John Duriman

CPE 403

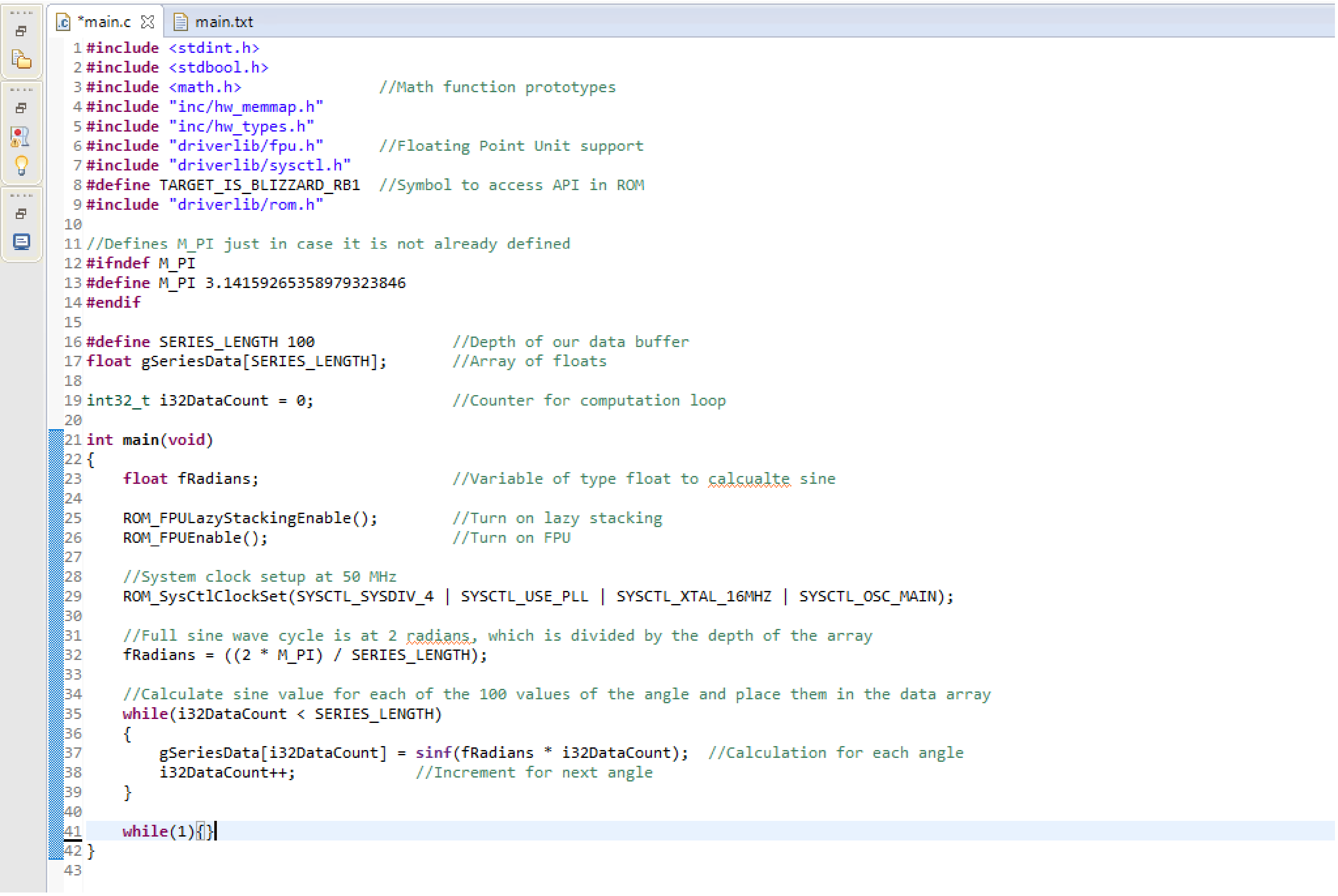
Lab 9

5002373995

Task 01: Submit a comprehensive commented file of the original code

Youtube Link:

Modified Code:

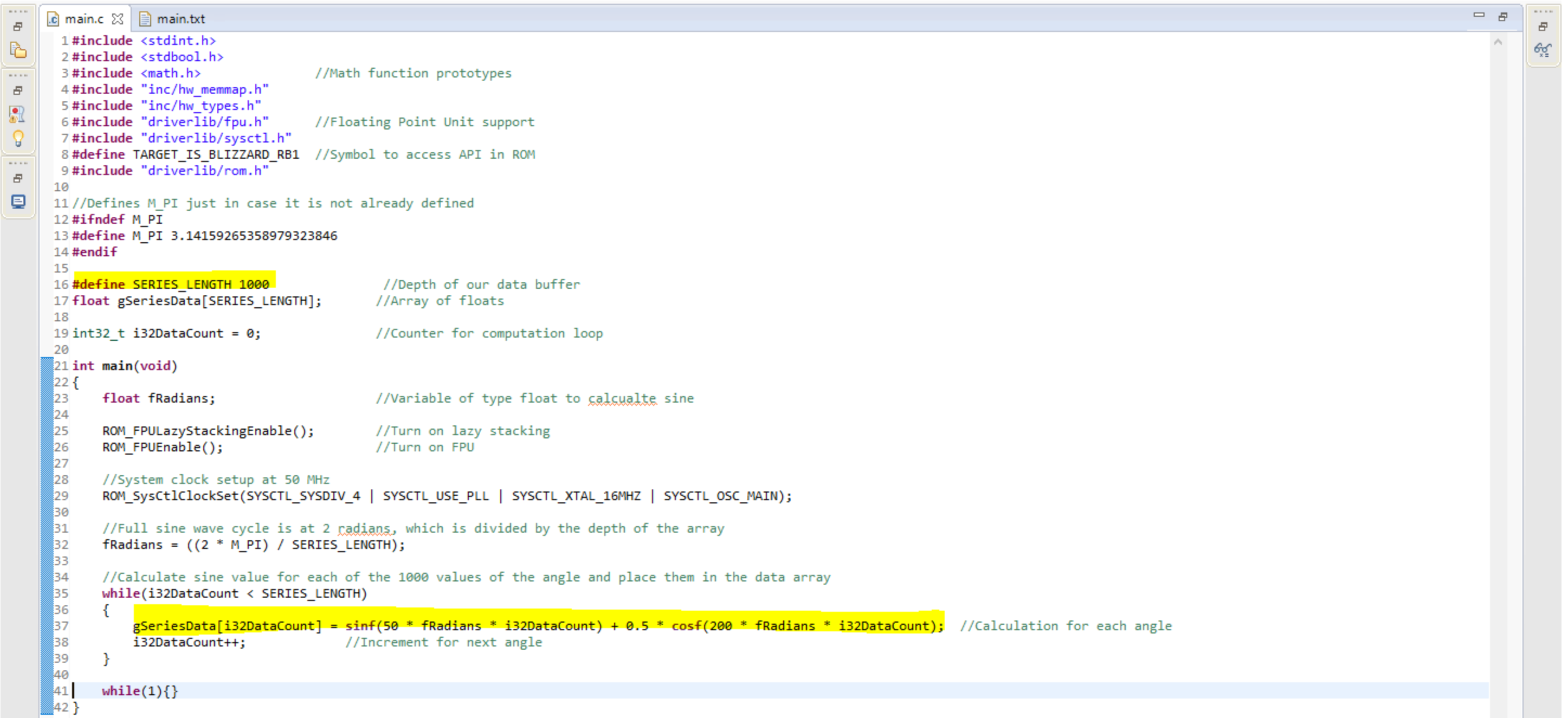


```c
#include <stdint.h>
#include <stdbool.h>
#include <math.h>               //Math function prototypes
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h"      //Floating Point Unit support
#include "driverlib/sysctl.h"
#define TARGET_IS_BLIZZARD_RB1  //Symbol to access API in ROM
#include "driverlib/rom.h"

//Defines M_PI just in case it is not already defined
#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

#define SERIES_LENGTH 100               //Depth of our data buffer
float gSeriesData[SERIES_LENGTH];       //Array of floats

int32_t i32DataCount = 0;               //Counter for computation loop

int main(void)
{
    float fRadians;                     //Variable of type float to calcualte sine

    ROM_FPULazyStackingEnable();        //Turn on lazy stacking
    ROM_FPUEnable();                    //Turn on FPU

    //System clock setup at 50 MHz
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    //Full sine wave cycle is at 2 radians, which is divided by the depth of the array
    fRadians = ((2 * M_PI) / SERIES_LENGTH);

    //Calculate sine value for each of the 100 values of the angle and place them in the data array
    while(i32DataCount < SERIES_LENGTH)
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);  //Calculation for each angle
        i32DataCount++;             //Increment for next angle
    }

    while(1){}
}
```

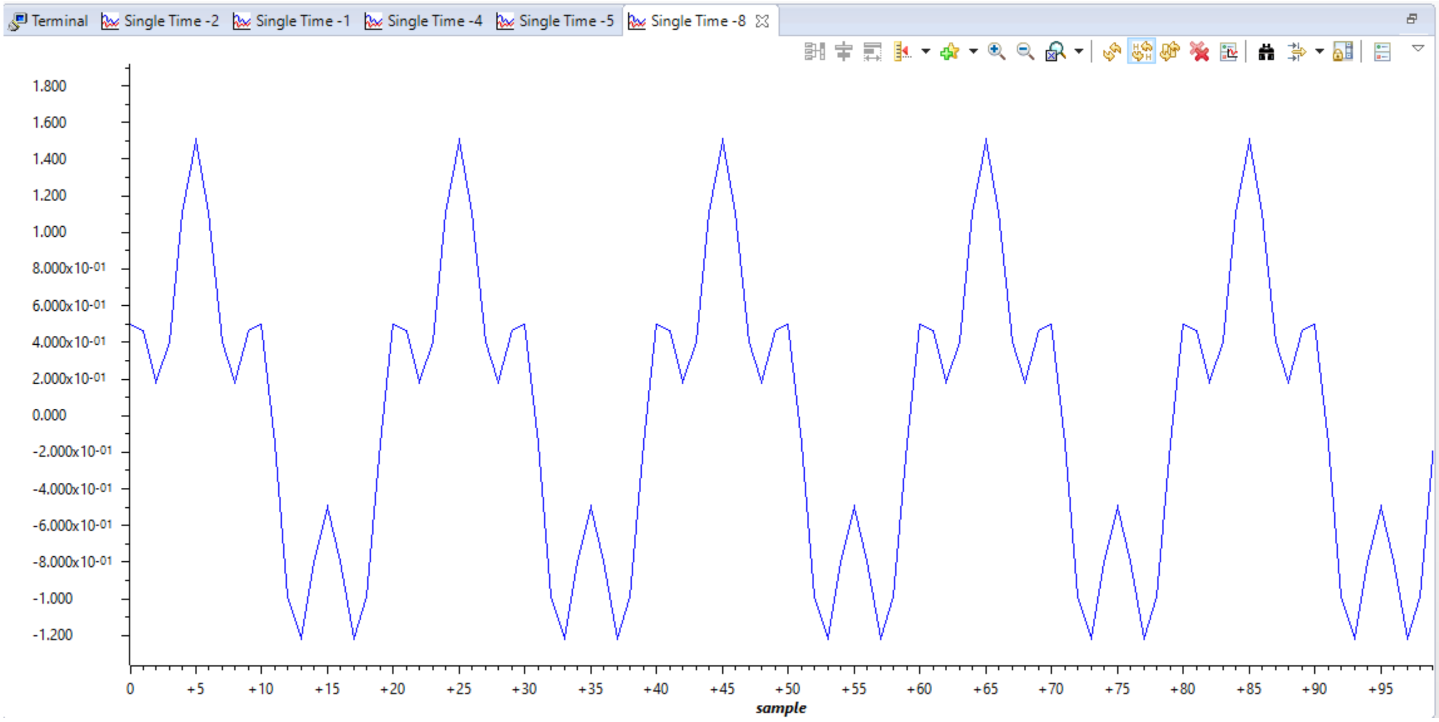
Task 02: Modify the code to implement the below equation to generate a frequency of 5Hz. Display the equation for 1 second.

Youtube Link:

Modified Code:



```c
#include <stdint.h>
#include <stdbool.h>
#include <math.h>                //Math function prototypes
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h"       //Floating Point Unit support
#include "driverlib/sysctl.h"
#define TARGET_IS_BLIZZARD_RB1  //Symbol to access API in ROM
#include "driverlib/rom.h"

//Defines M_PI just in case it is not already defined
#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

#define SERIES_LENGTH 1000                //Depth of our data buffer
float gSeriesData[SERIES_LENGTH];        //Array of floats

int32_t i32DataCount = 0;                //Counter for computation loop

int main(void)
{
    float fRadians;                      //Variable of type float to calcualte sine

    ROM_FPULazyStackingEnable();         //Turn on lazy stacking
    ROM_FPUEnable();                     //Turn on FPU

    //System clock setup at 50 MHz
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    //Full sine wave cycle is at 2 radians, which is divided by the depth of the array
    fRadians = ((2 * M_PI) / SERIES_LENGTH);

    //Calculate sine value for each of the 1000 values of the angle and place them in the data array
    while(i32DataCount < SERIES_LENGTH)
    {
        gSeriesData[i32DataCount] = sinf(50 * fRadians * i32DataCount) + 0.5 * cosf(200 * fRadians * i32DataCount);  //Calculation for each angle
        i32DataCount++;             //Increment for next angle
    }

    while(1){}
}
```


Terminal
Single Time -2
Single Time -1
Single Time -4
Single Time -5
Single Time -8
8.000x10-01
6.000x10-01
4.000x10-01
2.000x10-01
-2.000x10-01
-4.000x10-01
-6.000x10-01
-8.000x10-01
sample