**Date Submitted: 10/23/19**

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

**Task 01:**

Comprehensive commented file of the of the original code below

**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#define** TARGET\_IS\_BLIZZARD\_RB1

**#include** <math.h> //the code uses the sinf() function prototyped by 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"

**#ifndef** M\_PI //just in case M\_PI is not already defined, this code will do so

**#define** M\_PI 3.14159265358979323846

**#endif**

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

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

int32\_t i32DataCount = 0; //a counter

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

{

**float** fRadians; //float variable used to calculate sine

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

ROM\_FPUEnable(); //turn on FPU

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

//set up system clock 50 MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH);

//full sine wave cycle is 2pi radians...divide 2pi by the depth of the array

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

{

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

//calculate sine value for each of the 100 values of the angle and place them

//in the data array

i32DataCount++; //increment

}

**while**(1) //infinite loop

{

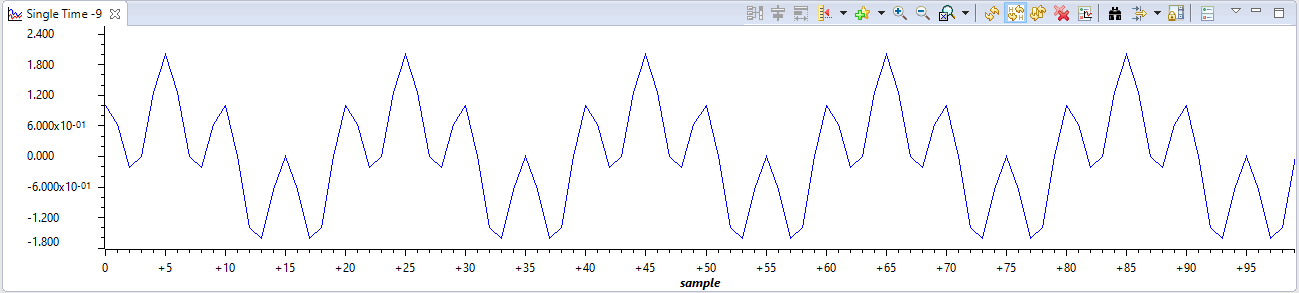
}

}

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

**Task 02:**

Screen Capture of equation wave:

****

**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#define** TARGET\_IS\_BLIZZARD\_RB1

**#include** <math.h> //the code uses the sinf() function prototyped by 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"

**#ifndef** M\_PI //just in case M\_PI is not already defined, this code will do so

**#define** M\_PI 3.14159265358979323846

**#endif**

**#define** SERIES\_LENGTH 1000 //this is the depth of our data buffer

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

int32\_t i32DataCount = 0; //a counter

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

{

**float** fRadians; //float variable used to calculate sine

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

ROM\_FPUEnable(); //turn on FPU

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

//set up system clock 50 MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH);

//full sine wave cycle is 2pi radians...divide 2pi by the depth of the array

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

{

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

//sin (2π⁡\*⁡50t)⁡+⁡0.5⁡\*⁡cos(2π⁡\*⁡200t)

//generates frequency 5Hz

//calculate sine value for each of the 100 values of the angle and place them

//in the data array

i32DataCount++; //increment

}

**while**(1) //infinite loop

{

}

}

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