

UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Reporte de práctica

Alarma para casa



Presenta:

ZAHARA NATHALIA IBARRA CASILLAS
KEVIN EDUARDO LÓPEZ MARTÍNEZ

19398 - D01
Seminario de Solución de Problemas de Programación de Sistemas Embebidos
Hecho en L^AT_EX

Índice

1. Introducción	2
1.1. Objetivos	2
2. Desarrollo	2
2.1. Marco teórico	2
2.2. Diagrama esquemático	3
2.3. Diagrama de flujo	4
2.4. Metodología/Procedimiento	5
3. Resultados	5
4. Conclusión	7

1. Introducción

Antes de realizar esta práctica, previamente habíamos realizado actividades. Por ejemplo, habíamos usado un display de 16 segmentos para mostrar los nombres del equipo en el display y también habíamos usado el multiplexado de displays cuando hicimos un contador de 0 a 9999. Con estas actividades particulares logramos incluirlas para poder realizar esta alarma.

1.1. Objetivos

- (i) Leer el encendido y apagado de los switch.
- (ii) Multiplexar los displays.
- (iii) Agregar un LED que alerte la actividad de la alarma.

2. Desarrollo

Primero, veremos unos conceptos fundamentales para comprender el por qué de las funciones mencionadas y el desarrollo de nuestro código.

2.1. Marco teórico

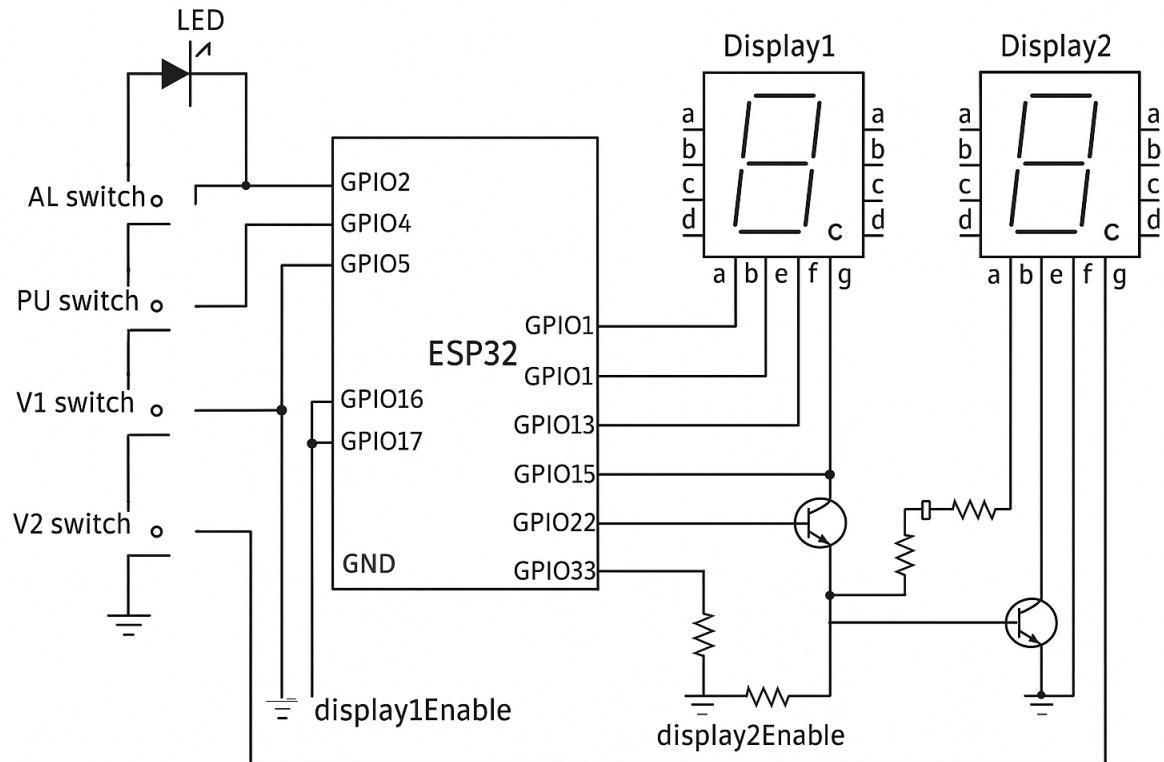
- (i) ¿Cómo funciona la resistencia pull-down?

En electrónica, una resistencia pull-down (PD) es una resistencia que se utiliza para fijar una señal a un nivel lógico bajo (0) cuando no hay otra fuerza que la mantenga en un estado alto (1). Esto es particularmente útil en entradas de circuitos digitales para asegurar que el pin de entrada no quede en un estado flotante o indeterminado cuando no está siendo impulsado por una fuente externa.

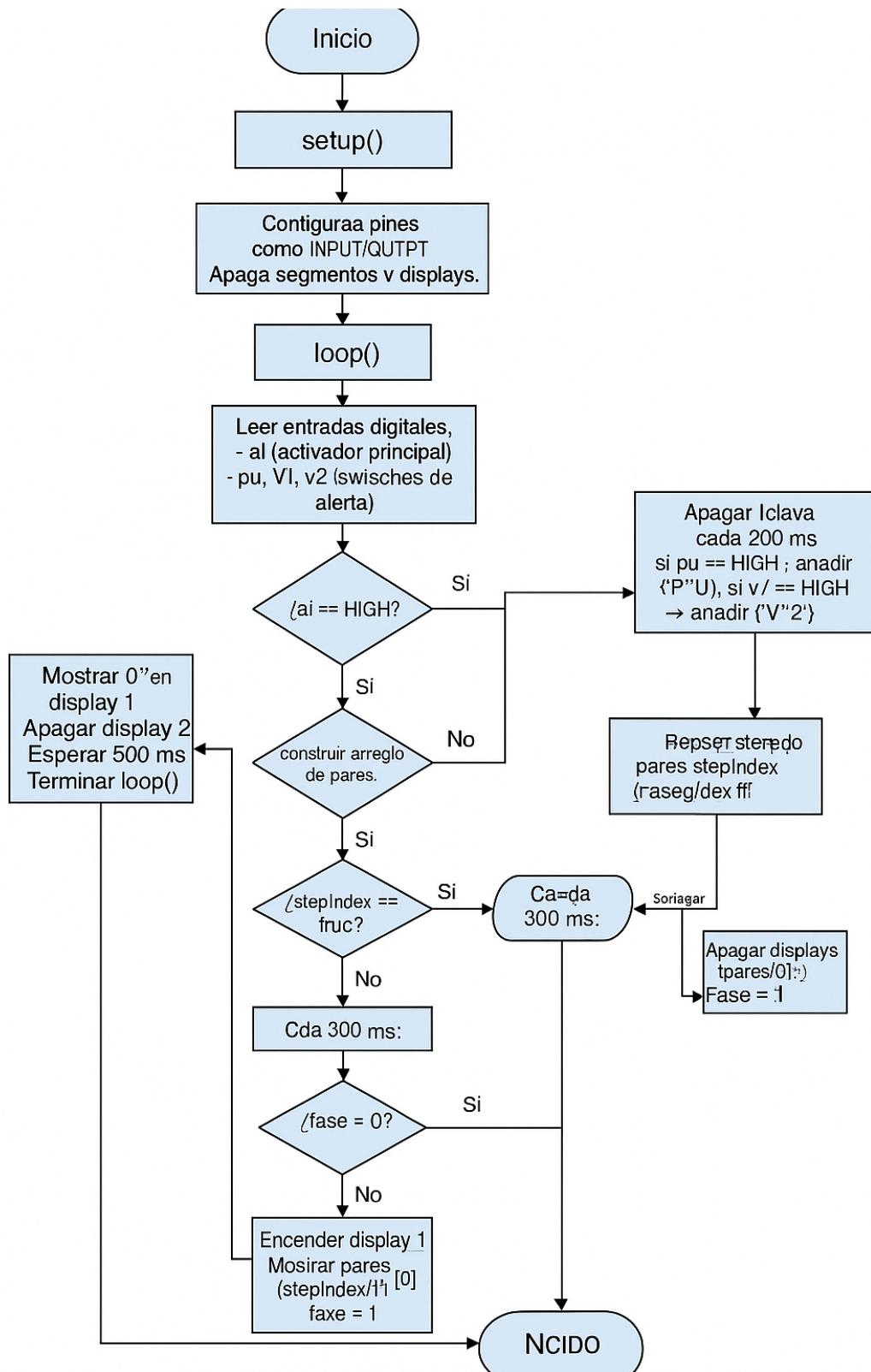
- (ii) ¿En qué consiste la multiplexación?

La multiplexación, o "muxing." en inglés, es una técnica que permite transmitir múltiples señales de información a través de un mismo canal de comunicación simultáneamente. En esencia, se trata de combinar varias señales en una sola, transmitirlas y luego separarlas nuevamente en el receptor, optimizando el uso del recurso de transmisión.

2.2. Diagrama esquemático



2.3. Diagrama de flujo



2.4. Metodología/Procedimiento

Para comenzar, definimos los pines del LED, los switch (.^L", "PU", "V2", "V1") y los displays de 16 segmentos. También asignamos unas variables de control.

Creamos dos funciones:

.^apagarSegmentos(): donde apaga todos los segmentos del display, poniéndolos en LOW.

"mostrarCaracter(char c)": que activa ciertos segmentos dependiendo del carácter a mostrar (letras y números), según el diseño del display de 16 segmentos.

En la función setup(): configuramos los pines como entradas o salidas y apaga todos los segmentos y displays al inicio..

En la función loop(): lee las entradas, detecta si hay una alerta activa (pu, v1, v2) y si el sistema está habilitado (al).

Si el sistema no está activado (al == LOW), apaga el LED, displays, resetea índices de control y no muestra nada.

Si el sistema esta activado (al == HIGH), parpadea el LED si hay alguna alerta activa (pu, v1, v2), el LED parpadea cada 200 ms.

Si no hay alertas activas, el LED se queda encendido fijo.

Construcción del arreglo "pares[][]": si "pu == HIGH", se agrega ['P', 'U'], si "v1 == HIGH", se agrega ['V', '1'], y si "v2 == HIGH", se agrega ['V', '2']. Y si no hay alertas activas (totalPares == 0), muestra un '0' en el primer display y apaga el segundo.

En la multiplexación de displays cada 300 ms, alterna entre los displays para mostrar un carácter del par:

"fase == 0."enciende display 1 y muestra el primer carácter del par.

"fase == 1."enciende display 2 y muestra el segundo carácter del par.

Luego de cada par, incrementa el índice (stepIndex) para mostrar el siguiente en la siguiente iteración. Cuando se llega al último par, regresa al primero "(stepIndex totalPares)".

3. Resultados

A continuación, añadimos algunas fotos del microcontrolador con el código ya cargado.

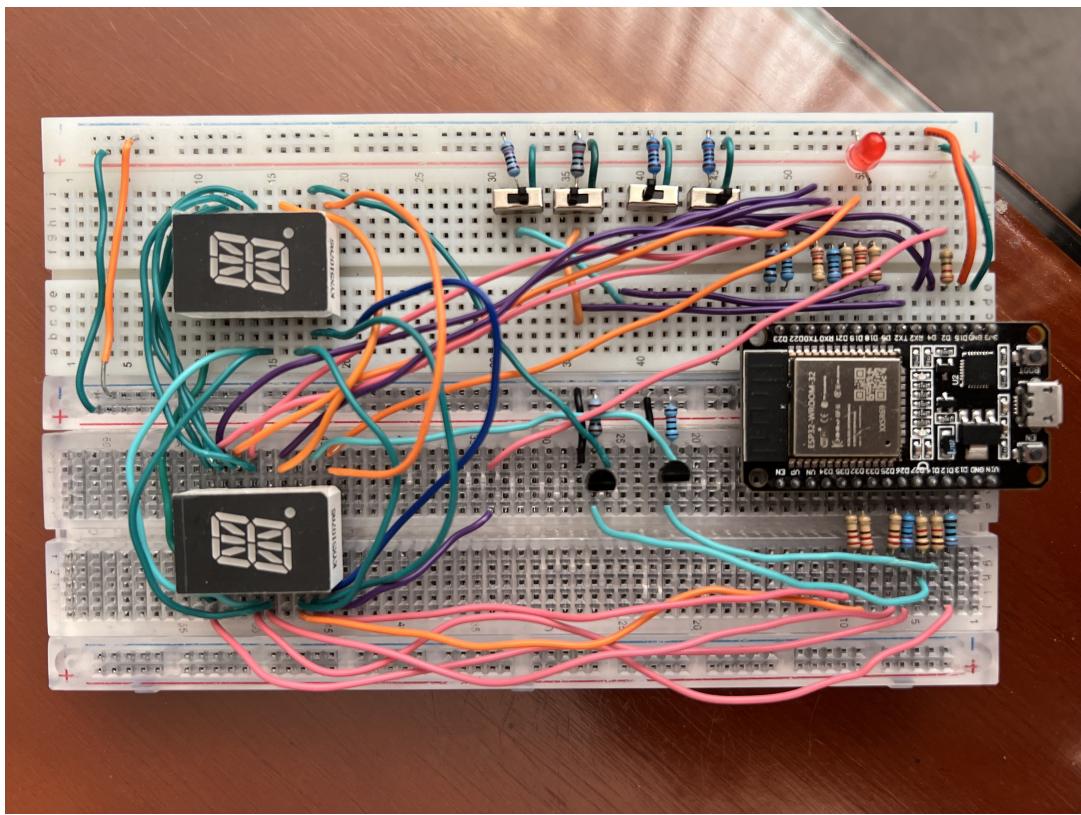


Figura 1: Foto de la alarma terminada.

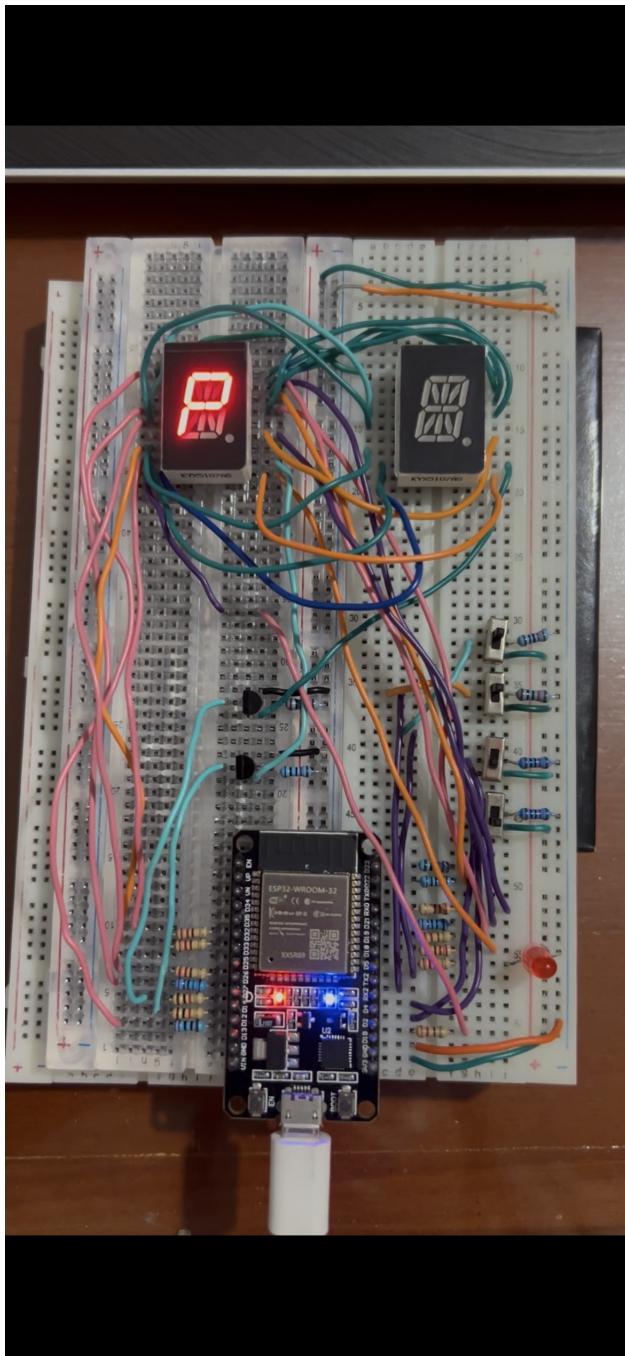


Figura 2: Otra foto de la alarma terminada.

4. Conclusión

Implementamos un sistema de alarma con visualización dinámica en dos displays de 16 segmentos, controlado por una ESP32. Uno de los elementos más destacados del diseño es el uso de multiplexación por tiempo, mediante el cual se alterna el encendido entre dos displays

utilizando transistores controlados por los pines display1Enable y display2Enable. Esto permite compartir los mismos pines de segmentos entre ambos displays, optimizando el número de GPIOs utilizados en el ESP32.

La multiplexación se logra mediante el uso de la variable fase, que alterna entre mostrar el primer carácter de un par (pares[[0]]) en el primer display y el segundo carácter (pares[[1]]) en el segundo display. Un temporizador basado en millis() controla la frecuencia de conmutación (intervalo = 300 ms), manteniendo una persistencia visual adecuada para el usuario.

Además, el sistema utiliza una técnica de pull-down de resistencias implícita. Al leer el estado de los pulsadores (PU, V1, V2, AL) conectados a entradas digitales (INPUT), se espera que cada pin se mantenga en nivel bajo (LOW) cuando el botón está sin presionar, gracias al uso de resistencias externas pull-down o pull-down internas si están habilitadas (aunque no se especifican en el código). Esta configuración evita lecturas flotantes e inestables, asegurando una detección precisa del estado de los switches.

Referencias

- [1] Feria de tecnología.*Resistencias Pull up y Pull down*[En línea]. Disponible en:
<https://ditecnomakers.com/resistencias-pull-up-y-pull-down/>: :text=Las
- [2] R. Awati, R. Sheldon, J. Burke. *¿Qué es la multiplexación y cómo funciona?*[En línea]. Disponible en: https://www-tchtarget-com.translate.goog/searchnetworking/definition/multiplexing?xtrsl=en_xtrtl=es_xtrhl=es_xtrpto=sge::text=La