User Interface for eZdsp

EXPERIMENT 1.3

Propose of the experiment

- Continue from previous experiments to be familiar with CCS environment
- Write a user interface (UI) program in C to interact with the program from CCS Console Window
- The UI functions will be used in future experiments for runtime control of the experimental parameters
- Understand C functions that used for UI interface, condition (*if-else*), and flow control (*switch*) statement
- Familiar with CCS tools (Breakpoint, Memory Watch Window, Variable Watch Window, and Graph plotting)

Experiment preparation

- Start CCS
- Create workspace Exp1.3
- Create a new project and name it as userInterface
- Copy experiment files to workspace folder

Setup build environment

- Right click on project “UI” then select Property
- Select and expand C/C++ Build option
- Select Settings, then Runtime Options
- Set type size to 16 and memory model to large

Set target configuration

- Right click on Project->New->Target Configuration File
- Create a target configuration file name, *userInterface.ccxml*
- Select Texas Instruments XDS100v2 USB Emulator
- Check the box for USBSTK5505
- Save the configuration

Launch target

- From Target Configuration window
- Open Project and right click on *userInterface.ccxml*
- Select Launch target configuration
- In Debug window, right click on Texas Instruments XDS100v2 USB Emulator_0/C55xx
- Select Connect Target to launch the target
- You shall see target reset and configured automatically

Build and run the program

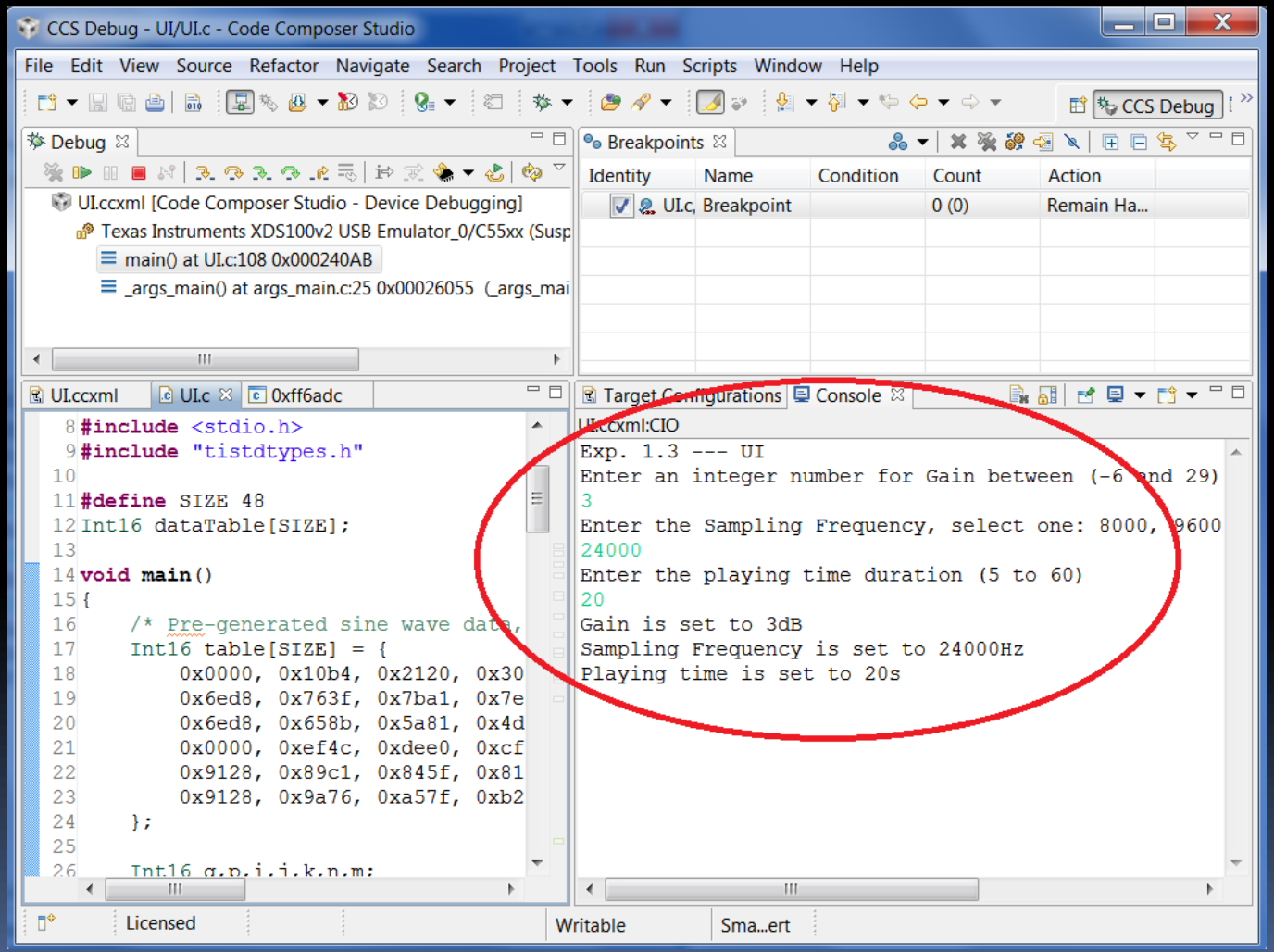
- Build the project (user Build All)
- Load the program
- Run the program
- Move mouse cursor to CCS Console Window, key in the proper numbers when UI program asks

Verify the program result

- Gain parameter initialization
  - The program accepts an initial gain from -6 to 29
  - Any value outside above range will be replaced with the default odB gain
- Sampling Frequency initialization
  - The program requests an input from one of the numbers listed: 8KHz, 12KHz, 16KHz, 24KHz, or 48KHz
  - Any other value given will be invalid it a invalid variable is entered, the Sampling Frequency will be set to default at 48KHz
- Playtime initialization
  - The program will set up the play duration, the range is between 5 seconds and 60 seconds. If the playing time is entered outside above range, the program uses 10 seconds as default
- The Gain, Sampling Frequency and Playtime will be displayed on the CCS Console
- Once the program completed, an array of data will be filled based on the Sampling Frequency

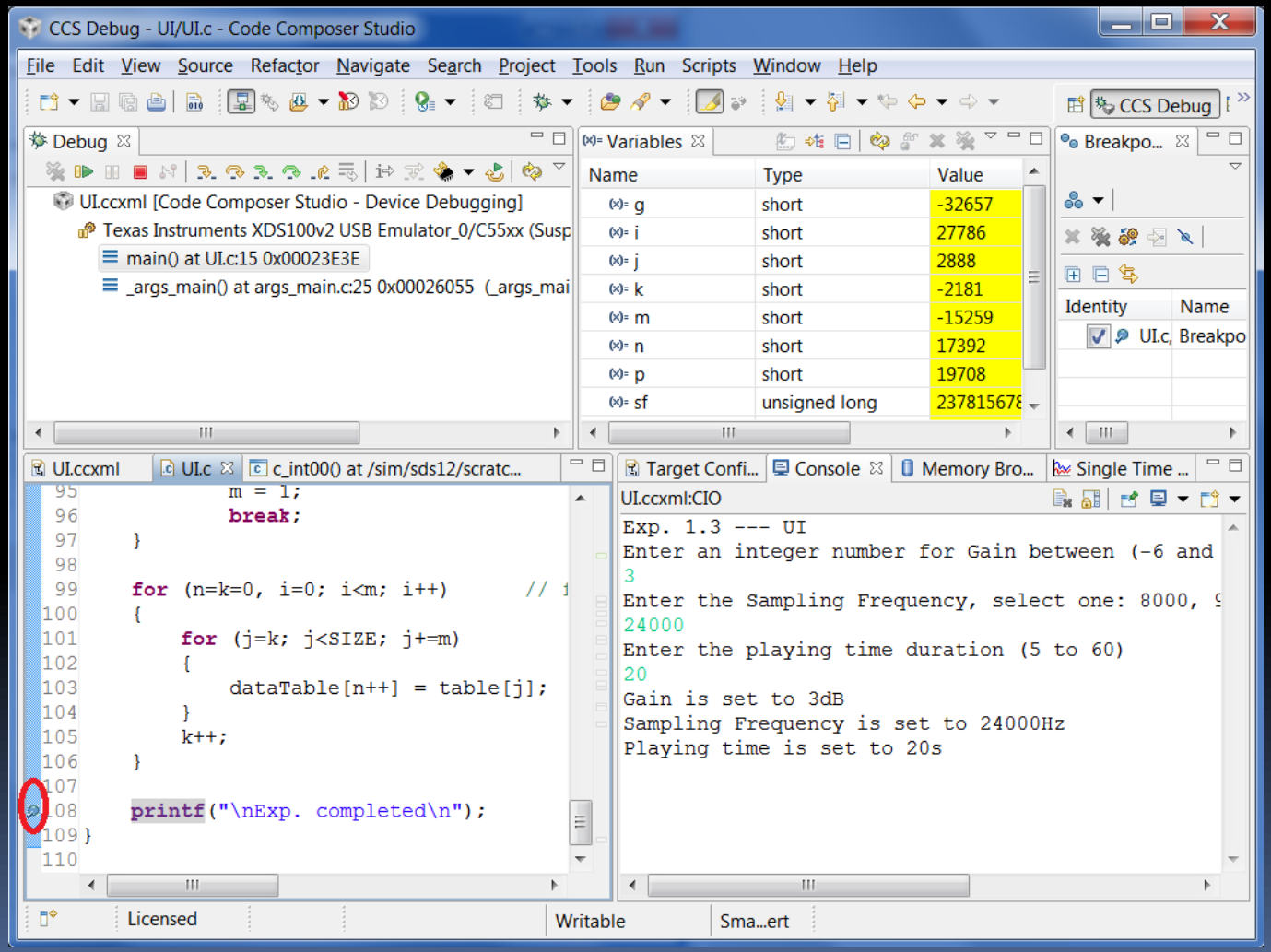
Program Console Interaction

(Example: Gain=3dB, Sampling Freq.=24000Hz, Playtime=20s



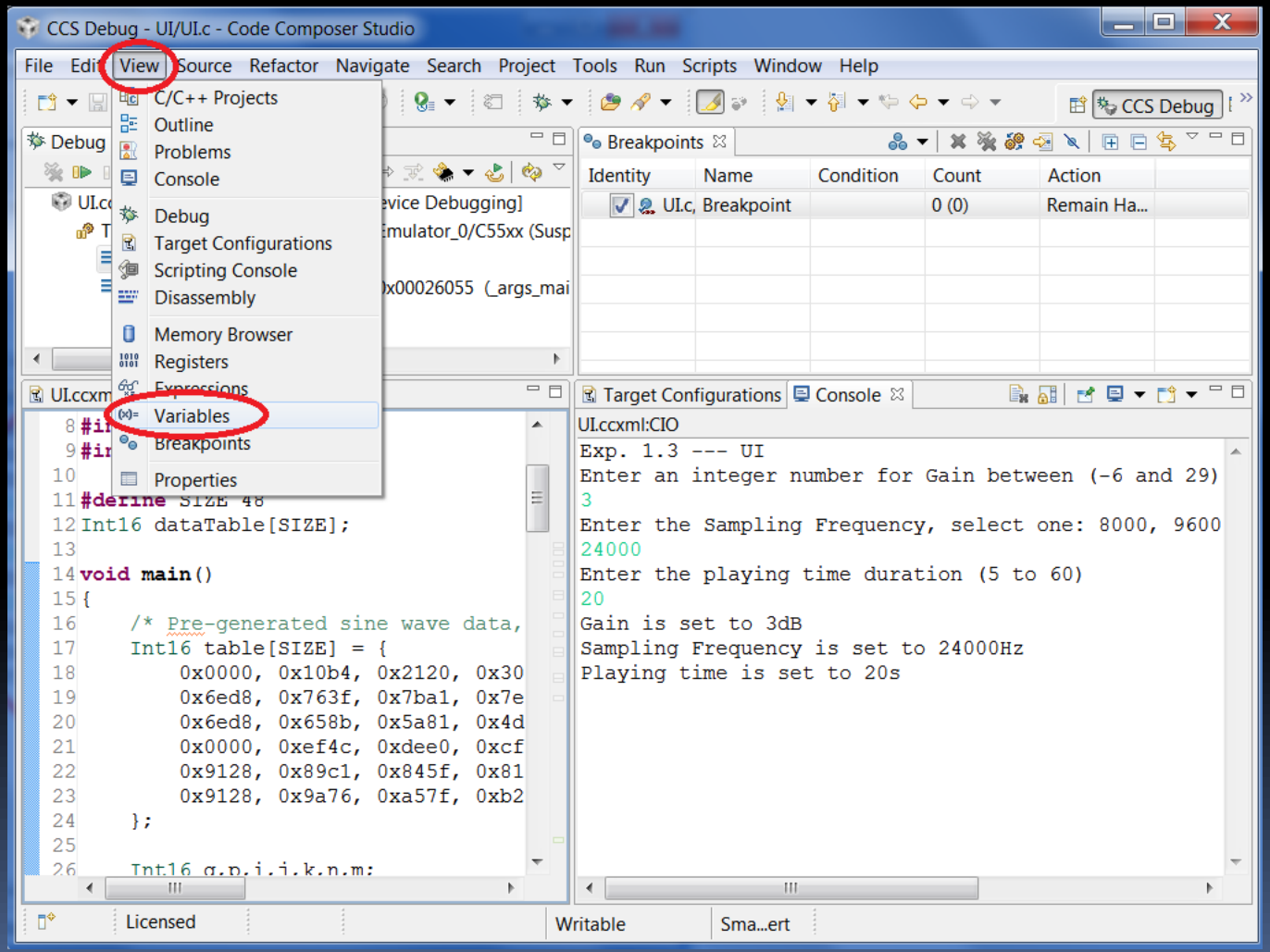
CCS Breakpoint

(Example: double click the line to set / remove breakpoint)



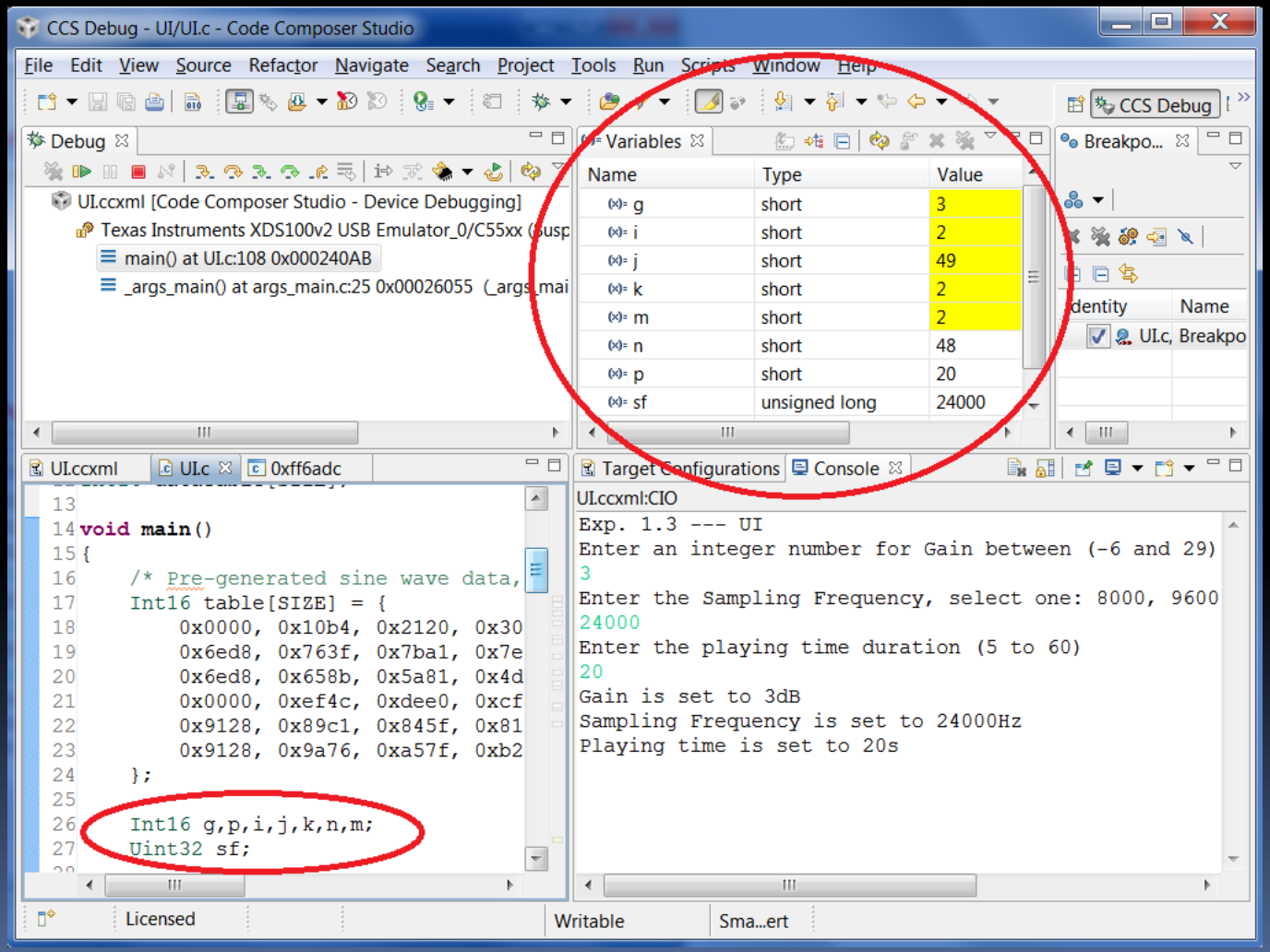
Set up Variable Watch Window

(Example: View->Variables to open variable viewing window)



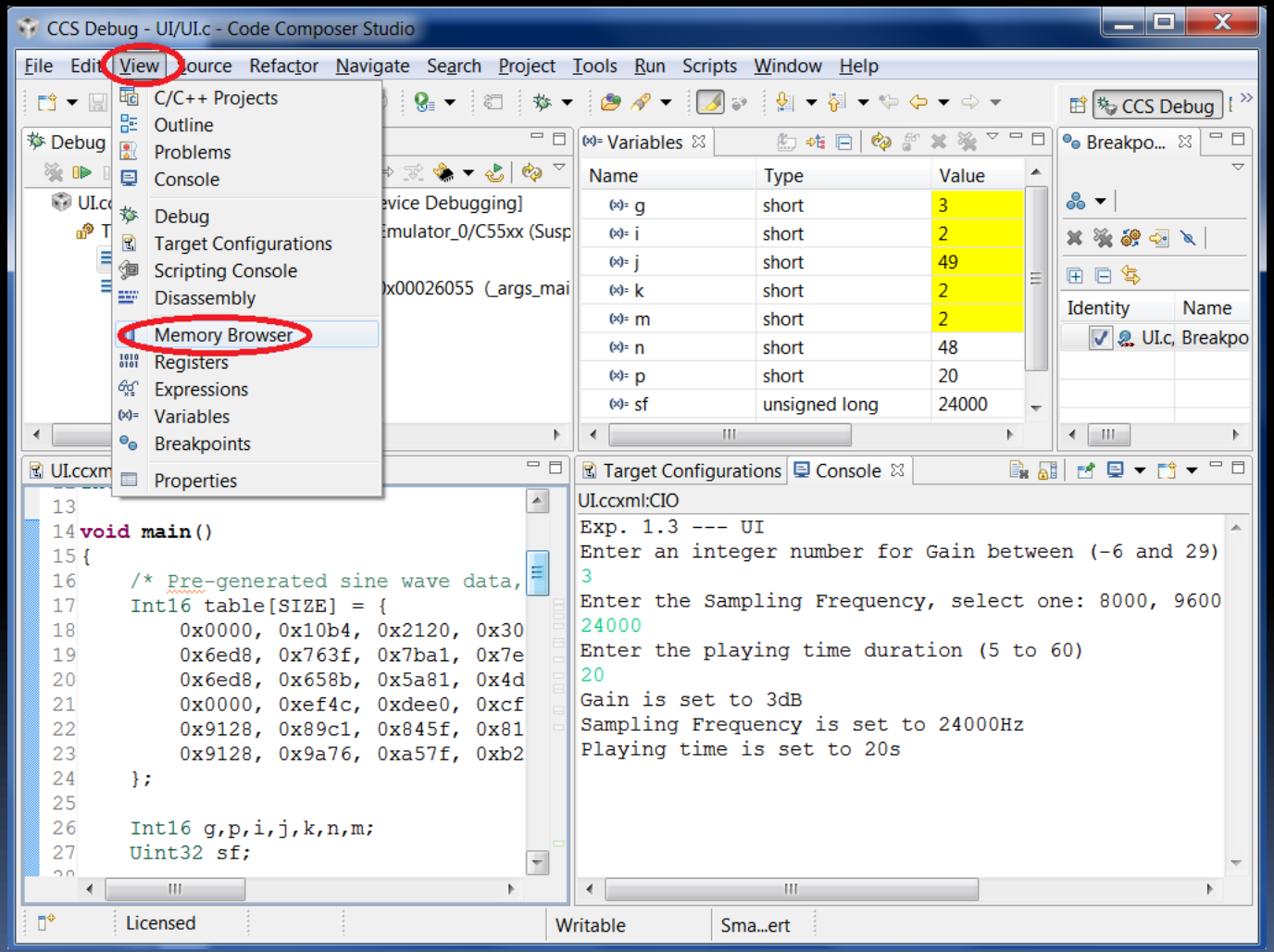
CCS Variable Watch Window

(Monitoring variables for debugging)



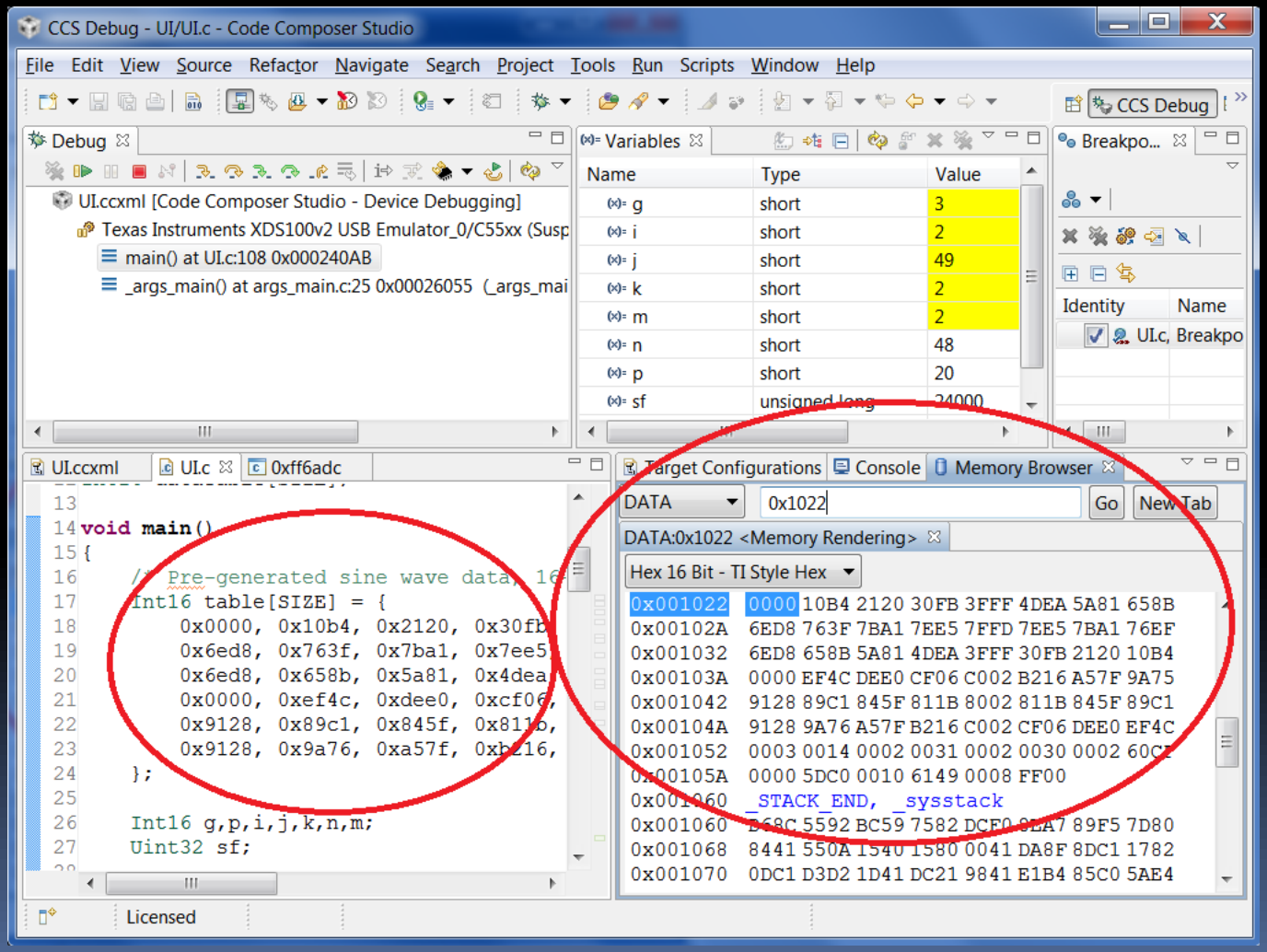
Set up Memory Watch Window

(Example: View->Memory Browser to open memory watch window)



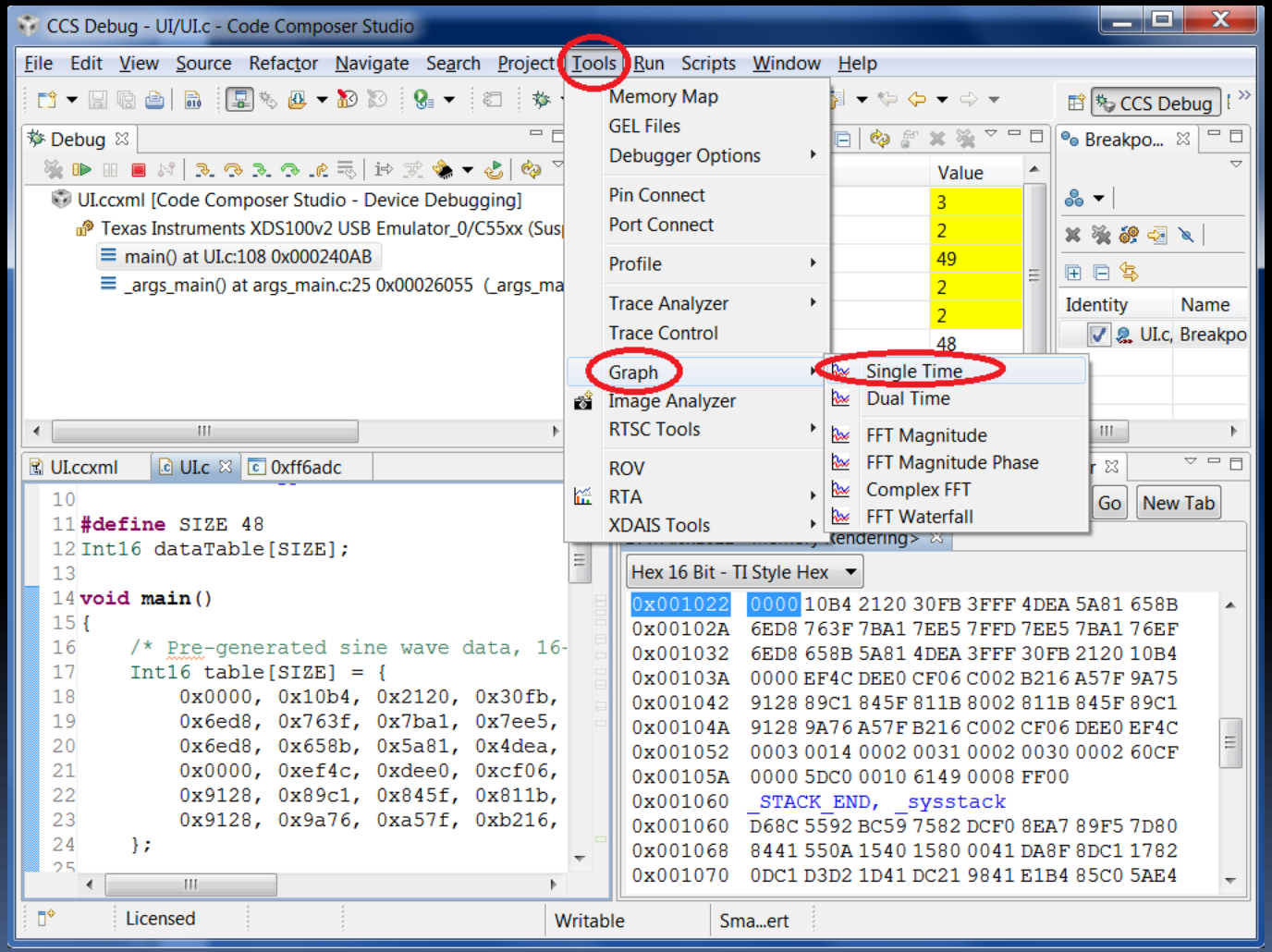
CCS Memory Watch Window

(Verifying data memory for debugging)



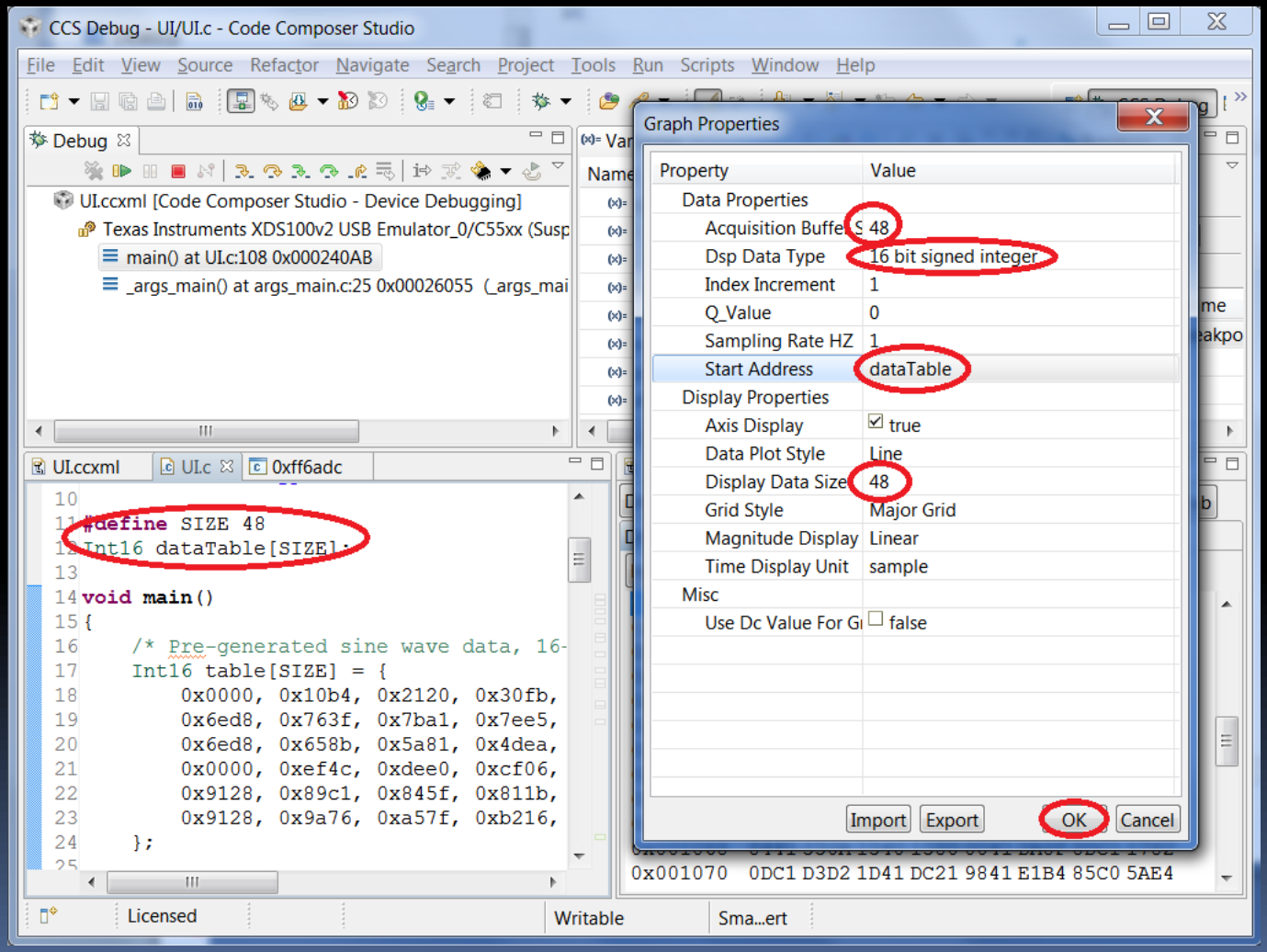
Set up Graph Tool for Plot

(Example: tools->Graph->Single Time)



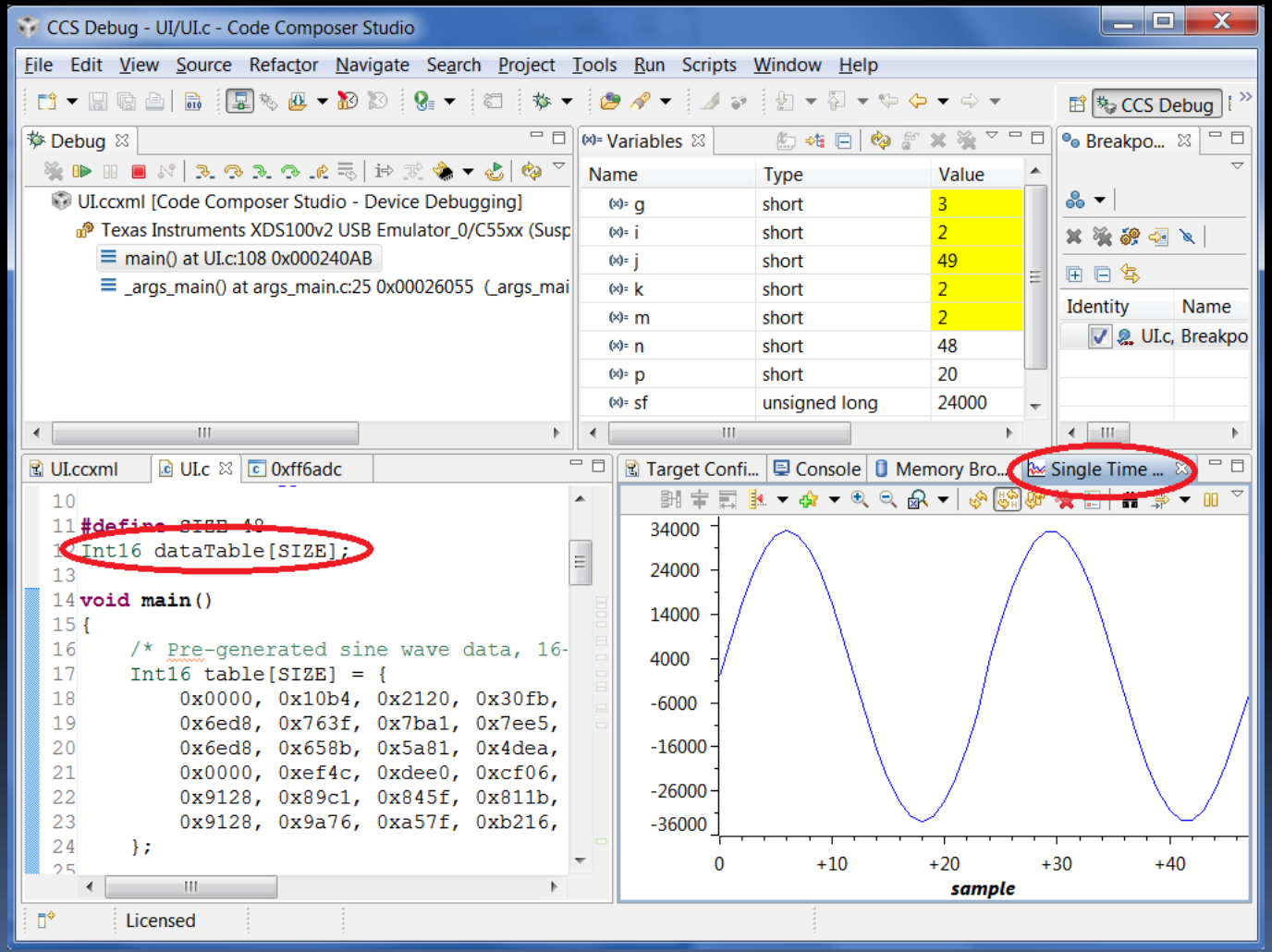
CCS Graph Parameter for Plot

(Example: Parameters to plot dataTable[])



CCS Graph Plot

(Example: dataTable[] has been plotted after execution)



New experiment assignments

- CCS Breakpoint
  - Build and load Exp1.3 program, set up a breakpoint on the last line of the program, and run the program (F8), what happens?
  - Reload program, set a new breakpoint in the middle of the program, run the program again. What happens? Resume the program (F8), what happens?
- CCS Data and Variable Watch Windows
  - Set up variable watch window, as Exp1.3. What other data types the watch window can supports? Change the data type and observe the watch window display.
  - How can you find out what memory address to use for setting up the watch windows?
  - Setup both data and variable watch windows, single step through the program and watch how the data and variables being updated.
  - Do you know you can modify data memory via CCS by directly editing the memory location?
- CCS Graph Tool
  - Restart the Exp1.3 program and check what graphs CCS can plot. How do you select the Graph parameters, why? Plot the data[ ] in the same window.
  - Add x-y axis labels to your plot. Add a grid to the graph plot.
  - Change the display from line to large square.
- CCS Programming
  - Replace the *switch ( )* statement in Exp1.3 with *if-else* statement and re-run the experiment. That is, the program uses only the *if-else* statement. If you see incorrect results, using breakpoint and watch window tools to debug your program and fix the error.
  - Replace the *if-else* statement in Exp1.3 with *switch ( )* statement and re-run the experiment. That is, the program uses only the *switch ( )* statement.
- CCS User Specified File Name
  - Write a program that will read character string from CCS console.
  - Modify Exp 1.2 such that you can tell the program the file name of the output WAV file through the CCS console.

Programming quick review

- The *scanf( )* function is used to accept user keyboard (standard input device) inputs. The basic data type the *scanf( )* can accept includes “%c”, “%s”, “%i”, “%d”, “%x”, “%u”, “%f”, for character, stream, integer, decimal integer, hexadecimal, unsigned decimal integer, and floating-point data.
- This experiment used C conditional statement *if-else* to check the parameters entered by user. Different actions will be taken depend upon the condition is true or false.
- To control the program flow, the *switch( )* statement is used.

References

- C Programming Language (2nd Edition), by Brian Kernighan and Dennis Ritchie, Prentice Hall