



**UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA**



**ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA**

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DISEÑO DE UN SISTEMA PARA LA AUTOMATIZACIÓN DE PROCESOS DE RECOGIDA Y DEPÓSITO DE OBJETOS: APLICACIÓN A UNA PALETIZADORA

AUTOR: GUILLERMO PIQUER RODRIGO

TUTOR: JUAN PÉREZ CRUZ

COTUTOR: MANUEL PINEDA SÁNCHEZ

Curso Académico: 2016-17



RESUMEN

En el proyecto “Diseño de un sistema para la automatización de procesos de recogida y depósito de objetos: aplicación a una paletizadora” se ha diseñado un sistema para automatizar un proceso de recogida y deposición de objetos en un almacén inteligente, en este caso concreto se trata de una paletizadora.

El sistema se ha basado en la tecnología de los autómatas programables (PLCs), que ofrecen una programación muy robusta, necesaria para este tipo de procesos debido a la eficacia y seguridad que requieren. Actualmente los PLCs se encuentran en numerosas industrias, siendo estos una pieza clave en sus procesos.

Primeramente, se ha elaborado un estudio de los diversos PLCs disponibles, analizando su CPU y el número de entradas y salidas de las que disponen para ver cual se adapta más a las necesidades del proyecto. Posteriormente se ha realizado un programa sencillo en cada uno de ellos para ver cuál será el elegido finalmente.

El proyecto se ha diseñado para un almacén inteligente que ha sido simulado en una máquina en concreto, pero es adaptable, realizando ciertas modificaciones, a cualquier otro tipo de máquina que tenga como función el almacenamiento y recuperación.

El sistema dispone un modo de funcionamiento manual y otro automático. En el caso del modo manual la máquina es controlada totalmente por el operario mientras que en el modo automático tan solo debe pulsar el botón de marcha y paro cuando se desee, además cuenta con una seta de emergencia cuya función es parar de inmediato la máquina en el momento que se pulse, para evitar cualquier situación peligrosa tanto para la máquina como para las personas.

El control de la máquina se realiza mediante una pantalla conectada al autómata que hace la función de panel de control.

Siempre que se inicie la máquina, esta debe realizar la llamada puesta a cero, que la situará siempre en el mismo punto inicial tomándolo como punto de referencia.

Palabras Clave: autómata, programable, PLC, programación, proceso, programa, almacén, control.



ÍNDICE DE CONTENIDO

MEMORIA	4
1. OBJETIVOS	
2. PRESENTACIÓN DEL PROYECTO	
3. ANTECEDENTES	
4. JUSTIFICACIÓN	
5. MOTIVACIÓN	
6. ESTUDIO DE LOS AUTÓMATAS	
7. PROGRAMACIÓN	
8. PANEL DE CONTROL	
9. CONCLUSIÓN	
PRESUPUESTO	61
1. INTRODUCCIÓN	
2. COSTES MATERIALES	
3. COSTES MANO DE OBRA	
4. COSTES GENERALES	
5. MARGEN DE BENEFICIO	
6. COSTE FINAL DEL PROYECTO	
PLIEGO DE CONDICIONES	69
1. INTRODUCCIÓN	
2. CONDICIONES DE ÍNDOLE TÉCNICA	
3. CONDICIONES DE ÍNDOLE FACULTATIVA	
4. CONDICIONES DE ÍNDOLE ECONÓMICA	
5. CONDICIONES DE ÍNDOLE LEGAL	
ANEXO I. PROGRAMACIÓN	77
ANEXO II. COMPONENTES	88



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

*Diseño de un sistema para la automatización
de procesos de recogida y depósito
de objetos: aplicación a una paletizadora*



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

MEMORIA



ÍNDICE DE LA MEMORIA

1. OBJETIVOS	7
2. PRESENTACIÓN DEL PROYECTO	8
3. ANTECEDENTES	11
4. JUSTIFICACIÓN.....	12
5. MOTIVACIÓN.....	13
6. ESTUDIO DE LOS AUTÓMATAS	14
6.1. Autómata Schneider.....	15
6.2. Autómata ABB	19
6.3. Autómata B&R.....	21
6.4. Elección del Autómata.....	24
7. PROGRAMACIÓN	26
7.1. Entradas y Salidas.....	27
7.2. Rango de Movimiento	28
7.3. Control de Posición	29
7.3.1. Posición del Eje X.....	29
7.3.2. Posición del Eje Y.....	30
7.3.3. Posición del Eje Z.....	31
7.4. Variables Internas.....	33
7.5. Modos de Funcionamiento	36
7.6. Explicación de los Programas	38
7.6.1. Programa del Modo Automático.....	39
7.6.2. Programa de Posición Z.....	41
7.6.3. Programación en Texto Estructurado.....	43
7.6.4. Programa de Zona de Seguridad	47
7.7. Seguridad.....	48
8. CONTROL DESDE PANTALLA.....	49
8.1. Conexión.....	50
8.2. Transferencia de Variables	51



8.3.	Pantallas Diseñadas.....	52
8.3.1.	Pantalla Principal.....	53
8.3.2.	Pantalla Selección Modo Manual.....	54
8.3.3.	Pantalla de Control desde Máquina.....	55
8.3.4.	Pantalla Control desde Panel.....	56
8.3.5.	Pantalla de Modo Automático.....	58
9.	CONCLUSIONES.....	60



1. OBJETIVOS

El principal objetivo del presente proyecto es la automatización de un sistema que será capaz de realizar un proceso basado en el almacenamiento y recuperación de objetos (Automated Storage and Retrieval System, AS/RS). Este proceso se realizará utilizando la tecnología de los Programmable Logic Controller (en adelante PLC), que proporcionan un funcionamiento rápido y seguro gracias a su robusto sistema operativo.

Entre los objetivos secundarios, se encuentra la búsqueda del autómatas que más se ajuste a las características de este proyecto, para ello se estudiarán tres autómatas disponibles en el Departamento de Ingeniería Eléctrica (en adelante DIE). Los autómatas que serán objeto de estudio serán de los fabricantes Schneider, ABB y B&R.

En cada autómatas se realizará un programa sencillo de arranque y parada de la máquina y se tendrán en cuenta las entradas y salidas disponibles para ver qué autómatas será el elegido para controlar la máquina de tres ejes.

Los objetivos del proyecto están dirigidos a la recogida y depósito de objetos en un almacén, y se simularán en la máquina de tres ejes disponible en el laboratorio del DIE. Se trata de una paletizadora en este caso, pero podrá ser adaptado a otras máquinas similares.

Se controlará el proceso mediante panel de control con una pantalla acoplada al autómatas, para supervisar y controlar el proceso. Este panel tendrá la capacidad de enviar órdenes al autómatas.

El proceso de almacenamiento y recuperación recogerá objetos de una zona establecida que será llamada zona de recogida y los llevará hasta su lugar correspondiente, que será elegido por el operario en cada caso o serán llevados manualmente por él, según el modo que este seleccionado en ese momento.

El operario podrá seleccionar en la pantalla dos modos distintos de funcionamiento de la máquina que serán los siguientes:

- Manual: El operario seleccionará en cada momento donde desea mover los ejes, la máquina se limitará a realizar las tareas indicadas.
- Automático: En este caso el operario seleccionara una casilla de depósito, a partir de este momento la máquina realizará una serie de tareas automáticamente hasta terminar el proceso o hasta que el operario quiera detenerla.

El objetivo de estos modos de funcionamiento es obtener un sistema automatizado, con opción de manejo manual para casos concretos en los cuales no se quiera realizar un proceso continuo, por ejemplo, para mantenimiento o cualquier otra tarea que requiera un manejo personalizado por parte del operario.



2. PRESENTACIÓN DEL PROYECTO

En el proyecto “Diseño de un sistema para la automatización de procesos de recogida y depósito de objetos: aplicación a una paletizadora” se diseña un sistema para automatizar un proceso de recogida y deposición de objetos, se trata de modernizar los procesos actuales de almacenamiento.

Esta mejora se consigue con el sistema AS/RS que optimiza el espacio, agiliza los procesos, mejora la precisión ya que reduce el error humano y garantiza la seguridad de los operarios dentro de la industria. Este proceso está presente en multitud de industrias y comercios en un grado más o menos amplio, ya que es un sistema adaptable a las necesidades del lugar y del proceso. Puede llevarse a cabo para grandes productos y pesados en una gran nave industrial o pequeños objetos en comercios.

Para simular el proceso se utiliza una máquina de tres ejes, situada en el laboratorio del DIE. Se realiza una programación específica para las características de esta máquina, pero es muy similar a la de cualquier otra máquina con el mismo fin, pudiéndose adaptar la programación realizando unos pocos cambios. Toda la información técnica acerca de la máquina se encuentra en el Anexo II de este proyecto.

Para llevar a cabo la automatización se hace uso de los autómatas programables. La programación se realiza desde el ordenador con el software que requiere el autómata a programado. Posteriormente se descarga el programa en el PLC mediante un cable de Ethernet. Una vez descargado, toda la información de la máquina es recibida y procesada por el PLC, para enviar nuevas órdenes en respuesta a las entradas.

Para la elección del PLC más adecuado para este proyecto, se estudian las características de tres autómatas distintos disponibles en el laboratorio del DIE. Es necesario conocer las entradas y salidas de las que dispone cada PLC para ver si cuenta con los recursos necesarios del presente proyecto. En el caso de tener entradas o salidas insuficientes para recoger toda la información de la máquina, se estudia una ampliación de los módulos para ver si por otras características es factible, tales como CPU o software de programación.

Los autómatas a seleccionar son de los fabricantes Schneider, ABB y B&R, con los modelos de CPU TM258LF66DT4LS0, PM583-ETH y X20CP1483-1 respectivamente. Se hace una prueba en cada PLC con una programación sencilla para comprobar que funcionan correctamente y así conocer el software de programación de cada autómata para tenerlo también cuenta en la elección.

Se realiza una pantalla de operador para el control remoto de la máquina y visualización a tiempo real. En el panel se puede ver cuál es el proceso que se está realizando y controlarlo. Este panel también sirve para seleccionar los modos de funcionamiento, apagar el proceso o controlarlo manualmente. Mientras está en funcionamiento, el PLC está en todo momento comunicándose con la máquina y el panel. De esta forma se puede controlar el proceso sin estar en contacto con la máquina.

El panel es una pantalla de Schneider acoplada al autómata mediante un cable de Ethernet que está comunicándose de manera continua con el autómata.

Se programan dos modos de funcionamiento, el manual y el automático cuya finalidad es automatizar el proceso al mismo tiempo que permitir el control humano sobre los movimientos. El modo manual tiene dos opciones posibles, desde la máquina o desde el panel de control. En el automático se elige la celda de depósito y la máquina automáticamente recoge la pieza de la zona de recogida y la lleva al sitio escogido.

De esta forma, y a modo de resumen, los modos son los siguientes:

- Modo manual: la máquina se desplaza únicamente a petición del operario, esto puede servir para tareas de mantenimiento o cualquier uso concreto que se le quiera dar en algún momento determinado.
 - Desde el panel: el operario controla los motores desde este, dispone de un botón para cada movimiento del eje y sentido del motor, también puede bajar y subir la velocidad con dos botones puestos para dicho fin.
 - Desde la máquina: este método es similar al anterior con la diferencia de que se utilizan los interruptores y pulsadores que hay en la propia máquina.
- Modo automático: el operario seleccionará el modo automático y posteriormente el panel mostrará las zonas de depósito disponibles. Se deberá pulsar una y automáticamente la máquina se dirigirá a la zona de recogida para cargar la pieza y seguidamente la llevará a la celda de depósito escogida. En caso de estar la celda ya ocupada por un objeto la máquina no efectuara ningún movimiento.

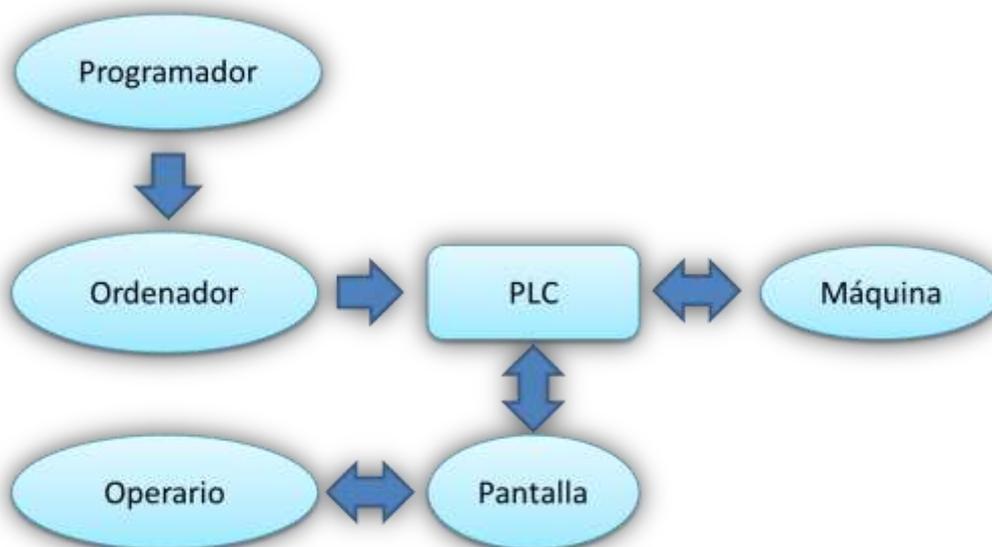


Figura 1. Relación entre equipos

El modo principal de funcionamiento sería el automático, ya que en él reside la importancia y la utilidad de los autómatas programables. Además, es la finalidad de los sistemas AS/RS en las industrias. Pero también se ha visto necesario implementar el modo de funcionamiento manual ya que por cualquier problema que pueda surgir, u operaciones específicas que se necesiten realizar, se puede requerir de este modo.

En la Figura 2 se puede ver un esquema de las posibles opciones que aparecen en pantalla para ser seleccionadas por el operario.

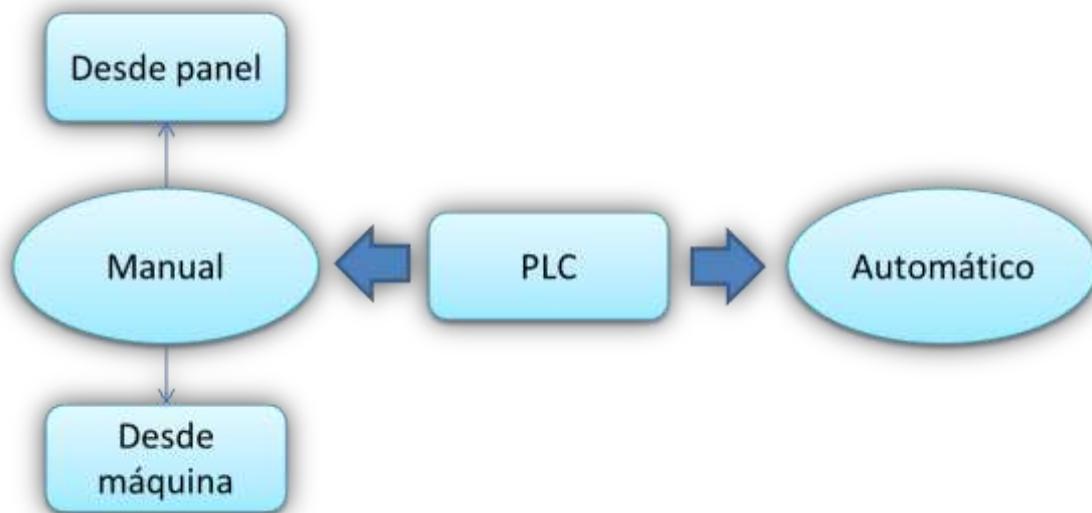


Figura 2. Esquema de los modos de funcionamiento

Todos estos modos son independientes entre sí, pudiéndose eliminar cualquiera de ellos o añadir alguno más en caso de que se quieran adaptar otros procesos. Solamente tendría que añadirse el botón correspondiente en pantalla e implementar la programación que requiera el nuevo modo. De esta forma se tiene un proceso fácilmente ampliable y adaptable a múltiples máquinas cuya función sea la misma. Para la adaptación a otra máquina se deberán realizar algunas modificaciones en función de las capacidades y recursos del nuevo mecanismo.

3. ANTECEDENTES

Desde los inicios el ser humano ha desarrollado herramientas y métodos que le facilitarán cualquier tipo de tarea. Con el avance de los tiempos y la aparición de cada vez un mayor número de máquinas y más avanzadas, el hombre las ha utilizado para realizar las tareas más peligrosas y repetitivas desarrollando el campo de la automatización para realizar el control eficiente de las mismas.

El origen de la automatización se remonta en la revolución industrial que provocó el paso de una economía rural basada principalmente en la agricultura y el comercio a una economía de carácter urbano, industrializada y mecanizada.

El desarrollo de la mecanización ha tenido un papel muy importante ya que permitió el desarrollo de la automatización. La simplificación del trabajo, dividiéndolo en pequeños procesos posibilitó la creación de máquinas que reproducían los movimientos del trabajador.

A medida que la tecnología de transferencia de energía evolucionó, las máquinas se motorizaron, mejorando así de forma notable la producción.

La automatización surgió para mejorar la calidad y uniformidad del producto, minimizar los esfuerzos y tiempos de producción, mejorar la productividad reduciendo costes de manufacturación y aumentando la calidad del producto mediante procesos repetitivos. De esta forma se reduce la posibilidad de error humano. En muchos casos son procesos que pueden generar un riesgo para la seguridad de los operarios, la automatización también vela por la salud de los trabajadores.

Para que esto sea posible se han desarrollado los autómatas programables o PLC, que son el elemento clave de la automatización. El PLC es un equipo electrónico programable que ha sido diseñado para controlar en tiempo real procesos secuenciales. Recibe información a través de los sensores y trabaja esta información en su programa lógico interno actuando sobre el sistema.

Las principales ventajas de los PLC son el menor tiempo de elaboración de proyectos, la posibilidad de añadir modificaciones sin coste añadido, la posibilidad de controlar varias máquinas con un mismo autómata y un menor coste en la mano de obra.

Este campo no deja de avanzar y permite controlar procesos cada vez más complejos, se trata de un mundo muy amplio que está en continuo desarrollo.



Figura 3. Evolución de los procesos de almacenamiento



4. JUSTIFICACIÓN

La justificación del presente proyecto se basa en la importancia y las numerosas ventajas de la automatización en las industrias. En este caso se ha focalizado en la programación de un autómatas para un proceso de almacenamiento y recuperación, presente actualmente no solo en las grandes industrias sino en multitud de lugares cotidianos como comercios.

La automatización ha ido ganando fuerza con los años a medida que se han ido mejorando las tecnologías y los dispositivos utilizados para tal fin. Se ha diseñado con el propósito de llevar a cabo procesos industriales en un entorno de producción eficiente para poder satisfacer la demanda que cada vez requiere tiempos de producción más reducidos.

La mano de obra de los robots hace que dichos procesos transcurran con mayor rapidez y exactitud, con menor margen de error y menores índices de desperdicio. Es por esto que la automatización industrial se ha postulado como un elemento clave en el desarrollo y crecimientos de las empresas.

La implementación de esta tecnología puede suponer una gran inversión inicial, pero esto no debe suponer un problema ya que el retorno de la inversión es seguro y rápido. Además, una vez recuperada la inversión se verán reducidos drásticamente los costes de producción, lo que hace que a larga sea un sistema muy rentable.

Además de las ventajas económicas hay que tener en cuenta la seguridad que conllevan estos procesos ya que la mayoría de veces se trata de operaciones que pueden suponer un peligro para el operario que debe realizarlas, así que un fallo del sistema o por parte del operario podría suponerle lesiones graves o incluso la muerte. En caso de que una máquina realice estas operaciones el único problema que puede surgir es la avería de dicha máquina, cuya solución es simplemente sustituirla.

Los autómatas programables tratan de liberar a las personas de las tareas repetitivas, fatigosas y poco seguras.

En el caso de los sistemas AS/RS cuya función es la de almacenar objetos de forma controlada para su posterior recuperación la importancia reside en el ahorro de espacio de almacenamiento. Este sistema permite distribuir los objetos de una manera que con otros modos de almacenamiento en el cual intervenga la mano humana no podría ser posible, bien por la imposibilidad de alcanzarlos o por la peligrosidad que esto supondría.

Es decir, con la automatización del almacenamiento se maximiza al máximo el aprovechamiento de espacio, la velocidad de almacenamiento, el control de los productos almacenados y la seguridad de las personas encargadas de este proceso.

La justificación de realizar este proyecto de automatización de un sistema AS/RS se basa tanto en la importancia de los autómatas como la de dicho proceso. Un proyecto que abarca una parte del amplio mundo de la automatización y concretamente de los procesos de almacenamiento y recuperación que tienen una gran aplicación en numerosos campos.



5. MOTIVACIÓN

A nivel personal la motivación del proyecto surge del gran desarrollo e importancia que están viviendo los autómatas programables. La realización de este trabajo va a permitir adquirir conocimientos para una programación correcta y adecuada de una máquina para un sistema AS/RS, muy implantado en las industrias y otros campos. Se aprenderá a realizar un diseño del control que resulte sencillo para la utilización de la máquina.

En la elaboración del proyecto, al trabajar con lenguajes de programación diversos, se adquirirá experiencia y agilidad con estos, lo cual permitirá también una rápida adaptación a otros tipos de sistemas que también utilicen estos lenguajes y no solamente al de AS/RS.

Por lo tanto, la finalidad del proyecto no solo es desarrollar la programación de una máquina para un solo sistema, si no también adquirir los máximos conocimientos posibles acerca de los PLCs, ya que son una pieza clave a nivel industrial.

Estos conocimientos dichos anteriormente pueden suponer una gran ventaja de cara al futuro laboral ya que actualmente la mayoría de las industrias cuentan con procesos de automatización en mayor o menor grado. Además, es un mundo que está en continuo desarrollo y va a necesitar siempre de personas con conocimientos acerca de estos autómatas.

Se puede pensar que una vez programada la máquina ya no necesita más acciones por parte del programador, pero esto no es así ya que cualquier fallo que pueda surgir puede ser que requiera una revisión del programa implantado en el autómata, o simplemente en el caso en que se quiera realizar un cambio en la producción se deberá cambiar la programación para que esta se adapte a la nueva necesidad.

Cada vez más el trabajo de los operarios se sustituye por el de las máquinas, pero esto no elimina directamente puestos de trabajo, sino que los traslada de alguna forma, ya que de la implantación de sistemas automatizados surge la necesidad de programar, revisar y reparar esa maquinaria, es decir, el futuro se sitúa en controlar esos robots.

En conclusión, la motivación del proyecto reside en la importancia de los PLCs y el sistema AS/RS citados en el punto "4. JUSTIFICACIÓN" y el interés por el aprendizaje del mundo tan complejo y a la vez tan útil de los autómatas programables.

6. ESTUDIO DE LOS AUTÓMATAS PARA LA ELECCIÓN DEL MÁS IDÓNEO

El primer paso para iniciar este proyecto ha sido la elección del PLC, pieza clave para el desarrollo del trabajo. Todo se centra en la programación del autómata, por ello era necesario escoger el que cumpliera todas las características técnicas como las entradas/salidas y de entre ellos, escoger el que ofreciese un software más cómodo de trabajo.

De entre la amplia gama de autómatas programables disponibles en el mercado, el proyecto se centró en tres de ellos, disponibles ya en el laboratorio del DIE, estos son el M258 Logic Controller modelo TM258LF66DT4LS0 de Schneider, el PM583-ETH de ABB y por último el X20CP1483-1 de la marca B&R.

Se ha realizado un programa sencillo en cada uno de ellos para familiarizarse con el entorno de programación de los autómatas.

Se trata de un programa que mueve los ejes en ambos sentidos controlados desde los interruptores que tiene la máquina. Se ha realizado en texto estructurado debido a la sencillez de la acción. Es similar al modo manual desde máquina “ver pag 36” que se ha diseñado finalmente, pero con algunos detalles menos. Únicamente con el objetivo de conocer el software de cada uno y comprobar su funcionamiento.

Se ha estudiado cuantas Entradas/Salidas tendrá el proceso y se ha determinado que se tratan de 11 entradas digitales, 7 salidas digitales y 2 entradas analógicas.

Las variables utilizadas en la programación son las que se muestran en la Figura 4 donde fwd y rev activan el variador de frecuencia para que active el motor en sentido horario y antihorario respectivamente, cx, cy y cz activan los contactores de los motores de sus ejes correspondientes.

El programa consiste en que cada interruptor activa un eje y un sentido de la marcha, en caso de no haber ningún interruptor activado no debe estar en movimiento motor alguno.

De esta forma la señal x_fwd activaría el contactor correspondiente al eje x y el variador de frecuencia para mover el motor de este eje en sentido horario. De la misma forma para el sentido anti horario, se deberá accionar el interruptor x_rev. Los otros dos ejes siguen exactamente el mismo funcionamiento. En la Figura 5 se puede ver la programación realizada en este programa prueba, realizada en texto estructurado.

```
1  PROGRAM POU
2  VAR
3      x_fwd: BOOL;
4      x_rev: BOOL;
5      cx: BOOL;
6      fwd: BOOL;
7      rev: BOOL;
8      y_fwd: BOOL;
9      y_rev: BOOL;
10     cy: BOOL;
11     z_fwd: BOOL;
12     z_rev: BOOL;
13     cz: BOOL;
14 END_VAR
15
```

Figura 4. Variables Prueba

```
1  IF(x_fwd=1 AND x_rev=0)
2      THEN
3          cx:=1;
4          fwd:=1;
5  END_IF;
6
7  IF(x_fwd=0 AND x_rev=1)
8      THEN
9          cx:=1;
10         rev:=1;
11 END_IF;
12
13 IF(y_fwd=1 AND y_rev=0)
14     THEN
15         cy:=1;
16         fwd:=1;
17 END_IF;
18
19 IF(y_fwd=0 AND y_rev=1)
20     THEN
21         cy:=1;
22         rev:=1;
23 END_IF;
24
25 IF(z_fwd=1 AND z_rev=0)
26     THEN
27         cz:=1;
28         fwd:=1;
29 END_IF;
30
31 IF(z_fwd=0 AND z_rev=1)
32     THEN
33         cz:=1;
34         rev:=1;
35 END_IF;
36
37 IF(x_fwd=0 AND y_fwd=0 AND z_fwd=0)
38     THEN
39         fwd:=0;
40 END_IF
41
42 IF(x_rev=0 AND y_rev=0 AND z_rev=0)
43     THEN
44         rev:=0;
45 END_IF
46
```

Figura 5. Programa de Prueba

A continuación, se va a describir en detalle cada uno de ellos para explicar el proceso de selección que se ha llevado a cabo.

6.1. Autómata Schneider

En primer lugar, se presenta el modelo M258 de Schneider, con la CPU TM258LF66DT4LS0 que cuenta con reloj a tiempo real, puerto Ethernet, Serial Line, USB Programming Port, USB Host y CAN. Consta de 7 módulos entre los cuales tenemos 10 entradas digitales rápidas, 28 normales, 4 salidas digitales rápidas, 24 normales y 4 entradas analógicas. El aspecto del autómata es como se ve en la Figura 6.

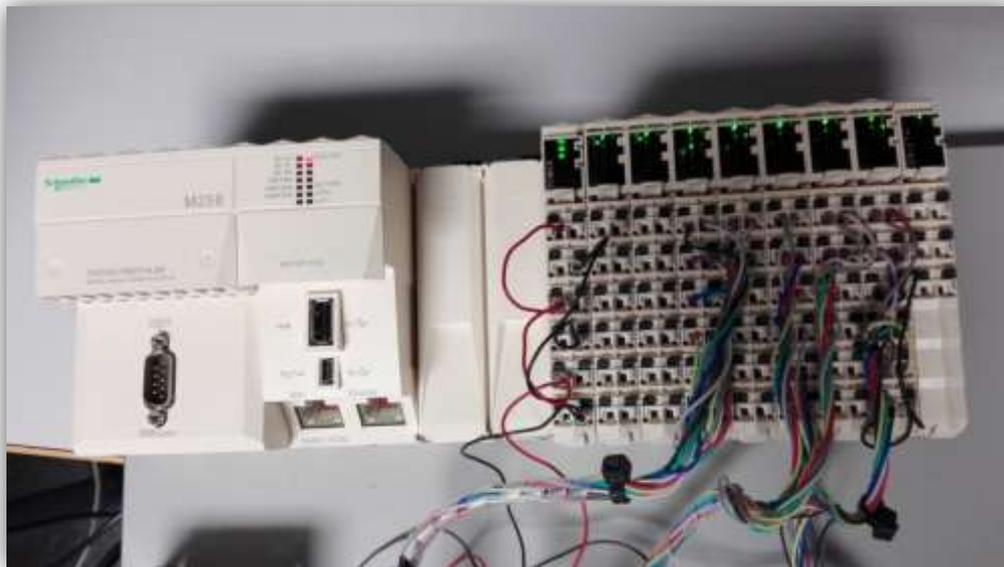


Figura 6. Autómata Schneider

A continuación en la Tabla 1 se resumen las entradas y salidas que tiene cada módulo del autómata.

Tabla 1. Entradas/Salidas PLC Schneider

Módulo	Entradas digitales	Entradas analógicas	Salidas digitales	Salidas analógicas
DM72F0	5 rápidas + 2 normales	0	2 rápidas	0
DM72F1	5 rápidas + 2 normales	0	2 rápidas	0
DI12DE	12	0	0	0
DI12DE	12	0	0	0
DO12TE	0	0	12	0
DO12TE	0	0	12	0
AI4LE	0	4	0	0

En cuanto a la programación, este PLC cuenta con el software de programación SoMachine, un programa disponible en diversos idiomas incluido entre ellos el español, ofrece una interfaz muy sencilla y agradable a la vista. Cuenta con varios tutoriales y guías dentro del programa. Este software requiere licencia para su uso.

Para el inicio de un nuevo proyecto en el SoMachine se debe seleccionar el modelo de autómatas del que se disponga, en este caso el TM258LF66DT4LS0. Con esta selección ya vienen definidos los módulos del PLC por defecto. En caso de ampliación de alguno de ellos o incorporación de nuevos como pantallas o diferentes conexiones se deben añadir en el programa a la configuración del autómatas.

En el caso de este proyecto se añade la pantalla Magelis XBTGT2120 de la cual se hablará más adelante.

La interfaz es como se muestra en la Figura 7.

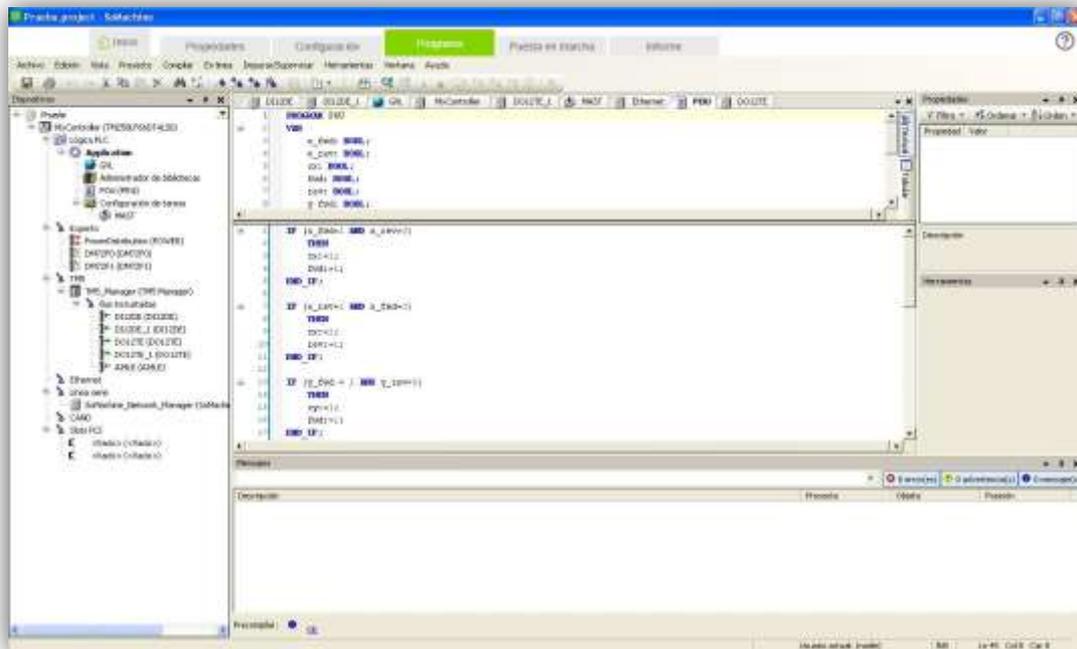


Figura 7. Interfaz SoMachine

Seguidamente se debe abrir una llamada POU (Program Organization Unit) e implementar la programación dentro de este.

Posteriormente se deberá compilar y comprobar que todo está correcto para transferir el programa al PLC. Esto se hace con una Gateway estableciendo como ruta activa el autómatas correspondiente. Se puede ver un ejemplo de conexión al PLC en la Figura 8.



Figura 8. Conexión con el autómeta

El ordenador desde el cual se ejecuta el software debe tener una IP con la cual pueda comunicarse con el autómeta, de lo contrario no se verán entre sí. Esto se realiza poniendo los tres primeros dígitos de la IP iguales.

En este caso concreto en SoMachine deberemos indicar una conexión remota introduciendo la IP del autómeta para poder conectarlo.

En este autómeta en concreto surgieron problemas ya que la versión instalada inicialmente era la versión 3.1.10.1. Esta versión incluye el PLC TM258LF66DT4L pero no el TM258LF66DT4LS0 con lo cual el Gateway no podía comunicarse con él. Solamente una versión anterior contenía en su librería de datos el autómeta TM258LF66DT4LS0.

Después de intentar conectar el autómeta de numerosas maneras con el modelo TM258LF66DT4L seleccionado en la configuración del software se descubrió que tenía que ser exactamente el modelo TM258LF66DT4LS0 el que debía aparecer en la configuración y el software no contaba con dicho modelo.

Para tener el software adecuado se ha utilizado una máquina virtual con Windows XP y el SoMachine versión 3.0.14.5, que si incluye el autómeta objeto de estudio para el presente proyecto. Trabajar con la máquina supone menor agilidad y rapidez en la programación y uso del software debido a que consume mucha memoria RAM del ordenador.

6.2. Autómata ABB

El autómata ABB, tiene una CPU PM 583-ETH con puerto serie, Ethernet, FBP, reloj a tiempo real y ranura para tarjeta SD. Con sus tres módulos dispone de 32 entradas y salidas digitales más 4 entradas y salidas analógicas. En la Figura 9 se puede ver el PLC.

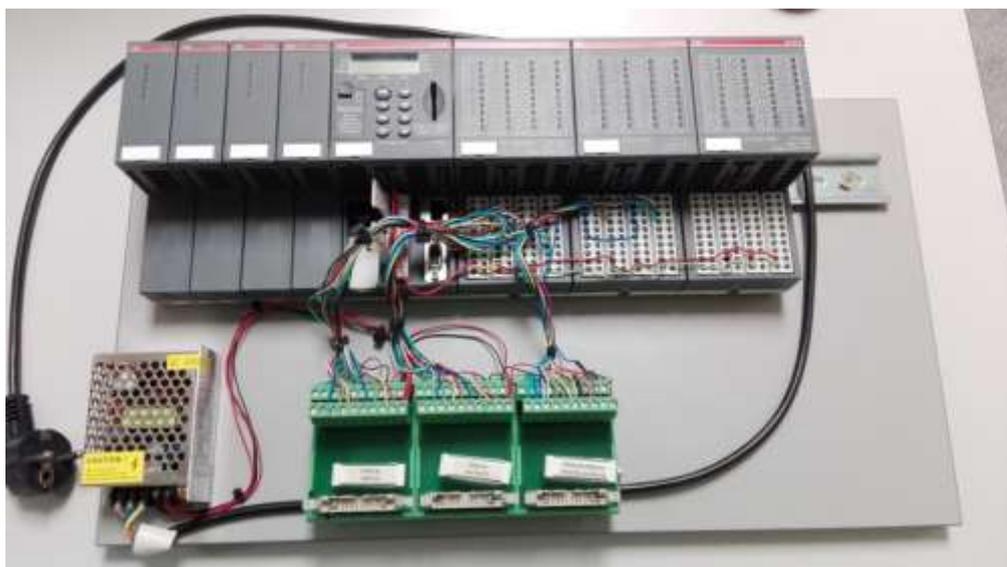


Figura 9. Autómata ABB

A continuación en la Tabla 2 se exponen a modo resumen los módulos de entradas y salidas del autómata.

Tabla 2. Entradas/Salidas PLC ABB

Módulo	Entradas digitales	Entradas analógicas	Salidas digitales	Salidas analógicas
DC532	16	0	16	0
AX521	0	4	0	4
DC532	16	0	16	0

La programación de este autómata se realiza con el software Control Builder Plus, un software gratuito disponible en inglés, francés, alemán y español.

Para iniciar un proyecto se selecciona nuevo proyecto y se escoge la plantilla AC500. Posteriormente el programa pedirá la introducción de un dispositivo, en este caso se selecciona PLC y concretamente el modelo AC500 PM 583-ETH.

Los módulos del tendrán que ser añadidos posteriormente al BUS de entradas y salidas. En este caso se trata de dos módulos DC532 para las entradas y salidas digitales y el módulo AX521 para las analógicas. En la Figura 10 se muestra la interfaz del programa con la CPU ya seleccionada.

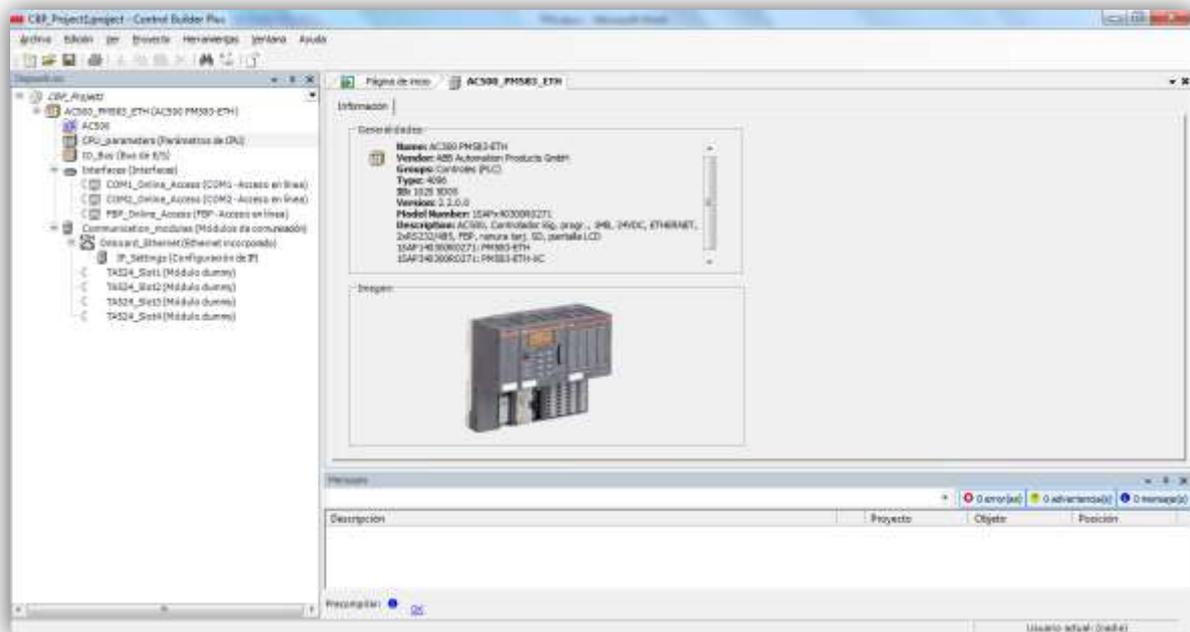


Figura 10. Interfaz software ABB

Seguidamente, con el PLC ya seleccionado, abrimos un POU con el tipo de lenguaje de programación que se desee, al igual que con el autómata de Schneider.

En este caso la programación se debe realizar desde el CoDeSys, donde estarán todos los POU del proceso y desde donde se va a escribir e implementar todo el programa.

Esta aplicación se abre en una ventana distinta y puede parecer otro programa, pero está ligado continuamente al Control Builder Plus.

Para abrir esta ventana CoDeSys se debe abrir el elemento AC500 de la ventana dispositivos en el programa. Desde ahí se compilará toda la programación implementada.

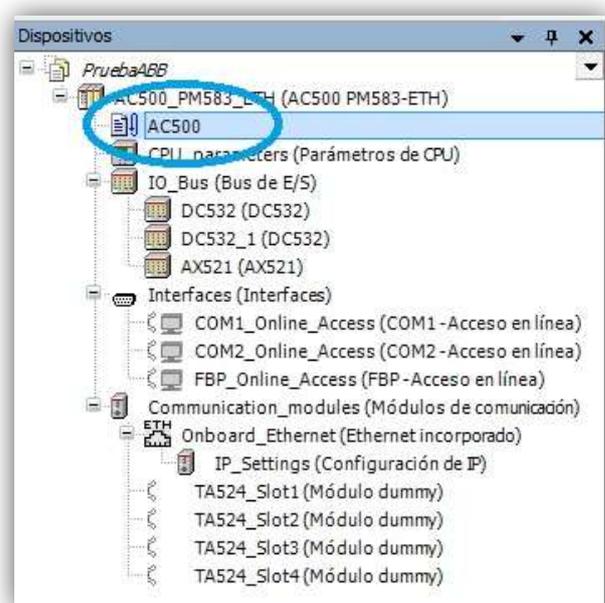


Figura 11. Abrir CoDeSys

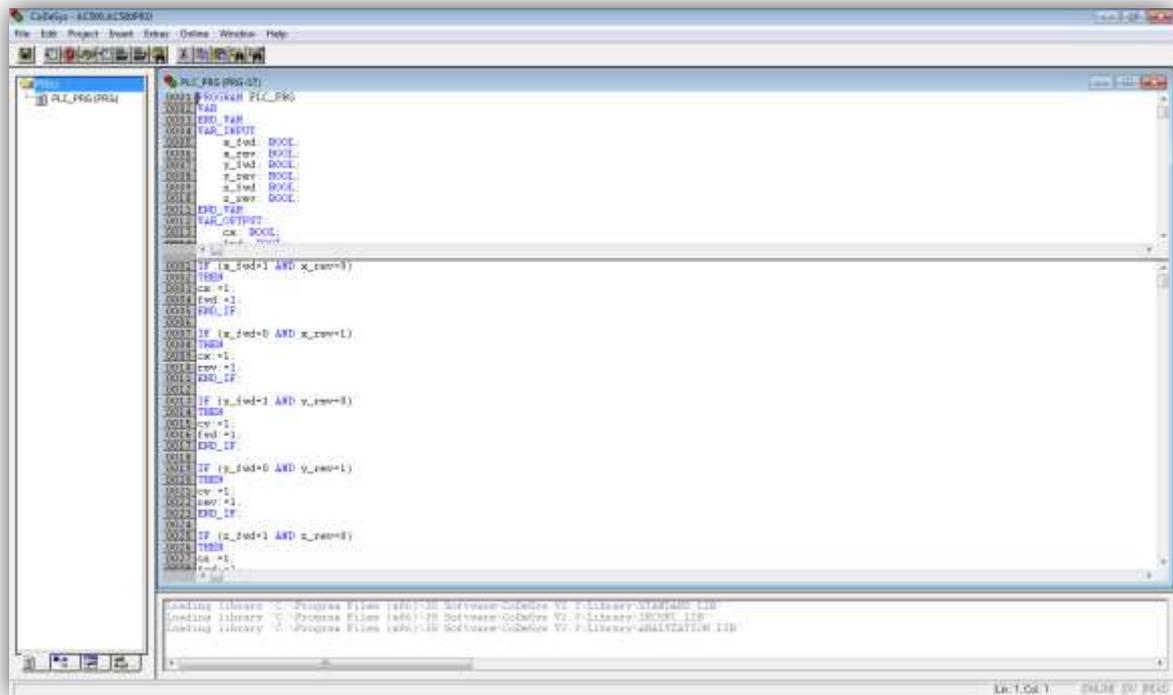


Figura 12. Interfaz de programación CoDeSys

Una vez terminada la POU se transfiere al PLC. Para ello se tiene que ir a la ventana del Control Builder Plus y pulsando con el botón derecho sobre el autómata se selecciona ajustar puerta de enlace. Se configura y después de ello se pulsa, mediante el mismo procedimiento el botón inicio de sesión. Se descargará el programa y se podrá realizar una simulación desde el mismo ordenador.

6.3. Autómata B&R

Por último, el autómata B&R cuenta con una CPU X20CP1483-1 con procesador Inter x86 de 100MHz, dos entradas Ethernet, otras dos USB, tarjeta CompactFlash extraíble y expansión modular de interfaces, además de reloj a tiempo real. Entre los cinco módulos que tiene cuenta con un total de 8 entradas y salidas digitales, 4 entradas analógicas y 2 salidas analógicas.

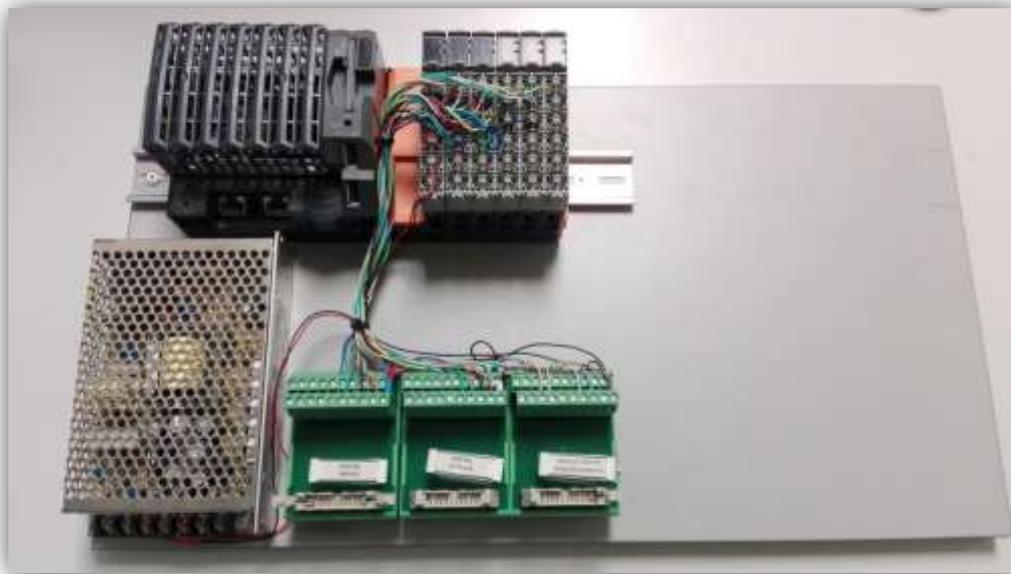


Figura 13. Autómata B&R

Tabla 3. Entradas/Salidas PLC B&R

Módulo	Entradas digitales	Entradas analógicas	Salidas digitales	Salidas analógicas
X20 DI 8371	8	0	0	0
X20 DO 8322	0	0	8	0
X20 AI 2622	0	2	0	0
X20 AO 2622	0	0	0	2
X20 AT 2222	0	2	0	0

Este PLC requiere el software Automation Studio para su programación, se necesita licencia y solo está disponible en inglés y alemán.

En este caso para el estudio previo se ha utilizado la versión de prueba, que es gratuita y tiene un periodo de 30 días.

Para la programación el primer paso es seleccionar un proyecto nuevo. El software solamente pedirá de inicio la CPU del autómata, en este caso la X20CP1483-1, sin ningún módulo predeterminado con lo cual se deberán añadir a posteriori.

Aparecerá un dibujo de la CPU elegida como se muestra a la Figura 14 y tendrán que añadirse los módulos correspondientes arrastrándolos hasta la CPU, en este caso son el X20 DI 8371, X20 DO 8322, X20 AI 2622, X20 AO 2622 y X20 AT 2222. Finalmente el dibujo queda tal y como se muestra en la Figura 15.



Figura 14. CPU sin módulos



Figura 15. CPU con módulos

Para la programación se deberá añadir un programa del tipo que se requiera seleccionando añadir objeto y posteriormente el programa como se ve en el ejemplo de la Figura 16.

A diferencia del resto, en este software existe una pestaña para el programa y otra diferente para declarar las variables.

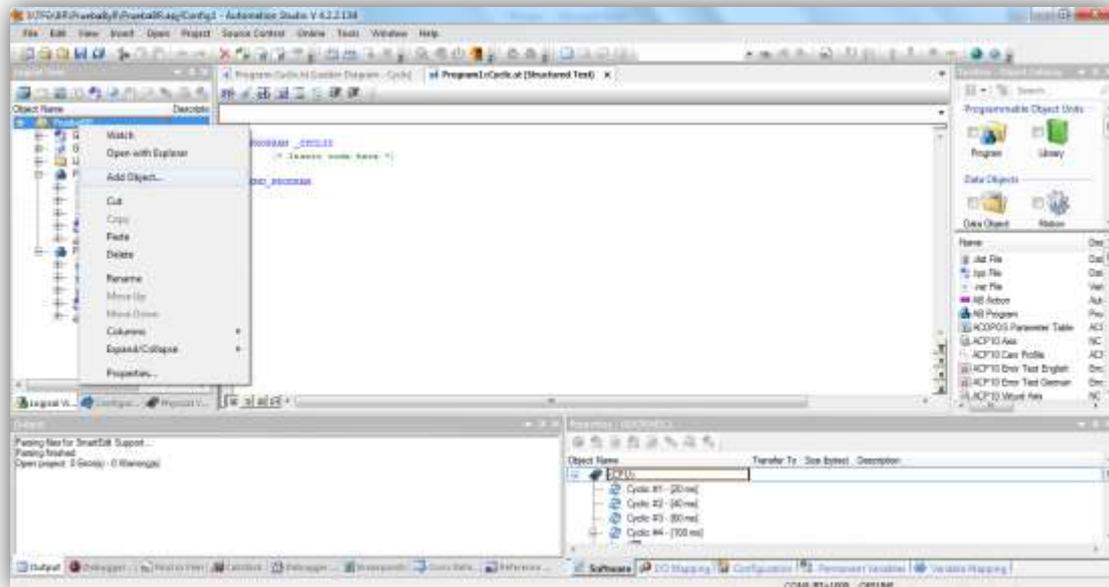


Figura 16. Añadir POU en Automation Studio

Para la transferencia al autómatas primero se debe comprobar la IP del dispositivo y configurarla desde el programa. Al igual que en los otros autómatas, para asegurar la comunicación se deberá cambiar la IP del ordenador a una con los mismos tres dígitos para que se vean entre sí.

Una vez hecho esto se compila el programa y se transfiere al PLC mediante una conexión Ethernet entre el ordenador y el autómatas.

6.4. Elección del Autómatas

Este apartado va a describir los criterios seguidos para la elección final del PLC según todas las características expuestas anteriormente.

En primer lugar, el estudio se ha focalizado en las entradas y salidas, ya que es el punto más importante debido a que sin el número suficiente de estas no se podría adaptar correctamente el proceso.

Visto esto, se descarta el autómatas de B&R por no contar con las entradas digitales suficientes. No se han visto características destacables que hagan replantearse la ampliación de los módulos. La programación es similar al resto y no cuenta con ningún accesorio que pueda ser de gran utilidad respecto a los demás.

Además, el software que utiliza ha presentado mayores problemas respecto a los otros dos y presenta una interfaz más complicada. Requiere más tiempo para su aprendizaje que con los otros dos softwares.

En cuanto al idioma es el único de los tres que no está en español, esto no es un inconveniente para el programa, pero sí una ventaja para los otros dos.

Por lo tanto, esto deja la elección en dos autómatas, ambos cumplen con todos los requisitos para la elaboración del proyecto por lo que la elección entre ellos ha sido determinada por la pantalla Magelis XBTGT2110 que puede ser acoplada al autómatas de Schneider, lo cual supone grandes ventajas para el control mediante SCADA.

Este panel de Schneider se encuentra en el laboratorio del DIE por lo que ha contribuido mucho en la elección final del autómatas.

Esta pantalla táctil actúa de panel de control. En ella se ha diseñado la interfaz gráfica para seleccionar los modos de funcionamiento, apagar la máquina, detenerla y controlarla mediante el modo manual desde panel.

Esto proporciona al proyecto mayor comodidad y profesionalidad, ya que asemeja más a un proceso industrial.

Además, también se ha escogido el PLC de Schneider por la interfaz intuitiva de su software, las múltiples guías que ofrece y la comodidad de programación.

Pese haber tenido problemas en la transferencia de información el autómatas Schneider se ha seguido adelante con él ya que una vez solventado el problema no hay más inconvenientes. Con el software adecuado no ha surgido ningún problema más.

El hecho de tener que trabajar con la máquina virtual no supone un grave retraso en la ejecución del proyecto.

El ordenador de trabajo ha soportado el consumo de memoria RAM producida por dicha máquina virtual sin más problemas que leves retrasos en la ejecución de algunas acciones o tiempos de espera mayor en el encendido del ordenador.

Debido a haber tenido que elegir entre varios PLC da la ventaja de que un si hubiese surgido algún problema en el autómatas escogido se hubiera podido cambiar a uno de los otros sin empezar de cero ya que ya se ha tenido contacto previo con ellos.

En este caso no se ha contemplado el precio de los autómatas por estar ya disponibles en el laboratorio. En un caso real sería un factor también a tener en cuenta ya que una gran diferencia de precio podría decantar la balanza.

Del mismo modo no se ha planteado comprar una pantalla para el PLC de ABB mientras que en un proceso real de programación se hubiera podido plantear dependiendo de los precios y capacidades de programación.

En conclusión, se puede decir que la elección del PLC para la realización del presente proyecto, después de eliminar obviamente los que no cumplieren el número mínimo de entradas y salidas necesarias para el proceso, se ha decantado por la pantalla disponible y utilizable a modo de panel de control mediante SCADA.

A parte de esto se considera que el software ofrece una mayor comodidad en la programación, aunque de lo contrario no hubiese sido un condicionante ya que los otros factores son de mayor peso en esta decisión.

Además, este software incluye herramientas para la programación de la pantalla, sin necesidad de instalar otros programas, lo cual también es un punto a favor.

Una vez escogido el autómatas y conocido el software ya se puede empezar a programar el proceso final con más garantías.

Tabla 4. Resumen de características de los PLC

Fabricante	Schneider	B&R	ABB
E/S suficientes	X		X
Buena interfaz de software	X	X	
Pantalla	X		
Software en español	X		X

7. PROGRAMACIÓN

Para la programación se han utilizado diferentes lenguajes de programación, todos ellos apoyados por la norma IEC 61131, diseñada para autómatas programables. Los lenguajes que contiene esta norma son los siguientes:

- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Structured Text (ST)
- Instruction List (IL)
- Sequential Function Chart (SFC)

Para este proyecto los lenguajes utilizados han sido principalmente el SFC, el ST y en menor medida el FBD.

El lenguaje de diagrama de GRAFCETs (SFC), es un lenguaje gráfico basado en GRAFCET. Puede ser usado en procesos que puedan dividirse en pasos. Los principales componentes de este lenguaje son los pasos asociados a acciones, las transiciones con condiciones lógicas asociadas y los enlaces directos entre pasos y transiciones.

Los pasos en un diagrama SFC puede ser activos o inactivos. Las acciones solo son ejecutadas mediante pasos activos mientras que los pasos inactivos pueden servir para mantener el proceso en espera hasta que se cumpla alguna condición. Un paso será activado cuando todos los pasos anteriores hayan sido activados y se cumpla la condición de su transición.

Las acciones asociadas con pasos son de diversos tipos, las más importantes son las Continuas (N), Set (S) y Reset (R). La acción N asegurada que la variable esté activada mientras dura el paso. Se pueden insertar diagramas Ladder dentro del programa SFC.

En lenguaje de texto estructurado (ST) es un lenguaje de alto nivel que está estructurado en bloques de códigos y sintácticamente se parece a Pascal, en el cual está basado. Consiste en secciones de códigos con declaraciones y sentencias. Es muy cómodo para operaciones sencillas que no requieran procesos ramificados. Entre sus estructuras disponibles se encuentran el IF, ELSE IF, CASE, WHILE, AND, OR, FOR, etc.

Por último, el diagrama de bloques funcionales (FBD) es un lenguaje gráfico que puede describir las funciones entre variables de entrada y variables de salida. La función es descrita como un conjunto de bloques elementales. Entradas y salidas son conectadas con líneas conectoras o enlaces. Una línea simple puede ser usada para conectar dos puntos lógicos del diagrama.

La conexión está orientada, siguiendo las líneas de izquierda a derecha. Las conexiones a la izquierda y a la derecha deben ser del mismo tipo. Este tipo de programación se asemeja a un circuito eléctrico y es útil para el uso de contadores, temporizadores y generadores de pulsos entre otros.

7.1. Entradas y Salidas

En todo momento la máquina está enviando y recibiendo información, para ello hace uso de sensores y elementos que conforman las entradas y salidas.

En las siguientes tablas se indican las variables de entrada y salida así como la dirección que tienen asociadas en el autómata.

Tabla 5. Entradas Digitales

ACCIONADOR	VARIABLE	DIRECCIÓN
Pulsador Speed Up	subir_vel	%IX6.0
Interruptor "X Right"	x_fwd	%IX6.1
Pulsador Speed Down	bajar_vel	%IX6.2
Interruptor "Y Right"	y_fwd	%IX6.3
Interruptor "X Left"	x_rev	%IX6.4
Interruptor "Z Down"	z_rev	%IX6.5
Interruptor "Y Left"	y_rev	%IX6.6
Sensor Inductivo X	zero_x	%IX6.7
Interruptor "Z Up"	z_fwd	%IX7.0
Sensor Inductivo Z	zero_z	%IX7.1
Sensor Inductivo Y	zero_y	%IX7.2

Tabla 6. Salidas Digitales

ACCIONADOR	VARIABLE	DIRECCIÓN
Contactador X	cx	%QX2.0
Contactador Y	cy	%QX2.1
Contactador Z	cz	%QX2.2
Motor Adelante	fwd	%QX2.3
Motor Atrás	rev	%QX2.4
Incremento de velocidad del variador	vel	%QX2.6

Tabla 7. Entradas Analógicas

ACCIONADOR	VARIABLE	DIRECCIÓN
Potenciómetro X	x_anal	%IW8
Potenciómetro Y	y_anal	%IW7

7.2. Rango de Movimiento

La máquina disponible en el laboratorio utilizada para llevar a cabo el presente proyecto se trata de una máquina pequeña y con un rango de movimiento corto. La programación se ha diseñado adaptada específicamente a esta, por ello hay que tener en cuenta cierto detalles y limitaciones a considerar.

Para no tener problemas en cuanto a desacople de los ejes en los tornillos sinfín la máquina cuenta con finales de carrera que cortan la alimentación de los motores al ser accionados. De esta forma el movimiento queda limitado a los finales de carrera. Pero esto es solamente una medida de seguridad que no debe ocurrir durante el proceso.

Además de estos finales de carrera la máquina cuenta con tres sensores inductivos, uno por eje. Estos detectan cuando el eje se ha situado encima y mandan la señal al autómatas. Están situados cerca de los finales de carrera, por lo que una velocidad alta de los motores puede provocar que al llegar a ellos, la inercia del mecanismo provoque la activación de estos elementos de seguridad.

Por lo tanto, se debe crear una zona de seguridad a partir de la cual la máquina solo funciona con velocidad mínima, para evitar el contacto con los finales de carrera y tener una mayor precisión en la maniobra.

De esta forma el rango de movimiento está limitado por los sensores inductivos, respetando la misma distancia respecto al final de carrera en el lado opuesto, en el cual no existe sensor. La zona de seguridad se sitúa a tres centímetros del final del recorrido programado, para asegurar el frenado antes de cada final de carrera.

Para medir la posición se hace uso de los potenciómetros, que ofrecen una señal analógica con la cual se puede saber la posición del eje en cada momento.

En el eje Z no se tiene el recurso del potenciómetro, con lo cual la información sobre la posición del eje debe ser calculada ya que ninguna entrada aporta información sobre él.

Debido a la falta de potenciómetro del eje Z este deberá llevar siempre la velocidad mínima, ya que de otra manera no se podría calcular correctamente la posición y no se llevaría un control adecuado de la máquina.

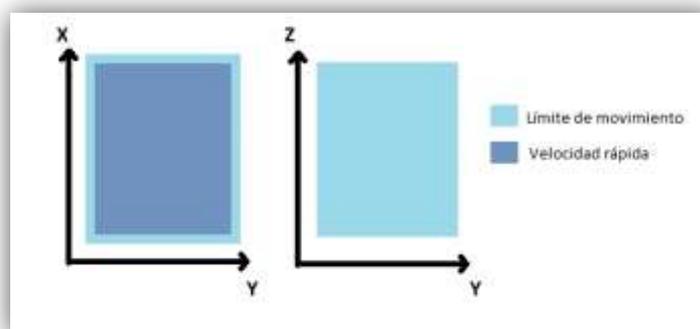


Figura 17. Rango de movimiento en los ejes X e Y (izquierda) y Z (derecha)

7.3. Control de Posición

Como se ha visto en el apartado anterior, para el control de posición se han utilizado los potenciómetros instalados en la máquina. Estos envían una señal analógica en función de la posición del eje. Esta señal debe convertirse para obtener la posición real como se explica a continuación. Para el eje Z, al no disponer de potenciómetro se obtendrá con un método diferente que dependerá del tiempo de funcionamiento de su motor.

7.3.1. Posición del Eje X

Para estudiar la posición de este eje se cuenta con un potenciómetro, como ya se ha dicho anteriormente, este envía una señal analógica que es recibida por el autómatas y debe ser convertida a un valor que se ajuste a las medidas de la máquina. Para ello se ha introducido en el panel un lector de la variable para poder realizar la conversión.

Una vez leída la señal que da el potenciómetro al principio y al final, y habiendo medido la distancia real que han recorrido, se procede a realizar la conversión, sacando la ecuación de la recta.

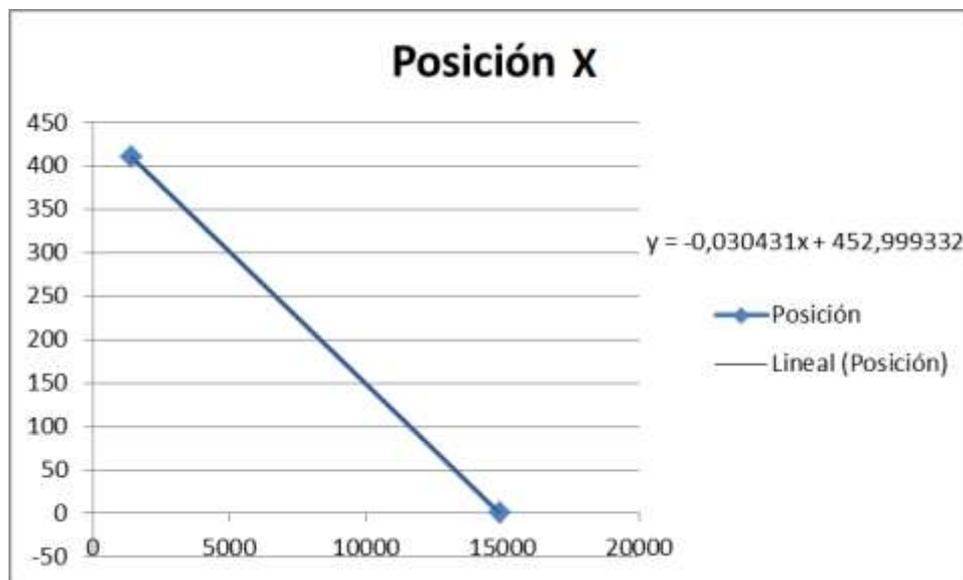


Figura 18. Recta de Posición Eje X

De esta forma, como se ve en el gráfico la ecuación que da como resultado en milímetros, la posición real del eje X:

$$X = -0,030431X_{anal} + 452,999332$$

Ecuación 1. Posición Real Eje X

7.3.2. Posición del Eje Y

Para conocer la posición real del eje Y se ha procedido de manera similar a la del eje X ya que este también cuenta con potenciómetro que envía una señal analógica. La diferencia es que este varía la intensidad de salida, y no la tensión como en el caso anterior. Por ello se ha tenido que configurar debidamente el canal de entrada del autómatas para que lea la intensidad de esta señal.

Al igual que en el apartado anterior, se ha medido la señal al principio y al final y se ha medido el desplazamiento para obtener la ecuación.

Se colocan estos datos en la gráfica y el resultado es el siguiente.

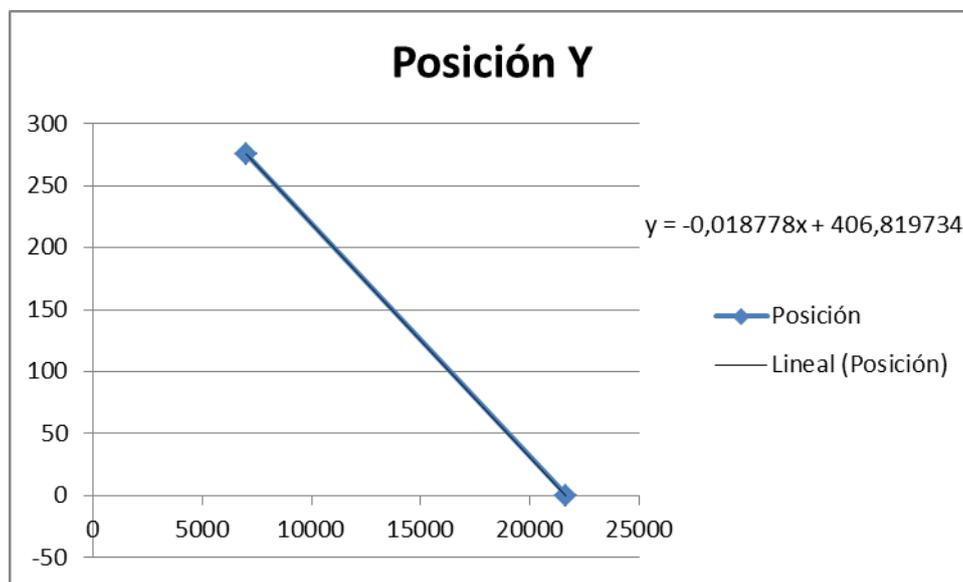


Figura 19. Recta de Posición Eje Y

Del gráfico se recoge la ecuación que indicará la posición real del eje Y en todo momento:

$$Y = -0,01878Y_{anal} + 406,81973$$

Ecuación 2. Posición Real Eje Y

El resultado de la posición real está en milímetros.

7.3.3. Posición del Eje Z

Para el control de la posición del eje Z no se cuenta con potenciómetro por lo que había que buscar alguna alternativa. No era factible en este caso la adquisición de un nuevo sensor así que para solucionar este problema se ha hecho un control del tiempo de marcha del motor del eje Z para calcular de forma aproximada la posición.

Una primera opción fue crear un sencillo programa que simplemente relaciona la posición actual con el tiempo, de forma que se suma a la posición actual una constante multiplicada por el tiempo de marcha de la siguiente forma:

$$pos_z = pos_z + C * t$$

Ecuación 3. Primera ecuación de control

Esta ecuación sería cuando el eje se desplaza hacia abajo y de signo contrario cuando se desplaza hacia arriba. Siendo C una constante que relaciona el tiempo de marcha del eje con la distancia recorrida por el mismo.

Pero esto no es un método eficaz ya que de esta forma el valor solamente se contabiliza una vez haya refrescado la variable, es decir, cuando ha terminado el movimiento. Por esta razón, con este método no se puede llevar a cabo un control instantáneo de la posición real del eje, por lo que se ha buscado otra solución más factible.

La solución ha sido restar tiempos de subida y de bajada utilizando contadores y generadores de pulsos, para ello se ha utilizado el lenguaje FBD y LD que permiten el uso de estos elementos en su programación.

El elemento clave de este proceso de control del eje Z ha sido el contador CTUD, que suma tiempo cuando el eje se desplaza hacia abajo y lo resta cuando se desplaza hacia arriba. Se crea una variable t de tipo real que indica la diferencia de tiempos entre subida y bajada. Si el valor vale 0 quiere decir que el eje se encuentra en la posición 0, es decir, donde se encuentra el sensor inductivo, que es el resetea el contador cuando se activa. Conociendo la variable t se puede calcular la posición de z mediante la siguiente ecuación:

$$pos_z = 4 * t$$

Ecuación 4. Ecuación de control de posición

Un inconveniente de este contador es que no cuenta de forma continua, solamente realiza una cuenta, arriba o abajo si recibe un flanco de subida. Para solucionar este pequeño problema se incluye un generador de pulsos BLINK delante de la entrada de subida y otra igual delante de la entrada de bajada.



Para que el generador de pulsos se active será necesario que las variables del contactor z junto con las variables de movimiento del motor, fwd o rev en cada caso, estén activas. De esta forma si el eje se está movimiento hacia abajo, estarán activas las variables cz y fwd, por lo que estará generando pulsos el BLINK situado en la entrada de subida del contador. En caso contrario se activaría el otro generador de pulsos.

Los módulos BLINK permiten controlar el tiempo de generación de cada pulso. Por lo que puede establecerse la duración de cada flanco de subida o bajada. Por lo tanto, esto implica conocer el tiempo en que el eje está subiendo y bajando. Solo deberá multiplicarse el tiempo del ciclo que genera este módulo por el resultado que proporciona el CTUD.

En el mismo POU se realizan las multiplicaciones necesarias para la conversión y devuelve finalmente un valor de posición para la variable pos_z.

Pese a todo esto, no es una solución que garantice una fiabilidad elevada. Existen numerosos factores que no se pueden controlar y que generan un error en la posición real del eje z. Además, estos errores se van acumulando en cada movimiento del eje, por lo que tiene que intentarse limpiar continuamente estos errores que pueden causar imprecisión. Son errores principalmente producidos por la inercia de la máquina.

El error de posición producido no es igual en ambos movimientos del eje, por lo que dificulta su cálculo con precisión ya que, como se ha dicho anteriormente, uno de los principales problemas es el de la inercia, y debido al peso, el eje avanza más distancia después de frenar cuando va hacia abajo que cuando va hacia arriba. Este es un error que no se puede corregir ya que es imposible saber cuál es el valor de ese desplazamiento añadido.

El mismo error sucede en los otros dos ejes debido al movimiento extra después del frenado, pero este un error que se acepta ya que no supone alteraciones en el proceso y además no es acumulativo ya que, al contar con potenciómetro, se tiene en todo momento el valor real de la posición.

La solución al error acumulativo es resetear la variable cada vez que se activa el sensor inductivo que supone la coordenada 0 de este eje. El problema es que no se puede aumentar la velocidad del eje ya que se desconocería la posición.

Esto ha sido una solución a un problema existente por falta de algunos recursos como es el caso del sensor en este eje. En su aplicación en la industria no existiría este problema.

7.4. Variables Internas

Las variables internas son variables con las que trabaja el autómatas para realizar ciertas operaciones, no se encuentran físicamente en la máquina sensores ni salidas que correspondan a estas variables. No por esto son menos importantes, ya que son vitales en el proceso de programación y dan lugar a la creación de una infinidad de procesos.

Pueden utilizarse para indicar que un proceso se ha realizado o para que se inicie otro, activando estas variables desde el panel de control.

En el presente proyecto se han hecho uso de las variables internas para la programación, la gran mayoría de ellas declaradas a nivel global, es decir, todos los programas pueden interactuar con ellas. Todas las variables están declaradas en la lista de variables globales de soMachine donde se indica el nombre y el tipo de variables. En ocasiones las variables serán booleanas, enteras y en algunos casos reales.

A continuación, se procede a describir estas variables ya que son parte fundamental para entender la programación de este trabajo.

Variables de control

Como su nombre indica ejercen un control sobre el proceso. Las variables auto, Manual_maq y manual_pan son las que indican el proceso que se está realizando. La primera activa el modo automático mientras que las otras dos el modo manual, la primera desde la máquina y la segunda desde el panel de control.

```
33  
34      (*Variables de control*)  
35      Manual_maq: BOOL;  
36      Manual_pan: BOOL;  
37      auto: BOOL;  
38      ir: BOOL;  
39      proceso_marcha: BOOL;  
40      ocupado: BOOL;  
41
```

Figura 20. Variables de Control

La variable ir da la orden a la máquina, dentro del modo automático, de recoger una pieza y depositarla en la casilla seleccionada.

La variable ocupado indica que la celda seleccionada está ocupada y por lo tanto no se iniciara el proceso.

Por último, la variable proceso_marcha indica que la máquina está realizando un proceso, por lo que no dejara seleccionar otra casilla diferente ni tampoco pulsar el botón ir hasta que se desactive la variable.

Selección de celda

Son variables que indican la casilla seleccionada por el operario. Será donde se deposite la pieza una vez recogida. El valor de la variable será 1 cuando esté seleccionada y 0 cuando no. El proceso de selección se explica más adelante en "8. CONTROL DESDE PANTALLA". La variable casilla_escogida indica que alguna de las anteriores vale 1, y se puede iniciar el proceso

Estado de celda

Esta variable indica que en dicha celda ya hay una pieza, por lo que si se selecciona una casilla ocupada no debe iniciarse el proceso. Esta variable se activa cuando hay una casilla seleccionada y se pulsa el botón ir. Las variables son iguales que en la Figura 21 pero con el formato OC en lugar de C. Existe una variable llamada casilla_elegida que indicará que se ha iniciado el proceso y se asignará a variable de estado correspondiente un 1.

```
57 (*Selección de celda*)
58 C11: BOOL;
59 C12: BOOL;
60 C13: BOOL;
61 C14: BOOL;
62 C15: BOOL;
63 C21: BOOL;
64 C22: BOOL;
65 C23: BOOL;
66 C24: BOOL;
67 C25: BOOL;
68 C31: BOOL;
69 C32: BOOL;
70 C33: BOOL;
71 C34: BOOL;
72 C35: BOOL;
73 C41: BOOL;
74 C42: BOOL;
75 C43: BOOL;
76 C44: BOOL;
77 C45: BOOL;
78 casilla_escogida: BOOL;
```

Figura 21. Selección de celda

Posición real de los ejes

Se trata de variables reales que surgen de la conversión de variables enteras analógicas proporcionadas por los potenciómetros y calculadas por el autómata en el caso del eje z. Muestran la posición en cada momento de la horquilla en las tres coordenadas cartesianas. Muy importante para el proceso automático.

```
51
52 (*Posición real de los ejes*)
53 pos_x: REAL;
54 pos_y: REAL;
55 pos_z: REAL;
```

Figura 22. Posición de los Ejes

Posición de recogida

Guardan el valor de la posición de la pieza en el punto de recogida, hay una variable para cada eje de coordenadas excepto en el eje x, donde incluye también la posición de enganche de la pieza. De esta forma se situará delante de la pieza en la posición Rx y posteriormente la cogerá yendo a la posición Rx_eng.

```
103 (*Posición de recogida*)
104 Rx: INT:=330;
105 Rx_eng: INT:=260;
106 Ry: INT:=0;
107 Rz: INT:=250;
108
```

Figura 23. Posición de Recogida

Posición del depósito

En este caso una de las variables será fija, las otras dos dependerán de la casilla escogida, tomando un valor u otro en función de esto. La variable x siempre tendrá el mismo valor por estar todas las celdas situadas a la misma distancia de este eje.

```
109 (*Posiciones de depósito*)
110 Dx: INT:=70;
111 y_deposito : INT;
112 z_deposito : INT;
```

Figura 24. Posiciones de Depósito

VARIABLES DE CONTROL DESDE PANEL

Son muy similares a las variables de entradas y salidas físicas que existen en la máquina, pero creadas con el único fin de controlarla desde el panel de control. Se ha asignado un nombre muy parecido al que tiene las variables de las entradas y salidas de la máquina.

```
42 (*Variable control desde panel*)
43 xp_fwd: BOOL;
44 xp_rev: BOOL;
45 yp_fwd: BOOL;
46 yp_rev: BOOL;
47 zp_fwd: BOOL;
48 zp_rev: BOOL;
49 sub_vel: BOOL;
50 baj_vel: BOOL;
```

Figura 25. Control desde Panel

PUESTA A CERO

Consiste en dos variables, donde Cero da paso a iniciar la puesta a cero y CeroHecho indica que ya se ha realizado dicha operación. Se explicará con más detalles más adelante.

```
3
4 (*Variable puesta a cero*)
5 Cero : BOOL;
6 CeroHecho : BOOL;
7
```

Figura 26. Variables de puesta a Cero

7.5. Modos de Funcionamiento

Se distinguen dos claros modos de funcionamiento, el manual y el automático. La selección de estos modos se realiza desde el panel de control como se verá en el siguiente apartado. El modo manual a su vez incluye dos sub-modos que son el manual desde panel y manual desde máquina. A continuación, se procede a explicar cada uno de los modos.

Manual desde máquina

Una vez seleccionado este modo, el operario puede manejar la máquina desde los interruptores presenten en la máquina. Hay un interruptor para eje y sentido de movimiento. Además, tiene dos pulsadores para aumentar o disminuir la velocidad de giro de los motores. Mientras no esté seleccionado este modo, el accionamiento de los interruptores no producirá movimiento alguno.

Cuando este modo este seleccionado, en la pantalla habrá un mensaje que lo indique además de unos indicadores que mostrarán la posición de los ejes.

Este modo no se puede ver como un proceso automatizado, pese a estar controlado por el autómatas, ya que requiere movimiento constante por parte del operario, pero es un modo muy útil ya que múltiples circunstancias pueden obligar a realizar un proceso que requiera la mano humana. Por ello se implementa este modo que pone solución a este problema.

Además, puede servir para cualquier tarea de mantenimiento o en la cual se quiera revisar cualquier eje o motor.

En la siguiente figura se muestra el aspecto de los controladores situados en la máquina mencionados anteriormente.



Figura 27. Interruptores de la máquina



Manual desde panel

Cuando se selecciona este modo, el manejo de la máquina llega desde el panel de control. Es un modo muy similar al anterior, pero permite su control a distancia. Además, debido a las múltiples opciones que proporciona el panel, permite una programación más sencilla que en el caso de manual desde máquina.

En cuanto a la justificación de este modo, se siguen las mismas razones que se han explicado en el modo manual desde máquina. Con la particularidad de que este modo se realiza desde el panel de control.

A parte de por lo mencionado anteriormente, este modo concreto tiene también la ventaja de que, en caso de avería de cualquier interruptor, puede seguir utilizándose el modo manual controlándolo desde la pantalla.

El aspecto del control de este modo se mostrará más adelante cuando se hable de la pantalla que realiza la función de panel de control.

Automático

Este modo recoge una pieza de la zona de recogida automáticamente y la lleva a la casilla seleccionada. Para este modo se debe realizar una puesta a cero previa, para situar a la máquina en un lugar fijo de referencia.

Se avisará al operario de que la puesta a cero ha finalizado mediante un mensaje en el panel de control.

Una vez hecho esto, el operario selecciona una casilla y ordena el inicio del proceso. Una vez terminado podrá seleccionar otra casilla que esté libre para efectuar otro proceso tantas veces como se quiera mientras que existan casillas libres en el depósito.

En este proceso no puede haber dos casillas seleccionadas a la vez ni llevarse una pieza a una celda que esté ocupada.

Una vez seleccionada una celda, el operario podrá cambiar de selección siempre y cuando no haya pulsado el botón ir para iniciar el proceso. Una vez hecho esto deberá esperar a que termine para seleccionar otra diferente.

Este proyecto se centra sobre todo en el modo automático, ya que es el que más dificultad conlleva y es el que más utilidad tiene, pero se ha visto conveniente también incorporar también los modos manuales.

En el siguiente apartado se explica la programación que se ha llevado en cada uno de ellos para conseguir el buen funcionamiento del proceso.

7.6. Explicación de los Programas

En primer lugar, cabe destacar que la programación realizada ha sido para la base de lo podría ser el proceso industrial. Se ha centrado únicamente en la deposición de las piezas sin tener en cuenta datos que sí que podrían valorarse en la industria. Por ejemplo, la celda de depósito más utilizada podría ser la que más cerca estuviese. Control desde ordenador vía Ethernet. Agilizaciones de los procesos basados en datos estadísticos recogidos a lo largo de numerosos procesos entre otras cosas.

La programación se ha estructurado en cuatro programas distintos, agrupados según su lenguaje de programación, importancia o necesidad. En la siguiente figura se puede ver que POU se han añadido y el orden en el cual están.

Prioridad (0..31): 15

Tipo: Cíclico Intervalo (por ejemplo t#200ms): 20 ms

Watchdog: Activar
Tiempo (por ejemplo t#200ms): 100 ms
Sensibilidad: 1

POU:

POU	Comentario
ModoAuto	
PosicionZ	
TextEstruc	
ZonaSeguridad	

Buttons: [Agregar POU](#), Borrar POU, Abrir POU, Cambiar POU..., Subir, Bajar

Figura 28. POUs Incluidos

El orden en el que están situados también tiene cierta importancia que se explicará en detalle dentro de la explicación de cada POU.

7.6.1. Programa del Modo Automático

Este programa ha sido diseñado en lenguaje SFC, un lenguaje basado en GRAFCET. Este lenguaje es muy útil para este tipo de modos ya que actúa por etapas, de esta forma va siguiendo los procesos deseados de manera correlativa. Es sencillo y bastante intuitivo.

Dentro de este programa se ha incluido la puesta a cero, necesaria para la recogida posterior de las piezas. Una vez realizada la puesta a cero, si se quiere depositar otra pieza no será necesario ya que el programa se saltará las etapas correspondientes de este proceso.

Las transiciones de este modo se han escrito en texto estructurado (ST) cuando se requería una condición de mayor o igual o en casos que se requerían comandos como el AND o el OR. También existen transiciones sencillas en las cuales solo se necesita el cumplimiento de una variable global.

El diseño de esta programación es como se muestra en la Figura 29.

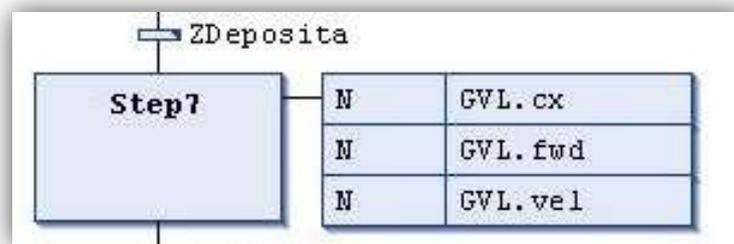


Figura 29. Aspecto SFC

Como se ve, cada etapa lleva asociadas unas acciones que están activas mientras lo esté la etapa. Para que una etapa esté activa, debe cumplirse que sea cierta su transición anterior y que la etapa anterior esté activa.

La etapa dará paso a la siguiente cuando se cumpla la transición que está a continuación, desactivando así las variables que tiene asociadas.

Este programa sigue el procedimiento siguiente, la máquina realiza la puesta a cero después que el operario presione el botón correspondiente. Una vez hecho esto la máquina estará esperando hasta que se seleccione una casilla. Una vez seleccionada, la máquina seguirá esperando hasta que se pulse el botón Ir. En caso de presionarse sin haber seleccionada casilla, o habiendo seleccionado una casilla vacía, no se iniciará el proceso.

Una vez iniciado, se desactivará la selección de celdas y el botón Ir, aparecerá un mensaje de que hay un proceso en marcha que no desaparecerá hasta que la máquina pare.

El primer paso que realiza la máquina es descender el eje Z para la recogida de la pieza. Una vez ha llegado a la posición, inicia el movimiento el eje X que efectuará el enganche de la pieza, cuando la pieza este en la horquilla, la máquina empezará el movimiento hacia la casilla seleccionada por el operario, realizando para ello los movimientos pertinentes en el eje Z e Y.

Una vez este frente la casilla con la pieza, el eje X la introducirá en la celda y una vez hecho esto, el eje Z bajará una pequeña distancia para soltar la pieza dentro de la casilla, ya que si no al volver atrás se quedaría la pieza en la horquilla.

Una vez depositada la pieza en la casilla correspondiente, la máquina volverá a su posición inicial y se pondrá a la espera de un nuevo proceso. Cuando haya llegado a la posición de espera, volverá a aparecer la selección de celdas y el botón Ir. El mensaje de proceso en marcha desaparece y puede iniciarse de nuevo un proceso.

Se podrá seleccionar la misma casilla que antes, pero cuando se pulse el botón Ir no se realizará movimiento alguno ya que la casilla estará ocupada, condición que no permite a la máquina empezar un nuevo proceso.

Este programa trabaja principalmente con variables de posición. Una gran mayoría de las transiciones son del tipo $pos_x \geq Rx$ o similares. No se puede introducir una igualdad para las transiciones ya que al ser las variables de posición variables reales, puede ser que el valor nunca llegue a coincidir exactamente, por lo que el eje pasaría de largo y la transición nunca se cumpliría, provocando el fallo del proceso.

En el programa se utilizan todas las variables de posición vistas en el apartado anterior, ya que han sido creadas para el modo automático, ya que el modo manual, o bien no se utiliza para recoger y depositar piezas o bien esta tarea reside en la pericia del operario, ordenando a la máquina cada movimiento para lograr tal fin.

Para que el programa no requiera realizar una puesta a cero en cada proceso, se ha añadido en la etapa de después la activación de una variable llamada *CeroHecho*. Cada transición de la puesta a cero tiene incluida la condición de que, si esta variable está activa, pasa directamente a la siguiente etapa, de esta forma se evitan pérdidas grandes de tiempo.

Este programa mueve los ejes a velocidad rápida excepto cuando esté cerca de la posición de cero o de la zona de depósito. El eje z por no disponer de potenciómetro se mueve siempre a velocidad lenta, esto ralentiza significativamente el proceso pero no puede ser de otra manera que aumentando la velocidad no sería posible calcular la posición. Además el error por la inercia aumentaría demasiado, a valores muy significativos que imposibilitarían la correcta deposición de las piezas.

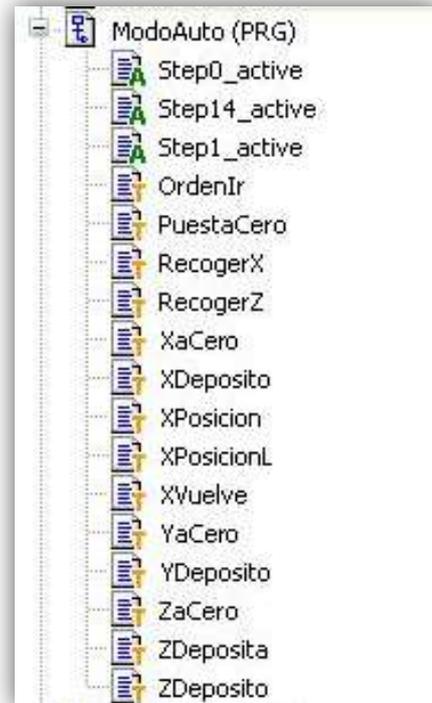


Figura 30.
Transiciones y
Acciones en ST

7.6.2. Programa de Posición Z

La posición de los ejes X e Y viene dada por los potenciómetros situados en cada uno de ellos. Esto permite un mayor control y reduce el error, ya que cualquier movimiento extra realizado por los ejes debido a la inercia o cualquier otro facto es también recogido por estos sensores.

Como se ha dicho anteriormente, el eje z no dispone de potenciómetro por lo que conocer su posición va a depender únicamente de este programa, por ello se ha intentado maximizar la precisión. Aunque surgen errores como un mayor desplazamiento hacia abajo que hacia arriba debido al desplazamiento extra que produce el propio peso del eje. Se ha conseguido una aproximación bastante aceptable, que permite ejecutar los procesos de forma correcta. Este error no se acumula demasiado debido a que cada vez que termina la deposición de una pieza se pasa por el sensor inductivo, reiniciando así la variable y eliminando el error.

Para ello se ha utilizado el lenguaje FBD junto con LD, que gracias a un contador y unos generadores de pulsos han permitido el control de posición de este eje.

La pieza principal es el contador CTUD, que cuenta tanto hacia abajo como hacia arriba, de esta forma el tiempo de bajada suma la cuenta y el tiempo de subida baja la cuenta, hasta que llega a tocar el sensor inductivo donde se hace un reset del contador. El único dato de interés en este contador es el valor de la variable que se cuenta, por lo que el valor de QU no ha sido de importancia. Este valor emite una señal cuando la cuenta coincide con un valor indicado, debido a que es un dato irrelevante se ha puesto una variable interna del programa la cual no se tendrá en cuenta en ningún proceso.

Al mismo tiempo cabe destacar que, este es un contador que solo realiza la cuenta cuando recibe un pulso. Si se colocaran los contactores correspondientes al movimiento hacia arriba y hacia abajo en las entradas CU y CD, no se produciría una cuenta correcta, ya que solo contaría el momento en que se activan estas señales, así que sería inservible. Una solución ha sido incorporar entre estos contactores de movimiento el contador, unos módulos BLINK, que son módulos generadores de pulsos que estarán activos cuando sus movimientos correspondiente estén activos, de esta forma se soluciona el problema del contador.

Los módulos se han diseñado de la siguiente forma, tienen un tiempo de subida de 100ms y uno de bajada igual y están conectados a los contactores de subida y bajada del eje como se muestra en la siguiente figura.

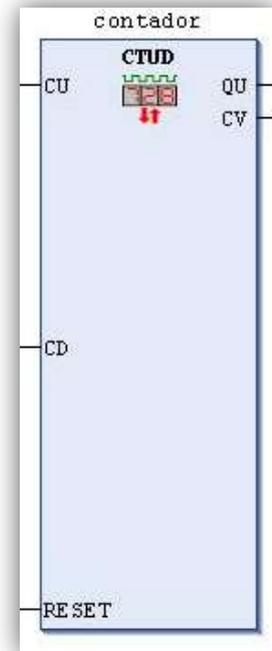


Figura 31.
Contador

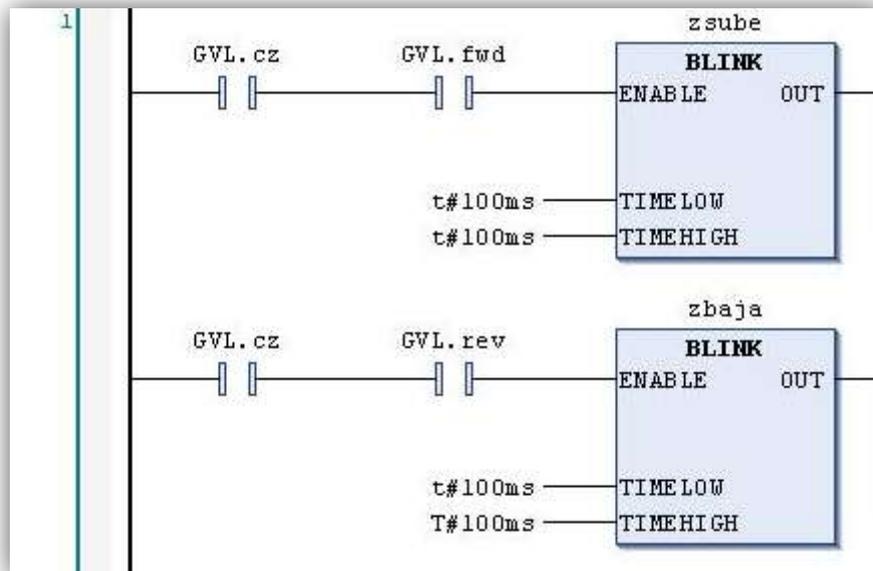


Figura 32. Generadores de Pulsos

Como se puede ver, con el tiempo de bajada y subida establecido, el programa generará cinco pulsos cada segundo, no sería la cuenta deseada, ya que se están contando los segundos que está el eje en movimiento. Por lo tanto se ha dividido entre cinco la variable resultante del contador, o lo que es lo mismo multiplicarla por 0,2. Para ello se ha hecho uso del operador multiplicación que ofrece este tipo de programación.

Una vez hecho esto ya se tiene una cuenta del tiempo de movimiento de la horquilla en eje z, tan solo falta multiplicarlo por la distancia que recorre en cada segundo, que en este caso son 4 milímetros.

Con este programa se ha solventado de una forma más o menos fiable el problema de la falta de sensor en el eje z. Las dimensiones del almacén, las piezas y la horquilla permiten el pequeño error inducido en el eje, por lo que no supone un grave problema.

7.6.3. Programación en Texto Estructurado

Se ha escogido este tipo de programación debido a que las acciones que se ejecutan se pueden realizar con condiciones simples utilizando comandos como el IF.

Toda la programación en texto estructurado se ha agrupado en la misma POU, excepto la zona de seguridad, que se explicará más tarde el motivo, en el siguiente apartado.

Esto se ha hecho ya que el tipo de programación lo permite y no interfieren entre ellos, de esta forma se evita la introducción excesiva de POU en el autómatas.

Pese a ser el mismo programa, dentro de este se distinguen claramente diferentes programas que hubieran podido ir en POUs independiente, a continuación, se va a proceder a explicar su funcionamiento.

Manual desde máquina

Es la parte de la programación que permiten manejar la máquina desde los interruptores existentes en la misma, para ello debe activarse la variable Manual_maq, que incluye dentro de un comando IF todas las condiciones que deben cumplirse para su funcionamiento.

En la Figura 33 se muestra un fragmento de la programación donde se puede apreciar el comando IF que permitiría la ejecución de todos los siguientes. Es decir, cuando se active este modo de funcionamiento desde la pantalla. Una vez desactivado todas las variables que estén incluidas dentro se desactivarán también.

Como se puede ver, se trata de un programa sencillo donde únicamente existe una respuesta a cada señal de entrada que se activa, por esta razón se ha realizado en texto estructurado.

```
1  IF (Manual_maq=1)
2  THEN
3    IF (x_fwd=1 AND x_rev=0)
4    THEN
5      cx:=1;
6      fwd:=1;
7    END_IF;
8
9    IF (x_rev=1 AND x_fwd=0)
10   THEN
11     cx:=1;
12     rev:=1;
13   END_IF;
14
15   IF (y_fwd = 1 AND y_rev=0)
16   THEN
17     cy:=1;
18     fwd:=1;
19   END_IF;
20
```

Figura 33. Programación Manual desde Máquina

Manual desde panel

Este programa es muy parecido al anterior, se basa en la misma lógica y solamente cambian las variables incluidas en él, por ello no se va a desarrollar una mayor explicación, las variables que utiliza son las vistas anteriormente en el apartado “7.4 Variables Internas”. Concretamente las variables de control desde panel. Se expondrá el programa completo más adelante junto con el anterior en el Anexo I.

Conversión de entradas analógicas

Como se ha visto anteriormente, los valores que llegan desde los potenciómetros son valores analógicos que no se corresponde con la posición real de los ejes. Para tener unos valores de utilidad se ha tenido que realizar una conversión de esta señal, a valores de posición en milímetros. Para ello se ha medido la posición inicial y final y leído los potenciómetros en cada caso, para realizar una recta y obtener su ecuación.

Una vez obtenido, se procede a implementar un programa de texto estructurado que simplemente ejecuta la ecuación siempre que no se encuentre el eje en los sensores inductivos y aplica el valor a las variables seleccionadas. Mientras el eje este activando el sensor inductivo, la variable cogerá el valor de 0.

Como se puede ver no existe ecuación para el eje z ya que su valor no viene dado por los potenciómetros si no por el programa visto anteriormente del control de posición del eje z.

```
150 IF (zero_x=1)
151 THEN
152     pos_x:=0;
153 END_IF;
154
155 IF (zero_x=0)
156 THEN
157     pos_x:=452.999332-0.030431*xanal;
158 END_IF;
159
160 IF (zero_y=1)
161 THEN
162     pos_y:=0;
163 END_IF;
164
165 IF (zero_y=0)
166 THEN
167     pos_y:=406.81973-0.01878*yanal;
168 END_IF;
```

Figura 34. Conversión de entradas analógicas

Casilla escogida

La siguiente programación sirve para indicar al autómatas que el operario ha seleccionado una casilla en el modo automático para que permita iniciar un proceso. De esta forma, seleccionar la orden de empezar el proceso no surtirá ningún efecto sin una casilla seleccionada.

En la siguiente Figura se muestra un fragmento, ya que por su longitud y repetitividad no es necesario mostrarlo entero.

```
IF (C11=1 OR C12=1 OR C13=1 OR C14=1 OR C15=1 OR C21=1 OR C22=1 OR C23=1 OR C24=1 OR C25=1 OR
THEN
casilla_escogida:=1;
ELSE
casilla_escogida:=0;
END_IF;
```

Figura 35. Selección de Casilla

Posición de las celdas

Cada celda tiene una posición concreta, todas comparten la misma coordenada en el eje X, pero cada una cuenta con unas coordenadas X e Y diferentes. Es cierto que las casillas de una misma columna tendrán la misma coordenada Y, el mismo caso para las filas y la coordenada Z, pero se debe distinguir claramente la posición particular de cada una de ellas, que nunca es igual para dos casillas distintas, por lo tanto, se ha diseñado el siguiente método, donde dependiendo de la casilla seleccionada por el operario, las variables creadas para indicar la posición del depósito cambian. El programa es como se ve en la Figura 36. Solamente se muestra hasta la celda C14, pero sería el mismo procedimiento para el resto con diferentes coordenadas cartesianas. Todas las posiciones están indicadas en milímetros y como se puede ver solo debe cumplirse la condición de que la casilla este seleccionada para la edición de las variables.

```
177 IF (C11=1)
178 THEN
179     y_deposito:=0;
180     z_deposito:=0;
181 END_IF;
182
183 IF (C12=1)
184 THEN
185     y_deposito:=70;
186     z_deposito:=0;
187 END_IF;
188
189 IF (C13=1)
190 THEN
191     y_deposito:=140;
192     z_deposito:=0;
193 END_IF;
194
195 IF (C14=1)
```

Figura 36. Posición de Celdas

Ocupación de casilla

Esta programación sirve para saber que una casilla está ocupada. La necesidad de este programa surge que no bastaría con activar estas variables OC tan solo con la selección de una casilla, ya que el operario puede ir seleccionando casillas y cambiando su selección antes de iniciar el proceso. Por ello debe coincidir esa selección con la variable casilla_elegida, que se activará cuando se inicie un proceso. De esta forma solo contabilizarán como casillas ocupadas las que se hayan seleccionada y, además, posteriormente, hayan iniciado el proceso de deposición de la pieza.

```
297 IF (C11=1 AND casilla_elegida=1)
298 THEN
299     OC11:=1;
300 END_IF;
301
302 IF (C12=1 AND casilla_elegida=1)
303 THEN
304     OC12:=1;
305 END_IF;
306
307 IF (C13=1 AND casilla_elegida=1)
308 THEN
309     OC13:=1;
310 END_IF;
311
312 IF (C14=1 AND casilla_elegida=1)
313 THEN
314     OC14:=1;
315 END_IF;
```

Figura 37. Ocupación de Casilla

Casilla ocupada

Este programa consiste en activar la variable ocupado en caso de haber seleccionado una celda que ya haya sido ocupada. Esta variable de la que se ha hablado anteriormente, no permite que no se inicie el proceso si la casilla seleccionada por el operario es una casilla ya ocupada, es decir, tiene la variable OC activada. De esta forma se evita la deposición de dos piezas en la misma celda ya que el almacén está diseñado para almacenar únicamente una pieza por casillero. Dependiendo del tipo de industria y almacén, esto podría adaptarse a la deposición de un número limitado de piezas diferente de 1, mediante contadores, por ejemplo. Como este caso solo admite una pieza por casillero, se ha implementado la variable ocupado para que no se acumule más de una pieza por celda.

Una vez el operario seleccione otra casilla diferente a que está ocupada, la variable ocupada volverá a su valor de 0 y podrá iniciarse de nuevo el proceso.

```
397 IF (C11=1 AND OC11=1)
398 THEN
399 ocupado:=1;
400 END_IF;
401
402 IF (C12=1 AND OC12=1)
403 THEN
404 ocupado:=1;
405 END_IF;
406
407 IF (C13=1 AND OC13=1)
408 THEN
409 ocupado:=1;
410 END_IF;
411
412 IF (C14=1 AND OC14=1)
413 THEN
414 ocupado:=1;
415 END_IF;
416
417 IF (C15=1 AND OC15=1)
418 THEN
419 ocupado:=1;
420 END_IF;
421
422 IF (C21=1 AND OC21=1)
423 THEN
424 ocupado:=1;
425 END_IF;
426
```

**Figura 38. Casilla
Seleccionada Ocupada**

7.6.4. Programa de Zona de Seguridad

Este programa es el que impide que la máquina llegue a los finales de carrera que cortan la alimentación y que llegue a los límites establecidos con la velocidad de giro del motor alta.

Por esta razón se ha colocado al final, detrás de los otros POU, para que sea el último en leerse y en caso de que este dé la orden de parada del motor, sea esta señal la que finalmente el autómeta transmite a la máquina. De esta forma, aunque otro programa, por sus condiciones, ordene al motor seguir girando, si este programa la cancela no se dará movimiento alguno del motor.

Está escrito en texto estructurado, pero debido a las características indicadas anteriormente no se ha incluido en la programación del apartado anterior.

La programación se basa la posición de cada eje. Este programa lee las variables de posición de los tres ejes y limita su velocidad o movimiento. En el caso del eje Z, al no disponer de distintas velocidades solamente se limita su movimiento.

```
1  IF (pos_x>410 AND cx=1)      16  IF (zero_x=1 AND cx=1)
2      THEN                    17      THEN
3      fwd:=0;                 18      rev:=0;
4  END_IF;                     19  END_IF;
5
6  IF (pos_x>380 AND cx=1)     20
7      THEN                    21  IF (pos_y>270 AND cy=1)
8      vel:=0;                 22      THEN
9  END_IF;                     23      fwd:=0;
10
11 IF (pos_x<50 AND cx=1)      24  END_IF;
12     THEN                    25
13     vel:=0;                 26  IF (pos_y>240 AND cy=1)
14 END_IF;                     27     THEN
15                             28     vel:=0;
16                             29  END_IF;
17                             30
```

Figura 39. Zona de Seguridad

Como se ve la Figura anterior, se marcan unos límites para las velocidades y unos un poco más amplios para el movimiento, de esta manera, el eje llega a la posición final con velocidad reducida para que la frenada sea más precisa y no llegue a tocar el final de carrera.

7.7. Seguridad

En la programación no se ha diseñado ningún programa de seguridad específico salvo el de Zona de Seguridad comentado anteriormente que limita la máquina a ciertas coordenadas. El grado de seguridad del programa reside en que al ser el último no permite activar las salidas cuando no dentro del margen de seguridad.

En cuanto a la alimentación de la máquina, se ha dejado la seguridad para los finales de carrera instalados en la máquina de los cuales se habla en más detalles en el Anexo II, estos cortan la alimentación en caso de el programa implementado de seguridad fallara por cualquier motivo, cosa muy poco probable.

Si se produjese alguna situación que pueda poner en peligro la máquina o la integridad física de alguna persona que este manejándola, o simplemente esté situada en una zona de riesgo, la máquina dispone de una seta de emergencia que corta la alimentación de la máquina inmediatamente después de su pulsación.



Figura 40. Seta de Emergencia

De esta forma la seguridad de la máquina reside en tres puntos clave que son los siguientes:

- Programa Zona de Seguridad
- Finales de carrera
- Pulsador de emergencia

8. CONTROL DESDE PANTALLA

La pantalla utilizada ha sido el modelo Magelis XBTGT2110 de Schneider y ha servido como panel de control del autómat, siendo una pieza fundamental tanto para el control como para el seguimiento del proceso.

De esta forma se ha realizado un SCADA desde el SoMachine que es visible desde la pantalla. El diseño se confeccionó a partir de la herramienta de Vijeo Designer que lleva incorporada el software SoMachine para este tipo de programaciones con pantallas de control.

El programa Vijeo ofrece multitud de opciones para el diseño de la pantalla, cuenta con variables internas, alarmas, recetas y multitud de interruptores entre otras cosas. Como se ve en la Figura 41 del navegador del Vijeo, tiene una interfaz muy sencilla e intuitiva, permitiendo la fácil creación de pantallas así como una navegación rápidas entre ellas. Es posible editar el nombre desde el mismo navegador. Desde él se puede acceder a los ajustes del Administrador de E/S donde se ajusta la configuración de conexión de la pantalla, algo esencial para su correcta transmisión de datos. Este navegador permite un fácil acceso a herramientas importantes como la gestión de alarmas, que avisarán al operario de cualquier problema surgido en el proceso o las recetas, que permiten la realización de procesos en cadena, cambiando automáticamente el valor de las variables que se están utilizando en el proceso según la etapa. Otros elementos importantes como las variables o acciones están presentes también en este navegador.

Para la edición de los paneles el programa cuenta con múltiples elementos de utilidad como interruptores, botones, texto, imágenes, pilotos, líneas, etc.



Figura 41. Navegador Vijeo Designer



Figura 42. Herramientas del Vijeo

8.1. Conexión

La configuración de conexión de la pantalla con el autómatas se realiza desde el soMachine, donde se indica el modelo de la pantalla y la conexión que se va a realizar, en este caso ha sido vía ModBus como se puede ver en la siguiente Figura.

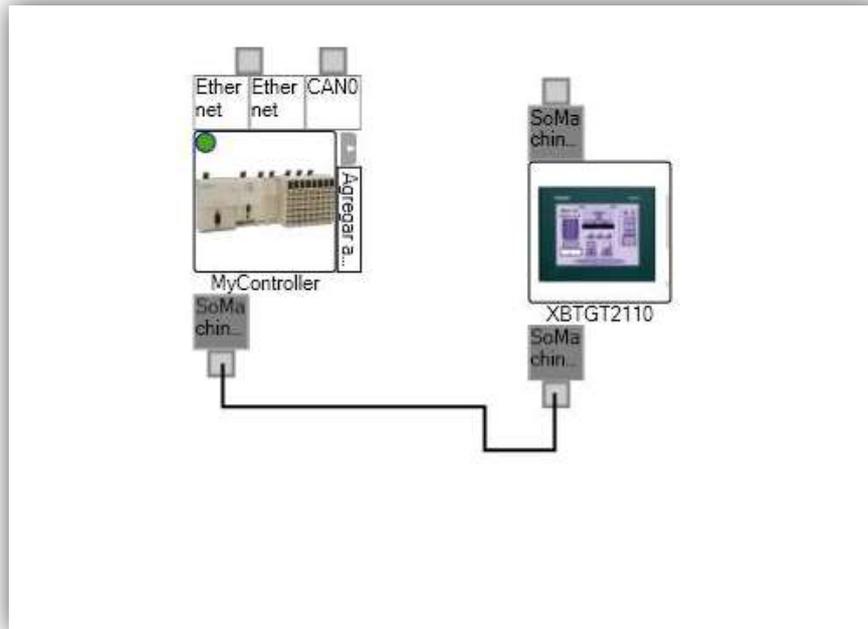


Figura 43. Conexión ModBus entre autómatas y pantalla

Esta configuración permite tener la pantalla únicamente conectada al PLC, sin necesidad de conexión al ordenador, la información va desde la computadora hasta la pantalla pasando por el autómatas, de esta forma se evitan conexiones excesivas.

Esta configuración no puede realizarse la primera vez que se conecta la pantalla, ya que en este caso requiere una conexión directa con el ordenador para su correcta configuración. Esta conexión se realiza mediante USB.

Una vez realiza esta primera descarga de datos, la pantalla puede empezar a funcionar correctamente vía ModBus con el autómatas.

Se debe tener cuidado en el enlace que se realiza dentro del programa, ya que debe coincidir con el puesto seleccionado, este caso se trata del COM2.

En caso de no coincidir el autómatas no podrá encontrar la pantalla y será imposible la transmisión de datos entre ellos.

8.2. Transferencia de Variables

Las variables que interactúan entre la pantalla y el autómatas deben estar declaradas en ambos programas, el soMachine y el Vijeo Designer, para ello, se transfieren las variables creadas en la programación, variables globales en este caso, al programa de la pantalla.

Para ello se debe realizar una transferencia de estas variables en soMachine, desde la configuración de símbolos.



Figura 44. Configurador de símbolos

La ventana izquierda corresponde a las variables de soMachine y la ventana derecha a las de Vijeo Designer, se pueden modificar los derechos de acceso de las variables.

Una vez transferidas, en el Vijeo se guardan de la siguiente forma.

Nombre	Tipo de datos	Origen de datos	Grupo de escaneo	Dirección de dis...	Grupo de alarmas	Grupo de registros
SoM						
MyController						
Application						
GVL						
auto	BOOL	Externo	SOM_MyController	Application.GVL.aut	Desactivado	Ninguno
baj_vel	BOOL	Externo	SOM_MyController	Application.GVL.baj	Desactivado	Ninguno
C11	BOOL	Externo	SOM_MyController	Application.GVL.C11	Desactivado	Ninguno

Figura 45. Variables en Vijeo

8.3. Pantallas Diseñadas

El Viejo permite el diseño de pantallas para su implementación en el panel de control, las pantallas diseñadas para este proyecto han sido las que se muestran en la siguiente Figura y que se explican a continuación.

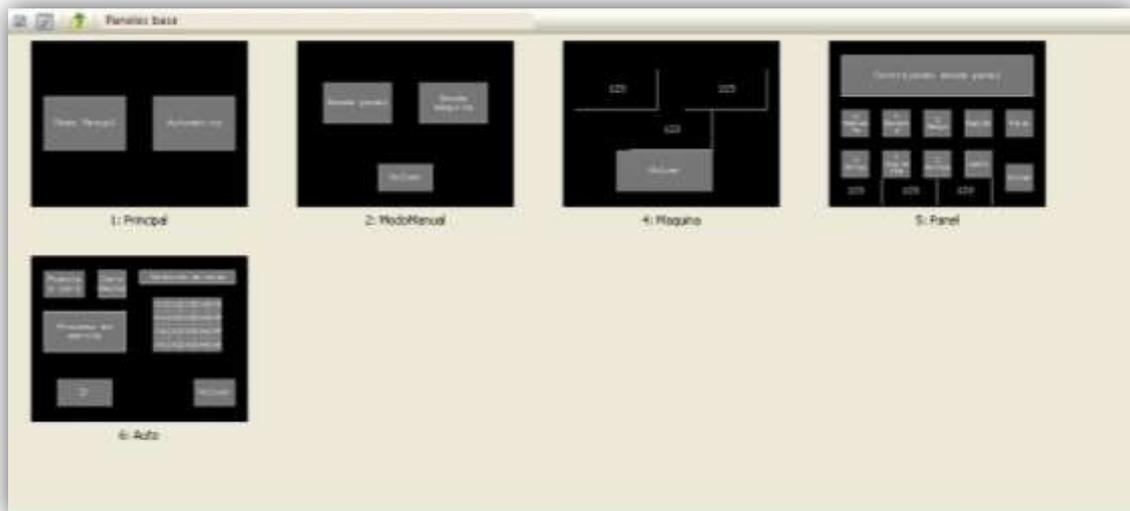


Figura 46. Pantallas Diseñadas

En los siguientes apartados se van a detallar las funciones de cada pantalla así como las características de los elementos que incluyen.

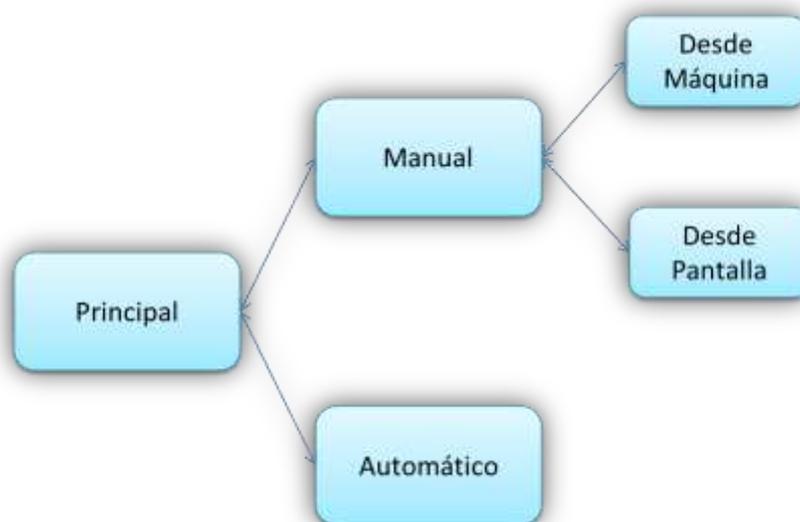


Figura 47. Relación entre pantallas

8.3.1. Pantalla Principal

Esta pantalla se muestra al inicio, es la pantalla que se muestra cada vez que se enciende el panel de control. Desde ella se accederá a las otras pantallas. Su aspecto es el siguiente.

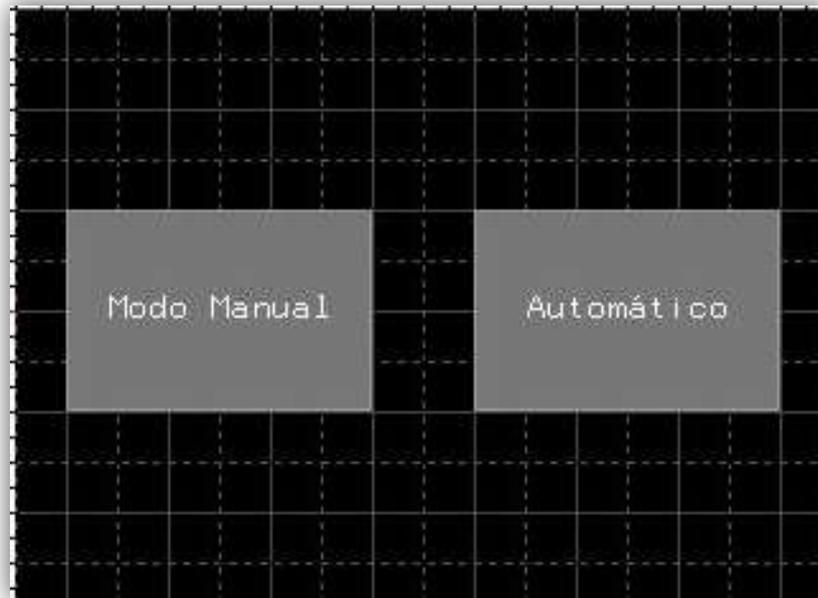


Figura 48. Pantalla Principal

Esta pantalla cuenta con dos interruptores que tienen la función de cambio de panel, uno lleva al modo manual mientras que el otro al automático.

Tanto esta imagen como las siguientes que se muestren del panel se han extraído del programa Vijeo Designer y no desde el panel de control para mejorar su visibilidad.

Los colores escogidos son los únicos que permite la pantalla, ya que es monocolor y solo permite distintas tonalidades de gris.

8.3.2. Pantalla Selección Modo Manual

Esta pantalla tiene las mismas funciones que la anterior, con la particularidad que contiene el botón Volver, para ver otra vez el panel principal. Los otros dos botones conducirán al modo manual desde máquina y el modo manual desde panel.

Estos botones, aparte de dirigir el panel hacia otra pantalla, según el botón que se pulse se activará una variable de control. En caso de pulsar el botón desde máquina, se activará la variable Manual_maq y si se pulsa el botón desde panel se activará la variable Manual_pan.

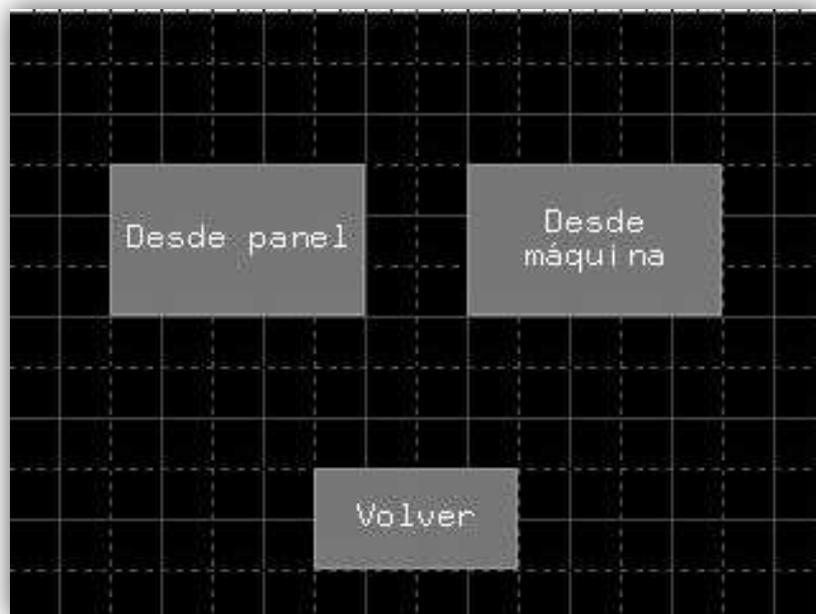


Figura 49. Pantalla Modo Manual

Se hubiera podido poner estos dos botones en la pantalla anterior, ya que el botón Modo Manual no realiza ninguna otra función que la de cambiar de panel. A diferencia de estas que activan sus variables correspondientes.

Se ha visto adecuado agrupar estos dos botones dentro de una pantalla individual para una mejor estructuración del panel.

Estos dos botones conducen a las pantallas que se van a exponer en los dos siguientes apartados.

8.3.3. Pantalla de Control desde Máquina

Una vez en este panel, la variable Manual_maq estará activada y se podrá manejar la máquina desde sus propios interruptores.

En esta pantalla se muestra un mensaje que indica que se está realizando el control desde la máquina, no necesita ms botones ya que en este caso se hace uso de los interruptores que están en la máquina.

Contiene tres visualizadores que muestran el valor de las variables de posición. La ventana de configuración de los visualizadores se muestra a continuación en la Figura 50.

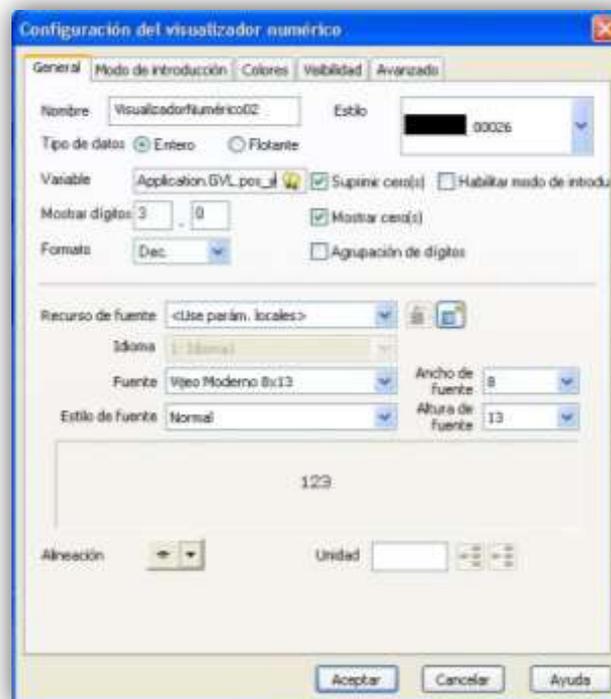


Figura 50. Configuración del visualizador

Para la configuración se selecciona la variable a mostrar y los dígitos que se quieran visualizar además del tipo de letra y tamaño.

En la siguiente Figura se puede ver el aspecto de este panel.



Figura 51. Pantalla Control desde Máquina

8.3.4. Pantalla Control desde Panel

Esta pantalla es similar a la anterior, una vez dentro de ella la variable `Manual_pan` estará seleccionada y se podrá manejar la máquina desde el panel.

La diferencia con la anterior es que esta, al no contar con interruptores físicos se debe incorporar una serie de interruptor en la pantalla para el control.

Se ha colocado un interruptor en el panel por cada interruptor físico que tiene la máquina, incluidos los de aumento de velocidad y disminución de velocidad.

Al igual que en la pantalla anterior, en esta también se cuenta con unos visualizadores para las variables de posición de los ejes. En este caso es más útil ya que al estar manejándose desde el panel puede no tenerse una visión directa de la máquina, por lo que estos visualizadores ayudan en el control de esta.

Cuenta con un letrero donde indica que se está manejando la máquina desde el panel de control, al igual que en el caso anterior.

El aspecto de esta pantalla es la que se muestra en la siguiente Figura, donde cada interruptor tiene la etiqueta del movimiento o función que realiza.



Figura 52. Pantalla de Control desde Panel

Como se ha dicho cada botón del programa se correspondería a cada interruptor físico de su misma función. El botón parar resetea los botones activos, de esta forma se detiene el movimiento de la máquina.

El funcionamiento es sencillo, cada interruptor cambia el bit de la variable interna que tiene seleccionada, de manera que pulsando una vez, se inicia el movimiento, pulsando de nuevo la máquina se parará.

Al mismo tiempo, pulsar el botón de movimiento contrario hace un reset de la variable contraria, es decir, si se pulsa por ejemplo X Adelante y posteriormente, sin desactivar este se pulsa X Atras, esta última hace un reset de la primera y activa su variable correspondiente. De esta forma se asegura el buen funcionamiento de la máquina.

El botón Volver desactiva la variable Manual_pan y lleva al panel a la pantalla de modo manual, donde se podrá seleccionar el otro modo de funcionamiento manual o podrá volverse atrás de nuevo para llegar al panel principal.

Todas estas variables que se activan desde el panel están en la programación del soMachine y han sido transferidas al Vijeo Designer desde la configuración de símbolos, como se ha explicado anteriormente.

8.3.5. Pantalla de Modo Automático

Por último se va a detallar el panel de modo automático. Esta pantalla es la más completa de todas, cuenta con animaciones de visibilidad, botones de selección e interruptores, como se muestra en la Figura 53.

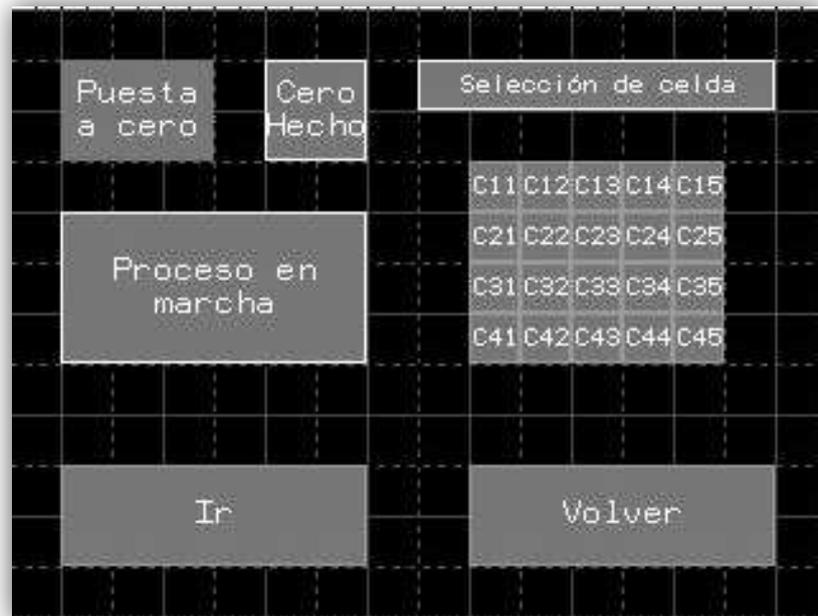


Figura 53. Pantalla Modo Automático

Este es el dibujo de la pantalla completa, pero durante su funcionamiento nunca se podrá ver de esta forma, ya que los elementos aparecen o desaparecen en función del estado del proceso. Esto se ha realizado mediante una animación de visibilidad.

De esta forma, en un momento inicial se podrán ver los elementos “Puesta a cero”, la selección de casillas, el botón “Ir” y el botón “Volver”, que estará siempre visible.

Una vez realizada la puesta a cero, el interruptor “Puesta a cero” desaparecerá para dejar paso al texto “Cero Hecho”.

Cuando un proceso de inicie, se activa la variable interna proceso_marcha, la activación de esta supone una desaparición de la selección de casillas y del botón “Ir”, para que no pueda seleccionarse otro proceso distinto mientras ya hay uno en funcionamiento.

Al mismo tiempo que desaparecen estos, aparece el mensaje “Proceso en marcha” para indicar al operario que debe esperar hasta la finalización. Una vez finalizado el proceso el mensaje desaparece y vuelven a aparecer las casillas de selección y el interruptor “Ir”.

La visualización de los botones se ha configurado desde la opción animaciones como se muestra en la siguiente Figura.

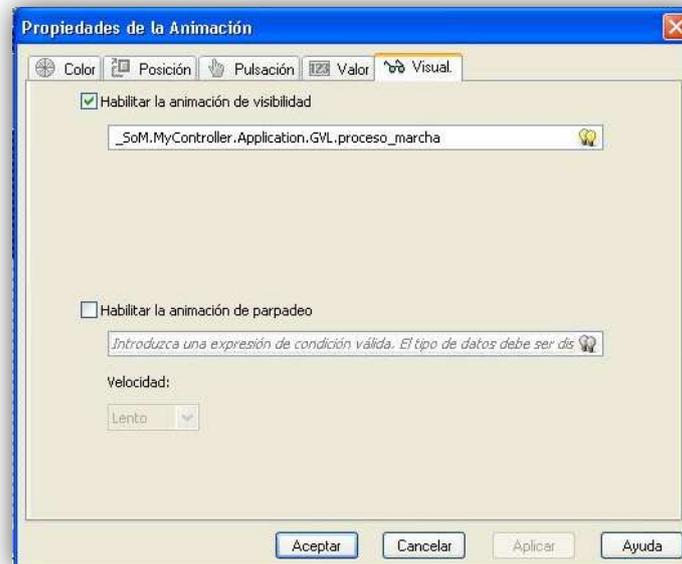


Figura 54. Propiedad de la Animación

La particularidad de los botones de selección de celda es que no son interruptores normales, si no que se trata de botones de selección. Esto ha sido de gran ayuda ya que de esta forma no puede haber más de una casilla seleccionada, lo cual dificultaría el correcto funcionamiento, pudiendo provocar errores de posicionamiento de las piezas.

Además ahorra mucho trabajo a la hora de configurar el panel, ya que de no ser así la otra opción sería colocar un reset de todas las demás casillas, lo cual supondría una pérdida de tiempo innecesaria.

Siempre que esté el panel del modo automático visible, la variable auto estará activada, que es la que da el paso al programa SFC para poder iniciar un proceso. Al pulsar el botón "Volver", la variable auto se resetea y la pantalla vuelve al panel principal.



9. CONCLUSIONES

Para cerrar esta memoria de contenido, se van a realizar las conclusiones del presente proyecto realizado en el departamento de ingeniería eléctrica.

En el trabajo realizado se han alcanzado los objetivos planteados inicialmente, la elaboración de un modo manual y otro automático con control desde pantalla con el autómatas que más se ha ajustado a las necesidades. El estudio de los autómatas ha sido un proceso positivo ya que se ha aprendido a relacionarse rápidamente con cada software de programación y se ha visto que sus similitudes son muy grandes en cuanto a programación se trata, es decir, que con la elaboración de este proyecto con el autómatas escogido también se han obtenido conocimientos para la realización de otro proyecto similar con otro autómatas diferente.

En cuanto a la programación, ya se tenían conocimientos previos obtenidos durante el grado, pero este proyecto los ha ampliado de forma notable debido a las complejas situaciones que presenta. Además, también se ha aprendido gracias a la búsqueda en internet sobre partes de la programación que se desconocían.

Ha sido gratificante y muy útil trabajar con pantalla además de con el autómatas, ya que se asemeja más a un proceso industrial real y ha permitido obtener conocimientos acerca de estas, de cómo conectarlas a los PLC y como configurarlas con el software adecuado. La pantalla permite una programación más amplia debido a que posibilita el manejo de las variables internas por parte del operario con el panel táctil y permite la visualización y control del proceso.

En resumen, la realización de este proyecto no solo ha ayudado a ampliar conocimientos de programación y autómatas, sino a mejorar la forma de analizar las situaciones, ya que en el mundo de los autómatas programables surgen muchos problemas hasta conseguir la programación correcta y hay que saber buscar entre todo el código el error, hecho que mejora la capacidad de análisis de un problema.

A nivel personal ha resultado ser un proyecto muy interesante y con gran aplicación al mundo laboral, ya que a día de hoy, cualquier industria puede contar con procesos de automatización como este o similares.



PRESUPUESTO



ÍNDICE DEL PRESUPUESTO

1.	INTRODUCCIÓN	63
2.	COSTE DE MATERIALES.....	64
2.1.	Circuitos Acondicionadores de Señal de Potenciómetros.....	64
2.2.	Elementos de Control.....	64
2.3.	Software	65
2.4.	Componentes de la Máquina	65
2.5.	Elementos de Seguridad.....	66
2.6.	Coste Total de Materiales	66
3.	COSTE MANO DE OBRA	67
4.	COSTES GENERALES.....	68
5.	MARGEN DE BENEFICIO.....	68
6.	COSTE FINAL DEL PROYECTO.....	68



1. INTRODUCCIÓN

En el presente documento se detallan los costes del proyecto mediante un análisis de todas las partes que han formado parte de este.

“El Presupuesto es el documento que refleja el coste del proyecto, la inversión necesaria para llevarlo a cabo”. (Gómez-Senent, 2000)

Los costes se analizarán primero de forma independiente según su naturaleza. De esta forma se podrá distinguir entre costes de materiales, mano de obra y costes generales.

Debido al carácter académico del proyecto se van a despreciar gastos generales como pueden ser el agua, la luz o algunos otros.

Una vez analizados estos gastos se les aplicará el margen de beneficio y posteriormente se añadirán los impuestos para obtener el coste definitivo del proyecto.

El documento tan solo se limita a analizar el coste económico del proyecto. Sin profundizar en los campos de estudio previo de mercado ni ningún tipo de análisis de las posibles repercusiones económicas. En el caso de no tratarse de un proyecto académico sí que se hubieran realizado estos estudios previos en la empresa contratante y a partir de estos, se hubiesen obtenido estimaciones de los resultados económicos finales.

Con dichas estimaciones y el presupuesto del proyecto se podría obtener el beneficio o ahorro final que supondría la realización del mismo. En función de este se discutiría la viabilidad del proyecto o el rechazo del mismo.

Se supondrá que la empresa encarga la automatización y control de una máquina de la cual ya disponen y necesitan su adaptación y programación del proceso.

En el presupuesto estarán incluidos todos los elementos necesarios para la automatizarla, incluyendo todos los sensores y piezas necesarias para la interpretación de la información proporcionada por los sensores, los elementos de control y seguridad, el autómatas y el panel de control, así como la mano de obra del contratista.

2. COSTE DE MATERIALES

En el presente apartado se van a exponer todos los elementos utilizados en el proyecto. Se ha tenido en cuenta el precio de venta al público.

2.1. Circuitos Acondicionadores de Señal de Potenciómetros

El autómata debe poder leer los datos enviados por los potenciómetros y para ello se debe amplificar la señal que generan. La función de estos circuitos es amplificar dicha señal. En la Tabla 8 se presentan los elementos.

Tabla 8. Circuitos Acondicionadores de Señal

DESCRIPCIÓN	UNIDADES	PRECIO	IMPORTE
Transistor BC 309	2	0,12 €	0,24 €
Resistencia 1,5k Ω	2	0,05 €	0,10 €
Resistencia 2,2k Ω	2	0,05 €	0,10 €
Resistencia 10k Ω	2	0,05 €	0,10 €
Placa de montaje	1	0,50 €	0,50 €
SUBTOTAL			1,04 €

2.2. Elementos de Control

Para el control y ejecución de los distintos procesos se han utilizado los elementos que se exponen en la Tabla 9.

Tabla 9. Elementos de control

DESCRIPCIÓN	UNIDADES	PRECIO	IMPORTE
TM258LF66DT4LS0	1	985,35 €	985,35 €
Magelis XBTGT 2110	1	637,23 €	637,23 €
Variador de frecuencia	1	301,20 €	301,20 €
Ordenador	1	424,90 €	424,90 €
Contactores	3	14,50 €	43,50 €
SUBTOTAL			2.392,18 €

2.3. Software

Este apartado describe únicamente el coste de adquisición de los programas informáticos utilizados en este proyecto. El software con el que se ha programado el autómatas ha sido el SoMachine del fabricante Schneider.

Tabla 10. Software utilizado

DESCRIPCIÓN	UNIDADES	PRECIO	IMPORTE
SoMachine	1	215,44 €	215,44 €
SUBTOTAL			215,44 €

2.4. Componentes de la Máquina

Se exponen todos los elementos que se han acoplado a la máquina para su control.

Tabla 11. Componentes

DESCRIPCIÓN	UNIDADES	PRECIO	IMPORTE
Sensores inductivos	3	35,32 €	105,96 €
Potenciómetros	2	495,90 €	991,80 €
Pulsadores	2	3,09 €	6,18 €
Botonera	1	15,23 €	15,23 €
Regletas de conexión	15	0,06 €	0,90 €
Manguera 4x1.5mm ²	10m	0,99 €/m	9,90 €
Manguera 4x2.5mm ²	10m	1,05 €/m	10,50 €
Cable flexible 1.5mm ²	5m	0,15 €/m	0,75 €
Otros			8,20 €
SUBTOTAL			1.149,42 €

2.5. Elementos de Seguridad

Son los encargados de la protección de la máquina y los usuarios que la utilicen antes posibles accidentes.

Tabla 12. Seguridad

DESCRIPCIÓN	UNIDADES	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 €
SUBTOTAL			43,30 €

2.6. Coste Total de Materiales

Tabla 13. Subtotal Materiales

DESCRIPCIÓN	IMPORTE
Circuitos Acondicionadores	1,04 €
Elementos de Control	2.392,18 €
Software	215,44 €
Componentes de la Máquina	1.149,42 €
Elementos de Seguridad	43,30 €
SUBTOTAL	3.801,38 €



3. COSTE MANO DE OBRA

Para el cálculo de la mano de obra se ha dividido el tiempo dedicado al proyecto en diversas tareas. Se ha considerado que lo realiza el autor del proyecto con la titulación que obtendría al finalizarlo. Por lo que se considerará como un trabajador con el Grado de Ingeniería en Tecnologías Industriales. Se ha establecido un precio de 50€/h.

Tabla 14. Mano de Obra

TAREA	HORAS	IMPORTE
Estudio previo	60	3.000,00 €
Puesta a punto	30	1.500,00 €
Programación	100	5.000,00 €
Diseño del control	80	4.000,00 €
Ensayos y comprobaciones	30	1.500,00 €
Ajustes generales	10	500,00 €
SUBTOTAL		15.500,00 €



4. COSTES GENERALES

En este apartado se incluyen los gastos comunes a todos los proyectos que se desarrollan en cualquier empresa. El agua, la luz, internet y teléfono, las instalaciones, impuestos generales, etc. Para incluir estos gastos en el presupuesto se aplica un porcentaje sobre los costes de materiales y mano de obra. Este porcentaje puede variar mucho en función de la empresa.

Estimando un porcentaje del 13% sobre los costes.

Material más mano de obra:

$$3.801,38 + 15.500,00 = 19.301,38 \text{ €}$$

Ecuación 5. Presupuesto de ejecución material

Tras aplicar el porcentaje de gastos generales:

$$19.301,38 * 0,13 = 2.509,18 \text{ €}$$

Ecuación 6. Gastos generales

5. MARGEN DE BENEFICIO

El beneficio industrial se aplica sobre el coste total de fabricación del proyecto y representa la parte que gana la empresa dependiendo del tipo de industria y la misma empresa. En este caso se aplicará un porcentaje del 6%.

$$19.301,38 * 0,06 = 1.158,08 \text{ €}$$

Ecuación 7. Beneficio industrial

6. COSTE FINAL DEL PROYECTO

Finalmente se debe aplicar el IVA que actualmente es de un 21%.

$$(19.301,38 + 2.509,18 + 1.158,08) * 1,21 = 27.792,05 \text{ €}$$

Ecuación 8. Presupuesto base de licitación

PRESUPUESTO FINAL DEL PROYECTO:

Veintisiete mil setecientos noventa y dos euros y cinco céntimos.



PLIEGO DE CONDICIONES



ÍNDICE DEL PLIEGO DE CONDICIONES

1.	INTRODUCCIÓN	71
2.	CONDICIONES DE ÍNDOLE TÉCNICA.....	72
	CAPÍTULO I: CONDICIONES QUE DEBEN REUNIR LOS MATERIALES.....	72
	CAPÍTULO II: CONTROL DEL FUNCIONAMIENTO.....	73
3.	CONDICIONES DE ÍNDOLE FACULTATIVA	74
	CAPÍTULO I: RESPONSABILIDADES DEL CONTRATISTA.....	74
	CAPÍTULO II: RESPONSABILIDADES DEL CONTRATANTE	74
4.	CONDICIONES DE ÍNDOLE ECONÓMICA.....	75
	CAPÍTULO I: BASE FUNDAMENTAL.....	75
	CAPÍTULO II: PRECIOS Y RECARGOS	75
5.	CONDICIONES DE ÍNDOLE LEGAL	76
	CAPÍTULO I: EL CONTRATO.....	76
	CAPÍTULO II: RESCISIÓN DEL CONTRATO	76



1. INTRODUCCIÓN

En el presente documento se van a detallar las condiciones que deben cumplir las partes que intervienen en este proyecto.

“Un pliego de condiciones es un documento de cláusulas, especificaciones, limitaciones o condiciones, debidamente articulados, que se precisan para poder realizar algo.” (Martínez y Pellicer, 2006).

En esta parte del proyecto se deben tener en cuenta condiciones de diversa índole. Las diferentes condiciones que se van a exponer en esta memoria son cuatro, entre las que se encuentran las técnicas, facultativas, legales y económicas.

Las condiciones técnicas incluyen todas las características técnicas mínimas que deben cumplir los elementos del proyecto durante su funcionamiento. Describe los elementos materiales que integran el objeto del proyecto.

En las condiciones facultativas se recogen todas las obligaciones y derechos que deben cumplir las partes implicadas. Regulan la ejecución del proyecto.

En cuanto a las condiciones legales, se expone la forma de adjudicación, el tipo de contrato, la obligación de contar con seguros de responsabilidad civil, etc.

En las condiciones económicas se presenta las formas de pago y las posibles indemnizaciones.



2. CONDICIONES DE ÍNDOLE TÉCNICA

CAPÍTULO I: CONDICIONES QUE DEBEN REUNIR LOS MATERIALES

NORMATIVA

Los equipos que se requieren para el proyecto, son componentes eléctricos que están conectados a la red eléctrica de corriente alterna de baja tensión con un voltaje de 230V y una frecuencia de 50Hz. El reglamento a seguir en una instalación de estas características es el Reglamento Electrotécnico de Baja Tensión (RBT) con sus Instrucciones Complementarias.

Las instrucciones técnicas complementarias a seguir en el presente proyecto son las siguientes:

- 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 sobretensiones.
- 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: instalaciones de receptores. Prescripciones generales.
- ITC-BT-47: instalaciones de receptores. Motores.
- ITC-BT-51: instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad.

También se deberá seguir la norma UNE 20 514 1M: reglas de seguridad para aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uso general análogo conectados a una red de energía.

ORDENADOR PERSONAL

El ordenador utilizado para la programación deberá disponer como mínimo de CPU, pantalla, teclado, ratón, conexión Ethernet y USB.

Las características técnicas mínimas serán: 2 procesadores, 2GB de memoria RAM, un espacio en disco de 50GB y un sistema operativo Windows XP o superiores.

PLC

El autómata a utilizar será el M258 de Schneider, modelo TM258LF66DT4LS0 que utiliza el software SoMachine. El PLC se conectará a la red de baja tensión a través de una fuente de alimentación que proporcione al autómata una corriente continua de 24V. Debe cumplir la norma DIN VDE 0160.



VARIADOR DE FRECUENCIA

Según las normas DIN VDE 0160 y la norma NEMA IP21. No se debe destapar la carcasa protectora mientras esté en funcionamiento. Tampoco debe estar conectado a la red mucho tiempo si no está conectado a algún motor.

CONTACTOR

Debe activarse con una señal de 24V recibidos por el autómatas para conectar los diversos motores a la red. Debe ser un elemento robusto y que permita conectar en serie un interruptor automático.

PANTALLA

Debe ser táctil y a ser posible con distintas tonalidades de color de forma que el operario puede interactuar con ella. Debe estar comunicada con el PLC y tendrá un tamaño que permita al operario actuar y visualizarla con facilidad. Debe ser compatible con el autómatas seleccionado.

CAPÍTULO II: CONTROL DEL FUNCIONAMIENTO

AUTOMATIZACIÓN

Se deberá realizar la comprobación de las respuestas de la máquina antes de probar la máquina con la programación terminada para asegurar que va a responder como se espera. En función de esto se realizarán las modificaciones pertinentes.

CONTROL

Se comprobará que el panel de control simula correctamente según lo esperado. Después de comprobar que está todo vinculado y definido correctamente se procederá a realizar las pruebas finales.

PRUEBAS FINALES

Se realizarán pruebas de todas las funciones programadas en la máquina y de las medidas de seguridad, en todo tipo de situaciones y estados del proceso. Una vez hecho esto el proyecto se considerará finalizado.



3. CONDICIONES DE ÍNDOLE FACULTATIVA

CAPÍTULO I: RESPONSABILIDADES DEL CONTRATISTA

Debe actuar según las leyes y normas que rigen su actividad profesional y estar informado. Será una persona capacitada y cualificada para esta labor.

Conocerá las especificaciones técnicas y normas de seguridad aplicables a los elementos del proyecto, cumpliendo en todo momento la normativa establecida en el Reglamento Electrotécnico de Baja Tensión.

Es su responsabilidad comprobar que todos los elementos utilizados en el proyecto cumplen los requisitos establecidos en el mismo. Asimismo, deberá realizar las comprobaciones que se han especificados anteriormente y es su responsabilidad dar el visto bueno para dar el proyecto por finalizado y comenzar con la utilización de la maquinaria.

CAPÍTULO II: RESPONSABILIDADES DEL CONTRATANTE

Facilitará todos los elementos e información al contratista siempre que este lo necesite y estén relacionados con el proyecto en cuestión. Debe entregar también por escrito las especificaciones del proyecto que desea que se lleven a cabo.

La obtención de todos los permisos obligatorios para la realización de este proyecto, quedan completamente a su cargo.

Si existen retrasos causados por motivos totalmente ajenos al contratista, no podrá hacer ninguna reclamación. Los motivos de estos retrasos deben ser justificables. En caso de producirse una demora en el tiempo no justificada se producirán reducciones en el precio del proyecto tal y como se indica en el siguiente apartado.



4. CONDICIONES DE ÍNDOLE ECONÓMICA

CAPÍTULO I: BASE FUNDAMENTAL

El contratista tiene derecho a cobrar por el trabajo realizado siempre y cuando haya cumplido las partes estipuladas en el proyecto.

CAPÍTULO II: PRECIOS Y RECARGOS

El precio de ejecución de este proyecto se establece en el presupuesto.

Si se producen retrasos en el pago desde la finalización del proyecto se establecerán una serie de recargos según se indica a continuación:

- Entre 1 y 15 días: no hay recargo.
- Entre 16 y 30 días: 2% de recargo.
- Entre 31 y 60 días: 5% de recargo.
- Entre 61 y 90 días: 10% de recargo.
- Entre 91 y 150 días: 15% de recargo.
- Entre 151 y 250 días: 25% de recargo.
- Entre 251 y 365 días: 35% de recargo.

Si el contratante no realiza el pago tras 365 días naturales desde la finalización del proyecto, el contratista tiene derecho a realizar una demanda antes los tribunales.

En caso de producirse retrasos en la conclusión del proyecto según los plazos estipulados y sin justificación alguna, el precio se verá reducido de la siguiente forma:

- Entre 1 y 10 días: no hay reducción.
- Entre 11 y 30: 2% de reducción.
- Entre 31 y 60: 5% de reducción.
- Entre 61 y 100: 10% de reducción.
- Entre 101 y 150: 20% de reducción.

Si se diera la circunstancia de no haber finalizado el proyecto después de 150 días respecto al plazo establecido y como se ha dicho anteriormente, sin justificación de motivos ajenos al contratista, se podrá rescindir el contrato por parte del contratante.

En todo momento se habla de días naturales y consecutivos.



5. CONDICIONES DE ÍNDOLE LEGAL

Aquí figuran los documentos y obligaciones legales que tiene cada una de las partes integrantes del proyecto.

CAPÍTULO I: EL CONTRATO

El contrato deberá recoger el precio final del proyecto junto con las cláusulas vistas anteriormente en las condiciones de índole económicas.

Recogerá todas las cláusulas que se negocien entre las partes, se realizará por escrito y se firmará por todas las partes implicadas.

Este contrato une legalmente a ambas partes y las compromete a la realización del proyecto, cada uno realizando su papel dentro de las condiciones acordadas.

CAPÍTULO II: RESCISIÓN DEL CONTRATO

El contrato podrá ser rescindido en el caso de que se den ciertas circunstancias.

Para la rescisión del contrato deberán darse algunas de las condiciones que se nombran a continuación:

- Acuerdo entre ambas partes para rescindirlo.
- Incumplimiento de pago dentro de los plazos.
- Modificaciones de aspectos fundamentales.
- Impago.
- Actuaciones de mala fe.

La rescisión del contrato por cualquiera de estas condiciones será efectuada por la parte que haya salido perjudicada.



ANEXO I



ÍNDICE DEL ANEXO I

1. MANUAL DESDE Máquina	79
2. MANUAL DESDE PANEL	80
3. CONVERSIÓN DE POSICIÓN	81
4. POSICIÓN DE CELDAS	82
5. CASILLA OCUPADA.....	83
6. CASILLA SELECCIONADA OCUPADA.....	84
7. POSICIÓN EJE Z	85
8. ZONA DE SEGURIDAD	86
9. MODO AUTOMÁTICO	87

1. MANUAL DESDE MÁQUINA

```
1  IF (Manual_maq=1) 43
2  THEN 44
3  IF (x_fwd=1 AND x_rev=0) 45
4  THEN 46
5  cx:=1; 47
6  fwd:=1; 48
7  END_IF; 49
8  50
9  IF (x_rev=1 AND x_fwd=0) 51
10 THEN 52
11 cx:=1; 53
12 rev:=1; 54
13 END_IF; 55
14 56
15 IF (y_fwd = 1 AND y_rev=0) 57
16 THEN 58
17 cy:=1; 59
18 fwd:=1; 60
19 END_IF; 61
20 62
21 IF (y_rev=1 AND y_fwd=0) 63
22 THEN 64
23 cy:=1; 65
24 rev:=1; 66
25 END_IF; 67
26 68
27 IF (z_fwd = 1 AND z_rev=0) 69
28 THEN 70
29 cz:=1; 71
30 fwd:=1; 72
31 END_IF; 73
32 74
33 IF (z_rev=1 AND z_fwd=0)
34 THEN
35 cz:=1;
36 rev:=1;
37 END_IF;
38
39 IF(x_fwd=0 AND x_rev=0)
40 THEN
41 cx:=0;
42 END_IF
43
44 IF(y_fwd=0 AND y_rev=0)
45 THEN
46 cy:=0;
47 END_IF
48
49 IF(z_fwd=0 AND z_rev=0)
50 THEN
51 cz:=0;
52 END_IF
53
54 IF (x_fwd=0 AND y_fwd=0 AND z_fwd=0)
55 THEN
56 fwd:=0;
57 END_IF;
58
59 IF (x_rev=0 AND y_rev=0 AND z_rev=0)
60 THEN
61 rev:=0;
62 END_IF;
63
64 IF(subir_vel=1)
65 THEN
66 vel:=1;
67 END_IF;
68
69 IF(bajar_vel=0)
70 THEN
71 vel:=0;
72 END_IF;
```

Este es el programa que se ejecuta mientras está el modo manual desde máquina activado desde la pantalla.

2. MANUAL DESDE PANEL

```
76 IF (Manual_pan=1)
77 THEN
78 IF (xp_fwd=1 AND xp_rev=0)
79 THEN
80 cx:=1;
81 fwd:=1;
82 END_IF;
83
84 IF (xp_rev=1 AND xp_fwd=0)
85 THEN
86 cx:=1;
87 rev:=1;
88 END_IF;
89
90 IF (yp_fwd = 1 AND yp_rev=0)
91 THEN
92 cy:=1;
93 fwd:=1;
94 END_IF;
95
96 IF (yp_rev=1 AND yp_fwd=0)
97 THEN
98 cy:=1;
99 rev:=1;
100 END_IF;
101
102 IF (zp_fwd = 1 AND zp_rev=0)
103 THEN
104 cz:=1;
105 fwd:=1;
106 END_IF;
107
108 IF (zp_rev=1 AND zp_fwd=0)
109 THEN
110 cz:=1;
111 rev:=1;
112 END_IF;
113
114 IF (xp_fwd=0 AND xp_rev=0)
115 THEN
116 cx:=0;
117 END_IF
118
119 IF (yp_fwd=0 AND yp_rev=0)
120 THEN
121 cy:=0;
122 END_IF
123
124 IF (zp_fwd=0 AND zp_rev=0)
125 THEN
126 cz:=0;
127 END_IF
128
129 IF (xp_fwd=0 AND yp_fwd=0 AND zp_fwd=0)
130 THEN
131 fwd:=0;
132 END_IF;
133
134 IF (xp_rev=0 AND yp_rev=0 AND zp_rev=0)
135 THEN
136 rev:=0;
137 END_IF;
138
139 IF (sub_vel=1)
140 THEN
141 vel:=1;
142 END_IF;
143
144 IF (baj_vel=1)
145 THEN
146 vel:=0;
147 END_IF;
148 END_IF;
149
```

Este es el programa que se ejecuta mientras está el modo manual desde panel activado desde la pantalla.

3. CONVERSIÓN DE POSICIÓN

```
150     IF (zero_x=1)
151         THEN
152             pos_x:=0;
153     END_IF;
154
155     IF(zero_x=0)
156         THEN
157             pos_x:=452.999332-0.030431*xanal;
158     END_IF;
159
160     IF (zero_y=1)
161         THEN
162             pos_y:=0;
163     END_IF;
164
165     IF(zero_y=0)
166         THEN
167             pos_y:=406.81973-0.01878*yanal;
168     END_IF;
169
```

Conversión de la señal analógica en valores reales expresados en milímetros para conocer la posición de la máquina durante el proceso.

4. POSICIÓN DE CELDAS

```
177 IF (C11=1) 219 IF (C23=1) 261 IF (C35=1)
178 THEN 220 THEN 262 THEN
179 y_deposito:=0; 221 y_deposito:=140; 263 y_deposito:=275;
180 z_deposito:=0; 222 z_deposito:=75; 264 z_deposito:=150;
181 END_IF; 223 END_IF; 265 END_IF;
182 224 266
183 IF (C12=1) 225 IF (C24=1) 267 IF (C41=1)
184 THEN 226 THEN 268 THEN
185 y_deposito:=70; 227 y_deposito:=205; 269 y_deposito:=0;
186 z_deposito:=0; 228 z_deposito:=75; 270 z_deposito:=220;
187 END_IF; 229 END_IF; 271 END_IF;
188 230 272
189 IF (C13=1) 231 IF (C25=1) 273 IF (C42=1)
190 THEN 232 THEN 274 THEN
191 y_deposito:=140; 233 y_deposito:=275; 275 y_deposito:=70;
192 z_deposito:=0; 234 z_deposito:=75; 276 z_deposito:=220;
193 END_IF; 235 END_IF; 277 END_IF;
194 236 278
195 IF (C14=1) 237 IF (C31=1) 279 IF (C43=1)
196 THEN 238 THEN 280 THEN
197 y_deposito:=205; 239 y_deposito:=0; 281 y_deposito:=140;
198 z_deposito:=0; 240 z_deposito:=150; 282 z_deposito:=220;
199 END_IF; 241 END_IF; 283 END_IF;
200 242 284
201 IF (C15=1) 243 IF (C32=1) 285 IF (C44=1)
202 THEN 244 THEN 286 THEN
203 y_deposito:=275; 245 y_deposito:=70; 287 y_deposito:=205;
204 z_deposito:=0; 246 z_deposito:=150; 288 z_deposito:=220;
205 END_IF; 247 END_IF; 289 END_IF;
206 248 290
207 IF (C21=1) 249 IF (C33=1) 291 IF (C45=1)
208 THEN 250 THEN 292 THEN
209 y_deposito:=0; 251 y_deposito:=140; 293 y_deposito:=275;
210 z_deposito:=75; 252 z_deposito:=150; 294 z_deposito:=220;
211 END_IF; 253 END_IF; 295 END_IF;
212 254 296
213 IF (C22=1) 255 IF (C34=1)
214 THEN 256 THEN
215 y_deposito:=70; 257 y_deposito:=205;
216 z_deposito:=75; 258 z_deposito:=150;
217 END_IF; 259 END_IF;
218 260
```

Este código asigna un valor a las variables de posición del depósito según la celda seleccionada.

5. CASILLA OCUPADA

```

297 IF (C11=1 AND casilla_elegida=1) 337 IF (C24=1 AND casilla_elegida=1) 357 IF (C33=1 AND casilla_elegida=1)
298 THEN 338 THEN 358 THEN
299 OC11:=1; 339 OC24:=1; 359 OC33:=1;
300 END_IF; 340 END_IF; 360 END_IF;
301 341 361
302 IF (C12=1 AND casilla_elegida=1) 342 IF (C25=1 AND casilla_elegida=1) 362 IF (C34=1 AND casilla_elegida=1)
303 THEN 343 THEN 363 THEN
304 OC12:=1; 344 OC25:=1; 364 OC34:=1;
305 END_IF; 345 END_IF; 365 END_IF;
306 346 366
307 IF (C13=1 AND casilla_elegida=1) 347 IF (C31=1 AND casilla_elegida=1) 367 IF (C35=1 AND casilla_elegida=1)
308 THEN 348 THEN 368 THEN
309 OC13:=1; 349 OC31:=1; 369 OC35:=1;
310 END_IF; 350 END_IF; 370 END_IF;
311 351 371
312 IF (C14=1 AND casilla_elegida=1) 352 IF (C32=1 AND casilla_elegida=1) 372 IF (C41=1 AND casilla_elegida=1)
313 THEN 353 THEN 373 THEN
314 OC14:=1; 354 OC32:=1; 374 OC41:=1;
315 END_IF; 355 END_IF; 375 END_IF;
316 356 376
317 IF (C15=1 AND casilla_elegida=1) 357 IF (C33=1 AND casilla_elegida=1) 377 IF (C42=1 AND casilla_elegida=1)
318 THEN 358 THEN 378 THEN
319 OC15:=1; 359 OC33:=1; 379 OC42:=1;
320 END_IF; 360 END_IF; 380 END_IF;
321 361 381
322 IF (C21=1 AND casilla_elegida=1) 362 IF (C34=1 AND casilla_elegida=1) 382 IF (C43=1 AND casilla_elegida=1)
323 THEN 363 THEN 383 THEN
324 OC21:=1; 364 OC34:=1; 384 OC43:=1;
325 END_IF; 365 END_IF; 385 END_IF;
326 366 386
327 IF (C22=1 AND casilla_elegida=1) 367 IF (C35=1 AND casilla_elegida=1) 387 IF (C44=1 AND casilla_elegida=1)
328 THEN 368 THEN 388 THEN
329 OC22:=1; 369 OC35:=1; 389 OC44:=1;
330 END_IF; 370 END_IF; 390 END_IF;
331 371 391
332 IF (C23=1 AND casilla_elegida=1) 372 IF (C41=1 AND casilla_elegida=1) 392 IF (C45=1 AND casilla_elegida=1)
333 THEN 373 THEN 393 THEN
334 OC23:=1; 374 OC41:=1; 394 OC45:=1;
335 END_IF; 375 END_IF; 395 END_IF;
336 376 396

```

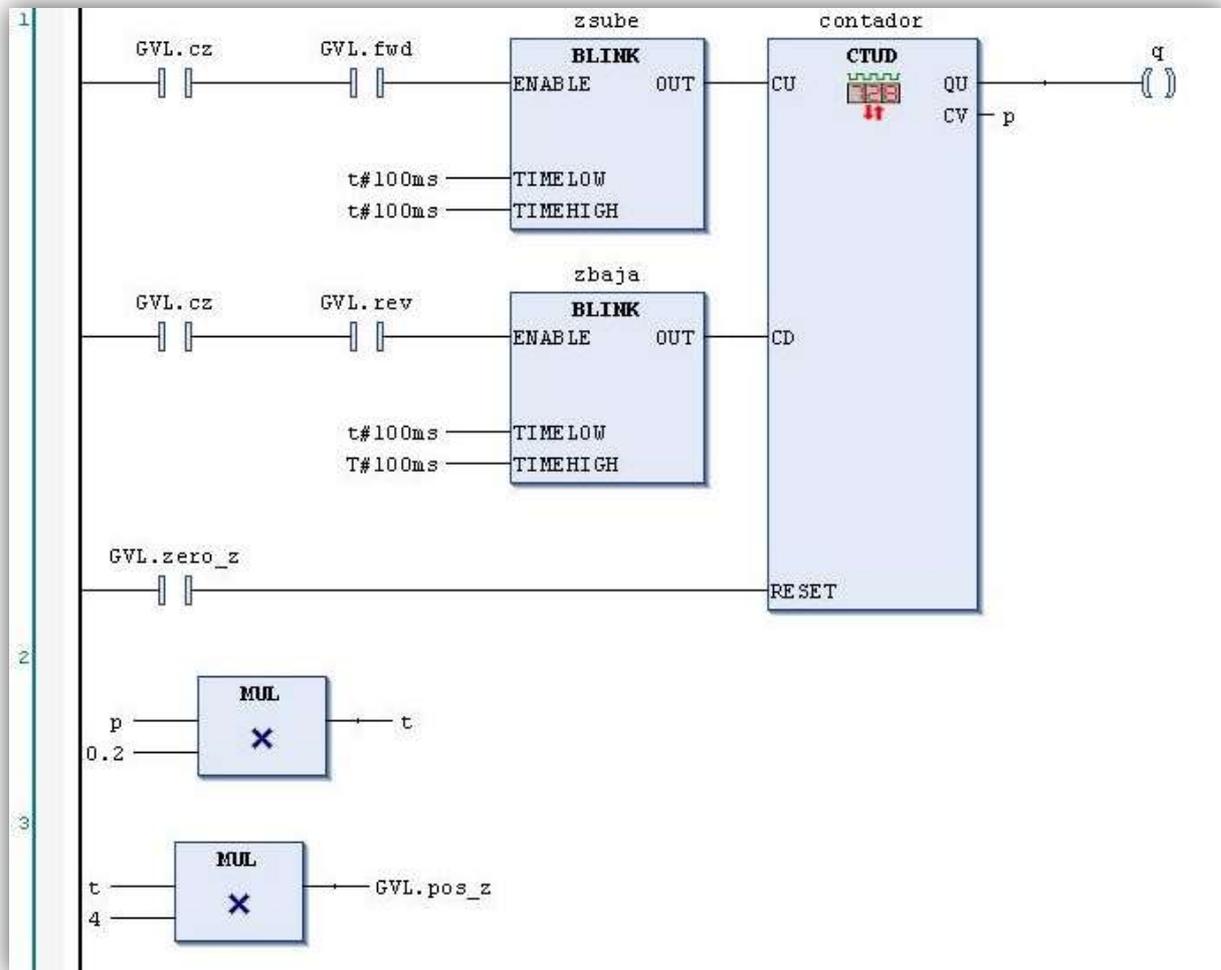
Una vez elegida una casilla e iniciado un proceso, este programa marca la casilla seleccionada como ocupada.

6. CASILLA SELECCIONADA OCUPADA

```
397 IF (C11=1 AND OC11=1) 432 IF (C23=1 AND OC23=1) 462 IF (C34=1 AND OC34=1)
398 THEN 433 THEN 463 THEN
399 ocupado:=1; 434 ocupado:=1; 464 ocupado:=1;
400 END_IF; 435 END_IF; 465 END_IF;
401 436 466
402 IF (C12=1 AND OC12=1) 437 IF (C24=1 AND OC24=1) 467 IF (C35=1 AND OC35=1)
403 THEN 438 THEN 468 THEN
404 ocupado:=1; 439 ocupado:=1; 469 ocupado:=1;
405 END_IF; 440 END_IF; 470 END_IF;
406 441 471
407 IF (C13=1 AND OC13=1) 442 IF (C25=1 AND OC25=1) 472 IF (C41=1 AND OC41=1)
408 THEN 443 THEN 473 THEN
409 ocupado:=1; 444 ocupado:=1; 474 ocupado:=1;
410 END_IF; 445 END_IF; 475 END_IF;
411 446 476
412 IF (C14=1 AND OC14=1) 447 IF (C31=1 AND OC31=1) 477 IF (C42=1 AND OC42=1)
413 THEN 448 THEN 478 THEN
414 ocupado:=1; 449 ocupado:=1; 479 ocupado:=1;
415 END_IF; 450 END_IF; 480 END_IF;
416 451 481
417 IF (C15=1 AND OC15=1) 452 IF (C32=1 AND OC32=1) 482 IF (C43=1 AND OC43=1)
418 THEN 453 THEN 483 THEN
419 ocupado:=1; 454 ocupado:=1; 484 ocupado:=1;
420 END_IF; 455 END_IF; 485 END_IF;
421 456 486
422 IF (C21=1 AND OC21=1) 457 IF (C33=1 AND OC33=1) 487 IF (C44=1 AND OC44=1)
423 THEN 458 THEN 488 THEN
424 ocupado:=1; 459 ocupado:=1; 489 ocupado:=1;
425 END_IF; 460 END_IF; 490 END_IF;
426 461 491
427 IF (C22=1 AND OC22=1) 462 IF (C34=1 AND OC34=1) 492 IF (C45=1 AND OC45=1)
428 THEN 463 THEN 493 THEN
429 ocupado:=1; 464 ocupado:=1; 494 ocupado:=1;
430 END_IF; 465 END_IF; 495 END_IF;
```

En caso de seleccionar una casilla ya ocupada, la variable ocupado se activa y no deja iniciar un nuevo proceso.

7. POSICIÓN EJE Z



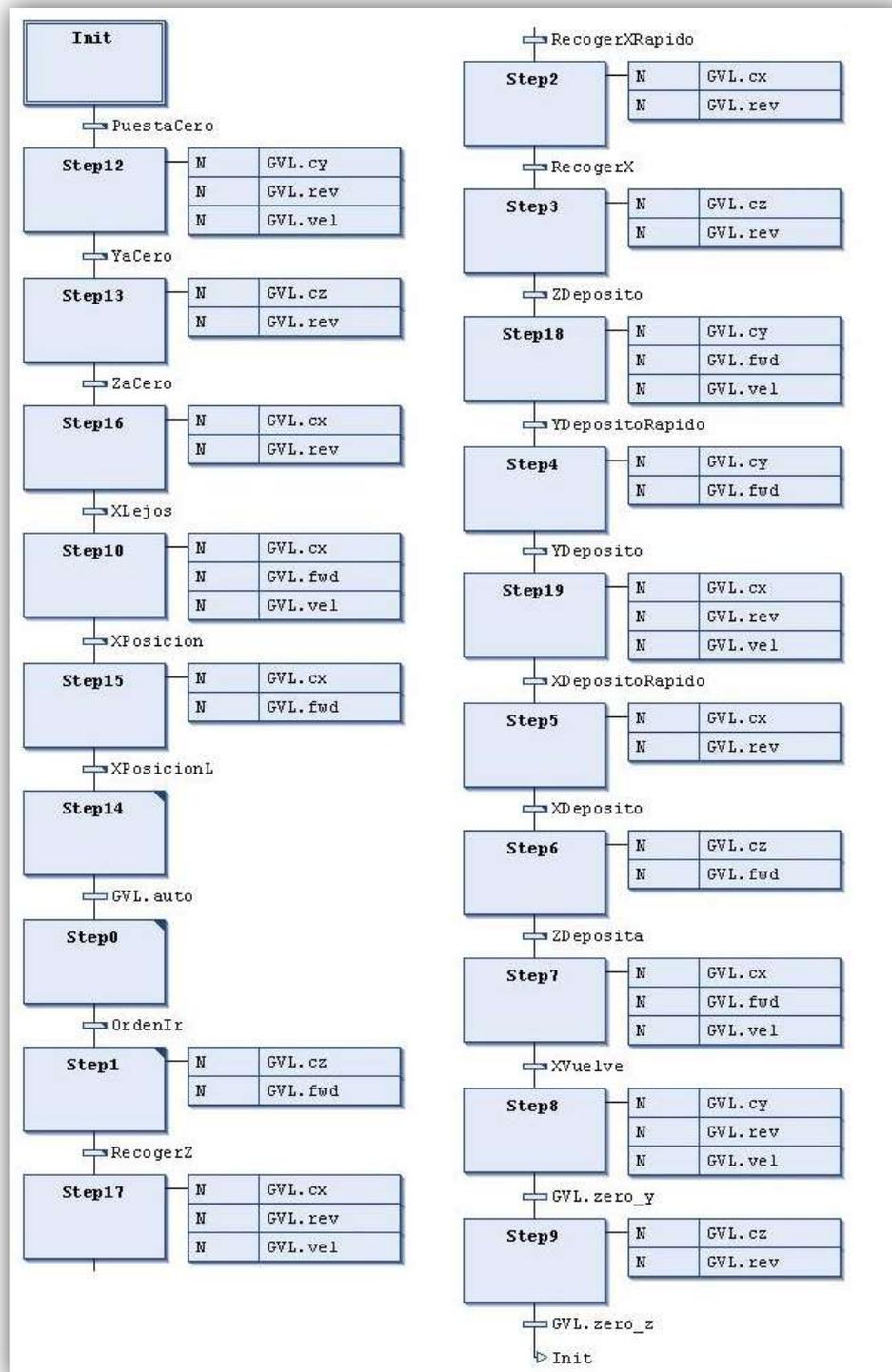
Debido a la falta de potenciómetro en este eje la posición se calcula por el tiempo a través de este programa.

8. ZONA DE SEGURIDAD

```
1  IF (pos_x>410 AND cx=1)    31  IF (pos_y<30 AND cy=1)
2      THEN                  32      THEN
3      fwd:=0;                33      vel:=0;
4  END_IF;                    34  END_IF;
5
6  IF (pos_x>380 AND cx=1)    36  IF (zero_y=1 AND cy=1)
7      THEN                  37      THEN
8      vel:=0;                38      rev:=0;
9  END_IF;                    39  END_IF;
10
11 IF (pos_x<50 AND cx=1)     41  IF (pos_z>255 AND cz=1)
12     THEN                  42     THEN
13     vel:=0;                43     fwd:=0;
14 END_IF;                    44 END_IF;
15
16 IF (zero_x=1 AND cx=1)     46  IF (cz=1)
17     THEN                  47     THEN
18     rev:=0;                48     vel:=0;
19 END_IF;                    49 END_IF;
20
21 IF (pos_y>270 AND cy=1)    51  IF (zero_z=1 AND cz=1)
22     THEN                  52     THEN
23     fwd:=0;                53     rev:=0;
24 END_IF;                    54 END_IF;
25
26 IF (pos_y>240 AND cy=1)
27     THEN
28     vel:=0;
29 END_IF;
```

Este código en texto estructurado no permite que los ejes se muevan más allá de la posición indicada y controla su velocidad en puntos críticos.

9. MODO AUTOMÁTICO



Este es el modo automático programado en SFC. Empieza en el bloque Init y continua hacia abajo hasta pasar a la columna de la derecha y llegar a la última transición donde vuelve a iniciarse el programa.



ANEXO II



ÍNDICE DEL ANEXO II

1. LA MÁQUINA Y SUS COMPONENTES.....	90
2. COMPONENTES DEL PROCESO AS/RS	96
3. AUTÓMATA PROGRAMABLE	99
4. PANTALLA	100
5. ORDENADOR.....	101

1. LA MÁQUINA Y SUS COMPONENTES

MÁQUINA

Se trata de una máquina de tres ejes que tiene la capacidad de desplazarse en las tres coordenadas cartesianas. Su aspecto se muestra en la Figura 55.



Figura 55. Máquina de tres ejes

Cada eje cuenta con un motor asíncrono, un tornillo sinfín encargado de transmitir el movimiento desde el motor a la pieza móvil de la máquina, un potenciómetro que no se encuentra disponible en el eje Z y determina la posición de los ejes X e Y. También cuenta con un sensor inductivo que sirve para dar una posición inicial a la máquina en su puesta a cero y finales de carrera como elemento de seguridad. Estos delimitan el movimiento cortando la corriente cuando son accionados para que en caso de error no se dañe la máquina. Si no existiesen las piezas móviles podrían salirse de los tornillos sinfín dañando el motor.

Esta máquina permite desplazar sus ejes hacia adelante y hacia atrás. Los elementos móviles son los que desplazan las piezas de una zona a otra.

MOTOR

Se trata de un motor asíncrono trifásico que va instalado en el eje de movimiento. La máquina cuenta con tres de ellos y cada uno se encarga de proporcionar movimiento a un eje. El sistema está formado por un reductor, una serie de engranajes y una cadena que transmite el movimiento al tornillo sinfín. En las siguientes imágenes se puede ver el motor y su placa de características técnicas.



Figura 56. Motor asíncrono Parvalux

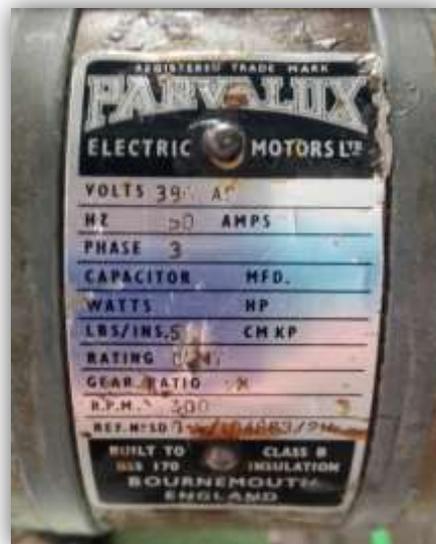


Figura 57. Placa de características

El motor está conectado en estrella con tensión entre fases de 380V, frecuencia de 50Hz y 300 rpm. Los motores pueden ser accionados a la vez, pero no con una velocidad distinta cada uno ni un sentido de giro contrario.

SENSOR INDUCTIVO

Cada eje cuenta con un sensor inductivo que se encarga de detectar objetos de metal. Este funcionamiento se basa en la generación de un campo magnético a través de una bobina. Cuando un objeto metálico se acerca se producen sobre este corrientes de Foucault que se inducen a causa del campo magnético. Esto reduce una reducción de la energía del circuito y provoca una reducción de la amplitud de onda del campo y de las ondas del circuito. Esta disminución de amplitud emite una señal que avisa de la presencia del objeto metálico. Este sensor se utiliza en este proyecto para identificar la posición inicial, información necesaria para realizar la puesta a cero cada vez que se inicie un modo o se encienda la máquina.



Figura 58. Sensor Inductivo

FINAL DE CARRERA

Un interruptor final de carrera es un elemento de seguridad. Impide que siga el desplazamiento de las partes móviles de la máquina, protegen la máquina de posibles averías y evita accidentes. Como se ha dicho anteriormente hay dos finales de carrera por eje y cortan la alimentación en caso de ser accionados.



Figura 59. Final de Carrera

BOTONERA

Contiene los interruptores necesarios para realizar el control desde la máquina además de un pulsador de emergencia. Al pulsarlo detiene de forma automática los motores con el fin de evitar cualquier situación de peligro para los propios motores o para algún usuario.



Figura 60. Botonera

POTENCIOMETRO

Un potenciómetro es un sensor de desplazamiento de tipo resistivo. Contiene una resistencia de valor fijo y un cursor deslizante sobre ella. El cursor o el potenciómetro deben estar en la parte móvil de la máquina de manera que al moverse la máquina hay un desplazamiento relativo entre ambos. Esto hace variar la tensión o la intensidad que da el potenciómetro. La señal proporcionada es analógica.



Figura 61. Potenciómetro

En este caso se cuenta con un potenciómetro que varía la tensión y otro que varía la intensidad, por lo que esta opción debe configurarse en voltios para el canal del eje X y en amperios para el canal del eje Y, ya que en caso contrario el autómat no puede leer las señales analógicas enviadas por la máquina.

A continuación se muestra la ficha de características de cada potenciómetro.



Figura 62. Potenciómetro Eje X



Figura 63. Potenciómetro Eje Y

CONTACTORES

Los contactores son elementos electromecánicos que conectan o desconectan el circuito. Pueden ser accionados a distancia. Desde la maniobra cuando se da tensión a la bobina esta conecta el circuito de potencia. En la máquina del proyecto se tienen tres contactores, uno para cada motor.



Figura 64. Contactores

VARIADOR DE FRECUENCIA

El variador de frecuencia es un sistema capaz de variar la frecuencia de una corriente alterna con lo cual se puede utilizar para controlar la velocidad de un motor controlando su frecuencia de alimentación. La máquina cuenta con un variador de la marca Siemens como se muestra en la siguiente la Figura 65.



Figura 65. Variador de frecuencia

2. COMPONENTES DEL PROCESO AS/RS

HORQUILLA

Debido a que la máquina no tenía ningún elemento para coger las piezas, se diseñó una horquilla con dos imanes en su base para ser acoplada a la máquina. Estos imanes proporcionan la resistencia necesaria para levantar las piezas.



Figura 66. Horquilla

PIEZAS

Las piezas están hechas de corcho en forma de U invertida para favorecer su desplazamiento por parte de la horquilla. Su forma y tamaño favorece a aumentar el error admisible. Es muy similar al proceso de recogida de pales en las industrias.



Figura 67. Pieza

ZONA DE RECOGIDA

Esta será la zona donde sea recogida la pieza para llevarla a las casillas de depósito. En este caso cada vez se coloca manualmente una pieza, en la industria sería llevada de diversas formas como cintas transportadoras o por algún operario. Debido a la limitación del eje Y, que su rango de movimiento cubre solamente la anchura de la zona de depósito, no se puede colocar la zona de recogida a los lados del casillero ya que la máquina no podría llegar ahí. Como el eje X barre toda la zona en la cual se mueve tampoco se puede colocar delante del casillero, por lo tanto la única opción es utilizar un brazo que sitúa la pieza delante de la zona de depósito y luego puede apartarse para que no choque con el eje X. La zona de recogida se muestra en la



Figura 68. Zona de Recogida

ZONA DE DEPÓSITO

Se trata de un casillero con 20 celdas hecho de corcho que representa el almacén. Sus medidas son 340x80x300mm. Las celdas están provistas de soportes laterales que sustentan la piezas y tienen un tope que facilita el enchanche de las piezas. Las medidas de cada celda son 65x80x75 mm.



Figura 69. Depósito

3. AUTÓMATA PROGRAMABLE

Toda la programación realizada desde el ordenador se descarga al PLC, que es el encargado de controlar el funcionamiento de la máquina siguiendo las instrucciones programadas. Los autómatas controlan procesos secuenciales, barriendo estas secuencias de arriba hacia abajo muchas veces en un corto periodo de tiempo.

Los PLC reciben señales de entrada que le llegan desde los sensores instalados en el proceso y emiten señales de salida como respuesta. Ambas señales pueden ser digitales o analógicas.

Las señales digitales transmiten valores TRUE o FALSE, pudiendo transmitir valores de gran tamaño en un rango establecido por los bits asignados a cada señal. Suelen disponer de 8 o 16 bits.

Los PLC permiten la comunicación con el operario, de esta forma se permite el control humano sobre los procesos. Este control puede realizarse a nivel local desde la propia máquina o desde paneles de control táctiles, que permiten la interconexión con el autómata y el control centralizado de estos.

En este proyecto se ha utilizado el autómata TM258LF66DT4LS0 de Schneider Electric, con resolución analógica de 16 bits.

El autómata se muestra en la Figura 70.

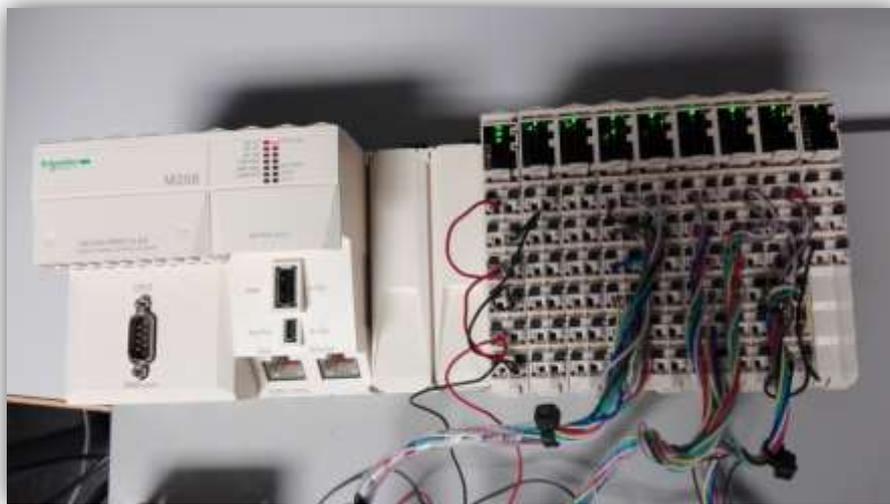


Figura 70. PLC de Schneider

4. PANTALLA

La pantalla utilizada como panel de control ha sido la Magelis XBTGT2110. Con una resolución de 320x240 pixel, calidad QVGA de 5'7 pulgadas alimentada a 24V.

Se configura con el VijeoDesigner, los colores disponibles en la pantalla son el azul, el blanco y 16 tonalidad de gris. Cuenta con un procesador de 133MHz.

Esta pantalla puede conectarse mediante USB o ModBus. En este caso en primer lugar se ha conectado a través del puerto USB con el ordenador, ya que la primera descarga debe realizar mediante USB. Posteriormente se ha transferido la información mediante ModBus con el autómata, pasando siempre la información a través del PLC.

Se trata de una pantalla analógica de 18W que requiere una temperatura ambiente para su funcionamiento entre 0°C y 50°C. Sus medidas son 167'5x135x59'5 mm.



Figura 71. Pantalla Magelis



5. ORDENADOR

El ordenador utilizado tiene un sistema operativo Windows 7 de 64 bits, con un procesador Intel® Core(TM) i3-4150 CPU 3.50GHz y 4 GB de memoria RAM. Este ordenador se encuentra en el laboratorio del DIE.

Se ha utilizado una máquina virtual con el Windows XP instalado que cuenta con el programa SoMachine versión 3.0.14.5 de Schneider Electric que es la única versión que contiene en su base de datos el autómata de este proyecto, el modelo TM258LF66DT4LS0. Por lo tanto cualquier otra versión no servía ya que era imposible conectar con el autómata.

Para abrir esta máquina virtual se ha utilizado el software VMware Workstation 12 Player. Utilizar una máquina virtual no es la mejor manera de trabajo ya que consume mucha memoria RAM del ordenador y lo puede llegar a ralentizar de forma significativa. Pero en algunas ocasiones es el único recurso disponible.