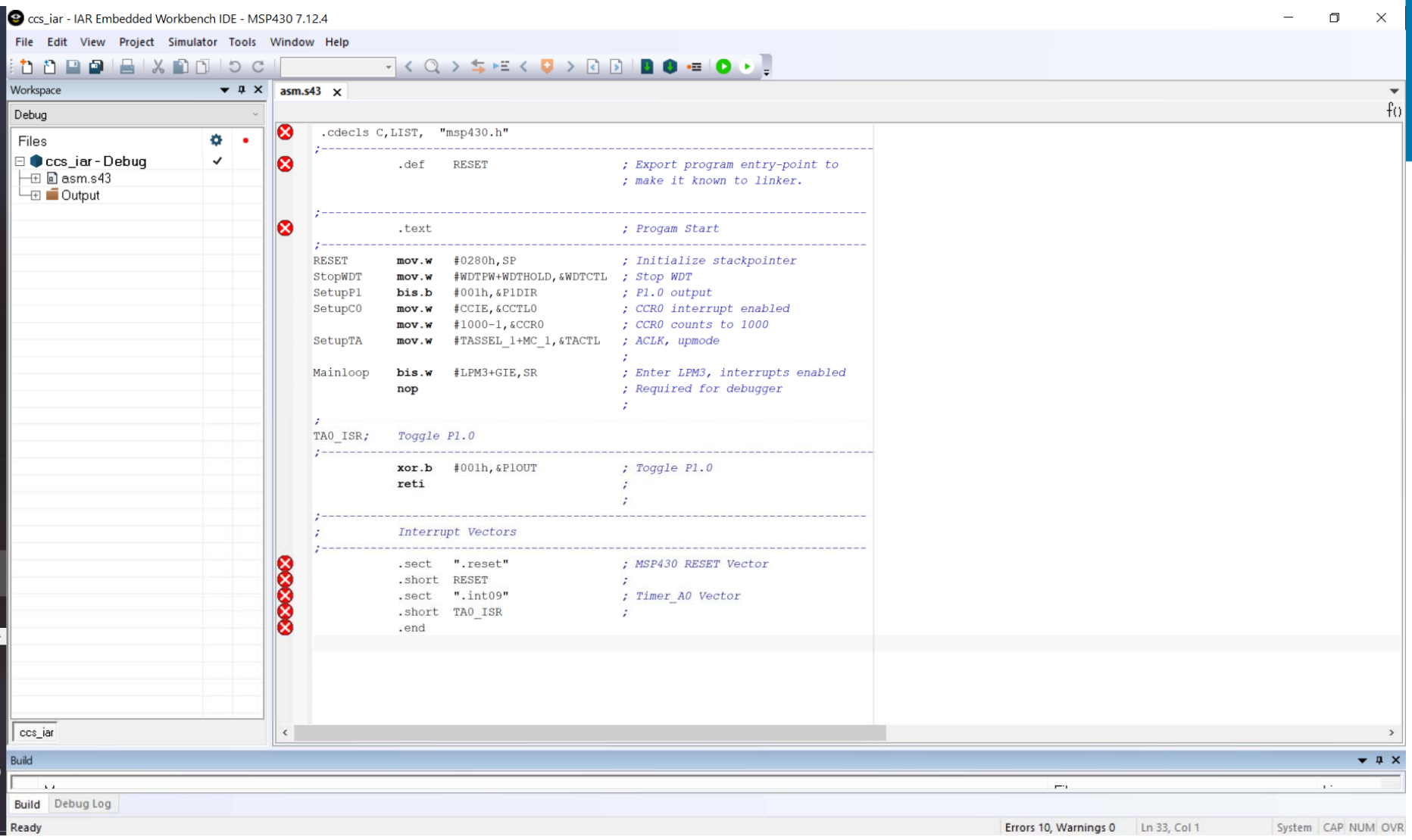
University of Strathclyde Engineering

# CCS to IAR Assembler File Conversion

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
;-------------------------------------------------------------------------------
RESET       mov.w   #0280h,SP               ; Initialize stackpointer
StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL  ; Stop WDT
SetupP1     bis.b   #001h,&P1DIR            ; P1.0 output
SetupC0     mov.w   #CCIE,&CCTL0            ; CCR0 interrupt enabled
            mov.w   #1000-1,&CCR0           ; CCR0 counts to 1000
SetupTA     mov.w   #TASSEL_1+MC_1,&TACTL   ; ACLK, upmode
                                            ;


Mainloop    bis.w   #LPM3+GIE,SR            ; Enter LPM3, interrupts enabled
            nop                             ; Required for debugger
                                            ;
;-------------------------------------------------------------------------------
TA0_ISR;    Toggle P1.0
;-------------------------------------------------------------------------------
            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
```




University of Strathclyde Engineering

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
;-------------------------------------------------------------------------------
RESET       mov.w   #0280h,SP               ; Initialize stackpointer
StopWDT     mov.w   #WDTPW+WDTHOLD,&WDTCTL  ; Stop WDT
SetupP1     bis.b   #001h,&P1DIR            ; P1.0 output
SetupC0     mov.w   #CCIE,&CCTL0            ; CCR0 interrupt enabled
            mov.w   #1000-1,&CCR0           ; CCR0 counts to 1000
SetupTA     mov.w   #TASSEL_1+MC_1,&TACTL   ; ACLK, upmode
                                            ;


Mainloop    bis.w   #LPM3+GIE,SR            ; Enter LPM3, interrupts enabled
            nop                             ; Required for debugger
                                            ;
;-------------------------------------------------------------------------------
TA0_ISR;    Toggle P1.0
;-------------------------------------------------------------------------------
            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
```

# Empty IAR asm module

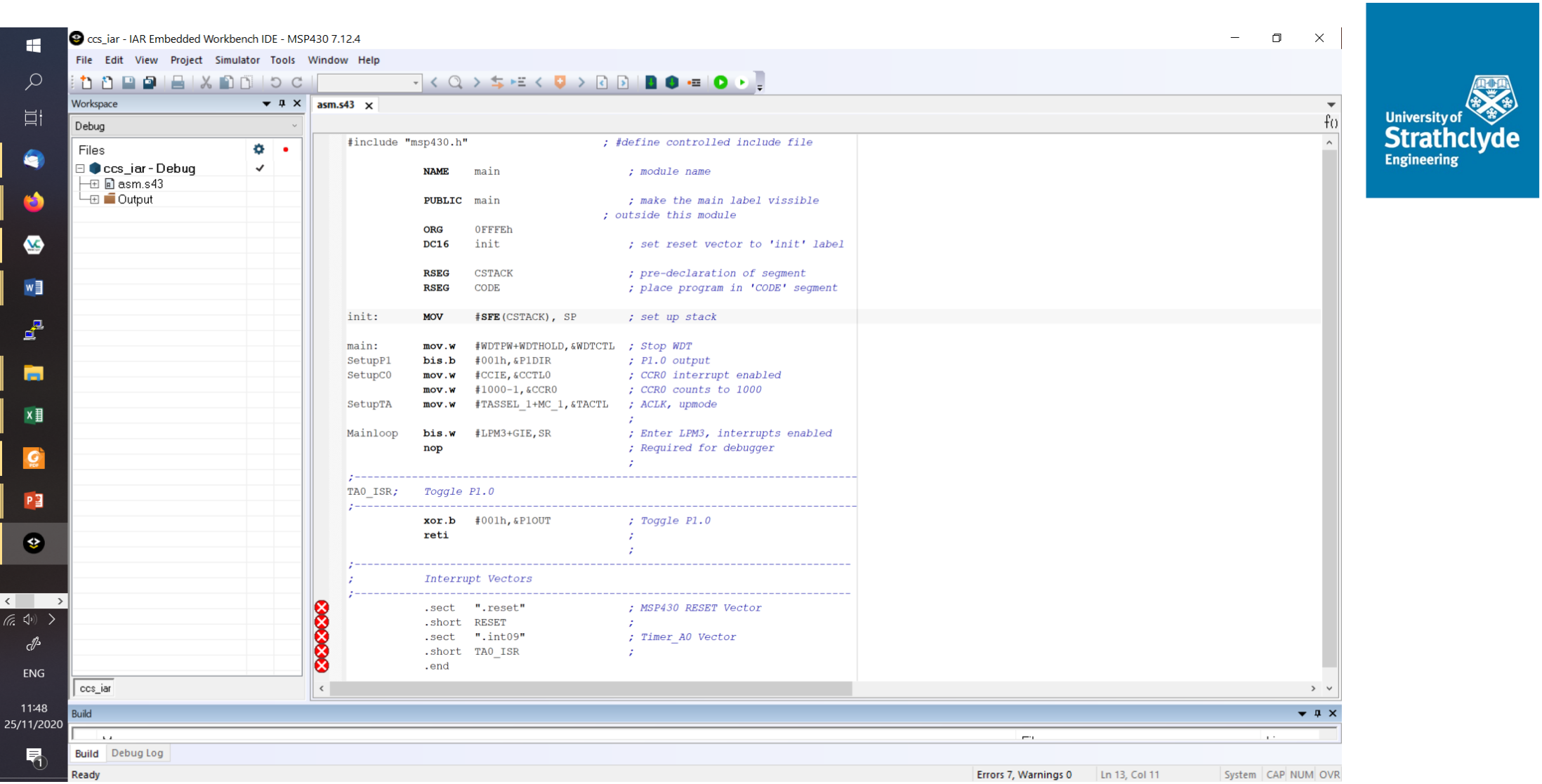
```
#include "msp430.h"                     ; #define controlled include file

        NAME    main                    ; module name

        PUBLIC  main                    ; make the main label vissible
                                        ; outside this module
        ORG     0FFFEh
        DC16    init                    ; set reset vector to 'init' label

        RSEG    CSTACK                  ; pre-declaration of segment
        RSEG    CODE                    ; place program in 'CODE' segment

init:   MOV     #SFE(CSTACK), SP        ; set up stack

main:   NOP                             ; main program
        MOV.W   #WDTPW+WDTHOLD,&WDTCTL  ; Stop watchdog timer
        JMP $                           ; jump to current location '$'
                                        ; (endless loop)
        END
```

# Empty IAR asm module

```
#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

        RSEG    CSTACK                  ; pre-declaration of segment
        RSEG    CODE                    ; place program in 'CODE' segment

init:   MOV     #SFE(CSTACK), SP        ; set up stack

main:   NOP                             ; main program
        MOV.W   #WDTPW+WDTHOLD,&WDTCTL  ; Stop watchdog timer
        JMP $                           ; jump to current location '$'
                                        ; (endless loop)
        END
```

# Reset is handled differently (called init)

```
#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

        RSEG    CSTACK                  ; pre-declaration of segment
        RSEG    CODE                    ; place program in 'CODE' segment

init:   MOV     #SFE(CSTACK), SP        ; set up stack

main:   NOP                             ; main program
        MOV.W   #WDTPW+WDTHOLD,&WDTCTL  ; Stop watchdog timer
        JMP $                           ; jump to current location '$'
                                        ; (endless loop)

        END
```




University of Strathclyde Engineering

DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

# 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

            RSEG    CSTACK                 ; pre-declaration of segment
            RSEG    CODE                   ; place program in 'CODE' segment

init:       MOV     #SFE(CSTACK), SP       ; set up stack

main:       mov.w   #WDTPW+WDTHOLD,&WDTCTL ; Stop WDT
SetupP1     bis.b   #001h,&P1DIR           ; P1.0 output
SetupC0     mov.w   #CCIE,&CCTL0           ; CCR0 interrupt enabled
            mov.w   #1000-1,&CCR0          ; CCR0 counts to 1000
SetupTA     mov.w   #TASSEL_1+MC_1,&TACTL  ; ACLK, upmode
                                           ;

Mainloop    bis.w   #LPM3+GIE,SR           ; Enter LPM3, interrupts enabled
            nop                            ; Required for debugger
                                           ;
;-------------------------------------------------------------------------------
TA0_ISR;    Toggle P1.0
;-------------------------------------------------------------------------------
            xor.b   #001h,&P1OUT           ; Toggle P1.0
            reti                           ;
                                           ;
;-------------------------------------------------------------------------------
;           Interrupt Vectors
;-------------------------------------------------------------------------------
            COMMON INTVEC
; Defaults to starting at 0

            ORG TIMER0_A0_VECTOR ; Interrupt vector for Timer
            DC16 TA0_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 USCIAB0TX_VECTOR    (6u * 2u)  /* 0xFFEC USCI A0/B0 Transmit */
#define USCIAB0RX_VECTOR    (7u * 2u)  /* 0xFFEE USCI A0/B0 Receive */
#define TIMER0_A1_VECTOR    (8u * 2u)  /* 0xFFF0 Timer0)A CC1, TA0 */
#define TIMER0_A0_VECTOR    (9u * 2u)  /* 0xFFF2 Timer0_A CC0 */
#define WDT_VECTOR          (10u * 2u) /* 0xFFF4 Watchdog Timer */
#define COMPARATORA_VECTOR  (11u * 2u) /* 0xFFF6 Comparator A */
#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 */
#define RESET_VECTOR        (15u * 2u) /* 0xFFFE Reset [Highest Priority] */
```

# Neater

```
#include "msp430.h"                    ; #define controlled include file

            NAME    main                    ; module name

            PUBLIC  main                    ; make the main label vissible

            RSEG    CSTACK                  ; pre-declaration of segment
            RSEG    CODE                    ; place program in 'CODE' segment

init:       MOV     #SFE(CSTACK), SP        ; set up stack

main:       mov.w   #WDTPW+WDTHOLD,&WDTCTL  ; Stop WDT
SetupP1     bis.b   #001h,&P1DIR            ; P1.0 output
SetupC0     mov.w   #CCIE,&CCTL0            ; CCR0 interrupt enabled
            mov.w   #1000-1,&CCR0           ; CCR0 counts to 1000
SetupTA     mov.w   #TASSEL_1+MC_1,&TACTL   ; ACLK, upmode
                                            ;

Mainloop    bis.w   #LPM3+GIE,SR            ; Enter LPM3, interrupts enabled
            nop                             ; Required for debugger
                                            ;
;-------------------------------------------------------------------------------
TA0_ISR;    Toggle P1.0
;-------------------------------------------------------------------------------
            xor.b   #001h,&P1OUT            ; Toggle P1.0
            reti                            ;
                                            ;
;-------------------------------------------------------------------------------
;           Interrupt Vectors
;-------------------------------------------------------------------------------
            COMMON INTVEC

            ORG TIMER0_A0_VECTOR            ; Interrupt vector for Timer
            DC16 TA0_ISR                    ; Point to the Timer interrupt routine
            ORG RESET_VECTOR                ; Interrupt vector for reset
            DC16 init                       ; Point to initialisation code

            END
```