Universidad Del Valle De Guatemala 14/05/2021

Digital 2 Helder Ovalle Barrios

Sección:21 18349

Labrotorio #7

Link de github:

https://github.com/Helder1121/labo_07

Link de youtube:

https://www.youtube.com/watch?v=kYTFWZo8eg0

Progra comentada:

```
//**********************************************************************************************

// Laboratorio 7

// Helder Ovalle

// 18349

//**********************************************************************************************


//**********************************************************************************************

//Librerias

//**********************************************************************************************

#include <stdint.h>

#include <stdbool.h>

#include "inc/tm4c123gh6pm.h"

#include "inc/hw_memmap.h"

#include "inc/hw_types.h"

#include "inc/hw_ints.h"

#include "driverlib/sysctl.h"

#include "driverlib/sysctl.c"
```

```
#include "driverlib/interrupt.h"
#include "driverlib/interrupt.c"
#include "driverlib/gpio.h"
#include "driverlib/gpio.c"
#include "driverlib/timer.h"
#include "driverlib/timer.c"
#include "driverlib/uart.h"
#include "driverlib/uart.c"
#include "driverlib/pin_map.h"
#include "driverlib/debug.h"
#include "driverlib/rom.h"
#include "grlib/grlib.h"
#include <string.h>
//*****************************************************************************
**************
//Prototipos de funciones
//*****************************************************************************
**************
void UARTIntHandler(void);
void Timer0IntHandler(void);
void UARTSend(const uint8_t *pui8Buffer, uint32_t ui32Count);
//*****************************************************************************
**************
//Variables
//*****************************************************************************
**************
int ON = false; //Estado del led
char color='a'; //Color que se mostrara
//*****************************************************************************
**************
//Principal
```

```
//*****************************************************************************
**************

int main(void)

{

   //Configuraciones

   //Config clock

   SysCtlClockSet ( SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN |
SYSCTL_XTAL_16MHZ ) ;

   //Puerto F habilitado

   SysCtlPeripheralEnable (SYSCTL_PERIPH_GPIOF ) ;

   while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOF)){}

   //Se habiita el UART

   SysCtlPeripheralEnable (SYSCTL_PERIPH_UART0);

   while(!SysCtlPeripheralReady(SYSCTL_PERIPH_UART0)){}

   //Puerto A habilitado para el UART

   SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

   //Se habiita el RX y TX para la comunic.

   while(!SysCtlPeripheralReady(SYSCTL_PERIPH_UART0))

   GPIOPinConfigure(GPIO_PA0_U0RX);

   GPIOPinConfigure(GPIO_PA1_U0TX);

   IntMasterEnable();

   GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

   UARTDisable(UART0_BASE);


   //Config de los baudios a utilizar y demas parametros.

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

   IntEnable (INT_UART0);

   UARTIntEnable (UART0_BASE, UART_INT_RX);

   UARTEnable (UART0_BASE);
```

```
   IntPrioritySet(INT_UART0, 0x0);
   IntRegister(INT_UART0, UARTIntHandler);
   UARTFIFOEnable(UART0_BASE);
   UARTFIFOLevelSet(UART0_BASE,UART_FIFO_TX1_8,UART_FIFO_RX1_8);

   //Config del TIMER0
   SysCtlPeripheralEnable (SYSCTL_PERIPH_TIMER0);
   while(!SysCtlPeripheralReady(SYSCTL_PERIPH_TIMER0)){}
   SysCtlPeripheralReset (SYSCTL_PERIPH_TIMER0);
   SysCtlDelay (5);
   TimerDisable(TIMER0_BASE, TIMER_A|TIMER_B);
   TimerConfigure (TIMER0_BASE,TIMER_CFG_PERIODIC);
   TimerLoadSet (TIMER0_BASE, TIMER_A, 20000000 -1);
   TimerEnable (TIMER0_BASE, TIMER_A|TIMER_B);
   TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);//TIMER A
   TimerIntRegister(TIMER0_BASE, TIMER_A, Timer0IntHandler);//Establecer la interrupcion
   IntEnable(INT_TIMER0A); //Habilitar interrupción por parte del NVIC
   TimerEnable(TIMER0_BASE, TIMER_A);//Habilitar temporizador

   //Configuración puertos salidas
   GPIOPinTypeGPIOOutput (GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3)
;//rojo,azul y verde
   while(1){}
}

//Interrupcion del TIMER
void Timer0IntHandler(){
   TimerIntClear(TIMER0_BASE,  TIMER_TIMA_TIMEOUT);
      if (ON){
```

```
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x0);
    }else{
        switch(color){
                case 'r':
                    GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,0x02);
                    break;
                case 'g':
                    GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,0x08);
                    break;
                case 'b':
                    GPIOPinWrite(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3,0x04);
                    break;
                case 'o':
                    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x0);
                    break;
                    }
    }
    ON = !ON;
}
//Interrupcion del UART
void UARTIntHandler(){
  uint32_t ui32Status;
    //Inicia el estatus de la interrupcion
    ui32Status = UARTIntStatus(UART0_BASE, true);
    UARTIntClear(UART0_BASE, ui32Status);
    while(UARTCharsAvail(UART0_BASE))
    {
        color=UARTCharGet(UART0_BASE);
        UARTCharPutNonBlocking(UART0_BASE,color);
```

```
        }
}
//Envio de datos para la tiva
void UARTSend(const uint8_t *pui8Buffer, uint32_t ui32Count)
{
   while(ui32Count--)
   {
      UARTCharPutNonBlocking(UART0_BASE, *pui8Buffer++);
   }
}
```