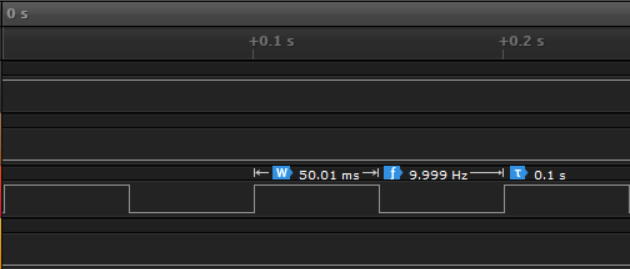
**Date Submitted: 10/20/19**

**Task 00: Execute provided code**

Youtube Link:<https://youtu.be/opduzr5G9yU>

****

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

**Task 01:**

Youtube Link: <https://youtu.be/baHZctpp0s8>

**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)**

**{**

**uint32\_t ui32Period;**

**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);**

**ui32Period = (SysCtlClockGet() / 10) /2;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32Period -1);**

**IntEnable(INT\_TIMER0A);**

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

**IntMasterEnable();**

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

**while(1)**

**{**

**}**

**}**

**void Timer0IntHandler(void)**

**{**

**uint32\_t ui32PeriodOn;**

**uint32\_t ui32PeriodOff;**

**// 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))**

**{**

**ui32PeriodOff = (SysCtlClockGet() / 10) \*.57;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodOff -1);**

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

**}**

**else**

**{**

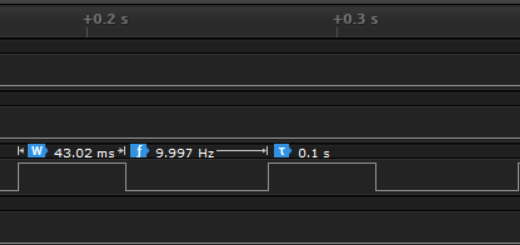
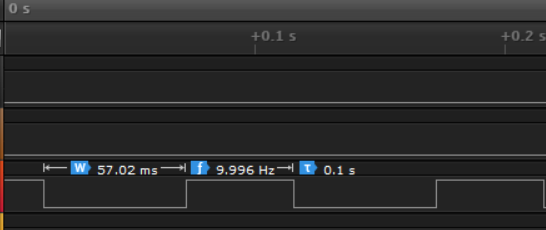
**ui32PeriodOn = (SysCtlClockGet() / 10) \*.43;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodOn -1);**

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

**}**

**}**

****

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

**Task 02:**

Youtube Link: <https://youtu.be/3mACettFDCA>

**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"**

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

**int main(void)**

**{**

**uint32\_t ui32Period;**

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

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

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

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

**GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);**

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

**IntEnable(INT\_GPIOF);**

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

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

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

**ui32Period = (SysCtlClockGet() / 10) \*.57;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32Period -1);**

**IntEnable(INT\_TIMER0A);**

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

**IntMasterEnable();**

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

**while(1)**

**{**

**}**

**}**

**void PortFIntHandler(){**

**int status=0; //checks if button is pressed**

**status = GPIOIntStatus(GPIO\_PORTF\_BASE,true);**

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

**if(status & GPIO\_INT\_PIN\_0)**

**{**

**//Then there was a Button pin interrupt**

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

**Timer1ADelay(5); //Delays for 1 second**

**}**

**/\***

**This delay is for deboucing but since it's in a interrupt it**

**should be used a better method that is faster**

**\*/**

**SysCtlDelay(100000);**

**}**

**void Timer1ADelay(int timer) //Delays timer for .5 seconds**

**{**

**int i;**

**SYSCTL\_RCGCTIMER\_R |= 2; //Enable Timer1.**

**TIMER1\_CTL\_R = 0; //Disable Timer1A during setup.**

**TIMER1\_CFG\_R = 0x04; //Configure to 16-bit mode.**

**TIMER1\_TAMR\_R = 0x02; //Configure for periodic mode, default down.**

**TIMER1\_TAILR\_R = 64000-1; //Period=64000. Reload Value.**

**TIMER1\_TAPR\_R = 625-1; //Bus Clock Resolution.**

**TIMER1\_ICR\_R = 0x1; //Clear Timer1A timeout flag.**

**TIMER1\_CTL\_R |= 0x1; //Enable Timer1A.**

**for(i = 0; i < timer; i++)**

**{**

**while((TIMER1\_RIS\_R &0x01) == 0); //Timeout wait.**

**TIMER1\_ICR\_R = 0x1; //Flag clear.**

**}**

**}**

**void Timer0IntHandler(void)**

**{**

**uint32\_t ui32PeriodOn;**

**uint32\_t ui32PeriodOff;**

**// 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))**

**{**

**ui32PeriodOff = (SysCtlClockGet() / 10) \*.57;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodOff -1);**

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

**}**

**else**

**{**

**ui32PeriodOn = (SysCtlClockGet() / 10) \*.43;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32PeriodOn -1);**

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

**}**

**}**