Saul Mendoza
Github root directory:
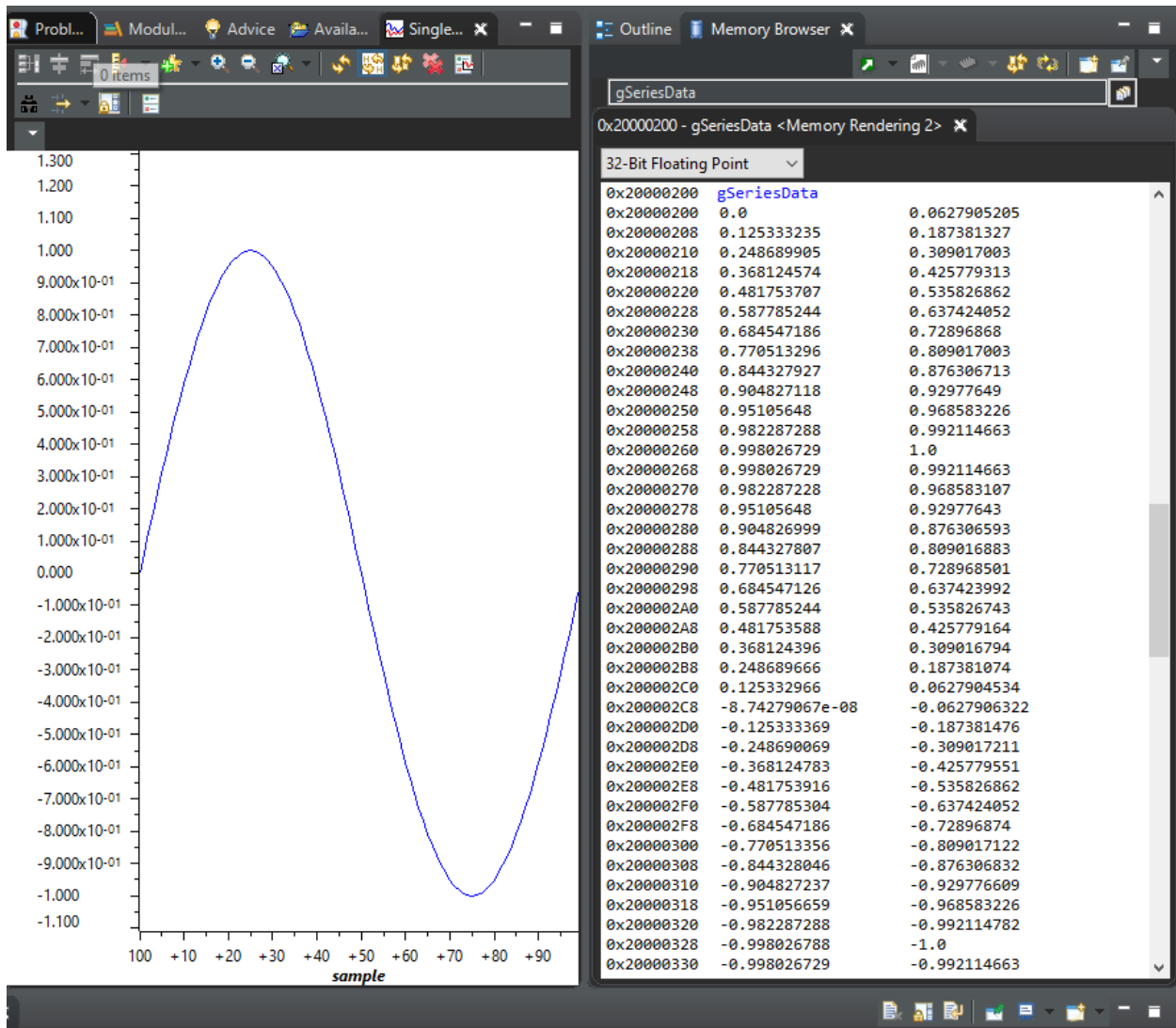https://github.com/mendos1/Submission_Link/tree/master/Tiva_C

Date Submitted: 11/4/19

Task 01:

Youtube Link:
https://www.youtube.com/watch?v=ioUYoaFctrU

Modified Schematic (if applicable):

Identity	Name	Condition	Count	Action
☑ Clock Cycles [H/W]	Count Event		11412	
☑ main.c, line 30 (main	Breakpoint		0 (0)	Remain Halt
☑ main.c, line 38 ($CSL	Breakpoint		0 (0)	Remain Halt

Modified Code:

```c
// Insert code here
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //code uses sinf() function which is in this header file
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" //support for floating point unit
#include "driverlib/sysctl.h"
#include "driverlib/rom.h"

//Just in case the M_PI is not already defined this will do it for us
#ifndef M_PI
#define M_PI    3.14159265358979323846
#endif

#define SERIES_LENGTH 100 //depth of our data buffer
float gSeriesData[SERIES_LENGTH]; //an array of floats SERIES_LENGTH long (100)

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

int main(void)
{
    float fRadians; //need a variable of type float to calculate sine
    ROM_FPULazyStackingEnable();//turn on lazy stacking
    ROM_FPUEnable();//Turn on the FPU

    //Set the system clock to 50MHz
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    //a full sine wave cycle is 2pi radians. Divide 2pi by the depth of the array
fRadians = ((2*M_PI)/SERIES_LENGTH);
    //calculate the sine value for each of the 100 values of the angle and place them
in our data array
    while(i32DataCount < SERIES_LENGTH)
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
i32DataCount++;
    }
```

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

```
    //endless loop
while(1)
    {
    }
}
```
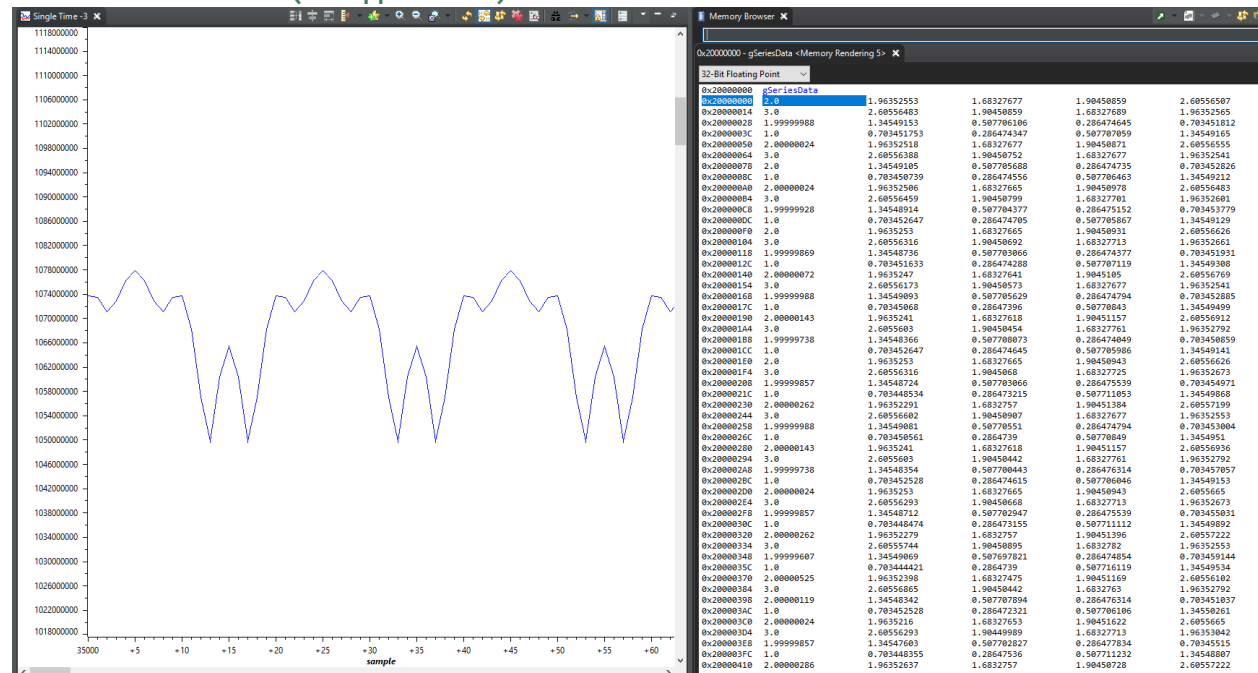--

Task 02:

I had to implement the equation below in my code:

$$\sin(2\pi * 50t) + 0.5 * \cos(2\pi * 200t)$$

Youtube Link:
https://www.youtube.com/watch?v=g7CfzVLjm0s

Modified Schematic (if applicable):

Modified Code:

```c
// Insert code here
#include <stdint.h>
#include <stdbool.h>
```

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

Saul Mendoza
Github root directory:
https://github.com/mendos1/Submission_Link/tree/master/Tiva_C

```c
#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 1000 //size of buff array
float gSeriesData[SERIES_LENGTH]; //used to store the array of floats (1000)
 int32_t i32DataCount = 0;

int main void {
    float fRad_mid_part, fRad_end_part; //used to calculate the intermediate vars 0f
the equation
    ROM_FPULazyStackingEnable();//enable lazy stacking
    ROM_FPUEnable();// enable the FPU
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    fRad_mid_part =  2*M_PI*50 /SERIES_LENGTH;  // used to model the equations that
    fRad_end_part =  2*M_PI*200 /SERIES_LENGTH; // generates the 50Hz

    // here we calculate the values of the wave
    while(i32DataCount < SERIES_LENGTH) {
        gSeriesData[i32DataCount] = 1.5+ 1.0*sinf(fRad_mid_part *
i32DataCount)+0.5*cosf(fRad_end_part * i32DataCount));
        i32DataCount++;
    }

    //endless loop
    while(1)
    {

    }
}
```

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