

Task 00: Execute provided code

```
//
/* Melissa Cordova
 * CPE 403 - LAB 3
 *
 * Program that will show how to initialize the clock system and the GPIO
 * peripheral using
 * TivaWare. It will use the GPIO output to blink an LED
 * on the evaluation board.
 *
 * main.c
 */

#include <stdint.h>           //variable definitions for the C99 standard
#include <stdbool.h>          //boolean definitions for the C99 standard
#include "inc/hw_memmap.h"    //macros defining the memory map of the TivaC
#include "inc/hw_types.h"     //defines common types and macros
#include "driverlib/sysctl.h" //defines and macros for System Control API
#include "driverlib/gpio.h"   //Defines and macros for GPIO API of DriverLib

uint8_t ui8PinData=2;        //creates an integer variable called ui8PinData
                              //and initializes it to 2.

int main(void)
{
    //sets clock: xtal= 16MHz, 400MHz PLL divided by 10
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

    //enable the clock for peripheral
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    //configures the three GPIO pins connected to the LEDs
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
    while(1)
    {
        //turn on LED as specified in ui8PinData
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2| GPIO_PIN_3, ui8PinData);
        SysCtlDelay(2000000);           //loop timer provided in TivaWare
        //turn LEDs off
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(2000000);           //delay = 200000*3 = 600000 CPU cycles
        //set ui8PinData to the next LED color
        if(ui8PinData==8) {ui8PinData=2;} else {ui8PinData=ui8PinData*2;}
    }
}
```

Task 01: Determine the current period and on-time of the LED blinking. Change the delay of the LED blink (approx. 0.333 sec) – determine the CLK frequency – verify delay to be approx., 0.333 sec.

```
#include <stdint.h>           //variable definitions for the C99 standard
#include <stdbool.h>          //boolean definitions for the C99 standard
#include "inc/hw_memmap.h"    //macros defining the memory map of the TivaC
#include "inc/hw_types.h"     //defines common types and macros
#include "driverlib/sysctl.h" //defines and macros for System Control API
#include "driverlib/gpio.h"   //Defines and macros for GPIO API of DriverLib

uint8_t ui8PinData=2;        //creates an integer variable called ui8PinData
                              //and initializes it to 2.

int main(void)
{
    //2000000*3 CPU cycles = 6000000 CPU cycles
    //6000000/frequency = 0.333sec -> frequency = 6000000/0.333sec = 18.18MHZ
    //400MHz PLL divided by 22
    SysCtlClockSet(SYSCTL_SYSDIV_11|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

    //enable the clock for peripheral
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    //configures the three GPIO pins connected to the LEDs
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
    while(1)
    {
        //turn on LED as specified in ui8PinData
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2| GPIO_PIN_3, ui8PinData);
        SysCtlDelay(2000000); //loop timer provided in TivaWare
        //turn LEDs off
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(2000000); //delay = 200000*3 = 6000000 CPU cycles
        //set ui8PinData to the next LED color
        if(ui8PinData==8) {ui8PinData=2;} else {ui8PinData=ui8PinData*2;}
    }
}
```

Task 02: a) Change the sequence of LED blinking

```
#include <stdint.h>           //variable definitions for the C99 standard
#include <stdbool.h>          //boolean definitions for the C99 standard
#include "inc/hw_memmap.h"    //macros defining the memory map of the TivaC
#include "inc/hw_types.h"     //defines common types and macros
#include "driverlib/sysctl.h" //defines and macros for System Control API
#include "driverlib/gpio.h"   //Defines and macros for GPIO API of DriverLib

uint8_t ui8PinData=8;        //creates an integer variable called ui8PinData
                              //and initializes it to 2.

int main(void)
{
    //2000000*3 CPU cycles = 6000000 CPU cycles
```

```
//6000000/frequency = 0.333sec -> frequency = 6000000/0.333sec = 18.18MHz
//400MHz PLL divided by 22

SysCtlClockSet(SYSCTL_SYSDIV_11|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

//enable the clock for peripheral
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
//configures the three GPIO pins connected to the LEDs
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
while(1)
{
    //turn on LED as specified in ui8PinData
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2| GPIO_PIN_3, ui8PinData);
    SysCtlDelay(2000000); //loop timer provided in TivaWare
    //turn LEDs off
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
    SysCtlDelay(2000000); //delay = 200000*3 = 600000 CPU cycles
    //set ui8PinData to the next LED color
    if(ui8PinData==2) {ui8PinData=8;} else {ui8PinData=ui8PinData/2;}
}
}
```

Task 02: b) Blink two LED at an instance and with a sequence

```
#include <stdint.h> //variable definitions for the C99 standard
#include <stdbool.h> //boolean definitions for the C99 standard
#include "inc/hw_memmap.h" //macros defining the memory map of the TivaC
#include "inc/hw_types.h" //defines common types and macros
#include "driverlib/sysctl.h" //defines and macros for System Control API
#include "driverlib/gpio.h" //Defines and macros for GPIO API of DriverLib

uint8_t ui8PinData=6; //creates an integer variable called ui8PinData
//and initializes it to 2.

int main(void)
{
    //2000000*3 CPU cycles = 6000000 CPU cycles
    //6000000/frequency = 0.333sec -> frequency = 6000000/0.333sec = 18.18MHz
    //400MHz PLL divided by 22

    SysCtlClockSet(SYSCTL_SYSDIV_11|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

    //enable the clock for peripheral
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    //configures the three GPIO pins connected to the LEDs
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
    while(1)
    {
        //turn on LED as specified in ui8PinData
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2| GPIO_PIN_3, ui8PinData);
        SysCtlDelay(2000000); //loop timer provided in TivaWare
        //turn LEDs off
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(2000000); //delay = 200000*3 = 600000 CPU cycles
    }
}
```

```
//set ui8PinData to the next LED color
if(ui8PinData==6) {ui8PinData=10;} //0110 (red+blue=purple)
else if(ui8PinData==10) //1010 (green+red=yellow)
{ui8PinData=12;}
else {ui8PinData=6;} //1100 (blue+green=aqua)
}
```