

# Universidad Autónoma de Tamaulipas

## Facultad de Ingeniería Tampico



### ASIGNATURA

**Diseño Electrónico Basado en Sistemas Embebidos**

**8vo.** Semestre – Grupo “G<sub>s</sub>”  
2025 -1

### TRABAJO

**Desarrollo de Prácticas y Proyectos**

### UNIDAD

**1**

**Docente:** Dr. García Ruiz Alejandro H.

| <b>Integrante del Equipo</b>     | <b>Nivel de Participación</b> |
|----------------------------------|-------------------------------|
| Antonio Guzman, Iris Lucero      | 40                            |
| Hernández Lara, Ana Patricia     | 40                            |
| Segoviano Arbona, Frida Fernanda | 20                            |
| <b>Total:</b>                    | 100%                          |

## Índice

|  |          |
|--|----------|
| <b>Índice .....</b>  | <b>1</b> |
| <b>Repositorio(s) de Prácticas.....</b>                    | <b>2</b> |
| <b>P1. Lectura de 4 Sensores (Potenciometros).....</b>     | <b>2</b> |
| <b>Descripción de la practica.....</b>                     | <b>2</b> |
| <b>Introducción.....</b>                                   | <b>2</b> |
| <b>Componentes para el desarrollo de la practica .....</b> | <b>2</b> |
| <b>Desarrollo.....</b>                                     | <b>3</b> |
| <b>Conclusiones.....</b>                                   | <b>4</b> |

## Repositorio(s) de Prácticas

| Practica   | Repositorio   |
|------------|---|
| Practica 1 | <a href="https://github.com/lara5tar/SE_PORTAFOLIO-DE-EVIDENCIAS_2025-_1-_EQ6/tree/main/Unidad_01/Arduino/Practica_1">https://github.com/lara5tar/SE_PORTAFOLIO-DE-EVIDENCIAS_2025-_1-_EQ6/tree/main/Unidad_01/Arduino/Practica_1</a> |

### P1. Lectura de 4 Sensores (Potenciometros)

#### Descripción de la practica

La Práctica 1 utiliza cuatro potenciómetros conectados a un Arduino, pero en nuestro caso a Esp32 para medir valores aleatorios. Estos valores se envían a un programa en Python quien los recibe y verifica su formato (ej: i100-200-300-400F), calculando su funcion objetivo y guardando los resultados en una tabla (archivo CSV). El sistema permite ver cómo los ajustes físicos de las perillas se convierten en datos digitales para su análisis y almacenamiento.

#### Introducción

En esta práctica se explorará la interacción entre dispositivos físicos (potenciometros) y un programa en python. Se generarán datos para enviarlos desde el Esp32 al programa. Allí, estos datos se validarán, procesarán y guardarán automáticamente en un CSV. El objetivo es demostrar cómo acciones simples en el mundo real pueden convertirse en información útil para proyectos de medición, automatización o experimentación, destacando la conexión entre hardware y software para sistemas embebidos.

#### Componentes para el desarrollo de la practica

- **Componente 1 (ESP32).** El ESP32 es una placa de desarrollo electrónico programable que actúa como el "cerebro" del sistema. Cuenta con entradas analógicas y digitales, conectividad Wi-Fi/Bluetooth y capacidad para interactuar con sensores y actuadores. En esta práctica, se encarga de leer los valores de los potenciómetros, convertirlos en señales digitales y transmitirlos a la computadora mediante comunicación serial a través del cable USB.
- **Componente 2 (Cables macho a macho).** Son cables conductores con conectores metálicos en ambos extremos (pines macho). Se usan para unir eléctricamente los componentes en la protoboard (como el ESP32 y los potenciómetros), transmitiendo señales eléctricas o alimentación. Su

flexibilidad y variedad de colores ayudan a organizar visualmente el circuito y evitar cortocircuitos.

- **Componente** 3 **(Protoboard).**  
La protoboard, también conocida como breadboard, es una placa rectangular con orificios interconectados eléctricamente en su interior. Permite montar circuitos temporales sin soldar componentes, facilitando la conexión de potenciómetros, cables y el ESP32. Se organizan en filas y columnas para crear rutas de conexión, simplificando el ensamblaje y las pruebas del sistema.
- **Componentes** 4 **(Potenciometros).**  
Los potenciómetros son resistencias variables ajustables manualmente mediante una perilla giratoria. Al girarlos, cambian su resistencia eléctrica, lo que el ESP32 interpreta como un valor numérico (ej: 0-1023). En esta práctica, funcionan como entradas físicas para que el usuario controle los datos que se enviarán a la computadora, simulando sensores o ajustes manuales en sistemas reales.

## Desarrollo

Se conectaron los potenciómetros a la protoboard usando cables macho a macho. Cada potenciómetro se vinculó al ESP32 mediante sus pines analógicos (A0, A1, A2, A3), asegurando que las terminales de alimentación (+5V y GND) estuvieran correctamente enlazadas a la placa. La protoboard permitió organizar las conexiones de forma ordenada y reversible, evitando errores como cortocircuitos o falsos contactos.

Se cargó en el ESP32 un código de arduino para leer los valores analógicos de los potenciómetros. El programa convierte estas lecturas en una cadena de texto estructurada (ej: i512-1023-2048-3000F), que se envía por comunicación serial la computadora cada cierto intervalo de tiempo.

En un script en Python se recibe la cadena enviada por el ESP32, verifica su formato (eliminando datos corruptos o incompletos), extrae los valores numéricos y calcula la función objetivo (suma de los valores). Finalmente, los resultados se guardan en un archivo CSV, generando un registro histórico de las mediciones para su posterior análisis.



```

File Edit Selection View Go Run Terminal Help
SE_PORTAFOLIO-DE-EVIDENCIAS_2025-1_E06
Unidad_01 > Arduino > Practica_1 > ejercicio1.py
1 from funciones_auxiliares import funcion_objetivo
2
3 import serial
4 import csv
5 import sys
6
7 # Configurar la conexión serial
8 try:
9     ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=1) # Added timeout
10 except serial.SerialException as e:
11     print(f'Error al conectar al puerto serial: {e}')
12     sys.exit(1)
13
14 if __name__ == '__main__':
15     lista_resultados = []
16
17     try:
18         while True:
19             line = ser.readline().decode('utf-8').strip()
20             if not line:
21                 continue
22
23             if line.startswith('I') and line.endswith('F'):
24                 valores_str = line[1:-1].split(',')
25                 if len(valores_str) != 4:
26                     print("Error: Vector incompleto")
27                     continue
28
29                 vector = [int(valor) for valor in valores_str]
30
31                 # Calcular la función objetivo
32                 resultado = funcion_objetivo(vector)
33
34                 # Agregar el vector y su resultado a la lista
35                 lista_resultados.append((vector, resultado))
36
37                 print(f"Vector: {vector}, Función objetivo: {resultado}")
38
39             except (ValueError, UnicodeDecodeError) as e:
40                 print(f'Error en la lectura de datos: {e}')
41             except serial.SerialException as e:
42                 print(f'Error en la comunicación serial: {e}')
43                 break
44
45     except KeyboardInterrupt:
46         print("\nPrograma interrumpido por el usuario.")
47
48     finally:
49         # Guardar los resultados en un archivo CSV
50         with open('resultados.csv', 'w') as f:
51             writer = csv.writer(f)
52             writer.writerow(['vector', 'funcion_objetivo'])
53             for (vector, resultado) in lista_resultados:
54                 writer.writerow([vector, resultado])
55
56         print("Proceso finalizado. Resultados guardados en resultados.csv")
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

## Conclusiones

Esta práctica ejemplifica el funcionamiento de sistemas embebidos, donde dispositivos como el ESP32 actúan como núcleos de procesamiento dedicados

para interactuar con el entorno físico (mediante sensores como los potenciómetros) y comunicar datos a sistemas externos (como la computadora). Al integrar hardware, firmware y software, se demuestra cómo los sistemas embebidos permiten convertir acciones manuales en información digital procesable, siendo esenciales en aplicaciones como automatización, IoT o monitorización, donde se requiere eficiencia, respuesta en tiempo real y conexión entre el mundo físico y el digital.