

TivaC Lab 9 - FPU

CPE 403

Checklist for Lab 9

- ☑ *A text/word document of the initial code with comments*
- ☑ *In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also include the comments.*
- ☑ *Provide a permanent link to all main and dependent source code files only (name them as LabXX-TYY, XX-Lab# and YY-task#)Screenshots of debugging process along with pictures of actual circuit*
- ☑ *Video link of demonstration.*

Code for Experiment

Task 1:

```

#include <stdint.h>
#include <stdbool.h>
#include <math.h> // Include math functions
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" // Used for floating point
#include "driverlib/sysctl.h"

#define TARGET_IS_BLIZZARD_RB1 //Used to access API's in ROM.
#include "driverlib/rom.h"

#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

#define SERIES_LENGTH 100 // 100 calulations

float gSeriesData[SERIES_LENGTH]; // Array for floating pt data
int32_t i32DataCount = 0; // Iteration counter

int main(void) {
    float fRadians; // floating pt radians

    ROM_FPULazyStackingEnable(); // Enable lazy stacking
    ROM_FPUEnable(); // Enable FPU

    // Using 50 MHz clock
    ROM_SysCtlClockSet(
        SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ
        | SYSCTL_OSC_MAIN);

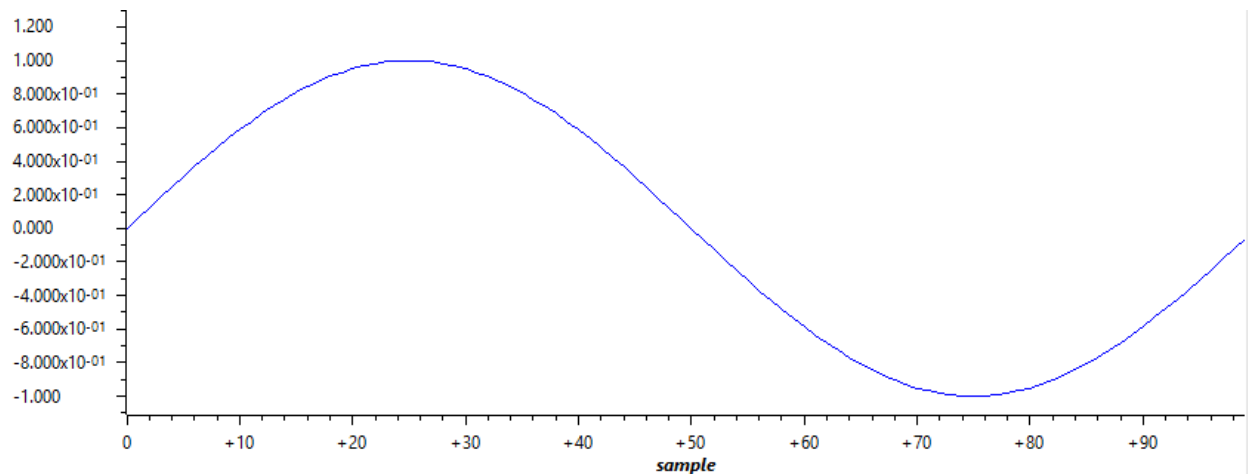
    fRadians = ((2 * M_PI) / SERIES_LENGTH); // Compute radians

    // Calculate 100 values
    while (i32DataCount < SERIES_LENGTH) {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount); // store sine wave dadta
        i32DataCount++; // go to next increment
    }

    while (1)
        ;
}

```

Image of Sine Wave



Task 2:

```
#include <stdint.h>
#include <stdbool.h>
#include <math.h> // Include math functions
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" // Used for floating point
#include "driverlib/sysctl.h"

#define TARGET_IS_BLIZZARD_RB1 //Used to access API's in ROM.
#include "driverlib/rom.h"

#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

#define SERIES_LENGTH 1000 // 100 calculations

float gSeriesData[SERIES_LENGTH]; // Array for floating pt data
int32_t i32DataCount = 0; // Iteration counter

int main(void) {
    float fRadians; // floating pt radians

    ROM_FPULazyStackingEnable(); // Enable lazy stacking
    ROM_FPUEnable(); // Enable FPU

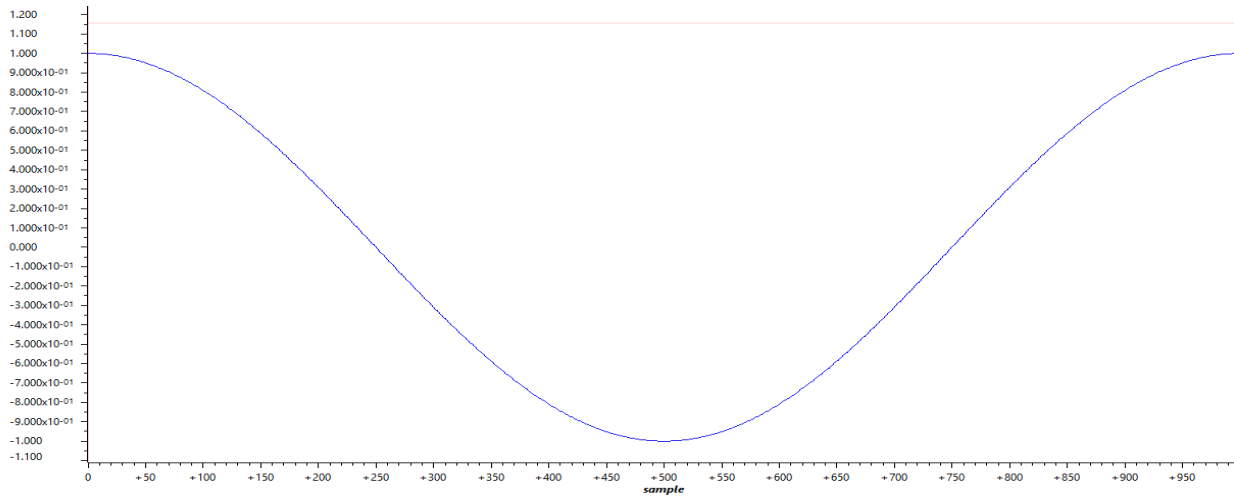
    // Using 50 MHz clock
    ROM_SysCtlClockSet(
        SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ
        | SYSCTL_OSC_MAIN);

    fRadians = ((2 * M_PI) / SERIES_LENGTH); // Compute radians

    // Calculate 1000 values
    while (i32DataCount < SERIES_LENGTH) {
        gSeriesData[i32DataCount] = cosf(fRadians * i32DataCount); // store sine wave data
        i32DataCount++; // go to next increment
    }
}
```

```
}  
    while (1)  
        ;  
}
```

Image of Cosine Wave



Video Link to Demo

NONE