

# Homework5 板子测试

## Problem

1. 编写上电后能够启动的程序并在实验板上运行。
  - (1) 程序分为C和汇编两个部分。
  - (2) 汇编部分对系统进行初始化 (s2410a.s) 。
  - (3) C部分控制实验板上LED循环显示。
  - (4) 通过仿真器下载程序到RAM中运行，观察结果。
  - (5) 将程序写入板子NOR FLASH中，实现上电后自动启动。

## Answer

### 程序分为C和汇编两个部分

C代码：

```
#include<stdlib.h>
#include<stdio.h>
int *rGPFCON = (int *) 0x56000050;
int *rGPFDAT = (int *) 0x56000054;
int delay(int times);

int main(void)
{
    *rGPFCON=0x5500;

    while(1)
    {
        int i;
        *rGPFDAT = 0x10;
        delay(500);

        *rGPFDAT = 0x00;
        delay(500);
        *rGPFDAT = 0xf0;
        delay(500);
    }
    return 1;
}

int delay(int times)
{
    int i,j;
    for(i=0;i<times;i++)
        for(j=0;j<times;j++)
        {
            }
    return 1;
```

```
}
```

汇编代码即s2410a.s初始化文件和inisdram.s文件：

```
MC_BASE EQU 0x48000000 ; Memory Controller Base Address
BWSCON_OFS EQU 0x00 ; Bus Width and Wait Status Ctrl offset
BANKCON0_OFS EQU 0x04 ; Bank 0 Control Register offset
BANKCON1_OFS EQU 0x08 ; Bank 1 Control Register offset
BANKCON2_OFS EQU 0x0C ; Bank 2 Control Register offset
BANKCON3_OFS EQU 0x10 ; Bank 3 Control Register offset
BANKCON4_OFS EQU 0x14 ; Bank 4 Control Register offset
BANKCON5_OFS EQU 0x18 ; Bank 5 Control Register offset
BANKCON6_OFS EQU 0x1C ; Bank 6 Control Register offset
BANKCON7_OFS EQU 0x20 ; Bank 7 Control Register offset
REFRESH_OFS EQU 0x24 ; SDRAM Refresh Control Register offset
BANKSIZE_OFS EQU 0x28 ; Flexible Bank Size Register offset
MRSRB6_OFS EQU 0x2C ; Bank 6 Mode Register offset
MRSRB7_OFS EQU 0x30 ; Bank 7 Mode Register offset

MC_SETUP EQU 0
BWSCON_val EQU 0x22111110
BANKCON0_val EQU 0x00000700
BANKCON1_val EQU 0x00000700
BANKCON2_val EQU 0x00000700
BANKCON3_val EQU 0x00000700
BANKCON4_val EQU 0x00000700
BANKCON5_val EQU 0x00000700
BANKCON6_val EQU 0x00018005
BANKCON7_val EQU 0x00018005
REFRESH_val EQU 0x008e0459
BANKSIZE_val EQU 0x000000b2
MRSRB6_val EQU 0x00000030
MRSRB7_val EQU 0x00000030

AREA SDRAMINI, CODE, readonly
EXPORT INISDRAM
```

#### INISDRAM

```
LDR R0, =MC_BASE
LDR R1, =BWSCON_val
STR R1, [R0, #BWSCON_OFS]
LDR R1, =BANKCON0_val
STR R1, [R0, #BANKCON0_OFS]
LDR R1, =BANKCON1_val
STR R1, [R0, #BANKCON1_OFS]
LDR R1, =BANKCON2_val
STR R1, [R0, #BANKCON2_OFS]
LDR R1, =BANKCON3_val
STR R1, [R0, #BANKCON3_OFS]
LDR R1, =BANKCON4_val
STR R1, [R0, #BANKCON4_OFS]
LDR R1, =BANKCON5_val
STR R1, [R0, #BANKCON5_OFS]
LDR R1, =BANKCON6_val
STR R1, [R0, #BANKCON6_OFS]
```

```

LDR    R1,      =BANKCON7_Va]
STR    R1, [R0, #BANKCON7_OFS]
LDR    R1,      =REFRESH_Va]
STR    R1, [R0, #REFRESH_OFS]
MOV    R1,      #BANKSIZE_Va]
STR    R1, [R0, #BANKSIZE_OFS]
MOV    R1,      #MRSRB6_Va]
STR    R1, [R0, #MRSRB6_OFS]
MOV    R1,      #MRSRB7_Va]
STR    R1, [R0, #MRSRB7_OFS]

MOV    pc,lr
END

```

## 汇编部分对系统进行初始化 (s2410a.s)

因为没有引入中断程序，暂时不需要修改该文件

## C部分控制实验板上LED循环显示。

c代码的逻辑是控制四个LED等同时亮同时灭，延时0.5秒。

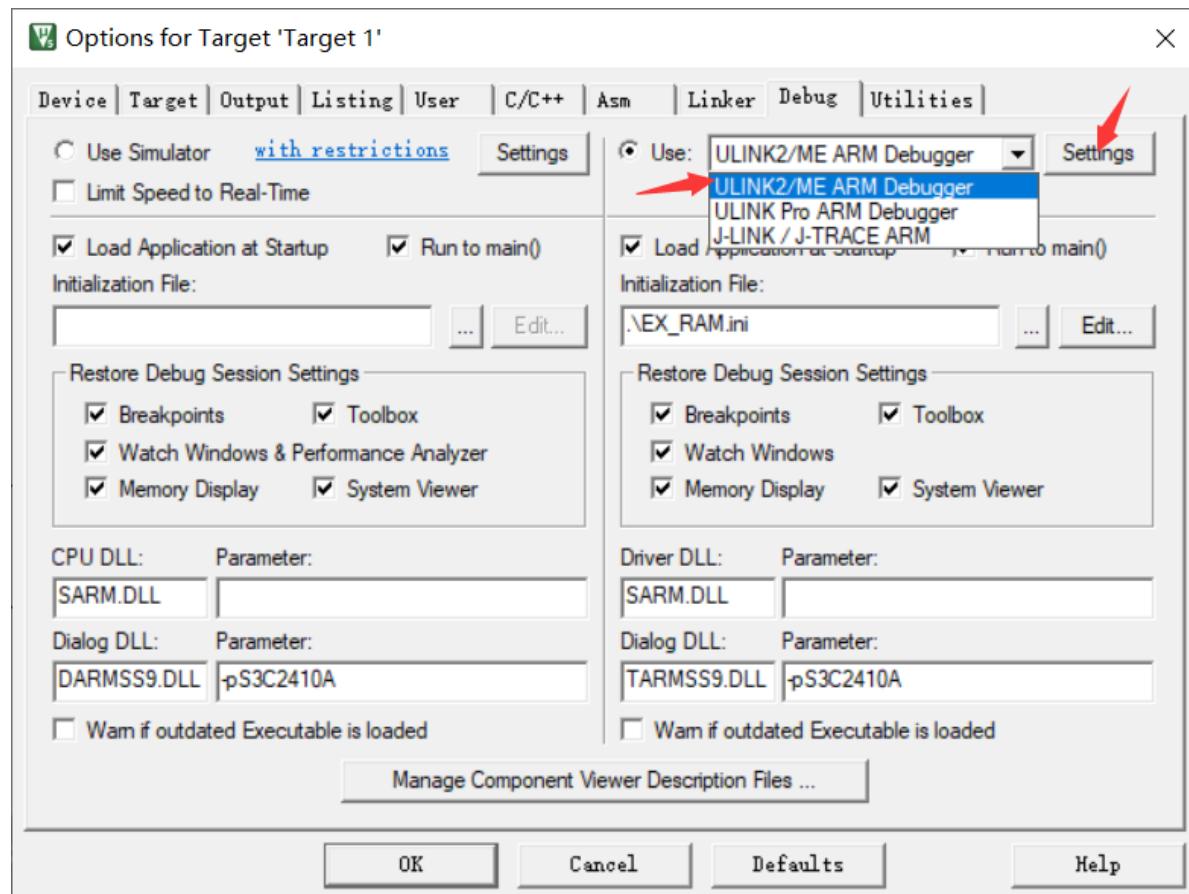
## 通过仿真器下载程序到RAM中运行，观察结果。

具体设计流程：

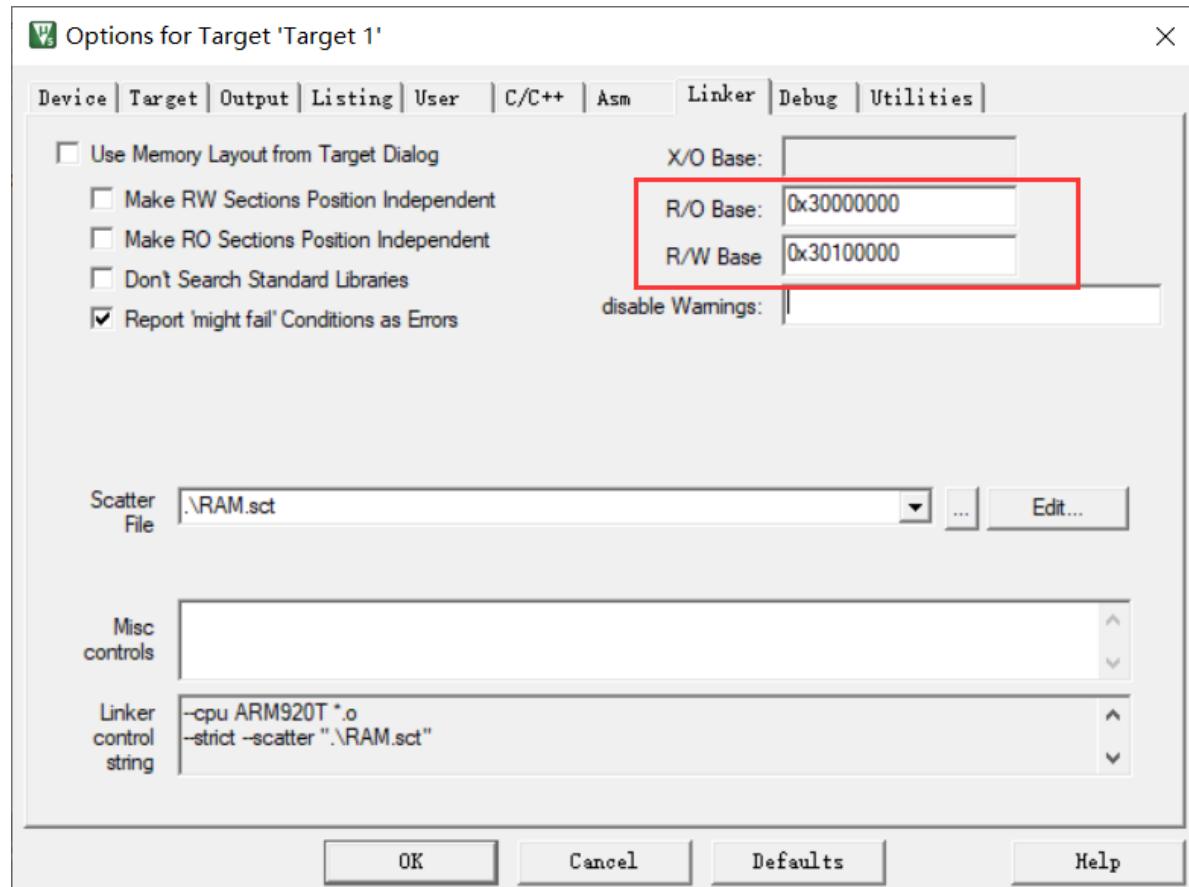
1. 将板子连接到电脑：



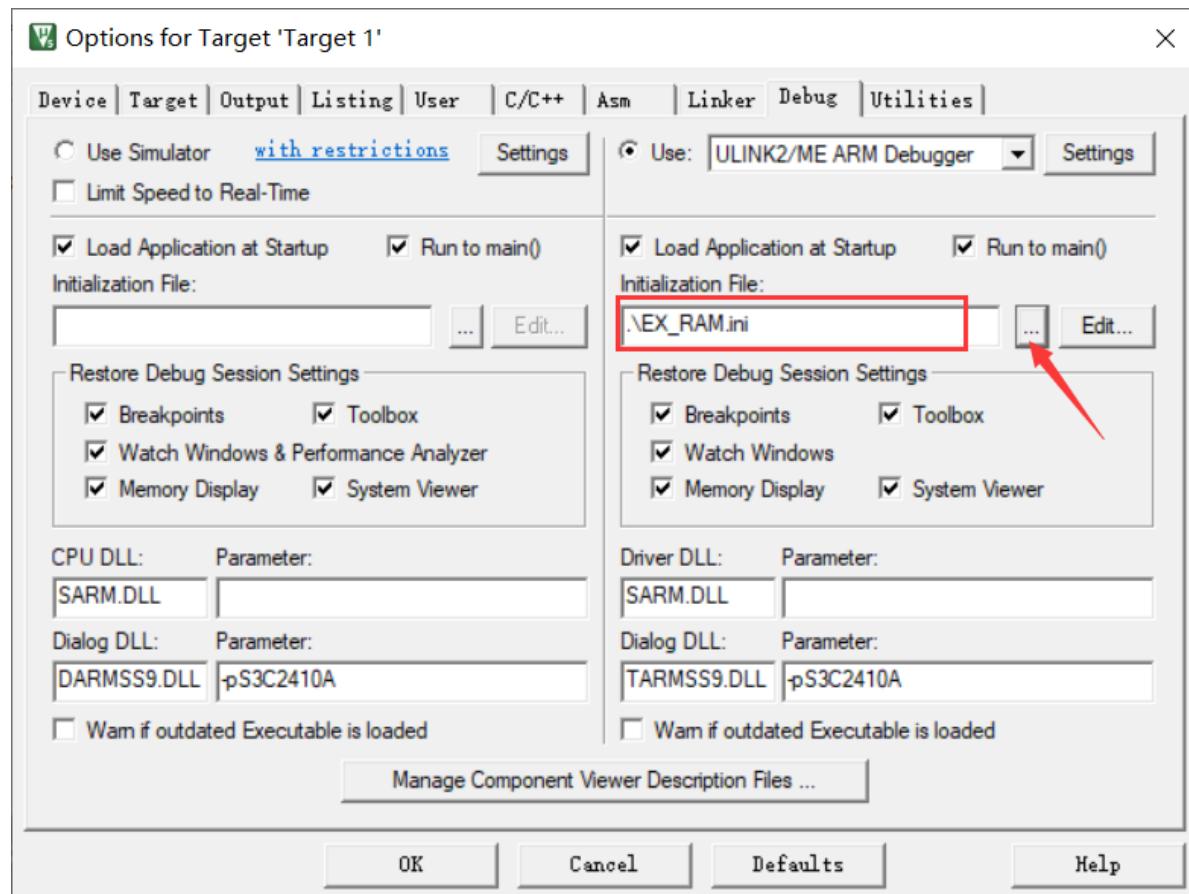
2. 配置Option -> Debug选项，选择ULINK2并在Setting中由连通显示



3. 设置Linker的R/O和R/W:



4. 加载Debug init

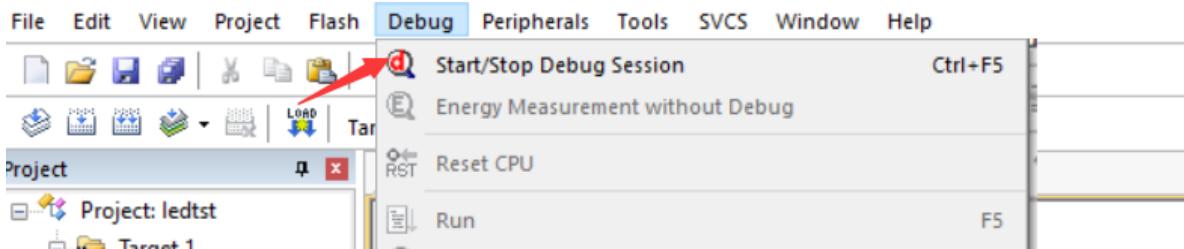


ini文件内容：

```
//map 0x48000000, 0x6000000 read write ;  
  
_WDWORD(0x53000000,0x00000000);  
_WDWORD(0x4a000008,0xffffffff);  
_WDWORD(0x4a00001c,0x000007ff);  
_WDWORD(0x4c000014,0x03);  
_WDWORD(0x4c000004,0x5c042);  
_WDWORD(0x56000070,0x00280000);  
_WDWORD(0x56000078,0x00000000);  
_WDWORD(0x48000000,0x22111110);  
_WDWORD(0x48000004,0x00000700);  
_WDWORD(0x48000008,0x00000700);  
_WDWORD(0x4800000c,0x00000700);  
_WDWORD(0x48000010,0x00000700);  
_WDWORD(0x48000014,0x00000700);  
_WDWORD(0x48000018,0x00000700);  
_WDWORD(0x4800001c,0x00018005);  
_WDWORD(0x48000020,0x00000700);  
_WDWORD(0x48000024,0x008e0459);  
_WDWORD(0x48000028,0x000000b2);  
_WDWORD(0x4800002c,0x00000030);  
_WDWORD(0x48000030,0x00000030);  
_WDWORD(0x56000014,0x01);  
_WDWORD(0x56000020,0xaaaa55aa);  
_WDWORD(0x56000028,0x0fff);  
_WDWORD(0x56000024,0x00000000);  
NOP  
NOP  
NOP
```

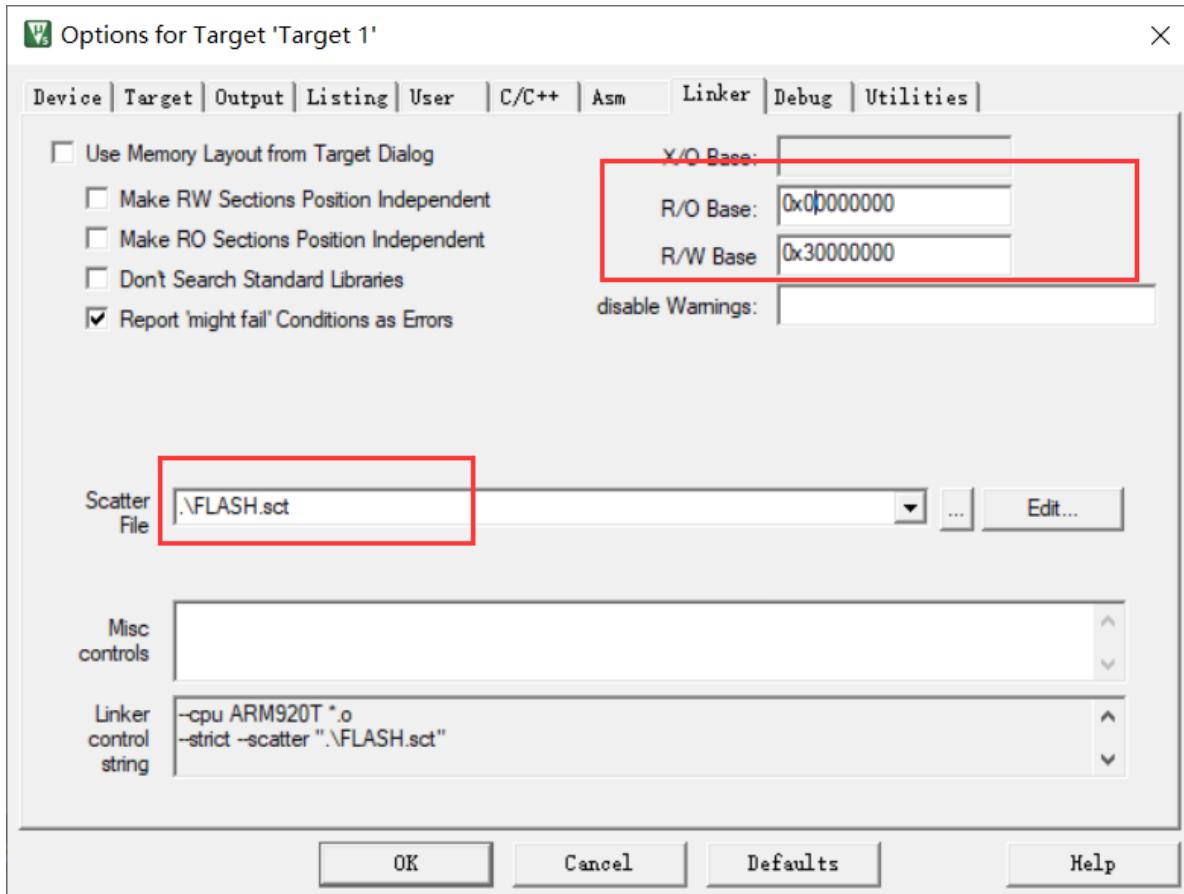
```
LOAD .\objects\ledtst.axf INCREMENTAL  
//pc=0x000  
pc=0x30000000
```

## 5. Debug以及Run

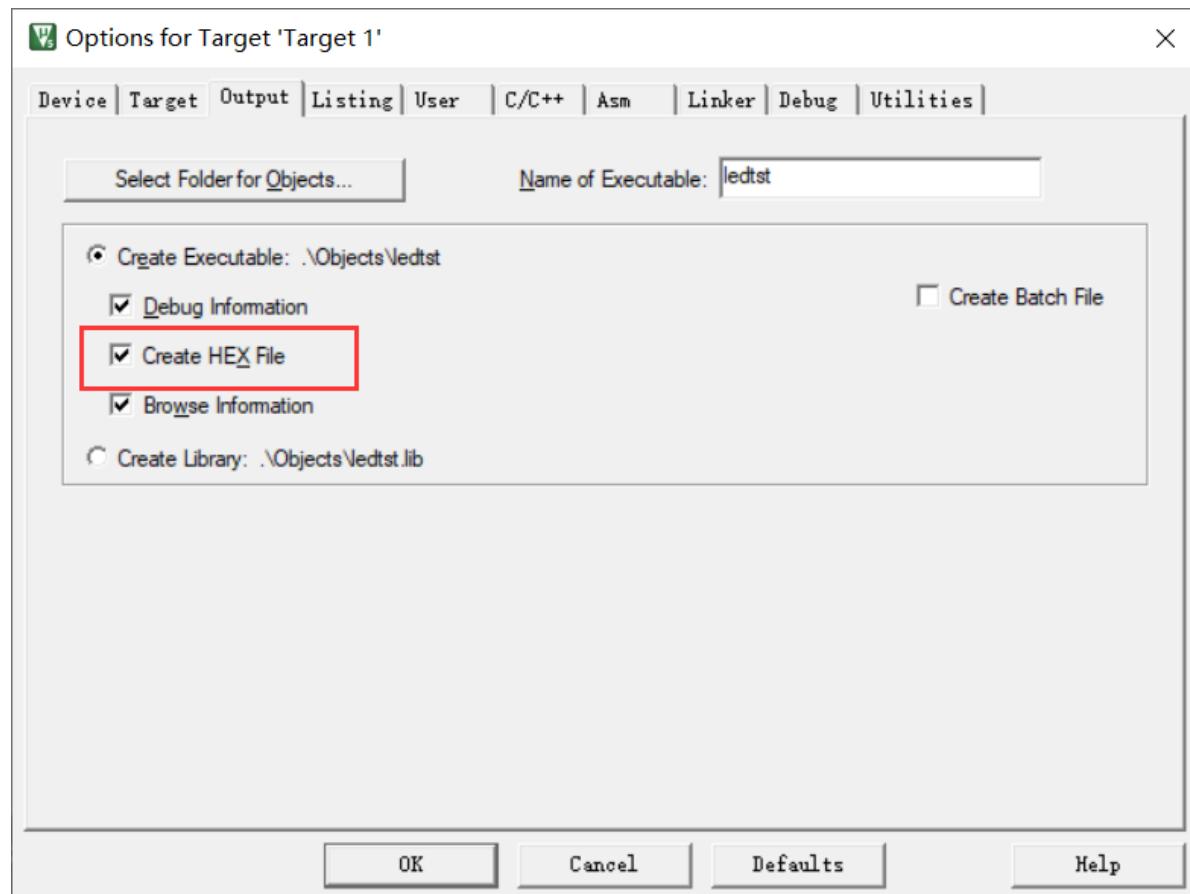


## 将程序写入板子NOR FLASH中，实现上电后自动启动。

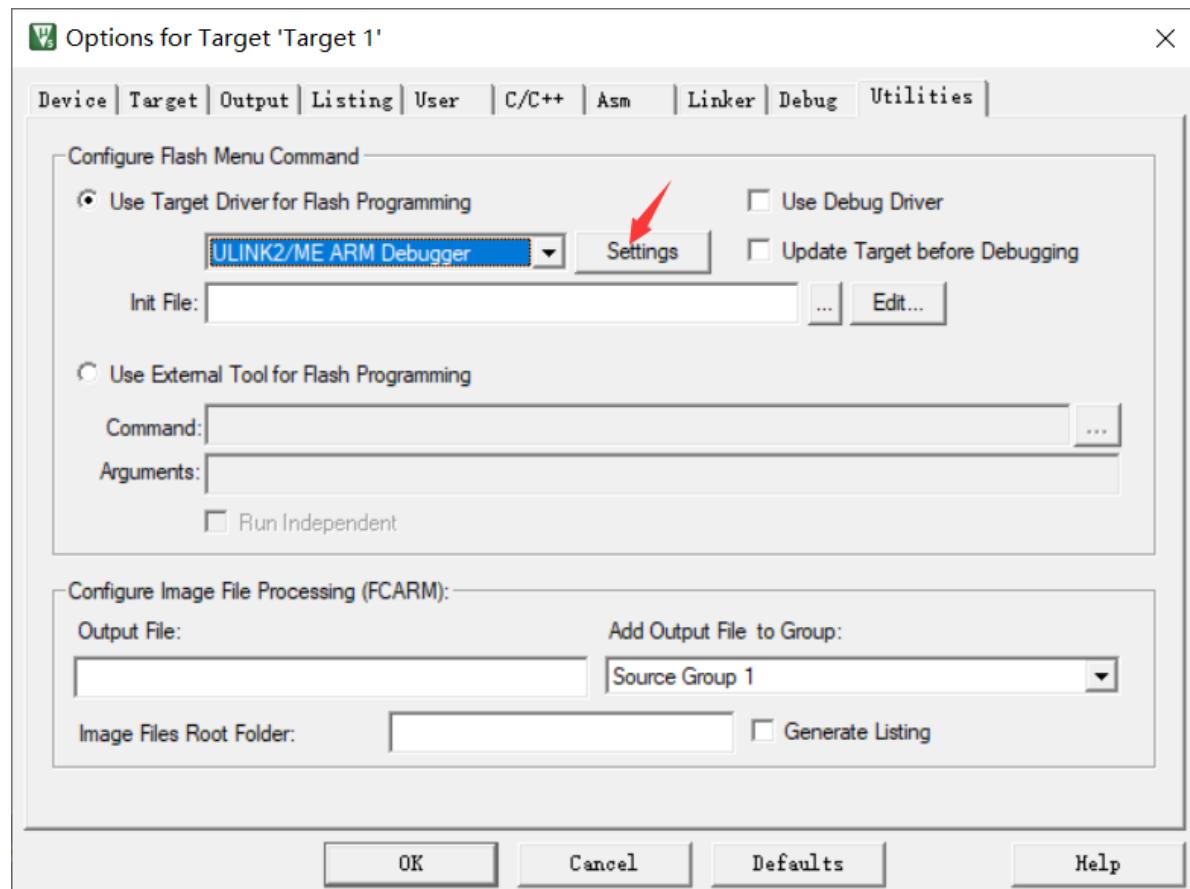
1. 设置Linker配置：配置RO和RW地址并选择FLASH.sct,

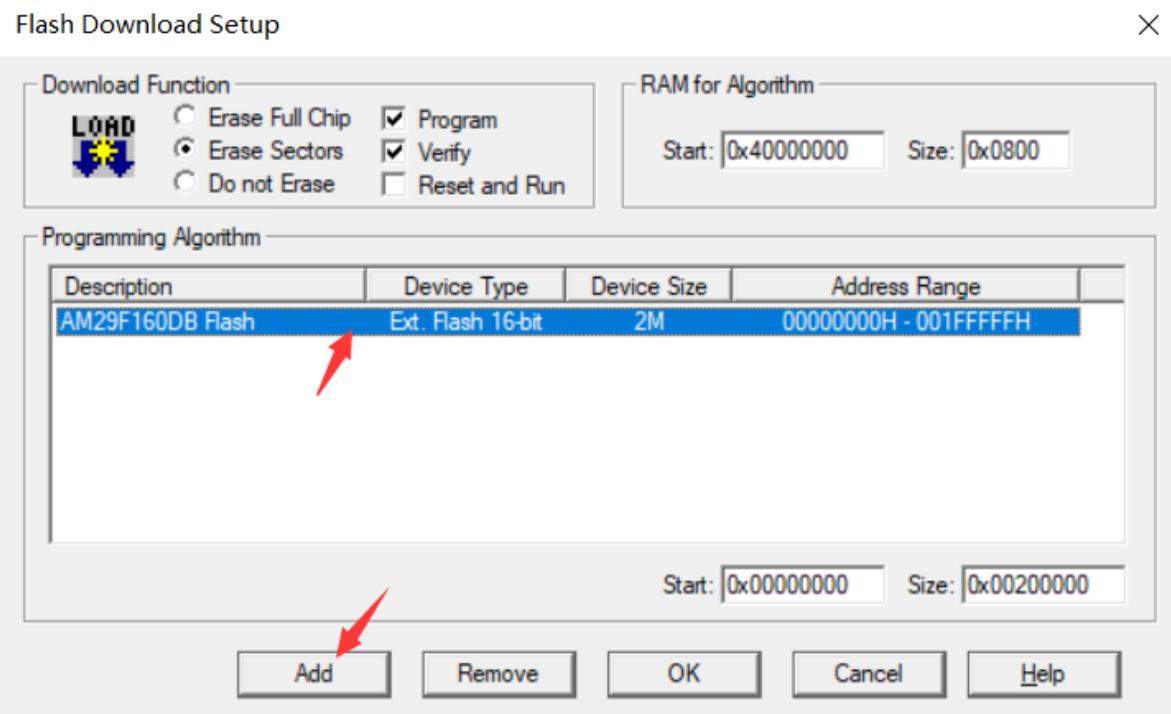


2. 输出创建Hex文件：

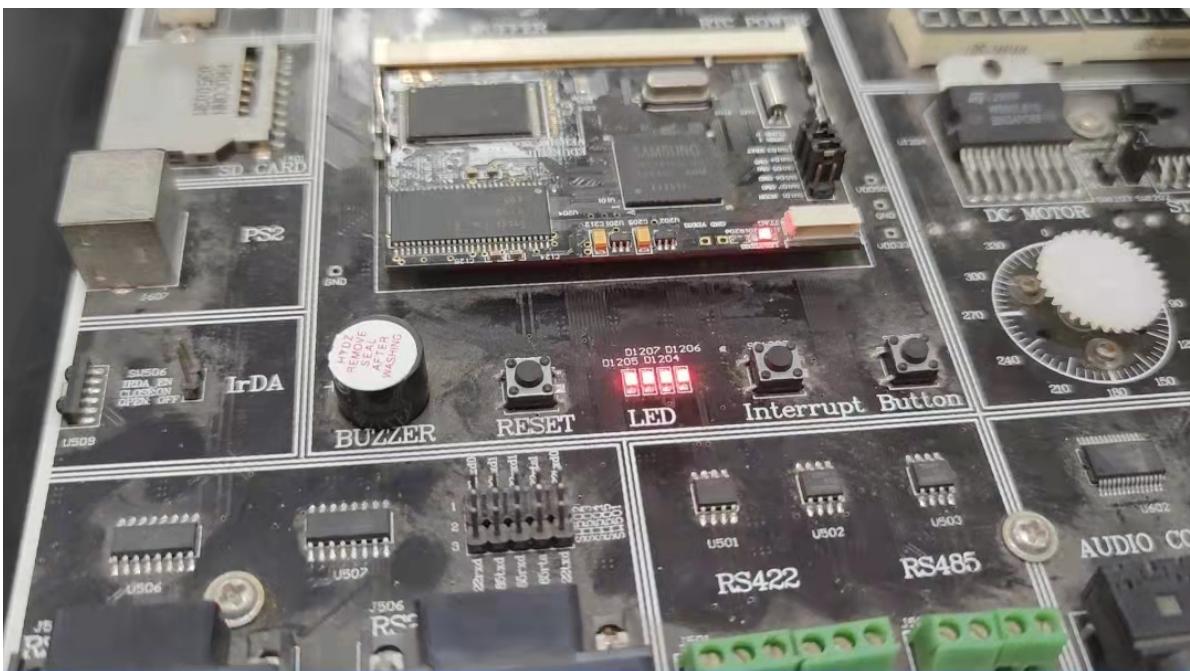


### 3. 配置Flash:





#### 4. Load到开发板中



## 改进

实现流水灯和其他样式的灯闪：

```
#include<stdlib.h>
#include<stdio.h>
int *rGPFCON = (int *) 0x56000050;
int *rGPFDAT = (int *) 0x56000054;
int delay(int times);

int main(void)
{
    *rGPFCON=0x5500;
```

```

while(1)
{
    int i;
    *rGPFDAT = 0x10;
    delay(500);

    for(i=1; i<=4; i=i+1){
        *rGPFDAT <<= 1;
        delay(500);
    }

    for(i=1; i<=4; i=i+1){
        *rGPFDAT >>= 1;
        delay(500);
    }

    /*
    *rGPFDAT = 0x00;
    delay(500);
    *rGPFDAT = 0x60;
    delay(500);
    *rGPFDAT = 0x90;
    delay(500);*/
}

return 1;
}

int delay(int times)
{
    int i,j;
    for(i=0;i<times;i++)
        for(j=0;j<times;j++)
    {

    }
    return 1;
}

```

## 遇到的问题

- 开发板上的LED不是按照序号顺序从左到右或从右到左排列的
- 有时候JAT无法通信，检查板子连接是否正常或板子是否老旧更换芯片试一试