

Universidad de Guadalajara

Piensa y trabaja



REPORTE DE PRÁCTICA

Práctica 4 (Contador en formato de reloj)

Participantes:

Lorena Villalobos Carrillo

Ximena Lucía Rodríguez Oliva

Brayam De Jesus De La Cerda Valdivia

Seminario de problemas de programación embebidos

(D01 –I9893)

Fecha: 18/02/2025

1. Resumen

En esta práctica se desarrolló un contador en formato de horas utilizando una tarjeta de desarrollo ESP32 y dos displays de 7 segmentos dobles. El sistema muestra la hora en formato de 00:00 a 23:59 de manera cíclica. Se implementó un algoritmo en C++ dentro del entorno de desarrollo Arduino IDE, configurando los pines del ESP32 para manejar los segmentos de los displays y controlar la actualización de los dígitos. Esta práctica permitió comprender el manejo de displays de 7 segmentos y la programación de temporizadores en microcontroladores.

2. Objetivos

- Implementar un contador de horas utilizando un ESP32 y displays de 7 segmentos.
- Aprender a controlar múltiples segmentos de un display utilizando multiplexación.
- Comprender la lógica de control de tiempos en microcontroladores.
- Familiarizarse con la programación en Arduino IDE para sistemas embebidos.

Universidad de Guadalajara

Piensa y trabaja

3. Introducción

Los sistemas embebidos desempeñan un papel fundamental en la automatización y control de procesos en el ámbito de la ingeniería. Uno de los componentes esenciales en estos sistemas son los displays de 7 segmentos, los cuales permiten representar información numérica de forma sencilla y efectiva.

En esta práctica, se diseñó un contador de horas que avanza desde las 00:00 hasta las 23:59, utilizando un ESP32 como unidad de control. El sistema emplea dos displays de 7 segmentos dobles para representar las horas y los minutos. La actualización de los dígitos se realiza mediante un método de multiplexación, activando cada dígito de forma secuencial para optimizar el uso de pines del microcontrolador.

4. Marco Teórico

4.1 Tarjeta de desarrollo ESP32

El ESP32 es una tarjeta de desarrollo de bajo costo y alto rendimiento desarrollado por Espressif Systems. Cuenta con conectividad Wi-Fi y Bluetooth, así como una gran cantidad de pines GPIO para la interacción con periféricos. Su uso en sistemas embebidos es común debido a su potencia de procesamiento y versatilidad.

4.2 Displays de 7 Segmentos

Los displays de 7 segmentos son dispositivos de salida utilizados para representar números. Cada dígito está compuesto por siete LEDs dispuestos en forma de "8", que pueden encenderse de manera independiente para formar diferentes números. En esta práctica, se usaron dos displays dobles para mostrar las horas y los minutos.

Universidad de Guadalajara

Piensa y trabaja

4.3 Multiplexación

La multiplexación es una técnica utilizada para reducir el número de pines requeridos al controlar varios dígitos de un display. En lugar de encender todos los dígitos simultáneamente, se activan de manera secuencial a alta velocidad, lo que permite que el ojo humano perciba la información como si estuviera fija.

5. Metodología

5.1 Materiales y Herramientas

- ESP32
- Dos displays de 7 segmentos dobles
- Resistencias de multiples ohmios
- Protoboard y cables de conexión
- Computadora con Arduino IDE

5.2 Explicación del Código

El código está estructurado en varias partes clave:

- **Inicialización y Configuración:** Se definen los pines del ESP32 asociados a los segmentos y dígitos del display.

```
const int segmentos[] = {14, 4, 5, 19, 18, 13, 21};  
const int digitos[] = {23, 22, 25, 26};
```

Estas variables se declaran como const porque su contenido no cambia durante la ejecución del programa. Esto optimiza el uso de memoria y evita modificaciones accidentales en el código.

- **Gestor de Tiempo:**

```
while (millis() - lastMillis < w1s) {  
    mostrarNumero(horas, minutos);  
}
```

Este fragmento de código gestiona el tiempo de actualización de la pantalla. La función `millis()` devuelve el tiempo transcurrido desde el inicio del programa, asegurando que la función `mostrarNumero()` se ejecute durante un tiempo controlado antes de actualizar los minutos.

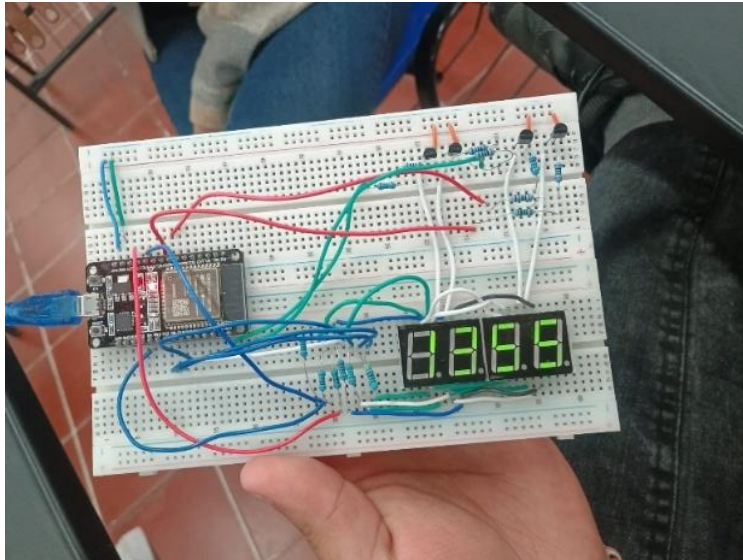
- **Multiplexación de los Dígitos:**

```
for (int j = 0; j < 15; j++) {  
    for (int i = 0; i < 4; i++) {  
        mostrar(cifras[i], digitos[i]);  
        delay(2);  
        apagar();  
    }  
}
```

Este fragmento maneja la visualización de los dígitos de forma secuencial:

- El primer `for (j)` asegura que cada dígito se muestre de manera repetida para evitar parpadeos.
- El segundo `for (i)` recorre los dígitos (horas y minutos) para mostrarlos secuencialmente.
- `mostrar()` enciende el dígito correspondiente, `delay(2)` permite su visualización y `apagar()` lo desactiva antes de mostrar el siguiente.

6. Resultados



7. Conclusión

Esta práctica permitió comprender el funcionamiento de los displays de 7 segmentos y su implementación en sistemas embebidos con ESP32. Se logró diseñar un contador de horas eficiente que muestra la información de manera clara y precisa en los displays, utilizando la técnica de multiplexación para optimizar el uso de los pines del microcontrolador.

La programación del ESP32 permitió implementar un control de tiempo preciso mediante la función `millis()`, asegurando que el sistema avanzara correctamente en el conteo de minutos y horas. Además, se optimizó la visualización evitando parpadeos no deseados a través del control secuencial de los dígitos.

Como dificultades, se presentaron algunos problemas iniciales con la configuración de los pines y la multiplexación, los cuales fueron corregidos con ajustes en el código. Este ejercicio proporcionó un conocimiento valioso sobre el control de periféricos y la gestión de tiempo en microcontroladores, sentando una base sólida para futuros proyectos más complejos en sistemas embebidos.