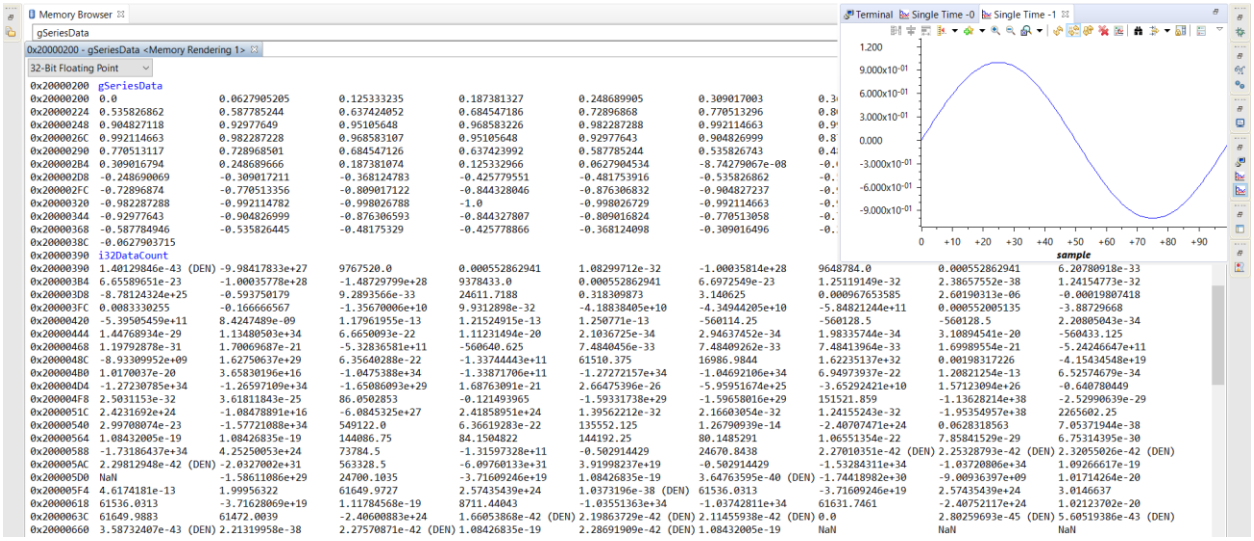
Ameera Essaqi

Github root directory: https://github.com/AmeeraE/microcontrollers/tree/master/TIVAC

Date Submitted: **11/30/19**

Task 01:

Youtube Link: N/A



0x20000200 gSeriesData

0x20000200	0.0	0.0627905205	0.125333235	0.187381327	0.248689905	0.309017003	0.3(	
0x20000224	0.535826862	0.587785244	0.637424052	0.684547186	0.72896868	0.770513296	0.8(	
0x20000248	0.904827118	0.92977649	0.95105648	0.968583226	0.982287288	0.992114663	0.9(	
0x2000026C	0.992114663	0.982287228	0.968583107	0.95105648	0.92977643	0.904826999	0.8	
0x20000290	0.770513117	0.728968501	0.684547126	0.637423992	0.587785244	0.535826743	0.4	
0x200002B4	0.309016794	0.248689666	0.187381074	0.125332966	0.0627904534	-8.74279067e-08	-0.(	
0x200002D8	-0.248690069	-0.309017211	-0.368124783	-0.425779551	-0.481753916	-0.535826862	-0.	
0x200002FC	-0.72896874	-0.770513356	-0.809017122	-0.844328046	-0.876306832	-0.904827237	-0.(	
0x20000320	-0.982287288	-0.992114782	-0.998026788	-1.0	-0.998026729	-0.992114663	-0.(	
0x20000344	-0.92977643	-0.904826999	-0.876306593	-0.844327807	-0.809016824	-0.770513058	-0.	
0x20000368	-0.587784946	-0.535826445	-0.48175329	-0.425778866	-0.368124098	-0.309016496	-0.	
0x2000038C	-0.0627903715							

0x20000390 i32DataCount

0x20000390	1.40129846e-43 (DEN)	-9.98417833e+27	9767520.0	0.000552862941	1.08299712e-32	-1.00035814e+28	9648784.0	0.000552862941	6.20780918e-33
0x200003B4	6.65589651e-23	-1.00035778e+28	-1.48729799e+28	9378433.0	0.000552862941	6.6972549e-23	1.25119149e-32	2.38657552e-38	1.24154773e-32
0x200003D8	-8.78124324e+25	-0.593750179	9.2893566e-33	24611.7188	0.318309873	3.140625	0.000967653585	2.60190313e-06	-0.000198074 18
0x200003FC	0.0083330255	-0.166666567	-1.35670006e+10	9.9312898e-32	-4.18838405e+10	-4.34944205e+10	-5.84821244e+11	0.000552005135	-3.88729668
0x20000420	-5.39505459e+11	8.4247489e-09	1.17961955e-13	1.21524915e-13	1.250771e-13	-560114.25	-560128.5	-560128.5	2.20805043e-34
0x20000444	1.44768934e-29	1.13480503e+34	6.6650093e-22	1.11231494e-20	2.1036725e-34	2.94637452e-34	1.98335744e-34	3.10894541e-20	-560433.125
0x20000468	1.19792878e-31	1.70069687e-21	-5.32836581e+11	-560640.625	7.4840456e-33	7.48409262e-33	7.48413964e-33	1.69989554e-21	-5.24246647e+11
0x2000048C	-8.93309952e+09	1.62750637e+29	6.35640288e-22	-1.33744443e+11	61510.375	16986.9844	1.62235137e+32	0.00198317226	-4.15434548e+19
0x200004B0	1.0170037e-20	3.65830196e+16	-1.0475388e+34	-1.33871706e+11	-1.27272157e+34	-1.04692106e+34	6.94973937e-22	1.20821254e-13	6.52574679e-34
0x200004D4	-1.27230785e+34	-1.26597109e+34	-1.65086093e+29	1.68763091e-21	2.66475396e-26	-5.95951674e+25	-3.65292421e+10	1.57123094e+26	-0.640780449
0x200004F8	2.5031153e-32	3.61811843e-25	86.0502853	-0.121493965	-1.59331738e+29	-1.59658016e+29	151521.859	-1.13628214e+38	-2.52990639e-29
0x2000051C	2.4231692e+24	-1.08478891e+16	-6.0845325e+27	2.41858951e+24	1.39562212e-32	2.16603054e-32	1.24155243e-32	-1.95354957e+38	2265602.25
0x20000540	2.99708074e-23	-1.57721088e+34	549122.0	6.36619283e-22	135552.125	1.26790939e-14	-2.40707471e+24	0.0628318563	7.05371944e-38
0x20000564	1.08432005e-19	1.08426835e-19	144086.75	84.1504822	144192.25	80.1485291	1.06551354e-22	7.85841529e-29	6.75314395e-30
0x20000588	-1.73186437e+34	4.25250053e+24	73784.5	-1.31597328e+11	-0.502914429	24670.8438	2.27010351e-42 (DEN)	2.25328793e-42 (DEN)	2.32055026e-42 (DEN)
0x200005AC	2.29812948e-42 (DEN)	-2.0327002e+31	563328.5	-6.09760133e+31	3.91998237e+19	-0.502914429	-1.53284311e+34	-1.03720806e+34	1.09266617e-19
0x200005D0	NaN	-1.58611086e+29	24700.1035	-3.71609246e+19	1.08426835e-19	3.64763595e-40 (DEN)	-1.74418982e+30	-9.00936397e+09	1.01714264e-20
0x200005F4	4.6174181e-13	1.99956322	61649.9727	2.57435439e+24	1.0373196e-38 (DEN)	61536.0313	-3.71609246e+19	2.57435439e+24	3.0146637
0x20000618	61536.0313	-3.71628069e+19	1.11784568e-19	8711.44043	-1.03551363e+34	-1.03742811e+34	61631.7461	-2.40752117e+24	1.02123702e-20
0x2000063C	61649.9883	61472.0039	-2.40600883e+24	1.66053868e-42 (DEN)	2.19863729e-42 (DEN)	2.11455938e-42 (DEN)	0.0	2.80259693e-45 (DEN)	5.60519386e-43 (DEN)
0x20000660	3.58732407e-43 (DEN)	2.21319958e-38	2.27570871e-42 (DEN)	1.08426835e-19	2.28691909e-42 (DEN)	1.08432005e-19	NaN	NaN	NaN

Modified Code:

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //uses sinf() function
#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"

#ifndef M_PI  //defines M_PI
#define M_PI                        3.14159265358979323846
#endif

#define SERIES_LENGTH 100   //creates depth of data buffer

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

int32_t i32DataCount = 0;   //computation loop counter

int main(void)
{
    float fRadians;  //used to calculate sine

    ROM_FPULazyStackingEnable();    //turn on Lazy Stacking
    ROM_FPUEnable();    //turn on FPU, from reset it is off

    //set up system clock to 50MHz
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    fRadians = ((2 * M_PI) / SERIES_LENGTH); //creates full sine wave cycle
```

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

```
    while(i32DataCount < SERIES_LENGTH)
    {
        //loop to calculate the sine value for each
        //of the 100 values of the angle
        //then places them in the array
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount); //calculation
        i32DataCount++;  //add to array count
    }

    while(1)
    {
    }
}
```

Task 02:

Youtube Link: https://youtu.be/iGIinh2atBs

Modified Code:

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> //uses sinf() function
#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"

#ifndef M_PI  //defines M_PI
#define M_PI                    3.14159265358979323846
#endif

#define SERIES_LENGTH 100   //creates depth of data buffer

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

int32_t i32DataCount = 0;   //computation loop counter

int main(void)
{
    float fRadians_one;  //used to calculate sine at 50t
    float fRadians_two; //used to calculate sine at 200t
    FPULazyStackingEnable();    //turn on Lazy Stacking
    FPUEnable();    //turn on FPU, from reset it is off

    //set up system clock to 50MHz
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    fRadians_one = ((2 * M_PI*50) / SERIES_LENGTH); // First pi
    fRadians_two = ((2 * M_PI*200) / SERIES_LENGTH); // second pi

    while(i32DataCount < SERIES_LENGTH)
    {
        //loop to calculate the sine value for each
        //of the 100 values of the angle
        //then places them in the array
```

```
        gSeriesData[i32DataCount] = 1.0*sinf(fRadians_one * i32DataCount) +
(0.5*(cosf(fRadians_two * i32DataCount)));
        //calculation
        i32DataCount++;  //add to array count
    }

    while(1)
    {
    }
}
```

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