

Github root directory: <https://github.com/Ayertena/AdvEmbeddedSys>

Date Submitted: 10/02/2018

Task 00: Execute the provided code, no submission is required.

Task 01: Determine the current period and on-time of the LED blinking. Change the delay of the LED blink (approx. 0.425 sec) by changing the delay and clock source and configuration –determine the CLK frequency – verify the delay to be approx. 0.425 sec.

Youtube Link: <https://youtu.be/WT208FMXqQc>

```
#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
Tiva C Series device.
// This includes defines such as peripheral base address locations
// such as GPIO_PORTF_BASE.
#include "inc/hw_types.h"     // Defines common types and macros
#include "driverlib/sysctl.h" // Defines and macros for System
Control API of DriverLib.
// This includes API functions such as SysCtlClockSet and
SysCtlClockGet.
#include "driverlib/gpio.h"   // Defines and macros for GPIO API of
DriverLib.
// This includes API functions such as GPIOPinTypePWM and
GPIOPinWrite.

uint8_t ui8PinData=2;
/*
Creates an integer variable called ui8PinData and initializes it to 2.
This will be used to cycle through the three LEDs, lighting them one
at a time.
Note that the C99 type is an 8-bit unsigned integer and that the
variable name reflects this. */

int main(void)
{
```

```
/*Delay calculation:
```

```
* F = 1/T
```

```
* 2000000*3*T = 0.425S
```

```
* T= .0.425/6000000 = 70.833nS
```

```
* F= 1/70.833nS = 14MHz approx *
```

```
* 14MHz = 400MHz/(2*14.12) so use SYSCTL_SYSDIV_11 divider
```

```
* 0.425s = 70.833ns * 3(2000000 cycles) */
```

```
SysCtlClockSet(SYSCTL_SYSDIV_14|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
```

```
/* Before calling any peripheral specific driverLib function, we must enable the clock for that peripheral. If you fail to do this, it will result in a Fault ISR (address fault). */
```

```
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); // loop timer provided in TivaWare.
```

```
    // The count parameter is the loop count, not the actual delay in clock cycles.
```

```
    // Each loop is 3 CPU cycles.
```

```
    // Turn off all three pins.
```

```
    GPIOPinWrite(GPIO_PORTF_BASE,  
GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
```

```
    SysCtlDelay(2000000); // Delay
```

```
    if(ui8PinData==8) {ui8PinData=2;} else  
{ui8PinData=ui8PinData*2;}
```

```
    }
```

```
}
```

Task 02_A: Change the sequence of LED blinking (from RGB sequence to BGR),

Youtube Link: <https://youtu.be/gazRrObqUh8>

```
#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
Tiva C Series device.
// This includes defines such as peripheral base address locations
// such as GPIO_PORTF_BASE.
#include "inc/hw_types.h"      // Defines common types and macros
#include "driverlib/sysctl.h"  // Defines and macros for System
Control API of DriverLib.
// This includes API functions such as SysCtlClockSet and
SysCtlClockGet.
#include "driverlib/gpio.h"    // Defines and macros for GPIO API of
DriverLib.
// This includes API functions such as GPIOPinTypePWM and
GPIOPinWrite.
```

```
uint8_t ui8PinData=2;
/* Creates an integer variable called ui8PinData and initializes it to
2.
This will be used to cycle through the three LEDs, lighting them one
at a time.
Note that the C99 type is an 8-bit unsigned integer and that the
variable name reflects this. */
```

```
int main(void)
{
    /*Delay calculation:
    * F = 1/T
    * 20000000*3*T = 0.425S
    * T= .0.425/60000000 = 70.833nS
    * F= 1/70.833nS = 14MHz approx *
    * 14MHz = 400MHz/(2*14.12) so use SYSCTL_SYSDIV_11 divider
    * 0.425s = 70.833ns * 3(2000000 cycles) */
```

```
SysCtlClockSet(SYSCTL_SYSDIV_14|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
```

```

    // Before calling any peripheral specific driverLib function, we
must enable the clock
    // for that peripheral. If you fail to do this, it will result in
a Fault ISR (address fault).
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE,
GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_1);

    while(1)
    {

        // Write the value '2' into all three pins
        GPIOPinWrite(GPIO_PORTF_BASE,
GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_1, ui8PinData);
        SysCtlDelay(2000000); // loop timer provided in TivaWare.
        // The count parameter is the loop count, not the actual delay
in clock cycles.
        // Each loop is 3 CPU cycles.

        // Turn off all three pins.
        GPIOPinWrite(GPIO_PORTF_BASE,
GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
        SysCtlDelay(2000000); // Delay

        if(ui8PinData==8) {ui8PinData=2;} else
{ui8PinData=ui8PinData*2;}
    }
}

```

Task 02_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/ew8NBES-suU>

```
#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 Tiva C Series device.
// This includes defines such as peripheral base address locations
// such as GPIO_PORTF_BASE.
#include "inc/hw_types.h"      // Defines common types and macros
#include "driverlib/sysctl.h"  // Defines and macros for System
Control API of DriverLib.
// This includes API functions such as SysCtlClockSet and
SysCtlClockGet.
#include "driverlib/gpio.h"    // Defines and macros for GPIO API of
DriverLib.
// This includes API functions such as GPIOPinTypePWM and
GPIOPinWrite.

uint8_t ui8PinData=2;
/* Creates an integer variable called ui8PinData and initializes it to
2.
This will be used to cycle through the three LEDs, lighting them one
at a time.
Note that the C99 type is an 8-bit unsigned integer and that the
variable name reflects this. */

int main(void)
{
    /*Delay calulation:
    * F = 1/T
    * 2000000*3*T = 0.425S
    * T= .0.425/6000000 = 70.833nS
    * F= 1/70.833nS = 14MHz approx *
    * 14MHz = 400MHz/(2*14.12) so use SYSCTL_SYSDIV_11 divider
    * 0.425s = 70.833ns * 3(2000000 cycles) */
```

```
SysCtlClockSet(SYSCTL_SYSDIV_14|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
```

```
// Before calling any peripheral specific driverLib function, we must enable the clock
```

```
// for that peripheral. If you fail to do this, it will result in a Fault ISR (address fault).
```

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

```
while(1)  
{
```

```
// Write the value '2' into all three pins  
GPIOPinWrite(GPIO_PORTF_BASE,  
GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, ui8PinData);  
SysCtlDelay(2000000); // loop timer provided in TivaWare.  
// The count parameter is the loop count, not the actual delay in clock cycles.  
// Each loop is 3 CPU cycles.
```

```
// Turn off all three pins.  
GPIOPinWrite(GPIO_PORTF_BASE,  
GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);  
SysCtlDelay(2000000); // Delay
```

```
// R = 2, G = 8, B = 4  
// RG = 10 (yellow), RB = 6 (purple) , GB = 12 (cyan) , RGB = 14 (white) , R, G, B
```

```
if(ui8PinData == 8 ) {  
    ui8PinData = ui8PinData / 2 ;}  
else {  
    ui8PinData = ui8PinData + 6;  
    if(ui8PinData == 16 ) {ui8PinData = 6;}  
    if(ui8PinData == 18 ) {ui8PinData = 14;}  
    if(ui8PinData == 20 ) {ui8PinData = 2;}  
    }  
}
```