



# Trabajo Fin de Grado

---

Control y supervisión de un almacén automatizado en una línea de fabricación.

**Autor del proyecto:**

**Jorge Vidal Tortajada**

**Director del Proyecto:**

**Dr.Ing.Ind. Ramón Piedrafita Moreno**

**20/11/2016**



**Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza**





(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

TRABAJO DE FIN DE GRADO / FIN DE MÁSTER

D./D<sup>a</sup>. Jorge Vidal Tortajada

con nº de DNI 72983410V en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado en Ingeniería Electrónica y Automática, (Título del Trabajo) Control y supervisión de un almacén automatizado en una línea de fabricación.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 20 de Noviembre de 2016

Fdo: Jorge Vidal Tortajada



## Resumen

En el laboratorio L0.06 del departamento de Informática e Ingeniería de Sistemas de la EINA se encuentra una línea de fabricación flexible que permite recrear un proceso de fabricación industrial más o menos complejo: varias etapas de fabricación, un almacén intermedio, y una zona de expedición. Todo éste entorno está diseñado para realizar diversos tipos de manipulación de piezas pero siempre en condiciones de seguridad para que los estudiantes puedan realizar prácticas en la línea, bien de forma individual, estación aislada, o bien de forma integrada en el sistema.

Es objeto de éste proyecto diseñar e implementar el control y supervisión de la estación del almacén intermedio o estación 5, que nos permita realizar las funciones principales de un almacén como son recepción, almacenamiento, organización y emisión de pedidos.

Para ello, sobre el sistema hardware, se han programado los modos de funcionamiento industrial de la guía GEMMA, *Guide d'Etude des Modes de Marches et d'Arrêts*.

Para la gestión de los modos de funcionamiento así como la supervisión del almacén, se desarrollará también un sistema HMI (*Human Machine Interface*) que permita a un operario gestionar de forma segura el funcionamiento del almacén de forma remota mediante el terminal táctil Magelis o bien mediante la aplicación del fabricante disponible para dispositivos móviles.

El control de los modos de funcionamiento vendrá dado por el terminal Magelis debidamente programado, que nos posibilitará de forma visual mandar ordenes al autómatas para su gobierno.

De éste modo, nuestro operario dispondrá de una estación completamente funcional, en la cual podrá gestionar los eventos de almacenamiento, organización y gestión de pedidos inherente a un almacén en una línea de producción.



# Índice general

---

<b>1. Introducción</b>	<b>1</b>
1.1. Emplazamiento y situación. . . . .	1
1.2. Tipos de piezas. . . . .	2
1.3. Almacenamiento de piezas. . . . .	3
<b>2. Elementos de la estación</b>	<b>5</b>
2.1. Actuadores . . . . .	6
2.1.1. Motores paso a paso o Steppers . . . . .	6
2.1.2. Cilindro neumático de simple efecto . . . . .	7
2.1.3. Generador de vacío . . . . .	8
2.1.4. Motor de continua y cinta transportadora . . . . .	8
2.1.5. Tope fin de cinta neumático . . . . .	9
2.1.6. Neumática . . . . .	9
2.2. Sensores . . . . .	10
2.2.1. Detección del tipo de piezas . . . . .	11
2.2.2. Detección de salida de piezas . . . . .	13
2.2.3. Detección de posición del cilindro . . . . .	13
2.2.4. Detección del vacío . . . . .	14
<b>3. Hardware y Software</b>	<b>15</b>
3.1. Hardware . . . . .	15
3.1.1. Autómata industrial Premium . . . . .	16
3.1.2. Terminal Magelis XBTGT4330 . . . . .	19
3.2. Software . . . . .	20
3.2.1. Unity Pro V8.0 . . . . .	21
3.2.2. Vijeo Designer v6.2 . . . . .	21
<b>4. Funcionamiento de la estación</b>	<b>22</b>
4.1. F6- Marchas de prueba (Test) . . . . .	24
4.2. F1-Producción Normal. . . . .	25
4.2.1. Automático. . . . .	25
4.2.2. Manual. . . . .	25
4.2.3. Manual Avanzado. . . . .	26
4.3. F3-Marchas de cierre. . . . .	26

4.4. D1- Parada de emergencia. . . . .	26
<b>5. Implementación y programación.</b>	<b>28</b>
5.1. Programación del funcionamiento. . . . .	28
5.1.1. Jerarquía y conexiones. . . . .	30
5.1.2. Algoritmo de ordenación. . . . .	34
5.2. Programación de la supervisión. . . . .	35
<b>6. Conclusiones</b>	<b>39</b>
<b>Anexos</b>	<b>43</b>
<b>A. Control avanzado de un almacén de piezas</b>	<b>43</b>
A.1. Objetivos . . . . .	43
A.2. Descripción general de la estación . . . . .	43
A.2.1. Actuadores . . . . .	44
A.2.2. Sensores . . . . .	45
A.2.3. Detección del tipo de piezas . . . . .	46
A.2.4. Variables del sistema . . . . .	46
A.3. Instrucción SMOVE . . . . .	49
A.4. Ejemplo de aplicación . . . . .	50
A.5. Preparación . . . . .	52
A.6. Almacenamiento. . . . .	53



## Índice de figuras

---

1.1. Plano de la línea de producción. . . . .	1
1.2. Vista frontal del almacén. . . . .	2
1.3. Secuencia de orden Estándar. . . . .	3
1.4. Ejemplo de secuencia de orden complementaria. . . . .	4
2.1. Entorno de trabajo de la estación. . . . .	6
2.2. Motor paso a paso o Stepper . . . . .	7
2.3. Cilindro Neumático con sensores de posición . . . . .	8
2.4. Generador de vacío en la estación. . . . .	8
2.5. Detalle del tope situado al fin de la cinta. . . . .	9
2.6. Equipo de neumática . . . . .	9
2.7. Control de neumática. . . . .	10
2.8. sensores capacitivo, óptico e inductivo, por orden, a la entrada de la estación. . . . .	11
2.9. Esquemático del campo eléctrico y la capacitancia entre las placas de un detector de proximidad capacitivo en objetos dieléctricos y metálicos. . . . .	12
2.10. Representación del funcionamiento del sensor óptico. . . . .	12
2.11. Detección de salida de piezas. . . . .	13
2.12. Sensores utilizados para el movimiento. . . . .	14
3.1. Hardware de la estación. . . . .	15
3.2. Autómata Premium en la estación 5. 1-Fuente Alimentación; 2- CPU; 3- Entradas binarias; 4- Salidas binarias; 5-Control Steppers 6-Ethernet . . . . .	16
3.3. Conexión de entradas digitales al autómata. . . . .	17
3.4. Traductor Portescap ESD1200 . . . . .	19
3.5. Bastidor y esquemático del husillo. . . . .	19
3.6. Terminal Magelis XBTGT4330 . . . . .	20
4.1. Estación controlada desde el terminal Magelis. . . . .	22
4.2. Estados y transiciones de la Guía GEMMA implementados . . . . .	23
5.1. Ejemplo de red de Petri. . . . .	29
5.2. Unity Pro. 1 Ejemplo Secciones. 2 Ejemplo SFC. . . . .	29
5.3. Esquema de relaciones entre las distintas secciones programadas y el funcionamiento del sistema. . . . .	30

5.4. Ejemplo de lectura de piezas, cálculo de la posición de guardar y excepción en caso de querer sacarla. . . . .	31
5.5. Graficet de conducción implementado en el autómata. . . . .	32
5.6. Ejemplo de sección de emergencia en la estación. . . . .	32
5.7. Parte del graficet Automático. Cada secuencia gestiona un proceso. . . . .	33
5.8. Ejemplo de bloqueo en la secuencia de ordenación del almacén. . . . .	34
5.9. Sección de código en la que buscamos huecos o piezas erróneas. . . . .	35
5.10. Interfaz de programación Vijeo designer para el terminal Magelis. . . . .	36
5.11. Paneles principales de funcionamiento. . . . .	36
5.12. Paneles para el funcionamiento manual. . . . .	37
5.13. Paneles de testeo, edición y supervisión del funcionamiento. . . . .	38
A.1. Vista frontal del almacén. . . . .	44
A.2. Principales actuadores en la estación. . . . .	45
A.3. Posibles tipos de piezas en la estación. . . . .	45
A.4. sensores capacitivo, óptico e inductivo, por orden, a la entrada de la estación. . . . .	46
A.5. Lista de variables propias de cada motor paso a paso. . . . .	48
A.6. Ejemplo resumido de funcionamiento. . . . .	50
A.7. Elementos control preparación. . . . .	52
A.8. Pantalla de la terminal Magelis a utilizar en la práctica . . . . .	53

# 1. Introducción

---

## 1.1. Emplazamiento y situación.

EL desarrollo de éste TFG será llevado a cabo en el laboratorio L0.06, perteneciente al departamento de Informática e Ingeniería de Sistemas de la EINA.

En el interior del laboratorio se encuentra una línea de fabricación flexible correspondiente con la figura 1.1, que consta de 3 zonas funcionales, siendo de izquierda a derecha:

- Zona de fabricación.
- Almacén intermedio.
- Zona de expedición.

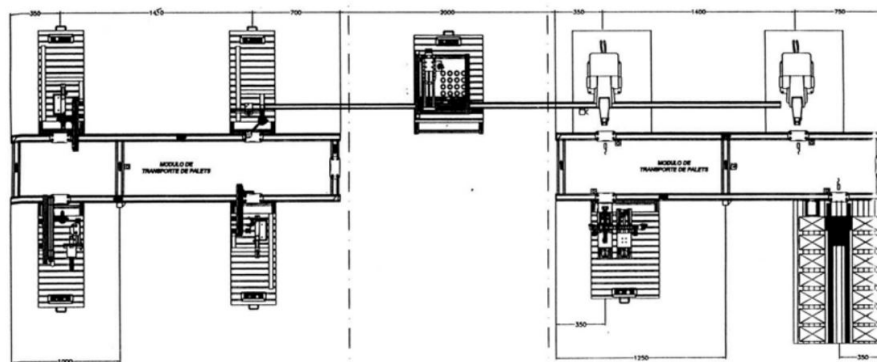


Figura 1.1: Plano de la línea de producción.

El objeto del presente proyecto es la puesta en marcha de la zona de almacén intermedio, actualmente fuera de servicio dada la sustitución del autómatas y la falta de compatibilidad del software actual con el que se programó originalmente[1].

El cometido del almacén intermedio es el almacenamiento de piezas ya fabricadas procedentes desde la cinta transportadora y su posterior entrega bajo pedido a la salida de la estación, en dirección a la zona de pedidos.

El almacén intermedio consta, como se ve en la figura 1.2 de 16 posiciones de almacenaje donde alojar a las piezas hasta su solicitud. Dichas posiciones se encuentran distribuidas en una matriz de 4 filas y 4 columnas.

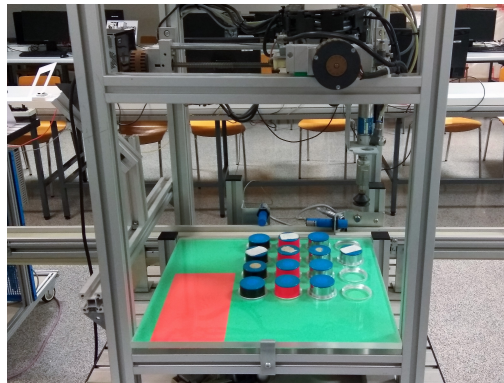
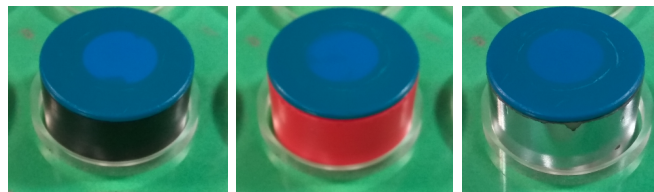


Figura 1.2: Vista frontal del almacén.

Para la gestión del movimiento de piezas en el almacén desde la posición de entrada, al almacén y a la salida, se dispone de un brazo manipulador X-Y-Z tipo pórtico, con 3 grados de libertad de los cuales los ejes X e Y son accionados mediante motores paso a paso, y el eje Z mediante un cilindro neumático. Las manipulaciones de las piezas se realizarán por medio de una ventosa de succión. Si bien los ejes X e Y se utilizarán para la selección de piezas, el eje Z será utilizado para realizar los desplazamientos en el almacén de forma segura, manteniéndose arriba durante éstos.

## 1.2. Tipos de piezas.

La línea de fabricación contempla la producción de 3 tipos de piezas (Fig.1.2) distintas, siendo éstas denominadas por su color ( negra, roja o metálica ) simulando la fabricación de productos similares pero con distintas prestaciones.



(a) Pieza Negra.

(b) Pieza Roja.

(c) Pieza Metálica.

### 1.3. Almacenamiento de piezas.

Éste almacén, tal y como indica su nombre, está situado en mitad de un proceso de fabricación y expedición de pedidos. Por ende, se entiende como un lugar que nos permite poner en consonancia el flujo de producción con el flujo de pedidos. Para ello, es vital dar una respuesta de la forma mas rápida posible, considerándose prioritaria la emisión de piezas hacia la zona de expedición frente a la recepción de las piezas procedentes de la zona de fabricación.

Teniendo en cuenta ésto, la política de almacenaje que se seguirá en éste proyecto se ha basado en minimizar los tiempos de emisión de piezas, colocandose éstas preferiblemente en la posición mas cercana a la posición de salida (Fig.1.3), sin tenerse en cuenta el tipo.

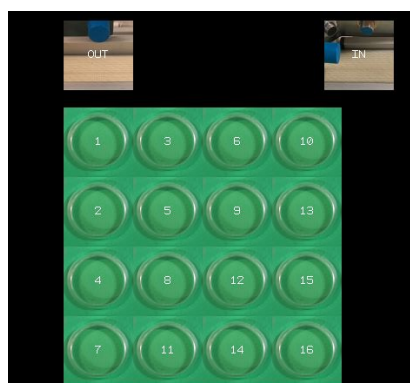


Figura 1.3: Secuencia de orden Estándar.

A su vez, si bien de éste modo se favorece minimizar el tiempo de emisión de piezas, no asegura una distribución homogénea de ellas, pudiendo fallar en caso de que hallan sido fabricadas en orden, primero de un tipo y luego de otro. Para paliar éste posible problema, se ha diseñado un sistema de ordenación de modo que a cada columna se le asignará un tipo de pieza, siendo la 4a columna utilizada como almacén en caso de exceso de algún tipo de pieza, según se muestra en la figura 1.4.

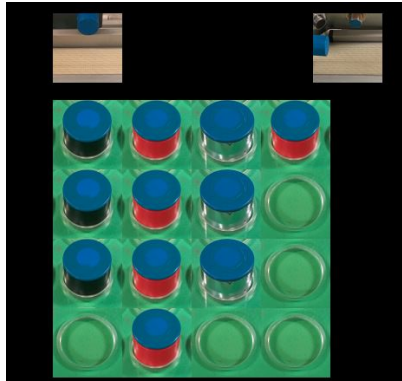


Figura 1.4: Ejemplo de secuencia de orden complementaria.

El trabajo se ha dividido en 5 capítulos:

- **Elementos de la estación:** En este capítulo se van a explicar las características más relevantes de la estación de las que los elementos de control automático podrán hacer uso.
- **Hardware y Software:** En este capítulo se va a dar una visión global de la estructura hardware integrada en la estación y se van a explicar las características básicas de el software que ha sido utilizado en la programación y diseño.
- **Funcionamiento de la estación:** En este capítulo se van a analizar los diferentes modos de funcionamiento implementados en la estación, así como la relación entre ellos.
- **Implementación y programación:** Aquí es donde se explica como se ha llevado a acabo la integración de dichos modos de funcionamiento así como la gestión de la estación por un operario.
- **Conclusiones:** En este capítulo se recogen las conclusiones obtenidas con este TFG.

## 2. Elementos de la estación

---

Para poder realizar cualquier tipo de control programado, es necesario contar con algún tipo de elemento que de información del sistema o proceso que queremos controlar, así como algún elemento que permita realizar alguna acción sobre el mismo. Éstos elementos se denominan comúnmente sensores y actuadores. Los utilizados en la estación (Fig. 2.1) serán los siguientes:

- Actuadores
  - Motores paso a paso o Steppers
  - cilindro neumático simple efecto
  - Generador de vacío
  - Motor de continua y cinta transportadora
  - Tope fin de cinta neumático
  - Neumática
  
- Sensores
  - Detección del tipo de piezas
  - Detección de salida de piezas
  - Detección de posición del cilindro
  - Detección del vacío

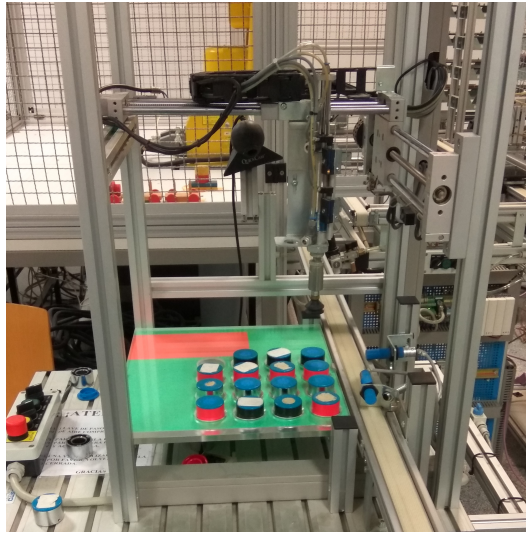


Figura 2.1: Entorno de trabajo de la estación.

Además, dada la naturaleza de alguno de ellos, la estación dispone de un sistema de neumática que será el que realmente nos permita ejercer las acciones, controladas electrónicamente. Posee un sistema de regulación para la neumática así como un depósito de aire a presión y un compresor para su correcto funcionamiento.

## 2.1. Actuadores

Para llevar a cabo cualquier tarea en un autómatas industrial será necesario el uso de actuadores. Los actuadores son, por definición, elementos controlados que nos permiten modificar el estado del sistema dando lugar a un gran abanico de posibilidades de actuación.

Dada la naturaleza de nuestra aplicación, que se puede resumir en el movimiento de piezas entre un almacén y una cinta transportadora, contaremos con un sistema de actuadores que nos permita realizar la acción de transporte de piezas en el almacén y gestionar la entrada y salida. Para ello, los actuadores utilizados serán los siguientes:

### 2.1.1. Motores paso a paso o Steppers

Para realizar los movimientos en los ejes X e Y se utilizarán dos motores paso a paso o Steppers (Fig.2.2) Portescap P532.258.004.03.





(a) Ubicación Stepper en el eje X o Vía 0 (b) Primer plano Stepper. Vista lateral.

Figura 2.2: Motor paso a paso o Stepper

El motor paso a paso o *Stepper* es un dispositivo electromecánico que convierte una serie de pulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de girar una cantidad de grados (paso o medio paso) dependiendo de sus entradas de control. [2]

Ya que éstos motores no cuentan con ningún sistema de realimentación para el control de la posición, es necesario un control electrónico preciso que permita asegurar que en las condiciones de carga y aceleración de trabajo no se pierda ningún paso. De una forma muy simplificada, las variaciones en las consignas de control serán del tipo rampa, hasta una velocidad constante, siempre sin pasar los límites del motor.

Gracias a la combinación del módulo de control de posición de los motores y el traductor Portescap correctamente configurados podemos realizar el control de los motores con las librerías propias del software.

Cabe mencionar, que la salida del motor tenemos un husillo o tornillo sin fin de circulación de bolas cuyo avance lineal por vuelta es de 3 mm. En combinación con el paso de nuestros motores,  $3.6^\circ$ , da lugar a que con una configuración *half step* como es la nuestra, 200 pasos equivaldrían a una vuelta, siendo ésto 3mm de avance real.

### 2.1.2. Cilindro neumático de simple efecto

Para el movimiento en el eje Z, se ha utilizado un cilindro neumático de simple efecto (Fig.2.3), largo recorrido, y con vástago hueco para llevar el vacío a una ventosa colocada en su extremo. De éste modo permitiremos un movimiento a lo largo del eje con dos posiciones: una alta ,para el movimiento, y otra baja, para la sujeción de piezas.



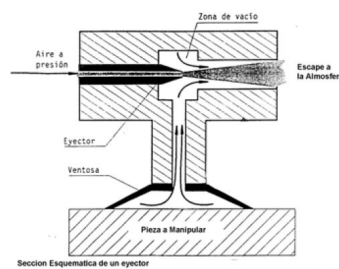
Figura 2.3: Cilindro Neumático con sensores de posición

### 2.1.3. Generador de vacío

Para el transporte de las piezas, utilizaremos la ventosa situada en el vástago del cilindro neumático. Para ello, generaremos vacío gracias al módulo Festo (Fig. 2.4(a)) y al efecto Venturi (Fig. 2.4(b)). *Grosso modo* realizar un escape de aire a presión aguas arriba del émbolo, arrastrará el aire del interior del émbolo generándose así una depresión o vacío.



(a) Módulo Festo con silenciador.



Sección Esquemática de un eyector

(b) Diagrama efecto Venturi.

Figura 2.4: Generador de vacío en la estación.

### 2.1.4. Motor de continua y cinta transportadora

Para poder establecer una interacción entre las distintas estaciones de la línea, así como de abastecer y emitir pedidos de piezas en la estación, contaremos con una cinta transportadora desde la entrada hasta la salida de la estación accionada por un motor de continua. El control

del motor se realizará desde el autómatas mediante una de sus salidas a relé, teniendo éste un único sentido de giro en caso de ser accionado.

### 2.1.5. Tope fin de cinta neumático

El tope fin de cinta (Fig.2.5), como se puede intuir por su nombre, nos permitirá la acción de retener las piezas al final, de modo que se permita su paso mediante la activación de la salida en el autómatas. El mecanismo consiste en un cilindro del simple efecto activado por una electroválvula que al activarse permite el paso de las piezas a lo largo de la cinta.

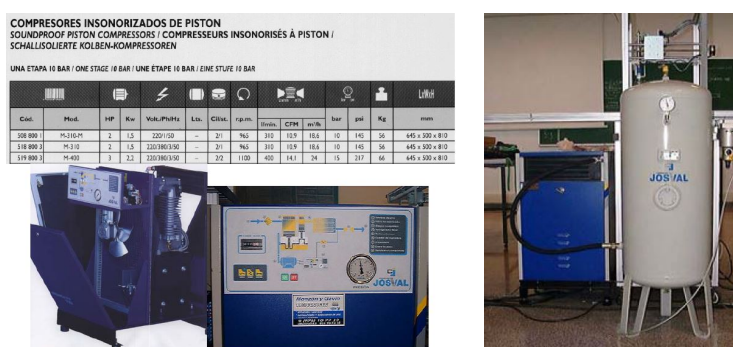


(a) Tope cerrado. Bloquea el paso de piezas. (b) Tope abierto. Permite el paso de piezas.

Figura 2.5: Detalle del tope situado al fin de la cinta.

### 2.1.6. Neumática

Como hemos visto, varios de nuestros actuadores tienen componente neumática. El uso de la neumática viene justificado por el hecho que mediante un correcto control de las válvulas, con un sólo equipo compresor podemos alimentar a diferentes actuadores.

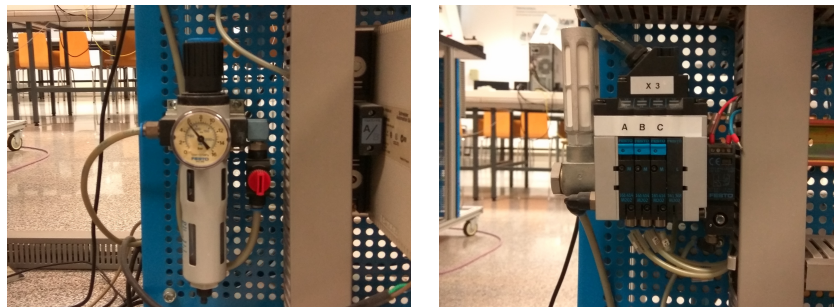


(a) Compresor Josval y principales características (b) Depósito aire comprimido.

Figura 2.6: Equipo de neumática

Para ello, se ha utilizado un equipo compresor Josval M-310 (Fig.2.6(a)) con un motor de 2 HP, de alimentación trifásica, un caudal de 310L a una presión de 10 bar. Además se cuenta con un depósito de aire comprimido de 100L (Fig.2.6(b)) que nos mantendrá la presión en el sistema sin tener el motor funcionando continuamente.

Por otra parte, para regular la entrada al sistema y proteger la línea y estación de posibles fugas, se incluye un regulador de la presión con llave de paso (Fig.2.7(a)), en cuya salida estará el control de las 3 electroválvulas (Fig.2.7(b)) que controlan los actuadores neumáticos: control del tope, control del cilindro de doble efecto y control de la generación de vacío.



(a) Regulador de caudal a la entrada de la estación. (b) Accionadores mecánicos de las electroválvulas.

Figura 2.7: Control de neumática.

## 2.2. Sensores

Para identificar las distintas etapas del funcionamiento, utilizaremos el uso combinado de sensores dando lugar así a posibles eventos en el funcionamiento, que clasificaremos, según su uso, como la identificación de piezas o identificación de movimientos.

- Como bien indica su nombre, los sensores destinados a la identificadores de pieza, situados a la entrada, se utilizarán para realizar la gestión del almacén y que el sistema pueda memorizar el tipo de pieza que entra.
- Mientras que los sensores encargados de la identificación de movimientos nos servirán de cara a:
  - gestionar el correcto funcionamiento de los actuadores en movimiento.
  - evaluar si procesos muy concretos como la extracción de piezas, se han llevado a cabo <sup>1</sup>

<sup>1</sup>Una vez introducida la pieza en la estación, no hay forma de comprobar el tipo mas que teniendo en cuenta el valor memorizado.



Figura 2.8: sensores capacitivo, óptico e inductivo, por orden, a la entrada de la estación.

### 2.2.1. Detección del tipo de piezas

Para realizar la tarea de identificación de piezas se utilizará la combinación de 3 sensores distintos: capacitivo, óptico e inductivo (Fig.2.8) situados a la entrada de la estación.

A modo resumen, las diferentes combinaciones posibles de entrada son las siguientes, debiendo ser consideradas el resto como erróneas.

Tipo de pieza	Capacitivo	Óptico	Inductivo
No Pieza	0	0	0
Pieza Negra	1	0	0
Pieza Roja	1	1	0
Pieza Metálica	1	1	1

Cuadro 2.1: Tabla donde se recogen las entradas del sistema en función del tipo de pieza

Un poco más en detalle, en funcionamiento de cada uno de los sensores mencionados será descrito a continuación.

#### 2.2.1.1. Capacitivo:

El sensor capacitivo consta de un condensador que posee un campo eléctrico entre sus placas. Éste condensador forma parte de un circuito resonador, de manera que cuando un objeto se acerca a este campo, la capacidad aumenta y el circuito empieza a resonar (Fig.2.9). En la estación se dispone de dos, uno a la entrada y otro en la posición de salida del almacén.

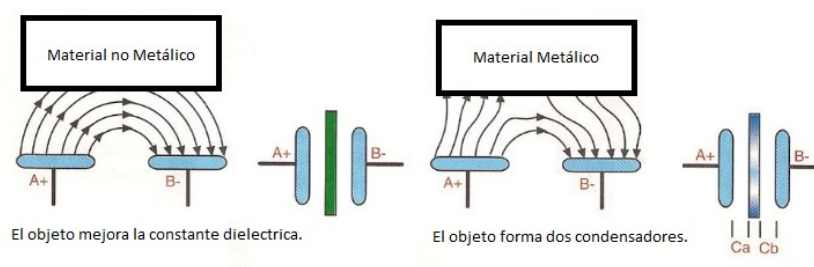


Figura 2.9: Esquemático del campo eléctrico y la capacitancia entre las placas de un detector de proximidad capacitivo en objetos dieléctricos y metálicos.

Gracias a éste comportamiento, nos permitirá identificar si hay o no alguna pieza en la cinta, justo en la entrada.(Fig.2.8)

### 2.2.1.2. Óptico:

El sensor óptico consta de un emisor de luz infrarroja y un receptor(Fig.2.10). En presencia de un objeto, según el color, la luz reflejada tendrá una energía u otra pudiendo así mediante una calibración establecer un umbral de detección de piezas mas oscuras o mas claras<sup>2</sup>. Con la calibración realizada para el sensor en la entrada, permitirá distinguir si la pieza situada en la cinta es de algún color ( en nuestro caso tanto roja como metálica ) o no ( negra ).



Figura 2.10: Representación del funcionamiento del sensor óptico.

### 2.2.1.3. Inductivo:

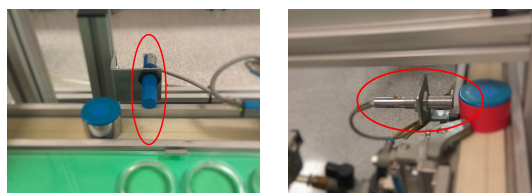
El sensor inductivo (Fig.2.8, consta de un núcleo magnético excitado en tensión, lo que genera un campo magnético que en presencia de un material metálico inducirá corrientes denominadas de Foucault, lo cual reduce la corriente por el sensor, pudiendo detectarse así la presencia de dicho material. Por lo que como no podría ser de otra manera, estará activo siempre que la pieza sea metálica.

<sup>2</sup>El color de un cuerpo depende del espectro de luz absorbida y reflejada, siendo los colores oscuros los que mas luz absorben, o lo que es lo mismo: menos energía reflejan.

### 2.2.2. Detección de salida de piezas

La detección de salida de piezas se realizará mediante un sensor capacitivo situado en la posición de salida del almacén (Fig.2.11(a)) que simplemente nos permitirá verificar si realmente se ha realizado la operación de emisión de una pieza, siendo en ningún caso posible la comprobación del tipo.

Además, situado al final de la cinta transportadora (Fig.2.11(b)) disponemos de un sensor óptico. A diferencia del caso de la entrada, éste está calibrado para detectar cualquier tipo de pieza, permitiéndonos así identificar si hay alguna pieza esperando a avanzar a la siguiente estación o no.



(a) Sensor Capacitivo a la salida de la estación. (b) Sensor óptico a la salida de la estación.

Figura 2.11: Detección de salida de piezas.

### 2.2.3. Detección de posición del cilindro

La identificación de posición del cilindro nos permite saber si algún proceso se ha finalizado, pudiendo dar por concluida una salida en caso de que fuere necesario.<sup>3</sup>

Para la detección de posición del cilindro, utilizaremos sensores magnetointuctivos para conocer si se encuentra arriba, abajo o en la posición *Home* y finales de carrera para seguridad en el final del recorrido.

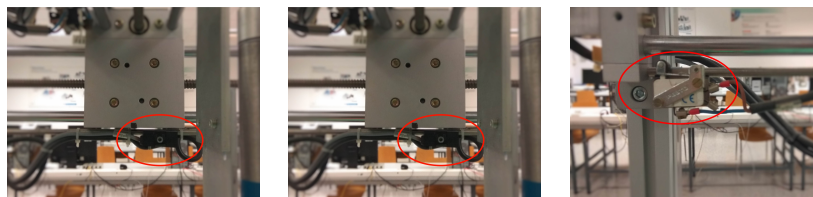
#### 2.2.3.1. Detección de posición:

Para la detección de la posición se utilizan los sensores magnetointuctivos, a diferencia de los finales de carrera, detectan sin contacto objetos de metal. De éste modo, colocando un imán en el émbolo del cilindro de simple efecto o brazo, y otros dos en el bastidor solidario al brazo podremos saber mediante los detectores, situados en la parte superior e inferior del brazo (Fig.2.12(a)) y en la posición *Home* (Fig.2.12(b)), si el brazo está en un sitio u otro.

<sup>3</sup>Véase en el caso de bajar brazo, una vez detectado que está abajo, no es necesario mantener la salida activa al ser un cilindro de doble efecto.

### 2.2.3.2. Seguridad final de recorrido:

Para la seguridad en el fin del recorrido utilizaremos los finales de carrera. Los finales de carrera son micro interruptores activados por leva situados en los límites de movimiento (Fig.2.12(c)) de los ejes X e Y, controlado por los steppers, que en caso de producirse un desplazamiento hasta dicha posición, activará el interruptor. En éstos casos particulares, los sensores serán cableados directamente al módulo de control de los steppers provocando automáticamente la detención del movimiento en ese eje. Esta gestión la realiza de forma autónoma el propio módulo de control.



(a) Detector inductivo situado en la posición Home.  
(b) Detector inductivo situado en la posición Home.  
(c) Fin de carrera situado en el eje.

Figura 2.12: Sensores utilizados para el movimiento.

### 2.2.4. Detección del vacío

Para medir el vacío, Festo, fabricante de la mayor parte de componentes presentes en la estación, cuenta con sensores que hacen la función de presostato y vacuostato. En este caso, gracias al vacuostato mediremos el vacío generado en el interior del cilindro para poder concluir, una vez superado un umbral, si una pieza ha sido correctamente absorbida.



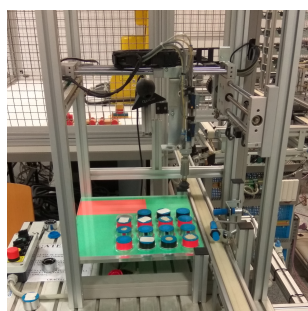
## 3. Hardware y Software

---

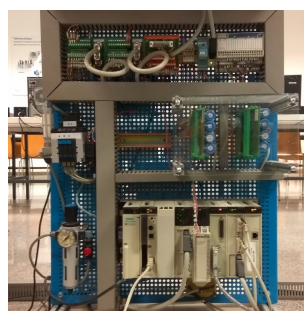
### 3.1. Hardware

Para llevar a cabo el control del proceso de producción y almacenamiento en el almacén intermedio (Fig.3.1(a)) en la línea de producción se ha utilizado , junto a los sensores y actuadores del capítulo anterior, un autómata industrial (Fig.3.1(b)) y una terminal Magelis (Fig.3.6) del fabricante Schneider Electric. Los principales elementos Hardware serán:

- Autómata industrial Premium
  - Fuente de alimentación TSX PCY 2600
  - Módulo CPU TSX P57 4634M
  - Módulo Entradas Binarias TSX DEY 16FK
  - Módulo Salidas Binarias TSX DSY 08R5A
  - Módulo Control Steppers TSX CFY21
  - Módulo Ethernet TSX ETY 5103
- Traductor Portascap ESD1200
- Terminal Magelis XBTGT4330



(a) Zona de trabajo de la estación. Parte superior.



(b) Elementos de control de la estación. Parte inferior.

Figura 3.1: Hardware de la estación.

### 3.1.1.1. Autómata industrial Premium

Los autómatas o PLCs ("Programmable Logic Controller") son las unidades básicas implementadas extensamente en la industria como elementos de control y gobierno del funcionamiento de cualquier proceso industrial.

Disponen principalmente de módulos de entradas y de salidas, tanto analógicas como digitales, que permiten dar una combinación de respuestas frente a una combinación de entradas según la lógica programada, así como módulos de comunicación para el envío de datos.

Su uso en la industria viene justificado principalmente por sus buenas prestaciones en cantidad de E/S, su capacidad de control y comunicación así como su diseño específico para uso en ambientes hostiles frente al posible uso de micro-controladores que disponen de un reducido número de E/S o un PC industrial cuyo precio es mucho mas elevado.

En nuestro caso, montado sobre un rack extensible de 8 posiciones TSX RKY 8EX, encontraremos los siguientes módulos de la familia TSX Premium (Fig.3.2).

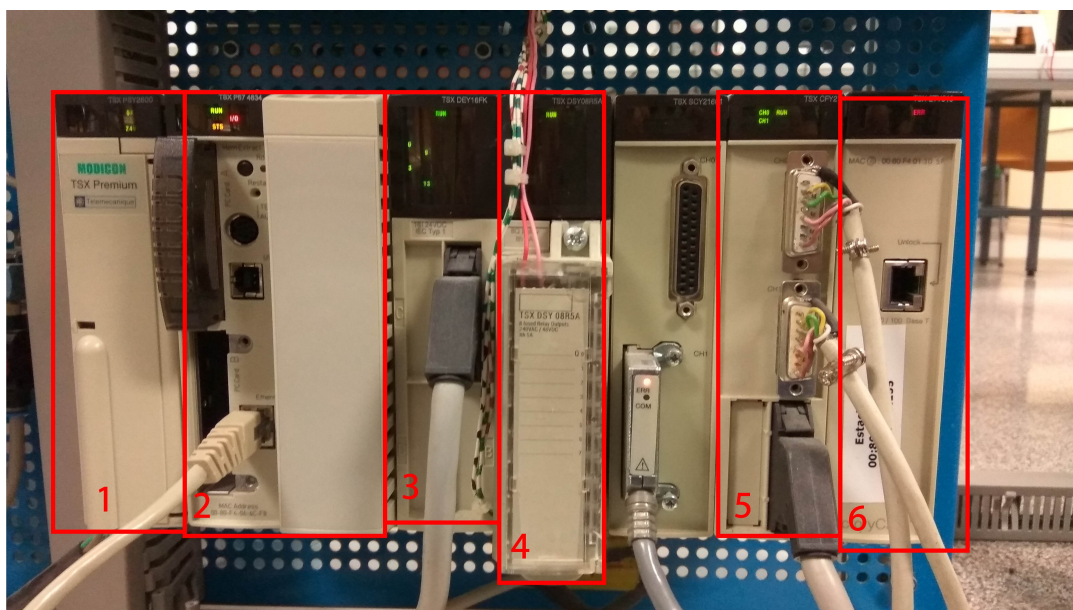


Figura 3.2: Autómata Premium en la estación 5. 1-Fuente Alimentación; 2- CPU; 3- Entradas binarias; 4- Salidas binarias; 5-Control Steppers 6-Ethernet

#### 3.1.1.1.1. Fuente de Alimentación TSX PCY 2600

La estación dispone de dos fuentes de alimentación, una integrada en el rack del autómata [3], correspondiente con el módulo TSX PSY 2600 del fabricante Schneider Electric.

La primera da servicio al resto de módulos presentes en el rack. Alimentamos la fuente desde la red (220V AC) para dar salida a los diferentes módulos del autómata, alimentados a 24V DC.

La segunda, da servicio a los sensores y actuadores de la estación, con similares características.

### 3.1.1.2. Módulo CPU TSX P57 4634M

El módulo CPU es el que se encargará de interpretar y ejecutar los comandos del software transferido, en nuestro caso, mediante USB o TCP IP.

Éste módulo cuenta con 2040 E/S discretas, 256 E/S analógicas, 64 canales de aplicación específica, 440Kb de memoria integrados y 2048Kb a través de PCMCIA. Por otra parte incluye conexión mediante USB tipo B y Ethernet 802.3.

### 3.1.1.3. Módulo Entradas Binarias TSX DEY 16FK

Módulo de 16[4] entradas digitales con filtrado de ruidos y rebotes individual configurable que da servicio a los sensores de la estación.

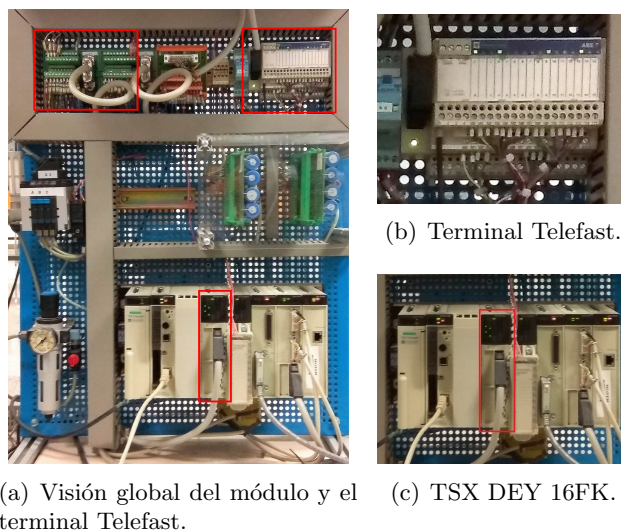


Figura 3.3: Conexión de entradas digitales al autómata.

En la Figura 3.3(a) podemos ver que dichas entradas están cableadas mediante un terminal TeleFast (Fig.3.3(b)), lo que permite una alta densidad de entradas por módulo de autómata (Fig.3.3(c)) así como su fácil cableado y visualizado in situ (leds verdes indicadores del estado de la señal correspondiente).

#### 3.1.1.4. Módulo Salidas Binarias TSX DSY 08R5A

Módulo que proporciona al autómatas 8 salidas a relé de 24V DC y hasta 5A dándole así la capacidad de controlar los diferentes actuadores enunciados anteriormente, a excepción de los motores paso a paso o *steppers*, que vendrán controlados por el siguiente módulo. Dado el bajo número de salidas necesarias: electroválvulas del cilindro, tope final, vacío, y motores de la cinta transportadora; éstas han sido cableadas directamente al módulo.

#### 3.1.1.5. Módulo Control Steppers TSX CFY21

Módulo de control de los motores paso a paso que permite a la estación el posicionamiento a lo largo de todo el recorrido, sin restricciones. Éste módulo facilita la programación al tener incorporado en el software los comandos para el movimiento controlado de dichos motores. <sup>1</sup>

A su vez, dicho módulo será conectado a al traductor Portscap ESD1200. <sup>2</sup>

Dispone de entradas de fin de recorrido, punto de origen o *Home* y de sucesos que permite controlar el desplazamiento. Las principales entradas al módulo serán punto de origen, fin de recorrido y parada de emergencia y como salidas del módulo la entrada de control del traductor y las salidas de impulsos, velocidad y sentido de giro.

Para el correcto funcionamiento del módulo se ha de configurar todos los parámetros de los que dispone, explicados en el apartado de software.

#### 3.1.1.6. Módulo Ethernet TSX ETY 5103

Los módulos Ethernet disponen [5]de una interfaz estándar para la conexión a una red 10/100BASET y, en el panel frontal, de un conector RJ45 para la conexión pin a pin mediante un cable de enlace compuesto por dos pares trenzados independientes. Entre otras, provee al autómatas de servicios como intercambio de mensajes, intercambio de datos entre estaciones, sitio web (http de hasta 7.5 Mb) y notificación por correo electrónico.

#### 3.1.1.7. Traductor Portscap ESD1200

Utilizaremos los traductores Portscap ESD1200 (Fig.3.4 )como tarjetas de control de los motores. Los traductores Portscap son tarjetas multipropósito para motores paso a paso las cuales incluyen las etapas de potencia ( puente en H principalmente ) que permiten el funcionamiento del motor en configuraciones *Half step* o *Full step*. Ésta tarjeta puede trabajar hasta

---

<sup>1</sup>Al tratarse de motores sin encoder es preciso un minucioso cálculo de aceleraciones y velocidad para el control de posición.

<sup>2</sup>Tarjeta multipropósito para el control de motores paso a paso . Véase sección 3.1.1.7. Traductor Portscap ESD1200.

30KHz, a 60 V DC y 2A de salida por fase.

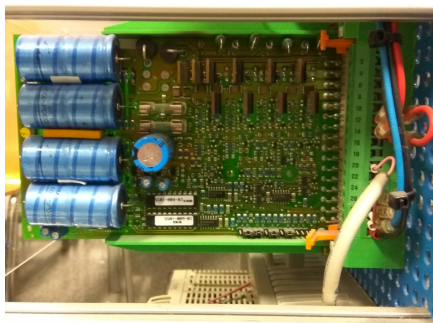
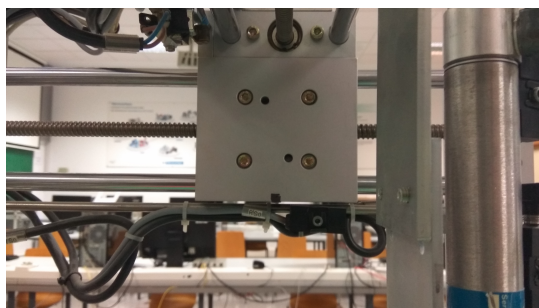


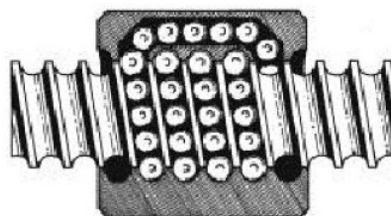
Figura 3.4: Traductor Portescap ESD1200

Gracias a la combinación del módulo de control de posición de los motores y el traductor Portescap correctamente configurados podremos realizar el control de los motores con las librerías propias del software.

Cabe mencionar, que la salida del motor tenemos un husillo o tornillo sin fin de circulación de bolas (Fig.3.5) cuyo avance lineal por vuelta es de 3 mm. En combinación con el paso de nuestros motores,  $3,6^\circ$  dando lugar a que con una configuración Half step como es la nuestra, 200 pasos equivaldrían a una vuelta, siendo éste 3mm de avance real.



(a) Bastidor del husillo asociado al eje X.



(b) Esquemático del mecanismo.

Figura 3.5: Bastidor y esquemático del husillo.

### 3.1.2. Terminal Magelis XBTGT4330

La terminal Magelis XBTGT4330 es un terminal gráfico táctil en el que se implementa mediante una conexión de red TCP-IP un sistema *Human Machine Interface* para la supervisión del funcionamiento del autómatas por un operario. Gracias a éste terminal y a un buen desarrollo de la interfaz el operario encargado de la estación podrá de forma remota controlar el funcionamiento en cualquier situación sin la necesidad de estar presente, o detener el funcionamiento para un mantenimiento o reparación.

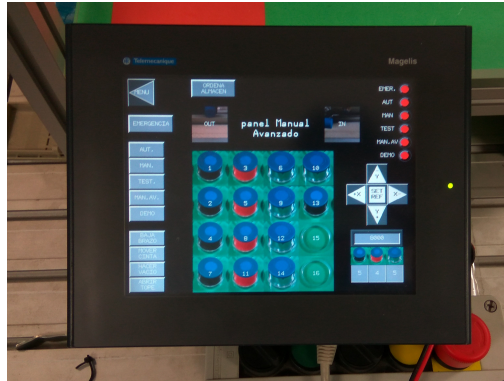


Figura 3.6: Terminal Magelis XBTGT4330

Tamaño de pantalla	7,5"
Designación de software	Vijeo Designer
Sistema operativo	Magelis
Frecuencia de procesador	266 MHz
Descripción de memoria	Memoria de aplicaciones flash EPROM 32 MB Copia seg. datos SRAM 512 kB batería litio
Conexiones	Puerto maest. USB tipo A (V1.1) Salida audio blq term rosca extrbls 3 salidas digitales blq term rosca extrbls Ethernet TCP/IP RJ45 Alimentación blq term rosca extrbls Enl. serie COM1 RJ12/M8 RS232C/RS422/RS485 $\leq 115,2 \frac{kbits}{s}$ Enl. serie COM2 RJ45 RS485 $\leq 187,5 \frac{kbits}{s}$
Tensión de alimentación	19.2...28.8 V
Consumo de potencia	28 W

Cuadro 3.1: Principales características técnicas Magelis

Utilizaremos la conexión RJ45 con el protocolo TCP/IP para transferir en programa desde el ordenador a la magelis. Una vez configurada la conexión y transferido el programa a la magelis mediante vijeo designer y al autómatas mediante Unity Pro, podremos realizar el control desde el terminal, sin necesidad del uso del PC.

## 3.2. Software

Para realizar el control del autómatas, y por ende de la estación, utilizaremos el software del propio fabricante Unity Pro V8.0 y Vijeo Designer V6.2.

### 3.2.1. Unity Pro V8.0

Unity Pro V8.0 es un software de programación, depuración y operación común para las gamas Modicon Premium, Atrium and Quantum PLC. Ofrece acceso a todos los servidores de objetos de software de Unity Pro y Unity Studio. Permite desarrollar soluciones a medida para facilitar la colaboración de Unity con el sistema CAD eléctrico, el generador de códigos y variables, etc.

Unity Pro combina la comunicación y gestión de los módulos del autómatas a la vez que permite la programación en los diferentes lenguajes:

- Texto estructurado (ST)
- Lenguaje Grafcet o *Sequential function chart* (SFC)
- Diagrama de contactos o *Ladder Diagram* (LD)
- Diagrama de bloques de función secuencial (FBD)

Todos estos posibles lenguajes se pueden combinar en pro de la facilidad de implementación, dotando de una capacidad de realizar tareas con una complejidad mayor de forma más sencilla y facilitando la legibilidad del programa.

### 3.2.2. Vijeo Designer v6.2

Vijeo Designer V6.2 es un software *Human Machine interfaz* con el cual mediante un terminal Magelis se dará la posibilidad al usuario de controlar el proceso de funcionamiento así como modificar el modo de funcionamiento y gestionar las alarmas.

Vijeo Designer permite de una forma fácil e intuitiva desarrollar una interfaz gráfica de control con un alto contenido visual, simplificando así su uso. Entre muchas otras funciones, permite crear botones, visualizadores, eventos de alarma, pantallas emergentes, modificación de variables, etc. de forma flexible pudiendo darle a cada elemento distintos modos de actuación. Por ejemplo, si un botón se mantiene pulsado o no, si se lanza una acción al pulsar, mientras esta pulsada, o al soltar; o si algún elemento sólo debe ser visible bajo ciertas condiciones de funcionamiento. Todas estas funciones dotan al programa de gran flexibilidad de cara a un desarrollo preciso y complejo en su funcionamiento a la par de intuitivo para su uso.

## 4. Funcionamiento de la estación

---

Nuestra estación se ha diseñado para realizar la función de almacén intermedio entre el proceso de fabricación de piezas y el proceso de expedición de pedidos. Por ello, implementaremos diversas funciones inherentes a un almacén, que podríamos simplificar en:

- Recepción de piezas.
- emisión de piezas.
- Ordenación del almacén.

Desde la zona de fabricación recibiremos piezas a almacenar en cualquier orden y emitiremos piezas según pedido del operario. Además, se implementará un sistema de ordenación del almacén en función del tipo de pieza.

Estas 3 funciones básicas se podrán realizar esencialmente de 2 formas: automáticamente, sin que el operario tenga que hacer los movimientos, o manualmente, en la que el operario realizará los movimientos que desee. Los modos de funcionamiento se controlarán desde la terminal Magelis pudiendo intercambiarse durante el funcionamiento. Además, por razones obvias de seguridad, se implementará un modo de emergencia que detendrá automáticamente el funcionamiento, permitiendo así un uso seguro de la estación ya sea in situ o de forma remota.

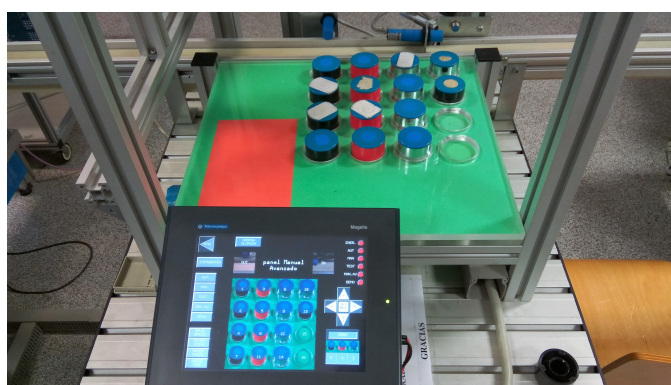


Figura 4.1: Estación controlada desde el terminal Magelis.



En éste vídeo de funcionamiento global podemos ver un ejemplo del funcionamiento de la estación, combinando los distintos modos.

Todas estas operaciones han sido implementadas acorde a la guía GEMMA [6] (*Guide d'Etude des Modes de Marches et d'Arrêts*) que establece la secuencia de funcionamiento genérica de puesta en marcha, funcionamiento, y paradas en un autómatas industrial. Para ser mas concretos, los modos de funcionamiento implementados acorde con la guía GEMMA son:

- A1-Parada en estado inicial.
- A6-Puesta del sistema en estado inicial.
- F1-Producción normal.
- F2-Marcha de preparación.
- F3-Marcha de cierre.
- F6-Marchas de prueba (Test)
- D1-Parada de emergencia

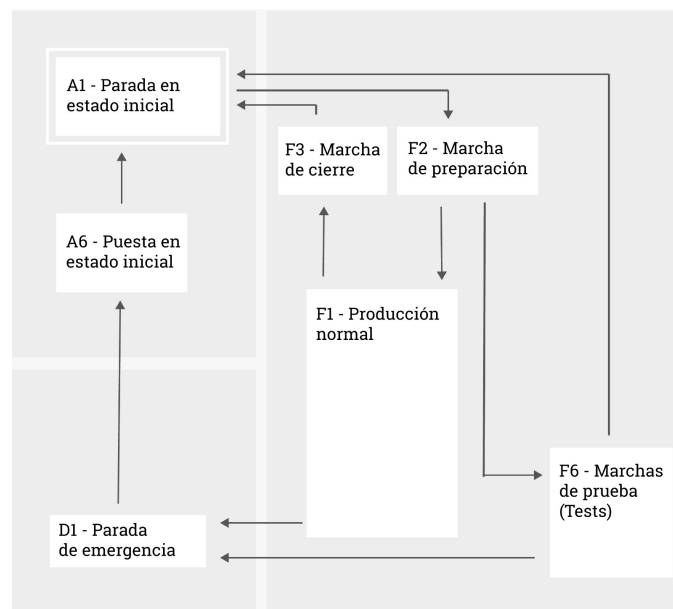


Figura 4.2: Estados y transiciones de la Guía GEMMA implementados

El uso de la guía GEMMA establece los posibles ciclos de operación del autómatas, empezando éste siempre en el estado A1- Parada en el estado inicial. El estado inicial se corresponde

con el reposo en la estación, siendo la primera vez necesaria realizar una preparación ( F2 de la guía GEMMA), referenciar los ejes X e Y correspondientes con el movimiento de los *Steppers*. Una vez realizado dicho proceso, podremos acceder a los distintos modos de funcionamiento:

#### 4.1. F6- Marchas de prueba (Test)

Modo desde el cual se comprobará el correcto funcionamiento de los actuadores y sensores del sistema, comprobando la acción y respuesta que dispone de una pantalla propia en la Magelis para su control. Dados los sensores y actuadores de los que se dispone, las diferentes marchas de prueba serán:

- **Movimiento en los ejes:**

Ordenaremos mover los ejes X e Y secuencialmente hasta los límites marcados por los finales de carrera de modo que la trayectoria sea un cuadrado, y aumentando cada vez la velocidad. En caso de superar el tiempo límite ( acorde a la velocidad del movimiento) lanzaremos una alarma al sistema en función de en qué movimiento haya fallado.

Una vez realizados satisfactoriamente dichos movimientos, mandaremos el brazo a la posición de reposo y testaremos el movimiento en el eje Z del brazo. Se realizará el mismo proceso con el detector inductivo.

- **Detección en sensores:**

El testeo de los sensores capacitivo, óptico, inductivo y de vacío se realizará manualmente y tendrá que ser el usuario el que verifique su correcto funcionamiento visualizando la salida del sensor( en otras palabras, si detecta o no correctamente).

- **Funcionamiento actuadores:**

Tanto para la apertura del tope como el movimiento de la cinta, no se dispone de un sensor que nos dé esa información. Es por ello, que el modo de test sólo consistirá en la posibilidad de accionamiento manual por parte del usuario para la comprobación.

En caso de que haya algún fallo en el testeo, se dispondrá de un historial de avisos en la sección de emergencia en la que el operario podrá identificar que actuador o sensor ha fallado y cuando.

En éste vídeo de [marchas de test](#) podemos ver un ejemplo de test de la estación, combinando los distintos modos.

## 4.2. F1-Producción Normal.

la producción normal se supone que es el estado permanente del autómatas. Es aquí donde se realizarán las operaciones de entrada/salida de piezas al almacén, así como la gestión del stock. Dentro del modo de funcionamiento normal nos encontraremos con varios posibles modos de manejo, en función de la elección del operario mediante el terminal Magelis.

Una primera distinción a realizar será en función de cómo se realizarán los movimientos, siendo posible los siguientes modos, acordes con las pantallas implementadas.

### 4.2.1. Automático.

En el modo automático el operario sólo se encargará del control de las piezas que salen, siendo la recepción de piezas sin necesidad supervisión alguna, automática. Aquí se realizará un registro autónomo del movimiento de las piezas, sabiéndose su posición en el almacén.

El almacenamiento de las piezas se ha diseñado de modo que se reduzcan los tiempos de preparación de pedidos. La posición calculada será siempre la mas cercana a la posición de salida, tanto para la entrada como salida de las piezas <sup>1</sup> de modo que los tiempos de transporte se minimicen, y flexibilizando la capacidad de almacenamiento al no imponerse piezas en posiciones fijas<sup>2</sup>.

El usuario podrá realizar pedidos de forma genérica (seleccionando el tipo) o de forma concreta ( seleccionando qué pieza quiere sacar). Otra función que podrá realizar será el cambio de posición de una pieza en el almacén.

Dado que los eventos de salida/entrada de piezas pueden ser simultáneos, se ha diseñado una política de gestión que da prioridad al sacado de piezas, mejorando de éste modo la disponibilidad de espacio en el almacén; o bien si la pieza que entra es del tipo que se desea sacar, directamente sale fuera sin pasar por el interior del almacén.

En éste vídeo del modo automático podemos apreciar un ejemplo de funcionamiento automático de la estación.

### 4.2.2. Manual.

En éste modo de funcionamiento, el operario podrá moverse hacia las posiciones predeterminadas de entrada, almacenamiento y salida, así como hacer uso de los actuadores de los que dispone la estación. Dado que los movimientos no van necesariamente asociados a ningún tipo

---

<sup>1</sup>Salvo que se solicite una pieza concreta.

<sup>2</sup>Los procesos de almacenamiento con un tamaño preasignado son poco flexibles frente a la variabilidad de la producción al no poder gestionar un exceso de stock de forma eficiente.

de lógica <sup>3</sup>, en éste modo se pierde la gestión autónoma de las posiciones del almacén.<sup>4</sup>

Éste vídeo ejemplifica el modo manual de funcionamiento de la estación, en coordinación la pantalla de administrador.

### 4.2.3. Manual Avanzado.

El modo manual avanzado está diseñado pensando en necesidades un poco mas complejas. Incorpora el funcionamiento manual, y le añade:

- Posibilidad de movimiento en las dos vías hacia cualquier posición dentro del rango de movimiento.
- Calibrado de la posición de referencia.<sup>5</sup>
- Ordenación del almacén. Se colocará cada pieza en la columna que tenga preasignada siguiendo el patrón de la figura siguiente, pudiendo ser los huecos vacíos rellenos exclusivamente por el tipo de pieza preasignado. La última columna se utilizará como almacén para el exceso de algún tipo de pieza en caso de ser necesario.

Éste vídeo se corresponde con el funcionamiento manual avanzado de la estación, donde podemos observar el calibrado del brazo y la ordenación del almacén.

## 4.3. F3-Marchas de cierre.

Como se ha mencionado anteriormente, el funcionamiento en permanente se corresponde con el estado F1-Producción normal. Si se utiliza el modo manual, cualquier movimiento que se produzca es a cargo del operario, de modo que se presupone controlado. Si se utiliza el modo automático, el último movimiento del autómatas es desconocido, por lo que para evitar problemas en el arranque<sup>6</sup> se llevará el brazo cada vez que se salga de dicho modo a la posición de referencia.

## 4.4. D1- Parada de emergencia.

Para preservar la seguridad y el correcto funcionamiento del almacén, se han implementado diferentes eventos de alarmas, asociados al funcionamiento de la estación. Principalmente, son

---

<sup>3</sup>El hecho de ir a una posición y activar el vacío no tiene por qué significar coger una pieza.

<sup>4</sup>Para evitar el desacople de la información con la realidad se ha implementado una pantalla , Administrador, en la que el operario pueda actualizar el registro.

<sup>5</sup>El correcto calibrado es esencial para el funcionamiento de la estación pues las posiciones y por ende los movimientos van relacionados a la posición de referencia.

<sup>6</sup>Ver capítulo de programación. sección del módulo PAP.

eventos asociados a una temporización, en nuestro caso por encima de un umbral, el cual se considera el tiempo máximo de realizar la operación correspondiente.

En caso de entrar en éste modo, se detendrán automáticamente el movimiento de la estación hasta que el operario compruebe y verifique que el sistema está en condiciones de continuar con la producción.

## 5. Implementación y programación.

---

Para implementar el funcionamiento explicado en el capítulo anterior, 4 **Funcionamiento de la estación**, se ha utilizado el software descrito en la sección 3.2 **Software**.

Siendo mas concretos, la programación se ha realizado por un lado en:

- **Unity Pro V8.0** para realizar el código que le da la funcionalidad al autómata y por ende a la estación, o dicho de otra manera, establecer la lógica de actuación frente a los sucesos que ocurran en el almacén.
- **Vijeo Designer V6.2** para que un operario pueda supervisar, controlar y modificar el funcionamiento del almacén de forma remota mediante una interfaz visual e intuitiva.

### 5.1. Programación del funcionamiento.

Haciendo alusión al apartado 3.2 **Software**, Unity (Fig.5.2) brinda la posibilidad de programar el funcionamiento de varias maneras, dada la multitud de lenguajes de programación que admite.

Atendiendo a criterios de legibilidad del código, robustez y versatilidad, la programación se ha realizado principalmente en dos de los lenguajes de los que dispone: Lenguaje Grafcet o *Sequential function chart* (SFC) y Texto estructurado (ST), combinados entre sí. Bien cada uno en una sección<sup>1</sup>, o bien anidando código en ST dentro de una sección SFC.

El lenguaje SFC es un lenguaje de programación normalizado en la norma IEC 61131-3, el cual está basado en las redes de Petri (Fig.5.1). Éste lenguaje cuenta con 3 elementos básicos: Etapas o estados, acciones y transiciones.

---

<sup>1</sup>Una sección es una parte de código, del lenguaje que sea, que se ejecuta continuamente durante cada ciclo del autómata

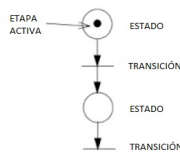


Figura 5.1: Ejemplo de red de Petri.

En cada estado se ejecutará una acción asociada<sup>2</sup>, y se cambiará de estado siempre que se cumpla alguna condición de transición contigua al estado activo. Es un lenguaje de programación diseñado para procesos secuenciales, por lo que lo implementaremos para controlar los estados de la guía GEMMA, marcha de preparación, modo automático, test, y manual avanzado.

El lenguaje ST es, al igual que el SFC, es un lenguaje de programación normalizado en la norma IEC 61131-3. Éste, basado en la programación en PASCAL y utilizado también en el control industrial. Se asemeja a la programación típica, la cual no es necesariamente secuencial. De éste modo podremos de una forma mas intuitiva darle complejidad al sistema, al ser similar al típico lenguaje de programación con sentencias del tipo *FOR*, *IF*, *WHILE*, etc.

Es por estas estructuras que en éste lenguaje se ha implementado la gestión de los sensores y paneles para la magelis, gestión del almacén, gestión de emergencias, y el método de ordenación del almacén.

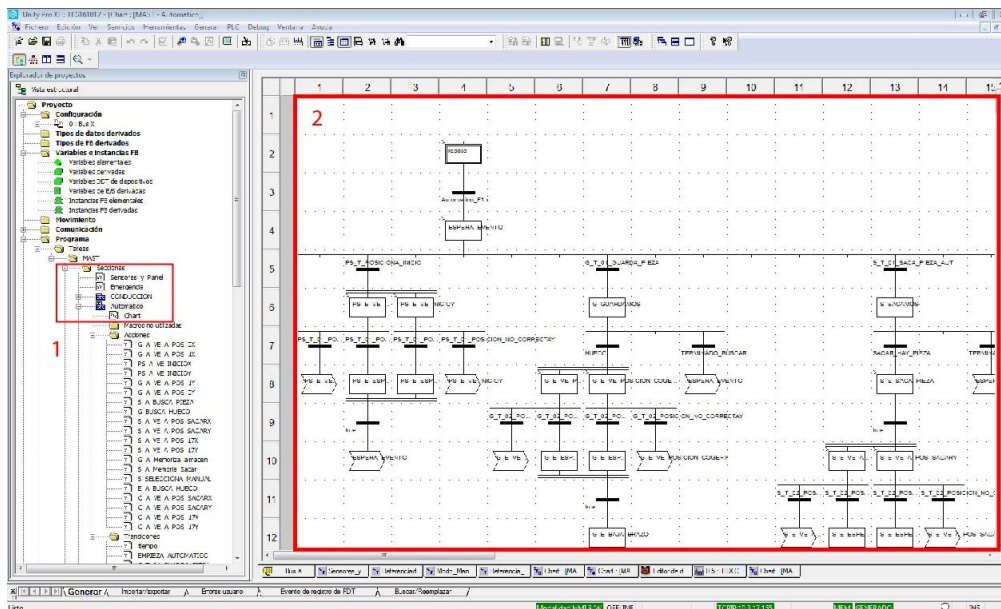


Figura 5.2: Unity Pro. 1 Ejemplo Secciones. 2 Ejemplo SFC.

<sup>2</sup>Pudiendo ser temporal o mantenida, sobre un actuador o sobre una variable, según su declaración

### 5.1.1. Jerarquía y conexiones.

Cada una de las secciones se ejecuta simultáneamente, de modo que todas las secciones están activas y pueden relacionarse. Las tareas o procesos relacionados con la gestión y supervisión de la estación han sido programadas en ST mientras que las de funcionamiento han sido programadas mediante SFC. Las relación siendo el SFC el que active procesos críticos como la actualización de valores del almacén y la propia sección ST la que calcule las acciones a ejecutar por el SFC, según la figura 5.3.

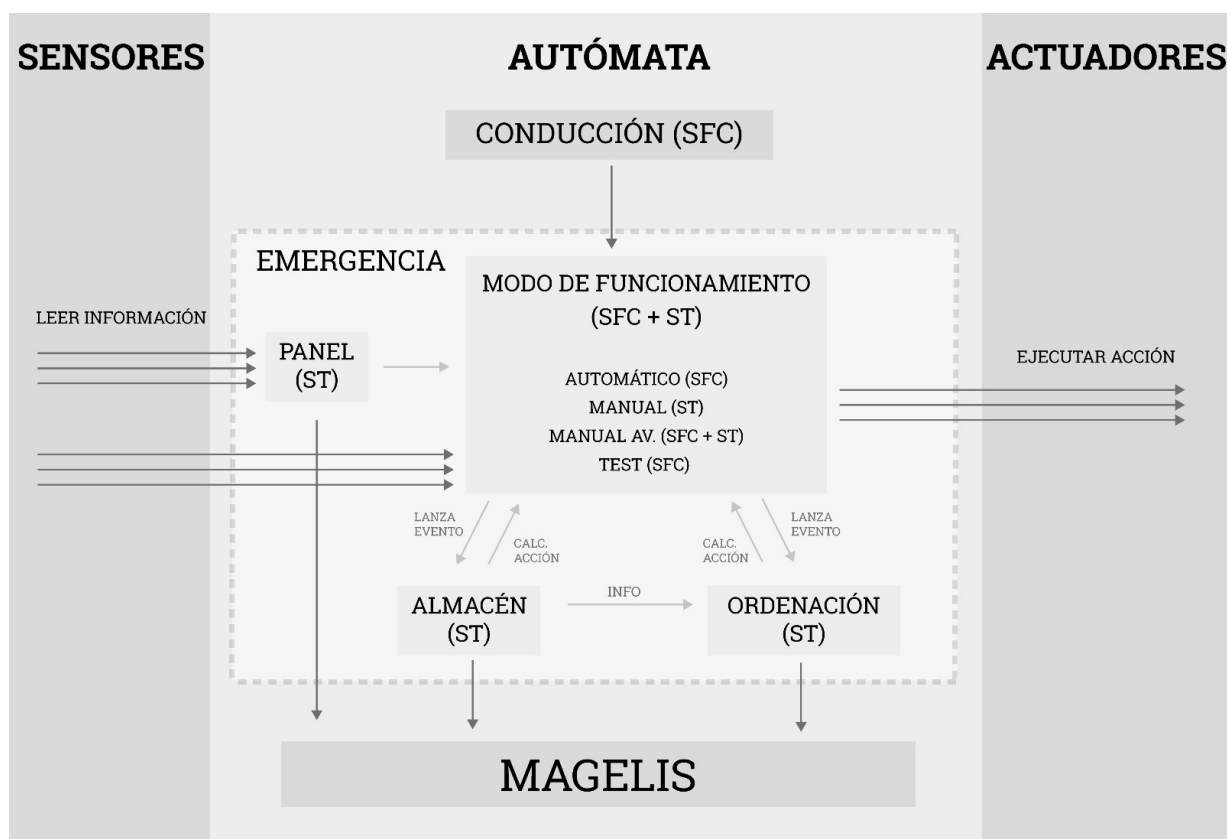


Figura 5.3: Esquema de relaciones entre las distintas secciones programadas y el funcionamiento del sistema.

- **Sensores y Paneles: ST**

En ésta sección se implementan las variables que nos permiten identificar piezas (no pieza o pieza del tipo que corresponda), siendo éstas resultado de la combinación de los sensores correspondientes. A su vez, dispone de variables asociadas a los estados de la guía GEMMA para informar en qué modo de funcionamiento está (p.e. automático, manual, test, etc) y lee el estado de las posiciones del almacén. Ésta sección se encarga



de recopilar información para comunicar con la magelis y los modos de funcionamiento.

#### ■ Almacén: ST

En ésta sección nos encargaremos de memorizar los valores del almacén<sup>3</sup>. En el modo automático, además, se encarga de la gestión del movimiento de piezas, calculando de dónde a dónde tiene que mover una pieza(Fig.5.4). La activación del cálculo de las posiciones vendrá dado por el propio SFC al activar temporalmente la condición IF asociada<sup>4</sup> en donde se realizan dichos cálculos.

```

IF (P_PIEZA_NEGRA) THEN                                (* LEEMOS EL TIPO DE PIEZA *)
  GUARDA_TIPO_PIEZA:=1;
  ELSIF (P_PIEZA_ROJA) THEN
    GUARDA_TIPO_PIEZA:=2;
  ELSIF (P_PIEZA_METALICA) THEN
    GUARDA_TIPO_PIEZA:=3;
  ELSE
    GUARDA_TIPO_PIEZA:=0;
  END_IF;

  FOR i:=16 TC 1 BY -1 DO                               (* BUSCAMOS EL HUECO, MAS CERCANO A SALIDA *)
    if (ALMACENESTADC[i] = 0 ) then
      POS_GUARDAR := i;
      HUECO:=true;
    END_IF;
  END_FOR;

  IF (SACA_TIPO_PIEZA=GUARDA_TIPO_PIEZA) THEN          (* SI ES DEL TIPO QUE QUERAMOS SACAR, LA SACAMOS*)
    POS_GUARDAR:=17;
    P_PIEZA_SACANDO:=GUARDA_TIPO_PIEZA;
    HUECO:=TRUE; (*CAMBIAR VARIABLE A OTRA, HUECO Y "POSIBLE"???? *)
  END_IF;

  IF (NOT HUECO) THEN                                   (* SI NO HAY HUECO EN EL ALMACEN, NO SE PUEDE GUARDAR *)
    TERMINADO_BUSCAR:=TRUE;
  ELSE
    TERMINADO_BUSCAR:=FALSE;
  END_IF;
END IF;

```

Figura 5.4: Ejemplo de lectura de piezas, cálculo de la posición de guardar y excepción en caso de querer sacarla.

#### ■ Conducción: SFC

El Grafcet de conducción, como se ha mencionado anteriormente, se encargará de controlar los modos de arranque, marcha y paro de la guía GEMMA, garantizando el correcto funcionamiento del sistema.

<sup>3</sup>Leer qué pieza entra y memorizarla en la posición que sea o bien eliminarla del registro del almacén si se extrae.

<sup>4</sup>Según si es una operación de almacenamiento o emisión de piezas, se activará temporalmente en el SFC la variable que inicie el calculo en el ST de las posiciones en las que mover la pieza correspondiente.

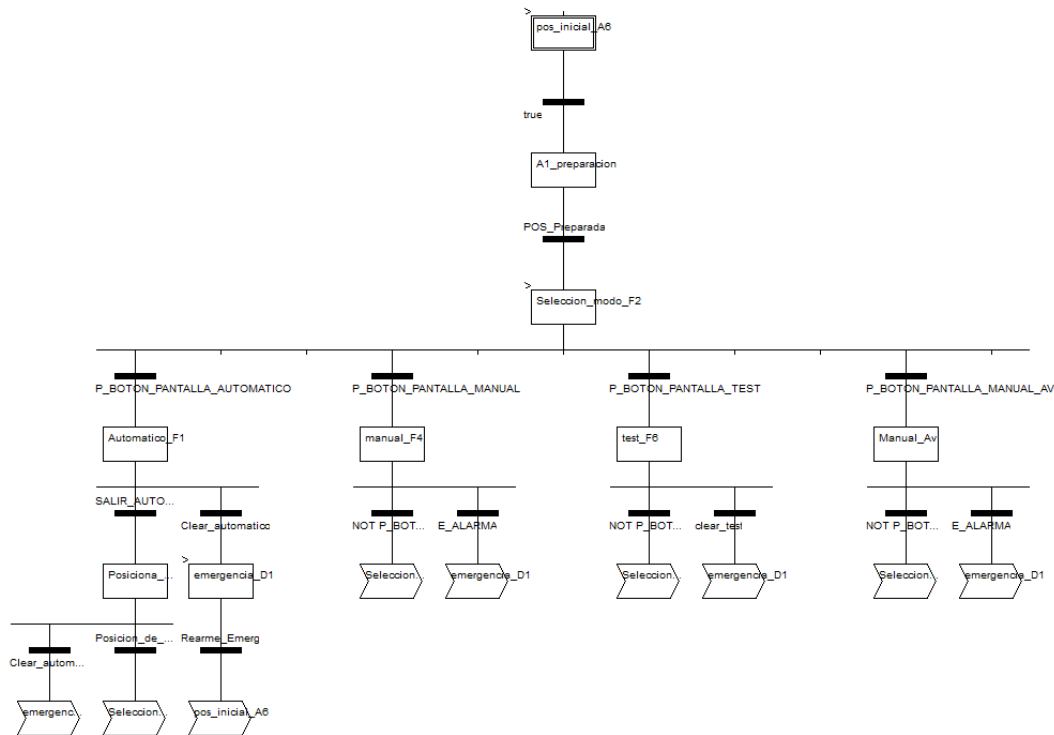


Figura 5.5: Grafcet de conducción implementado en el autómata.

#### ■ Emergencia: ST

La sección de emergencia se encargará de evaluar de forma continua todas las posibles causas de error en el sistema (Fig.5.6), en cuyo caso cambiará el estado del sistema en el que se encuentre a el estado "D1-Emergencia" de la guía GEMMA <sup>5</sup> y detendrá cualquier movimiento o proceso en curso.

```
(*POSIBLES EMERGENCIAS*)
E_ALARMA:= P_BOTON_EMERGENCIA or Not I_Seta_Emergencia;

E_N_AUT:=      (*EMERGENCIA NEUMATICA MODO AUTOMATICO*)

      (* PARTE GUARDAR PIEZAS*)
(G_E_BAJA_BRAZO.t>t#3s and G_E_BAJA_BRAZO.x)
or (G_E_COGE_PIEZA.t>t#5s and G_E_COGE_PIEZA.x)
or (G_E_COGIDA.t>t#3s and G_E_COGIDA.x) or (G_E_COGIDA.x and not I_SensVacio )
or (G_E_SUBE_BRAZO.t>t#1s and G_E_SUBE_BRAZO.x) or (G_E_SUBE_BRAZO.x and not I_SensVacio )
or (G_E_VE_A_ALMACENARX.t>t#5s and G_E_VE_A_ALMACENARX.x) or (G_E_VE_A_ALMACENARX.x and not I_SensVacio )
or (G_E_VE_A_ALMACENARY.t>t#5s and G_E_VE_A_ALMACENARY.x) or (G_E_VE_A_ALMACENARY.x and not I_SensVacio )
or (G_E_ESPERAX_2.t>t#3s and G_E_ESPERAX_2.x) or (G_E_ESPERAX_2.x and not I_SensVacio )
or (G_E_ESPERAY_2.t>t#3s and G_E_ESPERAY_2.x) or (G_E_ESPERAY_2.x and not I_SensVacio )
or (G_E_BAJA_BRAZO1.t>t#3s and G_E_BAJA_BRAZO1.x) or (G_E_BAJA_BRAZO1.x and not I_SensVacio )
or (G_E_SUELTA_PIEZA.t>t#1s and G_E_SUELTA_PIEZA.x)
or (G_E_SUBE_BRAZO1.t>t#1s and G_E_SUBE_BRAZO1.x)
```

Figura 5.6: Ejemplo de sección de emergencia en la estación.

<sup>5</sup>Estado con nombre análogo en el grafcet de conducción.

- **Modos de funcionamiento:**

- **Automático: SFC**

El modo de funcionamiento automático se ha programado de forma secuencial (Fig.5.7), atendiendo a los diferentes comportamientos que se han considerado:

- Almacenamiento de piezas.
- Extracción de piezas de un tipo, de una posición concreta o sin pasar por el interior del almacén.
- Movimiento de una pieza desde una posición deseada a otra conocida.

En caso de que se desee dar por concluido el modo automático, éste terminará la tarea que esté llevando a cabo y procederá a realizar la marcha de cierre consistente en el posicionamiento

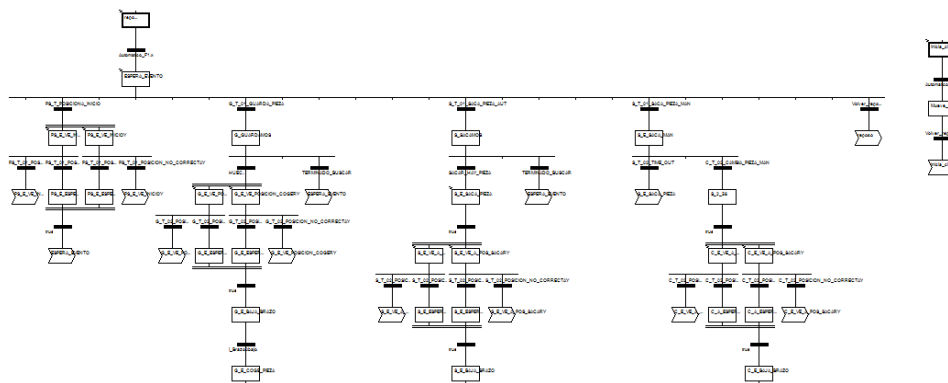


Figura 5.7: Parte del grafcet Automático. Cada secuencia gestiona un proceso.

- **Manual: ST**

el modo manual ha sido implementado como una acción en ST asociada al estado del grafcet de conducción correspondiente.

- **Manual Avanzado: SFC + ST**

El modo manual avanzado, al igual que el modo manual, parte de las acciones que lleva a cabo son asociadas a una acción ST en el estado propio del Grafcet de conducción en el que además de las funciones del manual, se complementa con la posibilidad del uso de los comandos de calibración manual del autómatas<sup>6</sup>, permitiendo hacer desplazamientos reducidos o cambiar la posición de referencia para los movimientos. Por otro lado, éste modo dispone de la ordenación del almacén, tarea para la cual se ha implementado una sección SFC propia consistente en el intercambio de posición de una pieza en el almacén. Al tratarse de un proceso complejo, en ésta sección sólo se realizará el movimiento de las piezas con las posiciones calculadas<sup>7</sup> e iniciará el proceso de cálculo, llevado a cabo en la sección de Ordenación del almacén.

<sup>6</sup>Mediante las instrucciones JOG y RP\_HERE, propias del módulo del autómatas.

<sup>7</sup>Se calculará la posición de origen y de destino.

- **Ordenación del almacén: ST**

El proceso de ordenación del almacén es un proceso complejo, dada la gran variabilidad de las condiciones iniciales que debe de afrontar el algoritmo. Se trata de un algoritmo complejo que debe de asegurar un movimiento mínimo de piezas para converger en la solución y que sea capaz de lidiar con mínimos locales, como en la figura 5.8 en la que el único movimiento de piezas posible<sup>8</sup> no lleva a una situación mas ordenada en sí mismo, sólo teniendo en cuenta el movimiento que se va a realizar, teóricamente no conduce a ordenar el almacén.

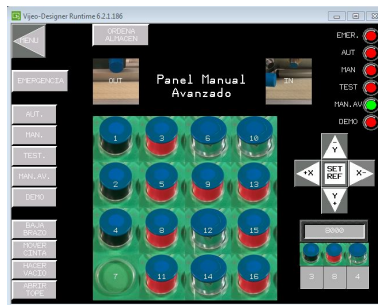


Figura 5.8: Ejemplo de bloqueo en la secuencia de ordenación del almacén.

### 5.1.2. Algoritmo de ordenación.

Como ya hemos dicho, nuestro algoritmo ha de ser capaz de poder ordenar el almacén bajo cualquier circunstancia siendo clave discernir primeramente posibles casos de partida así como definir a qué se considera ordenado.

Se ha entendido por ordenado el hecho de tener las piezas de un tipo en una columna fija<sup>9</sup>, siendo la 4ª tomada como un *buffer* de piezas en exceso, y en las 3 primeras columnas de modo que estén lo mas cerca posible a la cinta transportadora.

La secuencia de ordenamiento será la siguiente:

- Eliminar excesos de piezas si lo hubiere: en caso de que haya mas piezas de un tipo de las que se pueden almacenar, se sacarán al exterior del almacén. De este modo se generará al menos un hueco, con el que podremos asegurar el poder ordenar el almacén.
- Posteriormente comprobaremos si se dispone de algún hueco o no. mientras haya un hueco podemos asegurar que el almacén se puede ordenar de alguna forma. En caso de

<sup>8</sup>En la figura podemos observar que intercambiando las piezas número 9 y 10, tendríamos el almacén ordenado, pero para ello habría que mover antes una de las dos al hueco disponible, siendo que éste movimiento utiliza una posición que no debiera ser ocupada.

<sup>9</sup>Siendo el orden de izquierda a derecha : negra, roja, metálicas y *buffer* de excesos

no disponer de un hueco en el almacén, se sacará una pieza del *buffer* con la cinta parada, pudiéndolo usar temporalmente como posición de almacenaje hasta ordenar el almacén, en cuyo momento se volverá a introducir la pieza en el almacén en la posición que le corresponda, dando por concluido el proceso.

- Una vez se disponga de un hueco (Fig.5.9), comenzaremos con el movimiento de piezas en el interior del almacén. El algoritmo se ha desarrollado de manera que se busque dónde está el primer hueco indebido<sup>10</sup>, de modo que si está en una posición asociada a un tipo, se buscará una pieza de ese tipo en posición incorrecta, y en caso de estar en el *buffer*, buscaremos la primera incorrecta. Este proceso se repetirá hasta converger en la solución.

```

IF (HUECO) THEN
  (* SI HAY HUECO: SI ES EN UNA COLUMNA EXCLUSIVA: RELLENAMOS CON PIEZA CORRESPONDIENTE
  SI ES EN LA ULTIMA COLUMNA: MOVEMOS PRIMERA PIEZA INCORRECTA*)

  (*PRIMERO: BUSCAMOS HUECO Y MIRAMOS DONDE ES*)
  FOR j:=1 TC 4 BY 1 DC
    FOR i:=1 TC 4 BY 1 DC

      IF (ALMACEN_MATRIZ[i][j]=j AND j<4) THEN
        E_CONTADOR_FILA[j]:=E_CONTADOR_FILA[j]+1;
        END_IF;

      IF (ALMACEN_MATRIZ[i][j]=0) THEN
        IF ((E_CONTADOR_FILA[j]=CONTADOR_PIEZA[j] AND j<4)) THEN
          EXIT;
        END_IF;

        POS_GUARDAR:=POS_ALMACEN_MATRIZ[i][j];
        TIPO_HUECO:=j;
        EXIT;
      END_IF;
    END_FOR;
  IF (TIPO_HUECO<>0) THEN
    EXIT;
  END_IF;
END_FOR;
(* SI EL HUECO ES EN J<>4, BUSCAMOS UNA PIEZA DE ESE TIPO*)

```

Figura 5.9: Sección de código en la que buscamos huecos o piezas erróneas.

- Por último y como solución a un posible bloqueo, en caso de disponer de algún hueco pero no concluir que pieza se ha de mover, se moverá la primera pieza del *buffer* a esa posición, permitiéndonos deshacer el bloqueo y seguir ordenando hasta finalizar.

## 5.2. Programación de la supervisión.

Como ya se ha mencionado, una vez implementado el funcionamiento en el autómata, se utilizará el terminal Magelis para que el operario pueda realizar las tareas de puesta en marcha y supervisión del sistema. Para ello, ha sido necesaria la programación en vijeo (Fig.5.10) y

<sup>10</sup>Si el contador de piezas de ese tipo es mayor que el número de piezas de ese tipo en la columna propia.

definición de las variables claves para el control de la estación que se han de compartir entre el autómata y la Magelis, bien sea para editarlas o bien para poder ver el estado de la estación.

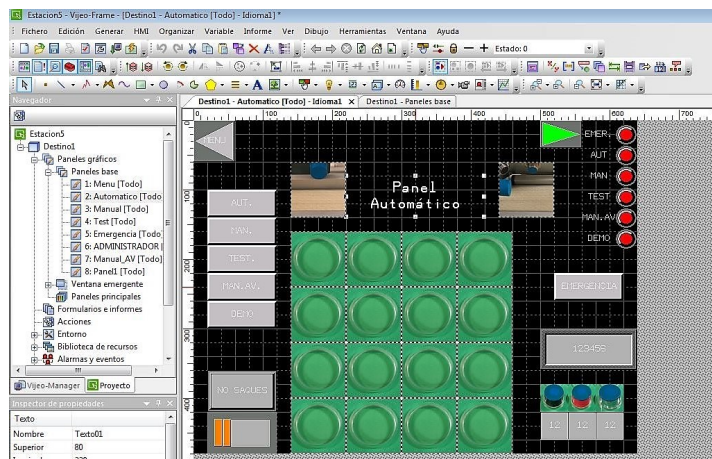


Figura 5.10: Interfaz de programación Vijeo designer para el terminal Magelis.

Con ésta finalidad, se han desarrollado diversas pantallas para cada modo de funcionamiento mediante las cuales el usuario podrá realizar todas las acciones disponibles en ese modo. Siendo éstas las siguientes:

- **Menú principal** (Fig.5.11(a)). Para la navegación entre las pantallas así como para realizar la marcha de preparación
- **Modo Automático**(Fig.5.11(b)). Suponiéndose éste el modo de funcionamiento normal donde podremos configurar el pedido de piezas, seleccionando el tipo abajo a la izquierda, o seleccionando la posición concreta pulsando sobre la pantalla, así como cambiar de posición una pieza dentro del almacén seleccionando la posición de origen y de partida.

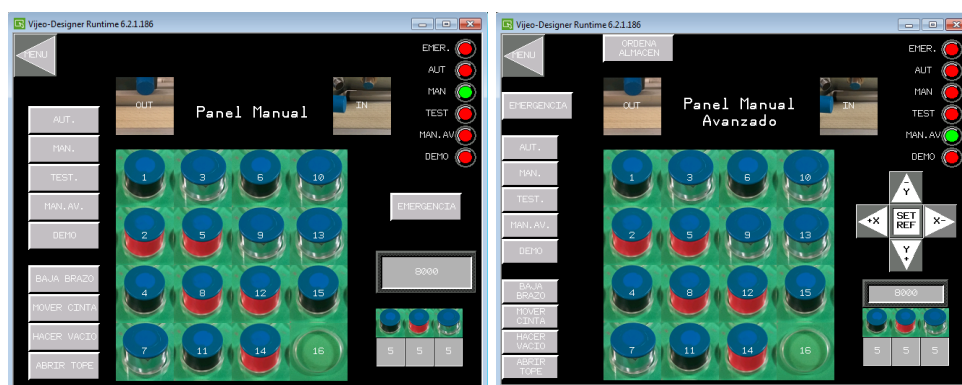


(a) Menú principal.

(b) Modo automático.

Figura 5.11: Paneles principales de funcionamiento.

- **Modo Manual.** (Fig.5.12(a)). Permitirá el accionamiento de los diferentes actuadores de forma manual, así como desplazamiento a las posiciones predefinidas de almacenaje y entrada o salida en la cinta pulsando la posición deseada.
- **Modo Manual Avanzado.** (Fig.5.12(b)). Incorpora las opciones del modo manual y además nos permite lanzar el ordenamiento del almacén ( parte superior izquierda) o el movimiento de modo manual del brazo de forma libre mediante las flechas, así como redefinir la posición de referencia para nuestros desplazamientos.



(a) Modo Manual.

(b) Modo Man. Avanzado.

Figura 5.12: Paneles para el funcionamiento manual.

- **Modo Test.** (Fig.5.13(a)). Permitirá lanzar las marchas o acciones que nos permiten comprobar el correcto funcionamiento de los actuadores y sensores bajo la supervisión del operario. Dará marcha a cada uno de los posibles tests, indicando cual está en curso, en error, o si ha acabado.
- **Modo Admin** (Fig.5.13(b)). Para corregir los posibles fallos en las piezas memorizadas, cambiando el valor memorizado en cada posición en caso de algún posible fallo de escritura en el movimiento de piezas o una posible identificación errónea a la entrada.
- **Modo Emergencia.** Para revisar el historial de fallos y poder determinar cual ha sido la causa de error en el funcionamiento, y continuar el funcionamiento una vez revisado el historial de fallos. (Fig.5.13(c))





## 6. Conclusiones

---

El objetivo de este TFG es el desarrollo del proceso de control y supervisión de un almacén en una línea de producción que emula a una industrial, basado en el uso de diversos actuadores y sensores genéricos disponibles, permitiendo que un uso combinado de la información proporcionada por cada uno de forma individual, de lugar a un ambiente de producción en el que un operario de forma remota pueda controlar la estación con la certeza de que en caso de que hubiere algún fallo, sería notificado y detenido el funcionamiento en condiciones seguras tanto para el operario, para el rearme, y para la producción, al minimizarse el funcionamiento con fallos.

Estos objetivos se han conseguido de la siguiente forma:

- Puesta en marcha de los actuadores, sensores y módulos del PLC del Almacén.
- Configuración y uso del módulo de control de los motores paso a paso o *Steppers*.
- Diseño de los modos de funcionamiento.
- Implementación y testeo de los modos de funcionamiento.
- Desarrollo de la *Human Machine Interface* para la Magelis.
- Depuración del funcionamiento conjunto ( Magelis y Autómata).

Se ha diseñado y desarrollado el funcionamiento del autómata así como el sistema de supervisión propio para el control por parte del usuario. Ésto abarca desde la configuración software en Unity Pro de los módulos hardware disponibles, haciendo principal hincapié en el módulo de control de los motores paso a paso. Una de las tareas destacables es la configuración y conocimiento de las funciones de control del módulo de los motores paso a paso, siendo éste determinante para el desarrollo del funcionamiento de la estación.

Se ha llevado a cabo el diseño de la estructura de datos utilizada para la gestión, así como realizar un diseño de las diferentes secciones de forma que pueda ser lo más legible posible de cara a una posible implementación en el proceso conjunto de la estación. Pese a la planificación, la implementación del funcionamiento de un proceso industrial requiere de un diseño minucioso en el que hay que estar a pie de máquina para comprobar y evaluar su correcto funcionamiento

y puesta en marcha, siendo necesario numerosos cambios tanto del funcionamiento en sí, como de la estructura propuesta. Una vez implementado el funcionamiento, se tienen que contemplar todas las posibles necesidades de control de un operario, de tal manera que con las funciones que se diseñen en la magelis, se pueda aprovechar y rectificar el funcionamiento ante cualquier situación posible sin la necesidad de editar la programación, algo que en una situación real requeriría de un técnico de la empresa instaladora en lugar del trabajo de propio operario incurriéndose así en sobrecostos.

Este TFG, como ya se ha mencionado, es un paso hacia la puesta en marcha de la línea de producción de forma global, pero también un paso hacia el uso de la estación para prácticas de las asignaturas de "Sistemas Automáticos", impartidas en todas las ingenierías de ámbito industrial de en la EINA.

## Bibliografía

---

- [1] Ismael Blasco. *Automatización del Almacén de piezas y Monitorización del Almacén y transporte*. 2001.
- [2] Wikipedia. Motor paso a paso.
- [3] Schneider Electric. Premium and atrium using unity pro processors, racks and power supply modules. implementation manual.
- [4] Schneider Electric. Premium y atrium en unity pro. módulos de e/s binarios. manual de instalación.
- [5] Schneider Electric. Premium y atrium con unity pro, módulos de red ethernet. manual de usuario.
- [6] Ramón Piedrafito. *Ingeniería de la Automatización Industrial*. Ra-Ma Editorial, 1999.

# ANEXOS

## Anexo A

# Control avanzado de un almacén de piezas

### A.1. Objetivos

Los objetivos de ésta práctica son:

- Comprender el funcionamiento de un módulo de motores paso a paso.
- Saber utilizar el objeto IODDT T\_STEPPERS de Unity para el control de un motor paso a paso.
- Realizar una puesta en marcha bajo cualquier circunstancia.
- Aplicar el uso a una tarea de almacenaje.

### A.2. Descripción general de la estación

El almacén intermedio, utilizado para el almacenamiento de piezas ya fabricadas procedentes desde la cinta transportadora y su posterior entrega bajo pedido a la salida de la estación, en dirección a la zona de pedidos.

El almacén intermedio consta, como se ve en la figura A.1 de 16 posiciones de almacenaje donde alojar a las piezas hasta su solicitud. Dichas posiciones se encuentran distribuidas en una matriz de 4 filas y 4 columnas.

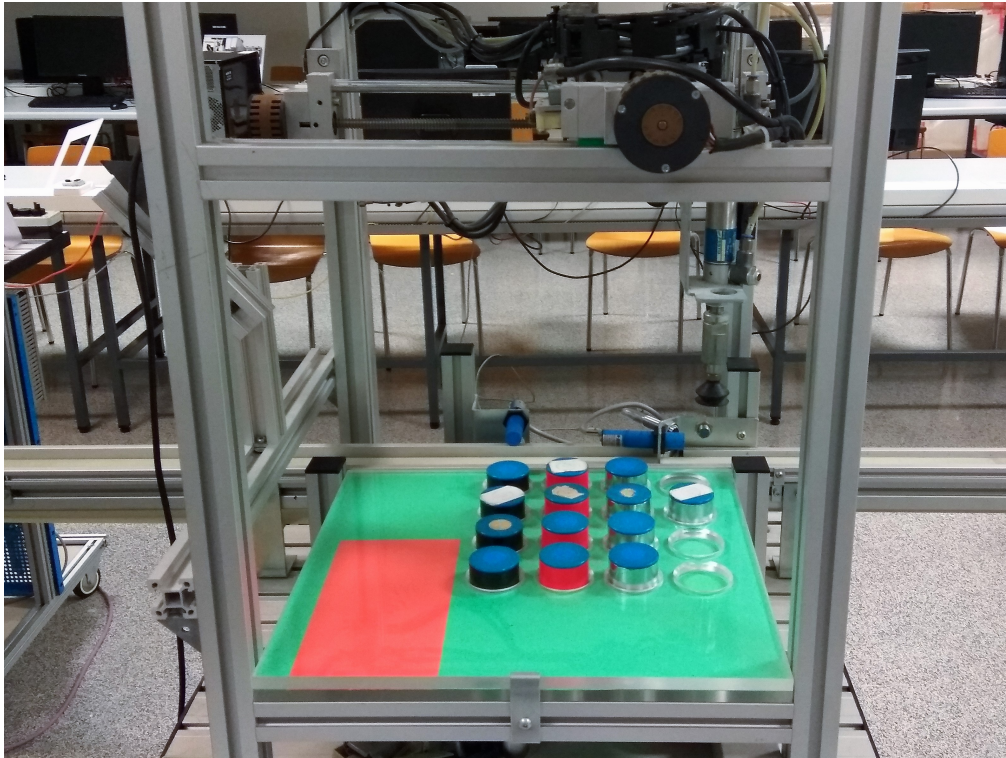


Figura A.1: Vista frontal del almacén.

Para la gestión del movimiento de piezas en el almacén desde la posición de entrada, al almacén y a la salida, se dispone de un brazo manipulador X-Y-Z tipo pórtico, con 3 grados de libertad de los cuales los ejes X e Y son accionados mediante motores paso a paso, y el eje Z mediante un cilindro neumático. Las manipulaciones de las piezas se realizarán por medio de una ventosa de succión. Si bien los ejes X e Y se utilizarán para la selección de piezas, el eje Z será utilizado para realizar los desplazamientos en el almacén de forma segura, manteniéndose arriba durante éstos.

### A.2.1. Actuadores

La estación dispone de los siguientes actuadores

- **Motores paso a paso o Steppers**

Los motores paso a paso o *Steppers* nos permitirán realizar el movimiento del brazo manipulador, llevándolo de una posición a otra. Dada la configuración del módulo de control, *Half Step*, y el husillo de bolas en el eje, el avance lineal en el eje será de 3 mm cada 200 pasos.

- **Cilindro neumático de simple efecto.** Los cilindros neumáticos de simple efecto

disponen de una entrada de control que mantiene la posición mientras esté activada.

- **Generador de vacío.** Mediante una salida del autómata generaremos vacío en la ventosa gracias al efecto venturi.
- **Motor de continua y cinta transportadora.** Mediante un motor de continua realizaremos el movimiento de la cinta en el sentido de entrada hacia salida de forma conjunta ( misma cinta a lo largo de toda la estación.).
- **Tope fin de cinta neumático.** También accionado como un cilindro neumático, que en reposo, bloquea el paso.

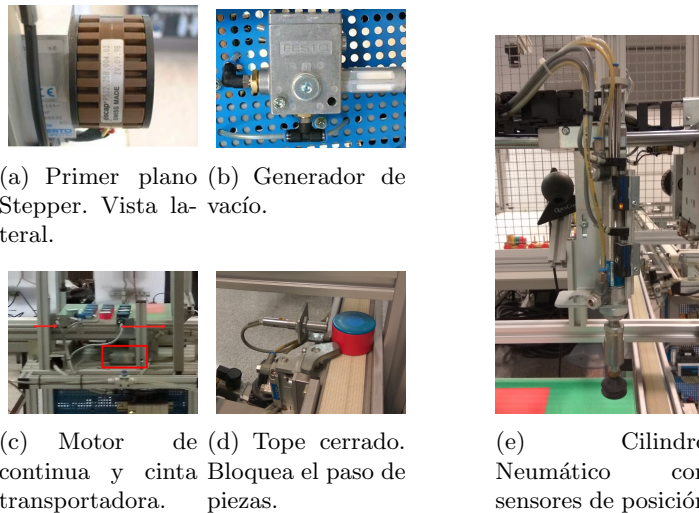


Figura A.2: Principales actuadores en la estación.

### A.2.2. Sensores

Para identificar las distintas etapas del funcionamiento, utilizaremos el uso combinado de sensores dando lugar así a posibles eventos en el funcionamiento, que clasificaremos, según su uso, como la identificación de piezas ( Fig.A.3) o identificación de movimientos ( Fig.A.4).

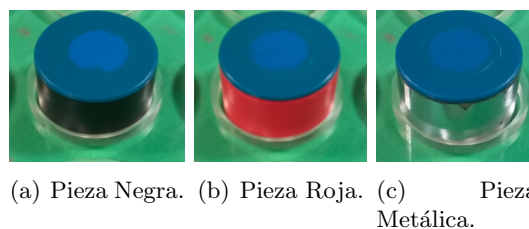


Figura A.3: Posibles tipos de piezas en la estación.

- Como bien indica su nombre, los sensores destinados a la identificadores de pieza, situados a la entrada, se utilizarán para realizar la gestión del almacén y que el sistema pueda memorizar el tipo de pieza que entra.
- Mientras que los sensores encargados de la identificación de movimientos nos servirán de cara a:
  - gestionar el correcto funcionamiento de los actuadores en movimiento.
  - evaluar si procesos muy concretos como la extracción de piezas, se han llevado a cabo <sup>1</sup>



Figura A.4: sensores capacitivo, óptico e inductivo, por orden, a la entrada de la estación.

### A.2.3. Detección del tipo de piezas

Para realizar la tarea de identificación de piezas se utilizará la combinación de 3 sensores distintos: capacitivo, óptico e inductivo (Fig.A.4) situados a la entrada de la estación.

A modo resumen, las diferentes combinaciones posibles de entrada son las siguientes, debiendo ser consideradas el resto como erróneas.

Tipo de pieza	Capacitivo	Óptico	Inductivo
No Pieza	0	0	0
Pieza Negra	1	0	0
Pieza Roja	1	1	0
Pieza Metálica	1	1	1

Cuadro A.1: Tabla donde se recogen las entradas del sistema en función del tipo de pieza

### A.2.4. Variables del sistema

En el autómatas, las entradas y salidas digitales son:

<sup>1</sup>Una vez introducida la pieza en la estación, no hay forma de comprobar el tipo mas que teniendo en cuenta el valor memorizado.



Sensores	Variable	Descripción (a 1)
I.Marcha	%I0.2.1	Botón de marcha.
I.Reset	%I0.2.2	Botón de reset.
I.IND/INT	%I0.2.3	Selector de individual o integrado.
I.EMERG	%I0.2.4	Seta de emergencia.
I.Capacitivo_salida	%I0.2.5	Detecta si ha salido pieza.
I.Capacitivo_entrada	%I0.2.6	Cuadro A.1.
I.Inductivo_entrada	%I0.2.7	CuadroA.1.
I.Optico_entrada	%I0.2.8	CuadroA.1.
I.Brazo_arriba	%I0.2.9	Detecta si el brazo está arriba.
I.Brazo_abajo	%I0.2.10	Detecta si el brazo está abajo.
I.Vacio	%I0.2.11	Detecta si se realiza vacío.
I.Tope_paso	%I0.2.12	Detecta la posición del tope.
I.Optico_salida	%I0.2.13	Detecta si ha llegado pieza al final.

Cuadro A.2: Tabla donde se recogen las entradas del sistema.

Actuadores Binarios	Variable	Descripción (a 1)
O_Brazo abajo	%Q0.3.0	Mantiene el brazo abajo
O_AbrirTope	%Q0.3.1	Abre el tope de paso
O_Vacio	%Q0.3.2	Hace vacío en la ventosa
O_Cinta	%Q0.3.3	Activa el movimiento de la cinta

Cuadro A.3: Tabla donde se recogen las salidas del sistema.

Ésta práctica trata sobre el control y gestión de un módulo paso a paso, por lo que previamente a ello, explicaremos las funciones básicas que incorpora unity para ello.

En el lenguaje de Unity Pro, cada uno de los movimientos se describe en una función de control de movimiento SMOVE. A partir de dicho comando SMOVE y de la posición de la pieza en movimiento, el módulo TSX CFY crea la consigna de posición/velocidad y genera los pulsos del movimiento.

En el programa facilitado para ésta practica podremos ver el objeto IODDT que incorpora Unity Pro para el control de los motores.

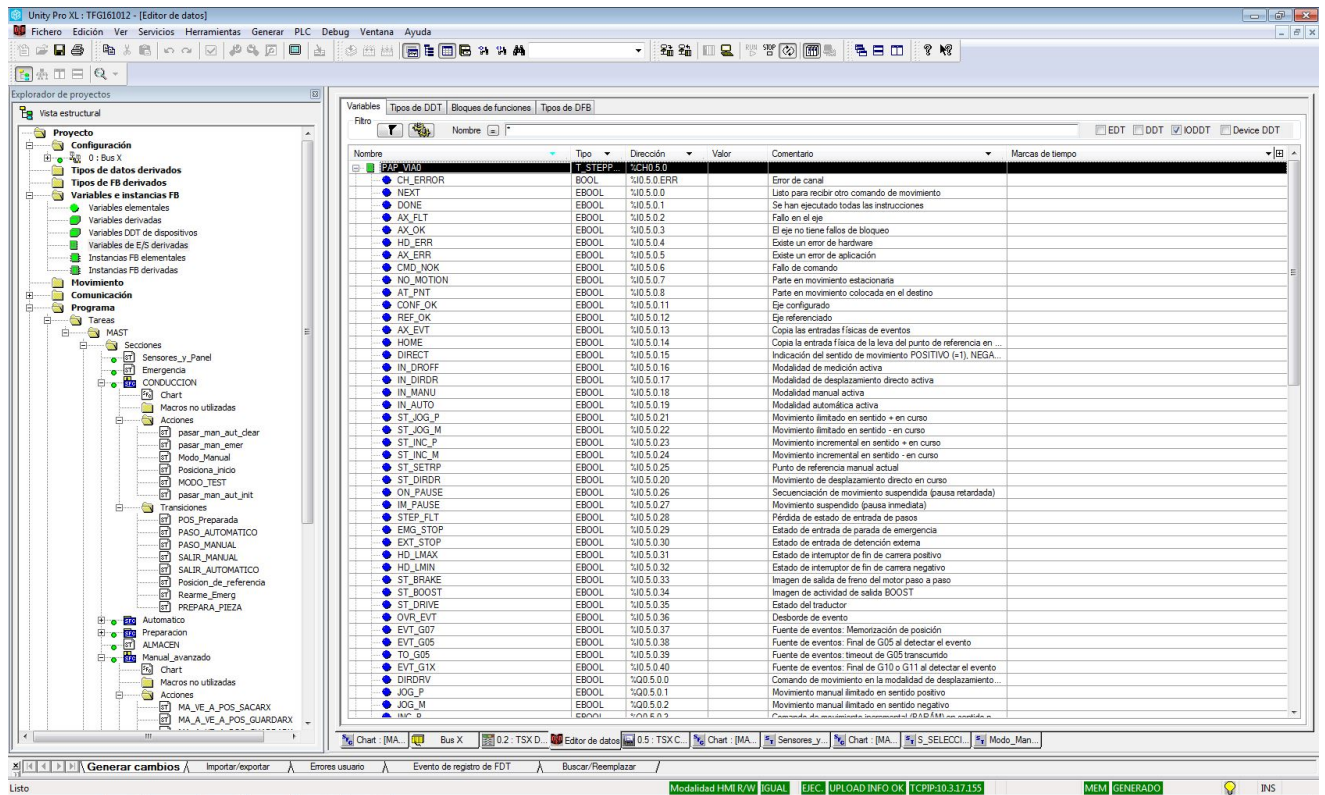


Figura A.5: Lista de variables propias de cada motor paso a paso.

Como se puede apreciar, es larga la lista de variables que nos darán información acerca del funcionamiento y estado del motor. Como una primera aproximación, nos centraremos en las siguientes:

variable	descripción
AX_FLT	Fallo en el eje.
DONE	Se han ejecutado todas las instrucciones
REF_OK	Eje referenciado
AT_POINT	Parte en movimiento colocada en el destino
NEXT	Listo para recibir otro comando de movimiento
MOD.SELECT	Comando de selección de modalidad

Cuadro A.4: Tabla donde se recogen las salidas del sistema.

### A.3. Instrucción SMOVE

El comando SMOVE del módulo sigue la siguiente sintaxis:  $SMOVE(Axis, N_{run}, G9x, G, X, F, M)$  siendo los campos de entrada de a función:

variable	descripción
AXIS	Objeto IODDT del tipo T.STEPPER.STD
N_run	Número de movimiento
G9x	Eje Tipo de movimiento
G	Código de instrucciones
X	Coordenada de posición que debe alcanzarse
F	Velocidad de movimiento
M	Procesamiento de eventos

Cuadro A.5: Tabla donde se recogen las salidas del sistema.

**N\_run** será un entero que identifique el número de movimiento de la función SMOVE. No será de aplicación en ésta práctica.

#### Tipo de movimiento: G9\_:

Código	descripción
90	Movimiento absoluto
91	Movimiento relativo a la posición actual
98	Movimiento relativo a la posición de referencia almacenada *

Cuadro A.6: Tabla donde se recogen los tipos de movimiento.

\*Pudiendo ser ésta diferente a la posición real de referencia. Ver apartado de preparación.

#### Código de instrucciones: G :

Código	descripción
09	Movimiento a la posición con detención
14	Movimiento a la posición de referencia

Cuadro A.7: Tabla donde se recogen los tipos de movimiento.

**M:** M se puede utilizar para el disparo de procesamiento de eventos de la aplicación (instrucciones G10, G11, G05 y G07)

M= 16#1000: activación de la tarea de eventos  
M=16#0000: no activación de la tarea de eventos

En nuestra aplicación, no haremos uso del procesamiento de eventos, por lo que utilizaremos todo el rato dicho parámetro con el valor M=16#0000

### A.4. Ejemplo de aplicación

Antes de empezar con la gestión de los motores paso a paso para usarlos en el almacén, veremos un sencillo ejemplo de aplicación consistente en la inicialización estándar del módulo y el movimiento a una posición. Partiremos de una situación de reposo, en la que en caso de que no haya error en el canal,  $PAP.CH\_ERROR := 0$ , pondremos de forma permanente  $PAP.ENABLE:=1$ .

Para ello, utilizaremos el siguiente graficet de referencia:

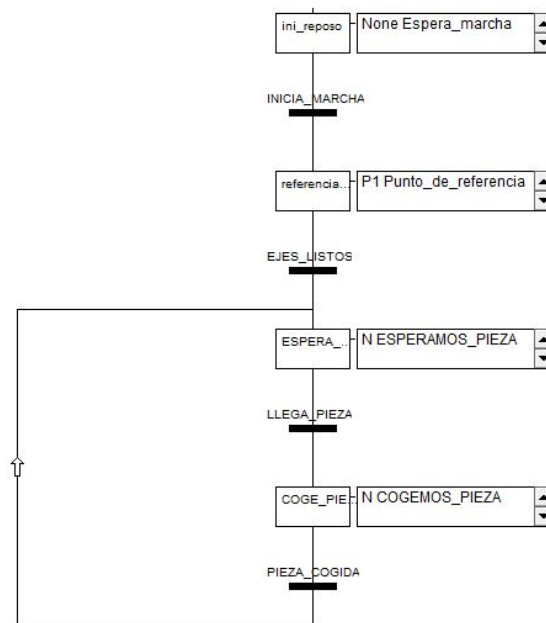


Figura A.6: Ejemplo resumido de funcionamiento.

En la primera transición, a parte de darle a marcha al programa, comprobaremos que no existe fallo en el eje, en cuyo caso deberíamos subsanarlos antes de empezar.

El código de las transiciones y acciones será :

---

**INICIA\_MARCHA:**

NOT PAP\_VIA0.Ax\_ft AND NOT PAP\_VIA1.Ax\_ft AND P\_BOTON\_MARCHA

---

Una vez asegurados de que podemos dar comienzo al movimiento, la acción que se ejecutará sobre cada paso a paso (PAP) para referenciar, serán:

---

**P1 Punto\_de\_referencia:**

*SMOVE(PAP\_VIA0, 0, 90, 14, CorrecciónX, Velocidad\_Preparacion, 16#0000)*  
*SMOVE(PAP\_VIA1, 0, 90, 14, CorrecciónY, Velocidad\_Preparacion, 16#0000)*

---

debiendo ser ésta acción del tipo P1 ( lanzar con flanco de subida, 1 vez), y *CorrecciónX* el parámetro utilizado para corregir una posible desacoplo entre nuestra posición de referencia para las piezas y la posición real del electroimán por un posible desplazamiento de éste.

A continuación, comprobaremos que efectivamente se ha terminado el movimiento y se han referenciado los ejes.

---

**EJES\_LISTOS:**

PAP\_VIA0.DONE AND PAP\_VIA1.DONE AND PAP\_VIA0.REF\_OK AND PAP\_VIA1.REF\_OK

---

Una vez referenciados los ejes, podemos dar comienzo al movimiento. La acción que se ejecutará sobre cada paso a paso (PAP) para realizar un movimiento, será:

---

**P1 Mueve\_a\_posicion:**

*SMOVE(PAP\_VIA0, 1<sup>2</sup>, 98, 09, PosicionX, Velocidad\_Funcionamiento, 16#0000)*  
*SMOVE(PAP\_VIA1, 1, 98, 09, PosicionY, Velocidad\_Funcionamiento, 16#0000)*

---

Debiendo ser también ésta acción del tipo P1. Nótese que aquí el movimiento ya lo definimos como relativo a la posición de referencia (G98) y con detención (09), de modo que no tendremos que calcular nosotros los desplazamientos, ya que lo hará el autómata.

En la siguiente transición, comprobaremos que el brazo realmente ha llegado a la posición solicitada y que está listo para un nuevo pedido.

---

**PIEZA\_COGIDA:**

PAP\_VIA0.AT\_POINT AND PAP\_VIA1.AT\_POINT AND PAP\_VIA0.NEXT AND PAP\_VIA1.NEXT

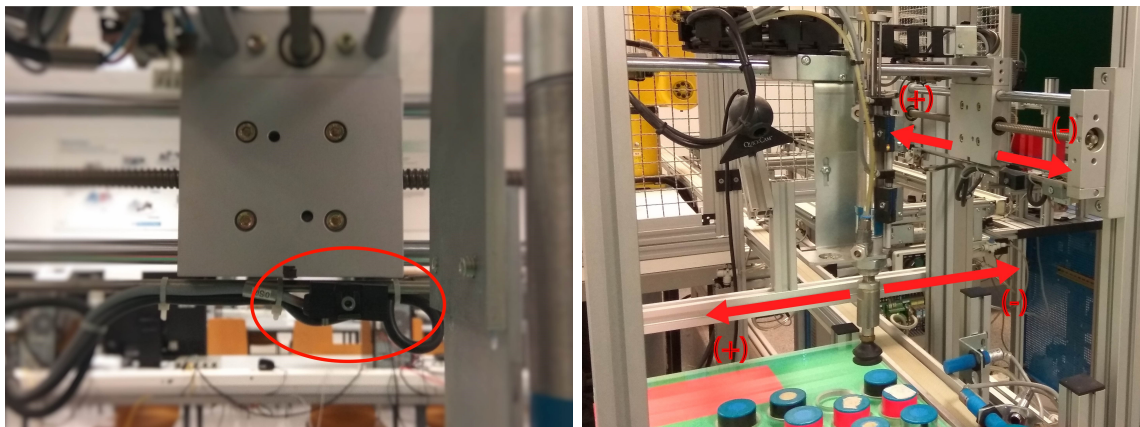
---

---

<sup>2</sup>Valor utilizado para identificación de movimientos, puede mantenerse el 0.

## A.5. Preparación

El funcionamiento de los motores paso a paso, al no disponer de ningún sistema de re-alimentación, requiere de un punto de referencia para inicializar el módulo. Éste punto de referencia viene de un electroimán situado en el eje de movimiento asociado ( Fig.A.7(a)).



(a) Detector inductivo situado en la posición Home.

(b) Sentidos de avance de los ejes.

Figura A.7: Elementos control preparación.

La instrucción SMOVE, utilizando el código de instrucciones **14**, lo que realmente ejecuta es un movimiento continuado en el sentido negativo (Fig.A.7(b)) hasta encontrar la posición de referencia *.HOME*, por lo que en caso de encontrarnos previamente en una posición negativa<sup>3</sup> del autómeta, tendremos que de algún modo desplazar el autómeta en el sentido positivo hasta sobrepasar la posición de referencia o detectarla.

Otras 4 de las variables útiles de las que dispone el IODDT para el movimiento de los motores son *.INC\_P* y *.INC\_M* para realizar un movimiento incremental(Parametro) en el sentido positivo o negativo respectivamente del parametro dado o *.JOG\_P* y *.JOG\_M* para un movimiento **MANUAL**<sup>4</sup> en el sentido indicado mientras dicha instrucción se mantenga activa.

Para la primera parte de la práctica rellenaremos el graficet de preparación usando la instrucción SMOVE en su modalidad *14* : *Movimiento a la posición de referencia*.

Cabe mencionar que el parametro *X*, *posición a alcanzarse*, que le pasemos en ésta situación lo que hará será indicar el valor que debe almacenar el autómeta para dicha posición a la que ha llegado gracias al electroimán.

<sup>3</sup>Tomando como 0 la posición *Home*.

<sup>4</sup>atentos al valor de *.MOD\_SELECT*, **3: Automatico** y **2: Manual**

## A.6. Almacenamiento.

Una vez realizada la preparación del brazo, comenzaremos a usar la estación para la tarea de almacenaje. Para ello, disponemos de una terminal magelis desde la que indicaremos dónde guardaremos la pieza situada en la entrada de la estación. A su vez, ésta terminal nos permitirá llevar el recuento de las piezas en el almacén. Dicha terminal utiliza dos variables que deberemos utilizar en nuestro código: **POS\_ORIGEN** y **POS\_FINAL**.

Cada vez que toquemos en la pantalla de la terminal una posición:

- 1º estableceremos ésta como posición de origen para nuestro movimiento.
- 2º estableceremos dicha posición como posición final donde depositar la pieza.

**TENED CUIDADO DE QUE DICHAS POSICIONES NO SE ENCUENTREN OCUPADAS.**

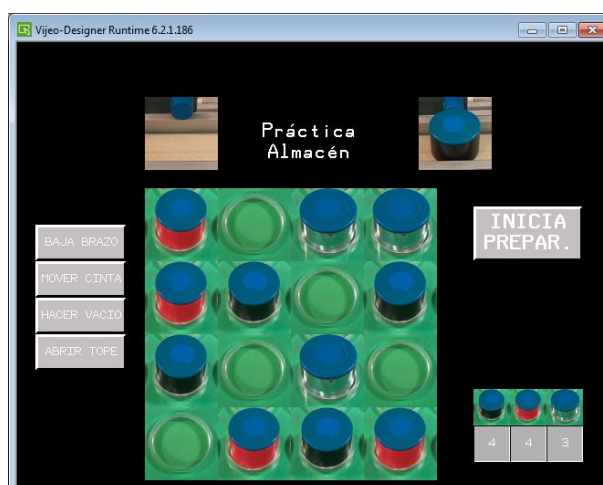


Figura A.8: Pantalla de la terminal Magelis a utilizar en la práctica

Para llevar a cabo ésta tarea será necesario desarrollar el funcionamiento en el autómata mediante una sección que use **POS\_ORIGEN** y **POS\_FINAL** como coordenadas de movimiento en la instrucción SMOVE y escriba en la variable **PIEZA\_TRANSPORTANDO** el valor de la lectura de los sensores a la entrada.

En éste vídeo de **funcionamiento de ejemplo** podemos ver en marcha el comportamiento deseado de la estación, combinando los distintos modos.