**Date Submitted: 10/01/19**

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

**Task 01:**

Youtube Link: <https://www.youtube.com/watch?v=GM712P_yRxs>

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

**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"**

**#include "driverlib/gpio.h"**

**#ifdef DEGUB**

**void\_\_error\_\_(char \*pcFilename, uint32\_t ui32Line)**

**{**

**}**

**#endif**

**uint8\_t ui8PinData=4;**

**int main(void)**

**{**

**uint32\_t ui32ADC0Value[4];**

**volatile uint32\_t ui32TempAvg;**

**volatile uint32\_t ui32TempValueC;**

**volatile uint32\_t ui32TempValueF;**

**//Set Clock to 40MHz**

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

**//Enable ADC0 peripheral**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);**

**//Enable PortF**

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

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

**GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0x00); //All off**

**//Configure ADC sequencer 1**

**ADCSequenceConfigure(ADC0\_BASE, 1, ADC\_TRIGGER\_PROCESSOR, 0);**

**//Sample Internal temp sensor with sequencer 1**

**ADCSequenceStepConfigure(ADC0\_BASE, 1, 0, ADC\_CTL\_TS);**

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

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

**//Sample temp sensor, set interrupt flag to end conversion, enable ADC**

**ADCSequenceStepConfigure(ADC0\_BASE,1,3,ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);**

**ADCSequenceEnable(ADC0\_BASE, 1);**

**while(1)**

**{**

**//Clear interrupt flag**

**ADCIntClear(ADC0\_BASE, 1);**

**//Trigger ADC conversion with software**

**ADCProcessorTrigger(ADC0\_BASE, 1);**

**//wait for ADC conversion to finish**

**while(!ADCIntStatus(ADC0\_BASE, 1, false))**

**{**

**}**

**//Get ADC values from SS1**

**ADCSequenceDataGet(ADC0\_BASE, 1, ui32ADC0Value);**

**//Average and Calculate Temperature**

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

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

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

**//turn on Blue LED if temp is > 72F**

**if(ui32TempValueF > 72)**

**GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, ui8PinData);**

**else**

**GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0x0);**

**}**

**}**

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

**Task 02:**

Youtube Link: <https://www.youtube.com/watch?v=kN5DgYvDVT0>

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

**Modified Code:**

**#include <stdint.h>**

**#include <stdbool.h>**

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

**#include "inc/tm4c123gh6pm.h"**

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

**#include "driverlib/debug.h"**

**#include "driverlib/sysctl.h"**

**#include "driverlib/adc.h"**

**#include "driverlib/gpio.h"**

**#include "driverlib/interrupt.h"**

**#include "driverlib/timer.h"**

**#ifdef DEGUB**

**void\_\_error\_\_(char \*pcFilename, uint32\_t ui32Line)**

**{**

**}**

**#endif**

**uint8\_t ui8PinData=4;**

**uint32\_t ui32ADC0Value[4];**

**volatile uint32\_t ui32TempAvg;**

**volatile uint32\_t ui32TempValueC;**

**volatile uint32\_t ui32TempValueF;**

**int main(void)**

**{**

**uint32\_t ui32Period;**

**//Set Clock to 40MHz**

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

**//Enable ADC0 peripheral and hardware averaging**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);**

**ADCHardwareOversampleConfigure(ADC0\_BASE, 32);**

**//Enable PortF**

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

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

**GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0x00); //All off**

**//Enable Timer1A**

**SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER1);**

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

**//Configure ADC sequencer 1**

**ADCSequenceConfigure(ADC0\_BASE, 1, ADC\_TRIGGER\_PROCESSOR, 0);**

**//Sample Internal temp sensor with sequencer 1**

**ADCSequenceStepConfigure(ADC0\_BASE, 1, 0, ADC\_CTL\_TS);**

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

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

**//Sample temp sensor, set interrupt flag to end conversion, enable ADC**

**ADCSequenceStepConfigure(ADC0\_BASE,1,3,ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);**

**ADCSequenceEnable(ADC0\_BASE, 1);**

**//Set timer to 2Hz**

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

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

**//Enable timer interrupt**

**IntEnable(INT\_TIMER1A);**

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

**IntMasterEnable();**

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

**while(1)**

**{**

**//wait for timer interrupt every .5 seconds for ADC conversion**

**}**

**}**

**void Timer1IntHandler(void)**

**{**

**uint32\_t ui32Period2;**

**// Clear the timer interrupt**

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

**//Clear interrupt flag**

**ADCIntClear(ADC0\_BASE, 1);**

**//Trigger ADC conversion with software**

**ADCProcessorTrigger(ADC0\_BASE, 1);**

**//wait for ADC conversion to finish**

**while(!ADCIntStatus(ADC0\_BASE, 1, false))**

**{**

**}**

**//Get ADC values from SS1**

**ADCSequenceDataGet(ADC0\_BASE, 1, ui32ADC0Value);**

**//Average and Calculate Temperature**

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

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

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

**if(ui32TempValueF > 72)**

**GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, ui8PinData);**

**else**

**GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3, 0x0);**

**ui32Period2 = SysCtlClockGet()/ 2;**

**TimerLoadSet(TIMER1\_BASE, TIMER\_A, ui32Period2 -1);**

**}**

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