

Universidad del Valle de Guatemala

Electrónica Digital II

Ing. Pablo Mazariegos

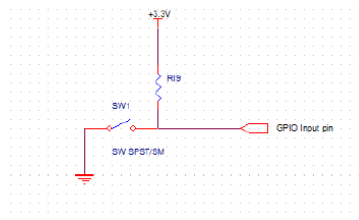
Rodrigo Díaz, 18265

Sección 20

Mini#4: Web Server

Circuitos Utilizados

Para los circuitos utilizados se utilizaron botones en configuración pullup, conexión de LEDs con su resistencia y conexión de UART.



Circuito Pullup

Datos

La planificación inicial del proyecto era tener los 4 pushbuttons y solamente mientras estuvieran presionados se marcaba como ocupado. Sin embargo, se utilizaron una serie de if y else if para poder realizar esta lógica y se mandaban demasiados datos y no se podían leer por UART. Entonces, se tuvo que modificar para que solamente se mande un dato cuando se presiona un botón y ahora el botonazo alterna entre ocupado y libre. De esta manera, se envían los datos de una manera más ordenada. Para poder realizar el toggle de ocupado vs libre, se utilizaron banderas que alternaban entre 0 y 1. Inicialmente están en 0 y al primer botonazo se convierte a 1.

En cuanto a la programación del ESP, el WebServer se creó sin problemas utilizando las guías de creación de tablas de Bootstrap. Se implementó una tabla con escala de colores claros, pero el header es oscuro. La casilla del total de parqueos está resaltada y las casillas de ocupado y libre cambian de color. También se tiene una descripción al pie de la tabla. Finalmente se añadieron emojis para que se mire más amigable. En cuanto a la comunicación UART, se tuvo problemas al recibir datos que variaban en longitud, entonces se decidió solamente utilizar una sola letra por comando y pasar los datos por una serie de ifs para determinar que variable debe cambiar. El valor que se lee por UART se guarda en una variable de tipo char para evitar problemas. Finalmente, la función de SendHTML tiene como inputs todas las variables de los parqueos individuales y el total de parqueos. Como también se implementó una rutina para que la página hiciera refresh cada segundo, esta se actualiza en tiempo real. Como ya se tenía una variable para el total de parqueos disponibles, solo se utiliza un switch case para utilizar el 7seg.

Gráficos

En cuanto al Web Server, primero se tomó como base el ejemplo de Bootstrap4. El título de la página viene acompañado de unos emojis. Se añadió una caption debajo de la tabla para que sepan que pertenece al primer nivel. La tabla se crea para que sea de colores claros, pequeña, texto centrado y con bordes. Sin embargo, después se modifica la head de la tabla por separado para que tenga un color oscuro. La primera columna de la tabla es el número de parqueos con un emoji de carro parqueado y la segunda es el estado acompañado de un emoji de indicador. Al comenzar el body de la tabla, se implementan ifs dependiendo de cada variable de cada parqueo. Si está en 1 está ocupado y 0 está libre. Dependiendo de que valor tenga cada variable de parqueo, cuales casillas se despliegan en la segunda columna. Finalmente se añade un foot a la tabla, donde se despliega el total de parqueos disponibles. Esta despliega el número del total de parqueos, que ya fue calculado y solamente es input de la función.

Parqueo-Matic 📋 🚌 🚗	
# Parqueo 🚗	Estado 🚦
1	Disponible 🟢
2	Disponible 🟢
3	Disponible 🟢
4	Disponible 🟢
Total Disponible	4

Control de parqueos: Primer Nivel

Web Server Parqueo-Matic

Explicaciones

Primero, la TIVA controla los sensores, que en este caso son los pushbuttons, para saber cuando están ocupados o desocupados los parqueos. Los botones funcionan como toggle, para comunicarse efectivamente por UART. Esto se guarda en variables y se utilizan banderas para realizar el toggle. Solamente se mandan datos si se presiona un botón y cada botón puede mandar dos letras, dependiendo si necesita ocuparse o desocuparse. Después estos datos se comunican por medio de UART a el ESP32 este agarra los datos y los compara utilizando una seria de ifs para poder determinar cual variable es la que tiene que cambiar. Después de esto, se calcula el total de parqueos disponibles al restar de 4 el número de parqueos ocupados porque 4 es el total. Se mandan los datos de cada parqueo y el total a SendHTML para que sepa que imprimir en la tabla del Web Server. Finalmente, se agarra el dato del total de espacios libres y se utiliza un switch case para determinar que número desplegar en el 7 segmentos.

Código Comentado

Viene incluido en el zip, muy largo para poner acá.

Links

Video: <https://youtu.be/0uQp8J90q24>

Github: https://github.com/RodDia2/Labs_Digital_2