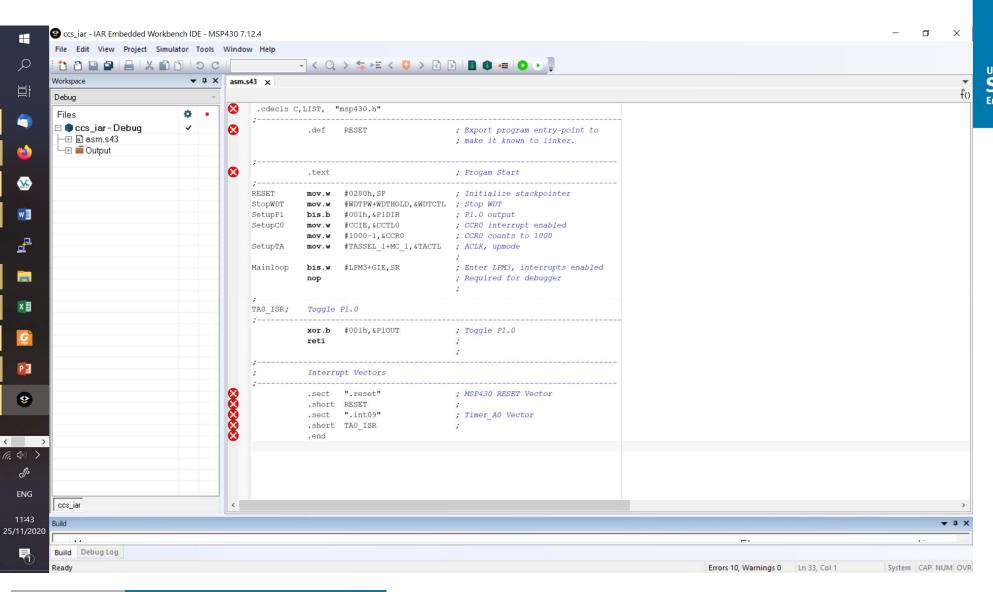


CCS to IAR Assembler File Conversion

Timer_A, Toggle P1.0, CCR0 Up Mode ISR, 32kHz ACLK

```
.cdecls C,LIST, "msp430.h"
           .def RESET
                                       ; Export program entry-point to
                                         ; make it known to linker.
           .text
                                        ; Progam Start
                                        ; Initialize stackpointer
RESET
           mov.w #0280h,SP
           mov.w #WDTPW+WDTHOLD, &WDTCTL ; Stop WDT
StopWDT
                                       ; P1.0 output
SetupP1
           bis.b #001h,&P1DIR
SetupC0
           mov.w #CCIE,&CCTL0
                                       ; CCR0 interrupt enabled
                 #1000-1,&CCR0 ; CCR0 counts to 1000
           mov.w
SetupTA
           mov.w #TASSEL 1+MC 1,&TACTL ; ACLK, upmode
Mainloop
           bis.w #LPM3+GIE,SR
                                        ; Enter LPM3, interrupts enabled
                                         ; Required for debugger
TAO ISR;
           xor.b #001h,&P1OUT
                                       ; Toggle P1.0
           reti
           Interrupt Vectors
           .sect
                 ".reset"
                                        ; MSP430 RESET Vector
           .short RESET
           .sect
                 ".int09"
                                        ; Timer A0 Vector
           .short TA0_ISR
           .end
```





DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

Timer_A, Toggle P1.0, CCR0 Up Mode ISR, 32kHz ACLK

```
.cdecls C,LIST, "msp430.h"
          .def RESET
                                      ; Export program entry-point to
                                        ; make it known to linker.
           .text
                                       ; Progam Start
                                       : Initialize stackpointer
RESET
          mov.w #0280h,SP
          mov.w #WDTPW+WDTHOLD, &WDTCTL ; Stop WDT
StopWDT
                                      ; P1.0 output
SetupP1
          bis.b #001h,&P1DIR
                                     ; CCR0 interrupt enabled
SetupC0
          mov.w #CCIE,&CCTL0
                #1000-1,&CCR0 ; CCR0 counts to 1000
          mov.w
SetupTA
          mov.w
                #TASSEL 1+MC 1,&TACTL ; ACLK, upmode
Mainloop
          bis.w #LPM3+GIE,SR
                                       ; Enter LPM3, interrupts enabled
                                        ; Required for debugger
          Toggle P1.0
TAO ISR;
          xor.b #001h,&P1OUT
                                      ; Toggle P1.0
           reti
           Interrupt Vectors
           .sect ".reset"
                                      ; MSP430 RESET Vector
           .short RESET
                                      ; Timer_A0 Vector
                ".int09"
           .short TAO ISR
```



Empty IAR asm module

```
#include "msp430.h"
                                         ; #define controlled include file
                                         ; module name
        NAME
                main
                                         ; make the main label vissible
        PUBLIC main
                                         ; outside this module
                0FFFEh
        ORG
        DC16
                init
                                         ; set reset vector to 'init' label
                                         ; pre-declaration of segment
        RSEG
                CSTACK
                CODE
                                         ; place program in 'CODE' segment
        RSEG
init:
                #SFE(CSTACK), SP
        MOV
                                         ; set up stack
                                         ; main program
main:
        NOP
                                         ; Stop watchdog timer
        MOV.W
                #WDTPW+WDTHOLD, &WDTCTL
                                         ; jump to current location '$'
        JMP $
                                         ; (endless loop)
        END
```



Empty IAR asm module

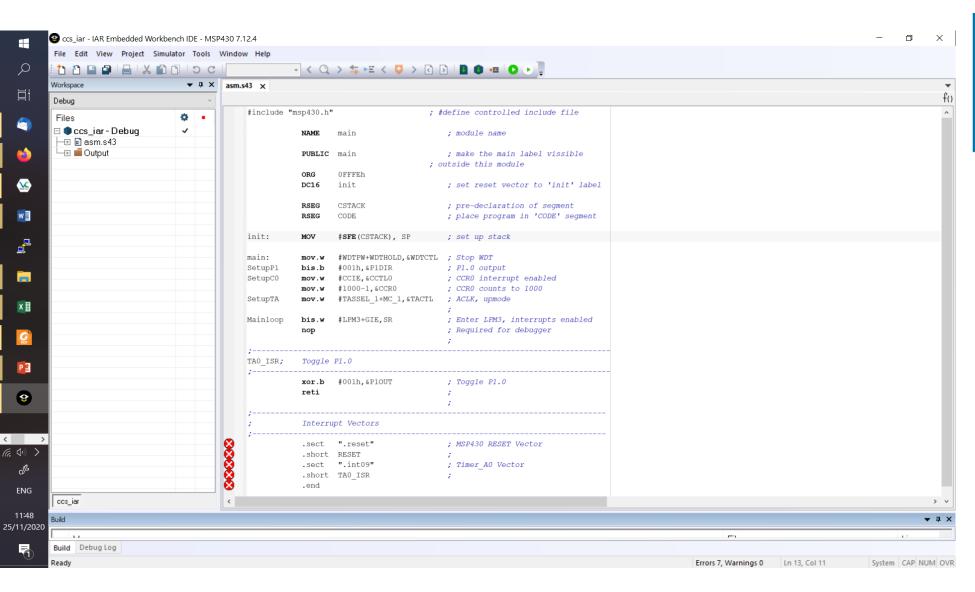
```
#include "msp430.h"
                                         ; #define controlled include file
                                         ; module name
        NAME
                main
                                         ; make the main label vissible
        PUBLIC main
                                         ; outside this module
                0FFFEh
        ORG
        DC16
                init
                                         ; set reset vector to 'init' label
                                         ; pre-declaration of segment
        RSEG
                CSTACK
                CODE
                                         ; place program in 'CODE' segment
        RSEG
init:
                #SFE(CSTACK), SP
        MOV
                                         ; set up stack
                                         ; main program
main:
        NOP
                                         ; Stop watchdog timer
        MOV.W
                #WDTPW+WDTHOLD, &WDTCTL
                                         ; jump to current location '$'
        JMP $
                                         ; (endless loop)
        END
```



Reset is handled differently (called init)

```
#include "msp430.h"
                                         ; #define controlled include file
                                         ; module name
        NAME
                main
                                         ; make the main label vissible
        PUBLIC main
                                         ; outside this module
        ORG
                0FFFEh
        DC16
                init
                                         ; set reset vector to 'init' label
                                         ; pre-declaration of segment
        RSEG
                CSTACK
                                         ; place program in 'CODE' segment
        RSEG
                CODE
init:
        MOV
                #SFE(CSTACK), SP
                                         ; set up stack
                                         ; main program
main:
        NOP
                                         ; Stop watchdog timer
        MOV.W
                #WDTPW+WDTHOLD, &WDTCTL
        JMP $
                                         ; jump to current location '$'
                                         ; (endless loop)
        END
```







Clean up the interrupt vector table definition

```
#include "msp430.h"
                                        ; #define controlled include file
            NAME
                   main
                                            : module name
            PUBLIC main
                                            ; make the main label vissible
                                        ; outside this module
            ORG
                   0FFFEh
            DC16
                   init
                                            ; set reset vector to 'init' label
                   CSTACK
                                            ; pre-declaration of segment
            RSEG
                                            ; place program in 'CODE' segment
            RSEG
                   CODE
init:
            MOV
                   #SFE(CSTACK), SP
                                           ; set up stack
main:
                   #WDTPW+WDTHOLD, &WDTCTL ; Stop WDT
            mov.w
                   #001h,&P1DIR
                                           ; P1.0 output
SetupP1
            bis.b
SetupC0
                   #CCIE,&CCTL0
                                           ; CCR0 interrupt enabled
            mov.w
                   #1000-1,&CCR0
                                           ; CCR0 counts to 1000
                   #TASSEL_1+MC_1,&TACTL ; ACLK, upmode
SetupTA
Mainloop
                   #LPM3+GIE,SR
                                            ; Enter LPM3, interrupts enabled
            bis.w
                                            ; Required for debugger
            xor.b #001h,&P1OUT
                                            ; Toggle P1.0
           reti
            COMMON INTVEC
; Defaults to starting at 0
            ORG TIMERO_AO_VECTOR ; Interrupt vector for Timer
            DC16 TAO ISR; Point to the Timer interrupt routine
            END
```



From msp430g2553.h

```
/***********************
* Interrupt Vectors (offset from 0xFFE0)
#define TRAPINT VECTOR
                         (0u * 2u) /* 0xFFE0 TRAPINT */
#define PORT1_VECTOR
                         (2u * 2u) /* 0xFFE4 Port 1 */
#define PORT2_VECTOR
                         (3u * 2u) /* 0xFFE6 Port 2 */
#define ADC10 VECTOR
                         (5u * 2u) /* 0xFFEA ADC10 */
#define USCIABOTX_VECTOR
                         (6u * 2u) /* 0xFFEC USCI A0/B0 Transmit */
                         (7u * 2u) /* 0xFFEE USCI A0/B0 Receive */
#define USCIABORX_VECTOR
#define TIMER0_A1_VECTOR
                         (8u * 2u) /* 0xFFF0 Timer0)A CC1, TA0 */
                         (9u * 2u) /* 0xFFF2 Timer0_A CCO */
#define TIMER0_A0_VECTOR
#define WDT_VECTOR
                         (10u * 2u) /* 0xFFF4 Watchdog Timer */
                        (11u * 2u) /* 0xFFF6 Comparator A */
#define COMPARATORA_VECTOR
#define TIMER1 A1 VECTOR
                         (12u * 2u) /* 0xFFF8 Timer1 A CC1-4, TA1 */
#define TIMER1 A0 VECTOR
                         (13u * 2u) /* 0xFFFA Timer1 A CC0 */
#define NMI VECTOR
                         (14u * 2u) /* 0xFFFC Non-maskable */
                         (15u * 2u) /* 0xFFFE Reset [Highest Priority] */
#define RESET VECTOR
```



Neater

```
#include "msp430.h"
                                        ; #define controlled include file
            NAME
                                             ; module name
                    main
            PUBLIC
                   main
                                             ; make the main label vissible
            RSEG
                                             ; pre-declaration of segment
                    CSTACK
            RSEG
                                            ; place program in 'CODE' segment
                    CODE
init:
            MOV
                    #SFE(CSTACK), SP
                                             ; set up stack
main:
                    #WDTPW+WDTHOLD, &WDTCTL ; Stop WDT
            mov.w
            bis.b
                    #001h,&P1DIR
SetupP1
                                             ; P1.0 output
SetupC0
            mov.w
                    #CCIE,&CCTL0
                                            ; CCR0 interrupt enabled
            mov.w
                    #1000-1,&CCR0
                                             ; CCR0 counts to 1000
                    #TASSEL 1+MC 1,&TACTL
                                            ; ACLK, upmode
SetupTA
            mov.w
Mainloop
                                             ; Enter LPM3, interrupts enabled
            bis.w
                    #LPM3+GIE,SR
                                             ; Required for debugger
TA0_ISR;
            Toggle P1.0
            xor.b #001h,&P10UT
                                            ; Toggle P1.0
            reti
            Interrupt Vectors
            COMMON INTVEC
            ORG TIMERO_AO_VECTOR
                                            ; Interrupt vector for Timer
                                            ; Point to the Timer interrupt routine
            DC16 TA0_ISR
            ORG RESET_VECTOR
                                            ; Interrupt vector for reset
            DC16 init
                                            ; Point to initialisation code
            END
```



DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING