Universidad del Valle de Guatemala           1 de junio del 2023

Facultad de Ingeniería Electrónica

electrónica Digital 2

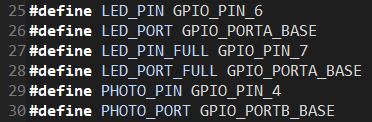
Catedrático Pablo Mazariegos

Aldo Renato Avila Escobar – 21257

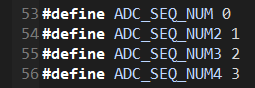
Byron Estuardo Barrientos Perez - 21457

**Entregable Final**

**Explicación del código**

****

Se declaran los puertos y pines que se usaran para las fotorresistencias y los leds que indicaran si el espacio se encuentra disponible o no. Declararlos de esta forma ayuda a que, si se necesitara hacer un cambio en los pines a usar, solo se tendrá que cambiar en esta sección del código para que sea efectivo. Esta misma declaración se hizo tres veces mas para las otras fotorresistencias.

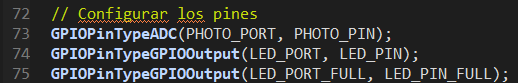


Nos ayuda a saber la secuencia en la que se ejecutara cada canal del ADC.

A screen shot of a computer program

Description automatically generated with low confidence

El setup del código comienza con la configuración del oscilador, en este caso se usará uno interno usando el PLL. De igual forma se habilita los tiempos de reloj para cada uno de los puertos a usar.



Se configuran los pines que se usaran como salidas y como ADC. Esto se repite tres veces más para cada uno de los canales ADC.

A picture containing text, screenshot, software, font

Description automatically generated

Aquí se configura el ADC, cuando es que se va a dar la lectura y que canales son los que se van a utilizar. En la parte de abajo se habilita la secuencia en la que se estarán realizando las lecturas.

A picture containing text, screenshot, font, software

Description automatically generated

Se habilita y configura el UART a utilizar, en este caso se utilizará el UART1, se configuran los pintes 0 y 1 del puerto b como RX y TX respectivamente. Se tuvo que especificar que estos pines se iban a usar para el UART1 debido a que este UART esta multiplexado con otro par de pines. No se utilizó una interrupción ya que solo se iban a estar enviando datos, con un baudrate de 115200 sin paridad.

A picture containing text, screenshot, font

Description automatically generated

En el loop principal se llevan a cabo las lecturas de cada uno de los canales del ADC0 indicados. Mientras se realiza la lectura, el código se queda en ciclo while, para después el dato ser guardado en la variable indicada, como por ejemplo “photo\_value”.

A picture containing text, screenshot, font

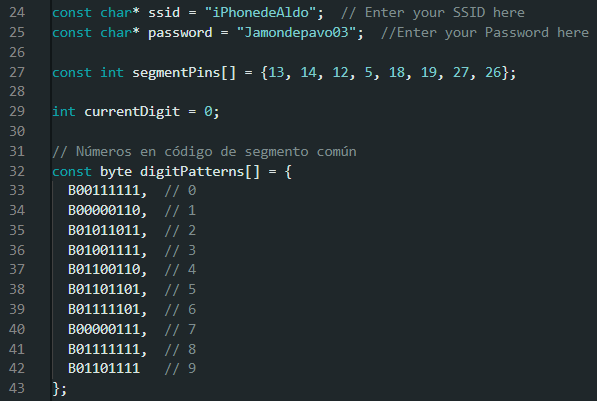
Description automatically generated

Para cada una de las variables se realiza una comparación. En caso de que la lectura sea mayor a 1500, la LED que indica que el espacio esta libre, estará encendida. Esto también quiere decir que la fotorresistencia se encuentra recibiendo algún tipo de luz. Además de encender la LED, la tiva enviara un carácter ‘a’ por UART para indicarle al ESP que este espacio esta libre. Si la lectura es menor a 1500 el LED que indica que el espacio está ocupado se encenderá y se enviará un carácter ‘b’ para indicarle al ESP que el espacio está ocupado. Cabe resaltar que esto ocurrirá cuando la fotorresistencia no este recibiendo algún tipo de luz.

A screen shot of a computer

Description automatically generated with low confidence

En la parte del ESP32 se comienza declarando que pines serán usados para cada UART y cual es el RX y TX. También los baudrate, en este caso es de 115200 para coincidir con el colocado en la Tiva C.



Aquí se declaran las variables a usar, en este caso también se coloca el nombre de la red wifi a usar con su contraseña. También se declaran los pines a usar para el display de 7 segmentos con el contador que nos dirá cual es el numero de parqueos disponibles. También está la lista con los bits que nos indicaran que pines se tendrán que encender para que se despliegue el numero deseado. Cabe resaltar que estos están en el orden contrario que los puertos por como se iba asignando cada valor a su respectivo pin.

A screen shot of a computer

Description automatically generated with medium confidence A screen shot of a computer

Description automatically generated with medium confidence

Se declaran las variables en las cuales se guardaran los valores obtenidos del UART. La variable parqueo1 obtendra los datos del UART1 y la variable parqueo2 los datos del UART2. Las variables que comienzan con Flag nos ayudaran con el contador, serviran de antirrebote. El webserver esta por defaul el 80.

A screen shot of a computer

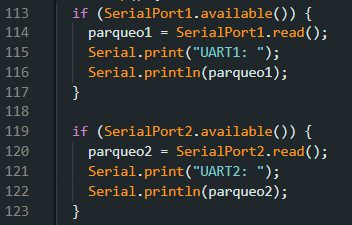
Description automatically generated with low confidence

Se configuran e inician los UART a utilizar, también se colocan los pines a utilizar en el display de 7 segmentos como salidas.

A picture containing text, screenshot, font

Description automatically generated

Se inicia nuestro servidor WIFI en el ESP32 y pedimos que mientras se conecta a la red se mantenga en el ciclo while para asegurar la conexión. De igual forma pedimos que imprima en la consola la dirección IP de la pagina para poder accederla luego.



Se leen los datos del puerto UART y se almacenan en su respectiva variable.

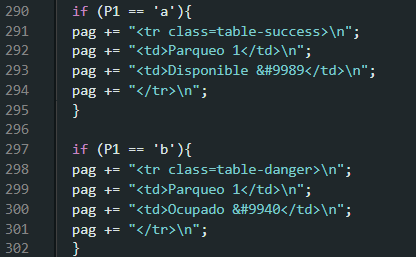
A screen shot of a computer program

Description automatically generated with low confidence

Esta será una comparación que se de para cada uno de los parqueos, así aseguramos que el programa tenga una variable por parqueo. En esta verificamos el carácter que recibe el ESP, y si es igual al que se encuentra comparando en el if, lo guardara en la variable respectiva al parqueo; hay desde P1 hasta P8. También están las comparaciones que sumaran o restaran al contador, junto on la bandera que sube o baja dependiendo de si el parqueo esta ocupado o no, esta nos ayudara a que no se este sumando constantemente cada vez que entre al if.



En esta parte del código se entrará a una función la cual coloque en el display de 7 segmentos el valor de la variable currentDigit.



En esta parte del código ya se esta creando el servidor en base al String SendHTML(). Estas comparaciones se realizan para que en la tabla se muestre si se encuentra libre o no el espacio.

A screenshot of a computer

Description automatically generated with low confidence

Esta sería la página resultante del servidor en la cual se observa que espacios se encuentran libres y cuales ocupados.

**Links:**

<https://youtu.be/oUMaKEWcxOY>

<https://github.com/Aldoa912/Proyecto_parqueo.git>