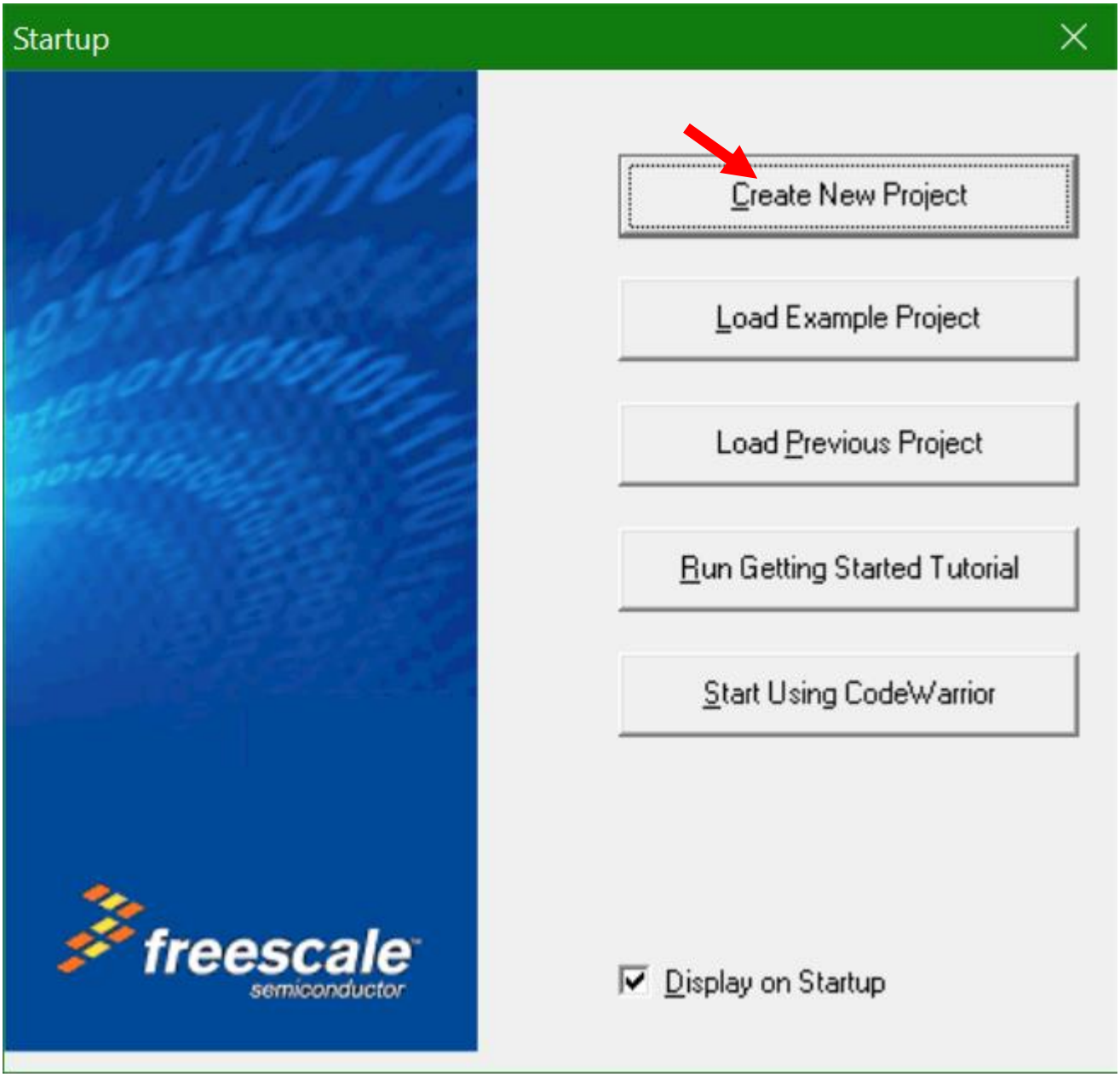
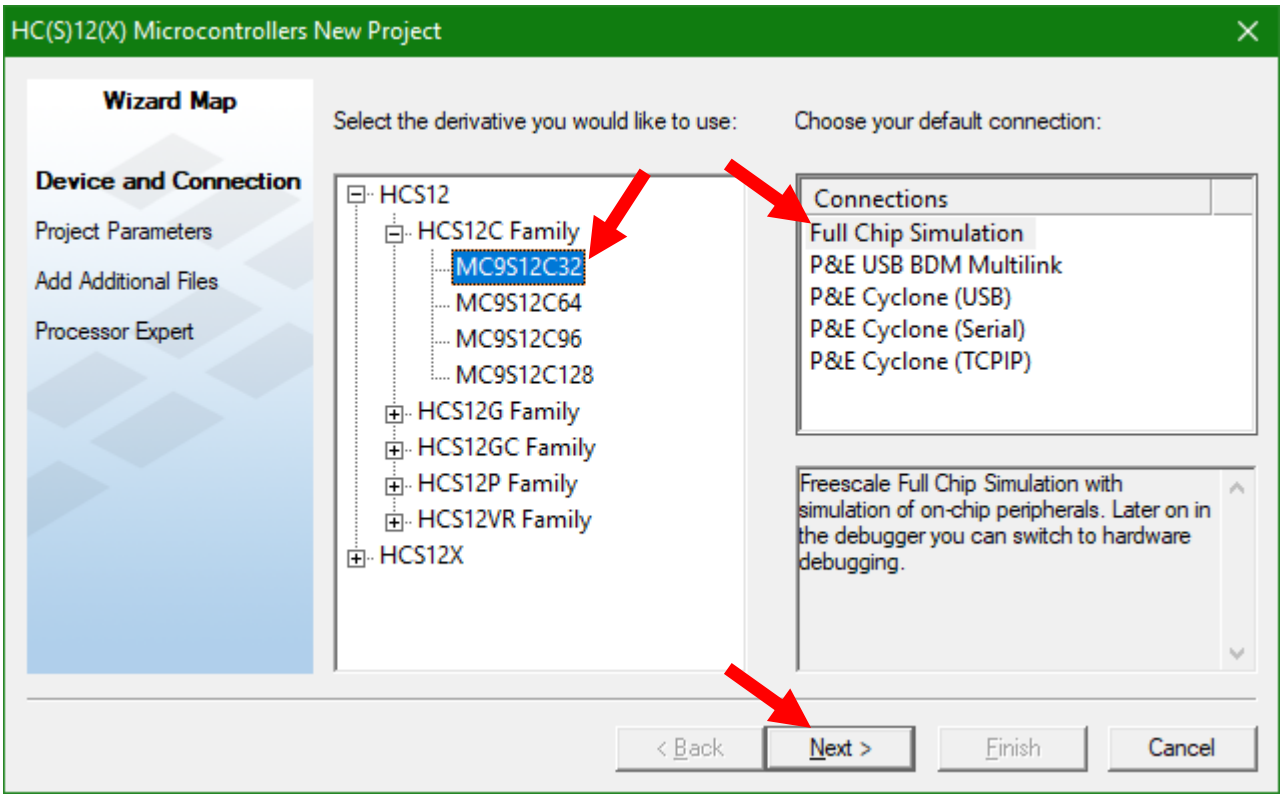
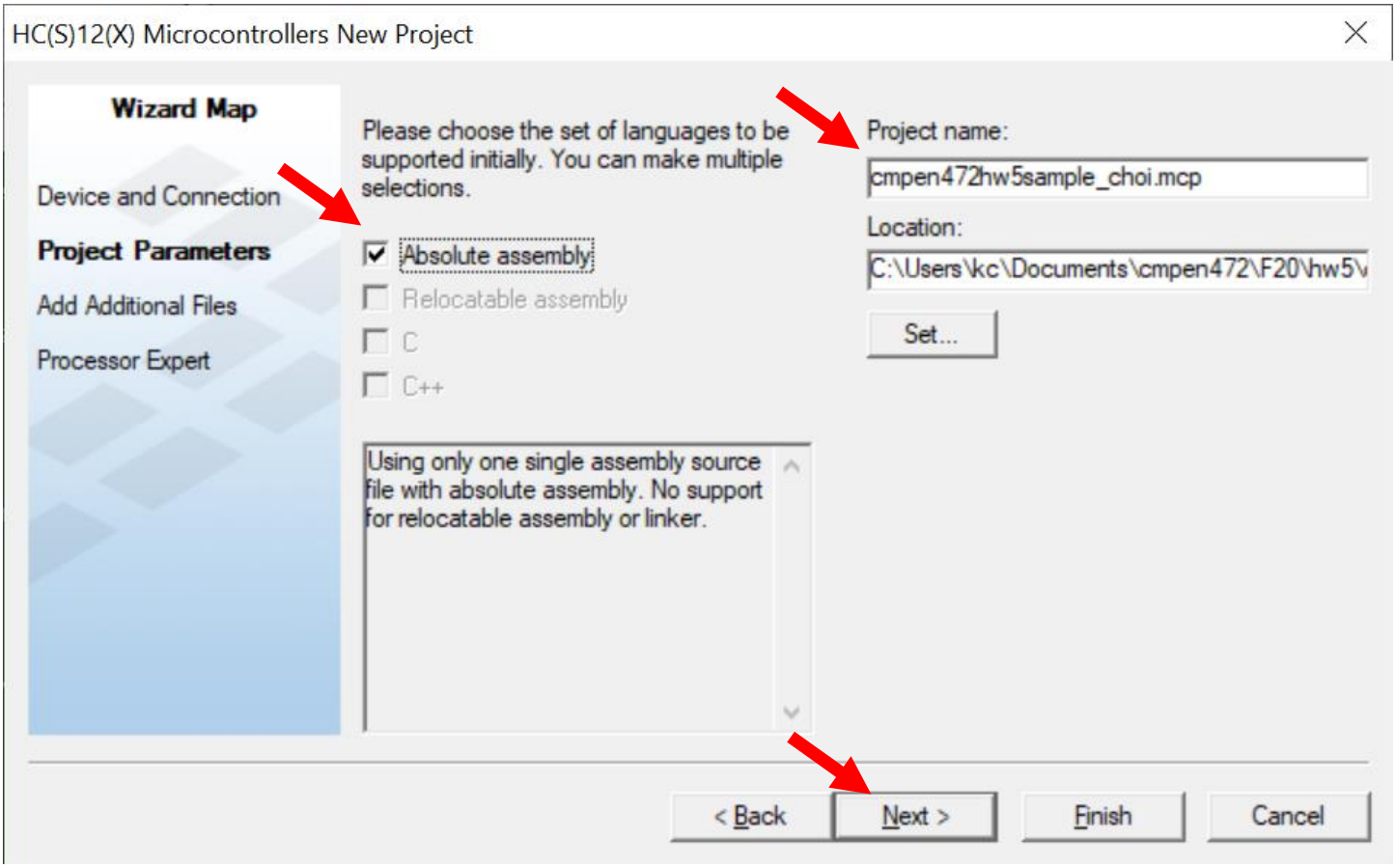
CMPEN 472, **CodeWarrior Full Chip Simulator (Debugger) Guide, Step 3 – Terminal Component**

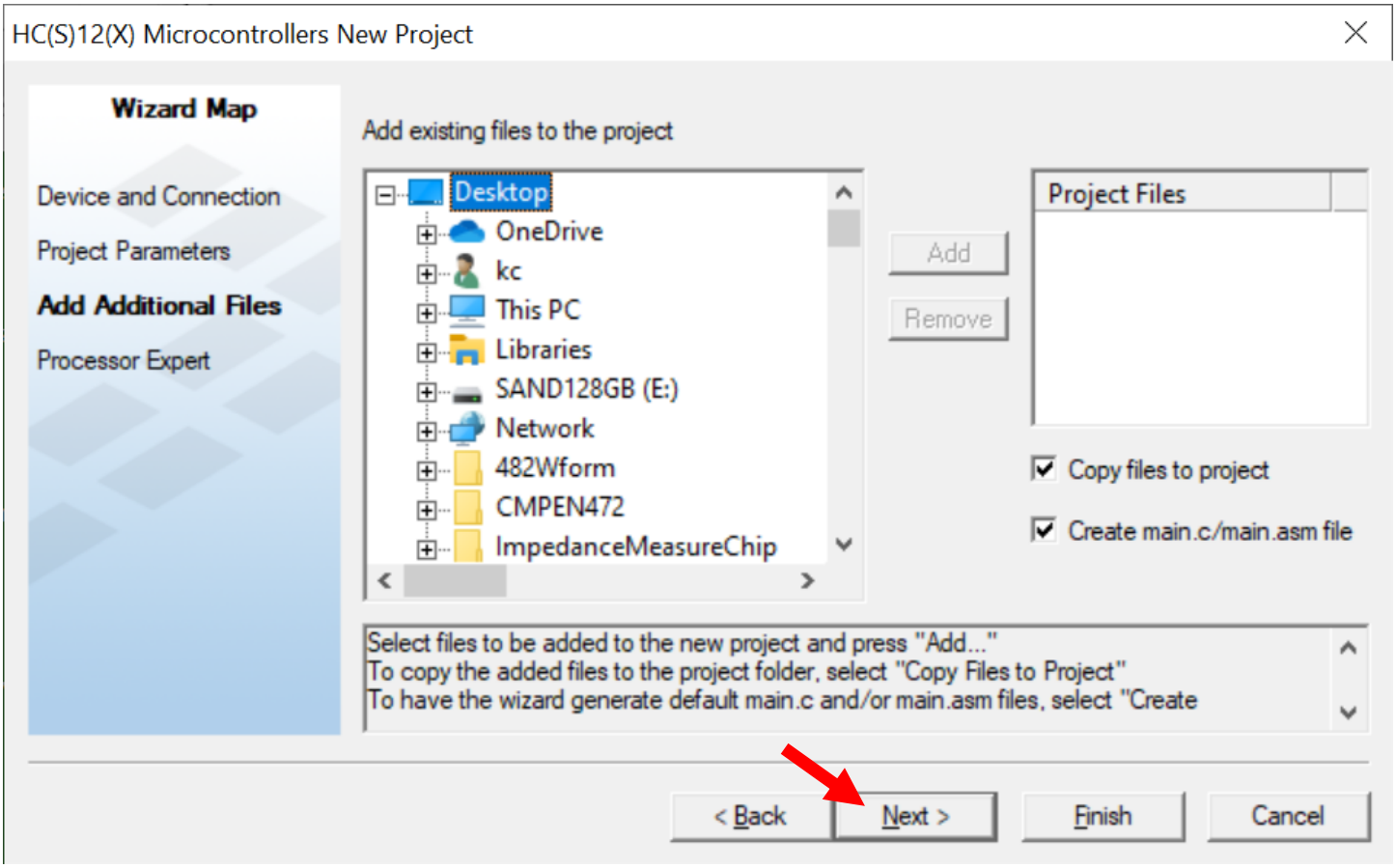
First step: Create CodeWarrior New Project for the Homework 5 Sample program. You can do this by few ways: (1) by starting CodeWarrior IDE or (2) by selecting **New Project . . .** in **File** menu of the CodeWarrior IDE - if you already have CodeWarrior IDE running for another project.

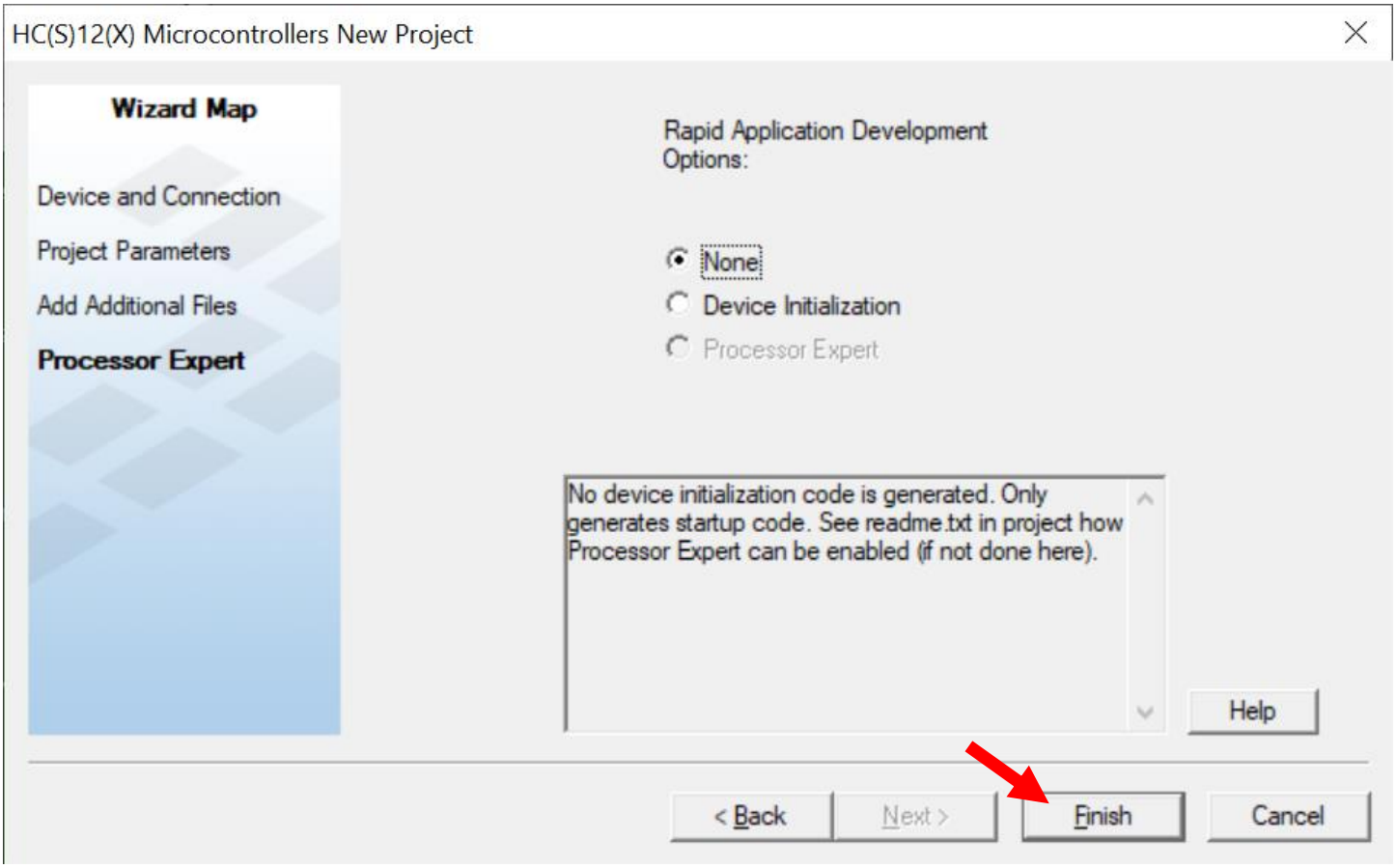


You MUST select MC9S12C32 chip (not MC9S12C128 chip) for proper simulation. (No worries, you will be able to do everything for MC9S12C128 for this class even if you select MC9S12C32 chip.)

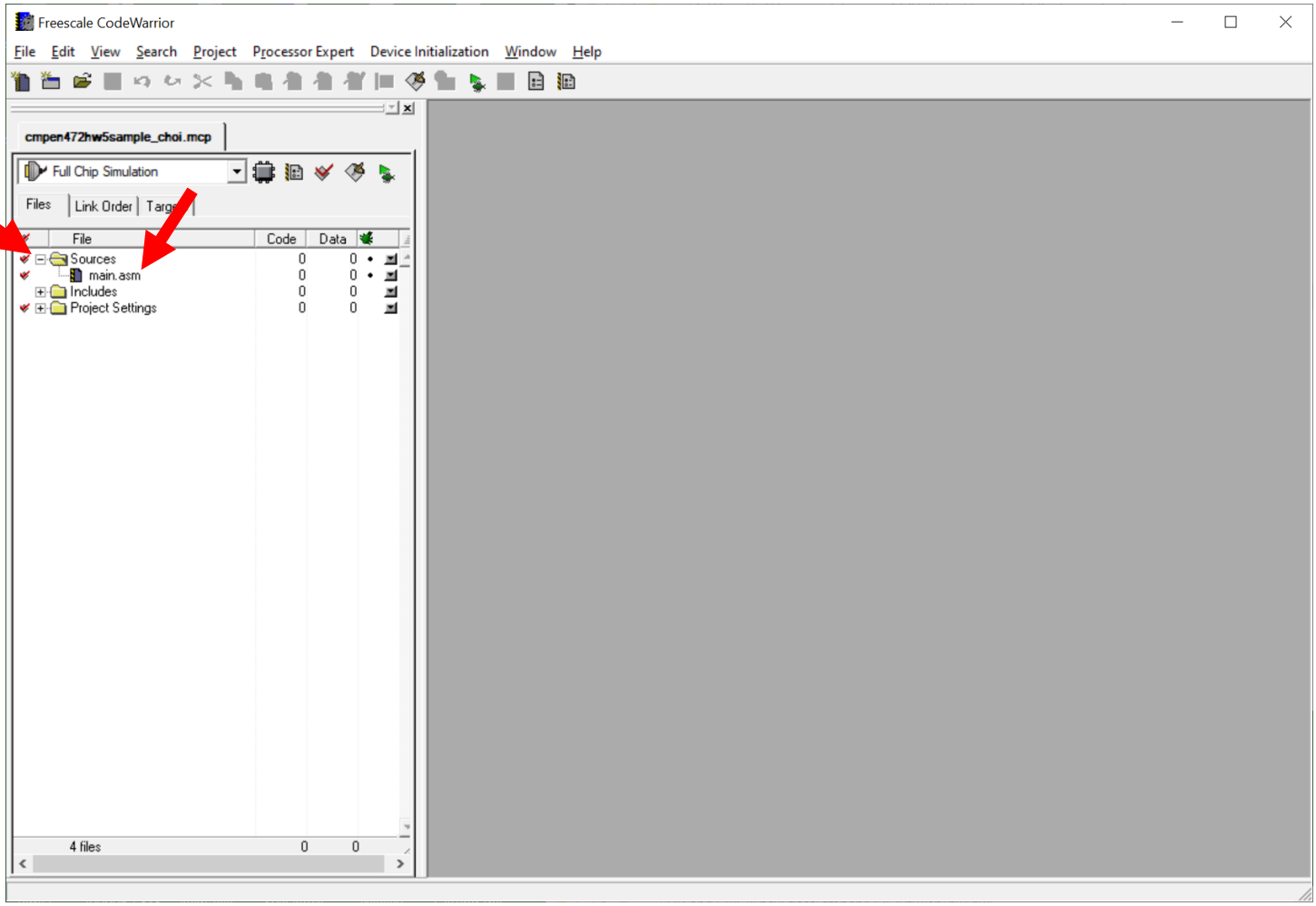



HC(S)12(X) Microcontrollers New Project
Wizard Map
Device and Connection
Project Parameters
Add Additional Files
Processor Expert
Please choose the set of languages to be supported initially. You can make multiple selections.
Absolute assembly
Relocatable assembly
C
C++
Using only one single assembly source file with absolute assembly. No support for relocatable assembly or linker.
Project name:
cmpen472hw5sample_choi.mcp
Location:
C:\Users\kc\Documents\cmpen472\F20\hw5\
Set...
< Back
Next >
Finish
Cancel

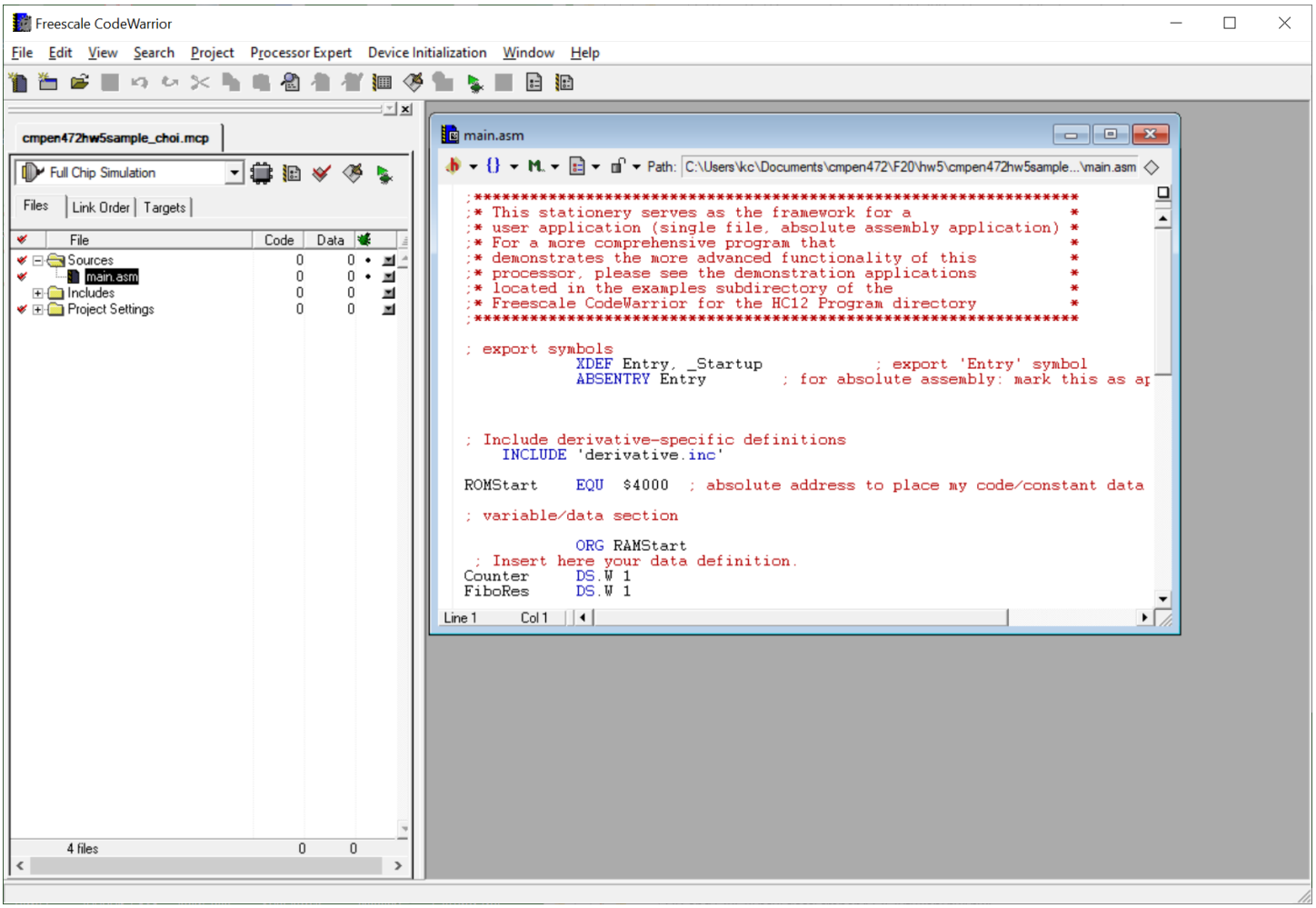

HC(S)12(X) Microcontrollers New Project
Wizard Map
Device and Connection
Project Parameters
Add Additional Files
Processor Expert
Add existing files to the project
Desktop
OneDrive
kc
This PC
Libraries
SAND128GB (E:)
Network
482Wform
CMPEN472
ImpedanceMeasureChip
Add
Remove
Project Files
Copy files to project
Create main.c/main.asm file
Select files to be added to the new project and press "Add..."
To copy the added files to the project folder, select "Copy Files to Project"
To have the wizard generate default main.c and/or main.asm files, select "Create
< Back
Next >
Finish
Cancel


HC(S)12(X) Microcontrollers New Project
Wizard Map
Device and Connection
Project Parameters
Add Additional Files
Processor Expert
Rapid Application Development Options:
None
Device Initialization
Processor Expert
No device initialization code is generated. Only generates startup code. See readme.txt in project how Processor Expert can be enabled (if not done here).
Help
< Back
Next >
Finish
Cancel

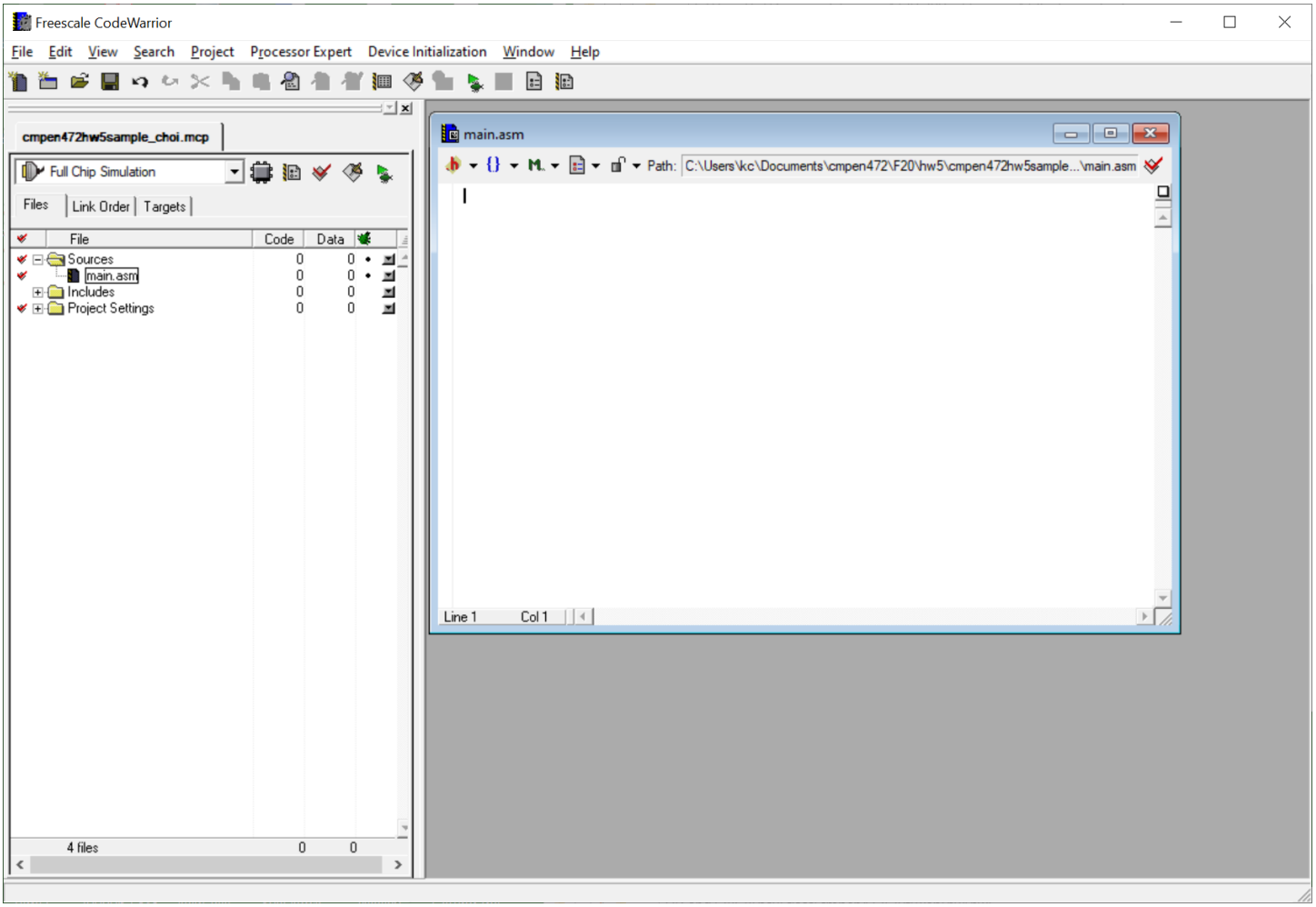
Click on Sources folder, double click on ‘main.asm’



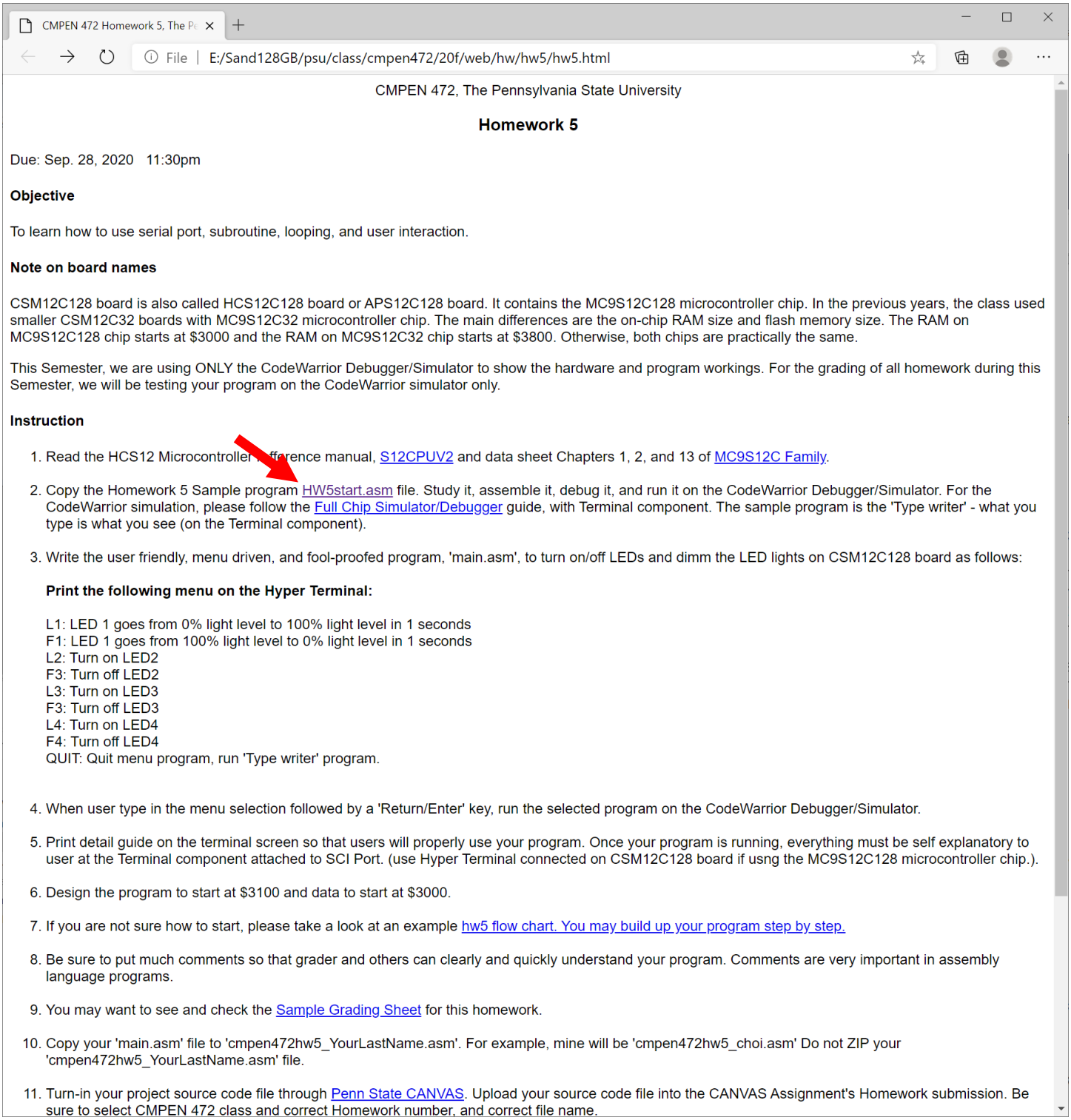
When you see the main.asm file editor, the default template program will be displayed. So DELETE all text and type (copy and paste) Homework 5 Sample program, in to the main.asm file editor.



You may use ‘control a’ and ‘DEL’ to delete all text.



You may copy the Homework 5 Sample program at the CMPEN 472 course Homework 5 page.



CMPEN 472 Homework 5, The P

File | E:/Sand128GB/psu/class/cmpen472/20f/web/hw/hw5/hw5.html

CMPEN 472, The Pennsylvania State University

Homework 5

Due: Sep. 28, 2020 11:30pm

Objective

To learn how to use serial port, subroutine, looping, and user interaction.

Note on board names

CSM12C128 board is also called HCS12C128 board or APS12C128 board. It contains the MC9S12C128 microcontroller chip. In the previous years, the class used smaller CSM12C32 boards with MC9S12C32 microcontroller chip. The main differences are the on-chip RAM size and flash memory size. The RAM on MC9S12C128 chip starts at $3000 and the RAM on MC9S12C32 chip starts at $3800. Otherwise, both chips are practically the same.

This Semester, we are using ONLY the CodeWarrior Debugger/Simulator to show the hardware and program workings. For the grading of all homework during this Semester, we will be testing your program on the CodeWarrior simulator only.

Instruction

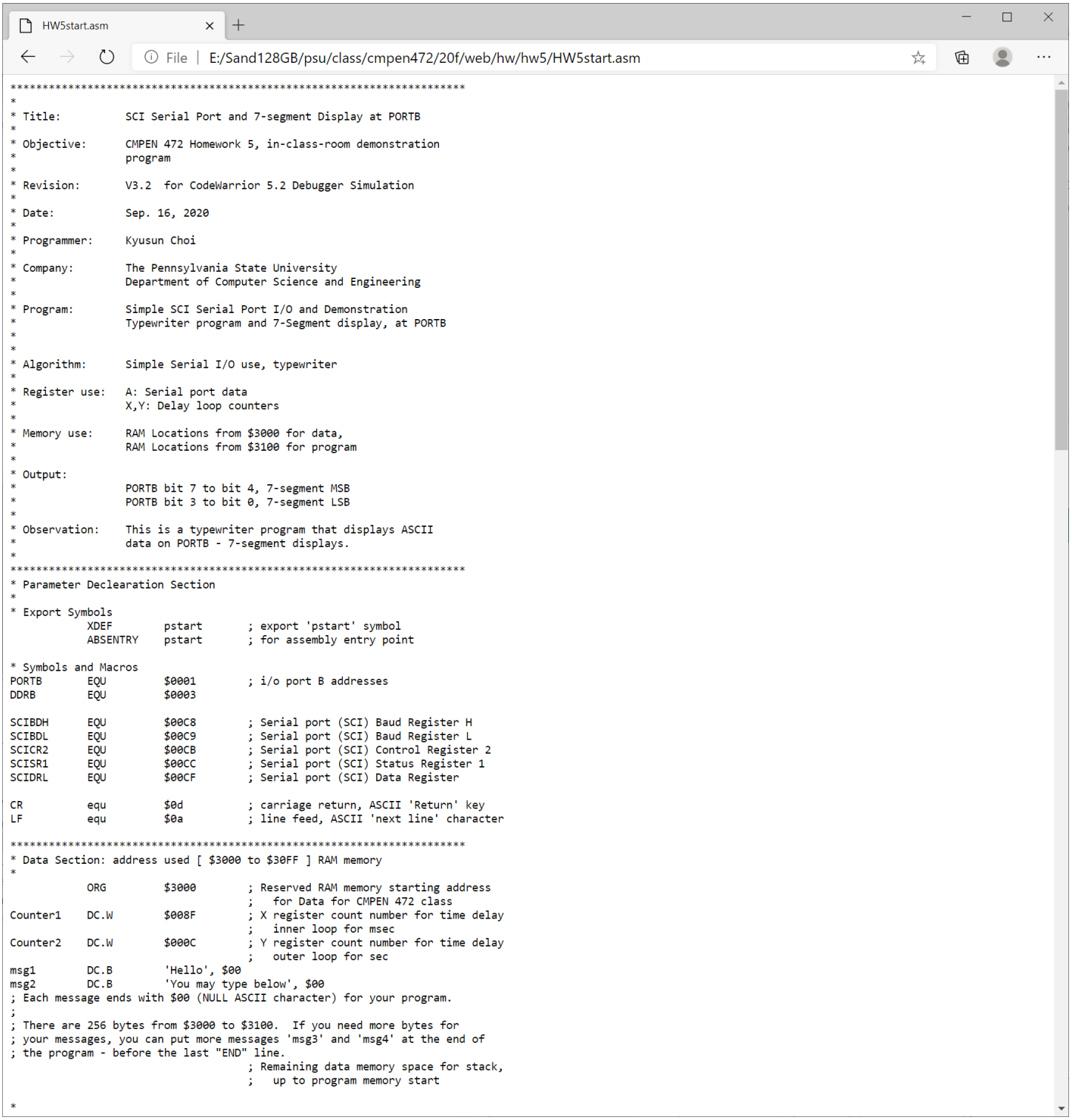
1. Read the HCS12 Microcontroller Reference manual, S12CPUV2 and data sheet Chapters 1, 2, and 13 of MC9S12C Family.
2. Copy the Homework 5 Sample program HW5start.asm file. Study it, assemble it, debug it, and run it on the CodeWarrior Debugger/Simulator. For the CodeWarrior simulation, please follow the Full Chip Simulator/Debugger guide, with Terminal component. The sample program is the 'Type writer' - what you type is what you see (on the Terminal component).
3. Write the user friendly, menu driven, and fool-proofed program, 'main.asm', to turn on/off LEDs and dimm the LED lights on CSM12C128 board as follows:

   Print the following menu on the Hyper Terminal:

   L1: LED 1 goes from 0% light level to 100% light level in 1 seconds
   F1: LED 1 goes from 100% light level to 0% light level in 1 seconds
   L2: Turn on LED2
   F3: Turn off LED2
   L3: Turn on LED3
   F3: Turn off LED3
   L4: Turn on LED4
   F4: Turn off LED4
   QUIT: Quit menu program, run 'Type writer' program.

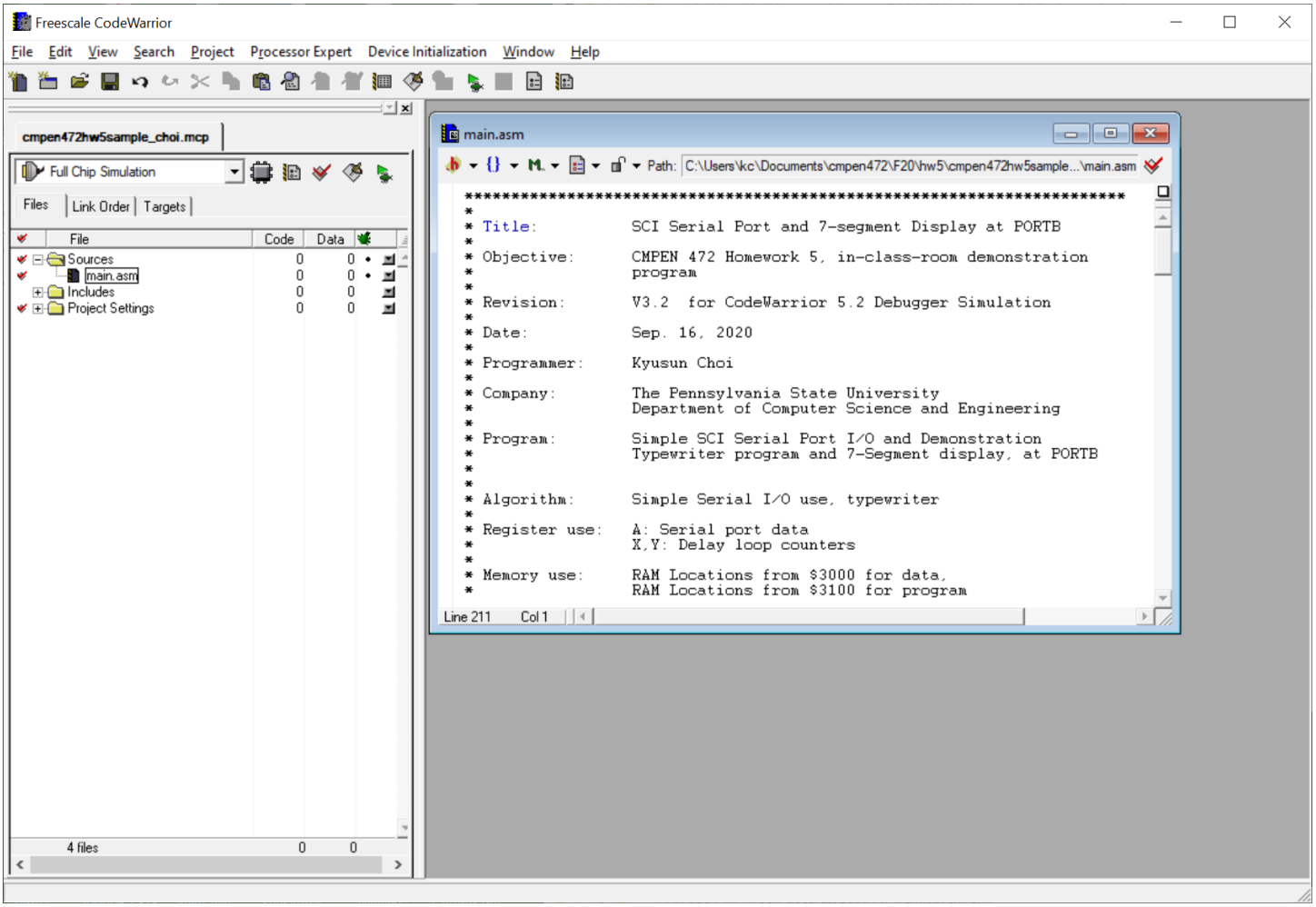
4. When user type in the menu selection followed by a 'Return/Enter' key, run the selected program on the CodeWarrior Debugger/Simulator.
5. Print detail guide on the terminal screen so that users will properly use your program. Once your program is running, everything must be self explanatory to user at the Terminal component attached to SCI Port. (use Hyper Terminal connected on CSM12C128 board if usng the MC9S12C128 microcontroller chip.).
6. Design the program to start at $3100 and data to start at $3000.
7. If you are not sure how to start, please take a look at an example hw5 flow chart. You may build up your program step by step.
8. Be sure to put much comments so that grader and others can clearly and quickly understand your program. Comments are very important in assembly language programs.
9. You may want to see and check the Sample Grading Sheet for this homework.
10. Copy your 'main.asm' file to 'cmpen472hw5_YourLastName.asm'. For example, mine will be 'cmpen472hw5_choi.asm' Do not ZIP your 'cmpen472hw5_YourLastName.asm' file.
11. Turn-in your project source code file through Penn State CANVAS. Upload your source code file into the CANVAS Assignment's Homework submission. Be sure to select CMPEN 472 class and correct Homework number, and correct file name.

You may use ‘control a’ and ‘control c’ to copy all text.

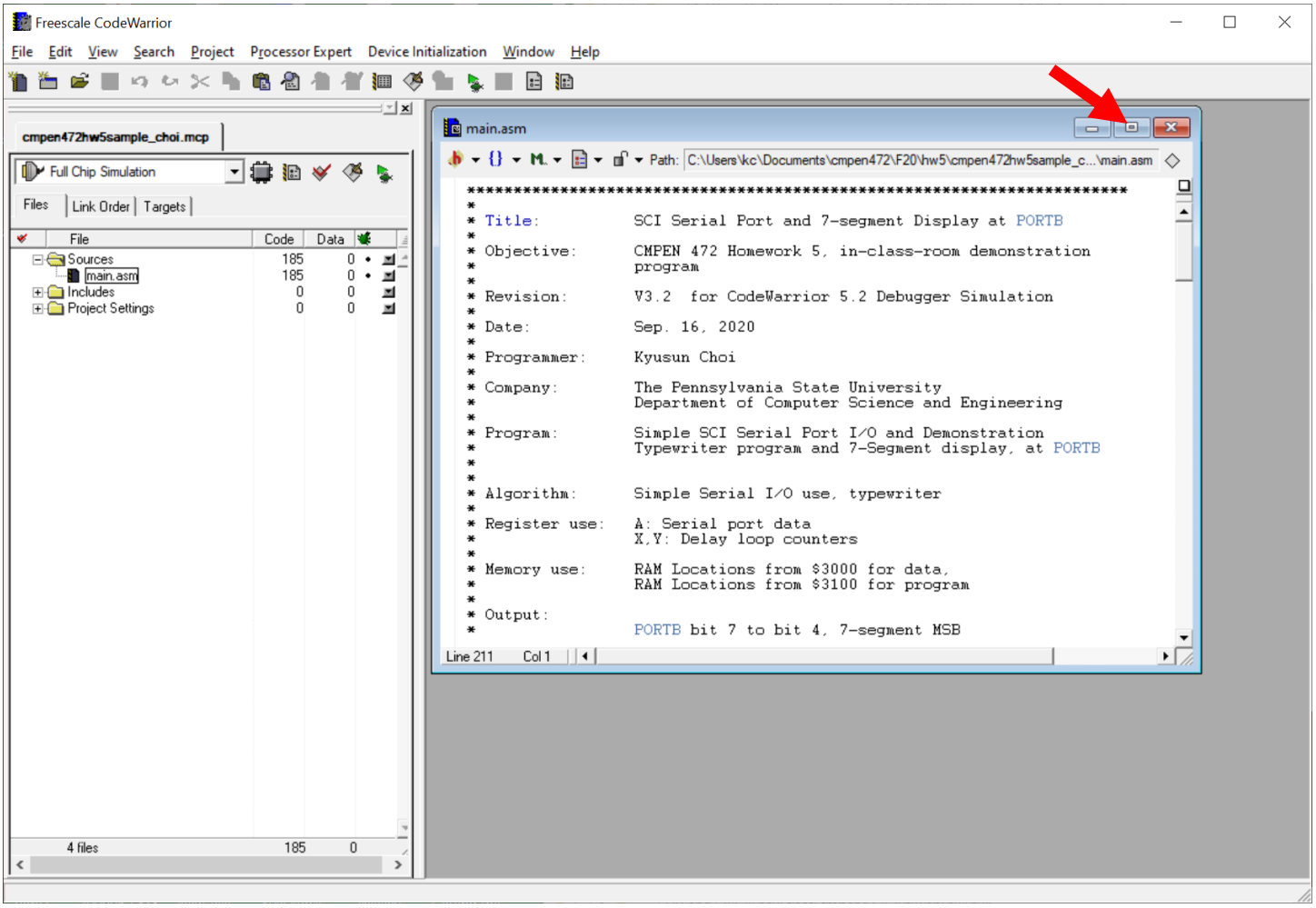


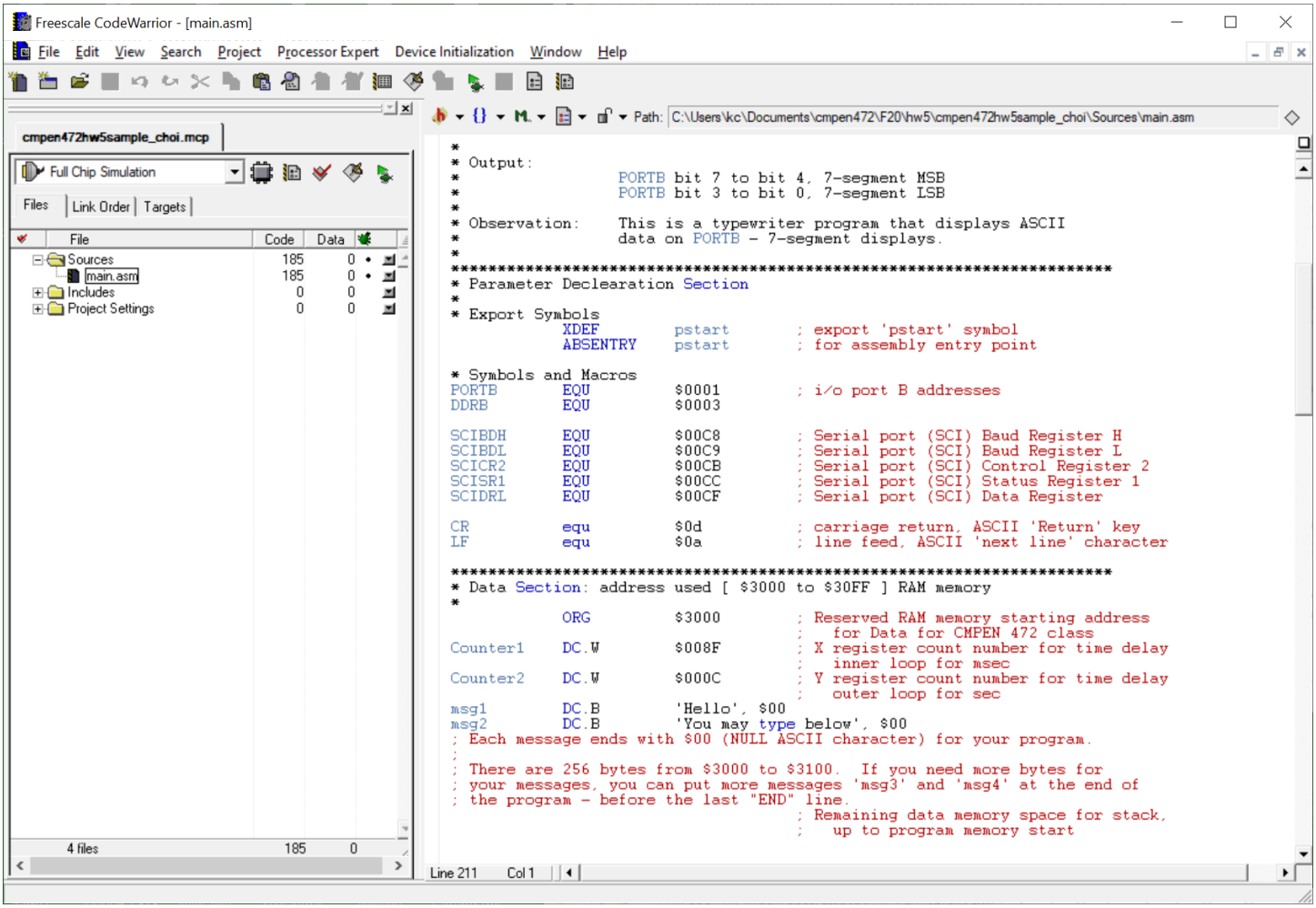

```
*****************************************************************
*
* Title:          SCI Serial Port and 7-segment Display at PORTB
*
* Objective:      CMPEN 472 Homework 5, in-class-room demonstration
*                 program
*
* Revision:       V3.2  for CodeWarrior 5.2 Debugger Simulation
*
* Date:           Sep. 16, 2020
*
* Programmer:     Kyusun Choi
*
* Company:        The Pennsylvania State University
*                 Department of Computer Science and Engineering
*
* Program:        Simple SCI Serial Port I/O and Demonstration
*                 Typewriter program and 7-Segment display, at PORTB
*
*
* Algorithm:      Simple Serial I/O use, typewriter
*
* Register use:   A: Serial port data
*                 X,Y: Delay loop counters
*
* Memory use:     RAM Locations from $3000 for data,
*                 RAM Locations from $3100 for program
*
* Output:
*                 PORTB bit 7 to bit 4, 7-segment MSB
*                 PORTB bit 3 to bit 0, 7-segment LSB
*
* Observation:    This is a typewriter program that displays ASCII
*                 data on PORTB - 7-segment displays.
*
*****************************************************************
* Parameter Declearation Section
*
* Export Symbols
            XDEF        pstart       ; export 'pstart' symbol
            ABSENTRY    pstart       ; for assembly entry point

* Symbols and Macros
PORTB       EQU         $0001        ; i/o port B addresses
DDRB        EQU         $0003

SCIBDH      EQU         $00C8        ; Serial port (SCI) Baud Register H
SCIBDL      EQU         $00C9        ; Serial port (SCI) Baud Register L
SCICR2      EQU         $00CB        ; Serial port (SCI) Control Register 2
SCISR1      EQU         $00CC        ; Serial port (SCI) Status Register 1
SCIDRL      EQU         $00CF        ; Serial port (SCI) Data Register

CR          equ         $0d          ; carriage return, ASCII 'Return' key
LF          equ         $0a          ; line feed, ASCII 'next line' character

*****************************************************************
* Data Section: address used [ $3000 to $30FF ] RAM memory
*
            ORG         $3000        ; Reserved RAM memory starting address
                                     ;   for Data for CMPEN 472 class
Counter1    DC.W        $008F        ; X register count number for time delay
                                     ;   inner loop for msec
Counter2    DC.W        $000C        ; Y register count number for time delay
                                     ;   outer loop for sec
msg1        DC.B        'Hello', $00
msg2        DC.B        'You may type below', $00
; Each message ends with $00 (NULL ASCII character) for your program.
;
; There are 256 bytes from $3000 to $3100.  If you need more bytes for
; your messages, you can put more messages 'msg3' and 'msg4' at the end of
; the program - before the last "END" line.
                                     ; Remaining data memory space for stack,
                                     ;   up to program memory start

*
```

You may use ‘control v’ to paste all text in to the main.asm window.

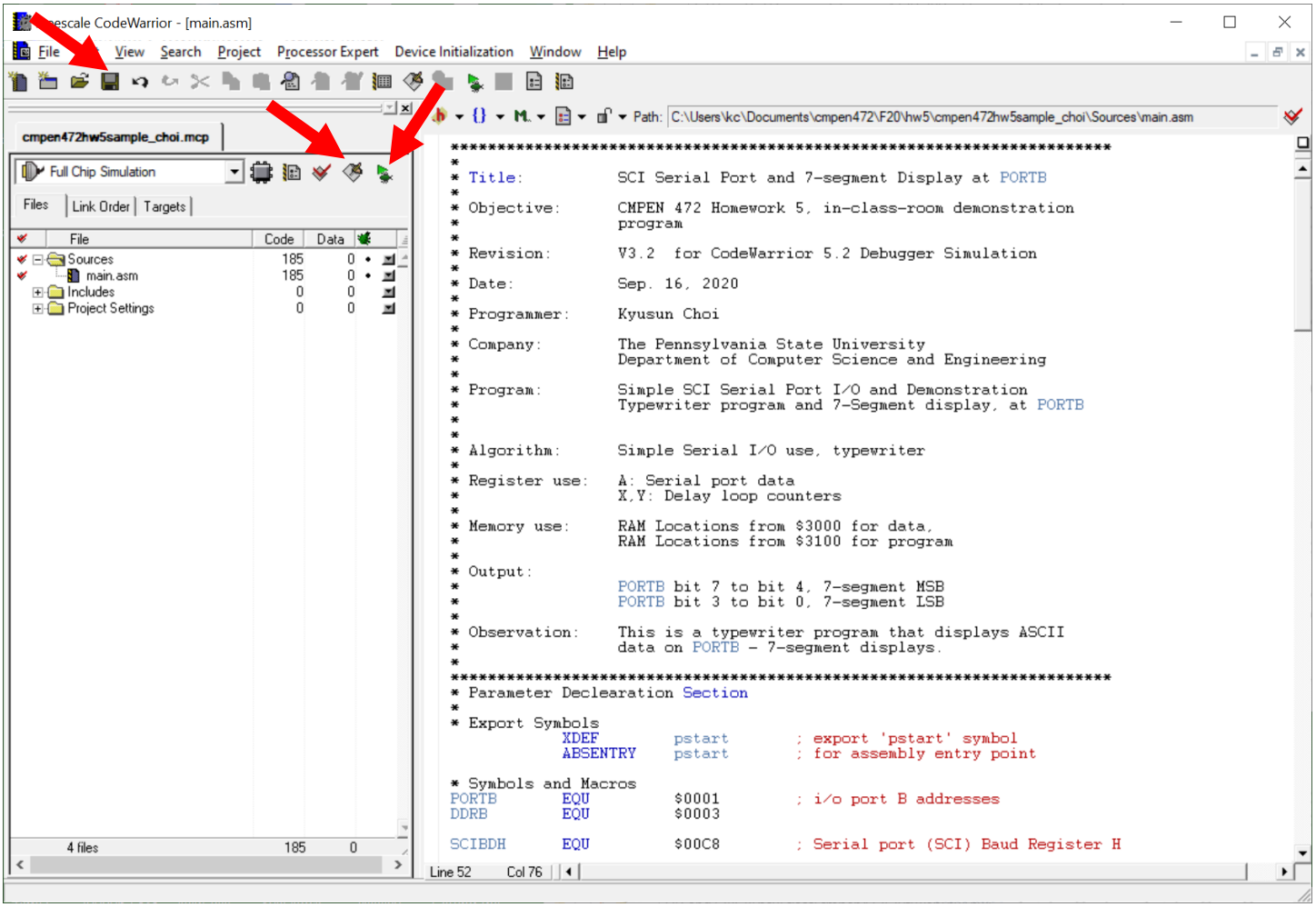


The sample program was designed to work on the CodeWarrior Debugger/Simulator. If you want to run this sample program on the actual CSM-12C128 board, please modify the Homework 5 Sample program – the serial port (SCI PORT) baud rate number. Check the comments in the program and use the baud rate number suggested for the program modifications. One must set 9600 baud rate for the actual CSM-12C128 board running with 24MHz Bus Clock. On the other hand, the SCI PORT baud rate for simulation on the CodeWarrior Debugger/Simulator is 1.0 Mega baud. (The 9600 baud will be too slow (X100) for the simulation!)

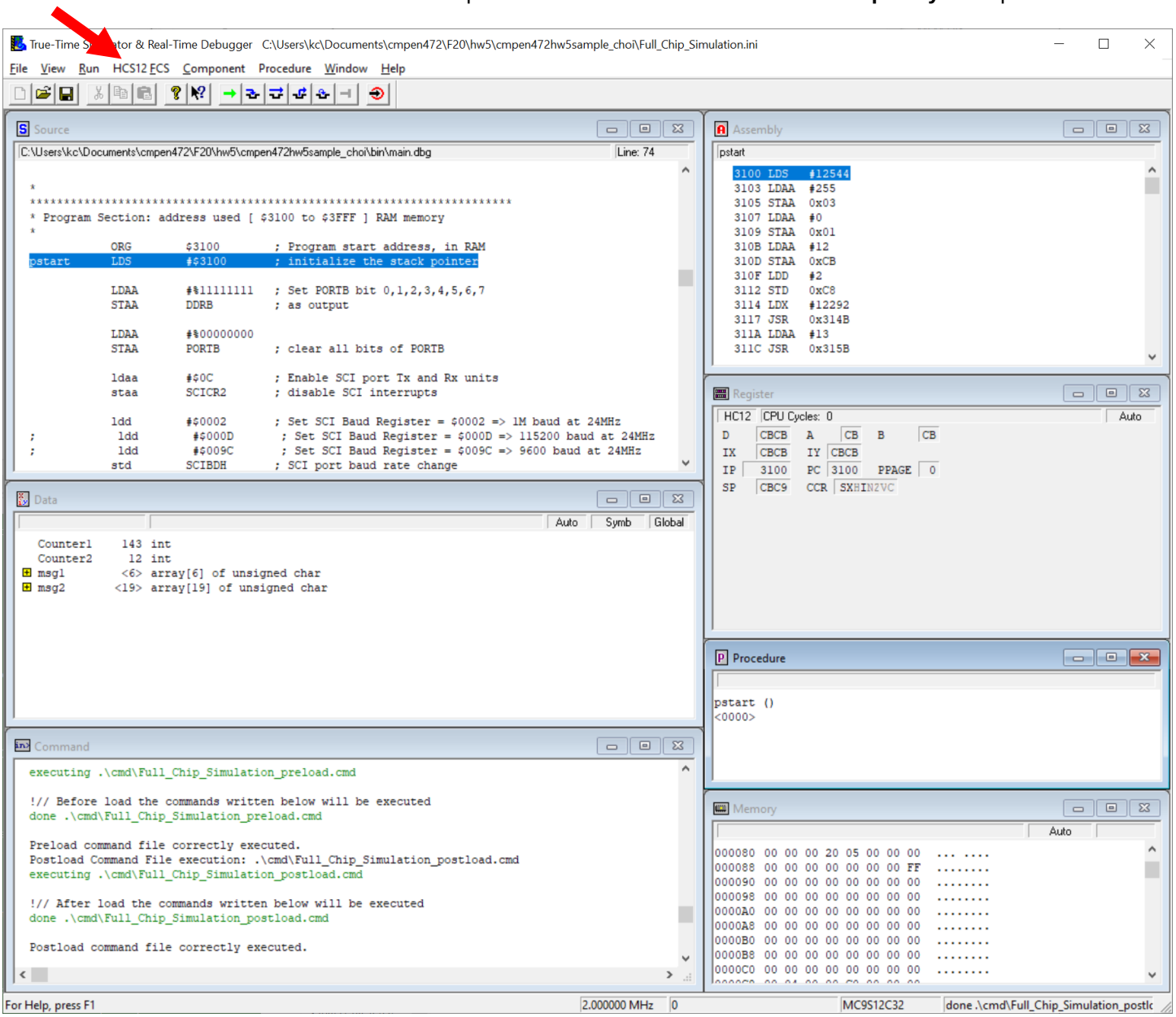


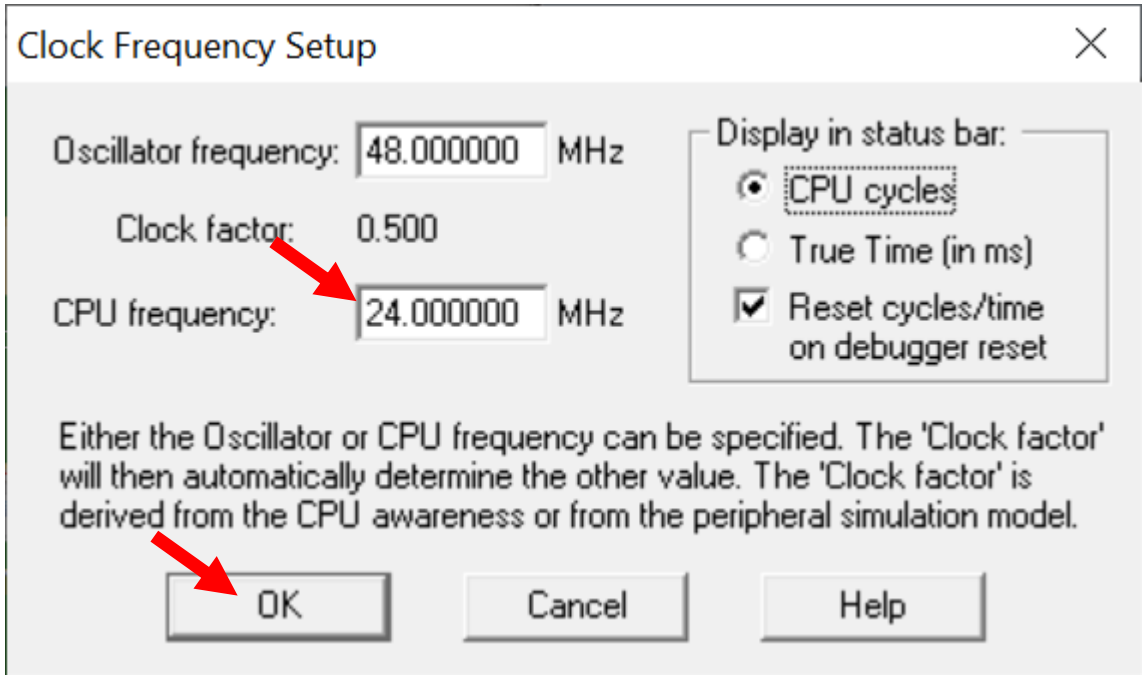

Freescale CodeWarrior - [main.asm]
File Edit View Search Project Processor Expert Device Initialization Window Help
cmpen472hw5sample_choi.mcp
Full Chip Simulation
Files Link Order Targets
File Code Data
Sources 185 0
main.asm 185 0
Includes 0 0
Project Settings 0 0
4 files 185 0
Path: C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\Sources\main.asm
* Output:
* PORTB bit 7 to bit 4, 7-segment MSB
* PORTB bit 3 to bit 0, 7-segment LSB
* Observation: This is a typewriter program that displays ASCII
* data on PORTB - 7-segment displays.
* Parameter Declearation Section
* Export Symbols
XDEF pstart ; export 'pstart' symbol
ABSENTRY pstart ; for assembly entry point
* Symbols and Macros
PORTB EQU $0001 ; i/o port B addresses
DDRB EQU $0003
SCIBDH EQU $00C8 ; Serial port (SCI) Baud Register H
SCIBDL EQU $00C9 ; Serial port (SCI) Baud Register L
SCICR2 EQU $00CB ; Serial port (SCI) Control Register 2
SCISR1 EQU $00CC ; Serial port (SCI) Status Register 1
SCIDRL EQU $00CF ; Serial port (SCI) Data Register
CR equ $0d ; carriage return, ASCII 'Return' key
LF equ $0a ; line feed, ASCII 'next line' character
* Data Section: address used [$3000 to $30FF] RAM memory
ORG $3000 ; Reserved RAM memory starting address
; for Data for CMPEN 472 class
Counter1 DC.W $008F ; X register count number for time delay
; inner loop for msec
Counter2 DC.W $000C ; Y register count number for time delay
; outer loop for sec
msg1 DC.B 'Hello', $00
msg2 DC.B 'You may type below', $00
; Each message ends with $00 (NULL ASCII character) for your program.
;
; There are 256 bytes from $3000 to $3100. If you need more bytes for
; your messages, you can put more messages 'msg3' and 'msg4' at the end of
; the program - before the last "END" line.
; Remaining data memory space for stack,
; up to program memory start
Line 211 Col 1

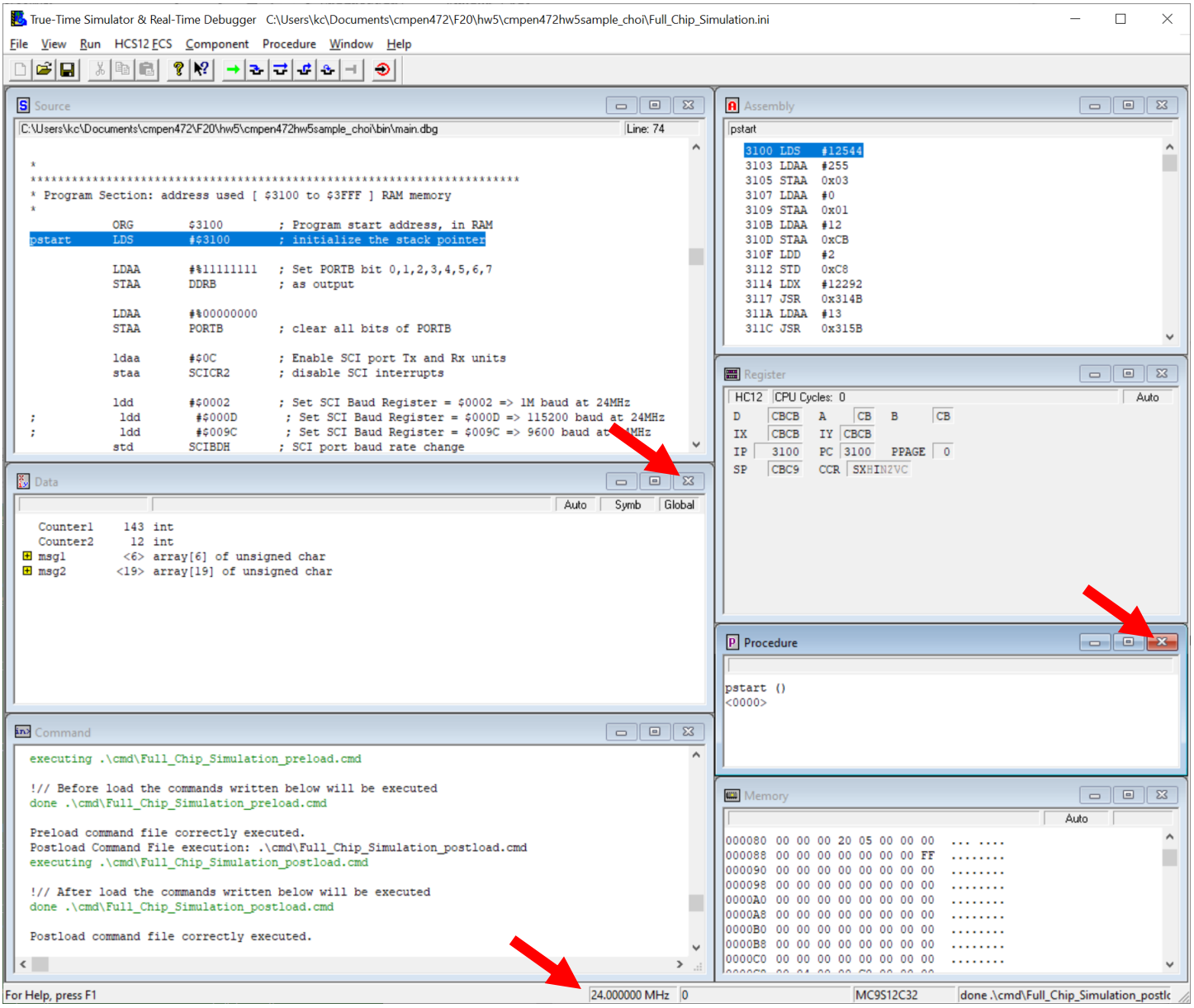
Now save the program, do ‘make’, and ‘Debug’. Fix errors if you have any.



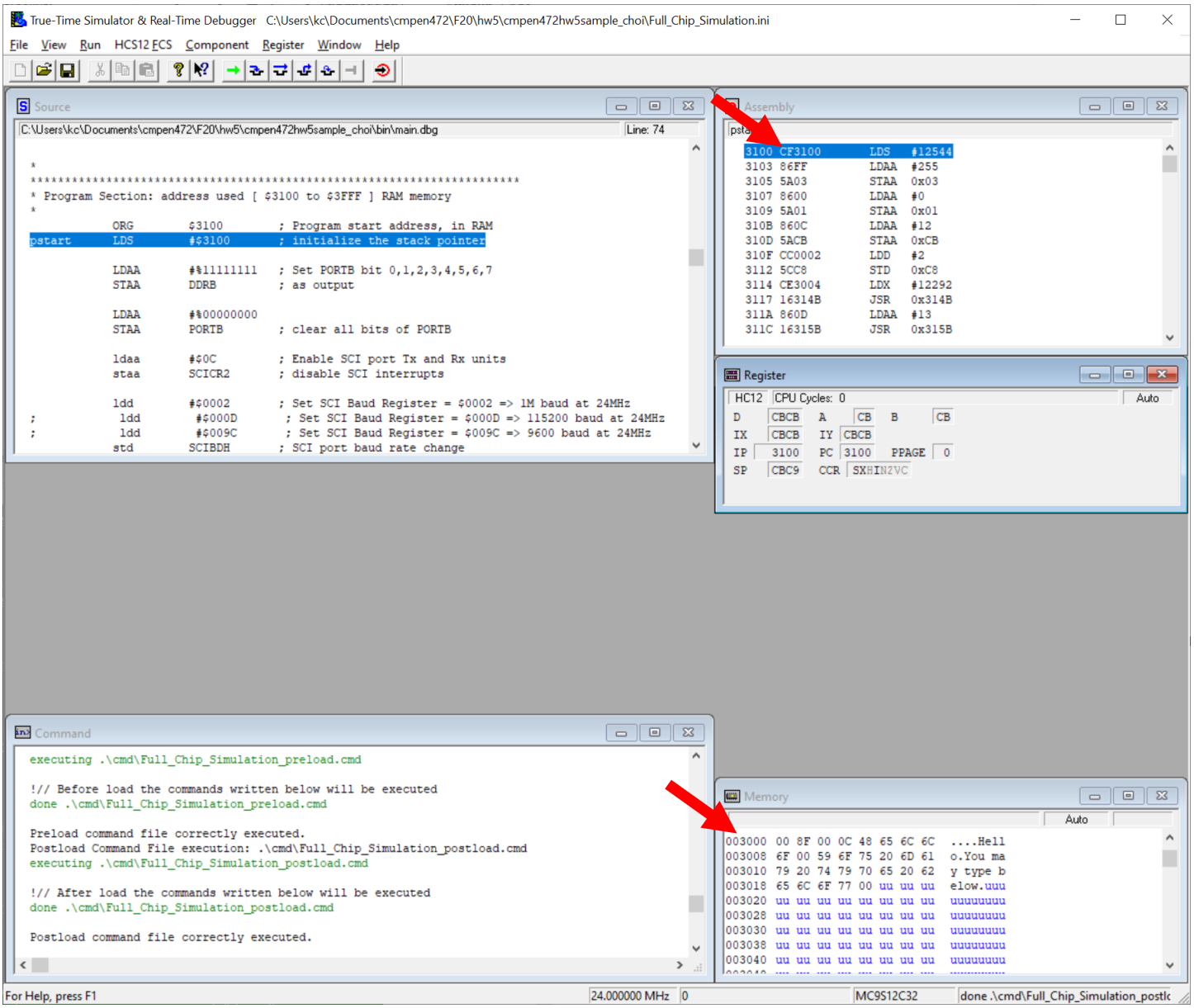
Once the Debugger/Simulator is launched, you must set it up to see the I/O activity when your program runs. Resize the Simulator window as shown. Click HCS12 FCS pull down menu and select the **Clock Frequency . . .** option.

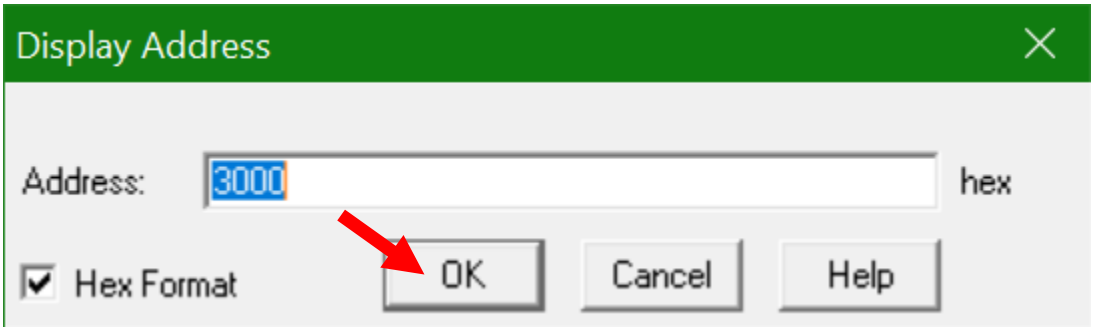



Clock Frequency Setup
Oscillator frequency:
48.000000
MHz
Clock factor:
0.500
CPU frequency:
24.000000
MHz
Display in status bar:
CPU cycles
True Time (in ms)
Reset cycles/time on debugger reset
Either the Oscillator or CPU frequency can be specified. The 'Clock factor' will then automatically determine the other value. The 'Clock factor' is derived from the CPU awareness or from the peripheral simulation model.
OK
Cancel
Help

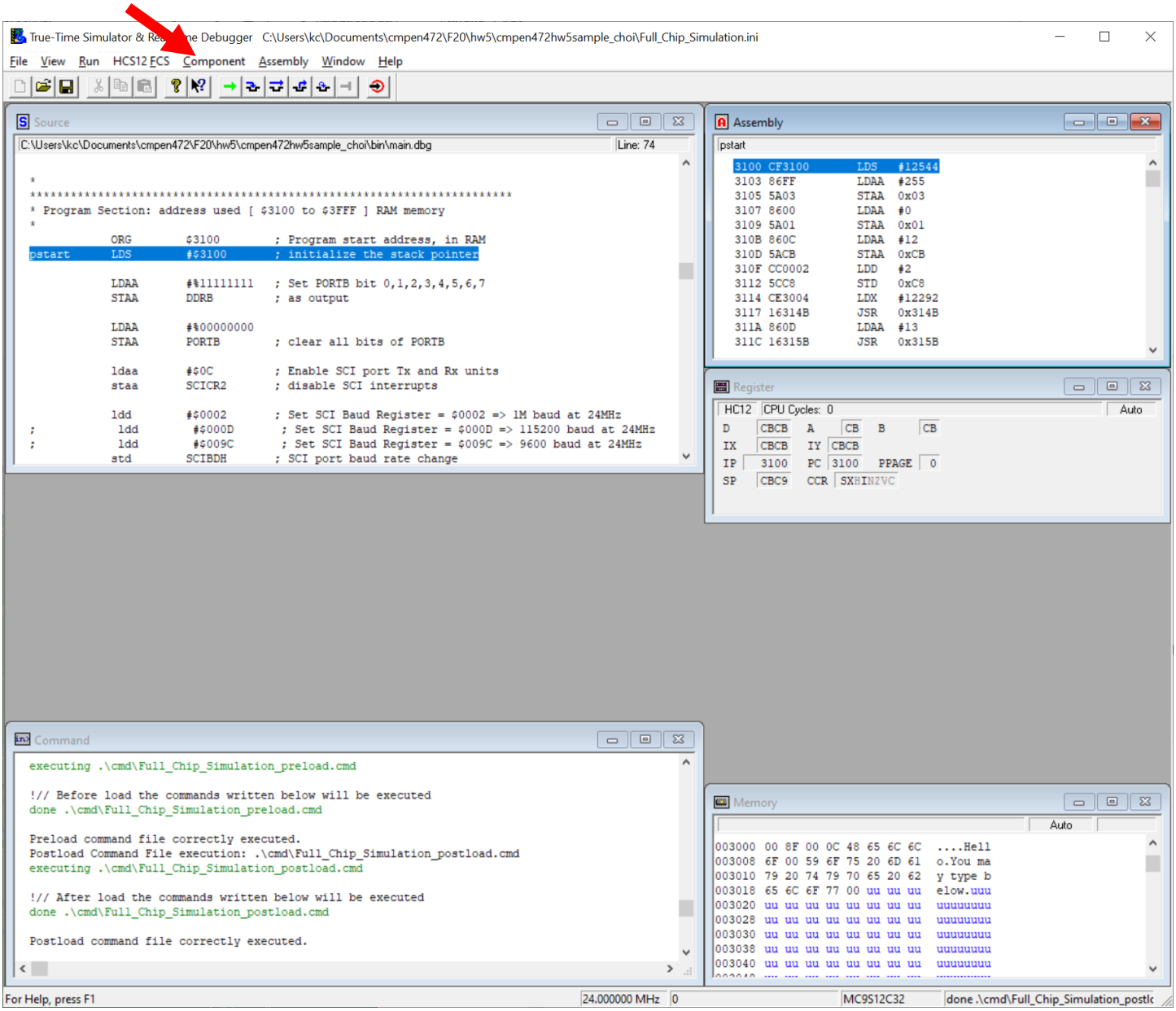

True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Procedure Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\bin\main.dbg
Line: 74
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
LDAA #%11111111 ; Set PORTB bit 0,1,2,3,4,5,6,7
STAA DDRB ; as output
LDAA #%00000000
STAA PORTB ; clear all bits of PORTB
ldaa #$0C ; Enable SCI port Tx and Rx units
staa SCICR2 ; disable SCI interrupts
ldd #$0002 ; Set SCI Baud Register = $0002 => 1M baud at 24MHz
; ldd #$000D ; Set SCI Baud Register = $000D => 115200 baud at 24MHz
; ldd #$009C ; Set SCI Baud Register = $009C => 9600 baud at 24MHz
std SCIBDH ; SCI port baud rate change
Assembly
pstart
3100 LDS #12544
3103 LDAA #255
3105 STAA 0x03
3107 LDAA #0
3109 STAA 0x01
310B LDAA #12
310D STAA 0xCB
310F LDD #2
3112 STD 0xC8
3114 LDX #12292
3117 JSR 0x314B
311A LDAA #13
311C JSR 0x315B
Register
HC12 CPU Cycles: 0
Auto
D CBCB A CB B CB
IX CBCB IY CBCB
IP 3100 PC 3100 PPAGE 0
SP CBC9 CCR SXHINZVC
Data
Auto Symb Global
Counter1 143 int
Counter2 12 int
msg1 <6> array[6] of unsigned char
msg2 <19> array[19] of unsigned char
Procedure
pstart ()
<0000>
Command
executing .\cmd\Full_Chip_Simulation_preload.cmd
!// Before load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_preload.cmd
Preload command file correctly executed.
Postload Command File execution: .\cmd\Full_Chip_Simulation_postload.cmd
executing .\cmd\Full_Chip_Simulation_postload.cmd
!// After load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_postload.cmd
Postload command file correctly executed.
Memory
Auto
000080 00 00 00 20 05 00 00 00 ... .
000088 00 00 00 00 00 00 00 FF ....
000090 00 00 00 00 00 00 00 00 ....
000098 00 00 00 00 00 00 00 00 ....
0000A0 00 00 00 00 00 00 00 00 ....
0000A8 00 00 00 00 00 00 00 00 ....
0000B0 00 00 00 00 00 00 00 00 ....
0000B8 00 00 00 00 00 00 00 00 ....
0000C0 00 00 00 00 00 00 00 00 ....
For Help, press F1
24.000000 MHz
0
MC9S12C32
done .\cmd\Full_Chip_Simulation_postlc

Right click Memory window and set the ‘Address . . . ‘ type ‘3000’
Right click Assembly window and set the ‘Display’, select ‘Code’

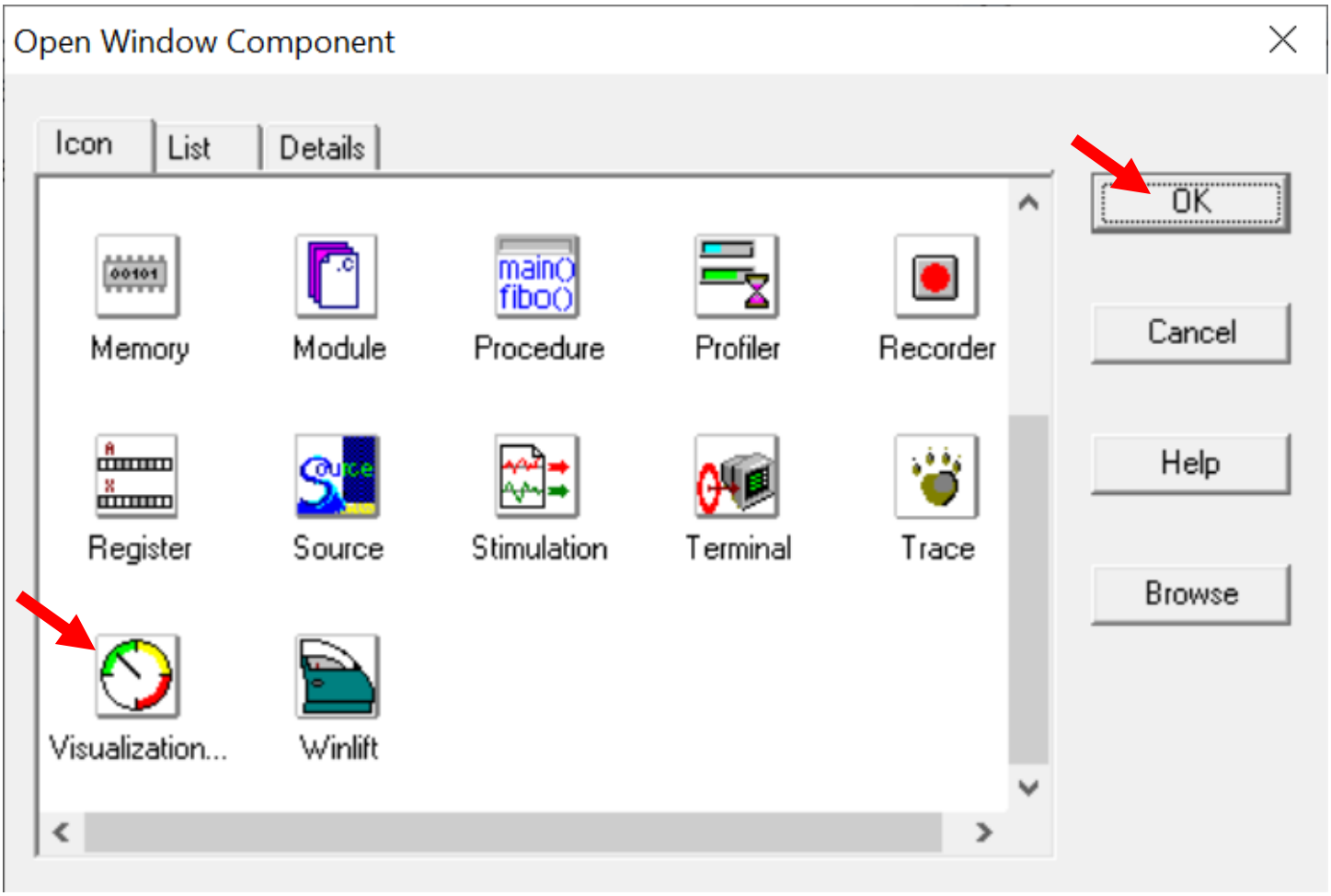




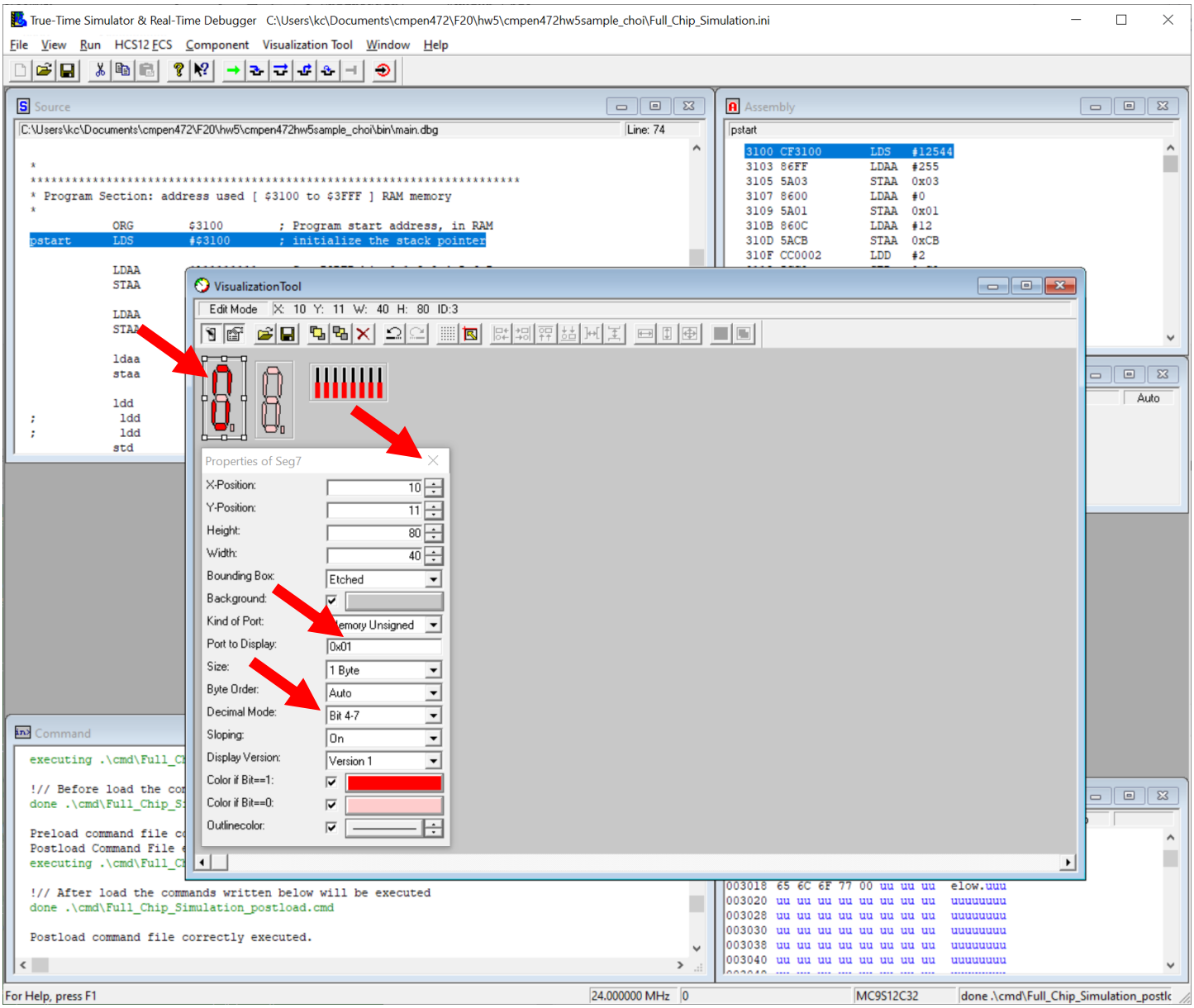
Click ‘Component’ menu and select ‘Open...’ option.

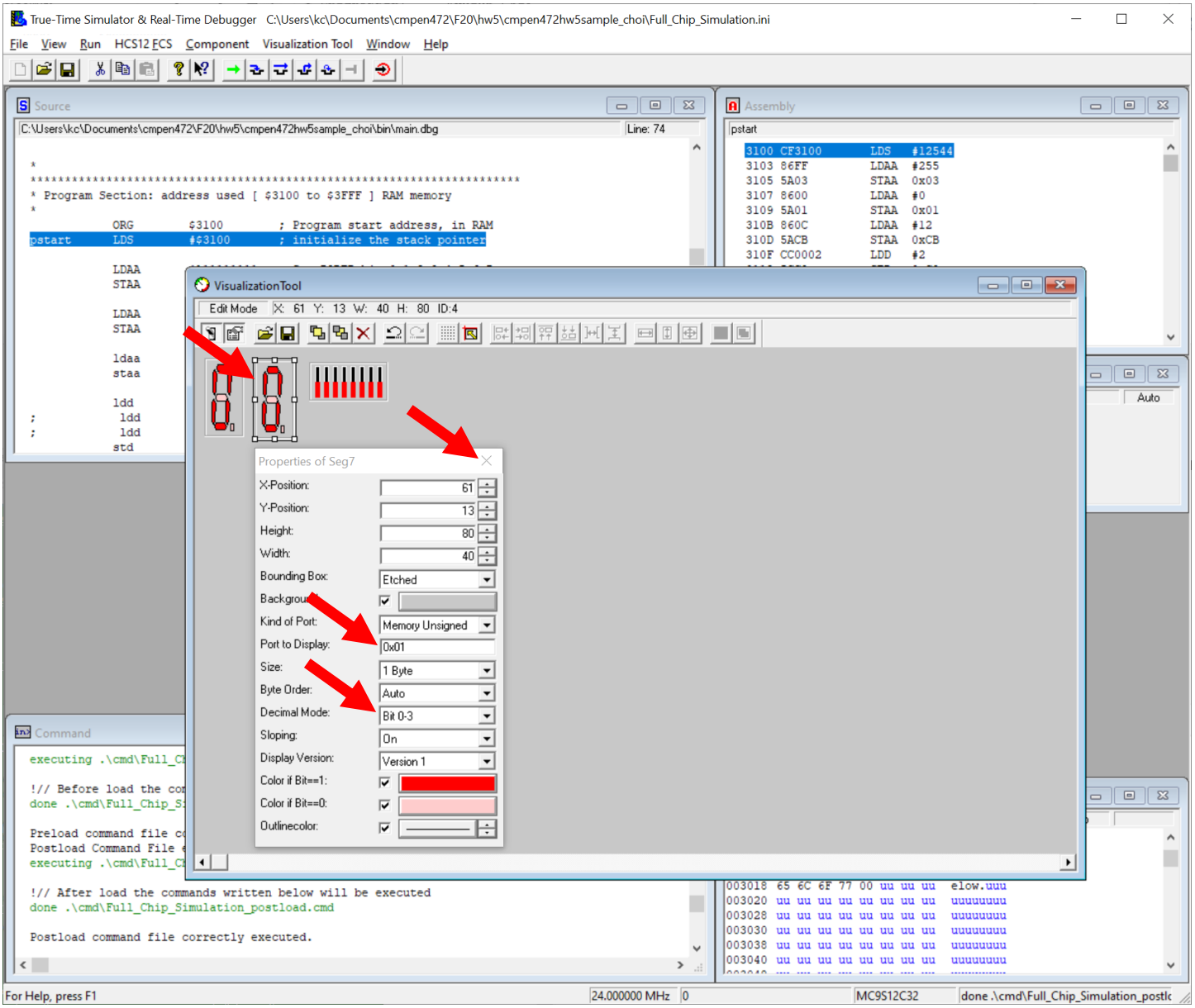


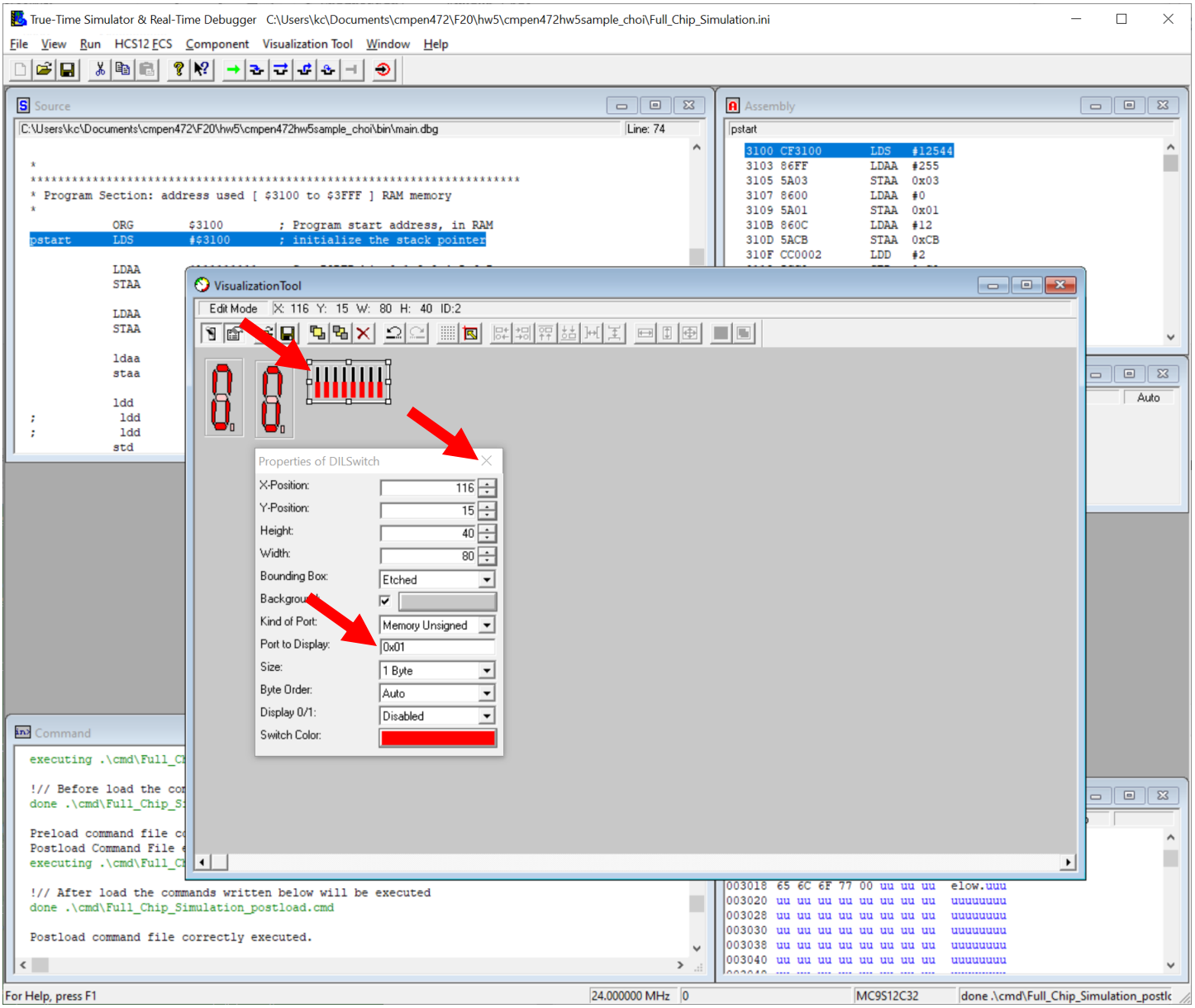
This Homework 5 Sample program uses PORTB as a display of SCI PORT data – received data. So put display components on PORTB.



In the Visualization Tool, do Add New Instrument, the 7 Segment Displays as shown. Also add the DIL Switch, which will allow you to display/read the binary number of the data received from the SCI PORT. How to Add New Instrument to the Visualization Tool are shown in the Homework 3 Sample Program guide.



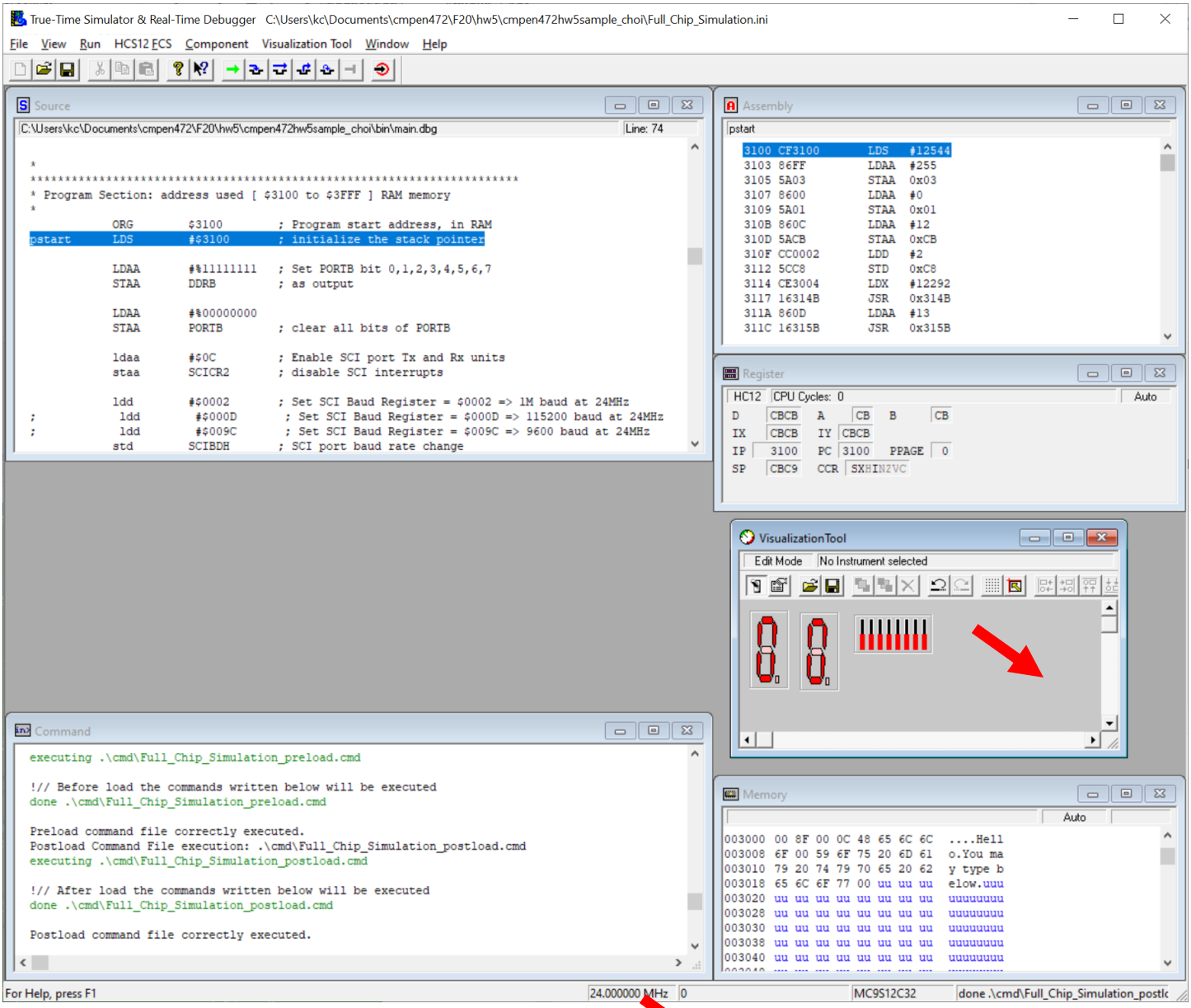

True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Visualization Tool Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\bin\main.dbg
Line: 74
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
Assembly
pstart
3100 CF3100 LDS #12544
3103 86FF LDAA #255
3105 5A03 STAA 0x03
3107 8600 LDAA #0
3109 5A01 STAA 0x01
310B 860C LDAA #12
310D 5ACB STAA 0xCB
310F CC0002 LDD #2
VisualizationTool
Edit Mode X: 61 Y: 13 W: 40 H: 80 ID:4
Properties of Seg7
X-Position: 61
Y-Position: 13
Height: 80
Width: 40
Bounding Box: Etched
Kind of Port: Memory Unsigned
Port to Display: 0x01
Size: 1 Byte
Byte Order: Auto
Decimal Mode: Bit 0-3
Sloping: On
Display Version: Version 1
Color if Bit==1:
Color if Bit==0:
Outlinecolor:
Command
executing .\cmd\Full_Ch
!// Before load the com
done .\cmd\Full_Chip_Si
Preload command file co
Postload Command File
executing .\cmd\Full_Ch
!// After load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_postload.cmd
Postload command file correctly executed.
003018 65 6C 6F 77 00 uu uu uu elow.uuu
003020 uu uu uu uu uu uu uu uu uuuuuuuu
003028 uu uu uu uu uu uu uu uu uuuuuuuu
003030 uu uu uu uu uu uu uu uu uuuuuuuu
003038 uu uu uu uu uu uu uu uu uuuuuuuu
003040 uu uu uu uu uu uu uu uu uuuuuuuu
For Help, press F1
24.000000 MHz
0
MC9S12C32
done .\cmd\Full_Chip_Simulation_postl

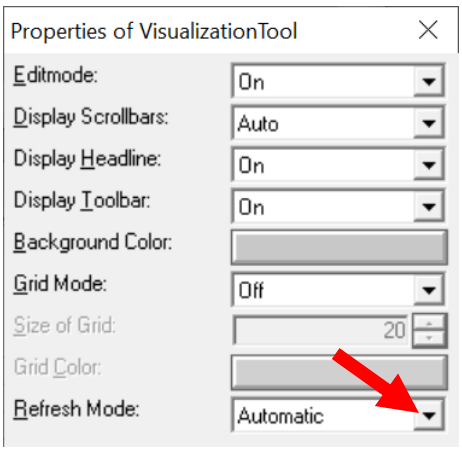

True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Visualization Tool Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\bin\main.dbg
Line: 74
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
Assembly
pstart
3100 CF3100 LDS #12544
3103 86FF LDAA #255
3105 5A03 STAA 0x03
3107 8600 LDAA #0
3109 5A01 STAA 0x01
310B 860C LDAA #12
310D 5ACB STAA 0xCB
310F CC0002 LDD #2
VisualizationTool
Edit Mode X: 116 Y: 15 W: 80 H: 40 ID:2
Properties of DILSwitch
X-Position: 116
Y-Position: 15
Height: 40
Width: 80
Bounding Box: Etched
Kind of Port: Memory Unsigned
Port to Display: 0x01
Size: 1 Byte
Byte Order: Auto
Display 0/1: Disabled
Switch Color:
Command
executing .\cmd\Full_Ch
!// Before load the co
done .\cmd\Full_Chip_S
Preload command file c
Postload Command File
executing .\cmd\Full_Ch
!// After load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_postload.cmd
Postload command file correctly executed.
003018 65 6C 6F 77 00 uu uu uu elow.uuu
003020 uu uu uu uu uu uu uu uu uuuuuuuu
003028 uu uu uu uu uu uu uu uu uuuuuuuu
003030 uu uu uu uu uu uu uu uu uuuuuuuu
003038 uu uu uu uu uu uu uu uu uuuuuuuu
003040 uu uu uu uu uu uu uu uu uuuuuuuu
For Help, press F1
24.000000 MHz
0
MC9S12C32
done .\cmd\Full_Chip_Simulation_postl

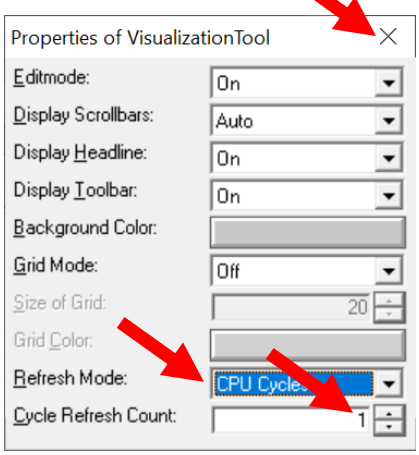
Place the Visualization Tool in the Simulator as shown below.

Important: Set the Visualization Tool's Refresh Mode.

Right click on the Visualization Tool, and then select "Properties".

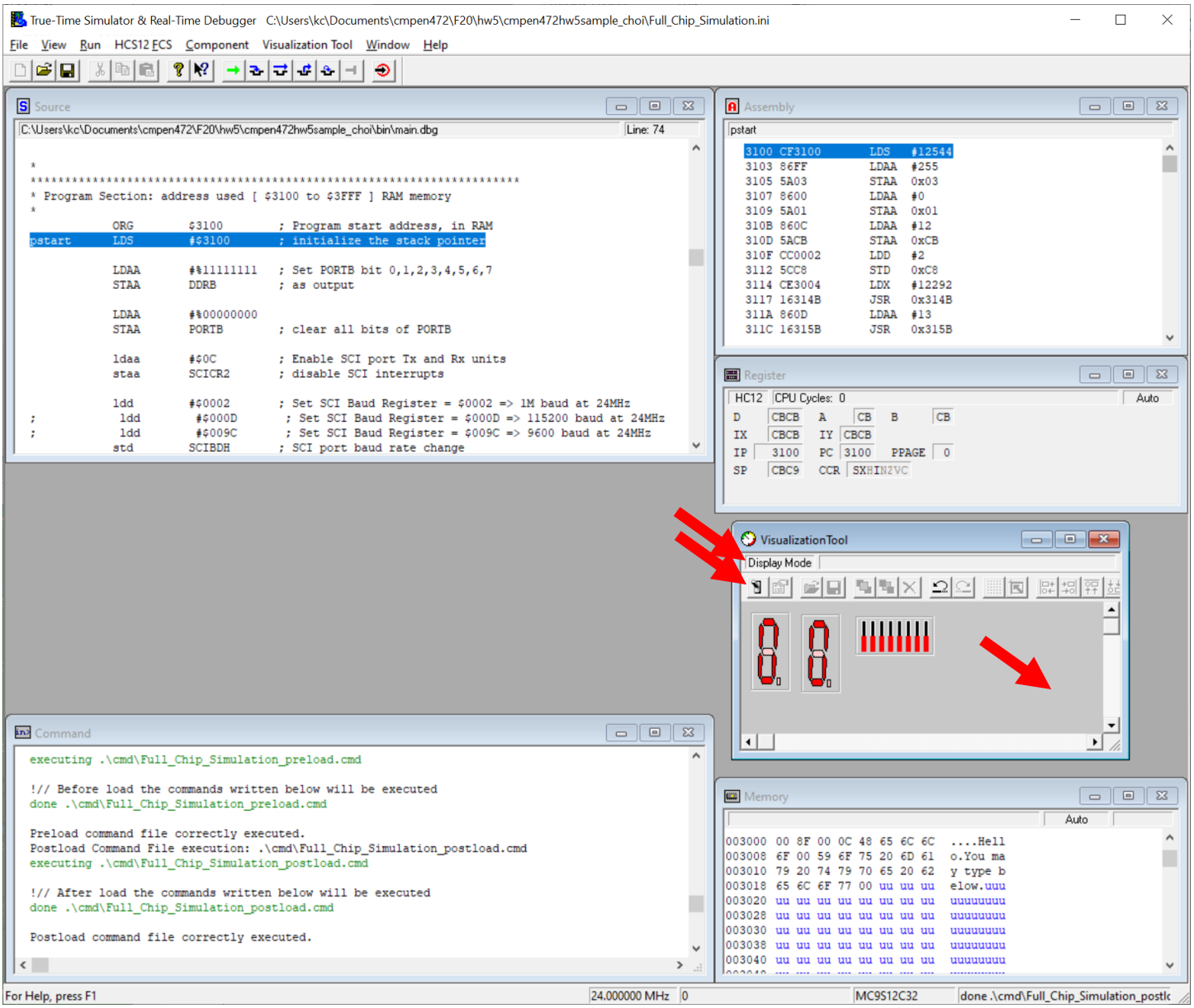






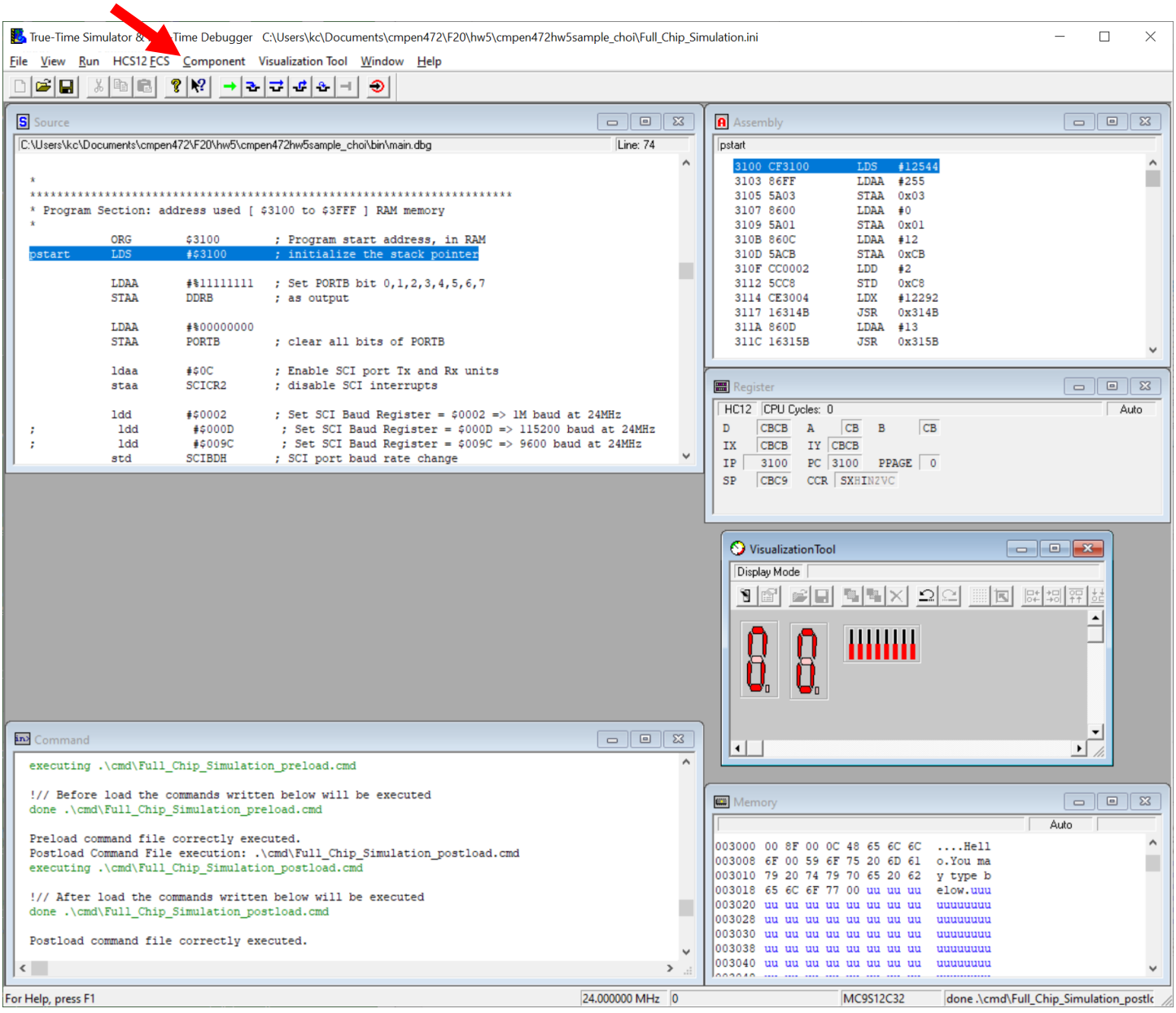
Enter "1" for Cycle Refresh Count.

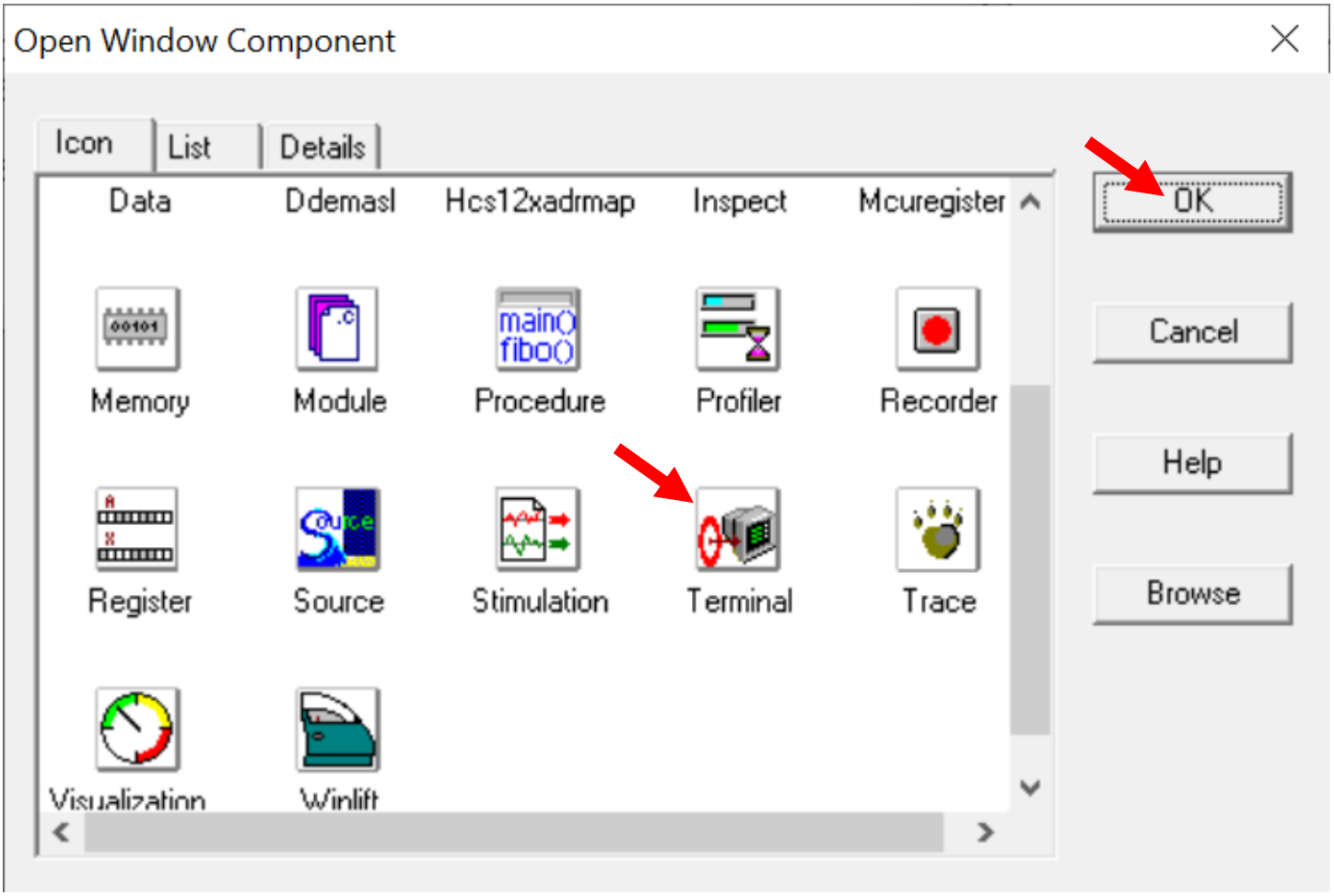
Now change your Visualization Tool from “Edit Mode” to “Display Mode”, by right clicking the Visualization Tool and select the “Edit Mode”. (The switch at PORTB will not work on “Edit Mode”, they work only on “Display Mode”.)



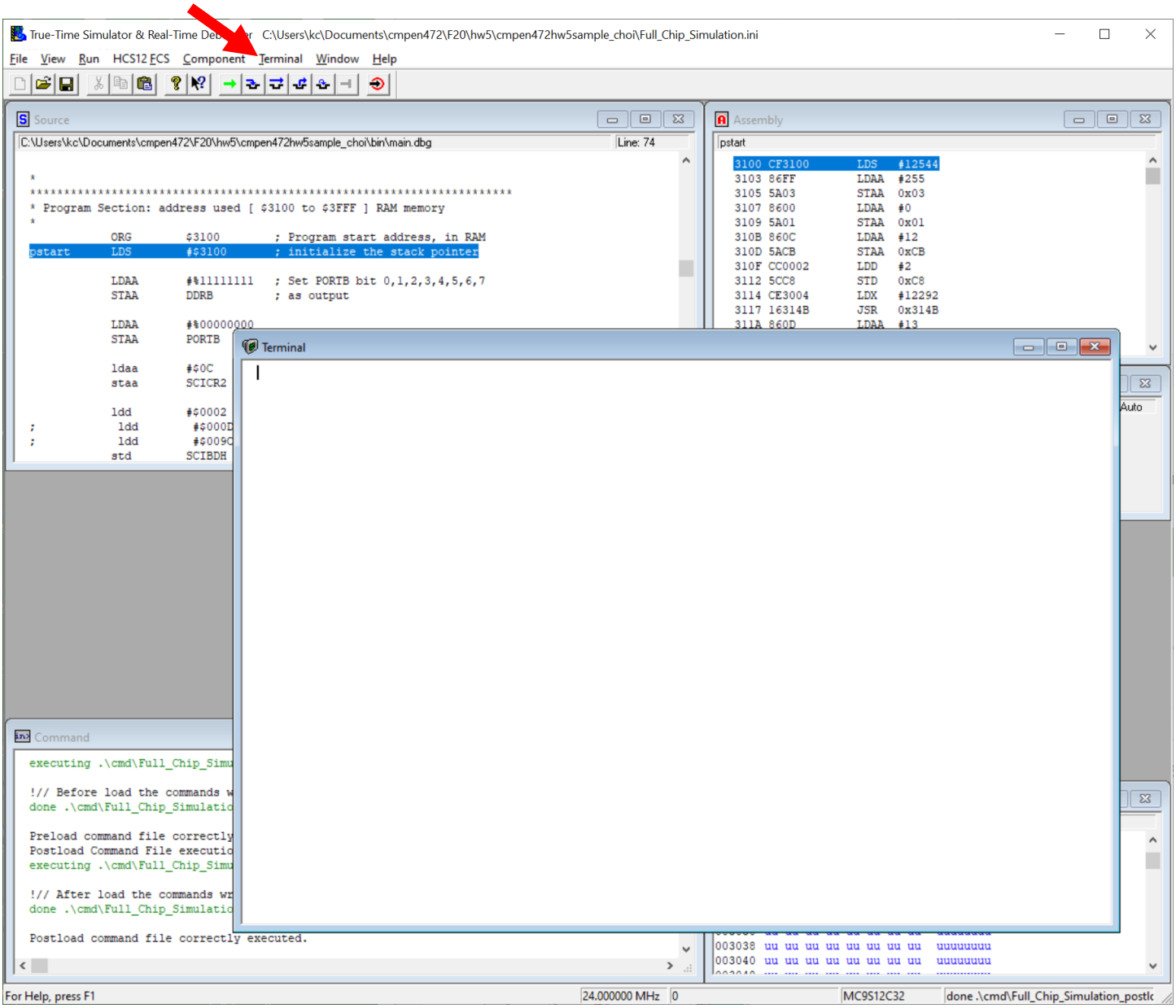
This Homework 5 Sample program uses SCI PORT for serial data communication. Put a terminal window in the Simulator window and attached it to the SCI PORT of the HCS12 processor, to get the data from and send the data to. (We already attached the LEDs and switches to the PORTB to see the data receiving.)

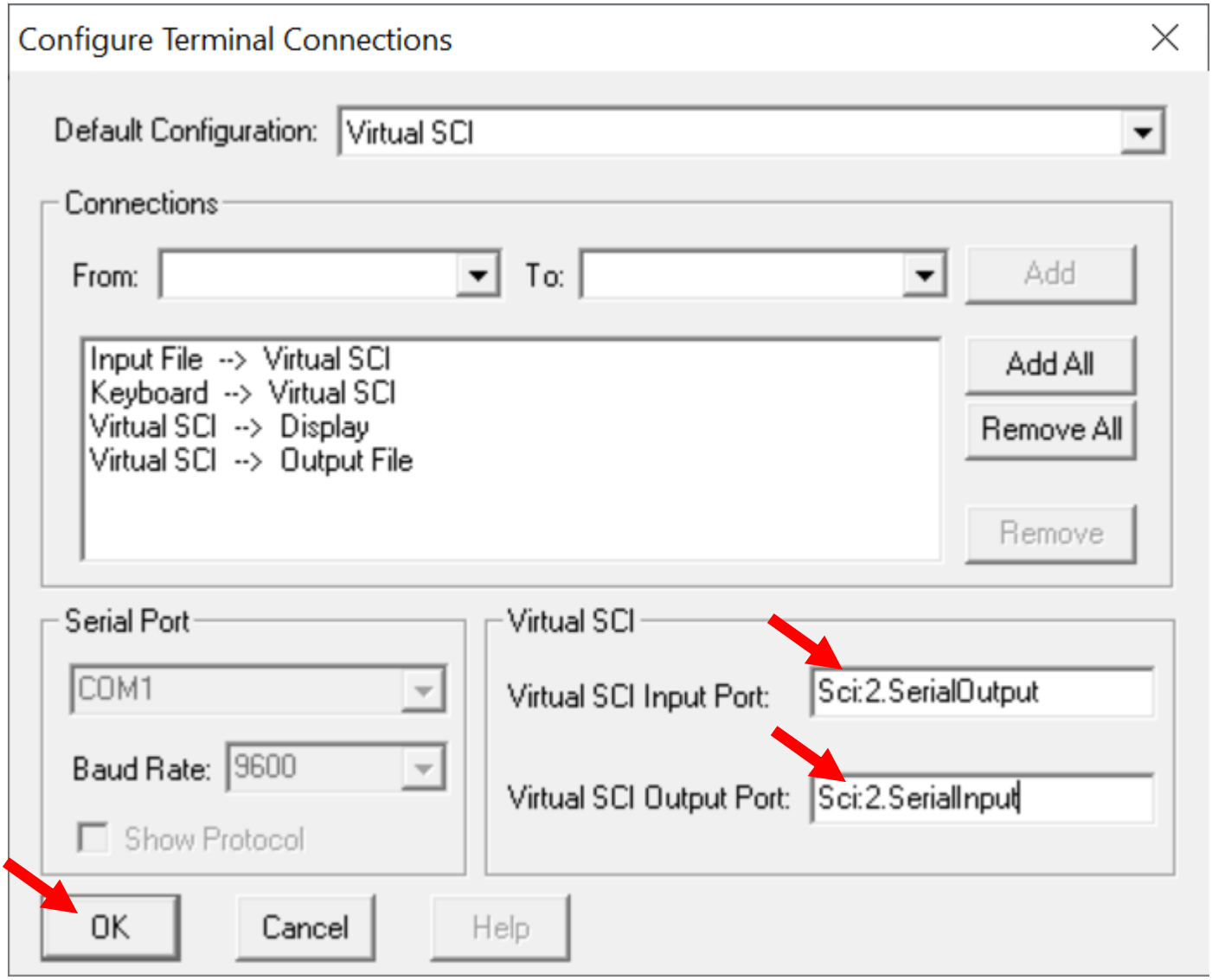
Click ‘Component’ menu and select ‘Open...’ option. Select the component “Terminal”.



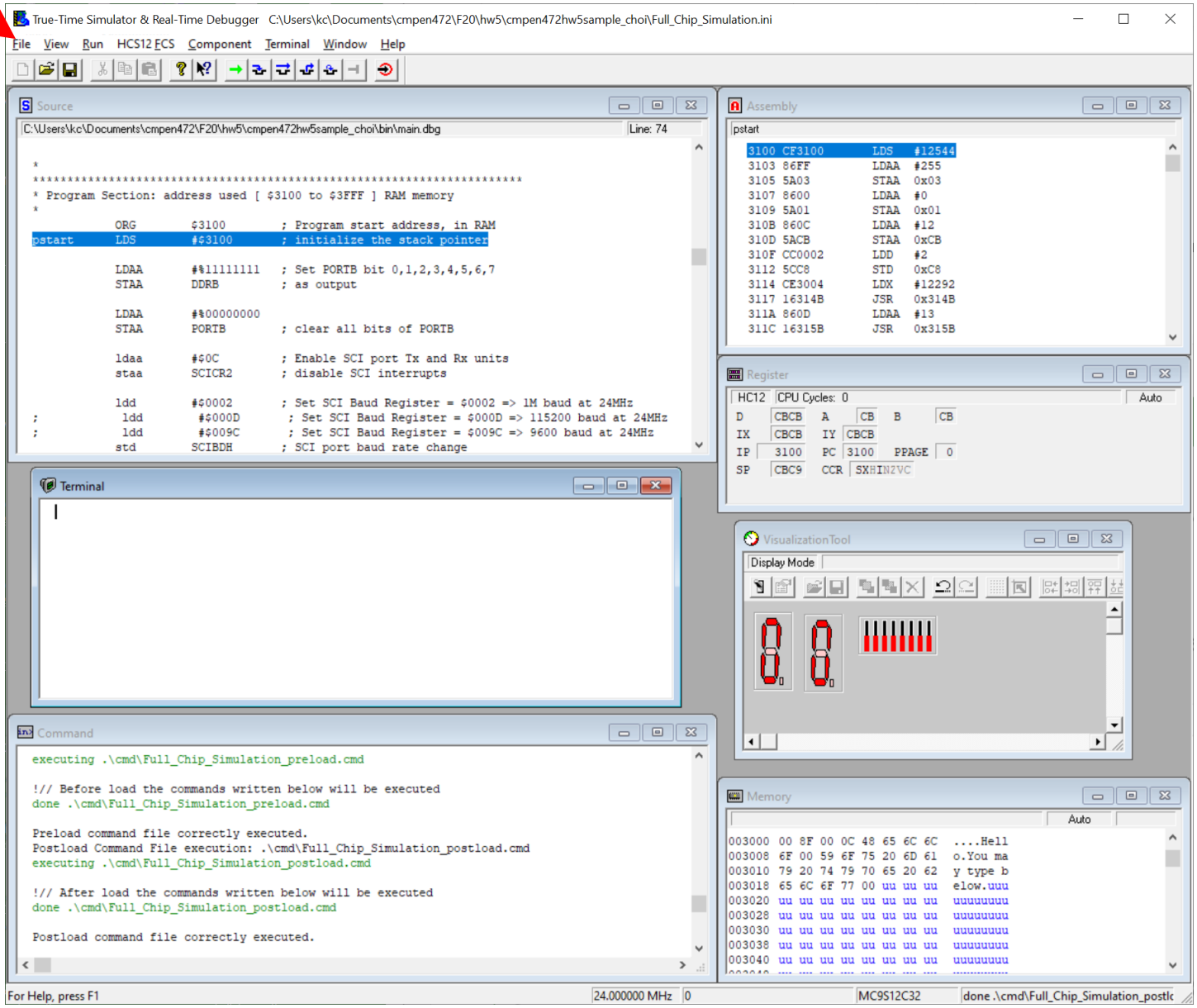

Open Window Component
Icon
List
Details
Data
Ddemasl
Hcs12xadrmap
Inspect
Mcuregister
Memory
Module
Procedure
Profiler
Recorder
Register
Source
Stimulation
Terminal
Trace
Visualization
Winlift
OK
Cancel
Help
Browse

Click on the Terminal window.
Then click on the ‘Terminal’ menu and select ‘Configure Connections ...’ option.

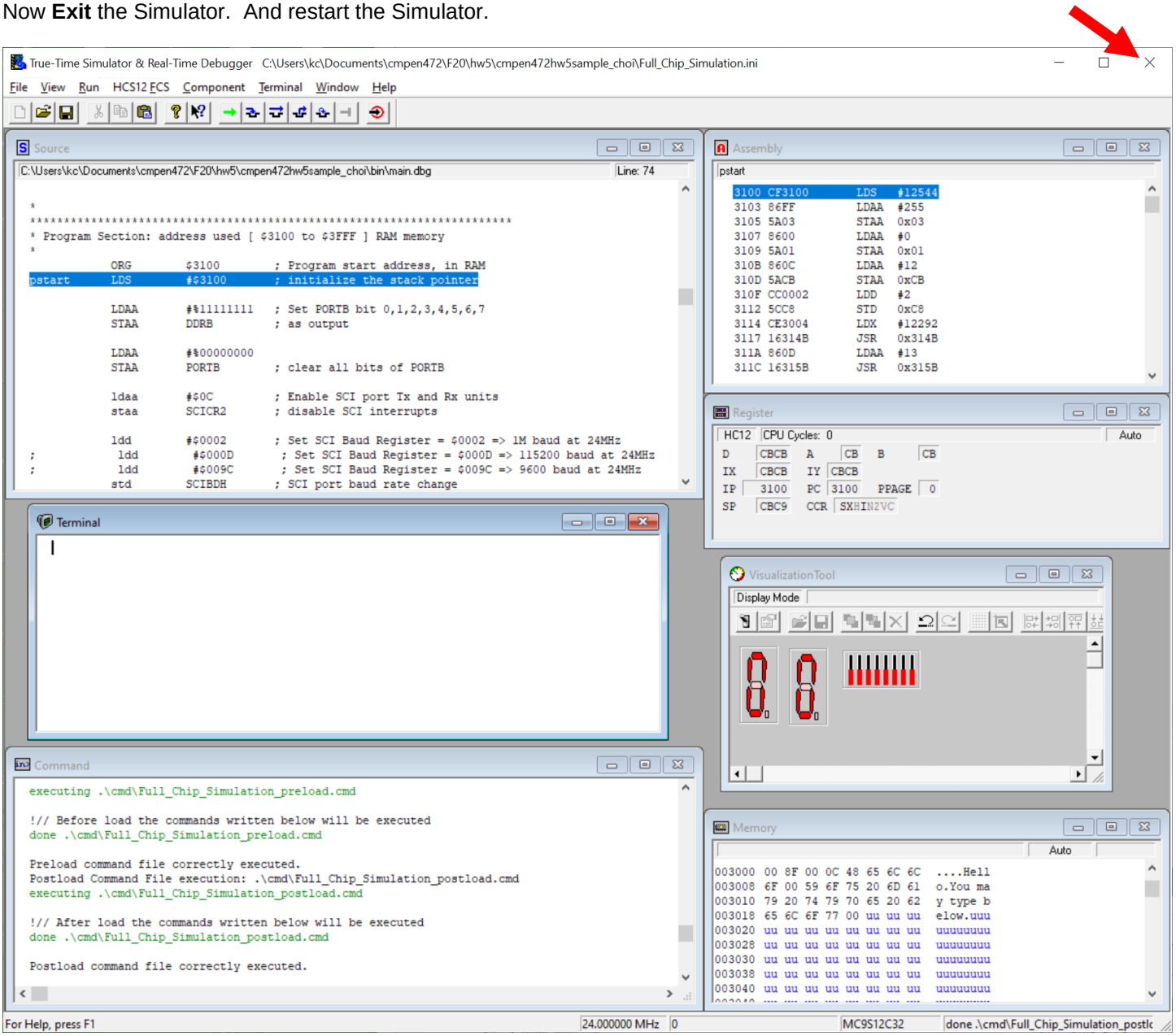


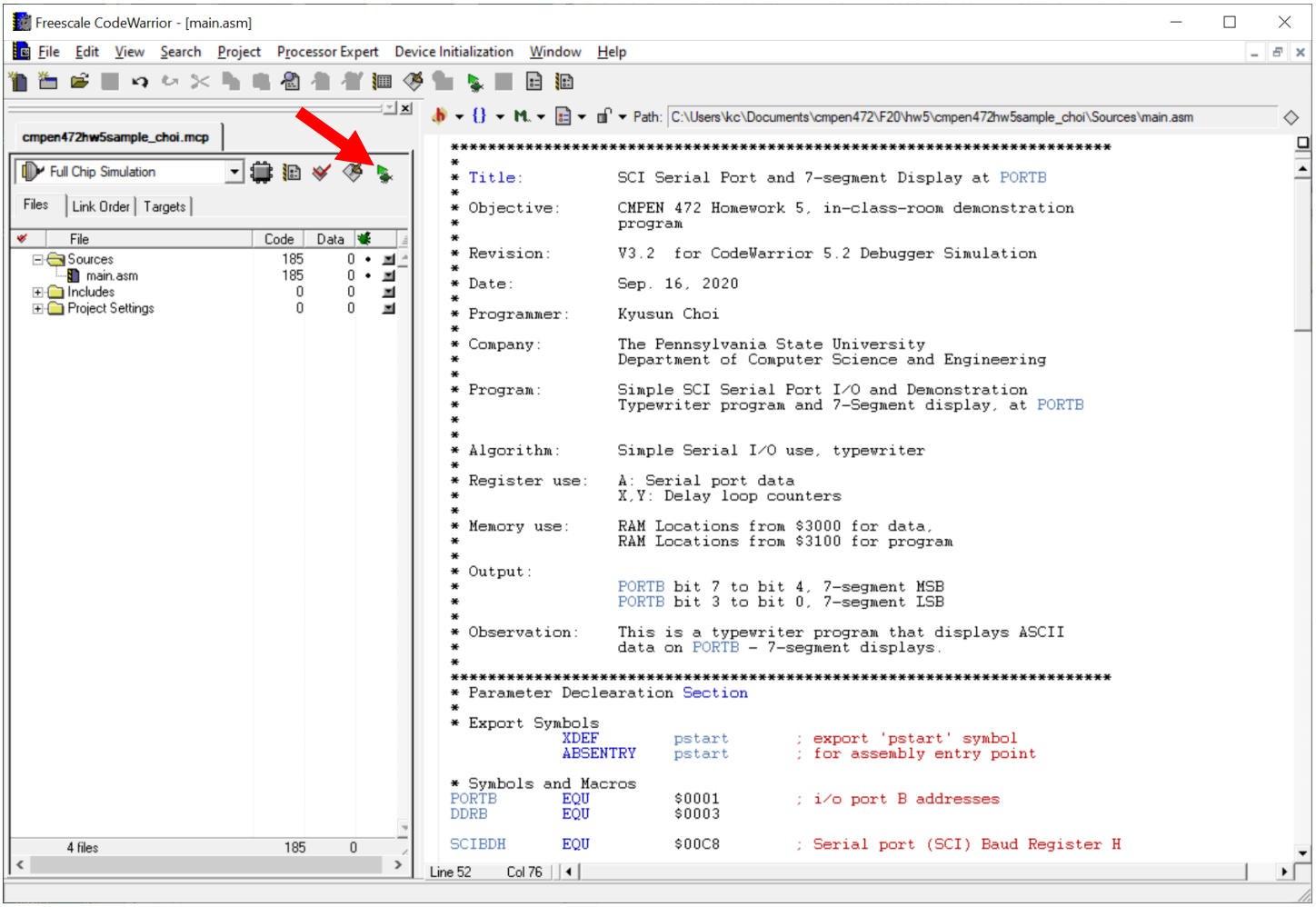

Configure Terminal Connections
Default Configuration:
Virtual SCI
Connections
From:
To:
Add
Input File --> Virtual SCI
Keyboard --> Virtual SCI
Virtual SCI --> Display
Virtual SCI --> Output File
Add All
Remove All
Remove
Serial Port
COM1
Baud Rate:
9600
Show Protocol
Virtual SCI
Virtual SCI Input Port:
Sci:2.SerialOutput
Virtual SCI Output Port:
Sci:2.SerialInput
OK
Cancel
Help

Click the ‘File’ menu and select ‘Save Configuration’ option, to save the current setting.

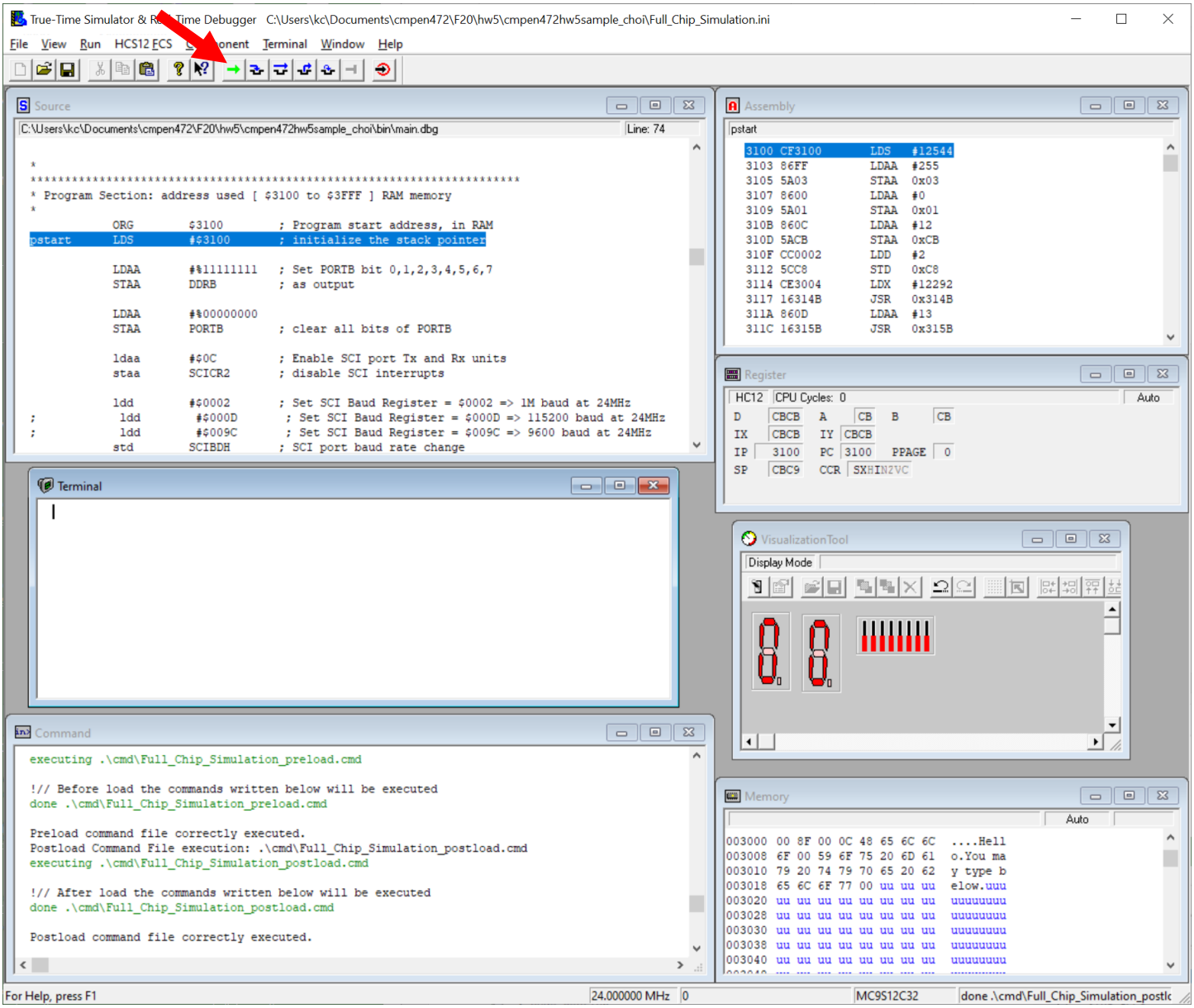


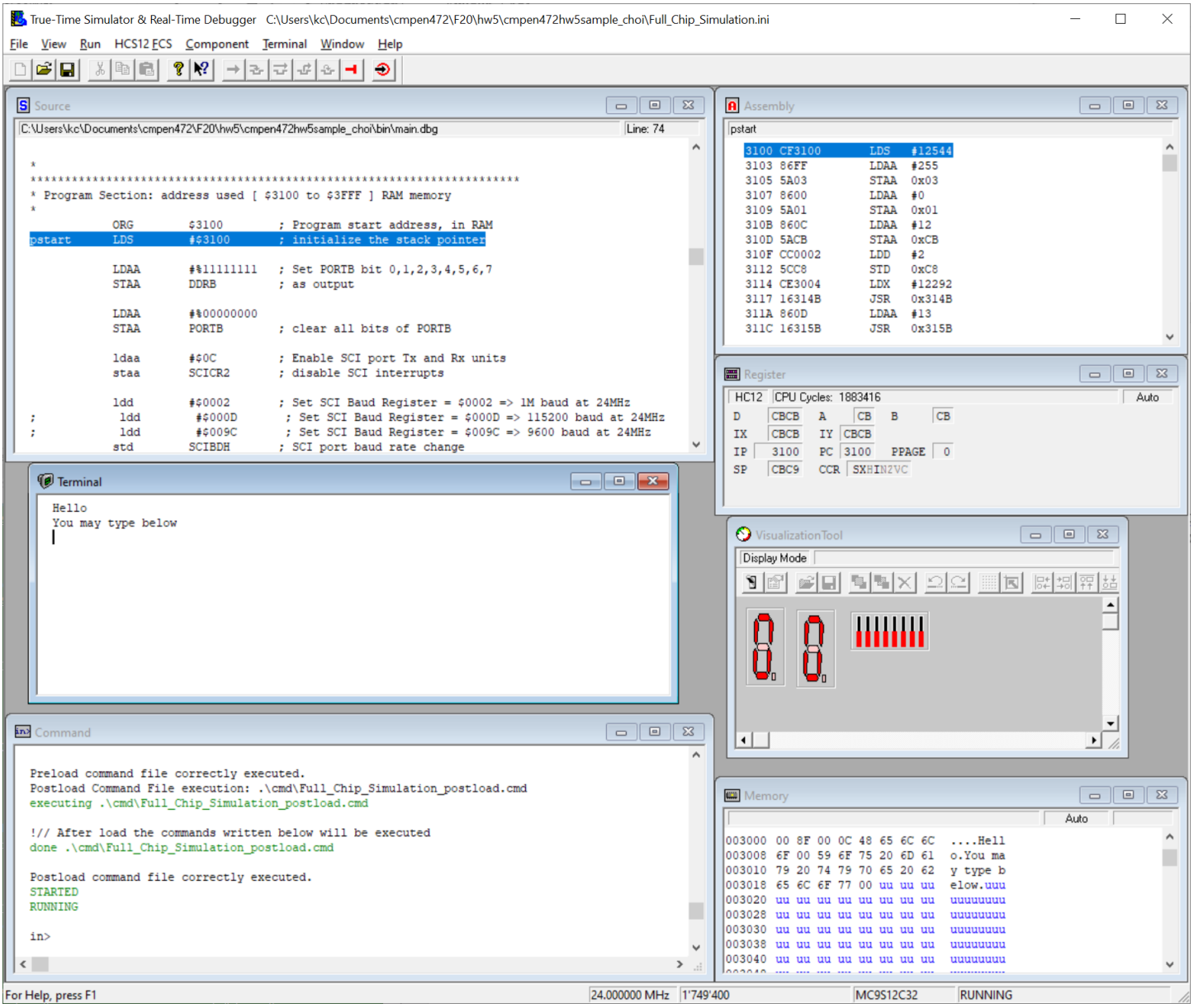
Now **Exit** the Simulator. And restart the Simulator.

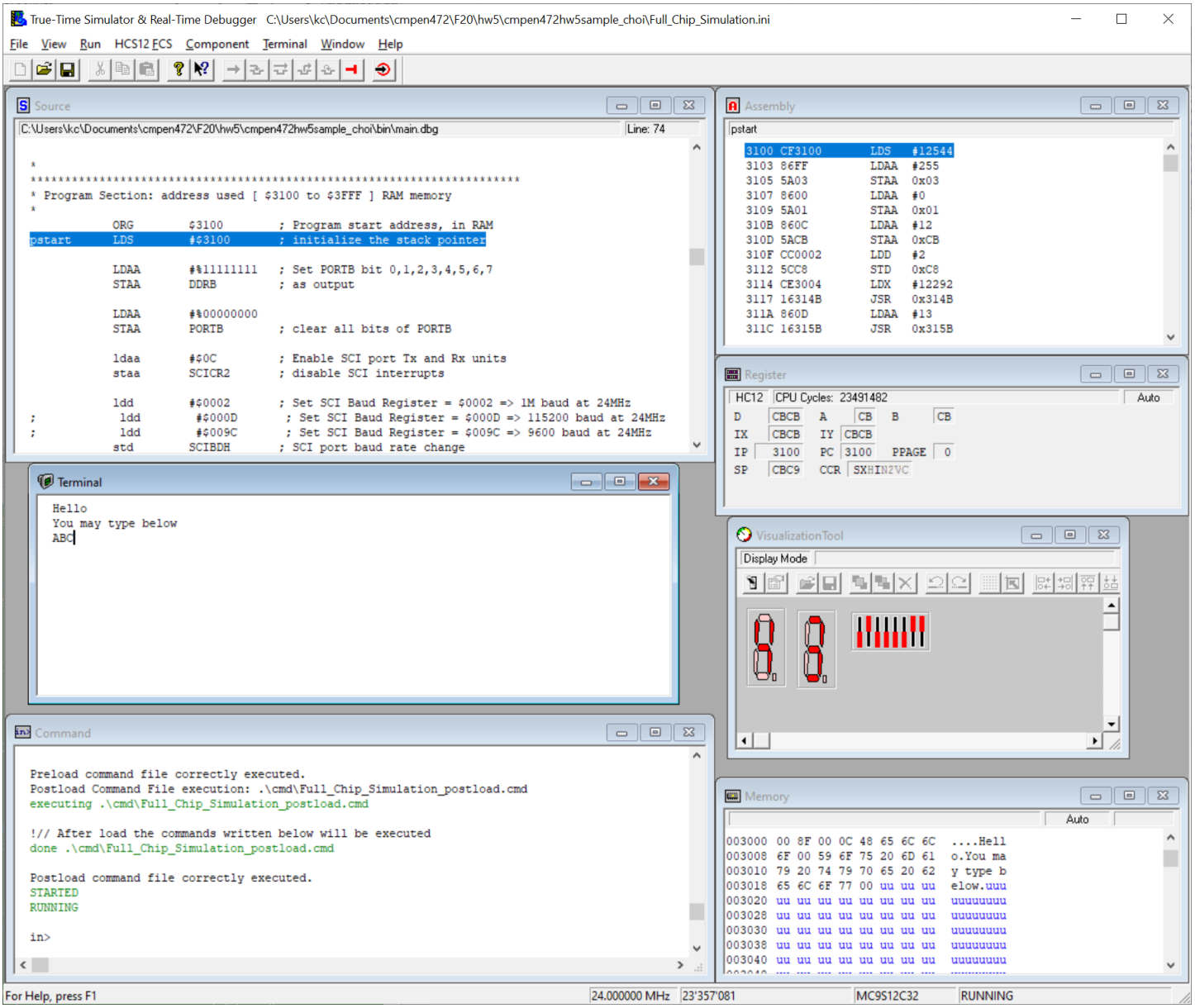



Freescale CodeWarrior - [main.asm]
File Edit View Search Project Processor Expert Device Initialization Window Help
cmpen472hw5sample_choi.mcp
Full Chip Simulation
Files Link Order Targets
File Code Data
Sources 185 0
main.asm 185 0
Includes 0 0
Project Settings 0 0
4 files 185 0
Path: C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\Sources\main.asm
* Title: SCI Serial Port and 7-segment Display at PORTB
* Objective: CMPEN 472 Homework 5, in-class-room demonstration program
* Revision: V3.2 for CodeWarrior 5.2 Debugger Simulation
* Date: Sep. 16, 2020
* Programmer: Kyusun Choi
* Company: The Pennsylvania State University
Department of Computer Science and Engineering
* Program: Simple SCI Serial Port I/O and Demonstration
Typewriter program and 7-Segment display, at PORTB
* Algorithm: Simple Serial I/O use, typewriter
* Register use: A: Serial port data
X,Y: Delay loop counters
* Memory use: RAM Locations from $3000 for data,
RAM Locations from $3100 for program
* Output:
PORTB bit 7 to bit 4, 7-segment MSB
PORTB bit 3 to bit 0, 7-segment LSB
* Observation: This is a typewriter program that displays ASCII
data on PORTB - 7-segment displays.
* Parameter Declearation Section
* Export Symbols
XDEF pstart ; export 'pstart' symbol
ABSENTRY pstart ; for assembly entry point
* Symbols and Macros
PORTB EQU $0001 ; i/o port B addresses
DDRB EQU $0003
SCIBDH EQU $00C8 ; Serial port (SCI) Baud Register H
Line 52 Col 76

Now run your program in Simulation.




True-Time Simulator & Real-Time Debugger C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\Full_Chip_Simulation.ini
File View Run HCS12 FCS Component Terminal Window Help
Source
C:\Users\kc\Documents\cmpen472\F20\hw5\cmpen472hw5sample_choi\bin\main.dbg
Line: 74
* Program Section: address used [$3100 to $3FFF] RAM memory
ORG $3100 ; Program start address, in RAM
pstart LDS #$3100 ; initialize the stack pointer
LDAA #%11111111 ; Set PORTB bit 0,1,2,3,4,5,6,7
STAA DDRB ; as output
LDAA #%00000000
STAA PORTB ; clear all bits of PORTB
ldaa #$0C ; Enable SCI port Tx and Rx units
staa SCICR2 ; disable SCI interrupts
ldd #$0002 ; Set SCI Baud Register = $0002 => 1M baud at 24MHz
; ldd #$000D ; Set SCI Baud Register = $000D => 115200 baud at 24MHz
; ldd #$009C ; Set SCI Baud Register = $009C => 9600 baud at 24MHz
std SCIBDH ; SCI port baud rate change
Assembly
pstart
3100 CF3100 LDS #12544
3103 86FF LDAA #255
3105 5A03 STAA 0x03
3107 8600 LDAA #0
3109 5A01 STAA 0x01
310B 860C LDAA #12
310D 5ACB STAA 0xCB
310F CC0002 LDD #2
3112 5CC8 STD 0xC8
3114 CE3004 LDX #12292
3117 16314B JSR 0x314B
311A 860D LDAA #13
311C 16315B JSR 0x315B
Register
HC12 CPU Cycles: 1883416 Auto
D CBCB A CB B CB
IX CBCB IY CBCB
IP 3100 PC 3100 PPAGE 0
SP CBC9 CCR SXHINZVC
Terminal
Hello
You may type below
VisualizationTool
Display Mode
Command
Preload command file correctly executed.
Postload Command File execution: .\cmd\Full_Chip_Simulation_postload.cmd
executing .\cmd\Full_Chip_Simulation_postload.cmd
!// After load the commands written below will be executed
done .\cmd\Full_Chip_Simulation_postload.cmd
Postload command file correctly executed.
STARTED
RUNNING
in>
Memory
Auto
003000 00 8F 00 0C 48 65 6C 6C ....Hell
003008 6F 00 59 6F 75 20 6D 61 o.You ma
003010 79 20 74 79 70 65 20 62 y type b
003018 65 6C 6F 77 00 uu uu uu elow.uuu
003020 uu uu uu uu uu uu uu uu uuuuuuuu
003028 uu uu uu uu uu uu uu uu uuuuuuuu
003030 uu uu uu uu uu uu uu uu uuuuuuuu
003038 uu uu uu uu uu uu uu uu uuuuuuuu
003040 uu uu uu uu uu uu uu uu uuuuuuuu
For Help, press F1
24.000000 MHz
1'749'400
MC9S12C32
RUNNING



While the program is running on the simulator, move your cursor on the Terminal window and hit any key. You will be able to see the characters on the Terminal window and see the characters on the Visualization Tool window LEDs. What you type is what you see. You may stop simulation by clicking the 'Halt' button on the Simulator window.