```
; <<< Use Configuration Wizard in Context Menu >>>
;*******************************************************************************
;
; startup_rvmdk.S - Startup code for use with Keil's uVision.
;
; Copyright (c) 2012-2014 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 2.1.0.12573 of the EK-TM4C123GXL Firmware Package.
;
;*******************************************************************************

;*******************************************************************************
;
; <o> Stack Size (in Bytes) <0x0-0xFFFFFFFF:8>
;
;*******************************************************************************
Stack   EQU     0x00000800

;*******************************************************************************
;
; <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 declarations for the interrupt handlers used by the application.
;
;*******************************************************************************
        EXTERN  SysTickIntHandler
        EXTERN  ShortTimerAHandler
        EXTERN  ShortTimerBHandler
        EXTERN  InputCaptureResponse_Encoder1
        EXTERN  OneShotIntResponse_Encoder1
        EXTERN  InputCaptureResponse_Encoder2
        EXTERN  OneShotIntResponse_Encoder2
        EXTERN  PeriodicIntResponse
        EXTERN  SPI_Interupt_Response
        EXTERN  InputCaptureResponse_Hall
        EXTERN  InputCaptureResponse_Ultrasonics
        EXTERN  OneShotIntResponse_Supply
;       EXTERN  UARTStdioIntHandler

;*******************************************************************************
;
; 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     SysTickIntHandler           ; 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     SPI_Interupt_Response       ; 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     0                           ; Reserved
        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           ; UART6 Rx and Tx
        DCD     IntDefaultHandler           ; UART7 Rx and Tx
        DCD     0                           ; Reserved
```

```
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     IntDefaultHandler           ; I2C2 Master and Slave
        DCD     IntDefaultHandler           ; I2C3 Master and Slave
        DCD     IntDefaultHandler           ; Timer 4 subtimer A
        DCD     IntDefaultHandler           ; Timer 4 subtimer B
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     ShortTimerAHandler          ; Timer 5 subtimer A
        DCD     ShortTimerBHandler          ; Timer 5 subtimer B
        DCD     InputCaptureResponse_Encoder1   ; Wide Timer 0 subtimer A
        DCD     OneShotIntResponse_Encoder1     ; Wide Timer 0 subtimer B
        DCD     InputCaptureResponse_Encoder2   ; Wide Timer 1 subtimer A
        DCD     OneShotIntResponse_Encoder2     ; Wide Timer 1 subtimer B
        DCD     IntDefaultHandler           ; Wide Timer 2 subtimer A
        DCD     IntDefaultHandler           ; Wide Timer 2 subtimer B
        DCD     PeriodicIntResponse         ; Wide Timer 3 subtimer A
        DCD     IntDefaultHandler           ; Wide Timer 3 subtimer B
        DCD     IntDefaultHandler           ; Wide Timer 4 subtimer A
        DCD     OneShotIntResponse_Supply          ; Wide Timer 4 subtimer B
        DCD     InputCaptureResponse_Hall          ; Wide Timer 5 subtimer A
        DCD     InputCaptureResponse_Ultrasonics   ; Wide Timer 5 subtimer B
        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 enty 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
```