

Universidad de Guadalajara

Piensa y trabaja



REPORTE DE PRÁCTICA

Práctica 6: Multímetro con ESP32

Participantes:

Lorena Villalobos Carrillo

Ximena Lucía Rodríguez Oliva

Brayam De Jesus De La Cerda Valdivia

Seminario de problemas de programación embebidos
(D01 –I9893)

Universidad de Guadalajara

Piensa y trabaja

1. Introducción

En esta práctica se diseñó y construyó un prototipo funcional de multímetro digital utilizando el microcontrolador ESP32, con el objetivo de medir dos tipos de magnitudes: el voltaje en una entrada analógica, y el nivel de iluminación a través de una fotoresistencia (LDR). La medición se visualiza en un display de 7 segmentos de dos dígitos, controlado mediante multiplexación, lo que permite al usuario observar los valores sin necesidad de una interfaz gráfica compleja.

Uno de los enfoques principales de esta práctica fue integrar múltiples conceptos aprendidos en clase, como la lectura de entradas analógicas, el uso de estructuras para manejar visualización numérica en hardware, y la programación de la lógica de conmutación entre modos de operación mediante un botón. Además, se consideró la eficiencia del sistema, la estabilidad de lectura y la escalabilidad del código.

El ESP32 es un microcontrolador versátil con múltiples entradas analógicas y capacidades de procesamiento suficientes para manejar tareas de visualización y lectura de sensores en tiempo real. Gracias a estas características, se convirtió en la plataforma ideal para este proyecto educativo.

2. Desarrollo

2.1 Materiales y herramientas

Para llevar a cabo la práctica se utilizaron los siguientes materiales:

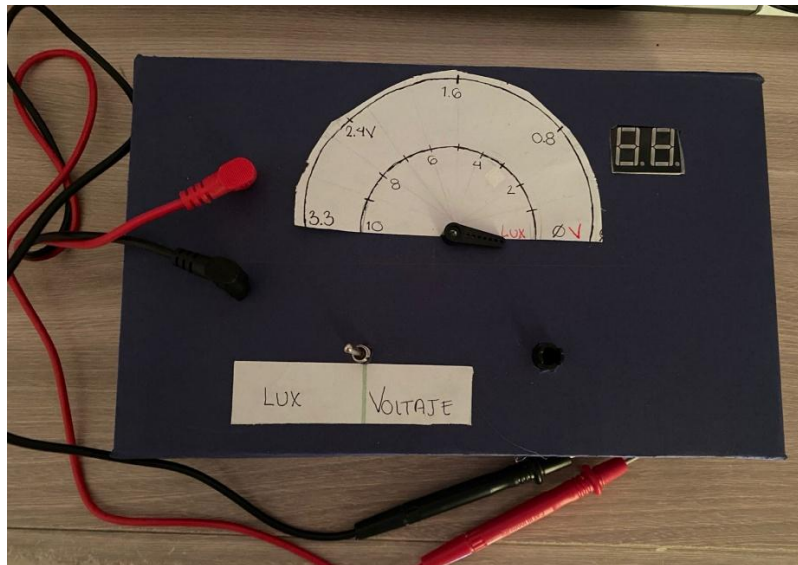
- Microcontrolador ESP32
- Display de 7 segmentos de 2 dígitos (ánodo común)
- Fotoresistencia (LDR)
- Resistencias de 220Ω y $10k\Omega$
- Botón pulsador (switch)
- Cables de conexión
- Protoboard
- Fuente de alimentación USB
- Computadora con Arduino IDE configurado para ESP32

2.2 Configuración del sistema

El diseño se basa en los siguientes elementos:

- Entrada analógica de voltaje: conectada al pin 34 del ESP32. Se pueden conectar fuentes externas para simular voltajes.
- Entrada analógica del fotómetro: mediante una LDR conectada al pin 32, formando un divisor de voltaje con una resistencia fija.
- Switch de selección de modo: conectado al pin digital 5, usando la resistencia pull-up interna del microcontrolador.
- Display de 7 segmentos: controlado por 8 pines digitales (segmentos A-G y punto decimal) y 2 pines para el encendido de cada dígito.

El código implementado permite alternar dinámicamente entre modos, detectar cambios, filtrar valores fuera de rango, y actualizar el display con una multiplexación rápida que evita parpadeos perceptibles.



2.3 Lógica de funcionamiento

El sistema desarrollado utiliza una estructura de control basada en el estado del botón pulsador para alternar entre dos modos de funcionamiento. Esta lógica puede dividirse en los siguientes bloques funcionales:

a) Lectura de voltaje

Se usa el pin analógico 34 para capturar la señal de voltaje. Esta lectura se convierte a voltios reales aplicando el factor de escala del ADC del ESP32 (resolución de 12 bits: 0-4095). Se agregó un pequeño ajuste de calibración para mejorar la precisión.

Se redondea el valor a un decimal y se limita inferiormente a 0.0 V para evitar lecturas residuales. Este valor se convierte en dos dígitos (por ejemplo, 3.3 → 33) y se presenta en el display.

b) Lectura de luz

Se emplea un divisor de voltaje entre la LDR y una resistencia de 10kΩ para traducir la intensidad lumínica en una señal analógica. Se mapea el rango de lectura del sensor a una escala de 0.0 a 9.9. Para mantener la visualización clara y sencilla, se presenta también en dos dígitos (por ejemplo, 7.8 → 78).

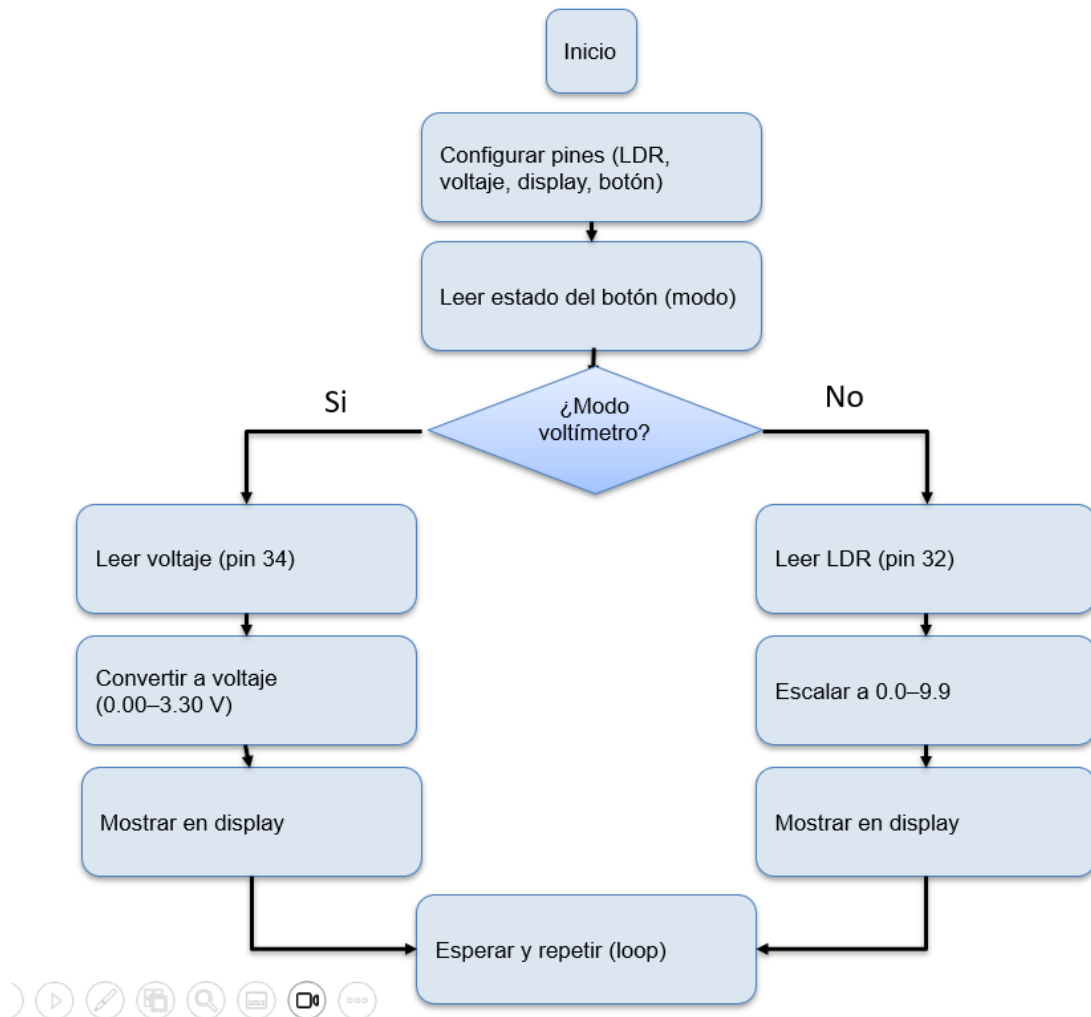
El mapeo se realiza con la función `map()`, ajustando los valores extremos del ADC obtenidos experimentalmente (lecturas mínimas en oscuridad y máximas en luz intensa).

c) Multiplexación del display

Para controlar un display de 2 dígitos con un número limitado de pines, se implementa multiplexación. Se alterna rápidamente entre los dígitos activando uno a la vez, mientras se encienden los segmentos correspondientes. El ojo humano percibe ambos dígitos como encendidos simultáneamente debido a la persistencia de visión.

El punto decimal solo se activa en el primer dígito, permitiendo mostrar valores con una precisión de una cifra decimal.

d) Diagrama de flujo



3. Resultados y observaciones

Durante las pruebas, se logró con éxito:

- Alternar entre modos con el botón sin interferencias.
- Medir con precisión voltajes analógicos entre 0 y 3.3 V.
- Detectar variaciones de luz con buena sensibilidad en interiores.
- Visualizar los valores en tiempo real en el display de 7 segmentos.
- Evitar parpadeos o fallas visuales gracias a la multiplexación eficiente.

También se observaron algunos desafíos:

- Las fluctuaciones rápidas en la lectura analógica de la LDR requerían estabilización. Se sugiere aplicar un filtro o promedio móvil.
- El voltaje mínimo del ESP32 no es exactamente cero, por lo que fue necesario corregir el valor base en el código.
- En condiciones de muy baja luz, el rango de la LDR puede reducirse drásticamente, afectando la precisión de los valores bajos.

4. Conclusión

La práctica permitió reforzar múltiples habilidades técnicas relevantes en el diseño de sistemas embebidos:

- Lectura y procesamiento de señales analógicas
- Control y visualización de datos con hardware limitado
- Multiplexación de displays de 7 segmentos
- Interacción de usuario mediante botones y lógica de control
- Manejo eficiente de recursos en microcontroladores

El proyecto demostró que es posible construir un multímetro digital básico y funcional utilizando componentes económicos y un ESP32. Esta solución puede escalarse fácilmente para incluir sensores adicionales, pantallas más avanzadas o almacenamiento de datos.

Universidad de Guadalajara

Piensa y trabaja

5. Código

```
#define analogPin 34 // Entrada de voltaje

#define ldrPin 32 // Entrada de fotoresistencia

#define switchPin 5 // Pin del switch para cambiar modo

int LEDS[] = {14, 12, 27, 33, 25, 13, 22, 18}; // A, B, C, D, E, F, G, DP

int digitos[] = {15, 2}; // Unidades, decenas

const byte numeros[10][7] = {

    {1, 1, 1, 1, 1, 1, 0}, // 0

    {0, 1, 1, 0, 0, 0, 0}, // 1

    {1, 1, 0, 1, 1, 0, 1}, // 2

    {1, 1, 1, 1, 0, 0, 1}, // 3

    {0, 1, 1, 0, 0, 1, 1}, // 4

    {1, 0, 1, 1, 0, 1, 1}, // 5

    {1, 0, 1, 1, 1, 1, 1}, // 6

    {1, 1, 1, 0, 0, 0, 0}, // 7

    {1, 1, 1, 1, 1, 1, 1}, // 8

    {1, 1, 1, 1, 0, 1, 1} // 9

};

bool modoAnterior = false;

void setup() {

    Serial.begin(115200);

    for (int i = 0; i < 8; i++) pinMode(LEDS[i], OUTPUT);
```

Universidad de Guadalajara

Piensa y trabaja

```
for (int i = 0; i < 2; i++) pinMode(digitos[i], OUTPUT);

pinMode(switchPin, INPUT_PULLUP);

}

void loop() {

    bool modoVoltmetro = digitalRead(switchPin) == LOW;

    if (modoVoltmetro != modoAnterior) {

        Serial.println(modoVoltmetro ? "Cambiado a: Modo Voltímetro" :
"Cambiado a: Modo Fotómetro");

        modoAnterior = modoVoltmetro;

    }

    if (modoVoltmetro) {

        float promedio = analogRead(analogPin);

        float voltaje = promedio * (3.3 / 4095.0) + 0.14;

        if (voltaje < 0.16) voltaje = 0.0;

        Serial.print("Voltaje: ");

        Serial.print(voltaje, 2);

        Serial.println(" V");

        int valor = voltaje * 10;

        mostrarNumero(valor, true);

    } else {

        int lectura = analogRead(ldrPin);

        float luz = map(lectura, 437, 4000, 0, 99) / 10.0;
```


Universidad de Guadalajara

Piensa y trabaja

```
    if (luz < 0) luz = 0.0;

    if (luz > 9.9) luz = 9.9;

    Serial.print("Luz: ");

    Serial.print(luz, 1);

    Serial.println(" nivel");

    int valor = luz * 10;

    mostrarNumero(valor, true);

}

}

void mostrarNumero(int valor, bool punto) {

    int unidades = valor % 10;

    int decenas = (valor / 10) % 10;

    for (int j = 0; j < 20; j++) {

        mostrar(unidades, digitos[0], punto);

        delay(3);

        apagar();

        mostrar(decenas, digitos[1], false);

        delay(3);

        apagar();

    }

}
```

Universidad de Guadalajara

Piensa y trabaja

```
void mostrar(int numero, int digito, bool puntoDecimal) {  
  
    digitalWrite(digitos[0], LOW);  
  
    digitalWrite(digitos[1], LOW);  
  
    digitalWrite(digito, HIGH);  
  
    for (int i = 0; i < 7; i++) {  
  
        digitalWrite(LEDs[i], numeros[numero][i]);  
  
    }  
  
    digitalWrite(LEDs[7], puntoDecimal ? LOW : HIGH);  
  
}  
  
void apagar() {  
  
    for (int i = 0; i < 8; i++) digitalWrite(LEDs[i], LOW);  
  
}
```