

# Curso de Introducción a los PLCs

# Objetivo

- ▶ El objetivo del curso es que los participantes adquieran la capacidad diagnosticar y detectar fallas en sistemas controlados por PLC's.
- ▶ El desarrollo del curso proveerá a los participantes las técnicas básicas de programación "LADDER".
- ▶ Los participantes aprenderán a programar, documentar, editar y establecer una comunicación "on line" con un PLC.
- ▶ Conocer y aplicar instrucciones básicas del PLC a través de la solución de pequeños problemas lógicos.

# Contenido

- ▶ **Introducción**
- ▶ **Relés vs PLCs**
- ▶ **Estructura interna**
- ▶ **Esquema de Control**
- ▶ **Criterios de Selección de equipos**
- ▶ **Descripción de Componentes Principales**
- ▶ **Tipos de estructura**
- ▶ **MicroLogix**
- ▶ **SLC 500**
- ▶ **Tipos de Entradas y Salidas**
- ▶ **Módulos Especiales**
- ▶ **Comunicaciones**
- ▶ **Estructura RAM**
- ▶ **Archivos de Procesador**
- ▶ **Direccionamiento**
- ▶ **Concepto de Ladder**
- ▶ **Conceptos de Lógica**
- ▶ **Ciclo de SCAN**
- ▶ **Instrucciones de BIT, temporizadores, contadores, de comparación y matemáticas.**
- ▶ **Redes y Protocolos**
- ▶ **DH 485, DH + y DF1**
- ▶ **Escalados y PIDs**

# Introducción

- ▶ Surgen al mercado en la década del '60  
La causa fundamental fue el gran desarrollo que tuvo la microelectrónica, en especial la aparición de los microprocesadores
- ▶ Al comienzo sólo manejaban señales ON/OFF, su aplicación típica fue sustituir tableros con lógica de relé y dispositivos discretos.

# Introducción

## ❖ ¿Qué es?

–Es un equipo electrónico programable, mediante el cual se pueden realizar ciertas funciones de control.

## ❖ ¿PLC?

▶ La sigla significa **Programable Logic Controller**, (Controlador Lógico Programable).

# Introducción

- ▶ Ventaja: menor tamaño, mayor versatilidad y mayor confianza.
- Actualmente pueden manejar señales más complejas (analógicas), interconexión con otros equipos, trabajos en red, etc..

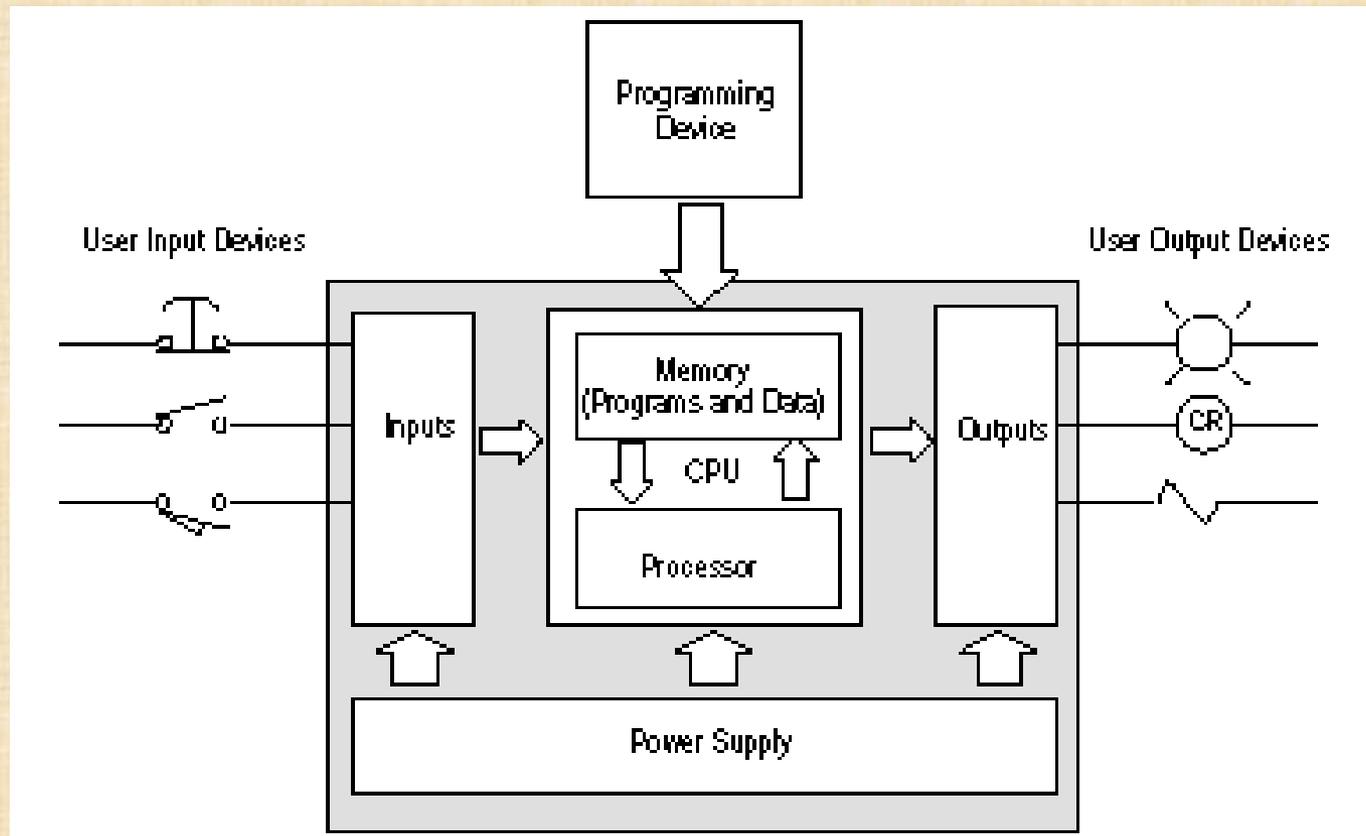
# Relés vs PLCs

	RELES	PLC'S
<b>TAMAÑO</b>	<b>GRANDE</b>	<b>PEQUEÑO</b>
	<i>Se necesitaban muchos dispositivos para realizar una funcion (temporizadores reles, etc.)</i>	<i>Un solo dispositivo es necesario para realizar el control</i>
<b>VERSATILIDAD</b>	<b>NINGUNA</b>	<b>MUCHA</b>
	<i>Para que el equipo realizase alguna otra funcion era necesario recablear todo el sistema</i>	<i>El control es un programa por lo que solo se necesita reprogramar el equipo para que realice otra tarea, puede realizar mas de una tarea a la vez.</i>
<b>ERRORES</b>	<b>MUCHA</b> posibilidad de cometerlos y <b>GRAN</b> dificultad para resolverlos ya que hay muchos dispositivos discretos para ajustar.	<b>MINIMA</b> los errores que suelen ocurrir son a nivel de software lo cual los hace muy facil de arreglar

# Relés vs PLCs

	<b>RELES</b>	<b>PLC'S</b>
<b>START-UP</b>	<b>GRAN</b> <i>complejidad a la hora de realizar ajustes y coordinaciones entre los diferentes dispositivos</i>	<b>FACIL</b> <i>todos se ajusta en el programa.</i>
<b>MANTENIMIENTO</b>	<b>MUCHO</b> <i>Existen elementos mecanicos que tienen desgastes y deben ser sustituidos. El desgaste de estos equipos influye tambien en la performance del sistema</i>	<b>NINGUNO</b> <i>Y si es necesario realizar algun ajuste este se realiza en el programa y no en un elemento mecanico.</i>
<b>COMUNICACIONES</b>	<b>NINGUNA</b> <i>Al no existir elementos inteligentes era imposible que existiese comunicación entre los equipos</i>	<b>SI</b> <i>como se vera mas adelante en el curso es posible comunicar PLC's entre si y con otros dispositivos como ser HMI's, PC's, etc.</i>

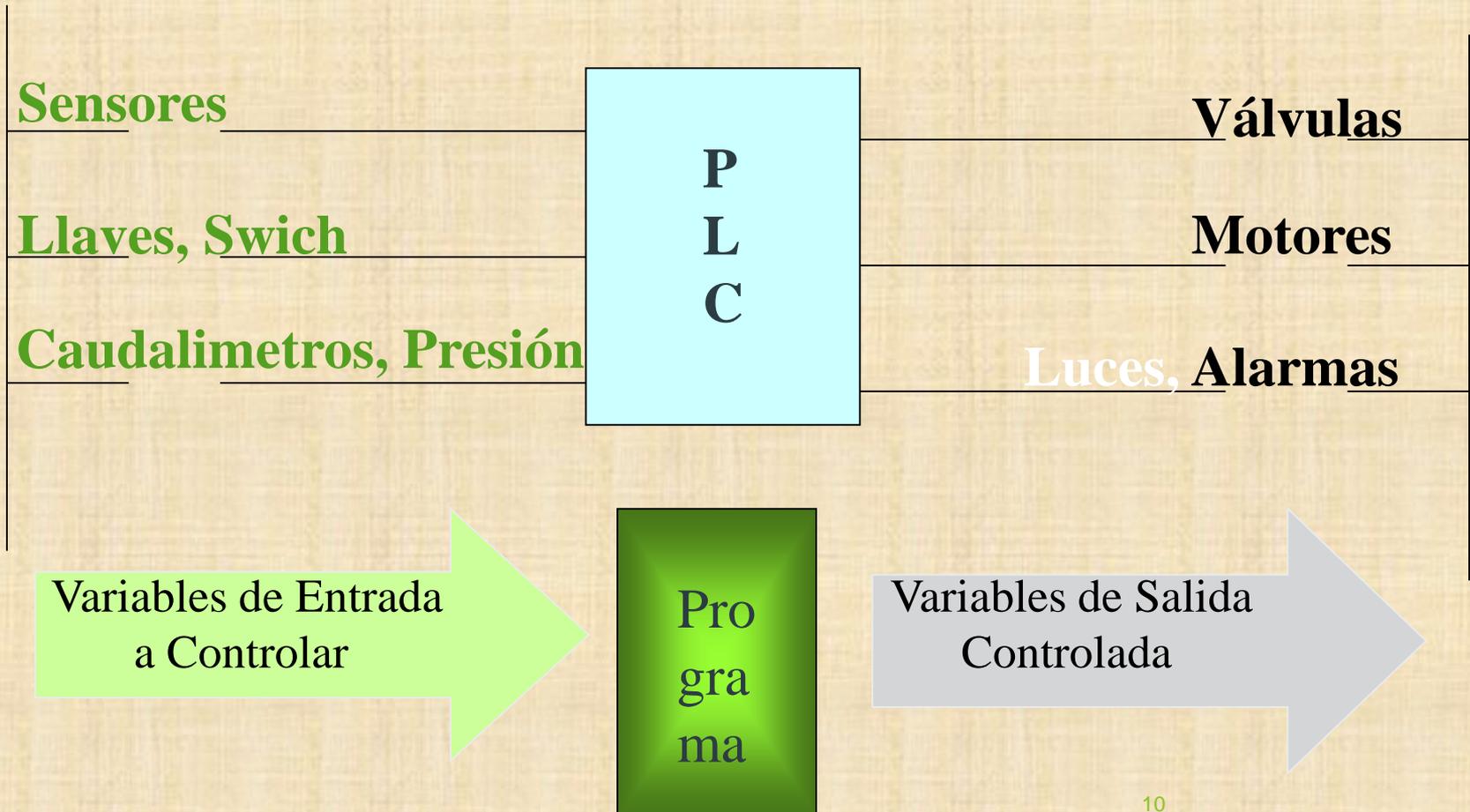
# Estructura interna



# Esquema de Control

## ELEMENTOS ENTRADA

## ELEMENTOS SALIDA



# Criterios de Selección de equipo

## ► Datos Necesarios:

- Tipo y cantidad de entradas
- Tipo y cantidad de salidas
- Capacidad de memoria
- Ciclo de Scan
- Capacidad de trabajo en red
- Características especiales

# Descripción de Componentes Principales

- Fuente de alimentación
- Unidad de procesamiento central (CPU)
- Módulos de interfases de entradas/salidas (E/S)
- Módulo de memorias
- Herramientas de Programación

# Descripción de Componentes Principales

## **Fuente de alimentación**

La función de la fuente de alimentación en un controlador, es suministrar la energía a la CPU y demás módulos según la configuración del PLC.

También puede generar una fuente auxiliar para elementos conectados a las entradas y/o acciones en las salidas.

# Descripción de Componentes Principales

## **Unidad de procesamiento central (C.P.U.)**

Es la parte más compleja e imprescindible del controlador programable, que en otros términos podría considerarse el **cerebro** del controlador.

La unidad central está diseñada a base de microprocesadores.

Su misión es leer los estados de las señales de las entradas, ejecutar el programa de control y gobernar las salidas, el procesamiento es permanente y a gran velocidad.

# Descripción de Componentes Principales

## **Módulos o interfases de entradas y salidas (E/S)**

Son los que proporciona el vínculo entre la CPU del controlador y los dispositivos de campo del sistema.

A través de ellos se origina el intercambio de información ya sea para la adquisición de datos o la del mando para el control de maquinas del proceso.

# Descripción de Componentes Principales

## **Tipos de módulos de entrada y salida**

Por la gran variedad de dispositivos exteriores (sensores y/o actuadores), encontramos diferentes tipos de módulos de entrada y salidas, cada uno de los cuales sirve para manejar cierto tipo de señal (digital o análoga) a determinado valor de tensión o de corriente en DC o AC.

Módulos de entradas digitales

Módulos de salidas digitales

Módulos de entrada analógica

Módulos de salida analógica

# Descripción de Componentes Principales

## Módulos de memoria

Son dispositivos destinados a guardar información de manera temporal o permanente.

Se cuenta con dos tipos de memorias más comunes:

- Volátiles (**RAM**)
- No volátiles (**EPROM y EEPROM**)

# Descripción de Componentes Principales

## **Herramienta de programación**

Los terminales de programación, son el medio para ingresar la lógica que deseamos incorporar al PLC.

Existen dos tipos de programadores: los manuales (Hand Held) tipo calculadora de mano o los Software en PC (los mas usados hoy en día). Una tercer opción en PLCs pequeños, son las pantallas de programación local (en el mismo PLC).

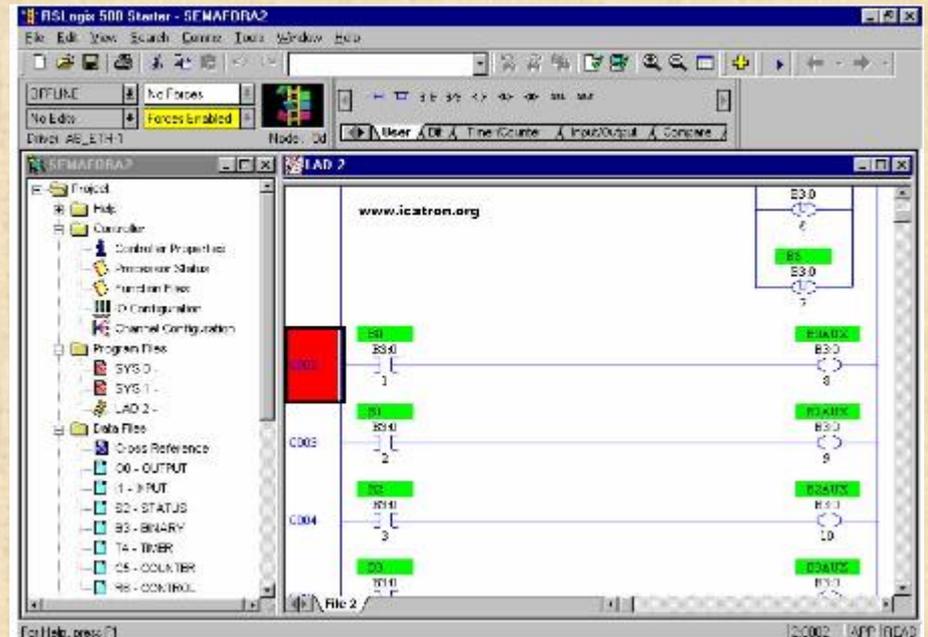
# Descripción de Componentes Principales

## Herramienta de Programación

Hand Held 1747-PT1



RSLogix 500



# Tipos de estructura

## SLC 500



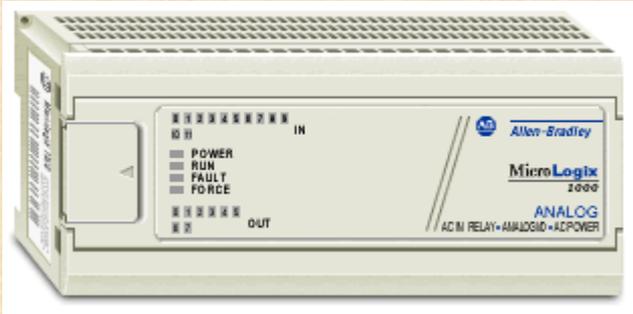
## Micrologix



Tanto los Micrologix como los SLC 500 se programan con el software RSLogix 500

# MicroLogix

## MicroLogix 1000



### Características

- Puerto serie
- No permite programación on-line
- Comunicación DH-485, DeviceNet y EtherNet/IP a través de interfaces 1761-NET-AIC, 1761-NET-DNI y 1761-NET-ENI
- Un contador rápido (sólo controladores con entradas de 24V DC)
- 10, 16, 20 (con 4 entradas y 1 salida analógicas ) o 32 E/S digitales
- EEPROM embebida, no necesita batería y ni módulo de memoria.

## MicroLogix 1100

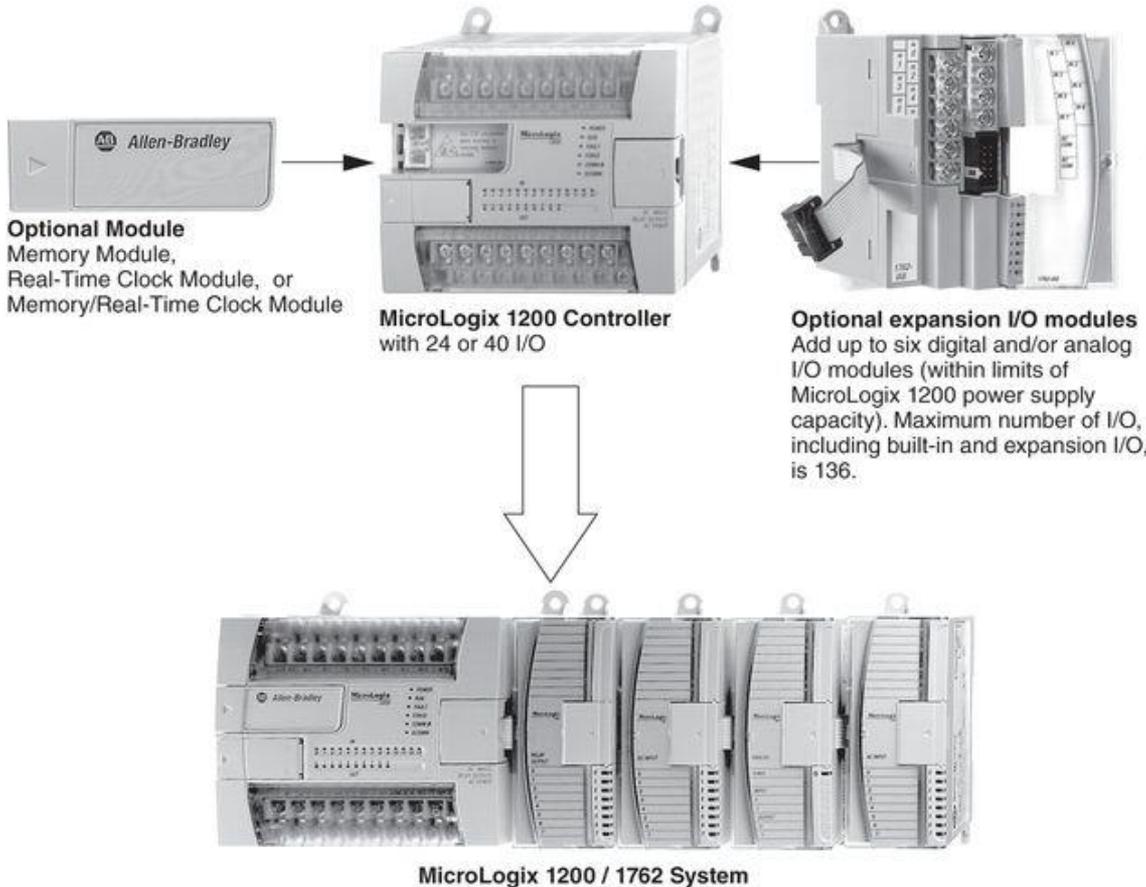


### Características

- Puerto Ethernet/IP 10/100 MBps
- Puerto RS-232/RS-485
- Programación on-line
- Memoria de 8 KB (4 KB programa y 4 KB datos)
- Servidor web embebido para mostrar datos
- Permite ver y modificar datos desde la pantalla LCD.
- Compatible con módulos de expansión (hasta 4 módulos)
- Hasta 80 E/S digitales

# MicroLogix

## MicroLogix 1200



### Características:

- Puerto serie RS-232/RS-485
- No permite programación on-line
- Contador rápido de 20 KHz
- Compatible con módulos de expansión (hasta 6 módulos o 136 E/S digitales)

# MicroLogix

## MicroLogix 1400



### Características

- Puerto Ethernet
  - Programación on-line
  - Servidor web y de correo
  - Permite ver y modificar datos desde la pantalla LCD.
- También permite monitorear E/S.
- Compatible con módulos de expansión (hasta 7 módulos y 256 E/S)
  - Hasta 6 contadores de alta velocidad (sólo en controladores con entradas de DC)
  - Dos puertos serie con soporte DF1, DH-485, Modbus RTU y ASCII.
  - Memoria de 20 KB (10 KB words programa y 10 KB words datos)
  - Hasta 256 E/S digitales
  - Hasta 128 KB para log de datos y 64 KB para recetas.

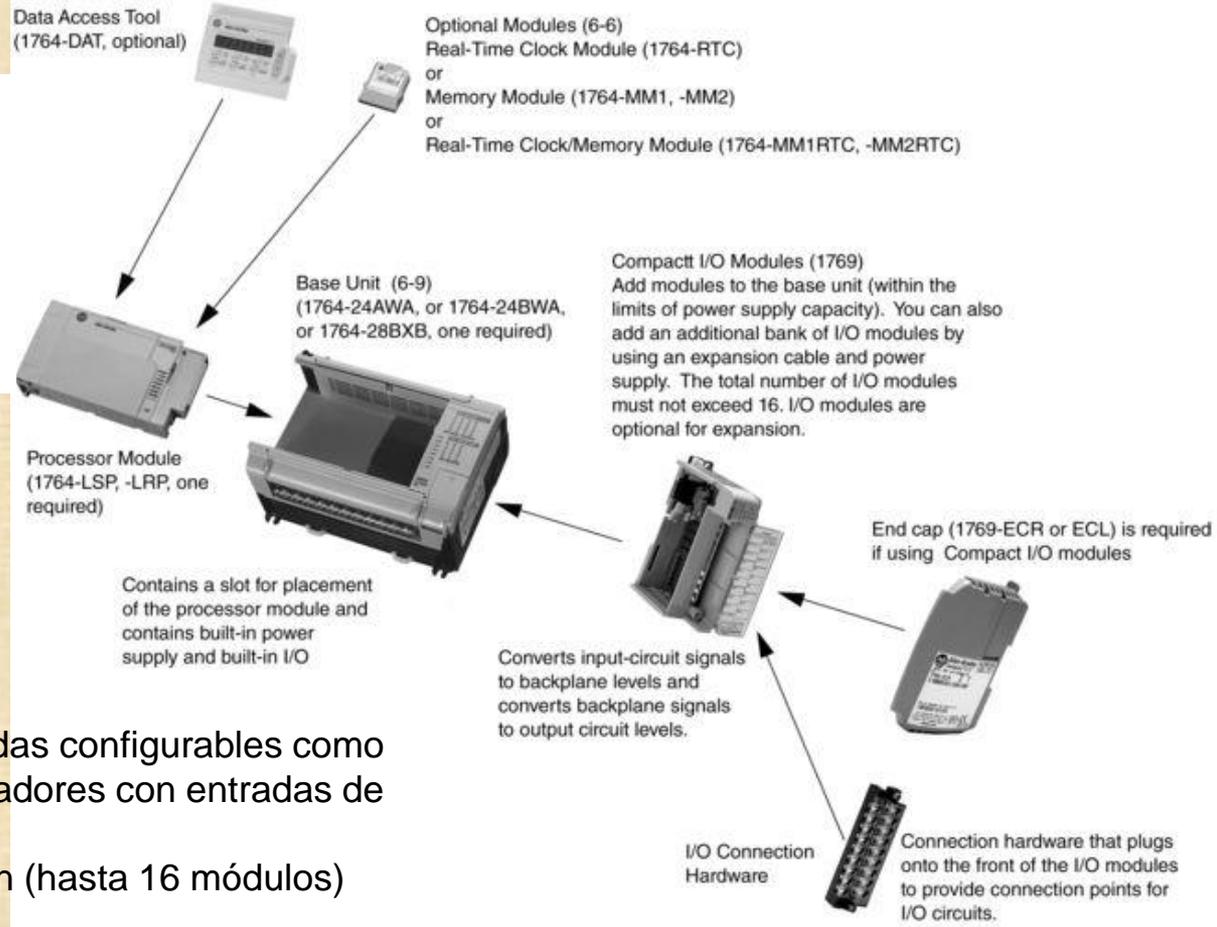
# Micrologix

## MicroLogix 1500



### Características:

- Memoria de 7 KB (3.65 KB programa y 4 KB datos), o 14 KB (10 KB programa, 4 KB datos)
- Módulo opcional para modificación de bits y enteros (1764-DAT)
- 2 contadores rápidos y 2 salidas rápidas configurables como tren de pulsos o PWM. (sólo en controladores con entradas de 24V DC integradas )
- Compatible con módulos de expansión (hasta 16 módulos)



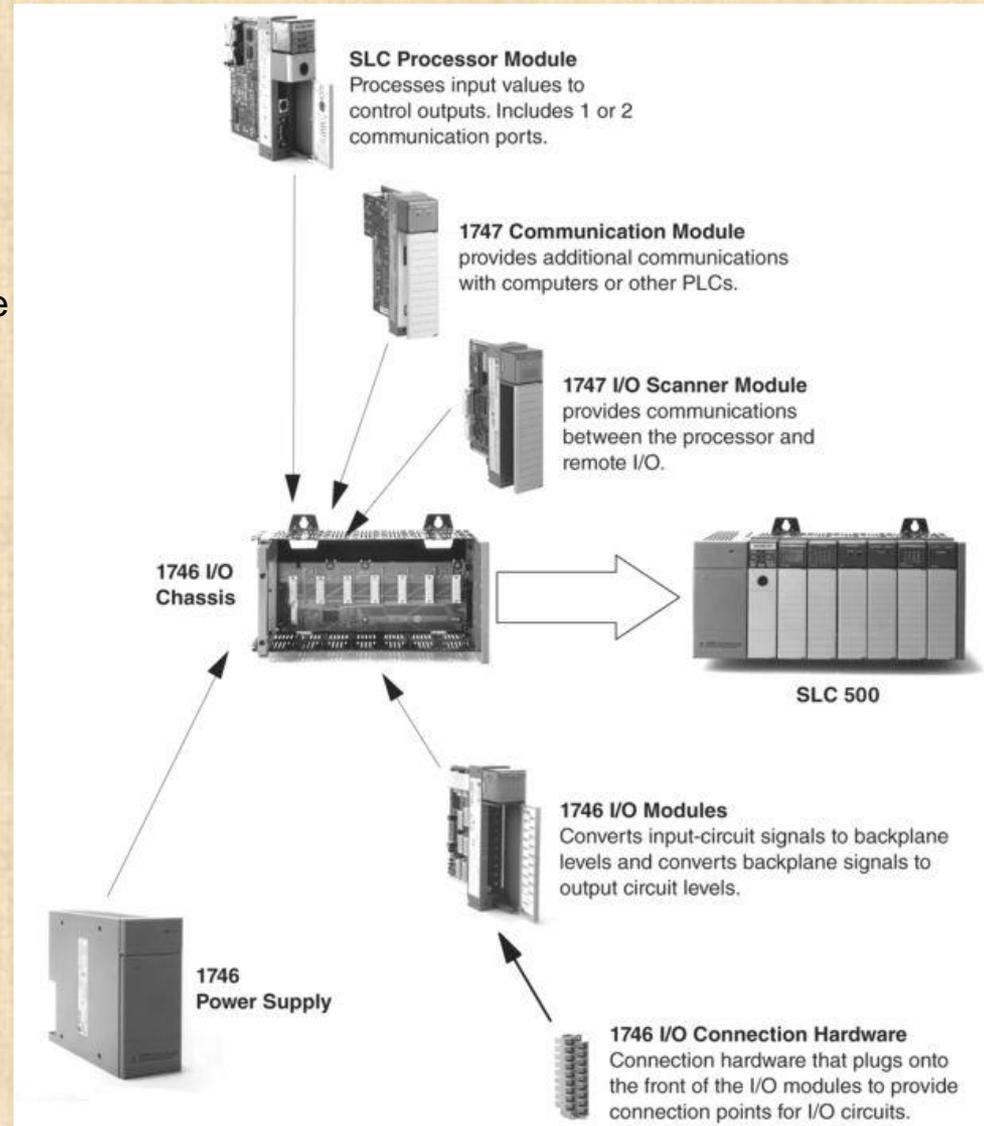
# SLC 500

## Características:

- Hasta 64 KW de memoria configurable para datos/programa.
- Control de hasta 4096 entradas y 4096 salidas
- Hasta 3 chasis locales con hasta 30 módulos de E/S o comunicación
- Soporte de módulos ControlNet, DeviceNet, Universal Remote I/O
- Según el CPU puede tener puertos Ethernet/IP, DH+, DH-485 o RS-232 integrados



Chasis de 4, 7, 10 y 13 slots



# Procesadores SLC

Processor Category	I/O, Max.	I/O Communication	User Memory, Max.	General Communication Ports
SLC 5/01	960	<ul style="list-style-type: none"> <li>•Processor-resident local I/O</li> </ul>	<ul style="list-style-type: none"> <li>•1K or 4K instructions</li> </ul>	<ul style="list-style-type: none"> <li>•DH-485 (cannot initiate messages)</li> </ul>
SLC 5/02	4096 in + 4096 out	<ul style="list-style-type: none"> <li>•Processor-resident local I/O</li> <li>•I/O at remote locations*</li> </ul>	<ul style="list-style-type: none"> <li>•4K instructions</li> </ul>	<ul style="list-style-type: none"> <li>•DH-485</li> </ul>
SLC 5/03	4096 in + 4096 out	<ul style="list-style-type: none"> <li>•Processor-resident local I/O</li> <li>•I/O at remote locations*</li> </ul>	<ul style="list-style-type: none"> <li>•8K, 16K, or 32K words total</li> </ul>	<ul style="list-style-type: none"> <li>•DH-485</li> <li>•RS-232*</li> </ul>
SLC 5/04	4096 in + 4096 out	<ul style="list-style-type: none"> <li>•Processor-resident local I/O</li> <li>•I/O at remote locations*</li> </ul>	<ul style="list-style-type: none"> <li>•16K, 32K, or 64K words total</li> </ul>	<ul style="list-style-type: none"> <li>•DH+</li> <li>•RS-232*</li> </ul>
SLC 5/05	4096 in + 4096 out	<ul style="list-style-type: none"> <li>•Processor-resident local I/O</li> <li>•I/O at remote locations*</li> </ul>	<ul style="list-style-type: none"> <li>•16K, 32K, or 64K words total</li> </ul>	<ul style="list-style-type: none"> <li>•Ethernet</li> <li>•RS-232*</li> </ul>

# Tipos de Entradas y Salidas

Entradas Digitales:

24 VDC Sink (PNP)

24 VDC Source (NPN)

110 VAC

Salidas Digitales:

24 VDC Sink

24 VDC Source

Triac VAC

Rele

# Tipos de Entradas y Salidas

Entradas Analógicas: 0 - 4 a 20 mA

0 - 1 a 10 V

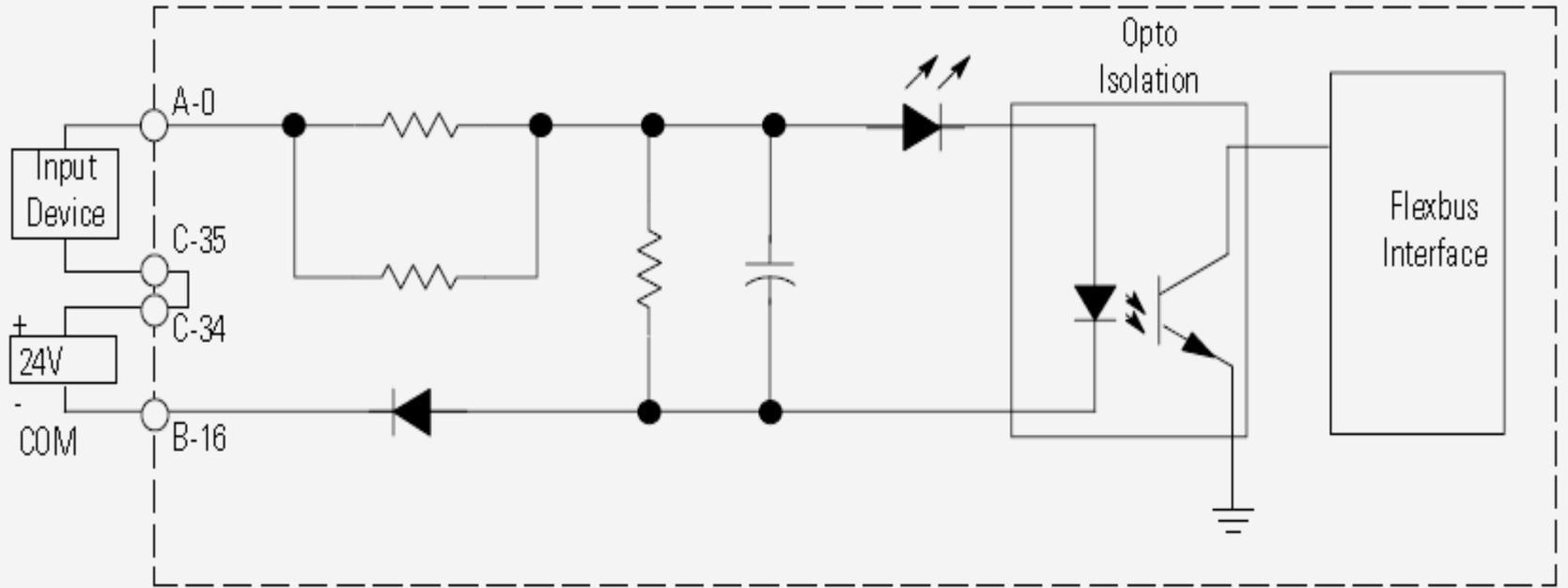
Salidas Analógicas: 0 - 4 a 20 mA

0 - 1 a 10 V

# Tipos de Entradas

## Entradas Digitales 24 VDC Sink

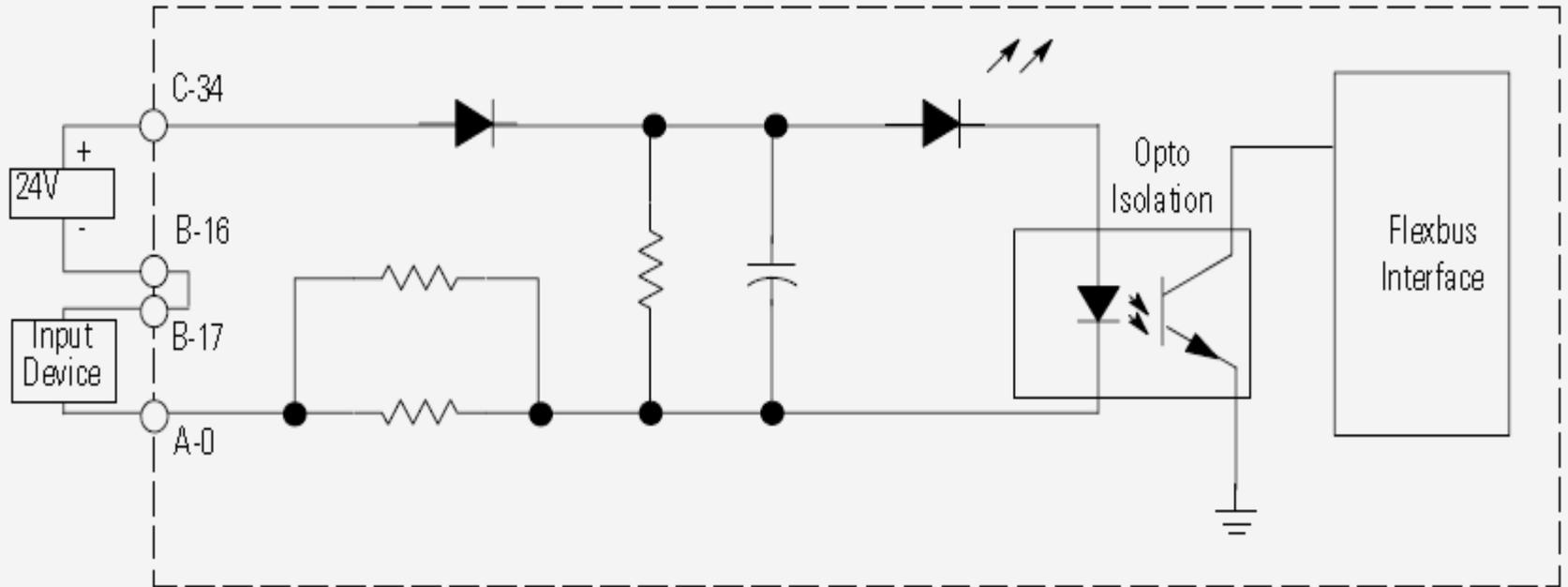
Typical Simplified Schematic for Input 0  
TB3 or TB3S shown



# Tipos de Entradas

## Entradas Digitales 24 VDC Source

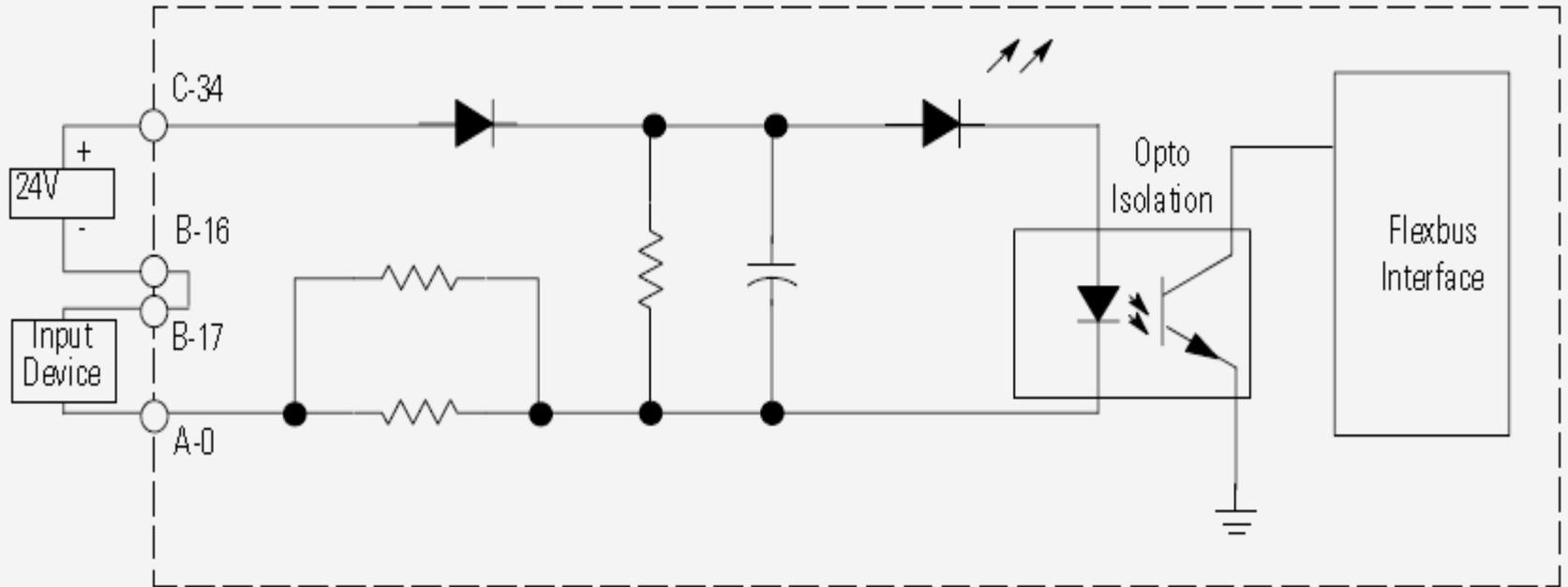
Typical Simplified Schematic for Input 0  
TB3, TB3S, or TB2 shown



# Tipos de Entradas

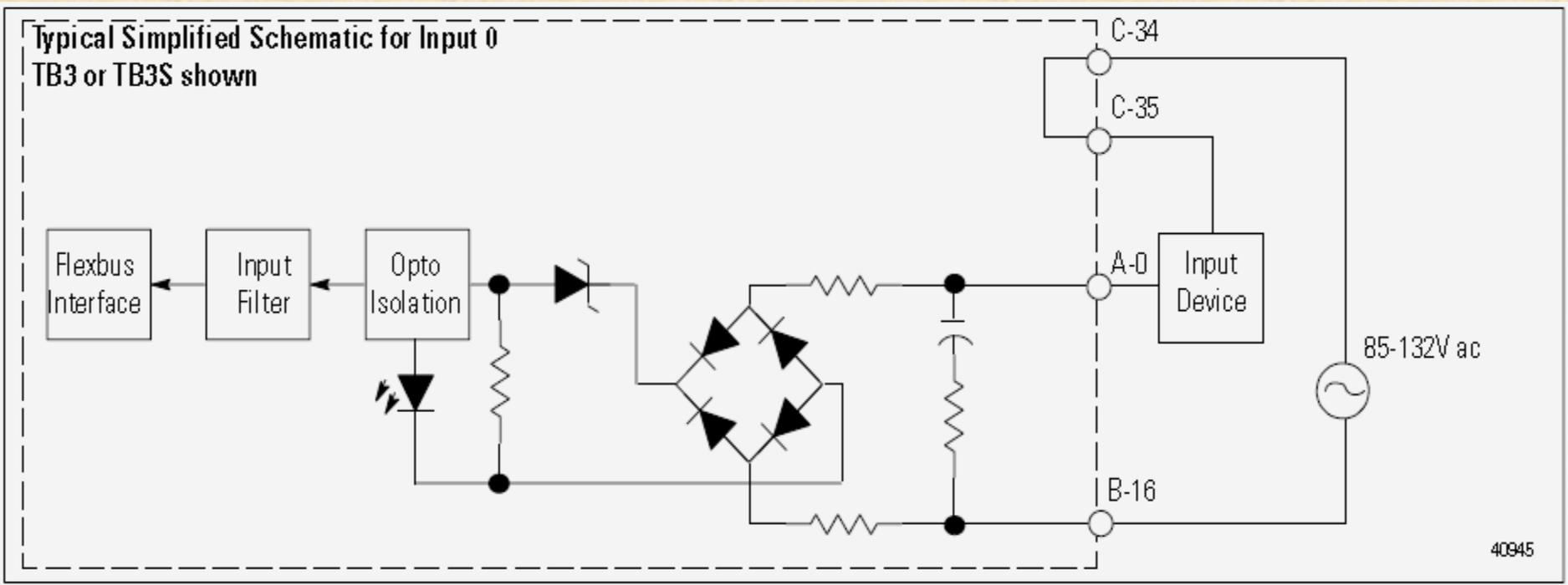
## Entradas Digitales 24 VDC Source

Typical Simplified Schematic for Input 0  
TB3, TB3S, or TB2 shown



# Tipos de Entradas

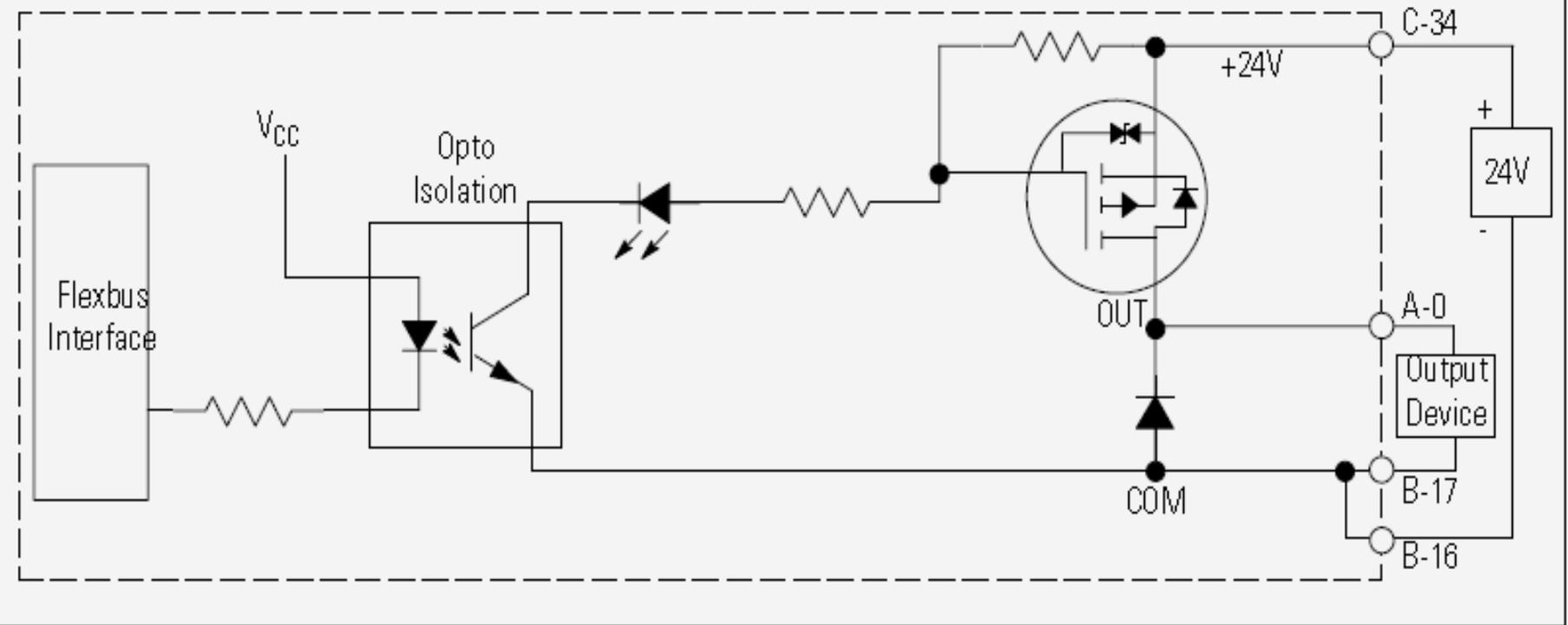
## Entradas Digitales 110 VAC



# Tipos de Salidas

## Salidas Digitales 24 VDC Source

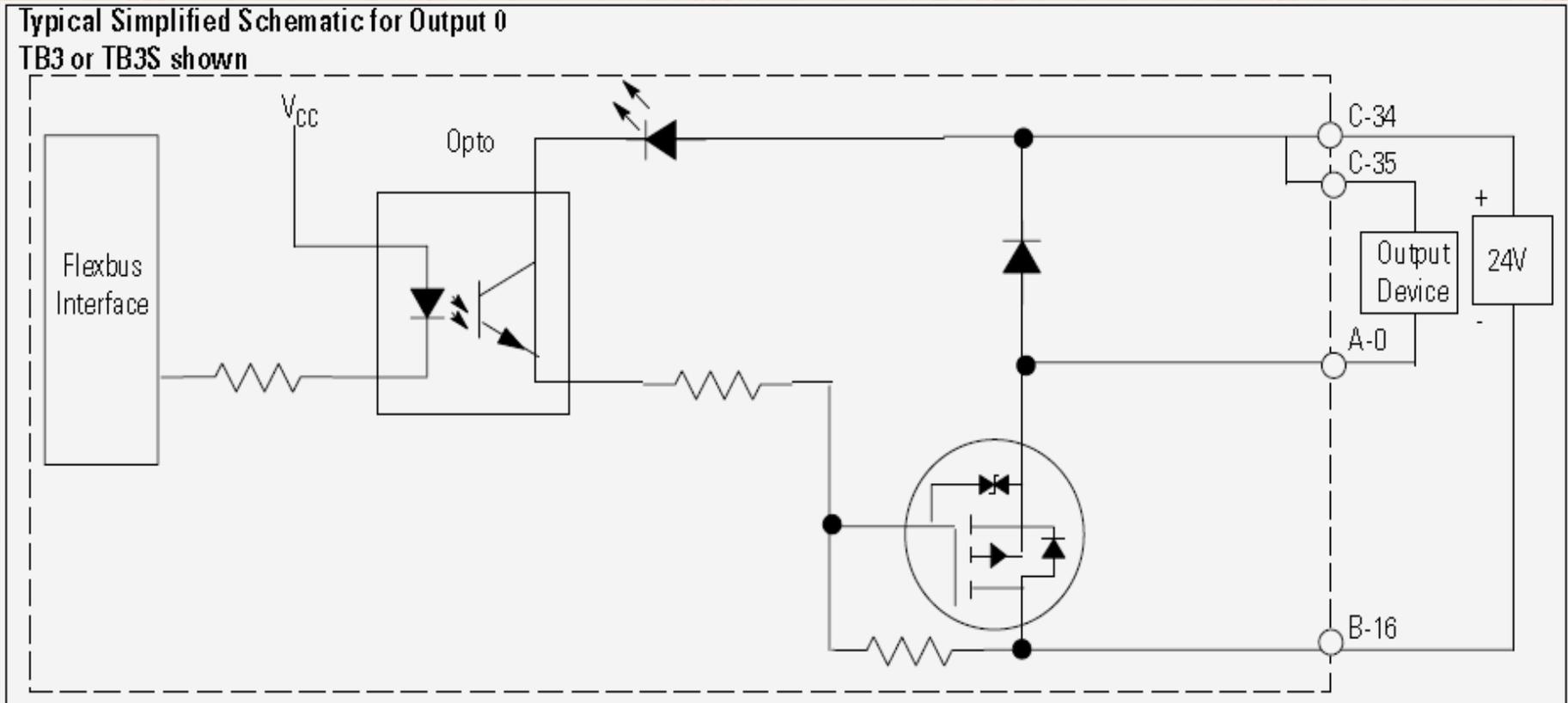
Typical Simplified Schematic for Output 0  
TB3, TB3S, or TB2 shown



# Tipos de Salidas

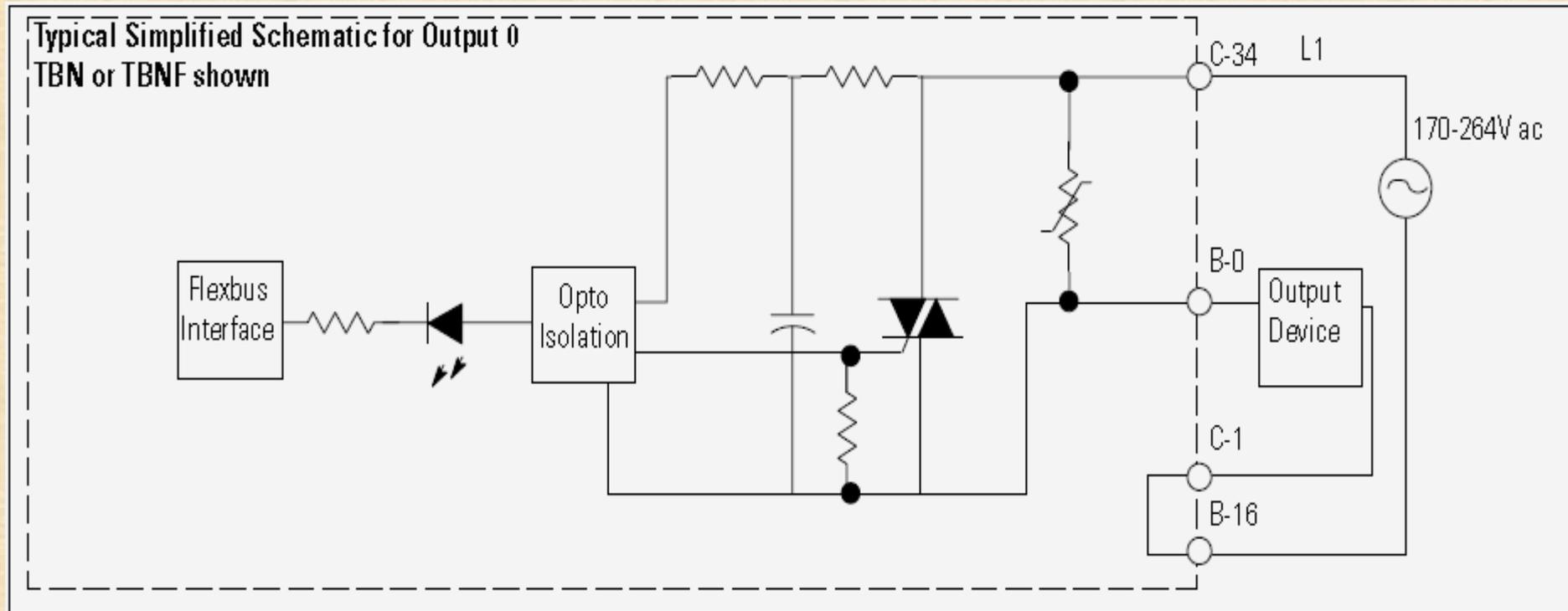
## Salidas Digitales 24 VDC Sink

Typical Simplified Schematic for Output 0  
TB3 or TB3S shown



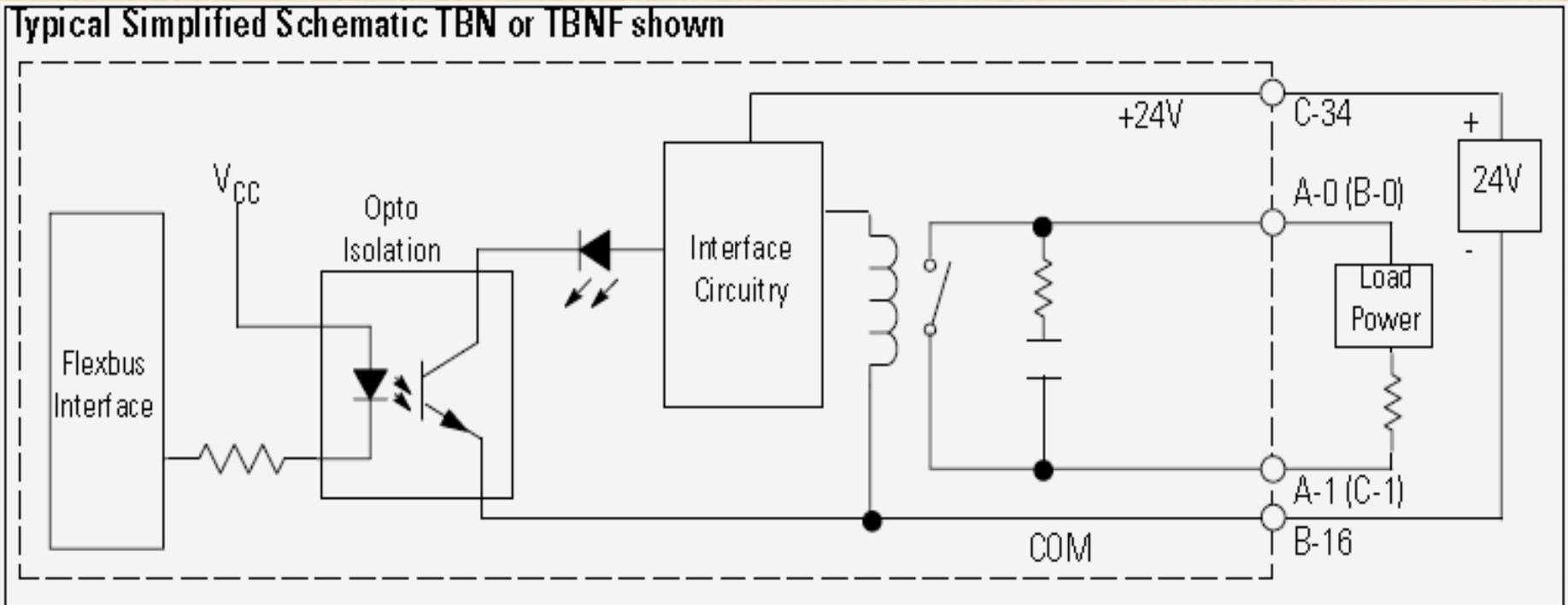
# Tipos de Salidas

## Salidas Digitales a Triac



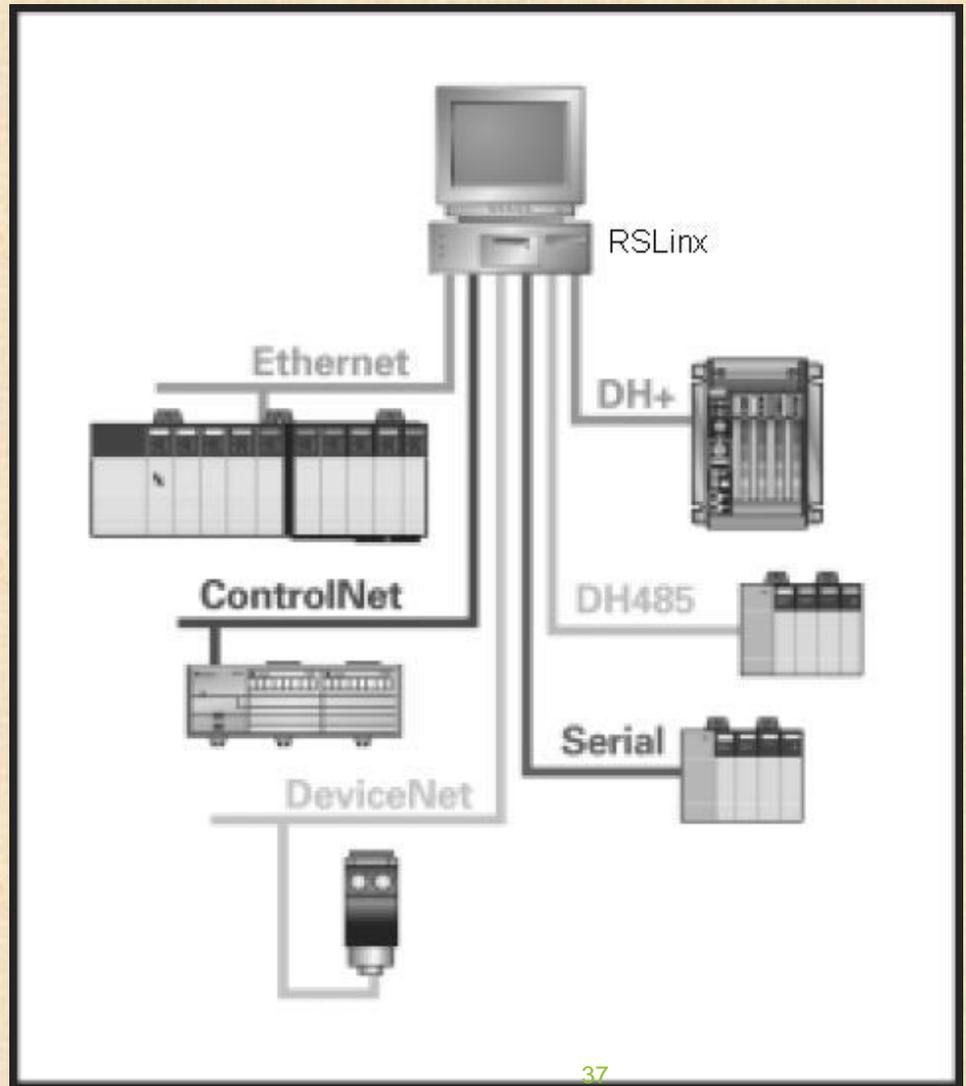
# Tipos de Salidas

## Salidas Digitales a Relé



# RSLinx

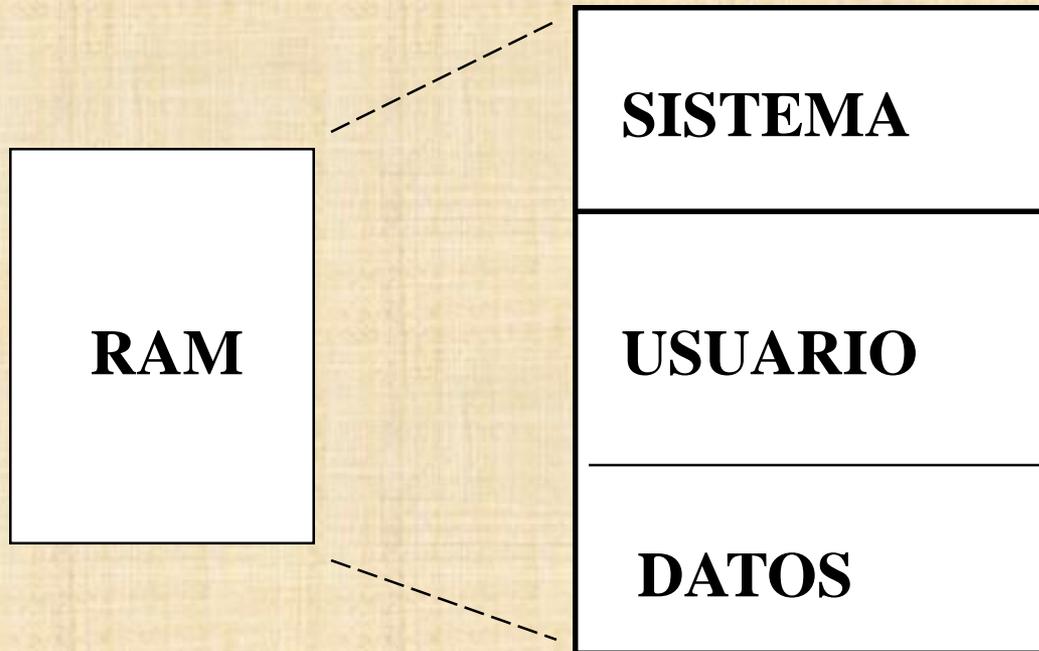
RSLinx es el software que establece la comunicación entre un PLC y la herramienta de programación (RSLogix 5, RSLogix 500, ...) así como con los sistemas SCADA (RSView32).



# RSLinx

- ▶ Configuración del driver de comunicación:
- ▶ Communication-> Configure Drivers...

# Estructura RAM



# Archivos de Procesador

## Archivos de Programa

Son aquellos archivos en los que se almacena la información correspondiente a la configuración del sistema, el programa principal y las subrutinas.

## Archivos de datos

Son aquellos archivos en los que se almacenan los datos que maneja el programa, Enteros, Bits, Temporizadores, etc..

# Archivos de Datos

- Archivo 0 : Imagen de las Salidas (O)
- Archivo 1 : Imagen de las Entradas (I)
- Archivo 2 : Archivo de Estado (S)
- Archivo 3 : Archivo de Bits (B)
- Archivo 4 : Archivo de Temporizadores (T)
- Archivo 5 : Archivo de Contadores (C)
- Archivo 6 : Archivo de Control (R)
- Archivo 7 : Archivo de Enteros (N)
- Archivo 8 : HF, 5/01, 5/02 Reservado;  
5/03, 5/04, 5/05 Punto Flotante (F)
- Archivo 9 a 255 : Configurables por el usuario  
(B, T, C, N, R, F,ST,A)

# Archivos de Programa

## **SLC500**

File 0 : Archivo de Sistema

File 1 : Reservado

File 2 : Programa Principal

File 3-255 : Subrutinas del Programa Principal

mLogix File 0 : Archivo de Sistema

File 1 : Reservado

File 2 : Programa Principal

File 3 : Rutina de corrección de errores

File 4 : Rutina de atención al contador de alta velocidad

File 5 : Rutina de atención a una STI

File 6-15 : Subrutinas del Programa Principal.

# Direccionamiento E/S

<b>O:e.s/b</b>	<b>:</b> delimitador <b>e</b> numero de slot
<b>I:e.s/b</b>	<b>.</b> delimitador de palabra <b>s</b> numero de palabra
	<b>/</b> delimitador de bit <b>b</b> Numero de terminal

# Direccionamiento en general

<b>Ff:s/b</b>	<b>F</b> identificador de archivo <b>f</b> numero de archivo
	<b>:</b> delimitador de elemento <b>s</b> numero de palabra
	<b>/</b> delimitador de bit <b>b</b> Numero de bit

# Ejemplos

Entrada 4 del slot 2

0				<b>4</b>											15
---	--	--	--	----------	--	--	--	--	--	--	--	--	--	--	----

I:2.0/4

I:2/4

Entrada 19 del slot 5

15															0
----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	---

palabra 0

31												<b>19</b>			16
----	--	--	--	--	--	--	--	--	--	--	--	-----------	--	--	----

palabra 1

I:5/19

I:5.1/3

I:5.0/19

# Creación de Archivos de Procesador

- ▶ Tipo de procesador

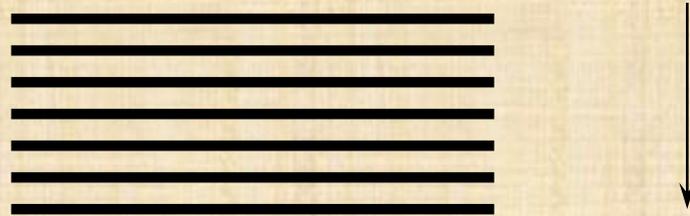
**SLC 500 o Micrologix**

- ▶ Tipo y cantidad de módulos

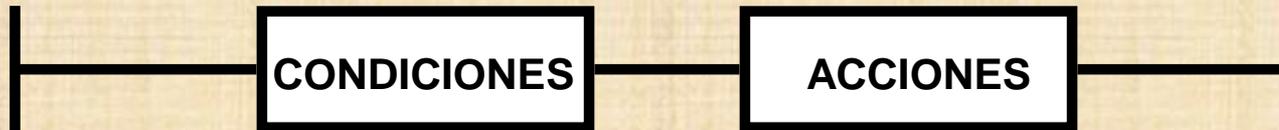
**Entradas digitales, Salidas analógicas, scanner, etc.**

# Concepto de Ladder (programa escalera)

Su nombre viene dado por su construcción



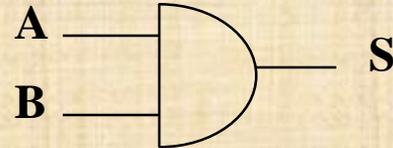
Cada línea esta dividida en dos partes → **CONDICIONES**  
→ **ACCIONES**



# Conceptos de Lógica

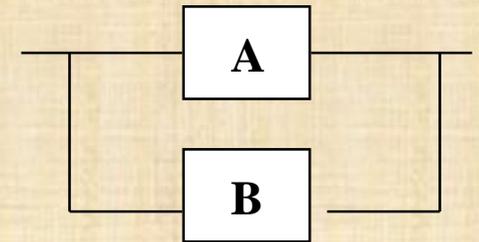
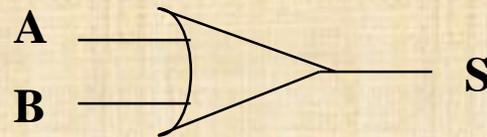
**AND**

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

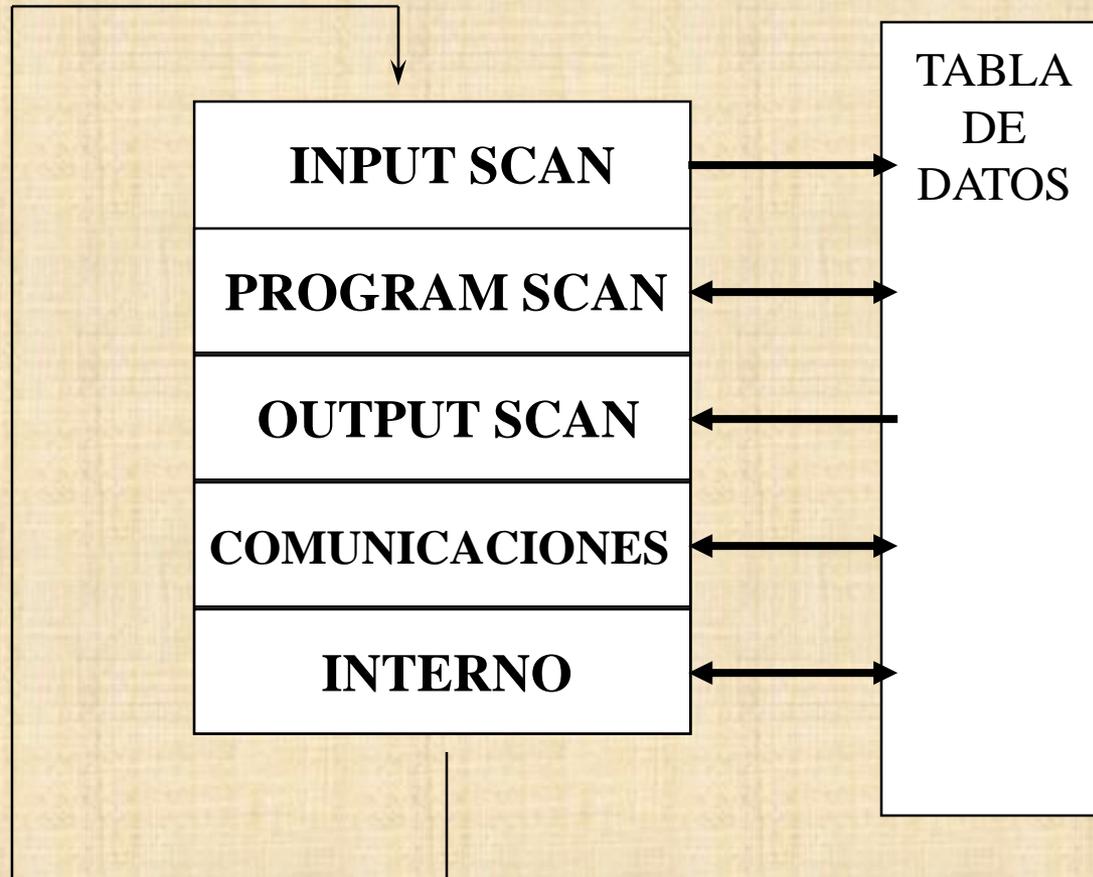


**OR**

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1



# Ciclo de SCAN



# Instrucciones Básicas

- ▶ INSTRUCCIONES DE BIT
- ▶ INSTRUCCIONES DE TEMPORIZADORES
- ▶ INSTRUCCIONES DE CONTADORES

# Instrucciones de BIT

**Condiciones:** → **XIC**  
→ **XIO**  
→ **OSR**

**Acciones:** → **OTE**  
→ **OTL**  
→ **OTU**

# Instrucción “XIC”

- ▶ Se utiliza para determinar si el bit esta activado (1)

## SIMBOLO



## TABLA DE VERDAD

Estado del bit	Instrucción XIC
0	Falsa
1	Verdadera

# Instrucción “XIO”

- ▶ Se utiliza para determinar si el bit esta desactivado (0)

**SIMBOLO**



**TABLA DE VERDAD**

<b>Estado del bit</b>	<b>Instrucción XIC</b>
<b>0</b>	<b>Verdadera</b>
<b>1</b>	<b>Falsa</b>

# Instrucción “OSR”

- ▶ Se utiliza para hacer que una condición sea verdadera solo por un scan

## SIMBOLO



# Instrucción “OTE”

- ▶ Se utiliza para activar o desactivar un bit según las condiciones del renglón sean verdaderas o falsas.

## SIMBOLO



## TABLA DE VERDAD

Condiciones	Acción
0 (Falsa)	Desactiva
1 (Verdadera)	Activa

# Instrucción “OTL / OTU”

- ▶ Son instrucciones que trabajan en conjunto. OTL se usa para activar un bit. OTU se usa para desactivar un bit.

**SIMBOLOS**      ———— [OTL] ————      ———— [OTU] ————

## TABLAS DE VERDAD

### OTL

Condiciones	Acción
0 (Falsa)	-----
1 (Verdadera)	Activa

### OTU

Condiciones	Acción
0 (Falsa)	-----
1 (Verdadera)	Desactiva

# Ejemplo Instrucciones “BiT”

# Instrucciones de Temporizadores

TON     Timer ON Delay

TOF     Timer OFF Delay

RTO     Retentive Timer ON Delay

RES     Reset

# Generalidades

- ▶ Cada instrucción esta formada por 3 palabras:

	<b>15 14 13</b>	
<b>Palabra 0</b>	<b>EN TT DN</b>	<b>Uso interno</b>
<b>Palabra 1</b>	<b>Preset (.Pre)</b>	
<b>Palabra 2</b>	<b>Acumulado (.Acc)</b>	

**EN: Bit de habilitación**

**TT: Bit de temporizador temporizando**

**DN: Bit de finalizado**

**Pre: Valor preseleccionado**

**Acc: Valor acumulado**

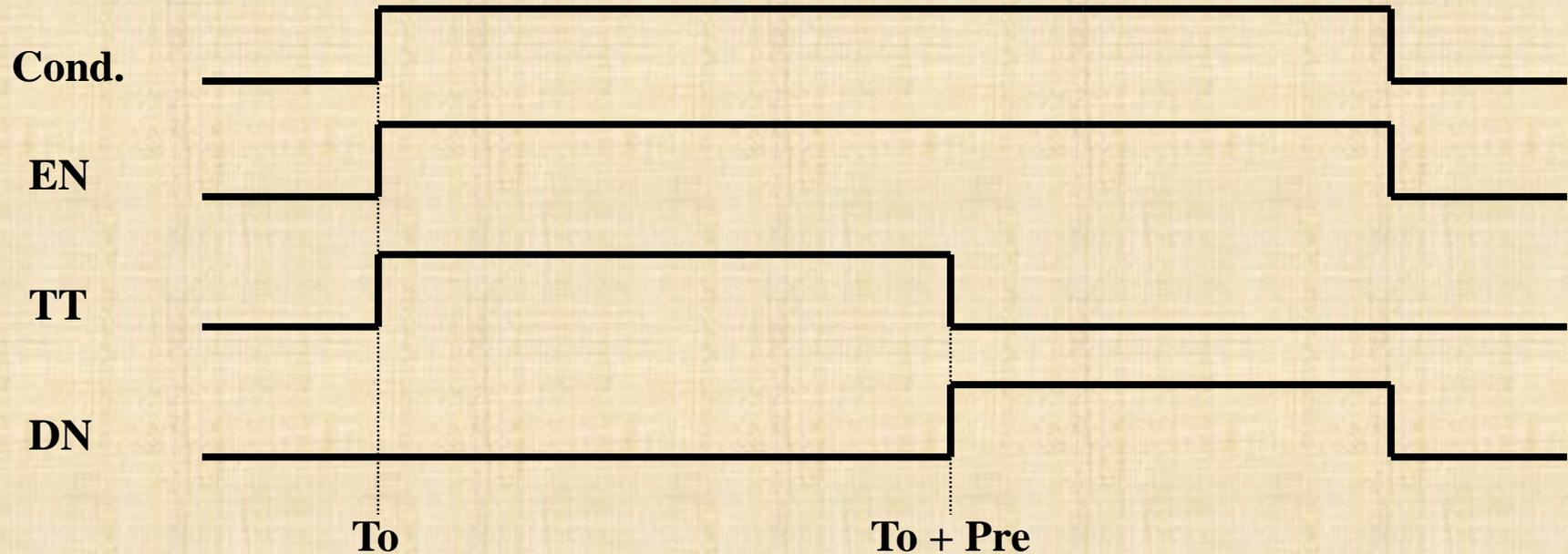
**TB: Base de tiempo (depende del modelo de procesador)**

**Tiempo real = TB x Pre**

Los temporizadores cuentan pulsos de la base de tiempo interna del PLC

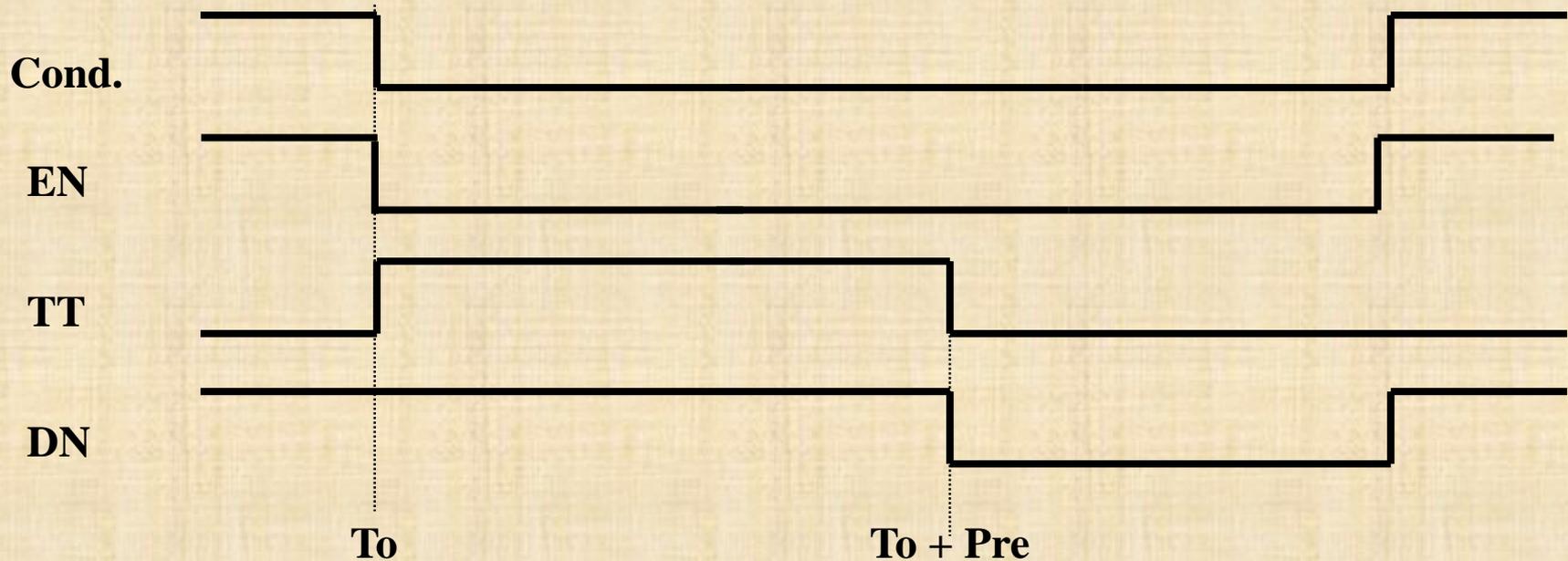
# Instrucción “TON”

- ▶ Temporiza la conexión. Cuenta pulsos de la base de tiempo mientras las condiciones son verdaderas



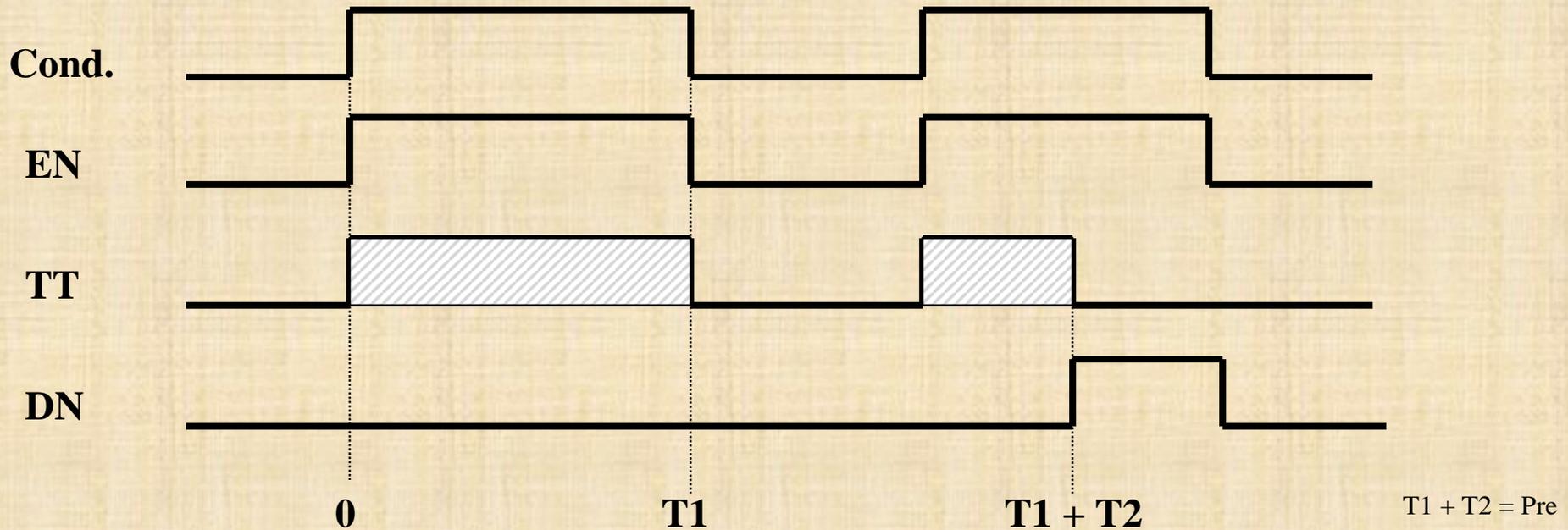
# Instrucción “TOF”

- ▶ Temporiza la desconexión. Cuenta pulsos de la base de tiempo mientras las condiciones son falsas.



# Instrucción “RTO”

- ▶ Es igual que el “TON” pero no restablece el acumulador cuando las condiciones se hacen falsas.



# Ejemplos Temporizadores

# Instrucciones de Contadores

**CTU Counter UP**

**CTD Counter DOWN**

**HSC High Speed Counter**

**RES Reset**

# Generalidades

- ▶ Cada instrucción está formada por 3 palabras:

	<b>15 14 13 12 11 10</b>	
<b>Palabra 0</b>	<b>CU CD DN OV UN UA</b>	<b>Uso interno</b>
<b>Palabra 1</b>	<b>Preset (.Pre)</b>	
<b>Palabra 2</b>	<b>Acumulado (.Acc)</b>	

**CU: Habilitación CTU**

**CD: Habilitación CTD**

**DN: Bit de finalizado**

**OV: Bit de overflow**

**UV: Bit de underflow**

**UA: Actualización de ACC (HSC)**

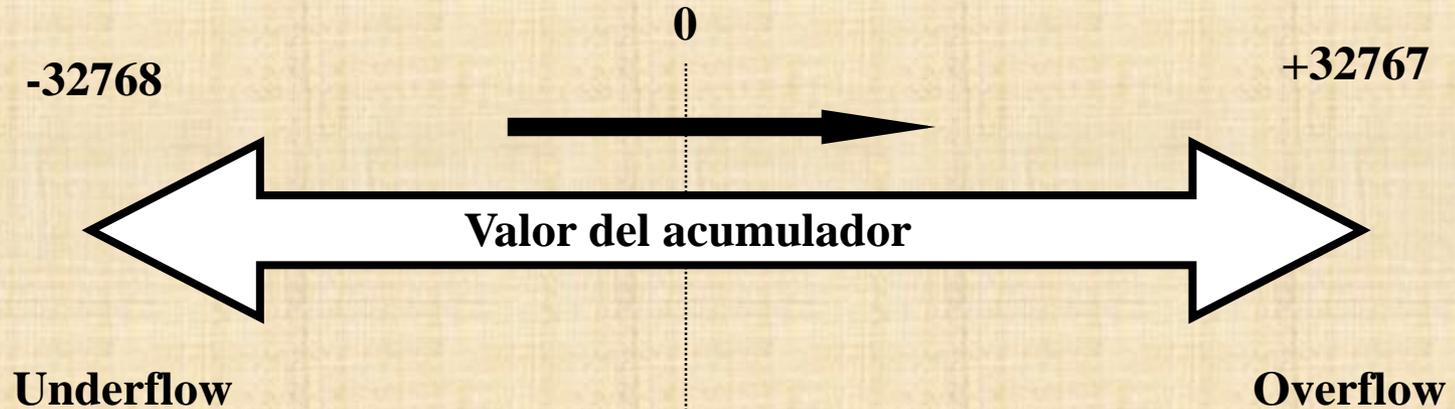
**Pre: Valor preseleccionado**

**Acc: Valor acumulado**

Los contadores cuentan transiciones de FALSO a VERDADERO de las condiciones

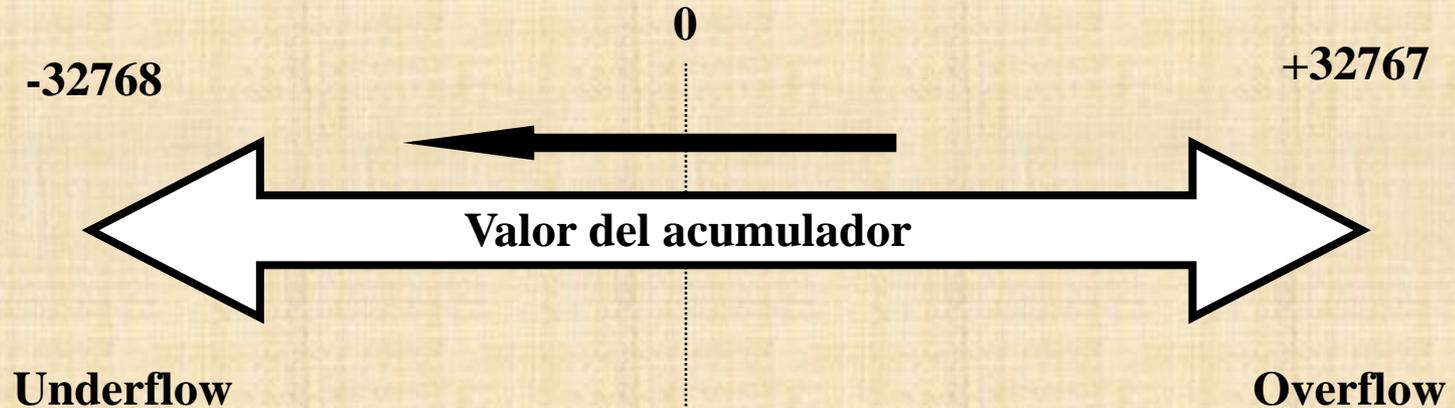
# Instrucción “CTU”

- ▶ Cuenta las transiciones de falso a verdadero de las condiciones, incrementando el acumulador en uno cada vez que esto ocurre.



# Instrucción CTD

- ▶ Cuenta las transiciones de falso a verdadero de las condiciones, decrementando el acumulador en uno cada vez que esto ocurre.



# Instrucción “HSC”

- ▶ Cuenta transiciones de la entrada I:0/0, NO de las condiciones de la línea.
- ▶ La interacción con el programa es asincrónica a la ejecución al mismo.
- ▶ Nos permite contar eventos que son más rápidos que el ciclo de scan

# Instrucción “RES”

- ▶ La instrucción RES sirve para resetear el valor acumulado de los temporizadores y contadores.
- ▶ TON, RTO, CTU, CTD, HSC.

# Ejemplos Contadores

# Instrucciones de comparación

- ▶ El resultado es verdadero o falso
- ▶ Cuando el resultado es verdadero se ejecuta la acción a la derecha de la instrucción
- ▶ Instrucciones: EQU, NEQ, LES, GRT, LEQ, GEQ y LIM
- ▶ En los que tienen dos operandos, el primero tiene que ser una variable, y el segundo puede ser variable o constante.

# Instrucciones de comparación

## EQU - Igual

Compara dos valores, el resultado es verdadero cuando son iguales.

EQU	
Equal	
Source A	B3:2
	0000000000000000
Source B	N7:4
	0

## NEQ – No igual

Compara dos valores, el resultado es verdadero cuando son diferentes.

NEQ	
Not Equal	
Source A	N7:9
	0
Source B	900
	900

# Instrucciones de comparación

## LES – Menor

Compara dos valores, el resultado es verdadero el primero es menor que el segundo.

LES	
Less Than (A<B)	
Source A	N7:6
	0
Source B	775
	775

## GRT – Mayor

Compara dos valores, el resultado es verdadero el primero es mayor que el segundo.

GRT	
Greater Than (A>B)	
Source A	N7:16
	0
Source B	1000
	1000

# Instrucciones de comparación

## LEQ – Menor o igual

Compara dos valores, el resultado es verdadero el primero es menor o igual que el segundo.

LEQ	
Less Than or Eql (A<=B)	
Source A	N7:3
	0
Source B	80
	80

## LEQ – Mayor o igual

Compara dos valores, el resultado es verdadero el primero es mayor o igual que el segundo.

GEQ	
Grtr Than or Eql (A>=B)	
Source A	B22:0
	000000000000000000
Source B	1000
	1000

# Instrucciones de comparación

## LIM – Limite

Compara el valor en **Test**, si es mayor o igual a **Low Lim** y menor o igual a **High Lim**, el resultado es verdadero.

Si **Test** es una constante **Low Lim** y **High Lim** tienen que ser variables. Si **Test** es una variable, **Low Lim** y **High Lim** pueden ser variables o constantes.

LIM	
Limit Test	
Low Lim	N7:9
	0
Test	N7:10
	0
High Lim	1000
	1000

# Ejemplos con instrucciones de comparación

# Instrucciones matemáticas

- ▶ El resultado se guarda en una variable
- ▶ Uno de los operandos tiene que ser una variable, el otro puede ser una variable o constante.
- ▶ Instrucciones: ADD, SUB, MUL, DIV, CPT

# Instrucciones matemáticas

## ADD – Suma

Suma los operandos **Source A** y **Source B**, y guarda el resultado en **Dest**.

ADD	
Add	
Source A	22406
	22406
Source B	N7:3
	0
Dest	N7:12
	0

## SUB – Resta

Resta los operandos **Source A** y **Source B**, y guarda el resultado en **Dest**.

SUB	
Subtract	
Source A	50000.0
	50000.0
Source B	N7:33
	0
Dest	N7:34
	0

# Instrucciones matemáticas

## Mul – Multiplicación

Multiplica los operandos **Source A** y **Source B**, y guarda el resultado en **Dest**.

MUL	
Multiply	
Source A	500
	500
Source B	N30:0
	0
Dest	N30:10
	0

## DIV – División

Divide los operandos **Source A** y **Source B**, y guarda el resultado en **Dest**.

DIV	
Divide	
Source A	N7:20
	0
Source B	44
	44
Dest	N7:5
	0

# Instrucciones matemáticas

## CPT – Computar

Calcula el valor de la expresión y lo guarda en la variable **Dest**. Se pueden utilizar las siguientes operaciones: suma, resta, división, raíz cuadrada, negación, or, or exclusivo, and, tangente, seno, coseno, logaritmo, potencia, etc.

```
CPT
Compute
Dest                                     B3:4
                                         000000000000000000
Expression  SQR(( N7:1 ** 2 ) + ( N7:2 ** 2 ) )
```

# Procesos Analógicos

- ESCALADO DE SEÑALES ANALÓGICAS

INSTRUCCIÓN SCL  
INSTRUCCIÓN SCP

- INSTRUCCIÓN PID

# Instrucción SCL

- La instrucción SCL se usa para convertir los valores de una señal “fuente” a los valores de otra señal “destino”
- Parámetros
  - Source
  - Rate
  - Offset
  - Destino
- Ejemplos y aplicaciones

SCL	
Scale	
Source	N7:6
	0
Rate [10000]	4000
	4000
Offset	100
	100
Dest	N7:20
	0

# Instrucción SCP

- La instrucción SCP se usa para producir valores de salida escalados que tienen una relación lineal con los valores de entrada.

- Parámetros

Input

Input min

Input max

Scaled min

Scaled max

Scaled output

- Ejemplos y aplicaciones

SCP	
Scale w/Parameters	
Input	N7:33
	0
Input Min.	500
	500
Input Max.	5000
	5000
Scaled Min.	N7:8
	0
Scaled Max.	N7:9
	0
Output	B3:0
	0000000000000000

# Instrucción PID

- Un PID (Proporcional Integral Derivativo) es un mecanismo de control por realimentación que se utiliza en sistemas de control industriales. Un controlador PID corrige el error entre un valor medido y el valor que se quiere obtener calculándolo y luego sacando una acción correctora que puede ajustar al proceso acorde. El algoritmo de cálculo del control PID se da en tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor Proporcional determina la reacción del error actual. El Integral genera una corrección proporcional a la integral del error, esto nos asegura que aplicando un esfuerzo de control suficiente, el error de seguimiento se reduce a cero. El Derivativo determina la reacción del tiempo en el que el error se produce. La suma de estas tres acciones es usada para ajustar al proceso vía un elemento de control como la posición de una válvula de control o la energía suministrada a un calentador, por ejemplo.

# Instrucción PID

PID	
Control Block	N7:9
Process Variable	I:9.0
Control Variable	N7:33
Control Block Length	23
Setup Screen	

**PID Setup** [X]

Tuning Parameters	Inputs	Flags
Controller Gain Kc = <input type="text" value="3.5"/>	Setpoint SP = <input type="text" value="8192"/>	TM = <input type="text" value="0"/>
Reset Ti = <input type="text" value="2.5"/>	Setpoint MAX(Smax) = <input type="text" value="16383"/>	AM = <input type="text" value="1"/>
Rate Td = <input type="text" value="0.00"/>	Setpoint MIN(Smin) = <input type="text" value="0"/>	CM = <input type="text" value="0"/>
Loop Update = <input type="text" value="0.50"/>	Process Variable PV = <input type="text" value="16383"/>	OL = <input type="text" value="0"/>
Control Mode = <input type="text" value="E=SP-PV"/>		RG = <input type="text" value="0"/>
PID Control = <input type="text" value="MANUAL"/>		SC = <input type="text" value="0"/>
Time Mode = <input type="text" value="STI"/>		TF = <input type="text" value="0"/>
Limit Output CV = <input type="text" value="NO"/>		DA = <input type="text" value="0"/>
Deadband = <input type="text" value="0"/>		DB = <input type="text" value="0"/>
Feed Forward Bias = <input type="text" value="0"/>		UL = <input type="text" value="0"/>
		LL = <input type="text" value="0"/>
		SP = <input type="text" value="0"/>
		PV = <input type="text" value="1"/>
		DN = <input type="text" value="1"/>
		RA = <input type="text" value="0"/>
		EN = <input type="text" value="1"/>

Control Output CV (%) =   
Output Max CV (%) =   
Output Min CV (%) =   
Scaled Error SE =

OK Cancel Help

# INGRESO AL SISTEMA

**ENCENDER EQUIPO**

**INGRESAR USUARIO**

**INGRESAR CONTRASEÑA**

**MENU INICIO**

**PROGRAMAS**

**ROCKWELL SOFTWARE**

**RSLOGIX 500 English**

**MENU INICIO**

**RSLOGIX 500 English**

**PROGRAMAS**

**ROCKWELL SOFTWARE**

**RSLINX**

**RSLINX**