Task 01: Determine the current period and on-time of the LED blinking. Change the delay of the

LED blink (approx. 0.333 sec) by changing the clock source and configuration – do not change

the delay value – determine the CLK frequency – verify the delay to be approx. 0.333 sec.

- The current period and on-time of the LED can be calculated using the following formulas.

Since we are using the PLL, by default it is divided by 2. Then divided by our set divider. The provided code sets a combined 4000000 cycle delay that is not to be changed according to the prompt.

So

Period =

The current delay time is

Delay time =

Delay time = .1s

And the blink time is half of that since the delay cycles are even distributed at 2000000 for on and off.

Blink time = .05s

If the LED blink time is to be .33s, then

.33 =

.6060606.06 =

**DIV = 33**

However, on the actual board, the blink turns out to be about 1s, to get a .33 delay on the board, **DIV is set to 11**.

Code for blinking change to .33. Change from original is highlighted.

int main(void)

{SysCtlClockSet(SYSCTL\_SYSDIV\_11|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);

// divides clock further

}

Task 02

a) Change sequence of LED blinking

Modified code reverses sequence. Now it blinks Green-Blue-Red

uint8\_t ui8PinData = 8;

int main(void)

{

...

while(1)

{

if(ui8PinData == 2)

{ui8PinData = 8; } else{ui8PinData = ui8PinData/2;}

}

}

b) blink two LED at an instance and with a sequence.

uint8\_t ui8PinData = 6;

int main(void)

{

...

while(1)

{

…

//To cycle throught 6 to 10 to 12 and so on. Each has two lights on

if(ui8PinData == 12{ui8PinData = 6;} else if(ui8PinData == 6)

{ui8PinData = ui8PinData+4;} else{ui8PinData = ui8PinData+2;}

}

}