HCS12C128 Board I/O Information:

User I/O

USER	Ref Des	Signal	Description
1	SW1	PP0/KWP0/PWM0/MISO1	Push Button Switch 1 (SW1)
2	SW2	PP1/KWP1/PWM1/MOSI1	Push Button Switch 2 (SW2)
3	SW3-1	PB0/**ADDR0/DATA0**	DIP Switch 1 (DIP-SW1)
4	SW3-2	PB1/**ADDR1/DATA1**	DIP Switch (DIP-SW2)
5	SW3-3	PB2**ADDR2/DATA2**	DIP Switch (DIP-SW3)
6	SW3-4	PB3/**ADDR3/DATA3**	DIP Switch (DIP-SW4)
7	LED1	PB4/**ADDR4/DATA4**	Green LED (LED1)
8	LED2	PB5/**ADDR5/DATA5**	Green LED (LED2)
9	LED3	PB6/**ADDR6/DATA6**	Green LED (LED3)
10	LED4	PB7/**ADDR7/DATA7**	Green LED (LED4)
11	RV1	PAD05/AN05	Potentiometer (RV1)
12	RZ1	PAD04/AN04	Light Sensor (RZ1)

****

HCS12C128 Board

http://www.axman.com/content/aps12slk-csmb12-module

APS12SLK (CSMB12) Module

Users Manual:
APS12UG_0.pdf

Schematic:
CSMB12_SCH_A1_0_0.pdf

Other File:
Quick Start
APS12-C128 CodeWarrior LCD Example Code
SLK12S CodeWarrior Demo Code
DBUG12 Reference Guide, V4
APS12_Silk_A.pdf

Links:
http://www.freescale.com/universityprograms
http://www.lulu.com/content/paperback-book/microcontroller-programming-for-engin...
Professor Mazidi Labs

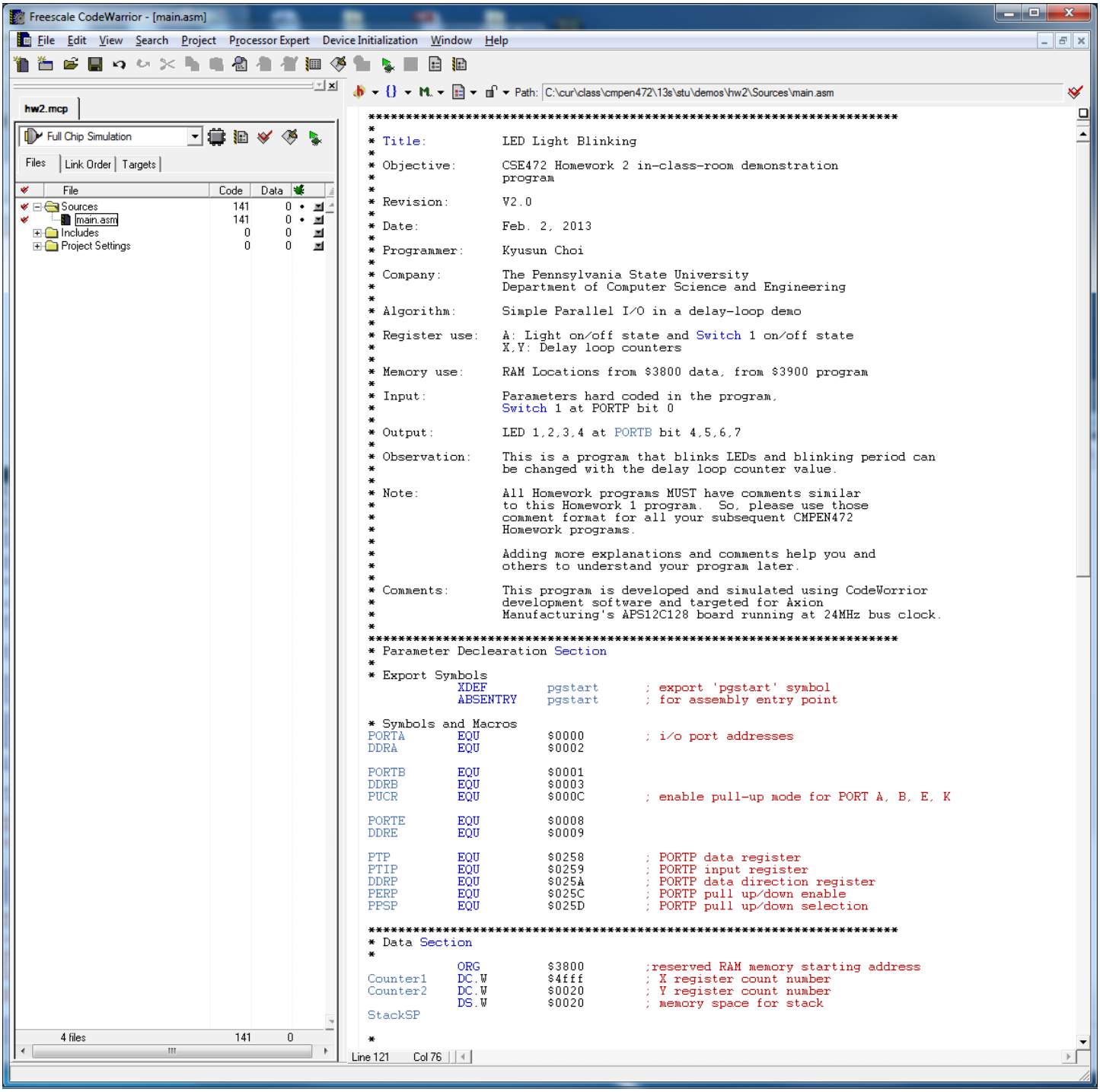
FAQ:
This module supports the MC9S12C128, MC9S12DT256, and MC9S12XDT512 MCU's
This module applies an integrated TBDML fully compatible with Freescale CodeWarrior

Q. APS12 board will not execute application code when powered from barrel connector.
A. When powering the board externally through the barrel connector, ensure a shunt is installed in the VR1 and VB positions. Alternately, remove both option headers at BDM_EN.

NOTE: The option headers at BDM_EN must be reinstalled in order to reestablish BDM communications.
Educational

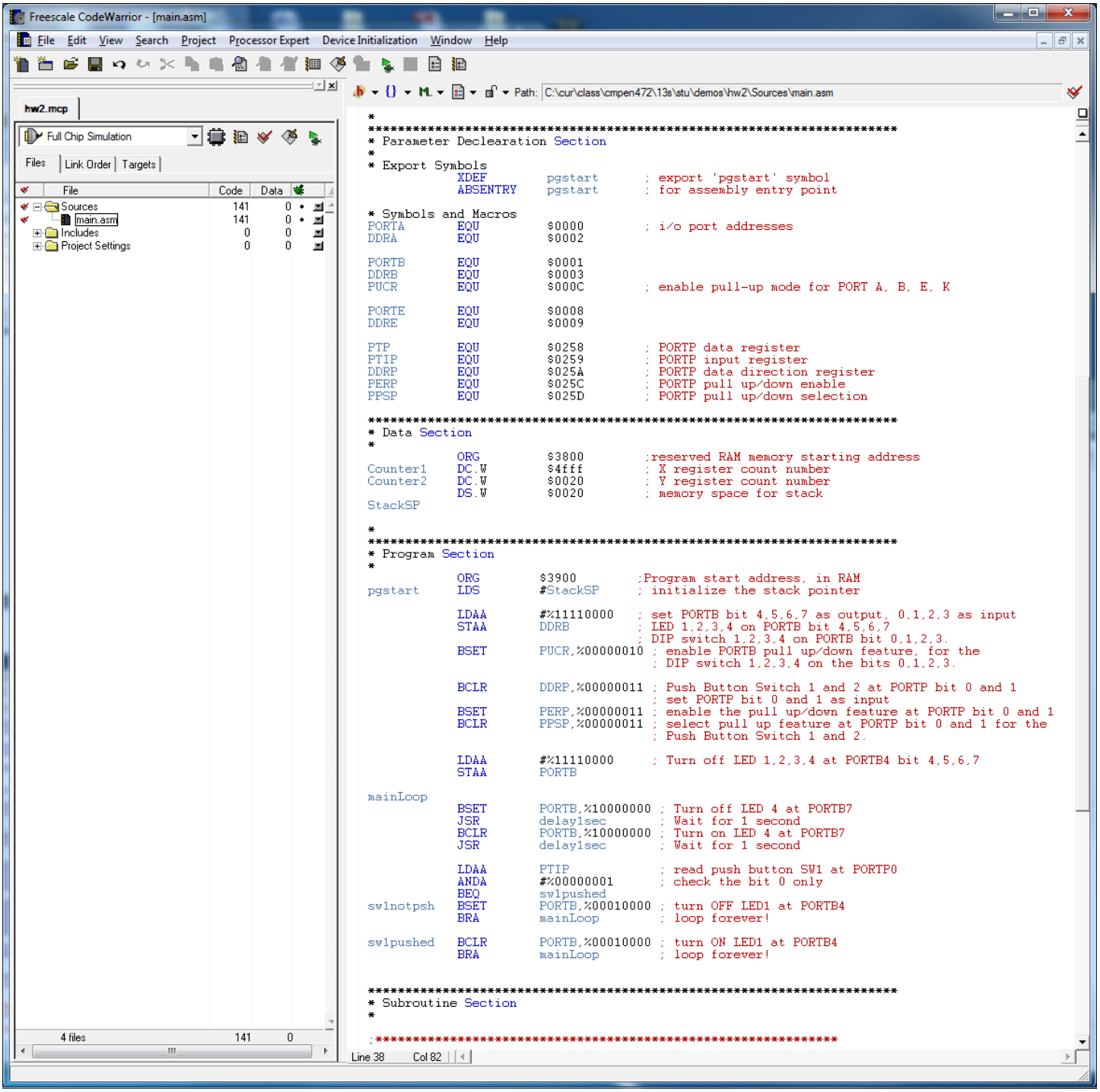
****

Sample program (Homework 2) on the HCS12C128 Board

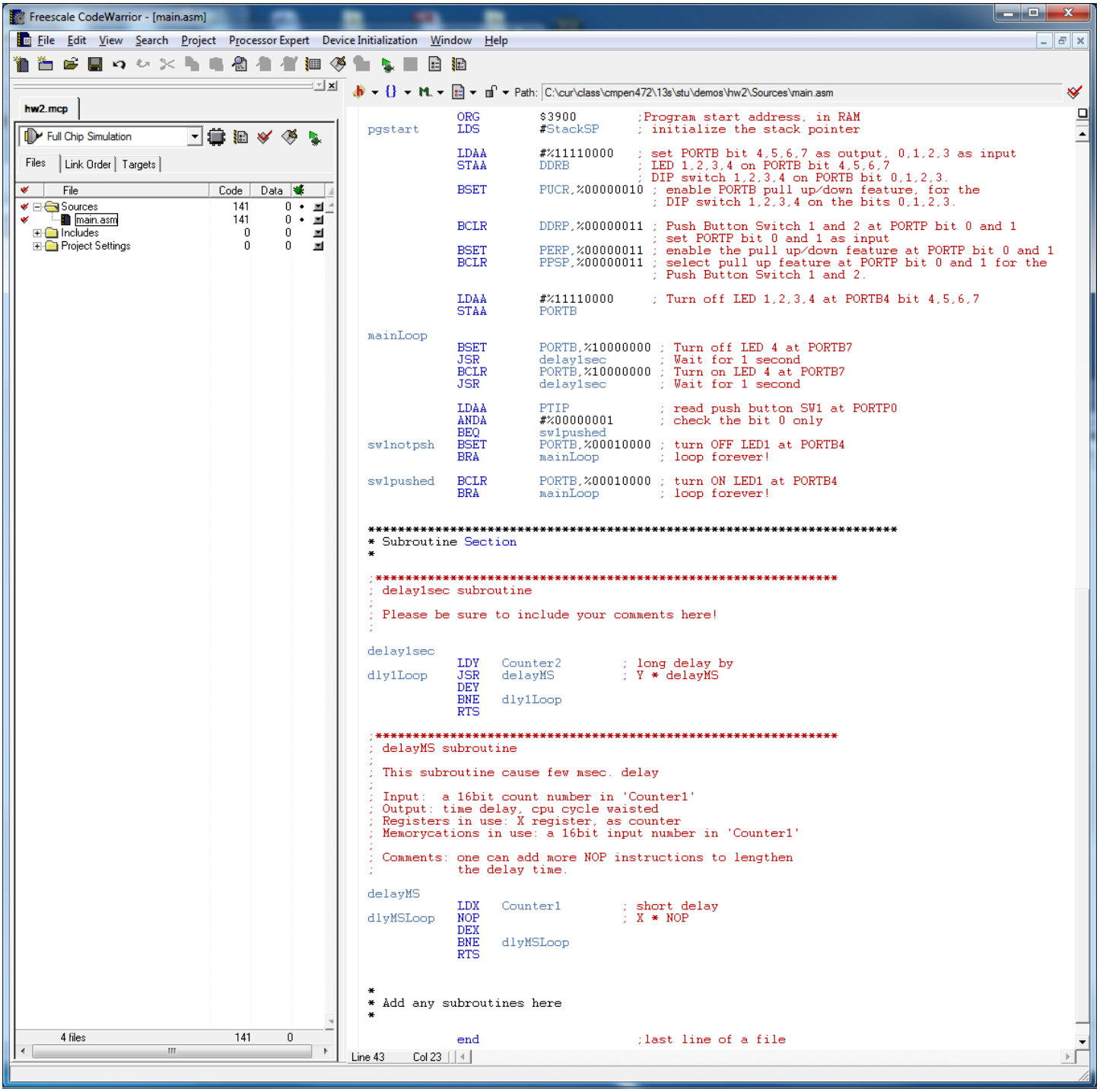



```
*****************************************************************
* Title:          LED Light Blinking
*
* Objective:      CSE472 Homework 2 in-class-room demonstration
*                 program
*
* Revision:       V2.0
*
* Date:           Feb. 2, 2013
*
* Programmer:     Kyusun Choi
*
* Company:        The Pennsylvania State University
*                 Department of Computer Science and Engineering
*
* Algorithm:      Simple Parallel I/O in a delay-loop demo
*
* Register use:   A: Light on/off state and Switch 1 on/off state
*                 X,Y: Delay loop counters
*
* Memory use:     RAM Locations from $3800 data, from $3900 program
*
* Input:          Parameters hard coded in the program,
*                 Switch 1 at PORTP bit 0
*
* Output:         LED 1,2,3,4 at PORTB bit 4,5,6,7
*
* Observation:    This is a program that blinks LEDs and blinking period can
*                 be changed with the delay loop counter value.
*
* Note:           All Homework programs MUST have comments similar
*                 to this Homework 1 program.  So, please use those
*                 comment format for all your subsequent CMPEN472
*                 Homework programs.
*
*                 Adding more explanations and comments help you and
*                 others to understand your program later.
*
* Comments:       This program is developed and simulated using CodeWorrior
*                 development software and targeted for Axion
*                 Manufacturing's APS12C128 board running at 24MHz bus clock.
*
*****************************************************************
* Parameter Declearation Section
*
* Export Symbols
            XDEF        pgstart       ; export 'pgstart' symbol
            ABSENTRY    pgstart       ; for assembly entry point

* Symbols and Macros
PORTA       EQU         $0000         ; i/o port addresses
DDRA        EQU         $0002

PORTB       EQU         $0001
DDRB        EQU         $0003
PUCR        EQU         $000C         ; enable pull-up mode for PORT A, B, E, K

PORTE       EQU         $0008
DDRE        EQU         $0009

PTP         EQU         $0258         ; PORTP data register
PTIP        EQU         $0259         ; PORTP input register
DDRP        EQU         $025A         ; PORTP data direction register
PERP        EQU         $025C         ; PORTP pull up/down enable
PPSP        EQU         $025D         ; PORTP pull up/down selection

*****************************************************************
* Data Section
*
            ORG         $3800         ;reserved RAM memory starting address
Counter1    DC.W        $4fff         ; X register count number
Counter2    DC.W        $0020         ; Y register count number
            DS.W        $0020         ; memory space for stack
StackSP

*
```

For this year, we are using MC9S12C128 board, the program starts at $3100 and data section starts at $3000. So, use 'ORG $3000' and 'ORG $3100'. (Previous years, we used MC9S12C32 and the address was $3800 and $3900.)




```
*
*****************************************************************
* Parameter Declearation Section
*
* Export Symbols
            XDEF        pgstart       ; export 'pgstart' symbol
            ABSENTRY    pgstart       ; for assembly entry point

* Symbols and Macros
PORTA       EQU         $0000         ; i/o port addresses
DDRA        EQU         $0002

PORTB       EQU         $0001
DDRB        EQU         $0003
PUCR        EQU         $000C         ; enable pull-up mode for PORT A, B, E, K

PORTE       EQU         $0008
DDRE        EQU         $0009

PTP         EQU         $0258         ; PORTP data register
PTIP        EQU         $0259         ; PORTP input register
DDRP        EQU         $025A         ; PORTP data direction register
PERP        EQU         $025C         ; PORTP pull up/down enable
PPSP        EQU         $025D         ; PORTP pull up/down selection

*****************************************************************
* Data Section
*
            ORG         $3800         ;reserved RAM memory starting address
Counter1    DC.W        $4fff         ; X register count number
Counter2    DC.W        $0020         ; Y register count number
            DS.W        $0020         ; memory space for stack
StackSP

*
*****************************************************************
* Program Section
*
            ORG         $3900         ;Program start address, in RAM
pgstart     LDS         #StackSP      ; initialize the stack pointer

            LDAA        #%11110000    ; set PORTB bit 4,5,6,7 as output, 0,1,2,3 as input
            STAA        DDRB          ; LED 1,2,3,4 on PORTB bit 4,5,6,7
                                      ; DIP switch 1,2,3,4 on PORTB bit 0,1,2,3.
            BSET        PUCR,%00000010 ; enable PORTB pull up/down feature, for the
                                       ; DIP switch 1,2,3,4 on the bits 0,1,2,3.

            BCLR        DDRP,%00000011 ; Push Button Switch 1 and 2 at PORTP bit 0 and 1
                                       ; set PORTP bit 0 and 1 as input
            BSET        PERP,%00000011 ; enable the pull up/down feature at PORTP bit 0 and 1
            BCLR        PPSP,%00000011 ; select pull up feature at PORTP bit 0 and 1 for the
                                       ; Push Button Switch 1 and 2.

            LDAA        #%11110000     ; Turn off LED 1,2,3,4 at PORTB4 bit 4,5,6,7
            STAA        PORTB

mainLoop
            BSET        PORTB,%10000000 ; Turn off LED 4 at PORTB7
            JSR         delay1sec       ; Wait for 1 second
            BCLR        PORTB,%10000000 ; Turn on LED 4 at PORTB7
            JSR         delay1sec       ; Wait for 1 second

            LDAA        PTIP            ; read push button SW1 at PORTP0
            ANDA        #%00000001      ; check the bit 0 only
            BEQ         sw1pushed
sw1notpsh   BSET        PORTB,%00010000 ; turn OFF LED1 at PORTB4
            BRA         mainLoop        ; loop forever!

sw1pushed   BCLR        PORTB,%00010000 ; turn ON LED1 at PORTB4
            BRA         mainLoop        ; loop forever!


*****************************************************************
* Subroutine Section
*

;*************************************************************
```


Freescale CodeWarrior - [main.asm]
File Edit View Search Project Processor Expert Device Initialization Window Help
hw2.mcp
Full Chip Simulation
Files Link Order Targets
File Code Data
Sources 141 0
main.asm 141 0
Includes 0 0
Project Settings 0 0
4 files 141 0
Path: C:\cur\class\cmpen472\13s\stu\demos\hw2\Sources\main.asm
ORG $3900 ;Program start address, in RAM
pgstart LDS #StackSP ; initialize the stack pointer
LDAA #%11110000 ; set PORTB bit 4,5,6,7 as output, 0,1,2,3 as input
STAA DDRB ; LED 1,2,3,4 on PORTB bit 4,5,6,7
; DIP switch 1,2,3,4 on PORTB bit 0,1,2,3.
BSET PUCR,%00000010 ; enable PORTB pull up/down feature, for the
; DIP switch 1,2,3,4 on the bits 0,1,2,3.
BCLR DDRP,%00000011 ; Push Button Switch 1 and 2 at PORTP bit 0 and 1
; set PORTP bit 0 and 1 as input
BSET PERP,%00000011 ; enable the pull up/down feature at PORTP bit 0 and 1
BCLR PPSP,%00000011 ; select pull up feature at PORTP bit 0 and 1 for the
; Push Button Switch 1 and 2.
LDAA #%11110000 ; Turn off LED 1,2,3,4 at PORTB4 bit 4,5,6,7
STAA PORTB
mainLoop
BSET PORTB,%10000000 ; Turn off LED 4 at PORTB7
JSR delay1sec ; Wait for 1 second
BCLR PORTB,%10000000 ; Turn on LED 4 at PORTB7
JSR delay1sec ; Wait for 1 second
LDAA PTIP ; read push button SW1 at PORTP0
ANDA #%00000001 ; check the bit 0 only
BEQ sw1pushed
sw1notpsh BSET PORTB,%00010000 ; turn OFF LED1 at PORTB4
BRA mainLoop ; loop forever!
sw1pushed BCLR PORTB,%00010000 ; turn ON LED1 at PORTB4
BRA mainLoop ; loop forever!
* Subroutine Section
; delay1sec subroutine
; Please be sure to include your comments here!
delay1sec
LDY Counter2 ; long delay by
dly1Loop JSR delayMS ; Y * delayMS
DEY
BNE dly1Loop
RTS
; delayMS subroutine
; This subroutine cause few msec. delay
; Input: a 16bit count number in 'Counter1'
; Output: time delay, cpu cycle waisted
; Registers in use: X register, as counter
; Memorycations in use: a 16bit input number in 'Counter1'
; Comments: one can add more NOP instructions to lengthen
; the delay time.
delayMS
LDX Counter1 ; short delay
dlyMSLoop NOP ; X * NOP
DEX
BNE dlyMSLoop
RTS
* Add any subroutines here
end ;last line of a file
Line 43 Col 23

****