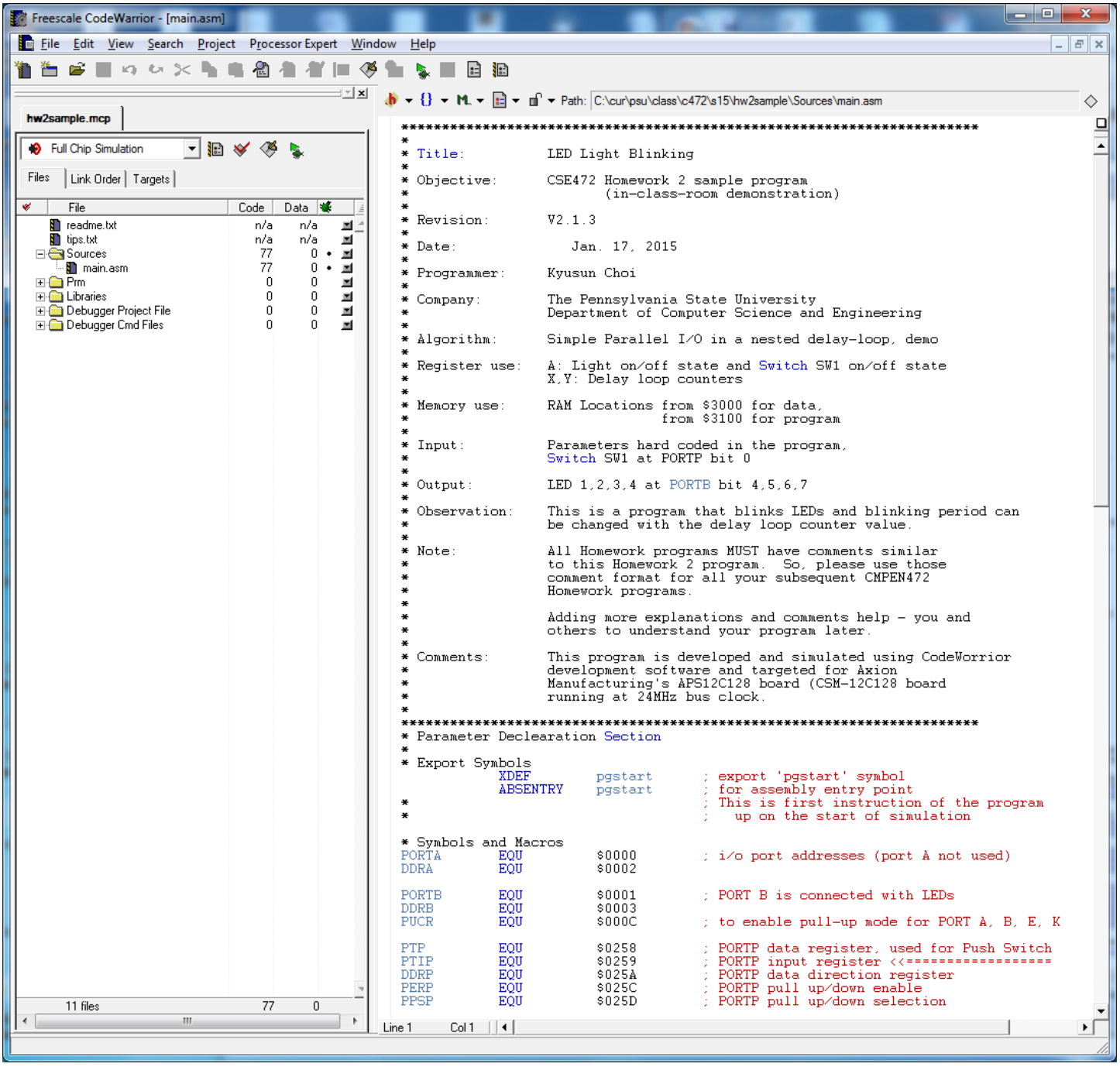
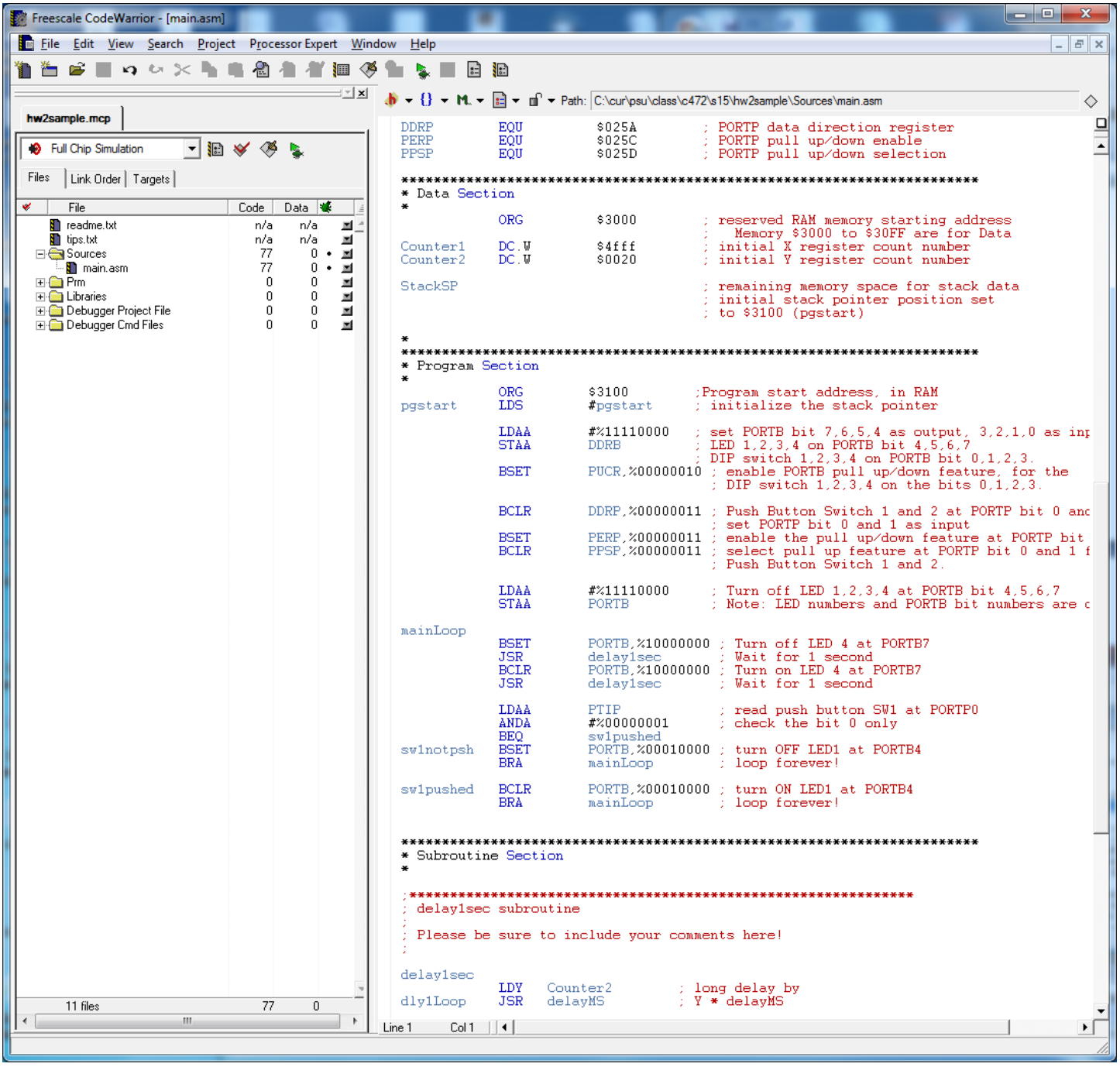
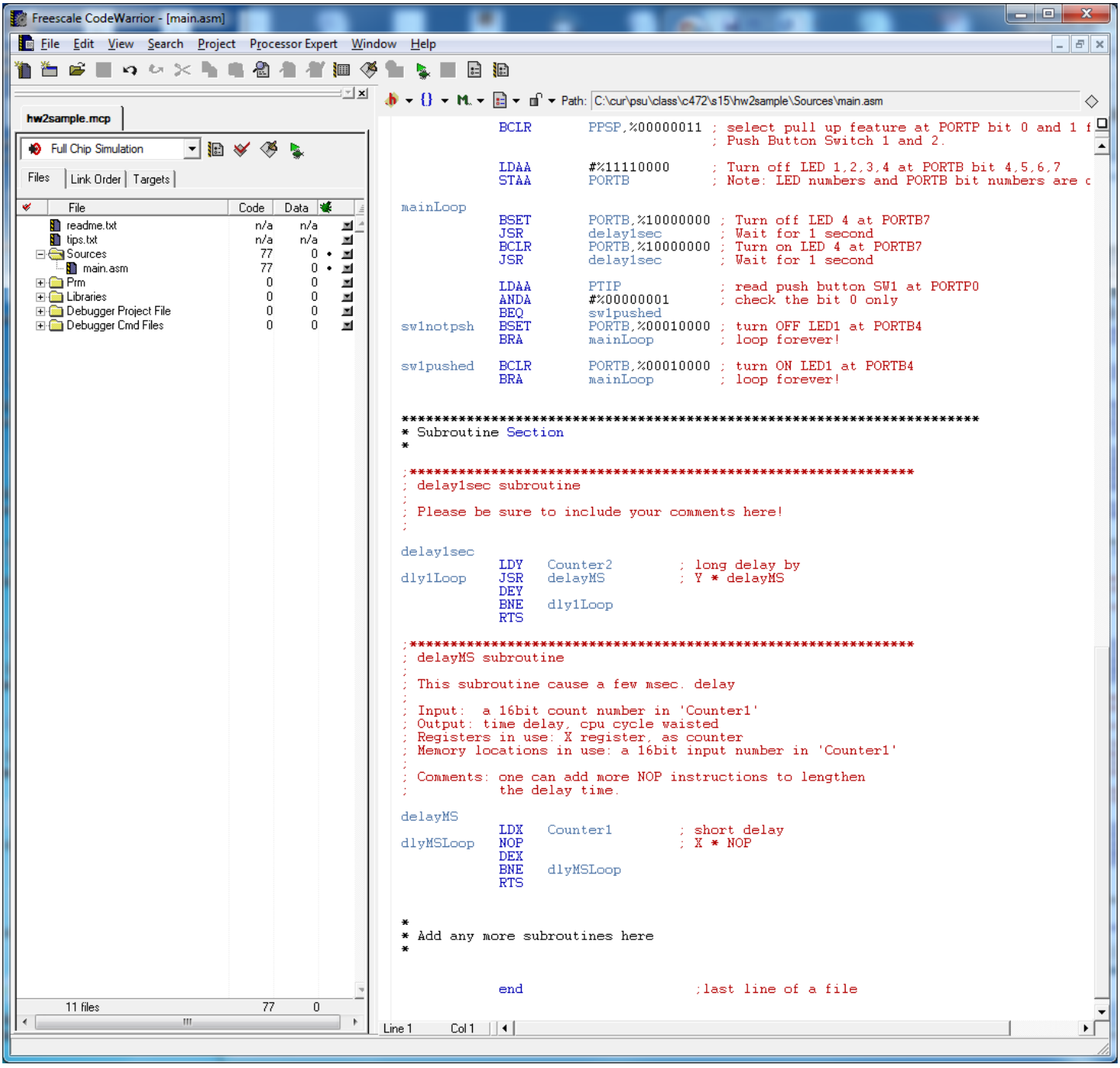
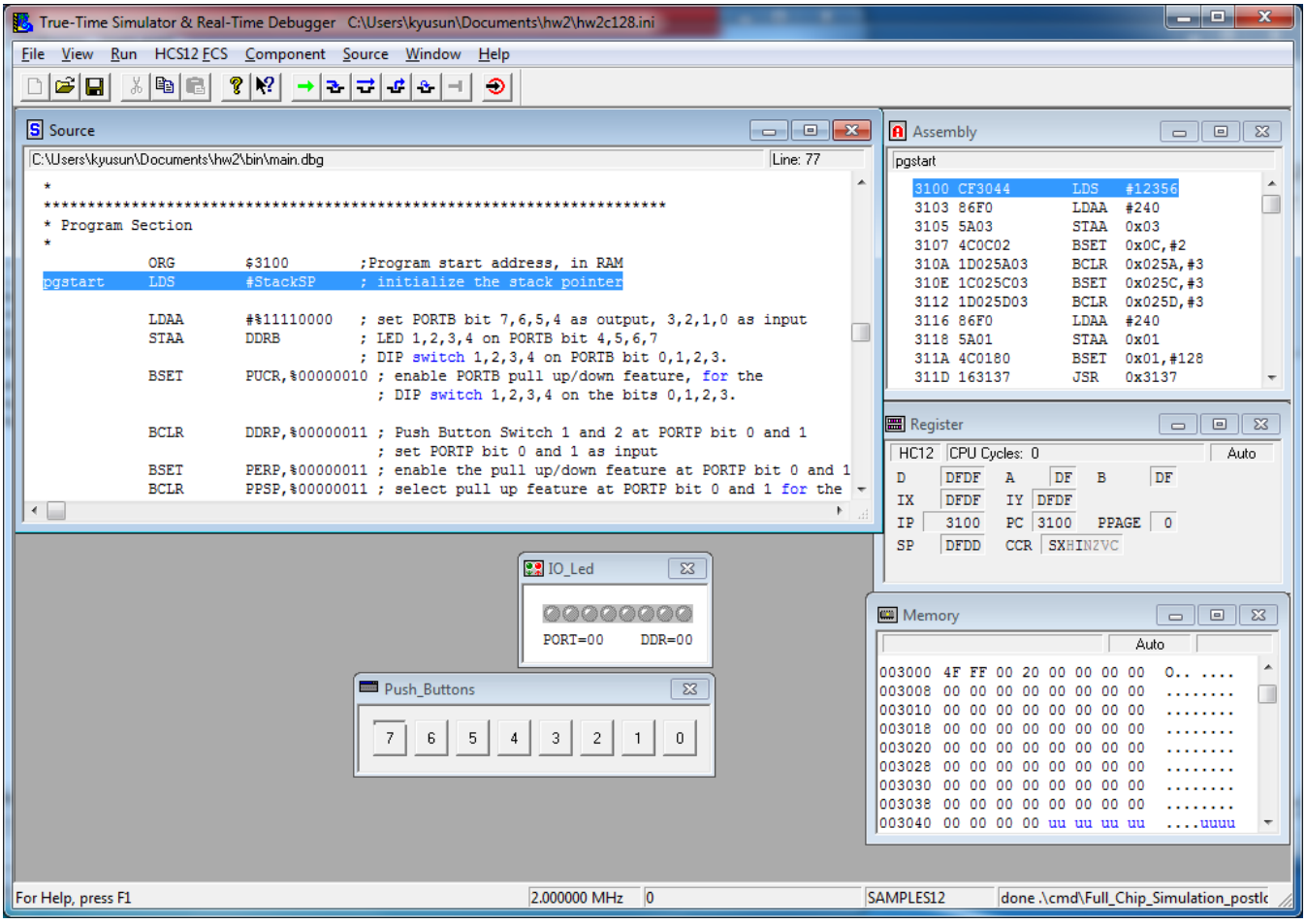
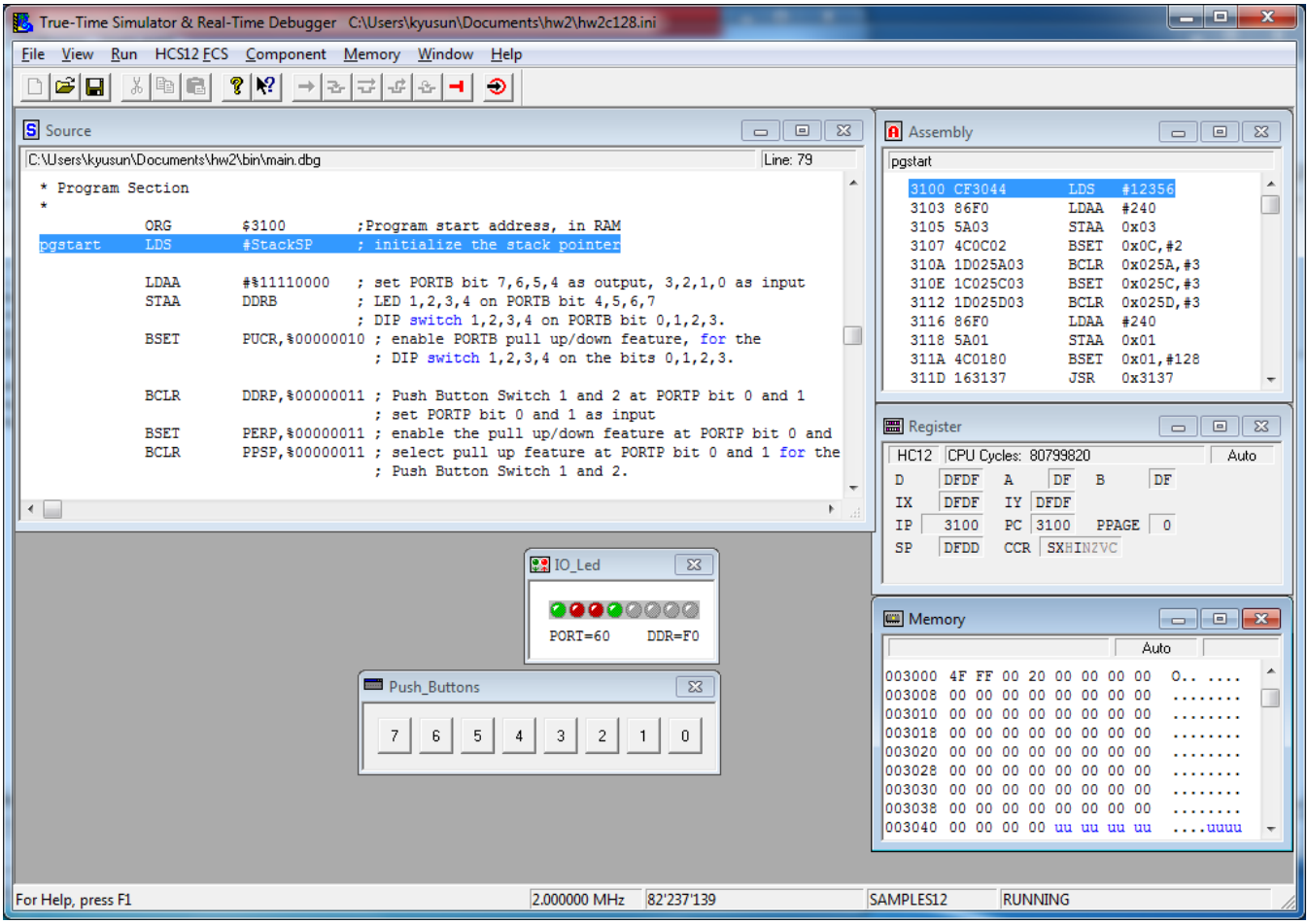



```
*****************************************************************
*
* Title:          LED Light Blinking
*
* Objective:      CSE472 Homework 2 sample program
*                       (in-class-room demonstration)
*
* Revision:       V2.1.3
*
* Date:              Jan. 17, 2015
*
* Programmer:     Kyusun Choi
*
* Company:        The Pennsylvania State University
*                 Department of Computer Science and Engineering
*
* Algorithm:      Simple Parallel I/O in a nested delay-loop, demo
*
* Register use:   A: Light on/off state and Switch SW1 on/off state
*                 X,Y: Delay loop counters
*
* Memory use:     RAM Locations from $3000 for data,
*                                   from $3100 for program
*
* Input:          Parameters hard coded in the program,
*                 Switch SW1 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 2 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 (CSM-12C128 board
*                 running at 24MHz bus clock.
*
*****************************************************************
* Parameter Declearation Section
*
* Export Symbols
            XDEF        pgstart       ; export 'pgstart' symbol
            ABSENTRY    pgstart       ; for assembly entry point
*                                     ; This is first instruction of the program
*                                     ;   up on the start of simulation

* Symbols and Macros
PORTA       EQU         $0000         ; i/o port addresses (port A not used)
DDRA        EQU         $0002

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

PTP         EQU         $0258         ; PORTP data register, used for Push Switch
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
```


Freescale CodeWarrior - [main.asm]
File Edit View Search Project Processor Expert Window Help
hw2sample.mcp
Full Chip Simulation
Files Link Order Targets
File Code Data
readme.txt n/a n/a
tips.txt n/a n/a
Sources 77 0
main.asm 77 0
Prm 0 0
Libraries 0 0
Debugger Project File 0 0
Debugger Cmd Files 0 0
11 files 77 0
Path: C:\cur\psu\class\c472\s15\hw2sample\Sources\main.asm
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 $3000 ; reserved RAM memory starting address
; Memory $3000 to $30FF are for Data
Counter1 DC.W $4fff ; initial X register count number
Counter2 DC.W $0020 ; initial Y register count number
StackSP ; remaining memory space for stack data
; initial stack pointer position set
; to $3100 (pgstart)
* Program Section
ORG $3100 ;Program start address, in RAM
pgstart LDS #pgstart ; initialize the stack pointer
LDAA #%11110000 ; set PORTB bit 7,6,5,4 as output, 3,2,1,0 as inp
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
; set PORTP bit 0 and 1 as input
BSET PERP,%00000011 ; enable the pull up/down feature at PORTP bit
BCLR PPSP,%00000011 ; select pull up feature at PORTP bit 0 and 1 f
; Push Button Switch 1 and 2.
LDAA #%11110000 ; Turn off LED 1,2,3,4 at PORTB bit 4,5,6,7
STAA PORTB ; Note: LED numbers and PORTB bit numbers are
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
Line 1 Col 1


Freescale CodeWarrior - [main.asm]
File Edit View Search Project Processor Expert Window Help
hw2sample.mcp
Full Chip Simulation
Files Link Order Targets
File Code Data
readme.txt n/a n/a
tips.txt n/a n/a
Sources 77 0
main.asm 77 0
Prm 0 0
Libraries 0 0
Debugger Project File 0 0
Debugger Cmd Files 0 0
11 files 77 0
Path: C:\cur\psu\class\c472\s15\hw2sample\Sources\main.asm
BCLR PPSP,%00000011 ; select pull up feature at PORTP bit 0 and 1
; Push Button Switch 1 and 2.
LDAA #%11110000 ; Turn off LED 1,2,3,4 at PORTB bit 4,5,6,7
STAA PORTB ; Note: LED numbers and PORTB bit numbers are
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 a few msec. delay
; Input: a 16bit count number in 'Counter1'
; Output: time delay, cpu cycle waisted
; Registers in use: X register, as counter
; Memory locations 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 more subroutines here
end ;last line of a file
Line 1 Col 1


True-Time Simulator & Real-Time Debugger C:\Users\kyusun\Documents\hw2\hw2c128.ini
File View Run HCS12 FCS Component Source Window Help
Source
C:\Users\kyusun\Documents\hw2\bin\main.dbg
Line: 77
* Program Section
ORG $3100 ;Program start address, in RAM
pgstart LDS #StackSP ; initialize the stack pointer
LDAA #%11110000 ; set PORTB bit 7,6,5,4 as output, 3,2,1,0 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
Assembly
pgstart
3100 CF3044 LDS #12356
3103 86F0 LDAA #240
3105 5A03 STAA 0x03
3107 4C0C02 BSET 0x0C,#2
310A 1D025A03 BCLR 0x025A,#3
310E 1C025C03 BSET 0x025C,#3
3112 1D025D03 BCLR 0x025D,#3
3116 86F0 LDAA #240
3118 5A01 STAA 0x01
311A 4C0180 BSET 0x01,#128
311D 163137 JSR 0x3137
Register
HC12 CPU Cycles: 0 Auto
D DFDF A DF B DF
IX DFDF IY DFDF
IP 3100 PC 3100 PPAGE 0
SP DFDD CCR SXHIN2VC
IO_Led
PORT=00 DDR=00
Push_Buttons
7 6 5 4 3 2 1 0
Memory
Auto
003000 4F FF 00 20 00 00 00 00 O.. ....
003008 00 00 00 00 00 00 00 00 ....
003010 00 00 00 00 00 00 00 00 ....
003018 00 00 00 00 00 00 00 00 ....
003020 00 00 00 00 00 00 00 00 ....
003028 00 00 00 00 00 00 00 00 ....
003030 00 00 00 00 00 00 00 00 ....
003038 00 00 00 00 00 00 00 00 ....
003040 00 00 00 00 uu uu uu uu ....uuuu
For Help, press F1
2.000000 MHz
0
SAMPLES12
done .\cmd\Full_Chip_Simulation_postlc


True-Time Simulator & Real-Time Debugger C:\Users\kyusun\Documents\hw2\hw2c128.ini
File View Run HCS12 FCS Component Memory Window Help
Source
C:\Users\kyusun\Documents\hw2\bin\main.dbg
Line: 79
* Program Section
*
ORG $3100 ;Program start address, in RAM
pgstart LDS #StackSP ; initialize the stack pointer
LDAA #%11110000 ; set PORTB bit 7,6,5,4 as output, 3,2,1,0 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
BCLR PPSP,%00000011 ; select pull up feature at PORTP bit 0 and 1 for the
; Push Button Switch 1 and 2.
Assembly
pgstart
3100 CF3044 LDS #12356
3103 86F0 LDAA #240
3105 5A03 STAA 0x03
3107 4C0C02 BSET 0x0C,#2
310A 1D025A03 BCLR 0x025A,#3
310E 1C025C03 BSET 0x025C,#3
3112 1D025D03 BCLR 0x025D,#3
3116 86F0 LDAA #240
3118 5A01 STAA 0x01
311A 4C0180 BSET 0x01,#128
311D 163137 JSR 0x3137
Register
HC12 CPU Cycles: 80799820 Auto
D DFDF A DF B DF
IX DFDF IY DFDF
IP 3100 PC 3100 PPAGE 0
SP DFDD CCR SXHINZVC
IO_Led
PORT=60 DDR=F0
Push_Buttons
7 6 5 4 3 2 1 0
Memory
Auto
003000 4F FF 00 20 00 00 00 00 O.. ....
003008 00 00 00 00 00 00 00 00 ....
003010 00 00 00 00 00 00 00 00 ....
003018 00 00 00 00 00 00 00 00 ....
003020 00 00 00 00 00 00 00 00 ....
003028 00 00 00 00 00 00 00 00 ....
003030 00 00 00 00 00 00 00 00 ....
003038 00 00 00 00 00 00 00 00 ....
003040 00 00 00 00 uu uu uu uu ....uuuu
For Help, press F1
2.000000 MHz
82'237'139
SAMPLES12
RUNNING