**Date Submitted: 10/2/2018**

**Task 00: Execute provided code**

**Youtube Link:** [**https://youtu.be/4ez2hilwOZA**](https://youtu.be/4ez2hilwOZA)

**------------------------------------------------------------------------------------**

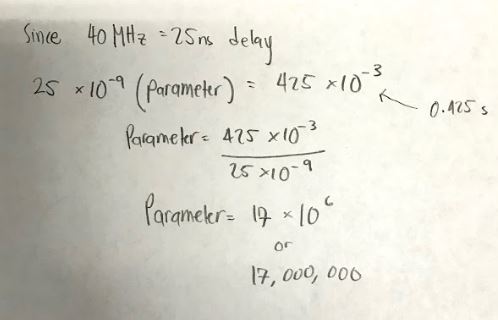
**Task 01:**

Youtube Link: <https://youtu.be/Vn32CGrduVA>

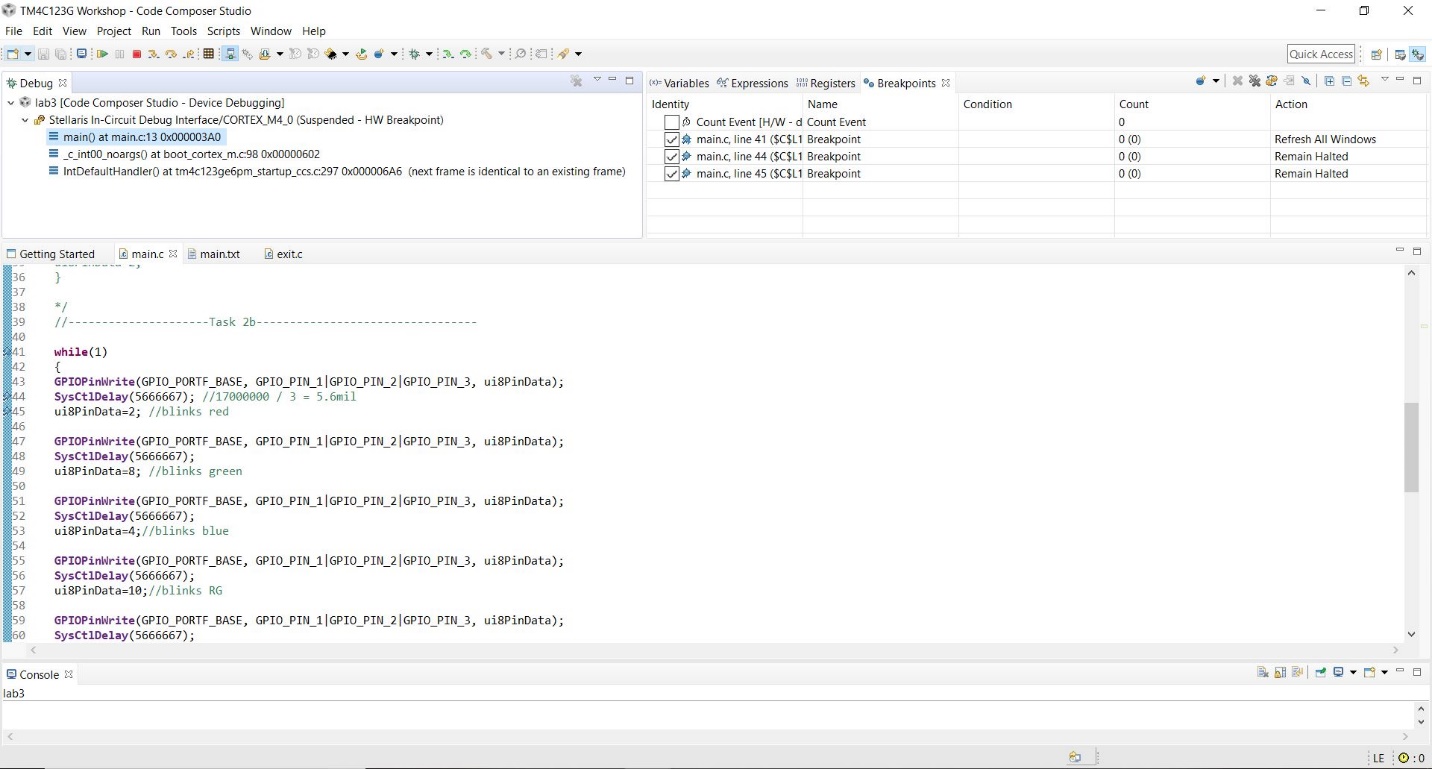
**Modified Schematic (if applicable): N/A**

**Notes:**

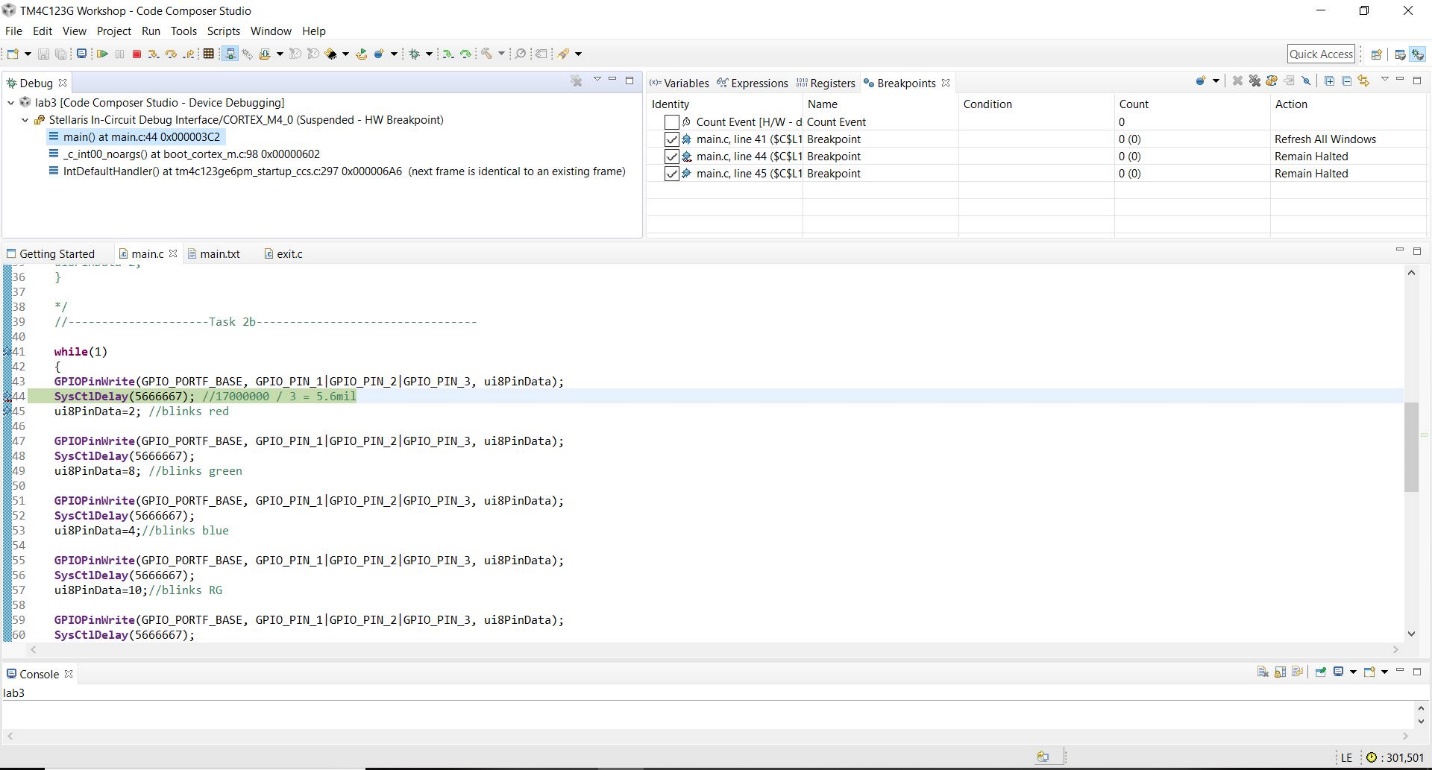
**For Task 1, I stuck with the original 40 MHz clock configuration. This was done by using a 16 MHz crystal to drive the 400MHz PLL. Since there is a default divider of 2 in the clock path we added another divider of 5 to achieve a system clock of 40MHz. Then I hand calculated the new parameter that would be passed into SysCtlDelay() which turns out to 17 million. The image below shows my hand calculations given the fact that our current clock gave us a delay of 25 ns.**

****

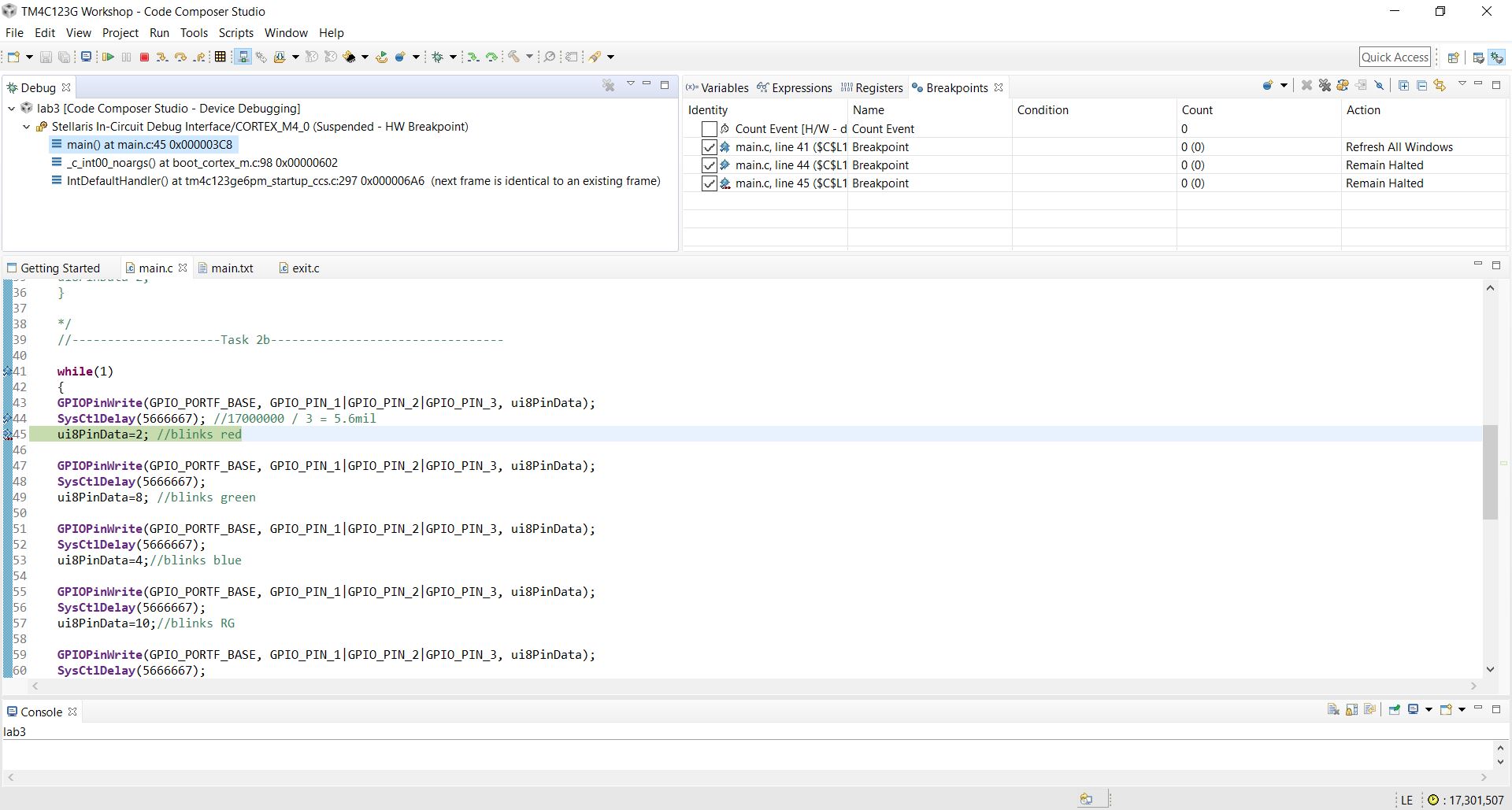
**Since the count parameter (in this case 17 million) is the loop count we must divide that number by 3 since each loop is 3 CPU cycles. This then gets us to roughly 5.666667 million. Theoretically this is the value that needs to be passed into SysCtlDelay() to achieve a 0.425s delay. The delay is then verified by the following images:**

****

**It might be a little hard to see but, on the bottom, right corner of the image above, we see a yellow clock symbol with the number 0 next to it. The zero signifies that the program is not currently running.**

****

**The image above is the result of clicking the resume button once, which stops right before the delay because of the breakpoint placed before the delay call. Now the value next to the yellow clock is 301, 501.**

****

**The image above is the result of clicking resume once more, which takes us to the line after the delay call. As we can see the value next to the yellow clock is now 17,301,507. This verifies our delay of 0.425s achieved properly using the value of 5.6666667 million as the value passed into SysCtlDelay().**

**Modified Code:**

**// Insert code here**

**#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; //this is for original

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

{

//sets a 40 MHz clock, 400MHz /10 = 40MHz

**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)

{

//Pin1 = red

//Pin2 = blue

//Pin3 = green

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

**SysCtlDelay**(5666667); //delay of 0.425s

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

**SysCtlDelay**(5666667); //delay of 0.425s

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

}

}

**------------------------------------------------------------------------------------**

**Task 02a:**

Youtube Link: <https://www.youtube.com/watch?v=HmU-y_DR6xg>

**Modified Schematic (if applicable): N/A**

**Modified Code:**

**// Insert code here**

**#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; //this is responsible for the color of the LED

//2 = red

//4 = Blue

//8 = green

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

{

//The line below sets a 40 MHz 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); //set pins as outputs

//--------------------------Task 2a----------------------------------------------

//25ns(x) = 425ms

//x = 17 x 10^6

**while**(1)

{

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

**SysCtlDelay**(5666667);

ui8PinData=4; //blinks Blue

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

**SysCtlDelay**(5666667);

ui8PinData=8; //blinks Green

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

**SysCtlDelay**(5666667);

ui8PinData=2; //blinks Red

}

}

**------------------------------------------------------------------------------------**

**Task 02b:**

Youtube Link: <https://www.youtube.com/watch?v=BPA2NghvboE>

**Modified Schematic (if applicable): N/A**

**Modified Code:**

**// Insert code here**

**#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; //this is responsible for the color of the LED

//2 = red

//4 = Blue

//8 = green

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

{

//The line below sets a 40 MHz 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); //set pins as outputs

**while**(1)

{

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

**SysCtlDelay**(5666667); //17000000 / 3 = 5.6mil

ui8PinData=2; //blinks red

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=8; //blinks green

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=4;//blinks blue

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=10;//blinks RG, add 8 and 2

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=6;//blinks RB add 4 and 2

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=12;//blinks GB add 4 and 8

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=14;//blinks white/RGB add 2,4 and 8

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=2; //blinks red

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

**SysCtlDelay**(5666667);//17000000 / 3 = 5.6mil

ui8PinData=8; //blinks green

}

}

**------------------------------------------------------------------------------------**