**Date Submitted: 10/02/19**

**Task 00: Execute provided code**

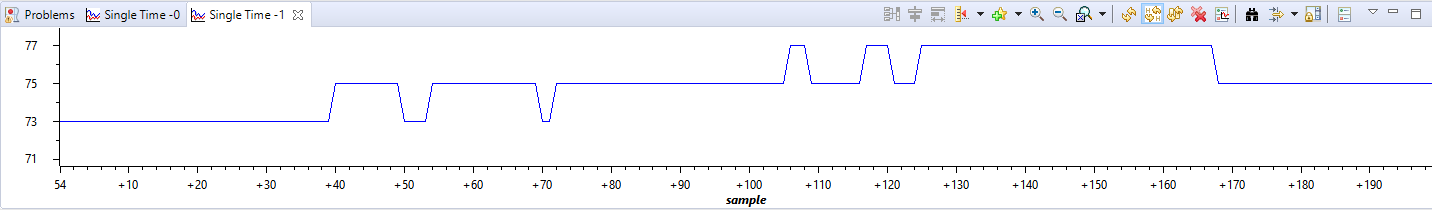
**Youtube Link:** <https://www.youtube.com/watch?v=TG88nQ4juU0&list=PLLbVEP8QAFUEmvgduFjYXjML-kptveWqB&index=2&t=0s>

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

**Task 01:**

Youtube Link: <https://www.youtube.com/watch?v=GBcSx1FBuhs&list=PLLbVEP8QAFUEmvgduFjYXjML-kptveWqB&index=2>

**Built-in graph:**



**After breathing on my temperature sensor, I’ve obtained these results…**

**Modified Schematic (if applicable):**

**Modified Code:**

**// Insert code here**

#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"

int main(void)

{

uint32\_t ui32ADC0Value[4];

volatile uint32\_t ui32TempAvg;

volatile uint32\_t ui32TempValueC;

volatile uint32\_t ui32TempValueF;

//clock initialization

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

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);

//GPIO settings

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

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

//ADC settings

ADCHardwareOversampleConfigure(ADC0\_BASE, 64);

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

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

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

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

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

ADCSequenceEnable(ADC0\_BASE, 1);

while(1)

{

ADCIntClear(ADC0\_BASE, 1); //clears ADC

ADCProcessorTrigger(ADC0\_BASE, 1);

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

{

}

ADCSequenceDataGet(ADC0\_BASE, 1, ui32ADC0Value);

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

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

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

//checks if temp is greater than 72

if(ui32TempValueF > 72)

{

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4); //turn led on when greater than 72

}

else

{

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 0); //turn led off when less than 72

}

}

}

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

**Task 02:**

Youtube Link: <https://www.youtube.com/watch?v=IG-gTDzRfrI&list=PLLbVEP8QAFUEmvgduFjYXjML-kptveWqB&index=3>

**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"

#define TARGET\_IS\_BLIZZARD\_RB1

#include "driverlib/rom.h"

#include "driverlib/gpio.h"

#include "driverlib/interrupt.h"

#include "driverlib/timer.h"

//Global Variables

uint32\_t ui32ADC0Value[4];

uint32\_t ui32Period;

volatile uint32\_t ui32TempAvg;

volatile uint32\_t ui32TempValueC;

volatile uint32\_t ui32TempValueF;

int main(void)

{

//clock initialization

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

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);

//GPIO settings

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

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

//ADC settings

ADCHardwareOversampleConfigure(ADC0\_BASE, 32); //changed from 64 to 32

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

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

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

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

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

ADCSequenceEnable(ADC0\_BASE, 1);

ADCIntEnable(ADC0\_BASE,2);

//Timer 1 settings

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER1);

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

ui32Period = (SysCtlClockGet()/2);

IntEnable(INT\_TIMER1A);

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

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

IntMasterEnable();

TimerEnable(TIMER1\_BASE, TIMER\_A);

while(1)

{

//main loop

}

}

void Timer1IntHandler(void)

{

//Clears the timer1 interrupt

TimerIntClear(TIMER1\_BASE, TIMER\_A);

ADCIntClear(ADC0\_BASE, 1); //clears ADC

ADCProcessorTrigger(ADC0\_BASE, 1);

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

{

}

ADCSequenceDataGet(ADC0\_BASE, 1, ui32ADC0Value);

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

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

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

//checks if temp is greater than 72

if(ui32TempValueF > 72)

{

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 4); //turn led on when greater than 72

}

else

{

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_2, 0); //turn led off when less than 72

}

}

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