

AW-SYS

software de simulación de PLCs y de instalaciones

Manual del usuario

Revisión: 2.02.06

Autoware srl

Via Pio XII, 15 – 74100 Taranto

Tel: 099.7303413

URL: <http://www.autoware.com>

e-mail: support@autoware.com

Via Roma, 6 –70010 Casamassima

Tel: 099.675426

SUMARIO

CAPÍTULO 1 INTRODUCCIÓN A AW-SYS

Generalidades.....	1-1
Composición	1-2
El módulo PC-Sim	1-3
El módulo PL-Sim	1-4
El módulo PC-RIS.....	1-5
El módulo PL-RIS.....	1-5
El módulo PL-Lab.....	1-6
El módulo PL-Net	1-6
Guía on line.....	1-6
Servicio de Soporte Técnico para Usuarios de AW-SYS.....	1-7
Autoware en Internet.....	1-8
Estructura del manual.....	1-8

CAPÍTULO 2 INSTALACION DE AW-SYS

Contenido del producto.....	2-1
Control de los requisitos del sistema.....	2-2
Instalación de AW-SYS	2-3
Desinstalación del programa	2-4

CAPÍTULO 3 PC-SIM: EL SIMULADOR DE PLC

Introducción	3-1
Composición de un proyecto.....	3-3
Ventana Lista bloques (S5 100U)	3-4
El editor Ladder	3-10
El editor AWL para el PLC PS3	3-24
El editor AWL para el PLC S5 100U.....	3-29
El editor de datos (S5 100U).....	3-37
Ventana Tabla de símbolos	3-42
Ventana Estado variables	3-47
Ventana Asignación de preselectores (FPC 404)	3-54
Ventana PLC	3-56
Ventana Informaciones sobre el proyecto	3-60
Ventana Informaciones sobre PC-Sim	3-62
Comandos de los menús	3-63

Barra del título.....	3-83
Barra instrumentos.....	3-84
Accesos rápidos con el teclado.....	3-85
Impresión de la documentación.....	3-86
Mensajes de aviso.....	3-90
Errores de compilación.....	3-99
Errores de ejecución.....	3-102
CAPÍTULO 4 PL-SIM: EL SIMULADOR DE INSTALACIONES	
Introducción	4-1
Puesta en marcha de las instalaciones.....	4-1
Instalación n.1: Electroneumática 1.....	4-2
Instalación n.2: Electroneumática 2.....	4-5
Instalación n.3: Electroneumática 3.....	4-8
Instalación n.4: Automatización puerta	4-11
Instalación n.5: Control del nivel de un depósito	4-15
Conectar las instalaciones al PLC.....	4-20
Comandos de los menú.....	4-22
Instalación n.6: Cintas transportadoras.....	4-25
Instalación n.7: Ascensor.....	4-32
Instalación n.8: Mezclado de líquidos	4-38
Instalación n.9: Cruce semaforizado.....	4-46
CAPÍTULO 5 PC-RIS: INTERFACE HACIA SISTEMAS DE CONTROL REALES	
Introducción	5-1
Por qué usar PC-RIS.....	5-2
Puesta en marcha del sistema	5-2
Continuación del ejercicio.....	5-6
CAPÍTULO 6 PL-RIS: INTERFACE HACIA INSTALACIONES REALES	
Introducción	6-1
Puesta en funcionamiento del sistema	6-2
CAPÍTULO 7 PL-LAB: INTERFACE HACIA INSTALACIONES DE FABRICACIÓN PROPIA	
Introducción	7-1
Puesta en marcha del sistema	7-2
Descripción de PL-Lab.....	7-2
Intercambio de datos con InTouch	7-4
Construcción de la primera instalación.....	7-4

Construcción de una instalación más compleja	7-12
La instalación prototipo.....	7-19
CAPÍTULO 8 PL-NET: INTERFACE HACIA SISTEMAS MPS	
Generalidades.....	8-1
Puesta en funcionamiento del sistema	8-1
Direccionamiento de las entradas/salidas	8-5
Direccionamiento de las word globales.....	8-6
Consideraciones para el uso de las word globales.....	8-7
CAPÍTULO 9 REALICEMOS JUNTOS LA PRIMERA APLICACIÓN	
Operaciones preliminares.....	9-1
PLC FPC 404	9-2
PLC PS3.....	9-7
PLC S5 100U	9-12
Test del programa.....	9-18
CAPÍTULO 10 NOTAS SOBRE LA PROGRAMACIÓN DEL PLC FPC 404	
Generalidades sobre el PLC FPC 404.....	10-1
Direccionamiento	10-1
Elaboración del programa en uso	10-3
Programación en Ladder	10-4
Operaciones de temporización	10-6
Operaciones de conteo	10-8
CAPÍTULO 11 EJEMPLOS DE PROGRAMACION PARA PLC FPC 404	
Ejemplo 1: combinación AND	11-2
Ejemplo 2: combinación OR.....	11-2
Ejemplo 3: combinación AND de OR.....	11-3
Ejemplo 4: combinación OR de AND.....	11-3
Ejemplo 5: combinación XOR	11-3
Ejemplo 6: autorretención	11-4
Ejemplo 7: set y reset	11-4
Ejemplo 8: activación por flancos.....	11-5
Ejemplo 9: temporizador a impulso	11-6
Ejemplo 10: temporizador a impulso prolongado	11-7
Ejemplo 11: temporizador con retardo a la activación.....	11-8
Ejemplo 12: temporizador con retardo a la desactivación.....	11-9
Ejemplo 13: temporizador con retardo a la activación con memoria	11-10

Ejemplo 14: temporizador con retardo a la activación y a la desactivación	11-11
Ejemplo 15: impulso retardado	11-12
Ejemplo 16: tren de impulsos	11-13
Ejemplo 17: conteo hacia atrás.....	11-14
Ejemplo 18: conteo hacia adelante	11-15
Ejemplo 19: conteo del tiempo de cierre de una entrada (en segundos).....	11-16
Ejemplo 20: conteo del tiempo de cierre de una entrada (en horas, minutos y segundos).....	11-17
Ejemplo 21: generador de onda cuadrada	11-18
Ejemplo 22: control temporizado de luces	11-19
Ejemplo 23: divisor de frecuencia (x4)	11-21
Ejemplo 24: semáforo para Fórmula 1	11-22

CAPÍTULO 12 NOTAS SOBRE LA PROGRAMACIÓN DEL PLC PS3

Generalidades sobre el PLC PS3	12-1
Direccionamiento	12-2
Elaboración del programa en uso	12-6
Registros.....	12-9
Programación en AWL.....	12-10
Programación en Ladder.....	12-22
Módulos de sistema	12-24

CAPÍTULO 13 EJEMPLOS DE PROGRAMACION PARA PLC PS3

Ejemplo 1: combinación AND	13-2
Ejemplo 2: combinación OR	13-2
Ejemplo 3: combinación AND de OR.....	13-3
Ejemplo 4: combinación OR de AND	13-3
Ejemplo 5: combinación XOR	13-4
Ejemplo 6: autorretención	13-5
Ejemplo 7: set y reset	13-5
Ejemplo 8: activación por flancos	13-6
Ejemplo 9: temporizador con retardo en la activación	13-7
Ejemplo 10: temporizador con retardo en la desactivación.....	13-8
Ejemplo 11: temporizador a impulso.....	13-9
Ejemplo 12: temporizador a impulso prolongado	13-10
Ejemplo 13: temporizador con retardo a la activación con memoria	13-12
Ejemplo 14: temporizador con retardo a la activación y a la desactivación	13-13
Ejemplo 15: impulso retardado	13-14
Ejemplo 16: tren de impulsos	13-15
Ejemplo 17: conteo hacia atrás.....	13-16

Ejemplo 18: conteo hacia adelante.....	13-16
Ejemplo 19: conteo del tiempo de cierre de una entrada (en segundos).....	13-17
Ejemplo 20: conteo tiempo de cierre de una entrada (en horas, minutos y segundos)	13-19
Ejemplo 21: generador de onda cuadrada	13-21
Ejemplo 22: otro generador de onda cuadrada.....	13-22
Ejemplo 23: control temporizado de luces	13-23
Ejemplo 24: divisor de frecuencia (x4)	13-26
Ejemplo 25: semáforo para Fórmula 1	13-27
Ejemplo 26: luces secuenciales con 4 canales.....	13-29
Ejemplo 27: conteo de entradas cerradas	13-32

CAPÍTULO 14 NOTAS SOBRE LA PROGRAMACION DEL PLC S5 100U

Generalidades sobre el PLC S5 100U	14-1
Direccionamiento	14-2
Elaboración del programa en uso	14-6
Programación estructurada.....	14-8
Programación en AWL.....	14-12
Programación en Ladder	14-37

CAPÍTULO 15 EJEMPLOS DE PROGRAMACION PARA PLC S5 100U

Ejemplo 1: combinación AND	15-2
Ejemplo 2: combinación OR.....	15-2
Ejemplo 3: combinación AND de OR.....	15-3
Ejemplo 4: combinación OR de AND.....	15-4
Ejemplo 5: combinación XOR	15-5
Ejemplo 6: autorretención	15-5
Ejemplo 7: set y reset	15-6
Ejemplo 8: elevación de flancos.....	15-6
Ejemplo 9: temporizador a impulso	15-8
Ejemplo 10: temporizador a impulso prolongado	15-9
Ejemplo 11: temporizador con retardo a la activación.....	15-10
Ejemplo 12: temporizador con retardo a la activación, con memoria y reset.....	15-11
Ejemplo 13: temporizador con retardo a la desactivación.....	15-12
Ejemplo 14: temporizador con retardo a la activación y a la desactivación.....	15-13
Ejemplo 15: impulso retardado	15-14
Ejemplo 16: tren de impulsos.....	15-15
Ejemplo 17: conteo hacia atrás.....	15-16
Ejemplo 18: conteo hacia delante.....	15-17
Ejemplo 19: conteo del tiempo de cierre de una entrada (en segundos).....	15-17

Ejemplo 20: conteo del tiempo de cierre de una entrada (en horas, minutos y segundos).....15-19

Ejemplo 21: generador de onda cuadrada15-20

Ejemplo 22: otro generador de onda cuadrada15-22

Ejemplo 23: control temporizado de luces15-22

Ejemplo 24: divisor de frecuencia (x4)15-25

Ejemplo 25: conteo de entradas cerradas (I solución)15-27

Ejemplo 26: conteo de entradas cerradas (II solución).....15-28

Ejemplo 27: semáforo para Fórmula 115-30

Ejemplo 28: luces secuenciales con 4 canales15-32

Ejemplo 29: luces secuenciales en barra15-35

APÉNDICE A LA INTERFACE EASYPORT D16

GeneralidadesA-1

Anillo de fibra óptica.....A-1

Función de los pilotos luminososA-2

RecomendacionesA-3

Configuración.....A-3

Identificación de los conductoresA-4

Datos técnicosA-5

Criterios utilizados en este documento

Criterios tipográficos

Un texto en *cursiva* indica nombres de componentes del programa en la terminología propia de AW-SYS. Los componentes del programa son, por ejemplo, las ventanas, los comandos de los menús, los pulsadores, etc. Además, con ese mismo tipo de letra se identifican las referencias a títulos de capítulos y secciones de este manual. En la versión electrónica, muy a menudo estos componentes llevan asociados un enlace con el tema de referencia. Si el puntero del mouse situado sobre ellos se transforma en una mano, haga clic y la visualización del manual se desplazará automáticamente al tema especificado.

Un texto en caracteres Tahoma se utiliza para indicar nombres de archivos y de directorios.

Un texto en caracteres **COURIER NEW NEGRITA** mayúscula se utiliza para indicar siglas de operandos en los programas para PLC o bien partes de código de los mismos.

Combinaciones de teclas

Cuando se hace referencia a una combinación de teclas (por ej.. Ctrl+N) se entiende que, para activar el comando, hay que mantener pulsada la primera tecla (en el ejemplo, la tecla Control) y a continuación pulsar la segunda (N en el ejemplo) para luego soltar las dos.

Operaciones con el mouse

Hacer clic sobre un objeto significa situar el puntero del mouse sobre el objeto visualizado en la pantalla, pulsar y luego soltar el botón principal del mouse.

Hacer doble clic sobre un objeto significa situar el puntero del mouse sobre el objeto visualizado en la pantalla y pulsar y soltar dos veces de forma rápida el botón principal del mouse.

Hacer clic con el botón derecho sobre un objeto significa situar el puntero del mouse sobre el objeto visualizado en la pantalla, pulsar y luego soltar el botón secundario del mouse.

CAPÍTULO 1

INTRODUCCIÓN A AW-SYS

Generalidades

La enseñanza de la automatización de base y de la programación de los PLC ha entrado de una manera preponderante en casi todas las disciplinas técnicas de los Institutos técnicos y profesionales. En ese mismo sentido trabajan también los Institutos y los Centros de Formación Profesional.

Organizar un taller eficiente, funcional y flexible, intentado limitar al máximo las inversiones en términos económicos y de instalación logística, representa para cualquiera un auténtico problema. De hecho, la realización de un ejercicio de automatización comporta la adecuación de puntos de trabajo que incluyan un PLC completo de sistema de desarrollo (que la mayoría de las veces funciona basándose en un ordenador personal), y uno o más paneles para la simulación de la instalación que es objeto del ejercicio en cuestión. Pronto nos daremos cuenta de la carga, tanto económica como logística, a la que nos enfrentamos en el caso de que queramos realizar más de un ejercicio. Además, muy a menudo, reproducir determinadas instalaciones reales con modelos físicos se convierte en algo prácticamente imposible

Por estos motivos, Autoware presenta AW-SYS, un sistema que le permite transformar su ordenador personal en un taller virtual de automatización. AW-SYS es un sistema de software y hardware que simula, en un ordenador personal trabajando en un entorno Windows, los PLCs más difundidos en el campo educativo así como otras instalaciones de gran validez didáctica.

AW-SYS está dirigido a todos aquellos que desarrollan actividades didácticas en el terreno de la automatización industrial:

- Institutos Técnicos y Profesionales con especialidades en Electrónica, Electrotécnica, Mecánica e Informática.
- Universidades.
- Centros de Formación Profesional.
- Empresas.

La utilización del software de simulación AW-SYS como laboratorio virtual de automatización, ofrece toda una serie de ventajas de índole práctico:

- simplicidad en la organización del taller, que puede ser un aula de informática normal y corriente.
- flexibilidad del taller, ya que los ordenadores personales también pueden utilizarse para otras actividades.
- economía respecto a las soluciones tradicionales.
- ninguna necesidad de mantenimiento.

y a nivel didáctico, permite:

- mayor activación de los mecanismos de percepción y de atención
- rapidez de interacción entre el alumno y el ejercicio a realizar
- activación de una fuerte carga motivadora
- activación de un proceso de enseñanza/aprendizaje individualizado y verificable por etapas
- visualización y verificación inmediata del trabajo realizado.

Con AW-SYS pueden efectuarse todas aquellas operaciones que conducen a la automatización de una instalación igual que en un caso real, es decir:

- escritura del programa del usuario.
- conexión de los puntos de trabajo con el PLC;
- verificación del programa;
- impresión de la documentación.

Todo ello en un ambiente de comprensión inmediata que rápidamente se le hará familiar, gracias también a la utilización de guías on line adecuadas a cada contexto, a la presencia de notas sobre la programación de los PLCs, a los numerosos ejemplos de programación y a las capacidades propias de las aplicaciones de Windows.

Además, los usuarios de AW-SYS podrán realizar sus propias instalaciones simuladas con el paquete de software InTouch de WonderWare y conectarlas al PLC simulado o al real, gracias al módulo **PL-Lab**.

Composición

AW-SYS está concebido de manera modular para ser utilizado de la forma más adecuada en relación con las diversas exigencias didácticas de los docentes.

AW-SYS está compuesto por los siguientes módulos integrados de software y hardware:

- **PC-Sim** Simulador de Controlador Lógico Programable (PLC)
- **PL-Sim** Simulador de instalaciones
- **PC-RIS** Interface entre sistemas de control real e instalaciones simuladas
- **PL-RIS** Interface entre PLC simulado e instalaciones reales
- **PL-Lab** Interface hacia instalaciones de fabricación propia
- **PL-Net** Interface hacia sistemas MPS.

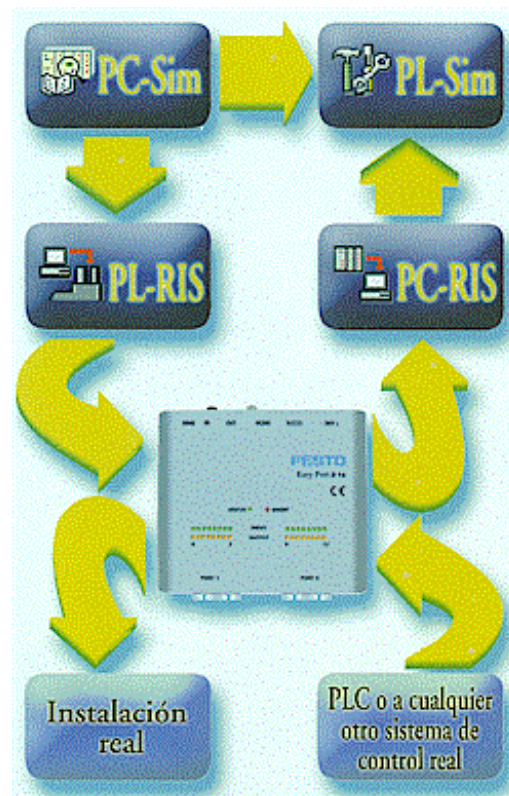


Figura 1: composición de AW-SYS

El módulo PC-Sim

PC-Sim es el módulo de AW-SYS dedicado a la simulación de controladores lógicos programables (PLC). Constituye un instrumento óptimo para la enseñanza de la programación de estos aparatos ya que, mediante la utilización de tan sólo un ordenador personal, permite:

- la visualización del PLC, que se representa gráficamente en una ventana específica en la pantalla.
- la escritura del programa del usuario por medio de texto o de gráficos, utilizando los lenguajes específicos más difundidos, Ladder (Diagrama de Contactos) y AWL (Lista de Instrucciones).
- la definición de símbolos, que podrán utilizarse en el programa para simplificar el trabajo, en lugar de los operandos absolutos.
- la comprobación del programa realizado, por medio de interruptores, pulsadores y pilotos luminosos indicadores del estado, todos ellos simulados y presentes en el interior del software.
- la corrección del propio programa, con la ayuda de potentes instrumentos de depuración (debug) en línea, que los mismos editores de programas ponen a disposición del usuario.
- la visualización del estado y/o del valor de todas las variables utilizadas por el programa, con su continua actualización.

Además, y gracias a la integración existente entre los distintos módulos, usted podrá utilizar PC-Sim y dirigir las instalaciones simuladas de PL-Sim o las instalaciones reales presentes en su taller, por medio del módulo PL-RIS.

En el momento de la impresión de este manual, PC-SIM simula los siguientes PLCs:

- Klöckner Moeller mod. PS3
- Festo mod. FPC 404
- Siemens mod. S5 100U

El módulo PL-Sim

PL-Sim es el módulo de AW-SYS dedicado a la simulación de instalaciones y permite realizar múltiples y variados ejercicios prácticos.

Las instalaciones de PL-Sim, todas de una gran validez didáctica, constituyen una alternativa o bien una incorporación a los bancos de simulación y ponen al usuario frente a diversas situaciones en el campo de la automatización, a menudo imposibles de reconstruir con otros instrumentos en un taller.

Las instalaciones de PL-Sim pueden controlarse directamente por medio de los PLC simulados de PC-Sim o, mediante la utilización del módulo PL-RIS, hacerlo a través de los PLC reales de cualquier marca presentes en su taller. Gracias a PL-RIS, las instalaciones también podrán controlarse por medio de sistemas de microprocesadores, microcontroladores, tarjetas electrónicas o cuadros eléctricos de lógica cableada.

En el momento de la impresión de este manual, se están distribuyendo las siguientes instalaciones:

- **Electroneumática 1:** banco de electroneumática con 3 cilindros de doble efecto y distribuidores 5/2 biestables.
- **Electroneumática 2:** banco de electroneumática con 3 cilindros de doble efecto y distribuidores 5/2 monoestables.
- **Electroneumática 3:** banco de electroneumática con 3 cilindros de simple efecto y distribuidores 3/2 monoestables.
- **Automatización puerta:** puerta automática con mando del motor, luz intermitente, final de carrera, célula fotoeléctrica y mando a distancia.
- **Control del nivel de un depósito:** depósito con indicadores de nivel de funcionamiento y de alarma, válvulas de entrada y salida, pilotos y timbre de alarma, registrador de nivel.
- **Cintas transportadoras:** instalación industrial con dos cintas transportadoras, sensores de proximidad y motores trifásicos con inversión de la marcha.
- **Ascensor:** instalación civil de un ascensor para cuatro pisos, con células fotoeléctricas, sensor de peso, botones de llamada y retorno e indicadores luminosos.

- **Mezclado de líquidos:** instalación industrial para la producción de un compuesto obtenido por medio de la mezcla y el calentamiento de diversos líquidos.
- **Cruce semaforizado:** instalación de un semáforo en el cruce de dos calles con posibilidad de funcionamiento automático, manual y nocturno.

El módulo PC-RIS

PC-RIS es el módulo de AW-SYS que permite controlar las instalaciones simuladas de PL-Sim con múltiples y variados sistemas de hardware que, si bien pueden estar ya presentes en su taller, no están incluidos en el sistema AW-SYS, tales como:

- PLC reales (de cualquier marca)
- sistemas con microprocesador
- microcontroladores
- tarjetas electrónicas de lógica cableada
- cuadros eléctricos de lógica cableada
- ordenadores personales con tarjetas de I/O

Para su funcionamiento es necesario disponer de los siguientes accesorios de hardware:

- interface de hardware para la conexión de 16 entradas y 16 salidas digitales
- cable de conexión al puerto serie PC
- n.2 cables multipolares de conexión al sistema de control real

Para aumentar el número de I/O conectables pueden interconectarse hasta un máximo de 8 interfaces por medio de un anillo de fibra óptica.

El módulo PL-RIS

PL-RIS es el módulo de AW-SYS que permite conectar los PLCs simulados de PC-Sim a las instalaciones reales o bancos de prácticas presentes en su taller:

Para su funcionamiento es necesario disponer de los siguientes accesorios de hardware:

- interface de hardware para la conexión de 16 entradas y 16 salidas digitales
- cable de conexión a la puerta serial del PC
- n.2 cables multipolares de conexión al sistema de control real

Para aumentar el número de I/O conectables pueden interconectarse hasta un máximo de 8 interfaces por medio de un anillo de fibra óptica.

El módulo PL-Lab

Otro componente del mundo AW-SYS es PL-Lab, controlador de interface entre PC-Sim o PC-RIS e InTouch™.

Permite la conexión de los PLC simulados de PC-Sim o bien de los reales, a través de PC-RIS, con el conocido paquete de supervisión de Wonderware, por medio del cual podrán construir nuevas ventanas de simulación de instalaciones

PL-Lab contiene además la fuente de una instalación prototipo y algunos ejemplos de aplicación de instalaciones realizadas con InTouch™.

El módulo PL-Net

PL-Net es el módulo de AW-SYS que permite el control de una instalación real, con un elevado número de entradas/salidas, por medio de más PLCs simulados en ejecución sobre otros tantos ordenadores personales conectados en red.

Además, con PL-Net, los diversos PLCs simulados pueden compartir informaciones bajo la forma de word globales y permitir así la sincronización de las distintas partes de las instalaciones controladas por los diferentes PLC.

Por lo tanto, PL-Net permite la organización de un taller de automatización flexible, en el que cada alumno, o cada grupo de alumnos, podrá dedicarse a la programación de la automatización de una parte de la instalación para luego comprobar el resultado con los demás. En fases sucesivas, las tareas podrán intercambiarse, incluso continuando el trabajo con el mismo PC.

Guía on line

Además del presente manual, en caso de dudas pueden consultar la *Guía on line* de PC-Sim o las correspondientes a las instalaciones de PL-Sim.

Puede accederse a la Guía on line seleccionando el [Comando Guía](#) en el *Menú Archivo*.

Las guías de AW-SYS son *sensibles al contexto*: es decir, puede accederse a términos específicos de cada una de las guías directamente, seleccionando el elemento acerca del cual se desean las informaciones y pulsando la tecla F1

Son sensibles al contexto:

- los términos de los menús.
- las ventanas del programa;
- las ventanas de diálogo.
- muchas de las ventanas de mensaje.

En distintas ventanas está también activa la función de *Guía rápida* que le permite obtener con gran rapidez informaciones acerca de un componente específico de una ventana. Para activar esta función, seleccione el comando *Guía rápida* en el *Menú ?*, o bien pulse Mayusculas+F1, y haga clic sobre el elemento acerca del cual desea obtener información. Una ventana popup que contiene las informaciones solicitadas, aparecerá en la pantalla.

La activación de esta modalidad de guía se pone de manifiesto mediante el puntero del mouse en estado de *Guía rápida* (flecha con interrogación).

Servicio de Soporte Técnico para Usuarios de AW-SYS

Autoware pone a disposición de los poseedores de una copia registrada de AW-SYS, un servicio de soporte técnico que les permitirá una mejor utilización del programa.

Para resolver cualquier duda sobre el producto o problema que se haya producido al utilizarlo, consulte en primer lugar este manual o la *Guía on line*. Si no consigue encontrar las respuestas deseadas, contacte con el servicio de una de las siguientes maneras:

- telefónicamente, al número 099.7303413
- por fax, al número 099.7360062
- por correo electrónico, a la dirección de e-mail support@autoware.com

En la solicitud de información deberá especificar:

- la versión del programa.
- el tipo de hardware usado.
- el contenido exacto de los mensajes visualizados.
- la descripción de la operación que se estaba ejecutando cuando se ha producido el problema.
- el programa que ha elaborado y la automatización que se desea realizar.

El *Servicio de Soporte Técnico para Usuarios de AW-SYS* es **gratuito e ilimitado** para quienes hayan enviado la *Tarjeta de registr.*

Autoware en Internet

Autoware mantiene un sitio propio en Internet en la dirección:

<http://www.autoware.com/>

que ustedes pueden visitar para descargar las actualizaciones de su software AW-SYS.

La dirección directa de la página para la descarga, que podrán hacer constar entre sus direcciones preferidas, es:

<http://www.autoware.com/italian/support/upgrade.htm>

La página se actualiza con regularidad, incorporando las novedades del software, las últimas versiones de los programas y nuevos ejemplos de programación. Los programas que no son de demostración sólo podrán ser utilizados por los poseedores de un código hardware AW-SYS.

Otras informaciones sobre el software pueden solicitarse por correo electrónico a la dirección:

info@autoware.com

Estructura del manual

El presente manual constituye la guía para el uso de AW-SYS. Está organizado por capítulos y cada uno de ellos trata un tema específico, tal como se explica a continuación.

El [Capítulo 2](#) les guía en la instalación del software en su PC.

El [Capítulo 3](#) está totalmente dedicado a la descripción del simulador de PLC (PC-Sim) para los tres PLC simulados.

El [Capítulo 4](#) está dedicado a la descripción de cada una de las instalaciones que componen el segundo módulo del paquete: PL-Sim. En este capítulo, además, se facilitan las informaciones necesarias para la conexión de las instalaciones a los PLC simulados.

El [Capítulo 5](#) y el [Capítulo 6](#) están dedicados respectivamente al módulo PC-RIS y al módulo PL-RIS y a los procedimientos de puesta en funcionamiento de los sistemas.

En el [Capítulo 7](#) se describe como realizar instalaciones por su propia cuenta utilizando el software InTouch™ y cómo conectarlos a los PLC simulados con PL-Lab. Los procedimientos de generación de las instalaciones se describen con detalle y van acompañados de ejemplos prácticos para su realización.

En el [Capítulo 8](#) se presenta el módulo PL-Net y se describen los procedimientos para la conexión de instalaciones MPS y de puesta en funcionamiento del sistema.

El [Capítulo 9](#) está dedicado a la realización guiada de un ejercicio de automatización utilizando los distintos PLC simulados Siguiendo paso a paso las instrucciones que se facilitan, aunque no se tengan conocimientos específicos sobre la programación de PLC, se estará en condiciones de realizar la primera automatización con AW-SYS.

El [Capítulo 10](#) , el [Capítulo 12](#) y el [Capítulo 14](#) proporcionan los datos para la programación de los PLC simulados FPC 404, PS3 y S5 100U respectivamente.

El [Capítulo 11](#), el [Capítulo 13](#) y el [Capítulo 15](#) incluyen ejemplos de programación con un nivel de complejidad creciente, totalmente resueltos y comentados, para cada uno de los PLCs.

Para finalizar, en [Apéndice A](#) se incluye una descripción técnica de la interface de hardware EasyPort D16.

Una copia del presente manual está disponible en versión electrónica en el CD de AW-SYS. Para consultarlo e imprimirlo, haga doble clic en *Manual del usuario* en la carpeta Español\Manuales.

CAPÍTULO 2

INSTALACIÓN DE AW-SYS

Contenido del producto

El contenido del producto AW-SYS es diferente en función del número de licencias adquiridas, tal como se describe a continuación.

Licencia sencilla

El paquete del producto contiene:

- una copia del presente *Manual de uso de AW-SYS* en lengua española
- la *Tarjeta de registro*
- un CD de instalación de AW-SYS
- un código de hardware

Multilicencia de 6

El paquete del producto contiene:

- dos copias del presente *Manual de uso de AW-SYS* en lengua española
- la *Tarjeta de registro*
- dos CD de instalación de AW-SYS
- seis códigos de hardware

Multilicencia de 12

El paquete del producto contiene:

- dos copias del presente *Manual de uso de AW-SYS* en lengua española
- la *Tarjeta de registro*
- dos CD de instalación de AW-SYS
- doce códigos de hardware

Tarjeta de registro

Rellene todos los apartados de la tarjeta de registro y envíela a Autoware, a la siguiente dirección:

Autoware s.r.l. - Servicio clientes
Via Pio XII, 15 - 74100 TARANTO

Este hecho nos permitirá mantenerles informados sobre los desarrollos del producto, y al mismo tiempo les permitirá disfrutar del servicio de asistencia, tal y como se especifica en el apartado [Servicio de Soporte Técnico para Usuarios de AW-SYS](#).

Como alternativa, en la página Web de Autoware está activado el *Servicio de Registro on Line* en la dirección:

<http://www.autoware.com/spanish/regawsys.htm>

Control de los requisitos del sistema

Características mínimas

Antes de iniciar la instalación verifique que su sistema tienen las características mínimas relacionadas a continuación.

- CPU 486
- Windows 95
- mouse
- 16 Mbyte RAM
- tarjeta gráfica SVGA con una resolución gráfica de 800x600 pixel .

Características aconsejadas

A ser posible, cargue AW-SYS en un ordenador que tenga por lo menos las siguientes características aconsejadas:

- CPU Pentium
- Windows 95
- mouse
- 32 Mbyte RAM
- tarjeta gráfica SVGA con una resolución gráfica de 800x600 pixel
- tarjeta de sonido.

Instalación de AW-SYS

Arranque de la instalación

Inserte el CD de AW-SYS. El programa de instalación se iniciará automáticamente.

Si el arranque automático no funciona correctamente, ejecute el comando:

```
[drive]:\setup
```

donde [drive] es la letra identificativa de su lector de CD-ROM.

Elección del directorio

En este punto el programa les propone seleccionar la carpeta en la que debe instalarse AW-SYS. Pueden aceptar la ruta predeterminada, pulsando *Continuar*, o cambiar la ruta, tecleando una nueva, y seguidamente pulsar *Continuar*. El *Programa de instalación de AW-SYS* crea automáticamente la carpeta especificada.

Dentro de ésta se crean otras carpetas de las que el usuario no pueda elegir el nombre durante la fase de instalación y que tampoco podrá modificar más adelante. Estas carpetas son:

- Fpc404 contiene los archivos de PC-Sim para FPC 404
- Ps3 contiene los archivos de PC-Sim para PS3
- S5100 contiene los archivos de PC-Sim para S5 100U
- Pl-sim contiene los archivos de las instalaciones de PL-Sim y de los módulos PL-RIS, PL-Lab y PL-Net
- Pc-ris contiene los archivos PC-RIS

Por su parte, las tres primeras contienen otra carpeta:

- Ejemplos contiene los proyectos de los PLC facilitados con el paquete.

Instalación de los ejemplos

El *Programa de instalación de AW-SYS* les propone, además, elegir si desean instalar los ejemplos facilitados con el programa: respondan seleccionando u omitiendo la casilla de control.

Si es usted profesor y está cargando el programa en su ordenador, instale los ejemplos: le darán inmediatamente una idea de las enormes posibilidades de AW-SYS. En cambio, si está cargando el programa en el ordenador de sus alumnos, puede obviar la instalación de los ejemplos para no facilitarles soluciones de los ejercicios prácticos que les propondrá a lo largo del curso.

Copia de los archivos

El *Programa de instalación de AW-SYS* se encarga a continuación de copiar los archivos necesarios para el funcionamiento de AW-SYS en la ruta seleccionada y en el directorio de sistema de Windows así como de la creación de los archivos de inicio.

Creación del grupo y de los iconos

El *Programa de instalación de AW-SYS* crea automáticamente el grupo AW-SYS y, en su interior, los iconos de PC-Sim para cada PLC simulado y el de PC-RIS.

Desinstalación del programa

La desinstalación del programa puede efectuarse en entorno Windows a través del *Panel de control*.

- Abra 'Inicio-Configuración' (ó Mi PC) → Panel de control
- Haga doble clic sobre *Agregar o quitar programas*.
- Seleccione AW-SYS en la lista de las aplicaciones
- Haga clic sobre *Agregar o Quitar*.

CAPÍTULO 3

PC-SIM: EL SIMULADOR DE PLC

Introducción

Tal como hemos visto en el capítulo anterior, **AW-SYS** tiene una estructura modular, es decir, está constituido por diversos componentes de software y por unos pocos accesorios de hardware, cada uno de ellos con funciones específicas pero perfectamente integrados con el resto. PC-Sim constituye el módulo de **AW-SYS** dedicado a la simulación de Controladores Lógicos Programables.

Este módulo simula tres tipos de PLC:

- Festo FPC 404
- Klöckner Moeller PS3
- Siemens S5 100U

PC-Sim permite editar un programa utilizable para PLC en lenguajes Ladder o AWL para así resolver un ejercicio de automatización basado en alguna de las instalaciones de PL-Sim. El programa puede instalarse y hacerlo “funcionar” en el PLC simulado, y a continuación verificarlo utilizando los instrumentos de debug (depuración) y de visualización del estado de las variables. Una vez concluido su trabajo podrá imprimir la documentación de todo cuanto haya realizado: de un modo sistematizado dispondrá sobre papel, del programa, de la tabla de símbolos, las listas de referencias cruzadas y de otras muchas cosas, tal como ya profundizaremos en el apartado destinado a este tema.

Dado que en la práctica los PLC simulados presentan características de hardware, de sistema y de programación de software muy distintas entre sí, se ha intentado hacer lo más uniforme posible la interface utilizable de este módulo, utilizando el máximo posible, ventanas de programa muy parecidas o idénticas, con el fin de hacer más fácil el paso de un PLC simulado a otro.

Sin embargo, son excepciones expresas a este principio general la [Ventana Lista bloques \(S5 100U\)](#) y la [Ventana Asignación de preselectores \(FPC 404\)](#): la primera tan sólo se utiliza para el PLC S5 100U, a causa de las peculiaridades en su programación estructurada sobre las que más adelante profundizaremos y que lo diferencian de los otros dos. La segunda sólo puede abrirse utilizando la simulación del PLC FPC 404, a causa de la particular programación de los preselectores de los temporizadores y de los contadores de dicho PLC.

La ventana de programación AWL, utilizada para una de las modalidades de programación de los PLC PS3 y de S5 100U, muestra diferencias sustanciales en la presentación y en la flexibilidad de utilización en los dos casos. En cambio, la ventana de programación Ladder, precisamente a causa de la mayor uniformidad de este lenguaje, se presenta de un modo sustancialmente idéntico en los tres PLC, a excepción de los

distintos tipos de instrumentos disponibles para dibujar, en relación con la diversa implementación de temporizadores, contadores, etc.

Las *Ventana Tabla de símbolos* a la *Ventana Estado variables* se presentan de una manera idéntica, y obviamente cada referencia a un operando efectuado en ellas deberá tener en cuenta la sintaxis específica que cada uno de los PLC reconoce como correcta para el acceso a las variables de entrada/salida y de memoria.

También de manera idéntica se presentan la *Ventana Informaciones sobre el proyecto* y las ventanas de diálogo *Imprimir documentación*, *Buscar operando* y *Reemplazar operando*.

Por último, si bien gráficamente ofrecen la representación de máquinas distintas, las ventanas PLC se presentan de un modo sustancialmente uniforme, gracias a las opciones operadas por modelos y composiciones: de hecho, cada PLC posee 16 canales digitales de entrada, dotados de LED indicadores del estado y de interruptores de control, y 16 canales digitales de salida, dotados también de pilotos indicadores. De todos modos, se presentan algunas diferencias (incluso en las máquinas reales) en el número de canales analógicos.

En este capítulo afrontamos la descripción simultánea de los simuladores de los tres PLC. Se especificarán, cuando ello sea necesario, sus diferencias respecto al tema examinado, mencionando entre paréntesis los modelos de los PLC simulados a los cuales se aplican las consideraciones que se están haciendo. Si no se especifica ningún PLC, las consideraciones se aplicarán a los tres modelos. Si aun no se posee suficiente práctica con el PLC real correspondiente, será conveniente integrar todo lo mencionado en este capítulo con lo que se explica en aquellos dedicados a la programación y, si ello fuera preciso, con la información facilitada en los manuales de la casa constructora del dispositivo real.

Cuando no se indique lo contrario, en este capítulo se entenderá por 'PLC', el PLC simulado y no la instalación real.

Arranque del programa

Para arrancar PC-Sim haga clic sobre el botón de Inicio de Windows, abra la carpeta Programas\Aw-sys y, dentro de ésta, elija la opción correspondiente al simulador de PLC que desee iniciar. En cada ocasión sólo podrá ejecutarse una sola copia de PC-Sim: no podrá iniciarse una segunda instancia del programa aunque se refiera a otro PLC distinto. El arranque de PC-Sim también es imposible cuando PC-RIS está ya ejecutándose.

Cuando se arranca el programa aparece la *Ventana Informaciones sobre PC-Sim*. En ella se incluyen informaciones acerca del programa, el tipo de PLC y el número de versión, el número de autorización de la copia que se esté utilizando, el productor del software; además de informaciones sobre el número de licencias de la multilicencia de la cual forma parte su copia.

Cierre esta ventana pulsando la tecla Ok para empezar a trabajar con PC-Sim.

Composición de un proyecto

Todo el trabajo desarrollado utilizando PC-Sim está organizado en proyectos. Un proyecto contiene:

- el programa del usuario (en lenguaje AWL, ladder o ambos);
- los símbolos y sus asociaciones con los operandos absolutos;
- el listado de las variables de estado;
- las informaciones sobre el proyecto;
- las programaciones de preselectores (solo FPC 404);
- los textos de las etiquetas del PLC (solo S5 100U);
- las posiciones y las dimensiones de las ventanas *Lista bloques* (solo S5 100U), *Estado variables* y *Tabla de símbolos*.

Todas estas informaciones se guardan con el comando *Guardar Proyecto*, o con uno de los métodos equivalentes, en un archivo con extensión prj y vuelven a cargarse a la siguiente apertura del mismo proyecto.

Ventana Lista bloques (S5 100U)

La ventana *Lista Bloques* tan solo está presente en PC-Sim para S5 100U ya que este es el único de los PLC simulados que permite que el programa del usuario se estructure en bloques (ver [Programación estructurada](#) en el [Capítulo 14](#) para más informaciones).

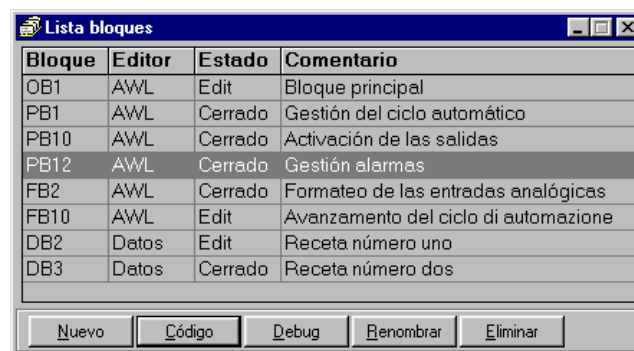
En esta ventana aparece, en forma de tabla, el listado de los bloques presentes en el proyecto, mostrando, para cada uno de ellos, el tipo de editor utilizado en la programación y su estado actual. Además, consiente la gestión de los bloques permitiendo crearlos, cambiar de nombre, eliminarlos, editar y el debug.

Apertura y cierre de la ventana

Cuando arranque PC-Sim, cuando cargue un proyecto ya existente, o cuando cree uno nuevo, la ventana *Lista bloques* se abre automáticamente. Podrá cerrarla haciendo clic sobre el botón de cierre situado en el extremo derecho de la barra del título y podrá reabrirla, o llevarla a primer plano con uno de los siguientes métodos:

- seleccionando el comando *Lista bloques* en el menú *Ventana*;
- pulsando la tecla *Ventana Lista bloques* en la [Barra instrumentos](#) de PC-Sim;
- pulsando la combinación de teclas Ctrl+B.

La ventana puede cambiar de tamaño o de posición a gusto del usuario. Las últimas programaciones de dimensión y posición se guardan con el archivo proyecto de manera que, al reabrir este último, PC-Sim se encargará de recuperar las últimas características de la ventana.



Bloque	Editor	Estado	Comentario
OB1	AWL	Edit	Bloque principal
PB1	AWL	Cerrado	Gestión del ciclo automático
PB10	AWL	Cerrado	Activación de las salidas
PB12	AWL	Cerrado	Gestión alarmas
FB2	AWL	Cerrado	Formateo de las entradas analógicas
FB10	AWL	Edit	Avanzamento del ciclo di automazione
DB2	Datos	Edit	Receta número uno
DB3	Datos	Cerrado	Receta número dos

Nuevo Código Debug Renombrar Eliminar

Composición de la ventana

La ventana *Lista bloques* está compuesta por las siguientes partes:

- Tabla de bloques
- Barra de pulsadores

A continuación les facilitamos una descripción detallada de dichos componentes.

Tabla de bloques

La *Tabla de bloques* contiene la lista de los bloques presentes en el proyecto y indica, para cada uno de ellos, el tipo de editor y su estado actual.

Haciendo clic sobre una fila de la tabla se realiza la selección y el bloque que se describe en esa fila se convierte en el bloque apuntado; es decir, la selección de la fila hace las funciones de puntero en el seno de la tabla. Cada clic sobre los pulsadores de la barra inferior, cuando están habilitados y con la excepción del pulsador *Nuevo*, actuará directamente tan sólo sobre ese bloque. Únicamente puede seleccionarse un bloque cada vez.

El puntero sobre el bloque actual se suspende automáticamente cada vez que se lleve a primer plano una ventana de editor o de debug; al variar la ventana en primer plano, se modifica la fila seleccionada en la tabla.

Cuando el listado de bloques no puede contenerse por entero en la ventana, PC-Sim muestra automáticamente una barra de desplazamiento a la derecha de la tabla que le permitirá desplazarse por su contenido y visualizar y seleccionar cada uno de los bloques del proyecto.

La tabla está formada por las cuatro columnas siguientes.

Columna Bloque

Indica las siglas de los bloques. Los bloques de PC-Sim para S5 100U pueden ser de tipo organizativo (**OB0** - **OB63**), de programa (**PB0** - **PB63**), funcionales (**FB0** - **FB63**) o de datos (**DB2** - **DB63**). Otras informaciones complementarias acerca de las características específicas de los distintos tipos de bloques pueden encontrarse en la opción [Programación estructurada](#) en el [Capítulo 14](#).

Columna Editor

Indica el tipo de lenguaje utilizado para la programación del bloque con los siguientes significados:

- **AWL** lenguaje AWL (Lista instrucciones)
- **Ladder** lenguaje Ladder diagram (Esquema de contactos)
- **Datos** ningún lenguaje, el bloque es de tipo datos.

Columna Estado

Indica el estado actual del bloque. El bloque puede encontrarse en uno de los tres estados siguientes:

- **Cerrado** el editor del bloque está cerrado
- **Abierto** el editor del bloque está abierto
- **Debug** cuando, con el PLC en On, está activo el debug sobre el bloque específico (ver la siguiente descripción del [Pulsador Debug](#)).

Columna Comentario

Indica el texto de comentario al bloque, tal como se escribió en el momento de la creación del bloque en cuestión (ver [Pulsador Nuevo](#) más adelante). Para modificar el texto del comentario, seleccione el bloque en la tabla y pulse el [Pulsador Renombrar](#).

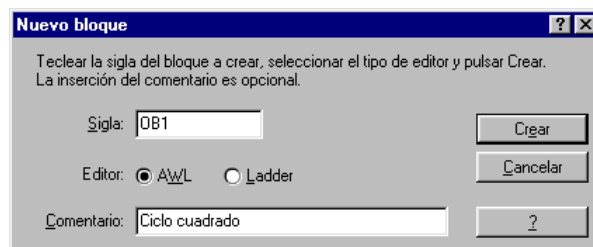
Barra de pulsadores

La parte inferior de la *Ventana Lista bloques* está constituida por la barra de pulsadores.

Haciendo clic sobre estos pulsadores pueden realizarse acciones sobre los bloques, en concreto sobre el que esté marcado en ese momento. Los pulsadores se desactivan por su cuenta cuando la acción relacionada con ellos resulta imposible o bien carece de sentido en aquel contexto o momento dado. Como alternativa a los pulsadores pueden usarse los comandos del [Menú de acceso rápido](#) activables haciendo clic sobre la tabla con el botón derecho del mouse.

Pulsador Nuevo

Permite crear un nuevo bloque y abrir su correspondiente editor. Haciendo clic sobre este pulsador se muestra la ventana de diálogo *Nuevo bloque*. Inserte en ella la sigla del bloque; si es necesario, haga clic sobre el pulsador con la opción para elegir el tipo de editor del bloque a crear (AWL o Ladder), y si lo cree necesario, inserte un texto de comentario en la casilla destinada a ello. A continuación pulse *Nuevo* para abrir el editor del nuevo bloque y añada su nombre al listado en la [Tabla de bloques](#). Los bloques se ordenan automáticamente según el tipo: primero los OB, luego los PB, a continuación los FB y por último los DB. En el interior de cada grupo la ordenación se efectúa en sentido creciente según el número del bloque.



Cuando inserte la sigla de un bloque de datos, el PC-Sim no dará ninguna importancia a la selección del *Editor* y se abrirá de todos modos, [El editor de datos \(S5 100U\)](#).

No es posible crear un bloque que tenga la misma sigla que otro ya presente en el proyecto. PC-Sim informa de dicha operación por medio de un mensaje de error.

Otros métodos equivalentes para crear un nuevo bloque son:

- seleccionar el [Comando Nuevo bloque \(S5 100U\)](#) en el [Menú Archivo](#);
- hacer clic sobre el botón *Nuevo bloque* de la [Barra instrumentos](#) de PC-Sim;
- pulsar la combinación de teclas Ctrl+N.

El pulsador *Nuevo* está desactivado cuando el PLC está en Run.

Pulsador Código

Este pulsador permite la visualización del código seleccionado en la tabla, abriendo su correspondiente editor. Haciendo clic sobre este pulsador se abre el tipo de editor previsto para el bloque (AWL, Ladder o datos) y, directamente, se carga y se muestra el código que aparece en la memoria del programa.

El pulsador *Código* se desactiva cuando el proyecto en uso no contiene ningún bloque.

Otro método equivalente para abrir un bloque ya existente es escoger el [Comando Abrir bloque \(S5 100U\)](#) del [Menú Archivo](#).

Si desea abrir simultáneamente todos los bloques que componen su proyecto, evitando repetir la selección y el clic para todos ellos, sea cual sea el bloque seleccionado, pulse y mantenga presionada la tecla Mayús y haga clic sobre el *Pulsador Código*, y a continuación suéltelos. Todos los bloques de su proyecto se abrirán y aparecerán en cascada en su pantalla.

Pulsador Debug

Un clic sobre este pulsador le permite pasar al debug sobre el bloque seleccionado en la [Tabla de bloques](#). Con PC-Sim puede activarse el debug en todos los bloques que desee, ¡simultáneamente! (Para más información ver [Debug en las ventanas Ladder](#) y [Debug en las ventanas AWL](#))

El debug sobre el bloque puede desactivarse en cualquier momento haciendo clic sobre [Pulsador Código](#).

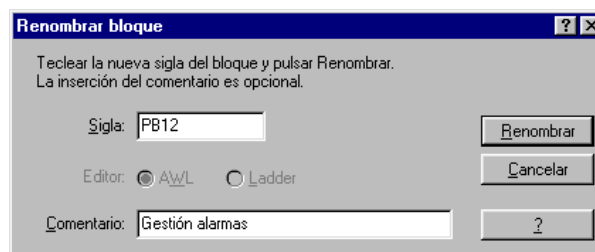
La elaboración del debug, sobre uno o más bloques, obliga al procesador de su ordenador a una actividad inmediata y puede provocar que la ejecución del programa sea más lenta, y en algunas ocasiones de manera sensible, en especial si el número de bloques en debug es elevado. Por lo tanto, cuando ya no sea necesario, desactive esta función haciendo que la elaboración recupere su velocidad original.

El pulsador *Debug* tan sólo se activa cuando la CPU del PLC recibe alimentación (ver los comandos del [Menú CPU](#)).

Pulsador Renombrar

El pulsador *Renombrar* le permite cambiar la sigla o el texto del comentario de un bloque, o ambas cosas.

Para cambiar el nombre: seleccione en la [Tabla de bloques](#) el bloque deseado y haga clic sobre el pulsador *Renombrar*.



PC-Sim le muestra la ventana de dialogo *Renombrar bloque*. Introduzca las modificaciones en la casilla *Bloque* y/o en la casilla *Comentario* y a continuación haga clic sobre el pulsador *Renombrar*. No se puede cambiar el tipo de bloque, de manera que los pulsadores de las opciones correspondientes se mantienen desactivados.

PC-Sim se ocupa de cambiar el nombre del bloque en la [Ventana Lista bloques \(S5 100U\)](#) y en la memoria de programa. Si ha cambiado la sigla del bloque, se le preguntará si desea actualizar todas las llamadas al bloque presentes en el programa. Si responde afirmativamente, todas las instrucciones de llamada al bloque renombrado (SPA o SPB) presentes en el programa se actualizarán a la nueva sigla.

Existen algunas limitaciones en el procedimiento para el cambio de la sigla de un bloque. De hecho, no se puede cambiar la sigla si:

- en el proyecto ya existe un bloque con la nueva sigla.
- el bloque es del tipo datos y se teclea la sigla de un bloque de otro tipo.
- el bloque no es del tipo datos y se teclea la sigla de un bloque de datos.
- el bloque está programado en ladder y se escribe la sigla de un bloque funcional.
- el bloque es de tipo funcional y parametrizado y se escribe la sigla de un bloque organizativo o de programa.
- el bloque es de tipo funcional y contiene operaciones que pertenecen al set integrativo y si escribe la sigla de un bloque organizativo o de programa.

Puede encontrar más aclaraciones a estas limitaciones en [Notas sobre la programación del PLC S5 100U](#) en el [Capítulo 14](#).

El pulsador *Renombrar* está desactivado cuando el editor del bloque en uso está abierto o cuando el PLC está en Run (ver los comandos del [Menú CPU](#)).

Pulsador Eliminar

Utilice este pulsador para eliminar un bloque del proyecto en uso.

Seleccione el bloque en la [Tabla de bloques](#), a continuación haga clic sobre el pulsador *Eliminar*. PC-Sim le pedirá que confirme su intención de eliminar el bloque con la correspondiente ventana de diálogo. Si responde afirmativamente el bloque será eliminado de la tabla y de la memoria del programa y, si estuviera abierto, el editor del bloque se cerrará.

Si en otros bloques aparecen llamadas (SPA o SPB) al bloque a eliminar, PC-Sim le advierte de esta situación con el correspondiente mensaje y le pide que vuelva a confirmar su eliminación. Si responde afirmativamente, el bloque será eliminado, pero el programa que haya escrito contendrá llamadas a un bloque inexistente. Si desea buscar todas las llamadas al bloque eliminado presentes en el programa use el [Comando Buscar operando...](#) del [Menú Modificar](#), tecleando, en la casilla *Buscar*, la sigla del bloque eliminado.

El pulsador *Eliminar* queda desactivado cuando en el proyecto no aparece ningún bloque o el PLC está en estado de Run (ver los comandos del [Menú CPU](#)).

Otro método equivalente para eliminar un bloque es seleccionar el [Comando Eliminar bloque \(S5 100U\)](#) del [Menú Archivo](#).

Menú de acceso rápido

La *Tabla de bloques* está dotada de un menú de acceso rápido. Algunas de las selecciones replantean los comandos de los pulsadores de la barra. Para activar un comando, haga clic con el botón derecho del mouse sobre la fila de la tabla que contiene el bloque en el que va a actuar y a continuación elija el comando deseado en el menú que aparece.

Para las explicaciones de los comandos *Nuevo*, *Código*, *Debug*, *Renombrar* y *Eliminar*, lea todo lo citado a propósito de los mismos pulsadores de la *Barra de pulsadores*.

El comando *Imprimir* facilita una modalidad rápida para la impresión del bloque marcado sin tener que recurrir al *Comando Imprimir documentación...* del *Menú Archivo*. El bloque seleccionado se imprime utilizando el tipo de operandos, absolutos o símbolos, especificado por la modalidad de visualización en uso (ver *Comando Mostrar símbolos* y *Comando Mostrar operandos absolutos*).

Por último, el comando *Barra pulsadores* permite esconder o visualizar toda la barra. Si prefiere utilizar el menú de acceso rápido o si desea ganar espacio en la pantalla, oculte la barra seleccionando este comando. Para visualizar nuevamente la barra, seleccione de nuevo el comando *Barra pulsadores*.

Si la tabla está vacía o si hace clic fuera de una de las líneas de los bloques, los únicos comandos que se visualizarán serán *Nuevo* y *Barra pulsadores*. De hecho, los demás están estrechamente ligados a un bloque determinado sobre el que operar y por tanto no pueden elegirse si no hay ningún bloque seleccionado.

El editor Ladder

Premisa

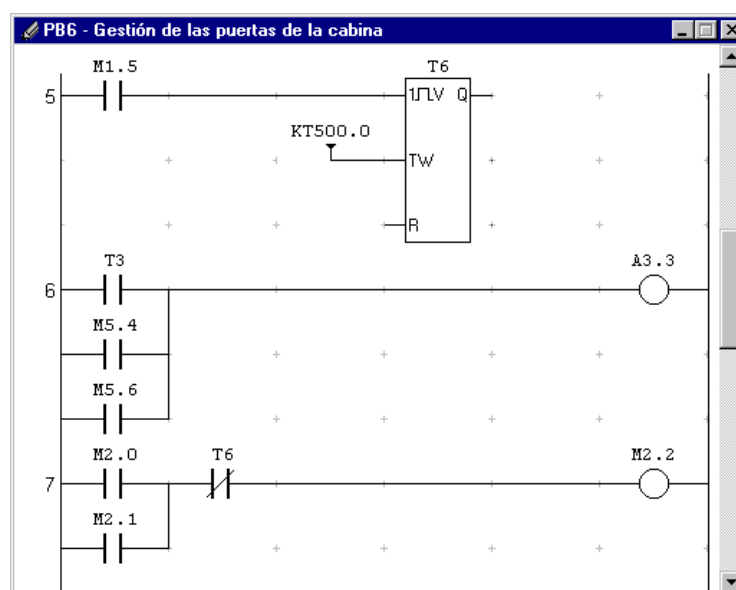
Esta descripción no es y no puede ser una lección, y mucho menos un curso, acerca de la utilización del lenguaje ladder para la programación de los PLC. Utilice textos especializados o el manual del programador de su PLC real para aprender los fundamentos de este lenguaje. De todos modos, PC-Sim le permitirá un conocimiento más rápido dada la posibilidad que ofrece de experimentar inmediata y fácilmente con el programa realizado.

También le serán muy útiles los últimos capítulos del presente manual. En ellos aparecen los PLC simulados con una especial atención a sus peculiaridades de programación y se facilita además una interesante serie de ejercicios de programación resueltos y comentados. Recuerde que Autoware está siempre a su disposición para más aclaraciones. Las distintas maneras de conectar con nosotros se describen en el apartado dedicado al [Servicio de Soporte Técnico para Usuarios de AW-SYS](#) en el [Capítulo 1](#).

La programación en ladder

El Ladder diagram o Esquema de contactos es sin duda el lenguaje más difundido de programación de PLC. Representa una metodología de programación gráfica. Entre dos líneas verticales, que representan la alimentación, se diseña un esquema eléctrico compuesto por diversos circuitos dispuestos sucesivamente. El PLC ejecuta por orden cada uno de ellos durante la elaboración del programa del usuario.

El editor ladder de PC-Sim le ofrece la posibilidad de escribir cómoda y rápidamente el programa, uniendo a la velocidad de las acciones con el mouse, la potencia de los comandos típicos de editor avanzados: por ejemplo los comandos cortar, copiar y pegar.



El diseño del esquema se realiza utilizando los instrumentos de la *Casilla instrumentos* y los *Menú de acceso rápido* activables haciendo clic con el botón derecho del mouse en el interior del esquema. La comprobación de la corrección sintáctica de los operandos, que podrán introducirse también en forma de símbolo, se efectúa al final de cada escritura. El mismo editor permitirá, durante la ejecución del programa realizado, la visualización del debug. Se mostrarán, con variaciones de colores, los contactos cerrados, las bobinas activas y la circulación de la corriente. La coloración se irá actualizando en tiempo real y como consecuencia de los resultados de la elaboración y de las variaciones de las señales visibles.

Otras informaciones acerca de las modalidades de programación ladder se facilitan en los apartados *Programación en ladder* en los capítulos sobre notas de programación de los distintos PLC.

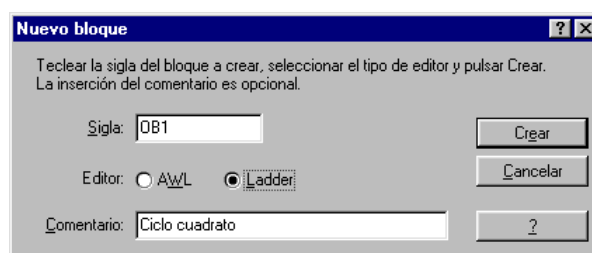
Apertura del editor ladder (S5 100U)

Para abrir el editor de un bloque ladder existente, seleccione el bloque en el interior de la *Tabla de bloques* y pulse el *Pulsador Código*, o bien haga doble clic sobre la fila de la tabla que contiene el bloque a editar.

En cambio, para crear un nuevo bloque ladder, abra también el editor y utilice uno de los siguientes procedimientos:

- Elija el *Comando Nuevo bloque (S5 100U)* en el *Menú Archivo*;
- haga clic sobre el *Pulsador Nuevo bloque* de la *Barra instrumentos* de PC-Sim;
- haga clic sobre *Pulsador Nuevo* de la *Ventana Lista bloques (S5 100U)*.

Los tres procedimientos son equivalentes y, a todos ellos, PC-Sim responde mostrando la ventana de diálogo *Nuevo bloque*. En dicha ventana, escriba la sigla del bloque a crear, elija el tipo de editor ladder como código de programación a adoptar para la escritura del bloque; si lo desea, inserte un comentario al bloque y, por último, pulse *Nuevo*.



El editor ladder del nuevo bloque se abrirá y la sigla de este último se añadirá en la *Tabla de bloques*.

A la apertura de la ventana de editor se le propondrá, ya incorporado, un primer recorrido de corriente en el que está presente tan sólo la bobina.

Apertura del editor ladder (FPC 404 y PS3)

Para abrir el editor del programa ladder existente, elija el *Comando Código (FPC 404 y PS3)* del *Menú Ventana* o bien haga clic sobre el *Pulsador Ventana Código* de la *Barra instrumentos* de PC-Sim.

Para crear un nuevo proyecto, usando la programación en ladder, elija el [Comando Nuevo proyecto ladder \(FPC 404 y PS3\)](#) del [Menú Archivo](#) o bien haga clic sobre el [Pulsador Nuevo proyecto ladder](#) de la [Barra instrumentos](#) de PC-Sim

Cierre del editor ladder (S5 100U)

El cierre del editor Ladder puede efectuarse accionando el pulsador de cierre situado a la derecha de la barra del título. Si se han efectuado modificaciones en el esquema, PC-Sim le solicitará, por medio de la correspondiente ventana de diálogo, si desea guardarlas.

En caso de que responda afirmativamente, recuerde que el código del bloque, al ser modificado, no se guarda en el archivo proyecto sino en la memoria de programa. Esta última, que incluye el código de todos los bloques de su proyecto, ya sea ladder, AWL o de datos, se registrará en el archivo únicamente si se especifica expresamente que el proyecto se guarde. Respecto a ello, ver el [Comando Guardar proyecto](#) del [Menú Archivo](#).

Si responde negativamente a la petición de guardar, las últimas modificaciones efectuadas en el bloque no se guardarán en la memoria del programa, es decir, se perderán. Si el bloque no se había guardado nunca desde su creación, la operación de cierre y la elección de no guardar las modificaciones conducirán a la pérdida total del bloque, que desaparecerá también de la [Tabla de bloques](#).

Un método equivalente para cerrar el bloque consiste en pulsar la combinación de teclas Alt+F4.

Cierre del editor ladder (FPC 404 y PS3)

El cierre del editor Ladder puede efectuarse accionando el pulsador de cierre situado a la derecha de la barra del título. Si se han efectuado modificaciones en el esquema, PC-Sim le solicitará, por medio de la correspondiente ventana de diálogo, si desea guardarlas.

En el caso de que responda afirmativamente, recuerde que al haber modificado el código, éste se guarda en la memoria del programa y no en el archivo proyecto. La memoria de programa, que contiene el código del programa de su proyecto Ladder, se registrará en el archivo proyecto solo si se especifica que se guarde concretamente en éste. Respecto a ello, ver el [Comando Guardar proyecto](#) del [Menú Archivo](#).

Si responde negativamente a la petición de guardar, las últimas modificaciones efectuadas en el bloque no se guardarán en la memoria del programa, es decir, se perderán.

Un método equivalente para cerrar el bloque consiste en pulsar la combinación de teclas Alt+F4.

Guardar el bloque ladder (S5 100U)

Además del cierre del editor, tal como acabamos de ver, puede guardarse en cualquier momento el contenido de un bloque, seleccionando el [Comando Guardar bloque \(S5 100U\)](#) en el [Menú Archivo](#). También en este caso son válidas las consideraciones hechas en los apartados anteriores respecto a cómo guardar los bloques y, para más información, puede consultar la descripción del [Comando Guardar proyecto](#) del [Menú Archivo](#).

Casilla instrumentos

La *Casilla instrumentos* contiene las “herramientas” que le permitirán diseñar cómoda y rápidamente sobre su esquema Ladder. La selección del instrumento a utilizar se efectúa haciendo clic sobre él.



























Cuando cree un nuevo bloque ladder (S5 100U) o cuando cree un nuevo proyecto ladder (FPC 404 y PS3), PC-Sim se encarga automáticamente de abrir la casilla. Cuando abre un bloque Ladder existente con la intención de modificarlo, solicite la *Casilla instrumentos* con uno de los siguientes métodos:

- Elija el *Comando Casilla instrumentos* en el *Menú Ventana*;
- haga clic sobre el pulsador *Casilla instrumentos* de la *Barra instrumentos* de PC-Sim;
- haga clic sobre uno de los elementos del esquema con el botón derecho y elija el comando *Casilla instrumentos* en el *Menú de acceso rápido* que aparece.

La *Tabla 1* muestra el listado de los pulsadores disponibles para cada PLC con una breve descripción.

Tabla 1: Listado de los instrumentos disponibles en la Casilla instrumentos

Pulsador	Descripción	FPC 404	PS3	S5 100U
	Selección del recorrido	✓	✓	✓
	Eliminación contacto, recorrido, ramal paralelo, etc.	✓	✓	✓
	Construcción de un ramal en paralelo	✓	✓	✓
	Contacto normalmente abierto	✓	✓	✓
	Contacto normalmente cerrado	✓	✓	✓
	Bobina directa	✓	✓	✓
	Bobina inversa	✓	✓	
	Bobina de set (activación)	✓	✓	✓
	Bobina de reset (desactivación)	✓	✓	✓
	Bobina de incremento	✓		
	Bobina de decremento	✓		
	Bobina de preselector	✓		
	Temporizador		✓	
	Temporizador a impulso SI			✓
	Temporizador a impulso prolongado SV			✓
	Temporizador con retardo a la activación SE			✓
	Temporizador retardo a la activación con memoria SS			✓
	Temporizador con retardo a la desactivación SA			✓
	Contador		✓	✓
	Registro de desplazamiento		✓	
	Comparador		✓	
	Salto incondicional		✓	
	Salto condicionado sobre RA=1		✓	
	Salto condicionado sobre RA=0		✓	

Edición del esquema ladder

En los apartados siguientes encontrará las indicaciones acerca de cómo utilizar los instrumentos de la [Casilla instrumentos](#) y las descripciones de otras opciones ofrecidas por el editor ladder, aunque en la mayoría de los casos, su utilización es tan intuitiva que casi no requeriría aclaraciones.

Introducción de un contacto NA o NC

En la [Casilla instrumentos](#) seleccione el instrumento de dibujo de contactos normalmente abiertos o el de dibujo de contactos normalmente cerrados. A continuación sitúe el puntero del mouse en el editor ladder, señalando la posición de introducción. En ese momento haga clic para obtener la escritura del contacto.

Después de la primera escritura, la función de introducción del contacto elegida por usted permanece activa para permitirle inserciones múltiples sucesivas de una manera extremadamente rápida: basta con que siga situando el puntero y haciendo clic.

Los contactos normalmente abiertos o cerrados tan sólo pueden colocarse en las cinco primeras posiciones de cada ramal. Tras la introducción de un contacto, el campo operando se visualiza automáticamente y el puntero de inserción de texto se sitúa ya encima de éste para permitirle la inmediata inserción de la sigla.. Podrá teclear enseguida el operando relativo al contacto en cuanto lo haya introducido, o bien continuar dibujando los contactos e insertar después todos los operandos. La elección es suya.

Igualmente sencillo es sobrescribir un tipo de contacto con otro, por ejemplo, transformar un contacto NA en uno NC. También en esta ocasión bastará con seleccionar el instrumento del contacto a insertar y luego hacer clic sobre la posición del contacto a sobrescribir. La sobrescritura de un contacto no modifica el operando.

La escritura o sobrescritura de un contacto puede efectuarse también utilizando los [Menú de acceso rápido](#), haciendo clic con el botón derecho del mouse sobre la posición de inserción deseada y seleccionando a continuación la correspondiente opción del menú.

Eliminación de contactos

Seleccione el instrumento de eliminación en la [Casilla instrumentos](#) y a continuación haga clic sobre el contacto a borrar. Tanto el contacto como el operando asociado serán eliminados del esquema.

También en este caso, la operación puede efectuarse solicitando, tal como ya hemos descrito, el menú de acceso rápido sobre el contacto a eliminar.

Sobrescritura de bobinas

En la última posición de cada nuevo recorrido generado en el editor ladder se inserta automáticamente la bobina predeterminada: la directa. Si necesita otro tipo de bobina (inversa, set, reset, incremento o decremento en función del PLC que esté utilizando) seleccione el instrumento del tipo de bobina que desee y luego, en el esquema, haga clic sobre la bobina que pretende cambiar. La operación de sobrescritura de bobina no modifica el operando.

También en este caso la operación puede efectuarse solicitando, tal como ya se ha descrito, el [Menú de acceso rápido](#) sobre la bobina a cambiar.

Introducción de los operandos

La introducción de la sigla de un operando en el editor ladder se efectúa en el campo operando, situado inmediatamente encima del respectivo elemento: contacto, bobina, temporizador, etc. Cuando está presente un elemento de esquema pero el operando correspondiente aun no se ha escrito, el campo aparece con un fondo gris claro, para recordarle que la operación de inserción no se ha completado. Para introducir el operando haga clic sobre el campo, escriba la sigla del operando absoluto o del símbolo al cual se refiere el elemento situado debajo y pulse Intro. Consulte los capítulos dedicados a las notas de programación de los PLC para un listado de los operandos absolutos válidos.

Si teclea un símbolo recuerde que como máximo puede estar formado por ocho caracteres alfanuméricos y que debe ir precedido por el carácter '-'.

El operando introducido se convierte automáticamente, si es necesario, en la modalidad de visualización en uso: es decir, por ejemplo, si está activa la visualización de símbolos e introduce un operando absoluto, éste será substituido por el correspondiente símbolo, si existe uno. Lo contrario sucede si está activa la visualización de los operandos absolutos y escribe un símbolo declarado en la [Tabla de símbolos](#).

La introducción de un operando absoluto erróneo o de un símbolo no declarado se pone de manifiesto, una vez se ha acabado de escribir, con el paso del texto a caracteres rojos.

Puede moverse rápidamente de un operando a otro: hacia adelante, pulsando la tecla Intro o el tabulador, hacia atrás pulsando la combinación de teclas Mayús+Tab.

El editor ladder de PC-Sim le ayuda a acelerar la fase de inserción de los operandos y, por lo tanto, de la escritura del programa: cada vez que inserte un nuevo elemento, el cursor de inserción de texto se sitúa automáticamente en el campo operando de este último. De modo que le bastará con teclear el nombre del operando inmediatamente después de haber introducido el elemento.

La introducción o la variación de un operando conduce automáticamente a la selección de la función de [Selección de recorrido](#) en la [Casilla instrumentos](#).

Agregar un nuevo recorrido

Para agregar un nuevo recorrido seleccione uno de los instrumentos de inserción de contactos en la [Casilla instrumentos](#) y, a continuación, haga clic sobre la primera fila vacía al final del esquema, en una de las cinco primeras posiciones de inserción. Obtendrá así la escritura de un nuevo recorrido, provisto ya de la bobina y del contacto que haya elegido en la posición sobre la que ha hecho clic. O bien, seleccione en la [Casilla instrumentos](#) uno de los instrumentos de inserción de bobinas y haga clic sobre la primera fila vacía situada al final del esquema, en la última posición de inserción.

Si después del último recorrido la primera fila libre del editor no fuera visible, mueva la ventana utilizando la barra de desplazamiento.

Otros métodos alternativos para agregar un recorrido al final del esquema son:

- hacer clic con el botón derecho sobre uno de los números de recorrido y elegir el comando *Añadir* en el *Menú de acceso rápido*
- elegir el *Comando Añadir* en el *Menú Modificar*

Inserción de un nuevo recorrido

Para insertar un nuevo recorrido en una posición determinada del esquema ladder que está dibujando, seleccione el recorrido anterior al que desea insertar, haciendo clic con el botón derecho sobre su número identificativo, y elija el comando *Insertar* en el *Menú de acceso rápido*: un nuevo recorrido se insertará en la posición indicada y los siguientes se renumerarán hacia abajo.

Como método alternativo, seleccione el recorrido haciendo clic sobre su número identificativo y elija el *Comando Insertar* en el *Menú Modificar*.

Introducción de módulos de sistema (PS3) o de temporizadores y contadores (S5 100U)

Seleccione el instrumento deseado en la *Casilla instrumentos* y a continuación haga clic en la posición de inserción en el editor ladder. Tan sólo se permiten las posiciones de la 2 a la 4

El recorrido no debe tener ramales en paralelo ni paralelos de bobinas.

Para el PS3, además, valen las siguientes limitaciones:

- el recorrido no debe tener ramales en paralelo ni paralelos de bobinas
- sobre el ramal no debe existir más de un contacto y, en todo caso, a la izquierda de la posición de introducción del módulo de sistema
- la bobina presente sobre el ramal tan sólo puede ser directa o inversa.

Creación de ramales de entrada y de salida a módulos de sistema (PS3) o de temporizadores y contadores (S5 100U)

Cuando inserte un temporizador en un esquema ladder, el ramal sobre el cual ha realizado la inserción se dividirá en dos: la parte izquierda actuará como ramal de entrada para el terminal de set y la parte derecha, como ramal de salida del propio temporizador. Análogas consideraciones pueden hacerse para la inserción de contadores o de los otros módulos de sistema para el PS3.

Si necesita utilizar las otras entradas o las otras salidas (donde existan) deberá crear, para cada una de ellas, un ramal de entrada o salida al módulo.

Para crear un ramal de entrada, seleccione uno de los instrumentos de inserción de contactos y luego haga clic sobre el esquema ladder, situado a la izquierda del módulo, en correspondencia con la entrada que desee insertar. Obtendrá así la inserción de un nuevo ramal de entrada, provisto ya del contacto que ha elegido en la posición sobre la que ha hecho clic.

Del mismo modo, para crear un ramal de salida a un módulo, seleccione uno de los instrumentos de inserción de bobinas y a continuación haga clic sobre el editor que corresponda a la última posición de inserción y de la fila de salida al módulo que desea crear.

Otro método equivalente consiste en elegir entre los comandos presentes en el *Menú de acceso rápido*, que aparece haciendo clic con el botón derecho sobre el módulo, el que corresponde a la acción que pretende realizar.

Eliminación de ramales de entrada y de salida a módulos de sistema (PS3) o de temporizadores y contadores (S5 100U)

Para eliminar ramales de entrada o salida a un temporizador o contador (S5 100U) o a un módulo de sistema (PS3), seleccione el instrumento de eliminación en la *Casilla instrumentos* y a continuación haga clic sobre el ramal a eliminar.

Si la posición que ha marcado sobre el editor Ladder no contiene ningún contacto, se obtendrá la eliminación inmediata del ramal. En cambio, si la posición contiene un contacto, al primer clic se obtendrá la eliminación del contacto y será necesario un segundo clic para la completa eliminación del ramal.

Un método equivalente consiste en abrir el menú de acceso rápido haciendo clic con el botón derecho sobre el módulo y eligiendo entre los comandos presentes en él, el que corresponda a la eliminación que pretende efectuar.

Eliminación de un recorrido

Para eliminar un recorrido completo, seleccione el instrumento de eliminación en la *Casilla instrumentos* y a continuación haga clic sobre el número identificativo del recorrido, situado a su izquierda. PC-Sim eliminará el recorrido y se ocupará de ordenar y reenumerar de manera progresiva los siguientes. La eliminación de un recorrido completo comporta la eliminación del ramal base y de todos los ramales paralelos conectados a él.

Como alternativa:

- active el *Menú de acceso rápido* sobre el número de recorrido y elija el comando *Eliminar*
- seleccione el recorrido y elija el *Comando Eliminar* en el *Menú Modificar*.

Eliminación de módulos de sistema (PS3) o de temporizadores y contadores (S5 100U)

Seleccione el instrumento de eliminación en la *Casilla instrumentos* y a continuación haga clic sobre el temporizador, el contador (S5 100U) o sobre el módulo de sistema (PS3) a eliminar. PC-Sim se ocupará de eliminar el módulo y todos los ramales de entrada y de salida siguientes al primero que estén conectados a él.

Si quiere eliminar el módulo y todos los ramales, incluido el primero, siga el procedimiento descrito en el apartado *Eliminación de un recorrido*.

Como alternativa, y como siempre, utilice el menú de acceso rápido del módulo, seleccionando el comando *Eliminar*.

Edición avanzada sobre recorridos

Con PC-Sim pueden realizarse operaciones de edición sobre recorridos completos. De hecho, los comandos del [Menú Modificar](#), o los de los [Menú de acceso rápido](#) sobre los números de recorrido, permiten cortar, copiar y pegar recorridos en un mismo programa Ladder, o entre distintos bloques ladder del mismo proyecto (S5 100U) o de proyectos distintos.

Las operaciones se efectúan sobre el recorrido seleccionado, es decir, el que está rodeado por un marco de color azul. Para seleccionar un recorrido haga clic con el botón izquierdo del mouse sobre su número identificativo. Un nuevo clic sobre el mismo número eliminará la selección.

Para más información acerca de las operaciones de edición sobre recorridos, ver las descripciones facilitadas para el [Menú Modificar](#) y todo lo dicho para los [Menú de acceso rápido](#).

Creación de un ramal paralelo

Para crear un ramal paralelo, seleccione el instrumento para la creación de paralelos en la [Casilla instrumentos](#); a continuación pulse el botón izquierdo del mouse sobre el ramal a crear el paralelo en correspondencia con la posición inicial de construcción del paralelo y, manteniendo pulsado el botón, arrastre el puntero del mouse hasta la posición final. Tras ello, suelte el botón del mouse.

Durante el arrastre PC-Sim dibuja, con líneas esbozadas, la forma del ramal paralelo que está construyendo, para ayudarle en dicha operación.

Lea el apartado [Limitaciones del editor ladder](#) para obtener informaciones acerca de las restricciones en la operación de construcción de un ramal paralelo.

Paralelo de una bobina

Para crear un paralelo a una bobina, seleccione el instrumento de creación de paralelos en la [Casilla instrumentos](#) y luego haga clic sobre la bobina a la que desea crear el paralelo. PC-Sim se ocupa de dibujar el paralelo y de insertar sobre él la bobina predeterminada: la directa. Si desea otro tipo de bobina siga las instrucciones descritas con anterioridad en el apartado [Sobreescritura de bobinas](#). PC-Sim sitúa inmediatamente el cursor de inserción de texto sobre el operando de la bobina recién insertada para permitirle así una rápida escritura del operando.

Remítase al apartado [Limitaciones del editor ladder](#) para informaciones acerca de las restricciones de la operación de paralelo de una bobina.

Eliminación de ramal paralelo

Para eliminar un ramal paralelo seleccione el instrumento de eliminación en la [Casilla instrumentos](#), y a continuación haga clic sobre el ramal paralelo a eliminar.

Si la posición que ha señalado no contiene ningún contacto, obtendrá la eliminación inmediata del paralelo. Si, por el contrario, la posición contiene un contacto, al primer clic se efectuará la eliminación del contacto y será necesario un segundo clic para la eliminación del paralelo.

Remítase al apartado *Limitaciones del editor ladder* en este mismo capítulo, para informaciones sobre los casos en los que es posible eliminar un ramal paralelo.

Punteros del mouse

Durante la edición del esquema ladder, el puntero del mouse realiza una importante función. Asumiendo en cada ocasión un aspecto distinto, en función del instrumento que ha seleccionado en la *Casilla instrumentos*, el puntero le recuerda la operación que va a realizar sobre el esquema. Así, por ejemplo, PC-Sim mostrará un cursor para la inserción de contactos NA, otro para los NC, otro para la eliminación y así sucesivamente, en función del instrumento que haya seleccionado en la casilla.

Menú de acceso rápido

En todo el editor ladder pueden activarse los menús de acceso rápido haciendo clic con el botón derecho en correspondencia con un elemento del esquema. Así, desde el menú que aparece cerca de la posición del puntero del mouse, podrá activar rápidamente los comandos más comunes para el elemento seleccionado.

Además de todos los símbolos del editor ladder (líneas de corriente, contactos, bobinas, etc.), también son sensibles al clic del botón derecho los números identificativos de los recorridos.

Acceso rápido sobre líneas de corriente

El menú está compuesto por los comandos *Contacto NA* y *Contacto NC* que permiten insertar el contacto especificado en la posición sobre la que se ha hecho clic.

Acceso rápido sobre contactos

Son activos el comando *Eliminar*, que borra el elemento sobre el que se ha hecho clic y su operando, y el comando *Invertir*, que invierte el tipo de contacto (de NA a NC y viceversa) sin alterar el operando.

Acceso rápido sobre bobinas

Un clic con el botón derecho del mouse sobre una bobina, activa un menú de acceso rápido compuesto por los siguientes comandos:

- Bobina directa
- Bobina inversa (FPC 404 y PS3)
- Bobina de set (activar)
- Bobina de reset (desactivar)
- Bobina de incremento (FPC 404)
- Bobina de decremento (FPC 404)
- Bobina de preselector (FPC 404)
- Jump con subcomandos JP, JC y JCN (PS3)

Estos permiten transformar la bobina sobre la que se ha hecho clic en el tipo especificado por el comando elegido.

Acceso rápido sobre módulos de sistema (PS3) o sobre temporizadores y contadores (S5 100U)

Los comandos activos son:

- Eliminar
- Añadir ramal
- Eliminar ramal

El primero elimina por completo el módulo de sistema, el temporizador o el contador.

Los demás se utilizan para borrar o dibujar ramales de entrada o de salida. Estos comandos, que se repiten para cada uno de los ramales de entrada y salida del módulo, tan sólo son visibles si es necesario. Es decir, por ejemplo, el comando *Eliminar ramal salida* será visible sólo si está presente ese específico ramal de salida. En el caso contrario, será visible el comando *Añadir ramal salida*.

Acceso rápido sobre un número de recorrido

Haciendo clic con el botón derecho del mouse sobre el número identificativo de un recorrido se activa un menú de acceso rápido compuesto por las siguientes opciones.

Eliminar:	elimina del esquema el recorrido seleccionado.
Insertar:	inserta un nuevo recorrido antes del seleccionado.
Añadir:	añade un nuevo recorrido al final del esquema.
Cortar:	elimina el recorrido seleccionado creando una copia en la memoria.
Copiar:	crea una copia en la memoria del recorrido seleccionado.
Pegar:	inserta antes de la selección en uso el recorrido copiado o cortado; si no hay ninguna selección activa, el recorrido se añade al final del esquema.

Limitaciones del editor ladder

Dimensión máxima

La máxima dimensión que puede alcanzar un esquema ladder es de 999 ramales. Por lo tanto, el número máximo de recorridos dependerá de cuantos paralelos haya dibujado, o de cuantos y cuales módulos de sistema (PS3), o de cuantos temporizadores y contadores (S5 100U) haya insertado pero, en cualquiera de los casos no podrá superar 999.

Ramales paralelos

Un ramal paralelo no puede trazarse si se verifica una de las siguientes condiciones:

- sobre el recorrido existe un módulo de sistema (PS3) o un temporizador o un contador (S5 100U);
- el ramal base posee ya cuatro ramales en paralelo y se intenta dibujar el quinto;

- se está intentando trazar un paralelo sobre un ramal, que ya es un paralelo, bypassando una conexión hacia el ramal superior;
- el ramal ya tiene un paralelo en la posición inicial de trazado del nuevo paralelo;
- se ha alcanzado la dimensión máxima del programa ladder.

Un ramal paralelo puede eliminarse si no se verifica ninguna de las siguientes condiciones:

- el ramal posee a su vez otro paralelo;
- el ramal paralelo a eliminar aun tiene contactos.

Paralelo de bobina

En una bobina no puede crearse un paralelo si se verifica una de las siguientes condiciones:

- la bobina base posee ya cuatro bobinas en paralelo y se intenta insertar la quinta;
- la bobina está conectada a la salida de un módulo de sistema (PS3);
- se ha alcanzado la dimensión máxima del programa ladder.

Debug en las ventanas Ladder

La funcionalidad de debug (depuración) de las ventanas ladder permite controlar el funcionamiento del programa que usted ha redactado y resulta sumamente útil cuando éste no se comporta tal como había previsto.

La activación del debug tan sólo es posible cuando el PLC se encuentra en estado de On o de Run, es decir, cuando el conmutador On-Off de la CPU en la *Barra instrumentos* de PC-Sim se encuentra en la posición de On.

En estas condiciones, haga clic sobre el *Pulsador Debug* en la *Barra de pulsadores* de la *Ventana Lista bloques (S5 100U)* o seleccione el *Comando Debug (FPC 404 y PS3)* del *Menú Ventana*.

La sustitución, sobre la barra del título del editor, del símbolo del lápiz por el de una lupa, le indicará la condición de debug. El comentario en la barra del título se sustituye por una indicación que informa si el bloque o el programa está actualmente elaborado, es decir, si la CPU del PLC simulado está ejecutando su código. De hecho, podría no ejecutarse si la CPU está en Stop o, sólo para S5 100U, si no existe ninguna llamada al bloque o bien si las condiciones para la llamada no se han satisfecho en otro bloque ejecutado.

Cuando la función de debug está activa, el editor ladder muestra el estado de los contactos y de los recorridos cerrados por medio de cambios en el color de los elementos del esquema.

Un contacto normalmente abierto se muestra en:

- negro, si la correspondiente variable está en cero. En el caso de una entrada, por ejemplo, si el correspondiente contacto real está abierto;
- rojo, si la correspondiente variable está en uno. En el caso de una entrada, por ejemplo, si el correspondiente contacto real está cerrado.

Un contacto normalmente cerrado se visualiza en:

- negro, si la correspondiente variable está en uno. En el caso de una entrada, por ejemplo, si el correspondiente contacto real está cerrado.;
- rojo, si la correspondiente variable está en cero; En el caso de una entrada, por ejemplo, si el correspondiente contacto real está abierto;

Así pues, en resumen:

- el color negro identifica el cierre del contacto en el esquema, independientemente del estado normal del propio contacto.
- el color rojo identifica la apertura del contacto en el esquema, independientemente del estado normal del propio contacto.

Preste atención a no confundir el tipo de contacto real (NA o NC) con el contacto en el esquema o a no determinar el tipo del segundo en base al del primero. El PLC no conoce, y no puede conocer, el tipo de contacto real, sino tan sólo su estado (cerrado o abierto), y no importa que esa sea su condición normal o no.

Si un recorrido de corriente está cerrado y por tanto, si la bobina de salida está activa, también esta última se visualiza en color rojo. En estas condiciones, si el recorrido está formado por más ramales en paralelo, podrá observar cual es el 'trayecto' de la corriente en su recorrido.

Las variaciones del color de visualización se efectúan en tiempo real a la vez que van modificándose las variables del PLC. Ello constituye un potente instrumento de verificación y corrección de sus aplicaciones.

La activación de la función de debug implica al procesador de su ordenador en una tarea posterior, con la consiguiente disminución de la velocidad del sistema, sobre todo con CPU no muy potentes. Recuerde desactivar esta función cuando ya no sea necesaria, per así permitir que PC-Sim 'funcione' a velocidades normales. Por otro lado, en fase de debug, casi siempre es necesario una disminución de las velocidades, precisamente para poder observar mejor el comportamiento de la instalación y captar los posibles defectos en el programa que ha realizado. Así pues, en cualquier caso, le resultará conveniente disminuir aún más la velocidad de las instalaciones o, cuando ello sea posible, proceder paso a paso, tal como se explica en el [Capítulo 4](#) dedicado a PL-Sim.

El editor AWL para el PLC PS3

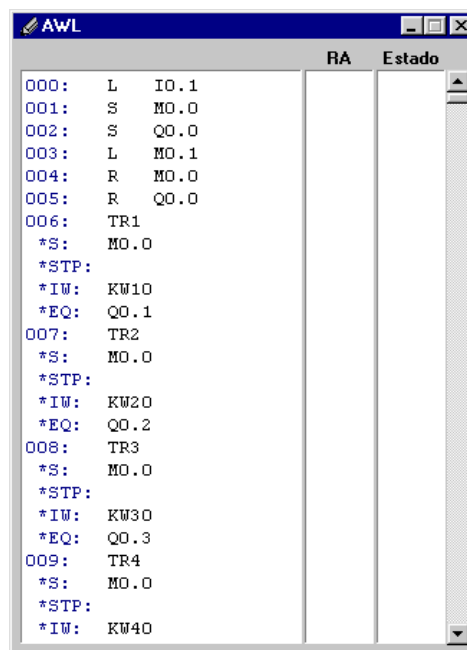
Premisa

Esta descripción no es y no puede ser una lección, y mucho menos un curso, acerca de la utilización del lenguaje AWL para la programación de los PLC. Utilice textos especializados o el manual del programador de su PLC real para aprender los fundamentos de este lenguaje. De todos modos, PC-Sim le permitirá un conocimiento más rápido dada la posibilidad que ofrece de experimentar inmediata y fácilmente con el programa realizado.

También le serán muy útiles los últimos capítulos del presente manual. En ellos aparecen los PLC simulados con una especial atención a sus peculiaridades de programación y se facilita además una interesante serie de ejercicios de programación resueltos y comentados. Recuerde que Autoware está siempre a su disposición para más aclaraciones. Las distintas maneras de conectar con nosotros se describen en el apartado dedicado al [Servicio de Soporte Técnico para Usuarios de AW-SYS](#).

La programación en AWL

El lenguaje AWL o STL o Lista de instrucciones es una de las modalidades más difundidas de programación de los PLC. Redactar un programa AWL significa escribir una serie de instrucciones, cada una de las cuales representa una tarea elemental, que la CPU del PLC ejecutará de manera secuencial. Cada instrucción ocupa una sola fila del programa y cada fila contiene una sola instrucción.



	RA	Estado
000:	L IO.1	
001:	S MO.0	
002:	S QO.0	
003:	L MO.1	
004:	R MO.0	
005:	R QO.0	
006:	TR1	
*S:	MO.0	
*STP:		
*IW:	KW10	
*EQ:	QO.1	
007:	TR2	
*S:	MO.0	
*STP:		
*IW:	KW20	
*EQ:	QO.2	
008:	TR3	
*S:	MO.0	
*STP:		
*IW:	KW30	
*EQ:	QO.3	
009:	TR4	
*S:	MO.0	
*STP:		
*IW:	KW40	

Una instrucción está compuesta por una operación y, generalmente, por un operando. La operación especifica la acción que la CPU debe realizar y el operando especifica el objeto sobre el cual debe realizarse dicha acción.

El editor AWL de PC-Sim para PS3 permite una cómoda y rápida introducción de las instrucciones y de las llamadas a módulo de sistema. La sintaxis de cada una de ellas se controla de inmediato y, en caso de error, el programador recibe un aviso. El mismo editor permitirá, durante la ejecución del programa realizado, la visualización del debug. En correspondencia con cada fila, se visualizarán informaciones sobre los registros principales del PLC y sobre el estado de los operandos. Estas informaciones se actualizarán en tiempo real y como consecuencia de los resultados de la elaboración y de las variaciones de las señales reales.

En el [Capítulo 12](#) se incluye una lista detallada de las operaciones que este PLC permite y de los operandos que cada una de ellas admite.

Apertura del editor AWL

Para utilizar el lenguaje de programación AWL con este PLC es preciso seleccionar el [Comando Nuevo proyecto AWL \(PS3\)](#) del [Menú Archivo](#). Ello creará un nuevo proyecto y abrirá a la vez un editor AWL vacío.

Por el contrario, para llamar a la ventana del editor, es decir, para reabrirla tras haberla cerrado, o devolverla a sus dimensiones normales tras una minimización, o simplemente para llevarla a primer plano, seleccione el [Comando Código \(FPC 404 y PS3\)](#) del [Menú Ventana](#).

Cierre del editor AWL

El cierre del editor AWL puede efectuarse accionando el pulsador de cierre situado a la derecha de la barra del título.

Si se han hecho modificaciones en el código, PC-Sim le pide, con la correspondiente ventana de diálogo, si desea guardarlas. En caso de que responda afirmativamente, recuerde que el código del bloque, al ser modificado, no se guarda en el archivo proyecto sino en la memoria de programa. Esta última, que incluye el código de todos los bloques de su proyecto AWL se registrará en el archivo únicamente si se especifica expresamente que el proyecto se guarde. Respecto a ello, ver el [Comando Guardar proyecto](#) del [Menú Archivo](#). Si responde negativamente a la petición de guardar, las últimas modificaciones efectuadas en el bloque no se guardarán en la memoria de programa, es decir, se perderán.

Un método equivalente para cerrar el bloque consiste en pulsar la combinación de teclas Alt+F4.

Edición del programa AWL

Introducción de una instrucción

La escritura de instrucciones en el interior del editor AWL se produce de manera secuencial, cada instrucción ocupa una sola fila y no está permitido dejar líneas vacías. PC-Sim se encarga de reordenar automáticamente una instrucción escrita, dejando una o más líneas vacías delante de ella.

Cada fila del editor está identificada por un número correlativo de tres cifras: así pues, el máximo número de líneas que el editor puede contener es 1000, de 000 a 999.

Para introducir una instrucción, sitúe el cursor sobre la primera fila vacía del editor y teclee el código, sin preocuparse de dejar espacios de separación, y finalice pulsando Intro. PC-Sim se ocupa de formatear la

instrucción en cuanto se ha escrito, separando y alineando operaciones y operandos para una lectura más cómoda. A continuación, el cursor se sitúa sobre la fila siguiente, para permitirle así escribir de inmediato la siguiente instrucción.

El operando insertado se convierte automáticamente, si ello fuera necesario, en la modalidad de visualización en uso: es decir, por ejemplo, si está activa la visualización símbolos e inserta un operando absoluto, éste será sustituido por el símbolo correspondiente, si existe uno. Lo contrario sucede si está activa la visualización de los operandos absolutos y escribe un símbolo declarado en la [Tabla de símbolos](#). Para más información ver el [Comando Mostrar símbolos](#) y el [Comando Mostrar operandos absolutos](#) del [Menú Modificar](#).

Introducción de un módulo de sistema

En el editor AWL puede introducirse los siguientes módulos de sistema:

- temporizadores (de TR0 a TR31)
- contadores (de C0 a C31)
- registros de desplazamiento (de SR0 a SR31)
- comparadores (de CP0 a CP31).

Para introducir un módulo de sistema teclee la sigla en la primera fila vacía y pulse Intro. El editor AWL inserta automáticamente al final de esta instrucción, las líneas necesarias para la asignación de las entradas y de las salidas del módulo para la conexión con el programa del usuario. Para finalizar, escriba, cuando sea necesario, las siglas de los operandos a conectar al bloque.

Eliminación de una instrucción

Para eliminar una instrucción insertada con anterioridad actúe de una de las siguientes maneras:

- Sitúe el cursor sobre la instrucción a eliminar y seleccione el [Comando Eliminar](#) en el [Menú Modificar](#)
- Sitúe el cursor sobre la instrucción a eliminar y pulse Ctrl+Y
- Elimine todos los caracteres de la fila y pulse Intro o una de las teclas de dirección

Eliminación de un módulo de sistema

Debe procederse tal como se ha descrito para la eliminación de instrucciones, eliminando la llamada al módulo, es decir, la fila que incluye la sigla del mismo. La eliminación de la llamada eliminará todo el módulo, incluidas las líneas para la definición de las entradas/salidas.

Inserción de una fila vacía

Sitúe el cursor sobre la fila del editor antes de la cual desea insertar una fila vacía y seleccione el [Comando Insertar](#) del [Menú Modificar](#) o bien teclee Ctrl+I.

Operaciones soportadas

El editor AWL reconoce todas las operaciones que soporta el PLC real. El listado y la descripción detallada de estas se facilita en el [Capítulo 12](#) en el apartado [Descripción de las operaciones](#).

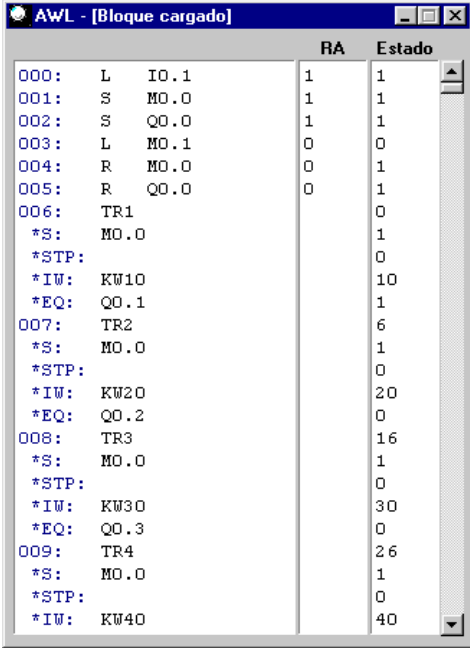
Detección de errores sintácticos

Las operaciones de inserción del código AWL están completa y permanentemente controladas por el editor. Cada vez que inserte una instrucción, el editor controla que tanto la operación como el operando sean correctos y, en caso de control positivo, formatea la introducción alineándola para una lectura más cómoda. Por el contrario, si la instrucción que ha insertado resulta errónea, el editor no le permite cambiar de fila y le indica el error con una señal acústica.

Debug en la ventana AWL

Para activar la función de debug en una ventana AWL seleccione el [Comando Debug \(FPC 404 y PS3\)](#) en el [Menú Ventana](#). La función sólo está activa si el PLC se encuentra en estado de ON o en el de RUN.

Las dos columnas de la derecha de la ventana del editor AWL le permiten efectuar un debug eficaz durante la ejecución de su código.



			RA	Estado
000:	L	IO.1	1	1
001:	S	MO.0	1	1
002:	S	QO.0	1	1
003:	L	MO.1	0	0
004:	R	MO.0	0	1
005:	R	QO.0	0	1
006:	TR1			0
*S:	MO.0			1
*STP:				0
*IW:	KW10			10
*EQ:	QO.1			1
007:	TR2			6
*S:	MO.0			1
*STP:				0
*IW:	KW20			20
*EQ:	QO.2			0
008:	TR3			16
*S:	MO.0			1
*STP:				0
*IW:	KW30			30
*EQ:	QO.3			0
009:	TR4			26
*S:	MO.0			1
*STP:				0
*IW:	KW40			40

La columna denominada RA le facilita, fila por fila, allí donde eso tenga un sentido, el contenido del registro general en formato bit, byte o word según el tipo de instrucción presente en cada fila.

La otra columna le permite valorar el estado del operando sobre la fila correspondiente y representa:

- para las señales digitales (entradas, salidas, merker), el estado actual (0 o 1);
- para los operandos a byte, el valor actual (de 0 a 255);
- para los operandos a word, el valor actual (de 0 a 65535)

- para los temporizadores, el valor actual del registro del temporizador expresado en décimas de segundo, como un número de 0 a 65535
- para los contadores: el valor actual de conteo expresado como un número de 0 a 65535
- para los registros de desplazamiento: el valor actual del registro de 16 bit expresado como número de 0 a 65535.

El editor AWL para el PLC S5 100U

Premisa

Esta descripción no es y no puede ser una lección, y mucho menos un curso, acerca de la utilización del lenguaje AWL para la programación de los PLC. Utilice textos especializados o el manual del programador de su PLC real para aprender los fundamentos de este lenguaje. Sin embargo, PC-Sim le permitirá un conocimiento más rápido dada la posibilidad que ofrece de experimentar inmediata y fácilmente con el programa realizado.

También le serán muy útiles los últimos capítulos del presente manual. En ellos aparecen los PLC simulados con una especial atención a sus peculiaridades de programación y se facilita además una interesante serie de ejercicios de programación resueltos y comentados. Recuerde que Autoware está siempre a su disposición para más aclaraciones. Las distintas maneras de conectar con nosotros se describen en el apartado dedicado al [Servicio de Soporte Técnico para Usuarios de AW-SYS](#).

La programación en AWL

El lenguaje AWL o STL o Lista de instrucciones es una de las modalidades más difundidas de programación de los PLC. Escribir un programa AWL significa teclear una serie de instrucciones, cada una de las cuales representa una tarea elemental, que la CPU del PLC ejecutará de manera secuencial. Cada instrucción ocupa una sola fila del programa y cada fila contiene una sola instrucción.

Cada una de ellas está compuesta por una operación y, generalmente, por un operando. La operación especifica la acción que la CPU debe realizar y el operando especifica el objeto sobre el cual debe ejecutarse tal acción.

El editor AWL de PC-Sim para S5 100U permite una cómoda y rápida introducción de las instrucciones, también gracias a la posibilidad de utilizar los comandos de cortar, copiar y pegar. La sintaxis de cada una de ellas se controla de inmediato y el programador recibe un aviso en caso de error. El programa, además, puede ir acompañado con textos como comentario.

El mismo editor le permitirá, durante la ejecución del programa realizado, la visualización del debug. En correspondencia con cada fila, se visualizarán informaciones sobre los registros principales del PLC y sobre el estado de los operandos. Estas informaciones se actualizarán en tiempo real y como consecuencia de los resultados de la elaboración y de las variaciones de las señales reales.

En el [Capítulo 14](#) se incluye una lista detallada de las operaciones que este PLC permite y de los operandos admisibles por cada una de ellas.

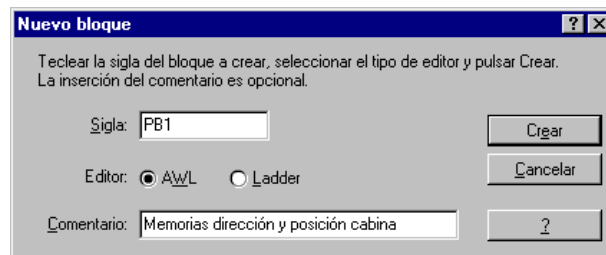
Apertura del editor AWL

Para abrir el editor de un bloque AWL existente, seleccione el bloque en el interior de la [Tabla de bloques](#) y pulse el [Pulsador Código](#), o bien haga doble clic sobre la fila de la tabla que contiene el bloque a editar.

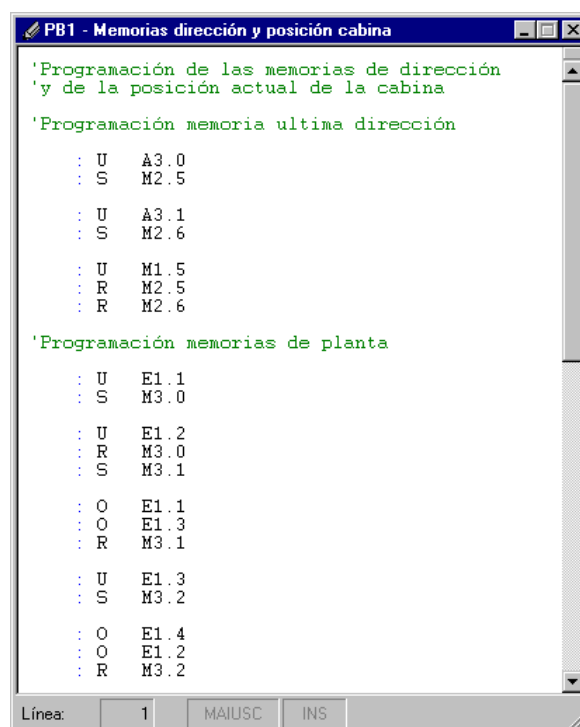
Para crear un nuevo bloque AWL, abriendo también el editor, utilice uno de los siguientes procedimientos:

- seleccione el *Comando Nuevo bloque (S5 100U)* en el *Menú Archivo*;
- haga clic sobre el *Pulsador Nuevo bloque* de la *Barra instrumentos* de PC-Sim;
- haga clic sobre el *Pulsador Nuevo* en la *Ventana Lista bloques (S5 100U)*.

Los tres procedimientos son equivalentes y en todos ellos, PC-Sim prosigue mostrando la ventana de diálogo *Nuevo bloque*. Escriba entonces la sigla del bloque a crear, seleccione como editor de bloque a adoptar el tipo AWL, si lo desea inserte un texto de comentario al bloque y, por último, accione el pulsador *Nuevo*.



Si la sigla escrita es válida y el bloque no está ya presente en el proyecto, éste se añade a la *Tabla de bloques* y el editor AWL se abre.



Cierre del editor AWL

El cierre del editor AWL puede efectuarse accionando el pulsador de cierre situado a la derecha de la barra del título. Si se han hecho modificaciones en el texto del bloque, PC-Sim le pide, con la correspondiente ventana de diálogo, si desea guardarlas. En caso de responder afirmativamente, recuerde que el texto del bloque, si se ha modificado, no se guarda en el archivo proyecto sino en la memoria de programa. Esta última, que incluye el código de todos los bloques de su proyecto, ya sea ladder, AWL o de datos, se registrará en el archivo únicamente si se especifica expresamente que el proyecto se guarde. Respecto a ello, ver el [Comando Guardar proyecto](#) del [Menú Archivo](#).

Si responde negativamente a la petición de guardar, las últimas modificaciones efectuadas en el bloque no se guardarán en la memoria de programa, es decir, se perderán. Si el bloque no se había guardado en ninguna ocasión desde su creación, la operación de cierre y la elección de no guardar los cambios, comportarán la pérdida total del bloque, que será también eliminado de la [Tabla de bloques](#).

Un método equivalente para cerrar el bloque consiste en pulsar la combinación de teclas Alt+F4.

Guardar el bloque

Además del cierre del editor, tal como acabamos de ver, en cualquier momento puede guardarse el contenido de un bloque seleccionando el [Comando Guardar bloque \(S5 100U\)](#) en el [Menú Archivo](#). También en este caso son válidas las consideraciones hechas en el apartado precedente en lo referente a guardar, y para más información, le remitimos a la descripción del [Comando Guardar proyecto](#) del [Menú Archivo](#).

Edición del programa AWL

Composición de una instrucción

Cada instrucción en código AWL está compuesta por una operación y, generalmente, por un operando sobre el cual actúa la operación. Sin embargo, existen algunas operaciones que no necesitan operando o en las cuales el operando a utilizar está implícito y por lo tanto no es necesario mencionarlo.

Para informaciones sobre las operaciones implementadas en esta simulación y sobre los operandos que soportan, consulte el apartado [Programación en AWL](#) en el [Capítulo 14](#).

Introducción de una instrucción

Escribir en código AWL con PC-Sim es una cosa sencillísima y, por lo tanto, no requiere excesivos comentarios.

Sitúe el puntero de inserción de texto en el editor AWL, haciendo clic sobre el mismo, y empiece a escribir su programa sin preocuparse de dejar espacios, de usar tabulaciones o de escribir en mayúsculas o minúsculas. ¡El editor de PC-Sim se ocupa de todo! Una única advertencia: recuerde regresar al principio de cada fila pulsando Intro después de cada instrucción insertada.

Cada vez que complete la inserción de una instrucción, el editor se ocupa de formatear correctamente el texto insertado, regulando los espacios, colocando en columnas las operaciones y los operandos y convirtiendo el texto en mayúsculas.

Introducción de etiquetas

Cada fila de un programa AWL en un bloque funcional (FB) puede identificarse por medio de una etiqueta. La etiqueta es un “cinta” alfanumérica de cuatro caracteres como máximo, de los cuales el primero es necesariamente una letra, que se sitúa al inicio de la fila a la izquierda del carácter ‘:’. Las etiquetas identifican las posiciones de llegada de una instrucción de salto.

Cuando desee insertar una etiqueta sobre la fila en uso, teclee la sigla y a continuación el carácter ‘:’ y, si lo desea, haga que la instrucción continúe, y termine pulsando Intro. Una vez más, el editor AWL de PC-Sim se ocupará de dar formato a la fila apenas la haya insertado, tal como ya hemos visto anteriormente, mostrando la etiqueta en color azul.

Introducción de textos de comentario

En cada fila del editor, contenga o no una instrucción y/o una etiqueta, es posible insertar un texto de comentario. Le bastará escribir un carácter de apóstrofe (‘) y añadirle a continuación el comentario. Para finalizar la operación pulse la tecla Intro. En cuanto cambia de fila en uso, PC-Sim se encarga de dar formato, además de a las posibles instrucciones y etiquetas presentes sobre la fila, también al comentario, programando su posición de inicio y el color, que para el comentario es el verde.

La posición inicial del comentario es el margen izquierdo, si en la fila no aparece ninguna otra instrucción, o una posición adecuada para dejar sitio a la instrucción más larga (L KM ...). Excepción a esta regla es el caso de instrucción de salto, condicionado o no condicionado, a bloque funcional, cuya longitud puede variar con el número de parámetros presentes. En este caso la posición inicial del comentario, si no es compatible con cuanto se ha dicho anteriormente, se fijará al final de la instrucción.

Operaciones de edición avanzadas

Utilice los comandos *Cortar*, *Copiar* y *Pegar* del *Menú Modificar* o las correspondientes *Accesos rápidos con el teclado* Ctrl+X, Ctrl+C y Ctrl+V para cortar, copiar y pegar código en el mismo bloque o de un bloque a otro del mismo proyecto o de proyectos distintos.

Detección de errores sintácticos

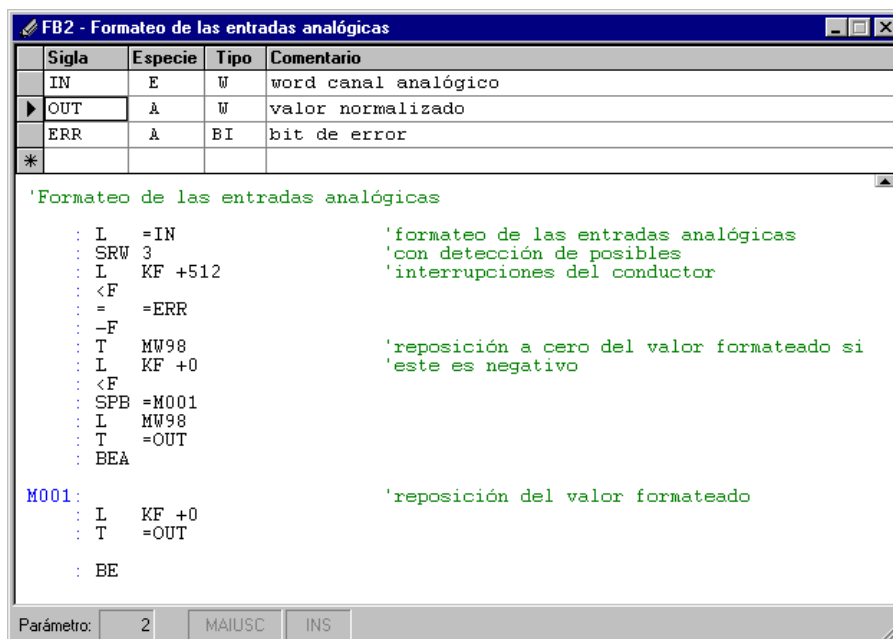
Si inserta una instrucción cometiendo un error sintáctico, en el momento de la validación de aquella (paso a otra fila) la instrucción errónea se manifiesta en color rojo. Podrá seguir realizando todas las acciones que desee en este u otro editor pero la instrucción errónea continuará apareciendo en rojo hasta que sea corregida.

Parametrización de bloques funcionales

Los bloques funcionales de este PLC, y sólo estos, pueden parametrizarse, es decir, en ellos pueden definirse variables ficticias, llamadas operandos formales, a utilizar en la escritura del programa. Cuando se llama al bloque con una instrucción SPA FB o SPB FB, será preciso especificar en la llamada una lista de variables (operandos absolutos o símbolos) correspondientes por número y por tipo a los parámetros que el bloque acepta. Estas variables, en la elaboración del bloque funcional parametrizado, se utilizarán en lugar de los operandos formales. Esto es lo que sucede, para ciertas acciones, también en lenguajes de alto nivel para PC cuando se llama a un subprograma para el cual se han definidos parámetros.

Por tanto, es indispensable declarar el número y tipo de parámetros. Con esta finalidad el editor AWL de PC-Sim ofrece la posibilidad de abrir una tabla para su definición. Haciendo clic sobre el pulsador marcado con un triángulo invertido, situado justo encima de la barra de desplazamiento, aparece la tabla de inserción de parámetros, que inicialmente está vacía.

Para insertar un parámetro es preciso especificar la sigla y la especie. Para algunas especies es necesario especificar a continuación el tipo. El del comentario es un campo opcional.



La tabla está constituida por cuatro columnas.

Columna Sigla

En ella debe especificarse la sigla del operando formal que se utilizará en el código del bloque con las operaciones que lo prevean. La sigla puede tener una longitud máxima de cuatro caracteres alfanuméricos y debe necesariamente empezar con una letra. La inserción en este campo es obligatoria.

Columna Especie

En esta columna se declara la especie del operando formal, cuya sigla está en la columna anterior. Las distintas especies posibles son:

E parámetro de entrada
A parámetro de salida
D dato constante
B parámetro bloque
T temporizador
Z contador

Columna Tipo

Para algunas de las especies es necesario facilitar una declaración posterior del tipo, que informa al compilador del programa del usuario sobre el número de bit del parámetro, cuando esta información no se deduzca unívocamente de la declaración de la especie.

Tipos válidos para las especies A y E son:

B parámetro de bit
BY parámetro de byte
W parámetro de word

Tipos válidos para la especie D son:

KF constante decimal con señal
KH constante hexadecimal
KM constante binaria
KY constante de doble byte
KT constante de temporización
KZ constante de conteo

Las demás especies no necesitan la declaración del tipo y por tanto, para éstas, la celda correspondiente en esa columna no debe cumplimentarse.

Columna Comentario

Esta columna permite asociar a cada parámetro un texto de comentario para aclarar su utilización o su función específica. La inserción de datos en esta columna es voluntaria.

Debug en las ventanas AWL

La función de debug de los bloques AWL permite controlar el funcionamiento del programa que ha redactado y resulta sumamente útil cuando éste no se comporta tal como estaba previsto.

La activación del debug tan sólo es posible cuando el PLC se encuentra en estado de ON o en el de RUN, es decir, cuando el conmutador On-Off de la CPU en la [Barra instrumentos](#) de PC-Sim se encuentra en la posición de On. En esta situación, seleccione, si fuera necesario, en la [Tabla de bloques](#) el bloque a analizar y haga clic sobre el [Pulsador Debug](#).

La sustitución, sobre la barra del título del editor, del símbolo del lápiz por el de una lupa le señalará la condición de bloque en debug. El comentario al bloque en la barra del título será sustituido por una

indicación que informa si el bloque está actualmente elaborado, es decir, si la CPU del PLC simulado está ejecutando el código del bloque. De hecho, el código podría no ejecutarse si la CPU está en Stop o si no existe ninguna llamada al bloque o bien si las condiciones para la llamada no se han satisfecho en otro bloque ejecutado.

Instrucción	RLC	Estado/Accu	Comentario
: L MW100	1	0000 0014	'alarma de máximo depósito mezclador
: L KF +2000	1	07D0 0000	
: >F	0		
: = M5.0	0	0	'memoria para activación alarma
	0		
: L MW100	0	0000 07D0	'alarma de mínimo depósito mezclador
: L KF +15	0	000F 0000	
: <F	1		
: = M5.1	1	1	'memoria para activación alarma
	1		
: L MW102	1	0000 000F	'alarma de máximo depósito producto
: L KF +2000	1	07D0 0000	
: >F	0		
: = M5.2	0	0	'memoria para activación alarma
	0		
: L MW102	0	0000 07D0	'alarma de mínimo depósito producto
: L KF +15	0	000F 0000	
: <F	1		
: = M5.3	1	1	'memoria para activación alarma
	1		
: L MW104	1	0000 000F	'alarma máxima temperatura
: L KF +2000	1	07D0 0000	
: >F	0		
: = M5.4	0	0	'memoria para activación alarma
	0		

La visualización normal del bloque se sustituirá por una tabla formada por tantas filas como instrucciones haya y por cuatro columnas.

Columna Instrucción

Incluye las instrucciones que constituyen el código del bloque AWL y a las cuales hacen referencia las celdas de las columnas siguientes.

Columna RLC

Muestra el resultado lógico combinatorio después de la elaboración de la instrucción sobre la misma fila. Para posteriores aclaraciones, ver el apartado [Registros](#) en el [Capítulo 14](#).

Columna Estado/Accu

Esta columna incluye:

- para operandos de bit, el estado del operando (0 o 1)
- para operaciones de carga o transferencia (L o T) y, en general, para operaciones que involucran operandos de más bits, el valor de los acumuladores ACCU1 y ACCU2, ver el apartado [Registros](#) en el [Capítulo 14](#)
- para operaciones de set de temporizadores (SI, SV, SE, SS, SA), el valor actual del temporizador, expresado en la base de tiempos programada para ello.

Para más informaciones sobre los acumuladores ver el apartado [Registros](#) en el [Capítulo 14](#).

Columna Comentario

La última columna incluye el comentario eventualmente presente en la correspondiente fila.

Solo las columnas *RLC* y *Estado/Accu* son dinámicas, es decir, su contenido varia en relación con la elaboración del programa.

Si lleva el PLC a STOP, la visualización activa del debug queda congelada en los últimos valores elaborados. El retorno del PLC al estado de OFF devuelve automáticamente la ventana al estado de Edit.

El editor de datos (S5 100U)

La ventana de editor de datos tan sólo está disponible en la simulación del PLC S5 100U siendo éste el único entre los simulados que permite ese tipo de programación. A diferencia de los demás editores, éste no permite redactar un programa sino que da la posibilidad al programador de definir áreas de datos.

El editor muestra una tabla en la que cada fila contiene un dato en formato de 16 bit (data word). El número de estos podrá definirse libremente, si bien debe mantenerse por debajo de un valor máximo. El usuario también podrá seleccionar su formato, es decir, la base de representación, entre las varias previstas. Asimismo, están disponibles herramientas para la inserción y la eliminación de filas sencillas o múltiples.

Los datos así programados podrán dirigirse, en lectura o en escritura, hacia los bloques de programa del proyecto.

Dispone de más informaciones sobre los bloques de datos y sobre las operaciones de acceso a estos en el [Capítulo 14](#).

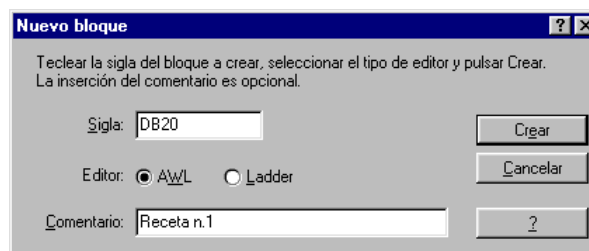
Apertura del editor de datos

Para abrir el editor de un bloque de datos existente, seleccione el bloque en el interior de la [Tabla de bloques](#) y pulse el [Pulsador Código](#), o bien haga doble clic sobre la fila de la tabla que contiene el bloque a editar.

Para crear un nuevo bloque de datos, abriendo también el editor, utilice uno de los siguientes procedimientos:

- seleccione el [Comando Nuevo bloque \(S5 100U\)](#) en el [Menú Archivo](#);
- haga clic sobre el [Pulsador Nuevo bloque](#) de la [Barra instrumentos](#) de PC-Sim;
- haga clic sobre el [Pulsador Nuevo](#) en la [Ventana Lista bloques \(S5 100U\)](#).

Los tres procedimientos son equivalentes y en todos ellos, PC-Sim prosigue mostrando la ventana de dialogo [Nuevo bloque](#). Escriba la sigla del bloque a crear, si lo desea inserte un texto de comentario al bloque y, por último, accione el [Pulsador Nuevo](#).



Si la sigla tecleada es válida y el bloque no está ya presente en el proyecto, éste se incluye en la [Tabla de bloques](#) y el editor de datos se abre.

Cierre del editor de datos

El cierre del editor de datos puede efectuarse accionando el pulsador de cierre situado a la derecha en la barra del título. Si se han efectuado modificaciones en el esquema, PC-Sim le solicitará, por medio de la

correspondiente ventana de diálogo, si desea guardarlas. En caso de que responda afirmativamente, recuerde que los datos del bloque, al ser modificado, no se guardan en el archivo proyecto sino en la memoria de programa. Esta última, que incluye el código de todos los bloques de su proyecto, ya sea ladder, AWL o de datos, se registrará en el archivo únicamente si se especifica expresamente que el proyecto se guarde. Respecto a ello, ver el [Comando Guardar proyecto](#) del [Menú Archivo](#).

Si responde negativamente a la petición de guardar, las últimas modificaciones efectuadas en el bloque no se guardarán en la memoria del programa, es decir, se perderán. Si el bloque no se había guardado nunca desde su creación, la operación de cierre y la opción de no guardar las modificaciones comportarán la pérdida total del bloque, que también será eliminado de la [Tabla de bloques](#).

Otro método equivalente para cerrar el bloque consiste en pulsar la combinación de teclas Alt+F4

Guardar el bloque

Además del cierre del editor, como acabamos de ver, puede guardarse el contenido de un bloque de datos en cualquier momento, seleccionando el [Comando Guardar bloque \(S5 100U\)](#) en el [Menú Archivo](#). También en este caso son válidas las consideraciones hechas en el apartado anterior respecto a como guardar. Para más informaciones, también en este caso le remitimos a la descripción del [Comando Guardar proyecto](#) del [Menú Archivo](#).

Composición de la ventana

Barra del título

La barra del título incluye la sigla y el comentario al bloque, tal como se escribieron en el momento de su creación, o si han sido posteriormente modificados.

Tabla de datos

La *Tabla de datos* contiene, subdivididos en filas, los datos del bloque y permite la introducción de nuevos y la modificación de los ya existentes. Sus dimensiones se adaptan automáticamente a las de la ventana. Cuando estas no permitan la visualización completa de los datos, PC-Sim muestra una barra de desplazamiento vertical a la derecha de la tabla.

La última fila de la tabla está siempre vacía y permite la introducción de un nuevo dato.

Columna DW

En esta columna, el usuario no puede modificar el campo, tal como recuerda el fondo gris. PC-Sim inserta en él, el número progresivo de la data word, partiendo de 0 hasta el máximo permitido de 255.

Cuando inserte o elimine filas de la tabla, los números en esta columna se reordenarán automáticamente.

DW	Tipo	Valor	Comentario
0	KM	00000000 00000000	Datos del ciclo
1	KM	00000000 00000001	
2	KM	00000000 00000010	
3	KM	00000000 00000100	
4	KM	00000000 00001000	
5	KM	00000000 00010000	
6	KM	00000000 00100000	
7	KM	00000000 01000000	
8	KM	00000000 10000000	
9	KM	00000000 10000001	
10	KM	00000000 10000010	
11	KM	00000000 10000100	
12	KM	00000000 10001000	
13	KM	00000000 10010000	
14	KM	00000000 10100000	

Columna Tipo

Especifica el tipo de representación para el dato contenido en la columna siguiente. Los tipos reconocidos, todos de 16 bit, son:

- KH número hexadecimal
- KF número decimal con signo
- KT valor de tiempo
- KZ valor de conteo
- KY número de doble byte
- KM número binario

Cada celda de esta columna es, en realidad, una casilla combinada descendiente. Es decir, usted puede introducir directamente el texto, tecleando una de las siglas válidas, o bien hacer clic sobre la flecha situada a la derecha para visualizar el listado de opciones y seleccionar una de ellas.

Columna Valor




Incluye el valor del dato en la representación establecida en la *Columna Tipo*. El formateo del valor se efectúa automáticamente. Una modificación del tipo de representación, especificado en la columna anterior, comporta una adaptación automática del valor visualizado. Si en la nueva base tiene sentido la representación del dato en vigor, esta se visualizará y en caso contrario se mostrará un valor nulo.

Columna Comentario

En esta columna puede insertarse, a voluntad del usuario, un comentario al dato en la misma fila, que sirva como aclaración o que recuerde su función.

Selectores de fila

La parte de la tabla a la izquierda de la primera columna está formada por los selectores de fila. Estos le permiten seleccionar las líneas de la tabla y, a la vez, facilitan informaciones sobre la fila correspondiente mediante la visualización de imágenes:

-  fila de agregación
-  fila en uso
-  fila corriente en edición

Barra pulsadores

La *Barra pulsadores* está en la parte inferior de la ventana de edición. Contiene los pulsadores necesarios para las operaciones de inserción y eliminación de líneas de datos, ya sean sencillas o múltiples. Los procedimientos iniciados accionando los pulsadores actúan sobre las líneas seleccionadas.

Para seleccionar un único dato haga clic sobre el correspondiente selector de fila.

Para seleccionar líneas múltiples contiguas: seleccione la primera fila tal como se ha descrito anteriormente, a continuación presione y mantenga presionada la tecla Mayús y haga clic sobre el selector de la última fila. Todas las líneas, de la primera a la última quedarán seleccionadas y todas serán objeto de las acciones de los pulsadores *Eliminar e Insertar*.

Por último, puede también efectuar selecciones múltiples de líneas no contiguas: seleccione la primera; a continuación, presione y mantenga presionada la tecla Control y haga clic sobre los selectores de las líneas que desea agregar a la selección, de uno en uno. Un clic sobre el selector de una fila no seleccionada añadirá la fila a la selección; mientras que un clic sobre el selector de una fila ya seleccionada eliminará la fila de la selección.

Pulsador Eliminar

El pulsador está activado únicamente si existe una selección en la tabla. Haciendo clic sobre él se obtiene la eliminación de las líneas seleccionadas, la reordenación de la tabla y la reorganización de la numeración en la columna DW.

Pulsador Insertar

El pulsador está activado únicamente se existe una selección en la tabla. Haciendo clic sobre él se obtiene la inserción de una o más filas vacías en correspondencia con la primera seleccionada, el desplazamiento hacia debajo de las filas seleccionadas y de todos los siguientes y la reorganización de la numeración de la columna DW. Se puede insertar más de una fila cada vez, tan sólo si la selección está formada por filas contiguas.

Edición de datos

Incorporación de un dato

La incorporación de un dato se efectúa sobre la fila de agregación, es decir, la última de la tabla que tiene campos completamente vacíos y que está marcada con un asterisco situado sobre el selector de fila. Cuando se abra un nuevo bloque esta será la única fila presente.

Para agregar un dato a la tabla:

- seleccione, en la fila de agregación, la celda correspondiente a la *Columna Tipo*, haciendo clic sobre ella,
- Teclee la sigla de la representación o bien selecciónela en el listado. Automáticamente se cumplimentará la celda en la columna DW con el número progresivo del dato. La selección pasará a la celda situada inmediatamente a su derecha.
- Teclee el valor del dato de manera congruente con la base especificada y pulse Intro. La celda seleccionada se convierte ahora en aquella en la *Columna Comentario*.
- Si lo desea, escriba un texto que sirva para identificar el dato y pulse Intro. Si no desea insertar ningún comentario, pulse sólo la tecla Intro.

El dato se ha agregado a la tabla, y una nueva fila de agregación se visualiza al final de aquella.

Modificación de los datos

Para modificar un dato es preciso seguir el procedimiento que a continuación se describe.

- Seleccione la celda a modificar haciendo clic sobre ella
- Si desea sobrescribir por completo el nuevo valor, tecléelo directamente en el campo
- Si quiere modificar parte del valor preexistente, pulse la tecla F2, que le permite entrar en la edición de la celda y, a continuación, introduzca las modificaciones deseadas en el correspondiente campo
- Pulse la tecla Intro tantas veces como sea necesario para pasar a las filas siguientes.

Ventana Tabla de símbolos

La *Ventana Tabla de símbolos* permite al programador la definición de las correspondencias entre operandos absolutos y símbolos. Una vez definidas dichas correspondencias podrá utilizar, en la programación, los segundos en el lugar de los primeros, con una notable ventaja respecto a las facilidades de lectura del código.

Las asociaciones se introducen en una tabla específica, y cada incorporación está controlada por PC-Sim en lo que se refiere a su corrección sintáctica y su univocidad. Para cada asociación podrá insertarse un texto descriptivo del operando. Además, están disponibles funciones de inserción y de eliminación de líneas sencillas o múltiples.

Operandos absolutos y símbolos

Los operandos absolutos son aquellos operandos cuya sintaxis está en estrecha relación con la marca y el modelo del PLC, así como con el número del módulo y del canal de la señal a la que se refieren o bien con su posición en la memoria.

PC-Sim le permite utilizar, durante la fase de edición de los programas, además de los absolutos, también los operandos simbólicos. A partir de ahora, en el manual, se hará referencia a éstos denominándolos simplemente símbolos. Usted podrá definir a su gusto la sigla de estos, si bien deberá respetar algunas reglas, y posiblemente se referirá a la función del objeto conectado al PLC o de la variable interna que desea direccionar.

PC-Sim se encargará automáticamente de la sustitución de los símbolos por sus correspondientes operandos absolutos en la memoria del PLC simulado a cada compilación del programa del usuario, de una manera absolutamente transparente para usted: verá y usará cómodos símbolos, mientras el PLC simulado trabajará sobre los correspondientes operandos absolutos.

Los símbolos utilizables en PC-Sim pueden tener una longitud de ocho caracteres como máximo. El primero de estos debe ser necesariamente una letra mayúscula (de la 'A' a la 'Z'), los otros pueden ser tanto letras como números (de '0' a '9'). Cada símbolo utilizado en el programa deberá ir precedido por el carácter '-' (guión o signo menos) que avisa al compilador de que el operando examinado debe considerarse un símbolo y que como tal deberá ser tratado.

Para poder utilizar los símbolos en lugar de los operandos absolutos es necesario especificar su correspondencia en la *Ventana Tabla de símbolos*. Desde ese momento, cada asociación de un operando absoluto con un símbolo estará considerada como una atribución.

Apertura de la ventana

Para abrir la ventana *Tabla de símbolos*, o para llevarla a primer plano, opere de una de las siguientes maneras:

- seleccione el *Comando Tabla de símbolos* en el *Menú Ventana*;
- haga clic sobre el pulsador *Tabla de símbolos* de la *Barra instrumentos* de PC-Sim ;

Puede cambiar el tamaño y la posición de esta ventana a su gusto. Las últimas variaciones de dimensión y de posición, se guardarán con el archivo proyecto de manera que, al volver a abrirlo, PC-Sim se encargará de restablecer las últimas características guardadas de la ventana.

Cierre de la ventana

La ventana *Tabla de símbolos* puede cerrarse haciendo clic sobre el pulsador de cierre situado en el extremo derecho de la barra del título. Hay que dejar claro que el cierre de la ventana no implica la pérdida de sus atribuciones sino tan sólo la desaparición de esta de la pantalla.

Guardar las atribuciones

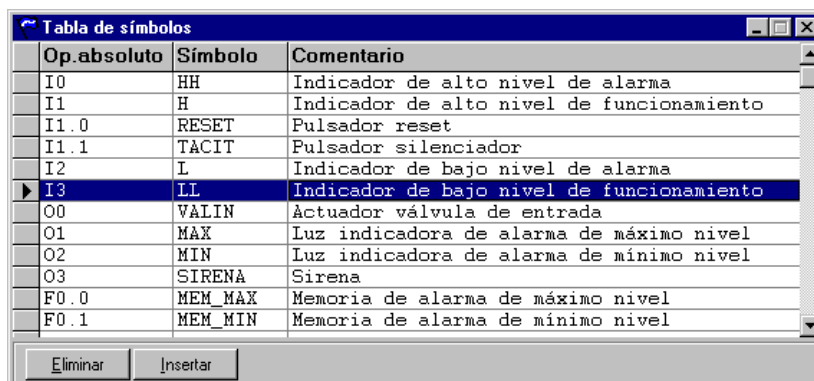
Como ya se ha explicado, las atribuciones de los símbolos se guardan en el archivo proyecto junto con el código del programa y las demás informaciones descritas en el apartado *Composición de un proyecto*.

En cualquier momento pueden guardarse las atribuciones insertadas en la ventana. Si desea guardarlas en el archivo proyecto en uso elija el *Comando Guardar proyecto* del *Menú Archivo*. Por el contrario, si desea guardarlas utilizando un nombre de proyecto distinto, elija, en el mismo menú, el *Comando Guardar proyecto como...*

Composición de la ventana

Tabla de símbolos

La *Tabla de símbolos* muestra la lista de las atribuciones válidas para el proyecto. Está formada por 100 filas, en las que pueden insertarse otras tantas atribuciones.



Op.absoluto	Símbolo	Comentario
I0	HH	Indicador de alto nivel de alarma
I1	H	Indicador de alto nivel de funcionamiento
I1.0	RESET	Pulsador reset
I1.1	TACIT	Pulsador silenciador
I2	L	Indicador de bajo nivel de alarma
I3	LL	Indicador de bajo nivel de funcionamiento
O0	VALIN	Actuador válvula de entrada
O1	MAX	Luz indicadora de alarma de máximo nivel
O2	MIN	Luz indicadora de alarma de mínimo nivel
O3	SIRENA	Sirena
F0.0	MEM_MAX	Memoria de alarma de máximo nivel
F0.1	MEM_MIN	Memoria de alarma de mínimo nivel

Columna Operando absoluto

Contiene el operando absoluto al cual se asociará el símbolo de la columna siguiente. El operando absoluto está compuesto por una o más letras que indican su tipo (ej. E = bit de entrada, MW = palabra de memoria, etc.) y por una o más cifras que especifican su dirección (ej. E1.2 = bit de entrada 2 del módulo 1, MW60 = palabra de memoria 60, etc.).

Para una descripción completa de los operandos absolutos utilizables en la programación, remítase al capítulo que incluye las notas de programación del PLC que esté utilizando.

Columna Símbolo



Contiene el símbolo que se usará en la programación en lugar del operando absoluto de la primera columna. La inserción del símbolo en esta columna se realiza omitiendo el carácter inicial '-', que sí es necesario utilizar en la escritura del programa. Sin embargo, si escribe también el guión inicial, PC-Sim se encargará de eliminarlo automáticamente.

Columna Comentario

Contiene un comentario descriptivo ligado a la función del operando. La introducción en este campo es facultativa.

Selectores de fila

La parte de la tabla situada a la izquierda de la primera columna está formada por los selectores de fila. Estos le permiten seleccionar las filas de la tabla y, simultáneamente, facilitan informaciones sobre la fila correspondiente mediante la visualización de imágenes:

-  fila en uso
-  fila en uso en edición

Barra de pulsadores

La *Barra de pulsadores* de la ventana *Tabla de símbolos* contiene los pulsadores necesarios para las operaciones de inserción y de eliminación de declaraciones sencillas o múltiples.

Para seleccionar una atribución sencilla haga clic sobre el selector de fila, situado a la izquierda de la fila de la tabla que contiene la declaración a seleccionar.

Para seleccionar líneas múltiples contiguas: seleccione la primera fila tal como se ha descrito anteriormente, a continuación presione y mantenga presionada la tecla Mayús y haga clic sobre el selector de la última fila. Todas las líneas, de la primera a la última, quedarán seleccionadas y todas serán objeto de las acciones de los pulsadores *Eliminar* e *Insertar*.

Por último, se puede también efectuar selecciones múltiples de líneas no contiguas: seleccione la primera; a continuación, presione y mantenga presionada la tecla Control y haga clic sobre los selectores de las líneas que desea agregar a la selección, de uno en uno. Un clic sobre el selector de una fila no seleccionada añadirá la fila a la selección; mientras que un clic sobre el selector de una fila ya seleccionada eliminará la fila de la selección.

Pulsador Eliminar

Utilice este pulsador para eliminar de la lista las atribuciones seleccionadas. Seleccione la fila o filas de la tabla que contengan las atribuciones a eliminar y accione el *Pulsador Eliminar*.

El pulsador está activado sólo si se ha hecho una selección en la tabla.

Pulsador Insertar

Utilice este pulsador para insertar una o más filas vacías en la tabla. Seleccione la fila de la tabla anterior a la que desea insertar, a continuación seleccione, si es necesario, otras filas (con tal de que sean contiguas) y accione el *Pulsador Insertar*.

De este modo se crea un número de filas vacío igual al número de líneas seleccionadas y los siguientes se desplazan hacia abajo en la tabla. Preste atención al hecho de que las atribuciones que ocupan las últimas filas de la tabla, si existen, se perderán.

El pulsador sólo está activado si en la tabla está presente una selección.

Edición de los símbolos

Introducir una atribución

Sitúe el puntero del mouse sobre una celda en la columna del operando absoluto y haga clic. La fila seleccionada se convierte en la fila en uso, tal como pone de manifiesto el triángulo sobre el correspondiente selector de fila.

Escriba el nombre del operando absoluto. En cuanto se inicie la inserción, la imagen del triángulo situada sobre el selector de fila se transforma en lápiz, indicándole que la fila en uso se está editando.

Una vez completada la inserción del operando absoluto, pulse Intro o la flecha a la derecha para situar el cursor en el campo siguiente. Teclee el símbolo y vuelva a pulsar Intro o flecha a la derecha para pasar al campo comentario. Inserte el comentario y pulse Intro para aceptar la atribución introducida. Si ésta es correcta, la declaración se admite; de lo contrario, una ventana de diálogo le señalará el error.

PC-Sim controla la corrección de la declaración introducida señalándole posibles errores con los siguientes mensajes:

- En la declaración falta el operando absoluto
- Operando absoluto incorrecto
- En la declaración falta el símbolo
- Operando absoluto ya presente
- Operando simbólico ya presente
- Atribución ya presente

Para una explicación detallada diríjase al apartado [Mensajes de aviso](#) situado más adelante en este mismo capítulo.

No se pueden asociar a un operando absoluto dos a más símbolos ni a un único símbolo dos o más operandos absolutos. Por eso, la inserción de las atribuciones está controlada por el programa también desde el punto de vista de la univocidad, en los términos que se acaban de explicar.

Modificar una atribución

Para modificar una atribución es preciso seguir el procedimiento aquí descrito.

- seleccione la celda a modificar haciendo clic sobre ella
- si desea sobrescribir por completo el nuevo texto, tecléelo directamente en el campo
- si desea modificar parte del texto preexistente, pulse la tecla F2 que le permite entrar en el editing de la celda y a continuación introduzca las modificaciones deseadas en el campo correspondiente
- pulse Intro tantas veces como sea necesario para pasar a la fila siguiente.

Eliminar una atribución

Seleccione la atribución o las atribuciones a eliminar en la *Tabla de símbolos* y pulse *Eliminar*. La atribución seleccionada se elimina y las siguientes ascienden una fila en la tabla.

Importar las atribuciones

PC-Sim le permite, además, recuperar las atribuciones que ya ha efectuado en otro proyecto y añadirlas a la *Tabla de símbolos* del proyecto en uso. Para hacer eso elija el *Comando Importar símbolos...* del *Menú Archivo* y a continuación teclee o seleccione el nombre del proyecto del cual va a importar en la ventana de dialogo que aparece.

Menú de acceso rápido

En toda la *Tabla de símbolos* puede abrir el menú de acceso rápido haciendo clic con el botón derecho del mouse sobre él. Los comandos activables de este menú son los siguientes.

Eliminar

Misma función que el pulsador homónimo.

Insertar

Misma función que el pulsador homónimo.

Barra pulsadores

Esconde o visualiza la *Barra de pulsadores* de la ventana *Tabla de símbolos*. Si se acostumbra a usar los menús de acceso rápido, la *Barra de pulsadores* dejará de serle útil, de manera que podrá ocultarla para ganar espacio para la visualización de la tabla.

Ventana Estado variables

En la siguiente descripción entenderemos por 'variable' un operando cualquiera de uno o más bits, válido para el PLC en uso.

'Estado' de la variable será el contenido de un operando booleano y podrá asumir los valores 0 ó 1.

'Valor' de la variable será el contenido de un operando de más bits (8 ó 16) y asumirá los valores representables con relación al número de bits.

La *Ventana Estado variables* permite la lectura del estado y/o del valor de todas las variables del PLC, cuando la CPU se encuentra en estado de STOP o de RUN. Se podrá conocer el estado y/o el valor de las variables de entrada, de salida, de memoria, de los contadores, de los temporizadores, etc. simplemente tecleando la sigla, o el símbolo que la representa, en la tabla correspondiente. El contenido de esta última se actualizará en tiempo real y en relación con los resultados de la elaboración del programa y de las variaciones de las señales activas.

Además, la tabla permitirá al usuario programar manualmente el estado y/o el valor de las variables introducidas.

Apertura de la ventana

Para abrir la ventana *Estado variables*, o para llevarla a primer plano, opere de una de las siguientes maneras:

- Elija el [Comando Estado variables](#) en el [Menú Archivo](#);
- haga clic sobre el [Pulsador Estado variables](#) de la [Barra instrumentos](#) de PC-Sim ;

Si lo desea, puede cambiar el tamaño y la posición de esta ventana a su gusto. Las últimas variaciones de dimensión y de posición se guardarán con el archivo proyecto de manera que, al volver a abrirlo, PC-Sim se encargará de restablecer las últimas características guardadas de la ventana.

Cierre de la ventana

La ventana *Estado variables* puede cerrarse haciendo clic sobre el pulsador de cierre situado en el extremo derecho de la barra del título. El cierre de la ventana no implica la pérdida de sus atribuciones sino tan sólo la desaparición de ésta de la pantalla.

Guardar las variables

Como ya se ha explicado, el listado de las variables activas se guardan en el archivo proyecto junto con el código del programa y a las demás informaciones descritas en el apartado [Composición de un proyecto](#).

En cualquier momento pueden guardarse las variables insertadas en la ventana. Si desea guardarlas en el archivo proyecto en uso elija el [Comando Guardar proyecto](#) del [Menú Archivo](#). Por el contrario, si desea guardarlas utilizando un nombre de proyecto distinto, elija, en el mismo menú, el [Comando Guardar proyecto como...](#)

Composición de la ventana

Tabla variables

La *Tabla variables* está organizada en 100 filas, tantas como variables pueden introducirse, y cuatro columnas.

Columna Operando absoluto

Muestra el operando absoluto al que se refiere el estado y/o valor expresados en las celdas de las columnas homónimas de la misma fila.


Columna Símbolo

Muestra, si está declarado, el símbolo correspondiente al operando absoluto de la primera columna tal como se ha definido en la [Tabla de símbolos](#).

Columna Estado

Tan sólo tiene significado para operandos de bit y muestra '0' o '1' según si el estado de la variable en la fila es bajo o alto.

Si la variable no es de bits, la celda en esta columna aparecerá vacía.



Op. absoluto	Símbolo	Estado	Base	Valor
EW80	LMIX		KF	+9304
EW82	LPRO		KF	+4096
EW84	TEMP		KF	+7376
MW100	WORDLMIX		KF	+651
MW102	WORDLPRO		KF	+0
MW104	WORDTEMP		KF	+410
Z0	CON	1	KZ	001
M3.0	ABILITA	1		
T0	RIT	1	KT	67.0
M6.3	ERRCH1	0		
M0.0	PASS01	0		
M7.2	RIC1	1		

Columna Base

Esta columna sólo tiene significado para operandos de más bits. En ella se puede programar el valor de la base de representación de la cantidad visualizada en la [Columna Valor](#).

Si la variable es de bits, la celda en esta columna aparecerá vacía.

Las bases válidas para FPC 404 y PS3 son las siguientes:

- 2 representación binaria
- 10 representación decimal sin signo
- ±10 representación decimal con signo
- 16 representación hexadecimal

Las bases válidas para S5 100U son las siguientes:

KM representación binaria

KF representación decimal con signo

KH representación hexadecimal

KY representación doble decimal sin signo

KT representación valor de tiempo

KZ representación valor de conteo

En los capítulos dedicados a las notas de programación para los distintos PLC encontrará más información sobre los tipos de representación.

Columna Valor

Tan sólo tiene significado para los operandos de byte y de word además de para los temporizadores, contadores y registros de shift. Muestra el valor del operando especificado en la columna *Op.absoluto* expresado en la base de representación especificada en la columna *Base*.

Selectores de fila

La parte de la tabla situada a la izquierda de la primera columna está formada por los selectores de fila. Estos le permiten seleccionar las filas de la tabla y, simultáneamente, facilitan informaciones sobre la fila correspondiente mediante la visualización de imágenes:



fila en uso



fila en uso en edición

Barra de pulsadores

La *Barra de pulsadores* de la ventana *Estado variables* contiene los pulsadores necesarios para las operaciones de inserción y de eliminación de variables sencillas o múltiples.

Para seleccionar una variable sencilla haga clic sobre el selector de fila, situado a la izquierda de la fila de la tabla que contiene la variable a seleccionar.

Para seleccionar líneas múltiples contiguas: seleccione la primera fila tal como se ha descrito anteriormente, a continuación presione y mantenga presionada la tecla Mayús y haga clic sobre el selector de la última fila. Todas las líneas, de la primera a la última, quedarán seleccionadas y todas serán objeto de las acciones de los pulsadores *Eliminar* e *Insertar*.

Por último, pueden también efectuarse selecciones múltiples de líneas no contiguas: seleccione la primera; a continuación, presione y mantenga presionada la tecla Control y haga clic sobre los selectores de las líneas que desea agregar a la selección, de uno en uno. Un clic sobre el selector de una fila no seleccionada añadirá la fila a la selección; mientras que un clic sobre el selector de una fila ya seleccionada eliminará la fila de la selección.

Pulsador Eliminar

Utilice el pulsador *Eliminar* para eliminar una o más variables de la tabla. Seleccione las líneas de la tabla que contengan las variables a eliminar y pulse *Eliminar*.

Pulsador Insertar

Utilice este pulsador para insertar una o más filas vacías en la tabla. Seleccione la fila de la tabla anterior a la que desea insertar, a continuación seleccione, si fuera necesario, otras líneas con tal que sean contiguas, y accione el *Pulsador Insertar*.

Las nuevas filas vacías se insertarán a partir de la primera fila seleccionada y las posteriores descenderán hacia abajo en la tabla. Preste atención al hecho de que las variables de las últimas filas útiles de la tabla podrían perderse.

Introducción de variables

Para introducir una variable en la tabla haga clic sobre una casilla vacía en la primera columna y escriba el operando absoluto o el símbolo (precedido por el carácter '-') y pulse Intro. Este procedimiento es válido aunque se desee sobrescribir una variable introducida anteriormente.

Con independencia del hecho de que haya escrito la sigla de un operando absoluto o un símbolo, el operando absoluto se colocará en la primera columna y el correspondiente símbolo, si se le ha atribuido, en la segunda.

Introducción de data byte y data word (S5 100U)

La introducción de data byte y data word para el PLC S5 100U necesita una descripción aparte.

Para que PC-Sim recupere el valor de tales variables y lo muestre en la ventana, será preciso informarle acerca del número del bloque de datos al que pertenece. Para hacerlo, escriba en una celda de la *Columna Operando absoluto* la sigla del bloque de datos y, a continuación, en las filas siguientes, la o las variables de datos a visualizar.

Una vez escrita la sigla de un bloque de datos, esta referencia se mantiene activa en todas las filas siguientes, hasta una nueva sigla de bloque de datos. La fila en la que se especifica el nombre del bloque no se utiliza para mostrar ningún otro valor.

PC-Sim podría no conseguir recuperar el valor de un byte o de una palabra de datos si no ha especificado primero el bloque de pertenencia, si el bloque especificado no existe en el proyecto o bien si el dato no existe, es decir, si el número del dato requerido es mayor que la última data word insertada en aquel bloque. En ese caso PC-Sim muestra en la correspondiente celda de la *Columna Valor* el texto '???'.

Para más informaciones sobre el uso de bloques de datos ver el *Capítulo 14* y el apartado *El editor de datos (S5 100U)* en este capítulo.

Introducción de constantes

Si bien no pertenecen en sentido estricto a la categoría de las variables, las constantes también son operandos admisibles en la programación de los PLC y, como tales, pueden incluirse en la definición dada al principio de este apartado. Así pues, la columna *Op.absoluta* también acepta operandos de este tipo.

Puede utilizar esta característica de la tabla para conversiones de base. Por ejemplo, si desea conocer la representación hexadecimal del número decimal 2000 y utiliza el PLC PS3, escriba en la columna *Op.absoluta* la constante $\kappa W2000$ e introduzca la base hexadecimal ('16') en la columna *Base*. En la columna valor aparecerá el valor convertido 7D0.

Introducción múltiple de variables

Existe un método de introducción rápida de variables sucesivas. Si desea conocer, por ejemplo, el valor de las 8 primeras entradas del PLC PS3, haga clic sobre una celda de la columna *Op.absoluta* y escriba $I0.0*8$, donde $I0.0$ representa la primera de las entradas a introducir, el carácter '*' informa a PC-Sim que deseamos realizar una introducción múltiple y el número final le informa de cuantos operandos queremos introducir. Pulsando Intro, el operando $I0.0$ se introducirá en la celda en uso y en las filas siguientes aparecerán automáticamente los operandos sucesivos, hasta el $I0.7$.

El método descrito es válido para cualquier tipo de operando y para todos los PLC.

Forzado de variables

La ventana *Estado variables* permite, además, la programación manual del valor o del estado de una variable, es decir, el forzado de la misma.

Se pueden forzar bit, byte o word de memoria, valores de los contadores, entradas, salidas, registros de desplazamiento (PS3), data byte y data word (S5 100U). Recuerde que, con CPU en RUN, el programa del usuario podría reescribir las variables recién forzadas, haciendo que esta función pareciera desactivada.

Forzado de bit de merker (flag)

Para forzar un bit de merker (flag) haga clic sobre la columna de estado correspondiente a la fila que contiene el operando a forzar y escriba 1 si quiere programar un valor lógico alto o 0 si desea un valor lógico bajo. A continuación pulse Intro.

Como método alternativo, haga clic sobre la misma celda con el botón derecho del mouse y, en el [Menú de acceso rápido](#) seleccione el comando relativo al valor que desea programar.

Forzado de byte y word de merker (PS3 y S5 100U)

Para forzar un merker byte o una merker word haga clic sobre la celda correspondiente a la fila que contiene el operando a forzar en correspondencia con la columna valor, escriba el nuevo valor y pulse Intro. El valor escrito deberá ser coherente con la base de representación programada para la variable en uso y con su número de bit. Por ejemplo, si está forzando un merker byte del PLC PS3 con representación decimal ('10'

en la columna *Base*) sólo podrá insertar valores entre 0 y 255. Una señal acústica señalará cada inserción situada fuera de este margen y la indicación de la celda volverá al valor anterior al introducido erróneamente.

Forzado de contadores

Para forzar el valor de conteo de un contador, opere tal como se ha descrito para las merker word. El estado del contador también variará en concordancia con el valor insertado. Es decir, por ejemplo, si fuerza el valor del contador Z15 del PLC S5 100U a 0, el estado del propio contador se actualizará automáticamente a 1.

Para más informaciones sobre el funcionamiento de los contadores, ver los capítulos sobre notas de programación de los PLC.

Forzado de registros de desplazamiento (PS3)

Para el PLC PS3 puede forzarse el valor de los registros de desplazamiento y por tanto influir directamente en las salidas de este módulo de sistema. Siga todo lo descrito para las merker word.

Forzado de byte y palabras de datos (S5 100U)

Para el PLC S5 100U se puede forzar el valor de data byte y data word. Opere tal como se ha descrito para las merker word pero recuerde todo lo dicho en el apartado [Introducción de data byte y data word \(S5 100U\)](#).

Forzado de salidas (PS3 y S5 100U)

Con CPU en el estado de STOP puede efectuarse también el forzado de las salidas, como bit, byte o word. Escriba el valor deseado en la columna de estado para los bit o en la columna *Valor* para los byte o las word y pulse Intro. Las salidas se activarán o desactivarán según la introducción efectuada, pero la indicación en la celda no variará ya que esta indica el valor de los correspondientes bit de las memorias de imagen y no de los canales físicos de salida. Respecto a ello, ver lo descrito en el capítulo dedicado a las notas de programación del PLC que esté utilizando.

Al pasar a RUN las salidas se borrarán.

Menú de acceso rápido

Haciendo clic con el botón derecho del mouse sobre la *Tabla de variables* se activa un menú de acceso rápido. Los comandos contenidos en este menú varían en función de la columna sobre la que se ha hecho clic.

En las columnas *Op.absoluto* y *Símbolo* se visualizan los comandos *Eliminar* e *Insertar*, cuya acción es equivalente a la de los pulsadores homónimos de la barra.

En la columna de estado se visualizan los comandos *Forzar a 0* y *Forzar a 1* que permiten programar manualmente el valor de un operando booleano.

Sin embargo, en todas las columnas son activos los comandos que se describen a continuación.

Siempre en primer plano

Activando este comando la *Ventana Estado variables* se mantendrá siempre en primer plano, es decir, no quedará escondida por otras ventanas. Este comando es muy útil cuando se precisa ir controlando el valor de una o más variables y a la vez es necesario efectuar otras operaciones que podrían hacer que las demás ventanas se superpusieran a la de *Estado variables*, ocultándola del todo o en parte.

Para desactivar esta función, vuelva a seleccionar el comando en el menú de acceso rápido. Un signo de "visto" junto a la opción del menú le recuerda cuando esta función está activada.

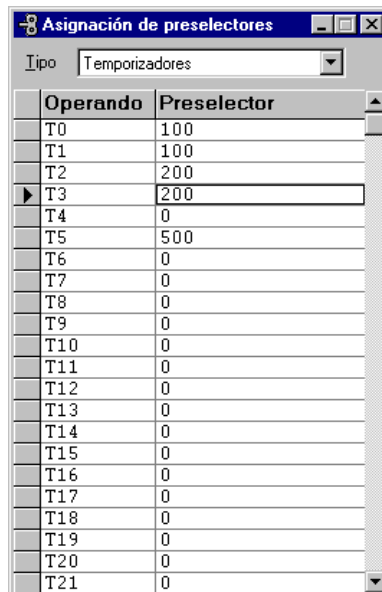
Barra de pulsadores

Oculto o visualiza la *Barra de pulsadores* de la ventana *Estado variables*. Si se acostumbra a usar los menús de acceso rápido la *Barra de pulsadores* dejará de serle útil, de manera que podrá ocultarla y ganar espacio para la visualización de la tabla.

Ventana Asignación de preselectores (FPC 404)

Esta ventana se utiliza para la asignación de los valores de tiempo y de conteo en la simulación del PLC FPC 404. De hecho, el editor de programa de este PLC no permite realizar estas operaciones directamente, tal como ocurre en los demás PLC simulados.

Los valores programados en esta ventana se guardan en el archivo proyecto y se recuperan cuando vuelve a abrirse.



Operando	Preselector
T0	100
T1	100
T2	200
T3	200
T4	0
T5	500
T6	0
T7	0
T8	0
T9	0
T10	0
T11	0
T12	0
T13	0
T14	0
T15	0
T16	0
T17	0
T18	0
T19	0
T20	0
T21	0

Apertura de la ventana

Para abrir la ventana *Asignación de preselectores*, o para llevarla a primer plano, opere de una de las siguientes maneras:

- Elija el *Comando Asignación de preselectores (FPC 404)* del *Menú Ventana*;
- haga clic sobre el pulsador *Asignación de preselectores* de la *Barra instrumentos* de PC-Sim ;

Cierre de la ventana

La ventana *Asignación de preselectores* puede cerrarse haciendo clic sobre el pulsador de cierre situado en el extremo derecho de la barra del título. El cierre de la ventana no comporta la pérdida de las programaciones sino tan sólo la desaparición de ésta de la pantalla.

Guardar los preselectores

Tal como se ha explicado ya, los valores de preselector se guardan en el archivo proyecto junto al código del programa y el resto de informaciones descritas en el apartado *Composición de un proyecto*.

En cualquier momento pueden guardarse los valores introducidos en la ventana. Si desea guardarlas en el archivo proyecto en uso elija el *Comando Guardar proyecto* del *Menú Archivo*. Por el contrario, si desea

guardarlos utilizando un nombre de proyecto distinto elija el *Comando Guardar proyecto como...* en el mismo menú.

Composición de la ventana

Casilla Tipo

Permite la elección del tipo de operando para la programación de los valores de preselector.

Las opciones posibles son: Temporizadores y Contadores.

Tablas de los preselectores

Las tablas de los preselectores incluyen el listado de los valores de tiempo de los temporizadores y de los de conteo de los contadores presentes en el PLC. Sólo es visible una de las dos tablas en cada ocasión. Para seleccionar la tabla a visualizar utilice la casilla *Tipo*.

Columna Operando

Presenta el listado de las siglas de los temporizadores o de los contadores presentes en el PLC. Las celdas de esta columna no pueden modificarse.

Columna Preselector

Presenta el listado de los valores de preselector de los temporizadores o de los contadores especificados en la primera columna. Los valores de tiempo se expresan en centésimas de segundo.

Programación de los preselectores

Programación de un preselector de temporización

Para programar o modificar un valor de preselector de tiempo haga clic sobre la columna *Preset* y en la fila que incluye la sigla del temporizador a programar y escriba el nuevo valor. A continuación pulse Intro. El valor debe expresarse en centésimas de segundo. Por tanto, si por ejemplo quiere programar un tiempo de 5 segundos, escriba 500.

El máximo valor programable es 65535, es decir, 655.35 segundos.

Programación de un preselector de conteo

Para programar o modificar un valor de preselector de conteo haga clic sobre la columna *Preset* y en la fila que indica la sigla del contador a programar y luego escriba el nuevo valor. A continuación pulse Intro.

El máximo valor programable es 65535.

Ventana PLC

Esta ventana muestra la imagen del PLC y, obviamente, su contenido varía en función del modelo que esté utilizando. Esta ventana ha sido estudiada y realizada para representar una instalación lo más parecida posible en sus formas externas al PLC real. Esto, aunque no tiene utilidad desde un punto de vista funcional, le facilitará el camino hacia el PLC real, cuando decida hacerlo. En todas las ventanas se han incorporado las señales luminosas con LED del estado de las entradas y de las salidas digitales. Igual que en el PLC real el encendido de un LED de entrada indica el cierre del contacto conectado a él y el encendido de un LED de salida indica la activación de la salida correspondiente.

Estado de la CPU

La CPU puede estar en uno de los siguientes estados:

OFF: la CPU no recibe alimentación, el conmutador de encendido en la *Barra instrumentos* de PC-Sim está en posición de Off. Sin embargo, en este estado, los módulos de I/O permanecen alimentados para el PLC S5 100U. En efecto, cerrando uno de los interruptores de mando de las entradas digitales se activa el correspondiente LED. En cambio, para los otros dos PLC, también la periferia deja de recibir alimentación y por tanto no sucede nada al accionar los interruptores.

STOP: la CPU recibe alimentación pero el programa del usuario no se ejecuta, el conmutador de encendido en la *Barra instrumentos* de PC-Sim está en posición de On, y el conmutador de funcionamiento está en posición de Stop. En este estado, para los PLC FPC 404 y PS3 reciben alimentación los periféricos: el cierre de los interruptores de entrada se indicará con el encendido de los LED correspondientes.

RUN: la CPU recibe alimentación y se ejecuta el programa del usuario, el conmutador de encendido en la *Barra instrumentos* de PC-Sim está en posición de On y el conmutador de funcionamiento se encuentra en posición de Run.

El estado actual de la CPU se visualiza en la barra del título de la *Ventana PLC* y aparece, entre corchetes, junto al indicador del modelo.

Cuando arranque PC-Sim, cuando cree un nuevo proyecto o cuando cargue uno ya existente, la CPU pasa automáticamente al estado de OFF. Cuando sitúe el conmutador de encendido en On la CPU pasa incondicionalmente al estado de STOP. Cuando, a continuación, ponga el conmutador del modo de funcionamiento en Run, el programa del usuario se compila y, si no se detecta ningún error, se transfiere al PLC. En este momento es cuando la CPU pasa a estado de RUN e inicia la elaboración del programa del usuario.

Si, partiendo del estado de OFF, sitúa el conmutador del modo de funcionamiento en Run, no ocurre nada. Cuando, en este momento, pone el conmutador de encendido en On, el programa del usuario se compila y se transfiere al PLC: entonces la CPU pasa directamente al estado de RUN.

Cuando ha producido un error en la ejecución del programa del usuario, la CPU pasa automáticamente al estado de STOP.

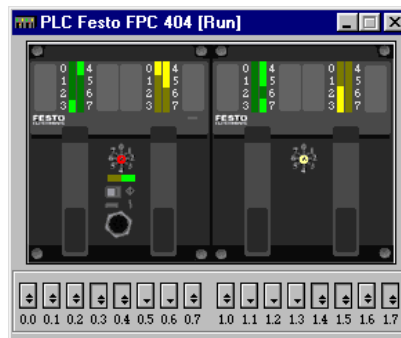
FPC 404

En la simulación del FPC 404 la ventana incluye un PLC compuesto por dos módulos:

- módulo 0 - CPU con 8 entradas digitales y 8 salidas digitales;
- módulo 1 - I/O con 8 entradas digitales y 8 salidas digitales.

El tipo, el número y la posición de los módulos no pueden modificarse.

En la parte frontal del módulo 0 aparecen los LED luminosos indicadores del estado de la CPU.



LED Stop

El LED es de color amarillo y se enciende cuando la CPU recibe alimentación, es decir, cuando el pulsador *CPU On* en la *Barra instrumentos* de PC-Sim está presionado y el PLC está en Stop, es decir cuando el pulsador *CPU Stop* está presionado.

LED Run

El LED es de color amarillo y se enciende cuando la CPU está en Run, es decir, cuando los pulsadores *CPU On* y *CPU Run* de la *Barra instrumentos* de PC-Sim están presionados y no se ha verificado ningún error en el programa.

Encontrará más informaciones acerca de este PLC en el [Capítulo 9](#).

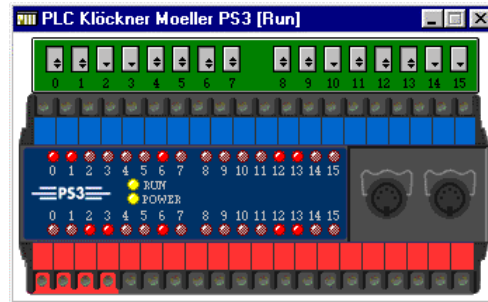
PS3

Este PLC es un 'compacto' y por tanto no tiene una estructura modular. Dispone de 16 entradas digitales, 16 salidas digitales, 4 entradas analógicas y 1 salida analógica.

En la parte frontal del módulo aparecen los LED indicadores del estado de la CPU.

LED Power

El LED es de color amarillo y se enciende cuando la CPU recibe alimentación, es decir, cuando el pulsador *CPU On* de la *Barra instrumentos* de PC-Sim está accionado.



LED Run

El LED es de color amarillo y se enciende cuando la CPU está en Run, es decir, cuando los pulsadores *CPU On* y *CPU Run* de la *Barra instrumentos* de PC-Sim están accionados y no se ha detectado ningún error en el programa.

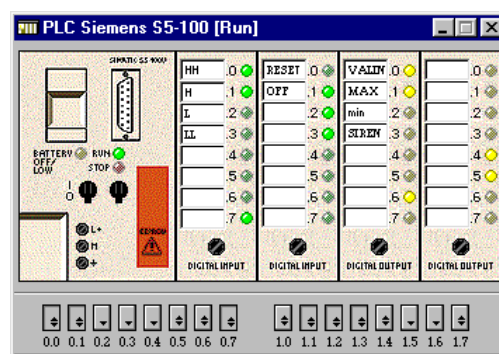
Encontrará más informaciones sobre este PLC en el [Capítulo 12](#).

S5 100U

El PLC simulado está compuesto por los siguientes módulos:

- CPU;
- módulo 0: 8 entradas digitales;
- módulo 1: 8 entradas digitales;
- módulo 2: 8 salidas digitales;
- módulo 3: 8 salidas digitales;
- módulo 4: 4 entradas analógicas (no visualizado)
- módulo 5: 4 salidas analógicas (no visualizado)

El tipo y la posición de los módulos no pueden modificarse.



En correspondencia con cada canal de entrada o de salida aparecen etiquetas para la inserción del tag de la señal en vigor. Para escribir una breve descripción de la señal haga clic con el mouse sobre la etiqueta y escriba: se permiten un máximo de cinco caracteres. El contenido de las etiquetas se guardará junto con el proyecto y se recargará cuando éste vuelva a abrirse.

En la parte frontal del módulo CPU están los LED indicadores del estado.

LED Stop

El LED es de color rojo y se enciende cuando la CPU recibe alimentación y se encuentra en estado de Stop, es decir, cuando los pulsadores *CPU On* y *CPU Stop* de la [Barra instrumentos](#) de PC-Sim están accionados. Cuando está apagado sucede lo contrario.

LED Run

El LED es de color verde y se enciende cuando la CPU está en Run, es decir, cuando los pulsadores *CPU On* y *CPU Run* de la [Barra instrumentos](#) de PC-Sim están presionados y no se ha verificado ningún error en el programa.

LED Battery Off/Low

El LED es de color amarillo y se enciende cuando la CPU recibe alimentación y la batería está descargada o no está presente, es decir, cuando los pulsadores *CPU On* y *Batería fuera de servicio* de la [Barra instrumentos](#) de PC-Sim están activados.

Pulsadores/interruptores

En la ventana de simulación del PLC aparecen 16 pulsadores/interruptores que se utilizan para el control de las entradas digitales cuando el PLC se usa sin instalación o cuando no todas las entradas del PLC están conectadas a la instalación.

Estos pulsadores/interruptores pueden programarse como pulsadores o como interruptores; su aspecto especificará su función:



interruptor;



pulsador;

donde una doble flecha indica la función biestable y una sola flecha, la función monoestable.

Para intercambiar las distintas programaciones, haga clic con el botón derecho del mouse sobre el elemento que desee cambiar; a cada presión se conmuta la función: si es pulsador se convierte en interruptor y viceversa.

Cuando el interruptor asume este aspecto:



desactivado

éste se desactiva: hacer clic sobre él no producirá ningún efecto.

PC-Sim desactiva automáticamente un interruptor o un pulsador cuando a la correspondiente entrada del PLC se conecta el sensor de una instalación (por ejemplo, un final de carrera de cilindro o un detector de nivel de un depósito) para evitar interferencias inoportunas entre los contactos de componentes de este último y los pulsadores/interruptores locales.

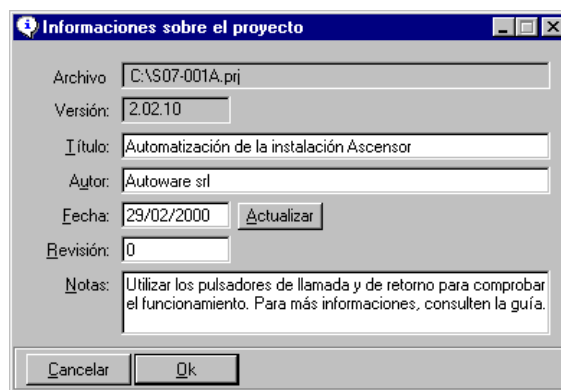
Para activar nuevamente el pulsador/interruptor será necesario desconectar la entrada del PLC de la instalación. Para más informaciones, lea la sección [Conectar las instalaciones al PLC](#) en el [Capítulo 4](#).

Ventana Informaciones sobre el proyecto

Esta ventana permite obtener o introducir informaciones sobre el proyecto en uso. Esta ventana muestra, en los oportunos campos, el nombre del archivo del proyecto y la versión de PC-Sim con el que se ha realizado. La introducción de datos en los otros campos corre a cargo del usuario, que podrá escribir en ellos el título del ejercicio desarrollado, sus datos personales, etc.

Todas estas informaciones se imprimirán con la documentación del proyecto y constituirán referencias válidas para su archivo y su reconocimiento.

A continuación, sigue una breve descripción de los campos que, de todos modos, no es obligatorio formalizar.



Campo Archivo proyecto

El campo detalla el nombre del archivo proyecto en uso. Para abrir un nuevo proyecto o cambiar de nombre al actual, vea lo dicho a propósito de los comandos del [Menú Archivo](#) de PC-Sim.

Este campo no puede modificarse y por eso aparece con fondo gris.

Campo Versión

Detalla la versión de PC-Sim con el que se ha realizado el proyecto o con el que se guardó la última vez.

Tampoco este campo puede modificarse. PC-Sim efectúa automáticamente su actualización, y por eso aparece con fondo gris.

Campo Título

Inserte en este campo el título que desea atribuir al ejercicio que ha realizado.

Campo Autor

Inserte su nombre o los de los miembros del grupo de trabajo que ha realizado el ejercicio.

Campo Fecha

En este campo aparece automáticamente la fecha, a cada apertura de un nuevo proyecto. Si está modificando un proyecto realizado con anterioridad pulse *Actualizar* para insertar en el campo la fecha actual o bien escríbala manualmente.

Campo Revisión

Inserte el número de revisión del proyecto. Cada vez que se aportan modificaciones al trabajo sería conveniente incrementar este número para así diferenciar las actualizaciones.

Campo Notas

Escriba en esta casilla un texto de aclaración o de comentario al ejercicio desarrollado. Durante la fase de impresión, si este texto es demasiado largo, se reducirá para ocupar hasta un máximo de dos líneas en el correspondiente recuadro destinado a la impresión de notas.

Barra pulsadores

Pulsador Cancelar

Cierra la ventana y cancela todas las modificaciones aportadas desde su apertura.

Pulsador Ok

Cierra la ventana y convalida las modificaciones introducidas. Las modificaciones sólo se guardarán en el archivo en el momento en que se guarde el proyecto.


Ventana Informaciones sobre PC-Sim

Esta ventana aparece cada vez que arranca PC-Sim pero puede abrirse en cualquier momento seleccionando el *Comando Informaciones sobre PC-Sim* del *Menú ?*.

Puede cerrarse accionando el *Pulsador Ok*.


En la parte superior de esta ventana puede encontrar información acerca de la versión del programa que está utilizando, el modelo del PLC simulado, el número de autorización de su código hardware.

En su parte inferior se muestran informaciones sobre el productor del software.



Copyright © 1995-2000 Autoware srl
Todos los derechos reservados

Software de simulación
de PLCs y de instalaciones



PC-Sim - ver: 2.02.10
Software didáctico de simulación para control lógico programable

Siemens S5 100U

Autorización código nº.: 06.0015.0000
**¡Atención! El código en uso forma parte de una multilicencia de 6 copias.
Si el número de código en su poder es inferior, sus copias no están autorizadas.
Contacte inmediatamente con su proveedor o directamente con Autoware.**

Este programa está protegido por las leyes sobre el copyright, las leyes sobre los derechos de autor y por las disposiciones sobre los tratados internacionales. La reproducción o distribución de este programa o de parte de él, será perseguida civil y penalmente.

Realización:
AUTOWARE
e-mail: support@autoware.com
http://www.autoware.com

Comandos de los menús

Menú Archivo

Comando Nuevo proyecto (S5 100U)

Este comando da inicio a un nuevo proyecto. Una vez se ha seleccionado, PC-Sim realiza las siguientes acciones:

- pregunta si se desea guardar el proyecto en uso si éste ha sufrido cambios;
- pone en OFF el PLC;
- borra la memoria de programa;
- vacía la [Tabla de bloques](#);
- borra las atribuciones de los símbolos;
- inicializa el nombre del proyecto en uso;
- borra los operandos en la [Tabla variables](#);
- borra las etiquetas del PLC;
- borra las informaciones sobre el proyecto.

También puede iniciarse un nuevo proyecto accionando el *Pulsador Nuevo proyecto* en la [Barra instrumentos](#) de PC-Sim.

Comando Nuevo proyecto ladder (FPC 404 y PS3)

Este comando da inicio a un nuevo proyecto utilizando el editor de programa ladder. Tras su selección, PC-Sim realiza las siguientes acciones:

- pregunta si se desea guardar el proyecto en uso si éste ha sufrido cambios;
- pone en OFF el PLC;
- borra la memoria de programa;
- borra las atribuciones de los símbolos;
- inicializa el nombre del proyecto en uso;
- borra los operandos en la [Tabla variables](#);
- inicializa los valores de los preselectores (FPC 404);
- borra las informaciones sobre el proyecto.
- abre el editor de programa ladder y la casilla instrumentos.

Como alternativa a este comando puede accionarse el *Pulsador Nuevo proyecto ladder* en la [Barra instrumentos](#) de PC-Sim.

Comando Nuevo proyecto AWL (PS3)

Este comando da inicio a un nuevo proyecto utilizando el editor AWL. Tras su selección PC-Sim realiza las siguientes acciones:

- pregunta si se desea guardar el proyecto en uso si éste ha sufrido cambios;
- pone en OFF el PLC;
- borra la memoria de programa;
- borra las atribuciones de los símbolos;
- inicializa el nombre del proyecto en uso;
- borra los operandos en la [Tabla variables](#);
- borra las informaciones sobre el proyecto.
- abre el editor de programa AWL.

Como alternativa a este comando puede accionar el *Pulsador Nuevo proyecto AWL* en la [Barra instrumentos](#) de PC-Sim.

Comando Abrir proyecto...

Este comando permite abrir un proyecto realizado y guardado con anterioridad. Como respuesta a la selección de este comando, PC-Sim mostrará la ventana de diálogo *Abrir proyecto*. Seleccione en ella la ruta y el nombre del archivo proyecto (.prj) que va a abrir y haga clic sobre el *Pulsador Abrir*.

Cuando se abre un proyecto, PC-Sim realiza las siguientes acciones:

- pregunta si se desea guardar el proyecto en uso si éste ha sufrido modificaciones;
- pone en OFF el PLC;
- carga el listado de bloques en la [Tabla de bloques](#) (S5 100U);
- lee el programa del usuario del archivo, transfiriéndolo a la memoria de programa;
- carga las atribuciones de los símbolos;
- actualiza el nombre del proyecto en uso;
- carga los operandos en la [Tabla variables](#);
- carga las etiquetas del PLC (S5 100U);
- carga los valores de los preselectores (FPC 404);
- actualiza las informaciones sobre el proyecto.
- recupera las dimensiones y la posición programadas de la [Ventana Lista bloques \(S5 100U\)](#), de la [Ventana Tabla de símbolos](#) y de la [Ventana Estado variables](#);
- abre el editor de programa ladder o AWL en función del tipo de código en la memoria de programa (FPC 404 y PS3)

También puede abrirse un proyecto accionando el *Pulsador Abrir proyecto* en la [Barra instrumentos](#) de PC-Sim.

Opciones de la ventana de diálogo Abrir proyecto

Buscar en:

Seleccione la ruta en la que PC-Sim deberá leer el archivo que desea abrir.

Nombre archivo

Escriba el nombre del archivo que desee abrir o bien selecciónelo en el listado. Este último detalla los archivos con la extensión seleccionada en la casilla *Tipo de archivo*.

Tipo de archivo

Seleccione el tipo de archivo que se desea visualizar en el listado. Las opciones posibles son:

- Archivo conexiones (*.prj) registra los archivos con extensión 'prj' presentes en la carpeta en uso.
- Todos los archivos (*.*) registra todos los archivos de la carpeta en uso, prescindiendo de su extensión.

Comando Guardar proyecto

Este comando permite guardar en el disco el proyecto en uso. Como respuesta a su elección, PC-Sim controla que el proyecto en uso tenga un nombre. Si el proyecto abierto aun no tiene nombre, es decir, si lo está guardando por primera vez, abre la ventana de diálogo *Guardar con nombre* que le permitirá asignar un nombre al proyecto en uso (ver [Comando Guardar proyecto como...](#)).

Tras haber dado un nombre al proyecto, o si el proyecto en uso ya lo posee, PC-Sim realiza las siguientes acciones:

- registra el contenido del editor de programa abierto, si éste se ha modificado, en la memoria de programa (FPC 404 y PS3);
- registra el contenido de todos los bloques abiertos y modificados en la memoria de programa (S5 100U);
- guarda en el archivo las informaciones sobre el proyecto;
- guarda las dimensiones y la posición programadas de la [Ventana Lista bloques \(S5 100U\)](#), de la [Ventana Tabla de símbolos](#) y de la [Ventana Estado variables](#).
- guarda las atribuciones de los símbolos;
- guarda los operandos de la [Tabla variables](#);
- guarda los textos de las etiquetas del PLC (S5 100U);
- guarda los valores de los preselectores (FPC 404);
- guarda el programa del usuario registrado en la memoria de programa.

El proyecto en uso puede guardarse en cualquier momento durante la utilización del software.

También puede guardarse un proyecto accionando el *Pulsador Guardar proyecto* en la [Barra instrumentos](#) de PC-Sim.

Comando Guardar proyecto como...

Permite guardar un proyecto con un nombre distinto al vigente o bien nombrar y guardar un proyecto sin nombre.

Cuando se acciona este comando, PC-Sim abre la ventana de diálogo *Guardar con nombre*. Seleccione o escriba la ruta y el nombre del archivo en el cual pretende guardar el proyecto y pulse *Guardar*. El proyecto se guardará en el archivo del nuevo nombre. Para una descripción detallada de las operaciones para guardar, vea lo dicho anteriormente para el [Comando Guardar proyecto](#).

Opciones de la ventana de diálogo Guardar proyecto como

Guardar en:

Seleccione la carpeta en la que desea guardar el proyecto.

Nombre archivo

Escriba un nuevo nombre de archivo para guardar el proyecto con un nombre distinto o bien seleccione uno del listado para guardarlo con un nombre de archivo ya existente. PC-Sim se encarga de añadir la extensión 'prj' si no se ha especificado.

El listado incluye sólo los archivos de la carpeta que tienen la extensión seleccionada en la casilla *Tipo de archivo*.

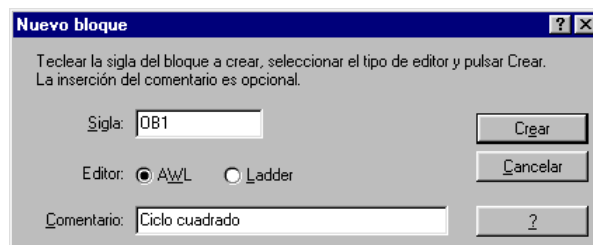
Tipo de archivo

Seleccione el tipo de archivo que desea visualizar en el listado. Las opciones posibles son:

- Archivo proyecto (*.prj) relaciona los archivos con extensión 'prj' presentes en la carpeta en uso.
- Todos los archivos (*.*) relaciona todos los archivos de la carpeta en uso prescindiendo de su extensión.

Comando Nuevo bloque (S5 100U)

Permite crear un nuevo bloque en el programa. Al seleccionar este comando se muestra la ventana de diálogo *Nuevo bloque*. Inserte el nombre del bloque y si es necesario haga clic en el pulsador opción para elegir el tipo de editor a utilizar (AWL o Ladder). Si lo desea, inserte un texto de comentario en la casilla correspondiente y a continuación pulse *Nuevo* para abrir el editor del bloque.



No está permitido crear un bloque que tenga el mismo nombre que otro ya presente en el proyecto. PC-Sim señala dicha operación con un mensaje de error.

Otros métodos equivalentes para crear un nuevo bloque son:

- hacer clic sobre el *Pulsador Nuevo* de la *Ventana Lista bloques (S5 100U)*;
- hacer clic sobre el *Pulsador Nuevo bloque* de la *Barra instrumentos* de PC-Sim;
- pulsar la combinación de teclas Ctrl+N.

Comando Abrir bloque (S5 100U)

Abre el editor del bloque seleccionado en la *Tabla de bloques* o lo lleva a primer plano si ya está abierto. Seleccionar este comando equivale a pulsar el *Pulsador Código* de la *Ventana Lista bloques (S5 100U)*.

Comando Guardar bloque (S5 100U)

Guarda el bloque seleccionado en la *Tabla de bloques* en el interior de la memoria de programa.

Recordamos que guardar en la memoria de programa no comporta el registro en el archivo. Este sólo se efectuará después de un *Comando Guardar proyecto*. Para más información le remitimos a lo dicho a propósito de este último comando.

Comando Eliminar bloque (S5 100U)

Elimina la sigla del bloque seleccionado de la *Tabla de bloques* y su código de la memoria de programa.

Si aún no se ha creado ningún bloque, el comando se mantiene desactivado.

Para más información, le remitimos a lo descrito para el *Pulsador Eliminar* de la *Ventana Lista bloques (S5 100U)* que es equivalente al comando en cuestión.

Comando Importar símbolos...

Permite importar al proyecto en uso los símbolos ya definidos anteriormente en otro proyecto. Cuando se seleccione este comando PC-Sim responde con la visualización de la ventana de diálogo *Importar símbolos* que le permite especificar la ruta y el nombre del archivo proyecto del cual desea recuperar los símbolos.

PC-Sim controla la univocidad de las atribuciones importadas respecto a aquellas eventualmente ya presentes y, si no hay incompatibilidad, las añade a estas últimas.

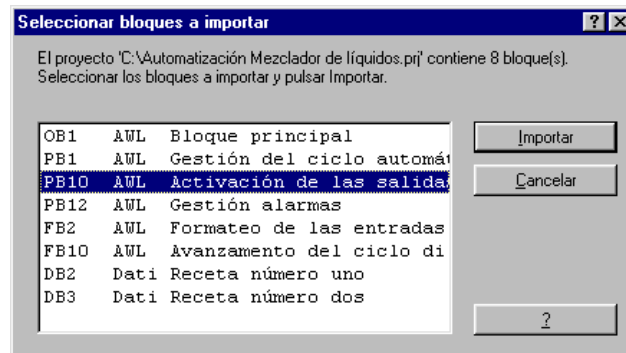
Al término de la operación, un mensaje le informará acerca del número de las atribuciones importadas.

Comando Importar programa...

Utilice este comando cuando desee añadir al proyecto en uso un código ya desarrollado anteriormente. Como respuesta a eso, PC-Sim muestra la ventana de diálogo *Importar programa*. En dicha ventana, escriba o seleccione el nombre del archivo proyecto del cual va a importar y accione el *Pulsador Abrir*.

Para los PLC FPC 404 y PS3 el procedimiento ha finalizado. Si el proyecto indicado contiene un código compatible con el actual (es decir, si es del mismo tipo, ladder o AWL), el programa se lee y se inserta en el editor en uso. Si el editor es AWL, el código se insertará en correspondencia con la posición del cursor. Si el editor es ladder, los nuevos recorridos se insertarán antes del recorrido seleccionado o, si no hay ninguno seleccionado, al final del esquema.

Por el contrario, para el PLC S5 100U el procedimiento comporta un paso más: PC-Sim lee el archivo proyecto preseleccionado y extrae la lista dei bloques. A continuación, muestra la ventana de diálogo *Selección bloques a importar* que detalla la sigla y el comentario de cada bloque encontrado.



Seleccione los bloques que desea importar y pulse *Importar*. Puede realizarse la selección múltiple de bloques contiguos con la tecla Mayús, o no contiguos con la tecla Control.

Mientras en el proyecto en uso no existan bloques con la misma sigla, PC-Sim añade la sigla de cada bloque a la *Tabla de bloques* y registra el correspondiente código en la memoria de programa.

Comando Configurar impresora...

Utilice este comando para seleccionar la impresora que utilizará para imprimir el proyecto en uso y una conexión para ésta.

Opciones de la ventana de diálogo

Impresora: Seleccione la impresora que desea utilizar. Aparecerán solo las impresoras instaladas. Pueden instalarse las impresoras y configurar las correspondientes conexiones por medio del *Panel de control* de Windows. Para más informaciones sobre la instalación de las impresoras, lea la documentación de Windows.

Propiedades: Muestra una ventana de diálogo que permite controlar las propiedades de la impresora seleccionada.

Comando Imprimir documentación...

Permite imprimir el proyecto en uso o parte de éste. A este tema se ha dedicado un apartado completo, con el título de *Impresión de la documentación añadida*, que encontrará más adelante en este capítulo y al cual le remitimos para explicaciones más detalladas.

Ultimos archivos proyecto abiertos

Los nombres de los últimos archivos proyecto que ha utilizado con PC-Sim, con un máximo de seis, se ofrecen en una sección del *Menú Archivo*.

El último proyecto abierto ocupa la primera posición, le sigue el penúltimo y así sucesivamente. El listado de este archivo se actualiza automáticamente a cada apertura, creación o registro de proyecto. Los nombres de los proyectos anteriores al sexto se pierden.

Para abrir uno de los proyectos de la lista, haga clic sobre el comando que incluye el nombre. Este método resulta mucho más rápido que la apertura clásica con el [Comando Abrir proyecto...](#) del mismo menú.

Comando Salir

Sale de la sesión de PC-Sim. Antes del cierre, PC-Sim realiza las siguientes acciones:

- cierra el editor de programa y, si hay variaciones no registradas, le pregunta si desea guardar los cambios en la memoria de programa (FPC 404 y PS3);
- cierra todos los editores de bloque abiertos y, si hay variaciones en el código de los bloques, le pregunta, de uno en uno, si desea guardar los cambios en la memoria del programa (S5 100U);
- si el proyecto ha sido modificado, es decir, si han variado la memoria de programa, las variables de estado, los símbolos o las etiquetas del PLC (S5 100U) o también las programaciones de preselectores (FPC 404), pregunta si se desea registrar las modificaciones en un archivo;
- en caso de respuesta afirmativa, y si no se ha asignado aun un nombre al proyecto, muestra la ventana de diálogo *Guardar proyecto como* para la asignación del nombre del archivo y a continuación Guardar el proyecto en el archivo especificado (ver [Comando Guardar proyecto como...](#));
- si el proyecto abierto ya tiene un nombre y responde que sí desea guardarlo, PC-Sim guarda el proyecto directamente en aquel archivo (ver [Comando Guardar proyecto](#));
- por último, actualiza la lista de los últimos proyectos abiertos en el archivo de inicialización del programa.

Menú Modificar

Comando Eliminar

En el editor AWL del PLC PS3 elimina la línea en uso, haciendo subir una posición al resto del programa.

Por el contrario, para el PLC S5 100U borra la selección en vigor en el editor AWL de bloque. El comando sólo está activado si en la ventana de editor en uso existe una selección de texto. En estas condiciones, la elección del comando conduce a la eliminación del texto seleccionado.

Si el bloque (S5 100U) o el proyecto (FPC 404 y PS3) es de tipo ladder, el comando sólo está activado si en la ventana de editor en uso está seleccionado un recorrido. La elección de este comando provoca la eliminación del recorrido seleccionado y el desplazamiento hacia arriba y la renumeración de los recorridos siguientes.

Comando Insertar

Si el editor en uso es de tipo ladder, inserta un recorrido vacío en correspondencia con la selección hecha. El comando sólo está activado si en la ventana de editor hay un recorrido seleccionado. La elección de este comando conduce al desplazamiento hacia abajo y a la renumeración progresiva de todos los recorridos a partir del que se ha seleccionado y a la inserción de un nuevo recorrido vacío.

Para el editor AWL del PLC PS3 este comando inserta una fila vacía en la posición en uso.

Para el editor AWL del PLC S5 100U este comando no está activado.

Comando Añadir

Si el editor en uso es de tipo ladder, añade un recorrido vacío al final del esquema y lo numera.

Si el editor en uso es de tipo AWL este comando queda desactivado.

Comando Cortar

Realiza una operación de eliminación sobre el editor y copia en la memoria.

Para el PLC S5 100U, si el editor en uso es de tipo AWL, el comando sólo está activado si parte del texto está seleccionado. En estas condiciones, activar este comando conduce a la eliminación de la parte de texto seleccionada y a transportarla a los *Apuntes*. Luego, este texto podrá reinsertarse con el [Comando Pegar](#) en otro punto del mismo bloque, en otro bloque AWL del mismo proyecto, en otro bloque AWL de otro proyecto, o en otra aplicación de Windows (p.ej. *Bloque notas*).

Para el PLC PS3, la operación de eliminación se efectúa sobre la línea en uso, aunque no se haya hecho una selección. No se puede cortar más de una línea cada vez.

Por el contrario, si el editor en uso es de tipo ladder, el comando sólo está activado cuando en el esquema hay un recorrido seleccionado. La elección de este comando provoca la eliminación del recorrido del esquema y su copia en la memoria (no en los *Apuntes*). Luego, el recorrido cortado podrá pegarse en el mismo editor, o en un editor de un bloque distinto en el mismo proyecto (S5 100U) o en otro proyecto, pero no en otra aplicación de Windows.

Ver el [Comando Pegar](#) para más informaciones.

Si tan sólo desea borrar, sin copiar en la memoria todo cuanto consta en el bloque en uso, pulse la tecla Supr.

Teclado: Ctrl+X

Comando Copiar

Realiza una operación de copia en la memoria.

Para el PLC S5 100U, si el editor usado es de tipo AWL, el comando tan sólo está activado si está seleccionada una parte del texto. En estas condiciones, optar por este comando hace que la parte de texto seleccionada se copia en los *Apuntes*. Este texto podrá reinsertarse luego con el [Comando Pegar](#) en otro

punto del mismo bloque, en otro bloque AWL del mismo proyecto, en otro bloque AWL de otro proyecto, o en otra aplicación de Windows (ej. *Bloque notas*).

En cambio, para el PLC PS3, la operación de copia tiene por objeto la línea en uso, aunque no exista una selección previa. Puede copiarse una sola línea en cada ocasión.

Si el bloque en uso es de tipo ladder, el comando sólo está activado cuando hay un recorrido seleccionado en el esquema. La elección de este comando hace que el recorrido se copie en la memoria (no en los *Apuntes*). El recorrido copiado podrá pegarse en el mismo editor, en un editor de un bloque distinto en el mismo proyecto (S5 100U) o en otro proyecto, pero no en otra aplicación de Windows.

Teclado: Ctrl+C

Comando Pegar

Permite insertar en el editor en uso una parte de programa, cortada o copiada con anterioridad (ver [Comando Cortar](#) y [Comando Copiar](#) del mismo menú).

Para el PLC S5 100U, si el editor en uso es de tipo AWL, la operación de copiado hace que el texto se inserte donde está situado el cursor. El texto también puede haberse copiado o cortado de otra aplicación de Windows.

Para el PLC PS3, la operación hace que la instrucción copiada o cortada con anterioridad, se inserte en la posición vigente del cursor.

Si el bloque en uso es Ladder, la inserción del recorrido se produce en correspondencia con la selección o, si no se hay selección previa, se inserta al final del bloque.

Teclado: Ctrl+V

Comando Mostrar símbolos

Opte por este comando cuando tenga la intención de usar operandos simbólicos en la programación. Para más información sobre los tipos de operandos, ver el apartado [Operandos absolutos y símbolos](#). Una señal de "visto" junto al comando le indica que este tipo de visualización está activada.

Para usar los operandos simbólicos es indispensable que haya escrito atribuciones de símbolos o que las haya importado de otro proyecto. Sobre este tema, remítase a la información sobre la [Ventana Tabla de símbolos](#) o del [Comando Importar símbolos...](#) del [Menú Archivo](#).

Cuando programe este modo de visualización, se efectúa la conversión de los operandos en todos los editores abiertos y, en cada editor que abra, ya aparecerán los operandos simbólicos. Cada nuevo operando que vaya a insertar en un editor ladder o AWL se convertirá automáticamente en la forma activa. Es decir, si inserta un operando absoluto en la modalidad de visualización de símbolos, PC-Sim lo convertirá en el correspondiente símbolo .

Si no ha definido símbolos, la programación de esta modalidad no hará que su visualización se modifique: al no haber símbolos disponibles que sustituyan a los operandos absolutos, PC-Sim seguirá mostrándole estos últimos.

Si, por el contrario, existe una lista de atribuciones activa, pero no todos los operandos usados en el programa tienen símbolos atribuidos, PC-Sim sustituirá tan sólo los operandos absolutos para los cuales ha declarado un símbolo, dejando el resto como estaba .

Teclado: Ctrl+Q

Comando *Mostrar operandos absolutos*

Seleccione este comando cuando pretenda programar usando los operandos absolutos. Para más información sobre tipos de operandos, ver el apartado [Operandos absolutos y símbolos](#). Una señal de “visto” junto al comando le indica que este tipo de visualización está activada. La visualización preprogramada con PC-Sim es la que muestra los operandos absolutos, de manera que cuando arranque el programa esta será la visualización utilizada.

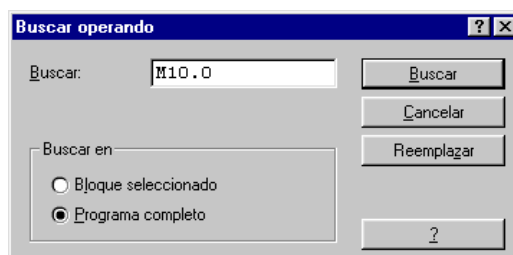
Cuando programe esta modalidad de visualización, PC-Sim efectúa la conversión de todos los símbolos presentes en todos los editores abiertos. Además, a cada editor que vaya abriendo, se irán visualizando los operandos absolutos. Cada nuevo operando que vaya a insertar en un editor ladder o AWL será convertido automáticamente en la forma activa. Es decir, en este caso, al ser activa la modalidad de visualización de los operandos absolutos, si escribe un símbolo, PC-Sim lo convertirá en el correspondiente operando absoluto.

Teclado: Ctrl+W

Comando *Buscar operando...*

Este comando le permite buscar todas las situaciones de un dato operando en el seno de su programa.

Como respuesta, PC-Sim muestra la ventana de diálogo *Buscar operando*. Inserte la sigla del operando que intenta localizar en la casilla *Buscar*. Puede insertar tanto operandos absolutos como símbolos, pero recuerde que a estos últimos hay que ponerles delante un guión (signo de menos).



Si está utilizando la simulación del PLC S5 100U, seleccione el área de búsqueda haciendo clic sobre el pulsador opción *Bloque seleccionado* o sobre el de *Todo el programa*. Haciendo clic sobre el primero, la búsqueda se efectuará sólo sobre el bloque seleccionado en ese momento en la [Tabla de bloques](#); haciendo clic sobre el segundo, la búsqueda se efectuará en todos los bloques del proyecto sin tener en cuenta el tipo de editor (AWL o ladder). Para las demás simulaciones de PLC, el pulsador *Bloque seleccionado* está desactivado y la búsqueda se hará dentro del único editor disponible.

Haga clic sobre el pulsador *Buscar* para iniciar la búsqueda a partir del inicio del área especificada, del inicio del bloque seleccionado, del inicio del primer bloque de la [Tabla de bloques](#) (S5 100U) o del inicio del

programa (FPC404 y PS3). Si el operando se localiza, PC-Sim mostrará el editor que lo contiene y activará la selección del nombre del operando. La ventana *Buscar operando* seguirá abierta y, a cada clic sobre *Buscar siguiente*, PC-Sim mostrará la siguiente localización. Un mensaje le avisará cuando la búsqueda haya llegado al final del área prevista. Otro mensaje le informará si se ha encontrado alguna ubicación del operando buscado.

Si es necesario, PC-Sim convierte automáticamente el operando insertado en la modalidad de visualización programada con el [Comando Mostrar símbolos](#) o el [Comando Mostrar operandos absolutos](#). Por ejemplo, si esta activada la modalidad de visualización de símbolos y para la búsqueda inserta un operando absoluto, éste será sustituido por el correspondiente símbolo, en caso de que haya uno definido.

Para finalizar la búsqueda y cerrar la ventana pulse *Cancelar*.

Para activar la sustitución del operando encontrado con un operando nuevo pulse *Sustituir*. Para más información sobre la sustitución, ver la descripción del [Comando Reemplazar operando...](#)

Para el PLC S5 100U, el comando *Buscar operando* está desactivado si en el proyecto no hay ningún bloque.

Para los PLC FPC 404 y PS3 el pulsador opción *Bloque seleccionado* está desactivado, puesto que el programa no puede estructurarse en bloques.

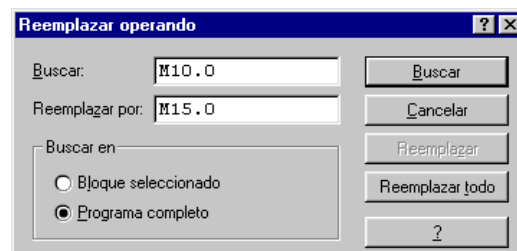
Teclado: Ctrl+F

Comando Reemplazar operando...

Se utiliza para la búsqueda de un determinado operando y su sustitución por otro determinado.

Como respuesta a esto comando, PC-Sim muestra la ventana de diálogo *Reemplazar operando*. Inserte el nombre del operando a buscar en la casilla *Buscar* y el nombre del operando sustituir en la casilla *Sustituir por*. Puede insertar tanto operandos absolutos como símbolos, pero recuerde poner un guión delante de estos últimos.

Si está utilizando la simulación del PLC S5 100U, seleccione el área de búsqueda haciendo clic sobre *Bloque seleccionado* o sobre *Todo el programa*. Haciendo clic sobre el primero, la sustitución se hará sólo sobre el bloque seleccionado en la [Tabla de bloques](#); En cambio, haciendo clic sobre el segundo, se realizará en todos los bloques del proyecto sin tener en cuenta el tipo de editor (AWL o ladder). Para las demás simulaciones de PLC, el pulsador opción *Bloque seleccionado* está desactivado.



Haga clic sobre el pulsador *Sustituir* para sustituir la primera ubicación del operando buscado. A cada clic sobre este pulsador PC-Sim realizará una única sustitución e iniciará la búsqueda de la siguiente ubicación.

Haga clic sobre el pulsador *Buscar siguiente* para saltar la sustitución del operando recién encontrado e iniciar la búsqueda de una nueva ubicación.

Haga clic sobre *Sustituir todo* para sustituir automáticamente todas las localizaciones del operando en el área de búsqueda especificada.

Si es necesario, PC-Sim convierte automáticamente los operandos insertados en la modalidad de visualización programada con el [Comando Mostrar símbolos](#) o el [Comando Mostrar operandos absolutos](#). Es decir, por ejemplo, si está activada la modalidad de visualización de símbolos e inserta un operando absoluto, éste será sustituido por el correspondiente símbolo, si hay uno definido.

Para terminar la sustitución y cerrar la ventana pulse *Cancelar*.

Para el PLC S5 100U el comando *Reemplazar operando* queda desactivado si en el proyecto no existe ningún bloque.

Para los PLC FPC 404 y PS3 el pulsador opción *Bloque seleccionado* queda desactivado dado que el programa no puede estar estructurado en bloques.

Teclado: Ctrl+H

Menú Ventana

Comando Código (FPC 404 y PS3)

Abre o lleva a primer plano el editor de programa. Si está activada la visualización del debug regresa a la condición de edición. Recuerde que tan sólo es posible editar un programa si la CPU se encuentra en estado de OFF o en el de STOP. Respecto a ello ver el apartado [Estado de la CPU](#) y los comandos del [Menú CPU](#).

En cambio, para el PLC S5 100U utilice el [Pulsador Código](#) en la [Ventana Lista bloques \(S5 100U\)](#) o el [Comando Abrir bloque \(S5 100U\)](#) del [Menú Archivo](#) tras haber seleccionado el bloque en la [Tabla de bloques](#).

Comando Debug (FPC 404 y PS3)

Lleva la visualización del editor de programa a debug. Este comando está desactivado si la CPU se encuentra en estado de OFF. Para devolver el editor a la visualización normal, utilice el [Comando Código \(FPC 404 y PS3\)](#) en el mismo menú.

Para más informaciones ver los apartados [Debug en las ventanas Ladder](#), [Debug en la ventana AWL](#), [Estado de la CPU](#) y los comandos del [Menú CPU](#).

Para el PLC S5 100U utilice el [Pulsador Debug](#) en la [Ventana Lista bloques \(S5 100U\)](#) una vez haya seleccionado el bloque en la [Tabla de bloques](#).

Comando Lista bloques (S5 100U)

Abre o lleva a primer plano la [Ventana Lista bloques \(S5 100U\)](#). Equivale a accionar el [Pulsador Ventana Lista bloques](#) en la [Barra instrumentos](#) de PC-Sim.

Teclado: Ctrl+B

Comando Tabla de símbolos

Abre o lleva a primer plano la [Ventana Tabla de símbolos](#). Equivale a accionar el *Pulsador Ventana Tabla de símbolos* en la [Barra instrumentos](#) de PC-Sim.

Teclado: Ctrl+A

Comando Estado variables

Abre o lleva a primer plano la [Ventana Estado variables](#). Equivale a accionar el *Pulsador Ventana Estado variables* en la [Barra instrumentos](#) de PC-Sim.

Comando Asignación de preselectores (FPC 404)

Abre o lleva a primer plano la [Ventana Asignación de preselectores \(FPC 404\)](#). Equivale a accionar el *pulsador Ventana Asignación de preselectores* en la [Barra instrumentos](#) de PC-Sim.

Comando Casilla instrumentos

Abre o lleva a primer plano la [Casilla instrumentos](#) para la edición del esquema ladder. Para el PLC PS3 este comando no está activado si el proyecto en uso es de tipo AWL. Equivale a accionar el *Pulsador Casilla instrumentos* en la [Barra instrumentos](#) de PC-Sim.

Teclado: F9

Comando PLC

Abre o lleva a primer plano la [Ventana PLC](#). Equivale a accionar el *Pulsador Ventana PLC* en la [Barra instrumentos](#) de PC-Sim.

Teclado: F11

Comando Informaciones sobre el proyecto

Abre o lleva a primer plano la [Ventana Informaciones sobre el proyecto](#). Equivale a accionar el *Pulsador Ventana Informaciones sobre el proyecto* en la [Barra instrumentos](#) de PC-Sim.

Teclado: F12

Comando Instalación en uso

Lleva a primer plano la ventana de la instalación abierta en ese momento. Si no hay ninguna instalación abierta, el comando está desactivado. Equivale a accionar el *Pulsador Instalación en uso* en la [Barra instrumentos](#) de PC-Sim.

Teclado: Ctrl+J

Menú CPU

Comando CPU Off

Accione este pulsador para simular la desconexión de la alimentación a la CPU. Cuando se corta la alimentación a la CPU se realizan las siguientes operaciones:

- se apagan los LED de estado de la CPU;
- se desactivan las salidas y se apagan los LED correspondientes.

La CPU del PLC S5 100U tiene un alimentador, o por lo menos un interruptor, distinto del de los periféricos. De esta manera, cortando la tensión a la CPU, estos últimos siguen recibiendo alimentación: por eso los LED de entrada se mantienen encendidos cuando los contactos conectados a ellos están cerrados.

Los demás PLC tienen un único interruptor de alimentación para la CPU y para la periferia por lo cual, cortando la alimentación a la CPU, los LED indicadores del estado de las entradas se apagan.

Comando CPU On

Accione este comando para proporcionar alimentación a la CPU.

Las operaciones efectuadas por el PLC virtual son las siguientes:

FPC 404

- encendido del LED Stop de la CPU;

y, si el pulsador de Run sobre la [Barra instrumentos](#) de PC-Sim ya está pulsado:

- compilación del programa en código máquina;

y, si la compilación se produce correctamente:

- borrado de los temporizadores, los contadores y los flags;
- encendido del LED Run de la CPU;
- apagado del LED Stop;
- elaboración cíclica del programa del usuario.

PS3

- encendido del LED Power de la CPU;

y, si el pulsador de Run sobre la [Barra instrumentos](#) de PC-Sim ya está accionado:

- compilación del programa en código máquina;

y, si la compilación se produce correctamente:

- borrado de las imágenes de proceso y de los merker no remanentes, inicialización de los módulos de sistema;
- encendido del LED Run de la CPU;
- lectura de la IPI (ver [Imágenes del proceso](#) en el [Capítulo 12](#));
- elaboración del programa del usuario;

- emisión de la IPU (ver *Imágenes del proceso* en el *Capítulo 12*).

Los tres últimos pasos se repiten hasta el siguiente paso a STOP o a OFF y constituyen la elaboración cíclica.

S5 100U

- encendido del LED Stop de la CPU;

y, si falta la batería:

- encendido del LED Battery Off/Low;

y, si el pulsador de Run sobre la *Barra instrumentos* de PC-Sim ya está accionado:

- compilación del programa en código máquina;

y, si la compilación se produce correctamente:

- borrado de las imágenes de proceso, de los temporizadores, los contadores no remanentes y los merker no remanentes;
- encendido del LED Run de la CPU;
- elaboración bloque de puesta en marcha OB22 (ver *Bloques Organizativos (OB)* en el *Capítulo 14*);
- apagado del LED Stop;
- lectura de la IPI (ver *Imágenes del proceso* en el *Capítulo 14*);
- elaboración del OB1 (ver *Bloques Organizativos (OB)* en el *Capítulo 14*);
- emisión de la IPU (ver *Imágenes del proceso* en el *Capítulo 14*).

Los tres últimos pasos se repiten hasta el siguiente paso a STOP o a OFF y constituyen la elaboración cíclica.

Comando CPU Stop

Con la CPU alimentada y en estado de RUN:

- se apaga el LED Run
- se enciende el LED Stop (FPC 404 y S5 100U);
- se desactivan las salidas y se apagan los LED correspondientes:
- se termina la elaboración del programa del usuario.

Comando CPU Run

Si la CPU no recibe alimentación (pulsador CPU Off accionado) la selección de este comando no produce ningún efecto. En cambio, si la CPU está alimentada (pulsador CPU On accionado) se verifican los siguientes pasos:

FPC 404

- compilación del programa en código máquina;
- y, si la compilación se produce correctamente:

- borrado de los temporizadores, los contadores y los flags;
- encendido del LED Run;
- apagado del LED Stop;
- elaboración cíclica del programa del usuario.

PS3

- compilación del programa en código máquina;
- y, si la compilación se produce correctamente:
- borrado de las imágenes del proceso e inicialización de los módulos de sistema;
 - encendido del LED de Run de la CPU;
 - lectura de la IPI (ver [Imágenes del proceso](#) en el [Capítulo 12](#));
 - elaboración del programa del usuario;
 - emisión de la IPU (ver [Imágenes del proceso](#) en el [Capítulo 12](#)).

Los tres últimos pasos se repiten hasta el siguiente paso a STOP o a OFF y constituyen la elaboración cíclica.

S5 100U

- compilación del programa en el código de la máquina del PLC;
- y, si la compilación se produce correctamente:
- borrado de las imágenes del proceso, de los temporizadores, los contadores no remanentes y de los merker no remanentes;
 - encendido del LED Run de la CPU;
 - elaboración bloque de puesta en marcha OB21 (ver [Bloques Organizativos \(OB\)](#) en el [Capítulo 14](#));
 - apagado del LED Stop;
 - lectura de la IPI (ver [Imágenes del proceso](#) en el [Capítulo 14](#));
 - elaboración del OB1 (ver [Bloques Organizativos \(OB\)](#) en el [Capítulo 14](#));
 - emisión de la IPU (ver [Imágenes del proceso](#) en el [Capítulo 14](#)).

Los tres últimos pasos se repiten hasta el siguiente paso a STOP o a OFF y constituyen la elaboración cíclica.

Comando Batería en servicio

Haga clic sobre este comando para simular la inserción de una nueva batería tampón en la CPU del PLC. Cuando la batería está insertada, las variables residentes en la memoria RAM mantenida por la batería retienen su valor incluso en caso de falta de alimentación al PLC simulado.

Para el PLC S5 100U, cuando la batería vuelve a funcionar deja de elaborarse el bloque OB34 al inicio de cada ciclo (ver [Bloques Organizativos \(OB\)](#) en el [Capítulo 14](#)).

Comando Batería fuera de servicio

Este comando sirve para simular la extracción o el agotamiento de la batería tampón. Cuando la batería no está colocada, las variables residentes en la memoria RAM se ponen a cero en caso de falta de alimentación.

Para el PLC S5 100U, en estas condiciones y antes del inicio de cada ciclo, la CPU del PLC simulado elaborará el bloque OB34, si está programado, hasta al reinsertión de la batería tampón (ver [Bloques Organizativos \(OB\)](#) en el [Capítulo 14](#)).

Menú Instalaciones

Por medio de este menú pueden ponerse en marcha las instalaciones de PL-Sim. Cada una de las opciones representa y llama a una instalación específica. PC-Sim controla la puesta en marcha de las instalaciones presentes en su PC y, para todos aquellos previstos y no encontrados, se encarga de desactivar el correspondiente comando.

Además, desde este menú pueden arrancarse las interfaces hacia las instalaciones reales (PL-RIS y PL-Net) y hacia aquellos de fabricación propia (PL-Lab).

Para poner en marcha una instalación basta con seleccionar la opción que identifica el nombre y su número de serie. Cuando se pone en marcha una instalación, el menú se desactiva para evitar la ejecución simultánea de varias instalaciones. Volverá a activarse al cierre de la instalación. Dicha operación se realiza desde la ventana correspondiente.

Comando Electroneumática 1

Pone en marcha la instalación n.1, Electroneumática 1, que simula un banco de electroneumática compuesto por tres cilindros neumáticos de doble efecto y tres distribuidores biestables con control eléctrico.

Para ampliar informaciones ver [Instalación n.1: Electroneumática 1](#) en el [Capítulo 4](#).

Comando Electroneumática 2

Pone en marcha la instalación n.2, Electroneumática 2, que simula un banco de electroneumática compuesto por tres cilindros neumáticos de doble efecto y tres distribuidores monoestables con control eléctrico.

Para ampliar informaciones ver [Instalación n.2: Electroneumática 2](#) en el [Capítulo 4](#).

Comando Electroneumática 3

Pone en marcha la instalación n.3, Electroneumática 3, que simula un banco de electroneumática compuesto por tres cilindros neumáticos de simple efecto y tres distribuidores monoestables con control eléctrico.

Para ampliar información ver [Instalación n.3: Electroneumática 3](#) en el [Capítulo 3](#).

Comando Automatización puerta

Pone en marcha la instalación n.4, Automatización puerta, que simula una clásica aplicación de la automatización en el campo civil, la automatización de una puerta eléctrica. Dispone de final de carrera, célula fotoeléctrica, mando a distancia, motor y indicador luminoso.

Para más informaciones ver [Instalación n.4: Automatización puerta](#) en el [Capítulo 4](#).

Comando Control nivel depósito

Pone en marcha la instalación n.5, Control nivel depósito, que representa una clásica aplicación de la automatización industrial. Dispone de los controladores de nivel de funcionamiento y de alarmas, de las válvulas y de un panel para la gestión de las alarmas.

Para ampliar información ver en el [Instalación n.5: Control del nivel de un depósito](#) en el [Capítulo 4](#).

Comando Cintas transportadoras

Pone en marcha la instalación n.6, Cintas transportadoras, que simula una instalación de distribución de cajas con cintas transportadoras. Presenta sensores, células fotoeléctricas, finales de carrera, motores y un panel para el control de la instalación y la gestión de las alarmas.

Para más información [Instalación n.6: Cintas transportadoras](#) ver en el [Capítulo 3](#).

Comando Ascensor

Pone en marcha la instalación n.7, Ascensor, otra clásica aplicación de la automatización en instalaciones civiles. Se trata de una instalación de ascensor para cuatro pisos con finales de carrera, botones de llamada y de retorno, luces indicadoras, célula fotoeléctrica, motor.

Para más informaciones ver [Instalación n.7: Ascensor](#) en el [Capítulo 3](#).

Comando Mezclador de líquidos

Pone en marcha la instalación n.8, Mezclador de líquidos, que permite elaborar ejercicios de automatización de productos que tienen, como componentes, tres líquidos distintos. Forman parte de esta instalación: los controladores de estado, termostato, válvulas y electrobombas, sondas de nivel y temperatura, además de un panel operativo para el control del ciclo de funcionamiento y las alarmas.

Para más informaciones ver [Instalación n.8: Mezclado de líquido](#) en el [Capítulo 4](#).

Comando Cruce semaforizado

Pone en marcha la instalación n.9, Cruce semaforizado, que permite elaborar ejercicios sobre la automatización de una instalación semafórica con modos de funcionamiento automático, manual y nocturno.

Para más informaciones ver [Instalación n.9: Cruce](#) en el [Capítulo 4](#).

Comando PL-RIS

Pone en marcha la interface hacia las instalaciones reales PL-RIS.

PL-RIS es el módulo de AW-SYS que permite conectar los PLC simulados de PC-Sim a las instalaciones reales o bancos de prácticas presentes en su taller.

Para más informaciones ver el [Capítulo 6](#).

Comando PL-Lab

Pone en marcha la interface hacia las instalaciones de fabricación propia PL-Lab.

PL-Lab permite la conexión entre los PLC simulados de PC-Sim y de InTouch, el conocido paquete de supervisión de la Wonderware, mediante el cual podrá construir por su cuenta nuevas ventanas simulación de instalaciones.

Para más informaciones ver el [Capítulo 7](#).

Comando PL-Net

Pone en marcha la interface hacia los sistemas MPS PL-Net.

PL-Net es el módulo de AW-SYS que permite el control de una instalación real que tenga un elevado número de entradas/salidas, por medio de más PLC simulados que se ejecutan sobre otros tantos ordenadores conectados en red.

Para más informaciones ver el [Capítulo 8](#).

Menú ?

Comando Guía

Abre la guía de PC-Sim.

La guía de PC-Sim tiene las funciones propias de las guías de Windows: es sensible al contexto y posee las funciones de desplazamiento adelante-atrás, resumen, cronología, búsqueda. Además, puede imprimirse un tema, copiar para después pegar, definir un punto de libro.

La guía está formada por dos ventanas. La ventana principal muestra el resumen de temas y la ventana secundaria muestra los temas concretos que se han llamado, en cada ocasión, desde la primera.

Aunque PC-Sim ha sido concebido para que sea fácil de usar y para una comprensión intuitiva e inmediata, utilice la guía ampliamente: le será muy útil al principio de sus prácticas.

Teclado: F1

Comando Autoware en Internet

Llama la guía on line en la opción que facilita informaciones sobre la página WEB de Autoware.

Comando Servicio Soporte Técnico Usuarios AW-SYS

Llama la guía on line en la opción que facilita informaciones sobre las modalidades de asistencia a los usuarios de AW-SYS.

Comando *Informaciones sobre PC-Sim*

Muestra la ventana de las informaciones sobre el programa PC-Sim.

Desde esta ventana podrá obtener informaciones acerca de la versión del programa que usted posee, el tipo de PLC simulado y sobre el productor del software.

La ventana de las informaciones se muestra también cada vez que se inicia PC-Sim.

Para cerrar la ventana accione el pulsador Ok.

Barra del título























La barra del título de la ventana principal de PC-Sim contiene, además del nombre del programa que especifica también el modelo de PLC en uso, el nombre y la extensión del archivo del proyecto abierto.

Si se acaba de abrir PC-Sim, o si se ha iniciado un nuevo proyecto y aun no se ha guardado, es decir, en todos los casos en los que el proyecto corriente no tienen aun un nombre, en la barra del título, en el lugar del nombre del proyecto se mostrará el texto “[Sin nombre]”.

Barra instrumentos

Los pulsadores de la barra de instrumentos de PC-Sim le permiten realizar algunas acciones, vistas ya por los comandos de los menús, pero de una forma más rápida. En la barra se han reagrupado aquellos comandos que se reclaman más a menudo al usar PC-Sim. Para más explicaciones al respecto, el lector puede remitirse a todo lo ya dicho en los comandos equivalentes de los menús.

Tabla 2: Lista pulsadores de la barra instrumentos de PC-Sim y comandos equivalentes

Pulsador	Descripción	Comando equivalente	Menú	FPC	PS3	S5
	Nuevo proyecto	Nuevo proyecto	Archivo			✓
	Nuevo proyecto ladder	Nuevo proyecto ladder	Archivo	✓	✓	
	Nuevo proyecto AWL	Nuevo proyecto AWL	Archivo		✓	
	Nuevo bloque	Nuevo bloque	Archivo			✓
	Abrir proyecto	Abrir proyecto	Archivo	✓	✓	✓
	Guardar proyecto	Guardar proyecto	Archivo	✓	✓	✓
	Código	Código	Ventana	✓	✓	
	Ventana Lista bloques	Lista bloques	Ventana			✓
	Ventana Tabla de símbolos	Tabla de símbolos	Ventana	✓	✓	✓
	Ventana Estado variables	Estado variables	Ventana	✓	✓	✓
	Ventana Programaciones de preselectores	Programaciones preselectores	Ventana	✓		
	Ventana PLC	PLC	Ventana	✓	✓	✓
	Ventana Informaciones sobre el proyecto	Informaciones sobre el proyecto	Ventana	✓	✓	✓
	Casilla instrumentos	Casilla instrumentos	Ventana	✓	✓	✓
	Buscar operando	Buscar operando	Modificar	✓	✓	✓
	Instalación en uso	Instalación en uso	Ventana	✓	✓	✓
	CPU Off	CPU Off	CPU	✓	✓	✓
	CPU On	CPU On	CPU	✓	✓	✓
	CPU Stop	CPU Stop	CPU	✓	✓	✓
	CPU Run	CPU Run	CPU	✓	✓	✓
	Batería en servicio	Batería en servicio CPU	CPU	✓	✓	✓
	Batería fuera servicio	Batería fuera servicio	CPU	✓	✓	✓

En la [Tabla 2](#) se incluye el listado de los pulsadores, del nombre del comando equivalente y del menú al que pertenecen, así como el modelo de PLC simulado para el que están disponibles.

Accesos rápidos con el teclado

En todo el PC-Sim están activados distintos accesos rápidos con el teclado que permiten llamar algunos comandos, evitando el uso de los menús. Para activar el comando basta con accionar la tecla o la combinación de teclas especificada.

En [Tabla 3](#) facilitamos el listado de las teclas para su activación.

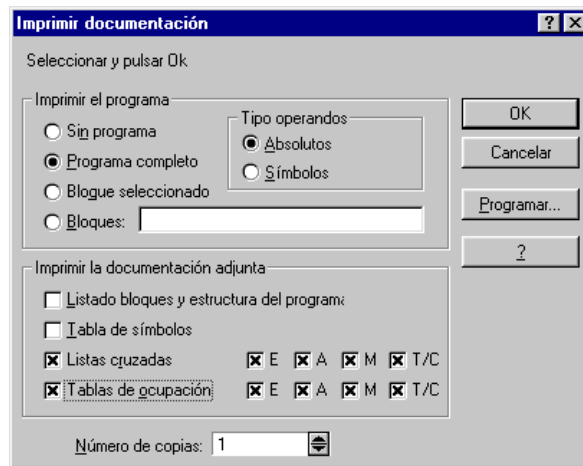
Tabla 3: Listado de los accesos rápidos con el teclado de PC-Sim

Tecla	Función
Ctrl+A	Llamar la ventana <i>Tabla de símbolos</i>
Ctrl+B	Llamar la ventana <i>Lista bloques</i>
Ctrl+C	Copiar selección
Ctrl+F	Poner en marcha la búsqueda de operandos
Ctrl+H	Poner en marcha la sustitución de operandos
Ctrl+J	Llamar la instalación en uso
Ctrl+N	Nuevo bloque
Ctrl+S	Guardar proyecto
Ctrl+V	Pegar selección
Ctrl+X	Cortar selección
Alt+F4	Cerrar la ventana en uso o finalizar la sesión de PC-Sim
Supr	Suprimir la selección
F1	Activar la guía
F2	Edición de celda en una tabla
F5	CPU Off
F6	CPU On
F7	CPU Stop
F8	CPU Run
F9	Llamar la <i>Casilla instrumentos</i>
F11	Llamar la ventana <i>PLC</i>

Impresión de la documentación

Cuando se selecciona el [Comando Imprimir documentación...](#) PC-Sim responde mostrando en pantalla la ventana de diálogo *Imprimir documentación* que permite escoger las partes del programa a imprimir y las modalidades de impresión.

En el presente apartado examinamos con detalle estas opciones.



Impresión del programa

Este recuadro permite la elección de las partes del programa que deberán imprimirse y en qué modalidad. Haga clic sobre los pulsadores de opción en base a las siguientes descripciones.

Sin programa

No se imprimirá el código de ningún bloque presente en el proyecto (S5 100U) o no se imprimirá el programa (FPC 404 y PS3).

Todo el programa

Se imprimirá el código de todos los bloques presentes en el proyecto, ya sea ladder, AWL o de datos (S5 100U) o se imprimirá el programa (FPC 404 y PS3).

Bloque seleccionado (S5 100U)

Se imprimirá solamente el código correspondiente al bloque seleccionado en ese momento en la [Tabla de bloques](#).

Bloques: (S5 100U)

Permite insertar, en la casilla situada al lado, el listado de los bloques a imprimir. Las siglas de los bloques deben ir separadas por comas y también están permitidos los caracteres especiales '-' y '*'. A continuación detallamos algunos ejemplos que aclararán el uso de la casilla.

PB45: sólo se imprime el código del bloque **PB45**.

OB1 , **OB34** , **PB45**: se imprimen los códigos de los tres bloques mencionados.

FB10–FB14: se imprime el código de los bloques, de **FB10** a **FB14**, es decir, **FB10**, **FB11**, **FB12**, **FB13** y **FB14**.

OB*: se imprime el código de todos los bloques organizativos (OB) presentes en el programa.

Sin embargo, también pueden realizarse inserciones múltiples, p. ej. **OB*** , **PB12–PB14** , **DB35** que permiten imprimir todos los **OB**, los **PB 12**, **13**, **14** y el **DB35**.

Tipo operandos

Permite elegir, con independencia de la modalidad de visualización en vigor, el tipo de operandos a imprimir entre absolutos y símbolos (ver [Operandos absolutos y símbolos](#) en este capítulo).

Impresión de la documentación añadida

Además del listado del programa, PC-Sim permite imprimir elementos posteriores del proyecto en uso. En este recuadro pueden seleccionarse las partes que se añadirán a la impresión del programa, en función de las clarificaciones que se hacen a continuación.

Listado bloques y estructura programa (S5 100U)

La activación de esta casilla de control permite imprimir la lista de los bloques presentes en el programa, que contiene las columnas *Bloque*, *Tipo* y *Comentario*, tal como aparecen en la [Tabla de bloques](#).

Esta también permite imprimir la estructura del programa, es decir, el árbol de las llamadas a los bloques. Los árboles tienen como raíz (bloque generante) los bloques llamados de sistema operativo, es decir, **OB1**, **OB21**, **OB22** y **OB34**. En la impresión sólo estarán presentes los árboles relativos a los bloques raíz programados. Por ejemplo, si no ha programado el **OB34** para que reaccione ante la descarga o la extracción de la batería tampón, el correspondiente árbol no se imprimirá, ya que no existe. Para más informaciones respecto a los bloques llamados de sistema operativo consulte [Bloques Organizativos \(OB\)](#) en el [Capítulo 14](#).

Tabla de símbolos

Permite la impresión de la lista de las atribuciones en uso tal como está definida en la [Ventana Tabla de símbolos](#), organizada en tabla con columnas para el operando absoluto, el simbólico y el comentario.

Listas cruzadas

Con la activación de esta casilla de control se obtendrá la impresión de las listas cruzadas de los operandos; es decir, se imprimirán todos los operandos encontrados en el programa. En correspondencia con cada uno de ellos, se detallarán los bloques en los que se han encontrado (S5 100U) y, para los bloques Ladder, el número de recorrido, o el número de línea para los bloques AWL, que contienen el operando considerado. La lista cruzada es de gran importancia en la fase de depuración (debug) del programa porque, tal como acaba de explicarse, le permite conocer las posiciones de cada uno de los operandos en el seno del programa.

Además, la casilla de control *Listas cruzadas*, una vez activada, le permite decidir con qué operandos desea ejecutar e imprimir la búsqueda. Podrá habilitarla para las entradas, las salidas, los merker o flag, los temporizadores y contadores (FPC 404 y S5 100) o para los módulos de sistema (PS3).

Las listas cruzadas para las entradas, las salidas y los merker para los PLC PS3 y S5 100U proporcionan informaciones no sólo para los operandos específicos direccionados a bit, sino también para los mismos tipos de operandos direccionados a byte y a word.

Tablas de ocupación

La activación de esta casilla de control de la ventana *Imprimir documentación* permite la impresión de tablas en las que cada celda identifica a un operando específico. Las tablas están organizadas por byte o por word según el PLC: cada fila corresponde a un byte o word de entrada, de salida o de memoria merker (flag). Las columnas identifican en el interior de cada byte o word un bit específico, que corresponde al número de la columna. Allí donde es posible, otras columnas identifican todo el byte y toda la word.

Las celdas correspondientes a los operandos utilizados en el programa se oscurecen. Un vistazo rápido a esta impresión le permitirá saber si ya ha utilizado un cierto operando en el programa o bien localizar, por ejemplo, un merker libre a utilizar.

Es decir, si por ejemplo en el programa para S5 100U está direccionada la entrada **E1.4**, la celda que corresponde a la fila **EB1** y a la columna 4 se oscurecerá; al igual que la celda correspondiente a la fila **AB2** y la columna B, si en el programa ha utilizado el byte de salida **AB2**. Aun más, si en el programa ha utilizado la **MW40**, las celdas correspondientes a las filas **MB40** y **MB41** y a la columna W se oscurecerán.

También en este caso pueden activarse individualmente la impresión de las tablas correspondientes a las entradas, salidas, merker (flag), temporizadores y contadores (FPC 404 y S5 100U) o a los módulos de sistema (PS3).

Además, para el PLC S5 100U, podrán obtenerse informaciones acerca de superposiciones en la utilización de las word (particularmente interesantes son aquellos para las merker word) entre direcciones pares e impares. Como ejemplo, considere el caso de haber utilizado en su programa la **MW20** y la **MW21**, de manera que el byte 21 está repartido entre las dos word (ver *Direccionamiento* en el *Capítulo 14* para más informaciones al respecto). Si no se ha hecho a propósito, esto podría provocar efectos no deseados. El diferente tipo de gráfica de llenado elegido para las merker word pares y las impares le permitirá localizar rápidamente la superposición. De hecho, la celda correspondiente al **MB21** se mostrará más oscura.

Mensajes de aviso

En este apartado incluimos el listado de algunos de los mensajes con los que PC-Sim responde a acciones o situaciones anómalas. Tan sólo se incluyen aquellos para los cuales es necesaria alguna aclaración. Para cada uno de ellos se describen los procedimientos para resolver o evitar el problema.

Código de hardware ausente o erróneo

El código de hardware que se facilita con el PC-Sim no se ha colocado en el puerto paralelo de su ordenador. Si su ordenador está dotado de más de un puerto paralelo inserte el código llave en uno cualquiera de estos: PC-Sim es capaz de buscar el código en cualquier puerto paralelo.

O bien, Windows no reconoce el puerto paralelo en la que ha insertado el código: verifique la programación del sistema desde el panel de control.

El código de hardware es para una versión anterior de PC-Sim

Ha insertado en el puerto paralelo de su PC un código de hardware para la versión 1 de PC-Sim. Contacte con *Autoware srl – servicio comercial* para informaciones sobre las modalidades de actualización de AW-SYS.

Una instancia de PC-Sim o PC-RIS está ya en ejecución! No se puede poner en marcha a la vez (otro) PC-Sim

Las copias de PC-Sim sólo pueden ejecutarse de una en una: no se podrá poner en marcha una segunda instancia del programa, aunque corresponda a un PLC distinto. La puesta en marcha de PC-Sim también es imposible cuando PC-RIS ya está en ejecución. Cierre la instancia de PC-Sim o PC-RIS en ejecución y reinténtelo.

AW-SYS podría no funcionar correctamente con esta modalidad de vídeo. Es aconsejable programar 'caracteres pequeños' en las propiedades de la pantalla.

La utilización de una modalidad de vídeo 'caracteres grandes' o 'caracteres muy grandes' podría provocar irregularidades en la visualización de las imágenes. Cierre PC-Sim y programe una modalidad 'caracteres pequeños' en la ventana de diálogo *Propiedades de Pantalla* que puede abrirse desde el *Panel de control* de Windows.

El archivo [nombre proyecto] es para un modelo distinto de PLC. Imposible abrir el proyecto.

El archivo proyecto especificado en la ventana de diálogo *Abrir proyecto* se ha realizado con la versión de PC-Sim destinada a la simulación de otro PLC. Cierre la sesión en uso de PC-Sim, lance el simulador del otro PLC y reintente abrir el proyecto.

El archivo [nombre proyecto] no es un proyecto de PC-Sim.

El archivo especificado en la ventana de diálogo *Abrir proyecto* no es un proyecto de PC-Sim. Es imposible cargar el archivo.

No es posible localizar el archivo proyecto [nombre archivo].

El nombre del proyecto insertado en la ventana de diálogo *Abrir proyecto*, o seleccionado directamente en el *Menú Archivo*, no es válido o bien no existe ningún proyecto con ese nombre. Corregir el nombre del archivo y reintentar.

No es posible localizar la ruta [nombre ruta].

El nombre de la ruta del archivo insertado en la ventana de diálogo *Abrir proyecto*, o seleccionado directamente en el *Menú Archivo*, no es válido o bien no existe ningún proyecto con ese nombre. Corregir el nombre del archivo y reintentar.

Error al guardar. El archivo podría ser sólo de lectura.

Ha intentado sobrescribir un archivo con atributo de sólo lectura. Guárdelo con otro nombre o bien desactive el atributo 'Sólo lectura' del archivo utilizando la ficha *Propiedades*.

Error al abrir el proyecto.

Durante la apertura del proyecto se ha producido un error indeterminado. Contacte con el [Servicio de Soporte Técnico para Usuarios de AW-SYS](#).

¡El archivo proyecto ya existe! ¿Sobreescribir?

Tras haber puesto en marcha el [Comando Guardar proyecto como...](#) en el *Menú Archivo*, se está pidiendo que el proyecto en uso se guarde con un nombre de archivo ya existente.

Responder:

- Si si se desea sobrescribir el proyecto preexistente;
- No si se desea cambiar el nombre del archivo a registrar y por tanto no sobrescribir el antiguo proyecto;
- Cancelar si se opta por cancelar el registro del proyecto abierto.

El proyecto en uso no se ha guardado. ¿Guardar los cambios antes de cerrarlo?

Ha pedido cerrar el proyecto en uso, con uno de los comandos *Nuevo proyecto*, *Abrir proyecto* o *Salir* del *Menú Archivo* o con una de las acciones equivalentes, pero este contiene cambios que aun no se han guardado. PC-Sim le invita a guardar los cambios antes de abandonar el proyecto.

Responder:

- Si si se desea guardar el proyecto;
- No si no se desea guardar el proyecto, perdiendo por tanto las últimas modificaciones realizadas en él;
- Cancelar si se opta por cancelar la operación de cierre del proyecto en uso.

El proyecto en uso se ha guardado en un formato antiguo. ¿Actualizarlo al formato actual?

Ha pedido cerrar el proyecto en uso, con uno de los comandos *Nuevo proyecto*, *Abrir proyecto* o *Salir* del *Menú Archivo* o con una de las acciones equivalentes, pero éste se ha realizado con una versión anterior de PC-Sim. Se le pregunta si desea actualizar el archivo proyecto al nuevo formato.

Responder:

- Si si se desea actualizar el proyecto al nuevo formato;
- No si se desea dejar el archivo proyecto en el formato antiguo;
- Cancelar si se pretende cancelar la operación de cierre del proyecto en uso.

La (nueva) sigla del bloque no es válida.

Al crear o renombrar un bloque se ha insertado, en la casilla *Bloque*, una sigla no válida. Las siglas de bloque válidas para PC-Sim para S5 100U van de **OB0** a **OB63** para los bloques organizativos, de **PB0** a **PB63** para los bloques de programa, de **FB0** a **FB63** para los bloques funcionales y de **DB2** a **DB63** para los bloques de datos. Corregir la sigla y reintentar.

El bloque ya está presente en el proyecto!

Al crear un bloque, o al renombrar uno ya existente, se ha usado la sigla de un bloque ya presente en la [Tabla de bloques](#). Eso no es posible: la sigla de cada bloque debe ser unívoca. Asignar un nombre unívoco al bloque que se está creando o renombrarlo.

Se ha alcanzado el número máximo de bloques permitido en un proyecto

Es imposible crear un bloque más en el proyecto en vigor porque los presentes son ya el máximo número que PC-Sim para S5 100U puede gestionar: 100.

Se ha alcanzado la dimensión máxima permitida del {bloque | programa}. Imposible ejecutar el comando.

Una operación iniciada, generación de un ramal paralelo, introducción de un temporizador, introducción de un contador etc. Llevaría a superar la dimensión máxima prevista por el editor ladder de 999 ramales. Para S5 100U, divide el bloque en dos conectando el segundo al primero con una llamada incondicionada como última instrucción. Para los demás PLC no tiene posibilidad de intervenir.

El programa {AWL | ladder} ha sido modificado. ¿Guardar los cambios actuales?

Está en uso la simulación para FPC 404 o PS3 y se está cerrando un editor de programa cuyo contenido no se ha guardado, o cuyos últimos cambios no se han archivado.

Responder:

- Si si se desea guardar el código en la memoria de programa.
- No si no se desea registrar el código, perdiendo así las modificaciones posteriores a la última vez que se archivó.
- Cancelar si se pretende cancelar la operación, es decir, el cierre del editor de programa.

El bloque [sigla bloque] ha sido modificado. ¿Guardar los cambios actuales?

Está en uso la simulación de S5 100U y se está cerrando un editor de bloque cuyo contenido no se ha guardado o cuyas últimas modificaciones no se han archivado.

Responder:

- Si si se desea guardar el bloque en la memoria de programa.
- No si no se desea guardar el bloque, perdiendo así los cambios posteriores a la última vez que se archivó. Si el bloque no se ha guardado nunca desde su creación, se perderá completamente y su sigla se eliminará de la [Tabla de bloques](#).
- Cancelar si se pretende cancelar la operación, es decir, el cierre del editor de bloque.

Eliminar el bloque [sigla bloque]?

Ha solicitado eliminar un bloque del proyecto en uso. El mensaje le pide confirmación por seguridad. Una vez eliminado un bloque, ya no será posible recuperarlo.

¡Atención! El programa contiene llamadas al bloque a eliminar. ¿Eliminar de todos modos?

Después de haber confirmado la eliminación de un bloque, PC-Sim controla que en el resto del programa no hayan operaciones de llamada (SPA o SPB) al bloque a eliminar. Si se encuentran llamadas, este mensaje le avisará pidiéndole de nuevo que confirme la eliminación.

De hecho, las llamadas a un bloque no presente, si bien no provocarán un error en la compilación y en la ejecución del programa, sin duda constituyen una anomalía de programación. Para encontrar rápidamente

todas las llamadas al bloque en el seno del programa, utilice el *Comando Buscar operando...* del *Menú Modificar*, tecleando la sigla del bloque eliminado en la casilla *Buscar*.

Imposible renombrar un bloque abierto. Cerrar el bloque y reintentar.

Se ha pedido renombrar un bloque en Edit o en Debug, tal como se indica en la *Columna Estado* de la *Tabla de bloques*. La operación es imposible. Cierre el bloque y reintente renombrar.

¿Actualizar también todas las llamadas al bloque con la nueva sigla?

Si se ha renombrado un bloque cambiando su sigla. PC-Sim le pregunta si desea que actualice automáticamente las posibles llamadas (SPA o SPB) al bloque renombrado, sustituyendo en éste la nueva sigla del bloque.

Si no actualiza las llamadas automáticamente (respondiendo afirmativamente a la pregunta) o manualmente (respondiendo negativamente), en el momento de la compilación, cuando el PLC pasa a Run, PC-Sim podría señalarle la presencia de llamadas a un bloque inexistente, si el bloque renombrado era un bloque funcional.

Responder:

- Si si se desea actualizar automáticamente todas las llamadas al bloque que tengan el nombre antiguo presentes en el programa
- No si se desea dejar inalteradas las llamadas al bloque con el antiguo nombre

En la atribución falta el operando absoluto!

Durante la fase de input de una nueva atribución en la *Tabla de símbolos* se ha omitido cumplimentar el campo correspondiente al operando absoluto, o bien durante la modificación de una atribución existente se ha borrado ese mismo campo. Insertar un operando absoluto válido en el campo.

Operando absoluto incorrecto!

Durante la fase de input de una atribución se ha escrito un nombre de operando no válido en el campo operando absoluto de la *Tabla de símbolos*. Corregir el operando.

La lista completa de los operandos válidos se incluye en los párrafos *Direccionamiento*, que aparece en los capítulos dedicados a las notas de programación de los distintos PLC simulados.

En la atribución falta el símbolo!

Durante la fase de input de una nueva atribución en la *Tabla de símbolos* se ha omitido cumplimentar el campo correspondiente al símbolo, o bien durante la modificación de una atribución existente se ha borrado ese mismo campo. Insertar el nombre de un símbolo válido en la casilla.

Símbolo incorrecto!

Durante la fase de input o de modificación de una atribución en la *Tabla de símbolos* se ha escrito un operando simbólico no válido en el campo *Símbolo*. Corregir el símbolo.

Consulte el apartado [Operandos absolutos y símbolos](#) en este capítulo para informaciones sobre la sintaxis de los operandos simbólicos.

Operando absoluto ya presente!

El operando absoluto que se está utilizando en el fila en uso de la [Tabla de símbolos](#) ya se ha utilizado en otra atribución. PC-Sim no permite la atribución de más símbolos a un único operando absoluto. Desista de la atribución o utilice otro operando absoluto.

Símbolo ya presente!

El símbolo que se está utilizando en el fila en uso de la [Tabla de símbolos](#) ya se ha utilizado en otra atribución. PC-Sim no permite atribuir un único símbolo a varios operandos absolutos. Insertar un símbolo distinto.

Atribución ya presente!

Toda la atribución, es decir, la pareja operando absoluto y símbolo, está ya presente en la [Tabla de símbolos](#) y por tanto, ya es válida en el interior del proyecto. Cancele la inserción.

El operando insertado para el estado no es válido!

Durante la fase de input en el campo *Op.absoluto* de la [Tabla variables](#) se ha escrito la sigla de un operando absoluto o simbólico no válido. Corregir el operando.

La lista completa de los operandos absolutos válidos se incluye en los párrafos *Direccionamiento*, que aparece en los capítulos dedicados a las notas de programación de los distintos PLC simulados.

Las reglas sintácticas para la definición de los símbolos se detallan en el apartado [Operandos absolutos y símbolos](#) de este capítulo.

El operando insertado para la búsqueda no es válido!

En el campo *Buscar* de la ventana de diálogo *Buscar operando* se ha escrito un operando absoluto o simbólico no válido. Corregir el operando.

La lista completa de los operandos absolutos válidos se incluye en los párrafos *Direccionamiento*, que aparece en los capítulos dedicados a las notas de programación de los distintos PLC simulados.

Las reglas sintácticas para los símbolos se detallan en el apartado [Operandos absolutos y símbolos](#) de este capítulo.

Búsqueda finalizada en el área especificada.

PC-Sim ha terminado la búsqueda del operando indicado en la casilla *Buscar* de las ventanas de diálogo *Buscar Operando* o *Reemplazar operando* dentro del área especificada. Todas las ubicaciones del operando presentes en el programa se han encontrado.

Búsqueda finalizada en el área especificada. No se ha encontrado ninguna coincidencia.

PC-Sim ha terminado la búsqueda del operando indicado en la casilla *Buscar* de las ventanas de diálogo *Buscar Operando* o *Reemplazar operando* dentro de área especificada, sin encontrar ninguna localización del operando.

El operando insertado para la sustitución no es válido!

En el campo *Sustituir por* de la ventana de diálogo *Reemplazar operando* se ha escrito un operando absoluto o simbólico no válido. Corregir el operando.

El listado completo de los operandos absolutos válidos se incluye en los párrafos *Direccionamiento*, que aparece en los capítulos dedicados a las notas de programación de los distintos PLC simulados.

Las reglas sintácticas para los símbolos se detallan en el apartado [Operandos absolutos y símbolos](#) de este capítulo.

Tipo operandos no compatible! Imposible sustituir.

La casilla *Sustituir por* de la ventana de diálogo *Reemplazar operando* contiene un operando incompatible con el que aparece en la casilla *Buscar*. La sustitución es imposible.

Dos operandos son compatibles cuando tienen el mismo número de bits y, para los módulos de sistema del PS3, cuando son del mismo tipo.

Sustitución finalizada en el área especificada. Realizadas [número] sustituciones.

La sustitución de operando se ha realizado con éxito dentro del área especificada. El operando contenido en la casilla *Buscar* de la ventana de diálogo *Reemplazar operando* ha sido sustituido por el de la casilla *Sustituir por* el número indicado de veces.

Sustitución finalizada en el área especificada. No se ha encontrado ninguna coincidencia.

La sustitución de operando no se ha realizado. En el área especificada no se ha encontrado ninguna ubicación del operando indicado en la casilla *Buscar* de la ventana de diálogo *Reemplazar operando*.

El bloque es demasiado largo para la visualización de su estado.

PC-Sim para S5 100U admite un límite de 200 líneas para la longitud de los bloques AWL que pueden soportar la función de debug. Si el bloque supera este límite, no se puede efectuar el debug y este mensaje le avisa de tal incidencia. Intente dividir el bloque en dos partes, llamando al segundo al final del primero, con una instrucción de llamada incondicionada de bloque SPA y a activar el debug para ambos.

La casilla 'Bloques:' contiene datos no válidos.

En la casilla *Bloques:* de la ventana de diálogo *Imprimir documentación* se ha insertado un listado de bloques no válido. Corregir la inserción.

Encontrará más información acerca de la compilación de esta casilla en el apartado [Impresión de la documentación](#) del presente capítulo.

El proyecto especificado no contiene un programa AWL.

La operación de importación de un programa AWL no ha concluido con éxito ya que el proyecto especificado en la ventana de diálogo *Importar programa* no contiene código en este lenguaje. El programa AWL en uso no ha variado en nada.

El proyecto especificado no contiene un programa ladder.

La operación de importación de un programa ladder no ha concluido con éxito ya que el proyecto especificado en la ventana de diálogo *Importar programa* no contiene código en este lenguaje. El esquema de contactos en uso no ha variado.

El operando absoluto [sigla operando] ya está asociado al símbolo [sigla símbolo] en la Tabla símbolos en vigor. ¿Desea actualizar el comentario?

Durante un procedimiento de importación de símbolos, PC-Sim ha detectado que la pareja de [sigla operando] [sigla símbolo] ya forma parte de la *Tabla símbolos en vigor* con un comentario distinto y le pregunta si debe actualizar el comentario de la atribución con el que aparece en el archivo especificado para la importación.

Responder:

- Si si se desea sobrescribir el comentario actual con el que extraerá del archivo
- No si se desea dejar inalterado el comentario actual.

El operando absoluto [sigla operando] ya está presente en la tabla símbolos en vigor en asociación al símbolo [sigla símbolo]. ¿Desea actualizar la atribución con el nuevo símbolo?

Durante un procedimiento de importación de símbolos, PC-Sim ha detectado que el operando absoluto de la atribución que se está importando ya está presente en la *Tabla símbolos en vigor* asociado a un símbolo distinto. Un operando absoluto puede ir asociado a un único símbolo y por tanto se le pregunta si quiere sustituir el existente por el nuevo.

Responder:

- Si si se desea sobrescribir el símbolo actual con el que se extraerá del archivo
- No si se desea dejar inalterado el símbolo actual.

¡El símbolo [sigla símbolo] ya está presente en la Tabla símbolos en vigor! Imposible importar la atribución.

Durante un procedimiento de importación de símbolos, PC-Sim ha detectado que el símbolo que se está importando ya está presente en la *Tabla símbolos* en uso, asociado a un operando absoluto distinto. Un símbolo puede asociarse a un único operando absoluto y por tanto no es posible importar la atribución.

¡Operación finalizada! No se ha importado o actualizado ninguna atribución.

La operación de importación de símbolos ha finalizado sin que PC-Sim haya logrado importar nuevas atribuciones o actualizar las ya existentes.

¡Operación finalizada! Importadas o actualizadas [número] atribuciones.

La operación de importación de símbolos ha finalizado. PC-Sim ha importado o actualizado el número de atribuciones indicado y las ha insertado en la *Tabla símbolos*.

Operación de edit activa en una tabla de la ventana. ¿Realmente desea cerrar?

Se está cerrando la Ventana Tabla símbolos, la Ventana Estado variables, un editor de datos o bien un editor AWL de un bloque funcional pero aun está pendiente una operación de edit en una tabla de la ventana.

Responder:

- Si si se desea cerrar la ventana perdiendo todo lo que se estaba editando
- No si se desea continuar y terminar la operación de edit

'Tipo bloques no compatible! Imposible renombrar.

Se ha pedido el cambio de nombre de un bloque pero la nueva sigla escrita no es compatible con la anterior.

Consulte la tabla siguiente para aclaraciones acerca de la compatibilidad entre siglas.

Sigla anterior	Nueva sigla
OB	OB, PB, FB (*)
PB	OB, PB, FB (*)
FB	FB
DB	DB

(*) Un bloque OB o PB puede renombrarse en FB sólo si el editor utilizado es AWL.

Ver también [Programación estructurada](#) en el [Capítulo 14](#).

Errores de compilación

Al pasar al estado de RUN de la CPU, PC-Sim compila el programa del usuario en el código máquina ejecutable por el PLC simulado. Durante esta operación, se efectúan controles acerca de la corrección del código. En caso de error, el PLC no se pondrá en marcha y se mostrará un mensaje.

En esta sección detallamos y describimos los mensajes de error que pueden aparecer y, cuando ello sea posible, sugerimos una solución al problema.

Instrucción no reconocida

La instrucción del programa o del bloque AWL es errónea, o bien no pertenece al juego de instrucciones del PLC simulado. Corregir la instrucción o sustituirla, si es posible, por una o más instrucciones incluidas en el juego.

Programa del usuario demasiado largo

El programa del usuario, una vez compilado, excede los límites de memoria del PLC simulado. Si es posible, recorte el programa y reinicie la compilación.

Número de parámetros erróneo en la llamada a bloque funcional

El número de los parámetros en la instrucción de llamada a bloque funcional parametrizado no corresponde con el que el bloque espera. Añadir los parámetros que faltan o eliminar los sobrantes.

Ver [Bloques Funcionales \(FB\)](#) en el [Capítulo 14](#) para más informaciones sobre la parametrización de los bloques funcionales.

Tipo de parámetro erróneo en la llamada a bloque funcional

El tipo de uno de los parámetros en la instrucción de llamada a bloque funcional parametrizado no corresponde al que el bloque espera.

Ver [Bloques Funcionales \(FB\)](#) en el [Capítulo 14](#) para más informaciones sobre la parametrización de los bloques funcionales.

Detectada la ausencia de la instrucción de fin de bloque

Falta el finalizador de bloque, es decir, la operación BE. Escriba la operación al final del bloque y reinicie la compilación poniendo la CPU en Run.

Etiqueta no encontrada

El compilador del programa del usuario no logra localizar, en interior del bloque, la etiqueta de llegada de una instrucción de salto. Verificar el código del bloque e insertar la etiqueta.

Para más informaciones ver los apartados [Introducción de etiquetas](#) en este capítulo y [Operaciones de salto](#) en el [Capítulo 14](#).

Operación integrativa en bloque organizativo o de programa

Un bloque organizativo (OB) o de programa (PB) contiene una operación del set integrativo. Estas operaciones tan sólo pueden programarse en el interior de bloques funcionales (FB). Si es posible, sustituya la operación por una o más operaciones del set fundamental o bien renombre el bloque asignándole la sigla de un bloque funcional.

Para más informaciones ver el apartado *Programación estructurada*, la **Tabla 18: operaciones fundamentales** implementadas y la **Tabla 19: operaciones integrativas** implementadas en el *Capítulo 14* y todo lo dicho a propósito del *Pulsador Renombrar* de la *Ventana Lista bloques (S5 100U)* en este capítulo.

Utilización múltiple de módulo de sistema

En el interior del programa del usuario existen dos o más referencias al mismo módulo de sistema. Cambiar los operandos repetidos para resolver el problema.

Para más informaciones sobre este tema ver *Módulos de sistema* en el *Capítulo 12*

Operando no especificado

En el programa, en el bloque ladder o AWL no se ha insertado el operando correspondiente a una operación o un elemento del esquema. Insertar el operando y reiniciar la compilación poniendo en Run la CPU.

Operando no reconocido

En el programa o en el bloque ladder o AWL se ha insertado un operando, que el compilador no reconoce, en correspondencia con una operación o un elemento del esquema. Corregir el operando y reiniciar la compilación poniendo de nuevo en Run la CPU.

Para ver un listado de los operandos que los PLC simulados soportan, consulte los apartados *Direccionamiento* en el capítulo dedicado a las notas de programación del PLC que esté utilizando.

Tipo de operando erróneo

En el programa o en el bloque ladder o AWL se ha insertado un operando no válido en correspondencia con una operación o un elemento del esquema. Corregir el operando y reiniciar la compilación poniendo de nuevo en Run la CPU.

Para ver un listado de los operandos que los PLC simulados soportan, consulte los apartados *Direccionamiento* en el capítulo dedicado a las notas de programación del PLC que esté utilizando.

Instrucción no compatible con la secuencia

En el interior de una secuencia AWL para PLC PS3 se ha utilizado una instrucción con un tipo de datos diferente respecto al de la secuencia. Corregir la instrucción.

Para más informaciones ver el apartado *Secuencia* en el *Capítulo 12*.

Superación de la capacidad del stack

En el interior de una secuencia AWL binaria para PLC PS3 se han utilizado más de 8 operaciones de carga consecutivas. En esta situación el registro de stack se saturaría en la ejecución del programa. Divida la secuencia binaria en secuencias más cortas, apoyando los resultados intermedios en merker bit.

Para más informaciones ver [Registro de stack](#) y la operación *L (Carga)* en el [Capítulo 12](#).

Petición de datos de stack vacío

En el interior de una secuencia AWL binaria para PLC PS3, el número de operaciones con el registro de stack es mayor que su contenido. Corregir la secuencia binaria.

Para más informaciones ver [Registro de stack](#) y le operaciones *A (And)*, *O (Or)* y *XO (Or exclusivo)* en el [Capítulo 12](#).

Registro de stack no vacío al final de una secuencia binaria

En el interior de una secuencia AWL binaria para PLC PS3, el número de operaciones con el registro de stack es menor mayor que su contenido. Corregir la secuencia binaria.

Para más informaciones ver [Registro de stack](#) y las operaciones *L (Carga)*, *A (And)*, *O (Or)* y *XO (Or exclusivo)* en el [Capítulo 12](#).

Operación de inicio de secuencia no válida

En el interior de un programa AWL para PS3, una secuencia se inicia con una operación no prevista para esta finalidad. La de carga es la única operación válida para el inicio de una secuencia. Corregir la instrucción.

Para más informaciones ver [Secuencia](#) y la operación *L (Carga)* en el [Capítulo 12](#).

Llamada a bloque funcional no presente en el proyecto

Está programada una llamada a bloque funcional SPA FB o SPB FB no presente en el proyecto. A diferencia de lo que sucede para las llamadas a bloques organizativos o de programa (OB y PB), aquellas a bloques funcionales (FB) necesitan que el bloque esté presente en el proyecto para el control de los parámetros. Eliminar la llamada o crear el bloque en el proyecto.

Ver [Programación estructurada](#) y [Operaciones sobre bloques](#) en el [Capítulo 14](#) y [Comando Nuevo bloque \(S5 100U\)](#) en el presente capítulo.

Error en la correspondencia del número de paréntesis

Un bloque AWL con la simulación del PLC S5 100U contiene un número de operaciones de apertura de paréntesis 'U(' o 'O(' distinto del número de las de cierre ')

Verificar el bloque especificado y corregir poniendo un número igual de paréntesis.

Errores de ejecución

Los errores relacionados a continuación no pueden, por su naturaleza, detectarse en la fase de compilación del programa del usuario y sólo se manifiestan durante su ejecución. Cuando PC-Sim encuentra uno de estos errores sitúa automáticamente la CPU en STOP.

Superado el máximo número posible de llamadas de bloque anidadas

El sistema operativo del PLC simulado ha detectado que se ha superado el límite de 16 en las llamadas de bloque anidadas y señala el error. Corregir el programa del usuario.

Para más informaciones ver [Programación estructurada](#) y [Operaciones sobre bloques](#) en el [Capítulo 14](#).

Bloque de datos inexistente

En la ejecución del programa del usuario se ha encontrado una operación de apertura de bloque de datos que hace referencia a un bloque no presente en el proyecto. Programar el DB requerido en el interior del proyecto o bien corregir el operando de la instrucción, cambiando la sigla por la de un DB existente.

Ver el apartado [Bloques de Datos \(DB\)](#) y la descripción de la operación [A \(Apertura de un bloque de datos\)](#) en el [Capítulo 14](#) y el [Comando Nuevo bloque \(S5 100U\)](#) en el presente capítulo.

Ningún bloque de datos abierto

En la ejecución del programa del usuario ha aparecido una operación de acceso a datos sin que en ese momento esté abierto ningún bloque de este tipo. Corregir el programa abriendo el bloque de datos requerido, con la instrucción A DB, antes de la operación de acceso a datos.

Ver el apartado [Bloques de Datos \(DB\)](#) y la descripción de la operación [A \(Apertura de un bloque de datos\)](#) en el [Capítulo 14](#).

Superado el número máximo de data word en el bloque de datos en uso

En la ejecución del programa del usuario se ha encontrado una operación de acceso a datos que hace referencia a una data word posterior a la última programada en el bloque de datos. Corregir el programa, verificando también la apertura del bloque de datos deseado, o aumentar el número de data word en el bloque de datos en cuestión.

Ver el apartado [Bloques de Datos \(DB\)](#) en el [Capítulo 14](#).

Operación no prevista después de una operación de elaboración B MW o B DW

Durante la ejecución del programa del usuario se ha encontrado una operación de elaboración seguida de una instrucción no direccionable. Corregir el programa.

Para más informaciones ver la descripción de la operación [B \(Elaboración de palabra de datos o merker\)](#) en el [Capítulo 14](#)

El Programa del usuario no devuelve el control al sistema operativo del PLC simulado. ¿Llevar a Stop la CPU?

Aparentemente la CPU, a causa de las operaciones de salto programadas, ejecuta siempre la misma parte de código, sin regresar al sistema operativo durante cerca de 10 segundos. PC-Sim ha detectado este problema y le pide si quiere terminar la elaboración del programa del usuario.

Responder:

- Si si se desea terminar poniendo en Stop la CPU.
- No si se desea esperar otros 10 segundos a que el programa se desbloquee.

Para más información lea el apartado *Control del tiempo de ciclo* en el capítulo dedicado a las notas de programación del PLC en concreto que esté utilizando.

Llamada indirecta a bloque funcional parametrizado

Una operación de elaboración va seguida de una llamada a bloque funcional, pero este último resulta ser parametrizado. La operación no es posible.

Controlar el contenido de la data word o de la merker word direccionada en la operación de elaboración o eliminar los parámetros del bloque funcional.

Para más información ver los apartados [Operaciones de elaboración](#) en el [Capítulo 14](#) y [Parametrización de bloques funcionales](#) en el presente capítulo.

Llamada indirecta a bloque funcional inexistente

Una operación de elaboración va seguida de una llamada a bloque funcional, pero este último no está presente en el proyecto en uso. La operación no es posible.

Controlar el contenido de la data word o de la merker word direccionada en la operación de elaboración o añadir el bloque funcional al proyecto.

Para más informaciones, ver el apartado [Operaciones de elaboración](#) en el [Capítulo 14](#).

CAPÍTULO 4

PL-SIM: EL SIMULADOR DE INSTALACIONES

Introducción

PL-Sim es el módulo de AW-SYS dedicado a la simulación de instalaciones.

Las instalaciones de PL-Sim han sido ideadas y realizadas teniendo siempre presente su finalidad didáctica, son fáciles de usar y, generalmente, ofrecen la posibilidad de realizar diferentes ejercicios. Abarcan distintos campos: desde las simples simulaciones de bancos de electropneumática a las más complejas aplicaciones de la automatización en instalaciones civiles e industriales.

Respecto a los bancos de simulación o a los modelos reales, las instalaciones simuladas tienen la ventaja de ser fáciles de usar, de no exigir tarado y, algo no menos importante, ¡de poder usarse una y otra vez sin romperse jamás!. Además, algunas de las instalaciones simuladas, serían casi imposibles de realizar como modelos reales, aparte de su precio absolutamente prohibitivo, en términos del espacio ocupado, los costes de realización y de mantenimiento. Imaginen, por ejemplo, lo que podría costar la realización o la compra de un modelo de ascensor o, aún peor, de una instalación de mezclado, del tipo como el que se describe más adelante en este capítulo y que se facilita, como todos los demás, en el paquete AW-SYS.

Los sensores y los actuadores presentes en las instalaciones simuladas pueden conectarse a las entradas y salidas de los PLC de PC-Sim o a sistemas de control reales a través del módulo PC-RIS. La activación de un sensor ON-OFF cerrará la entrada del PLC al que está conectado, y la activación de una salida del PLC activará el actuador de la instalación conectada a aquella.

La misión del usuario de AW-SYS será la de proyectar y redactar, una vez haya elegido uno de los PLC y una de las instalaciones, un programa para el primero que resuelva la tarea de hacer automático el funcionamiento del segundo.

En este capítulo se describen las distintas instalaciones que componen PL-Sim y las modalidades de conexión de sus componentes a los PLC simulados.

Puesta en marcha de las instalaciones

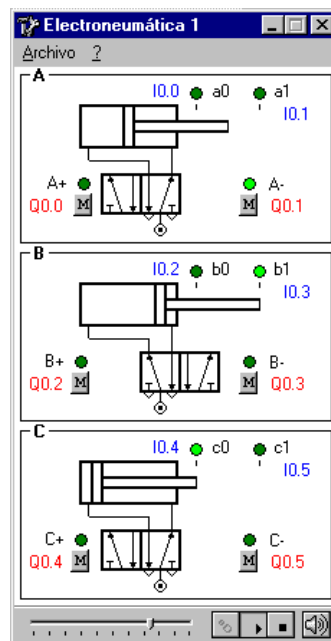
Las instalaciones de PL-Sim pueden ponerse en marcha tan sólo a través del [Menú Instalaciones](#) de PC-Sim o de PC-RIS: basta con hacer clic sobre la opción del menú que describe la instalación a poner en marcha. Esta se pondrá en marcha y el menú se desactivará para impedir el funcionamiento simultáneo con otras instalaciones. El [Menú Instalaciones](#) no volverá a activarse hasta que no se cierre la instalación.

Instalación n.1: Electroneumática 1

Composición de la instalación

Esta instalación de PL-Sim simula un banco para ejercicios de electroneumática. Está compuesto por tres cilindros neumáticos de doble efecto (A, B y C) pilotados, cada uno de ellos, por un distribuidor 5/2 biestable. El mando de los distribuidores es de tipo eléctrico, y cada uno de ellos va provisto tanto de un solenoide positivo (A+, B+ y C+) como de uno negativo (A-, B- y C-). Todos los cilindros van equipados con finales de carrera negativos (a0, b0 e c0) y positivos (a1, b1 e c1). los finales de carrera de los cilindros y los solenoides de los distribuidores disponen de pilotos luminosos, indicadores de estado. El encendido del LED indica la activación del final de carrera o la activación del solenoide. los distribuidores están provistos de pulsadores de control manual.

A partir de una condición de cilindro con final de carrera negativo, la activación del solenoide positivo hace conmutar el distribuidor, y el aire comprimido puede entrar en la cámara negativa del cilindro y salir de la positiva. El pistón se moverá, abandonando el final de carrera negativo. A continuación, recorriendo la carrera positiva, alcanzará el final de carrera positivo.



Por el contrario, la activación del solenoide negativo conmuta de nuevo el distribuidor y el aire comprimido puede fluir, en esta ocasión hacia la cámara positiva al salir de la negativa. El pistón se moverá, abandonando el final de carrera positivo; y a continuación, una vez recorrida la carrera negativa, alcanzará el final de carrera negativo.

En cuanto se ha activado un solenoide, el distribuidor conmuta y, al ser biestable, permanece en aquella posición, aunque se corte la alimentación al solenoide, hasta que no se activa el solenoide opuesto. La actuación simultánea de los dos solenoides conduce a la indeterminación del estado del distribuidor y por consiguiente de la posición del cilindro, y constituye una condición que debe evitarse en la programación del

PLC. Por otro lado, la activación de un solenoide cuando ya esté activo el opuesto, no hace conmutar el distribuidor, que se mantiene en su estado hasta que se elimina la activación del segundo solenoide.

Los pulsadores de control situados a los lados de cada distribuidor permiten la comutación manual y por ello resultan útiles durante la fase de puesta a punto del programa. Accionando estos pulsadores podrá hacer conmutar el distribuidor, haciendo que el cilindro efectúe la carrera positiva o la negativa.

Barra de instrumentos

La *Barra de instrumentos* aparece en el extremo inferior de la ventana. En ella están agrupados los controles generales de la instalación.



A continuación se muestra el listado de dichos controles.

Pulsador Control de audio

Activa o desactiva la emisión de sonidos a través de la tarjeta de sonido. Recuerde que sólo podrá escuchar los sonidos si en su ordenador está instalada una tarjeta de sonido y los controladores de audio de Windows se han cargado correctamente. Para informaciones sobre como instalar los controladores, consulte su manual de Windows.

Pulsador Run

Permite la puesta en marcha de la base de tiempos de la instalación y desactiva el funcionamiento por pasos. Normalmente debe estar pulsado con el fin de permitir el movimiento de los distribuidores y los cilindros.

Pulsador Stop

Permite detener la base de tiempos de la instalación, es decir, congela la posición actual de los componentes neumáticos. Activa el funcionamiento por pasos.

Pulsador Paso

Permite el avance a pasos de la instalación y es útil cuando se tenga que examinar atentamente la evolución del ciclo que se ha realizado. A cada clic sobre este pulsador la neumática avanza un paso. Durante el funcionamiento a pasos se desactiva el sonido.

Selector de velocidad

Permite regular la velocidad del movimiento de los cilindros. Desplace el cursor hacia la izquierda para disminuir la velocidad de funcionamiento de la neumática o a la derecha para acelerarla.

Conexiones con el PLC

La tabla siguiente detalla los componentes de la instalación que pueden conectarse al PLC, subdivididos en entradas y salidas.

Entradas digitales (6)	Salidas digitales (6)
Final de carrera negativo a0	Solenoide A+
Final de carrera positivo a1	Solenoide A-
Final de carrera negativo b0	Solenoide B+
Final de carrera positivo b1	Solenoide B-
Final de carrera negativo c0	Solenoide C+
Final de carrera positivo c1	Solenoide C-

Cada final de carrera está dotado de un conmutador y, durante la operación de conexión al PLC, se podrá elegir el tipo de contacto a utilizar, NA o NC.

Para efectuar las conexiones, proceda según se describe en el apartado [Conectar las instalaciones al PLC](#) situado más adelante en este capítulo. Junto a cada componente aparecerá la dirección del canal conectado: en azul para las entradas, en rojo para las salidas.

Recuerde que no se pueden conectar dos sensores de la instalación al mismo canal de entrada del PLC. PL-Sim señala esta eventualidad con un mensaje de error.

Test de automatización

Para verificar la corrección del programa realizado, ponga en Run el PLC y controle que los movimientos de los cilindros se desarrollen siguiendo la secuencia prefijada. Si es necesario, reduzca la velocidad o avance paso a paso, tal como se ha explicado con anterioridad. Si ha usado merkers (flags), temporizadores o contadores en su programa, podrá habilitar [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si el ciclo no funciona correctamente, o no funciona en absoluto, le podrán resultar muy útiles las potencialidades de debug ofrecidas por el editor que haya utilizado para escribir el programa.

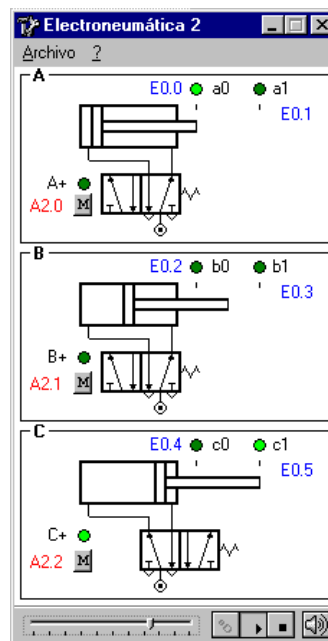
Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Instalación n.2: Electroneumática 2

Composición de la instalación

Esta instalación de PL-Sim simula un banco para ejercicios de electroneumática. Está compuesto por tres cilindros neumáticos de doble efecto (A, B y C) pilotados cada uno por un distribuidor 5/2 monoestable. El mando de los distribuidores es de tipo eléctrico y cada uno de ellos está provisto de un solenoide positivo (A+, B+ y C+), el retorno es de tipo mecánico a muelle. Todos los cilindros están equipados con final de carrera negativos (a0, b0 y c0) y positivos (a1, b1 y c1). Final de carrera de los cilindros y solenoides de los distribuidores están dotados de pilotos luminosos indicadores de estado. El encendido del LED indica la actuación del final de carrera o la activación del solenoide. Los distribuidores van provistos de pulsadores para el accionamiento manual.

A partir de una condición de cilindro con final de carrera negativo, la activación del solenoide hace conmutar el distribuidor y el aire comprimido puede entrar en la cámara negativa del cilindro y salir de la positiva. El pistón se moverá, abandonando el final de carrera negativo y a continuación, realizando la carrera positiva, alcanzará el final de carrera positivo.



Por el contrario, la desactivación del solenoide permite que el muelle conmute nuevamente el distribuidor, y llevarlo otra vez a la posición estable. El aire comprimido puede entrar, en esta ocasión, en la cámara positiva y salir de la negativa. El pistón se moverá, abandonando el final de carrera positivo; y a continuación, tras hacer la carrera negativa, alcanzará el final de carrera negativo.

En cuanto se haya excitado el solenoide, el distribuidor conmuta y, al ser monoestable, permanece en dicha posición, mientras el solenoide recibe alimentación. Cortando la alimentación, el distribuidor vuelve a su posición original.

El pulsador de control situado al lado de cada distribuidor permite la conmutación manual y por ello resulta útil en la fase de puesta a punto del programa. Pulsando este botón puede hacer conmutar el distribuidor, haciendo que el cilindro efectúe la carrera positiva y, soltándolo, conmutará nuevamente y el cilindro realizará la carrera negativa.

Barra de instrumentos

La *Barra de instrumentos* está situada en el extremo inferior de la ventana. En ella están reunidos los controles generales de la instalación.



A continuación, se detalla el listado relativo a esta instalación.

Pulsador Control de audio

Activa o desactiva la emisión de sonidos a través de la tarjeta de audio. Recuerde que tan solo podrá escuchar los sonidos si su ordenador dispone de una tarjeta de audio y los controladores de audio de Windows se han instalado correctamente. Para informaciones sobre cómo instalar los controladores, consulte su manual de Windows.

Pulsador Run

Permite la puesta en marcha de la base de tiempos de la instalación y desactiva su funcionamiento por pasos. Normalmente debe estar pulsado con el fin de permitir el movimiento de los distribuidores y de los cilindros.

Pulsador Stop

Permite detener la base de tiempos de la instalación, es decir, congela la posición actual de los componentes neumáticos. Activa el funcionamiento por pasos.

Pulsador Paso

Permite el avance por pasos de la instalación y es útil cuando se tenga que examinar atentamente el desarrollo del ciclo que se ha realizado. A cada clic sobre este pulsador la neumática avanza un paso. Durante el funcionamiento por pasos se desactiva el audio.

Selector de velocidad

Permite regular la velocidad de movimiento de los cilindros. Desplace el cursor a la izquierda para disminuir la velocidad de funcionamiento de la neumática o a la derecha para acelerarla.

Conexiones con el PLC

La tabla siguiente detalla los componentes de la instalación que pueden conectarse al PLC, subdivididos por entradas y salidas.

Entradas digitales (6)	Salidas digitales (3)
Final de carrera negativo a0	Solenoide A+
Final de carrera positivo a1	Solenoide B+
Final de carrera negativo b0	Solenoide C+
Final de carrera positivo b1	
Final de carrera negativo c0	
Final de carrera positivo c1	

Cada final de carrera está dotado de un conmutador y, durante la operación de conexión al PLC, se podrá elegir el tipo de contacto que se vaya a utilizar, NA o NC.

Para efectuar las conexiones proceda según las indicaciones del apartado [Conectar las instalaciones al PLC](#) situado más adelante en este capítulo. Junto a cada componente aparecerá la dirección del canal conectado: en azul para las entradas y en rojo para las salidas.

Recuerde que no se pueden conectar dos sensores de la instalación al mismo canal de entrada del PLC. PL-Sim informa de esta eventualidad con un mensaje de error.

Test de la automatización

Para verificar la corrección del programa realizado, ponga el PLC en Run y controle que los movimientos de los cilindros se realizan siguiendo la secuencia prefijada. Si es necesario, reduzca la velocidad o avance paso a paso, tal como se ha explicado con anterioridad. Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si el ciclo no funciona correctamente, o no funciona en absoluto, le podrán resultar muy útiles las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

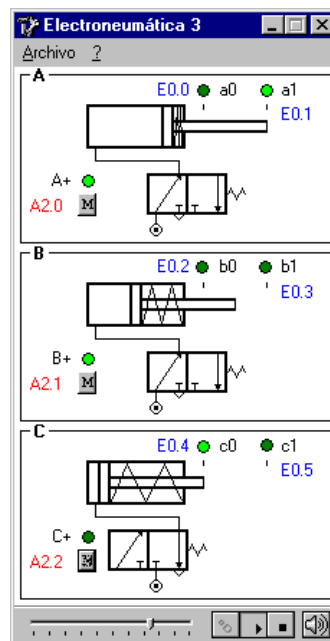
Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y conteo.

Instalación n.3: Electroneumática 3

Composición de la instalación

Esta instalación de PL-Sim simula el último banco para ejercicios de electroneumática. Está formado por tres cilindros neumáticos de simple efecto (A, B y C) pilotados cada uno de ellos por un distribuidor 3/2 monoestable. El mando de los distribuidores es de tipo eléctrico, y cada uno de ellos va provisto de un solenoide positivo (A+, B+ y C+), el retorno es del tipo mecánico a muelle. Todos los cilindros están equipados con finales de carrera negativos (a0, b0 y c0) y positivos (a1, b1 y c1). los finales de carrera de los cilindros y los solenoides de los distribuidores cuentan con pilotos luminosos indicadores de estado. El encendido del piloto indica la actuación del final de carrera o la activación del solenoide. los distribuidores cuentan con pulsadores para el control manual.

A partir de una condición de cilindro con final de carrera negativo, la activación del solenoide hace conmutar el distribuidor y el aire comprimido puede entrar en la cámara positiva del cilindro. El pistón se moverá, abandonando el final de carrera negativo. A continuación, recorriendo la carrera positiva y comprimiendo el muelle, alcanzará el final de carrera positivo.



Por el contrario, la desactivación del solenoide permite que el muelle conmute de nuevo el distribuidor y llevarlo a la posición estable. Ahora el aire comprimido puede salir de la cámara positiva y el muelle del cilindro llevará al pistón hacia atrás. Este, abandonando el final de carrera positivo, recorrerá la carrera negativa, hasta alcanzar el final de carrera negativo.

En cuanto se ha activado el solenoide, el distribuidor conmuta y, al ser monoestable, permanecer en esa posición mientras el solenoide sigue recibiendo alimentación. Cortando dicha alimentación, el distribuidor vuelve a su posición original.

El pulsador de control situado al lado de cada distribuidor permite la conmutación manual y por ello resulta útil en la fase de puesta a punto del programa. Accionando este pulsador podrá hacer conmutar el distribuidor, lo que hará que el cilindro recorra la carrera positiva. Al soltar el pulsador, el distribuidor conmutará de nuevo y el cilindro recorrerá la carrera negativa.

Barra de instrumentos

La *Barra de instrumentos* está situada en el extremo inferior de la ventana. En ella se muestran todos los controles generales de la instalación.



A continuación, se detalla el listado relativo a esta instalación.

Pulsador Control de audio

Activa o desactiva la emisión de sonidos a través de la tarjeta de audio. Recuerde que tan solo podrá escuchar los sonidos si su ordenador dispone de una tarjeta de audio y los controladores de audio de Windows se han instalado correctamente. Para informaciones sobre cómo instalar los controladores, consulte su manual de Windows.

Pulsador Run

Permite la puesta en marcha de la base de tiempos de la instalación y desactiva el funcionamiento por pasos. Normalmente debe estar presionado con el fin de permitir el movimiento de los distribuidores y de los cilindros.

Pulsador Stop

Permite detener la base de tiempos de la instalación, es decir, congela la posición en vigor de los componentes neumáticos. Activa el funcionamiento por pasos.

Pulsador Paso

Permite el avance por pasos de la instalación y es útil cuando se tenga que examinar atentamente el desarrollo del ciclo que se haya realizado. A cada clic sobre este pulsador la neumática avanza un paso. Durante el funcionamiento por pasos se desactiva el audio.



Selector de velocidad

Permite regular la velocidad de movimiento de los cilindros. Desplace el cursor a la izquierda para disminuir la velocidad de funcionamiento de la neumática o hacia la derecha para acelerarla.

Conexiones con el PLC

La tabla siguiente muestra los componentes de la instalación que pueden conectarse al PLC, subdivididos en entradas y salidas.

Entradas digitales (6)	Salidas digitales (3)
Final de carrera negativo a0	Solenoide A+
Final de carrera positivo a1	Solenoide A-
Final de carrera negativo b0	Solenoide B+
Final de carrera positivo b1	
Final de carrera negativo c0	
Final de carrera positivo c1	

Cada final de carrera está dotado de un conmutador y, durante la operación de conexión al PLC, se podrá elegir el tipo de contacto a utilizar, NA o NC.

Para efectuar las conexiones, proceda según lo descrito en el apartado [Conectar las instalaciones al PLC](#) que aparece más adelante en este mismo capítulo. Junto a cada componente aparecerá la dirección del canal conectado: en azul para las entradas y en rojo para las salidas.

Recuerde que no se pueden conectar dos sensores de la instalación a un mismo canal de entrada del PLC. PL-Sim indica este hecho con un mensaje de error.

Test de la automatización

Para verificar la corrección del programa realizado, ponga el PLC en Run y controle que los movimientos de los cilindros se realizan siguiendo la secuencia prefijada. Si es necesario, reduzca la velocidad o avance paso a paso, tal como se ha explicado con anterioridad. Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si el ciclo no funciona correctamente, o no funciona en absoluto, le podrán resultar muy útiles las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y conteo.

Instalación n.4: Automatización puerta

Composición de la instalación

Esta instalación de PL-Sim le propone una clásica aplicación de la automatización en el campo civil: una puerta eléctrica.

La parte superior de la ventana muestra la imagen de la entrada de una casa. En primer plano vemos la puerta, que dispone de un final de carrera de apertura y de cierre, célula fotoeléctrica, motor eléctrico y luz intermitente.

En la parte inferior de la ventana, están representados el mando a distancia, con los pulsadores de apertura y cierre, la regleta del cuadro eléctrico, destinada a efectuar las conexiones de los componentes de la instalación con las entradas y las salidas del PLC, y los pulsadores para el movimiento manual.

La regleta está dividida en dos partes: a la izquierda se recogen las señales de entrada y a la derecha las de salida. Para cada uno de ellos hay un piloto luminoso indicador del estado, de color verde, cuyo encendido se corresponde con la activación de la entrada o de la salida .



Si los dos últimos pilotos, de color amarillo, parpadean, nos informan de que se ha producido la intervención de una de las protecciones, térmica o magnética, del circuito de alimentación del motor.

La luz intermitente no está dotada de circuito de parpadeo autónomo, es decir, cuando recibe alimentación se mantiene encendida continuamente. Pensando siempre en el aspecto didáctico, usted será quien tendrá que realizar el intermitente, mediante una adecuada programación del PLC.

Los pulsadores de control manual le permitirán mover y elegir la posición de la puerta manualmente y no forman parte de la automatización, es decir, no están conectados al PLC.

La carrera de la puerta se detiene mecánicamente en correspondencia con el final de carrera de cierre y un poco después si se trata del final de carrera de apertura. Continuar alimentando el motor en esas

condiciones provocaría una sobrecarga y la consiguiente intervención del relé térmico de protección. La intervención del térmico, que se indica por medio del parpadeo del correspondiente piloto, mantiene abiertos los contactores de control del motor hasta que se efectúa su reinicio. Esta acción solo podrá efectuarse manualmente accionado el pulsador R. Por este motivo, cuando realice la automatización, tenga muy presente que debe quitar la alimentación al motor cuando la puerta alcanza una posición de final de carrera. Del mismo modo, si activa a la vez los contactores de apertura y cierre, al no estar dotados de interbloqueo ni eléctrico ni mecánico, se genera un cortocircuito que provocará la intervención del relé magnético del interruptor de protección. Este hecho, indicada con el parpadeo del piloto correspondiente, impedirá la alimentación del motor hasta su reinicio, que podrá efectuarse con el mismo pulsador R.

A continuación le facilitamos la descripción de las siglas usadas en la regleta eléctrica virtual y de las funciones que desempeñan su correspondiente señal.

Sigla	Descripción
FCC	final de carrera cierre puerta. Se activa cuando la puerta está completamente cerrada.
FCA	final de carrera apertura puerta. Se activa cuando la puerta está completamente abierta.
FOTOC.	célula fotoeléctrica. Se activa con la simulación de la interrupción del rayo.
TELESX	contacto excitado por la presión sobre el botón izquierdo del mando a distancia.
TELEDX	contacto excitado por la presión sobre el botón derecho del mando a distancia.
CIERRE	actuación del motor (cierre puerta), activando esta salida se pone en marcha el motor con sentido de rotación que permite el cierre de la puerta.
APERTURA	actuación del motor (apertura puerta), activando esta salida se pone en marcha el motor con sentido de rotación que permite la apertura de la puerta.
PARPAD.	Control luz intermitente .
TÉRMICO	indicador luminoso de la intervención del relé termico.
MAGNÉTICO	indicador luminoso de la intervención del relé magnetico.

Test de las entradas

Para comprobar la actuación del final de carrera FCA, pulse el control manual para la apertura de la puerta hasta que ésta se haya abierto del todo: el piloto correspondiente se activará.

Para el final de carrera FC, accione el control manual de cierre hasta que la puerta se haya cerrado del todo: el LED correspondiente se activará.

Para simular la intervención de la célula fotoeléctrica, situe el puntero del mouse entre los detectores, en la zona de apertura de la puerta. El puntero se transforma, para informarle de que en esa posición es posible interrumpir el rayo. Haga clic, el rayo se interrumpe y en la regleta, el piloto luminoso correspondiente a la célula fotoeléctrica se enciende.

Pulse los botones del mando a distancia para comprobar la activación de los contactos TELESX y TELEDX y verifique el encendido del LED correspondiente..

Barra de instrumentos

La *Barra de instrumentos* está situada en el extremo inferior de la ventana. En ella se muestran todos los controles generales de la instalación.

En esta instalación tan sólo está presente:

Pulsador Control de audio

Activa o desactiva la emisión de sonidos a través de la tarjeta de audio. Recuerde que tan solo podrá escuchar los sonidos si su ordenador dispone de una tarjeta de audio y los controladores de audio de Windows se han instalado correctamente. Para informaciones sobre cómo instalar los controladores, consulte su manual de Windows.

Modalidades de empleo

El propósito del ejercicio es redactar un programa para PLC que permita un funcionamiento correcto de la puerta. La automatización debe ejecutar las siguientes funciones:

- apertura de la puerta tras la orden de apertura desde el mando a distancia;
- cierre del puerta tras la orden de cierre desde el mando a distancia;
- paro del motor al alcanzar los finales de carrera;
- control temporizado del cierre;
- inversión de la marcha de la puerta si durante la fase de cierre se interrumpe el rayo de la célula fotoeléctrica;
- encendido de la luz intermitente durante las fases de movimiento de la puerta.

Con esta instalación se facilitan dos ejemplos para programar su automatización. En la *Guía on line de PC-Sim* están descritos detalladamente los procedimientos para su carga y comprobación.

Conexiones con el PLC

La tabla siguiente muestra los componentes de la instalación que pueden conectarse al PLC, subdivididos en entradas y salidas.

Entradas digitales (5)	Salidas digitales (3)
Final de carrera de cierre puerta	Control del motor para cierre
Final de carrera apertura puerta	Control del motor para apertura
Célula fotoeléctrica	Luz intermitente
Cierre con mando a distancia	
Apertura con mando a distancia	

Cada entrada está dotada de un conmutador y, durante la operación de conexión al PLC, se podrá elegir el tipo de contacto a utilizar, NA o NC.

Para efectuar las conexiones, proceda siguiendo las indicaciones descritas en el apartado [Conectar las instalaciones al PLC](#) que aparece más adelante en este mismo capítulo. Junto a cada componente aparecerá la dirección del canal conectado: en azul para las entradas y en rojo para las salidas.

Recuerde que no se pueden conectar dos sensores de la instalación a un mismo canal de entrada del PLC. PL-Sim indica este hecho con un mensaje de error.

Test de la automatización

Una vez haya escrito su programa o cargado el ejemplo, ponga el PLC en Run. A continuación, pulse el botón de apertura. El motor debe recibir la orden de abrir y, simultáneamente, deberá activarse el piloto intermitente hasta que se alcance el final de carrera de apertura.

Una vez haya transcurrido el tiempo programado, la puerta deberá cerrarse.

Ahora, intente verificar la orden cierre desde el mando a distancia. Si la puerta está abierta o se está abriendo, deberá iniciar la carrera de cierre.

Si durante una fase de cierre interrumpe el rayo de la célula fotoeléctrica, tal como se ha descrito anteriormente, la puerta deberá invertir el sentido de la marcha.

Durante todas las fases del movimiento de la puerta, la luz intermitente deberá encenderse.

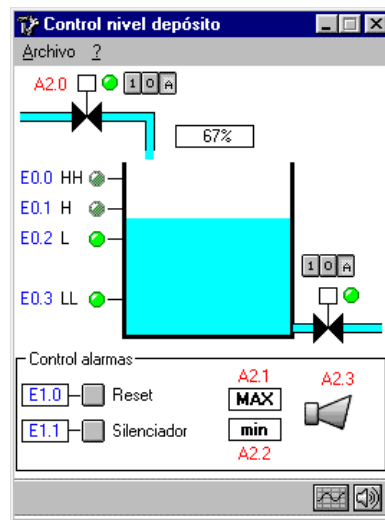
Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si la automatización no funciona correctamente, o no funciona en absoluto, le podrán resultar muy útiles las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Se utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Instalación n.5: Control del nivel de un depósito

Descripción

Esta instalación le propone la misión de regular el nivel de un depósito de líquido de manera que se pueda garantizar un suministro suficiente a las instalaciones situadas a un nivel más bajo. Cuando la cantidad de líquido sea demasiado escasa o bien cuando se corra el riesgo de que el depósito rebose, será preciso introducir o gestionar unas alarmas.



La parte superior de la ventana muestra el depósito equipado con cuatro indicadores de nivel que ejecutan las funciones que se describen a continuación.

Indicador de nivel	Función
--------------------	---------

HH	alto nivel de alarma: si el líquido supera este nivel, deberá activarse una alarma para advertir al operador de que el depósito está a punto de rebosar.
H	alto nivel de funcionamiento: indica el nivel máximo que el líquido puede alcanzar a lo largo de un funcionamiento normal.
L	bajo nivel de funcionamiento: indica el nivel mínimo que el líquido puede alcanzar a lo largo de un funcionamiento normal.
LL	bajo nivel de alarma: si el nivel del líquido se sitúa por debajo de este indicador, deberá activarse una alarma para advertir al operador de que el depósito no dispone de suministro suficiente para las instalaciones.

Cada indicador se activa cuando la altura del líquido en el depósito supera ese nivel. Dicha activación se manifiesta con el encendido del correspondiente piloto luminoso: en el ejemplo de la figura, aparecen activados LL y L y desactivados H y HH.

La canalización de acceso (introducción), la superior, está interceptada por la válvula de entrada. El actuador de esta última está controlado por el selector del modo de funcionamiento y dispone de las siguientes opciones:

Posición	Modo de funcionamiento
1	funcionamiento manual: válvula en posición de apertura;
0	funcionamiento manual: válvula en posición de cierre;
A	funcionamiento automático: la posición de cada válvula está determinada por la salida del PLC a la que está conectada.

La canalización de salida, la inferior, está interceptada por la válvula de salida. El actuador de esta última está controlado por el selector del modo de funcionamiento y dispone de las siguientes opciones:

Posición	Modo de funcionamiento
1	funcionamiento manual: válvula en posición de apertura;
0	funcionamiento manual: válvula en posición de cierre;
A	funcionamiento automático: la posición de la válvula puede controlarse en el modo automático y en el modo manual para simular las diversas necesidades de suministro a las instalaciones situadas a un nivel más bajo.

No se puede, ni tendría sentido teniendo en cuenta la finalidad propuesta, que esta válvula estuviera controlada por el PLC.

La parte inferior de la ventana muestra el cuadro de indicadores y de control de las alarmas. En dicho cuadro podrá disponer de los componentes que se detallan a continuación.

Pulsador Reset

Permite restaurar una alarma. Debe apagar la señal de alarma si la causa que la ha provocado ha cesado.

Pulsador Silenciador

Permite el silenciado de la sirena. Debe apagar la sirena mientras no se genere una nueva alarma.

Luz indicadora de alarma de máximo nivel

Es la representación de un indicador óptico de alarma de máximo nivel. Admite dos representaciones gráficas:


 luz apagada con salida del PLC a cero.

 luz encendida con salida del PLC a uno.

Luz indicadora de alarma de mínimo nivel



Es la representación de un indicador óptico de alarma de mínimo nivel. Admite dos representaciones gráficas:

 luz apagada con salida del PLC a cero.

 luz encendida con salida del PLC a uno.

Sirena

Es la representación de la sirena de alarma. Puede estar conectada a una salida cualquiera del PLC. Admite dos representaciones gráficas:

-  sirena apagada con salida del PLC a cero
-  sirena activada con salida del PLC a uno.

Si en su PC hay una tarjeta de sonido instalada, podrá oír la sirena. Si no es así, deberá orientarse por su representación gráfica.

Barra de instrumentos

La *Barra de instrumentos* está situada en extremo inferior de la ventana. En ella se muestran todos los controles generales de la instalación.



A continuación, se detalla el listado correspondiente a esta instalación.

Pulsador Control de audio

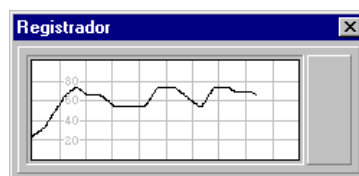
Activa o desactiva la emisión de sonidos por medio de la tarjeta de audio. Recuerde que tan solo podrá escuchar los sonidos si su ordenador dispone de una tarjeta de audio y los controladores de audio de Windows se han instalado correctamente. Para más información sobre cómo instalar los controladores, consulte su manual de Windows.

Pulsador Registrador

Muestra la simulación del registrador de papel que muestra el avance temporal (el trend) de la señal del transmisor de nivel.

Ventana Registrador

Simula un componente muy común en las aplicaciones industriales: la grabadora de papel. Sobre un soporte de papel, con un eje vertical graduado con porcentajes y un eje horizontal con una escala de tiempos, se va trazando, momento a momento, el punto que corresponde al valor del nivel del depósito. De este modo se plasma gráficamente la evolución a lo largo del tiempo (el trend) de la señal de nivel, lo cual le permite valorar la eficacia de la regulación efectuada.



Una vez se ha agotado el espacio de visualización, la ventana “cae”: los datos más antiguos se pierden y así se dispone de espacio para los nuevos que van llegando.

Modalidades de empleo

La finalidad del ejercicio es escribir un programa para PLC que permita mantener el nivel del depósito entre los niveles H y L y genere una alarma cuando el nivel sea superior a HH o inferior a LL. Por ejemplo, el control de la alarma podría producirse en alguna de las siguientes modalidades:

- si el nivel está por encima de HH o por debajo de LL deberá ponerse en marcha la sirena y la luz correspondiente a la alarma (min o max)
- la sirena deberá apagarse si se pulsa el *Pulsador Silenciador* o si el nivel vuelve a situarse fuera de las condiciones de alarma
- la luz deberá apagarse si el nivel vuelve a situarse fuera de las condiciones de alarma y si se hace clic en el *Pulsador Reset*. Esto garantiza la memoria de la alarma y el hecho de que un operador, momentáneamente distraído, no pase por alto una posible activación de la alarma y su posterior restauración.

Además, el programa podría mejorarse haciendo parpadear las luces indicadoras de min y MAX durante la situación de alarma y haciendo que estén fijas cuando la alarma se restaura. Se apagarán a continuación de haber pulsado *Pulsador Reset*.

Conexiones con el PLC

La tabla siguiente detalla los componentes de la instalación que pueden conectarse al PLC, subdivididos en entradas y salidas.

Entradas digitales (6)	Salidas digitales (4)
Indicador de nivel HH	Actuador válvula de entrada
Indicador de nivel H	Luz de alarma de máximo
Indicador de nivel L	Luz de alarma de mínimo
Indicador de nivel LL	Sirena
Pulsador Reset	
Pulsador Silenciador	

Cada entrada está dotada de un conmutador y durante la operación de conexión al PLC, podrá elegir el tipo de contacto a utilizar, NA o NC.

Para efectuar las conexiones, proceda según lo descrito en el apartado *Conectar las instalaciones al PLC* que aparece más adelante en este mismo capítulo. Junto a cada componente aparecerá la dirección del canal conectado: en azul para las entradas y en rojo para las salidas.

Recuerde que no se pueden conectar dos sensores de la instalación a un mismo canal de entrada del PLC. PL-Sim indica este hecho con un mensaje de error.

Test de la automatización

Para verificar la corrección del programa realizado ponga en Run el PLC y en automático las válvulas de entrada y de salida. Mientras, observe el nivel del depósito: una vez superado el nivel H, la válvula de entrada deberá cerrarse y el depósito se irá vaciando en función de las exigencias de las instalaciones situadas a un nivel más bajo. Cuando el nivel sea inferior a L, la válvula de entrada deberá abrirse de nuevo.

Para comprobar la alarma de máximo, ponga la válvula de entrada en posición de apertura manual (selector en 1) y cierre también la válvula de salida (selector en 0), simulando así un mal funcionamiento de los elementos a analizar. El nivel del depósito aumentará progresivamente y, una vez superado el nivel HH, deberán encenderse la sirena y el indicador de máximo (MAX). La primera deberá apagarse al pulsar el silenciador y el segundo, tras pulsar el botón reset de alarma, una vez haya desaparecido la causa que ha provocado la alarma.

Para hacer desaparecer la causa de alarma es preciso que el nivel vuelva a estar comprendido entre LL y HH. Para conseguirlo, vuelva a poner en automático las dos válvulas (Selectores en A) y espere.

Para comprobar la alarma de mínimo ponga la válvula de entrada en situación de cierre manual (selector en 0) y abra la válvula de salida (selector en 1) simulando así un mal funcionamiento de los elementos a verificar. El nivel del depósito disminuirá progresivamente y, una vez esté por debajo del nivel LL, deberán encenderse la sirena y el indicador de mínimo (min). La primera deberá apagarse al pulsar el botón del silenciador y el segundo, tras pulsar el botón de reset de alarmas, una vez haya desaparecido la causa que ha provocado la alarma.

Para hacer desaparecer la causa de alarma es preciso que el nivel vuelva a estar comprendido entre LL y HH. Para conseguirlo, vuelva a poner en automático las dos válvulas (Selectores en A) y espere.

Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si la automatización no funciona correctamente, o no funciona en absoluto, le podrán resultar muy útiles las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Por último, si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Como última posibilidad, intente comparar la solución que Ud. Ha encontrado con la que Autoware propone, cargando uno de los ejemplo que se proporcionan para esta instalación. La descripción de los procedimientos para su instalación y posterior test podrá encontrarlas en la [Guía on line de PC-Sim](#).

Conectar las instalaciones al PLC

Todo cuanto se detalla en este apartado únicamente es válido para las instalaciones descritas hasta ahora. Para las siguientes, las modalidades de conexión serán distintas y se describirán en un apartado especial.

Configurar las conexiones

Para configurar una conexión entre un componente de una instalación y el PLC, proceda ejecutando los siguientes pasos:

Haga clic sobre el componente de la instalación que desee conectar al PLC para que así aparezca la ventana de dialogo *Seleccionar conexión* para las entradas o para las salidas.

PL-Sim determina automáticamente el tipo de conexión en función del componente sobre el que se ha hecho clic. Además, la numeración de los canales varía automáticamente en función del PLC simulado en uso. En caso de conexión a un PLC real, a través del módulo PC-RIS, se mostrarán numeraciones que harán referencia a los canales de la EasyPort D16.



- seleccione el pulsador opción que indica la dirección del canal que se quiere conectar.
- sólo para las entradas, seleccione el tipo de contacto que se desee utilizar, normalmente abierto (NA) o normalmente cerrado (NC).
- pulse OK para confirmar.

En correspondencia con el componente preseleccionado aparece la ruta de la conexión al PLC, en azul para las entradas o en rojo para las salidas.

En el caso de que se haya efectuado una conexión a una entrada del PLC, PC-Sim se encarga de desactivar el pulsador/interruptor conectado a tal entrada. El LED del canal correspondiente asumirá ese estado (encendido o apagado) en función del estado del contacto (cerrado o abierto) del componente conectado. Ver [Pulsadores/interruptores](#) en el apartado [Ventana PLC](#) del [Capítulo 3](#).

Se pueden conectar dos o más actuadores de la instalación a una misma salida del PLC, pero a cada entrada no podrá conectarse más de un sensor. La operación de conexión de un sensor a una entrada ya utilizada se indica mediante un mensaje de error.

Modificar las conexiones

Para modificar una conexión entre un componente de una instalación y el PLC, proceda siguiendo los siguientes pasos:

- haga clic sobre el componente de la instalación del que se quiere modificar la conexión para que así aparezca la ventana de dialogo *Seleccionar conexión* para las entradas o para las salidas.

PL-Sim determina automáticamente el tipo en función del componente clicado.

Las opciones hechas con anterioridad para el canal o también para el tipo de contacto aparecen ya seleccionadas..

- seleccione el pulsador opción que indica la dirección del canal que se quiere conectar.
- sólo para las entradas, seleccione el tipo de contacto que se desee utilizar, normalmente abierto (NA) o normalmente cerrado (NC).
- pulse OK para confirmar.

PL-Sim se encarga de actualizar la conexión en su memoria y la referencia junto a su componente. Si se trataba de un sensor, PC-Sim se ocupa de reactivar el pulsador/interruptor correspondiente a la conexión antigua y de desactivar el del nuevo. Ver [Pulsadores/interruptores](#) en el apartado [Ventana PLC](#) del [Capítulo 3](#).

Eliminar conexiones individuales

Para retirar una conexión entre un componente de la instalación y el PLC, proceda siguiendo los siguientes pasos:

- haga clic sobre el componente de la instalación del que se quiere retirar la conexión para que así aparezca la ventana de dialogo *Seleccionar conexión* para las entradas o para las salidas. PL-Sim determina automáticamente el tipo en función del componente clicado.
- pulse *Desconectar*.

PL-Sim se encarga de eliminar la conexión de su memoria, de borrar la referencia situada junto al componente. Si se trataba de un sensor, PC-Sim se encarga de reactivar el pulsador/interruptor correspondiente. Ver [Pulsadores/interruptores](#) en el apartado [Ventana PLC](#) del [Capítulo 3](#).

Comandos de los menú

Las primeras cinco instalaciones de PL-Sim, las descritas hasta ahora, están provistas de menús que sirven principalmente para la gestión de las conexiones con el PLC. En este apartado le facilitamos una descripción de los comandos que se dichos menús incluyen.

Menú Archivo

En el menú *Archivo* están agrupadas las órdenes para la gestión (abrir, cerrar, guardar) de los archivos que contienen las conexiones entre la instalación y el PLC. Además, presenta la orden para el cierre de la instalación.

Comando Eliminar conexiones

Utilice este comando para suprimir todo el set de conexiones entre la instalación y el PLC.

Si las conexiones no se han guardado, PL-Sim le preguntará si desea hacerlo antes de eliminarlas. En caso de una respuesta afirmativa, si un archivo de conexiones está ya abierto, PL-Sim salva las conexiones en este archivo y so no le pedirá el nombre de un nuevo archivo con la correspondiente ventana de diálogo. Si responde negativamente, PL-Sim no procederá a guardar las conexiones actuales.

Tras la eliminación de las conexiones se habilitan de nuevo todos los pulsadores/interruptores del PLC.

Para más detalles sobre como efectuar conexiones, ver [Conectar las instalaciones al PLC](#) en este capítulo.

Comando Cargar conexiones

Este comando le permite cargar un set de conexiones entre la instalación y el PLC que haya guardado con anterioridad con el [Comando Guardar conexiones](#) del mismo menú.

Después de cargarlo, PC-Sim se encarga de desactivar los pulsadores/interruptores de las entradas del PLC para las que está previsto una conexión a la instalación.

Los archivo de las conexiones tienen extensión lin.

Para detalles sobre como efectuar conexiones ver [Conectar las instalaciones al PLC](#) en este capítulo.

Opciones de la ventana de diálogo

Buscar en:

Seleccione el recorrido en el que PL-Sim deberá leer el archivo que desea abrir.

Nombre archivo

Escriba el nombre del archivo que desee abrir o bien selecciónelo en el listado. Este último incluye los archivos con la extensión seleccionada en la casilla *Tipo de archivo*.

Tipo de archivo

Seleccione el tipo de archivo que se desea visualizar en el listado. Las opciones posibles son:

- Archivo conexiones (*.lin) relaciona los archivos con extensión 'lin' presentes en la carpeta en uso.
- Todos los archivos (*.*) relaciona todos los archivos de la carpeta en uso prescindiendo de su extensión.

Comando Guardar conexiones

Utilice este comando para guardar un set de conexiones realizadas entre la instalación y el PLC. A continuación, podrá recargarse el set con el comando *Cargar conexiones*, del mismo menú.

Los archivos de las conexiones tienen extensión lin.

Para detalles sobre como efectuar conexiones ver [Conectar las instalaciones al PLC](#) en este capítulo.

Opciones de la ventana de diálogo

Guardar en:

Seleccione la carpeta en la que desea guardar las conexiones.

Nombre archivo

Escriba un nuevo nombre de archivo para guardar las conexiones con un nombre distinto o bien seleccione uno del listado para guardarlo con el nombre de un archivo ya existente. PL-Sim se encarga de añadir la extensión lin si no se ha especificado.

EL listado incluye tan sólo los archivos de la carpeta que tienen la extensión seleccionada en la carpeta *Tipo de archivo*.

Tipo de archivo

Seleccione el tipo de archivo que se desea visualizar en el listado. Las opciones posibles son:

- Archivo conexiones (*.lin) relaciona todos los archivos con extensión 'lin' presentes en la carpeta en uso.
- Todos los archivos (*.*) relaciona todos los archivos de la carpeta en uso prescindiendo de su extensión.

Comando Salir

Este comando permite cerrar la instalación. Si las conexiones no se han registrado, PL-Sim le pregunta si desea guardarlas antes de salir.

En caso de una respuesta afirmativa, si existe un nombre de archivo en uso, PL-Sim guarda las conexiones en este archivo; si no es así, le pedirá el nombre de un nuevo archivo con la correspondiente ventana de diálogo. Si responde negativamente, PL-Sim no procede a guardar las conexiones en uso.

Tras el cierre de la instalación se habilitan de nuevo todos los pulsadores/interruptores del PLC y el menú *Instalaciones* de PC-Sim.

El cierre de PC-Sim provoca automáticamente el cierre de la instalación.

Teclado: Alt+F4

Menú ?

Se reúnen en este menú los comandos para obtener informaciones sobre la instalación y sobre su funcionamiento.

Comando Guía

Pone en marcha la guía de la instalación.

La guía de PL-Sim cumple las mismas funciones que las guías de Windows. Es sensible al contexto y posee las funciones de desplazamiento adelante-atrás, sumario, cronología y búsqueda. Además, se puede anotar, imprimir un tema, copiar para luego pegar, definir puntos de libro (señales de lectura).

La guía está compuesta por dos ventanas. La ventana principal muestra el sumario de temas, la ventana secundaria muestra los temas específicos que, periódicamente, le reclama la ventana principal.

Si bien las instalaciones de PL-Sim se han estudiado para ser fáciles de usar y de comprensión rápida e intuitiva, use ampliamente la guía: le será muy útil al inicio de sus práctica.

Teclado: F1. Si desea informaciones sobre un comando en concreto sitúese sobre éste usando el teclado y pulse F1. La guía, sensible al contexto, se abrirá en la opción que usted le haya sugerido.

Comando Guía rápida

Activa la modalidad de *Guía rápida* cambiando el puntero del mouse por el de guía rápida (flecha y signo interrogativo). En estas condiciones, haga clic sobre un elemento cualquiera de la instalación para recibir informaciones específicas a través de una ventana popup.

Teclado: Mayús+F1

Comando Acerca de [nombre instalación]

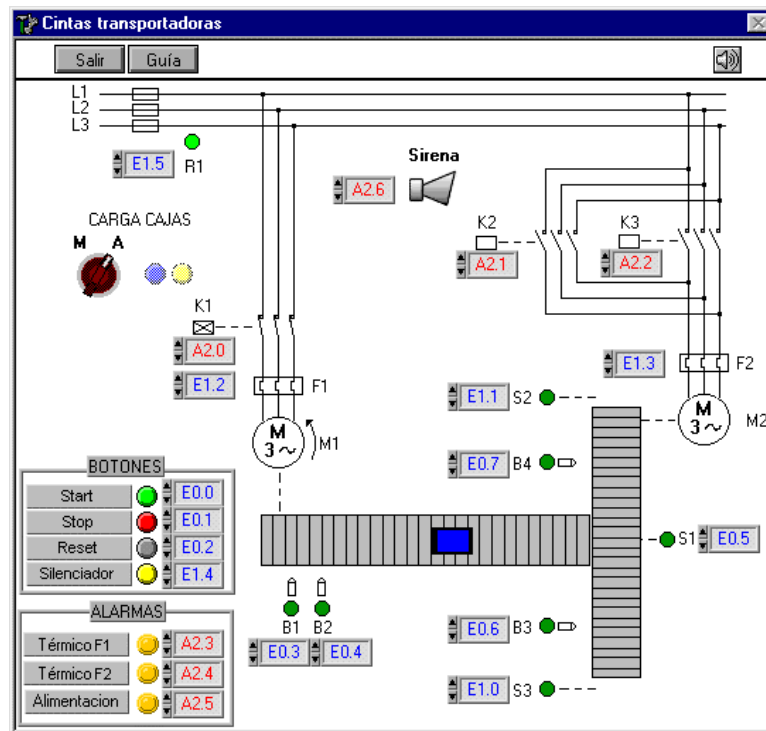
Muestra la ventana de informaciones acerca de la instalación en uso. En la que [nombre instalación] es una de las instalaciones descritas hasta ahora.

En ella podrá encontrar informaciones sobre la versión del programa, el nombre de la instalación y sobre el productor del software. Para cerrar la ventana pulse OK.

Instalación n.6: Cintas transportadoras

Descripción

Esta instalación muestra un clásico escenario industrial. Está formado por dos cintas transportadoras, cada una de ellas accionada por un motor eléctrico trifásico. La primera cinta, la horizontal, puede circular en un sólo sentido. La segunda, la vertical, puede hacerlo en los dos sentidos.



Su misión es la de transportar las cajas que se colocan sobre la línea, al inicio de la primera cinta. Esta cinta efectúa un primer movimiento y descarga las cajas sobre la segunda cinta. Desde ese punto podrán transportarse en las dos direcciones, en función de la rotación del motor M2. Las cajas pueden ser de dos longitudes diferentes, que se corresponden con dos colores. La diferencia de longitud podrá ser detectada por unos sensores, tal como veremos más adelante. La diferencia de color es un mero atributo estético, que no interviene en la programación para nada, pero que ayuda al operario al reconocimiento visual de cada tipo de caja.

Motores

Los motores representados en este escenario son de tipo asíncrono trifásico y se dirigen por medio de los contactores. El cierre de los contactores y la rotación de los motores se visualizan gráficamente: para los primeros se observará el desplazamiento de los contactos, y para los segundos aparecerá una flecha para indicar el sentido de rotación.

El primer motor (M1) mueve la cinta horizontal en un solo sentido y está alimentado a través del contactor K1.

El segundo (M2) controla la traslación de la cinta vertical en los dos sentidos. Para realizar la inversión de la marcha del segundo motor, se utilizan dos contactores (K2 y K3). Estos realizan el intercambio de las fases L1 y L3 de la alimentación. Cuando cierra K3, la secuencia de las fases en el motor M2 es L1-L2-L3, y cuando cierra K2 la secuencia se invierte a L3-L2-L1.

Por este motivo, preste atención a que los dos contactores no se activen a la vez. Esto causaría un cortocircuito entre las de la alimentación. En realidad, esta situación no podría darse en las instalaciones ya que la buena norma impone que se efectúe un interbloqueo entre los dos contactores. En efecto, los contactores de inversión deben interbloquearse eléctricamente por medio de los contactos auxiliares normalmente cerrados y también pueden interbloquearse mecánicamente. De este modo se impide que se cierren a la vez ambos contactores, situación que causaría un cortocircuito entre las dos fases. El bloque mecánico tiene el contactor de inversión en posición de abierto, si el contactor directo está aun cerrado, incluso en presencia de fuertes vibraciones mecánicas.

Por el contrario, aquí no se ha efectuado intencionadamente ningún interbloqueo para permitir su realización por enclavamiento lógico (en el programa del PLC) y para poder verificar su correcto funcionamiento. En consecuencia, un programa erróneo podría excitar a la vez los contactores en cuestión y provocar así un cortocircuito entre las fases de alimentación, determinando que se fundieran los fusibles de protección.

Intente provocar intencionadamente un cortocircuito: por ejemplo, conecte a una misma salida del PLC los contactores K2 y K3. Luego, escriba un simple programa que active esta salida al cierre de una determinada entrada. Ponga en Run el PLC y cierre la entrada.

Fusibles

En la línea de alimentación están dispuestos los fusibles de tipo aM (acompañamiento de motores) como protección contra los cortocircuitos y las sobrecargas fuertes.

Si provoca accidentalmente un cortocircuito en la alimentación, los fusibles se funden. En esta situación los motores no reciben alimentación, aunque los aparatos de control si continúen teniéndola.

Para sustituir virtualmente los fusibles, haga clic sobre uno de ellos. Los fusibles se restablecerán permitiendo de nuevo la alimentación de los motores .

Carga de cajas

La carga de las cajas sobre la cinta horizontal se realiza por medio de dos modalidades de funcionamiento cuya elección se realiza mediante el selector.



En la posición 'manual', la elección del instante de carga de la caja y del color de ésta se deja en manos del operador. En esta modalidad, los pulsadores de carga situados al lado del selector están activados y bastará con accionar el botón amarillo para cargar una caja corta, o el pulsador azul para una larga.



En cambio, si el selector del modo de funcionamiento se sitúa en 'automático', los pulsadores de carga se desactivan y las cajas, cuyo color (y por tanto la dimensión) se determina de manera casual, se cargarán continua y automáticamente sobre la primera cinta. Cada vez que una caja sale de la segunda cinta se carga una nueva en la primera.

Presencia de red

La presencia de alimentación se comprueba mediante el correspondiente relé de línea (R1) instalado a continuación de los fusibles.

Si los fusibles se interrumpen, R1 se desactiva y su contacto se abre. Por el contrario, si después de los fusibles sigue habiendo alimentación en el circuito, el contacto asociado al relé R1 se cerrará. Este contacto está conectado al PLC y Ud. puede comprobar su estado en su programa de automatización con el fin de señalar una alarma con el encendido de la luz indicada como 'ALIMENT.' y quizás también la sirena.

Relés térmicos

La protección frente a sobrecargas débiles se efectúa a través de los relés térmicos F1 y F2 situados por encima de cada uno de los motores. Se puede simular su intervención haciendo clic sobre ellos.

La norma impone que el contacto auxiliar normalmente cerrado del relé térmico se inserte en serie en el circuito de la bobina de mando del contactor de manera que, después de la intervención del térmico, eso impida que el motor vuelva a ponerse en marcha. Tampoco esta condición se ha incorporado expresamente a la instalación para permitir así su realización y una posible verificación posterior a nivel lógico, es decir, en el programa del PLC.







Panel de pulsadores

Este panel está formado por cuatro pulsadores, de los cuales le sugerimos a continuación una posibilidad de empleo.

Start		Puesta en marcha del ciclo automático de clasificación de cajas
Stop		Interrupción del ciclo o emergencia
Reset		Reconocimiento de alarmas
Silenciador		Silenciador de la sirena



Panel de alarmas

En el panel de alarmas están disponibles tres luces indicadoras. También para éstas le sugerimos una posibilidad de empleo, coherente con las etiquetas situadas al lado de las luces.

		Señal luminosa de la intervención del térmico F1
		Señal luminosa de la intervención del térmico F2
		Señal luminosa de la intervención de los fusibles de protección

Sirena

La instalación dispone de una sirena que puede activarse en caso de alarma. Admite dos representaciones gráficas:

-  sirena apagada con salida del PLC a cero
-  sirena activada con salida del PLC a uno.

Si en su PC hay instalada una tarjeta de audio podrá oír el sonido. Si no es así, podrá orientarse con la representación gráfica.

Sensores

Completan la instalación una serie de sensores, necesarios para la realización de la automatización. A continuación se detalla una lista con sugerencias para su utilización.

- Células fotoeléctricas B1 y B2 al principio de la primera cinta, para la detección de la presencia y las dimensiones de la caja. B1 se activa en cuanto se carga una nueva caja sobre la línea. B2 está situado a una distancia de B1 superior a la longitud de las cajas más cortas, pero inferior a la de las cajas más largas. Así, tras la puesta en marcha de la primera cinta, el cierre simultáneo de los contactos de las dos células fotoeléctricas indicará que la caja detectada es larga y lo contrario indicará que es corta.
- Final de carrera S1 a mitad de la segunda cinta: detección del paso de la caja por la segunda cinta.
- Final de carrera S2 y S3 en los extremos de la segunda cinta: detección de la salida de la caja de dicha cinta..
- Células fotoeléctricas B3 y B4 en posición intermedia sobre la segunda cinta: detección del paso de las cajas para un eventual conteo.

Barra de instrumentos

La *Barra de instrumentos* está situada en la parte superior de la ventana. En ella se reúnen los controles generales de la instalación. A continuación, se detalla el listado correspondiente a esta instalación.

 **Pulsador Salir**

Pulsándolo podrá cerrar la instalación.

Tras el cierre de la instalación se activan de nuevo todos los pulsadores/interruptores del PLC y el menú *Instalaciones* de PC-Sim.

El cierre de PC-Sim provoca automáticamente el cierre de la instalación.

 **Pulsador Guía**

Carga la guía de la instalación en uso.

Si bien las instalaciones de PL-Sim se han estudiado para ser fáciles de usar y para comprenderse rápida e intuitivamente, utilice la guía a menudo: le será muy útil al principio de sus prácticas.

 **Pulsador Control audio**

Activa o desactiva la emisión de sonidos a través de la tarjeta de audio. Se recuerda que puede escuchar los sonidos tan sólo si su ordenador tiene instalada una tarjeta de audio y los controladores de audio de Windows se han cargado correctamente. Para informaciones sobre cómo instalar los controladores, consulte su manual de Windows.

Modalidades de empleo

Puede automatizar esta instalación de varias maneras. Por ejemplo, intente distribuir las cajas azules (más largas) en la segunda cinta y en un sentido y las cajas amarillas (más cortas) en el sentido opuesto. Programe el conteo de todas las cajas que se han repartido en total, cuantas de color azul y cuantas de color amarillo.

O bien, pueden distribuir las cajas sin tener en cuenta su tamaño, sino tan sólo su número: por ejemplo, cinco hacia arriba y otras tantas hacia abajo.

Estamos seguros de que su fantasía le sugerirá otros modos posibles de funcionamiento


Encontrará ejemplos de programación para esta instalación en la carpeta Ejemplos del PLC que utilice y sus descripciones en la *Guía on line* de PC-Sim.

Conexiones con el PLC

La tabla siguiente enumera los componentes de la instalación que pueden conectarse al PLC, clasificados por entradas y salidas.

Entradas digitales (14)	Salidas digitales (7)
Célula fotoeléctrica B1	Contactador K1
Célula fotoeléctrica B2	Contactador K2
Célula fotoeléctrica B3	Contactador K3
Célula fotoeléctrica B4	Alarma térmica F1
Final de carrera S1	Alarma térmica F2
Final de carrera S2	Alarma falta de alimentación
Final de carrera S3	Sirena
Relé térmico F1	
Relé térmico F2	
Relé de línea R1	
Pulsador de start	
Pulsador de Stop	
Pulsador de reset	
Pulsador del silenciador	

Las conexiones de los sensores y de los actuadores al PLC se efectúan por medio de los correspondientes selectores situados junto a cada componente conectable. El texto que aparece en estos indica la ruta del canal de entrada o de salida del PLC al cual está conectado el citado elemento de la instalación. En azul se indican las conexiones a entradas del PLC y en rojo las conexiones a las salidas.

 selector de ruta del canal de entrada (caracteres en azul);

 selector de ruta del canal de salida (caracteres en rojo).

Para modificar una ruta de salida o de entrada:

- haga clic sobre el selector de la conexión a modificar y mantenga la presión; aparece el listado de las entradas o salidas en la escritura propia del PLC que esté utilizando;
- moviendo el mouse sitúese sobre la ruta preseleccionada;
- suelte el botón del mouse.

O bien:

- haga clic sobre los pulsadores con un triángulo, que están situados a la izquierda del selector: Arriba, para obtener el incremento de la ruta y abajo para que disminuya.

Si desea desconectar un elemento de instalación sitúe el selector de una de las dos maneras que acabamos de ver, sobre 'NC'.

De todos modos, cuando cargue la instalación le será propuesto un juego de conexiones preterminadas. Podrá utilizarlo o cambiarlo pero, en este último caso, recuerde que no se podrán guardar las conexiones

modificadas. En otras palabras, cada vez que inicie la ventana de esta instalación se le propondrán de nuevo las conexiones predeterminadas..

A diferencia de las cinco primeras instalaciones de PL-Sim, esta no controla la univocidad de la conexión a una entrada del PLC. Por tanto, preste mucha atención a evitar esta eventualidad. En cualquier caso, si conecta dos sensores a la misma entrada del PLC, estos realizarán una conexión lógica OR, es decir, sus contactos se conectarán en paralelo; o sea, será necesario y suficiente que uno de los dos esté cerrado para obtener el cierre de la entrada del PLC.

Test de la automatización

Para verificar la corrección del programa realizado ponga en Run el PLC y verifique que los movimientos de las cintas transportadoras se ejecutan siguiendo la secuencia prefijada.

Si ha usado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si la automatización no funciona correctamente, o no funciona en absoluto, le podrán ser de mucha utilidad las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Guía rápida

Si desea obtener informaciones relacionadas con un elemento concreto de la instalación, opere de la siguiente manera:

- sitúe el puntero del mouse sobre el elemento;
- pulse la combinación de teclas Ctrl+H;

en ese momento aparecerá, si está previsto para ese elemento en concreto, una ventana de guía rápida que lo describe brevemente.

La guía rápida es sensible al contexto, es decir, al desplazar el puntero del mouse sobre otro elemento, la descripción que aparece en la ventana se actualizará en función del nuevo elemento señalado.

Si desea fijar el contenido de la ventana haciendo que sea independiente de la posición del puntero, pulse la combinación de teclas Ctrl+G. Para hacer que de nuevo sea variable la descripción de la ventana en función del contexto, vuelva a pulsar Ctrl+G.

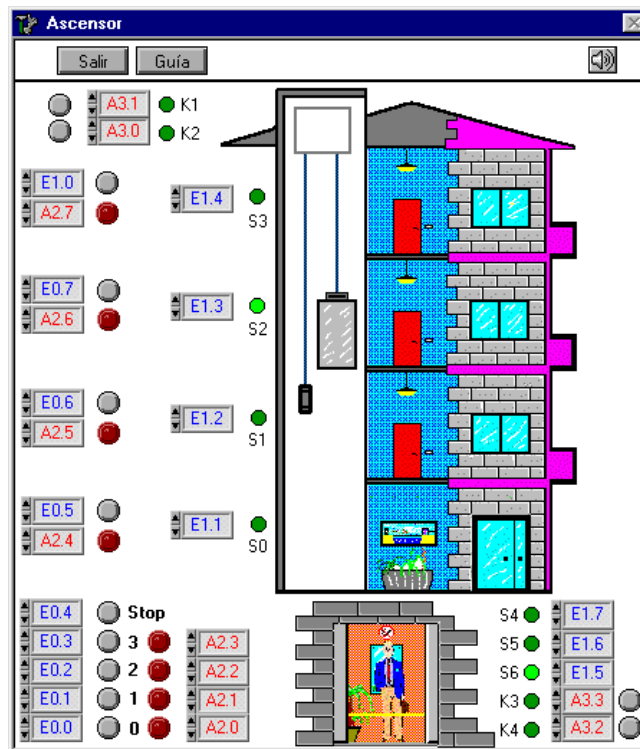
Por último si pulsa de nuevo la combinación de teclas Ctrl+H, la ventana de guía rápida se cerrará.

Instalación n.7: Ascensor

Descripción

El escenario de esta instalación muestra una finca de cuatro pisos dotada de un ascensor del tipo tradicional, es decir, constituido por una cabina con contrapeso, accionada por un motor eléctrico. El motor está situado en un local habilitado para ello situado encima del hueco del ascensor. A la izquierda, para cada piso, se sitúan los finales de carrera, los botones de llamada y retorno y las luces indicadoras.

En la parte de abajo aparece la entrada de la cabina, no visible en la representación del edificio, que permitirá controlar el estado y el movimiento de las puertas, la presencia de personas y la interrupción del rayo de la célula fotoeléctrica. A la izquierda de la puerta está representado el panel de control interno de la cabina, que aquí, por comodidad para su representación y su utilización, se ha situado fuera.



Mando de la cabina

La cabina del ascensor se mueve gracias a un motor asíncrono trifásico accionado a través de un teleinvertidor, formado por los contactores K1 y K2. La activación del contactor K1 determina la rotación del motor y la subida de la cabina. La activación del contactor K2 determina la rotación en sentido contrario del motor y la consiguiente bajada de la cabina.

Mando de las puertas

La cabina está dotada de puertas automáticas. Para abrir y cerrar las puertas basta controlar con un impulso las bobinas de los contactores K3 y K4, respectivamente.




Mandos manuales

Se puede controlar manualmente la subida y la bajada de la cabina del ascensor por medio de los correspondientes pulsadores situados, respectivamente, junto a los indicadores de estado de los contactores K1 y K2.

Así mismo, utilice los pulsadores que se encuentran junto a las bobinas K3 y K4 para controlar manualmente la apertura y el cierre de las puertas de la cabina.




Controles de piso

Con cada piso van asociados los siguientes controles:

-  pulsador de llamada al piso;
-  luz indicadora;
-  final de carrera de presencia de la cabina en el piso (S0, S1, S2, S3).

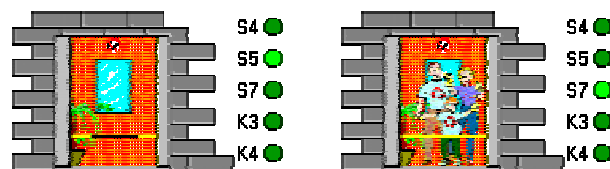
Controles de cabina

La cabina del ascensor está dotada de los siguientes controles:

-  S4 final de carrera de cierre de puertas;
-  S5 célula fotoeléctrica de las puertas;
-  S6 sensor de peso.

Para provocar la intervención de la célula fotoeléctrica, haga clic en el interior de la cabina, por debajo del rayo amarillo. El rayo se interrumpirá y el indicador S5 cambiará de estado.

Un funcionamiento análogo se obtiene pulsando la tecla de función F5.



Para desactivar el sensor haga clic nuevamente por debajo de la línea amarilla.

Para simular la entrada de personas en la cabina, haga clic en el interior de la cabina por encima de la línea amarilla. Aparecerán personas en el interior de la cabina y podrá observar el cambio de estado del sensor de peso S6.

Para simular la salida de las personas haga clic nuevamente por encima de la línea amarilla.



En el interior de la cabina hay una botonera compuesta por:

- pulsadores de envío al piso (0, 1, 2, 3);
- pulsador Stop;
- luces indicadoras del piso actual (0, 1, 2, 3).

Convenciones para las entradas y las salidas

Cada entrada o salida digital está dotada de un piloto indicador de estado: el encendido del piloto indica la activación de la entrada o bien la de la salida.

Barra de instrumentos

La *Barra de instrumentos* está situada en la parte superior de la ventana. En ella se agrupan los controles generales de la instalación. A continuación, se detalla el listado correspondiente a esta instalación.



Pulsador Salir

Pulsándolo podrá cerrar la instalación.

Tras el cierre de la instalación se activan de nuevo todos los pulsadores/interruptores del PLC y el menú *Instalaciones* de PC-Sim.

El cierre de PC-Sim provoca automáticamente el cierre de la instalación abierta.



Pulsador Guía

Carga la guía de la instalación en uso.

Si bien las instalaciones de PL-Sim se han estudiado para ser fáciles de usar y para comprenderse rápida e intuitivamente, utilice la guía a menudo: le será muy útil al principio de sus prácticas.



Pulsador Control de audio

Activa o desactiva la emisión de sonidos a través de la tarjeta de audio. Le recordamos que puede escuchar los sonidos tan sólo si su ordenador tiene instalada una tarjeta de audio y los controladores de audio de Windows se han cargado correctamente. Para informaciones sobre cómo instalar los controladores, consulte su manual de Windows.

Modalidades de empleo

Con esta instalación pueden simularse todas las posibilidades de funcionamiento de un típico ascensor. Se puede empezar con procesos simples y tradicionales y llegar hasta programas más complejos que realizan las funciones de reserva de llamada y de retorno a un piso, con optimización de las paradas. Como siempre, todo se deja en manos de la inventiva del usuario de AW-SYS.

Encontrará ejemplos de programación para esta instalación en la carpeta Ejemplos del PLC que utilice. La descripción de los procedimientos de carga y de test se detallan en la *Guía on line* de PC-Sim.

Conexiones con el PLC

La tabla siguiente enumera los componentes de la instalación que pueden conectarse al PLC, clasificados por entradas y salidas.

Entradas digitales (16)	Salidas digitales (12)
Pulsador de envío al piso 0	Contactador K1 subida cabina
Pulsador de envío al piso 1	Contactador K2 bajada cabina
Pulsador de envío al piso 2	Relé K3 apertura de puertas
Pulsador de envío al piso 3	Relé K4 cierre de puertas
Pulsador Stop	Luz de cabina piso 0
Pulsador de llamada al piso 0	Luz de cabina piso 1
Pulsador de llamada al piso 1	Luz de cabina piso 2
Pulsador de llamada al piso 2	Luz de cabina piso 3
Pulsador de llamada al piso 3	Luz piso 0
Final de carrera del piso 0 S0	Luz piso 1
Final de carrera del piso 1 S1	Luz piso 2
Final de carrera del piso 2 S2	Luz piso 3
Final de carrera del piso 3 S3	
Final de carrera de cierre de puertas S4	
Célula fotoeléctrica puertas S5	
Sensor de presencia S6	

Las conexiones de los sensores y de los actuadores al PLC se efectúan per medio de los correspondientes Selectores situados junto a cada componente conectable. El texto que los acompaña indica la ruta del canal de entrada o de salida del PLC al cual está conectado dicho elemento de la instalación. En azul se muestran las conexiones a entradas del PLC y en rojo las conexiones a las salidas.

 selector de ruta del canal de entrada (caracteres en azul);

 selector de ruta del canal de salida (caracteres en rojo).

Para modificar una ruta de salida o de entrada:

- haga clic sobre el selector de la conexión a modificar y mantenga la presión;

aparece el listado de las entradas o salidas en la escritura propia del PLC que esté utilizando;

- moviendo el mouse sitúese sobre la ruta preseleccionada;
- suelte el botón del mouse.

O bien:

- haga clic sobre los pulsadores con un triángulo, que están situados a la izquierda del selector: Hacia arriba, para obtener el incremento de la ruta y hacia abajo para que disminuya.

Si desea desconectar un elemento de instalación sitúe el selector de una de las dos maneras que acabamos de ver, sobre 'NC'.

De todos modos, cuando cargue la instalación le será propuesto un set de conexiones predeterminadas. Podrá utilizarlo o cambiarlo pero, en este último caso, recuerde que no se podrán guardar las conexiones modificadas. Dicho de otro modo, cada vez que inicie la ventana de esta instalación se le propondrán de nuevo las conexiones predeterminadas..

A diferencia de las cinco primeras instalaciones de PL-Sim, ésta no controla la univocidad de la conexión a una entrada del PLC. Por tanto, preste mucha atención a evitar esta eventualidad. En cualquier caso, si conecta dos sensores a la misma entrada del PLC, estos realizarán una conexión lógica OR, es decir, sus contactos se conectarán en paralelo; o sea, será necesario y suficiente que uno de los dos esté cerrado para obtener el cierre de la entrada del PLC.

Test de la automatización

Para verificar la corrección del programa realizado ponga en Run el PLC y verifique que los movimientos de la cabina del ascensor se ejecutan siguiendo la secuencia prefijada.

Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si la automatización no funciona correctamente, o no funciona en absoluto, le podrán ser de mucha utilidad las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Guía rápida

Si desea obtener informaciones relacionadas con un elemento concreto de la instalación, opere de la siguiente manera:

- sitúe el puntero del mouse sobre el elemento;
- pulse la combinación de teclas Ctrl+H;

en ese momento aparecerá, si está previsto para ese elemento en concreto, una ventana de guía rápida que lo describe brevemente.

La guía rápida es sensible al contexto, es decir, al desplazar el puntero del mouse sobre otro elemento, la descripción que aparece en la ventana se actualizará en función del nuevo elemento señalado.

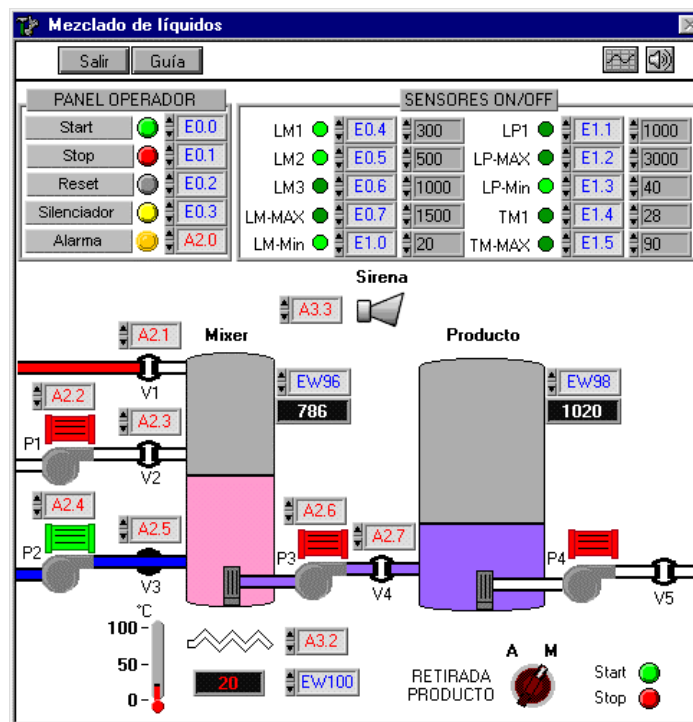
Si desea fijar el contenido de la ventana haciendo que sea independiente de la posición del puntero, pulse la combinación de teclas Ctrl+G. Para hacer que de nuevo sea variable la descripción de la ventana en función del contexto, vuelva a pulsar Ctrl+G.

Por último si pulsa de nuevo la combinación de teclas Ctrl+H, la ventana de guía rápida se cerrará.

Instalación n.8: Mezclado de líquidos

Descripción

Esta instalación de PL-Sim simula una parte de un establecimiento industrial que tiene la misión de producir y distribuir un compuesto que, en su ciclo productivo, está destinado a ser utilizado en otro sitio. El escenario tiene, como elementos principales, dos depósitos. En el primero, llamado *Mixer*, se mezclan y calientan tres líquidos distintos que constituyen los ingredientes de la fórmula. El segundo, denominado *Producto*, se utiliza para el almacenamiento del compuesto.



El *Mixer* está alimentado por tres tuberías distintas, cada una de las cuales vierte en su interior un ingrediente diferente. Las válvulas y las bombas situadas en estas tuberías, además de los indicadores de nivel y el transmisor de nivel que equipa el depósito, permiten una dosificación cuidadosa de los componentes. Estos tres ingredientes son, respectivamente, de color rojo, verde y azul. La mezcla adquiere un color que está en función del porcentaje entre los distintos líquidos que la componen, según las leyes de la colorimetría. Por ejemplo, si se vierten cantidades iguales de rojo y de verde en el depósito, la mezcla resultante será de un color amarillo.

Con la activación del calentador y la verificación del estado del termostato, o con la lectura del valor del transmisor de temperatura después, el compuesto podrá calentarse hasta la temperatura deseada.

Una pareja bomba-válvula permite el trasvase del producto acabado desde el primero al segundo depósito, donde será almacenado. La pareja bomba-válvula, a la salida del depósito del producto, sirve para alimentar otras instalaciones de la fábrica que lo utilizan, que aquí no están representadas y que forman parte de la

automatización. En el depósito *Producto* deberá quedar siempre una cantidad de compuesto suficiente para cubrir las necesidades, hasta que se complete un nuevo ciclo de producción.

En la parte superior de la ventana aparece un panel operador en el que están reagrupados los botones y pilotos para el control de la instalación y la señalización de su estado.

Un último recuadro reagrupa las señalizaciones de estado y las regulaciones de los sensores ON-OFF instalados en los depósitos.

Válvulas

La válvula es un dispositivo por medio del cual puede controlarse una corriente de fluido. En la instalación hay cinco válvulas, todas del tipo ON-OFF. Las tres primeras están destinadas al control del flujo de los ingredientes en el depósito *Mixer*. La cuarta válvula controla el trasvase de la mezcla al depósito de almacenaje. La última controla la extracción del producto acabado del depósito de almacenaje.

Electrobombas

Cuando se acciona una electrobomba se puede impulsar un fluido a través de una conducción. En la instalación hay cuatro electrobombas: dos a la entrada del *Mixer*, una entre los dos depósitos y la última en la salida del depósito de almacenaje.

Calentador

El encendido del calentador eléctrico permite calentar la mezcla en el depósito *Mixer*. Una vez se haya apagado, la temperatura del líquido disminuirá gradualmente a causa de las dispersiones térmicas. También la entrada de nuevo líquido, que se supone a 20 °C, produce una bajada de la temperatura en el depósito.

Sirena

La instalación dispone de una sirena que puede activarse en caso de alarma. Esta admite dos representaciones gráficas:



sirena apagada con salida del PLC a cero



sirena activada con salida del PLC a uno.

Si en su PC tiene instalada una tarjeta de audio podrá oír el sonido. Si no es así, podrá orientarse con su representación gráfica.

Sensores ON/OFF

La instalación está dotada de sensores ON/OFF. Parte de estos sensores pueden utilizarse como alternativa a los transmisores de nivel y de temperatura presentes en la instalación y que están conectados a los canales de entrada analógicos del PLC. Esto ofrece la posibilidad de elegir si efectuar la automatización de la instalación utilizando los sensores digitales o bien los analógicos.

En la instalación están presentes los siguientes sensores ON/OFF:

Sensor	Descripción
LM1	1º Indicador del nivel de servicio del depósito <i>Mixer</i> . Se activa cuando se alcanza la cantidad deseada del primer ingrediente.
LM2	2º Indicador de nivel de servicio del depósito <i>Mixer</i> . Se activa cuando se alcanza la cantidad deseada del segundo ingrediente.
LM3	3º Indicador de nivel de servicio del depósito <i>Mixer</i> . Se activa cuando se alcanza la cantidad deseada del tercer ingrediente.
LM-MAX	Indicador de nivel máximo del depósito <i>Mixer</i> . Generación de una alarma.
LM-Min	Indicador de nivel mínimo del depósito <i>Mixer</i> . Generación de una alarma.
LP1	Indicador de nivel de servicio del depósito <i>Producto</i> . Se activa para requerir el inicio de un nuevo ciclo de producción.
LP-MAX	Indicador de nivel máximo del depósito <i>Producto</i> . Generación de una alarma.
LP-Min	Indicador de nivel mínimo del depósito <i>Producto</i> . Generación de una alarma.
T1	termostato del depósito <i>Mixer</i> , activación cuando se alcanza la temperatura de mezclado.
T1-MAX	termostato de máxima temperatura del depósito <i>Mixer</i> . Generación de una alarma.

La programación de los niveles de intervención de los indicadores de nivel LM1, LM2 y LM3 debe ser progresiva. Es decir, supongamos que se desea producir una mezcla según una fórmula que esté compuesta por 500 l del primer ingrediente, 300 l del segundo y 200 l del tercero. El Indicador de nivel LM1 deberá estar programado para intervenir a los 500 l, LM2 a 800 l (500+300) y LM3 a 1000 l (500+300+200).

Cada sensor está dotado de una histéresis del orden del 2% del campo de regulación, es decir:

Sensor	Histéresis
Indicador de nivel del depósito Mixer	±15 litros
Indicador de nivel del depósito Producto	±30 litros
Termostato del depósito Mixer	±1°C

Transductores

El transductor es un aparato capaz de transformar una magnitud física en una magnitud eléctrica. En la instalación están presentes un transductor de temperatura, que mide la temperatura de la mezcla presente en el depósito *Mixer*, y dos transductores de nivel para la medida del nivel de los dos depósitos.

Cada uno de ellos emite una señal analógica proporcional a la magnitud medida. Esta señal puede ser captada por los PLC de PC-Sim que admiten canales analógicos (PS3 y S5 100U).

Extracción del producto

Se puede intervenir sobre la modalidad de extracción del producto acabado actuando sobre el correspondiente selector.

En la modalidad A se puede confiar la extracción del producto al PLC o bien seleccionar una modalidad de extracción aleatoria.

Con la modalidad M se puede controlar manualmente la extracción del producto.

Panel operador

El panel operador dispone de cuatro pulsadores para la conexión con el PLC, de los cuales le sugerimos una posibilidad de utilización:

Pulsador	Utilización
Start	puesta en marcha del proceso productivo
Stop	interrupción del proceso productivo o emergencia
Reset	reconocimiento de la alarma
Silenciador	silenciado de la sirena

Además, el panel dispone de un piloto luminoso que podrá utilizar para señalar una alarma que se haya producido durante el proceso productivo, por ejemplo una alarma de alta temperatura o de máximo nivel.

Convenciones para las entradas y las salidas digitales

Cada entrada o salida digital está provista de un piloto luminoso indicador de estado. El encendido del piloto indica la actuación de la entrada o la activación de la salida.

Barra de instrumentos

La *Barra de instrumentos* está situada en la parte superior de la ventana. En ella se reúnen los controles generales de la instalación. A continuación, se detalla el listado correspondiente a esta instalación.

Pulsador Salir

Pulsándolo podrá cerrar la instalación.

Tras el cierre de la instalación se activan de nuevo todos los pulsadores/interruptores del PLC y el menú *Instalaciones* de PC-Sim.

El cierre de PC-Sim provoca automáticamente el cierre de la instalación abierta.

Pulsador Guía

Carga la guía de la instalación en uso.

Si bien las instalaciones de PL-Sim se han estudiado para ser fáciles de usar y para comprenderse rápida e intuitivamente, utilice la guía a menudo: le será muy útil al principio de sus prácticas.

Pulsador Registradores

Muestra la simulación del registro en papel, que informa del proceso en el tiempo (el trend) de las señales de los transmisores de nivel y de temperatura.

Pulsador Control de audio

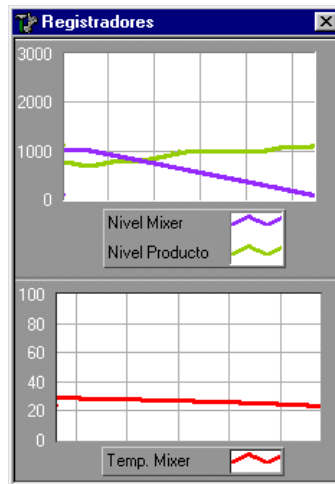
Activa o desactiva la emisión de sonidos mediante la tarjeta de audio. Le recordamos que puede escuchar los sonidos tan sólo si su ordenador dispone de una tarjeta de audio y los controladores de audio de Windows se han cargado correctamente. Para más información sobre cómo instalar los controladores, consulte su manual de Windows.

Ventana Registradores

Simula un componente muy común en las aplicaciones industriales: la registradora en papel.

En el recuadro superior aparece trazado el proceso de los niveles en los dos depósitos. El color del trazo es diferente para cada uno de ellos para así permitir el reconocimiento inmediato de la línea.

En el inferior se indica el avance de la temperatura en el depósito Mixer, entre los límites de 0 y 100°C.



Modalidades de empleo

Intente mezclar los tres ingredientes según las proporciones predeterminadas. Caliente la mezcla obtenida durante un cierto tiempo a una temperatura determinada y trasvásela al depósito *Producto*. Estamos seguros de que su fantasía les sugerirá otros posibles modos de empleo.

En sus primeras aplicaciones, no es preciso que utilicen todos los indicadores que están disponibles. Pueden usar tan sólo dos ingredientes, saltarse la fase de calentamiento y evitar la aparición de alarmas. Luego, poco a poco, podrán ir añadiendo estas funciones y otras más como, por ejemplo, la gestión de más recetas.


Encontrará más ejemplos de programación para esta instalación en la carpeta Ejemplos del PLC que esté utilizando. La descripción de los procedimientos de carga y de test se detalla en la *Guía on line* de PC-Sim.

Conexiones con el PLC

La tabla siguiente detalla los componentes de la instalación que pueden conectarse al PLC, clasificados por entradas y salidas.

Entradas digitales (14)	Salidas digitales (12)	Entradas analógicas (3)
Pulsador de Start	Alarma	Temperatura mezcla
Pulsador de Stop	Sirena	Nivel del depósito Mixer
Pulsador de Reset	Válvula 1° ingrediente	Nivel del depósito Producto
Pulsador de Silenciado	Electrobomba 2° ingrediente	
1° Indicador de nivel – Mixer	Válvula 2° ingrediente	
2° Indicador de nivel – Mixer	Electrobomba 3° ingrediente	
3° Indicador de nivel – Mixer	Válvula 3° ingrediente	
Indicador de nivel máximo - Mixer	Electrobomba extracción mezcla	
Indicador de nivel mínimo – Mixer	Válvula extracción mezcla	
Indicador de nivel – Producto	Electrobomba extracción producto	
Indicador de nivel de máximo – Producto	Válvula extracción producto	
Indicador de nivel de mínimo – Producto	Calentador	
Termostato		
Termostato de máximo		

Las conexiones de los sensores y de los actuadores al PLC se efectúan por medio de los correspondientes selectores situados junto a cada componente conectable. El texto que los acompaña indica la ruta del canal de entrada o de salida del PLC al cual está conectado dicho elemento de la instalación. En azul se muestran las conexiones a entradas del PLC y en rojo las conexiones a las salidas.

 selector de ruta del canal de entrada (caracteres en azul);

 selector de ruta del canal de salida (caracteres en rojo).

Para modificar una ruta de salida o de entrada:

- haga clic sobre el selector de la conexión a modificar y mantenga la presión; aparece el listado de las entradas o salidas en la escritura propia del PLC que esté utilizando;
- moviendo el mouse sitúese sobre la ruta preseleccionada;
- suelte el botón del mouse.

O bien:

- haga clic sobre los pulsadores con un triángulo, que están situados a la izquierda del selector: Hacia arriba, para obtener el incremento de la ruta y hacia abajo para que disminuya.

Si desea desconectar un elemento de instalación sitúe el selector, de una de las dos maneras que acabamos de ver, sobre 'NC'.

De todos modos, cuando cargue la instalación le será propuesto un juego de conexiones predeterminadas. Podrá utilizarlo o cambiarlo pero, en este último caso, recuerde que no se podrán guardar las conexiones modificadas. Dicho de otro modo, cada vez que inicie la ventana de esta instalación se le propondrán de nuevo las conexiones predeterminadas..

A diferencia de las cinco primeras instalaciones de PL-Sim, ésta no controla la univocidad de la conexión a una entrada del PLC. Por tanto, preste mucha atención a evitar esta eventualidad. En cualquier caso, si conecta dos sensores a la misma entrada del PLC, estos realizarán una conexión lógica OR, es decir, sus contactos se conectarán en paralelo; o sea, será necesario y suficiente que uno de los dos esté cerrado para obtener el cierre de la entrada del PLC.

Test de la automatización

Para verificar la corrección del programa realizado ponga en Run el PLC y compruebe que el proceso se realiza siguiendo los pasos prefijados.

Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si la automatización no funciona correctamente, o no funciona en absoluto, le podrán ser de mucha utilidad las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Guía rápida

Si desea obtener informaciones relacionadas con un elemento determinado de la instalación, opere de la siguiente manera:

- sitúe el puntero del mouse sobre el elemento;
- pulse la combinación de teclas Ctrl+H;

en ese momento aparecerá, si está previsto para ese elemento en concreto, una ventana de guía rápida que lo describe brevemente.

La guía rápida es sensible al contexto, es decir, al desplazar el puntero del mouse sobre otro elemento, la descripción que aparece en la ventana se actualizará en función del nuevo elemento señalado.

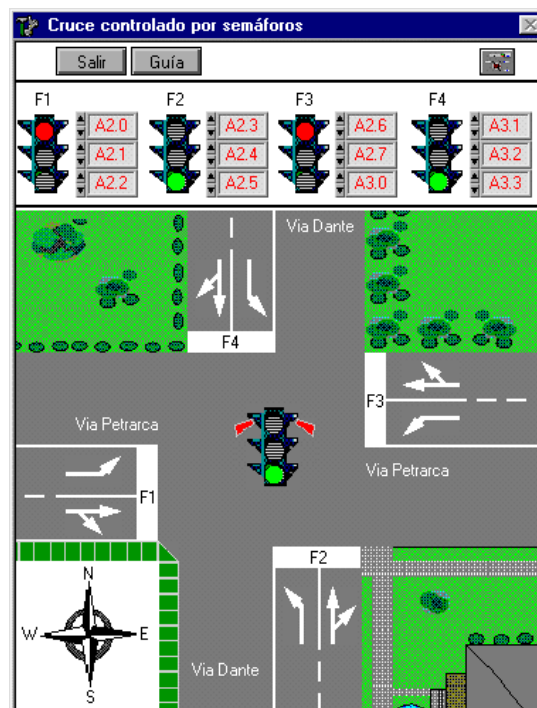
Si desea fijar el contenido de la ventana haciendo que sea independiente de la posición del puntero, pulse la combinación de teclas Ctrl+G. Para hacer que de nuevo sea variable la descripción de la ventana en función del contexto, vuelva a pulsar Ctrl+G.

Por último si pulsa de nuevo la combinación de teclas Ctrl+H, la ventana de guía rápida se cerrará

Instalación n.9: Cruce semaforizado

Descripción

Esta ventana de simulación le propone un escenario constituido por un cruce (o una intersección) con su correspondiente instalación semafórica. Aunque hoy en día el semáforo ya forme parte de nuestra vida cotidiana, ya sea como peatones o como automovilistas, y que sin duda no sería necesario explicar de qué se trata, nos parece oportuno proponer consideraciones de orden general que servirán para encuadrar mejor el problema de la instalación de semáforos.



Las instalaciones semafóricas desarrollan la función de regulación del tráfico en las intersecciones viales. Se utilizan tanto en los centros urbanos, donde constituyen un instrumentos obligatorio en la regulación del tráfico, como en las arterias extraurbanas.

Estas constituidos por:

- una instalación de señalización;
- una centralita semafórica.

Instalación de la señalización

Está constituido por un semáforo a cuatro caras. Las caras pueden verse de una en una, en la parte superior de la ventana y están numeradas de F1 a F4.

- F1 identifica el semáforo correspondiente a la calle Petrarca lado oeste.
- F2 identifica el semáforo correspondiente a la calle Dante lado sur
- F3 identifica el semáforo correspondiente a la calle Petrarca lado este
- F4 identifica el semáforo correspondiente a la calle Dante lado norte.

Centralita semafórica

La centralita semafórica está constituida por el PLC simulado de PC-Sim y por un *Panel de Control* visualizado en la correspondiente ventana y de la cual hablaremos más adelante.

Características del cruce

La ventana de simulación muestra el plano del cruce, los diferentes carriles y la señalización horizontal que marca el flujo de vehículos. Aunque se facilita una configuración predeterminada, esta señalización puede variarse a gusto del usuario: sitúe el mouse sobre de una señal y pulse el botón izquierdo; a continuación, seleccione un nuevo tipo de señalización y suelte el botón del mouse. Naturalmente, procure seleccionar configuraciones que sean coherentes con las normas de circulación viaria.

Fases y temporizaciones

Para el análisis del problema y el siguiente desarrollo del software le aconsejamos que subdivida el movimiento de los vehículos en varias fases: a cada fase le corresponde el movimiento de unas determinadas corrientes del tráfico. Construya un *esquema de fases* en el que cada fase vaya asociada a una indicación de las corrientes de tránsito. A continuación, construya un *diagrama de fases* que muestre el esquema de las temporizaciones de la señal luminosa, que irá asociada a cada una de las caras del semáforo que controla el movimiento de vehículos.

Barra de instrumentos

La *Barra de instrumentos* está situada en la parte superior de la ventana. En ella se reúnen los controles generales de la instalación. A continuación, se detalla el listado correspondiente a esta instalación.

 **Pulsador Salir**

Pulsándolo podrá cerrar la instalación.

Tras el cierre de la instalación se activan de nuevo todos los pulsadores/interruptores del PLC y el menú *Instalaciones* de PC-Sim.

El cierre de PC-Sim provoca automáticamente el cierre de la instalación abierta.

Guía

Pulsador Guía

Carga la guía de la instalación en uso.

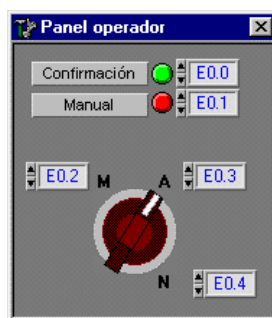
Si bien las instalaciones de PL-Sim se han estudiado para ser fáciles de usar y para comprenderse rápida e intuitivamente, utilice la guía a menudo: le será muy útil al principio de sus prácticas.



Pulsador Panel de control

Abre una ventana que contiene la simulación del *Panel de control* de la centralita semafórica.

Ventana Panel de control



Contiene el *Panel de control* de la centralita semafórica. El *Panel de control* está constituido por:

- pulsador *Confirmación*
- pulsador *Manual*
- selector con las tres siguientes posiciones:
 - M -> Manual
 - A -> Automático
 - N -> Nocturno

Modos de funcionamiento

La instalación semafórica puede funcionar de tres maneras, que pueden seleccionarse manualmente con el selector situado en el *Panel de control*. Situando el selector sobre la función deseada y accionando el pulsador *Confirmar* se activa la modalidad preseleccionada.

Modo de funcionamiento automático:

- Selector en posición **A**;
- el PLC ejecuta secuencialmente las fases semafóricas respetando las temporizaciones indicadas en el diagrama de fases.

Modo de funcionamiento manual:

- Selector en posición **M**;
- el PLC ejecuta secuencialmente las fases semafóricas. El paso de una fase a la siguiente se activa exclusivamente por medio de la presión sobre el pulsador *Manual* efectuada por usted.

Modo de funcionamiento nocturno:

- Selector en posición **N**;
- el PLC fuerza el estado de *ámbar intermitente* en todos los carriles del tráfico.

Esquema de fases

Las figuras que aparecen más abajo muestran las dos fases en las que se ha subdividido el ciclo. A cada fase corresponde el movimiento de una determinada corriente de tráfico seleccionada de manera que pueda limitar al máximo las situaciones conflictivas. Naturalmente, el que se muestra aquí es el ejemplo más sencillo: para limitar aun más las situaciones conflictivas, puede intentar aumentar el número de fases.

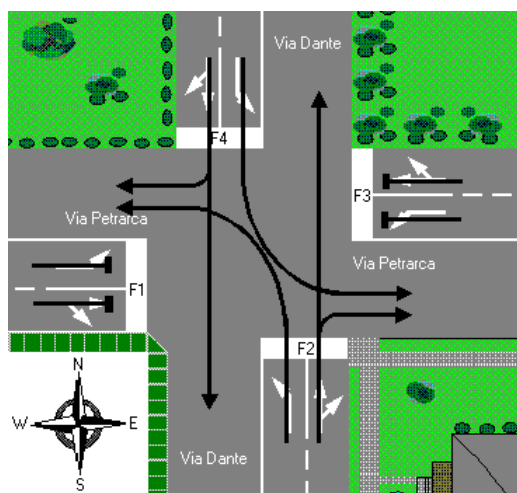
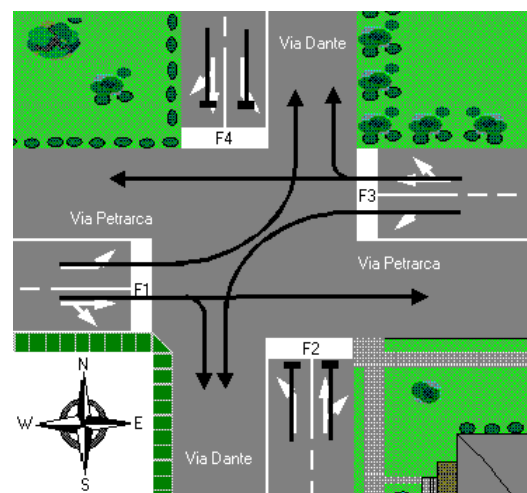
**Fase A****Fase B**

Diagrama de fases

Las columnas representan las fases de movimiento, tal como se han definido en el esquema de fases. Las filas representan las caras del semáforo. En la última fila se detalla la duración de cada fase expresada en segundos.


Cara semafórica	Fases					
	A			B		
F1	Verde	Ambar	Rojo	Rojo		
F2	Rojo			Verde	Ambar	Rojo
F3	Verde	Ambar	Rojo	Rojo		
F4	Rojo			Verde	Ambar	Rojo
Tiempo en seg	80	6	5	80	6	5


Conexiones con el PLC

La tabla siguiente detalla los componentes de la instalación que pueden conectarse al PLC, subdivididos por entradas y salidas.

Entradas digitales (5)	Salidas digitales (12)
Pulsador <i>Confirmación</i>	Luz Roja F1
Pulsador <i>Manual</i>	Luz Ambar F1
Selector posición <i>M</i>	Luz Verde F1
Selector posición <i>A</i>	Luz Roja F2
Selector posición <i>N</i>	Luz Ambar F2
	Luz Verde F2
	Luz Roja F3
	Luz Ambar F3
	Luz Verde F3
	Luz Roja F4
	Luz Ambar F4
	Luz Verde F4

La conexiones de los sensores y de los actuadores al PLC se efectúan por medio de los correspondientes selectores situados junto a cada componente conectable. El texto que los acompaña indica la ruta del canal de entrada o de salida del PLC al cual está conectado dicho elemento de la instalación. En azul se muestran las conexiones a entradas del PLC y en rojo las conexiones a las salidas

 **I0.13** selector ruta de entrada (caracteres en azul);

 **Q0.11** selector ruta de salida (caracteres en rojo).

Para modificar una ruta de salida o de entrada:

- haga clic sobre el selector de la conexión a modificar y mantenga la presión; aparece el listado de las entradas o salidas en la escritura propia del PLC que esté utilizando;
- moviendo el mouse sitúese sobre la ruta preseleccionada;
- suelte el botón del mouse.

O bien:

- haga clic sobre los pulsadores con un triángulo, que están situados a la izquierda del selector: Hacia arriba, para obtener el incremento de la ruta y hacia abajo para que disminuya.

Si desea desconectar un elemento de instalación sitúe el selector de una de las dos maneras que acabamos de ver, sobre 'NC'.

De todos modos, cuando carga la instalación se le propone un juego de conexiones predeterminadas. Puede utilizarlo o cambiarlo pero, en este último caso, recuerde que no se podrán guardar las conexiones modificadas. En otras palabras, cada vez que inicie la ventana de esta instalación se le propondrán de nuevo las conexiones predeterminadas.

A diferencia de las cinco primeras instalaciones de PL-Sim, ésta no controla la univocidad de la conexión a una entrada del PLC. Por tanto, preste mucha atención a evitar esta eventualidad. En cualquier caso, si conecta dos sensores a la misma entrada del PLC, estos realizarán una conexión lógica OR, es decir, sus contactos se conectarán en paralelo; o sea, será necesario y suficiente que uno de los dos esté cerrado para obtener el cierre de la entrada del PLC.

Test de la automatización

Para verificar la corrección del programa realizado ponga en Run el PLC y compruebe que el proceso se realiza siguiendo la secuencia predeterminada.

Si ha utilizado merkers (flags), temporizadores o contadores en su programa, le podrá ser útil la [Ventana Estado variables](#) de PC-Sim para controlar su estado y/o valor. Si la automatización no funciona correctamente, o no funciona en absoluto, le podrán ser de mucha utilidad las potencialidades de debug que le ofrece el editor que haya utilizado para escribir el programa.

Si utiliza una simulación de PLC FPC 404 la [Ventana Asignación de preselectores \(FPC 404\)](#) le servirá para programar valores de temporización y de conteo.

Guía rápida

Si desea obtener informaciones relacionadas con un elemento determinado de la instalación, opere de la siguiente manera:

- sitúe el puntero del mouse sobre el elemento;
- pulse la combinación de teclas Ctrl+H;

en ese momento aparecerá, si está previsto para ese elemento en concreto, una ventana de guía rápida que lo describe brevemente.

La guía rápida es sensible al contexto, es decir, al desplazar el puntero del mouse sobre otro elemento, la descripción que aparece en la ventana se actualizará en función del nuevo elemento señalado.

Si desea fijar el contenido de la ventana haciendo que sea independiente de la posición del puntero, pulse la combinación de teclas Ctrl+G. Para hacer que de nuevo sea variable la descripción de la ventana en función del contexto, vuelva a pulsar Ctrl+G.

Por último si pulsa de nuevo la combinación de teclas Ctrl+H, la ventana de guía rápida se cerrará.

CAPÍTULO 5

PC-RIS: INTERFACE HACIA SISTEMAS DE CONTROL REALES

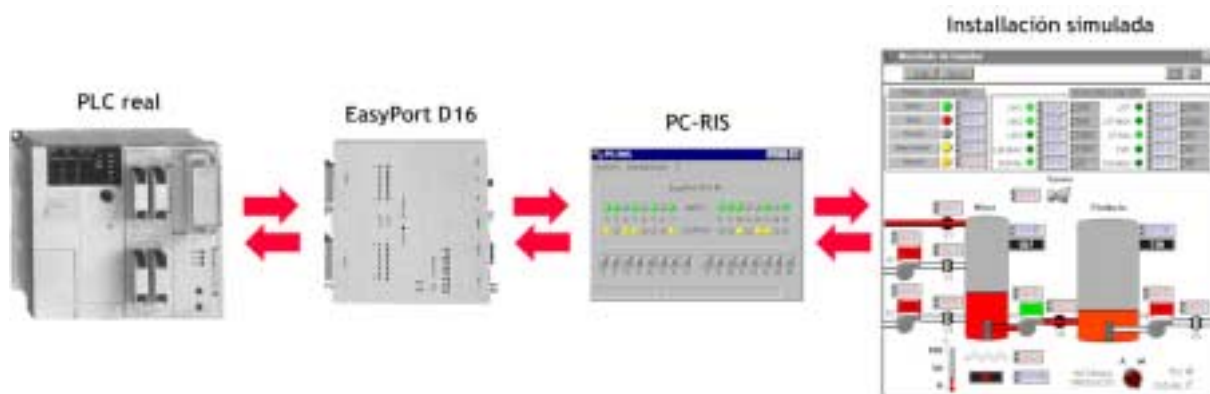
Introducción

Tal como ya se ha señalado en la introducción de este manual, PC-RIS es el módulo de AW-SYS que permite controlar las instalaciones simuladas de PL-Sim con múltiples y variados sistemas de hardware, tales como:

- PLC reales (de cualquier marca)
- sistemas de microprocesadores
- microcontroladores
- fichas electrónicas de lógica cableada
- cuadros eléctricos de lógica cableada
- ordenadores personales con tarjeta de I/O.

Todo ello es posible gracias a la utilización de la interface de hardware EasyPort D16. Una o más de estas interfaces pueden conectarse al PC, tal como se explicará a continuación. Además, por una parte, permitirán la adquisición de las salidas del sistema de control real para el accionamiento de los actuadores de la instalación simulada y, por otra, la transferencia del estado de los sensores de la instalación a las entradas del sistema de control, para un máximo de 16 entradas y 16 salidas digitales.

La figura que aparece a continuación ilustra el funcionamiento del sistema. De izquierda a derecha: las salidas del PLC real se adquieren por medio de la interface EasyPort D16. El PC-Ris detecta las señales de ésta última, a través del puerto en serie del PC, y las transmite a los actuadores de la instalación simulada.



En el otro sentido: el PC-Ris lee el estado de los sensores de la instalación a través del puerto serie y se encarga de transmitirlos a las salidas digitales de la interface. A partir de ahí, ésta controla las entradas del PLC.

Si no dispone de la interface EasyPort D16 no podrá utilizar PC-RIS. Diríjase a su distribuidor de AW-SYS si desea más información sobre cómo adquirir dicha interface.

En el curso de este capítulo nos referiremos al sistema de control real 'PLC', pero todo lo que se diga respecto a este sistema es igualmente válido para todos el resto de sistemas de control, teniendo presentes las necesarias comprobaciones sobre la compatibilidad entre señales de la interface y las señales del sistema de control.

Por qué usar PC-RIS

PC-RIS es un óptimo instrumento didáctico dado que le permite utilizar los PLC de los que ya disponga en su taller y en su sistema de desarrollo, ofreciéndole además la posibilidad de experimentar con la automatización de instalaciones que presentan diferentes grados de dificultad y que no requieren puesta a punto o mantenimiento. Todo ello le permitirá concentrar las energías de sus alumnos en el aprendizaje de los lenguajes y las técnicas de programación y a Ud. le liberará de tener que verificar los resultados de la programación a través tan sólo de la observación de los LED de las salidas del PLC.

Puesta en marcha del sistema

En este apartado se describen las operaciones que llevan a la puesta en marcha del sistema. Ejecútelas una tras otra y con atención.

Configuración de la interface

Como operación preliminar es preciso configurar los DIP switch que se encuentran en el panel posterior de la EasyPort D16, tal como sigue.

Número	Estado	Posición
1	ON	abajo
2	OFF	arriba
3	OFF	arriba

Conexión de la interface al PC

Con el cable que se proporciona con el equipo, conecte el puerto serie de la EasyPort D16 a uno de los puertos serie del PC. También pueden conectarse otras interfaces del mismo tipo a la primera mediante fibra óptica, pero el sistema no las verá y por lo tanto no las utilizará.

Conecte las regletas +/- 24V de la EasyPort D16 a un alimentador de 24V. Como alternativa, y con el mismo efecto, se puede proporcionar alimentación a través de los pins de PORT1 o PORT2 identificados con OVCC y 24VCC (ver [Apéndice A](#)) pero muy probablemente, será conveniente reservar a estos últimos para la alimentación de los módulos de entrada/salida del PLC.

Conexión del PLC real a la EasyPort D16

Conecte los dos cables multipolares, por el lado del conector “de bandeja”, a los correspondientes conectores de la EasyPort D16 identificados como PORT1 y PORT2.

A continuación, conecte los conductores de las otras dos extremidades de los cables a las entradas y salidas del PLC. La [Tabla 23](#), indicada en [Apéndice A](#), permite la identificación de los conductores en relación con su color y con la puerta a la que pertenecen.

Al realizar la conexión, observe las normas siguientes:

- Controle la compatibilidad entre las especificaciones de tensión para las entradas y salidas del PLC y de la EasyPort D16. Remítase al manual del PLC real de que dispone y a la [Tabla 24](#) en [Apéndice A](#) que incluye los datos técnicos de la EasyPort D16.
- Los conductores de salida de la EasyPort D16 se conectan a las entradas del PLC.
- Los conductores de entrada de la EasyPort D16 se conectan a las salidas del PLC.
- Para evitar confusiones en las fases siguientes, efectúe las conexiones de manera progresiva, siguiendo la numeración de las entradas/salidas del PLC y de la EasyPort D16.

Puesta en marcha de PC-RIS

Para poner en marcha PC-RIS haga clic sobre la tecla Inicio de Windows; a continuación abra la carpeta Programas\Aw-sys y, en su interior, seleccione la opción correspondiente a PC-RIS. Tan sólo puede ejecutarse una instancia de PC-RIS en cada ocasión. Además, tampoco es posible poner en marcha PC-RIS cuando PC-Sim ya se está ejecutando.

Al poner en marcha el programa aparece la *Ventana de Informaciones sobre PC-RIS*. En ella aparecen diversos datos del programa, su número de versión, el número de autorización de la copia que está utilizando y el productor del software. Cierre esta ventana pulsando Ok para empezar a trabajar.

Búsqueda de la interface

En ese momento PC-RIS ejecuta una exploración de los puertos serie del PC para buscar la EasyPort D16.

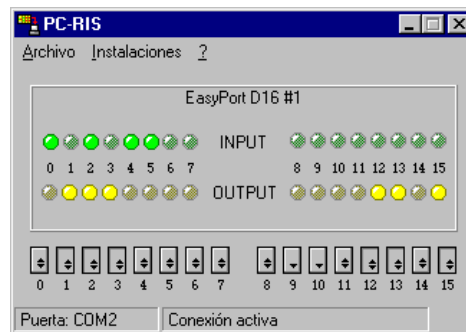
Durante todo este proceso, en la *Barra de estado* aparecerán indicaciones sobre el avance de la búsqueda. En concreto, la casilla de la izquierda indicará el número del puerto serie que se está verificando en ese momento y la otra mostrará el estado del test.

Al final de la búsqueda, si se ha encontrado una EasyPort D16, la primera casilla indicará el puerto al que está conectada y en la segunda aparecerá el texto 'Conexión activa'.

Por el contrario, en el caso de que no se localice la interface, la primera casilla quedará vacía mientras que la segunda mostrará el texto 'EasyPort D16 no encontrada'. En esta coyuntura, una vez haya verificado que todas las conexiones son correctas, repita el proceso de búsqueda seleccionando el *Comando Buscar EasyPort D16* en el *Menú Archivo*.



Composición de la ventana

PC-RIS está compuesto por una única ventana. La parte superior reproduce, en disposición y en color, la visualización de los indicadores luminosos de la EasyPort D16. El encendido de uno de los LED OUTPUT (color amarillo) coincide con la activación de la correspondiente salida en la interface, mientras que la activación de una de las entradas de la interface coincide con el encendido del correspondiente LED INPUT (color verde). En cada momento, si la conexión serie está activada, el estado de los pilotos luminosos en la interface coincidirá con el estado de los pilotos correspondientes en la ventana de PC-RIS.



En la parte inferior de la ventana están representados 16 pulsadores/interruptores. Estos podrán utilizarse para el control de las entradas digitales del PLC real que no estén conectados a la instalación.

Estos pulsadores/interruptores pueden programarse indistintamente como pulsadores o como interruptores. Su aspecto especificará su función:

-  interruptor
-  pulsador

donde una doble flecha indica la función biestable y una sola flecha la función monoestable.

Para cambiar de una función a la otra, haga clic con el botón derecho del mouse sobre el elemento que desee cambiar. Esto conmutará de función a cada clic: de pulsador pasará a interruptor y viceversa.

Cuando el pulsador/interruptor asume este aspecto:

-  desactivado

significa que está desactivado: hacer clic sobre él no producirá ningún efecto.

PC-RIS desactiva automáticamente un interruptor o un pulsador cuando a la entrada correspondiente se conecta el sensor de una instalación (por ejemplo el final de carrera de un cilindro o el indicador de nivel de un depósito) para evitar interferencias inoportunas entre los contactos de este último y los pulsadores/interruptores locales.

Para habilitar de nuevo el pulsador/interruptor será preciso desconectar la entrada del PLC de la instalación. Para más información al respecto, ver el apartado [Conectar las instalaciones al PLC](#) en el [Capítulo 4](#).

Apertura de una instalación

Las instalaciones simuladas de PL-Sim deben abrirse desde el *Menú Instalaciones* de PC-RIS: para ello, basta con seleccionar, entre los que aparecen, el comando con el nombre de la instalación que se desea iniciar. Dado que tan sólo puede abrirse una instalación cada vez, el menú se desactiva inmediatamente después de haberla abierto. En el momento en que se cierre la instalación en cuestión, el menú volverá a activarse.

Para la descripción de las diferentes instalaciones, ver el [Capítulo 4](#) dedicado a ellas.

Conexión de la instalación a PC-RIS

Como ya se ha descrito en el capítulo dedicado a PL-Sim, algunas de las instalaciones simuladas muestran, al abrirse, un juego de conexiones preestablecidas, mientras que otras (las cinco primeras) no lo hacen.

Si la instalación a utilizar pertenece a este último grupo, para efectuar una conexión opere tal como se describe a continuación.

- Haga clic sobre el componente a conectar.

La instalación responde mostrando la correspondiente ventana de diálogo *Seleccionar conexión* en función del componente que se ha seleccionado: entrada para los sensores, salida para los actuadores.



- Seleccione el canal del PLC al cual va a conectar el componente.
- Solo para las entradas, seleccione el tipo de contacto (NA o NC) a utilizar.
- Haga clic sobre el pulsador *Ok*.

Dado que PC-RIS no es capaz de determinar qué PLC se ha conectado a la interface y qué conexiones ha efectuado, los canales de entrada se denominarán con siglas genéricas, entre **I0** a **I15**, mientras que los de salida lo harán con las siglas, igualmente genéricas, entre **O0** a **O15**.

Por ejemplo, conectar un sensor de la instalación simulada a la entrada **I0**, significa conectar el sensor a la entrada del PLC que esté eléctricamente conectado con la salida OUTPUT 0 de la EasyPort D16, es decir, con el conductor blanco del cable multipolar de la PORT 1. Del mismo modo, y también como ejemplo, conectar un actuador de la instalación simulada a la salida **O15** significa conectar el actuador a la salida del PLC que está conectada eléctricamente con la entrada INPUT 15 de la EasyPort D16, es decir, con el conductor gris/marrón del cable multipolar del PORT 2.

Una vez conectado un sensor ON/OFF de la instalación, su estado estará inmediatamente disponible en la correspondiente salida de la interface y en la entrada del PLC conectada a ella. El cierre del contacto del sensor corresponderá a la emisión de una tensión de 24V en la salida de la EasyPort y por tanto a la activación de la entrada del PLC. La apertura de ésta corresponderá a la emisión de una tensión de 0V y por consiguiente a la desactivación de la entrada del PLC.

En el otro sentido: la activación de una de las salidas del PLC proporciona una tensión de 24V a la entrada de la EasyPort D16 correspondiente, que permitirá la activación del actuador de la instalación conectado a éste. La desactivación de esa misma salida proporcionará una tensión de 0V a la entrada de la interface y el actuador de la instalación se desactivará.

Continuación del ejercicio

La continuación del ejercicio consistirá en programar o cablear el sistema de control que haya decidido utilizar. Así pues, para ello deberá remitirse a los manuales que se facilitan junto con esos sistemas. Sin embargo, para más información sobre las instalaciones simuladas consulte el [Capítulo 4](#) de este manual o bien la *Guía en línea* de cada instalación concreta abriendo el *Comando Guía* del *Menú ?* o bien haciendo clic sobre el pulsador *Guía*, en función de la instalación que esté utilizando.

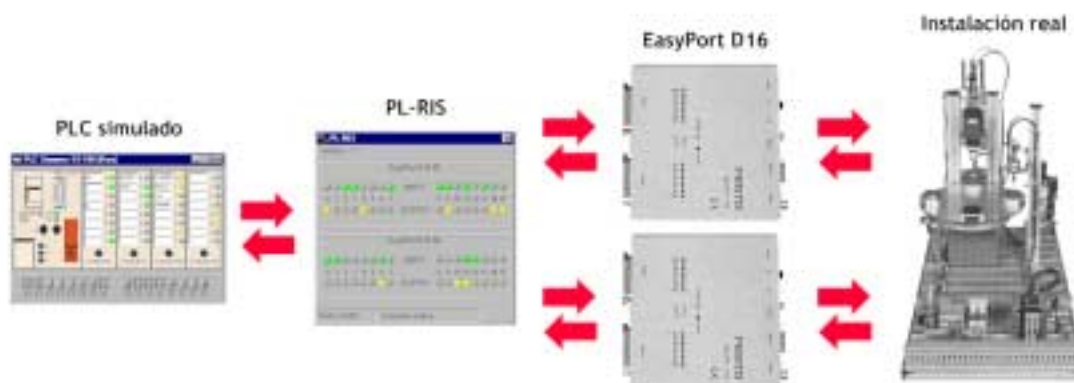
CAPÍTULO 6

PL-RIS: INTERFACE HACIA INSTALACIONES REALES

Introducción

PL-RIS es el módulo de AW-SYS que permite conectar los PLC simulados de PC-Sim a las instalaciones reales o a los bancos de ejercicios presentes en su taller.

La figura que sigue ilustra el funcionamiento del sistema. De izquierda a derecha: PL-RIS lee las salidas del PLC simulado y se encarga de enviarlas a través de la conexión en serie y un eventual anillo de fibra óptica a las interfaces EasyPort D16. Las salidas de estas últimas irán a pilotar los actuadores de la instalación real. En el otro sentido: el estado de los sensores de la instalación se adquiere por medio de las interfaces y a continuación PL-RIS efectúa la lectura. Este último se encarga de pilotar las entradas del PLC.



Si no dispone de la interface EasyPort D16 no podrá utilizar PC-RIS. Diríjase a su distribuidor de AW-SYS si desea más información sobre cómo adquirir dicha interface.

Puesta en funcionamiento del sistema

En este apartado se describen las operaciones que conducen a la puesta en funcionamiento del sistema. Ejecútelas una tras otra y con atención.

Configuración de la interface

Como operación preliminar es preciso configurar los DIP switch que se encuentran en el panel posterior de la EasyPort D16, tal como sigue.

Prima EasyPort D16	Número	Estado	Posición
	1	OFF	arriba
	2	ON	abajo
	3	OFF	arriba
EasyPort D16 siguientes	Número	Estado	Posición
	1	OFF	arriba
	2	OFF	arriba
	3	ON	abajo

Conexión de la interface al PC

Con el cable que se proporciona con el equipo, conecte el puerto en serie de la primera EasyPort D16 a uno de los puertos serie de su ordenador. Las otras interface que puedan haber, con un máximo de siete, se conectan a la primera, siguiendo las instrucciones que se facilitan en [Apéndice A](#). El sistema las verá (detectará) a todas y las utilizará dentro de los límites que cada PLC simulado impone, tal como se aclarará a continuación.

Conecte las regletas +/- 24V de la EasyPort D16 a un alimentador de 24V (no incluido). Como alternativa, y con el mismo efecto, se puede proporcionar alimentación a través de los pins de PORT1 o PORT2 identificados con OVCC y 24VCC (ver [Apéndice A](#)) pero muy probablemente, será conveniente reservar a estos últimos para la alimentación de los módulos de entrada/salida del PLC.

Conexión de la instalación real a la EasyPort D16

Conecte los dos cables multipolares, por el lado del conector “de bandeja”, a los correspondientes conectores de la EasyPort D16 identificados como PORT1 y PORT2.

A continuación, conecte los conductores de las otras dos extremidades de los cables a las entradas y salidas de la instalación real. La [Tabla 23](#), indicada en [Apéndice A](#), permite la identificación de los conductores en relación con su color y con la puerta a la que pertenecen.

Al realizar la conexión, observe las normas siguientes.

- Controle la compatibilidad entre las especificaciones de tensión y consumo para las entradas y las salidas de la instalación y de la EasyPort D16. Remítase a los diversos manuales de la instalación y de sus componentes y a la [Tabla 24](#) de [Apéndice A](#) que incluye los datos técnicos de la EasyPort D16.
- Los conductores de la salida de la EasyPort D16 están conectados a los actuadores de la instalación.
- Los conductores de entrada de la EasyPort D16 están conectados a los sensores de la instalación.

Puesta en marcha de PC-Sim

PL-RIS sólo puede funcionar en conexión con uno de los simuladores de PLC. Por ello, antes de ponerlo en marcha, es preciso iniciar PC-Sim.

Haga clic sobre la tecla Inicio de Windows, a continuación, abra la carpeta Programas\Aw-sys y, en su interior, seleccione la opción correspondiente al simulador de PLC que desee poner en marcha.

Puesta en marcha de PL-RIS

Para poner en marcha PL-RIS seleccione el [Comando PL-RIS](#) en el [Menú Instalaciones](#) de PC-Sim.

PC-Sim ve a PL-RIS como si fuera una de las instalaciones simuladas y por ello el [Menú Instalaciones](#) se desactiva inmediatamente después de que se haya puesto en marcha. Dicho menú no se activará otra vez hasta que se cierre PL-RIS, para impedir la apertura simultánea de otra instalación.

Búsqueda de la interface

Al ponerse en marcha, PL-RIS ejecuta una exploración de los puertos en serie del PC para buscar una EasyPort D16 conectada a ellos.

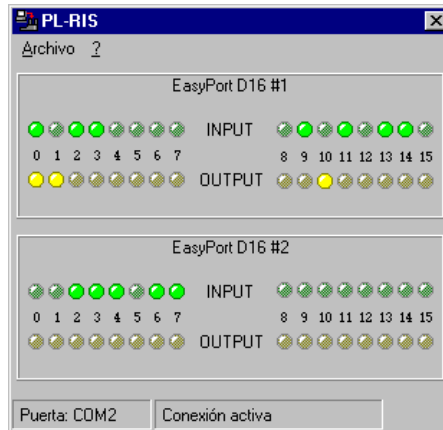
Durante todo este proceso, en la *Barra de estado* aparecerán indicaciones sobre el avance de la búsqueda. En concreto, la casilla de la izquierda indicará el número del puerto serie que se está verificando en ese momento y la otra mostrará el estado del test.

Al final de la búsqueda, si se ha encontrado una EasyPort D16, la primera casilla indicará el puerto en el que está conectada y en la segunda aparecerá el texto 'Conexión activa'.

Por el contrario, en el caso de que no se localice la interface, la primera casilla quedará vacía mientras que la segunda mostrará el texto 'EasyPort D16 no encontrada'. En esta coyuntura, una vez haya verificado todas las conexiones, vuelva a repetir el proceso de búsqueda seleccionando el [Comando Buscar EasyPort D16](#) en el [Menú Archivo](#).

Composición de la ventana

PL-RIS es capaz, no sólo de encontrar una EasyPort D16 conectada a uno de los puertos en serie del PC sino también de determinar si a ésta última hay otras conectadas mediante un anillo de fibra óptica y cuantas de ellas hay.



En la ventana de PL-RIS se visualizarán, numerados correlativamente, tantos paneles con pilotos luminosos como interfaces estén presentes. La ventana cambiará de dimensiones automáticamente para poder contener todos los paneles. Sin embargo, existe un límite superior para los paneles en función del tipo de PLC simulado. Si utiliza PC-Sim para FPC 404 o para PS3 el número máximo de paneles visualizables es de cuatro: tantos como las word de entrada/salida gestionables por estos PLC. Para la simulación del PLC S5 100U no existe este límite y por tanto el número máximo de paneles viene dado por el número máximo de EasyPort D16 conectables con anillo: ocho.

Direccionamiento de las entradas

El número de entradas del sistema depende, tal como ya se ha explicado, del número de las EasyPort D16 y del tipo de PLC simulado que se utilice, con un máximo de 64 para los PLC FPC 404 y PS3, y de 128 para el PLC S5 100U. Cada una de las entradas queda identificada de manera unívoca por dos términos:

- número de la EasyPort D16 a la que está conectada, con valores entre 1 y 8
- número del canal de entrada de esta última, con valores entre INPUT 0 e INPUT 15

El estado de las entradas de la EasyPort D16 puede adquirirse direccionando en el programa (ladder o AWL) las entradas del PLC de acuerdo con lo que se indica en la tabla siguiente. Preste atención al hecho que las entradas para los dos primeros PLC están en secuencia mientras que para el último esto no sucede, y existe una discontinuidad entre el byte 1 y el byte 8.

El estado de los primeros dieciséis bit de entrada será visible en la parte frontal del PLC simulado a través de las oportunas señales luminosas por medio de pilotos. En cambio, el estado de los siguientes tan sólo podrá valorarse desde la ventana de PL-RIS al igual que, como con todas las demás variables, desde la [Ventana Estado variables](#) de PC-Sim.

Tabla 4: Direccionamiento a bit de las entradas de las EasyPort D16 con PL-RIS

EasyPort #1	FPC 404	PS3	S5 100U
INPUT 0	I0	I0.0	E0.0
INPUT 1	I1	I0.1	E0.1
INPUT 2	I2	I0.2	E0.2
INPUT 3	I3	I0.3	E0.3
INPUT 4	I4	I0.4	E0.4
INPUT 5	I5	I0.5	E0.5
INPUT 6	I6	I0.6	E0.6
INPUT 7	I7	I0.7	E0.7
INPUT 8	I1.0	I0.8	E1.0
INPUT 9	I1.1	I0.9	E1.1
INPUT 10	I1.2	I0.10	E1.2
INPUT 11	I1.3	I0.11	E1.3
INPUT 12	I1.4	I0.12	E1.4
INPUT 13	I1.5	I0.13	E1.5
INPUT 14	I1.6	I0.14	E1.6
INPUT 15	I1.7	I0.15	E1.7
EasyPort #2	FPC 404	PS3	S5 100U
INPUT 0	I2.0	I1.0	E8.0
INPUT 1	I2.1	I1.1	E8.1
INPUT 2	I2.2	I1.2	E8.2
INPUT 3	I2.3	I1.3	E8.3
INPUT 4	I2.4	I1.4	E8.4
INPUT 5	I2.5	I1.5	E8.5
INPUT 6	I2.6	I1.6	E8.6
INPUT 7	I2.7	I1.7	E8.7
INPUT 8	I3.0	I1.8	E9.0
INPUT 9	I3.1	I1.9	E9.1
INPUT 10	I3.2	I1.10	E9.2
INPUT 11	I3.3	I1.11	E9.3
INPUT 12	I3.4	I1.12	E9.4
INPUT 13	I3.5	I1.13	E9.5
INPUT 14	I3.6	I1.14	E9.6
INPUT 15	I3.7	I1.15	E9.7

EasyPort #3	FPC 404	PS3	S5 100U
INPUT 0	I4.0	I2.0	E10.0
INPUT 1	I4.1	I2.1	E10.1
INPUT 2	I4.2	I2.2	E10.2
INPUT 3	I4.3	I2.3	E10.3
INPUT 4	I4.4	I2.4	E10.4
INPUT 5	I4.5	I2.5	E10.5
INPUT 6	I4.6	I2.6	E10.6
INPUT 7	I4.7	I2.7	E10.7
INPUT 8	I5.0	I2.8	E11.0
INPUT 9	I5.1	I2.9	E11.1
INPUT 10	I5.2	I2.10	E11.2
INPUT 11	I5.3	I2.11	E11.3
INPUT 12	I5.4	I2.12	E11.4
INPUT 13	I5.5	I2.13	E11.5
INPUT 14	I5.6	I2.14	E11.6
INPUT 15	I5.7	I2.15	E11.7
EasyPort #4	FPC 404	PS3	S5 100U
INPUT 0	I6.0	I3.0	E12.0
INPUT 1	I6.1	I3.1	E12.1
INPUT 2	I6.2	I3.2	E12.2
INPUT 3	I6.3	I3.3	E12.3
INPUT 4	I6.4	I3.4	E12.4
INPUT 5	I6.5	I3.5	E12.5
INPUT 6	I6.6	I3.6	E12.6
INPUT 7	I6.7	I3.7	E12.7
INPUT 8	I7.0	I3.8	E13.0
INPUT 9	I7.1	I3.9	E13.1
INPUT 10	I7.2	I3.10	E13.2
INPUT 11	I7.3	I3.11	E13.3
INPUT 12	I7.4	I3.12	E13.4
INPUT 13	I7.5	I3.13	E13.5
INPUT 14	I7.6	I3.14	E13.6
INPUT 15	I7.7	I3.15	E13.7

EasyPort #5	FPC 404	PS3	S5 100U
INPUT 0	-	-	E14.0
INPUT 1	-	-	E14.1
INPUT 2	-	-	E14.2
INPUT 3	-	-	E14.3
INPUT 4	-	-	E14.4
INPUT 5	-	-	E14.5
INPUT 6	-	-	E14.6
INPUT 7	-	-	E14.7
INPUT 8	-	-	E15.0
INPUT 9	-	-	E15.1
INPUT 10	-	-	E15.2
INPUT 11	-	-	E15.3
INPUT 12	-	-	E15.4
INPUT 13	-	-	E15.5
INPUT 14	-	-	E15.6
INPUT 15	-	-	E15.7
EasyPort #6	FPC 404	PS3	S5 100U
INPUT 0	-	-	E16.0
INPUT 1	-	-	E16.1
INPUT 2	-	-	E16.2
INPUT 3	-	-	E16.3
INPUT 4	-	-	E16.4
INPUT 5	-	-	E16.5
INPUT 6	-	-	E16.6
INPUT 7	-	-	E16.7
INPUT 8	-	-	E17.0
INPUT 9	-	-	E17.1
INPUT 10	-	-	E17.2
INPUT 11	-	-	E17.3
INPUT 12	-	-	E17.4
INPUT 13	-	-	E17.5
INPUT 14	-	-	E17.6
INPUT 15	-	-	E17.7

EasyPort #7	FPC 404	PS3	S5 100U
INPUT 0	-	-	E18.0
INPUT 1	-	-	E18.1
INPUT 2	-	-	E18.2
INPUT 3	-	-	E18.3
INPUT 4	-	-	E18.4
INPUT 5	-	-	E18.5
INPUT 6	-	-	E18.6
INPUT 7	-	-	E18.7
INPUT 8	-	-	E19.0
INPUT 9	-	-	E19.1
INPUT 10	-	-	E19.2
INPUT 11	-	-	E19.3
INPUT 12	-	-	E19.4
INPUT 13	-	-	E19.5
INPUT 14	-	-	E19.6
INPUT 15	-	-	E19.7
EasyPort #8	FPC 404	PS3	S5 100U
INPUT 0	-	-	E20.0
INPUT 1	-	-	E20.1
INPUT 2	-	-	E20.2
INPUT 3	-	-	E20.3
INPUT 4	-	-	E20.4
INPUT 5	-	-	E20.5
INPUT 6	-	-	E20.6
INPUT 7	-	-	E20.7
INPUT 8	-	-	E21.0
INPUT 9	-	-	E21.1
INPUT 10	-	-	E21.2
INPUT 11	-	-	E21.3
INPUT 12	-	-	E21.4
INPUT 13	-	-	E21.5
INPUT 14	-	-	E21.6
INPUT 15	-	-	E21.7

Además, para los PLC PS3 y S5 100U existe la posibilidad de direccionamiento a byte y a word en los términos que aparecen en la tabla siguiente.

Tabla 5: Direccionamiento a byte y a word de las entradas de las EasyPort D16 con PL-RIS

EasyPort #1	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	IB0.0	EB0
INPUT 8 – INPUT 15	-	IB0.8	EB1
INPUT 0 – INPUT 15	-	IW0	EW0
EasyPort #2	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	IB1.0	EB8
INPUT 8 – INPUT 15	-	IB1.8	EB9
INPUT 0 – INPUT 15	-	IW1	EW8
EasyPort #3	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	IB2.0	EB10
INPUT 8 – INPUT 15	-	IB2.8	EB11
INPUT 0 – INPUT 15	-	IW2	EW10
EasyPort #4	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	IB3.0	EB12
INPUT 8 – INPUT 15	-	IB3.8	EB13
INPUT 0 – INPUT 15	-	IW3	EW12
EasyPort #5	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	-	EB14
INPUT 8 – INPUT 15	-	-	EB15
INPUT 0 – INPUT 15	-	-	EW14
EasyPort #6	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	-	EB16
INPUT 8 – INPUT 15	-	-	EB17
INPUT 0 – INPUT 15	-	-	EW16
EasyPort #7	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	-	EB18
INPUT 8 – INPUT 15	-	-	EB19
INPUT 0 – INPUT 15	-	-	EW18
EasyPort #8	FPC 404	PS3	S5 100U
INPUT 0 – INPUT 7	-	-	EB20
INPUT 8 – INPUT 15	-	-	EB21
INPUT 0 – INPUT 15	-	-	EW20

Direccionamiento de las salidas

El número de salidas del sistema depende, tal como ya se ha explicado, del número de las EasyPort D16 y del tipo de PLC simulado que se utilice, con un máximo de 64 para los PLC FPC 404 y PS3, y de 128 para el PLC S5 100U. Cada una de las salidas queda identificada de manera unívoca por dos términos:

- número de la EasyPort D16 a la que está conectada, con valores entre 1 y 8
- número del canal de salida de esta última, con valores entre OUTPUT 0 y OUTPUT 15

El estado de las salidas de la EasyPort D16 puede configurarse y adquirirse direccionando en el programa (ladder o AWL) las salidas del PLC de acuerdo con las indicaciones de la tabla siguiente. Preste atención al hecho de que, para los dos primeros PLC, las salidas están en secuencia mientras que para el último, esto no ocurre y hay una discontinuidad entre el byte 3 y el byte 8.

El estado de los dieciséis primeros bits de la salida podrá verse en la parte frontal del PLC simulado por medio de las oportunas señales de los pilotos luminosos. En cambio, el estado de los siguientes tan sólo podrá valorarse desde la ventana de PL-RIS al igual que, como con todas las demás variables, desde la [Ventana Estado variables](#) de PC-Sim.

Tabla 6: Direccionamiento a bit de las salidas de las EasyPort D16 con PL-RIS

EasyPort #1	FPC 404	PS3	S5 100U
OUTPUT 0	O0	Q0.0	A2.0
OUTPUT 1	O1	Q0.1	A2.1
OUTPUT 2	O2	Q0.2	A2.2
OUTPUT 3	O3	Q0.3	A2.3
OUTPUT 4	O4	Q0.4	A2.4
OUTPUT 5	O5	Q0.5	A2.5
OUTPUT 6	O6	Q0.6	A2.6
OUTPUT 7	O7	Q0.7	A2.7
OUTPUT 8	O1.0	Q0.8	A3.0
OUTPUT 9	O1.1	Q0.9	A3.1
OUTPUT 10	O1.2	Q0.10	A3.2
OUTPUT 11	O1.3	Q0.11	A3.3
OUTPUT 12	O1.4	Q0.12	A3.4
OUTPUT 13	O1.5	Q0.13	A3.5
OUTPUT 14	O1.6	Q0.14	A3.6
OUTPUT 15	O1.7	Q0.15	A3.7
EasyPort #2	FPC 404	PS3	S5 100U
OUTPUT 0	O2.0	Q1.0	A8.0
OUTPUT 1	O2.1	Q1.1	A8.1
OUTPUT 2	O2.2	Q1.2	A8.2
OUTPUT 3	O2.3	Q1.3	A8.3
OUTPUT 4	O2.4	Q1.4	A8.4
OUTPUT 5	O2.5	Q1.5	A8.5
OUTPUT 6	O2.6	Q1.6	A8.6
OUTPUT 7	O2.7	Q1.7	A8.7
OUTPUT 8	O3.0	Q1.8	A9.0
OUTPUT 9	O3.1	Q1.9	A9.1
OUTPUT 10	O3.2	Q1.10	A9.2
OUTPUT 11	O3.3	Q1.11	A9.3
OUTPUT 12	O3.4	Q1.12	A9.4
OUTPUT 13	O3.5	Q1.13	A9.5
OUTPUT 14	O3.6	Q1.14	A9.6
OUTPUT 15	O3.7	Q1.15	A9.7

EasyPort #3	FPC 404	PS3	S5 100U
OUTPUT 0	O4.0	Q2.0	A10.0
OUTPUT 1	O4.1	Q2.1	A10.1
OUTPUT 2	O4.2	Q2.2	A10.2
OUTPUT 3	O4.3	Q2.3	A10.3
OUTPUT 4	O4.4	Q2.4	A10.4
OUTPUT 5	O4.5	Q2.5	A10.5
OUTPUT 6	O4.6	Q2.6	A10.6
OUTPUT 7	O4.7	Q2.7	A10.7
OUTPUT 8	O5.0	Q2.8	A11.0
OUTPUT 9	O5.1	Q2.9	A11.1
OUTPUT 10	O5.2	Q2.10	A11.2
OUTPUT 11	O5.3	Q2.11	A11.3
OUTPUT 12	O5.4	Q2.12	A11.4
OUTPUT 13	O5.5	Q2.13	A11.5
OUTPUT 14	O5.6	Q2.14	A11.6
OUTPUT 15	O5.7	Q2.15	A11.7
EasyPort #4	FPC 404	PS3	S5 100U
OUTPUT 0	O6.0	Q3.0	A12.0
OUTPUT 1	O6.1	Q3.1	A12.1
OUTPUT 2	O6.2	Q3.2	A12.2
OUTPUT 3	O6.3	Q3.3	A12.3
OUTPUT 4	O6.4	Q3.4	A12.4
OUTPUT 5	O6.5	Q3.5	A12.5
OUTPUT 6	O6.6	Q3.6	A12.6
OUTPUT 7	O6.7	Q3.7	A12.7
OUTPUT 8	O7.0	Q3.8	A13.0
OUTPUT 9	O7.1	Q3.9	A13.1
OUTPUT 10	O7.2	Q3.10	A13.2
OUTPUT 11	O7.3	Q3.11	A13.3
OUTPUT 12	O7.4	Q3.12	A13.4
OUTPUT 13	O7.5	Q3.13	A13.5
OUTPUT 14	O7.6	Q3.14	A13.6
OUTPUT 15	O7.7	Q3.15	A13.7

EasyPort #5	FPC 404	PS3	S5 100U
OUTPUT 0	-	-	A14.0
OUTPUT 1	-	-	A14.1
OUTPUT 2	-	-	A14.2
OUTPUT 3	-	-	A14.3
OUTPUT 4	-	-	A14.4
OUTPUT 5	-	-	A14.5
OUTPUT 6	-	-	A14.6
OUTPUT 7	-	-	A14.7
OUTPUT 8	-	-	A15.0
OUTPUT 9	-	-	A15.1
OUTPUT 10	-	-	A15.2
OUTPUT 11	-	-	A15.3
OUTPUT 12	-	-	A15.4
OUTPUT 13	-	-	A15.5
OUTPUT 14	-	-	A15.6
OUTPUT 15	-	-	A15.7
EasyPort #6	FPC 404	PS3	S5 100U
OUTPUT 0	-	-	A16.0
OUTPUT 1	-	-	A16.1
OUTPUT 2	-	-	A16.2
OUTPUT 3	-	-	A16.3
OUTPUT 4	-	-	A16.4
OUTPUT 5	-	-	A16.5
OUTPUT 6	-	-	A16.6
OUTPUT 7	-	-	A16.7
OUTPUT 8	-	-	A17.0
OUTPUT 9	-	-	A17.1
OUTPUT 10	-	-	A17.2
OUTPUT 11	-	-	A17.3
OUTPUT 12	-	-	A17.4
OUTPUT 13	-	-	A17.5
OUTPUT 14	-	-	A17.6
OUTPUT 15	-	-	A17.7

EasyPort #7	FPC 404	PS3	S5 100U
OUTPUT 0	-	-	A18.0
OUTPUT 1	-	-	A18.1
OUTPUT 2	-	-	A18.2
OUTPUT 3	-	-	A18.3
OUTPUT 4	-	-	A18.4
OUTPUT 5	-	-	A18.5
OUTPUT 6	-	-	A18.6
OUTPUT 7	-	-	A18.7
OUTPUT 8	-	-	A19.0
OUTPUT 9	-	-	A19.1
OUTPUT 10	-	-	A19.2
OUTPUT 11	-	-	A19.3
OUTPUT 12	-	-	A19.4
OUTPUT 13	-	-	A19.5
OUTPUT 14	-	-	A19.6
OUTPUT 15	-	-	A19.7
EasyPort #8	FPC 404	PS3	S5 100U
OUTPUT 0	-	-	A20.0
OUTPUT 1	-	-	A20.1
OUTPUT 2	-	-	A20.2
OUTPUT 3	-	-	A20.3
OUTPUT 4	-	-	A20.4
OUTPUT 5	-	-	A20.5
OUTPUT 6	-	-	A20.6
OUTPUT 7	-	-	A20.7
OUTPUT 8	-	-	A21.0
OUTPUT 9	-	-	A21.1
OUTPUT 10	-	-	A21.2
OUTPUT 11	-	-	A21.3
OUTPUT 12	-	-	A21.4
OUTPUT 13	-	-	A21.5
OUTPUT 14	-	-	A21.6
OUTPUT 15	-	-	A21.7

Además, para los PLC PS3 y S5 100U existe la posibilidad de direccionamiento a byte y a word en los términos que muestra la tabla siguiente.

Tabla 7: Direccionamiento a byte e a word de las salidas de las EasyPort D16 con PL-RIS

EasyPort #1	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	QB0 . 0	AB2
OUTPUT 8 – OUTPUT 15	-	QB0 . 8	AB3
OUTPUT 0 – OUTPUT 15	-	QW0	AW2
EasyPort #2	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	QB1 . 0	AB8
OUTPUT 8 – OUTPUT 15	-	QB1 . 8	AB9
OUTPUT 0 – OUTPUT 15	-	QW1	AW8
EasyPort #3	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	QB2 . 0	AB10
OUTPUT 8 – OUTPUT 15	-	QB2 . 8	AB11
OUTPUT 0 – OUTPUT 15	-	QW2	AW10
EasyPort #4	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	QB3 . 0	AB12
OUTPUT 8 – OUTPUT 15	-	QB3 . 8	AB13
OUTPUT 0 – OUTPUT 15	-	QW3	AW12
EasyPort #5	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	-	AB14
OUTPUT 8 – OUTPUT 15	-	-	AB15
OUTPUT 0 – OUTPUT 15	-	-	AW14
EasyPort #6	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	-	AB16
OUTPUT 8 – OUTPUT 15	-	-	AB17
OUTPUT 0 – OUTPUT 15	-	-	AW16
EasyPort #7	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	-	AB18
OUTPUT 8 – OUTPUT 15	-	-	AB19
OUTPUT 0 – OUTPUT 15	-	-	AW18
EasyPort #8	FPC 404	PS3	S5 100U
OUTPUT 0 – OUTPUT 7	-	-	AB20
OUTPUT 8 – OUTPUT 15	-	-	AB21
OUTPUT 0 – OUTPUT 15	-	-	AW20

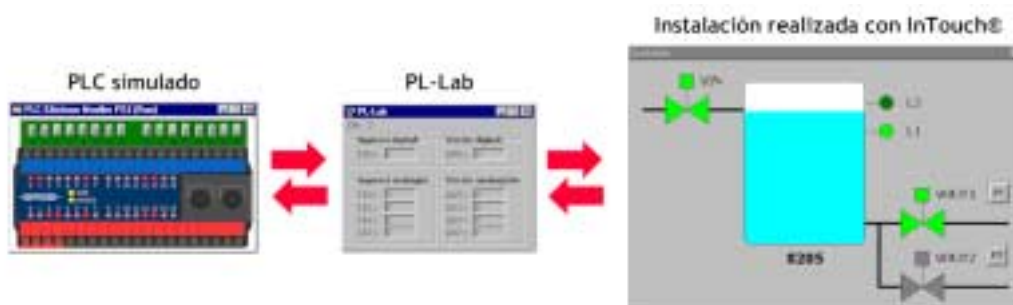
CAPÍTULO 7

PL-LAB: INTERFACE HACIA INSTALACIONES DE FABRICACIÓN PROPIA

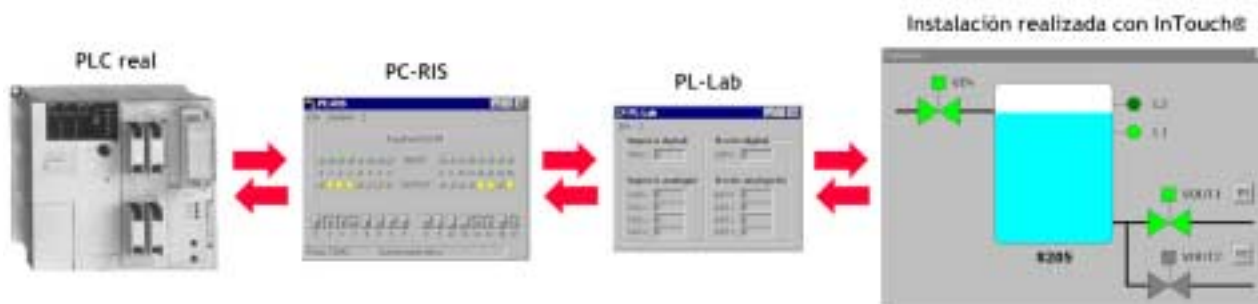
Introducción

Un último componente del mundo AW-SYS es PL-Lab, interface entre PC-Sim o PC-RIS e InTouch.

Tal como se muestra en la siguiente figura, PL-Lab permite el intercambio de datos, y, por tanto, la conexión entre los PLC simulados de PC-Sim e InTouch, el conocido paquete de supervisión de Wonderware, mediante el cual podrá construir por su cuenta nuevas ventanas de simulación de instalaciones.



Esas mismas instalaciones también podrán controlarse desde PLC reales, o desde otros sistemas de control, por medio del módulo PC-RIS del que ya se ha hablado antes.



Con PL-Lab se proporcionan la fuente de una instalación prototipo y algunos ejemplos de instalaciones realizadas con InTouch.

En este capítulo, además de describir el funcionamiento de PL-Lab, se presentarán algunos ejemplos de instalaciones realizadas con él. Con respecto a los procedimientos de construcción de los mismos, se describirán con detalle, pero no se facilitarán explicaciones específicas acerca de la utilización de InTouch. Por tanto, salvo que sea un usuario experto de InTouch, será conveniente que compagine la lectura de este capítulo con otras informaciones incluidas en los manuales de InTouch.

Puesta en marcha del sistema

Puesta en marcha de PC-Sim o PC-RIS

PL-Lab puede funcionar tan sólo en conexión con uno de los simuladores de PLC o con PC-RIS. Por ello, es necesario que se esté ejecutando uno de éstos antes de poner en marcha que PL-Lab.

Para iniciar PC-Sim o PC-RIS, haga clic sobre la tecla Inicio de Windows, a continuación abra la carpeta Programas\Aw-sys y, en el interior de ésta, seleccione la opción de uno de los simuladores de PLC o de la interface hacia los PLC reales: PC-RIS.

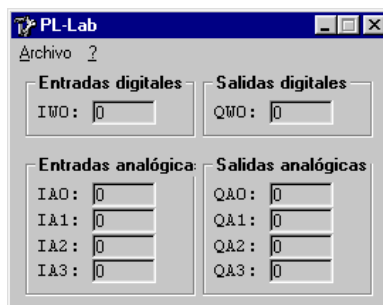
Puesta en marcha de PL-Lab

Para poner en marcha PL-Lab seleccione el *Comando PL-Lab* en el *Menú Instalaciones* de PC-Sim o de PC-RIS.

PC-Sim y PC-RIS ven a PL-Lab como si fuera una de las instalaciones simuladas presentes en el paquete, y por ello, inmediatamente después de su puesta en marcha, el *Menú Instalaciones* se desactiva para impedir la apertura simultánea de otra instalación. Dicho menú no volverá a activarse hasta el cierre de PL-Lab.

Descripción de PL-Lab

La ventana de PL-Lab contiene una serie de casillas que muestran los valores de las entradas y de las salidas del PLC. Las casillas de entrada permiten que la instalación pueda asignar el valor de las entradas digitales y analógicas del PLC. En cambio, desde las casillas de salida, la instalación podrá leer los valores de las salidas digitales y analógicas del PLC.



En concreto, la casilla IW0 contiene el valor, expresado en base decimal, de las 16 entradas digitales del PLC, con límites situados entre 0 y 65535. Lo mismo puede decirse para la casilla QW0, que contiene el valor de las 16 salidas del PLC.

Para el PLC PS3 simulado, el estado de cada uno de los canales de I/O ocupa su bit correspondiente en la word específica. Así, pues, el canal de entrada **I0.0** ocupa el bit 0 de la word **IW0**, el canal **I0.1** el bit 1, y así sucesivamente para terminar con el canal **I0.15** que, en la misma word, ocupa el bit de mayor peso: el 15. Lo mismo sucede con los canales de salida en la composición de **QW0**.

Word	IW0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Entrada	I0.15	I0.14	I0.13	I0.12	I0.11	I0.10	I0.9	I0.8	I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0

Word	QW0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Salida	Q0.15	Q0.14	Q0.13	Q0.12	Q0.11	Q0.10	Q0.9	Q0.8	Q0.7	Q0.6	Q0.5	Q0.4	Q0.3	Q0.2	Q0.1	Q0.0

Para el PLC FPC 404 los ocho canales de entrada del módulo 0 ocupan el byte menos significativo de la **IW0**, conservando su orden en el interior. En cambio, el byte más significativo está compuesto de manera análoga a las entradas del módulo 1. Las salidas de esos mismos módulos admiten una representación similar en el interior de la word **QW0**.

Word	IW0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Entrada	I1.7	I1.6	I1.5	I1.4	I1.3	I1.2	I1.1	I1.0	I0.7	I0.6	I0.5	I0.4	I0.3	I0.2	I0.1	I0.0

Word	QW0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Salida	O1.7	O1.6	O1.5	O1.4	O1.3	O1.2	O1.1	O1.0	O0.7	O0.6	O0.5	O0.4	O0.3	O0.2	O0.1	O0.0

Para el PLC S5 100U la forma de las word es similar a la que acabamos de ver, salvo en la diversa numeración de los módulos de salida: 2 y 3.

Word	IW0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Entrada	E1.7	E1.6	E1.5	E1.4	E1.3	E1.2	E1.1	E1.0	E0.7	E0.6	E0.5	E0.4	E0.3	E0.2	E0.1	E0.0

Word	QW0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Salida	A3.7	A3.6	A3.5	A3.4	A3.3	A3.2	A3.1	A3.0	A2.7	A2.6	A2.5	A2.4	A2.3	A2.2	A2.1	A2.0

Las casillas entre **IA0** y **IA3** visualizan el valor de las entradas analógicas del PLC. Tal valor, con independencia del modelo del PLC simulado, está normalizado entre 0 y 10000. Éste se irá graduando después, dependiendo del PLC conectado, para estar disponible para él de una manera congruente con la representación esperada. Por ejemplo, para el PLC PS3 el valor 0-10000 se convertirá en un valor 0-255, coherente con la representación sobre 8 bit de las entradas analógicas de este PLC. Para más información

sobre la codificación de las entradas analógicas, lea todo lo dicho en los capítulos dedicados a las notas de programación de los PLC PS3 y S5 100U. En la simulación del PLC FPC 404 se han implementado canales analógicos.

Intercambio de datos con InTouch

PL-Lab es sustancialmente un *server* DDE del cual la instalación personalizada que se ha realizado con InTouch podrá obtener o introducir datos, comportándose como *client* del DDE.

El protocolo DDE identifica un elemento de datos a través de una convención semántica formada de tres partes: *Application Name*, *Topic Name* e *Item Name*. Para obtener datos de una aplicación *server* DDE, el programa *cliente* DDE abre un canal hacia ella especificando estos tres elementos.

Por tanto, para adquirir datos desde otra aplicación, InTouch deberá conocer el nombre de la aplicación que proporciona los datos, el nombre del *topic* en el que está contenido el dato y el nombre del *item* específico, situado en el interior del *topic* y que identifica unívocamente el dato en cuestión. Además, para la gestión interna, será necesario especificar el tipo de dato: *Discrete*, *Integer*, *Real* o *Message*. El conjunto de estas informaciones especifica el DDE para el tag cuando éste se ha definido en la base de datos de InTouch. Una vez se ha establecido todo esto y se ha iniciado el entorno runtime, se ejecutarán automáticamente todas las acciones para adquirir y mantener actualizado el dato de este tag.

En el detalle, para intercambiar datos con PL-Lab, deberán especificarse las siguientes líneas:

- *Application Name* PLLAB
- *Topic Name* PLCIO

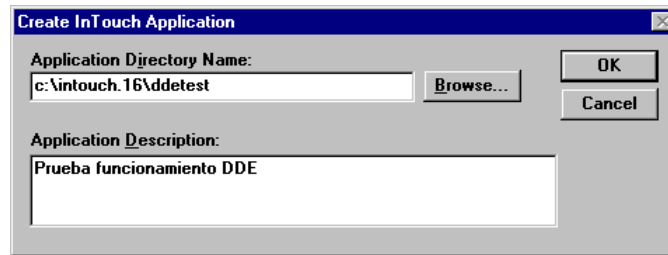
En cambio, el *Item Name* dependerá del dato concreto al que se quiere hacer referencia, tal como queda resumido a continuación:

- Entradas digitales del PLC IW0
- Salidas digitales del PLC QW0
- Entradas analógicas del PLC IA0, IA1, IA2, IA3
- Salidas analógicas del PLC QA0, QA1, QA2, QA3

Construcción de la primera instalación


Tras haber analizado lo específico del intercambio de datos entre InTouch y PL-Lab, hay que lanzarse a la construcción de una primera instalación.

Verifique que PC-Sim o PC-RIS y PL-Lab están en ejecución; a continuación inicie InTouch y cree una nueva aplicación accionando el pulsador *Create Directory...* . Luego, en la ventana de diálogo *Create InTouch Application*, cumplimente los campos tal como se muestra en la figura siguiente.



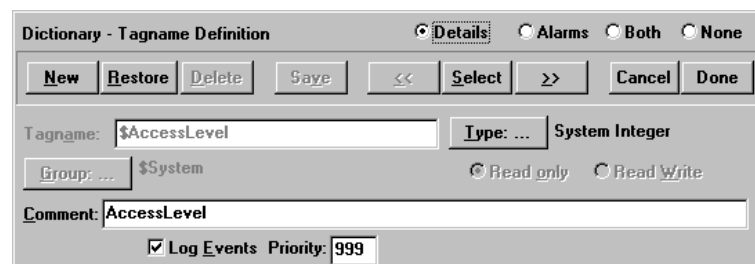
Donde la ruta especificada en la casilla *Application Directory Name* depende de si ha elegido insertar el nuevo directorio en la carpeta de InTouch, aunque también puede insertar cualquier nombre válido de su elección.

Pulse *Ok* para cerrar la ventana.

Accione ahora el pulsador de puesta en marcha del *WindowMaker*  para entrar en el entorno de desarrollo.

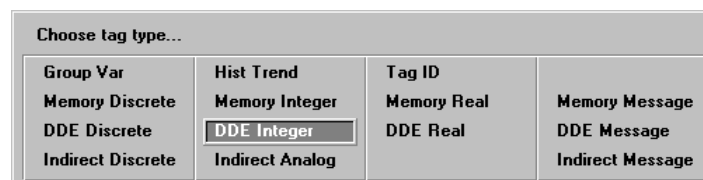
Veamos cual es el procedimiento a seguir para configurar un tag de la base de datos para que su valor refleje el contenido de la word QW0, es decir, la word de las salidas digitales del PLC.

En el menú *Special* de *WindowMaker* seleccione el comando *Tagname Dictionary*. Se abrirá la ventana de definición del tag.



Para declarar un nuevo tag pulse *New*. En la casilla *Tagname* escriba "QW0" y en la *Comment* escriba "Salidas digitales PLC (word)".

Para definir el tipo de dato y su posición, pulse *Type* y, en la ventana que aparece, seleccione *DDE Integer* haciendo doble clic sobre éste.



Tras esta operación, la ventana *Tagname Definition* se amplia para hacer sitio a las específicas de acceso al DDE.

Haga clic sobre *DDE Access Name* para que aparezca la ventana de diálogo *DDE Access Name Definition*.

Haga clic sobre el pulsador *Add*. Cumplimente las casillas como en la figura que sigue para definir así dos de los tres parámetros del DDE: *Application Name* (PLLAB) y *Topic Name* (PLCIO) y asociar un nombre identificativo al conjunto (PL-Lab).

A continuación pulse *OK*. El nuevo nombre de acceso al DDE se añadirá a la lista. Haga clic sobre el pulsador *Done* para cerrar la ventana.

Corrijamos los valores predeterminados para la casillas *Max EU* y *Max Raw* tecleando en ellas 65535: el valor máximo de QW0 que se obtiene cuando los 16 bits de salida del PLC están a 1.

Para terminar, seleccione el botón opción *Use Tagname as Item Name*, definiendo así el último elemento del DDE: como *Item Name* se utilizará el mismo nombre que ya se ha especificado para el tag (QW0). En ese instante, la ventana *Tagname Definition* aparecerá así.

Dictionary - Tagname Definition

Details Alarms Both None

New Restore Delete Save << Select >> Cancel Done

Tagname: QW0 Type: ... DDE Integer

Group: ... \$System Read only Read Write

Comment: Salidas digitales del PLC (word)

Log Data Log Events Retentive Value Retentive Parameters

Initial Value: 0 Min EU: 0 Max EU: 65535

Deadband: 0 Min Raw: 0 Max Raw: 65535

Eng Units: Conversion Linear Square Root

DDE Access Name: ... PL-Lab

Item: QW0

Use Tagname as Item Name Log Deadband: 0

Hemos terminado la definición del tag QW0. Si usa PL-RIS, cualquier referencia a este tag conducirá a la obtención del valor de la word de salida del PLC simulado o del real. Cierre la ventana haciendo clic sobre el pulsador *Done*. Ya estamos listos para experimentar con lo realizado.

En el menú *Archivo* del *WindowMaker* seleccione el comando *New Window* y, en la casilla *Name* de la ventana de diálogo *Window Property* escriba "Mi instalación", dejando igual el resto de las configuraciones.

Haga clic sobre *OK* para confirmar y abrir la nueva ventana.

Window Properties

Name: Mi instalación Window Color: OK

Comment: Cancel

Window Type Replace Overlay Popup

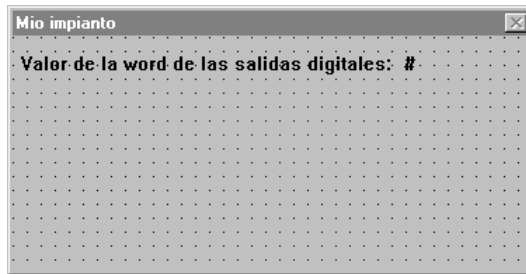
Frame Style Single Double None

Title Bar Size Controls

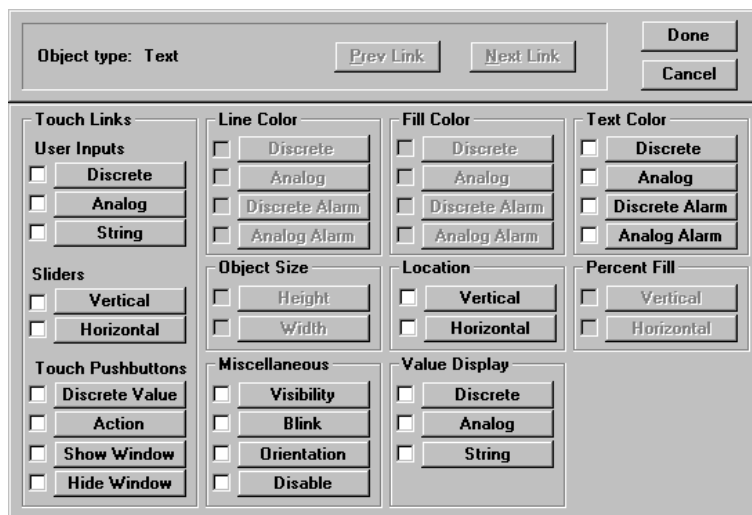
Dimensions X Location: 4 Y Location: 4 Window Width: 632 Window Height: 278

Scripts ...

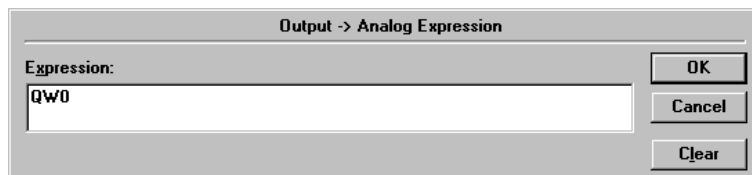
Si la *Toolbox* no está abierta, ábrala seleccionando el comando *Toolbox Show/Hide* en el menú *Special*.
Seleccione la herramienta *Text* y a continuación *Left Justify* . Luego, haga clic en la ventana *Mi instalación*, y escriba "Valor de la word de las salidas digitales:". Seleccione de nuevo la herramienta texto y haga clic otra vez sobre la ventana. Teclee ahora "#" para especificar a InTouch que el campo se utilizará para visualizar un valor numérico. Haga clic fuera del texto que acaba de insertar para terminar la fase de input. Si es necesario, desplace los dos objetos para obtener una visualización similar a la que aparece en la figura siguiente.



Definamos ahora las características de animación del último objeto insertado. Haga doble clic sobre él, o bien selecciónelo y pulse Intro, para abrir la ventana de diálogo de definición de la animación.



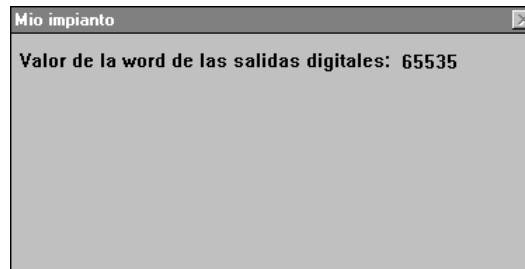
Haga clic sobre el pulsador *Analog* en el panel *Value Display*. En la casilla *Expression* de la ventana de diálogo que aparece, escriba "QW0": así, en runtime al símbolo # le sustituirá la visualización del valor de QW0.



Haga clic sobre *OK*. Ahora, el pulsador opción situado junto al pulsador *Analog* que ha accionado antes, será el seleccionado para indicar que se ha efectuado una asignación para tal característica.

Pulse *Done* para terminar.

Ya estamos listos para comprobar el funcionamiento del DDE. Haga clic sobre el menú *Runtime!*, situado en el extremo derecho de la barra de menús del *WindowMaker*, para poner en marcha el entorno runtime de InTouch. La figura siguiente muestra lo que aparecerá.

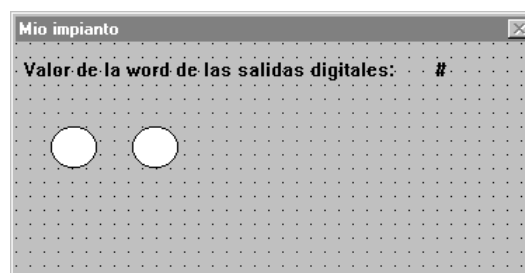


El número visualizado como valor de la word de las salidas digitales depende de si se han activado todas las salidas del PLC. Para hacer variar la visualización intente forzar las salidas con valores distintos. Consulte todo lo mencionado en el apartado *Forzado de variables* del *Capítulo 3* o, si usa PL-RIS, el manual del software de programación de su PLC real.

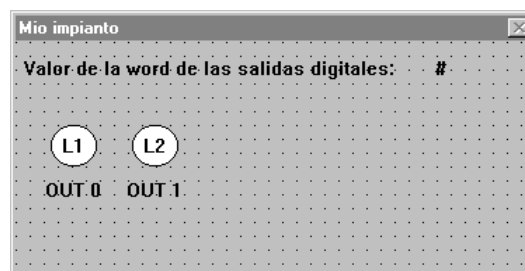
Vamos a enriquecer ahora la instalación, que por el momento no hace honor en absoluto a este nombre, insertando dos luces que estarán controladas por las dos primeras salidas del PLC.

Regresemos a la modalidad de proyecto haciendo clic en el menú *Development!* situado en el extremo derecho de la barra de menús de la ventana *WindowViewer*.

En la *Toolbox* seleccione la herramienta *Ellipse* y luego trace un círculo en la ventana *Mi instalación*. Dibuje otro igual o bien seleccione el primero y accione el comando *Duplicar* del menú *Edit*. Determine la posición y el tamaño de los dos círculos para conseguir algo similar a la figura siguiente.



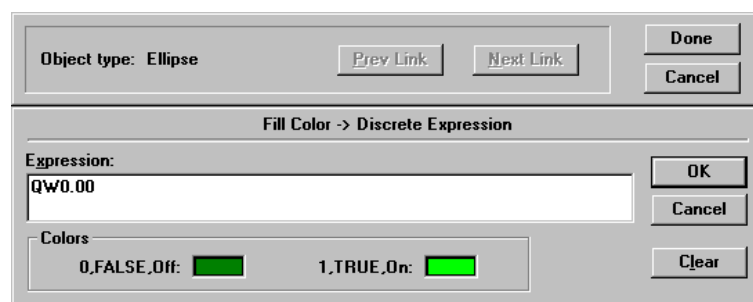
Tal como se ha indicado anteriormente, inserte cuatro líneas para mejorar la lectura, con el fin de obtener lo que se muestra en la figura siguiente.



L1 y L2 serán las siglas de las dos luces, mientras que OUT0 y OUT1 nos recordarán a qué salidas del PLC están conectadas cada una de ellas. Para alinear los textos con los círculos serán de utilidad los comandos del menú *Arrange - Align*.

Definamos ahora las características de animación para las dos luces. Haga doble clic sobre la primera para abrir la ventana de diálogo de la animación. En esa ventana, haga clic sobre el pulsador *Discrete* del panel *Fill color*.

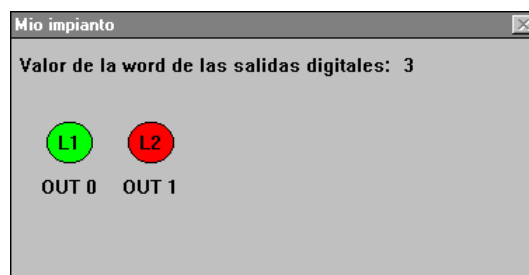
En la casilla *Expression* de la ventana de diálogo *Fill Color* escriba "QW0.00" para especificar que el bit 0 de la word QW0 determinará el color de la luz. A continuación, programe dos colores distintos para las casillas FALSE y TRUE del panel *Colors*, por ejemplo verde oscuro y verde claro respectivamente. La ventana aparecerá así:



Haga clic sobre el pulsador *Done* para terminar.

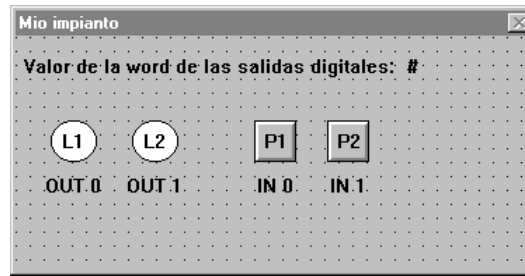
Repita las operaciones anteriores para la segunda luz, especificando como expresión "QW0.01" es decir, el bit 1 de la word de las salidas del PLC y, si lo desea, seleccionando dos colores diferentes.

Inicie el *WindowViewer* seleccionando el comando *RunTime!*. En la figura siguiente está representado lo que debería aparecer. La visualización se ha obtenido con un valor de QW0 igual a 3, es decir, con bit 0 y bit 1 ambos ciertos.



Variando el estado de las salidas del PLC las luces cambiarán de color, simulando así su encendido o apagado.

Añadamos ahora unos pulsadores para controlar dos entradas del PLC. Seleccione la herramienta *Button* en la *Toolbox* del *WindowViewer*. En la ventana *Mi instalación* dibuje dos pulsadores; cambie su forma, tamaño y posición para obtener el resultado que muestra la siguiente figura. Después, añada o modifique los textos.



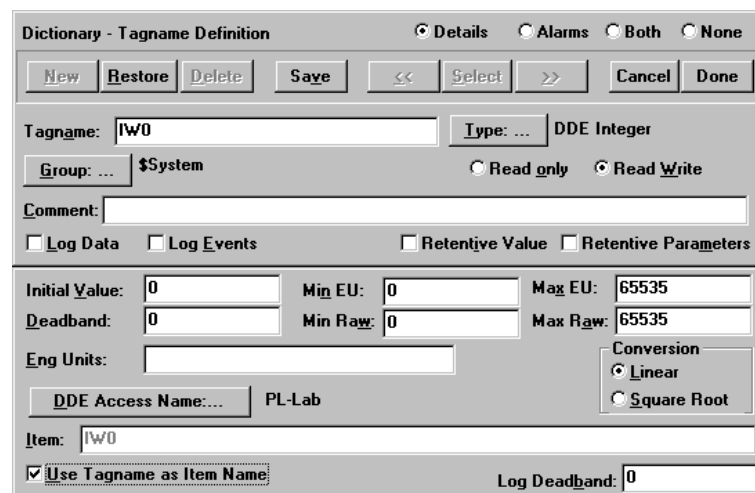
Los textos del interior de los pulsadores pueden modificarse seleccionándolos, de uno en uno, y eligiendo el comando *Reemplazar Strings* en el menú *Special*.

Ocupémonos de la conexión de los pulsadores con el PLC, empezando por P1. Haga doble clic sobre él y aparecerá la ventana de diálogo para seleccionar la animación. A continuación, haga clic sobre el pulsador *Discrete value* en el grupo *Touch Pushbuttons* del panel *Touch Links*.



En la casilla *Tagname* escriba "IW0.00", para indicar que el bit que cambiará al presionar el pulsador es el bit 0 de la word de las entradas digitales del PLC, y luego pulse *OK*. Dado que aún no está definido el tag IW0, Intouch le pregunta si desea hacerlo en ese momento. Responda afirmativamente.

En la casilla *Tagname* de la nueva ventana de diálogo que aparece se muestra ya "IW0". Haga clic sobre el pulsador *Type*, seleccione el tipo *DDE Integer* en el listado y pulse *OK*. El contenido de las casillas *Max EU* y *Max Raw* deberían estar ya en el valor correcto: el último programado para la word de salida, es decir, 65535. Si no fuera así, escriba el valor justo en dichas casillas. Para acabar, seleccione el pulsador opción *Use Tagname as Item Name* y pulse *Done*.



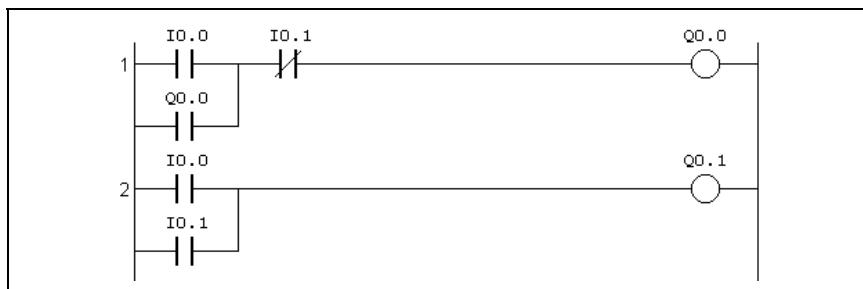
Repita la operación para el pulsador P2, el valor a programar para el *TagName* es IW0.01, es decir, el bit 1 de la word de las entradas. No se le pedirá que defina el tag IW0 ya que acaba de hacerlo para el otro pulsador.

¡Bien! La instalación ya está lista. Podemos ponerla en marcha situando InTouch en runtime.

Para verificar el funcionamiento de las entradas intentemos accionar el pulsador P1. La casilla IW0 de PL-Lab mostrará el valor 1, informándonos de que está activado el bit 0. Haciendo clic sobre P2 la misma casilla mostrará el valor 2 señalándonos que, en esta ocasión, está activado el bit 1. En función del valor de estos bits, los primeros pilotos indicadores del estado de las entradas del PLC se encenderán o se apagarán.

Ha llegado el momento de escribir un programa en el PLC para verificar el funcionamiento de toda la instalación y de las conexiones. Pongámonos como objetivo hacer que la luz L1 se encienda con el pulsador P1 y se apague con el P2. Por el contrario, la luz L2 se encenderá siempre, cada vez que se accione uno de los dos pulsadores

Supongamos que estamos utilizando el PLC simulado PS3. El programa ladder a diseñar es el siguiente:



Inicie el PLC e intente accionar los dos pulsadores para verificar el funcionamiento de la instalación y del programa.

Si el procedimiento descrito hasta aquí se ha ejecutado correctamente, el estado de las luces simuladas variará en función de la acción sobre los pulsadores, tal como se ha especificado anteriormente. Por el contrario, si la instalación no funciona correctamente, debe verificar el estado de las entradas y de las salidas del PLC, observando el estado de los LED correspondientes situados en la parte frontal de la instalación y luego verifique que ha programado correctamente los tag de InTouch.

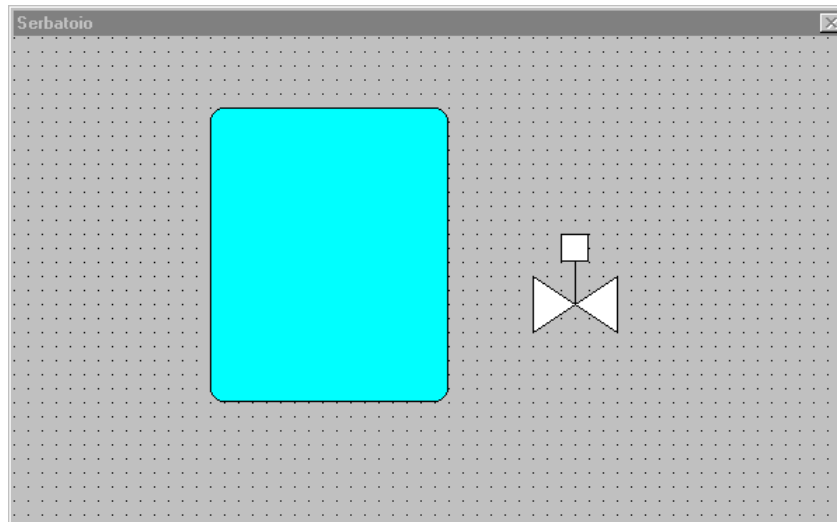
Construcción de una instalación más compleja

Realicemos ahora otra instalación, de mayor complejidad. Se trata de la reproducción de un depósito alimentado por medio de una tubería de entrada del que se puede extraer líquido a través de dos tuberías de salida. La primera tubería está interceptada por la válvula VIN controlada por el PLC. Las segundas están interceptadas por las válvulas VOUT1 y VOUT2 que pueden controlarse manualmente mediante dos pulsadores: P1 y P2.

Tal como ha hecho en el ejemplo anterior, cree una nueva aplicación seleccionando un nuevo directorio. A continuación, abra el *WindowMaker* y cree la ventana de la instalación, seleccionando el comando *New Window* en el menú *File* y escribiendo "Depósito" en la casilla *Name*.

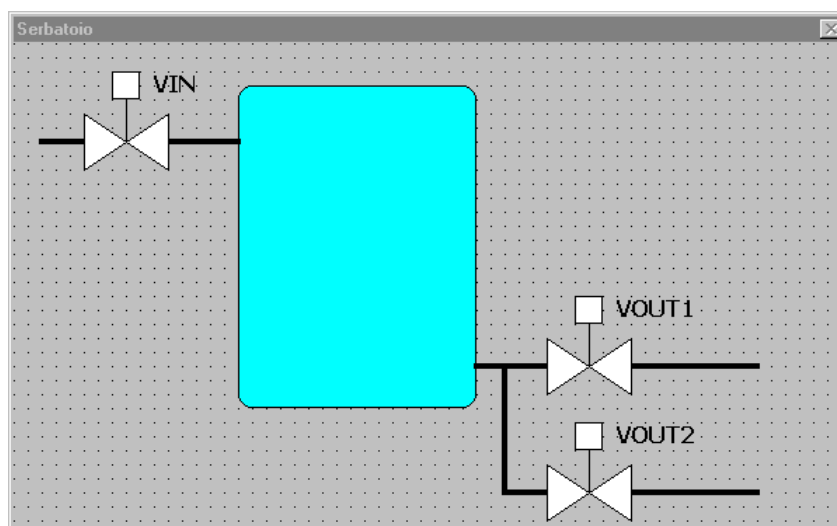
En la ventana que acaba de crear dibuje el depósito, seleccionando la herramienta *Rounded Rectangle* en la *Toolbox* del *WindowMaker*, y trace un rectángulo más alto que ancho.

A continuación, seleccione la herramienta *Polygon* y trace la parte inferior de la figura de una válvula. Utilice las herramientas *Line* y *Rectangle* para completar el esquema, tal como se muestra en la figura siguiente.



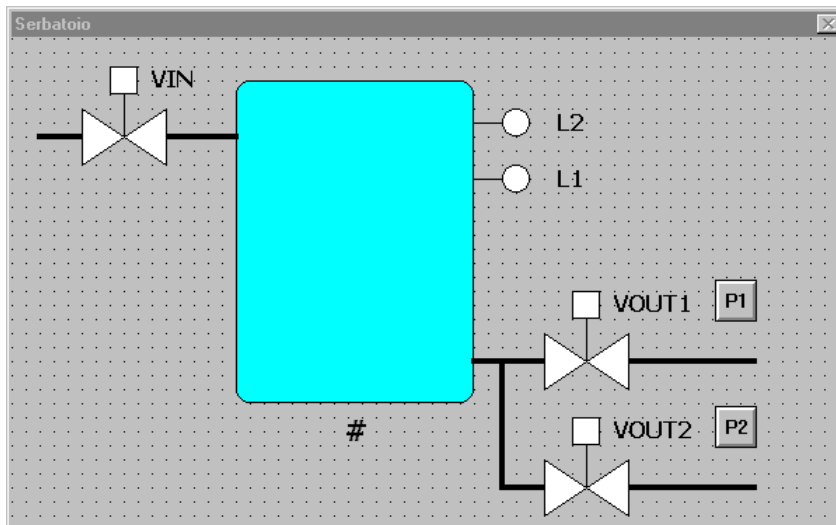
Ahora, seleccione toda la válvula y elija el comando *Make symbol* en el menú *Arrange*: la válvula, constituida antes por un conjunto de partes, se ha convertido en un único elemento .

Duplique dos veces la válvula, desplace los objetos y complete el dibujo trazando las líneas de las tuberías. A continuación, incluya los textos para conseguir lo que muestra la figura siguiente.



Con la herramienta *Button* dibuje los pulsadores P1 y P2, y con la herramienta *Ellipse*, dibuje dos círculos que representarán los indicadores de nivel. Complete el trabajo escribiendo o modificando los textos e insertando debajo del depósito el texto "#", lo cual le permitirá visualizar, en formato numérico, el nivel del líquido que hay en su interior.

Entonces, la ventana *Depósito* se presentará tal como muestra la siguiente figura.



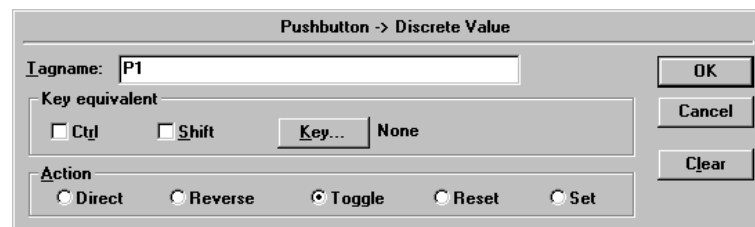
Definamos ahora las características de animación de cada uno de los objetos. Recuerde que el acceso a la ventana de diálogo, para definir la animación, se obtiene haciendo doble clic sobre el objeto. No se volverá a describir este procedimiento: se hará referencia, directamente, a las características específicas de animación del objeto que se analice en cada caso. Empecemos por las del depósito.

Haga clic sobre el pulsador *Vertical* en el panel *Percent Fill* de la ventana de definición de la animación y cumplimente sus campos tal como sigue:

Cuando pulse *OK*, se le pedirá que defina el tag *LEVEL*. En la ventana de diálogo *Tagname Definition*, programe los valores que muestra la siguiente figura, prestando atención a escoger el tipo *Memory Integer*.

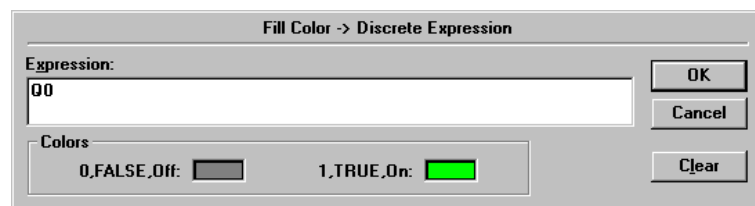
Las configuraciones especifican, entre otras cosas, que la cantidad máxima de líquido en el depósito será de 10000 unidades y que la instalación se pondrá en marcha con el depósito lleno hasta la mitad (5000 unidades).

Pasemos ahora a la definición de la animación de los pulsadores. Para P1, haga clic sobre el pulsador *Discrete Value* en el grupo *Touch Pushbuttons* del panel *Touch Links* y programe el *Tagname* en "P1". A continuación, pulse *OK*.



Se le pedirá que defina el tag P1. La única configuración que se puede efectuar es la del tipo: *Memory Discrete*. Siga un procedimiento análogo para el pulsador P2, definiendo como tag a "P2".

Pasemos ahora a las válvulas. La válvula VIN tendrá la propiedad *Fill Color - Discrete* programada como en la figura siguiente.



El color de llenado dependerá del tag Q0, es decir, del canal de salida del PLC. Los colores seleccionados son el gris para el estado OFF y el verde claro para el estado ON.

Pulse *OK* para cerrar la ventana. Se le pedirá que defina el tag Q0. La única configuración que se puede efectuar es la del tipo: *Memory Discrete*.

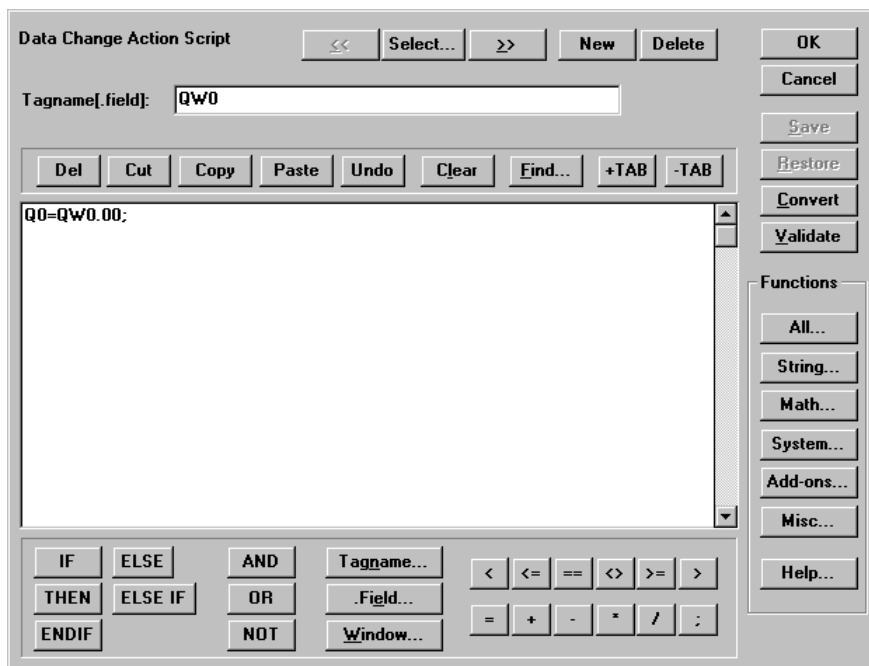
Para las válvulas VOUT1 y VOUT2, efectúe configuraciones análogas y asócielas, respectivamente, a los tag P1 y P2.

Las configuraciones para los indicadores de nivel son sencillas: se trata de definir para estos la propiedad *Fill Color - Discrete* en relación con los tag I0 para L1 y I1 para L2. Esta configuración permitirá, a través de la variación del color del indicador de nivel, la visualización del estado de la entrada del PLC específico. Por el contrario, la configuración de la entrada como valor verdadero o falso, se efectuará a continuación.

Tan sólo falta animar el indicador numérico de nivel. Definimos para él la propiedad *Value Display - Analog* asociándola al tag LEVEL.

Definamos ahora las funciones que nos permitirán pasar de las word de entrada/salida a cada uno de los bits, y en concreto los que se utilizan en la presente instalación.

Seleccione el comando *Special – Scripts – Data Change Scripts...* y cumplimente los campos como en la figura siguiente.



A cada variación del valor de QW0, Q0 se actualizará con el contenido del bit 0 de la word.

Presione ahora el pulsador *New* y en la casilla *Tagname* escriba 'I0'. A continuación inserte el siguiente script:

```
IW0.00=I0;
```

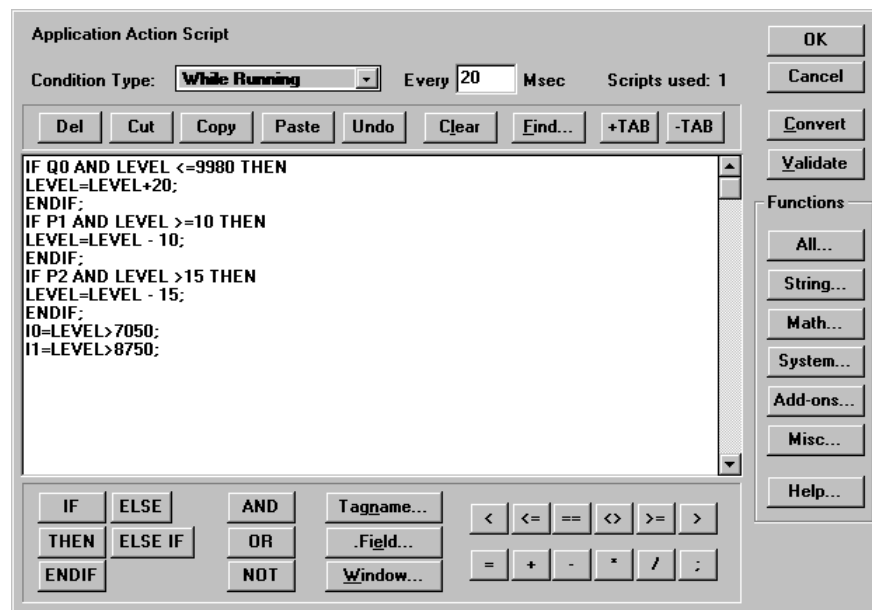
A cada variación de I0 el bit 0 de la word IW0 se actualizará en base al primero.

Siga el mismo procedimiento para el tag I1, definiendo el script:

```
IW0.01=I1;
```

Se trata ahora de escribir unas pocas expresiones que permitirán que los distintos componentes de la instalación puedan interactuar entre ellos. Es decir, el nivel del depósito deberá aumentar si la válvula VIN se abre, y disminuir cuando lo hagan VOUT1 y/o VOUT 2. Además, los contactos de los indicadores de nivel deberán cerrarse frente a determinados niveles de líquido en el depósito y, como consecuencia, las entradas del PLC que están conectadas a aquellos tendrán que activarse.

Para hacerlo, seleccione el comando *Special / Scripts / Application Scripts...* y, en la ventana de diálogo que aparece, cumplimente los campos tal como se muestra en la siguiente figura.



La primera parte del script, que se ejecuta cada 20 ms tal como se especifica en la casilla *Every*, y que es:

```
IF Q0 AND LEVEL <=9980 THEN
LEVEL=LEVEL+20;
ENDIF;
```

se encarga de aumentar en 20 unidades el nivel del depósito cuando la válvula de entrada está abierta y en él hay espacio suficiente. El código siguiente,

```
IF P1 AND LEVEL >=10 THEN
LEVEL=LEVEL - 10;
ENDIF;
IF P2 AND LEVEL >15 THEN
LEVEL=LEVEL - 15;
ENDIF;
```

sirve, por el contrario, para disminuir el nivel cuando están abiertas una o ambas válvulas en los tubos de salida y en el depósito aun hay una mínima cantidad de líquido. El nivel disminuirá en 10 unidades con la VOUT1 abierta y en 15 con la VOUT2 abierta. Así pues. la segunda tubería tiene un mayor caudal.

La última parte del script, es decir,

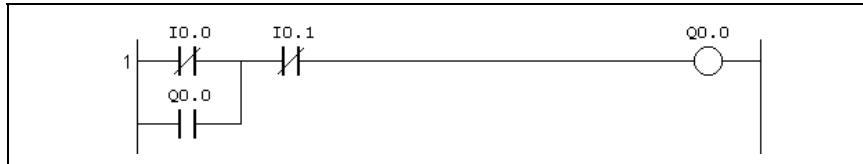
```
I0=LEVEL>7050;
I1=LEVEL>8750;
```

se encarga de la programación de las entradas del PLC conectadas a los indicadores de nivel. Las cantidades comparadas se han obtenido experimentalmente para hacer que la conmutación se produzca en correspondencia con los indicadores de nivel del dibujo. Por lo tanto, en la instalación que Ud. ha realizado los valores correctos podrían ser diferentes.

Para encontrar experimentalmente los valores correctos a insertar, actúe de la siguiente manera: cierre las válvulas de salida y llene el depósito hasta el indicador de nivel L1, forzando la salida del PLC al cual está conectado el actuador de la válvula VIN. En estas condiciones, el valor indicado por el texto de debajo del depósito es el que debe insertar para la primera comparación. Por otra parte, si sin darse cuenta supera el

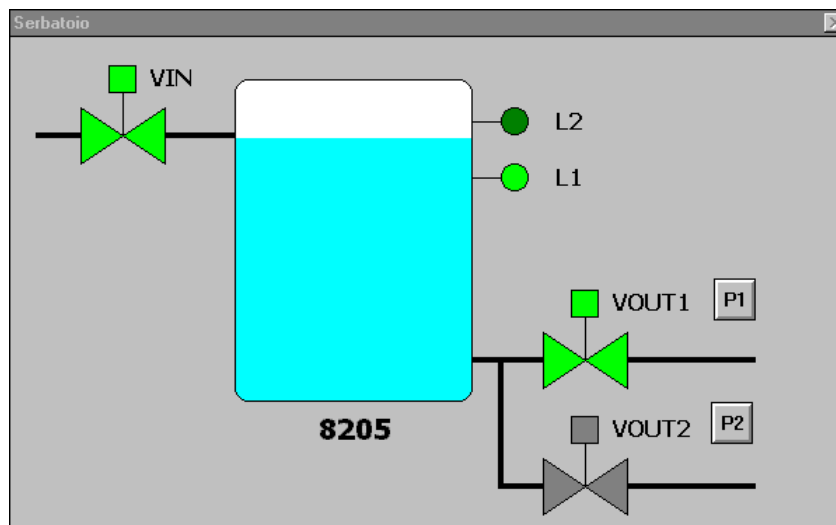
citado nivel, puede vaciar un poco el depósito accionando P1. Repita el procedimiento para L2 para obtener el segundo valor de comparación.

Escribamos ahora un programa para un PLC, por ejemplo el PS3, que controle la válvula VIN de manera que el nivel en el depósito se mantenga entre L1 y L2. La cosa es muy sencilla y se muestra en la siguiente figura.



Ponemos ahora la instalación en marcha pasando a la modalidad runtime. Arrancamos también el PLC simulado llevándolo a Run.

En la figura siguiente se muestra lo que Ud. debería ver. Puede abrir o cerrar las dos válvulas VOUT1 y VOUT2 haciendo clic sobre los respectivos pulsadores. El PLC, controlando la válvula VIN, se encargará de mantener el nivel entre los dos indicadores de nivel. Todo ello, salvo que abra a la vez las válvulas de salida. En tal caso, el caudal de la tubería de entrada será menor que el total de salida y el nivel del depósito, lenta pero inexorablemente, disminuirá.

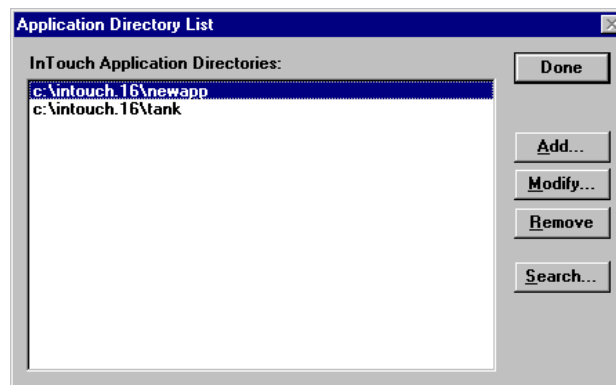


Las instalaciones descritas en este capítulo están presentes en las carpetas AW-SYS\PL-Lab\DdeTest y AW-SYS\PL-Lab\Tank del CD de instalación de AW-SYS. Para probarlos, copie toda la carpeta en el disco duro de su ordenador; luego, elimine el atributo 'Solo lectura' en todos los archivos presentes y abra la aplicación con InTouch.

Los ejemplos se han realizado usando una resolución de imagen de 800x600 pixels. Si se abren con una resolución distinta, InTouch le pedirá autorización para reconvertir dicha resolución. Después de efectuar esta operación podrían verificarse leves irregularidades en la visualización de las imágenes.

La instalación prototipo

En la carpeta AW-SYS\PL-Lab\Prototyp del CD-ROM de AW-SYS hay una instalación sin ninguna ventana, en la que ya se han implementado los tag y los script para la gestión del DDE con los PLC simulados o con los reales a través de PC-RIS. Para explotar estas potencialidades, copie la carpeta en el disco duro de su PC, y si es preciso, cámbiele el nombre para que ligue con la instalación que pretende realizar. Elimine el atributo 'Solo lectura' en todos los archivos presentes. A continuación, inicie InTouch y haga clic sobre el pulsador *Modify Directory List...* En la ventana que aparece, haga clic sobre el pulsador *Add...* y escriba el nombre de la carpeta, o búsquela haciendo clic sobre *Browse...*, luego pulse *OK* y de nuevo *Done*.



La carpeta aparece ahora en la lista; selecciónela y accione el pulsador para la puesta en marcha del *WindowMaker*.

Para generar una ventana, seleccione el comando *New Window...* en el menú *Archivo* y cumplimente los campos *Name* y *Comment* con nombres adecuados; a continuación, pulse *OK*. Se le mostrará la ventana vacía de su instalación. Se tratará ahora de dibujar los componentes y de definir las características de animación y los posibles scripts para las interconexiones. De todos modos, acuérdesese de utilizar los tags predefinidos para dar referencias a las variables de entrada y de salida del PLC.

La tabla siguiente detalla el listado de estos tags, con una breve descripción, y muestra la correspondencia con los operandos de cada uno de los PLC simulados o con el canal de la EasyPort D16 cuando se utiliza PL-RIS. Mientras las entradas y las salidas digitales están disponibles para todos, las entradas y salidas analógicas lo están en función de las peculiaridades del PLC simulado.

Tag Name	Descripción	FPC 404	PS3	S5 100U	PL-RIS
I0	Canal de entrada digital 0	I0	I0.0	E0.0	INPUT0
I1	Canal de entrada digital 1	I1	I0.1	E0.1	INPUT1
I2	Canal de entrada digital 2	I2	I0.2	E0.2	INPUT2
I3	Canal de entrada digital 3	I3	I0.3	E0.3	INPUT3
I4	Canal de entrada digital 4	I4	I0.4	E0.4	INPUT4
I5	Canal de entrada digital 5	I5	I0.5	E0.5	INPUT5
I6	Canal de entrada digital 6	I6	I0.6	E0.6	INPUT6
I7	Canal de entrada digital 7	I7	I0.7	E0.7	INPUT7
I8	Canal de entrada digital 8	I1.0	I0.8	E1.0	INPUT8
I9	Canal de entrada digital 9	I1.1	I0.9	E1.1	INPUT9
I10	Canal de entrada digital 10	I1.2	I0.10	E1.2	INPUT10
I11	Canal de entrada digital 11	I1.3	I0.11	E1.3	INPUT11
I12	Canal de entrada digital 12	I1.4	I0.12	E1.4	INPUT12
I13	Canal de entrada digital 13	I1.5	I0.13	E1.5	INPUT13
I14	Canal de entrada digital 14	I1.6	I0.14	E1.6	INPUT14
I15	Canal de entrada digital 15	I1.7	I0.15	E1.7	INPUT15
Q0	Canal de salida digital 0	O0	Q0.0	A2.0	OUTPUT0
Q1	Canal de salida digital 1	O1	Q0.1	A2.1	OUTPUT1
Q2	Canal de salida digital 2	O2	Q0.2	A2.2	OUTPUT2
Q3	Canal de salida digital 3	O3	Q0.3	A2.3	OUTPUT3
Q4	Canal de salida digital 4	O4	Q0.4	A2.4	OUTPUT4
Q5	Canal de salida digital 5	O5	Q0.5	A2.5	OUTPUT5
Q6	Canal de salida digital 6	O6	Q0.6	A2.6	OUTPUT6
Q7	Canal de salida digital 7	O7	Q0.7	A2.7	OUTPUT7
Q8	Canal de salida digital 8	O1.0	Q0.8	A3.0	OUTPUT8
Q9	Canal de salida digital 9	O1.1	Q0.9	A3.1	OUTPUT9
Q10	Canal de salida digital 10	O1.2	Q0.10	A3.2	OUTPUT10
Q11	Canal de salida digital 11	O1.3	Q0.11	A3.3	OUTPUT11
Q12	Canal de salida digital 12	O1.4	Q0.12	A3.4	OUTPUT12
Q13	Canal de salida digital 13	O1.5	Q0.13	A3.5	OUTPUT13
Q14	Canal de salida digital 14	O1.6	Q0.14	A3.6	OUTPUT14
Q15	Canal de salida digital 15	O1.7	Q0.15	A3.7	OUTPUT15
IA0	Canal de entrada analógico 0	—	IA0	EW96	-
IA1	Canal de entrada analógico 1	—	IA1	EW98	-
IA2	Canal de entrada analógico 2	—	IA2	EW100	-
IA3	Canal de entrada analógico 3	—	IA3	EW102	-
QA0	Canal de salida analógico 0	—	QA0	AW104	-
QA1	Canal de salida analógico 1	—	—	AW106	-
QA2	Canal de salida analógico 2	—	—	AW108	-
QA3	Canal de salida analógico 3	—	—	AW110	-

Los scripts preinstalados en la instalación prototipo realizan una doble función: por una parte, actualizan los tag de Q0 a Q15, los que corresponden a las salidas digitales del PLC, a cada variación de la word QW0 que los representa en el conjunto. Por otra parte, actualizan la word IW0 a cada variación de los tags de I0 a I15, aquellos que corresponden a las entradas del PLC y que la citada word idealmente van a formar.

Para acceder a estos seleccione el comando *Scripts – Data Change Scripts...* en el menú *Scripts* del *WindowMaker*.

La tabla siguiente incluye estos *scripts* y el nombre del tag cuyo evento *Change* hace que se ejecute.

TAGNAME	SCRIPT
QW0	Q0=QW0.00; Q1=QW0.01; Q2=QW0.02; Q3=QW0.03; Q4=QW0.04; Q5=QW0.05; Q6=QW0.06; Q7=QW0.07; Q8=QW0.08; Q9=QW0.09; Q10=QW0.10; Q11=QW0.11; Q12=QW0.12; Q13=QW0.13; Q14=QW0.14; Q15=QW0.15;
I0	IW0.00=I0;
I1	IW0.01=I1;
I2	IW0.02=I2;
I3	IW0.03=I3;
I4	IW0.04=I4;
I5	IW0.05=I5;
I6	IW0.06=I6;
I7	IW0.07=I7;
I8	IW0.08=I8;
I9	IW0.09=I9;
I10	IW0.10=I10;
I11	IW0.11=I11;
I12	IW0.12=I12;
I13	IW0.13=I13;
I14	IW0.14=I14;
I15	IW0.15=I15;

CAPÍTULO 8

PL-NET: INTERFACE HACIA SISTEMAS MPS

Generalidades

PL-Net es el módulo de AW-SYS que permite el control de una instalación real, con un elevado número de entradas/salidas, por medio de varios PLC simulados que se ejecutan sobre otros tantos ordenadores personales conectados en red con protocolo TCP/IP. La interface hacia la instalación se realiza mediante varias EasyPort D16, hasta un máximo de ocho, conectadas entre sí con un anillo de fibra óptica y a uno de los PC a través del puerto serie.

A cada uno de los PLC se podrá asociar lógicamente una o más interfaces, de manera que se hagan corresponder las entradas/salidas de los primeros con las de las segundas. Así, el cierre del contacto de un sensor de la instalación comportará la activación de la correspondiente entrada del PLC simulado; a la activación de una de las salidas de éste último corresponderá la activación de un actuador sobre la instalación. PL-Net se ocupará de leer, interpretar y canalizar los datos en la red y en el anillo, de un modo absolutamente transparente para el usuario.

Además, con PL-Net, los distintos PLC simulados pueden compartir informaciones, bajo la forma de word globales, permitiendo así la sincronización de diferentes partes de la instalación controladas por los diferentes PLC.

Por lo tanto, PL-Net permite la organización de un taller de automatización flexible, en el que cada alumno, o cada grupo de alumnos, podrá dedicarse a la programación de la automatización de una parte de instalación para luego comprobar el resultado con los demás. En fases sucesivas, las tareas podrán intercambiarse, incluso si continúan trabajando con el mismo PC.

Puesta en funcionamiento del sistema

Puesta en marcha de PC-Sim

PL-Net tan sólo puede funcionar en conexión con uno de los simuladores de PLC. Por ello es necesario que PC-Sim ya esté ejecutándose antes de que PL-Net se ponga en marcha.

Para iniciar PC-Sim, en cada uno de los PC de la red, haga clic sobre el pulsador Inicio de Windows, a continuación, abra la carpeta Programas\Aw-sys y en el interior de ésta, seleccione el comando necesario para arrancar uno de los simuladores de PLC. No es necesario que el tipo de PLC simulado sea el mismo en cada PC, sino que un grupo podría trabajar con el PLC FPC 404; otro, con el PS3, un tercer grupo con el S5 100U y así sucesivamente.

Puesta en marcha de PL-Net

Para poner en marcha PL-Net seleccione el *Comando PL-Net* en el *Menú Instalaciones* de PC-Sim.

PC-Sim ve a PL-Net como si fuera una de las instalaciones simuladas presentes en el paquete AW-SYS y por eso, inmediatamente después de ponerse en marcha, el *Menú Instalaciones* queda desactivado para impedir la apertura simultánea de otra instalación. Dicho menú no se reactivará hasta que se cierre PL-Net.

Ponga en marcha primero PL-Net en el ordenador al que está conectado el anillo de las EasyPort D16. Al finalizar el proceso descrito a continuación, podrá abrir PL-Net en los demás ordenadores.

Búsqueda de las EasyPort D16

Tanto si el PC con el que trabaja es el que está conectado a las EasyPort D16 como si usa uno de los otros de la red, PL-Net, cuando se inicia, busca automáticamente las EasyPort D16, realizando una exploración de los puertos en serie del PC.

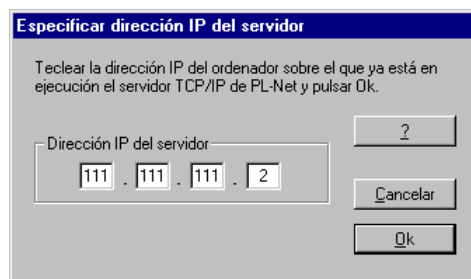
Si la búsqueda da un resultado positivo, se carga el servidor TCP/IP de PL-Net, que proporcionará los datos a la red de ordenadores. A continuación, se cargará el cliente local, que podrá acceder a los datos presentes en el servidor que se esté ejecutando en el mismo ordenador.

Si no se encuentra ninguna EasyPort D16, tan sólo se carga el cliente TCP/IP que, para acceder a los datos presentes en el servidor, deberá conocer su dirección IP.

Durante todo el proceso, el desarrollo de la búsqueda se visualiza en la *Barra de estado* de la ventana de PL-Net.

Especificación de la dirección del servidor

En el caso en que al poner en marcha PL-Net no se encuentre ninguna EasyPort D16 conectada, aparece la ventana de diálogo *Especificar dirección IP del servidor*. Inserte en las casillas correspondientes los cuatro números, con un valor comprendido entre 0 y 255, que constituyen la dirección IP del servidor en la red. El ordenador servidor es aquel al que están conectadas las EasyPort D16 y su dirección IP puede leerse haciendo doble clic sobre el icono *Red* del *Panel de control* de Windows. Una vez escrita la dirección, pulse Ok para continuar.



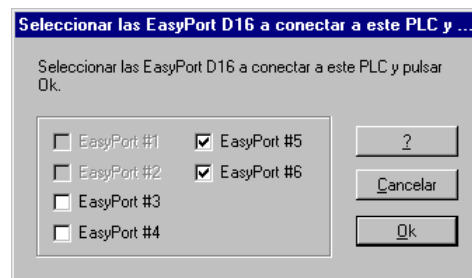
PL-Net le informará, con el correspondiente mensaje, si la dirección del servidor es errónea o si el servidor aun no se ha iniciado. Si la dirección es correcta y se establece correctamente una conexión con el servidor, PL-Net pasa a la fase siguiente.

Si PL-Net está funcionando en el servidor, el procedimiento descrito en este apartado se salta ya que el cliente reconoce el servidor que ya está en ejecución en la misma máquina y automáticamente determina la dirección IP del ordenador.

El procedimiento se concluye con la visualización, en la *Barra de estado* de PL-Net, de la dirección IP del servidor, especificada manualmente o determinada de manera automática.

Asignación de las EasyPort D16

Una vez se ha especificado, si es necesario, la dirección del servidor, el cliente lo interrogará para conocer el número de EasyPort D16 y la disponibilidad de cada una de ellas. Una vez obtenidas estas informaciones se le mostrará la ventana de diálogo *Requerir EasyPort D16* en la cual aparecen tantas casillas de control como EasyPort D16 hay conectadas al PC que actúa de servidor. Las casillas de control de las EasyPort D16 ya asignadas a otros clientes se muestran desactivadas.



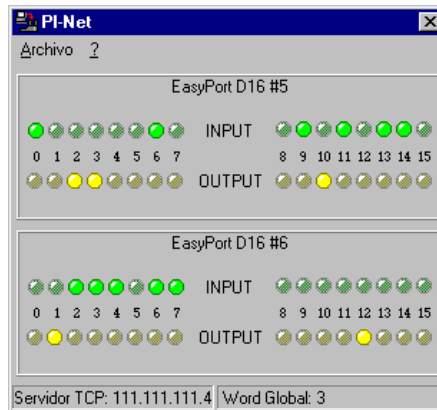
Seleccione aquellas que desea que se conecten virtualmente al PLC y pulse Ok.

Por ejemplo, en la figura anterior, hay seis EasyPort D16 que forman parte del anillo de fibra óptica: las dos primeras ya han sido asignadas por el servidor a otro cliente mientras que las siguientes están disponibles, o al menos lo eran en el momento de la pregunta. Las casillas seleccionadas especifican, para este cliente, un requerimiento de conexión lógica a las interfaces 5 y 6.

Tenga presente que el procedimiento de solicitud de las EasyPort D16 puede ejecutarse a la vez en varios clientes. Por lo tanto, una EasyPort D16 que aparece como disponible en la ventana de diálogo podría ser solicitada más tarde por otro cliente, y ser asignada a éste, antes de que Ud. finalice el proceso pulsando *Ok*. De todos modos, el servidor sólo asignará las EasyPort D16 disponibles en el momento decisivo de la petición es decir, cuando se pulsa *Ok*.

Actualización de la ventana de PL-Net

Una vez resuelto el problema de la asignación de las interfaces, la ventana de PL-Net se actualiza con la inserción de un panel de pilotos luminosos para cada una de las EasyPort D16 asignadas. Cada uno de estos paneles se identifica con el número correspondiente a la EasyPort D16 que representa e informa sobre ella con 16 LED de entrada (color verde) y 16 LED de salida (color amarillo).



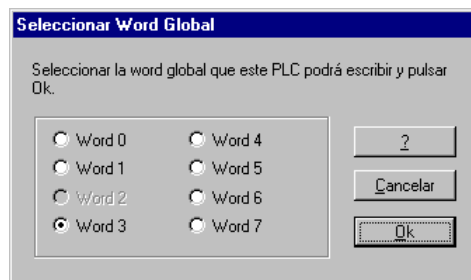
En el ejemplo de la figura, las EasyPort D16 asignadas al cliente son la número 5 y la número 6.

Elección de la word global

Las word globales son un servicio del que pueden disponer los usuarios de AW-SYS cuando se está ejecutando PL-Net a la vez en varios ordenadores conectados en red, y además está activado el protocolo TCP/IP. Las word constituyen un modo de compartir informaciones entre los PLC simulados con el principal objetivo de sincronizar los procesos que se están controlando, o parte de estos.

Las word globales que PL-Net pone a su disposición son ocho. Cada uno de los PLC simulados puede leer todas las word globales pero tan sólo puede escribir una. El número de la word que puede escribirse es distinto para cada PLC.

El cliente de PL-Net interroga el servidor acerca de la disponibilidad de word globales y le muestra la ventana de diálogo *Seleccionar Word Global*. En esta aparecen ocho pulsadores opción, tantos como words, que sirven para elegir la word que será posible escribir desde este PLC. Los pulsadores opción correspondientes a words que el servidor ya ha asignado a otros cliente aparecen desactivados.



Haga clic sobre el pulsador opción que identifica la word global que desea que sea asignada a este cliente y pulse *Ok*. Si la word global ha quedado disponible, es decir, si mientras tanto ningún otro cliente la ha

solicitado y la ha obtenido, la word será asignada al PLC en uso. La indicación del número de la word global asignada se mostrará en la *Barra de estado* de PL-Net.

Por el contrario, si mientras tanto la word ha sido asignada, PL-Net le informará de este hecho por medio de un mensaje y le mostrará nuevamente la ventana de diálogo para que seleccione otra distinta.

Las word globales constituyen un potente medio para el intercambio de datos entre PLC simulados conectados en red con el objetivo esencial de obtener la sincronización de los distintos procesos a automatizar, o de parte de estos. Para ello, suponga que un PLC necesita una información booleana que representa el final de un determinado ciclo controlado por otro de los PLC de la red. Bastará con programar (configurar) un bit global en éste último cuando el ciclo termina y controlar su estado desde el primero. Cuando el estado del bit se hace alto, el primer PLC recibirá la información de que el ciclo ha finalizado y entonces ya podrá ejecutar las acciones que sean de su competencia.

La actualización de las word globales que se leen se efectúa antes de la lectura de la imagen de las entradas para los PLC PS3 y S5 100U, y antes de la ejecución cíclica del programa del usuario, para el FPC 404. La escritura de la única word global se efectúa después de la actualización de las salidas para los primeros y después de la ejecución del programa para el último PLC.

Direccionamiento de las entradas/salidas

Las entradas y las salidas de las EasyPort D16 lógicamente conectadas a cada PLC simulado pueden direccionarse siguiendo las habituales reglas sintácticas del direccionamiento del PLC en uso. La correspondencia entre I/O del PLC simulado y número de la EasyPort D16 conectada sigue normas similares a las ya expuestas para PL-RIS.

Supongamos, por ejemplo, que estamos utilizando un PLC simulado PS3 y que hemos solicitado y obtenido para éste la conexión a las EasyPort D16 #3 y #4. La word de entrada de la EasyPort D16 #3, antes de la orden de las interfaces requeridas, corresponderá a la dirección IW0, mientras que la word de entrada de la otra interface corresponderá a la word IW1. Así, las word de salida corresponderán respectivamente a QW0 y QW1.

En las mismas condiciones, si hubiéramos usado un PLC simulado S5 100U, deberíamos haber direccionado las word de entrada con EW0 y EW8 y las de salida con AW2 y AW8.

Consulte las tablas sobre el direccionamiento de las entradas y de las salidas que aparecen en el capítulo dedicado a PL-RIS, pero tenga presente que con PL-RIS el orden de las EasyPort D16 empieza por 1 y llega hasta el número total de interfaces conectadas, mientras que con PL-Net la cantidad y el número identificativo de las interfaces conectadas depende de las selecciones que se han efectuado al iniciar el programa.

Direccionamiento de las word globales

El direccionamiento de las variables globales puede efectuarse, en los programas ladder o AWL, por word o por bit. El direccionamiento por byte no está previsto.

El direccionamiento por word se obtiene con una sigla compuesta de la siguiente forma:

- letra que identifica al operando global (G)
- letra que especifica el direccionamiento por word (W)
- número entre 0 y 7 que especifica a cual de las word globales se desea acceder.

El direccionamiento por bit se obtiene con operandos compuestos de la siguiente forma:

- letra que identifica al operando global (G)
- número entre 0 y 7 que especifica a cual de las word globales pertenece el bit
- punto de separación
- número entre 0 y 15 que especifica el bit en el interior de la word.

El tipo de direccionamiento que acepta el editor de programa depende del modelo de PLC simulado que esté utilizando. La disponibilidad para cada uno de estos se detalla en la tabla siguiente.

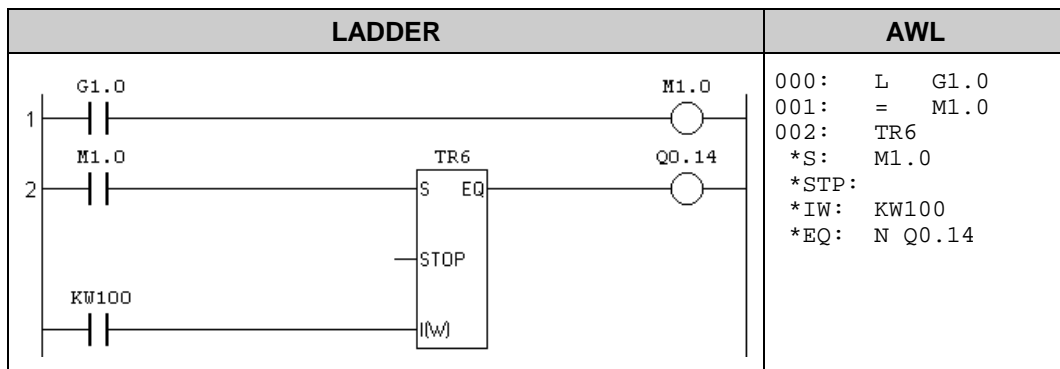
Word global	FPC 404		PS3		S5 100U	
	Bit	Word	Bit	Word	Bit	Word
0	G0.0-G0.15	-	G0.0-G0.15	GW0	-	GW0
1	G1.0-G1.15	-	G1.0-G1.15	GW1	-	GW1
2	G2.0-G2.15	-	G2.0-G2.15	GW2	-	GW2
3	G3.0-G3.15	-	G3.0-G3.15	GW3	-	GW3
4	G4.0-G4.15	-	G4.0-G4.15	GW4	-	GW4
5	G5.0-G5.15	-	G5.0-G5.15	GW5	-	GW5
6	G6.0-G6.15	-	G6.0-G6.15	GW6	-	GW6
7	G7.0-G7.15	-	G7.0-G7.15	GW7	-	GW7

Consideraciones para el uso de las word globales

PLC PS3

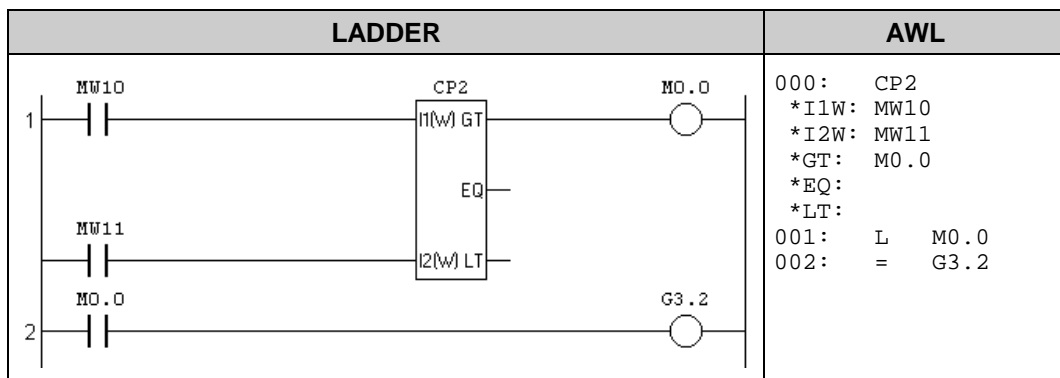
Este PLC acepta el direccionamiento de variables globales tanto por word como por bit, pero no se puede utilizar bit o word globales como entradas o salidas de los módulos del sistema. En esa coyuntura, será necesario emplear marcas (flags/merker) como variables de apoyo.

Supongamos, por ejemplo, que queremos activar el temporizador TR6 al paso a estado cierto del bit global G1.0. Los programas siguientes resuelven el problema en los dos lenguajes:

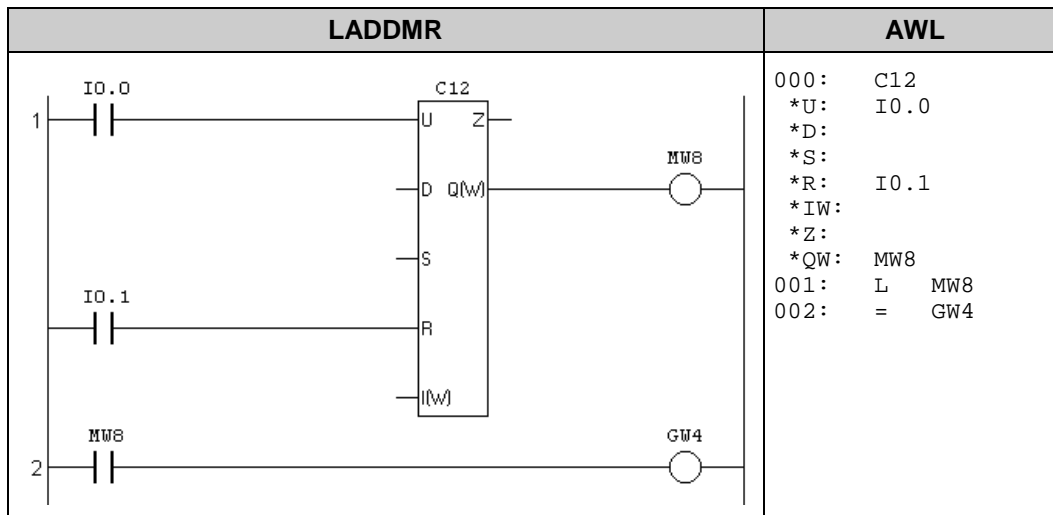


En primer lugar, el estado del bit global G1.0 se copia en el merker M1.0, y luego, éste último se usa para pilotar la entrada S del temporizador.

Deberá actuar de manera análoga si desea controlar un bit global desde una salida de un módulo de sistema. En el ejemplo siguiente se desea activar el bit global G3.2 cuando la merker word MW10 es mayor que la merker word MW11. La salida GT del comparador se conecta a un merker bit y, a continuación, el estado de ese mismo merker bit se copia en el bit global.



Veamos por último como compartir el valor de conteo de un contador: Supongamos que la word global asignada a este PLC es la GW4 y que precisamente en esa word se desea registrar el valor de conteo del contador C12. Los programas que se muestran en la figura siguiente resuelven el problema en ladder y en AWL.



El contador C12 cuenta los impulsos en la entrada I0.0 y se reinicia cuando la entrada I0.1 está cerrada. El valor de conteo, es decir, el número de impulsos contados, se deposita en la merker word MW8 a través de la salida QW del contador. A continuación, el contenido de esta salida vuelve a copiarse en la word global GW4.

PLC S5 100U

Tal como se especifica en la tabla del apartado anterior, en este PLC no es posible el direccionamiento por bit de las word globales. En el caso de que se desee efectuar este tipo de direccionamiento hay que recopiar las word globales que interesan en merker word y luego utilizar los bits de las merker word en el lugar de los bits globales. En el código del ejemplo siguiente se supone que se desea activar la salida A2.5 del PLC en función del estado del bit 2 de la word global 7.

```

: L GW7
: T MW114
: U M114.2
: = A2.5

```

En éste, se copia primero el contenido de la GW7 en la MW114. A continuación, el bit 2 de esta última, igual al bit situado en la misma posición de la word global, se utiliza para el comando de salida.

Tenga presente la diferencia de direccionamiento de bit que existe entre word globales en general y merker word en este PLC. Cuando copie la GW7 en la MW100, la parte menos significativa de la primera ocupará el byte MB100 mientras que la más significativa lo hará con el byte MB101. Así, si en el ejemplo que acabamos de ver, el bit a verificar hubiera sido el 12, hubiéramos tenido que escribir como tercera instrucción del programa U M101.4, no siendo posible escribir U M100.12.

Un procedimiento análogo debe seguirse cuando se desea escribir un bit de una word global. Supongamos que se quiere transmitir el final de un ciclo de elaboración al PLC que controla la estación de producción siguiente y que dicha información está contenida en el merker bit M3.0. Supongamos también que la word global asignada al PLC en uso para la escritura es la GW2. El siguiente código se encarga de realizar dicha tarea.

```
: U   M3.0
: =   M104.3
: L   MW104
: T   GW2
```

En éste, la MW104 se usa como variable de apoyo para la GW2. Su bit 3 se ha elegido para la escritura de la información para completar el ciclo. Si es preciso, los demás bits de la misma word podrán programarse en otro lugar del programa. Las dos primeras instrucciones se ocupan de la programación del bit de la merker word y las siguientes de transferir a ésta a la word global.

CAPÍTULO 9

REALICEMOS JUNTOS LA PRIMERA APLICACIÓN

Operaciones preliminares

En este capítulo le guiaremos, paso a paso, en la realización de su primera aplicación con AW-SYS, utilizando todos los PLC simulados: FPC 404, PS3 y S5 100U.

Nos planteamos como objetivo la realización de un ciclo cuadrado que utilice los cilindros A y B de la instalación *Electroneumática 1*. El interruptor 0.7 de la *Ventana PLC* servirá de ejecución del ciclo: activando el interruptor se iniciará el ciclo y desactivándolo el ciclo finalizará una vez que los cilindros hayan vuelto a su posición inicial.

Proceda siguiendo los pasos indicados a continuación.

1. En la barra de Inicio de Windows, abra PC-Sim para el PLC que desee utilizar, seleccionándolo entre los presentes en la carpeta Aw-sys.
2. Accione el *Pulsador Ok* para cerrar la *Ventana Informaciones sobre PC-Sim*.
3. Ponga en marcha la instalación seleccionando, en el *Menú Instalaciones*, el *Comando Electroneumática 1*.

Aparece la ventana de simulación del banco de electroneumática con cilindros de doble efecto y distribuidores biestables.

4. Abra la *Ventana PLC*, seleccionando en el *Menú Ventana* el *Comando PLC*.

En este momento podrá visualizar en el monitor tanto el PLC como la instalación a automatizar. Proceda leyendo, de entre los apartados siguientes, el dedicado al PLC que ha optado por utilizar. Después, continúe el proceso con la lectura del apartado *Test del programa*.

PLC FPC 404

Conexión de entradas y salidas

La escritura del programa del usuario deberá ir precedida por la definición de las conexiones entre los PLC y los componentes de la instalación, en este caso, los contactos de final de carrera y los solenoides de los distribuidores. Empezaremos con las señales de entrada. Más adelante nos ocuparemos de las de salida.

La tabla siguiente detalla las conexiones que realizaremos para los finales de carrera de los cilindros.

Final de carrera	Canal de entrada
a0	I0
a1	I1
b0	I2
b1	I3

Proceda de la siguiente manera:

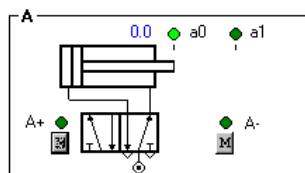
1. Haga clic sobre la instalación en correspondencia con el final de carrera negativo del cilindro A (a0).

PL-Sim responde mostrando la ventana de diálogo *Seleccionar conexión* para las entradas del PLC. El tipo de ventana a visualizar lo decide automáticamente PC-Sim ya que reconoce como entrada o como salida el componente de la instalación que haya sido seleccionado.



2. Seleccione la entrada I0.0 (módulo 0 y canal 0) y accione el *Pulsador Ok*.

La ventana se cierra y en la instalación de electroneumática, en correspondencia con el final de carrera negativo del cilindro A, se muestra la dirección de conexión en color azul, tal como se muestra en la figura siguiente.



3. Programe una tras otra las conexiones para los demás finales de carrera, tal como ha hecho en el primero, siguiendo las especificaciones de la tabla anterior.

Ocupémonos ahora de los solenoides, que conectaremos a los canales de salida del PLC, según las indicaciones de la tabla siguiente.

Solenoides	Canal de salida
A-	O0
A-	O1
B+	O2
B-	O3

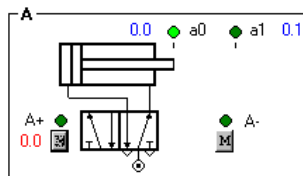
4. Haga clic sobre la instalación en correspondencia con el solenoide positivo del cilindro A (A+).



PL-Sim responde mostrando (en imágenes) la ventana de diálogo *Seleccionar conexión* para las salidas del PLC.

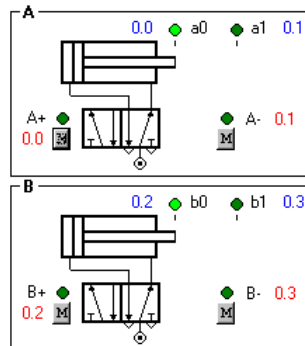
5. En ésta, seleccione la salida O0.0 (módulo 0 y canal 0) y accione el *Pulsador Ok*.

La ventana se cierra y, en la instalación de electroneumática, en correspondencia con el del solenoide positivo del cilindro A, se muestra la dirección de la conexión en color rojo, tal como se muestra en la siguiente figura.



6. Programe una tras otra las conexiones para los demás solenoides tal como ha hecho con el primero, y siguiendo las especificaciones de la tabla precedente.

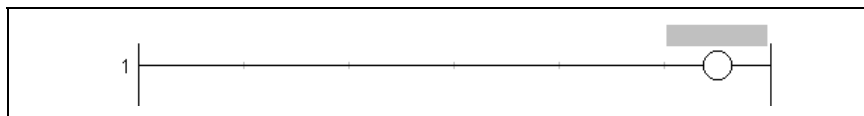
Así, hemos finalizado las conexiones. Las direcciones de conexión para los dos primeros cilindros de la instalación aparecerán tal como muestra la figura. Recordemos que el color azul identifica las conexiones a canales de entrada y el rojo, las conexiones a canales de salida.




Escritura del programa

1. En el *Menú Archivo* de PC-Sim seleccione el *Comando Nuevo proyecto ladder (FPC 404 y PS3)*.

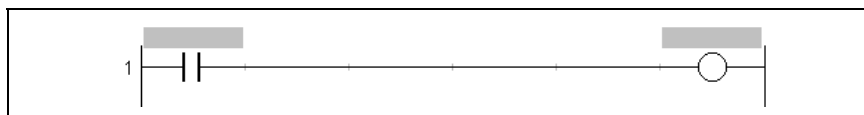
Se abrirá la ventana del editor de programa ladder en cuyo interior ya está preparada la primera línea de contactos vacía.



2. Seleccione la herramienta de inserción de contactos NA () en la *Casilla instrumentos*.

3. Haga clic sobre la línea de contactos en la primera posición de inserción;

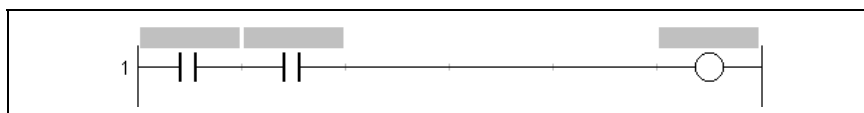
Se dibujará un contacto normalmente abierto.



4. Haga clic nuevamente sobre la segunda posición de inserción.

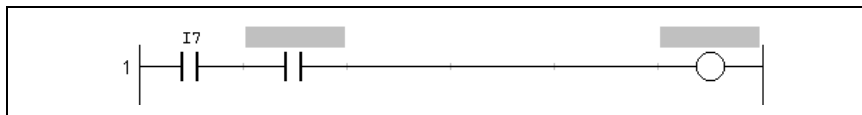
Aparecerá un segundo contacto normalmente abierto, en serie con el primero.

La imagen siguiente es la que debería aparecer en su pantalla en este momento de la fase de introducción del programa.



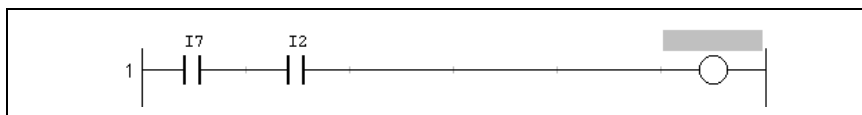
Pasemos ahora a la introducción de los operandos para cada uno de los elementos del esquema presentes: los dos contactos y la bobina.

5. Haga clic en el campo con fondo gris por encima del primer contacto.
6. Escriba el operando **I7** y pulse Intro.



El cursor de inserción de caracteres se desplaza sobre el campo del operando del contacto siguiente. Entonces, el operando del primer contacto aparece con fondo blanco.

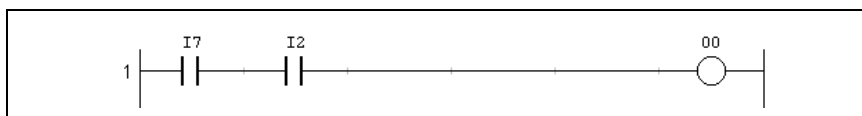
7. Escriba el operando **I2** y pulse Intro.



El cursor de inserción de caracteres se desplaza sobre el campo del operando de la bobina.

8. Escriba **O0** (letra O y número 0) y pulse Intro.

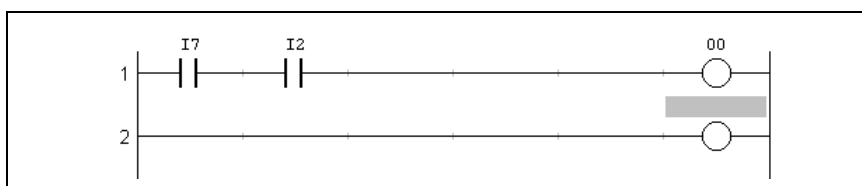
De este modo, hemos concluido el diseño de la primera línea de contactos que aparecerá así en la pantalla.



Añadamos ahora una nueva línea.

9. En el *Menú Modificar* de PC-Sim seleccione el *Comando Añadir*.

Una línea de contactos vacía se añade al final del esquema.



10. Dibuje un contacto NA en la primera posición, tal como hemos aprendido a hacer al dibujar el primer recorrido.

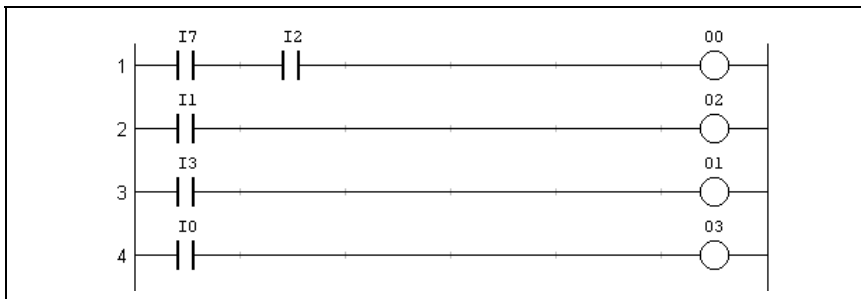
11. Escriba los operandos I1 e O2 tal como se indica en el esquema siguiente.



12. Añada otras dos líneas más, tal como se ha hecho anteriormente.

13. Dibuje los contactos.

14. Escriba los operandos I3, O1, I0 y O3, tal como se muestra en el siguiente esquema.



Así, el esquema ladder ya está completo y la descripción específica para este PLC ha finalizado. Continúe la lectura del apartado [Test del programa](#) que encontrará más adelante, en este capítulo.

PLC PS3

Conexión de entradas y salidas

La escritura del programa del usuario deberá ir precedida por la definición de las conexiones entre los PLC y los componentes de la instalación, en este caso, los contactos de final de carrera y los solenoides de los distribuidores. Empezaremos con las señales de entrada. Más adelante nos ocuparemos de las de salida.

La tabla siguiente detalla las conexiones que realizaremos para los finales de carrera de los cilindros.

Final de carrera	Canal de entrada
a0	I0.0
a1	I0.1
b0	I0.2
b1	I0.3

Proceda de la siguiente manera.

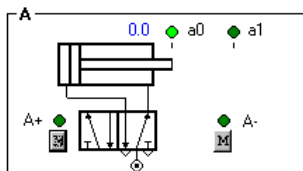
1. Haga clic sobre la instalación en correspondencia con el final de carrera negativo del cilindro A (a0).

PL-Sim responde mostrando la ventana de diálogo *Seleccionar conexión* para las entradas del PLC. El tipo de ventana a visualizar lo decide automáticamente PC-Sim ya que reconoce como entrada o como salida el componente de la instalación que se haya seleccionado.



2. Seleccione la entrada I0.0 (canal 0) y accione el *Pulsador Ok*.

La ventana se cierra y, en la instalación de electroneumática, en correspondencia con el final de carrera negativo del cilindro A, se muestra la dirección de la conexión en color azul, tal como se refleja en la figura siguiente.



3. Programe las conexiones una tras otra, para los demás finales de carrera, tal como ha hecho para el primero y según se especifica en la tabla precedente.

Ocupémonos ahora de los solenoides, que conectaremos a los canales de salida del PLC, según las indicaciones de la tabla siguiente.

Solenoide	Canal de salida
A-	Q0.0
A-	Q0.1
B+	Q0.2
B-	Q0.3

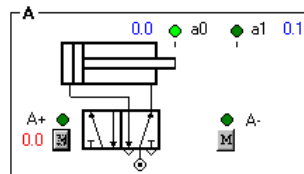
4. Haga clic sobre la instalación en correspondencia con el solenoide positivo del cilindro A (A+).



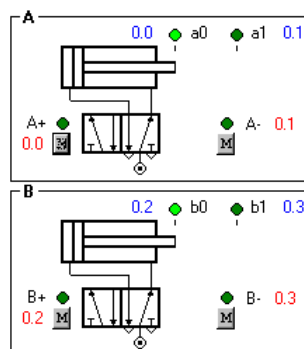
PL-Sim responde mostrando (en imagen) la ventana de diálogo *Seleccionar conexión* para las salidas del PLC.

5. En ésta, seleccione la salida Q0.0 (canal 0) y accione el *Pulsador Ok*.

La ventana se cierra y, en la instalación de electroneumática, en correspondencia con el solenoide positivo del cilindro A, se muestra la dirección de conexión en color rojo, tal como se muestra en la figura siguiente.



6. Programe una tras otra las conexiones para los demás solenoides, tal como se ha hecho para el primero, y siguiendo las especificaciones de la tabla precedente.



Así, hemos finalizado las conexiones. Las direcciones de conexión para los dos primeros cilindros de la instalación aparecerán tal como muestra la figura. Recordemos que el color azul identifica las conexiones a canales de entrada y el rojo, las conexiones a canales de salida.

Escritura del programa AWL

1. En el *Menú Archivo* de PC-Sim seleccione el *Comando Nuevo proyecto AWL (PS3)*.

Se abre la ventana del editor de programa AWL.

2. Escriba el código siguiente, pulsando Intro al final de cada instrucción.

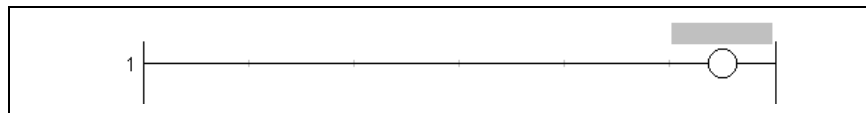
```
000: L   I0.7
001: A   I0.2
002: =   Q0.0
003: L   I0.1
004: =   Q0.2
005: L   I0.3
006: =   Q0.1
007: L   I0.0
008: =   Q0.3
```


Con esto, la fase de introducción del programa ha finalizado. Ya se puede poner en marcha el PLC y verificar la automatización. Continúe la lectura en el apartado *Test del programa*.

Escritura del programa ladder

1. En el *Menú Archivo* de PC-Sim seleccione el *Comando Nuevo proyecto ladder (FPC 404 y PS3)*.

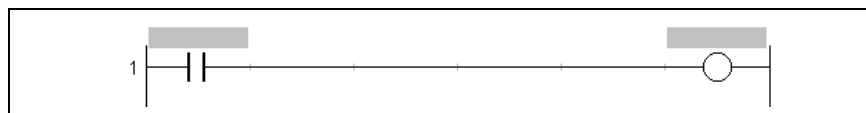
Se abre la ventana del editor de programa ladder en cuyo interior ya está preparada la primera línea de contactos vacía.



2. Seleccione la herramienta de inserción de contactos NA () en la *Casilla instrumentos*.

3. Haga clic sobre la línea de contactos en la primera posición de inserción;

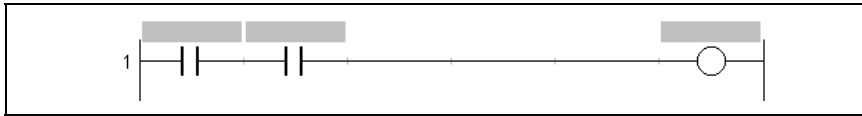
Se dibuja un contacto normalmente abierto.



4. Vuelva a hacer clic en la segunda posición de inserción.

Un segundo contacto normalmente abierto se dibuja en serie con el primero.

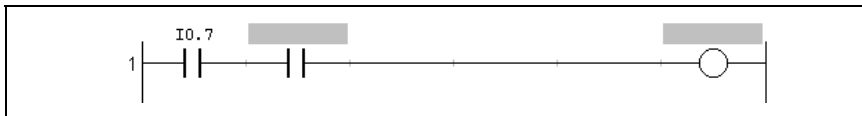
Esto es lo que debería ver en su pantalla en este momento de la fase de introducción del programa.



Pasemos a la introducción de los operandos para cada uno de los elementos presentes en el esquema: los dos contactos y la bobina.

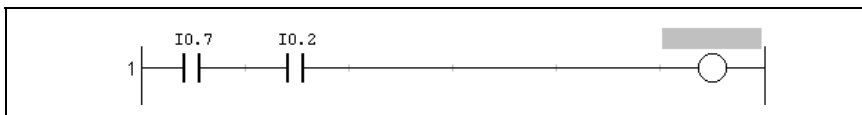
5. Haga clic en el campo con fondo gris, encima del primer contacto.

6. Escriba el operando `I0.7` y pulse Intro.



El cursor de inserción de caracteres se desplaza al campo operando del contacto siguiente. Ahora, el operando sobre el primer contacto tiene el fondo blanco.

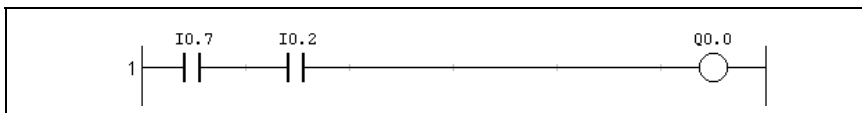
7. Escriba el operando `I0.2` y pulse Intro;



El cursor de inserción de caracteres se desplaza hasta el campo operando de la bobina.

8. Escriba `Q0.0` y pulse Intro.

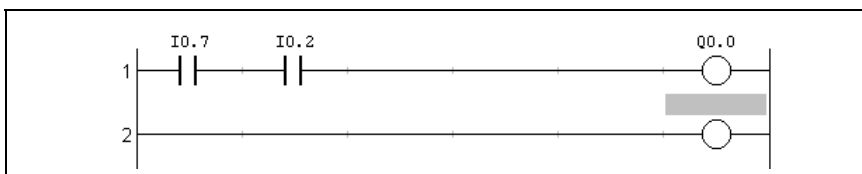
Con esto hemos finalizado el diseño de la primera línea de contactos, que aparecerá así en la pantalla.



Añadamos ahora una nueva línea de contactos.

9. En el *Menú Modificar* de PC-Sim seleccione el *Comando Añadir*.

Al final del esquema se añade una línea de contactos vacía.



10. Dibuje un contacto NA en primera posición, tal como hemos aprendido a hacer al dibujar la primera línea.

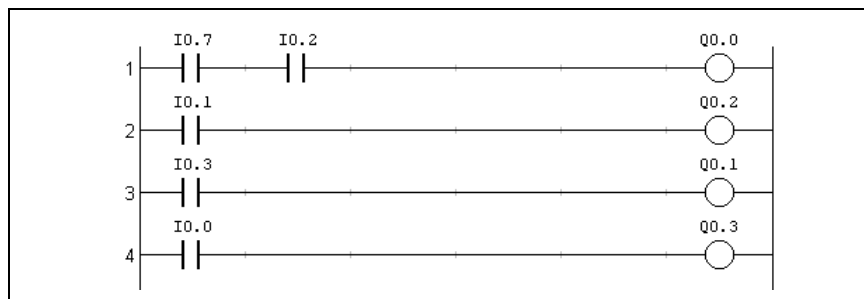
11. Escriba los operandos I0.1 y Q0.2 tal como se indica en el esquema siguiente.



12. Añada dos líneas más, tal como ha hecho anteriormente.

13. Diseñe los contactos.

14. Escriba los operandos I0.3, Q0.1, I0.0 y Q0.3, tal como muestra el siguiente esquema.



El esquema ladder está ahora completo y la descripción específica para este PLC ya ha terminado. Continúe la lectura en el apartado [Test del programa](#) que encontrará más adelante en este capítulo.

PLC S5 100U

Conexión de entradas y salidas

La escritura del programa del usuario deberá ir precedida por la definición de las conexiones entre los PLC y los componentes de la instalación, en este caso, los contactos de final de carrera y los solenoides de los distribuidores. Empezaremos con las señales de entrada. Más adelante nos ocuparemos de las de salida.

La tabla siguiente detalla las conexiones que realizaremos para los finales de carrera de los cilindros.

Final de carrera	Canal de entrada
a0	E0.0
a1	E0.1
b0	E0.2
b1	E0.3

Proceda de la siguiente manera.

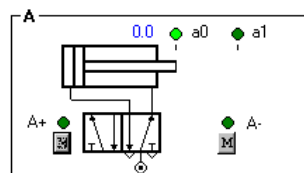
- Haga clic sobre la instalación en correspondencia con el final de carrera negativo del cilindro A (a0).

PL-Sim responde mostrando la ventana de diálogo *Seleccionar conexión* para las entradas del PLC. El tipo de ventana a visualizar lo decide automáticamente PC-Sim ya que reconoce como entrada o como salida el componente de la instalación que haya sido seleccionado.



- Seleccione la entrada E0.0 (módulo 0 y canal 0) y accione el *Pulsador Ok*.

La ventana se cierra y, en la instalación de electropneumática, en correspondencia con el final de carrera negativo del cilindro A, se muestra la dirección de la conexión en color azul, tal como se refleja en la figura siguiente.

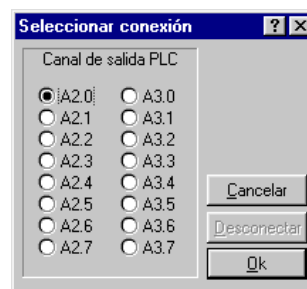


- Programe las conexiones una tras otra para los demás finales de carrera, tal como se ha hecho para el primero y siguiendo las indicaciones de la tabla anterior.

Ocupémonos ahora de los solenoides, que conectaremos a los canales de salida del PLC según la relación de la tabla siguiente.

Solenoide	Canal de salida
A+	A2.0
A-	A2.1
B+	A2.2
B-	A2.3

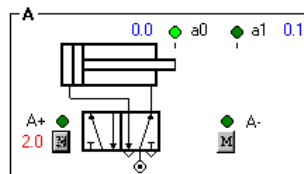
4. Haga clic sobre la instalación en correspondencia con el solenoide positivo del cilindro A (A+).



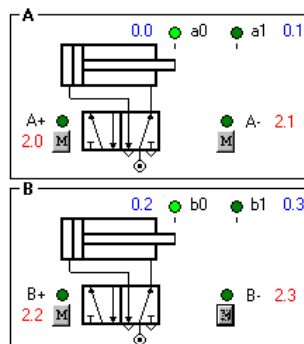
PL-Sim responde mostrando (en video) la ventana de diálogo *Seleccionar conexión* para las salidas del PLC.

5. En dicha ventana seleccione la salida A2.0 (módulo 2 y canal 0) y accione el *Pulsador Ok*.

La ventana se cierra y, en la instalación de electroneumática, en correspondencia con el solenoide positivo del cilindro A, se muestra la dirección de la conexión en color rojo, tal como se refleja en la figura siguiente.



6. Programe las conexiones para los demás solenoides, tal como se ha hecho para el primero y según lo que se especifica en la tabla precedente.



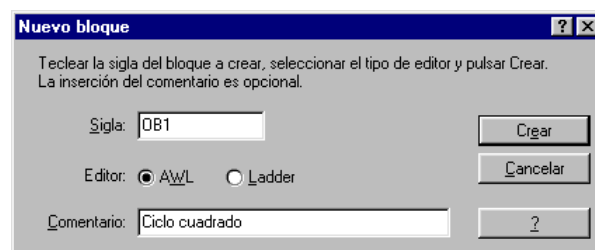
Así, hemos finalizado las conexiones. Las direcciones de conexión para los dos primeros cilindros de la instalación aparecerán tal como muestra la figura. Recordemos que el color azul identifica las conexiones a canales de entrada y el rojo, las conexiones a canales de salida.

Escritura del programa en AWL

1. En el *Menú Archivo* de PC-Sim seleccione el *Comando Nuevo proyecto (S5 100U)*.
2. En la *Ventana Lista bloques (S5 100U)* haga clic sobre *Pulsador Nuevo* o bien seleccione el *Comando Nuevo bloque (S5 100U)* en el *Menú Archivo*.

Como respuesta, PC-Sim muestra la ventana de diálogo *Nuevo bloque*.

3. Inserte los datos siguiendo lo detallado en la figura. El tipo de bloque AWL es la selección predeterminada.



4. Accione el *Pulsador Nuevo*.

Se abre el editor AWL para el bloque que se acaba de crear (OB1) y su sigla se añade a la *Tabla de bloques*.

5. Escriba el código siguiente, pulsando Intro al finalizar cada instrucción. Los espacios de separación se añaden automáticamente.

```

U   E0.7
U   E0.2
=   A2.0
U   E0.1
=   A2.2
U   E0.3
=   A2.1
U   E0.0
=   A2.3
BE

```

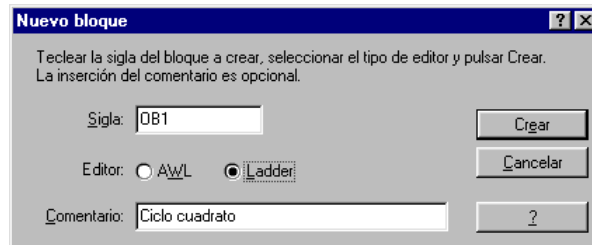
Con esto, la fase de introducción del programa ha finalizado. Ya puede ponerse en marcha el PLC y verificar la automatización. Prosiga la lectura con el apartado *Test del programa* que encontrará más adelante en este capítulo.

Escritura del programa en ladder

1. En el *Menú Archivo* de PC-Sim seleccione el *Comando Nuevo proyecto (S5 100U)*.
2. En la *Ventana Lista bloques (S5 100U)* haga clic sobre *Pulsador Nuevo* o bien seleccione el *Comando Nuevo bloque (S5 100U)* en el *Menú Archivo*.

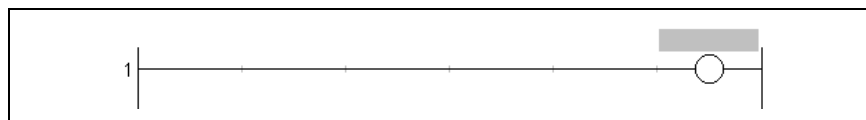
Como respuesta, PC-Sim muestra la ventana de diálogo *Nuevo bloque*.


3. Inserte los datos según las indicaciones de la figura. Observe que es preciso seleccionar *Ladder* como editor de bloque.



4. Accione el pulsador *Nuevo*.

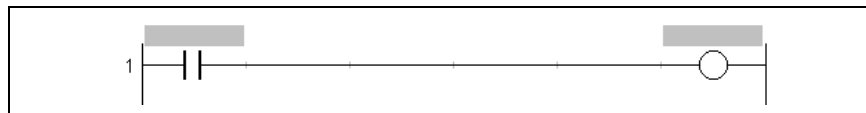
La sigla del bloque que acaba de crearse se añade a la *Tabla de bloques* y se abre la ventana de editor ladder para el bloque especificado así como y la *Casilla instrumentos*. En el editor ya está presente el la primera línea de contactos vacía.



5. Seleccione la herramienta de inserción de contactos NA () en la *Casilla instrumentos*.

6. Haga clic sobre la línea en la primera posición de inserción;

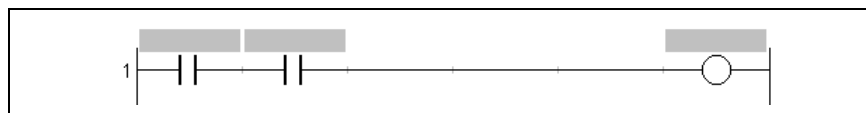
Se dibuja un contacto normalmente abierto.



7. Haga clic nuevamente en la segunda posición de inserción.

Un segundo contacto normalmente abierto se dibuja en serie con el primero.

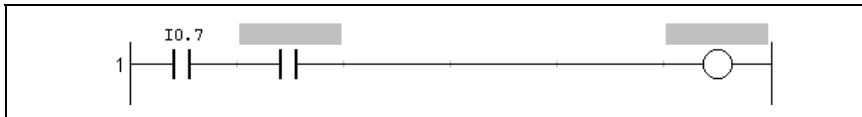
Esto es lo que debería ver en la pantalla en este momento de la fase de edición.



Pasemos a la inserción de los operandos para cada uno de los elementos presentes en el esquema: los dos contactos y la bobina.

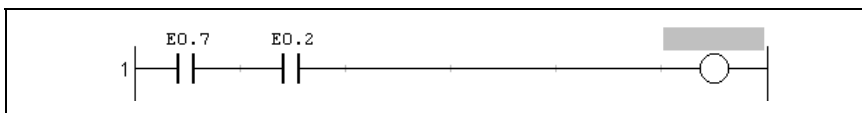
8. Haga clic en el campo con fondo gris, encima del primer contacto.

9. Escriba el operando **E0.7** y pulse Intro.



El cursor de inserción de caracteres se desplaza hasta el campo operando del contacto siguiente. El operando sobre el primer contacto tiene ahora un fondo blanco.

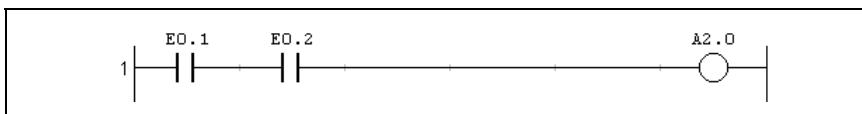
10. Escriba el operando **E0.2** y pulse Intro.



El cursor de inserción de caracteres se desplaza hasta el campo operando de la bobina.

11. Escriba **A2.0** y pulse Intro.

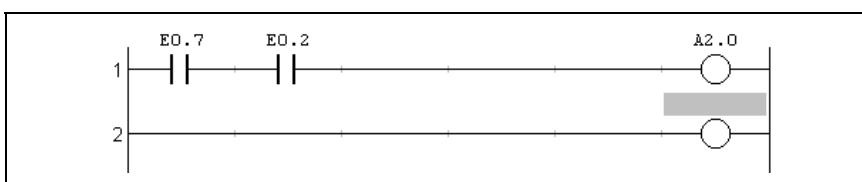
Con esto hemos terminado el diseño de la primera línea de contactos que le aparecerá así en la pantalla.



Añadamos ahora una nueva línea.

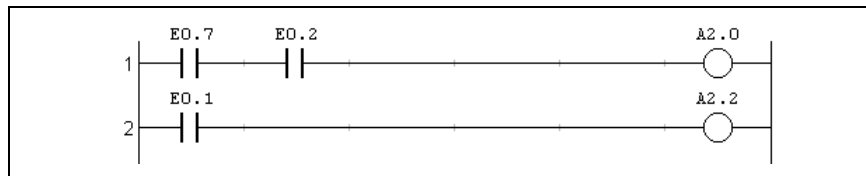
12. En el *Menú Modificar* de PC-Sim seleccione el *Comando Añadir*.

Al final del esquema se añade una línea de contactos vacía.



13. Diseñe un contacto NA en la primera posición, tal como hemos aprendido a hacer al diseñar la primera línea de contactos.

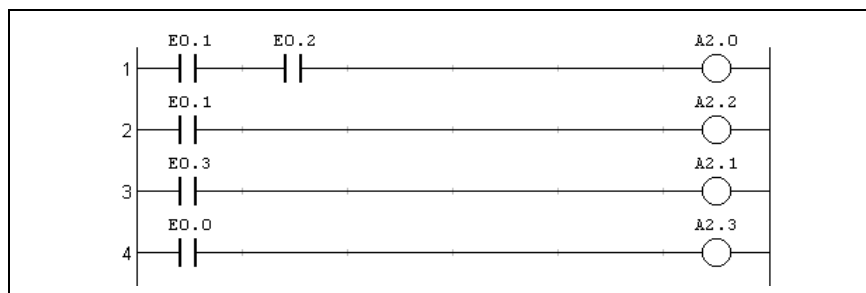
14. Escriba los operandos **E0.1** y **A2.2** tal como hemos indicado en el esquema siguiente.






15. Añada dos líneas más, tal como ha hecho anteriormente.

16. Diseñe los contactos.

17. Escriba los operandos **E0.3**, **A2.1**, **E0.0** y **A2.3**, tal como se muestra en el último esquema que aparece a continuación.



Test del programa


1. Proporcione alimentación al PLC virtual accionando el pulsador *CPU On* () en la *Barra instrumentos* de PC-Sim.
2. Ponga en marcha el PLC virtual accionando el pulsador *CPU Run* () en la misma barra.
3. Inicie el ciclo accionando el interruptor 0.7 () en la *Ventana PLC*.

Si todos los pasos anteriores se han realizado correctamente, verá los cilindros A y B moviéndose según un ciclo cuadrado, mientras el interruptor 0.7 permanezca conectado.



4. Detenga el ciclo desconectando el interruptor 0.7.
5. Espere a que los cilindros se sitúen en la posición inicial para confirmar la parada del ciclo.
6. Reinicie el ciclo volviendo a accionar el interruptor 0.7.
7. Intente activar el debug: pulse el *Pulsador Debug* de la *Ventana Lista bloques (S5 100U)* o bien seleccione en el *Menú Ventana* el *Comando Debug (FPC 404 y PS3)*.

Si ha realizado la programación en AWL, en la ventana de editor, en correspondencia con cada instrucción, podrá leer el estado del operando y el valor activo de los registros.

En cambio, si ha programado en ladder, el esquema de contactos se animará: el color de los contactos, de las bobinas y de las líneas de corriente variará en el tiempo, y le indicará los contactos y las bobinas activados y los recorridos atravesados por la corriente.

8. Intente ahora variar la base de tiempos de la instalación, actuando sobre el selector de velocidades () en la *Barra de instrumentos* de la instalación.

Intente de nuevo el funcionamiento por pasos.

9. Accione el *Pulsador de Stop* () en la *Barra de instrumentos* de la instalación.
10. Presione repetidamente el *Pulsador de paso* ()

A cada clic, la instalación avanzará un paso, permitiéndole así examinar con atención el funcionamiento del ciclo.

Ya está Ud. preparado para experimentar otros ciclos con ésta y con las otras instalaciones de electroneumática así como con las automatizaciones de las demás instalaciones de PL-Sim. Recuerde que, en caso de dificultad, podrá consultar la Guía on line de PC-Sim y la de PL-Sim o bien el presente manual. Además, podrá cargar los numerosos ejemplos que se facilitan junto con las instalaciones.

Si no consigue resolver el problema de automatización que esté realizando, contacte con el *Servicio de Soporte Técnico para Usuarios de AW-SYS* de Autoware, estaremos encantados de ayudarle.

CAPÍTULO 10

NOTAS SOBRE LA PROGRAMACIÓN DEL PLC FPC 404

Premisa

Las notas facilitadas en esta sección se refieren únicamente al PLC simulado, es decir, a PC-Sim para FPC 404. Sin embargo, muy a menudo, y precisamente por el hecho de que en toda la simulación con AW-SYS se procura un acercamiento lo mayor posible a la realidad, también pueden ser útiles para el PLC real. Le aconsejamos que utilice los manuales originales del PLC y de los dispositivos de programación para obtener todas las informaciones necesarias para la programación del dispositivo real.

Algunos de los temas tratados, por su naturaleza, son comunes a todos los PLC simulados. Sin embargo, se ha preferido repetir las mismas descripciones para todos ellos, con el fin de convertir a cada uno de estos capítulos en una unidad autónoma: un cómodo e indispensable manual para tenerlo al alcance de la mano durante la programación del PLC simulado.

Generalidades sobre el PLC FPC 404

El FPC 404 es un PLC modular y, en la simulación, está constituido por los siguientes módulos:

Nº del módulo Tipo

0	CPU + 8 entradas digitales + 8 salidas digitales
1	8 entradas digitales + 8 salidas digitales

La CPU está dotada de un conmutador de encendido (On-Off) y de un conmutador del modo de funcionamiento (Stop-Run). Dos pilotos luminosos, STOP de color amarillo y RUN de color verde, señalan el estado en vigor de la CPU. El encendido del LED amarillo indica que la CPU recibe alimentación y se encuentra en estado de STOP. El encendido del LED verde indica que la CPU recibe alimentación y se encuentra en estado de RUN.

Cada entrada y cada salida digital dispone de un piloto indicador de estado: piloto encendido para el estado activo, piloto apagado para el estado inactivo.

Direccionamiento

Por direccionamiento se entiende la serie de reglas sintácticas que deben observarse al referirse a variables o constantes durante la escritura de un programa para PLC. El direccionamiento en la simulación del PLC FPC 404 se efectúa de acuerdo con las reglas que se exponen a continuación y que están resumidas en la [Tabla 8](#).

Entradas - Salidas

La sigla para el direccionamiento de las entradas y de las salidas en este modelo de PLC está compuesta por:

- letra que identifica las entradas (I) o las salidas (O);
- número de módulo;
- punto de separación;
- número de canal.

En el direccionamiento del I/O del módulo 0 se usa la forma abreviada con la omisión del número del módulo y del punto de separación.

Por ejemplo:

I1.4 identifica el canal de entrada 4 del módulo 1;

O1.3 identifica el canal de salida 3 del módulo 1;

I2 identifica el canal de entrada 2 del módulo 0;

O3 identifica el canal de salida 3 del módulo 0;

Así, resumiendo:

las 8 entradas del módulo 0 corresponden a las direcciones abreviadas **I0 ÷ I7**;

las 8 entradas del módulo 1 corresponden a las direcciones **I1.0 + I1.7**;

las 8 salidas del módulo 0 corresponden a las direcciones abreviadas **O0 + O7**;

las 8 salidas del módulo 1 corresponden a las direcciones **O1.0 ÷ O1.7**.

Flag

La memoria flag está compuesta por 16 word. Cada bit de esta memoria puede direccionarse de uno en uno con las siglas compuestas de la siguiente manera:

- letra que identifica los flag (F);
- número de word;
- punto de separación;
- número de bit.

Así pues, el campo de los flag direccionables está entre **F0.0** y **F15.15**.

Mención aparte requiere el flag **FI**. Con esta sigla se puede direccionar un flag que el sistema se encarga de programar a 1 al inicio del primer ciclo de elaboración del programa y de llevar a cero en el ciclo siguiente, para permanecer en este estado hasta que se inicie de nuevo. Este flag solo puede utilizarse con operaciones de test, es decir, como operando de contactos, y se emplea para las programaciones iniciales que deban efectuarse en el paso a RUN.

Temporizadores

En el PLC simulado se han implementado los 40 temporizadores presentes en el real. Las direcciones van de T0 a T39.

Contadores

En el PLC simulado se han implementado los 48 contadores presentes en el real. Sus direcciones van de C0 a C47.

Tabla 8: resumen de los operandos para PLC FPC 404 simulado

Tipo	Q.tà	Min	Max
Entradas a bit	16	I0 (I0.0)	I1.7
Salidas a bit	16	O0 (O0.0)	O1.7
Flag bit	256	F0.0	F15.15
Flag primer ciclo	1	FI	
Temporizadores	40	T0	T39
Contadores	48	C0	C47

Elaboración del programa del usuario

Sistema operativo

El corazón del funcionamiento del PLC es su Sistema Operativo: un programa que gestiona el funcionamiento de base de los dispositivos, grabado en ROM por la casa constructora.

El examen detallado y exhaustivo de las funciones del Sistema Operativo no forma parte de la misión de este manual, y también en los manuales de los PLC reales es difícil encontrar aclaraciones detalladas al respecto. Por otra parte, la ejecución de este programa es completamente transparente para el programador que, en algunos aspectos, puede incluso ignorar su existencia.

Sin embargo, y como complemento, relacionamos algunas de las funciones que dicho sistema operativo realiza y que consideramos las más importantes para la comprensión de los apartados que seguirán:

- control del tiempo de ciclo (watch-dog)
- test del hardware y de la memoria
- llamada del programa del usuario
- gestión de la programación

Estado de la CPU

La CPU puede encontrarse en uno de los estados siguientes:

OFF: el PLC no recibe alimentación, el conmutador de encendido en la *Barra de instrumentos* de PC-Sim se encuentra en posición de Off. Los pilotos luminosos Stop y Run, situados en la parte frontal del PLC están apagados.

STOP: el PLC recibe alimentación pero el programa del usuario no se ejecuta, el conmutador de encendido en la *Barra de instrumentos* de PC-Sim se encuentra en la posición de On y el conmutador de funcionamiento se encuentra en la posición de Stop. El piloto de Stop está encendido, y el piloto de Run está apagado.

RUN: El PLC recibe alimentación y se ejecuta el programa del usuario, el conmutador de encendido en la *Barra de instrumentos* de PC-Sim se encuentra en la posición de On y el conmutador de funcionamiento se encuentra en la posición de Run. El piloto Stop está apagado, y el piloto Run esta encendido.

El estado en vigor de la CPU se visualiza en la barra del título de la *Ventana PLC* encerrado entre corchetes.

Control del tiempo de ciclo

El sistema operativo, justo antes de efectuar la llamada al programa del usuario, pone en marcha un monoestable que, por lo general, tiene una duración que va de unas décimas a unos poquísimos centenares de milésimas de segundo. Supongamos que el programa del usuario requiera, para su ejecución, un tiempo inferior al del monoestable. En esta situación, antes de cada una de las sucesivas llamadas, el monoestable se inicia de nuevo, como resultado, la salida de este último se mantiene continuamente alta.

En cambio, si el programa del usuario requiriese, para su ejecución, más tiempo que el que dura el monoestable, entonces la salida de éste sí que llevaría al estado bajo, provocando el paso inmediato del PLC a Stop.

Un programa mal realizado podría llevar a la CPU a ejecutar indefinidamente, o durante un tiempo excesivo para los fines de rapidez en el control de la instalación, un ciclo cerrado de instrucciones sin volver a la elaboración del sistema operativo. Esta condición podría provocar daños a la instalación y un grave peligro para el personal que está trabajando. El control del tiempo del ciclo, que en la literatura técnica especializada podrá encontrar también bajo el nombre de *watch-dog*, evita precisamente este problema.

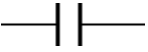
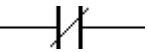



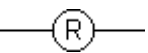
PC-Sim también controla que su programa no genere un ciclo cerrado pero, en este caso, las múltiples y cambiantes tareas de un PC, imponen tiempos más largos para la detección del problema, del orden de algunos segundos.

Programación en Ladder

La escritura de un programa para PLC en Ladder diagram o Esquema de contactos se realiza de manera similar al diseño de un circuito eléctrico con relés. Entre dos líneas verticales, dispuestas en los márgenes

del esquema, que representan la alimentación del circuito, pueden colocarse los elementos del circuito eléctrico dispuestos sobre líneas horizontales (que a partir de ahora llamaremos ramales) conectados entre ellos por líneas horizontales y verticales.

Tabla 9: elementos del ladder para FPC 404 para entradas, salidas y flag

Elemento	Descripción
Zona condicional	
	Contacto normalmente abierto , test sobre el estado 1 del operando. El contacto sobre el esquema se cierra cuando el operando correspondiente se encuentra en el estado lógico alto. Por ejemplo, si el operando es una entrada digital del PLC, el contacto está cerrado cuando la entrada está cerrada y viceversa.
	Contacto normalmente cerrado , test sobre el estado 0 del operando. El contacto sobre el esquema se cierra cuando el operando correspondiente se encuentra en el estado lógico bajo. Por ejemplo, si el operando es una entrada digital del PLC, el contacto está cerrado cuando la entrada está abierta y viceversa.
Zona ejecutiva	
	Bobina directa Si la bobina está activada, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico alto. Si la bobina no está activada, al operando se le asigna el valor lógico bajo.
	Bobina inversa Si la bobina está activada, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico bajo. Si la bobina no está activada, al operando se le asigna el valor lógico alto.
	Bobina de set Si la bobina está activa, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico alto. Si la bobina no está activa, no se realiza ninguna acción sobre el estado del operando.
	Bobina de reset Si la bobina está activa, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico bajo. Si la bobina no está activa, no se realiza ninguna acción sobre el estado del operando.

Cada ramal puede contener como máximo 6 elementos del circuito en unas posiciones muy determinadas. En las 5 primeras posiciones pueden diseñarse tan sólo contactos y en la última, solamente bobinas.

Cada ramal puede subdividirse en dos zonas: la zona condicional, formada por las 5 primeras posiciones, que contiene las condiciones que deben cumplirse para realizar una acción determinada, y la zona ejecutiva, la última posición, que contiene las acciones a realizar cuando se cumplen todas las condiciones.

Esto es igual a lo que sucede en un circuito eléctrico: una carga eléctrica representa la parte ejecutiva de un ramal. Los contactos en serie o en paralelo, de tipo normalmente cerrados o abiertos, representan la parte condicional.

Cada contacto puede ponerse en paralelo solo o por grupos, diseñando contactos sobre el ramal siguiente que se conectarán al ramal base, o al ramal paralelo anterior, a través de líneas verticales. El máximo número de ramales paralelos permitidos es de 4. El conjunto del ramal base y de todos los ramales paralelos conectados a estos se detectará a continuación como recorrido de corriente. El editor numera cada recorrido de corriente automática y correlativamente

También las bobinas pueden conectarse en paralelo pero deben tener la entrada en común. Es decir, un recorrido de corriente puede enviar una única señal lógica a todas las bobinas.

En el interior del esquema de contactos pueden situarse los elementos descritos en la [Tabla 9](#).

Operaciones de temporización

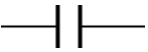
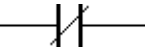




La inicialización de un tiempo se efectúa utilizando una bobina e introduciendo como operando una sigla válida para temporizador.

Si la operación de inicialización del tiempo se efectúa con la bobina de set se generará un impulso prolongado. En el flanco de activación de la bobina, el valor de preset se carga en el temporizador y se inicia el conteo del tiempo. El estado del temporizador está en 1 durante todo el transcurso del tiempo. Transcurrido éste, el estado se lleva a 0.

En cambio, si la operación de inicialización del tiempo se efectúa con la bobina directa se generará un impulso. En el flanco de activación de la bobina, el valor de preset se carga en el temporizador y se inicia el conteo del tiempo. El estado del temporizador está en 1 durante todo el transcurso del tiempo. Una vez ha transcurrido, el estado se pone en 0. Si durante dicho transcurso, la bobina se desactiva, el tiempo se borra y el estado se sitúa inmediatamente en 0.

El test sobre el estado del temporizador se efectúa utilizando un contacto (NA o NC) al cual se asigna como operando la sigla del temporizador.

Tabla 10: elementos del ladder para FPC 404 para temporizadores

Elemento	Descripción
Zona condicional	
	Contacto normalmente abierto , test sobre el estado 1 del operando. El contacto en el esquema se cierra cuando el operando temporizador se encuentra en el estado lógico alto.
	Contacto normalmente cerrado , test sobre el estado 0 del operando. El contacto en el esquema se cierra cuando el operando temporizador se encuentra en el estado lógico bajo.
Zona ejecutiva	
	Bobina directa Cuando se activa la bobina (flanco ascendente), es decir cuando se cierra una cadena de contactos hacia la línea izquierda de alimentación, el tiempo se inicia. Durante el transcurso del tiempo el estado del temporizador es alto. A la apertura de la cadena o al término del tiempo, el estado del temporizador se resitúa en cero.
	Bobina inversa Cuando se desactiva la bobina (flanco descendente), es decir, al abrir una cadena de contactos hacia la línea izquierda de alimentación, el tiempo se inicia. Durante el transcurso del tiempo, el estado del temporizador es alto. Al cierre de la cadena o al acabar el tiempo, el estado del temporizador se resitua en 0.
	Bobina de set Cuando se activa la bobina (flanco ascendente), es decir cuando se cierra una cadena de contactos hacia la línea izquierda de alimentación, el tiempo se inicia. Durante el transcurso del tiempo el estado del temporizador es alto. Al término del tiempo, el estado del temporizador se resitua en cero.
	Bobina de reset Cuando se activa la bobina (flanco ascendente), es decir, al cerrar una cadena de contactos hacia la línea izquierda de alimentación, el tiempo se repone y el estado se lleva a 0.

Operaciones de conteo

Las operaciones de conteo se programan utilizando distintos tipos de bobina y haciéndoles corresponder operandos válidos de contadores.

Con este PLC pueden efectuarse operaciones de conteo tanto hacia adelante como hacia atrás, utilizando los 47 contadores implementados. Es preciso tener presente las siguientes reglas:

- La activación de una bobina de set con operando contador repone el valor de conteo y lleva el estado a 1.
- La activación de una bobina de reset con operando contador actúa tan solo sobre el estado, llevándolo a 0.
- La activación de una bobina de preset con operando contador carga como conteo el valor de preset y lleva el estado a 1.
- La activación de una bobina de decremento con operando contador disminuye en 1 el valor de conteo
- La activación de una bobina de incremento con operando contador aumenta en 1 el valor de conteo.
- El test sobre el estado del contador se efectúa utilizando un contacto (NA o NC) al cual se asigna el operando del contador.

Conteo hacia adelante

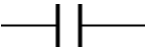
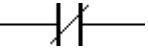



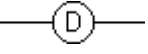

Para construir un contador para conteo hacia adelante, es necesario ejecutar, antes que nada, un set del mismo. Esta operación lleva el valor de conteo a 0 y la salida a 1. Incrementando el valor de conteo, el estado sigue manteniéndose en 1 hasta que se ha alcanzado el valor de preset, momento en que la salida pasa a 0.

Conteo hacia atrás

Para construir un contador para conteo hacia atrás, es necesario ejecutar antes un preset del mismo.

Esta operación programa el valor de conteo al de preset y la salida a 1. A partir de ahí, se puede decrementar el valor. El estado permanece en 1 hasta que el valor de conteo alcanza el 0, momento en que la salida pasa a 0.

Tabla 11: elementos del ladder para FPC 404 para contadores

Elemento	Descripción
Zona condicional	
	Contacto normalmente abierto , test sobre el estado 1 del operando. El contacto sobre el esquema se cierra cuando el operando contador se encuentra en el estado lógico alto.
	Contacto normalmente cerrado , test sobre el estado 0 del operando. El contacto sobre el esquema se cierra cuando el operando contador se encuentra en el estado lógico bajo.
Zona ejecutiva	
	Bobina de set A la activación de la bobina (flanco ascendente), es decir, cuando se cierra una cadena de contactos cerrados hacia la línea izquierda de alimentación, el valor de conteo del contador indicado como operando se pone a cero y su estado se pone en 1.
	Bobina de reset A la activación de la bobina (flanco ascendente), es decir, cuando se cierra una cadena de contactos cerrados hacia la línea izquierda de alimentación, el estado del contador indicado como operando se repone a 0.
	Bobina de incremento A la activación de la bobina (flanco ascendente), es decir, cuando se cierra una cadena de contactos hacia la línea izquierda de alimentación, el valor de conteo del contador indicado como operando se incrementa en 1.
	Bobina de decremento A la activación de la bobina (flanco ascendente), es decir, cuando se cierra una cadena de contactos hacia la línea izquierda de alimentación, el valor de conteo del contador indicado como operando disminuye en 1.
	Bobina de preset A la activación de la bobina (flanco ascendente), es decir, cuando se cierra una cadena de contactos hacia la línea izquierda de alimentación, el valor de conteo del contador indicado como operando se pone en el valor de preset.

La programación del valor de preset de un contador en el PLC real se produce de manera diferente. En la línea de uniformizar lo máximo posible los distintos simuladores de PLC, en especial respecto a la metodología de programación ladder, se ha optado por introducir la bobina de preset que, en estos términos concretos, no tiene una correspondencia en el lenguaje ladder del FPC 404 real.

CAPÍTULO 11

EJEMPLOS DE PROGRAMACIÓN PARA PLC FPC 404

En este capítulo se plantean, ordenados de menor a mayor complejidad, numerosos ejercicios de programación para el PLC FPC 404, cuya solución se facilita en forma de esquema de contactos.

Para comprobar su funcionamiento tan sólo serán necesarios los interruptores o pulsadores para el control de las entradas del PLC. Los pilotos luminosos situados en la parte frontal del dispositivo valorarán el estado de las salidas. Todos los elementos necesarios se encuentran en la ventana PLC de PC-Sim.

Si durante la instalación de AW-SYS ha optado por instalar también los ejemplos, encontrará todos los que se presentan aquí en la carpeta Aw-sys\Fpc404\Esempi de su ordenador. Todos ellos pueden cargarse y verificarse con PC-Sim para FPC-404.

El nombre del archivo proyecto a cargar tiene el siguiente formato Fxx-yyyL.prj, donde:

- F identifica un proyecto para FPC 404;
- xx son dos cifras que equivalen siempre a 00;
- yyy es el número correlativo del ejemplo expresado siempre con tres dígitos.
- L especifica que se trata de un proyecto ladder.

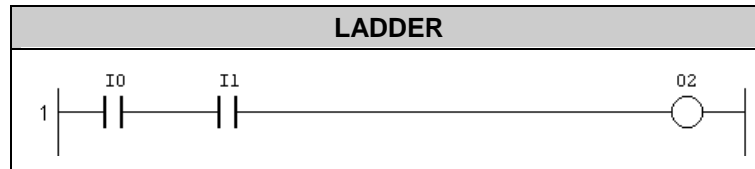
Es decir, si desea cargar el ejemplo de programación 8, abra el proyecto F00-008L.prj.

En esa misma carpeta podrá encontrar también otros archivos proyecto, en los que las cifras xx son distintas a 00. En ese caso, los ejemplos se refieren a las instalaciones y xx indica el número correspondiente a la instalación, tal como aparece en el [Menú Instalaciones](#). En la *Guía on line* de PC-Sim podrá encontrar información sobre como cargar y probar los citados ejemplos.

Ejemplo 1: combinación AND

Realizar $O2 = I0 \text{ AND } I1$

La salida $O2$ debe activarse única y exclusivamente si los interruptores conectados a las entradas $I0$ y $I1$ están cerrados.

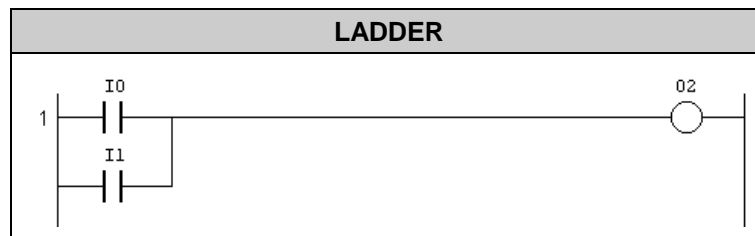


La solución se obtiene poniendo en serie dos contactos, con operandos $I0$ y $I1$, y la bobina $O2$. De hecho, la combinación lógica AND, traducida al lenguaje ladder, equivale a la serie de dos contactos: en la disposición en serie 'se lee' el cierre del circuito tan sólo cuando ambos contactos están cerrados. Por lo tanto, ésta es la única condición que activa la bobina.

Ejemplo 2: combinación OR

Realizar $O2 = I0 \text{ OR } I1$

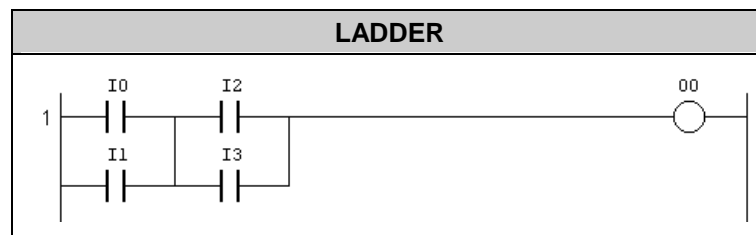
La salida $O2$ debe activarse si al menos uno de los interruptores conectados a las entradas $I0$ o $I1$ está cerrado.



La solución se obtiene poniendo en serie con la bobina $O2$, el paralelo de dos contactos, con operandos $I0$ y $I1$. De hecho, la combinación lógica OR, traducida al lenguaje ladder, equivale al paralelo de dos contactos: a la salida de la conexión en paralelo 'se lee' el cierre del circuito cuando al menos uno de los contactos está cerrado. Por tanto, esta es la única condición que conduce a la activación de la bobina.

Ejemplo 3: combinación AND de OR

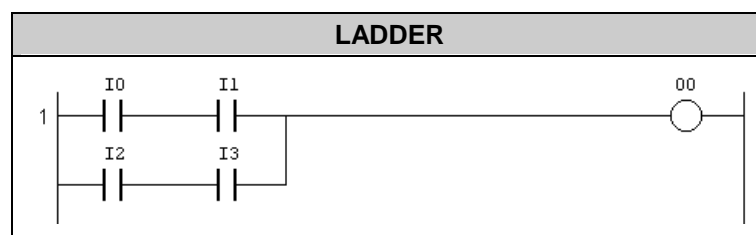
Realizar $o7 = (I0 \text{ OR } I1) \text{ AND } (I2 \text{ OR } I3)$



La solución debería conseguirse de manera intuitiva después de haber realizado los ejercicios precedentes: se disponen en serie (AND) dos paralelos (OR) de contactos, conectando adecuadamente los operandos correspondientes a estos y a la bobina.

Ejemplo 4: combinación OR de AND

Realizar $o7 = (I0 \text{ AND } I1) \text{ OR } (I2 \text{ AND } I3)$. Donde el paréntesis, aunque no sean necesarios por cuanto la operación AND tiene preferencia sobre la OR, se han añadido para mayor claridad.

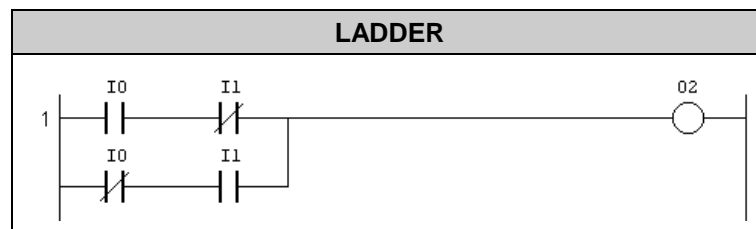


La solución propuesta dispone en paralelo (OR) dos serie (AND) de contactos, conduciendo adecuadamente los operandos en correspondencia con estos y con la bobina.

Ejemplo 5: combinación XOR

Realizar $o2 = I0 \text{ XOR } I1$

La operación lógica XOR aplicada a dos variables booleanas da un resultado cierto sólo cuando una de las dos variables es cierta.

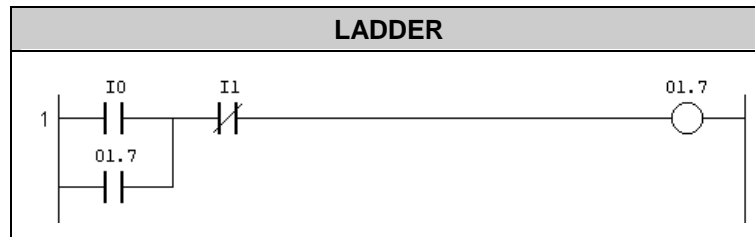


La primera serie de contactos del programa ladder está cerrada sólo cuando I0 está cerrado y I1 está abierto. Por el contrario, la segunda serie está cerrada sólo cuando I0 está abierto y I1 está cerrado.

Realizando el paralelo de las dos se obtiene la función deseada. Es decir, la bobina se activa sólo cuando una entrada está cerrada y la otra está abierta.

Ejemplo 6: autorretención

Un pulsador conectado a la entrada **I0** debe activar la salida **O1.7** y un segundo pulsador conectado a la entrada **I1** debe desactivarla.

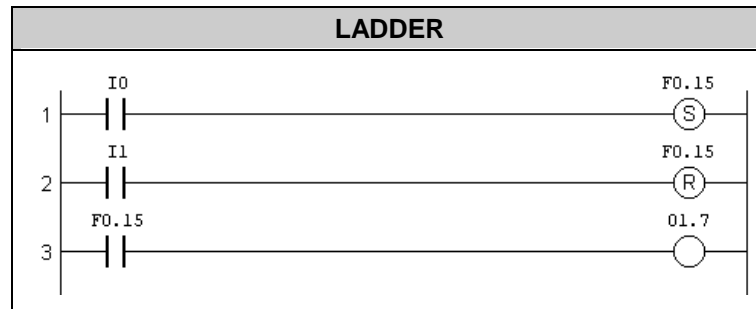


En el programa ladder propuesto se realiza un circuito con autorretención. Accionando el pulsador conectado a **I0**, la bobina **O1.7** se activa. A continuación, el contacto con el mismo operando situado en la segunda línea se cierra (imagine que el contacto y la bobina forman parte del mismo relé **O1.7**) y así sigue manteniendo activada la bobina incluso después de la apertura de **I0**. La desactivación del pulsador situado a la entrada **I1** provoca la apertura del contacto normalmente cerrado en el esquema, desactivando así la bobina y cortando la autorretención.

Ejemplo 7: set y reset

Un pulsador conectado a la entrada **I0** debe activar la salida **O1.7** y un segundo pulsador, conectado a la entrada **I1**, debe desactivarla

El ejercicio es idéntico al anterior pero, en esta ocasión, en la solución se usan bobinas de set y reset. La utilización del flag de apoyo **F0.15** es necesaria para impedir las oscilaciones de la salida cuando están cerradas las dos entradas. De hecho, en estas condiciones y debido a la falta de la imagen de las salidas, la conexión directa de las bobinas de set y reset a los contactos **I0** y **I1** conduciría a obtener, para cada ciclo de ejecución, primero, la activación de la salida física **O1.7** y más tarde su desactivación. Utilizando un flag como operando de las bobinas y siempre basándonos en la hipótesis de que ambos pulsadores estén activados, aquel continua oscilando y asume el estado alto y a continuación el bajo, pero el único valor que pasará a la salida, al tercer recorrido, será el último asumido: el valor bajo.

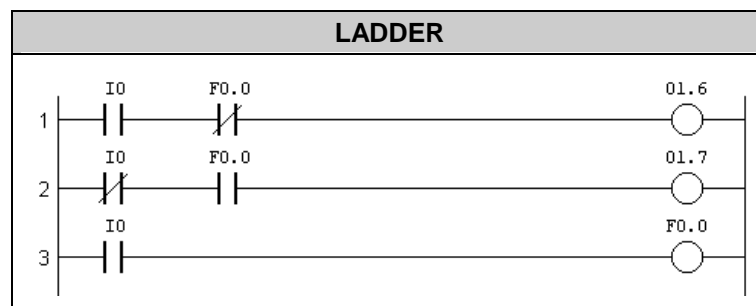


Así pues, con esta escritura de programa hemos logrado que prevalezca el reset sobre el set. Si desea obtener lo contrario, le bastará invertir la posición de los dos primeros recorridos en el esquema de contactos.

Ejemplo 8: activación por flancos

Activar las salidas O1.6 y O1.7 que se corresponden, respectivamente, con los flancos ascendente y descendente de la entrada I0.

Observemos que el último recorrido del esquema ladder impone la igualdad del flag F0.0 con la entrada I0 al término de la ejecución del programa. Pero, en correspondencia con los flancos, así como para los recorridos anteriores, se da el hecho de que el estado de las dos variables es opuesto y que tan sólo al final de la elaboración es igual.



Todo ello queda representado en las dos primeras líneas del diagrama de [Figura 2](#) con un retardo temporal de F0.0 respecto a I0 que equivale a un ciclo de ejecución.



Figura 2: diagrama temporal activador por flancos

La bobina O1.6 está activa para un ciclo de programa cuando I0 está en 1 y F0.0 está en 0, es decir, en correspondencia con el flanco ascendente de I0, tal como se muestra en la tercera línea del diagrama.

Por el contrario, la bobina O1.7 está activa para un ciclo de programa cuando I0 está en 0 y F0.0 está en 1, es decir, en correspondencia con el flanco descendente de I0, tal como se muestra en la cuarta línea del diagrama.

Ejemplo 9: temporizador a impulso

La salida O5 se activa al cerrarse la entrada I0 y se desactiva 5 segundos después. Si la entrada se reabre durante este periodo, la salida se desactiva inmediatamente.

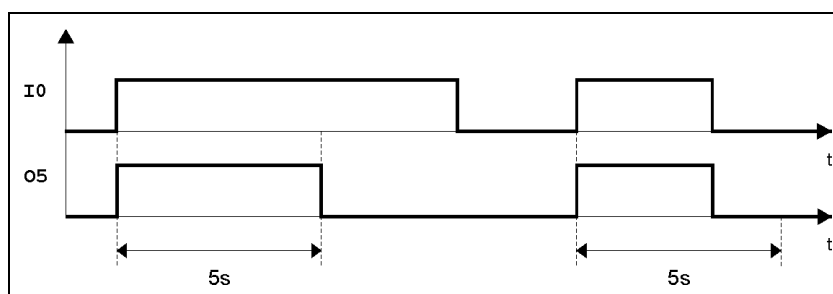


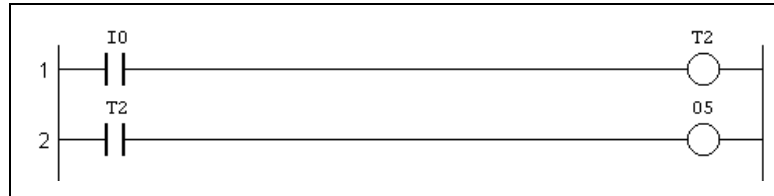
Figura 3: diagrama temporal del temporizador a impulso

El funcionamiento del temporizador a impulso queda ilustrado en el diagrama temporal de [Figura 3](#). En la primera línea se representa la señal de entrada y en la segunda, la de salida.

La bobina directa, utilizada para la activación del temporizador, es sensible tanto al flanco ascendente como al descendente de la entrada I0. En el flanco ascendente, el estado de T2 se pone en 1 y se inicia el tiempo. Al término de la temporización, el estado vuelve a 0. Si, durante el transcurso del tiempo, la entrada se abre, es decir, se genera un flanco descendente, el estado del temporizador se lleva a 0 y el tiempo se borra.

El valor de preset del temporizador debe programarse a 500 centésimas de segundo en la [Ventana Asignación de preseletores \(FPC 404\)](#).

LADDER



Ejemplo 10: temporizador a impulso prolongado

La salida O5 se activa al cierre de la entrada I0 y se desactiva 5 segundos después, independientemente del hecho de que la entrada se reabra o no durante este periodo.

El funcionamiento del temporizador a impulso prolongado queda ilustrado en el diagrama temporal de [Figura 4](#). En la primera línea está representada la señal de entrada, y en la segunda, la correspondiente salida.

A diferencia de lo realizado en el ejemplo anterior, aquí se utiliza una bobina de set para la puesta en marcha del temporizador. Esta, a diferencia de la bobina directa, tan sólo es sensible al flanco ascendente para la puesta en marcha de un temporizador.

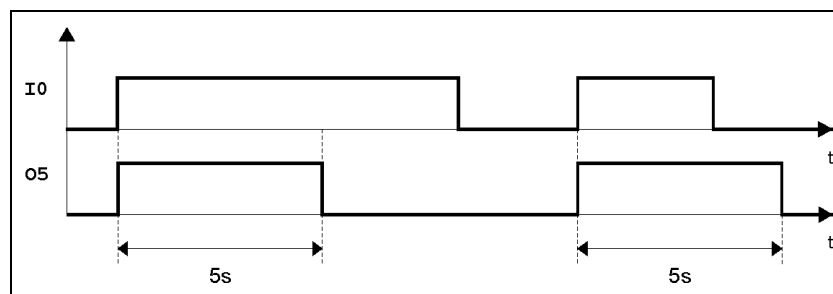
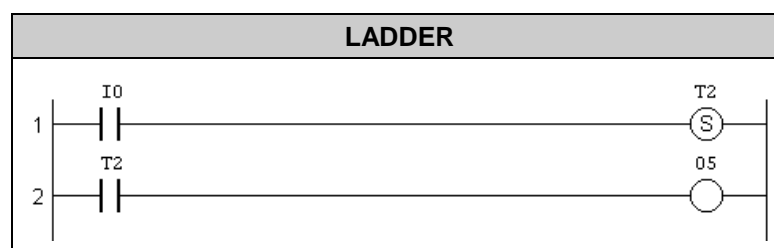


Figura 4: diagrama temporal del temporizador a impulso prolongado

Como antes, en el flanco ascendente de I0, el estado de T2 se pone en 1 y empieza a correr el tiempo. Al término de la temporización, el estado de T2 vuelve a 0. Pero, durante el transcurso del tiempo, una posible apertura de la entrada no se percibe, es decir, el flanco descendente se ignora de modo que no ejerce influencia en la salida del temporizador, que sólo volverá a cero cuando se agote el tiempo.



El valor de preset del temporizador debe programarse a 500 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 11: temporizador con retardo a la activación

La salida 05 se activa 3,5 segundos después del cierre de la entrada I0. Cuando la entrada se reabre la salida se desactiva.

El diagrama temporal de [Figura 5](#) ilustra el funcionamiento de la temporización mostrando, en la primera línea, la entrada y, en la última, la salida. Por su parte, la segunda línea representa el estado del temporizador T2 que, como se deduce del esquema ladder, está conectado para funcionar a impulso (ver el ejemplo anterior).

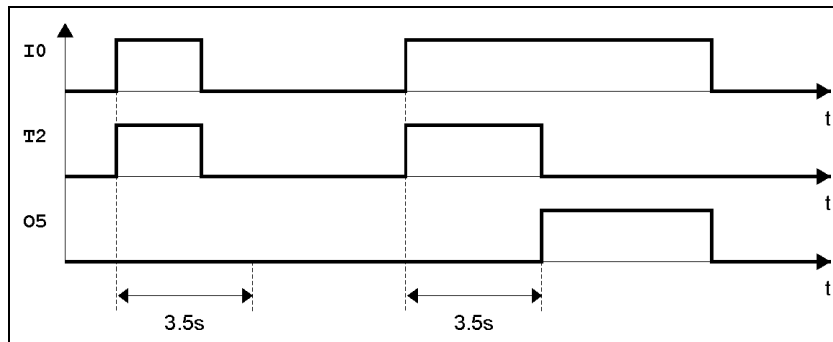
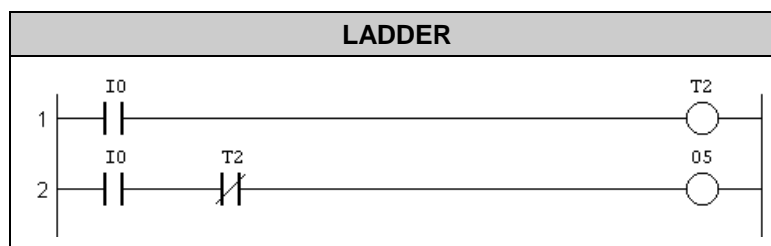


Figura 5: diagrama temporal del temporizador con retardo a la activación

Comparando los recorridos, observamos como 05 debe ser cierto cuando la entrada es cierta y T2 es falso; es decir:

$$O5 = I0 \cdot \overline{T2}$$

Hemos aprendido anteriormente a traducir esta expresión a un diagrama ladder: basta con dibujar un recorrido en el que se pilota la bobina de 05 con dos contactos en serie, uno NA de I0 y uno NC (por medio de la negación) de T2.



El valor de preset del temporizador debe estar programado a 350 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 12: temporizador con retardo a la desactivación

La salida O5 debe activarse a la vez que se cierra la entrada I0 y desactivarse 5 segundos después de su reapertura.

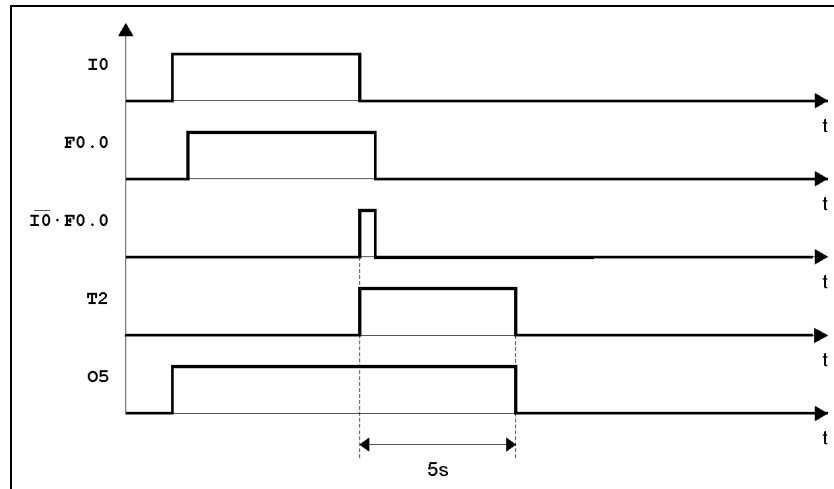


Figura 6: diagrama temporal del temporizador con retardo a la desactivación

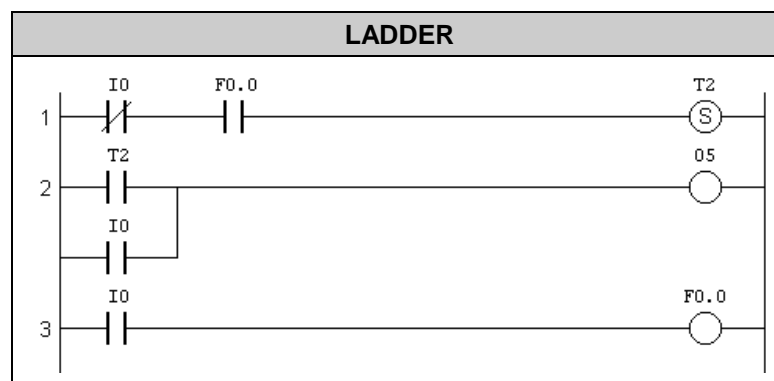
La solución se construye poniendo en marcha un temporizador en correspondencia con el flanco descendente de la señal de entrada. La elevación del flanco está constituida por la serie de contactos del primer recorrido y por el último recorrido al completo, tal como ya hemos aprendido a hacer en uno de los ejemplos anteriores y como se muestra en las tres primeras líneas del diagrama.

La cuarta línea detalla el avance del estado del temporizador a impulso prolongado (ver ejemplo correspondiente) y como se ha obtenido iniciando el temporizador en correspondencia con el flanco descendente de I0.

Observamos que la salida O5 debe ser cierta cuando es cierta la entrada o bien es cierta T2, es decir, en términos de expresión booleana:

$$O5 = I0 + T2$$

Lo que, en términos de diagrama ladder, equivale a pilotar la bobina de O5 con el paralelo de dos contactos NA con operandos I0 y T2.



El valor de preset del temporizador debe estar programado a 500 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 13: temporizador con retardo a la activación con memoria

La salida O5 se activa 5 segundos después del cierre de la entrada I0 (aunque si mientras tanto, éste último vuelve a abrirse) y se desactiva en correspondencia con el cierre de la entrada I1.

El funcionamiento del temporizador con retardo a la activación con memoria se obtiene de la comparación de las dos primeras líneas y la última del diagrama temporal de [Figura 7](#). Los primeros representan las señales de entrada y el último, la salida correspondiente a estos.

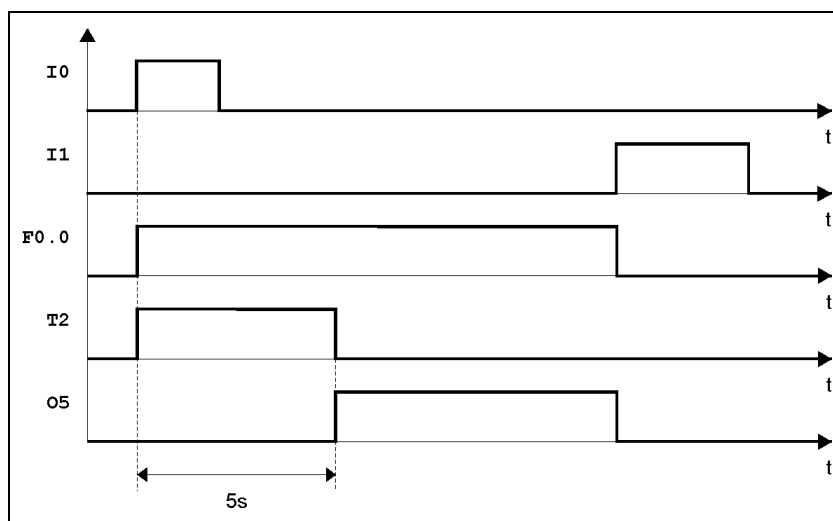


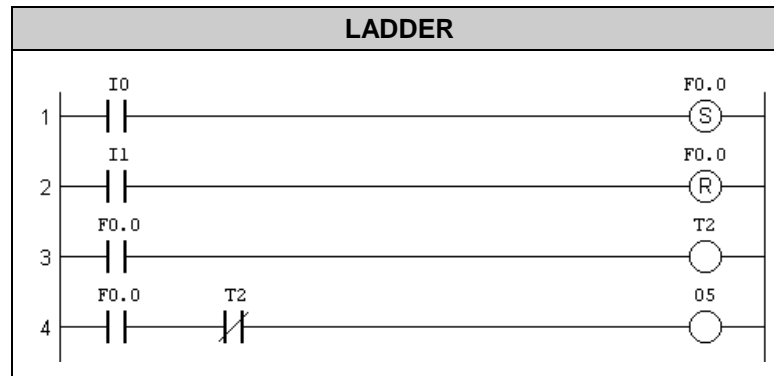
Figura 7: diagrama temporal para un temporizador con retardo a la activación con memoria y reset

La tercera línea del diagrama temporal muestra el avance del flag F0.0 tal como se ha obtenido en la elaboración de los dos primeros recorridos del esquema ladder (en los ejemplos anteriores, ver set y reset). Este flag inicia, al tercer recorrido, un temporizador a impulso, cuyo avance temporal se muestra en la cuarta línea.

De nuevo, al igual que en los ejemplos precedentes, comparamos los recorridos para obtener la expresión lógica que describe el comportamiento de O5 en los términos de las variables de las líneas anteriores.

$$O5 = F0.0 \cdot \overline{T2}$$

La expresión ha sido implementada para el cuarto recorrido del esquema de contactos.



El valor de preset del temporizador debe programarse a 500 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 14: temporizador con retardo a la activación y a la desactivación

La salida *o1.3* se activa 2 segundos después del cierre de la entrada *i1* y se desactiva 7 segundos después de su reapertura.

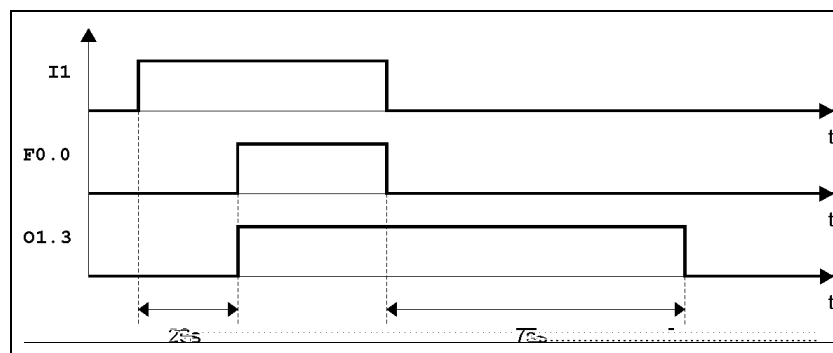
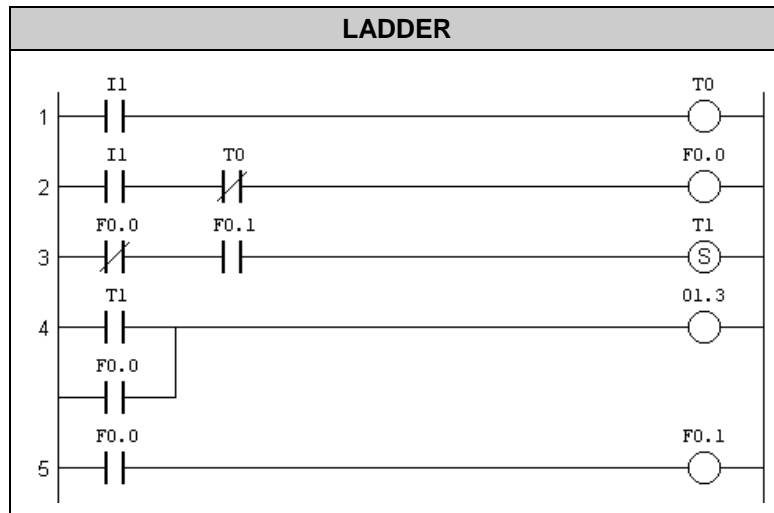


Figura 8: diagrama temporal del temporizador con retardo a la activación y a la desactivación

La [Figura 8](#) ilustra, en la primera y la última línea del diagrama, el avance de las dos señales del trazado. En la línea intermedia se ha dibujado el avance de un flag que representa el estado de un temporizador con retardo a la activación, que se ha activado precisamente por *I1*. Observemos como el avance de la salida *O5*, respecto al avance del flag, representa un retardo a la desactivación.

Y ahora, ¡hagan juego, señores! Se trata de realizar un programa que contenga dos temporizadores: el primero, con retardo a la activación de 2 seg., activado por *I1*, el segundo, retardo a la desactivación de 7 seg, activado por el primer.

Observando la solución propuesta, vemos que los dos primeros recorridos implementan un temporizador con retardo a la activación con entrada *I1* y salida *F0.0*. Por su parte, los otros tres recorridos implementan un temporizador con retardo a la desactivación que tiene *F0.0* como entrada y *O1.3* como salida. El flag *F0.1* es necesario para la elevación del flanco descendente de *F0.0*. ¡El ejercicio está resuelto!



El valor de preset del temporizador $T0$ debe programarse a 500 y el de $T1$ a 700 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 15: impulso retardado

La salida $O7$ se activa 2 segundos después de la apertura de la entrada $I5$ por un período de 1 segundo.

El diagrama en [Figura 9](#) ilustra, en la primera línea, el avance de la entrada y en la última, el de la salida del temporizador requerido. La segunda y la tercera línea representan los avances del estado de dos temporizadores ($T10$ e $T11$) con retardo a la desactivación (de 2 seg y 3 seg.) que ya hemos aprendido a realizar en los ejemplos anteriores. Comparándolos se puede obtener la expresión booleana que describe la salida:

$$O7 = \overline{T10} \cdot T11$$

El primer y el último recorrido del esquema implementan los dos temporizadores con retardo a la desactivación y el segundo implementa la expresión lógica.

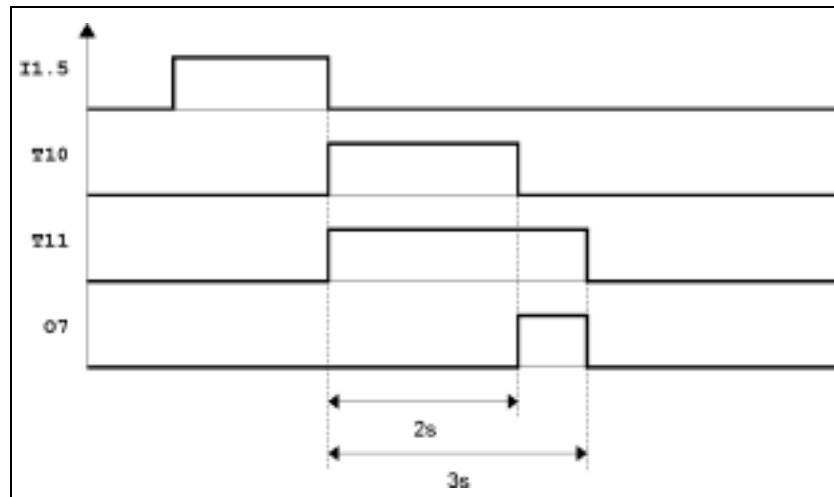
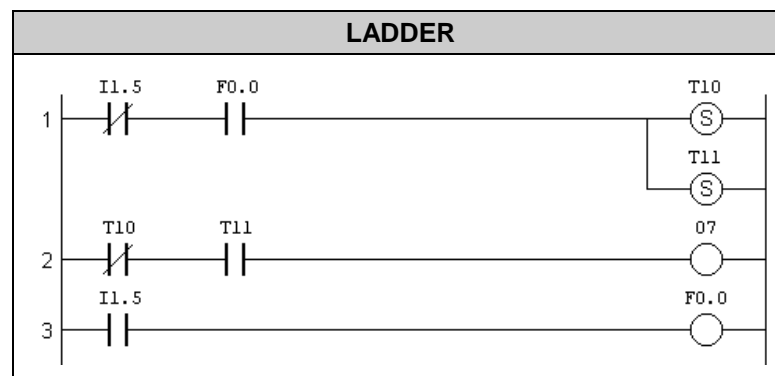


Figura 9: diagrama temporal del impulso retardado



El valor de preset del temporizador $T10$ debe programarse a 200 y el de $T1$ a 300 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 16: tren de impulsos

La salida $o4$ debe activarse un instante a cada segundo.

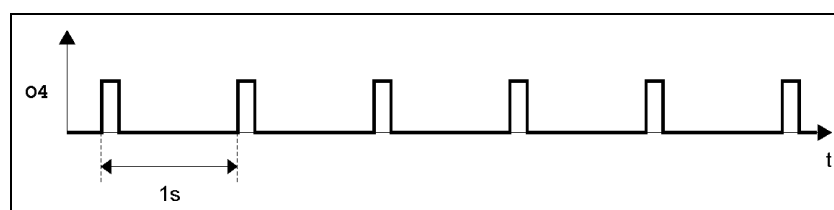
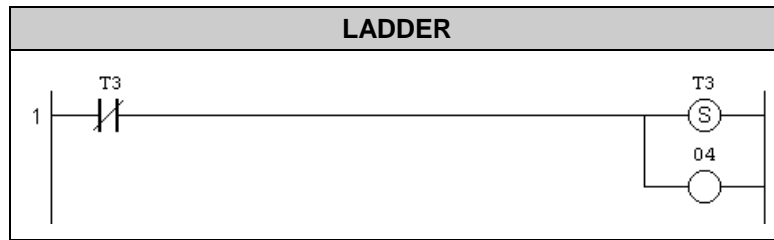


Figura 10: diagrama temporal del tren de impulsos

El ejercicio se resuelve utilizando un temporizador ($T3$) que se pone en marcha cíclicamente. Cada vez que el tiempo llega a cero, el estado del temporizador pasa a bajo. A continuación, el contacto NC se cierra y $T3$ vuelve a ponerse en marcha. Su estado vuelve a ser alto y el contacto se reabre, y así sucesivamente.

En paralelo a la bobina de $T3$ se ha insertado una de $o4$ para controlar la salida, tal como requiere el trazado.



El valor de preset del temporizador debe programarse a 100 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

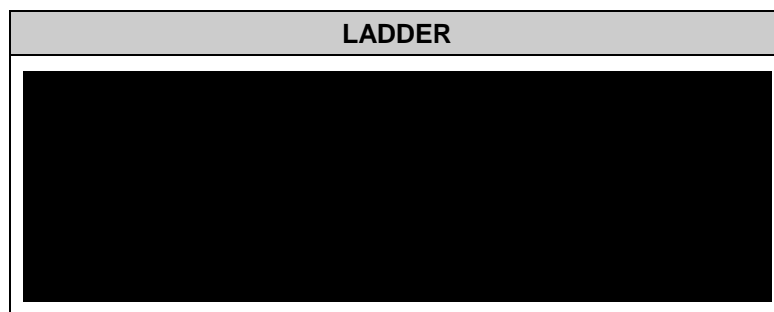
Ejemplo 17: conteo hacia atrás

El contador C4 se pone enl valor 10 en correspondencia con la activación de la entrada I1.0 y va decreciendo a cada cierre de la entrada I1.1. La salida O1.7 se activa en correspondencia con el valor 0 del conteo.

En el primer recorrido del programa ladder, el contador C4 se utiliza como operando de una bobina de preset. Su valor de conteo se pone enl valor de preset cuando se cierra la entrada I1.0. En correspondencia con ello, su estado se hace cierto.

En el segundo recorrido, la bobina de decremento se encarga de disminuir en 1 el valor de conteo de C4 a cada activación de la entrada I1.1. Al llegar a cero, el estado del contador se convierte en falso.

En el último recorrido, la salida O1.7 se dirige con el complemento del estado del contador.



El valor de preset del contador debe programarse a 10 en la [Ventana Asignación de preselectores \(FPC 404\)](#).

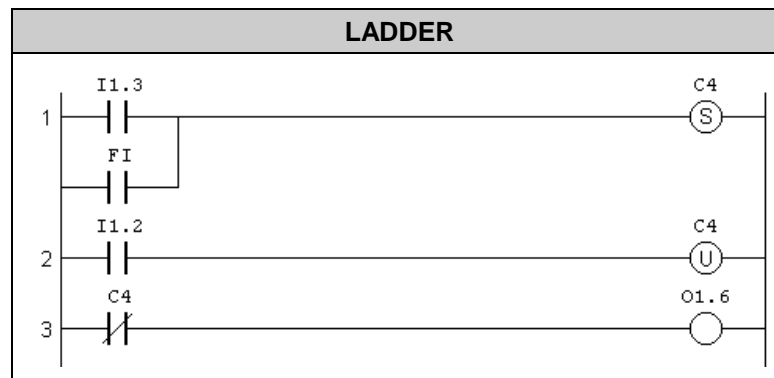
Ejemplo 18: conteo hacia adelante

El contador **C4** se incrementa a cada cierre de la entrada **I1.2** y vuelve a ponerse a cero en correspondencia con el estado alto de la entrada **I1.3**. La salida **O1.6** se activa cuando el valor de conteo llega a 10.

En el primer recorrido del programa ladder, el contador **C4** se utiliza como operando de una bobina de set. Su valor de conteo se pone a cero al cierre de la entrada **I1.3**. En correspondencia con ello, su estado se hace cierto.

En el segundo recorrido, la bobina de incremento se encarga de aumentar en 1 el valor de conteo de **C4** a cada activación de la entrada **I1.2**. Cuando se alcanza el preset, el estado del contador se hace falso.

En el último recorrido, la salida **O1.6** se controla con el complemento del estado del contador.



El valor de preset del contador debe programarse a 10 en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 19: conteo del tiempo de cierre de una entrada (en segundos)

Determinar durante cuantos segundos se mantiene cerrada la entrada I5 y utilice la entrada I6 para poner a cero el conteo del tiempo.

En primer lugar, es preciso realizar una base de tiempos de un segundo, es decir, un tren de impulsos con un periodo de un segundo. A continuación, será necesario contar cuantos impulsos de la base de tiempos se generan durante el cierre de la entrada, es decir, los instantes en los que la entrada y la base de tiempos son ciertos a la vez.

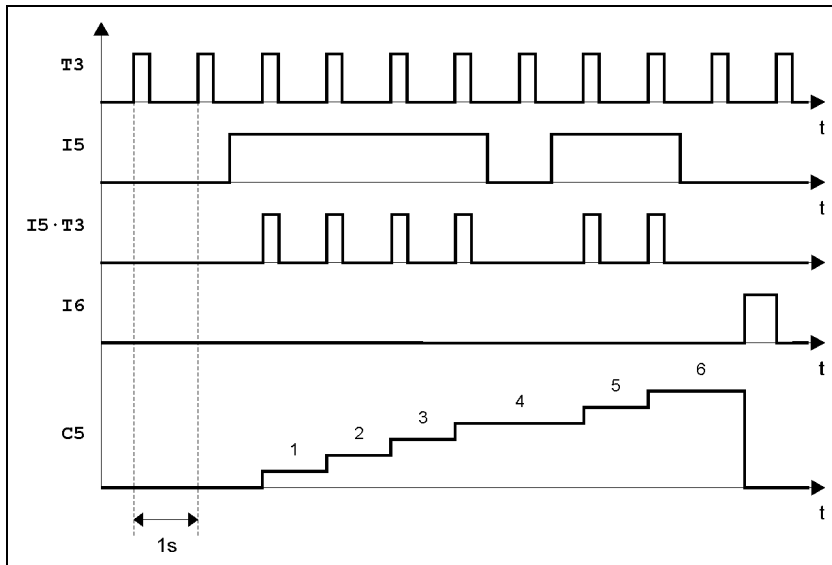
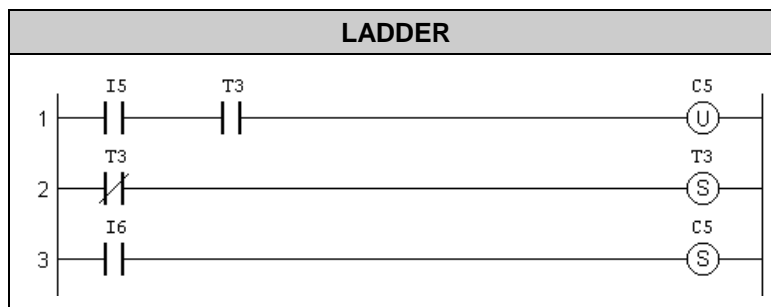


Figura 11: diagrama temporal del contador de tiempo de cierre de la entrada

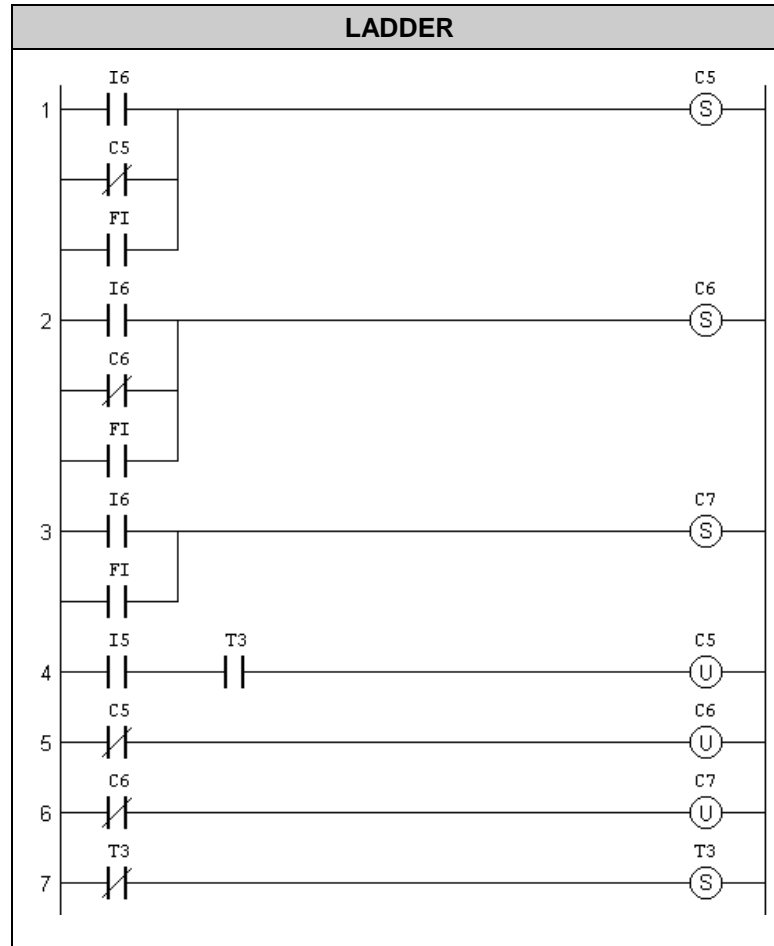
El recorrido 2 genera el tren de impulsos, tal como ya ha quedado claro en el ejemplo 16. El primer recorrido tiene la misión de incrementar el conteo a cada segundo, en correspondencia con el cierre de I5. El último recorrido ejecuta la puesta a cero del conteo, al cierre de I6.



El valor de preset del temporizador debe programarse a 100 centésimas de segundo en la [Ventana Asignación de preselectores \(FPC 404\)](#).

Ejemplo 20: conteo del tiempo de cierre de una entrada (en horas, minutos y segundos)

Determinar cuantas horas, minutos y segundos permanece cerrada la entrada I5 y utilizar la entrada I6 para reponer el conteo del tiempo.



En primer lugar, observamos al final del programa, la construcción de un tren de impulsos con el temporizador **T3**, que actuará como base de tiempos. Para obtener un período de 1 segundo, el valor de preset debe ser igual a 100.

Al recorrido 4 viene incrementado el contador **C5** cuando, durante el cierre de la entrada, se verifica un impulso de base de tiempos, es decir, cuando son ciertos **I5** y **T3** simultáneamente. Por tanto, con la entrada cerrada, **C5** se incrementa a cada segundo y representa así el contador de los segundos.

Cuando el conteo de **C5** alcanza el valor de preset, programado a 60, el estado del contador va bajo. Dos contactos NC de éste, que en ese momento se cierran, están presentes en el esquema en los recorridos 1 y 5. El primero, por medio de la bobina de set, pone **C5** a cero y el segundo incrementa el contador **C6**. Por lo tanto, este último se incrementa en 1 cada vez que **C5** alcanza su preset, es decir cada 60 segundos. Por consiguiente, **C6** representa el contador de los minutos.

También **C6** tiene un valor de preset igual a 60 y, cuando se alcanza, el estado del contador se hace bajo. En el esquema, junto a los recorridos 2 y 6, están presentes dos contactos NC de **C6** que, en esta condición, reponen a cero el valor de conteo de **C6** e incrementan el de **C7**. Este último se incrementa en 1 cada vez que **C6** alcanza el su preset, es decir, cada 60 minutos. Por lo tanto, **C7** representa el contador de las horas.

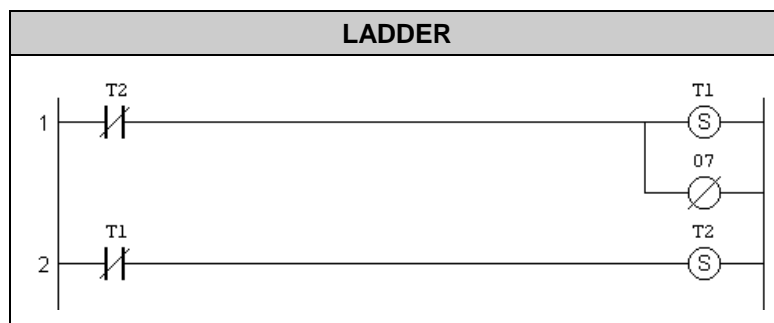
Los contactos NA de **I6**, presentes en los paralelos de los tres primeros recorridos, se encargan de la función de reposición de los tres contadores a continuación del cierre de la entrada, es decir, cuando todo el cuentatiempos se ha puesto a cero. Los contactos con operando **FI** que ejecutan la puesta a cero cuando se pone en marcha el PLC realizan una función idéntica, en el paso Stop-Run.

Como conclusión, los contadores de los segundos y de los minutos se reponen a cero cuando se alcanza el valor 60, cuando se cierra la entrada **I6** y durante el primer ciclo de elaboración. El contador de las horas, en cambio, se repone después de los dos últimos eventos y no tiene un limite de conteo, salvo la longitud del registro interno de 16bit. Así pues, el cuentatiempo de software construido le permite contar hasta 65535 horas, 59 minutos y 59 segundos (¡precisos!). Todo ello, suponiendo que pueda considerarse un valor tan preciso, sobre un tiempo tan largo, teniendo presentes los inevitables errores del reloj interno tanto del PLC real como del PC en el que “actúa” el simulado.

En la [Ventana Asignación de preselectores \(FPC 404\)](#) es necesario programar el valor de preset del temporizador a 100 centésimas de segundo y los de los contadores **C5** y **C6** a 60.

Ejemplo 21: generador de onda cuadrada

La salida **O7** debe estar controlada por una señal de onda cuadrada con $T_{on} = 0.5 \text{ seg}$ e $T_{off} = 1.5 \text{ seg}$.



En la solución propuesta se utilizan dos temporizadores que se “rebotan” mutuamente la activación. Inicialmente, los dos temporizadores están detenidos. La primera elaboración del recorrido 1 del esquema de contactos activa el temporizador **T1** ya que el contacto NC de **T2** está cerrado, puesto que **T2** está inactivo. Pero la elaboración del recorrido siguiente ya encuentra a **T1** iniciado; entonces, su contacto NC está abierto y **T2** no se pone en funcionamiento. No hay ningún cambio hasta que concluye el tiempo de **T1**. En ese momento, el contacto del recorrido 2 se cierra e inicia **T2**, como consecuencia de lo cual el contacto en el recorrido 1 se abre. Agotado **T2**, el contacto del recorrido 1 vuelve a cerrarse y **T1** se inicia de nuevo...y así sucesivamente.

El resultado de este proceso se muestra en el diagrama temporal de [Figura 12](#) donde el avance de **T2** representa justamente la forma de onda requerida por el trazado.

Podremos añadir un recorrido 3 al esquema, en el que un contacto NA de **T2** pilota una bobina de la salida **O7**. Sin embargo, hemos preferido utilizar el contacto de **T2** ya existente, pero, puesto que este último es de tipo NC ha sido necesario invertir también la bobina para obtener un resultado idéntico.

En la [Ventana Asignación de preselectores \(FPC 404\)](#) es preciso programar el valor de preset del temporizador **T1** a 150 y el de **T2** a 50 centésimas de segundo. Modificando adecuadamente dichos valores pueden variarse T_{on} y T_{off} realizando una onda cuadrada con diferentes *duty cycle* y frecuencias.

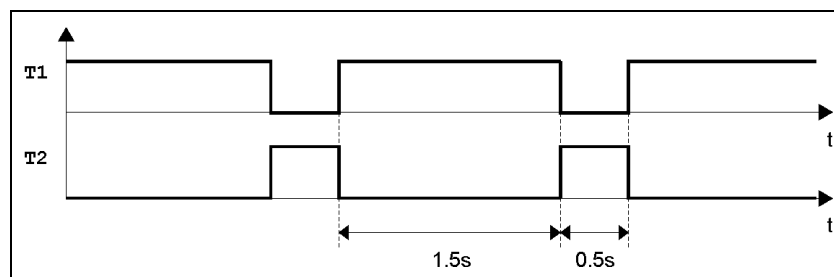


Figura 12: diagrama temporal del generador de onda cuadrada

Ejemplo 22: control temporizado de luces

Un pulsador conectado a la entrada **I0** activa durante tres minutos un grupo de luces conectadas a la salida **O1**. Junto a estas se activa un piloto luminoso conectado a la salida **O2** que, 15 segundos antes de que las luces se apaguen, empieza a parpadear para avisar que el tiempo está a punto de agotarse. El piloto se apaga definitivamente a la vez que las luces.

Para la solución de este ejercicio se han utilizado 4 temporizadores. **T2** y **T3** se emplean como generadores de onda cuadrada con valores de preset de 0.5 seg., de manera que la onda tenga un período de 1 segundo y un *duty cycle* del 50% (ver ejemplo 21). Esta señal se utilizará para el control del parpadeo del piloto luminoso. Por su parte, los temporizadores **T0** y **T1** se ponen en marcha como impulsos prolongados (recorrido 3) por la misma señal de entrada (**I0**). **T0** tiene un valor de preset de 3 minutos y se utilizará para controlar directamente el grupo de luces de la salida **O1** (recorrido 4). Si comparamos la segunda, tercera y cuarta línea del diagrama con el último observamos que la salida **O2** debe estar activada cuando está activado **T0** o cuando, a la vez, está activado **T1**, no activado **T0** y activo **T2**, es decir, en términos de expresión booleana:

$$O2 = T0 + T1 \cdot \overline{T0} \cdot T2$$

Traduciendo esta expresión a términos eléctricos, se obtiene el recorrido 5 del esquema.

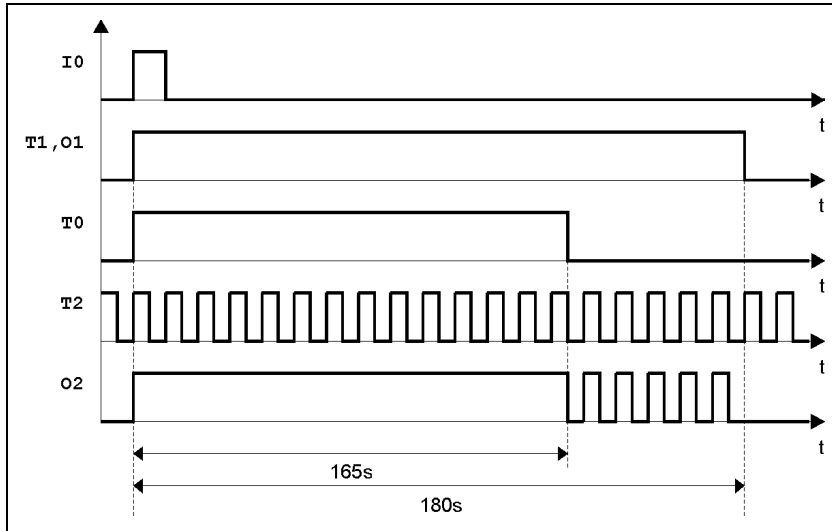
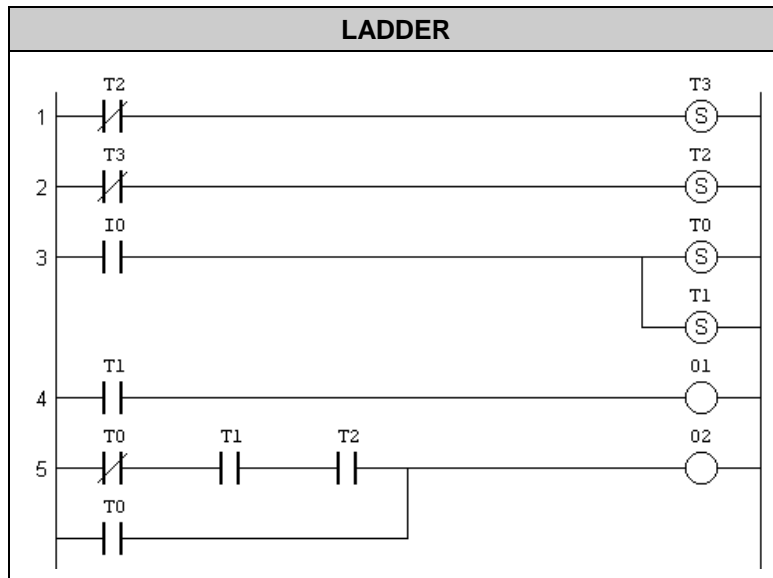


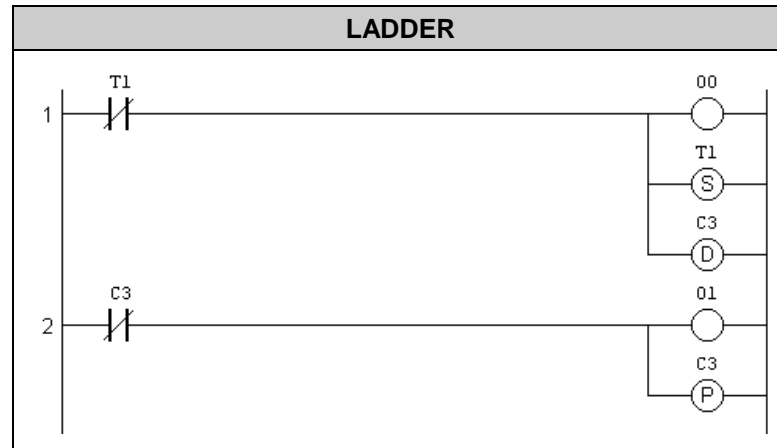
Figura 13: diagrama temporal del control de luces temporizado



En la [Ventana Asignación de preselectores \(FPC 404\)](#) se debe programar el valor de preset de los temporizadores T2 y T3 a 50; el de T0 a 1800 y el de T1 a 1650 centésimas de segundo.

Ejemplo 23: divisor de frecuencia (x4)

Realizar un divisor de frecuencia por 4: cada cuatro impulsos en la entrada se activa un impulso en la salida O1.7.



El primer recorrido realiza simultáneamente tres funciones:

- generar impulsos con un periodo de 1 segundo, que constituirán la señal a dividir (ver ejemplo 16);
- controlar la salida O1.6 con la señal generada;
- disminuir el contador C3.

Por su parte, el segundo recorrido asume la doble función de:

- restablecer el contador al valor de preset (4) cuando el conteo llega a cero;
- transferir a la salida O1.7 la señal dividida.

Basta con cambiar el valor de preset de C3 para dividir por un número cualquiera de veces (hasta 65535).

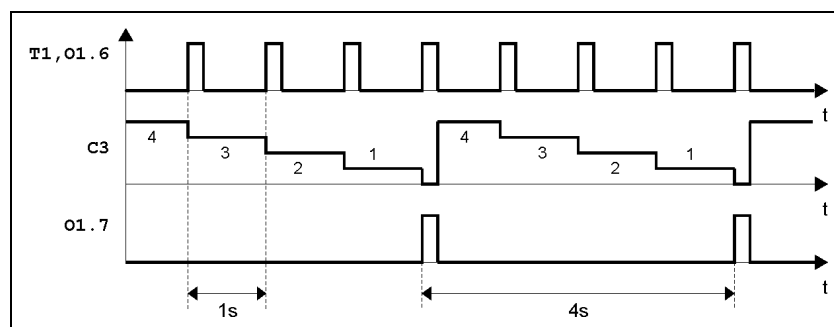


Figura 14: diagrama temporal del divisor de frecuencia por 4

En la [Ventana Asignación de preselectores \(FPC 404\)](#) hay que programar el valor de preset del contador a 4 y el del temporizador a un valor conveniente para que su funcionamiento sea válido. Programando 100, el tren de impulsos tendrá un periodo de 1 segundo y el piloto luminoso de la salida se encenderá a cada 4 segundos.

Ejemplo 24: semáforo para Fórmula 1

Con la activación del pulsador conectado a la entrada I0 las cinco luces de un semáforo deben encenderse una tras otra, una a cada segundo. Un segundo después del encendido completo, las luces deben apagarse.

Para programar la solución a este problema se ha hecho uso de símbolos. Su correspondencia con los operandos absolutos se ha establecido según la tabla siguiente.

Op. absoluto	Símbolo	Comentario
I0	START	Pulsador de puesta en marcha
O1.0	L1	Luz 1
O1.1	L2	Luz 2
O1.2	L3	Luz 3
O1.3	L4	Luz 4
O1.4	L5	Luz 5

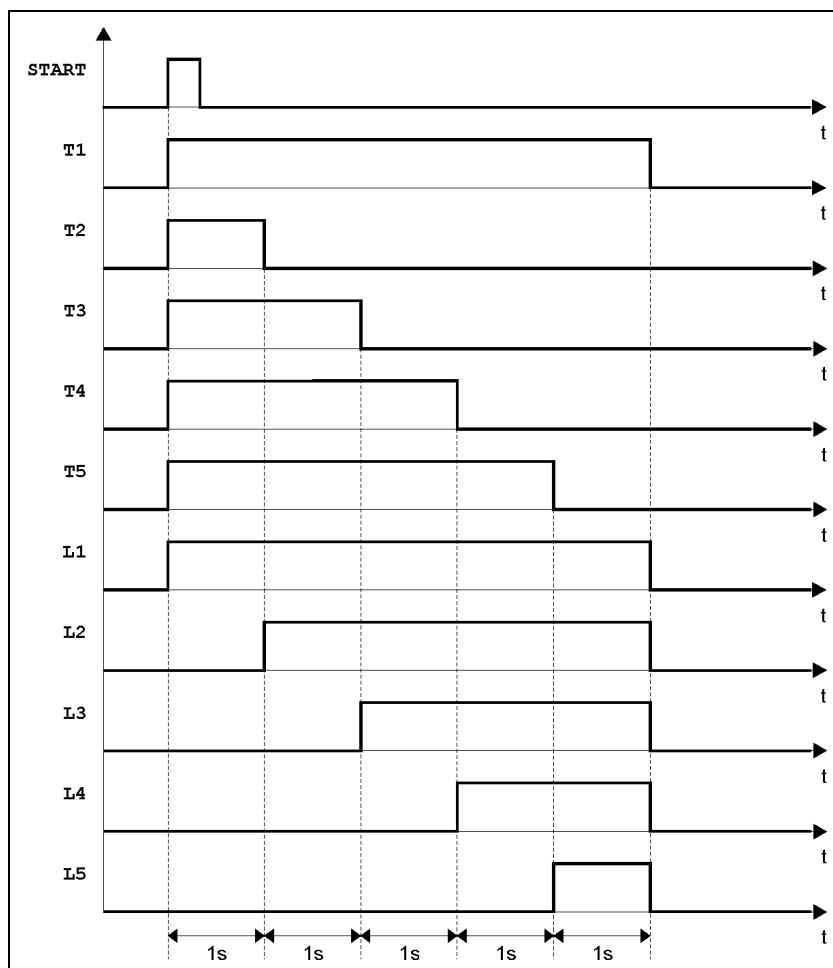


Figura 15: diagrama temporal para semáforo de Fórmula 1

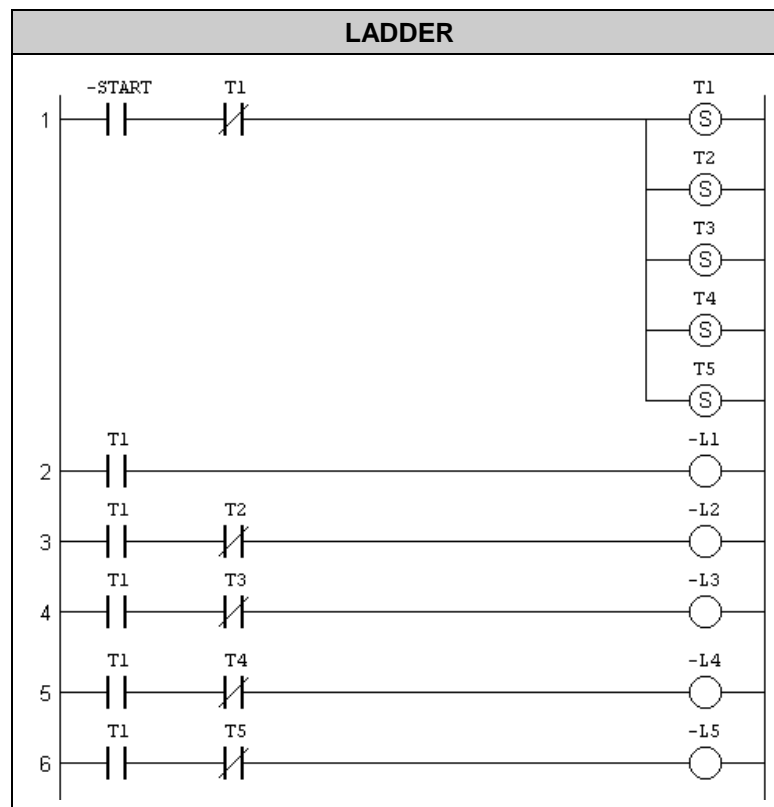
El diagrama temporal de [Figura 15](#) muestra, en las últimas 5 líneas, el desarrollo de las salidas del PLC que controlan las luces del semáforo, en función de la entrada **START** que aparece en la primera línea.

El ejercicio se ha resuelto iniciando cinco temporizadores de impulso prolongado con la entrada **START** (recorrido 1). El contacto NC de **T1** se ha insertado en serie con la activación, para evitar así reinicios en caso de cierres reiterados de **START** durante el ciclo de encendido. El valor de preset de **T1** se ha programado a 5 seg.; los de los temporizadores de **T2** a **T5** van de 1 s a 4 s, con incrementos de 1 s. El desarrollo temporal de éstos se muestra en las líneas 2 a 6 de la misma figura.

Comparando las líneas correspondientes a los temporizadores con las que representan a las salidas para las luces, se observa que:

$$\begin{aligned} L1 &= T1 \\ L2 &= T1 \cdot \overline{T2} \\ L3 &= T1 \cdot \overline{T3} \\ L4 &= T1 \cdot \overline{T4} \\ L5 &= T1 \cdot \overline{T5} \end{aligned}$$

Los recorridos entre el 2 y el 6 del esquema de contactos implementan estas expresiones.



En la [Ventana Asignación de preselectores \(FPC 404\)](#) hay que programar el valor de preset del temporizador **T1** a 500, el de **T2** a 100, el de **T3** a 200, el de **T4** a 300 y el de **T5** a 400 centésimas de segundo.

Ponga en Run el PLC, trasforme el interruptor 0.1 en un pulsador, acciónelo y...¡que gane el mejor!

CAPÍTULO 12

NOTAS SOBRE LA PROGRAMACIÓN DEL PLC PS3

Premisa

Las notas facilitadas en esta sección se refieren únicamente al PLC simulado, es decir, a PC-Sim para PS3. Sin embargo, muy a menudo, y precisamente por el hecho de que en toda la simulación con AW-SYS se procura un acercamiento lo más cercano posible a la realidad, también pueden ser útiles para el PLC real. Le aconsejamos que utilice los manuales originales del PLC y de los dispositivos de programación para obtener todas las informaciones necesarias para la programación del dispositivo real

Algunos de los temas tratados son, por su naturaleza, comunes a todos los PLC simulados. Sin embargo, se ha preferido repetir las mismas descripciones para todos ellos, con el fin de convertir a cada uno de estos capítulos en una unidad autónoma: un cómodo e indispensable manual para tener al alcance de la mano durante la programación del PLC simulado.

Generalidades sobre el PLC PS3

El PS3 es un PLC compacto, y engloba en un único contenedor la CPU, la memoria y los circuitos de entrada/salida (I/O). Está dotado de los siguientes canales de I/O:

- 16 entradas digitales
- 16 salidas digitales
- 4 entradas analógicas
- 1 salida analógica

Cada uno de los canales digitales está provisto de un piloto indicador para la visualización del estado: piloto encendido, para el estado activo; piloto apagado, para el estado inactivo.

Otros dos pilotos, situados en la parte frontal del PLC, están destinados a la señalización de la presencia de alimentación y del estado de Run.

El PLC dispone de un espacio para alojar las baterías que protegen parte de la RAM en momentos de falta de alimentación.

Direccionamiento

Por direccionamiento se entiende la serie de reglas sintácticas que deben observarse al referirse a variables o constantes durante la escritura de un programa de aplicación para PLC. El direccionamiento en la simulación del PLC PS3 se efectúa de acuerdo con las reglas que se describen a continuación.

La sigla que compone cada referencia a un operando la forma una parte constituida por letras, seguida de una parte numérica. La letra inicial especifica la naturaleza del operando: dicha letra será una 'I' para las entradas, una 'Q' para las salidas, una 'M' para los merker (flags), etc.

Para magnitudes de byte, a esta letra le seguirá una B; para magnitudes de word, será una W, mientras que una A servirá para identificar un canal analógico. Para el direccionamiento por bit, no se incluirá ninguna letra.

Por el tanto, la parte formada por letras nos informa (o mejor dicho, informa a la máquina) acerca de la naturaleza y el número de bit del operando. La parte numérica que le sigue identifica, de manera unívoca, a un operando específico entre todos los de la misma naturaleza y número de bits. Así pues, para señalar la merker word 20 deberá escribir **MW20**; 'M' por merker, 'W' por word y el número de la merker word '20'. Dado que la organización de la memoria de este PLC está estructurada por word, la referencia a operandos de 16 bit resulta inmediata: número de la word del operando.

Para direccionar un único bit será necesario especificar la word a la que pertenece y la posición del bit en el interior de ésta, separando con un punto las dos informaciones. Por ejemplo, el bit 10 de la merker word 20 tendrá la dirección **M20.10**.

El direccionamiento de los bytes se efectúa de manera similar. Cada byte se identifica con el número de la word a la que pertenece y el número del bit menos significativo entre los que el componen. Así pues, el byte alto de la merker word 20 será **MB20.8**, mientras que el bajo será **MB20.0**.

Un caso a parte el constituye el direccionamiento de las constantes que, no obstante, sigue unas reglas bastante parecidas. La letra que especifica la naturaleza de un operando como constante es la 'K', que deberá ir seguida de una 'B' para las constantes de byte o de una 'W' para las constantes de word. A esta primera parte le deberá seguir el valor numérico de la constante, con los límites propios de la representación en el número de bits que la componen.

Examinemos ahora más detenidamente el direccionamiento en relación con las distintas tipologías de operandos. Todo el referido aquí se ha resumido en [Tabla 12](#) para mayor comodidad en la consulta.

Entradas

Las dieciséis entradas digitales del PLC corresponden a las direcciones entre **I0.0** y **I0.15**. La word completa formada por estos 16 bit se direcciona con **IW0**. Los dos bytes que la componen tienen direcciones **IB0.0** el menos significativo, y **IB0.8** el más significativo.

Las cuatro entradas analógicas corresponden a las direcciones situadas entre **IA0.0** y **IA0.3**. Cada uno de los canales está representado en un byte; y por tanto puede asumir 256 valores distintos, de 0 a 255.

Al dispositivo real pueden conectarse en red otros tres PLC idénticos. Las entradas y las salidas de estos últimos, denominados expansiones, y que van numerados correlativamente de 1 a 3, se hacen así accesibles al primero, denominado dispositivo base, y que está identificado con el número 0. Los canales de entrada de las expansiones asumen siglas entre **I1.0** y **I3.15**, para las digitales, y de **IA1.0** hasta **IA3.3**, para las analógicas. En todas ellas, la primera cifra de la parte numérica representa el número de expansión, al igual que el 0 en el direccionamiento de los canales internos identifica al dispositivo base. Las word de entrada de las expansiones asumen las direcciones **IW1**, **IW2** y **IW3**. Por su parte, los 6 bytes que las componen se identificarán con siglas que van de **IB1.0** a **IB3.8**.

Al PLC simulado no pueden conectarse expansiones, pero de todos modos, los editor aceptan las correspondientes direcciones y las compila. La verificación de una entrada de un dispositivo de ampliación comportará siempre el resultado 0, salvo en la utilización de PL-RIS o PL-Net.

Salidas

Las dieciséis salidas digitales del PLC corresponden a las direcciones de **Q0.0** a **Q0.15**. La word completa formada por estos 16 bits se direcciona con **QW0**. Los dos bytes que la componen tienen direcciones **QB0.0** el menos significativo y **QB0.8** el más significativo.

El único canal de salida analógica tiene dirección **QA0.0**. Este está representado en un byte; y puede asumir 256 valores distintos, de 0 a 255.

Consideraciones análogas a las referidas respecto a las entradas, en el que se refiere a los dispositivos de ampliación, pueden repetirse para las salidas. En el PLC simulado, una determinada configuración de una de estas salidas no causa ningún efecto, salvo en la utilización de PL-RIS o PL-Net.

Merker

La memoria merker está compuesta por 36 palabras merker (merker word) direccionables con siglas que van de **MW0** a **MW35**.

Cada bit de éstas es direccionable de uno en uno, con siglas operando de **M0.0** hasta **M35.15**. En ellas, la 'M' identifica el operando merker; la primera cifra, el número de la word y la segunda, la posición del bit en el interior de la citada word.

Cada uno de los bytes que componen las merker word puede direccionarse con siglas que van de **MB0.0** a **MB35.8**.

Para mayor claridad, como ejemplo detallamos en el esquema siguiente, el direccionamiento por word, byte y bit para la merker word 2.

MW2															
MB2.8								MB2.0							
M2.15	M2.14	M2.13	M2.12	M2.11	M2.10	M2.9	M2.8	M2.7	M2.6	M2.5	M2.4	M2.3	M2.2	M2.1	M2.0

Los merker **M34.14** y **M34.15** asumen una función especial: estos son dos multivibradores que oscilan con periodos de 100 ms y 2 s respectivamente.

Constantes

En la programación pueden utilizarse dos constantes booleanas (de bit).

K0 valor falso

K1 valor cierto

En el programa también pueden reclamarse constantes de 8 bits, con siglas entre **KB0** y **KB255**, y de 16 bits, con siglas que irán de **KW0** a **KW65535**, en las que la parte numérica identifica precisamente el valor de la constante.

Temporizadores

En el PLC 32 están presentes temporizadores del tipo de retardo a la activación con direcciones que van de **TR0** a **TR31**. El retardo programado oscila entre 0 y 6553,5 segundos.

Contadores

Si dispone de 32 contadores para la programación y sus direcciones van de **C0** a **C31**. El valor de conteo tiene un campo que se sitúa entre 0 y 65535. Además, pueden efectuarse tanto operaciones de conteo hacia delante como hacia atrás.

Registros de desplazamiento

Están presentes 32 registros de desplazamiento (*shift register*) cuyas direcciones van de **SR0** a **SR31**. El número de bits de cada registro es 16 y pueden efectuarse desplazamientos tanto a la izquierda como a la derecha.

Comparadores

Por último, se dispone de 32 comparadores cuyas direcciones van de **CP0** a **CP31**.

Tabla 12: relación de operandos para PLC PS3 (Dispositivo base)

Tipo	Q.tà	Min.	Max
Entradas de bit	16	I0.0	I0.15
Entradas de byte	2	IB0.0	IB0.8
Entradas de word	1	IW0	
Entradas analógicas	4	IA0.0	IA0.3
Salidas de bit	16	Q0.0	Q0.15
Salidas de byte	2	QB0.0	QB0.8
Salidas de word	1	QW0	
Salidas analógicas	1	QA0.0	
Merker bit (retentivos)	64	M0.0	M3.15
Merker bit (no retentivos)	512	M4.0	M35.15
Merker byte (retentivos)	8	MB0.0	MB3.8
Merker byte (no retentivos)	64	MB4.0	MB35.8
Merker word (retentivas)	4	MW0	MW3
Merker word (no retentivas)	32	MW4	MW35
Temporizadores	32	TR0	TR31
Contadores	32	C0	C31
Registros de shift	32	SR0	SR31
Comparadores	32	CP0	CP31
Constantes de bit	2	K0	K1
Constantes de byte	256	KB0	KB255
Constante de word	65536	KW0	KW65535

Elaboración del programa del usuario

Sistema operativo

El corazón del funcionamiento del PLC es su Sistema Operativo: un programa que gestiona el funcionamiento de base del dispositivo, grabado en ROM por la casa constructora.

Un examen minucioso de las funciones del Sistema Operativo no forma parte de los objetivos de este manual. También en los manuales de los PLC reales es difícil encontrar aclaraciones detalladas al respecto. Por otra parte, la ejecución de este programa es completamente transparente para el programador que, en algunos aspectos, puede incluso ignorar su existencia.

Sin embargo, para completar la información, relacionamos algunas de las funciones que el sistema desarrolla, aquellas que consideramos más importantes para los fines recogidos en los siguientes apartados:

- control del tiempo de ciclo (*watch-dog*)
- test del hardware y de la memoria
- llamada al programa del usuario
- actualización de la imagen de las entradas
- transferencia de la imagen de las salidas
- gestión de la toma de programación
- control de la batería tampón.

Estado de la CPU

La CPU puede encontrarse en uno de los siguientes estados:

OFF: el PLC no recibe alimentación, el conmutador de encendido en la [Barra de instrumentos](#) de PC-Sim se encuentra en la posición de Off. Los pilotos luminosos de Power y Run situados en el frontal del PLC están apagados.

STOP: el PLC recibe alimentación pero el programa del usuario no se ejecuta, el conmutador de encendido en la [Barra de instrumentos](#) de PC-Sim se encuentra en la posición de On y el conmutador de funcionamiento se encuentra en la posición de Stop. El piloto luminoso de Power está encendido y el piloto luminoso Run está apagado.

RUN: El PLC recibe alimentación y el programa del usuario se ejecuta; el conmutador de encendido en la [Barra de instrumentos](#) de PC-Sim se encuentra en la posición de On y el conmutador de funcionamiento se encuentra en la posición de Run. Los pilotos de Power y Run situados en el frontal del PLC están encendidos.

El estado vigente de la CPU se visualiza en la barra del título de la [Ventana PLC](#), encerrado entre corchetes.

Imágenes del proceso

Una parte de la RAM del PLC está destinada a servir de memoria intermedia para el acceso a los periféricos. Esta memoria toma el nombre de imagen de proceso y se divide en dos partes: imagen de proceso de las entradas y imagen de proceso de las salidas (a partir de ahora, se denominarán sólo imagen de las entradas y imagen de las salidas).

Antes de la elaboración del programa del usuario, todas las entradas del PLC se leen y su estado se copia en la imagen de las entradas. Al final de la elaboración del programa del usuario, la memoria imagen de las salidas se recopia en las salidas del PLC.

Durante la elaboración del programa, cada acceso a las entradas o a las salidas no está referido al canal físico efectivo, sino a un bit, el correspondiente a aquel canal, en la memoria imagen de proceso. Cuando se requiere la lectura de una entrada, lo que en realidad se lee es un bit de memoria que, con anterioridad ha sido copiado por la entrada efectiva. Cuando se escribe sobre una salida, en realidad no se escribe directamente sobre el canal de salida, sino que se escribe un bit de memoria que a continuación se recopiará sobre el canal físico de salida.

Por el tanto, el microprocesador situado en el interior del PLC deberá dividir su tiempo entre diferentes funciones: ejecutar el Sistema operativo, actualizar la imagen de las entradas, ejecutar el programa de la aplicación y transferir la imagen de las salidas. El diagrama circular de la [Figura 16](#) ilustra estas actividades. El sentido del recorrido del diagrama es el indicado por la flecha. Los porcentajes respectivos de tiempo son absolutamente indicativos y, en particular, el dedicado a la ejecución del programa del usuario varía con las dimensiones del mismo, es decir, en general, cuanto más largo es el programa más tiempo necesitará la CPU para ejecutarlo (excluyendo de esta consideración, los programas que contienen operaciones de salto y ramificación).

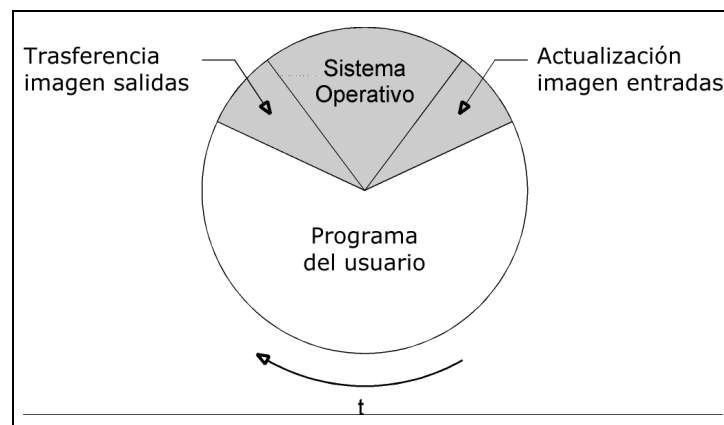


Figura 16: reparto del tiempo del microprocesador del PLC

Control del tiempo del ciclo

El sistema operativo, justo antes de efectuar la llamada al programa del usuario, inicia un monoestable que, por lo general, tiene una duración que va de unas decenas a poquísimas centésimas de milisegundo. Supongamos que el programa del usuario requiera, para su ejecución, un tiempo inferior al del monoestable. En esta situación, antes de cada una de la serie de llamadas, el monoestable se iniciará nuevamente y, como resultado, la salida de este último se mantendría continuamente alta.

En cambio, si el programa del usuario requiriera, para su ejecución más tiempo del que dura el monoestable, la salida de este se situaría en el estado bajo, provocando el paso inmediato del PLC a Stop.

Un programa mal realizado podría llevar a la CPU a ejecutar indefinidamente, o durante un tiempo excesivo para los fines de rapidez en el control de la instalación, un ciclo cerrado de instrucciones sin volver a la elaboración del sistema operativo. Esta condición podría provocar daños a la instalación y un grave peligro para el personal que está trabajando. El control del tiempo del ciclo, que en la literatura técnica especializada podrá encontrar también bajo el nombre de *watch-dog*, evita precisamente este problema.

PC-Sim también controla que su programa no genere un ciclo cerrado pero, en este caso, las múltiples y cambiantes tareas de un PC, imponen tiempos más largos para la detección del problema, del orden de algunos segundos.

Registros

El PLC PS3 utiliza los registros relacionados a continuación para memorizaciones temporales durante la ejecución del programa del usuario.

Registro general (RA)

Se utiliza como memoria de trabajo en la elaboración del programa del usuario. En estos se depositan los resultados de las operaciones de carga o de las operaciones lógicas y de él se obtienen los valores para las operaciones de asignación.

Este registro tiene una longitud variable: será de 16 bits en secuencias de instrucciones que trabajan sobre word, de 8 bits en secuencias de instrucciones que trabajan sobre bytes y, por último, será de 1 bit en secuencias de instrucciones binarias. Ver más adelante la descripción de una secuencia.

Registro auxiliar (RH)

Es un registro de 8 bit que se utiliza en las operaciones de multiplicación y división.

Registro de estado

El registro de estado es un registro de 2 bits. El primero, Bit de overflow (C), se lleva al valor 1 cuando una operación aritmética genera un overflow. El segundo, Bit de cero (Z), se pone en 1 cuando, tras una operación aritmética, el registro general es nulo. Como veremos a continuación, las operaciones de ramificación comprueban el estado de estos dos bits para valorar si es preciso realizar el salto a la etiqueta especificada.

Registro de stack

El PLC posee un registro de stack de 8 bits con estructura LIFO (Last In First Out) en el cual se memorizan los resultados intermedios para la implementación de funciones lógicas que necesitan paréntesis (ver los detalles en las descripciones de las operaciones lógicas y en los ejemplos de programación).

Programación en AWL

La modalidad de programación AWL, o Lista de instrucciones, constituye una de las dos metodologías de implementación de programas que puede utilizar este PLC. Esta modalidad consiste en la escritura de instrucciones consecutivas; cada una de ellas es la unidad más pequeña de programa y constituye una función básica para la CPU.

Instrucciones

Cada instrucción está compuesta, tal como se ilustra en la [Figura 17](#), por una parte operación y, en general, por una parte operando. De esta última ya hemos hablado en el apartado dedicado al direccionamiento. La primera parte, a su vez, está compuesta por una operación y un eventual complemento. Las operaciones que este PLC soporta son de carga (L), de asignación (=, R, S), aritméticas (ADD, SUB, MUL, DIV), de salto y ramificación (JP, JC, JCN, BC, BNC, BZ, BNZ), sobre registros (NOT, GOR).

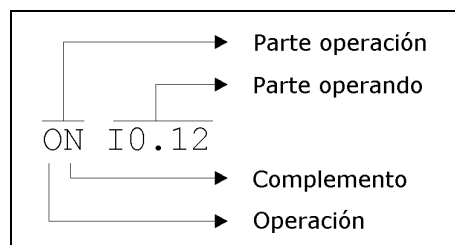


Figura 17: composición de una instrucción AWL

Más adelante, en este mismo apartado, se ofrece una descripción detallada de las operaciones, presentadas por orden alfabético. Para hacer más cómoda la consulta al programador, en [Tabla 13](#) se facilita un resumen con las operaciones agrupadas según el tipo.

Secuencia

Una secuencia es una sucesión de varias instrucciones de programa AWL (al menos dos) que deben cumplir las siguientes reglas:

- La primera instrucción de una secuencia debe ser una instrucción de carga.
- El tipo de operando de esta primera instrucción (bit, byte o word) determina el tipo de datos de toda la secuencia.
- El tipo de datos no puede cambiarse en el interior de una secuencia. Es decir, cada instrucción posterior a la primera, debe hacer referencia a un operando del mismo número de bits.
- Una secuencia puede acabar con una o más operaciones de asignación, set/reset o salto condicionado.

Sin embargo, la instrucción **GOR** es una excepción a esta última norma. El efecto de cierre de secuencia de una instrucción de asignación con byte quedará anulado si a continuación hay una instrucción **GOR** ya que reabre la secuencia de byte.

Ejemplo de una secuencia binaria:

L I0.2 Apertura de una secuencia binaria con instrucción de carga de bit
A I0.3 Operación lógica binaria
ON I0.5 Operación lógica binaria
= Q0.14 Cierre de la secuencia binaria con instrucción de asignación

Ejemplo de una secuencia de byte:

L MB20.8 Apertura de una secuencia de byte con instrucción de carga de byte
DIV KB10 Operación aritmética de byte
= MB21.0 Cierre de la secuencia de byte con instrucción de asignación
GOR Reapertura de la secuencia de byte
= MB21.8 Cierre definitivo de la secuencia de byte con instrucción de asignación

Descripción de las operaciones

En orden alfabético, se describen aquí las operaciones AWL que pueden ser programadas en este PLC y que se resumirán a continuación, subdividas por tipo, en [Tabla 13](#).

= (Asignación)

El contenido del registro general se copia en el operando indicado.

Solo para operaciones de bit, si está presente el operador de complemento 'N', se copia el complemento del registro general.

El registro general y el registro auxiliar no sufren modificaciones.

La tabla siguiente muestra los tipos de operandos admitidos.

Operación	Tipo datos	Operandos admitidos
=	Bit	Q0.0 – Q0.15
		M0.0 – M35.15
	Byte	QB0.0 – QB0.8
		MB0.0 – MB34.8
		QA0.0 – QA0.3
	Word	QW0
MW0 – MW35		
=N	Bit	Q0.0 – Q0.15
		M0.0 – M35.15

A (And)

Si se ha especificado un operando, ejecuta la operación lógica AND entre el operando, eventualmente negado, y el registro general. El resultado se deposita en el registro general. El operando no se modifica. Para operación AND sobre byte se efectúa el AND bit a bit, de los dos términos.

Si no se especifica ningún operando, la operación AND se efectúa entre el último valor binario insertado en el registro de stack, eventualmente negado, y el contenido del registro general. El resultado se deposita también en el registro general.

Operación	Tipo datos	Operandos admitidos
A	Bit	I0.0 - I0.15
		Q0.0 - Q0.15
		M0.0 - M35.15
		K0 - K1
	Byte	IB0.0 - IB0.8
		IA0.0 - IA0.3
		QB0.0 - QB0.8
		QA0.0
		MB0.0 - MB34.8
		KB0 - KB255
AN	Bit	I0.0 - I0.15
		Q0.0 - Q0.15
		M0.0 - M35.15
		K0 - K1

Registro de estado	Actualización	
Bit de overflow (C)	no se modifica	
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

ADD (Adición)

Ejecuta la adición entre el operando especificado y el registro general. La operación tan sólo puede utilizarse en secuencias de byte. El resultado se deposita en el registro general. El operando no sufre modificaciones.

La operación de adición no provoca modificaciones en el registro auxiliar.

Operación	Tipo datos	Operandos admitidos
ADD	Byte	IB0.0 - IB0.8
		IA0.0 - IA0.3
		QB0.0 - QB0.8
		QA0.0
		MB0.0 - MB35.8

Operación	Tipo datos	Operandos admitidos
		KB0 - KB255

Registro de estado	Actualización	
Bit de overflow (C)	1	Si se ha tenido un overflow, es decir, si el resultado es mayor de 255
	0	Si no se ha tenido overflow, es decir, si el resultado es menor o igual de 255
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

B... (Ramificaciones condicionadas)

El contenido del registro de estado se compara con la condición de ramificación. En caso de concordancia, la instrucción especificada en el operando continúa el programa.

Operación	Condición de ramificación	Operandos admitidos
BC	C = 1	Número de línea de una instrucción
BNC	C = 0	Número de línea de una instrucción
BZ	Z = 1	Número de línea de una instrucción
BNZ	Z = 0	Número de línea de una instrucción

Las ramificaciones condicionadas tan sólo están admitidas en secuencias de byte y solo tienen sentido después de operaciones que influyen a los bits del registro de estado.

Los registros no sufren la influencia de las operaciones de ramificación.

DIV (División)

Ejecuta la división del contenido del registro general para el operando especificado. El resultado se deposita en el registro general y el resto en el registro auxiliar. Los valores se interpretan como números enteros positivos representados por un 1 byte (de 0 a 255). La operación tan sólo puede ser programada en secuencias de byte. El operando no se modifica.

El contenido del registro auxiliar podrá elaborarse a continuación con una instrucción GOR.

La operación de división por 0 no se ejecuta.

Operación	Tipo datos	Operandos admitidos
DIV	Byte	IB0.0 - IB0.8
		IA0.0 - IA0.3
		QB0.0 - QB0.8
		QA0.0
		MB0.0 - MB35.8
		KB0 - KB255

Registro de estado	Actualización	
Bit de overflow	no se modifica	
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

GOR (Carga del registro auxiliar)

Copia el contenido del registro auxiliar en el registro general.

La operación tan sólo está admitida en secuencias de byte y su aplicación asume significado solo después de operaciones que influyen sobre el registro auxiliar (multiplicación y división).

El contenido de los registros auxiliares y de estado no se modifica.

JC (Salto condicionado)

Si el contenido del registro general de bit está de acuerdo con la operación especificada, la elaboración se retoma a partir de la instrucción cuyo número se ha especificado como operando. Si no es así, la elaboración continúa a partir de la instrucción siguiente.

La operación tan sólo está admitida en secuencias de bit.

El contenido del registro general y del registro auxiliar no se modifica.

Operación	Condición de salto	Operandos admitidos
JC	RA = 1	Número de línea de una instrucción
JCN	RA = 0	Número de línea de una instrucción

Registro de estado	Actualización	
Bit de overflow	no se modifica	
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

JP (Salto absoluto)

La elaboración del programa se continúa incondicionadamente desde la instrucción cuyo número se ha especificado en el operando.

El contenido del registro general y del registro auxiliar no se modifica.

L (Carga)

Carga el valor de operando en el registro general.

Si la operación se encuentra en el interior de una secuencia de bit, es decir, si la secuencia aún no se ha concluido con una de las operaciones previstas, el contenido original del registro general se deposita en el registro de stack.

El contenido del registro auxiliar no se modifica.

Operación	Tipo datos	Operandos admitidos
L	Bit	I0.0 – I0.15
		Q0.0 – Q0.15
		M0.0 – M35.15
		K0 – K1
	Byte	IB0.0 – IB0.8
		IA0.0 – IA0.3
		QB0.0 – QB0.8
		QA0.0
		MB0.0 – MB34.8
		KB0 – KB255
	Word	IW0
		QW0
		MW0 – MW35
KW0 – KW65535		
LN	Bit	I0.0 – I0.15
		Q0.0 – Q0.15
		M0.0 – M35.15
		K0 – K1

MUL (Multiplicación)

Ejecuta la multiplicación entre el contenido del registro general y el operando especificado. Los valores se interpretan como números enteros positivos con una representación de 1 byte (de 0 a 255). El resultado se deposita en el registro general (byte menos significativo) y en el registro auxiliar (byte más significativo). El contenido del registro auxiliar podrá elaborarse utilizando una operación GOR.

El operando no se modifica.

La operación tan solo está admitida en secuencias de byte.

Operación	Tipo datos	Operandos admitidos
MUL	Byte	IB0.0 – IB0.8
		IA0.0 – IA0.3
		QB0.0 – QB0.8
		QA0.0
		MB0.0 – MB35.8

Operación	Tipo datos	Operandos admitidos
		KB0 - KB255

Registro de estado	Actualización	
Bit de overflow (C)	1	Si se ha tenido un overflow, es decir, si el resultado es mayor de 255
	0	Si no se ha tenido overflow, es decir, si el resultado es menor o igual a 255
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

NOT (Negación)

El contenido del registro general resulta negado, es decir, no se ejecuta el complemento de 1.

El contenido del registro auxiliar no se modifica.

La operación tan sólo está admitida en secuencias de byte.

Registro de estado	Actualización	
Bit de overflow	no se modifica	
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

O (Or)

Si se ha especificado un operando, ejecuta la operación lógica OR entre el operando, eventualmente negado, y el registro general. El resultado se deposita en el registro general. El operando no se modifica. Para una operación OR sobre byte se efectúa la OR bit a bit para los dos términos.

Si no se ha especificado ningún operando, la operación OR se efectuará entre el último valor binario insertado en el registro de stack, eventualmente negado, y el contenido del registro general. El resultado se deposita también en el registro general.

Operación	Tipo datos	Operandos admitidos
O	Bit	I0.0 - I0.15
		Q0.0 - Q0.15
		M0.0 - M35.15
		K0 - K1
	Byte	IB0.0 - IB0.8
		IA0.0 - IA0.3
		QB0.0 - QB0.8
		QA0.0
		MB0.0 - MB34.8

Operación	Tipo datos	Operandos admitidos
ON	Bit	KB0 - KB255
		I0.0 - I0.15
		Q0.0 - Q0.15
		M0.0 - M35.15
		K0 - K1

Registro de estado	Actualización	
Bit de overflow	no se modifica	
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

R (Reset)

El bit especificado por el operando se borra si el contenido del registro general de bit es igual a 1. De no ser así, no se ejecuta ninguna modificación del operando.

La operación tan sólo está admitida en secuencias de bit.

El contenido del registro general y del registro auxiliar no se modifica.

Operación	Tipo datos	Operandos admitidos
R	Bit	Q0.0 - Q0.15
		M0.0 - M35.15

S (Set)

El bit especificado por el operando se pone en 1 si el contenido del registro general de bits es igual a 1. Si no es así, no se ejecuta ninguna modificación del operando.

La operación tan sólo está admitida en secuencias de bit.

El contenido del registro general y del registro auxiliar no se modifica.

Operación	Tipo datos	Operandos admitidos
S	Bit	Q0.0 - Q0.15
		M0.0 - M35.15

SUB (Sustracción)

El operando especificado se sustrae del contenido del registro general. El resultado se deposita en el registro general. El operando no se modifica.

La operación de sustracción no modifica el contenido del registro auxiliar.

La operación tan sólo está admitida en secuencias de byte.

Operación	Tipo datos	Operandos admitidos
		IB0.0 - IB0.8

Operación	Tipo datos	Operandos admitidos
		IB0.0 - IB0.8
SUB	Byte	QB0.0 - QB0.8
		QA0.0
		MB0.0 - MB35.8
		KB0 - KB255

Registro de estado	Actualización	
Bit de overflow (C)	1	Si ha sido necesario un préstamo, es decir, si el minuendo era inferior al sustraendo ($Rab < op$)
	0	Si no ha sido necesario un préstamo, es decir, si el minuendo era mayor o igual al sustraendo ($Rab \geq op$)
Bit de cero (Z)	1	Si el contenido del registro general tras la ejecución de la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

XO (Or exclusivo)

Si se ha especificado un operando, ejecuta la operación lógica OR exclusivo (XOR) entre el operando, eventualmente negado, y el registro general. El resultado se deposita en el registro general. El operando no se modifica. Para una operación XOR sobre byte se efectúa el XOR, bit a bit, de los dos términos.

Si no se ha especificado ningún operando, la operación XOR se efectúa entre el último valor binario insertado en el registro de stack, eventualmente negado, y el contenido del registro general. El resultado se deposita también en el registro general.

Operación	Tipo datos	Operandos admitidos
XO	Bit	I0.0 - I0.15
		Q0.0 - Q0.15
		M0.0 - M35.15
		K0 - K1
	Byte	IB0.0 - IB0.8
		IA0.0 - IA0.3
		QB0.0 - QB0.8
		QA0.0
		MB0.0 - MB34.8
		KB0 - KB255
		I0.0 - I0.15
		Q0.0 - Q0.15
		M0.0 - M35.15
		K0 - K1

Registro de estado	Actualización	
Bit de overflow	no se modifica	
Bit de cero (Z)	1	Si el contenido del registro general tras la operación es 0
	0	Si el contenido del registro general tras la operación es distinto de 0

Resumen de las operaciones

La tabla siguiente incluye un resumen de la descripción de las diversas operaciones que hemos visto en las páginas anteriores, ordenadas por tipos. Las abreviaturas que se utilizan en ella tienen los significados siguientes.

Sigla Significado

RA	registro general de bit
RAb	registro general de byte
RAw	registro general de word
RS	registro de stack
RH	registro auxiliar
C	flag de overflow
Z	flag de cero
Op	operando

Tabla 13: resumen de operaciones PLC PS3

Operación	Operando	Descripción	
Operaciones de carga			
L	I, Q, M, K	Carga el estado del operando en el registro general (bit)	RS = RA RA = Op
L	IB, IA, QB, QA, MB, KB	Carga el estado del operando en el registro general (byte)	RAb = Op
L	IW, QW, MW	Carga el estado del operando en el registro general (word)	RAw = Op
LN	I, Q, M, K	Carga el complemento del operando en el registro general (bit)	RS = RA RA = \overline{Op}
Operaciones de asignación			
=	Q, M	Asigna al operando el valor del registro general (bit)	Op = RA
=	QB, QA, MB	Asigna al operando el valor del registro general (byte)	Op = RAb
=	QW, MW	Asigna al operando el valor del registro general (word)	Op = RAw
=N	Q, M	Asigna al operando el valor del complemento del registro general (bit)	Op = \overline{RA}

Operación	Operando	Descripción	
R	Q, M	Asigna al operando el valor 0 si el registro general (bit) es 1	Si RA = 1 entonces Op = 0
S	Q, M	Asigna al operando el valor 1 si el registro general (bit) es 1	Si RA = 1 entonces Op = 1
Operaciones lógicas			
A	I, Q, M, K	Ejecuta la AND entre el registro general (bit) y el operando y pone el resultado en el registro general (bit)	$RA = RA \text{ AND } Op$
A	IB, IA, QB, QA, MB, KB	Ejecuta la AND bit a bit entre el registro general (byte) y el operando y pone el resultado en el registro general (byte)	$RAb = RAb \text{ AND } Op$
A		Ejecuta la AND entre el stack y el registro general (bit) y pone el resultado en el registro general (bit)	$RA = RS \text{ AND } RL$
AN	I, Q, M, K	Ejecuta la AND entre el registro general (bit) y el complemento del operando y pone el resultado en el registro general (bit)	$RA = RA \cdot \overline{Op}$
AN		Ejecuta la AND entre el stack y el complemento del registro general (bit) y pone el resultado en el registro general (bit)	$RA = RS \cdot \overline{RA}$
O	I, Q, M, K	Ejecuta la OR entre el registro general (bit) y el operando y pone el resultado en el registro de trabajo (bit)	$RA = RA + Op$
O	IB, IA, QB, QA, MB, KB	Ejecuta la OR bit a bit entre el registro general (byte) y el operando y pone el resultado en el registro general (byte)	$RAb = RAb + Op$
O		Ejecuta la OR entre el stack y el registro general (bit) y pone el resultado en el registro general (bit)	$RA = RS + RA$
ON	I, Q, M, K	Ejecuta la OR entre el registro general (bit) y el complemento del operando y pone el resultado en el registro general (bit)	$RA = RA + \overline{Op}$
ON		Ejecuta la OR entre el stack y el complemento del registro general (bit) y pone el resultado en el registro general (bit)	$RA = RS + \overline{RA}$
XO	I, Q, M, K	Ejecuta la XOR entre el registro general (bit) y el operando y pone el resultado en el registro general (bit)	$RA = RA \oplus Op$
XO	IB, IA, QB, QA, MB, KB	Ejecuta la XOR bit a bit entre el registro general (byte) y el operando y pone el resultado en el registro general (byte)	$RAb = RAb \oplus Op$
XO		Ejecuta la XOR entre el stack y el registro general (bit) y pone el resultado en el registro general (bit)	$RA = RS \oplus RA$
XON	I, Q, M, K	Ejecuta la XOR entre el registro general (bit) y el complemento del operando y pone el resultado en el registro general (bit)	$RA = RA \oplus \overline{Op}$
XON		Ejecuta la XOR entre el stack y el complemento del registro general (bit) y pone el resultado en el registro general (bit)	$RA = RS \oplus \overline{RA}$

Operación	Operando	Descripción	
Operaciones aritméticas			
ADD	IB, IA, QB, QA, MB, KB	El contenido del registro general (byte) se suma al operando y el resultado se deposita en el registro general (byte)	$RAb = RAb + Op$
DIV	IB, IA, QB, QA, MB, KB	El contenido del registro general (byte) se divide por el operando, el resultado se deposita en el registro general (byte) y el resto en el registro auxiliar	$RAb = RAb / Op$ $RH = Resto(RAb / Op)$
MUL	IB, IA, QB, QA, MB, KB	El contenido del registro general (byte) se multiplica por el operando y el resultado se deposita en el registro general (parte baja) y en el registro auxiliar (parte alta)	$RAb = RAb \times Op (L)$ $RH = RAb \times Op (H)$
SUB	IB, IA, QB, QA, MB, KB	El operando se resta del contenido del registro general y el resultado se deposita en el registro general	$RAb = RAb - Op$
Operaciones de salto y de ramificación			
BC	N. instr.	Salta a la instrucción especificada si flag C = 1	
BNC	N. instr.	Salta a la instrucción especificada si flag C = 0	
BZ	N. instr.	Salta a la instrucción especificada si flag Z = 1	
BNZ	N. instr.	Salta a la instrucción especificada si flag Z = 0	
JC	N. instr.	Salta a la instrucción especificada si el registro general (bit) es 1	
JCN	N. instr.	Salta a la instrucción especificada si el registro general (bit) es 0	
JP	N. instr.	Salto incondicional a la instrucción especificada	
Operaciones sobre registros			
GOR		El contenido del registro auxiliar está cargado en el registro general (byte)	$RAb = RH$
NOT		El contenido del registro general (byte) se niega bit a bit	$RAb = \overline{RAb}$

Programación en Ladder

La escritura de un programa para PLC en Ladder diagram o Esquema de contacto se realiza de manera similar al diseño de un circuito eléctrico con relé. Entre dos líneas verticales, dispuestas en los márgenes del esquema, que representan la alimentación del circuito, pueden colocarse los elementos del circuito eléctrico dispuestos sobre líneas horizontales (que a partir de ahora llamaremos ramales) conectados entre ellos por líneas horizontales y verticales.

Cada ramal puede contener como máximo 6 elementos del circuito en unas posiciones muy determinadas. En las 5 primeras posiciones pueden diseñarse tan sólo contactos y en la última, solamente bobinas.

Además, cada ramal puede subdividirse en dos zonas: la zona condicional, formada por las 5 primeras posiciones, que contiene las condiciones que deben cumplirse para realizar una acción determinada, y la zona ejecutiva, la última posición, que contiene las acciones a realizar cuando se cumplen todas las condiciones.

Esto es igual a lo que sucede en un circuito eléctrico: una carga eléctrica representa la parte ejecutiva de un ramal. Los contactos en serie o en paralelo, de tipo normalmente cerrados o abiertos, representan la parte condicional.

Cada contacto puede ponerse en paralelo solo o por grupos, dibujando contactos sobre el ramal siguiente que se conectarán al ramal base, o al ramal paralelo anterior, a través de líneas verticales. El máximo número de ramales paralelos permitidos es de 4. El conjunto del ramal base y de todos los ramales paralelos conectados a estos se detectará a continuación como recorrido de corriente. El editor numera cada recorrido de corriente automática y correlativamente

También las bobinas pueden conectarse en paralelo pero deben tener la entrada en común. Es decir, un recorrido de corriente puede enviar una única señal lógica a todas las bobinas.

En el interior del esquema de contactos se pueden situar los elementos descritos en la tabla. A estos se añadirán los módulos de sistema, que se relacionarán más adelante y por separado.

Elemento	Descripción
Zona condicional	
	Contacto normalmente abierto , test sobre el estado 1 del operando. El contacto sobre el esquema se cierra cuando el operando correspondiente se encuentra en el estado lógico alto. P. ej., si el operando es una entrada digital del PLC, el contacto está cerrado cuando la entrada está cerrada y viceversa.
	Contacto normalmente cerrado , test sobre el estado 0 del operando. El contacto sobre el esquema se cierra cuando el operando correspondiente se encuentra en el estado lógico bajo. P. ej., si el operando es una entrada digital del PLC, el contacto está cerrado cuando la entrada está abierta y viceversa.
Zona ejecutiva	
	Bobina directa Si la bobina está activa, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico alto. Si la bobina no está activa, al operando se le asigna el valor lógico bajo.
	Bobina inversa Si la bobina está activa, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico bajo. Si la bobina no está activa, al operando se le asigna el valor lógico alto.
	Bobina de set Si la bobina está activa, es decir, si hay una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico alto. Si la bobina no está activa, no se da ninguna acción sobre el estado del operando.
	Bobina de reset Si la bobina está activa, es decir, si hay una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico bajo. Si la bobina no está activa no se da ninguna acción sobre el estado del operando.
	Salto incondicional La elaboración del programa la continua el recorrido cuyo número identificativo se incluye como operando.
	Salto condicionado sobre RA=1 Si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, la elaboración del programa la continua el recorrido cuyo número identificativo se incluye como operando. De lo contrario, la elaboración continua normalmente.
	Salto condicionado sobre RA=0 Si no existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, la elaboración del programa se continua en el recorrido cuyo número identificativo se incluye como operando. De lo contrario, la elaboración continua normalmente.

Módulos de sistema

Generalidades

El PLC PS3 simulado pone a su disposición los siguientes módulos de sistema:

- 32 Temporizadores
- 32 Contadores
- 32 Registros de desplazamiento
- 32 Comparadores

Todos los módulos son unidades de programa acabadas y diferenciadas, y en su interior no se permiten operaciones.

Un módulo determinado sólo puede usarse una vez en el programa. No es necesario utilizar los módulos siguiendo el orden creciente de su numeración.

Todos los módulos son remanentes: conservan el valor al retornar la tensión, si está presente la batería tampón.


Para poder funcionar deben estar conectados al programa del usuario. Esta operación se realiza escribiendo adecuadamente los operandos válidos en las líneas que siguen la llamada al módulo y que el editor AWL añade automáticamente. Si se está programando en ladder, también puede hacerse uniendo los contactos a las entradas del módulo y las bobinas a sus salidas y atribuyéndoles unos operandos. Las bobinas conectadas a las salidas sólo pueden ser de tipo directo o inverso, los otros tipos no están permitidos. Si fuera necesario la conexión de bobinas de set, de reset, o saltos, será preciso memorizar temporalmente la salida del módulo en un merker bit y, a continuación, en el recorrido siguiente, utilizar un contacto del merker para controlar la bobina o el salto deseados.

Si una entrada no se conecta, tomará el valor bajo predeterminado (0).

Temporizadores

El PLC pone a disposición del programador 32 temporizadores con retardo a la activación con direcciones que van de **TR0** a **TR31**.

La representación del módulo es la siguiente.

LADDER	AWL
	<pre>000: TR5 *S: *STP: *IW: *EQ:</pre>

Donde:

S (entrada): en el flanco ascendente, el temporizador se pone en el valor de preset y se inicia.

STOP (entrada): en el estado alto congela el avance del tiempo.

WI (entrada): valor de preset de tiempo sobre 16 bit (de 0 a 65535) con una base de tiempos fija de 0,1 segundos. Por tanto, la temporización puede programarse entre 0 y 6553.5 s (poco más de dos horas).

EQ (salida): salida del temporizador, vale 1 si la entrada S está en 1 y si ha transcurrido el tiempo.

Ante el flanco positivo de la entrada S se activa la temporización para el tiempo expresado en la entrada WI. Cuando acaba el tiempo, se activa la salida Q. La salida se desactiva en correspondencia con la desactivación de la entrada S.

Un estado alto en la entrada STOP congela la temporización, que se reanuda en el punto en que se había detenido, cuando dicha entrada vuelva al estado 0.

La figura que sigue clarifica el funcionamiento del temporizador. En ella $T1 + T2 = T$.

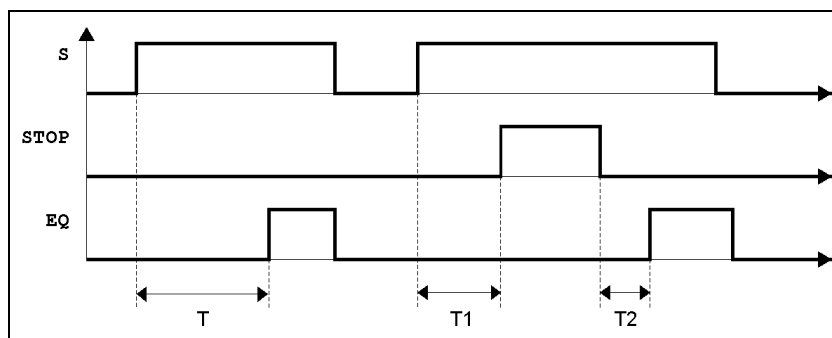


Figura 18: diagrama temporal del temporizador predefinido del PS3

Contadores

El PLC pone a disposición del programador 32 contadores con direcciones entre C0 y C31.

La representación del módulo en los dos lenguajes es la siguiente.

LADDER	AWL
	<pre> 000: C10 *U: *D: *S: *R: *IW: *Z: *QW: </pre>

Donde:

- U** (entrada): en el flanco ascendente se incrementa en 1 el valor de conteo.
- D** (entrada): en el flanco ascendente disminuye en 1 el valor de conteo.
- S** (entrada): en el flanco ascendente, el valor de conteo se pone en el valor de preset.
- R** (entrada): en el estado alto, el valor de conteo se repone a cero.
- WI** (entrada): valor de preset, entre 0 y 65535.
- Z** (salida): salida del contador, en el estado alto cuando el conteo en vigor vale 0.
- QW** (salida): salida de word del contador, informa del valor en vigor de conteo.

En el flanco ascendente de la entrada S, el valor atribuido a la entrada IW se transfiere al contador. Para cada flanco ascendente en la entrada D, el valor del contador disminuye en 1 (conteo hacia atrás). Para cada flanco ascendente de la entrada U, el valor del contador se incrementa en 1 (conteo hacia adelante). Un nivel alto en la entrada R borra el valor del contador e inhibe todo tipo de conteo.

La salida Z será 1 cuando el valor de conteo es 0.

La [Figura 19](#) clarifica el funcionamiento del contador.

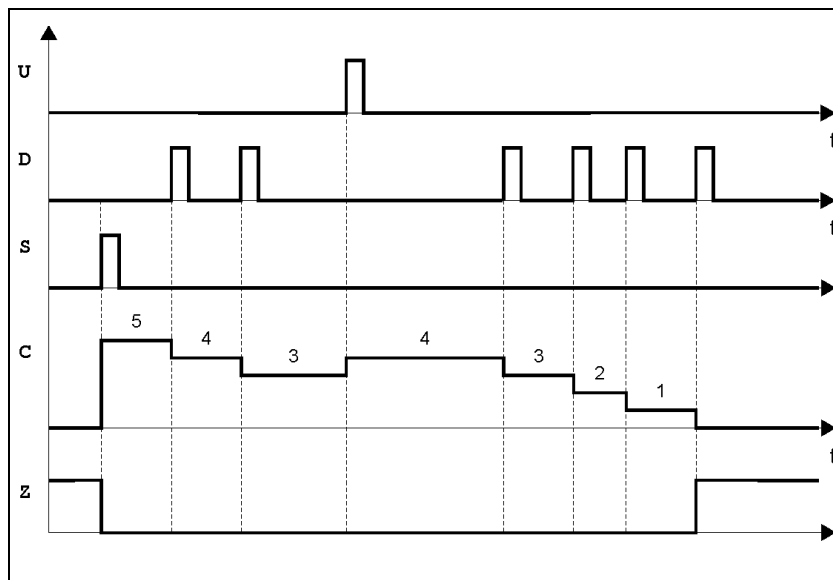


Figura 19: diagrama temporal del contador del PS3

Registros de desplazamiento

El PLC pone a disposición del programador 32 registros de desplazamiento con direcciones entre **SR0** y **SR31**.

La representación del módulo es la siguiente.

LADDER	AWL
	<pre> 000: SR0 *U: *D: *R: *IU: *ID: *Q0: *Q1: *Q2: *Q3: *Q4: *Q5: *Q6: *Q7: *Q8: *Q9: *Q10: *Q11: *Q12: *Q13: *Q14: *Q15: </pre>

Donde:

- U** (entrada): en correspondencia con el flanco ascendente, desplaza el contenido del registro un bit hacia la izquierda.
- D** (entrada): en correspondencia con el flanco descendente, desplaza el contenido del registro un bit hacia la derecha.
- R** (entrada): en el estado alto, repone a cero el contenido del registro y también el de todas las salidas.
- IU** (entrada): valor del bit menos significativo para la siguiente operación de desplazamiento a la izquierda.
- ID** (entrada): valor del bit más significativo para la siguiente operación de desplazamiento a la derecha.
- Q0** (salida): salida 0 igual al valor del bit 0 del registro.
- Q1** (salida): salida 1 igual al valor del bit 1 del registro.
- ...
- ...
- Q15** (salida): salida 15 igual al valor del bit 15 del registro.

A cada flanco ascendente de la entrada U, el registro se desplaza una posición hacia la izquierda, el bit menos significativo toma el valor de IU mientras que el bit más significativo se pierde. A cada flanco

ascendente de la entrada D, el registro se desplaza una posición hacia la derecha. El bit más significativo asume el valor de ID mientras que el bit menos significativo se pierde. En correspondencia con el estado alto a la entrada R, el registro se repone a cero.

Las 16 salidas, de Q0 a Q15, tienen el valor del correspondiente bit del registro.

Comparadores

El PLC pone a disposición del programador 32 comparadores con direcciones entre CP0 y CP31.

La representación del módulo es la siguiente.

LADDER	AWL
	<pre> 000 : CP0 *I1W : *I2W : *GT : *EQ : *LT :</pre>

Donde:

I1(W) (entrada): valor de comparación 1

I2(W) (entrada): valor de comparación 2

GT (salida): es 1 si $I1(W) > I2(W)$

EQ (salida): es 1 si $I1(W) = I2(W)$

LT (salida): es 1 si $I1(W) < I2(W)$

Las entradas están en formato word y las salidas en formato bit.

El módulo compara los valores en las entradas I1 y I2 y programa las salidas tal como se especifica en la tabla.

Entradas	LT	EQ	GT
$I1 < I2$	1	0	0
$I1 = I2$	0	1	0
$I1 > I2$	0	0	1

CAPÍTULO 13

EJEMPLOS DE PROGRAMACIÓN PARA PLC PS3

En este capítulo se exponen, ordenados de menor a mayor complejidad, numerosos ejercicios de programación para el PLC PS3, cuya solución se facilita en forma de esquema de contactos y en la de lista de instrucciones. Para comprobar su funcionamiento tan sólo serán necesarios los interruptores o pulsadores para el control de las entradas del PLC. Los pilotos luminosos situados en la parte frontal del dispositivo valorarán el estado de las salidas. Todos los elementos necesarios se encuentran en la ventana PLC de PC-Sim.

Si durante la instalación de AW-SYS ha optado por instalar también los ejemplos, encontrará todos los que se presentan aquí en la carpeta Aw-sys\Ps3\Esempi de su ordenador. Todos ellos pueden cargarse y verificarse con PC-Sim para PS3.

El nombre del archivo proyecto a cargar tiene el siguiente formato Pxx-yyyL.prj, donde:

- P identifica un proyecto para PS3;
- xx son dos cifras que equivalen siempre a 00;
- yyy es el número correlativo del ejemplo expresado siempre con tres dígitos.
- Z es una letra que especifica el tipo de editor usado en el proyecto (L=ladder, A=AWL).

Es decir, si desea cargar el ejemplo de programación 8, en ladder abra el proyecto P00-008L.prj.

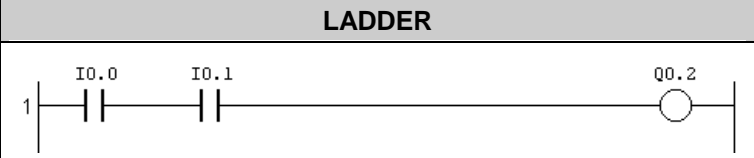
En esa misma carpeta podrá encontrar también otros archivos proyecto, en los que las cifras xx son distintas a 00. En ese caso, los ejemplos se refieren a las instalaciones y xx indica el número correspondiente a la instalación, tal como aparece en el [Menú Instalaciones](#). En la *Guía on line* de PC-Sim podrá encontrar información sobre como cargar y probar los citados ejemplos.

Ejemplo 1: combinación AND

Realizar $q0.2 = I0.1 \text{ AND } I0.1$

La salida $Q0.2$ debe activarse única y exclusivamente si los dos interruptores conectados a las entradas $I0.0$ y $I0.1$ están cerrados.

La solución ladder se obtiene pilotando la bobina $Q0.2$ a través de la serie dos contactos con operandos $I0.0$ y $I0.1$. De hecho, la combinación lógica AND, traducida al lenguaje ladder, equivale a la serie de dos contactos: En la disposición en serie 'se lee' el cierre del circuito solo cuando los dos contactos están cerrados. Así pues, ésta es la única condición que activa la bobina.

LADDER	AWL
	<pre>000: L I0.0 001: A I0.1 002: = Q0.2</pre>

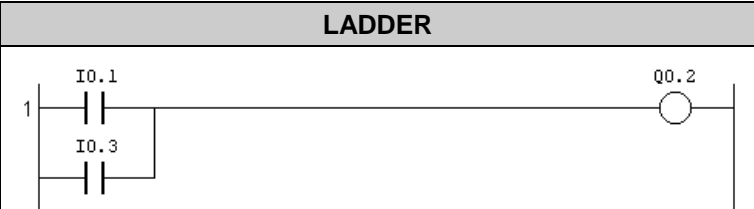
La solución AWL se obtiene cargando primero, el estado de $I0.0$ en el registro general de bit (**L** $I0.0$), a continuación se efectúa una AND entre este último y la entrada $I0.1$ (**A** $I0.1$) y el resultado se deposita asimismo en el registro general. La última instrucción (**=** $Q0.2$) se encarga de transferir el contenido del registro general, que en este instante representa la combinación lógica $I0.0 \text{ AND } I0.1$, a la salida $Q0.2$.

Ejemplo 2: combinación OR

Realizar $q0.2 = I0.1 \text{ OR } I0.3$

La salida $Q0.2$ debe activarse si al menos uno de los interruptores conectados a las entradas $I0.0$ o $I0.1$ está cerrado.

La solución ladder se obtiene pilotando la bobina $Q0.2$ a través del paralelo de dos contactos, con operandos $I0.0$ e $I0.1$. De hecho, la combinación lógica OR, traducida al lenguaje ladder, equivale al paralelo de dos contactos: en las cabezas de la disposición en paralelo 'se lee' el cierre del circuito cuando al menos uno de los contactos está cerrado. Así pues, esta es la condición que conduce a la excitación de la bobina.

LADDER	AWL
	<pre>000: L I0.1 001: O I0.3 002: = Q0.2</pre>

La solución AWL se obtiene cargando primero el estado de $I0.0$ en el registro general de bit (**L** $I0.0$), a continuación, se efectúa una OR entre este último y la entrada $I0.1$ (**O** $I0.1$) y el resultado se deposita

también en el registro general. La última instrucción (= Q0.2) se encarga de transferir el contenido del registro general, que en ese momento representa la combinación lógica I0.0 OR I0.1, a la salida Q0.2.

Ejemplo 3: combinación AND de OR

Realizar $Q0.0 = (I0.0 \text{ OR } I0.1) \text{ AND } (I0.2 \text{ OR } I0.3)$

La solución ladder debería obtenerse de manera intuitiva después de haber realizado los ejercicios anteriores: la bobina Q0.0 es pilotada por la serie (AND) de dos paralelos (OR) de contactos, en correspondencia con los cuales se han conectado adecuadamente los operandos.

LADDER		AWL
		<pre> 000: L I0.0 001: O I0.1 002: L I0.2 003: O I0.3 004: A 005: = Q0.0 </pre>

En cambio, la solución AWL requiere alguna aclaración, ya que se ha introducido una nueva operación que utiliza el registro de stack. En primer lugar, observamos que después de la ejecución de la instrucción 001, el registro general de bit contiene el resultado de la combinación lógica OR entre I0.0 y I0.1 (ver Ejemplo de programación 2). La siguiente operación de carga, y puesto que la secuencia aun no está concluida, introduce primero el contenido del registro general en el registro de stack, y luego carga el estado del operando especificado en el registro general. Una vez se ha ejecutado la instrucción 003, el registro de stack contiene aún, en primera posición, el valor calculado con anterioridad I0.0 OR I0.1 y el registro general contiene el valor de I0.2 OR I0.3, tal y como se ha obtenido ejecutando las instrucciones 002 y 003. La instrucción 004, una AND sin operando, ordena a la CPU del PLC que ejecute una AND entre el registro de stack y el registro general, es decir, en este caso, entre los resultados de las dos combinaciones OR, y que deposite de nuevo el resultado en el registro general. Tan sólo queda copiar el contenido de este último en la salida Q0.0 con la última instrucción.

Ejemplo 4: combinación OR de AND

Realizar $Q0.0 = (I0.0 \text{ AND } I0.1) \text{ OR } (I0.2 \text{ AND } I0.3)$. Donde los paréntesis, aunque no sean necesarios dado que la operación AND tiene preferencia sobre la OR, se han añadido para mayor claridad.

En la solución ladder, la bobina Q0.0 se pilota por el paralelo (OR) de dos serie (AND) de contactos, en correspondencia con los cuales se han conectado adecuadamente los operandos.

LADDER		AWL
		<pre> 000: L I0.0 001: A I0.1 002: L I0.2 003: A I0.3 004: O 005: = Q0.0 </pre>

En la solución AWL observamos en primer lugar, que después de la ejecución de la instrucción 001, el registro general de bit contiene el resultado de la combinación lógica AND entre I0.0 y I0.1 (ver Ejemplo de programación 1). La siguiente operación de carga, dado que aun no ha concluido la secuencia, introduce primero el contenido del registro general en el registro de stack, y luego carga el estado del operando especificado en el registro general. Una vez se ha ejecutado la instrucción 003, el registro de stack contiene aun, en primera posición, el valor calculado con anterioridad I0.0 AND I0.1 y el registro general contiene el valor de I0.2 AND I0.3, tal y como se ha obtenido siguiendo las instrucciones 002 y 003. La instrucción 004, una OR sin operando, ordena a la CPU del PLC que ejecute una OR entre el registro de stack y el registro general, es decir, en este caso, entre los resultados de las dos combinaciones AND, y que deposite de nuevo el resultado en el registro general. Tan sólo queda copiar el contenido de este último en la salida Q0.0 con la última instrucción.

Ejemplo 5: combinación XOR

Realizar $Q0.0 = I0.1 \text{ XOR } I0.2$

La operación lógica XOR aplicada a dos variables booleanas da resultado cierto solo cuando únicamente una de las dos variables es cierta.

La primera serie de contactos del programa ladder está cerrada sólo cuando I0.1 está cerrada y I0.2 está abierta; la segunda serie está cerrada sólo cuando, por el contrario, I0.1 está abierta y I0.2 está cerrada. Realizando el paralelo de las dos se obtiene la función deseada: la bobina se activa tan sólo cuando una entrada está cerrada y la otra está abierta.

LADDER		AWL
		<pre> 000: L I0.1 001: AN I0.2 002: LN I0.1 003: A I0.2 004: O 005: = Q0.2 </pre>
		<pre> 000: L I0.1 001: XO I0.2 002: = Q0.2 </pre>

La primera solución AWL no es más que la traducción del programa ladder y su funcionamiento es muy similar al del ejemplo anterior. La segunda solución utiliza, por el contrario, la operación **xo** que realiza directamente la XOR entre los operandos.

Ejemplo 6: autorretención

Un pulsador conectado a la entrada I0.0 debe activar la salida Q0.15; y un segundo pulsador conectado a la entrada I0.1 debe desactivarla.

En el programa ladder propuesto se realiza un circuito con autorretención. Pulsando el botón conectado a I0.0 la bobina Q0.15 se activa y entonces, el contacto con el mismo operando en la segunda línea se cierra (imaginen que el contacto y la bobina son parte del mismo relé Q0.15) y continúa manteniendo activada la bobina incluso después de la apertura de I0.0. El cierre del pulsador en la entrada I0.1 provoca la apertura del contacto, normalmente cerrado en el esquema, desactivando la bobina y cortando la autorretención.

LADDER	AWL
	<pre> 000: L I0.0 001: O Q0.15 002: AN I0.1 003: = Q0.15 </pre>

El programa AWL propone la conversión de todo cuanto se acaba de describir. El valor de la salida Q0.15 en la última instrucción se calcula cargando el estado de I0.0, a continuación, valorando la OR con Q0.15 y por último poniendo en AND el resultado con el complemento de I0.1.

Ejemplo 7: set y reset

Un pulsador conectado a la entrada I0.0 debe activar la salida Q0.15; un segundo pulsador, conectado a la entrada I0.1, debe desactivarla

El ejercicio es idéntico al anterior pero, en esta ocasión, en la solución se emplean bobinas de set y reset de la misma salida Q0.15 pilotadas por separado por medio de contactos de I0.0 y I0.1.

LADDER	AWL
	<pre> 000: L I0.0 001: S Q0.15 002: L I0.1 003: R Q0.15 </pre>

Si hacemos funcionar el programa, observamos que al presionar dos pulsadores a la vez se desactiva la salida. En efecto, en ambos lenguajes, cuando las condiciones de test son válidas a la vez, el operando

Q0.15 se activa primero, en el recorrido 1 o con las dos primeras instrucciones, y luego se desactiva, en el recorrido 2 o con la tercera y cuarta instrucción. Pero recordemos que Q0.15 no representa efectivamente la salida física del PLC, sino el correspondiente bit en el interior de la memoria de las imágenes de proceso. Dicho bit de memoria es llevado efectivamente a 1 y luego a 0 pero, tan sólo al final de la carga del programa utilizado, el valor elaborado por él se transfiere al canal físico de salida correspondiente, que se mantiene constantemente en el valor bajo cuando ambas entradas están cerradas.

Con esta escritura de programas hemos hecho prevalecer el reset respecto al set. Si desea obtener lo contrario, le bastará con invertir la posición de los recorridos en el esquema de contactos o cambiar la primera secuencia por la segunda en el programa AWL.

Ejemplo 8: activación por flancos

Activar las salidas Q0.8 y Q0.9 que corresponden, respectivamente, al flanco ascendente y al flanco descendente de la entrada I0.0.

Observamos que el último recorrido del esquema ladder y las dos últimas instrucciones del programa AWL imponen, al final de la ejecución del programa, la igualdad del merker bit M0.0 al estado de la entrada I0.0. Pero, en correspondencia con los flancos y para los recorridos o las instrucciones anteriores, se da el hecho que el estado de las dos variables es opuesto y que tan sólo al final de la carga del programa utilizado se convierten en iguales. Todo ello queda representado en las dos primeras líneas del diagrama de [Figura 20](#) con un retraso temporal entre M0.0 respecto a I0.0 que equivale a un ciclo de ejecución.

LADDER		AWL
1	I0.0 M0.0 /	Q0.8
2	I0.0 M0.0 /	Q0.9
3	I0.0 	M0.0
		000: L I0.0 001: AN M0.0 002: = Q0.8 003: LN I0.0 004: A M0.0 005: = Q0.9 006: L I0.0 007: = M0.0

La parte inicial de los dos programas activa la bobina Q0.8, para un ciclo de ejecución, cuando I0.0 está en 1 y M0.0 está en 0, es decir, en correspondencia con el flanco ascendente de I0.0, tal como aparece indicado en la tercera línea del diagrama.

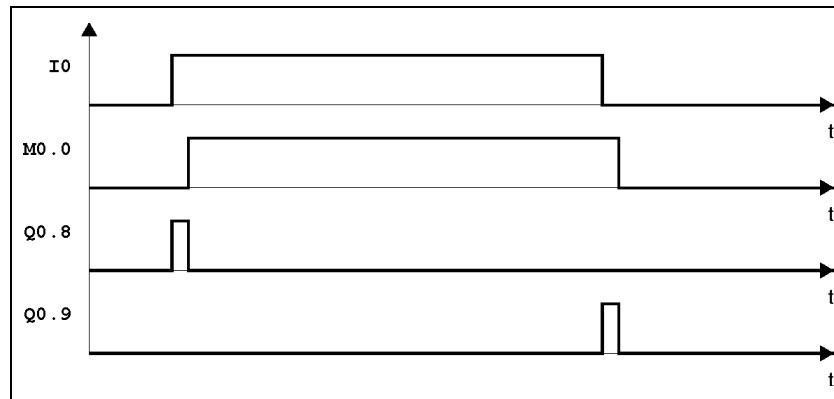


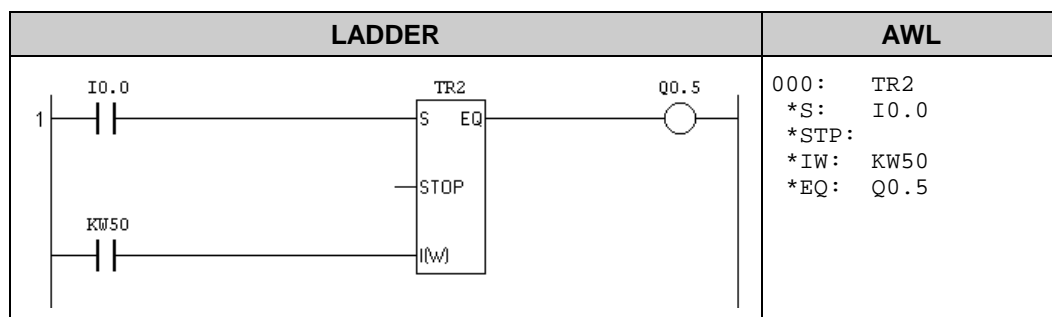
Figura 20: diagrama temporal de un elevador de flancos

En cambio, la bobina $Q0.9$ se activará en el segundo recorrido o a la segunda secuencia, siempre para un ciclo de ejecución, cuando $I0.0$ está en 0 y $M0.0$ está en 1, es decir, en correspondencia con el flanco descendente de $I0.0$, tal como aparece indicado en la cuarta línea del diagrama.

Ejemplo 9: temporizador con retardo en la activación

La salida $Q0.0$ se activa 5 segundos después de la activación de la entrada $I0.2$. Cuando la entrada se hace baja, la salida se desactiva.

Los programas siguientes resuelven el problema en Ladder y en AWL.



La solución es sumamente sencilla ya que existe un módulo de sistema que realiza precisamente la función requerida por el trazado: el temporizador con retardo a la activación, el único tipo de temporizador que este PLC posee (ver [Figura 18](#)).

En ladder se trata de conectar las entradas y la salida del módulo de sistema a contactos y bobinas. En correspondencia con estos deberá indicar los operandos apropiados. Así, la entrada S se ha conectado a un contacto NA de $I0.0$ mientras que la salida EQ se ha conectado a una bobina de $Q0.5$ y la entrada I(W) se ha conectado a un contacto con operando $KW50$ (50 décimas de segundo).

Las operaciones que hay que realizar para la escritura del programa AWL son similares. Después de haber insertado la sigla del temporizador, hay que teclear los operandos de los que el módulo deberá tomar los datos o a los que el módulo deberá enviar las órdenes.

Tanto en el programa ladder como en el AWL, la entrada STOP no se utiliza. En el primer caso es suficiente con no conectar nada a la entrada. En el segundo, basta con dejar vacía la línea correspondiente. En ambos casos, la entrada del módulo se situará automáticamente a valor lógico bajo.

Ejemplo 10: temporizador con retardo en la desactivación

La salida $Q0.5$ debe activarse simultáneamente con el cierre de la entrada $I0.0$ y desactivarse 5 segundos después de su apertura.

Para resolver este problema, y dado que este PLC no está dotado de temporizadores de ese tipo, es preciso utilizar el único temporizador que se proporciona, retardo a la activación, y escribir un programa que modifique su funcionamiento. Consulte la [Figura 21](#) que clarifica el método empleado para solucionar este problema.

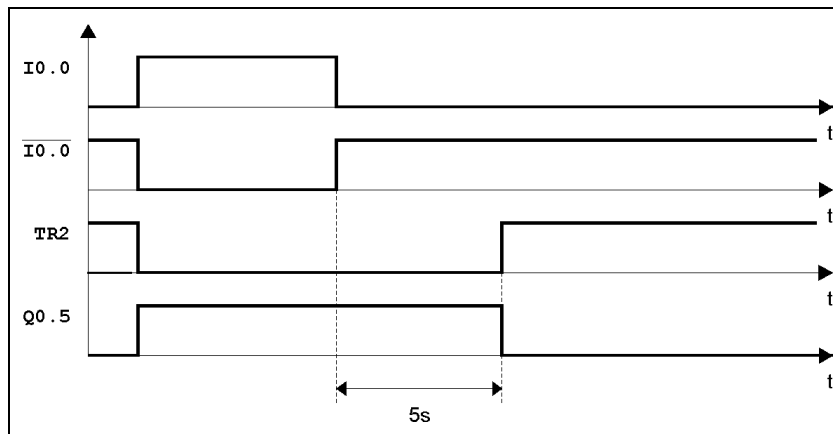
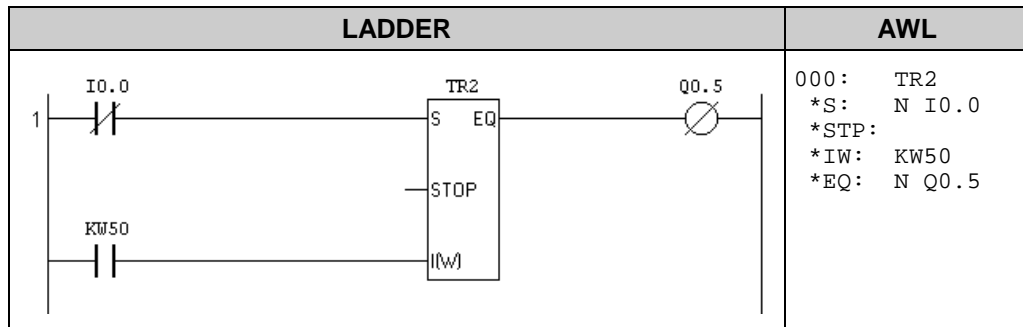


Figura 21: diagrama temporal del temporizador con retardo a la desactivación

La primera y la última línea del diagrama representan el desarrollo temporal de la entrada y de la salida de un temporizador con retardo a la desactivación: la salida se activa en correspondencia con el flanco ascendente de la entrada y se desactiva, después de un tiempo prefijado, del flanco descendente de la misma señal.

La segunda línea del diagrama representa la señal que se obtiene complementando la entrada. En cambio, la tercera línea representa la salida de un temporizador con retardo a la activación que tiene como entrada la señal de la línea superior (ver [Figura 18](#)). Observemos como esta señal representa el complemento de la que buscamos.

Es decir, como conclusión, un temporizador con retardo a la desactivación se obtiene a partir de uno con retardo a la activación, simplemente negando las señales de entrada y de salida.



En ladder la negación de una entrada se obtiene utilizando un contacto normalmente cerrado (test sobre el estado negado de la señal) y la negación de una salida se obtiene utilizando una bobina inversa. En lenguaje AWL basta con hacer que el indicador de complemento 'N' preceda a los operandos.

Ejemplo 11: temporizador a impulso

La salida $Q0.0$ se activa con el cierre de la entrada $I0.0$ y se desactiva 5 segundos después. Si la entrada se reabre en este periodo, la salida se desactiva inmediatamente.

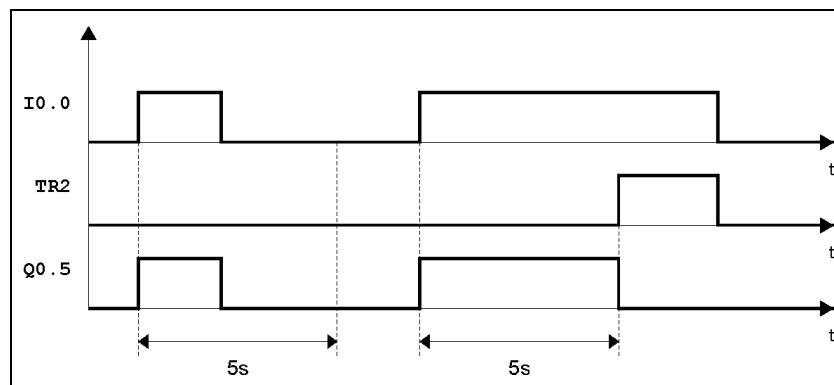
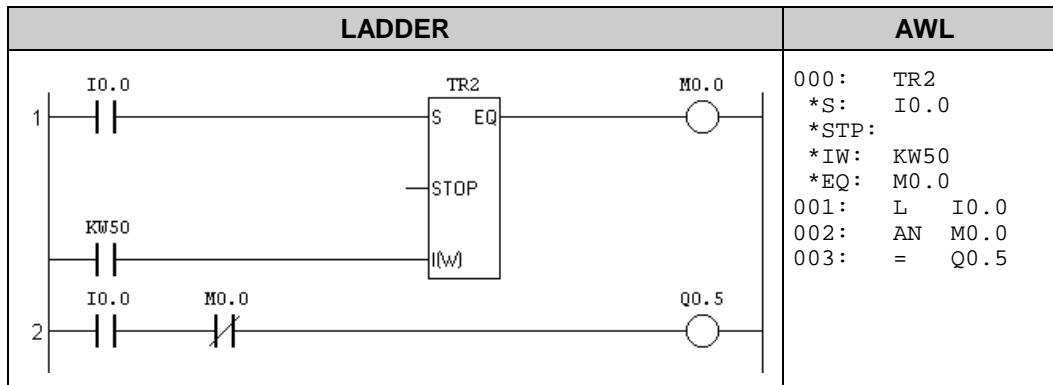


Figura 22: diagrama temporal de un temporizador a impulso

La primera línea del diagrama temporal de [Figura 22](#) indica la entrada del temporizador a impulso y la última indica su salida.

También en este caso debemos utilizar el único temporizador que este PLC posee, el retardo a la activación, para conseguir un temporizador del tipo deseado. Con ese propósito se ha construido la línea intermedia del diagrama que representa la salida de un temporizador con retardo a la activación, a cuya entrada se ha conectado la señal de la primera línea.

Observamos pues que la salida $Q0.5$ debe ser cierta (estado lógico 1) cuando la entrada $I0.0$ está en el estado alto y, a la vez, la salida del temporizador se encuentra en el estado bajo. En términos de expresión booleana: $Q0.5 = I0.0 \text{ AND NOT } TR2$.



La solución en ladder prevé entonces la escritura de un primer recorrido para la activación de un temporizador con retardo a la activación con entradas (S) $I0.0$ y (IW) $KW50$ (5 segundos), la salida está apoyada en un merker bit ($M0.0$). En el segundo recorrido de corriente, la bobina de la salida $Q0.5$ está pilotada por la serie de la entrada y por el complemento de $M0.0$ que corresponde a la salida del temporizador.

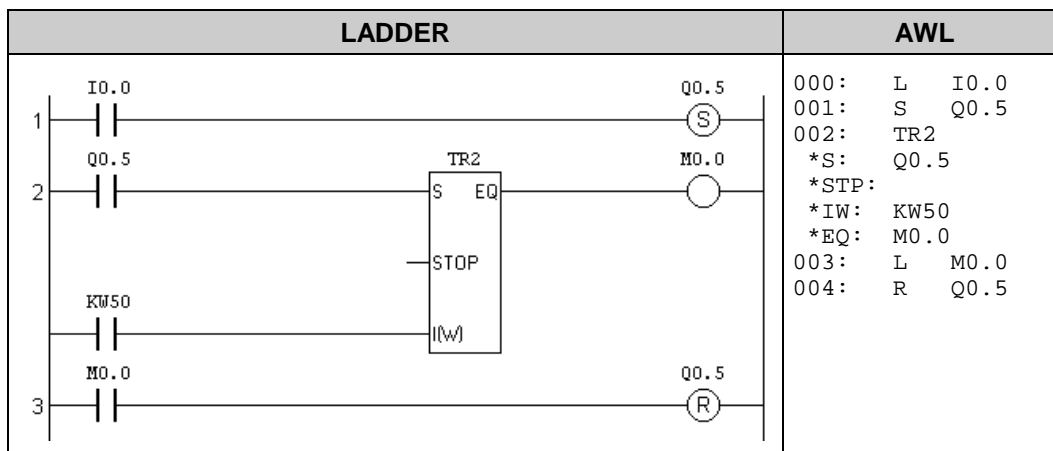
La solución AWL, aunque presente una escritura diversa, es análoga a la ladder.

Ejemplo 12: temporizador a impulso prolongado

La salida $q0.5$ se activa al cierre de la entrada $I0.0$ y se desactiva 5 segundos después, con independencia de si la entrada se reabre o no durante este periodo.

El funcionamiento del temporizador a impulso prolongado se deduce de la comparación de la primera y la última línea del diagrama temporal de [Figura 23](#). En la primera línea está representada la señal de entrada y en la última, la correspondiente salida.

También en este caso debemos utilizar el único temporizador que posee este PLC, el retardo a la activación, para conseguir un temporizador del tipo deseado. La misión es llevar al estado alto (activar) la salida $Q0.5$ cuando la entrada va alta y llevarla al estado bajo (desactivarla) cuando haya transcurrido el tiempo.



El primer recorrido del diagrama ladder se encarga de la primera tarea: activar la salida cuando la entrada está alta. El temporizador del recorrido siguiente se inicia por medio de un contacto de la propia salida y,

pasados 5 segundos, se encarga de activar el merker $M0.0$. En el último recorrido, el mismo merker lleva de nuevo a 0 la salida $Q0.5$. El apoyo de la salida de $TR2$ en $M0.0$ tan sólo es necesario porque no se puede utilizar la bobina de reset de $Q0.5$ directamente a la salida del temporizador. Recuerde que, a las salidas de los módulos de sistema, únicamente pueden conectarse bobinas directas o inversas.

La *Figura 23* representa el diagrama temporal de los operandos principales presentes en este ejemplo. Además de la entrada y la salida del PLC, en la primera y la última línea respectivamente, en la línea intermedia se representa la salida del temporizador $TR2$ (correspondiente perfectamente al merker $M0.0$). Observe como esta señal se mantiene activa durante un tiempo muy breve, un sólo ciclo de ejecución del programa, que en el diagrama se ha alargado para facilitar su lectura. En efecto, en cuanto se ha activado, la salida del temporizador se encarga de desactivar $Q0.0$ pero, dado que ésta representa su entrada, en la siguiente elaboración del programa, se repone a cero, incluso el estado del temporizador.

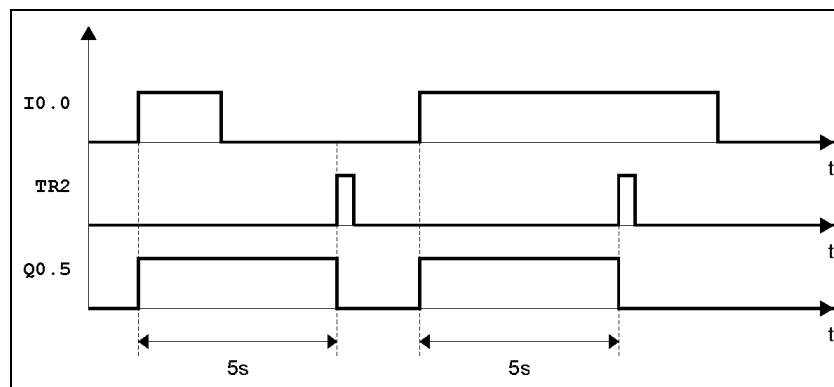


Figura 23: diagrama temporal del temporizador a impulso prolongado

Por último, observemos como, en realidad, en el primer recorrido se continúa programando la salida durante todo el tiempo en que la entrada es alta, pero el siguiente reset al recorrido 3 sobrescribe la imagen de las salidas, impidiendo que el estado alto en esta condición se propague a la periferia durante la transferencia de la imagen.

También en esta ocasión, el programa AWL es la fiel traducción del ladder.

Ejemplo 13: temporizador con retardo a la activación con memoria

La salida $Q0.5$ se activa 5 segundos después del cierre de la entrada $I0.0$ (aunque si este último se reabre mientras tanto) y se desactiva en correspondencia con el cierre de la entrada $I0.1$.

El funcionamiento del temporizador con retardo a la activación con memoria se deduce de la comparación de las dos primeras líneas del diagrama temporal de *Figura 24* con la última. Las primeras líneas representan las señales de entrada y la última informa de la salida correspondiente.

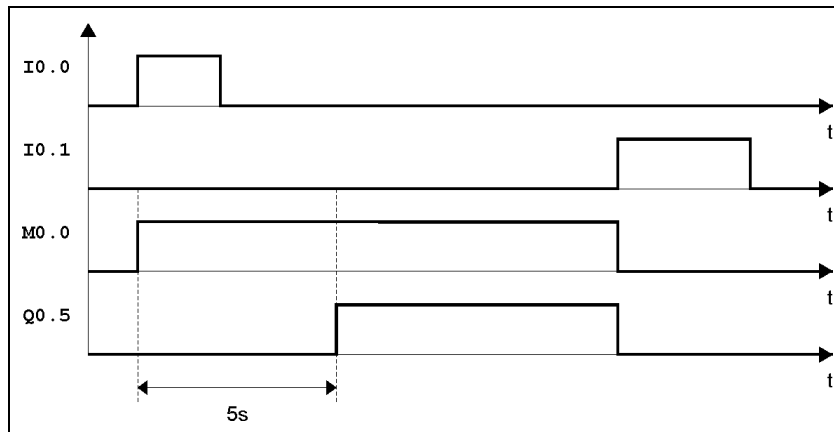


Figura 24: diagrama temporal temporizador con retardo a la activación con memoria

También en este caso debemos utilizar el único temporizador que este PLC posee, el retardo a la activación, para obtener un temporizador del tipo deseado. Con tal propósito, utilizamos un merker que se activa desde $I0.0$ y se restaura desde $I0.1$, y cuyo trazado se detalla en la tercera línea del diagrama. Observe como la salida $Q0.5$, respecto de esta última, no es otra cosa que un retardo a la activación: ¡Ya tiene la solución!

LADDER		AWL
1	$I0.0$ (NO) — $M0.0$ (S)	000: L $I0.0$
2	$I0.1$ (NC) — $M0.0$ (R)	001: S $M0.0$
3	$M0.0$ (NO) — TR2 (S EQ)	002: L $I0.1$
	TR2 (S EQ) — $Q0.5$ (C)	003: R $M0.0$
	TR2 (S EQ) — STOP	004: TR2
	KW50 — I(W)	*S: $M0.0$
		*STP:
		*IW: KW50
		*EQ: $Q0.5$

En el esquema de contactos, los dos primeros recorridos tienen la finalidad de programar el estado del merker $M0.0$. Desde éste, en el tercer recorrido, se conecta la salida, a través de un temporizador con retardo a la activación.

El programa AWL es la traducción del ladder.

Ejemplo 14: temporizador con retardo a la activación y a la desactivación

La salida $Q0.7$ se activa 2 segundos después del cierre de la entrada $I0.1$ y se desactiva 7 segundos después de su reapertura.

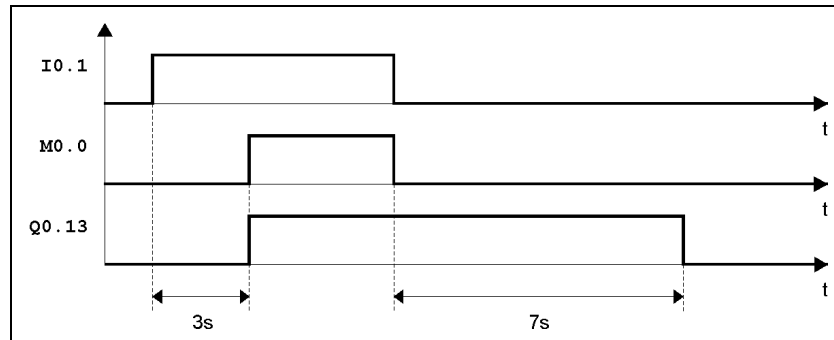
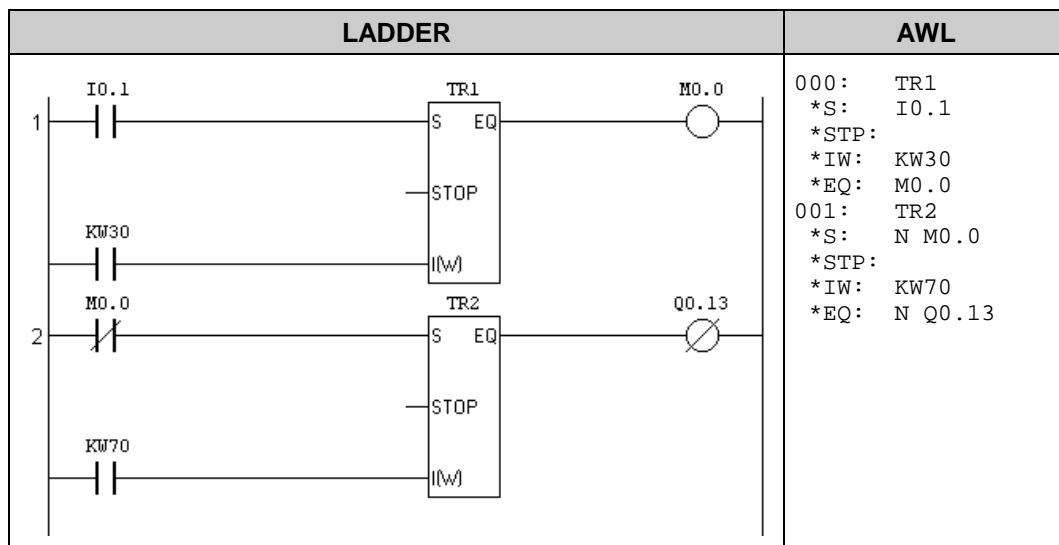


Figura 25: diagrama temporal del temporizador con retardo a la activación y a la desactivación

La [Figura 25](#) ilustra, en la primera y la última línea del diagrama, el desarrollo de las dos señales del trazado. La línea intermedia muestra el desarrollo de un merker que es la salida de un temporizador con retardo a la activación cuya entrada es precisamente $I0.1$. Observe que el proceso de la salida $Q0.13$, respecto al desarrollo del merker, representa un retardo a la desactivación. El temporizador con retardo a la activación está presente en el PLC mientras que, en uno de los ejemplos precedentes, hemos aprendido a construir uno con retardo a la desactivación. ¡Hagan juego, señores! Ahora se trata de escribir un programa que contenga dos temporizadores: el primero, un retardo a la activación, tiene por entrada $I0.1$ y como salida $M0.0$. El segundo es un retardo a la desactivación, que tiene como entrada $M0.0$ y como salida $Q0.13$.

Los programas ladder y AWL indicados a continuación implementan esta solución.



Ejemplo 15: impulso retardado

La salida $Q0.7$ se activa 2 segundos después de la apertura de la entrada $I0.13$ durante un período de 1 segundo.

El diagrama en [Figura 26](#) ilustra en la primera línea, el desarrollo de la entrada, y en la última, el de la salida del temporizador requerido. Por su parte, la segunda línea indica el desarrollo del complemento de $I0.13$. Esta señal se utilizará para poner en marcha dos temporizadores con retardo a la activación $TR10$ y $TR11$, en 2 y 3 segundos respectivamente, y cuyas señales de salida quedan reseñadas en las siguientes líneas del diagrama.

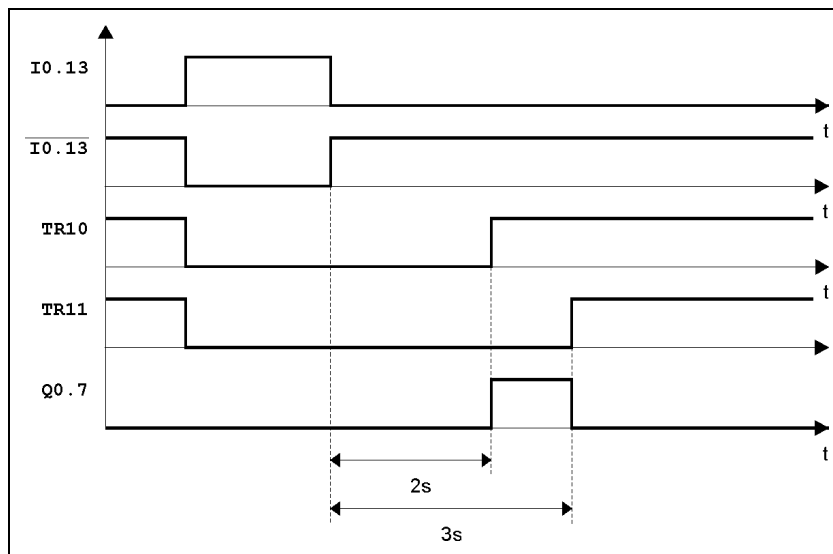


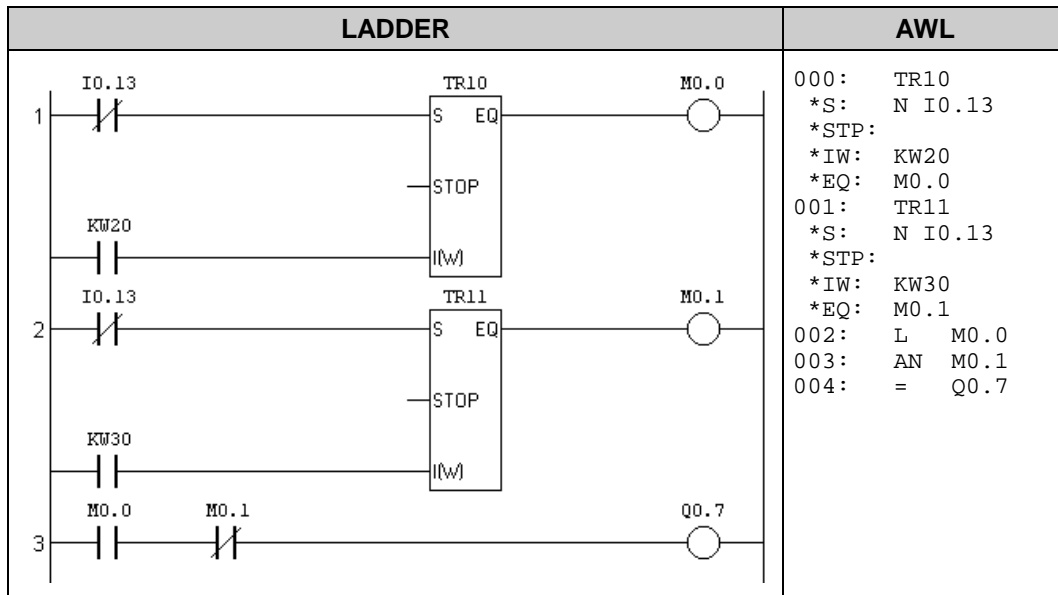
Figura 26: diagrama temporal del impulso retardado

Observe entonces que la salida $Q0.7$ debe ser cierta cuando se dan a la vez las siguientes condiciones: salida de $TR10$ cierta y salida de $TR11$ falsa, es decir, en términos de expresión booleana:

$$Q0.7 = TR10 \cdot \overline{TR11}$$

En el programa ladder, los dos primeros recorridos están dedicados a los temporizadores: ambos tienen por entrada el complemento de $I0.13$ (contacto NC = test sobre el estado 0); las salidas están apoyadas en dos merker. En el tercer recorrido se implementa la expresión lógica recién obtenida: los dos contactos puestos en serie desarrollan la función de la operación AND. En particular, para el segundo se ha usado el tipo NC con el fin de ejecutar el complemento de su operando.

De manera análoga, en la solución AWL, los dos temporizadores se requieren y se conectan al programa del usuario. Entonces, la secuencia final elabora el valor de la salida.



Ejemplo 16: tren de impulsos

La salida $Q0.4$ debe activarse un instante a cada segundo.

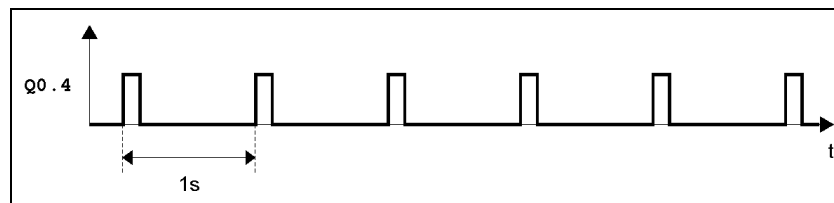
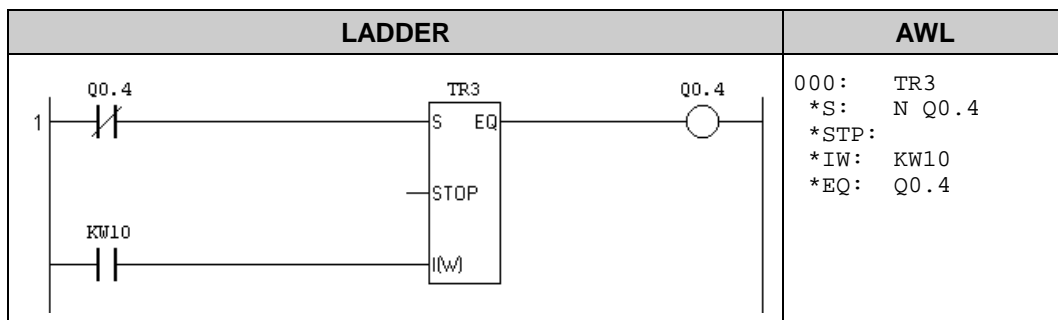


Figura 27: diagrama temporal del tren de impulsos

El ejercicio se resuelve utilizando un temporizador que se autoinicia cíclicamente.

En el primer ciclo de ejecución, el complemento de $Q0.4$ está alto. La temporización empieza y, mientras el tiempo va transcurriendo, la salida de $TR3$ se mantiene baja. Cuando el tiempo llega a cero, la salida de $TR3$ se hace alta así como el estado de $Q0.4$ conectado a ella. En el ciclo siguiente, la entrada del temporizador será de nuevo baja y también lo será su salida.

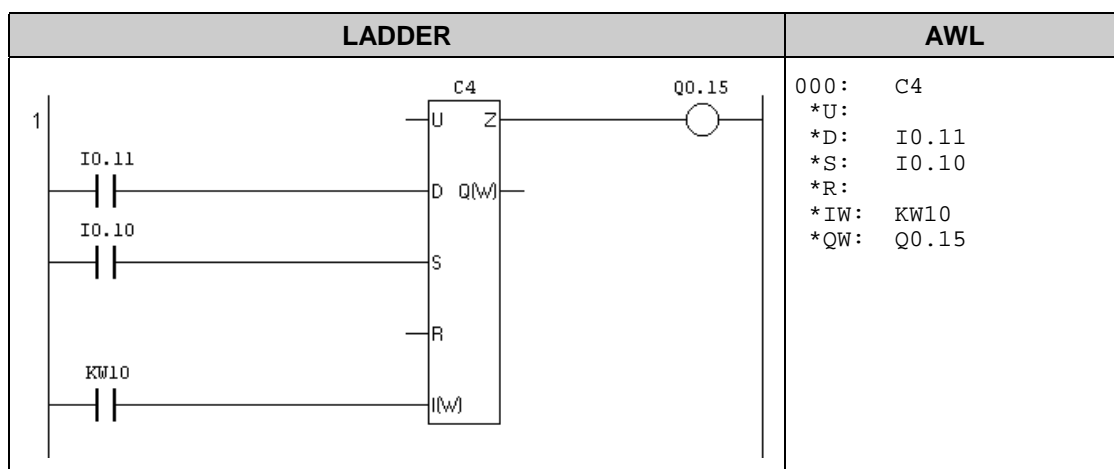
De ese modo, hemos vuelto a las condiciones iniciales: el funcionamiento descrito se repetirá indefinidamente, generando el tren de impulsos requerido por el trazado.



Ejemplo 17: conteo hacia atrás

El contador **C4** se pone en el valor 10 en correspondencia con la activación de la entrada **I0.10** y este valor disminuye a cada cierre de la entrada **I0.11**. La salida **Q0.15** se activa en correspondencia con el valor 0 del conteo.

El programa ladder es muy sencillo. La constante **KW10**, asignada a la entrada **IW**, facilitará el valor de preset 10. La entrada de set (S) del contador **C4** se conecta a un contacto de **I0.10**, mientras que el decremento (D) se conecta a un contacto de **I0.11**. En el flanco de cierre de **I0.10** se produce la programación del valor de conteo a 10 y en cada flanco de cierre de **I0.11** el conteo va disminuyendo en 1. La salida del contador (Q) está alta cuando el valor de conteo es igual a 0. Por tanto, será suficiente conectar a ésta una bobina de la salida **Q0.15** del PLC.

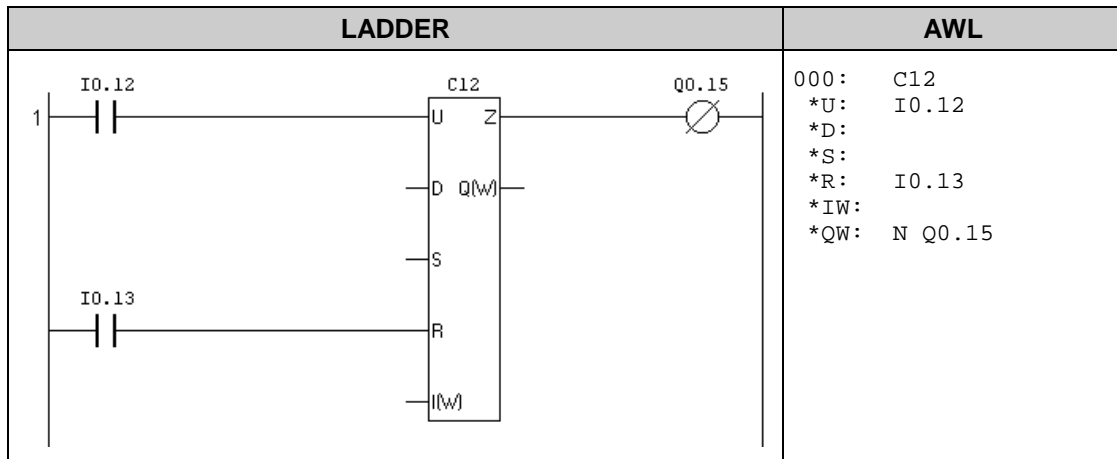


La instrucción 000 del programa AWL constituye una llamada al módulo de sistema **C4** que está conectado de una manera análoga a como se hizo en el programa ladder.

Ejemplo 18: conteo hacia adelante

El contador **C12** se incrementa a cada cierre de la entrada **I0.12** y se repone a cero en correspondencia con el estado alto de la entrada **I0.13**. La salida **Q0.15** está activa cuando el valor de conteo es distinto de cero.

La entrada de reset (R) del contador **C12** se conecta a un contacto de **I0.13**, mientras que el de incremento (U) se conecta a un contacto de **I0.12**. En el flanco de cierre de **I0.13** se produce el reset del contador, es decir, la reposición del valor de conteo a 0. Por el contrario, en cada flanco de cierre de **I0.12** el conteo se incrementa en 1. La salida del contador (Q) es alta cuando el valor de conteo es igual a 0. Así pues, será suficiente conectar a ésta una bobina inversa de la salida **Q0.15** del PLC para obtener el funcionamiento deseado.



La instrucción 000 del programa AWL constituye una llamada al módulo de sistema C12 que se conecta de una manera análoga a como se ha hecho en el programa ladder.

Ejemplo 19: conteo del tiempo de cierre de una entrada (en segundos)

Determinar cuantos segundos la entrada I0.5 permanece cerrada y utilizar la entrada I0.6 para restaurar el conteo del tiempo.

En primer lugar, es preciso realizar una base de tiempos de un segundo, es decir, un tren de impulsos que tenga dicho período. Luego, será necesario contar cuantos impulsos de la base de tiempos se generan durante el cierre de la entrada, o sea, contar los instantes en los cuales la entrada y el impulso son ciertos a la vez (ver [Figura 28](#)).

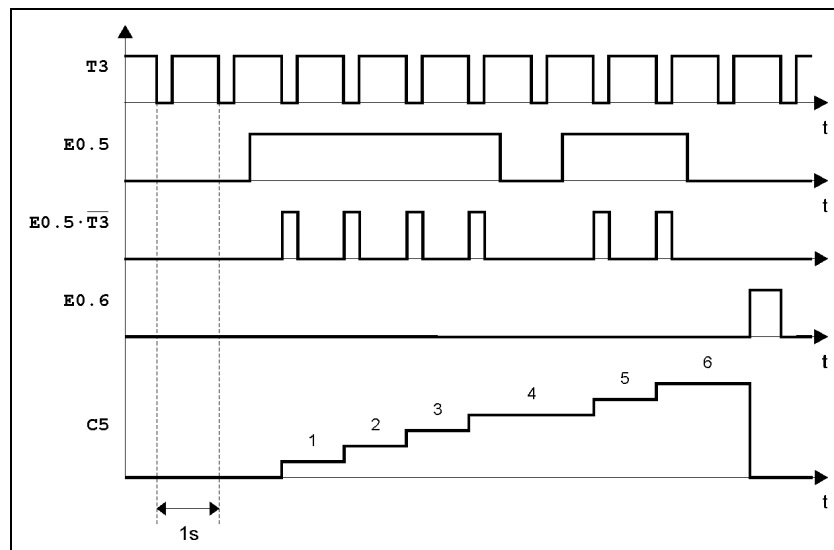
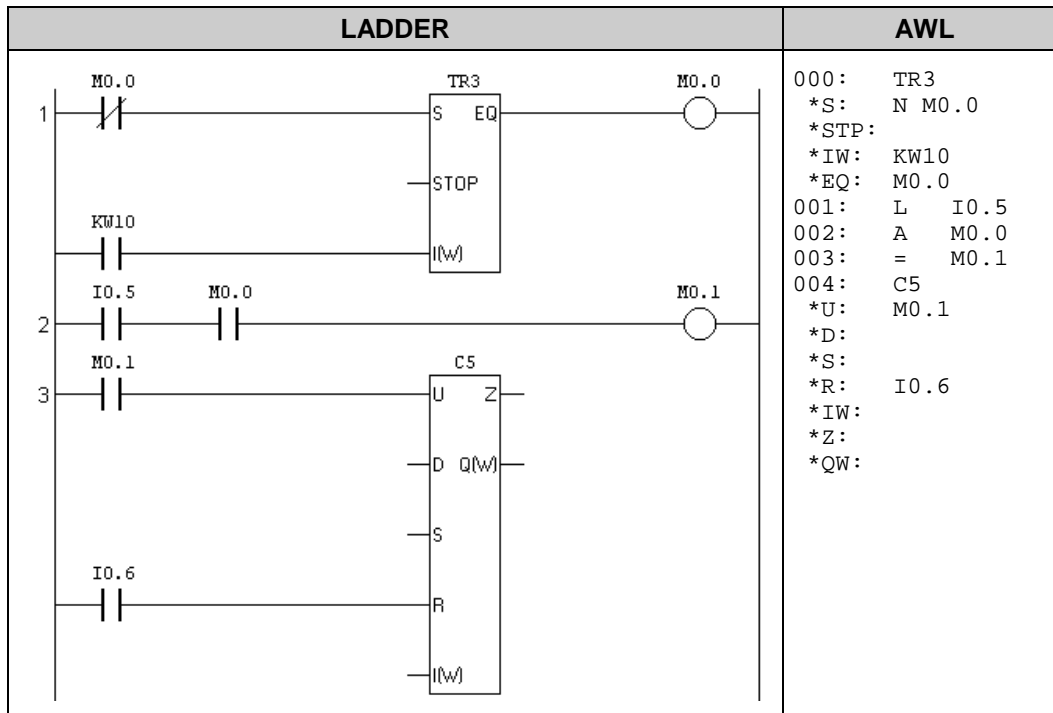


Figura 28: diagrama temporal del contador de tiempo de cierre de una entrada



El recorrido 1 del programa ladder genera el tren de impulsos de 1 segundo de período, tal como hemos visto en el Ejemplo 16. En esta ocasión utilizamos un merker (M0.0) como entrada/salida del temporizador, ya que no se requiere llevar hasta el exterior del PLC la señal generada. Por su parte, el merker M0.1 se pone en la combinación lógica AND (serie de contactos en el diagrama) entre la entrada y el tren de impulsos, es decir, lo que representa justo los impulsos a contar. Este merker se emplea en el recorrido siguiente para pilotar la entrada de conteo hacia adelante del contador C5. Por su parte, la entrada I0.6 pilota la entrada reset del contador para obtener la reposición del conteo, tal como se ha requerido.

El programa AWL realiza las mismas funciones de un modo análogo

El valor del contador C5 representa el número de segundos que la entrada ha estado cerrada, con el límite de 65535 propio de los contadores de este PLC.

Ejemplo 20: conteo tiempo de cierre de una entrada (en horas, minutos y segundos)

Determinar cuantas horas, minutos y segundos se mantiene cerrada la entrada I0.5 y utilizar la entrada I0.6 para reponer a cero el conteo del tiempo.

Los programas propuestos como solución concluyen con la construcción de un tren de impulsos con merker **M0.0**, que servirá de base de tiempos con un período de 1 segundo. Pero al inicio de los mismos, se carga el valor del merker **M0.1**, que se programará a 1 en presencia de un impulso de la base de tiempos, cuando el contacto en la entrada **I0.5** esté cerrado. Entonces, **M0.1** se activa cada segundo cuando la entrada está cerrada: el conteo de los impulsos de este merker nos permitirá valorar el tiempo tal como se requiere en el trazado.

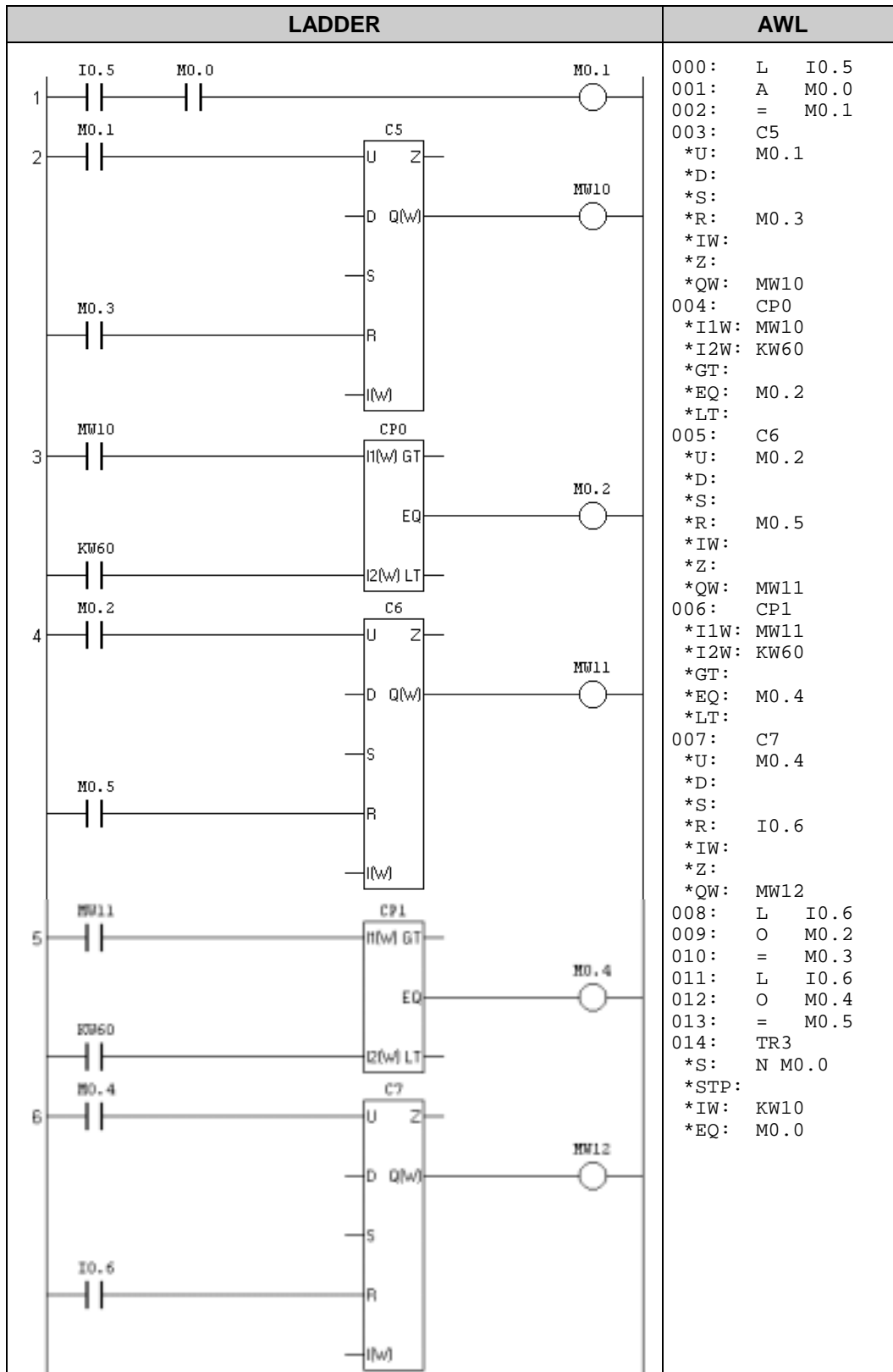
En efecto, la entrada de conteo hacia adelante (U) del contador **C5** está pilotado precisamente **M0.1**; por lo tanto, **C5** constituye el contador de los segundos. El valor vigente del conteo, presente en su salida QW, se deposita en la merker word **MW10**.

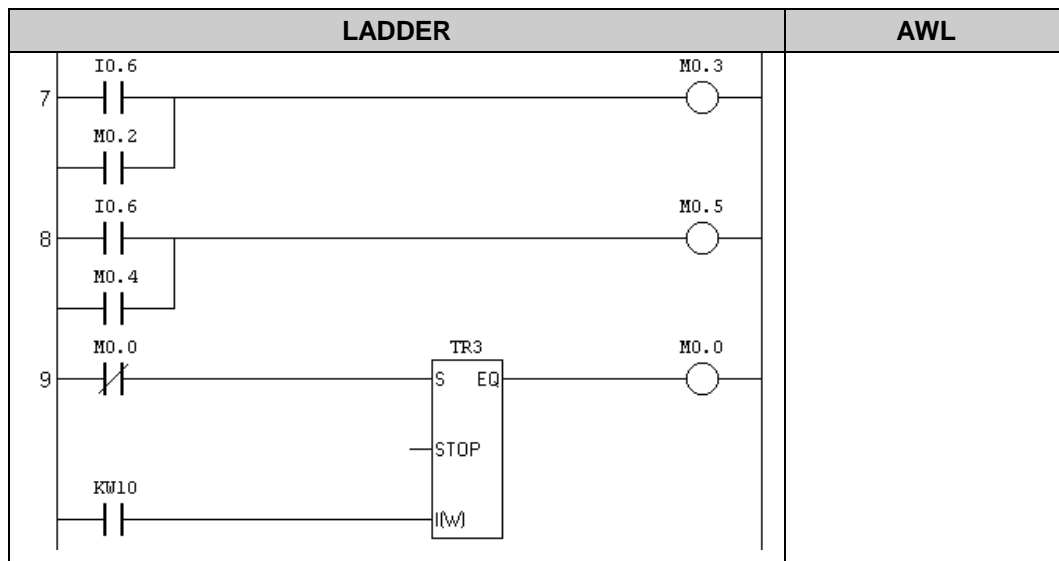
A continuación de la llamada de **C5**, encontramos la del otro módulo de sistema **CP0** (comparador) que se emplea para comparar **MW10**, es decir, el número de segundos contados con la constante numérica 60 (**KW60**). Cuando las dos cantidades resultan iguales, es decir, cuando se han contado 60 segundos, el merker **M0.2** conectado a la salida EQ se pone en 1.

Este último merker pilota la entrada U del contador **C6** que, de este modo, se incrementará cada 60 segundos. Por tanto, **C6** constituirá el contador de los minutos. El número de minutos contados se deposita en **MW11**.

El comparador **CP1** desarrolla una función análoga a **CP0**: lleva a 1 el merker **M0.4** cuando los minutos contados alcanzan el valor 60. Así pues, el contador **C7**, que tiene **M0.4** conectado a la entrada U, contará las horas de cierre del contacto y depositará su conteo en **MW12**.

Queda por analizar como los contadores de segundos y minutos se reponen a cero al impulso de entrada que hace 60. Observamos que la entrada de reset (R) de **C5** está conectada al merker **M0.3** y que éste vale 1 si **I0.6** o **M0.2** están en 1 (recorrido 7). Recordemos que este último merker es la salida del comparador de los segundos. Por tanto, el contador de segundos se repondrá a cero, bien correspondiendo con el cierre de la entrada **I0.6**, tal como se requiere en el trazado, o bien cuando el conteo de segundos llegue a 60.





Del mismo modo, observando la elaboración del merker **M0.5** en el recorrido 8, el contador de los minutos se pondrá a cero en correspondencia con el cierre de **I0.6** o cuando el conteo de los minutos alcanza el valor 60.

Por su parte, el contador de las horas solo se repone a cero tras el cierre de la entrada **I0.6**.

Así pues, para concluir, el cuentatiempo de software que hemos construido nos permite contar hasta 65535 horas, 59 minutos y 59 segundos (precisos!). Todo ello, admitiendo que se pueda considerar un valor tan exacto, en un período de tiempo tan largo, teniendo presentes los inevitables errores del reloj interno, tanto del PLC real como el del PC sobre el que 'gira' el PCL simulado.

Como ejercicio práctico, puede intentar modificar el programa añadiendo un contador de los días, que se incrementará en 1 cada 24 horas.

Para verificar el programa, sin tener que esperar tiempos muy largos, puede disminuir la constante de tiempo en la carga de **TR3**, aumentando así la frecuencia del tren de impulsos, o bien forzar manualmente valores de conteo próximos a los de comparación.

Ejemplo 21: generador de onda cuadrada

*La salida **Q0.7** debe estar controlada por una señal de onda cuadrada con $T_{off}=1.5$ seg. y $T_{on}=0.5$ seg.*

En la solución propuesta se utilizan dos temporizadores que se "rebotan" mutuamente la activación. El merker **M0.0**, inicialmente inactivo, pone en marcha el temporizador **TR1** durante un tiempo de 1.5 s. Al finalizar el tiempo, la salida **Q0.7** se activa desde **TR1**. Entonces, el contacto del recorrido 2 **Q0.7** inicia el temporizador **TR2** con un tiempo de 0,5 s durante el cual **Q0.7** s mantiene activado. Al acabar este otro tiempo, el merker **M0.0** se pone en 1 y al siguiente ciclo de ejecución, la entrada S de **TR1** se lleva al nivel 0, así como su salida. También la entrada S de **TR2** se lleva a nivel lógico bajo y, con él, su salida **M0.0**.

Hasta este momento, la salida **Q0.7** se ha mantenido inactiva durante 1,5 s y se ha activado durante 0,5 s: Ya hemos analizado un periodo completo de la señal. Pero, en ese instante, el sistema se ha llevado de

nuevo a sus condiciones iniciales, tal como muestra el diagrama de [Figura 29](#), y todo el conjunto vuelve a iniciarse de nuevo de forma cíclica, realizando el generador deseado.

Programando adecuadamente el valor de las dos constantes de word se puede variar T_{on} y T_{off} , realizando una onda cuadrada con frecuencias y *duty cycle* distintas.

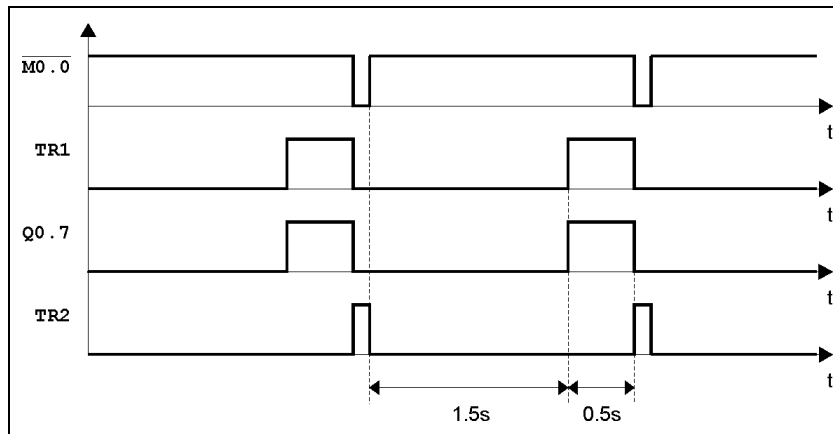
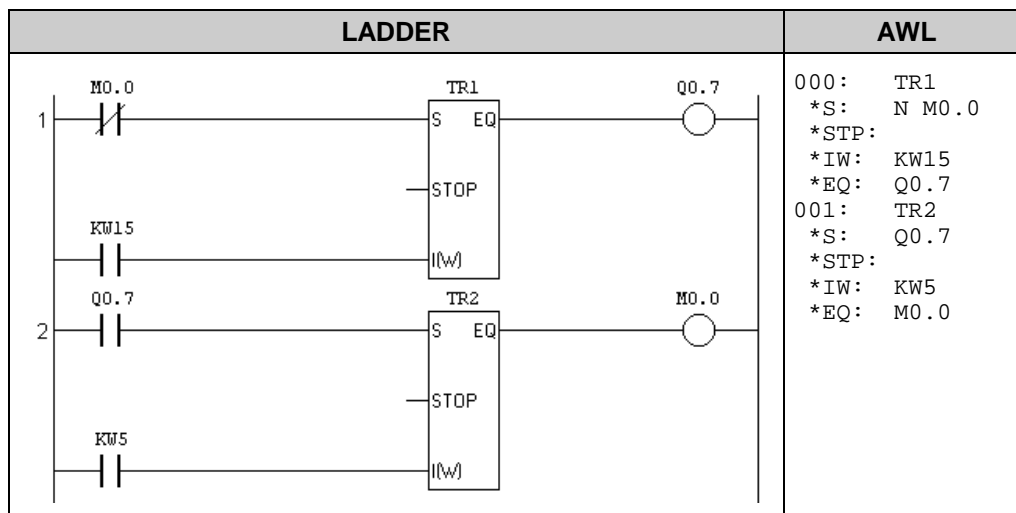


Figura 29: diagrama temporal generador de onda cuadrada



Ejemplo 22: otro generador de onda cuadrada

La salida $q0.7$ debe estar controlada por una señal de onda cuadrada con $T_{off}=1.5$ seg. y $T_{on}=0.5$ seg.

La solución presentada aquí es diferente a la propuesta en el ejemplo anterior, aunque con el mismo trazado. El temporizador $TR1$ se usa para generar un tren de impulsos con un periodo de dos segundos. En [Figura 30](#) se recoge, en la primera línea, el desarrollo de su salida y, en el segundo, el complemento del merker $M0.0$ conectado a ésta. Esta señal se aplica a la entrada de un temporizador con retardo a la activación, a cuya salida está conectada una bobina de $q0.7$. El desarrollo de $q0.7$ se muestra en la última línea del diagrama y representa la solución al problema.

También en este caso, programando de manera adecuada las dos constantes de tiempo, se puede variar la frecuencia y *duty cycle* de la onda cuadrada.

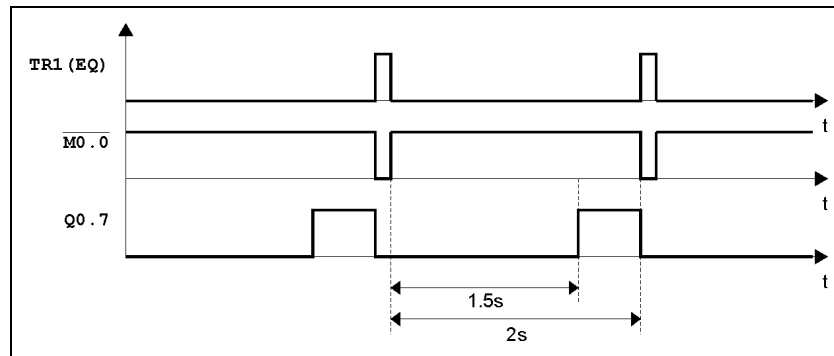
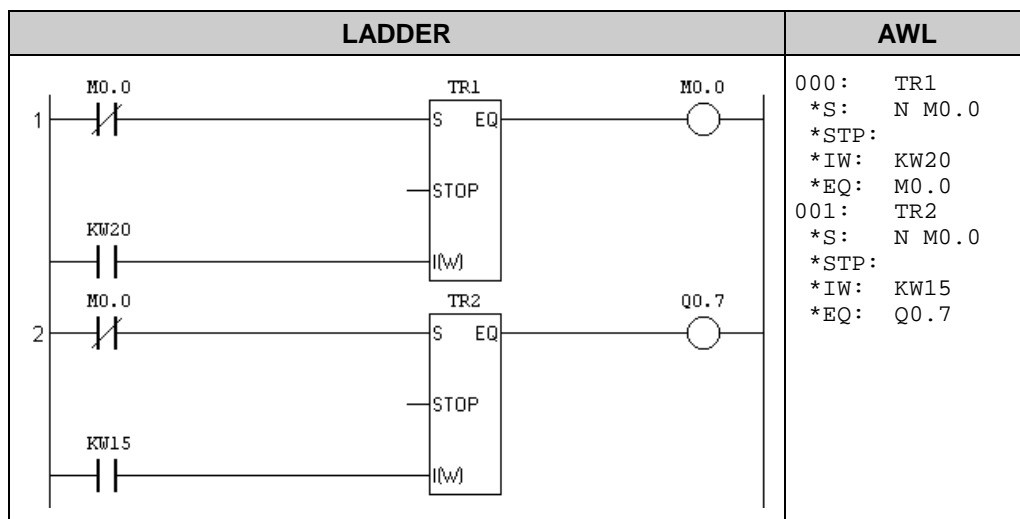


Figura 30: diagrama temporal de un generador de onda cuadrada



Ejemplo 23: control temporizado de luces

Un pulsador conectado a la entrada $I0.0$ activa durante tres minutos un grupo de luces conectadas a la salida $Q0.1$, y junto a éstas se activa un piloto luminoso, conectado a la salida $Q0.2$ que, 15 segundos antes de que las luces se apaguen, empieza a parpadear para avisar que el tiempo está a punto de agotarse. El piloto luminoso se apaga definitivamente a la vez que las luces.

Se utilizan dos temporizadores conectados para funcionar como retardo a la desactivación. El primero, cargado con un tiempo de 3 minutos ($KW1800$, es decir, 1800 décimas de segundo), controla directamente la salida del grupo de luces. El segundo, cargado con un tiempo inferior en 15 segundos ($KW1650$), activa un merker de apoyo.

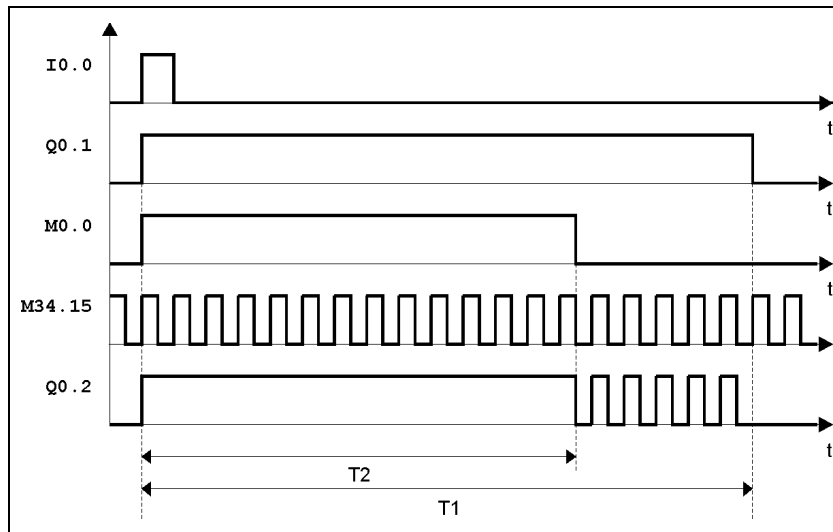


Figura 31: diagrama temporal del control de luces temporizado

Los recorridos temporales de $Q0.1$ y $M0.0$, a continuación del impulso sobre $I0.0$, se muestran en la segunda y tercera línea del diagrama de [Figura 31](#). La cuarta línea informa del desarrollo no a escala del merker $M34.15$ que actúa como multivibrador astable con un periodo de 2 segundos, controlado directamente por el Sistema Operativo del PLC. Este merker será útil para el parpadeo del piloto luminoso.

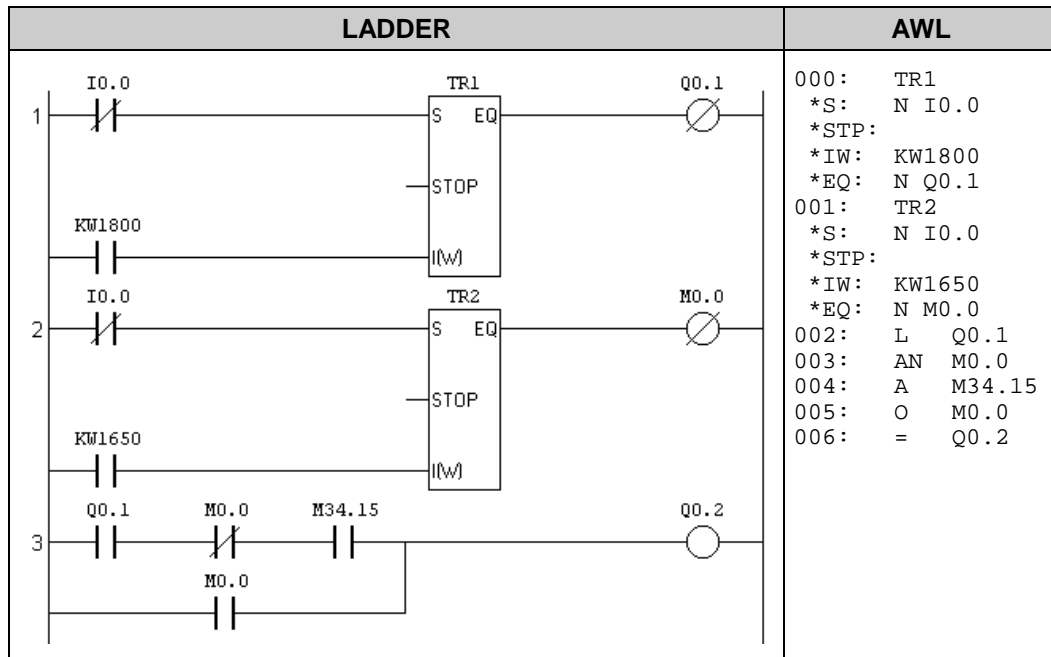
Observando la quinta línea del diagrama, se comprueba que la luz indicadora debe estar encendida, o bien cuando $M0.0$ está activo o, bien cuando están activos $Q0.1$ y $M34.15$ y, simultáneamente $M0.0$ está inactivo. Es decir, en términos de función booleana:

$$Q0.2 = M0.0 + (Q0.1 \cdot \overline{M0.0} \cdot M34.15)$$

Las soluciones propuestas implementan precisamente todo cuanto se ha descrito.

En la primera, realizada en ladder, se utilizan los dos primeros recorridos para la gestión de los temporizadores con retardo a la desactivación. Por su parte, el último recorrido está dedicado a la construcción de la lógica para el funcionamiento del piloto indicador.

En el programa AWL, las dos primeras instrucciones incorporan las llamadas a los módulos de sistema temporizadores, y las siguientes, implementan la lógica de funcionamiento del piloto indicador.



Ejemplo 24: divisor de frecuencia (x4)

Realizar un divisor de frecuencia por 4: a cada cuatro impulsos en la entrada se activa un impulso sobre la salida Q0.1.

La primera parte de las soluciones propuestas genera un tren de impulsos, tal como hemos aprendido a hacer en los ejemplos anteriores, mostrando la correspondiente señal sobre la salida Q0.0. Esto constituirá la señal a dividir.

La segunda parte implementa efectivamente el divisor. Este está constituido por un contador que va disminuyendo a cada impulso del generador (entrada D conectada a Q0.0) y que, cuando llega a cero, se autoprograma al valor 4 (salida Z llevada a la entrada S a través de Q0.1).

A cada 4 impulsos de Q0.0, para un solo ciclo de ejecución, el conteo se repone a cero. En este ciclo, la salida del contador se activa así como la salida Q0.1 del PLC conectada a ella.

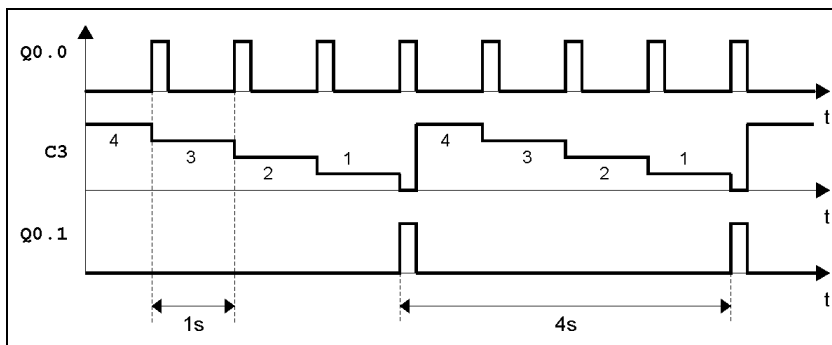
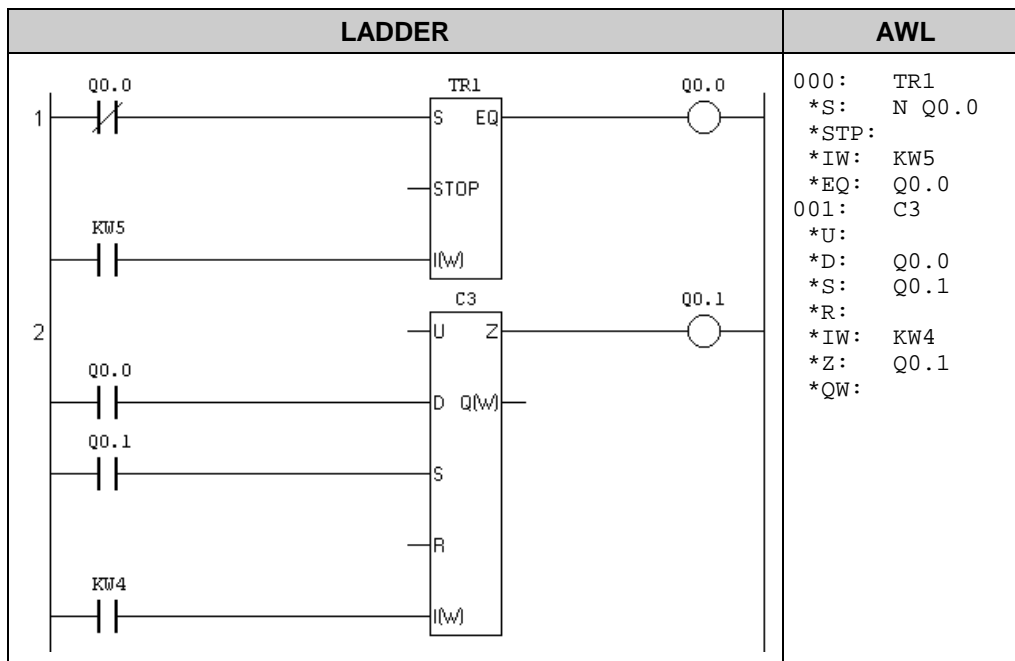


Figura 32: diagrama temporal del divisor de frecuencia por 4



Ejemplo 25: semáforo para Fórmula 1

Al activarse el pulsador conectado a la entrada **I0.0**, las cinco luces un semáforo deben encenderse una tras otra, a cada segundo. Un segundo después del encendido completo, las luces deberán apagarse.

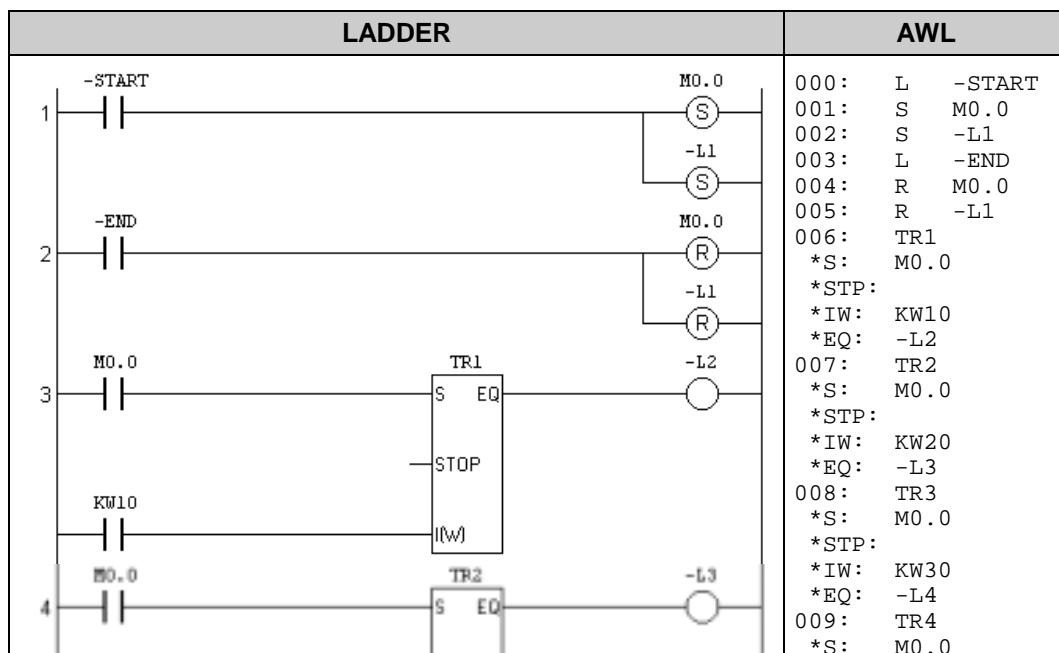
Para programar la solución de este problema se han usado símbolos. Su correspondencia con los operandos absolutos se ha establecido según la tabla siguiente.

Op. absoluto	Símbolo	Atributo
I0.0	START	Pulsador de puesta en marcha
Q0.1	L1	Luz 1
Q0.2	L2	Luz 2
Q0.3	L3	Luz 3
Q0.4	L4	Luz 4
Q0.5	L5	Luz 5

EL diagrama temporal de [Figura 33](#) muestra en las líneas de la 2 a la 6 el desarrollo de las salidas del PLC que controlan las luces del semáforo, en función de la entrada **START** que se muestra en la primer línea.

Al cierre de la entrada **START** se activa el merker **M0.0**, que se mantendrá activo incluso cuando se reabra la entrada, tal como se muestra en las líneas 1 y 2 del diagrama. Con este merker se controla la luz **L1**, que es la primera en encenderse, simultáneamente con la activación de la entrada **START**, y se inician cinco temporizadores con retardo a la activación, programados con tiempos progresivos de 1 a 5 segundos. Los 4 primeros tienen la función de controlar la salida de la luz correspondiente. La misión del último es restaurar el merker **M0.0** así como llevar a cero la entrada de todos los temporizadores y, como consecuencia, apagar las cinco luces conectadas a las correspondientes salidas.

Ponga en Run el PLC, trasforme el interruptor 0.1 en un pulsador, acciónelo y ...¡que gane el mejor!



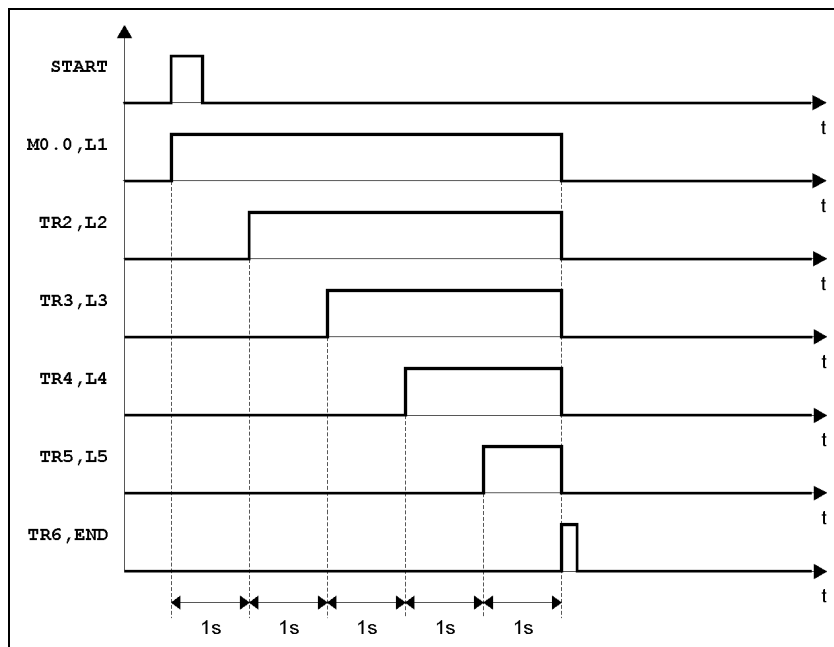
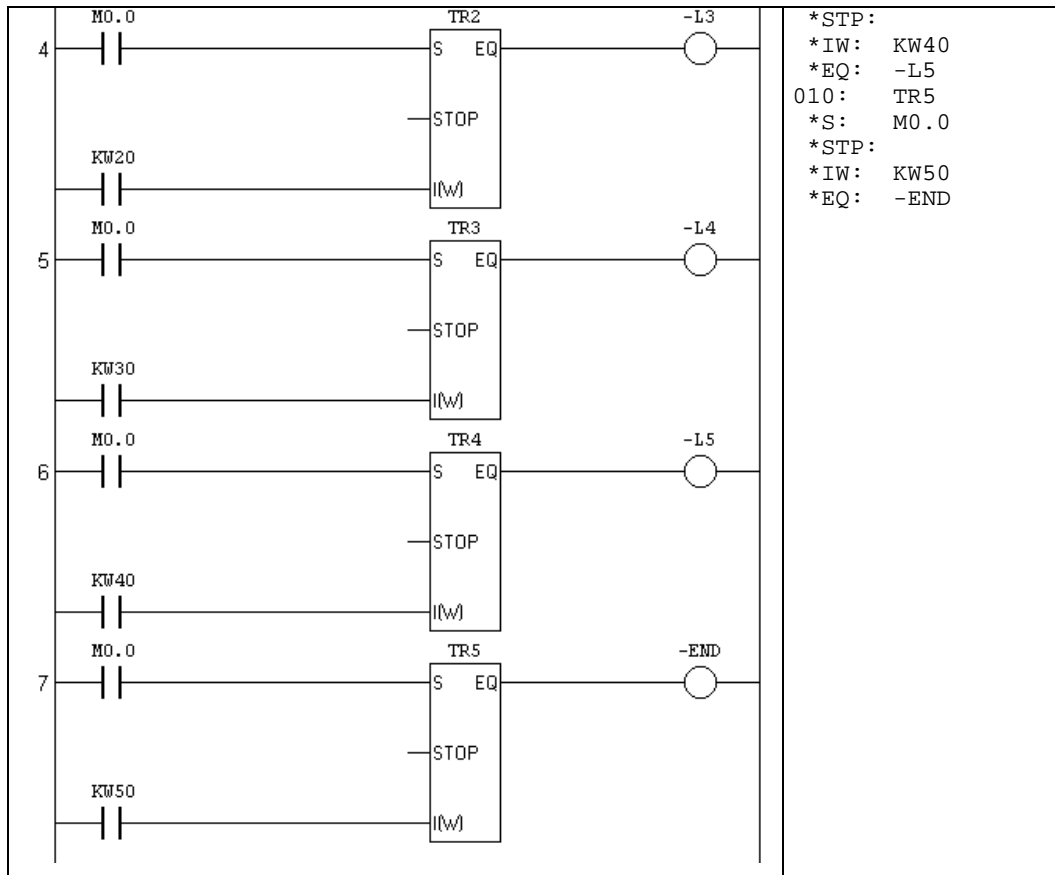


Figura 33: diagrama temporal para semáforo de Formula 1

Ejemplo 26: luces secuenciales con 4 canales

Construir un secuenciador para 4 canales que prevea el siguiente esquema de encendido.

P \ C	0	1	2	3	4	5
0	●	○	○	○	○	○
1	○	●	○	○	○	●
2	○	○	●	○	●	○
3	○	○	○	●	○	○

Figura 34: esquema de encendido para secuenciador para 4 canales

El esquema establece la secuencia de encendido de las luces conectadas a cuatro canales. Los círculos negros indican la activación del canal durante su paso específico. Así, durante el paso 0 será activo el canal 0, durante el paso 1 el canal 1, y así sucesivamente. Si las luces se colocan en línea, el efecto será el de un desplazamiento de la fuente luminosa desde la primera hasta la última posición y luego al contrario.

Al paso 6 le sucede un paso 7 idéntico al 0 y luego otro idéntico al paso 1, es decir, el diagrama se va recorriendo cíclicamente. Imaginen que lo recortan y lo enroscan formando un cilindro y hacen coincidir los límites opuestos del paso 0 y del 5, sería algo similar al tambor de un carillón: cuando acaba de tocar su musiquilla, empieza de nuevo.

Continuando con el símil sonoro, para que un carillón funcione necesita un cilindro con unas levas dispuestas de un modo adecuado sobre su superficie lateral y un mecanismo que lo haga girar.

Empezaremos por construir este último. El mecanismo de avance del secuenciador será un tren de impulsos con un período de 0,2 segundos, es decir, una base de tiempos con un período elegido de manera que se obtenga un desplazamiento de la fuente lumínica que se considera el adecuado. El temporizador **TIMER** se ocupa de esta función de la manera ya descrita en el ejemplo 16.

Por su parte, el cilindro estará formado por un contador que, partiendo del valor 6, va decreciendo a cada impulso. La salida de **TIMER** se envía, a través del merker **M10.0**, a la entrada D del contador **COUNTER**. El valor de conteo vigente se deposita en la merker word **MW9**. La salida del contador se reconduce a la entrada de set a través del merker **M10.1** y, a continuación, éste se autoprograma a 6 en cuanto llega a 0.

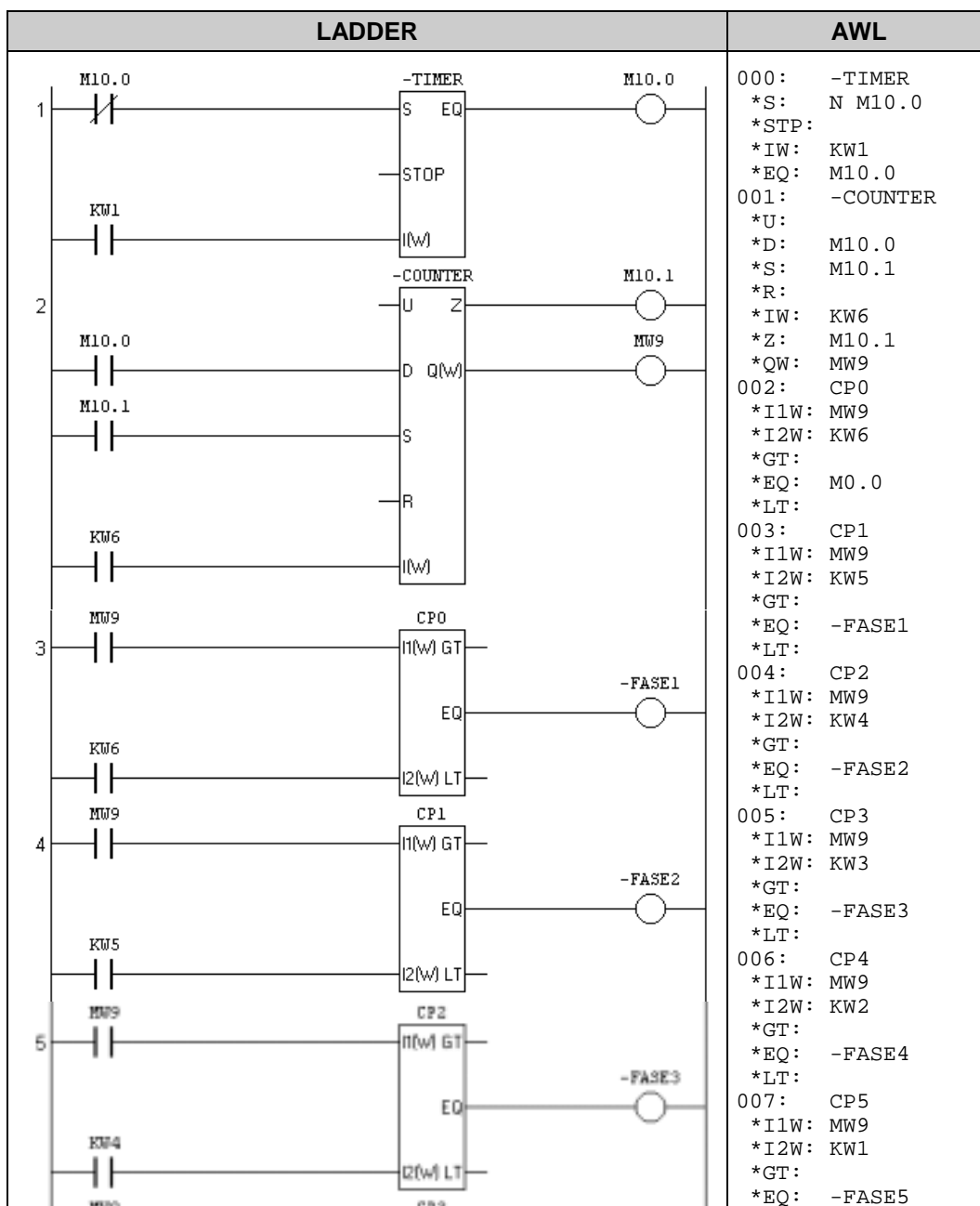
Antes de colocar las levas en el cilindro, identificamos los tramos del cilindro que corresponden a cada paso específico. El comparador **CP0** programa a 1 el merker **PASSO0** cuando el contador vale 6, identificando entre los posibles valores del contador el que corresponde a dicho paso. Los siguientes comparadores programan el merker correspondiente a cada uno de los demás pasos. Así, al final, a cada 0,2 s estará activo un merker distinto, yendo a 1 a continuación de **PASSO0** a **PASSO5** y, luego, empezando de nuevo, desde **PASSO0**.

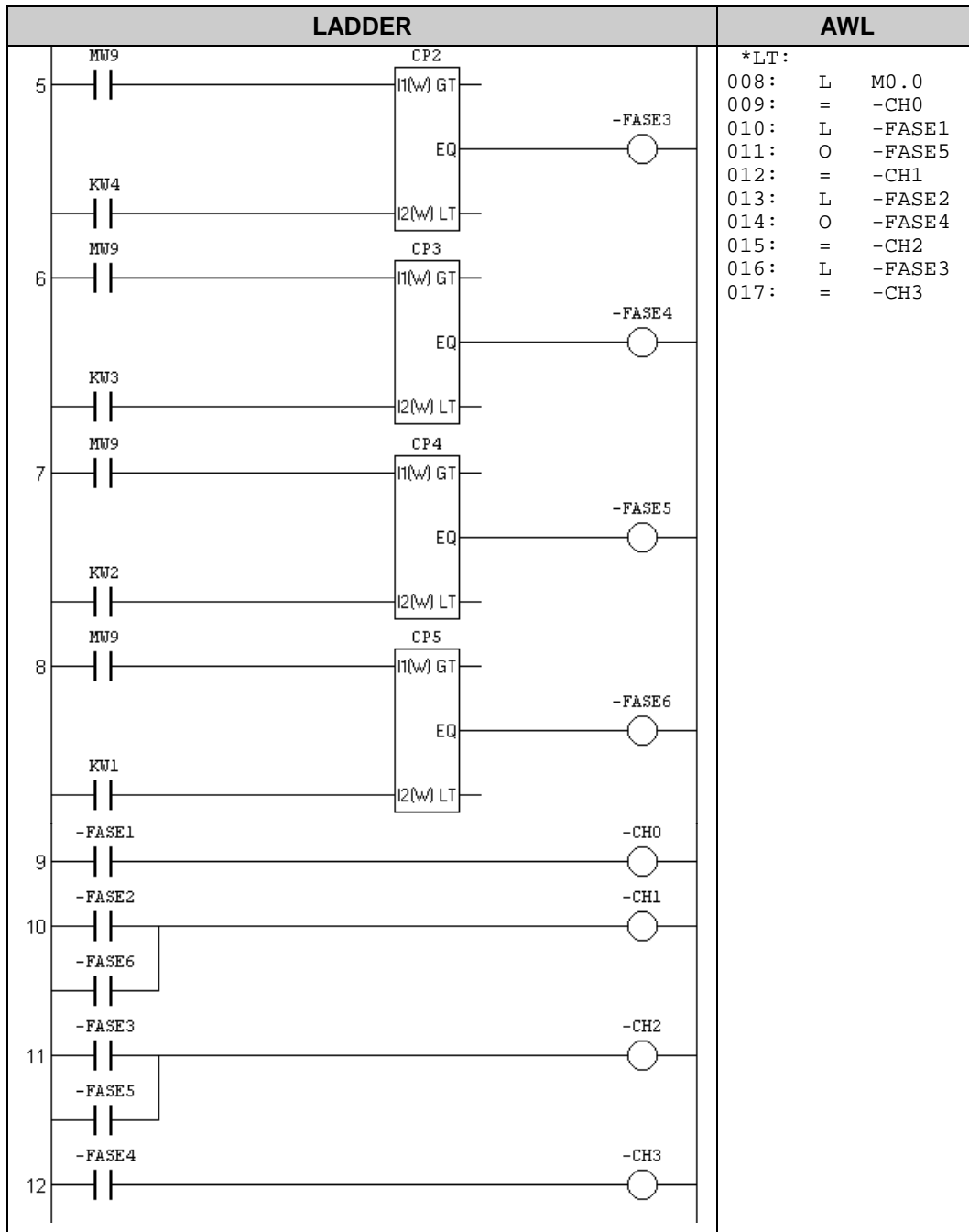
Ahora que hemos identificado las posiciones podemos insertar las levas. Empezamos por el canal 0, observamos nuevamente la parrilla de [Figura 34](#), el canal 0 es activo tan sólo durante el paso 0, por lo tanto **CH0=PASSO0**.

EL canal 1 debe estar activado tanto durante el paso 1 como durante el paso 5, es decir, **CH1=PASSO1+PASSO5**

Proseguimos así para los otros dos canales hasta terminar el carillón o, dejando ya a un lado el símil didáctico, el secuenciador.

En este ejemplo se puede aumentar o disminuir la duración de los pasos simplemente cambiando la constante con la que se carga el temporizador, produciendo el efecto de variar la velocidad del desplazamiento aparente de la fuente luminosa. Se puede modificar el número de pasos, cambiando la constante con la cual se carga el contador y añadiendo otros controles para discriminar los pasos añadidos. Por último, también se puede cambiar la secuencia de encendido de las luces, modificando las condiciones en los grupos de OR que constituyen la última parte del programa.





Ejemplo 27: conteo de entradas cerradas

Contar el número de las entradas cerradas entre los 8 primeros.

MB20.0 está destinada a contener el número de entradas cerradas que se ha contado. Dicho merker byte se pone en 0 al inicio del programa. Por su parte, **MB10.0** representa una máscara de 8 bit, de los que cada vez, solo uno estará a 1. El valor inicial es 1, es decir, $(00000001)_2$ donde sólo el bit 0 es cierto.

Las instrucciones 004 y 005 ejecutan la AND bit a bit de la máscara y del byte de entrada menos significativo del PLC. Al primer paso, con el valor de la máscara que acabamos de ver, la AND da un resultado distinto de 0 sólo si **IB0.0** está cerrado. A continuación, la derivación sobre cero prevista por la instrucción siguiente sólo se efectúa para la entrada abierta.

Si la entrada está cerrada, incrementamos el byte de conteo **MB20.0**, cargamos su valor en el registro general (**L MB20.0**), sumando 1 (**ADD KB1**) y transferimos el resultado de nuevo a **MB20.0** (**= MB20.0**).

En uno u otro caso, los dos recorridos de elaboración se reúnen en la etiqueta 010 donde, tras haber cargado la máscara en el registro general de byte (**L MB10.0**), se dispone el desplazamiento de una posición hacia la izquierda, lo cual equivale a una multiplicación por 2 (**MUL KB2**). El resultado del desplazamiento se retransfiere nuevamente a **MB10.0**. La máscara vale ahora 2, es decir, $(00000010)_2$, donde sólo el bit 1 es cierto y, siendo distinta de 0, la operación de desplazamiento sobre 0 (**BNZ 4**) se ejecuta y la elaboración continua desde la instrucción 004, examinando la entrada consecutiva.

Para cada una de las entradas que se han encontrado cerradas el merker byte **MB20.0** se aumenta en 1.

Después de 8 desplazamientos, el bit en estado alto, que poco a poco se ha ido trasladando en la máscara, sale por la izquierda y **MB10.0** pasa a 0. En esta condición, la última operación de derivación se ignora y el ciclo termina. En ese momento, **MB20.0** tiene un valor que representa el número de entradas que se han encontrado cerradas.

LADDER	AWL
	000: L KB0
	001: = MB20.0
	002: L KB1
	003: = MB10.0
	004: L MB10.0
	005: A IB0.0
	006: BZ 10
	007: L MB20.0
	008: ADD KB1
	009: = MB20.0
	010: L MB10.0
	011: MUL KB2
	012: = MB10.0
	013: BNZ 4

CAPÍTULO 14

NOTAS SOBRE LA PROGRAMACIÓN DEL PLC S5 100U

Premisa

Las notas facilitadas en esta sección se refieren únicamente al PLC simulado, es decir, a PC-Sim para S5 100U. Sin embargo, muy a menudo, y precisamente por el hecho de que en toda la simulación con AW-SYS se procura el mayor acercamiento posible a la realidad, también pueden ser útiles para el PLC real. Le aconsejamos, por ello, que utilice los manuales originales Siemens del PLC real y de los dispositivos de programación y obtener así todas las informaciones necesarias para la programación del dispositivo real.

Algunos de los temas tratados son, por su naturaleza, comunes a todos los PLC simulados. Sin embargo, se ha preferido repetir las mismas descripciones para todos ellos, con el fin de convertir a cada uno de estos capítulos en una unidad autónoma: un cómodo y indispensable manual para tener al alcance de la mano durante la programación del PLC simulado.

Generalidades sobre el PLC S5 100U

El S5 100U es un PLC modular y, en la simulación, está formado por la CPU y los siguientes módulos de entrada/salida:

Nº del módulo	Tipo
0	8 entradas digitales
1	8 entradas digitales
2	8 salidas digitales
3	8 salidas digitales
4	4 entradas analógicas (no visualizado en la Ventana PLC)
5	4 salidas analógicas (no visualizado en la Ventana PLC)

La CPU está dotada de un conmutador de encendido (On-Off) y de un conmutador del modo de funcionamiento (Stop-Run). Dos pilotos luminosos, STOP, de color rojo, y RUN, de color verde, señalan el estado en vigor de la CPU. El encendido del piloto rojo indica que la CPU recibe alimentación y se encuentra en el estado de Stop. El encendido del piloto verde indica que la CPU recibe alimentación y se encuentra en estado de Run. Existe también un piloto de color amarillo, cuyo encendido señala la falta de batería.

Cada entrada y cada salida digital dispone de su piloto indicador de estado: Piloto encendido para el estado activo y piloto apagado para el estado inactivo.

Direccionamiento

Bajo el nombre de direccionamiento se engloban la serie de normas sintácticas que deben observarse al referirse a variables o constantes durante la escritura de un programa usado por el PLC. El direccionamiento en la simulación del PLC S5 100U se efectúa de acuerdo con las reglas expuestas a continuación y que se resumen en la [Tabla 13](#).

Entradas–Salidas digitales

La sigla para el direccionamiento de las entradas y de las salidas digitales en este modelo de PLC está compuesta por:

- letra que identifica las entradas (E) o las salidas (A)
- número de módulo
- punto de separación
- número de canal

Los módulos se numeran correlativamente: el módulo más cercano a la CPU lleva el número 0, el siguiente, el número 1 y así sucesivamente.

Los módulos digitales utilizados en la simulación son cuatro y llevan números identificativos comprendidos entre 0 y 3. Tanto los de entrada como los de salida son del tipo de 8 canales. Los canales disponibles en cada módulo llevan un número identificativo comprendido entre 0 y 7.

Así, por ejemplo:

E1.4 identifica el canal de entrada 4 del módulo periférico 1;

A2.3 identifica el canal de salida 3 del módulo periférico 2.

El PLC simulado está compuesto, además de por la CPU, por cuatro módulos digitales periféricos: por este orden, dos de entrada y dos de salida. Así pues, las direcciones para los canales de estos módulos son:

- de **E0.0** a **E1.7** para las entradas digitales
- de **A2.0** a **A3.7** para las salidas digitales.

A continuación veremos que esta modalidad de direccionamiento no hace referencia directa a las entradas y a las salidas físicas del PLC, sino a la respectiva imagen que el sistema operativo construye en el interior de la memoria de la CPU. En realidad, las imágenes de la periferia se han reproducido por completo en la CPU simulada. De este modo, en el interior de su programa, podrá dirigir, sin generar ningún error, toda la imagen de las entradas de **E0.0** a **E127.7** y toda la imagen de las salidas, de **A0.0** a **A127.7**. El direccionamiento de variables en las imágenes de proceso, no correspondientes a módulos de I/O, conduce sencillamente a pruebas o programaciones de bit, byte o word de memoria, sin llevar a la activación de salidas o a la valoración del estado de las entradas.

Entradas y salidas (que siempre deben entenderse como imagen de proceso de las entradas e imagen de proceso de las salidas) también son direccionables en byte y en word.

EB0 representa el byte de entrada del módulo 0; **EB1** representa el byte de entrada del módulo 1 y así sucesivamente hasta **EB127**. Con la limitación obvia de que sólo los byte a los cuales corresponden efectivamente los módulos de entrada asumen un significado práctico.

Por su parte, **AB2** y **AB3** representan los byte de salida para los módulos digitales presentes en el PLC simulado aunque también en esta ocasión, el campo admitido para este tipo de operando va de **AB0** a **AB127**.

EW0 representa la word de entrada 0, compuesta por los bytes **EB0** y **EB1**. El campo direccionable en este PLC llega hasta **EW126**.

AW2 representa la palabra de salida 2, compuesta por los bytes **AB2** y **AB3**. El campo direccionable en este PLC va de **AW0** a **AW126**.

Entradas-Salidas analógicas

El módulo 4 posee cuatro canales analógicos de entrada que se representan en la imagen de las entradas en correspondencia con las word **EW96**, **EW98**, **EW100** y **EW102**. La codificación del valor utiliza 12 de los 16 bit disponibles. Tal como se indica en la figura, el bit menos significativo del código (B0) ocupa el bit 3 de la word, B1 ocupa el bit 4 y así sucesivamente hasta B11 que ocupa el bit 14. Los bit 15, 2, 1 y 0 siempre están a cero en la simulación.

Bit de EW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Código	0	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	0	0	0

Al valor mínimo de la entrada (p. ej. 0V) le corresponde un código B11-B0 igual a 0 y al valor máximo le corresponde un código de 2048. Así pues, como conclusión, el valor de una de las word de la imagen puede variar de 0 a $2048 \times 8 = 16384$. La multiplicación por 8 se deriva del desplazamiento de 3 posiciones a la izquierda ($2^3 = 8$) que prevé el código.

El módulo 5 posee cuatro canales analógicos de salida que se representan en la imagen de las salidas en correspondencia con las word **AW104**, **AW106**, **AW108** y **AW110**. La codificación del valor utiliza 11 de los 16 bit disponibles. Tal como se indica en el esquema siguiente, el bit menos significativo del código (B0) ocupa el bit 4 de la word, B1 ocupa el bit 5 y así sucesivamente hasta B10 que ocupa el bit 14. Los bit 15, 3, 2, 1 y 0 no tienen influencia en la finalidad de la emisión de un cierto valor de tensión.

Bit de AW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Código	X	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	X	X	X	X

Para obtener el valor mínimo en la salida (p. ej. 0V) es necesaria una codificación B10-B0 igual a 0. Por el contrario, el valor máximo se obtiene con una codificación de 1024. Así pues, como conclusión, el valor que puede escribirse en una de las word de la imagen puede variar de 0 a $1024 \times 16 = 16384$. La multiplicación por 16 deriva del desplazamiento 4 posiciones a la izquierda ($2^4 = 16$) que prevé el código.

Merker

Los merker o flag constituyen una parte de la memoria RAM de la CPU. Puede accederse directamente a ella desde el programa del usuario y éste la utiliza para memorizaciones intermedias de resultados de cálculos o temporales de datos. En este PLC, está compuesta por 128 byte (merker byte). Cada byte de esta memoria es direccionable a través de la sigla **MB_x**, donde **x** es el número progresivo de byte. Por ejemplo, **MB100** identifica el byte 100 de la memoria merker. Así, las direcciones posibles van de **MB0** a **MB127**.

Cada bit de la misma memoria es direccionable individualmente con la sigla **M_x.y**, donde **x** sigue siendo el número de byte e **y** es el número del bit en el interior del byte. Así pues, son posibles las direcciones de **M0.0** a **M127.7**.

Por último, la memoria merker puede direccionarse por words utilizando la sigla **MW_x** donde **x** es el número de word. La **MW0** (merker word número 0) está compuesta por los merker byte **MB0** y **MB1**, la **MW1** de **MB1** y **MB2**, y así sucesivamente. El campo de operandos posibles va de **MW0** a **MW126**. De hecho, la **MW126**, que reagrupa **MB126** y **MB127**, es la última merker word disponible, siendo **MB127** el último byte direccionable.

Los merker byte de **MB0** a **MB63** son retentivos, es decir, conservan su valor después de una falta de alimentación y obviamente, también lo son los merker bit que contienen y las merker word que componen.

Los merker byte entre **MB64** y **MB127** son no retentivos, de manera que pierden su valor después de una falta de alimentación en la CPU, es decir, se reponen a cero. Lo mismo puede decirse para los merker bit que aquellos contienen y las merker word que componen.

Temporizadores

En el interior del programa del usuario pueden utilizarse 64 temporizadores con direcciones entre **T0** y **T63**. Todos los temporizadores son no retentivos.

Contadores

El programador dispone de 64 contadores, direccionables con las siglas que van de **C0** a **C63**. Los 8 primeros (de **C0** a **C7**) son retentivos y los restantes (de **C8** a **C63**) son no retentivos.

Constantes

En el programa existe la posibilidad de utilizar constantes para asignar valores a merker word, para cargar temporizadores y contadores, para operaciones de comparación, etc. Las constantes definidas para este PLC son de distintos tipos para que puedan adaptarse a las distintas exigencias que se plantean en la fase de programación. A continuación, le facilitamos una relación.

Constantes de byte

Son constantes constituidas por un solo byte. Su valor va de 0 a 255 y la sigla correspondiente, de **KB0** a **KB255**. Por ejemplo, **KB100** es una constante de byte con un valor de (100)₁₀.

Constantes de coma fija

Son constantes expresadas con 16 bits incluido el signo. Los números negativos están representados en complemento a dos. Las siglas para estas constantes van de **KF -32768** a **KF +32767**. Por ejemplo, la constante **KF -1200** identifica una constante de 16 bits con valor $(-1200)_{10}$.

Constantes hexadecimales

Son constantes expresadas con 16 bits en formato hexadecimal de **KH 0000** a **KH FFFF**. Por ejemplo, **KH 0120** identifica una constante que vale $(120)_{16}$ o, si lo prefieren, $(288)_{10}$.

Tabla 14: resumen del direccionamiento de variables de PC-Sim para Siemens S5 100U

Tipo	Min	Max
Entradas de bit	E0.0	E127.7
Entradas de byte	EB0	EB127
Entradas de word	EW0	EW126
Salidas de bit	A0.0	A127.7
Salidas de byte	AB0	AB127
Salidas de word	AW0	AW126
Merker bit (retentivos)	M0.0	
Merker bit (no retentivos)	M64.0	M127.7
Merker byte (retentivos)	MB0	MB63
Merker byte (no retentivos)	MB64	MB127
Merker word (retentivas)	MW0	MW62
Merker word (no retentivas)	MW64	MW126
Data byte	DL0	DR255
Data word	DW0	DW255
Temporizadores	T0	T63
Contadores (retentivos)	Z0	Z7
Contadores (no retentivos)	Z8	Z63
Constante de byte	KB0	KB255
Constante de coma fija	KF -32768	KF +32767
Constante hexadecimal	KH 0000	KH FFFF
Constante binaria	KM 00000000 00000000	KM 11111111 11111111
Constante de dos byte	KY 0,0	KY 255,255
Constante de tiempo	KT 0.1	KT 999.3
Constante de conteo	KZ 0	KZ 999
Bloques organizativos	OB 0	OB 63
Bloques de programa	PB 0	PB 63
Bloques funcionales	FB 0	FB 63
Bloques datos	DB 2	DB 63

Constantes binarias

Son constantes expresadas con 16 bit con descripción directa binaria. Su representación va de **KM 00000000 00000000** a **KM 11111111 11111111**. Por ejemplo, **KM 00100000 11100111** identifica una constante del valor $(0010000011100111)_2$ o, en base hexadecimal, $(20E7)_{16}$.

Constantes a dos byte

Expresan constantes de 16 bit como combinación de los valores de los dos bytes. La sigla tiene como límites **KY 0,0** y **KY 255,255**. Por ejemplo, la constante **KY 25,100** identifica una constante de 16 bits cuyo byte superior vale $(25)_{10}$ y el inferior $(100)_{10}$.

Constantes de tiempo

Expresan constantes para la carga de los temporizadores. Están compuestas por la sigla **KT** seguida de un número que identifica los intervalos de tiempo, de un punto de separación y de otro número que identifica la base de tiempos. Los valores admisibles tienen por límites **KT 0.0** y **KT 999.3**. Para más detalles, ver más adelante [Operaciones de temporización](#).

Constantes de conteo

Expresan constantes de 16 bit en formato BCD para la programación de los contadores. Valores de **KZ 0** a **KZ 999**. Ver más adelante [Operaciones de conteo](#).

Elaboración del programa del usuario

Sistema operativo

El corazón del funcionamiento del PLC es su Sistema Operativo: un programa que gestiona el funcionamiento de base del dispositivo, grabado en la ROM por la casa constructora.

Un examen minucioso de las funciones del Sistema Operativo no forma parte de los objetivos de este manual. También en los manuales de los PLC reales es difícil encontrar aclaraciones detalladas al respecto. Por otra parte, la ejecución de este programa es completamente transparente para el programador que, en algunos aspectos, puede incluso ignorar su existencia.

Sin embargo, para completar la información, relacionamos algunas de las funciones que el sistema desarrolla, aquellas que consideramos más importantes para los fines recogidos en los siguientes apartados:

- control del tiempo de ciclo (*watch-dog*)
- test del hardware y de la memoria
- llamada del programa del usuario
- actualización de la imagen de las entradas
- transferencia de la imagen de las salidas
- gestión toma de programación

- control de la batería tampón.

Estado de la CPU

La CPU puede encontrarse en uno de los siguientes estados:

OFF: el PLC no recibe alimentación, el conmutador de encendido en la *Barra de instrumentos* de PC-Sim se encuentra en la posición de Off. Los pilotos luminosos de Power y Run situados en el frontal del PLC están apagados.

STOP: el PLC recibe alimentación pero el programa del usuario no se ejecuta, el conmutador de encendido en la *Barra de instrumentos* de PC-Sim se encuentra en la posición de On y el conmutador de funcionamiento se encuentra en la posición de Stop. El piloto luminoso de Power está encendido y el piloto luminoso Run está apagado.

RUN: la CPU recibe alimentación y se ejecuta el programa del usuario, el conmutador de encendido en la *Barra de instrumentos* de PC-Sim se encuentra en la posición de On y el conmutador de funcionamiento se encuentra en la posición de Run. El piloto indicador de Stop está apagado, y el de Run está encendido.

El estado en vigor de la CPU se visualiza en la barra del título de la *Ventana PLC*. encerrado entre corchetes.

Imágenes del proceso

Una parte de la RAM de la CPU está destinada a servir de memoria intermedia para el acceso a los periféricos. Esta memoria toma el nombre de imagen de proceso y se divide en dos partes: imagen de proceso de las entradas e imagen de proceso de las salidas (a partir de ahora, se denominarán sólo imagen de las entradas e imagen de las salidas).

Antes de la elaboración del programa del usuario, todas las entradas del PLC se leen y su estado se copia en la imagen de las entradas. Al final de la elaboración del programa del usuario, la memoria imagen de las salidas se recopia en las salidas del PLC.

Durante la elaboración del programa, cada acceso a las entradas o a las salidas no está referido al canal físico efectivo, sino a un bit, el correspondiente a aquel canal, en la memoria imagen de proceso. Cuando se requiere la lectura de una entrada, el que en realidad se lee es un bit de memoria que, con anterioridad ha sido copiado por la entrada efectiva. Cuando se escribe sobre una salida, en realidad no se escribe directamente sobre el canal de salida, sino que se escribe un bit de memoria que a continuación se recopiará sobre el canal físico de salida. Veremos como también se puede escribir una entrada y leer una salida, pero esto no cambia los términos de la cuestión.

Por lo tanto, el microprocesador situado en el interior del PLC deberá dividir su tiempo entre diferentes funciones: ejecutar el Sistema operativo, actualizar la imagen de las entradas, ejecutar el programa de la aplicación y transferir la imagen de las salidas. El diagrama circular de la *Figura 16* ilustra estas actividades. El sentido del recorrido del diagrama es el indicado por la flecha. Los porcentajes respectivos de tiempo son absolutamente indicativos y, en particular, el dedicado a la ejecución del programa del usuario varía con las

dimensiones del mismo, es decir, en general, cuanto más largo es el programa, más tiempo necesitará la CPU para ejecutarlo (excluyendo de esta consideración, los programas que contienen operaciones de salto).

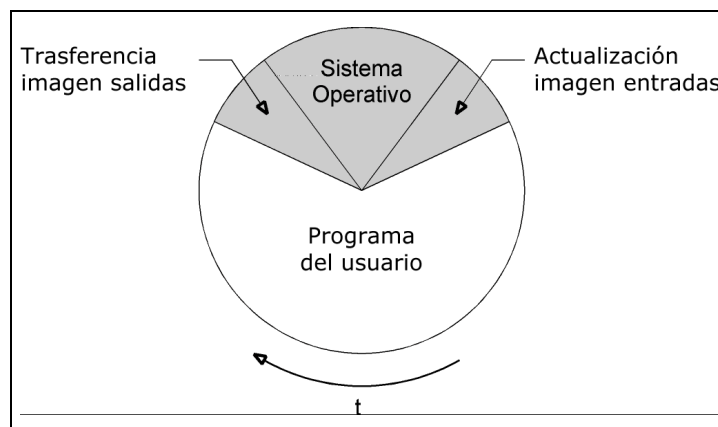


Figura 35: reparto del tiempo del microprocesador del PLC

Control del tiempo del ciclo

El sistema operativo, justo antes de efectuar la llamada al programa del usuario, inicia un monoestable que, por lo general, tiene una duración que va de unas decenas a poquísimas centésimas de milisegundo. Supongamos que el programa del usuario requiera, para su ejecución, un tiempo inferior al del monoestable. En esta situación, antes de cada una de las series de llamadas, el monoestable se iniciará nuevamente y, como resultado, la salida de este último se mantendría continuamente alta.

En cambio, si el programa del usuario requiriera para su ejecución, más tiempo que el que dura el monoestable, la salida de este se situaría en estado bajo, provocando el paso inmediato del PLC a Stop.

Un programa mal realizado podría llevar a la CPU a ejecutar indefinidamente, o durante un tiempo excesivo para los fines de rapidez en el control de la instalación, un ciclo cerrado de instrucciones, sin volver a la elaboración del sistema operativo. Esta condición podría provocar daños a la instalación y un grave peligro para el personal que esté trabajando. El control del tiempo del ciclo, que en la literatura técnica especializada podrá encontrar también bajo el nombre de *watch-dog*, evita precisamente este problema.

PC-Sim también controla que su programa no genere un ciclo cerrado pero, en este caso, las múltiples y cambiantes tareas de un PC, imponen tiempos más largos para la detección del problema, del orden de algunos segundos.

Programación estructurada

El programa para este PLC puede redactarse dividiendo su contenido en varias partes. Cada una de estas partes toma el nombre de bloque. Existen bloques organizativos (OB), bloques de programa (PB), bloques funcionales (FB) y bloques de datos (DB). Los distintos bloques pueden llamarse el uno al otro, utilizando las operaciones de llamada o de apertura que se tratarán en el apartado dedicado a la descripción de las operaciones. Un programa escrito según este criterio se denomina programa estructurado.

La operación de división del programa en bloques, es decir, su estructuración, no es indispensable: el programa puede redactarse tranquilamente en un único bloque y funcionar perfectamente. Esto es lo que haremos al inicio de nuestro ejercicio práctico con el PLC. En el capítulo siguiente se incluyen una serie de programas, la mayoría de los cuales, los más sencillos, están escritos en un bloque único. Pero, cuando los programas se hacen largos, su división puede tener distintas ventajas:

- Agrupación de partes idénticas de código en un bloque y llamada de éste desde donde sea necesario. Es similar en todo a la escritura de subprogramas en lenguaje de programación desarrollados para PC.
- Transportabilidad de bloques con funciones específicas de un proyecto al otro.
- Mayor facilidad de lectura del programa y, por tanto, más sencillez en el posterior mantenimiento.

En esta sección se describirán los distintos tipos de bloques, sus peculiaridades y sus funciones. La [Tabla 15](#) resume las características principales de los distintos tipos de bloque.

Tabla 15: resumen de tipos de bloques

	OB	PB	FB	DB
Número	0...63	0...63	0...63	2...63
Longitud max				256 word
Set operaciones	Fundamentales	Fundamentales	Fundamentales e integrativas	
Programación	Ladder y AWL	Ladder y AWL	Solo AWL	

Bloques Organizativos (OB)

Los bloques organizativos representan la interface entre el sistema operativo y el programa del usuario. Es decir, son bloques que el usuario puede programar pero que normalmente se llaman desde el sistema operativo de forma cíclica o después de la verificación de unas condiciones específicas. Pueden programarse en ladder o en AWL con el set de instrucciones básicas. A continuación, relacionamos los bloques organizativos que el sistema operativo del PLC simulado llama y las condiciones para las que se verifica la citada llamada.

OB1 Elaboración cíclica del programa

El sistema llama periódicamente al bloque organizativo OB1, inmediatamente después de la actualización de la imagen de proceso de las entradas. Cada programa deberá contener este bloque y este podrá contener, eventualmente, llamadas de otros bloques. Si pretende escribir todo el programa en un único bloque y si el set de instrucciones fundamentales le es suficiente, precisamente es el OB1 el que deberá programar.

OB21 Inserción manual (Stop → Run)

Cuando se inserta el selector del modo de funcionamiento Stop-Run (paso a Run) se borran la imagen de proceso, los temporizadores, los contadores no retentivos y los merker no retentivos. A continuación, se elabora, si está programado, el bloque OB21. Después se ejecutará el programa cíclico normal (llamada al OB1). Durante la elaboración del OB21, el piloto rojo de Stop y el verde de Run están encendidos a la vez.

Este bloque puede ser utilizado para las programaciones iniciales que eventuales merker deban asumir cuando se reinicie la CPU.

OB22 Retorno de la tensión (Off → On)

Cuando se inserta el selector de alimentación Off-On (paso a On), si el selector del modo de funcionamiento se encuentra ya en Run, se borran la imagen de proceso, los temporizadores, los contadores no retentivos y los merker no retentivos. A continuación, se elabora, si está programado, el bloque OB22. Después se ejecutará el programa cíclico normal (llamada al OB1). Durante la elaboración del OB22, el piloto rojo de Stop y el verde de Run están encendidos a la vez.

Este bloque puede ser utilizado para las programaciones iniciales que eventuales merker deban asumir, después de un corte de tensión.

OB34 Batería fuera de servicio

La CPU del PLC simulado controla continuamente el estado de la batería. Cuando se extrae la batería (en sentido estricto no se puede simular que la batería se haya descargado) el OB34 se elabora antes de cada ciclo, si está programado. En éste es posible escribir la reacción que el programa del usuario debe tener en caso de que la batería esté fuera de servicio.

El OB34 se continuará elaborando antes de cada ciclo, hasta que vuelva a colocarse la batería tampón.

La función de la batería en el PLC real es la de mantener el programa en RAM cuando la CPU no recibe alimentación y de mantener, en esas mismas condiciones, el valor de los merker y de los contadores retentivos. En la simulación se ha repropuesto un comportamiento análogo ya que se ha considerado importante desde un punto didáctico.

Bloques de Programa (PB)

Son bloques en los que es posible escribir el programa utilizando el lenguaje ladder o el AWL con el set de instrucciones fundamentales. Estos bloques representan partes completadas del programa. Con el fin de que se ejecuten deberán ser llamados por otros bloques con operaciones SPA o SPB.

Bloques Funcionales (FB)

Son bloques en los que se puede escribir el programa utilizando solo el lenguaje AWL. Representan partes completas de programa y en ellas es programable todo el set de operaciones (fundamentales e integrativas). Para que sean ejecutados deberán ser llamados por otros bloques con operaciones SPA o SPB.

Tabla 16: especie y tipo de parámetros para bloque funcional

Especie	Tipo	Operandos vigentes
E, A	BI (operandos de bit)	E
		A
		M
	BY (operandos de byte)	EB
AB		

		MB
		DR
		DL
	W (operandos de word)	EW
		AW
		MW
		DW
D	KB	
	KF	
	KH	
	KM	
	KT	
	KY	
	KZ	
B	No está admitida la indicación del tipo	OB
		PB
		FB (no parametrizados)
		DB
T	No está admitida la indicación del tipo	T
Z	No está admitida la indicación del tipo	Z

Además, estos bloques son parametrizables, es decir, pueden pasarse parámetros desde el bloque que hace la llamada. Para las instrucciones que las usan, los parámetros serán sustituidos en el bloque llamado por los operandos formales. El máximo número de parámetros por bloque es de 40. Cada parámetro debe ser declarado al inicio del bloque en la correspondiente parrilla del editor AWL especificando la sigla, la especie y, si es necesario, el tipo. Los parámetros en las operaciones SPA FB o SPB FB deben corresponder en tipo y número a los esperados. Para más detalles, ver [El editor AWL para el PLC S5 100U](#) en el [Capítulo 3](#).

La [Tabla 16](#) resume los posibles parámetros y los operandos en vigor correspondientes.

Bloques de Datos (DB)

En estos bloques pueden almacenarse datos que servirán para la elaboración del programa del usuario. Los bloques de datos están estructurados en words, a partir de la número 0 hasta la número 255, como máximo. El acceso a los datos podrá efectuarse con operaciones de lectura y escritura de word (DW) o de byte (izquierdo DL, derecho DR). En la fase de edición del bloque pueden memorizarse todos los tipos de constantes de 16 bit.

Antes de acceder a los datos contenidos en un bloque, es necesaria su apertura mediante la instrucción A DBn, donde n es el número del bloque. Las operaciones de acceso a datos con bloques no abiertos, o el acceso a datos más allá de la longitud del bloque, provocan un error de ejecución y llevan la CPU a Stop.

Programación en AWL

La modalidad de programación AWL, o Lista de instrucciones, constituye una de las dos metodologías de creación de programas que puede utilizar este PLC. Esta modalidad consiste en la escritura consecutiva de instrucciones. Cada una de ellas es la unidad más pequeña de programa y constituye una función elemental para la CPU en la elaboración del programa del usuario.

Instrucciones

Cada instrucción AWL está compuesta por una operación y, donde sea necesario, por un operando. De estos últimos ya hemos hablado en el apartado dedicado al direccionamiento. En los siguientes apartados facilitamos una descripción detallada de las operaciones, agrupadas por tipología. Además, se facilitan, para mayor comodidad en la consulta por parte del programador, dos tablas que resumen las operaciones en función del set al que pertenecen. Las operaciones del set fundamental pueden programarse en cualquier bloque, con la exclusión obvia de los DB que, por su propia naturaleza, no soportan instrucciones sino datos. Por el contrario, las operaciones integrativas pueden programarse sólo en los bloques funcionales.

Registros

Para la ejecución del programa del usuario, la CPU utiliza distintos registros con el propósito de memorizar resultados intermedios de cálculo y/o valorar entonces si ejecuta o no ciertas operaciones. A continuación les proporcionamos el listado y la descripción de estos registros.

RLC

Es un registro de 1 bit y se utiliza como memoria de trabajo en la elaboración del programa del usuario. En éste se depositan los resultados de las operaciones lógicas o de las operaciones de test sobre el estado. De él se extraen los valores para las operaciones de memorización.

Acumuladores

Son dos registros de 16 bits que se identifican con las siglas ACCU1 y ACCU2. En ACCU1 se depositan los valores leídos con operaciones de carga, obtenidos de operaciones lógicas sobre word, aritméticas y de conversión y de él se extrae el valor a recopiar en las operaciones de transferencia y de configuración de temporizadores y contadores. ACCU2 sirve como registro auxiliar para ACCU1 en las operaciones lógicas sobre word, de comparación y aritméticas.

Indicadores

Por último, la CPU utiliza tres registros de un bit (indicadores):

- ANZ 0
- ANZ 1
- OV

Estos están influenciados por operaciones de comparación, cálculo, desplazamiento y conversión y los verifican las operaciones de salto.

Ver la [Tabla 20](#) para más informaciones sobre la influencia de las operaciones sobre los indicadores ANZ 0 y ANZ 1.

El bit de overflow (OV) se pone a nivel alto si en una operación de cálculo, el resultado es mayor que +32767 o menor que -32768.

Operaciones lógicas

U (And o test sobre el estado 1)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la AND entre el contenido de RLC y el estado del operando especificado, y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir si la instrucción precedente es RLC limitante, el estado del operando se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para conocer que operaciones son RLC limitantes.

Operandos admitidos: E, A, M, T, Z.

O (Or o test sobre el estado 1)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la OR entre el contenido de RLC y el estado del operando especificado, y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir, si la instrucción precedente es RLC limitante, el estado del operando se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para conocer que operaciones son RLC limitantes.

Operandos admitidos: E, A, M, T, Z.

UN (And Not o test sobre el estado 0)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la AND entre el contenido de RLC y el complemento del estado del operando especificado, y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir, si la instrucción precedente es RLC limitante, el complemento del estado del operando se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para conocer que operaciones son RLC limitantes.

Operandos admitidos: E, A, M, T, Z.

ON (Or Not o test sobre el estado 0)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la OR entre el contenido de RLC y el complemento del estado del operando especificado, y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir, si la instrucción precedente es RLC limitante, el complemento del estado del operando se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para conocer que operaciones son RLC limitantes.

Operandos admitidos: E, A, M, T, Z.

O (Or de AND)

El resultado lógico combinatorio RLC actual se deja a un lado para ponerse en OR con el RLC en vigor a la primera operación RLC limitante. El resultado se deposita en RLC.

Operandos admitidos: ninguno

U((Apertura de paréntesis en combinación And)

El RLC en vigor se deja a un lado para retomarse tras el cierre del paréntesis y se pone en AND con el RLC que resulta del cálculo de la expresión incluida dentro del paréntesis.

Operandos admitidos: ninguno

O((Apertura de paréntesis en combinación Or)

El RLC en vigor se deja a un lado para retomarse tras el cierre del paréntesis y se pone en OR con el RLC que resulta del cálculo de la expresión incluida dentro del paréntesis.

Operandos admitidos: ninguno.

) (Paréntesis de cierre)

Indica el final de una expresión entre paréntesis. El RLC que se dejó de lado con una operación de apertura de paréntesis se recupera y se pone en AND o en OR con el RLC en vigor, según si la operación que ha iniciado la expresión entre paréntesis es una U(o una O(. El resultado se deposita en RLC.

Operandos admitidos: ninguno

Operaciones de memorización**S (Set)**

Si RLC=1, el operando se pone a 1. Si RLC=0, el operando no se modifica. La operación limita el RLC, es decir, constituye una operación de cierre de secuencia y la siguiente operación lógica U, O, UN o ON se interpretará como un test sobre el estado del operando.

Operandos admitidos: E, A, M

R (Reset)

Si RLC=1, el operando se pone a 0. Si RLC=0 el operando no se modifica. La operación limita el RLC, es decir, constituye una operación de cierre de secuencia y la siguiente operación lógica U, O, UN o ON será interpretada como un test sobre el estado del operando.

Operandos admitidos: E, A, M

= (Asignación)

El valor de RLC se copia en el operando. La operación limita el RLC, es decir, constituye una operación de cierre de secuencia y la siguiente operación lógica U, O, UN o ON se interpretará como un test sobre el estado del operando.

Operandos admitidos: E, A, M

Operaciones de carga y transferencia**L (Carga)**

Con las operaciones de carga se puede recopiar en el registro ACCU1 el valor de operandos de byte y de word. Antes de cada carga, el valor de ACCU1 se transfiere a ACCU2.

Para operaciones de carga de byte, el valor del operando se recopia en el byte menos significativo de ACCU1, mientras que el byte más significativo se repone a cero.

Operandos admitidos: EB, EW, AB, AW, MB, MW, DR, DL, DW, T, Z, KM, KH, KF, KY, KB, KT, KZ

La [Figura 36](#) ilustra las modificaciones que se producen en los acumuladores tras dos operaciones consecutivas de carga. La primera línea muestra el estado precedente a la primera operación de carga en la que se ha asumido que los valores de los acumuladores eran cero. La segunda y la tercera línea muestran, respectivamente, el contenido de ACCU1 y ACCU2 tras la operación de carga sobre la misma línea. Al final de la ejecución de las dos instrucciones, el primer valor cargado se encuentra en ACCU2 y el segundo en ACCU1.

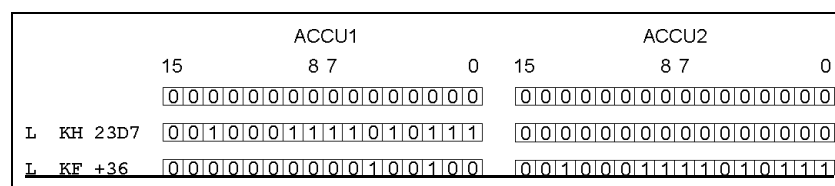


Figura 36: ilustración operaciones de carga

LC (Carga BCD)

Con esta operación se carga en el registro ACCU1 el valor del operando codificado en BCD. Antes de la carga, el valor de ACCU1 se transfiere a ACCU2.

Operandos admitidos: T, Z

T (Transferencia)

Con las operaciones de transferencia, se puede copiar en operandos de byte y de word, el valor del registro ACCU1. En caso de operando de byte, tan sólo se copia el byte menos significativo de ACCU1.

Operandos admitidos: EB, EW, AB, AW, MB, MW, DR, DL, DW

Operaciones de temporización

Antes de la descripción de las operaciones de temporización, hagamos algunas aclaraciones referentes a la codificación de los tiempos.

Cuando se pone en marcha un temporizador con una de las operaciones expuestas más adelante en esta sección, el valor en ACCU1 se utilizará para la programación del tiempo tal como se muestra en el siguiente esquema.

Bit de ACCU1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Codificación			B1	B0	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

Los bits 13 y 12 determinan la base de tiempos; los que van de 0 a 11, en codificación BCD, determinan el número de intervalos en la citada base. Los bit 14 y 15 no son significativos. La base se determina del modo especificado en la tabla siguiente:

B1	B0	Valor	Base
0	0	0	0.01 s
0	1	1	0.1
1	0	2	1s
1	1	3	10s

En realidad, las cosas son más sencillas de lo que pudieran hacer pensar estas advertencias, y es posible cargar un valor de tiempo codificado directamente con la operación de carga de constante de tiempo (L KT).

Una constante de tiempo está formada como sigue:

- Sigla **KT**
- Número que expresa los intervalos de tiempo (de 0 a 999)
- Carácter punto (.)
- Número que expresa la base de tiempos (de 0 a 3)

Así, la constante **KT 250.1** expresa un tiempo de 25 segundos (250x0.1s) y, viceversa, si deseamos programar un tiempo de 10 segundos podremos elegir entre las constantes **KT 100.1**, **KT 10.2** y **KT 1.3**. En realidad, ya que el error máximo que se comete en una temporización es igual a un intervalo de la base de tiempos, la representación más conveniente resulta ser la primera, aquella con la base más pequeña.

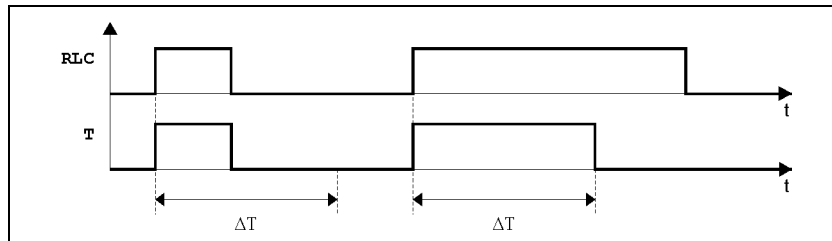
El tiempo representable más pequeño es **KT1.0** (1 centésima de segundo); el mayor es **KT999.3** (9990 segundos).

SI (Impulso)

La temporización se activa en el flanco ascendente de RLC con el tiempo especificado y codificado en ACCU1 y continua transcurriendo mientras RLC=1. Si RLC pasa a ser 0, el tiempo se programa sobre 0.

El estado del temporizador es alto durante todo el transcurso del tiempo.

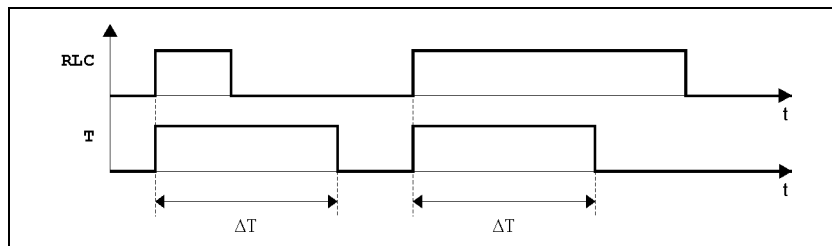
Operandos admitidos: T

**SV (Impulso prolongado)**

La temporización se activa en el flanco ascendente de RLC con el tiempo especificado y codificado en ACCU1. El estado de RLC no influye a posteriori la temporización.

El estado del temporizador es alto durante todo el transcurso del tiempo.

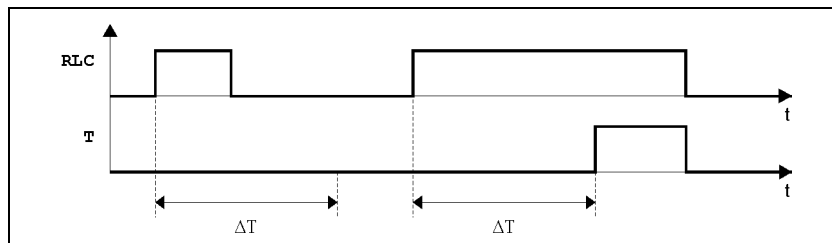
Operandos admitidos: T

**SE (Retardo a la activación)**

La temporización se activa en correspondencia con el flanco ascendente de RLC, con el tiempo especificado y codificado en ACCU1. Si RLC pasa a ser 0, el tiempo se programa sobre 0.

El estado del temporizador es alto cuando el tiempo ha transcurrido y RLC se encuentra aun en estado alto.

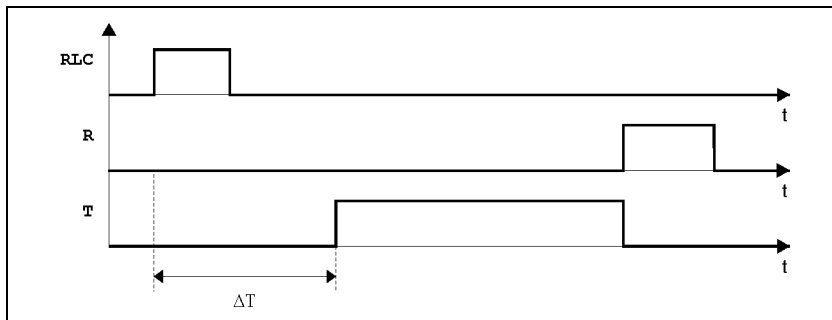
Operandos admitidos: T

**SS (Retardo a la activación con memoria)**

La temporización se activa en correspondencia con el flanco ascendente de RLC, con el tiempo especificado y codificado en ACCU1. Si RLC pasa a ser 0, el tiempo no recibe ninguna influencia.

El estado del temporizador es alto cuando el tiempo ha transcurrido. Solo regresa al estado bajo después de un reset del temporizador con operación R.

Operandos admitidos: T

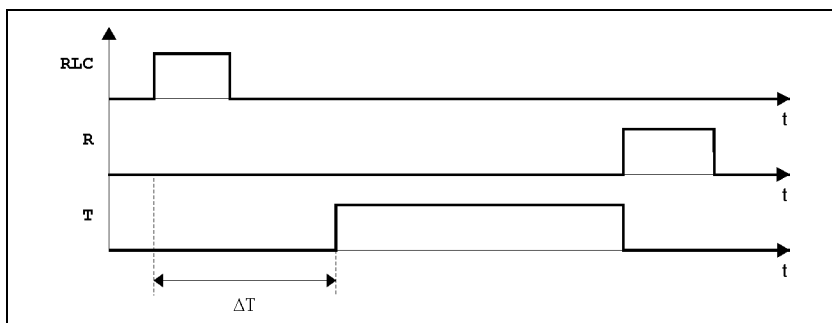


SA (Retardo a la desactivación)

La temporización se activa en correspondencia con el flanco descendiente de RLC con el tiempo especificado y codificado en ACCU1. Si RLC pasa a ser 1, el tiempo se reprograma al valor inicial.

El estado del temporizador es alto mientras RLC = 1 o bien mientras el tiempo está transcurriendo.

Operandos admitidos: T



R (Reset de un temporizador)

Si RLC=1, el tiempo se reprograma al valor inicial. Si RLC=0, la operación se ignora.

El estado del temporizador es bajo hasta que el tiempo se restablece. (RLC=0).

Operandos admitidos: T

Operaciones de conteo

Antes de la descripción de las operaciones, conviene hacer algunas aclaraciones en referencia a la codificación de los valores de programación de los contadores.

Cuando se programa un contador, es necesario especificar el valor preset en ACCU1, tal como se muestra en el siguiente esquema.

Bit de ACCU1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Codificación	X	X	X	X	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

Los bits de 0 a 11, en la codificación BCD, representan el valor inicial del conteo. Los bit de 12 a 15 no son significativos.

S (Set de un contador)

El conteo se pone en el valor en ACCU1 en correspondencia con el flanco ascendente de RLC.

El estado del contador va alto si el valor de conteo programado es distinto de 0.

Operandos admitidos: Z

R (Reset de un contador)

Mientras RLC = 1, el valor de conteo se pone en 0 y el estado del contador va bajo.

Operandos admitidos: Z

ZV (Conteo hacia delante)

El valor de conteo se incrementa en 1 en correspondencia con el flanco ascendente de RLC.

El estado del contador es o se mantiene alto.

Operandos admitidos: Z

ZR (Conteo hacia atrás)

El valor de conteo disminuye en 1 en correspondencia con el flanco ascendente de RLC.

El estado del contador va bajo si el valor de conteo pasa a ser 0.

Operandos admitidos: Z

Operaciones de comparación

!=F (Comparación de igualdad)

Si ACCU2 es igual a ACCU1, el RLC se pone en 1. De lo contrario, se pone en 0.

Operandos admitidos: ninguno

><F (Comparación de desigualdad)

Si ACCU2 es distinto de ACCU1, el RLC se pone en 1. De lo contrario, se pone en 0.

Operandos admitidos: ninguno

>F (Comparación de superioridad)

Si ACCU2 es mayor que ACCU1, el RLC se pone en 1. De lo contrario, se pone en 0.

Operandos admitidos: ninguno

>=F (Comparación de superioridad o igualdad)

Si ACCU2 es mayor o igual que ACCU1, el RLC se pone en 1. De lo contrario, se pone en 0.

Operandos admitidos: ninguno

<F (Comparación de inferioridad)

Si ACCU2 es menor que ACCU1, el RLC se pone en 1. De lo contrario, se pone en 0.

Operandos admitidos: ninguno

<=F (Comparación de inferioridad o igualdad)

Si ACCU2 es menor o igual que ACCU1, el RLC se pone en 1. De lo contrario, se pone en 0.

Operandos admitidos: ninguno

Operaciones aritméticas

+F (Adición)

Los contenidos de ACCU1 y ACCU2 se suman como números de 16 bit con signo. El resultado se memoriza en ACCU1.

Operandos admitidos: ninguno

-F (Sustracción)

El contenido de ACCU1 se resta de ACCU2. Los valores se interpretan como números de 16 bit con signo. El resultado se memoriza en ACCU1.

Operandos admitidos: ninguno

ADD (Adición de una constante)

La constante especificada como operando, de byte (BF) o de word (KF), se suma a ACCU1. El resultado se memoriza en ACCU1.

Operandos admitidos: BF, KF

Operaciones sobre bloques

SPA (Llamada absoluta a bloque)

Con independencia de RLC, la elaboración del programa se hace proseguir por la primera instrucción del bloque especificado como operando. Tras la elaboración de una instrucción BE, BEB o BEA en este último, la elaboración regresa al bloque que llama en correspondencia con la instrucción siguiente.

En caso de llamada de bloque funcional parametrizado se debe especificar, entre paréntesis y separados por el carácter ‘;’ (punto y coma), los operandos que hay que pasar al bloque llamado.

Operandos admitidos: OB, PB, FB

SPB (Llamada condicionada a bloque)

Si RLC=1, la elaboración del programa se hace proseguir por la primera instrucción del bloque especificado como operando. Tras la elaboración de una instrucción BE, BEB o BEA en este último, la elaboración regresa al bloque que llama en correspondencia con la instrucción siguiente.

Si RLC=0, la operación se ignora, RLC se pone en 1 y la elaboración continua con la instrucción posterior.

En caso de llamada de bloque funcional parametrizado se debe especificar, entre paréntesis y separados por el carácter ';' (punto y coma), los operandos que hay que pasar al bloque llamado.

Operandos admitidos: OB, PB, FB

A (Apertura de un bloque de datos)

Con independencia de RLC, se programa el bloque de datos en vigor, es decir el bloque al que harán referencia todas las operaciones de lectura y escritura de datos, ya sean de byte o de word. El bloque abierto se mantiene activo hasta la apertura de un nuevo bloque. En caso de una posterior llamada a bloque con operaciones SPA o SPB, el bloque de datos que ya esté abierto, se mantiene activo en el bloque llamado.

Cada acceso a datos, sin que previa y cautelarmente se haya abierto un bloque de este tipo, genera error.

Operandos admitidos: DB

BE (Final de bloque)

Es la instrucción de cierre, es decir, la última de un bloque. Independientemente de RLC, se termina la elaboración del bloque en uso con el consiguiente retorno al bloque que ha llamado. El valor de RLC, no modificado, retorna (se devuelve) al bloque que ha llamado. En el caso de la OB1, después de la elaboración de esta instrucción, el control regresa al Sistema Operativo.

Operandos admitidos: ninguno

BEA (Final de bloque incondicionado)

Independientemente de RLC, termina la elaboración del bloque en uso con el consiguiente retorno al bloque que ha llamado. El valor de RLC, no modificado, se retorna al bloque que ha llamado. En el caso de la OB1, después de la elaboración de esta instrucción, el control regresa al Sistema Operativo. La operación BEA, a diferencia de la anterior, también puede utilizarse en el interior de un bloque.

Operandos admitidos: ninguno

BEB (Final de bloque condicionado)

Si RLC=1, termina la elaboración del bloque en uso con el consiguiente retorno al bloque que ha llamado. El valor de RLC (1 en este caso), no modificado, se retorna al bloque que realiza la llamada. En el caso de la OB1, después de la elaboración de esta instrucción, el control regresa al Sistema Operativo.

Si RLC=0, la operación no se ejecuta, RLC se pone en 1 y la elaboración continua con la instrucción siguiente.

Operandos admitidos: ninguno

Operaciones lógicas sobre palabras

UW (And sobre word)

Ejecuta la combinación lógica AND bit a bit de ACCU1 y ACCU2. El resultado se deposita en ACCU1.

Operandos admitidos: ninguno

OW (Or sobre word)

Ejecuta la combinación lógica OR bit a bit de ACCU1 y ACCU2. El resultado se deposita en ACCU1.

Operandos admitidos: ninguno

XOW (Xor sobre word)

Ejecuta la combinación lógica XOR bit a bit de ACCU1 y ACCU2. El resultado se deposita en ACCU1.

Operandos admitidos: ninguno

Operaciones de desplazamiento

SLW (Desplazamiento a la izquierda)

Desplazamiento de los bits de ACCU1 a la izquierda, un número de posiciones expresadas por el parámetro. Los bits que se salen por la izquierda se pierden, mientras que por la derecha entran ceros. En caso de operando igual a 0, no se realiza ningún desplazamiento y la instrucción se ignora.

Operandos admitidos: número de 0 a 15

SRW (Desplazamiento a la derecha)

Desplazamiento de los bit de ACCU1 a la derecha, un número de posiciones expresadas por el parámetro. Los bits que salen por la derecha se pierden mientras que por la izquierda entran ceros. En caso de operando igual a 0, no se realiza ningún desplazamiento y la instrucción se ignora.

Operandos admitidos: número de 0 a 15

Operaciones de conversión

KEW (Complemento a 1)

ACCU1 se transforma en su complemento a 1, es decir, cada bit se invierte.

Operandos admitidos: ninguno

KZW (Complemento a 2)

ACCU1 se transforma en su complemento a 2, es decir, en primer lugar, cada bit se invierte y, a continuación, se le suma +1.

Operandos admitidos: ninguno

Operaciones de incremento/decremento

D (Decremento)

El contenido del byte menos significativo de ACCU1 disminuye en un valor igual al del operando. El byte más significativo no sufre cambios, y se omite incluso un posible préstamo.

Operandos admitidos: número de 0 a 255

I (Incremento)

El contenido del byte menos significativo de ACCU1 se aumenta en un valor igual al del operando. El byte más significativo no sufre cambios y se omite incluso una posible suma.

Operandos admitidos: número de 0 a 255

Operaciones de elaboración

B (Elaboración de palabra de datos o merker)

Con esta operación pueden realizarse indexaciones de muchas de las otras operaciones. El contenido del operando de la instrucción se utiliza para obtener el operando de la instrucción siguiente. Las operaciones indexables son las de carga, transferencia, llamada de bloques, lógicas, de memorización, de temporización y de desplazamiento.

En caso de indexación de operaciones binarias, la dirección de bit deberá estar presente en el byte más significativo y la dirección de byte en el menos significativo.

Operandos admitidos: MW, DW

Por ejemplo, la pareja de instrucciones

```
B    MW100
L    MB0
```

carga en el acumulador, por efecto de la segunda, el merker byte cuyo número está contenido en el operando de la primera. Si el contenido de **MW100** fuera $(36)_{10}$, entonces, el efecto de las dos instrucciones

combinadas sería **L MB36**, es decir, la carga en el acumulador del contenido de **MB36**. Si, al siguiente ciclo de ejecución, se incrementa en 1 el contenido de **MW100**

```

L   MW100
I   1
T   MW100

```

la pareja de instrucciones precedentes equivaldrá ahora a **L MB37**.

Otro ejemplo: supongamos que ya esté abierto un bloque de datos y la pareja de instrucciones

```

B   DW4
U   E0.0

```

ejecuta una AND con RLC del bit de entrada especificado en la data word. Si DW4 contiene $(0701)_{16}$ entonces la entrada direccionada será el canal 7 del módulo 1. Es decir, la pareja de instrucciones tendrá el mismo efecto que **U E1.7**.

En realidad la combinación entre el contenido del primer operando y la parte numérica del segundo es de tipo OR bit a bit, y solo en los casos en los que este último es nulo, es decir, igual a 0 ó 0.0, la combinación OR da un resultado exactamente igual al contenido del operando de la primera instrucción. Si esto no sucede, la indexación continua funcionando, pero hace falta mucha cautela para valorar plenamente sus efectos.

B = (Elaboración de bloque)

Llama al bloque transferido como parámetro. Si el bloque es OB, PB o FB, la elaboración continúa desde la primera instrucción de éste. Si es DB, el bloque datos se abre. Los FB sólo pueden llamarse si en ellos no se han definido parámetros.

Operandos admitidos: formal (Especie parámetro: B). Ver [Tabla 16](#).

Operaciones de salto

SPA = (Salto incondicional)

La elaboración prosigue en la instrucción identificada por la etiqueta especificada como operando.

Operandos admitidos: etiqueta

SPB = (Salto condicionado)

Si RLC=1, la elaboración prosigue en la instrucción identificada por la etiqueta especificada como operando.

Si RLC=0, el salto se ignora, RLC se pone en 1 y la elaboración continua desde la instrucción siguiente.

Operandos admitidos: etiqueta

SPZ = (Salto sobre 0)

El salto se ejecuta cuando ANZ1=0 y ANZ0=0. RLC no se modifica. Ver [Tabla 20](#).

Operandos admitidos: etiqueta

SPN = (Salto sobre no 0)

El salto se ejecuta cuando ANZ1 es distinto de ANZ0. RLC no se modifica. Ver [Tabla 20](#).

Operandos admitidos: etiqueta

SPP = (Salto sobre positivo)

El salto se ejecuta cuando ANZ1=1 y ANZ0=0. RLC no se modifica. Ver [Tabla 20](#).

Operandos admitidos: etiqueta

SPM = (Salto sobre negativo)

El salto se ejecuta cuando ANZ1=1 y ANZ0=1. RLC no se modifica. Ver [Tabla 20](#).

Operandos admitidos: etiqueta

SPO = (Salto sobre overflow)

El salto se ejecuta cuando se verifica un overflow (desbordamiento), bit OV=1. RLC no se modifica.

Operandos admitidos: etiqueta

Operaciones lógicas sobre operando formal**U = (And o test sobre el estado 1)**

Si la instrucción no es la primera de una secuencia, es decir si la instrucción precedente no es RLC limitante, ejecuta la AND entre el contenido de RLC y el estado del operando formal especificado, y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir, si la instrucción precedente es RLC limitante, el estado del operando formal se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para conocer qué operaciones son RLC limitantes.

Operandos admitidos: formal (Especie parámetro: E, A, T, Z; Tipo parámetro: BI donde esté admitido). Ver [Tabla 16](#).

O = (Or o test sobre el estado 1)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la OR entre el contenido de RLC y el estado del operando formal especificado y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir, si la instrucción precedente es RLC limitante, el estado del operando formal se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para saber qué operaciones son RLC limitantes.

Operandos admitidos: formal (Especie parámetro: E, A, T, Z; Tipo parámetro: BI donde esté admitido). Ver [Tabla 16](#).

UN = (And Not o test sobre el estado 0)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la AND entre el contenido de RLC y el complemento del estado del operando formal especificado y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir si la instrucción precedente es RLC limitante, el complemento del estado del operando formal se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para saber qué operaciones son RLC limitantes.

Operandos admitidos: formal (Especie parámetro: E, A, T, Z; Tipo parámetro: BI donde esté admitido). Ver [Tabla 16](#).

ON = (Or Not o test sobre el estado 0)

Si la instrucción no es la primera de una secuencia, es decir, si la instrucción precedente no es RLC limitante, ejecuta la OR entre el contenido de RLC y el complemento del estado del operando formal especificado, y pone el resultado en RLC.

Si la instrucción es la primera de una secuencia, es decir si la instrucción precedente es RLC limitante, el complemento del estado del operando formal se copia en RLC.

Consulte la [Tabla 18](#) y la [Tabla 19](#) para saber qué operaciones son RLC limitantes.

Operandos admitidos: formal (Especie parámetro: E, A, T, Z; Tipo parámetro: BI donde esté admitido). Ver [Tabla 16](#).

Operaciones de memorización sobre operando formal**S = (Set)**

Si RLC = 1, el operando formal se pone en 1. Si RLC=0, el operando formal no se modifica. La operación limita RLC, es decir, constituye una operación de cierre de secuencia y la siguiente operación lógica será interpretada como un test sobre el estado del operando.

Operandos admitidos: formal (Especie parámetro: E, A; Tipo parámetro: BI). Ver [Tabla 16](#).

RB = (Reset)

Si RLC = 1, el operando formal se pone en 0. Si RLC=0, el operando formal no se modifica. La operación limita RLC, es decir, constituye una operación de cierre de secuencia y la siguiente operación lógica será interpretada como un test sobre el estado del operando.

Operandos admitidos: formal (Especie parámetro: E, A; Tipo parámetro: BI). Ver [Tabla 16](#).

== (Asignación)

El valor de RLC se copia en el operando formal. La operación limita RLC, es decir, constituye una operación de cierre de secuencia y la siguiente operación lógica será interpretada como un test sobre el estado del operando.

Operandos admitidos: formal (Especie parámetro: E, A; Tipo parámetro: BI) . Ver [Tabla 16](#).

Operaciones de carga y de transferencia de operando formal**L = (Carga)**

Con las operaciones de carga se puede recopiar en el registro ACCU1 el valor de operandos formales de byte y de word. Antes de cada carga, el valor de ACCU1 se transfiere a ACCU2.

Para operaciones de carga de byte, el valor del operando se recopia en el byte menos significativo de ACCU1, mientras que el byte más significativo se repone a cero.

Operandos admitidos: formal (Especie parámetro: E, A, T, Z; Tipo parámetro: BY, W donde esté admitido). Ver [Tabla 16](#).

LC = (Carga BCD)

Con esta operación se carga, en el registro ACCU1, el valor del operando formal con codificación BCD. Antes de la carga, el valor de ACCU1 se transfiere a ACCU2.

Operandos admitidos: formal (Especie parámetro: T, Z). Ver [Tabla 16](#).

LW = (Carga de constante)

Se carga en el registro ACCU1 el valor del operando formal constante. Antes de la carga, el valor de ACCU1 se transfiere a ACCU2.

Operandos admitidos: formal (Especie parámetro: D; Tipo parámetro: KF, KH, KM, KY, KT, KZ). Ver [Tabla 16](#).

T = (Transferencia)

Con las operaciones de transferencia se puede recopiar en operandos formales de byte y de word, el valor del registro ACCU1. En caso de operando de byte, tan sólo se recopia el byte menos significativo de ACCU1.

Operandos admitidos: formal (Especie parámetro: E, A Tipo parámetro: BY, W). Ver [Tabla 16](#).

Operaciones de temporización y de conteo con operando formal**SI = (Impulso)**

El temporizador especificado como operando formal se inicia en correspondencia con el flanco ascendente de RLC, con el tiempo especificado en ACCU1. El tiempo va transcurriendo mientras RLC=1. Si RLC pasa a 0, el valor del tiempo se programa sobre 0.

El estado del temporizador es alto durante todo el transcurso del tiempo.

Operandos admitidos: formal (Especie parámetro: T). Ver [Tabla 16](#).

SE = (Retardo a la activación)

El temporizador especificado como operando formal se inicia en correspondencia con el flanco ascendente de RLC, con el tiempo especificado en ACCU1. El tiempo sigue transcurriendo mientras RLC=1. Si RLC pasa a 0 el tiempo se programa sobre 0.

El estado del temporizador es alto cuando el tiempo ha transcurrido y RLC se encuentra aún en el estado alto.

Operandos admitidos: formal (Especie parámetro: T). Ver [Tabla 16](#).

SVZ = (Impulso prolongado o set de contador)

Si el operando formal es un temporizador, éste se activa en correspondencia con el flanco ascendente de RLC, con el tiempo especificado en ACCU1. El estado de RLC no influencia a posteriori la temporización. El estado del temporizador es alto durante todo el transcurso del tiempo.

Si el operando formal es un contador, el conteo se pone en ACCU1 en correspondencia con el flanco ascendente de RLC. El estado del contador va alto si el valor de conteo programado es distinto de 0.

Operandos admitidos: formal (Especie parámetro: T, Z). Ver [Tabla 16](#).

SSV = (Retardo a la excitación con memoria o conteo hacia adelante)

Si el operando formal es un temporizador, éste se activa en correspondencia con el flanco ascendente de RLC con el tiempo especificado en ACCU1. Si RLC pasa a 0, el tiempo no recibe ninguna influencia. El estado del temporizador es alto cuando el tiempo ha transcurrido. Este tan sólo retorna al estado bajo después de un reset del temporizador con operación R.

Si el operando formal es un contador, el valor de conteo se incrementa en 1 en correspondencia con el flanco ascendente de RLC. El estado del contador sube o se mantiene alto.

Operandos admitidos: formal (Especie parámetro: T, Z). Ver [Tabla 16](#).

SAR = (Retardo a la desactivación o conteo hacia atrás)

Si el operando formal es un temporizador, éste se activa en correspondencia con el flanco descendente de RLC con el tiempo especificado en ACCU1. Si RLC pasa a 1, el tiempo se reprograma al valor inicial. El estado del temporizador es alto cuando RLC = 1 o bien cuando el tiempo está transcurriendo.

Si el operando formal es un contador, el valor de conteo disminuye en 1 en correspondencia con el flanco ascendente de RLC. El estado del contador va bajo si el valor de conteo pasa a 0.

Operandos admitidos: formal (Especie parámetro: T, Z). Ver [Tabla 16](#).

RD = (Reset de temporizador o contador)

Si RLC=1 el operando formal, temporizador o contador, se restaura.

Operandos admitidos: formal (Especie parámetro: T, Z). Ver [Tabla 16](#).

Operaciones especiales***STP (Stop al final de ciclo)***

La elaboración del programa continua hasta al final del ciclo de ejecución en vigor, la imagen de las salidas se transfiere y, a continuación, la CPU se lleva al estado de Stop.

Operandos admitidos: ninguno

STS (Stop inmediato)

La CPU se lleva al estado de Stop inmediatamente.

Operandos admitidos: ninguno

TAK (Intercambio de acumuladores)

Los contenidos de ACCU1 y ACCU2 se intercambian.

Operandos admitidos: ninguno

NOP 0 (Operación nula)

La operación se ignora. No se emprende ninguna acción.

Operandos admitidos: ninguno

NOP 1 (Operación nula)

La operación se ignora. No se emprende ninguna acción.

Operandos admitidos: ninguno

Tabla 17: conexión entre operaciones y RLC

RLC	Significado	Valores	Explicación
D	Dependiente	S	La ejecución de la operación depende del valor de RLC; para RLC=1 se ejecuta, para RLC=0 no se ejecuta.
		S↑	La ejecución de la operación depende del valor de RLC. Ésta se ejecuta solo en correspondencia con el flanco ascendente de RLC.
		S↓	La ejecución de la operación depende del valor de RLC. Ésta se ejecuta solo en correspondencia con el flanco descendente de RLC.
		N	La ejecución de la operación no depende del valor de RLC. Ésta se ejecuta siempre.
I	Influyente	S	La elaboración de la operación influye sobre el valor de RLC. El nuevo RLC dependerá del tipo de operación, del estado del operando y del anterior RLC.
		N	La elaboración de la operación no influye sobre el valor de RLC que así se mantiene inalterado.
L	Limitante	1	La elaboración de la instrucción influye sobre el valor de RLC que se sitúa incondicionadamente en 1.
		S	La operación es limitante en el cálculo de RLC. Cuando se encuentre la próxima operación RLC influyente, el valor de RLC dependerá solo del tipo de operación y del estado del operando, pero no del valor anterior de RLC.
		N	La operación no es limitante en el cálculo de RLC. Cuando se encuentre la próxima operación RLC influyente, el cálculo del nuevo RLC se efectuará en base al tipo de operación, al estado del operando y al valor anterior de RLC.

Set de operaciones

Las operaciones AWL implementadas en el PLC simulado, es decir, aquellas en que el compilador interno de PC-Sim reconoce y compila correctamente, constituyen, tal como acabamos de ver, una buena parte del set de operaciones del correspondiente PLC real.

La [Tabla 18](#) detalla sintéticamente el listado de las operaciones fundamentales implementadas en el PLC simulado, mientras que la [Tabla 19](#) indica el de las integrativas. Recordamos que estas últimas están disponibles para la programación tan solo en bloques funcionales (FB).

Las columnas correspondientes al resultado lógico combinatorio (RLC) tienen el significado expresado en [Tabla 17](#).

Tabla 18: operaciones fundamentales implementadas

Operación AWL	Operandos admitidos	RLC			Descripción
		D	I	L	
Operaciones lógicas					
U	E, A, M, T, Z	N	S	N	Combinación AND o test sobre el estado "1"
UN	E, A, M, T, Z	N	S	N	Combinación AND o test sobre el estado "0"
O	E, A, M, T, Z	N	S	N	Combinación OR o test sobre el estado "1"
ON	E, A, M, T, Z	N	S	N	Combinación OR o test sobre el estado "0"
O		N	S	S	Combinación OR de funciones AND
U(N	S	S	Combinación AND de expresión entre paréntesis
O(N	S	S	Combinación OR de expresión entre paréntesis
)		N	S	N	Final de una expresión entre paréntesis
Operaciones de memorización					
S	E, A, M	S	N	S	Activación del operando sobre RLC=1
R	E, A, M	S	N	S	Desactivación del operando sobre RLC=1
=	E, A, M	N	N	S	Asignación del operando al valor de RLC
Operaciones de carga					
L	EB	N	N	N	Carga de un byte de la IPI en ACCU1
L	AB	N	N	N	Carga de un byte de la IPO en ACCU1
L	EW	N	N	N	Carga de una word de la IPI en ACCU1
L	AW	N	N	N	Carga de una word de la IPO en ACCU1
L	MB	N	N	N	Carga de un merker byte en ACCU1
L	MW	N	N	N	Carga de una merker word en ACCU1
L	DL	N	N	N	Carga del byte de la izquierda de una data word en ACCU1
L	DR	N	N	N	Carga del byte de la derecha de una data word en ACCU1
L	DW	N	N	N	Carga de una data word en ACCU1
L	KB	N	N	N	Carga de una constante de byte en ACCU1
L	KF	N	N	N	Carga de una constante en coma fija en ACCU1
L	KH	N	N	N	Carga de una constante hexadecimal en ACCU1
L	KM	N	N	N	Carga de una constante binaria en ACCU1
L	KY	N	N	N	Carga de una constante de doble byte en ACCU1
L	KT	N	N	N	Carga de una constante de tiempo en ACCU1
L	KZ	N	N	N	Carga de una constante de conteo en ACCU1
L	T,Z	N	N	N	Carga de un valor de tiempo o de conteo en ACCU1
LC	T,Z	N	N	N	Carga de un valor de tiempo o de conteo, codificado BCD, en ACCU1
Operaciones de transferencia					
T	EB	N	N	N	Trasferencia del contenido de ACCU1 a un byte de la IPI

Operación AWL	Operandos admitidos	RLC			Descripción
		D	I	L	
T	AB	N	N	N	Transferencia del contenido de ACCU1 a un byte de la IPO
T	EW	N	N	N	Transferencia del contenido de ACCU1 a una word de la IPI
T	AW	N	N	N	Transferencia del contenido de ACCU1 a una word de la IPO
T	MB	N	N	N	Transferencia del contenido de ACCU1 a un merker byte
T	MW	N	N	N	Transferencia del contenido de ACCU1 a una merker word
T	DL	N	N	N	Transferencia del contenido de ACCU1 al byte de la izquierda de una data word
T	DR	N	N	N	Transferencia del contenido de ACCU1 al byte de la derecha de una data word
T	DW	N	N	N	Transferencia del contenido de ACCU1 a una data word
Operaciones de temporización					
SI	T	S↑	N	S	Activación de un temporizador a impulso
SV	T	S↑	N	S	Activación de un temporizador a impulso prolongado
SE	T	S↑	N	S	Activación de un temporizador con retardo a la activación
SS	T	S↑	N	S	Activación de un temporizador con retardo a la activación con memoria
SA	T	S↓	N	S	Activación de un temporizador con retardo a la desactivación
R	T	S	N	S	Reset de un temporizador
Operaciones de conteo					
ZV	Z	S↑	N	S	Incremento de contador
ZR	Z	S↑	N	S	Decremento de contador
S	Z	S↑	N	S	Programación de un contador
R	Z	S	N	S	Reset de un contador
Operaciones aritméticas					
+F		N	N	N	Adición de dos números en coma fija (ACCU2 + ACCU1)
-F		N	N	N	Sustracción de dos números en coma fija (ACCU2 – ACCU1)
Operaciones de comparación					
!=F		N	S	N	Comparación de igualdad de dos números en coma fija (ACCU2 = ACCU1)
><F		N	S	N	Comparación de desigualdad de dos números en coma fija (ACCU2 ≠ ACCU1)
>F		N	S	N	Comparación de superioridad de dos números en coma fija (ACCU2 > ACCU1)

Operación AWL	Operandos admitidos	RLC			Descripción
		D	I	L	
>=F		N	S	N	Comparación de superioridad o igualdad de dos números en coma fija ($ACCU2 \geq ACCU1$)
<F		N	S	N	Comparación de inferioridad de dos números en coma fija ($ACCU2 < ACCU1$)
<=F		N	S	N	Comparación de inferioridad o igualdad de dos números en coma fija ($ACCU2 \leq ACCU1$)
Operaciones de llamada de bloques					
SPA	PB	N	N	S	Salto absoluto a un bloque de programa
SPA	FB	N	N	S	Salto absoluto a un bloque funcional
SPA	OB	N	N	S	Salto absoluto a un bloque organizativo
SPB	PB	S	1	S	Salto condicionado a un bloque de programa
SPB	FB	S	1	S	Salto condicionado a un bloque funcional
SPB	OB	S	1	S	Salto condicionado a un bloque organizativo
A	DB	N	N	N	Apertura de un bloque datos
Operaciones de salto de retorno					
BE		N	N	S	Final de bloque
BEB		S	1	S	Final de bloque condicionado
BEA		N	N	S	Final de bloque absoluto
Operaciones especiales					
STP		N	N	N	Stop de la elaboración. La CPU se lleva a Stop al final del ciclo
NOP0		N	N	N	Ninguna operación
NOP1		N	N	N	Ninguna operación

Tabla 19: operaciones integrativas implementadas

Operación AWL	Operandos admitidos	RLC			Descripción
		D	I	L	
Operaciones lógicas sobre operando formal					
U=	Operando formal E, A, M, T, Z	N	S	N	Combinación AND o test sobre el estado "1"
UN=	Operando formal E, A, M, T, Z	N	S	N	Combinación AND o test sobre el estado "0"
O=	Operando formal E, A, M, T, Z	N	S	N	Combinación OR o test sobre el estado "1"
ON=	Operando formal E, A, M, T, Z	N	S	N	Combinación OR test sobre el estado "0"
Operaciones lógicas sobre word					
UW		N	N	N	Combinación AND (bit a bit) entre ACCU1 y ACCU2 con resultado en ACCU1
OW		N	N	N	Combinación OR (bit a bit) entre ACCU1 y ACCU2 con resultado en ACCU1
XOW		N	N	N	Combinación XOR (bit a bit) entre ACCU1 y ACCU2 con resultado en ACCU1
Operaciones de memorización sobre operando formal					
S =	Operando formal E, A, M	S	N	S	Activación del operando formal sobre RLC=1
RB =	Operando formal E, A, M	S	N	S	Desactivación del operando formal sobre RLC=1
= =	Operando formal E, A, M	S	N	S	Asignación del operando formal el valor de RLC
Operaciones de carga y transferencia de operando formal					
L =	Operando formal E, A, M, T, Z	N	N	N	Carga en ACCU1 del valor del operando formal
LC =	Operando formal T, Z	N	N	N	Carga en ACCU1 del valor codificado BCD del operando formal
LW =	Operando formal	N	N	N	Carga en ACCU1 del valor del operando formal constante
T =	Operando formal E, A, M	N	N	N	Transferencia del valor en ACCU1 al operando formal
Operaciones de temporización y conteo con operando formal					
SI =	Operando formal T	S↑	N	S	Activación de un temporizador a impulso
SE =	Operando formal T	S↑	N	S	Activación de un temporizador con retardo a la activación
SVZ =	Operando formal T, Z	S↑	N	S	Activación de un temporizador a impulso prolongado o bien programación de un contador
SSV =	Operando formal T, Z	S↑	N	S	Activación de un temporizador con retardo a la activación con memoria o bien conteo hacia delante de un contador

Operación AWL	Operandos admitidos	RLC			Descripción
		D	I	L	
SAR =	Operando formal T, Z	S↓ S↑ (*)	N	S	Activación de un temporizador con retardo a la desactivación o bien conteo hacia atrás de un contador
RD =	Operando formal T, Z	S	N	S	Desactivación del operando formal sobre RLC=1
Operaciones aritméticas					
ADD	BF	N	N	N	Suma a ACCU1 la constante de byte con signo (-128...+127), resultado en ACCU1
ADD	KF	N	N	N	Suma a ACCU1 la constante de word con signo (-32768...+32767), resultado en ACCU1
Operaciones de conversión					
KEW		N	N	N	Complemento de 1 de ACCU1
KZW		N	N	N	Complemento de 2 de ACCU1
Operaciones de desplazamiento					
SLW	n = 0...15	N	N	N	Desplazamiento hacia la izquierda del contenido de ACCU1 el número de bits indicado en el parámetro. Los sitios libres se llenan con 0.
SRW	n = 0...15	N	N	N	Desplazamiento hacia la derecha del contenido de ACCU1 el número de bits indicado en el parámetro. Los sitios libres se llenan con 0.
Operaciones de salto					
SPA =	Etiqueta	N	N	N	Salto absoluto a la etiqueta
SPB =	Etiqueta	S	1	S	Salto condicionado a la etiqueta (si RLC=1)
SPZ =	Etiqueta	N	N	N	Salto si el resultado es nulo
SPN =	Etiqueta	N	N	N	Salto si el resultado no es nulo
SPP =	Etiqueta	N	N	N	Salto si es positivo
SPM =	Etiqueta	N	N	N	Salto si es negativo
SPO =	Etiqueta	N	N	N	Salto por overflow
Operaciones de incremento/decremento					
D	n = 0...255	N	N	N	Decremento del byte inferior de ACCU1 en n
I	n = 0...255	N	N	N	Incremento del byte inferior de ACCU1 en n
Operaciones de elaboración					
B	DW	N	N	N	Elaboración de palabra datos: la instrucción siguiente se combina con el parámetro y se ejecuta
B	MW	N	N	N	Elaboración de palabra merker: la instrucción siguiente se combina con el parámetro y se ejecuta
B=	Operando formal B	N	N	N	Elaboración del bloque para OB, PB y FB; apertura del bloque para DB

(*) S↓ en caso de operando temporizador, S↑ en caso de operando contador

Operación AWL	Operandos admitidos	RLC			Descripción
		D	I	L	
Operaciones especiales					
STS		N	N	N	Stop de la elaboración. La CPU se lleva a Stop inmediatamente.
TAK		N	N	N	Intercambio del contenido de ACCU1 y ACCU2

Tabla 20: influencia de las operaciones sobre ANZ 1 y ANZ 0

Operaciones					Indicadores	
Aritméticas	Lógicas	Comparación	Desplazamiento	Conversión	ANZ 1	ANZ 0
Resultado =0	Resultado =0	ACCU2 =ACCU1	Bit desplazado=0	Resultado =0	0	0
Resultado <0		ACCU2 <ACCU1		Resultado <0	0	1
Resultado >0	Resultado ≠0	ACCU2 >ACCU1	Bit desplazado=1	Resultado >0	1	0

Programación en Ladder

La escritura de un programa para PLC en Ladder diagram o Esquema de contactos se realiza de manera similar al diseño de un circuito eléctrico con relés. Entre dos líneas verticales, dispuestas en los márgenes del esquema, que representan la alimentación del circuito, pueden colocarse los elementos del circuito eléctrico dispuestos sobre líneas horizontales (que a partir de ahora llamaremos ramales) conectados entre ellos por líneas horizontales y verticales.

Cada ramal puede contener, como máximo, 6 elementos del circuito en posiciones muy determinadas. En las 5 primeras posiciones pueden diseñarse tan sólo contactos y en la última, tan sólo bobinas.

Además, cada ramal puede subdividirse en dos zonas: la zona condicional, formada por las 5 primeras posiciones, que contiene las condiciones que deben cumplirse para realizar una acción determinada, y la zona ejecutiva, la última posición, que contiene las acciones a realizar cuando se cumplen todas las condiciones.

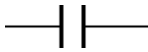
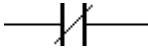
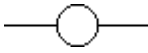
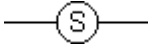

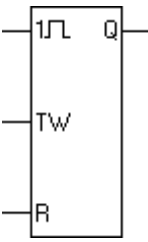
Esto es igual a lo que sucede en un circuito eléctrico: una carga eléctrica representa la parte ejecutiva de un ramal. Los contactos en serie o en paralelo, de tipo normalmente cerrados o abiertos, representan la parte condicional.

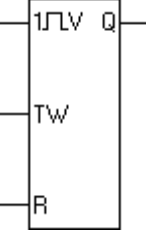
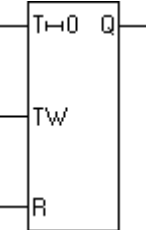
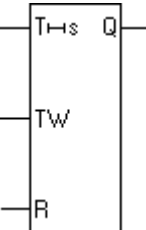
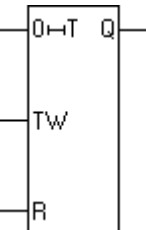
Cada contacto puede ponerse en paralelo solo o por grupos, dibujando contactos sobre el ramal siguiente que se conectarán al ramal base, o al ramal paralelo anterior, a través de líneas verticales. El máximo número de ramales paralelos permitidos es de 4. El conjunto del ramal base y de todos los ramales paralelos conectados a estos se detectará a continuación como recorrido de corriente. El editor numera cada recorrido de corriente automática y correlativamente.

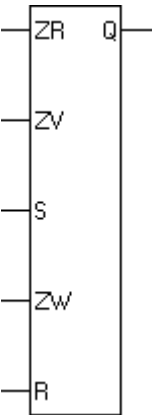
También las bobinas pueden conectarse en paralelo pero deben tener la entrada en común. Es decir, un recorrido de corriente puede enviar una única señal lógica a todas las bobinas.

En el interior del esquema de contactos pueden situarse los elementos descritos en la [Tabla 21](#).

Tabla 21: Elementos del ladder para S5 100U

Elemento	Descripción
Zona condicional	
	<p>Contacto normalmente abierto, test sobre el estado 1 del operando.</p> <p>El contacto sobre el esquema se cierra cuando el operando correspondiente se encuentra en el estado lógico alto. P. ej., si el operando es una entrada digital del PLC, el contacto está cerrado cuando la entrada está cerrada y viceversa.</p>
	<p>Contacto normalmente cerrado, test sobre el estado 0 del operando.</p> <p>El contacto sobre el esquema se cierra cuando el operando correspondiente se encuentra en el estado lógico bajo. P. ej., si el operando es una entrada digital del PLC, el contacto está cerrado cuando la entrada está abierta y viceversa.</p>
Zona ejecutiva	
	<p>Bobina directa</p> <p>Si la bobina está activa, es decir, si existe una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico alto. Si la bobina no está activa, al operando se le asigna el valor lógico bajo.</p>
	<p>Bobina con acción de set</p> <p>Si la bobina está activa, es decir, si hay una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico alto. Si la bobina no está activa, no se da ninguna acción sobre el estado del operando.</p>
	<p>Bobina con acción de reset</p> <p>Si la bobina está activa, es decir, si hay una cadena de contactos cerrados hacia la línea izquierda de alimentación, se asigna al operando el valor lógico bajo. Si la bobina no está activa no se da ninguna acción sobre el estado del operando.</p>
Temporizadores y contadores	
	<p>Temporizador SI (impulso)</p> <p>Entradas/salidas:</p> <p>1I entrada de puesta en marcha del temporizador</p> <p>TW entrada para constante de tiempo</p> <p>R entrada de reset</p> <p>Q salida del temporizador</p> <p>La temporización se activa sobre el flanco de cierre, hacia la alimentación de la entrada 1I (ver Figura 38) con el tiempo especificado en la entrada TW. Si la entrada se reabre el tiempo se repone a cero. El estado del temporizador es alto cuando el tiempo está transcurriendo y la entrada está en potencial alto. El estado se obtiene directamente de la salida Q o por medio de cualquier contacto con el mismo operando. Cuando la entrada de reset se conecta a la línea de alimentación, el tiempo se restaura y la salida se lleva a potencial bajo.</p>

Elemento	Descripción
	<p>Temporizador SV (impulso prolongado)</p> <p>Entradas/salidas:</p> <p>1I1V entrada de puesta en marcha del temporizador</p> <p>TW entrada para constante de tiempo</p> <p>R entrada de reset</p> <p>Q salida del temporizador</p> <p>La temporización se inicia sobre el flanco de cierre hacia la alimentación de la entrada 1I1V (ver Figura 39) con el tiempo especificado en la entrada TW. El estado del temporizador es alto cuando el tiempo está transcurriendo. El estado se obtiene directamente de la salida Q o por medio de cualquier contacto con el mismo operando. Cuando la entrada de reset se conecta a la línea de alimentación el tiempo se restaura y la salida se lleva a potencial bajo.</p>
	<p>Temporizador SE (retardo a la activación)</p> <p>Entradas/salidas:</p> <p>TI=0 entrada de puesta en marcha del temporizador</p> <p>TW entrada para constante de tiempo</p> <p>R entrada de reset</p> <p>Q salida del temporizador</p> <p>La temporización se inicia sobre el flanco de cierre hacia la alimentación de la entrada TI=0 (ver Figura 40) con el tiempo especificado en la entrada TW. El estado del temporizador es alto cuando el tiempo ha transcurrido y cuando la entrada está en potencial alto. El estado se obtiene directamente de la salida Q o por medio de cualquier contacto con el mismo operando. Cuando la entrada de reset se conecta a la línea de alimentación, el tiempo se restaura y la salida se lleva a potencial bajo.</p>
	<p>Temporizador SS (retardo a la activación con memoria)</p> <p>Entradas/salidas:</p> <p>TI=s entrada de puesta en marcha del temporizador</p> <p>TW entrada para constante de tiempo</p> <p>R entrada de reset</p> <p>Q salida del temporizador</p> <p>La temporización se inicia sobre el flanco de cierre hacia la alimentación de la entrada TI=s (ver Figura 41) con el tiempo especificado a la entrada TW. El estado del temporizador es alto cuando el tiempo ha transcurrido y se mantiene en este estado hasta la activación de la entrada reset. El estado se obtiene directamente de la salida Q o a través de cualquier contacto con el mismo operando.</p>
	<p>Temporizador SA (retardo a la desactivación)</p> <p>Entradas/salidas:</p> <p>0=T entrada de puesta en marcha del temporizador</p> <p>TW entrada para constante de tiempo</p> <p>R entrada de reset</p> <p>Q salida del temporizador</p> <p>La temporización se inicia sobre el flanco de apertura hacia la alimentación de la entrada 0=T (ver Figura 42) con el tiempo especificado en la entrada TW. El estado del temporizador es alto cuando la entrada está a potencial alto o cuando el tiempo está transcurriendo. El estado se obtiene directamente de la salida Q o a través de cualquier contacto con el mismo operando. Cuando la entrada de reset se conecta a la línea de alimentación el tiempo se restaura y la salida se lleva a potencial bajo.</p>

Elemento	Descripción
 <p>El diagrama muestra un rectángulo que representa el contador. A lo largo de su lado izquierdo hay cinco terminales etiquetados como ZR, ZV, S, ZW y R. A lo largo de su lado derecho hay un terminal etiquetado como Q.</p>	<p>Contador</p> <p>Entradas/salidas:</p> <p>ZR entrada conteo hacia atrás</p> <p>ZV entrada conteo hacia delante</p> <p>S entrada de programación</p> <p>ZW entrada para constante de tiempo</p> <p>R entrada de reset</p> <p>Q salida del contador</p> <p>Sobre el flanco de cierre hacia la alimentación de la entrada ZR, el conteo decrece en 1, y sobre el flanco de cierre hacia la alimentación de la entrada ZV, el conteo se incrementa en 1. Sobre el flanco de cierre hacia la alimentación de la entrada S el conteo se pone en el valor especificado en la entrada ZW. Sobre el cierre de la entrada R el conteo se restaura. La salida está a potencial alto si el valor de conteo es distinto de 0.</p> <p>El estado se obtiene directamente de la salida Q o a través de cualquier contacto con el mismo operando.</p>

CAPÍTULO 15

EJEMPLOS DE PROGRAMACIÓN PARA PLC S5 100U

En este capítulo se exponen, ordenados de menor a mayor complejidad, numerosos ejercicios de programación para el PLC S5 100U y cuya solución se facilita en forma de esquema de contactos y en la de lista de instrucciones. Para comprobar su funcionamiento tan sólo serán necesarios los interruptores o pulsadores para el control de las entradas del PLC. Los pilotos luminosos situados en la parte frontal del dispositivo valorarán el estado de las salidas. Todos los elementos necesarios se encuentran en la ventana PLC de PC-Sim.

Si durante la instalación de AW-SYS ha optado por instalar también los ejemplos, encontrará todos los que se exponen aquí en la carpeta Aw-sys\S5100\Esempi de su ordenador. Todos ellos pueden cargarse y verificarse con PC-Sim para S5 100U.

El nombre del archivo proyecto a cargar tiene el siguiente formato Sxx-yyyZ.prj, donde:

- S identifica un proyecto para S5 100U
- xx son dos cifras que valen siempre 00;
- yyy es el número correlativo del ejemplo expresado con tres cifras;
- Z es una letra que especifica el tipo de editor usado en el proyecto (L=ladder, A=AWL).

Es decir, si desea cargar el ejemplo de programación 8, en ladder abra el proyecto S00-008L.prj.

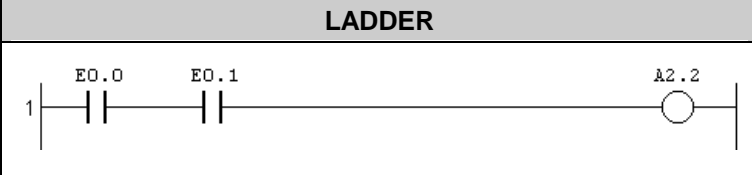
En esa misma carpeta podrá encontrar también otros archivos proyecto, en los que las cifras xx son distintas a 00. En ese caso, los ejemplos se refieren a las instalaciones y xx indica el número correspondiente a la instalación, tal como aparece en el [Menú Instalaciones](#). En la *Guía on line* de PC-Sim podrá encontrar más información sobre como cargar y comprobar los citados ejemplos.

Ejemplo 1: combinación AND

Realizar $A2.2 = E0.0 \text{ AND } E0.1$

La salida **A2.2** debe activarse única y exclusivamente si los dos interruptores conectados a las entradas **E0.0** y **E0.1** están cerrados.

La solución ladder se obtiene pilotando la bobina **A2.2** a través de la serie dos contactos con operandos **E0.0** y **E0.1**. De hecho, la combinación lógica AND, traducida al lenguaje ladder, equivale a la serie de dos contactos: En la disposición en serie 'se lee' el cierre del circuito solo cuando los dos contactos están cerrados. Así pues, ésta es la única condición que activa la bobina.

LADDER	AWL
	<pre> : U E0.0 : U E0.1 : = A2.2 : BE </pre>

La solución AWL se obtiene cargando en primer lugar el estado de **E0.0** en el registro RLC ($\text{U } E0.0$). De hecho, en un sentido estricto, la operación es una AND pero aquí, al ser la primera de una secuencia, se interpreta como una operación de carga de bytes, por lo tanto, el estado del operando se copia en RLC. A continuación, se efectúa una AND entre este último y el estado de la entrada **E0.1** ($\text{U } E0.1$) y el estado también se deposita en RLC. Esta instrucción no es la primera de una secuencia y por lo tanto la operación se interpreta en la práctica como una AND. La última instrucción ($= A2.2$) se encarga de transferir el contenido de RLC, que en este momento representa la combinación lógica **E0.0 AND E0.1**, a la salida **A2.2**.

Ejemplo 2: combinación OR

Realizar $A2.2 = E0.0 \text{ OR } E0.1$

La salida **A2.2** debe activarse si al menos uno de los interruptores conectados a las entradas **E0.0** o **E0.1** está cerrado.

La solución ladder se obtiene pilotando la bobina **A0.2** a través del paralelo de dos contactos, con operandos **E0.0** y **E0.1**. De hecho, la combinación lógica OR, traducida a esquema de contactos, equivale al paralelo de dos contactos: en las cabezas de la disposición en paralelo 'se lee' el cierre del circuito cuando al menos uno de los contactos está cerrado. Así pues, esta es la condición que conduce a la activación de la bobina.

LADDER	AWL
	<pre> : O E0.0 : O E0.1 : = A2.2 : BE </pre>

La solución AWL se obtiene cargando en primer lugar el estado de **E0.0** en el registro RLC (o **E0.0**). En efecto la operación en un sentido estricto es una OR pero aquí, siendo la primera de una secuencia, se interpreta como una operación de carga de bit y por tanto el estado del operando se copia en RLC. En este caso la instrucción es perfectamente equivalente a la Υ **E0.0**, que podrá sustituirse por esta obteniendo un programa con un funcionamiento idéntico. A continuación se efectúa una OR entre el RLC y la entrada **E0.1** (o **E0.1**), que deposita también el resultado en RLC. Esta instrucción no es la primera de una secuencia y por tanto la operación se interpreta efectivamente como una OR. La última instrucción (= **A2.2**) se ocupa de transferir el contenido de RLC, que en ese momento representa la combinación lógica **E0.0 OR E0.1**, a la salida **A2.2**.

Ejemplo 3: combinación AND de OR

Realizar **A2.0** = (**E0.0 OR E0.1**) AND (**E0.2 OR E0.3**)

La solución ladder debería obtenerse de manera intuitiva después de haber realizado los ejercicios anteriores: la bobina **A2.0** es pilotada por la serie (AND) de dos paralelos (OR) de contactos, en correspondencia con los cuales se han conectado adecuadamente los operandos.

LADDER	AWL
	<pre> : O E0.0 : O E0.1 : U(: O E0.2 : O E0.3 :) : = A2.0 : BE </pre>

Por su parte, la solución AWL requiere alguna aclaración ya que se han introducido dos nuevas operaciones. Observemos en primer lugar que, después de la ejecución de la segunda instrucción, el RLC contiene el resultado de la combinación lógica OR entre **E0.0** y **E0.1** (ver Ejemplo 2). La siguiente operación es una apertura de paréntesis; el RLC actual se deja de lado momentáneamente para combinarse a continuación en AND con el resultado de la expresión entre paréntesis. La operación Υ (es RLC limitante (ver [Tabla 18](#)) y por consiguiente la instrucción siguiente será la primera de una nueva secuencia. Así, la cuarta instrucción (o **E0.2**) será interpretada como un carga en RLC del estado de **E0.2** y, después de la ejecución de la instrucción siguiente, RLC contendrá el resultado de la combinación lógica **E0.2 OR E0.3**. La siguiente instrucción de cierre de paréntesis hará ejecutar al PLC la combinación AND (la tercera era Υ () entre el RLC actual, es decir, el resultado de la OR en el paréntesis, con el RLC que se había dejado a un lado con

anterioridad, o sea, el resultado de la primera OR. La última instrucción (= **A2.0**) se encarga de transferir el contenido de RLC, que en ese momento representa la combinación lógica (**E0.0 OR E0.1**) AND (**E0.2 OR E0.3**), en la salida **A2.0**.

Ejemplo 4: combinación OR de AND

Realizar **A2.0 = (E0.0 AND E0.1) OR (E0.2 AND E0.3)**. Donde los paréntesis, si bien no son necesarios ya que la operación AND tiene prioridad sobre la OR, se han añadido para mayor claridad.

En la solución ladder la bobina **A2.0** está pilotada por el paralelo (OR) de dos series (AND) de contactos, en correspondencia con los cuales se han conectado adecuadamente los operandos.

LADDER	AWL
	<pre> : U E0.0 : U E0.1 : O(: U E0.2 : U E0.3 :) : = A2.0 : BE </pre>

En lo que se refiere a la solución AWL, observemos en primer lugar que, después de la ejecución de la segunda instrucción, RLC contiene el resultado de la combinación lógica AND entre **E0.0** y **E0.1** (ver Ejemplo 1). La siguiente operación es una apertura de paréntesis, el RLC actual se deja de lado momentáneamente para combinarse más tarde en OR con el resultado de la expresión entre paréntesis. La operación **O(** es RLC limitante (ver [Tabla 18](#)) y, por lo tanto, la instrucción que le sucede será la primera de una nueva secuencia. Del mismo modo, la cuarta instrucción (**U E0.2**) será interpretada como una carga en RLC del estado de **E0.2** y, después de la ejecución de la siguiente instrucción, RLC contendrá el resultado de la combinación lógica **E0.2 AND E0.3**. La siguiente instrucción de cierre de paréntesis hará ejecutar al PLC la combinación OR (la tercera era **O(**) entre el RLC actual, o sea, el resultado de la AND entre paréntesis, con el RLC que antes se dejó de lado, es decir, el resultado de la primera AND. La última instrucción (= **A2.0**) se encarga de transferir el contenido de RLC, que en este momento representa la combinación lógica (**E0.0 AND E0.1**) OR (**E0.2 AND E0.3**), a la salida **A2.0**.

Ejemplo 5: combinación XOR

Realizar $A2.2 = E0.1 \text{ XOR } E0.2$

La operación lógica XOR a dos variables booleanas da resultado cierto solo cuando únicamente una de las dos variables es cierta.

La primera serie de contactos del programa ladder está cerrada solo cuando **E0.1** está cerrada y **E0.2** está abierta. Por el contrario, la segunda serie está cerrada solo cuando **E0.1** está abierta y **E0.2** está cerrada. Realizando el paralelo de las dos se obtiene la función deseada: la bobina se activa tan sólo cuando una entrada está cerrada y la otra está abierta.

LADDER	AWL
	<pre> : U E0.1 : UN E0.2 : O(: UN E0.1 : U E0.2 :) : = A2.2 : BE </pre>

La solución AWL no es más que la traducción del programa ladder y su funcionamiento es muy similar al del ejemplo anterior con la diferencia de que, en correspondencia con los contactos NC, se programan operaciones **UN**, test sobre el estado negado del operando.

Ejemplo 6: autorretención

Un pulsador conectado a la entrada **E0.0** debe activar la salida **A3.7**; y un segundo pulsador conectado a la entrada **E0.1** debe desactivarla.

En el programa ladder propuesto se realiza un circuito con autorretención. Pulsando el botón conectado a **E0.0** la bobina **A3.7** se activa y entonces, el contacto con el mismo operando en la segunda línea se cierra (imaginen que el contacto y la bobina son parte del mismo relé **A3.7**) y continua manteniendo activada la bobina incluso después de la apertura de **E0.0**. El cierre del pulsador en la entrada **E0.1** provoca la apertura del contacto, normalmente cerrado en el esquema, desactivando la bobina y cortando la autorretención.

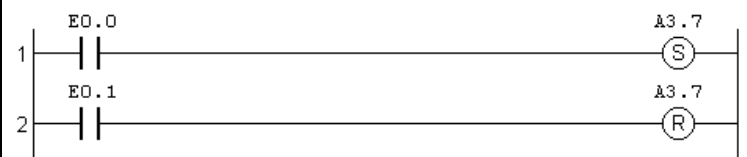
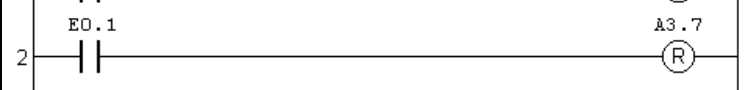
LADDER	AWL
	<pre> : U E0.0 : O A3.7 : UN E0.1 : = A3.7 : BE </pre>

El programa AWL propone la conversión de todo cuanto se acaba de describir. El valor de la salida **A3.7** en la última instrucción se calcula cargando el estado de **E0.0**; a continuación, valorando la OR con **A3.7** y por último poniendo en AND el resultado con el complemento de **E0.1**.

Ejemplo 7: set y reset

Un pulsador conectado a la entrada **E0.0** debe activar la salida **A3.7**; un segundo pulsador conectado a la entrada **E0.1** debe desactivarla

El ejercicio es idéntico al anterior pero, en esta ocasión, en la solución se emplean bobinas de set y reset de la misma salida **A3.7** pilotadas por separado por medio de contactos de **E0.0** y **E0.1**.

LADDER		AWL
1		: U E0.0
2		: S A3.7
		: U E0.1
		: R A3.7
		: BE

Cuando hacemos funcionar el programa, observamos que al presionar dos pulsadores a la vez se desactiva la salida. En efecto, en ambos lenguajes, cuando todas las condiciones de test son válidas a la vez, el operando **A3.7** se activa primero, en el recorrido 1 o con las dos primeras instrucciones, y luego, se desactiva en el recorrido 2 o con la tercera y cuarta instrucción. Pero recordemos que **A3.7** no representa efectivamente la salida física del PLC, sino el correspondiente bit en el interior de la memoria de las imágenes de proceso. Dicho bit de memoria se lleva efectivamente a 1 y luego a 0 pero, tan sólo al final de la carga del programa utilizado, el valor elaborado por él se transfiere al canal físico de salida correspondiente, que se mantiene constantemente en el valor bajo cuando ambas entradas están cerradas.

De este modo, con esta escritura de programas hemos hecho prevalecer el reset respecto al set. Si desea obtener lo contrario, le bastará con invertir la posición de los recorridos en el esquema de contactos o cambiar la primera secuencia por la segunda en el programa AWL.

Ejemplo 8: elevación de flancos

Activar las salidas **A2.0** y **A2.1** que corresponden, respectivamente, al flanco ascendente y al flanco descendente de la entrada **E0.0**.

Observamos que el último recorrido del esquema ladder y las dos últimas instrucciones del programa AWL imponen, al final de la ejecución del programa, la igualdad del merker bit **M0.0** con el estado de la entrada **E0.0**. Pero, en correspondencia con los flancos y para los recorridos o las instrucciones anteriores, se da el hecho de que el estado de las dos variables es opuesto y que tan sólo al final de la carga del programa utilizado se convierten en iguales. Todo ello queda representado en las dos primeras líneas del diagrama de [Figura 37](#) con un retraso temporal de **M0.0** respecto a **E0.0** que equivale a un ciclo de ejecución.

La parte inicial del programa activa la bobina **A2.0**, para un ciclo de ejecución, cuando **E0.0** está a 1 y **M0.0** está a 0, es decir, en correspondencia con el flanco ascendente de **E0.0**, tal como se muestra en la tercera línea del diagrama.

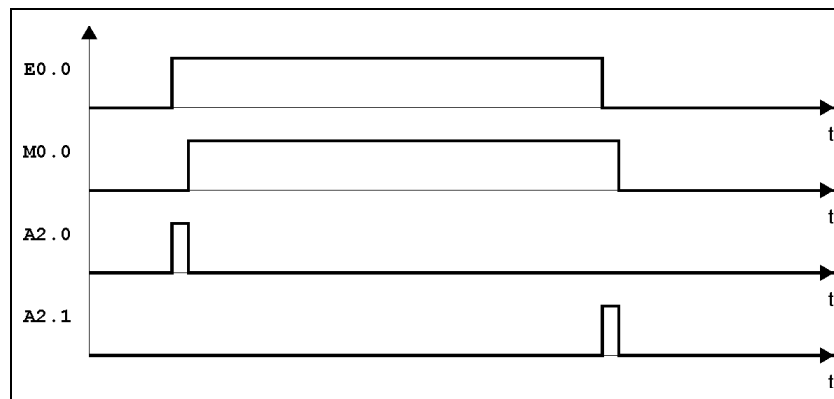
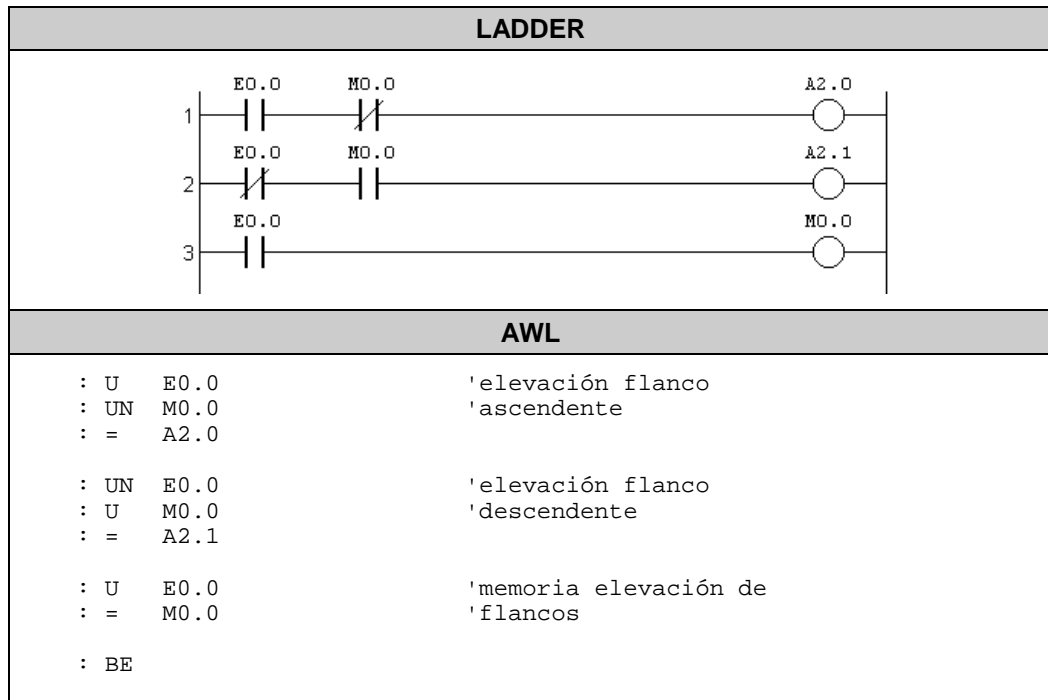


Figura 37: diagrama temporal de elevación de flancos

Por su parte, la bobina **A2.1** se activará, siempre para un ciclo de ejecución, cuando **E0.0** está a 0 y **M0.0** está a 1, es decir, correspondiendo con el flanco ascendente de **E0.0**, tal como muestra la cuarta línea del diagrama.

Ejemplo 9: temporizador a impulso

La salida **A2.5** se activa con el cierre de la entrada **E0.0** y se desactiva 5 segundos después. Si la entrada se reabre en este periodo, la salida se desactiva inmediatamente.

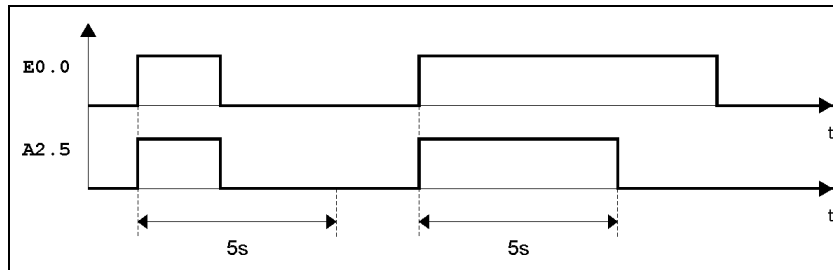
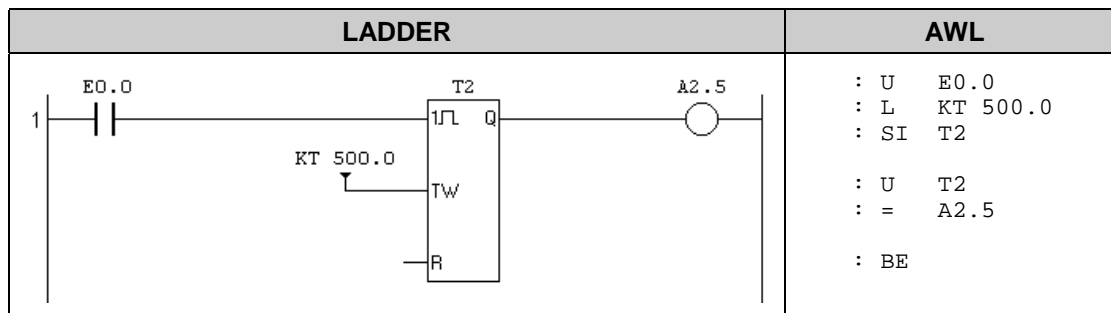


Figura 38: diagrama temporal temporizador a impulso

El funcionamiento del temporizador se muestra en el diagrama temporal de [Figura 38](#). En la primera línea está representada la señal de entrada y en la segunda, su salida.

El programa ladder se ha realizado utilizando un temporizador SI (impulso) activado por el contacto NA de **E0.0**, con una constante de tiempo igual a 500 centésimas de segundo y salida conectada a la bobina de **A2.5**.



El primer grupo de instrucciones del programa AWL constituye la activación del temporizador **T2** como impulso (**SI T2**) a continuación de un flanco ascendente de la entrada (**U E0.0**), con la constante de tiempo de 5 segundos cargada en el acumulador 1 (**L KT500.0**). El segundo grupo de instrucciones copia el estado del temporizador (**U T2**) en la salida del PLC (**= A2.5**).

Ejemplo 10: temporizador a impulso prolongado

La salida **A2.5** se activa al cierre de la entrada **E0.0** y se desactiva 5 segundos después, con independencia de si la entrada vuelve a abrirse o no durante este periodo.

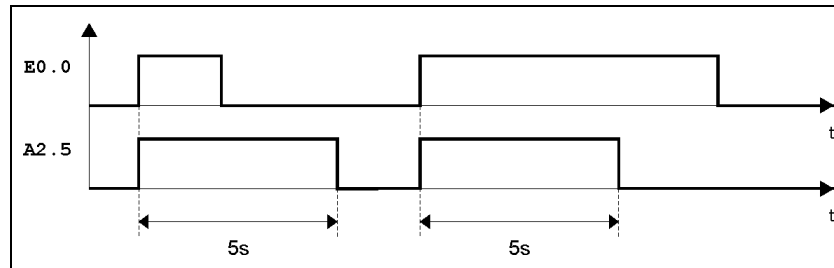
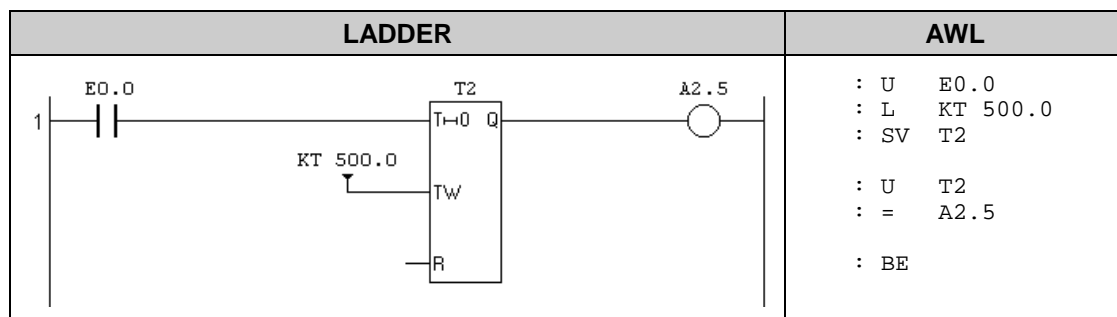


Figura 39: diagrama temporal del temporizador a impulso prolongado

El funcionamiento del temporizador está ilustrado en el diagrama temporal de [Figura 39](#). En la primera línea está representada la señal de entrada y en la segunda, su salida.

El programa ladder se ha realizado utilizando un temporizador SV (impulso prolongado) activado por el contacto NA de **E0.0**, con una constante de tiempo igual a 500 centésimas de segundo y salida conectada a la bobina de **A2.5**.



El primer grupo de instrucciones del programa AWL constituye la activación del temporizador **T2** como impulso prolongado (**SV T2**) a continuación de un flanco ascendente de la entrada (**U E0.0**), con la constante de tiempo de 5 segundos cargada en el acumulador 1 (**L KT500.0**). El segundo grupo de instrucciones copia el estado del temporizador (**U T2**) en la salida del PLC (**= A2.5**).

Ejemplo 11: temporizador con retardo a la activación

La salida **A2.5** se activa 5 segundos después de la cierre de la entrada **E0.0**. Cuando la entrada se reabre, la salida se desactiva.

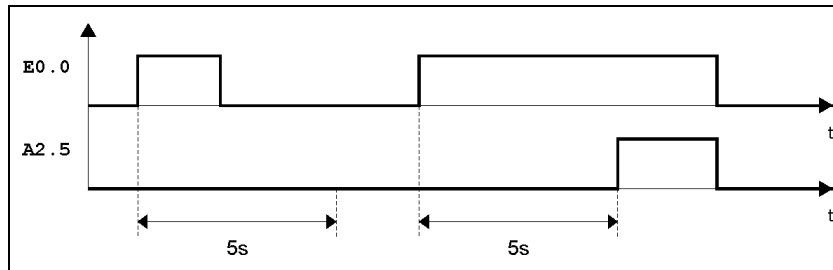
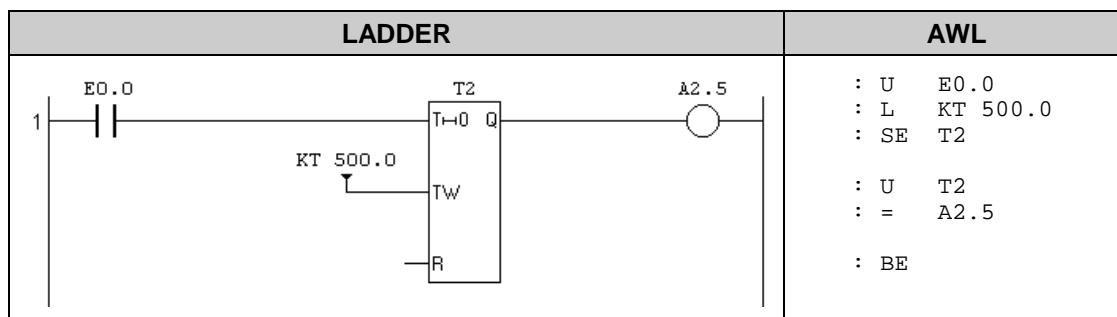


Figura 40: diagrama temporal del temporizador con retardo a la activación

El funcionamiento del temporizador queda ilustrado en el diagrama temporal de [Figura 40](#). En la primera línea está representada la señal de entrada y en la segunda, la salida que se obtiene.

El programa ladder se ha realizado utilizando un temporizador SE (retardo a la activación) activado por el contacto NA de **E0.0**, con una constante de tiempo igual a 500 centésimas de segundo y una salida conectada a la bobina de **A2.5**.



El primer grupo de instrucciones del programa AWL constituye la activación del temporizador **T2** como retardo a la activación (**SE T2**) a continuación un flanco ascendente de la entrada (**U E0.0**), con la constante de tiempo de 5 segundos cargada en el acumulador 1 (**L KT500.0**). El segundo grupo de instrucciones copia el estado del temporizador (**U T2**) en la salida del PLC (**= A2.5**).

Ejemplo 12: temporizador con retardo a la activación, con memoria y reset

La salida **A2.5** se activa 5 segundos después del cierre de la entrada **E0.0** (aunque si esta última, mientras tanto, vuelve a abrirse) y se desactiva en correspondencia con el cierre de la entrada **E0.1**.

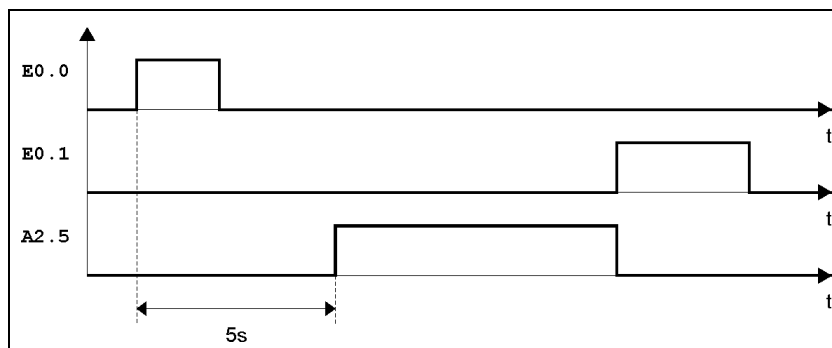
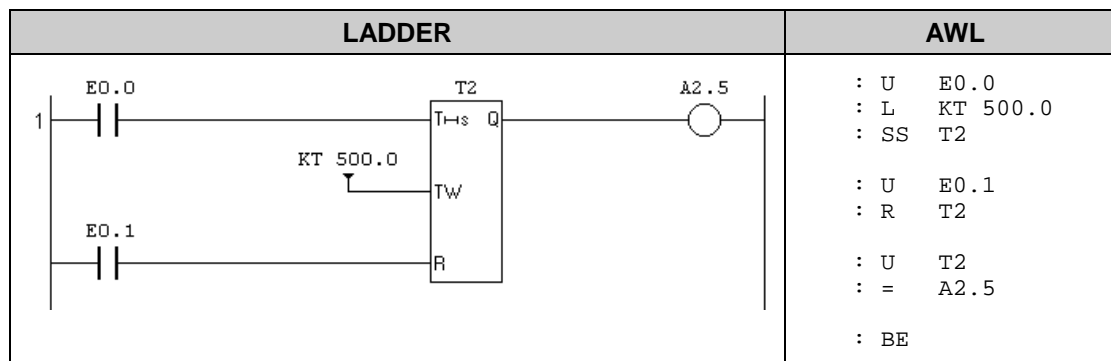


Figura 41: diagrama temporal para temporizador con retardo a la activación con memoria y reset

El funcionamiento del temporizador con retardo a la activación con memoria y reset se obtiene a partir de la comparación de las dos primeras líneas del diagrama temporal de [Figura 41](#) con la última. En las primeras están representadas las señales de entrada y en la última, la de la salida correspondiente.

El programa ladder se ha realizado utilizando un temporizador SS (retardo a la activación con memoria) activado por el contacto NA de **E0.0**, con una constante de tiempo igual a 500 centésimas de segundo, la entrada de reset conectada a un contacto NA de **E0.1** y la salida conectada a la bobina de **A2.5**.



El primer grupo de instrucciones del programa AWL constituye la activación del temporizador **T2** como retardo a la activación con memoria (**SS T2**) a continuación de un flanco ascendente de la entrada **E0.0** (**U E0.0**), con la constante de tiempo de 5 segundos cargada en el acumulador 1 (**L KT500.0**). El segundo grupo de instrucciones se encarga del reset del temporizador (**R T2**) en correspondencia con el estado alto de **E0.1** (**U E0.1**). El último grupo copia el estado del temporizador (**U T2**) en la salida del PLC (**= A2.5**).

Ejemplo 13: temporizador con retardo a la desactivación

La salida **A2.5** debe activarse al cierre de la entrada **E0.0** y desactivarse 5 segundos después de su reapertura.

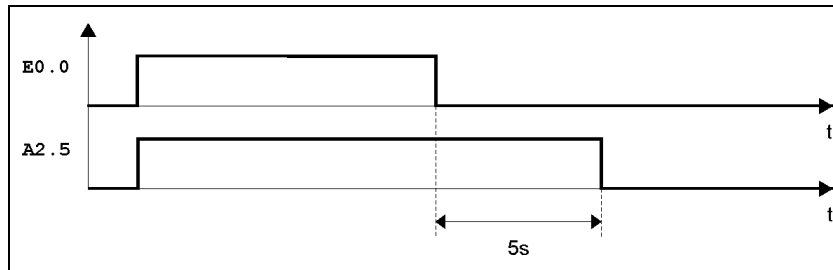
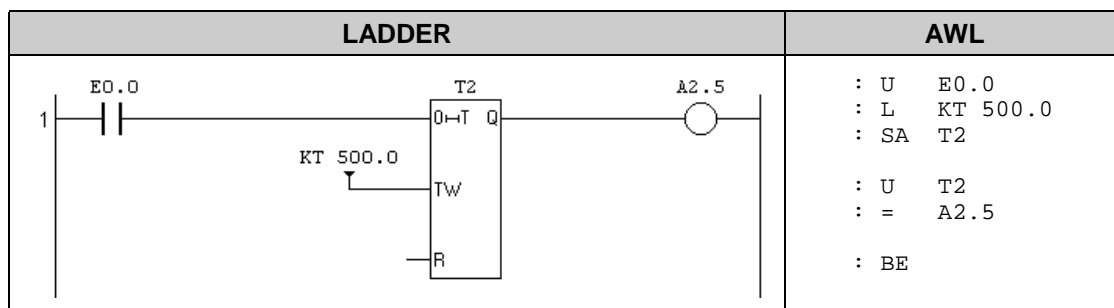


Figura 42: diagrama temporal de un temporizador con retardo a la desactivación

El funcionamiento del temporizador queda ilustrado en el diagrama temporal de [Figura 42](#). En la primera línea está representada la señal de entrada y en la segunda, la correspondiente salida.

El programa ladder se ha realizado utilizando un temporizador SA (retardo a la desactivación) activado por el contacto NA de **E0.0**, con una constante de tiempo igual a 500 centésimas de segundo y salida conectada a la bobina de **A2.5**.



El primer grupo de instrucciones del programa AWL constituye la activación del temporizador **T2** como retardo a la desactivación (**SA T2**) a continuación de un flanco descendiente de la entrada (**U E0.0**), con la constante de tiempo de 5 segundos cargada en el acumulador 1 (**L KT500.0**). El segundo grupo de instrucciones copia el estado del temporizador (**U T2**) en la salida del PLC (**= A2.5**).

Ejemplo 14: temporizador con retardo a la activación y a la desactivación

La salida **A3.3** se activa 3 segundos después del cierre de la entrada **E0.1** y se desactiva 7 segundos después de su reapertura.

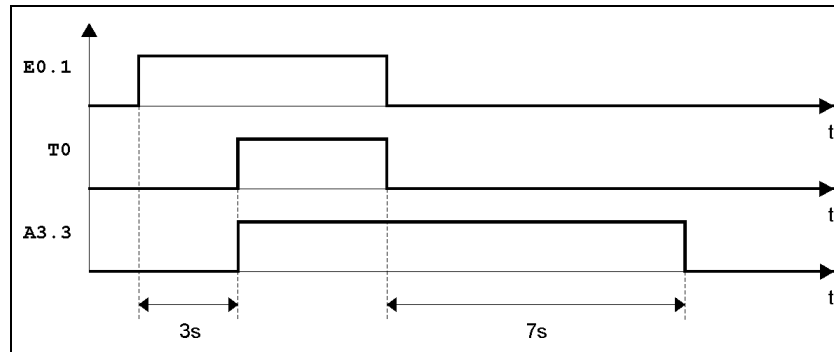
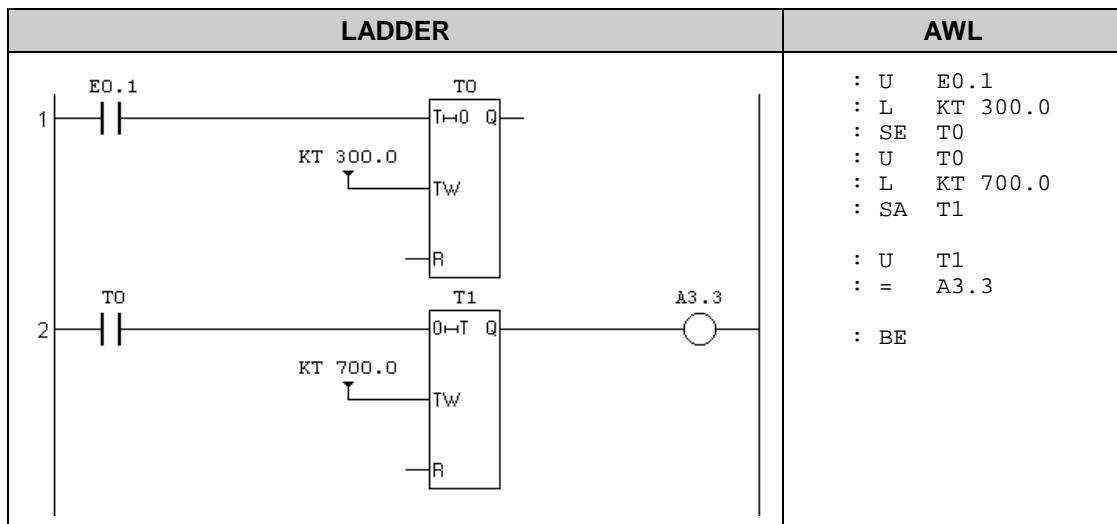


Figura 43: diagrama temporal del temporizador con retardo a la activación y a la desactivación

La [Figura 43](#) ilustra, en la primera y última línea del diagrama, el avance de las dos señales descritas en el trazado. En la línea intermedia se ha dibujado el desarrollo de un temporizador con retardo a la activación que precisamente tiene como entrada a **E0.1**. Observamos como el desarrollo de la salida **A3.3**, respecto al de **T0**, representa un retardo a la desactivación. ¡Hagan juego, señores! Ahora se trata de escribir un programa que contenga dos temporizadores: el primero, un retardo a la activación de 3 s, tiene como entrada **E0.1**; el segundo, un retardo a la desactivación de 7 s, tiene como entrada el estado del primer temporizador y como salida **A3.3**.

Los programas ladder y AWL que siguen, implementan esta solución.



Ejemplo 15: impulso retardado

La salida **A2.7** se activa 2 segundos después de la apertura de la entrada **E1.5** durante 1 segundo.

El diagrama en *Figura 44* ilustra, en la primera línea, el avance de la entrada y, en la última, el de la salida que se desea obtener. La segunda y la tercera línea representan el desarrollo de dos temporizadores con retardo a la desactivación **T10** y **T11**, en 2 y 3 segundos respectivamente, que tienen, como señal de entrada, precisamente **E1.5**.

Observamos entonces que la salida debe ser cierta cuando se cumplen a la vez las condiciones: **T11** cierta y **T10** falsa. Así pues, en términos de expresión joleana:

$$A2.7 = T11 \cdot \overline{T10}$$

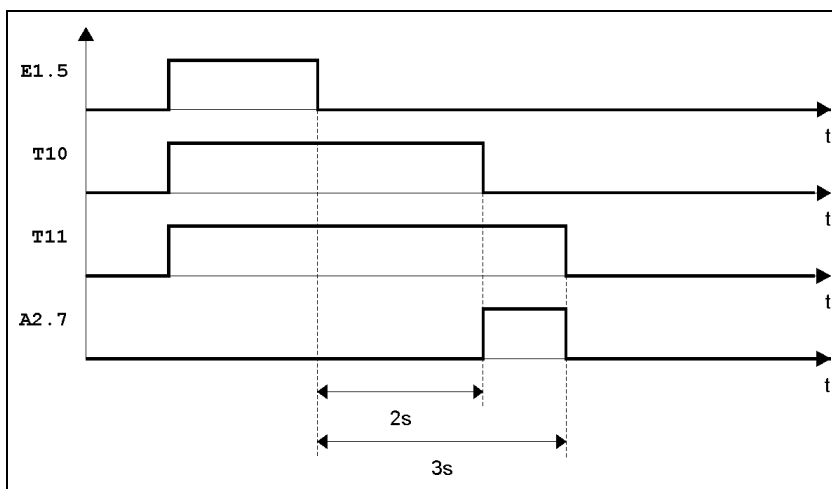


Figura 44: diagrama temporal del impulso retratado

LADDER	AWL
	<pre> : U E1.5 : L KT 200.0 : SA T10 : L KT 300.0 : SA T11 : U T11 : UN T10 : = A2.7 : BE </pre>

En el programa ladder, los dos primeros recorridos están destinados a la activación de los temporizadores y ambos tienen como entrada la señal **E1.5**. En el tercer recorrido, se implementa la expresión lógica recién

obtenida: los dos contactos dispuestos en serie desarrollan la función de la operación AND y, en concreto para el segundo, se ha usado el tipo NC con el fin de ejecutar el complemento de su operando.

Del mismo modo, para el programa AWL, el primer grupo de instrucciones activa los dos temporizadores sobre la señal de entrada. El segundo grupo calcula el valor de la expresión booleana y lo asigna a la salida.

Ejemplo 16: tren de impulsos

La salida **A2.4** debe activarse un instante a cada segundo.

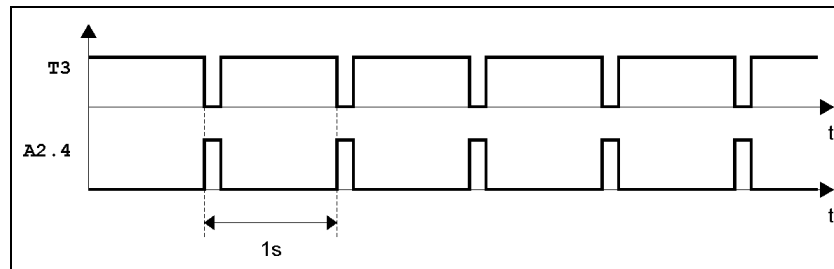
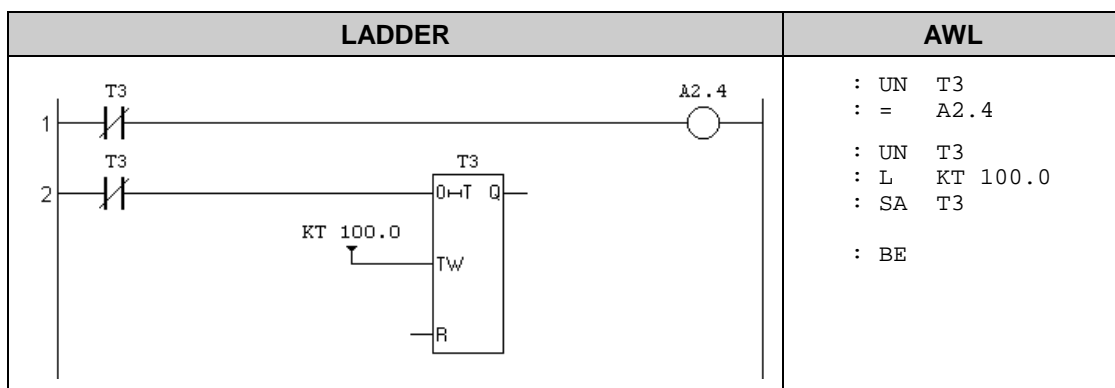


Figura 45: diagrama temporal del tren de impulsos

La ejercicio se resuelve utilizando un temporizador que se autoinicia cíclicamente. Concentrémonos en primer lugar en el recorrido 2 del ladder y en el segundo grupo de instrucciones de la AWL. La primera ejecución del programa encuentra el temporizador detenido, y por tanto su contacto NC en el esquema ladder está cerrado y el test sobre el estado bajo en el programa AWL queda resuelto satisfactoriamente. En ambos casos, la salida del temporizador con retardo a la desactivación se lleva al estado alto. A causa de ello, al ciclo siguiente, los tests que acabamos de describir dejarán de verificarse. Así pues, la entrada del temporizador se ha llevado a cero y el tiempo empieza transcurrir. Durante todo este periodo, la salida permanece alta. Al finalizar el tiempo, se repone a cero. A continuación, las condiciones de test respecto a la entrada del temporizador vuelven a comprobarse y todo se repite tal como se ha descrito.

El diagrama de [Figura 45](#) describe, en la primera línea, el desarrollo de la salida del temporizador que resulta ser la señal opuesta a la que pide el trazado. Ya solo queda invertir esta señal y asignarla a la salida **A2.4**. El primer recorrido del esquema de contactos y el primer grupo de instrucciones del programa AWL se encargan de cumplir esta misión.

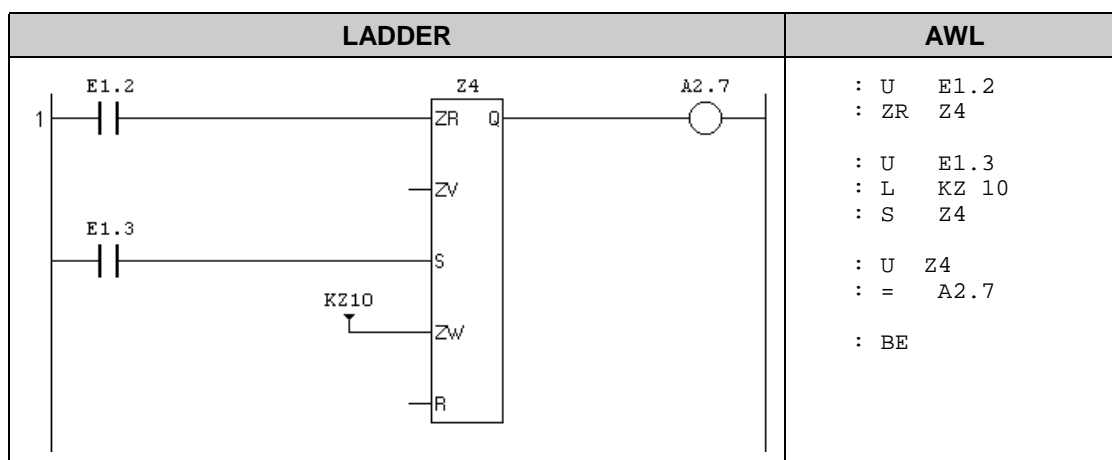


Solo falta por aclarar que el recorrido para la asignación de **A2.4** debe preceder necesariamente al de la activación del temporizador. De hecho, si no se hiciera así, el tiempo se reiniciaría antes de que **A2.4** pudiera ser programado a 1. **A2.4** permanecería siempre en el estado bajo ya que el estado de **T3**, visto en este momento de la elaboración del programa, aparecería siempre alto. Lo mismo puede decirse para el programa AWL, en el que el grupo de instrucciones de asignación de **A2.4** debe preceder, por esas mismas razones, al del inicio de **T3**. Es conveniente recordar estas consideraciones cada vez que, en los ejemplos siguientes, utilizaremos un tren de impulsos.

Ejemplo 17: conteo hacia atrás

*El contador **Z4** se pone en el valor 10 en correspondencia con el cierre de la entrada **E1.3** y se va decrementando a cada cierre de la entrada **E1.2**. La salida **A2.7** se desactiva al final del conteo (valor 0).*

El programa ladder es muy sencillo. La constante **KZ10**, asignada a la entrada **ZW**, proporcionará el valor de preset 10 al contador. Su entrada de set (S) se ha conectado a un contacto de **E1.3**, mientras que el de decremento (ZR) lo ha hecho a un contacto de **E1.2**. Sobre el flanco de cierre de **E1.3** se programa el valor de conteo a 10, y a cada flanco de cierre de **E1.2** el conteo disminuye en 1. La salida del contador (Q) está en potencial alto cuando el valor de conteo es distinto de 0. Así pues, bastará con pilotar con ésta la salida **A2.7** del PLC.

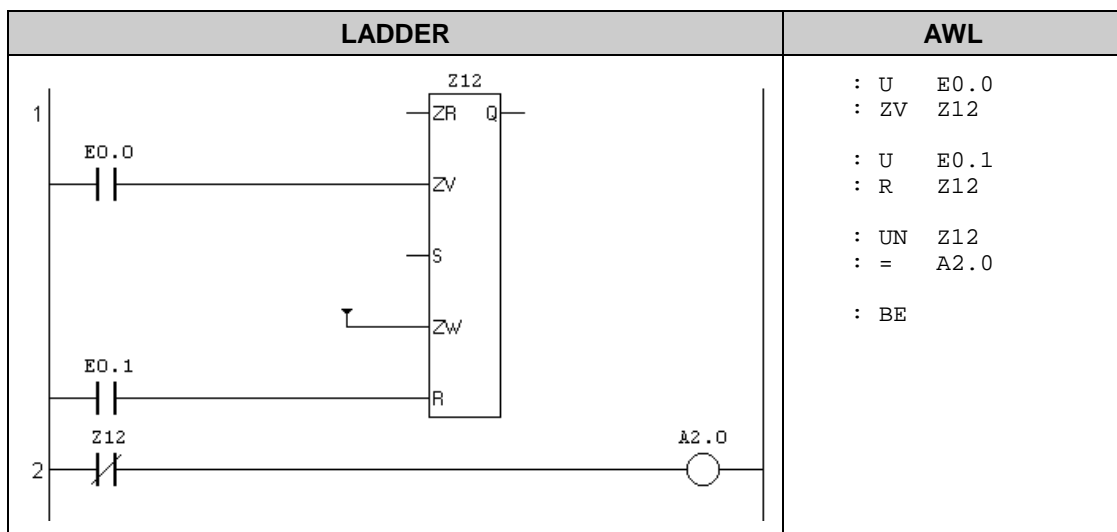


El primer grupo de instrucciones del programa AWL se encarga de incrementar en 1 el valor de conteo del contador **Z4** a cada flanco ascendente de **E1.2**. El segundo efectúa la programación del conteo al valor cargado en ACCU1 (10) en correspondencia con el flanco ascendente de **E1.3**. El último grupo se encarga de transferir el estado del contador a la salida **A2.7**.

Ejemplo 18: conteo hacia delante

El contador **Z12** se incrementa a cada cierre de la entrada **E0.0** y se restaura a cero en correspondencia con el cierre de la entrada **E0.1**. La salida **A2.0** está activa cuando el valor de conteo es cero.

La entrada de reset (R) del contador **Z12** se ha conectado a un contacto de **E0.1**, mientras que el de incremento (ZV) lo ha hecho a un contacto de **E0.0**. Sobre el flanco de cierre de **E0.1** se produce el reset del contador, es decir, la programación del valor de conteo a 0, y a cada flanco de cierre de **E0.0** el conteo se incrementa en 1. La salida del contador (Q) está en potencial alto cuando el valor de conteo es distinto de 0. Así pues, bastará con invertirla para pilotar la salida **A2.0** del PLC, tal como se ha hecho en el recorrido 2.



El primer grupo de instrucciones del programa AWL se encarga de incrementar en 1 el valor de conteo del contador **Z12** a cada flanco ascendente de **E0.0**. El segundo efectúa el reset del conteo en correspondencia del estado alto de **E0.1**. El último grupo se encarga de transferir el complemento del estado del contador a la salida **A2.0**.

Ejemplo 19: conteo del tiempo de cierre de una entrada (en segundos)

Determinar durante cuantos segundos permanece cerrada la entrada **E0.5** y utilizar la entrada **E0.6** para reponer a cero el conteo del tiempo.

En primer lugar, hay que realizar una base de tiempos de un segundo, es decir, un tren de impulsos que tenga este período. Así pues, será suficiente contar cuantos impulsos de base de tiempos se generan durante el cierre de la entrada, es decir, los instantes en los cuales son ciertos a la vez, la entrada y el impulso (ver [Figura 46](#)).

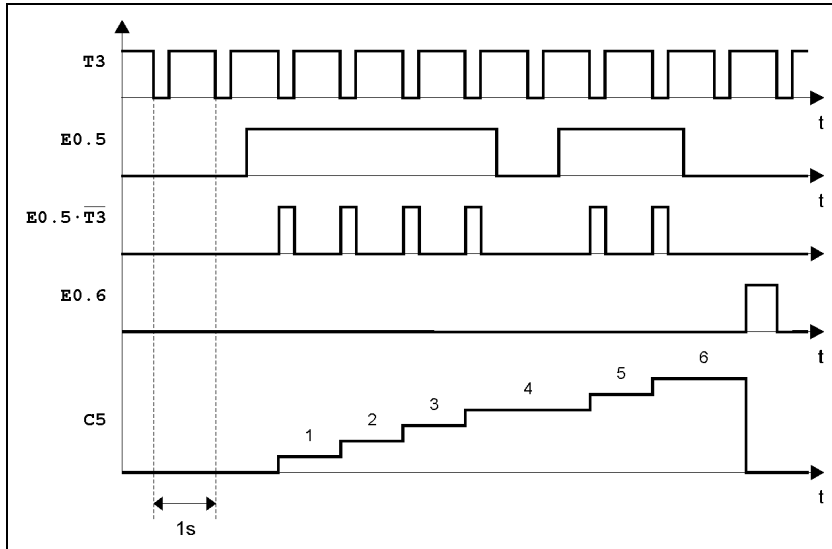
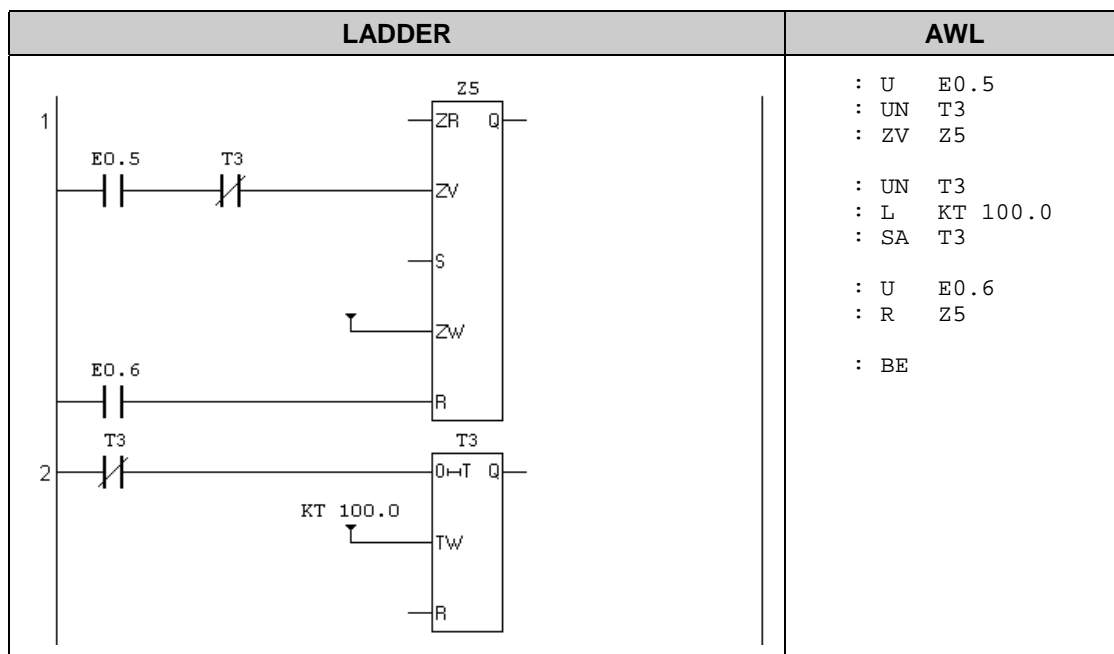


Figura 46: diagrama temporal del contador de tiempo de cierre de una entrada

El recorrido 2 del esquema de contactos implementa el tren de impulsos de 1 segundo de periodo, tal como hemos visto en el ejemplo 16. En el recorrido 1 observamos que la entrada ZV del contador está pilotada por la combinación lógica AND (serie de contactos en el diagrama) entre la entrada E0.5 y el tren de impulsos, es decir, está pilotado precisamente por los impulsos que hay que contar. Por su parte, un contacto de E0.6 controla la entrada reset del contador para restaurar el conteo, tal como lo requiere el trazado.

En el programa AWL, el primer grupo de instrucciones se ocupa del incremento del contador en correspondencia con el flanco ascendente de la AND entre E0.5 y el tren de impulsos generado por T3 en el segundo grupo de instrucciones. El último grupo opera el reset del conteo sobre el estado alto de la entrada E0.6.



El valor de conteo de **z5** representa el número de segundos durante los cuales la entrada ha estado cerrada, con el límite propio de los contadores de este PLC, que es 999.

Ejemplo 20: conteo del tiempo de cierre de una entrada (en horas, minutos y segundos)

*Determinar cuantas horas, minutos y segundos la entrada **E0.5** permanece cerrada y utilizar la entrada **E0.6** para restaurar el conteo del tiempo.*

Para programar la solución de este problema se ha hecho uso de símbolos. Su correspondencia con los operandos absolutos se ha establecido según la tabla siguiente.

Op. absoluto	Símbolo	Comentario
E0.5	INGRESSO	Entrada para conteo del tiempo
E0.6	RESET	Entrada para reset del conteo
T3	CLOCK	Tren de impulsos con periodo 1seg
Z5	SEC	Contador de los segundos
Z6	MIN	Contador de los minutos
Z7	ORE	Contador de las horas

El programa propuesto termina con la construcción de un tren de impulsos con el temporizador **CLOCK**, que actuará como base de tiempos con periodo de 1 segundo. Al inicio del mismo, las tres primeras instrucciones hacen avanzar el conteo del contador **SEG** cuando un impulso de **CLOCK** se detecta durante el cierre de **INGRESSO SEC**, y por tanto **Z5**, es el contador de los segundos.

El segundo grupo de instrucciones se ocupa de la carga del valor de los segundos en **ACCU2** y de la constante 60 en **ACCU1**. A continuación, se busca la igualdad de los dos valores y, en caso de test afirmativo, el contador **MIN** se incrementa y el **SEG** se repone a cero. Así, el contador **MIN** se incrementa cada 60 segundos y constituye el contador de los minutos.

El tercer grupo de instrucciones se ocupa de la carga del valor de los minutos en **ACCU2** y de la constante 60 en **ACCU1**. Después, se busca la igualdad de los dos valores y, en caso afirmativo, el contador **HORAS** se incrementa y el contador **MIN** se repone a cero. Así, el contador **HORAS** se incrementa cada 60 minutos y constituye el contador de las horas.

Como resumen, el cuentatiempo de software que hemos construido nos permite contar hasta 999 horas, 59 minutos y 59 segundos (precisos!). Todo ello, admitiendo que se pueda considerar un valor tan preciso, a lo largo de un período de tiempo tan largo, teniendo presentes los inevitables errores del reloj interno, tanto del PLC real como el del PC sobre el que 'gira' el simulado.

Como ejercicio práctico, puede intentar modificar el programa añadiendo un contador **DIAS**, que se incrementará en 1 cada 24 horas.

Para verificar el programa, sin tener que esperar tiempos excesivamente largos, puede disminuir la constante de tiempo en la carga de **CLOCK**, aumentando así la frecuencia del tren de impulsos, o bien forzar manualmente valores de conteo próximos a los de comparación.

AWL		
: U	-INGRESSO	'cada segundo incrementa el contador
: UN	-CLOCK	'de los segundos
: ZV	-SEC	
: L	-SEC	'si ha llegado a 60 segundos
: L	KF +60	
: !=F		
: ZV	-MIN	'incrementa el contador de los minutos
: R	-SEC	'y restaura el de los segundos
: L	-MIN	'si ha llegado a 60 minutos
: L	KF +60	
: !=F		
: ZV	-ORE	'incrementa el contador de las horas
: R	-MIN	'y restaura el de los minutos
: U	-RESET	'repone los contadores tras el
: R	-SEC	'cierre de la entrada RESET
: R	-MIN	
: R	-ORE	
: UN	-CLOCK	'construcción tren de impulsos con
: L	KT 100.0	'periodo de 1 segundo
: SA	-CLOCK	
: BE		

Ejemplo 21: generador de onda cuadrada

La salida **A2.7** debe estar controlada por una señal de onda cuadrada con $T_{off}=1,5\text{ s}$ y $T_{on}=0,5\text{ s}$

En la solución propuesta se utilizan dos temporizadores que se “rebotan” la activación el uno al otro.

En la primera elaboración del programa ladder, el contacto NC de **T2** está cerrado y por lo tanto, la salida del temporizador **T1** (retardo a la desactivación) se activa. En ese momento, también el contacto NA de **T1** en el recorrido 2 está cerrado y entonces la salida de **T2** se activa.

Al ciclo siguiente, el contacto NC de **T1** está abierto y el tiempo de **T1** empieza su avance. Su salida continua manteniéndose alta y deja a **T2** en el mismo estado.

Finalizado el tiempo de **T1** (1,5 s) la salida del mismo va baja y el contacto NA sobre el recorrido 2 se abre, el temporizador **T2** se inicia y su tiempo empieza a transcurrir, mientras que su salida sigue manteniéndose alta.

Transcurrido el tiempo **T2**, la salida del temporizador va baja. Hemos vuelto así a la condición inicial y el ciclo se repite tal como se ha descrito, indefinidamente.

La desarrollo temporal de las señales **T1** y **T2** se muestra en las dos primeras líneas de la [Figura 47](#); mientras que la tercera línea muestra el desarrollo que debería tener la salida **A2.7**. Observemos como esta salida es el complemento de la señal **T1**, salvo en la pequeña diferencia de tiempo igual a un ciclo de

ejecución que, para mayor claridad, se ha exagerado expresamente en el diagrama. Concluimos el programa implementando, en el tercer recorrido, la asignación de la salida del PLC segundo este criterio.

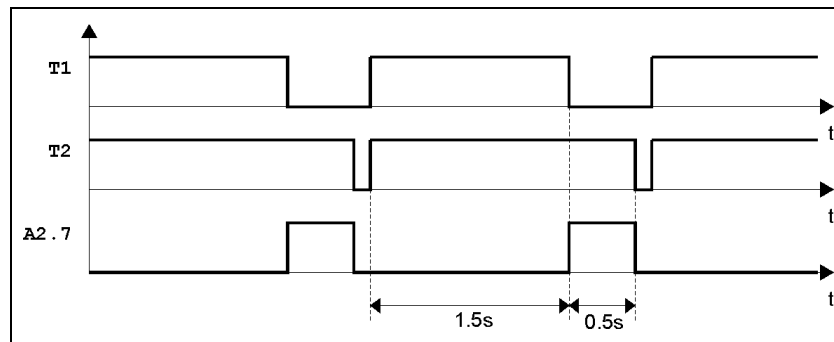
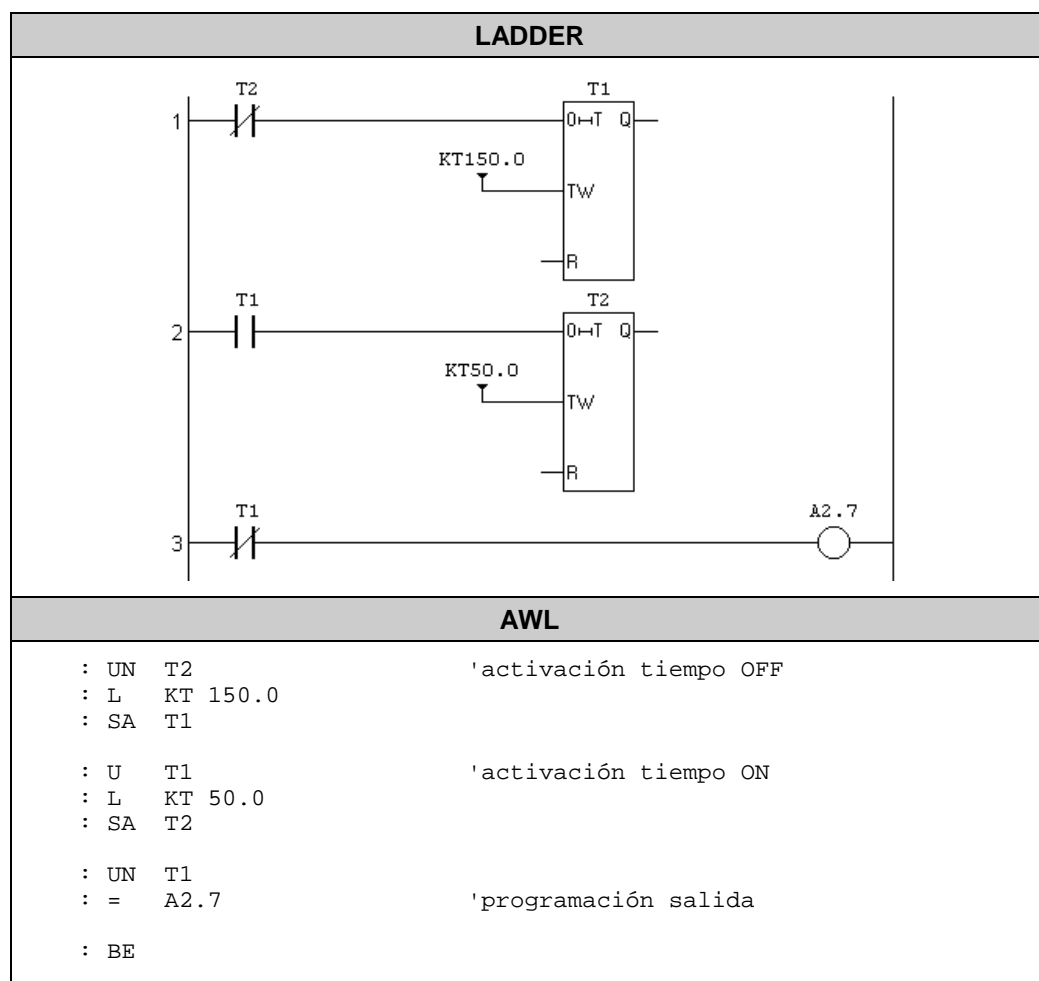


Figura 47: diagrama temporal del generador de onda cuadrada

El programa AWL es la simple conversión del ladder y consideramos que, a estas alturas, no debería ser de difícil comprensión.

Programando adecuadamente el valor de las dos constantes de tiempo se puede variar T_{on} y T_{off} realizando una onda cuadrada con diversos *duty cycle* y frecuencias.



Ejemplo 22: otro generador de onda cuadrada

La salida **A2.7** debe estar controlada por una señal de onda cuadrada con $T_{off}=1.5$ seg. y $T_{on}=0.5$ seg.

La solución aquí presentada se diferencia de la que se propone en el ejemplo precedente y que tiene el mismo trazado. En efecto, aquí se ha utilizado un único temporizador de autoarranque que se programa, en el primer grupo de instrucciones, con un tiempo igual al periodo de la señal requerida (2 s). Así pues, en su transcurrir, el tiempo varía entre 200 y 0 centésimas de segundo. Entre los valores 200 y 50 la salida **A2.7** debe programarse al estado bajo, mientras que para valores menores de 50, esa misma deberá asumir el estado alto.

El segundo grupo de instrucciones se encarga de la comparación del valor de tiempo actual con la constante 50 y, si es menor, activa la salida.

También en este caso se puede cambiar tanto el periodo como la señal, variando la constante de tiempo de **T3**, así como el tiempo en el estado alto, variando la constante decimal de comparación.

AWL		
: UN	T3	'puesta en marcha temporizador
: L	KT 200.0	'periodo onda cuadrada
: SA	T3	
: L	T3	'comparación con tiempo
: L	KF +50	'en el estado ON
: <F		
: =	A2.7	'activación salida
: BE		

Ejemplo 23: control temporizado de luces

Un pulsador conectado a la entrada **E0.0** activa durante tres minutos un grupo de luces conectadas a la salida **A2.1**. Junto a éstas se activa un piloto luminoso conectado a la salida **A2.2** que, 15 segundos antes de que las luces se apaguen, empieza a parpadear para avisar que el tiempo está a punto de agotarse. El piloto luminoso se apaga definitivamente a la vez que las luces.

Los temporizadores **T2** y **T3** se utilizan para generar una onda cuadrada con periodo de 1 s y *duty cycle* del 50%. Ambos están cargados con una constante de tiempo de 50 centésimas de segundo. Para más información al respecto, consulte el primer ejemplo sobre la generación de una onda cuadrada.

Además se utilizan otros dos temporizadores a impulso prolongado. **T1**, cargado con un tiempo de 3 minutos (**KT180.2** es decir 180 segundos), controla directamente la salida del grupo de luces. **T0**, cargado con un tiempo inferior en 15 segundos (**KT165.2**), será utilizado para determinar el instante del inicio del parpadeo del piloto luminoso.

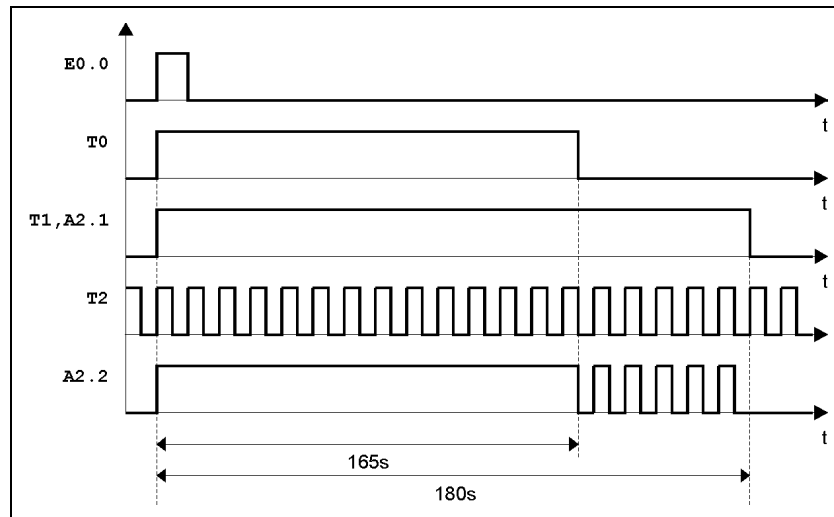


Figura 48: diagrama temporal del control de luces temporizado

Los desarrollos temporales de $T0$ y $T1$ a continuación del impulso sobre $E0.0$ se recogen, respectivamente, en la segunda y tercera línea del diagrama de *Figura 48*. La cuarta línea muestra el desarrollo, no a escala, del temporizador $T2$.

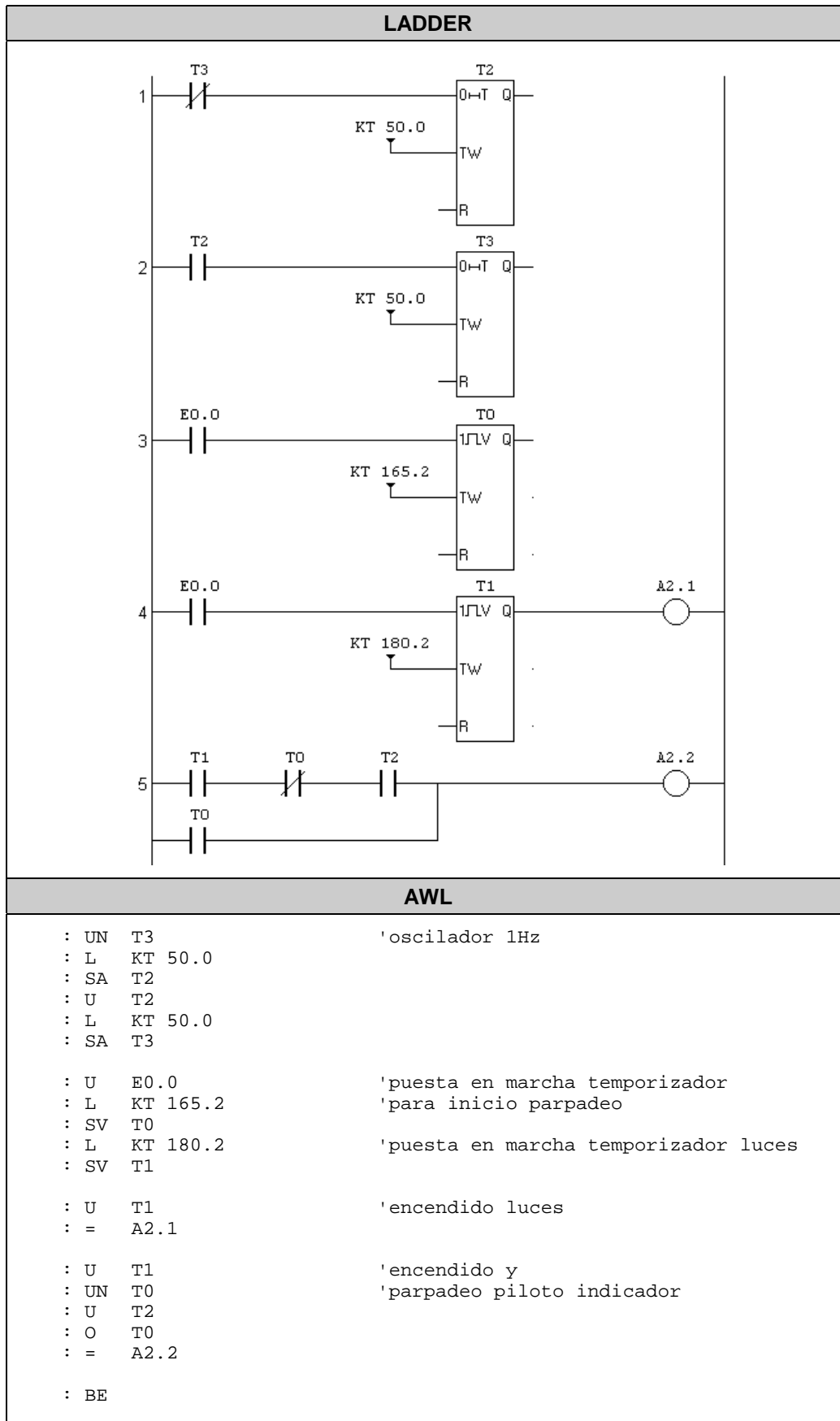
Observando la última línea se observa que el piloto luminoso debe estar encendido o bien cuando está activada $T0$ o, también, cuando están activados $A2.1$ y $T2$ y, simultáneamente $T0$ está desactivado. Es decir, en términos de expresión booleana:

$$A2.2 = T0 + (\overline{T0} \cdot A2.1 \cdot T2)$$

Las soluciones propuestas implementan precisamente lo que se ha descrito.

En la primera, realizada en ladder, se utilizan los dos primeros recorridos para la construcción de la señal de onda cuadrada. Los dos siguientes activan los temporizadores de las luces y el piloto indicador. La última constituye la implementación de la expresión lógica que determina el estado de encendido de esta última.

En el programa AWL, el primer grupo de instrucciones genera la señal de onda cuadrada. El segundo inicia los temporizadores para las luces y el piloto. El tercero enciende las luces y, por último, el cuarto enciende el piloto luminoso según las modalidades determinadas por la expresión que hemos visto antes.



Ejemplo 24: divisor de frecuencia (x4)

Realizar un divisor de frecuencia por 4: a cada cuatro impulsos en la entrada se activa un impulso sobre la salida **A2.1**.

La primera parte de las soluciones propuestas genera un tren de impulsos, tal como hemos aprendido a hacer en los ejemplos anteriores, mostrando la correspondiente señal sobre la salida **A2.0**. La segunda parte implementa efectivamente el divisor. Este está constituido por un contador que va disminuyendo a cada impulso del generador (entrada ZR pilotada por **A2.0**) y, que al llegar a cero, se autoprograma al valor 4 (salida Q traspasada a la entrada S a través de **M0.0**).

A cada cuatro impulsos de **A2.0**, para un solo ciclo de ejecución, el conteo se repone a cero. En este ciclo, la salida del contador se desactiva al igual que el merker **M0.0** conectado a ella. Por el contrario, la salida **A2.1** del PLC se activa, a causa de la negación del contacto de **M0.0** que la controla.

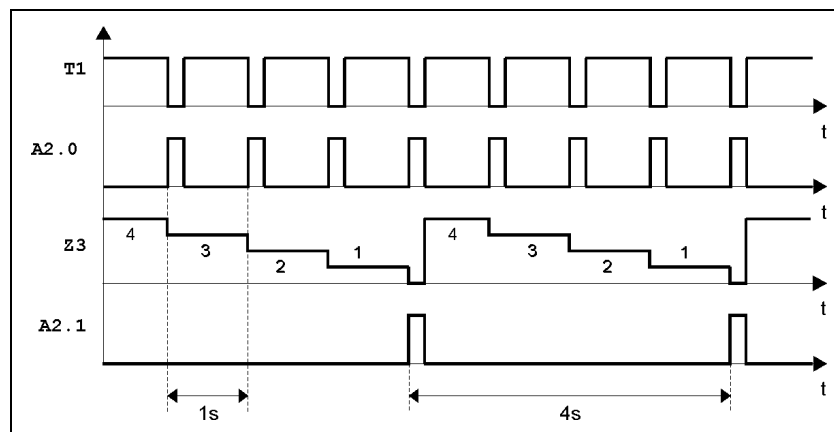
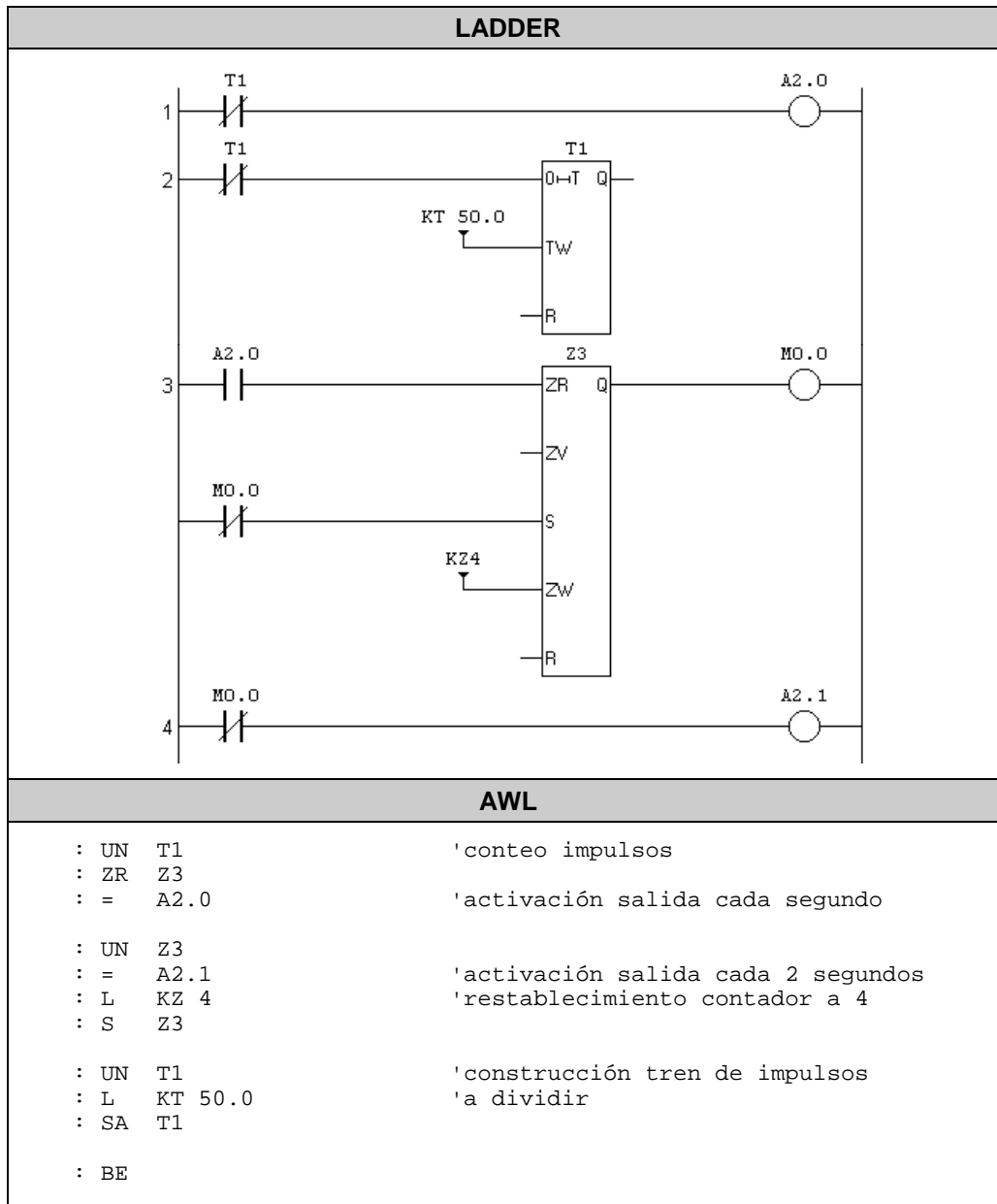


Figura 49: diagrama temporal divisor de frecuencia por 4



Ejemplo 25: conteo de entradas cerradas (I solución)

Contar el número de las entradas cerradas del módulo 0.

Para resolver este ejemplo son necesarias operaciones que pertenezcan al set integrativo (ver [Tabla 19](#)). Por lo tanto, es necesario escribir el programa en un bloque funcional y luego llamarlo desde la **OB1** para permitir su ejecución cíclica. Así pues, no será posible programar en ladder.

Depositaremos en **MB80** el número de las entradas cerradas y utilizaremos **MW10** como puntero en la entrada que se esté examinando. La primera parte del programa en **OB1** consiste en la inicialización a 0 de estas dos variables. La instrucción siguiente llama al bloque **FB4** que contiene el núcleo del programa. Como consecuencia de la instrucción de llamada de bloque incondicionada, la elaboración del programa prosigue desde la primera instrucción de **FB4**.

Las dos primeras instrucciones de este bloque permiten cargar en RLC el complemento de la entrada señalado por **MW10**, con una dirección de canal en el byte alto y dirección de módulo en el byte bajo. Al primer paso, al estar **MW10** a 0, se cargará el complemento del estado de **E0.0**.

La instrucción siguiente es un salto condicionado: si $RLC=1$, es decir, si el complemento de **E0.0** es igual a 1, o sea, si la entrada está abierta, la elaboración continúa en la etiqueta **INC**. En cambio, si la entrada está cerrada, el salto no se efectúa y la elaboración continúa con la instrucción siguiente.

En este último caso, las cuatro instrucciones siguientes incrementan el valor de **MB80**, es decir, del número que representa el conteo de entradas que se han encontrado cerradas, cargando primero el valor en **ACCU1**, y, a continuación, sumándole 1. Por último, transfiere el resultado de nuevo a **MB80**. La misión global estas instrucciones y de las anteriores es la de incrementar la variable **MB80** si la entrada está cerrada y de no incrementarla si la entrada está abierta.

En un caso o en el otro, la elaboración llega al grupo de instrucciones identificado por la etiqueta **INC** que se encarga, como primera acción, de incrementar el byte alto de **MW10**, es decir, **MB10**, que es el byte que representa el número del canal de la entrada en la operación de carga indexada que hemos visto antes (con una finalidad didáctica, hemos utilizado una operación distinta para obtener el incremento del byte). Luego, se compara el valor recién obtenido con el valor 7. Si es menor o igual, la entrada señalada existe y debemos valorar su estado volviendo a la etiqueta **TEST**, tal como especifica la instrucción de salto condicionado. Si no, en el caso de que el valor sea 8, el canal no existe y debemos salir del bloque, tras haber examinado las ocho entradas posibles, de **E0.0** a **E0.7**.

Así pues, la parte de programa que va de la etiqueta **TEST** a la instrucción **SPB =TEST** se ejecuta ocho veces, antes de regresar al bloque que hace la llamada **OB1** por medio de la última instrucción **BE**. Cada vez, **MW10** contará un valor distinto: uno tras otro y en hexadecimal 0000, 0100, 0200, 0300, 0400, 0500, 0600, 0700; y cada vez, el grupo de instrucciones **B MW10** y **UN E0.0** cargará en RLC el complemento de una entrada del módulo distinta, desde la primera hasta la última.

La parte de programa que va de **L MB80** hasta **T MB80** tan solo se elabora si la entrada que está comprobando está cerrada, con el resultado de la ejecución de un incremento de **MB80** únicamente si se da

esta condición y, puesto que el valor inicial de este merker byte es igual a 0, al final de la ejecución del bloque, éste contendrá efectivamente el número de las entradas cerradas.

AWL		
OB1		
<pre> : L KF +0 : T MB80 : T MW10 : SPA FB4 () : BE </pre>	<pre> 'programaciones iniciales </pre>	
FB4		
<pre> TEST: B MW10 : UN E0.0 : SPB =INC : L MB80 : L KF +1 : +F : T MB80 INC : L MB10 : I 1 : T MB10 : L KF +7 : <=F : SPB =TEST : BE </pre>	<pre> 'indexación entrada 'la entrada está activa: 'incremento del merker byte 'de conteo 'incremento índice para 'examen entrada siguiente 'control si ha acabado </pre>	

Si desea analizar las entradas cerradas del módulo 1 bastará con cargar 1 como valor de inicialización para **MW10**. También su puede valorar el total de las entradas cerradas para los dos módulos. Se trata de llamar dos veces al **FB4**, la primera, inicializando el **MW10** a 0 y la segunda inicializándolo a 1. En cambio, para **MB80** será precisa una sola inicialización a 0 al inicio de la **OB1**. Dejamos al lector la realización de este programa.

Ejemplo 26: conteo de entradas cerradas (II solución)

Contar el número de las entradas cerradas del módulo 0.

En este ejercicio, con el fin de utilizar una operación de desplazamiento, afrontamos una solución diferente a la que se ha programado en el ejemplo anterior.

El contador de entradas cerradas sigue siendo **MB80** y su valor también se ha inicializado a 0 al inicio de la **OB1**. En cambio, **MB10** representa una máscara de 8 bit, uno solo de los cuales estará, por turno, a 1. Su valor inicial es 1, es decir $(00000001)_2$, donde sólo el bit 0 es cierto.

Las tres primeras instrucciones del bloque **FB4** cargan en los acumuladores los valores de la máscara y del byte de entrada correspondiente al módulo 0 y ejecutan la AND bit a bit. Al primer paso, con el valor de la máscara recién visto, la AND da un resultado distinto de 0 tan sólo si **E0.0** está cerrada. Así, el salto sobre cero previsto por la instrucción siguiente, únicamente se efectúa para una entrada abierta.

Si la entrada está cerrada, tal como hemos hecho en el ejemplo anterior, incrementamos el byte de conteo **MB80**.

En un caso o en el otro, los dos recorridos de elaboración se reúnen en la etiqueta **INC** en la que, después de haber cargado la máscara en ACCU1, se gestiona el desplazamiento de una posición hacia la izquierda. El resultado del desplazamiento, o mejor dicho, la parte baja de éste, se transfiere a **MB10**.

Tras la instrucción **L KF+256**, ACCU1 contendrá el valor 256 y ACCU2 contendrá el resultado del desplazamiento. Si el bit de máscara, después de 8 desplazamientos, ha finalizado en posición 8 en el registro de 16 bit del acumulador, el valor de esta última será $2^8=256$ y entonces la instrucción de salto condicionado se ignora y la elaboración del bloque termina. Si el bit ocupa posiciones inferiores, y por lo tanto el contenido del acumulador resulta menor de 256, el salto a la etiqueta **TEST** se ejecuta para repetir la elaboración con el fin de examinar la entrada siguiente.

AWL		
OB1		
: L	KF +0	'inicialización merker byte
: T	MB80	'de conteo
: L	KF +1	'inicialización máscara
: T	MB10	
: SPA	FB4 ()	
: BE		
FB4		
TEST: L	MB10	'control de la entrada
: L	EB0	'no enmascarada
: UW		
: SPZ	=INC	
: L	MB80	'la entrada está activa:
: L	KF +1	'incremento del merker byte
: +F		'de conteo
: T	MB80	
INC : L	MB10	'shift de la máscara para
: SLW	1	'examen entrada siguiente
: T	MB10	
: L	KF +256	'control si ha terminado
: <F		
: SPB	=TEST	
: BE		

Ejemplo 27: semáforo para Fórmula 1

Al activarse el pulsador conectado a la entrada $E0.0$, las cinco luces un semáforo deben encenderse una tras otra, a cada segundo. Un segundo después del encendido completo, las luces deberán apagarse.

Para programar la solución de este problema se han usado símbolos. Su correspondencia con los operandos absolutos se ha establecido según la siguiente tabla.

Op. absoluto	Símbolo	Comentario
$E0.0$	START	Pulsador de puesta en marcha
$A2.0$	L1	Luz 1
$A2.1$	L2	Luz 2
$A2.2$	L3	Luz 3
$A2.3$	L4	Luz 4
$A2.4$	L5	Luz 5

El diagrama temporal de [Figura 50](#) muestra, en las líneas intermedias, el desarrollo de las salidas del PLC que controlan las luces del semáforo en función de la entrada **START**, ilustrada en la primera línea.

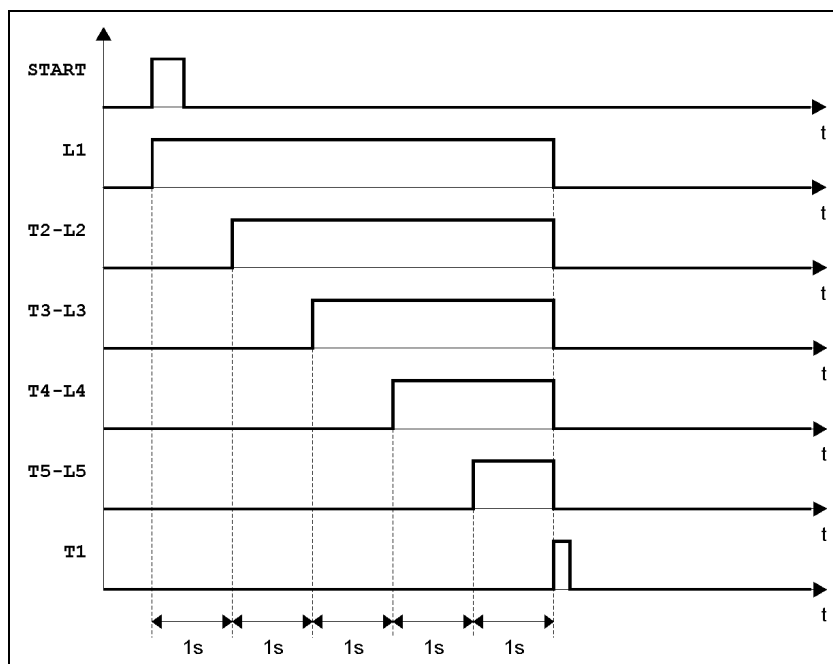
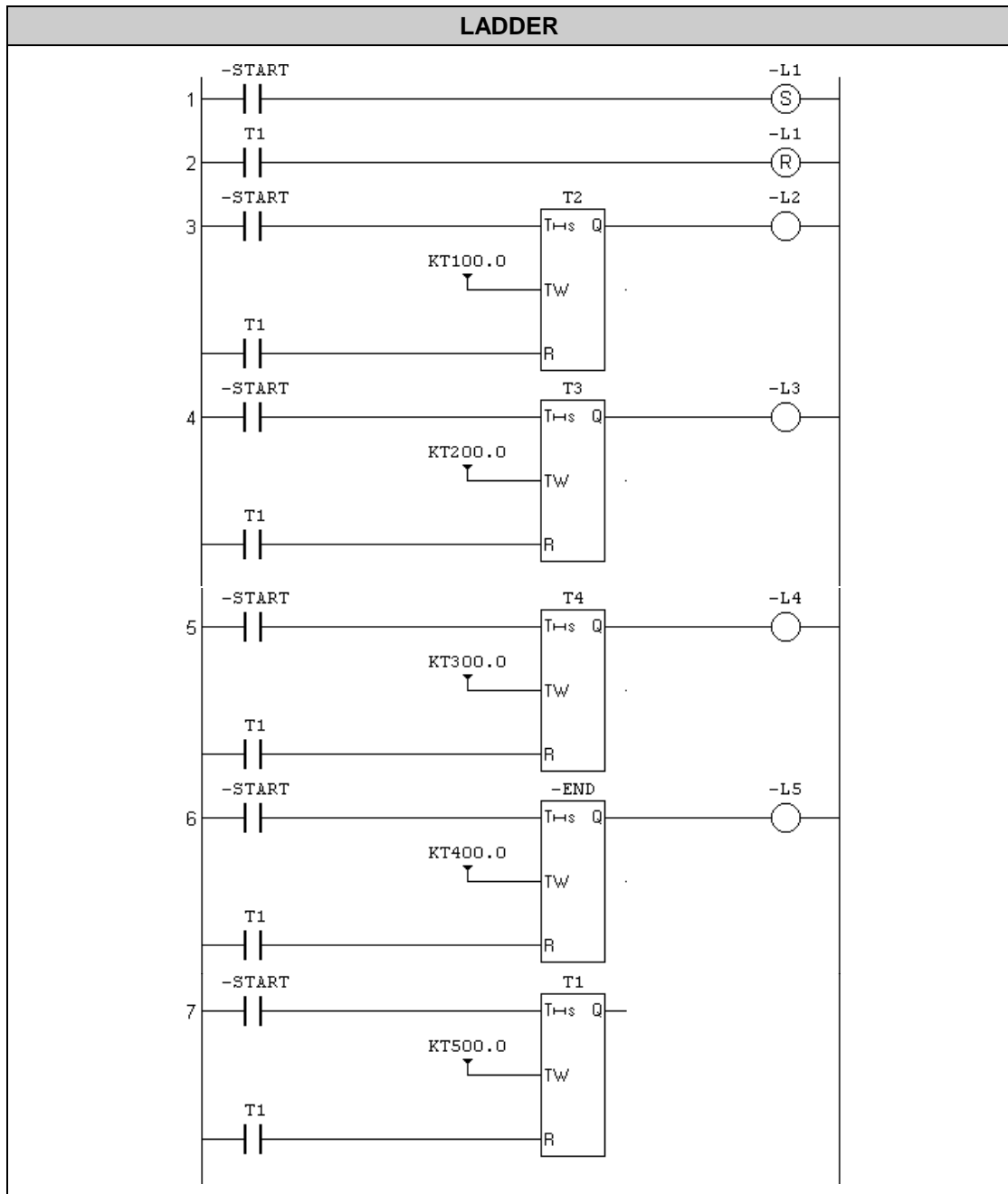


Figura 50: diagrama temporal para semáforo de Fórmula 1

El cierre de esta entrada, además de programar la salida que controla a la primera luz, determinando su encendido inmediato, activa cinco temporizadores con retardo a la activación con memoria, programados con tiempos de 1 a 5 segundos. La salida de cada uno de ellos, a excepción de **T1**, una vez transcurrido el tiempo programado, se llevará al estado alto y activará la correspondiente luz, obteniendo así, con toda facilidad, la secuencia de encendido. El temporizador **T1**, en cambio, se encarga del apagado de todas las luces restaurando **L1** y todos los temporizadores, comprendido él mismo.

Ponga en Run el PLC, trasforme el interruptor 0.1 en un pulsador, acciónelo y ... ¡que gane el mejor !



```

AWL

: U  -START          'programación temporizadores para la
: L  KT 100.0        'secuencia
: SS T2
: L  KT 200.0
: SS T3
: L  KT 300.0
: SS T4
: L  KT 400.0
: SS T5
: L  KT 500.0
: SS T1

: U  -START          'programación salidas
: S  -L1
: U  T2
: =  -L2
: U  T3
: =  -L3
: U  T4
: =  -L4
: U  T5
: =  -L5

: U  T1              'reset temporizadores
: R  -L1
: R  T1
: R  T2
: R  T3
: R  T4
: R  T5

: BE
    
```

Ejemplo 28: luces secuenciales con 4 canales

Construir un secuenciador para 4 canales que prevea el siguiente esquema de encendido.

	P	0	1	2	3	4	5
C		0	1	2	3	4	5
0		●	○	○	○	○	○
1		○	●	○	○	○	●
2		○	○	●	○	●	○
3		○	○	○	●	○	○

Figura 51: esquema de encendido para secuenciador para 4 canales

El esquema establece la secuencia de encendido de las luces conectadas a cuatro canales. Los círculos negros indican la activación del canal durante su paso específico. Así, durante el paso 0 será activo el canal 0, durante el paso 1 el canal 1, y así sucesivamente. Si las luces se colocan en línea, el efecto será el de un desplazamiento de la fuente luminosa desde la primera hasta la última posición y luego al contrario.

Al paso 5 le sucede un paso 6 idéntico al 0 y luego otro idéntico al paso 1, es decir, el diagrama se va recorriendo cíclicamente. Imaginen que lo recortan y lo enroscan formando un cilindro y hacen coincidir los

límites opuestos del paso 0 y del 5, sería algo similar al tambor de un carillón: cuando acaba de tocar su musiquilla, vuelve a empezar.

Continuando con el símil sonoro, para que un carillón funcione necesita un cilindro con unas levas dispuestas de un modo adecuado sobre su superficie lateral y un mecanismo que lo haga girar.

Empezaremos por construir este último. El mecanismo de avance de nuestro secuenciador será un tren de impulsos con un período de 0,2 segundos, es decir, una base de tiempos con un período elegido de manera que se obtenga el desplazamiento de la fuente lumínica que se considere el adecuado. Las instrucciones

```
UN  -TIMER
L   KT20.0
SA  -TIMER
```

construyen un tren de impulsos, tal como hemos aprendido a hacerlo antes. Por su parte, el cilindro estará constituido por un contador que, partiendo de 0, se va incrementando a cada impulso.

```
UN  -TIMER
ZV  -COUNTER
```

Al llegar a 6, es decir, después del último paso, deberá reprogramarse al valor de partida, 0, y, precisamente, el paso 6 será idéntico al paso 0.

```
L   -COUNTER
L   KF+6
! =F
R   -COUNTER
```

Antes de disponer las levas sobre el cilindro, identificamos los “gajos” del cilindro que corresponden a cada paso específico. Las instrucciones:

```
L   -COUNTER
L   KF+0
! =F
=   -PASO0
```

programan a 1 el merker **PASO0** cuando el contador vale 0, identificando entre los posibles valores del contador el que corresponde al citado paso. Los siguientes grupos de instrucciones programan los merker que corresponden a cada uno de los otros pasos. Así, al final, cada 0,2 s será activo un merker distinto, desde **PASO0** a **PASO5** uno tras otro y, a continuación, vuelta a empezar desde **PASO0**.

Ahora que hemos identificado las posiciones, podemos insertar las levas. Empezamos por el canal 0, observamos nuevamente la parrilla de [Figura 51](#), el canal 0 está activado solo durante el paso 0, entonces:

```
O   -PASO0
=   -CH0
```

El canal 1 debe estar activo tanto durante el paso 1 como durante el paso 5:

```
O   -PASO1
O   -PASO5
=   CH1
```

Proseguimos así para los otros dos canales hasta concluir el carillón o, dejando ya a un lado el símil didáctico, el secuenciador.

AWL		
: UN	T3	'avance paso
: ZV	Z5	
: UN	T3	'construcción base de tiempos
: L	KT 10.0	
: SA	T3	
: L	Z5	'reinicialización ciclo
: L	KF +6	
: !=F		
: R	Z5	
: L	Z5	'test fase 0
: L	KF +0	
: !=F		
: =	M0.0	
: L	Z5	'test fase 1
: L	KF +1	
: !=F		
: =	M0.1	
: L	Z5	'test fase 2
: L	KF +2	
: !=F		
: =	M0.2	
: L	Z5	'test fase 3
: L	KF +3	
: !=F		
: =	M0.3	
: L	Z5	'test fase 4
: L	KF +4	
: !=F		
: =	M0.4	
: L	Z5	'test fase 5
: L	KF +5	
: !=F		
: =	M0.5	
: O	M0.0	'programación canal 0
: =	A2.0	
: O	M0.1	'programación canal 1
: O	M0.5	
: =	A2.1	
: O	M0.2	'programación canal 2
: O	M0.4	
: =	A2.2	
: O	M0.3	'programación canal 3
: =	A2.3	
: BE		

En este ejemplo se puede aumentar o disminuir la duración de los pasos simplemente cambiando la constante con la que se carga el temporizador, produciendo así el efecto de variar la velocidad del desplazamiento aparente de la fuente luminosa. Se puede modificar el número de pasos, cambiando la constante con la cual se carga el contador y añadiendo otros controles para discriminar los nuevos pasos

añadidos. Por último, también se puede cambiar la secuencia de encendido de las luces, modificando las condiciones en los grupos de OR que constituyen la última parte del programa.

A buen seguro, la que hemos presentado no es la única solución al problema y, por lo general, a medida que éste se hace más complejo, las soluciones posibles aumentan. Probablemente ésta no es ni siquiera la mejor en lo que a versatilidad, simplicidad de mantenimiento o elegancia de programación se refiere, si bien en el próximo ejemplo propondremos una solución que responde mejor a estos requisitos. Pero es la aproximación más sencilla y que utiliza las instrucciones más comunes que hemos logrado construir: éste es un requisito fundamental para el propósito que este capítulo del manual se ha propuesto: acompañarle en sus primeros pasos en el mundo de la programación de los PLC.

Ejemplo 29: luces secuenciales en barra

Construir un secuenciador de 8 canales que prevea el siguiente esquema de encendido.

P \ C	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○
1	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○
2	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●
3	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○
4	○	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○
5	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○
6	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●	●	●	●
7	○	○	○	○	○	○	○	○	●	●	●	●	●	●	●	●	●	●	●

P \ C	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	○	○	○	●	○	○	○	○	●	○	○	○	●	○	○	●	○	●
1	○	○	○	○	●	○	○	○	○	●	○	○	○	●	○	○	●	●
2	○	○	○	○	○	●	○	○	○	○	●	○	○	○	●	●	●	●
3	●	○	○	○	○	○	●	○	○	○	○	●	●	●	●	●	●	●
4	○	●	○	○	○	○	○	●	●	●	●	●	●	●	●	●	●	●
5	○	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
6	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
7	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

El efecto óptico, si las luces están dispuestas en vertical, tal como están los pilotos luminosos del módulo del PLC, es el de un llenado progresivo de la barra debido a fuentes luminosas que aparecen en la parte alta y que poco a poco descienden hasta ocupar la última posición libre, es decir, apagada. Una vez iluminada toda la barra, el ciclo vuelve a empezar, apagando las luces y reiniciando el llenado.

La secuencia es lo bastante más complicada que la anterior como para empujarnos a buscar una solución distinta, más funcional y, tal vez, más versátil. La prevista utiliza un bloque de datos para memorizar las combinaciones de encendido de la secuencia. Cada data word contiene, en el byte bajo, el código de encendido de las luces. La última word contiene el dato (FFFF)₁₆ que actúa como finalizador de secuencia con las modalidades que veremos dentro de poco.

El corazón del programa está constituido por el bloque **FB10** que se reclama cada 2 décimas de segundo y que se encarga de leer los datos y de enviarlos al módulo de salida. Veamos como.

MW100 constituye el puntero a la data word en vigor. Su valor inicial es 0. El grupo de instrucciones en la etiqueta **READ**, siguiente a la apertura del **DB20**, carga en primer lugar el valor (FFFF)₁₆ en el acumulador. A continuación, el valor de la data word en vigor, utiliza una instrucción indexada, y compara los dos valores: si son distintos, la elaboración continua transfiriendo el contenido de ACCU1, el dato de la secuencia, al módulo 2 de salida. A continuación, después de incrementarse el valor del puntero, se retorna al bloque que hace la llamada de manera que, cuando se llame nuevamente a **FB4**, la data word cargada será la siguiente. Si, por el contrario, el valor leído es igual a (FFFF)₁₆, la elaboración continúa en primer lugar, en la etiqueta **INIT** donde se efectúa la reposición a cero del índice para el retorno al inicio de la secuencia, y luego, en la etiqueta **READ**, para la nueva lectura de la primera combinación de la secuencia, **DW0**.

AWL	
OB1 (Principal)	
: UN T0 : SPB FB10 ()	'llamada tiempo del FB10
: UN T0 : L KT 20.0 : SA T0 : BE	'base de tiempos de 0.2 segundos
OB21 (Inicialización sobre STOP-RUN)	
: L KF +0 : T MW100 : BE	'reinicialización índice
OB22 (Inicialización sobre OFF-ON)	
: L KF +0 : T MW100 : BE	'reinicialización índice

AWL			
FB10 (Encendido luces)			
	: A	DB20	'apertura bloque de datos
READ:	L	KH FFFF	'carga código fin de ciclo
	: B	MW100	'lectura del dato en vigor
	: L	DW0	'y control si es fin de ciclo
	: !=F		
	: SPB	=INIT	
	: T	AB2	'encendido luces
	: L	MW100	'incremento del índice
	: L	KF +1	
	: +F		
	: T	MW100	
	: BEA		
INIT:	L	KF +0	'reinicialización índice
	: T	MW100	
	: SPA	=READ	
	: BE		
DB20 (Datos para la secuencia)			
0	KM	00000000 00000000	Datos del ciclo
1	KM	00000000 00000001	
2	KM	00000000 00000010	
3	KM	00000000 00000100	
4	KM	00000000 00001000	
5	KM	00000000 00010000	
6	KM	00000000 00100000	
7	KM	00000000 01000000	
8	KM	00000000 10000000	
9	KM	00000000 10000001	
10	KM	00000000 10000010	
11	KM	00000000 10000100	
12	KM	00000000 10001000	
13	KM	00000000 10010000	
14	KM	00000000 10100000	
15	KM	00000000 11000000	
16	KM	00000000 11000001	
17	KM	00000000 11000010	
18	KM	00000000 11000100	
19	KM	00000000 11001000	
20	KM	00000000 11010000	
21	KM	00000000 11100000	
22	KM	00000000 11100001	
23	KM	00000000 11100010	
24	KM	00000000 11100100	
25	KM	00000000 11101000	
26	KM	00000000 11110000	
27	KM	00000000 11110001	
28	KM	00000000 11110010	
29	KM	00000000 11110100	
30	KM	00000000 11111000	
31	KM	00000000 11111001	
32	KM	00000000 11111010	
33	KM	00000000 11111100	
34	KM	00000000 11111101	
35	KM	00000000 11111110	
36	KM	00000000 11111111	
37	KH	FFFF	Finalizador de ciclo

APÉNDICE A LA INTERFACE EASYPORT D16

Generalidades

La interface EasyPort D16 permite la transmisión bidireccional de señales ON/OFF entre un dispositivo o una instalación real y el PC, usando señales de baja tensión (24VCC).

La conexión entre la EasyPort D16 y el PC se realiza mediante una conexión serie. Otras EasyPort D16, hasta siete, pueden conectarse a la primera mediante una interconexión con un anillo de fibra óptica. La dirección de módulo se asigna automáticamente, en función de su posición en el anillo. El módulo conectado al PC mediante una conexión RS232 tendrá la dirección 1; el siguiente, es decir, aquel cuyo conector IN está unido con el conector OUT del primero, asumirá la dirección 2 y así sucesivamente hasta el último módulo, que asumirá una dirección igual al número total de EasyPort D16 interconectadas.



Anillo de fibra óptica

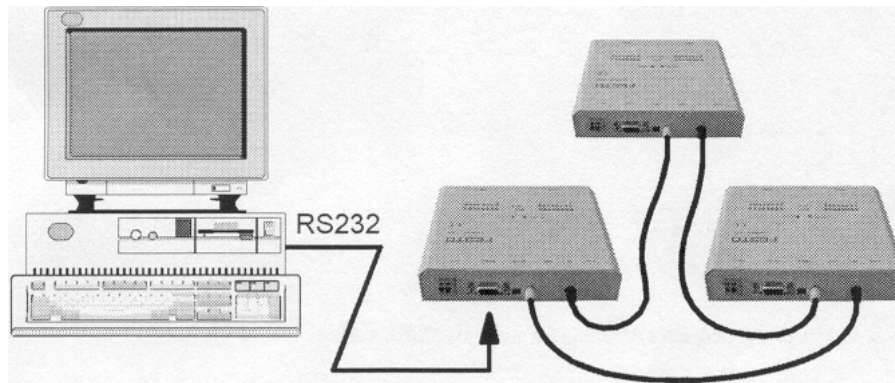
La conexión opcional de los módulos de expansión se realiza por medio de fibras ópticas. Con tal fin, están previstos dos conectores en el panel posterior de la unidad.

- Conector blanco = RING OUT
- Conector negro = RING IN

El conector blanco debe empalmarse con el conector negro del módulo siguiente. El conector blanco del último módulo deberá unirse al conector negro del primero, con el fin de cerrar el anillo.

Para efectuar la conexión, inserte, sin forzar, el extremo de la fibra en el conector y, a continuación, fije este último atornillándolo. Ejecute las mismas operaciones en el orden inverso para deshacer la conexión.

La longitud de cada tramo de fibra debe estar comprendida entre 0,3 y 5 metros.



Función de los pilotos luminosos

La parte superior de la EasyPort D16 incorpora numerosos pilotos luminosos, indicadores de estado.

Piloto luminoso SHORT

El encendido del piloto rojo SHORT indica que la parte electrónica del sistema ha detectado un cortocircuito en una de las salidas. En esta situación, todas las salidas se apagan. Durante la fase de encendido de la interface este piloto se enciende un tiempo breve.

Piloto luminoso STATUS

El piloto verde denominado STATUS facilita dos tipos de informaciones en función de la modalidad de encendido:

- Parpadeando a 1 Hz Estado que sigue al encendido, el módulo aun no está comunicando
- Pulsante El módulo se ha direccionado. El número correlativo de la dirección se visualiza cada dos segundos, mediante el apagado del piloto luminoso durante unos instantes. Por ejemplo, si el módulo tiene dirección tres, cada dos segundos, el piloto STATUS se apagará brevemente tres veces.

Piloto luminoso INPUT

Las entradas de la EasyPort están organizadas en dos grupos de ocho. El primer grupo forma parte de las señales de PORT 1 y el segundo, forma parte de las señales de PORT2. Las entradas de PORT 1 están numeradas de 0 a 7 y las de PORT 2, de 8 a 15.

El estado de cada una de las entradas se visualiza mediante el estado del piloto luminoso correspondiente: apagado para el estado 0, es decir, tensión de 0V en la entrada; encendido para el estado 1, es decir, tensión de 24V a la entrada.

Piloto luminoso OUTPUT

El estado de las 16 salidas digitales se muestra por medio de los pilotos amarillos denominados OUTPUT. La agrupación y la numeración de las salidas son análogas a las de las entradas.

El estado de cada una de las salidas se visualiza a través del piloto correspondiente: apagado para el estado 0 es decir, tensión de 0V en la salida y encendido para el estado 1, es decir, tensión de 24V en la salida.

Recomendaciones

Preste mucha atención al hecho de que ni la tensión de alimentación ni la tensión en las entradas pueden superar el valor de 30V.

Efectúe o elimine las conexiones tan sólo cuando no haya tensión de alimentación.

Configuración

Seleccione la configuración para las comunicaciones con el DIP switch de 3 pins, según las indicaciones de la tabla siguiente. Tan sólo puede estar en ON un switch (posición en bajo) y los otros dos deben estar en OFF (posición en alto).

Tabla 22: Configuración de los DIP switch

ON	Interface activa	Configuración
1	Solo interface RS232	Solo un módulo está conectado al PC a través del puerto de serie RS232
2	Interface RS232 y fibra óptica	El módulo está conectado al PC a través del puerto de serie RS232, otros módulos están conectados a éste a través de la fibra óptica.
3	Solo fibra óptica	El módulo es un módulo de expansión en el anillo de fibra óptica.

La primera configuración es la que debe utilizar si desea conectar una sola EasyPort D16 a su PC. En cambio, la segunda es la configuración que deberá programar para la EasyPort D16 que se va a conectar al PC, en el caso de conexión de varios módulos de anillo de fibra óptica. Los restantes módulos se programarán siguiendo la tercera configuración.

Identificación de los conductores

Cada uno de los dos cables multipolares que se proporcionan con la EasyPort D16 está compuesto por 24 conductores. Al final, 6 de estos están conectados entre si de dos en dos. Así pues, cada cable lleva 21 señales distintas. De estos, 16 se utilizan para las entradas y las salidas, y los restantes están reservados para la alimentación.

La tabla siguiente permite la identificación de los conductores en función de su color y del puerto a que pertenecen.

Tabla 23: Identificación de los conductores de los cables multipolares

PORT1	PORT2	PIN	Color cable
OUTPUT 0	OUTPUT 8	1	Blanco
OUTPUT 1	OUTPUT 9	2	Marrón
OUTPUT 2	OUTPUT 10	3	Verde
OUTPUT 3	OUTPUT 11	4	Amarillo
OUTPUT 4	OUTPUT 12	5	Gris
OUTPUT 5	OUTPUT 13	6	Rosa
OUTPUT 6	OUTPUT 14	7	Azul
OUTPUT 7	OUTPUT 15	8	Rojo
INPUT 0	INPUT 8	13	Gris/Rojo
INPUT 1	INPUT 9	14	Rojo/Azul
INPUT 2	INPUT 10	15	Blanco/Verde
INPUT 3	INPUT 11	16	Marrón/Verde
INPUT 4	INPUT 12	17	Blanco/Amarillo
INPUT 5	INPUT 13	18	Amarillo/Marrón
INPUT 6	INPUT 14	19	Blanco/Gris
INPUT 7	INPUT 15	20	Gris/Marrón
OVCC	OVCC	11	Rosa/Marrón
OVCC	OVCC	12	Violeta
OVCC	OVCC	23/24	Blanco/Azul
24VCC	24VCC	9/10	Negro
24VCC	24VCC	21/22	Blanco/Rojo

Datos técnicos

La siguiente tabla relaciona los datos técnicos de la interface EasyPort D16.

Tabla 24: Datos técnicos de la EasyPort D16

Tensión de alimentación	24 VCC \pm 10%
Consumo	3 VA
Número de salidas	16 digitales
Corriente max	0,3 A por salida, incluso las simultaneas
Protección cortocircuito	si
Número de entradas	16 digitales
Umbral	12 VCC
Histéresis	3 V
Filtrado	10 ms
Interface de comunicación	V.24 aislada galvánicamente
Longitud conexiones de fibra óptica	0.3 – 5 m cada uno
Protocolo	ASCII, 19.2 Kbaud
Protección	IP20
Resistencia a perturbaciones DIN/IEC 801/4	Clase 4
Temperatura de funcionamiento/almacenaje	0 - 55 °C / 0-70 °C
Dimensiones (LxPxH)	162x148x36 m
Peso	0.65 Kg

