Real-Time and Embedded Systems Design – Lab 6 Report Submission

Team 17

Eman Khaled 18P9713

Omar Hussien 18P1265

Farah Essam 18P3448

Mohamed Mostafa 18P9474

Shehab El Din Adel 18P3863

**Video link:** <https://drive.google.com/file/d/1QSkjDwMbQIF801iArG0tzad4wJaD5P51/view?usp=sharing>

#include <stdint.h>

#include "tm4c123gh6pm.h"

#include "FreeRTOS.h"

#include "task.h"

#include "timers.h"

#include "queue.h"

# define CLK\_FREQ  16000000

#define DELAY\_DEBOUNCE  CLK\_FREQ/1000

#define Get\_Bit(Register, Bit) (Register & ( 1 << Bit )) >> Bit

void Delay(int count);

void Init (void);

xTaskHandle uart0task;

void UART0Txchar(char a) ;

static void vSender1Task( void \*pvParameters );

static void vSender2Task( void \*pvParameters );

static void vReceiverTask( void \*pvParameters );

xQueueHandle xQueue;

static char counter = 0;

int main( void )

{

    Init();

    xQueue = xQueueCreate( 5, sizeof( int ) );

    if( xQueue != NULL )

    {

        xTaskCreate( vSender1Task, (const portCHAR \*)"Sender1", configMINIMAL\_STACK\_SIZE, NULL, 1, NULL );

        xTaskCreate( vSender2Task, (const portCHAR \*)"Sender2", configMINIMAL\_STACK\_SIZE, NULL, 1, NULL );

        xTaskCreate( vReceiverTask, (const portCHAR \*)"Receiver", configMINIMAL\_STACK\_SIZE, NULL, 2, &uart0task );

        vTaskStartScheduler();

    }

    else

    {

    }

    for( ;; );

}

/\*-----------------------------------------------------------\*/

static void vSender1Task( void \*pvParameters )

{

    for( ;; )

    {

        if(((   GPIO\_PORTF\_DATA\_R >>4) &1)==0)

        {

            do

            {

                Delay(20);

            }

            while(((    GPIO\_PORTF\_DATA\_R >>4) &1)==1);

            counter++;

            while(((    GPIO\_PORTF\_DATA\_R >>4) &1)==0);

            do

            {

                Delay(20);

            }

            while(((    GPIO\_PORTF\_DATA\_R >>4) &1)==0);

            taskYIELD();

        }

    }

}

static void vSender2Task( void \*pvParameters )

{

    portBASE\_TYPE xStatus;

    unsigned portBASE\_TYPE uxPriority;

    uxPriority=uxTaskPriorityGet(NULL);

    for( ;; )

    {

        if(((   GPIO\_PORTF\_DATA\_R >>0) &1)==0)

        {

            do

            {

                Delay(20);

            }

            while(((    GPIO\_PORTF\_DATA\_R >>0) &1)==1);

            vTaskPrioritySet(uart0task,(uxPriority-1));

            xStatus = xQueueSendToBack( xQueue, &counter, 0 );

            counter=0;

            vTaskPrioritySet(uart0task,(uxPriority+1));

            while(((    GPIO\_PORTF\_DATA\_R >>0) &1)==0);

            do

            {

                Delay(20);

            }

            while(((    GPIO\_PORTF\_DATA\_R >>0) &1)==0);

            taskYIELD();

        }

    }

}

/\*-----------------------------------------------------------\*/

static void vReceiverTask( void \*pvParameters )

{

  char lReceivedValue=0;

    portBASE\_TYPE xStatus;

    const portTickType xTicksToWait = 100000 / portTICK\_RATE\_MS;

    for( ;; )

    {

        xStatus = xQueueReceive( xQueue, &lReceivedValue, xTicksToWait );

        UART0Txchar(lReceivedValue);

    }

}

void Init(void){

        SYSCTL\_RCGCGPIO\_R |= 0X20;

        GPIO\_PORTF\_DIR\_R |= 0X0E;

        GPIO\_PORTF\_LOCK\_R = 0x4C4F434B;

        GPIO\_PORTF\_CR\_R = 0x1F;

        GPIO\_PORTF\_PUR\_R = 0x11;

        GPIO\_PORTF\_DEN\_R = 0x1F;

        SYSCTL\_RCGCUART\_R |= 0X01;

        SYSCTL\_RCGCGPIO\_R |= 0X01;

        UART0\_CTL\_R = 0;         /\* UART0 module disbable \*/

    UART0\_IBRD\_R = 65;      /\* for 9600 baud rate, integer = 104 \*/

        UART0\_FBRD\_R = 7;       /\* for 9600 baud rate, fractional = 11\*/

    UART0\_CC\_R = 0;          /\*select system clock\*/

    UART0\_LCRH\_R = 0x60;     /\* data lenght 8-bit, not parity bit, no FIFO \*/

    UART0\_CTL\_R = 0x301;        /\* Enable UART5 module, Rx and Tx \*/

        GPIO\_PORTA\_AFSEL\_R = (1<<1)|(1<<0);

        GPIO\_PORTA\_PCTL\_R = (1<<0)|(1<<4);

        GPIO\_PORTA\_DEN\_R = (1<<0)|(1<<1);

}

void UART0Txchar(char a)

{

    while((UART0\_FR\_R & (1<<5)) != 0);

    UART0\_DR\_R = a;

    if (a==1)

    {

        GPIO\_PORTF\_DATA\_R ^=0x3;

    }

}

void Delay(int count)

{

    int i,j;

  for(i = 0; i<count; i++)

        for(j = 0; j<3180; j++);

}