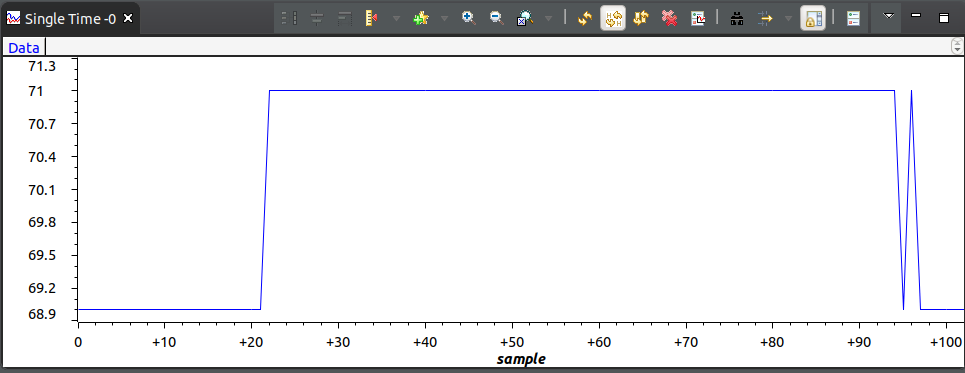
**Date Submitted:10/25/2019**

**Task 01:**



**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"

#define TARGET\_IS\_BLIZZARD\_RB1

#include "driverlib/rom.h"

int main(void)

{

uint32\_t ui32ADC0Value[4];

volatile uint32\_t ui32TempAvg;

volatile uint32\_t ui32TempValueC;

volatile uint32\_t ui32TempValueF;

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

//Configuring LED

ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

ROM\_GPIOPinTypeGPIOOutput(GPIO\_PORTF\_BASE,GPIO\_PIN\_2);

ROM\_SysCtlPeripheralEnable(SYSCTL\_PERIPH\_ADC0);

ROM\_ADCHardwareOversampleConfigure(ADC0\_BASE, 64);

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

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

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

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

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

ROM\_ADCSequenceEnable(ADC0\_BASE, 1);

while(1)

{

ROM\_ADCIntClear(ADC0\_BASE, 1);

ROM\_ADCProcessorTrigger(ADC0\_BASE, 1);

while(!ROM\_ADCIntStatus(ADC0\_BASE, 1, false))

{

}

ROM\_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;

if(ui32TempValueF>72){

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

}

else{

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

}

}

}

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

**Task 02:**

Youtube Link: <https://youtu.be/WYWZZcQ7GyM>

**#include <stdint.h>**

**#include <stdbool.h>**

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

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

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

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

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

**//uint32\_t ui32period;**

**void configureTimer1A(void){**

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

**IntMasterEnable();**

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

**//ui32period = (SysCtlClockGet()/2);**

**TimerLoadSet(TIMER1\_BASE, TIMER\_A, 20000000-1);**

**IntEnable(INT\_TIMER1A);**

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

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

**}**

**int main(void)**

**{**

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

**//Configuring LED**

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

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

**configureTimer1A();**

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

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

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

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

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

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

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

**ROM\_ADCSequenceEnable(ADC0\_BASE, 1);**

**while(1){**

**}**

**}**

**void Timer1IntHandler(void){**

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

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

**volatile uint32\_t ui32TempAvg;**

**volatile uint32\_t ui32TempValueC;**

**volatile uint32\_t ui32TempValueF;**

**ROM\_ADCIntClear(ADC0\_BASE, 1);**

**ROM\_ADCProcessorTrigger(ADC0\_BASE, 1);**

**while(!ROM\_ADCIntStatus(ADC0\_BASE, 1, false)){**

**}**

**ROM\_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;**

**if(ui32TempValueF>72){**

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

**}**

**else{**

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

**}**

**}**