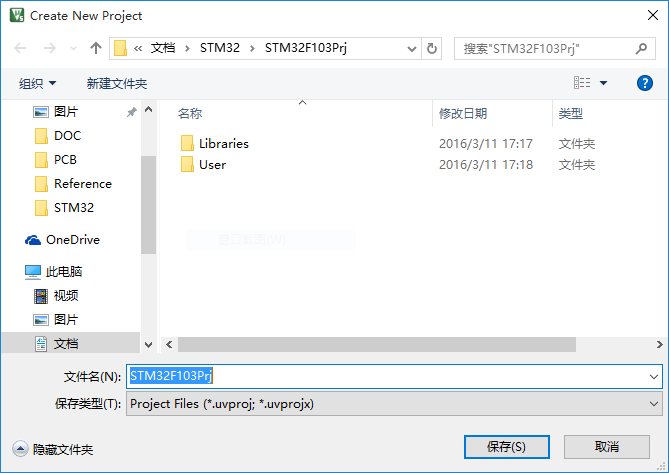
**开始编写第一个STM32程序**

找个地方新建一个用于存放项目的目录，建议用纯英文目录! 例如：STM32F103Prj

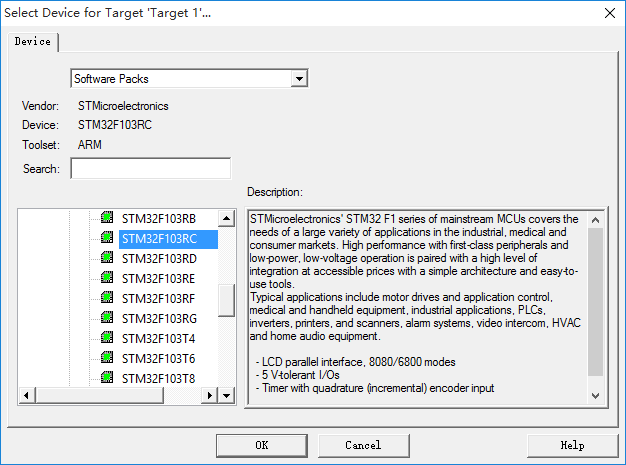
将标准库中的Libraries整个目录复制到项目目录下。

在项目目录下新建User目录用于存放你自己的源代码。将stm32f10x\_it.h、stm32f10x\_it.c和stm32f10x\_conf.h文件复制到User目录下。

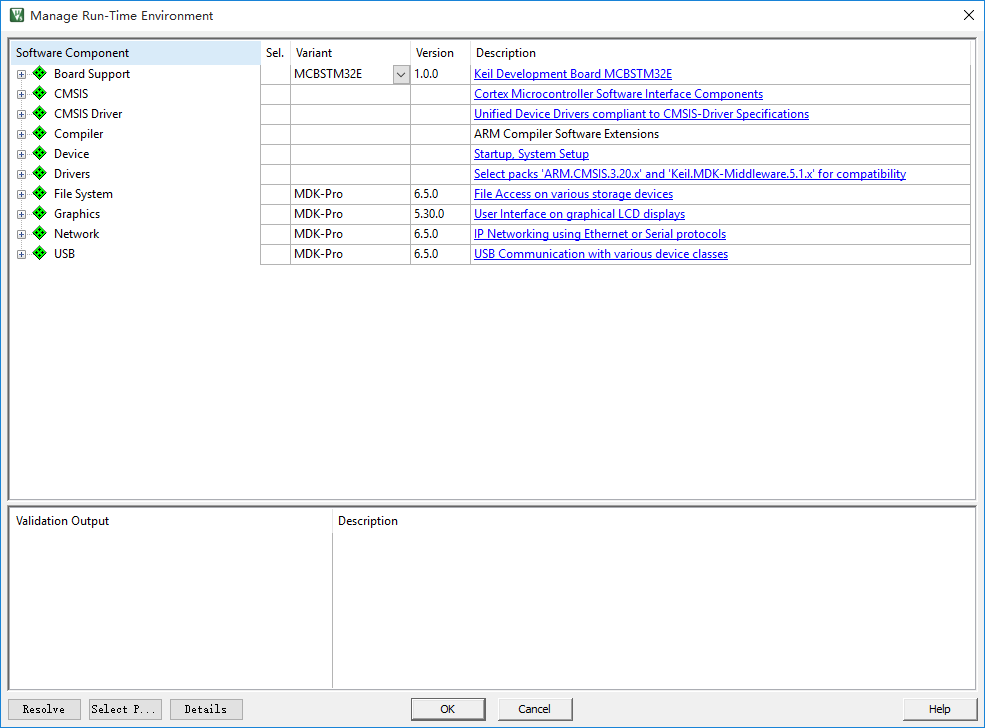
在Keil中新建一个项目：



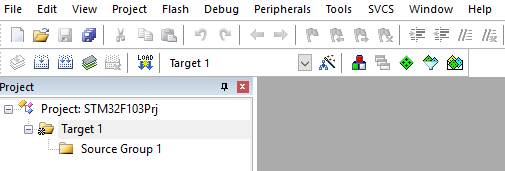
点保存，跳出选择芯片窗口



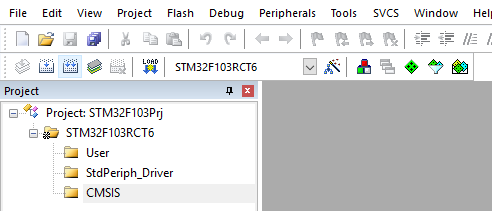
选择你要的芯片，然后点OK



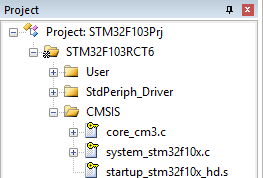
继续点OK



将Project：下的Target 1改成你的芯片型号，例如STM32F103RCT6；将Source Group 1改成User用于放你自己的代码，对应项目目录中的User目录。新建CMSIS和StdPeriph\_Driver两个组，分别放系统文件和驱动标准库。

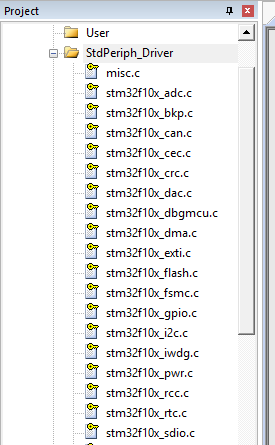


将Libraries\CMSIS目录下的三个文件添加到CMSIS组中。

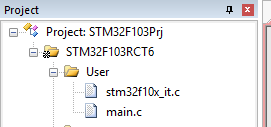


其中startup\_stm32f10x\_hd.s可以根据你的芯片选择不同的，如startup\_stm32f10x\_md.s。

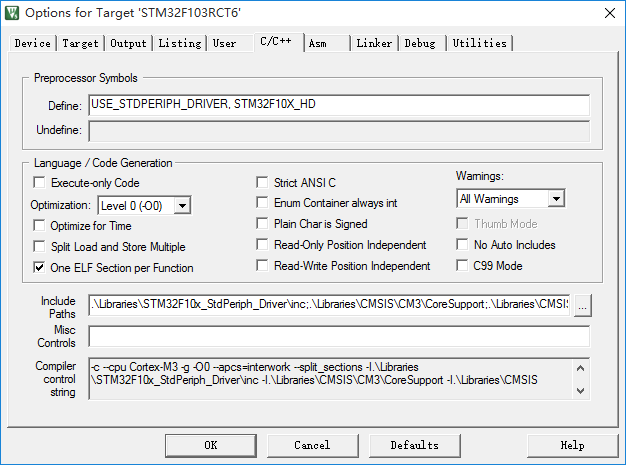
将Libraries\STM32F10x\_StdPeriph\_Driver\src目录下的驱动标准库源文件添加到StdPeriph\_Driver组中，你可以根据你的需要进行筛选，用不到的可以不添加。



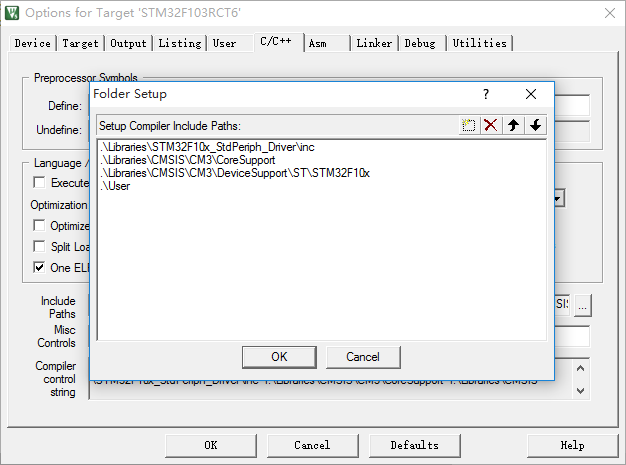
在User组中将刚才复制到User目录下的stm32f10x\_it.c文件添加进去。在User组中新建一个main.c文件作为你自己的主函数文件，建议main.c文件也要保存到项目目录的User目录下。



设置C符号定义，Define：USE\_STDPERIPH\_DRIVER, STM32F10X\_HD。这里的符号定义STM32F10X\_HD要根据你的芯片有所区别，譬如可能是STM32F10X\_MD



将系统和库函数的头文件\*.h所在目录包含到项目中去，以便编译器可以找到这些头文件。



在main.c文件中输入：

#include "stm32f10x.h"

#include "misc.h"

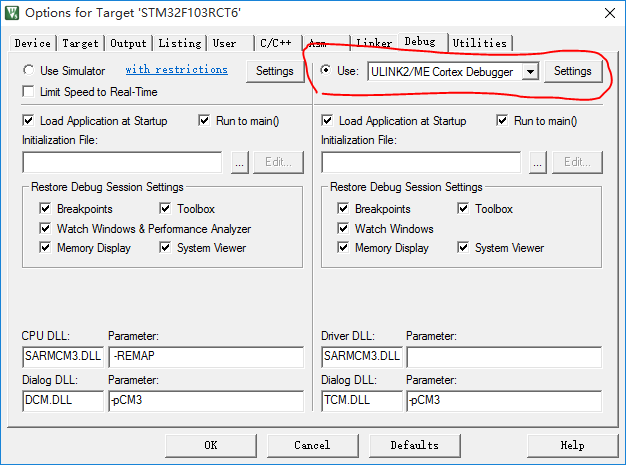
int main(void)

{

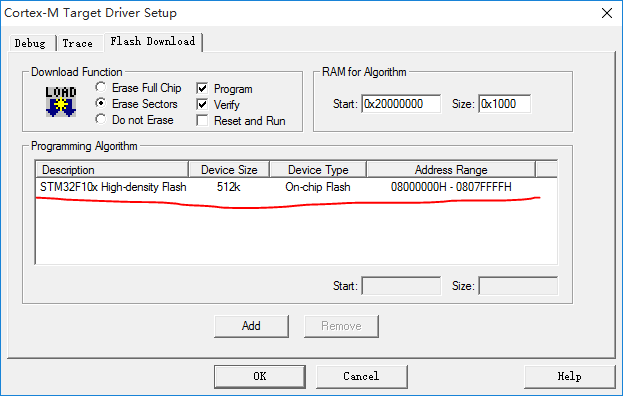
}

尝试编译，如果编译通过，则可以开始编程了，如果编译出错，多半是因为符号定义与芯片不符，或者所需要的头文件没找到。根据芯片存储容量调整STM32F10X\_HD这个符号定义，并且添加相应的.s到CMSIS组中；检查是否设置了头文件路径。

在Options里设置调试器/仿真器



点击Settings按钮，查看Flash Download项下是否有目标芯片Flash。一切顺利的话，现在可以下载程序调试了。



准备好后开始写程序吧！

对于一个STM32项目来讲，一般先要做系统始终初始化，中断向量初始化，端口初始化这三件事，分别对应

void RCC\_Configuration(void);

void NVIC\_Configuration(void);

void GPIO\_Config(void);

这三个函数。为了方便，就把这三个函数写在main文件里了。在引用之前，先要声明一下。具体参见项目范例。

先来做一个延时功能吧！

新建两个文件：delay.h和delay.c，别忘了把delay.c加到User组里去。头文件放声明，源文件放定义。

我们利用系统时钟滴答中断来做精确延时。在中断配置NVIC\_Configuration函数里启用系统时钟滴答中断

SysTick\_Config(64000000/100000)

假设系统时钟为64MHz的话，64000000/100000=640，则640个时钟脉冲中断一次，就是10us中断一次了。没必要搞得太快，否则浪费cpu啊

设置好后，每10us会调用stm32f10x\_it.c中的SysTick\_Handler()函数。我们的思路是：需要延时的时候设置一个初值，然后在SysTick\_Handler中每次减一，减到0时延时结束。

在delay.c中定义一个全局变量

\_\_IO uint32\_t TimingDelay;

同时定义一个减一函数，直到0

void TimingDelay\_Decrement(void)

{

if (TimingDelay != 0x00)

{

TimingDelay--;

}

}

别忘了在delay.h中声明一下这个函数，并在stm32f10x\_it.c中include下delay.h。就可以在SysTick\_Handler函数中调用TimingDelay\_Decrement()减一计算了。

我们再在delay.c中写两个延时函数，并在delay.h中声明一下。

void Delay\_10us(\_\_IO uint32\_t nTime)

{

TimingDelay = nTime;

while(TimingDelay != 0);

}

void Delay\_ms(\_\_IO uint32\_t nTime)

{

uint32\_t i;

i=nTime;

while(i--)

Delay\_10us(100);

}

注意：函数定义和函数声明要同时进行哦!

搞定后，在main.c文件中include下delay.h就可以用delay功能了。

Delay\_ms(1);

GPIO\_SetBits(GPIOC,GPIO\_Pin\_1);

Delay\_ms(1);

GPIO\_ResetBits(GPIOC,GPIO\_Pin\_1);

注：PC1端口的初始化在GPIO\_Config中完成了。

编译下载，没问题的话，运行后用示波器看下PC1出来一个1KHz的方波就正确了。