

Lab 10 Exp 3

Pseudocódigo

1. Importa librerías
2. Configurar reloj del sistema
3. Habilitar perifericos de puertos necesarios para pines de LEDs, botones y puertos UART1 y UART0
4. Configurar pines de los LEDs como salidas
5. Configurar pines de botones como entradas y como weak pull ups
6. Configurar los pines TX y RX para UART0 y UART1
7. Habilitar interrupciones para UART1
8. En la interrupción del puerto UART1, si recibe algún valor entonces debe ponerlo en el puerto UART0; de esta forma sabremos qué está mandando el ESP8266 ya que este estará conectado al puerto UART1 y lo desplegamos en la pantalla de la computadora
9. Se crea una función para mandar varios caracteres en vez de solo uno (por comodidad)
10. Mandar el comando ATE0 para quitar el eco
11. Mandar el comando AT para poner el módulo en modo cliente
12. Mandar el comando AT para conectarnos a la red deseada ingresando nombre y contraseña de la red
13. En el código principal, se implementa antirrebote para cada botón
14. Si el botón 1 es presionado entonces aumenta el valor de una variable, pero esta solo puede tomar valores de 0 a 7
15. Por medio de lógica y manipulación de bits, se obtendrá el binario del valor de la variable que contiene el color y cada bit indicará el estado (encendido o apagado) de cada LED.
16. Si el otro botón es presionado, este enviará el post con las variables "id_color", "color" y "carnet" en el formato adecuado aprendido en clase a la IP del servidor de nuestra página.

Código de la Tiva C

```
/*  
 * Autor: Daniel Fuentes Oajaca  
 * Carnet: 17083  
 * Curso: Digital 2  
 * Seccion: 20  
 * Experimento 3  
 */
```

```
#include <stdint.h>  
#include <string.h>  
#include <stdio.h>
```

```
#include <stdbool.h>
#include "inc/tm4c123gh6pm.h"           // Librerías a utilizar
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "inc/hw_ints.h"
#include "inc/hw_gpio.h"
#include "driverlib/debug.h"
#include "driverlib/sysctl.h"
#include "driverlib/interrupt.h"
#include "driverlib/gpio.h"
#include "driverlib/uart.h"
#include "driverlib/pin_map.h"

/**
 * main.c
 */

int32_t uVal = 0;                       // Variable para guardar el valor
que venga del UART
int color = 0;
char* carnet = "17083";
char* id_color = "";
char* server = "192.168.1.33";
void sendString(char word[]);
void sendPOST(void);

int main(void)
{

SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN)
;           // Se configura el reloj a 20MHz
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
// Se activan los periferico GPIOF para los LEDs
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
// Se activan los periferico GPIOA para el UART0
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
// Se activan los periferico GPIOB para el UART1
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
// Se configuran los pines 1, 2 y 3 del puerto F (LED RGB) como salidas

    HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = GPIO_LOCK_KEY;
// Se realizan configuraciones para SW2
    HWREG(GPIO_PORTF_BASE + GPIO_O_CR) = 0X1;
    GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_4|GPIO_PIN_0);
// Se configura el pin 4 del puerto F (boton SW1 y SW2) como entrada
    GPIOPadConfigSet(GPIO_PORTF_BASE, GPIO_PIN_4|GPIO_PIN_0, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU);           // Se configura el boton con una "fuerza" de 2mA
y como weak Pull-Up

    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
// Se activa el periferico del puerto UART0
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART1);
// Se activa el periferico del puerto UART0
```

```
    GPIOPinConfigure(GPIO_PB0_U1RX);
//Se configuran TX y RX del UART1 en el puerto B
    GPIOPinConfigure(GPIO_PB1_U1TX);
    GPIOPinTypeUART(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1);
    UARTConfigSetExpClk(UART1_BASE, SysCtlClockGet(), 115200,
(UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE)); // Se
configura el UART1 igual al UART0

    IntMasterEnable();
// Se habilitan las interrupciones

    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0|GPIO_PIN_1);
// Se indican los pines para el puerto UART
    UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 115200,
(UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE)); // Se
configura el UART, sera el puerto 0, con el reloj del sistema, a un BaudRate de
115200 y el formato de datos sera de 8 bits con 1 STOP bit y sin parity bit
    IntEnable(INT_UART1);
// Se habilita la interrupcion del puerto UART0
    UARTIntEnable(UART1_BASE, UART_INT_RX|UART_INT_RT);
// Se habilita las interrupciones por dato recibido y Receive Timeout interrup
que es cuando termina de leerlo
    SysCtlDelay(30000000);
    sendString("ATE0\r\n");
// Comando AT para quitar eco
    SysCtlDelay(30000000);
    sendString("AT+CWMODE=1\r\n");
// Comando AT para poner el modulo en modo cliente
    SysCtlDelay(30000000);
    sendString("AT+CWJAP=");
// Comando AT para conectarse a alguna red
    UARTCharPut(UART1_BASE, '');
    sendString("HOTEL_DON_ALFONSO");
// Nombre de la red
    UARTCharPut(UART1_BASE, '');
    UARTCharPut(UART1_BASE, ',');
    UARTCharPut(UART1_BASE, '');
    sendString("Alfonso123");
// Contraseña de la red
    UARTCharPut(UART1_BASE, '');
    sendString("\r\n");
    SysCtlDelay(30000000);

    while(1){
// Loop principal
        if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_4) == 0){
            while(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_4) == 0);
// Espera a que deje de presionar para comenzar la rutina
            color = (color + 1)%8;
// Aumenta el valor de la variable color en el rango de 0 a 7
        }
        if(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_0) == 0){
            while(GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_0) == 0);
// Espera a que deje de presionar para comenzar la rutina
```

```
        sendPOST();
// Funcion para hacer el POST en la pagina
    }
    if(color&0x01 == 1){
// Mediante una revision de cada bit de la variable color enciende o apaga la
LED correspondiente
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0x0F);
    } else{
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0x00);
    }
    if((color>>1)&0x01 == 1){
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 0x0F);
    } else{
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 0x00);
    }
    if((color>>2)&0x01 == 1){
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0x0F);
    } else{
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0x00);
    }
}
}

void sendString(char word[]){
// Funcion para enviar Strings al UART1
    int i;
    for(i = 0;i<(strlen(word)); i = i+1){
        unsigned char letra = word[i];
        UARTCharPut(UART1_BASE, letra);
    }
}

void UARTIntHandler(void){
// Interrupcion del puerto UART1
    uint32_t ui32Status;
    ui32Status = UARTIntStatus(UART1_BASE, true);
// Se obtiene el masked interrupt status
    UARTIntClear(UART1_BASE, ui32Status);
// Se limpian la bandera de la interrupcion que se dio
    while(UARTCharsAvail(UART1_BASE)){
// Si se recibio un dato...
        uVal = UARTCharGet(UART1_BASE);
        UARTCharPut(UART0_BASE, uVal);
// Pone el caracter en el UART0 para poder ver la respuesta del módulo
    }
}

void sendPOST(void){
    sendString("AT+CIPSTART=");
// Comando AT para iniciar un post
    UARTCharPut(UART1_BASE, '"');
    sendString("TCP");
// Tipo del post
    UARTCharPut(UART1_BASE, '"');
    UARTCharPut(UART1_BASE, ',');
```

```
    UARTCharPut(UART1_BASE, '');
    sendString(server);
// Server al que se desea hacer el POST
    UARTCharPut(UART1_BASE, '');
    SysCtlDelay(3000);
    sendString(",80\r\n");
// Puerto
    SysCtlDelay(30000000);
    if (color == 0){
        sendString("AT+CIPSEND=170\r\n");
// Comando AT para enviar la cantidad de caracteres del post
        SysCtlDelay(30000000);
// contenido del POST (este es igual para todos, unicamente cambia el valor de
// las variables dependiendo del color del led
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 35\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=0&carnet=17083&color=Negro");
    }else if (color == 1){
        sendString("AT+CIPSEND=170\r\n");
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 34\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=1&color=Azul&carnet=17083");
    }else if (color == 2){
        sendString("AT+CIPSEND=170\r\n");
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 35\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=2&color=Verde&carnet=17083");
    }else if (color == 3){
        sendString("AT+CIPSEND=170\r\n");
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 38\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=3&color=Turquesa&carnet=17083");
    }else if (color == 4){
        sendString("AT+CIPSEND=170\r\n");
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 34\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=4&color=Rojo&carnet=17083");
    }else if (color == 5){
        sendString("AT+CIPSEND=170\r\n");
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 37\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=5&color=Violeta&carnet=17083");
    }else if (color == 6){
        sendString("AT+CIPSEND=170\r\n");
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 38\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=6&color=Amarillo&carnet=17083");
    }else {
        sendString("AT+CIPSEND=170\r\n");
```

```
        SysCtlDelay(30000000);
        sendString("POST /index.php HTTP/1.0\r\nHost: 192.168.1.33\r\nAccept:
/*\r\nContent-Length: 36\r\nContent-Type: application/x-www-form-
urlencoded\r\n\r\nid_color=7&color=Blanco&carnet=17083");
    }
    SysCtlDelay(40000000);
    sendString("\r\nAT+CIPCLOSE\r\n");
// Se cierra el POST
}
```

Código de la Pagina Web

```
<?php
header("Content-type: text/html; charset=utf-8");

$carnet = 0;
$id_color = 0;
$color = 0;

//print_r($_POST);

if($_POST){
    $carnet = $_POST['carnet'];
    $id_color = $_POST['id_color'];
    $color = $_POST['color'];

    $conexion = mysqli_connect("localhost:3306","root","", "color");
    if(!$conexion){
        echo "Error: No se pudo conectar a MySQL.".PHP_EOL;
        echo "errno de depuración: ". mysqli_connect_errno().PHP_EOL;
        echo "error de depuración: ". mysqli_connect_error().PHP_EOL;
    }else{
        $query = "INSERT INTO `datos` (`id`, `fecha`, `carnet`, `id_color`, `color`) VALUES (NULL,
CURRENT_TIMESTAMP, '$carnet', '$id_color', '$color');";

        mysqli_query($conexion, $query);
```

```
}

mysqli_close($conexion);
}

$conexion = mysqli_connect("localhost:3306","root","", "color");
if(!$conexion){
    echo "Error: No se pudo conectar a MySQL.".PHP_EOL;
    echo "errno de depuración: ". mysqli_connect_errno().PHP_EOL;
    echo "error de depuración: ". mysqli_connect_error().PHP_EOL;
}else{
    $query = "SELECT * FROM `datos` ORDER BY id DESC LIMIT 1";
    $result = mysqli_query($conexion,$query);
    $fecha =1;
}

mysqli_close($conexion);

?>

<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
```

```
integrity="sha384-
9alt2nRpC12Uk9gS9baDI411NQApFmC26EwAOH8WgZl5MYxXfFc+NcPb1dKGj7Sk"
crossorigin="anonymous">
```

```
<title>Lab 10 Exp 3</title>
```

```
</head>
```

```
<?php
```

```
$conexion = mysqli_connect("localhost:3306","root","", "color");
```

```
if(!$conexion){
```

```
    echo "Error: No se pudo conectar a MySQL.".PHP_EOL;
```

```
    echo "errno de depuración: ". mysqli_connect_errno().PHP_EOL;
```

```
    echo "error de depuración: ". mysqli_connect_error().PHP_EOL;
```

```
}else{
```

```
    $query = "SELECT id_color FROM `datos` ORDER BY id DESC LIMIT 1";
```

```
    $result2 = mysqli_query($conexion,$query);
```

```
    while($fila = mysqli_fetch_array($result2)){
```

```
        $ultimo_color = $fila['id_color'];
```

```
    }
```

```
}
```

```
mysqli_close($conexion);
```

```
if ($ultimo_color == 0){
```

```
    echo sprintf('<style type="text/css">
```

```
body{
```

```
    background-color:black
```

```
}
```

```
</style>');
```

```
}
```



```
if ($ultimo_color == 1){  
    echo sprintf('<style type="text/css">  
    body{  
        background-color:blue  
    }  
</style>');  
}  
  
if ($ultimo_color == 2){  
    echo sprintf('<style type="text/css">  
    body{  
        background-color:green  
    }  
</style>');  
}  
  
if ($ultimo_color == 3){  
    echo sprintf('<style type="text/css">  
    body{  
        background-color:cyan  
    }  
</style>');  
}  
  
if ($ultimo_color == 4){  
    echo sprintf('<style type="text/css">  
    body{  
        background-color:red  
    }  
</style>');  
}  
  
if ($ultimo_color == 5){
```

```
echo sprintf('<style type="text/css">
```

```
body{
```

```
    background-color:fuchsia
```

```
}
```

```
</style>');
```

```
}
```

```
if ($ultimo_color == 6){
```

```
echo sprintf('<style type="text/css">
```

```
body{
```

```
    background-color:yellow
```

```
}
```

```
</style>');
```

```
}
```

```
if ($ultimo_color == 7){
```

```
echo sprintf('<style type="text/css">
```

```
body{
```

```
    background-color:white
```

```
}
```

```
</style>');
```

```
}
```

```
?>
```

```
<div class='container'>
```

```
    <br>
```

```
    <h1><strong>LEDs Color</strong></h1>
```

```
    <br>
```

```
    <table class="table table-light">
```

```
<thead class="bg-success text-white">

<tr>

<th>Fecha</th>

<th>Carnet</th>

<th>id_color</th>

<th>Color</th>

</tr>

</thead>

<tbody>

<?php

while($fila = mysqli_fetch_array($result)){

echo "<tr>

<td>".$fila['fecha']. "</td>

<td>".$fila['carnet']. "</td>

<td>".$fila['id_color']. "</td>

<td>".$fila['color']. "</td>

</tr>";

}

?>

</tbody>

</table>

<br>

<div class="container" id="formulariodiv">

<form id="formulario" action="index.php" method="POST" accept-charset="utf-8">

<br><h2> Formulario </h2>

<br> Carnet: <br>
```

```
<input type="text" name="carnet" id="carnet">

<br> id_color: <br>

<input type="text" name="id_color" id="id_color">

    <br> Color: <br>

<input type="text" name="color" id="color">

<br>

<br><input type="submit" value="Enviar">

</form>

</div>

<!-- Optional JavaScript -->

<!-- jQuery first, then Popper.js, then Bootstrap JS -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

    integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"

    crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

    integrity="sha384-
Q6E9RHvblyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"

    crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"

    integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"

    crossorigin="anonymous"></script>

</body>

</html>
```