PROGRAMACIÓN PLC SIEMENS II

Operación con bits

Instrucciones de temporización

Contadores

Instrucciones matemáticas

Movimiento de datos

OPERACIÓN CON BITS

1. Lectura de bits

La lectura de un valor binario (*bool*) de una posición de memoria o de una entrada digital se realiza mediante contactos normalmente abiertos (NO) o cerrados (NC):

NO:

- Si el valor de la variable es *true* (1) deja pasar el flujo de ejecución.
- Si el valor de la variable es *false* (0) **no** deja pasar el flujo de ejecución.

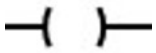
NC:

- Si el valor de la variable es *true* (1) **no** deja pasar el flujo de ejecución.
- Si el valor de la variable es *false* (0) deja pasar el flujo de ejecución.

2. Escritura de bits

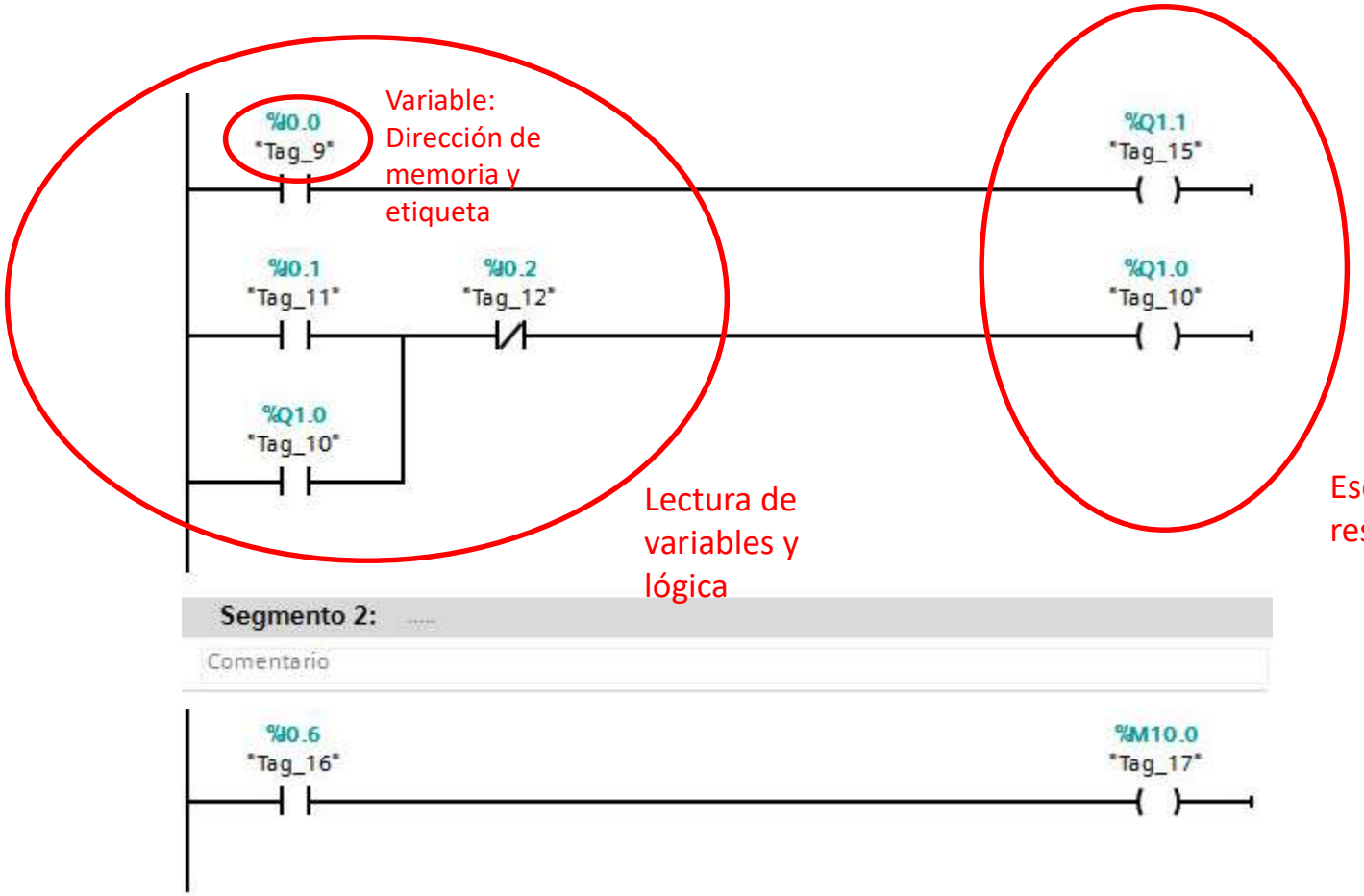
La escritura de un valor binario (*bool*) de una posición de memoria o de una salida digital se realiza mediante bobinas.

- Siempre se colocan al final de una línea de ejecución (o circuito), como resultado de la lógica de esa línea



OPERACIÓN CON BITS

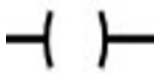
Por ejemplo,



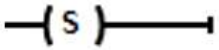
OPERACIÓN CON BITS

Las escrituras pueden realizarse mediante:

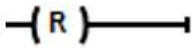
- **bobinas:** el valor de salida se obtiene de la lógica previa



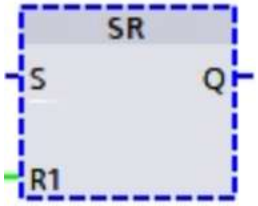
- **bobinas con enclavamiento:** el valor de salida se obtiene de la lógica previa y queda fijado.
  Set (S): la posición de memoria asignada queda fijado a *true* (‘1’)



Reset (R): la posición de memoria asignada se resetea, queda fijado a *false* (‘0’)

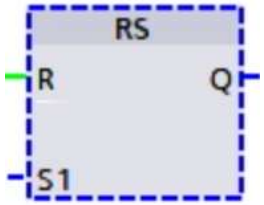


- **bloques biestables RS y SR:** asignación del valor mediante bloques SET-RESET, RESET-SET



- Entrada SET (S) a *true,* la salida Q queda enclavada a *true*
- Entrada RESET (R1) tiene valor *true* la salida Q queda enclavada a *false*.
- Entrada S y R1 a *true* simultáneamente, la salida Q quedará a *false.* **La entrada R tiene prioridad respecto entrada S**

OPERACIÓN CON BITS



- Entrada RESET (R) a *true,* la salida Q queda enclavada a *false*
- Entrada SET (S1) tiene valor *true* la salida Q queda enclavada a *true*.
- Entrada R y S1 a *true* simultáneamente, la salida Q quedará a *true.* **La entrada S1 tiene prioridad respecto entrada R**

Por ejemplo,

Si M1.0 tiene el valor *true*, la salida Q136.0 queda enclavada a *true*
Si M1.1 tiene el valor *true* la salida Q136.0 queda enclavada a *false*

OPERACIÓN CON BITS

3. Detección de flancos

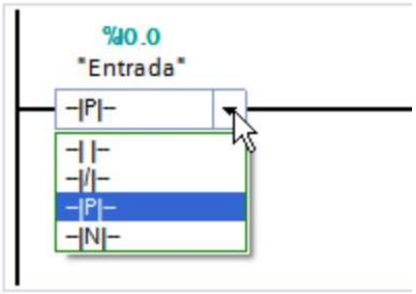
Un flanco es la transición de un bit cuando cambia de valor:

- Flanco de subida (o positivo P): si el bit cambia de *false* (‘0’) a *true* (‘1’)
- Flanco de bajada (o negativo N): si el bit cambia de *true* (‘1’) a *false* (‘0’)

Para detectar el cambio es necesario memorizar el estado actual de la variable para poder compararlo con el valor de la variable en el siguiente ciclo *scan.* Se utilizan contactos detectores de flanco:

Valor actual de “Entrada”

Valor de “Entrada” en el *scan* anterior



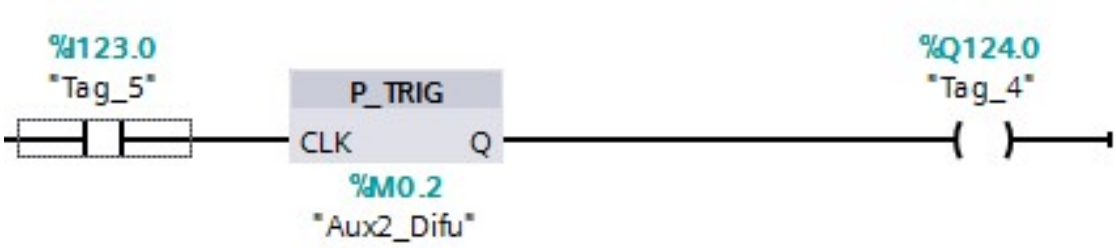
Cuando el valor de las variables es diferente el contacto se cierra durante un *scan,* es decir, se genera un pulso de una duración de un 1 *scan:*

- Tipo P: se genera el pulso en flancos de subida
- Tipo N: se genera el pulso en flancos de bajada

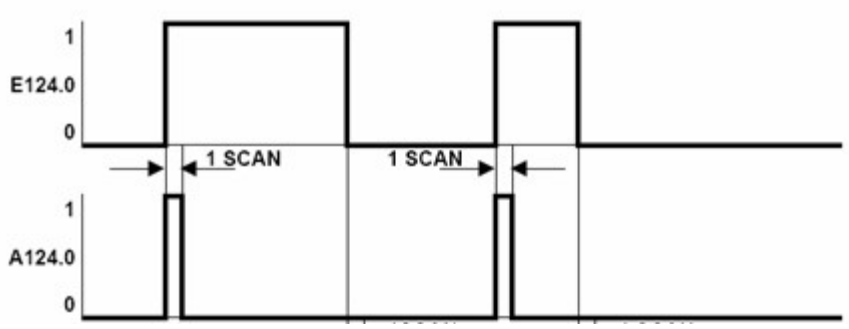
OPERACIÓN CON BITS

Otra forma de detectar el flanco es mediante el bloque *trigger.* El funcionamiento es idéntico a los contactos detectores de flanco: P_TRIG, N_TRIG

Por ejemplo,



cuando E124.0 pasa de *false* a *true*, A124.0 tendrá el valor *true* durante un solo ciclo de *scan*



INTRUCCIONES DE TEMPORIZACIÓN

1. Temporización a la conexión (TON)

La temporización se realiza mediante el siguiente bloque:

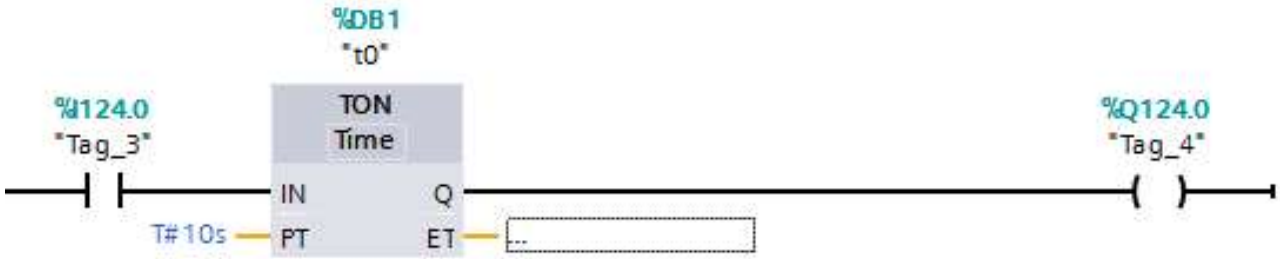
El temporizador siempre tiene asignado variables de tipo DB, por ejemplo IEC_Timer_0_DB. **Cada temporizador debe tener su variable única.**

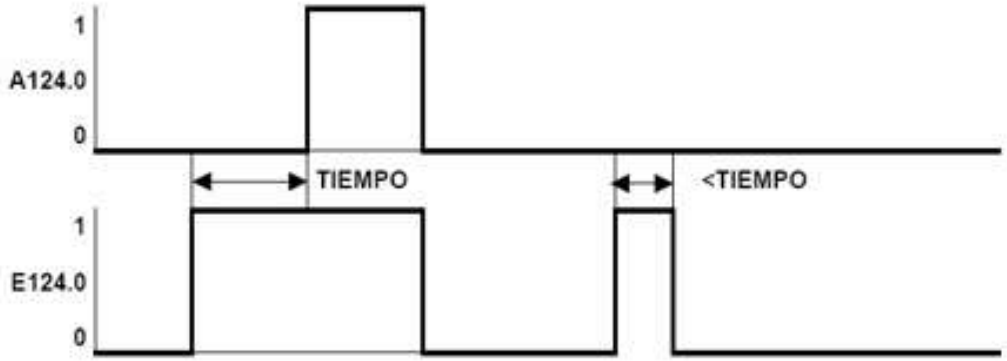
- **IN:** Entrada, activación del *timer*
- **PT:** tiempo de conexión (**indicar unidades: s, ms**): # + Tiempo + Unidad de Tiempo'
  Por ejemplo: t#100ms para 100 ms
- **Q:** Salida, se activa cuando se haya cumplido el tiempo
- **ET:** Tiempo transcurrido

1- La entrada (IN) toma el valor *true* y se mantiene
2- El *timer* detecta el flanco de subida de la entrada (IN) y pone en marcha el temporizador
3- Espera el tiempo configurado (PT): en ET se muestra el valor actual del temporizador (si hay una variable asignada)
4- Transcurrido el tiempo PT, se activa la señal de salida (Q) a *true*
5- La salida queda activada hasta que la entrada tome el valor *false*

INTRUCCIONES DE TEMPORIZACIÓN

Por ejemplo,





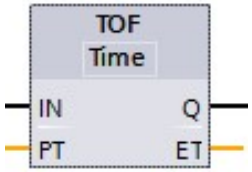
SALIDA

ENTRADA

INTRUCCIONES DE TEMPORIZACIÓN

2. Temporización a la desconexión (TOF)

La temporización se realiza mediante el siguiente bloque:



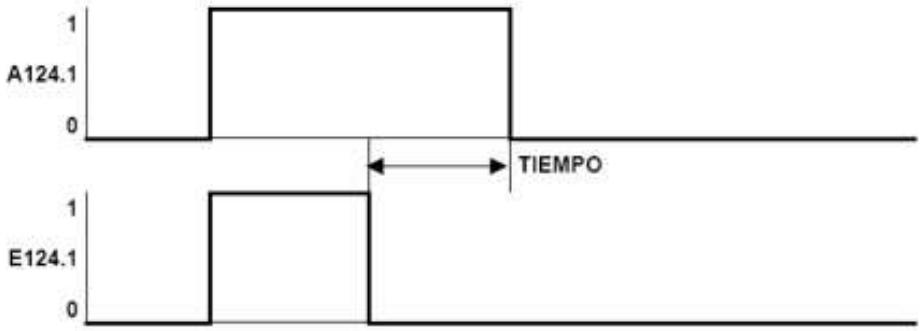
El temporizador siempre tiene asignado variables de tipo DB, por ejemplo IEC_Timer_0_DB. **Cada temporizador debe tener su variable única.**

- **IN:** Entrada, activación del *timer* a la desconexión
- **PT:** tiempo de desconexión (**indicar unidades: s, ms**): # + Tiempo + Unidad de Tiempo'
  Por ejemplo: t#100ms para 100 ms
- **Q:** Salida, se desactiva cuando se haya cumplido el tiempo
- **ET:** Tiempo transcurrido

1- La entrada (IN) toma el valor *true* y se mantiene
2- El *timer* detecta el flanco de subida de la entrada (IN) y activa la salida Q a *true* y se mantiene
3- La entrada (IN) toma el valor *false* y activa el temporizador
4- Espera el tiempo configurado (PT): en ET se muestra el valor actual del temporizador (si hay una variable asignada)
5- Transcurrido el tiempo PT, se desactiva la señal de salida (Q) a *false*

INTRUCCIONES DE TEMPORIZACIÓN

Por ejemplo,



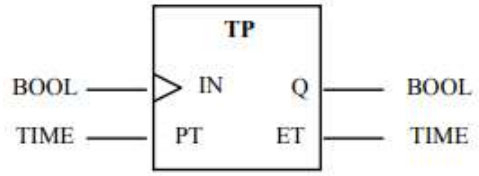
SALIDA

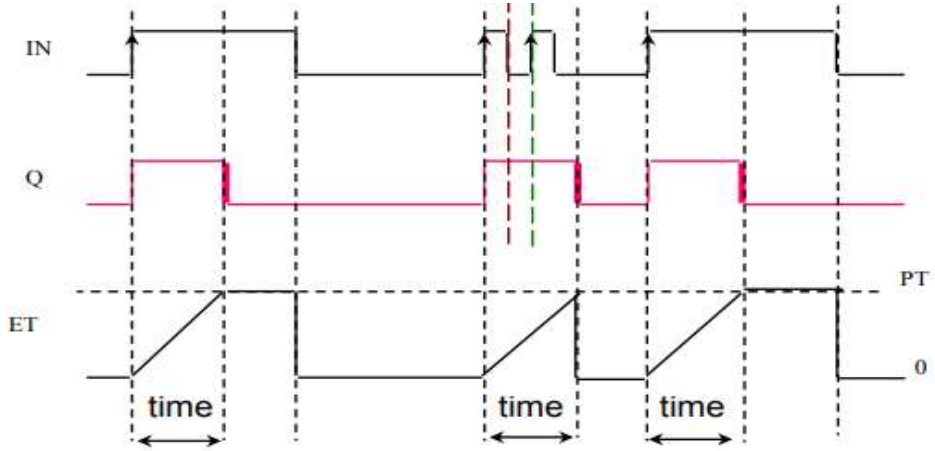
ENTRADA

INTRUCCIONES DE TEMPORIZACIÓN

3. Temporización de pulso (TP)

La temporización se realiza mediante el siguiente bloque:





El temporizador activa la salida en el flanco de subida de la entrada. Hasta que no finaliza el tiempo programado no se puede reactivar.

Al activarse la entrada (IN) se activa la salida (Q) durante un determinado tiempo. Al finalizar este tiempo la salida se desactiva.

INTRUCCIONES DE TEMPORIZACIÓN

4. Bits temporizados y marcas de sistema

En PLC SIEMENS es posible asignar un byte de memoria para activar bits temporizados. Estos bits generan trenes de pulsos a diferentes frecuencias. Para activar estos bits:
En el menú contextual de la PLC:

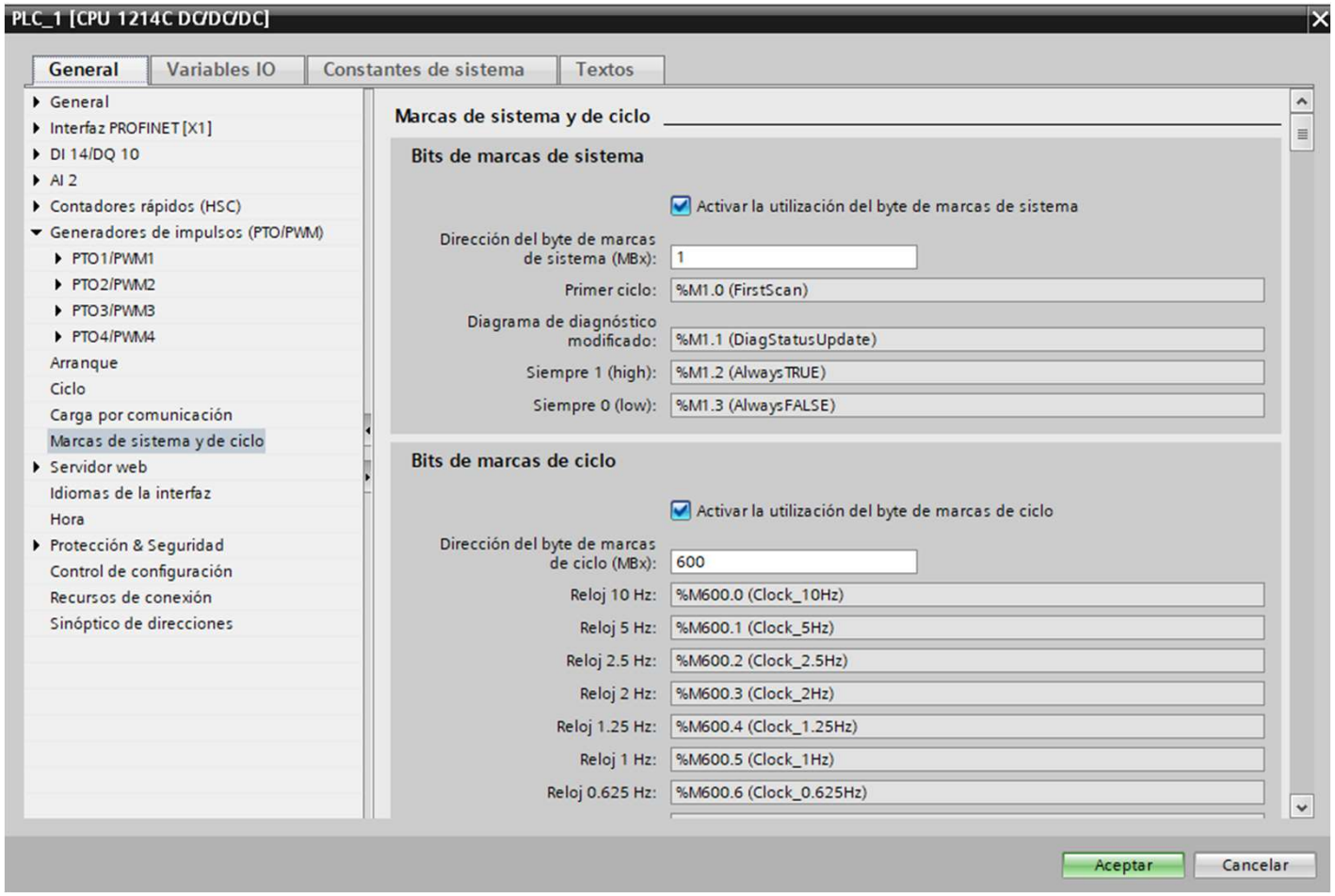
"Propiedades" > "Marcas de sistema y de ciclo" > "Bits de marcas de ciclo"

- Activar *"Bits de marcas de ciclo"*
- Asignar una dirección de memoria. Hay que tener en cuenta que se reserva todo el byte, por tanto, debe estar libre.

"Propiedades" > "Marcas de sistema y de ciclo" > "Marcas de sistema"

- Activar *"Bits de marcas de sistema",* para indicar el inicio de primer *scan*
- Asignar una dirección de memoria. Hay que tener en cuenta que se reserva todo el byte, por tanto, debe estar libre.

INTRUCCIONES DE TEMPORIZACIÓN



En este caso, en el byte %MB600 se ha activado los bits de ciclo, en cada bit se genera un tren a una determinada frecuencia (ya fijada)

INTRUCCIONES DE TEMPORIZACIÓN

Cada bit en la memoria asignada generará un tren de pulso con una determinada frecuencia:

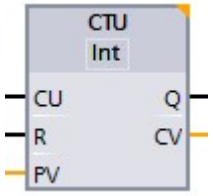
	Clock_Byte	Byte	%MB800
	Clock_10Hz	Bool	%M800.0
	Clock_5Hz	Bool	%M800.1
	Clock_2.5Hz	Bool	%M800.2
	Clock_2Hz	Bool	%M800.3
	Clock_1.25Hz	Bool	%M800.4
	Clock_1Hz	Bool	%M800.5
	Clock_0.625Hz	Bool	%M800.6
	Clock_0.5Hz	Bool	%M800.7

Por ejemplo, generación de un tren de pulsos de 0.5Hz (2s segundos de periodo, 1s de pulso)

INTRUCCIONES DE CONTADOR

1. Contador ascendente (CTU)

Se realiza mediante el siguiente bloque:



El contador siempre tiene asignado variables de tipo DB, por ejemplo IEC_Counter_0_DB.
Cada contador debe tener su variable única.

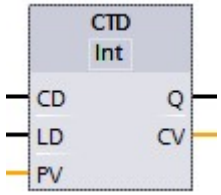
- **CU:** Entrada, incrementa el contador en cada flanco de subida
- **R:** Resetea el contador, lo reinicia a 0
- **PV:** Consigna, valor del contador para activar la salida
- **Q:** Salida, se activa a *true* cuando el contador haya alcanzado la consigna
- **CV:** Valor actual del contador

1- En la entrada (IN) se detecta un flanco de subida
2- Se incrementa el contador
3- Si el contador (CV) es igual a la consigna (PV) se activa Q
4- Si *reset* (R) toma el valor *true* se resetea el contador (pasa a valer 0)
5- Salto a punto 1

INTRUCCIONES DE CONTADOR

2. Contador descendente (CTD)

Se realiza mediante el siguiente bloque:



El contador siempre tiene asignado variables de tipo DB, por ejemplo IEC_Counter_0_DB.
Cada contador debe tener su variable única.

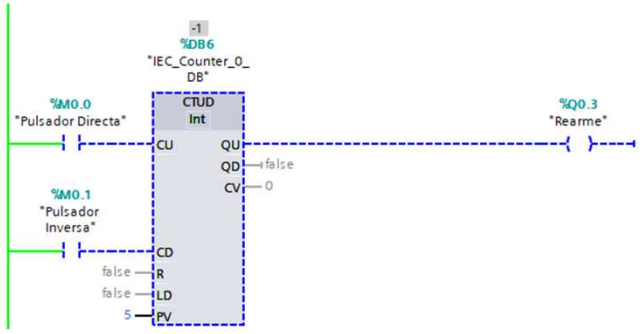
- **CD:** Entrada, decrementa el contador en cada flanco de subida
- **LD:** Resetea el contador, lo reinicia a PV
- **PV:** Consigna, valor del contador inicial
- **Q:** Salida, se activa a *true* cuando el contador tome el valor cero
- **CV:** Valor actual del contador

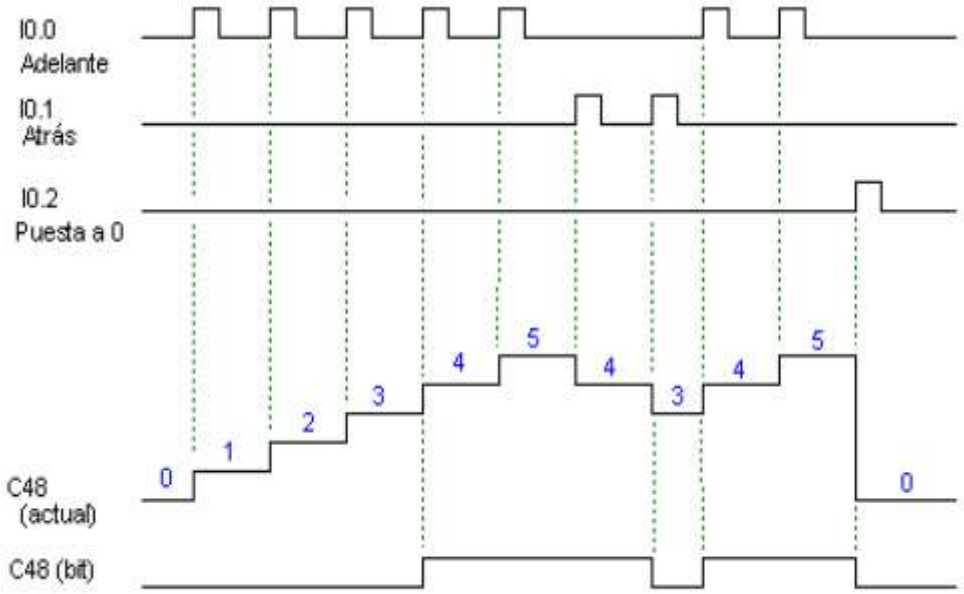
1- En la entrada (IN) se detecta un flanco de subida
2- Se decrementa el contador
3- Si el contador (CV) es cero se activa Q
4- Si *reset* (LD) toma el valor *true* se resetea el contador (pasa a valer PV)
5- Salto a punto 1

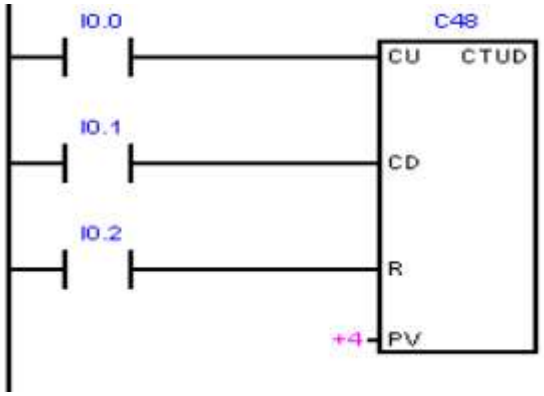
INTRUCCIONES DE CONTADOR

3. Contador Ascendente-Descendente (CTUD)

Combina un contador ascendente y descendente en el mismo bloque.

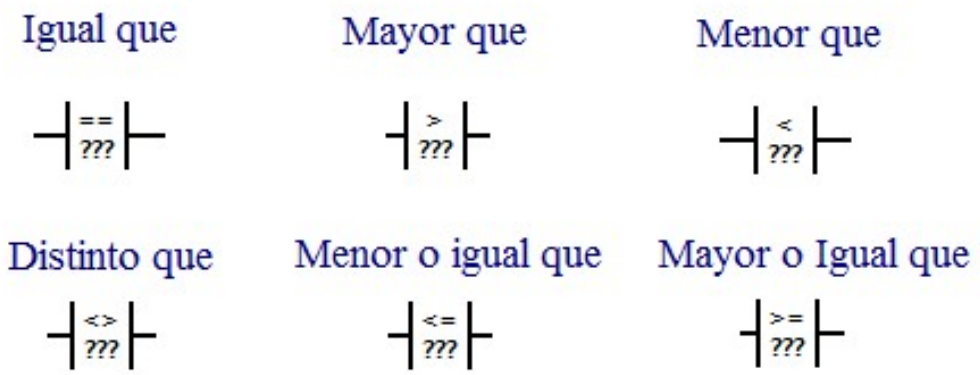




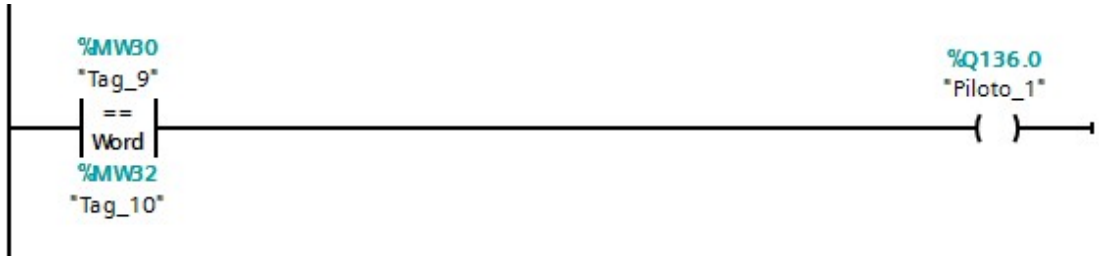


INTRUCCIONES DE COMPARACIÓN

Las instrucciones de comparación son contactos que se abre o cierran dependiendo del resultado de la comparación entre dos valores numéricos. Esto valores pueden ser números o variables.
Las instrucciones de comparación son:



Por ejemplo,



Cuando la variable “Tag_9” es igual a “Tag_10”, la salida “Piloto_1” toma el valor *true,* si no toma el valor *false*

INTRUCCIONES MATEMÁTICAS

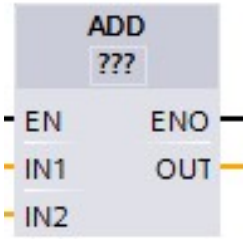
Las instrucciones matemáticas son bloques que realizan determinadas operaciones matemáticas sobre los valores de entrada. Se debe especificar el tipo de variable numérica o de número con el que se va a operar. **Los tipos en las entradas y en las salidas deben de ser los mismos**

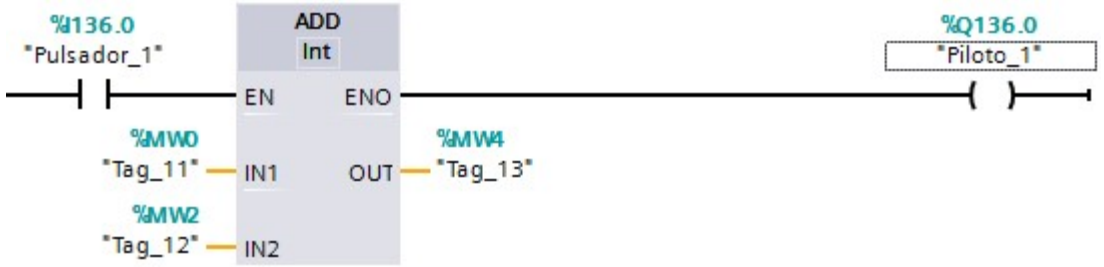
Funciones matemáticas	
ADD	Sumar
SUB	Restar
MUL	Multiplicar
DIV	Dividir
MOD	Obtener resto de división
NEG	Generar complemento a dos
ABS	Calcular valor absoluto
MIN	Determinar mínimo
MAX	Determinar máximo
LIMIT	Ajustar valor límite
SQR	Calcular cuadrado
SQRT	Calcular raíz cuadrada
LN	Calcular logaritmo natural
EXP	Calcular valor exponencial
SIN	Calcular valor de seno
COS	Calcular valor de coseno
TAN	Calcular valor de tangente
ASIN	Calcular valor de arcoseno
ACOS	Calcular valor de arcocoseno
ATAN	Calcular valor de arcotangente

INTRUCCIONES MATEMÁTICAS

Por ejemplo, la instrucción ADD (suma):

- EN: Entrada de activación. Si su valor es *true* se ejecuta la operación.
- IN1: Operando 1
- IN2: Operando 2
- OUT: Resultado de la operación IN1 + IN2
- ENO: Salida a *true* cuando se ejecuta la operación.
- ???: Se debe indicar el tipo de datos con el que se está operando. Se despliega la lista de tipos de datos numéricos
- El símbolo en forma de estrella permite añadir nuevos operandos.

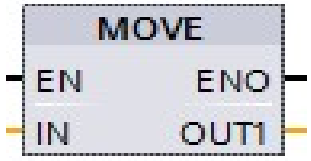




MOVIMIENTO DE DATOS

Instrucción MOVE:

Permite mover un dato de IN a una dirección de memoria OUT1. El dato de IN1 puede ser un valor o bien otra dirección de memoria, por ejemplo:



En el primer ejemplo se mueve el numero decimal 10 a la palabra MW10, y en el segundo se mueve el contenido de la palabra MW0 a la MW10