Ivan Soto
Github root directory: https://github.com/sotoi2/Class3.0.4

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

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

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

    {

    }

}
```