**Date Submitted: 9/28/2019**

**Task 00: Execute provided code**

**Youtube Link: https://youtu.be/ofXYd0N8hbg**

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

**Task 01:**

Youtube Link: https://youtu.be/-WVT88S6aKE

**A screenshot of a social media post

Description automatically generated**

**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"

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

{

**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);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER0);

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

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, 0); //this will get overwritten in interrupt

**IntEnable**(INT\_TIMER0A);

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

**IntMasterEnable**();

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

**while**(1)

{

}

}

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

{

uint32\_t ui32PeriodH;

uint32\_t ui32PeriodL;

ui32PeriodH = (**SysCtlClockGet**() / 10) \* .43; //toggle GPIO high @ 10Hz & 43% duty cycle

ui32PeriodL = (**SysCtlClockGet**() / 10) \* .57; //toggle GPIO low @ 10Hz & 57% duty cycle

// 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))

{

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

}

**else**

{

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

}

}

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

**Task 02:**

Youtube Link: https://youtu.be/MWt9\_DR-aWE

**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"

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

{

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

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

// GPIO CONFIG + INTERRUPT CONFIG

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

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

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

**GPIOPinTypeGPIOInput**(GPIO\_PORTF\_BASE, GPIO\_PIN\_4); //switch configuration

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

**GPIOIntEnable**(GPIO\_PORTF\_BASE,GPIO\_PIN\_4); //interrupt due to activity on pin 4

**GPIOIntTypeSet**(GPIO\_PORTF\_BASE,GPIO\_INT\_PIN\_4, GPIO\_RISING\_EDGE); //set interrupt

**IntEnable**(INT\_GPIOF);

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

// TIMER0 CONFIG

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

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER0);

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

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, 0);

**IntEnable**(INT\_TIMER0A);

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

**IntMasterEnable**();

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

**while**(1)

{

}

}

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

{

uint32\_t ui32PeriodH;

uint32\_t ui32PeriodL;

ui32PeriodH = (**SysCtlClockGet**() / 10) \* .43; //toggle GPIO @ 10Hz & 43% duty cycle

ui32PeriodL = (**SysCtlClockGet**() / 10) \* .57; //toggle GPIO @ 10Hz & 43% duty cycle

// 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, ui32PeriodL);

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

}

**else**

{

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, ui32PeriodH);

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

}

}

**void** **Timer1IntHandler**(**void**)

{

//Clear the timer interrupt

**TimerIntClear**(TIMER1\_BASE, TIMER\_TIMA\_TIMEOUT);

**TimerLoadSet**(TIMER0\_BASE, TIMER\_A, **SysCtlClockGet**()); //stop timer 0 for 1 second

**GPIOPinWrite**(GPIO\_PORTF\_BASE,GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 14);//white 1s

}

**void** **Pin4InterruptHandler**(**void**)

{

**GPIOIntClear**(GPIO\_PORTF\_BASE, GPIO\_PIN\_4); //Clear the timer interrupt

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_TIMER1); //Enable timer 0

**IntMasterEnable**();

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

**TimerLoadSet**(TIMER1\_BASE, TIMER\_A, 0);

**IntEnable**(INT\_TIMER1A);

**TimerIntEnable**(TIMER1\_BASE, TIMER\_TIMA\_TIMEOUT);

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

}

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

**Task 03:**

Youtube Link:

**Modified Schematic (if applicable):**

**Modified Code:**

**// Insert code here**

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