

Date Submitted: 10/18/18**Task 01:**

Youtube Link: N/A

```

#define SERIES_LENGTH 100    // amount of values to store in buffer

float gSeriesData[SERIES_LENGTH]; // will store all values of sine wave
int32_t i32DataCount = 0;

int main(void)
{
    float fRadians;
    FPU_LazyStackingEnable(); // enable lazy stacking which prevents stacking floating
    point
                                // values to lower latency time between interrupts
    FPU_Enable();    // turn on floating point

    // set the clock for 50MHz
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);
    fRadians = ((2 * M_PI) / SERIES_LENGTH); // get 2*PI value
    while(i32DataCount < SERIES_LENGTH)
    {
        // get the 100 sine values 1 at a time and store them into the array
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
        i32DataCount++;
    }
    while(1)
    {
        // once done, go into infinite loop...
    }
}

```

Task 02:

Youtube Link: N/A

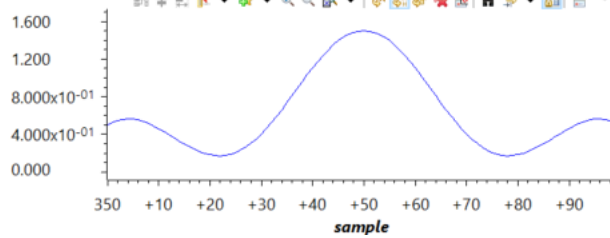
```

int main(void)
{
    float fRadians;
    FPU_LazyStackingEnable(); // enable lazy stacking which prevents stacking floating
    point
                                // values to lower latency time between interrupts
    FPU_Enable();    // turn on floating point

    // set the clock for 50MHz
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);
    fRadians = ((2 * M_PI) / 200); // get 2*PI value
    while(i32DataCount < SERIES_LENGTH)

```

```
{  
    // get the 100 sine values 1 at a time and store them into the array  
    gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount) + 0.5 *  
    cosf(fRadians * 4 * i32DataCount);  
    i32DataCount++;  
}  
while(1)  
{  
    // once done, go into infinite loop...  
}  
}
```



Brian Lopez
Github root directory: github.com/brian4280/Bed_Assign

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