

Resumen parte diseño automatización industrial

1) Sensores

- a. Presencia: Solo detecta si hay un elemento o no
 - i. Con contacto: finales de carrera mecánicos, siempre son interruptores pudiendo ser capacitivos o inductivos, en el argot **Micro de contacto**
 - ii. Sin contacto
 - 1. Inductivos: solo se pueden aplicar en metales y funcionan detectando la variación del campo magnético por la presencia de un objeto
 - 2. Capacitivos: tienen mayor campo de aplicación ya que permite detectar mas variedad de elementos. Además, se puede emplear para detectar nivel en líquidos y humedad
 - 3. Efecto Hall: detecta la interacción de la corriente con un campo magnético permanente que desvía las cargas mediante fuerza de Lorentz generándose de esta manera una diferencia de potencial. Muy baratos de fabricar (medidas rpm)
 - 4. Células Reed: Cuando los contactos están normalmente abiertos se cierran en la presencia de un campo magnético; cuando están normalmente cerrados se abren en presencia de un campo magnético.
 - 5. Barreras ópticas: detectan si un elemento interrumpe un haz de luz. Tiene problemas con ambientes sucios (partículas en suspensión)
 - 6. Ultrasonidos: detectar cuanto tarda en volver un pulso de sonido, puede usarse para medir nivel de líquidos. Nunca se usan solos por pérdidas y refracción
- b. Posición/desplazamiento: Medir distancia
 - i. Con contacto
 - 1. Potenciómetros resistivos lineales
 - 2. Reglas de contactos
 - ii. Sin contacto
 - 1. Encoders lineales ópticos (reglas ópticas). Medir mediante un haz de luz marcas que se desplazan
 - 2. LVDT: medir mediante un ferromagnético que se desplaza la inducción entre bobinas. Tiene rozamiento muy bajo
 - 3. Inductosyn lineal: dos partes acopladas magnéticamente similar a un resolver (encoder analógico) donde el deslizador se mueve a lo largo de una regla y se mide como varían las corrientes inducidas
 - 4. Láser:
 - a. Triangulación, se mide entre mm a decenas de m. La precisión no es uniforme en el rango de la medida. Va midiendo, usando varios rayos.
 - b. Tiempo de propagación: medida de tiempo de vuelo para distancias grandes >50 m y de cambio de fase para distancias más cortas

- c. Giros y rotaciones
 - i. Contacto: Potenciómetros resistivos
 - ii. Sin contacto
 - 1. Potenciómetros de efecto hall
 - 2. Encoder incremental: para diferenciar el sentido de giro se ponen dos franjas de marcas de manera que los trenes de pulsos estén desfasados 90° y existe una marca de cero para contar vueltas. Tienen gran precisión, pero son sensibles a perder la cuenta o a perturbaciones.
 - 3. Encoder absoluto: proporcionan una salida digital absoluta correspondiente a la posición absoluta del elemento móvil. Se basa en el mismo principio que los incrementales. Es inmune a las interferencias y no necesita punto de referencia, pero los cabezales son muy complejos
 - 4. Resolvers y sincroresolvers: se mide la respuesta en el estator ante rotaciones en el rotor
 - d. Medida de velocidad: mejorar el comportamiento dinámico de los actuadores, se puede obtener la velocidad a partir del tiempo en que tarda en adquirir una posición determinada
 - i. Tacómetros o dinamos: mide el campo magnético y crea señal
 - ii. Convertidores
 - iii. Sensores de efecto hall
 - e. Aceleración: acelerómetros
 - f. Presión: sensores de medida directa o indirecta mediante diafragma
 - g. Temperatura: RTD, PTC, termopar
 - h. Fuerza y par: pueden usarse para percibir la forma o posición de objetos midiendo el esfuerzo ejercido en la superficie de contacto
 - i. Galgas extensiométricas: varía su resistencia ante la deformación
 - i. Sensores de nivel
 - i. Ultrasonidos
 - ii. Presión diferencial
 - iii. Variación de capacidad
 - iv. Variación de conductividad
 - v. Flotador
- 2) **Actuadores:** dispositivo que usa energía para realizar un efecto sobre un proceso automatizado
- a. Eléctricos
 - i. Motores de continua: hasta miles de kW
 - 1. Es fácil de regular, modelo sencillo muy similar al real (2º orden estable). Tanto en velocidad como en par.
 - 2. Puede dar par a 0 rpm
 - 3. Tiene el problema de las escobillas, rendimiento peor que en síncronos (están siendo sustituidos por estos)
 - 4. Aplicaciones
 - a. Regulación de velocidad
 - b. Precisión
 - c. Regulación de par y par a 0 rpm
 - d. Altas potencias

ii. Motores de alterna

1. Asíncronos

a. Jaula de ardilla centenares de kW

- i. Muy ligeros y buena respuesta dinámica, bajo tiempo de subida y buen ancho de banda
- ii. Problema arranque (contactores y sobreintensidades)
- iii. Coste bajo de mantenimiento, pero elevado coste de variador para que funcione a distintas velocidades

b. Rotor bobinado: centenares de kW

- i. Arranque progresivo, resistencias de rotor
- ii. Regulación de velocidad por deslizamiento
- iii. Coste alto de variador

c. Aplicaciones

- i. Accionamientos directos
- ii. Variación de velocidad a baja potencia y poca precisión
- iii. Aplicaciones sin regulación
- iv. Maniobras de elevación

2. Síncronos

- a. Velocidad depende de la frecuencia (buena regulación)
- b. No hay deslizamiento entre el vector deslizamiento y el de giro
- c. Alto rendimiento
- d. Maniobra de arranque crítica (problemas de vibraciones). Se necesitan variadores de frecuencia

iii. Motores paso a paso: potencia y velocidades bajas

- 1. No hace falta realimentación al conocerse su posición y por ello, trabajan en lado abierto
- 2. No dan mucho par → se saltan pasos. Por ello, se meten reductoras
- 3. Se emplea para mover cosas pues tiene poca inercia

iv. Servomotores (brushless): capacidad de regulación en posición y velocidad. Potencias pequeñas

1. DC o brushless AC (sin escobillas)

- a. Los brushless DC producen mayor rizado, pero requieren electrónica mas barata para su control y no necesitan encoders absolutos
- 2. Son rápidos (buena respuesta) -> elevada respuesta dinámica
- 3. Gran precisión
- 4. Estables en velocidad y par
- 5. Aplicaciones
 - a. Robótica
 - b. Radio control
 - c. Control posición

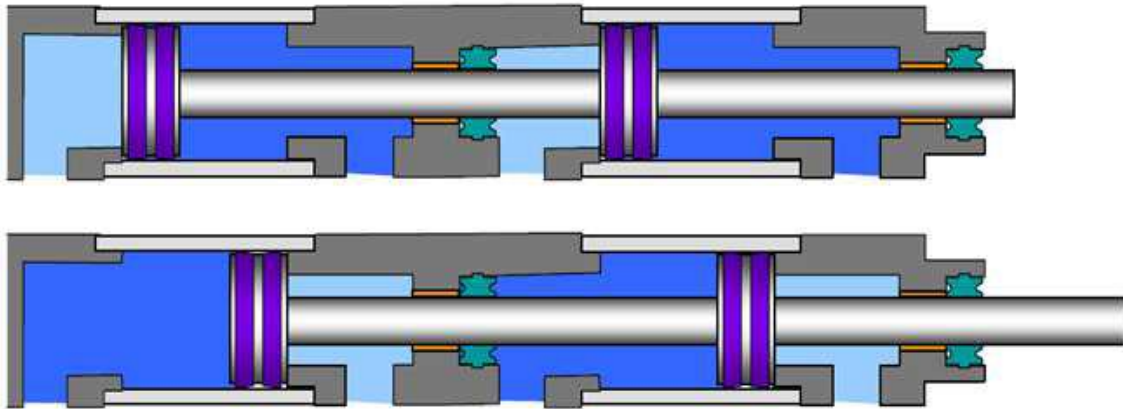
MOTOR + EQUIPO	COMPRA	INSTALACIÓN	MANTENIMIENTO
DC	ALTO	MEDIO	ALTO
CA (JAULA)	BAJO	BAJO	MUY BAJO
CA (ROTOR BOBINADO)	ALTO	MEDIO	ALTO
PASO A PASO	BAJO	BAJO	BAJO
SERVO	ALTO	MEDIO	ALTO

	AC Jaula	AC Rot.Bob.	Servo	Paso a paso	Continua
Coste del motor	Bajo	Elevado	Elevado	Bajo	Elevado
Arranque directo de red	Estándar	Arranque especial	-	-	-
Variación velocidad	Fácil	Posible	Siempre	Siempre	Siempre
Coste Motor + variador	Cada vez menor	Económico	Bastante económico	Muy económico	Muy económico
Prestaciones	Cada vez mayores	Medias	Muy elevadas	Medias	Elevadas
Uso	Universal	En disminución	Máquinas herramienta	Posicionamiento en lazo abierto. Pequeñas potencias	En disminución

b. Actuadores fluidos:

- i. Neumáticos: actuaciones todo-nada donde no se requieran elevadas precisiones
 1. Líquido compresible, el único modelo manejable es el de gas ideal, pero no funciona bien. Difícil manejo posición.
- ii. Hidráulicos: cuando se requiere maximizar la potencia/volumen o potencia/masa
 1. Líquido incompresible. Problemas seguridad en caso de fuga por las presiones tan altas manejadas
- iii. Hacer vacío: se aprovecha el efecto Venturi donde un fluido a alta velocidad arrastra fluidos desde cavidades.
- iv. Actuadores lineales
 1. Simple efecto
 - a. Desarrolla trabajo en un solo sentido, se emplean para sujetar, marcar o expulsar
 - b. Tienen menores consumos que los de doble efecto, pero tiene una longitud de carrera limitada debido a los resortes
 2. Doble efecto
 - a. Realizan trabajo de avance y de retroceso
 - b. No se pierde fuerza en el accionamiento debido a la inexistencia de muelle en oposición

- c. Para una misma longitud la carrera es mayor en el doble efecto
- d. Se pueden usar con amortiguación fija (cargas ligeras) o regulable (altas cargas, ralentiza poco a poco)
- e. Cilindros tándem: dos cilindros de doble efecto que forman una unidad, de esta forma se puede obtener el doble de fuerza

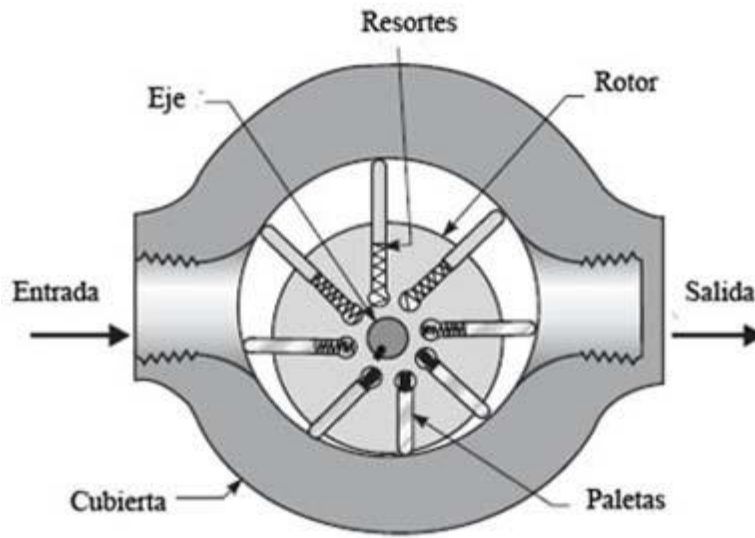


- f. Doble vástago, permite recorrido hacia ambos lados (también llamado cilindro compensado)
3. De impacto
 - a. Cilindro que se mueve a velocidad elevada, se emplea en trabajos de marcado
 - b. Se basa en cambiar rápidamente la superficie donde se aplica la presión para tener un impacto repentino
 4. Sin vástago
 - a. El movimiento del cilindro esta contenido en el propio cuerpo y se transmite mediante un carro exterior que se desplaza. Se suele usar para trabajar a través de líneas transportadoras
 5. Guiados
 - a. Se impide el giro de los cilindros empleando pistones auxiliares a costa de tener que emplear mayores fuerzas

v. Actuadores de giro

1. Motor de paletas

- a. Para caudales altos, no permiten presiones altas porque el contacto se da mediante capa de lubricante. Se emplea en aplicaciones que no necesiten pares elevados (neumática)



2. Motor de engranajes

- a. El par de rotación es generado por la presión que ejerce el aire sobre los dientes de los piñones engranados
- b. Se utilizan para altas potencias y presiones elevadas
- c. Tienen sentido de rotación reversible

3. Bomba de engranajes

- a. Se sube la presión de fluidos mediante el uso de engranajes, entra a menor presión de la que sale

4. Pistones radiales y axiales

- a. Suben la presión
- b. Se transforma un movimiento circular a uno lineal

	Neumático	Hidráulico	Eléctrico
Energía	Aire a presión (5-10 bar)	Aceite mineral (50-100 bar)	Corriente eléctrica
Opciones	Cilindros Motor de paletas Motor de pistón	Cilindros Motor de paletas Motor pistones ax.	Corriente continua Corriente alterna Motor paso a paso
Ventajas	Neumático Baratos Rápidos Sencillos Robustos	Hidráulico Rápidos Alta relación potencia-peso Autolubricantes Alta capacidad de carga Estabilidad a cargas estáticas	Eléctrico Precisos Fiabiles Fácil control Sencilla instalación Silenciosos
Desventajas	Dificultad de control continuo Instalac. especial Ruidoso	Difícil mantenimiento Instalación especial Frecuentes fugas Caros	Potencia limitada

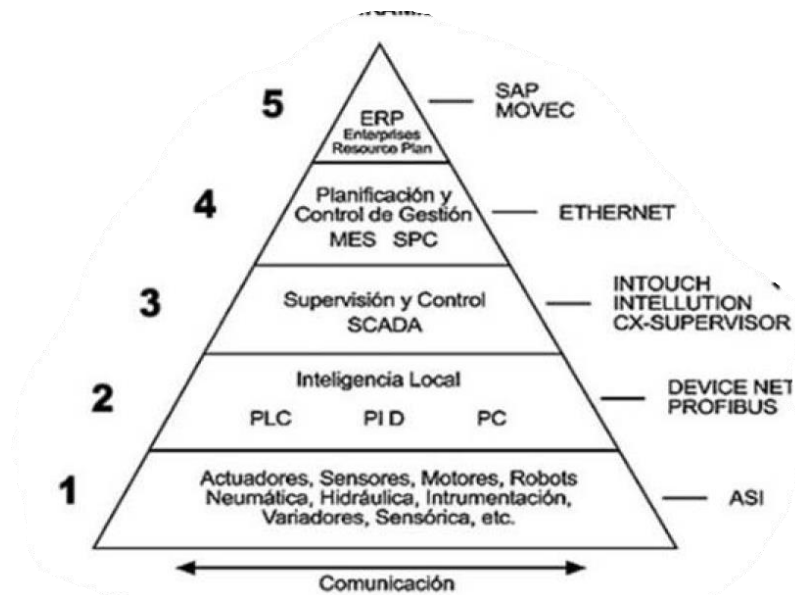
- c. Transmisiones: reducen momento de inercia y transforman movimiento
 - i. Reductoras: adaptan par y velocidad del actuador a los necesarios. Tienen bajo peso, tamaño y rozamiento
 - ii. Harmonic-Drive: un rodamiento elíptico donde el vaso adquiere la forma del rodamiento y se obtienen pares muy elevados con una relación par/volumen alta y rendimiento alto. Al ir moviéndose se deforma lentamente la elipse cambiando de diente.
- d. Brazos robóticos: un robot es la unión de eslabones con articulaciones de rotación (R) y traslación (P)
 - i. Un sistema de referencia por eslabón que se compone posteriormente para un observador fijo. En R^3 hay 3 grados de libertad (x,y,z) y 3 angulares. Por ello, cada articulación normalmente fija uno de ellos. Y para cambiar de coordenadas debe hacerse cinemática inversa y directa.
 - ii. Características
 - 1. Grados de libertad
 - a. RPP: Cilíndrico
 - b. RRR: Esférico: orientar, hacer pallets (a veces 4 grados)
 - c. RRP: SCARA: característica fundamental inserciones por su capacidad de flexar. Muy usado como pick and place y ensamblaje

- d. PPP: Cartesiano: pintar planos, plotters
 - e. RRRRRR: antropomórfico, capaces de maniobras muy rebuscadas
- 2. Alcance: dimensiones espacio de trabajo
- 3. Capacidad de carga: la velocidad baja con la masa, los SCARA son más rápidos
- 4. Programación
 - a. Guiado: programar haciendo que aprenda los movimientos. No siempre se sabe la trayectoria
 - b. Textual
- 5. Tipo de trayectoria
 - a. Punto a punto (da igual como se mueva)
 - b. Lineal: usan DSP de altas prestaciones, problema con robot antropomórfico y se realiza la aproximación mediante splines cúbicas
 - c. Punto paso: como punto a punto, pero con tolerancias
- 6. Repetibilidad: valor absoluto del radio del círculo que engloba a todos los puntos
- 7. Redundante: más grados de libertad que los necesarios para una tarea (evitar obstáculos)

3) Diseño de procesos

a. Niveles de automatización

- i. Operación manual: sin sistemas automáticos
- ii. Mecanizado: la máquina realiza la operación controlada por el ser humano
- iii. Automatización parcial: la maquina realiza una secuencia limitada y necesita intervención humana para completar la tarea
- iv. Automatización total: la máquina es autónoma y el ser humano solo supervisa
- v. Integración total: no hay operador humano en la planta y solo monitorea en la distancia



b. Automatización fija:

- i. Un único producto
- ii. Elevado volumen de producción

c. Fabricación por lotes

- i. Cadena lineal de ensamblado
- ii. Estrategia enfocada a maximizar el número de unidades producidas

d. Automatización programada

- i. Bajo volumen de producción
- ii. Encaminado a la variedad mediante software de producción

e. Sistemas mixtos

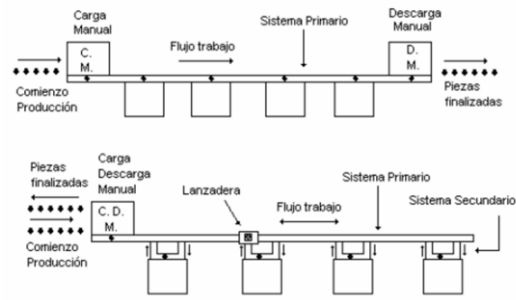
- i. Conseguir variedad de productos sin sacrificar volumen de producción
- ii. Origen de los sistemas de fabricación flexible FMS

f. Sistemas flexibles

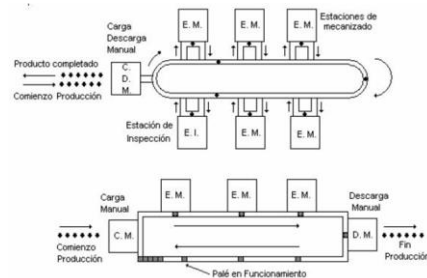
- i. En procesos flexibles nunca se ponen estaciones en mitad del transporte
- ii. Reglas de oro
 1. Si hago foto en planta deben aparecer todos los productos del catalogo

2. Si imagino la planta dos unidades del mismo producto en instantes distintos y grabo un video no tienen por qué ser iguales
- iii. Celdas altamente automatizadas y los elementos son flexibles y versátiles
- iv. Características
 1. Puede realizar distintas piezas de forma simultanea
 2. Puede variar entre distintos tipos de productos
 3. Ningún FMS es completamente flexible
 4. Los productos que se producen son de una familia parecida
- v. Teste de flexibilidad
 1. Test de variedad de productos: ¿Más de un producto que no sea por lotes?
 2. Test cambio programación: ¿Pueden cambiarse cantidades?
 3. Test recuperación errores: ¿Puede recuperarse de fallos?
 4. Test ampliar gama de fabricación: ¿Puede fabricar nuevos elementos?
- vi. Tipos de flexibilidad
 1. Maquinaria: facilidad de una máquina de adaptarse
 2. Producción: rango de diferentes productos
 3. Mezcla de piezas: capacidad para variar el % de productos
 4. Producto: facilidad de introducir modificaciones
 5. Enrutado: facilidad de modificar las secuencias debido a roturas
 6. Volumen: producir en un rango
 7. Expansión: poder aumentar la producción
- vii. Tipos de sistemas
 1. Celda de una única estación: formada por un CNC combinado con almacenamiento. Puede fabricar por lotes o flexibilidad en función del número de productos, aunque tiene una flexibilidad limitada al ser una única máquina que no puede recuperarse de errores
 2. Celda de fabricación flexible: dos o 3 CNC y un transporte conectado a una estación de carga y descarga. Capacidad de almacenamiento limitada
 3. Sistemas de fabricación flexible: varias máquinas conectadas con un transporte común.
- viii. Un FMS tiene celdas que funcionan como diagnóstico y monitorización.
- ix. Nivel de flexibilidad
 1. FMS dedicados: especialización exhaustiva en el proceso y maquinaria. Solo se fabrica una pequeña familia de productos. (menos flexible pero más productivo)
 2. FMS de orden aleatorio: alta variedad de productos que necesita maquinaria de propósito general
- x. Transporte y almacenamiento

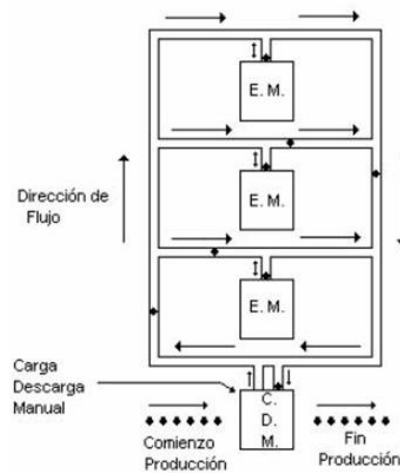
1. Distribución en línea



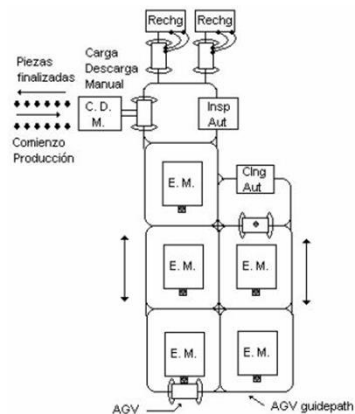
2. Distribución en bucle



3. Escalada



4. Campo abierto (incluir estaciones lineales o poco flexibles)



4) Simulación de procesos

- a. Premisas
 - i. No gemelo digital (no 1 a 1)
 - ii. Se debe simplificar (nivel de abstracción)
 - iii. Extrapolar resultados al diseño
- b. Criterios optimización
 - i. Detectar problemas: cambiar numero de recursos, capacidades o flujos de entrada. No se cambian características de las máquinas ni de las cintas
 - 1. Cuello de botella: Puntos de mucha ocupación frente a otros con poca → Añadir máquinas
 - 2. Bloqueos máquinas: Se suele dar por asincronía en el proceso ya que dos máquinas no se han sincronizado → Añadir Buffers

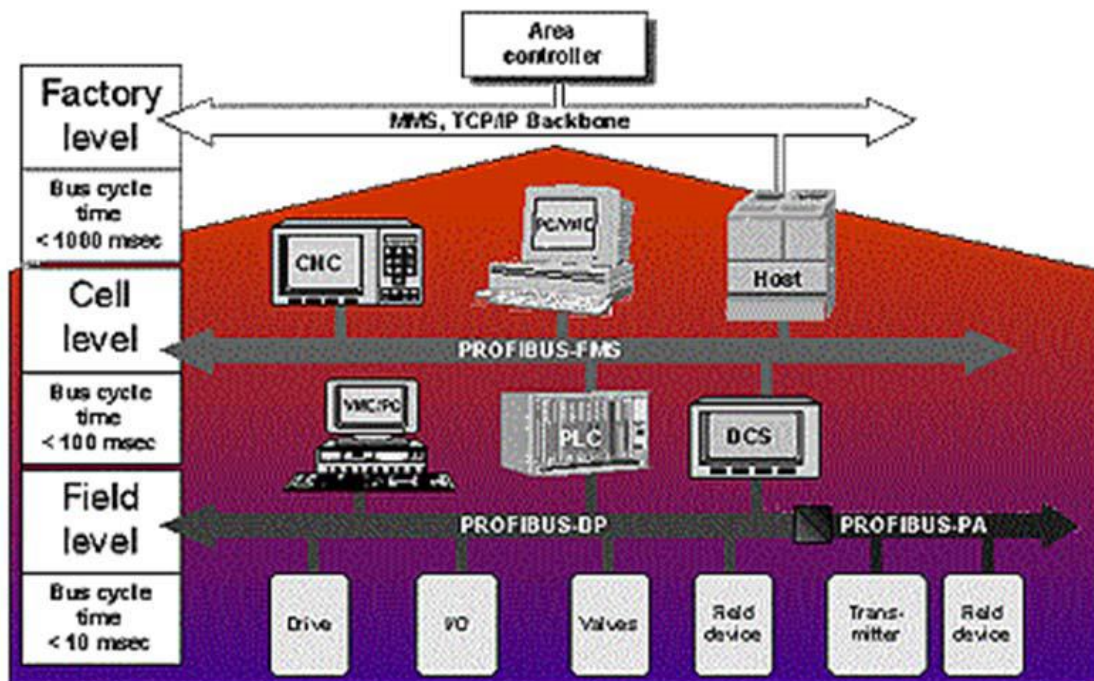
5) Comunicaciones

- a. Red de área extensa WAN: cobertura amplia
- b. Red conmutada, se crea un circuito físico que une dos destinos
- c. Red de conmutación de paquetes: pensada para la comunicación de datos (no existe un camino físico permanente)
- d. Red de área local (LAN): comunicación entre equipos en áreas. Alto volumen y velocidad
- e. Topologías de una LAN
 - i. Estrella: se comunican a través de un dispositivo central por donde pasan todas las comunicaciones. Facilidad de quitar o añadir estaciones. La avería del nodo central afecta a toda la red (no muy habituales)
 - ii. topología en anillo: Cada estación esta conectada a otras dos de forma que forma un anillo y los mensajes viajan de estación por estación en una única dirección. El control lo tienen las estaciones con token que se va pasando entre nodos (es muy costoso). Para caso de fallo se emplean concentradores que si falla un equipo lo cortocircuita.
 - iii. Topología en bus lineal: todos los nodos se encuentran conectados a un único canal. Normalmente no hay token y se basa en la contienda
 - iv. Topología bus anillo: permite ramificar el bus
- f. Banda base: transmisión mediante señales binarias → Fibra óptica
- g. Banda ancha: transmisión mediante portadora Modulada (permite transmisiones simultaneas), requiere un canal diferente para cada emisión y recepción
 - i. AM: amplitud modulada: se envía una onda de alta frecuencia
 - ii. FM frecuencia modulada: info en la frecuencia
- h. Control de acceso
 - i. Testigo: se pasa el token
 - 1. Token ring: cuando una estación recibe el token puede transmitir y el orden de la conexión fija el orden de acceso
 - 2. Token bus: se usa cuando hay estaciones que no pueden transmitir nunca mensajes pero si recibirlos (maestro-esclavo)
 - ii. Sondeo o polling: la estación recibe el turno de escritura desde una controladora

- iii. Contienda: las estaciones compiten entre si por quien tiene el derecho del canal. A pesar de no transmitir si se escucha otro mensaje, a veces por los retardos se dan colisiones que deben resolverse normalmente emitiendo primero un mensaje y luego el otro. El rendimiento depende del numero de nodos.
- i. Ethernet
 - i. Basado en TCP/IP (identificación de cada nodo: __.__.__.__) y para resolverlos se emplean DNS (domain name servers que resuelven los símbolos)
 - ii. Topologia de bus o árbol
 - iii. Velocidad 10 MBps
- j. Componentes de la red
 - i. HUB: ladrón o repetidor: sincroniza y repite las señales de entrada a la salida
 - ii. Switch: Más responsabilidad que HUB pues diferencia por bocas
 - iii. Bridget: 2 nodos que actúan de puente (conectan 2 segmentos y crean una sola red)
 - iv. Gateway: conectar redes de distinta topología, adaptan hardware y software
- k. Buses: todos necesitan token, no hay contienda
 - i. Ethernet no vale en tiempos rápidos y el tiempo en los buses debe ser constante para los lazos de control. El bus debe ser siempre en tiempo real (ser predictivo)
 - ii. No hace falta ancho de banda elevado (pocos datos)
 - iii. Son anillos o token bus
 - iv. Evitar tanto cableado
 - v. Tipos
 - 1. ASI: cualquier topología. Conectar sensores y actuadores a PLC o PC. 4 bits de entrada y salida. 2 hilos sin apantallar y el cable se regenera solo.
 - 2. Bitbus: topología lineal. Usado en robótica
 - 3. Modbus: topología de bus. Más avanzado que anteriores
 - 4. Profibus: topología lineal o anillo. Cada nodo tiene un identificador y es único por bus (mecanismo conmutación paquete). Problema programar comunicaciones con el de flashear identificaciones.
 - 5. Fieldbus: profibus en EEUU

I. Profibus

- i. Distingue entre maestros (determinan la comunicación, estaciones activas) y esclavos (estaciones pasivas). Nivel de enlace. Las estaciones activas se van pasando el token y solo hay un maestro cada vez. Si hubiese más de uno cada uno tiene esclavos distintos.



- ii. FMS: permite nivel jerárquico (nivel 7 OSI), es decir, aplicaciones cliente servidor. Como medio físico usa el RS485 que es un bucle de corriente que permite colocar elementos en paralelo (poner y quitar dispositivos sin interferir)
- iii. DP: distributed peripherals, usa el RS485. Comunicaciones rápidas con las controladoras.
- iv. PA: para ambientes especiales: lleva alimentación en el bus (zonas con peligro de explosión). Cable IEC 1158-2 que permite esta alimentación y los dispositivos actúan como elementos RC que se van cargando (ramas tiempo)
- v. Fibra óptica: aísla galvánicamente no pasa electricidad, sino que se comunica por luz.

Resumen parte programación automatización industrial

Tipos de bloques:

- OB: de organización, llaman al sistema operativo
- FC: funciones simples sin memoria
- FB: funciones complejas con memoria
- DB: bloques de datos

Comando: Las etiquetas para los saltos

ETIQUETA	OPERADOR	OPERANDO	COMENTARIO
_001:	U	E 124.0	// Carga E 124.0 en el RLO

Repaso contadores:

U "Evento" L C#10 S ZI	U "Incre" ZV ZI	U "Dec" ZR ZI	UN ZI = "M _{z=0} "
Carga ZI(ZW)=10	Incremento ZI:=ZI+1	Decremento ZI:=ZI-1	ZI==0?

Flancos: Para flanco positivo FP, para flanco negativo FN

U E 124.0
FP M_Flanco
= M_Pulso

Escritura de datos

#Si son parámetros

"Si son globales"

Para transferir datos (no depende del RLO, hace falta salto lógico) → "Nombredatos".atributo

// Instrucciones equivalentes en AWL

L "DATOS_MODO_A".tiempo_watchdog

T "DATOS_PRODUCCION".tiempo_watchdog

Final de bloque:

- Incondicional BEA
- Condicional BEB

Salto de un mismo bloque:

□ SPA <ETIQUETA>

- Salto incondicional a la instrucción con ETIQUETA
- Equivale a `goto ETIQUETA`

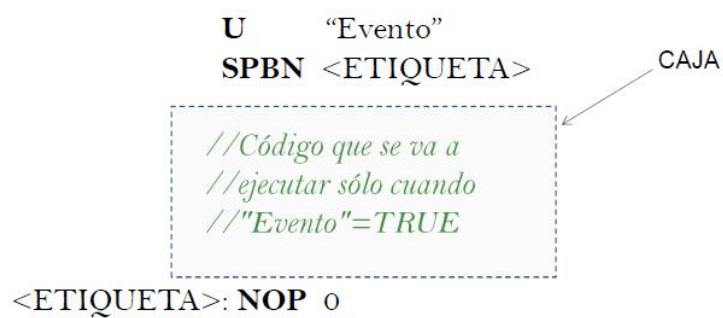
□ SPB <ETIQUETA>

- Salto a la instrucción con ETIQUETA si el RLO = 1
- Equivale a `if (RLO = 1) then goto ETIQUETA`

□ SPBN <ETIQUETA>

- Salto a la instrucción con ETIQUETA si el RLO = 0
- Equivale a `if (RLO = 0) then goto ETIQUETA`

Plantilla de salto



Repaso temporizadores clásicos

IVESA → TP / TP memoria / TON / TON con memoria / TOFF

U evento

L #temporizador

S<tipo> T1

Repaso temporizadores modernos

- Preset timer

CALL RESET_TIMER //Se debe limpiar el temporizador antes de su uso
TIMER := <nombre del temporizador (DB)>

- Reset timer

CALL PRESET_TIMER
Time IEC_TIMER
PT := T#<introducir tiempo <m, s, ms,...>>
TIMER := <nombre del temporizador (DB)>

- Parámetros: el R solo en el TONR

IN:(input) BOOL → Evento de disparo

R:(input) BOOL → Reset temporizador

PT:(input) TIME/LTIME → Duración del pulso (**debe ser positivo**)

Q:(output) BOOL → Salida lógica

ET:(output) TIME/LTIME → Tiempo actual

Para limpiar temporizadores y contadores clásicos se usan los bloques FR y R

Plantilla de stop (bloque genérico FC):

Segmento 1: plantilla STOP (desconexión)

```

UN #START           //por nivel

R   #Actuador1       //reset parámetros OUT / IN-OUT

    //...             //reset bits de control...

R   #Temporizador1   //reset de componentes (T, Z,...)
FR #Temporizador1   //habilitación de componentes (T, Z ...)

    //...

BEB                 //fin condicionado del bloque
  
```

Variables bloque:

GENERADOR_CON_ARRANQUE			
	Nombre	Tipo de datos	Valor predet.
1	Input		
2	Output		
3	InOut		
4	Static		
5	ETAPA_SEÑAL_NIVEL_ALTO	Bool	false
6	ETAPA_SEÑAL_NIVEL_BAJO	Bool	false
7	Temp		

Plantilla de conexión START (colocado justo después de la desconexión)

```
U    #START
FP   #marca_de_flanco_START  //condición de arranque/conexión

S    #etapa_reposo_A
S    #etapa_reposo_B

    // ...

R    #bit de control_1
R    #bit de control_2

    // ...
```

ESTABLECER LA SITUACIÓN DE ARRANQUE DEL CONTROL AQUÍ

Hace falta limpiar el flanco

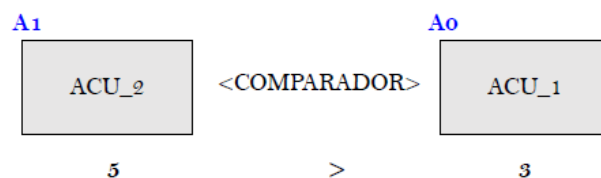
Segmento 1: Desconexión (seg. anterior al de conexión)

```
UN   #START                //desconexión por nivel
R    #marca de flanco START //para un arranque futuro (seg. 2)

....

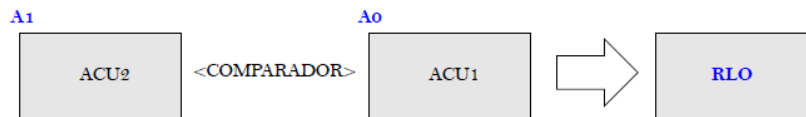
BEB                                     //fin condicionado de bloque
```

Variables analógicas en TIA portal (int o real) -> EW :: El más grande 1 y el más pequeño 0. Si iguales los dos 0



- ☐ A1: biestable que representa el **operando de la izquierda de la comparación.**
- ☐ A0: biestable que representa el **operando de la derecha de la comparación.**
- ☐ Tras una operación de comparación, los bits A0 y A1 se actualizan de acuerdo con el tamaño relativo de los operandos.

R real, D enteros e I enteros



Comparador	Tamaño(bits)	Comentario
(==,<>,>,<,>=,<=) I	16	ACU-1 y ACU-2 no varían
(==,<>,>,<,>=,<=) D	32	
(==,<>,>,<,>=,<=) R	32	

Se lee de arriba hacia abajo → Se puede poner el resultado en una variable

```

L 5
L 8
>=I
  
```

ACU1=8, ACU2=5

A1=0, A0=1, RLO=0

Biestables para lógica más compleja

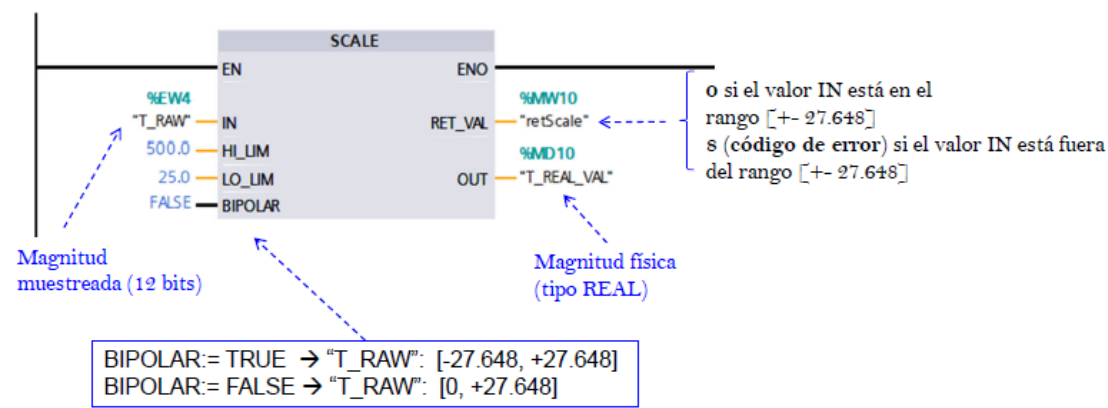
Consulta	Devuelve TRUE si se cumple
==0	A1=0 AND A0=0
<>0	A1 distinto de A0
>0	A1=1 AND A0=0
<0	A1=0 AND A0=1
>=0	==0 OR >0
<=0	==0 OR <0
UO	A1=1 AND A0=1 (Error en tiempo de ejecución: p.ej. división por cero)

//consulta si los bits
A1 (7º) y A0 (6º) valen
0.

U ==0

S M 1.0

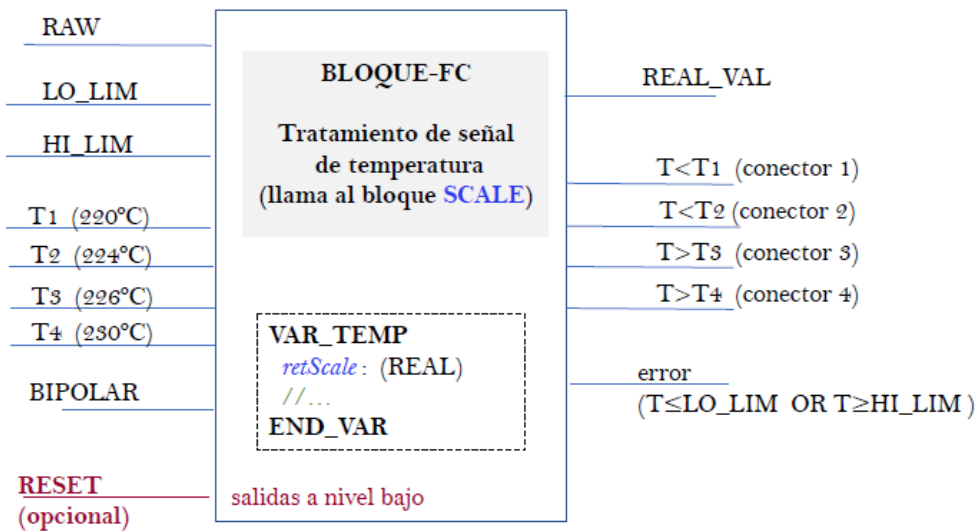
Bloque Scale



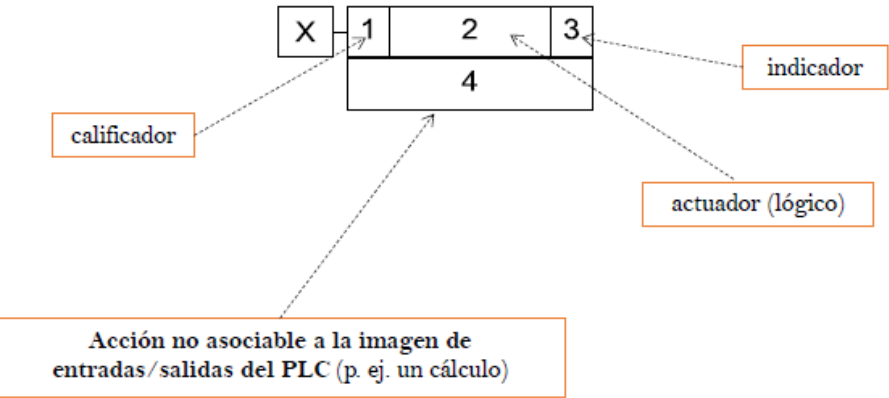
Nombre	Dirección	Formato visualiza...	Valor de observac...
"T_RAW"	%EW4	DEC+/-	29000
"retScale"	%MW10	DEC	8
"T_REAL_VAL"	%MD50	Número en coma...	500.0

Y en unscale se hace igual pero cambiado las variables de IN y de OUT

Conectores: convertir una comparación de un Grafcet en variables booleanas (se implementan en un FC con el bloque scale). Se crea un bloque que acepta los valores a comparar, realiza las comparaciones y devuelve los valores booleanos



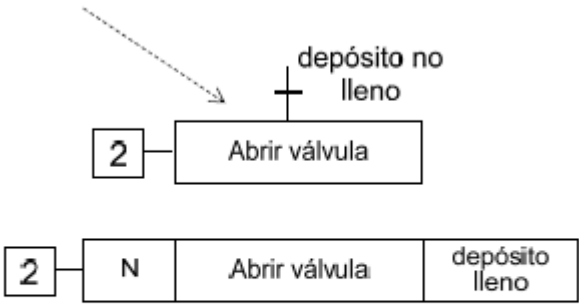
Normas GRAFCET: el indicador es el evento de final de acción (la acción se hace hasta que pase lo de 3) y 4 se pone lo que se quiera (nombre macroetapa)



Calificadores

	Símbolo	Descripción
1	ninguno	acción continua mientras dura la etapa
2	N	acción continua mientras dura la etapa
3	R	desactivación de acción mantenida
4	S	acción mantenida
5	L	acción limitada tras la activación de la etapa
6	D	acción retardada tras la activación de la etapa
7	P	acción impulsional
8	SD	acción mantenida y retardada
9	DS	acción retardada y mantenida
10	SL	acción mantenida y limitada en el tiempo
11	P1	acción de pulso por activación de la etapa
12	P0	acción de pulso por desactivación de la etapa

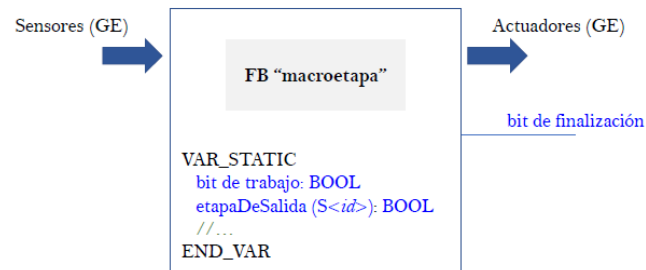
IEC 60848



IEC 61131-3

Plantilla de programación. Usa los siguientes bits de control

- Bit de trabajo: memoria auxiliar para construir la condición de arranque, es true durante toda la ejecución (equivale a la marca de flanco en la plantilla START)
- Bit de finalización : memoria auxiliar que representa la finalización de la macroetapa
- Bit de macroetapa: memoria auxiliar que representa la macroetapa en el GRAFCET principal

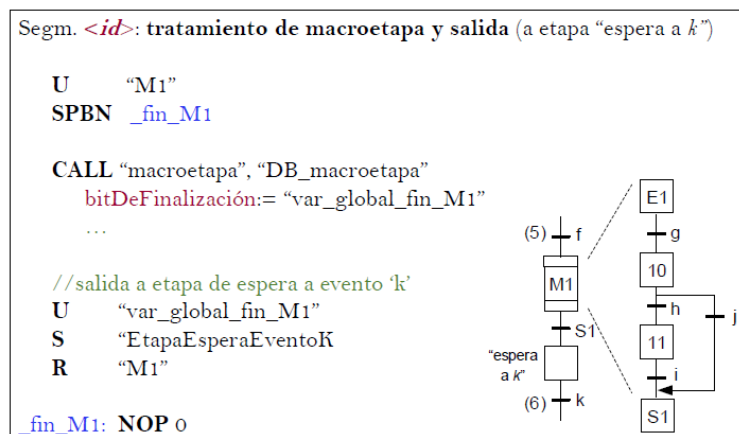


Plantilla START/STOP: (Maniobras)

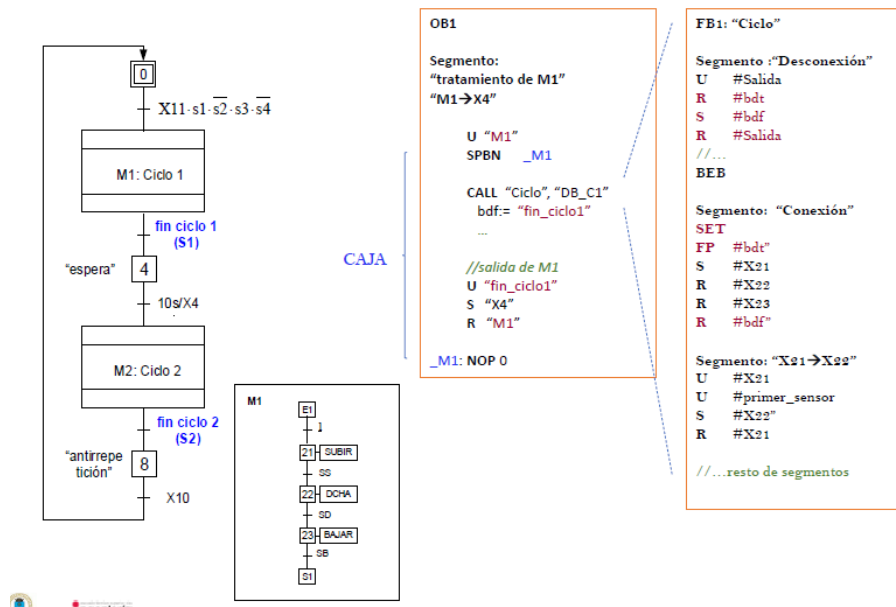
- Evento STOP: activación de la etapa de salida
 - o Desactiva bit de trabajo
 - o Activa bit de finalización
- Evento START: true siempre → por ello el SET y flanco positivo para que solo ocurra 1 vez

<p>Segmento 1: desconexión del grafcet GE encapsulado por <i>M<id></i></p> <pre> U #S<id> //etapa de salida=TRUE R #bitDeTrabajo //para un arranque futuro S #bitDeFinalizacion //informa al cliente R #S<id> //para evitar futuras desconexiones //otras tareas de desconexión //T, Z, salidas, ... BEB </pre>	<p>Segmento 2: conexión / arranque del grafcet GE encapsulado por <i>M<id></i></p> <pre> SET //RLO:=TRUE FP #bitDeTrabajo //evento arranque R #bitDeFinalizacion //act. de estados iniciales del GE S #E<id> //etapa de entrada=TRUE R #resto de estados de GE... </pre>
---	---

Invocación



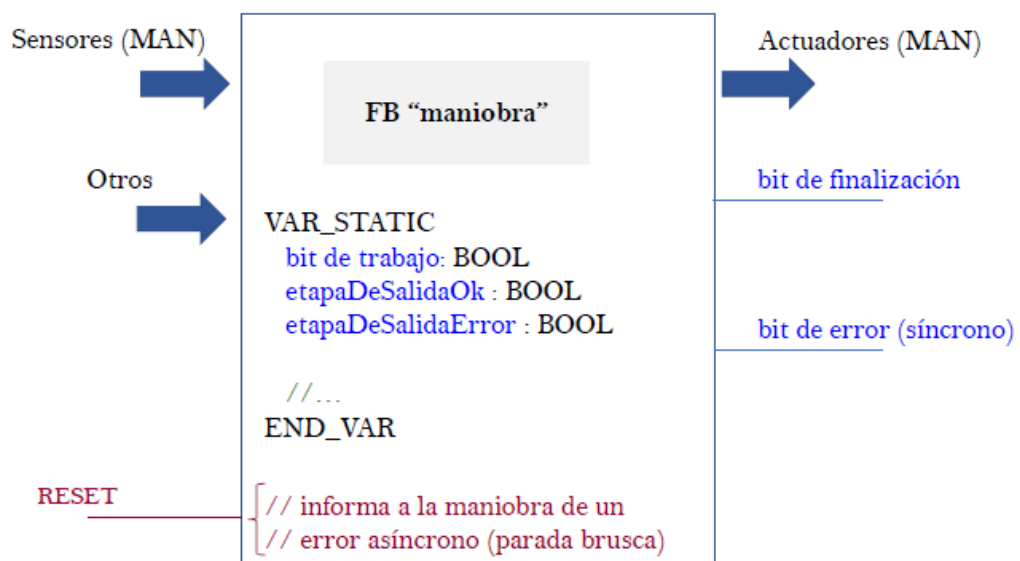
Ej: puente grúa (para llamar a M" simplemente se cambia el bloque de datos)



Maniobra: especificación de control asociada con eventos de inicio y finalización tasados (duración limitada). Plantilla de programación para maniobras (vistos anteriormente salvo bit de error)

- Bit de trabajo
- Bit de finalización
- Bit de error: informa al exterior y termina la maniobra
- Bit de maniobra

Diseño bloques de maniobra (para tratar el evento EMER se hace a través de la entrada RESET)



Plantilla RESET (se deben añadir tantas entradas reset como eventos asíncronos) / Integración del lado del cliente / la start es la vista anteriormente

Segmento 1: plantilla RESET

//parada brusca y reinicio desde la situación inicial

//emplea la plantilla de desconexión ya vista

```
U #RESET //señal de mando – evento asíncrono

R #bitDeTrabajo //actualización de los bits de control
R #bitDeFinalizacion //actualización de los bits de control

R #actuadores... //actualización param. OUT/IN-OUT

R #T, Z... //habilitación de comp. transitorios
FR #T, Z...

//otras tareas de desconexión...

BEB
```

Seg: x1→x2 y tratam. de x1

```
U "X1"
SPBN _001
CALL FB1,DB1
  bdf : "bit_de_fin_EXP"
  bde : "bit_de_error_EXP"
  //... otros parámetros

U "bit_de_error_EXP"
S "X10"
R "X1"

U "bit_de_fin_EXP"
S "X2"
R "X1"

_001: NOP 0
```

Gestión de emergencias

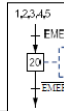
```
U "x_DEFECTO_EMER (X20)"
SPBN _020
CALL "MANIOBRA_PISTON_CON_RESET" , "DATOS_PRIMERA_EXPANSION"
  RESET := TRUE
  actuador_ON := "EXP"
  //... no es necesario rellenar más campos

CALL "MANIOBRA_PISTON_CON_RESET" , "DATOS_PRIMERA_COMPRESION"
  RESET := TRUE
  actuador_ON := "COMP"
  //... no es necesario rellenar más campos

CALL "MANIOBRA_PISTON_CON_RESET" , "DATOS_SEGUNDA_EXPANSION"
  RESET := TRUE
  actuador_ON := "EXP",
  //... no es necesario rellenar más campos

CALL "MANIOBRA_PISTON_CON_RESET" , "DATOS_SEGUNDA_COMPRESION"
  RESET := TRUE
  actuador_ON := "COMP"
  //... no es necesario rellenar más campos

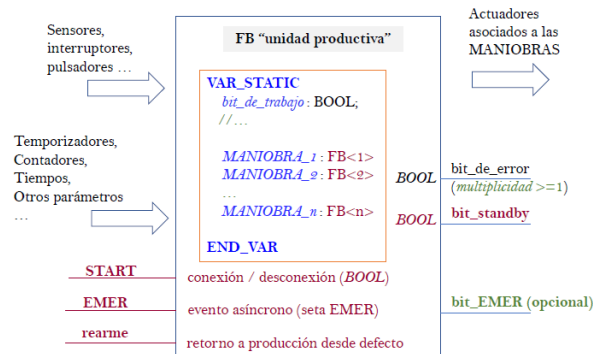
_020: NOP 0
```



Bloques multinstancia o unidades productivas (contienen FB en su interior como multinstancias): esto reduce el numero de bloques de datos y permite que sean portables.

CONTROL_PUENTE_GRUA_DB		
	Nombre	Tipo de datos
1	Input	
2	Output	
3	InOut	
4	Static	
5	man_3A_ida	"man_3A"
6	Input	
7	s_fin1	Bool
8	s_fin2	Bool
9	s_fin3	Bool
10	Output	
11	InOut	
12	act_1	Bool
13	act_2	Bool
14	act_3	Bool
15	aviso_fin_exter...	Bool
16	Static	
17	bit_de_trabajo	Bool
18	x_man1	Bool
19	x_man2	Bool
20	x_man3	Bool
21	aviso_fin_inter...	Bool
22	man_3A_vuelta	"man_3A"

Diseño unidades productivas multi-instancia: se encienden mediante START en lugar del RESET

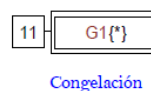
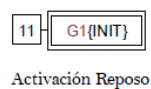
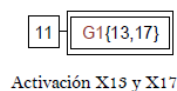


Reglas de forzado

Restricciones al forzado:

- Coordinación **no reflexiva**: un grafcet NO puede forzarse a sí mismo.
- Coordinación **no simétrica**: dos grafcets NO pueden forzarse mutuamente.
- Coordinación con **unicidad**: Un grafcet SÓLO puede tener un ÚNICO grafcet maestro.

Sintaxis de la acción de forzado



Tratamiento congelación: Se añade la congelación como una entrada adicional

```
//FB "G1"

Segmento: CONGELACIÓN
//basado en la plantilla de desconexión
//posterior al segmento de conexión

U #CONGELACION

R #Accion_A
R #Accion_B
R #Accion_C

//informar a maniobras, si hubiere,
//del evento de CONGELACIÓN

BEB
```

**TEMPORIZADORES
MODERNOS STEP 7**

**AUTOMATIZACIÓN
INDUSTRIAL**

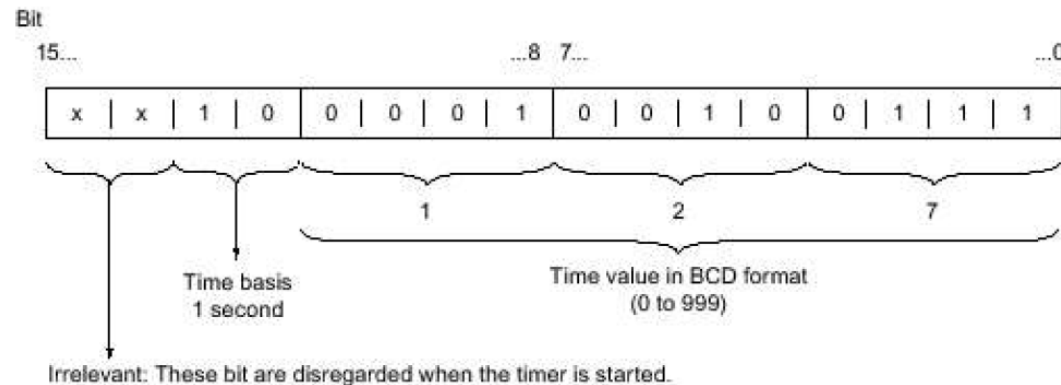


Necesidad de uso nuevos temporizadores (I)

- Los temporizadores clásicos S5Time almacenan el tiempo en 16 bits (Rango limitado de datos) y usan una resolución variable en función del intervalo de uso (una vez fijada una escala se fija la resolución).

Resolution	Range
0.01 s	10 ms to 9 s 990 ms
0.1 s	100 ms to 1 min 39 s 900 ms
1 s	1 s to 16 min 39 s
10 s	10 s to 2 h 46 min 30 s

- Limitación de tiempo**
- Limitación de resolución**



Necesidad de uso nuevos temporizadores (II)

Temporizadores TIME:

Tamaño de 32 bits y resolución constante de 1ms.

Formato T#24d_23h_59min_59s_999ms

Temporizadores LTIME:

Tamaño de 64 bits y resolución constante de 1ns.

Formato T#106751d_23h_59min_59s_999ms_999us_999ns

No es necesario especificar todos los tiempos.

En los formatos se especifican los valores que no se pueden superar.

Value range

T#-24d_20h_31m_23s_64
8ms to
T#+24d_20h_31m_23s_64
7ms

Value range

LT#-106751d_23h_47m_16s_
854ms_775us_808ns to
LT#+106751d_23h_47m_16s_
_854ms_775us_807ns

Timers

Timers	S7-300/400	S7-1200	S7-1500
S5TIME	X	-	X
TIME	X	X	X
LTIME	-	-	X

Temporizadores modernos

- Implementación en AWL
 - RESET_TIMER
 - PRESET_TIMER
 - TON
 - TONR
 - TOF
 - TP

RESET_TIMER

Parámetros:

IEC timer:(**output**) <tipo_temporizador>_TIME/LTIME → estructura de datos donde almacenar tiempo

No cambia el RLO.

RESETEA TOTALMENTE el bloque de datos del temporizador

Ej: Plantilla de desconexión

1	U	"Reset"	%M1.0
2	SPBN	_reset_timer	
3	CALL	RESET_TIMER	
4		IEC_TIMER	
5		TIMER := "IEC_Timer_0_DB"	%DB1
6			
7			
8		_reset_timer : NOP 0	

PRESET_TIMER (I)

Parámetros

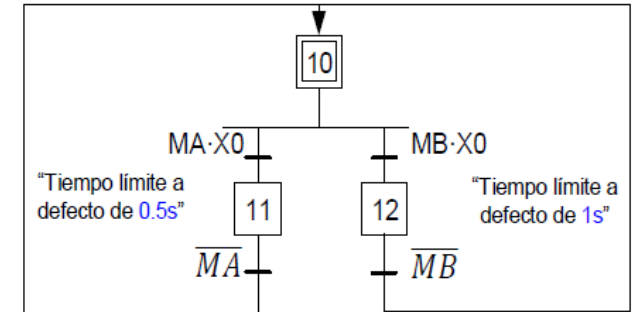
PT:(**input**) TIME/LTIME → duración del temporizador IEC

IEC timer:(**output**) <tipo_temporizador>_TIME/LTIME → estructura de datos donde almacenar tiempo

Función empleada para configurar el tiempo de un temporizador IEC (Bloque de datos).

SE DEBE EVITAR MIENTRAS UN TEMPORIZADOR ESTA EN FUNCIONAMIENTO.

Ej: selección modos de producción:



```
1      U      #Evento_A
2      SPBN   _MA
3      CALL   PRESET_TIMER
4      Time   IEC_TIMER
5      PT     :=#T_A
6      TIMER  := "Tiempo_produccion"
7
8      _MA:   NOP 0
9
10     U      #Evento_B
11     SPBN   _MB
12     CALL   PRESET_TIMER
13     Time   IEC_TIMER
14     PT     :=#T_B
15     TIMER  := "Tiempo_produccion"
16
17     _MB:   NOP 0
18
```


PRESET_TIMER (II): Plantilla Start

U "Start"

FP "M_Start" //Solo se ejecuta una vez

SPBN _fin_start

CALL RESET_TIMER //Se debe limpiar el temporizador antes de su uso

TIMER := <nombre del temporizador (DB)>

CALL PRESET_TIMER

Time IEC_TIMER

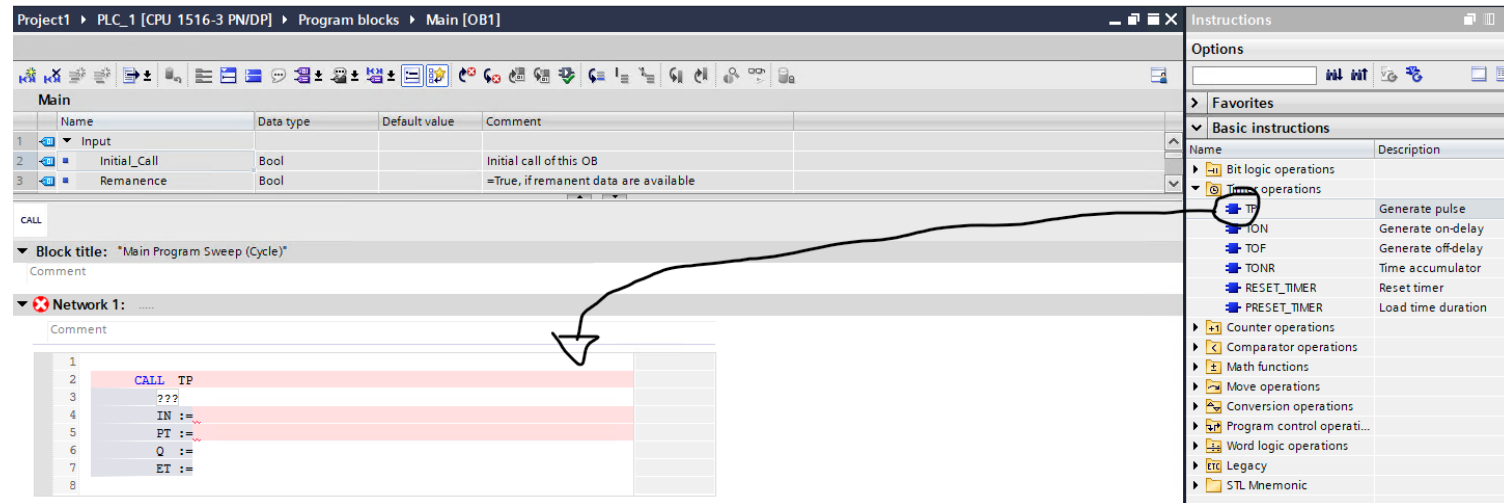
PT := T#<introducir tiempo <m, s, ms,...>>

TIMER := <nombre del temporizador (DB)>

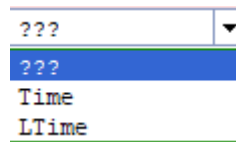
_fin_start : NOP 0

Configuración (I)

Configuración: al arrastrar el bloque sale la siguiente interfaz.



Se pulsa sobre las interrogaciones y se abre una ventana de dialogo donde se selecciona el tipo de tiempo a emplear y se crea un bloque de datos donde se almacena este tiempo.



Configuración (II)

Una vez introducido, el bloque con el temporizador actualiza su nombre donde aparece el nombre de la estructura de datos y se crea un bloque con las variables.

1		
2	CALL TP , "Tiempo_produccion"	%DB1
3	Time	
4	IN :=	
5	PT :=	
6	Q :=	
7	ET :=	
8		

Tiempo_produccion							
Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PT	Time	T#0ms	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ET	Time	T#0ms	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
IN	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Q	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Ejemplo usando el bloque de datos:

1	
2	CALL TP , "Tiempo_produccion"
3	Time
4	IN := "IN"
5	PT := "Tiempo_produccion".PT
6	Q :=
7	ET :=

%DB1
%M1.0

TON (I)

Parámetros

IN:(input) BOOL → Evento de disparo

PT:(input) TIME/LTIME → Duración del pulso (**debe ser positivo**)

Q:(output) BOOL → Salida lógica

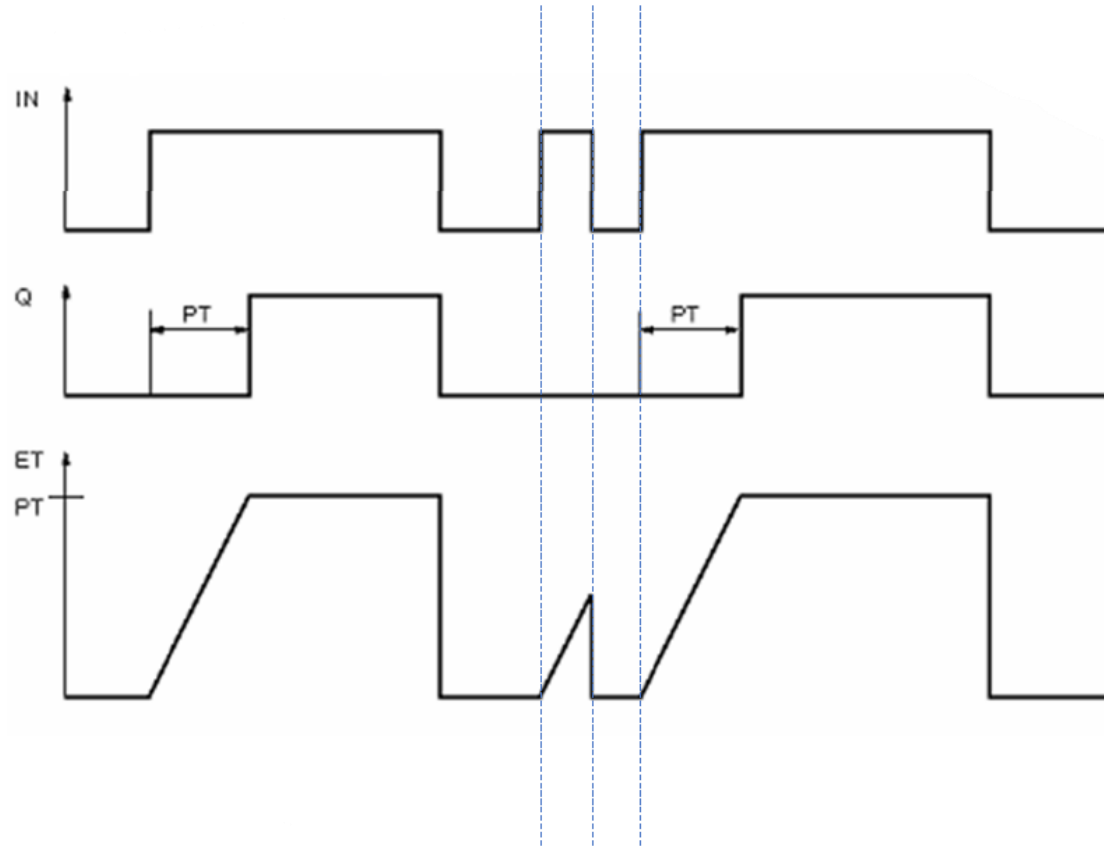
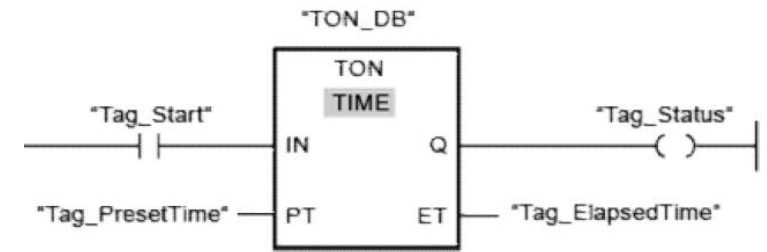
ET:(output) TIME/LTIME → Tiempo actual

En **ET** se obtiene el tiempo transcurrido una vez conectado y si **llega a PT toma un valor constante**. Se resetea si el temporizador se desconecta.

Si **IN** cambia a **0** → **Q** se resetea = **0**.

NO TIENE MEMORIA

TON (II)



TON (III) / Aplicaciones

Retardos o tiempos de espera.

//Configuración temporizador

CALL TON, "Tiempo_producción"

Time

IN := "X2"

PT := "Tiempo_producción".PT

Q := "T1"

ET :=

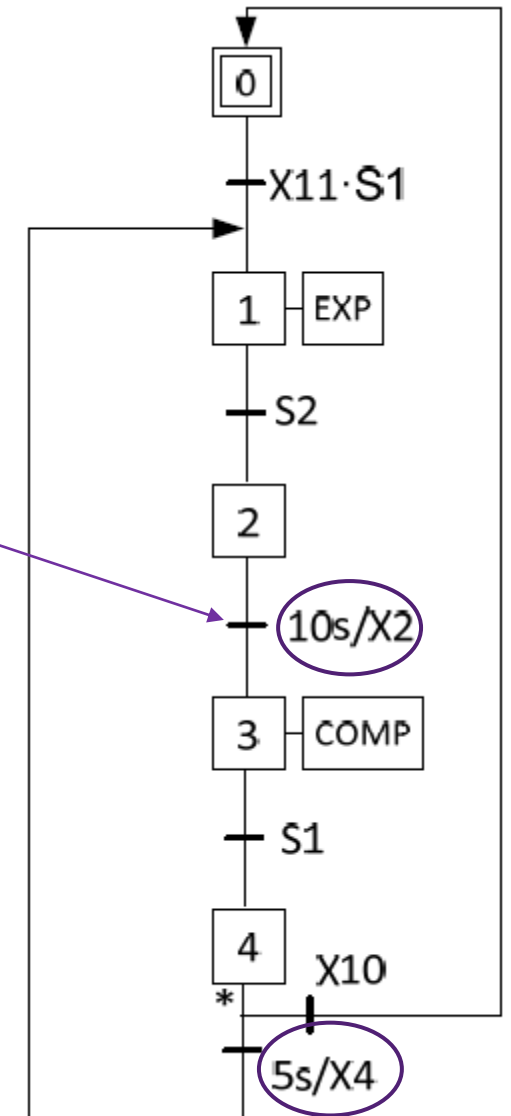
//Segmento X2 → X3

U "X2"

U "T1"

S "X3"

R "X2"



TONR (I)

Parámetros

IN:(input) BOOL → Evento de disparo

R :(input) BOOL → Reset temporizador

PT:(input) TIME/LTIME → Duración del pulso (**debe ser positivo**)

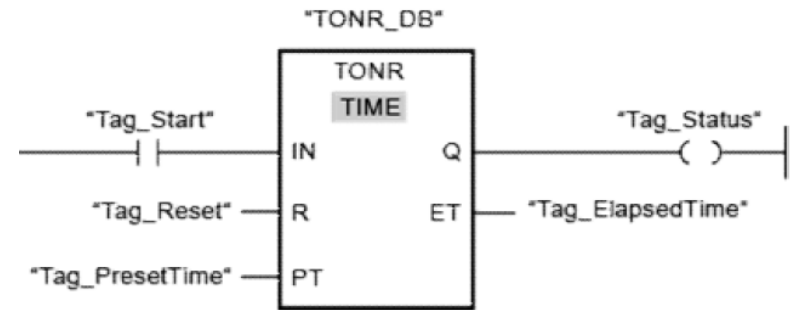
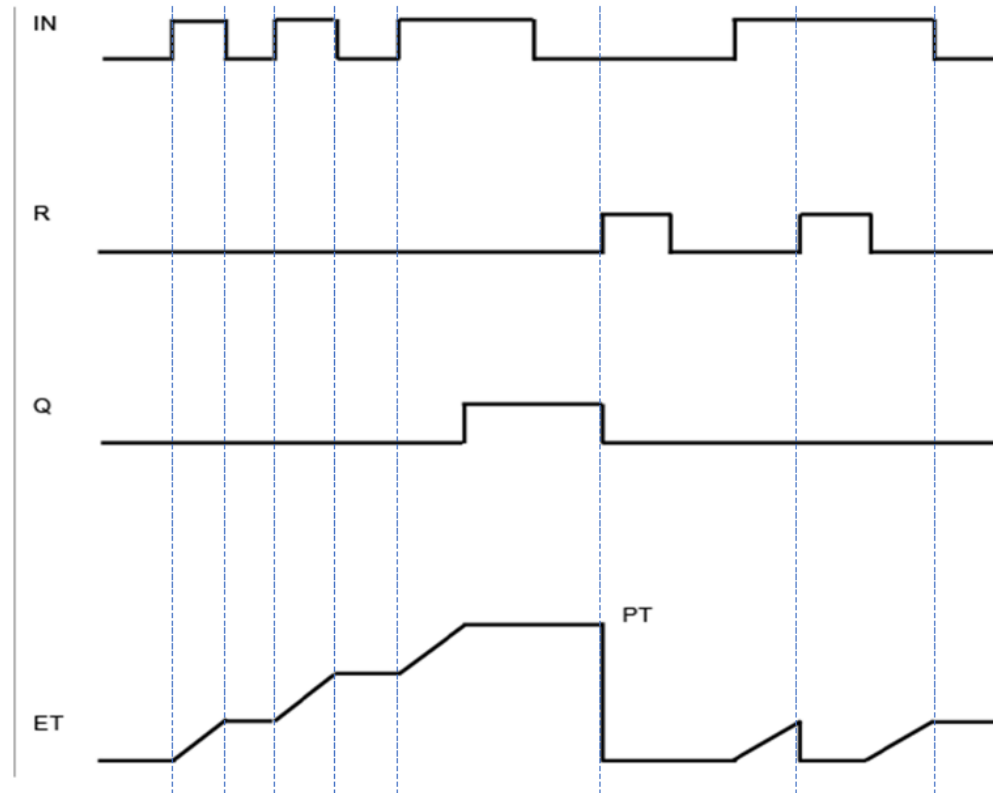
Q:(output) BOOL → Salida lógica

ET:(output) TIME/LTIME → Tiempo actual

En ET se almacena el valor de cuenta que permanece constante una vez se termina de contar.

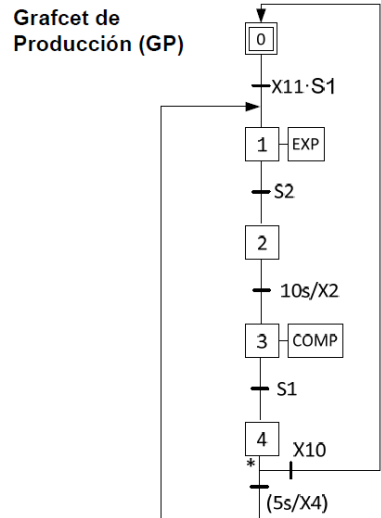
IMPORTANTE: No depende de IN. **RESET SI R = 1 → Con memoria ≠ Redisparable**

TONR (II)

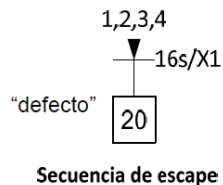
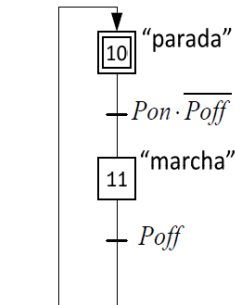


TONR (III) / Aplicaciones

La aplicación principal es la de perro guardián. **Programación segmento escape.**



Grafcet de Marcha-Paro (GMP)



//Ley de control

CALL TONR, "Tiempo_escape"

Time

IN := "Conector"

R := **"Reset_escape"**

PT := "Tiempo_escape".PT

Q := "T_Defecto"

ET :=

//Segmento redisparo

//Es necesario que ocurra

//antes de la ley de control

//para que no acabe

//cayendo en el escape

//siempre

U "X1"

FP "Redisparo"

S **"Reset_escape"**

//Segmento Escape

U "T_Defecto"

S "X20"

R "Resto memorias"

S **"Reset_escape"**

//Segmento conector

O "X1"

O "X2"

O "X3"

O "X4"

= "Conector"

//Activación temporizador

//si se activa X1 se

//desactiva reset

UN "Conector"

= "Reset_escape"

TP(I)

Parámetros

IN:(input) BOOL → Evento de disparo

PT:(input) TIME/LTIME → Duración del pulso (**debe ser positivo**)

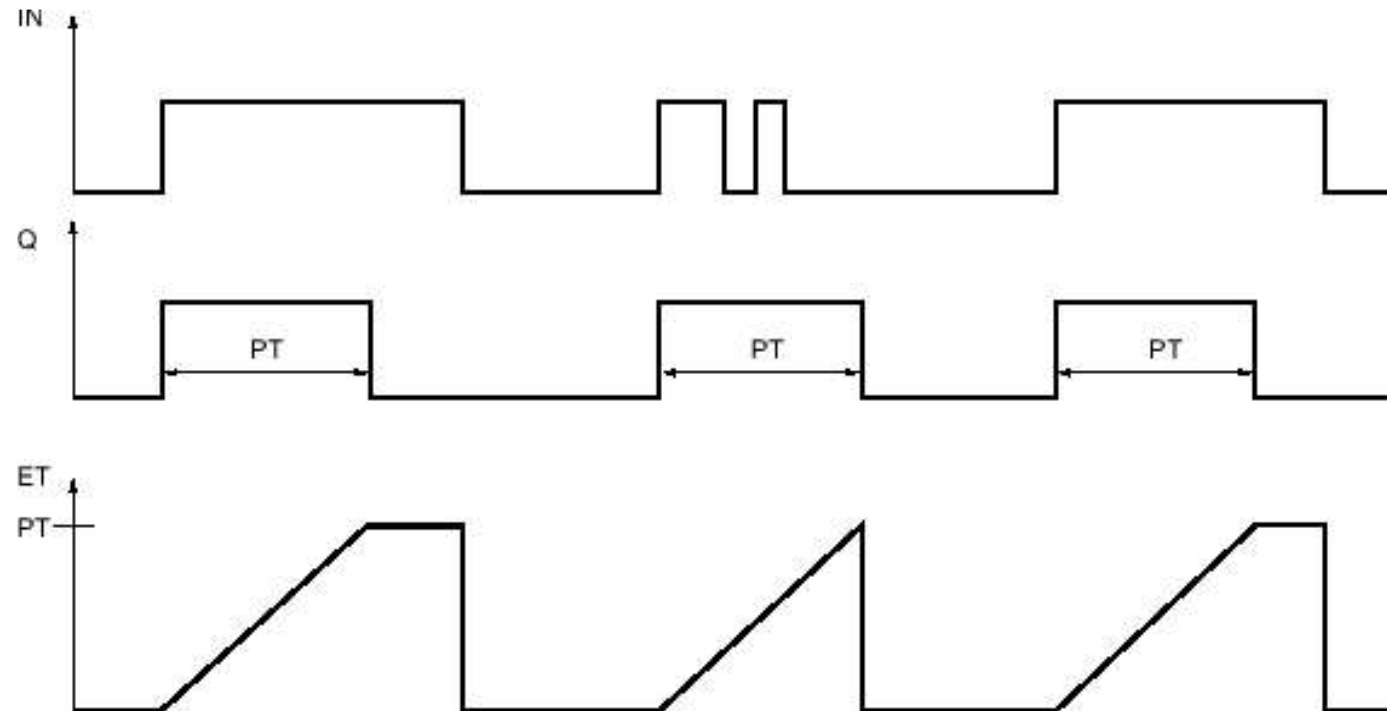
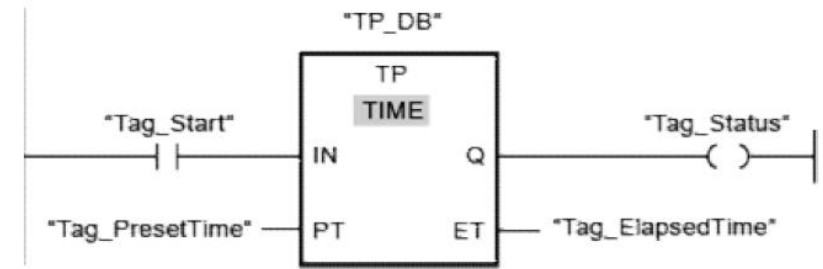
Q:(output) BOOL → Salida lógica

ET:(output) TIME/LTIME → Tiempo actual

En **ET** se obtiene el valor del tiempo transcurrido que se resetea cuando pasa el tiempo e **IN** = 0.

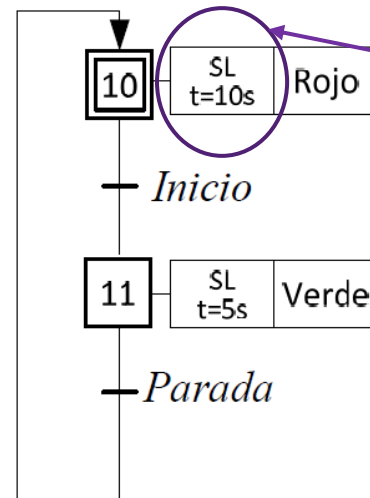
Tiene memoria -- Pero no es redisparable

TP(II)



TP(III) / Aplicaciones

Implemente el detector de marcha y parada del ejercicio anterior con la diferencia de que los pilotos **verde** y **rojo** se **mantienen encendidos** durante **5s** y **10s** respectivamente, aunque cambie el estado (ambas acciones **mantenidas y limitadas**).



Acciones limitadas con enclavamiento

EJERCICIO: Crear TP sin memoria para las acciones limitadas

TOF (I)

Parámetros

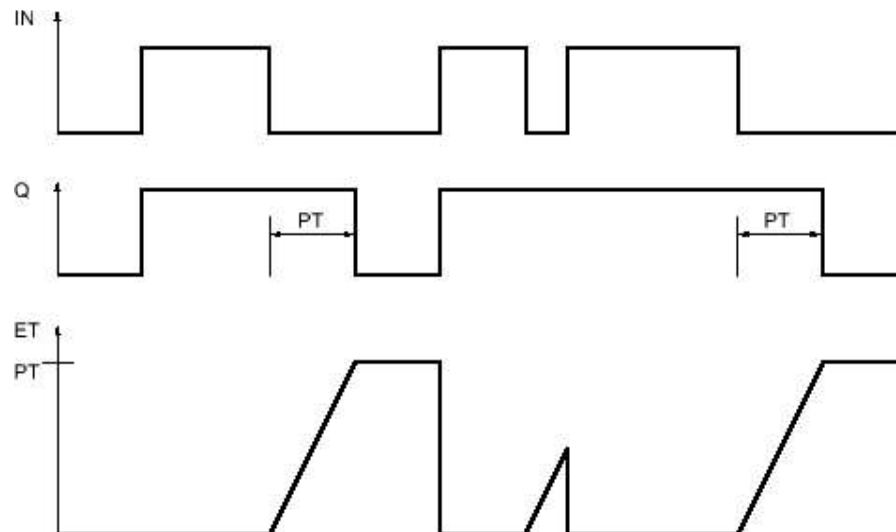
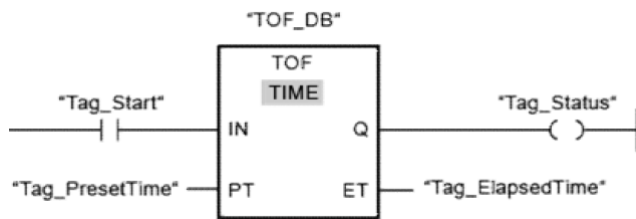
IN:(input) BOOL → Evento de disparo

PT:(input) TIME/LTIME → Duración del pulso (**debe ser positivo**)

Q:(output) BOOL → Salida lógica

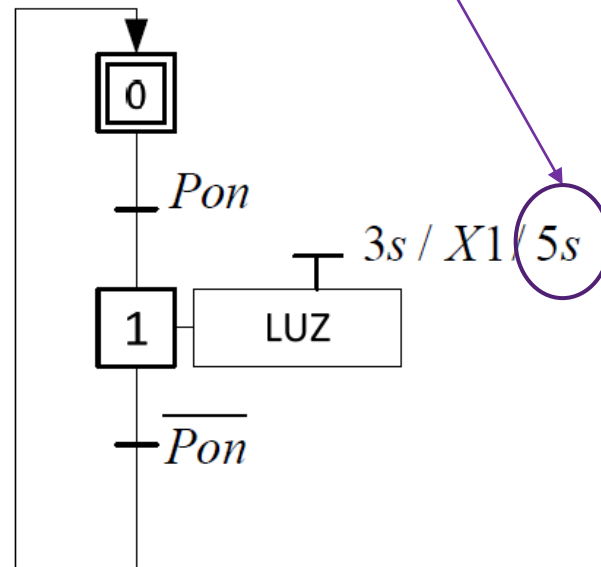
ET:(output) TIME/LTIME → Tiempo actual

En **ET** se almacena el valor una vez se desconecta hasta alcanzar **PT**.



TOF (II) / Aplicaciones

Desactivación retardada de acciones.



The background of the image is a dark gray field filled with a repeating pattern of light gray speech bubbles. Each bubble contains a black question mark. The bubbles are scattered across the entire frame, creating a textured, thematic background for the central text.

¿PREGUNTAS?
