Guillermo Gálvez

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)

Date Submitted: **10.01.2018**

Task 00: *No submission*

Youtube Link: N/A

Task 01: Determine Current period and on-time of the LED Blinking. Change delay of LED blink to ~0.425 sec by changing the delay and clock source and configuration-determine the CLK ƒ then verify delay

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

As given the code provides an on-time of ~0.150s and off-time of ~0.751s. This makes our period ~0.9001s


```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
uint8_t ui8PinData=2;
int main void)

    {//Setup clock
        SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
        SysCtlPeripheralEnable SYSCTL_PERIPH_GPIOF);
        GPIOPinTypeGPIOOutput GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
    }
    while 1)
    {
        GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, ui8PinData);
        SysCtlDelay 2000000 ;
        GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay 2000000 ;
        if ui8PinData==8   ui8PinData=2;  else  ui8PinData=ui8PinData*2;
    }

}//end main
```

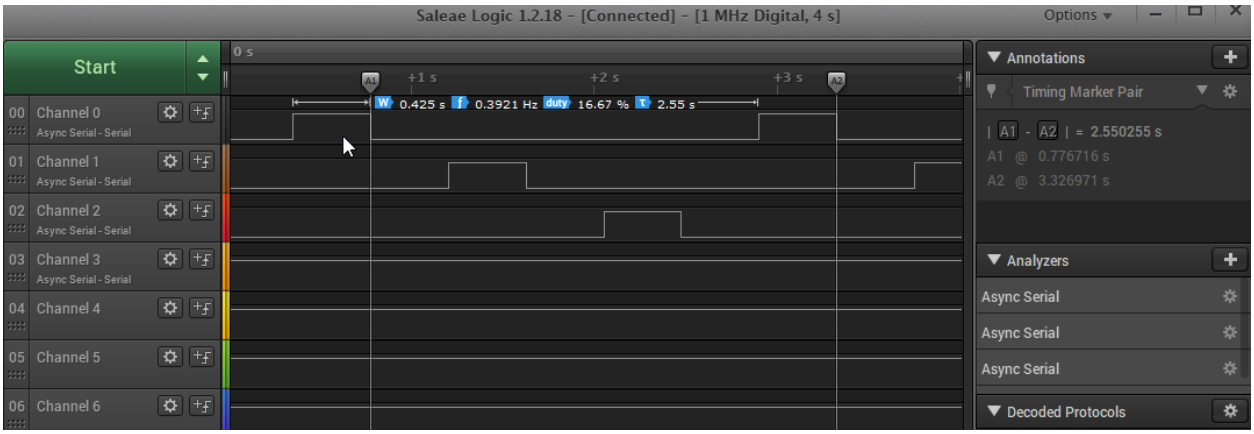
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Guillermo Gálvez

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)

Youtube Link: https://youtu.be/nA-Wb1xLf20

If we take the delay count of 2000000 and divided by the on-time we see that each tick is approximately 75ns. Therefore, if we want an on-time of 0.425s we divide 0.425s by 75ns and get a count of approximately 5.6…million. From the analyzer data we can see the period is ~2.55s.




```
while(1)
    {//Modified for 0.425s on-time
        GPIOPinWrite(GPIO_PORTF_BASE,PIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, ui8PinData);
        SysCtlDelay(5666666);
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(5666666);
        if(ui8PinData==8) {ui8PinData=2;} else {ui8PinData=ui8PinData*2;}
```

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

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)

Task 02: Change the a) sequence of LED blinking (from RGB sequence to BGR), and b) blink one LED, two LED, and three LED at an instance and with a sequence (sequence of blinking with delay – R, G, B, RG, RB, GB, RGB, R, G, …)

Youtube Link: https://youtu.be/81TU1GpTW0w

a)

```
uint8_t ui8PinData=4;    //Task 01 a. B

int main(void)
{
    {
 //Setup clock
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
        GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
    }

    while(1)
    {
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, ui8PinData);
        SysCtlDelay(5666666);
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(5666666);
        //First pass !8 -> 4*2 = 8 G, BG
        //Second pass is 8 -> 2 R, BGR
        if(ui8PinData==8) {ui8PinData=2;} else {ui8PinData=ui8PinData*2;}
    }

}//end main
```

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

Github root directory:(https://github.com/galveg1/VMs_House-of-Fun-Or-Pain.git)

Task 02:b

Youtube Link: https://youtu.be/UQHsSGh1QbI

```
//Initially set to red

uint8_t ui8PinData=2;   //Task 02 b. Red

int main void)
{
    {
 //Setup clock
SysCtlClockSet SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

        SysCtlPeripheralEnable SYSCTL_PERIPH_GPIOF);
        GPIOPinTypeGPIOOutput GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
    }

    while 1)
    {
        GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, ui8PinData); //Red
        SysCtlDelay 5666666);
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x08); //Green
        SysCtlDelay 5666666);
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x04); //Blue
        SysCtlDelay 5666666);
        GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x0A); //RG
        SysCtlDelay 5666666);
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x06); //RB
        SysCtlDelay 5666666 ;
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x0C); //GB
        SysCtlDelay 5666666);
        GPIOPinWrite GPIO_PORTF_BASE, GPIO_PIN_1 |GPIO_PIN_2|GPIO_PIN_3, 0x0E); //RGB
        SysCtlDelay 5666666);
    }

}//end main
```

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