



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

# **PROYECTO DE AUTOMATIZACIÓN Y CONTROL MEDIANTE SCADA DE UN SISTEMA DE ALMACENAMIENTO Y RECUPERACIÓN INDUSTRIAL DE PIEZAS**

AUTOR: JOSE CABOT ALMELA  
TUTOR: JUAN PÉREZ CRUZ  
COTUTOR: MANUEL PINEDA SÁNCHEZ

**Curso Académico: 2014-15**

Como un famoso dicho anónimo bien dice, "para muchas y muy cambiantes tareas, es difícil remplazar al ser humano, quienes son fácilmente vueltos a entrenar dentro de un amplio rango de tareas, más aún, son producidos a bajo costo por personal sin entrenamiento". Para las demás, automaticemos.



## RESUMEN

En el proyecto “PROYECTO DE AUTOMATIZACIÓN Y CONTROL MEDIANTE SCADA DE UN SISTEMA DE ALMACENAMIENTO Y RECUPERACIÓN INDUSTRIAL DE PIEZAS” se diseña un sistema inteligente de distribución basado en la moderna tecnología de los PLC’s. También se realiza un SCADA con el que se controlan todos los procesos y se observa de manera instantánea el desarrollo del proceso que se lleva a cabo.

Para automatizar el sistema AS/RS se realiza una programación robusta y muy completa, que permite el uso manual y el automático.

En cuanto al uso manual, se permite el control tanto local, desde la propia máquina, como centralizado, desde el ordenador. En cuanto al uso automático, se programan cinco modos de funcionamiento distintos que abarcan todas las funciones que se esperan de un sistema AS/RS.

Además, se dota al sistema de una opción de apagado que finaliza los procesos llevando la máquina a una posición final.

Toda la programación se realiza de manera estructurada, y pese a la complejidad de algunas partes, se consigue que se puedan cambiar los parámetros de la automatización sin dificultad. Gracias a esto se puede aplicar este proyecto a diferentes máquinas que tengan que realizar las funciones propias de un sistema AS/RS.

El usuario deberá introducir las instrucciones necesarias para que el sistema ejecute los procesos deseados. Para la comunicación de dichas instrucciones se diseña un panel de control sencillo, intuitivo y completo. Este panel de control dispone una representación visual de la máquina que sirve para conocer el estado en el que se encuentra en cada momento y evaluar si funciona como se desea.

**Palabras Clave:** programación, automatización, SCADA, AS/RS, control, autómata



## CONTENIDOS

<b>MEMORIA</b> .....	4
1. Objetivos	
2. Presentación del proyecto	
3. Antecedentes	
4. Justificación	
5. Componentes	
6. Condicionantes a considerar	
7. Programación	
8. Control por tareas	
9. Panel de control	
10. Conclusión	
<b>PRESUPUESTO</b> .....	62
1. Introducción	
2. Costes materiales	
3. Coste mano de obra	
4. Costes generales	
5. Margen de beneficio	
6. Coste final del proyecto	
<b>PLIEGO DE CONDICIONES</b> .....	70
1. Introducción	
2. Condiciones de índole técnica	
3. Condiciones de índole facultativa	
4. Condiciones de índole económica	
5. Condiciones de índole legal	
<b>ANEXO I. PROGRAMAS</b> .....	78
<b>ANEXO II. ACCIONES MODO AUTOMÁTICO</b> .....	95
<b>ANEXO III. COMPONENTES</b> .....	104





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA



# MEMORIA

Proyecto de Automatización y Control mediante  
SCADA de un Sistema de Almacenamiento y  
Recuperación Industrial de Piezas



**JOSE CABOT ALMELA**

Tutor: Juan Pérez Cruz

Cotutor: Manuel Pineda Sánchez



## ÍNDICE MEMORIA

<b>1. OBJETIVOS .....</b>	<b>7</b>
<b>2. PRESENTACIÓN DEL PROYECTO .....</b>	<b>8</b>
<b>3. ANTECEDENTES .....</b>	<b>11</b>
<b>4. JUSTIFICACIÓN .....</b>	<b>12</b>
<b>5. CONDICIONANTES A CONSIDERAR .....</b>	<b>13</b>
5.1 Limitaciones de la Máquina Prototipo .....	13
5.2 Características del Movimiento de la Máquina.....	14
<b>6. PROGRAMACIÓN .....</b>	<b>15</b>
6.1 Funcionamiento global.....	16
6.2 Entradas y Salidas.....	17
6.3 Variables de Interés.....	18
6.4 Control de Posición .....	21
6.4.1 COORDENADA X .....	21
6.4.2 COORDENADA Y .....	22
6.4.3 COORDENADA Z.....	23
6.5 Zona de Seguridad.....	25
6.6 Parada de Emergencia.....	26
6.7 Estado Casillero .....	27
6.8 Modo Selección .....	28
6.9 Modos de Funcionamiento .....	31
6.9.1 AUTOMÁTICO .....	31
6.9.1.1 Modo 1 .....	31
6.9.1.2 Modo 2 .....	35
6.9.1.3 Modo 3 .....	37
6.9.1.4 Modo 4 .....	40
6.9.1.5 Modo 5 .....	43
6.9.2 MANUAL.....	48
6.9.2.1 Desde máquina.....	48
6.9.2.2 Desde PC.....	51
6.9.3 APAGAR .....	52
6.10 Detalles a Tener en Cuenta en la Programación .....	53



<b>7. CONTROL POR TAREAS.....</b>	<b>54</b>
<b>8. PANEL DE CONTROL .....</b>	<b>55</b>
8.1 Panel de Control Inicial.....	55
8.2 Automático .....	56
8.3 Manual.....	59
8.4 Apagar .....	60
<b>9. CONCLUSIÓN.....</b>	<b>61</b>



## 1. OBJETIVOS

El objetivo principal del proyecto será realizar un sistema inteligente de almacenamiento y recuperación. Se realizará la automatización utilizando la moderna tecnología de los PLC's, que permite un control instantáneo y eficiente del sistema.

El segundo objetivo será diseñar un control de los procesos haciendo uso de un SCADA. Se pretende transmitir las instrucciones necesarias al PLC para el manejo de los procesos de una manera sencilla y predictiva. Con este fin se diseñará un panel de control que permita seleccionar y utilizar los distintos modos de funcionamiento.

En el SCADA se diseñará también una representación gráfica de la máquina automatizada que nos permitirá saber en tiempo real el estado de ésta. Es una herramienta muy útil para la supervisión de los procesos a distancia.

En particular, este proyecto tratará de crear un sistema AS/RS automatizado que distribuirá piezas de una zona de entrega a una zona de depósito, entre la propia zona de depósito o desde la zona de depósito a la zona de entrega.

Se pretende diseñar dos modos principales de funcionamiento, el modo manual y el modo automático. Con esto se consigue que el sistema sea versátil y se pueda adaptar en todo momento a los requerimientos que exija cada situación.

- Modo manual: el usuario tendrá el control total del sistema. Dicho control se podrá hacer desde el propio sistema o desde un panel de control.
- Modo automático: el usuario escogerá entre distintos modos de funcionamiento, introducirá las especificaciones que desee, y el sistema ejecutará las acciones oportunas para que se realice el trabajo demandado.

Se crearán dos modos manuales, uno que permita el control desde el sistema y otro desde el panel de control. En cuanto al modo automático, se diseñarán distintos modos que dotarán al sistema de un gran abanico de posibilidades.

## 2. PRESENTACIÓN DEL PROYECTO

El trabajo final de grado “PROYECTO DE AUTOMATIZACIÓN Y CONTROL MEDIANTE SCADA DE UN SISTEMA DE ALMACENAMIENTO Y RECUPERACIÓN INDUSTRIAL DE PIEZAS” trata de mejorar un método anticuado de almacenaje de una industria.

Para llevar a cabo la modernización del proceso se va a implantar un sistema inteligente de AS/RS. La realización de un sistema de estas características es una de las tareas más comunes de modernización industrial de los últimos tiempos, pues agiliza los procesos dentro de la industria, ahorra espacio y mejora la precisión.

La automatización de sistemas de almacenamiento y recuperación, o AS/RS (Automated Storage and Retrieval System), se puede llevar a cabo, y de hecho se está llevando, en todo tipo de industria: desde grandes naves industriales, donde, por ejemplo, se encargue de distribuir piezas a diversos talleres; hasta en pequeños comercios como pueda ser una farmacia, donde su función sea guardar y entregar medicamentos.

En concreto, se trabaja sobre una máquina de tres ejes disponible en el departamento de Ingeniería Eléctrica. Pese a que la programación se podrá adaptar a cualquier tipo de máquina similar, las características específicas de dicha programación se ajustarán a las características de la máquina en cuestión. A lo largo del proyecto se hacen referencias a los elementos involucrados en el proyecto. Todos estos componentes, incluida la máquina y sus partes, se encuentran explicados en el Anexo 3.

La automatización se realiza mediante la tecnología de los PLC's. En el PLC se descarga la programación desde el ordenador a través de una conexión Ethernet. Además, durante el funcionamiento, es al PLC donde llega tanto la información del estado de la máquina (desde los sensores de ésta), como las instrucciones del operario a través del SCADA. Esta información es interpretada por la programación realizada que realiza una serie de instrucciones. Las instrucciones también son enviadas tanto a la máquina como al panel de control desde el PLC.

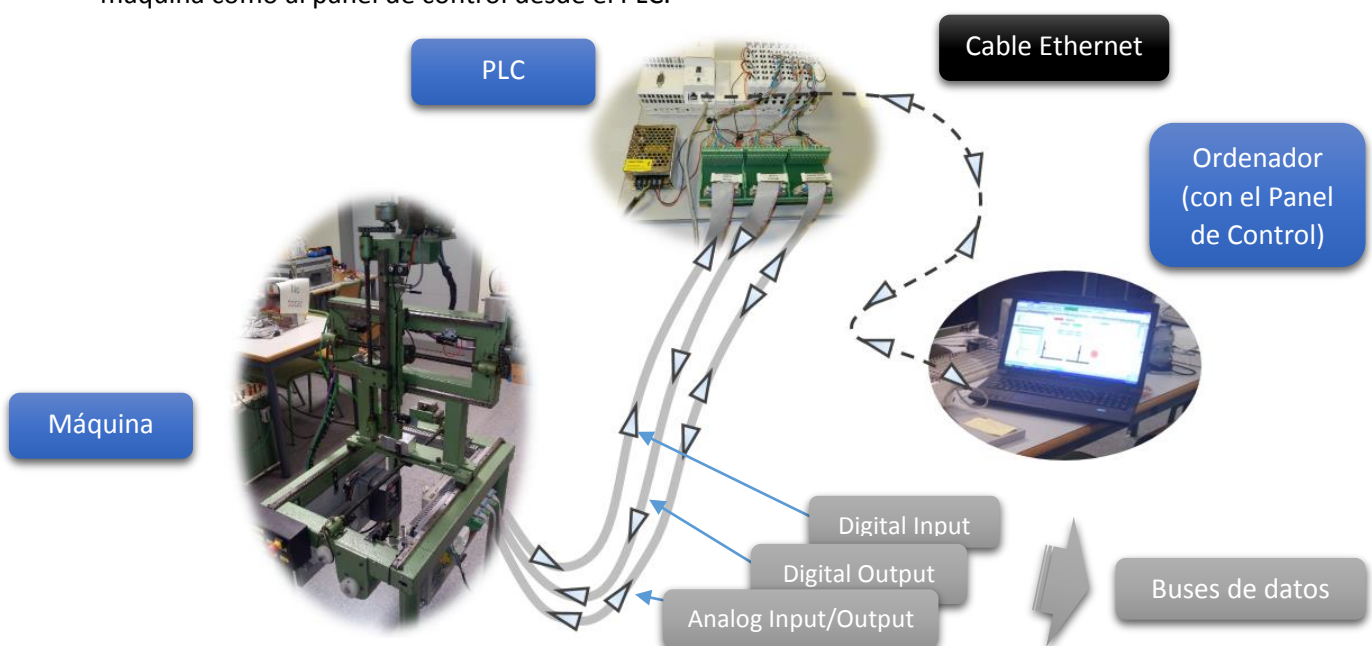
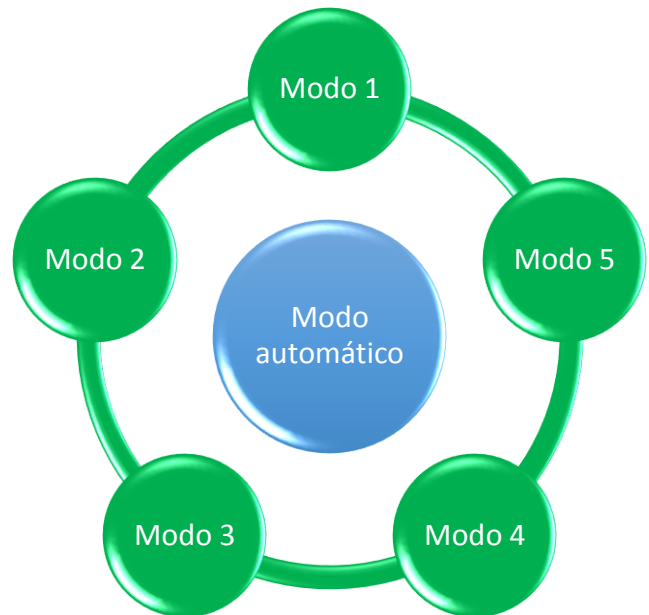
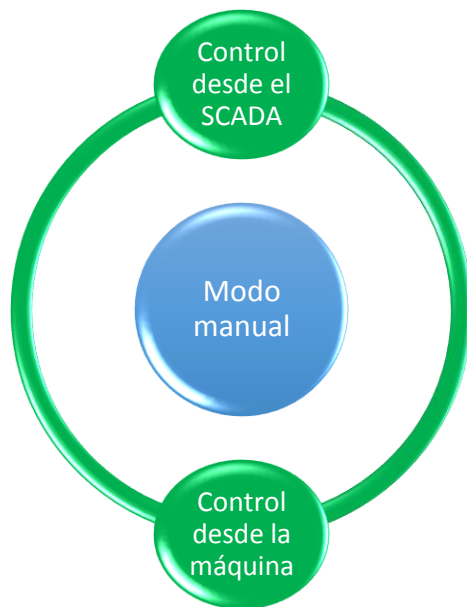


Figura 2.1  
Comunicación Máquina – PLC – Panel de Control



Se establecen dos métodos principales de funcionamiento, el manual y el automático. Además, dentro de cada uno de ellos, se diseñan distintos modos de funcionamiento. Se consigue una programación amplia, capaz de responder a todas las exigencias que se le puedan dar a un sistema de almacenamiento y recuperación. A continuación se describen y explican brevemente cada uno de los modos particulares de funcionamiento:

- Modo manual:
  - Desde la máquina: la máquina dispone de seis interruptores y dos pulsadores con los que se controlan los movimientos y la velocidad respectivamente. El objetivo será accionar los movimientos mediante los interruptores y aumentar y reducir la velocidad mediante los pulsadores.
  - Desde el PC: el control de cada eje y de la velocidad se realiza desde el panel.
- Modo automático:
  - Modo 1: el operario selecciona en el panel una casilla del depósito y, siempre que dicha casilla no esté ocupada, la máquina recogerá una pieza de la zona de recogida y la dejará en la casilla seleccionada.
  - Modo 2: similar al modo 1 pero de manera continua. Cuando la máquina trabaja en este modo va cogiendo piezas de la zona de entrega y las va colocando, por orden, en las celdas de depósito, siempre y cuando hayan huecos en los depósitos.
  - Modo 3: el operario selecciona una celda del depósito y una de la entrega y, siempre que la casilla del depósito esté ocupada y la de la entrega libre, la máquina trasladará la pieza de la primera celda a la de la entrega.
  - Modo 4: similar al modo 3 pero de manera continua. Si un operario selecciona este modo la máquina irá trasladando, siempre por orden, las piezas del depósito a la entrega. Esta tarea se realizará mientras haya piezas disponibles en los depósitos y entregas libres.
  - Modo 5: el operario selecciona dos celdas de depósito y la máquina trasladará la pieza de una a otra. La primera seleccionada será la celda de origen y la segunda la de destino. Para que se lleve a cabo la acción la celda de origen deberá contener una pieza y la de destino estar libre.



*Figuras 2.2 y 2.3  
Modos de funcionamiento*

Hay que tener en cuenta en todo momento que la máquina a la que se le aplica el proyecto es una máquina pequeña y limitada comparada con las que se utilizarían en la industria real. Aun así, como ya se ha dicho, la programación estructural sería válida para cualquier otra máquina con cierta similitud, adaptando para ello solamente partes muy concretas de la programación a sus características específicas.

### 3. ANTECEDENTES

#### *Automatización*

La automatización de procesos industriales empezó a hacerse hueco notablemente en los 60, y desde entonces ha adquirido una importancia crucial en la actividad industrial.

Frente al control humano de las máquinas, los procesos automatizados ofrecen mayor precisión y gran velocidad, lo que aumenta notablemente la eficiencia de los procesos. Y la búsqueda de la eficiencia es un gran objetivo de la ingeniería, pues es la clave para ser competitivo en el mundo empresarial.

Además, gracias a la automatización, cada vez son más procesos los que pueden ser realizados por máquinas sin la intervención humana. En las actividades que puedan resultar peligrosas para los humanos, la automatización disminuye o elimina el riesgo sobre estos.

#### *PLC's*

Los autómatas programables son un elemento clave en la automatización. Permiten el control de los procesos de manera rápida y eficiente. El avance de esta tecnología supone la posibilidad de automatizar procesos más complejos y más difíciles.

Por otro lado está la comunicación. Los PLC's permiten crear redes donde varios autómatas controlen partes de un mismo proceso, unos autómatas controlen a otros, o se realice la comunicación a distancia.

El campo de los autómatas programables es muy amplio y sigue desarrollándose.

#### *AS/RS*

Los sistema de almacenamiento y recuperación automatizado o AS/RS (Automated Storage and Retrieval System), son un conjunto de máquinas y autómatas programados para realizar las funciones de distribución de cargas en un almacén.

La inmensa mayoría de industrias utilizan, o podrían utilizar, uno de estos sistemas. Todas ellas necesitan almacenes donde tener organizados materiales, productos, o cualquier otro elemento utilizado en la industria.

La automatización de máquinas de almacenaje se ha convertido en algo habitual desde hace unos años atrás. Esto es porque permite un almacenaje y disposición rápidos, sin perder tiempo en buscar los productos o en coger una escalera; pero la mayor ventaja es el ahorro de espacio. En naves de gran altura, si no se tiene un sistema AS/RS, el almacén no podrá ocupar todo el espacio disponible, pues no es eficiente ni sensato hacer que un operario tenga que subir a esa altura. En cambio sí sería posible con el debido sistema automatizado.



## 4. JUSTIFICACIÓN

La justificación de este proyecto reside en la utilidad e importancia del trabajo en el campo de la automatización y el control de procesos industriales. Además el hecho de automatizar un proceso de un AS/RS le da al proyecto un amplio abanico de aplicación dentro de industrias muy distintas.

La automatización de procesos industriales es una actividad que se lleva a cabo en todo tipo de industrias y para un sinnúmero de procesos muy distintos. Hoy en día es imprescindible en la industria, y cada vez lo es más. La gran mayoría de industrias poseen máquinas controladas por autómatas para realizar su actividad, y cada una necesita una programación adaptada a sus características. Esto abre un amplio abanico de posibilidades para que un proyecto de este tipo sea aplicado.

Además de la automatización, es muy importante el control.

Se diseña un panel de control gráfico con la intención de que sea fácil de utilizar para cualquier usuario. Por muy complejos que sean los procesos que realiza la máquina, o los programas que regulan el funcionamiento, cualquier operario con un mínimo conocimiento del proceso será capaz de controlar la máquina a través de dicho panel. Esto agiliza el trabajo por una parte y reduce costes por otra, pues la formación necesaria del usuario será menor.

La justificación de automatizar un sistema AS/RS y no otro tipo de proceso reside en la versatilidad de campos en que éste puede ser aplicado. Además ofrece distintas posibilidades de programación, que ha permitido que se consigan distintos modos de funcionamiento para la máquina.

A nivel personal, este proyecto ofrece la posibilidad de estar capacitado para realizar una programación adecuada y adaptada a las distintas exigencias que pueda tener una máquina, y de diseñar un control de ésta que resulte sencillo e intuitivo. Dos aptitudes muy útiles para el mercado actual.

A lo largo de este proyecto se trabaja y mejora en este aspecto, lo que permite adquirir experiencia y agilidad en este campo.

## 5. CONDICIONANTES A CONSIDERAR

Como se ha comentado anteriormente, el trabajo se realiza sobre la máquina disponible en el departamento de Ingeniería Eléctrica. Esta máquina sirve para simular una máquina que se tendría en una industria, pero tiene varias características o carece de ciertos recursos que condicionan el trabajo. De todas formas, se les ha buscado una solución.

Si bien las distintas soluciones no serían las óptimas en un caso industrial real, dadas las circunstancias, funcionan de manera muy satisfactoria.

Se dividen en limitaciones de la máquina prototipo y características del movimiento de la máquina.

### 5.1 Limitaciones de la Máquina Prototipo

La falta de recursos afecta en gran medida a la programación y al control de la máquina. Serían necesarios más sensores para poder tener el control adecuado del funcionamiento de ésta. Todos los elementos de los que se dispone se presentan y explican en el Anexo III.

#### *Control de posición del eje z*

Disponer de otro potenciómetro sería realmente útil para controlar la posición en el eje z. El control de la posición en estas condiciones sería como el de los ejes x e y.

La solución adoptada ha sido calcular la posición en función del tiempo de marcha del motor en el eje z. Debido a la imprecisión a la hora de contabilizar el tiempo y, sobre todo, al efecto de la inercia, el error se acumula en la variable de posición z. No obstante, para minimizar el error, se lleva la horquilla a la posición más alta del eje z, donde se encuentra el sensor inductivo. Al llevar la horquilla hasta dicho sensor, la variable z se actualiza a su valor real y se elimina el error acumulado hasta el momento. Esta acción se lleva a cabo siempre que se inicializa la máquina y siempre que se cambia de un modo de funcionamiento a otro.

El programa encargado de controlar la posición en este eje es “variable\_posicion\_z”. Más adelante, en el apartado “6.4.3 Coordenada z” (pág. 23), se explicará con más detalle su funcionamiento.

A pesar de que la solución permite el correcto funcionamiento del proceso, el error sería mucho más preciso y la programación más eficiente si se dispusiera del sensor mencionado anteriormente.

#### *Control de estado de las celdas de depósito y entrega*

Saber si las celdas del depósito y de la entrega están ocupadas o vacías es un problema que se solventaría fácilmente con un sensor en cada celda que nos indicase si hay o no un objeto.

Con sensores fotoeléctricos o ultrasónicos se daría solución al problema fácilmente, y una vez más facilitaría la programación y el uso. En una industria se dispondría de ellos, pero para el proyecto no ha sido posible su instalación. Esto se debe a que el casillero que simula la zona de depósito y entrega se ha fabricado con recursos limitados y específicamente para el proyecto.

Las soluciones adoptadas han sido distintas para la zona de depósito y para la zona de entrega.

A las celdas de depósito solo llegan y salen las piezas a través de los distintos modos de funcionamiento programados. Por tanto no supone un gran problema establecer en la programación cuándo las celdas de depósito estarán ocupadas o no, teniendo en cuenta los procesos que se van realizando.

En cuanto a las celdas de entrega, el caso es distinto. No hay manera de saber cuándo una pieza se retira. La solución por la que se ha optado ha sido permitir cambiar el valor de la variable que marca el estado de las celdas de entrega desde el mismo panel de control.

Como se verá más adelante, en el panel de control se encuentra una representación del casillero con las celdas de depósito en azul y las de entrega en rojo. Las celdas que contienen una pieza están rellenas de su color correspondiente y las que están vacías solamente tienen en color el borde, mientras que el interior es blanco.

El estado de las celdas de entrega se puede cambiar simplemente pulsando sobre ellas. El operario deberá de actualizar el estado de las celdas de entrega cada vez que se retire una pieza. Esto no será necesario cuando una pieza se deposite, pues de esto se encarga la programación.

Como se aprecia las soluciones a las que se ha llegado son muy satisfactorias.

#### *Control de pieza en zona de entrega*

Cuando se recoge una pieza de la zona de entrega lo ideal sería saber si hay alguna pieza preparada para ser recogida. Para este problema serviría el mismo tipo de sensores mencionados anteriormente, fotoeléctricos o ultrasónicos, entre otros. Si este sensor detectara una pieza, es que estaría lista para ser recogida.

La solución empleada en este caso ha sido utilizar una variable llamada *pieza\_colocada*. Cuando la horquilla llega a la posición de recogida, aparece en el panel de control el botón PIEZA COLOCADA. Cuando el operario puede garantizar que la pieza está lista pulsa el botón y el proceso continúa.

Es necesaria la actuación de un operario pero no hay otra manera de asegurarse que hay una pieza lista para ser recogida.

### **5.2 Características del Movimiento de la Máquina**

Los posibles movimientos de las máquinas y sus características se pueden encontrar en el Anexo III. Como allí se explica, toda la superficie horizontal del área de trabajo es barrida al realizarse un desplazamiento sobre el eje x. Esto impide el establecimiento de una zona fija de recogida distinta de las zonas de depósito y entrega. Este problema se podría solucionar si la máquina permitiese el movimiento de rotación sobre el eje z de la horquilla.

La solución que se ha adoptado ha sido la de crear una plataforma móvil que se colocará manualmente cada vez que el proceso requiera la recogida de una pieza. Esta plataforma simularía lo que podría realizar en una industria real una cinta transportadora, otra máquina del mismo tipo o incluso un operario.

Se disponen de más detalles y fotografías en el Anexo 3.

## 6. PROGRAMACIÓN

Para la programación se han utilizado tres tipos de lenguajes. Texto estructurado (ST), diagrama de funciones (FBD) y graficets (SFC).

El lenguaje de programación de texto estructurado o, ST (Structured Text), es muy parecido al lenguaje c. Es muy cómodo y rápido de manejar para los programas y acciones que no requieren una ramificación muy amplia. Dispone de estructuras como "IF () THEN ELSE", "WHILE () DO", "FOR () TO", etc., y de operaciones booleanas como "OR", "AND"...

El lenguaje de diagrama de funciones, o FBD (Function Block Diagram), permite la programación mediante bloques de funciones. Funciona asemejándose a un circuito eléctrico. Los bloques de funciones se pueden representar mediante funciones lógicas. Este tipo de lenguaje es muy útil si se desea emplear bloques ya programados, como puedan ser contadores, temporizadores, generadores de pulsos, generadores de flancos de subida, etc...

El lenguaje de diagrama de graficets, o SFC (Section Function Chart) es un lenguaje gráfico, que funciona a base de ramificaciones. Se divide en etapas, donde se realizan acciones que pueden ser de entrada, continuas mientras se esté en la etapa o de salida. Además, según el programa informático con el que se realiza la programación, también se puede retrasar el comienzo de una acción en una etapa, retrasar la finalización una vez se ha salido de ésta, etc. Entre etapa y etapa hay transiciones que deben cumplirse para que el paso de una a otra se cumpla.

El lenguaje graficet permite programar, tanto las acciones como las transiciones, con otro tipo de lenguaje. La flexibilidad de este tipo de lenguaje le permite ser realmente útil y eficaz en la programación de tareas con gran número de condiciones, en tareas que rigen procesos de varios subprocesos y en tareas de control.

### 6.1 Funcionamiento global

La automatización del sistema es posible gracias al establecimiento de una jerarquía dentro de los programas. Hay programas de seguridad (ROJO), programas de control (AMARILLO/AZUL), programas de ejecución (VERDE) y programas auxiliares (NARANJA).

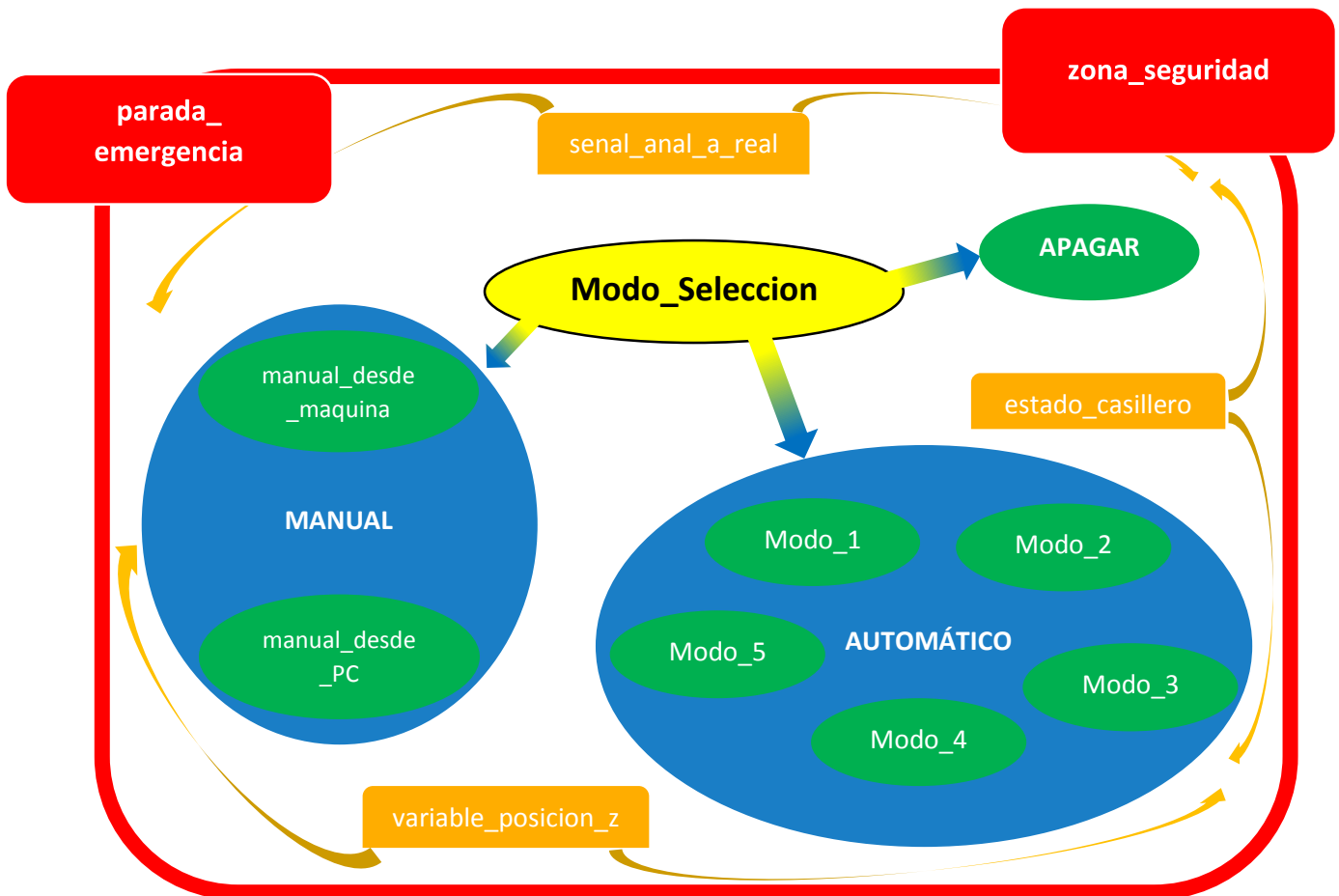


Figura 6.1  
Modos de funcionamiento

- Programas de SEGURIDAD
- Programas de CONTROL
- Programas de EJECUCIÓN
- Programas AUXILIARES

## 6.2 Entradas y Salidas

Para realizar el control de los procesos es necesario recibir información de la máquina y enviar instrucciones. Todo ello se realiza a través de las señales de entrada y salida.

A continuación se presentan las señales disponibles, la dirección de éstas y las variables que almacenan el valor de cada una.

### Entradas digitales

SENSOR/ACCIONADOR	VARIABLE	DIRECCIÓN
PULSADOR SPEED UP	<i>speed_up</i>	%IX6.0
INTERRUPTOR "X RIGHT"	<i>x_adelante</i>	%IX6.1
PULSADOR SPEED DOWN	<i>speed_down</i>	%IX6.2
INTERRUPTOR "Y RIGHT"	<i>y_derecha</i>	%IX6.3
INTERRUPTOR "X LEFT"	<i>x_atras</i>	%IX6.4
INTERRUPTOR "Z DOWN"	<i>z_abajo</i>	%IX6.5
INTERRUPTOR "Y LEFT"	<i>y_izquierda</i>	%IX6.6
ZERO_X (sensor inductivo eje x)	<i>cero_x</i>	%IX6.7
INTERRUPTOR "Z UP"	<i>z_arriba</i>	%IX7.0
ZERO_Z (sensor inductivo eje z)	<i>cero_z</i>	%IX7.1
ZERO_Y (sensor inductivo eje y)	<i>cero_y</i>	%IX7.2

### Salidas digitales

SENSOR/ACCIONADOR	VARIABLE	DIRECCIÓN
CONTACTOR_X	<i>eje_x</i>	%QX2.0
CONTACTOR_Y	<i>eje_y</i>	%QX2.1
CONTACTOR_Z	<i>eje_z</i>	%QX2.2
GIRO HORARIO (acercar sensores)	<i>adelante</i>	%QX2.3
GIRO ANTIHORARIO (alejar sensores)	<i>atras</i>	%QX 2.4
2ª señal digital del variador de frecuencia	<i>vel_2</i>	%QX 2.5
1ª señal	<i>vel_1</i>	%QX 2.6

### Entradas analógicas

SENSOR/ACCIONADOR	VARIABLE	DIRECCIÓN
POTENCIÓMETRO_Y	<i>y_anal</i>	%IW7
POTENCIÓMETRO_X	<i>x_anal</i>	%IW8

### 6.3 Variables de Interés

Para la programación del funcionamiento de la máquina es necesaria la utilización de muchas más variables además de las asociadas a las señales de entrada y salida. La mayoría de ellas se declaran a nivel global, es decir, toda la programación interactúa con ellas. Estas variables se declaran en una lista de variables globales de nombre "GVL". Otras son auxiliares de cada programa concreto y se declaran dentro del propio programa.

A continuación se presentan las más importantes, pues se mencionan con frecuencia en las explicaciones de los distintos programas que vienen seguidas a este punto.

#### Selección de celda.

Estas variables son de tipo booleano. Se activan y se desactivan desde el panel de control como se explica en el apartado "8. Panel de control" (pág. 55). Sirven para trasladar las instrucciones del usuario al programa. Estas variables se activarán cuando el usuario seleccione la celda correspondiente.

```
(*Celdas ocupadas/libres*)  
obj_E01:BOOL;  
obj_E02:BOOL;  
obj_E03:BOOL;  
obj_E04:BOOL;  
obj_E05:BOOL;  
obj_D11:BOOL;  
obj_D12:BOOL;  
obj_D13:BOOL;  
obj_D14:BOOL;  
obj_D15:BOOL;  
obj_D21:BOOL;  
obj_D22:BOOL;  
obj_D23:BOOL;  
obj_D24:BOOL;  
obj_D25:BOOL;  
obj_D31:BOOL;  
obj_D32:BOOL;  
obj_D33:BOOL;  
obj_D34:BOOL;  
obj_D35:BOOL;
```

Figura 6.3

Extracto lista global de variables GVL

```
(*Variables de selección de celdas*)  
E01:BOOL;  
E02:BOOL;  
E03:BOOL;  
E04:BOOL;  
E05:BOOL;  
D11:BOOL;  
D12:BOOL;  
D13:BOOL;  
D14:BOOL;  
D15:BOOL;  
D21:BOOL;  
D22:BOOL;  
D23:BOOL;  
D24:BOOL;  
D25:BOOL;  
D31:BOOL;  
D32:BOOL;  
D33:BOOL;  
D34:BOOL;  
D35:BOOL;
```

Figura 6.2

Extracto lista global de variables GVL

#### Estado de las celdas

Estas variables, también de tipo booleano, sirven para conocer el estado de las celdas de depósito y de entrega. Saber si en una celda hay una pieza o si está vacía es de gran importancia en la programación.

### Posición real en los ejes

Almacenan las coordenadas reales de la máquina. Son de tipo real.

```
(*Variables posición*)  
x:REAL;  
y:REAL;  
z:REAL;
```

Figura 6.4

Extracto lista global de variables GVL

### Posiciones de los procesos

Almacenan las coordenadas conocidas de la zona de recogida, depósitos y entregas y la posición final. Estas variables son de tipo entero.

También se establecen las posiciones características por filas y columnas de las zonas de depósito y entrega.

El hecho de utilizar todas estas variables e inicializarlas desde una lista de variables globales permite modificarlas cuando se desee con rapidez y sencillez.

```
(*Variables posición de recogida*)  
xR:INT:=215;  
x_eng:INT:=165;  
yR:INT:=135;  
zR:INT:=10;  
  
(*Variables posición de depósito/entrega*)  
xD:INT:=0;  
xD_:INT:=50;  
yDi:INT;  
zDi:INT;  
zDi_:INT;  
yEi:INT;  
zE:INT:=30;  
zE_:INT:=10;
```

Figura 6.5

Extracto lista global de variables GVL

```
(*Variables y posición de depósito según columnas*)  
yDi_c1:INT:=5; (*0*)  
yDi_c2:INT:=69; (*67.5*)  
yDi_c3:INT:=135; (*135*)  
yDi_c4:INT:=201; (*202.5*)  
yDi_c5:INT:=265; (*270*)  
  
(*Variables z posición de depósito según filas*)  
zDi_f1:INT:=85; (*75*)  
zDi_f2:INT:=160; (*150*)  
zDi_f3:INT:=235; (*225*)
```

Figura 6.6

Extracto lista global de variables GVL

```
(*Variables posición final*)  
xF:INT:=300;  
yF:INT:=0;
```

Figura 6.7

Extracto lista global de variables GVL

### Contadores

Estas variables enteras se utilizan en los contadores de los modos 2 y 4.

```
(*Contadores*)  
Dii_c:INT;  
Dini_c:INT:=1;  
Dfin_c:INT:=15;  
Eii_c:INT;  
Eini_c:INT:=1;  
Efin_c:INT:=5;
```

Figura 6.8

Extracto lista global de variables GVL



## VARIABLES DE CONTROL

Las variables *automático/manual/apagar*, la de los distintos modos: *modo1, modo2...* y las variables *go* y *pieza\_colocada* se utilizan, entre otras, para elegir y controlar desde el panel de control los procesos que se llevan a cabo.

```
(*Automático/Manual/Apagar Scada*)
automatico:BOOL;
manual:BOOL;
apagar:BOOL;

(*Variables selección de modo*)
modo1:BOOL;
modo2:BOOL;
modo3:BOOL;
modo4:BOOL;
modo5:BOOL;

(*GO*)
go:BOOL;

(*Pieza colocada*)
pieza_colocada:BOOL;
```

Figura 6.9

Extracto lista global de variables GVL

## ESTADO DE DEPÓSITO Y ENTREGA

Estas variables booleanas valen 1 o 0 en función del estado del depósito y de la entrega. Sirven en los distintos programas para saber si se va a poder realizar la función o no.

```
(*Variables de estado del casillero*)
deposito_lleno:BOOL;
deposito_vacio:BOOL;
entrega_llena:BOOL;
```

Figura 6.10

Extracto lista global de variables GVL

## PROCESO TERMINADO

Es una variable muy importante que se desactiva cuando empieza algún proceso y se vuelve a activar cuando termina. Sirve para impedir que se ejecute una acción de un modo de funcionamiento si hay otro modo de funcionamiento en marcha.

```
(*Proceso terminado*)
proceso_terminado:BOOL:=1;
```

Figura 6.11

Extracto lista global de variables GVL

## PARO

Esta variable activa el programa de seguridad “parada\_emergencia”.

Se activa a través del panel de control, pulsando el botón STOP.

```
(*Paro*)
paro:BOOL;
```

Figura 6.12

Extracto lista global de variables GVL

## 6.4 Control de Posición

Como ya se ha mencionado con anterioridad, para el control de la posición de la horquilla, la máquina dispone de dos potenciómetros. Estos potenciómetros proporcionan una señal analógica en función de la posición sobre los ejes x e y. Para obtener la posición real respecto del sistema de referencia que establecido se ha de realizar una conversión. En cuanto al control de la variable z se utiliza un control mediante el tiempo.

### 6.4.1 COORDENADA X

El potenciómetro del que se dispone en el eje x produce una señal analógica en función de la posición real de la siguiente manera:



Figura 6.13  
Recta de conversión señal analógica a posición real eje x

Para obtener la coordenada x se emplea la siguiente conversión:

$$x = 374.06 - 0.0251 * xanal$$

Ecuación 1

Siendo *xanal* la variable de tipo entera que almacena la salida analógica del potenciómetro del eje x. La variable x es de tipo real y contiene el valor de la coordenada.

#### 6.4.2 COORDENADA Y

En el caso del eje y el procedimiento es el mismo que en el eje x. La posición también es controlada a través de un potenciómetro que produce una señal analógica en función de la posición real. En este caso la señal analógica respecto de la real cumple la siguiente recta:



Figura 6.14  
Recta de conversión señal analógica a posición real eje y

Para obtener la coordenada y se emplea la siguiente conversión:

$$y = 413.83 - 0.0191 * yanal$$

Ecuación 2

Siendo *yanal* la variable de tipo entera que almacena la salida analógica del potenciómetro del eje y. La variable *y* es de tipo real y contiene el valor de la coordenada.

El programa que realiza las conversiones de las señales analógicas a reales de las posiciones en los ejes x e y es "senal\_anal\_a\_real" y está escrito en el lenguaje de texto estructurado.

El programa se encuentra en el Anexo I.

### 6.4.3 COORDENADA Z

En cuanto a la posición sobre el eje z el asunto es más complicado. Sobre este eje no se dispone de ningún sensor que permita controlar la posición. Para solucionarlo, se ha optado por controlar el tiempo de marcha de la horquilla sobre el eje z, y en función de éste estimar la posición real.

La primera impresión es que parece sencillo de programar, refrescando el valor de la variable z con igualdades tipo:

$$z = z + A * t$$

*Ecuación 3*

$$z = z - A * t$$

*Ecuación 4*

La primera sería para cuando la horquilla está subiendo y la segunda para cuando está bajando, siendo A un factor que relacione el tiempo de marcha con el desplazamiento y t el tiempo de marcha de la horquilla.

Pero con este método solo se obtienen resultados válidos si se refresca la variable una vez haya acabado el movimiento, lo que no permite un control instantáneo de la variable de posición.

La mejor solución que se puede dar a este problema es restar los tiempos de subida y de bajada. Para realizar la programación apropiada a dicha tarea, se emplea el lenguaje de diagramas de contactores.

Se utiliza un contador CTUP, "counter up down". De esta manera el tiempo se suma cuando la horquilla se mueve hacia arriba y cuando se mueve hacia abajo el tiempo se resta. La variable t es de tipo real e indica el tiempo de subida menos el de bajada. Cuando vale 0, significa que la horquilla está en la posición z=0. A partir de ahí conociendo la relación entre tiempo y desplazamiento en el eje z se puede calcular, en función de t, la posición deseada.

$$z = 4 * t$$

*Ecuación 5*

El problema reside en que el módulo CUP solamente suma o resta si recibe un flanco de subida, por lo tanto, se coloca un módulo BLINK, generador de pulsos, previo a la entrada "up" y otro delante de la entrada "down". Estos módulos generarán pulsos cuando las variables booleanas que implican que la horquilla se mueve en el eje z estén activas. Si se mueve hacia arriba los generará el módulo previo a la entrada "up", y si se mueve hacia abajo los generará el previo a la entrada "down".

Estos módulos permiten el control del tiempo con el que se generan los pulsos. Se puede establecer la duración tanto del flanco de subida como del de bajada. De esta manera, al conocer el tiempo con el que se generan dichos pulsos, para calcular el tiempo que la horquilla está subiendo menos el tiempo que está bajando, tan solo habrá que multiplicar el tiempo de un ciclo generado en el módulo BLINK por el resultado del contador.

Como se ha explicado anteriormente, al conocer dicho tiempo ya se puede obtener el valor de la posición z, que se guarda en la variable real z.



Cabe destacar que la precisión de dicho método no es alta. Influyen en el error producido distintos factores. El error producido al contabilizar el tiempo y sobre todo el error producido por la inercia de la horquilla se acumulan conforme más tiempo se lleva usando la máquina.

El problema principal es el de la inercia, pues una vez desactivadas las variables de marcha la horquilla aún realiza un pequeño desplazamiento. En los ejes x e y esto no supone un problema, pues la posición simplemente varía ligeramente y es un error aceptable y no acumulativo. Pero en el eje z dicho error no se puede corregir, pues no hay manera de saber el valor del desplazamiento extra. Además no es un error que se produzca de la misma manera cuando el movimiento es de subida que cuando es de bajada, pues el efecto de la inercia es notablemente superior en la bajada.

La solución empleada para minimizar este error es actualizar el valor de la variable z a través del sensor inductivo. Cada vez que la horquilla llega a la posición de éste se actualiza su valor y se elimina el error acumulado hasta el momento.

Es por esto que en el programa que controla los distintos modos viene incluido un desplazamiento de la horquilla hasta la posición de dicho sensor al conectar la máquina y cada vez que se cambia de modo de funcionamiento. Este programa es el "Modo\_Selección\_Manual". Se explica con detalle en el apartado "6.8 Modo Selección" (pág. 28).

La precisión no es la deseada, pero se está hablando de una solución a un problema de recursos que en cualquier industria no existiría, pues la manera óptima de resolverlo sería la instalación de un sensor extra que permitiese un control preciso.

Otro inconveniente es que siempre se ha de utilizar el eje z a la misma velocidad. Pues de otro modo no sería posible saber la posición

El programa que realiza dicha conversión es "variable\_posicion\_z" y se encuentra en el Anexo I.

### 6.5 Zona de Seguridad

El área de trabajo de la máquina está delimitado por los interruptores de final de carrera de cada eje. De todos modos, utilizar estos como recursos para delimitar el uso de la máquina no sería apropiado, pues son mecanismos de seguridad que cortan la alimentación del motor de su eje. Por lo tanto, si la parte móvil llega a accionar uno de esos interruptores, ya no es posible su vuelta atrás, pues al motor no le llega corriente.

Es por ello que se define una zona de seguridad, de menor longitud en cada eje que la distancia que separa los finales de carrera.

La zona de seguridad delimita pues la zona de trabajo en la que puede trabajar la máquina. Se ha diseñado con unas medidas de (305, 270, 255) mm.

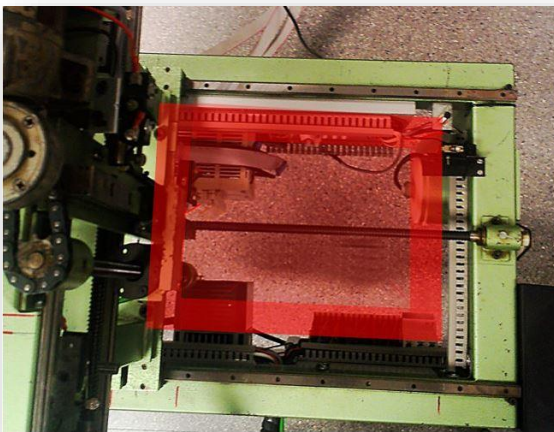


Figura 6.15  
Zona de seguridad y zona lenta plano XY

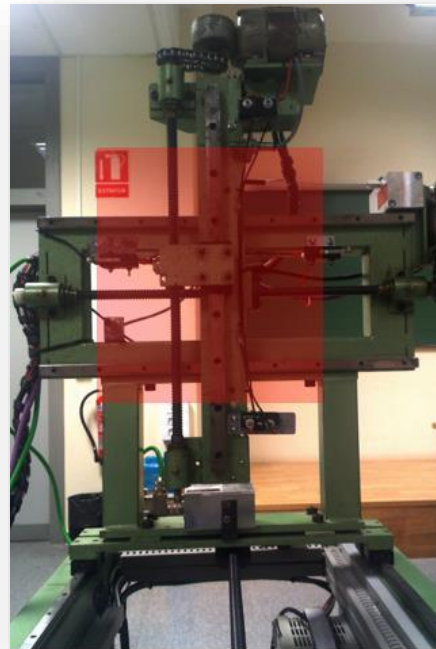


Figura 6.16  
Zona de seguridad plano YZ

En las figuras se puede apreciar la zona de seguridad, correspondiente a toda la superficie coloreada de rojo.

Además, en la figura 6.15 correspondiente al plano XY se diferencian dos tonalidades distintas. La zona roja más oscura representa la zona de “velocidad lenta”, dentro de la zona de seguridad.

Siempre que la máquina se esté moviendo hacia el exterior de la zona de seguridad, al llegar a esta zona, la velocidad, fuera cual fuera, pasará a ser la más lenta posible.

La zona de velocidad lenta se sitúa sobre la zona de seguridad, rodeándola como se aprecia en la figura. Su espesor es de 30 mm.

Por ejemplo, si se acciona la máquina de manera que se mueva a toda velocidad a lo largo del eje x, en sentido negativo, cuando la horquilla llegue a  $x=30$  mm la velocidad se reducirá a la mínima.

Cabe remarcar que la programación está hecha de manera que, aunque la parte móvil esté dentro de la zona de velocidad lenta, si el movimiento es hacia el interior de la zona de seguridad sí se podrá emplear cualquier velocidad distinta a la mínima.

La existencia de esta zona es necesaria. Esto es porque aunque se pare el motor el efecto de la inercia es mayor a mayor velocidad. Aunque se parara el motor en  $x=0$ , la horquilla se desplazaría hasta una posición demasiado alejada de la deseada si no fuese por esta zona de velocidad lenta.

Se puede observar que en el eje z no existe esta zona. Esto es porque en este eje la velocidad siempre es lenta. Como se vio en el apartado de “6.4.3 Coordenada z”, es necesario que esto sea así.

El programa “zona\_seguridad” se encarga tanto del establecimiento de la zona de seguridad, como de la zona de velocidad lenta como de que la velocidad en el eje z siempre sea la más lenta.

Está escrito en lenguaje de texto estructurado y se puede encontrar en el Anexo I.

## 6.6 Parada de Emergencia

La máquina dispone de una seta de emergencia que al pulsarla corta la alimentación de los motores. Si de verdad existe un riesgo considerable en el desarrollo de un proceso se ha de hacer uso de este mecanismo de seguridad.

En caso de que el peligro no sea muy elevado, se ha diseñado un programa que servirá para el mismo fin, pero sin ser tan drástico.

Este programa se llama “parada\_emergencia”, está escrito en lenguaje estructurado y es un muy sencillo. Se puede encontrar en el Anexo I.

El panel de control dispone de un botón de STOP que al accionarlo para todos los procesos que se estén llevando a cabo, y cuando se vuelve a pulsar para desactivarlo, el proceso sigue funcionando tal y como lo estaba haciendo originalmente.

Esto se hace posible de una manera muy sencilla. El programa en cuestión no interfiere en ningún otro programa de funcionamiento, de hecho, actúa directamente sobre las variables booleanas que activan los movimientos en cada eje.

Antes de ser accionado el botón, el estado de dichas variables se guarda en unas variables booleanas auxiliares. Cuando se pulsa el botón de STOP en el panel de control, las variables que activan el movimiento se desactivan, deteniéndose toda la máquina. Al volver a pulsarlo, se reestablecen dichas variables con los valores que se habían guardado en las variables auxiliares.



Figura 6.17  
Botón STOP

## 6.7 Estado Casillero

Es muy importante para algunos modos de funcionamiento saber si en el depósito ya no caben más piezas o si está vacío, y si la entrega está completamente ocupada o aún caben piezas.

En los modos 1 y 2 es necesario que el depósito no esté lleno. Para que puedan funcionar los modos 3 y 4, debe de haber alguna pieza en el depósito y la entrega no estar llena. Por último, para utilizar el modo 5, se necesita que haya piezas en el depósito, pero que este no esté lleno.

De esto se encarga el programa “estado\_casillero”. Está escrito en lenguaje estructurado y es muy sencillo. Se encuentra en el Anexo I. Su funcionamiento se basa en evaluar las variables de estado del depósito y de la entrega, y activar o desactivar las variables *deposito\_lleno*, *deposito\_vacio* y *entrega\_llena*. En ningún caso es necesario una variable que indique que la entrega no contiene piezas.

Para que la variable *deposito\_lleno* valga 1, todas las variables de estado del depósito deben estar activadas. En caso contrario, vale 0.

Para que la variable *deposito\_vacio* valga 1, todas las variables de estado del depósito deben estar desactivadas. En caso contrario, vale 0.

Para que la variable *entrega\_llena* valga 1, todas las variables de estado de la entrega deben estar activadas. En caso contrario, vale 0.

```
IF (obj_D12=1 AND obj_D12=1 AND obj_D13=1 AND obj_D14=1 AND obj_D15=1 AND obj_D21=1 AND
    THEN
    deposito_lleno:=1;
    ELSE
    deposito_lleno:=0;
END_IF

IF (obj_D12=0 AND obj_D12=0 AND obj_D13=0 AND obj_D14=0 AND obj_D15=0 AND obj_D21=0 AND
    THEN
    deposito_vacio:=1;
    ELSE
    deposito_vacio:=0;
END_IF

IF (obj_E02=1 AND obj_E02=1 AND obj_E03=1 AND obj_E04=1 AND obj_E05=1)
    THEN
    entrega_llena:=1;
    ELSE
    entrega_llena:=0;
END_IF
```

Figura 6.18  
Sección de código del programa “estado\_casillero”



## 6.8 Modo Selección

El usuario debe ser capaz de elegir el modo de funcionamiento que desea para la máquina.

La programación que rige cómo y cuándo se entra en un modo de funcionamiento u otro y cómo y cuándo se sale es de vital importancia para el correcto funcionamiento de la automatización. Esta tarea la desempeña un programa escrito en lenguaje grafset llamado "Modo\_Seleccion". Se puede encontrar en el Anexo I.

Dentro de este programa se realiza una división jerárquica, cuyo primer escalón es diferenciar tres ramas. La rama de "automático", la rama de "manual" y la rama de "apagar". Para acceder a cada una de estas ramas se ha de activar una variable booleana del mismo nombre: *automatico*, *manual* y *apagar*; cuyo control se realiza desde el panel de control.

Se procede a analizar cada una de estas ramas con detalle:

### Automático

La rama de los procesos automáticos se divide en seis etapas. Cinco de ellas se corresponden a los cinco modos de funcionamiento descritos en el apartado de "Objetivos", y cuyo funcionamiento se analizará más adelante en este mismo punto. La etapa restante sirve como paso previo a la salida de la rama de procesos automáticos.

A las etapas de los distintos modos se llega si, estando en la rama automático, se selecciona el modo correspondiente a la etapa a través del panel de control. Estas llevan una acción asociada, que inhabilitan la selección de los depósitos, entregas o ambos, en los modos que no es necesario su uso. Esto se hace igualando a cero las variables booleanas que controlan dicha selección.

De esta manera, en el modo 1 y en el modo 5 se desactivan las entregas, y en el modo 2 y el modo 4 se desactivan tanto los depósitos como las entregas.

Se puede salir de estas etapas cumpliendo la condición "salir\_automatico" o "salir\_modox" (X representa el número del modo para cada caso).

La condición salir automático se cumple si se desactiva la variable *automatico*, se activa la variable *manual* o se activa la variable *apagar*.

Si esta condición se cumple, se accede a la sexta etapa en que se divide la rama automático. En esta etapa se desactiva la variable *automatico* (que puede llegar ya desactivada o no) y se dirige la acción a la primera etapa del grafset.

La condición "salir\_modox" funciona de manera similar a la anterior. Se cumple si se desactiva la variable del modo correspondiente o se activa la de cualquiera de los otros modos.

Si esta se cumple, se accede a la etapa de la rama automático.

## Manual

En la rama manual la primera etapa contiene una acción de entrada que pone a cero las variables booleanas *desde\_maquina* y *desde\_PC*. Estas variables son las encargadas de dar paso a los dos tipos de modo manual programados. Tanto el primero, desde la máquina, como el segundo, desde el PC, necesitan que la variable con su mismo nombre esté activa para funcionar.

También desactivan las variables de selección, tanto de depósitos como de entregas, y las variables de cada modo automático (*modo1*, *modo2*...).

Para salir de esta etapa, pueden darse tres condiciones. La primera implica salir del modo manual. Para que sea así, se deberá cumplir la condición “*salir\_manual*”. Funciona con la misma lógica que las condiciones vistas anteriormente. La condición se cumple si la variable manual se desactiva o cualquiera de las variables *automatico* o *apagar* se activan. Al cumplirse esta condición la variable *manual* pasa valer 0 y se vuelve a la primera etapa del grafset.

La segunda condición es que se seleccione desde el panel de control la opción Desde PC. Al darse esta condición se accede a una etapa que contiene una acción con el programa “*manual\_desde\_PC*”. Este programa se encarga de controlar el funcionamiento del modo manual controlado desde el panel de control. Se puede encontrar en el Anexo I, y está explicado con detalle en el punto “6.9.2.2 Desde PC”.

Para salir de esta etapa se puede hacer saliendo directamente del modo manual, esto es cumpliendo la transición “*salir\_manual*”, o bien saliendo del modo Desde PC. Para esto se ha desactivar la variable *desde\_PC* o activar la variable *desde\_maquina*. En el primer caso se accede a la primera etapa del grafset, mientras que en el segundo se llega a la primera etapa de la rama manual. En ambos casos se iguala a 0 la variable *desde\_PC* antes de llegar a estas etapas.

La última opción es que se seleccione la opción Desde Máquina. El proceso es equivalente al anterior pero ahora se permite el control desde los interruptores de la máquina. Se accede a una etapa con la acción que contiene el programa “*manual\_desde\_maquina*”. Este programa también está disponible en el Anexo I, y está explicado en el apartado “6.9.2.1 Desde máquina”

Para salir, igual que antes, se puede hacer cumpliendo la transición “*salir\_manual*” y acceder a la primera etapa del grafset o la transición “*salir\_maquina*” y acceder a la etapa de la rama manual. Para cumplir esta última transición se debe deseleccionar el botón desde máquina o seleccionar desde PC.

## Apagar

La rama apagar tiene prioridad sobre las demás. Cuando la variable *apagar* se activa desde el panel de control se accede a la primera etapa de dicha rama. La primera acción que se lleva a cabo es desactivar las variables *automatico* y *manual*.

Igual que en la rama manual, también se desactivan las variables de selección de depósitos y entregas y las variables de cada modo automático.

Para salir de esta rama la condición es diferente a las anteriores. En este caso solo se accederá de nuevo a la primera etapa del grafset si la variable *apagar* se desactiva. Por ello se puede decir que tiene prioridad sobre las demás ramas.

Además, el programa “Modo\_Seleccion” también realiza dos tareas más muy importantes para el correcto funcionamiento de toda la máquina: el reseteo de la variable de posición del eje z y el control de finalización de los procesos.

### **Reseteo de la variable de posición del eje z**

En la primera etapa del graficet y en la etapa de la rama automatico se ejecuta la acción “reiniciar\_z”. Esta acción activa las variables booleanas *eje\_z* y *adelante*, correspondientes al contactor que permite la alimentación del motor del eje z y al sentido horario de la marcha; de modo que la horquilla sube.

Para salir de esta etapa se debe cumplir que se active la variable booleana *cero\_z*, la cual tiene asignado el valor de la salida digital del sensor inductivo del eje en cuestión.

En otras palabras, la horquilla sube hasta la altura del sensor, gracias a lo cual se reinicia el valor de la variable de la coordenada del eje z.

Esta acción es necesaria para eliminar el error acumulado del que ya se ha hablado a lo largo de esta memoria. Cabe destacar que la acción se realiza en las dos etapas mencionadas porque son, en todos los casos, o una u otra, el destino donde se accede cuando se cambia de modo de funcionamiento. Así pues, el reseteo se llevará a cabo tanto si se cambia entre los modos automático, manual o apagar; como si se cambia entre los distintos modos del modo automático.

### **Control de finalización de los procesos**

Es muy importante que no se ejecute más de un modo a la vez. Se estarían realizando acciones con fines completamente distintos de manera simultánea y el funcionamiento no sería el deseado.

Para ello se crea una variable booleana, *proceso\_terminado* cuya misión será indicar si hay un proceso en marcha o no. Cuando el estado de esta variable sea TRUE significará que no hay ningún proceso llevándose a cabo, mientras que si es FALSE significa lo contrario. La variable se inicializa a 1, pues nada más empezar no habrá ningún proceso ejecutándose.

Así pues, en todos los programas que controlan los procesos, se desactiva esta variable cuando estos comienzan, y se vuelve a activar cuando los procesos mecánicos han finalizado.

En el programa “Modo\_Seleccion” esta variable impide que se ejecute el proceso de reseteo de la variable de posición z mientras se estén ejecutando otros programas.

Se verá en los puntos siguientes como esta variable se tiene en cuenta en todos los programas que controlan los procesos.

Como se puede apreciar después de la explicación del programa, éste se encarga de dar paso y salidas a los distintos modos, eliminar con cierta frecuencia el error de posición del eje z y evitar que se ejecuten varios procesos a la vez.

## 6.9 Modos de Funcionamiento

Después de ver cómo el programa “Modo\_Seleccion” organiza los distintos modos disponibles, se va a explicar el funcionamiento de cada uno de ellos.

### 6.9.1 AUTOMÁTICO

El modo automático se encarga de la distribución de piezas. Dentro de este se pueden diferenciar cinco modos distintos. En los modos 1 y 2 se recogen las piezas de la zona de recogida y se llevan a las celdas de la zona de depósito. Los modos 3 y 4 se encargan de trasladar las piezas de las celdas de depósito a las celdas de entrega. Por último, el modo 5 tiene como función cambiar las piezas de celda dentro de la zona de depósito.

A continuación se explica el funcionamiento de cada modo con detalle y las diferencias entre ellos. Las acciones que se emplean dentro de cada modo son, a menudo, muy parecidas. Este conjunto de acciones se encuentran en el Anexo II. Durante la siguiente explicación de los modos se limitará a explicar la función que desempeñan o características específicas que puedan tener en cada caso.

Las acciones muy sencillas se explicarán y no se incluirán en el Anexo.

#### 6.9.1.1 Modo 1

En el modo 1 el usuario escoge una celda de la zona del depósito donde quiere que se deje una pieza. Esto lo realiza a través del panel de control. El programa se encarga de recoger la pieza y llevarla a dicha celda, siempre y cuando ésta esté vacía.

El usuario debe seleccionar el botón del modo 1 del panel de control para utilizar este método. A continuación, se tiene que seleccionar la celda del depósito donde se desea que se quede la pieza recogida. El último paso es darle al “GO”. Entonces el proceso comienza a ejecutarse y terminará cuando la pieza se haya quedado en la celda seleccionada.

El programa que se encarga de los procesos correspondientes a este modo es “Modo\_1”. Está programado en lenguaje grafcet y se puede encontrar completo en el Anexo I.

Cuando se selecciona el modo 1 se activa la variable *modo1*. Si se da esta condición la primera acción que se ejecuta es “un\_origen”. Su misión es desactivar todas las variables de selección de celda de depósito menos la que esté seleccionada. Sirve para evitar que haya dos variables de este tipo activas a la vez. Por lo tanto cuando se seleccione una de estas celdas, las demás, aunque hubiese alguna activa, se desactivan.

De esta etapa se puede salir de dos maneras distintas. Una de ellas es si se quiere salir del modo 1. Para esto se debe dar que la variable *modo1* pase a valer 0 o que cualquiera de las otras 4 variables de los demás modos pasen a valer 1.

```
salir_modol:=(modo1=0 OR modo2=1 OR modo3=1 OR modo4=1 OR modo5=1);
```

Figura 6.19  
Transición “salir\_modol”

Hay que tener en cuenta que si se sale del modo automático, o se selecciona el modo manual o apagar, el programa "Modo\_Seleccion" se encarga de poner a cero todas las variables de modo. Si esto se cumple también se saldría de la etapa.

A través de esta salida se accede a la primera etapa del graficet.

La otra manera de salir de la etapa es si se cumplen varias condiciones simultáneamente.

Estas condiciones son:

Variable *proceso\_terminado* activa. Implica que no hay ningún otro proceso ejecutándose.

Variable *deposito\_lleno* inactiva. Significa que el depósito no está lleno y cabe otra pieza.

Se cumpla la condición "seleccion\_terminada". Esta condición se cumple si hay alguna variable de selección de depósito activa.

```
seleccion_terminada:=(D11=1 OR D12=1 OR D13=1 OR D14=1 OR D15=1 OR D21=1 ...
```

Figura 6.20  
Transición "seleccion\_terminada"

Variable *go* activa. Esta variable se activa cuando el usuario pulsa GO en el panel de control, por tanto las condiciones anteriores deberán cumplirse antes de que esto ocurra para que se inicie el proceso.

Si estas condiciones se cumplen se pasa a la siguiente etapa.

Las siguientes acciones a realizarse son "asignar\_variables\_posicion" y "asignar\_obj\_Dii".

La primera acción asigna a las variables de posición que se utilizarán a lo largo del programa un valor u otro en función de la selección que haya hecho el usuario. Además le da un valor a una variable auxiliar del 1 al 15, en función también de la selección realizada. Esto sirve para almacenar la selección realizada. Aunque se desactive la casilla de selección del depósito ya no hay problema.

```
IF (D24=1)  
THEN  
yDi:=yDi_c4;  
zDi_:=zDi_f2;  
c:=9;  
END_IF
```

Figura 6.21  
Sección de código de la acción "asignar\_variables\_posicion"

Además, la acción "asignar\_variables\_posicion" ayuda a eliminar el error acumulado en la variable de posición z. La posición *zDi* que corresponde a las celdas de depósito de la fila superior coincide con la máxima posición del eje z. Ésta es la misma posición que ocupa el sensor inductivo de este eje. Cuando se activa dicho sensor la variable z se reinicia y se elimina el error. Es por ello, que para favorecer que se llegue a dicha posición, cuando el usuario selecciona una casilla de esta fila, el valor asignado a *zDi* es mucho mayor del que debería (470 frente a 270).

```
IF (c=1 OR c=2 OR c=3 OR c=4 OR c=5 OR c=6 OR c=7 OR c=8 OR c=9 OR c=10)
  THEN
    zDi:=zDi_+35;
END_IF

IF (c=11 OR c=12 OR c=13 OR c=14 OR c=15)
  THEN
    zDi:=zDi_+235;
END_IF
```

Figura 6.22

Sección final del código de la acción "asignar\_variables\_posicion"

La segunda guarda en la variable *obj\_Dii* el valor de la variable de estado de la casilla seleccionada. Servirá para evaluar si es posible depositar la pieza en esta celda o si ya está ocupada. Aquí ya se utiliza la variable auxiliar *c* para saber la selección realizada.

```
IF (c=9)
  THEN
    obj_Dii:=obj_D24;
END_IF
```

Figura 6.23

Sección1 de código de la acción "asignar\_obj\_Dii"

Ambas acciones se encuentran en el Anexo II.

Para pasar a la siguiente etapa se evalúa el valor de la variable *obj\_Dii*. Si el depósito está vacío se pondrá en marcha la máquina. Si está ocupado se accede a la última etapa del grafset. Más adelante se verá que en esta etapa se igualan a 0 todas las variables de selección de depósito.

Si la celda está libre se empieza por desactivar la variable *proceso\_terminado*.

A continuación se comienza a mover la máquina. El primer paso es cargar la pieza en la zona de recogida. Después se llevará a la zona de depósito.

#### Carga de la pieza

Se empieza moviendo la máquina en el eje x, a continuación el eje y, y por último el eje z. La horquilla se lleva al punto determinado por las variables de posición de recogida (*xR*, *yR*, *zR*).

Las acciones encargadas del movimiento en cada eje son, respectivamente, "*x\_R*", "*y\_R*" y "*z\_R*". En el movimiento de los ejes x e y, cuando la posición de la horquilla dista más de 20 mm de la posición de recogida, la velocidad es la máxima. Si la distancia es menor la velocidad es la mínima. La programación y explicación de las tres acciones se encuentran en el Anexo II.

A continuación se realiza el enganche de la pieza. No se dispone de sensor para saber si la pieza está o no disponible en la zona de recogida, por lo que tiene que ser el usuario quien lo indique pulsando el botón PIEZA COLOCADA en el panel de control. Para que este botón solo se vea cuando se ha de utilizar, en la etapa previa al enganche se activa una variable auxiliar *a*, que se utilizará para hacer visible dicho botón. Al pulsarlo se activa momentáneamente la variable *pieza\_colocada*, y se pasa a la siguiente etapa.

**PIEZA COLOCADA**

Figura 6.24

Botón PIEZA COLOCADA

La acción “x\_enganche” se encarga de que la horquilla se mueva en el eje x hasta que enganche la pieza, esto es, hasta que la posición x sea la marcada por la variable  $x_{eng}$ .

Traslado de la pieza al depósito

El siguiente paso es subir la horquilla hasta la posición almacenada en  $z_{Di}$ . Se recuerda que el valor de estas variables de posición se cargaba en función de la selección del usuario por la acción “asignar\_variable\_posicion”.

De subir la horquilla hasta  $z_{Di}$  se encarga la acción “z\_arriba”. De esta etapa se sale cuando se llega a esta posición o si se activa el sensor inductivo del eje z. La última condición es necesaria, pues se recuerda que la posición  $z_{Di}$  guardada para las celdas de la última fila es superior a la máxima, y se activará el sensor antes de alcanzarla.

A continuación la acción “y\_Di” se encarga de mover la horquilla sobre el eje y hasta la posición guardada en  $y_{Di}$ .

Una vez alcanzada la posición del depósito en el eje y, la acción “x\_adelante” se encarga de que la horquilla se desplace a lo largo del eje x hasta la posición guardada en la variable  $x_D$ . En este caso esta posición es la  $x=0$ . Una vez sea alcanzada la horquilla ya estará dentro de la celda.

El siguiente paso es hacer descender la horquilla para que la pieza sea depositada en la celda. De esto se encarga la acción “z\_abajo”. La horquilla baja hasta que alcanza la posición guardada en  $z_{Di}$ .

El último movimiento que se realizará en este modo será el de extraer la horquilla de la zona de depósito. La función es ejecutada por la acción “x\_atras” y termina cuando la coordenada en el eje x alcanza el valor de  $x_{D_}$ .

El funcionamiento de las acciones “z\_abajo”, “y\_Di”, “x\_adelante”, “z\_abajo” y “x\_atras” son muy parecidos a las acciones “x\_R”, “y\_R” y “z\_R”. La única diferencia reside en las variables que tienen en cuenta, y que salvo la acción “y\_Di”, en estas acciones el desplazamiento solo se hace en un sentido. Todas ellas se encuentran en el Anexo II.

Una vez terminada la deposición de la pieza, la acción “casilla\_ocupada” hace que la variable  $obj_{Dii}$  pase a valer 1. A continuación se ha de trasladar el valor de estado auxiliar  $obj_{Dii}$  a las variables de estado de las celdas de depósito. Esta es la función de la acción “reset\_variables”.

Por último, en la última etapa (a la cual también se llega si cuando se comprobaba que el estado del depósito éste está lleno), se activa la variable  $proceso\_terminado$  y se igualan a cero todas las variables de selección de depósito.

Aquí concluye el modo 1. Se vuelve de nuevo a la primera etapa del graficet donde se mantiene a la espera.

```
IF (c=9)
THEN
obj_D24:=obj_Dii;
END_IF
```

Figura 6.25  
Sección de código de la acción “reset\_variables”

### 6.9.1.2 Modo 2

El funcionamiento del modo 2 es igual que el del modo 1, salvo porque se elimina la parte en que el usuario selecciona la celda del depósito donde va la pieza. En lugar de eso, en este modo se recogen piezas de forma continua y se depositan en las celdas que estén libres por orden.

La programación es, por tanto, muy similar a la del modo 1. Se omite la explicación de las acciones cuya función ya se haya expuesto en el modo 1. La mayor diferencia entre las dos programaciones es el empleo de un contador para ir ocupando las celdas de manera consecutiva en el modo 2.

El programa que lo controla, “Modo\_2”, también escrito en grafset, se encuentra en el Anexo I.

Para iniciar el modo 2 se han de cumplir las siguientes condiciones son:

Variable *modo2* activa.

Variable *proceso\_terminado* activa.

Variable *deposito\_lleno* inactiva.

Variable *go* activa.

Si estas condiciones se cumplen se pasa a la segunda etapa.

La primera acción que se lleva a cabo en el programa es “inicializar\_el\_contador”. Como su nombre indica, su función es inicializar el contador. La variable que servirá de contador es *Dii\_c*, y se inicializa con el valor de la variable *Dini\_c*, cuyo valor es 1.

Se sigue con la etapa “modo2\_2”, donde se llevan a cabo las acciones “asignar\_variables\_posicion” y “asignar\_obj\_Dii”. Actúan de manera idéntica que en el modo 1, salvo que en lugar de evaluar las variables de selección del depósito, en este caso se compara el valor del contador. Tampoco se utiliza una variable auxiliar como la variable *c* del modo 1, pues ahora no es necesaria.

```
IF (Dii_c=9)
THEN
yDi:=yDi_c4;
zDi_:=zDi_f2;
END_IF
```

Figura 6.26

Sección de código de la acción “asignar\_variables\_posicion”

```
IF (Dii_c=9)
THEN
obj_Dii:=obj_D24;
END_IF
```

Figura 6.27

Sección de código de la acción “reset\_variables”

Seguidamente se comprueba si la celda correspondiente al valor del contador está vacía o llena. Esto se hace comprobando el valor de *obj\_Dii*.



Si la celda del depósito está llena, se accede a la etapa modo2\_4. Esta etapa tiene asociada como acción de entrada “contador\_mas\_uno”.

La función de esta acción es aumentar en uno el valor del contador, y en caso de que haya llegado al valor máximo, determinado por la variable *Dfin\_c*, reiniciarlo. Se ha establecido 15 como valor de *Dfin\_c*, pues es el número de celdas de depósito.

```
IF (Dii_c=Dfin_c)
THEN
    Dii_c:=Dini_c;
ELSE
    Dii_c:=Dii_c+1;
END_IF
```

Figura 6.28  
Acción “contador\_mas\_uno”

Si la variable *modo2* sigue activa y siguen habiendo huecos (*deposito\_lleno* vale 0), se continúa de nuevo en la etapa modo2\_2, y se repite el proceso con el siguiente valor del contador. Si no se cumplen estas condiciones se vuelve a la primera etapa del graficet.

Por otra parte, si el depósito está vacío, se desactiva la variable *proceso\_terminado* y se empiezan a efectuar los movimientos necesarios para llevar la horquilla a la zona de recogida.

Esta parte es idéntica al modo 1, por lo que se explicará por encima.

Las acciones “x\_R”, “y\_R” y “z\_R” se encargan de mover la máquina hasta la zona donde se recoge la pieza. Cuando el usuario accione el botón PIEZA COLOCADA, la acción “x\_enganche” desplazará la horquilla sobre el eje x hasta coger la pieza. A continuación se ejecutarán, una detrás de otra, las acciones “z\_arriba”, “y\_Di” y “x\_adelante” trasladan la pieza a la celda del depósito que le corresponda al contador. Por último, las acciones “z\_abajo” y “x\_atras” depositarán la pieza y sacarán la horquilla de la celda respectivamente.

Por último iguala a 1 la variable *obj\_Dii*, la acción “reset\_variables” guarda en la variable de estado de la casilla correspondiente al contador el valor de *obj\_Dii*, y se activa la variable *proceso\_terminado*.

De aquí se pasa a la etapa modo2\_4, donde, como ya se ha visto, se aumenta el contador y si el modo 2 sigue seleccionado y queda espacio en el depósito se sigue con el ciclo; si no, se vuelve a la primera etapa.

### 6.9.1.3 Modo 3

El modo 3 tiene la función de llevar una pieza de una celda de la zona de depósito a una de la zona de entrega. Ambas serán escogidas por el usuario a través del panel de control.

El programa encargado de la ejecución de dicha tarea es “Modo\_3”. Está hecho en lenguaje grafcet, y se encuentra íntegro en el Anexo I.

Una vez más las acciones que se llevan a cabo son similares a las explicadas en el modo 1, por lo que la explicación se limitará a las diferencias con lo que ya se ha visto.

Cuando el usuario selecciona el modo 3 en el panel de control, la variable *modo3* se activa. De esta manera, se accede a una etapa del grafcet donde se espera a que el usuario seleccione la celda del depósito que contiene la pieza y la celda de la zona de entrega donde la quiere trasladar.

En esta etapa se ejecuta la acción “un\_origen\_un\_destino”. Su función es muy similar a la acción “un\_origen” explicada en el modo 1. Sirve para evitar que haya dos variables de selección activas a la vez. La diferencia reside en que, en este caso, desactiva tanto las variables de selección de celdas del depósito como las de la entregas. Por lo tanto, haya alguna selección hecha o no, cuando se selecciona una de estas celdas, las demás se desactivan.

Hay dos caminos distintos para salir de esta etapa. Uno consiste en salir del modo 3. Para ello, el usuario debería deseleccionar el modo 3, o seleccionar cualquiera de los otros 4 modos de funcionamiento automático. Como ya se ha visto, si se deselecciona el modo automático o se selecciona el modo manual o apagado, el programa “Modo\_Seleccion” se encarga de desactivar la variable *modo3*, por lo que también se saldría de la etapa.

```
salir_mod03:=(mod01=1 OR mod02=1 OR mod03=0 OR mod04=1 OR mod05=1);
```

Figura 6.29  
Transición “salir\_mod03”

Por este camino se vuelve a la primera etapa del grafcet.

La otra manera de salir de la etapa es si se cumplen las siguientes condiciones simultáneamente:

Variable *proceso\_terminado* activa. Implica que no hay ningún otro proceso ejecutándose.

Variable *deposito\_vacio* inactiva. Significa que el depósito no está vacío y dispone de piezas para llevar a la entrega.

Variable *entrega\_llena* inactiva. Significa que la entrega no está llena y cabe otra pieza.

Se cumpla la condición “seleccion\_terminada”. Esta condición se cumple si hay alguna variable de selección de celda de depósito y alguna de celda de entrega activas.

```
seleccion_terminada:=(E01=1 OR E02=1 OR... OR E05=1) AND (D11=1 OR D12=1 OR ...OR D35=1)
```

Figura 6.30  
Transición “seleccion\_terminada”

Variable *go* activa.

Si estas condiciones se cumplen se pasa a la siguiente etapa.

En esta etapa se ejecutan las acciones “asignar\_variables\_posicion” y “obj\_Dii\_obj\_Eii”. Las diferencias con las etapas equivalentes del modo 1 son que tienen en cuenta también las variables de la zona de entrega. La programación sigue las mismas directrices.

La primera acción asigna a las variables de posición que se utilizarán a lo largo del programa, tanto de la zona de depósito como la de la entrega, el valor correspondiente a la selección que del usuario. Además aquí se vuelve a emplear variables auxiliares para almacenar la selección realizada. Se utilizará la variable entera *c* para las celdas de depósito, con valores de 1 a 15; y la variable entera *e* para las celdas de entrega, con valores de 1 a 5.

```
IF (E03=1)
THEN
yEi:=yDi_c3;
e:=3;
END_IF

IF (D15=1)
THEN
yDi:=yDi_c5;
zDi:=zDi_f1;
c:=5;
END_IF
```

Figuras 6.31 y 6.32  
Secciones de código de la acción “asignar\_variables\_posicion”

También asigna un valor muy superior al que correspondería a la variable *zDi* si la celda del depósito seleccionada es de la fila superior. La finalidad es favorecer que el error acumulado en la variable de posición *z* se elimine con más frecuencia.

La segunda acción guarda en la variable *obj\_Dii* y *obj\_Eii* los valores de las variables de estado de las casillas seleccionadas del depósito y de la entrega respectivamente. En este punto ya se hace uso de las variables auxiliares *c* y *e*.

```
IF (e=2)
THEN
obj_Eii:=obj_E02;
END_IF

IF (c=12)
THEN
obj_Dii:=obj_D32;
END_IF
```

Figuras 6.33 y 6.34  
Secciones de código de la acción “obj\_Dii\_obj\_Eii”

Ambas acciones se encuentran Anexo II.

El siguiente paso es comprobar si la celda de origen (depósito) está llena y la celda de destino (entrega) vacía. Si no es así se desactivan todas las variables de selección y se vuelve a la primera etapa del graficet. Si se cumple se desactiva la variable *proceso\_terminado* y se procede a extraer la pieza del depósito.

Extracción de la pieza del depósito.

Lo primero es meter la horquilla en la celda que se ha seleccionado. Se deberán alcanzar, y por este orden, las posiciones en los ejes marcadas por las variables *zDi*, *yDi* y *xD*. De esto se encargan las acciones “z\_Di\_”, “y\_Di” y “x\_adelante” respectivamente.



A continuación se ha de subir la horquilla lo suficiente para enganchar la pieza, pero menos de lo que haría que ésta tocara el techo de la celda. Esta posición viene determinada por la variable  $zDi$ . De llevar la horquilla a dicha posición se encarga la acción “z\_arriba”.

El último movimiento para finalizar la extracción de la pieza es sacar la horquilla de la celda hasta la posición  $xD_$ . La acción encargada de este movimiento es “x\_atras”.

Una vez acabada la extracción, la variable  $obj\_Dii$  pasa a valer 0, y se ejecuta la acción “reset\_variables”. Con esta acción en la variable de estado de la celda de depósito vaciada se refleja que ya no contiene ninguna pieza. Además permite la visualización instantánea en el panel de control.

Deposición de la pieza en la entrega.

Las acciones “y\_Ei” y “z\_abajo” llevan, en este orden, la horquilla a las posiciones  $yEi$  y  $zEi$ . Una vez colocada la horquilla delante de la celda de la entrega, la acción “x\_adelante” introduce la horquilla. A continuación la pieza se deposita con la acción “z\_abajo” y por último “x\_atras” se encarga de extraer la horquilla de la celda.

La variable  $obj\_Dii$  pasa a valer 1 y se vuelve a ejecutar la acción “reset\_variables”. En el panel de control se podrá apreciar el traslado de la pieza.

El último paso del programa es igualar a 0 todas las variables de selección de las celdas de depósito y entrega y activar la variable  $proceso\_terminado$ .

#### 6.9.1.4 Modo 4

El modo 4 realiza la misma función que el modo 3 pero de manera automática. Recoge las piezas de las celdas del depósito por orden y las va dejando en las celdas de la entrega también por orden. Siempre y cuando haya piezas en el depósito y celdas libres en la entrega.

Para evaluar y actuar sobre las distintas celdas siguiendo un orden se utiliza un sistema de contadores. Funciona de la misma manera que en el modo 2, pero ahora se dispone de dos contadores en lugar de uno. Uno para las celdas de la entrega, *Eii\_c*, y otro para las del depósito, *Dii\_c*.

El programa encargado de todo el proceso es “Modo\_4” y se encuentra disponible en el Anexo I.

Para que el “Modo\_4” empiece a ejecutarse se deben cumplir las siguientes condiciones:

Variable *modo4* activa.

Variable *proceso\_terminado* activa.

Variable *deposito\_vacio* inactiva.

Variable *entrega\_llena* inactiva.

Variable *go* activa.

El significado de cada condición ya ha sido explicado a lo largo de este punto.

La primera acción que se lleva a cabo es “inicializa\_contadores”. Como su nombre bien indica, su misión es asignar los valores iniciales a los contadores de las celdas de depósito y entrega.

```
Dii_c:=Dini_c;  
Eii_c:=Eini_c;
```

El valor de *Dini\_c* y *Eini\_c* es 1.

Figura 6.35  
Acción “inicializa\_contadores”

A continuación se accede a la etapa “modo4\_2”. En esta etapa se ejecutan las acciones “asignar\_variables\_posicion” y “obj\_Dii\_obj\_Eii”. Funcionan exactamente igual que en el modo 3, pero en lugar de utilizar como referencia las variables de selección y las variables auxiliares *c* y *e*, utilizan los contadores. En este caso, tampoco son necesarias dichas variables auxiliares.

Para salir de esta etapa se evalúa si el depósito está vacío o la entrega llena. Si una de estas dos condiciones se cumple se accede a la primera etapa. En caso contrario se continúa con la siguiente etapa.

Esta condición debe resultar redundante ahora mismo. La explicación es que a la etapa “modo4\_2” se puede acceder desde más etapas a lo largo del graficet, y en estas condiciones sí tiene sentido se realice este análisis. Más adelante se verá con claridad.

El siguiente paso es evaluar si la celda del depósito contiene o no una pieza. Para esto se compara el valor de la variable *obj\_Dii*.

Si el depósito está vacío, se accede a una etapa con una acción de entrada asociada llamada “contador\_dep\_mas\_uno” que aumenta en uno el contador de las celdas del depósito. Si el contador ha llegado al número máximo, guardado en la variable *Dfin\_c* (cuyo valor es 15, igual al número de celdas), éste se reinicia.

Una vez aumentado el contador se accede de nuevo a la etapa “modo4\_2”.

Si el depósito está lleno se realiza el mismo proceso con las entregas.

Se evalúa ahora la variable *obj\_Eii* para saber el estado de la celda de entrega.

Si la celda está ocupada se accede a una etapa con una acción de entrada asociada llamada “contador\_ent\_mas\_uno” que, igual que antes, aumenta en uno el contador, pero ahora el de las celdas de la entrega. Si el contador ha llegado al número máximo, guardado en la variable *Efin\_c* (cuyo valor es 5, igual al número de celdas), éste se reinicia.

Si, en cambio, la celda está libre, se desactiva la variable *proceso\_terminado* y se empiezan a ejecutar las acciones.

Las acciones que vienen a continuación y que mueven la máquina se explicarán por encima y sin entrar en detalle, pues son las mismas que se utilizan en el modo 3.

El primer proceso que se lleva a cabo es la retirada de la pieza de la celda del depósito correspondiente. Las acciones “z\_Di\_”, “y\_Di” y “x\_adelante” introducen la horquilla en la celda, la acción “z\_arriba” levanta la pieza y “x\_atrás” extrae la horquilla con la pieza de la zona de depósito.

Al finalizar la extracción, se iguala a 0 la variable *obj\_Dii* y se ejecuta la acción “reset\_variables”. De esta manera, se actualiza el estado de la celda de depósito vaciada y se permite la visualización instantánea en el panel de control.

El siguiente proceso es la deposición de la pieza en la celda de la entrega que corresponda según el contador. En este caso las acciones “y\_Ei” y “z\_abajo” se encargan de colocar la horquilla enfrente de la celda de entrega. A continuación la acción “x\_adelante” introduce la horquilla en la celda y con la acción “z\_abajo” se deposita la pieza. Por último la horquilla se extrae mediante “x\_atras”.

El movimiento de la máquina termina así un ciclo, por lo que se activa la variable *proceso\_terminado*.

El siguiente paso es activar la variable *obj\_Eii* y ejecutar de nuevo la acción “reset\_variables”. En el panel de control se apreciará que la pieza ha sido depositada en una celda de la entrega.

A continuación se accede a la última etapa del grafset. Esta etapa tiene asociada otra acción de entrada que también sirve para actualizar los contadores. La diferencia con las anteriores es que ahora se actualizan ambos.

```
IF (Dii_c=Dfin_c)
THEN
    Dii_c:=Dini_c;
ELSE
    Dii_c:=Dii_c+1;
END_IF
```

Figura 6.36  
Acción “contador\_dep\_mas\_uno”

```
IF (Eii_c=Efin_c)
THEN
    Eii_c:=Eini_c;
ELSE
    Eii_c:=Eii_c+1;
END_IF
```

Figura 6.37  
Acción contador\_ent\_mas\_uno”

```
(*Pasar a la siguiente casilla de la entrada*)  
  
IF (Eii_c=Efin_c)  
    THEN  
        Eii_c:=Eini_c;  
    ELSE  
        Eii_c:=Eii_c+1;  
    END_IF  
  
(*Pasar a la siguiente casilla del depósito*)  
  
IF (Dii_c=Dfin_c)  
    THEN  
        Dii_c:=Dini_c;  
    ELSE  
        Dii_c:=Dii_c+1;  
    END_IF
```

Figura 6.38  
Acción "contador\_mas\_uno"

Para salir de esta etapa se evalúa la variable *modo4*. Si ha pasado a estar inactiva, se vuelve a la etapa inicial del grafset; y si sigue activa, la acción del grafset se dirige a la etapa "modo4\_2".

Como se ha visto anteriormente, después de esta etapa se comprueba si el depósito sigue disponiendo de alguna pieza o si la entrega ya está llena. Ahora se explica lo que antes parecía una redundancia, pues es posible que el último ciclo realizado haya extraído la última pieza del depósito u ocupado el último hueco de la entrega.

#### 6.9.1.5 Modo 5

El modo 5 permite desplazar una pieza de una celda del depósito a otra. Una vez seleccionado este modo de funcionamiento, el usuario debe seleccionar las dos celdas, la que contiene la pieza que desea mover y la celda donde desea moverla. Lo debe hacer en ese orden, y darle al botón GO.

El programa que ejecuta todas las acciones requeridas es "Modo\_5". Está escrito en lenguaje grafcet y se encuentra en el Anexo I.

La programación de este modo de funcionamiento es de las más complejas. En todos los modos de funcionamiento automático que se han visto, la misión es recoger una pieza de un lugar y depositarla en otro. Y lo mismo ocurre con este.

Pero hay una gran diferencia. En los modos anteriores el lugar originario de la pieza y el de destino están claramente diferenciados. En este modo de funcionamiento esto no es así. El hecho de trasladar una pieza de una celda de depósito a otra implica que la celda origen podría ser perfectamente la de destino en otra ocasión, y viceversa. Y esto implica ciertas complicaciones a la hora de programar.

Para seleccionar el origen y el destino se seleccionan el mismo tipo de botones, que están asociados a las variables de selección de las celdas de depósito. Por ello es necesario, primero, diferenciar cuando se está seleccionando el origen y cuando el destino, y segundo, guardar las selecciones en lugares distintos para tenerlas identificadas en todo momento.

Para ello se utilizan unas variables booleanas auxiliares de dos tipos, unas para el origen y unas para el destino. Hay quince variables de cada tipo, tantas como número de celdas de depósito. El nombre de las variables son *orij* y *destij*, donde *i* representa el número de fila, del 1 al 3, y *j* el número de columna, del 1 al 5. Un poco más adelante se verá como en estas variables se guarda la selección realizada por el usuario.

Para solucionar cómo diferenciar la selección del origen de la selección del destino se hace uso de un contador. Este contador servirá para saber el número de celdas seleccionadas que hay en cada momento en el panel de control. La variable entera que almacenará el dato es *c*. A lo largo de este mismo punto se verá su modo de funcionamiento.

También se han de usar para el correcto funcionamiento quince variables auxiliares más. Estas variables son de tipo entero y se nombran *cij*, donde *i* y *j* representan lo mismo que en las anteriores.

A continuación se analiza el programa, y se explica cada parte. Se hablará de origen y destino para referirse a la celda que contiene la pieza que se desea mover y a la celda donde se quiere mover respectivamente.

Para empezar, se ha de cumplir que *modo5* esté activa y que el depósito no esté ni lleno ni vacío, *deposito\_lleno* y *deposito\_vacio* deben valer 0. Si se dan estas condiciones se accede a la etapa "modo5\_1"

En esta etapa hay asociada una acción de entrada que inicializa a 0 la variable del contador *c* y las variables auxiliares de tipo *orij* y *destij*. Se continúa con la etapa "modo5\_2", donde se realizan los procesos necesarios de selección.



Selección de origen y destino.

A continuación se ejecuta la acción “contador\_seleccionados”. Su misión es escribir en las variables auxiliares *cij* un 1 si la celda del depósito correspondiente está seleccionada y un 0 si no es así. Además suma todas estas variables y guarda el resultado en *c*, de manera que se tiene en esta última variable el número de cuantas celdas hay seleccionadas.

```
c:=c11+c12+c13+c14+c15+c21+c22+c23+c24+c25+c31+c32+c33+c34+c35;
```

Figura 6.40

Sección final de código de la acción “contador\_seleccionados”

```
IF (D33=1)  
THEN  
c33:=1;  
END_IF  
IF (D33=0)  
THEN  
c33:=0;  
END_IF
```

Figura 6.39

Sección de código de la acción “contador\_seleccionados”

Simultáneamente se ejecuta “origen\_destino”. Es probablemente la acción más importante del programa. Su función es guardar la selección hecha en el panel de control por el usuario en las variables *orij* y *destij*.

Se recuerda que el usuario selecciona primero la celda de origen y a continuación la de depósito. Se ha realizado la programación de manera que permita mantener activa la primera selección a la vez que se activa la segunda. Para que esto sea posible se utiliza el contador.

La programación de la acción “origen\_destino” se basa en lo siguiente: cuando el contador vale 1, la celda seleccionada es la de origen; cuando vale 2, la celda seleccionada que antes no estaba seleccionada es la de destino.

Para conseguir este objetivo la acción se divide en dos partes, una para el origen y otra para el destino.

La primera parte está, toda ella, condicionada a que el valor de *c* sea 1. Si esto se cumple, se evalúa la variable *cij* y si vale 1 se activa la variable *orij*. De esta manera queda registrada la selección de la celda de origen. Este hecho es muy importante en la segunda parte, pues es necesario identificar, de las dos celdas activas, cual es la última que ha sido seleccionada.

```
IF (c13=1)  
THEN  
or13:=1;  
END_IF  
IF (c13=0)  
THEN  
or13:=0;  
END_IF
```

Figura 6.41

Sección de código de la primera parte de la acción “origen\_destino”

```
IF (c13=1 AND or13=0)  
THEN  
dest13:=1;  
END_IF  
IF (c13=0 OR or13=1)  
THEN  
dest13:=0;  
END_IF
```

Figura 6.42

Sección de código de la segunda parte de la acción “origen\_destino”

La segunda parte está condicionada a que el valor de  $c$  sea 2. La condición para activar la variable  $destij$  es que la celda esté activa ( $cij=1$ ) y que no sea la de origen ( $orij=0$ ).

En las imágenes se aprecia con más claridad el funcionamiento de la acción.

Utilizando los datos almacenados en las variables auxiliares sobre la selección del usuario, la acción "asignar\_variables\_posicion" guarda los datos de las posiciones características de las dos celdas seleccionadas. Las variables auxiliares de posición están duplicadas en la programación de este modo de funcionamiento. Las correspondientes a la celda de origen llevan en su nombre  $or_$ , mientras que las de destino llevan  $dest_$ . En las siguientes imágenes se aprecia con claridad.

```
IF (or13=1)
  THEN
    yD_or_i:=yDi_c3;
    zD_or_i:=zDi_fl;
  END_IF
```

Figura 6.43  
Sección de código de la acción  
"asignar\_variables\_posicion"

```
IF (dest13=1)
  THEN
    yD_dest_i:=yDi_c3;
    zD_dest_i:=zDi_fl;
  END_IF
```

Figura 6.44  
Sección de código de la acción  
"asignar\_variables\_posicion"

Esta acción, igual que todas las equivalentes de los modos vistos anteriormente, también contribuye a eliminar el error acumulado en la variable de posición  $z$ . Si alguna de las celdas seleccionadas pertenece a la última fila del depósito, a la variable de posición que le corresponda,  $zD_or_i$  o  $zD_dest_i$ , se le asigna un valor muy superior al que debería.

```
IF (or11=1 OR or12=1 OR or13=1 OR ... OR or25=1)
  THEN
    zD_or_i:=zD_or_i+35;
  END_IF

IF (or31=1 OR or32=1 OR or33=1 OR or34=1 OR or35=1)
  THEN
    zD_or_i:=zD_or_i+235;
  END_IF
```

Figura 6.45  
Sección de código de la acción  
"asignar\_variables\_posicion"

```
IF (dest11=1 OR dest12=1 OR dest13=1 OR ... OR dest25=1)
  THEN
    zD_dest_i:=zD_dest_i+35;
  END_IF

IF (dest31=1 OR dest32=1 OR dest33=1 OR dest34=1 OR dest35=1)
  THEN
    zD_dest_i:=zD_dest_i+235;
  END_IF
```

Figura 6.46  
Sección de código de la acción  
"asignar\_variables\_posicion"

Del mismo modo, la acción “obj\_or\_Dii\_obj\_dest\_Eii” asigna a las variables auxiliares *obj\_or\_Dii* y *obj\_dest\_Dii* el valor correspondiente a la variable de estado de las celdas de origen y de destino respectivamente.

```
IF (or13=1)
  THEN
    obj_or_Dii:=obj_D13;
  END_IF
```

```
IF (dest13=1)
  THEN
    obj_dest_Dii:=obj_D13;
  END_IF
```

Figuras 6.47 y 6.48  
Secciones de código de la acción “obj\_or\_Dii\_obj\_dest\_Dii”

Si durante la selección, el usuario desactiva el modo 5, activa uno de los otros o sale del modo manual, el graficet vuelve a la primera etapa.

Una vez finalizada la selección, se deberán cumplir las siguientes condiciones para continuar:

Variable *proceso\_terminado* activa.

Se cumpla la condición “seleccion\_terminada”. Esta condición se cumple si  $c=2$ .

Variable *go* activa.

Si todo esto se cumple, lo siguiente es evaluar si la celda de origen está llena y la de destino vacía.

Si no es así, se vuelve a la primera etapa del graficet. Si estas condiciones se cumplen se desactiva la variable *proceso\_terminado* y se pone en marcha la máquina. Se empieza con la extracción de la pieza de la celda de origen y a continuación se realiza la deposición en la celda de destino.

A partir de aquí las acciones que toman parte en el proceso son conocidas, y siguen el mismo funcionamiento que en los modos anteriores.

Extracción de la pieza

Para empezar, la acción “z\_or” mueve la horquilla en el eje z hasta la variable la posición que determina *z\_D\_or\_i*. A continuación, la acción “y\_or” realiza lo propio en el eje y. Ahora queda que la horquilla avance hasta el interior de la celda, de lo cual se encarga la acción “x\_adelante”. Para subir la horquilla y enganchar la pieza se activa la acción “z\_arriba”. Por último, la acción “x\_atras” extrae de la celda la horquilla con la pieza hasta la posición marcada por *xD\_*.

A continuación la acción “deposito\_or\_libre” iguala a 0 la variable *obj\_or\_Dii* y “reset\_variables” se encarga de actualizar dicho valor en la variable de estado de la celda origen. De manera queda reflejada la extracción de la pieza, y se puede apreciar en el panel de control.

Hasta ahora se han utilizado las variables de origen (*\_or\_*), desde aquí se emplearán las de destino (*\_dest\_*).

Deposición de la pieza



Para depositar la pieza, la primera acción que tiene lugar es "y\_dest". Esta acción traslada la horquilla hasta la posición que marca la variable  $y_{D\_dest\_i}$ . A continuación, la acción "z\_dest" mueve la horquilla en el eje z hasta alcanzar  $z_{D\_dest\_i}$  y la acción "x\_adelante" introduce la horquilla y la pieza en la celda de destino. El siguiente paso es bajar la horquilla para depositar la celda. Esto lo hace la acción "z\_abajo". Por último, "x\_atras" saca la horquilla de la celda.

Con esto se finalizan los movimientos del modo 5.

A continuación "deposito\_dest\_ocupado" desactiva la variable  $obj\_des\_Dii$  y "reset\_variables" actualiza el estado del depósito.

Las últimas acciones que se llevan a cabo son desactivar todas las variables de selección de depósito, y se activa la variable  $proceso\_terminado$ . Con esto termina el ciclo del modo y se vuelve a la primera etapa.

### 6.9.2 MANUAL

El modo manual proporciona el control de los movimientos de la máquina a un operario. Se diferencia dos tipos de control manual, desde máquina y desde PC.

En el primero el manejo se efectúa desde seis interruptores y dos pulsadores que se encuentran en la propia máquina. Los interruptores controlaran el eje en el que se desea el movimiento y el sentido de éste. Los pulsadores sirven para regular la velocidad. Con el pulsador superior se aumentará y con el inferior disminuirá.

En el control manual desde PC se tienen los mismos componentes, pero en el panel de control. La otra diferencia esencial es que para controlar los movimientos, en lugar de interruptores se tienen también pulsadores.

La programación de ambos programas es muy parecida. A continuación se explica el funcionamiento del programa que permite el control manual desde la máquina. Después, para entender el segundo programa, se explican las pequeñas diferencias que hay con el primero.

#### 6.9.2.1 Desde máquina

Antes de empezar, y para que la explicación sea más sencilla, se va a presentar las variables que se utilizan específicamente para este programa.

Las variables se asocian a la señal digital emitida por los 6 interruptores y los 2 pulsadores.

Variables booleanas del control del movimiento:

*x\_adelante*

*x\_atras*

*y\_derecha*

*y\_izquierda*

*z\_arriba*

*z\_abajo*

Variables booleanas del control de velocidad:

*speed\_up*

*speed\_down*

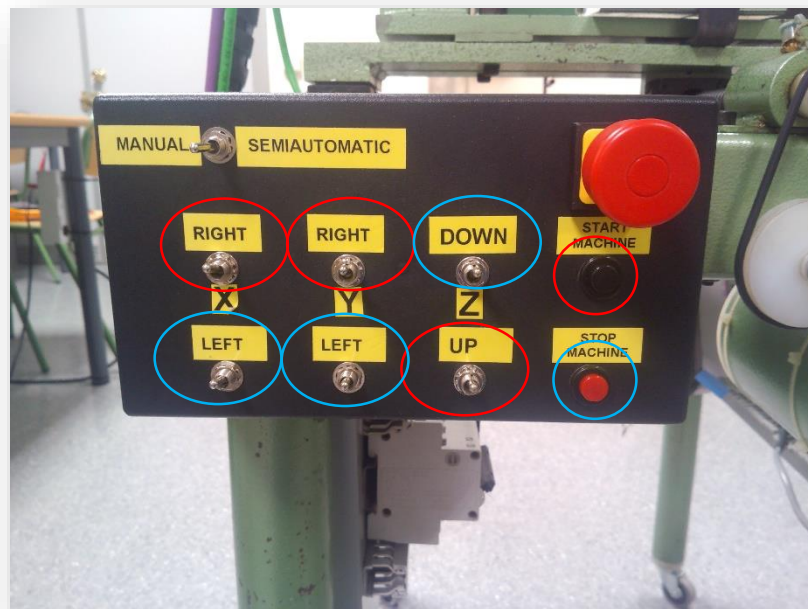


Figura 6.49  
Controles del modo manual desde máquina

La programación del modo manual desde la máquina persigue, por una parte, que al seleccionar uno de estos interruptores se produzca el movimiento pertinente, y por otra, proporcionar control de la velocidad.

El programa que se encarga de esta tarea es “manual\_desde\_maquina”. Es una acción asociada al programa “Modo\_Seleccion”. Está escrito en texto estructurado y se encuentra en el Anexo I.

Para que se ejecute el programa se ha de activar desde el panel de control la variable *desde\_maquina*. Para hacer esto, el usuario ha de pulsar el botón con el mismo nombre, como se verá más adelante cuando se explique el panel de control. Por lo tanto todo el programa está condicionado a que dicha variable esté activa.

La programación también se divide en el control de movimiento y en el control de velocidad.

### Control de movimiento

Esta es la parte más sencilla. Contiene seis partes iguales, una para cada interruptor. Si la señal del interruptor está activa, el programa iguala a 1 las dos variables de movimiento correspondientes, la del eje y la del sentido de la marcha.

Por ejemplo, si se acciona el interruptor para subir la horquilla en el eje z, se activará la variable *z\_arriba*, y el programa se encargará de poner a 1 las variables *eje\_z* y *adelante*.

```
IF (z_arriba=1)
  THEN
    eje_z:=1;
    adelante:=1;
```

Por último, esta parte también se encarga de desactivar las variables de movimiento una vez los interruptores cambien a la posición de paro.

Figura 6.50  
Sección de código del programa  
“manual\_desde\_maquina”

Si se igualasen a 0 las variables de movimiento cuando los interruptores no están accionados, sería imposible, por ejemplo, mover hacia adelante la máquina en el eje x y hacia la izquierda en el eje y. A continuación se explica la manera correcta de programarlo.

Hay que desactivar las cinco variables correspondientes a los ejes y al sentido de giro del motor: *eje\_x*, *eje\_y*, *eje\_z*, *adelante* y *atras*. La lógica del programa es la misma para ambos casos. Cada una de estas variables se activa con un conjunto de interruptores. Si ninguno de ellos está activado, se desactiva dicha variable.

Por ejemplo, la variable *eje\_y* se activa si se selecciona uno de los movimientos en el eje y. Por lo tanto, si ninguno de los interruptores correspondientes a estos dos movimientos está accionado la variable se desactiva.

```
IF (y_derecha=0 AND y_izquierda=0)
  THEN
    eje_y:=0;
  END_IF
```

Figura 6.51  
Sección de código del programa  
“manual\_desde\_maquina”

Para desactivar las variables del sentido del motor es exactamente lo mismo, si ninguno de los interruptores que las activan está accionado pasan a valer 0.

```
IF (x_adelante=0 AND y_izquierda=0 AND z_arriba=0)
  THEN
    adelante:=0;
  END_IF
```

Figura 6.52  
Sección de código del programa  
"manual\_desde\_maquina"

### Control de velocidad

La segunda parte corresponde al control de la velocidad a través de los dos pulsadores.

Como se explica en el Anexo 3, el variador de frecuencia y las salidas digitales disponibles nos permiten obtener 4 velocidades distintas del motor.

La programación está pensada para que, cada vez que se accione el pulsador superior, la velocidad pase a la siguiente más alta, y si se acciona el pulsador inferior, a la siguiente más baja. Además, para permitir mejor control y evitar que errores humanos puedan causar daños (se recuerda que es un control manual por un usuario), si la velocidad es la máxima y se acciona el pulsador superior, la velocidad pasará a ser la mínima.

Para escribir esta parte del programa se evalúa la situación de la velocidad y si se ha accionado el pulsador. A modo de ejemplo se muestra en la siguiente imagen el trozo de programa que se encarga de cambiar de la segunda velocidad a la tercera si se acciona el pulsador superior.

```
IF (speed_up=1 AND vel_1=0 AND vel_2=1 AND u=1)
  THEN
    vel_1:=1;
    vel_2:=0;
    u:=0;
  END_IF
```

Figura 6.53  
Sección de código del programa "manual\_desde\_maquina"

La variable booleana *speed\_up* vale 1 siempre que esta el pulsador activado. La variable *u* es una variable booleana auxiliar clave en esta parte de la programación.

Para entender su utilidad se expone el siguiente supuesto.

Se supone que la velocidad del motor es la mínima, y el usuario utiliza el pulsador superior, por lo que la variable *speed\_up* se activa. Esta variable vale 1 mientras éste esté pulsado. La velocidad pasaría a la siguiente de manera instantánea y, si no fuese por la variable auxiliar *u*, antes de que se soltase el pulsador también pasaría a la siguiente velocidad y a la siguiente y a la siguiente, etc.

La función de la variable auxiliar *u* es detectar cuando se deja de pulsar el botón.

Para ello, se emplea el siguiente trozo de código, que activa dicha variable cuando el pulsador no está accionado. Es por esto, que como se observa en la imagen anterior, se pone como condición necesaria para que se cambie de velocidad que  $u$  valga 1, pues implica que se ha terminado de pulsar el botón la anterior vez que se cambió de velocidad.

```
IF (speed_up=0)
  THEN
    u:=1;
  END_IF
```

Para descender de velocidad la programación es exactamente la misma cambiando las variables por las equivalentes.

Figura 6.54  
Sección de código del programa  
"manual\_desde\_maquina"

### 6.9.2.2 Desde PC

El control manual desde el PC puede realizar las mismas funciones que el control manual desde la máquina.

El programa que controla este proceso es "manual\_desde\_PC". También se puede encontrar en el Anexo I. Es exactamente igual que el programa "manual\_desde\_maquina" sustituyendo las variables que se utilizan en dicho programa por variables diferentes que se exponen a continuación. Estas variables se activan desde el panel de control, como se explicará con más detalle en el apartado "8. Panel de Control" (pág. 55).

Variables booleanas del control del movimiento:

$x\_ade$

$x\_atr$

$y\_der$

$y\_izq$

$z\_arr$

$z\_aba$

Variables booleanas del control de velocidad:

$s\_up$

$s\_down$

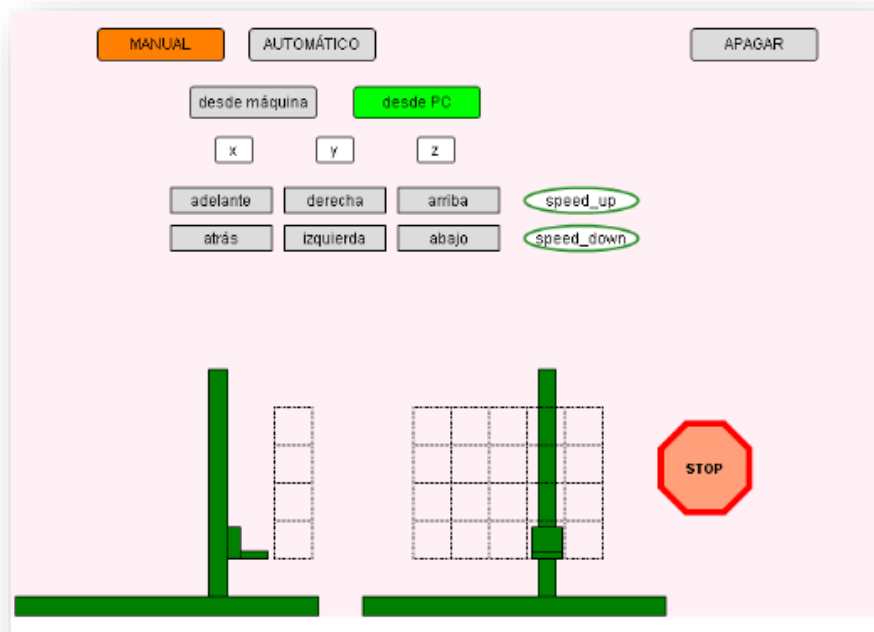


Figura 6.55  
Panel de control en el modo manual desde PC



### 6.9.3 APAGAR

El modo apagar se encarga de llevar la horquilla a una posición final.

La posición final elegida es (300, 0, 255). Esta posición queda determinada por las variables  $x_F$  e  $y_F$ . La coordenada  $z$  no necesita variable pues al ser la posición máxima del eje se controlará a través del sensor inductivo. Debido a las características vistas del control de posición de dicho es mejor así.

Se escoge este punto porque se estima que se deja la horquilla apartada, y pensando en posibles operaciones de limpieza o reparación esta posición es una de las menos molestas.

El programa que controla este proceso está escrito en lenguaje grafset, y las acciones asociadas en texto estructurado.

Para que se inicie el proceso se debe seleccionar en el panel de control el botón apagar, que activa la variable de mismo nombre. Además la variable *proceso\_finalizado* deberá valer 1.

La primera acción que se realiza es desactivar la variable *proceso\_finalizado* y activar el movimiento en el eje  $x$ . La acción “*x\_apagado*” se encarga de ello. Evalúa el valor de la variable de posición  $x$ , y la compara con  $x_F$ . Si la primera es mayor el movimiento en el eje  $x$  será negativo y si es menor positivo. Mientras la diferencia entre  $x$  y  $x_F$  sea mayor de 20 (mm) la velocidad será la máxima, de lo contrario será la mínima. Como ya se ha mencionado esto ofrece precisión en los desplazamientos.

Cuando las variables de posición, la real y la de posición final, se igualan se pasa a la siguiente etapa.

La siguiente acción que se lleva a cabo es “*y\_apagado*”. Es exactamente igual que la explicada anteriormente, sustituyendo  $x$  por  $y$  y  $x_R$  por  $y_R$ . También va a máxima velocidad cuando la posición real dista más de 20 mm de la deseada.

Cuando se alcanza la posición de  $y_R$  se sale de la etapa del grafset.

El último movimiento que se realiza es el del eje  $z$ . Lo lleva a cabo la acción “*z\_apagado*”. Debido a que se ha escogido como posición de apagado la posición máxima del eje  $z$ , esta acción solamente activa las variables que hacen subir a la horquilla. La acción se realizará hasta que se active la variable *cero\_z*.

Todas estas acciones se pueden consultar en el Anexo II.

Por último, una vez colocada la horquilla en su posición final, la variable *proceso\_terminado* se vuelve a activar y la acción “*salir\_apagado*” desactiva la variable *apagado*.

## 6.10 Detalles a Tener en Cuenta en la Programación

En las explicaciones dadas de los programas del modo automático hay dos detalles a tener en cuenta que no se han mencionado.

El primero de ellos es que después de cada etapa que lleva asociada una acción que ejecuta un movimiento se accede a otra etapa con una acción llamada “todo\_parado”. Esta acción sirve para detener el movimiento que se estaba produciendo en la etapa anterior. Las acciones activan las variables que provocan el movimiento, pero si no fuese por la acción “todo\_parado” no se desactivarían.

```
(*Todo parado*)  
  
adelante:=0;  
atras:=0;  
eje_x:=0;  
eje_y:=0;  
eje_z:=0;
```

Figura 6.56  
Acción “todo\_parado”

El segundo trata sobre las condiciones de salida de las etapas. Las condiciones para salir de una etapa que provoca el movimiento de la máquina debe ser que se haya alcanzado la posición deseada. Se programó pues, de manera inicial, condiciones del tipo salir si “x=x\_deseada”. No funciona.

Como se explica en el punto “6.4 Control de posición”, para calcular la posición real en los ejes x e y, se realiza una conversión de la señal analógica proporcionada por los potenciómetros. Al convertir dicho valor, se obtienen resultados de tipo real. Por lo tanto es posible que, por ejemplo x, nunca llegue a valer exactamente x\_deseada.

La solución es utilizar intervalos en las condiciones. A continuación se muestra como ejemplo la transición “y\_igual\_yDi\_cond”. Se encuentra en el programa “Modo\_3” y se tiene que cumplir para salir de la etapa que contiene la acción “y\_Di”:

```
y_igual_yDi_cond:=(y>=yDi-2.5 AND y<=yDi+2.5);
```

Figura 6.57  
Transición “y\_igual\_yDi\_cond”

## 7. CONTROL POR TAREAS

Para que los programas sean ejecutados deben de ser llamados a través de tareas. Las tareas permiten establecer prioridad y configurar el tipo de ejecución que se desea para los programas que contienen.

En este proyecto se definen 4 tareas. Todas ellas se ejecutan de modo cíclico con un intervalo de 20 ms. Las tareas, con los programas que contienen, se presentan a continuación:

### *variable\_z*

Esta tarea solo contiene el programa “variable\_posicion\_z”. Es el único programa de esta tarea debido a su importancia. Como ya se ha dicho en repetidas ocasiones, el principal problema del funcionamiento de todo el programa y de la máquina es el control de la coordenada z. Es por ello que el programa que la controla está en una tarea con la prioridad más alta, 2 de 31.

### *Mast*

Es la tarea principal. Contiene los programas “zona\_seguridad”, “parada\_emergencia”, “senal\_anal\_a\_real” y “estado\_casillero”. Los dos primeros programas son de seguridad. Necesitan tener prioridad sobre cualquier programa que se encargue de accionar la máquina. La prioridad de esta tarea es 10 sobre 31.

### *Modo\_Seleccion*

Solamente contiene el programa “Modo\_Seleccion”. Este programa controla todos los programas que ejecutan los distintos modos de funcionamiento, por esto debe de priorizar sobre ellos. Su prioridad es 15 de 31.

### *modos*

En esta tarea se incluyen los programas que ejecutan los modos de funcionamiento. Contiene los programas “Apagado”, “Modo\_1”, “Modo\_2”, “Modo\_3”, “Modo\_4” y “Modo\_5”.

Tiene 20 sobre 31 de prioridad.

## 8. PANEL DE CONTROL

El programa SoMachine también permite realizar un panel de control SCADA.

A continuación se presenta la visualización y se explica la función de cada parte.

### 8.1 Panel de Control Inicial

Apretando el botón MANUAL o AUTOMÁTICO se activa la variable *manual* o *automatico* y se accede a los paneles característicos de cada modo. Con el botón APAGAR se activa la variable *apagar* y se ejecuta el programa correspondiente. Si se aprietan estos botones mientras están activos, la variable correspondiente pasa a valer 0.

El botón STOP siempre estará visible. Al apretarlo se activa la variable *paro*, y el programa "parada\_emergencia" detiene todos los procesos. Si se vuelve a pulsar la variable se desactiva y todo vuelve a estar como antes de pulsarlo.

Todos estos botones cambian de estado cuando su variable asociada vale 1.

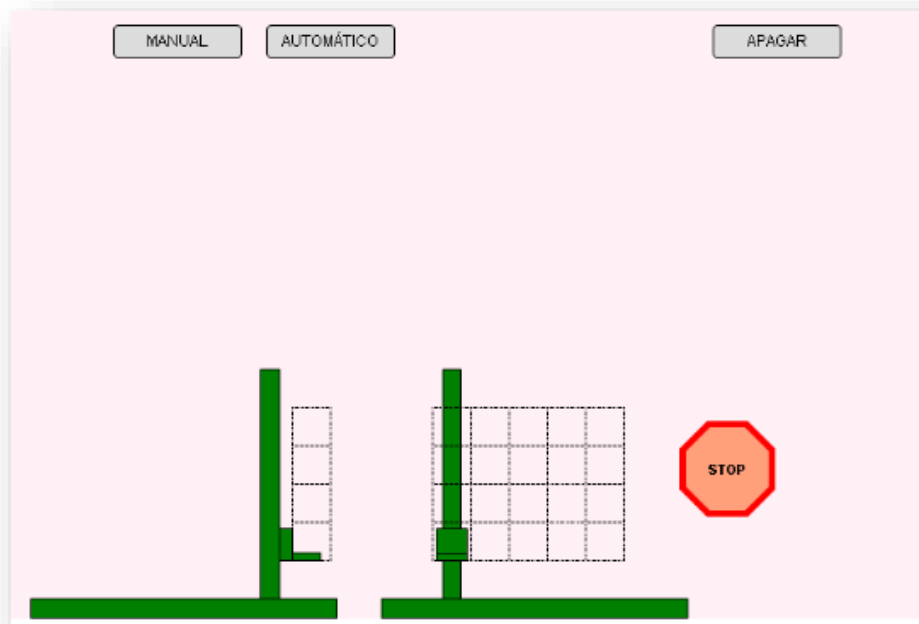


Figura 8.1  
Vista inicial Panel de Control

Las figuras de la parte inferior representan a la máquina. A la izquierda de la imagen se encuentra la representación del perfil izquierdo, en la cual se aprecia los movimientos en el eje x y en el eje z. A la derecha, la representación de la vista frontal, donde se aprecian los movimientos en los ejes y z.

Las dos barras horizontales representan la parte fija inferior de la máquina. Las barras verticales el soporte de la horquilla y las piezas más pequeñas la propia horquilla. Las cuadrículas representan la zona de depósito y entrega.

Para dotar a las figuras de movimiento se recurre de la opción “movimiento\_absoluto, que permite establecer el desplazamiento deseado para una figura en función de una variable.

De esta manera, con la conversión apropiada, se consigue una representación exacta de los movimientos que se estén llevando a cabo en la máquina. Se muestran a continuación una serie de imágenes con la posición real en el pie de foto. La primera corresponde a los ceros de los tres ejes, la segunda corresponde a la posición determinada por  $(xD_, yDi\_c3, zDi\_f2)$  y la última a los valores máximos.

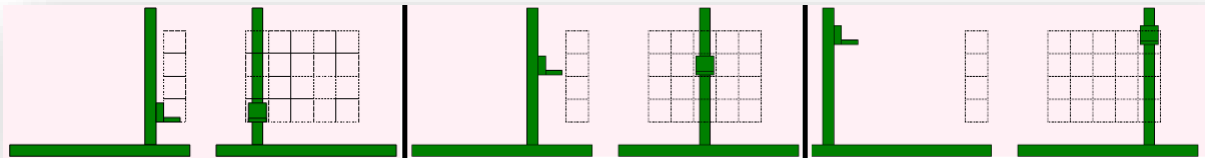


Figura 8.2  
(0, 0, 0)

Figura 8.3  
(50, 135, 160)

Figura 8.4  
(305, 270, 255)

## 8.2 Automático

Al pulsar en el botón automático aparece el siguiente panel:

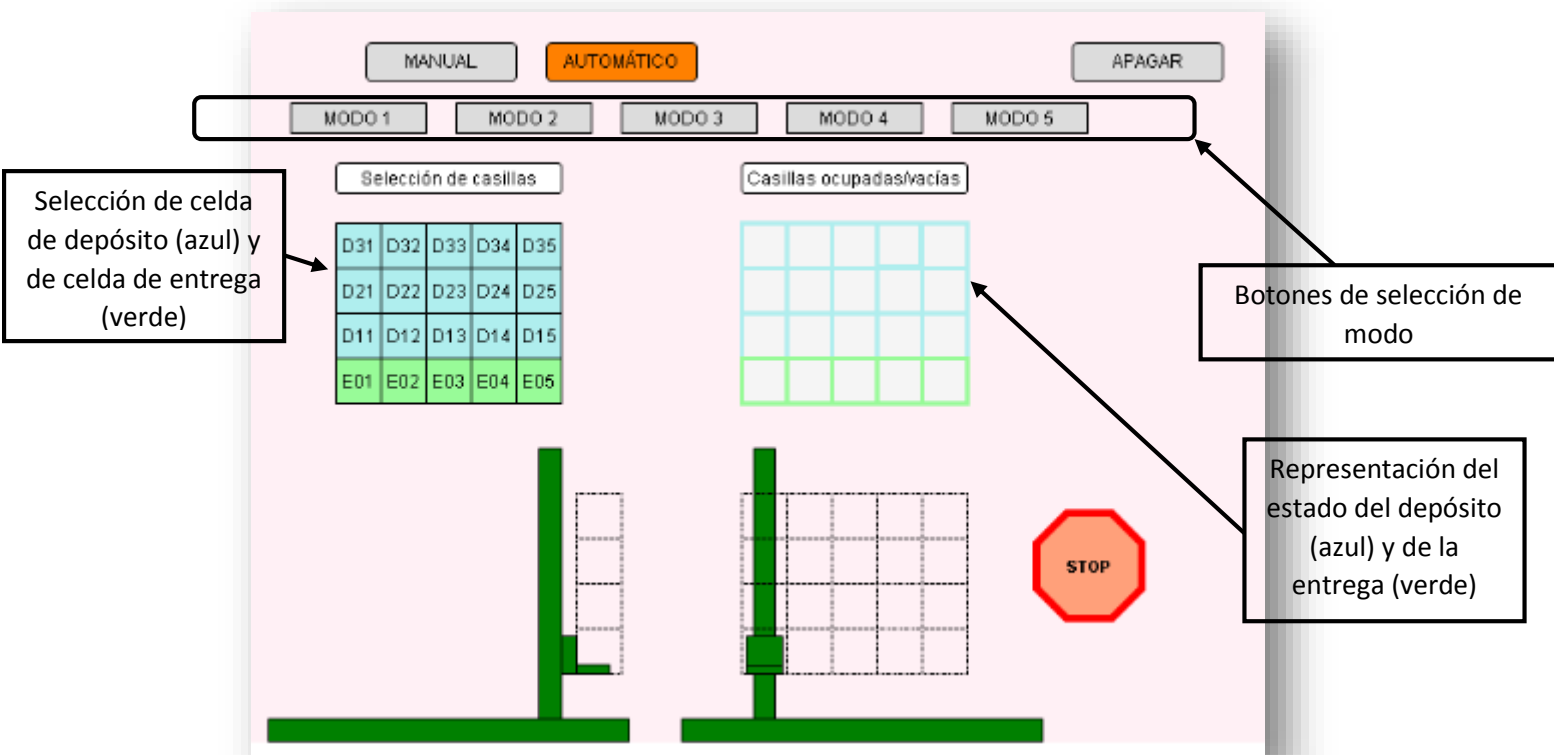


Figura 8.5  
Modo Automático

Selección de depósito: cada celda lleva asociada la variable con su mismo nombre. Al seleccionar una celda esta variable se activa y el color de la visualización pasa a ser oscuro. En imágenes posteriores se podrá apreciar el cambio de color.

Estado del depósito: cada celda se asocia con su variable correspondiente de estado (*obj\_Eij* y *obj\_Dij*). Cuando una de estas variables vale 0, la casilla a la que está asociada no tiene relleno (como en la imagen anterior), pero si vale 1 la casilla aparece rellena del mismo color que el contorno. En imágenes posteriores se podrá apreciar el cambio.

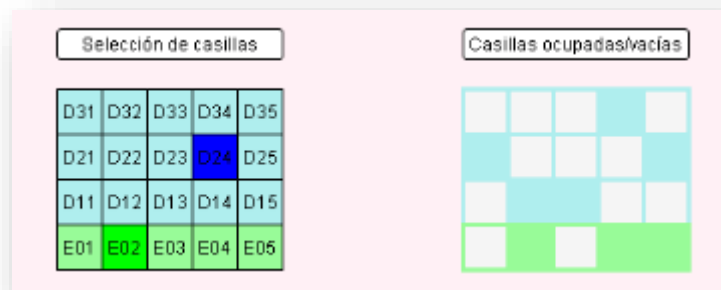


Figura 8.6

Celdas D24 y E02 seleccionadas. Celdas del depósito D22, D23, D21, D25 y D34; y celdas de la entrega E02, E04 y E05 ocupadas.

Selección de modo: los botones de selección de modo llevan asociadas las variables *modo\_1*, *modo\_2*, etc. Al accionarlos se activan dichas variables y al volverlos a accionar se desactivan. También cambian de estado cuando su variable asociada vale 1.

Una vez se entra en algún modo de funcionamiento, aparece el botón GO. Después de haber hecho las selecciones necesarias hay que pulsar sobre él para que se empiece a ejecutar el modo. Lleva asociada la variable *go*, que se activará solamente mientras se esté pulsando el botón.



Figura 8.7  
Botón GO

Como ya se ha explicado, en los modos de funcionamiento 1 y 2 es necesario pulsar el botón PIEZA COLOCADA para que continúen los procesos. Este botón aparecerá cuando la horquilla llegue a la zona de recogida. Su variable asociada es *pieza\_colocada*, que solo valdrá 1 mientras se esté pulsando. Además, al pulsarlo el proceso continúa y el botón desaparece de nuevo.



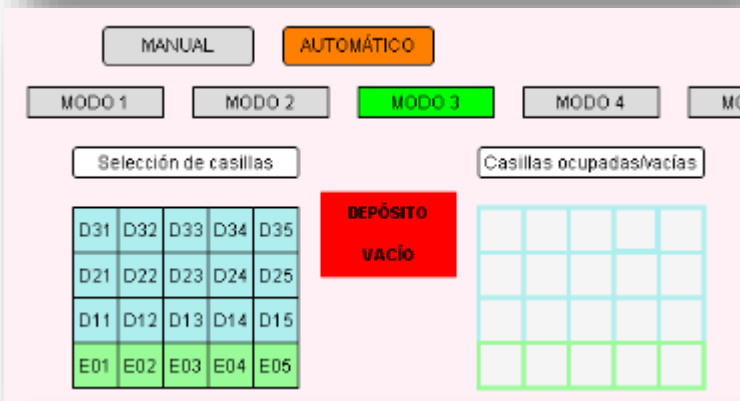
Figura 8.8  
Botón PIEZA COLOCADA

Por último, se ha programado el panel de control para que aparezcan una serie de carteles informativos si se dan las condiciones necesarias.



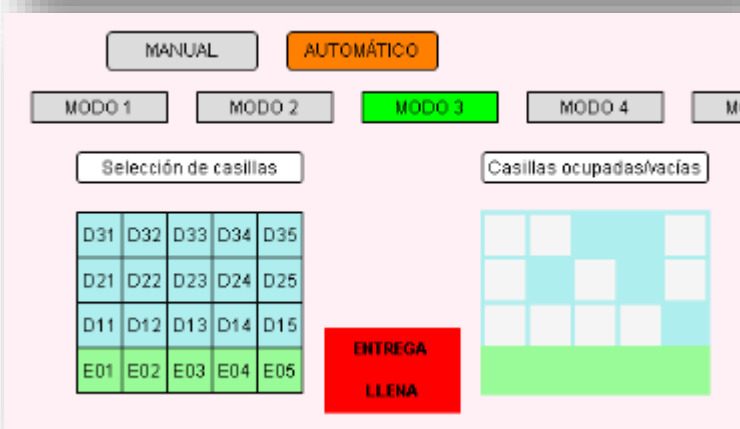
### DEPÓSITO LLENO

Este cartel solo aparece en los modos 1, 2 y 5 cuando la variable *deposito\_lleno* vale 1. Indica que no se puede realizar el modo, pues no hay espacio en el depósito.



### DEPÓSITO VACÍO

Este cartel solo aparece en los modos 3, 4 y 5 cuando la variable *deposito\_vacio* vale 1. Indica que no se puede realizar el modo, pues no hay piezas disponibles en el depósito.



### ENTREGA LLENA

Este cartel solo aparece en los modos 3 y 4 cuando la variable *entrega\_llena* vale 1. Indica que no se puede realizar el modo, pues no hay espacio en la entrega.

Figuras 8.9, 8.10 y 8.11

Zona superior izquierda del panel de control en distintas situaciones

### PROCESO EN MARCHA

Este cartel aparece cuando la variable *proceso\_finalizado* vale 0. Informa de que se está llevando a cabo un proceso.

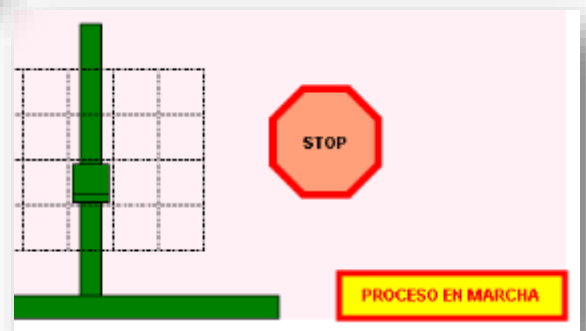


Figura 8.12

Zona inferior izquierda del Panel de control mientras se está ejecutando un proceso

### 8.3 Manual

Al pulsar MANUAL, en el panel de control aparecen dos botones nuevos. Estos botones, desde máquina y desde PC, dan acceso a las dos modalidades de control manual. Llevan asociadas las variables *desde\_maquina* y *desde\_PC*, que cambian de valor cada vez que se aprieta el botón.

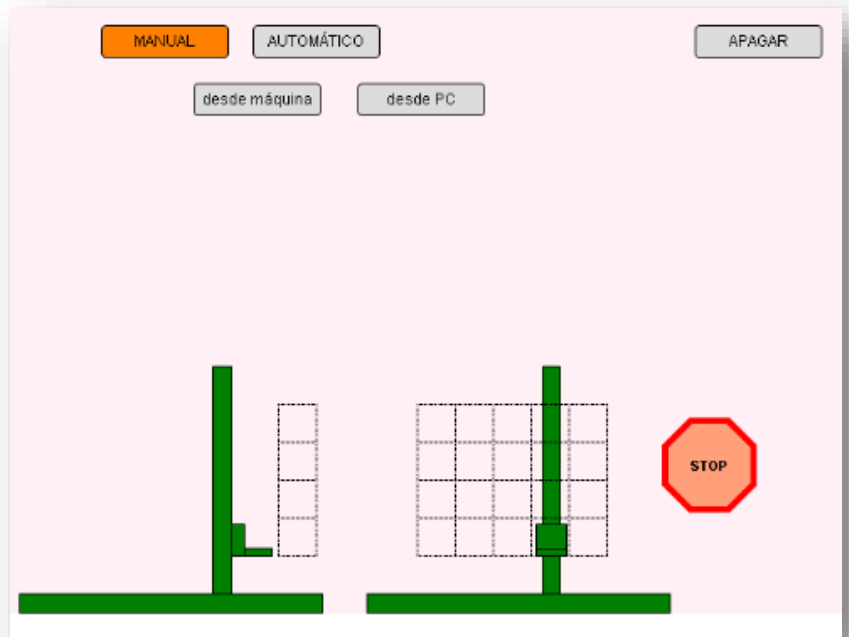


Figura 8.13

Panel de control al accionar el botón MANUAL

#### Desde máquina

Si se selecciona la opción desde máquina, en el panel de control aparece el cartel que aparece en la imagen siguiente. Es meramente informativo y solamente aparece cuando la variable *desde\_maquina* vale 1.

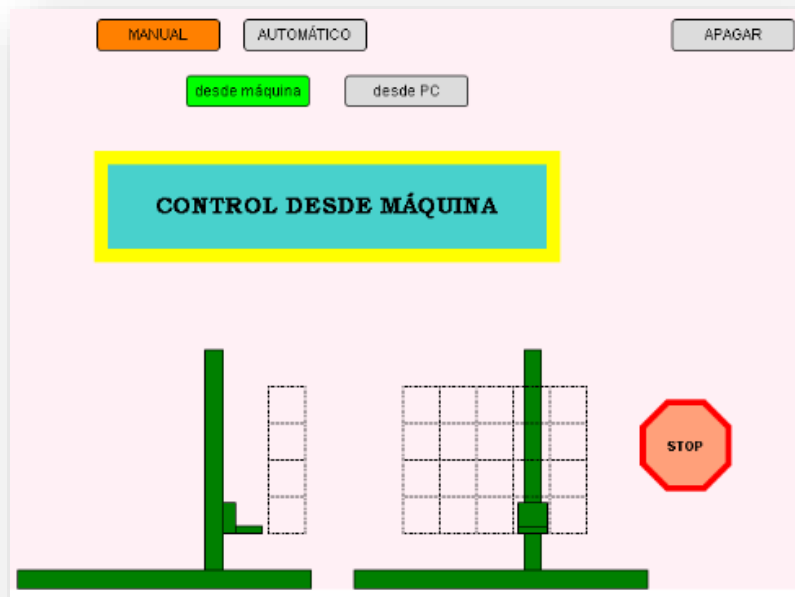


Figura 8.14

Panel de control en el modo manual DESDE MÁQUINA



### Desde PC

Si se selecciona la opción desde PC, en el panel de control aparecen los botones para manejar la máquina. Los botones están separados en cuatro columnas. Las tres primeras según los ejes y última es para los controles de velocidad.

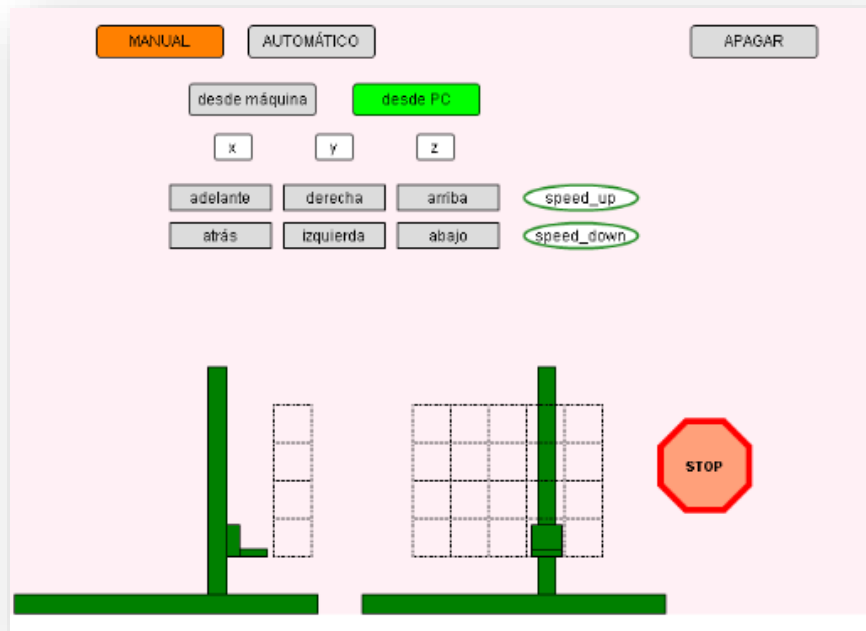


Figura 8.15

Panel de control en el modo manual DESDE PC

Cada botón activa alguna de las variables utilizadas el modo manual desde PC ( $x_{ade}$ ,  $x_{atr}$ ,  $y_{der}$ ,  $y_{izq}$ ,  $z_{arr}$ ,  $z_{aba}$ ,  $s_{up}$  y  $s_{down}$ ). Se ha diseñado para que estas variables solo estén activas mientras el botón correspondiente esté pulsado.

Cuando uno de estos botones se está pulsando cambia de color.



Figura 8.16

Panel de control en el modo manual DESDE PC. Botones arriba y speed\_down pulsados.

### 8.4 Apagar

El único cambio producido al pulsar apagar es que cambia el estado del botón. Por supuesto, la representación de la máquina se moverá hacia la posición de apagado al igual que la máquina real.



Figura 8.17  
Apagar activo



## 9. CONCLUSIÓN

El trabajo realizado ha alcanzado los objetivos más ambiciosos que se habían marcado. Se ha conseguido darle un uso muy amplio a la máquina de tres ejes. Dos modos manuales, cinco modos automáticos, paro de emergencia desde el panel de control y una posición final al apagar la máquina. Además, se han presentado problemas importantes, tanto de recursos como por las características de la máquina, y las soluciones realizadas han sido muy satisfactorias.

Y no solo la programación, el resultado final del panel SCADA ha sido incluso mejor de lo que se pretendía hacer en un principio. Mientras se diseñaba, y mientras se programaban los procesos, surgían ideas que incorporar al panel de control. El resultado ha sido un control sencillo, dinámico y bonito.

La representación de la máquina en el panel de control es muy útil. Se han realizado pruebas en las que se controlaba la máquina sin mirarla, mirando solamente el panel de control, y los resultados fueron muy buenos. La sencillez del panel de control también es algo destacable. Los botones aparecen cuando se necesitan y desaparecen cuando no. Gracias a esto nunca se ve sobrecargado difícil de utilizar. Por último, los carteles informativos facilitan el control a los operarios.

A nivel personal, este proyecto me ha hecho trabajar de una manera distinta a la hecha hasta ahora. A lo largo de este proyecto he adquirido diversas aptitudes que estoy seguro que me servirán de mucho en mi futuro. Por último, me enorgullece el esfuerzo realizado y, sobretodo, los resultados obtenidos.



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA



# PRESUPUESTO

Proyecto de Automatización y Control mediante  
SCADA de un Sistema de Almacenamiento y  
Recuperación Industrial de Piezas



**JOSE CABOT ALMELA**

Tutor: Juan Pérez Cruz

Cotutor: Manuel Pineda Sánchez



## ÍNDICE PRESUPUESTO

<b>1. INTRODUCCIÓN.....</b>	<b>64</b>
<b>2. COSTES MATERIALES .....</b>	<b>65</b>
2.1 Circuitos Acondicionadores de Señal de Potenciómetros.....	65
Para que el autómata sea capaz de leer los datos proporcionados por los potenciómetros, se ha de amplificar la señal que estos crean. Ésta es la función de estos circuitos. A continuación se presentan sus elementos: .....	65
2.2 Elementos de Control.....	65
2.3 Software .....	66
2.4 Componentes de la Máquina .....	66
2.5 Elementos de Seguridad.....	67
2.6 Coste Materiales Total .....	67
<b>3. COSTE MANO DE OBRA .....</b>	<b>68</b>
<b>4. COSTES GENERALES.....</b>	<b>69</b>
<b>5. MARGEN DE BENEFICIO.....</b>	<b>69</b>
<b>6. COSTE FINAL DEL PROYECTO .....</b>	<b>69</b>



## 1. INTRODUCCIÓN

En este documento se recogen los costes del proyecto. Se hace un análisis económico de todas las partes que han intervenido a lo largo del trabajo.

Los costes se van a analizar en este documento según su naturaleza. Así, se diferencian costes de material, costes de mano de obra y costes generales. Este presupuesto es parte de un trabajo académico, por lo que no se van a contabilizar con detalle gastos generales como pudieran ser la luz, el agua, etc.

Tras el análisis de cada uno de ellos se aplicará el margen de beneficio. Por último, solo quedará añadir los impuestos para obtener el coste final del proyecto.

Este documento se limita al análisis del coste económico de la ejecución del trabajo. No se abarcan los campos del estudio previo de mercado ni del análisis de las repercusiones económicas. Aun así, cabe decir que estos estudios previos sí se hubiesen realizado en la empresa en la que se lleve a cabo el trabajo y, a partir de ellos, se hubiesen obtenido unas estimaciones de los resultados económicos finales de la aplicación del proyecto. Con estos datos y con el coste final del proyecto se obtendría el beneficio o ahorro final. Por último, en función de éste, habría que discutir la viabilidad o no de aplicar el trabajo.

Se va a suponer que el encargo por parte de la empresa es automatizar y controlar la máquina que ellos ya disponen. Por tanto, en el presupuesto se incluyen todos los elementos necesarios para dicho propósito. Esto incluye los sensores que hay que instalar en la máquina, los elementos necesarios para la interpretación de los resultados de los sensores, los distintos elementos de control y seguridad, el autómatas y el ordenador.

## 2. COSTES MATERIALES

En este apartado se presentan los diferentes elementos utilizados para la ejecución del proyecto. Se tiene en cuenta el precio de venta al público.

### 2.1 Circuitos Acondicionadores de Señal de Potenciómetros

Para que el autómatas sea capaz de leer los datos proporcionados por los potenciómetros, se ha de amplificar la señal que estos crean. Ésta es la función de estos circuitos. A continuación se presentan sus elementos:

ELEMENTO	CANTIDAD	PRECIO	IMPORTE
Transistor BC 309	2	0.12€	0.24€
Resistencia 1.5k	2	0.05€	0.10€
Resistencia 2.2k	2	0.05€	0.10€
Resistencia 10k	2	0.05€	0.10€
Placa de montaje	1	0.50€	0.50€
<b>SUBTOTAL</b>			<b>1.04€</b>

Tabla 1

### 2.2 Elementos de Control

El control de la ejecución de los distintos procesos se efectúa mediante los elementos que se exponen a continuación.

ELEMENTO	CANTIDAD	PRECIO	IMPORTE
PLC Modicon MT258LF66DT4LSO de Schneider Electric	1	985.35€	985.35€
Magelis HMISTU	1	524.39€	524.39€
Variador de frecuencia	1	301.20€	301.20€
Ordenador (con periféricos)		467.30€	467.30€
Contactores	3	14.50€	43.50€
<b>SUBTOTAL</b>			<b>2321.74€</b>

Tabla 2



### 2.3 Software

En este apartado se abarca el coste de adquisición de las licencias de los programas utilizados. En nuestro proyecto se ha utilizado el programa SoMachine, un software libre proporcionado por Schneider Electric disponible en su página web.

ELEMENTO	CANTIDAD	PRECIO	IMPORTE
SoMachine	1	0€	0€
<b>SUBTOTAL</b>			<b>0€</b>

Tabla 3

### 2.4 Componentes de la Máquina

Se incluyen en este apartado los elementos que es necesario acoplar a la máquina para su control.

ELEMENTO	CANTIDAD	PRECIO	IMPORTE
Sensores inductivos	3	35.32€	105.93€
Potenciómetros	2	500€	1000€
Pulsadores	2	3.09€	6.18€
Botonera	1	15.23€	15.23€
Regletas de conexión	15	0.06€	0.90€
Manguera 4x1.5mm <sup>2</sup>	10m	0.99€/m	9.90€
Manguera 4x2.5mm <sup>2</sup>	10m	1.05€/m	10.50€
Cable flexible 1.5mm <sup>2</sup>	5m	0.15€/m	0.75€
Otros			8.20€
<b>SUBTOTAL</b>			<b>1157.59</b>

Tabla 4

## 2.5 Elementos de Seguridad

Estos elementos son los encargados de proteger a la máquina y usuarios de posibles accidentes.

ELEMENTO	CANTIDAD	PRECIO	IMPORTE
Pulsador de emergencia	1	5.21€	5.21€
Interruptor final de carrera	6	2.13€	12.78€
Interruptor magnetotérmico	1	25.31€	25.31€
<b>SUBTOTAL</b>			<b>43.30€</b>

Tabla 5

## 2.6 Coste Materiales Total

ELEMENTO	IMPORTE
Circuitos Acondicionadores	1.04€
Elementos de control	2321.74€
Software	0€
Componentes de la máquina	1157.59€
Elementos de seguridad	43.30€
<b>SUBTOTAL</b>	<b>3523.67€</b>

Tabla 6





### 3. COSTE MANO DE OBRA

Para calcular el coste de la mano de obra se divide el tiempo dedicado al trabajo en tareas. Se considera que el trabajador es el autor de este proyecto con la titulación obtenida tras finalizarlo. Por tanto se considera un trabajador con la titulación de graduado en ingeniería industrial. Según el convenio del Metal, para ingenieros técnicos industriales el precio por hora 13.68€ por hora.

ELEMENTO	HORAS	IMPORTE
Estudio previo	30	410.40€
Puesta a punto	20	273.60€
Programación	150	2052.00€
Diseño del control	70	957.60€
Ensayos y comprobaciones	30	410.40€
Ajustes generales	20	273.60€
<b>SUBTOTAL</b>		<b>4377.60€</b>

Tabla 7

#### 4. COSTES GENERALES

Este apartado incluye los gastos comunes a todos los proyectos que se desarrollan en una empresa. La luz, el agua, las instalaciones, teléfono e internet, impuestos generales, etc... Para incluirlo en el presupuesto se aplica un porcentaje sobre los costes materiales y de mano de obra. El porcentaje varía mucho en función de cada empresa.

Se estima un porcentaje del 15% sobre los costes vistos anteriormente.

Costes de material más costes de mano de obra:

$$4377.60\text{€} + 3523.67\text{€} = 7901.27\text{€}$$

*Ecuación 6*

Costes de fabricación tras aplicar el porcentaje de gastos generales:

$$7901.27\text{€} * 1.15 = 9086.46\text{€}$$

*Ecuación 7*

#### 5. MARGEN DE BENEFICIO

El beneficio industrial es un porcentaje aplicado sobre el coste total de fabricación del proyecto. Representa la parte que se gana la empresa. Depende del tipo de industria y de la empresa. En este caso, se va a aplicar un porcentaje de un 12%.

$$9086.46\text{€} * 1.12 = 10176.84\text{€}$$

*Ecuación 8*

#### 6. COSTE FINAL DEL PROYECTO

Por último se debe de aplicar el impuesto correspondiente. Se le aplica el IVA, que actualmente vale un 21%, y se obtiene el coste final del proyecto.

$$10176.84\text{€} * 1.21 = 12313.97\text{€}$$

*Ecuación 9*

**PRESUPUESTO FINAL DEL PROYECTO: 12313.97€**

**Doce mil trescientos trece euros y noventa y siete céntimos.**



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA



# PLIEGO DE CONDICIONES

Proyecto de Automatización y Control mediante  
SCADA de un Sistema de Almacenamiento y  
Recuperación Industrial de Piezas



**JOSE CABOT ALMELA**

Tutor: Juan Pérez Cruz

Cotutor: Manuel Pineda Sánchez



## ÍNDICE PLIEGO DE CONDICIONES

<b>1. INTRODUCCIÓN.....</b>	<b>72</b>
<b>2. CONDICIONES DE ÍNDOLE TÉCNICA .....</b>	<b>73</b>
CAPÍTULO I: CONDICIONES QUE DEBEN REUNIR LOS MATERIALES.....	73
CAPÍTULO II: CONTROL DEL FUNCIONAMIENTO.....	74
<b>3. CONDICIONES DE ÍNDOLE FACULTATIVA .....</b>	<b>75</b>
CAPÍTULO I: RESPONSABILIDADES DEL CONTRATISTA.....	75
CAPÍTULO II: RESPONSABILIDADES DEL CONTRATANTE .....	75
<b>4. CONDICIONES DE ÍNDOLE ECONÓMICA .....</b>	<b>76</b>
CAPÍTULO I: BASE FUNDAMENTAL.....	76
CAPÍTULO II: PRECIOS Y RECARGOS .....	76
<b>5. CONDICIONES DE ÍNDOLE LEGAL .....</b>	<b>77</b>
CAPÍTULO I: EL CONTRATO.....	77
CAPÍTULO II: RESCISIÓN DEL CONTRATO .....	77



## 1. INTRODUCCIÓN

Este documento recoge los detalles a cumplir por todas las partes intervinientes en el proyecto.

Se abarcará diferenciando cuatro apartados: condiciones técnicas, condiciones facultativas, condiciones económicas y condiciones legales.

Condiciones técnicas: incluyen las características de los materiales y los procedimientos a emplear a lo largo de la realización del trabajo.

Condiciones facultativas: recogen los derechos y obligaciones de las partes.

Condiciones económicas: se presentan las formas de pago e indemnizaciones.

Condiciones legales: hacen referencia al perfil de contratista, la forma de adjudicación, el tipo de contrato, la obligatoriedad de suscripción de seguros de responsabilidad civil y otros asuntos relacionados.

## 2. CONDICIONES DE ÍNDOLE TÉCNICA

### CAPÍTULO I: CONDICIONES QUE DEBEN REUNIR LOS MATERIALES

#### NORMATIVA GENERAL.

Las instalaciones requeridas para el presente proyecto son en su mayor parte componentes eléctricos. Están conectados a la red de corriente alterna de 220V y 50 Hz, por lo que se rigen por el Reglamento Electrotécnico de Baja Tensión (RBT) y sus Instrucciones Complementarias.

También se deben considerar las siguientes normas UNE y DIN:

ITC-BT-18: instalaciones de puesta a tierra.

ITC-BT-19: instalaciones interiores o receptoras. Prescripciones generales.

ITC-BT-20: instalaciones interiores o receptoras. Sistemas de instalación.

ITC-BT-22: instalaciones interiores o receptoras. Protección contra sobreintensidades.

ITC-BT-23: instalaciones interiores o receptoras. Protección contra sobretensiones.

ITC-BT-24: instalaciones interiores o receptoras. Protección contra contactos directos e indirectos.

ITC-BT-43: instalación de receptores. Prescripciones generales.

ITC-BT-47: instalación de receptores. Motores.

ITC-BT-51: instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad.

UNE 20 514 1M: reglas de seguridad para aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uno general análogo conectado a una red de energía.

#### ORDENADOR PERSONAL

El ordenador dispondrá como mínimo de los siguientes elementos: CPU, pantalla, ratón, teclado, conexión Ethernet y entrada de puerto USB.

Además las características técnicas mínimas serán: 2 procesadores COREi3, memoria RAM de 2GB, espacio libre en el disco duro de 50Gb y sistema operativo Windows Xp.

La temperatura de trabajo deberá estar 0°C y 60°C para su correcto funcionamiento.

#### EL AUTOMATA PROGRAMABLE

Se utilizará el PLC M258 de Schneider Electric compatible con SoMachine. El autómata programable estará alimentado a través de una fuente de alimentación de corriente continua de 24V conectada a la red.

Debe cumplir las normas DIN VDE 0160.



## **EL VARIADOR DE FRECUENCIA**

Debe cumplir las normas de seguridad de uso:

No conectarlo a la red mucho tiempo largo si no se encuentra conectado a un motor y no descubrir la carcasa protectora mientras esté en funcionamiento.

Además debe cumplir las normas DIN VDE 0160 y la norma NEMA IP21.

## **CAPÍTULO II: CONTROL DEL FUNCIONAMIENTO**

### **AUTOMATIZACIÓN. MOVIMIENTOS**

Antes de hacer pruebas con la automatización terminada, se deberá realizar la comprobación de las respuestas de la máquina. Se comprobará que responde como se espera, y se harán las modificaciones necesarias en función de los resultados.

### **CONTROL**

El siguiente paso será la comprobación del panel de control en el modo de simulación. Una vez comprobado que todas las partes están bien vinculadas, que los accesos están bien definidos y que el control se puede realizar sin problemas se podrá proceder a las pruebas finales.

### **PRUEBAS FINALES**

Se realizarán pruebas de todos los modos de funcionamiento, desde distintas situaciones originales y en distinto orden. Cuando se superen estas pruebas el proyecto se dará por finalizado.



### **3. CONDICIONES DE ÍNDOLE FACULTATIVA**

#### **CAPÍTULO I: RESPONSABILIDADES DEL CONTRATISTA**

Debe estar informado y actuar según las leyes y normas que rigen su actividad profesional.

Debe conocer las especificaciones técnicas y normas de seguridad aplicables a los elementos del proyecto. Se deberá cumplir la normativa establecida en el Reglamento Electrotécnico de Baja Tensión.

Tiene la responsabilidad de comprobar que los elementos utilizados cumplan los requisitos recogidos en el proyecto.

Es su función realizar las comprobaciones especificadas en el apartado anterior y su responsabilidad el dar el visto bueno o no.

#### **CAPÍTULO II: RESPONSABILIDADES DEL CONTRATANTE**

Deberá facilitar todos los elementos e información de que disponga que estén relacionados con el proyecto al Contratista.

Debe también entregar por escrito las especificaciones del proyecto que desea que se lleven a cabo.

Es su responsabilidad obtener los permisos obligatorios para la realización del proyecto.

No podrá reclamar por retrasos en el proyecto causados por motivos ajenos al contratista. Estos motivos deberán de poderse justificar. En el caso de que se produzcan retrasos y no se justifiquen se producirán reducciones en el precio del proyecto según lo establecido en el siguiente apartado.



## 4. CONDICIONES DE ÍNDOLE ECONÓMICA

### CAPÍTULO I: BASE FUNDAMENTAL

El Contratista tiene derecho a cobrar por el trabajo realizado, siempre que se haya atendido a lo estipulado en el proyecto.

### CAPÍTULO II: PRECIOS Y RECARGOS

El precio de ejecución del proyecto se establece en Presupuesto.

Si se producen retrasos en el pago desde la conclusión del proyecto se cargarán recargos en función de lo siguiente:

Entre 1 y 7 días	–	Sin recargo
Entre 8 y 15 días	–	1% de recargo
Entre 16 y 25 días	–	2% de recargo
Entre 26 y 50 días	–	4% de recargo
Entre 51 y 100 días	–	8% de recargo
Entre 101 y 200 días	–	20% de recargo
Entre 201 y 300 días	–	35% de recargo

En caso de impago tras 300 días naturales desde la conclusión del proyecto el Contratista tiene derecho a realizar una demanda ante los tribunales.

Si se producen retrasos en la finalización del proyecto según los plazos establecidos por motivos que no estén justificados como ajenos al Contratista, se reducirá el precio en función de lo siguiente:

Entre 1 y 7 días	–	Sin reducción
Entre 8 y 15 días	–	2% de reducción
Entre 16 y 25 días	–	3% de reducción
Entre 26 y 50 días	–	5% de reducción
Entre 51 y 100 días	–	15% de reducción

En caso de no haber finalizado el proyecto después de 101 según los plazos acordados, siempre que los retrasos no sean por causas justificadas ajenas al Contratista, el contratante tendrá derecho a rescindir el contrato.

Nótese que se refiere en todo caso a días naturales y consecutivos.



## 5. CONDICIONES DE ÍNDOLE LEGAL

### CAPÍTULO I: EL CONTRATO

El contrato recogerá el precio final del proyecto recogido en el presupuesto junto con las cláusulas vistas en el apartado anterior en caso de retrasos del pago.

Se realizará por escrito y deberá ser firmado por todas las partes implicadas.

También recogerá todas las cláusulas que se negocien entre las partes.

### CAPÍTULO II: RESCISIÓN DEL CONTRATO

Para la posible rescisión del contrato se deberá de dar una de las siguientes condiciones:

- Acuerdo entre ambas partes para rescindir el contrato
- Modificación de los aspectos fundamentales del proyecto acordados
- Incumplimiento de los plazos establecidos
- Impago
- Actuaciones con mala fe

Por supuesto, la decisión de rescindir el contrato si se da una de estas condiciones la tendrá la parte perjudicada.



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA



# ANEXO I. PROGRAMAS

Proyecto de Automatización y Control mediante  
SCADA de un Sistema de Almacenamiento y  
Recuperación Industrial de Piezas



**JOSE CABOT ALMELA**

Tutor: Juan Pérez Cruz

Cotutor: Manuel Pineda Sánchez



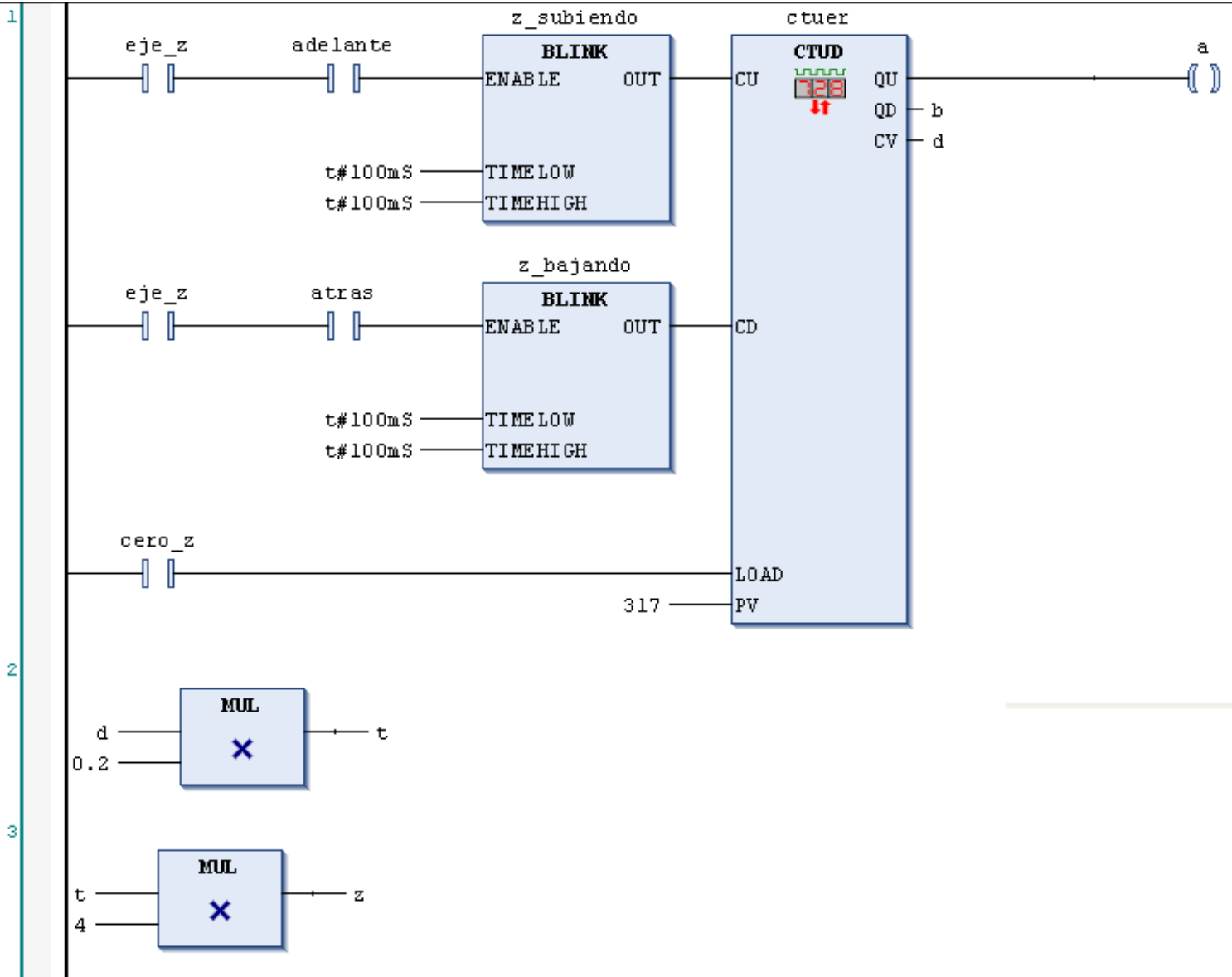
## ÍNDICE ANEXO I

1. senal_anal_a_real .....	80
2. variable_posicion_z.....	81
3. zona_seguridad .....	82
4. parada_emergencia .....	83
5. estado_casillero .....	84
6. Modo_Selección.....	85
7. Modo_1 .....	88
8. Modo_2 .....	89
9. Modo_3 .....	90
10. Modo_4 .....	91
11. Modo_5 .....	92
12. manual_desde_maquina.....	93
13. manual_desde_PC.....	94

## 1. senal\_anal\_a\_real

```
1  IF (cero_x=1)
2      THEN
3      x:=0;
4  END_IF
5
6  IF (cero_x=0)
7      THEN
8      x:=374.06-0.0251*xanal;
9  END_IF
10
11 IF (cero_y=1)
12     THEN
13     y:=0;
14 END_IF
15
16 IF (cero_y=0)
17     THEN
18     y:=413.83-0.0191*yanal;
19 END_IF
```

## 2. variable\_posicion\_z



### 3. zona\_seguridad

```
1  IF (ceros_x=1 AND adelante=1)
2      THEN
3          eje_x:=0;
4  END_IF
5
6  IF (x>=305 AND atras=1)
7      THEN
8          eje_x:=0;
9  END_IF
10
11 IF (x<=30 AND adelante=1 AND eje_x=1)
12     THEN
13         vel_1:=0;
14         vel_2:=0;
15 END_IF
16
17 IF (atras=1 AND x>=275 AND eje_x=1)
18     THEN
19         vel_1:=0;
20         vel_2:=0;
21 END_IF
22
23 IF (ceros_y=1 AND adelante=1)
24     THEN
25         eje_y:=0;
26 END_IF
27
28 IF (y>=270 AND atras=1)
29     THEN
30         eje_y:=0;
31 END_IF
32
33 IF (y<=30 AND adelante=1 AND eje_y=1)
34     THEN
35         vel_1:=0;
36         vel_2:=0;
37 END_IF
38
39 IF (atras=1 AND y>=240 AND eje_y=1)
40     THEN
41         vel_1:=0;
42         vel_2:=0;
43 END_IF
44
45 IF ((ceros_z=1 AND adelante=1) OR (z=0 AND atras=1))
46     THEN
47         eje_z:=0;
48 END_IF
49
50 IF (z<=0 AND atras=1)
51     THEN
52         eje_z:=0;
53 END_IF
54
55 (*Velocidad siempre lenta en el eje z*)
56
57 IF (eje_z=1)
58     THEN
59         vel_1:=0;
60         vel_2:=0;
61 END_IF
```

#### 4. parada\_emergencia

```
1  IF (paro=0)
2      THEN
3          a:=eje_x;
4          b:=eje_y;
5          c:=eje_z;
6          d:=1;
7  END_IF
8
9  IF (paro=1)
10     THEN
11         eje_x:=0;
12         eje_y:=0;
13         eje_z:=0;
14     END_IF
15
16     IF (paro=0 AND d=1)
17         THEN
18             eje_x:=a;
19             eje_y:=b;
20             eje_z:=c;
21             d:=0;
22     END_IF
```



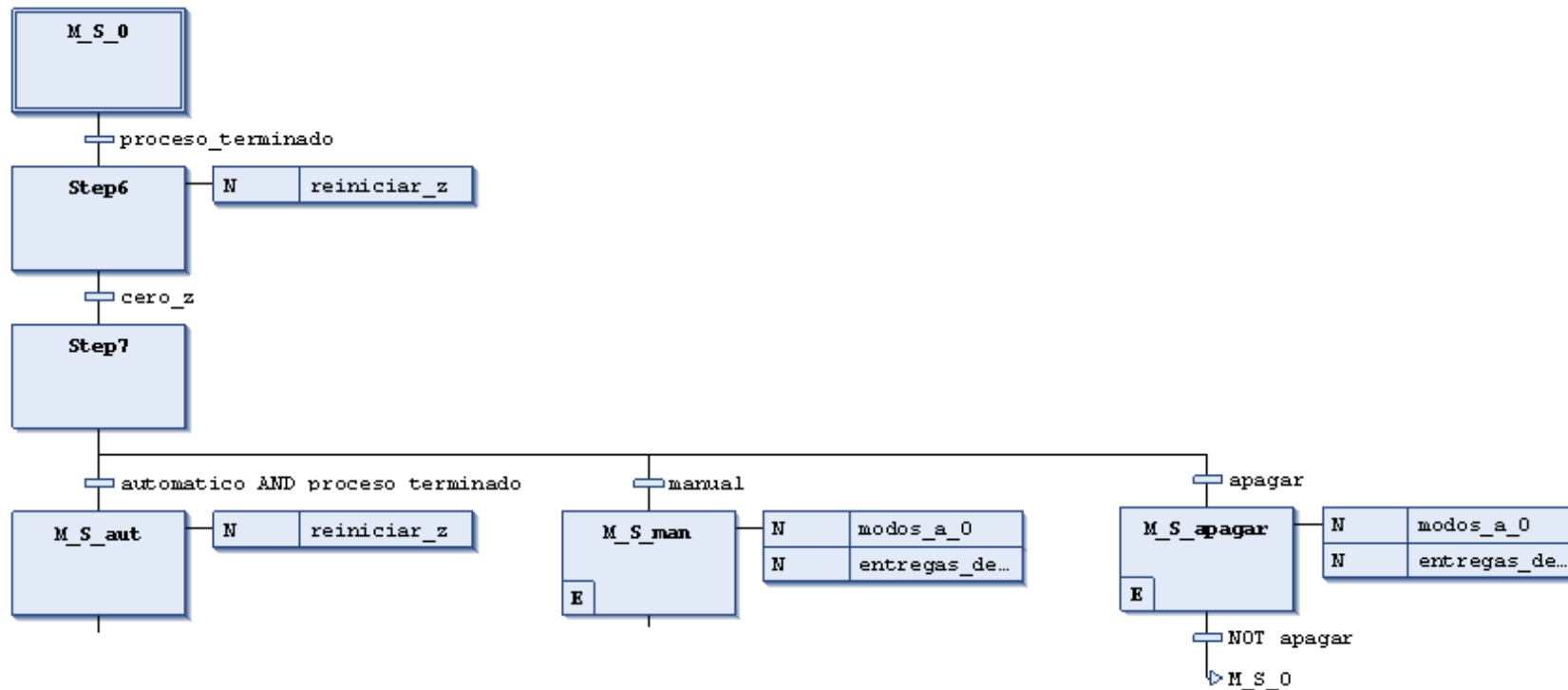
## 5. estado\_casillero

```
1  IF (obj_D12=1 AND obj_D12=1 AND obj_D13=1 AND obj_D14=1 AND obj_D15=1 AND obj_D21=1 AND ... AND obj_D33=1 AND obj_D34=1 AND obj_D35=1)
2  THEN
3  deposito_lleno:=1;
4  ELSE
5  deposito_lleno:=0;
6  END_IF
7
8  IF (obj_D12=0 AND obj_D12=0 AND obj_D13=0 AND obj_D14=0 AND obj_D15=0 AND obj_D21=0 AND ... AND obj_D33=0 AND obj_D34=0 AND obj_D35=0)
9  THEN
10  deposito_vacio:=1;
11  ELSE
12  deposito_vacio:=0;
13  END_IF
14
15  IF (obj_E02=1 AND obj_E02=1 AND obj_E03=1 AND obj_E04=1 AND obj_E05=1)
16  THEN
17  entrega_llena:=1;
18  ELSE
19  entrega_llena:=0;
20  END_IF
```

## 6. Modo\_Selección

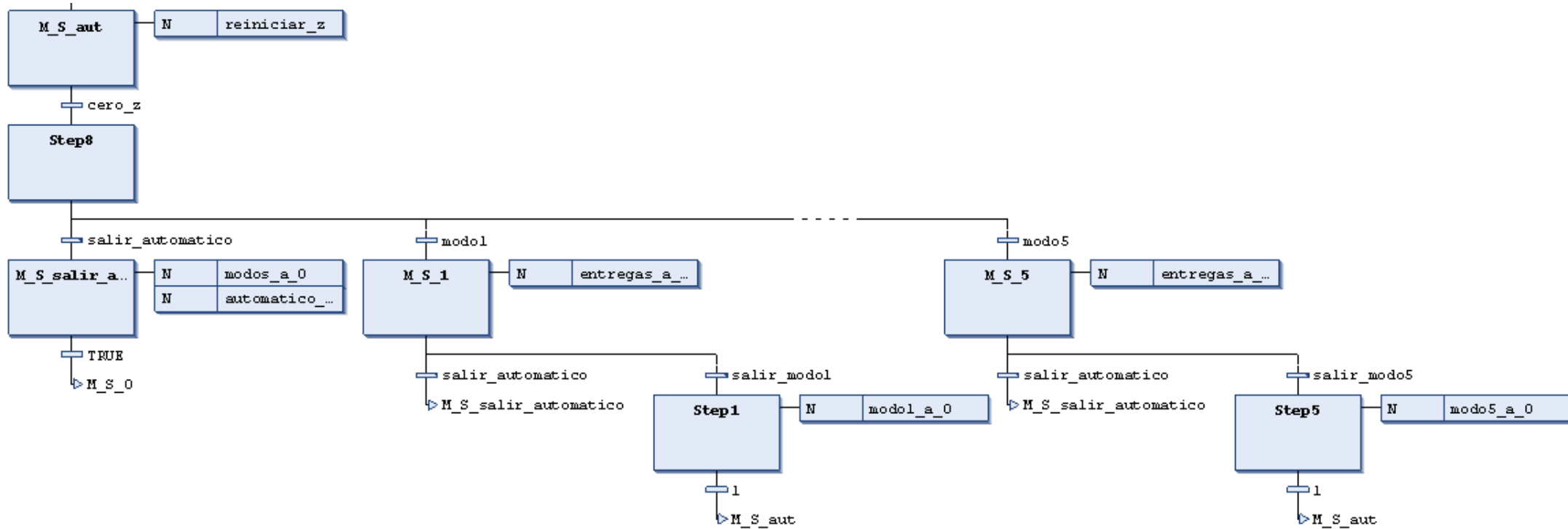
A continuación se muestra el programa Modo\_Seleccion. Este programa es muy largo, por lo que se presenta en tres partes. En la primera se podrá contemplar la manera de acceder a cada una de las tres ramas en las que se divide toda la programación. Automático, manual y apagar. La rama apagar ocupa poco espacio, y cabe entera en la primera parte. La segunda parte corresponde a la rama automático y la tercera a la manual.

### Parte 1

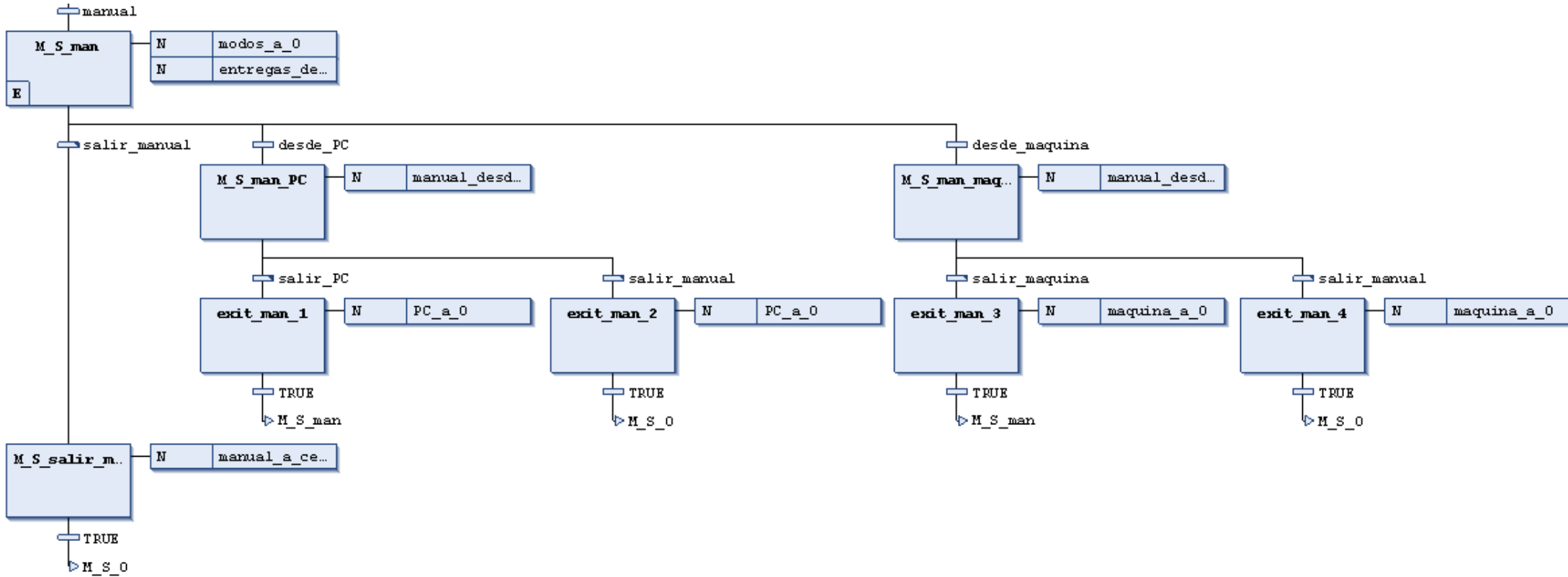


## Parte 2

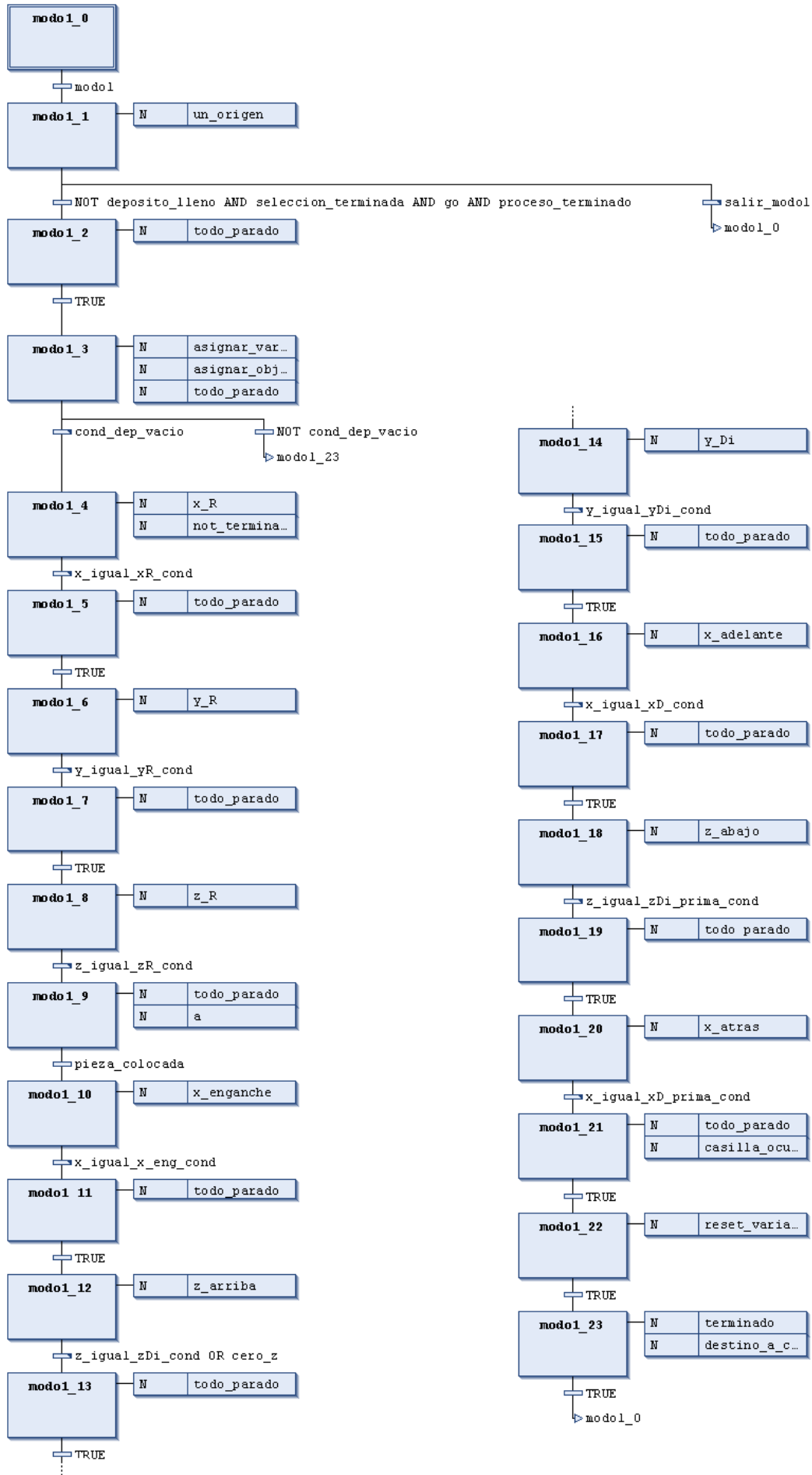
En esta parte se omite la programación correspondiente a los modos 2, 3 y 4, pues son idénticas a las de los modos 1 y 5.



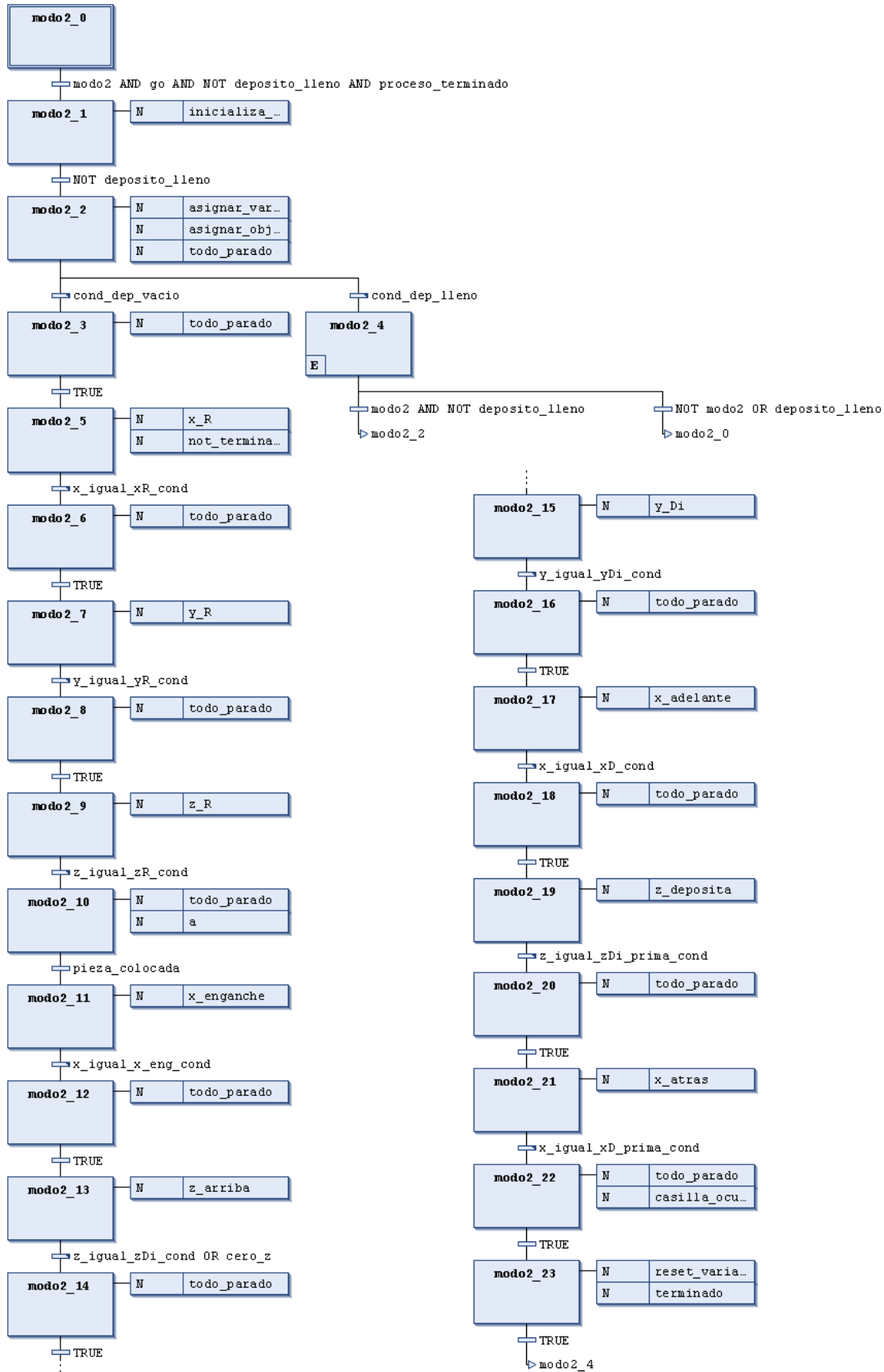
Parte 3



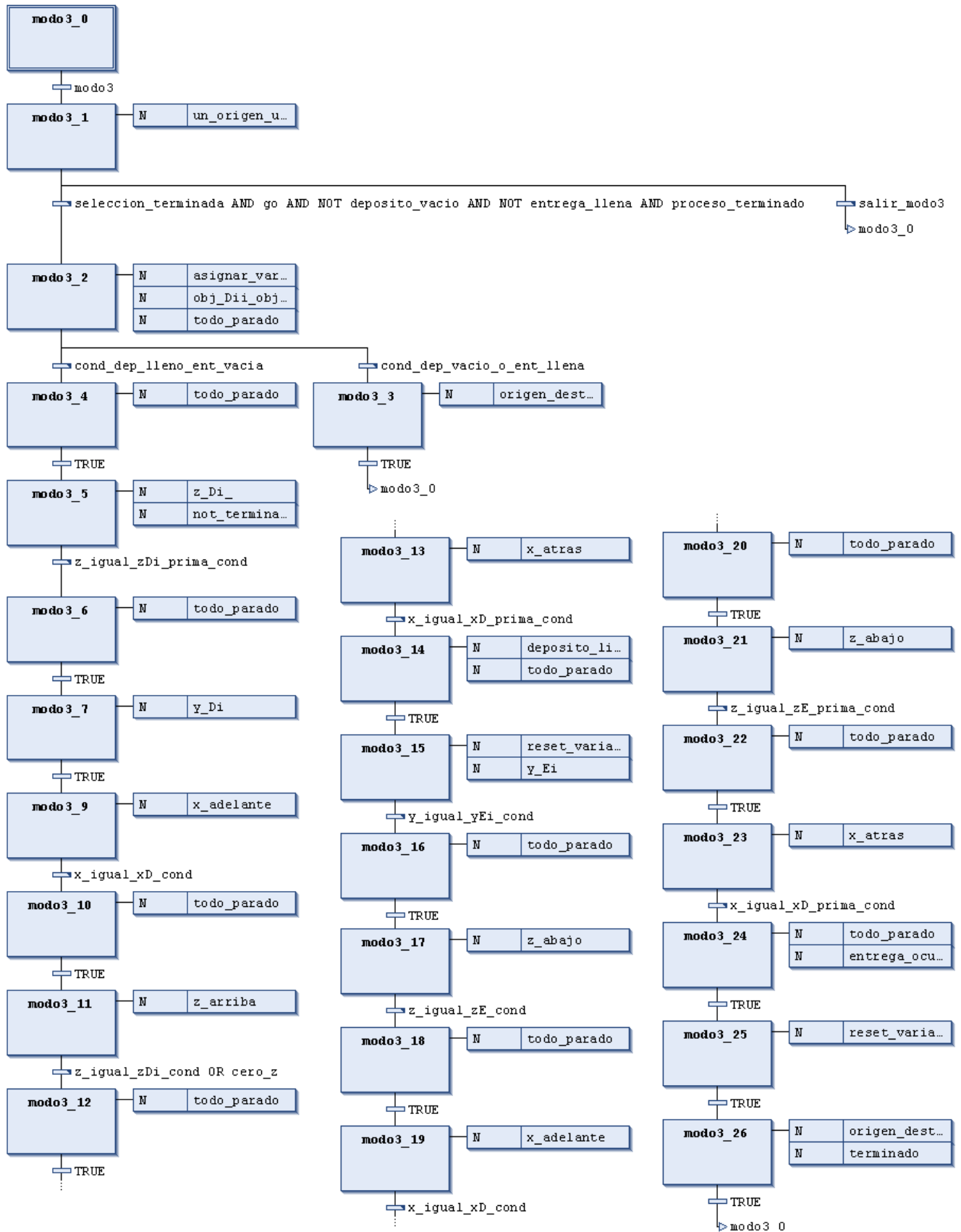
## 7. Modo\_1



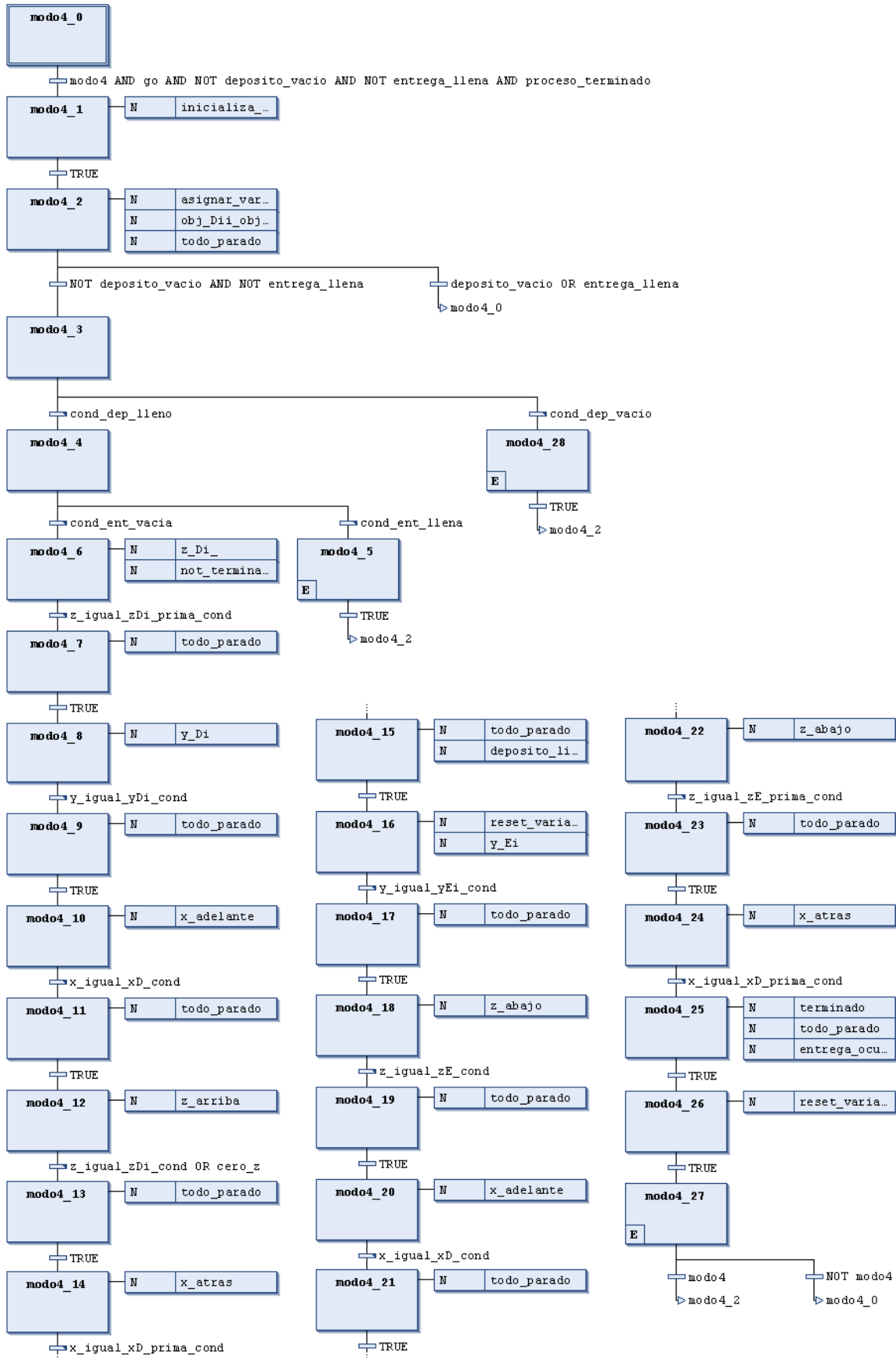
## 8. Modo\_2



## 9. Modo\_3

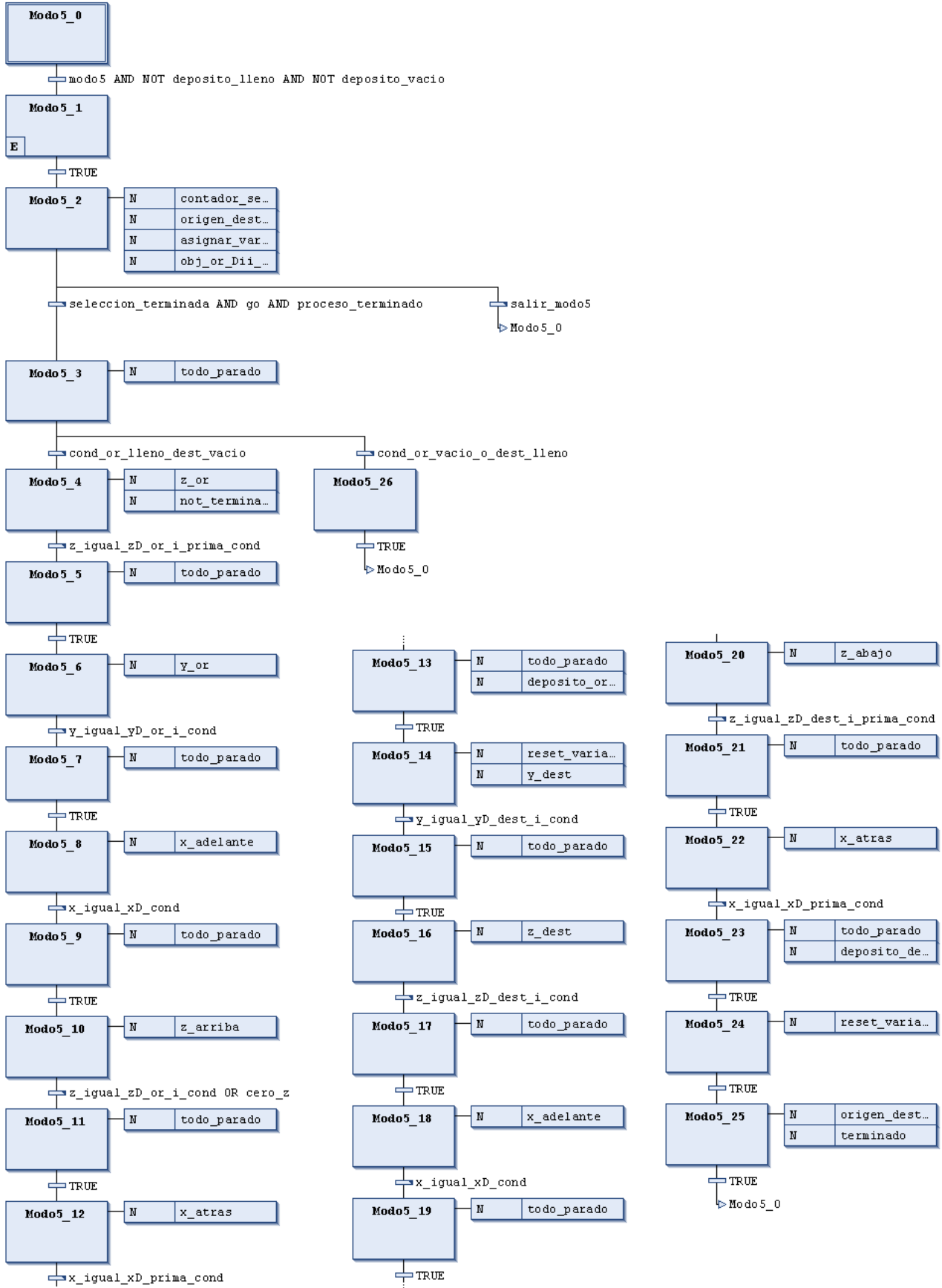


### 10.Mod0\_4





## 11.Mod0\_5



## 12.MANUAL\_DESDE\_MAQUINA

```

1  IF (desde_maquina=1)
2  THEN
3
4  (*Control de movimiento*)
5
6  IF(x_adelante=1)
7  THEN
8  eje_x:=1;
9  adelante:=1;
10 END_IF
11
12 IF(x_atras=1)
13 THEN
14 eje_x:=1;
15 atras:=1;
16 END_IF
17
18 IF(y_izquierda=1)
19 THEN
20 eje_y:=1;
21 adelante:=1;
22 END_IF
23
24 IF(y_derecha=1)
25 THEN
26 eje_y:=1;
27 atras:=1;
28 END_IF
29
30 IF(z_arriba=1)
31 THEN
32 eje_z:=1;
33 adelante:=1;
34 END_IF
35
36 IF(z_abajo=1)
37 THEN
38 eje_z:=1;
39 atras:=1;
40 END_IF
41
42 IF (x_adelante=0 AND x_atras=0)
43 THEN
44 eje_x:=0;
45 END_IF
46
47 IF (y_derecha=0 AND y_izquierda=0)
48 THEN
49 eje_y:=0;
50 END_IF
51
52 IF (z_arriba=0 AND z_abajo=0)
53 THEN
54 eje_z:=0;
55 END_IF
56
57 IF (x_adelante=0 AND y_izquierda=0 AND z_arriba=0)
58 THEN
59 adelante:=0;
60 END_IF
61
62 IF (x_atras=0 AND y_derecha=0 AND z_abajo=0)
63 THEN
64 atras:=0;
65 END_IF
66
67 (*Control de velocidad*)
68
69 (*Speed_up*)
70 IF (speed_up=1 AND vel_1=0 AND vel_2=0 AND u=1)
71 THEN
72 vel_2:=1;
73 u:=0;
74 END_IF
75
76 IF (speed_up=1 AND vel_1=0 AND vel_2=1 AND u=1)
77 THEN
78 vel_1:=1;
79 vel_2:=0;
80 u:=0;
81 END_IF
82
83 IF (speed_up=1 AND vel_1=1 AND vel_2=0 AND u=1)
84 THEN
85 vel_2:=1;
86 u:=0;
87 END_IF
88
89 IF (speed_up=1 AND vel_1=1 AND vel_2=1 AND u=1)
90 THEN
91 vel_1:=0;
92 vel_2:=0;
93 u:=0;
94 END_IF
95
96 IF (speed_up=0)
97 THEN
98 u:=1;
99 END_IF
100
101 (*Speed_down*)
102 IF (speed_down=0 AND vel_1=1 AND vel_2=1 AND d=1)
103 THEN
104 vel_2:=0;
105 d:=0;
106 END_IF
107
108 IF (speed_down=0 AND vel_1=1 AND vel_2=0 AND d=1)
109 THEN
110 vel_1:=0;
111 vel_2:=1;
112 d:=0;
113 END_IF
114
115 IF (speed_down=0 AND vel_1=0 AND vel_2=1 AND d=1)
116 THEN
117 vel_2:=0;
118 d:=0;
119 END_IF
120
121 IF (speed_down=1)
122 THEN
123 d:=1;
124 END_IF
125 END_IF

```

### 13.MANUAL\_DESDE\_PC

```

1  IF(desde_PC=1)
2  THEN
3
4      (*Control de Posición*)
5
6      IF(x_ade=1)
7          THEN
8              eje_x:=1;
9              adelante:=1;
10         END_IF
11
12         IF(x_atr=1)
13             THEN
14                 eje_x:=1;
15                 atras:=1;
16             END_IF
17
18         IF(y_izq=1)
19             THEN
20                 eje_y:=1;
21                 adelante:=1;
22             END_IF
23
24         IF(y_der=1)
25             THEN
26                 eje_y:=1;
27                 atras:=1;
28             END_IF
29
30         IF(z_arr=1)
31             THEN
32                 eje_z:=1;
33                 adelante:=1;
34             END_IF
35
36         IF(z_aba=1)
37             THEN
38                 eje_z:=1;
39                 atras:=1;
40             END_IF
41
42         IF (x_ade=0 AND x_atr=0)
43             THEN
44                 eje_x:=0;
45             END_IF
46
47         IF (y_der=0 AND y_izq=0)
48             THEN
49                 eje_y:=0;
50             END_IF
51
52         IF (z_arr=0 AND z_aba=0)
53             THEN
54                 eje_z:=0;
55             END_IF
56
57         IF (x_ade=0 AND y_izq=0 AND z_arr=0)
58             THEN
59                 adelante:=0;
60             END_IF
61
62         IF (x_atr=0 AND y_der=0 AND z_aba=0)
63             THEN
64                 atras:=0;
65             END_IF
66
67         (*Control de velocidad*)
68
69         (*Speed_up*)
70         IF (s_up=1 AND vel_1=0 AND vel_2=0 AND u=1)
71             THEN
72                 vel_2:=1;
73                 u:=0;
74             END_IF
75
76         IF (s_up=1 AND vel_1=0 AND vel_2=1 AND u=1)
77             THEN
78                 vel_1:=1;
79                 vel_2:=0;
80                 u:=0;
81             END_IF
82
83         IF (s_up=1 AND vel_1=1 AND vel_2=0 AND u=1)
84             THEN
85                 vel_2:=1;
86                 u:=0;
87             END_IF
88
89         IF (s_up=1 AND vel_1=1 AND vel_2=1 AND u=1)
90             THEN
91                 vel_1:=0;
92                 vel_2:=0;
93                 u:=0;
94             END_IF
95
96         IF (s_up=0)
97             THEN
98                 u:=1;
99             END_IF
100
101         (*Speed_down*)
102         IF (s_down=0 AND vel_1=1 AND vel_2=1 AND d=1)
103             THEN
104                 vel_2:=0;
105                 d:=0;
106             END_IF
107
108         IF (s_down=0 AND vel_1=1 AND vel_2=0 AND d=1)
109             THEN
110                 vel_1:=0;
111                 vel_2:=1;
112                 d:=0;
113             END_IF
114
115         IF (s_down=0 AND vel_1=0 AND vel_2=1 AND d=1)
116             THEN
117                 vel_2:=0;
118                 d:=0;
119             END_IF
120
121         IF (s_down=1)
122             THEN
123                 d:=1;
124             END_IF
125
126         END_IF

```



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA



# ANEXO II. ACCIONES MODO AUTOMÁTICO

Proyecto de Automatización y Control mediante  
SCADA de un Sistema de Almacenamiento y  
Recuperación Industrial de Piezas



**JOSE CABOT ALMELA**

Tutor: Juan Pérez Cruz

Cotutor: Manuel Pineda Sánchez



## ÍNDICE ANEXO II

<b>1. asignar_variables_posicion .....</b>	<b>97</b>
<b>2. asignar_obj_Dii / obj_Dii_obj_Eii / obj_or_Dii_obj_dest_Eii.....</b>	<b>98</b>
<b>3. ACCIONES DE MOVIMIENTO.....</b>	<b>99</b>
3.1 x_R, y_R, z_R.....	99
3.2 z_arriba, z_abajo .....	99
3.3 x_adelante, x_atras, x_enganche.....	99
3.4 y_Di, y_Ei .....	100
3.5 x_apagado, y_apagado, z_apagado .....	100
<b>4. reset_variables.....</b>	<b>101</b>
<b>5. contador_seleccionados.....</b>	<b>102</b>
<b>6. origen_destino .....</b>	<b>103</b>



## 1. asignar\_variables\_posicion

La programación de esta acción es muy repetitiva. Se expone a modo de ejemplo una parte de ella, la suficiente como para entenderla e imaginarse el resto.

```
1  (*Asignar las variables de posición en función de la seleccion*)
2
3  (*Entrega*)
4
5  IF (E01=1)
6  THEN
7  yEi:=yDi_c1;
8  e:=1;
9  END_IF
10
11 IF (E02=1)
12 THEN
13 yEi:=yDi_c2;
14 e:=2;
15 END_IF
16
29 IF (E05=1)
30 THEN
31 yEi:=yDi_c5;
32 e:=5;
33 END_IF
34
35 (*Depósitos*)
36
37 (*Fila1*)
38
39 IF (D11=1)
40 THEN
41 yDi:=yDi_c1;
42 zDi_:=zDi_f1;
43 c:=1;
44 END_IF
45
46 IF (D12=1)
47 THEN
48 yDi:=yDi_c2;
49 zDi_:=zDi_f1;
50 c:=2;
51 END_IF
141 IF (D35=1)
142 THEN
143 yDi:=yDi_c5;
144 zDi_:=zDi_f3;
145 c:=15;
146 END_IF
147
148 IF (c=1 OR c=2 OR c=3 OR c=4 OR c=5 OR c=6 OR c=7 OR c=8 OR c=9 OR c=10)
149 THEN
150 zDi:=zDi_+35;
151 END_IF
152
153 IF (c=11 OR c=12 OR c=13 OR c=14 OR c=15)
154 THEN
155 zDi:=zDi_+235;
156 END_IF
```

Este fragmento de código solo está incorporado para los modos 1 y 3.

## 2. asignar\_obj\_Dii / obj\_Dii\_obj\_Eii / obj\_or\_Dii\_obj\_dest\_Eii

Las diferencias entre estas acciones son mínimas. Se omiten muchas partes previsibles.

```
1 (*Para comprobar el estado del origen y
2 destino cargamos en obj_Dii y en obj_Eii
3 el estado de las casillas seleccionadas*)
4
```

```
5 (*Entregas*)
```

```
6
7 IF (e=1)
8 THEN
9 obj_Eii:=obj_E01;
10 END_IF
```

```
11
12 IF (e=2)
13 THEN
14 obj_Eii:=obj_E02;
15 END_IF
```

```
16
17 IF (e=5)
18 THEN
19 obj_Eii:=obj_E05;
20 END_IF
```

```
21 (*Depósitos*)
```

```
22 (*Filas*)
```

```
23
24 IF (c=1)
25 THEN
26 obj_Dii:=obj_D11;
27 END_IF
```

```
28
29 IF (c=2)
30 THEN
31 obj_Dii:=obj_D12;
32 END_IF
```

La acción "asignar\_obj\_Dii"  
no contiene esta parte  
correspondiente a las entregas

Acción "obj\_Dii\_obj\_Eii"

```
1 (*Asignar valor a obj_or_Dii y a obj_dest_Dii*)
2
```

```
3 (*Origen*)
```

```
4
5 IF (c=1)
6 THEN
```

```
7
8 (*Fila1*)
```

```
9
10 IF (or11=1)
```

```
11 THEN
12 obj_or_Dii:=obj_D11;
13 END_IF
```

```
14
15 IF (or35=1)
```

```
16 THEN
17 obj_or_Dii:=obj_D35;
18 END_IF
```

```
19 END_IF
```

```
20 (*Destino*)
```

```
21
22 IF (c=2)
```

```
23 THEN
24 (*Fila1*)
```

```
25
26 IF (dest11=1)
```

```
27 THEN
28 obj_dest_Dii:=obj_D11;
29 END_IF
```

```
30
31 IF (dest12=1)
```

```
32 THEN
33 obj_dest_Dii:=obj_D12;
34 END_IF
```

Acción  
"obj\_or\_Dii\_obj\_dest\_Eii"

### 3. ACCIONES DE MOVIMIENTO

#### 3.1 x\_R, y\_R, z\_R

<pre> 1  (* x_R *) 2 3  eje_x:=1; 4  IF (x&gt;xR) 5      THEN 6      adelante:=1; 7  END_IF 8 9  IF (x&lt;xR) 10     THEN 11     atras:=1; 12  END_IF 13 14  IF (x&lt;xR-20 OR x&gt;xR+20) 15     THEN 16     vel_1:=1; 17     vel_2:=1; 18  END_IF 19 20  IF (x&gt;xR-20 AND x&lt;xR+20) 21     THEN 22     vel_1:=0; 23     vel_2:=0; 24  END_IF </pre>	<pre> 1  (* y_R *) 2 3  eje_y:=1; 4  IF (y&gt;yR) 5      THEN 6      adelante:=1; 7  END_IF 8 9  IF (y&lt;yR) 10     THEN 11     atras:=1; 12  END_IF 13 14  IF (y&lt;yR-20 OR y&gt;yR+20) 15     THEN 16     vel_1:=1; 17     vel_2:=1; 18  END_IF 19 20  IF (y&gt;yR-20 AND y&lt;yR+20) 21     THEN 22     vel_1:=0; 23     vel_2:=0; 24  END_IF </pre>	<pre> 1  (* z_R *) 2 3  eje_z:=1; 4 5  IF (z&gt;zR) 6      THEN 7      atras:=1; 8  END_IF 9 10 IF (z&lt;zR) 11     THEN 12     adelante:=1; 13  END_IF </pre>
---	---	--

#### 3.2 z\_arriba, z\_abajo

<pre> 1  (* z_arriba *) 2 3  eje_z:=1; 4  adelante:=1; </pre>	<pre> 1  (* z_abajo *) 2 3  eje_z:=1; 4  atras:=1; </pre>
---	---

#### 3.3 x\_adelante, x\_atras, x\_enganche

<pre> 1  (* x_adelante *) 2 3  eje_x:=1; 4  adelante:=1; 5 6  IF (x&gt;xD+20) 7      THEN 8      vel_1:=1; 9      vel_2:=1; 10  END_IF 11 12 IF (x&lt;xD+20) 13     THEN 14     vel_1:=0; 15     vel_2:=0; 16  END_IF </pre>	<pre> 1  (* x_atras *) 2 3  eje_x:=1; 4  atras:=1; 5 6  IF (x&lt;xD_-20) 7      THEN 8      vel_1:=1; 9      vel_2:=1; 10  END_IF 11 12 IF (x&gt;xD_-20) 13     THEN 14     vel_1:=0; 15     vel_2:=0; 16  END_IF </pre>	<pre> 1  (* x_enganche *) 2 3  eje_x:=1; 4  adelante:=1; </pre>
--	--	---



### 3.4 y<sub>Di</sub>, y<sub>Ei</sub>

<pre> 1  (* y_Di *) 2 3  eje_y:=1; 4 5  IF (y&gt;yDi) 6    THEN 7    adelante:=1; 8  END_IF 9 10 IF (y&lt;yDi) 11   THEN 12   atras:=1; 13 END_IF 14 15 IF (y&lt;yDi-20 OR y&gt;yDi+20) 16   THEN 17   vel_1:=1; 18   vel_2:=1; 19 END_IF 20 21 IF (y&gt;yDi-20 AND y&lt;yDi+20) 22   THEN 23   vel_1:=0; 24   vel_2:=0; 25 END_IF </pre>	<pre> 1  (* y_Ei *) 2 3  eje_y:=1; 4 5  IF (y&gt;yEi) 6    THEN 7    adelante:=1; 8  END_IF 9 10 IF (y&lt;yEi) 11   THEN 12   atras:=1; 13 END_IF 14 15 IF (y&lt;yEi-20 OR y&gt;yEi+20) 16   THEN 17   vel_1:=1; 18   vel_2:=1; 19 END_IF 20 21 IF (y&gt;yEi-20 AND y&lt;yEi+20) 22   THEN 23   vel_1:=0; 24   vel_2:=0; 25 END_IF </pre>
---	---

### 3.5 x<sub>apagado</sub>, y<sub>apagado</sub>, z<sub>apagado</sub>

<pre> 1  (* x_apagado *) 2 3  (* Avanzar hacia la posición y=yF*) 4 5  eje_x:=1; 6 7  IF (x&gt;xF) 8    THEN 9    adelante:=1; 10 END_IF 11 12 IF (x&lt;xF) 13   THEN 14   atras:=1; 15 END_IF 16 17 IF (x&lt;xF-20 OR x&gt;xF+20) 18   THEN 19   vel_1:=1; 20   vel_2:=1; 21 END_IF 22 23 IF (x&gt;xF-20 AND x&lt;xF+20) 24   THEN 25   vel_1:=0; 26   vel_2:=0; 27 END_IF </pre>	<pre> 1  (* y_apagado *) 2 3  (* Avanzar hacia la posición y=yF*) 4 5  eje_y:=1; 6 7  IF (y&gt;yF) 8    THEN 9    adelante:=1; 10 END_IF 11 12 IF (y&lt;yF) 13   THEN 14   atras:=1; 15 END_IF 16 17 IF (y&lt;yF-20 OR y&gt;yF+20) 18   THEN 19   vel_1:=1; 20   vel_2:=1; 21 END_IF 22 23 IF (y&gt;yF-20 AND y&lt;yF+20) 24   THEN 25   vel_1:=0; 26   vel_2:=0; 27 END_IF </pre>	<pre> 1  (* z_apagado *) 2 3  (* Subir hacia la posición del sensor *) 4 5  eje_z:=1; 6  adelante:=1; </pre>
--	--	--

#### 4. reset\_variables

```
1  (*Se carga a las variables de estado (libre/ocupado) correspondientes a las
2  casillas seleccionadas el cambio que se ha producido en el proceso.*)
3
4  (*Entregas*)
5
6  IF (e=1)
7  THEN
8  obj_E01:=obj_Eii;
9  END_IF
10
11 IF (e=2)
12 THEN
13 obj_E02:=obj_Eii;
14 END_IF
15
16
17
18
19
20
21
22
23
24
25
26 IF (e=5)
27 THEN
28 obj_E05:=obj_Eii;
29 END_IF
30
31 (*Depósitos*)
32
33 (*Fila1*)
34
35 IF (c=1)
36 THEN
37 obj_D11:=obj_Dii;
38 END_IF
39
40 IF (c=2)
41 THEN
42 obj_D12:=obj_Dii;
43 END_IF
```

## 5. contador\_seleccionados

```
1 (*Fila 1*)
2
3 IF (D11=1)
4 THEN
5     c11:=1;
6 END_IF
7
8 IF (D11=0)
9 THEN
10     c11:=0;
11 END_IF
12
13 IF (D12=1)
14 THEN
15     c12:=1;
16 END_IF
17
18 IF (D12=0)
19 THEN
20     c12:=0;
21 END_IF
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37 IF (D34=1)
38 THEN
39     c34:=1;
40 END_IF
41
42 IF (D34=0)
43 THEN
44     c34:=0;
45 END_IF
46
47 IF (D35=1)
48 THEN
49     c35:=1;
50 END_IF
51
52 IF (D35=0)
53 THEN
54     c35:=0;
55 END_IF
56
57 c:=c11+c12+c13+c14+c15+c21+c22+c23+c24+c25+c31+c32+c33+c34+c35;
```

## 6. origen\_destino

```
1  (*Diferenciar entre origen y destino*) 149  (*Destino*)
2  150  IF (c=2)
3  (*Origen*) 151  THEN
4  IF (c=1) 152
5  THEN 153  (*Filas1*)
6  154
7  (*Filas1*) 155  IF (c11=1 AND or11=0)
8  156  THEN
9  IF (c11=1) 157  dest11:=1;
10  THEN 158  END_IF
11  or11:=1; 159
12  END_IF 160  IF (c11=0 OR or11=1)
13  IF (c11=0) 161  THEN
14  THEN 162  dest11:=0;
15  or11:=0; 163  END_IF
16  END_IF 164
17  165  IF (c12=1 AND or12=0)
18  IF (c12=1) 166  THEN
19  THEN 167  dest12:=1;
20  or12:=1; 168  END_IF
21  END_IF 169
22  IF (c12=0) 170  IF (c12=0 OR or12=1)
23  THEN 171  THEN
24  or12:=0; 172  dest12:=0;
25  END_IF 173  END_IF
```



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA



# ANEXO III

# COMPONENTES

Proyecto de Automatización y Control mediante  
SCADA de un Sistema de Almacenamiento y  
Recuperación Industrial de Piezas



**JOSE CABOT ALMELA**

Tutor: Juan Pérez Cruz

Cotutor: Manuel Pineda Sánchez



## ÍNDICE ANEXO III

<b>1. La Máquina .....</b>	<b>106</b>
1.1 Motor.....	109
1.2 POTENCIÓMETRO .....	110
1.3 SENSOR INDUCTIVO .....	111
1.4 INTERRUPTOR DE FINAL DE CARRERA.....	112
1.5 PULSADOR EMERGENCIA .....	113
1.6 CIRCUITOS ELÉCTRICOS .....	114
<b>2. Horquilla .....</b>	<b>115</b>
<b>3. Piezas .....</b>	<b>116</b>
<b>4. Zonas de Recogida, Depósito y Entrega .....</b>	<b>117</b>
4.2 RECOGIDA.....	117
4.3 DEPÓSITO Y ENTREGA .....	118
<b>5. PLC .....</b>	<b>120</b>
<b>6. Ordenador .....</b>	<b>121</b>



## 1. LA MÁQUINA

La máquina con la que se trabaja es una máquina de tres ejes que permite el desplazamiento en las tres coordenadas cartesianas: x, y, z.



Figura 1.1  
Máquina de tres ejes

Componentes por eje:

- |                     |  |
|---------------------|--|
| Motor asíncrono:    | se encarga de proporcionar el movimiento de cada eje a través de un tornillo sin fin.  |
| Tornillo sin fin:   | trasmite el movimiento del motor a las piezas móviles de la máquina.   |
| Potenciómetro:      | (no disponible en el eje z) proporciona una señal analógica que sirve para determinar la posición de los ejes x e y.   |
| Sensor inductivo:   | señaliza una posición concreta de la máquina.  |
| Finales de carrera: | dos por eje. Delimitan los límites del movimiento. Desactivan mecánicamente el motor de su eje impidiendo problemas en la máquina. Son elementos de seguridad. |

*Sistema de referencia*

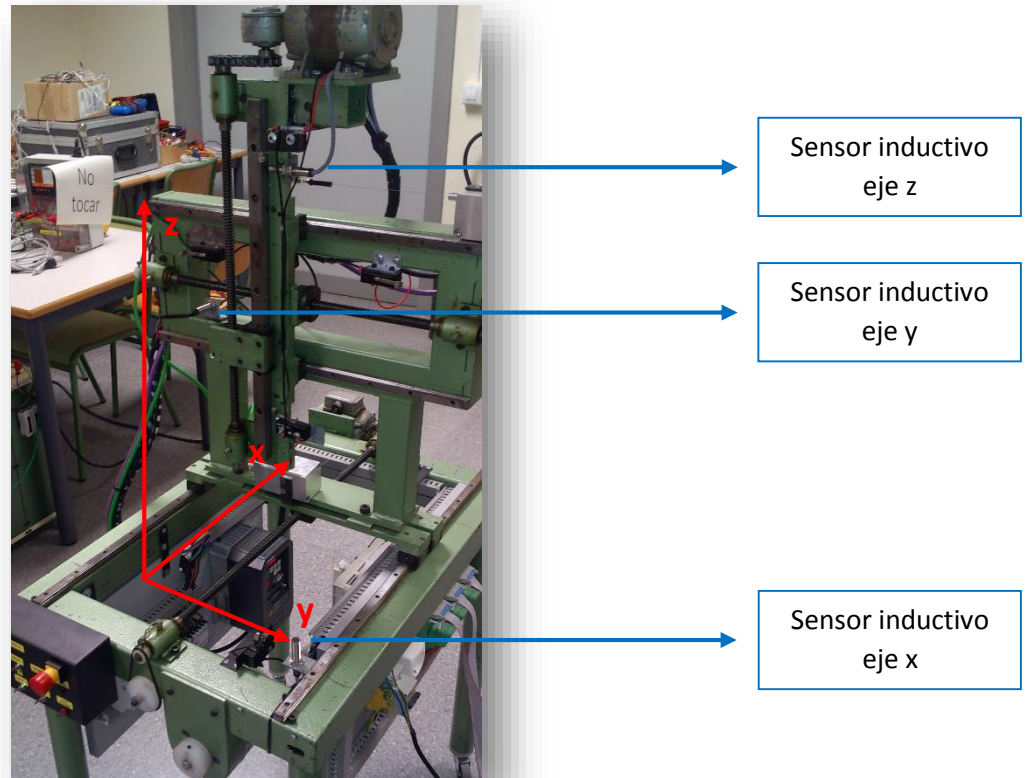


Figura 1.2  
Sistema de referencia

El sistema de referencia escogido se aprecia en la imagen anterior.

Los sensores inductivos de los ejes x e y coinciden con los ceros de las coordenadas, mientras que el sensor inductivo del eje z se corresponde a la coordenada máxima de este eje.

#### *Desplazamientos sobre los ejes*

Esta máquina es capaz de realizar movimientos de avance y retroceso en los tres ejes. El elemento móvil de la máquina es el encargado de trasladar las piezas de una zona a otra del área de trabajo. En dicho elemento se acopla un accesorio fijado mediante imanes que permite que sea utilizado como horquilla. A partir de ahora se referirá al conjunto como horquilla.

Para realizar el movimiento en la dirección del eje z, el único elemento móvil que se desplaza es la horquilla. Se mueve directamente sobre el tornillo sin fin de dicho eje.

En el movimiento en la dirección y, se desplazan, en conjunto, tanto la horquilla como el tornillo sin fin utilizado en el movimiento anterior. Este movimiento se realiza dentro de una pared vertical que integra al tornillo sin fin con el que se transmite el movimiento al eje y.

Por último, en el movimiento en la dirección del eje x se mueve toda la pared mencionada anteriormente, barriendo toda la superficie horizontal de la máquina.



A continuación se muestran dos imágenes con indicaciones que servirán para entender lo explicado anteriormente.

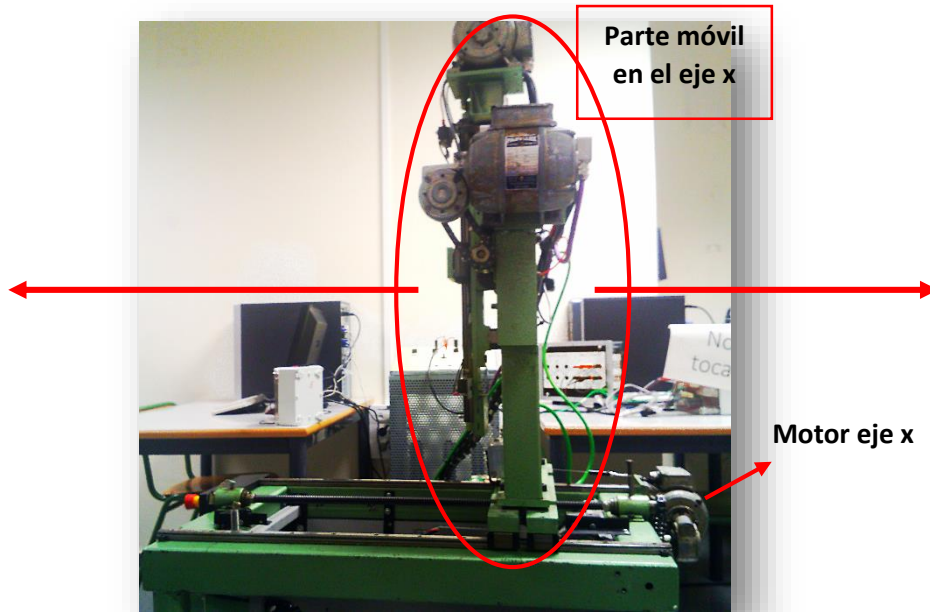


Figura 1.3  
Vista lateral de la máquina

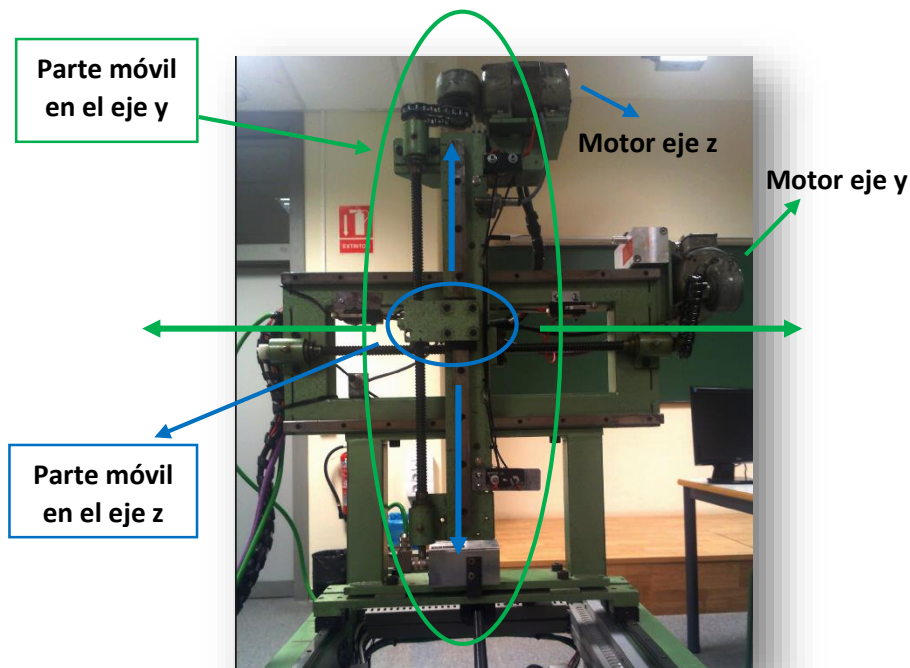


Figura 1.4  
Vista frontal de la máquina

### 1.1 Motor

La máquina lleva instalados tres motores trifásicos asíncronos. Cada uno de ellos se encarga de proporcionar movimiento a un eje. El motor utiliza un reductor y un conjunto de engranajes y cadena para transmitir la acción al tornillo sin fin. Se pueden accionar todos a la vez, pero todos con el mismo sentido de giro y misma velocidad.

Los motores están conectados en estrella, con tensión entre fases de 380V, frecuencia de 50Hz y 300 rpm.

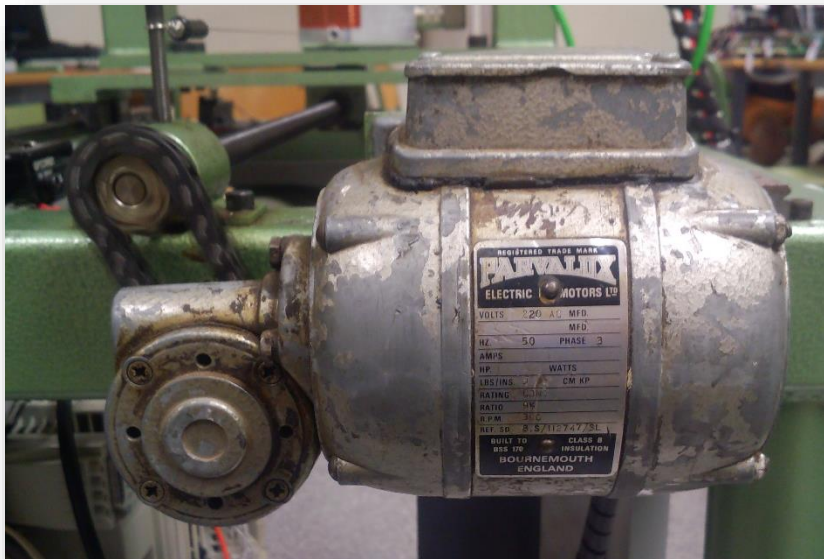


Figura 1.5  
Motor Parvalux. Nº de referencia 85/104863/2M

Sobre los motores actúa un variador de frecuencia, que permite variar sus velocidades.

Según la combinación de las señales digitales que recibe el variador, la frecuencia que le llega a los motores puede ser de 10, 15, 30 o 40 Hz.

DI1	DI2	Hz
0	0	10
0	1	15
1	0	30
1	1	40

## 1.2 POTENCIÓMETRO

En la máquina se encuentran dos potenciómetros que se utilizan para determinar la posición en los ejes x e y.

Los potenciómetros son sensores de desplazamiento de tipo resistivo. Contienen una resistencia de valor fijo y un cursor que se desliza sobre ella. El cursor está enganchado mediante un hilo a la parte móvil de la máquina (o a la parte fija, si el potenciómetro está en la móvil), de manera que si se mueve la máquina, se mueve el cursor. Cuando el cursor se desplaza sobre la resistencia varía su valor, por lo que si se alimenta a un potencial fijo, el incremento de intensidad se puede utilizar para calcular el desplazamiento.



Figura 1.6  
Potenciómetro utilizado para controlar la posición en el eje x

Los potenciómetros proporcionan una señal analógica. Esta señal será transformada mediante una recta de conversión a la posición real según el sistema de referencia utilizado. De esto se encarga el programa “señal\_anal\_a\_señal\_real”, cuyo funcionamiento y explicación más amplia se verá más adelante en el apartado “7.3 Control de posición” (pág. 34).

### 1.3 SENSOR INDUCTIVO

En cada eje se tiene un sensor inductivo. Este tipo de sensores sirve para detectar objetos metálicos.

El funcionamiento de un sensor inductivo se basa en la generación de un campo magnético a través de una bobina. Al acercarse lo suficiente un objeto metálico, sobre este se producen corrientes de Foucault inducidas por el campo magnético, lo cual reduce la energía del circuito que lo produce y provoca una reducción tanto de la amplitud de las ondas de dicho campo como de las ondas del propio circuito. Al disminuir lo suficiente la amplitud (amplitud umbral), el sensor emite una señal que avisa de la proximidad de un objeto metálico.



*Figura 1.7  
Sensor inductivo eje y*

Estos sensores se utilizan en la programación para identificar posiciones dentro del sistema de referencia. Además, como se verá más adelante, el sensor inductivo situado en el eje z desempeña un papel crucial en el control de la posición de dicha coordenada, pues actualiza el valor de ésta e impide que se acumule el error. En el apartado “7.3.3 Coordenada z” (pág. 36), se explica con más detalle cómo se controla la posición del eje z y el papel que desempeña el sensor inductivo.



#### 1.4 INTERRUPTOR DE FINAL DE CARRERA

Los interruptores de final de carrera son elementos de seguridad. Impiden que continúe el desplazamiento de las partes móviles de la máquina a partir de cierto punto. De esta manera protegen a la máquina de posibles averías, ya sea en las partes móviles, elementos auxiliares o incluso en los motores. Además también se evita un posible accidente que pudiese afectar a los usuarios de la máquina.

Hay dos finales de carrera en cada eje, cada uno de los cuales delimita el desplazamiento máximo que se puede realizar en cada sentido.



*Figura 1.8  
Interruptor de final de carrera eje x*

Su funcionamiento se basa en el corte de la alimentación del motor que proporciona movimiento a dicho eje. Es completamente mecánico. No depende del autómatas lo cual implica que un fallo en éste no deja la máquina desprotegida.

### 1.5 PULSADOR EMERGENCIA

Además de los elementos anteriores, en la zona de control manual de la máquina hay disponible un pulsador de emergencia, rojo, con forma de seta. Al pulsarlo inhabilita los tres motores automáticamente y de manera mecánica. Su misión es detener cualquier proceso que se esté llevando a cabo y suponga un peligro para un operario o la propia máquina.

Del mismo modo que los interruptores finales de carrera, el pulsador de emergencia actuará independientemente del autómeta.



*Figura 1.9  
Pulsador de emergencia*

El estado del pulsador es de normalmente cerrado y, mientras no se accione, permite la alimentación de los tres motores. Cuando se acciona, se corta la alimentación y los motores que están en funcionamiento se paran automáticamente.

### 1.6 CIRCUITOS ELÉCTRICOS

Los elementos mencionados anteriormente son componentes de los circuitos de control y de potencia que regulan la máquina.

El circuito de control se encarga de regular los contactores que permiten el movimiento en cada eje. Hay tres contactores, Kx, Ky y Kz, uno por eje. Cuando les llega intensidad, permiten el movimiento de la máquina sobre su eje, como se observa en la figura 1.11.

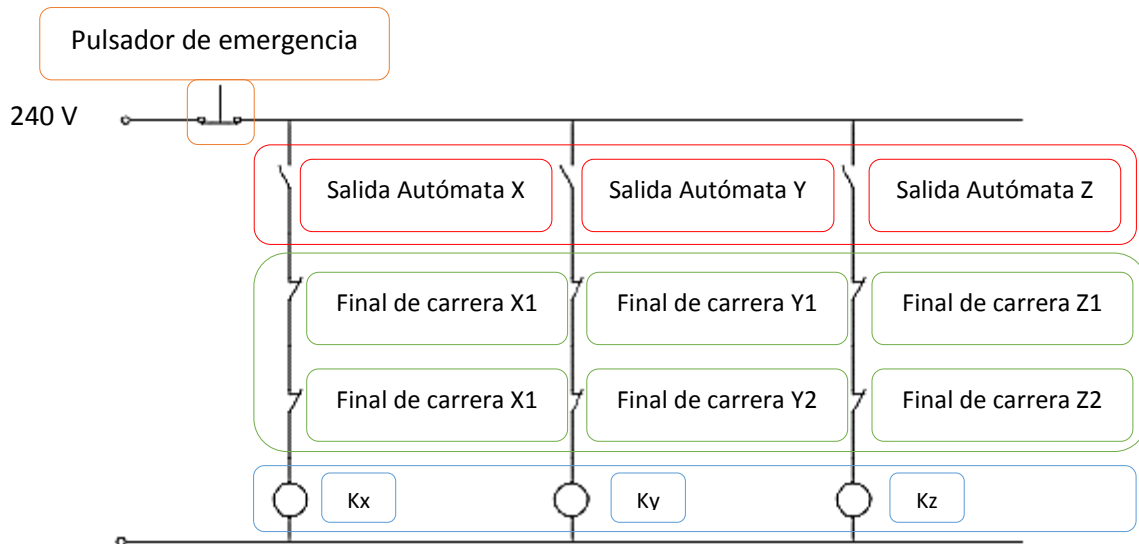


Figura 1.10  
Circuito de control

El circuito de potencia se encarga de alimentar a los motores. La alimentación está controlada por los contactores y por un interruptor magnetotérmico que abre el circuito al producirse sobreintensidades.

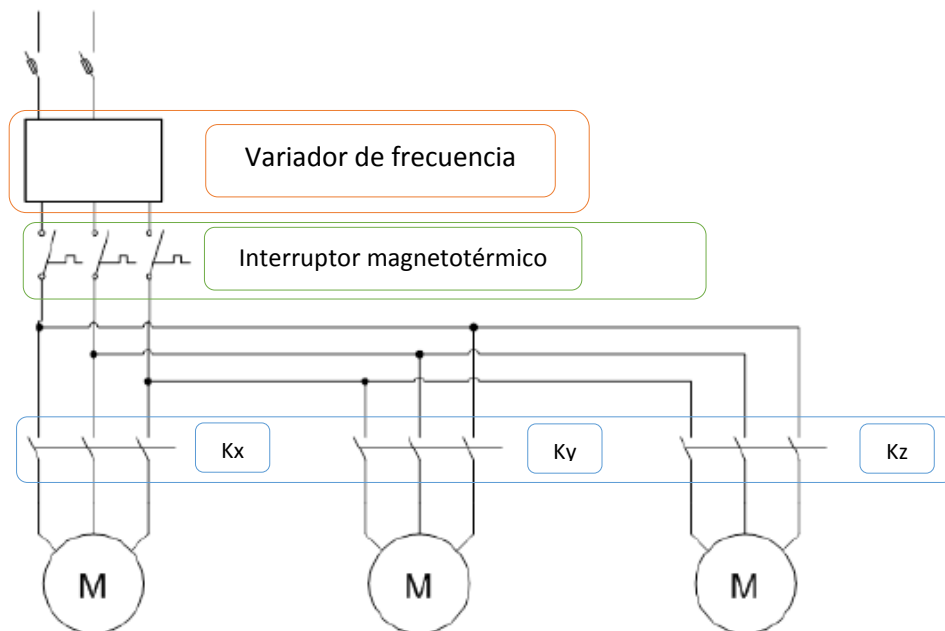


Figura 1.11  
Circuito de potencia

## 2. HORQUILLA

El elemento móvil de la máquina es una plancha plana de metal. Para que pueda funcionar como una horquilla se fabrica un accesorio.

Se fabrica con contrachapado. Por la parte posterior se pegan dos imanes de neodimio para conseguir una unión magnética con la superficie metálica de la máquina. Los imanes, que se pueden apreciar en la imagen 4.2.c, proporcionan suficiente resistencia para las piezas que se va a trasladar.



*Figuras 2.1, 2.2 y 2.3  
Accesorio horquilla para la máquina*

Las horquillas se diseñan con terminación en punta. Junto con el diseño de la plataforma de recogida y de las celdas, con soporte y tope ambas, permiten el acople correcto de las piezas cuando son recogidas, aunque se produjese un pequeño error de posición.

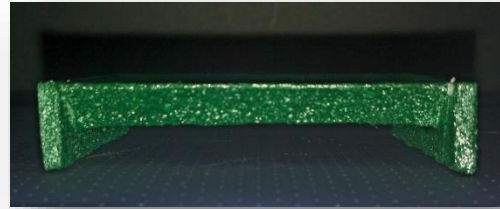
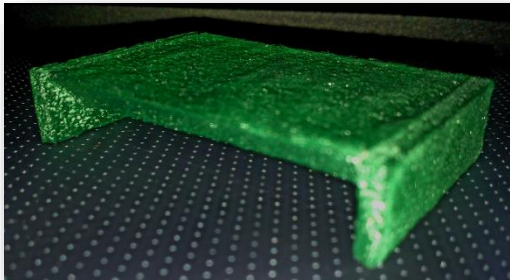


*Figura 2.4  
Accesorio horquilla acoplado a la máquina*



### 3. PIEZAS

Para la realización del proyecto se han diseñado unas piezas hechas de deprom de 3 y 6 mm. El deprom es un tipo de corcho de alta densidad. Es el mismo material con el que se ha fabricado la mayor parte del casillero.



Figuras 3.1 y 3.2  
Pieza

Como se puede ver en la figura la forma de las piezas es óptima para el tipo de horquilla que se ha diseñado. Al tener una forma de u invertida aumenta el error admisible en la coordenada z. Además disponen de un chaflán para disminuir aún más, si cabe, las posibilidades de fallo.

En una industria real se podría trasladar cualquier tipo de piezas. Lo común es hacerlo mediante palés, donde se engancharía la horquilla. Aun así, se recuerda que la fabricación de casillero, piezas, plataforma de recogida, etc., sirve para realizar una simulación. El hecho importante es que la programación sí sería válida a escala industrial.

## 4. ZONAS DE RECOGIDA, DEPÓSITO Y ENTREGA

### 4.2 RECOGIDA

La zona de recogida será donde se encontrará la pieza antes de ser conducida a los depósitos. Se supondrá que, en función de la industria en la que se utilice la máquina, las piezas llegarán a dicha zona a través de una cinta transportadora, otra máquina similar, un operario, etc. Las coordenadas que se han elegido para la recogida son (215, 165, 235) mm, pudiéndose modificar sencillamente si se desea.

Debido a las características de la máquina no ha sido posible realizar una zona de recogida fija como se daría en la realidad. Esto es debido a las características de sus movimientos. Como se ha mencionado anteriormente, al avanzar a lo largo del eje x, la parte móvil de la máquina barre toda la superficie horizontal. Tampoco es posible la alternativa de rotar sobre el eje z que permitiría un acceso por el lateral de la máquina.

Por lo tanto se ha optado por crear una plataforma móvil para solucionar este problema.

Esta plataforma se ha fabricado con dos planchas de madera de contrachapado unidas mediante un rodete. Sobre una de estas maderas se levanta un soporte de corcho para que la horquilla enganche bien la pieza.



Figura 4.1  
Zona de entrega

Se le ha acoplado un tope de deprom al final de la plataforma para impedir que la pieza se desplace y ayudar a su correcto posicionamiento encima de la horquilla.

### 4.3 DEPÓSITO Y ENTREGA

La zona de depósito y entrega forman parte de un mismo “casillero” que representaría un almacén real. Consta de 4 filas y 5 columnas. Las 3 filas superiores forman la zona de depósito y la fila inferior es la zona de entrega.

Las medidas del casillero son 340x80x300 mm.

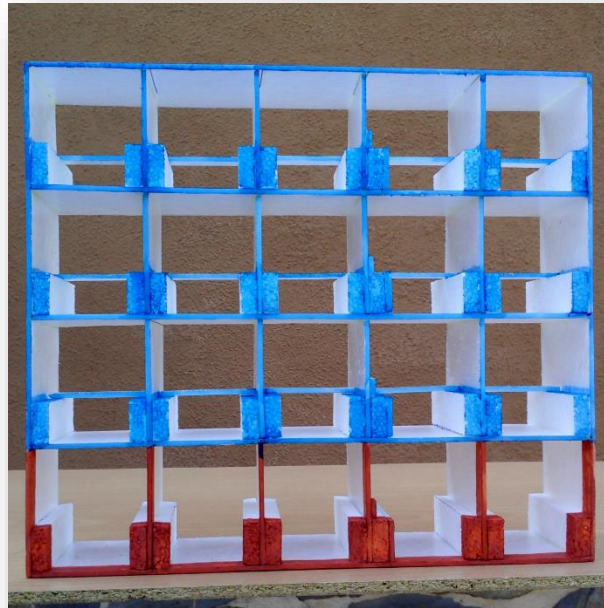


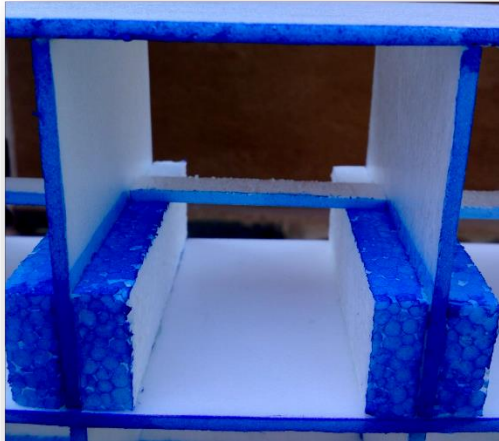
Figura 4.2  
Casillero. Zona de depósito (azul) y zona de entrega (marrón)

Las baldas horizontales y los tabiques se han fabricado con deprom y los soportes con corcho normal.

En el depósito se colocan las piezas provenientes de la recogida. En total hay 15 celdas de depósito, que a lo largo del proyecto se denominan  $D_{ij}$ , siendo  $i$  el número de fila, de 1 a 3, y  $j$  el número de columna, de 1 a 5.

La zona de entrega es el lugar donde las piezas se llevan cuando quieren ser utilizadas. Vienen del depósito y se supondrá que serán extraídas, al igual que llegaban a la zona de entrega, por una cinta transportadora, máquina similar, operario, etc., en función del tipo de industria.

Las celdas de depósito y entrega disponen de unos soportes en los laterales que, al igual que en la plataforma de la entrega, sirven para sustentar a la pieza. Además, las celdas de depósito están provistas de un tope que facilita el enganche de las piezas.



*Figura 4.3*  
*Celda de depósito*



*Figura 4.4*  
*Celda de entrega*

Más adelante se verá que el control de la posición en el eje z no es tan preciso como se desearía, ni como sería en una industria. Estos soportes ayudan a minimizar los problemas que podría ocasionar dicha falta de precisión.

Las medidas de cada celda son 65x80x75 mm.

## 5. PLC

La programación realizada en el ordenador se descarga a un autómata, quien se encarga de controlar en tiempo real el funcionamiento de la máquina en función de lo programado.

Los autómatas, o PLC's (Programmable Logic Controller), son equipos electrónicos programables capaces de controlar procesos secuenciales. Para ejecutar dicho control, los PLC's reciben señales de entrada procedentes de los sensores que controlan el proceso y emiten otras de salida para actuar en consecuencia. Estas señales pueden ser analógicas o digitales.

Las señales digitales pueden transmitir valores de TRUE o FALSE. Las señales digitales pueden transmitir valores de gran tamaño, en un rango establecido por los bits asignados a cada señal. En función del fabricante se suelen disponer de 8 o 16 bits.

Además, los PLC's permiten la comunicación de las instrucciones de los operarios, de modo que se pueda realizar un control humano sobre los procesos. Dicho control se puede ejecutar a nivel local o a través de redes de comunicación, que permiten la interconexión de autómatas y el control centralizado de estos.

El autómata utilizado en el proyecto es el MT258LF66DT4LS0 de Schneider Electric, con resolución analógica de 16 bits.

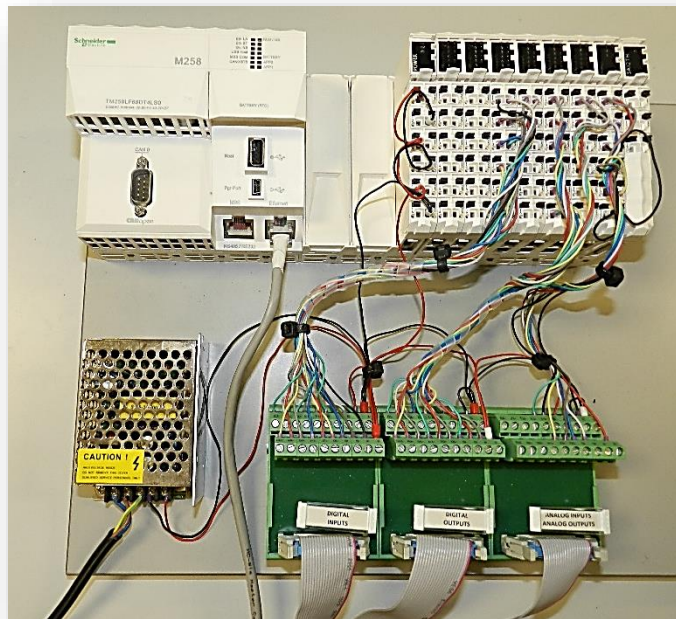


Figura 5.1  
Celda de depósito

La programación del PLC's se realiza con el programa SoMachine, también de Schneider Electric. La conexión del PLC con el ordenador se realiza a través de un cable de Ethernet.



## 6. ORDENADOR

Para la utilización del programa informático con el que se realiza la programación y control de la máquina se ha utilizado un Lenovo B590 de 2 procesadores CORE i3 de 4 GB de memoria RAM.

El programa informático con el que se programa la máquina y se diseña y realiza el control es el SoMachine, de Schneider Electric.

Se utiliza la versión 3.4.4, ya que las últimas versiones del programa no incorporan al PLC MT258LF66DT4LS0. Por tanto es necesaria la utilización de una máquina virtual con el Windows XP instalado.

La máquina virtual empleada en el proyecto es la VMware Player. La utilización de una máquina virtual para trabajar supone muchos inconvenientes. Sobre todo hace todos los procesos del ordenador más lentos y se pierde mucho tiempo.

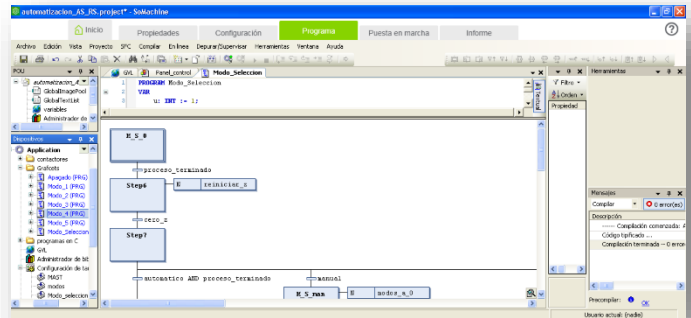


Figura 6.1  
Interfaz de programación de SoMachin

# REFERENCIAS

[www.schneider-electric.com](http://www.schneider-electric.com)

Proyecto final de carrera “CONTROL Y VISUALIZACIÓN DE UNA MÁQUINA DE 3 EJES”, autor José Antonio Gómez Oliva, año de publicación 2013.

Guía de programación SoMachine. Fecha de publicación 04/2014.

“Automatización de maniobras industriales mediante autoómatas programables”, autores Juan Pérez Cruz y Manuel Pineda Sánchez, editorial Universitat Politècnica de València.

“Maniobras en autoómatas programables”, autores Juan Pérez Cruz, Manuel Pineda Sánchez, Rubén Puche Panadero, José Roger Folch y Ángel Sapena Bañó, editorial Universitat Politècnica de València.