<Ex4>:

Objective: Write a program for Basic file I/O for CCS5.3 using EPB_C5515 target board (,i.e., reading/writing data from/to a file stored in your system into code composer studio)

Workflows you learned in the previous lab

- Connecting your DSP kit EPB_C5515 to CCS5.3
- Creating a new project or copying an existing project into the workspace
- Configuring the linker options and file-search paths
- Building/Compiling and running/Executing a project on the kit EPB_C5515
- Making use of breakpoints for debugging the code and using watch window to track variable values.

The above workflows will be frequently required in this and all the other lab-sessions.
If you get stuck somewhere while performing them, go back to Lab 1 manual. (Chapter 5.1 to chapter 5.3)

After reading this section you will be able to,

- Write the program for File input/output from PC to the CCS for TMS320C5515
- i.e., reading/writing data from/to a file stored in your system into code composer studio
- Generate graph from the stored buffer in the Code composer Studio 5.3
- Create the .dat file for square wave

Hardware Part List:

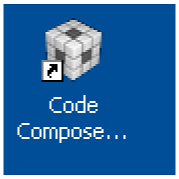
- PC
- Code Composer Studio v5.3
- +5v DC Power supply
- EPB_C5515
- Emulator + Emulator cable (USB A to Mini-A Cable, 14 pin FRC Flat cable)

List of Files Required:

sine.c	(Program application file)
sine_int16.dat	(sine database file to take input from PC)
lnkx.cmd	(Command file)
usbstk5515bsl.lib	(Library file)

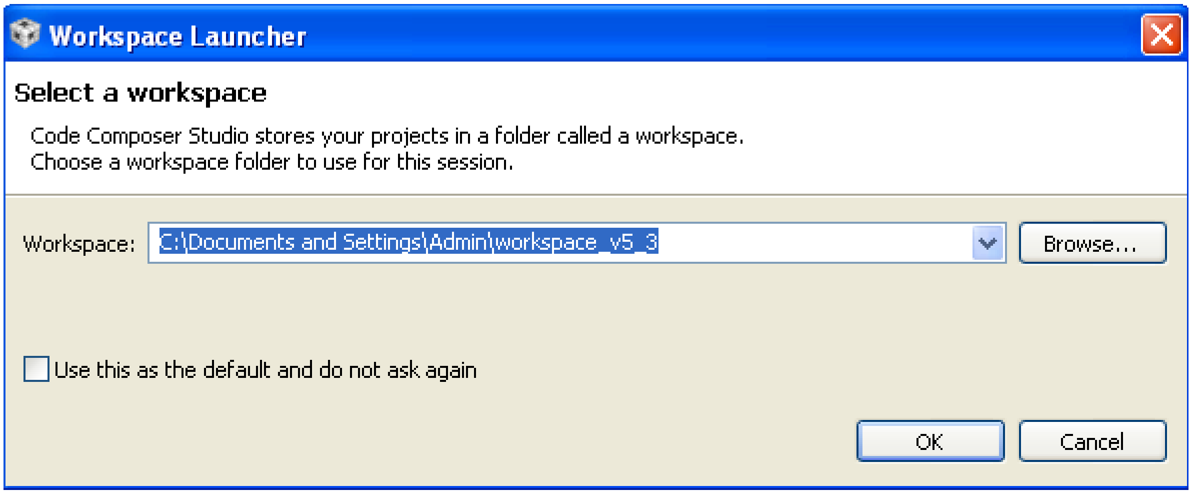
Steps for creating new project:
(We will create new project for hello world and then will edit that project)

Open *CCS V5.3* from desktop shortcut



It will open default *CCS V5* screen.

Then it will ask for workspace path
Select path “*C:\Documents and Settings\<User Name>\workspace_v5_3*” for windows XP OS
Select path “*C:\Users\<User Name>\workspace_v5_3*” for windows7 OS



Workspace Launcher

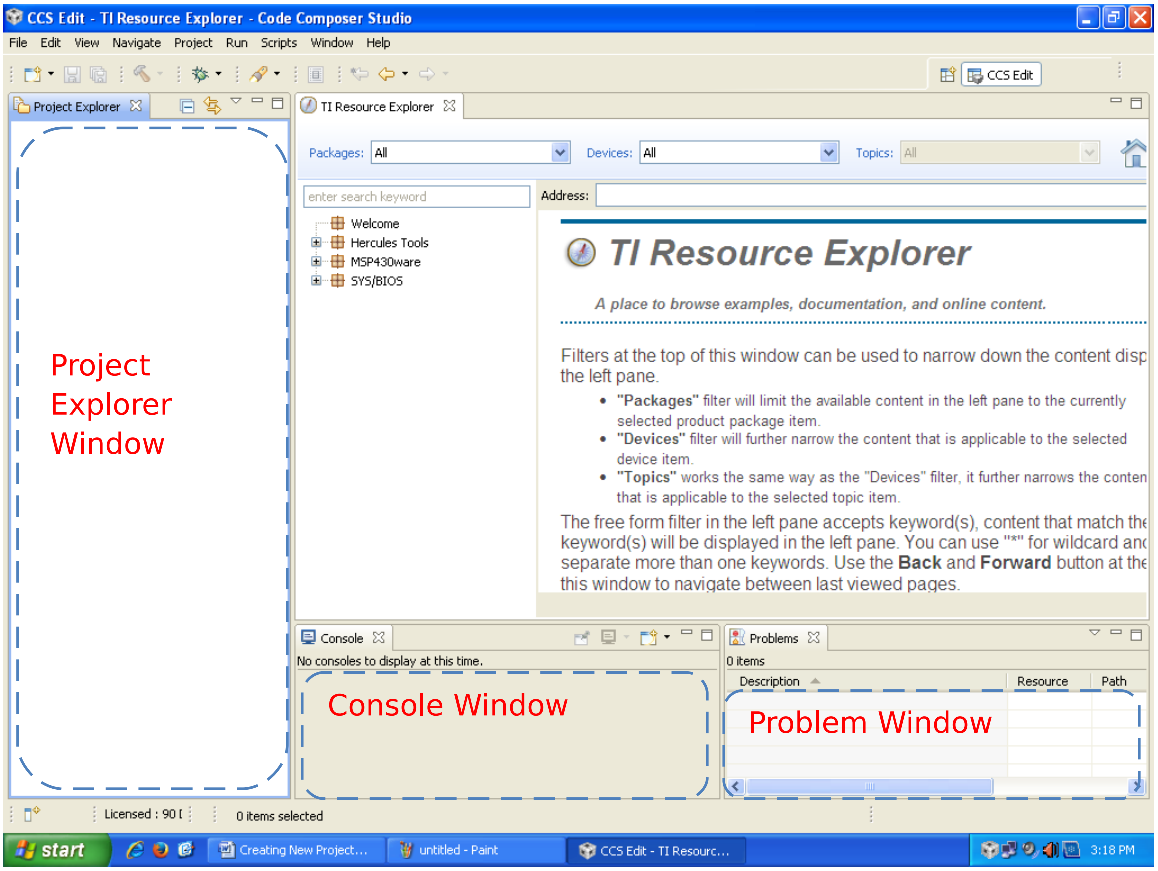
Select a workspace

Code Composer Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Documents and Settings\Admin\workspace_v5_3

□ Use this as the default and do not ask again

Then it will open Default CCS5 screen as shown below



CCS Edit - TI Resource Explorer - Code Composer Studio

TI Resource Explorer

A place to browse examples, documentation, and online content.

Filters at the top of this window can be used to narrow down the content disp the left pane.

- **"Packages"** filter will limit the available content in the left pane to the currently selected product package item.
- **"Devices"** filter will further narrow the content that is applicable to the selected device item.
- **"Topics"** works the same way as the "Devices" filter, it further narrows the conten that is applicable to the selected topic item.

The free form filter in the left pane accepts keyword(s), content that match the keyword(s) will be displayed in the left pane. You can use "*" for wildcard anc separate more than one keywords. Use the **Back** and **Forward** button at the this window to navigate between last viewed pages.

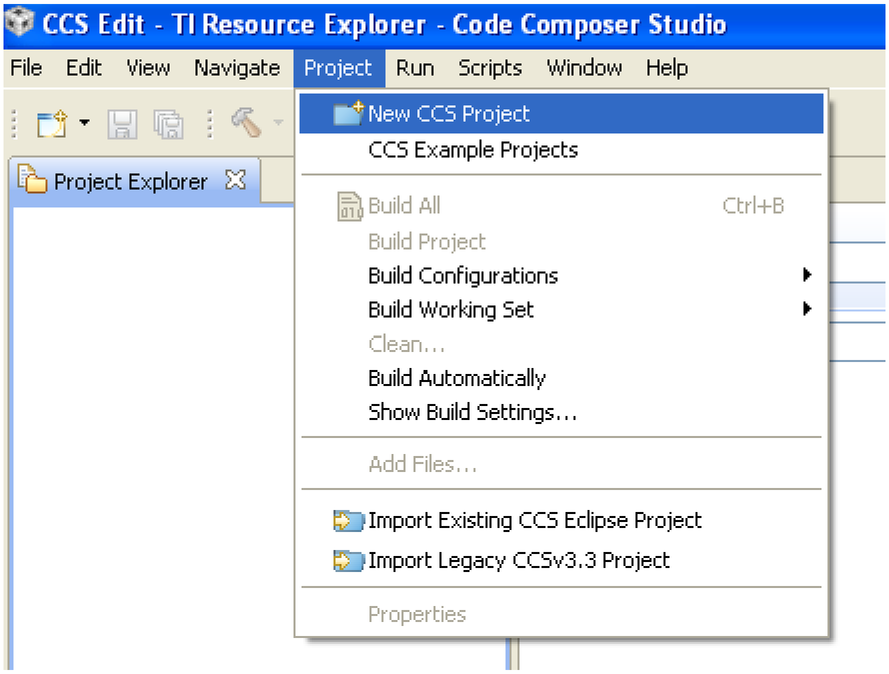
No consoles to display at this time.

0 items

Licensed : 90 I

0 items selected

Click "*Project -> New CCS Project*" menu.

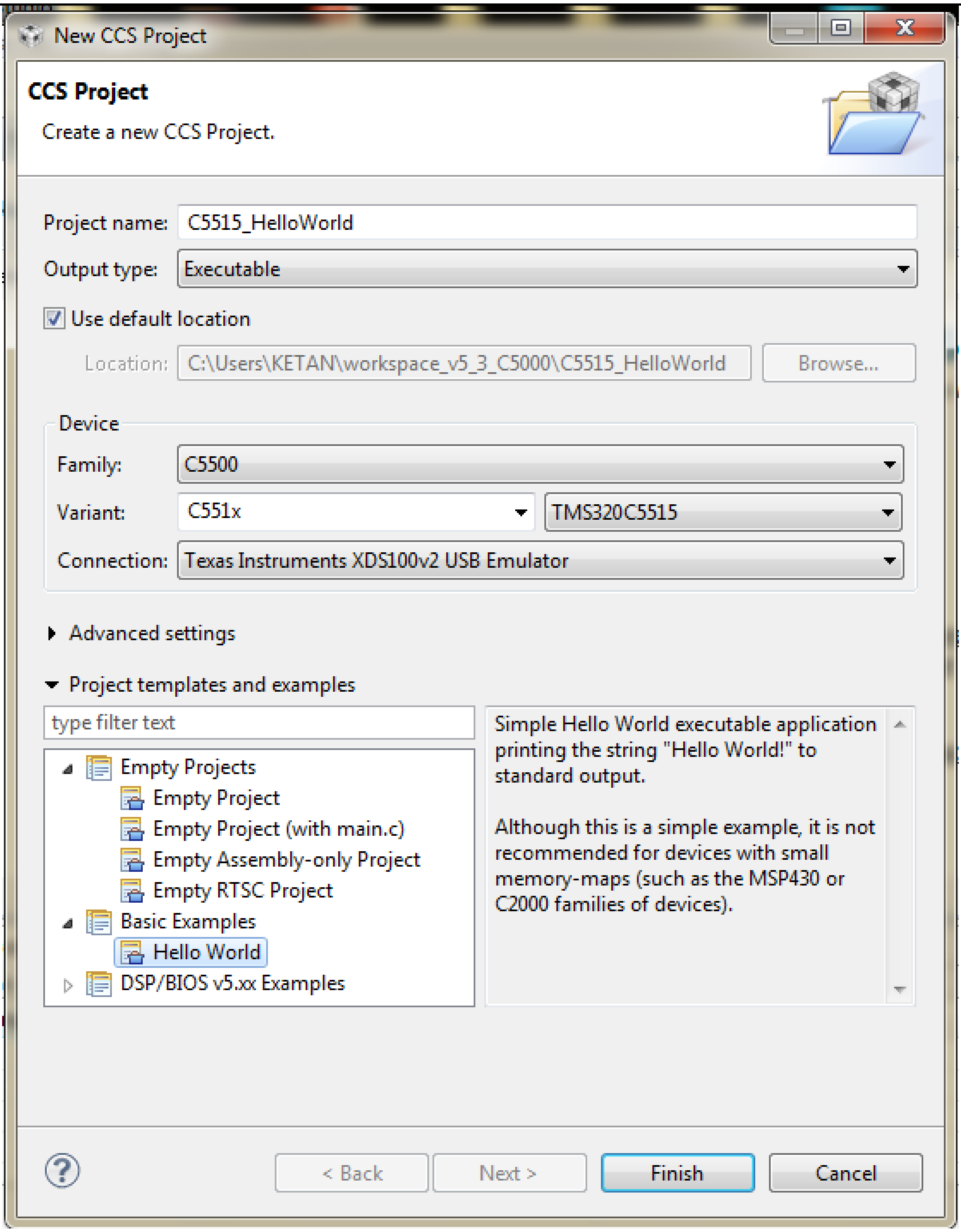


It will open following screen

Project name as desired, - e.g. "C5515_hello_world"
Output type: *Executable* as in figure.
And keep selected "***use default location*** " so that project will be created in workspace with project name typed

Select family: *C5500*,
Variant: *C551x*
Processor: *TMS320C5515*
And use **connection** type as *Texas Instruments XDS100V2 USB Emulator.*

Then at last select "*Hello World*" example from "*Basic Examples*" location in **Project Templates and example** tab. And **Finish**



New CCS Project

CCS Project

Create a new CCS Project.

Project name: C5515_HelloWorld

Output type: Executable

☑ Use default location

Location: C:\Users\KETAN\workspace_v5_3_C5000\C5515_HelloWorld

Device

Family: C5500

Variant: C551x TMS320C5515

Connection: Texas Instruments XDS100v2 USB Emulator

Advanced settings

Project templates and examples

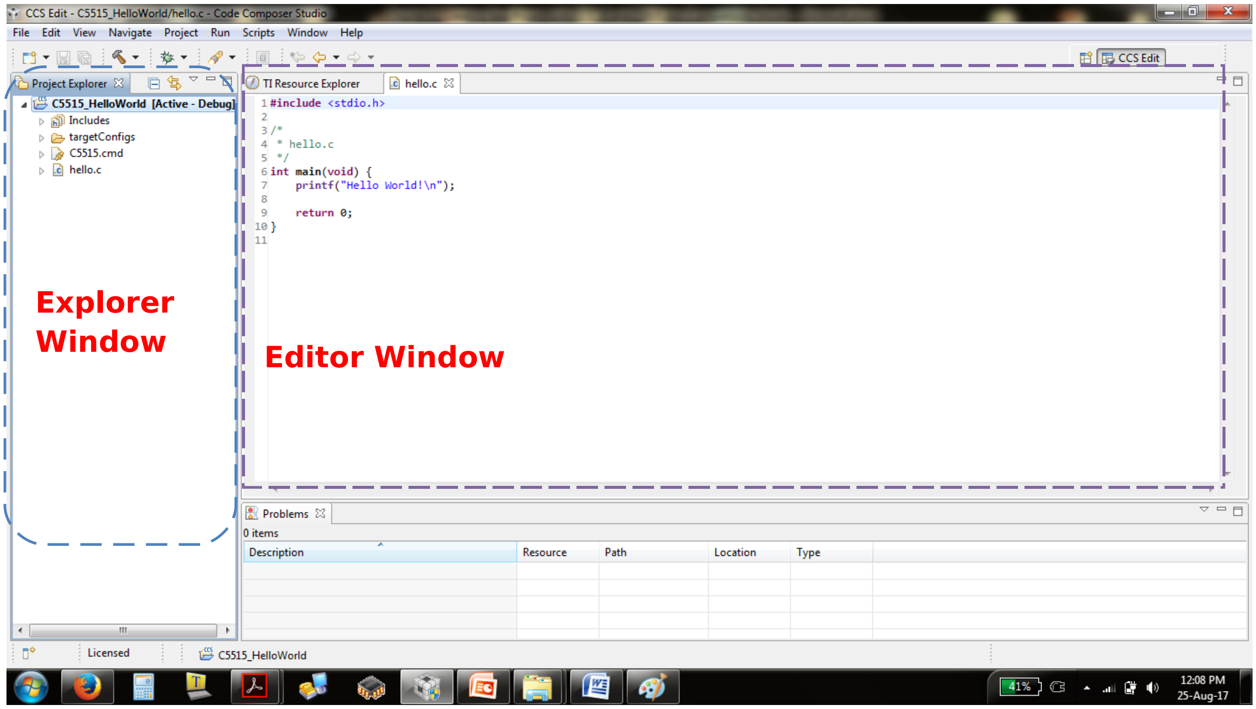
- Empty Projects
  - Empty Project
  - Empty Project (with main.c)
  - Empty Assembly-only Project
  - Empty RTSC Project
- Basic Examples
  - Hello World
- DSP/BIOS v5.xx Examples

Simple Hello World executable application printing the string "Hello World!" to standard output.

Although this is a simple example, it is not recommended for devices with small memory-maps (such as the MSP430 or C2000 families of devices).

It will open screen as shown here. Here project is already created and it can be seen from “**project explorer**”

Editor window will show *hello.c* file which can be edited as per requirement.



```
#include <stdio.h>

/*
 * hello.c
 */
int main(void) {
    printf("Hello World!\n");

    return 0;
}

```

- Delete **C5515.cmd** file from the project explorer and copy-paste **lnkx.cmd** file from the CD content given or reference example given.
- Delete **hello.c** file from the project explorer and copy-paste **sine.c** file from the CD content given or reference example “LAB2/IIT_Lab_2.1” given.
- Make sure **“int x[100]”** is present in the sine.c file.
- Also copy-paste **sine_int16.dat** file from the CD content given or reference example “LAB2/IIT_Lab2.1” given to the current project
- Rebuild/Build/Compile the project

Steps to Run the project:

Steps for Hardware connection:

- Power on EPB_C5515 hardware using +5V Power supply or USB A-to-B cable
- Connect XDS100V2 with EPB_C5515 using USB A-to-miniA cable with CPU
- Reset CPU

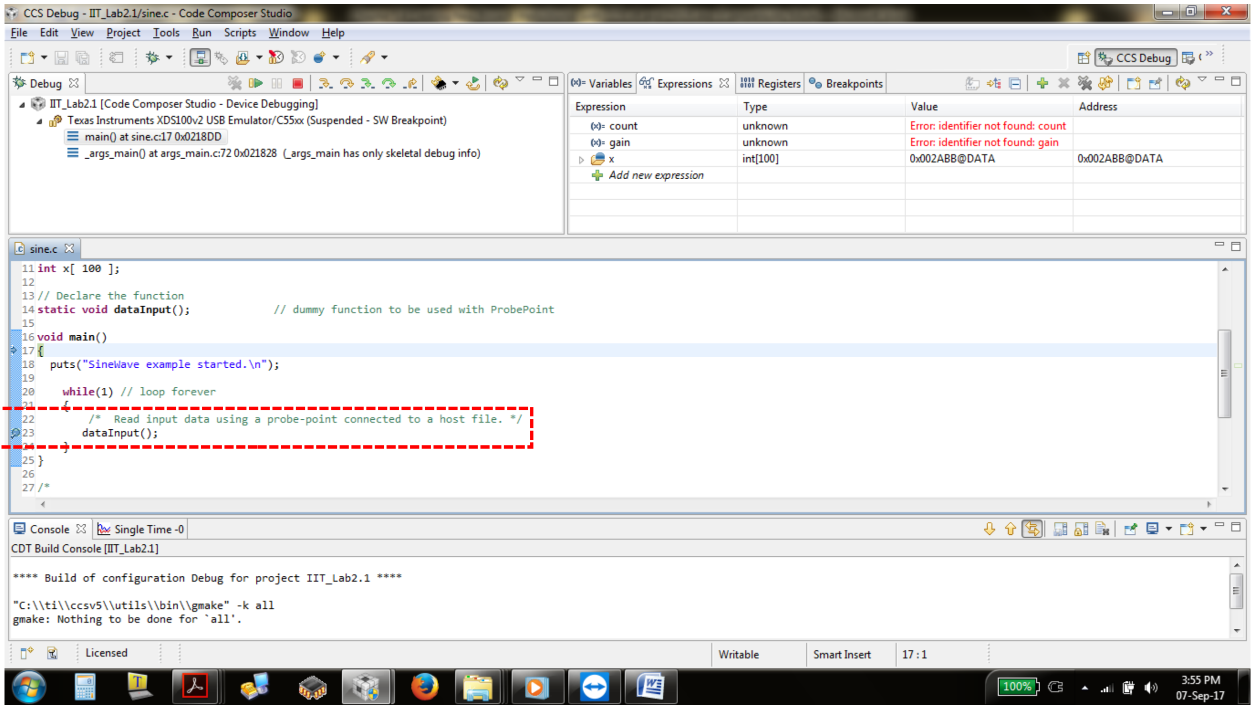
Steps to run/debug program:

- Now to debug the program click "**debug**" as shown in the screen from home screen icon **OR** from "**run->debug**" menu.

- It will configure/connect EPB_C5515 kit with the CCSV5 using XDS100V2 and download the program in C5515 CPU. It will be done automatically.

- Once Configuration is over, it will start loading program into the CPU using JTAG emulator

- Now open the sine.c file again and put a breakpoint at dataInput(). You can keep breakpoint by double clicking on the left side of line number 23.



IIT_Lab2.1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Emulator/C55xx (Suspended - SW Breakpoint)
main() at sine.c:17 0x0218DD
_args_main() at args_main.c:72 0x021828 (_args_main has only skeletal debug info)

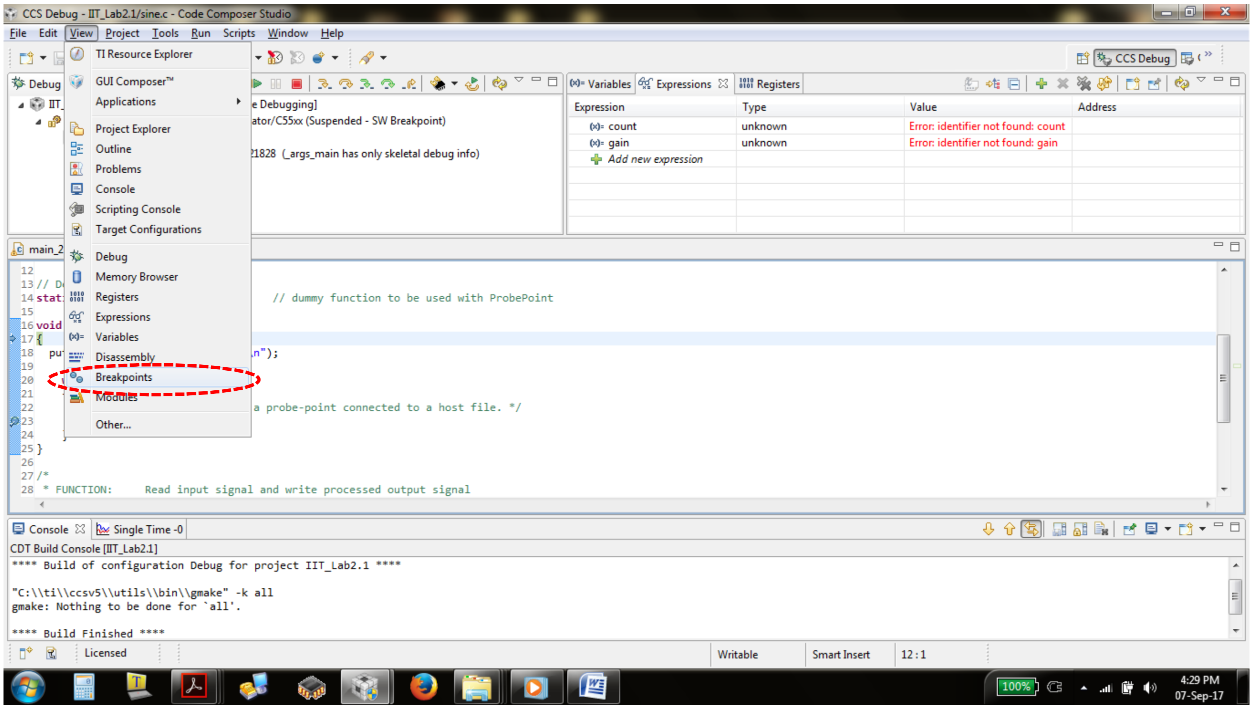
Expression	Type	Value	Address
count	unknown	Error: identifier not found: count	
gain	unknown	Error: identifier not found: gain	
x	int[100]	0x002ABB@DATA	0x002ABB@DATA
Add new expression			

```
11 int x[ 100 ];
12
13 // Declare the function
14 static void dataInput();             // dummy function to be used with ProbePoint
15
16 void main()
17 {
18   puts("SineWave example started.\n");
19
20     while(1) // loop forever
21     {
22         /*  Read input data using a probe-point connected to a host file. */
23        dataInput();
24     }
25 }
26
27 /*
```

CDT Build Console [IIT_Lab2.1]

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.
```

- Now Open the breakpoints window by clicking “**View->Breakpoints**” and select the Actions column (you need to scroll a bit to right) of the marked breakpoint.



e Debugging]
ator/C55xx (Suspended - SW Breakpoint)
21828 (_args_main has only skeletal debug info)

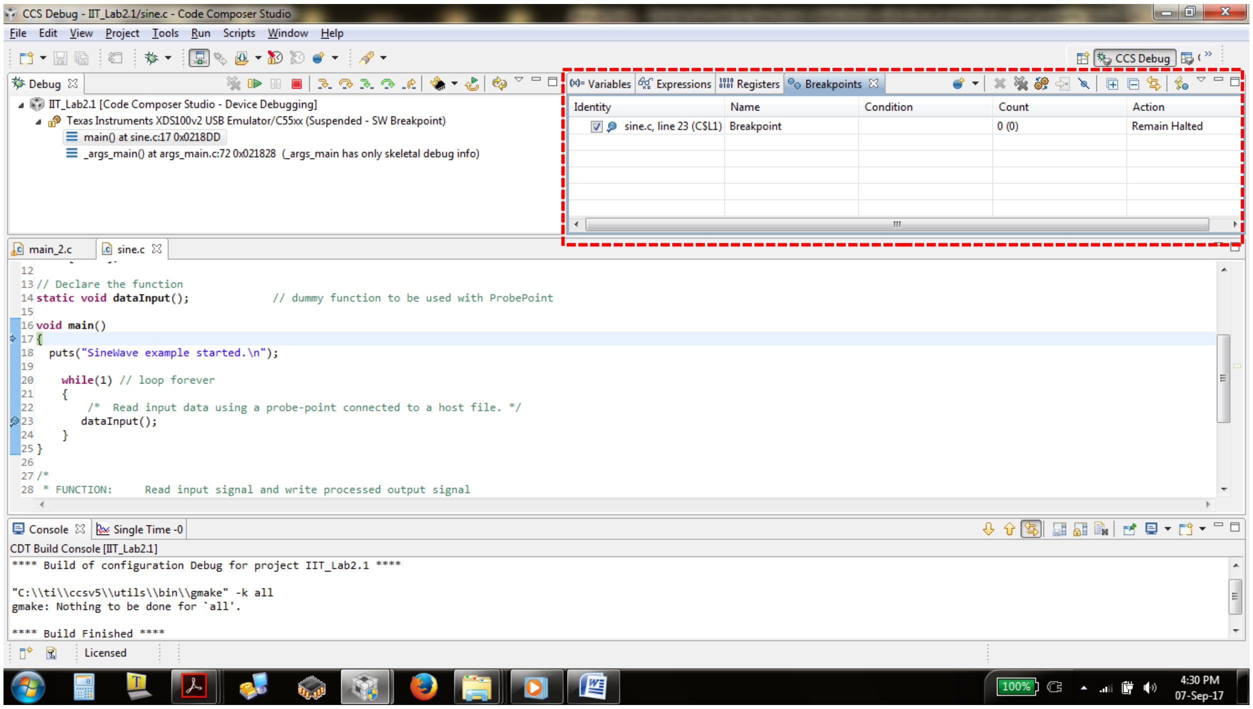
Expression	Type	Value	Address
count	unknown	Error: identifier not found: count	
gain	unknown	Error: identifier not found: gain	
Add new expression			

```
// D
stat                          // dummy function to be used with ProbePoint

void
{
  pu                          n");



                              a probe-point connected to a host file. */


}

/*
 * FUNCTION:     Read input signal and write processed output signal
```

CDT Build Console [IIT_Lab2.1]

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

- It will be seen as shown here.



CCS Debug - IIT_Lab2.1/sine.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Debug
Variables Expressions Registers Breakpoints
CCS Debug
main_2.c sine.c
Console Single Time -0
Licensed
100%
4:30 PM
07-Sep-17

IIT_Lab2.1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Emulator/C55xx (Suspended - SW Breakpoint)
main() at sine.c:17 0x0218DD
_args_main() at args_main.c:72 0x021828 (_args_main has only skeletal debug info)

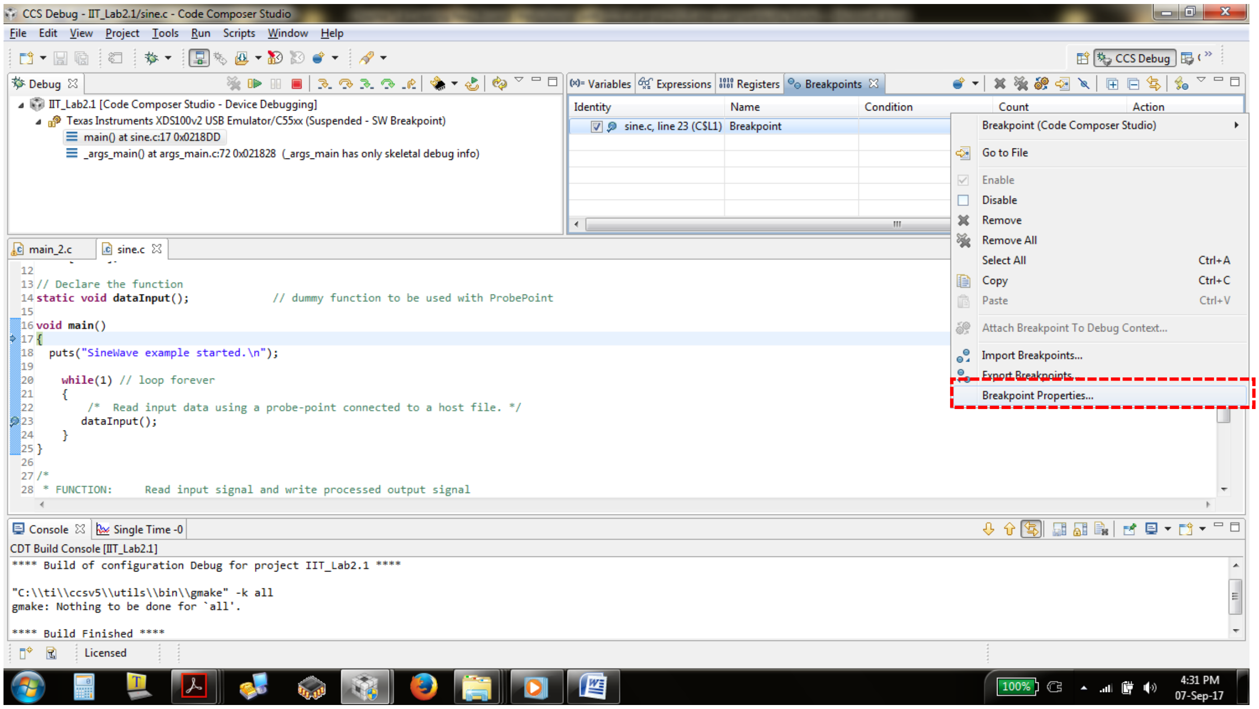
Identity	Name	Condition	Count	Action
☑ sine.c, line 23 (C$L1)	Breakpoint		0 (0)	Remain Halted

```
// Declare the function
static void dataInput();             // dummy function to be used with ProbePoint

void main()
{
  puts("SineWave example started.\n");

    while(1) // loop forever
    {
       /*  Read input data using a probe-point connected to a host file. */
      dataInput();
    }
}

/*
 * FUNCTION:     Read input signal and write processed output signal
```

CDT Build Console [IIT_Lab2.1]

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

- Right click on action column and click “breakpoint properties”



IIT_Lab2.1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Emulator/C55xx (Suspended - SW Breakpoint)
main() at sine.c:17 0x0218DD
_args_main() at args_main.c:72 0x021828 (_args_main has only skeletal debug info)

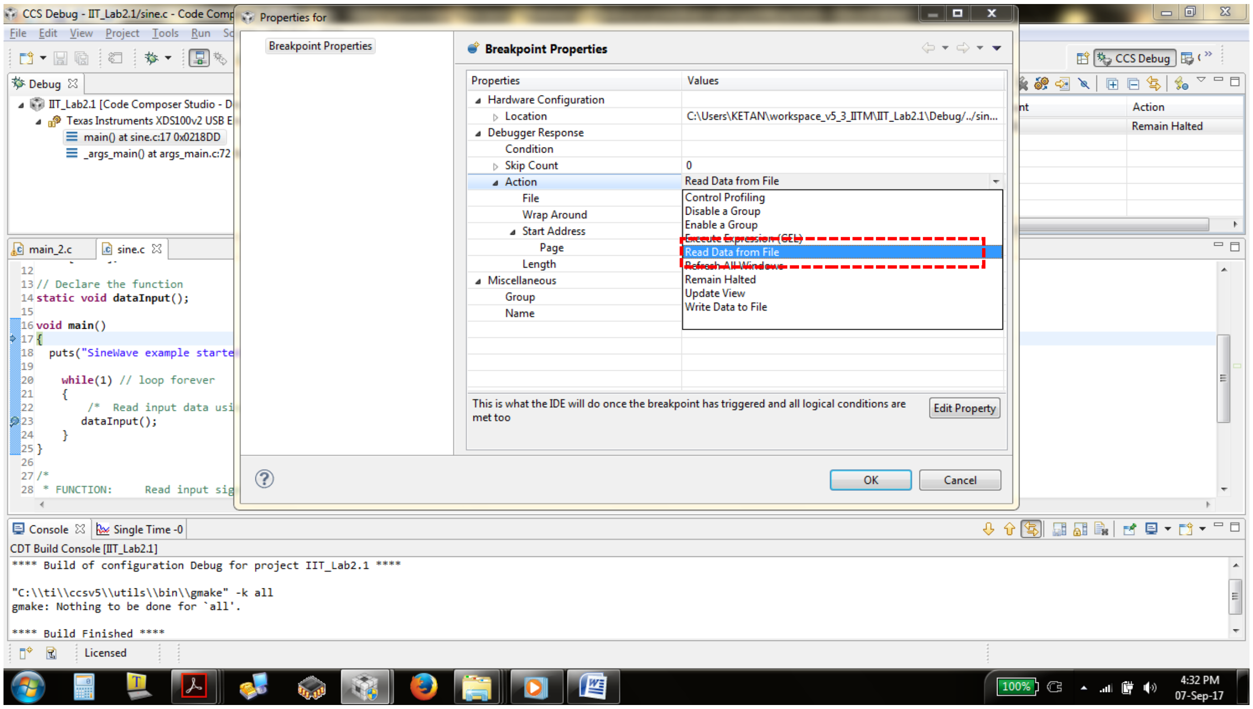
Identity	Name	Condition	Count	Action
☑ sine.c, line 23 (C$L1)	Breakpoint			

```
// Declare the function
static void dataInput();            // dummy function to be used with ProbePoint

void main()
{
  puts("SineWave example started.\n");

    while(1) // loop forever
    {
        /*  Read input data using a probe-point connected to a host file. */
        dataInput();
    }
}

/*
 * FUNCTION:    Read input signal and write processed output signal
```

CDT Build Console [IIT_Lab2.1]

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

- From the “action” tab select “Read Data from File ”



Breakpoint Properties

Properties	Values
Hardware Configuration	
Location	C:\Users\KETAN\workspace_v5_3_IITM\IIT_Lab2.1\Debug/../sin...
Debugger Response	
Condition	
Skip Count	0
Action	Read Data from File
File	Control Profiling
Wrap Around	Disable a Group
Start Address	Enable a Group
Page	Execute Expression (GEL)
Length	Read Data from File
Miscellaneous	Refresh All Windows
Group	Remain Halted
Name	Update View
	Write Data to File

This is what the IDE will do once the breakpoint has triggered and all logical conditions are met too

Action: Remain Halted

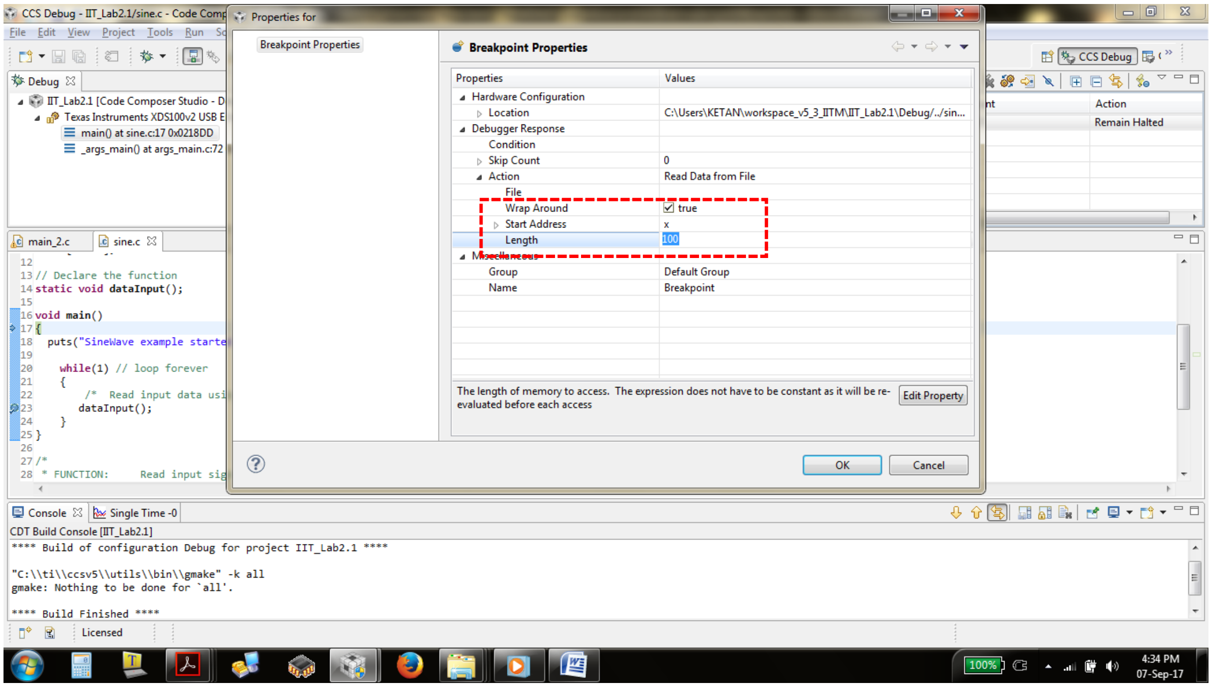
IIT_Lab2.1 [Code Composer Studio - D
Texas Instruments XDS100v2 USB E
main() at sine.c:17 0x0218DD
_args_main() at args_main.c:72

```
// Declare the function
static void dataInput();

void main()
{
  puts("SineWave example starte

    while(1) // loop forever
    {
        /*  Read input data usi
        dataInput();
    }
}

/*
 * FUNCTION:    Read input sig
```

CDT Build Console [IIT_Lab2.1]

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

Select “Wrap Around”, keep start address as our buffer name “x” and Length as 100 as per our program’s buffer size. Settings are as shown here.


CCS Debug - IIT_Lab2.1/sine.c - Code Comp
File Edit View Project Tools Run Sc
Debug
IIT_Lab2.1 [Code Composer Studio - D
Texas Instruments XDS100v2 USB E
main() at sine.c:17 0x0218DD
_args_main() at args_main.c:72
main_2.c
sine.c
Properties for
Breakpoint Properties
CCS Debug
Action
Remain Halted
OK
Cancel
Edit Property
Console
Single Time -0
Licensed
100%
4:34 PM
07-Sep-17

Breakpoint Properties

Properties	Values
Hardware Configuration	
Location	C:\Users\KETAN\workspace_v5_3_IITM\IIT_Lab2.1\Debug/../sin...
Debugger Response	
Condition	
Skip Count	0
Action	Read Data from File
File	
Wrap Around	☑ true
Start Address	x
Length	100
Miscellaneous	
Group	Default Group
Name	Breakpoint

The length of memory to access. The expression does not have to be constant as it will be re-evaluated before each access

```
// Declare the function
static void dataInput();

void main()
{
  puts("SineWave example starte

    while(1) // loop forever
    {
        /*  Read input data usi
        dataInput();
    }
}

/*
 * FUNCTION:     Read input sig
```

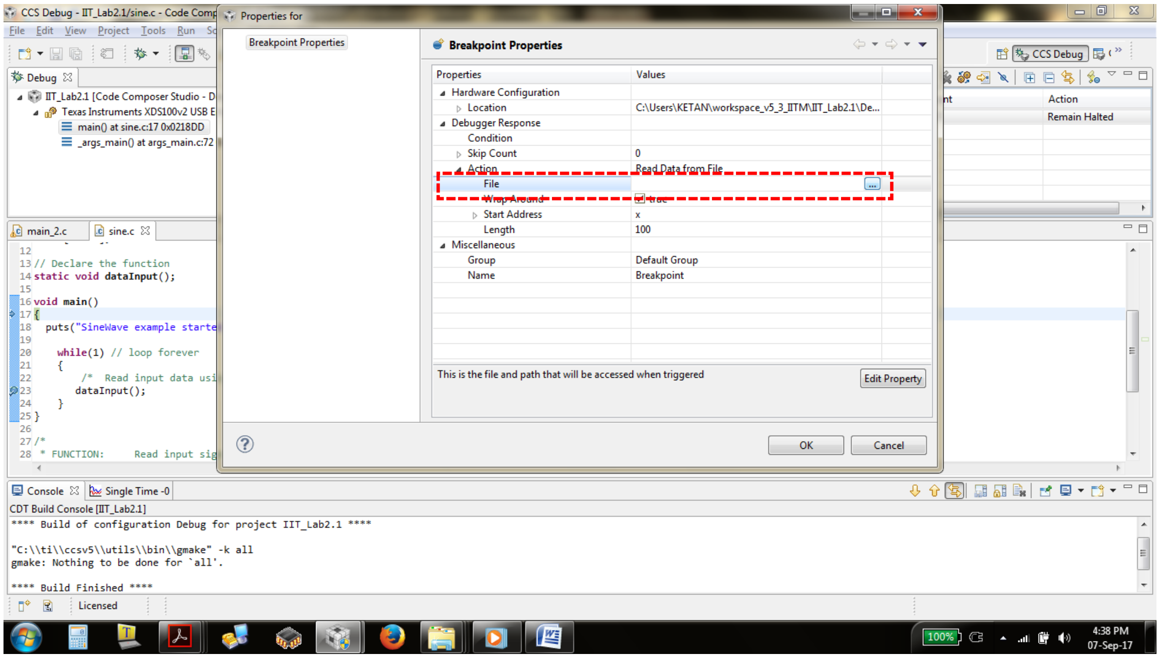
CDT Build Console [IIT_Lab2.1]

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

- The **Address** field specifies where the data from the file is to be placed. The **Length** field specifies how many samples from the data file are read each time the Probe Point is reached.
- In this case **we are taking 100 samples** from the file each time

- The **Wrap Around** option causes the IDE to start reading from the beginning of the file when it reaches the end of the file. This allows the data file to be treated as a continuous stream of data even though it contains only 1000 values and 100 values are read each time the Probe Point is reached

Select the file “sine_int16.dat” for the file location from our project folder “C:\Users\KETAN\workspace_v5_3_IITM\IIT_Lab2.1”.

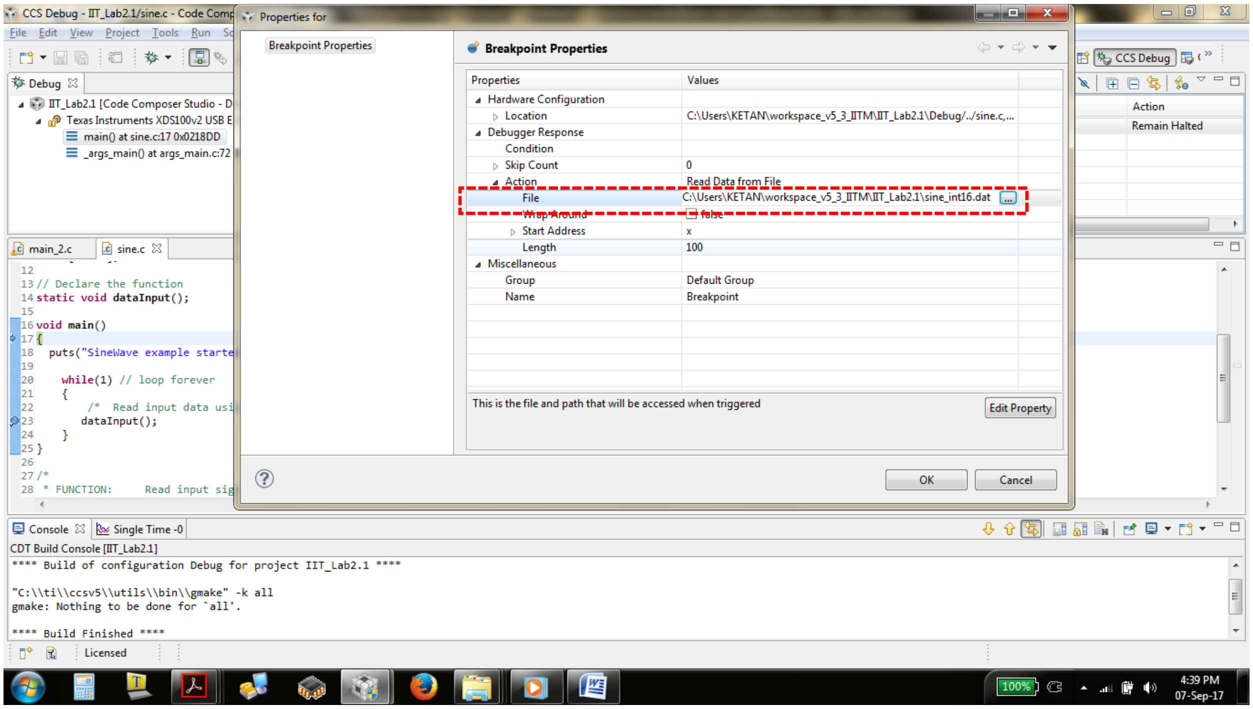


Breakpoint Properties

Properties	Values
Hardware Configuration	
Location	C:\Users\KETAN\workspace_v5_3_IITM\IIT_Lab2.1\De...
Debugger Response	
Condition	
Skip Count	0
Action	Read Data from File
File	
Wrap Around	☑ true
Start Address	x
Length	100
Miscellaneous	
Group	Default Group
Name	Breakpoint

This is the file and path that will be accessed when triggered

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```



Breakpoint Properties

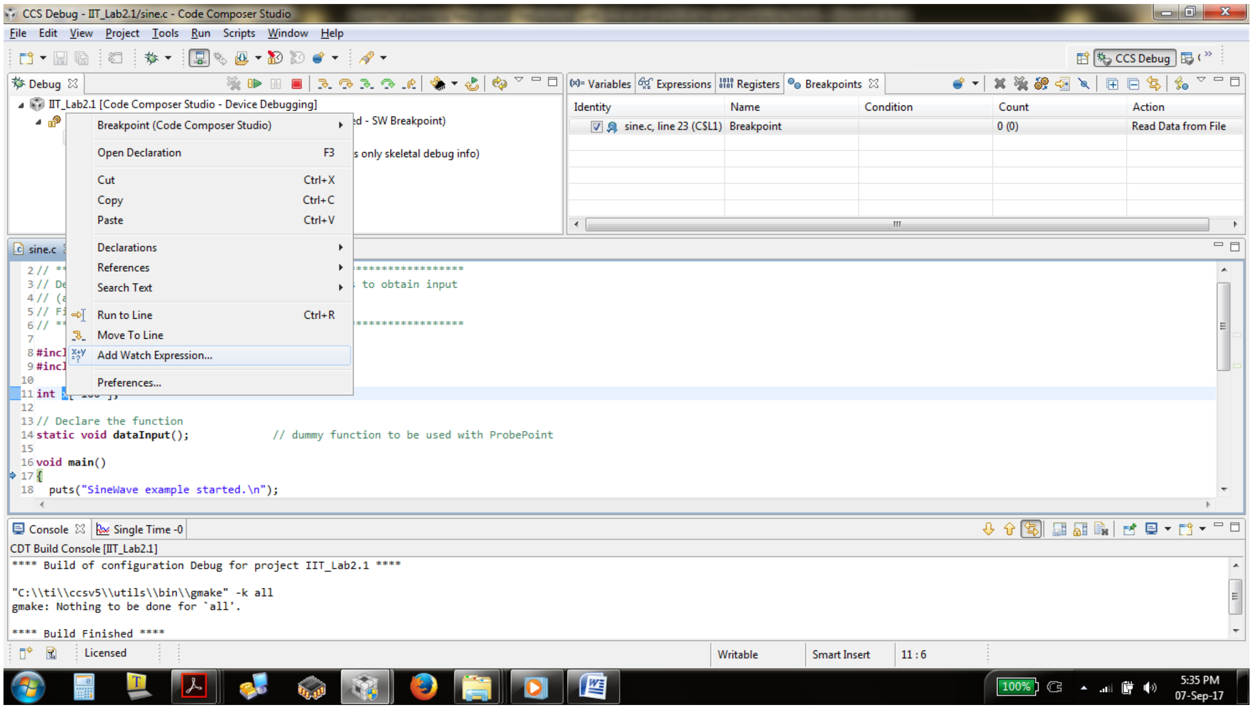
Properties	Values
Hardware Configuration	
Location	C:\Users\KETAN\workspace_v5_3_IITM\IIT_Lab2.1\Debug/../sine.c,...
Debugger Response	
Condition	
Skip Count	0
Action	Read Data from File
File	C:\Users\KETAN\workspace_v5_3_IITM\IIT_Lab2.1\sine_int16.dat
Wrap Around	☐ false
Start Address	x
Length	100
Miscellaneous	
Group	Default Group
Name	Breakpoint

This is the file and path that will be accessed when triggered

And click OK.

Time to Run/Execute the project:

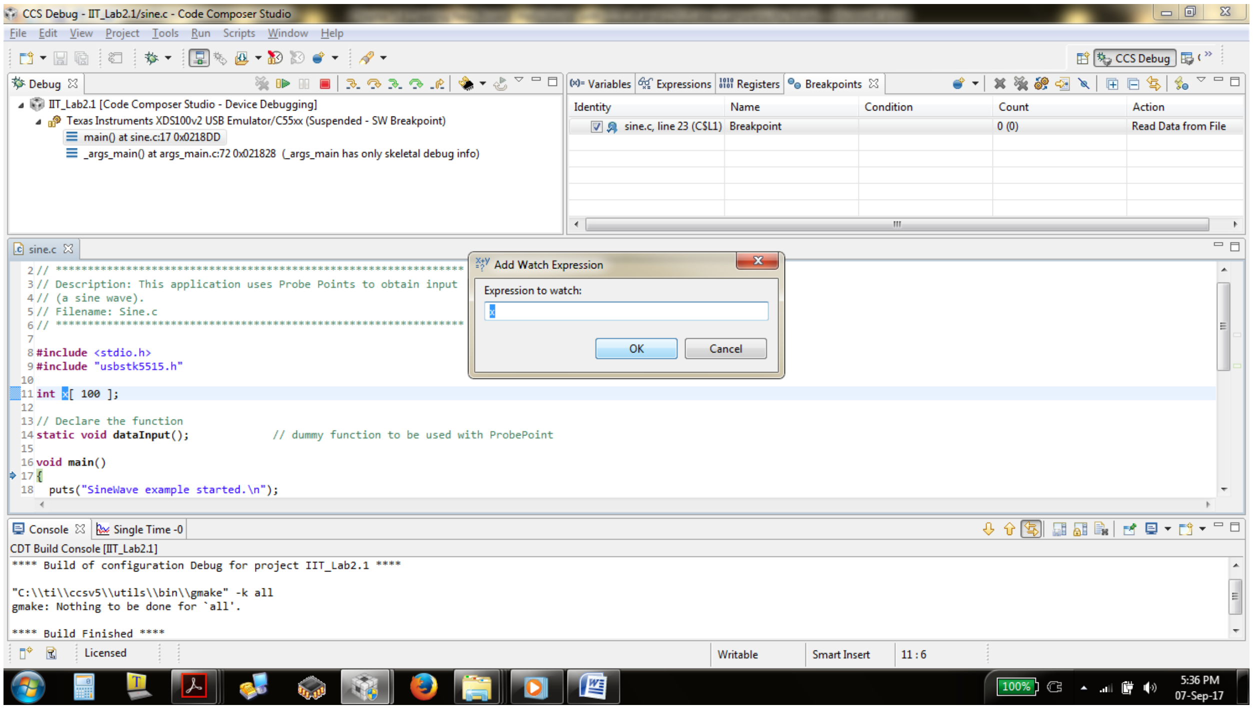
- Add x[100] variable to the watch window by right clicking on “x” variable and click “Add watch Expression”



Identity	Name	Condition	Count	Action
☑ sine.c, line 23 (C$L1)	Breakpoint		0 (0)	Read Data from File

```
// **                ******************
// De                  to obtain input
// (a
// Fi
// **                ******************

#incl
#incl

int x[ 100 ];

// Declare the function
static void dataInput();             // dummy function to be used with ProbePoint

void main()
{
  puts("SineWave example started.\n");
```

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

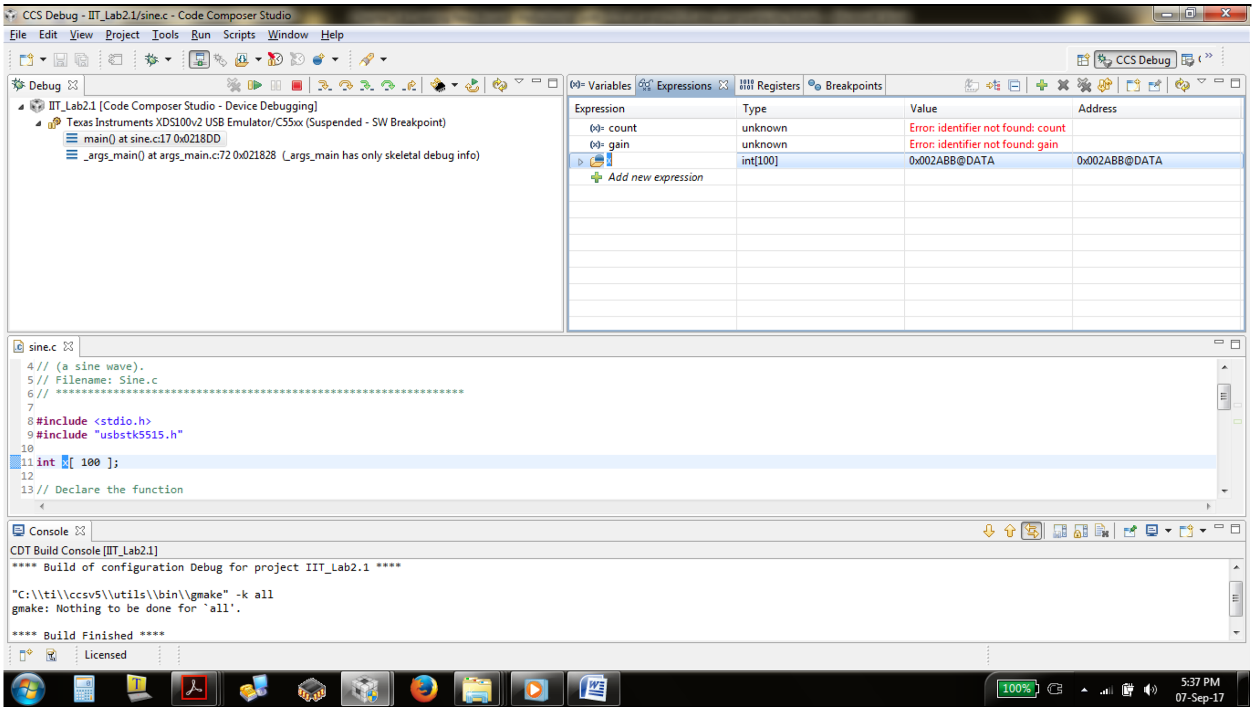

CCS Debug - IIT_Lab2.1/sine.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
CCS Debug
Debug
IIT_Lab2.1 [Code Composer Studio - Device Debugging]
Texas Instruments XDS100v2 USB Emulator/C55xx (Suspended - SW Breakpoint)
main() at sine.c:17 0x0218DD
_args_main() at args_main.c:72 0x021828 (_args_main has only skeletal debug info)
Variables
Expressions
Registers
Breakpoints
sine.c
Add Watch Expression
OK
Cancel
Console
Single Time -0
CDT Build Console [IIT_Lab2.1]
Licensed
Writable
Smart Insert
11 : 6
100%
5:36 PM
07-Sep-17

Identity	Name	Condition	Count	Action
☑ sine.c, line 23 (C$L1)	Breakpoint		0 (0)	Read Data from File

Expression to watch: x

```
// ******************************************************************
// Description: This application uses Probe Points to obtain input
// (a sine wave).
// Filename: Sine.c
// ******************************************************************

#include <stdio.h>
#include "usbstk5515.h"

int x[ 100 ];

// Declare the function
static void dataInput();             // dummy function to be used with ProbePoint

void main()
{
  puts("SineWave example started.\n");
```

```
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```


CCS Debug - IIT_Lab2.1/sine.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
CCS Debug
Debug
Variables Expressions Registers Breakpoints
sine.c
Console
Licensed
100%
5:37 PM
07-Sep-17

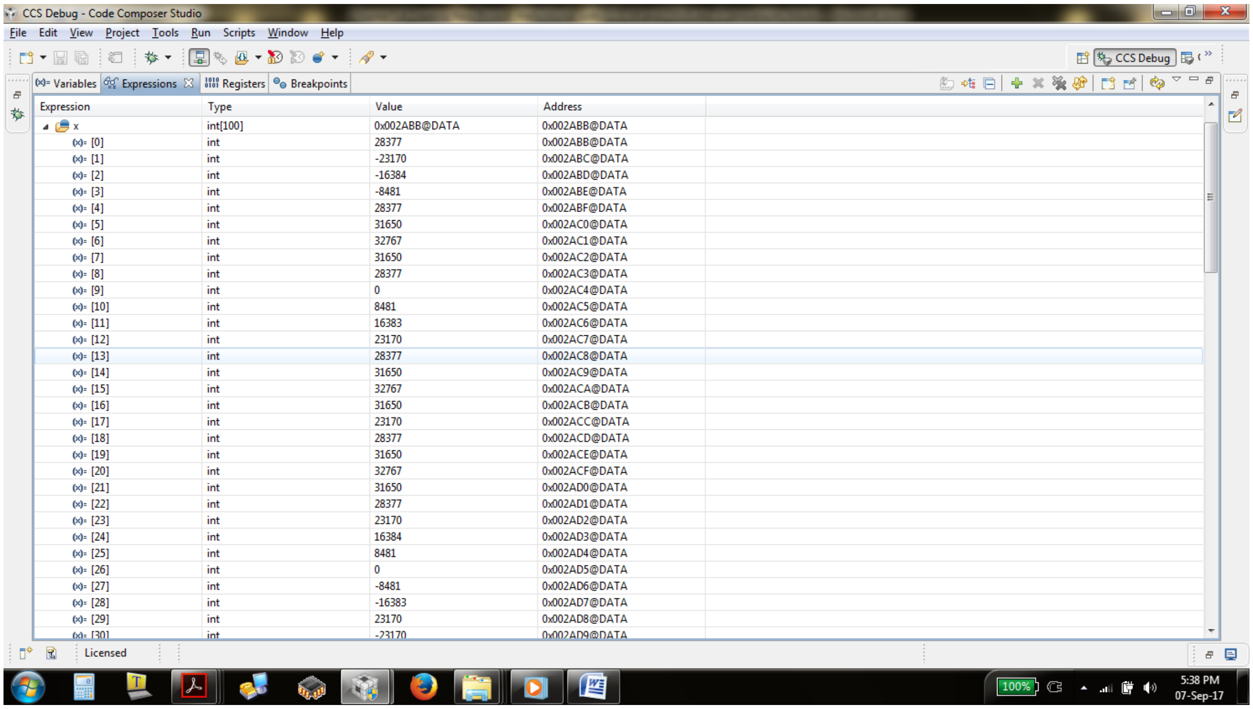
```
IIT_Lab2.1 [Code Composer Studio - Device Debugging]
  Texas Instruments XDS100v2 USB Emulator/C55xx (Suspended - SW Breakpoint)
    main() at sine.c:17 0x0218DD
    _args_main() at args_main.c:72 0x021828  (_args_main has only skeletal debug info)
```

Expression	Type	Value	Address
count	unknown	Error: identifier not found: count	
gain	unknown	Error: identifier not found: gain	
x	int[100]	0x002ABB@DATA	0x002ABB@DATA
Add new expression			

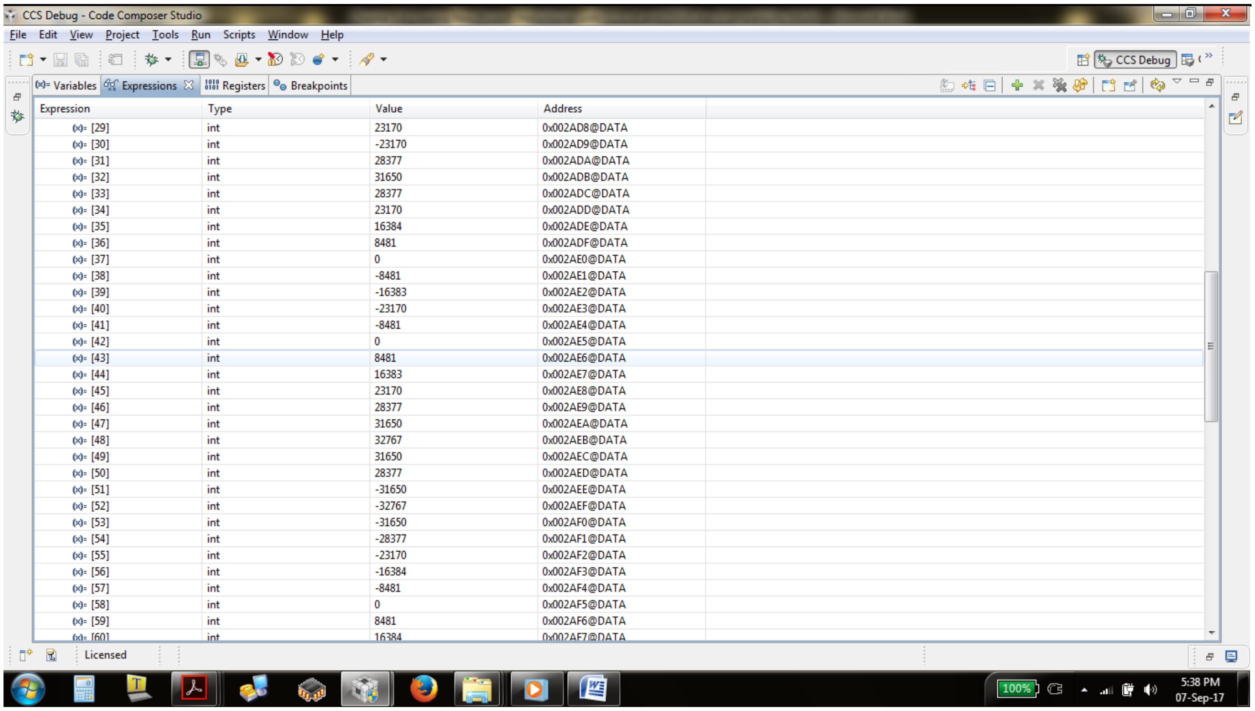
```
// (a sine wave).
// Filename: Sine.c
// ******************************************************************

#include <stdio.h>
#include "usbstk5515.h"

int x[ 100 ];

// Declare the function
```

```
CDT Build Console [IIT_Lab2.1]
**** Build of configuration Debug for project IIT_Lab2.1 ****

"C:\\ti\\ccsv5\\utils\\bin\\gmake" -k all
gmake: Nothing to be done for `all'.

**** Build Finished ****
```

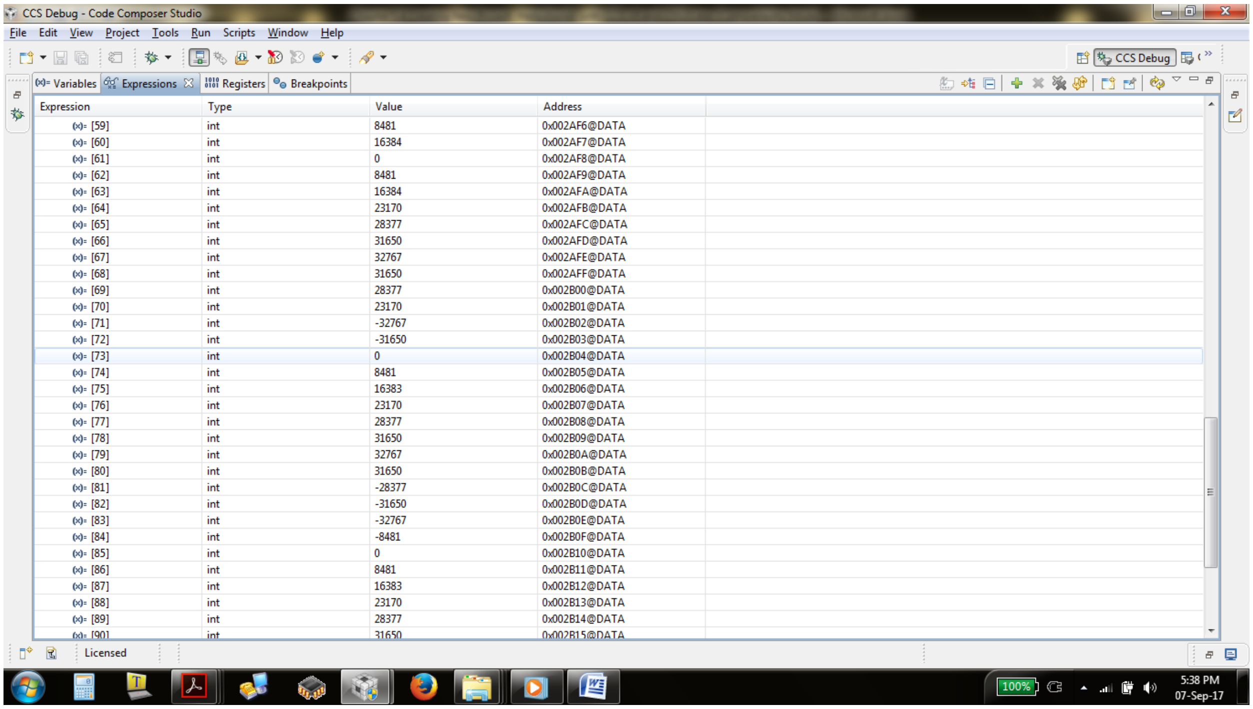
- Now run/Execute/Resume program by clicking on “resume” icon.
- When you run the program, the data from sine int16.dat will be read into the array x. and check “x” variable in watch window. And expand it.


CCS Debug - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
CCS Debug
Variables Expressions Registers Breakpoints
Licensed
100%
5:38 PM
07-Sep-17

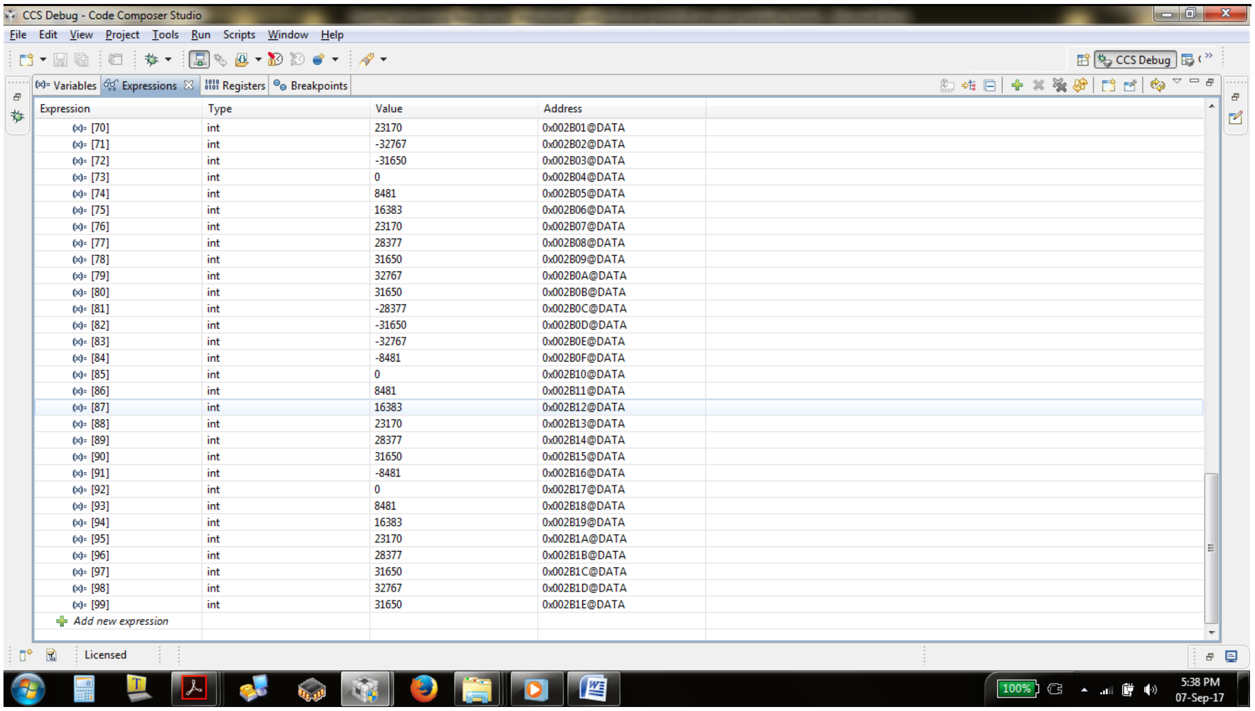
Expression	Type	Value	Address
x	int[100]	0x002ABB@DATA	0x002ABB@DATA
[0]	int	28377	0x002ABB@DATA
[1]	int	-23170	0x002ABC@DATA
[2]	int	-16384	0x002ABD@DATA
[3]	int	-8481	0x002ABE@DATA
[4]	int	28377	0x002ABF@DATA
[5]	int	31650	0x002AC0@DATA
[6]	int	32767	0x002AC1@DATA
[7]	int	31650	0x002AC2@DATA
[8]	int	28377	0x002AC3@DATA
[9]	int	0	0x002AC4@DATA
[10]	int	8481	0x002AC5@DATA
[11]	int	16383	0x002AC6@DATA
[12]	int	23170	0x002AC7@DATA
[13]	int	28377	0x002AC8@DATA
[14]	int	31650	0x002AC9@DATA
[15]	int	32767	0x002ACA@DATA
[16]	int	31650	0x002ACB@DATA
[17]	int	23170	0x002ACC@DATA
[18]	int	28377	0x002ACD@DATA
[19]	int	31650	0x002ACE@DATA
[20]	int	32767	0x002ACF@DATA
[21]	int	31650	0x002AD0@DATA
[22]	int	28377	0x002AD1@DATA
[23]	int	23170	0x002AD2@DATA
[24]	int	16384	0x002AD3@DATA
[25]	int	8481	0x002AD4@DATA
[26]	int	0	0x002AD5@DATA
[27]	int	-8481	0x002AD6@DATA
[28]	int	-16383	0x002AD7@DATA
[29]	int	23170	0x002AD8@DATA
[30]	int	-23170	0x002AD9@DATA


CCS Debug - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
CCS Debug
Variables Expressions Registers Breakpoints
Licensed
100%
5:38 PM
07-Sep-17

Expression	Type	Value	Address
[29]	int	23170	0x002AD8@DATA
[30]	int	-23170	0x002AD9@DATA
[31]	int	28377	0x002ADA@DATA
[32]	int	31650	0x002ADB@DATA
[33]	int	28377	0x002ADC@DATA
[34]	int	23170	0x002ADD@DATA
[35]	int	16384	0x002ADE@DATA
[36]	int	8481	0x002ADF@DATA
[37]	int	0	0x002AE0@DATA
[38]	int	-8481	0x002AE1@DATA
[39]	int	-16383	0x002AE2@DATA
[40]	int	-23170	0x002AE3@DATA
[41]	int	-8481	0x002AE4@DATA
[42]	int	0	0x002AE5@DATA
[43]	int	8481	0x002AE6@DATA
[44]	int	16383	0x002AE7@DATA
[45]	int	23170	0x002AE8@DATA
[46]	int	28377	0x002AE9@DATA
[47]	int	31650	0x002AEA@DATA
[48]	int	32767	0x002AEB@DATA
[49]	int	31650	0x002AEC@DATA
[50]	int	28377	0x002AED@DATA
[51]	int	-31650	0x002AEE@DATA
[52]	int	-32767	0x002AEF@DATA
[53]	int	-31650	0x002AF0@DATA
[54]	int	-28377	0x002AF1@DATA
[55]	int	-23170	0x002AF2@DATA
[56]	int	-16384	0x002AF3@DATA
[57]	int	-8481	0x002AF4@DATA
[58]	int	0	0x002AF5@DATA
[59]	int	8481	0x002AF6@DATA
[60]	int	16384	0x002AF7@DATA


CCS Debug - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
CCS Debug
Variables Expressions Registers Breakpoints
Licensed
100%
5:38 PM
07-Sep-17

Expression	Type	Value	Address
[59]	int	8481	0x002AF6@DATA
[60]	int	16384	0x002AF7@DATA
[61]	int	0	0x002AF8@DATA
[62]	int	8481	0x002AF9@DATA
[63]	int	16384	0x002AFA@DATA
[64]	int	23170	0x002AFB@DATA
[65]	int	28377	0x002AFC@DATA
[66]	int	31650	0x002AFD@DATA
[67]	int	32767	0x002AFE@DATA
[68]	int	31650	0x002AFF@DATA
[69]	int	28377	0x002B00@DATA
[70]	int	23170	0x002B01@DATA
[71]	int	-32767	0x002B02@DATA
[72]	int	-31650	0x002B03@DATA
[73]	int	0	0x002B04@DATA
[74]	int	8481	0x002B05@DATA
[75]	int	16383	0x002B06@DATA
[76]	int	23170	0x002B07@DATA
[77]	int	28377	0x002B08@DATA
[78]	int	31650	0x002B09@DATA
[79]	int	32767	0x002B0A@DATA
[80]	int	31650	0x002B0B@DATA
[81]	int	-28377	0x002B0C@DATA
[82]	int	-31650	0x002B0D@DATA
[83]	int	-32767	0x002B0E@DATA
[84]	int	-8481	0x002B0F@DATA
[85]	int	0	0x002B10@DATA
[86]	int	8481	0x002B11@DATA
[87]	int	16383	0x002B12@DATA
[88]	int	23170	0x002B13@DATA
[89]	int	28377	0x002B14@DATA
[90]	int	31650	0x002B15@DATA



CCS Debug - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
CCS Debug
(x)= Variables
Expressions
Registers
Breakpoints
Licensed
100%
5:38 PM
07-Sep-17

Expression	Type	Value	Address
(x)= [70]	int	23170	0x002B01@DATA
(x)= [71]	int	-32767	0x002B02@DATA
(x)= [72]	int	-31650	0x002B03@DATA
(x)= [73]	int	0	0x002B04@DATA
(x)= [74]	int	8481	0x002B05@DATA
(x)= [75]	int	16383	0x002B06@DATA
(x)= [76]	int	23170	0x002B07@DATA
(x)= [77]	int	28377	0x002B08@DATA
(x)= [78]	int	31650	0x002B09@DATA
(x)= [79]	int	32767	0x002B0A@DATA
(x)= [80]	int	31650	0x002B0B@DATA
(x)= [81]	int	-28377	0x002B0C@DATA
(x)= [82]	int	-31650	0x002B0D@DATA
(x)= [83]	int	-32767	0x002B0E@DATA
(x)= [84]	int	-8481	0x002B0F@DATA
(x)= [85]	int	0	0x002B10@DATA
(x)= [86]	int	8481	0x002B11@DATA
(x)= [87]	int	16383	0x002B12@DATA
(x)= [88]	int	23170	0x002B13@DATA
(x)= [89]	int	28377	0x002B14@DATA
(x)= [90]	int	31650	0x002B15@DATA
(x)= [91]	int	-8481	0x002B16@DATA
(x)= [92]	int	0	0x002B17@DATA
(x)= [93]	int	8481	0x002B18@DATA
(x)= [94]	int	16383	0x002B19@DATA
(x)= [95]	int	23170	0x002B1A@DATA
(x)= [96]	int	28377	0x002B1B@DATA
(x)= [97]	int	31650	0x002B1C@DATA
(x)= [98]	int	32767	0x002B1D@DATA
(x)= [99]	int	31650	0x002B1E@DATA
Add new expression			

- **Look at the “x” variable’s data carefully**. It’s sine wave’s digital manipulation and that is read from the **sine_int16.dat** file

- To stop the project use Halt/Terminate as shown here in dotted red colored highlight and it stops the running program and closes the active project.

Enjoy…!