

L^AT_EX- Practica 14:

Sistema de clasificación de dos colores

MIGUEL ÁNGEL MENDOZA HERNÁNDEZ

registro: 20110144

Grupo: 5F

Hidráulica, Neumática y Sensores

13 de marzo de 2023

1. Objetivo de la práctica

Realizar la programación de un conjunto de tres bandas transportadoras para el control de separación de productos a partir de su color con el uso de 7 sensores. Esto es, 2 sensores de inicio y final para cada banda transportadora y un sensor de visión artificial digital para detección de color que regresa un cero lógico para productos color verde y un uno lógico para productos de color rojo. Cada producto atraviesa la banda transportadora principal para después ser separada entre dos caminos de una bifurcación en las que se separan por su color y debemos de controlar la activación y desactivación de cada banda transportadora para cuando pasa a través de ella un producto a la vez.

2. Desarrollo Teórico

Comenzamos a resolver el problema analizando cada uno de los posibles estados en los que pueden encontrarse nuestras entradas y salidas. La siguiente tabla muestra una descripción acerca de esto.

Entradas	Salidas
S_0 : Detección del producto en el inicio de la primer banda transportadora.	M1: motor para primer banda transportadora
S_1 : Detección del producto en el final de la primer banda transportadora.	MP: Motor para puerta selectora.
S_{0V} : Detección del producto en el inicio de la banda transportadora para productos color verde.	MV: Motor para banda transportadora para productos color verde.
S_{1V} : Detección del producto en el final de la banda transportadora para productos color verde.	MR: Motor para banda transportadora para productos color rojo.
S_{0R} : Detección del producto en el inicio de la banda transportadora para productos color rojo.	
S_{1R} : Detección del producto en el final de la banda transportadora para productos color rojo.	
Arranque: Inicia el programa.	
Paro: Detiene el programa.	

Cuadro 1: Descripción entradas y salidas

Para resolver este problema debemos de coordinar cuatro salidas del plc las cuales se tratan de tres motores para cada una de las bandas transportadoras y un motor que nos permite modificar el camino que tomará el producto dependiendo de su color. Por lo tanto, debemos de obtener cuatro ecuaciones lógicas para la programación en escalera de la solución. Empezamos a resolver el problema tomando en cuenta qué variables de entrada se supone deben de activar y desactivar cada una de las salidas. Esto, para poder reducir la cantidad de entrada de cada tabla de verdad y así resolver cada una sin un esfuerzo innecesario.

Elaboramos un diagrama de transición de estados:

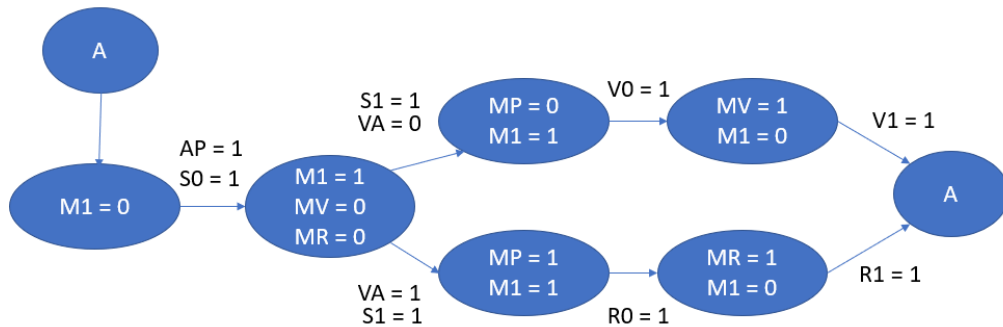


Figura 1: Diagrama de transición de estados

Entonces, realizando este análisis tendremos la siguiente relación de dependencias:

Salida	DEPENDENCIA		
	Activación	Desactivación	En serie con
M1	S0	R0 ó V0	AP
MV	S1 y MP	V1	AP y VA negada
MR	S1 y MP	R1	AP y VA
MP	VA	S0	nada

Entonces comenzamos a analizar cada salida con las respectivas entradas que anotamos en la tabla anterior. Comenzando con M1, tenemos:

Estado	S0	R0	V0	M1
1	0	0	0	0 (estado anterior)
2	1	0	0	1
3	0	0	0	1 (estado anterior)
4	0	1	0	0
5	0	0	0	0 (estado anterior)
6	0	0	1	0

Entonces tenemos lo siguiente:

$$M_1 = 1(S_0\bar{R}_0\bar{V}_1) + M_1(\bar{S}_0\bar{R}_0\bar{V}_0 = \bar{R}_0\bar{V}_0(S_0 + M_1\bar{S}_0))$$

por tanto:

$$M1 = AP\bar{R}_0\bar{V}_0(M_1 + S_0) \quad (1)$$

Para MP tenemos:

Estado	MP	VA	S1	S0
1	Estado anterior	0	0	0
2	1	1	0	0
3	1	1	1	0
4	Estado anterior	0	1	0

Luego:

$$MP = \bar{S}_0MP(\bar{V}_A\bar{S}_1 + \bar{V}_AS_1) + V_A\bar{S}_1 + V_AS_1$$

$$MP = \bar{S}_0(MP\bar{V}_A(S_1 + \bar{S}_1) + V_A(S_1 + \bar{S}_1))$$

$$MP = \bar{S}_0(MP\bar{V}_A + V_A)$$

$$MP = \bar{S}_0(MP\bar{V}_A + V_A) \quad (2)$$

Para MV y MR tendremos en cuenta el valor de MP y no simplemente la lectura del sensor de visión artificial. Esto es porque MP es una variable que mantendrá su valor hasta al menos volver a colocar otro producto en el inicio.

Sin embargo, el sensor de visión artificial solo estará activado un breve periodo y nos puede causar problemas esto. Por tanto, tomaremos en cuenta esto para el cálculo de las ecuaciones de MR y MV.

Estado	MR	MP	S1	R1
1	1	1	1	0
2	Estado anterior	1	0	0
3	0	1	0	1
4	Estado anterior	1	0	0

$$MR = MP(1(S_1\bar{R}_1 + MR\bar{S}_1\bar{R}_1))$$

$$MR = MP(\bar{R}_1(S_1 + MR\bar{S}_1))$$

$$MR = MPAP\bar{R}_1(S_1 + MR) \quad (3)$$

Estado	MV	MR	V1	S1
1	Estado anterior	0	0	0
2	1	0	0	1
3	Estado anterior	0	0	0
4	0	0	1	0

$$MV = 1(\bar{M}P\bar{V}_1S_1) + MV(\bar{M}P\bar{V}_1\bar{S}_1)$$

$$MV = \bar{M}P\bar{V}_1(S_1 + \bar{S}_1MV)$$

$$MV = AP\bar{M}P\bar{V}_1(S_1 + MV) \quad (4)$$

Para la programación en texto estructurado creamos un diagrama de flujo como el siguiente:

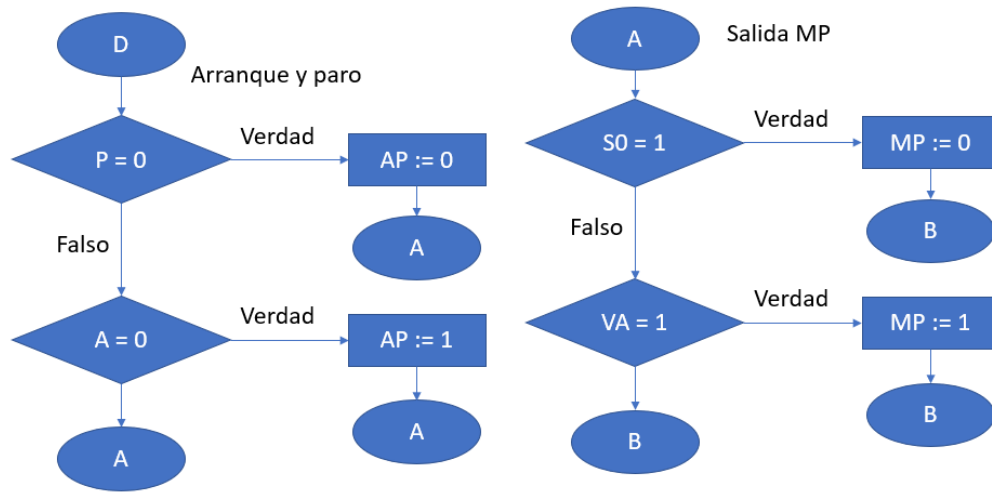


Figura 2: Diagrama de flujo para texto estructurado

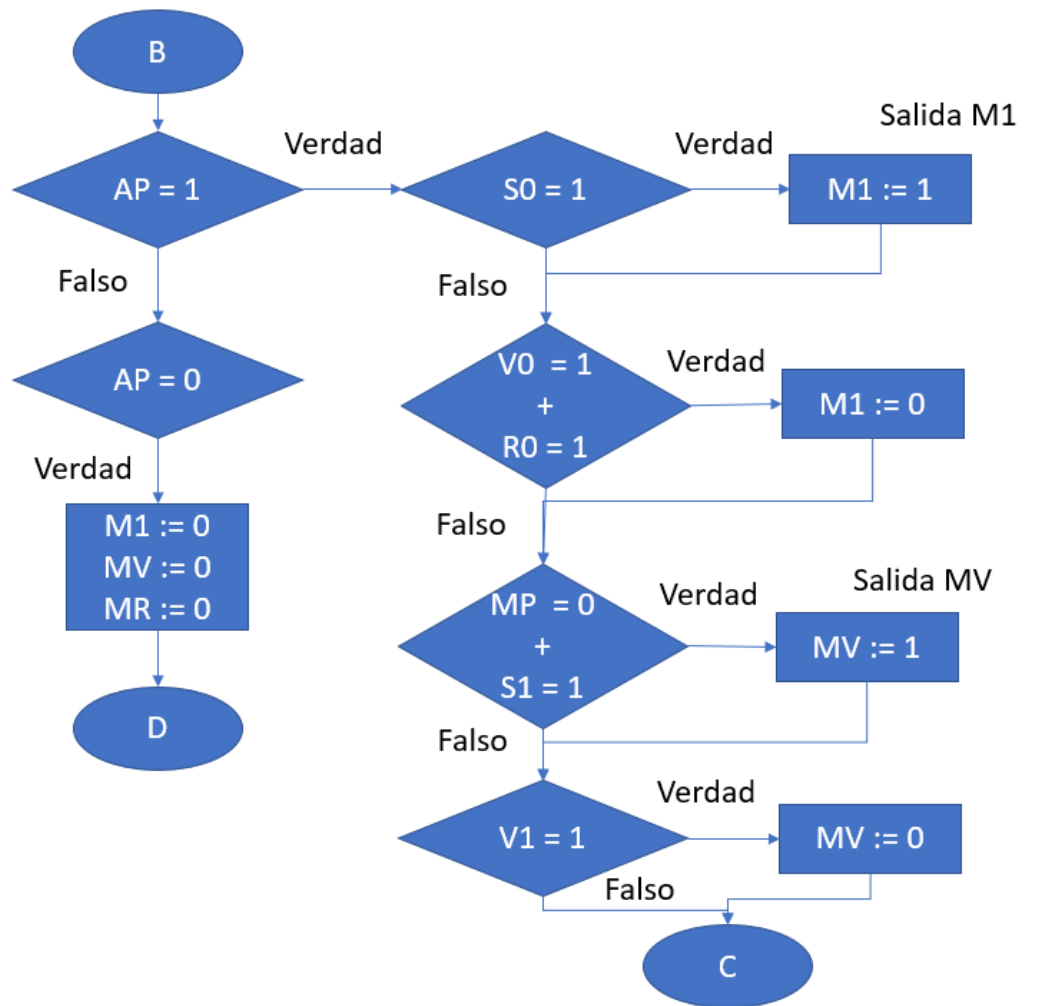


Figura 3: Diagrama de flujo para texto estructurado

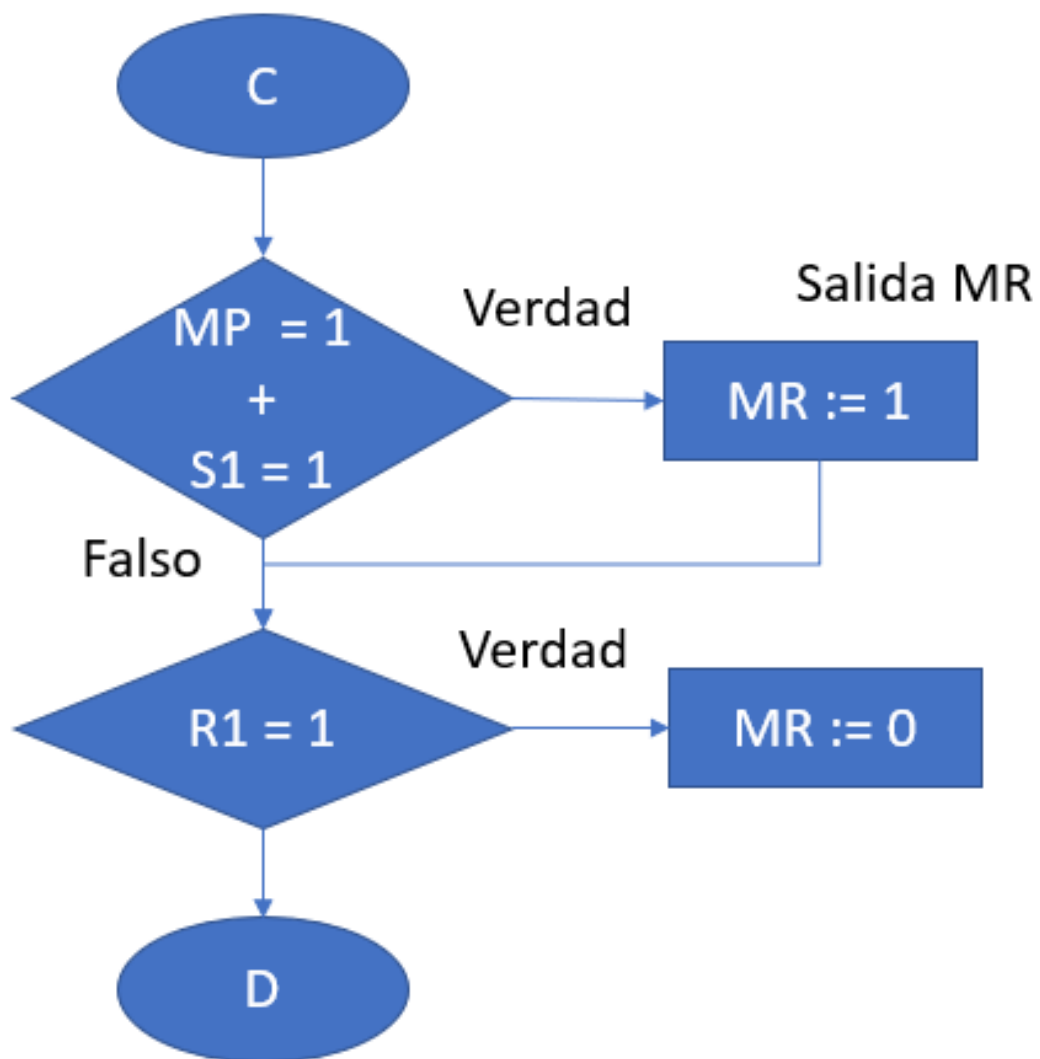


Figura 4: Diagrama de flujo para texto estructurado

3. Simulación del circuito neumático

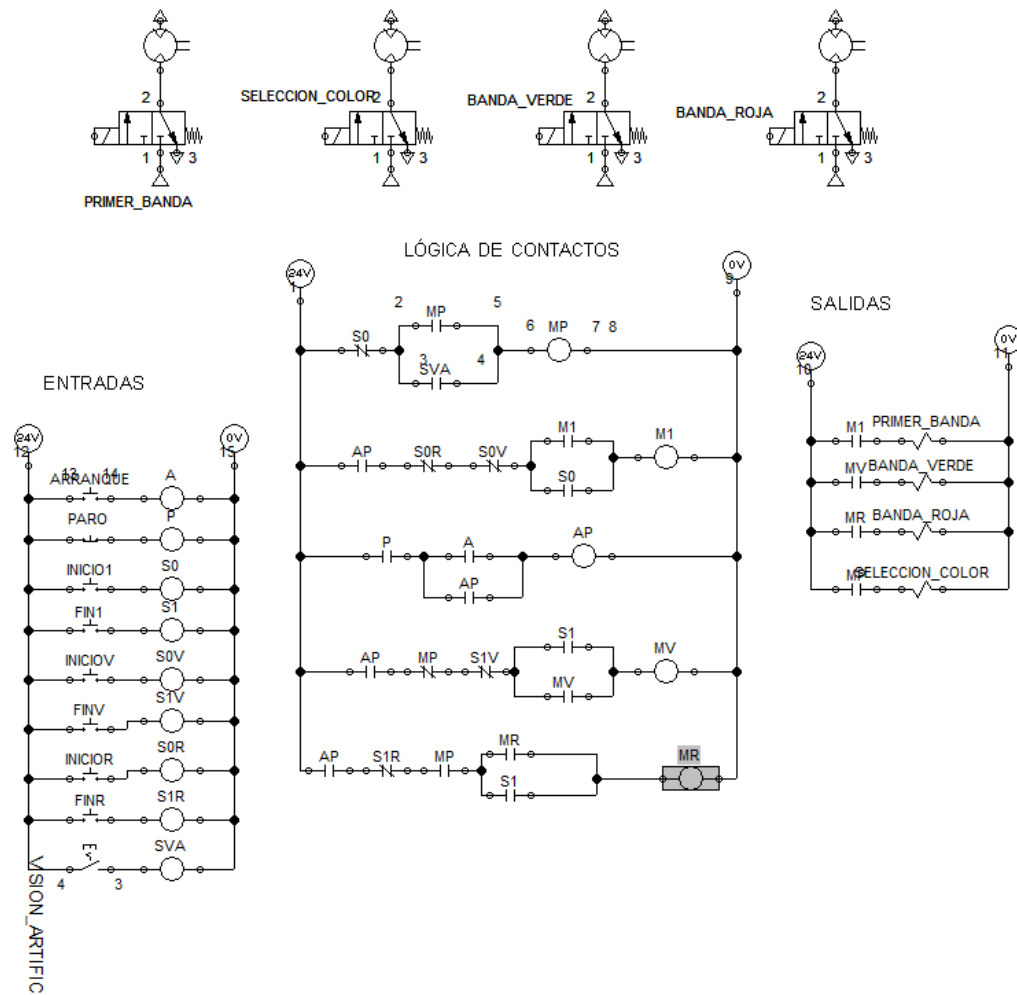


Figura 5: Diagrama neumático del programa en escalera

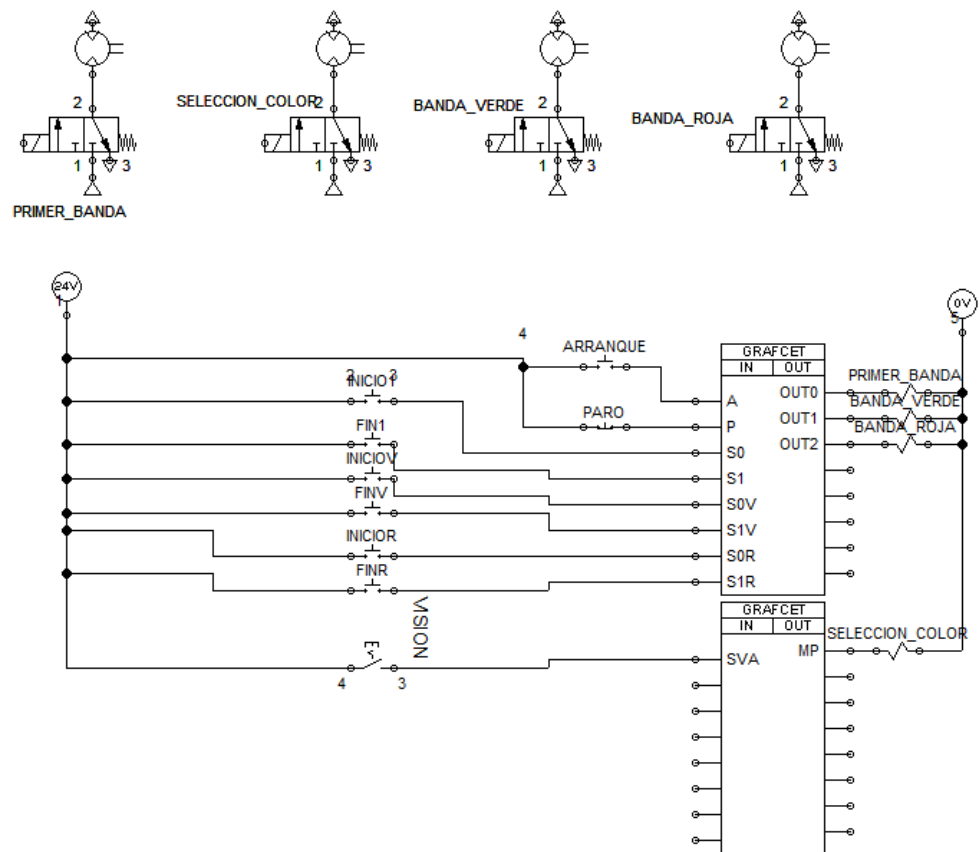


Figura 6: Diagrama neumático del programa en GRAFCET

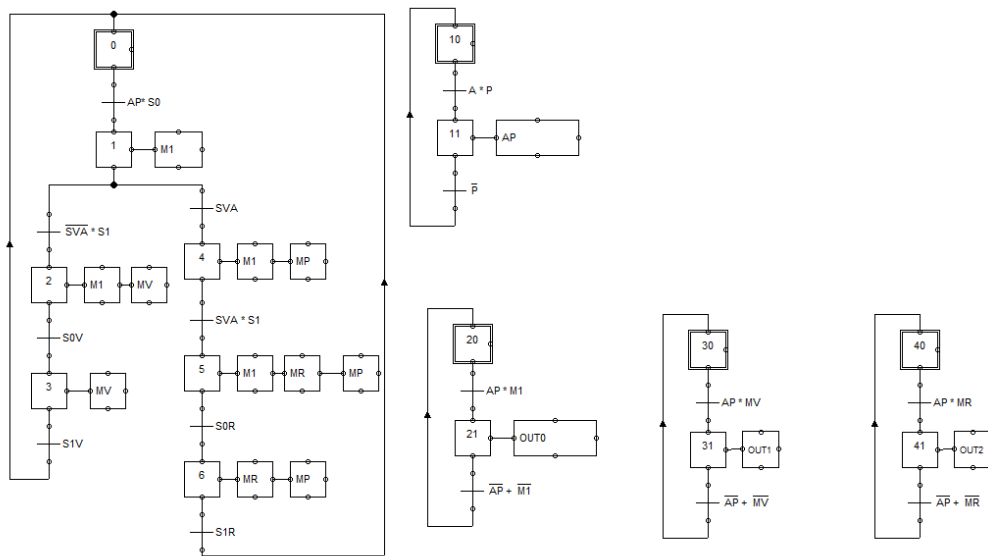


Figura 7: Diagrama neumático del programa en GRAFCET

4. Programa en Escalera

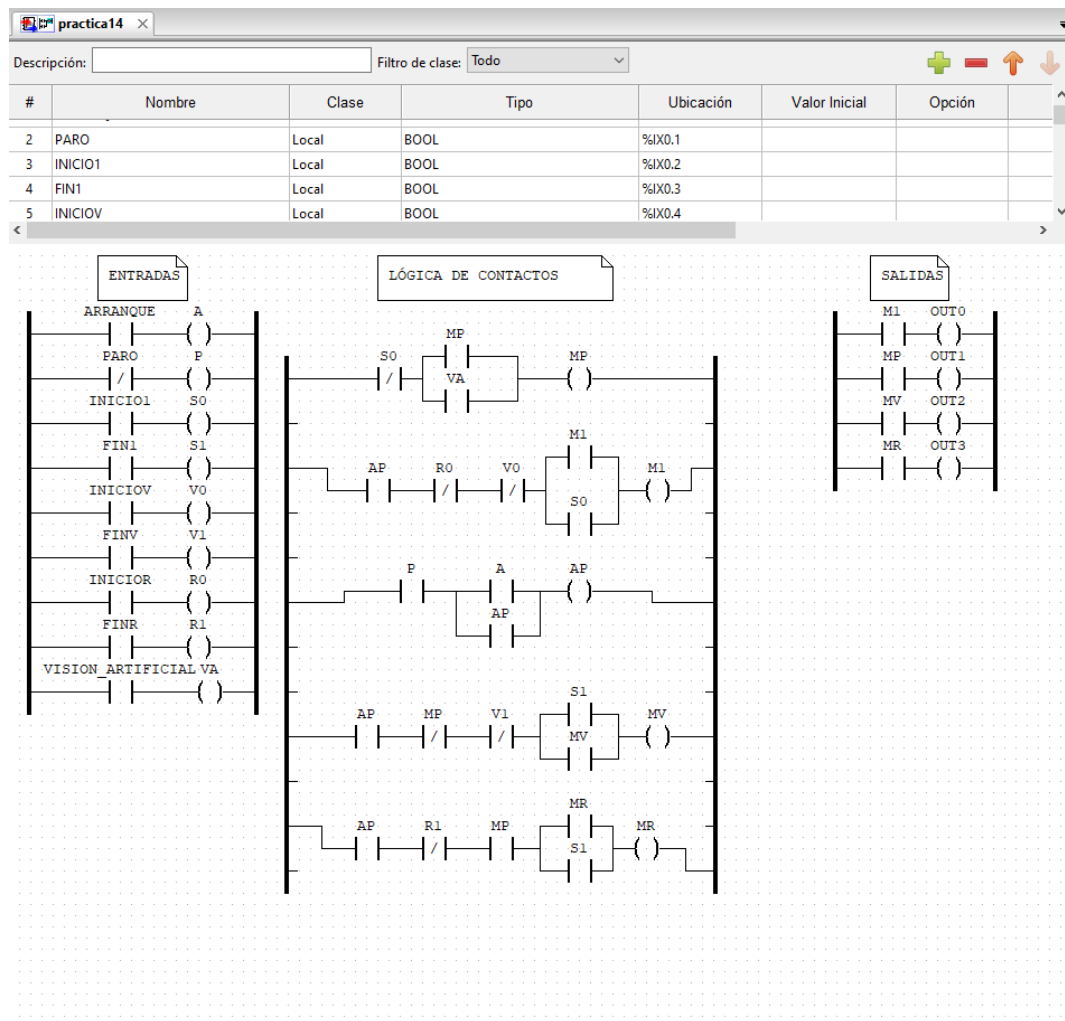


Figura 8: programa escalera en openplc

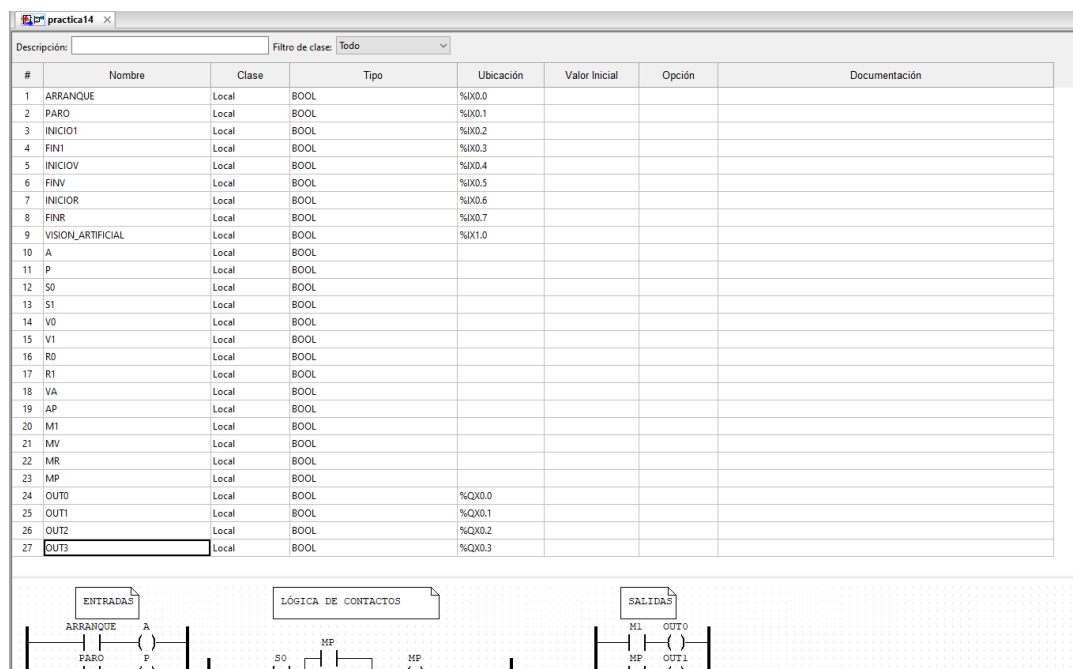


Figura 9: programa escalera en openplc

5. Programa en GRAFCET

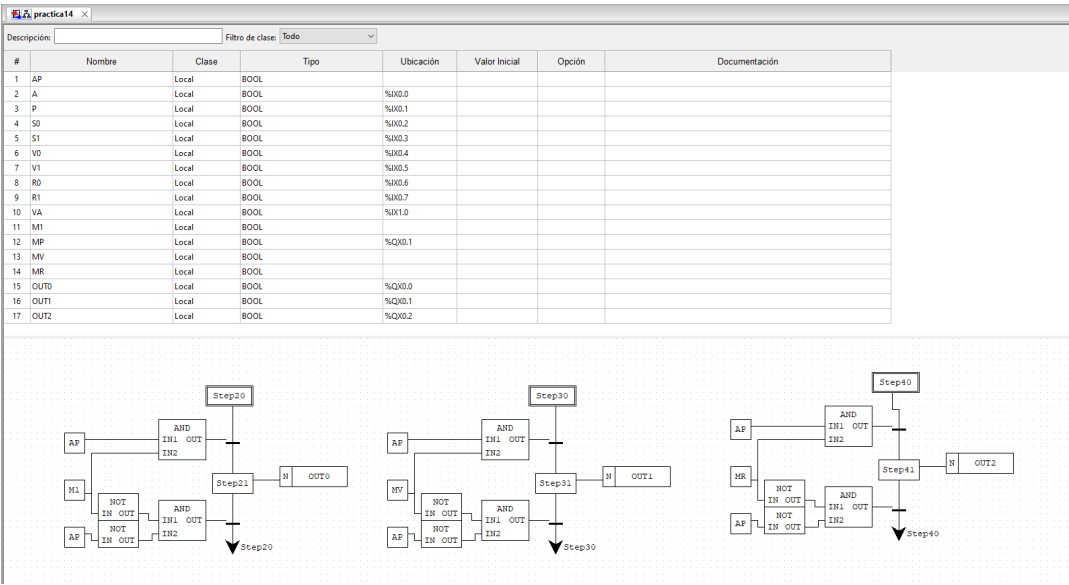


Figura 10: programa GRAFCET en openplc

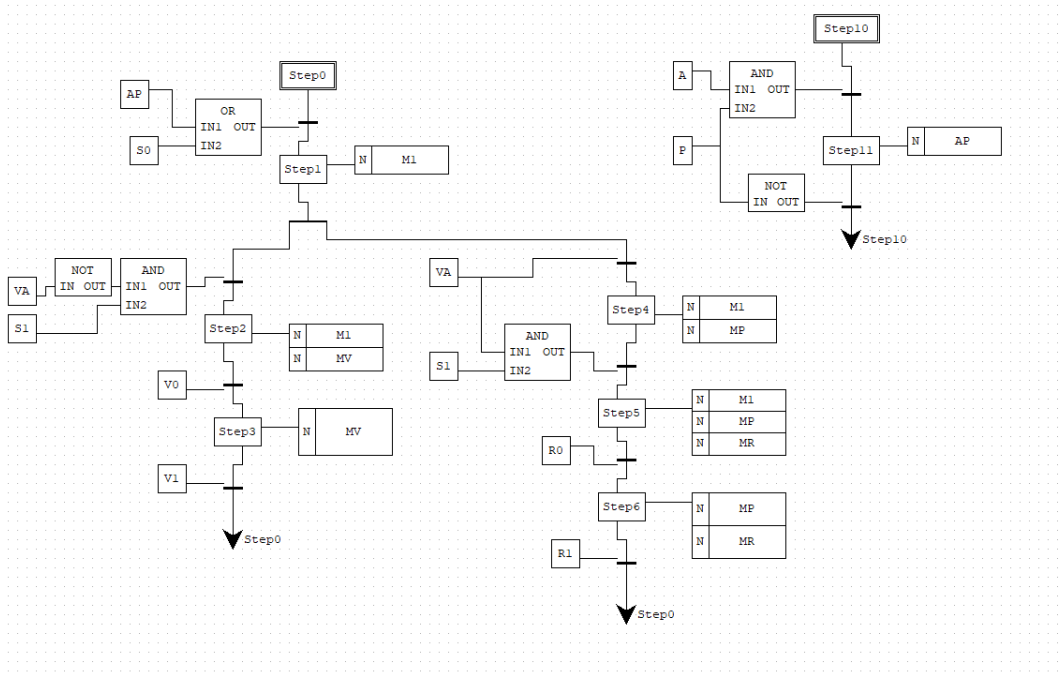


Figura 11: programa GRAFCET en openple

6. Programa texto estructurado

The screenshot shows the OpenPLC IDE interface. At the top, there's a tab labeled "PRACTICA14". Below it, a search bar and a filter dropdown set to "Todo". To the right are icons for adding, deleting, and moving elements. Below this is a table with 8 columns: #, Nombre, Clase, Tipo, Ubicación, Valor Inicial, Opción, and Documentación. The table contains 14 rows of variable declarations. Below the table is a code editor showing a structured text program. The program includes comments in Spanish and logic for controlling a system based on various inputs and outputs.

#	Nombre	Clase	Tipo	Ubicación	Valor Inicial	Opción	Documentación
1	AP	Local	BOOL				
2	ARRANQUE	Local	BOOL	%IX0.0			
3	PARO	Local	BOOL	%IX0.1			
4	S0	Local	BOOL	%IX0.2			
5	S1	Local	BOOL	%IX0.3			
6	V0	Local	BOOL	%IX0.4			
7	V1	Local	BOOL	%IX0.5			
8	R0	Local	BOOL	%IX0.6			
9	R1	Local	BOOL	%IX0.7			
10	VA	Local	BOOL	%IX1.0			
11	M1	Local	BOOL	%QX0.0			
12	MP	Local	BOOL	%QX0.1			
13	MV	Local	BOOL	%QX0.2			
14	MR	Local	BOOL	%QX0.3			

```
1 (* ARRANQUE Y PARO *)
2
3
4 IF PARO = FALSE THEN AP := FALSE;
5 ELSE IF ARRANQUE = TRUE THEN AP := TRUE;
6 END_IF;
7
8
9 (* SALIDA MP *)
10 IF S0 = TRUE THEN MP := FALSE;
11 ELSE IF VA = TRUE THEN MP := TRUE;
12 END_IF;
13
14
15 IF AP = TRUE THEN
16 (* SALIDA M1 *)
17 IF ( S0 = TRUE ) THEN M1 := TRUE;
18 END_IF;
19 IF ( V0 = TRUE OR R0 = TRUE ) THEN
20 M1 := FALSE;
21 END_IF;
22
23 (* SALIDA MV *)
24 IF ( MP = FALSE AND S1 = TRUE ) THEN
25 MV := TRUE;
26 END_IF;
27 IF ( V1 = TRUE ) THEN
28 MV := FALSE;
29 END_IF;
30
```

Figura 12: programa texto estructurado en openplc

7. Circuito físico

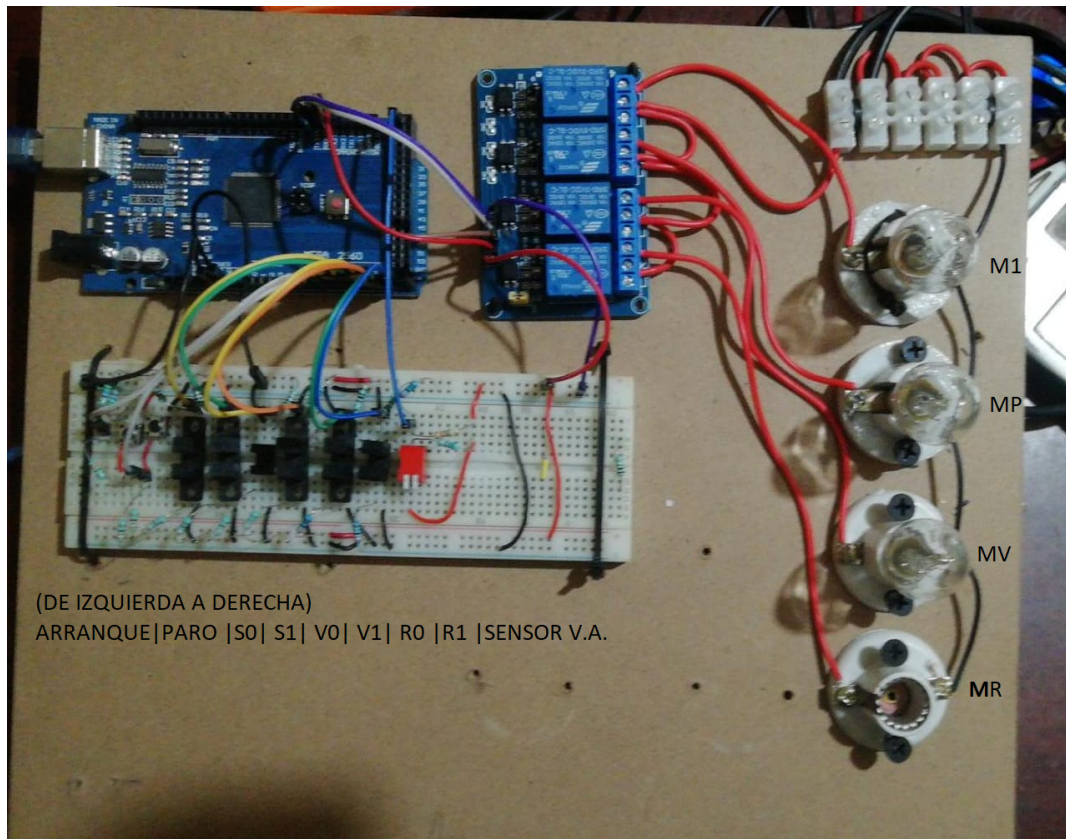


Figura 13: Circuito físico

8. Conclusiones y observaciones

Resulta sencillo el definir desde un principio qué variables de entrada son de las que depende cada salida ya que de esta forma nos ahorramos mucho tiempo al no tener que realizar una tabla de verdad para cada salida que incluya cada una de las variables de entrada de manera innecesaria.