**Task 0:**

Execute the provided code, no submission is required.

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

// stores LED pin data

uint8\_t ui8PinData=2;

**int** **main**(**void**)

{

// ==================================================

// Configure system clock

// CLK = 40MHz = (400MHz PLL / (5 \* 2))

// --------------------------------------------------

**SysCtlClockSet**(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);

// enable periphery for GPIO\_F and set pins 1 through 3 as output

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);

**while**(1)

{

// set LED's on depending on pin data

**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);

// cycle through R, G, and B pin data

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

}

}

**Task 1:**

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 –

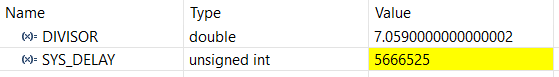
determine the CLK frequency – verify the delay to be approx. 0.425 sec.

Verifying the equation:

So, **SysCtlDelay**(**SysCtlClockGet**() / DIVISOR) =>

SysCtlDelay is approximately equal to **56.7x10^5** at a 40MHz clock.

Hence, **SysCtlDelay**(**SysCtlClockGet**() / 7.059);



**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

// stores LED pin data

uint8\_t ui8PinData=2;

**int** **main**(**void**)

{

// constant for clock delay

**double** DIVISOR = 7.059;

// ==================================================

// Configure system clock

// CLK = 40MHz = (400MHz PLL / (5 \* 2))

// --------------------------------------------------

**SysCtlClockSet**(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);

// enable periphery for GPIO\_F and set pins 1 through 3 as output

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);

**while**(1)

{

// set LED's on depending on pin data

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, ui8PinData);

/\*

\* DELAY: ((seconds) / ( (1/CLK) \* 3)

\*

\* For 0.425 seconds

\* ((0.425) / ( (1/40MHz) \* 3) = 56.7x10^5

\*

\* So close approximation gives

\* 40MHz / x = 56.7x10^5

\* x = 7.059 => DIVISOR

\*/

**SysCtlDelay**(**SysCtlClockGet**() / DIVISOR);

// debugging variable to confirm timing

uint32\_t SYS\_DELAY = **SysCtlClockGet**() / DIVISOR;

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0x00);

// delay to turn off LEDs

**SysCtlDelay**(**SysCtlClockGet**() / DIVISOR);

// cycle through R, G, and B pin data

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

}

}

**Task 2:**

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, …).

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/gpio.h"

// stores LED pin data

uint8\_t ui8PinData;

// LED array for asserting specific color configurations

// R, G, B, RG, RB, GB, RGB

uint8\_t array[7]={2,8,4,10,6,12,14};

// index counter

uint8\_t i = 0;

**int** **main**(**void**)

{

// constant for clock delay

**double** DIVISOR = 7.059;

// ==================================================

// Configure system clock

// CLK = 40MHz = (400MHz PLL / (5 \* 2))

// --------------------------------------------------

**SysCtlClockSet**(SYSCTL\_SYSDIV\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);

// enable periphery for GPIO\_F and set pins 1 through 3 as output

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF);

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3);

**while**(1)

{

// set LED's on depending on pin data

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1 | GPIO\_PIN\_2 | GPIO\_PIN\_3, ui8PinData);

/\*

\* DELAY: ((seconds) / ( (1/CLK) \* 3)

\*

\* For 0.425 seconds

\* ((0.425) / ( (1/40MHz) \* 3) = 56.7x10^5

\*

\* So close approximation gives

\* 40MHz / x = 56.7x10^5

\* x = 7.059 => DIVISOR

\*/

**SysCtlDelay**(**SysCtlClockGet**() / DIVISOR);

// debugging variable to confirm timing

uint32\_t SYS\_DELAY = **SysCtlClockGet**() / DIVISOR;

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0x00);

// delay to turn off LEDs

**SysCtlDelay**(**SysCtlClockGet**() / DIVISOR);

// assign pin values through predefined array for LED colors

ui8PinData = array[i];

// iterate through LED configuration array

i++;

// at the 7th index location, reset LED counter

**if**(i == 7)

i=0;

}

}