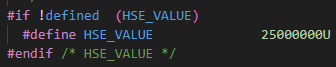
芯片是stm32f407，使用stm32cubemx生成代码，然后用vscode+platformio开发，发现调用HAL\_Delay函数延时不对，延时1s的代码感觉有3s左右。这么有规律且稳定应该就是时钟的问题。

直接上结论，platformio编译默认调用了库里自带的stm32f4xx\_hal\_conf.h头文件，这里定义的外部晶振值为25Mhz，各种配置按照这个值计算，

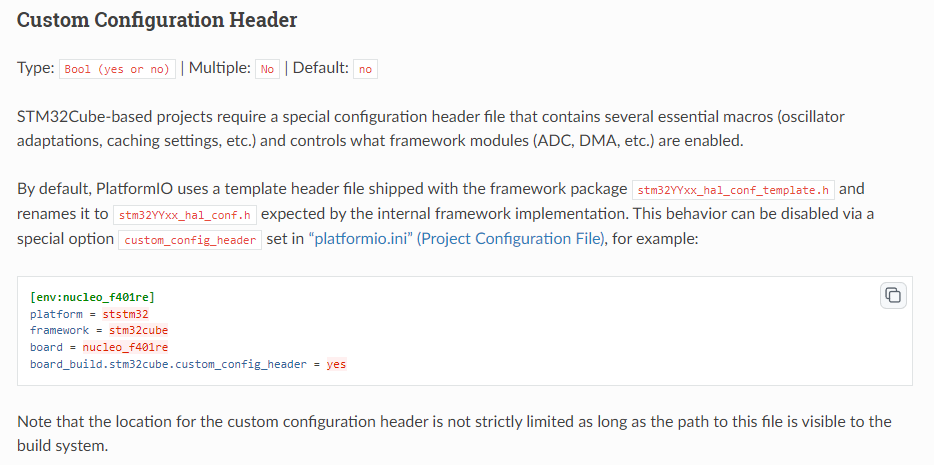


而我们板子实际是8Mhz，最终就导致延时慢了，而且25刚好是8的三倍多一点，跟我们观察到的实际情况相符。

解决办法是在“platformio.ini”文件中添加一个配置

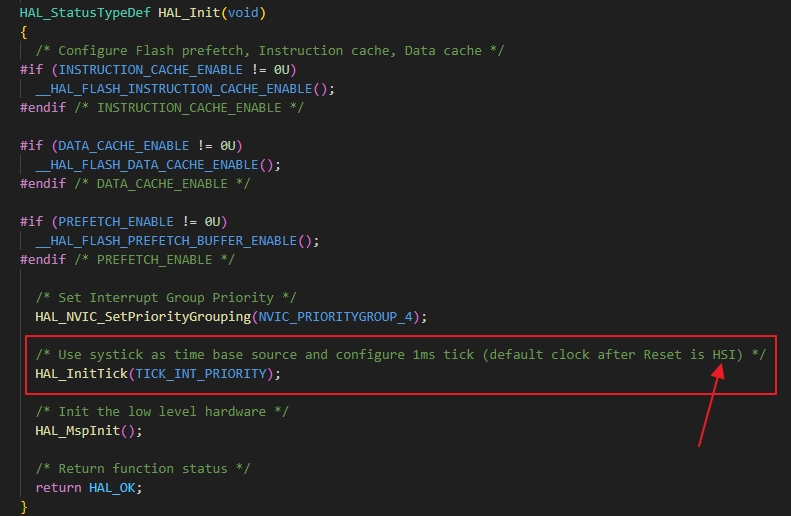
board\_build.stm32cube.custom\_config\_header = yes，这样就会使用我们用cubemx生成的stm32f4xx\_hal\_conf.h文件。这个头文件无所谓放在哪里，只要能被编译系统找到即可。

Platformio的docs文档有专门的介绍

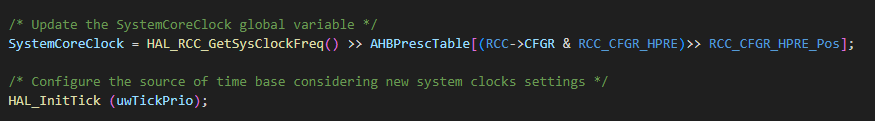


<https://docs.platformio.org/en/latest/frameworks/stm32cube.html#custom-configuration-header:~:text=%EF%83%81-,Custom%20Configuration%20Header,-%EF%83%81>

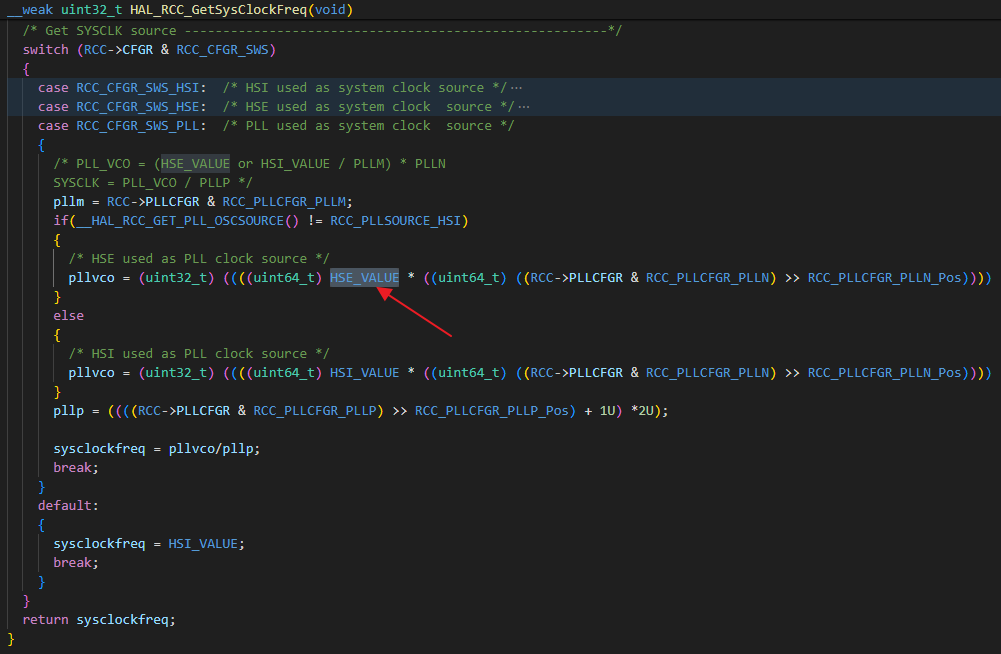
HAL\_Delay函数使用的是systick定时器，这个定时器在系统启动的时候进行过两次配置。在系统复位后，默认系统时钟为 HSI，f407芯片是16Mhz，程序刚开始调用HAL\_Init();函数时系统时钟SystemCoreClock就是16Mhz。



之后调用 SystemClock\_Config();函数，在HAL\_RCC\_ClockConfig()函数最后重新计算了SystemCoreClock，然后再一次初始化systick。



在HAL\_RCC\_GetSysClockFreq()函数里边用到了HSE\_VALUE来计算。



所以，如果HSE\_VALUE不对，HAL\_RCC\_GetSysClockFreq()返回的SYSCLK就不对，导致SystemCoreClock不对，进而导致systick配置不对，中断便不是1ms一次，延时函数也就不对了。