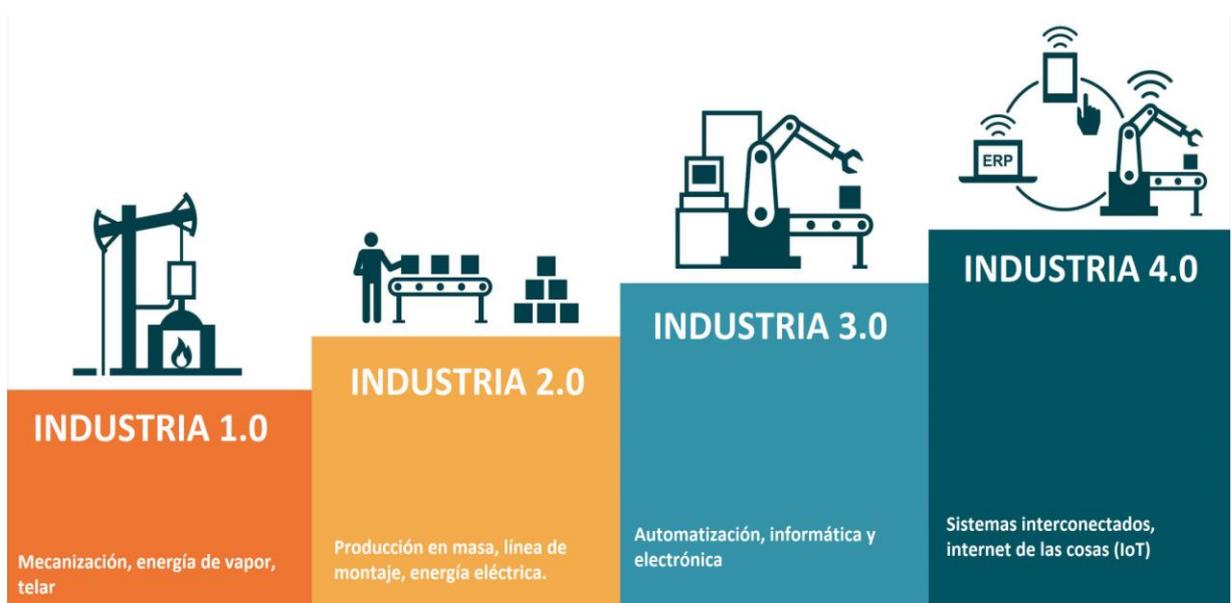




Control Lógico Programable Básico



Índice

Introducción	1
Objetivos	2
Tema 1.- Conceptos básicos	4
Tema 2.- Componentes de un sistema PLC	15
Tema 3.- Organización de la memoria y redireccionamiento	28
Tema 4.- Diagramas de escalera e instrucciones tipo relé	36
Tema 5.- Temporizadores	70
Conclusión	107
Bibliografía	108

Introducción

El ICET, es una institución social, que brinda servicios de capacitación y certificación para el trabajo y en el trabajo, elevando la empleabilidad de nuestros participantes y propiciando la productividad de las empresas, a través de un modelo académico y con estándares de competencia laboral.

Debido a la gran importancia que actualmente tiene el control de los procesos en la industria así como la necesidad de contar con mano de obra calificada para resolver problemáticas de campo en las que están involucrados equipos de alta tecnología vitales para el óptimo funcionamiento de la planta productiva, el ICET se ha preocupado por ayudar a las empresas diseñando cursos de capacitación en estos temas que permitan que el personal que interactúa día con día con las líneas de producción adquiera los conocimientos y habilidades necesarias para resolver cualquier problemática que se presente en los procesos mediante la utilización de estas herramientas.

Objetivo General

- ✓ Al término del curso, el participante aplicará el software de programación de PLC mediante el dominio de sus partes fundamentales y aplicaciones comunes.

Objetivos Particulares

- ✓ Al término del tema, el participante identificará las partes fundamentales de un sistema basado en Control Lógico Programable (PLC) de acuerdo al programa PLC 800 a un nivel básico.
- ✓ Al término del tema, el participante reconocerá las partes que componen el PLC 800 de acuerdo al sistema operativo correspondiente a un nivel básico.
- ✓ Al término del tema, el participante definirá a un nivel básico la forma de diagnosticar fallas en aplicaciones reales de PLC' s y mantenimiento preventivo.
- ✓ Al término del tema, el participante repetirá la forma adecuada de instalar y configurar un diagrama de lógica de escalera a un nivel básico.
- ✓ Al término del tema, el participante practicará la elaboración de un programa básico de acuerdo con la capacitación recibida.

PLC. -controlador lógico programable.

CHASIS. -es el receptáculo físico en donde se inserta el controlador programable.

PUNTO DIGITAL DE I/O .- Es la unidad mínima de identificación y ubicación el cual representa una entrada o salida física.

SLOT. -Espacio para módulo en el chasis.

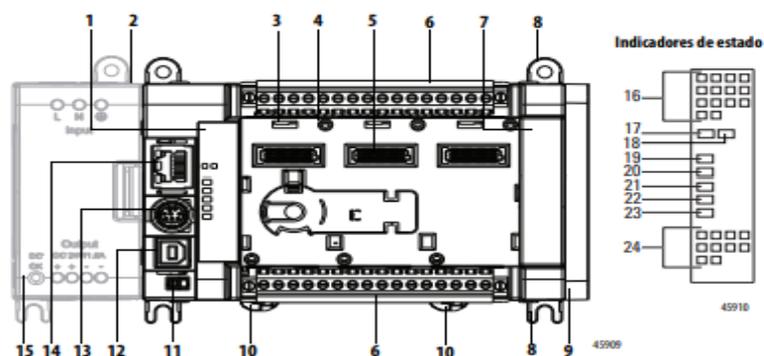
IDENTIFICACION DE UN PUNTO DE I/O.- Para identificar completamente un punto de I/O es necesario conocer la ubicación del mismo, lo cual nos da una dirección física y lógica.

ORGANIZACIÓN DE LA MEMORIA. - La estructura de la memoria en la familia de controladores programables Micro 800 de Allen Bradley está dividida en los siguientes archivos.

UDFB. -bloque de funciones definidas por el usuario.

Controladores Micro850

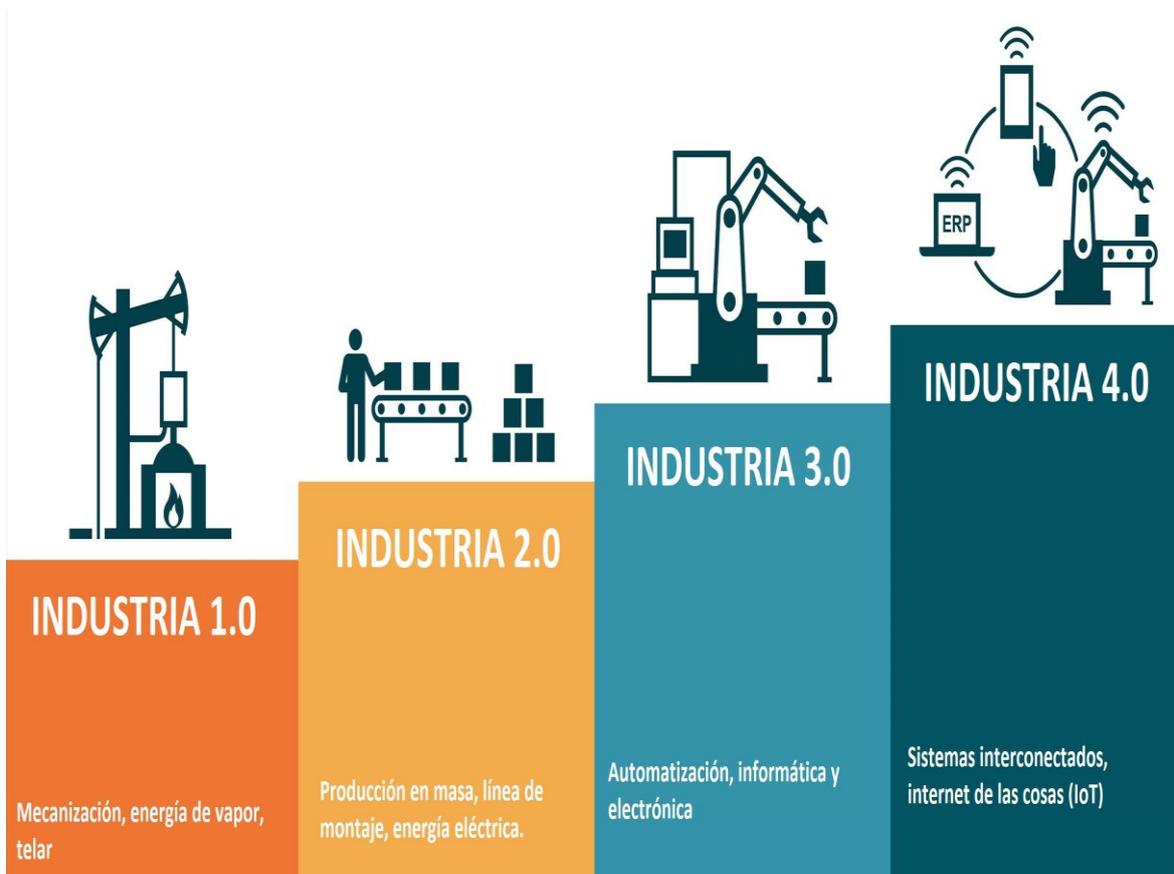
Controladores Micro850 de 24 puntos e indicadores de estado



Tema 1.-

Conceptos básicos

Objetivo particular.- Al término del tema, el participante identificará las partes fundamentales de un sistema basado en Control Lógico Programable (PLC) de acuerdo al programa PLC 800 a un nivel básico.



En la actualidad definir un controlador programable no es muy sencillo, debido a los avances tecnológicos que estos poseen y el gran campo de aplicación que tienen, pero a continuación se da una definición que menciona algunas de sus características.

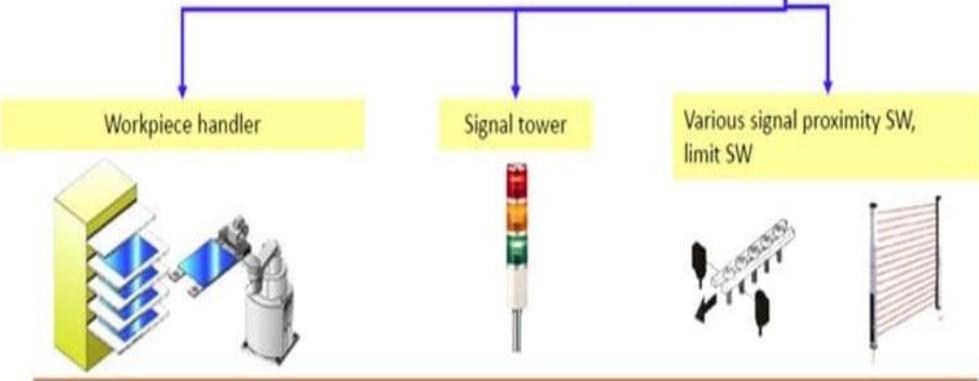
Definición: Es un dispositivo electrónico digital, su construcción está hecha en base a un micro procesador y memoria que se usa para emular; las facilidades de conectividad son muy grandes, se manejan cualquier tipo de señal de entrada o salida, sea digital o análoga, cualquier tipo de señal de campo y distintos tipos de redes de comunicación para tener conectividad.

Es además programable, para almacenar instrucciones con funciones específicas como:

1. Funciones lógicas secuenciales, tipo relevador.
2. Temporizadores (Timers)
3. Contadores
4. Control regulatorio de lazos
5. Monitoreo de procesos, etc.

Con el fin de utilizarlos en el control y monitoreo de procesos.

En la actualidad se conocen únicamente como Controladores Programables ya que no solo realizan funciones lógicas sino muchas más funciones en el campo analógico, digital y de comunicaciones.



The diagram illustrates three categories of sensors used in industrial automation, each with a corresponding image:

- Workpiece handler:** Shows a robotic arm interacting with a stack of blue blocks.
- Signal tower:** Shows a vertical stack of colored lights (red, yellow, green, blue).
- Various signal proximity SW, limit SW:** Shows a variety of sensors, including a proximity switch and a limit switch.

- Botones
- Switches de Selección
- Switches de Limite
- Switches de Nivel
- Sensores Photoelectricos
- Sensores de Proximidad
- Sensores de Temperatura



Antecedentes históricos

El criterio de diseño de un **controlador programable** fue especificado por primera vez en 1968 por la General Motors, que por tener una producción en serie y muy cambiante requería de un control mucho más flexible, acorde a sus cambios de modelo y eliminando el costo tan alto de los tradicionales tableros de relevadores.

Las especificaciones requerían un sistema de estado sólido con la flexibilidad de una computadora, con capacidad de soportar el ambiente industrial y además que fuese de fácil programación para ingenieros y técnicos de planta. También se necesitaba reducir los tiempos de falla de la máquina, así como permitir expansiones a futuro.

A medida que el avance tecnológico de la electrónica en el campo de los microprocesadores ha permitido impulsar enormemente a los **controladores programables** dotándolos cada vez más con mayor velocidad, flexibilidad e inteligencia.

Las facilidades de conectividad son muy grandes, se manejan cualquier tipo de señal de entrada o salida, sea digital o analógica, cualquier tipo de señal de campo, y distintos tipos de redes de comunicación para tener conectividad.

Si comparamos los primeros controladores programables con los de hoy en día, veremos que la filosofía y principios de operación han tratado de seguir por una misma línea, con las ventajas que esto significa en cuanto a la operación, programación y tiempo de uso útil.

La tendencia tecnológica de los **controladores programables** promete avances a mediano y largo plazo como son:

- ✓ Integrar sofisticados sistemas de visión.
- ✓ Sistemas de reconocimiento de voz.
- ✓ Sistemas de inteligencia artificial.

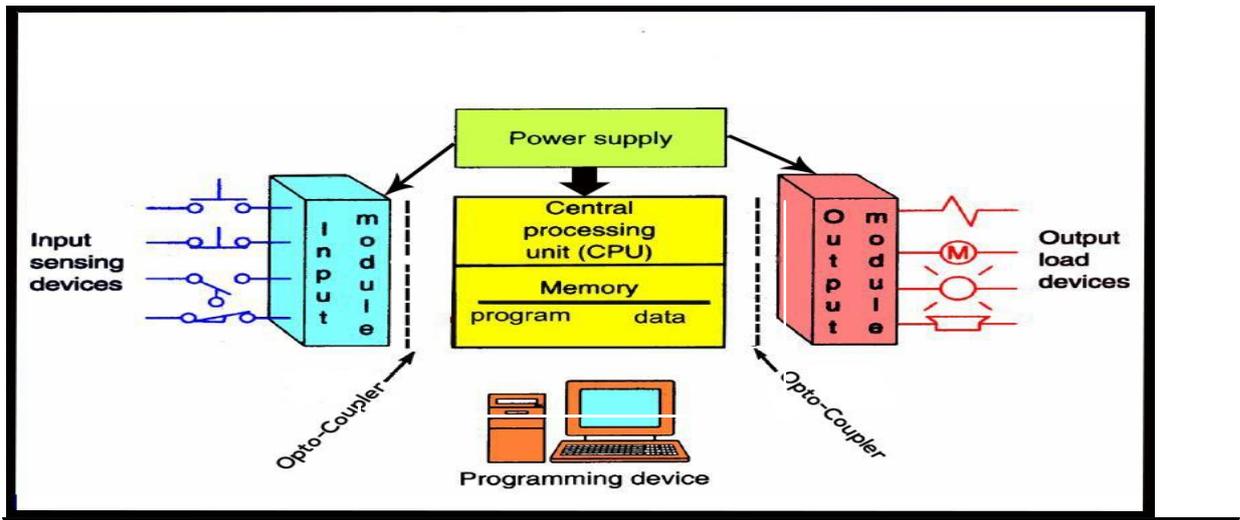
Ventajas

Alta flexibilidad
Poco mantenimiento
Fácil programación
Espacio reducido
Facilidades de comunicación
Bajo consumo de energía
Confiable

Estructura de un PLC

Los **controladores programables** están constituidos de tres partes principales que son unidad de procesamiento, interfaces para señales de entrada e interfaces de salida.

Teoría del SCAN



Uno de los conceptos más importantes de operación de un **controlador programable** es el principio del scan, el cual describe la forma en que se resuelve la lógica de programación que se le ha dado. Primeramente, el controlador programable se encarga de leer el estado de todas las señales de entrada que tiene conectadas a él, después resuelve la lógica del programa de arriba hacia abajo y de izquierda a derecha y finalmente actualiza en forma física el estado de las salidas conectadas y controladas por él mismo, esto se conoce como tiempo de scan, el cual se divide en E/S scan y programa scan, este es medido en milisegundos $\times K$ de memoria programada.

Uno de los conceptos más importantes de operación de un **controlador programable** es el principio del SCAN, el cual describe la forma en que se resuelve la lógica de programación que se le ha dado. Primeramente, el controlador programable se encarga de leer el estado de todas las señales de entrada que tiene conectadas a él, después resuelve la lógica del programa de arriba hacia abajo y de izquierda a derecha y finalmente actualiza en forma física el estado de las salidas conectadas y controladas por él mismo, esto se conoce como tiempo de SCAN, el cual se divide en E/S scan y programa scan, este es medido en milisegundos x K de memoria programada.



El controlador programable tiene watch dog timer para verificar que el procesador no se quede ciclado en el programa o que tenga alguna falla.

Tipos de señales que pueden manejar

Un controlador programable puede manejar una amplia variedad de tipo de señales que se pueden clasificar en su totalidad como:

Entradas		Salidas	
Digitales		Análogas	
C o r r i e n t e A l t e r n a	C o r r i e n t e D i r e c t a	V o l t a j e	C o r r i e n t e

Señales de Entrada

ELEMENTOS DE CAMPO QUE MANEJAN SEÑALES DIGITALES

- ✓ Micro switches
- ✓ Interruptores depresión
- ✓ Nivel, temperatura, etc.
- ✓ Selectores
- ✓ Botones

ELEMENTOS DE CAMPO QUE MANEJAN SEÑALES ANALOGAS

- ✓ Transmisores de presión, flujo
- ✓ Termopares
- ✓ RTD's
- ✓ Transductores en general

NIVELES DE VOLTAJE QUE MANEJAN LAS SEÑALES DIGITALES	NIVELES DE VOLTAJE QUE MANEJAN LAS SEÑALES ANALOGAS
110 VAC oVDC	0 a 10volts
12 – 24VDC	1 a 5volts
0-30VDC	0 a 20mA
220VAC	4 a 20mA

Señales de Salida

ELEMENTOS DE CAMPO QUE MANEJAN SEÑALES DIGITALES

- ✓ Contactor
- ✓ Lámparas indicadoras
- ✓ Display
- ✓ Alarmas sonoras
- ✓ Solenoides

ELEMENTOS DE CAMPO QUE MANEJAN SEÑALES ANALOGAS

- ✓ Drives
- ✓ Servo válvulas
- ✓ Damper's
- ✓ Servo mecanismos
- ✓ Niveles

NIVELES DE VOLTAJE QUE MANEJAN LAS SEÑALES DIGITALES	NIVELES DE VOLTAJE QUE MANEJAN LAS SEÑALES ANALOGAS
110 VAC oVDC	0 a 10volts
12 – 24VDC	1 a 5volts
0-30VDC	0 a 20mA
220VAC	4 a 20mA

Equipo periférico

Es aquel que se utiliza para las funciones auxiliares del controlador y que no son prioritarias en las actividades del control.

Ejemplos:

- Terminales industriales
- Computadoras de Programación
- Computadoras de Monitoreo
- Impresoras
- etc.



Aplicaciones

Los controladores programables son de aplicación universal, por lo que lo mismo vamos a encontrar controladores en la industria metalmecánica que, en la industria productora de alimentos, etc.

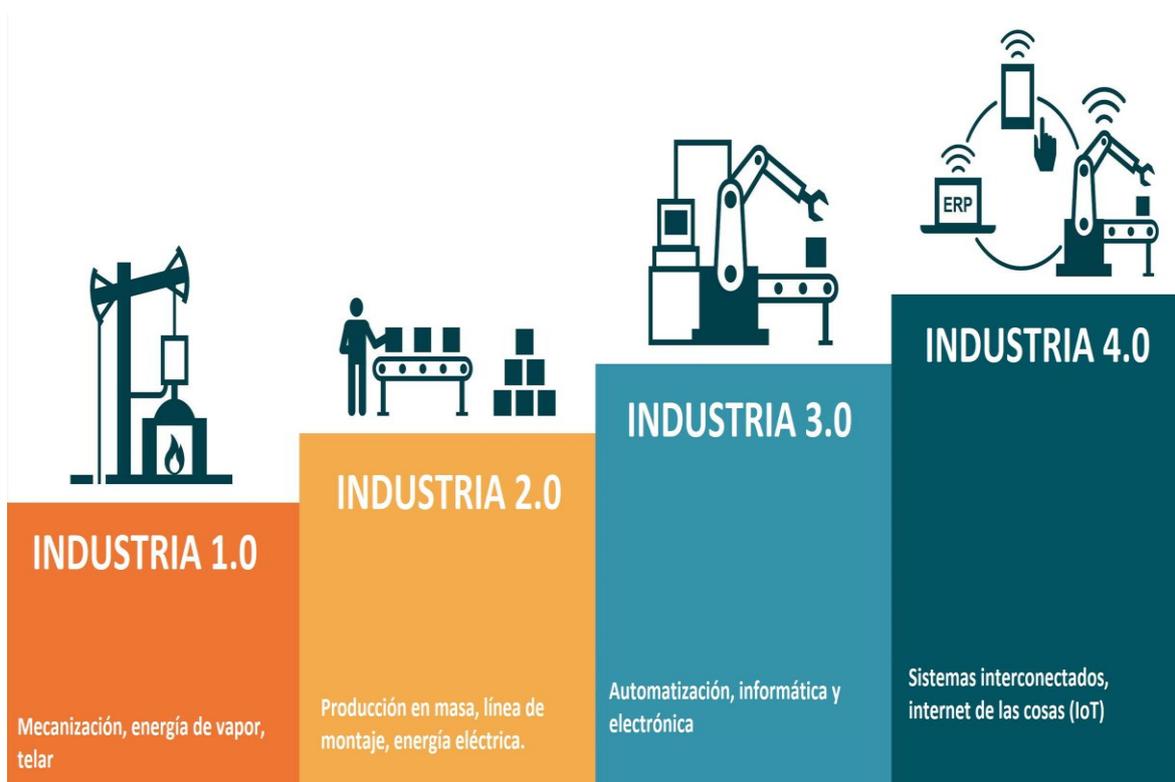
Industria	Procesos
1.- Industria química y petroquímica.	<ul style="list-style-type: none"> ✓ Procesos Bach ✓ Manejo de materiales ✓ Mezclado
2. - Maquinado y manufactura.	<ul style="list-style-type: none"> ✓ Control de demanda ✓ Ensamble de máquinas ✓ Bancos de prueba
3.- Minería.	<ul style="list-style-type: none"> ✓ Transporte de material ✓ Carga y descarga de material ✓ Lavado de material
4.- Industria del papel	<ul style="list-style-type: none"> ✓ Manejo de cortés ✓ Grabado y estampado ✓ Enrollado
5.- Industria del vidrio	<ul style="list-style-type: none"> ✓ Proceso u fundición ✓ Formado ✓ Acabado ✓ Empacado
6.- Alimentos	<ul style="list-style-type: none"> ✓ Procesamiento ✓ Pesado ✓ Embotellado y enlatado

Tema 2.-

Componentes

de un sistema PLC

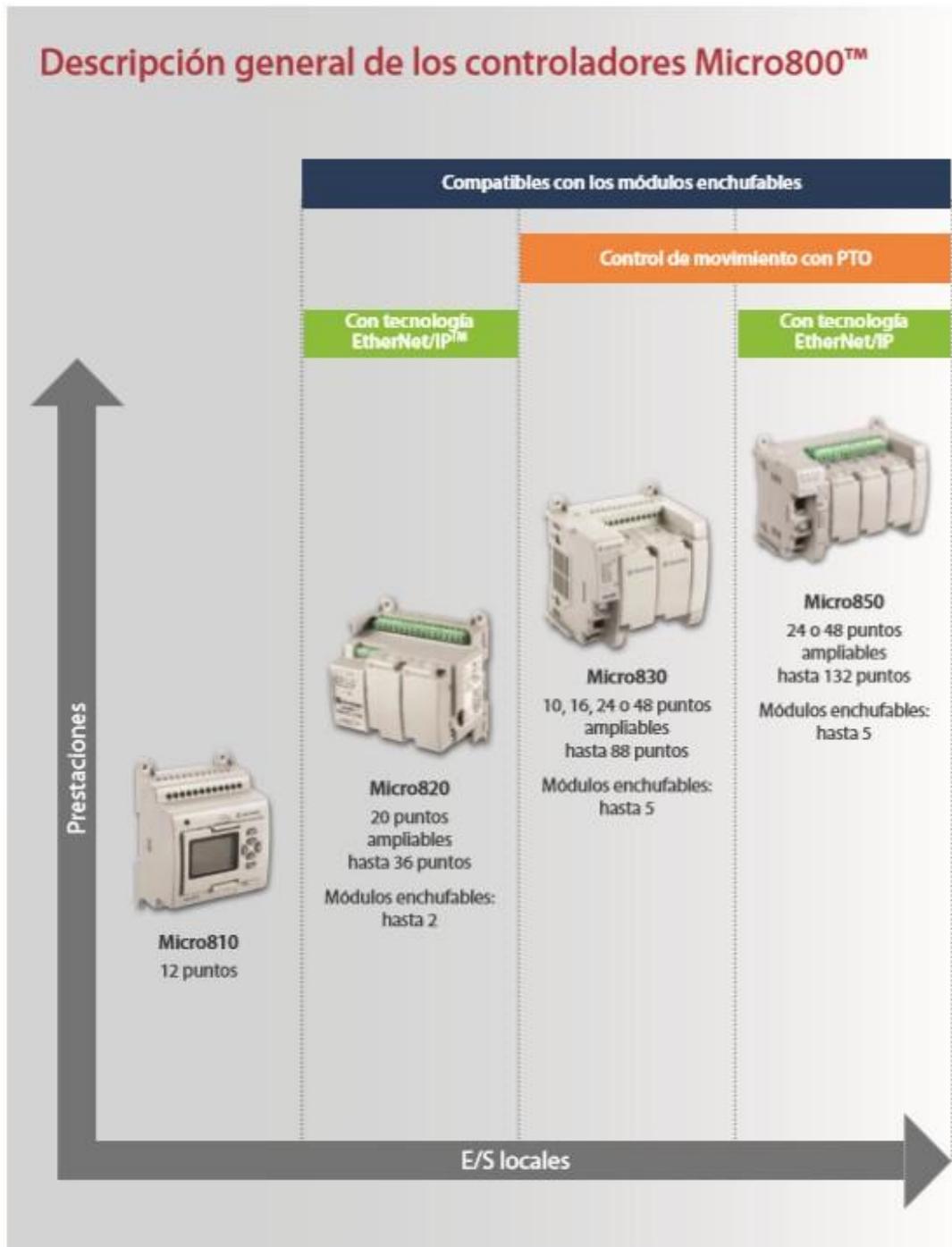
Objetivo particular.- Al término del tema, el participante reconocerá las partes que componen el PLC 800 de acuerdo al sistema operativo correspondiente a un nivel básico.



Los controladores programables Micro 800 son una herramienta vital para la adecuada operación de las plantas Industriales, debido a los avances tecnológicos que estos poseen y el gran campo de aplicación que tienen, a continuación, se describen los diferentes modelos y componentes con sus características.



Familias de Controladores Programables Micro800



Características Principales de Equipos Familia Micro800

Atributo	Micro810	Micro820	Micro830				Micro850	
	12 puntos	20 puntos	10 puntos	16 puntos	24 puntos	48 puntos	24 puntos	48 puntos
Puertos de comunicación, incorporados	USB 2.0 (con adaptador de USB)	Puerto 10/100 Base T Ethernet (RJ-45) Combinación de RS232/RS485 en serie no aislado	USB 2.0 (no aislado) Combinación de RS232/RS485 en serie no aislado				USB 2.0 (no aislado) Combinación de RS232/RS485 en serie no aislado Puerto 10/100 Base T Ethernet (RJ-45)	
Puntos de E/S digitales incorporados ⁽¹⁾	12	19	10	16	24	48	24	48
Canales de E/S analógicas base	Cuatro entradas digitales de 24 VCC se comparten como entradas analógicas de 0...10 V (modelos de entrada de CC solamente)	Una salida analógica de 0...10 V Cuatro entradas digitales de 24 VCC pueden configurarse como entradas analógicas de 0...10 V (modelos de entrada de CC solamente) y mediante módulos enchufables	mediante módulos enchufables				mediante módulos enchufables y E/S de expansión	
Número de módulos enchufables	0	2	2	2	3	5	3	5
Máximo de E/S digitales ⁽²⁾	12	35	26	32	48	88	132	
Tipos de accesorios o módulos enchufables aceptados	<ul style="list-style-type: none"> Pantalla LCD con módulo de memoria de respaldo Adaptador de USB 	<ul style="list-style-type: none"> Micro800 LCD remoto (2080-RIMLCD) Todos los módulos enchufables excepto 2080-MLMBAK-RTC (vea la página 51) 	Todos los módulos enchufables (vea la página 51)					
E/S de expansión compartidas	–	–	–				Todos los módulos de E/S de expansión (vea la página 41)	
Fuente de alimentación eléctrica	Opciones de 120/240 VCA y 12/24 VCC	La base tiene una fuente de alimentación de 24 VCC incorporada, y una fuente de alimentación opcional externa de 120/240 VCA disponible						
Velocidad de instrucción básica	2.5 µs por instrucción básica	0.30 µs por instrucción básica						
Tiempo mínimo de escán/ciclo ⁽³⁾	< 0.25 ms	< 4 ms	< 0.25 ms					
Software	Connected Components Workbench							

Atributo	Micro810, 12 puntos	Micro820, 20 puntos	Micro830, 10/16 puntos	Micro830, 24 puntos	Micro830, 48 puntos	Micro850, 24 puntos	Micro850, 48 puntos
Pasos del programa ⁽¹⁾	2 K	10 K	4 K	10 K	10 K	10 K	10 K
Bytes de datos	2 KB	20 KB	8 KB	20 KB	20 KB	20 KB	20 KB
Lenguajes IEC 61131-3	Diagrama de lógica de escalera, diagrama de bloques de funciones, texto estructurado						
Bloques de funciones definidos por el usuario	Sí						

Atributo	Micro810, 12 puntos	Micro820, 20 puntos	Micro830, 10/16 puntos	Micro830, 24 puntos	Micro830, 48 puntos	Micro850, 24 puntos	Micro850, 48 puntos
Punto flotante (coma flotante)	32 bits y 64 bits						
Control de lazo PID	Sí (número limitado sólo por la memoria)						
Protocolos de puerto serie incorporados	Ninguno	Maestro/esclavo RTU Modbus, ASCII/Binario, CIP en serie					

Controlador	Puerto de programación USB	Puerto serie incorporado, puerto serie enchufable			Ethernet incorporada	
		CIP en serie	Modbus RTU	ASCII/binario	EtherNet/IP	Modbus TCP
Micro810	Sí (con adaptador)	No				
Micro820	Sí (con 2080-REMLCD)	Sí	Maestro/esclavo	Sí	Sí	Sí
Micro830	Sí	Sí	Maestro/esclavo	Sí	No	No
Micro850	Sí	Sí	Maestro/esclavo	Sí	Sí	Sí

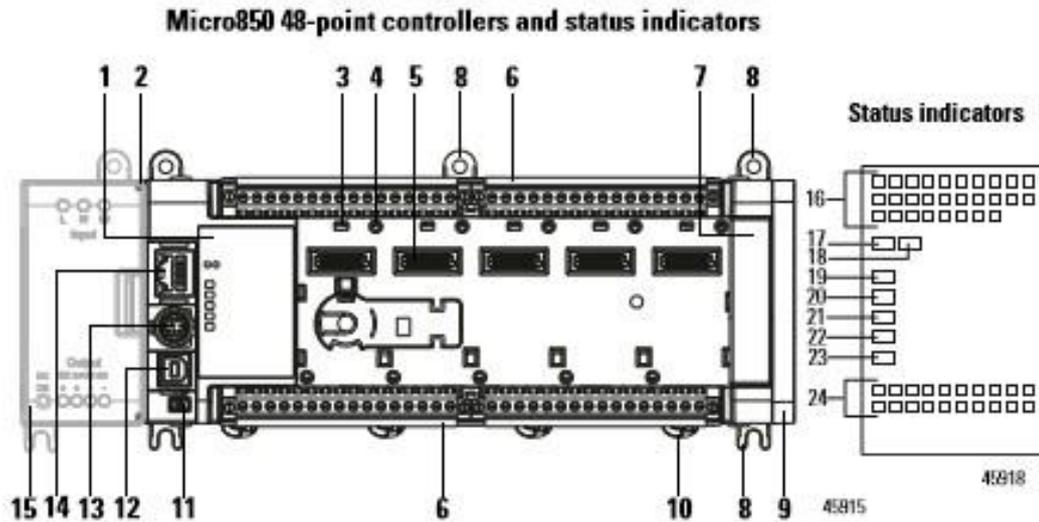
Atributo	Micro810	Micro820	Micro800 (con componentes enchufables)	Micro850 (con E/S de expansión)
Nivel de rendimiento	Bajo	Bajo	Mediano	Alto
Aislamiento al controlador (mayor inmunidad al ruido)	Ninguno	Ninguno	Ninguno	Sí
Resolución y exactitud nominal	Entrada analógica: 10 bits, 5% (2% con calibración)	E/S analógicas: 12 bits, 5% (2% con calibración)	E/S analógicas: 12 bits, 1% TC/RTD: ±1 °C CIC para TC: ±1.2 °C	Entrada analógica: Entrada de 14 bits, ±0.1% Salida analógica Salida de 12 bits, 0.133%, corriente, 0.425% voltaje TC: ±0.5 ... ±3.0 °C RTD: ±0.2 ... ±0.6 °C
Régimen de actualización de entrada y filtro	El régimen de actualización sólo depende del escán del programa, filtro limitado	El régimen de actualización sólo depende del escán del programa, filtro limitado	200 ms/cn, filtro de 50/60 Hz	8 ms todos los canales con o sin filtro de 50/60 Hz
Longitud máxima de cable blindado recomendada ⁽¹⁾	10 m			100 m

Número y tipos de entradas/ salidas

Número y tipos de entradas/ salidas para catálogos Micro810, Micro820, Micro830 y Micro850

Familia de controlador	Catálogos	Entradas				Salidas			Sal. analógica 0...10 VCC	Ent. analógica 0...10 V(compartida con ent. de CC)	Compatibilidad con PTO/PWM ⁽¹⁾	Compatibilidad con HSC ⁽²⁾
		120 VCA	120/240 VCA	24 VCC/ VCA	12 VCC	Relé	24 VCC surtidor	24 VCC drenador				
Micro810	2080-1C10-12QWB	-	-	8	-	4	-	-	-	4	-	-
	2080-1C10-12AWA	-	8	-	-	4	-	-	-	-	-	-
	2080-1C10-12QBB	-	-	8	-	-	4	-	-	4	-	-
	2080-1C10-12DWD	-	-	-	8	4	-	-	-	4	-	-
Micro820	2080-1C20-20QBB	-	-	12	-	-	7	-	1	4	1 (PWM)	-
	2080-1C20-20QWB	-	-	12	-	7	-	-	1	4	-	-
	2080-1C20-20AWB	8	-	4	-	7	-	-	1	4	-	-
	2080-1C20-20QBBR	-	-	12	-	-	7	-	1	4	1 (PWM)	-
	2080-1C20-20QWBR	-	-	12	-	7	-	-	1	4	-	-
	2080-1C20-20AWBR	8	-	4	-	7	-	-	1	4	-	-
Micro830	2080-1C30-10QWB	-	-	6	-	4	-	-	-	-	-	2
	2080-1C30-10QVB	-	-	6	-	-	-	4	-	-	1 (PTO/PWM)	2
	2080-1C30-16AWB	10	-	-	-	6	-	-	-	-	-	-
	2080-1C30-16QWB	-	-	10	-	6	-	-	-	-	-	2
	2080-1C30-16QVB	-	-	10	-	-	-	6	-	-	1 (PTO/PWM)	2
	2080-1C30-24QWB	-	-	14	-	10	-	-	-	-	-	4
	2080-1C30-24QVB	-	-	14	-	-	-	10	-	-	2 (PTO/PWM)	4
	2080-1C30-24QBB	-	-	14	-	-	10	-	-	-	2 (PTO/PWM)	4
	2080-1C30-48AWB	28	-	-	-	20	-	-	-	-	-	-
	2080-1C30-48QWB	-	-	28	-	20	-	-	-	-	-	6
	2080-1C30-48QVB	-	-	28	-	-	-	20	-	-	3 (PTO/PWM)	6
	2080-1C30-48QBB	-	-	28	-	-	20	-	-	-	3 (PTO/PWM)	6
Micro850	2080-1C50-24AWB	14	-	-	-	10	-	-	-	-	-	-
	2080-1C50-24QBB	-	-	14	-	-	10	-	-	-	2 (PTO/PWM)	4
	2080-1C50-24QVB	-	-	14	-	-	-	10	-	-	2 (PTO/PWM)	4
	2080-1C50-24QWB	-	-	14	-	10	-	-	-	-	-	4
	2080-1C50-48AWB	28	-	-	-	20	-	-	-	-	-	-
	2080-1C50-48QWB	-	-	28	-	20	-	-	-	-	-	6
	2080-1C50-48QBB	-	-	28	-	-	20	-	-	-	3 (PTO/PWM)	6
	2080-1C50-48QVB	-	-	28	-	-	-	20	-	-	3 (PTO/PWM)	6

Componentes Principales de los Equipos Micro800



Controller Description

Description	Description
1 Status indicators	9 Expansion I/O slot cover
2 Optional power supply slot	10 DIN rail mounting latch
3 Plug-in latch	11 Mode switch
4 Plug-in screw hole	12 Type B connector USB port
5 40-pin high speed plug-in connector	13 RS232/RS485 non-isolated combo serial port
6 Removable I/O terminal block	14 RJ-45 EtherNet/IP connector (with embedded yellow and green LED indicators)
7 Right-side cover	15 Optional AC power supply
8 Mounting screw hole / mounting foot	

Status Indicator Description⁽¹⁾

Description	Description
16 Input status	21 Fault status
17 Module status	22 Force status
18 Network status	23 Serial communications status
19 Power status	24 Output status
20 Run status	

Módulos de Entrada, Salida y Especiales



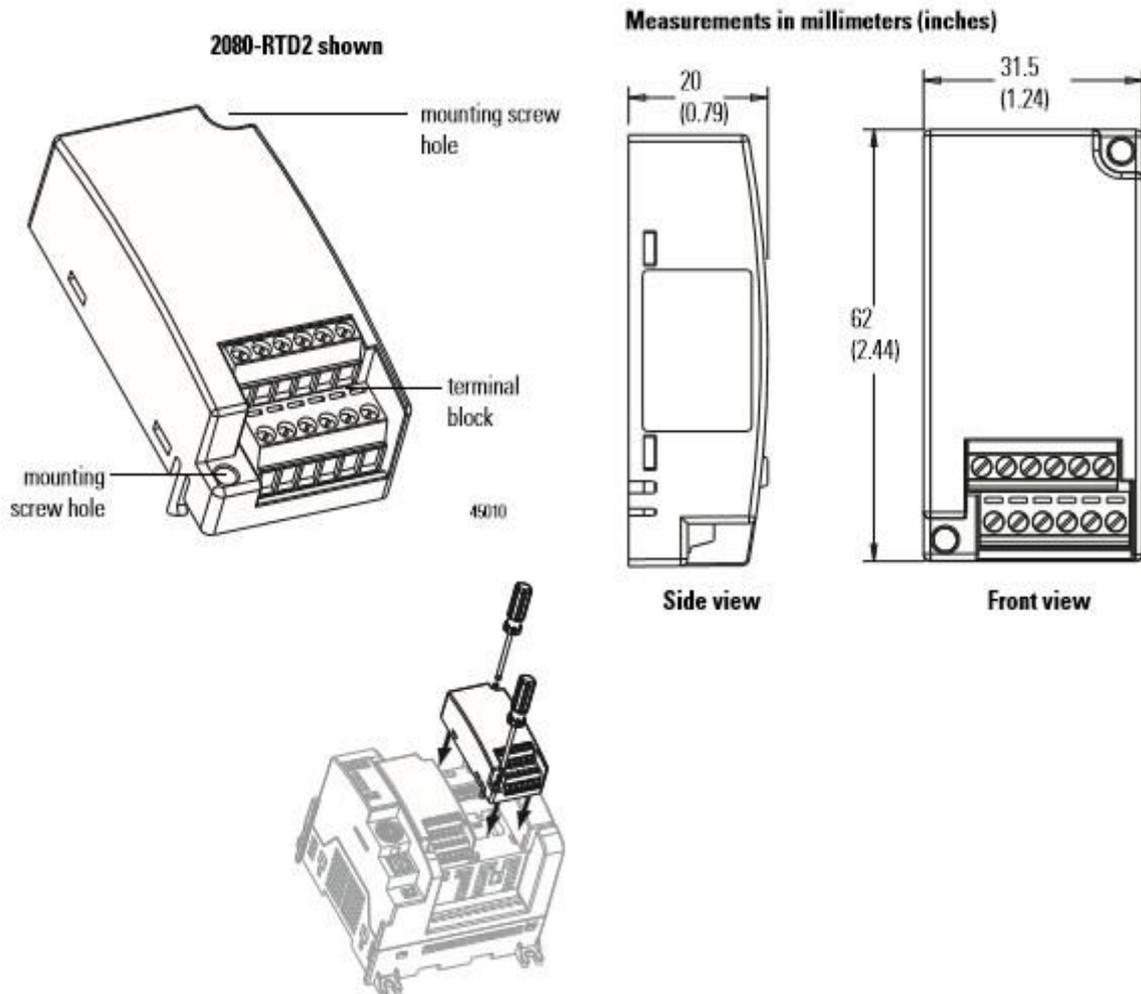
Module	Type	Description
2080-IO4	Digital	4-point, 12/24V DC Sink/Source input
2080-IO4OB4	Digital	8-point, Combo, 12/24V DC Sink/Source input 12/24V DC Source output
2080-IO4OV4	Digital	8-point, Combo, 12/24V DC Sink/Source input 12/24V DC Sink output
2080-OB4	Digital	4-point, 12/24V DC Source output
2080-OV4	Digital	4-point, 12/24V DC Sink output
2080-OW4I	Digital	4-point, AC/DC Relay output
2080-IF2	Analog	2-channel, Non-isolated unipolar voltage/current analog input
2080-IF4	Analog	4-channel, Non-isolated unipolar voltage/current analog input

Module	Type	Description
2080-OF2	Analog	2-channel, Non-isolated unipolar voltage/current analog output
2080-TC2	Specialty	2-channel, non-isolated thermocouple module
2080-RTD2	Specialty	2-channel, non-isolated RTD module
2080-MEMBAK-RTC ⁽¹⁾	Specialty	Memory backup and high accuracy RTC, 1 MB
2080-MEMBAK-RTC ⁽¹⁾	Specialty	Memory backup and high accuracy RTC, 4 MB
2080-TRIMPOT6	Specialty	6-channel trimpot analog input
2080-MOT-HSC	Specialty	High speed counter
2080-DNET20	Communication	20-node DeviceNet scanner
2080-SERIALISOL	Communication	RS232/485 isolated serial port

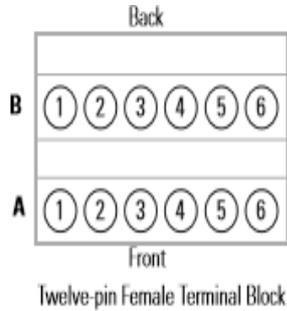
Plug-in Slots on Micro800 Controllers

Controller	Number of Plug-in Slots
Micro810	0
Micro820	2
Micro830	2 (10/16 points) 3 (24 points) 5 (48 points)
Micro850	3 (24 points) 5 (48 points)
Micro870	3

Instalación de los módulos



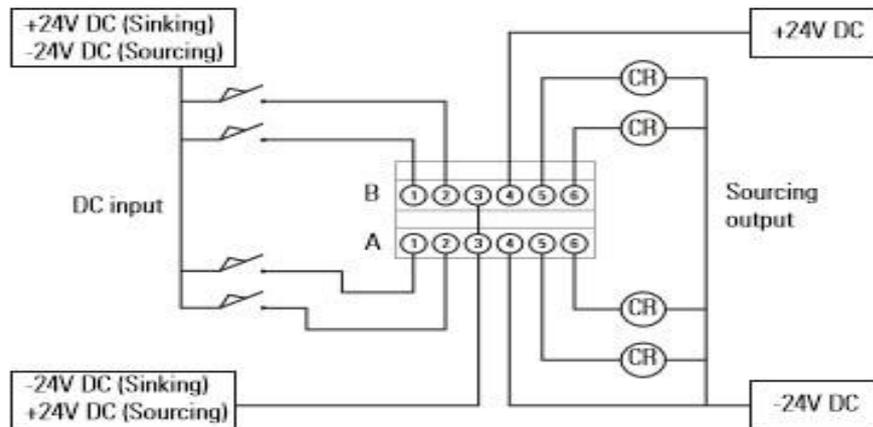
Cableado de Señales



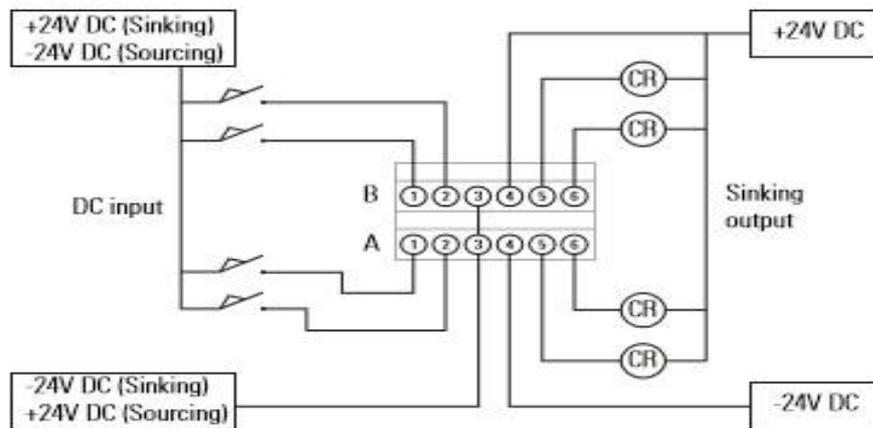
Pin Designations for 12-Pin Female Terminal Block Modules

Pin	2080-IQ4	2080-IQ40B4, 2080-IQ40V4	2080-OB4, 2080-OV4	2080-OW4I	2080-IF2	2080-IF4	2080-TC2	2080-RTD2
A1	I-02	I-02	Not used	COM3	COM	COM	CH0+	CH0+
A2	I-03	I-03	Not used	O-3	Not used	VI-2	CH0-	CH0-
A3	COM	COM	-24V DC	Not used	Not used	CI-2	CJC+	CH0L (Sense)
A4	COM	-24V DC	-24V DC	Not used	COM	COM	Not used	Not used
A5	Not used	O-02	O-02	Not used	Not used	VI-3	Not used	Not used
A6	Not used	O-03	O-03	Not used	Not used	CI-3	Not used	Not used
B1	I-00	I-00	Not used	COM0	VI-0	VI-0	CH1+	CH1+
B2	I-01	I-01	Not used	O-0	CI-0	CI-0	CH1-	CH1-
B3	COM	COM	+24V DC	COM1	COM	COM	CJC-	CH1L (Sense)
B4	COM	+24V DC	+24V DC	O-1	VI-1	VI-1	Not used	Not used
B5	Not used	O-00	O-00	COM2	CI-1	CI-1	Not used	Not used
B6	Not used	O-01	O-01	O-2	COM	COM	TH	Not used

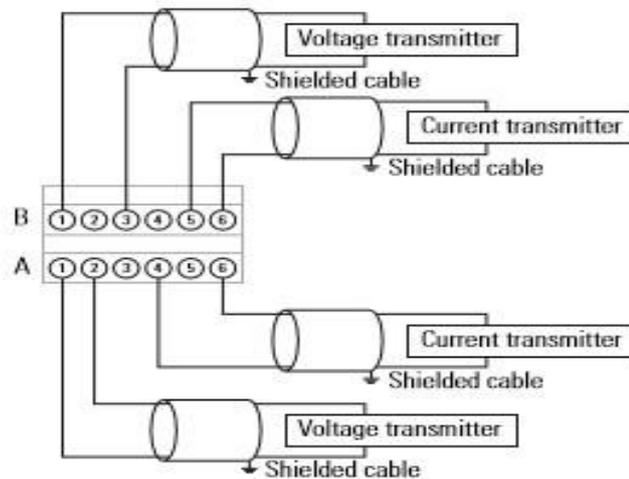
Example Wiring for 2080-IQ40B4

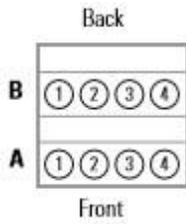


Example Wiring for 2080-IQ40V4



Example Wiring for 2080-IF4

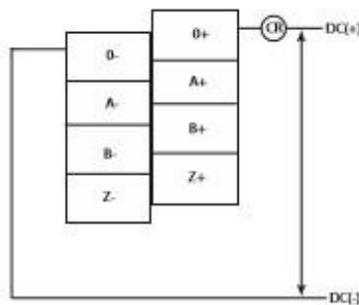




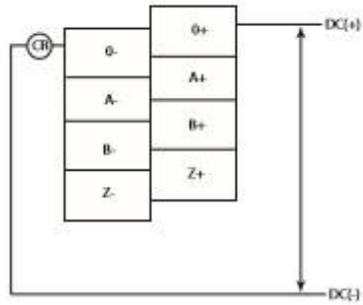
Pin Designations for 8-Pin Female Terminal Block Modules

Pin	2080-OF2	2080-SERIALISOL	2080-MOT-HSC ⁽¹⁾ (2)
A1	COM	RS485 B+	0-
A2	COM	GND	A-
A3	COM	RS232 RTS	B-
A4	COM	RS232 CTS	Z-
B1	VO-0	RS232 DCD	0+
B2	CO-0	RS232 RXD	A+
B3	VO-1	RS232 TXD	B+
B4	CO-1	RS485 A-	Z+

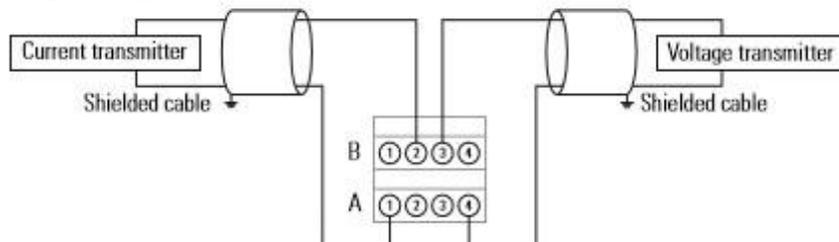
Sinking Output Wiring



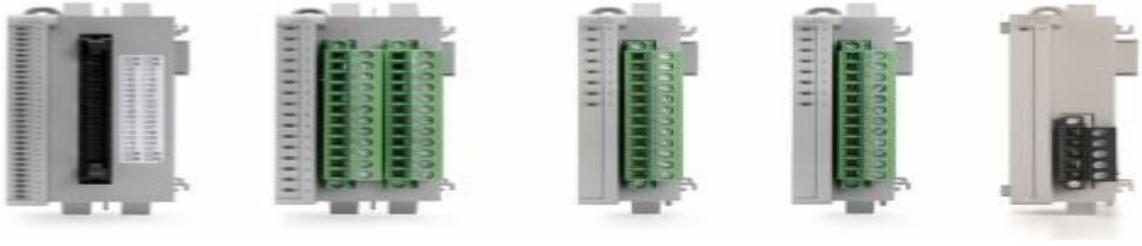
Sourcing Output Wiring



Example Wiring for 2080-OF2



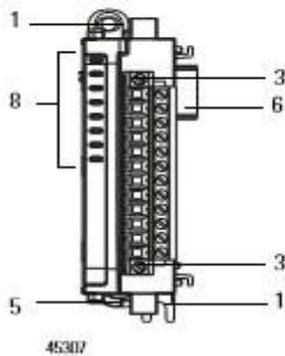
Módulos de Expansión



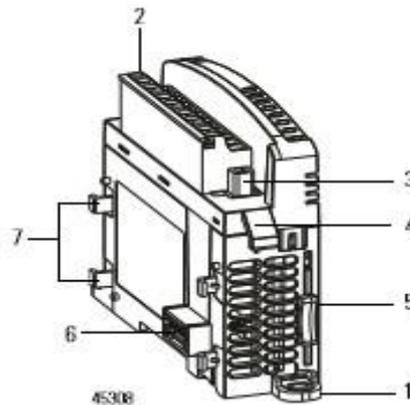
Micro800 Expansion I/O Modules

Catalog	Type	Description
2085-IA8	Discrete	8-point, 120V AC input
2085-IM8	Discrete	8-point, 240V AC input
2085-OA8	Discrete	8-point, 120/240V AC Triac Output
2085-IO16	Discrete	16-point, 12/24V Sink/Source Input
2085-IO32T	Discrete	32-point, 12/24V Sink/Source Input
2085-OV16	Discrete	16-point, 12/24V DC Sink Transistor Output
2085-OB16	Discrete	16-point, 12/24V DC Source Transistor Output
2085-OW8	Discrete	8-point, AC/DC Relay Output
2085-OW16	Discrete	16-point, AC/DC Relay Output
2085-IF4	Analog	4-channel, 14-bit isolated ⁽¹⁾ voltage/current input
2085-IF8	Analog	8-channel, 14-bit isolated ⁽¹⁾ voltage/current input
2085-OF4	Analog	4-channel, 12-bit isolated ⁽¹⁾ voltage/current output
2085-IRT4	Analog	4-channel, 16-bit isolated ⁽¹⁾ RTD and Thermocouple input module
2085-EP24VDC	Power supply	Supplies power for up to four expansion I/O modules ⁽²⁾
2085-ECR	Terminator	2085 bus terminator

2085-OW8 shown



Front view



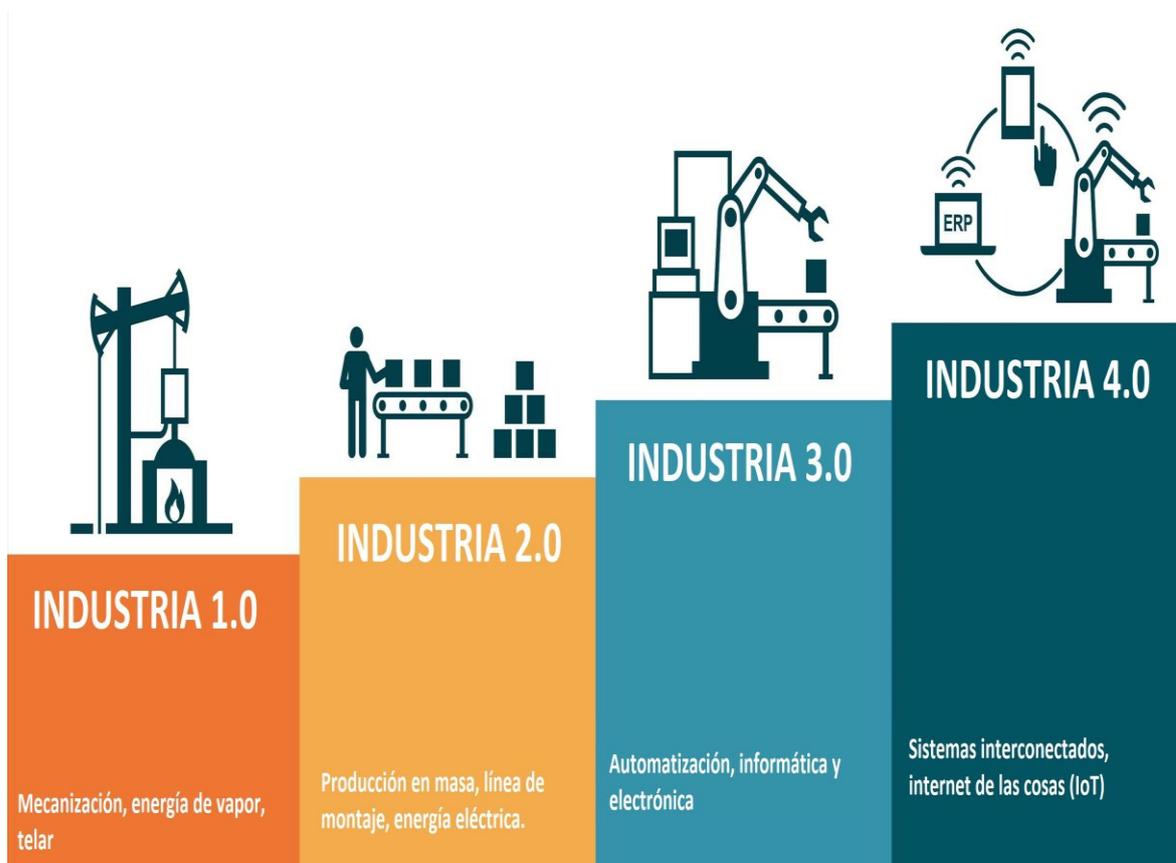
Right top view

Description		Description	
1	Mounting screw hole / mounting foot	6	Bus connector (male/female)
2	Removable Terminal Block (RTB) ⁽¹⁾	7	Latch hooks
3	RTB hold down screws	8	I/O status LED
4	Cable grip	9	DIN rail mounting latch
5	Module interconnect latch		

TEMA 3.-

Organización de la memoria y direccionamiento

Objetivo particular.- Al término del tema, el participante definirá a un nivel básico la forma de diagnosticar fallas en aplicaciones reales de PLC' s y mantenimiento preventivo.



Definiciones

Chasis

Es el receptáculo físico en donde se inserta el controlador programable y las tarjetas de I/O. Un chasis se denomina **local** cuando dentro de él reside el **PLC y remoto** si en él se encuentra un módulo de comunicación enlazado con el PLC del chasis local.

Punto digital de I/O.- Es la unidad mínima de identificación y ubicación el cual representa una entrada o salida física.

Slot.- ESPACIO PARA MODULO EN EL CHASIS

Identificación de un punto de I/O

Para identificar completamente un punto de I/O es necesario conocer la ubicación de este, lo cual nos da una dirección física y lógica.

Organización de la memoria

La estructura de la memoria en la familia de controladores programables Micro 800 de Allen Bradley está dividida en los siguientes archivos

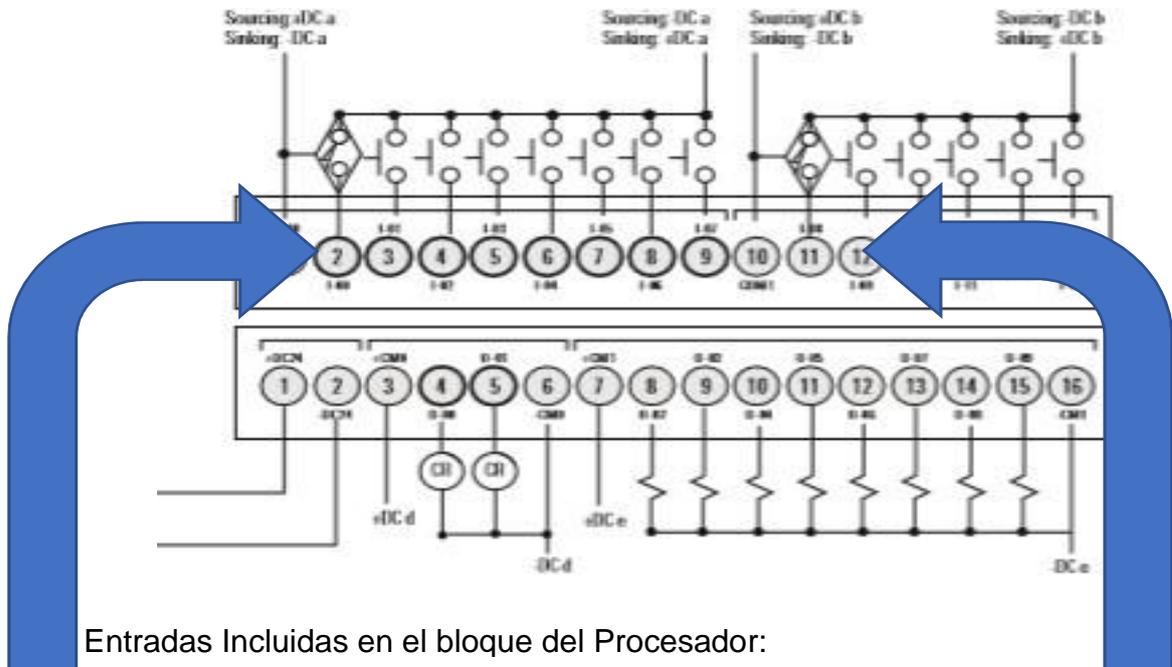
- ✓ Archivos de programas
- ✓ Archivos de datos
- ✓ Direccionamiento lógico

Memory Allocation for Micro800 Controllers

Attribute	10/16-poin (Micro830)t	20-point (Micro820)	24- and 48-points (Mico830, Micro850)	24-point (Micro870)
Program steps ⁽¹⁾	4 K	10 K	10 K	20 K
Data bytes	8 KB	20 KB	20 KB	40 KB

Direccionamiento de Entradas y Salidas

Las señales de campo denominadas entradas y salidas I/O se referencian de la siguiente forma:



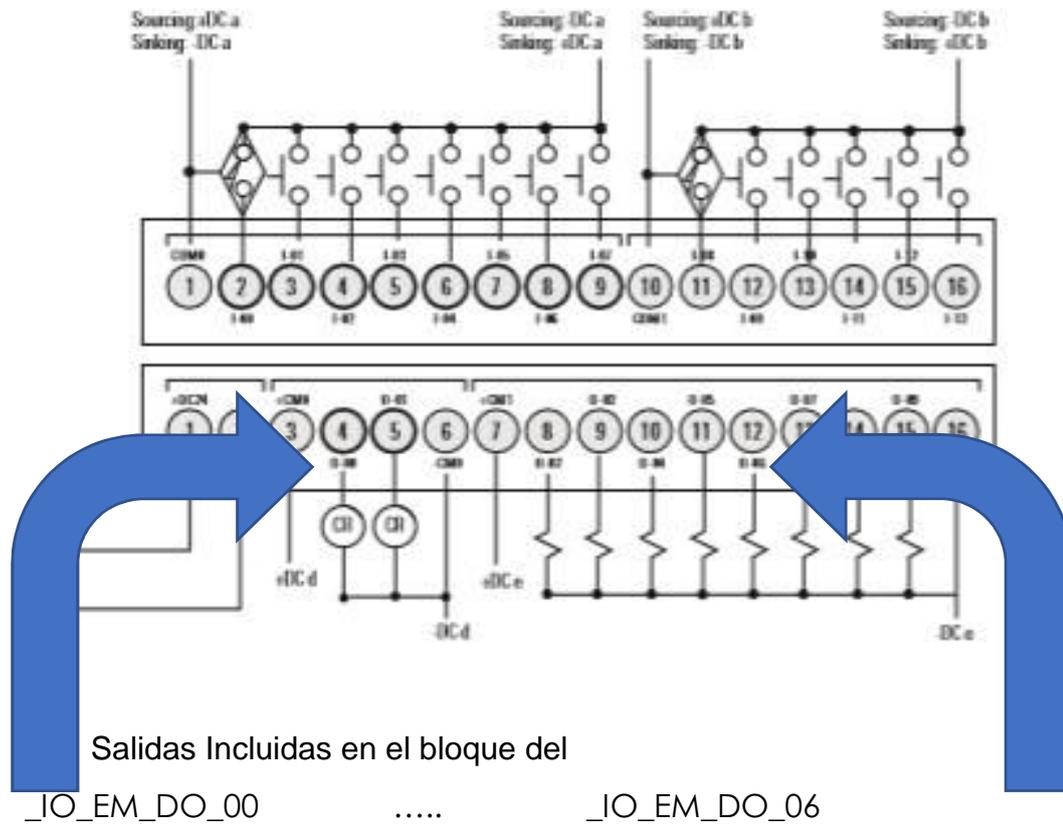
Entradas Incluidas en el bloque del Procesador:

ENTRADAS

`_IO_EM_DI_00` `_IO_EM_DI_09`

<code>_IO_EM_DI_00</code>		BOOL	▼			
<code>_IO_EM_DI_01</code>		BOOL	▼			
<code>_IO_EM_DI_02</code>		BOOL	▼			
<code>_IO_EM_DI_03</code>		BOOL	▼			
<code>_IO_EM_DI_04</code>		BOOL	▼			
<code>_IO_EM_DI_05</code>		BOOL	▼			
<code>_IO_EM_DI_06</code>		BOOL	▼			
<code>_IO_EM_DI_07</code>		BOOL	▼			
<code>_IO_EM_DI_08</code>		BOOL	▼			
<code>_IO_EM_DI_09</code>		BOOL	▼			
<code>_IO_EM_DI_10</code>		BOOL	▼			
<code>_IO_EM_DI_11</code>		BOOL	▼			
<code>_IO_EM_DI_12</code>		BOOL	▼			
<code>_IO_EM_DI_13</code>		BOOL	▼			
<code>_IO_EM_DI_14</code>		BOOL	▼			

SALIDAS

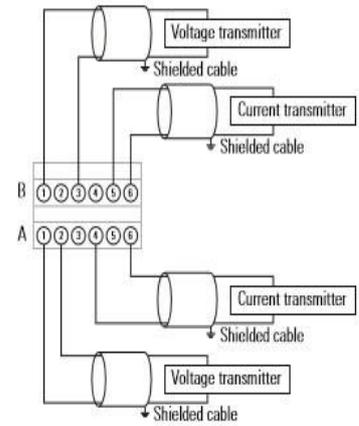


_IO_EM_DO_00		BOOL	▼					
_IO_EM_DO_01		BOOL	▼					
_IO_EM_DO_02		BOOL	▼					
_IO_EM_DO_03		BOOL	▼					
_IO_EM_DO_04		BOOL	▼					
_IO_EM_DO_05		BOOL	▼					
_IO_EM_DO_06		BOOL	▼					
_IO_EM_DO_07		BOOL	▼					
_IO_EM_DO_08		BOOL	▼					
_IO_EM_DO_09		BOOL	▼					
_IO_EM_DO_10		BOOL	▼					
_IO_EM_DO_11		BOOL	▼					
_IO_EM_DO_12		BOOL	▼					
_IO_EM_DO_13		BOOL	▼					
_IO_EM_DO_14		BOOL	▼					
_IO_EM_DO_15		BOOL	▼					
_IO_EM_DO_16		BOOL	▼					
_IO_EM_DO_17		BOOL	▼					
_IO_EM_DO_18		BOOL	▼					
_IO_EM_DO_19		BOOL	▼					

ENTRADAS / SALIDAS

Pin	2080-IQ4	2080-IQ40B4, 2080-IQ40V4	2080-OB4, 2080-OV4	2080-OW4I	2080-IF2	2080-IF4	2080-TC2	2080-RTD2
A1	I-02	I-02	Not used	COM3	COM	COM	CH0+	CH0+
A2	I-03	I-03	Not used	O-3	Not used	VI-2		
A3	COM	COM	-24V DC	Not used	Not used	CI-2		
A4	COM	-24V DC	-24V DC	Not used	COM	COM		
A5	Not used	O-02	O-02	Not used	Not used	VI-3		
A6	Not used	O-03	O-03	Not used	Not used	CI-3		
B1	I-00	I-00	Not used	COM0		VI-0		
B2	I-01	I-01	Not used	O-0	CI-0	CI-0		
B3	COM	COM	+24V DC	COM1	COM	COM		
B4	COM	+24V DC	+24V DC	O-1	VI-1	VI-1		
B5	Not used	O-00	O-00	COM2	CI-1	CI-1		
B6	Not used	O-01	O-01	O-2	COM	COM	TH	Not used

Wiring for 2080-IF4



Entradas Incluidas en el bloque del Procesador:

_IO_P1_AI_00

_IO_P1_AI_02

<u>_IO_P1_AI_00</u>		UINT	▼				
<u>_IO_P1_AI_01</u>		UINT	▼				
<u>_IO_P1_AI_02</u>		UINT	▼				
<u>_IO_P1_AI_03</u>		UINT	▼				
<u>_IO_P2_AO_00</u>		UINT	▼				
<u>_IO_P2_AO_01</u>		UINT	▼				
<u>_IO_P3_AO_00</u>		UINT	▼				
<u>_IO_P3_AO_01</u>		UINT	▼				

Variables internas

Los Equipos Micro 800 manejan datos o variables de diferentes tipos o características según el uso que se les quiere dar los más típicos son:

Tipo de datos ⁽¹⁾	Descripción
BOOL	Booleano lógico con valores TRUE(1) y FALSE(0) (usa hasta 8 bits de memoria)
SINT	Valor entero de 8 bits con signo
INT	Valor entero de 16 bits con signo
DINT	Valor entero de 32 bits con signo
LINT ⁽²⁾	Valor entero de 64 bits con signo
USINT	Valor entero de 8 bits sin signo
UINT	Valor entero de 16 bits sin signo
UDINT	Valor entero de 32 bits sin signo
ULINT ⁽²⁾	Valor entero de 64 bits sin signo
REAL	Valor de punto flotante (coma flotante) de 32 bits
LREAL ⁽²⁾	Valor de punto flotante (coma flotante) de 64 bits
STRING	Cadena de caracteres (1 byte por carácter)
DATE ⁽³⁾	Valor entero de 32 bits sin signo
TIME ⁽³⁾	Valor entero de 32 bits sin signo

__SYSVA_SUSPEND_ID	UINT	0	Last Suspend ID
__SYSVA_TCYWDG	UDINT	2000	Software Watchdog
__SYSVA_MAJ_ERR_HALT	BOOL	FALSE	Major Error Halted status
__SYSVA_ABORT_CYCLE	BOOL	FALSE	Aborting Cycle
__SYSVA_FIRST_SCAN	BOOL	TRUE	First scan bit
__SYSVA_USER_DATA_LOST	BOOL	FALSE	User data lost
__SYSVA_POWERUP_BIT	DINT	TRUE	Power-up bit
__SYSVA_PROJ_INCOMPL	INT	0	Project Incomplete
*			

Archivo del Programa

Un ciclo o SCAN Micro800 consta de lectura de entradas, ejecución de programas en orden secuencial, actualización de salidas, y realización de mantenimiento interno (registro de datos, recetas, comunicaciones).

Los nombres de los programas deben comenzar con una letra o con un carácter de subrayado, seguidos de hasta 127 letras, dígitos o caracteres de subrayado individuales. Use lenguajes de programación como lógica de escalera, diagramas de bloques de funciones y texto estructurado.

Es posible incluir hasta 256 programas en un proyecto, de acuerdo con la memoria disponible del controlador. De manera predeterminada, los programas son cíclicos (se ejecutan una vez por ciclo o scan). Cada vez que se añade un nuevo programa a un proyecto, se le asigna el siguiente número de orden consecutivo. Al iniciar Project Organizer en Connected Components Work bench, este muestra los iconos del programa en este orden. Se puede ver y modificar el número de orden de un programa mediante las propiedades de este. Sin embargo, el organizador de proyectos no muestra el nuevo orden, hasta que se vuelve a abrir el proyecto la próxima vez.

El controlador Micro800 acepta saltos dentro de un programa. Para llamar a una subrutina de código dentro de un programa, encapsule dicho código como un bloque de funciones definidas por el usuario (UDFB). Una UDF es semejante a una subrutina tradicional y usa menos memoria que un UDFB; en cambio, un UDFB puede tener múltiples instancias. Si bien un UDFB puede ejecutarse dentro de otro UDFB, se acepta una profundidad máxima de anidamiento de cinco. Se produce un error de compilación si se supera ese número. Esto se aplica también a las UDF.

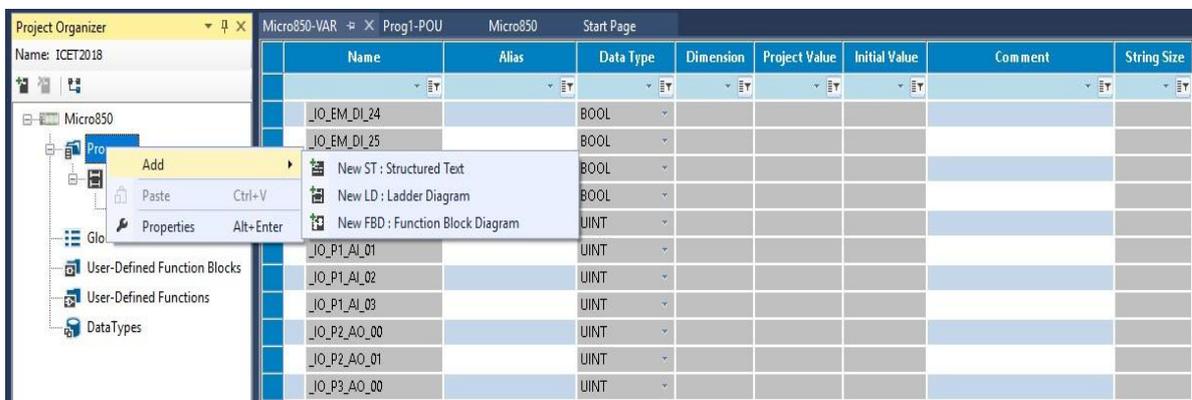
De manera alternativa, se puede asignar un programa a una interrupción disponible y ejecutarlo solo si se dispara la interrupción. Un programa asignado a la rutina de fallo de usuario se ejecuta una vez antes de que el controlador entre en el modo de fallo.

Además de la rutina de fallo de usuario, los controladores micro800 también aceptan dos interrupciones temporizadas seleccionables (STI). Las STI ejecutan programas asignados una vez durante cada intervalo de punto de ajuste (1...65,535 ms).

Las variables globales del sistema asociadas a los ciclos/scan son las siguientes:

- ✓ SYSVA_CYCLECNT – Contador de ciclos
- ✓ SYSVA_TCYCURRENT – Tiempo de ciclo actual
- ✓ SYSVA_TCYMAXIMUM – Máximo tiempo de ciclo a partir del último arranque.

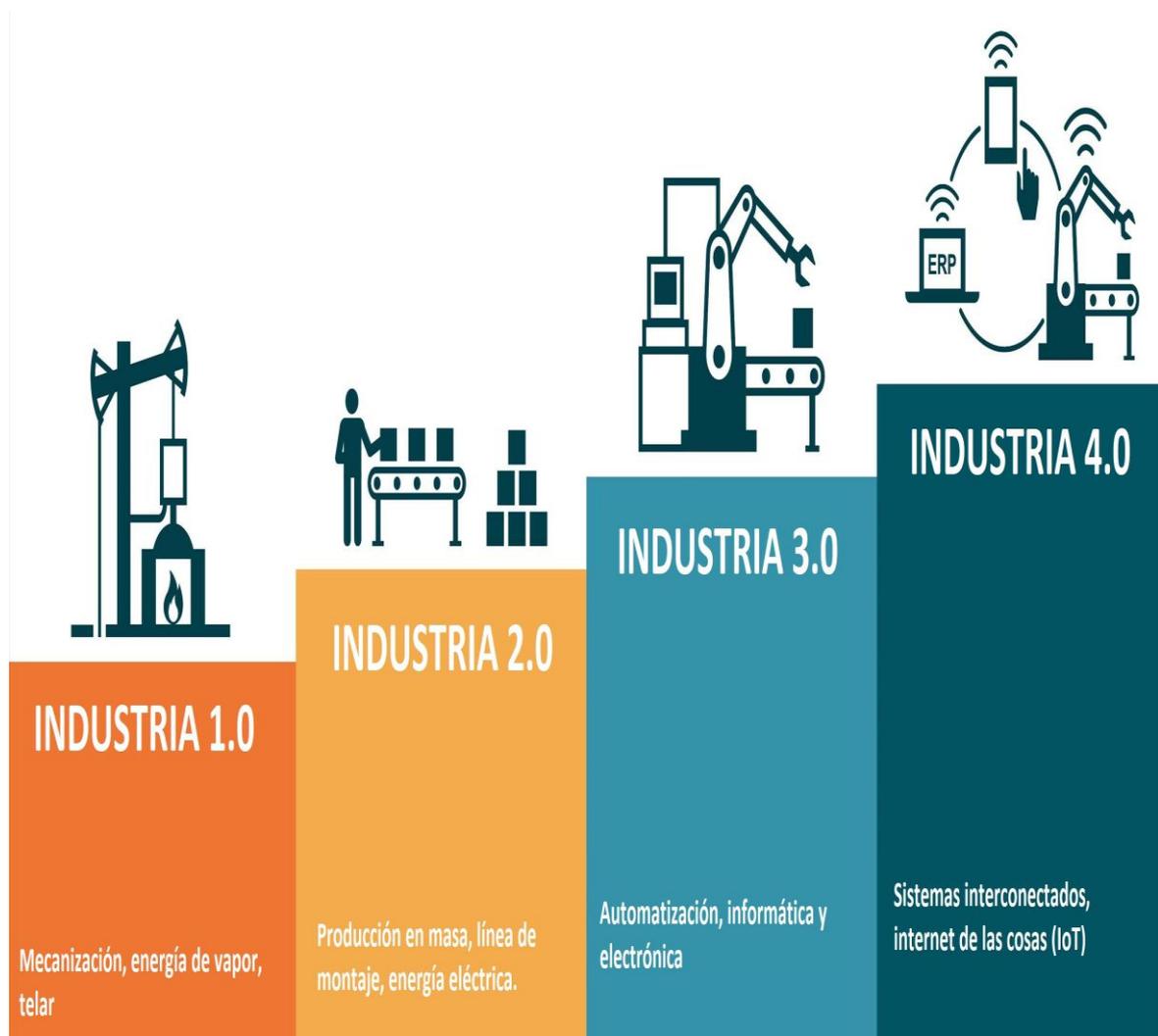
Tipos de Programación



Diagramas de escalera	LD: Ladder Diagram
Texto estructurado	ST: Structured Text
Diagramas de Bloques de función	FBD: Function Block Diagram

Tema 4.- Diagramas de escalera e instrucciones tipo relé

Objetivo particular.- Al término del tema, el participante repetirá la forma adecuada de instalar y configurar un diagrama de lógica de escalera a un nivel básico.



Un diagrama de lógica de escalera (LD) es una representación gráfica de ecuaciones booleanas que combina contactos (argumentos de entrada) con bobinas (resultados de salida). Mediante símbolos gráficos en un gráfico de programas (organizado como un diagrama de cableado de lógica de escalera de relés), el lenguaje de LD describe pruebas y modificaciones de datos booleanos.

Los símbolos gráficos de LD se organizan dentro del gráfico como un diagrama de contactos eléctricos. El término "escalera" está relacionado con el concepto de peldaños conectados a raíles de potencia verticales a ambos extremos, en los que cada peldaño representa un circuito individual.

El editor de idioma para un programa de diagrama de lógica de escalera (LD), donde se desarrolla una unidad organizativa de programa (POU) de LD.

Connected Components Work bench ofrece un editor de idioma de LD que solo es compatible con los elementos e instrucciones suministrados con el software de Connected Components Work bench.

La siguiente ilustración muestra las áreas principales del entorno de desarrollo de programas de LD.



	Nombre	Descripción
	Barra de herramientas de instrucción	Seleccione rápidamente un elemento de instrucción y colóquelo en el editor gráfico de LD, o bien haga clic para añadirlo al editor de texto de LD.
	Editor de texto de LD	Edita la lógica utilizando reglas mnemotécnicas de instrucciones ASCII.
	Editor gráfico de LD	Edita la lógica utilizando elementos de instrucción gráficos.
	Cuadro de herramientas de LD	Agrega elementos al editor gráfico de LD.

Elementos de diagrama de escalera (LD)

Peldaño

Las líneas son componentes gráficos de un programa de diagrama de lógica de escalera (LD) que representan un grupo de elementos de circuito que llevan a la activación de una bobina. Utilice etiquetas para identificar líneas del diagrama. Las etiquetas controlan, junto con los saltos, la ejecución de un diagrama.

Los comentarios son texto de formato libre que puede agregar encima de la línea como documentación.

Agregar una línea a un programa de LD

Las líneas son componentes gráficos de un programa de diagrama de lógica de escalera (LD) que representan un grupo de elementos de circuito que llevan a la activación de una bobina.

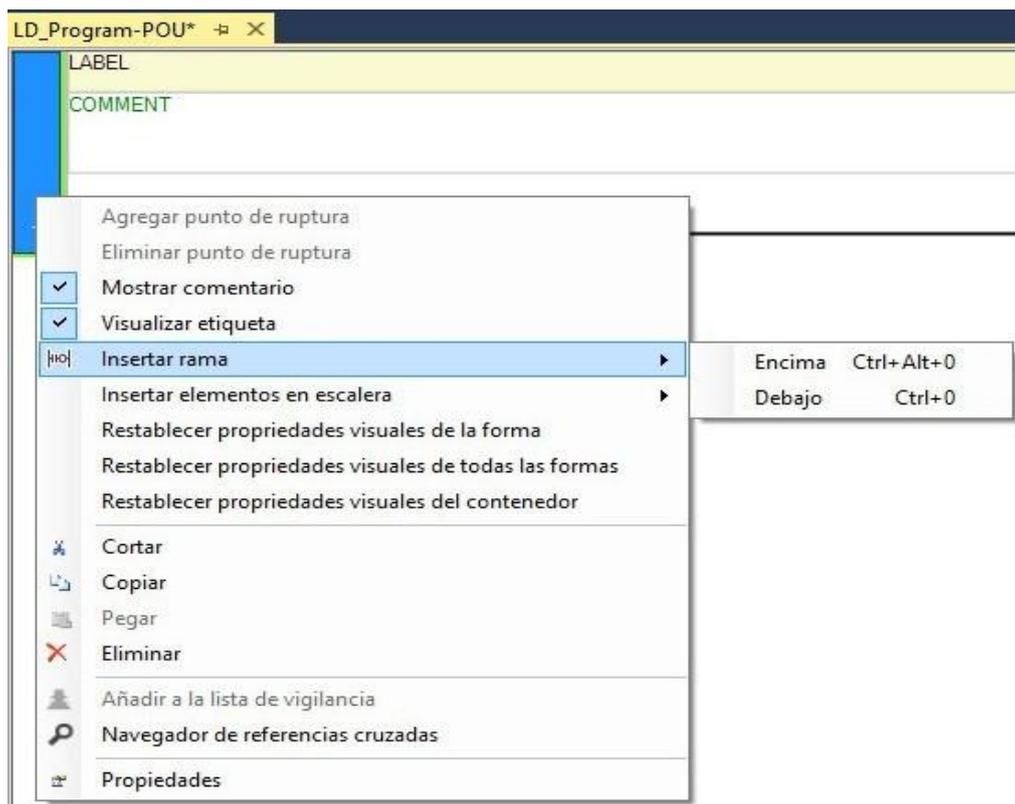
En Connected Components Work bench puede agregar una línea a un programa de Diagrama de lógica de escalera (LD) desde el:

- Editor de idioma de diagrama de idioma (LD)
- Editor multi-idioma ubicado en el menú Herramientas
- Cuadro de herramientas de LD

Para agregar un elemento de línea a un programa de Diagrama de lógica de escalera:

1.- En el editor de idioma de LD, haga clic con el botón secundario en una línea existente y:

- ✓ Haga clic en **Copiar** y, a continuación, haga clic en **Pegar** para insertar una copia de la línea al editor de idioma.
- ✓ Haga clic en **Insertar línea** y a continuación:
- ✓ Haga clic en **Por encima de** para agregar la línea por encima de la línea seleccionada.
- ✓ Haga clic en **Por debajo de** para agregar la línea por debajo de la línea seleccionada.



Seleccione una línea o un elemento en el editor de idioma de LD y pulse:
CTRL+ALT+0: para agregar la línea por encima de la línea seleccionada.

CTRL+0: para agregar la línea por debajo de la línea seleccionada.

Seleccione una línea o un elemento en el editor de idioma de LD y pulse:

Haga clic en Herramientas > Editor multi-idioma

Insertar línea debajo para agregar la línea por debajo de la línea seleccionada.

Haga clic en Herramientas > Editor multi- idioma

Insertar línea arriba para agregar la llamada por encima de la línea seleccionada.

2.- (opcional) Abra el cuadro de herramientas de LD para mostrar el elemento de línea.

Para insertar una línea por debajo de una línea existente, seleccione la línea en el editor de idioma de LD y, a continuación, haga doble clic en Línea en el cuadro de herramientas de LD.

Si un elemento no está seleccionado antes de hacer doble clic en Línea en el cuadro de herramientas de LD, se inserta la línea por debajo de la última línea en el editor de idioma de LD.

Seleccione Línea y, a continuación, arrastre el elemento al editor de idioma de LD.

Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para agregar el elemento.

Etiquetas de línea

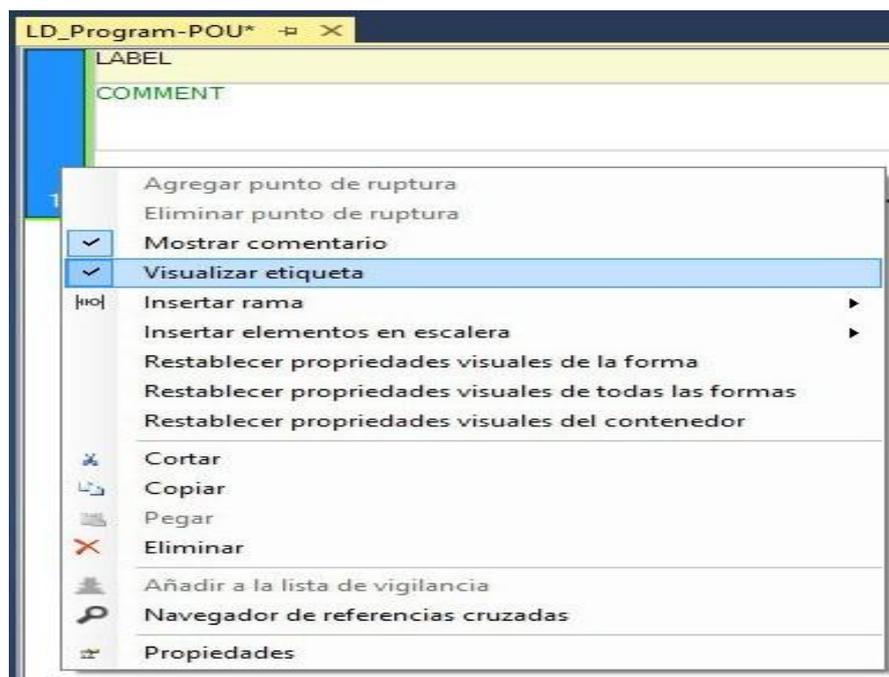
Los comentarios son adiciones opcionales para cada línea del editor de idioma (LD) del diagrama de lógica de escalera.

Las etiquetas pueden tener un número ilimitado de caracteres, comenzando por una letra o un carácter de subrayado, seguidos de letras, números y caracteres de subrayado. Las etiquetas no pueden tener espacios ni caracteres especiales (por ejemplo, '+', '-', o '\').

Para agregar una línea en el editor de idioma de LD:

1. Para agregar una etiqueta a una línea, haga clic en la línea para abrir el menú del editor de idioma de LD y, a continuación, seleccione **Etiqueta de pantalla**.

La línea se actualiza para incluir la etiqueta y el menú del editor de idioma de la LD muestra una marca de verificación junto a **Etiqueta de pantalla**.



- 1.- Seleccione la **Etiqueta** y escriba una descripción.
- 2.- (opcional) Para eliminar la etiqueta, haga clic en **Etiqueta de pantalla** en el menú de editor de idioma de LD.

Comentarios de Peldaño

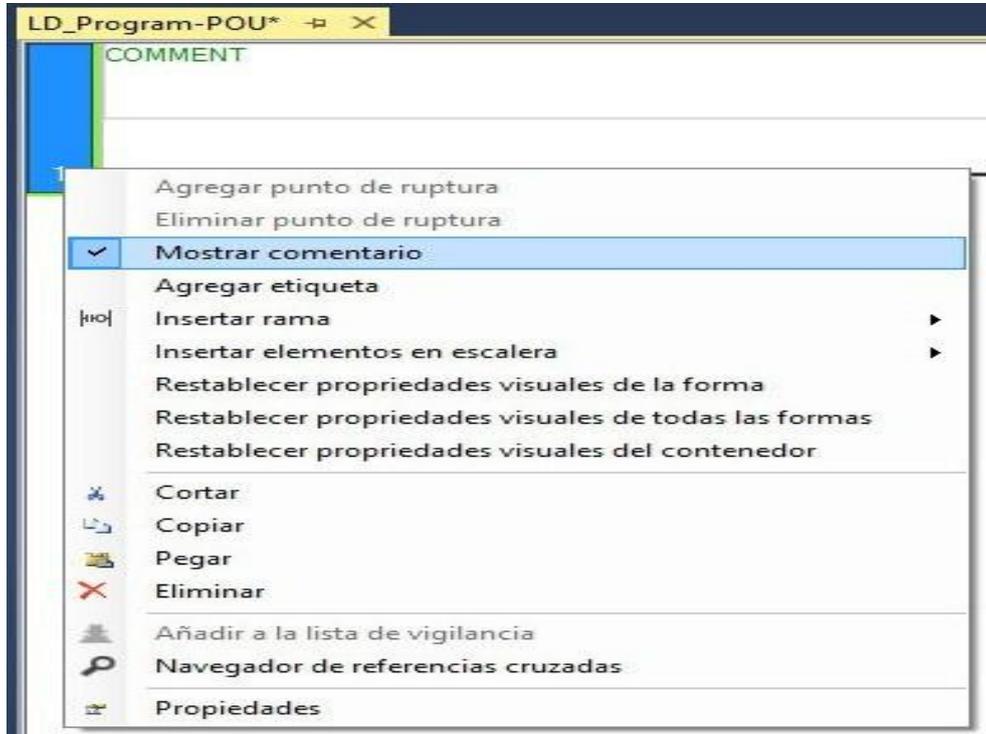
Los comentarios son adiciones opcionales para cada línea del editor de idioma (LD) del diagrama de lógica de escalera. De forma predeterminada, se incluye un comentario cuando se agrega un elemento de línea en el editor de idioma de LD.

Los comentarios:

- ✓ Se escriben en el espacio situado encima de la línea.
- ✓ Se guardan en formato de texto enriquecido.
- ✓ Se almacenan en el controlador.

Para agregar o eliminar un comentario para una línea en el editor de idioma de LD:

1. Para eliminar un comentario, haga clic con el botón secundario en la línea para abrir el menú del editor de idioma de LD.



En el menú de editor de idioma de LD, haga clic en **Mostrar comentario**. Se elimina el comentario de la línea, y se elimina la marca de verificación ubicada junto a **Mostrar comentario** en el menú del editor de idioma de LD.

(opcional) Para agregar un comentario a una línea, haga clic en **Mostrar comentario** en el menú del editor de idioma de LD.

Ramales

Los ramales crean enrutamientos alternativos para las conexiones. Puede agregar ramas de paralelo a los elementos de una línea con el editor de idioma de diagrama de lógica de escalera (LD).



Agregar un ramal a un programa de LD

Los ramales son componentes gráficos de los programas de diagrama de lógica de escalera (LD) que crean rutas alternativas para las conexiones y pueden incluir bifurcaciones en paralelo.

En Connected Components Work bench puede agregar un ramal a un programa de diagrama de lógica de escalera (LD) desde el:

- ✓ Editor de idioma de diagrama de lógica de escalera(LD)
- ✓ Editor multi-idioma ubicado en el menú Herramientas
- ✓ Cuadro de herramientas de LD

Para agregar un ramal a un programa de diagrama de lógica de escalera:

1. En el editor de idioma de LD, compruebe que el programa de LD tenga una línea definida para la bifurcación. A continuación, realice una de las siguientes acciones:

Haga clic en una línea o un elemento en el editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Bifurcación.

Si se abre el Selector de variables seleccione una variable o haga clic en Aceptar sin seleccionar una variable para agregar el elemento de Bifurcación.

- ✓ Seleccione una línea o un elemento en el editor de idioma de LD y pulse:

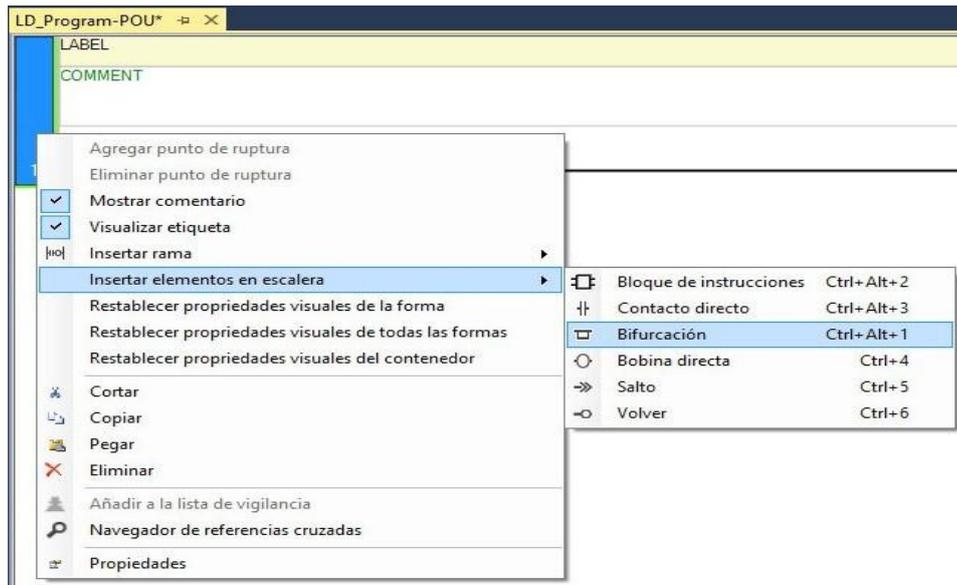
CTRL+ALT+1 para agregar la bifurcación a la izquierda del elemento o la línea seleccionados.

CTRL+1 para agregar la bifurcación a la derecha del elemento seleccionado.

- ✓ Seleccione una línea o un elemento en el editor de idioma de LD ya continuación:

Haga clic en Herramientas > Editor multi-idioma> Insertar bifurcación antes para agregar la bifurcación a la izquierda del elemento seleccionado.

Haga clic en Herramientas > Editor multi-idioma> Insertar bifurcación después para agregar la bifurcación a la derecha del elemento seleccionado.



Abra el cuadro de herramientas de LD para mostrar el elemento de bifurcación y, a continuación:

Haga doble clic en el elemento de bifurcación para agregarlo al editor de idioma de LD.

Arrastre el elemento de bifurcación al editor de idioma de LD y colóquelo en la línea.

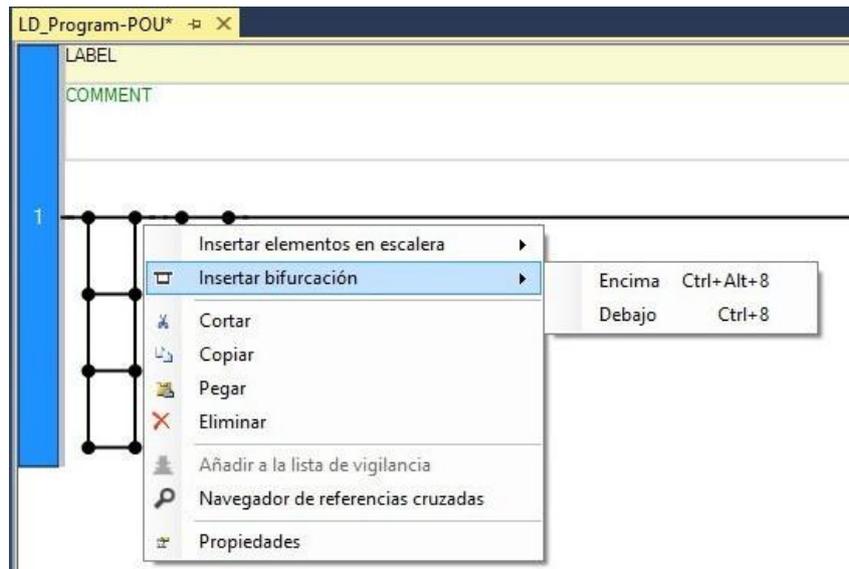
Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para agregar el elemento.

1. (opcional) Para insertar una bifurcación en paralelo:

a) En el editor de idioma de LD, haga clic con el botón secundario en la bifurcación para abrir el menú del editor de idioma de LD.

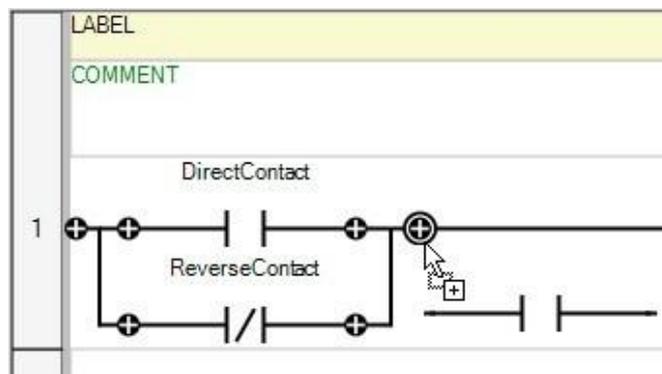
b) En el menú del editor de idioma de LD, seleccione Insertar bifurcación y, a continuación, haga clic en:

Arriba para agregar una bifurcación por encima de la bifurcación seleccionada. Debajo para agregar la bifurcación por debajo de la bifurcación seleccionada.



1. (opcional) Para mover un elemento de bifurcación a otra ubicación en un programa de LD, seleccione el elemento y arrástrelo al lugar deseado.

Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para insertar el elemento en la ubicación de destino.



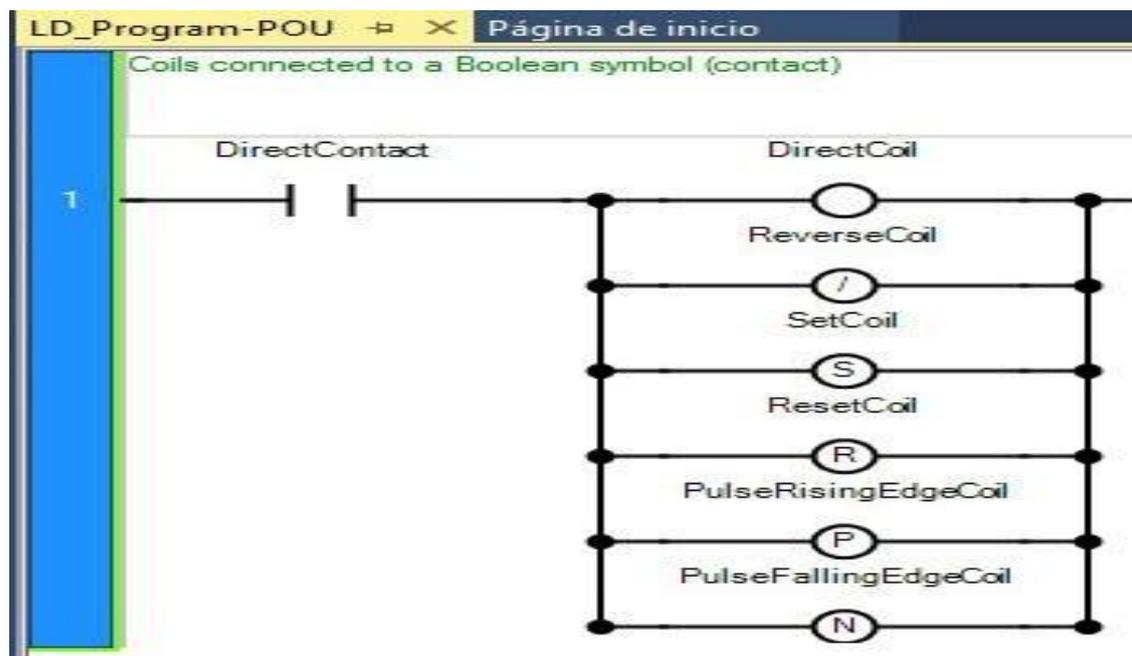
Bloque de instrucciones (LD)

Un elemento de bloque de instrucción de diagrama de lógica de escalera (LD) es un elemento funcional de un programa de LD que cumple con IEC61131-3 y que puede ser un bloque de funciones, una función, un bloque de funciones definidas por el usuario, una función definida por el usuario o un operador.

Bobina

Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna. En programas LD, una bobina representa una acción. Una bobina debe estar conectada a la izquierda de un símbolo booleano, como, por ejemplo, un contacto, o a la salida booleana de un bloque de instrucción. Solo se pueden agregar bobinas a una línea definida en el editor de idioma de LD. La definición de bobina puede modificarse una vez que la bobina se haya agregado a la línea.

El siguiente ejemplo muestra los tipos de elemento de bobina disponibles para los programas de diagrama de lógica de escalera.



Agregar elementos de bobina

Las bobinas son componentes gráficos de los programas de diagrama de lógica de escalera (LD) que representan la asignación de una salida o de una variable interna.

En Connected Components Work bench puede agregar una bobina a un programa de diagrama de lógica de escalera (LD) desde el:

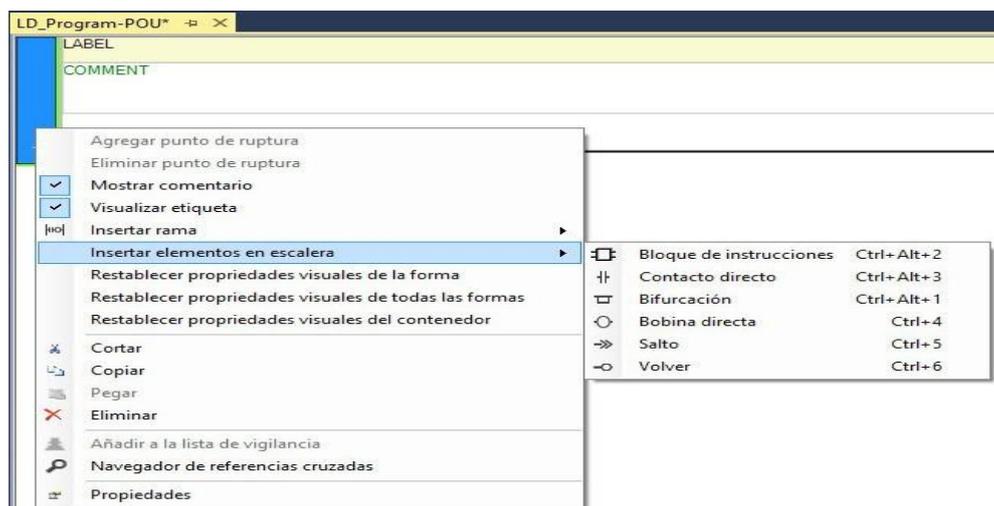
- ✓ Editor de idioma de diagrama de lógica de escalera(LD)
- ✓ Editor multi-idioma ubicado en el menú Herramientas
- ✓ Cuadro de herramientas de LD

Para agregar un elemento de bobina a un programa de diagrama de lógica de escalera:

1. En el editor de idioma de LD, compruebe que el programa de LD tenga una línea definida para la bobina. A continuación, realice una de las siguientes acciones:

- ✓ Haga clic con el botón secundario en una línea o un elemento en el editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Bobina directa.

Si se abre el Selector de variables seleccione una variable o haga clic en Aceptar sin seleccionar una variable para agregar el elemento de Bobina



Seleccione una línea o un elemento en el editor de idioma de LD y, a continuación, pulse CTRL+4 para agregar la Bobina directa a la derecha de la línea.

Seleccione una línea o un elemento en el editor de idioma de LD y, a continuación, haga clic en Herramientas > Editor multi- idioma > Insertar bobina para agregar la bobina a la derecha de la línea.

Abra el cuadro de herramientas de LD para mostrar los elementos de bobina (Bobina directa, Bobina inversa, Bobina Set, Bobina Reset, Bobina de Flanco ascendente del pulso, bobina de Flanco descendente del pulso)y:

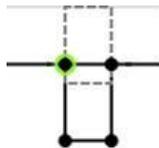
Haga doble clic en el elemento de bobina para agregarlo al editor de idioma de LD.

Arrastre el elemento de bobina al editor de idioma de LD y colóquelo en la línea.

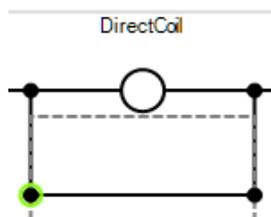
Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para agregar el elemento.

1. (opcional) Para insertar una bobina en paralelo,

En el editor de idioma de LD, compruebe que el programa de LD tenga una bifurcación definida y, a continuación, haga clic con el botón secundario en el nivel superior de la bifurcación para abrir el menú de editor de idioma de LD.

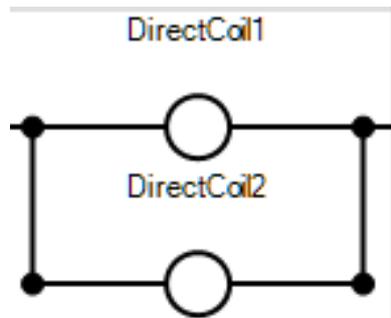


En el menú de editor de idioma de LD, selección el insertar elementos de lógica de escalera y, a continuación, haga clic en Bobina directa. El elemento se inserta en el nivel superior de la bifurcación.



Haga clic con el botón secundario en el nivel inferior de la bifurcación para abrir el menú de editor de idioma de LD.

En el menú de editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Bobina directa. El elemento se inserta en el nivel inferior de la bifurcación.



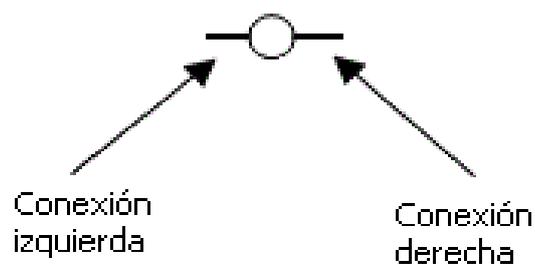
(opcional) Para cambiar el tipo de bobina, en el editor de idioma de LD seleccione la bobina y, a continuación, pulse la barra espaciadora hasta que aparezca el tipo de bobina en el editor de idioma.

Cada vez que pulse la barra espaciadora, el tipo de bobina cambia de directa, a inversa, ajuste, reset, flanco ascendente del pulso y flanco descendente del pulso.

Bobina directa

Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna.

Las bobinas directas admiten una salida booleana de una línea de conexión de estado booleano.



La variable asociada se asigna con el estado booleano de la conexión izquierda. El estado de la conexión izquierda se propaga a la conexión derecha. La conexión derecha debe estar conectada al raíl de potencia vertical derecho (a menos que tenga bobinas paralelas, donde solo la bobina superior debe estar conectada al raíl de potencia vertical derecho).

La variable booleana asociada debe ser una salida o estar definida por el usuario.

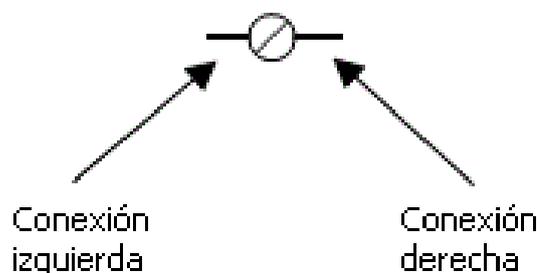
Ejemplo de bobina directa



Bobina inversa

Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna.

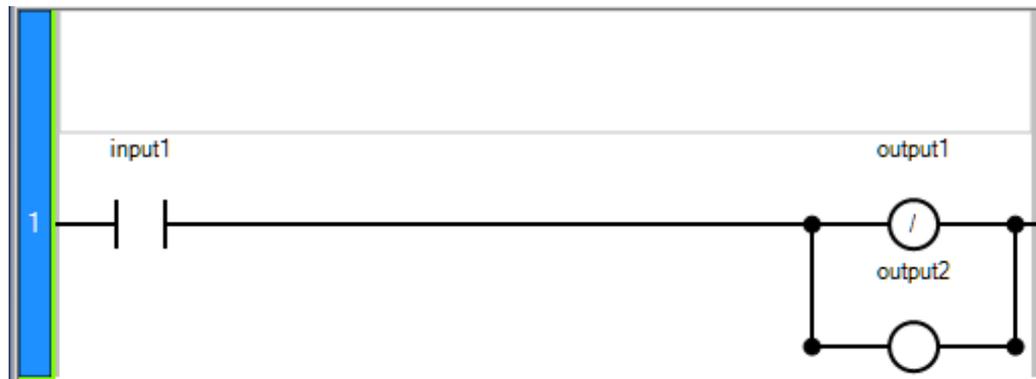
Los elementos de bobinas inversa admiten una salida booleana de acuerdo con la negación booleana de un estado de línea de conexión.



La variable asociada se asigna con la negación booleana del estado de la conexión izquierda. El estado de la conexión izquierda se propaga a la conexión derecha. La conexión derecha debe estar conectada al raíl de potencia vertical derecho (a menos que tenga bobinas paralelas, donde solo la bobina superior debe estar conectada al raíl de potencia vertical derecho).

La variable booleana asociada debe ser una salida o estar definida por el usuario.

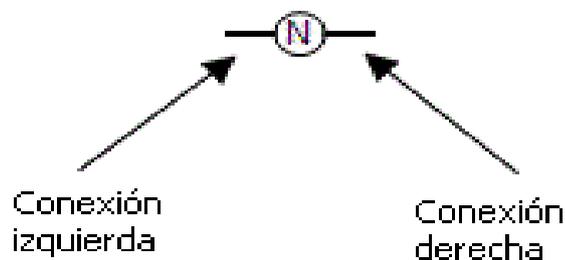
Ejemplo de bobina inversa



Bobina de extremo de decremento de pulso

Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna.

Las bobinas de contacto de flanco descendente (o negativo) admiten una salida booleana de una línea de conexión de estado booleano.



La variable asociada se define a Cierto si el estado booleano de la conexión izquierda cambia de Cierto a Falso. La variable de salida se restablece a Falso en el resto de los casos. El estado de la conexión izquierda se propaga a la conexión derecha. La conexión derecha debe estar conectada al raíl de potencia vertical derecho (a menos que tenga bobinas paralelas, donde solo la bobina superior debe estar conectada al raíl de potencia vertical derecho).

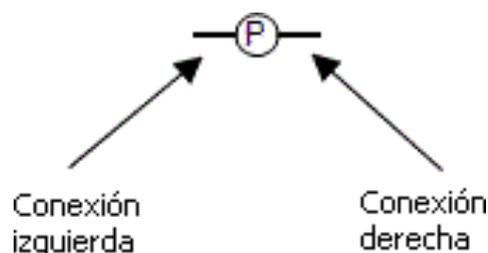
La variable booleana asociada debe ser una salida o estar definida por el usuario. Ejemplo de bobina de flanco descendente de pulso.



Bobina de extremo de incremento de pulsos

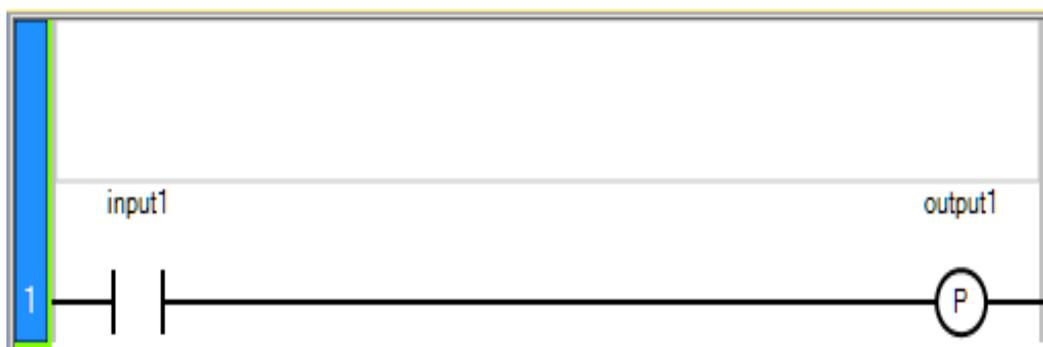
Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna.

Las bobinas de contacto de flanco ascendente (o positivo) admiten una salida booleana de una línea de conexión de estado booleano.



La variable asociada se define a Cierto si el estado booleano de la conexión izquierda cambia de Falso a Cierto. La variable de salida se restablece a Falso en el resto de los casos. El estado de la conexión izquierda se propaga a la conexión derecha. La conexión derecha debe estar conectada al raíl de potencia vertical derecho (a menos que tenga bobinas paralelas, donde solo la bobina superior debe estar conectada al raíl de potencia vertical derecho).

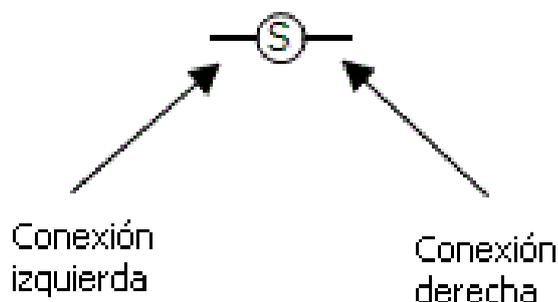
La variable booleana asociada debe ser de salida o definida por el usuario. Ejemplo de bobina de flanco ascendente de pulso



Bobina de ajuste

Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna. En programas LD, una bobina representa una acción.

Las bobinas de ajuste habilitan una salida booleana de una línea de conexión de estado booleano.



La variable asociada se define a Cierto si el estado booleano de la conexión izquierda es Cierto. La variable de salida mantiene este valor hasta que se realiza una orden inversa mediante una bobina de restablecimiento. El estado de la conexión izquierda se propaga a la conexión derecha. La conexión derecha debe estar conectada al raíl de potencia vertical derecho (a menos que tenga bobinas paralelas, donde solo la bobina superior debe estar conectada al raíl de potencia vertical derecho).

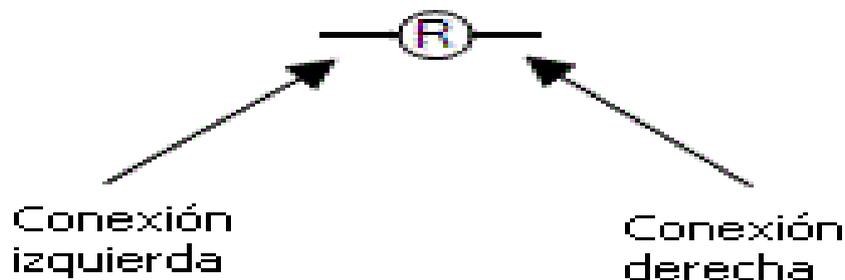
La variable booleana asociada debe ser una salida o estar definida por el usuario. Ejemplo de establecimiento de bobina



Bobina de restablecimiento

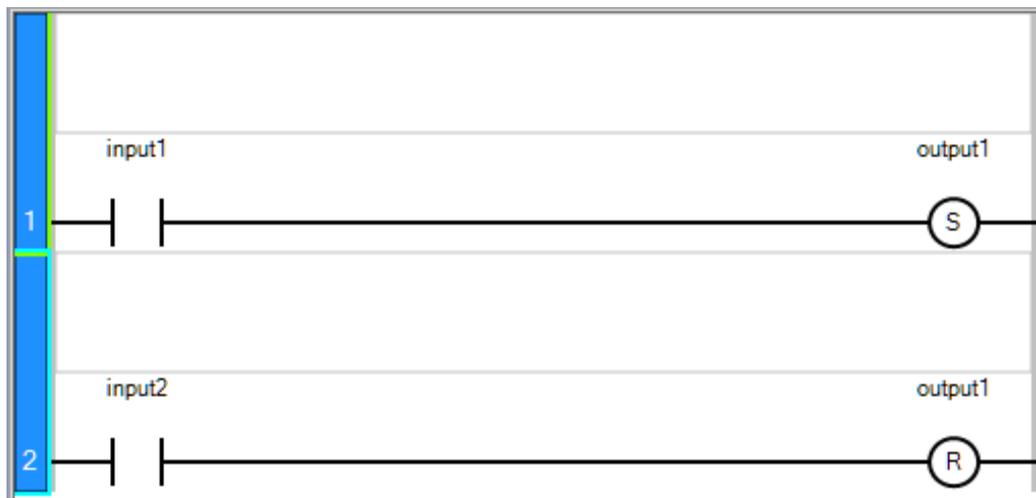
Las bobinas son componentes gráficos de un diagrama de lógica de escalera (LD) que representan la asignación de una salida o una variable interna.

Las bobinas de restablecimiento admiten una salida booleana de una línea de conexión de estado booleano.



La variable asociada se restablece a Falso si el estado booleano de la conexión izquierda es Cierto. La variable de salida mantiene este valor hasta que se realiza una orden inversa mediante una bobina de restablecimiento. El estado de la conexión izquierda se propaga a la conexión derecha. La conexión derecha debe estar conectada al raíl de potencia vertical derecho (a menos que tenga bobinas paralelas, donde solo la bobina superior debe estar conectada al raíl de potencia vertical derecho).

La variable booleana asociada debe ser de salida o definida por el usuario. Ejemplo de bobina Reset

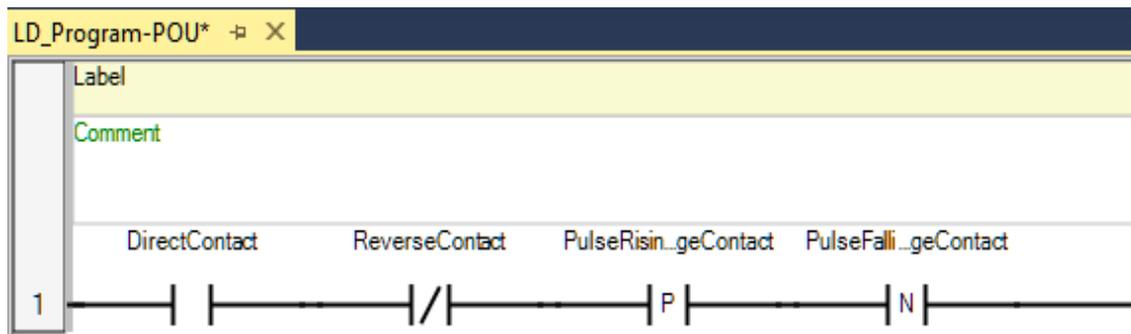


Contacto

Los contactos son componentes gráficos de programas de diagrama de lógica de escalera (LD). En función del tipo, un contacto representa el valor o la función de una entrada o variable interna. Solo se pueden agregar contactos a un peldaño definido en el editor de lenguaje de LD. Una vez agregado un contacto, puede modificarse su definición.

El ejemplo siguiente muestra los tipos de elemento de contacto disponibles para los programas de diagrama de lógica de escalera.

Ejemplo: Contactos



Agregar un contacto a un programa de LD

Los contactos son componentes gráficos de un programa de diagrama de lógica de escalera (LD). En función del tipo, un contacto representa el valor o la función de una entrada o variable interna. Solo se pueden agregar contactos a un peldaño definido en el editor de lenguaje de LD.

En Connected Components Work bench puede agregar un contacto a un programa de diagrama de lógica de escalera (LD) en él:

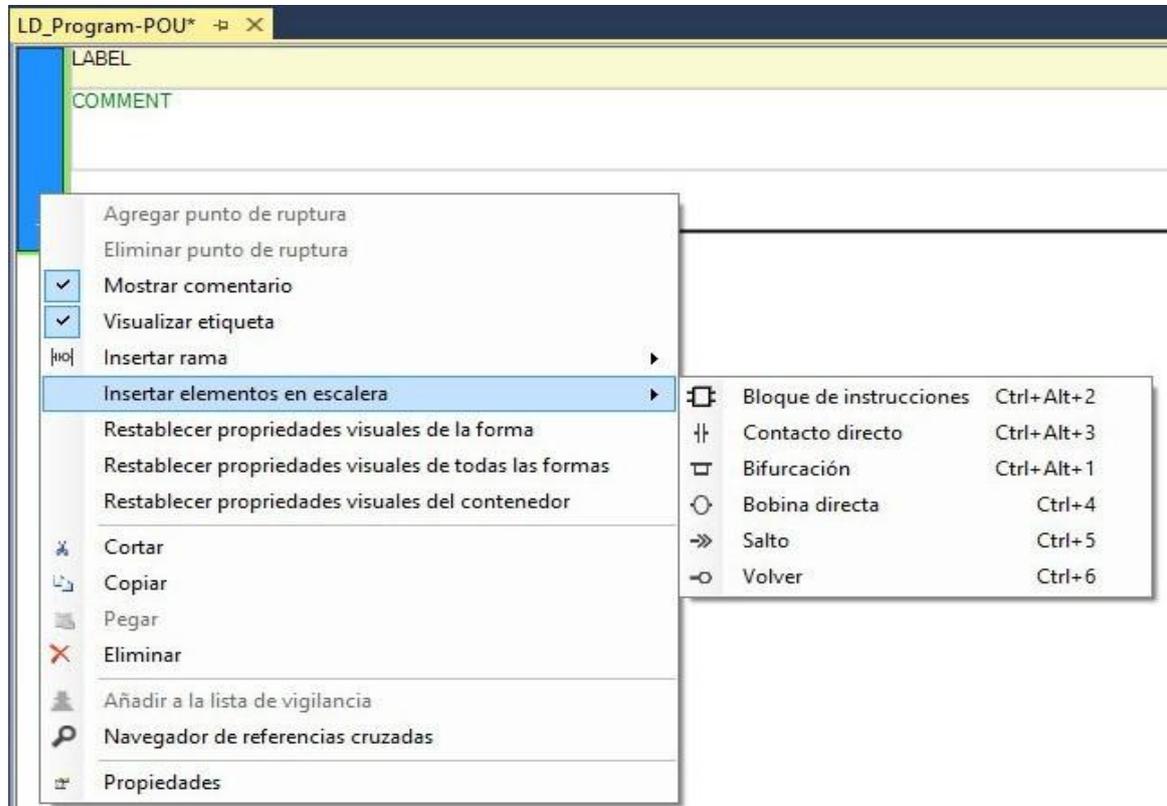
- ✓ Editor de idioma de diagrama de lógica de escalera(LD)
- ✓ Editor multi-idioma ubicado en el menú Herramientas
- ✓ Cuadro de herramientas de LD

Para agregar un elemento de contacto a un programa de diagrama de lógica de escalera:

En el editor de idioma de LD, compruebe que el programa de LD tenga una línea definida para el contacto. A continuación, realice una de las siguientes acciones:

Haga clic con el botón secundario en una línea o un elemento del editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Contacto directo.

Si se abre el Selector de variable, seleccione una variable o haga clic en Aceptar sin seleccionar una variable para agregar el elemento de Contacto directo.



- ✓ Seleccione una línea o un elemento en el editor de idioma de LD y pulse:

CTRL+ALT+3 para agregar el elemento de Contacto directo a la izquierda del elemento seleccionado o la línea.

CTRL+3 para agregar el elemento de Contacto directo a la derecha del elemento seleccionado o la línea.

- Seleccione una línea o un elemento en el editor de idioma de LD y a continuación:

Haga clic en Herramientas > Editor multi-idioma> Insertar contacto antes para agregar el contacto a la izquierda del elemento o línea seleccionados.

Haga clic en Herramientas > Editor multi-idioma> Insertar contacto después para agregar el contacto a la derecha del elemento o línea seleccionados.

✓ Abra el cuadro de herramientas de LD para mostrar los elementos de contacto(Contacto directo, Contacto inverso, Contacto del flanco ascendente del pulso, Contacto del flanco descendente del pulso) y, a continuación:

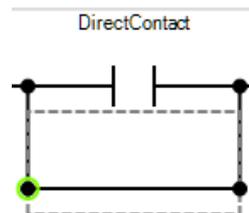
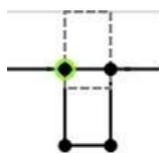
Haga doble clic en el elemento de contacto para agregarlo al editor de idioma de LD.

Arrastre el elemento de contacto al editor de idioma de LD y colóquelo en la línea.

Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para agregar el elemento.

1.-(opcional) Para insertar un contacto paralelo:

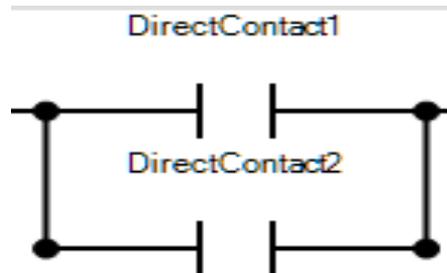
a) En el editor de idioma de LD, compruebe que el programa de LD tenga una bifurcación definida y, a continuación, haga clic con el botón secundario en el nivel superior de la bifurcación para abrir el menú de editor de idioma de LD.



b) En el menú de editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Contacto directo. El elemento se inserta en el nivel superior de la bifurcación.

c) Haga clic con el botón secundario en el nivel inferior de la bifurcación para abrir el menú de editor de idioma de LD.

d) En el menú de editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Contacto directo. El elemento se inserta en el nivel inferior de la bifurcación.



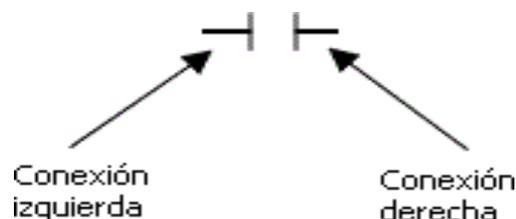
2.- (opcional) Para cambiar el tipo de contacto, en el editor de idioma seleccione el contacto y, a continuación, pulse la barra espaciadora hasta que aparezca el tipo de contacto en el editor de idioma.

Cada vez que pulse la barra espaciadora, el tipo de contacto cambia pasando de directo a inverso, a flanco ascendente del pulso y flanco descendente del pulso.

Contacto directo

Los contactos son componentes gráficos de programas de diagrama de lógica de escalera (LD).

Los contactos directos habilitan una operación booleana entre un estado de línea de conexión y una variable booleana.



El estado de la línea de conexión a la derecha del contacto es el AND lógico entre el estado de la línea de conexión izquierda y el valor de la variable asociada con el contacto.

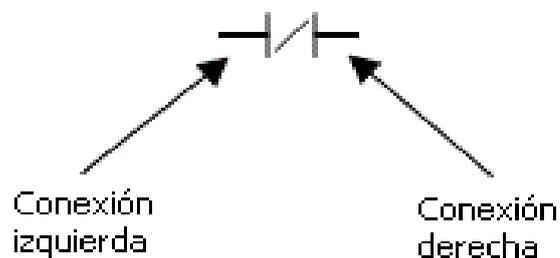
Ejemplo de contacto directo



Contacto inverso

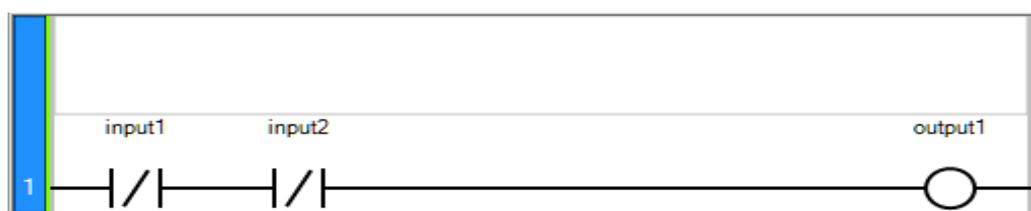
Los contactos son componentes gráficos de programas de diagrama de lógica de escalera (LD).

Los contactos inversos habilitan una operación booleana entre un estado de línea de conexión y la negación booleana de una variable booleana.



El estado de la línea de conexión a la derecha del contacto es el AND lógico entre el estado de la línea de conexión izquierda y la negación booleana del valor de la variable asociada con el contacto.

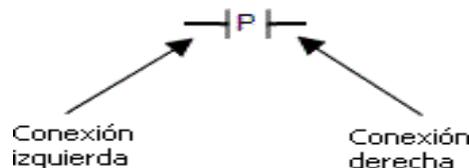
Ejemplo de contacto inverso



Contacto de flanco ascendente de pulso

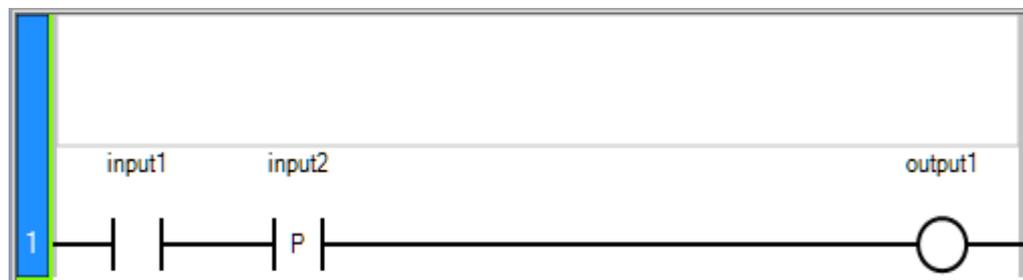
Los contactos son componentes gráficos de programas de diagrama de lógica de escalera (LD).

Los contactos de flanco ascendente de pulso (positivos) habilitan una operación booleana entre un estado de línea de conexión y el flanco ascendente de una variable booleana.



El estado de la línea de conexión a la derecha del contacto se define a Cierto si el estado de la línea de la conexión a la izquierda es Cierto, y el estado de la variable asociada cambia de Falso a Cierto. El estado se restablece a Falso en el resto de casos.

Ejemplo de contacto de flanco ascendente de pulso



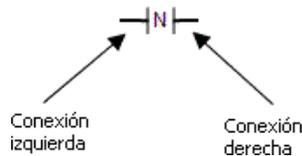
Recomendación: Restringir el uso de variables de salida con contactos de extremo.

Se recomienda no utilizar salidas ni variables con un contacto del flanco de subida del pulso (positivo) o un contacto del flanco descendente del pulso (negativo). Estos contactos son para entradas físicas en un diagrama de escalera. Para detectar el flanco de una variable o una salida, se recomienda utilizar el bloque de funciones R_TRIG/F_TRIG, que es compatible y funciona con cualquier lenguaje y en cualquier ubicación del programa.

Contacto de flanco descendente de pulso

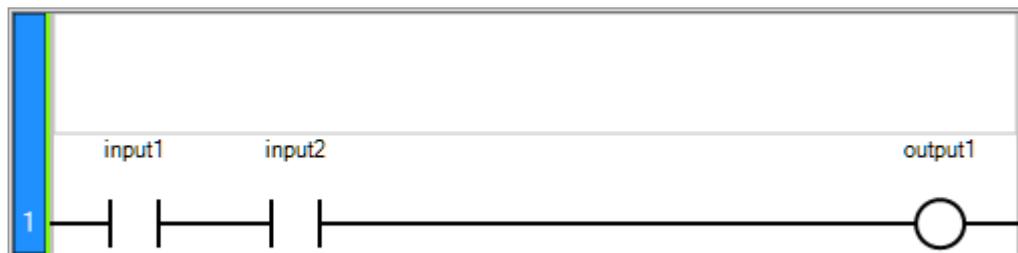
Los contactos son componentes gráficos de programas de diagrama de lógica de escalera (LD).

Los contactos de flanco descendente de pulso (negativos) habilitan una operación booleana entre un estado de línea de conexión y el flanco descendente de una variable booleana.



El estado de la línea de conexión a la derecha del contacto se define a Cierto si el estado de la línea de la conexión a la izquierda es Cierto, y el estado de la variable asociada cambia de Cierto a Falso. El estado se restablece a Falso en el resto de casos.

Ejemplo de contacto de flanco descendente de pulso



Recomendación: Restringir el uso de variables de salida con contactos de extremo.

Se recomienda no utilizar salidas ni variables con un contacto del flanco de subida del pulso (positivo) o un contacto del flanco descendente del pulso (negativo). Estos contactos son para entradas físicas en un diagrama de escalera. Para detectar el flanco de una variable o una salida, se recomienda utilizar el bloque de funciones R_TRIG/F_TRIG, que es compatible y funciona con cualquier lenguaje y en cualquier ubicación del programa.

Retorno

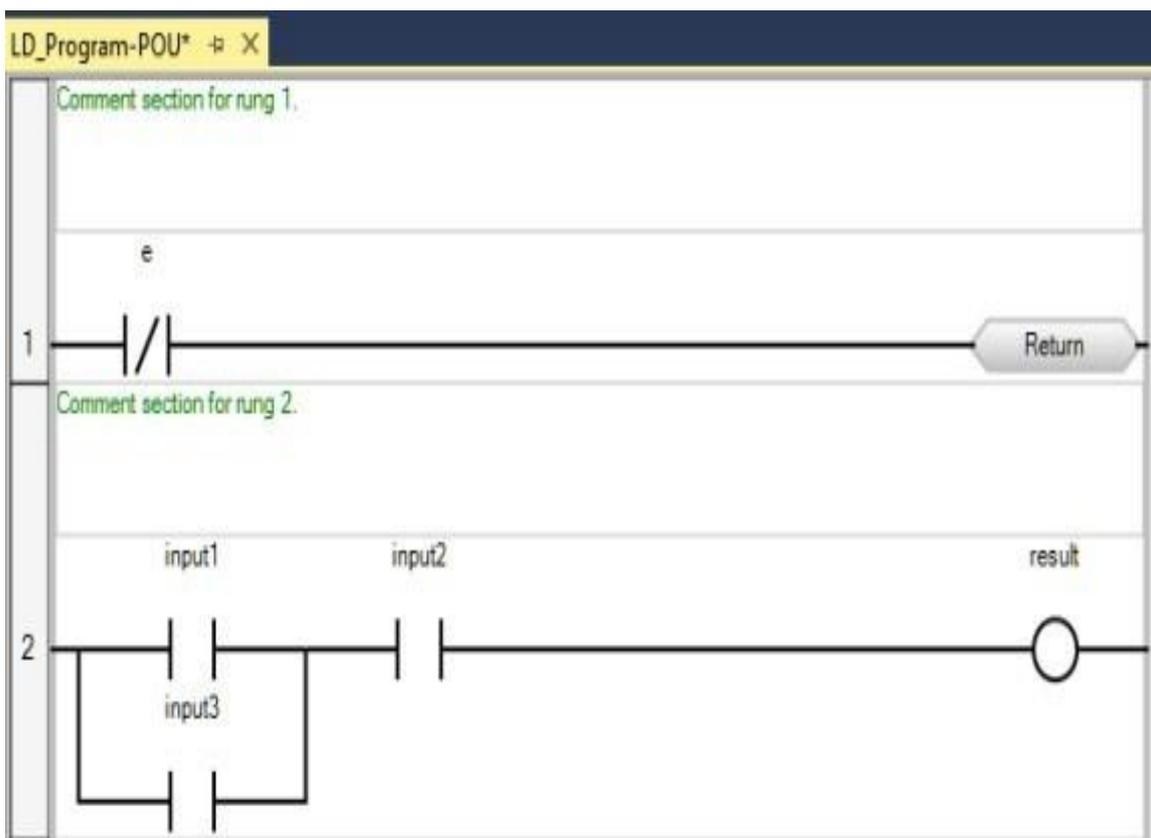
Los retornos son salidas que representan un final condicional de un programa de diagrama de lógica de escalera (LD).

No se pueden colocar conexiones a la derecha de un elemento de retorno.

Si la línea de conexión de la izquierda tiene el estado booleano Cierto, el diagrama finaliza sin ejecutar las instrucciones ubicadas en las siguientes líneas del diagrama.

Si el diagrama LD es una función, su nombre se asocia con una bobina de salida para establecer el valor de retorno (de vuelta al programa de llamada).

Ejemplo de retorno



Para insertar un retorno en un programa de diagrama de lógica de escalera: Realice una de las siguientes acciones:

- ✓ Haga clic en una línea o un elemento en el editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Retorno.
- ✓ Seleccione una línea o un elemento en el editor de idioma de LD y, a continuación, pulse CTRL+6.
- ✓ Seleccione una línea o un elemento en el editor de idioma de LD y, a continuación, haga clic en Herramientas > Editor multi-idioma> Insertar retorno.
- ✓ Abra el cuadro de herramientas de LD para mostrar el elemento de retorno y, a continuación:

Haga doble clic en el elemento de retorno para agregarlo al editor de idioma de LD.

Arrastre el elemento de retorno al editor de idioma de LD y colóquelo en la línea.

Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para agregar el elemento.

Salto

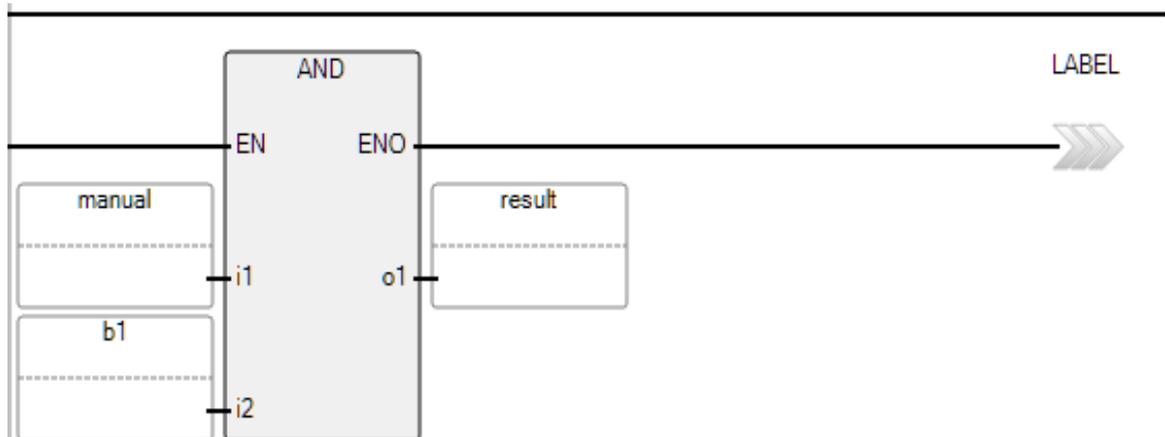
Los saltos son elementos condicionales o incondicionales que controlan la ejecución de programas de diagrama de lógica de escalera (LD).

Anotación de salto

La siguiente anotación indica un salto en una etiqueta:

>>LABEL: salto a una etiqueta de nombre "LABEL" Ejemplo de salto

Ejemplo de salto



Para insertar un salto:

Realice una de las siguientes acciones:

- ✓ Haga clic con el botón secundario en una línea o un elemento en el editor de idioma de LD, seleccione Insertar elementos de lógica de escalera y, a continuación, haga clic en Salto.
- ✓ Seleccione una línea o un elemento en el editor de idioma de LD y, a continuación, pulse CTRL+5.
- ✓ Seleccione una línea o un elemento en el editor de idioma de LD y, a continuación, haga clic en Herramientas > Editor multi-idioma> Insertar salto.
- ✓ Abra el cuadro de herramientas de LD para mostrar el elemento de salto y a continuación:

Haga doble clic en el elemento de salto para agregarlo al editor de idioma de LD.

Arrastre el elemento de salto al editor de idioma de LD y colóquelo en la línea.

Aparece un signo más (+) en el editor de idioma de LD para mostrar un destino válido. Suelte el botón del ratón para agregar el elemento.

Bloques de instrucciones en programas de LD

El conjunto de instrucciones de Connected Components Work bench incluye bloques de instrucciones conformes con IEC61131-3. Los bloques de instrucciones incluyen, en conjunto, bloques de funciones, funciones y operadores. Puede conectar entradas y salidas de bloque de instrucciones a variables, contactos, bobinas y otras entradas y salidas de bloques de instrucciones.

Convenciones de bloques de instrucciones

La especificación de lenguaje de programación IEC61131-3 trata numerosos aspectos de los controladores programables incluyendo la ejecución del sistema operativo, las definiciones de datos, los lenguajes de programación y los conjuntos de instrucciones. La especificación IEC61131-3 proporciona un conjunto mínimo de funcionalidad que se puede ampliar para ajustarse a las aplicaciones del usuario final.

Nombres de bloques de instrucciones

Las funciones y los bloques de funciones están representados por un cuadro que muestra el nombre de la instrucción y la versión abreviada de los nombres de parámetro. Para los bloques de funciones, el nombre de instancia se muestra sobre el nombre del bloque de funciones.

Parámetros de retorno de bloques de instrucciones

- ✓ El parámetro de retorno de una función tiene el mismo nombre que la función. El parámetro de retorno es la única salida.
- ✓ Los parámetros de retorno de un bloque de funciones pueden tener cualquier nombre. Múltiples parámetros de retorno pueden proporcionar múltiples salidas.
- ✓ Puede definir los parámetros de programas de varios dispositivos desplazándose por las pestañas de cada uno de los dispositivos en la vista Parámetros.

Trabajar en el editor de idioma de LD

Cuando agrega elementos a una línea en el programa de diagrama de lógica de escalera (LD), se agregan de acuerdo con los siguientes criterios.

- ✓ El primer elemento del peldaño se inserta en la posición seleccionada en el diagrama de lógica de escalera.
- ✓ Los elementos posteriores se insertan a la derecha del elemento seleccionado en el peldaño.
- ✓ No se puede insertar un elemento a la derecha de un retorno de bobina o salto.

Diferentes métodos para agregar un elemento al programa de diagrama de lógica de escalera:

Editor de lógica de escalera de LD

- ✓ Agregar elementos, eliminar elementos, y copiar y pegar elementos.
- ✓ Utilizar métodos abreviados de teclado de LD para agregar elementos.

Editor de texto de LD

- ✓ Agrega, modifica y elimina elementos.
- ✓ Copiar y pegar elementos de/a RS Logix 500y<RSLX5000>.

Editor multi-idioma ubicado en el menú Herramientas

- ✓ Agregar elementos.
- ✓ Exportar una imagen del programa de LD.
- ✓ Habilitar o deshabilitar la apertura automática del Selector de variables y del Selector de bloque de instrucciones.
- ✓ Cuadro de herramientas de LD
- ✓ Agregar elementos.

Barra de herramientas de instrucción

- ✓ Agregar elementos.

Para modificar una variable desde el editor de idioma:

1.-En el editor de lenguaje, haga clic en el nombre de la variable para mostrar una lista desplegable de variables globales y locales.

2.-Realice una de las siguientes acciones:

- ✓ Escriba un nuevo nombre de variable en el cuadro de texto.
- ✓ Seleccione un nombre de variable diferente en la lista desplegable.

Para modificar una variable desde el selector de variables:

1.- En el editor de lenguaje, haga doble clic en la variable para abrir el selector de variables.

2.- Haga clic en el nombre de la variable y seleccione una variable diferente en la lista desplegable de variables globales y locales.

Haga clic en una variable existente y escriba los valores constantes en el cuadro de texto proporcionado.

Ejercicios:

1.- Elabore un diagrama de escalera que permita arrancar y parar tres bandas de manera secuencial.

2.- Elabore un diagrama de escalera que permita arrancar y parar un agitador utilizando un solo push botón para arrancarlo y/o pararlo.

Tema 5.-

Temporizadores

Objetivo particular.- Al término del tema, el participante practicará la elaboración de un programa básico de acuerdo con la capacitación recibida.



Los timers y los contadores le permiten controlar las operaciones basadas en tiempo o número de eventos. La lista enlista las instrucciones de timer y de contador disponibles.

Si usted quiere:	Instrucción:
Temporizador, retraso de desactivación	TON
Temporizador, retraso de activación	TOF
Temporizador retentivo, activación con retraso	RTO
Tiempo de retraso, activado, desactivado	TONOFF
Temporización del pulso	TP
Comprobar año del reloj de tiempo real	DOY
Diferencia horaria	TDF
Comprobar semana del reloj de tiempo real	TOW
Cuenta ascendente	CTU
Cuenta descendente	CTD
Cuenta ascendente/descendente	CTUD

Configuración de instrucciones de temporizador

La precisión temporal hace referencia al tiempo entre el momento en que el procesador habilita una instrucción de temporizador y el momento en que completa el intervalo temporizado.

El procesador utiliza la siguiente información de la instrucción de temporizador:

- ✓ Temporizador: la dirección del control del temporizador en el área de almacenamiento de datos de este último.
- ✓ Base temporal: determina cómo funciona el temporizador.
- ✓ Pre-ajuste: especifica el valor que debe alcanzar el temporizador para que el procesador active el bit Done.
- ✓ Valor acumulado: el número de incrementos de tiempo que ha contado la instrucción. Cuando se habilita esta opción, el temporizador actualiza este valor de forma continua.

TOF (temporizador, retraso de desactivación)

Aumenta un temporizador interno hasta un valor concreto. Detalles de la operación:

- ✓ Si se utiliza el parámetro EN con este bloque de instrucción, el temporizador comienza a incrementarse si EN se establece en Cierto, y continúa incrementándose, aunque EN se establezca en Falso.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

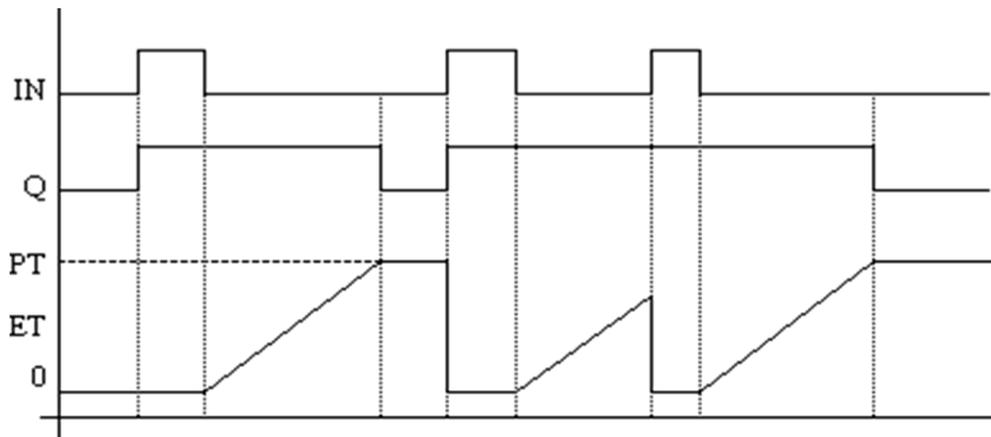
Esta instrucción se aplica a los controladores Micro810, Micro 820, Micro830, Micro850 y Micro870.



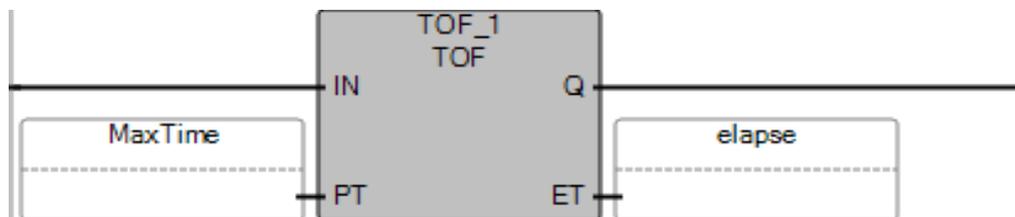
Utilice esta tabla para determinar los valores de parámetros de esta instrucción.

P a r á m e t r o	Tipo de parámetro	T i p o d e d a t o s	Descripción
I N	Entrada	B O O L	Control de entrada. Cierto: se ha detectado un flanco descendente, empieza a aumentar el temporizador interno. Falso: se ha detectado un flanco ascendente, detiene y restablece el temporizador interno.
P T	Entrada	T I M E	Tiempo programado máximo. Consulte Tipo de datos de tiempo.
Q	Salida	B O O L	Cierto: no transcurre el tiempo total. Falso: ha transcurrido el tiempo total.
E T	Salida	T I M E	Tiempo actual transcurrido. Los valores posibles van desde 0 ms a 1193 h, 2 m, 47 s, 294 ms.

Diagrama de temporización TOF



Ejemplo de diagrama de lógica de escalera TOF



Resultados

Monitorización de variable

Variables globales de usuario - Micro870 Variables locales - RA_TOF_LD Variables de

	Nombre	Valor lógico	Valor fisico	Bloquear	Tipo de d
+	TOF_1	<input type="checkbox"/>	TOF
▶	in	<input type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	MaxTime	T#3s	N/A	<input type="checkbox"/>	TIME
	elapse	T#2s18ms	N/A	<input type="checkbox"/>	TIME
	output	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL

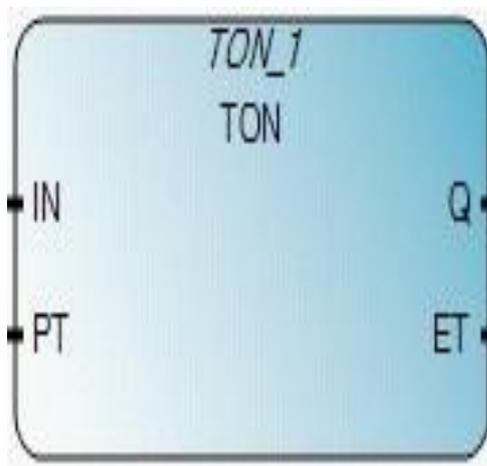
Cerrar

TON (temporizador, retraso de activación)

Aumenta un temporizador interno hasta un valor concreto. Detalles de la operación:

- ✓ No utilice un salto para omitir un bloque de instrucción TON en un diagrama de lógica de escalera (LD). Si se utiliza un salto, el temporizador TON continuará después de transcurrir el tiempo.
- ✓ Por ejemplo: La llamada 1 contiene un salto; la llamada 2 contiene un bloque de instrucción TON con un tiempo transcurrido de 10 segundos; habilite el salto de la llamada 1 a la 3; deshabilite el salto después de 30 segundos; el tiempo transcurrido será de 30 segundos; no 10 segundos, tal y como se define en el tiempo transcurrido.
- ✓ Si se utiliza el parámetro EN con TON, el temporizador comienza a incrementarse si EN se establece en Cierto, y continúa incrementándose aunque EN se establezca en Falso.

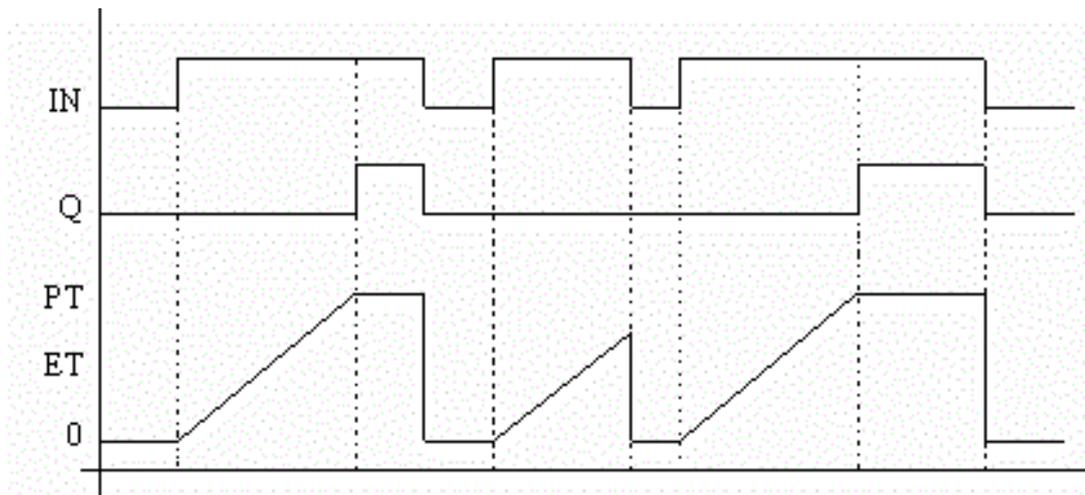
Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.



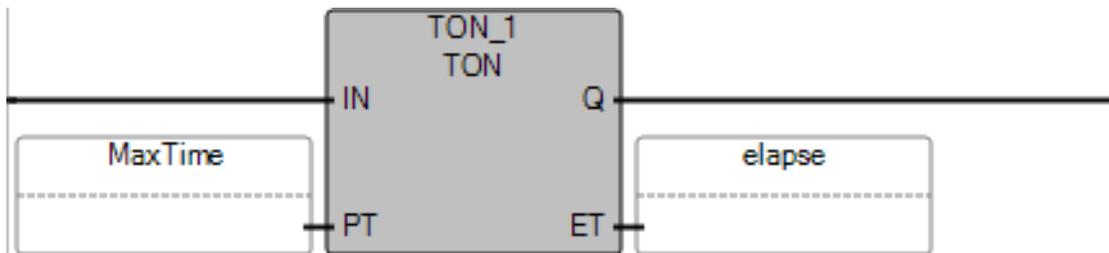
Utilice esta tabla para determinar los valores de parámetros de esta instrucción.

P a r á m e t r o	T i p o d e p a r á m e t r o	T i p o d e d a t o s	Descripción
I N	E n t r a d a	B O O L	Control de entrada. Cierto: si se detecta un flanco ascendente, empieza a aumentar el temporizador interno. Falso: si se detecta un flanco descendente, detiene y restablece el temporizador interno.
P T	E n t r a d a	T I M E	Tiempo programado máximo definido mediante el tipo de datos de tiempo.
Q	S a l i d a	B O O L	Cierto: ha transcurrido el tiempo programado. Falso: no ha transcurrido el tiempo programado.
E T	S a l i d a	T I M E	Tiempo actual transcurrido. Los valores posibles van desde 0 ms a 1193 h, 2 m, 47 s, 294 ms.

Diagrama de temporización TON



Ejemplo de diagrama de lógica de escalera TON



Resultados

Monitorización de variable

Variables globales de usuario - Micro870 Variables locales - RA_TON_LD Variables de

	Nombre	Valor lógico	Valor fisico	Bloquear	Tipo de d
+	TON_1	<input type="checkbox"/>	TON
	in	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	MaxTIME	T#3s	N/A	<input type="checkbox"/>	TIME
	output	<input type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
▶	elapse	T#1s493ms	N/A	<input type="checkbox"/>	TIME

Cerrar

TONOFF (tiempo de retraso, activado, desactivado)

Retarda la activación de una salida en una línea cierta, y, a continuación, retarda la desactivación de la salida en la línea falsa.

Detalles de la operación:

- ✓ Si se utiliza el parámetro EN con TONOFF, el temporizador comienza a incrementarse si EN se establece en Cierto, y continúa incrementándose aunque EN se establezca en Falso.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

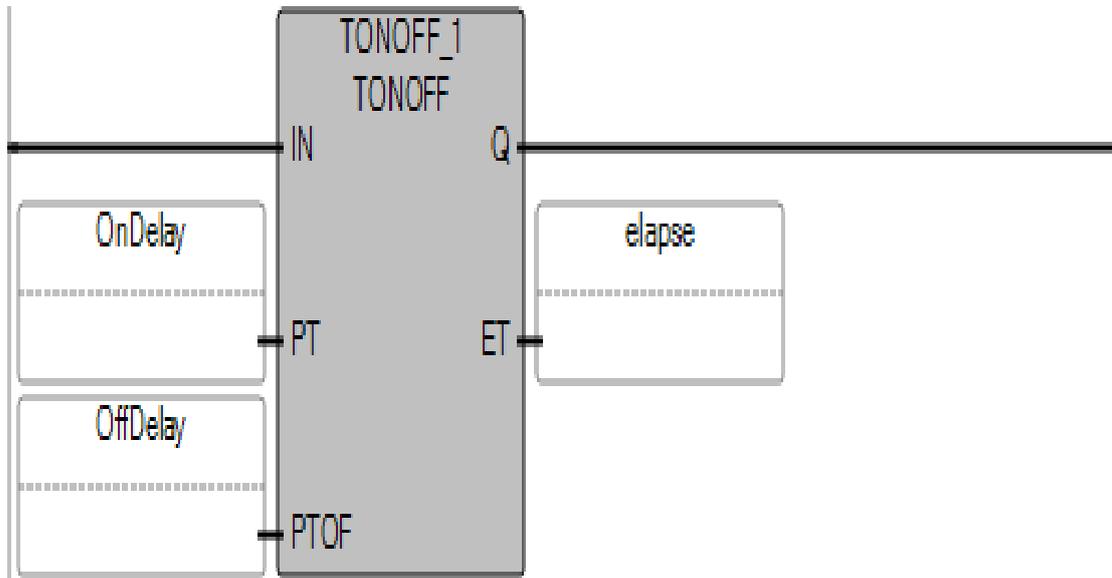


Utilice esta tabla para determinar los valores de parámetros de esta instrucción.

P a r á m e t r o	T i p o d e p a r á m e t r o	T i p o d e d a t o s	Descripción
I N	E n t r a d a	B O O L	<p>Control de entrada.</p> <p>Cierto: flanco ascendente detectado (EN pasa de 0 a 1):</p> <ul style="list-style-type: none"> • Inicia la activación de retardo del temporizador (PT). • Si no ha transcurrido el tiempo de desactivación del retardo programado (PTOF), reinicia el temporizador de retardo (PT). <p>Falso: flanco descendente detectado (EN pasa de 1 a 0):</p> <ul style="list-style-type: none"> • Si no ha transcurrido el tiempo de activación del retardo programado (PTOF), detiene el temporizador PT y restablece ET. • Si se supera el tiempo de activación de retardo programado (PT), inicia el temporizador de retardo de desactivación (PTOF).
P T	E n t r a d a	T I M E	<p>La configuración de tiempo de activación de retardo se define mediante el tipo de datos de tiempo.</p>

P T O F	E n t r a d a	T I M E	La configuración de tiempo de desactivación de retardo se define mediante el tipo de datos de tiempo.
Q	S a l i d a	B O O L	Cierto: el tiempo de retardo de conexión programado ha transcurrido y el tiempo de retardo de desconexión programado no ha transcurrido.
E T	S a l i d a	T I M E	<p>Tiempo actual transcurrido. Los valores posibles van desde 0 ms a 1193 h, 2 m, 47 s, 294 ms.</p> <p>Si el tiempo de retardo de conexión programado ha transcurrido y el temporizador de retardo de desconexión no empieza, el tiempo transcurrido (ET) permanece en el valor de retardo de conexión (PT). Si se supera el tiempo de retraso programado y no se inicia el temporizador de retardo, el tiempo transcurrido (ET) permanece dentro del valor de retardo de desconexión (PTOF) hasta que se vuelva a activar el flanco ascendente.</p>

Ejemplo de diagrama de lógica de escalera TONOFF



Resultados

Monitorización de variable

Variables globales de usuario - Micro870 Variables locales - RA_TONOFF_LD Variable

	Nombre	Valor lógico	Valor fisico	Bloquear	Tipo de d
+	TONOFF_1	<input type="checkbox"/>	TONOFF
▶	in	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	OnDelay	T#3s	N/A	<input type="checkbox"/>	TIME
	OffDelay	T#5s	N/A	<input type="checkbox"/>	TIME
	output	<input type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	elapse	T#1s60ms	N/A	<input type="checkbox"/>	TIME

Cerrar

TP (temporización del pulso)

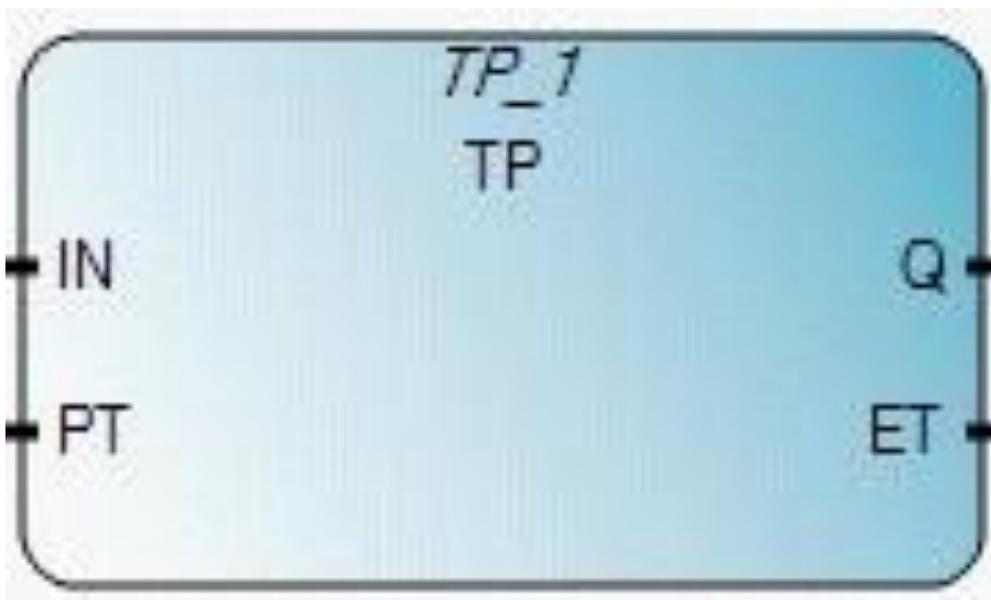
En un flanco ascendente, aumenta un temporizador interno hasta un valor concreto. Si el temporizador finaliza, se restablece la hora interna.

Detalles de la operación:

Si se utiliza el parámetro EN con TP, el temporizador comienza a incrementarse si EN se establece en Cierto, y continúa incrementándose aunque EN se establezca en Falso.

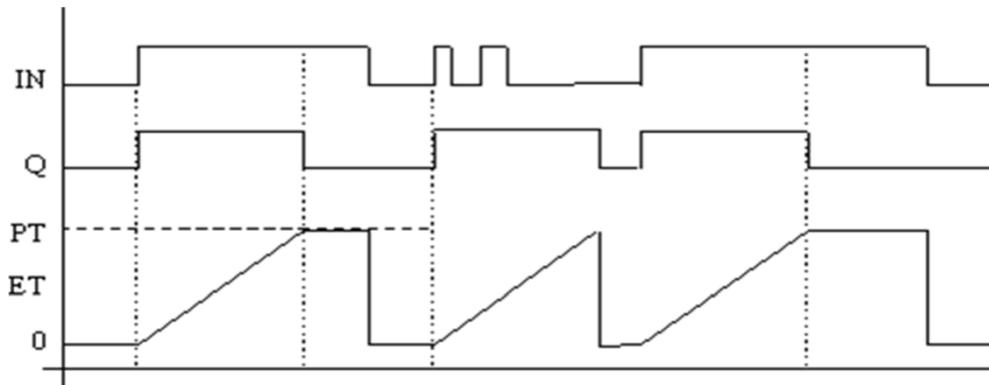
Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro 820, Micro830, Micro850 y Micro870.

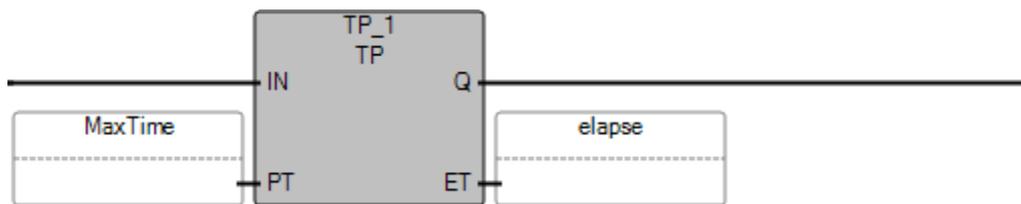


P a r á m e t r o	T i p o d e p a r á m e t r o	T i p o d e d a t o s	Descripción
I N	E n t r a d a	B O O L	Cierto: si se detecta un flanco ascendente, empieza a aumentar el temporizador interno (si no está aumentando ya). Falso: si el temporizador ha finalizado, restablece el temporizador interno. Cualquier cambio de IN durante el recuento no tiene efecto alguno.
P T	E n t r a d a	T I M E	Tiempo programado máximo definido mediante el tipo de datos de tiempo.
Q	S a l i d a	B O O L	Cierto: el temporizador está contando. Falso: el temporizador no está contando.
E T	S a l i d a	T I M E	Tiempo actual transcurrido. Los valores posibles van desde 0 ms a 1193 h, 2 m, 47 s, 294 ms.

Diagrama de temporización TP



Ejemplo de diagrama de lógica de escalera TP



Resultados

Monitorización de variable					
Variables globales de usuario - Micro870		Variables locales - RA_TP_LD		Variables del : ▾	
Nombre	Valor lógico	Valor físico	Bloquear	Tipo de d	
TP_1	<input type="checkbox"/>	TP	
in	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL	
MaxTime	T#3s	N/A	<input type="checkbox"/>	TIME	
output	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL	
elapse	T#593ms	N/A	<input type="checkbox"/>	TIME	

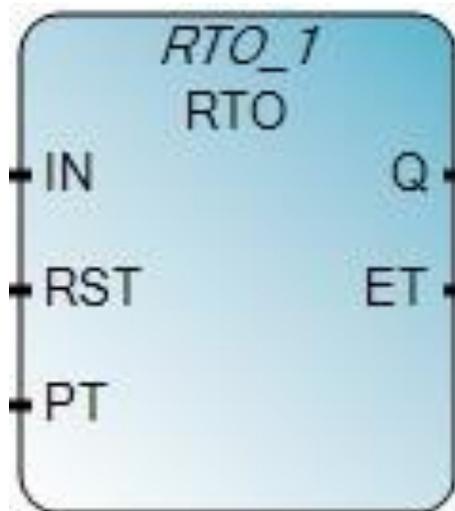
RTO (temporizador retentivo, activación con retraso)

Aumenta un temporizador interno si la entrada está activa, pero no restablece el temporizador interno si la entrada cambia a inactiva.

Detalles de la operación:

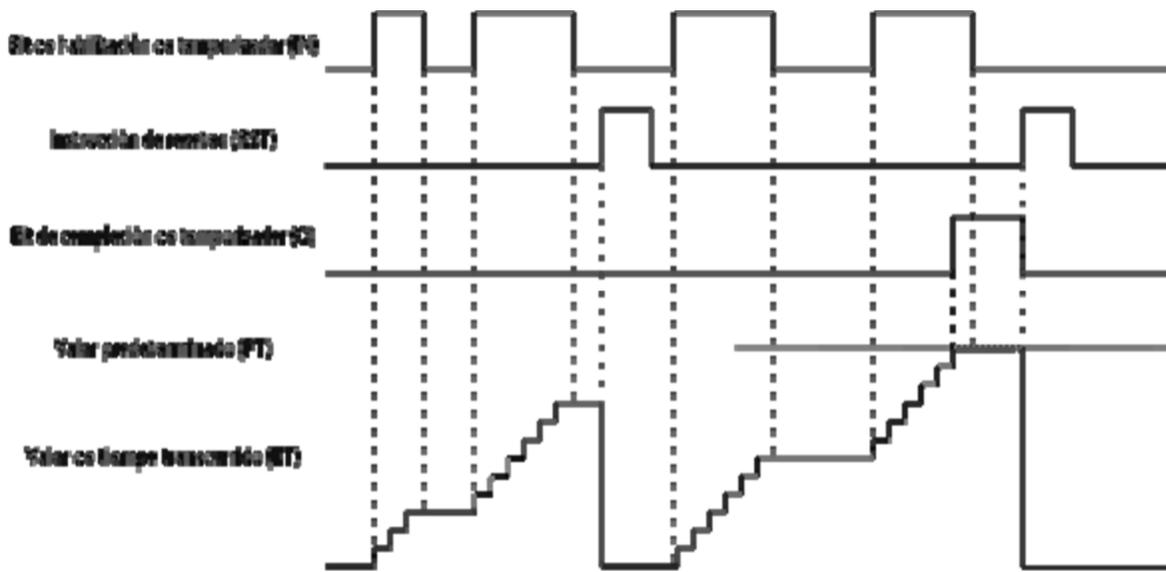
- ✓ Controlador Micro810 o Micro820, el temporizador interno de RTO no persiste en una desconexión y conexión de forma predeterminada. Para hacer que el temporizador interno persista, establezca el parámetro de configuración Retenido en Cierto.
- ✓ Controlador Micro830 o Micro850, el temporizador interno de RTO persiste en una desconexión y conexión.
Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro820, Micro830, Micro850 y Micro870.

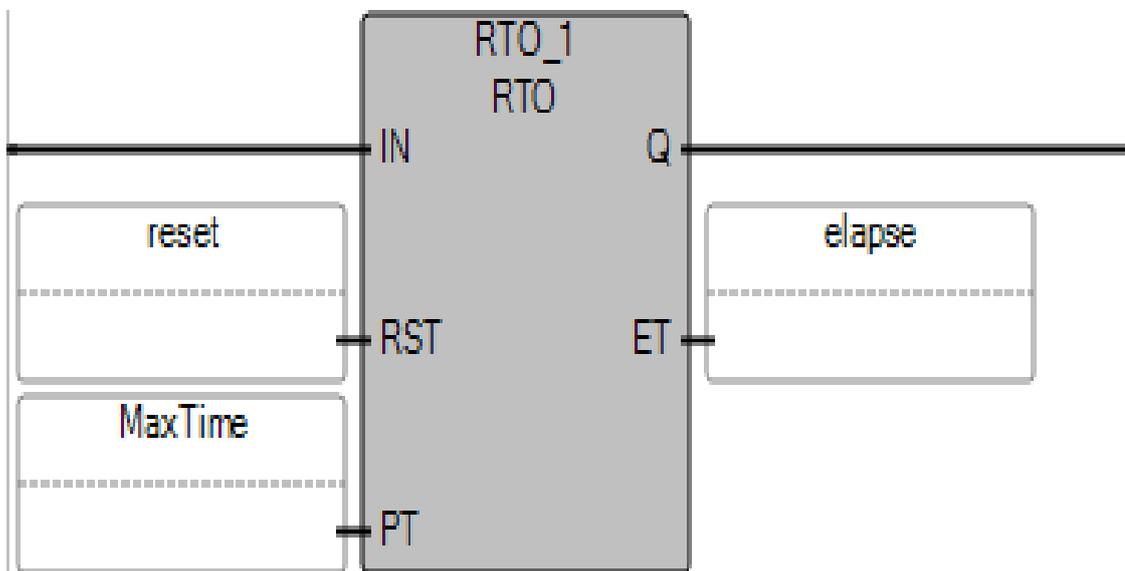


P a r á m e t r o	T i p o p a r á m e t r o	T i p o d e d a t o s	Descripción
I N	E n t r a d a	B O O L	Control de entrada. Cierto: flanco ascendente, comienza a aumentar el temporizador interno. Falso: flanco descendente, se detiene y no restablece el temporizador interno.
R S T	E n t r a d a	B O O L	Cierto: flanco ascendente, restablece el temporizador interno. Falso: no restablece el temporizador interno.
P T	E n t r a d a	T I M E	Tiempo de retraso programado máximo. PTO se define utilizando el tipo de datos de tiempo.
Q	S a l i d a	B O O L	Cierto: ha transcurrido el tiempo de retardo programado. Falso: no ha transcurrido el tiempo de retraso programado.
E T	S a l i d a	T I M E	Tiempo actual transcurrido. Los valores van desde 0 ms a 1193 h, 2 m, 47 s, 294 ms. ET se define utilizando el tipo de datos de tiempo.

Ejemplo de diagrama de temporización de RTO



Ejemplo de diagrama de lógica de escalera RTO



DOY (comprobar año del reloj de tiempo real)

Activa una salida si el valor del reloj de tiempo real (RTC) se encuentra dentro del intervalo del ajuste Momento del año.

Detalles de la operación:

Si no hay un RTC, la salida siempre está desactivada.

Configure cualquiera de los parámetros de entrada Time con intervalos válidos, como se especifica en el tipo de datos DOYDATA. Un valor no válido hace fallar al controlador si Time X. Enable está establecido en Cierto y hay un RTC habilitado.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro 820, Micro830, Micro850 y Micro870.



P a r á m e t r o	T i p o d e p	T d e d a t o s	Descripción
E N	E n t r a	B O O L	Habilitar instrucción. Cierto: realiza la operación. Falso: no realiza la operación.
T i m e A	E n t r a d a	D O Y D A T A	Configuración de momento del año del canal A. Utilice el tipo de datos DOYDATA para configurar Time A.
T i m e B	E n t r a d a	D O Y D A T A	Configuración de momento del año del canal B. Utilice el tipo de datos DOYDATA para configurar Time B.
T i m e C	E n t r a d a	D O Y D A T A	Configuración de momento del año del canal C. Utilice el tipo de datos DOYDATA para configurar Time C.
T i m e D	E n t r a d a	D O Y D A T A	Configuración de momento del año del canal D. Utilice el tipo de datos DOYDATA para configurar Time D.

P a r á m e t r o	T i p o d e p a r á m e t r o	T i p o d e d a t o s	Descripción
D O Y	S a l i d a	B O O L	Si es Cierto, el valor del reloj de tiempo real se encuentra dentro del intervalo de la configuración de momento del año de cualquiera de los cuatro canales.

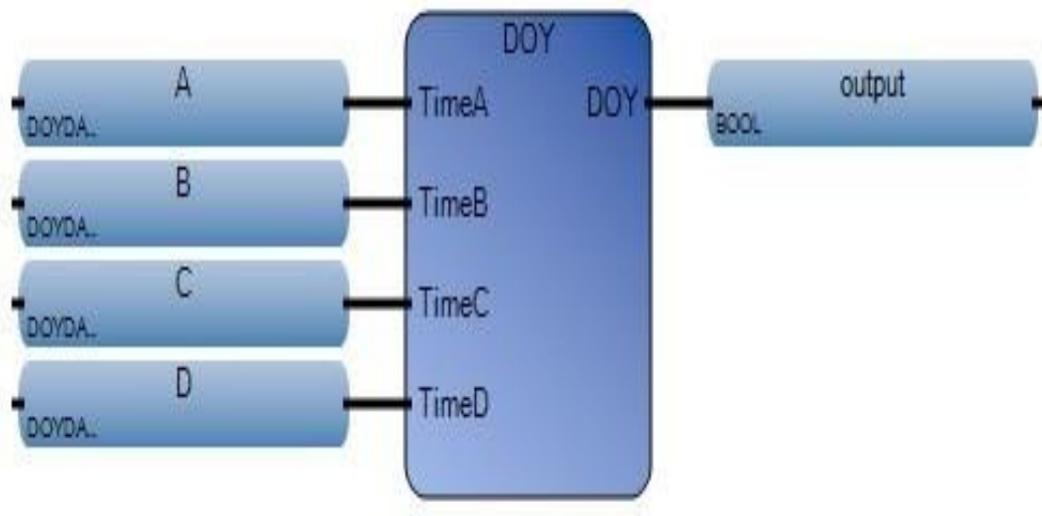
Tipo de datos DOYDATA

En la tabla siguiente se describe el tipo de datos DOYDATA.

P a r á m e t r o	T i p o d e d a t o s	Descripción
E n a b l e	B O O L	Cierto: Habilitado; Falso: Deshabilitado
Y e a r l y C e n t u r i a l	B O O L	Tipo de temporizador (0: temporizador anual; 1: temporizador secular).
Y e a r O n	U I N T	Valor de año activado (debe estar entre [2000...2098]).
M o n t h O n	U S I N T	Valor de mes activado (debe estar entre [1...12]).
D a y	U S I	Valor de día activado (debe estar entre [1...31], depende del valor "MonthOn").

O n	N T	
Y e a r O f f	U I N T	Valor de año desactivado (debe estar entre [2000...2098]).
M o n t h O f f	U S I N T	Valor de mes desactivado (debe estar entre [1...12]).
D a y O f f	U S I N T	Valor de día desactivado (debe estar entre [1...31], depende del valor "MonthOff").

Ejemplo de diagrama de bloque de funciones DOY

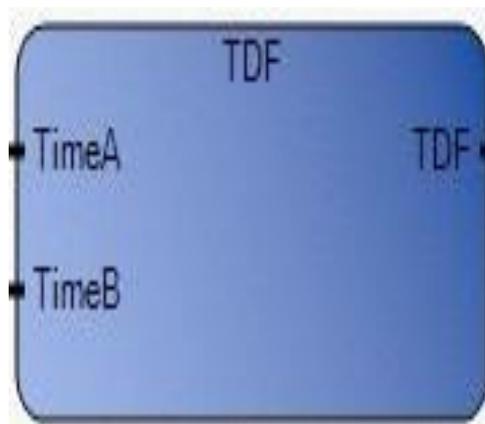


TDF (diferencia horaria)

Calcula la diferencia horaria entre Tiempo A y Tiempo B.

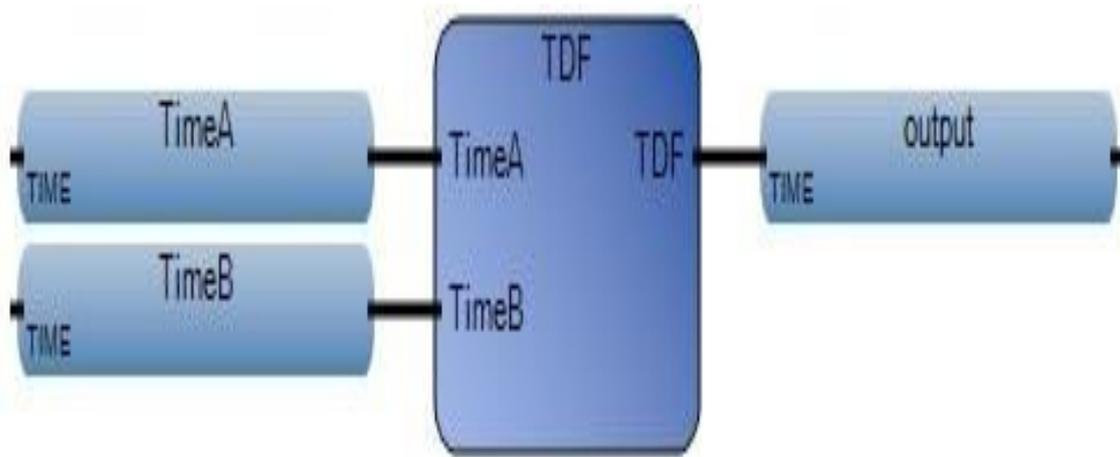
Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro 820, Micro830, Micro850 y Micro870.



P a r á m e t r o	T i p o d e p	Tipo de datos	Descripción
E N	E n t r a d a	B O O L	Habilitar instrucción. TRUE - perform current computation. FALSE - there is no computation. Se aplica a programas de diagrama de lógica de escalera.
T i m e A	E n t r a d a	T I M E	La hora de inicio para cálculo de diferencia de tiempo.
T i m e B	E n t r a d a	T I M E	La hora de finalización para cálculo de diferencia de tiempo.
E N O	S a l i d a	B O O L	Habilita la salida. Se aplica a programas de diagrama de lógica de escalera.
T D F	S a l i d a	T I M E	La diferencia horaria entre las dos entradas de hora. TDF es un nombre o ID de PIN

Ejemplo de diagrama de bloque de funciones TDF



Monitorización de variable

Variables globales de usuario - Micro870 Variables locales - RA_TDF_LD Variables de

	Nombre	Valor lógico	Valor fisico	Bloquear	Tipo de d
	TimeA	T#10s	N/A	<input type="checkbox"/>	TIME
	TimeB	T#5s	N/A	<input type="checkbox"/>	TIME
▶	output	T#49d17h2m4	N/A	<input type="checkbox"/>	TIME

Cerrar

TOW (comprobar semana del reloj de tiempo real)

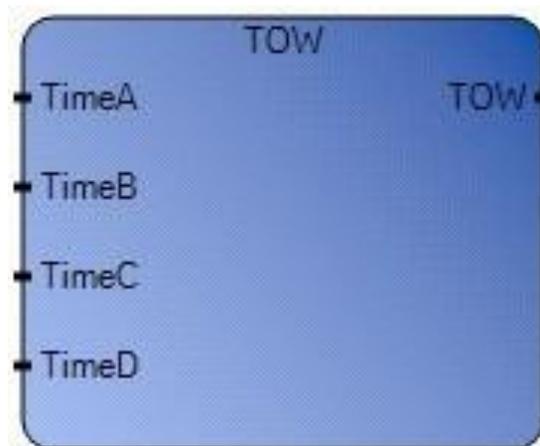
Activa una salida si el valor del reloj de tiempo real (RTC) se encuentra dentro del intervalo del ajuste Momento de la semana.

Detalles de la operación:

- ✓ Si no hay un RTC, la salida siempre está desactivada.
- ✓ Configure cualquiera de los parámetros de entrada Time con intervalos válidos, como se especifica en el tipo de datos TOWDATA. Un valor no válido hace fallar al controlador si Time X. Enable está establecido en Cierto y hay un RTC habilitado.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro 820, Micro830, Micro850 y Micro870.



P a r á m e t r o	T i p o d e p .	T i p o d e d a t o s	Descripción
E N	E n t r a d a	B O O L	Habilitar instrucción. Si EN = Cierto, se realiza la operación. Si EN = Falso, no se realiza la operación.
T i m e A	E n t r a d a	T O W D A T A	Configuración de hora del día del canal A. Utilice el tipo de datos TOWDATA para definir Tiempo A.
T i m e B	E n t r a d a	T O W D A T A	Configuración de hora del día del canal B. Utilice el tipo de datos TOWDATA para definir Tiempo B.
T i m e C	E n t r a d a	T O W D A T A	Configuración de hora del día del canal C. Utilice el tipo de datos TOWDATA para definir Tiempo C.

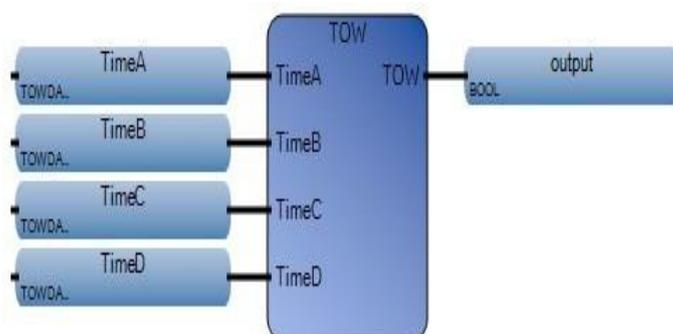
T i m e D	E n t r a d a	T O W D A T A	Configuración de hora del día del canal D. Utilice el tipo de datos TOWDATA para definir Tiempo D.
T O W	S a l i d a	B O O L	Si es Cierto, el valor del reloj de tiempo real se encuentra dentro del intervalo de la configuración de hora del día de cualquiera de los cuatro canales.

Tipo de datos TOWDATA

La tabla siguiente describe el tipo de datos TOWDATA:

		Descripción
Enable	BOOL	CIERTO: Habilitar; Falso: Deshabilitar.
Daily Weekly	BOOL	Tipo de temporizador (0: temporizador diario; 1: temporizador semanal).
Day On	USINT	Valor de día de la semana activado (debe estar entre [0...6]).
Hour On	USINT	Valor de hora activado (debe estar entre [0...23]).
Min On	USINT	Valor de minuto activado (debe estar entre [0...59]).
Day Off	USINT	Valor de día de la semana desactivado (debe estar entre [0...6]).
Hour Off	USINT	Valor de hora desactivado (debe estar entre [0...23]).
Min Off	USINT	Valor de minuto desactivado (debe estar entre [0...59]).

Ejemplo de diagrama de bloque de funciones TOW

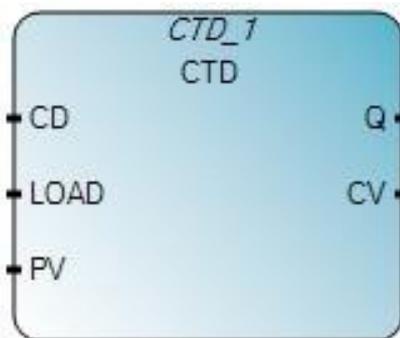


	Nombre	Valor lógico	Valor físico	Bloquear	Tipo de dato
-	TimeA	<input type="checkbox"/>	TOWDAT, ▾
	TimeA.Enable	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeA.Daily/Weekly	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeA.DayOn	1	N/A	<input type="checkbox"/>	USINT
	TimeA.HourOn	10	N/A	<input type="checkbox"/>	USINT
	TimeA.MinOn	20	N/A	<input type="checkbox"/>	USINT
	TimeA.DayOff	2	N/A	<input type="checkbox"/>	USINT
	TimeA.HourOff	15	N/A	<input type="checkbox"/>	USINT
	TimeA.MinOff	30	N/A	<input type="checkbox"/>	USINT
-	TimeB	<input type="checkbox"/>	TOWDAT, ▾
	TimeB.Enable	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeB.Daily/Weekly	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeB.DayOn	2	N/A	<input type="checkbox"/>	USINT
	TimeB.HourOn	15	N/A	<input type="checkbox"/>	USINT
	TimeB.MinOn	20	N/A	<input type="checkbox"/>	USINT
	TimeB.DayOff	3	N/A	<input type="checkbox"/>	USINT
	TimeB.HourOff	10	N/A	<input type="checkbox"/>	USINT
	TimeB.MinOff	30	N/A	<input type="checkbox"/>	USINT
-	TimeC	<input type="checkbox"/>	TOWDAT, ▾
	TimeC.Enable	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeC.Daily/Weekly	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeC.DayOn	3	N/A	<input type="checkbox"/>	USINT
	TimeC.HourOn	20	N/A	<input type="checkbox"/>	USINT
	TimeC.MinOn	10	N/A	<input type="checkbox"/>	USINT
	TimeC.DayOff	4	N/A	<input type="checkbox"/>	USINT
	TimeC.HourOff	25	N/A	<input type="checkbox"/>	USINT
	TimeC.MinOff	55	N/A	<input type="checkbox"/>	USINT
-	TimeD	<input type="checkbox"/>	TOWDAT, ▾
	TimeD.Enable	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeD.Daily/Weekly	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	TimeD.DayOn	5	N/A	<input type="checkbox"/>	USINT
	TimeD.HourOn	10	N/A	<input type="checkbox"/>	USINT
	TimeD.MinOn	15	N/A	<input type="checkbox"/>	USINT
	TimeD.DayOff	6	N/A	<input type="checkbox"/>	USINT
	TimeD.HourOff	35	N/A	<input type="checkbox"/>	USINT
	TimeD.MinOff	40	N/A	<input type="checkbox"/>	USINT
▶	output	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL ▾

CTD (recuento descendente)

Cuenta (enteros) de forma descendente desde un valor concreto hasta cero, de 1 en 1.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado. Esta instrucción se aplica a los controladores Micro810, Micro820, Micro830, Micro850 y Micro870.

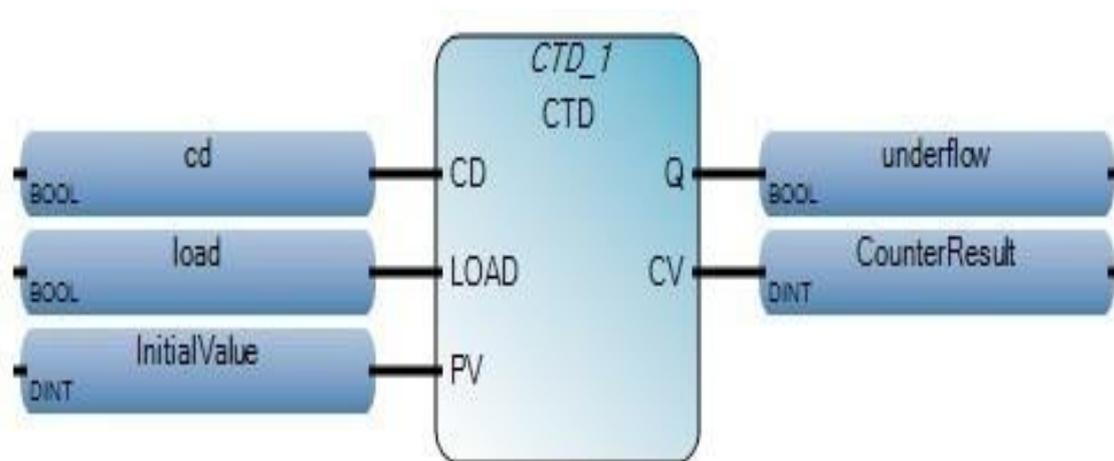


Utilice esta tabla para determinar los valores de parámetros de esta instrucción.

			Descripción
C D	E n t r a d a	B O O L	Recuento descendente. Cierto: flanco ascendente detectado, el recuento descendente se realiza en incrementos de uno. Falso: flanco descendente detectado, retiene el valor del contador en el mismo valor.
L O A D	E n t r a d a	B O O L	Carga comprueba el valor de PV respecto al valor de recuento descendente. Cierto: cuando el valor de recuento descendente es igual al valor PV, defina el valor CV para que sea igual que el valor PV. Falso: continúa incrementando el recuento descendente de uno en uno.

P V	E n t r a d a	D I N T	Valor máximo programado del contador.
Q	S a l i d a	B O O L	Indica si la instrucción del recuento descendente ha resultado en un número menor o igual al valor máximo del contador. Cierto: CV es menor o igual que cero (condición de caudal insuficiente). Falso: el valor de recuento descendente es mayor que cero.
C V	S a l i d a	D I N T	Valor de contador actual.

Ejemplo de diagrama de bloque de funciones CTD

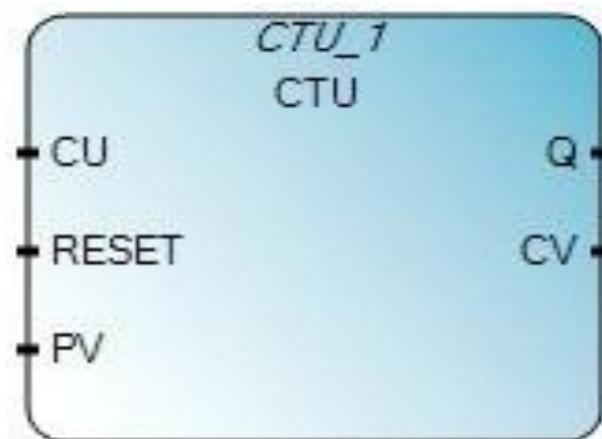


CTU (recuento ascendente)

CTU cuenta (enteros) de forma ascendente desde 0 hasta un valor concreto, de 1 en 1.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro820, Micro830, Micro850 y Micro870.



Monitorización de variable

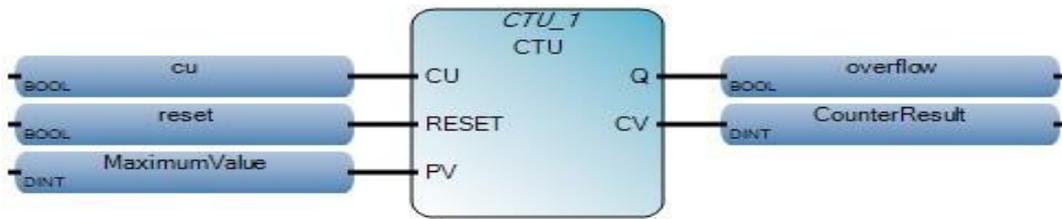
Variables globales de usuario - Micro870 Variables locales - RA_CTD_LD Variables del sistema - Micro870

	Nombre	Valor lógico	Valor físico	Bloquear	Tipo de datos
+	CTD_1	<input type="checkbox"/>	CTD
	cd	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	load	<input type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
	InitialValue	10	N/A	<input type="checkbox"/>	DINT
▶	CounterResult	9	N/A	<input type="checkbox"/>	DINT

Cerrar

Utilice esta tabla para determinar los valores de parámetros de esta instrucción.

			Descripción
C U	E n t r a d a	B O O L	Recuento ascendente. Cierto: flanco ascendente detectado, el recuento ascendente se realiza en incrementos de uno. Falso: flanco descendente detectado, retiene el valor del contador en el mismo valor.
R E S E T	E n t r a d a	B O O L	Reset comprueba el valor de PV respecto al valor de recuento ascendente. Cierto: cuando el valor de recuento ascendente es igual al valor PV, defina el valor CV como cero. Falso: continúa incrementando el recuento ascendente de uno en uno.
P V	E n t r a d a	D I N T	Valor máximo programado del contador.
Q	S a l i d a	B O O L	Indica si la instrucción del recuento ascendente ha resultado en un número mayor o igual al valor máximo del contador. Cierto: resultado del contador \leq PV (condición de desbordamiento). Falso: resultado del contador $>$ PV
C V	S a l i d a	D I N T	Resultado del contador actual.



Monitorización de variable

Variables globales de usuario - Micro870 Variables locales - RA_CTU_LD Variables del sistema - Micro870

Nombre	Valor lógico	Valor físico	Bloquear	Tipo de datos
CTU_1	<input type="checkbox"/>	CTU
cu	<input checked="" type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
reset	<input type="checkbox"/>	N/A	<input type="checkbox"/>	BOOL
MaximumValue	10	N/A	<input type="checkbox"/>	DINT
CounterReset	1	N/A	<input type="checkbox"/>	DINT

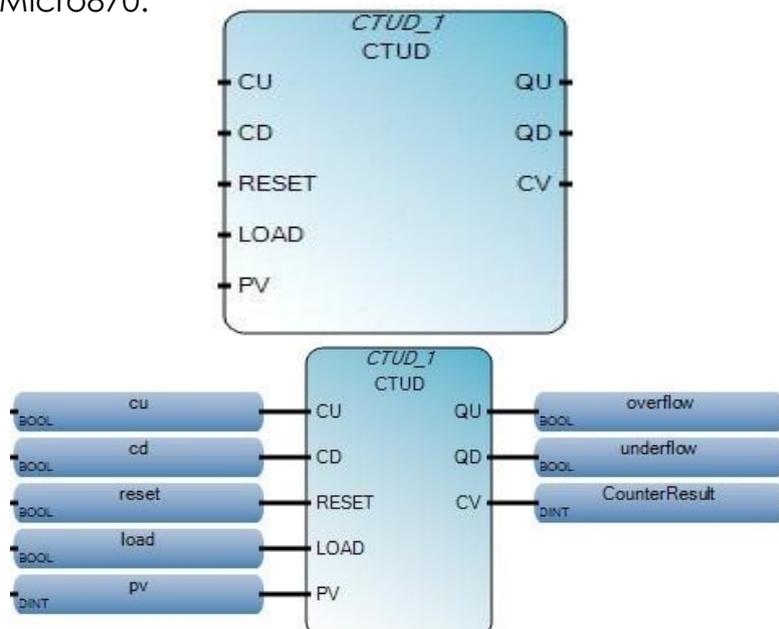
Cerrar

CTUD (recuento ascendente/descendente)

Cuenta enteros de forma ascendente desde 0 hasta un valor concreto, de 1 en 1, o de forma descendente desde un valor concreto hasta 0, de 1 en 1.

Idiomas compatibles: Diagrama de bloques de funciones, diagrama de lógica de escalera y texto estructurado.

Esta instrucción se aplica a los controladores Micro810, Micro820, Micro830, Micro850 y Micro870.



			Descripción
C U	E n t r a d a	B O O L	Cierto: flanco ascendente detectado, recuento ascendente.
C D	E n t r a d a	B O O L	Cierto: flanco ascendente detectado, recuento descendente.
R E S E T	E n t r a d a	B O O L	Restablecimiento de comando dominante. (CV = 0 si RESET es CIERTO).
L O A D	E n t r a d a	B O O L	Comando de carga. Cierto: si CV = PV.
P V	E n t r a d a	D I N T	Valor máximo programado.
Q U	S a l i d a	B O O L	Desbordamiento. Cierto: si CV >= PV.
Q D	S a l i d a	B O O L	Caudal insuficiente. Cierto: si CV <= 0.
C V	S a l i d a	D I N T	Resultado del contador.

Conclusiones

Sin duda los avances tecnológicos y la automatización van cambiando la forma de hacer las cosas en todos los ámbitos y la industria no podría quedar exenta de ello.

Hoy en día, para que una empresa se mantenga competitiva o busque serlo en un futuro deberá implementar medidas de transformación que incluyan la automatización de procesos y el uso de las comunicaciones para agilizar los tiempos de respuesta y optimizar recursos.

Este curso teórico-práctico será una base para reconocer que es el PLC, como utilizarlo, cuáles son sus terminologías, las características de sus sistemas de control, aplicación y los diferentes tipos de redes de comunicación que abarca.

Al término del curso y conociendo las bases del uso del PLC de la industria 4.0 el participante podrá acceder a nuevas oportunidades de crecimiento tanto personales como laborales.

Bibliografía

- ✓ Manual del proveedor (2019) Rockwell Automation
- ✓ <https://www.rockwellautomation.com/es-pr.html>