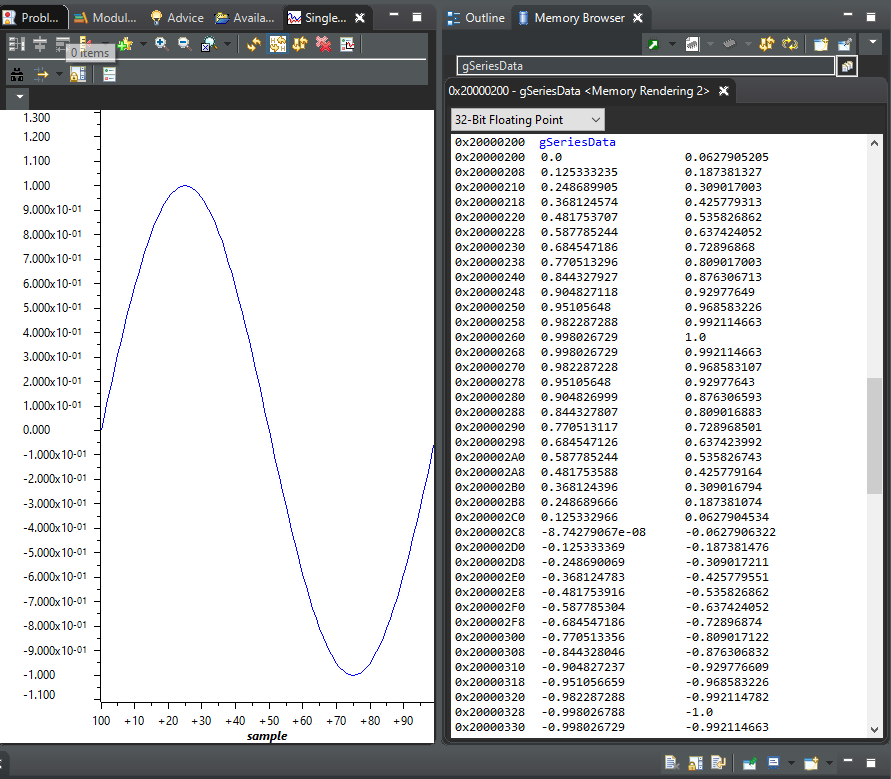
**Date Submitted: 11/4/19**

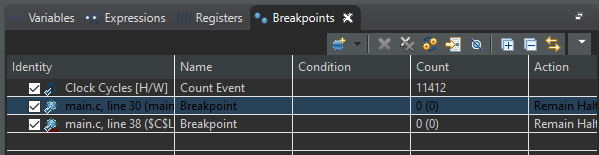
**Task 01:**

Youtube Link:

<https://www.youtube.com/watch?v=ioUYoaFctrU>

**Modified Schematic (if applicable):**





**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> //code uses sinf() function which is in this header file

**#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"

//Just in case the M\_PI is not already defined this will do it for us

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

**#define** SERIES\_LENGTH 100 //depth of our data buffer

**float** gSeriesData[SERIES\_LENGTH]; //an array of floats SERIES\_LENGTH long (100)

int32\_t i32DataCount = 0; //counter for our computation loop

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

{

**float** fRadians; //need a variable of type float to calculate sine

ROM\_FPULazyStackingEnable();//turn on lazy stacking

ROM\_FPUEnable();//Turn on the FPU

//Set the system clock to 50MHz

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

//a full sine wave cycle is 2pi radians. Divide 2pi by the depth of the array fRadians = ((2\*M\_PI)/SERIES\_LENGTH);

//calculate the sine value for each of the 100 values of the angle and place them in our data array

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

{

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

}

//endless loop **while**(1)

{

}

}

**------------------------------------------------------------------------------------ Task 02:**

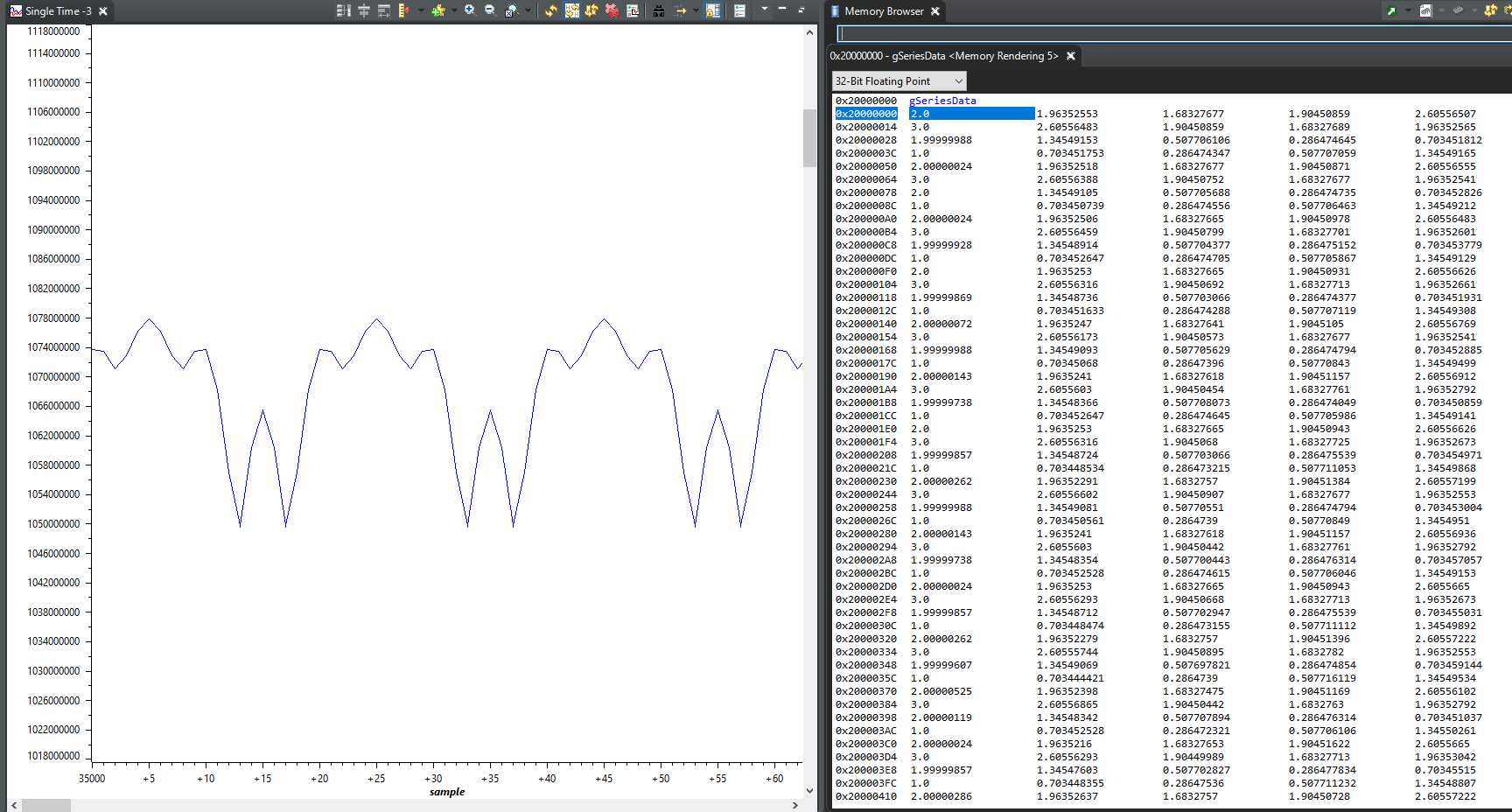
I had to implement the equation below in my code:

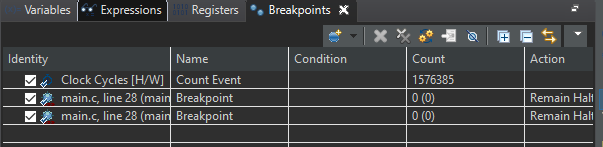


Youtube Link:

<https://www.youtube.com/watch?v=g7CfzVLjm0s>

**Modified Schematic (if applicable):**





**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h>

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

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

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

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

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

**#ifndef** M\_PI

**#define** **M\_PI** 3.14159265358979323846

**#endif**

**#define** **SERIES\_LENGTH** 1000 //size of buff array

**float** gSeriesData[SERIES\_LENGTH]; //used to store the array of floats (1000)

**int32\_t** i32DataCount = 0;

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

**float** fRad\_mid\_part, fRad\_end\_part; //used to calculate the intermediate vars 0f the equation

ROM\_FPULazyStackingEnable();//enable lazy stacking

ROM\_FPUEnable();// enable the FPU

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

fRad\_mid\_part = (2\*M\_PI\*50)/SERIES\_LENGTH; // used to model the equations that

fRad\_end\_part = (2\*M\_PI\*200)/SERIES\_LENGTH; // generates the 50Hz

// here we calculate the values of the wave

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

gSeriesData[i32DataCount] = 1.5+(1.0\***sinf**(fRad\_mid\_part \* i32DataCount)+0.5\***cosf**(fRad\_end\_part \* i32DataCount));

i32DataCount++;

}

//endless loop

**while**(1)

{

}

}

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