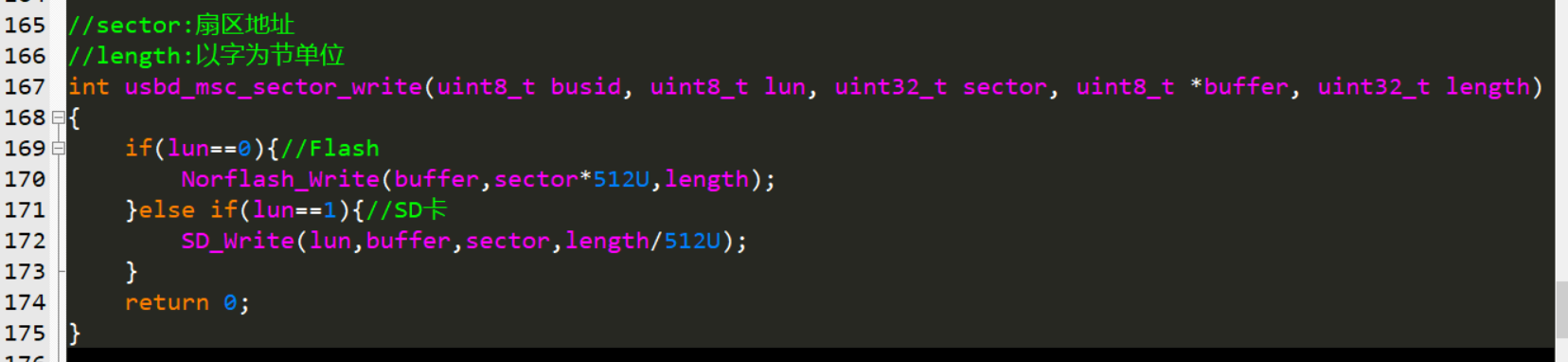
修改usbd\_msc\_get\_cap函数



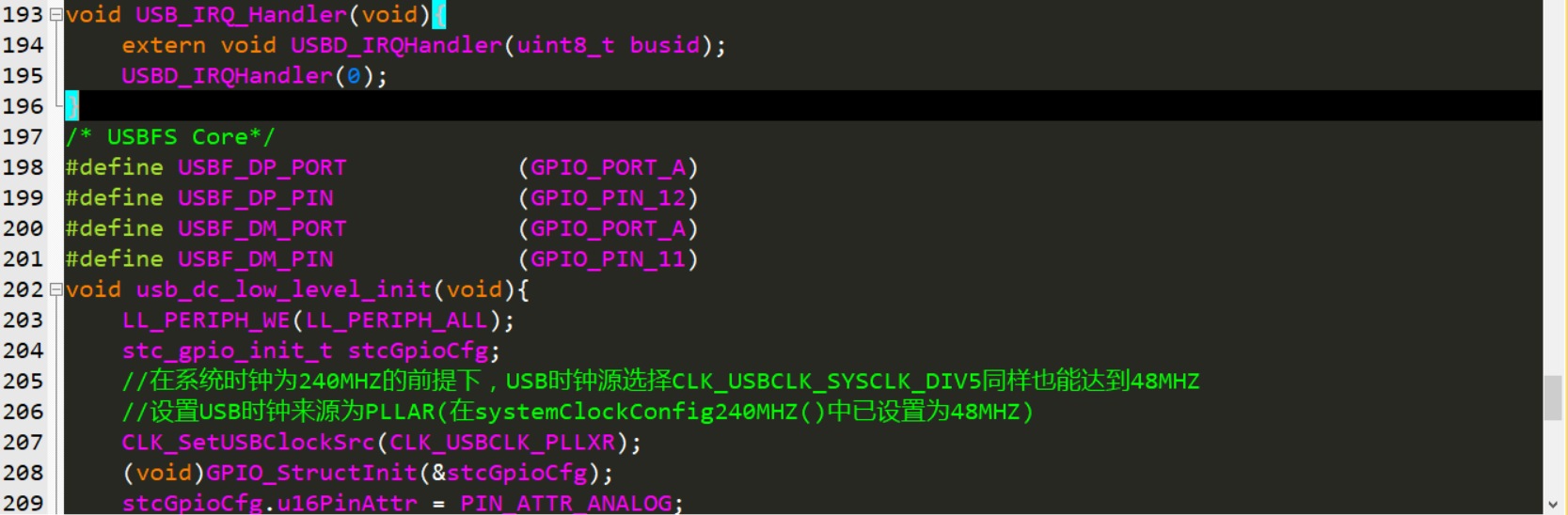
修改usbd\_msc\_sector\_read函数

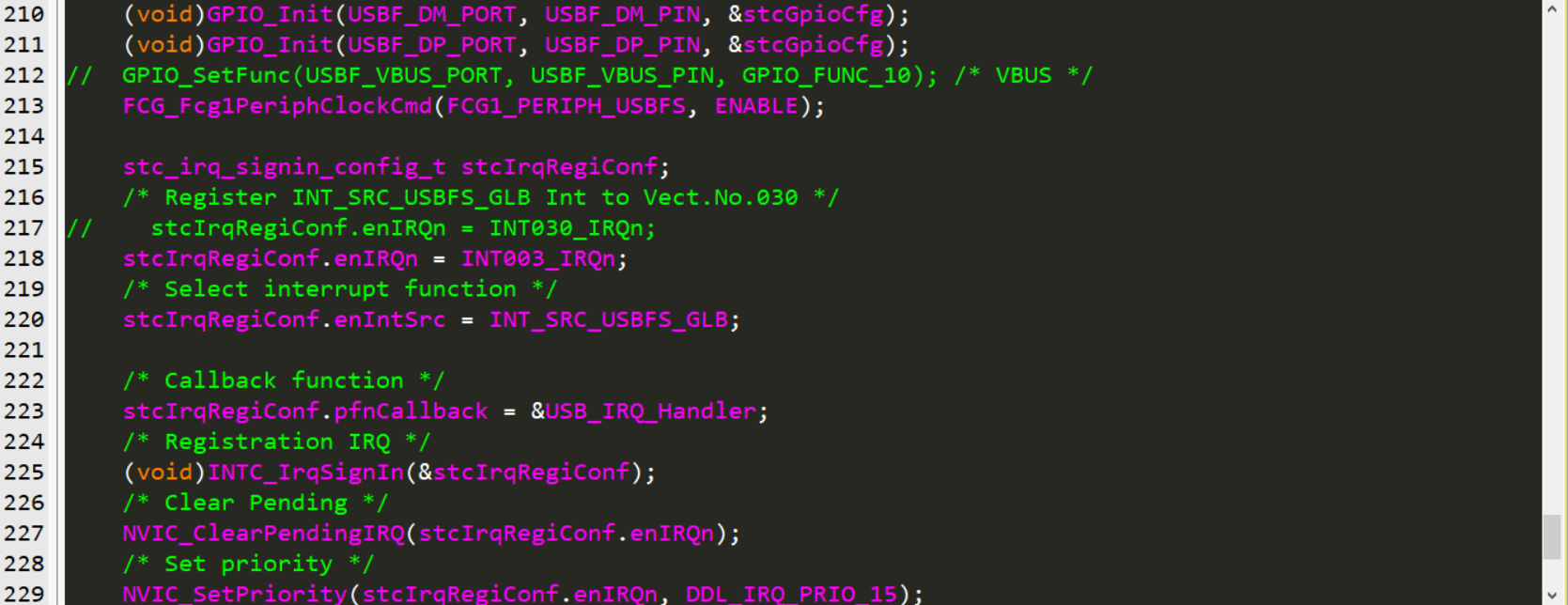


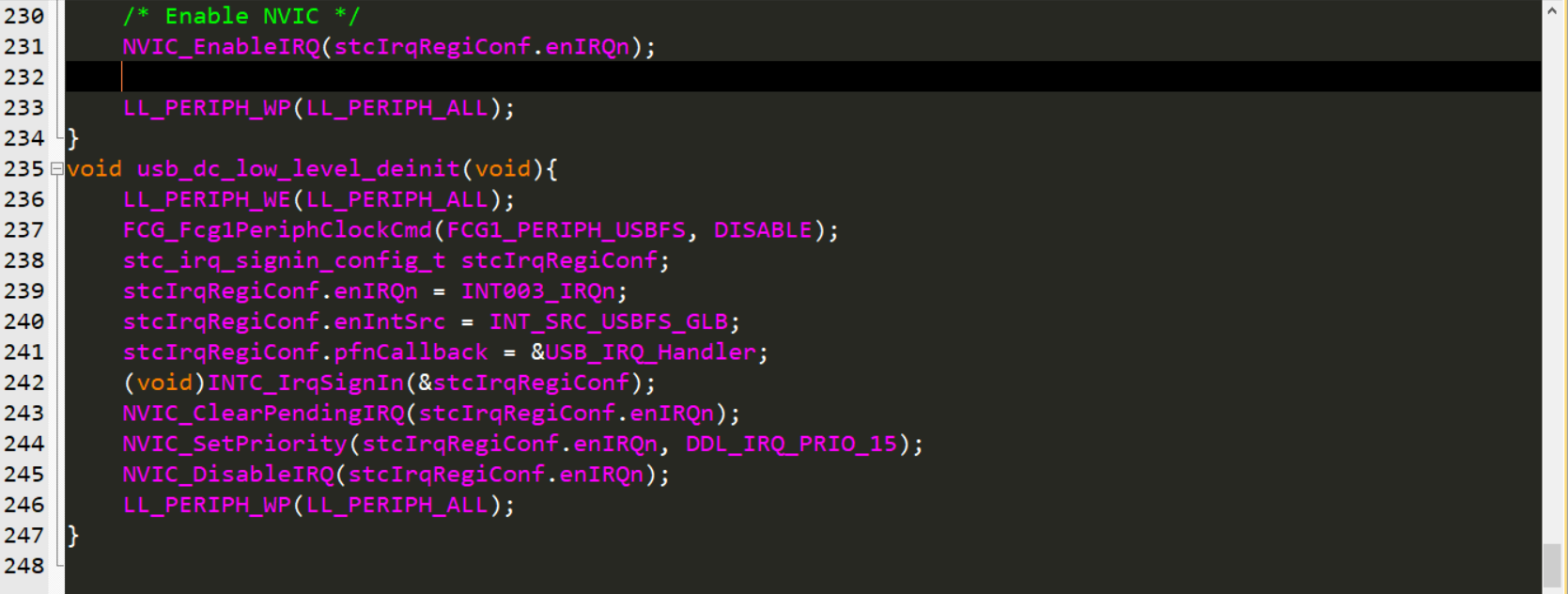
修改usbd\_msc\_sector\_write函数



在usb\_storage.c中加入下面内容

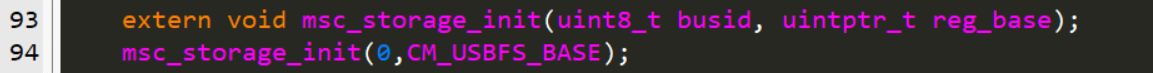






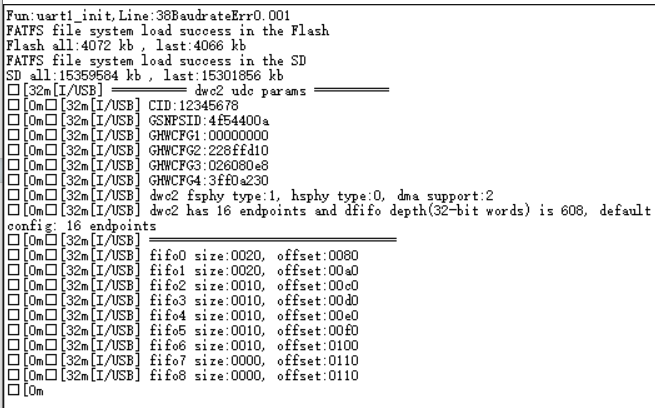
usb\_dc\_low\_level\_init 函数主要负责 USB 时钟、引脚、中断的初始化，该函数可以放在你想要放的任何参与编译的 c 文件中，这里就放在usb\_storage.c中。如何进行 USB 的时钟、引脚、中断等初始化，请自行根据你使用的芯片原厂提供的源码中进行添加。

在main中调用以下函数就能使用了

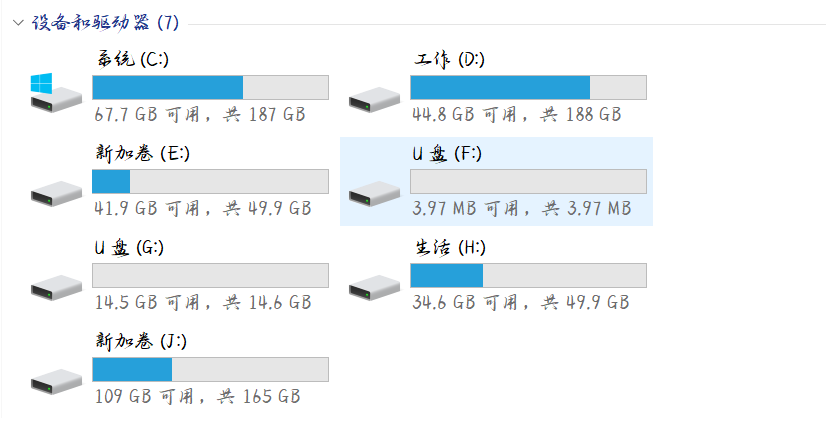


关于以下三个函数中usbd\_msc\_sector\_write、usbd\_msc\_sector\_read、usbd\_msc\_get\_cap

为什么lun=0对应FLash，lun=1对应SD，其实这个没有明确规定，只需要三个函数中都是lun=0对应FLash，lun=1对应SD就行，反过来也是一样可以的，Flash对应lun=0只是意味着电脑时先枚举出Flash的U盘。程序运行后，串口输出如下



电脑会弹出两个U盘



如果串口输出CherryUSB以下报错信息



大概率是usbd\_msc\_get\_cap函数中block\_num和block\_size搞反了

1. A
2. A
3. A