



LABORATORIO 3 ADC

PROGRAMA: INGENIERIA MECATRONICA

ASIGNATURA: MICROCONTROLADORES

💡 Guía de Laboratorio: ADC en ESP32-S3 con MicroPython

⌚ Objetivos

- Comprender el funcionamiento del ADC.
- Configurar y usar ADC en ESP32-S3.
- Implementar ejemplos prácticos con potenciómetro y LM35.
- Analizar ruido y aplicar filtrado digital.
- Resolver retos de dificultad creciente.

🔍 1. Introducción Teórica

El ADC convierte señales analógicas en digitales mediante:

1. Muestreo
2. Cuantificación
3. Codificación

Resolución: 12 bits (0-4095)

Paso mínimo: $V_{ref}/4095 \approx 0.805 \text{ mV}$

💡 3. Ejemplo 1 - Lectura de potenciómetro

Círculo: Potenciómetro conectado a 3.3V y GND, salida al GPIO4.

```
from machine import Pin, ADC
import time
```



```
adc = ADC(Pin(4))
adc.atten(ADC.ATTN_11DB)
adc.width(ADC.WIDTH_12BIT)

VREF = 3.3

while True:
    raw = adc.read()
    voltaje = (raw / 4095) * VREF
    print("RAW:", raw, " Voltaje:", round(voltaje,3), "V")
    time.sleep(0.5)
```

💡 4. Ejemplo 2 - Lectura de sensor LM35 (temperatura)

Circuito: LM35 conectado con Vout → GPIO4, Vs → 3.3V, GND → GND.

```
from machine import Pin, ADC
import time

adc = ADC(Pin(4))
adc.atten(ADC.ATTN_11DB)
adc.width(ADC.WIDTH_12BIT)

VREF = 3.3

def leer_temp():
    raw = adc.read()
    volt = (raw / 4095) * VREF
    temp_c = volt / 0.01
    return raw, volt, temp_c

while True:
    raw, volt, temp = leer_temp()
    print("RAW:", raw, " Voltaje:", round(volt,3), "V", " Temp:", round(temp,2),
"°C")
    time.sleep(1)
```

💡 5. Ejemplo 3 - Filtrado de ruido

```
def leer_filtrado(n=20):
    suma = 0
    for i in range(n):
        suma += adc.read()
```

- 📍 Sede Quirinal: Calle 21 No. 6 - 01
- 📍 Sede Prado Alto: Calle 8 No. 32 – 49 PBX: (608) 8754220
- 📍 Sede Pitalito: Carrera 2 No.1 - 27 - PBX: (608) 8360699
- ✉ Email: contacto@corhuila.edu.co - www.corhuila.edu.co

Personería Jurídica Res. Ministerio de Educación No. 21000 de Diciembre 22 de 1989
NIT. 800.107.584-2



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA

"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"



```
return suma // n

while True:
    valor = leer_filtrado()
    voltaje = (valor / 4095) * VREF
    print("Filtrado:", valor, " Voltaje:", round(voltaje,3), "V")
    time.sleep(0.5)
```

6. Reto inicial

Diseñar un sistema que lea temperatura con LM35, encienda un LED en GPIO2 si la temperatura supera 30°C y lo apague si baja de 28°C. Usar Timer con muestreo cada 200 ms.

7. Reto avanzado - Mini Data Logger

Implementar un sistema que muestre un potenciómetro cada 100 ms, almacene los últimos 50 valores y calcule mínimo, máximo y promedio en tiempo real.

```
from machine import Pin, ADC, Timer
import time

adc = ADC(Pin(4))
adc.atten(ADC.ATTN_11DB)
adc.width(ADC.WIDTH_12BIT)

VREF = 3.3
data = []

def muestreo(timer):
    global data
    raw = adc.read()
    data.append(raw)
    if len(data) > 50:
        data.pop(0)

t = Timer(0)
t.init(period=100, mode=Timer.PERIODIC, callback=muestreo)

try:
    while True:
        time.sleep(5)
        if data:
            minimo = min(data)
```



```
maximo = max(data)
promedio = sum(data)/len(data)
print("Mín:", minimo, " Máx:", maximo, " Promedio:",
round(promedio,2))
except KeyboardInterrupt:
    t.deinit()
```

📋 8. Preguntas de reflexión

1. Diferencia entre señal muestreada y continua.
2. Efecto de la resolución en precisión.
3. Razones para filtrar en ESP32-S3.
4. Aplicaciones prácticas del data logger.

☑ 9. Conclusiones

- El ADC convierte señales analógicas en valores digitales.
- El ESP32-S3 ofrece resolución de 12 bits y rangos configurables con atenuación.
- El ruido se mitiga con filtrado en hardware y software.
- Retos prácticos permiten afianzar conceptos de muestreo, histéresis y registro de datos.