

一、前期软件要求

需要预先安装如下软件：

- | | |
|-------------------------------|----------------------------|
| 1. MDK522 | KEIL5.22 安装软件 |
| 2. MDKCM522 | KEIL LEGACY 安装软件，兼容 5 以前版本 |
| 3. Keil.TM4C_DFP.1.1.0 | TM4C 芯片及板的 DFP 安装包 |
| 4. SW-EK-TM4C1294XL-2.1.4.178 | TM4C1294XL 驱动及样例程序 |

默认安装完成后，有两个目录

- | | |
|--|--------------|
| 1. C:\Keil_v5 | KEIL 可执行文件目录 |
| 2. C:\ti\TivaWare_C_Series-2.1.4.178 | TIVA 系列驱动及样例 |

二、硬件要求

WIN7 及以上操作系统，2G 内存

TM4C1294XL 板及 TM4C1294XL_SUBBOARD 组合板，即 S800 板

Micro-USB 数据线一根

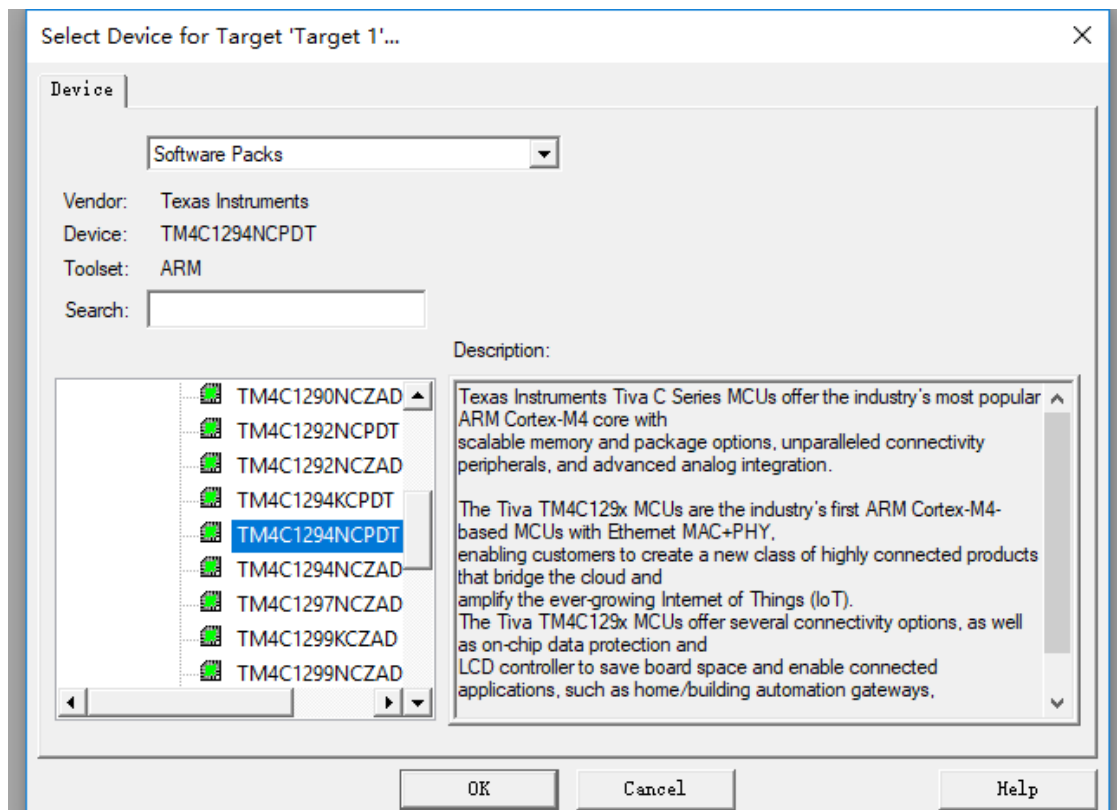
三、新建用户目录设为 [C:\S800\EXP1](#)

CPU 为 TM4C1294NCPDT

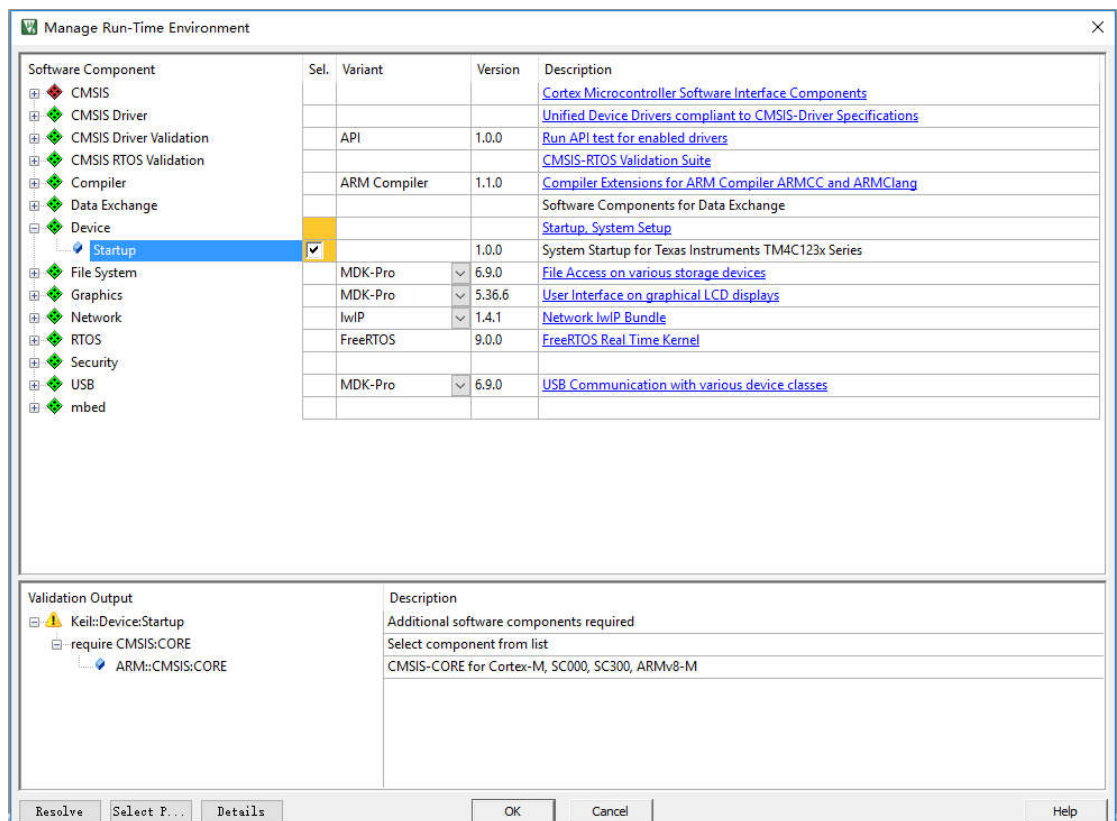
因为需要使用 TM4C1294 芯片的硬件定义以及固件库，因此从 [C:\ti\TivaWare_C_Series-2.1.4.178](#) 中将 *INC* 及 *DRIVERLIB* 两个子目录拷贝到用户目录中。

四、STEP-BY-STEP 建立一个项目

1. 将 MICRO-USB 数据线一端接电脑，一端接 TM4C1294XL 的数据口（非网口端）
2. 打开 KEILuVision5，Project→New uVisionProject，新建一个项目。选择目录 [C:\S800\EXP1](#) 后，建立新项目 exp1。选择芯片如下图：



3. 让 KEIL 代我们生成启动代码，在 DEVICE 中 STARTUP 打勾。



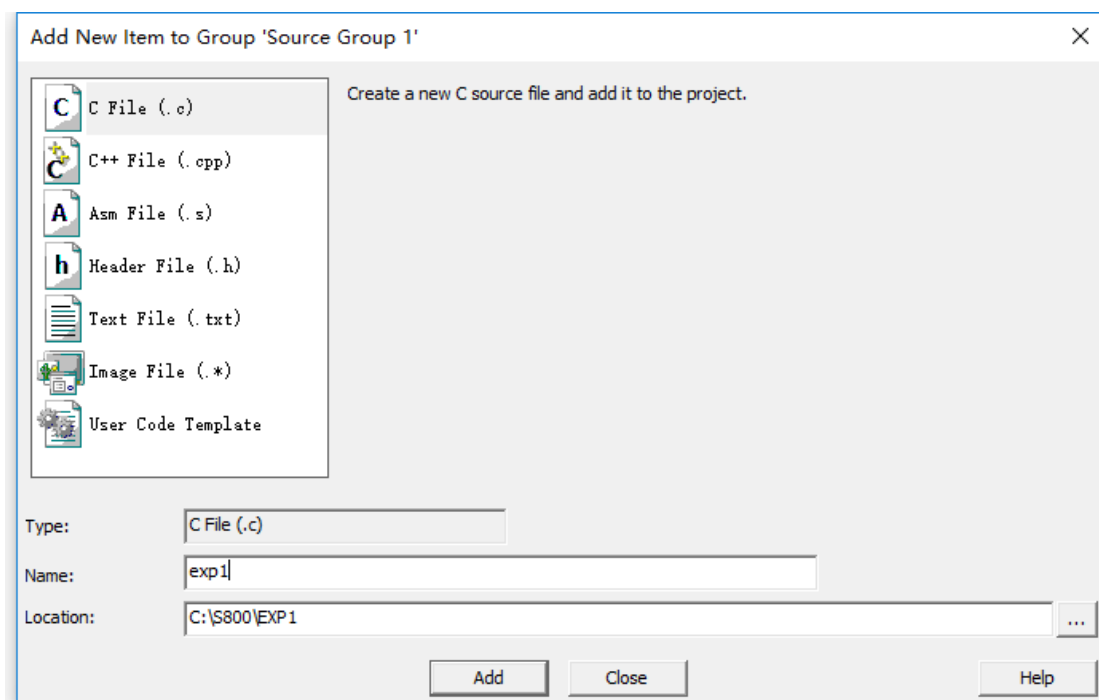
4. 这样系统自动生成一个项目，项目中包括了一个 Source Group 1，用来放置源文件，目前为空；以及一个设备目录 Device，包括 Startup_TM4C129.s 以及 System_TM4C129.c 两个文件。

Startup_TM4C129.s 配置了堆栈和中断函数名称以及从复位到 main 函数前的处理过程

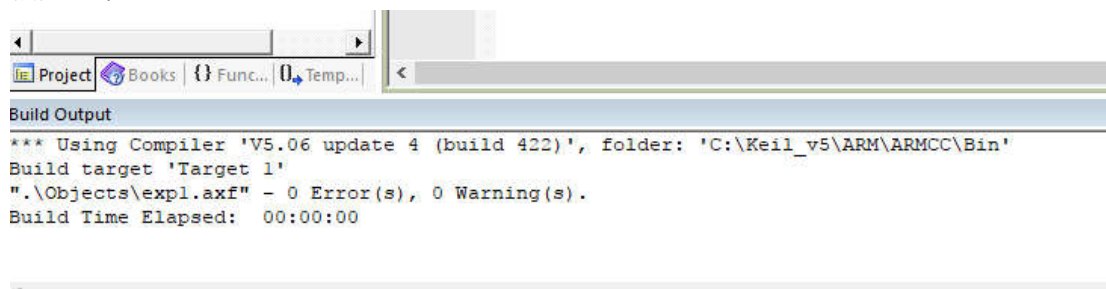
System_TM4C129.c 默认系统初始化，配置了默认时钟

5. 单点 Source Group 1，右键，选择 Add New Item to Group "Source Group 1"如下图所示，生成一个源文件 exp1.c。并完成一个最简单的主函数

```
int main(void)
{
    while (1);
}
```



6. 这样建立了一个最简单的项目，可以试着编译一下 Project→Build Target，可以看到应该是无错误。



7. 现在我们进行一个最简单的闪烁 LED 项目。

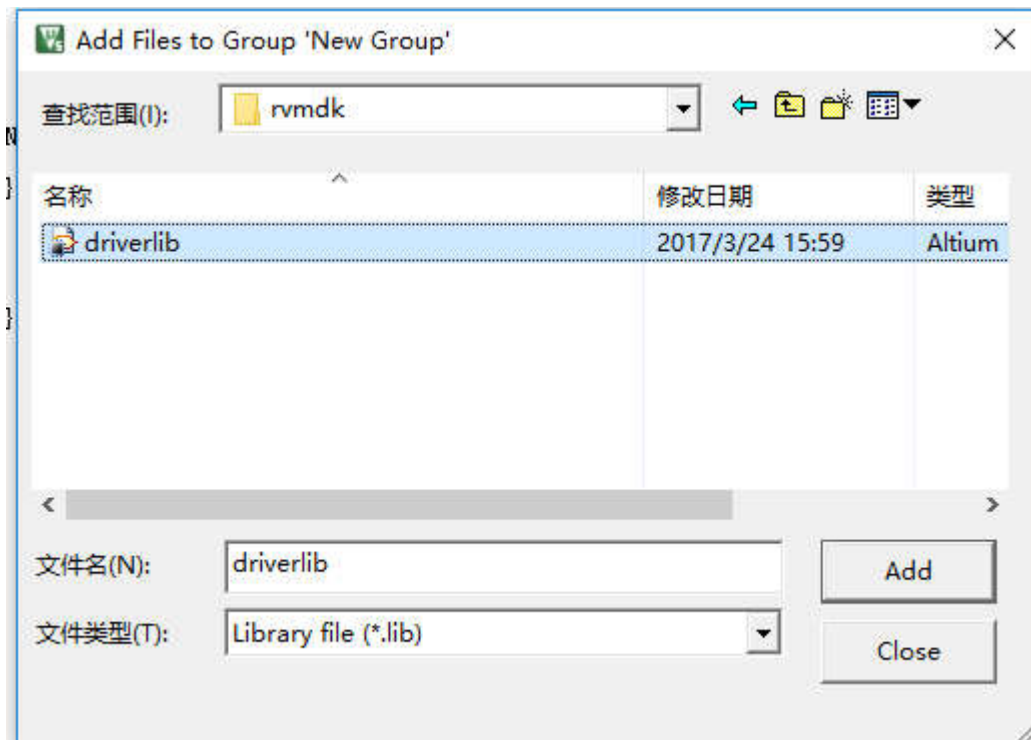
资源	管脚名称	LED 有效电平
	PF0	高

首先对项目进行进一步配置。

将固件库文件添加到项目中。

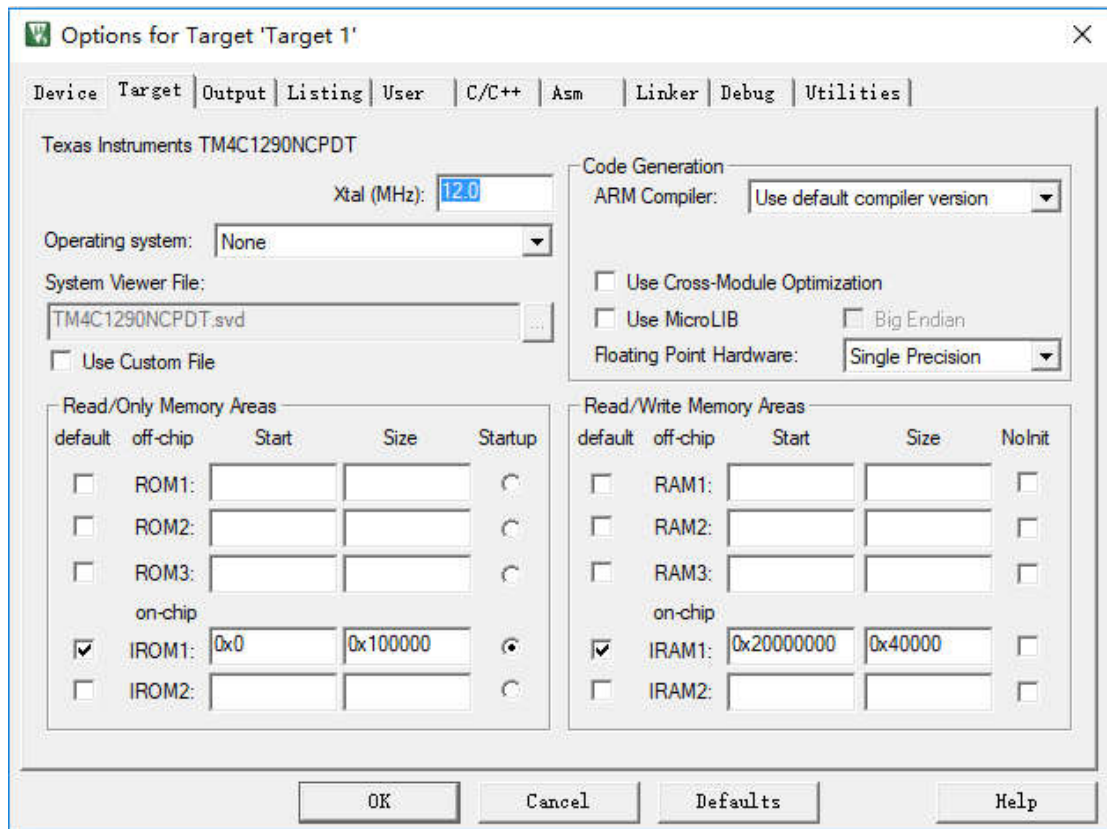
点在 Target 1, 右键, Add Group,

点在 New Group, 右键, Add Existing File to Group "New Group", 将 driverlib.lib 库文件添加到此组中。如下图所示。注意选择文件类型为 LIB。

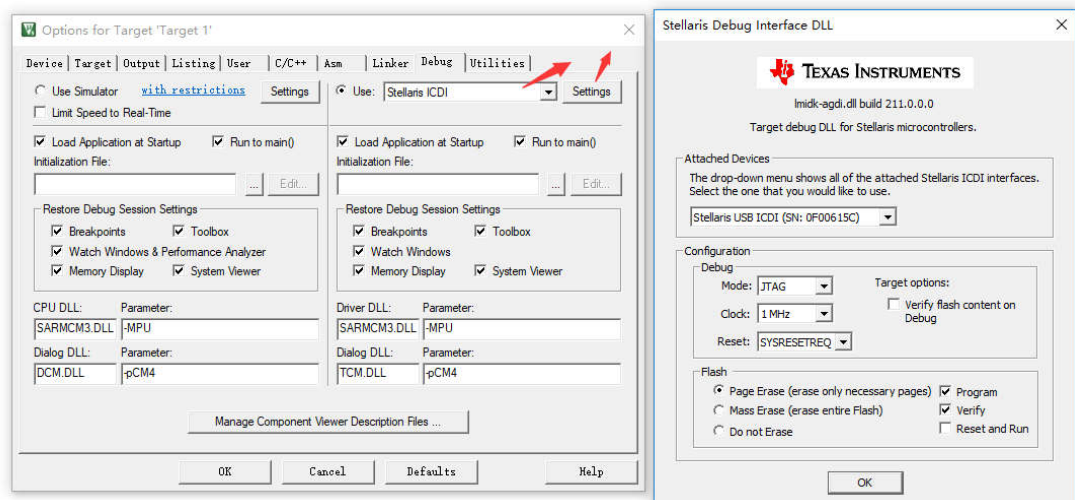


点在 Target 1，右键，选择 Option for Target “Target 1”，需要变动的项如下表：

栏目名称	动作	说明
Output	勾选 creat HEX File	生成供 uniflash 使用的文件
Debug	Use Stellaris ICDI	在线 DEBUG 仿真器
C/C++	Preprocessor Symbols-Define PART_TM4C1294NCPDT	CPU 型号预定义，因为 driverlib 中某些头文件需要根据 CPU 类型进行不同预定义
C/C++	.inc;\driverlib	指定头文件目录



如果一切正常，应该出现如下图所示，表示仿真器已经连接正常。否则说明数据线或实验板有问题。



8. 将 exp1.c 文件改成如下所示

```
#include <stdint.h>
#include <stdbool.h>
#include "hw_memmap.h"
#include "debug.h"
#include "gpio.h"
#include "hw_types.h"
#include "pin_map.h"
```

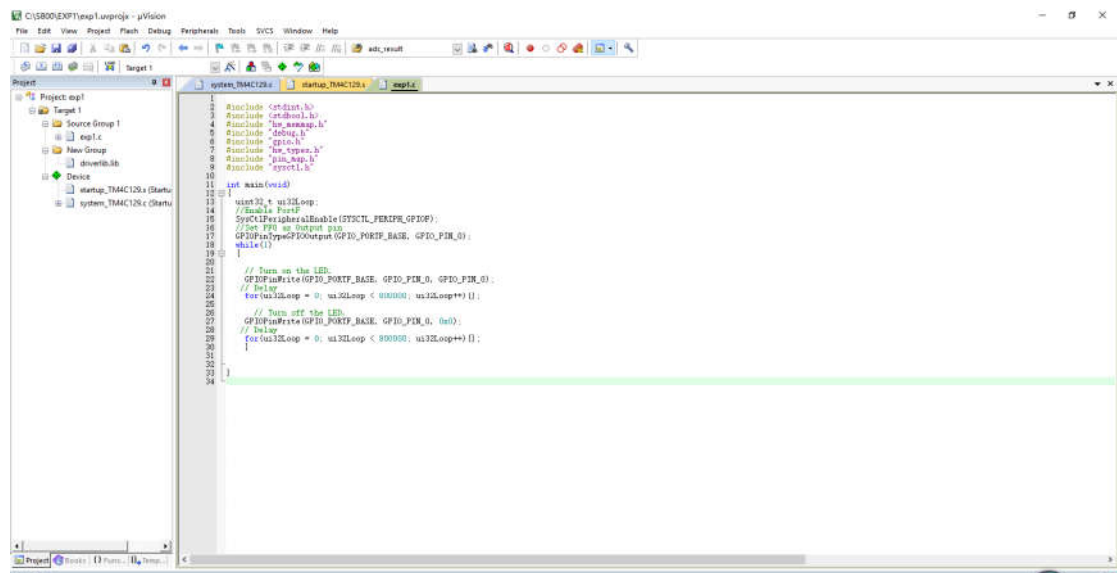
```

#include "sysctl.h"

int main(void)
{
    uint32_t ui32Loop;
    //Enable PortF
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    //Set PF0 as Output pin
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_0);
    while(1)
    {
        // Turn on the LED.
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_0, GPIO_PIN_0);
        // Delay
        for(ui32Loop = 0; ui32Loop < 800000; ui32Loop++){};

        // Turn off the LED.
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_0, 0x0);
        // Delay
        for(ui32Loop = 0; ui32Loop < 800000; ui32Loop++){};
    }
}

```



9. Project→Build Target, 没有错误
10. Flash→Download, 将程序下载到 MCU 中。
11. 按板上 RESET 键, 程序运行。子板及母板上的 PF0 灯均闪烁。
- 12.