**Date Submitted:**

**10-30**

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

**Task 01:**

Youtube Link: <https://www.youtube.com/watch?v=4_HSrc0m4tg>

**Modified Schematic (if applicable):**

**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

#define M\_PI 3.14159265358979323846

#endif

#define SERIES\_LENGTH 100

float gSeriesData[SERIES\_LENGTH]; // create array of amount wanted of calculations

int32\_t i32DataCount = 0; //initialize data count counter

int main(void)

{

float fRadians; // radians variable

ROM\_FPULazyStackingEnable(); // the function to start avoiding an increase in int latency.

ROM\_FPUEnable(); // enable the fpu for use

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

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // value used for graphing waves (2pi)

while(i32DataCount < SERIES\_LENGTH) // loop for series lenggth times.

{

gSeriesData[i32DataCount] = sinf(fRadians \* i32DataCount); // sets equation to graph (sin(2pi)).

i32DataCount++; //increment.

}

while(1) //loop forever for calc

{

}

}**------------------------------------------------------------------------------------**

**Task 02:**

Youtube Link: https://www.youtube.com/watch?v=twNUnETQB6Y

**Modified Schematic (if applicable):**

**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"**

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

**// used for assigning radians value**

**#ifndef M\_PI**

**#define M\_PI 3.14159265358979323846**

**#endif**

**#define SERIES\_LENGTH 1000**

**float gSeriesData[SERIES\_LENGTH];// create array of amount wanted of calculations**

**int32\_t i32DataCount = 0; //initialize data count counter**

**int main(void)**

**{**

**float fRadians; // radians variable**

**//enable fpu calculations**

**ROM\_FPULazyStackingEnable();// the function to start avoiding an increase in int latency.**

**ROM\_FPUEnable();// enable the fpu for use**

**// set clock**

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

**// set value for radians**

**fRadians = ((2 \* M\_PI) / SERIES\_LENGTH);// value used for graphing waves (2pi)**

**// count for 100 times**

**while(i32DataCount < SERIES\_LENGTH) // loop for series lenggth times.**

**{**

**// equation to graph / create**

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

**// sets equation to graph (sin(2pi)).**

**i32DataCount++; //increment.**

**}**

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

**{**

**}**

**}**