

**Date Submitted:** 10/30/2019**Task 01:**Youtube Link: <https://youtu.be/mcx4aCwnDgo>**Modified Code:**

```
//Lab9 Task 1
#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 //pi equation
#endif

#define SERIES_LENGTH 100 //array size

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

int32_t i32DataCount = 0; //a counter for our computation loop

int main(void)
{
    float fRadians; //float variable to store equation

    ROM_FPULazyStackingEnable(); //enables the lazy stacking of floating point
integers
    ROM_FPUEnable(); //enables floating point unit

    //clock initialization
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

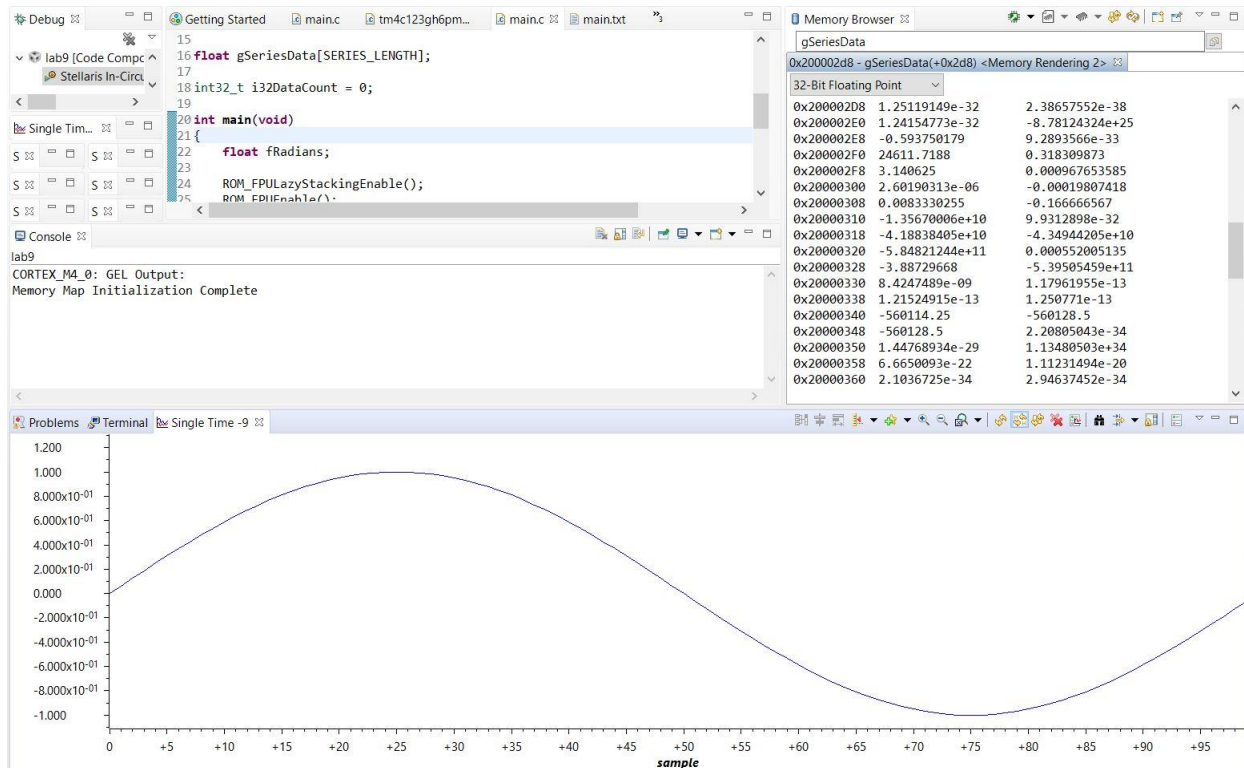
    fRadians = ((2 * M_PI) / SERIES_LENGTH); //setting equation to the variable

    //while loop to loop while i32DataCount is less than SERIES_LENGTH
    while(i32DataCount < SERIES_LENGTH)
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount); //data array

        i32DataCount++; //increment i32DataCount
    }

    while(1)
    {
    }
}
```

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



## Task 02:

Youtube Link: [https://youtu.be/U0U\\_eBzqHLE](https://youtu.be/U0U_eBzqHLE)

Modified Code:

```

//Lab9 Task 2
#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"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
#include "driverlib/timer.h"
#include "driverlib/debug.h"
#include "driverlib/uart.h"
#include "utils/uartstdio.h"
#include "driverlib/gpio.h"
#include "inc/hw_gpio.h"
#include "driverlib/pin_map.h"

#ifndef M_PI
#define M_PI 3.14159265358979323846

```

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

```

#endif

#define SERIES_LENGTH 1000 //array size

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

int32_t i32DataCount = 0; //i32DataCount initially set to 0

int main(void)
{
    float fRadians; //float variable to store equation
    float fRadians1; //float variable to store equation 2

    ROM_FPULazyStackingEnable(); //enables the lazy stacking of floating point
integers
    ROM_FPUEnable(); //enables floating point unit

    //clock initialization
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    //UART initialization
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    GPIOPinConfigure(GPIO_PA0_U0RX);
    GPIOPinConfigure(GPIO_PA1_U0TX);
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
    UARTClockSourceSet(UART0_BASE, UART_CLOCK_PIOSC);
    UARTStdioConfig(0, 115200, 16000000);

    //variables for both equations
    fRadians = ((2 * M_PI * 50) / SERIES_LENGTH);
    fRadians1 = ((2 * M_PI * 200) / SERIES_LENGTH);

    while(i32DataCount < SERIES_LENGTH)
    {
        //gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount) +
(0.5*cosf(fRadians1 * i32DataCount));

        //print onto terminal
        UARTprintf("sin(2pi * 50(%d)) + 0.5*cos(2pi * 200(%d))\n", i32DataCount,
i32DataCount);
        i32DataCount++; //increment i32DataCount
        SysCtlDelay(10000000); //delay
    }

    while(1)
    {
    }
}

```

