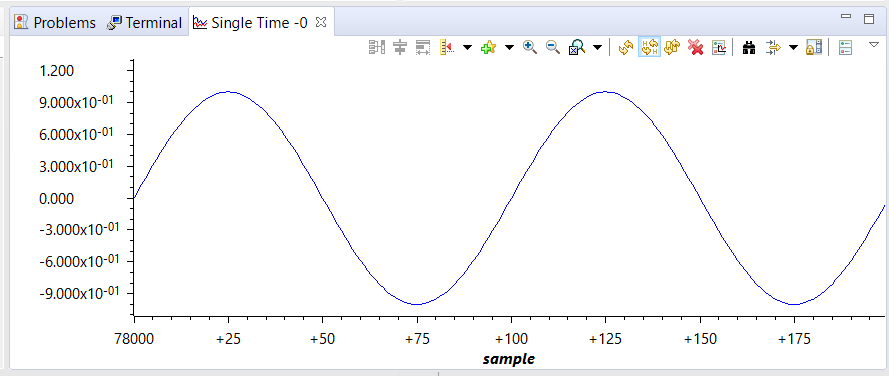
**Task 1:**

Submit a comprehensive commented file of the original code.

Two full oscillations of the function are shown below:



**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> // lib for math functions

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

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

**#include** "driverlib/fpu.h" // floating point lib

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

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

// define the constant for PI as a floating point number

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

// depth of the data buffer for samples

**#define** SERIES\_LENGTH 100

// floating point array that holds the series data points

**float** gSeriesData[SERIES\_LENGTH];

// the counter for computation

int32\_t i32DataCount = 0;

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

{

// radian variable

**float** fRadians;

// enable FP data onto stack, space is reserved

**FPULazyStackingEnable**();

// floating point unit is enabled

**FPUEnable**();

// ==================================================

// Configure system clock

// CLK = 50MHz = (400MHz PLL / (4 \* 2))

// --------------------------------------------------

**SysCtlClockSet**(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

// define radian with respect to the series length

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

// continue to compute while counter is under depth size

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

{

// sin function of given radian value

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

// increment counter by 1

i32DataCount++;

}

// loop here forever

**while**(1)

{

}

}

**Task 2:**

Modify the code to implement the below equation to generate a frequency of 5 Hz. Display

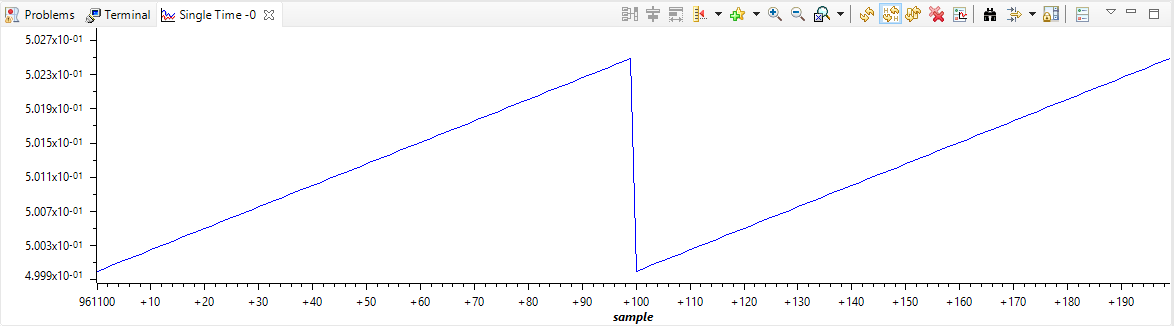
the equation for 1 sec.

1.0\*sin(2π50t) + 0.5\*cos(2π200t)

Recall that:

Therefore, with ω = 50 and series length = 100 we have:

For ω = 200 and series length = 100 we have:



**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> // lib for math functions

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

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

**#include** "driverlib/fpu.h" // floating point lib

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

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

// define the constant for PI as a floating point number

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

// depth of the data buffer for samples

**#define** SERIES\_LENGTH 100

// floating point array that holds the series data points

**float** gSeriesData[SERIES\_LENGTH];

// the counter for computation

int32\_t i32DataCount = 0;

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

{

// radian variable

**float** fRadians;

**float** fRadians\_50;

**float** fRadians\_200;

// enable FP data onto stack, space is reserved

**FPULazyStackingEnable**();

// floating point unit is enabled

**FPUEnable**();

// ==================================================

// Configure system clock

// CLK = 50MHz = (400MHz PLL / (4 \* 2))

// --------------------------------------------------

**SysCtlClockSet**(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

// define radian with respect to the series length

/\*

\* recall that radians can be calculated as:

\*

\* r = (2pi/wavelength) = (2\*pi\*f) / (wavelength / period)

\*

\* f = 1 / T = 1 / (2pi/omega)

\*/

// omega value, 50

fRadians\_50 = ( ((2 \* M\_PI) \* (1/(2 \* M\_PI) / 50)) / (SERIES\_LENGTH / ((2 \* M\_PI) / 50)) );

// omega value, 200

fRadians\_200 = ( ((2 \* M\_PI) \* (1/(2 \* M\_PI) / 200)) / (SERIES\_LENGTH / ((2 \* M\_PI) / 200)) );

// continue to compute while counter is under depth size

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

{

// sin function of given radian value

gSeriesData[i32DataCount] = 1.0 \* **sinf**(fRadians\_50 \* i32DataCount) + 0.5 \* **cosf**(fRadians\_200 \* i32DataCount);

// increment counter by 1

i32DataCount++;

}

// loop here forever

**while**(1)

{

}

}