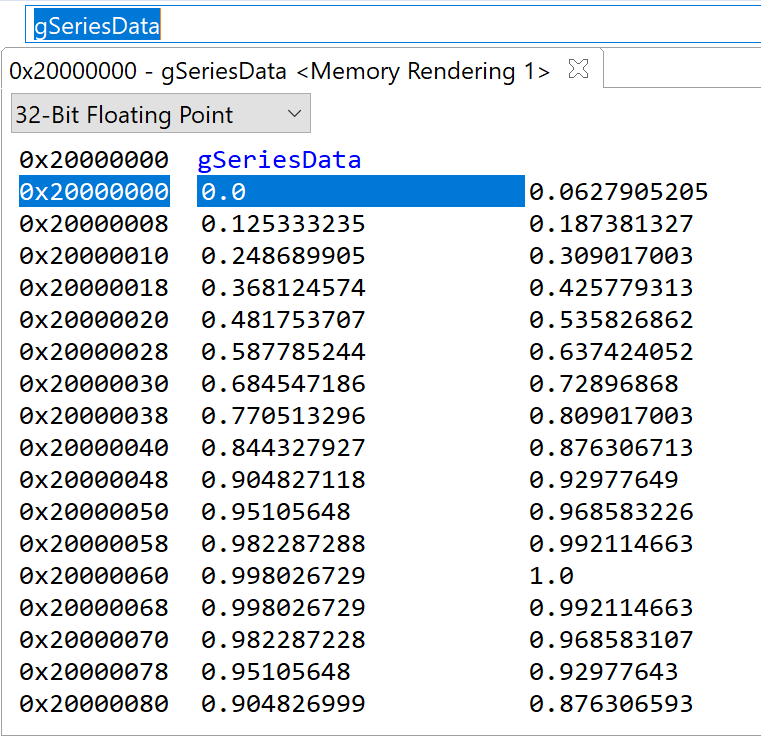
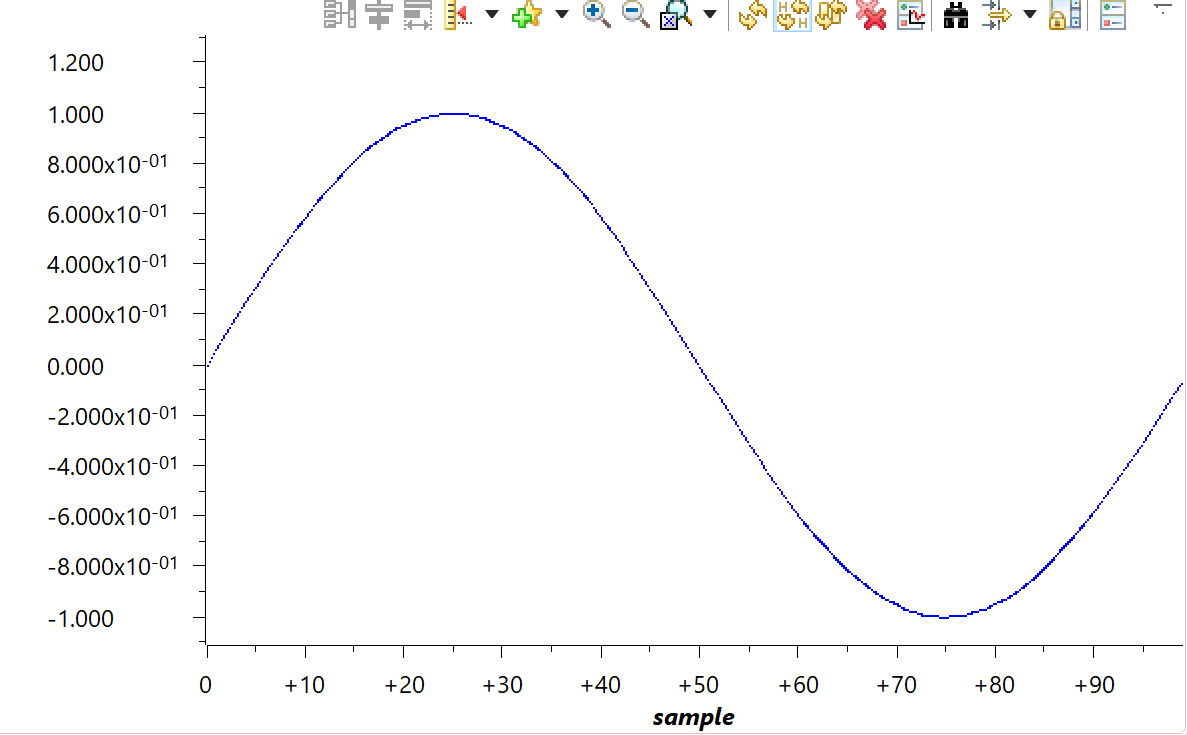
**Date Submitted: 10/17/2018 11:37 PM**

**Task 01: Submit a comprehensive commented file of the original code.**

**Youtube Link:**



**Modified Code:**

**#define** PART\_TM4C123GH6PM //define the board

**#define** TARGET\_IS\_TM4C123\_RB1 //define the target

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h>

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

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

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

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

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

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

**#include** "inc/tm4c123gh6pm.h"

**#ifndef** M\_PI //create PI

**#define** M\_PI 3.14159265358979323846 //define the value for PI

**#endif**

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

**float** gSeriesData[SERIES\_LENGTH]; //an array of floats

int32\_t i32DataCount = 0; //counter

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

**float** fRadians; //float for calculation purposes

ROM\_FPULazyStackingEnable(); //enable Lazy Stacking, skips the stacking of floating point registers to avoid latency

ROM\_FPUEnable();//enable FPU (Floating Point Unit)

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //enable Clock to 50MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); //divide 2PI by the array size

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

gSeriesData[i32DataCount] = **sinf**(fRadians \* i32DataCount); //calaculate sine wave value for each 100

i32DataCount++; }

**while**(1) {

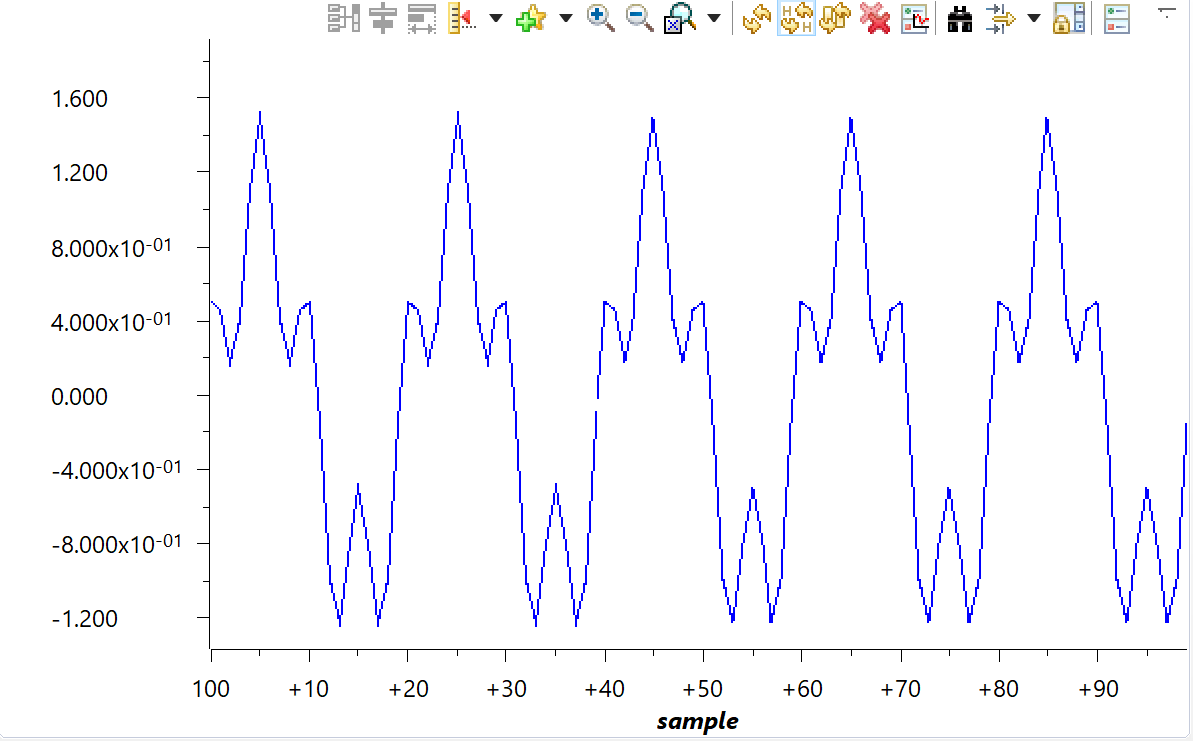
}

}

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

**Task 02:** Modify the code to implement the below equation to generate a frequency of 5 Hz. Display the equation for for 1 sec.

Youtube Link: <https://www.youtube.com/watch?v=H4bBfv8rbBA>



**Modified Code:**

**#define** PART\_TM4C123GH6PM //define the board

**#define** TARGET\_IS\_TM4C123\_RB1 //define the target

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h>

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

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

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

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

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

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

**#include** "inc/tm4c123gh6pm.h"

**#ifndef** M\_PI //create PI

**#define** M\_PI 3.14159265358979323846 //define the value for PI

**#endif**

**#define** SERIES\_LENGTH 1000 //depth of data buffer

**float** gSeriesData[SERIES\_LENGTH]; //an array of floats

int32\_t i32DataCount = 0; //counter

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

**float** fRadians; //float for calculation purposes

ROM\_FPULazyStackingEnable(); //enable Lazy Stacking, skips the stacking of floating point registers to avoid latency

ROM\_FPUEnable();//enable FPU (Floating Point Unit)

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //enable Clock to 50MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); //divide 2PI by the array size

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

gSeriesData[i32DataCount] = ((**sinf**(fRadians\* 50 \* i32DataCount)) + (0.5\*(**cosf**( fRadians \*200\* i32DataCount)))); //calaculate sine wave value for each 100

i32DataCount++; }

**while**(1) {

}

}