

Github root directory: <https://github.com/Ayertena/TIVAC>

Date Due: October 16th, 2018

Task 01: Submit a comprehensive commented file of the original code.

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //allows the use of sinf()
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" //for floating point values
#include "driverlib/sysctl.h"
#include "driverlib/rom.h" //for ROM functions

#ifndef M_PI
#define M_PI 3.14159265358979323846 //value for pi
#endif

#define SERIES_LENGTH 100 //amount of calculation (100) being made
float gSeriesData[SERIES_LENGTH]; //holds calculated number
int32_t i32DataCount = 0; //count number of calculations made
int main(void)
{
    float fRadians; // declare variable to hold 2pi / 100
    ROM_FPULazyStackingEnable(); // enable lazy stacking
    ROM_FPUEnable(); // enable FPU
    //50 MHz clock
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL |
    SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    fRadians = ((2 * M_PI) / SERIES_LENGTH); // 2pi / 100

    while(i32DataCount < SERIES_LENGTH) //100 calculations are made
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
        //obtain sine value
        i32DataCount++; //increment to obtain next sine value
    }
    while(1){} }
```

Task 02: Modify the code to implement the below equation to generate a frequency of 5 Hz. Display the equation for 1 sec. $1.0 \cdot \sin(2\pi \cdot 50t) + 0.5 \cdot \cos(2\pi \cdot 200t)$

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //allows the use of sinf()
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" //for floating point values
#include "driverlib/sysctl.h"
#define TARGET_IS_BLIZZARD_RB1
#include "driverlib/rom.h" //for ROM functions

#ifndef M_PI
#define M_PI 3.14159265358979323846 //value for pi
#endif

#define SERIES_LENGTH 1000 //amount of calculation (100) being made
float gSeriesData[SERIES_LENGTH]; //holds calculated number

int32_t i32DataCount = 0; //count number of calculations made

int main(void)
{
    float fRadians; // declare variable to hold 2pi / 100

    ROM_FPULazyStackingEnable();    // enable lazy stacking
    ROM_FPUEnable();                // enable FPU

    //50 MHz clock
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL |
        SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    fRadians = ((2 * M_PI) / SERIES_LENGTH); // 2pi / 100

    while(i32DataCount < SERIES_LENGTH) //100 calculations are made
    {
        //1.0*sin(2p50t) + 0.5*cos(2p 200t)
        //obtain sine value
        gSeriesData[i32DataCount] = sinf(50*fRadians * i32DataCount) +
0.5*cosf(200*fRadians * i32DataCount);
        i32DataCount++; //increment to obtain next sine value
    }
}
```

```
while(1) //continuous while loop
{
}
}
```

Waveform

