

## 56DZ ARM仿真器接口图

3V3	1 ● ● 2	3V3
JTRST	3 ● ● 4	GND
JTDI	5 ● ● 6	GND
JTMS	7 ● ● 8	GND
JTCK	9 ● ● 10	GND
NULL	11 ● ● 12	GND
JTDO	13 ● ● 14	GND
RSET	15 ● ● 16	GND
NULL	17 ● ● 18	GND
NULL	19 ● ● 20	GND

ARM仿真器有20个引脚，当凹口在左侧时，左上脚为一号引脚。本仿真器支持JTAG和SWJ模式，用20PIN线连接到开发板，或用杜邦线连接关键引脚。

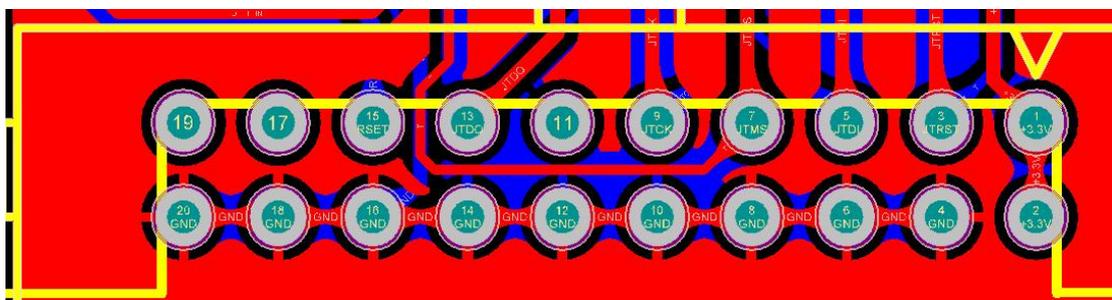
### JTAG接线方式：

1=>3V3、3=>TRST、4=>GND、  
5=>TDI、7=>TMS、9=>TCLK、  
13=>TDO、15=>RESET

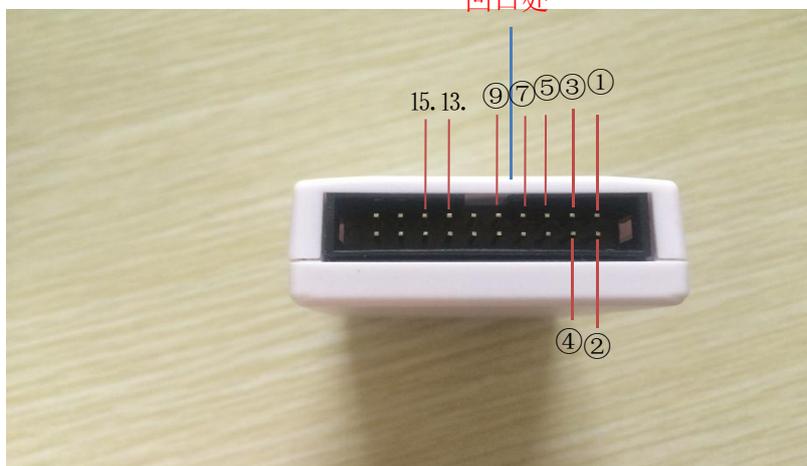
### SWJ接线方式：

1=>3V3、4=>GND、7=>SWDIO、  
9=>SWCLK

PCB 引脚图

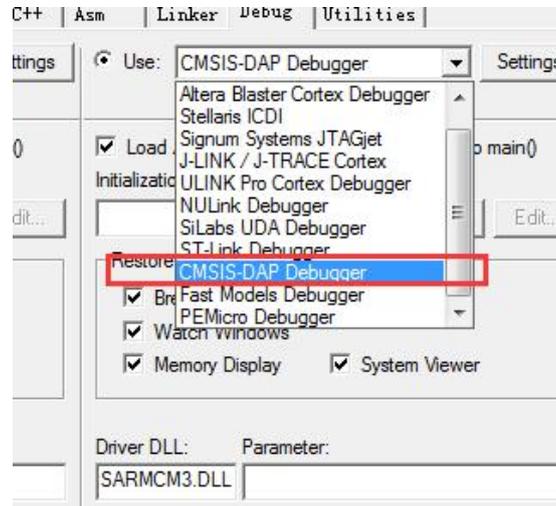


凹口处



# ARM 仿真器下载操作说明

本文用的 Keil 版本是 V5.21.1.0 版本,如果在硬件仿真设置中找不到 CMSIS-DAP 请更换版本,版本过低的 KEIL 不会显示 CMSIS-DAP-Debugger。

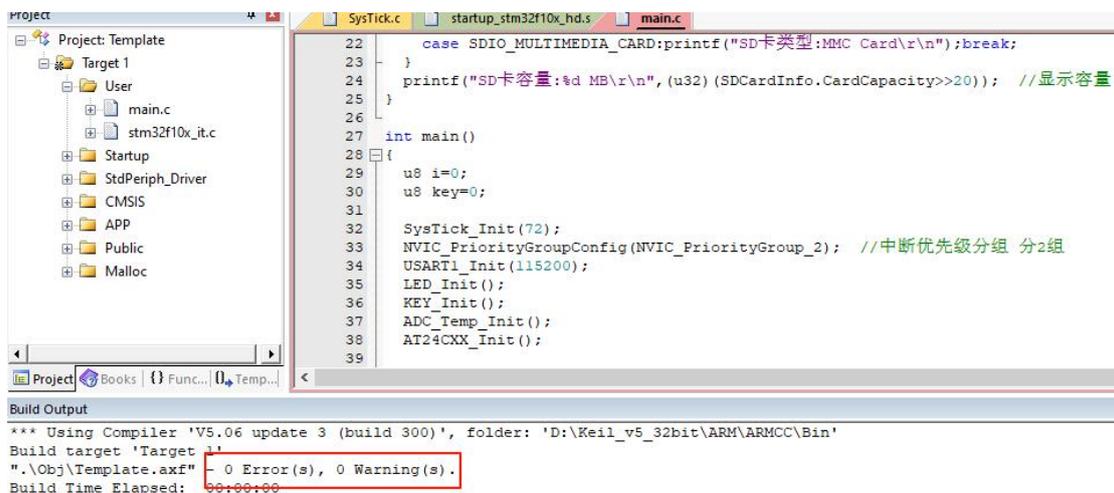


扫码观看视频教程

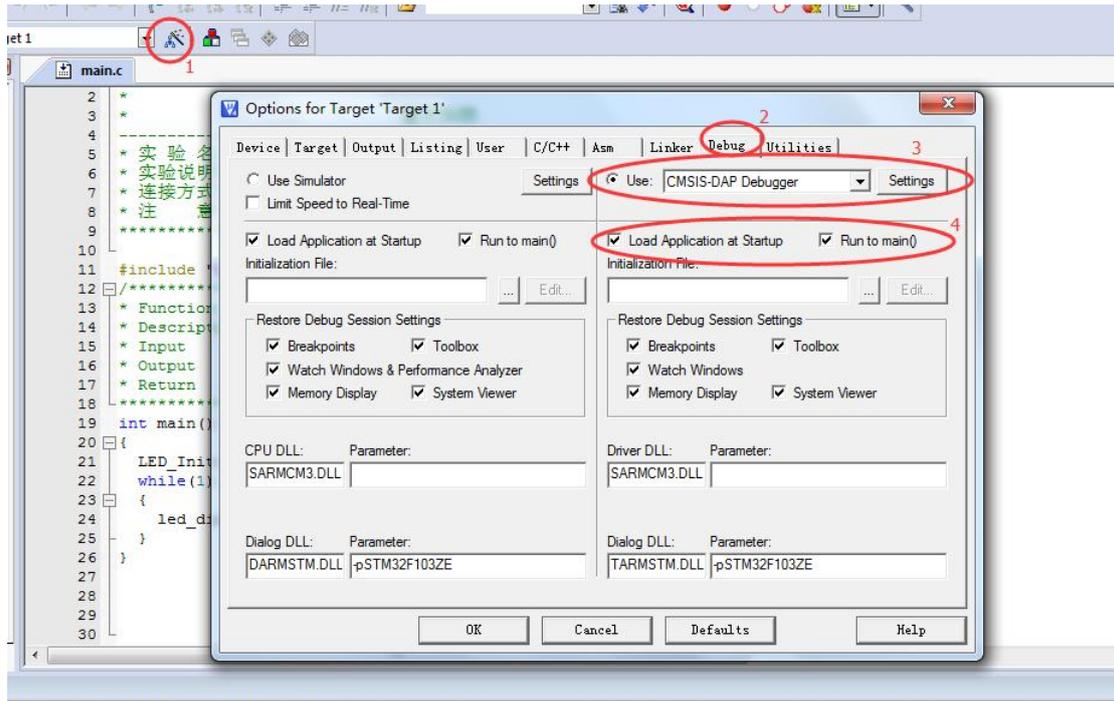


<https://www.56dz.com/p/4638.html>

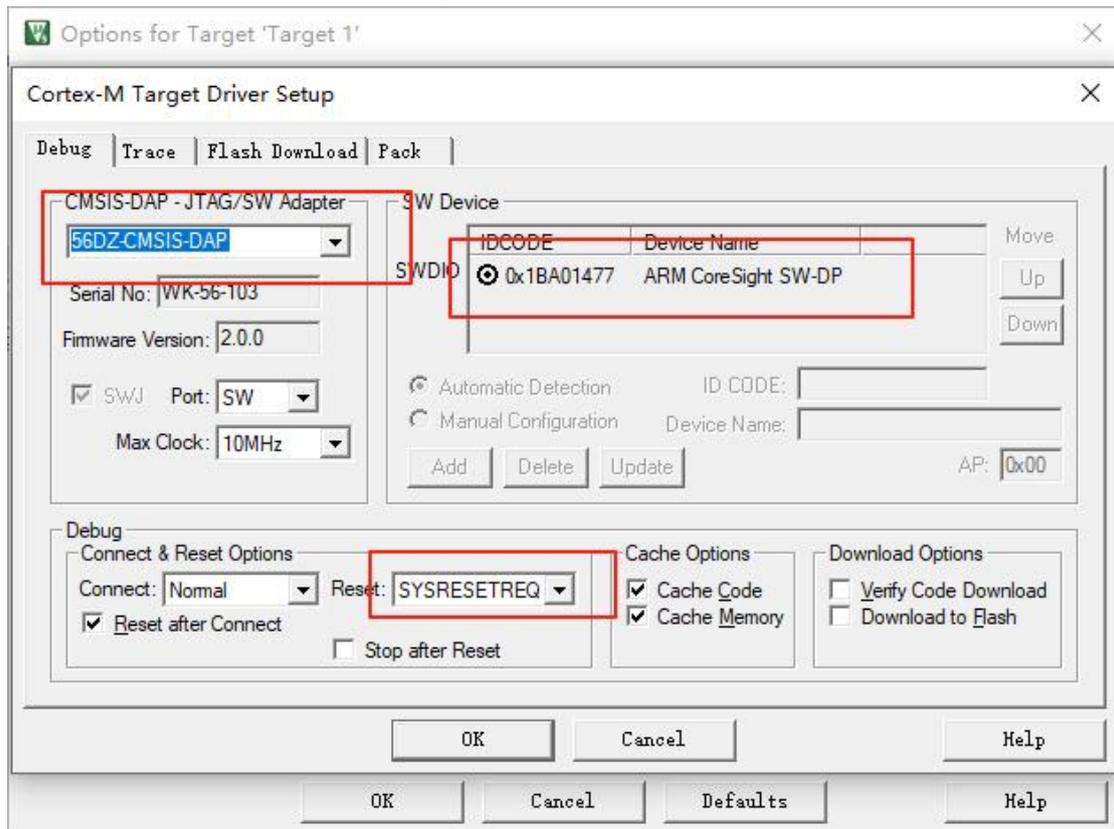
步骤: 打开一个能够正常编译通过的工程, 至少是 0Error

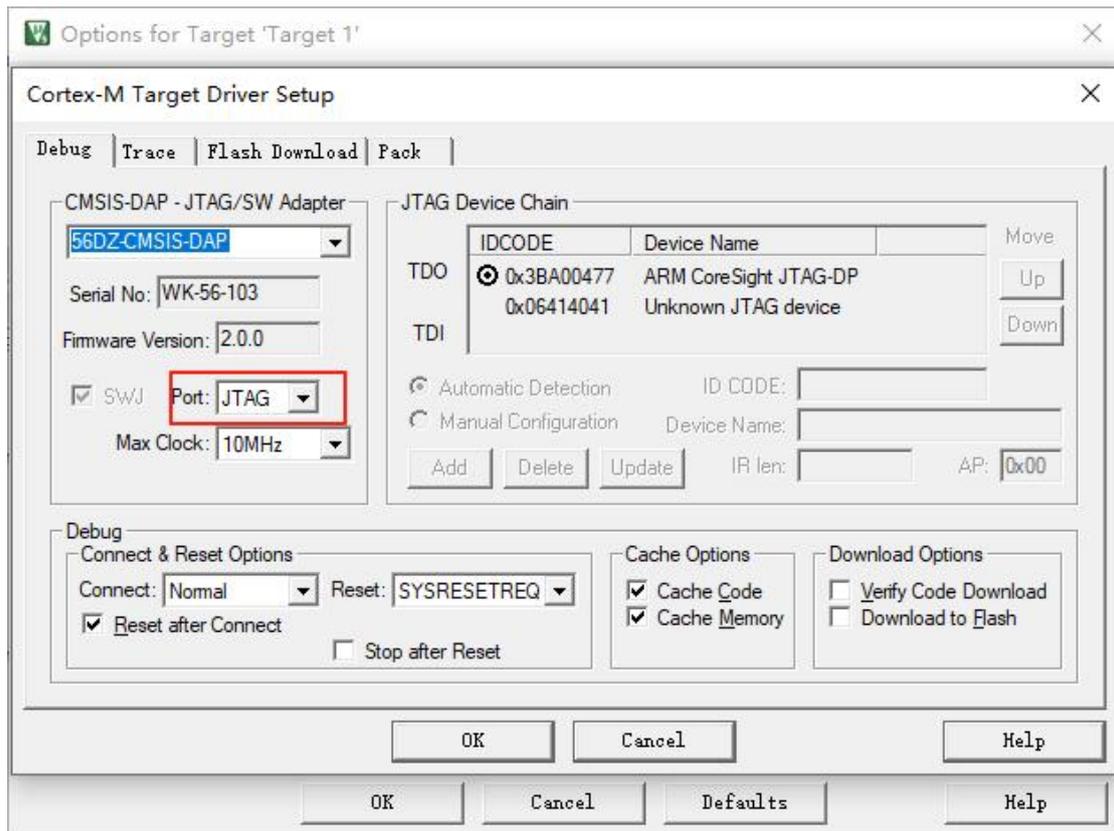


1、点击魔术棒->Debug 选择 CMSIS-DAP Debugger, 勾选 Load Application Startup 和 Run to main, 再点击 Setting 进入 KEIL 硬件仿真设置

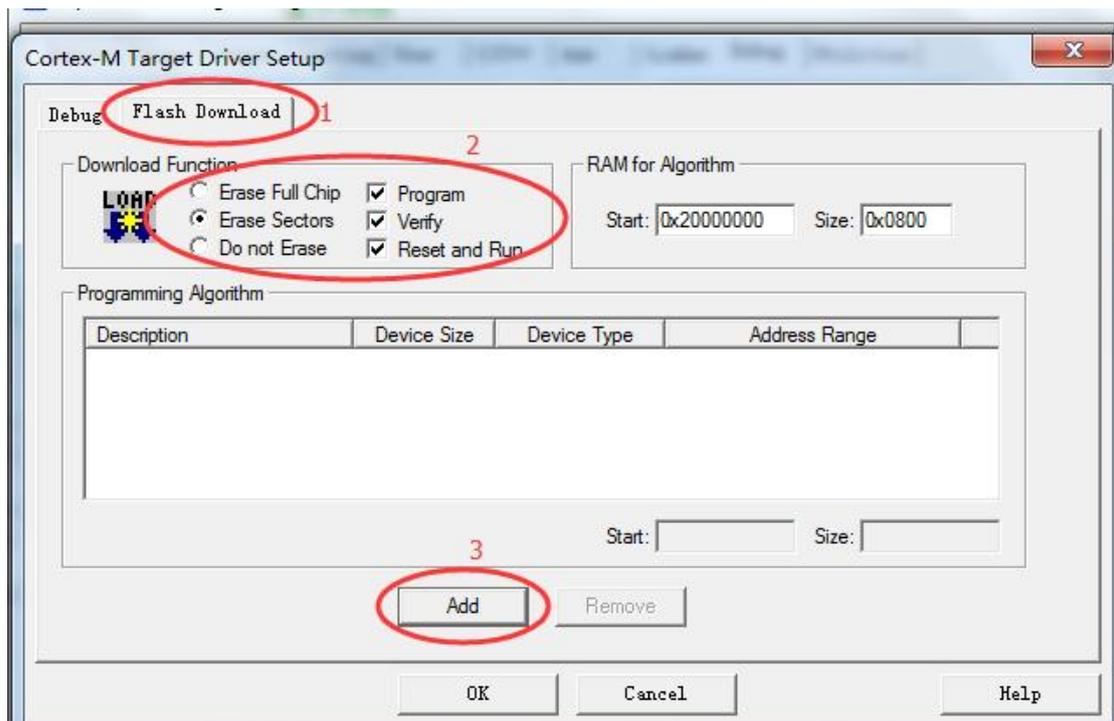


2、设置好硬件仿真后我们点 Settings 进入更深入的设置，请按照我这个面板这样设置，此处可以选择 SW 和 JTAG 两种下载模式，Reset 选择 SYSRESETREQ。

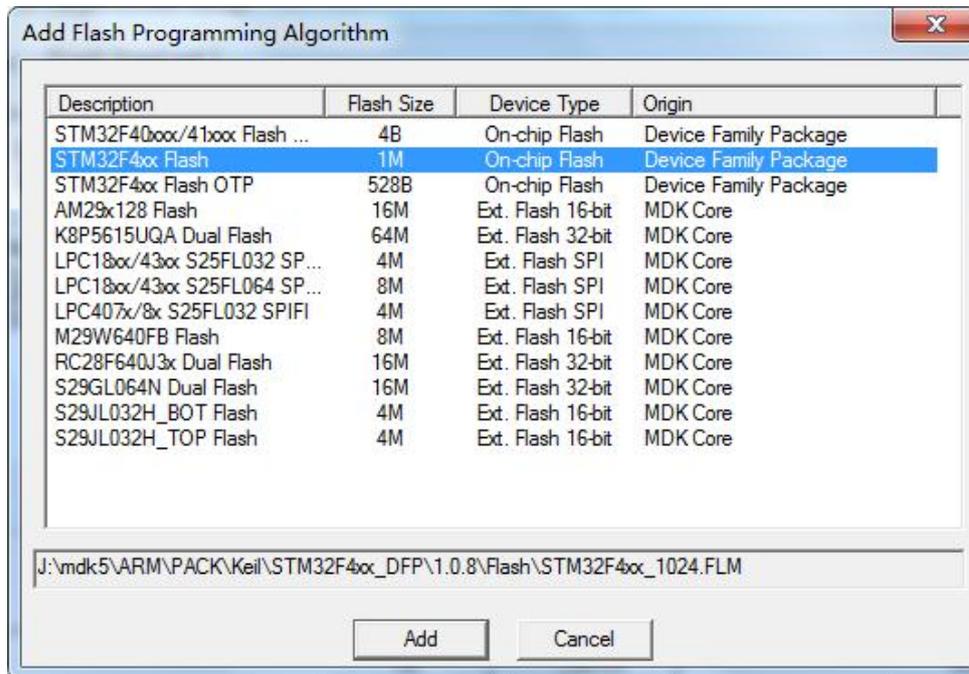




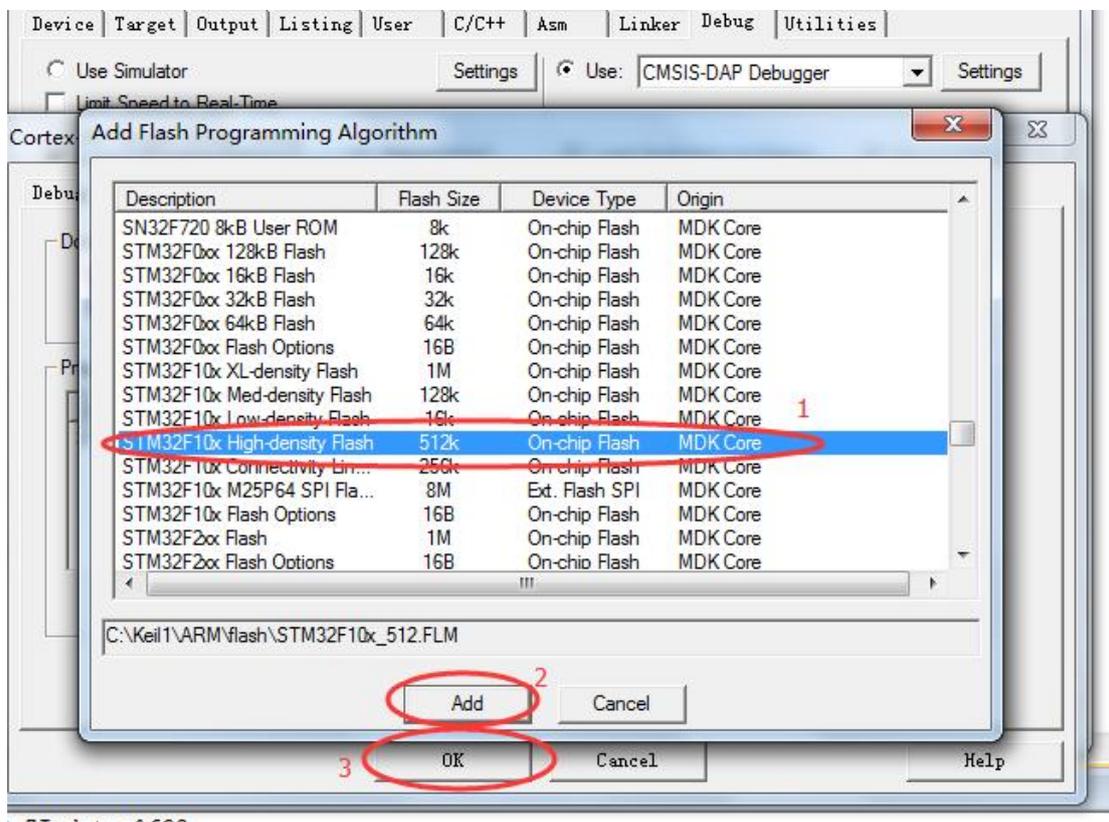
3、设置好 debug 页面。我们点击 Flash Download 进入下载设置根据需要勾选。分别代表的意思是下载之前擦除整块芯片/擦除用到的/不擦除/编程下载/校验/下载完后复位并且重启。一般建议勾选 Erase Sectors、Program、Verify、Reset and Run。



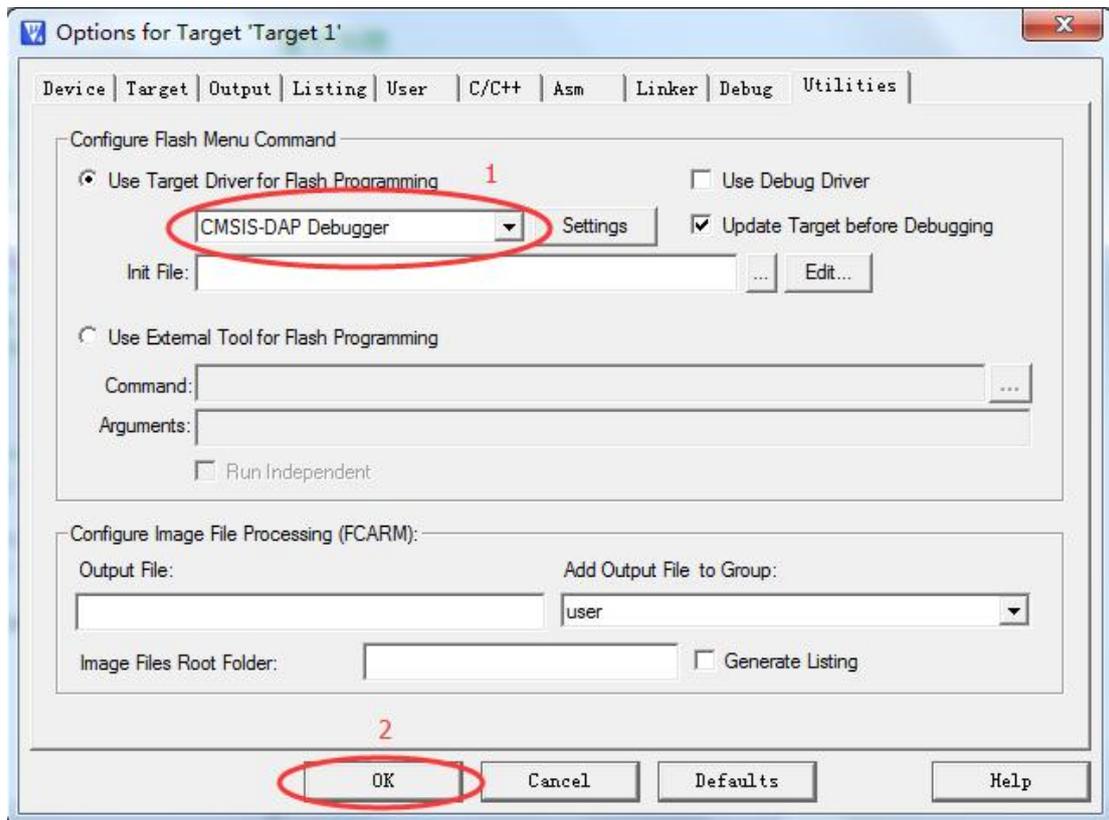
4、点击 add 设置编程算法。以 STM32F407ZG 为例，找到 STM32F4x x Flash 1M。



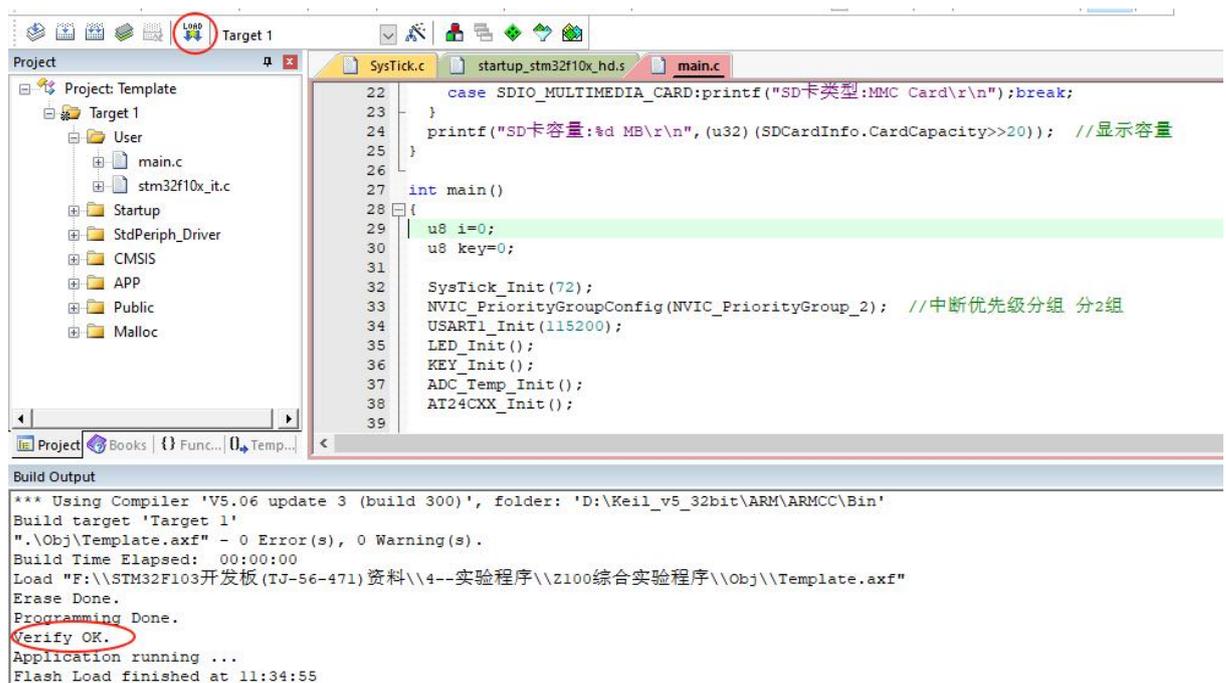
如果是 STM32F1 程序下载，找到 STM32F10x High-density Flash 512k 选中点 add 添加就回到第 4 步页面，有显示 STM32F10x High-density 512k 点 ok 完成设置。总之查询目标单片机对应的算法。



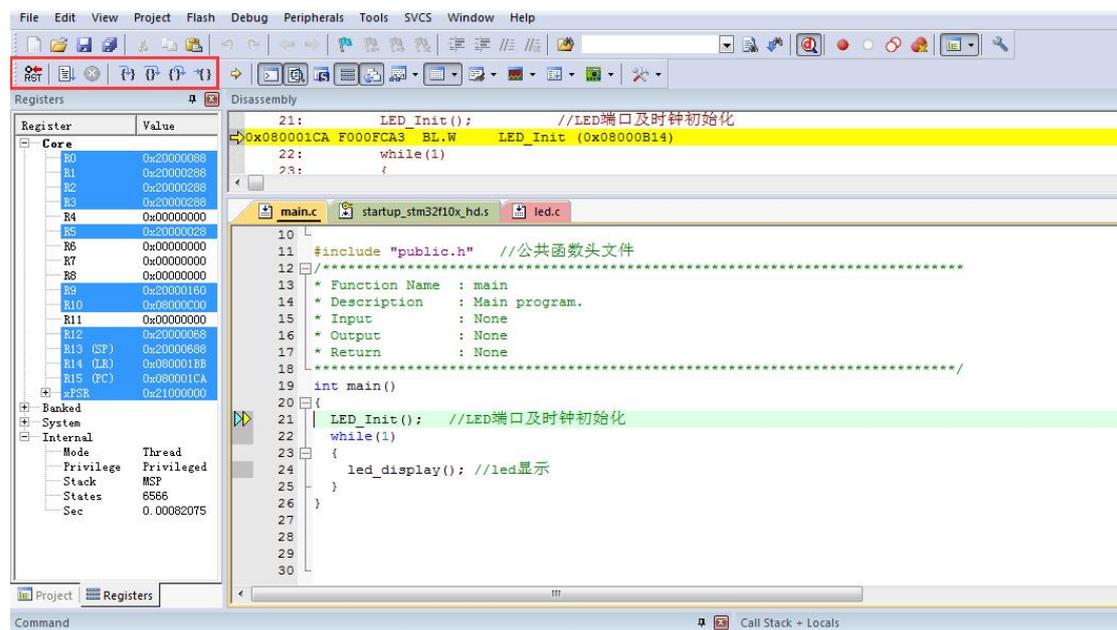
6、完成以上设置后我们点 Utilities 页面，这里也选择 CMSIS-DAP 选择好之后我们点 ok 完成设置(ps: 如果这个界面没有 Use Target for flash Programming 选择 CMSIS-DAP 就忽略这一项)



7、(keil 下载) 设置好之后我们编译程序没问题我们点 Download 进行下载提示 Verify OK 就说明已经下载成功了



8、（在线仿真调试）我们可以直接在 keil 里面调试程序，点工具栏的 start debug 开始调试，如果退出也是点这个按钮



点击之后我们就来到了这个页面，RST 是复位的意思、第二个是全速运行、第三个是停止运行，再过来就是我们程序检查程序中错误的时候会用到的功能的，第一个箭头是单步运行、第二个和第一个功能差不多，第三个是跳出这个函数，第四个是进入函数内部。



**复位：**相当于硬件上的复位按钮或者电源重启。

**执行到断点处：**或者说是开始或继续执行程序按钮。如果程序已经停在一个断点上，点击这个按钮将继续执行程序直到遇到下一个断点或程序结束。也就是说如果没有断点，程序会一直运行。

**停止执行：**中止当前正在运行的程序。

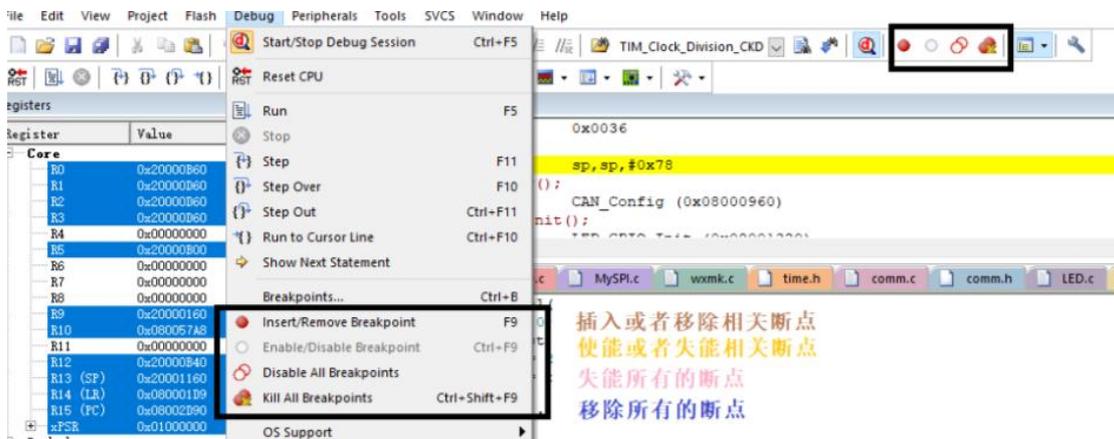
**执行进去：**执行下一行代码，如果是调用函数就进入到该函数内部。

**执行过去：**执行下一行代码，但是不进入函数内部。如果当前行调用了函数，IDE 将执行整

个函数，然后停在下一行。也就是说，如果在主函数中使用这个功能，会一直停留在主函数页面运行每一行代码，不会调转到其他函数中去。

**执行出去：**当运行在函数内部时，这个按钮将执行完当前函数的剩余代码，并停在返回点之后的下一行。

**执行到光标处：**程序执行将继续进行，直到到达当前代码编辑器中光标所在的行。



**反汇编窗口：**用来开启如下图，示当前执行的汇编代码，与源代码视图同步，有助于理解C/C++代码是如何转换为处理器指令的，以及在调试时实际执行的指令。

**符号窗口：**列出了所有编译后的符号，包括函数、变量和类型定义。这个窗口让你可以确认是否有符号未定义或检查变量的地址和值。

**内核寄存器：**显示处理器内核的寄存器值。有助于理解程序的当前状态和执行流程，尤其是在硬件层面的调试中。

**回调部变量：**当调试时停留在断点时，这里会显示函数的调用关系，以及压进的变量值，可以在这里看到函数的调用关系和传递给函数的参数值。这个一般和断点配合，反推异常调用

以及查看局部变量，无需使用变量查看串口。

**变量显示:**用于查看单片机中某个变量的值，一般多用于查看全局变量以及外设寄存器数值、表达式显示变量，也可直接操作变量值。

**内存窗口:**显示内存地址以及地址处内存的数据，一般可查看变量以及寄存器数据、函数地址等。

这些功能大家自己多运行体验一下效果就能理解是什么意思了。