**Date Submitted: 9/27/19**

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

**Task 01:**

Youtube Link: https://www.youtube.com/watch?v=g-GprItkVHA

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

**Modified Code:**

**int main(void)**

**{**

**uint32\_t ui32Period;**

**//Set Clock to 40MHz**

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

**//Enable PortF**

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

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

**//Enable Timer0A**

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

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

**//Set timer to 10Hz 43% duty cycle**

**ui32Period = (43\*SysCtlClockGet()/10) / 100;**

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

**//Enable timer interrupt**

**IntEnable(INT\_TIMER0A);**

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

**IntMasterEnable();**

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

**while(1)**

**{**

**}**

**}**

**void Timer0IntHandler(void)**

**{**

**uint32\_t ui32Period2;**

**// 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))**

**{**

**//Turn off LED and set 57% duty cycle**

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

**ui32Period2 = (57\*SysCtlClockGet()/10) / 100;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32Period2 -1);**

**}**

**else**

**{**

**//Turn on LED and set 43% duty cycle**

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

**ui32Period2 = (43\*SysCtlClockGet()/10) / 100;**

**TimerLoadSet(TIMER0\_BASE, TIMER\_A, ui32Period2 -1);**

**}**

**}**

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

**Task 02:**

Youtube Link: https://www.youtube.com/watch?v=-HwVGsCC02s

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

**Modified Code:**

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

**{**

**int i;**

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

**TIMER1\_CTL\_R = 0; //Disable Timer before initialization**

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

**TIMER1\_TAMR\_R = 0x02; //Period mode and down counter**

**TIMER1\_TAILR\_R = 40000-1;//TimerA interval load value reg**

**TIMER1\_TAPR\_R = 250-1; //TimerA Prescaler 20MHz/250 = 80000Hz**

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

**TIMER1\_CTL\_R |= 0x01; //Enable Timer A after initialization**

**for(i = 0; i < ttime\*2; i++) //Every Loop = 1 second**

**{**

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

**{**

**//wait for timer timeout**

**}**

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

**}**

**}**

**int main(void)**

**{**

**//uint32\_t period;**

**//Set Clock to 20MHz**

**SysCtlClockSet(SYSCTL\_SYSDIV\_10|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);**

**//SysCtlDelay(3);**

**//period = SysCtlClockGet();**

**//Enable PortF pin 4 for interrupt with weak pullup**

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

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

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

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

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

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

**//Clear LEDs initially**

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

**IntEnable(INT\_GPIOF);**

**while(1)**

**{**

**//wait for button to be pressed**

**}**

**}**

**void PortFPin4IntHandler(void)**

**{**

**// Clear the GPIO interrupt**

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

**SysCtlDelay(3);**

**// Disable the Interrupt**

**IntDisable(INT\_GPIOF);**

**GPIOIntDisable(GPIO\_PORTF\_BASE, GPIO\_INT\_PIN\_4);**

**// Turn on the LED**

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

**timer1A\_delaySec(1); //call delay function for timer1A**

**//turn off LED**

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

**//Enable the interrupt**

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

**IntEnable(INT\_GPIOF);**

**}**

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