

Código Comentado

Proyecto 4

Tivaware

```
// Larry Paul Fuentes
// Proyecto Parqueo
// 28/05/2021
// Digital 2

#include<stdint.h>
#include<stdbool.h>
#include"inc/hw_memmap.h"
#include"inc/hw_types.h"
#include"inc/tm4c123gh6pm.h"
#include"driverlib/sysctl.h"
#include"driverlib/gpio.h"
#include"driverlib/timer.h"
#include"driverlib/interrupt.h"
#include "driverlib/uart.h"
#include "driverlib/pin_map.h"
#include "inc/hw_gpio.h"

int p1;
int p2;
int p3;
int p4;
int disp = 0;
int d1;
int d2;
int d3;
int d4;

//Prototipo de funciones
void Revision(void);
void display(void);

int main(void)
{
    SysCtlClockSet(
        SYSCTL_SYSDIV_5 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    //Peripheral Enable
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);

    //LEDS
    GPIOPinConfigure(GPIO_PD0_WT2CCP0); //D0
    GPIOPinConfigure(GPIO_PD1_WT2CCP1); //D1
```

```

//PUSH BUTTONS
//  GPIOPinConfigure(GPIO_PE0_U7RX);    //1
//  GPIOPinConfigure(GPIO_PE1_U7TX);    //2
//GPIOPinConfigure(GPIO_PE4_U5RX);    //3
//GPIOPinConfigure(GPIO_PC4_WT0CCP0);    //4

    GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_4 |
GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7); //
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2
| GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6); //Display Output
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_0 | GPIO_PIN_1); //D
    GPIOPinTypeGPIOInput(GPIO_PORTE_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4); //Pushes E

//Pullups
    GPIOPadConfigSet(GPIO_PORTE_BASE, GPIO_PIN_1, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU);
    GPIOPadConfigSet(GPIO_PORTE_BASE, GPIO_PIN_2, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU);
    GPIOPadConfigSet(GPIO_PORTE_BASE, GPIO_PIN_3, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU);
    GPIOPadConfigSet(GPIO_PORTE_BASE, GPIO_PIN_4, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD_WPU);

//Luces inciales
    GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0 | GPIO_PIN_1, GPIO_PIN_0); //D0
    GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_4 |
GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,
                GPIO_PIN_2 | GPIO_PIN_4 | GPIO_PIN_6); //A2,4,6
    GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0 | GPIO_PIN_1, GPIO_PIN_1); // D1
    GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_2 | GPIO_PIN_3 | GPIO_PIN_4 |
GPIO_PIN_5 | GPIO_PIN_6 | GPIO_PIN_7,
                GPIO_PIN_3 | GPIO_PIN_5 | GPIO_PIN_7); //A3,5,7

    while (1)
    {

        //Leer pushes
        p1 = GPIOPinRead(GPIO_PORTE_BASE, GPIO_PIN_1);
        p2 = GPIOPinRead(GPIO_PORTE_BASE, GPIO_PIN_2);
        p3 = GPIOPinRead(GPIO_PORTE_BASE, GPIO_PIN_3);
        p4 = GPIOPinRead(GPIO_PORTE_BASE, GPIO_PIN_4);
        disp = d1 + d2 + d3 + d4; // Contador de Display
        Revision(); // Revisa los estados de los botones
        display(); // Display

    }
}

void Revision(void)
{
    if (p1 == 0)
    { //Parqueo 1 esta ocupado se enciende led roja

```

```

        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_1, 0);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, GPIO_PIN_0);
        d1 = 0; // La badera se apaga ya que tendran un espacio disponible menos
    }
    else
    {
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 0);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_1, GPIO_PIN_1);
        d1 = 1;
    }

    if (p2 == 0)
    { //Parqueo 2 esta ocupado se enciende led roja
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_3, 0);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_2, GPIO_PIN_2);
        d2 = 0;
    }
    else
    {
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_3, GPIO_PIN_3);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_2, 0);
        d2 = 1;
    }

    if (p3 == 0)
    { //Parqueo 2 esta ocupado se enciende led roja
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_5, 0);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4, GPIO_PIN_4);
        d3 = 0;
    }
    else
    {
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_5, GPIO_PIN_5);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4, 0);
        d3 = 1;
    }

    if (p4 == 0)
    { //Parqueo 2 esta ocupado se enciende led roja
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_7, 0);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6, GPIO_PIN_6);
        d4 = 0;
    }
    else
    {
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_7, GPIO_PIN_7);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_6, 0);
        d4 = 1;
    }
}

void display(void){

    // Display
    //-----

```

```

// PB0 - g
// PB1 - f
// PB2 - a
// PB3 - b
// PB4 - e
// PB5 - d
// PB6 - c
// Switch Case normal con todos los numeros.
switch (disp){
case 0:

    GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0| GPIO_PIN_1 | GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 , GPIO_PIN_0); //0
    break;
case 1:
    // 0,1,2,4,5
    GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0| GPIO_PIN_1 | GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 ,
GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2 | GPIO_PIN_4 |
GPIO_PIN_5); //1
    break;
case 2:
    //1,6
    GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0| GPIO_PIN_1 | GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 ,
GPIO_PIN_1|GPIO_PIN_6); //2
    break;
case 3:
    //1,4
    GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0| GPIO_PIN_1 | GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 ,
GPIO_PIN_1|GPIO_PIN_4); //3
    break;
case 4:
    //2,4,5
    GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0| GPIO_PIN_1 | GPIO_PIN_2 |
GPIO_PIN_3 | GPIO_PIN_4 | GPIO_PIN_5 | GPIO_PIN_6 ,
GPIO_PIN_2|GPIO_PIN_4|GPIO_PIN_5); //4
    break;
}
}

```

ESP32

```

//Larry Paul Fuentes
#include <WiFi.h>
#include <WebServer.h>
#include <SPIFFS.h>

const char* ssid = "CLAR01_1CFE9F";
const char* password = "485f7wvKFq";
WebServer server(80); // Object of WebServer(HTTP port, 80 is default)

```

```

int parqueo = 0;
int parqueolibre = 1;
int parqueo2libre = 1;
int parqueo3libre = 1;
int parqueo4libre = 1;

uint8_t LED1pin = 2;
bool LED1status = LOW;

//*****
// Configuración
//*****
void setup() {
  Serial.begin(115200);
  Serial2.begin(115200, SERIAL_8N1, 16, 17);

  while (!Serial);
  while (!Serial2);

  Serial.println("Try Connecting to ");
  Serial.println(ssid);

  pinMode(LED1pin, OUTPUT);

  // Connect to your wi-fi modem
  WiFi.begin(ssid, password);

  // Check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected successfully");
  Serial.print("Got IP: ");
  Serial.println(WiFi.localIP()); //Show ESP32 IP on serial

  server.on("/", handle_OnConnect); // Directamente desde e.g. 192.168.0.8
  server.on("/led1on", handle_led1on);
  server.on("/led1off", handle_led1off);

  server.onNotFound(handle_NotFound);

  server.begin();
  Serial.println("HTTP server started");
  delay(100);
}

void loop() {

  server.handleClient();

  if (Serial2.available() > 0) {

```

```

    parqueo = Serial2.read();
}

if (parqueo == 97) {
    parqueolibre = 0;
}

if (parqueo == 99) {
    parqueo2libre = 0;
}

    if (parqueo == 101){
        parqueo3libre = 0;
    }

if (parqueo == 103) {
    parqueo4libre = 0;
}

if (parqueo == 98) {
    parqueolibre = 1;
}

if (parqueo == 100) {
    parqueo2libre = 1;
}

if (parqueo == 102) {
    parqueo3libre = 1;
}

if (parqueo == 104) {
    parqueo4libre = 1;
}

}

void handle_OnConnect() {
    LED1status = LOW;
    Serial.println("GPIO2 Status: OFF");
    server.send(200, "text/html", SendHTML(LED1status));
}

void handle_led1on() {
    LED1status = HIGH;
    Serial.println("GPIO2 Status: ON");
    server.send(200, "text/html", SendHTML(LED1status));
}

```

```

void handle_led1off() {
    LED1status = LOW;
    Serial.println("GPIO2 Status: OFF");
    server.send(200, "text/html", SendHTML(LED1status));
}

```

```

String SendHTML(uint8_t led1stat) {
    String ptr = "<!DOCTYPE html> <html>\n";
    ptr += "<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0, user-scalable=no\">\n";
    ptr += "<title>Proyecto 4</title>\n";
    ptr += "<body bgcolor=#dddddd>";
    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin:
0px auto; text-align: center;}\n";
    ptr += "body {margin-top: 50px;} h1 {color: #FFFFFF;margin: 50px auto 30px;}
h4 {color: #FFFFFF;margin: 15px;} h3 {color: #FFFFFF;margin-bottom: 15px;}\n";
    ptr += "table {";
    ptr += "    font-family: Helvetica, sans-serif;";
    ptr += "    border-collapse: collapse;";
    ptr += "    width: 100%;";
    ptr += "    margin: 50px auto 30px;";
    ptr += "    text-align: left;";
    ptr += "};";
    ptr += "td, th {";
    ptr += "    border: 1px solid #101010;";
    ptr += "    text-align: center;";
    ptr += "    padding: 8px; ";
    ptr += "    background-color: #008080;";
    ptr += "};";
    ptr += "tr:nth-child(even) {";
    ptr += "    background-color: #dddddd;";
    ptr += "} ";
    ptr += ".button {display: block;width: 80px;background-color: #3498db;border:
none;color: white;padding: 13px 30px;text-decoration: none;font-size:
25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";
    ptr += ".button-on {background-color: #3498db;}\n";
    ptr += ".button-on:active {background-color: #2980b9;}\n";
    ptr += ".button-off {background-color: #34495e;}\n";
    ptr += ".button-off:active {background-color: #2c3e50;}\n";
    ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px;}\n";
    ptr += "</style>\n";
    ptr += "<script>\n";
    ptr += "<!--\n";
    ptr += "function timedRefresh(timeoutPeriod) {\n";
    ptr += "\tsetTimeout(\"location.reload(true);\",timeoutPeriod);\n";
    ptr += "}\n";
    ptr += "\n";
    ptr += "window.onload = timedRefresh(1000);\n";
    ptr += "\n";
    ptr += "//    -->\n";
    ptr += "</script>";
    ptr += "</head>\n";
    ptr += "<body>\n";

```

```

ptr += "<h1>Parqueos</h1>\n";
ptr += "<h3>LARRY PAUL FUENTES</h3>\n";
ptr += "<h4>Proyecto 4 </h1>\n";
ptr += "<h2>Parqueos</h2>";

ptr += "<table>";
ptr += " <tr>";
ptr += "    <th>Parqueo</th>";
ptr += "    <th>Estado</th>";
ptr += " </tr>";
ptr += " <tr>";
ptr += "    <td>Parqueo 1</td>";
ptr += "    <td>Libre</td>";
ptr += " </tr>";
ptr += " <tr>";
ptr += "    <td>Parqueo 2</td>";
ptr += "    <td>Libre</td>";
ptr += " </tr>";
ptr += " <tr>";
ptr += "    <td>Parqueo 3</td>";
ptr += "    <td>Libre</td>";
ptr += " </tr>";
ptr += " <tr>";
ptr += "    <td>Parqueo 4</td>";
ptr += "    <td>Libre</td>";
ptr += " </tr>";

ptr += "</table>";

if (led1stat)
{
    ptr += "<p>Presione el boton para refrescar la pagina.</p><a class=\"button button-off\" href=\"/led1off\">ACT</a>\n";
}
else
{
    ptr += "<p>Presione el boton para refrescar la pagina.</p><a class=\"button button-on\" href=\"/led1on\">ACT</a>\n";
}
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}

//*****
// Handler de not found
//*****
void handle_NotFound() {
    server.send(404, "text/plain", "Not found");
}

```