**Date Submitted:**

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

**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"

#define TARGET\_IS\_BLIZZARD\_RB1

#ifndef M\_PI

#define M\_PI 3.14159265358979323846

#endif

#define SERIES\_LENGTH 100

float gSeriesData[SERIES\_LENGTH];

int32\_t i32DataCount = 0;

int main(void)

{

float fRadians;

ROM\_FPULazyStackingEnable();

ROM\_FPUEnable();

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

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

while(i32DataCount < SERIES\_LENGTH)

{

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

i32DataCount++;

}

while(1)

{

}

}

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

**Task 02:**

Youtube Link: <https://www.youtube.com/watch?v=KSHmncGi6hg>

**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 100

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

int32\_t i32DataCount = 0;

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

{

**float** fRadians;

//enable fpu calculations

ROM\_FPULazyStackingEnable();

ROM\_FPUEnable();

// 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);

// count for 100 times

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

{

// equation to graph / create

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

i32DataCount++;

}

**while**(1)

{

}

}