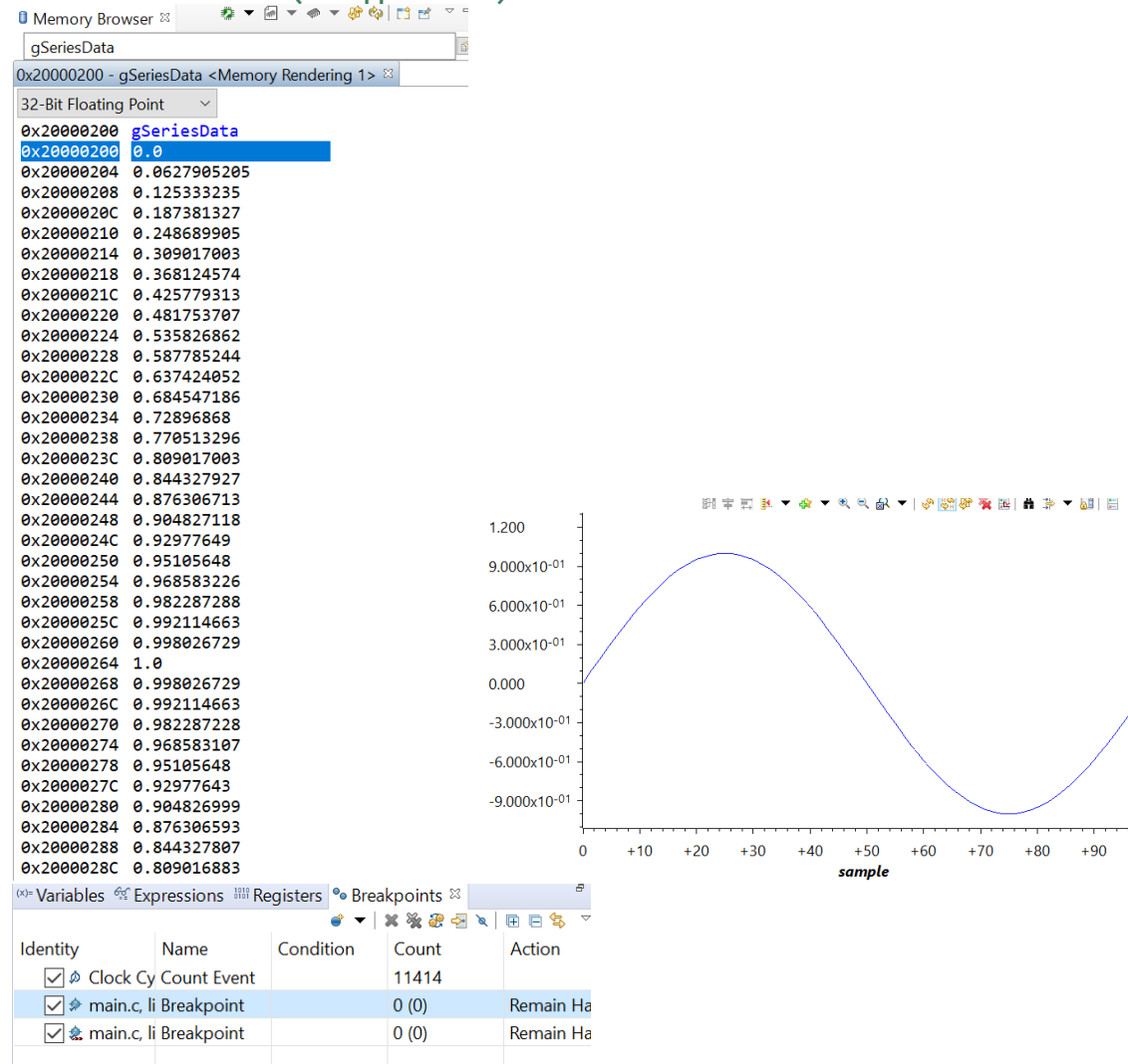


**Date Submitted: 10/29/19****Task 01:**

In this task, we are to follow along with the tutorial and comment the code extensively. I will be using an equation that calculates floating point values and then putting it into an array.

Youtube Link:

<https://www.youtube.com/watch?v=ZREeFBYWh4>

**Modified Schematic (if applicable):****Modified Code:**

```
// 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"
```

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

```

#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++;
    }

    //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:

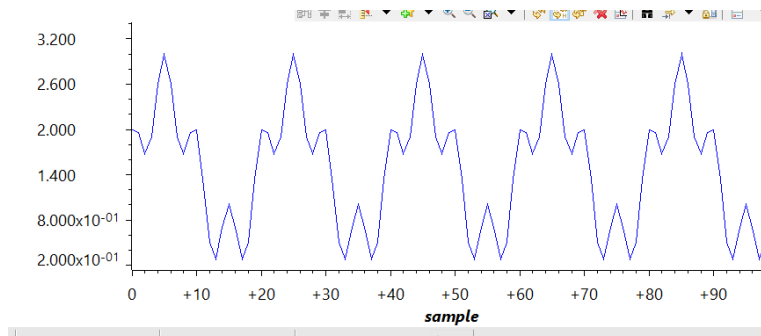
<https://www.youtube.com/watch?v=rWMSsDlZYuA>

Modified Schematic (if applicable):

```

0x20000000 - gSeriesData <Memory Rendering>
32-Bit Floating Point
0x20000000 gSeriesData
0x20000000 2.0
0x20000004 1.96352553
0x20000008 1.68327677
0x2000000C 1.90450859
0x20000010 2.60556507
0x20000014 3.0
0x20000018 2.60556483
0x2000001C 1.90450859
0x20000020 1.68327689
0x20000024 1.96352565
0x20000028 1.99999988
0x2000002C 1.34549153
0x20000030 0.507706106
0x20000034 0.286474645
0x20000038 0.703451812
0x2000003C 1.0
0x20000040 0.703451753
0x20000044 0.286474347
0x20000048 0.507707059
0x2000004C 1.34549165
0x20000050 2.00000024
0x20000054 1.96352518
0x20000058 1.68327677
0x2000005C 1.90450871
0x20000060 2.60556555
0x20000064 3.0
0x20000068 2.60556388
0x2000006C 1.90450752
0x20000070 1.68327677
0x20000074 1.96352541
0x20000078 2.0
0x2000007C 1.34549105
0x20000080 0.507705688
0x20000084 0.286474735
0x20000088 0.703452826

```



#### Modified Code:

```

// 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 1000 //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, fRadians1, fRadians2; //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
fRadians1 = (2*M_PI*50)/SERIES_LENGTH;
fRadians2 = (2*M_PI*200)/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] = 1.5+(1.0*sinf(fRadians1 *
i32DataCount)+0.5*cosf(fRadians2 * i32DataCount));
    i32DataCount++;
}

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

---