**Date Submitted: 09/28/19**

**Task 00: Execute provided code**

**Youtube Link:** <https://www.youtube.com/watch?v=JQNdKt8KZ5E&list=PLLbVEP8QAFUHqPdF1HjlqL0xlbC-GU_OS&index=2&t=0s>

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

**Task 01:**

Youtube Link: <https://www.youtube.com/watch?v=S-yEugaIvrI&list=PLLbVEP8QAFUHqPdF1HjlqL0xlbC-GU_OS&index=2>

**Modified Schematic (if applicable):**

**Modified Code:**

**// Insert code here**

#include <stdint.h>

#include <stdbool.h>

#include "inc/tm4c123gh6pm.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/sysctl.h"

#include "driverlib/interrupt.h"

#include "driverlib/gpio.h"

#include "driverlib/timer.h"

uint32\_t ui32PeriodHigh, ui32PeriodLow;

int main(void)

{

//uint32\_t ui32PeriodHigh, ui32PeriodLow;

//Clock Setup

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

//GPIO Configuration

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOB); //Used to test the output for waveform analysis

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

GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_0); //Used to test the output for waveform analysis

//Timer Configuration

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_PERIODIC);

//Calculate Delay

ui32PeriodHigh = 0.43\*(SysCtlClockGet()/1)/10;

ui32PeriodLow = 0.57\*(SysCtlClockGet()/1)/10;

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodHigh -1);

//Interrupt Enable

IntEnable(INT\_TIMER0A);

TimerIntEnable(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

IntMasterEnable();

//Timer Enable

TimerEnable(TIMER0\_BASE, TIMER\_A);

//Main Loop

while(1)

{

}

}

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))

{

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodLow); //loads in low

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

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 0); //Used to test the output for waveform analysis

}

else

{

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodHigh); //loads in high

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

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); //Used to test the output for waveform analysis

}

}

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

**Task 02:**

Youtube Link: <https://www.youtube.com/watch?v=chBtXA3loJE&list=PLLbVEP8QAFUHqPdF1HjlqL0xlbC-GU_OS&index=3>

**Modified Schematic (if applicable):**

**Modified Code:**

#include <stdint.h>

#include <stdbool.h>

#include "inc/tm4c123gh6pm.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/sysctl.h"

#include "driverlib/interrupt.h"

#include "driverlib/gpio.h"

#include "driverlib/timer.h"

void timer1A\_delaySec(int);

uint32\_t ui32PeriodHigh, ui32PeriodLow;

int main(void)

{

//uint32\_t ui32PeriodHigh, ui32PeriodLow;

uint32\_t ui32PeriodDelay;

//Clock Setup

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

//GPIO Configuration

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOB); //Used to test the output for waveform analysis

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

//-Switch -------------------

GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE, GPIO\_PIN\_4);

GPIOPadConfigSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_4, GPIO\_STRENGTH\_2MA, GPIO\_PIN\_TYPE\_STD\_WPU);

GPIOIntEnable(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_4);

GPIOIntTypeSet(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_4, GPIO\_RISING\_EDGE);

IntEnable(INT\_GPIOF);

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

GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_0); //Used to test the output for waveform analysis

GPIOPinTypeGPIOOutput(GPIO\_PORTB\_BASE, GPIO\_PIN\_1); //Used to test the output for waveform analysis

//Timer Configuration

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_PERIODIC);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER1);

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

//Calculate Delay

ui32PeriodHigh = 0.43\*(SysCtlClockGet()/1)/10;

ui32PeriodLow = 0.57\*(SysCtlClockGet()/1)/10;

//ui32PeriodDelay = SysCtlClockGet(); //1 second delay

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodHigh -1);

//Interrupt Enable

IntEnable(INT\_TIMER0A);

TimerIntEnable(TIMER0\_BASE, TIMER\_TIMA\_TIMEOUT);

IntMasterEnable();

//Timer Enable

TimerEnable(TIMER0\_BASE, TIMER\_A);

//Main Loop

while(1)

{

//

}

}

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))

{

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodLow); //loads in low

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

// GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 0); //Used to test the output for waveform analysis

}

else

{

TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodHigh); //loads in high

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

// GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); //Used to test the output for waveform analysis

}

}

void timer1A\_delaySec(int ttime)

// Used for delay using timer 1

{

int i;

SYSCTL\_RCGCTIMER\_R |= 2;

TIMER1\_CTL\_R = 0;

TIMER1\_CFG\_R = 0x04;

TIMER1\_TAMR\_R = 0x02;

TIMER1\_TAILR\_R = 64000 - 1;

TIMER1\_TAPR\_R = 250 - 1;

TIMER1\_ICR\_R = 0x1;

TIMER1\_CTL\_R |= 0x01;

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

{

while ((TIMER1\_RIS\_R & 0x1) == 0);

TIMER1\_ICR\_R = 0x1;

}

}

void SW1Int(void)

// Once button (switch 1) is pressed, it will check to see if it's pressed, if true then it will keep the blue LED on for 1 second

{

int pressed = 0;

pressed = GPIOIntStatus(GPIO\_PORTF\_BASE, true);

GPIOIntClear(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_4);

if (status & GPIO\_INT\_PIN\_4)

{

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

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 1); //Used to test the output for waveform analysis

timer1A\_delaySec(2);

}

GPIOPinWrite(GPIO\_PORTB\_BASE, GPIO\_PIN\_0, 0); //Used to test the output for waveform analysis

SysCtlDelay(1000000);

}

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