**10/16/2018**

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

Modifications highlighted in Yellow

**Task 01:**

**Modified Code:**

**#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 // In case not PI defined define it here

**#define** M\_PI 3.14159265358979323846

**#endif**

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

**float** gSeriesData[SERIES\_LENGTH]; // buffer holding float calculated sine values

int32\_t i32DataCount = 0; // declare count index

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

{

**float** fRadians; // 2pi multiplier

ROM\_FPULazyStackingEnable(); // Turning on lazy stacking which helps to avoid an

// increase of interrupt latency by skipping stacking of floating-point registers.

ROM\_FPUEnable(); // Enable the Floating Point module

// Setting clock to 50 MHz

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

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // 100 segments to make 2pi

**while**(i32DataCount < SERIES\_LENGTH) // iterate until count reaches 100 or 2pi

{

// Get sin from 0 to 2 pi in 100 counts and save all values in array

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

i32DataCount++; // increment count index

}

**while**(1) // loop forever

{

}

}

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

**Task 02:**

**Modified Code:**

**#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 // In case not PI defined define it here

**#define** M\_PI 3.14159265358979323846

**#endif**

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

**float** gSeriesData[SERIES\_LENGTH]; // buffer holding float calculated sine values

int32\_t i32DataCount = 0; // declare count index

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

{

**float** fRadians; // 2pi multiplier

ROM\_FPULazyStackingEnable(); // Turning on lazy stacking which helps to avoid an

// increase of interrupt latency by skipping stacking of floating-point registers.

ROM\_FPUEnable(); // Enable the Floating Point module

// Setting clock to 50 MHz

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

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // 1000 segments to make 2pi. To generate 5 Hz and

// display for 1 second, 5 periods are shown below,

// each period take 200 ms which is 200 samples

**while**(i32DataCount < SERIES\_LENGTH) // iterate until count reaches 1000

{

// Get sin from 0 to 1000 counts and save all values in array. This array will

// be viewed in memory

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

i32DataCount++; // increment count index

}

**while**(1) // loop forever

{

}

}

Generating a frequency of 5 Hz for 1 second of signal function

