**Date Submitted: 9/27/2019**

**Task 00: Execute provided code**

**Youtube Link:** <https://www.youtube.com/watch?v=pQZ3GjdlXGg>

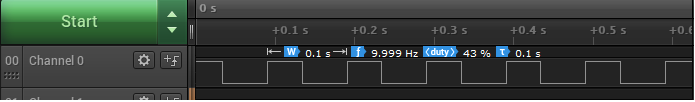
**------------------------------------------------------------------------------------**

**Task 01:**

Youtube Link: <https://www.youtube.com/watch?v=oXUyvd0Nq30>

**Modified Schematic (if applicable):**

**Waveform**



**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/tm4c123gh6pm.h" //definitions for interrupt and register assignments on Tiva C

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

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

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

**#include** "driverlib/interrupt.h" //Defines and macros for interrupts, functions -> IntEnable() and IntPrioritySet()

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

**#include** "driverlib/timer.h" //Defines and macros for timers, functions -> TimerConfigure(), TimerLoadSet()

//GLOBAL VARIABLES

uint32\_t timerDelayHigh; //variable to store delay for high time (43% of DC)

uint32\_t timerDelayLow; //variable to store delay for low time (57% of DC)

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

//Clock setup - using 40MHz clock

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

//GPIO setup

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF); //system control - enable peripheral PORTF

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3); //set PORTF pins 1-3 as OUTPUT

//Timer setup

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER0); //system control - enable peripheral TIMER0

**TimerConfigure**(TIMER0\_BASE, TIMER\_CFG\_PERIODIC); //32-bit timer configured in periodic mode

//timer0 = timer0A(16-bits) and timer0B(16-bits)

//Calculate Delay

timerDelayHigh = (**SysCtlClockGet**() / 10) \* .43; //delay that contributes to the 43% duty cycle ON time of LED

timerDelayLow = (**SysCtlClockGet**() / 10) \* .57; //delay that contributes to the OFF time of LED (57% low)

//initialize timer0A to begin with the delay time of when the LED is on

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, timerDelayHigh -1);

//Interrupt Enable

**IntEnable**(INT\_TIMER0A); //enables timer0A interrupt in the interrupt vector table

**TimerIntEnable**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT); //interrupt is triggered at TIMEOUT of timer0A

**IntMasterEnable**(); //enable ALL interrupts

//Timer Enable

**TimerEnable**(TIMER0\_BASE, TIMER\_A);

**while**(1) {

//wait for interrupt to occur...

}

}

**void** **Timer0IntHandler**(**void**)

{

// Clear the timer interrupt

**TimerIntClear**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

// Read the current state of the GPIO pin and

// write back the opposite state

**if**(**GPIOPinRead**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2)) //if GPIO\_PIN\_2 (blue LED) is on, turn LED off

{

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, timerDelayLow - 1); //delay for 57ms (57% of DC - OFF time)

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0); //turn off LED

}

**else** //if LED is off, turn LED on

{

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, timerDelayHigh - 1); //delay for 43ms (43% of DC - ON time)

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4); //turn on LED (blue)

}

}

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

**Task 02:**

Youtube Link: <https://www.youtube.com/watch?v=EYcv1a3vSK8>

**Modified Schematic (if applicable):**

**Modified Code:**

**// Insert code here**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/tm4c123gh6pm.h" //definitions for interrupt and register assignments on Tiva C

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

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

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

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

**#include** "driverlib/interrupt.h" //Defines and macros for interrupts, functions -> IntEnable() and IntPrioritySet()

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

**#include** "driverlib/timer.h" //Defines and macros for timers, functions -> TimerConfigure(), TimerLoadSet()

//GLOBAL VARIABLES

uint32\_t timerDelayHigh; //variable to store delay for high time (43% of DC)

uint32\_t timerDelayLow; //variable to store delay for low time (57% of DC)

//FUNCTION DECLARATION

**void** **timer1A\_delaySec**(**int** ttime) {

//Defined variables are found in "inc/tm4c123gh6pm.h" header file

//This function is used to delay for a number of seconds for timer 1

**int** i; //used to traverse through for loop

SYSCTL\_RCGCTIMER\_R |= 2; // enable clock to Timer Block 1

TIMER1\_CTL\_R = 0; // disable timer before initialization

TIMER1\_CFG\_R = 0x04; // 16-bit option

TIMER1\_TAMR\_R = 0x02; // periodic mode and down-counter

TIMER1\_TAILR\_R = 64000 - 1; // TimerA interval load value reg

TIMER1\_TAPR\_R = 250 - 1; // TimerA prescaler 16MHz/250 = 64000Hz

TIMER1\_ICR\_R = 0x1; // clear the TimerA timeout flag

TIMER1\_CTL\_R |= 0x01; // enable TimerA after initialization

**for** (i=0; i < ttime; i++) {

**while** ((TIMER1\_RIS\_R & 0x1) == 0); // wait for TimerA timeout flag

TIMER1\_ICR\_R = 0x1; // clear the TimerA timeout flag

}

}

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

//Clock setup - using 40MHz clock

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

//GPIO setup

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOF); //system control - enable peripheral PORTF

**GPIOPinTypeGPIOOutput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3); //set PORTF pins 1-3 as OUTPUT

//Timer0 setup

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER0); //system control - enable peripheral TIMER0

**TimerConfigure**(TIMER0\_BASE, TIMER\_CFG\_PERIODIC); //32-bit timer configured in periodic mode

//timer0 = timer0A(16-bits) and timer0B(16-bits)

//Calculate Delay

timerDelayHigh = (**SysCtlClockGet**() / 10) \* .43; //delay that contributes to the 43% duty cycle ON time of LED

timerDelayLow = (**SysCtlClockGet**() / 10) \* .57; //delay that contributes to the OFF time of LED (57% low)

//initialize timer0A to begin with the delay time of when the LED is on

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, timerDelayHigh -1);

//Interrupt Enable

**IntEnable**(INT\_TIMER0A); //enables timer0A interrupt in the interrupt vector table

**TimerIntEnable**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT); //interrupt is triggered at TIMEOUT of timer0A

**IntMasterEnable**(); //enable ALL interrupts

//Timer Enable

**TimerEnable**(TIMER0\_BASE, TIMER\_A);

//UNLOCK PIN 0 (SW2)

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= GPIO\_PIN\_0;

//GPIO INTERRUPT SETUP

//Set SW2 (GPIO\_PIN\_0) as input and pad configuration

**GPIOPinTypeGPIOInput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_0); //enable SW2 to be an input

**GPIOPadConfigSet**(GPIO\_PORTF\_BASE, GPIO\_PIN\_0, GPIO\_STRENGTH\_2MA, GPIO\_PIN\_TYPE\_STD\_WPU);

//Pin 0 will drive around 2mA of current and has pad type of "push-pull with weak pull-up"

//Setup GPIO interrupt

**GPIOIntEnable**(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0); //enable interrupt for pin 0 (SW2)

**GPIOIntTypeSet**(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0, GPIO\_RISING\_EDGE); //trigger interrupt on rising edge

**IntEnable**(INT\_GPIOF); //enable PORTF interrupt on vector table

//TIMER1 SETUP

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER1);

**TimerConfigure**(TIMER1\_BASE, TIMER\_CFG\_PERIODIC);

**TimerEnable**(TIMER1\_BASE, TIMER\_A);

**while**(1) {

//wait for interrupt to occur...

}

}

**void** **Timer0IntHandler**(**void**)

{

// Clear the timer interrupt

**TimerIntClear**(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

// Read the current state of the GPIO pin and

// write back the opposite state

**if**(**GPIOPinRead**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2)) //if GPIO\_PIN\_2 (blue LED) is on, turn LED off

{

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, timerDelayLow - 1); //delay for 57ms (57% of DC - OFF time)

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0); //turn off LED

}

**else** //if LED is off, turn LED on

{

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, timerDelayHigh - 1); //delay for 43ms (43% of DC - ON time)

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4); //turn on LED (blue)

}

}

**void** **PortFPin0IntHandler**(**void**) {

//Interrupt Subroutine for Pin 0 (SW2)

//clear interrupt for GPIO

**GPIOIntClear**(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_0);

//turn on LED

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4);

//perform delay of 1s using timer1 - call timer1A delay function

timer1A\_delaySec(1); //delay for 1s

//turn off LED

**GPIOPinWrite**(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 0);

}

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