**Date Submitted: 10/16/2018**

**Task 00: Execute provided code**

**Youtube Link:**

**No submission required for Task 00**

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

**Task 01:**

Youtube Link: No video demonstration required, just comment the code

**Modified Schematic (if applicable):**

**Modified Code:**

**#define** PART\_TM4C123GH6PM //define the chipset or variant of the TM4C123G board

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

**#include** <stdint.h> //include necessary libraries

**#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"

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

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

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846 //define float variable for Pi

**#endif**

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

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

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

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

{

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

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

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

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //set the system clock for 50MHz

fRadians = ((2 \* M\_PI)/SERIES\_LENGTH); //variable to hold the value 2pi/depth of data buffer

**while**(i32DataCount < SERIES\_LENGTH) //calculates the sine wave for each of the 1000 values

{

//The line below calculates each sine value for each t value

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

i32DataCount++; //increment counter

}

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

{

}

}

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

**Task 02:**

Youtube Link: <https://youtu.be/_8G3AHOgiZs>

**Modified Schematic (if applicable):**

**Modified Code:**

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

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

**#include** <stdint.h> //include necessary libraries

**#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"

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

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

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

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

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

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

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

{

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

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

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

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //set the system clock for 50MHz

fRadians = ((2 \* M\_PI)); //variable to hold the value 2pi

//can also do (2\*M\_PI)/1000 and take off the 0.001 in the gSeriesData calculation

**while**(i32DataCount < SERIES\_LENGTH) //calculates the sine wave for each of the 1000 values

{

//I multiplied the things inside the sinf and cof by 0.001 to convert 1s to 1ms

//this can also be done by dividing fRadians by 1000

gSeriesData[i32DataCount] = ((**sinf**(fRadians \* i32DataCount\* 0.001 \* 50) + (0.5 \* **cosf**(fRadians \* i32DataCount\*0.001 \* 200))));

i32DataCount++; //increment counter

}

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

{

}

}

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