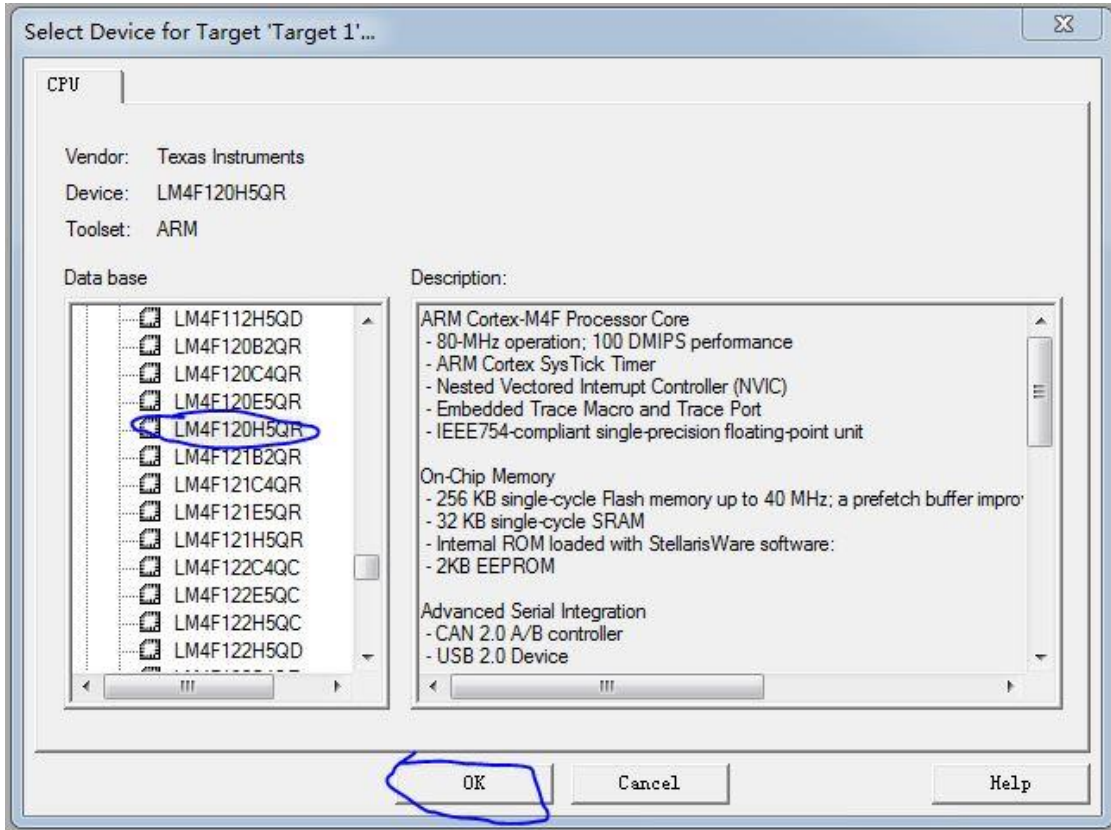


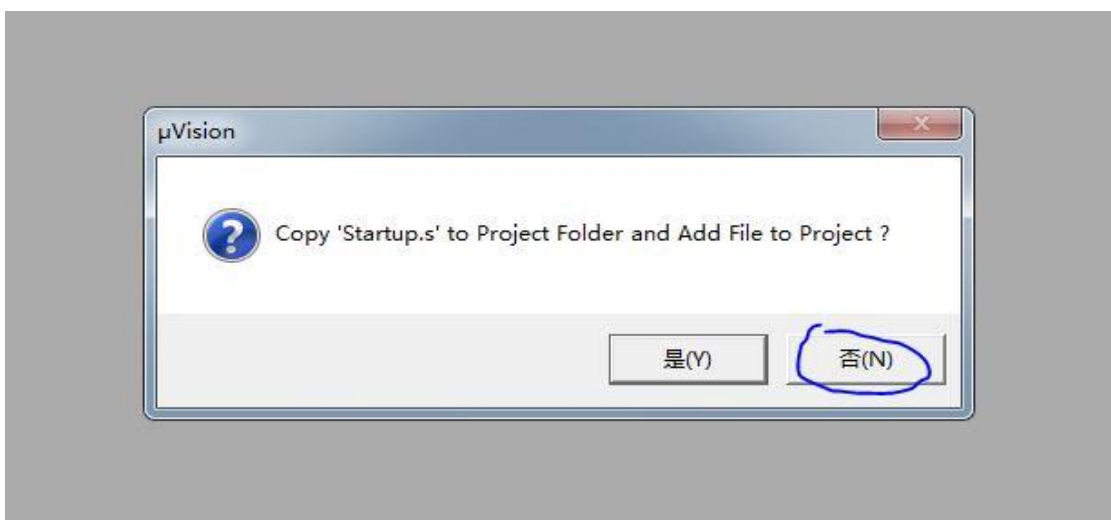
# 用keil mdk 作为TM4C123G Evaluation kit 的开发平台

Lengmi\_HUST

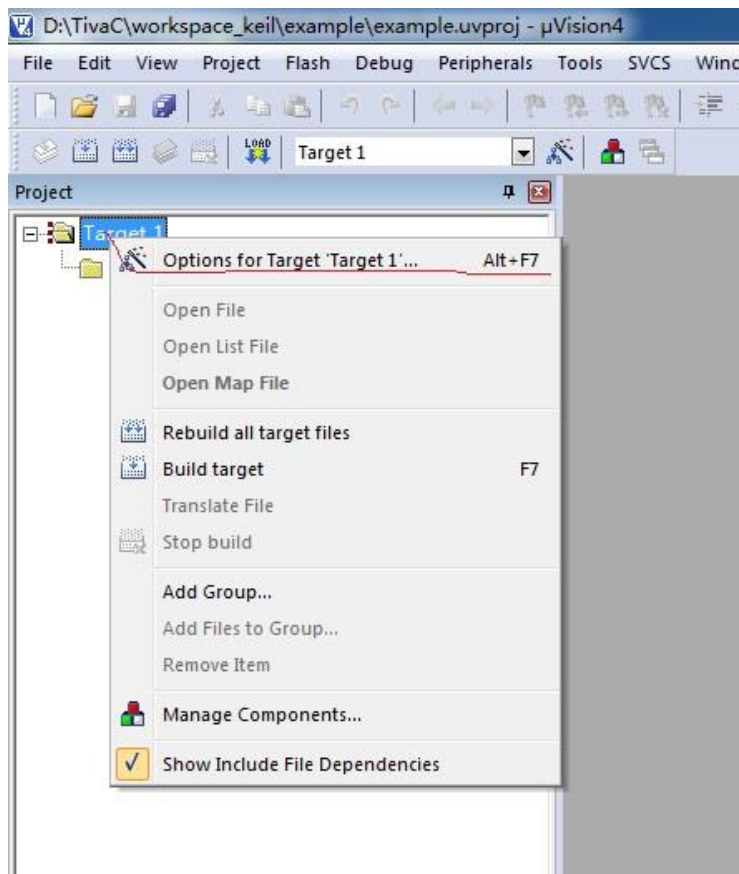
1、新建工程，选择器件，TM4C123G 是 LM4F120H5QR 的改进版



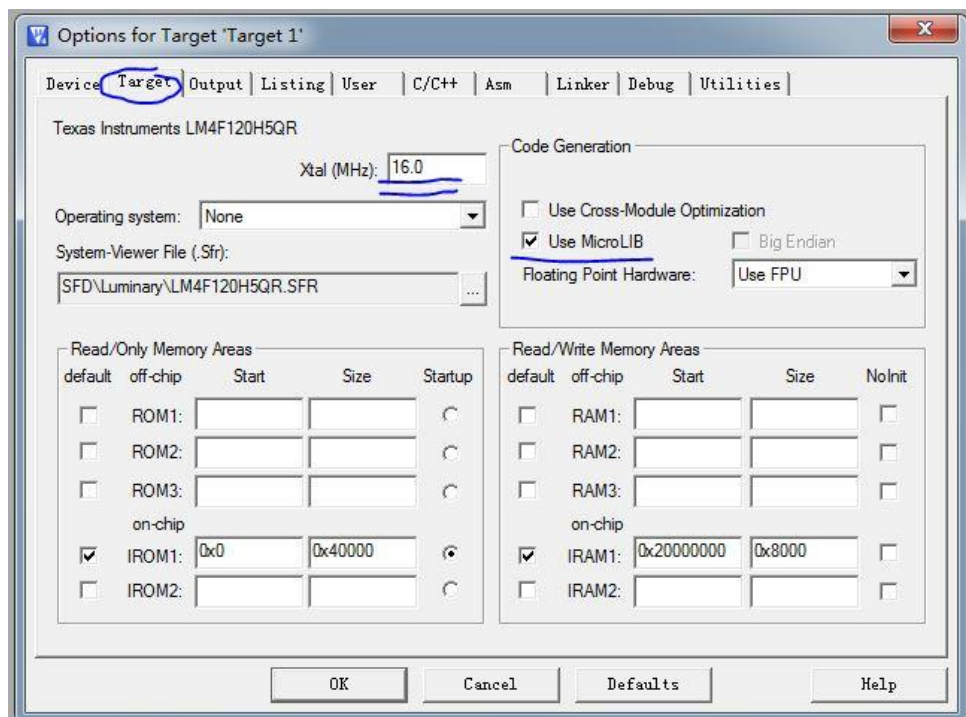
跳过添加 startup.s 文件，startup.s 可以从 TI 获得，也可以从附录中的代码中拷贝创建



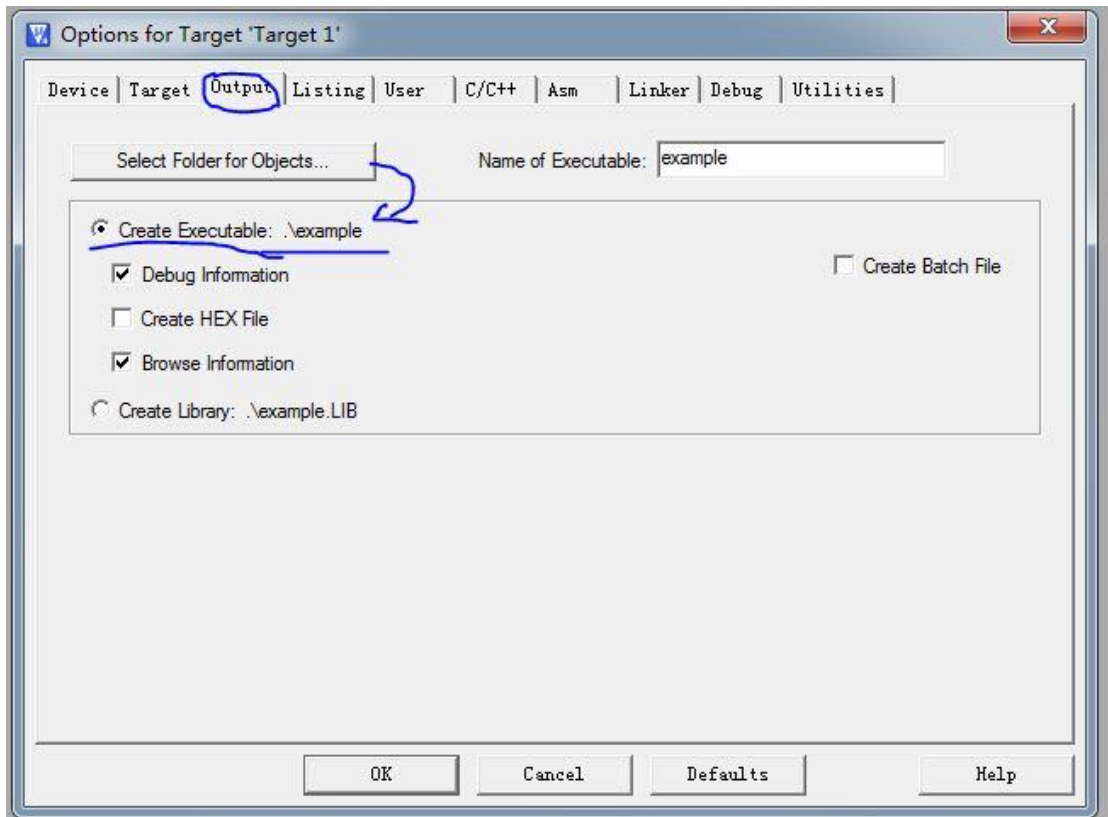
## 2、配置



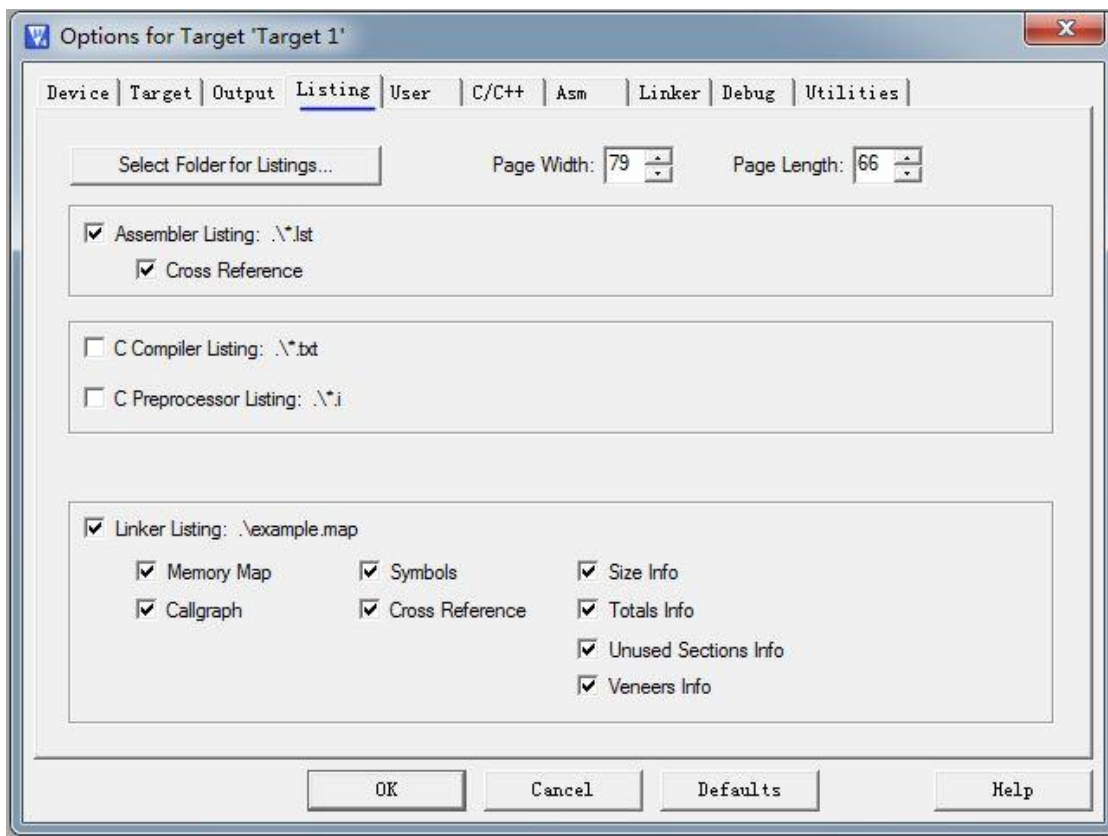
### 2.1 Target:



2.2 Output:可以更改 Output 文件的路径，output 中有很多编译过程产生的中间文件，可以在工程目录下创建一个新目录作为输出文件的路径，应该注意在配置 user 时应该把 Run 中的路径改为与这里一致。



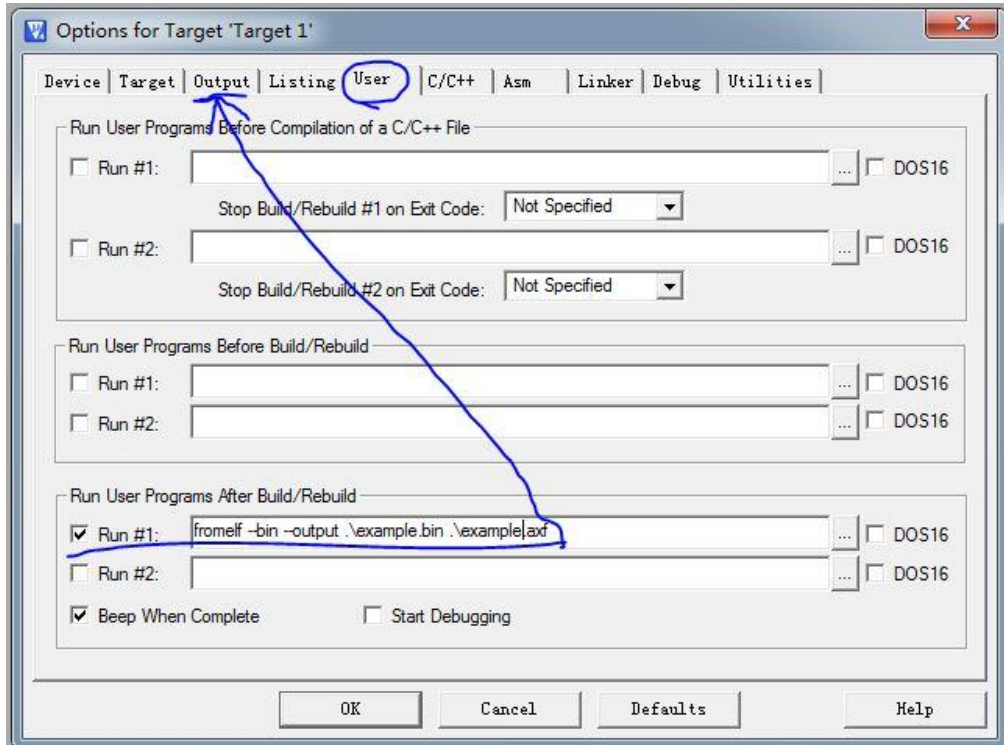
2.3 listing



## 2.4 user: fromelf --bin --output .\example.bin .\example.axf

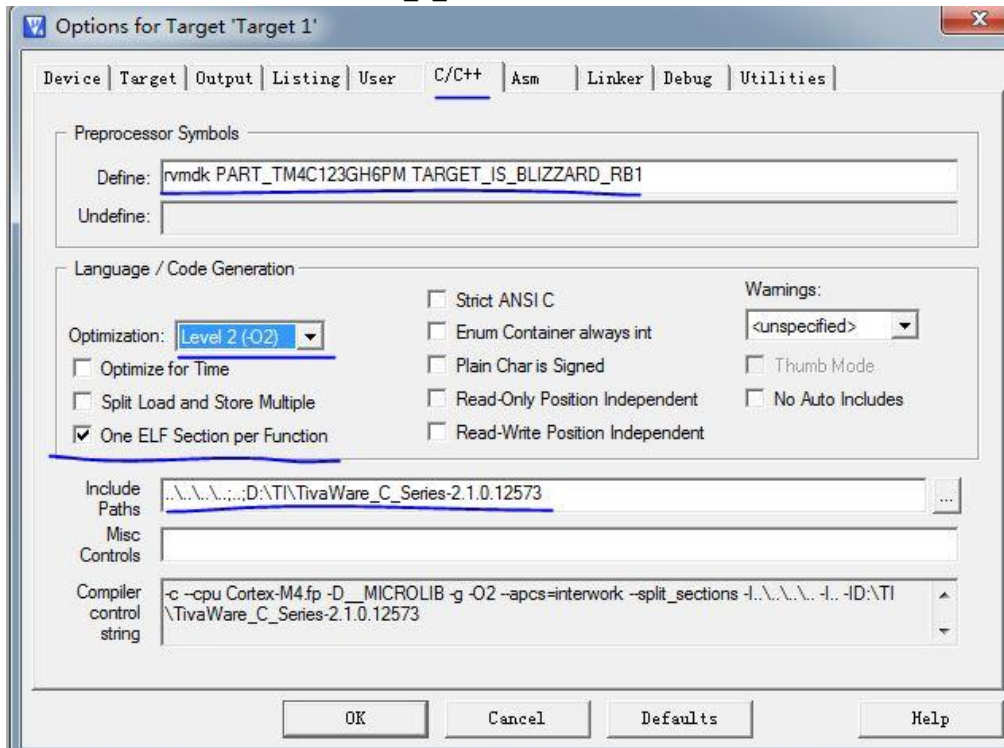
.\example.axf 是工程所在路径下 example.axf 文件的意思，这里的路径要与 2.2 选择的路径一致，如果你在工程路径中新建一个 output 文件用来接收编译产生的中间文件，那么应改为：

fromelf --bin --output .\example.bin .\output\example.axf

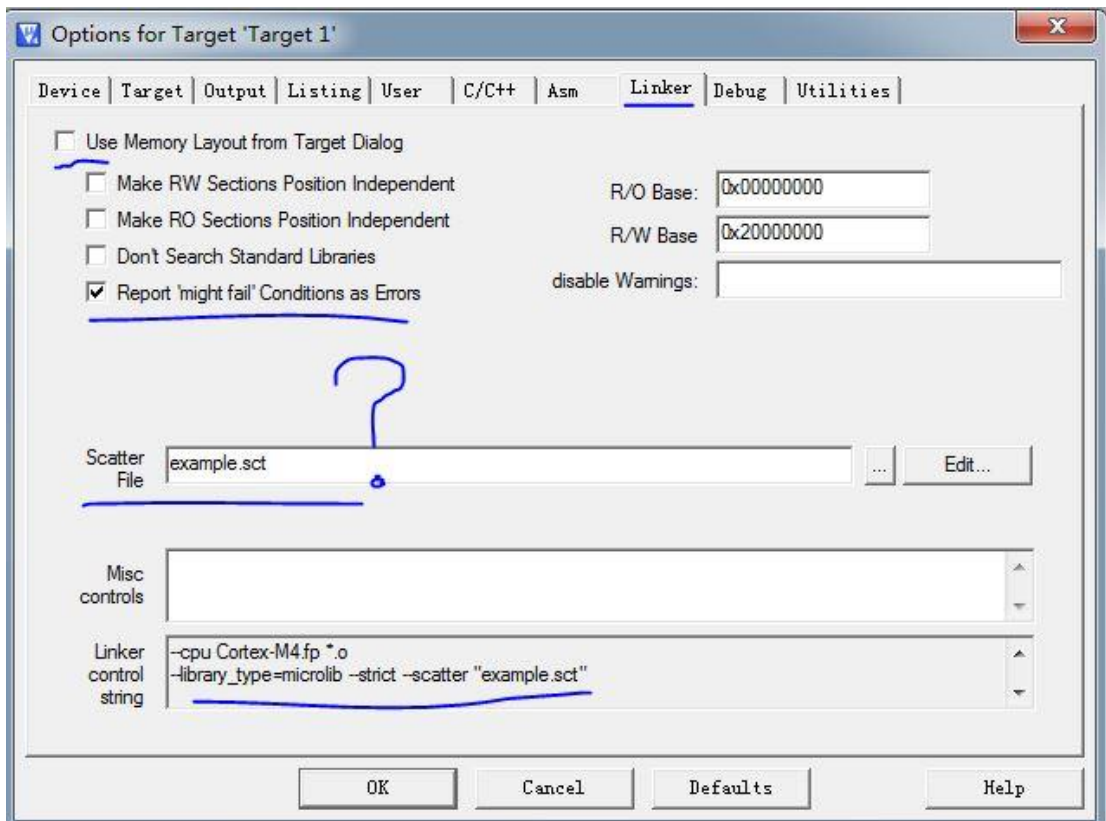


## 2.5 C/C++: rvmrk PART\_TM4C123GH6PM TARGET\_IS\_BLIZZARD\_RB1

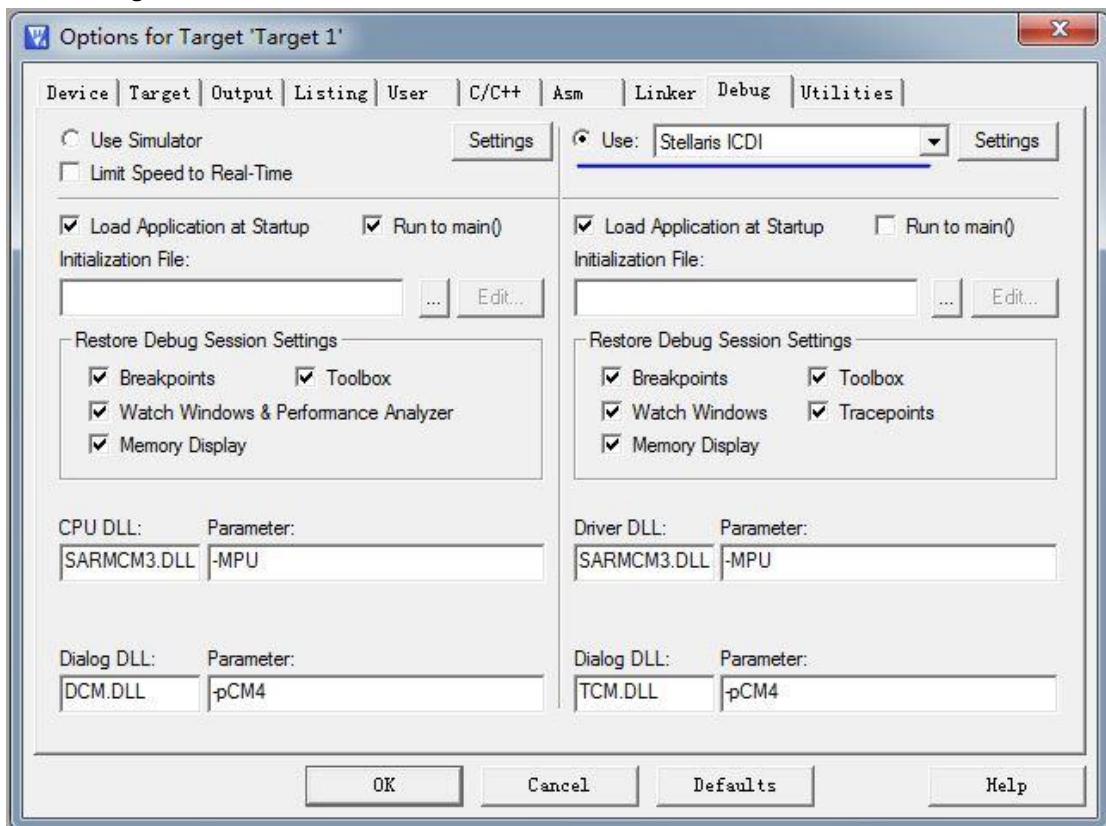
..\..\..\..\;D:\TI\TivaWare\_C\_Series-2.1.0.12573(改为您的 Tivaware 的安装路径)



## 2.6 Linker: sct 文件需要自己创建, 详见附录, 具体 google

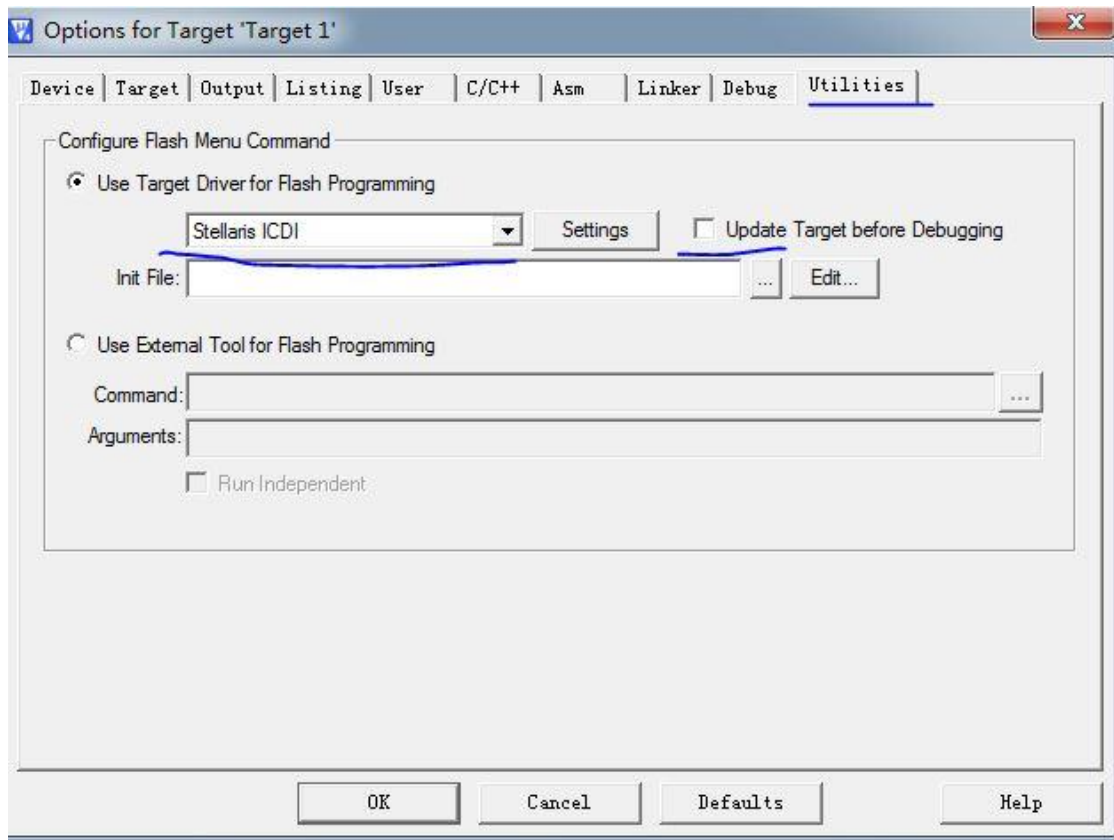


## 2.7 Debug:

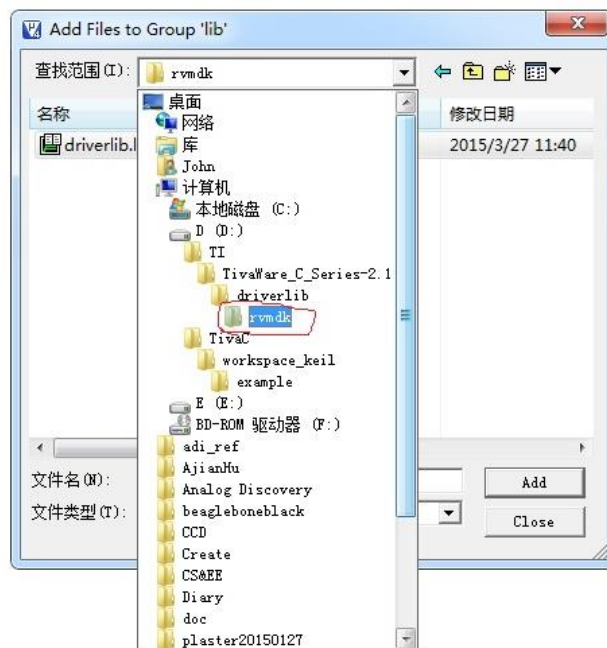


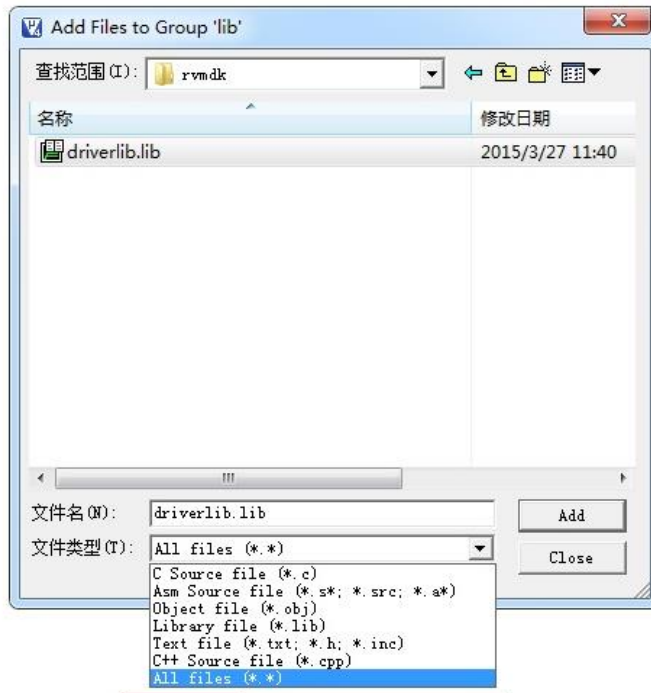


## 2.8 Utilities:



## 2.9 向工程中添加 driver 库文件:

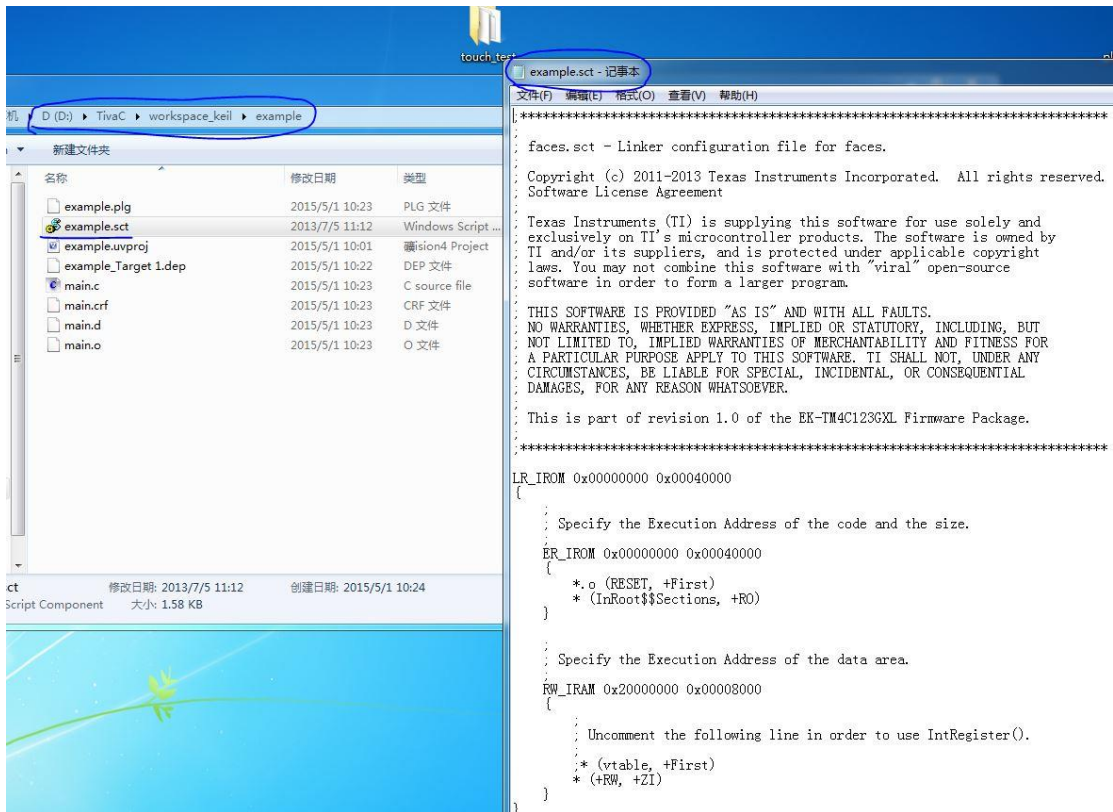




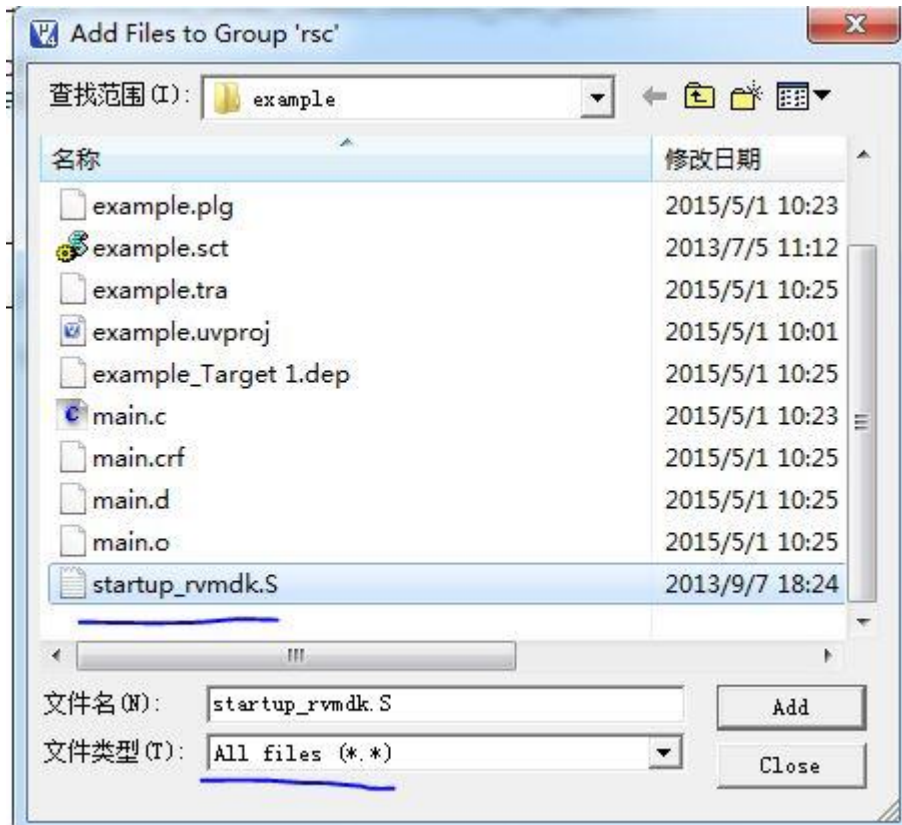
## 2.10 示例程序:

```
main.c
1 #include <stdint.h>
2 #include <stdbool.h>
3 #include "inc/hw_memmap.h"
4 #include "inc/hw_types.h"
5 #include "driverlib/sysctl.h"
6 #include "driverlib/gpio.h"
7
8 int main(void)
9 {
10     uint8_t ui8PinData=2;
11
12     SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
13
14     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
15     GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
16
17     while(1)
18     {
19         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, ui8PinData);
20         SysCtlDelay(2000000);
21         GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
22         SysCtlDelay(2000000);
23         if(ui8PinData==8) {ui8PinData=2;} else {ui8PinData=ui8PinData*2;}
24     }
25     return 0;
26 }
27
```

## 2.11 在工程路径下创建：example.sct 文件，与 2.6 中 scatter file 的路径一致

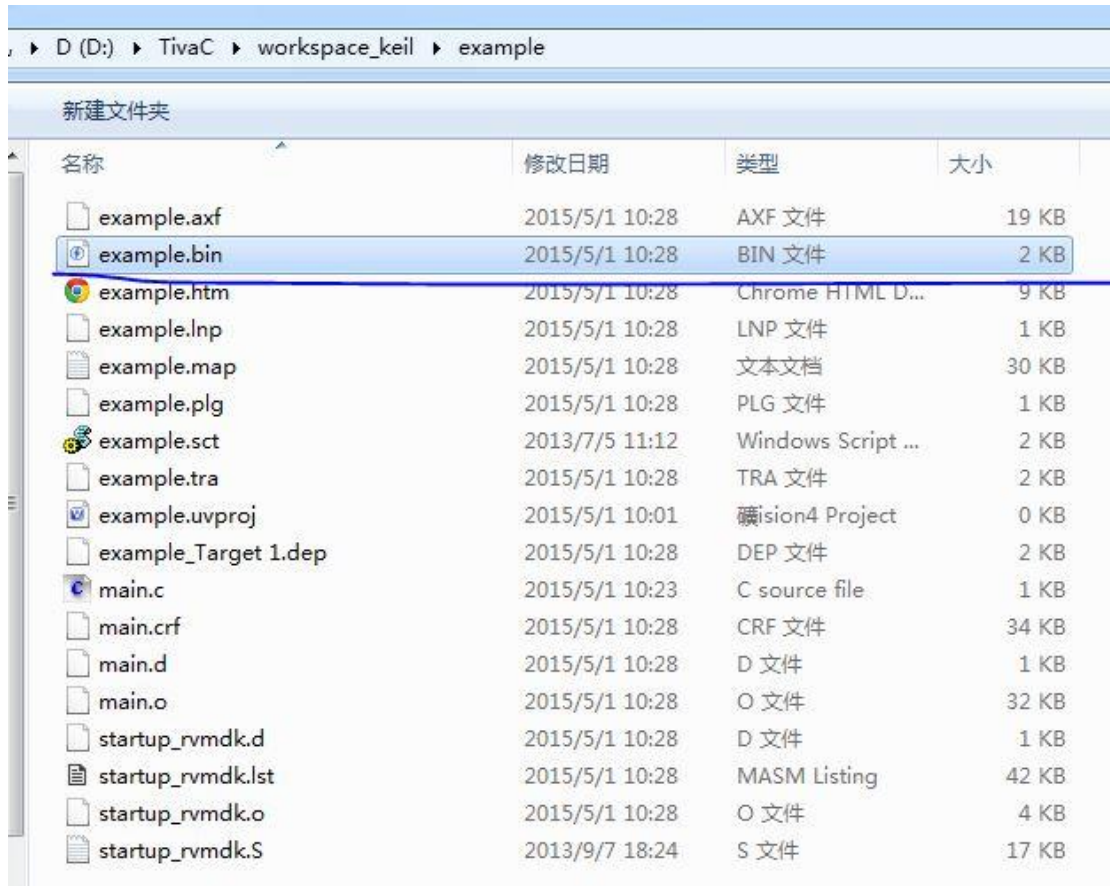


## 2.12 添加 startup.s 文件





### 3 编译产生可执行文件



## 4 附录

### 4.1 sct 文件

```
*****  
*****  
;  
; faces.sct - Linker configuration file for faces.  
;  
; Copyright (c) 2011-2013 Texas Instruments Incorporated. All rights reserved.  
; Software License Agreement  
;  
; Texas Instruments (TI) is supplying this software for use solely and  
; exclusively on TI's microcontroller products. The software is owned by  
; TI and/or its suppliers, and is protected under applicable copyright  
; laws. You may not combine this software with "viral" open-source  
; software in order to form a larger program.  
;  
; THIS SOFTWARE IS PROVIDED "AS IS" AND WITH ALL FAULTS.  
; NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT  
; NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
```

```

; A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. TI SHALL NOT, UNDER ANY
; CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
; DAMAGES, FOR ANY REASON WHATSOEVER.
;
; This is part of revision 1.0 of the EK-TM4C123GXL Firmware Package.
;
; *****
*****

```

```

LR_IROM 0x00000000 0x00040000
{
;
; Specify the Execution Address of the code and the size.
;
ER_IROM 0x00000000 0x00040000
{
.o (RESET, +First)
* (InRoot$$Sections, +RO)
}

;
; Specify the Execution Address of the data area.
;
RW_IRAM 0x20000000 0x00080000
{
;
; Uncomment the following line in order to use IntRegister().
;
;* (vtable, +First)
* (+RW, +ZI)
}
}

```

#### 4.2 startup.s

```

; <<< Use Configuration Wizard in Context Menu >>>
; *****
*****
;
; startup_rvmdk.S - Startup code for use with Keil's uVision.
;
; Copyright (c) 2011-2013 Texas Instruments Incorporated. All rights reserved.
; Software License Agreement
;

```

```

; Texas Instruments (TI) is supplying this software for use solely and
; exclusively on TI's microcontroller products. The software is owned by
; TI and/or its suppliers, and is protected under applicable copyright
; laws. You may not combine this software with "viral" open-source
; software in order to form a larger program.
;
; THIS SOFTWARE IS PROVIDED "AS IS" AND WITH ALL FAULTS.
; NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT
; NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
; A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. TI SHALL NOT, UNDER ANY
; CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
; DAMAGES, FOR ANY REASON WHATSOEVER.
;
; This is part of revision 1.0 of the EK-TM4C123GXL Firmware Package.
;
;*****
*****

;*****
*****

;
; <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
;
;*****
*****

Stack EQU 0x00000400

;*****
*****

;
; <o> Heap Size (in Bytes) <0x0-0xFFFFFFFF:8>
;
;*****
*****

Heap EQU 0x00000000

;*****
*****

;
; Allocate space for the stack.
;
;*****
*****

AREA STACK, NOINIT, READWRITE, ALIGN=3

```

```

StackMem
        SPACE    Stack
__initial_sp

;*****
;
;
; Allocate space for the heap.
;
;*****
*****

        AREA    HEAP, NOINIT, READWRITE, ALIGN=3
__heap_base
HeapMem
        SPACE    Heap
__heap_limit

;*****
;*****
;
; Indicate that the code in this file preserves 8-byte alignment of the stack.
;
;*****
*****

        PRESERVE8

;*****
;*****
;
; Place code into the reset code section.
;
;*****
*****

        AREA    RESET, CODE, READONLY
        THUMB

;*****
;*****
;
; External declaration for the interrupt handler used by the application.
;
;*****
*****

        EXTERN  SysTickIntHandler

```

```

,*****
;
*****
;
; The vector table.
;
,*****
*****

```

```

EXPORT __Vectors
__Vectors
DCD StackMem + Stack ; Top of Stack
DCD Reset_Handler ; Reset Handler
DCD NmiSR ; NMI Handler
DCD FaultISR ; Hard Fault Handler
DCD IntDefaultHandler ; The MPU fault handler
DCD IntDefaultHandler ; The bus fault handler
DCD IntDefaultHandler ; The usage fault handler
DCD 0 ; Reserved
DCD 0 ; Reserved
DCD 0 ; Reserved
DCD 0 ; Reserved
DCD IntDefaultHandler ; SVCall handler
DCD IntDefaultHandler ; Debug monitor handler
DCD 0 ; Reserved
DCD IntDefaultHandler ; The PendSV handler
DCD IntDefaultHandler ; The SysTick handler
DCD IntDefaultHandler ; GPIO Port A
DCD IntDefaultHandler ; GPIO Port B
DCD IntDefaultHandler ; GPIO Port C
DCD IntDefaultHandler ; GPIO Port D
DCD IntDefaultHandler ; GPIO Port E
DCD IntDefaultHandler ; UART0 Rx and Tx
DCD IntDefaultHandler ; UART1 Rx and Tx
DCD IntDefaultHandler ; SSI0 Rx and Tx
DCD IntDefaultHandler ; I2C0 Master and Slave
DCD IntDefaultHandler ; PWM Fault
DCD IntDefaultHandler ; PWM Generator 0
DCD IntDefaultHandler ; PWM Generator 1
DCD IntDefaultHandler ; PWM Generator 2
DCD IntDefaultHandler ; Quadrature Encoder 0
DCD IntDefaultHandler ; ADC Sequence 0
DCD IntDefaultHandler ; ADC Sequence 1
DCD IntDefaultHandler ; ADC Sequence 2
DCD IntDefaultHandler ; ADC Sequence 3

```



DCD	IntDefaultHandler	; Watchdog timer
DCD	IntDefaultHandler	; Timer 0 subtimer A
DCD	IntDefaultHandler	; Timer 0 subtimer B
DCD	IntDefaultHandler	; Timer 1 subtimer A
DCD	IntDefaultHandler	; Timer 1 subtimer B
DCD	IntDefaultHandler	; Timer 2 subtimer A
DCD	IntDefaultHandler	; Timer 2 subtimer B
DCD	IntDefaultHandler	; Analog Comparator 0
DCD	IntDefaultHandler	; Analog Comparator 1
DCD	IntDefaultHandler	; Analog Comparator 2
DCD	IntDefaultHandler	; System Control (PLL, OSC, BO)
DCD	IntDefaultHandler	; FLASH Control
DCD	IntDefaultHandler	; GPIO Port F
DCD	IntDefaultHandler	; GPIO Port G
DCD	IntDefaultHandler	; GPIO Port H
DCD	IntDefaultHandler	; UART2 Rx and Tx
DCD	IntDefaultHandler	; SSI1 Rx and Tx
DCD	IntDefaultHandler	; Timer 3 subtimer A
DCD	IntDefaultHandler	; Timer 3 subtimer B
DCD	IntDefaultHandler	; I2C1 Master and Slave
DCD	IntDefaultHandler	; Quadrature Encoder 1
DCD	IntDefaultHandler	; CAN0
DCD	IntDefaultHandler	; CAN1
DCD	IntDefaultHandler	; CAN2
DCD	0	; Reserved
DCD	IntDefaultHandler	; Hibernate
DCD	IntDefaultHandler	; USB0
DCD	IntDefaultHandler	; PWM Generator 3
DCD	IntDefaultHandler	; uDMA Software Transfer
DCD	IntDefaultHandler	; uDMA Error
DCD	IntDefaultHandler	; ADC1 Sequence 0
DCD	IntDefaultHandler	; ADC1 Sequence 1
DCD	IntDefaultHandler	; ADC1 Sequence 2
DCD	IntDefaultHandler	; ADC1 Sequence 3
DCD	0	; Reserved
DCD	0	; Reserved
DCD	IntDefaultHandler	; GPIO Port J
DCD	IntDefaultHandler	; GPIO Port K
DCD	IntDefaultHandler	; GPIO Port L
DCD	IntDefaultHandler	; SSI2 Rx and Tx
DCD	IntDefaultHandler	; SSI3 Rx and Tx
DCD	IntDefaultHandler	; UART3 Rx and Tx
DCD	IntDefaultHandler	; UART4 Rx and Tx
DCD	IntDefaultHandler	; UART5 Rx and Tx



```

DCD    IntDefaultHandler    ; FPU
DCD    0                    ; Reserved
DCD    0                    ; Reserved
DCD    IntDefaultHandler    ; I2C4 Master and Slave
DCD    IntDefaultHandler    ; I2C5 Master and Slave
DCD    IntDefaultHandler    ; GPIO Port M
DCD    IntDefaultHandler    ; GPIO Port N
DCD    IntDefaultHandler    ; Quadrature Encoder 2
DCD    0                    ; Reserved
DCD    0                    ; Reserved
DCD    IntDefaultHandler    ; GPIO Port P (Summary or P0)
DCD    IntDefaultHandler    ; GPIO Port P1
DCD    IntDefaultHandler    ; GPIO Port P2
DCD    IntDefaultHandler    ; GPIO Port P3
DCD    IntDefaultHandler    ; GPIO Port P4
DCD    IntDefaultHandler    ; GPIO Port P5
DCD    IntDefaultHandler    ; GPIO Port P6
DCD    IntDefaultHandler    ; GPIO Port P7
DCD    IntDefaultHandler    ; GPIO Port Q (Summary or Q0)
DCD    IntDefaultHandler    ; GPIO Port Q1
DCD    IntDefaultHandler    ; GPIO Port Q2
DCD    IntDefaultHandler    ; GPIO Port Q3
DCD    IntDefaultHandler    ; GPIO Port Q4
DCD    IntDefaultHandler    ; GPIO Port Q5
DCD    IntDefaultHandler    ; GPIO Port Q6
DCD    IntDefaultHandler    ; GPIO Port Q7
DCD    IntDefaultHandler    ; GPIO Port R
DCD    IntDefaultHandler    ; GPIO Port S
DCD    IntDefaultHandler    ; PWM 1 Generator 0
DCD    IntDefaultHandler    ; PWM 1 Generator 1
DCD    IntDefaultHandler    ; PWM 1 Generator 2
DCD    IntDefaultHandler    ; PWM 1 Generator 3
DCD    IntDefaultHandler    ; PWM 1 Fault

```

```

,*****
*****
;
; This is the code that gets called when the processor first starts execution
; following a reset event.
;
,*****
*****

```

```

EXPORT Reset_Handler
Reset_Handler

```

```

;
; Enable the floating-point unit. This must be done here to handle the
; case where main() uses floating-point and the function prologue saves
; floating-point registers (which will fault if floating-point is not
; enabled). Any configuration of the floating-point unit using
; DriverLib APIs must be done here prior to the floating-point unit
; being enabled.
;
; Note that this does not use DriverLib since it might not be included
; in this project.
;
MOVW    R0, #0xED88
MOVT    R0, #0xE000
LDR     R1, [R0]
ORR     R1, #0x00F00000
STR     R1, [R0]

;
; Call the C library entry point that handles startup. This will copy
; the .data section initializers from flash to SRAM and zero fill the
; .bss section.
;
IMPORT  __main
B       __main

;*****
;
; This is the code that gets called when the processor receives a NMI. This
; simply enters an infinite loop, preserving the system state for examination
; by a debugger.
;
;*****
*****
NmiSR
        B       NmiSR

;*****
;
; This is the code that gets called when the processor receives a fault
; interrupt. This simply enters an infinite loop, preserving the system state
; for examination by a debugger.
;

```

```

    ,*****
    ;
*****
    FaultISR
        B        FaultISR

    ,*****
    ;
*****
    ;
    ; This is the code that gets called when the processor receives an unexpected
    ; interrupt. This simply enters an infinite loop, preserving the system state
    ; for examination by a debugger.
    ;
    ,*****
    ;
*****
    IntDefaultHandler
        B        IntDefaultHandler

    ,*****
    ;
*****
    ;
    ; Make sure the end of this section is aligned.
    ;
    ,*****
    ;
*****
    ALIGN

    ,*****
    ;
*****
    ;
    ; Some code in the normal code section for initializing the heap and stack.
    ;
    ,*****
    ;
*****
    AREA    |.text|, CODE, READONLY

    ,*****
    ;
*****
    ;
    ; The function expected of the C library startup code for defining the stack
    ; and heap memory locations. For the C library version of the startup code,
    ; provide this function so that the C library initialization code can find out
    ; the location of the stack and heap.
    ;
    ,*****
    ;

```



```

*****
    IF :DEF: __MICROLIB
        EXPORT __initial_sp
        EXPORT __heap_base
        EXPORT __heap_limit
    ELSE
        IMPORT __use_two_region_memory
        EXPORT __user_initial_stackheap
__user_initial_stackheap
        LDR    R0, =HeapMem
        LDR    R1, =(StackMem + Stack)
        LDR    R2, =(HeapMem + Heap)
        LDR    R3, =StackMem
        BX    LR
    ENDIF

;*****
*****
;
; Make sure the end of this section is aligned.
;
;*****
*****

        ALIGN

;*****
*****
;
; Tell the assembler that we're done.
;
;*****
*****

        END

```