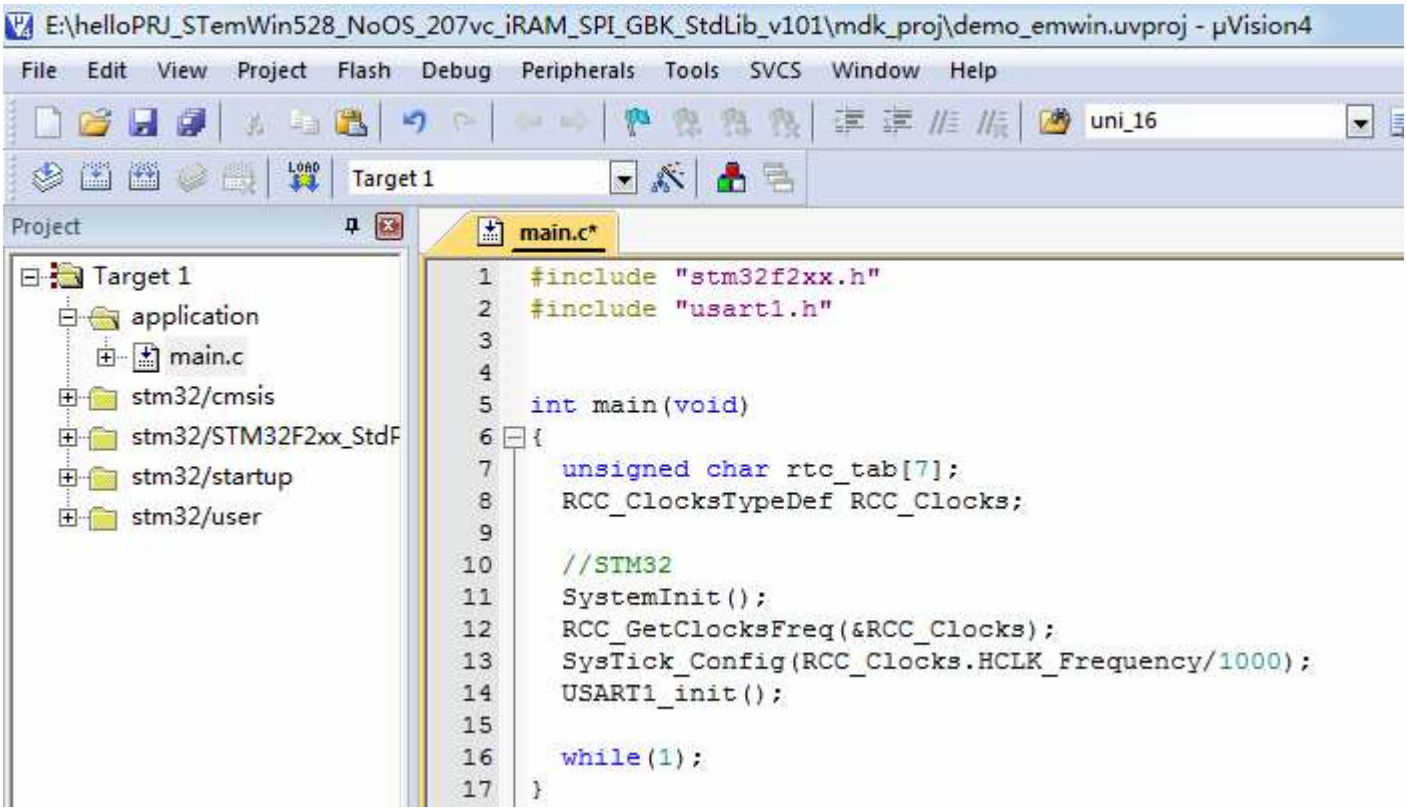
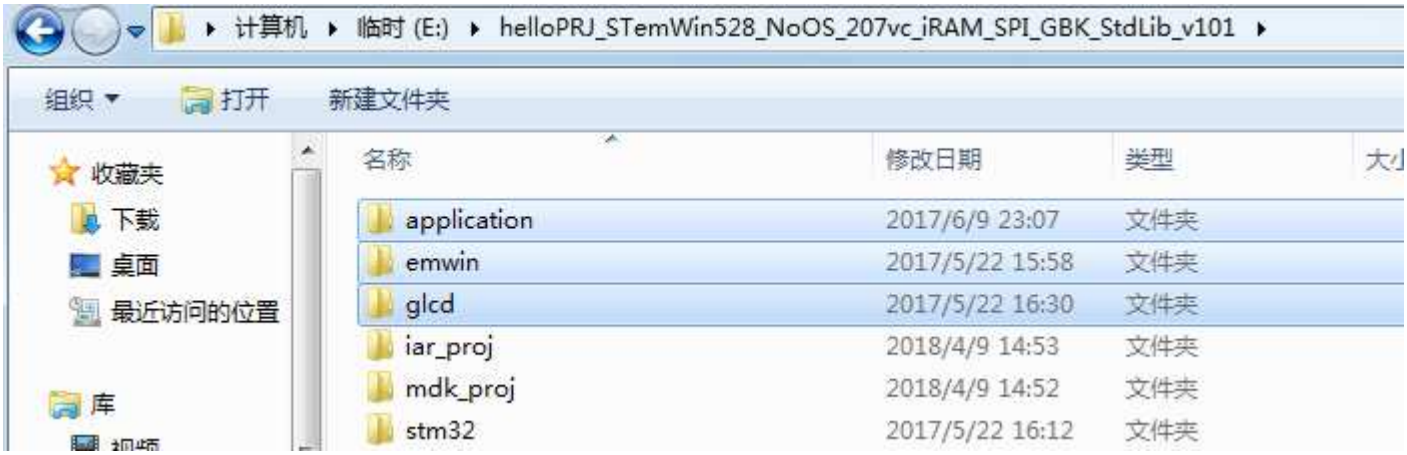
demo 工程中的 emWin 移植到其他 MDK 工程

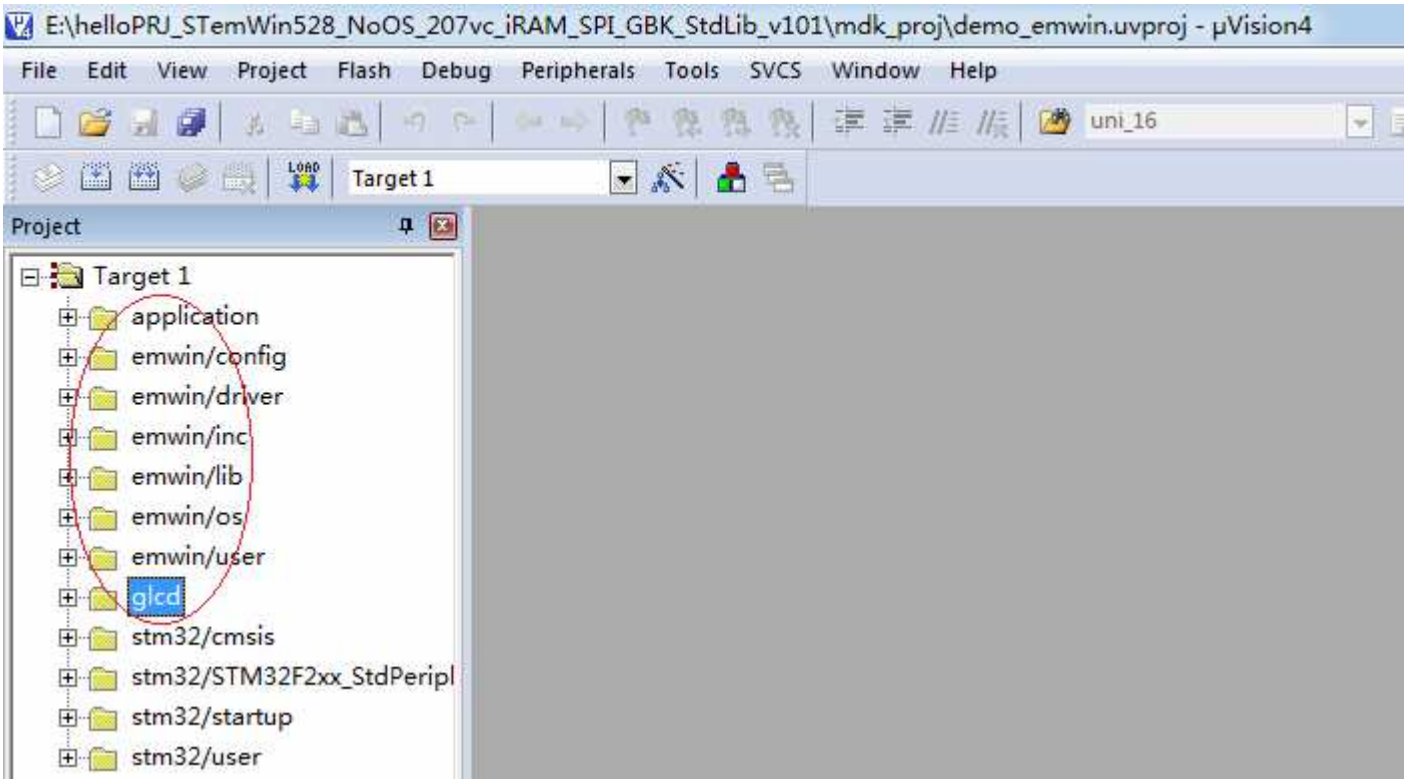
Porting emwin of demo project to other MDK project

1.准备一个可以正常运行的 MDK 工程(开发板/评估板都会提供) Prepare a working MDK project (both development and evaluation boards will be available)

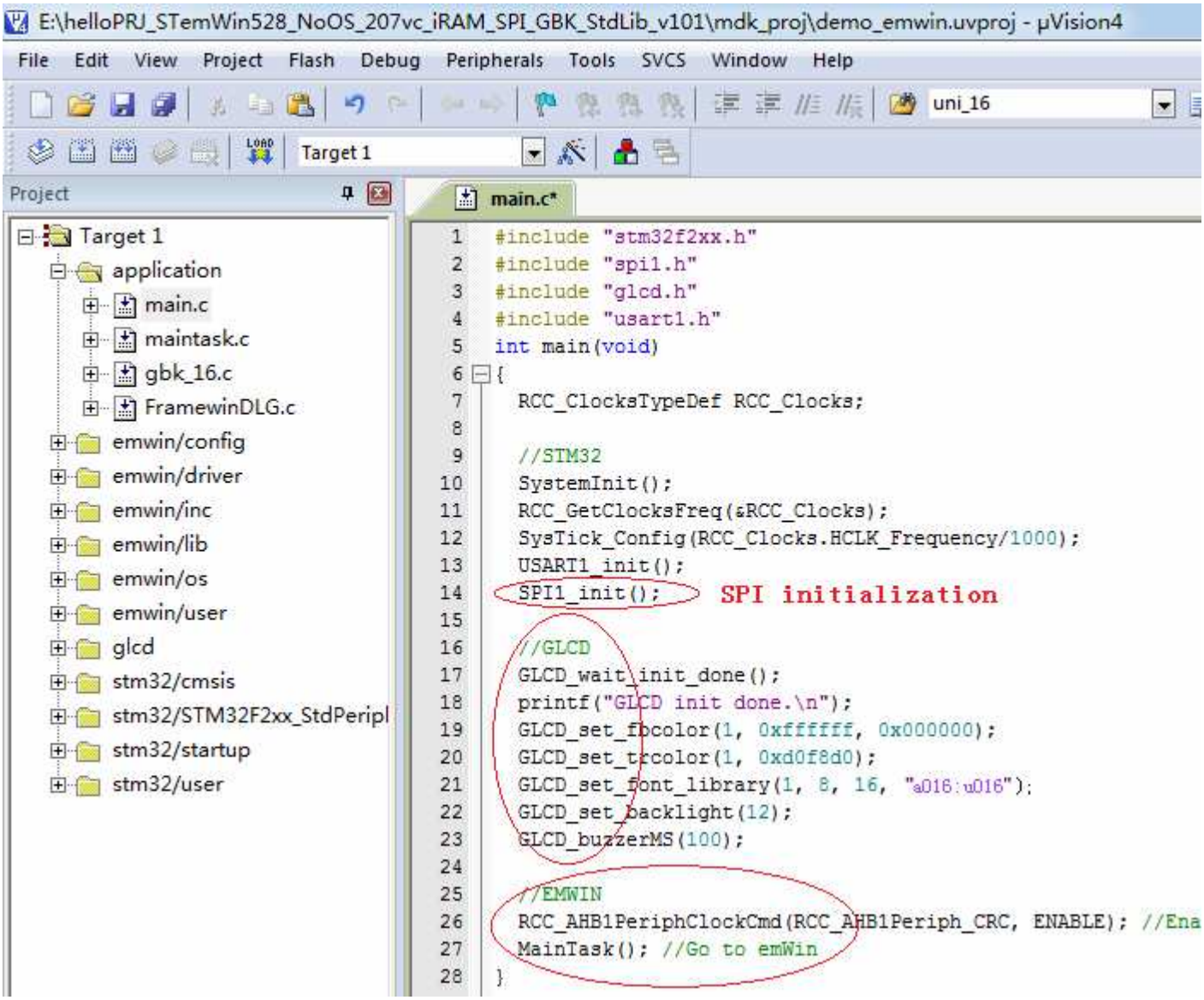


2.拷贝并添加相关文件到 MDK 工程(除了 main.c) Copy and add related files to the MDK project (except main.c):



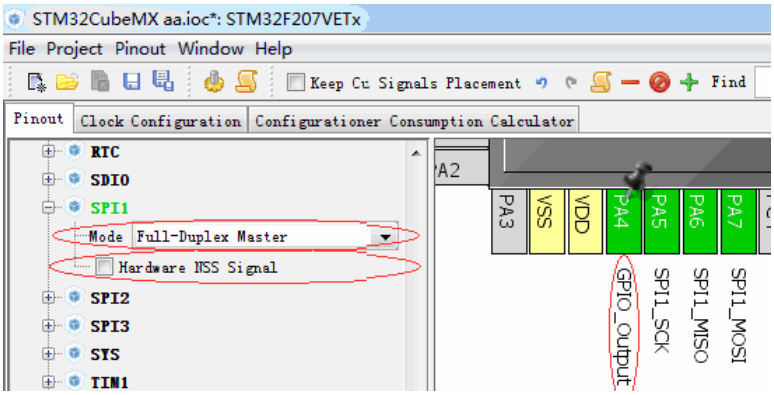

E:\helloPRJ_STemWin528_NoOS_207vc_iRAM_SPI_GBK_StdLib_v101\mdk_proj\demo_emwin.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
uni_16
Target 1
Project
Target 1
application
emwin/config
emwin/driver
emwin/inc
emwin/lib
emwin/os
emwin/user
glcd
stm32/cmsis
stm32/STM32F2xx_StdPeripl
stm32/startup
stm32/user

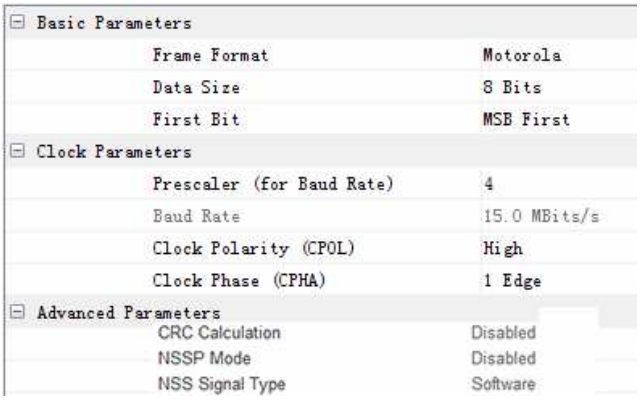
3.修改 main.c Modify main.c：

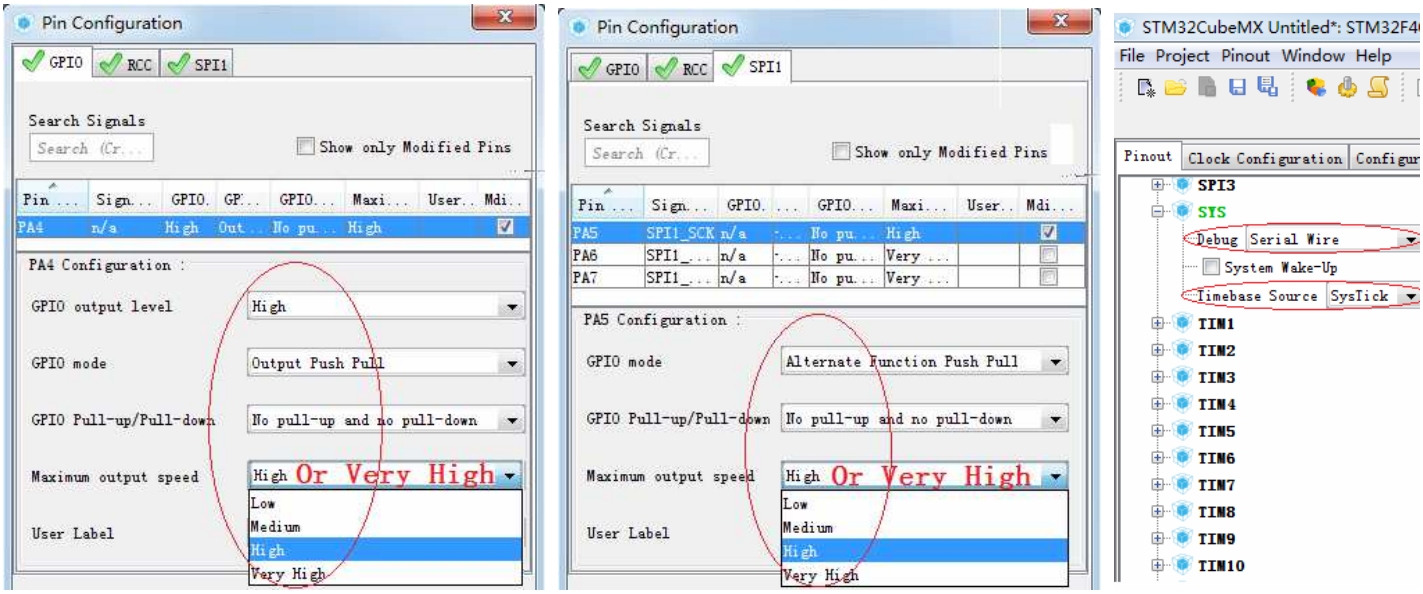


4.初始化 SPI 和 GPIO(NSS) SPI and GPIO(NSS) initialization

STM32CubeMX 生成 SPI 和 GPIO 初始化的设置 Setting for STM32CubeMX to generate SPI and GPIO initialization:







备注[1]：使用 STM32CubeMX 生成的 SPI 和 GPIO 初始化代码一般不会有问题，只是读写函数最好先用库的，测试正常工作之后再去优化以提升读写速度.

备注[2]：SPI 初始化之后，一定要记得使能 SPI，默认是关闭的，比如例程中的：SPI_Cmd(SPI1, ENABLE)，使能函数可在 HAL 库中找到.

备注[3]：无论是 SPI（特殊功能）引脚还是 NSS（GPIO）引脚，都需要设置成高速，否则翻转速度会非常慢，并且 NSS（GPIO）引脚初始化电平应设置为高（GPIO output level = High）.

备注[4]：一定要记得开启 Debug 接口，否则第一次下载成功之后将无法再下载程序. 如果发生这样的事情，可以将 BOOT1 引脚接高电平，然后通过 SWD 模式下载一次正确的程序即可.

NOTE[1]: The SPI and GPIO initialization code generated by STM32CubeMX is generally no problem. But read/write function is better to use libraries' s first, After being tested to work properly, it is optimized to improve the speed of reading and writing.

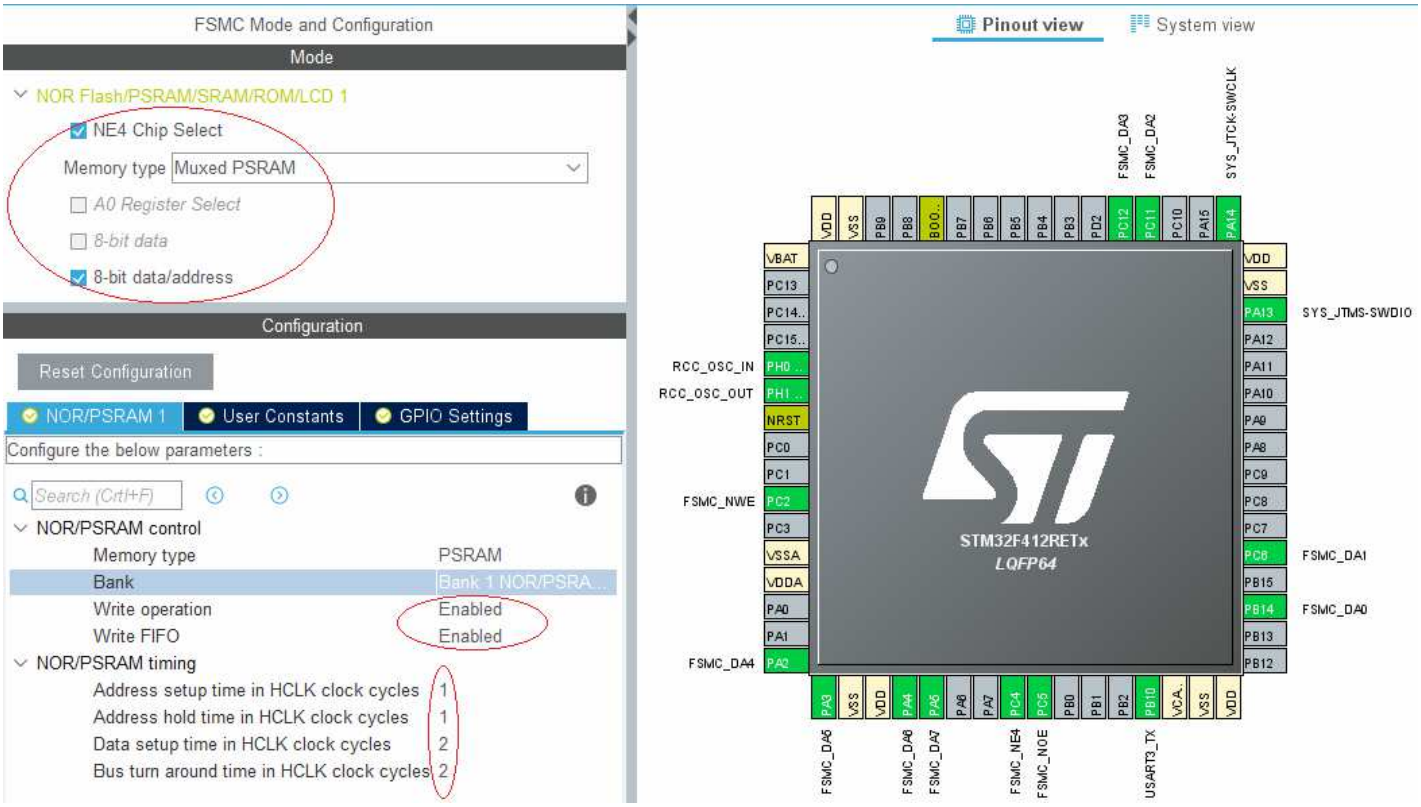
NOTE[2]: After the SPI is initialized, it must be enabled manually. It is disable by default. For example (in demo code):SPI_Cmd(SPI1, ENABLE). Enable function can be found in the HAL Library.

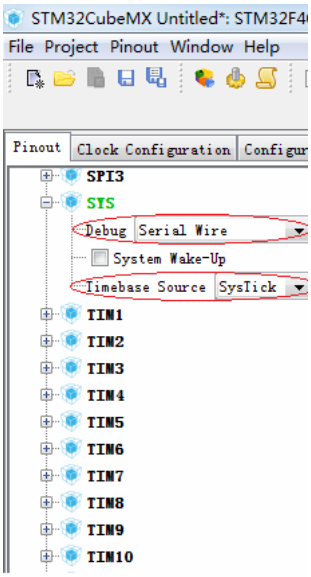
NOTE[3]: Whether SPI (alternate functions) pins or NSS (GPIO) pin, they need to be set to fast/high speed, otherwise the turnover speed will be very slow. And the initial level of NSS(GPIO) pin should be set to high(GPIO output level = High).

NOTE[4]: Be sure to open the Debug interface, otherwise you will not be able to download the program after the first successful download. If this happens, you can connect the BOOT1 pin to a high level and download the correct program once through SWD mode.

5.如果使用 8080 并口模式,初始化 FSMC FSMC Initialization If Using 8080 Interface Mode:

STM32CubeMX 生成 FSMC 初始化的设置 Setting for STM32CubeMX to generate FSMC initialization:





备注[1]: 设置中的时序参数可以先用默认较大值,等接口调通以后再通过测试方法逐个减小到最小值;方法是:(1)Bus turn around time 先设置为默认较大的值,其他参数逐个改变并测试直到最小值,注意同时测试触摸.(2)最后再调整 Bus turn around time 参数到最小值,注意同时测试触摸,因为和读有关.(3)Data 相关参数和 Bus turn around time 参数最好最好最好多加 1,因为处于临界状态时偶尔会漏读触摸而不易察觉!

备注[2]: 一定要记得开启 Debug 接口,否则第一次下载成功之后将无法再下载程序.如果发生这样的事情,可以将 BOOT1 引脚接高电平,然后通过 SWD 模式下载一次正确的程序即可.

备注[3]: 有些 MCU 的地址锁存脚 NAL 没有自动分配,比如 f412 的 PB7-FSMC_NL 引脚,需要在程序中手动配置引脚功能,比如在 f412 demo 例程中,在"stm32f4xx_hal_msp.c"添加了 PB7 的初始化配置代码.另外,Demo 板有些引脚是连在一起的,请勿同时启用这些引脚以免冲突.

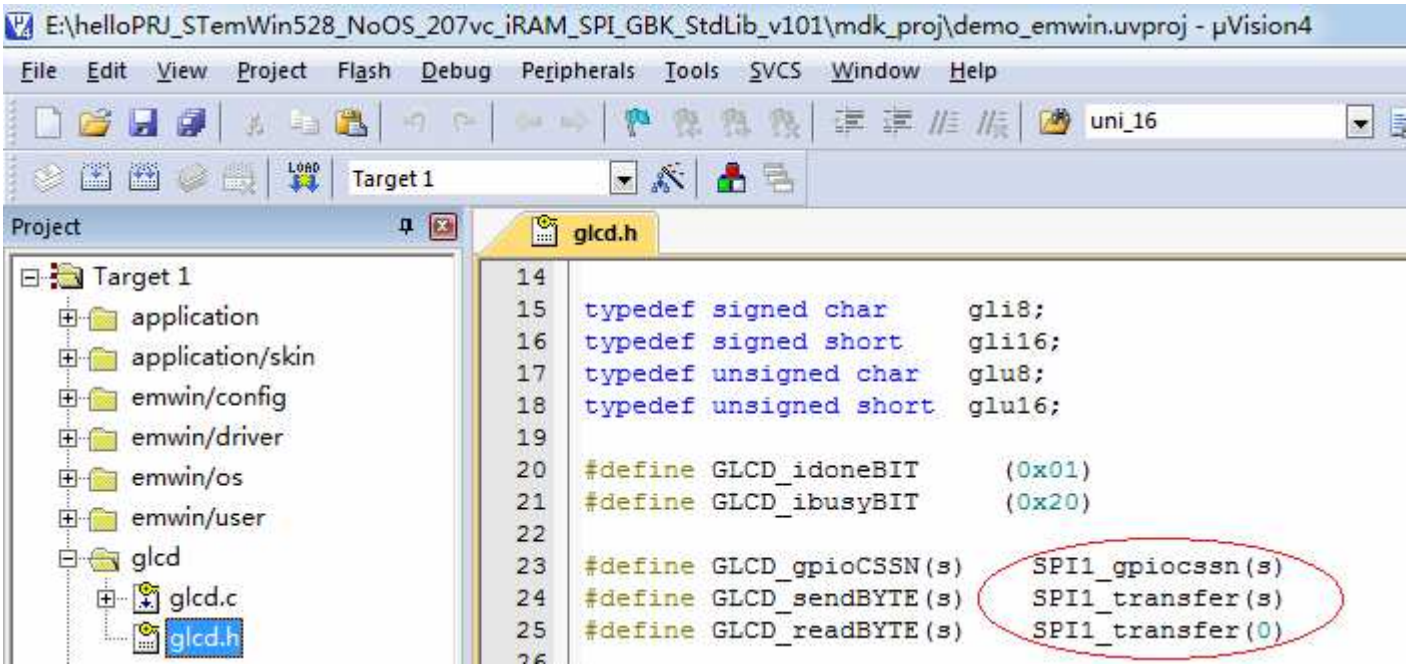
NOTE[1]: The timming parameter value can be the default or larger value first, After the interface debugging works normally, they can be reduced to the minimum value one by one through the actual test method. The methods is as follows: (1)bus turn around time is set to the default or larger value, and other parameters are changed one by one and tested until the minimum value, pay attention to testing touch at the same time. (2)finally, adjust the bus turn around time parameter to the minimum value, pay attention to testing touch at the same time. Because it is related to reading. (3)Data related parameters and Bus turn around time parameters are best best best added by 1, It's hard to detect because of the occasional missing touch when it's in a critical state!

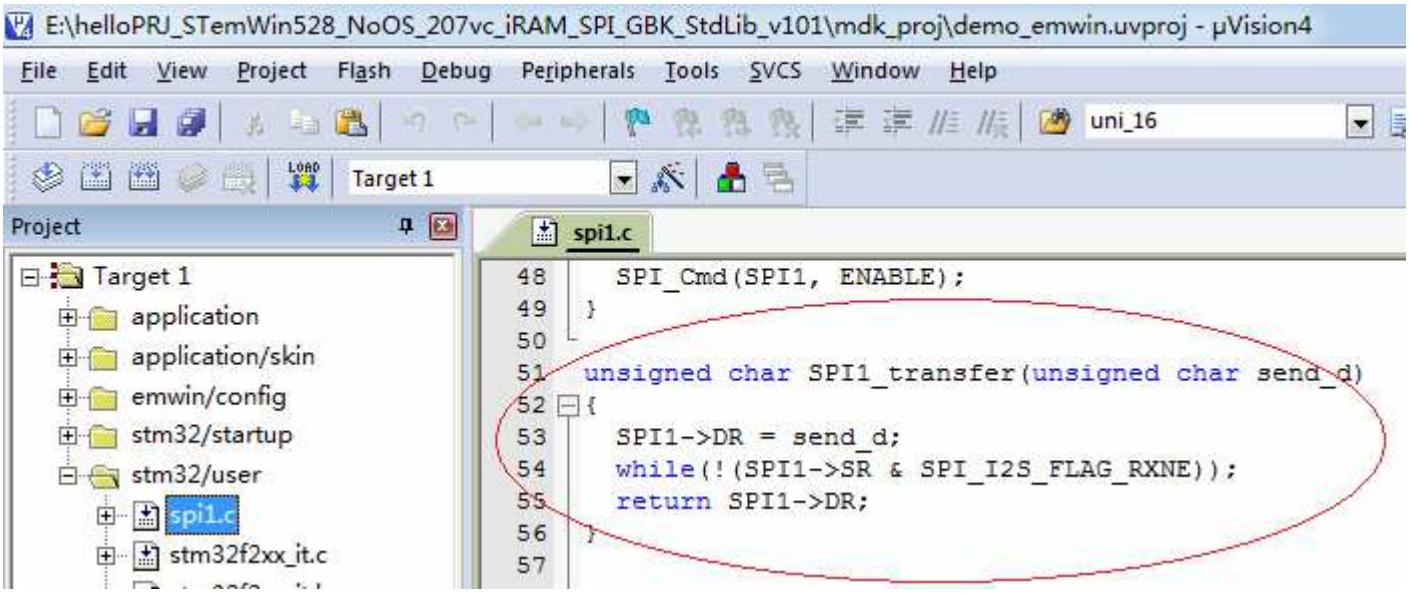
NOTE[2]: Be sure to open the Debug interface, otherwise you will not be able to download the program after

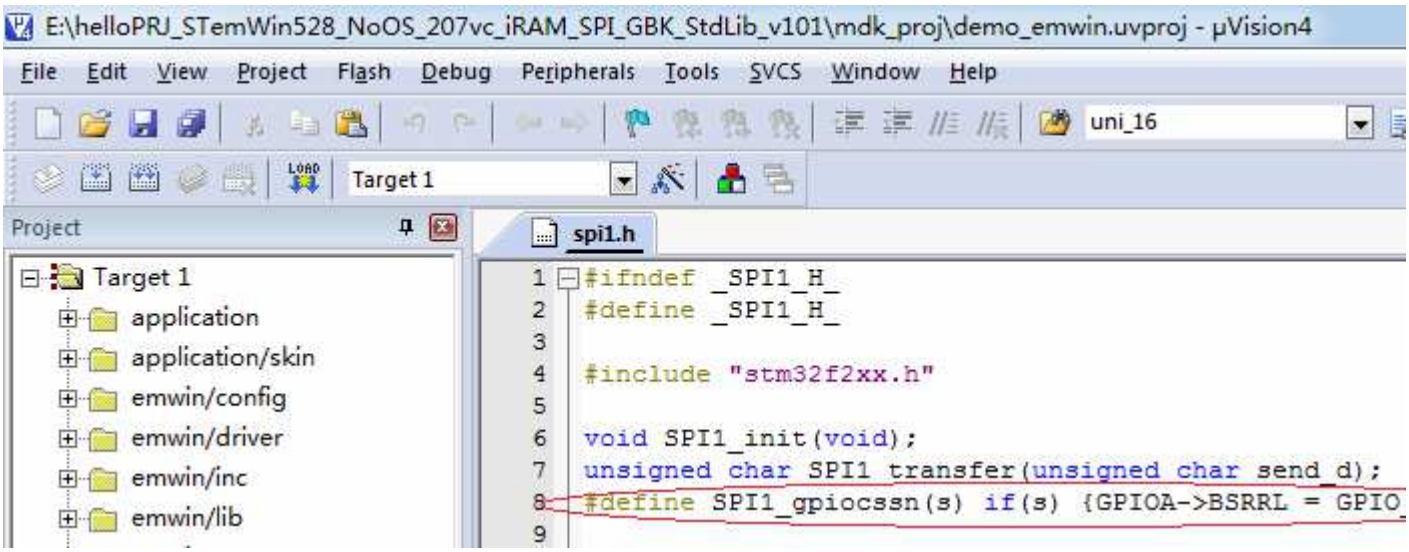
the first successful download. If this happens, you can connect the BOOT1 pin to a high level and download the correct program once through SWD mode.

NOTE[3]: Some MCU's address lock pin NAL is not automatically assigned, such as PB7-FSMC_NL pin of f412, The pin function needs to be configured manually in the program, For example, in the f412 demo code, the PB7 pin's initialization configuration code is added to "stm32f4xx_hal_msp.c". In addition, some pins of demo board are connected together, please do not enable these pins at the same time to avoid conflict.

6.实现 SPI 读写函数 Create SPI read and write functions：

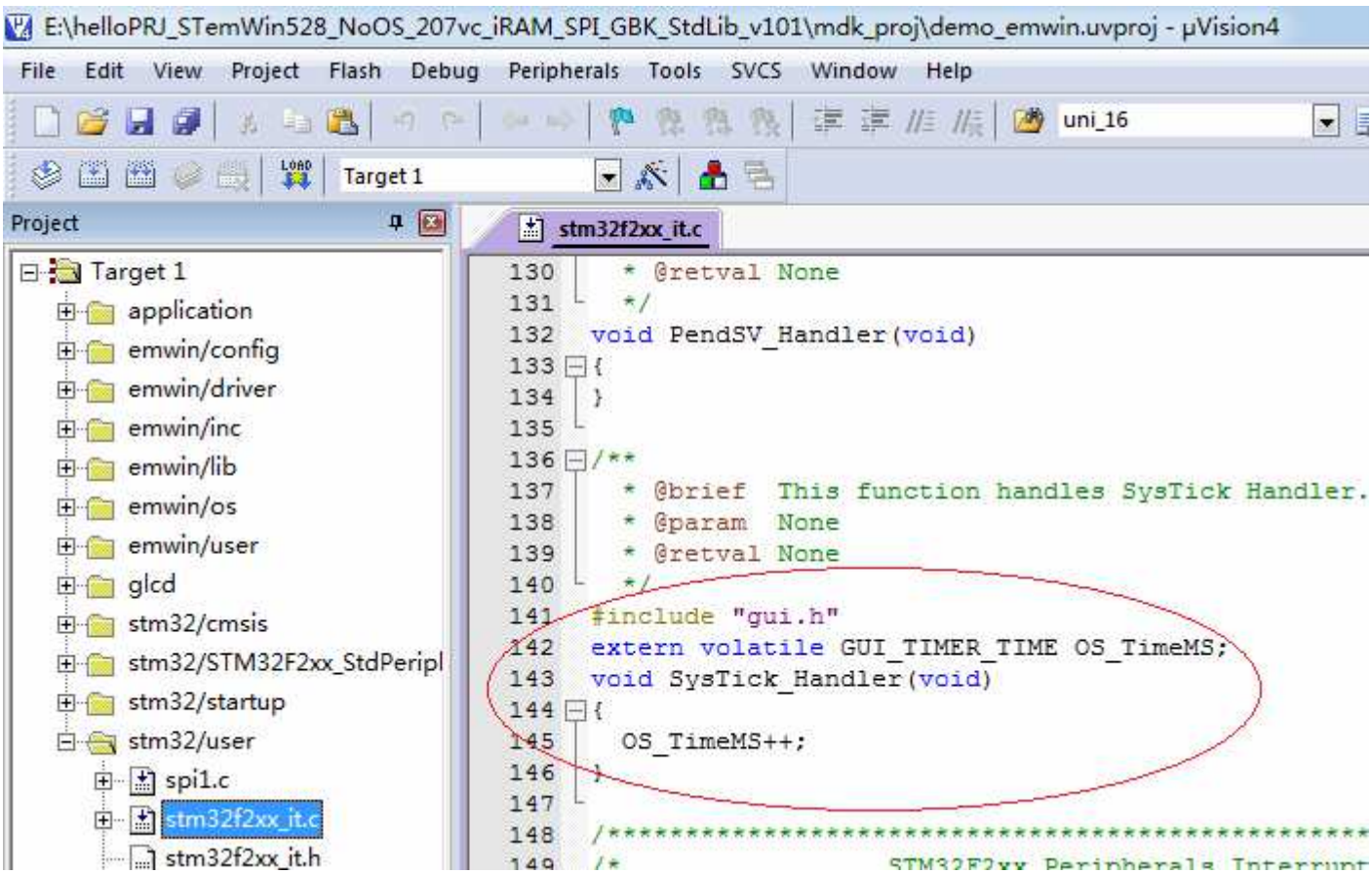






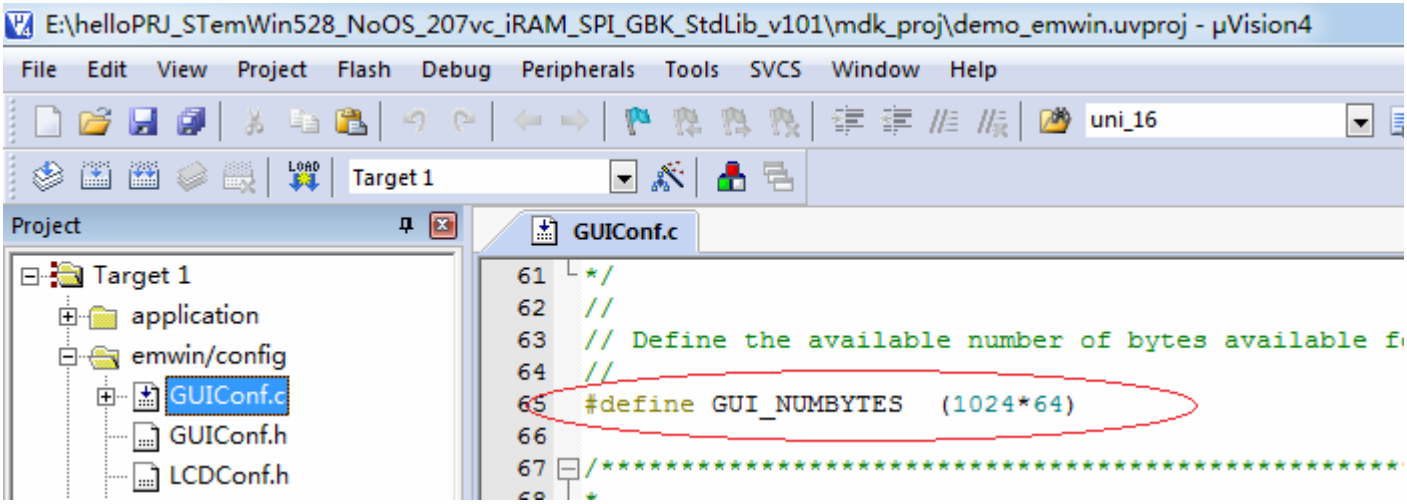
7.实现 OS_TimeMS 变量每 1ms 加 1(OS_TimeMS 是 emWin 的全局变量)，一般在系统滴答定时器中断中实现,OS_TimeMS 是 emWin 的时钟基准：

7.Implement OS_TimeMS variable increase by 1 per millisecond(OS_TimeMS is the global variable of emWin). It is generally implemented in the interruption of system tick timer.OS_TimeMS is the clock benchmark of emWin:



8.根据不同 CPU 的 RAM 大小不一样，调整 emWin 的内存分配，具体分配多大的内存才够，这和界面使用控件的情况有关，如果分配不够的话，操作界面的过程中会死机：

8. According to the different RAM sizes of different MCUs, adjust the memory allocation of emWin, and how much memory is allocated is enough. This is related to the use of widgets in the GUI interface. If the allocation RAM is not enough, Operating GUI interfaces can cause crashes:



9.需要注意的是要检查 MDK 工程的堆栈分配够不够大，否则操作界面的过程中会死机：

9. It should be noted that the stack allocation of the MDK project is not large enough, otherwise, Operating GUI interfaces can cause crashes: