TivaC Lab 12 - UART

CPE 403

Checklist for Lab 12

- ☑ *A text/word document of the initial code with comments*

- ☑ *In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also include the comments.*

- ☑ *Provide a permanent link to all main and dependent source code files only (name them as LabXX-TYY, XX-Lab# and YY-task#)Screenshots of debugging process along with pictures of actual circuit*

- ☑ *Video link of demonstration.*

Code for Experiment

Task 1:

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/pin_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"

void UARTIntHandler(void) {
        uint32_t ui32Status;

        ui32Status = UARTIntStatus(UART0_BASE, true); //get interrupt status

        UARTIntClear(UART0_BASE, ui32Status); //clear interrupts

        while (UARTCharsAvail(UART0_BASE)) //loop while chars are detected
        {
                UARTCharPutNonBlocking(UART0_BASE, UARTCharGetNonBlocking(UART0_BASE)); //print character
                GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2); //blink LED
                SysCtlDelay(SysCtlClockGet() / (1000 * 3)); //delay ~1 msec
                GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0); //turn off LED
        }
}

int main(void) {

        SysCtlClockSet(
                        SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN
                                        | SYSCTL_XTAL_16MHZ);

        SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

        // Initialize UART
        GPIOPinConfigure(GPIO_PA0_U0RX);
        GPIOPinConfigure(GPIO_PA1_U0TX);
        GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); //enable GPIO port for LED
        GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2); //enable pin for LED PF2

        UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 115200,
                        (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE));

        IntMasterEnable(); //enable processor interrupts
        IntEnable(INT_UART0); //enable the UART interrupt
        UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT); //only enable RX and TX interrupts

        UARTCharPut(UART0_BASE, 'E');
        UARTCharPut(UART0_BASE, 'n');
        UARTCharPut(UART0_BASE, 't');
        UARTCharPut(UART0_BASE, 'e');
        UARTCharPut(UART0_BASE, 'r');
        UARTCharPut(UART0_BASE, ' ');
        UARTCharPut(UART0_BASE, 'T');
        UARTCharPut(UART0_BASE, 'e');
        UARTCharPut(UART0_BASE, 'x');
        UARTCharPut(UART0_BASE, 't');
        UARTCharPut(UART0_BASE, ':');
        UARTCharPut(UART0_BASE, ' ');

        while (1) //let interrupt handler do the UART echo function
        {
                if (UARTCharsAvail(UART0_BASE))
                        UARTCharPut(UART0_BASE, UARTCharGet(UART0_BASE));
        }

}
```

Task 2:

```
#include <stdint.h>
#include <stdbool.h>
#include <string.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/pin_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"

void UARTIntHandler();

int main(void) {
        char *prompt = "Enter Text: ";
        int i;

        SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN | SYSCTL_XTAL_16MHZ);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

    GPIOPinConfigure(GPIO_PA0_U0RX);
    GPIOPinConfigure(GPIO_PA1_U0TX);
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); //enable GPIO port for LED
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2); //enable pin for LED PF2

    UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 115200,
        (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE));

    IntMasterEnable(); //enable processor interrupts
    IntEnable(INT_UART0); //enable the UART interrupt
    UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT); //only enable RX and TX interrupts

    for (i = 0; prompt[i] != '\0'; i++)  // Print prompt at start of program
        UARTCharPut(UART0_BASE, prompt[i]);
    UARTCharPut(UART0_BASE, ' ');

    while(1); //let interrupt handler do the UART echo function
}

void UARTIntHandler() {
    uint32_t ui32Status;
    char buffer;

    ui32Status = UARTIntStatus(UART0_BASE, true); //get interrupt status

    UARTIntClear(UART0_BASE, ui32Status); //clear the asserted interrupts

    while(UARTCharsAvail(UART0_BASE)) //loop while there are chars
    {
        buffer = UARTCharGetNonBlocking(UART0_BASE);
        if (isalpha(buffer))
                if (buffer < 91)
                        buffer += 32;
                else
                        buffer -= 32;
        UARTCharPutNonBlocking(UART0_BASE, buffer); //echo character
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2); //blink LED
        SysCtlDelay(SysCtlClockGet() / (1000 * 3)); //delay ~1 msec
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0); //turn off LED
    }
}
```

Video Link to Demo

Task 1: https://www.youtube.com/watch?v=qzFDqOQMKoo

Task 2: https://www.youtube.com/watch?v=ll1VDwqPLDQTask