**Date Submitted: 10/21/2019**



**Task 01:**

**Youtube Link:**

<https://youtu.be/1jGaAS1A9ck>

**Modified Code:**

#include <stdint.h>

#include <stdbool.h>

#include <math.h>

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_memmap.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_types.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\sysctl.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\rom.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\fpu.h" //Supports floating point

#ifndef M\_PI

#define M\_PI 3.14159265358979323846

#endif

#define SERIES\_LENGTH 100 //depth of buffer

float gSeriesData[SERIES\_LENGTH];

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

int main(void)

{

float fRadians; //Used to calculate sine

ROM\_FPULazyStackingEnable();

ROM\_FPUEnable(); //From reset this is off

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //50Mhz

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

while (i32DataCount < SERIES\_LENGTH)

{

gSeriesData[i32DataCount] = sinf(fRadians \* i32DataCount); //sinf() from math.h

i32DataCount++;

}

while (1)

{

}

}

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

**Task 02:**

Youtube Link:

<https://youtu.be/NRe3mv1Pq-E>

**Modified Code:**

#include <stdint.h>

#include <stdbool.h>

#include <math.h>

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_memmap.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\inc\hw\_types.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\sysctl.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\rom.h"

#include "\ti\tivaware\_c\_series\_2\_1\_4\_178\driverlib\fpu.h" //Supports floating point

#ifndef M\_PI

#define M\_PI 3.14159265358979323846

#endif

#define SERIES\_LENGTH 1000 //depth of buffer

float gSeriesData[SERIES\_LENGTH];

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

int main(void)

{

float fRadians; //Used to calculate sine

float fRadians1; //Used to calculate cosine

ROM\_FPULazyStackingEnable();

ROM\_FPUEnable(); //From reset this is off

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //50Mhz

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

fRadians1 = ((2 \* M\_PI \* 200) / SERIES\_LENGTH);

while (i32DataCount < SERIES\_LENGTH)

{

gSeriesData[i32DataCount] = 1.5 + (1.0\*sinf(fRadians \* i32DataCount)) + (0.5\*(cosf(fRadians1 \* i32DataCount)));

i32DataCount++;

}

while (1)

{

}

}

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