

Cristopher Sagastume 18640

PROYECTO No. 4

CONTROL DE REGISTRO PARA PARQUEO

Ha sido contratado en la empresa de parqueos Parqueo-matic, le asignan a usted la tarea de realizar un sistema embebido el cual es un plan piloto para recopilar la ocupación de espacios disponibles de un nivel de sótano, el cual posee 4 parqueos. La información la necesitan tener en tiempo real, puesto que están evaluando valdrá la pena invertir en los 10 sótanos restantes del edificio. Su tarea será enviar notificaciones al sistema de servidores de la empresa Parqueo-matic, cada vez que exista un cambio en dichos parqueos.

MATERIALES

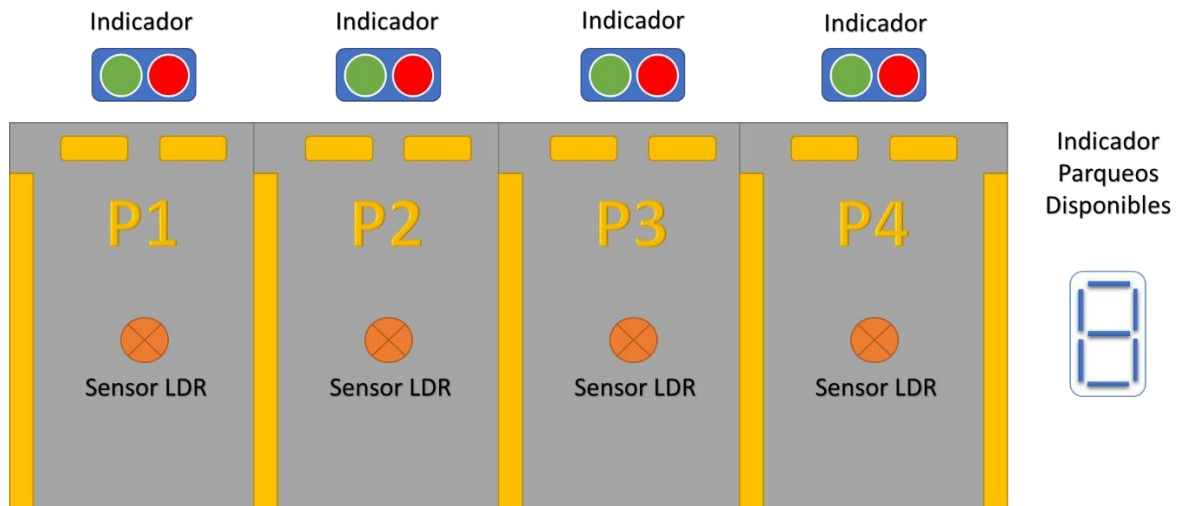
Deberá utilizar los siguientes materiales:

- TIVA C Launchpad
- ESP32
- Protoboard
- 8 Leds (4 Rojos y 4 Verdes)
- 4 Sensores de posición (puede ser pushbuttons /infrarrojos / ultrasónicos / láser + LDR (resistencia variable)
- 1 Display de 7 Segmentos o LCD
- Resistencias variadas
- Pushbuttons

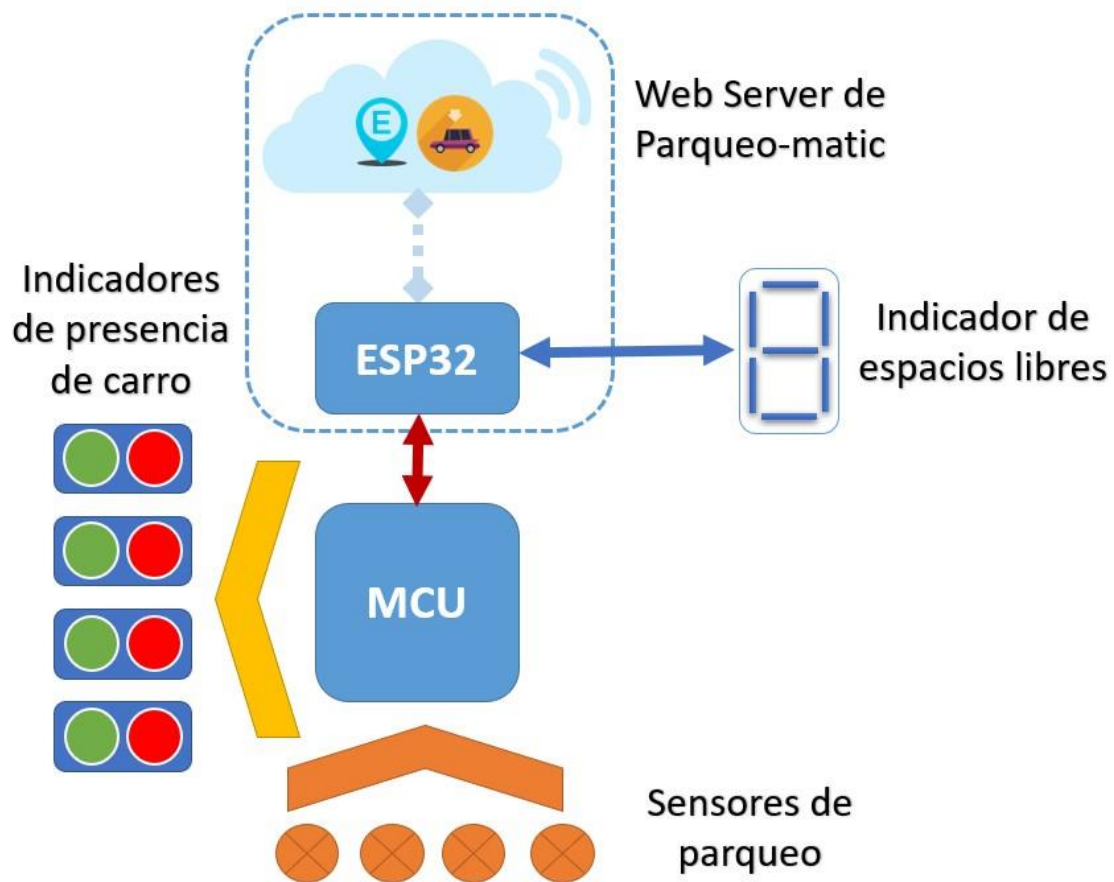
FUNCIONAMIENTO

Actualmente no tienen una infraestructura adecuada para realizar el control de parqueos disponibles, sin embargo, le piden que tenga lo siguiente:

- Un indicador general que establezca la cantidad de espacios disponibles que se tienen en el nivel.
- Por cada espacio un indicador con dos colores para establecer si está vacío (verde) o está ocupado (rojo).
- Para identificar si existe la presencia de los automóviles le piden que no gaste en sensores sofisticados puesto que quieren que el sistema sea lo más barato posible, porque lo quieren implementar en todos sus parqueos de los 10 sótanos.



CIRCUITO PROPUESTO



CONSIDERACIONES IMPORTANTES

- Trabajo escrito con la información y análisis que realizaron (circuitos utilizados, datos, gráficos, explicaciones, código debidamente comentado) (10 pts.)

- Link a un video de Youtube donde expliquen el proyecto (5 pts.)
- Funcionamiento de Proyecto (85 pts.)
 - Arranque del circuito normal (5 pts.)
 - Implementación de Sensores (25 pts.)
 - Implementación de indicadores (20 pts.)
 - Web Server ESP32 (35 pts.)

TOTAL (100 PTS.)

Pseudocodigo

- Se configuran los botones, leds, salidas de 7 segmentos, UART's y webserver (con wifi).
- Se inicia con 4 leds verdes (parqueos libres), el 4 en el 7 segmentos y mostrando 4 parqueos disponibles en el webser junto con su respectiva imagen de parqueos disponibles.
- Se verifica si se presionó alguno de los botones y dependiendo cual se presionó se cambia el estado del parqueo respectivo y se envía un dato por UART1 a la ESP32
- En la ESP32 se verifica que dato se recibió para poder aumentar o restar un contador que nos ayudara a mostrar el valor de número de parqueos disponibles en el 7 segmentos y en el webserver.
- Se verifica que se recibió en el ESP32 para poder cambiar el estado de los parqueos a las variables banderas que se le asigno a cada parqueo.
- Dependiendo el valor de la variable contador y de las banderas de estado se envía un código HTML especifico mostrando la cantidad de parqueos disponibles y cuales parqueos específicamente están libres y cuales ocupados.

Código Comentado

ESP32:

//Cristopher Sagastume 18640

```
/*
*****
ESP32 Web Server
Ejemplo de creación de Web server con acceso a SPIFFS
Basándose en los ejemplos de:
https://lastminuteengineers.com/creating-esp32-web-server-arduino-ide/
https://electropeak.com/learn
https://github.com/luisllamasbinaburo/ESP32-Examples
*****
*****/

//*****

// Librerías
//*****

#include <WiFi.h>
#include <SPIFFS.h>
#include <WebServer.h>
//definimos los pines de UART 2
#define RXD0 3
#define TXD0 1
//*****

// Variables globales
//*****

// SSID & Password Seteamos nuestro Wifi
const char* ssid = "TURBONETT_668E97"; // Enter your SSID here
const char* password = "JFCESG*8485"; //Enter your Password here
```

```
WebServer server(80); // Object of WebServer(HTTP port, 80 is default)
//variables que nos servirán para estados de parqueos etc.
char Led[20];
uint8_t LED1pin = 2;
uint8_t segm = 0;
uint8_t count = 4;
bool LED1status = LOW;
int Led2 = 0;
bool Parqueo1 = 0;
bool Parqueo2 = 0;
bool Parqueo3 = 0;
bool Parqueo4 = 0;

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

// Configuración
//*****
*****

void setup() {
  Serial.begin(115200, SERIAL_8N1, RXD0, TXD0);
  //Serial2.begin(115200, SERIAL_8N1, RXD0, TXD0);
  if (!SPIFFS.begin()) {
    Serial.println("An Error has occurred while mounting SPIFFS");
    return;
  }
  Serial.println("Try Connecting to ");
  Serial.println(ssid);
  //configuramos las salidas para el 7 segmentos
  pinMode(23, OUTPUT);
  pinMode(22, OUTPUT);
  pinMode(21, OUTPUT);
  pinMode(19, OUTPUT);
}
```

```
pinMode(18, OUTPUT);
pinMode(5, OUTPUT);
pinMode(15, 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.onNotFound([]() {          // If the client requests any URI
  if (!HandleFileRead(server.uri())) // send it if it exists
    handleNotFound();              // otherwise, respond with a 404 (Not Found) error
});
//server.onNotFound(handle_NotFound);

server.begin();
Serial.println("HTTP server started");
delay(100);
//colocamos en el 7 segmentos un 4 que equivale a 4 parqueos libres
digitalWrite(23, LOW);//a
digitalWrite(22, LOW);//b
digitalWrite(21, HIGH);//c
```

```
digitalWrite(19, LOW);//d
digitalWrite(18, HIGH);//e
digitalWrite(5, HIGH);//f
digitalWrite(15, HIGH);//g
}
//*****
*****

// loop principal
//*****
*****

void loop() {
  server.handleClient();

  //se recibio algo por Uart?
  if (Serial.available() > 0) {
    //guardamos el valor recibido en la variable Led y en la variable Led2 obtenemos el largo de la
    cadena recibida
    Led2 = Serial.readBytes(Led, 20);
    //verificamos si lo recibido equivale a los datos del parqueo ocupado para restarle 1 al contador y
    así modificar el 7 segmentos y el web server
    if (Led2 == 2 or Led2 == 4 or Led2 == 6 or Led2 == 8 ) {
      count--;

      //verificamos si lo recibido equivale a los datos del parqueo libre para sumarle 1 al contador y así
      modificar el 7 segmentos y el web server
      else if ( Led2 == 1 or Led2 == 3 or Led2 == 5 or Led2 == 7 ) {
        count++;
      }
      //en todo este bloque verificamos que dato se recibio para modificar las banderas de estado de
      parqueo que nos servirá para identificar en el web server que parqueo tuvo un cambio
      if (Led2 == 2) { //parqueo 1 ocupado
```



```
    Parqueo1 = 1;

}
else if (Led2 == 1) { //parqueo 1 disponible
    Parqueo1 = 0;
}
if (Led2 == 4) { //parqueo 2 libre
    Parqueo2 = 1;

}
else if (Led2 == 3) { //parqueo 2 ocupado
    Parqueo2 = 0;
}
if (Led2 == 6) { //parqueo 3 libre
    Parqueo3 = 1;

}
else if (Led2 == 5) { //parqueo 3 ocupado
    Parqueo3 = 0;
}
if (Led2 == 8) { //parqueo 4 libre
    Parqueo4 = 1;

}
else if (Led2 == 7) { //parqueo 4 ocupado
    Parqueo4 = 0;
}
```

```
server.send(200, "text/html", SendHTML(count, Parqueo1, Parqueo2, Parqueo3, Parqueo4));  
//enviamos los datos al web server inyectandola por medio de la funcion SendHTML
```

```
//verificamos el valor de count para asignarle un valor al 7 segmentos
```

```
switch (count) {
```

```
case 0:
```

```
    digitalWrite(23, HIGH);//a  
    digitalWrite(22, HIGH);//b  
    digitalWrite(21, HIGH);//c  
    digitalWrite(19, HIGH);//d  
    digitalWrite(18, HIGH);//e  
    digitalWrite(5, HIGH);//f  
    digitalWrite(15, LOW);//g
```

```
    break;
```

```
case 1:
```

```
    digitalWrite(23, LOW);//a  
    digitalWrite(22, LOW);//b  
    digitalWrite(21, LOW);//c  
    digitalWrite(19, LOW);//d  
    digitalWrite(18, HIGH);//e  
    digitalWrite(5, HIGH);//f  
    digitalWrite(15, LOW);//g
```

```
    break;
```

```
case 2:
```

```
    digitalWrite(23, HIGH);//a  
    digitalWrite(22, HIGH);//b  
    digitalWrite(21, LOW);//c  
    digitalWrite(19, HIGH);//d  
    digitalWrite(18, HIGH);//e  
    digitalWrite(5, LOW);//f
```

```
digitalWrite(15, HIGH);//g

break;
case 3:
digitalWrite(23, HIGH);//a
digitalWrite(22, LOW);//b
digitalWrite(21, LOW);//c
digitalWrite(19, HIGH);//d
digitalWrite(18, HIGH);//e
digitalWrite(5, HIGH);//f
digitalWrite(15, HIGH);//g
break;
case 4:
digitalWrite(23, LOW);//a
digitalWrite(22, LOW);//b
digitalWrite(21, HIGH);//c
digitalWrite(19, LOW);//d
digitalWrite(18, HIGH);//e
digitalWrite(5, HIGH);//f
digitalWrite(15, HIGH);//g

break;
}
delay(400);
}
}
//*****
*****

// Handler de Inicio página
//*****
*****

void handle_OnConnect() {
```

```
Serial.println("GPIO2 Status: OFF");
server.send(200, "text/html", SendHTML(count, Parqueo1, Parqueo2, Parqueo3, Parqueo4));
}
//*****
*****

// Procesador de HTML
//*****
*****

String SendHTML(uint8_t count, uint8_t FlagParqueo1, uint8_t FlagParqueo2, uint8_t
FlagParqueo3, uint8_t FlagParqueo4) {
    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>Control de Parqueos</title>\n";
    ptr += "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align:
center;}\n";
    ptr += "</style>\n";
    ptr += "</head>\n";
    ptr += "<body>\n";
    ptr += "<h1>ESP32 Web Server &#128664</h1>\n";
    ptr += "<h3>Control de Parqueos</h3>\n";
    //verificacion del valor de contador y de las banderas de estado de los parqueos para así asignarlas
    al web server el estado actual del parqueo y el numero de parqueos disponibles
    if (count == 0)
    {
        ptr += "<h4>Todos los parqueos estan ocupados</h4>\n";
        ptr += "\t\t\t<table style=\"width:100%\">\n";
        ptr += " <tr>\n";
        ptr += " <th>Parqueo1</th>\n";
        ptr += " <th>Parqueo2</th>\n";
        ptr += " <th>Parqueo3</th>\n";
```

```
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 1 and FlagParqueo1 == 0)
{
ptr += "<h4>Existe 1 parqueo disponible<h4>\n";
ptr += "\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";

}
else if (count == 1 and FlagParqueo2 == 0)
{
ptr += "<h4>Existe 1 parqueo disponible<h4>\n";
```

```
ptr += "\t\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";

}
else if (count == 1 and FlagParqueo3 == 0)
{
ptr += "<h4>Existe 1 parqueo disponible<h4>\n";
ptr += "\t\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " </tr>\n";
```

```
ptr += "</table>";

}
else if (count == 1 and FlagParqueo4 == 0)
{
ptr += "<h4>Existe 1 parqueo disponible</h4>\n";
ptr += "\t\t\t<table style='width:100%'\>\n";
ptr += " <tr>\n";
ptr += "  <th>Parqueo1</th>\n";
ptr += "  <th>Parqueo2</th>\n";
ptr += "  <th>Parqueo3</th>\n";
ptr += "  <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += "  <td>#10060;</td>\n";
ptr += "  <td>#10060;</td>\n";
ptr += "  <td>#10060;</td>\n";
ptr += "  <td>#10062;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";

}

else if (count == 2 and FlagParqueo1 == 0 and FlagParqueo2 == 0)
{
ptr += "<h4>Existen 2 parqueos disponibles</h4>\n";
ptr += "\t\t\t<table style='width:100%'\>\n";
ptr += " <tr>\n";
ptr += "  <th>Parqueo1</th>\n";
ptr += "  <th>Parqueo2</th>\n";
ptr += "  <th>Parqueo3</th>\n";
ptr += "  <th>Parqueo4</th>\n";
```

```
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 2 and FlagParqueo1 == 0 and FlagParqueo3 == 0)
{
ptr += "<h4>Existen 2 parqueos disponibles<h4>\n";
ptr += "\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 2 and FlagParqueo1 == 0 and FlagParqueo4 == 0)
{
ptr += "<h4>Existen 2 parqueos disponibles<h4>\n";
ptr += "\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
```



```
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>#10062;</td>\n";
ptr += " <td>#10060;</td>\n";
ptr += " <td>#10060;</td>\n";
ptr += " <td>#10062;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 2 and FlagParqueo2 == 0 and FlagParqueo3 == 0)
{
ptr += "<h4>Existen 2 parqueos disponibles</h4>\n";
ptr += "\t\t\t<table style='width:100%'\>\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>#10060;</td>\n";
ptr += " <td>#10062;</td>\n";
ptr += " <td>#10062;</td>\n";
ptr += " <td>#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 2 and FlagParqueo2 == 0 and FlagParqueo4 == 0)
```

```
{
    ptr += "<h4>Existen 2 parqueos disponibles</h4>\n";
    ptr += "\t\t\t\t<table style='width:100%'\>\n";
    ptr += " <tr>\n";
    ptr += " <th>Parqueo1</th>\n";
    ptr += " <th>Parqueo2</th>\n";
    ptr += " <th>Parqueo3</th>\n";
    ptr += " <th>Parqueo4</th>\n";
    ptr += " </tr>\n";
    ptr += " <tr>\n";
    ptr += " <td>#10060;</td>\n";
    ptr += " <td>#10062;</td>\n";
    ptr += " <td>#10060;</td>\n";
    ptr += " <td>#10062;</td>\n";
    ptr += " </tr>\n";
    ptr += "</table>";
}
else if (count == 2 and FlagParqueo3 == 0 and FlagParqueo4 == 0)
{
    ptr += "<h4>Existen 2 parqueos disponibles</h4>\n";
    ptr += "\t\t\t\t<table style='width:100%'\>\n";
    ptr += " <tr>\n";
    ptr += " <th>Parqueo1</th>\n";
    ptr += " <th>Parqueo2</th>\n";
    ptr += " <th>Parqueo3</th>\n";
    ptr += " <th>Parqueo4</th>\n";
    ptr += " </tr>\n";
    ptr += " <tr>\n";
    ptr += " <td>#10060;</td>\n";
    ptr += " <td>#10060;</td>\n";
    ptr += " <td>#10062;</td>\n";
    ptr += " <td>#10062;</td>\n";
}
```

```
ptr += " </tr>\n";  
ptr += "</table>";  
}
```

```
else if (count == 3 and FlagParqueo1 == 0 and FlagParqueo2 == 0 and FlagParqueo3 == 0)
```

```
{  
ptr += "<h4>Existen 3 parqueos disponibles<h4>\n";  
ptr += "\t\t\t<table style=\"width:100%\">\n";  
ptr += " <tr>\n";  
ptr += " <th>Parqueo1</th>\n";  
ptr += " <th>Parqueo2</th>\n";  
ptr += " <th>Parqueo3</th>\n";  
ptr += " <th>Parqueo4</th>\n";  
ptr += " </tr>\n";  
ptr += " <tr>\n";  
ptr += " <td>&#10062;</td>\n";  
ptr += " <td>&#10062;</td>\n";  
ptr += " <td>&#10062;</td>\n";  
ptr += " <td>&#10060;</td>\n";  
ptr += " </tr>\n";  
ptr += "</table>";  
}
```

```
else if (count == 3 and FlagParqueo1 == 0 and FlagParqueo2 == 0 and FlagParqueo4 == 0)
```

```
{  
ptr += "<h4>Existen 3 parqueos disponibles<h4>\n";  
ptr += "\t\t\t<table style=\"width:100%\">\n";  
ptr += " <tr>\n";  
ptr += " <th>Parqueo1</th>\n";  
ptr += " <th>Parqueo2</th>\n";  
ptr += " <th>Parqueo3</th>\n";  
ptr += " <th>Parqueo4</th>\n";  
ptr += " </tr>\n";
```

```
ptr += " <tr>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 3 and FlagParqueo2 == 0 and FlagParqueo3 == 0 and FlagParqueo4 == 0)
{
ptr += "<h4>Existen 3 parqueos disponibles</h4>\n";
ptr += "\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 3 and FlagParqueo1 == 0 and FlagParqueo3 == 0 and FlagParqueo4 == 0)
{
ptr += "<h4>Existen 3 parqueos disponibles</h4>\n";
ptr += "\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
```

```
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10060;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else if (count == 4)
{
ptr += "<h4>Existen 4 parqueos disponibles</h4>\n";
ptr += "\t\t\t\t<table style=\"width:100%\">\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " <td>&#10062;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
else
{
```

```
ptr += "<h4>Todos los parqueos estan ocupados</h4>\n";
ptr += "\t\t\t\t\t<table style='width:100%'\>\n";
ptr += " <tr>\n";
ptr += " <th>Parqueo1</th>\n";
ptr += " <th>Parqueo2</th>\n";
ptr += " <th>Parqueo3</th>\n";
ptr += " <th>Parqueo4</th>\n";
ptr += " </tr>\n";
ptr += " <tr>\n";
ptr += " <td>#10060;</td>\n";
ptr += " <td>#10060;</td>\n";
ptr += " <td>#10060;</td>\n";
ptr += " <td>#10060;</td>\n";
ptr += " </tr>\n";
ptr += "</table>";
}
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(3000);\n";
ptr += "\n";
ptr += "// -->\n";
ptr += "</script>";
ptr += "</body>\n";
ptr += "</html>\n";
return ptr;
}
//*****
*****
```

```
// Handler de not found
//*****
*****

void handleNotFound() {
    server.send(404, "text/plain", "Not found");
}
//*****
*****

// Obtener el tipo de contenido del archivo
//*****
*****

String GetContentType(String filename)
{
    if (filename.endsWith(".htm")) return "text/html";
    else if (filename.endsWith(".html")) return "text/html";
    else if (filename.endsWith(".css")) return "text/css";
    else if (filename.endsWith(".js")) return "application/javascript";
    else if (filename.endsWith(".png")) return "image/png";
    else if (filename.endsWith(".gif")) return "image/gif";
    else if (filename.endsWith(".jpg")) return "image/jpeg";
    else if (filename.endsWith(".ico")) return "image/x-icon";
    else if (filename.endsWith(".xml")) return "text/xml";
    else if (filename.endsWith(".pdf")) return "application/x-pdf";
    else if (filename.endsWith(".zip")) return "application/x-zip";
    else if (filename.endsWith(".gz")) return "application/x-gzip";
    return "text/plain";
}
//*****
*****

// Enviar al servidor el archivo desde SPIFFS
//*****
*****
```

```
void ServeFile(String path)
{
    File file = SPIFFS.open(path, "r");
    size_t sent = server.streamFile(file, GetContentType(path));
    file.close();
}
//*****
// Enviar al servidor el archivo desde SPIFFS
//*****

void ServeFile(String path, String contentType)
{
    File file = SPIFFS.open(path, "r");
    size_t sent = server.streamFile(file, contentType);
    file.close();
}
//*****

// Handler de not found
//*****

bool HandleFileRead(String path)
{
    if (path.endsWith("/")) path += "index.html";
    Serial.println("handleFileRead: " + path);

    if (SPIFFS.exists(path))
    {
        ServeFile(path);
        return true;
    }
}
```



```
Serial.println("\tFile Not Found");  
return false;  
}
```

Tiva C:

//[Cristopher Sagastume](#) 18640

```
#include <stdint.h>  
#include <stdbool.h>  
#include "inc/tm4c123gh6pm.h"  
#include "inc/hw_types.h"  
#include "inc/hw_memmap.h"  
#include "driverlib/sysctl.h"  
#include "inc/hw_ints.h"  
#include "driverlib/pin_map.h"  
#include "driverlib/debug.h"  
#include "driverlib/gpio.h"  
#include "driverlib/interrupt.h"  
#include "driverlib/timer.h"  
#include "driverlib/systick.h"  
#include "driverlib/uart.h"  
  
bool flag = 0;  
bool flag2=0;  
bool flag3=0;  
bool flag4=0;  
void DatosUart(char *Dat);  
  
int main (void){  
    //Se configura el reloj a 40Mhz  
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
```

```
//Se habilitan los botones como entradas
GPIOPinTypeGPIOInput(GPIO_PORTD_BASE, GPIO_PIN_2);
GPIOPadConfigSet(GPIO_PORTD_BASE, GPIO_PIN_2, GPIO_STRENGTH_2MA,
    GPIO_PIN_TYPE_STD_WPU);
GPIOPinTypeGPIOInput(GPIO_PORTD_BASE, GPIO_PIN_3);
GPIOPadConfigSet(GPIO_PORTD_BASE, GPIO_PIN_3, GPIO_STRENGTH_2MA,
    GPIO_PIN_TYPE_STD_WPU);
GPIOPinTypeGPIOInput(GPIO_PORTD_BASE, GPIO_PIN_6);
GPIOPadConfigSet(GPIO_PORTD_BASE, GPIO_PIN_6, GPIO_STRENGTH_2MA,
    GPIO_PIN_TYPE_STD_WPU);
GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_4);
GPIOPadConfigSet(GPIO_PORTF_BASE, GPIO_PIN_4, GPIO_STRENGTH_2MA,
    GPIO_PIN_TYPE_STD_WPU);

//Se habilitan los led como salidas
GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_2);
GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_3);
GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_4);
GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_5);
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1);
GPIOPinTypeGPIOOutput(GPIO_PORTA_BASE, GPIO_PIN_7);
GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_0);
GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_1);

//Se configura la comunicacion UART
SysCtlPeripheralEnable(SYSCTL_PERIPH_UART1); //Se habilita el UART1
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB); //Se habilita el puerto del UART1
GPIOPinConfigure(GPIO_PB0_U1RX);
GPIOPinConfigure(GPIO_PB1_U1TX);
GPIOPinTypeUART(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1); //definicion de los
pines del UART1
UARTClockSourceSet(UART1_BASE, UART_CLOCK_PIOSC);
//Configuracion de UART0 a 115200 con 8 bits de datos y 1 de stop sin
pariedad
UARTConfigSetExpClk(UART1_BASE, 16000000, 115200, (UART_CONFIG_WLEN_8 |
UART_CONFIG_STOP_ONE | UART_CONFIG_PAR_NONE));
```

```
UARTEnable(UART1_BASE);

//Se coloca como estado inicial los led como Libres
GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0xFF);
GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_7, 0xFF);
GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 0xFF);
GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_1, 0xFF);
while (1)
{
    //Se verifica si se presionó el botón Parqueo1
    if (!GPIOPinRead(GPIO_PORTD_BASE, GPIO_PIN_2))
    {
        SysCtlDelay(500000); //ESPERO 500 MS PARA EL ANTIRREBOTE
        while (!GPIOPinRead(GPIO_PORTD_BASE, GPIO_PIN_2)){ }
        //Si El parqueo esta ocupado se pone Libre
        if (flag == 1)
        {
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_2, 0x00);
            GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0xFF);
            SysCtlDelay(500000);
            DatosUart("a"); //Se envia por UART una cadena diferente para
cada estado de cada semaforo
            SysCtlDelay(500000);
            flag = 0;
        } //si el parqueo esta libre se coloca ocupado
        else if (flag == 0)
        {
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_2, 0xFF);
            GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0x00);
            SysCtlDelay(500000);
            DatosUart("aa"); //Se envia por UART una cadena diferente para
cada estado de cada semaforo
            SysCtlDelay(500000);
            flag = 1;
        }

        //Se verifica si se presionó el botón Parqueo2
        else if (!GPIOPinRead(GPIO_PORTD_BASE, GPIO_PIN_3))
        {
```

```
    SysCtlDelay(500000);    //ESPERO 500 MS PARA EL ANTIRREBOTE
    while (!GPIOPinRead(GPIO_PORTD_BASE, GPIO_PIN_3))
    {
    }//Si El parqueo esta desocupado se pone Libre
    if (flag2 == 1)
    {
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_3, 0x00);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_7, 0xFF);
        SysCtlDelay(500000);
        DatosUart("aaa");//Se envia por UART una cadena diferente para
cada estado de cada semaforo
        SysCtlDelay(500000);
        flag2 = 0;
    }//si el parqueo esta libre se coloca ocupado
    else if (flag2 == 0)
    {
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_3, 0xFF);
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_7, 0x00);
        SysCtlDelay(500000);
        DatosUart("aaaa");//Se envia por UART una cadena diferente para
cada estado de cada semaforo
        SysCtlDelay(500000);
        flag2 = 1;
    }
} //Se verifica si se presionó el botón Parqueo3
else if (!GPIOPinRead(GPIO_PORTD_BASE, GPIO_PIN_6))
{
    SysCtlDelay(500000);    //ESPERO 500 MS PARA EL ANTIRREBOTE
    while (!GPIOPinRead(GPIO_PORTD_BASE, GPIO_PIN_6))
    {
    }//Si El parqueo esta desocupado se pone Libre
    if (flag3 == 1)
    {
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4, 0x00);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 0xFF);
        SysCtlDelay(500000);
        DatosUart("aaaaa");//Se envia por UART una cadena diferente para
cada estado de cada semaforo
        SysCtlDelay(500000);
        flag3 = 0;
    }
```

```
    }//si el parqueo esta libre se coloca ocupado  
    else if (flag3 == 0)  
    {  
        GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_4, 0xFF);  
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 0x00);  
        SysCtlDelay(500000);  
        DatosUart("aaaaaa");//Se envia por UART una cadena diferente para  
cada estado de cada semaforo  
        SysCtlDelay(500000);  
        flag3 = 1;  
    }  
  
    }//Se verifica si se presionó el botón Parqueo4  
    else if (!GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_4))  
    {  
        SysCtlDelay(500000);    //ESPERO 500 MS PARA EL ANTIRREBOTE  
        while (!GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_4))  
        {  
        }  
        }//Si El parqueo esta desocupado se pone Libre  
        if (flag4 == 1)  
        {  
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_5, 0x00);  
            GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_1, 0xFF);  
            SysCtlDelay(500000);  
            DatosUart("aaaaaaa");//Se envia por UART una cadena diferente  
para cada estado de cada semaforo  
            SysCtlDelay(500000);  
            flag4 = 0;  
        }  
        }//si el parqueo esta libre se coloca ocupado  
        else if (flag4 == 0)  
        {  
            GPIOPinWrite(GPIO_PORTA_BASE, GPIO_PIN_5, 0xFF);  
            GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_1, 0x00);  
            SysCtlDelay(500000);  
            DatosUart("aaaaaaaa");//Se envia por UART una cadena diferente  
para cada estado de cada semaforo  
            SysCtlDelay(500000);  
            flag4 = 1;  
        }  
    }
```

```
    }  
}  
  
} //Funcion que nos servirá para enviar datos (verifica las banderas de la uart y  
envia datos)  
void DatosUart(char *Dat){  
    while(UARTBusy(UART1_BASE));  
    while(*Dat != '\0')  
    {  
        UARTCharPut(UART1_BASE, *Dat++);  
    }  
}
```

Repositorio:

https://github.com/sag18640/Electronica_Digital_2.git