

Universidad del Valle de Guatemala
Facultad de Ingeniería
Depto. de Ingeniería Electrónica, Mecatrónica y Biomédica
IE3027: Electrónica Digital II



Proyecto No. 4

Control de parqueos

Marco S. Alejandro Juárez Urrutia
Carné 18516

Introducción

El proyecto consiste en realizar un plan piloto para recopilar la ocupación de espacios disponibles de un nivel de sótano, el cual posee 4 parqueos en su estructura. Buscamos que la información sea en tiempo real. Esto lo logramos diseñando una página web donde se reciban notificaciones cada vez que exista un cambio en dichos parqueos.

El proyecto fue realizado con una Tiva C donde se programo todo el funcionamiento del circuito y la comunicación con los servidores se realizo mediante el microcontrolador ESP32.

Contenido

- Datos gráficos



Figura 1. Interfaz grafica con el usuario

- Explicaciones

Tabla 1. Conexiones de Display

Display	Pin Tiva C
A	PE4
B	PE5
C	PB2
D	PB3
E	PB4
F	PB5
G	PB6

Tabla 2. Conexiones de botones

Botón	Pin Tiva C
1	PC4
2	PC5
3	PC6
4	PC7
Botones configurados como pushdown	

Tabla 3. Conexiones LEDs Verdes

LED	Pin Tiva C
1	PE0
2	PE1
3	PE2
4	PE3

Tabla 4. Conexiones LEDs Rojos

LED	Pin Tiva C
1	PD0
2	PD1
3	PD2
4	PD3

Tabla 5. Conexión UART en Tiva C

LED	Pin Tiva C
RX	PB0
TX	PB1
Se utilizo el UART1 de la Tiva C	

Tabla 5. Conexión UART en ESP32

LED	Pin Tiva C
RX	Pin 3
TX	Pin 1
Se utilizo el UART0 del ESP32	

- Código comentado

Código CodeComposer

```

1 //-----Electrónica Digital II-----
2 //-----Proyecto #4-----
3 //-----Marco Juárez - 18516-----
4
5 //*****
6 // Librerías
7 //*****
8 #include <stdint.h>
9 #include <stdbool.h>
10 #include "inc/tm4c123gh6pm.h"
11 #include "inc/hw_memmap.h"
12 #include "inc/hw_types.h"
13 #include "inc/hw_ints.h"
14 #include "inc/hw_gpio.h"
15 #include "driverlib/sysctl.h"
16 #include "driverlib/interrupt.h"
17 #include "driverlib/gpio.h"
18 #include "driverlib/timer.h"
19 #include "driverlib/uart.h"
20 #include "driverlib/pin_map.h"
21
22 //*****
23 // Variables
24 //*****
25 uint8_t bandera = 0;
26 //*****
27 // Prototipos de funciones
28 //*****
29 void display(uint8_t data); //display
30 //*****
31 // Función Principal
32 //*****
33 int main(void)
34 {
35     //Configuraciones reloj
36     SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
37
38     //Configuración de puerto b para el display
39     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB); //Habilitamos puerto B
40     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE); //Habilitamos puerto E
41     GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6);
42     GPIOPinTypeGPIOOutput(GPIO_PORTC_BASE, GPIO_PIN_4|GPIO_PIN_5);
43
44     //Puerto c para los push
45     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
46     GPIOPinTypeGPIOInput(GPIO_PORTC_BASE, GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7); //pines a utilizar
47     GPIOPadConfigSet(GPIO_PORTC_BASE, GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7, GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD_WPD);
48
49     //Puerto D para los leds rojos
50     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD); //Habilitamos puerto D
51     GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
52
53     //Puerto E para leds verdes
54     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
55     GPIOPinTypeGPIOOutput(GPIO_PORTC_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
56
57     //Configuración UART
58     SysCtlPeripheralEnable(SYSCTL_PERIPH_UART1); //Habilitamos UART1
59     SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB); //Habilitamos puerto B
60     GPIOPinConfigure(GPIO_PB0_U1RX); //Habilitamos RX
61     GPIOPinConfigure(GPIO_PB1_U1TX); //Habilitamos TX
62     GPIOPinTypeUART(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1);
63     UARTConfigSetExpClk(UART1_BASE, SysCtlClockGet(), 115200, (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE)); // config uar
64     UARTIntClear(UART1_BASE, UART_INT_RX | UART_INT_RT | UART_INT_TX | UART_INT_FE | UART_INT_PE | UART_INT_BE | UART_INT_OE | UART_INT_RI | UAF
65     UARTEnable (UART1_BASE);
66
67     uint8_t bandera_dos = 1;
68
69     while (1)
70     {
71         bandera = GPIOPinRead(GPIO_PORTC_BASE, GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7) >> 4; //CORRIMIENTO DE BITS 1101 0110 0011 0001
72         if (bandera != bandera_dos) //Si es diferente a
73         {
74             GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, bandera);
75             GPIOPinWrite(GPIO_PORTC_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, ~bandera);
76             display( (4 - (((bandera >> 3) & 0x01) + ((bandera >> 2) & 0x01) + ((bandera >> 1) & 0x01) + (bandera & 0x01)));
77             bandera_dos = bandera;
78             UARTCharPut(UART1_BASE, bandera); //
79         }
80     }
81 }

```

```

83 void display(uint8_t bandera)
84 {
85     uint8_t temp = 0x71;
86     switch (bandera & 0x0F)
87     {
88         //Display inicio en 0
89         case 0:
90             GPIOWrite(GPIO_PORTA_BASE, GPIO_PIN_5, 0xFF);
91             GPIOWrite(GPIO_PORTA_BASE, GPIO_PIN_4, 0xFF);
92             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0xFF);
93             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0xFF);
94             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0xFF);
95             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_5, 0xFF);
96             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_6, 0x00);
97             break;
98         //Display en 1
99         case 1:
100             GPIOWrite(GPIO_PORTA_BASE, GPIO_PIN_5, 0x00);
101             GPIOWrite(GPIO_PORTA_BASE, GPIO_PIN_4, 0xFF);
102             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0xFF);
103             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0x00);
104             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0x00);
105             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_5, 0x00);
106             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_6, 0x00);
107             break;
108         //Display en 2
109         case 2:
110             GPIOWrite(GPIO_PORTA_BASE, GPIO_PIN_5, 0xFF);
111             GPIOWrite(GPIO_PORTA_BASE, GPIO_PIN_4, 0xFF);
112             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0x00);
113             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0xFF);
114             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0xFF);
115             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_5, 0x00);
116             GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_6, 0xFF);

```

```

109         case 2:
110             GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_5, 0xFF);
111             GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_4, 0xFF);
112             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0x00);
113             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0xFF);
114             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0xFF);
115             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_5, 0x00);
116             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_6, 0xFF);
117             break;
118         //Display en 3
119         case 3:
120             GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_5, 0xFF);
121             GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_4, 0xFF);
122             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0xFF);
123             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0xFF);
124             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0x00);
125             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_5, 0x00);
126             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_6, 0xFF);
127             break;
128         //Display en 4
129         case 4:
130             GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_5, 0x00);
131             GPIOPinWrite(GPIO_PORTE_BASE, GPIO_PIN_4, 0xFF);
132             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 0xFF);
133             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0x00);
134             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0x00);
135             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_5, 0xFF);
136             GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_6, 0xFF);
137             break;
138         default:
139             temp = 0xFF;
140     }
141 }
142

```

Código Arduino para ESP32

```
2   ESP32 Web Server
3   Ejemplo de creación de Web server
4   Basándose en los ejemplos de:
5   https://lastminuteengineers.com/creating-esp32-web-server-arduino-ide/
6   https://electropeak.com/learn
7   ****
8   // ****
9   // Librerías
10  // ****
11  #include <WiFi.h>
12  #include <WebServer.h>
13  // ****
14  // Variables globales
15  // ****
16  // SSID & Password
17  const char* ssid = "ASUS_68"; // Enter your SSID here
18  const char* password = "notebook_7946"; //Enter your Password here
19
20  WebServer server(80); // Object of WebServer(HTTP port, 80 is default)
21
22  //Variables
23  uint8_t dato = 0;
24  uint8_t espacio1 = 0;
25  uint8_t espacio2 = 0;
26  uint8_t espacio3 = 0;
27  uint8_t espacio4 = 0;
28  uint8_t cantidad = 0;
29  // ****
29  // ****
30  // Configuración
31  // ****
32  void setup() {
33    Serial.begin(115200);
34    Serial.println("Try Connecting to ");
35    Serial.println(ssid);
36
37    // Connect to your wi-fi modem
38    WiFi.begin(ssid, password);
39
40    // Check wi-fi is connected to wi-fi network
41    while (WiFi.status() != WL_CONNECTED) {
42      delay(1000);
43      Serial.print(".");
44    }
45    Serial.println("");
46    Serial.println("WiFi connected successfully");
47    Serial.print("Got IP: ");
48    Serial.println(WiFi.localIP()); //Show ESP32 IP on serial
49
50    server.on("/", handle_OnConnect); // Directamente desde e.g. 192.168.0.8
51
52    server.onNotFound(handle_NotFound);
53  }
```



```

54     server.begin();
55     Serial.println("HTTP server started");
56     delay(100);
57 }
58 //*****
59 // loop principal
60 //*****
61 void loop() {
62     server.handleClient();
63     if (Serial.available() > 0) {
64         dato =Serial.read();
65         Serial.println(dato);
66         //Para parqueo 1
67         if (dato & 0x01){
68             espaciol = 1;
69         }else {
70             espaciol = 0;
71         }
72         //Para parqueo 2
73         if (dato & 0x02){
74             espacio2 = 1;
75         }else {
76             espacio2 = 0;
77         }
78         //Para parqueo 3
79         if (dato & 0x04){
80             espacio3 = 1;
81         }else {
82             espacio3 = 0;
83         }
84         //Para parqueo 4
85         if (dato & 0x08){
86             espacio4 = 1;
87         }else {
88             espacio4 = 0;
89         }
90     }
91 }
92 //*****
93 // Handler de Inicio página
94 //*****
95 void handle_OnConnect() {
96     Serial.println("GPIO2 Status: Conected");
97     cantidad = -(espaciol + espacio2 + espacio3 + espacio4 ) + 4 ;//Muestra la cantidad de parqueos disponibles
98     server.send(200, "text/html", SendHTML2(espaciol, espacio2, espacio3, espacio4, cantidad));
99 }
100
101 //*****
102 // Procesador de HTML
103 //*****

```

```

101 //*****
102 // Procesador de HTML
103 //*****
104 String SendHTML2(uint8_t parq_1, uint8_t parq_2, uint8_t parq_3, uint8_t parq_4, uint8_t cantidad) {
105     String ptr = "<html>\n";
106     ptr += "<!doctype html>\n";
107     ptr += "<html lang=\"en\">\n";
108     ptr += "    <head>\n";
109     ptr += "        <style>html { font-family: Helvetica; display: inline-block; margin: 50px auto; text-align: center;}\n";
110     ptr += ".column {float: left ; width: 25%; padding: 10px}\n";
111     ptr += "row:after {content:\"\"; display: table; clear: both}\n";
112     ptr += "</style>\n";
113     ptr += "        <meta charset=\"utf-8\">\n";
114     ptr += "        <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\n";
115     ptr += "        <link href=\"https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css\" rel=\"stylesheet\">\n";
116     ptr += "        <title>Proyecto#4</title>\n";
117     ptr += "    </head>\n";
118     ptr += "    <body>\n";
119     ptr += "        <h1>Parqueo-Matic &#128664</h1>      \n";
120     ptr += "        <h3>Control de registro de parqueos</h3>      \n";
121     ptr += "        <h1></h1>      \n";
122     ptr += "        <table style=\"text-align:center;\"; class=\"table table-hover\">\n";
123
124
125     if (espacio1 == 1){
126         ptr += "            <th scope=\"row\">Parqueo 01</th>\n";
127         ptr += "            <td class=\"table table-hover\">Ocupado      &#10060</td>\n";
128         ptr += "        </tr>\n";
129         ptr += "        <tr>\n";
130     }else {
131         ptr += "            <th scope=\"row\">Parqueo 01</th>\n";
132         ptr += "            <td class=\"table table-hover\">Disponible &#9989</td>\n";
133         ptr += "        </tr>\n";
134         ptr += "        <tr>\n";
135     }
136
137     if (espacio2 == 1){
138         ptr += "            <th scope=\"row\">Parqueo 02</th>\n";
139         ptr += "            <td class=\"table table-hover\">Ocupado      &#10060</td>\n";
140         ptr += "        </tr>\n";
141         ptr += "        <tr>\n";
142     }else {
143         ptr += "            <th scope=\"row\">Parqueo 02</th>\n";
144         ptr += "            <td class=\"table table-hover\">Disponible &#9989</td>\n";
145         ptr += "        </tr>\n";
146         ptr += "        <tr>\n";
147     }
148     if (espacio3 == 1){
149         ptr += "            <th scope=\"row\">Parqueo 03</th>\n";
150         ptr += "            <td class=\"table table-hover\">Ocupado      &#10060</td>\n";
151         ptr += "        </tr>\n";
152         ptr += "        <tr>\n";
153     }else {
154         ptr += "            <th scope=\"row\">Parqueo 03</th>\n";
155         ptr += "            <td class=\"table table-hover\">Disponible &#9989</td>\n";

```

```

159 if (espacio4 == 1){
160     ptr += "        <th scope=\"row\">Parqueo 04</th>\n";
161     ptr += "        <td class=\"table table-hover\">Ocupado    &#10060</td>\n";
162     ptr += "    </tr>\n";
163     ptr += "    <tr>\n";
164 }else {
165     ptr += "        <th scope=\"row\">Parqueo 04</th>\n";
166     ptr += "        <td class=\"table table-hover\">Disponible &#9989</td>\n";
167     ptr += "    </tr>\n";
168     ptr += "    <tr>\n";
169 }
170
171 ptr += " </tbody>\n";
172 ptr += " </tfoot>\n";
173 ptr += " <t<tr>\n";
174 ptr += "<th class=table-active scope=row>Parqueos Disponibles</th>\n";
175 ptr += "<td class=table-info>";
176 ptr += (int) cantidad;
177 ptr += "</td>\n";
178 ptr += "    </tr>\n";
179 ptr += " </tfoot>\n";
180 ptr += "</table>\n";
181 ptr += "<script>\n";
182 ptr += "function timedRefresh(timeoutPeriod) {\n";
183 ptr += "\tsetTimeout(\"location.reload(true);\", timeoutPeriod);\n";
184 ptr += "window.onload = timedRefresh(1000);\n";
185 ptr += "</script>\n";
186 ptr += "    <script src=\"https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js\"
187 ptr += "    <script src=\"https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js\" integrity=\"sha384-gtEjrD/SeCtmISk7bNUAaR9oLDO//E1J19smoruRV6t3Iehde+3U1b98n9P1x0
188 ptr += "    <script src=\"https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js\" integrity=\"sha384-1QoLX15PILP0h7U9qD4m8W5LKq7k0Q8X4s40tQ8Z31iW4AwPrgFTxbJ8NT4GN188p\"
189 ptr += "    </body>\n";
190 ptr += "</html>";
191
192 return ptr;
193 }
194
195 //*****
196 // Handler de not found
197 //*****
198 void handle_NotFound() {
199     server.send(404, "text/plain", "Not found");
200 }

```