Geovanni Portillo
Github root directory: https://github.com/portig1/submissions_E2

Date Submitted: October 23, 2019

Task 01:

Youtube Link: https://youtu.be/UVrZKnKOIgs

Modified Schematic (if applicable):

Modified Code:

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //uses sinf function
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" //support floating point unit
#include "driverlib/sysctl.h"
#include "driverlib/rom.h"


#ifndef M_PI    //in case M_PI is undefined, this bit of code takes care of it
#define M_PI                    3.14159265358979323846
#endif

#define SERIES_LENGTH 100 //depth of data buffer

float gSeriesData[SERIES_LENGTH]; //an array of floats series_length long

int32_t i32DataCount = 0; //counter for computation loop

int main(void)
{
    float fRadians; //needed to calculate sine

    /*ROM_*/FPULazyStackingEnable(); //turns on Lazy Stacking
    /*ROM_*/FPUEnable(); //turns on FPU

    /*ROM_*/SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN); //sets system clock for 50MHz

    fRadians = ((2 * M_PI) / SERIES_LENGTH); //full sine wave cycle is 2pi radians.
divides 2pi by the depth of array

    while(i32DataCount < SERIES_LENGTH) //calculates sine value for each of the 100
values of the angle and places them in the array
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
        i32DataCount++;
    }

    while(1)
```

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

```
    {
    }
}
```

Task 02:

Youtube Link: https://youtu.be/zXwqzPat-_I

Modified Schematic (if applicable):

Modified Code:

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //uses sinf function
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" //support floating point unit
#include "driverlib/sysctl.h"
#include "driverlib/rom.h"


#ifndef M_PI    //in case M_PI is undefined, this bit of code takes care of it
#define M_PI                    3.14159265358979323846
#endif

#define SERIES_LENGTH 1000 //depth of data buffer

float gSeriesData[SERIES_LENGTH]; //an array of floats series_length long

int32_t i32DataCount = 0; //counter for computation loop

int main(void)
{
    float fRadians; //needed to calculate sine

    /*ROM_*/FPULazyStackingEnable(); //turns on Lazy Stacking
    /*ROM_*/FPUEnable(); //turns on FPU

    /*ROM_*/SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN); //sets system clock for 50MHz

    fRadians = ((2 * M_PI) / SERIES_LENGTH); //full sine wave cycle is 2pi radians.
divides 2pi by the depth of array

    while(i32DataCount < SERIES_LENGTH) //calculates sine value for each of the 100
values of the angle and places them in the array
    {
```

```
        gSeriesData[i32DataCount] = 1.5 + sinf(fRadians * 50 * i32DataCount) +
0.5*cosf(fRadians * 200 * i32DataCount);

        i32DataCount++;
    }

    while(1)
    {
    }
}
```
