

**Manual de formación
para soluciones generales en automatización
Totally Integrated Automation (T I A)**

MÓDULO D2

**AS-Interface/integración de un módulo lógico
LOGO!12/24RC e interconexión de AS-I**

Este documento fue suministrado por SIEMENS Siemens A&D SCE (Tecnología en Automatización y Accionamientos, Siemens A&D, coopera con la Educación) para formación. Siemens no hace ningún tipo de garantía con respecto a su contenido.

El préstamo o copia de este documento, incluyendo el uso e informe de su contenido, sólo se permite dentro de los centros de formación.

En caso de excepciones se requiere el permiso por escrito de Siemens A&D SCE (Mr. Kunst: E-Mail: michael.knust@hvr.siemens.de). Cualquier incumplimiento de estas normas estará sujeto al pago de los posibles perjuicios causados. Todos los derechos quedan reservados para la traducción y posibilidad de patente.

		PÁGINA:
1.	Inicio	5
2.	Notas sobre la operación de LOGO! Módulos Lógicos	7
3.	Primeros pasos con LOGO!	8
3.1	Terminales	8
3.2	Terminales desde LOGO!.....	8
3.3	LOGO! Conozca los siguientes terminales	9
3.4	Bloques y números de bloque	9
3.5	Bloques	9
3.6	Operaciones lógicas.....	9
3.7	Representación de bloques en la pantalla de LOGO!	10
3.8	Asignación de un número de bloque	10
3.9	Las 4 reglas de oro para la puesta en marcha de LOGO!	11
3.10	Visión global de los menús de LOGO!.....	12
4.	Ejercicio ejemplo: Control de una Puerta Corredera	13
4.1	Requisitos para el control de la puerta	13
4.2	Cableando el control de la puerta con LOGO!12/24RC.....	14
4.3	Componentes utilizados	14
4.4	Unidad Central y monitorización de más puertas industriales	15
4.5	Control de capas con AS-Interface.....	15
4.6	Diagrama de Bloques de Función con una solución en LOGO!	16
5.	Introduciendo el programa en LOGO!	17
5.1	Cambiando al modo de operación de programación	17
5.2	LOGO! Cambios en el menú de programas.....	17
5.3	Introduciendo un programa.....	18
5.4	Parametrizando un bloque	19
5.5	LOGO! Conmutando a RUN	22

		PAGE:
6.	Programación del SIMATIC S7-300.....	23
6.1	Generando un nuevo proyecto.....	23
6.2	Introduciendo la configuración hardware	25
6.3	Manejando la puerta corredera a través del AS-Interfase.....	26
6.4	Control del programa de la puerta corredera.....	27
6.5	Introduciendo bloque de organización OB1	28

Los símbolos siguientes acceden a los módulos especificados:



Información



Programación



Ejercicio ejemplo

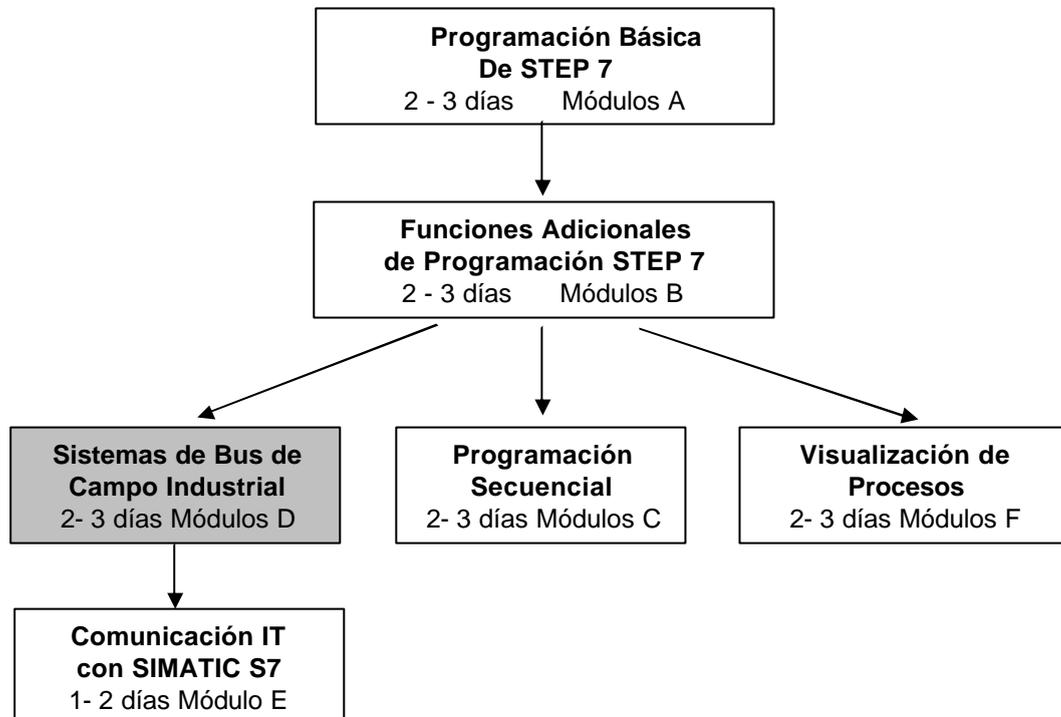


Notas

1. INICIO



El módulo D2 pertenece al contenido de **Sistemas de Bus de Campo Industrial** .



Finalidad del Aprendizaje:

En este módulo, el lector aprenderá sobre las funciones básicas del LOGO!, módulo lógico 24RCLB11 con interfase ASI integrado.

Las tareas de posicionamiento típicas se ejecutan bajo una máquina patrón y se ejecuta un proyecto ejemplo, a través de los siguientes pasos:

- Generación de un programa para el LOGO!12/24RC.
- Depuración de tareas de posicionamiento en modo RUN del LOGO!
- Generación de un proyecto para un PLC SIMATIC S7-300
- Integración de información ASI desde el LOGO!12/24RC en el programa de control de la CPU SIMATIC S7-300.
- Depuración de la tarea de posicionamiento con un PLC SIMATIC S7- 300 y el LOGO!12/24RC

Requisitos:

Para el correcto aprovechamiento de este módulo, se requieren los siguientes conocimientos:

- Conocimientos de uso de Windows 95/98/2000/ME/NT4.0
- Programación Básica de PLC con STEP 7 (Módulo A3 - 'Puesta en Marcha' programando PLC con STEP 7)
- Puesta en marcha del AS-Interfase con SIMATIC S7-300 (p.e. Módulo D1 – AS- Interfase con un SIMATIC S7-300 y una CP342-2)

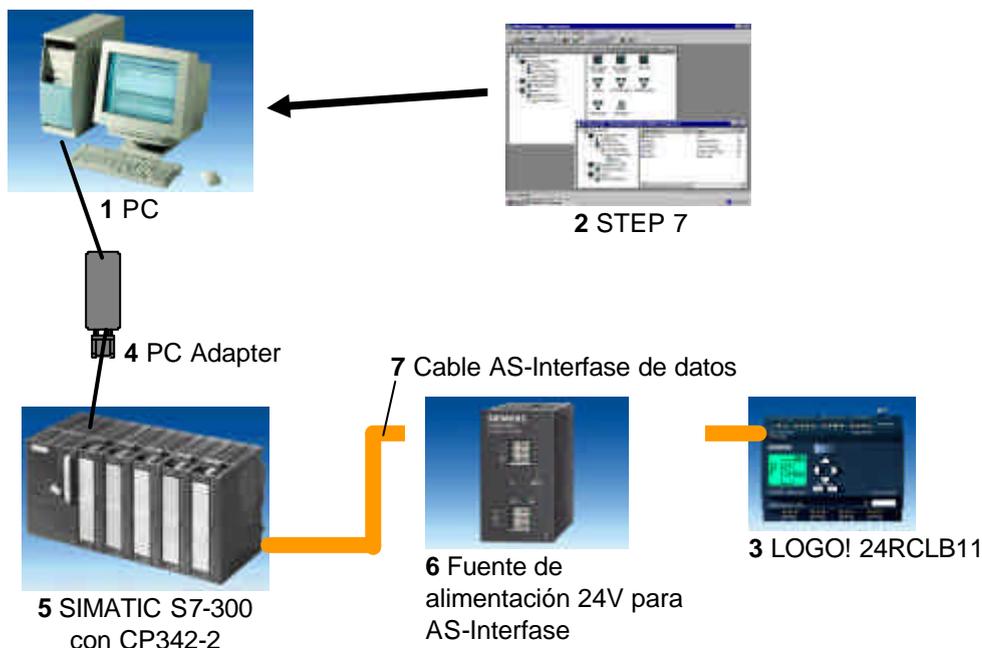


Hardware y software Necesarios

- 1 PC, Sistema Operativo Windows 95/98/2000/ME/NT4.0 con
 - Mínimo: 133MHz y 64MB RAM, aprox. 65 MB de espacio libre en disco duro
 - Óptimo: 500MHz y 128MB RAM, aprox. 65 MB de espacio libre en disco duro
- 2 Software STEP 7 V 5.x
- 3 LOGO!12/24RC
- 4 MPI- Interfase para PC (p.e. PC- Adapter)
- 5 PLC SIMATIC S7-300

Ejemplo de Configuración:

 - Fuente de Alimentación: PS 307 2A
 - CPU: CPU 314
 - Entradas Digitales: DE 16x DC24V
 - Salidas Digitales: DA 16x DC24V / 0.5 A
 - CP 342-2 AS-Interfase
- 6 Fuente de Alimentación de 30V para AS-Interfase
- 7 Cable Amarillo de datos AS-Interfase



2. NOTAS SOBRE LA OPERACIÓN DE LOGO! MÓDULOS LÓGICOS



LOGO! Es el módulo lógico universal de Siemens.

En LOGO!, el control se encuentra integrado con las unidades de visualización y puesta en marcha. Con estas unidades se puede, desde LOGO!, generar programas, así como editar y ejecutar funciones del sistema.

Se puede leer programas externos desde un módulo programable o desde un PC con el software de programación LOGO!-SOFT. Además de poder programar el software, se puede también ejecutar una simulación de su circuito en el ordenador o imprimir los esquemas eléctricos.

LOGO! dispone de funciones básicas ya creadas, que pueden utilizarse para generar los circuitos (p.e. retardar la conmutación de un relé, relé de intensidad, interruptor por tiempo, marcas de memoria, así como entradas y salidas).



Se puede solucionar tareas con LOGO! :

- En el hogar – y tecnología de instalaciones (p.e. iluminación de las escaleras de una casa, luces exteriores, cerraduras, persianas, luces de escaparates, etc.),
- En un fabricante de armarios eléctricos y en máquinas – y constructoras de otros tipos de aparatos (p.e. control de puertas, sistemas de ventilación, bombas de agua industriales, etc.).

Es más, LOGO! Puede ser configurado para control especial de preproceso de señales. A través de la variante AS-i, es posible aplicar la periferia distribuida con inteligencia separada para localizar el control de máquinas y otros procesadores. Con esto, las tareas de control pueden llevarse a cabo con el módulo lógico LOGO! Para liberar el control maestro.

Existen variantes especiales sin unidades de servicio para usos serios en pequeñas máquinas y aparatos de construcción, en un fabricante de armarios eléctricos y otros ámbitos de instalación. Estos deben de ser cargados en un módulo programable a través del software de PC LOGO-SOFT.

3. PRIMEROS PASOS CON LOGO!



Indicamos las entradas de un circuito a través de la programación. Un programa de LOGO! No se diferencia de un circuito eléctrico, dado que se representa de esta manera. Se debe de ajustar el contenido de ese circuito en la pantalla del LOGO!. En este capítulo, le introducimos en como convertir sus aplicaciones en programas de LOGO!.

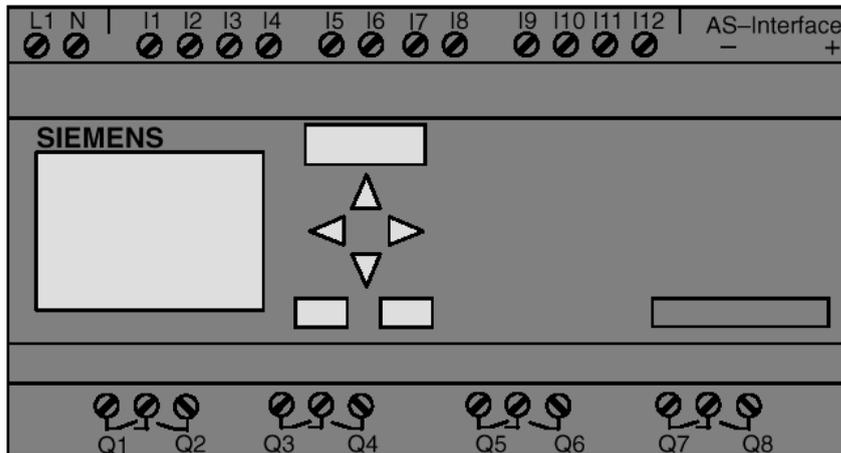
A continuación, le introduciremos en dos términos fundamentales, que son: **terminal** y **bloque**, mostrándole lo que se encierra detrás de ellos.

En segundo lugar, desarrollaremos un programa de un circuito simple y convencional que será introducido directamente en el LOGO! en tercer lugar. Tras leer algunas páginas del manual de servicio, su primer programa de LOGO!, se encontrará almacenado en este módulo programable a espera de su ejecución. Con el hardware apropiado (circuito) se podrá empezar a realizar los primeros ensayos.

3.1 Terminales



LOGO! contiene entradas y salidas:



Las entradas son identificadas con la letra I y un número. Cuando se mira al LOGO! En su frontal, podemos ver las entradas en la parte superior. Las salidas son identificadas con la letra Q y un número. Los terminales de salida se pueden ver en la parte inferior del dibujo.

3.2 Terminal de un LOGO!



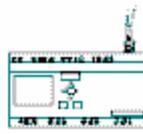
Identificamos todas las conexiones y estados que se encuentran en las aplicaciones con LOGO!. Las entradas / salidas pueden tener los estados '0' o '1'. El estado '0' significa que no se aplica tensión al terminal. El estado '1' significa que se está aplicando tensión al terminal. Esta información no debería de ser nueva para Vd. Hemos importado los terminales hi, lo y x para iluminar su entrada. 'hi' (high) es el estado '1' and 'lo' (low) es el estado '0'. Cuando no necesite conmutar una entrada de un bloque, se utilizará el terminal 'x'. Aprenderá qué es un bloque en la página siguiente.



Nota: Las entradas / salidas disponibles en LOGO!...B11 a través del interfase AS-I no son entradas físicas directas al LOGO!. Por favor, tenga en cuenta que el bus maestro apunta a estos dispositivos de entrada / salida a través del interfase AS-I.

3.3 LOGO! Conozca los Terminales Siguintes



TERMINALES			
	I1... I6, I7 (AI1), I8 (AI2)	I1...I12	I1...I12 Ia1...Ia4 (AS-Interface)
	Q1...Q4	Q1...Q8	Q1...Q8 Qa1...Qa4 (AS-Interface)
lo	Señal con nivel '0' (OFF)		
hi	Señal con nivel '1' (ON)		
x	Conector no utilizado		

3.4 Bloques y Números de Bloque



En este capítulo le introduciremos en la generación extensiva de circuitos con los elementos del LOGO! Y como los bloques operan los unos con los otros y con las entradas / salidas.

3.5 Bloques

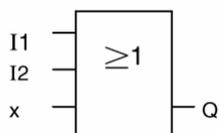


Un bloque en LOGO! Es una función que convierte la información de entrada en información de salida. Previamente deberá de cablear los elementos individuales a un armario eléctrico o a un panel de bornas. A través de la programación, se conectan los terminales a los bloques, por tanto, seleccionando simplemente, en el menú **Co**, la conexión deseada. El menú **Co** viene del término inglés "connector" (terminal).

3.6 Operaciones lógicas



Los bloques más simples son las operaciones lógicas (p.e. AND, OR).



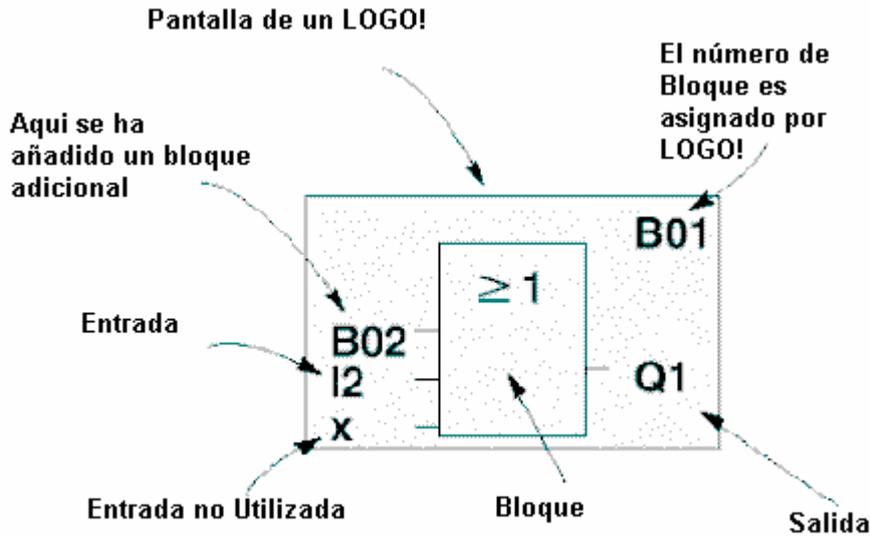
Aquí las entradas I1 e I2 se conectan a través de un bloque OR. La última entrada del bloque no se utiliza, por lo que se ha configurado como x.

Las otras funciones especiales son igualmente efectivas (p.e. relés, contadores, temporizadores,...)

3.7 Representación de Bloques en la pantalla del LOGO!



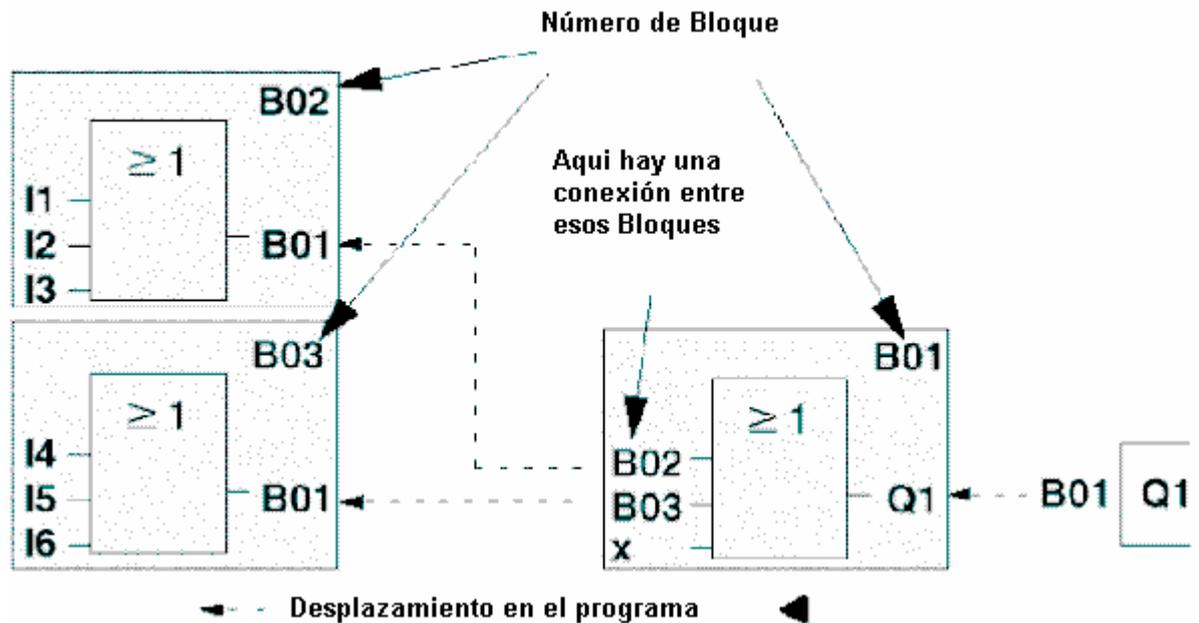
En la figura, le mostramos una pantalla típica de un LOGO!. Siempre se muestra en representación de bloques. Por lo tanto, hemos importado los números de bloque que deberían de serle de utilidad para manejar el circuito.



3.8 Asignación de un Número de Bloque



Siempre que se inserte un bloque en un programa, LOGO! Asigna a este bloque un número de bloque. LOGO! Le muestra las conexiones entre los bloques a través de los números de bloque. Los números de bloque sirven como ayuda orientativa en el programa.



En la figura, se muestran 3 vistas de un mismo LOGO!, que juntas dan lugar a un programa. Los bloques del LOGO! Se conectan unos a otros a través de los números de bloque.

3.9 Las 4 Reglas de Oro para la Puesta en Marcha de un LOGO!



Regla 1 – el control de los 3 dedos



Introducimos el circuito en el modo de operación programable. Se llega a este modo de operación al pulsar, de manera simultánea los 3 botones **Cursor izquierda**, **Cursor derecha** y **OK**.

En el modo de operación programable se puede modificar valores de tiempos y parámetros en el modo de operación parametrizable. Se llega al modo de operación parametrizable pulsando, de manera simultánea los 2 botones **ESC** y **OK**.

Regla 2 – Salidas y Entradas

Los circuitos se introducen siempre de salida a entrada.

Se puede conectar una salida con más entradas, pero nunca varias salidas a una entrada. No se puede conectar una salida a una entrada precedente dentro de la dirección del programa. Se pueden conmutar marcas de memoria con salidas para realimentar otras entradas.

Regla 3 - Cursor y movimiento del cursor

Cuando se introduzca un circuito, se tendrá en cuenta lo siguiente:

Si se representa el cursor con una ' _ ' se podrá **mover el cursor**:

- Con los botones \ddot{U} , \mathcal{P} , \mathcal{Y} o \mathcal{B} se mueve el cursor por el circuito
- Con **OK** se cambia a la "Opción Terminal/Bloque"
- Con **ESC** se abandona el modo entrada del circuito.

Si el cursor se representa como un bloque, entonces se podrá **seleccionar un Terminal/Bloque**:

- Con los botones \mathcal{Y} o \mathcal{B} se selecciona un Terminal/a Bloque
- Con **OK** se acepta la elección
- Con **ESC** se da un paso hacia atrás

Regla 4 – Planificación

Antes de introducir un circuito, se debería de planear todos los bloques en un papel o en el software LOGO!Soft Comfort.

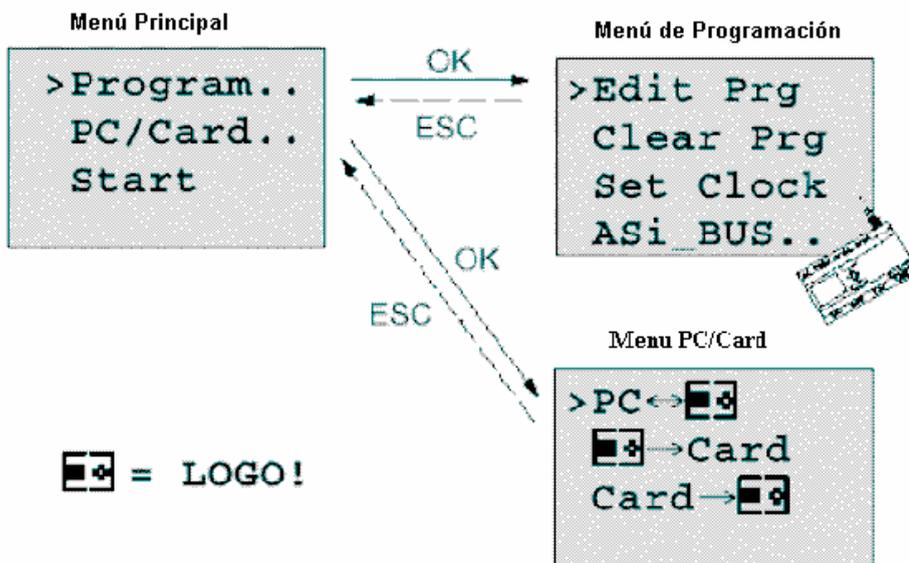
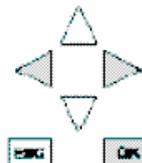
LOGO! Puede almacenar ahora todo el programa.

Cuando no se introduce un circuito por completo, **LOGO! no puede abandonar el modo de operación programable.**

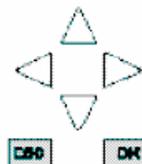
3.10 Visión Global de los Menús de LOGO!



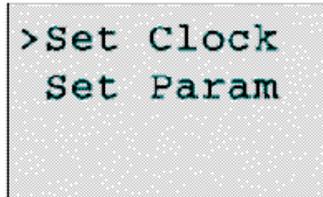
Modo de Operación Programable



Modo de Operación Parametrizable



Configuración de Parámetros



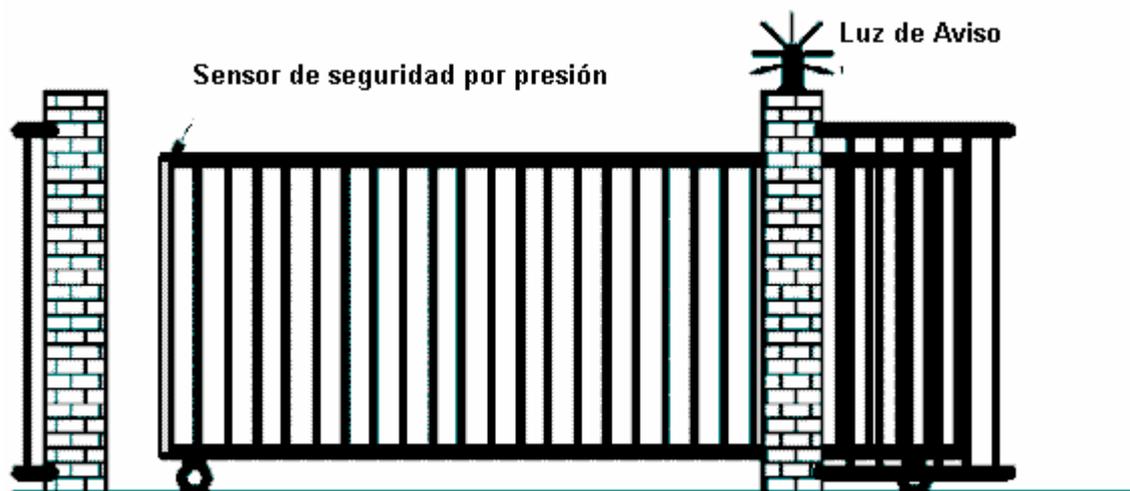
4. EJERCICIO EJEMPLO: CONTROL DE UNA PUERTA DESLIZANTE



El acceso a una fábrica, en muchos casos, puede realizarse desde diferentes lugares. El personal de seguridad no siempre puede vigilar todas las todas las puertas de acceso. Por tanto, deben de ser operadas y supervisadas por una unidad de control central. Adicionalmente se debe de garantizar que el personal pueda abrir y cerrar la puerta directamente.

Para cada puerta, utilizaremos un LOGO!12/24RC.

A través del AS-Interfase, los módulos son intercomunicados, unos con otros, y operados desde un maestro AS-I. En esta sección vamos a describir un controlador de puertas. Los otros controladores de puertas serán idénticos a éste.

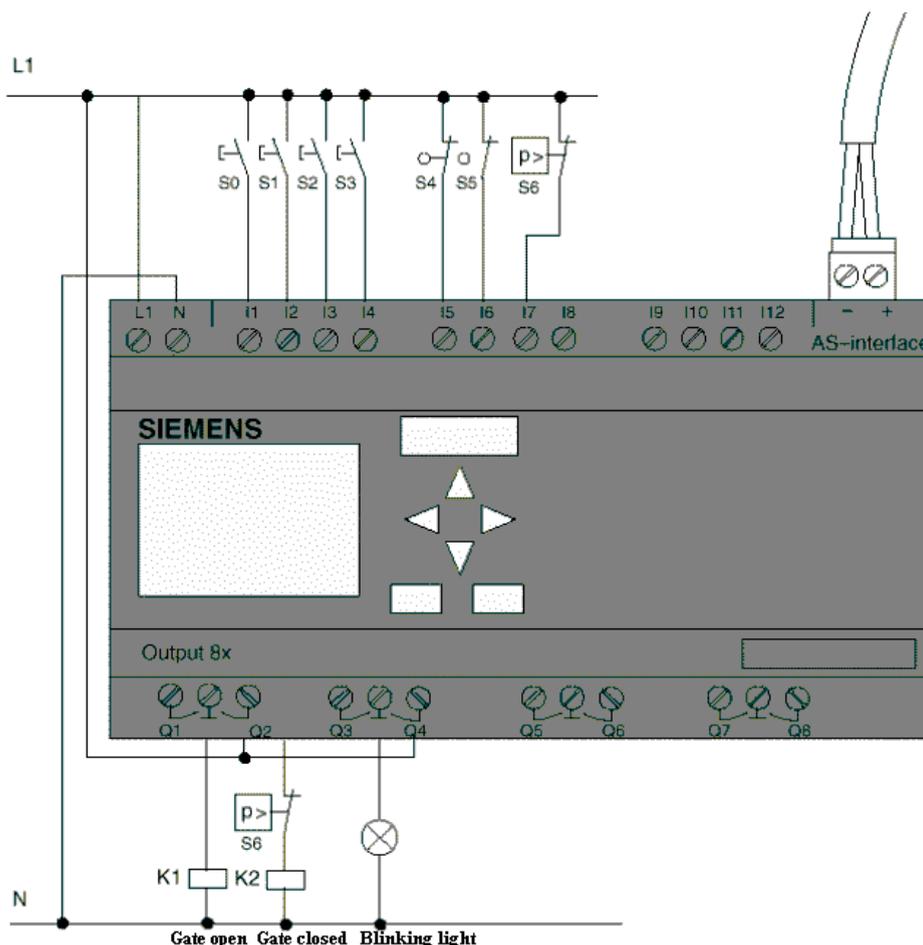


4.1 Requisitos para el Control de la Puerta



Cada puerta es abierta o cerrada a través de un conmutador accionado por un cable. La puerta se abre/cierra por completo. La acción de abrir y cerrar la realiza un portero a través de una conexión AS-I. Los estados PUERTA ABIERTA o PUERTA CERRADA son visualizados. Antes de que la puerta comience a moverse, una luz se iluminará intermitentemente durante 5 segundos. A través de un sensor de seguridad por presión, se garantiza que nadie resultará herido o que algún objeto pueda ser aplastado al cerrarse la puerta.

4.2 Cableado del control de puerta con el LOGO! 24RCLB11

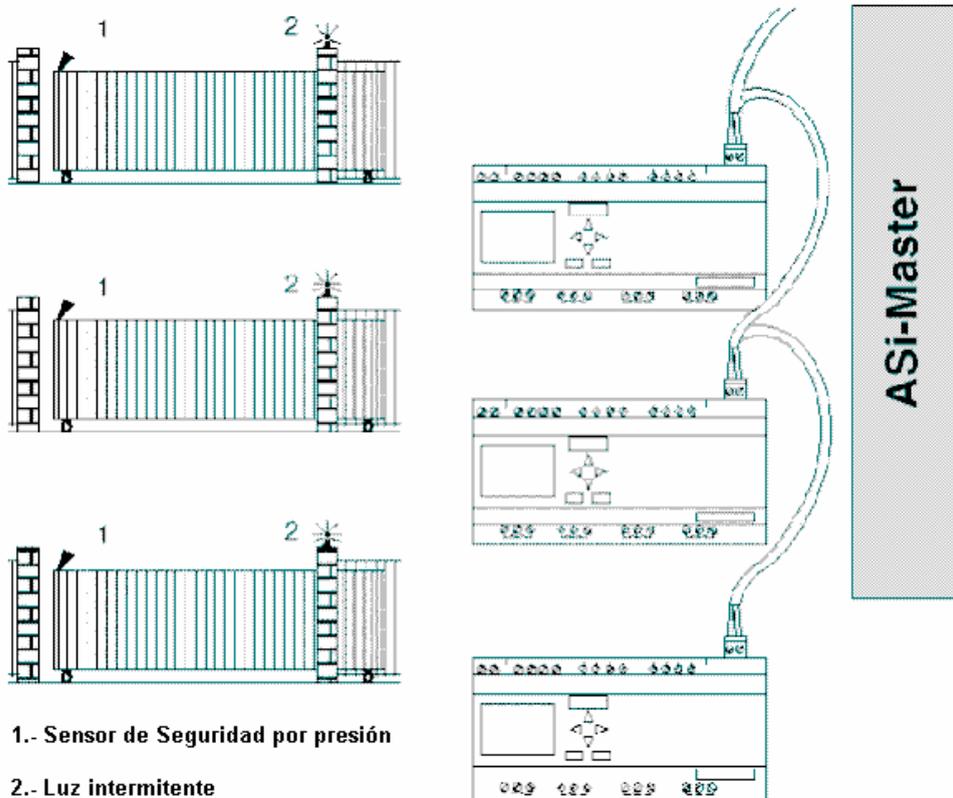


4.3 Componentes utilizados



- K1 Contacto principal abierto
- K2 Contacto principal cerrado
- S0 (Normalmente abierto) Conmutador ABRIR-PUERTA
- S1 (Normalmente abierto) Conmutador CERRAR-PUERTA
- S2 (Normalmente abierto) Pulsador ABRIR-PUERTA-MANUALMENTE
- S3 (Normalmente abierto) Pulsador CERRAR-PUERTA-MANUALMENTE
- S4 (Normalmente cerrado) Final de Carrera PUERTA-ABIERTA
- S5 (Normalmente cerrado) Final de Carrera PUERTA-CERRADA
- S6 (Normalmente cerrado) Sensor de seguridad por presión

4.4 Unidad Central y Monitorización de más Puertas de Acceso



4.5 Control por Capas a través del AS-Interfase

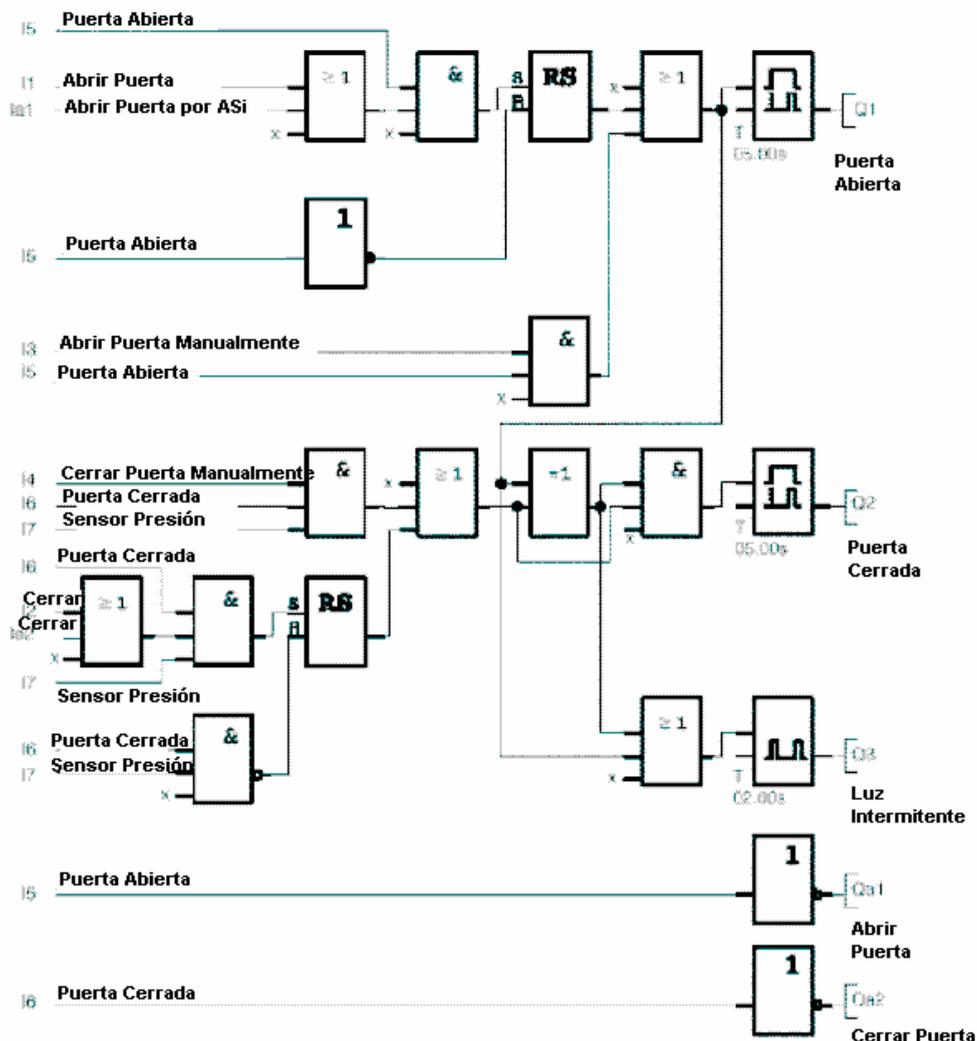


- Qa1 Final de Carrera PUERTA ABIERTA
- Qa2 Final de Carrera PUERTA CERRADA
- Ia1 Pulsador Externo PUERTA ABIERTA
- Ia2 Pulsador Externo PUERTA CERRADA



Nota: Tras utilizar la dirección del esclavo AS-I en el LOGO!RCLB11, se asigna un área de direcciones diferente en el Maestro AS-I.

4.6 Diagrama de Bloques de Función en LOGO!: Solución



A través del botón de arranque ABRIR PUERTA o CERRAR PUERTA, el movimiento de la puerta es ejecutado, siempre y cuando no se esté activando la orden de movimiento opuesto. El fin de movimiento se produce al pisar el final de carrera. El cierre de la puerta se puede detener a través del sensor de seguridad por presión.

5. INTRODUCIR EL PROGRAMA EN EL LOGO!

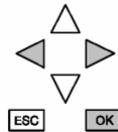
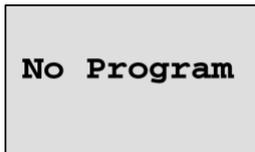


Una vez hemos diseñado el circuito, pasamos a introducirlo en el LOGO!. Este procedimiento se mostrará en los siguientes apartados.

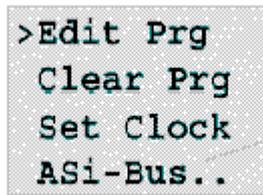
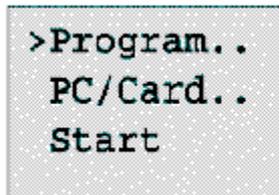
5.1 Cambiando al Modo de Operación "Programación"



Conectamos el LOGO! a la red y lo encendemos. En la pantalla se mostrará el siguiente mensaje:



Conmutamos el LOGO! al modo operativo de programación. Pulsamos los botones \uparrow , \downarrow y **OK** de manera simultánea.



El Menú de Programas del LOGO!

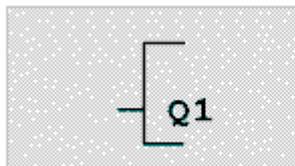
La posibilidad de conexión a Bus ASi sólo en los modelos LOGO!...LB11

En el primer campo o celda, se puede ver un ">". Con los cursores (\uparrow , \downarrow), movemos el ">" de arriba a abajo. Movemos el ">" a "Program.." y pulsamos el botón **OK**.

5.2 LOGO! Cambios en el menú de Programación



Aquí también se puede mover el ">" con los cursores (\uparrow , \downarrow). Activamos el ">" en "Edit Prg" (para editar programas) y pulsamos en **OK**.



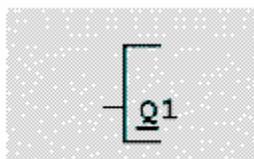
La Primera salida del LOGO!

LOGO! Le muestra ahora la primera salida: Con los cursores (\uparrow , \downarrow), se puede seleccionar el resto de salidas. Ahora puede comenzar a introducir su circuito.

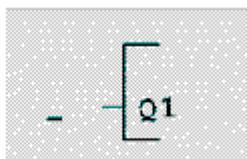
5.3 Introduciendo el Programa



Introducimos el programa de salida a entrada. Inicialmente, LOGO! Muestra la salida:



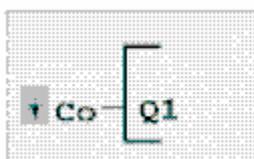
La primera salida de LOGO!



El cursor muestra dónde se encuentra en el programa

Bajo la salida Q1, se puede ver una “raya”. Llamaremos a esta “raya” un **cursor**. El cursor muestra la posición donde se encuentra en el programa. El cursor se puede desplazar con los botones de dirección. Pulsamos ahora el **cursor izquierda**.

A través de esta acción, entramos sólo en el primer bloque. Cambiamos al modo de entrada a pulsar el botón **OK**.



El cursor se representa resaltado. Se puede seleccionar entre un contacto o un bloque



En la lista SF se puede seleccionar bloques de funciones especiales

El cursor deja de representarse por una “raya” y parpadea a modo de bloque resaltado. Simultáneamente, LOGO! ofrece diferentes posibilidades de elección. Elija SF (pulsando **cursor abajo**, hasta que aparezca SF) y pulsamos el botón **OK**. LOGO! muestra ahora el primer bloque desde la lista de funciones especiales (SF):



Seleccionando el bloque de las funciones especiales de LOGO!, muestra el cursor dentro. Seleccionamos el bloque deseado con los cursores ▲ o ▼

El bloque de Retardo a la conexión contiene 3 entradas. La entrada superior es la entrada de trigger (Trg). Se arranca el temporizador a través de esta entrada. En nuestro ejemplo, el temporizador del bloque OR B02 es iniciado. El tiempo de temporizado se introduce a través del parámetro T.



Nota: Indicamos las funciones lógicas individuales con los números de bloque de su programa del LOGO! Desde el diagrama de bloques de función (Página 10). Se facilita la búsqueda o modificación de errores.

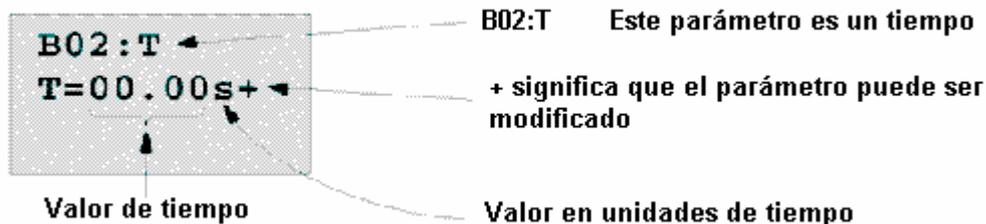
5.4 Parametrizando un Bloque



Introducimos ahora el tiempo T para el retardo a la conexión:

1. Si el cursor no se encuentra bajo la entrada T, lo moveremos con la ayuda de las teclas de cursor.
2. Para cambiar a modo de entrada: Pulsamos el botón **OK**

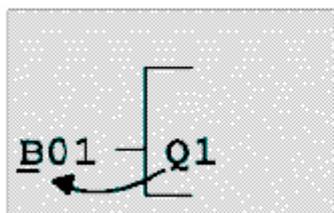
LOGO! muestra la ventana de parámetros:



Para modificar el valor de tiempo:

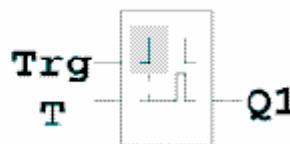
- Con los botones ◀ y ▶ se desplaza el cursor de izquierda a derecha
- Con los botones ▲ y ▼ se modifica el valor
- Una vez modificado el valor de tiempo, aceptamos con **OK**

Movemos el cursor bajo el B a B01 (B01 es el número de bloque del bloque temporizador). Pulsamos el botón de **cursor izquierda** dos veces para posicionarnos en la entrada **Trg**.



Movimiento de Cursor

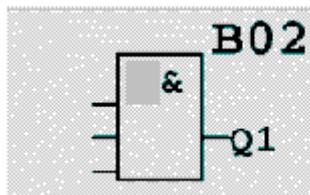
Pulsar en ◀



Pulsamos el botón **OK**.

Elegimos **BF** para Funciones Básicas con el **cursor abajo**.

Aceptamos con **OK** (Se muestra el bloque B02).



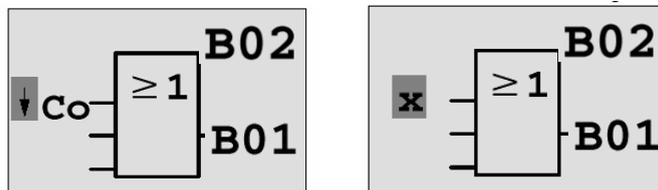
El primer bloque de la lista de funciones básicas es el **AND**

Elegimos un **OR** (función OR) con el **cursor abajo**.

Aceptamos con **OK**.



Pulsamos primero el botón **OK** (aparece CO) para la entrada del bloque OR, y pulsamos después el botón **OK** una vez más (aparece **x**) y por tercera vez, pulsamos **OK**. Se inserta una **x** para una entrada no utilizada. El cursor salta a la segunda entrada del bloque OR.



Pulsamos el botón **OK** en la segunda entrada y cambiamos a **SF** (Funciones Especiales) con el **cursor abajo**.

Aceptamos la opción con **OK**.

Seleccionamos un Flip-Flop **RS** con los cursores (**Y**, **B**) y aceptamos con **OK**.

Ahora nos encontramos en el bloque B03.

Añadimos ahora un bloque **AND** (función AND) en la entrada **Set** del Flip-Flop con **OK**, **cursor abajo**, **BF** para funciones básicas y una vez más en **OK**.

Aceptamos la elección con **OK**.

Estamos ahora en el bloque B04.

Insertamos la entrada **I5** a la primera entrada del Bloque AND con **OK** (aparece CO) y una vez más pulsamos en **OK** (se visualiza x). Pulsamos los cursores (**Y**, **B**).

Aceptamos con **OK**.

Insertamos un Bloque **OR** (función OR) en la segunda entrada con **OK**, **cursor izquierda**, **BF** (funciones básicas) y una vez más **OK** y **cursor abajo** .

Aceptamos con **OK**.

Estamos ahora en el bloque B05.

Insertamos la entrada **I1** en la primera entrada del Bloque **OR** con **OK** (aparece CO) y una vez más **OK** (aparece x) y luego los cursores (**Y**, **B**).

Aceptamos con **OK**.

Insertamos la entrada **Ia1** en la segunda entrada del Bloque **OR** con **OK** (aparece CO) y una vez más **OK** (aparece x) y luego los cursores (**Y**, **B**).

Aceptamos con **OK**.

Insertamos una **x** (lugar sin utilizar) en la tercera entrada del Bloque **OR** con **OK** (aparece CO) y una vez más **OK** (aparece x) y luego los cursores (**Y**, **B**).

Aceptamos con **OK**.

El Bloque B05 es cerrado y pasaremos al bloque B04.

Insertamos una **x** (lugar sin utilizar) en la tercera entrada del Bloque AND.

El Bloque B04 es cerrado y nos encontraremos en la entrada **Reset** del Bloque B03.

Insertamos un Bloque **NOT** (Inversor) en la entrada reset con **OK**, **cursor abajo**, **BF** (funciones básicas) y, una vez más, **OK** y dos veces en **cursor abajo**.

Aceptamos con **OK**.

Estamos ahora en el bloque B06.

Insertamos la entrada **I5** en la entrada del Bloque NOT con **OK** (aparece CO) y una vez más **OK** (aparece x) y los cursores (**Y**, **B**).

Aceptamos con **OK**.



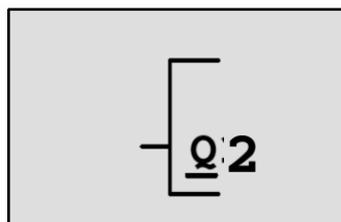
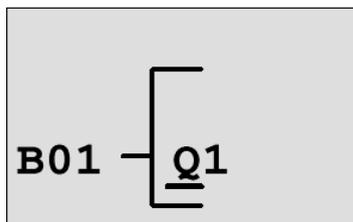
Se cierra el bloque B06 y nos encontraremos en la entrada del bloque B03.
Se puede configurar una retentividad del flip-flop con **OK** y los cursores (**Y**, **B**).
Aceptamos con **OK**.

Se cierra el Bloque B03 y nos encontraremos en la tercera entrada del bloque B02.
Insertamos un Bloque **AND** en la tercera entrada del Bloque OR con **OK**, **cursor abajo**, **BF** (funciones básicas) y **OK** una vez más.
Aceptar con **OK**.

Estamos ahora en el Bloque B05.
Insertamos la entrada **I3** en la primera entrada del Bloque AND con **OK** (aparece CO) y, una vez más **OK** (aparece x) y después los cursores (**Y**, **B**).
Aceptar con **OK**.

Insertamos la entrada **I5** en la segunda entrada del bloque **AND** con **OK** (aparece CO) y una vez más **OK** (aparece x) y los cursores (**Y**, **B**).
Aceptamos con **OK**.

Insertamos una **x** (lugar sin utilizar) en la tercera entrada del bloque **AND** con **OK** (aparece CO) y una vez más con **OK** (aparece x) y, por tercera vez, **OK**.
Las entradas de la salida Q1 ya están todas introducidas.



Seleccionamos ahora la salida Q2 con los cursores (**Y**, **B**).
Pulsamos **cursor izquierda** e introducimos el programa de la salida Q2 (Página 10).
Téngase en cuenta que los accesos son programados también aquí desde las salidas de los bloques disponibles.
Se encontrarán los bloques ya programados en **BN** (Número de Bloque), hacemos un clic en **CO** con **cursor arriba**).
Introducimos los programas para las salidas Q3, Qa1 y Qa2.

Cerramos ahora el programa introducido para el control de Puerta Deslizante en el LOGO!.

Abortamos ahora la edición de programa. Esto se lleva a cabo con:

1. Volver al menú de programa: Pulsando botón **ESC**.

En caso de no volver al menú de programa es porque nos hemos olvidado de programar todas las entradas de un bloque. LOGO! muestra la entrada en la que falta la asignación de una entrada / bloque. (LOGO! sólo acepta programas completos).

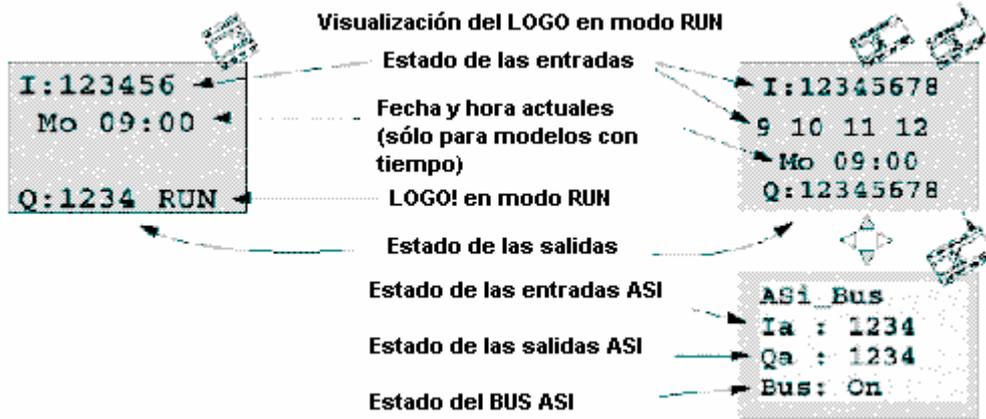
2. Volver al menú de programa: Pulsando botón **ESC**.

5.5 Pasando el LOGO! a modo RUN



1. Movemos '>' a 'Start' : con los cursores (Y, B).
2. Aceptamos Start: Pulsamos botón **OK** .

LOGO! pasa a modo RUN. LOGO! muestra la siguiente pantalla en RUN:



LOGO! ejecuta el programa en modo RUN. Después, LOGO! lee los estados de las entradas, detecta los estados de las salidas que están asociadas con el programa introducido, conmutando los relés de las salidas a '1' / '0'.



Ahora estamos preparados para probar el control de la puerta deslizante en el LOGO!

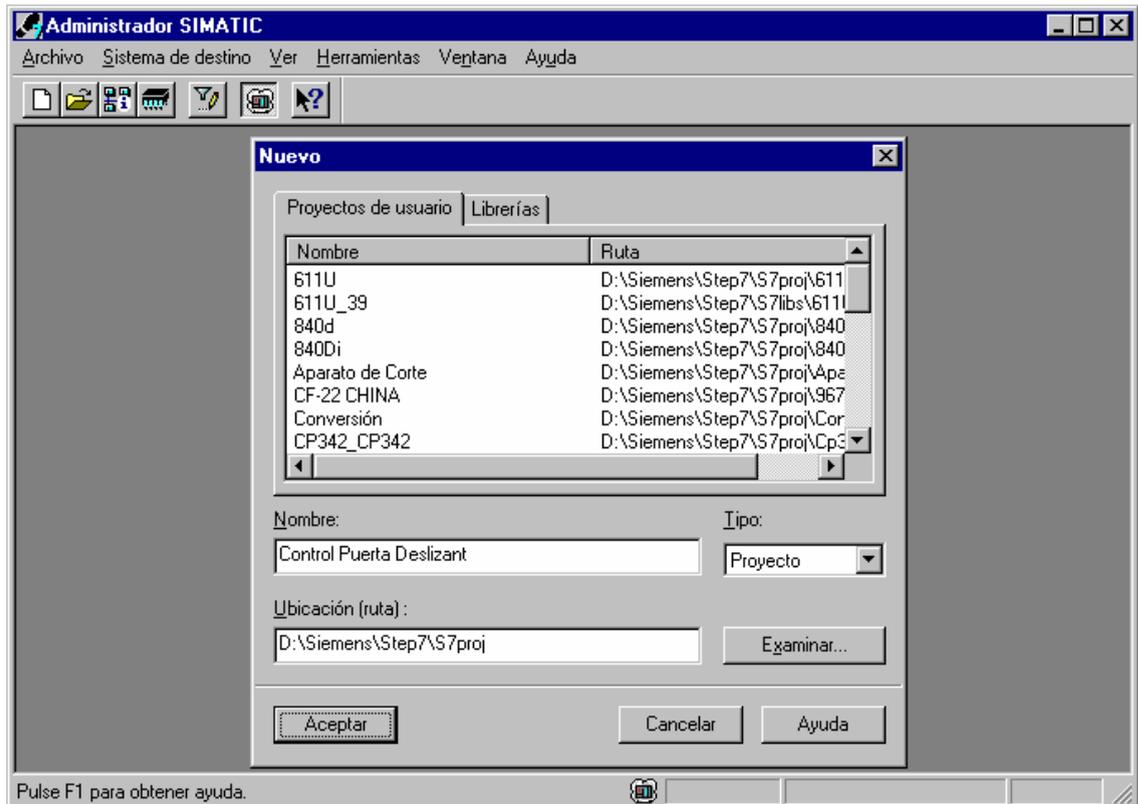
Nótese que, tanto los límites de carrera PUERTA ABIERTA, PUERTA CERRADA y el sensor de seguridad por presión, son normalmente cerrados. Con el botón manual ABRIR o CERRAR, el movimiento de la puerta es ejecutado, siempre y cuando el botón se active. Durante el procesamiento de la puerta, la lámpara de aviso parpadea intermitentemente 5 segundos antes de que comience el movimiento de la puerta (esto significa que el botón manual debe de mantenerse pulsado, al menos, durante este tiempo). Con esta configuración, la puerta se abre o se cierra en modo de operación automático.

6. GENERACIÓN DE UN PROYECTO STEP 7

6.1 Generación de un nuevo proyecto

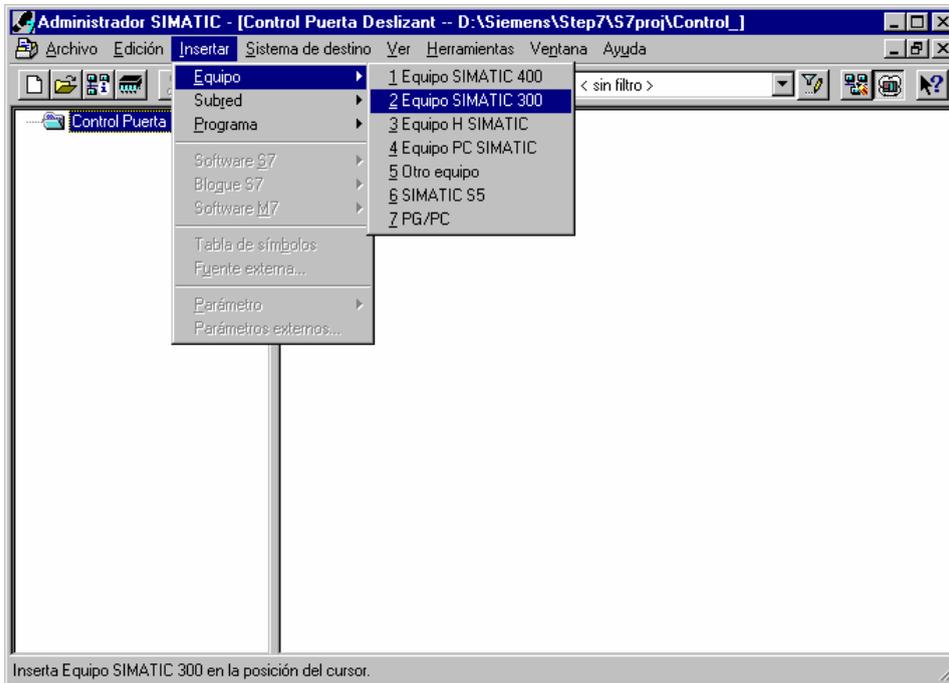


Arrancamos el **Administrador SIMATIC** y generamos un proyecto nuevo. Asignamos el nombre de **Control de Puerta Deslizante**. Confirmamos con **Aceptar**.

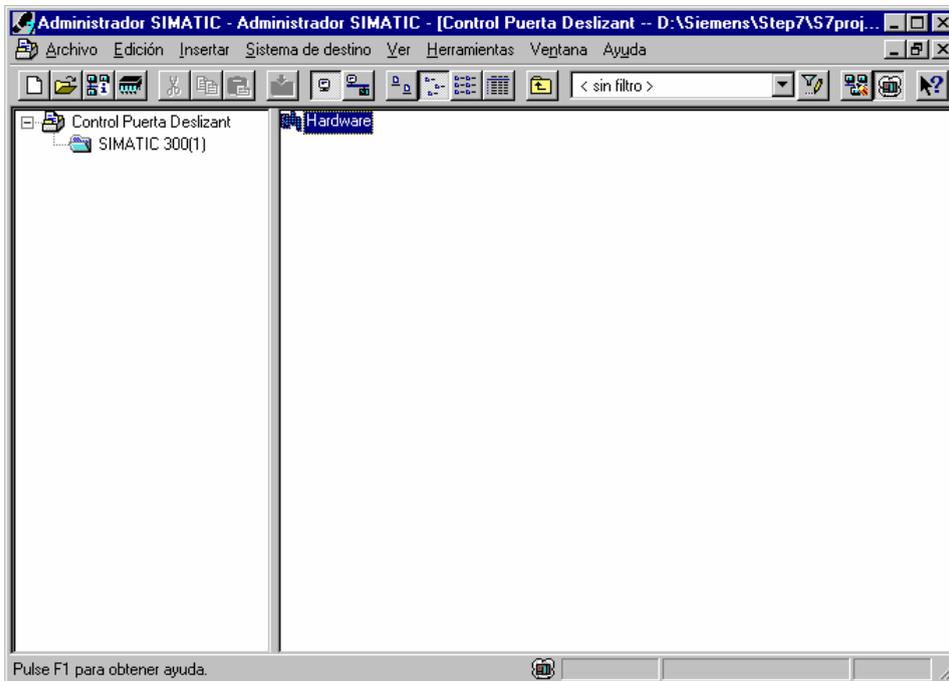




Insertamos ahora un **Equipo SIMATIC 300**.



Pulsamos en **SIMATIC 300(1)** en la parte izquierda de la pantalla y accedemos con doble click en **Hardware** a la pantalla de configuración de hardware.



6.2 Introduciendo la Configuración Hardware

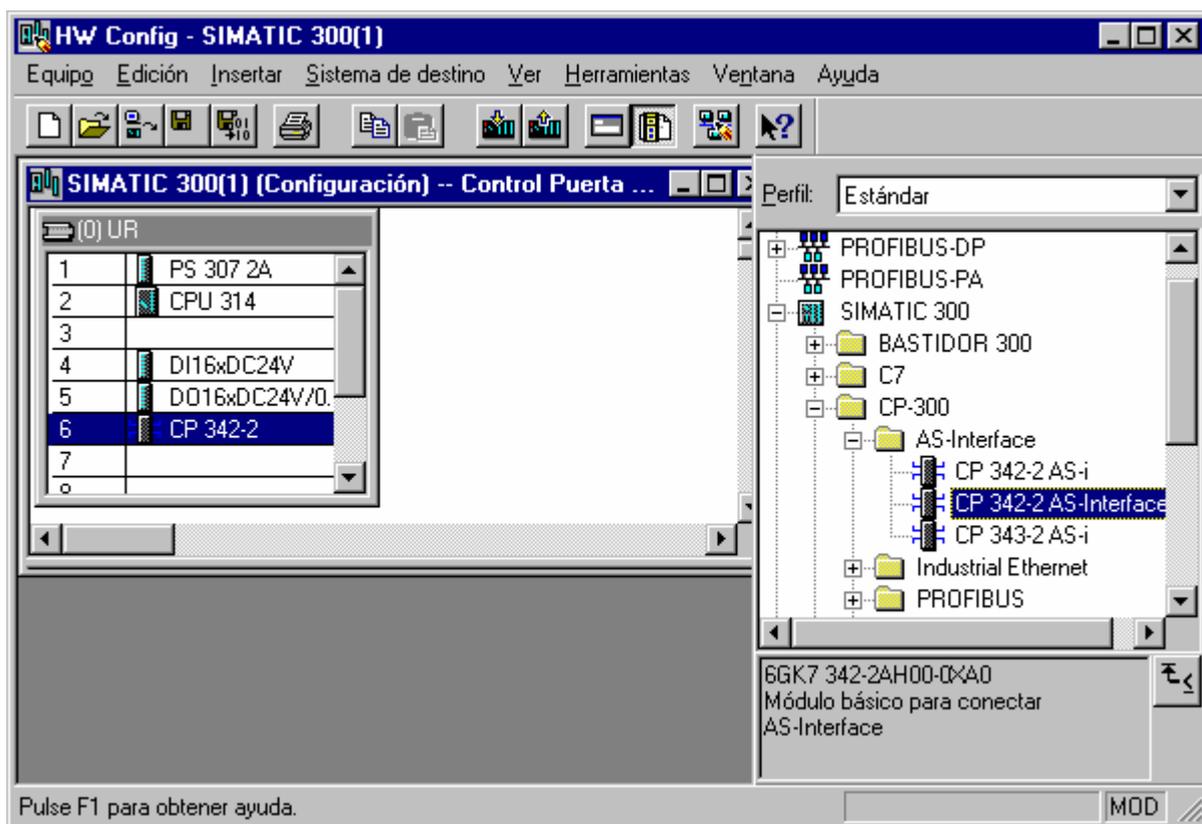


Introducimos la configuración hardware del PLC disponible. Téngase en cuenta las direcciones de las entradas / salidas y el módulo de comunicaciones del AS-Interfase.

Guardamos y Compilamos la configuración hardware.

Cargamos el hardware en la CPU.

Cerramos la ventana de la configuración hardware.



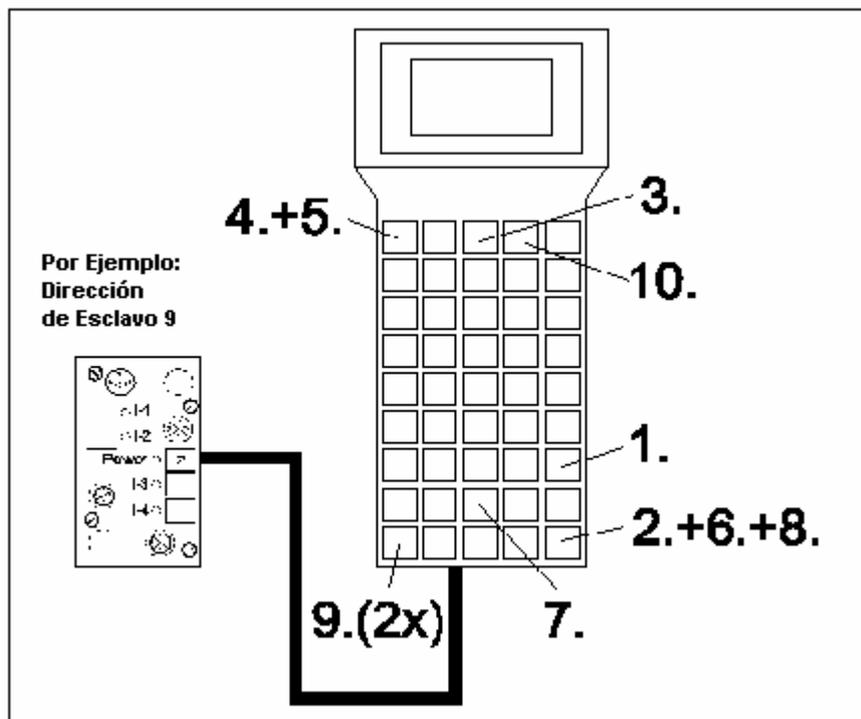
6.3 Accionando la Puerta Deslizante a través del AS-Interfase



El LOGO!12/24RC se conecta con el Maestro AS-I a través del AS-Interfase.

Primero se debe de asignar una dirección esclava AS-I al LOGO! a través de un elemento de programación externo. El Maestro AS-I reserva un área de memoria de 4 bits de entrada y 4 bits de salida para el LOGO!.

En nuestro ejemplo, asignamos al LOGO! la dirección 9.



LOGO!- conectado con la PSG.

1. Encendemos la PSG (START)
2. Activación (ENTER)
3. Elegimos 'Maestro' (F3)
4. Elegimos 'Operación Única' (F1)
5. Elegimos 'Nueva Dirección' (F1)
6. Activamos la dirección AS-I (ENTER)
7. Introducimos la nueva dirección (p.e.: 9)
8. Activamos las entradas (ENTER)
9. Volvemos al menú principal (2x ESC)
10. Apagamos la PSG (F4)



En nuestro programa ejemplo, la apertura y cierre de la puerta se ejecuta a través de los botones Puerta1-ABIERTA (Entrada I0.0) y Puerta1-CERRADA (Entrada I0.1).

La luz de aviso de puerta abierta o puerta cerrada, debería de llevarse a cabo con las salidas Q4.0 y Q4.1.

Las señales de entrada del AS-Interfase se transfieren a través del Byte de Entradas 10 en la imagen de proceso. Esta imagen de proceso pertenece al área de direcciones de la CP342-2, la cual puede ser leída desde la configuración hardware (aquí desde la PEW 288).

Las señales de salida del AS-Interfase se transfieren a través del Byte de Salidas 20 en la imagen de proceso. Esta imagen de proceso pertenece al área de direcciones de la CP342-2, la cual puede ser escrita desde la configuración hardware (aquí desde la PAW288).

La siguiente tabla debería aclarar cualquier tipo de duda.

Entradas PAE	ENTRADA / SALIDA				ENTRADA / SALIDA				Dirección CP342-2 (PE/PA)	Salidas PAA
	7	6	5	4	3	2	1	0		
	In4 Out4	In3 Out3	In2 Out2	In1 Out1	In4 Out4	In3 Out3	In2 Out2	In1 Out1		
10	Ocupada				Esclavo01				288	20
11	Esclavo02				Esclavo03				289	21
12	Esclavo04				Esclavo05				290	22
13	Esclavo06				Esclavo07				291	23
14	Esclavo08				Esclavo09				292	24
15	Esclavo10				Esclavo11				293	25
16	Esclavo12				Esclavo13				294	26
17	Esclavo14				Esclavo15				295	27
18	Esclavo16				Esclavo17				296	28
19	Esclavo18				Esclavo19				297	29
20	Esclavo20				Esclavo21				298	30
21	Esclavo22				Esclavo23				299	31
22	Esclavo24				Esclavo25				300	32
23	Esclavo26				Esclavo27				301	33
24	Esclavo28				Esclavo29				302	34
25	Esclavo30				Esclavo31				303	35

La dirección del Esclavo 9 es el acceso a nuestro LOGO!12/24RC.

Las salidas AS-I del esclavo LOGO! Nr.9 son asignadas en la imagen de proceso de la tabla de entradas de la CPU SIMATIC S7-300 CPU, de la siguiente manera:

Qa1 = E14.0, Qa2 = E14.1, Qa3 = E14.2 Qa4 =E14.3

Las entradas AS-I del esclavo LOGO! Nr.9 son asignadas en la imagen de proceso de la tabla de salidas de la CPU SIMATIC S7-300 CPU, de la siguiente manera:

Ia1 = A24.0, Ia2 = A24.1, Ia3 = A24.2, Ia4 = A24.3

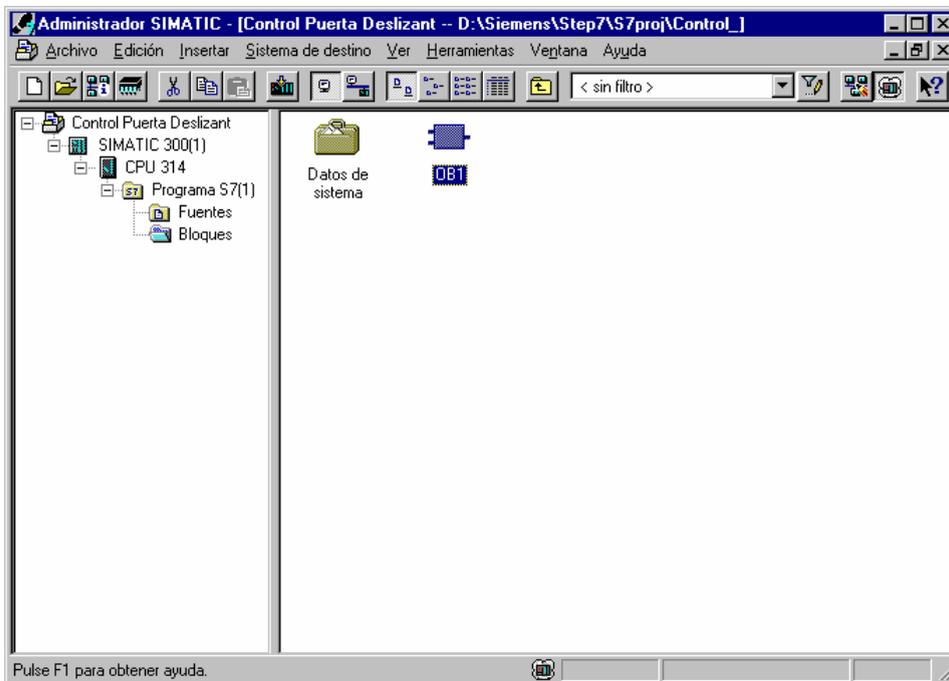


Nota: Una salida del Esclavo es una entrada para el Maestro.
Una entrada del Esclavo es una salida para el Maestro.

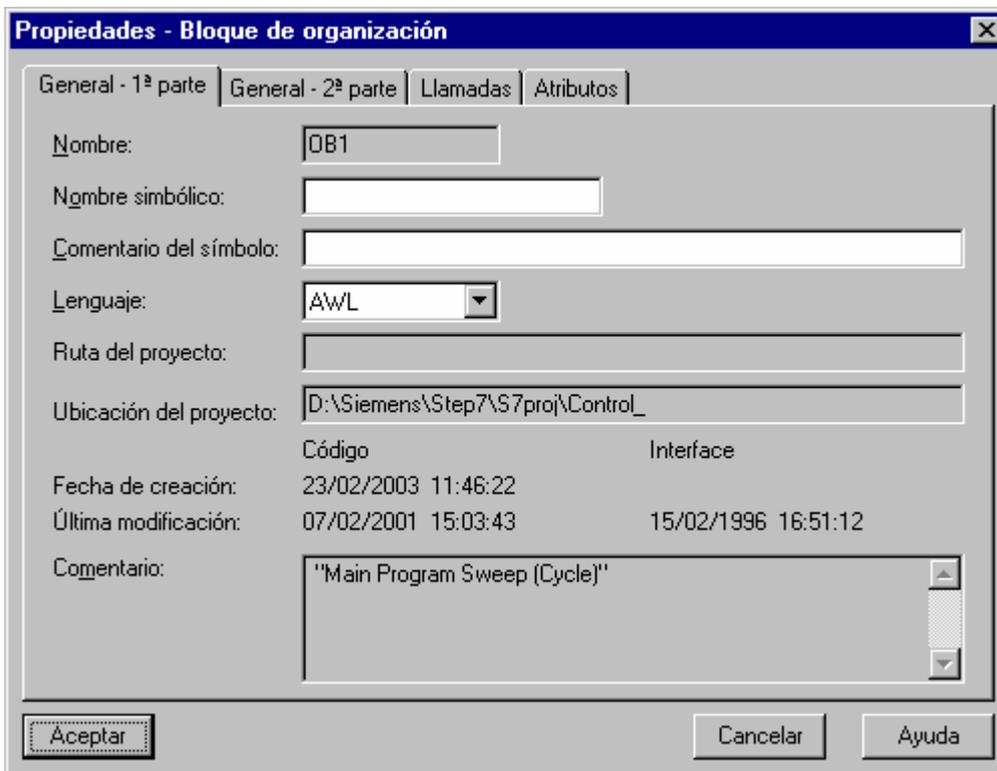
6.4 Control del Programa para el Accionamiento de una Puerta Deslizante



Desplegamos el árbol de directorios del **Equipo SIMATIC 300** hasta la carpeta **Bloques**. Hacemos doble click en **OB1**.



Aceptamos las propiedades con **Aceptar**.



6.5 Introduciendo el Bloque de Organización OB1



Segn. 1: Lectura de los Datos ASI en la PAE

Comentario:

```
L   PED  288
T   ED   10
L   PED  292
T   ED   14
```

Segn. 2: Esperamos a que se abra la puerta

Comentario:

```
U   E    0.0
=   A    24.0
```

Segn. 3: Esperamos a que se cierre la puerta

Comentario:

```
U   E    0.1
=   A    24.1
```

Segn. 4: Visualizamos la apertura y cierre de la puerta

Comentario:

```
U   E    14.0
=   A    4.0
U   E    14.1
=   A    4.1
```

Segn. 5: Trasferimos los datos de la PAA al Bus ASI

Comentario:

```
L   AD   20
T   PAD  288
L   AD   24
T   PAD  292
```

Después de introducir el OB1 **guardamos y lo cargamos** en la CPU.

Pasamos el selector de la CPU a RUN y probamos el programa.