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++);

}

}