**10/9/2018:**

**Task 00: Execute provided code**

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

**Task 01:**

Youtube Link: https://www.youtube.com/watch?v=XMVn6frWZVY

**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/interrupt.h"**

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

**#define TARGET\_IS\_BLIZZARD\_RB1**

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

**volatile uint32\_t ui32TempAvg;**

**volatile uint32\_t ui32TempValueC;**

**volatile uint32\_t ui32TempValueF;**

**int main(void)**

**{**

**uint32\_t ui32Period;**

**// set clock**

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

**// enable ADC and Port F**

**ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);**

**ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);**

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

**ROM\_ADCHardwareOversampleConfigure(ADC0\_BASE, 32);**

**// enable timer1**

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

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

**// configure ADC**

**ROM\_ADCSequenceConfigure(ADC0\_BASE, 2, ADC\_TRIGGER\_PROCESSOR, 0);**

**ROM\_ADCSequenceStepConfigure(ADC0\_BASE, 2, 0, ADC\_CTL\_TS);**

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

**ROM\_ADCSequenceStepConfigure(ADC0\_BASE, 2, 2, ADC\_CTL\_TS);**

**ROM\_ADCSequenceStepConfigure(ADC0\_BASE,2,3,ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);**

**ROM\_ADCSequenceEnable(ADC0\_BASE, 2);**

**//calculate period**

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

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

**// enable timer interrupt**

**IntEnable(INT\_TIMER1A);**

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

**IntMasterEnable();**

**// start timer1**

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

**// turn off LED**

**ROM\_GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 0);**

**while(1)**

**{**

**}**

**}**

**void Timer1IntHandler(void)**

**{**

**// Clear the timer interrupt**

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

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

**// clear ADC and then start it**

**ROM\_ADCIntClear(ADC0\_BASE, 2);**

**ROM\_ADCProcessorTrigger(ADC0\_BASE, 2);**

**// wait until ADC is finished**

**while(!ROM\_ADCIntStatus(ADC0\_BASE, 2, false))**

**{**

**}**

**// grab ADC data**

**ROM\_ADCSequenceDataGet(ADC0\_BASE, 2, ui32ADC0Value);**

**// calculate Temp**

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

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

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

**// check temp and update LED**

**if(ui32TempValueF>=72) {**

**ROM\_GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 8);**

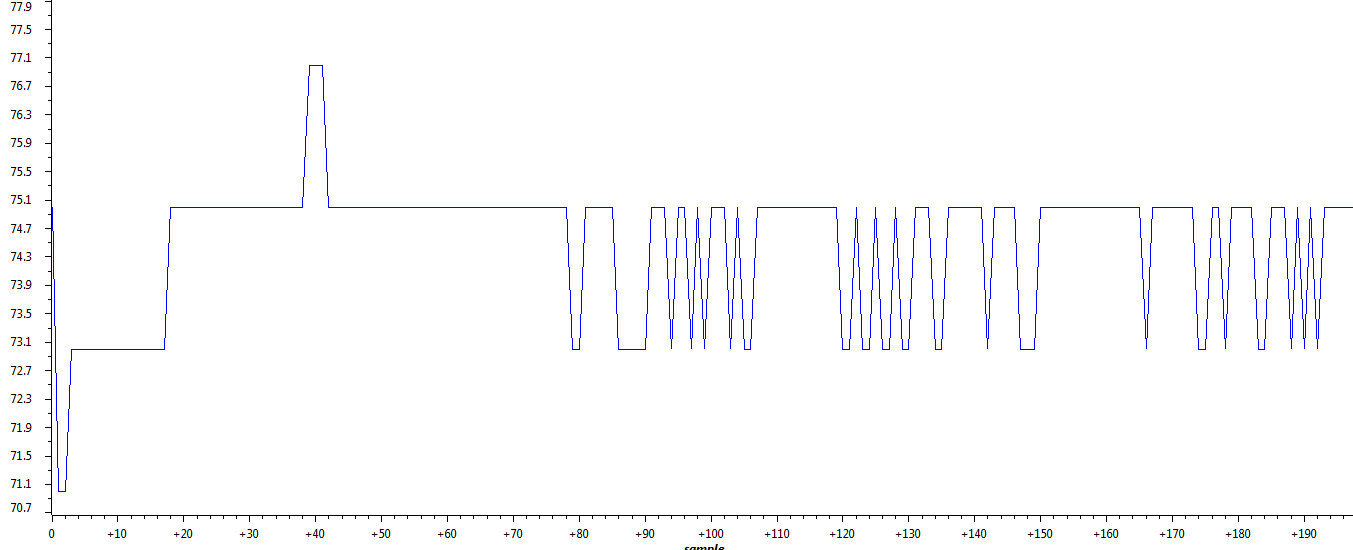
**} else {**

**ROM\_GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 0);**

**}**

**}**

**Temp Graph**

****

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

**Task 02:**

Youtube Link: https://www.youtube.com/watch?v=k9oGZZJ\_4dI

**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/interrupt.h"**

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

**#define TARGET\_IS\_BLIZZARD\_RB1**

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

**volatile uint32\_t ui32TempAvg;**

**volatile uint32\_t ui32TempValueC;**

**volatile uint32\_t ui32TempValueF;**

**int main(void)**

**{**

**uint32\_t ui32Period;**

**// set clock**

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

**// enable ADC and Port F**

**ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);**

**ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);**

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

**ROM\_ADCHardwareOversampleConfigure(ADC0\_BASE, 32);**

**// enable timer1**

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

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

**// configure ADC**

**ROM\_ADCSequenceConfigure(ADC0\_BASE, 2, ADC\_TRIGGER\_PROCESSOR, 0);**

**ROM\_ADCSequenceStepConfigure(ADC0\_BASE, 2, 0, ADC\_CTL\_TS);**

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

**ROM\_ADCSequenceStepConfigure(ADC0\_BASE, 2, 2, ADC\_CTL\_TS);**

**ROM\_ADCSequenceStepConfigure(ADC0\_BASE,2,3,ADC\_CTL\_TS|ADC\_CTL\_IE|ADC\_CTL\_END);**

**ROM\_ADCSequenceEnable(ADC0\_BASE, 2);**

**//calculate period**

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

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

**// enable timer interrupt**

**IntEnable(INT\_TIMER1A);**

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

**IntMasterEnable();**

**// start timer1**

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

**// turn off LED**

**ROM\_GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 0);**

**while(1)**

**{**

**}**

**}**

**void Timer1IntHandler(void)**

**{**

**// Clear the timer interrupt**

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

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

**// clear ADC and then start it**

**ROM\_ADCIntClear(ADC0\_BASE, 2);**

**ROM\_ADCProcessorTrigger(ADC0\_BASE, 2);**

**// wait until ADC is finished**

**while(!ROM\_ADCIntStatus(ADC0\_BASE, 2, false))**

**{**

**}**

**// grab ADC data**

**ROM\_ADCSequenceDataGet(ADC0\_BASE, 2, ui32ADC0Value);**

**// calculate Temp**

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

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

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

**// check temp and update LED**

**if(ui32TempValueF>=72) {**

**ROM\_GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 8);**

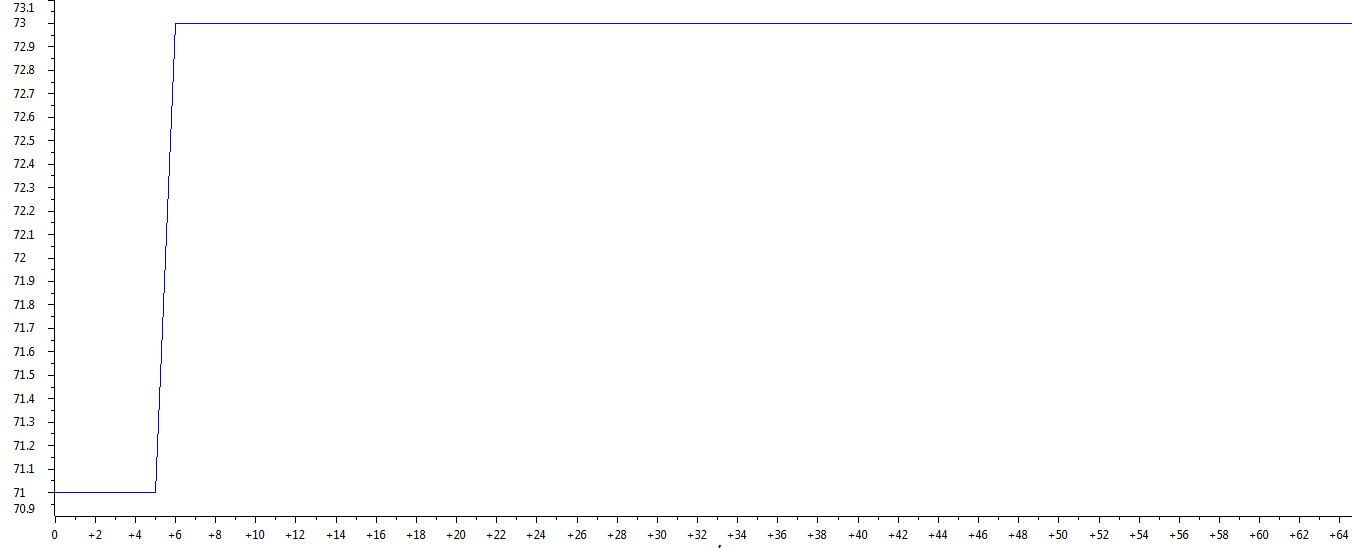
**} else {**

**ROM\_GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_3, 0);**

**}**

**}**

**Temp Graph**

****

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