

Argenis Jimenez Aguirre

CPE 403

TIVAC LAB 09

Task01: Submit a comprehensive commented file of the original code.

```
1 // Argenis Jimenez Aguirre
2 // CPE 403
3 // TIVAC LAB 09
4 // Task 01
5
6 #include <stdint.h>
7 #include <stdbool.h>
8 #include <math.h>
9 #include "inc/hw_memmap.h"
10 #include "inc/hw_types.h"
11 #include "driverlib/fpu.h"
12 #include "driverlib/sysctl.h"
13 #define TARGET_IS_BLIZZARD_RB1
14 #include "driverlib/rom.h"
15
16 // Define PI
17 #ifndef M_PI
18 #define M_PI 3.14159265358979323846
19 #endif
20
21 #define SERIES_LENGTH 100 // Depth of the data buffer
22
23 float gSeriesData[SERIES_LENGTH]; // Array of floats SERIES_LENGTH long
24
25 int32_t i32DataCount = 0; // Counter for the loop
26
27 int main(void)
28 {
29     float fRadians; // Variable to calculate sine
30
31     ROM_FPU_LazyStackingEnable(); // Turn on Lazy Stacking
32     ROM_FPU_Enable(); // Turn on FPU
33
34     // Set System Clock to 50MHz
35     ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);
36
37     // Divide 2*pi by array length since a full sine wave cycle is 2*pi radians
38     fRadians = ((2 * M_PI) / SERIES_LENGTH);
```

```

40 // Calculate 100 sine values
41 while(i32DataCount < SERIES_LENGTH)
42 {
43     // Store sine value into array
44     gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
45
46     // Increment to go to next value calculation
47     i32DataCount++;
48 }
49
50 while(1)
51 {
52 }
53 }

```

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

```

1 // Argenis Jimenez Aguirre
2 // CPE 403
3 // TIVAC LAB 09
4 // Task 02
5
6 #include <stdint.h>
7 #include <stdbool.h>
8 #include <math.h>
9 #include "inc/hw_memmap.h"
10 #include "inc/hw_types.h"
11 #include "driverlib/fpu.h"
12 #include "driverlib/sysctl.h"
13 #define TARGET_IS_BLIZZARD_RB1
14 #include "driverlib/rom.h"
15
16 // Define PI
17 #ifndef M_PI
18 #define M_PI 3.14159265358979323846
19 #endif
20
21 #define SERIES_LENGTH 1000 // Depth of the data buffer
22
23 float gSeriesData[SERIES_LENGTH]; // Array of floats SERIES_LENGTH long
24
25 int32_t i32DataCount = 0; // Counter for the loop
26
27 int main(void)
28 {
29     float fRadians; // Variable to calculate sine
30
31     ROM_FPUEnable(); // Turn on Lazy Stacking
32     ROM_FPUEnable(); // Turn on FPU
33
34     // Set System Clock to 50MHz
35     ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);
36
37     // Divide 2*pi by array length since a full sine wave cycle is 2*pi radians
38     fRadians = ((2 * M_PI) / SERIES_LENGTH);
39

```

```
40 // Calculate 1000 formula values
41 while(i32DataCount < SERIES_LENGTH)
42 {
43     // Store formula values into array
44     gSeriesData[i32DataCount] = sinf(fRadians * 50 * i32DataCount) + 0.5*cosf(fRadians * 200 * i32DataCount);
45
46     // Increment to go to next value calculation
47     i32DataCount++;
48 }
49
50 while(1)
51 {
52 }
53 }
54
```

---