



Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Memoria

PROYECTO FINAL DE CARRERA

**"CORTINA DE AGUA
PROGRAMABLE"**

PFC presentado para optar al título de Ingeniero
Técnico Industrial especialidad Electrónica
Industrial

por **Raúl Espín Torres**

Barcelona, 12 de Enero de 2011

Tutor proyecto: Joan Domingo Peña
Departamento de Ingeniería de Sistemas, Automática e Informática
Industrial (ESAII)
Universitat Politècnica de Catalunya (UPC)

Índice memoria	1
Resumen.....	5
Resum.....	5
Abstract.....	5
Agradecimientos.....	6
Capítulo 1: Introducción.....	7
1.1. Objeto del Proyecto	7
1.2. Alcance del proyecto.....	8
Capítulo 2: Comportamiento de las gotas de agua.....	9
2.1. Caída de las gotas	9
2.2. Sección del chorro agua.....	11
Capítulo 3: Elección de Software.....	12
Capítulo 4: Elección de Electroválvulas.....	13
Capítulo 5: Elección de Hardware.....	15
5.1. Posibles Hardware.....	15
5.2. Elección de componentes.....	15
Capítulo 6: Tratamiento de imágenes.....	19
6.1. Introducción.....	19
6.2. Conceptos básicos.....	20
6.3. Imágenes en Matlab.....	21
Capítulo 7: Programación Matlab.....	22
7.1. Programación GUI.....	22
7.2. Código.....	27
7.2.1. Código del botón <i>Abrir imagen</i>	28
7.2.2. Código del botón <i>Tonos grises</i>	29
7.2.3. Código del botón de desplazamiento(<i>Slider</i>).....	30
7.2.4. Código del botón <i>Ejecutar</i>	31
7.2.5. Organigrama del programa.....	35
7.2.6. Ejecutable del programa.....	36
Capítulo 8: Programación Siemens.....	38
8.1. Introducción.....	38
8.2. Introducción a la programación	39
8.3. Módulos de Organización.....	41

8.3.1. OB1.....	41
8.3.2. Funciones.....	42
8.3.3. Módulo de Función.....	43
8.3.4. Módulo de Función de Sistema.....	43
8.3.5. Función de Sistema.....	43
8.3.6. Bloque de Datos.....	43
8.4. Introducción al programa.....	43
8.5. Programación Hardware	45
8.6. Configuración CP.....	47
8.7. Configuración Enlace	48
8.8. Programación Bloques	49
8.8.1. OB1	49
8.8.1.1 Organigrama OB1	51
8.8.2. FC20 ("BloqueReciboTramas").....	52
8.8.2.1 Organigrama FC20	54
8.8.3. FC22 ("BloqueComparaTramas").....	56
8.8.2.1 Organigrama FC22	58
8.8.4. FC23 ("BloqueActivacionSalidas").....	59
8.8.4.1 Organigrama FC23	61
Capítulo 9: Diseño del Montaje.....	65
9.1. Esquema montaje.....	65
9.2. Elementos principales.....	66
9.2.1. Bomba de agua.....	66
9.2.2. Depósito superior.....	68
9.2.3. Depósito inferior.....	68
9.2.4. Electroválvulas.....	70
9.2.5. Chasis.....	70
9.2.6. Guía para tubos.....	70
9.2.7. Colector.....	70
9.2.8. Cuadro eléctrico.....	71
9.2.8.1. Cálculo de protecciones eléctricas.....	71
Capítulo 10: Prototipo.....	73
10.1. Elementos del prototipo.....	73
10.1.1. Chasis.....	74
10.1.2. Electroválvulas.....	74

10.1.3. Guía para tubos.....	77
10.1.4. Colector.....	78
10.1.5. Depósito inferior.....	78
10.1.6. Bomba de agua.....	79
10.1.7. Depósito superior.....	80
10.1.8. Cuadro eléctrico.....	81
Capítulo 11: Conclusiones.....	83
Capítulo 12: Posibles Mejoras.....	84
Capítulo 13: Bibliografía.....	85
13.1. Referencias bibliográficas.....	85
13.2. Bibliografía de consulta.....	85

Resumen

Este proyecto se basa en el diseño de una cortina de agua programable para fines publicitarios. Esta cortina es capaz de definir pequeñas imágenes con la caída de gotas de agua.

Para ello se ha diseñado una aplicación en ordenador, de tratamiento de imágenes, la cual comunica con un microcontrolador que activa las electroválvulas correspondientes.

Se ha escogido tanto el software como el hardware pensando en que la cortina fuera operativa.

Como resultado de este proyecto, se ha desarrollado un prototipo capaz de reproducir imágenes de 48 x 48 píxeles.

Resum

Aquest projecte es basa en el disseny d'una cortina d'aigua programable per a fins publicitaris. Aquesta cortina és capaç de definir petites imatges, amb la caiguda de les gotes d'aigua.

Per això s'ha dissenyat una aplicació a l'ordinador, de tractament d'imatges, la qual comunica amb un microcontrolador que activa les electrovàlvules corresponents.

S'ha escollit tant el software com el hardware pensant en que la cortina fos operativa.

Com a resultat d'aquest projecte, s'ha desenvolupat un prototip capaç de reproduir imatges de 48 x 48 píxels.

Abstract

This project is based on the design of a programmable waterfall for advertising ends. This waterfall is capable of defining small images by the fall of water drops.

For that reason, a computer application of treatment of images has been designed. It reports to a microcontroller who activates the corresponding solenoid valves.

Both, the software and the hardware, have been chosen thinking that the waterfall has to be operative.

As a result of this project, a prototype capable of reproducing images of 48 x 48 pixels has been developed.

Agradecimientos

En primer lugar agradecer el interés demostrado y el apoyo recibido por parte del tutor del proyecto Joan Domingo.

Agradecer el apoyo demostrado durante toda la carrera de los compañeros de trabajo, y en especial : Fernando, Carlos , Miguel y Félix. Así como los compañeros de universidad, que sin ellos no hubiera sido posible.

Este trabajo va dedicado a toda la familia, que tanto me ha ayudado en estos años y me ha motivado a seguir para adelante. En especial a mis padres, y a mi pareja.

CAPÍTULO 1: INTRODUCCIÓN

1.1. Objeto del proyecto

El proyecto tiene como objeto crear una aplicación y montaje para la creación de una cortina de agua programable con fines publicitarios.

La cortina de agua ha de ser capaz de reproducir imágenes estáticas mediante la caída libre de gotas de agua.

La cortina de agua programable ha de ser operativa, con lo cual se tendrá que elegir el software y hardware necesario para su implementación.

Las imágenes de entrada han de ser tratadas mediante tratamiento de imagen para poder ser vistas en la cortina. Para ello se deberán digitalizar utilizando una aplicación creada en ordenador.

Para poder efectuar las imágenes habrá que comunicar la aplicación con un controlador que a su vez activará las electroválvulas necesarias.

Estas electroválvulas han de ser capaces de reproducir la imagen con lo cual deberán ser de accionamiento rápido.

1.2. Alcance del proyecto

El presente proyecto tiene como alcance el diseño, montaje de prototipo, y puesta en marcha de una cortina de agua programable.

El prototipo de cortina programable de agua ha sido construido para la representación de pequeñas imágenes, tales como logotipos o pequeños textos con carácter publicitario.

El diseño consiste en crear una aplicación en ordenador, capaz de introducirle una foto o imagen en colores. La aplicación transforma la imagen en tonos grises, seguidamente el usuario puede elegir la cantidad de negros y blancos que desea representar, es decir, binarizar la imagen. Finalmente ha de ser capaz de comunicarse con un microcontrolador o autómata programable para que active las electroválvulas necesarias.

Mediante las electroválvulas se observará la imagen con la caída de agua.

Se ha escogido el software y Hardware necesario para llevar a cabo dicho proyecto.

CAPÍTULO 2:

COPORTAMIENTO DE LAS GOTAS DE AGUA

2.1. Caída de las gotas

Un cuerpo en caída libre se comporta de la misma forma que si estuviese en ingravidez. Por tanto, una gota de agua en caída libre, tenderá a adquirir forma de esfera.

El rozamiento del aire puede deformar una gota. A medida que la gota aumenta su velocidad en su caída, el rozamiento aumenta también, disminuyendo su aceleración hasta llegar a la velocidad terminal. A velocidad terminal la velocidad de caída permanece constante. La gota se achata, tomando una forma esferoide.

Si el goteo es suficientemente lento, podemos ver cómo el líquido, antes de caer, se mantiene suspendido en una pequeña esfera que cuelga del resto del líquido, hasta que su peso es demasiado grande y se desprende.

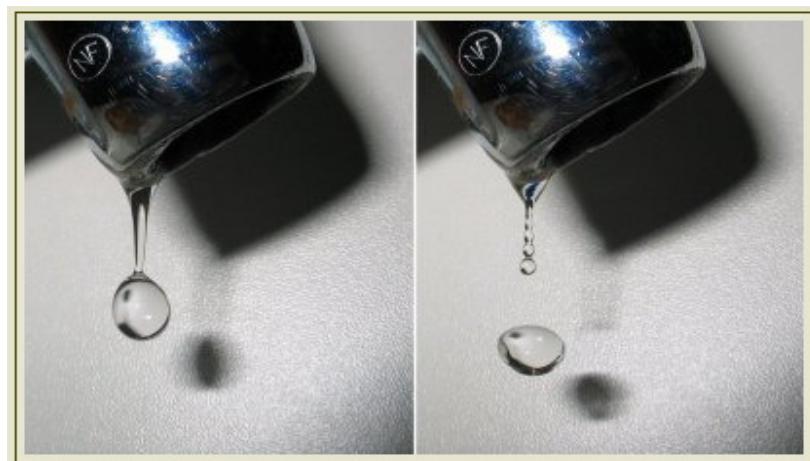


Figura 1. Caída de una gota

La velocidad de caída de una gota viene dada según la ley establecida del físico Stokes en el año 1850 :

$$v = 1,3 \cdot r^2 \cdot 10^6$$

r = el radio de las gotitas en cm.

(1)

Y la velocidad Terminal viene dada por las siguientes fórmulas dependiendo del flujo.

Flujo laminar

Para una esfera de radio R moviéndose en un flujo no turbulento dentro de un fluido de viscosidad η , la velocidad límite viene dada por la ley de Stokes, que postula que la fuerza de resistencia es proporcional a la velocidad. En ese caso la velocidad límite viene dada por:

$$v_\infty = \frac{F}{6\pi\eta R} \quad (2)$$

Flujo turbulento

Para un cuerpo moviéndose en un flujo turbulento en el que se producen remolinos alrededor del cuerpo en movimiento la fuerza de rozamiento depende de v^2 y es proporcional a la resistencia aerodinámica. En ese caso la velocidad límite viene dada por:

$$v_\infty = \sqrt{\frac{2F}{\rho A C_d}} \quad (3)$$

Donde:

v_∞ es la velocidad límite o terminal.

F es el peso del objeto que cae, para el caso de caída libre $F = mg$.

C_d es el coeficiente de resistencia aerodinámica

ρ es la densidad del fluido a través del cual se mueve el objeto,

A es la sección del objeto en dirección transversal a la de movimiento.

Para una gota de agua aproximadamente 9 m/s (32 km/h), dependiendo de su tamaño.

2.2. Sección del chorro de agua

También hay que tener en cuenta que el chorro de agua va disminuyendo de radio a medida que va cayendo hasta que se rompe en gotas.

La siguiente ecuación proporciona la forma de superficie del chorro de agua.

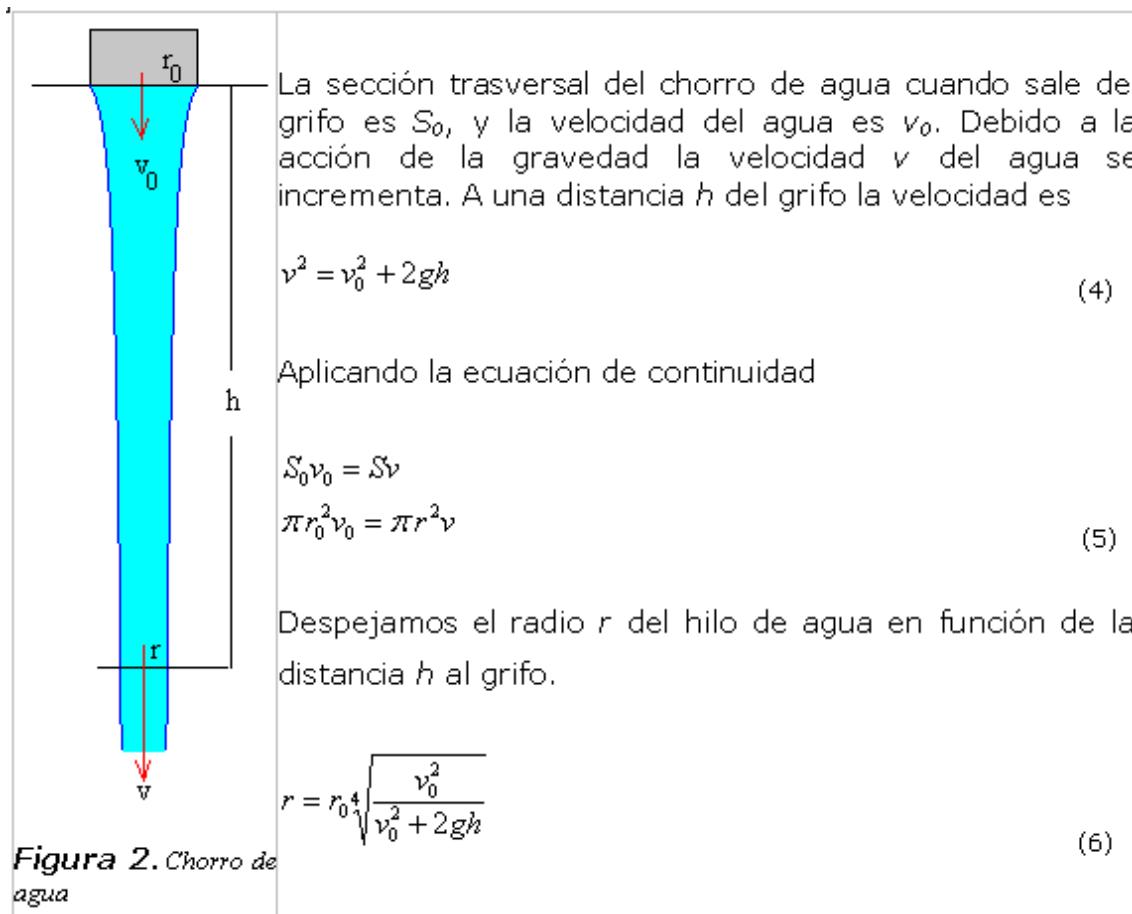


Figura 2. Chorro de agua

CAPÍTULO 3: ELECCIÓN DE SOFTWARE

En el mercado existen varios programas de ordenador, los cuales son capaces de efectuar tratamiento de imágenes. Cabe destacar Visual C++, Matlab, Borland C++ Builder.

En el proyecto me he decidido por el programa Matlab, ya que posee librerías específicas para el tratamiento de imágenes.

Las ventajas de *Image processing tollbox* son:

- Facilidad de programación.
- Gran potencia de cálculo en operaciones con matrices.
- Gran facilidad para obtener resultados gráficos.
- Código abierto disponible para Windows y Linux.

Se pueden crear interfaces gráficas de usuario. Estas interfaces pueden ser entregadas en el ordenador del cliente.

Matlab permite la comunicación con otros dispositivos de diferentes maneras: tanto por el Puerto Serie, el puerto TCP/IP y por USB.

Hemos utilizado la versión 7.0 y 7.6.0 para la ejecución de este proyecto.

CAPÍTULO 4:

ELECCIÓN

ELECTROVÁLVULAS

Las electroválvulas que se utilizarán han de ser rápidas para poder definir la imagen.

Las electroválvulas más rápidas son de mando directo, es decir no tienen membrana. Estas válvulas se abren directamente al ser activadas eléctricamente. El obturador deja libre el orificio de paso. Las válvulas actuadas directamente están cerradas por efecto del muelle y de la propia presión del fluido, en una construcción estándar.

Las electroválvulas idóneas son de la marca SMC. Tienen aplicaciones con aire y agua. La referencia es VDW33-5G-3-Q. Estas válvulas se activan a 24VDC , consumen una potencia de 3W, es decir 0,124 Amperios. Es un dato importante a la hora de elección del microcontrolador. Y el orificio interior es de 3mm.

Forma de pedido de las válvulas (para placa base)

VDW [Serie] - [Tensión] - [Número de válvula] - G - [Número de bobina] - [Tamaño orificio] - Q

Serie:

1	10
2	20
3	30

Tipo de válvula:

3	N.C. para placa base
---	----------------------

Tensión:

Símbolo	Tensión	Sin conector/cinta de bobinado	Terminal Faston TM, moldeada	Sin conector/moldeada
1	100 VAC (50/60 Hz)	●	—	●
2	200 VAC (50/60 Hz)	●	—	●
3	110 VAC (50/60 Hz)	●	—	●
4	220 VAC (50/60 Hz)	●	—	●
5	24 VDC	●	●	●
6	12 VDC	●	●	●
V	6 VDC	●	●	●
S	5 VDC	●	●	●
R	3 VDC	●	●	●

Nota) En caso de tensiones diferentes, consulte con SMC.

Tipo de material y aislamiento:

Símbolo	Material del cuerpo	Sellante	Aislamiento bobina
-	Latón (C37)	NBR	Clase B
A		FKM	
B		EPDM	
G		NBR	
H	Acero inoxidable	FKM	
J		EPDM	
L (Nota)		FKM	

Nota) Para agua desionizada: el conjunto de la armadura es resistente a la corrosión.

Modelo de bobina (Nota):

G	Sin conector/cinta de bobinado
F	Terminal Faston TM/moldeada
W	Sin conector/moldeada

Nota) Consulte la página 1 para ver las combinaciones de series y modelos de bobina.

Tamaño orificio:

Símbolo	Diámetro orificio (mm)	Serie
1	1	10
2	1.6	
1	1.6	
2	2.3	20
3	3.2	
2	2	
3	3	30
4	4	

Figura 3. Tabla pedido de válvulas.

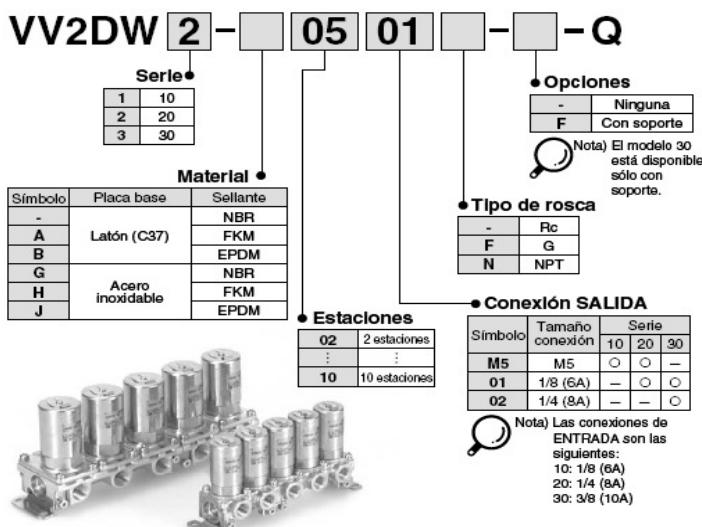
Estas válvulas son de alta frecuencia. El tiempo de respuesta es inferior a 15ms, lo que significa 33 ciclos segundo.

Fluid		Air		Water		Oil	Pressure
On/ Off		ON	OFF	ON	OFF	ON	OFF
VXZ2230		110	50	165	75	275	125
VXZ2240		120	100	180	150	300	250
VXZ2350		135	140	200	210	335	350
VXZ2360		160	170	240	255	400	425
VX2	AC	15	120	40	150	80	400
	DC	25	180	50	200	100	450
VX3	AC	25	40	50	100	70	150
	DC	35	50	60	100	80	150
VXD2130		50	120	60	150	100	400
VXD2140		70	160	500	600	600	1500
VXD2150		70	150	500	600	600	1500
VXD2260		70	160	500	600	600	1500
VXD2270		300	500	700	2000	1000	4000
VXD2380		200	500	500	2000	700	4000
VXD2390		300	500	700	2000	1000	4000
VXP		Same as VXD data					
VXH		Same as VXD2130 data					
VXA		11 be changed by air operate valve. It is unable to show indiscrimi					
VDW	[less than]	15	15	15	15		
		* VDW above mentioned data will be changed by orifice size.					

Figura 4. Características de las electroválvulas.

Las válvulas estarán instaladas en bloques de 5, con referencia VV2DW3-05-02-F-Q . El bloque tiene entradas de agua laterales de 3/8" ,y estarán conectadas a un colector que suministrará el agua. Cada válvula tiene salida de 1/4", en el cual se conectarán un racor de 1/4" a tubo de 4mm.

Forma de pedido de las placas base



Forma de pedido de conjuntos de válvulas en placa base

Añada las referencias de la válvula de montaje y de la opción debajo de la ref. de la placa base.

<Ejemplo>
VV2DW2-0501 1 Juego Ref. placa base
*VDW23-5G-2 5 Juegos Ref. válvula (estaciones 1 a 5)

"*" es el símbolo correspondiente al conjunto. Añada un "*" delante de las referencias de las electroválvulas, etc. que se deben montar.

Escriba las refs. en orden, empezando por la 1^a estación del lado D.



Figura 5. Pedido de placa para válvulas.

CAPÍTULO 5: ELECCIÓN HARDWARE

5.1. Posibles Hardware

Para la activación de las electroválvulas se podría utilizar o un microcontrolador o un autómata programable.

En el mercado existen microcontroladores capaces de activar varias salidas, y que permiten comunicación por Puerto Serie , USB, Ethernet como podría ser el caso de los PICS:

PIC24FJ256GB110 con puerto USB y 84 I/O pins.

5.2. Elección de componentes

Finalmente me he decantado por utilizar un autómata programable. En el mercado existen varios fabricantes como:

- Siemens
- Omron
- Allen Bradley

Cualquiera de los de los antes mencionados disponen de autómatas validos para realizar el proyecto.

Las ventajas de utilizar un autómata son:

- Facilidad de programación.
- Facilidad de ampliación de salidas y entradas.
- Gran fiabilidad

Así he decidido realizar el prototipo con una CPU de Siemens de la familia 300. con referencia 313-5BE00-0AB0, una CPU compacta con MPI, 24 entradas digitales, 16 salidas digitales, 4 entradas analógicas, 3 contadores de alta velocidad y 32 Kbyte de memoria de trabajo.



Figura 6. Imagen CPU 313C.

Para poder activar todas las electroválvulas será necesario ampliar las salidas digitales. La CPU tiene incorporadas 16, así nos hará falta un módulo de 32 salidas. El módulo de salidas digitales utilizado es de la familia SM 322.



Figura 7. Imagen módulo de salidas digitales.

El módulo utilizado es 6ES7 322 1BL00-0AA0.

Hay que tener en cuenta datos característicos relevantes para la elección del módulo.

	6ES7 322-1BH01-0AA0	6ES7 322-1BH10-0AA0	6ES7 322-1BL00-0AA0	6ES7 322-1BP00-0AA0	6ES7 322-1BP50-0AA0	6ES7 322-8BF00-0AB0
Tensiones de alimentación						
Tensión de carga L+	24 V					
• Valor nominal (DC)						
Módulos de S digitales						
Número de salidas digitales	16	16	32	64	64	8
Protección contra cortocircuitos en salida	sí; electrónico					
Intensidad de salida						
• para señal '1' valor nominal	0,5 A	0,5 A	0,5 A	0,3 A	0,3 A	0,5 A

Datos técnicos (continuación)

	6ES7 322-1BH01-0AA0	6ES7 322-1BH10-0AA0	6ES7 322-1BL00-0AA0	6ES7 322-1BP00-0AA0	6ES7 322-1BP50-0AA0	6ES7 322-8BF00-0AB0
Frecuencia de conmutación						
• con carga resistiva, máx.	100 Hz	1 000 Hz	100 Hz	100 Hz	100 Hz	100 Hz
• con carga inductiva, máx.	0,5 Hz	0,5 Hz		0,5 Hz	0,5 Hz	2 Hz
• con carga tipo lámpara, máx.	10 Hz					

Figura 8. Datos técnicos salida digital 322.

Como podemos observar este módulo contiene 32 salidas digitales, estas salidas digitales pueden suministrar 0,5 A a una tensión de 24 VDC, con lo cual son válidas para las electroválvulas anteriores.

También observamos que no tienen límite de frecuencia de conmutación para cargas inductivas.

Será necesario un módulo de comunicaciones para enlazar comunicación con el ordenador. Utilizaremos una comunicación TCP, y para ello utilizaré una CP 343-1 con referencia 343-1EX20-0XE0. Y Para la comunicación utilizaremos un cable cruzado UTP categoría 6.

**Figura 9.** Imagen módulo CP.

Table 9-1 Technical Specifications

	CP 343-1	CP 343-1EX20
Transmission rate	10 Mbps and 100 Mbps	
Interfaces		
Attachment to Industrial Ethernet (10/100 Mbps)	16-pin sub-D female connector (automatic switchover between AUI and Industrial Twisted Pair)	not applicable
Attachment to Twisted Pair	RJ-45 jack	
Power supply	+5 V DC (+/-5%) and +24 V DC (+/-5%)	
Current consumption		
• from backplane bus	70 mA	200 mA
• from external 24 V DC	AUI: approx. 0.73 A maximum TP/ITP: approx. 0.4 A maximum	TP/ITP: approx. 0.2 A maximal
Power loss	10 W	5,8 W
Permitted ambient conditions		
• Operating temperature	0 °C to +60 °C	
• Transportation/storage temperature	-40 °C to +70 °C	
• Relative humidity max.	95% at +25 °C	
• Altitude	up to 2000 m above sea level	
Design		
• Module format	Compact module S7-300; double width	
• Dimensions (W x H x D) in mm	80 x 125 x 120	
• Weight approx.	600 g	

Figura 10. Datos característicos CP.

Para alimentar la CPU , módulos y electroválvulas será necesario una fuente de alimentación de 24 V continua, Utilizaremos una fuente de la marca Siemens con referencia 307-1KA01-0AA0. Esta fuente dispone de regulador de tensión de salida y protección electrónica contra cortocircuitos.



Figura 11. Imagen fuente de alimentación.

CAPÍTULO 6:

TRATAMIENTO

IMÁGENES

6.1. Introducción

Para la realización del proyecto utilizaremos el programa Matlab ya que este dispone de funciones específicas para el tratamiento de imágenes.

La idea inicial es escoger una imagen en colores, pasarl a tonos grises y luego poder seleccionar el rango de blancos y negros de la imagen en tonos grises para poder binarizarla, obteniendo al final 1 y 0. Así podremos representarlo un 1 con una gota y 0 sin gota.

Una vez conseguida la imagen en binario habrá que establecer comunicación con un periférico, que en nuestro caso será un Autómata programable, el cual activará la válvula correspondiente.

El tamaño de la imagen vendrá dado por la cantidad de válvulas que dispongamos. El proyecto está pensado para representar pequeños logos. Se podría representar imágenes más grandes pero hemos de contar que cada válvula equivaldrá a un píxel.

Para la comunicación con el autómata utilizaremos el puerto TCP-IP. Donde enviaremos las tramas de 0 y 1.

Una vez recibidas estas tramas el autómata ejecutará su programa activando las válvulas necesarias.

6.2. Conceptos básicos

Píxel es el elemento básico de una imagen.

Imagen es el arreglo bidimensional de píxeles con diferente intensidad luminosa.

Si la intensidad luminosa de cada píxel se representa por n bits, entonces existirán 2^n escalas de gris diferentes. Matemáticamente, una imagen se representa por $r = f(x, y)$, donde r es la intensidad luminosa del píxel cuyas coordenadas son (x, y) . Matemáticamente, un sistema para procesar imágenes se representa como $g(x, y) = T[f(x, y)]$.

El color se forma mediante la combinación de los tres colores básicos rojo, azul y verde. La forma más sencilla de obtener un color específico es determinar la cantidad de color rojo, verde y azul que se requiere combinar para obtener el color deseado, para lo cual se realiza la suma aritmética de las componentes: $X = R + G + B$, gráficamente representada por un cubo.

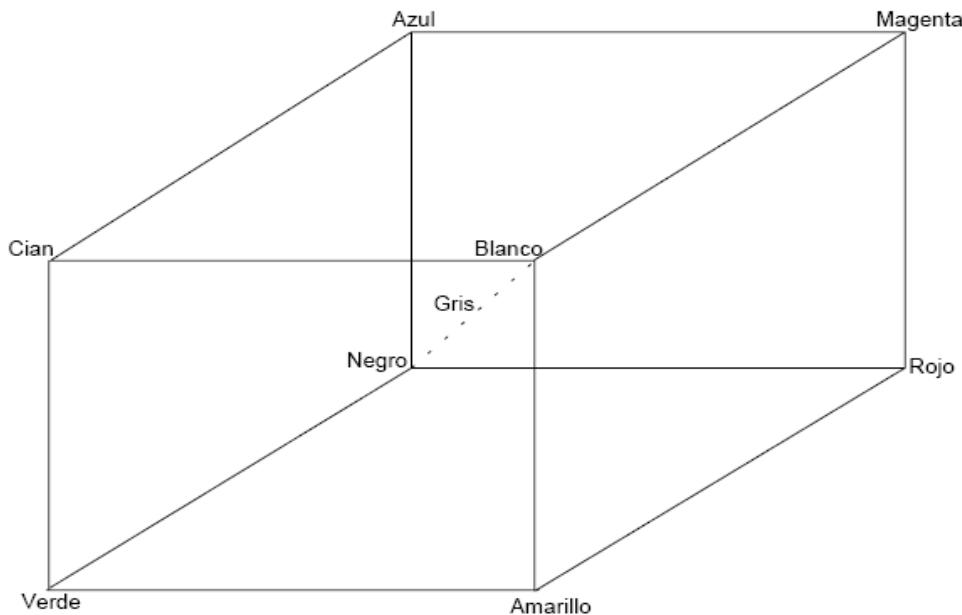


Figura 12. Relación colores RGB

En la recta que une el blanco con el negro se encuentran ubicados los grises (escala de gris) debido a que sus tres componentes son iguales.

6.3. Imágenes en Matlab

En Matlab una imagen a escala de grises es representada por medio de una matriz bidimensional de $m \times n$ elementos, donde n representa el número de píxeles de ancho y m el numero de píxeles de largo, donde cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco).

Por otro lado una imagen de color RGB es representada por una matriz tridimensional $m \times n \times p$, donde m y n tienen la misma significación que en el caso de las imágenes de escala de grises mientras p representa el plano, que para RGB puede ser 1 para el rojo, 2 para el verde y 3 para el azul. En la siguiente imagen se puede observar las matrices resultantes de una imagen en color en Matlab.

$$I_R(m,n,1) = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \quad I_G(m,n,2) = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn} \end{bmatrix} \quad I_B(m,n,3) = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

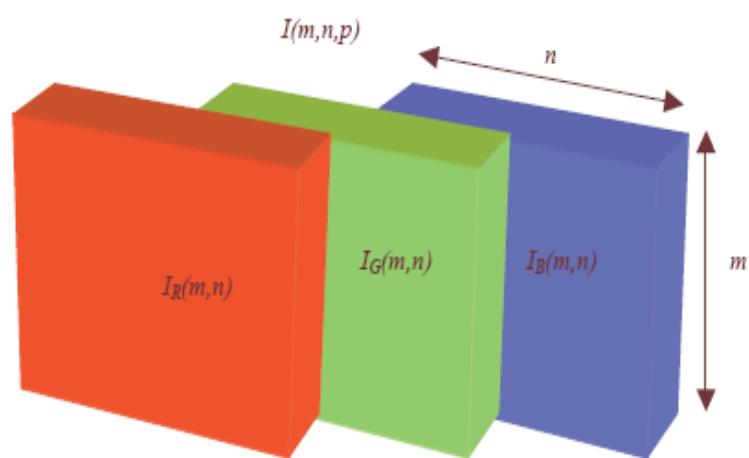


Figura 13. Matriz de una imagen en Matlab.

En la siguiente tabla podemos observar un ejemplo de la relación de RGB de los principales colores.

Tabla 1. Relación RGB de principales colores.

Color	R	G	B
Negro	0	0	0
Blanco	255	255	255
Amarillo	255	255	0
Verde	0	255	0
Azul	0	0	255
Gris	127	127	127

CAPÍTULO 7:

PROGRAMACIÓN

MATLAB

7.1. Programación GUI

Para la utilización del programa utilizaremos interface gráfica de usuario, conocida en Matlab como GUI.

Las GUI-s creadas con Matlab pueden ser entregadas al ordenador del cliente. Las GUI-s son herramientas muy útiles para entregar aplicaciones a aquellas personas que no saben lo suficiente de programación y que quieren beneficiarse de las ventajas de un programa.

El diseño inicial de GUI estará compuesto por tres botones, un botón de control deslizante y tres imágenes, como muestra la figura 14.

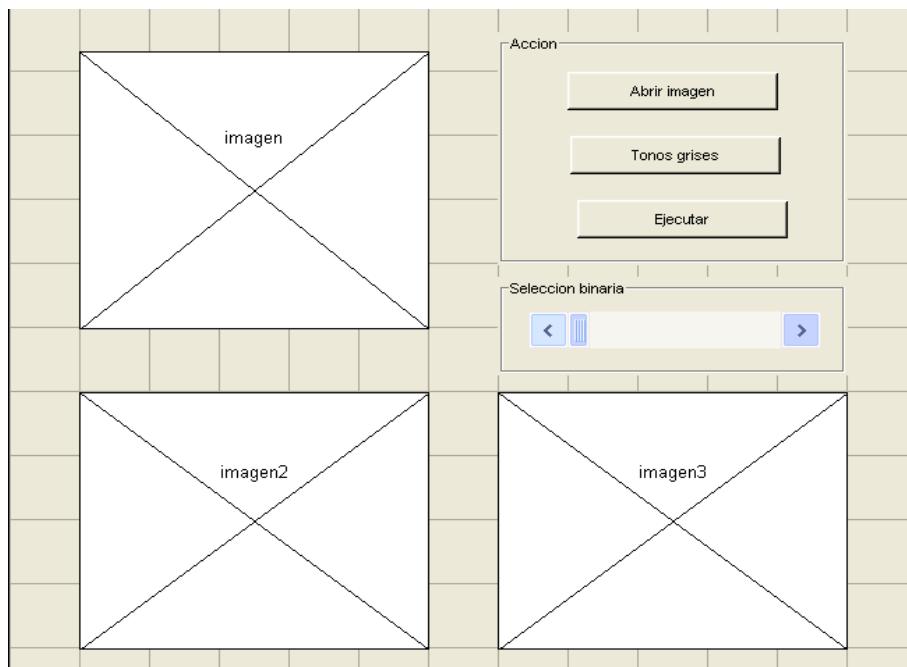


Figura 14. Configuración GUI.

Cuando ejecutamos el programa, es decir cuando lo ponemos en RUN, podemos apreciar la figura 15:

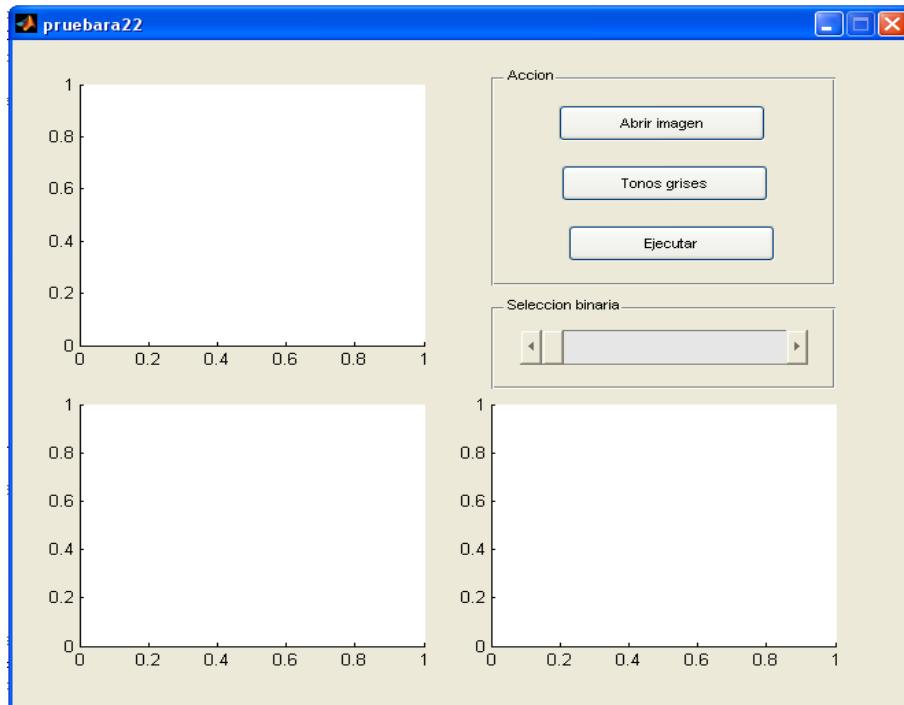


Figura 15. GUI en ejecución.

Cuando presionamos el botón de *abrir imagen* se abre una ventana en la cual podemos escoger una imagen que está en el directorio work de Matlab.

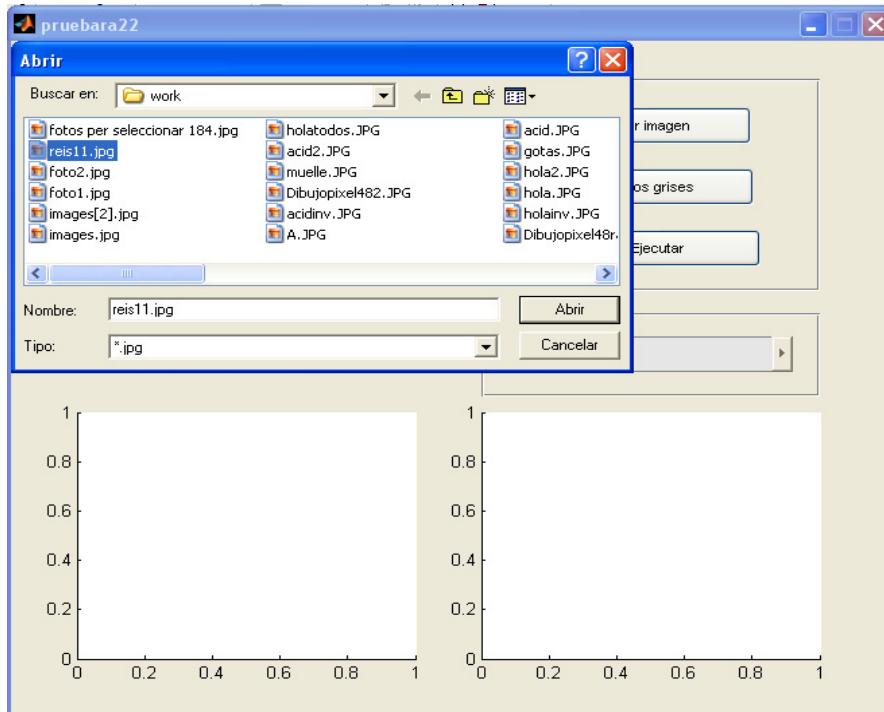


Figura 16. Imagen botón *abrir imagen*.

En la ventana podemos observar todos los archivos con extensión JPG. Esto se debe a que por defecto se ha programado así, pero podríamos abrirlo en otros formatos como TIF, GIF, etc.

Seleccionando una, por ejemplo reis11.JPG, y presionando el botón de *abrir* de la subventana se abrirá el dibujo en la *imagen* de la GUI:

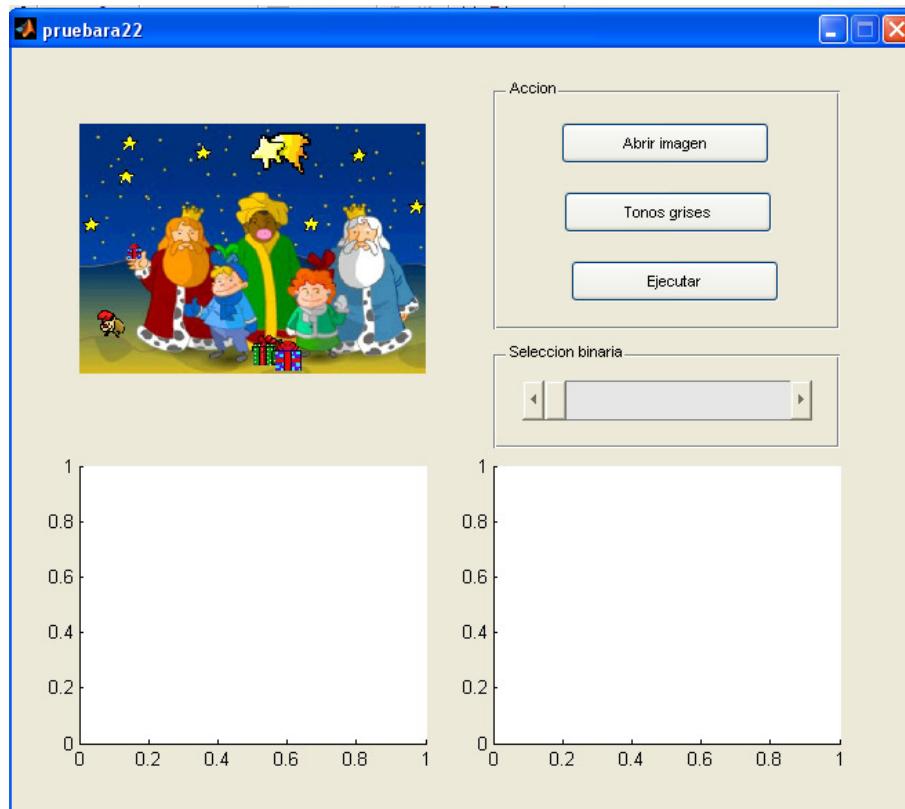


Figura 17. Imagen resultante de abrir una imagen reis11.JPG.

Seguidamente si presionamos el botón *Tonos grises* aparecerá la imagen anterior en la *imagen2* de la GUI

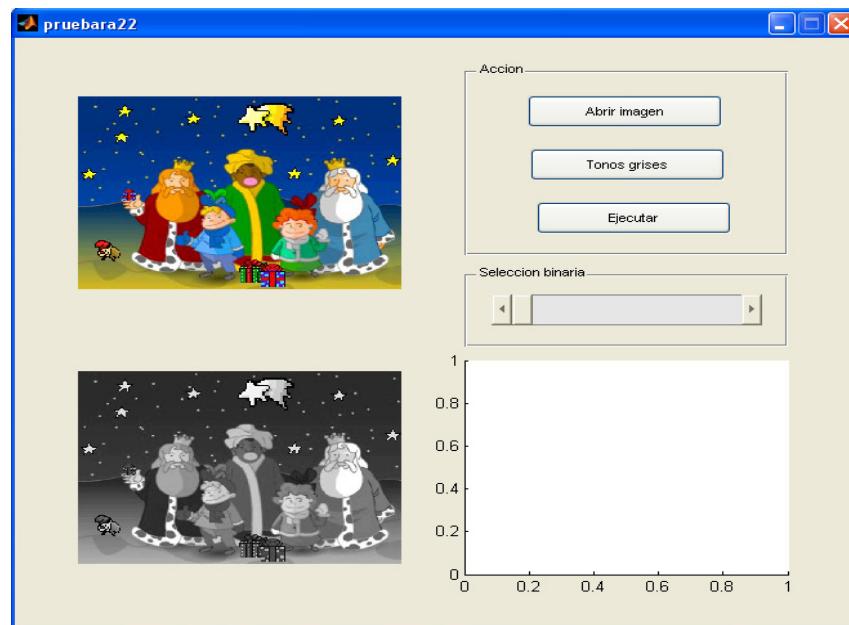


Figura 18. Imagen en tonos grises.

Y al mover el botón de control de desplazamiento aparecerá la imagen en la imagen3 de GUI en binario. Tal como desplacemos el botón seleccionaremos el rango en que queremos que los grises sean blancos o negros.



Figura 19. Imagen binaria blancos y negros.

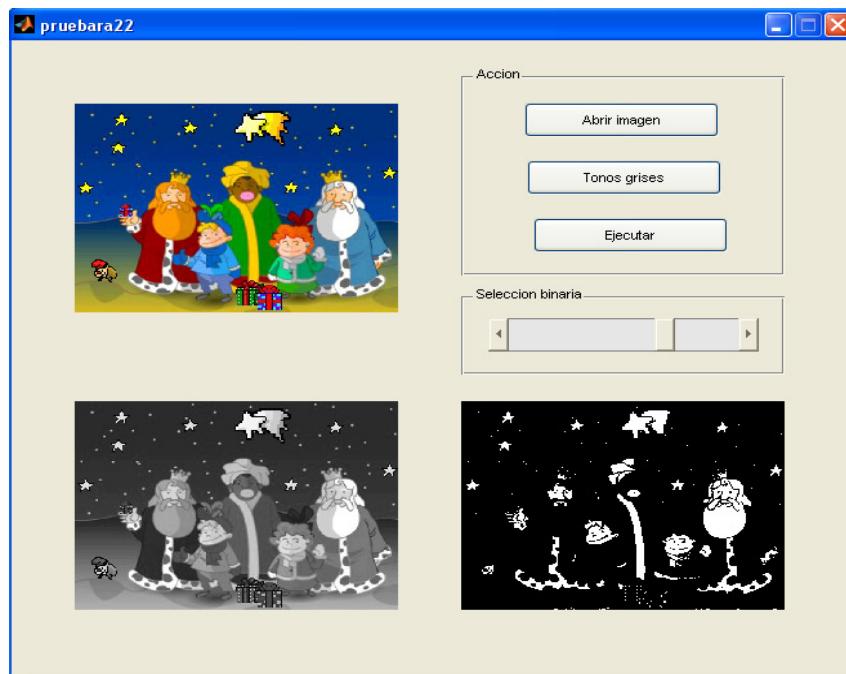


Figura 20. Imagen 2 binaria blancos y negros.

7.2. Código

Para probar el código empezaremos con una imagen de 5 X 5 píxeles creada en Paint, y guardada como *pruebamatrix1.JPG*.

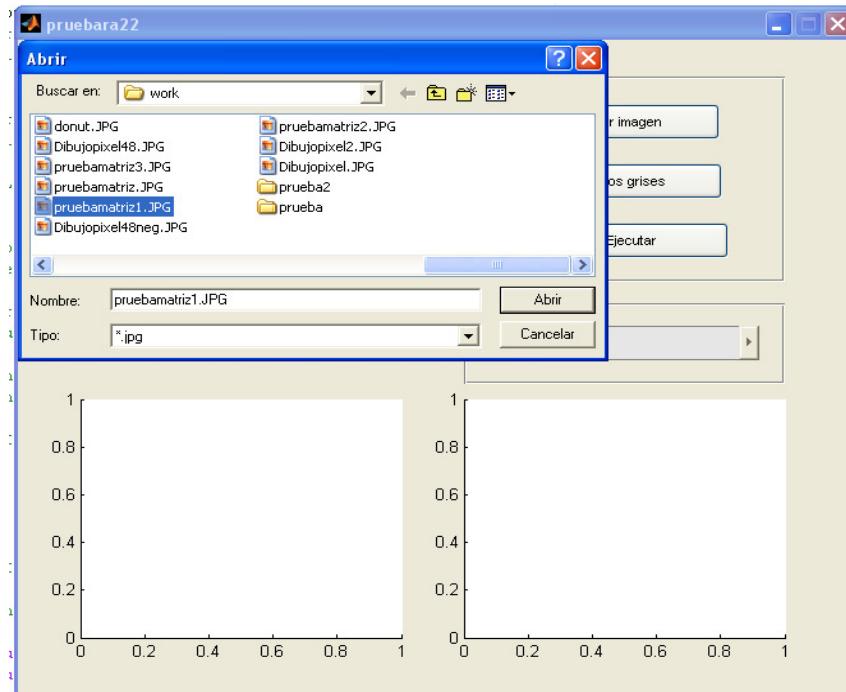


Figura 21. Imagen de abrir *pruebamatrix1.JPG*.

7.2.1. Código del botón *Abrir imagen*:

```
79 function pushbutton1_Callback(hObject, eventdata, handles)
80 % hObject    handle to pushbutton1 (see GCBO)
81 % eventdata  reserved - to be defined in a future version of MATLAB
82 % handles    structure with handles and user data (see GUIDATA)
83 - [nombre direc]=uigetfile('*.jpg','Abrir');
84 - if nombre ==0
85 -     return
86 - end
87 - global img
88 -
89 - img=imread(fullfile(direc,nombre));
90 - axes(handles.imagen)
91 - image(img)
92 - imshow(img)
93 - axis off
```

Podemos observar como especificamos que los archivos sean JPG, mediante la función *uigetfile*, y la subventana permanece abierta hasta que seleccionemos un archivo y lo abramos o cancellemos, y seguidamente almacena la imagen como una matriz con nombre *img* y la representa por pantalla con la función *imshow*.

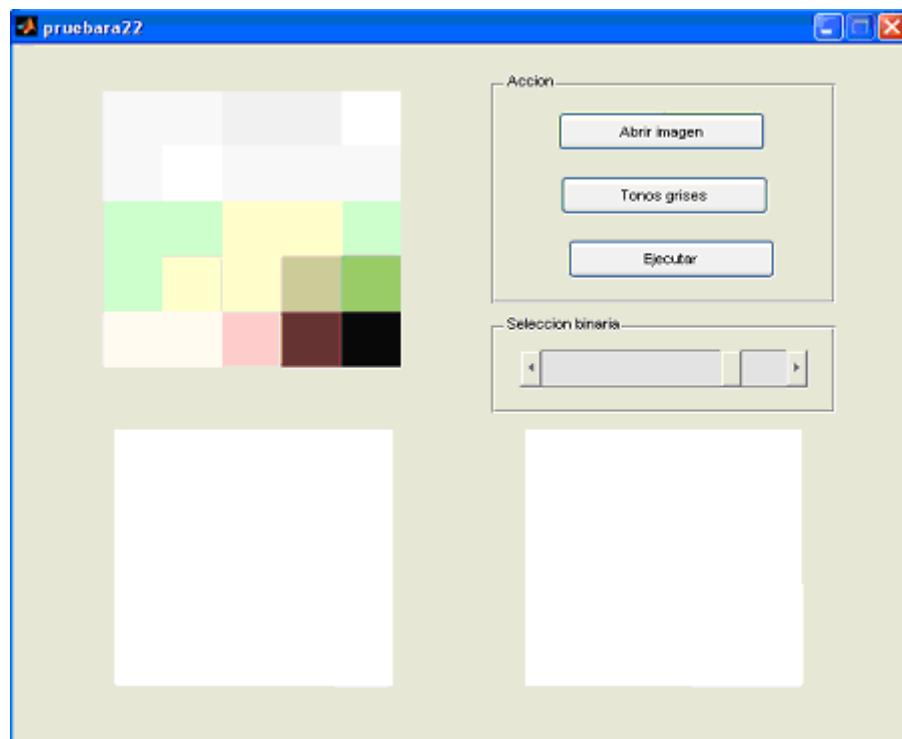


Figura 22. Imagen pruebamatriz1.JPG.

Podemos observar en el Comand Window de Matlab como se almacena la variable *img*. Podemos observar como la variable se almacena mediante tres matrices, que corresponden a los tres planos RGB.

```
K>> img

img(:,:,1) =

 250  255  255  255  255
 242  255  255  255  244
 207  221  255  254  218
 220  234  255  208  134
 253  255  255  127   14

img(:,:,2) =

 245  240  228  230  249
 255  254  241  239  255
 255  255  255  255  255
 255  255  255  214  191
 255  252  220    74   14

img(:,:,3) =

 255  255  255  255  255
 255  255  255  250  247
 207  205  212  209  205
 209  207  205  154  112
 237  237  219    70    0
```

7.2.2. Código del botón *Tonos grises*:

```
98 function pushbutton2_Callback(hObject, eventdata, handles)
99 % hObject    handle to pushbutton2 (see GCBO)
100 % eventdata  reserved - to be defined in a future version of MATLAB
101 % handles    structure with handles and user data (see GUIDATA)
102 %
103 % global img
104 % global grises
105 %
106 % grises=rgb2gray(img);
107 % axes(handles.imagen2)
108 % image(grises)
109 % imshow(grises)
110 % axis off
111
```

Para poder trabajar con la variable *img* esta tiene que estar definida como global, y mediante la función *rgb2gray* transforma la imagen a tonos grises.

```
K>> grises

grises =

248 246 239 240 251
251 254 247 245 251
235 239 250 249 238
239 243 249 205 165
252 251 230 89 12
```

7.2.3. Código del botón de desplazamiento (*Slider*):

```
201 function nivel_Callback(hObject, eventdata, handles)
202 % hObject    handle to nivel (see GCBO)
203 % eventdata   reserved - to be defined in a future version of MATLAB
204 % handles    structure with handles and user data (see GUIDATA)
205
206 % Hints: get(hObject,'Value') returns position of slider
207 %         get(hObject,'Min') and get(hObject,'Max') to determine range of slider
208 % global grises
209 % global bw
210 % global matriz1
211 % global v1
212 v=get(handles.nivel,'value');
213 bw=im2bw(grises,v);
214 axes(handles.imagen3)
215 image(bw)
216 imshow(bw)
217 matriz1=bw;
%--
```

Mediante *im2bw* binarizamos la imagen según el rango v que adquirimos del *slider*, para que funcione esta función el valor máximo del slider es 1 y el menor es 0

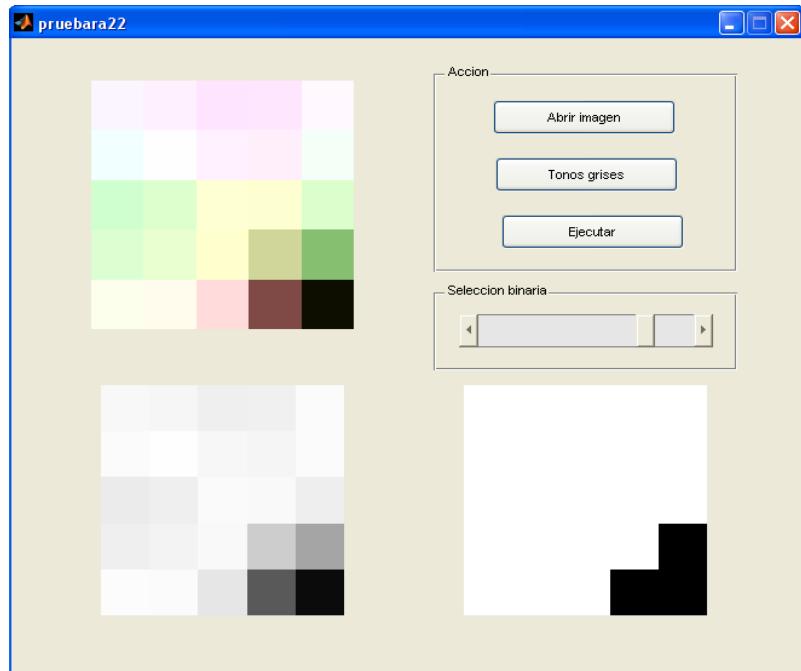


Figura 23. Imagen tonos grises de pruebamatrix1.JPG.

```
K>> matriz1

matriz1 =

1     1     1     1     1
1     1     1     1     1
1     1     1     1     1
1     1     1     1     0
1     1     1     0     0
```

Como podemos observar la matriz creada donde teníamos un píxel blanco equivale a un 1 y los negros equivale a un 0. Luego modificaremos la matriz para poderla enviar por Ethernet.

7.2.4. Código del botón *Ejecutar*:

Será el encargado de la modificación de la matriz y el envío de las tramas por Ethernet.

Este envío será por tramas, teniendo una imagen de 48x48 píxeles se efectuará 48 envíos de un vector ASCII con 48 caracteres.

Empezando por el vector inferior de la matriz, para que cuando luego activemos las electroválvulas la imagen corresponda con la observada en el ordenador. Así probaremos el programa con una imagen de 48 x 48 píxeles.

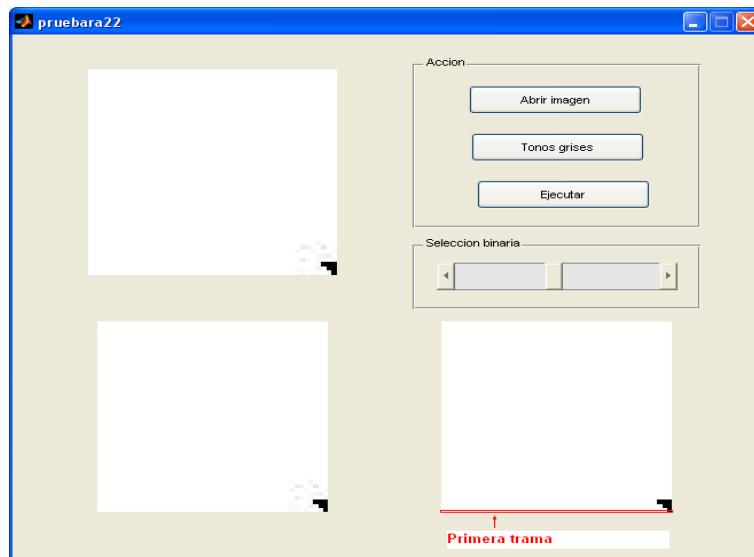


Figura 24. Prueba ejecutar imagen 48 x 48 píxeles.

Para poder enviar datos por TCP/IP habrá que crear un objeto IP en Matlab. Este objeto será el autómata el cual tendrá la siguiente dirección IP : 10.15.1.203 y especificaremos que comunicaremos con él por el puerto 10500.

```

function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global bw
global matriz1
global u
global grises
mm = int2str(matriz1)
u=tcpip('10.15.1.203',10500)
fopen(u);

for h = 48: -1: 1
    v2=mm(h,:)
    for c = 1: +1: 48
        e=c+2*(c-1);
        v3(c)=v2(e)

    end;
    for p = 1000000: -1: 0
    end;
    fwrite(u,v3,'int8')
    for a = 100.0: -0.1: 0.0
    end;
end;
fclose(u);
delete(u);

```

Transformaremos la *matriz1* que tiene formato lógico para poder enviar la trama, ya que mediante la instrucción *fwrite* no permite el envío de vectores con dicho formato. Para ello utilizo la instrucción *int2str* que genera una matriz de caracteres, pero el 0 lógico lo transforma en " 0" y el 1 lógico en " 1", es decir genera una matriz de caracteres con dos espacios en blanco entre caracteres. Para solucionar lo antes descrito utilizo un bucle, el cual descompone la matriz en vectores que equivalen a filas de la matriz y luego redimensiona el vector de 1 x 142 vector de caracteres a un vector 1 x 48 vector de caracteres.

A continuación podemos observar la transformación de $v2$ a $v3$ de la primera trama de la imagen anterior. Una vez la transformación esta realizada se envía mediante `fwrite` sobre el objeto IP .

Y así, hasta las 48 filas de la imagen. Podemos observar la segunda trama a enviar.

En la figura 25 podemos observar las especificaciones de las variables utilizadas.

```
K>> whos
  Name      Size            Bytes  Class
  a          1x1                  8  double array
  bw         48x48           2304  logical array (global)
  c          1x1                  8  double array
  e          1x1                  8  double array
  eventdata  0x0
  grises     48x48           2304  uint8 array (global)
  h          1x1                  8  double array
  hObject    1x1                  8  double array
  handles     1x1             1452  struct array
  matriz1    48x48           2304  logical array (global)
  man        48x142           13632 char array
  p          1x1                  8  double array
  u          1x1                 888  tcpip object (global)
  v2         1x142              284  char array
  v3         1x48                  96  char array

Grand total is 13964 elements using 23312 bytes
```

Figura 25. Variables utilizadas en Matlab.

7.2.5. Organigrama del programa

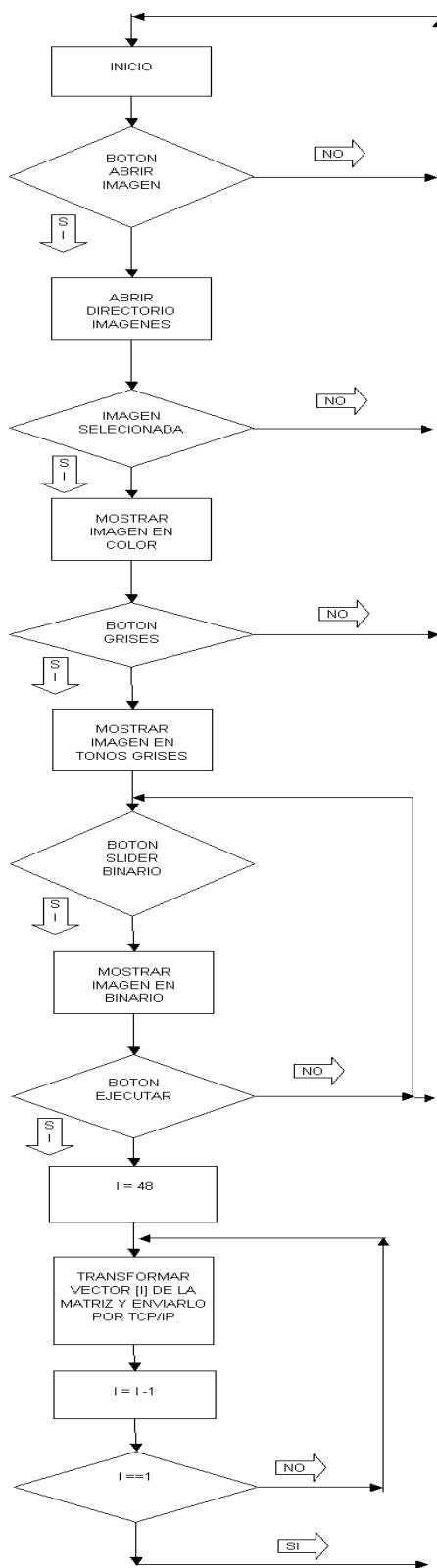


Figura 26. Organigrama programa Matlab

7.2.6. Ejecutable del programa

Matlab permite crear ejecutables , los cuales pueden ser usados en otros ordenadores sin necesidad de tener instalado Matlab.

Para crear dicho ejecutable utilizo el comando *Deploytool*. Para utilizar este comando se necesita como mínimo la versión 7.6.0 de Matlab.

Al efectuar ese comando, se abre una ventana donde indica los pasos que se tienen que efectuar, como indica la imagen siguiente:

Welcome to the Deployment Tool

To get started, do any of the following:

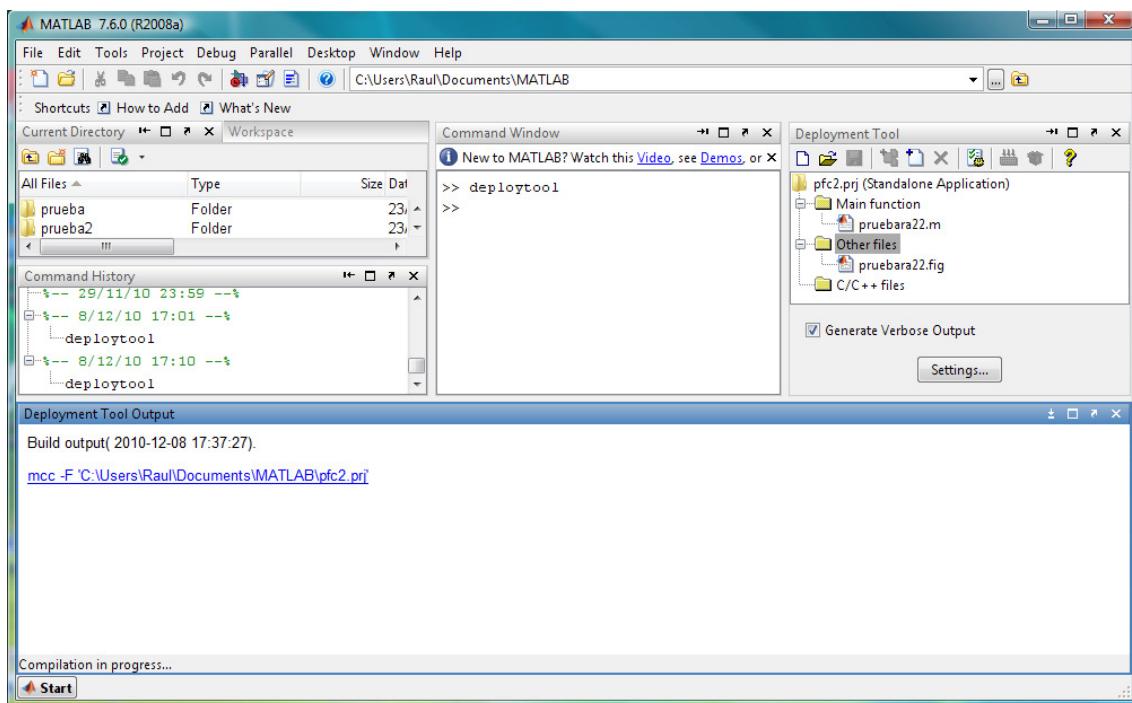
- Click the New Deployment Project icon  in the Deployment Tool toolbar.
- Click the Open Deployment Project icon  to open an existing project.
- Click the Help icon  in the toolbar.

Use the Deployment Tool to perform these tasks:

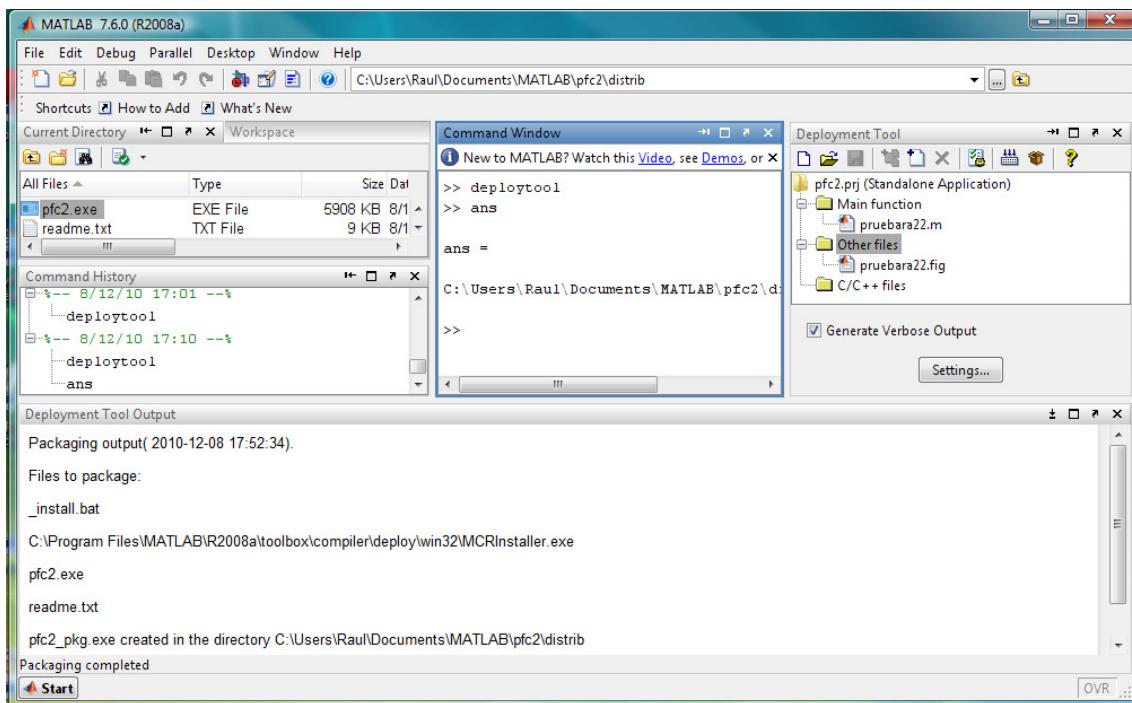


Figura 27. Guía ejecutable Matlab

Primeramente crearemos un proyecto *pfc2.prj*, luego adjuntaremos el programa principal *pruebara22.m*, y adjuntaremos otros ficheros necesarios, como en nuestro caso es *pruebara22.fig* y compilamos.

**Figura 28.** Compilación del programa.

Una vez compilado , efectuamos el empaquetado del proyecto pfc2_pkg. Con este archivo se puede ejecutar el programa en cualquier ordenador sin tener instalado Matlab.

**Figura 29.** Empaquetado del ejecutable.

CAPÍTULO 8: PROGRAMACIÓN SIEMENS

8.1. Introducción

Para la activación de las válvulas y la comunicación con la aplicación utilizaremos un autómata programable Siemens.

Esta CPU se alimenta a 24Vdc con lo cual necesitaremos una fuente de alimentación de 220V a 24V. Esta fuente suministrará 10 amperios, es de la marca Siemens con referencia 307-1KA01-0AA0.

Para poder introducir el programa utilizaremos una micro memory de 128KB con referencia 6ES7 953-8LG11-0AA0.

Aunque la CPU ya contiene salidas integradas tendremos que poner un módulo de 32 salidas digitales.

Y por último será necesario un módulo de comunicaciones para enlazar comunicación con el ordenador. Utilizaremos una comunicación TCP, y para ello utilizaré una CP 343-1. Y Para la comunicación utilizaremos un cable UTP categoría 6 cruzado.

Para la programación utilizaremos el programa Simatic Step 7, y programaremos en lenguaje KOP y AWL. Para comunicar con la CPU por el puerto MPI utilizaremos la tarjeta PCMCIA CP5512, y el PC ADAPTER USB con referencia 6ES7972-0CB20-0XA0.

8.2. Introducción a la programación

Un autómata se compone de:

- Hardware (lo tangible, circuitería..)
- Software (lo intangible, el programa, sistema operativo..)

Para la programación del autómata podemos utilizar varios lenguajes:

- Literal: AWL
- Gráfico: FUP(diagrama de funciones), KOP(esquema de contactos)

En un autómata un programa se ejecuta en tres modos diferentes:

- Ejecución cíclica.
- Ejecución por alarmas.
- Ejecución por tiempo.

En el proyecto actual utilizamos una ejecución cíclica:

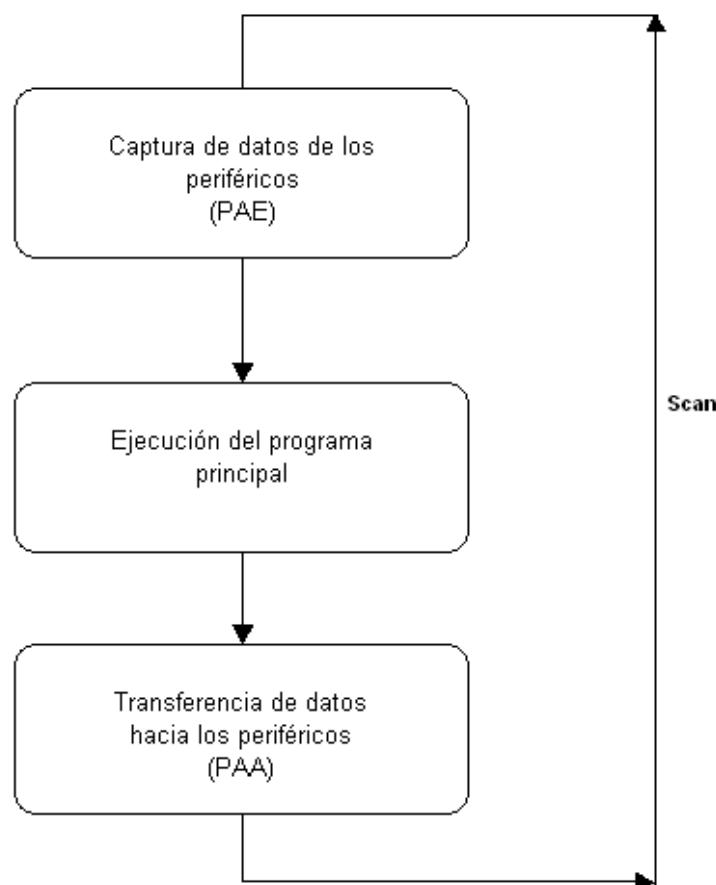


Figura 30. Tiempo de Scan

Cada vez que comienza el proceso, por la parte superior se dice que comienza un "Scan". El tiempo que destina el autómata en realizar un "Scan" se controla mediante un temporizador llamado "Watch Dog Timer". Si el tiempo excede de una cantidad prefijada, el autómata avisa con una alarma. Por defecto es de 150 ms.

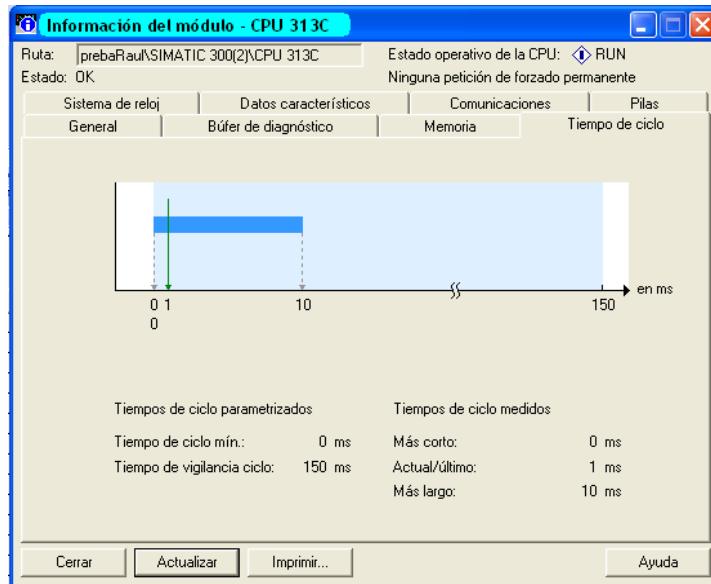


Figura 31. Tiempo de Scan del programa.

Este tiempo depende de la longitud del programa, ya que cada instrucción tarda un tiempo en ejecutarse. Este tiempo es muy importante, ya que para procesos rápidos podríamos pedir información, si no se tiene en cuenta.

En nuestro caso como podemos observar en la figura 31, cuando comutan las electroválvulas tenemos un ciclo de Scan de 1 ms. Esto significa que si queremos comutar las electroválvulas cada 20 ms, podemos afirmar que comutarán entre 21-22 ms.

Al comienzo de cada ciclo de Scan, el programa destina un tiempo en leer la información analógica y digital que pueda haber en los periféricos, y después deposita esa información en una memoria llamada PAE, organizada en bytes.

Seguidamente se ejecuta el programa principal estructuradamente, pero no pone inmediatamente el resultado sobre la periferia de salida. Estos resultados se almacenan en la PAA. Hasta que el programa no llega a su fin no se llevan a la periferia.

Todas las entradas y salidas tienen una dirección absoluta. Esta dirección se adopta directamente en el software. Las direcciones absolutas se pueden sustituir por nombres simbólicos definidos por el usuario utilizando la tabla de símbolos. Estos son válidos para todos los componentes del programa.

La programación simbólica permite mejorar la comprensión del programa.

Las marcas de memoria son bits internos y se utilizan para almacenar datos intermedios que queramos preservar. Pueden manejarse como bit, byte, word, dword, etc.

En sep 7 se utiliza una programación estructurada, mediante módulos. Los módulos son las partes funcionales del programa de usuario, representan el código ejecutable del programa. Se diferencian en su función, uso y estructura.

Módulos lógicos:

- OBs – Módulos de organización.
- FCs – Funciones.
- FBs - Módulos de función.
- SFBs - Módulos de Función de Sistema.
- SFCs - Funciones de Sistema.

Módulos de Datos:

- DBs - Módulos de Datos
- SDBs - Módulos de Datos de Sistema

Tipos de Datos definidos por el Usuario:

- UDTs

8.3. Módulos de Organización

Constituyen el interface entre la CPU y el programa. El sistema operativo llama a estos módulos de código para que realicen operaciones de acuerdo a su función.

8.3.1. OB1

Es un módulo de código que trabaja cíclicamente, el OB1 se ejecuta en cada ciclo de Scan. Se puede almacenar y ejecutar su programa completo en el OB1, o bien puede almacenar su programa en diferentes módulos lógicos y utilizar el OB1 para llamarlos. Esto se conoce como programación estructurada. La programación estructurada permite:

- Aligerar las operaciones del programa principal.
- Disminuir espacio de memoria ocupada.
- Simplificar la programación de proceso.

El OB1 es el modulo principal del programa. Además del OB1, el sistema operativo puede llamar a otros OBs . Los OBs sólo pueden ser llamados por el sistema operativo, no por otros módulo lógicos.

El sistema operativo de la CPU del S7 ofrece 26 clases de prioridad en la ejecución del programa. El sistema operativo procesa los OBs de acuerdo a

su prioridad. El OB1 tiene la prioridad más baja de todos los OBs. Esto permite que todos los otros OBs pueden interrumpir, si es necesario la ejecución cíclica del OB1.

Cada CPU del S7 soporta diferentes OBs y sus respectivas funciones. Los diferentes tipos de OBs realizan diversas funciones:

OB1 Programa principal Ejecución en ciclos continuos.

OB10.. OB17 Interrupciones y alarmas horarias.

OB20..OB23 Interrupción de Retardo.

OB30.OB38 Interrupción Cíclica.

OB40..OB47 Interrupción Hardware. Alarmas de proceso.

OB80..OB87 Error Asíncrono.

OB121,122 Error Síncrono Se ejecuta si hay errores síncronos.

OB100 Rearranque Completo Se ejecuta al pasar la CPU de STOP a RUN.

OB101 Rearranque Se ejecuta si hay un rearranque de la CPU.

8.3.2. Funciones

Las Funciones son como subrutina del programa. Son módulos de código que se ejecutan al ser llamadas desde otro módulo. Cuando se la llama, se ejecuta el código. Cuando termina, el control se devuelve al módulo y al punto del programa que fue llamado. Podemos observar como son llamados los módulos en la figura 32:

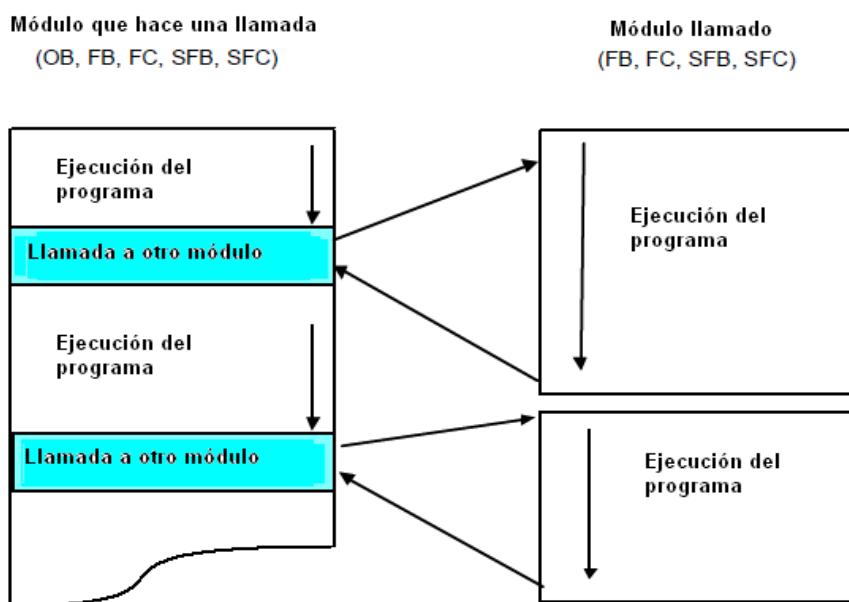


Figura 32. Ejemplo ejecución programa cíclico.

8.3.3. Módulos de función

Son muy útiles para las partes de programa que aparecen con frecuencia o con gran complejidad. Posee un área de memoria asignada para almacenar datos, conocida como bloques de datos de instancia.

8.3.4. Módulos de función de sistema

Son módulos ya programados, los cuales están preparados para realizar operaciones complejas: comunicación vía enlaces configurados, y funciones especiales integradas.

Al igual que los Módulos de función, disponen de un área de memoria asignada.

8.3.5. función de sistema

Son funciones integradas en el sistema operativo de la CPU, y pueden ser llamadas desde el programa de usuario.

8.3.6. Bloque de datos

Un bloque de datos es una parte del programa especializada en almacenar información. El tamaño de cada bloque, varia según el modelo de CPU, pero el valor medio es de 8 Kbytes. La cantidad de bloques de datos, también varía según el modelo.

Los bloques de datos se clasifican en dos categorías:

- Bloques datos globales
- Bloques datos de instancia.

Los bloques de datos globales la información que contienen esta disponible desde cualquier parte del programa. En los bloques de datos de instancia la información es accesible desde los bloques de función.

8.4. Introducción al programa

El programa se divide en cuatro partes esenciales:

La primera parte es la encargada de recibir los datos a través de TCP/IP. Cuando recibe una trama de 48 bytes, es decir 48 caracteres("1" o "0"), se encarga de guardar esos bytes en un bloque de datos en formato carácter, en particular se trata del DB 100.

La segunda parte se encarga de copiar los datos del DB 100 al DB 110, es decir con la primera trama copia los 48 bytes del DB 100 al DB 110. Cuando recibe la segunda trama, copia los 48 bytes del DB 100 a continuación de la

trama 1 en el DB 110. Cuando se hayan enviado las 48 tramas el DB 110 esta lleno con 2304 bytes,

La tercera parte se encarga de comparar uno a uno cada carácter del DB110 y si es un "1" pone a *false* el bit correspondiente del DB 120, Es decir cada byte del DB 110 equivale a un bit en el DB 120. Si fuese "0" lo pone el bit a *true*. Esto se debe a que un blanco en Matlab es un "1".

La cuarta parte se encarga de mover 48 bits del DB120 a las salidas digitales cada cierto tiempo. Hablamos de ciclos de pocos milisegundos. Cada ciclo mueve los 48 siguientes bits del DB 120 a las salidas digitales, hasta llegar a su fin.

Para una salida digital, cuando hablamos de que está a *true* quiere decir que esta activada.

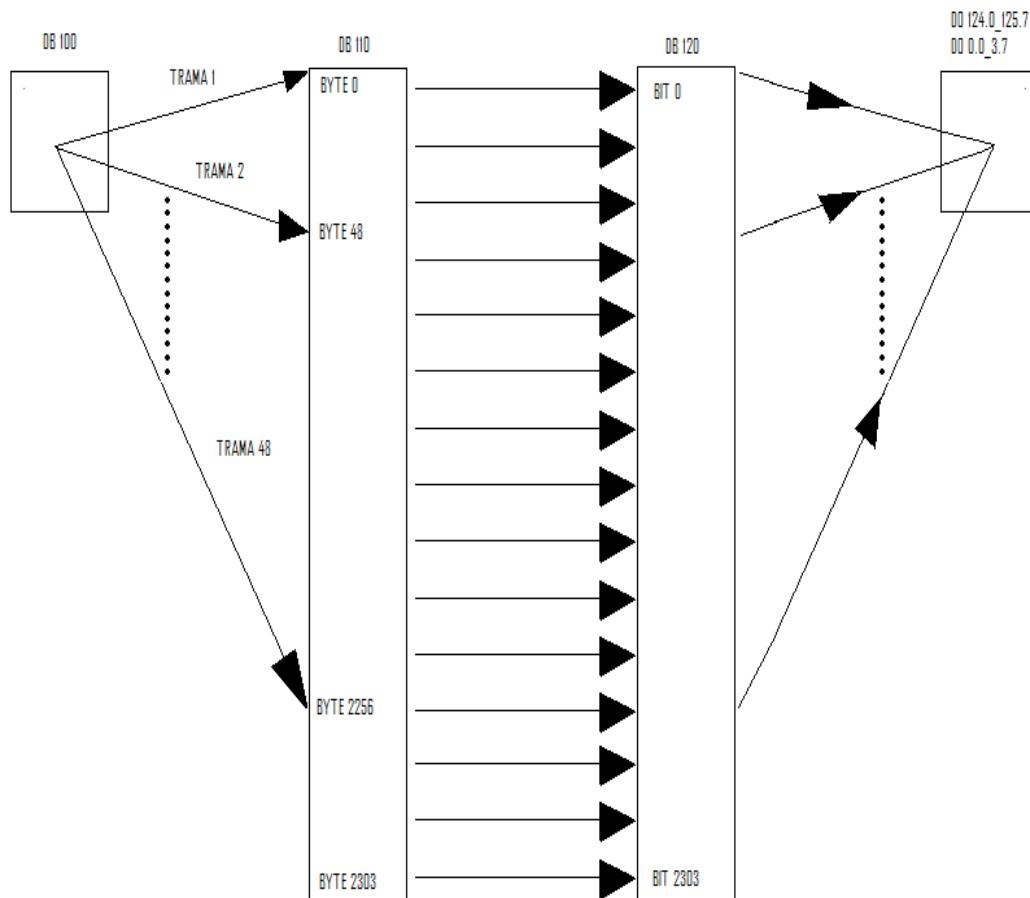


Figura 33. Esquema de programación.

8.5. Programación Hardware

Primeramente crearemos un proyecto nuevo, llamado *pruebraul* y en él especificaremos el hardware que dispondrá nuestro proyecto.

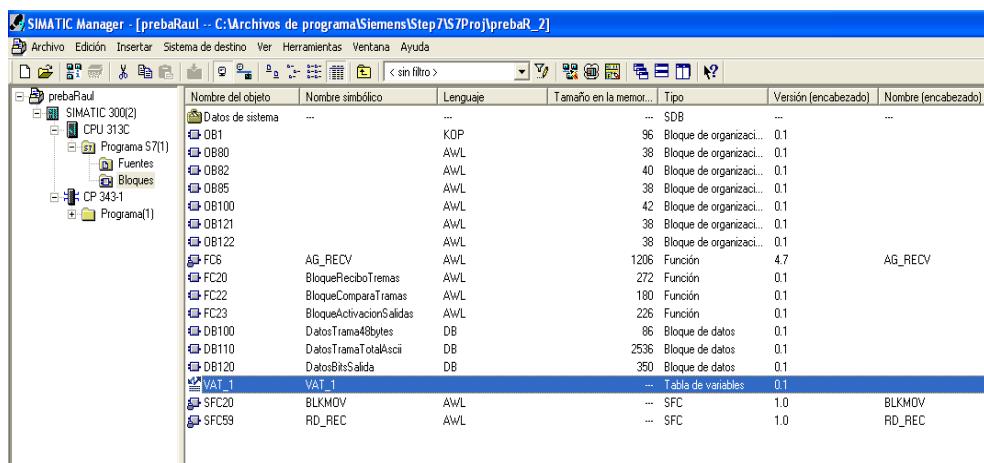


Figura 34. proyecto.

En la figura 35 podemos ver como está configurado el hardware: podemos ver que hemos introducido la CPU y los módulos de salidas digitales y el de comunicación en el bastidor 0 en la parte superior.

En la parte inferior podemos observar las referencias, la versión, la dirección MPI y las direcciones de las salidas y entradas correspondientes a cada objeto.

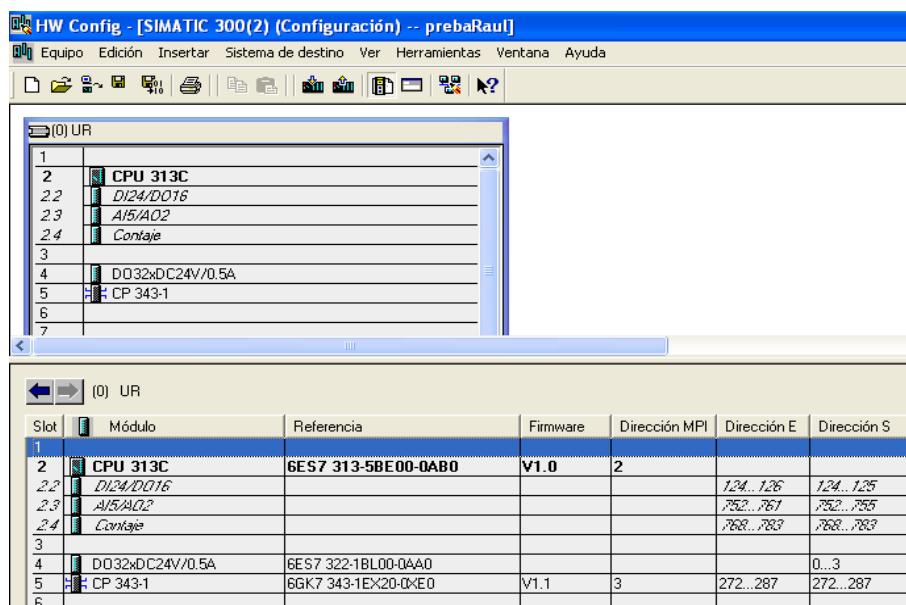


Figura 35. Configuración de hardware.

Esta CPU dispone de una *Imagen de proceso Salidas* de 1024 bits, esto significa que podría activar 1024 electroválvulas. Teniendo en cuenta que disponemos de 127 DBs con una memoria de 16420 bytes cada uno, es posible efectuar una imagen de 1024 x 1024 píxeles. Utilizaríamos un DB para la entrada de tramas, serían 1024 bytes. Para guardar todas tramas necesitaríamos 64 DBs, ya que caben 16 tramas por DBs, y para guardar los bits de las salidas digitales necesitaríamos 8 DBs, ya que serían $1024 \times 1024 = 1048576$ bits, 131072 bytes. Estamos hablando de una cortina de agua de 8-10 metros de ancho, pero tendríamos problemas con la altura y la imagen se deformaría.

En la siguiente imagen podemos observar los datos característicos de la CPU.

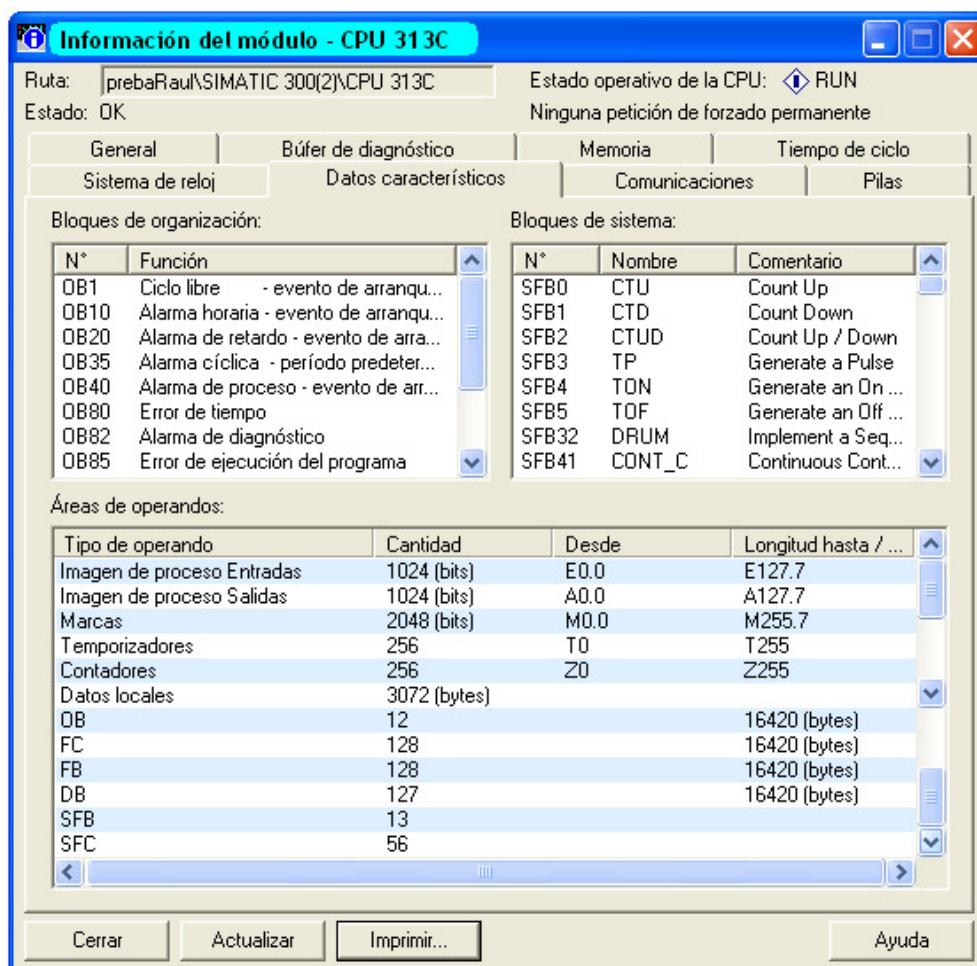


Figura 36. Datos característicos de la CPU 313C.

8.6. Configuración CP

Para la comunicación con el ordenador utilizaremos CP343-1. Este módulo permite conectar SIMATIC S7-300 a industrial Ethernet, a parte de posibilitar la comunicación con otros nodos en Ethernet. La CP funciona como PROFINET-IO Controller y como IO-Device, pero también se puede configurar para comunicación TCP/IP con otros dispositivos.

En la siguiente imagen podemos observar como le introducimos la IP y su Máscara correspondiente. Esta IP y Máscara tiene que estar en el mismo rango que el ordenador para poder comunicar.

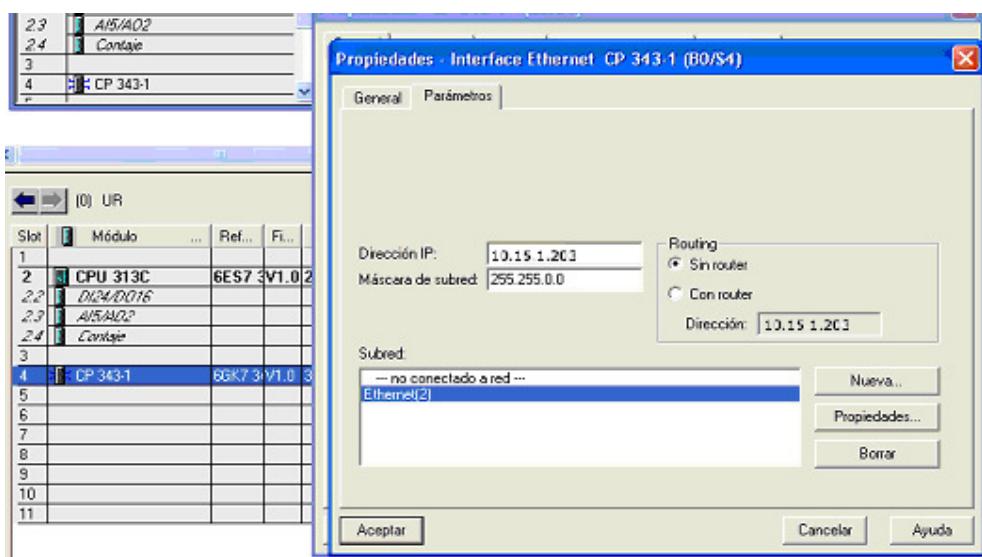


Figura 37. Configuración CP.

Como podemos observar el ordenador utilizado está en el mismo rango:

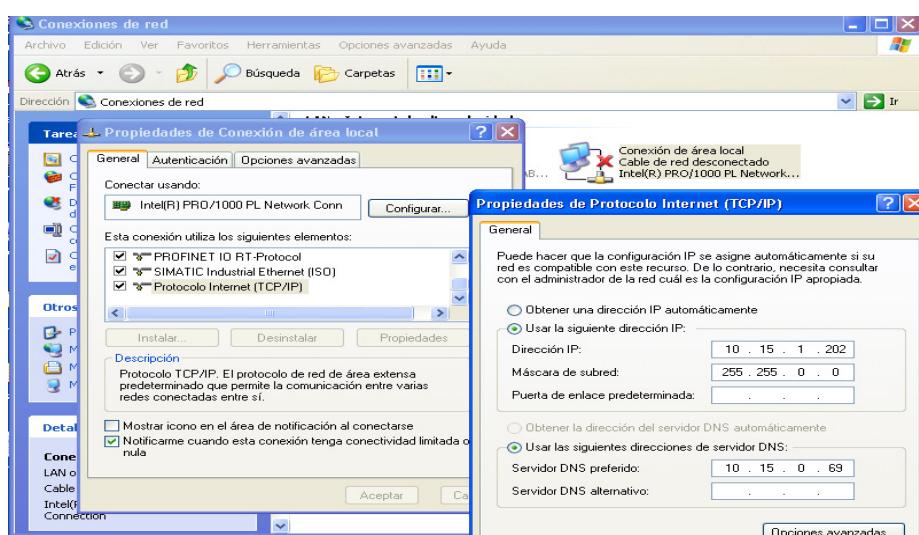


Figura 38. Configuración IP del ordenador.

8.7. Configuración Enlace

Seguidamente tenemos que realizar un enlace a la CPU para poder comunicar mediante TCP/IP.

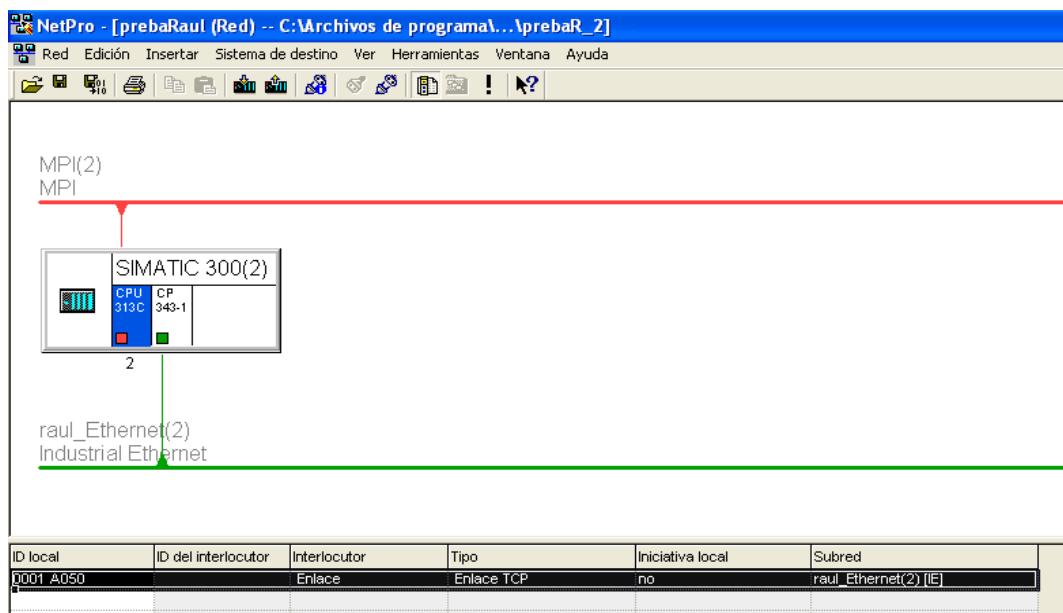


Figura 39. Configuración del enlace.

Ahora vemos las propiedades del enlace, donde especificamos que el enlace será TCP.

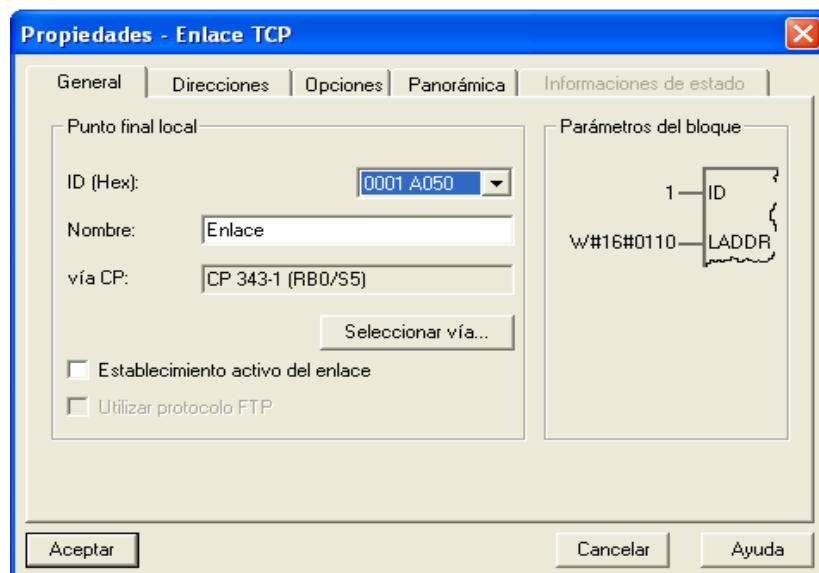


Figura 40. Enlace TCP.

En el apartado de direcciones ponemos la dirección y introduciremos la IP local y el puerto por el cual deseamos comunicar. Si no especificamos el interlocutor podrá conectarse cualquiera que esté en el mismo rango y comunique por el puerto 10500.

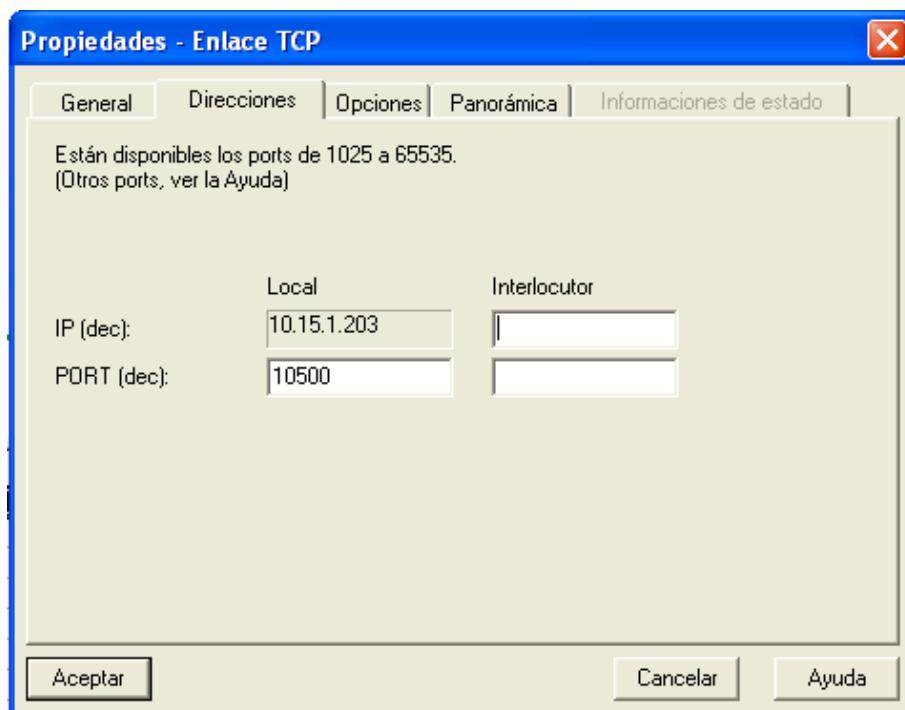


Figura 41. Direcciones del enlace.

8.8. Programación de Bloques

El programa cuando empieza a ejecutarse el primer elemento que abre es el OB 1 (bloque que organización).

8.8.1. OB1

Como se puede observar en el código siguiente el primer segmento hace una llamada al bloque FC 20. Este bloque es el destinado al recibo de tramas de 48 bytes, guardarlas en el bloque de datos DB 100 y posteriormente hacer la copia de dicho DB al DB 110 tal como se ha explicado anteriormente.

El segundo segmento hace una llamada al bloque FC 22. Este es el encargado de comparar el contenido del DB 110 y activar los bits del DB 120.

El tercer segmento hace una llamada al bloque FC 23. Este bloque de datos es el encargado de activar las salidas digitales correspondientes, moviendo sobre estas salidas los bits del DB 120.

En el segmento cuarto y quinto se genera un bit de intermitencia, que es el responsable del tiempo de ciclo de conmutación de las electroválvulas. Como podemos observar el ciclo de conmutación (40 ms utilizado en el prototipo). Utilizando las electroválvulas antes mencionadas podríamos conmutar a 20ms perfectamente, ya que esta conmutación se genera por un flanco positivo del "BitIntermitencia".

Segm. 1 : Título:

```
FC20
"BlqueRec
iboTremas"
(CALL)---
```

Segm. 2 : Título:

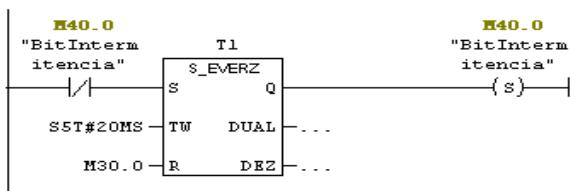
```
FC22
"BlqueCom
paraTramas
"
(CALL)---
```

Segm. 3 : Título:

```
FC23
"BlqueAct
ivacionSal
idas"
(CALL)---
```

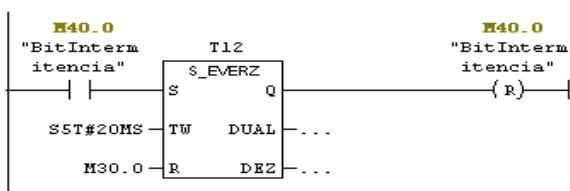
Segm. 4 : Título:

Comentario:



Segm. 5 : Título:

Comentario:



8.8.1.1. Organigrama OB1

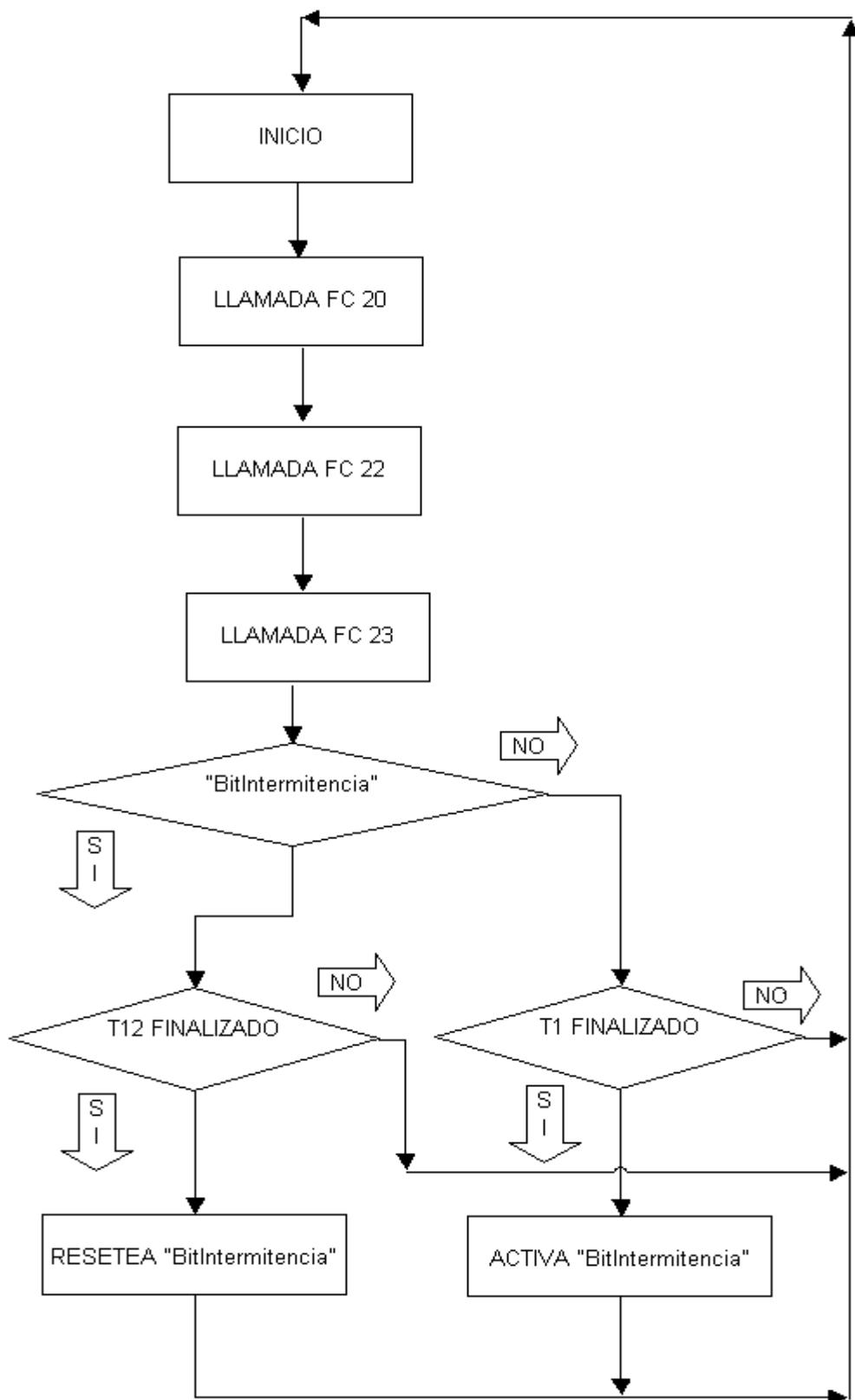


Figura 42. Organigrama OB1.

8.8.2. FC 20("BloqueReciboTramas")

En el interior del bloque FC 20 haremos uso de otro FC que esta en la librería de Simatic FC 6 AG_RECV

El bloque FC AG_RECV / AG_LRECV recibe del CPEthernet los datos transferidos a través de un enlace ISO-Transport configurado.

En el segmento 1 especificamos los bytes que recibiremos. Hasta que no recibamos 48 bytes no lo guardará en el DB100, y especificamos la posición. El LADDR ha de coincidir con la establecida en el enlace.

Una vez recibido los 48 bytes correctamente activamos una marca "ReciboTramas"

```
FC20 : Titulo:  
Comentario:  
  
Segm. 1: Título:  
Comentario:  
  
UN      "HabilitacionRecibo"          M100.0  
SPBN    col  
CALL    "AG_RECV"                   FC6  
ID      :=1  
LADDR  :=W#16#110  
RECV   :=P#DB100.DBX0.0 BYTE 48  
MNR    :="ReciboTrama"             M100.1  
ERROR  :=M60.0  
STATUS :=MW101  
LEN    :=MW135  
  
col: NOP    0
```

En el segmento 2, cuando hemos recibido una trama, tenemos la marca "ReciboTramas", empezamos a hacer la copia del DB100 al DB110 moviendo 48 posiciones cada vez que se recibe una trama.

Una vez recibidas 48 tramas , es decir hemos recibido toda la imagen, y la hemos guardado, activaremos la marca "ComparaTramas"

Segm. 2): Título:

Comentario:

```

U      "ReciboTrama"          M100.1
SPBN  co2
S      "HabilitacionRecibo"   M100.0
R      "ReciboTrama"          M100.1
L      48
T      "j"                   MW202
L      "I"                   MW200
L      "j"                   MW202
*I
T      "Resultado"           MW204

co3: NOP  0
      L      "Resultado"           MW204
      L      "u"                  MW218
      +I
      T      "v"                   MW220
      L      "v"                   MW220
      SLD  3
      T      "PunteroRsultado"    MD210
      AUF  "DatosTrama48bytes"   DB100
      L      "u"                  MW218
      SLD  3
      T      "punteroi"          MD214
      L      DBB ["punteroi"]      MD214
      AUF  "DatosTramaTotalAscii" DB110
      T      DBB ["PunteroRsultado"] MD210

      L      "u"                  MW218
      L      1
      +I
      T      "u"                   MW218
      L      "u"                   MW218
      L      48
      >I
      SPBN co3
      L      0
      T      "u"                   MW218
      L      0
      SLD  3
      T      "punteroi"          MD214
      L      1
      L      "I"                   MW200
      +I
      T      "I"                   MW200
      L      "I"                   MW200
      L      48
      >=I
      SPBN co4
      L      0
      T      "I"                   MW200
      S      "ComparaTramas"     M100.2
      SPA  co2

co4: NOP  0
      R      "HabilitacionRecibo"   M100.0
co2: NOP  0

```

8.8.2.1. Organigrama FC20

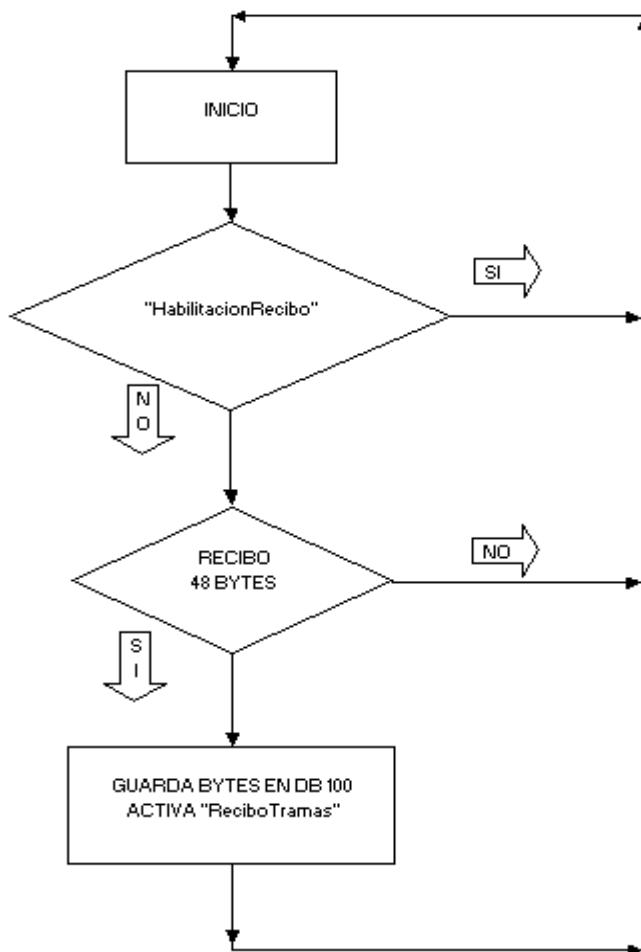
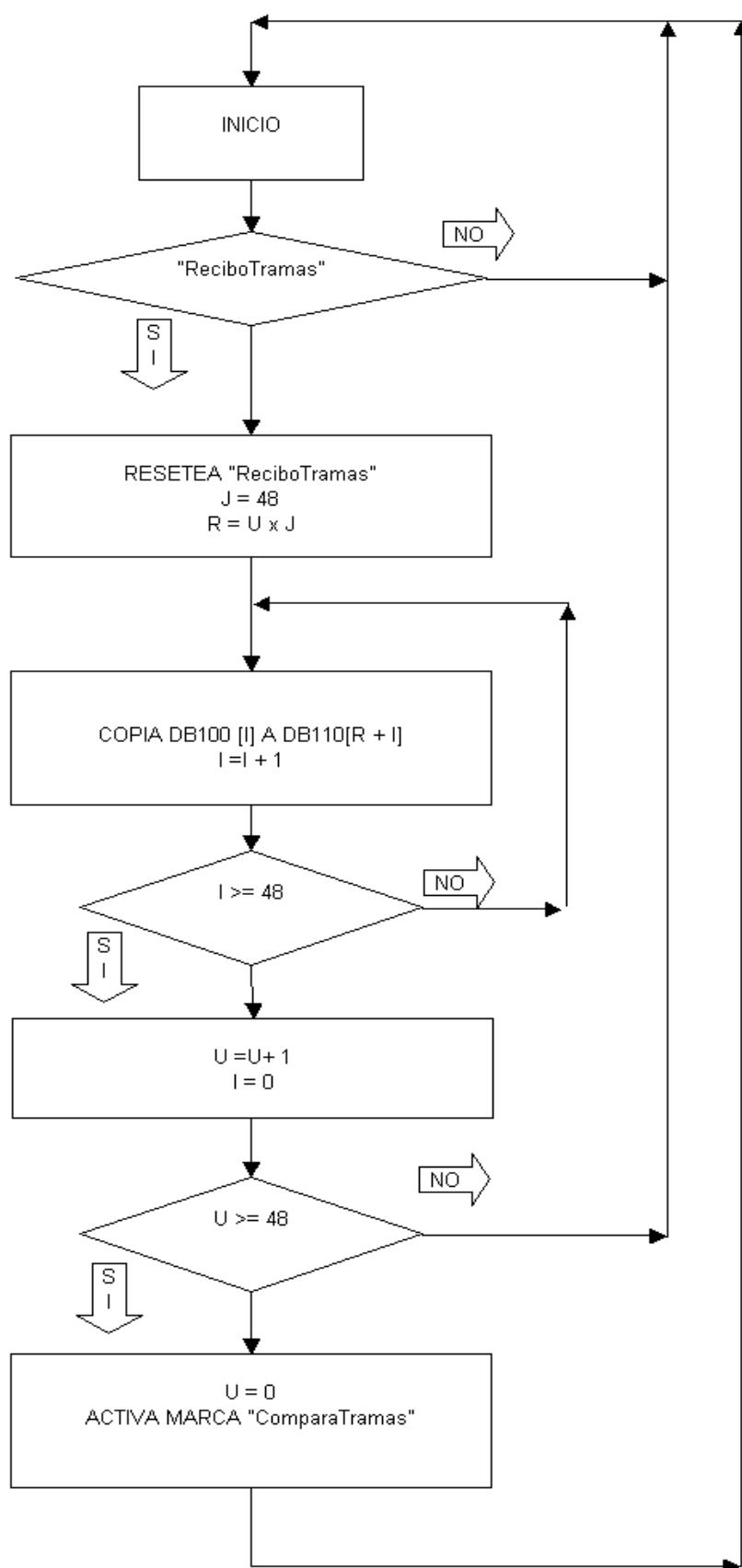


Figura 43. Organigrama segmento1 del FC 20.

En el segmento 2, cuando vemos la expresión DB100[I], I se refiere a la posición que ocupa en el DB. Como si fuera un vector. Aunque cuando programamos en STEP 7, lo que hay que hacer es apuntar al espacio de memoria del DB, utilizando punteros. Es decir si tenemos un DB con formato BOOL(true, false) cada posición ocupa un bit, pero si es formato CHAR(carácter) cada posición ocupa un byte.

Tabla 2. Relación vector, puntero.

posición del elemento	El puntero apunta a la dirección:	DB(BOOL)	DB(CHAR)
		0.0	0.0
2		0.1	1.0

**Figura 44.** Organigrama segmento 2 del FC 20.

8.8.3. FC 22 ("BloqueComparaTramas")

Una vez tengamos todas las tramas guardadas en el DB110, tendremos la marca "ComparaTramas" activa.

Podemos empezar a comparar uno a uno los caracteres. Si el carácter es un "0" activaremos el bit correspondiente en el DB 120, si en caso contrario es un "1" lo resetearemos.

Esta comparación es así, ya que la matriz binaria en Matlab, un píxel negro equivalía a un a "0". Y quería que las gotas dibujen la imagen, no que cayera el agua y el dibujo se creará por falta de agua.

Una vez se comparen los 2304 elementos de la imagen se activará la marca "PreparadoVal" que da paso a las activaciones de las electroválvulas. El bloque sería el siguiente:

```
FC22 : Título:  
Comentario:  
  
Segm. 1: copia DB110 a DB120  
Comentario:  
  
U      "ComparaTramas"          M100.2  
SPBN  co2  
UN    "m100.3"                  M100.3  
SPBN  co5  
S     "m100.3"                  M100.3  
L     0  
SLD   3  
T     #punteroH  
L     P#0.0  
T     #punteroH2  
  
co5: NOP  0  
L     "H"                      MW222  
SLD   3  
T     #punteroH  
AUF   "DatosTramaTotalAscii"    DB110  
L     DBB [#punteroH]  
L     '0'  
==I  
SPBN  co3  
AUF   "DatosBitsSalida"        DB120  
S     DBX [#punteroH2]  
SPA   co4
```

```
co3: NOP    0
      AUF    "DatosBitsSalida"           DB120
      R      DBX [#punteroH2]

co4: NOP    0
      L      #punteroH2
      L      P#0.1
      +I
      T      #punteroH2
      L      1
      L      "H"                      MW222
      +I
      T      "H"                      MW222
      L      "H"                      MW222
      L      2304
      >=I
      SPBN  co2
      L      0
      T      "H"                      MW222
      L      P#0.0
      T      #punteroH2
      S      "PreparadoVal"            M100.4
      R      "ComparaTramas"          M100.2
      R      "m100.3"                 M100.3

co2: NOP    0
```

8.8.3.1. Organigrama FC22

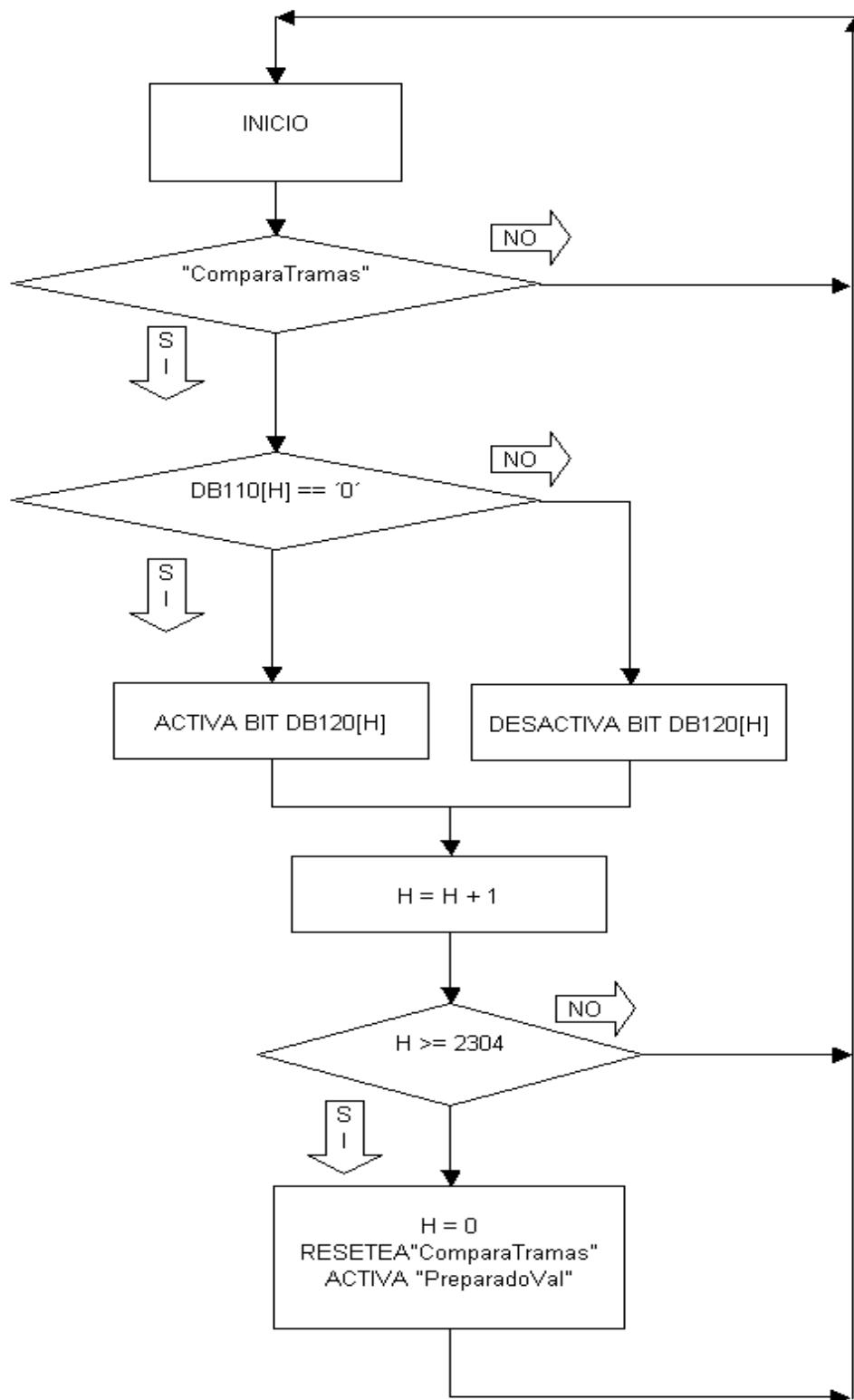


Figura 45. Organigrama FC 22.

8.8.4. FC 23("BloqueActivacionSalidas")

Una vez tengamos la marca "PreparadoVal" activa, con cada flanco positivo de la marca "BitIntermitencia", moveremos los bits del DB120 a las salidas digitales. Cada vez que tengamos este flanco moveremos los 48 bits siguientes del DB hasta llegar a las 48 tramas.

Los 16 primeros bits de cada trama, sobre las salidas comprendidas entre 124.0 hasta 125.7 correspondientes a las salidas digitales integradas en la CPU. Los 32 bits restantes los moveremos a las salidas digitales comprendidas entre 0.0 hasta la 3.7 correspondientes al modulo de 32 salidas digitales.

Una vez se llega a 48 movimientos en las salidas se activa la marca "final", y se resetearán todas las salidas. luego se resetearán las marcas "PreparadoVal", "HabilitacionRecibo", "final" dejando el programa a la espera de que reciba otra vez una imagen. El bloque sería el siguiente:

Segm. 1: Título:

Comentario:

```

U      "PreparadoVal"          M100.4
UN     "inicioPunteroVal"      M100.7
FP     M      50.0
SPBN   col
L      0
SLD    3
T      "punteroval"           MD232
L      2
SLD    3
T      "punteroval2"          MD240
S      "inicioPunteroVal"      M100.7
L      0
T      "cont2"                MW236

col: NOP 0

```

Segm. 2 : Título:

el 8 sera un 48

```

U      "PreparadoVal"          M100.4
U      "BitIntermitencia"       M40.0
FP     M      50.1
SPBN   co2
L      1
L      "contador1"             MW230
+I
T      "contador1"             MW230
L      "contador1"             MW230
L      48
>=I
SPBN   co3
R      "intermitencia"         M100.5
R      "PreparadoVal"          M100.4
S      "final"                 M100.6
L      0
T      "contador1"             MW230

co3: NOP 0
co2: NOP 0

```

Segm. 3 : Título:

Comentario:

```
U      "final"          M100.6
L      0
T      PAW  124
L      0
T      PAD   0
R      "PreparadoVal"    M100.4
R      "final"           M100.6
R      "inicioPunteroVal" M100.7
R      "HabilitacionRecibo" M100.0
```

Segm. 4 :

Comentario:

```
U      "PreparadoVal"    M100.4
UN     "final"           M100.6
SPBN   co9
AUF    "DatosBitsSalida" DB120
L      DBW ["punteroval"] MD232
T      PAW  124
L      DBD  ["punteroval2"] MD240
T      PAD   0
co9:  NOP   0
```

Segm. 5 : Título:

Comentario:

```
U      "preparadoVal"    M100.4
U      "BitIntermitencia" M40.0
FP    M      50.2
SPBN  co6
AUF    "DatosBitsSalida" DB120
L      "cont2"           MW236
L      6
+I
T      "cont2"           MW236
SLD   3
T      "punteroval"       MD232
L      "cont2"           MW236
L      2
+I
T      "puntval2"         MW238
L      "puntval2"         MW238
SLD   3
T      "punteroval2"       MD240
co6:  NOP   0
```

Así pues realizaremos el organigrama del bloque FC 23, resumiendo los segmentos anteriores en dos organigramas:

8.8.4.1. Organigrama FC23

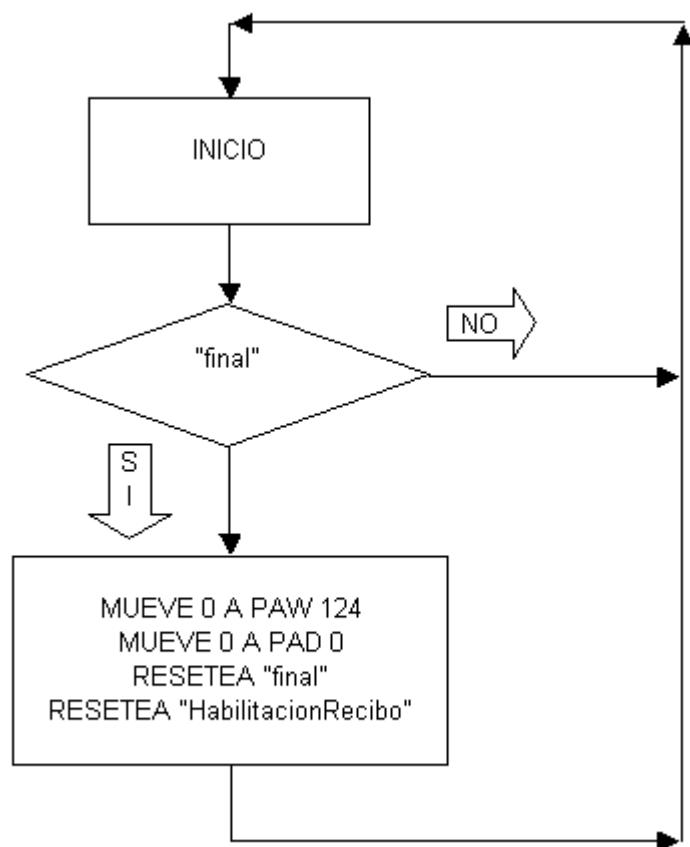


Figura 46. Organigrama FC 23 parte1.

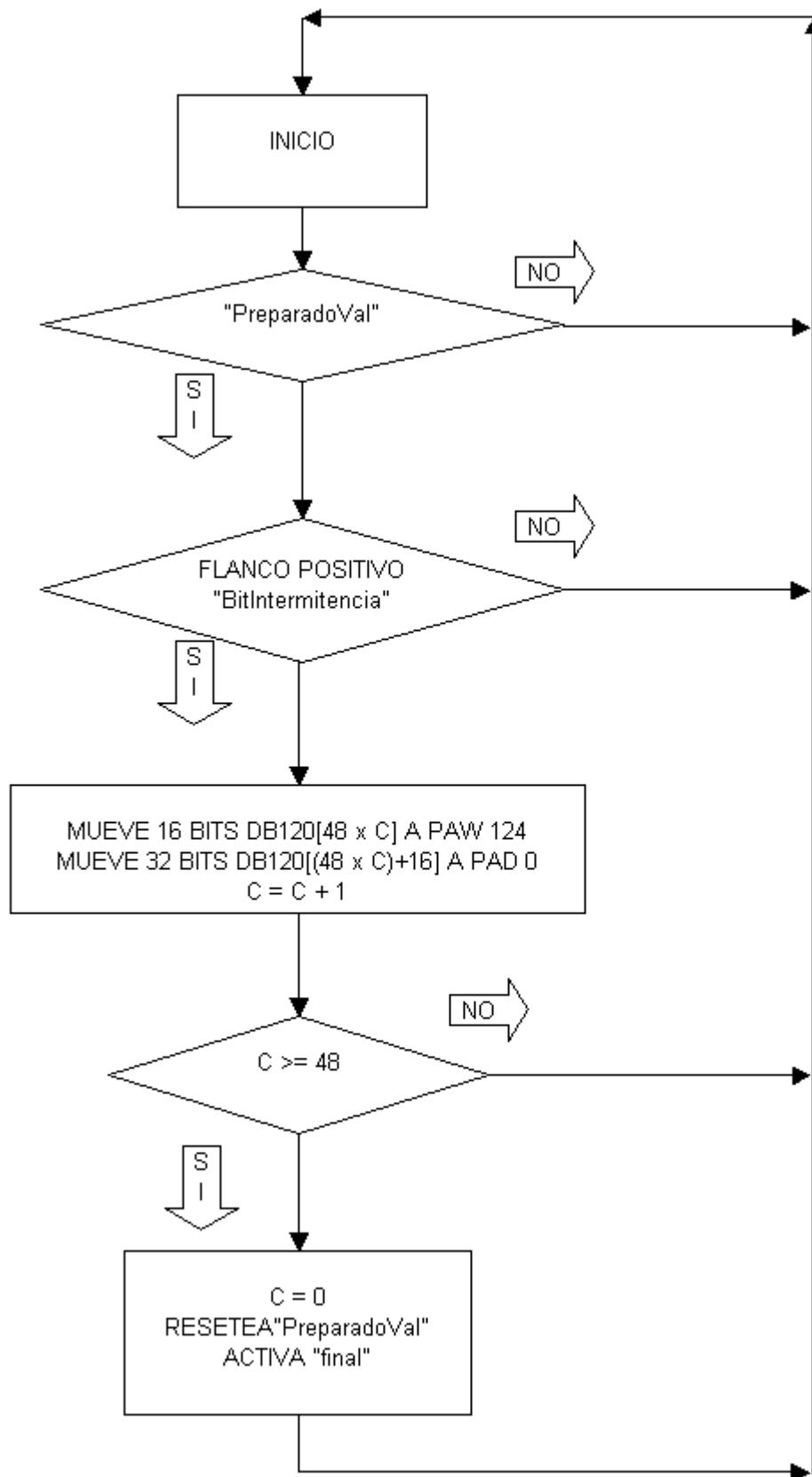


Figura 47. Organigrama FC23 parte2.

Si abrimos los tres DB en Online podemos observar como funciona el programa. En las figuras 48 y 49 podemos ver la última trama de la imagen. Como podemos observar en el dB 100 se guarda la trama, después se hace una copia en el DB 110 y luego se pasa a formato bit en el DB 120. Para ello abriremos la siguiente imagen:

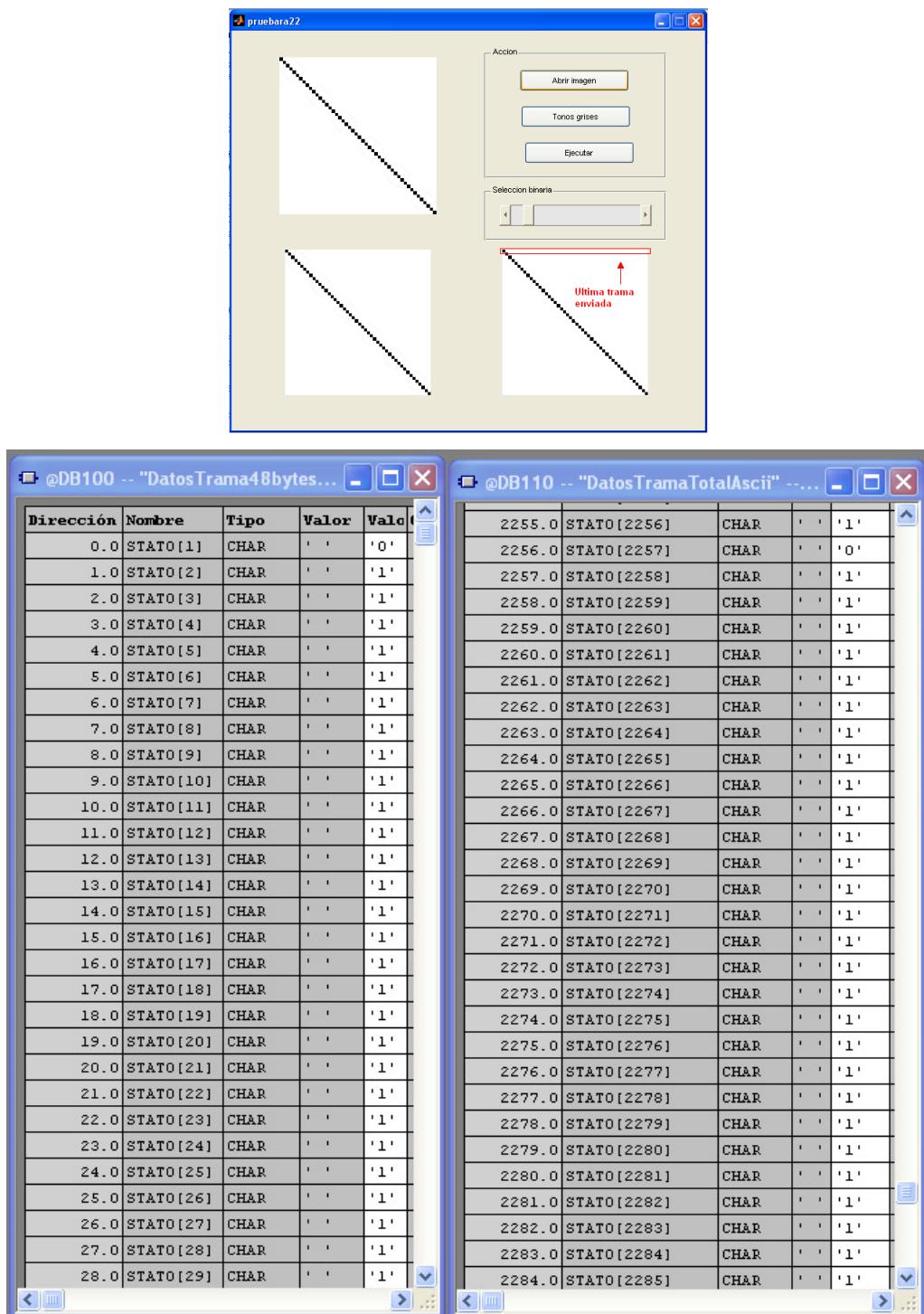


Figura 48. DB100 y DB110 Online.

Podemos observar como la última trama en el DB110 empieza en la dirección 2256 :

En la siguiente imagen podemos observar como se convierte el dato en formato carácter a formato bit. Pasando '1' a False y '0' a True.

The image shows two side-by-side data browsing windows from a SIMATIC Manager interface. The left window, titled '@DB110 -- "DatosTramaTotalAscii"', displays a list of memory locations from 2255.0 to 2284.0. Each location contains a character value ('1' or '0') and its data type is listed as CHAR. The right window, titled '@DB120 -- "DatosBitsSalida" -- prebaRau...', shows the corresponding bit values for each character. The data types here are BOOL, and the values are either FALSE or TRUE. The bit numbering starts at 281.7 for the first character ('1') and continues down to 285.4 for the last character ('0').

DB110 Address	DB110 Value	DB120 Address	DB120 Value
2255.0	STATO[2256]	CHAR	'1'
2256.0	STATO[2257]	CHAR	'0'
2257.0	STATO[2258]	CHAR	'1'
2258.0	STATO[2259]	CHAR	'1'
2259.0	STATO[2260]	CHAR	'1'
2260.0	STATO[2261]	CHAR	'1'
2261.0	STATO[2262]	CHAR	'1'
2262.0	STATO[2263]	CHAR	'1'
2263.0	STATO[2264]	CHAR	'1'
2264.0	STATO[2265]	CHAR	'1'
2265.0	STATO[2266]	CHAR	'1'
2266.0	STATO[2267]	CHAR	'1'
2267.0	STATO[2268]	CHAR	'1'
2268.0	STATO[2269]	CHAR	'1'
2269.0	STATO[2270]	CHAR	'1'
2270.0	STATO[2271]	CHAR	'1'
2271.0	STATO[2272]	CHAR	'1'
2272.0	STATO[2273]	CHAR	'1'
2273.0	STATO[2274]	CHAR	'1'
2274.0	STATO[2275]	CHAR	'1'
2275.0	STATO[2276]	CHAR	'1'
2276.0	STATO[2277]	CHAR	'1'
2277.0	STATO[2278]	CHAR	'1'
2278.0	STATO[2279]	CHAR	'1'
2279.0	STATO[2280]	CHAR	'1'
2280.0	STATO[2281]	CHAR	'1'
2281.0	STATO[2282]	CHAR	'1'
2282.0	STATO[2283]	CHAR	'1'
2283.0	STATO[2284]	CHAR	'1'
2284.0	STATO[2285]	CHAR	'1'
		281.7	DB_VAR[2256]
			BOOL
			FALSE
			FALSE
		282.0	DB_VAR[2257]
			BOOL
			FALSE
			TRUE
		282.1	DB_VAR[2258]
			BOOL
			FALSE
		282.2	DB_VAR[2259]
			BOOL
			FALSE
		282.3	DB_VAR[2260]
			BOOL
			FALSE
		282.4	DB_VAR[2261]
			BOOL
			FALSE
		282.5	DB_VAR[2262]
			BOOL
			FALSE
		282.6	DB_VAR[2263]
			BOOL
			FALSE
		282.7	DB_VAR[2264]
			BOOL
			FALSE
		283.0	DB_VAR[2265]
			BOOL
			FALSE
		283.1	DB_VAR[2266]
			BOOL
			FALSE
		283.2	DB_VAR[2267]
			BOOL
			FALSE
		283.3	DB_VAR[2268]
			BOOL
			FALSE
		283.4	DB_VAR[2269]
			BOOL
			FALSE
		283.5	DB_VAR[2270]
			BOOL
			FALSE
		283.6	DB_VAR[2271]
			BOOL
			FALSE
		283.7	DB_VAR[2272]
			BOOL
			FALSE
		284.0	DB_VAR[2273]
			BOOL
			FALSE
		284.1	DB_VAR[2274]
			BOOL
			FALSE
		284.2	DB_VAR[2275]
			BOOL
			FALSE
		284.3	DB_VAR[2276]
			BOOL
			FALSE
		284.4	DB_VAR[2277]
			BOOL
			FALSE
		284.5	DB_VAR[2278]
			BOOL
			FALSE
		284.6	DB_VAR[2279]
			BOOL
			FALSE
		284.7	DB_VAR[2280]
			BOOL
			FALSE
		285.0	DB_VAR[2281]
			BOOL
			FALSE
		285.1	DB_VAR[2282]
			BOOL
			FALSE
		285.2	DB_VAR[2283]
			BOOL
			FALSE
		285.3	DB_VAR[2284]
			BOOL
			FALSE
		285.4	DB_VAR[2285]
			BOOL
			FALSE
			FALSE

Figura 49. DB110 y DB120 Online.

CAPÍTULO 9:

DISEÑO DEL

MONTAJE

9.1. Esquema montaje

En la siguiente figura podemos observar una posibilidad del montaje de la cortina programable:

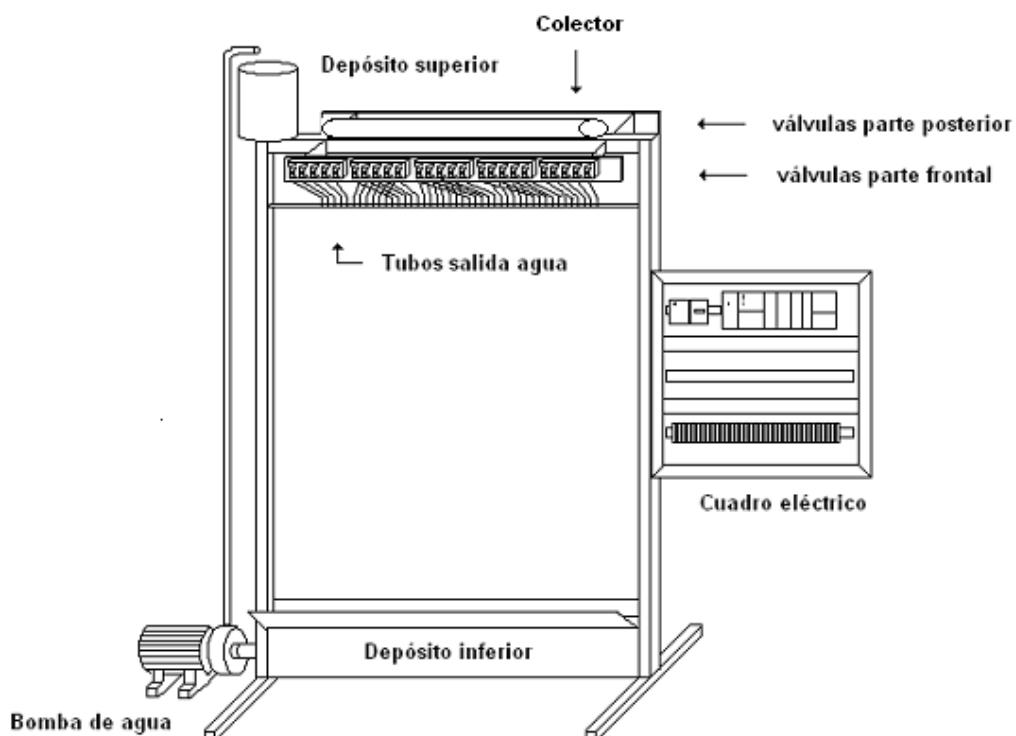


Figura 50. Esquema montaje.

En la figura 50 podemos observar como podría ser el montaje, aunque según las especificaciones del cliente se puede modificar.

El depósito superior , el colector y las electroválvulas podrían ir clavadas en el techo y así no habría chasis. El cuadro eléctrico puede ir separado del chasis, e iría en un armario estanco.

9.2. Elementos principales

Seguidamente especificaremos los elementos principales del montaje:

9.2.1. Bomba de agua

Esta bomba es la encargada de bombear agua del depósito inferior al depósito superior. En el depósito superior habrá un sensor de nivel de seguridad ,y en el depósito inferior habrá un sensor que arrancará la bomba y permanecerá el nivel entre un rango definido. La unión entre bomba y depósito se realiza mediante una manguera. Los cálculos de la bomba vendrán dados por la altura de la cortina y la cantidad de electroválvulas.

Para calcular la altura podemos hacer una aproximación : sin tener en cuenta el rozamiento del aire, excepto la velocidad terminal de las gotas. Y suponiendo que son gotas y no chorros de agua, ya que incrementaría su velocidad. Y sin tener en cuenta el ciclo de Scan. Suponemos las gotas como indica la imagen:

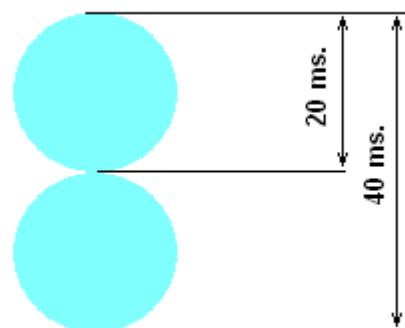


Figura 51. Suposición de gotas de agua.

Hacemos comutar las electroválvulas cada 20ms, lo que supone que la imagen formada por 48 píxeles tarda 0,96 s.

Utilizando la ecuaciones del movimiento rectilíneo uniformemente acelerado:

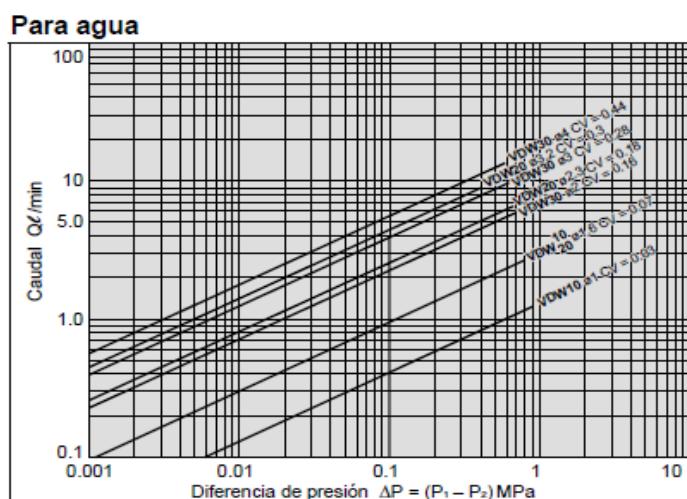
$$\begin{aligned}\alpha &= \text{cte} \\ v &= v_0 + \alpha \cdot t \\ x &= x_0 + v_0 \cdot t + \frac{1}{2} \cdot \alpha \cdot t^2\end{aligned}\quad (7)$$

Sabiendo que la velocidad terminal es de 9 m/s tenemos que tarda a llegar a esa velocidad 0,91 s., y recorre 4,05 metros. En los 0,05 segundos restantes de la imagen recorre 45 centímetros a 9 m/s. Lo que nos da una altura total de 4,5 metros.

La cortina será de 4,5 m de altura y la distancia del depósito superior a la bomba sería de 5m, con lo cual necesito una bomba que supere los 0,5 bares de presión.

Para el cálculo de caudal suponemos que hacemos una imagen cada 20 segundos y que la imagen esta formada por la mitad de los píxeles, es decir la mitad son gotas y la otra mitad es aire.

Si miramos las especificaciones de las válvulas:



Interpretación Para generar un caudal de agua de 4 l/min a una presión del gráfico diferencial de 0.1 MPa, se requiere un área efectiva con factor Cv 0.28 (VWD30@3) o mayor.

Cómo hallar el caudal para agua

- Fórmula basada en el factor Cv

$$Q = 14.2 \cdot Cv \cdot \sqrt{10.2 \cdot \Delta P} \dots \text{l/min}$$
- Fórmula basada en el área efectiva (Smm²)

$$Q = 0.8 \cdot S \cdot \sqrt{10.2 \cdot \Delta P} \dots \text{l/min}$$

Q: caudal (l/min), ΔP: diferencia de presión (P₁–P₂), P₁: presión de alimentación (MPa), P₂: presión secundaria (MPa), S: área efectiva (mm²), Cv: factor Cv

Figura 52. Especificaciones de las válvulas.

Suponemos que la distancia media entre el nivel del depósito superior y la electroválvula es de 40 cm y que a la salida tiene presión atmosférica. Sabiendo que 1 m.c.a. = 0,1 kg/cm² = 9,81 kPa (kilopascal), tenemos una diferencia de 0,0039 Mpa. Sabiendo que la electroválvula utilizada es una

VDW30 con orificio de 3 mm de diámetro. Y utilizando la formula basada en el factor Cv: tenemos un caudal de 0,79 l/min por electroválvula.

Sabiendo que tenemos 48 válvulas , que hacen 4 imágenes por minuto y duran 0,96 s cada imagen y que la imagen esta compuesta por la mitad de píxeles: Tenemos un caudal necesario para la bomba de mínimo 1,26 l/min.

En el caso de efectuar imágenes repetidamente sería 18,96 l/min.

9.2.2. Depósito superior

Este depósito alimenta de agua las válvulas, siempre tendrá un nivel definido. El material utilizado será tubo de acero inoxidable y la capacidad será de 15 litros. Siendo el volumen $V = \pi * r^2 * h$ y deseamos una altura de 50cm el diámetro será: 195,49 mm. En el mercado el tubo que mas se le asemeja sería el de 200mm de diámetro. A la salida de este depósito habrá una válvula 2/2 de 1/2", conectada al colector que alimenta todas las electroválvulas. Este colector tendrá también una pequeña válvula 2/2 por si algún día hay que reparar alguna electroválvula se pueda cerrar el grifo del depósito superior y vaciar el colector para poder desmontar la electroválvula.

El colector tendrá dos salidas por cada 5 electroválvulas. Esta conexión se realizará mediante tubo neumático de 8 mm y sus racores correspondientes.

9.2.3. Depósito inferior

Utilizado para recoger el agua que cae. El material usado será acero inoxidable. Consistirá en una U cuadrada con capacidad aproximada de 15 litros y una distancia de 1 metro de largo. Con lo cual tendría 120mm de lado.

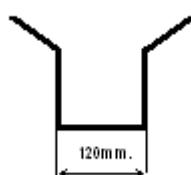


Figura 53. Detalle depósito inferior.

La bomba escogida de la marca Czerweny con referencia MB-QB60 : se trata de una bomba monofásica alimentada a 230 V, y una potencia de 0,37kW. Donde las principales características son:

- Aumento progresivo de presión durante el arranque, sin golpes o flujo pulsante.
- Silenciosas, económicas y libres de mantenimiento.

- Motor cerrado monofásico IP44, sello mecánico de cerámica y grafito, impulsor de bronce con canales radiales, Rodamiento sellados lubricados de por vida.



Figura 54. Bomba de agua escogida.

En la siguiente imagen podemos observar la relación entre altura y caudal suministrado por diferentes modelos de bombas. El modelo MB-QB60 sería suficiente para la aplicación:

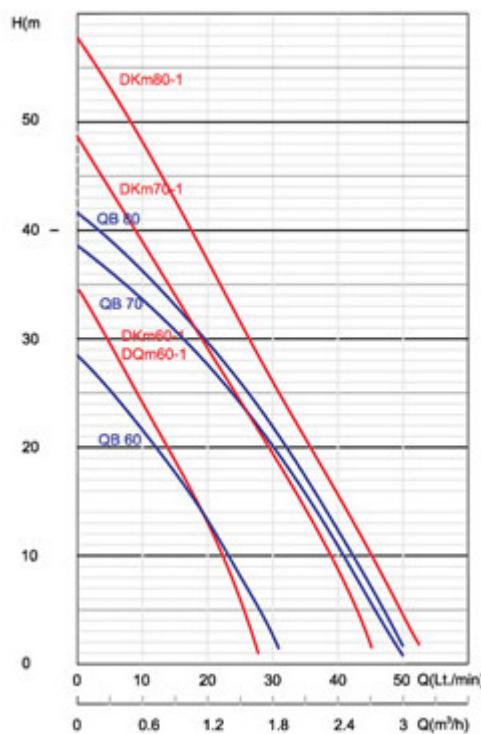


Figura 55. Especificaciones de la bomba de agua.

9.2.4. Electrovávulas

Las electroválvulas son de la marca SMC. La referencia es VDW33-5G-3-Q. Estas válvulas son activadas a 24VDC , consumen una potencia de 3W, Y el orificio interior es de 3mm. Las electroválvulas estarán montadas la mitad en la parte frontal y la otra mitad en la parte posterior. La explicación es que al tener las salidas muy separadas (17,5mm) habrá que guiar el agua mediante tubo de neumática de 4mm. Así colocaremos las salidas separadas 8 mm entre tubos para poder apreciar la imagen.

9.2.5. Chasis

Para la fabricación del chasis utilizaremos perfil de acero inoxidable 30 x 30. Utilizaremos 12 metros aproximadamente.

9.2.6. Guía para tubos

Para que la imagen sea nítida la caída de agua tiene que ser totalmente vertical para ello utilizaremos un guía de teflón de 30 x 30 mm de teflón ,la cual taladraremos como marca la imagen. El tamaño total de la cortina será de 66,2 centímetros de ancho.

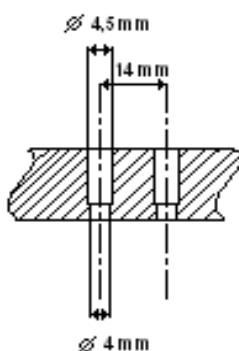


Figura 56. Detalle guía de tubos.

Taladrando el perfil de esta manera permitimos que el chorro de agua caiga verticalmente ,y que el tubo de 4 milímetros. Quede cogido por fricción a las paredes del perfil. Taladrados 25mm con una broca de 4,5mm y los 5mm restantes con una de 4mm.

9.2.7. Colector

Será un tubo de 40mm de diámetro de acero inoxidable de 1,5 mm de espesor. En el irán soldados manguitos de $\frac{1}{4}$ " , para conectar los rafrescos de

tubo de 8mm. habrá tantas salidas como el doble de bloques de válvulas, ya que las alimentamos por los dos laterales.

Podemos hacer una aproximación mediante el área de los conductos para definir estos.

Teniendo 48 válvulas con orificio de 3mm, resulta un área de $339,12 \text{ mm}^2$, Y el tubo de 40mm tiene un área de sección de 1256 mm^2 , con lo cual es más que suficiente.

9.2.8. Cuadro Eléctrico

El cuadro eléctrico será de inoxidable 600 x 600 x 250, estará formado por una lámina de inoxidable, y tendrá un diferencial, cuatro magnetotérmicos, un guardamotor, 2 sondas de nivel, el autómata programable y módulos entradas-salidas y módulo de comunicación, las guías y canaletas correspondientes.

9.2.8.1 Cálculo de protecciones eléctricas

Consumidores de potencia 24V DC:

- Electroválvulas: 3W(cada una)
- CPU: 16,8 W
- Modulo de Salidas digitales: 6,6W
- CP343-1: 10W

Total de consumo: 7,39 A.

La maniobra de los elementos conectados a 24VDC quedan protegidos por la fuente de alimentación , a demás, las salidas digitales también disponen de protección contra sobrecorrientes.

Así queda por definir las secciones de los cables que los alimentan, Para estandarizar utilizaré cable 1mm de sección, como podemos observar en la siguiente tabla este cable esta sobredimensionado.

Tabla 3.Caída de tensión de circuitos de continua.

Elemento	Potencia(W)	Intensidad (A)	Conductor de	Sección (mm ²)	Longitud Línea (m)	Caída de Tensión (V)	Caída de Tensión %
CPU	16,8	0,7	Cobre	1	1	0,02	0,09
Salidas digitales	6,6	0,27	Cobre	1	1	0,009	0,03
CP343-1	10	0,41	Cobre	1	1	0,01	0,04
8 Electroválvulas	24	1	Cobre	1	5	0,17	0,7

Consumidores de potencia 230V AC:

- Bomba monofásica: 0,37kW
- Luz: 50W
- Fuente alimentación: 352W
- Sondas de nivel: 2,4W

En la tabla siguiente podemos observar como se ha dividido la alimentación eléctrica en circuitos donde cada uno dispone de su protección:

Tabla 4. Protecciones de los circuitos 230 V alterna.

Circuito	Descripción	Interruptor
Alimentación	Interruptor magnetotérmico principal	QF0
	Interruptor Diferencial	QD0
Mando 24V DC	Interruptor magnetotérmico fuente	QF1
Mando 230V AC	Interruptor magnetotérmico sondas.	QF2
Bomba de agua	Guardamotor bomba de agua	QF3
Iluminación	Interruptor magnetotérmico luz	QF4

Para poder escoger las protecciones y secciones de cable, calcularemos los consumos, y las caídas de tensión. Intentando que las caídas de tensión sean mínimas, ya que no disponemos de las caídas de tensión acumuladas.

Tabla 5. Caídas de tensión 230 V alterna.

Elemento	Potencia (W)	Intensidad (A)	Cos φ	Conductor de	Sección (mm²)	Longitud Línea (m)	Caída de Tensión (V)	Caída de Tensión %
Bomba de agua	370	1,89	0,85	cobre	1,5	5	0,2	0,1
Fuente alimentación	352	1,8	0,85	cobre	1,5	0,5	0,1	0
Iluminación	50	0,25	0,85	cobre	1,5	5	0	0
Sondas de nivel	2,4	0,012	0,85	cobre	1	0,5	0	0

En la siguiente tabla observamos, las protecciones idóneas:

Tabla 6. Especificación de protecciones de los circuitos 230 V alterna.

Interruptor	Fabricante	Características	Refencia
QF0	Schneider Electric	2P- 16A- C- PC10kA	24337
QD0	Schneider Electric	2P- 25A- 0,30mA	15249
QF1	Schneider Electric	2P- 2A- C- PC 10kA	24332
QF2	Schneider Electric	2P- 0,5A- C- PC 10kA	24068
QF3	Schneider Electric	3P- 1,6 a 2,5A	21106
QF4	Schneider Electric	2P- 0,5A- C- PC 10kA	24068

CAPÍTULO 10: PROTOTIPO

10.1. Elementos del prototipo

Primeramente, recordar que el montaje del prototipo del proyecto está realizado con materiales reutilizados. El coste de este material es considerable, y ha sido reutilizado o pedido prestado.

Es un prototipo para demostrar el potencial que podría tener el proyecto.

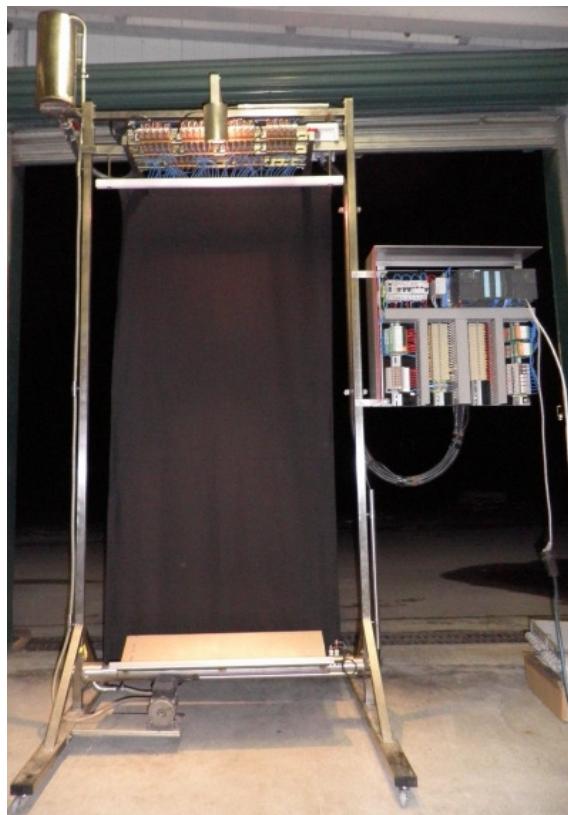


Figura 57. Vista general del prototipo.

10.1.1. Chasis

El chasis ha sido construido con perfil de acero inoxidable de 30 x 30mm. Y perfil 60 x 30. Se ha construido de tal manera que pueda ser desmontado para su transporte. A su vez también se le ha dotado de ruedas para poder moverse. Estas ruedas son capaces de soportar 20 kilos cada una. Por motivos económicos se les ha dotado de un freno construido con un tornillo.



Figura 58. Detalle de freno de rueda.

10.1.2. Electroválvulas

Las electroválvulas utilizadas son LUCIFER con referencia 131M7450. Estas electroválvulas son 3/2 de mando directo, pueden tener aplicaciones con gases y líquidos neutros. Van insertadas en un bloque colector de cinco electroválvulas de referencia 488860 05 con dos entradas de 1/8" y cinco salidas de 1/8".

Las bobinas empleadas son F 220V 50Hz, IP 65, 2W, ED100%. Con referencia 488980 3D.

En la siguiente imagen podemos observar las especificaciones.

General application valves 3/2 - Direct operated

Port size G	Orifice (mm)	Flow factors (L/min)		Admissible differential pressure bar		Fluid temp. °C			Seat disc	Reference numbers			OR	Power consumption (W) DC AC	Wt. (g)	El. Part Group	Dim ref.			
		Liquids kv	Gases Qmax On	Min DC	Max AC	Gas	Liquid	Oil		Global valve reference	Valve reference no.	Housing	Cell							
SB	1.5	0.9	2.4	70	0	7	7	75	75	75	FKM	-	131M74	8993	488980	2.5	2	125	1	26
	1.5	0.9	2.4	70	0	7	7	75	75	75	FKM	-	131M7450	1 8993	488980	2.5	2	125	1	26
Brass body/Sub-base mounting																				
Normally closed																				

Table continued on page 144

Notes:

* See Electrical Parts Group table at end of section

1. Manual override standard

Electrical parts options with 3/2 general application valves for dry or lubricated air, neutral gases and liquids

El.part Group	Coil	Protection class / Temperature class	Power		Coil Ref. No.	Connection	Housing Order No.	Housing Ref. No.	Ambient temp.	
			DC	AC					min.	max.
1	22 mm	IP 65	Class F	2.5 W	2 W	DA01	488980 for DIN plug	A0	8993	-40 50
		IP 65	Class F	2.5 W	2 W	DA02	481045 with DIN plug	A0	8993	-40 50
		IP 65	Class F	5 W	4 W	DA03	481180 for DIN plug	A0	8993	-40 50
		IP 65	Class F	5 W	4 W	DA04	481530 with DIN plug	A0	8993	-40 50
		EEx m II T4	5 W	4 W	VA01	482605 with 1500mm cable	00	-	-40	50
		IP 65	EEx m II T5	2.5 W	2 W	VA02	482606 with 1500mm cable	00	-	-40 50

Specification		Low power	High power	Standard UL / CSA*	Double frequency
Ref. (without plug)		488980 or DA01	481180 or DA03	492912 or DA05	483590 or DA07
Ref. (with plug)		481045 or DA02	481530 or DA04	492919 or DA06	
Degree of protection		IP65 according to IEC / EN 60529 standards (with plug connection)			
Classe of insulation		F 155°C	F 155°C	A 105°C for UL/CSA	F 155°C
Electrical connection		The coil is connected with a 2 P + E plug according to DIN 43650 type B			
Ambient temperature		-40°C to +50°C The application is limited also by the temperature range of the valve	-40°C to +50°C	-40°C to +50°C	-40°C to +50°C
Elect. Power	DC	Pn (hot)	2.5 W DC	5 W DC	4 W
		P (cold) 20°C	3 W	6.5 W	4.5 W
	AC	Pn (holding)	2 W	4 W	3 W
		Attraction cold	5.7 VA (2.5 W)	8.9 VA (5 W)	7.5 VA (4 W)

* This coil is UL/CSA accepted with corresponding approved valves only.

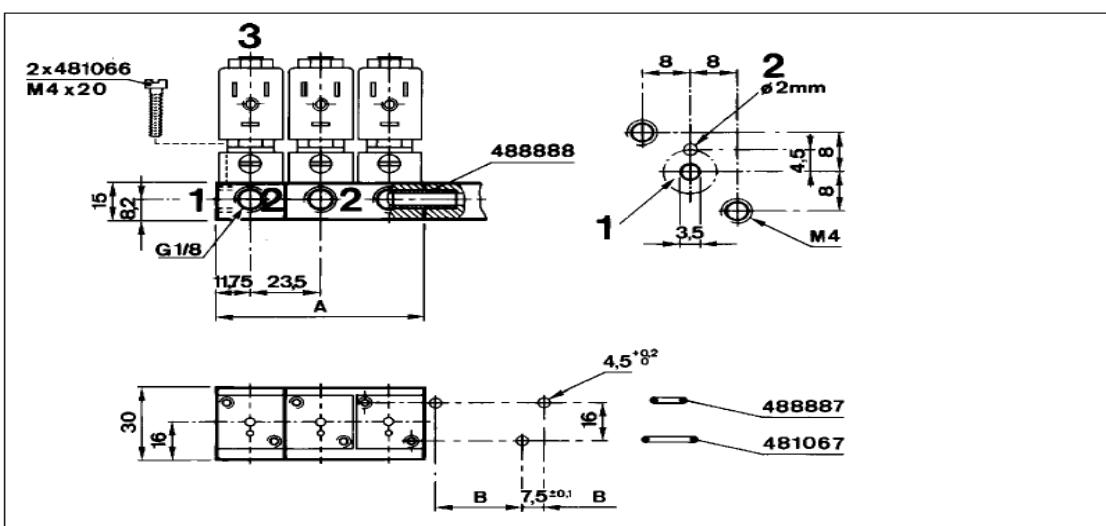
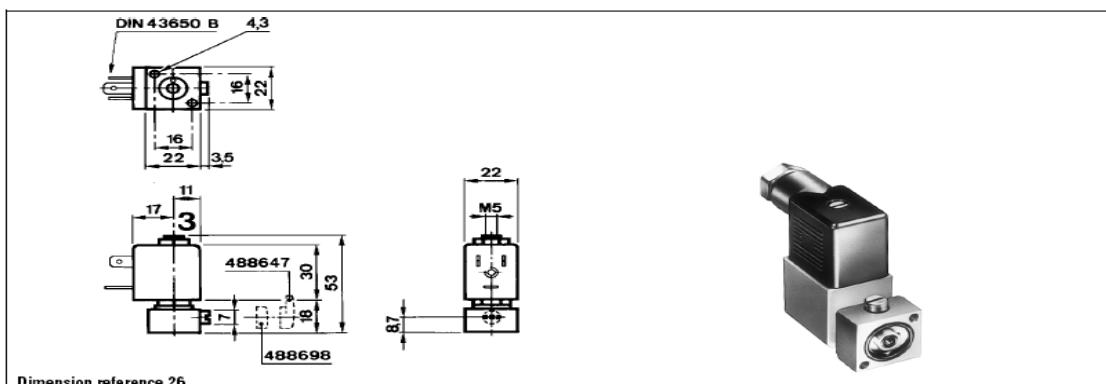


Figura 59. Especificaciones electroválvulas.

Según especificaciones del fabricante tienen una vida de 10 millones de ciclos pero al hacer conmutaciones con agua y a presión atmosférica la vida se alargará. El tiempo de respuesta está entre 10-15 ms

Para poder observar la gota en la salida de la electroválvula se ha tenido que modificar la salida del bloque, utilizando un tubo de neumática de 3 mm. Para ello utilizaremos un tapón de latón de 1/8" y lo taladraremos con una broca de 3mm.

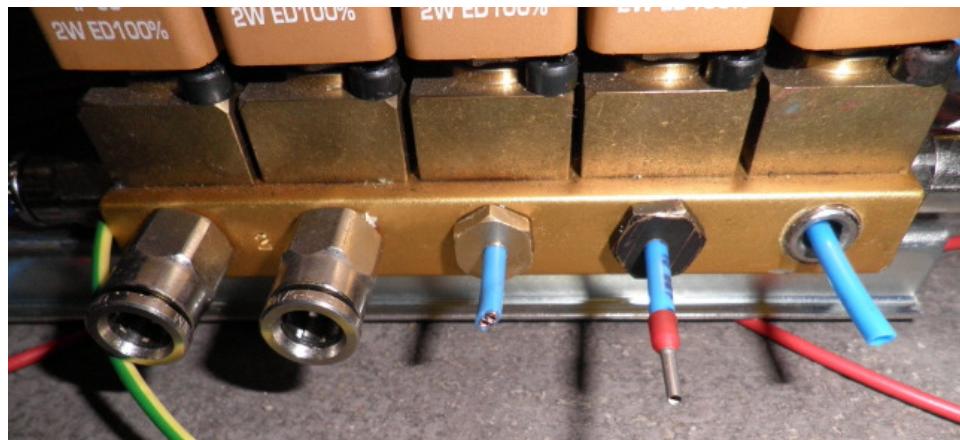


Figura 60. Detalle salida válvulas.

Como necesitamos electroválvulas 2/2 taparemos el orificio de escape, mediante un tornillo y teflón.

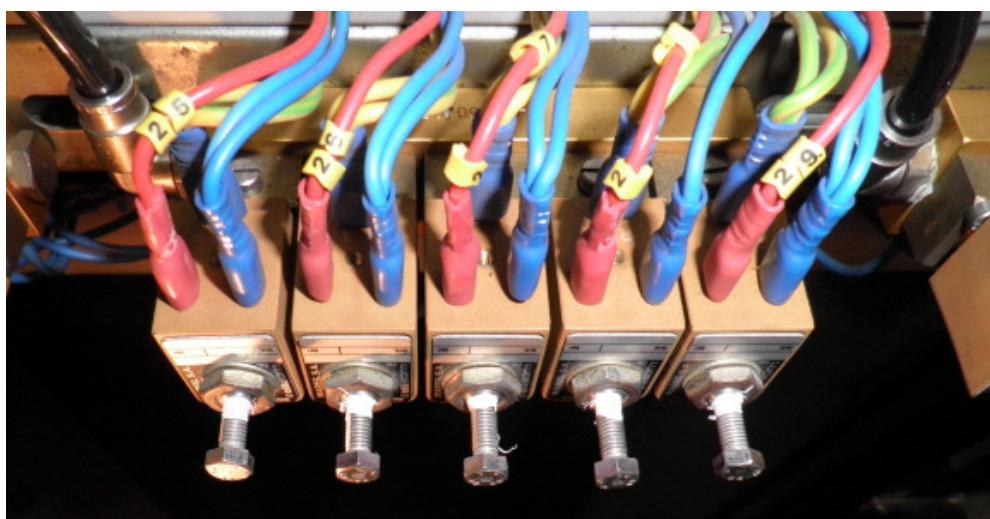


Figura 61. Detalle escape de las electroválvulas.

Como la salida del bloque de válvulas tiene una distancia de 23,5 mm guiaremos el agua mediante tubos de neumática de diámetro 3mm que guiaremos a un perfil de teflón.

Las conexiones eléctricas las realizamos mediante conectores faston y cable de 1mm.

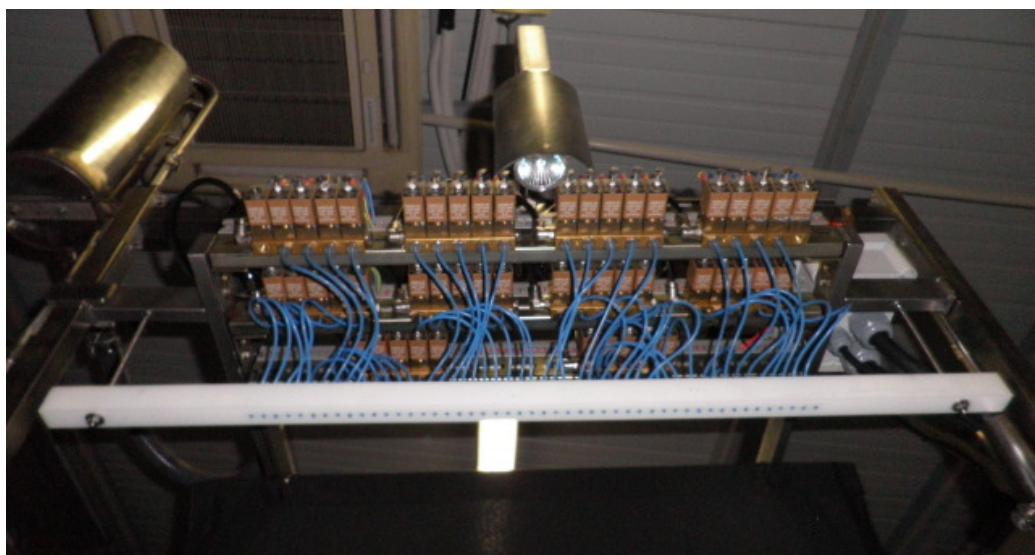


Figura 62. Guía de tubos de salida de agua.

10.1.3. Guía de tubos

Esta guía de teflón la hemos perforado como muestra la imagen para que los chorros de agua caigan verticalmente y a una distancia determinada del chorro contiguo.

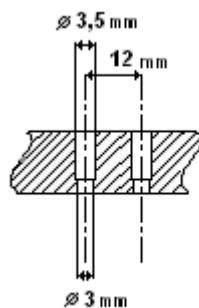


Figura 63. Distancias de la guía de tubos.

Para alimentar los bloques de válvulas utilizamos tubo neumático de 8mm. El cual se alimenta de un colector:

10.1.4. Colector

El colector se trata de un tubo rígido de PVC de 40 milímetros de diámetro. Se ha agujereado y roscado a $\frac{1}{4}$ ", para introducir el racor del tubo.

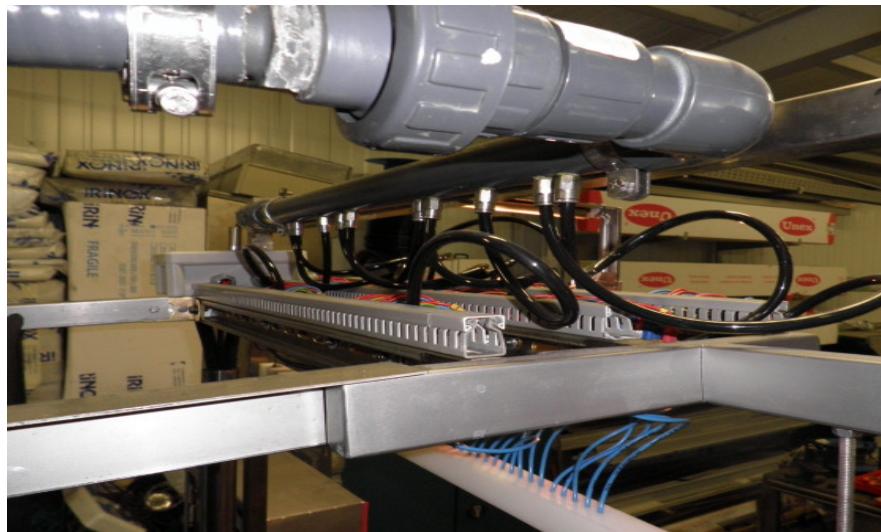


Figura 64. Vista del colector.

10.1.5. Depósito inferior

Para recoger el agua, se ha construido un depósito inferior. Este depósito ha sido construido con una tubería de acero inoxidable de 70 milímetros de diámetro y se ha cortado la parte superior, se le ha puesto unas alas para evitar salpicaduras.

La capacidad es de 2 litros aproximadamente.



Figura 65. Vista del depósito inferior

10.1.6. Bomba de agua

El depósito inferior dispone de un sensor de nivel el cual activa la bomba de agua cuando el nivel alcanza la varilla máxima y desactiva la bomba cuando deja de tocar el agua la varilla de mínima. La distancia entre estas dos varillas marcará el tiempo de funcionamiento de la bomba.



Figura 66. Sondas de nivel del depósito inferior.

El depósito inferior bombea agua al depósito superior mediante una bomba de agua. Se trata de una bomba centrífuga monofásica 230V alterna y con un consumo de 0,22kW, un caudal de 9 litros por minuto y una presión de 3,5 bar.

Al depósito inferior se le ha hecho una salida de agua como muestra la imagen para evitar remolinos y que la bomba se descebe .

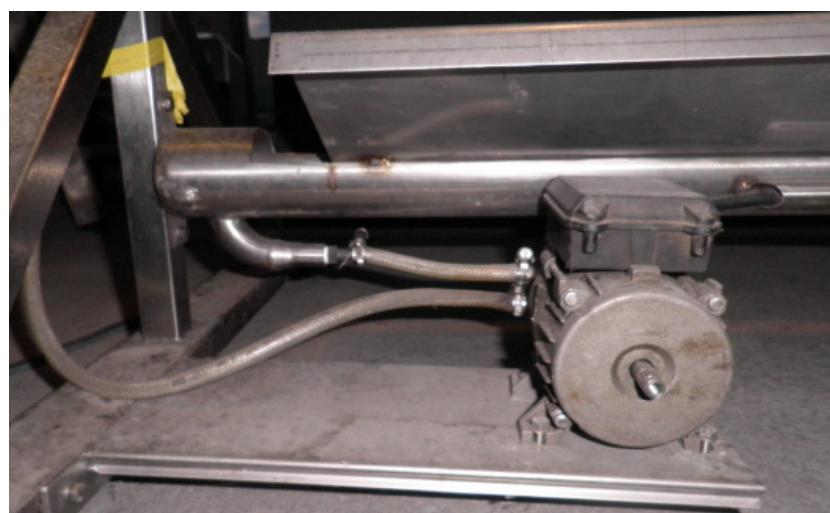


Figura 67. Salida de agua del depósito inferior.

Para bombear el agua al depósito superior se utiliza un manguera flexible, la cual desemboca en la parte superior del depósito.

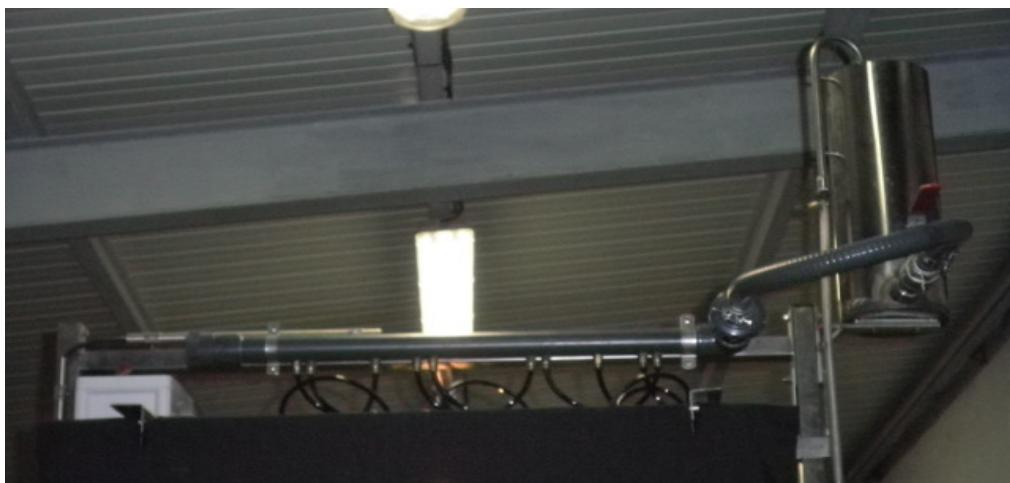


Figura 68. Vista del depósito superior.

10.1.7. Depósito superior

Para evitar que el agua retorne por su propio peso, ya que la bomba se queda encebada , el tubo de salida del agua no debe estar en contacto con el agua. En el depósito superior no hemos puesto sonda de nivel de seguridad ya que es un circuito cerrado y la capacidad del depósito es de 7 litros, y es mayor que la capacidad del depósito inferior. A la salida del depósito se ha colocado un grifo que conecta con el colector, por si hay algún problema poder cortar el suministro de agua.



Figura 69. Detalle del depósito superior.

10.1.8. Cuadro eléctrico

El cuadro eléctrico se ha construido sobre una placa donde se han insertado los componentes eléctricos:

En la parte superior izquierda podemos observar las protecciones eléctricas, cada circuito dispone de su protección. También podemos observar el control de nivel.

En la parte superior derecha podemos observar la CPU, junto con los módulos de salidas y comunicación.

En la parte inferior se han dispuesto los relés que activan las válvulas. Ya que las válvulas que disponíamos se activan a 230V alterna. Podríamos haber cambiado las bobinas pero el coste es de 20 euros cada una, así se ha optado por utilizar estos mini relés. Estos relés nos retardan la activación entre 5- 7,5 ms, pero no es crítico ya que en el prototipo hacemos conmutar las electroválvulas cada 40 ms.



Figura 70. Microrelé.

Dentro de las limitaciones, se ha intentado cumplir al máximo el Reglamento de Baja Tensión, como la IEC 60204 (Seguridad en Máquinas).

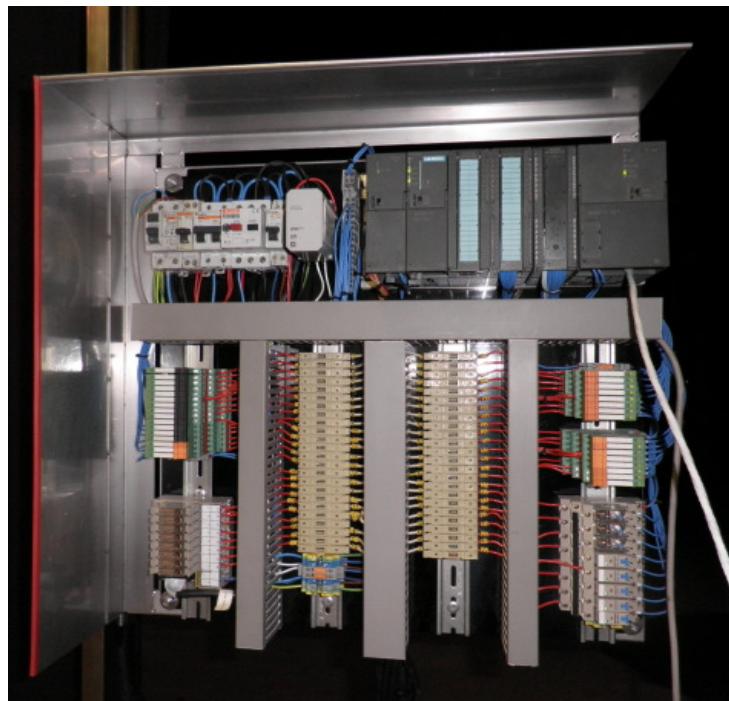


Figura 71. Cuadro eléctrico.

El cableado de tensión alterna de la maniobra lo hemos efectuado en color rojo, y el de tensión continua en azul. Las electroválvulas las hemos protegido mediante un fusible. El fabricante nos recomienda fusible de 32mA. El cable utilizado a la salida de los relés se trata de un multicable 24 x 1,5 que conectamos en una caja de empalmes en zona superior del prototipo.

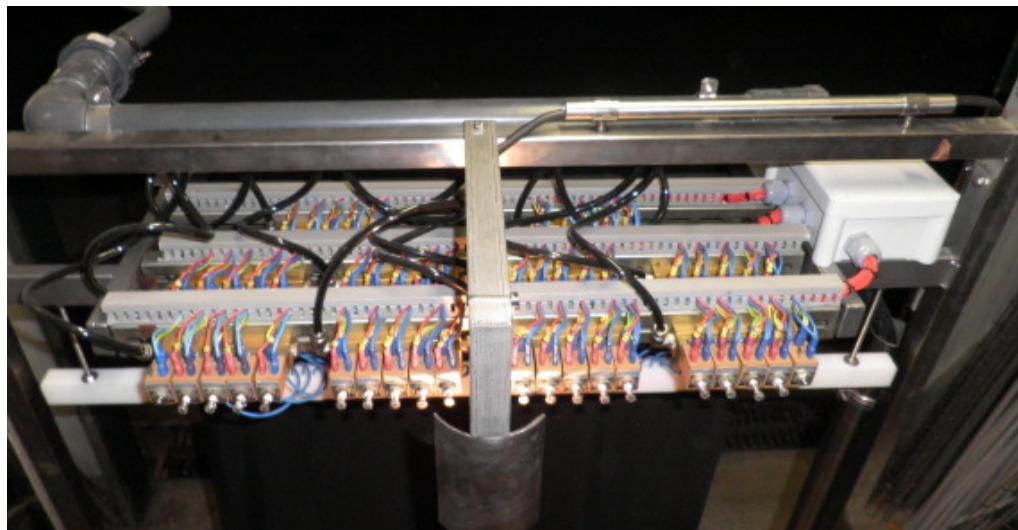


Figura 72. Conexión multicable con electroválvulas.

Por último, para que la imagen se vea mejor definida hemos colocado una cortina negra de fondo y una luz en la parte superior para que las gotas resalten.

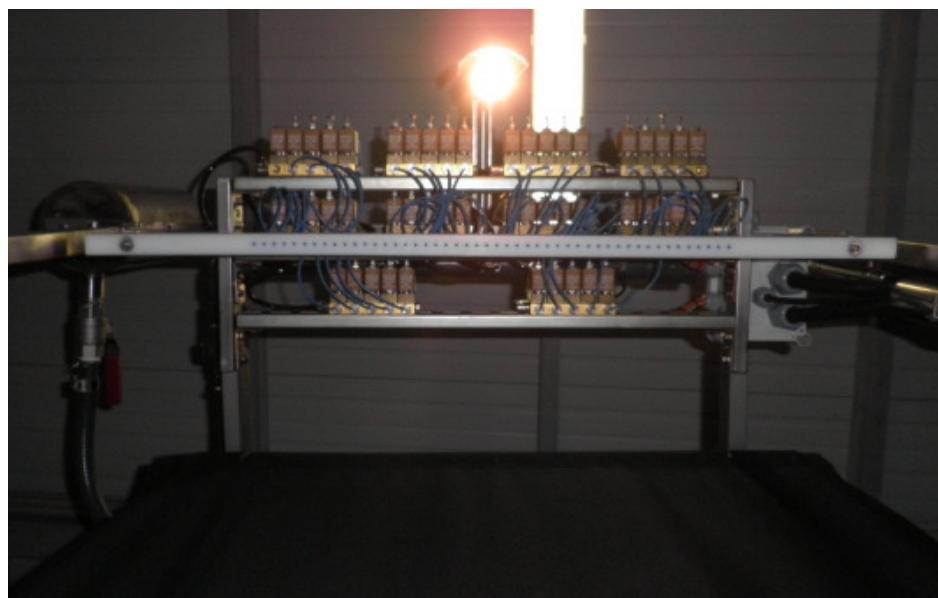


Figura 73. Cortina de agua vista desde abajo.

CAPÍTULO 11: CONCLUSIONES

En este documento se ha intentado mostrar el trabajo realizado durante varios cuatrimestres, en las asignaturas de PFC1 y PFC2.

El trabajo consta de las partes de diseño de una cortina de agua programable. Para ello se ha debido adquirir conocimientos en el tratamiento de imágenes, y programación en Matlab, tanto como en Step 7.

El trabajo se ha concluido con el montaje de un prototipo. Este prototipo es totalmente operativo. Sin embargo el prototipo tiene limitaciones causadas por los materiales utilizados.

Los materiales utilizados tienen un coste elevado, por lo que son reutilizados, así pues no se puede llegar a observar todo el potencial del software diseñado. Si se disminuye el tiempo de conmutación de las válvulas, la imagen acaba perdiendo definición. Esto se debe a los retardos acumulados, y al orificio de la electroválvula, que para tiempos ínfimos no da tiempo a crear la gota.

Utilizando las electroválvulas escogidas en la memoria, disminuiríamos los tiempos de retardo, y al ser el orificio mayor, la gota se crearía antes. No obstante podemos decir, que las imágenes que el prototipo efectúa están dentro de lo aceptable.

CAPÍTULO 12: POSIBLES MEJORAS

Este proyecto se podría haber mejorado en varios sentidos:

- Materiales prototipo.
- Software

Si hubiera habido capital para adquirir materiales mas convenientes se hubiera conseguido una cortina de agua mas definida. Si hubiera tenido mas electroválvulas se podría haber hecho imágenes en colores. Utilizando tres válvulas por cada píxel.

De disponer de más salidas digitales podríamos controlar la bomba y la luz mediante la CPU.

En cuanto al Software: se podría haber programado mas botones en la interface gráfica. Por ejemplo se podría haber creado un botón para grabar una secuencia de imágenes.

También se podría haber creado un diálogo entre la CPU y la aplicación, y no como ahora que la CPU solo está a la espera de la aplicación. Esto sería interesante a la hora de posibles errores en los envíos de las tramas, entonces la CPU nos podría avisar.

Se podría hacer un apartado de redimensionamiento de imágenes, es decir que el programa fuera capaz de redimensionar una imagen que es en un cierto número de píxeles mayor que electroválvulas disponemos.

CAPÍTULO 13: BIBLIOGRAFÍA

13.1. Referencias bibliográficas

Emilio González Rueda. Programación de Autómatas SIMATIC S7-300 (Lenguaje AWL). Ediciones CEYSA

Manual Matlab 7.0

M.C. José Jaime Esqueda Elizondo eds. 2002. Fundamento de Procesamiento de imágenes Universidad Autónoma de California.

[Http://mathworks.com](http://mathworks.com)

<http://www.automation.siemens.com>

<http://es.wikipedia.org>

13.2. Bibliografía de consulta

Aparatamiento Modular. Cofrets Modulares. Tomas Industriales. Baja Tensión Scheneider Electric 08

Catálogo Válvulas de proceso para fluidos SMC

<http://www.phoenixcontact.es>

<http://www.smces.es/>

