



- **Nombres:**

Robert Rosario (2021-1940)

- **Asignatura:**

ELECTIVA

- **Profesor/a:**

Carlos Pichardo

- **Fecha:**

09-10-2025

Investigación: Diseño de un PLC con múltiples entradas/salidas y comunicaciones

1. Filtro antialiasing / teoría del muestreo

1.1 Teorema de Nyquist y aliasing



- El **teorema de muestreo de Nyquist–Shannon** dice que para reconstruir correctamente una señal tras muestrearla, la frecuencia de muestreo (f_s) debe ser al menos dos veces la mayor frecuencia presente en la señal de interés (es decir, $(f_s \geq 2 f_{\text{máx}})$)).
- Si hay componentes de frecuencia mayores que $(f_s/2)$, estas se pliegan (“alias”) y aparecen como señales falsas de baja frecuencia.
- En la práctica, muchas señales tienen componentes de ruido o armónicos altos que hay que limitar antes del muestreo.

1.2 Qué es un filtro antialiasing

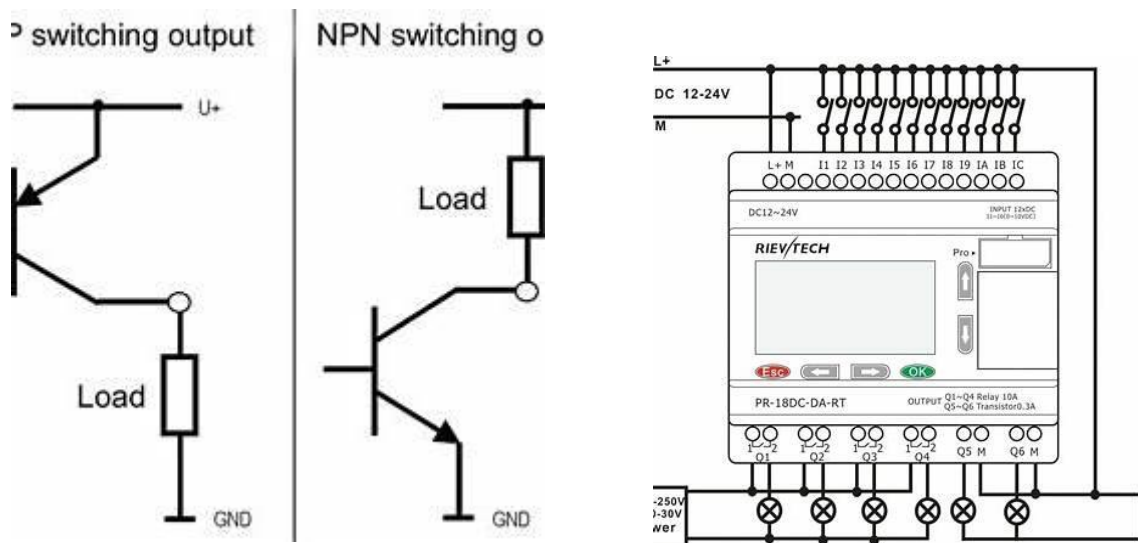
- Un **filtro antialiasing** (AAF, anti-aliasing filter) es un filtro analógico (generalmente pasa-bajos) colocado antes del convertidor analógico-digital (ADC) para atenuar las frecuencias por encima de la banda que queremos muestrear.
- Idealmente, un filtro paso-bajos ideal cortaría exactamente en la frecuencia límite, pero no es realizable en la práctica. Por eso se deja una **banda de transición** donde la atenuación va aumentando gradualmente.
- En muchos diseños la frecuencia de corte del filtro se coloca algo menor que $(f_s/2)$ para dar margen a la atenuación de los componentes indeseados.

1.3 Diseño del filtro y opciones prácticas

- Para señales de baja frecuencia (por ejemplo, 50/60 Hz, señales 0–10 V, 4–20 mA), un filtro **RC de primer orden** puede bastar para eliminar ruido de alta frecuencia.
- Si se necesita más rechazo de frecuencias superiores, se usan filtros activos de segundo orden (Sallen–Key, Butterworth, Bessel) con amplificadores operacionales.
- En sistemas de adquisición de datos de mayor velocidad, se suele **sobre muestrear** (oversampling) y luego aplicar filtrado digital, lo que relaja los requisitos del filtro analógico.
- También hay ICs de filtro activo listos, como el **LTC1562-2** (cuádruple filtro de 2° orden) que permite configurar filtros de Butterworth, Chebyshev, etc.

2. Esquemáticos / tipologías de entradas y salidas para el PLC

A continuación, se presentan las conexiones típicas y consideraciones para distintos tipos de I/O del PLC.



2.1 Entradas PNP

- Los sensores PNP entregan la señal positiva al activarse. En el PLC, puedes usar una **resistencia limitadora** y un optoacoplador para aislar y convertir la señal a niveles lógicos del microcontrolador (3.3 V o 5 V).

Configuración típica:

- Sensor PNP → Rserie → LED del opto → transistor interno del opto → salida al PLC
- Protección: diodo TVS, protección contra polaridad inversa, resistencias adecuadas.

2.2 Salidas PNP

- En este modo la salida del PLC entrega la tensión positiva al actuador cuando debe activarse.
- Usa un transistor PNP de potencia o MOSFET p-channel para controlar la carga. Añade protección: diodo anti-parásito si la carga es inductiva, limitación de corriente, vigilancia de sobrecarga.

2.3 Entradas 0–10 V

- Para medir una señal de voltaje de 0 a 10 V y llevarla al ADC (por ejemplo un rango de 0–3.3 V), se emplea un divisor de tensión + buffer con op-amp para bajar impedancia.
- También es recomendable agregar un filtro RC / activo antialiasing antes del ADC.
- Si se desea aislamiento, se podría usar un convertidor aislado o amplificador aislado.

2.4 Salidas 0–10 V

- Se requiere un DAC (12–16 bits) seguido de un amplificador de salida para generar la señal 0–10 V.
- Protección contra cortocircuitos y buffering para que la salida no se degrade ante cargas.

2.5 Entrada 4–20 mA

- Este es el estándar de señal industrial corriente. Se convierte en voltaje usando una **resistencia shunt** (por ejemplo $250\ \Omega \rightarrow 1\text{--}5\text{ V}$) y luego se procesa con op-amps.
- También hay ICs diseñados para recepción de 4–20 mA con ganancia interna y offset (por ejemplo RCV420 de TI).
- Agregar filtrado antialiasing y protección.

2.6 Salida 4–20 mA

- Puede ser **2-hilos** (alimentado por el lazo) o **3-hilos** (separar alimentación del lazo).
- Se implementa con un DAC + transistor / fuente de corriente controlada que produce la corriente deseada según la señal del DAC.
- Muchos transmisores industriales ya integran esta topología.

2.7 Entradas NPN

- Los sensores NPN conectan la salida a tierra (GND) al activarse.
- En el PLC se coloca una resistencia pull-up hacia el voltaje, luego un optoacoplador o lógica para traducir a 3.3/5 V.
- Protección similar: TVS, diodos, filtros.

2.8 Salidas NPN

- La salida del PLC “hunde” la carga a tierra cuando activa.
- Se puede usar un transistor NPN o MOSFET canal N para conmutar la carga. Añade diodos de flyback si la carga es inductiva.

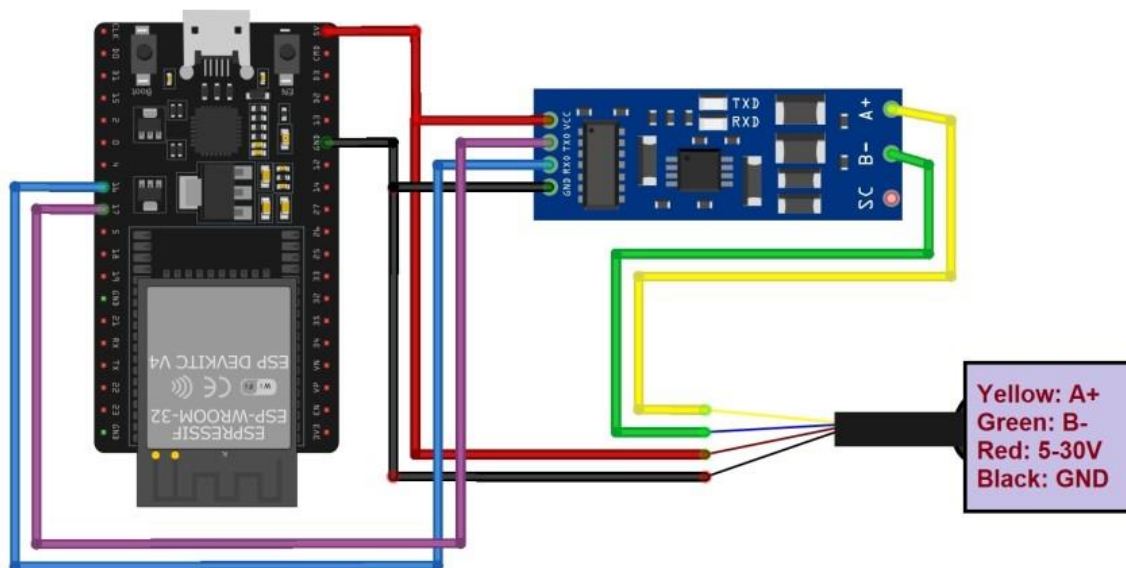
2.9 Driver de relé

- Un relé puede ser controlado desde el PLC mediante un transistor/MOSFET:
 - PLC → resistencia → base/gate del transistor → relé → alimentación
 - Se debe incluir un **diodo flyback** antiparalelo en la bobina del relé para proteger el transistor.
 - Para múltiples relés, se pueden usar drivers tipo ULN2003 / ULN2803.

2.10 Entrada digital con optoacoplador

- Señal externa → resistencia serie → LED del opto → transistor interno → salida al PLC.
- El aislamiento óptico ayuda a proteger al microcontrolador de ruidos e interferencias.

3. Interfaz RS-485 y protocolo Modbus con ESP32



3.1 RS-485 físico

- RS-485 es un estándar de comunicación diferencial balanceada, ideal para distancias largas y ambientes industriales.
- Se necesita un transceptor (por ejemplo, **MAX485**, **LTC2862**) conectado al microcontrolador.
- El bus debe tener terminación ($120\ \Omega$) en ambos extremos y resistencias de polarización (pull-up / pull-down) para definir el estado de reposo.
- El control de dirección (transmisión/recepción) se hace mediante pines DE/RE del transceptor.

3.2 Modbus RTU en ESP32

- El protocolo Modbus RTU es ampliamente usado en sistemas SCADA y PLCs industriales.
- En el ESP32, se puede implementar usando la librería **esp-modbus** (disponible en ESP-IDF) o adaptaciones de ModbusMaster para Arduino.
- Se configura el puerto UART con parámetros (baud, paridad, bits), se gestionan los pines DE/RE para alternar entre envío y recepción.
- Se define qué direcciones de esclavo atenderá y qué registros ofrecerá (entradas digitales, analógicas, salidas).
- Artículo de ejemplo: “How to use Modbus RTU with ESP32 to read Sensor Data”.
- Otro recurso: comunicación entre PLC (M-Duino) y ESP32 usando Modbus RTU.

4. Medición de corriente / energía: CT y sensores AC

4.1 Transformador de corriente (CT)

- Un CT mide corriente alterna sin conectar directamente al circuito de potencia.
- Su secundario debe tener carga (no debe abrirse) para evitar voltajes peligrosos.
- Se convierte la salida del CT (corriente) en una tensión con una resistencia shunt o carga conocida.

4.2 ICs para medición de energía / potencia

- Para facilitar el diseño, se usan ICs que integran cálculo de potencia, RMS, etc.
- **MCP39F511 (Microchip)**: monitor de potencia monofásica con interfaz I²C.
- **ADE7953 / ADE7880 / ADE7858 (Analog Devices)**: ICs de alta precisión para medición eléctrica, con interfaz SPI o registros digitales.
- En proyectos DIY, hay ejemplos con ESP32 + CT usando librerías como **Emonlib**.

4.3 Consideraciones adicionales

- Añadir filtro antialiasing en la señal de tensión y corriente antes del ADC / IC de medición.
- Aislamiento galvánico si es necesario.
- Calibración de offset y ganancia del sistema para obtener precisión.

5. Conclusión

En este documento hemos cubierto los conceptos teóricos del muestreo y el filtro antialiasing, y detallado las topologías de entrada y salida que necesitas para tu PLC (PNP/NPN, 0–10 V, 4–20 mA, relé, optoacoplador). También hemos visto cómo integrar la comunicación RS-485 con Modbus en un ESP32, y cuáles ICs usar para medición de energía con CTs.

Este material sirve como base de investigación para el diseño del hardware del PLC. En siguientes pasos puedes traducir estos esquemas a un diseño real (KiCad u otra herramienta) y probar prototipos.