**Date Submitted: 10/23/2019**

**Task 00: Execute provided code**

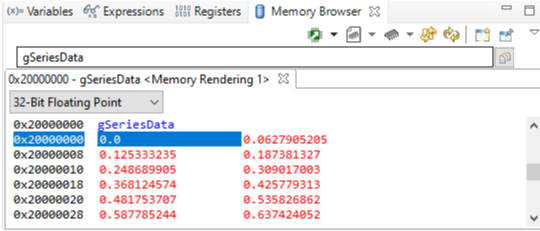
**Youtube Link: n/a**

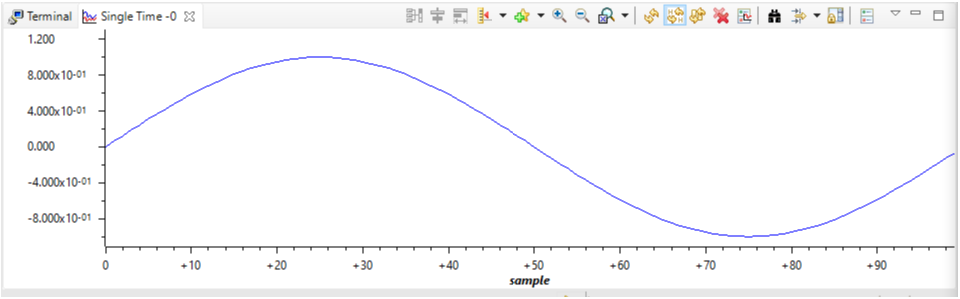
**------------------------------------------------------------------------------------**

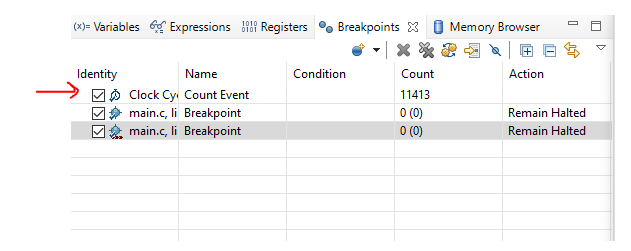
**Task 01:**

Youtube Link: n/a

**Modified Schematic (if applicable):**







**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> // included to use sinf() function

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/fpu.h" // support for Floating Point Unit

**#include** "driverlib/sysctl.h"

**#include** "driverlib/rom.h"

// Define pi value (3.14....)

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

// Define the length (size) of array containing floating point values

**#define** SERIES\_LENGTH 100

**float** gSeriesData[SERIES\_LENGTH]; // array of floating point values

int32\_t i32DataCount = 0; // index/counter of floating point array (gSeriesData[])

**int** **main**(**void**)

{

**float** fRadians; // variable used to store and calculate the sine value

ROM\_FPULazyStackingEnable(); // turn on Lazy Stacking

ROM\_FPUEnable(); // turn on FPU (it is off by default)

// set up system clock for 50MHz

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // a full sine wave cycle is 2pi radians.

// Divide 2pi by the depth of the array (SERIES\_LENGTH)

// using the sine function (sinf()) to calculate the sine values for each

// of the 100 values of the angle and place them into the array

**while**(i32DataCount < SERIES\_LENGTH)

{

// calculate, using the sine function, and store sine values into the array

gSeriesData[i32DataCount] = **sinf**(fRadians \* i32DataCount);

i32DataCount++; // increment array index

}

// loop forever..

**while**(1)

{

}

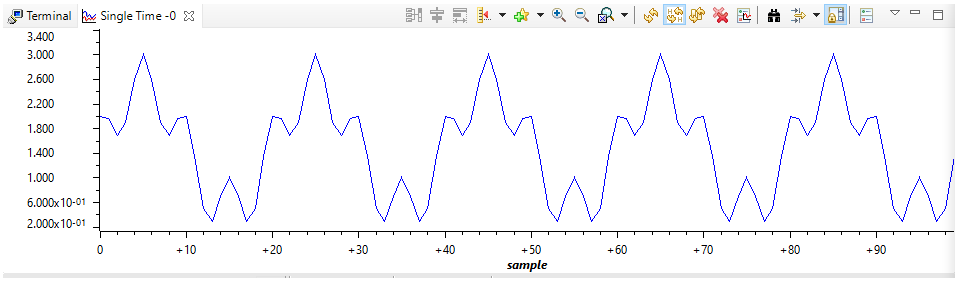
}

**------------------------------------------------------------------------------------**

**Task 02:**

Youtube Link: n/a

**Modified Schematic (if applicable):**



**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> // included to use sinf() & cosf() functions

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/fpu.h" // support for Floating Point Unit

**#include** "driverlib/sysctl.h"

**#include** "driverlib/rom.h"

// Define pi value (3.14....)

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

// Define the length (size) of array containing floating point values

**#define** SERIES\_LENGTH 1000

**float** gSeriesData[SERIES\_LENGTH]; // array of floating point values

int32\_t i32DataCount = 0; // index/counter of floating point array (gSeriesData[])

**int** **main**(**void**)

{

**float** fRadians; // variable used to store and calculate the sine value

ROM\_FPULazyStackingEnable(); // turn on Lazy Stacking

ROM\_FPUEnable(); // turn on FPU (it is off by default)

// set up system clock for 50MHz

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // a full sine wave cycle is 2pi radians.

// Divide 2pi by the depth of the array (SERIES\_LENGTH)

// using the sine function (sinf()) and cosine function (cosf()) to calculate the equation

**while**(i32DataCount < SERIES\_LENGTH)

{

gSeriesData[i32DataCount] = 1.5 + **sinf**(fRadians \* (50\*i32DataCount)) + 0.5\***cosf**(fRadians\*(200\*i32DataCount));

i32DataCount++; // increment array index

}

// loop forever..

**while**(1)

{

}

}

**------------------------------------------------------------------------------------**