**Date Submitted: September 27, 2019**

**Task 00: Execute provided code**

**Youtube Link:** https://youtu.be/pBW98WdJ8ZE

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

**Task 01:**

Youtube Link: https://youtu.be/1hP1qyyGREA

**Modified Schematic (if applicable):**

**Modified Code:**

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/debug.h"

#include "driverlib/sysctl.h"

#include "driverlib/adc.h"

//#define TARGET\_IS\_BLIZZARD\_RB1

//#include "driverlib/rom.h"

#include "driverlib/gpio.h"

//#ifdef DEBUG

//void\_\_error\_\_(char \*pcFilename, uint32\_t u132lLine)

//{

//}

//#endif

int main(void)

{

uint32\_t ui32ADC0Value[4]; //ADC Sequencer SS2 has FIFO depth of 4 so we need an array of the same size

volatile uint32\_t ui32TempAvg;

volatile uint32\_t ui32TempValueC;

volatile uint32\_t ui32TempValueF;

//LED Config

uint8\_t ui8LED1 = 2;

uint8\_t ui8LED2 = 4;

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

SysCtlDelay(3);

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

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

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);

ADCHardwareOversampleConfigure(ADC0\_BASE, 64);

ADCSequenceConfigure(ADC0\_BASE, 2, ADC\_TRIGGER\_PROCESSOR, 0); //We want to use ADC0, sampler sequencer 2, we want the processor to trigger the sequence and we want the highest priority

ADCSequenceStepConfigure(ADC0\_BASE, 2, 0, ADC\_CTL\_TS);

ADCSequenceStepConfigure(ADC0\_BASE, 2, 1, ADC\_CTL\_TS);

ADCSequenceStepConfigure(ADC0\_BASE, 2, 2, ADC\_CTL\_TS);

ADCSequenceStepConfigure(ADC0\_BASE, 2, 3, ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);

ADCSequenceEnable(ADC0\_BASE, 2);

while(1)

{

ADCIntClear(ADC0\_BASE, 2);

ADCProcessorTrigger(ADC0\_BASE, 2);

while(!ADCIntStatus(ADC0\_BASE, 2, false))

{

}

ADCSequenceDataGet(ADC0\_BASE, 2, ui32ADC0Value);

ui32TempAvg = (ui32ADC0Value[0] + ui32ADC0Value[1] + ui32ADC0Value[2] + ui32ADC0Value[3] + 2)/4;

ui32TempValueC = (1475 - ((2475 \* ui32TempAvg)) / 4096)/10;

ui32TempValueF = ((ui32TempValueC \* 9) + 160) / 5;

if(ui32TempValueF > 71) {

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, ui8LED2);

}

else {

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, ui8LED1);

}

}

}

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

**Task 02:**

Youtube Link: https://youtu.be/FuIStNpNuZc

**Modified Schematic (if applicable):**

**Modified Code:**

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "inc/tm4c123gh6pm.h"

#include "driverlib/debug.h"

#include "driverlib/sysctl.h"

#include "driverlib/adc.h"

#include "driverlib/gpio.h"

#include "driverlib/timer.h"

#include "driverlib/interrupt.h"

int main(void)

{

uint32\_t ui32Period;

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

//LED Config

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

SysCtlDelay(3);

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

//ADC Config

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);

ADCHardwareOversampleConfigure(ADC0\_BASE, 32); //Hardware averages 32

ADCSequenceConfigure(ADC0\_BASE, 2, ADC\_TRIGGER\_PROCESSOR, 0); //We want to use ADC0, sampler sequencer 2, we want the processor to trigger the sequence and we want the highest priority

ADCSequenceStepConfigure(ADC0\_BASE, 2, 0, ADC\_CTL\_TS);

ADCSequenceStepConfigure(ADC0\_BASE, 2, 1, ADC\_CTL\_TS);

ADCSequenceStepConfigure(ADC0\_BASE, 2, 2, ADC\_CTL\_TS);

ADCSequenceStepConfigure(ADC0\_BASE, 2, 3, ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);

ADCSequenceEnable(ADC0\_BASE, 2);

//Timer1A Config

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER1);

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

ui32Period = (SysCtlClockGet() / 1) / 2; //To get period of 0.5s, divide system clock for 1s period and divide by 2 for 0.5s

TimerLoadSet(TIMER1\_BASE, TIMER\_A, ui32Period -1);

IntEnable(INT\_TIMER1A);

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

IntMasterEnable();

TimerEnable(TIMER1\_BASE, TIMER\_A);

while(1)

{

}

}

void Timer1IntHandler(void)

{

uint32\_t ui32ADC0Value[4]; //ADC Sequencer SS2 has FIFO depth of 4 so we need an array of the same size

volatile uint32\_t ui32TempAvg;

volatile uint32\_t ui32TempValueC;

volatile uint32\_t ui32TempValueF;

// Clear the timer interrupt

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

ADCIntClear(ADC0\_BASE, 2);

ADCProcessorTrigger(ADC0\_BASE, 2);

while(!ADCIntStatus(ADC0\_BASE, 2, false))

{

}

ADCSequenceDataGet(ADC0\_BASE, 2, ui32ADC0Value);

ui32TempAvg = (ui32ADC0Value[0] + ui32ADC0Value[1] + ui32ADC0Value[2] + ui32ADC0Value[3] + 2)/4;

ui32TempValueC = (1475 - ((2475 \* ui32TempAvg)) / 4096)/10;

ui32TempValueF = ((ui32TempValueC \* 9) + 160) / 5;

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);

}

}

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