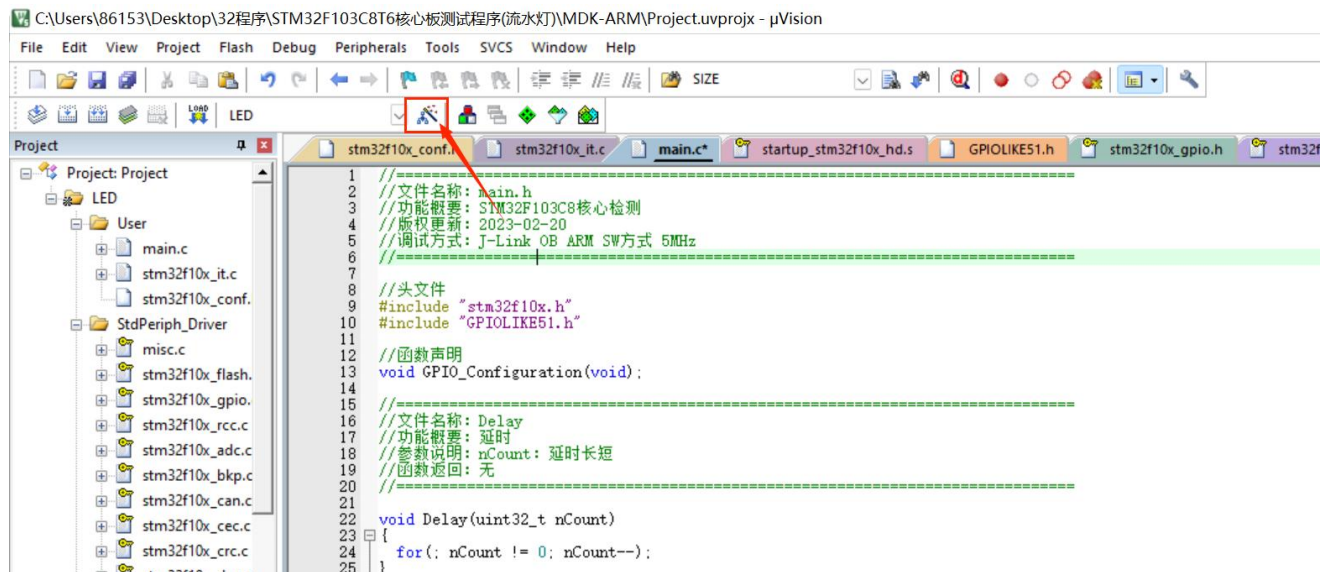
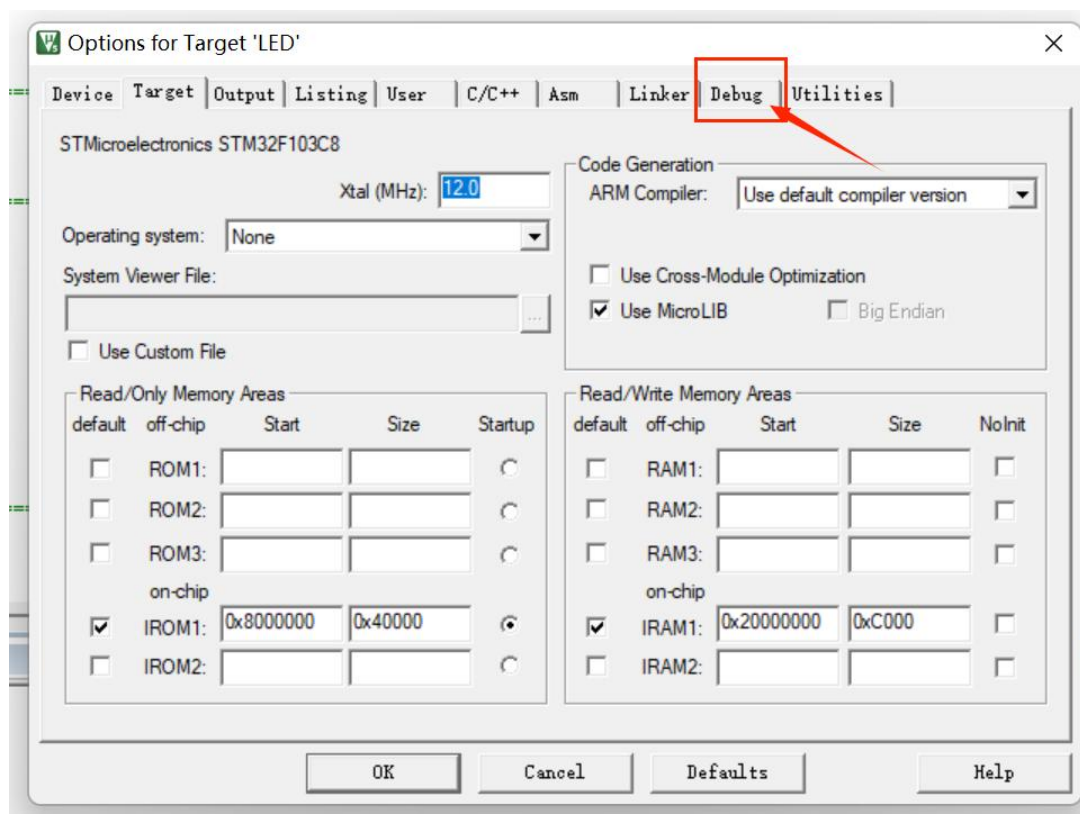


1. 打开 keil 软件，点击魔法棒。

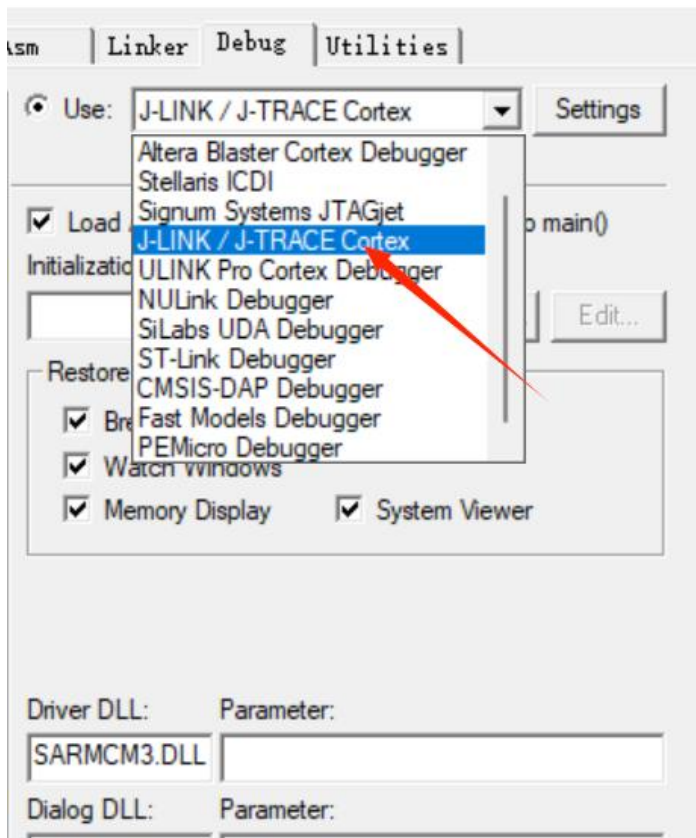


2. 点击 DUBeg 选项。

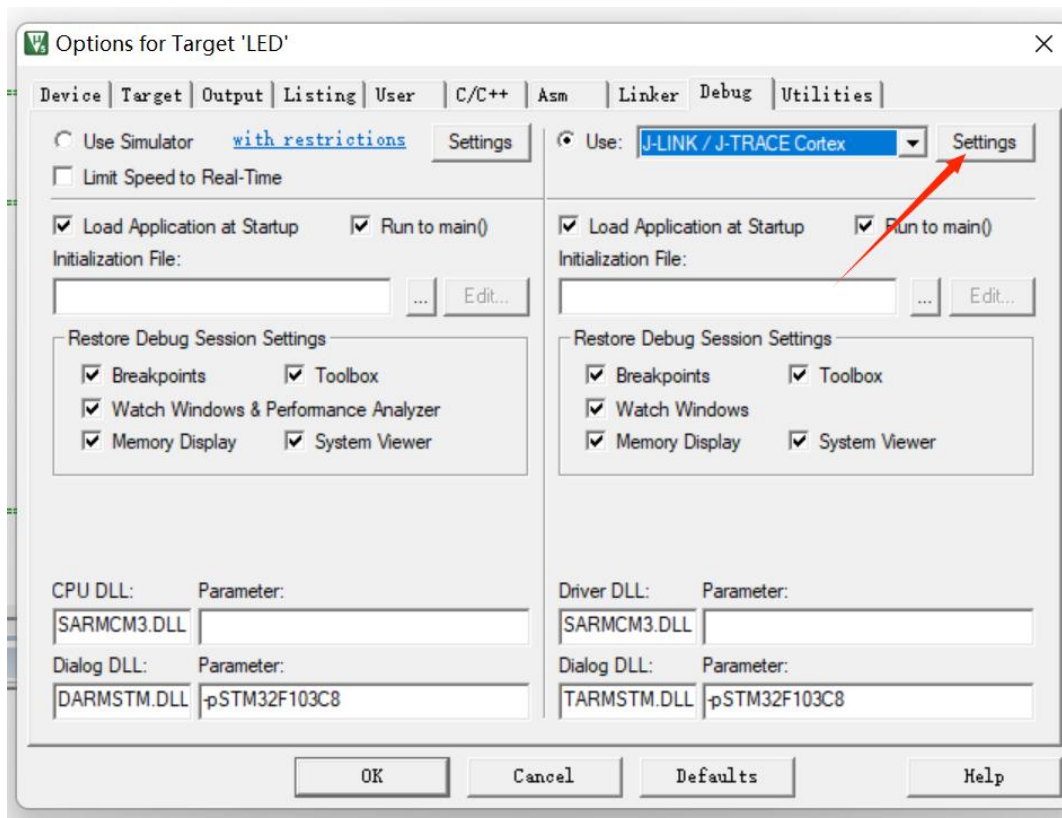


3. 选中 J-LINK 选项

×

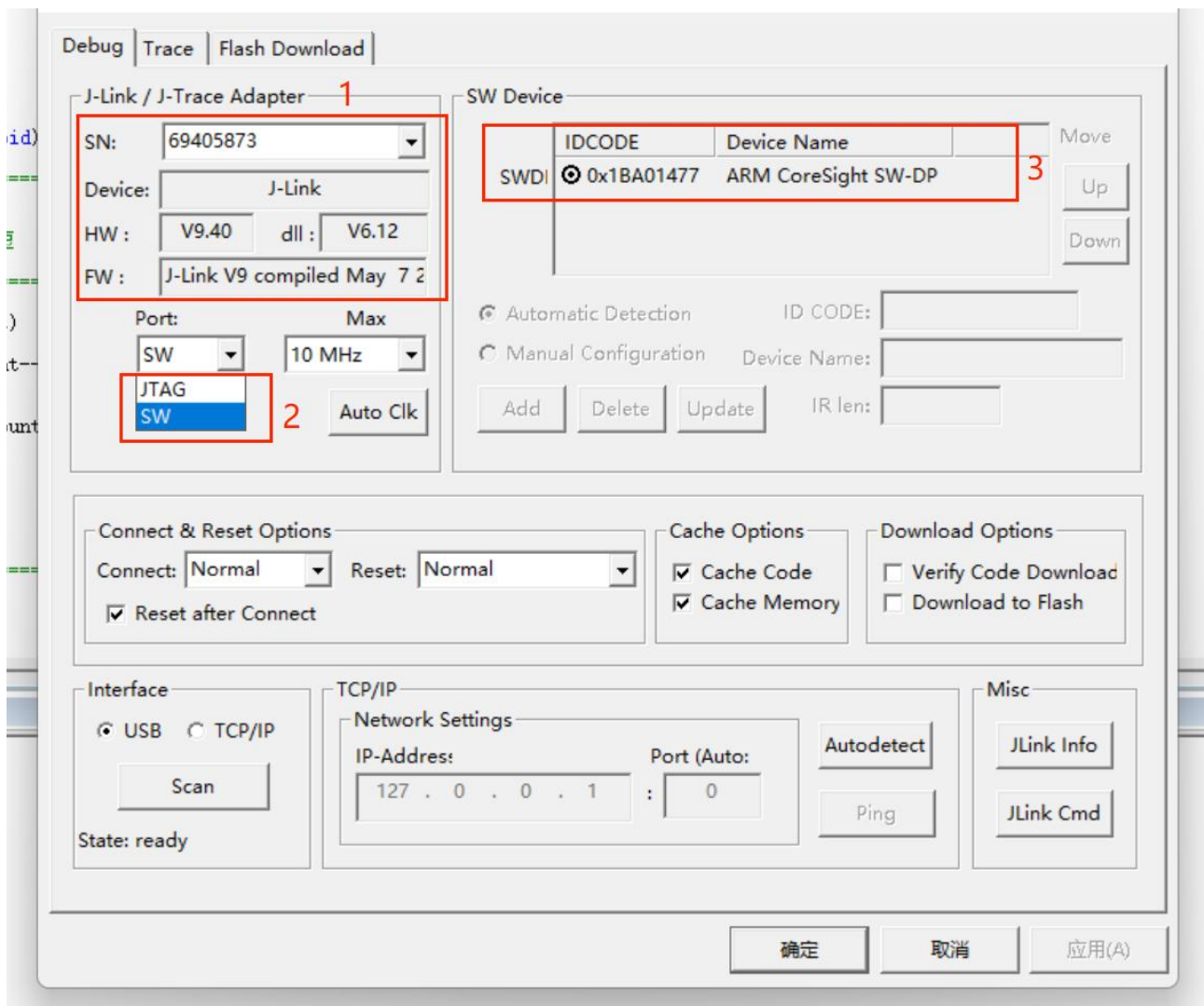


#### 4. 点击 Settings

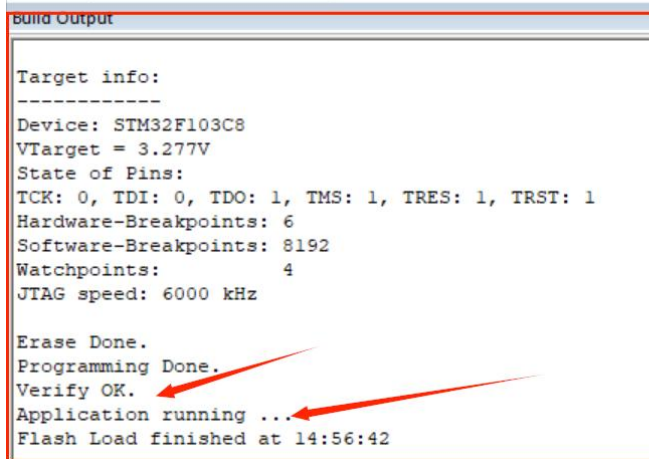
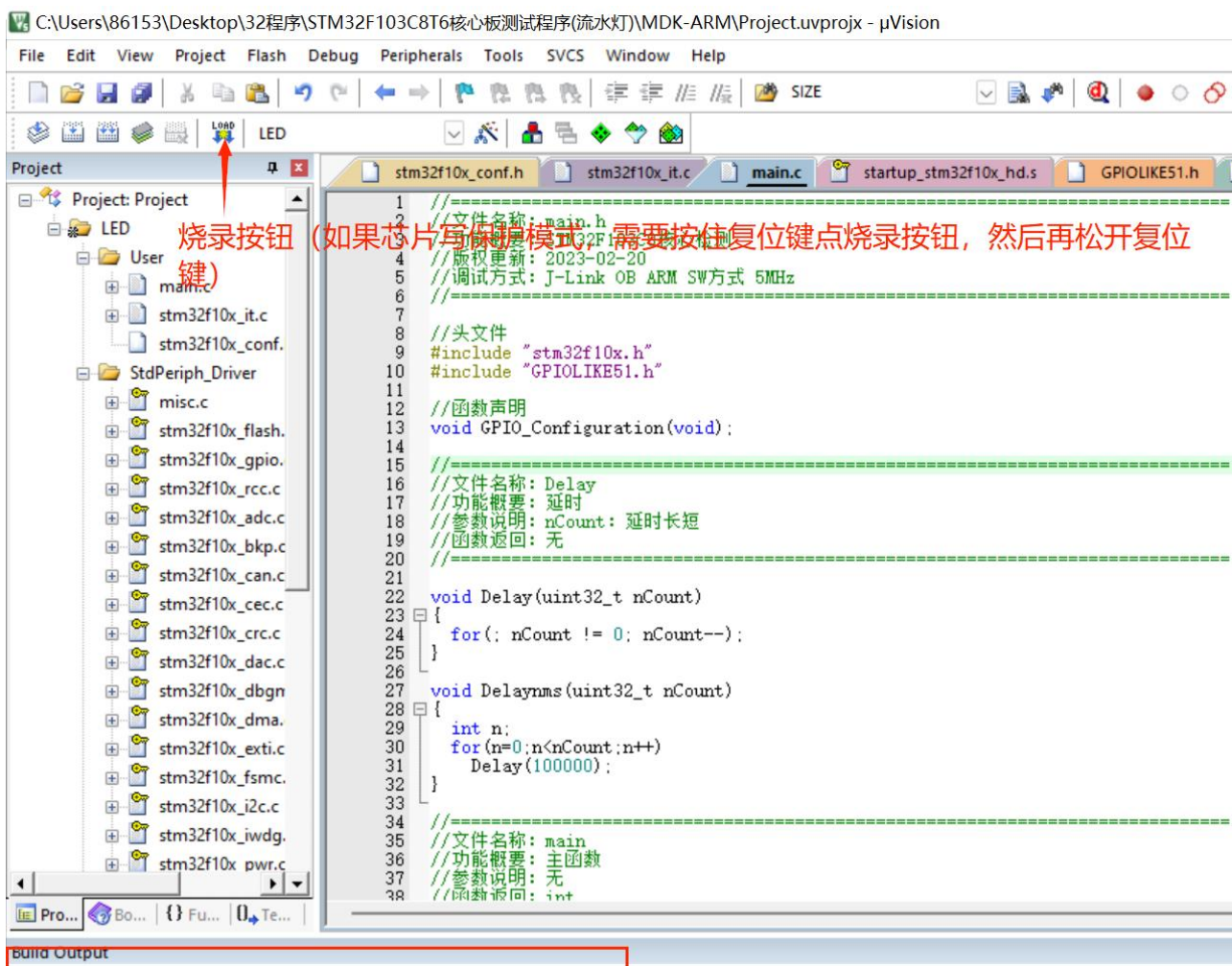


5. 设置界面介绍，1 处可以看到 JLINK 设备的 SN 码，硬件版本号，DLL 版本（即驱动版本）。2 处可以根据实现情况选择 SW（SWD）或 JTAG 接口，其中 SW 只需要 2 个通信线即可，比较常用。3 处可以看到目标芯片的 IDCODE。

（目标芯片即为 JLINK 仿真器所连接的芯片，目标板即为芯片所在的电路板）



6. 烧录程序：如果有源代码就可以用 keil 软件进行烧录。在步骤 5 中可以看到目标芯片的 IDCODE，则表明仿真器已经连接至目标芯片，已建立通信。那么我们打开源代码后，将源代码编译通过后即可烧录。

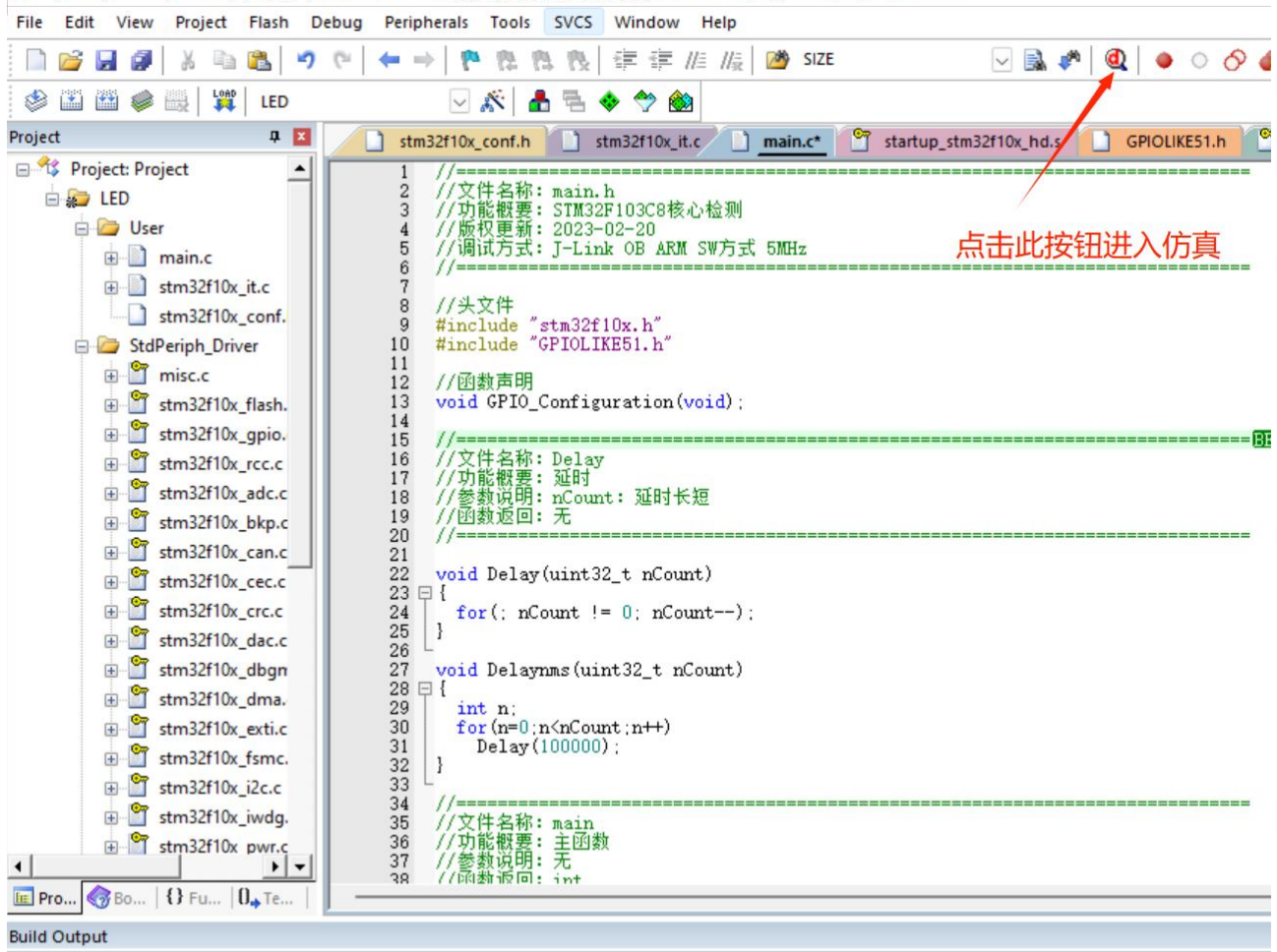


烧录状态框, 此时显示烧录完成, 程序运行中。

7. 代码仿真: 点击仿真按钮直接进入仿真界面即可。



C:\Users\86153\Desktop\32程序\STM32F103C8T6核心板测试程序(流水灯)\MDK-ARM\Project.uvprojx - μVision



C:\Users\86153\Desktop\32程序\STM32F103C8T6核心板测试程序(流水灯)\MDK-ARM\Project.uvprojx - uVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
R0	0x080005A1
R1	0x20000400
R2	0x00000000
R3	0x08000591
R4	0x080009B4
R5	0x080009B4
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x20000160
R10	0x00000000
R11	0x00000000
R12	0x00000010
R13 (SP)	0x20000400
R14 (LR)	0x08000185
R15 (PC)	0x080005A0
xPSR	0x61000000

Disassembly

30: for(n=0;n<nCount;n++)  
0x080001A0 2300 MOVS r3,#0x00  
0x080001A2 E003 B 0x080001AC  
Delay(1000000)  
31: Delay(1000000)  
32: Delay(1000000)  
33: Delay(1000000)  
34: Delay(1000000)  
35: Delay(1000000)  
36: Delay(1000000)  
37: Delay(1000000)  
38: Delay(1000000)  
39: Delay(1000000)  
40: Delay(1000000)  
41: Delay(1000000)  
42: Delay(1000000)  
43: Delay(1000000)  
44: Delay(1000000)  
45: Delay(1000000)  
46: Delay(1000000)  
47: Delay(1000000)  
48: Delay(1000000)  
49: Delay(1000000)  
50: Delay(1000000)  
51: Delay(1000000)  
52: Delay(1000000)

stm32f10x\_conf.h stm32f10x\_it.c main.c startup\_stm32f10x\_hd.s GPIO.h stm32f10x\_gpio.h stm32f10x\_gpio.c stm32f10x\_it.c

```
23 {  
24     for(; nCount != 0; nCount--);  
25 }  
26  
27 void Delayms(uint32_t nCount)  
28 {  
29     int n;  
30     for(n=0;n<nCount;n++)  
31         Delay(1000000);  
32 }  
33  
34 //文件名称: main  
35 //功能概要: 主函数  
36 //参数说明: 无  
37 //函数返回: int  
38 //-----  
39  
40 int main(void)  
41 {  
42     GPIO_Configuration();  
43     PBout(11)=0;PBout(10)=0;PBout(1)=0;PBout(0)=0;PAout(7)=0;PAout(6)=0;PAout(5)=0;PAout(4)=0;PAout(3)=0;PAout(2)=0;PAout(1)=0;  
44     PAout(0)=0;PCout(15)=0;PCout(14)=0;PCout(13)=0;PBout(9)=0;PBout(8)=0;PBout(7)=0;PBout(6)=0;PBout(5)=0;PBout(4)=0;PBout(3)=0;  
45     PAout(15)=0;PAout(12)=0;PAout(11)=0;PAout(10)=0;PAout(9)=0;PAout(8)=0;PBout(15)=0;PBout(14)=0;PBout(13)=0;PBout(12)=0;  
46     Delayms(10);  
47     PBout(11)=1;PBout(10)=1;PBout(1)=1;PBout(0)=1;PAout(7)=1;PAout(6)=1;PAout(5)=1;PAout(4)=1;PAout(3)=1;PAout(2)=1;PAout(1)=1;  
48     PAout(0)=1;PCout(15)=1;PCout(14)=1;PCout(13)=1;PBout(9)=1;PBout(8)=1;PBout(7)=1;PBout(6)=1;PBout(5)=1;PBout(4)=1;PBout(3)=1;  
49     PAout(15)=1;PAout(12)=1;PAout(11)=1;PAout(10)=1;PAout(9)=1;PAout(8)=1;PBout(15)=1;PBout(14)=1;PBout(13)=1;PBout(12)=1;  
50     Delayms(10);  
51     while (1)  
52 {
```

Command

Target info:

Name	Location/...	Type

再次点击可返回

仿真所需要用的功能按钮，具体操作请百度自行研究