

Curso Práctico

Introducción a los Autómatas Programables FATEK

EJERCICIOS (Nivel Básico)



INDICE

- Ejercicio	1º	Activación directa de una salida por medio de una entrada (Acción directa). 01
- Ejercicio	2º	Activación directa de una salida por medio de una entrada (Acción inversa). 01
- Ejercicio	3º	Activación de una salida con señales en serie (Acciones directas). 03
- Ejercicio	4º	Activación de una salida con señales en serie (Acciones directas e inversas combinadas). 05
- Ejercicio	5º	Activación de una salida con señales en paralelo 07
- Ejercicio	6º	Activación de una salida con señales en serie y paralelo combinadas. 10
- Ejercicio	7º	Activación de más de una salida en paralelo. 13
- Ejercicio	8º	Ramas en paralelo a la entrada y a la salida de renglón. 16
- Ejercicio	9º	Control de Marcha / Paro con realimentación. 18
- Ejercicio	10º	Control de Marcha / Paro con funciones SET y RST. 20
- Ejercicio	11º	Encendido temporizado de una luz (Con temporizador sin reset) 22
- Ejercicio	12º	Encendido temporizado de una luz (Con temporizador con reset) 24
- Ejercicio	13º	Control de aforo en la sala de un museo (Solución con dos contadores). 26

INDICE

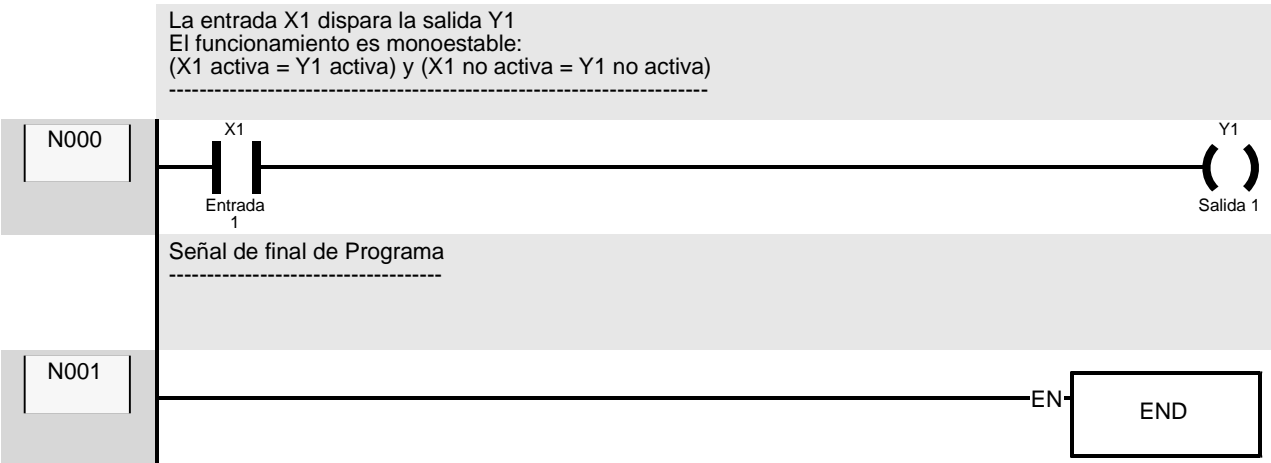
- Ejercicio 14º	Control del encendido automático de las luces de una biblioteca	30
- Ejercicio 15º	Control de llenado de una balsa de riego.	35
- Ejercicio 16º	Encendido temporizado variable de una luz (Con temporizador con preselección digital variable)	38

Ejercicio 1º

La práctica que proponemos para comenzar a trabajar con el Automata **FBS10MA**, es el programa más sencillo que existe:

Una sola entrada activa una salida determinada. Cuando excitamos la entrada programada, se acciona la salida que hemos elegido y al resetear la entrada, la salida se resetea igualmente.

En el ejemplo hemos seleccionado la entrada X1 y la salida Y1.

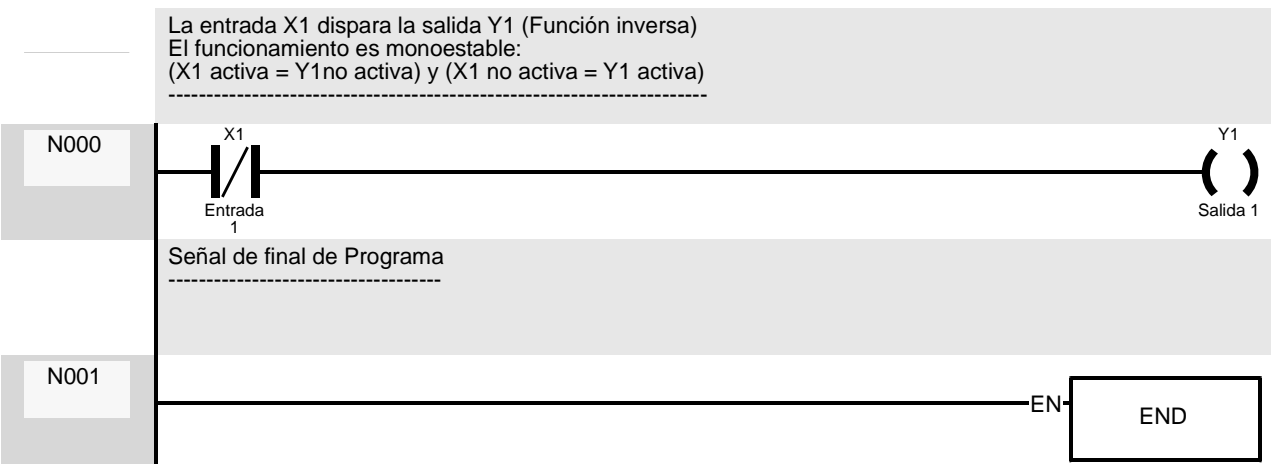


Ejercicio 2º

Este segundo ejercicio es igualmente de sencillo. El concepto es igual al anterior, pero en este caso se programa la función inversa. Es decir, la salida Y1 se activa cuando no está accionada la entrada X1. Como en el Ejercicio 1º, el funcionamiento es monoestable.

En esta ocasión, la salida Y1 se activa al no estar accionada la entrada X1.

Veamos el programa:

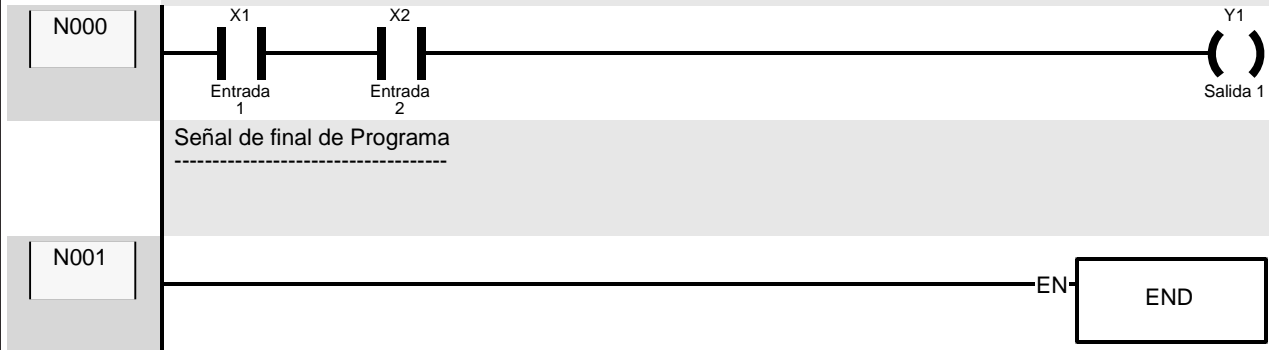


Ejercicio 3º

En este ejercicio vamos a complicar ligeramente el programa. Combinaremos la señal de dos entradas, usaremos X1 y X2 para activar la salida Y1.

Vamos a establecer un circuito serie de tal forma que será la simultaneidad de la activación de las dos entradas la que disparará la salida. Es decir, si las dos entradas están activas, se acciona la salida Y1. Este tipo de funcionamiento se denomina función lógica "Y". Veamos el programa correspondiente:

Mientras la entrada X1 y la entrada X2 estén accionadas al mismo tiempo se mantiene activa la salida Y1 ($X1 \cdot X2 = Y1$)
 Al desactivarse cualquiera de las entradas, o ambas, se resetea la salida

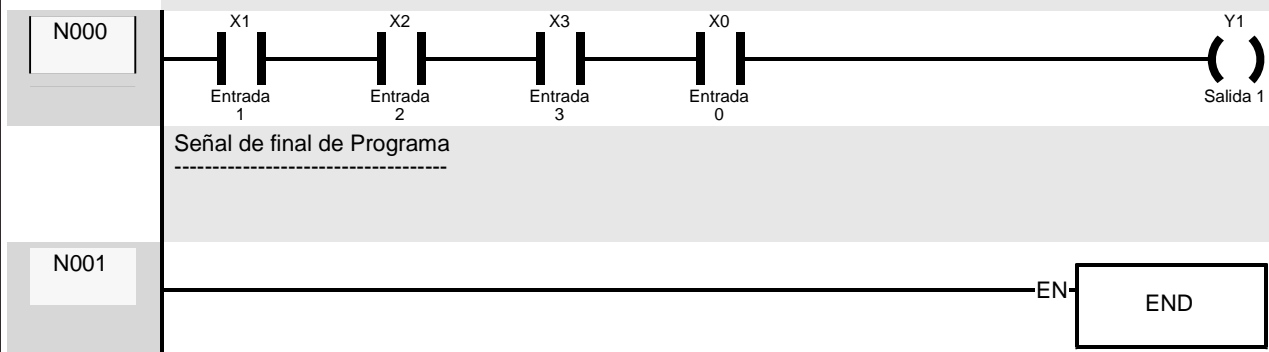


En este ejercicio se puede considerar distintas combinaciones entre contactos N.A. y N.C. de ambas salidas. Proponemos al cursillista que realice distintas pruebas en este sentido.

Comprobar que cuando falla, al menos, una de las condiciones de entrada del renglón, la salida se desactiva.

Naturalmente, se pueden colocar más instrucciones en serie. Veamos un ejemplo:

Las entradas X1, X2, X3 y X0 accionadas al mismo tiempo activan la salida Y1 ($X1 \cdot X2 \cdot X3 \cdot X0 = Y1$)
 Cualquiera que se desactive, ó 2 ó 3 ó todas, resetea la salida

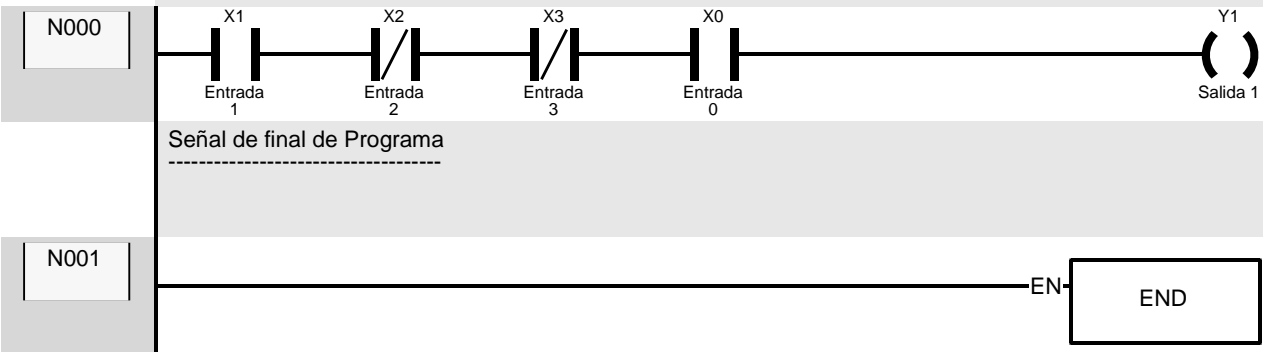


Ejercicio 4º

En el ejercicio anterior hemos usado solo instrucciones N.A. pero se puede emplear cualquier combinación de instrucciones N.A. y N.C., según las necesidades de nuestra maniobra.

Veamos el siguiente ejemplo:

Las entradas X1, X0 activadas y las X2, X3 no activadas activan la salida Y1
($X1 * \text{no}X2 * \text{no}X3 * X0 = Y1$)



Cuando activamos las entradas X1 y X0, estando sin activar las entradas X2 y X3, la salida se dispara.

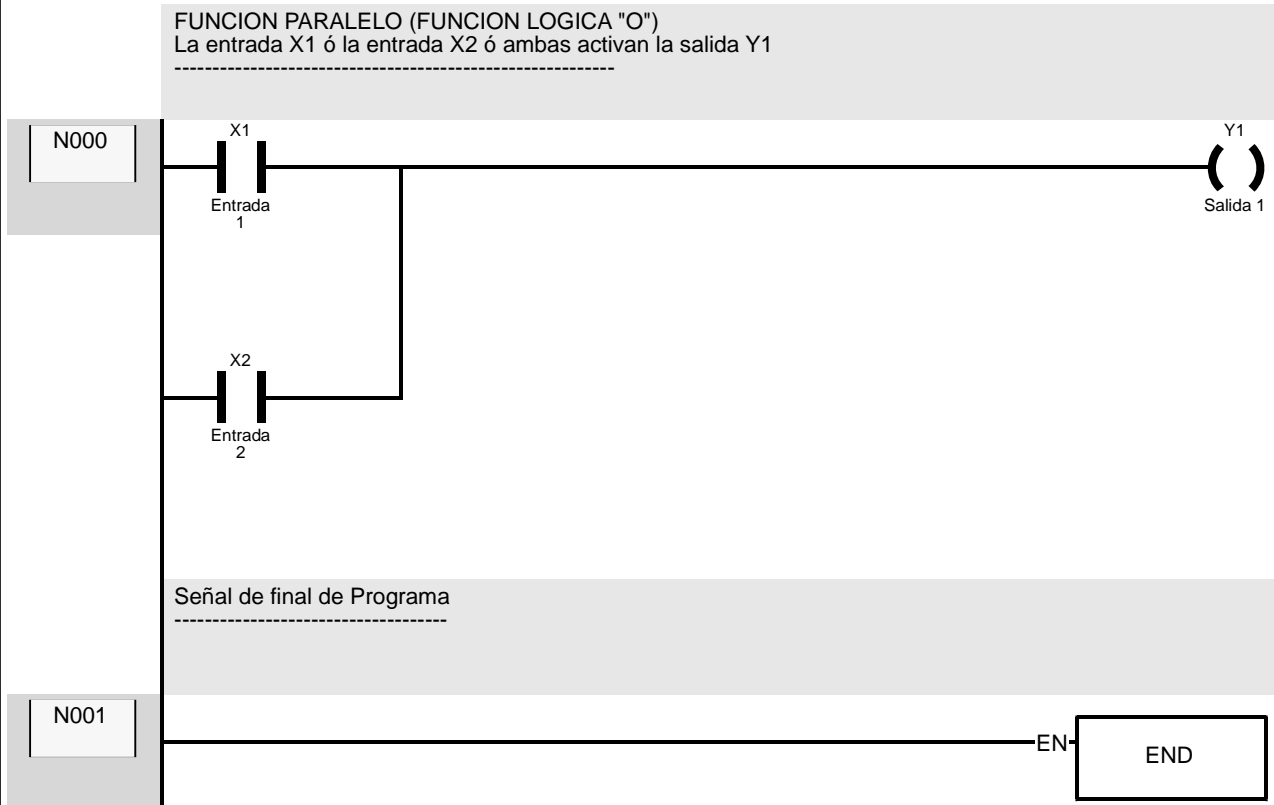
Proponemos al cursillista que realice cambios de estado en estas entradas y observe la respuesta del Automata fijandose en la conexión y desconexión de la salida Y1.

Ejercicio 5º

En este ejercicio vamos a introducir una variación. Consideraremos la activación de la salida Y1 por medio de dos entradas, X1 y X2, pero en esta ocasión estarán programadas en paralelo. Es decir, cualquiera de las dos dispara la salida aunque la otra entrada esté desactivada.

Este tipo de funcionamiento también se denomina función lógica "O".

Programemos el siguiente ejemplo:



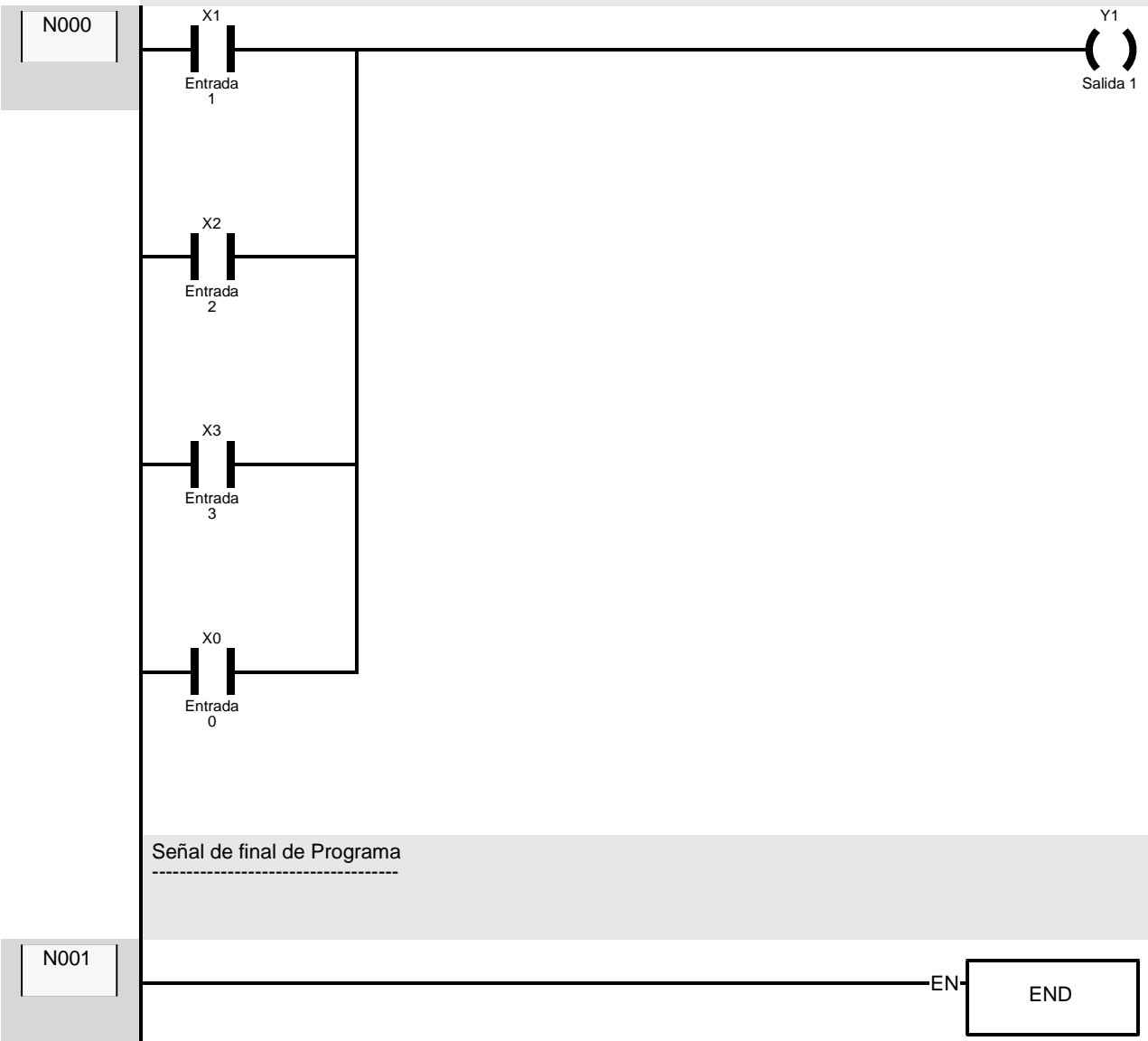
Comprobar el funcionamiento del programa activando y desactivando las entradas.

A continuación podemos programar un renglón con cuatro ramas en paralelo.

Además, proponemos al cursillista que realice cambios en el programa y combine instrucciones N.A. con N.C. para experimentar distintas secuencias.

Continuación Ejercicio 5º

FUNCION PARALELO (FUNCION LOGICA "O")
Las entradas X1, X2, X3 ó X0 activan la salida Y1

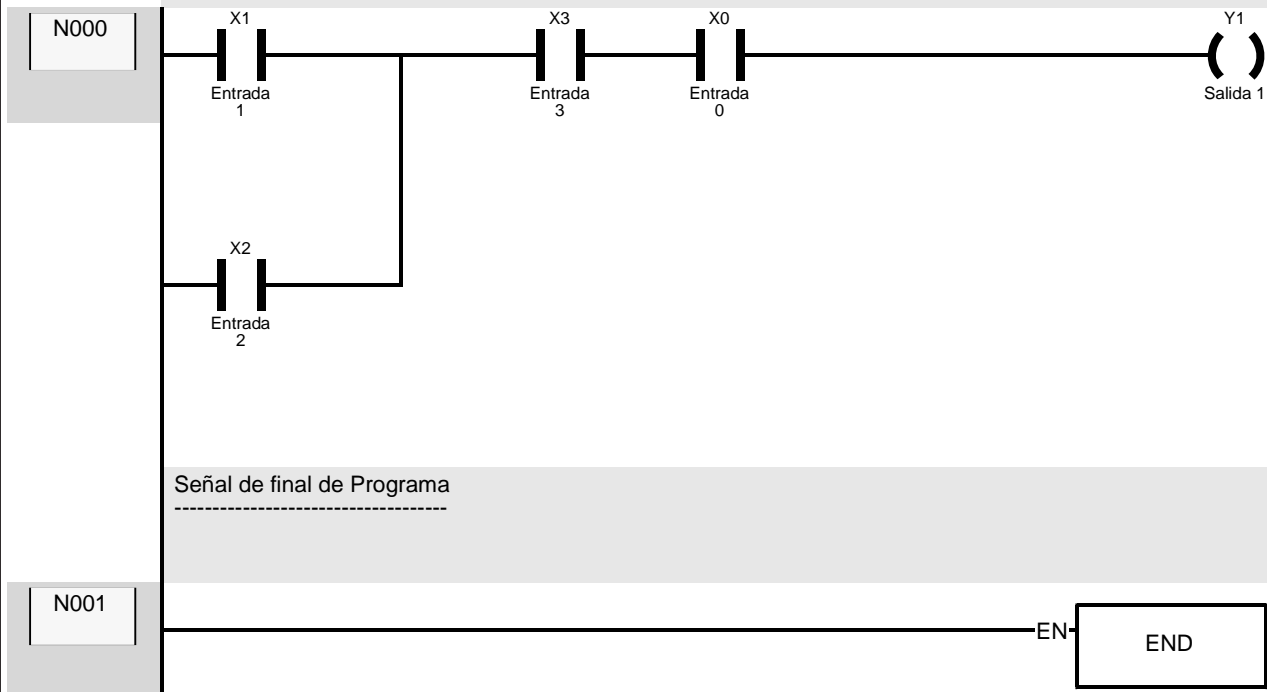


Ejercicio 6º

Vamos a combinar los circuitos serie con los paralelos. Realicemos el siguiente programa:

FUNCION PARALELO Y FUNCION SERIE

La entrada X1 ó la entrada X2 en serie con las entradas X3 y X0, activan la salida Y1



Realizar distintas activaciones y desactivaciones de las cuatro entradas y observar la respuesta del Automata.

Programamos otro ejemplo en la página siguiente.

Como se puede apreciar, hay dos ramas en paralelo. En la segunda rama hay una condición serie y despues hay una instruccion en serie.

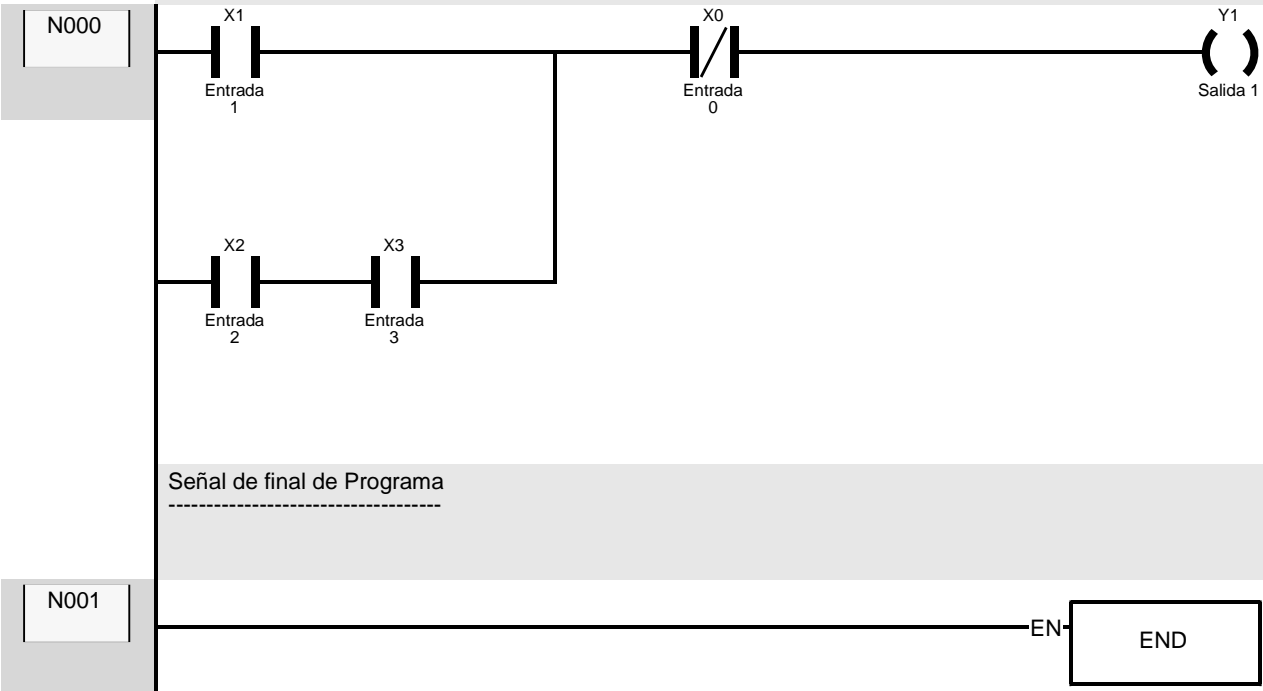
Para que se dispare la salida es imprescindible que la entrada X0 esté activa. Despues, dependiendo del estado de cada una de las dos ramas en paralelo, la salida se disparará o no

El cursillista debe activar/desactivar las entradas para ver la respuesta del programa.

Así mismo, proponemos que se realicen modificaciones sobre este programa, experimentando distintas combinaciones de ramas en paralelo y secuencias en serie.

Continuación Ejercicio 6º

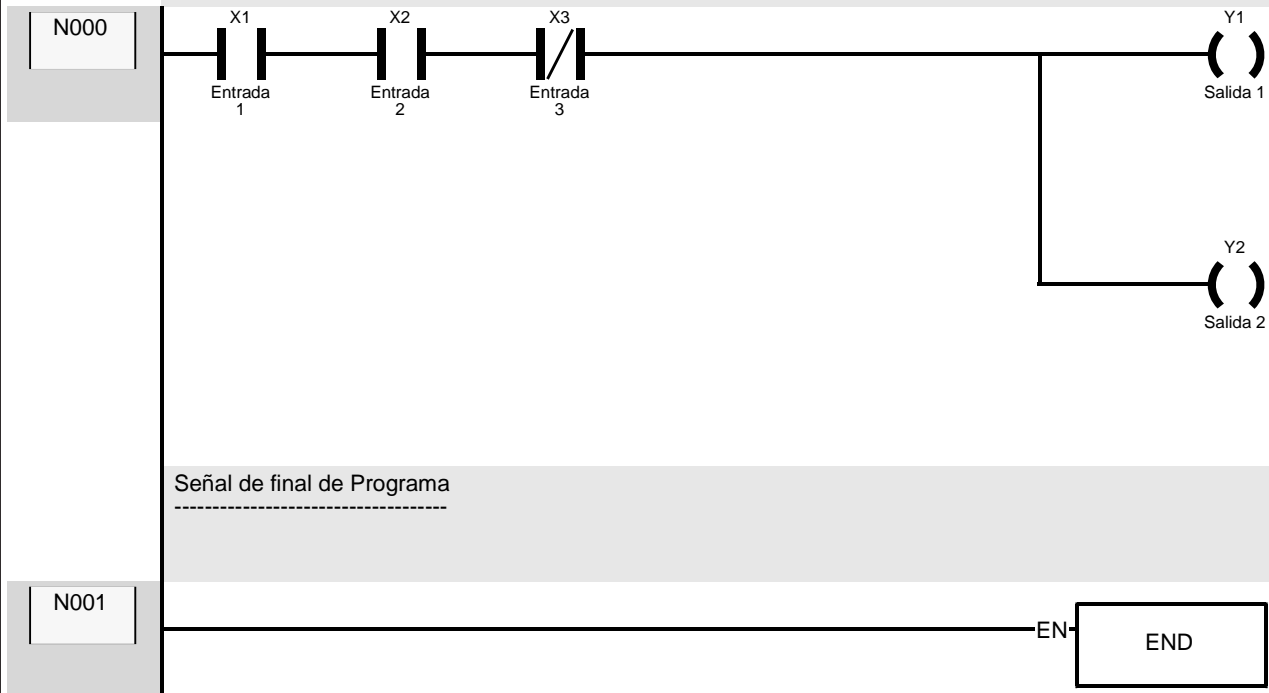
FUNCION PARALELO Y FUNCION SERIE
La entrada X1 ó la entrada X2 en serie con la entrada X3 y no estando X0, activan la salida Y1



Ejercicio 7º

Hasta ahora hemos trabajado solo con una salida de renglón, pero vamos a comprobar que podemos programar, a la salida del renglón, ramas en paralelo:

RAMAS PARALELAS EN LA SALIDA DE RENGLON
La serie de las tres entradas X1, X2 y X3 activa las salidas Y1 e Y2



Señal de final de Programa

La activación de las entradas X1, X2 y no X3 del Automata dispara las dos salidas Y1 e Y2.

En ejercicios posteriores podremos ver como se pueden poner, en paralelo, a la salida de un renglón, no solo señales de bit (como salidas o relés internos), sino además, funciones como transferencia de palabras, operaciones aritméticas, operaciones lógicas, comparaciones, etc.

En el siguiente ejemplo añadimos más ramas de salidas en paralelo

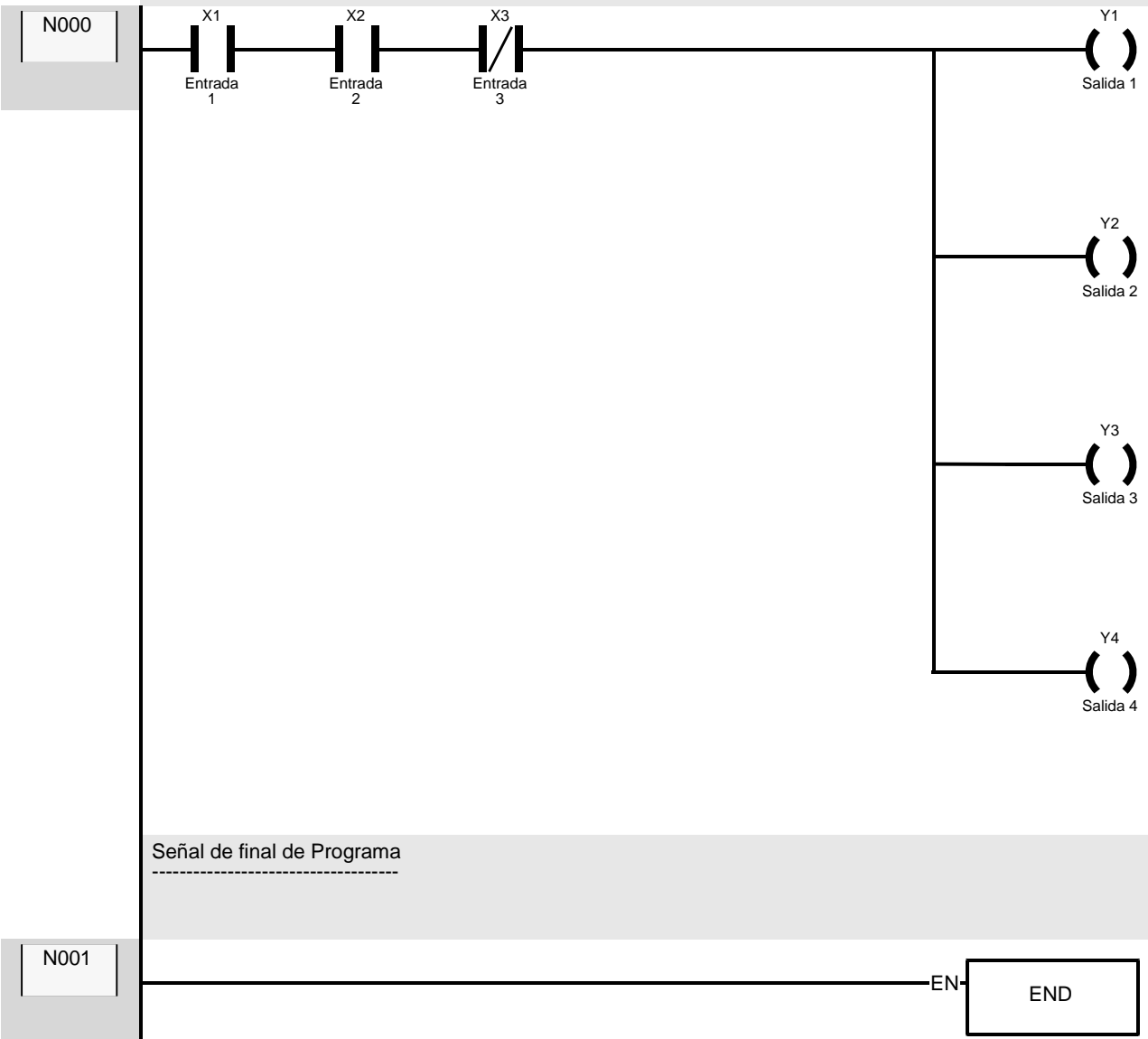
Las salidas Y0, Y1, Y2 e Y3 se activan simultáneamente cuando lo están las entradas X1, X2 y no X3.

Le proponemos al cursillista que escriba el programa y experimente la reacción del Automata ante distintas modificaciones que realice en las instrucciones.

Naturalmente, también se puede programar renglones con ramas en paralelo, tanto en la entrada como en la salida de renglón. Lo veremos en el próximo ejercicio.

Continuación Ejercicio 7º

RAMAS PARALELAS EN LA SALIDA DE RENGLON
La serie de las tres entradas X1, X2 y X3 activa las salidas Y1, Y2, Y3 e Y4

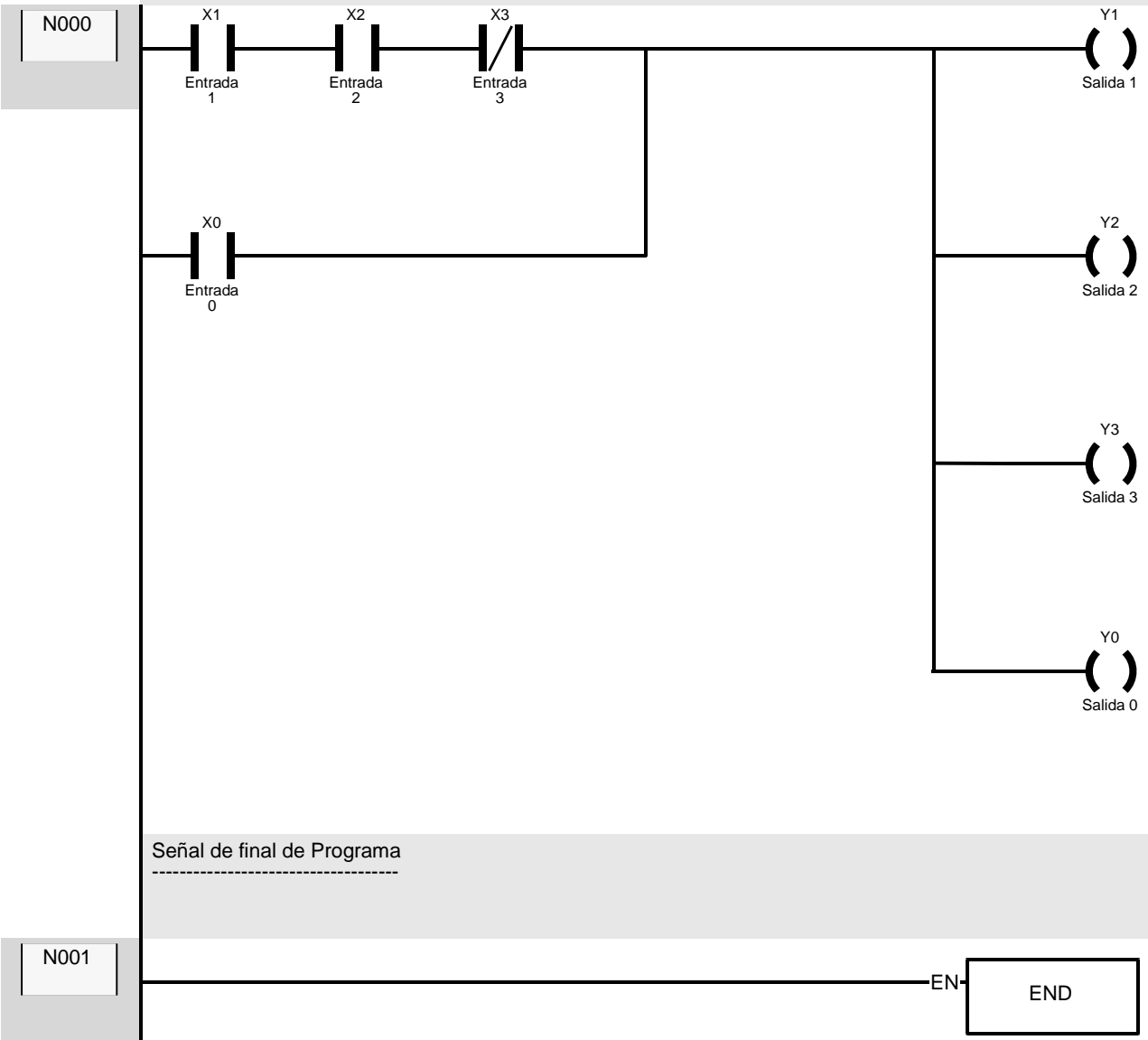


Naturalmente, también se puede programar renglones con ramas en paralelo, tanto en la entrada como en la salida de renglón. Lo veremos en el próximo ejercicio.

Ejercicio 8º

Veamos un ejemplo de renglón con ramas en paralelo, tanto en la entrada como en la salida:

RAMAS PARALELAS, TANTO EN LA ENTRADA DE RENGLON COMO EN LA SALIDA DE RENGLON
 La serie de las tres entradas X1, X2 y no X3 en paralelo con X0 activa las salidas Y1, Y2, Y3 e Y4



Señal de final de Programa

Aconsejamos al cursillista que realice este programa y compruebe su funcionamiento en el Automata programable.

Ejercicio 9º

En los ocho primeros ejercicios hemos adquirido los conocimientos suficientes para poder sesenvolvemos básicamente en la ejecución de programas sencillos de Marcha/Paro de salidas de Automatas.

Ha llegado el momento de realizar los ejercicios desde un punto de vista más práctico. A partir de ahora, todas las prácticas las realizaremos para resolver aplicaciones reales. Esperamos que de esta forma resulte más gratificante y divertido.

Para empezar vamos a programar el Marcha/Paro de una banda transportadora.

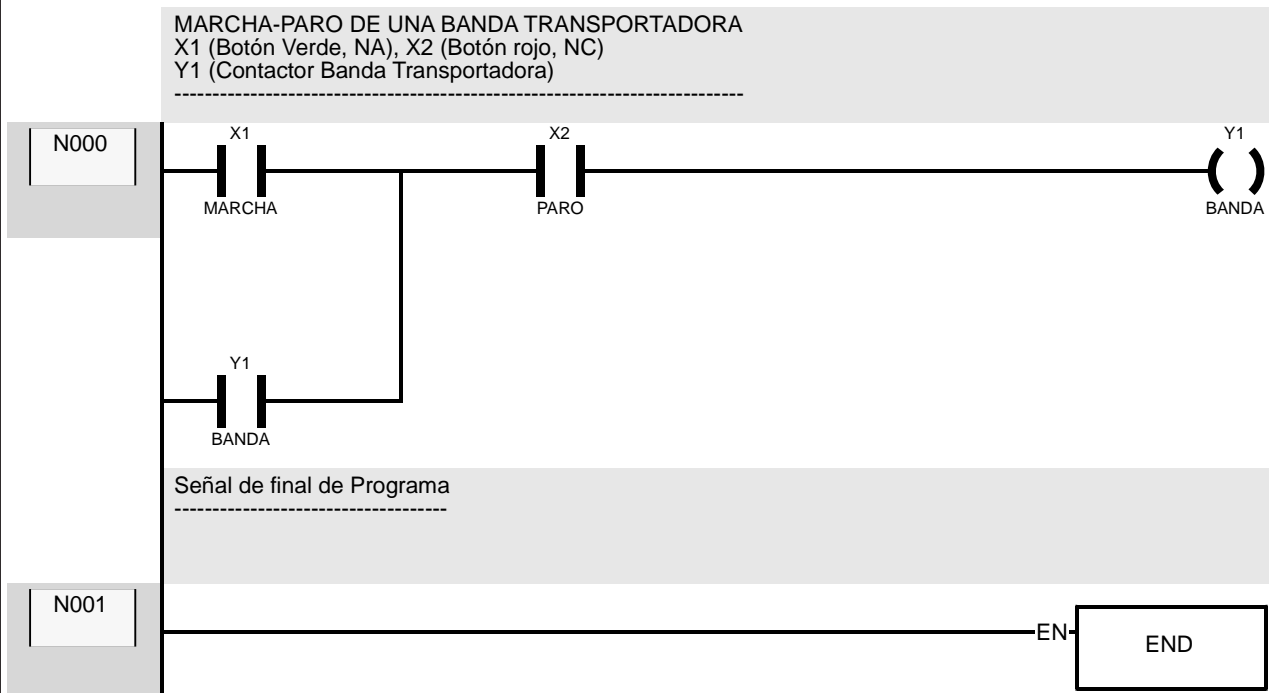
El planteamiento es el siguiente:

* Disponemos de dos botones:

- 1 verde de Marcha, normalmente abierto, NA (conectado a la entrada X1)
- 1 rojo de Paro, Normalmente cerrado, NC (conectado a la entrada X2).

* Hay un contactor, de arranque directo, para el motor de la banda transportadora, que hay que activar con la salida Y1.

Veamos el programa que realiza esta maniobra.

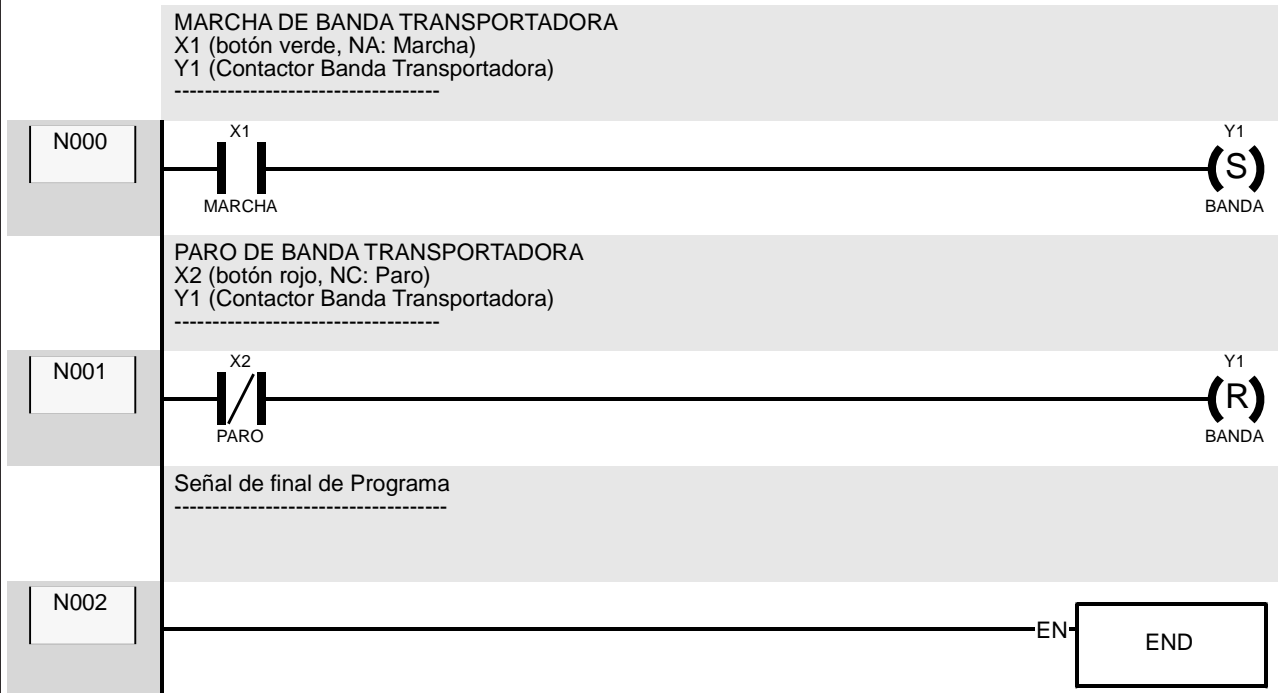


Al pulsar el botón verde (**Marcha**) y no estar pulsado el botón rojo (**Paro**), se activa la salida (**Banda**). Cuando soltamos el botón de Marcha, la señal de excitación de la salida se mantiene realimentada a través de la instrucción NA de Y1.

Esta situación se mantiene (la Banda en marcha) hasta que pulsamos el botón rojo (**Paro**) y abrimos su circuito. Entonces la salida Y1 se desactiva, y al soltar dicho botón de Paro, la salida sigue sesactivada hasta que volvamos a pulsar el botón de Marcha.

Ejercicio 10º

Vamos a realizar la misma maniobra que en el ejercicio anterior, pero en esta ocasión usaremos dos funciones básicas: Set (S) y Reset (R):

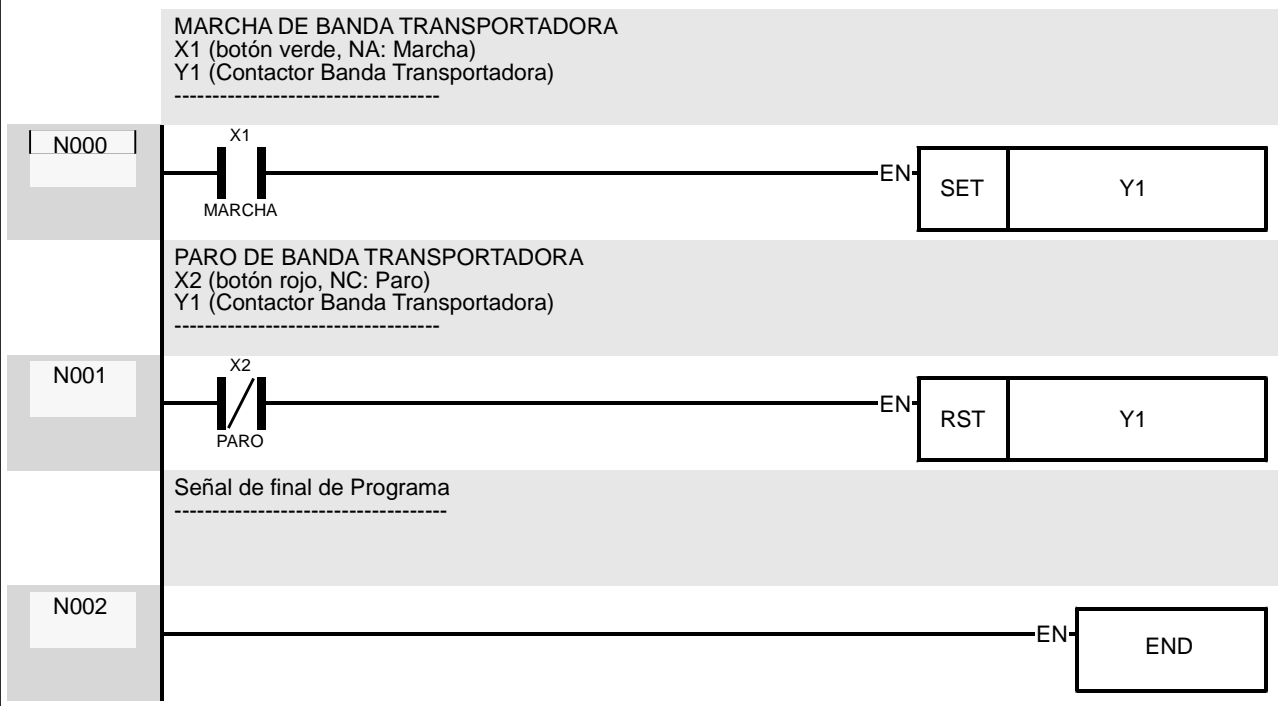


Al activar X1 se enclava la salida Y1. Cuando pulsamos el botón rojo (NC), se abre su contacto y desactivamos la entrada X2. Entonces la instrucción inversa del renglón N001 activa la función reset (R) y la salida Y1 se desactiva.

Estas funciones no son de flanco, por lo tanto se ejecutan en cada Scan en el que el renglón es verdadero.

También se puede realizar esta operación con las funciones SET y RST. Estas últimas también sirven para poner a "1" todos los bits de un registro (16 bits) o de dos registros (32 bits).

Se puede elegir funcionamiento por flanco o por nivel (en el ejemplo siguiente está por nivel).



Ejercicio 11º

En esta práctica vamos a programar un temporizador

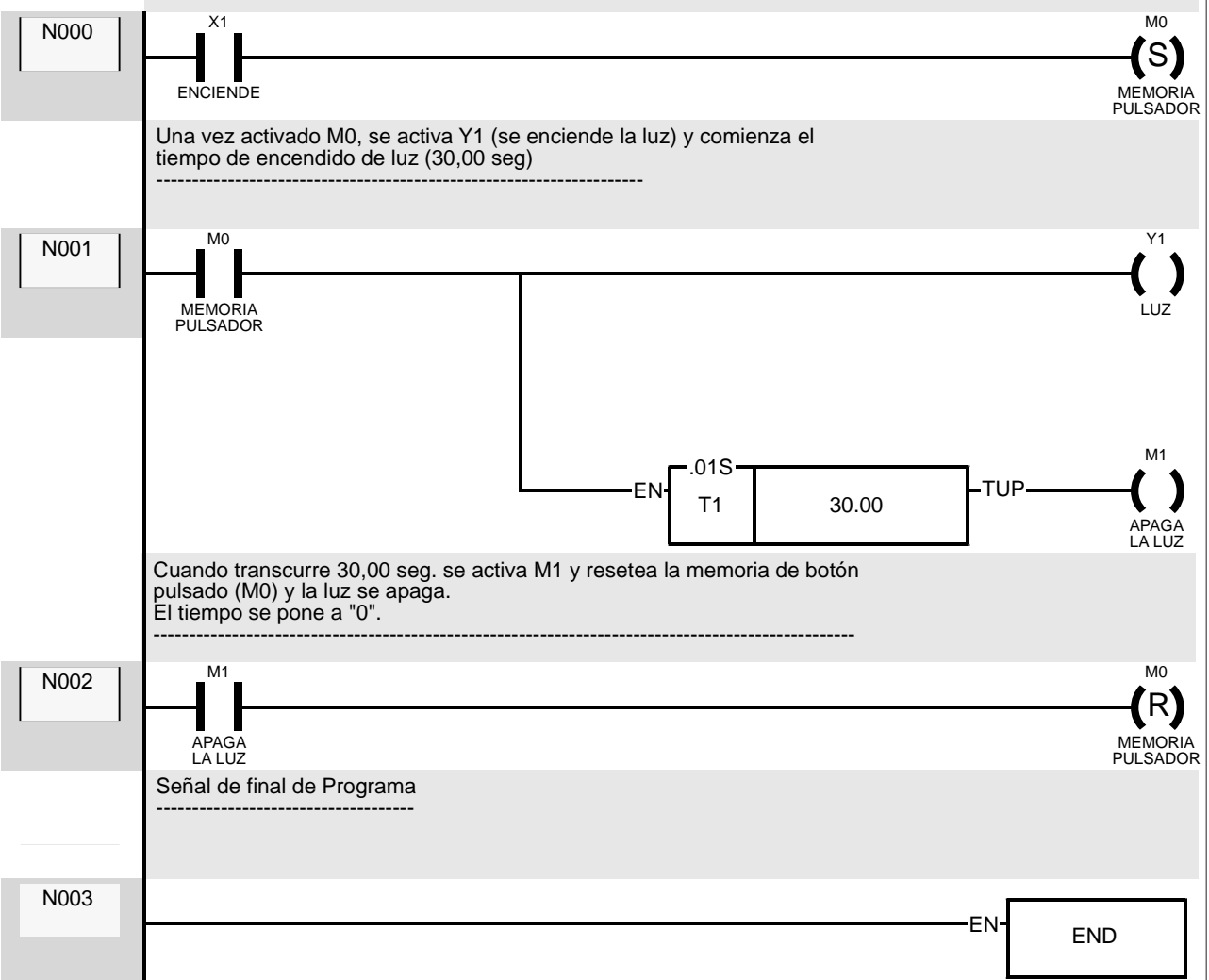
La aplicación consiste en encender una luz, durante un tiempo determinado, cada vez que pulsemos un botón.

* Disponemos de un botón Normalmente Abierto (NA) conectado a la entrada X1.

* Disponemos de un relé, conectado a la salida Y1, que conmuta la tensión a la lámpara que vamos a encender.

Veamos el programa:

Al pulsar el botón y activar la entrada X1, se memoriza esta acción.
Por eso, al soltar el botón, el relé interno M0 permanece a "1"



Al pulsar el botón activamos la entrada X1 y se enclava el bit M0, el cual enciende la luz (activa salida Y1), y pone en marcha el temporizador T1. Pasados 30.0 seg. se activa la salida del temporizador (M1) y en el renglón siguiente se resetea la memoria de botón pulsado (M0) y por lo tanto se apaga la luz y el temporizador se pone a "0".

La situación se queda en esta posición hasta que se vuelva a pulsar el botón de luz, en cuyo caso se repite la maniobra que hemos descrito.

Proponemos al cursillista que compruebe que ocurre si se pulsa el botón repetidas veces mientras transcurre los 30.0 seg

Ejercicio 12º

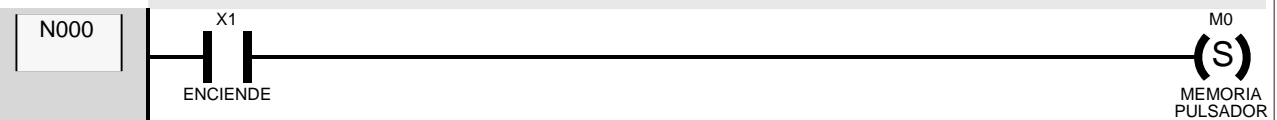
Respondiendo a la pregunta con la que hemos acabado el ejercicio anterior diremos que si pulsamos repetidas veces el botón de encendido, mientras la luz está encendida y el temporizador contando tiempo, no ocurre nada (la luz sigue encendida y el temporizador sigue contando tiempo).

Pues bien, vamos a realizar una variación sobre el ejercicio anterior. Programaremos un temporizador con reset para conseguir que el tiempo empiece de "0" cada vez que pulsamos el botón mientras está temporizando como consecuencia de una pulsación anterior.

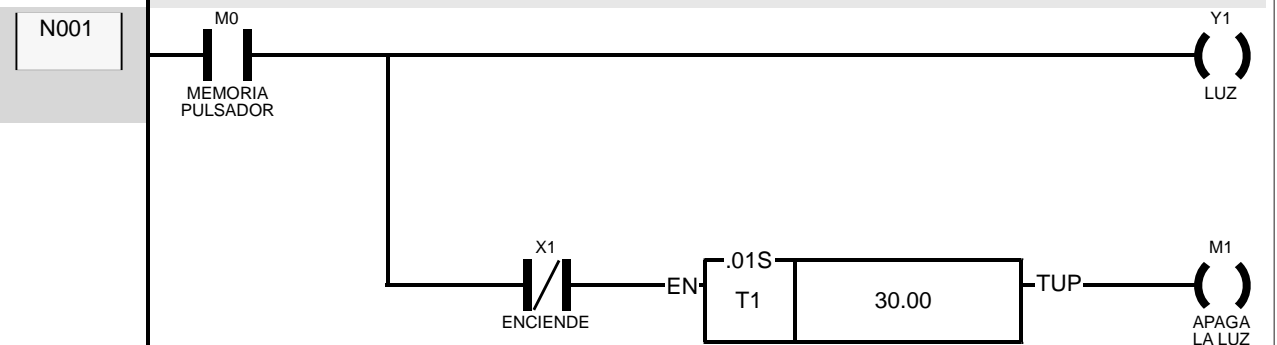
Esta solución puede ser adecuada para encender la luz con un detector de movimiento (en lugar de con un botón), que se active ante el paso de alguna persona o vehículo.

Veamos el programa:

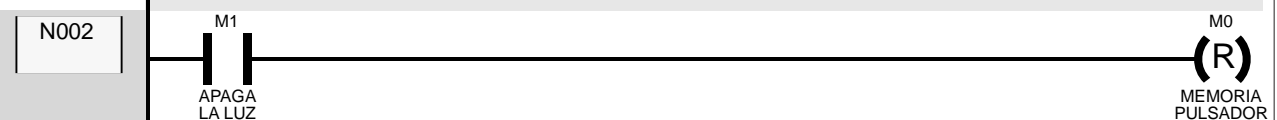
Al pulsar el botón y activar la entrada X1, se memoriza esta acción. Por eso, al soltar el botón, el relé interno M0 permanece a "1"



Una vez activado M0, se activa Y1 (se enciende la luz) y al soltar el botón comienza el tiempo de encendido de luz (30,00 seg) Cada vez que pulsemos el botón se resetea el temporizador y al soltarlo comienza de nuevo el conteo de tiempo.



Cuando transcurre 30,00 seg. se activa M1 y resetea la memoria de botón pulsado (M0) y la luz se apaga. El tiempo se pone a "0".



Señal de final de Programa



Activar la entrada X1 y comprobar el funcionamiento, pulsando repetidas veces dicha entrada para estudiar la respuesta del Automata Programable.

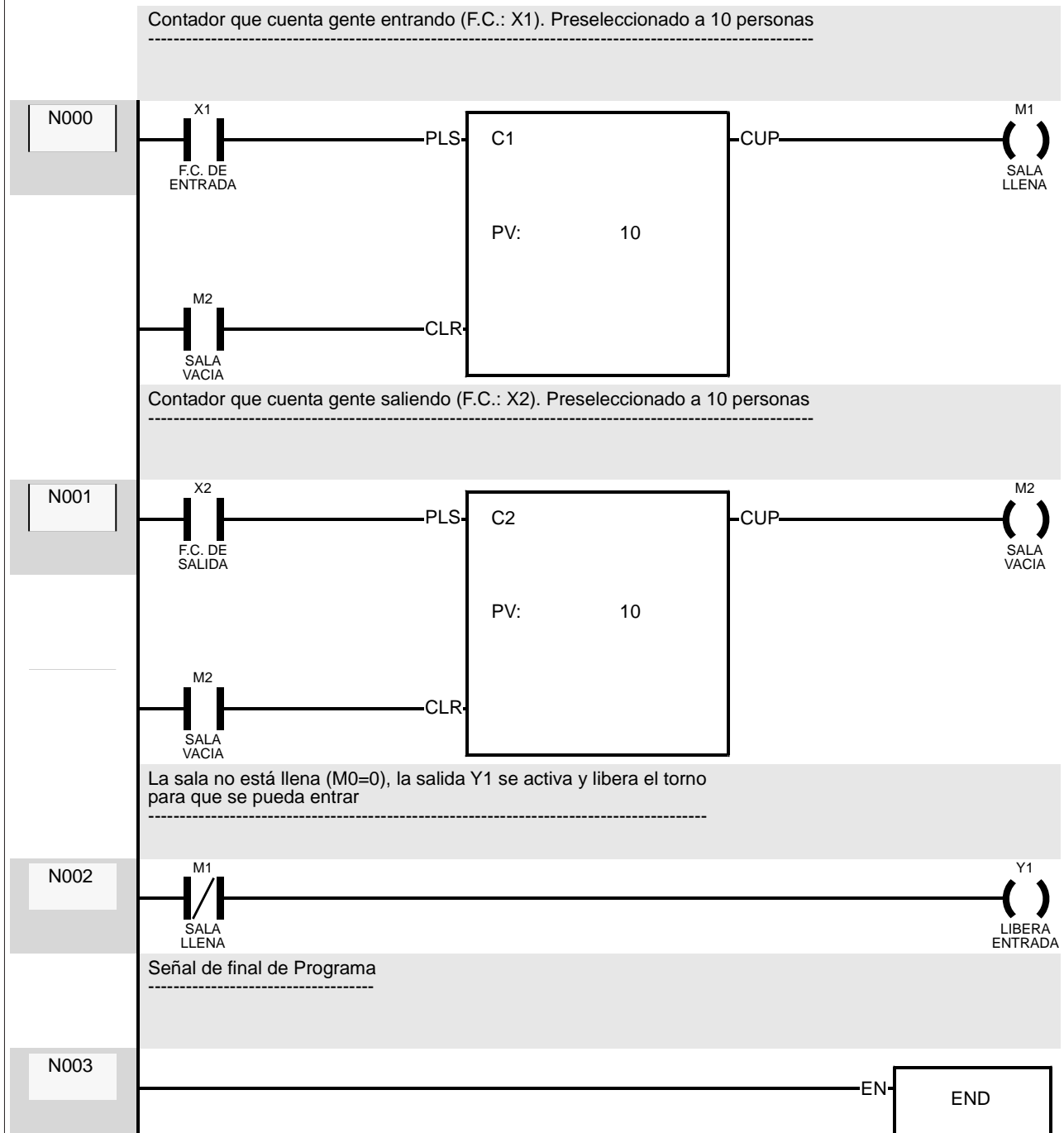
Ejercicio 13º

En este ejercicio vamos a introducir una instrucción nueva: el Contador de pulsos. Para ello vamos a considerar una sala de un museo que tiene un aforo de, por ejemplo 10 persona. Solo pueden entrar 10 visitantes cada vez. despues hay que esperar a que salgan todos para permitir el acceso a otros 10.

Para automatizar este control de acceso, disponemos de un torno en la puerta de entrada que deja pasar a las personas de una en una y otro torno a la salida para permitir a los visitantes salir de uno en uno. Aprovecharemos estos tornos para que, mediante un final de carrera, nos cuenten las personas mientras entran y mientras salen.

El Automata Programable, mediante una de sus salidas, liberará el torno de entrada cuando en la sala no haya nadie y permitirá a la gente que entre hasta que hayan pasado 10 personas. En cuyo momento bloqueará el torno para impedir que sigan entrando. Cuando las personas vayan saliendo, por el torno de la puerta de salida, las irá contando, y cuando cuente 10 liberará de nuevo el torno de la entrada. Por lo tanto podrán entrar 10 visitantes más.

Veamos el programa que realiza esta función:



Continuación ejercicio 13º

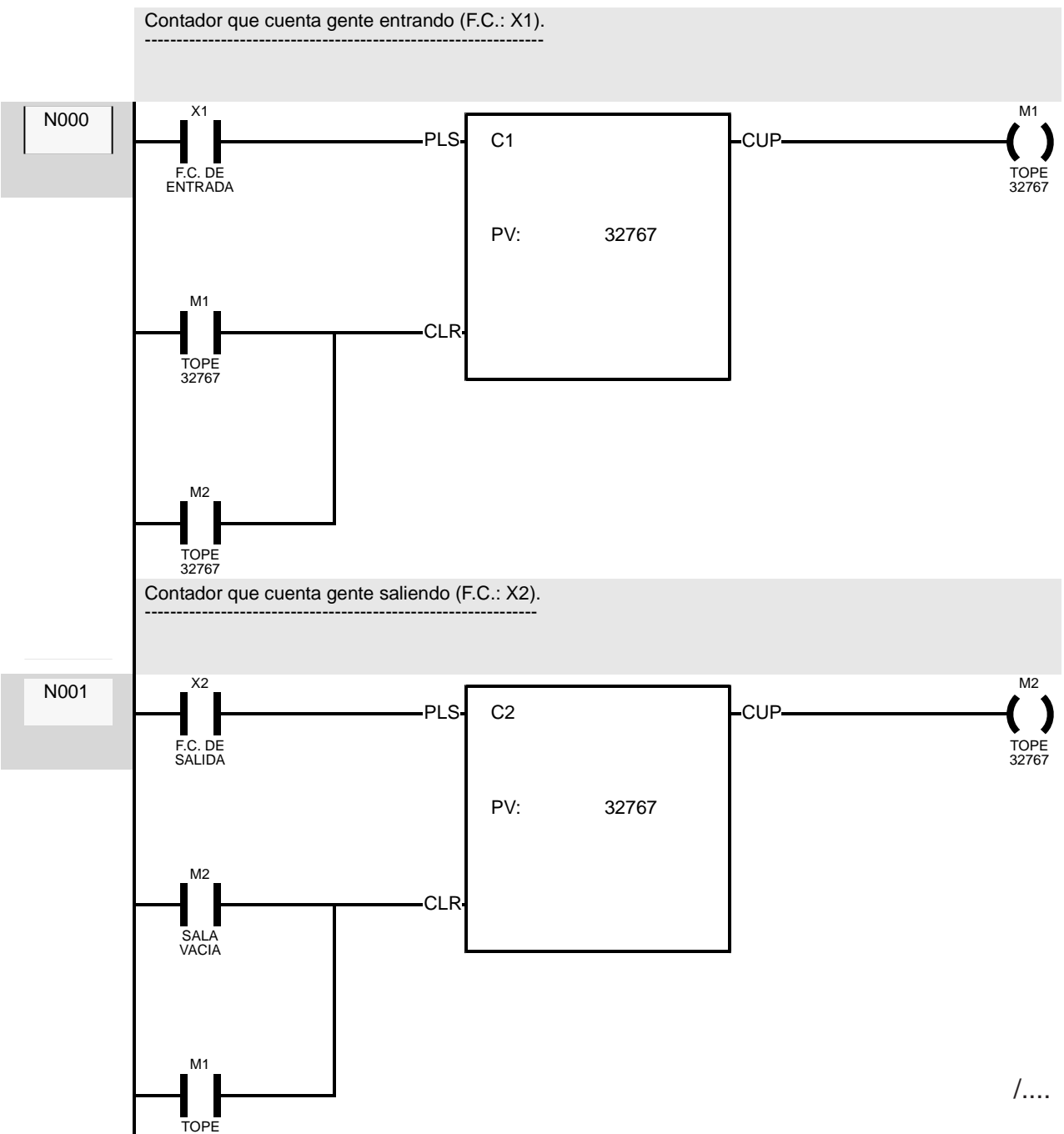
Como podemos ver en el programa de la página anterior, hemos usado dos contadores, uno C1 (cuenta las personas al entrar) y el otro C2 (cuenta a las personas al salir).

Los bits M1 y M2 son los “relés” de salida de dichos contadores preselectores (preseleccionados a 10). Cuando un contador acumula un número de pulsos igual a su preselección, el bit de salida se activa y permanece en esta posición hasta que reseteamos el contador (activando su entrada CLR).

De esta forma, el bit M1 activa la salida Y1 y libera el cerrojo cuando su valor es “0” (sala no llena). Al entrar 10 personas se activa M1 y se bloquea el cerrojo de la puerta de entrada.

El bit M2 es el que resetea esta condición de bloqueo del cerrojo cuando el contador C2 cuenta que han salido 10 personas.

Esta maniobra tiene el inconveniente de que hasta que no sale la última persona de las 10 que han entrado, no se libera el torno de entrada, permitiendo el acceso a otra tanda de 10 visitantes más. Si se quiere permitir el paso a una persona por cada una que salga, manteniendo siempre un máximo de 10, hay que modificar el programa anterior, quedando como sigue:



Continuación ejercicio 13º

..../

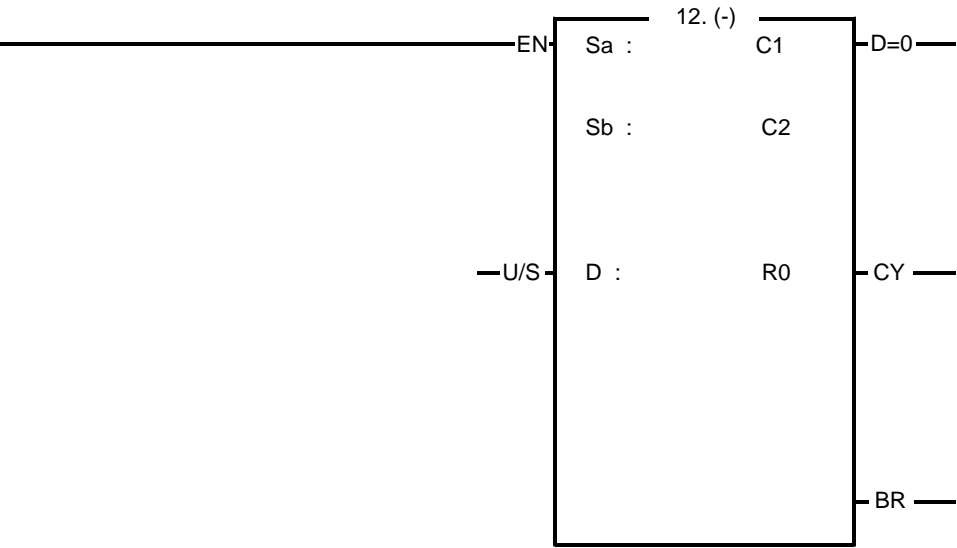
La sala no está llena ($M0=0$), la salida Y1 se activa y libera el torno para que se pueda entrar

N002



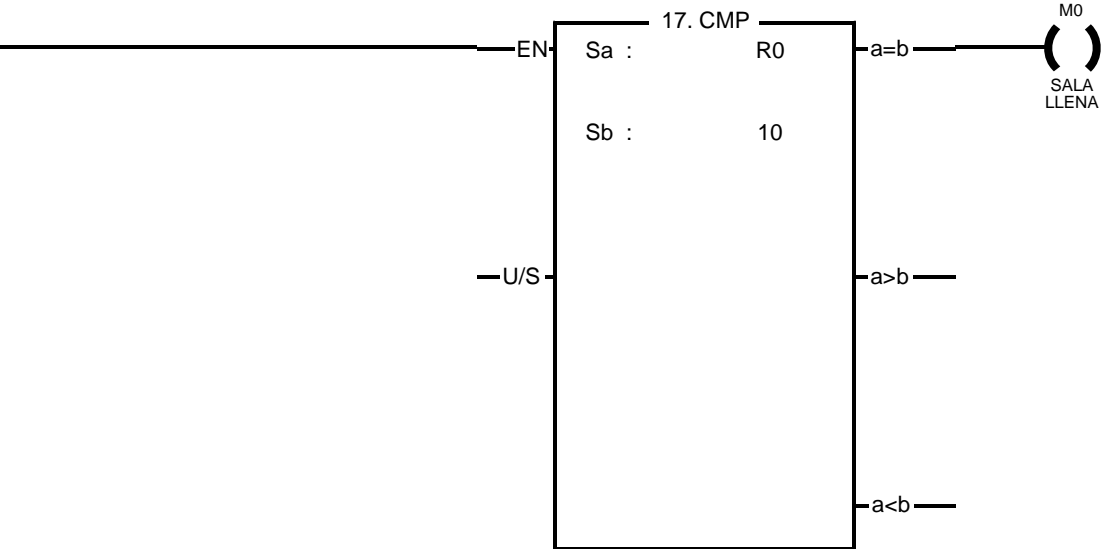
Esta resta calcula el número de personas que hay dentro y lo escribe en R0

N003



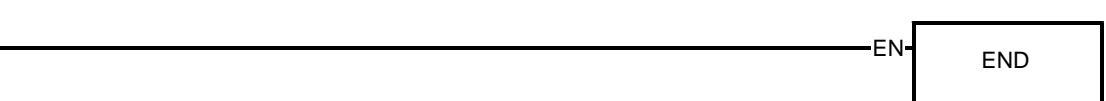
Compara el número de personas dentro (R0) con el tope (10 personas)
Si hay 10 personas ($R0=10$) se activa M0 para bloquear el rotor de entrada

N004



Señal de final de Programa

N005



NOTAS

Ejercicio 14º

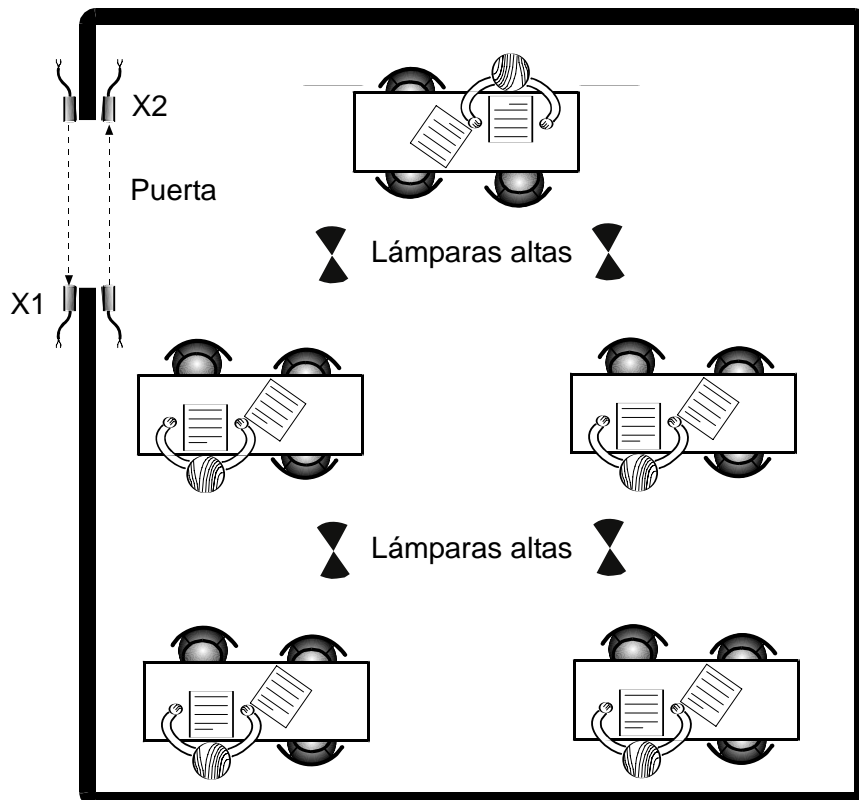
Vamos a realizar el programa para un control del encendido y apagado de la luz general de una biblioteca. El requisito es que la luz se encienda cuando entra la primera persona y se apague al salir la última.

Por lo tanto el programa debe ir contando y acumulando el número de usuarios que entra y descontando cada vez que sale una persona.

Se comparará en número de personas, que hay dentro de la biblioteca, con "0" y si el resultado es mayor que "0", la luz se encenderá o permanecerá encendida. Cuando el número de usuarios sea "0", la luz se apagará y permanecerá así hasta que entre alguien.

El primer aspecto que vamos a explicar es el método para contar las personas, tanto entrando como saliendo. Usaremos dos barreras fotoeléctricas colocadas en la puerta. De esta forma podremos discriminar el sentido de paso de la gente (hacia afuera o hacia adentro) y por lo tanto podremos sumar o restar una persona al valor acumulado de gente en el interior.

Veamos el siguiente croquis explicativo:

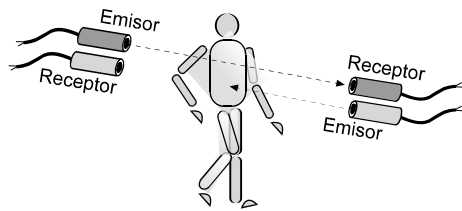


Continuación ejercicio 14º

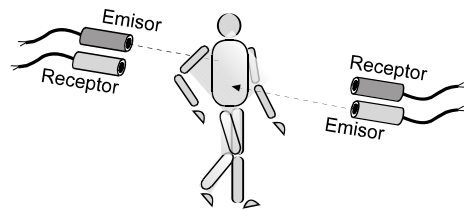
Como podemos ver en el dibujo anterior, solo hay una puerta de acceso por donde se sale y se entra en la biblioteca. Para que el sistema funcione se deben de cumplir dos condiciones:

- 1) La puerta debe ser lo suficientemente estrecha para que tengan que pasar las personas de una en una.
- 2) Las dos barreras fotoeléctricas deben estar juntas para que cuando alguien pase interrumpa las dos al mismo tiempo.

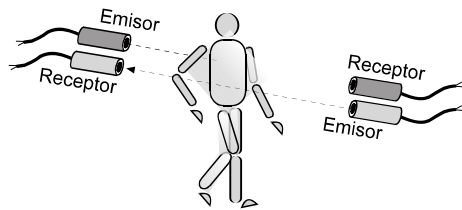
Veamos la secuencia de entrada que se produce cada vez que alguien llega a la biblioteca:



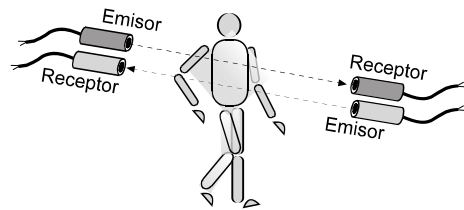
Paso 1º
(Empieza a entrar)



Paso 2º
(Siguen entrando)



Paso 3º
(Continúa entrando)



Paso 4º
(Dentro del todo)

Como vemos en el dibujo, cuando el usuario de la biblioteca llega y comienza su secuencia de entrada, interrumpe la barrera de afuera (activa la entrada X1 del Autómata).

Un instante despues interrumpe la barrera interior, sin dejar de interrumper la de afuera (activa las entradas X1 y X2).

La persona sigue entrando y deja de interrumper la barrera interior (desactiva la entrada X1 y X2 sigue activa).

El último paso de la secuencia de entrada se produce cuando el visitante ha entrado del todo y deja de interrumper la barrera interior (X1 y X2 desactivas).

Esta misma secuencia de entrada, pero a la inversa, se da cuando una persona sale.

El programa del Autómata aprovecha estas posiciones de las dos entradas X1 y X2 para contabilizar cuantas persona hay dentro de la biblioteca.

Además, compara el número de usuarios con "0" y enciende las luces altas si hay alguien (activa la salida Y1). En caso contrario las apaga (desactiva la salida Y1).

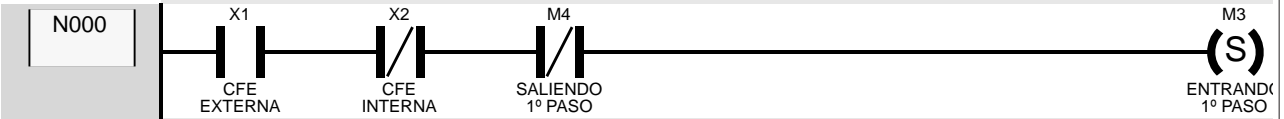
En la siguiente página podemos ver el programa que realiza toda esta función.

Continuación del Ejercicio 14º

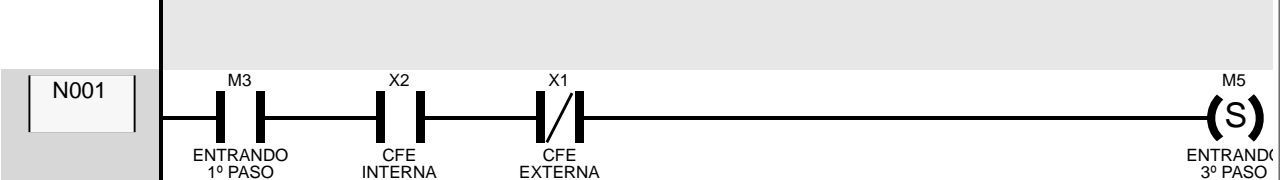
SECUENCIA DE ENTRADA (REGLONES 00-001)

#####

Se activa M3 indicando secuencia de entrada en Paso 1º



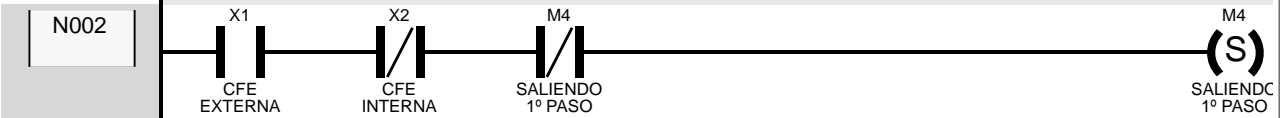
Se activa M5 indicando secuencia de entrada en Paso 3º



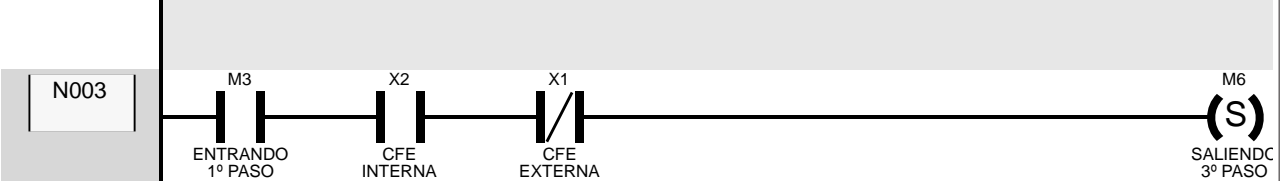
SECUENCIA DE SALIDA (REGLONES 002-003)

#####

Se activa M4 indicando secuencia de salida en Paso 1º



Se activa M6 indicando secuencia de salida en Paso 3º



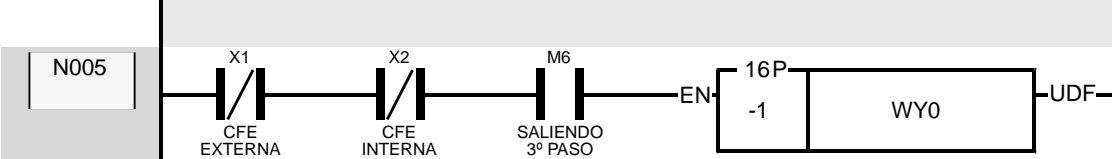
CONTABILIZACION DEL NUMERO DE PERSONAS DENTRO (REGLONES 004-005)

#####

El registro R0 acumula el número de personas dentro. Cuando dejan de interrumpirse las dos CFE's y está entrando se incrementa R0 en +1



El registro R0 acumula el número de personas dentro. Cuando dejan de interrumpirse las dos CFE's y está saliendo se decrementa R0 en -1



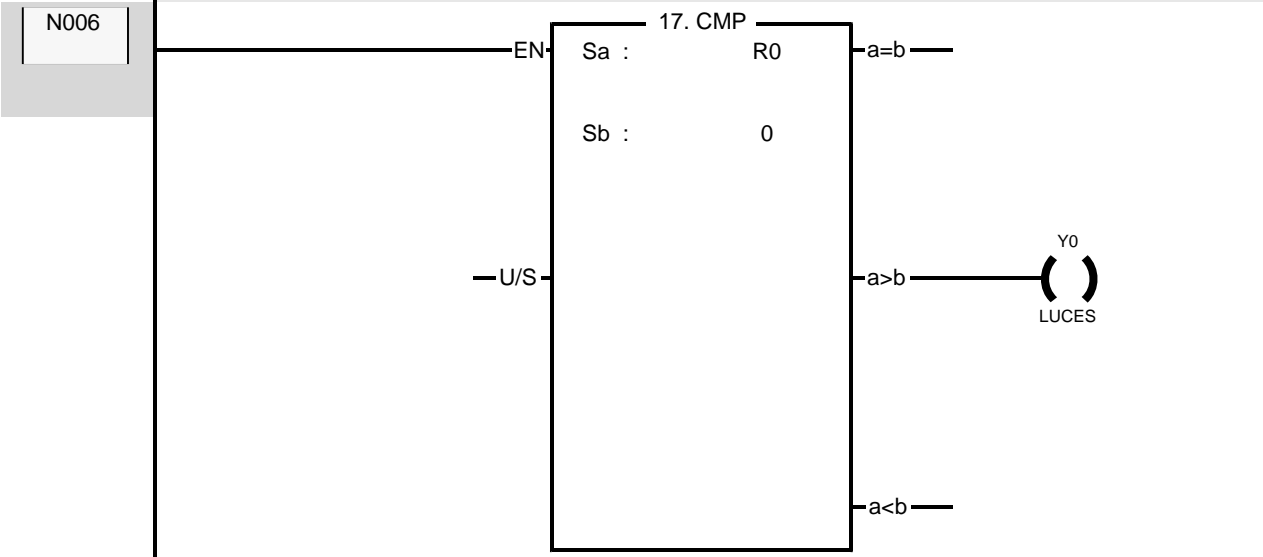
/....

Continuación del ejercicio 14º

..../

COMPARACION DEL NUMERO DE PERSONAS EN LA BIBLIOTECA CON EL NUMERO "0"
#####

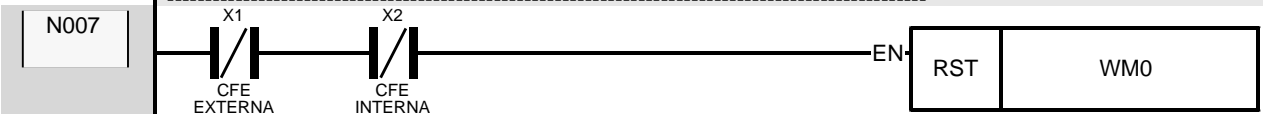
Si $R > 0$ (Hay almenos una persona) se activa Y1 y se encienden las luces



INICIALIZACION DE LAS MEMORIAS M3 A M6

#####

Despues de cada secuencia de entrada o salida se resetean las memorias de paso, para estar listas para la próxima vez que alguien pase.



Señal de final de Programa



Comentaremos brevemente el programa.

En el renglón 000 se memoriza en M3, el primer paso de la secuencia de entrada (barrera externa activa (X1) y barrera interna no activa (X2)).

En el renglón 001 se memoriza en M5, el tercer paso de la secuencia de entrada (barrera externa no activa (X1) y barrera interna activa (X2)). El segundo paso no es necesario memorizarlo, por eso no se programa.

En el renglón 004, cuando ya no estan activas ninguna de las dos barreras, se realiza el incremento de personas dentro, acumulandose en el archivo R0.

La secuencia de salida se realiza en los renglones 002 y 003. Y es en el renglón 005 donde se decrementa el número de personas dentro.

Despues, en el renglón 006, se realiza la comparación del número de personas con "0" y se enciende la luz cuando $R0 > 0$ ($a > b$).

Por último hay que resetear todos los bits que hemos enclavado en la secuencia de entrada o salida. Esta acción se realiza en el renglón 007. El renglón 008 indica el final del programa.

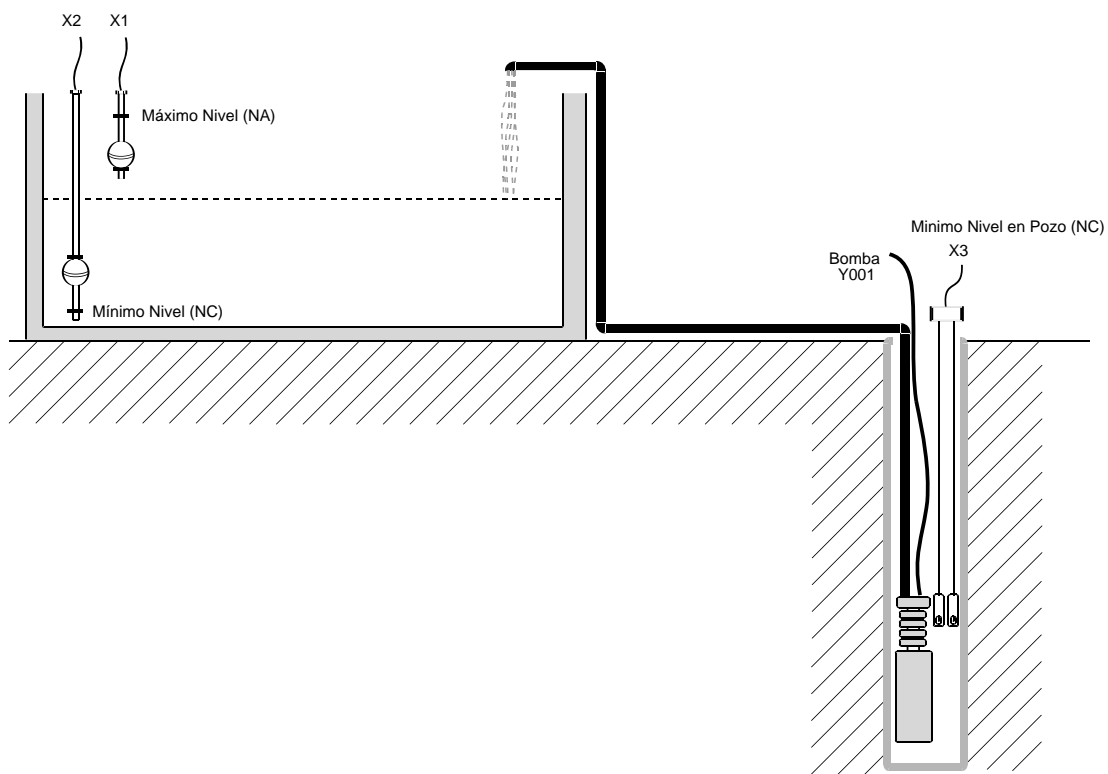
Ejercicio 15º

En este ejercicio vamos a programar una función muy fácil. Controlaremos el llenado de una balsa de riego con agua procedente de un pozo.

La balsa tendrá un control de nivel máximo para evitar que se desborde, y un control de nivel mínimo para pedir agua del pozo cuando haya poca agua.

Para sacar agua del pozo pondremos en marcha una bomba sumergida y dispondremos de un control de nivel mínimo en el pozo para parar la bomba si baja el nivel hídrico, de esta forma evitaremos que la bomba funcione sin agua y se quemé.

Veamos un croquis de la aplicación:



En el dibujo podemos ver, a la izquierda la balsa donde se acumula el agua que proviene del pozo que se puede ver a la derecha.

En la balsa hay colocadas dos boyas de nivel:

- 1) Boya de nivel máximo con contacto NA (abierto cuando el flotador está en posición baja). Esta señal la conectamos a la entrada X1.
- 2) Boya de nivel mínimo con contacto NC (cerrado cuando el flotador está en la posición baja). Esta señal la conectamos a la entrada X2.

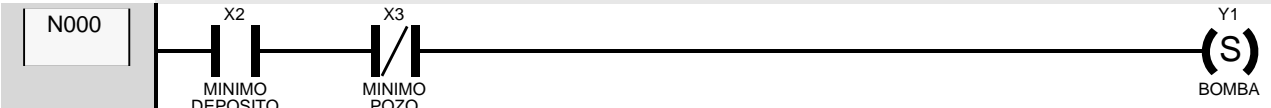
En el pozo hay colocada una sonda de nivel para líquidos conductivos, con señal NC (dará señal cuando no haya agua en el pozo). Esta señal la conectaremos a la entrada X3.

La bomba la pondremos en marcha y la pararemos con la salida Y1.

El programa es sencillo, vamos a verlo:

Continuación ejercicio 15º

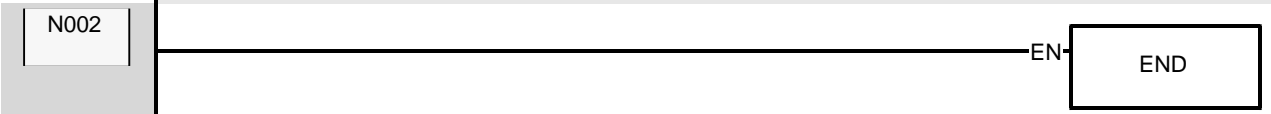
CONTROL DE LLENADO DE UN DEPOSITO
 Cuando la boya de nivel mínimo en el depósito no flota, activa X2 y si hay agua en el pozo, se activará la salida Y1 (Bomba)



Cuando La boya de nivel máximo flota se activa la entrada X1 y resetea Y1 por lo tanto se para la bomba. Tambien se para la bomba si hay señal de nivel mínimo en el pozo.



Señal de final de Programa



Como podemos ver en el programa, en el renglón 000 están ajustadas las condiciones de puesta en marcha de la bomba, cuando hay señal de nivel mínimo en el depósito y si no hay señal de nivel mínimo en el pozo.

La bomba se parará, según lo programado en el renglón 001, cuando hay señal de nivel máximo en el depósito o cuando haya señal de nivel mínimo en el pozo.

Ejercicio 16º

En esta ocasión vamos a usar unas funciones que son muy prácticas y se usan con mucha frecuencia. Se trata de la función transferencia (MOV) de valor de una constante o de una variable a un registro determinado, y de la función lógica puerta “Y” (AND).

Para explicar el uso de estas funciones vamos a recurrir a una aplicación práctica. Determinaremos el tiempo de encendido de una luz en un lugar de paso (corredor, pasillo, etc.) mediante un ajuste digital.

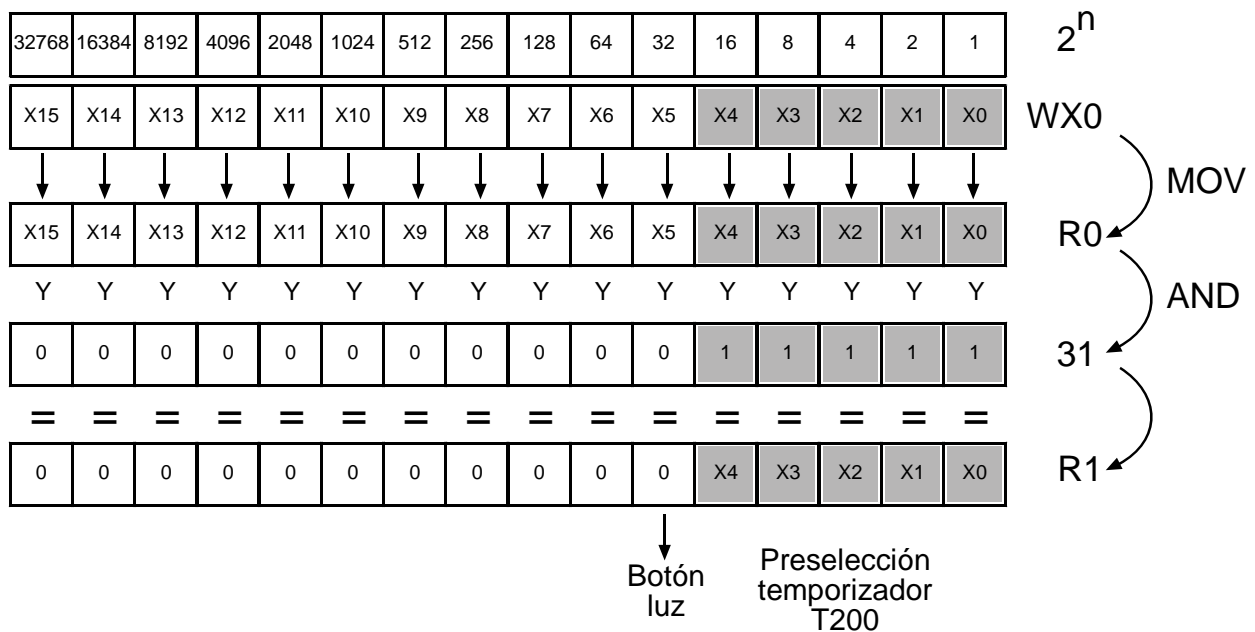
Usaremos las entradas X0 a X4 como preselector del tiempo de luz, y la entrada X5 será la entrada del pulsador de encendido.

Para realizar esta maniobra cambiaremos el valor de la preselección de un temporizador según el código binario que forman las entradas, X0 a X4, del Automata. Podremos obtener un número de 0 a 31.

En cada ciclo de scan se transfiere (con la función MOV) el valor binario de la palabra de 16 bits formada por las entradas X0 a X16, al registro R0.

Despues se filtra (con la función AND) el valor de las entradas X5 a X15 y solo se tiene en cuenta el valor de las entradas X0 a X4, alojandose este valor en el registro R1 que usaremos de preselección del temporizador T200 (con resolución de 1seg.)

Veamos el croquis explicativo:



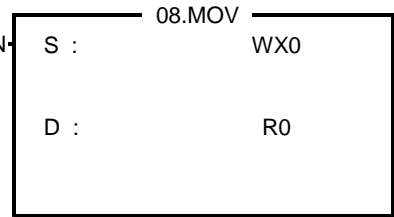
La luz se enciende (Y1 = 1) con la memoria de entrada X5 accionada (M0). Cuando M1 se activa al final de la temporización, se resetea la memoria (M0) y se apaga la luz (Y1 = 0). M1 tambien se pone a “0”.

En la página siguiente vemos el programa correspondiente:

Continuación ejercicio 16º

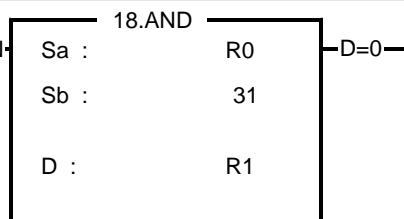
Traslada la palabra de Entradas WX0 (X0 a X15) al registro R0

N000



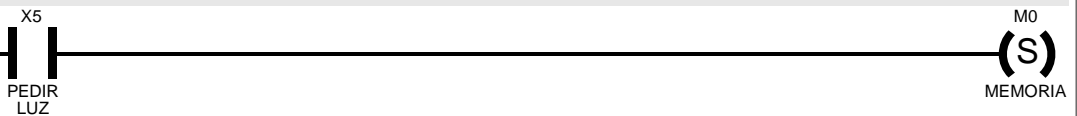
Filtra el valor de las entradas x5 a x16. Solo se tienen en cuenta las entradas X0 a X4 como valor de la preselección del Temporizador Este valor (0 a 31) se aloja en el registro R1 (Preselección del T200)

N001



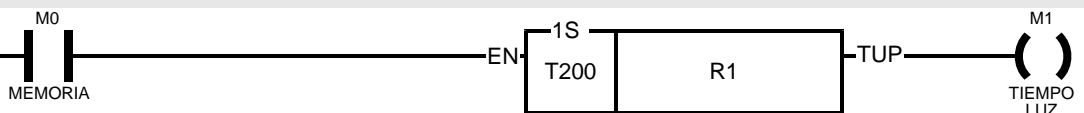
Se pulsa y activa X5 para encender la luz y se memoriza esta pulsación con M0

N002



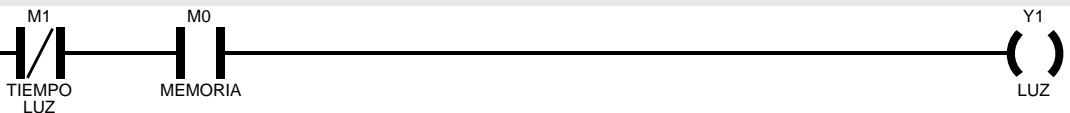
Se cuenta el tiempo de luz. Hasta R1 segundos

N003



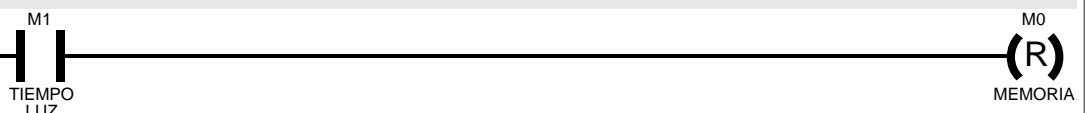
Se enciende la luz con la memoria de pulsación y mientras no ha transcurrido el tiempo de luz

N004



Pasado el tiempo de luz se resetea la memoria de pulsación con lo que se resetea el T200 y se apaga la luz

N005



Señal de final de Programa

N006

