```
; <<< Use Configuration Wizard in Context Menu >>>
;*******************************************************************************
;
; startup_rvmdk.S - Startup code for use with Keil's uVision.
;
; Copyright (c) 2012 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 9453 of the EK-LM4F120XL Firmware Package.
;
;*******************************************************************************
; Edited to conform with ISR names as described in
;   "Embedded Systems: Introduction to ARM Cortex M Microcontrollers",
;   ISBN: 978-1469998749, Jonathan Valvano, copyright (c) 2012
;   "Embedded Systems: Real Time Interfacing to ARM Cortex M Microcontrollers",
;   ISBN: 978-1463590154, Jonathan Valvano, copyright (c) 2012
;   "Embedded Systems: Real-Time Operating Systems for ARM Cortex M Microcontrollers",
;   ISBN: 978-1466468863, Jonathan Valvano, copyright (c) 2013
;
;*******************************************************************************
;
; <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

;*******************************************************************************
;
; The vector table.
;
;*******************************************************************************
        EXPORT  __Vectors
__Vectors
        DCD     StackMem + Stack            ; Top of Stack
        DCD     Reset_Handler               ; Reset Handler
        DCD     NMI_Handler                 ; NMI Handler
        DCD     HardFault_Handler           ; Hard Fault Handler
        DCD     MemManage_Handler           ; MPU Fault Handler
        DCD     BusFault_Handler            ; Bus Fault Handler
        DCD     UsageFault_Handler          ; Usage Fault Handler
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     0                           ; Reserved
        DCD     SVC_Handler                 ; SVCall Handler
        DCD     DebugMon_Handler            ; Debug Monitor Handler
        DCD     0                           ; Reserved
        DCD     PendSV_Handler              ; PendSV Handler
        DCD     SysTick_Handler             ; SysTick Handler
        DCD     GPIOPortA_Handler           ; GPIO Port A
        DCD     GPIOPortB_Handler           ; GPIO Port B
        DCD     GPIOPortC_Handler           ; GPIO Port C
        DCD     GPIOPortD_Handler           ; GPIO Port D
        DCD     GPIOPortE_Handler           ; GPIO Port E
        DCD     UART0_Handler               ; UART0 Rx and Tx
        DCD     UART1_Handler               ; UART1 Rx and Tx
        DCD     SSI0_Handler                ; SSI0 Rx and Tx
        DCD     I2C0_Handler                ; I2C0 Master and Slave
        DCD     PWM0Fault_Handler           ; PWM 0 Fault
        DCD     PWM0Generator0_Handler      ; PWM 0 Generator 0
        DCD     PWM0Generator1_Handler      ; PWM 0 Generator 1
        DCD     PWM0Generator2_Handler      ; PWM 0 Generator 2
        DCD     Quadrature0_Handler         ; Quadrature Encoder 0
        DCD     ADC0Seq0_Handler            ; ADC0 Sequence 0
        DCD     ADC0Seq1_Handler            ; ADC0 Sequence 1
        DCD     ADC0Seq2_Handler            ; ADC0 Sequence 2
        DCD     ADC0Seq3_Handler            ; ADC0 Sequence 3
        DCD     WDT_Handler                 ; Watchdog
        DCD     Timer0A_Handler             ; Timer 0 subtimer A
        DCD     Timer0B_Handler             ; Timer 0 subtimer B
        DCD     Timer1A_Handler             ; Timer 1 subtimer A
        DCD     Timer1B_Handler             ; Timer 1 subtimer B
        DCD     Timer2A_Handler             ; Timer 2 subtimer A
        DCD     Timer2B_Handler             ; Timer 2 subtimer B
        DCD     Comp0_Handler               ; Analog Comp 0
        DCD     Comp1_Handler               ; Analog Comp 1
        DCD     Comp2_Handler               ; Analog Comp 2
        DCD     SysCtl_Handler              ; System Control
        DCD     FlashCtl_Handler            ; Flash Control
        DCD     GPIOPortF_Handler           ; GPIO Port F
        DCD     GPIOPortG_Handler           ; GPIO Port G
        DCD     GPIOPortH_Handler           ; GPIO Port H
        DCD     UART2_Handler               ; UART2 Rx and Tx
        DCD     SSI1_Handler                ; SSI1 Rx and Tx
        DCD     Timer3A_Handler             ; Timer 3 subtimer A
        DCD     Timer3B_Handler             ; Timer 3 subtimer B
        DCD     I2C1_Handler                ; I2C1 Master and Slave
        DCD     Quadrature1_Handler         ; Quadrature Encoder 1
        DCD     CAN0_Handler                ; CAN0
        DCD     CAN1_Handler                ; CAN1
```

```
147             DCD     CAN2_Handler                ; CAN2
148             DCD     Ethernet_Handler            ; Ethernet
149             DCD     Hibernate_Handler           ; Hibernate
150             DCD     USB0_Handler                ; USB0
151             DCD     PWM0Generator3_Handler      ; PWM 0 Generator 3
152             DCD     uDMA_Handler                ; uDMA Software Transfer
153             DCD     uDMA_Error                  ; uDMA Error
154             DCD     ADC1Seq0_Handler            ; ADC1 Sequence 0
155             DCD     ADC1Seq1_Handler            ; ADC1 Sequence 1
156             DCD     ADC1Seq2_Handler            ; ADC1 Sequence 2
157             DCD     ADC1Seq3_Handler            ; ADC1 Sequence 3
158             DCD     I2S0_Handler                ; I2S0
159             DCD     ExtBus_Handler              ; External Bus Interface 0
160             DCD     GPIOPortJ_Handler           ; GPIO Port J
161             DCD     GPIOPortK_Handler           ; GPIO Port K
162             DCD     GPIOPortL_Handler           ; GPIO Port L
163             DCD     SSI2_Handler                ; SSI2 Rx and Tx
164             DCD     SSI3_Handler                ; SSI3 Rx and Tx
165             DCD     UART3_Handler               ; UART3 Rx and Tx
166             DCD     UART4_Handler               ; UART4 Rx and Tx
167             DCD     UART5_Handler               ; UART5 Rx and Tx
168             DCD     UART6_Handler               ; UART6 Rx and Tx
169             DCD     UART7_Handler               ; UART7 Rx and Tx
170             DCD     0                           ; Reserved
171             DCD     0                           ; Reserved
172             DCD     0                           ; Reserved
173             DCD     0                           ; Reserved
174             DCD     I2C2_Handler                ; I2C2 Master and Slave
175             DCD     I2C3_Handler                ; I2C3 Master and Slave
176             DCD     Timer4A_Handler             ; Timer 4 subtimer A
177             DCD     Timer4B_Handler             ; Timer 4 subtimer B
178             DCD     0                           ; Reserved
179             DCD     0                           ; Reserved
180             DCD     0                           ; Reserved
181             DCD     0                           ; Reserved
182             DCD     0                           ; Reserved
183             DCD     0                           ; Reserved
184             DCD     0                           ; Reserved
185             DCD     0                           ; Reserved
186             DCD     0                           ; Reserved
187             DCD     0                           ; Reserved
188             DCD     0                           ; Reserved
189             DCD     0                           ; Reserved
190             DCD     0                           ; Reserved
191             DCD     0                           ; Reserved
192             DCD     0                           ; Reserved
193             DCD     0                           ; Reserved
194             DCD     0                           ; Reserved
195             DCD     0                           ; Reserved
196             DCD     0                           ; Reserved
197             DCD     0                           ; Reserved
198             DCD     Timer5A_Handler             ; Timer 5 subtimer A
199             DCD     Timer5B_Handler             ; Timer 5 subtimer B
200             DCD     WideTimer0A_Handler         ; Wide Timer 0 subtimer A
201             DCD     WideTimer0B_Handler         ; Wide Timer 0 subtimer B
202             DCD     WideTimer1A_Handler         ; Wide Timer 1 subtimer A
203             DCD     WideTimer1B_Handler         ; Wide Timer 1 subtimer B
204             DCD     WideTimer2A_Handler         ; Wide Timer 2 subtimer A
205             DCD     WideTimer2B_Handler         ; Wide Timer 2 subtimer B
206             DCD     WideTimer3A_Handler         ; Wide Timer 3 subtimer A
207             DCD     WideTimer3B_Handler         ; Wide Timer 3 subtimer B
208             DCD     WideTimer4A_Handler         ; Wide Timer 4 subtimer A
209             DCD     WideTimer4B_Handler         ; Wide Timer 4 subtimer B
210             DCD     WideTimer5A_Handler         ; Wide Timer 5 subtimer A
211             DCD     WideTimer5B_Handler         ; Wide Timer 5 subtimer B
212             DCD     FPU_Handler                 ; FPU
213             DCD     PECI0_Handler               ; PECI 0
214             DCD     LPC0_Handler                ; LPC 0
215             DCD     I2C4_Handler                ; I2C4 Master and Slave
216             DCD     I2C5_Handler                ; I2C5 Master and Slave
217             DCD     GPIOPortM_Handler           ; GPIO Port M
218             DCD     GPIOPortN_Handler           ; GPIO Port N
219             DCD     Quadrature2_Handler         ; Quadrature Encoder 2
```

```
220             DCD     Fan0_Handler                ; Fan 0
221             DCD     0                           ; Reserved
222             DCD     GPIOPortP_Handler           ; GPIO Port P (Summary or P0)
223             DCD     GPIOPortP1_Handler          ; GPIO Port P1
224             DCD     GPIOPortP2_Handler          ; GPIO Port P2
225             DCD     GPIOPortP3_Handler          ; GPIO Port P3
226             DCD     GPIOPortP4_Handler          ; GPIO Port P4
227             DCD     GPIOPortP5_Handler          ; GPIO Port P5
228             DCD     GPIOPortP6_Handler          ; GPIO Port P6
229             DCD     GPIOPortP7_Handler          ; GPIO Port P7
230             DCD     GPIOPortQ_Handler           ; GPIO Port Q (Summary or Q0)
231             DCD     GPIOPortQ1_Handler          ; GPIO Port Q1
232             DCD     GPIOPortQ2_Handler          ; GPIO Port Q2
233             DCD     GPIOPortQ3_Handler          ; GPIO Port Q3
234             DCD     GPIOPortQ4_Handler          ; GPIO Port Q4
235             DCD     GPIOPortQ5_Handler          ; GPIO Port Q5
236             DCD     GPIOPortQ6_Handler          ; GPIO Port Q6
237             DCD     GPIOPortQ7_Handler          ; GPIO Port Q7
238             DCD     GPIOPortR_Handler           ; GPIO Port R
239             DCD     GPIOPortS_Handler           ; GPIO Port S
240             DCD     PWM1Generator0_Handler      ; PWM 1 Generator 0
241             DCD     PWM1Generator1_Handler      ; PWM 1 Generator 1
242             DCD     PWM1Generator2_Handler      ; PWM 1 Generator 2
243             DCD     PWM1Generator3_Handler      ; PWM 1 Generator 3
244             DCD     PWM1Fault_Handler           ; PWM 1 Fault
245
246     ;*******************************************************************************
247     ;
248     ; This is the code that gets called when the processor first starts execution
249     ; following a reset event.
250     ;
251     ;*******************************************************************************
252             EXPORT  Reset_Handler
253     Reset_Handler
254             ;
255             ; DO NOT enable the floating-point unit.  This must be done here to handle the
256             ; case where main() uses floating-point and the function prologue saves
257             ; floating-point registers (which will fault if floating-point is not
258             ; enabled).  Any configuration of the floating-point unit using
259             ; DriverLib APIs must be done here prior to the floating-point unit
260             ; being enabled.
261             ;
262             ; Note that this does not use DriverLib since it might not be included
263             ; in this project.
264             ;
265     ;        MOVW    R0, #0xED88
266     ;        MOVT    R0, #0xE000
267     ;        LDR     R1, [R0]
268     ;        ORR     R1, #0x00F00000
269     ;        STR     R1, [R0]
270
271             ;
272             ; Call the C library enty point that handles startup.  This will copy
273             ; the .data section initializers from flash to SRAM and zero fill the
274             ; .bss section.
275             ;
276             IMPORT  __main
277             B       __main
278
279     ;*******************************************************************************
280     ;
281     ; This is the code that gets called when the processor receives a NMI.  This
282     ; simply enters an infinite loop, preserving the system state for examination
283     ; by a debugger.
284     ;
285     ;*******************************************************************************
286     NMI_Handler     PROC
287                     EXPORT  NMI_Handler             [WEAK]
288                     B       .
289                     ENDP
290
291     ;*******************************************************************************
292     ;
```

```
293   ; This is the code that gets called when the processor receives a fault
294   ; interrupt.  This simply enters an infinite loop, preserving the system state
295   ; for examination by a debugger.
296   ;
297   ;*******************************************************************************
298   HardFault_Handler\
299                   PROC
300                   EXPORT  HardFault_Handler         [WEAK]
301                   B       .
302                   ENDP
303
304   MemManage_Handler\
305                   PROC
306                   EXPORT  MemManage_Handler         [WEAK]
307                   B       .
308                   ENDP
309   BusFault_Handler\
310                   PROC
311                   EXPORT  BusFault_Handler          [WEAK]
312                   B       .
313                   ENDP
314   UsageFault_Handler\
315                   PROC
316                   EXPORT  UsageFault_Handler        [WEAK]
317                   B       .
318                   ENDP
319   SVC_Handler     PROC
320                   EXPORT  SVC_Handler               [WEAK]
321                   B       .
322                   ENDP
323   DebugMon_Handler\
324                   PROC
325                   EXPORT  DebugMon_Handler          [WEAK]
326                   B       .
327                   ENDP
328   PendSV_Handler  PROC
329                   EXPORT  PendSV_Handler            [WEAK]
330                   B       .
331                   ENDP
332   SysTick_Handler PROC
333                   EXPORT  SysTick_Handler           [WEAK]
334                   B       .
335                   ENDP
336   IntDefaultHandler\
337                   PROC
338
339                   EXPORT  GPIOPortA_Handler         [WEAK]
340                   EXPORT  GPIOPortB_Handler         [WEAK]
341                   EXPORT  GPIOPortC_Handler         [WEAK]
342                   EXPORT  GPIOPortD_Handler         [WEAK]
343                   EXPORT  GPIOPortE_Handler         [WEAK]
344                   EXPORT  UART0_Handler             [WEAK]
345                   EXPORT  UART1_Handler             [WEAK]
346                   EXPORT  SSI0_Handler              [WEAK]
347                   EXPORT  I2C0_Handler              [WEAK]
348                   EXPORT  PWM0Fault_Handler         [WEAK]
349                   EXPORT  PWM0Generator0_Handler    [WEAK]
350                   EXPORT  PWM0Generator1_Handler    [WEAK]
351                   EXPORT  PWM0Generator2_Handler    [WEAK]
352                   EXPORT  Quadrature0_Handler       [WEAK]
353                   EXPORT  ADC0Seq0_Handler          [WEAK]
354                   EXPORT  ADC0Seq1_Handler          [WEAK]
355                   EXPORT  ADC0Seq2_Handler          [WEAK]
356                   EXPORT  ADC0Seq3_Handler          [WEAK]
357                   EXPORT  WDT_Handler               [WEAK]
358                   EXPORT  Timer0A_Handler           [WEAK]
359                   EXPORT  Timer0B_Handler           [WEAK]
360                   EXPORT  Timer1A_Handler           [WEAK]
361                   EXPORT  Timer1B_Handler           [WEAK]
362                   EXPORT  Timer2A_Handler           [WEAK]
363                   EXPORT  Timer2B_Handler           [WEAK]
364                   EXPORT  Comp0_Handler             [WEAK]
365                   EXPORT  Comp1_Handler             [WEAK]
```

```
366                     EXPORT  Comp2_Handler             [WEAK]
367                     EXPORT  SysCtl_Handler            [WEAK]
368                     EXPORT  FlashCtl_Handler          [WEAK]
369                     EXPORT  GPIOPortF_Handler         [WEAK]
370                     EXPORT  GPIOPortG_Handler         [WEAK]
371                     EXPORT  GPIOPortH_Handler         [WEAK]
372                     EXPORT  UART2_Handler             [WEAK]
373                     EXPORT  SSI1_Handler              [WEAK]
374                     EXPORT  Timer3A_Handler           [WEAK]
375                     EXPORT  Timer3B_Handler           [WEAK]
376                     EXPORT  I2C1_Handler              [WEAK]
377                     EXPORT  Quadrature1_Handler       [WEAK]
378                     EXPORT  CAN0_Handler              [WEAK]
379                     EXPORT  CAN1_Handler              [WEAK]
380                     EXPORT  CAN2_Handler              [WEAK]
381                     EXPORT  Ethernet_Handler          [WEAK]
382                     EXPORT  Hibernate_Handler         [WEAK]
383                     EXPORT  USB0_Handler              [WEAK]
384                     EXPORT  PWM0Generator3_Handler    [WEAK]
385                     EXPORT  uDMA_Handler              [WEAK]
386                     EXPORT  uDMA_Error                [WEAK]
387                     EXPORT  ADC1Seq0_Handler          [WEAK]
388                     EXPORT  ADC1Seq1_Handler          [WEAK]
389                     EXPORT  ADC1Seq2_Handler          [WEAK]
390                     EXPORT  ADC1Seq3_Handler          [WEAK]
391                     EXPORT  I2S0_Handler              [WEAK]
392                     EXPORT  ExtBus_Handler            [WEAK]
393                     EXPORT  GPIOPortJ_Handler         [WEAK]
394                     EXPORT  GPIOPortK_Handler         [WEAK]
395                     EXPORT  GPIOPortL_Handler         [WEAK]
396                     EXPORT  SSI2_Handler              [WEAK]
397                     EXPORT  SSI3_Handler              [WEAK]
398                     EXPORT  UART3_Handler             [WEAK]
399                     EXPORT  UART4_Handler             [WEAK]
400                     EXPORT  UART5_Handler             [WEAK]
401                     EXPORT  UART6_Handler             [WEAK]
402                     EXPORT  UART7_Handler             [WEAK]
403                     EXPORT  I2C2_Handler              [WEAK]
404                     EXPORT  I2C3_Handler              [WEAK]
405                     EXPORT  Timer4A_Handler           [WEAK]
406                     EXPORT  Timer4B_Handler           [WEAK]
407                     EXPORT  Timer5A_Handler           [WEAK]
408                     EXPORT  Timer5B_Handler           [WEAK]
409                     EXPORT  WideTimer0A_Handler       [WEAK]
410                     EXPORT  WideTimer0B_Handler       [WEAK]
411                     EXPORT  WideTimer1A_Handler       [WEAK]
412                     EXPORT  WideTimer1B_Handler       [WEAK]
413                     EXPORT  WideTimer2A_Handler       [WEAK]
414                     EXPORT  WideTimer2B_Handler       [WEAK]
415                     EXPORT  WideTimer3A_Handler       [WEAK]
416                     EXPORT  WideTimer3B_Handler       [WEAK]
417                     EXPORT  WideTimer4A_Handler       [WEAK]
418                     EXPORT  WideTimer4B_Handler       [WEAK]
419                     EXPORT  WideTimer5A_Handler       [WEAK]
420                     EXPORT  WideTimer5B_Handler       [WEAK]
421                     EXPORT  FPU_Handler               [WEAK]
422                     EXPORT  PECI0_Handler             [WEAK]
423                     EXPORT  LPC0_Handler              [WEAK]
424                     EXPORT  I2C4_Handler              [WEAK]
425                     EXPORT  I2C5_Handler              [WEAK]
426                     EXPORT  GPIOPortM_Handler         [WEAK]
427                     EXPORT  GPIOPortN_Handler         [WEAK]
428                     EXPORT  Quadrature2_Handler       [WEAK]
429                     EXPORT  Fan0_Handler              [WEAK]
430                     EXPORT  GPIOPortP_Handler         [WEAK]
431                     EXPORT  GPIOPortP1_Handler        [WEAK]
432                     EXPORT  GPIOPortP2_Handler        [WEAK]
433                     EXPORT  GPIOPortP3_Handler        [WEAK]
434                     EXPORT  GPIOPortP4_Handler        [WEAK]
435                     EXPORT  GPIOPortP5_Handler        [WEAK]
436                     EXPORT  GPIOPortP6_Handler        [WEAK]
437                     EXPORT  GPIOPortP7_Handler        [WEAK]
438                     EXPORT  GPIOPortQ_Handler         [WEAK]
```

```
439                     EXPORT  GPIOPortQ1_Handler          [WEAK]
440                     EXPORT  GPIOPortQ2_Handler          [WEAK]
441                     EXPORT  GPIOPortQ3_Handler          [WEAK]
442                     EXPORT  GPIOPortQ4_Handler          [WEAK]
443                     EXPORT  GPIOPortQ5_Handler          [WEAK]
444                     EXPORT  GPIOPortQ6_Handler          [WEAK]
445                     EXPORT  GPIOPortQ7_Handler          [WEAK]
446                     EXPORT  GPIOPortR_Handler           [WEAK]
447                     EXPORT  GPIOPortS_Handler           [WEAK]
448                     EXPORT  PWM1Generator0_Handler      [WEAK]
449                     EXPORT  PWM1Generator1_Handler      [WEAK]
450                     EXPORT  PWM1Generator2_Handler      [WEAK]
451                     EXPORT  PWM1Generator3_Handler      [WEAK]
452                     EXPORT  PWM1Fault_Handler           [WEAK]
453
454     GPIOPortA_Handler
455     GPIOPortB_Handler
456     GPIOPortC_Handler
457     GPIOPortD_Handler
458     GPIOPortE_Handler
459     UART0_Handler
460     UART1_Handler
461     SSI0_Handler
462     I2C0_Handler
463     PWM0Fault_Handler
464     PWM0Generator0_Handler
465     PWM0Generator1_Handler
466     PWM0Generator2_Handler
467     Quadrature0_Handler
468     ADC0Seq0_Handler
469     ADC0Seq1_Handler
470     ADC0Seq2_Handler
471     ADC0Seq3_Handler
472     WDT_Handler
473     Timer0A_Handler
474     Timer0B_Handler
475     Timer1A_Handler
476     Timer1B_Handler
477     Timer2A_Handler
478     Timer2B_Handler
479     Comp0_Handler
480     Comp1_Handler
481     Comp2_Handler
482     SysCtl_Handler
483     FlashCtl_Handler
484     GPIOPortF_Handler
485     GPIOPortG_Handler
486     GPIOPortH_Handler
487     UART2_Handler
488     SSI1_Handler
489     Timer3A_Handler
490     Timer3B_Handler
491     I2C1_Handler
492     Quadrature1_Handler
493     CAN0_Handler
494     CAN1_Handler
495     CAN2_Handler
496     Ethernet_Handler
497     Hibernate_Handler
498     USB0_Handler
499     PWM0Generator3_Handler
500     uDMA_Handler
501     uDMA_Error
502     ADC1Seq0_Handler
503     ADC1Seq1_Handler
504     ADC1Seq2_Handler
505     ADC1Seq3_Handler
506     I2S0_Handler
507     ExtBus_Handler
508     GPIOPortJ_Handler
509     GPIOPortK_Handler
510     GPIOPortL_Handler
511     SSI2_Handler
```

```
  512   SSI3_Handler
  513   UART3_Handler
  514   UART4_Handler
  515   UART5_Handler
  516   UART6_Handler
  517   UART7_Handler
  518   I2C2_Handler
  519   I2C3_Handler
  520   Timer4A_Handler
  521   Timer4B_Handler
  522   Timer5A_Handler
  523   Timer5B_Handler
  524   WideTimer0A_Handler
  525   WideTimer0B_Handler
  526   WideTimer1A_Handler
  527   WideTimer1B_Handler
  528   WideTimer2A_Handler
  529   WideTimer2B_Handler
  530   WideTimer3A_Handler
  531   WideTimer3B_Handler
  532   WideTimer4A_Handler
  533   WideTimer4B_Handler
  534   WideTimer5A_Handler
  535   WideTimer5B_Handler
  536   FPU_Handler
  537   PECI0_Handler
  538   LPC0_Handler
  539   I2C4_Handler
  540   I2C5_Handler
  541   GPIOPortM_Handler
  542   GPIOPortN_Handler
  543   Quadrature2_Handler
  544   Fan0_Handler
  545   GPIOPortP_Handler
  546   GPIOPortP1_Handler
  547   GPIOPortP2_Handler
  548   GPIOPortP3_Handler
  549   GPIOPortP4_Handler
  550   GPIOPortP5_Handler
  551   GPIOPortP6_Handler
  552   GPIOPortP7_Handler
  553   GPIOPortQ_Handler
  554   GPIOPortQ1_Handler
  555   GPIOPortQ2_Handler
  556   GPIOPortQ3_Handler
  557   GPIOPortQ4_Handler
  558   GPIOPortQ5_Handler
  559   GPIOPortQ6_Handler
  560   GPIOPortQ7_Handler
  561   GPIOPortR_Handler
  562   GPIOPortS_Handler
  563   PWM1Generator0_Handler
  564   PWM1Generator1_Handler
  565   PWM1Generator2_Handler
  566   PWM1Generator3_Handler
  567   PWM1Fault_Handler
  568
  569                   B       .
  570
  571                   ENDP
  572
  573   ;*******************************************************************************
  574   ;
  575   ; Make sure the end of this section is aligned.
  576   ;
  577   ;*******************************************************************************
  578           ALIGN
  579
  580   ;*******************************************************************************
  581   ;
  582   ; Some code in the normal code section for initializing the heap and stack.
  583   ;
  584   ;*******************************************************************************
```

```
585          AREA    |.text|, CODE, READONLY
586
587  ;*******************************************************************************
588  ;
589  ; Useful functions.
590  ;
591  ;*******************************************************************************
592          EXPORT  DisableInterrupts
593          EXPORT  EnableInterrupts
594          EXPORT  StartCritical
595          EXPORT  EndCritical
596          EXPORT  WaitForInterrupt
597
598  ;************ DisableInterrupts ***************
599  ; disable interrupts
600  ; inputs:  none
601  ; outputs: none
602  DisableInterrupts
603          CPSID  I
604          BX     LR
605
606  ;************ EnableInterrupts ***************
607  ; enable interrupts
608  ; inputs:  none
609  ; outputs: none
610  EnableInterrupts
611          CPSIE  I
612          BX     LR
613
614  ;************ StartCritical ************************
615  ; make a copy of previous I bit, disable interrupts
616  ; inputs:  none
617  ; outputs: previous I bit
618  StartCritical
619          MRS    R0, PRIMASK  ; save old status
620          CPSID  I            ; mask all (except faults)
621          BX     LR
622
623  ;************ EndCritical ************************
624  ; using the copy of previous I bit, restore I bit to previous value
625  ; inputs:  previous I bit
626  ; outputs: none
627  EndCritical
628          MSR    PRIMASK, R0
629          BX     LR
630
631  ;************ WaitForInterrupt ************************
632  ; go to low power mode while waiting for the next interrupt
633  ; inputs:  none
634  ; outputs: none
635  WaitForInterrupt
636          WFI
637          BX     LR
638
639  ;*******************************************************************************
640  ;
641  ; The function expected of the C library startup code for defining the stack
642  ; and heap memory locations.  For the C library version of the startup code,
643  ; provide this function so that the C library initialization code can find out
644  ; the location of the stack and heap.
645  ;
646  ;*******************************************************************************
647      IF :DEF: __MICROLIB
648          EXPORT  __initial_sp
649          EXPORT  __heap_base
650          EXPORT  __heap_limit
651      ELSE
652          IMPORT  __use_two_region_memory
653          EXPORT  __user_initial_stackheap
654  __user_initial_stackheap
655          LDR     R0, =HeapMem
656          LDR     R1, =(StackMem + Stack)
657          LDR     R2, =(HeapMem + Heap)
```

```
658          LDR     R3, =StackMem
659          BX      LR
660      ENDIF
661
662  ;*******************************************************************************
663  ;
664  ; Make sure the end of this section is aligned.
665  ;
666  ;*******************************************************************************
667          ALIGN
668
669  ;*******************************************************************************
670  ;
671  ; Tell the assembler that we're done.
672  ;
673  ;*******************************************************************************
674          END
675
```