1. 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
3. 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

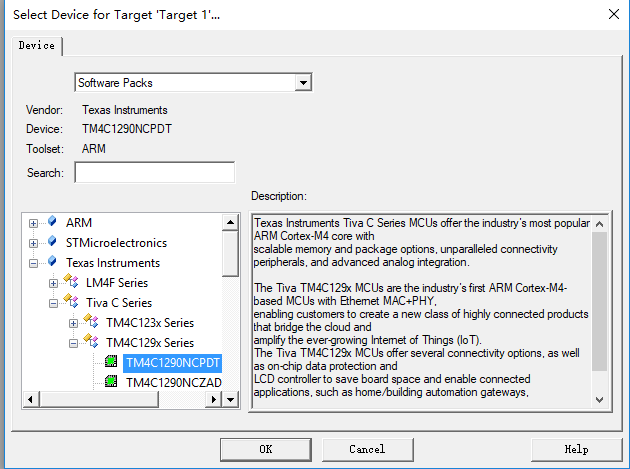
Micro-USB数据线一根

1. 新建用户目录设为*C:\S800\EXP1*

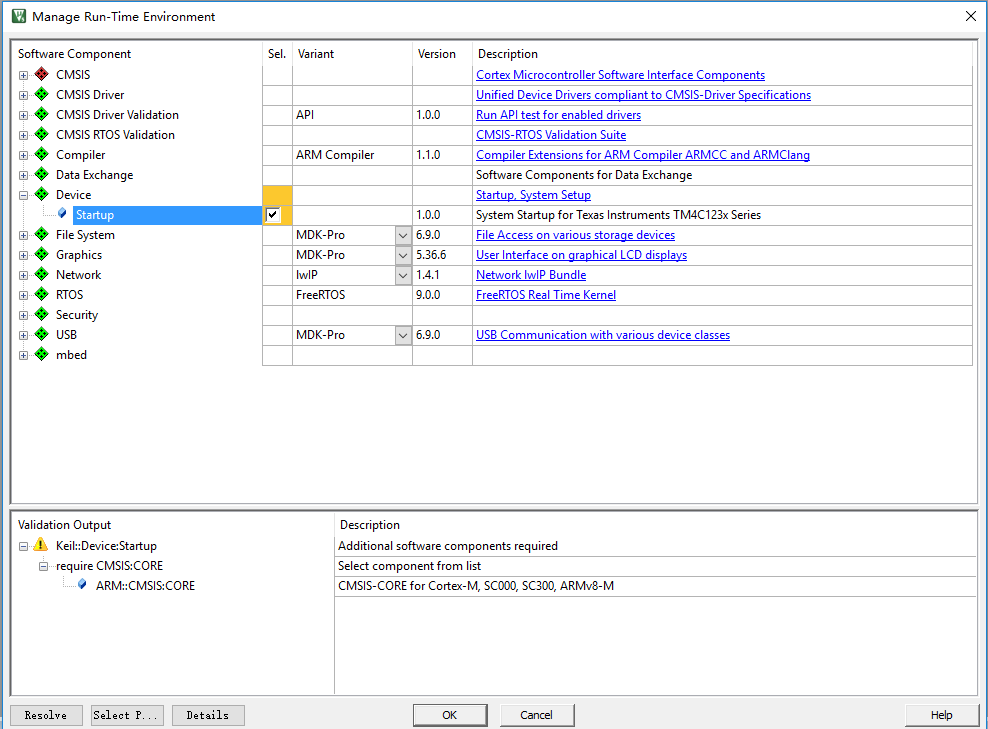
CPU为TM4C1294NCPDT

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。

1. STEP-BY-STEP建立一个项目
2. 将MICRO-USB数据线一端接电脑，一端接TM4C1294XL的数据口（非网口端）
3. 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：**///// TNM4C1294NCP ////////**



1. 让KEIL代我们生成启动代码，在DEVICE中STARTUP打勾。



1. 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

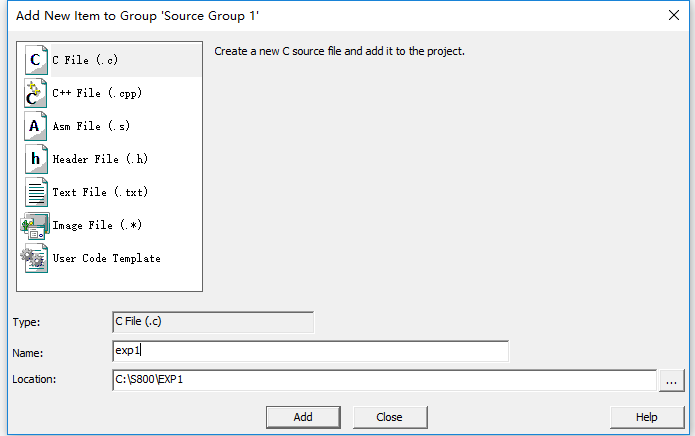
1. 单点Source Group 1，右键，选择Add New Item to Group “Source Group 1”如下图所示，生成一个源文件exp1.c。并完成一个最简单的主函数

int main(void)

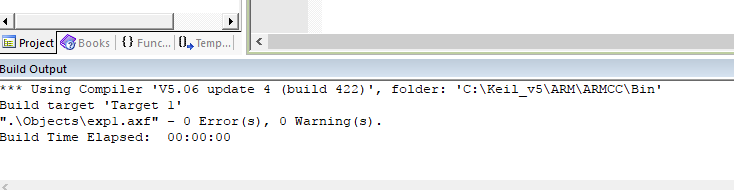
{

while (1);

}



1. 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。



1. 现在我们进行一个最简单的闪烁LED项目。

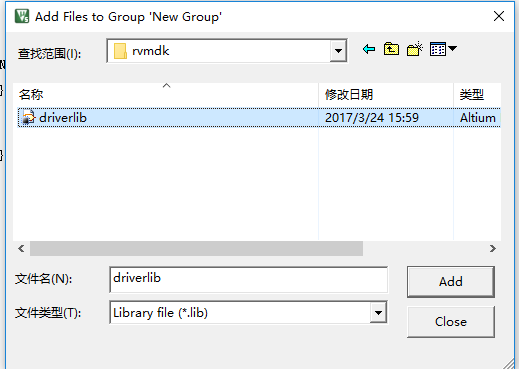
|  |  |  |
| --- | --- | --- |
| 资源 | 管脚名称 | LED有效电平 |
|  | PF0 | 高 |

首先对项目进行进一步配置。

将固件库文件添加到项目中。

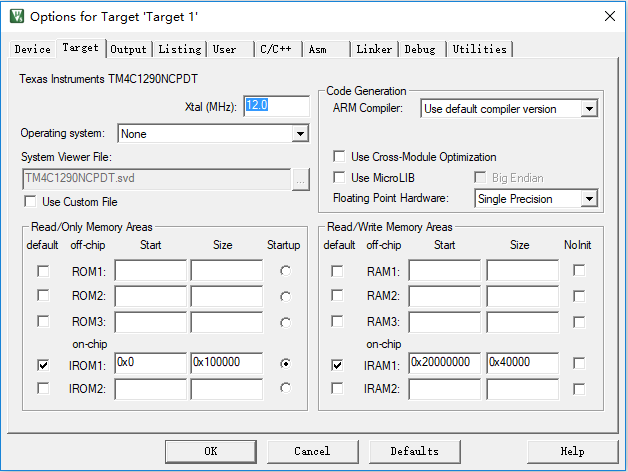
点在Target 1, 右键，Add Group，

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。**

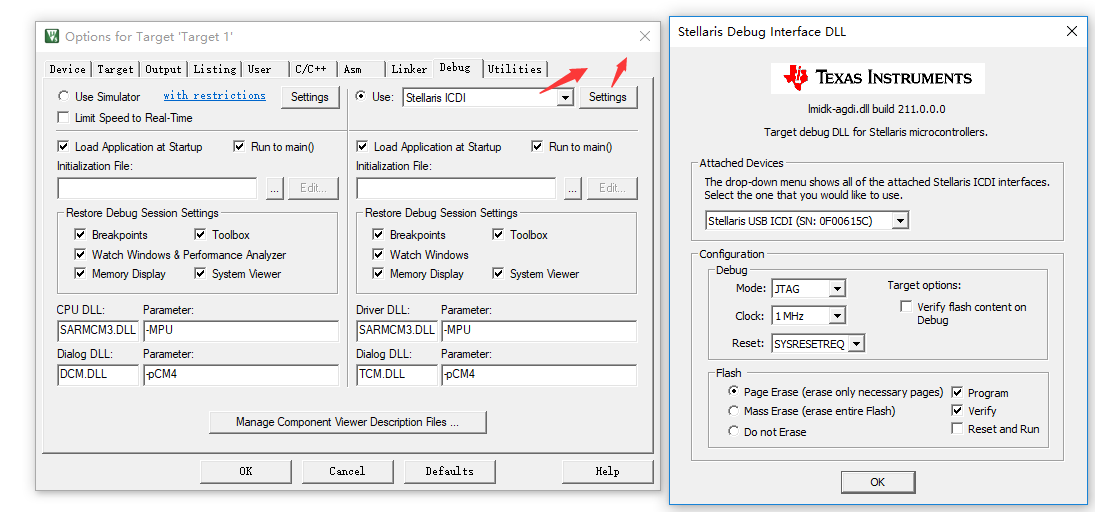


点在Target 1，右键，选择Option for Target “Target 1”，需要变动的项目如下表：

|  |  |  |
| --- | --- | --- |
| 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++  （自己打Define） | 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。



1. 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.

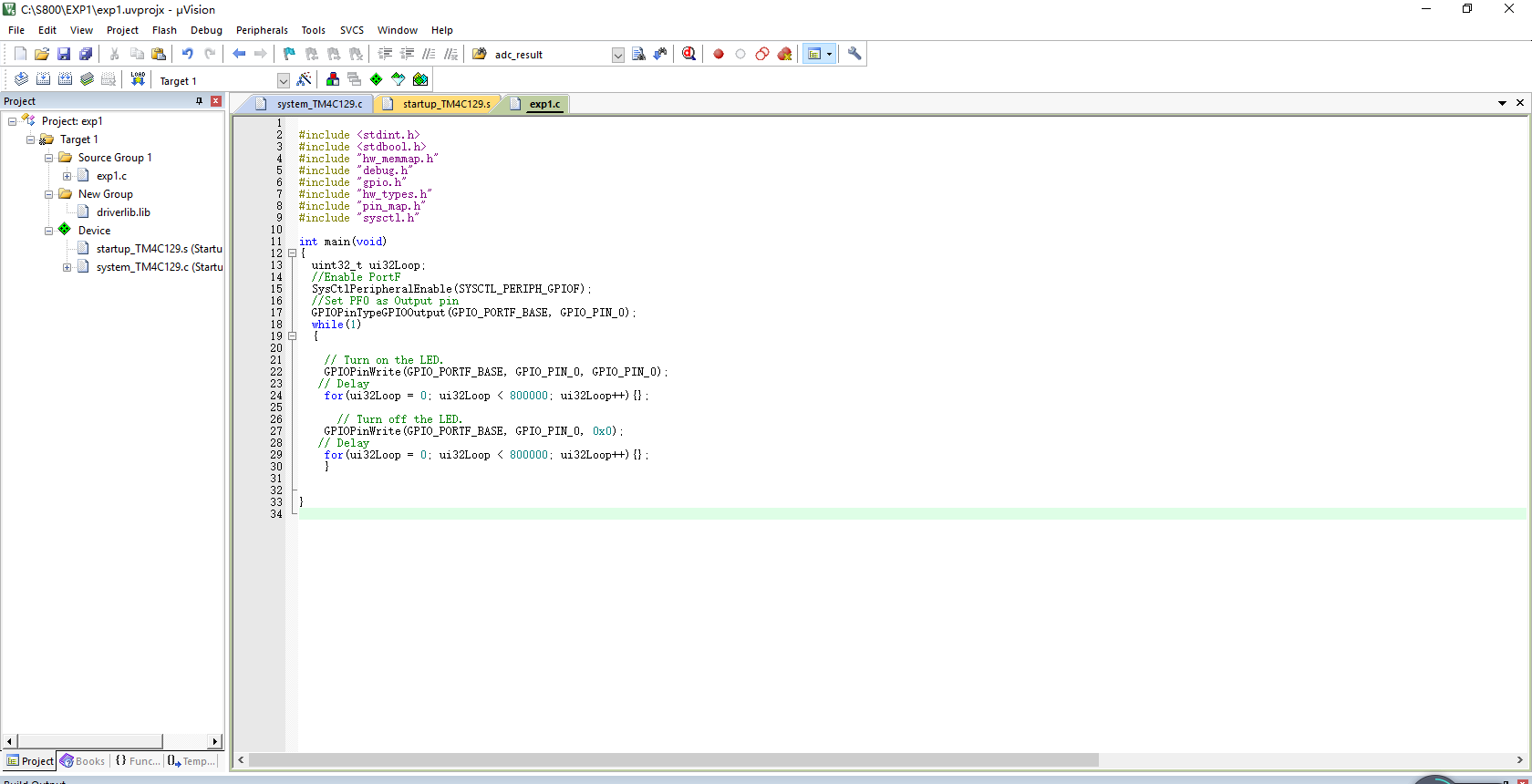
GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_0, 0x0);

// Delay

for(ui32Loop = 0; ui32Loop < 800000; ui32Loop++){};

}

}



1. Project🡪Build Target，没有错误
2. Flash🡪Download，将程序下载到MCU中。
3. 按板上RESET键，程序运行。子板及母板上的PF0灯均闪烁。