



Universidad
Zaragoza

Trabajo Fin de Grado

Diseño e implementación de un sistema de control de movimiento preciso y sincronizado para una aplicación industrial mediante un controlador, una pantalla y un servo.

Autor

Iván Garbayo Hernández

Director

Daniel Juvillar Lasus

Ponente

Claudio Carretero Chamarro

Titulación del autor

Ingeniería Electrónica y Automática

ESCUELA DE INGENIERÍA
Y ARQUITECTURA 2023

Trabajo Fin de Grado



Universidad
Zaragoza

Iván Garbayo Hernández

Resumen

Este Trabajo Fin de Grado implica el diseño y desarrollo de un sistema de control de movimiento preciso y sincronizado para una aplicación industrial. En este caso, se simula una máquina lijadora, pero podemos pensar en todo tipo de máquina herramienta, o similar, en la que haya un movimiento rectilíneo mediante uno o incluso mediante varios servomotores.

En este trabajo se utilizará un autómatas Modicon M241 de Schneider Electric, que informa del estado de la máquina y controla las señales eléctricas que actúan sobre el servomotor de la serie Lexium de Schneider Electric.

De manera similar, la serie Magelis de Schneider Electric introduce una terminal interactiva hombre-máquina a través de la cual el operador podrá controlar la máquina de forma automática o manual y ajustar parámetros. El terminal también se podrá utilizar para visualizar errores, guiar a los operarios a corregirlos, mostrar datos de los diferentes parámetros, etc.

En el PLC se registrarán los principales eventos de la máquina: fallos, posiciones, velocidades, tiempos, etc. Además, se proporcionará acceso remoto a la máquina vía servidor web para su mando y supervisión.

Con la ayuda del software disponible, Machine Expert para el PLC, Vijeo Designer para el terminal HMI y el conocimiento de automatización adquirido académicamente, el proyecto resultará en mejoras adicionales y una mayor versatilidad, lo que llevará a reducir costos de producción y mantenimiento.

Agradecimientos

En primer lugar, me gustaría agradecer a todas las personas que me han apoyado a lo largo de mi carrera, especialmente a mis padres y mi hermano.

También quisiera agradecer a mis maestros y mentores, en especial a Daniel Juvillar Lasus y a Claudio Carretero, quienes me brindaron experiencias durante mi etapa escolar y mi todavía corta etapa laboral. Gracias por compartir sus conocimientos, por apoyar este trabajo y por estar conmigo en el proceso de investigación.

Quisiera agradecer a mis amigos y compañeros universitarios por su colaboración y apoyo constante. Vivimos momentos juntos que nunca olvidaremos.

Muchísimas gracias a todas las personas mencionadas, les agradezco todo vuestro esfuerzo de corazón.

ACRÓNIMOS

FTP (Protocolo de Transferencia de Archivos).

QVGA (Resolución gráfica de 324 x 240 píxeles).

EPROM (ROM programable borrable).

MCB (disyuntor en miniatura).

DIN (Instituto Alemán de Normalización).

IEE (Instituto de Ingenieros Eléctricos y Electrónicos).

CSMA/CD (Acceso Múltiple con Detección de Colisiones).

UTC (Tiempo Universal Coordinado).

PLC (Controlador Lógico Programable).

VCC o VDC (Voltaje de corriente directa).

VCA (Voltaje de corriente alterna).

UE (Unión Europea).

CEE (Comunidad Económica Europea).

ISO (Organización Internacional de Normalización).

SMPS9 (Fuente de alimentación de modo conmutado).

TCP (Protocolo de Control de Transmisión).

CANopen (Protocolo de comunicaciones basado en objetos).

IP (Protocolo de Internet).

Modbus (Protocolo de comunicación abierto).

PROFIBUS (Bus de campo de proceso).

REFERENCIAS

- [1] Pantalla HMI de Schneider Electric.
- [2] PLC de Schneider Electric.
- [3] Servodrive de Schneider Electric.
- [4] Servomotor de Schneider Electric.
- [5] Módulo para la pantalla de Schneider Electric.
- [6] Fuente de alimentación del Trabajo Fin de Grado de Schneider Electric.
- [7] Fuente de alimentación equivalente de Schneider Electric.
- [8] Interruptor automático de Schneider Electric.
- [9] Router inalámbrico.
- [10] Conmutador TCP/IP del Trabajo Fin de Grado de Schneider Electric.
- [11] Conmutador TCP/IP equivalente de Schneider Electric.

INDICE

1. Introducción.....	11
1.1. Contexto y justificación	11
1.2. Objetivo y alcance del Trabajo de Fin de Grado	12
1.3. Metodología empleada.....	13
2. Descripción de la máquina	15
2.1. Componentes principales	15
2.1.1. Servo.....	16
2.1.2. Autómata programable	21
2.1.3. Interfaz hombre-máquina (HMI).....	23
2.2. Componentes secundarios.....	24
2.2.1. Fuente de Alimentación	24
2.2.2. Interruptor automático	25
2.2.3. Router inalámbrico	26
2.2.4. Conmutador TCP/IP Ethernet.....	27
3. Comunicación de los componentes de la máquina.....	28
4. Arquitectura del sistema	30
5. Solución alternativa	31
6. Programación del control.....	34
6.1. Secuencia principal.....	34
6.2. Estados de la secuencia principal	35
6.2.1. Estado inicial	35
6.2.2. Estado modo selección	35
6.2.3. Estado automático	36
6.2.4. Estado manual	37
6.3. Averías y avisos	37
6.4. Reloj	39
7. Programación del HMI	41
7.1. Pantalla de Inicio	41
7.2. Pantalla modo de operación.....	42
7.3. Pantalla ajustes manuales	42
7.4. Pantalla ajustes automáticos	43

7.5. Pantalla de supervisión del servomotor	44
8. Visualizador Web	47
9. Legislación	48
10. Presupuesto	49
10.1. Mano de obra	49
10.2. Hardware	49
10.3. Coste total	51
11. Conclusiones	52
12. Bibliografía	53
13. Anexos	55
13.1. Machine Expert	55
13.1.1. Variables de la secuencia principal	55
13.1.2. Variables de avisos y alarmas	56
13.1.3. Programación de lectura del reloj	56
13.1.4. Variable de consigue hora	57
13.1.5. Variables del reloj	58
13.2. Vijeo Designer	59
13.2.1. Script reloj	59
13.2.2. Declaración del grupo de alarmas	59
13.2.3. Declaración de registro de datos	59
13.2.4. Declaración de variables	60
13.2.5. Implementación en gráficas de supervisión del servomotor	61

ILUSTRACIONES

Figura 1: Máquina lijadora.	15
Figura 2: Servomotor BSH0551P01A1A.	16
Figura 3: Curva par/velocidad del servomotor.	16
Figura 4: Servodrive LXM32MU45M2.	17
Figura 5: Lazo cerrado del servomotor.	18
Figura 6: Corrientes de fase del servomotor.	19
Figura 7: Conmutación del servomotor.	20
Figura 8: Controlador TM241CEC24R.	21
Figura 9: Pantalla táctil HMISTU855.	23
Figura 10: Fuente de alimentación ABL8MEM24012.	24
Figura 11: Interruptor automático A9N21554.	25
Figura 12: Router inalámbrico TL-WR802.	26
Figura 13: Conmutador TCP/IP TCSESU053FN0.	27
Figura 14: Comunicación de dispositivos.	28
Figura 15: Arquitectura del proyecto.	30
Figura 16: Pirámide de las redes de comunicación.	31
Figura 17: Arquitectura de la solución alternativa.	32
Figura 18: Secuencia principal.	34
Figura 19: Estado inicial.	35
Figura 20: Estado del modo de selección.	36
Figura 21: Estado del modo automático.	36
Figura 22: Estado del modo manual.	37
Figura 23: Averías y avisos.	38
Figura 24: Contador de tiempo en tensión.	39
Figura 25: Reloj.	40
Figura 26: Pantalla de Inicio.	41
Figura 27: Pantalla de modo de operación.	42
Figura 28: Pantalla de ajustes manuales.	43
Figura 29: Pantalla de ajustes automáticos.	44
Figura 30: Pantalla de supervisión del servomotor.	45
Figura 31: Pantalla de registro de alarmas.	46
Figura 32: Pantalla del visualizador web.	47
Figura 33: Precios del hardware.	50



TABLAS

TABLA 1: PRECIO DE LA MANO DE OBRA	49
TABLA 2: PRECIO TOTAL	51

1. Introducción

En la era de la automatización industrial, la electrónica y la automatización juegan un papel importante en la optimización de procesos de fabricación. El control de movimiento preciso y sincronizado es un requisito previo para aumentar la eficiencia, la productividad y la calidad en muchas aplicaciones industriales.

Este Trabajo Fin de Grado aborda el emocionante desafío de diseñar e implementar un sistema de control de movimiento de última generación, utilizando los tres componentes principales suministrados por Schneider Electric: la pantalla HMISTU855 [1], el controlador TM241CEC24R [2] y el servo LX32MU45M2 [3] + BSH0551P01A1A [4].

1.1. Contexto y justificación

La selección de estos componentes se basa en la experiencia adquirida durante mis prácticas extracurriculares en Araíz Suministros Eléctricos en Zaragoza, donde he podido tener acceso a estos dispositivos de Schneider Electric. Esta prestigiosa marca es conocida mundialmente como líder en soluciones de automatización industrial y sus productos son ampliamente utilizados en una variedad de aplicaciones industriales.

La pantalla HMISTU855 presenta una interfaz intuitiva y fácil de usar que facilita la visualización de información en tiempo real y el control de los procesos industriales. Esta pantalla ofrece una experiencia de usuario muy mejorada con la capacidad de mostrar gráficos y alarmas personalizables.

El controlador TM241CEC24R, por otro lado, sirve como una plataforma versátil para controlar procesos industriales. Cuenta con un potente poder de cómputo, amplias opciones de conectividad y herramientas de programación avanzadas. Este PLC proporciona una base sólida para implementar soluciones de automatización altamente eficientes y flexibles.

El servodrive LX32MU45M2 junto con el servomotor BSH0551P01A1A complementan esta configuración y brindan un control de movimiento dinámico y de alta precisión. Con sus rápidas respuestas y alta capacidad de carga, este servo asegura un rendimiento óptimo en aplicaciones que requieren movimientos controlados y sincronizados con precisión milimétrica.

1.2. Objetivo y alcance del Trabajo de Fin de Grado

El objetivo de este Trabajo Fin de Grado consiste en desarrollar un sistema de control de movimiento utilizando los componentes nombrados anteriormente, cumpliendo con los más altos estándares de precisión y sincronización. Al utilizar las funciones avanzadas de estos tres componentes principales (pantalla, controlador y servo) se espera lograr excelentes niveles de rendimiento.

Con una visión clara de los resultados esperados, dentro del alcance de este Trabajo Fin de Grado se establecen los siguientes puntos:

- Adentrarse en un análisis completo de las características técnicas y las posibilidades innovadoras ofrecidas por la pantalla, el controlador y el servo. Esto incluye estudiar las especificaciones técnicas, los protocolos de comunicación y las herramientas de programación de estos componentes para aprender cómo funcionan y alcanzar su máximo potencial.
- Diseñar una arquitectura inteligente y eficaz que englobe todo el sistema de control de movimiento, teniendo en cuenta las necesidades específicas de la aplicación industrial. En este apartado se incluye una cuidadosa selección de los mejores componentes, una configuración óptima de las conexiones y una integración precisa de cada elemento para garantizar un funcionamiento fluido y sin problemas.
- Desarrollar algoritmos de control de movimiento que puedan lograr niveles elevados de precisión y sincronización en la aplicación industrial específica. Todo ello, utilizando habilidades de programación avanzadas, con un ajuste del PLC y el servo para ofrecer un movimiento coordinado y controlado perfectamente para las necesidades específicas de la aplicación.
- Realizar pruebas exhaustivas para verificar y validar la funcionalidad y el rendimiento del sistema de control de movimiento. Estas pruebas de rendimiento y precisión detalladas nos permitirán evaluar minuciosamente la eficiencia y garantizar que los objetivos que establecemos se cumplan o superen.
- Documentar y analizar detalladamente los resultados obtenidos durante el desarrollo e implementación del sistema de control de movimiento. Este paso puede servir para identificar posibles mejoras y ajustes para implementaciones futuras, lo que permitirá maximizar el potencial de los sistemas innovadores.

Lograr estos objetivos dará como resultado sistemas de control de movimiento sincronizados, precisos y de última generación que se adapten perfectamente a las necesidades de aplicaciones industriales específicas.

1.3. Metodología empleada

La ejecución de este Trabajo Fin de Grado requiere de una metodología rigurosa y bien estructurada para lograr resultados confiables y satisfactorios. Las metodologías utilizadas para llevar a cabo el desarrollo y lograr el objetivo y alcance establecidos se describen a continuación.

- Investigación y análisis inicial: En esta fase inicial, se compilaron y analizaron las especificaciones técnicas pertinentes, las capacidades funcionales y los protocolos de comunicación de la pantalla, el servo y el PLC. Además, se realizó una revisión de la literatura y se consultaron fuentes de información autorizadas y actualizadas para comprender las últimas tendencias y mejores prácticas en el campo del control de movimiento.
- Diseño del sistema: Se procedió al diseño detallado del sistema de control de movimiento en base al conocimiento obtenido en la etapa de investigación. Se determinaron los requisitos específicos de la aplicación industrial prevista y se tuvieron en cuenta las características y limitaciones de los componentes suministrados. Se realizó un análisis exhaustivo de las conexiones entre pantalla, controlador y servo para definir las conexiones físicas y los protocolos de comunicación correspondientes para garantizar una integración fluida y eficiente.
- Desarrollo del software: En esta etapa se desarrolló el software necesario para controlar y programar el sistema de control de movimiento. Usamos el programa Machine Expert de Schneider Electric para programar el controlador y aprovechar sus capacidades para desarrollar la aplicación de automatización industrial que, como he dicho con anterioridad, es la simulación de una máquina lijadora. Los parámetros requeridos y los algoritmos de control están configurados para aprovechar al máximo la capacidad del servo mientras proporcionan un movimiento preciso y sincronizado del motor.
- Implementación y pruebas: Después de completar el desarrollo del software, comenzó la implementación física del sistema de control de movimiento. Se realizaron pruebas para verificar el correcto funcionamiento de cada componente y evaluar la precisión del sistema en general. Se han realizado ajustes y optimizaciones cuando fueron necesarios para garantizar un rendimiento óptimo.
- Análisis de resultados: Los resultados obtenidos durante las pruebas fueron analizados y comparados con criterios de éxito previamente establecidos. Se evaluó la precisión del control de movimiento, la sincronización entre los componentes y la capacidad del sistema para cumplir con los requisitos de la aplicación industrial prevista.



- Documentación y conclusiones: Finalmente, se documentaron todos los pasos del desarrollo e implementación del sistema de control de movimiento, quedando así este documento. Este documento está destinado a ser una referencia futura y pretende ser una contribución al conocimiento en los campos de ingeniería electrónica y automática.

2. Descripción de la máquina

El ciclo de trabajo de la máquina lijadora ha de ser el siguiente:

- La herramienta ha de ir a una posición uno impuesta por el usuario, mediante una velocidad, también impuesta por el usuario. En esa posición, la herramienta hará su trabajo durante un determinado tiempo, también modificable por el usuario.
- Una vez acabado el trabajo en esa posición, hará lo mismo para una segunda posición. Una vez acabado este ciclo volverá a comenzar otro marchando a la posición uno y así sucesivamente.

Aquí muestro la máquina lijadora (figura 1):



Figura 1: Máquina lijadora.

Para conseguir este control de movimiento preciso y sincronizado, se han utilizado los tres componentes principales ya mencionados y otros componentes secundarios, que, sin ellos, no hubiera sido posible el resultado final.

2.1. Componentes principales

Como ya he mencionado con anterioridad, los componentes principales de este Trabajo Fin de Grado son la pantalla, el servo y el controlador.

2.1.1. Servo

Como ya he comentado, el servo está formado por el servomotor y por el servodrive.

El servomotor que vamos a utilizar es un motor de la familia BSH, en particular el BSH0551P01A1A, que aparece en la figura 2. Estos motores se utilizan como servomotores porque tienen la mejor densidad de potencia de su clase para cumplir con los requisitos de máquinas pequeñas como estas.



Figura 2: Servomotor BSH0551P01A1A.

Posee un par continuo en parada de $0.5 \text{ N}\cdot\text{m}$ para una velocidad máxima de 9000 rpm. La inercia promedio del servomotor proporciona una mayor inercia de carga por tamaño de motor y mayor ganancia para cumplir con los requisitos de carga resistiva. El servomotor tiene un perfil de curva de par/velocidad similar al ejemplo que se muestra en la figura 3.

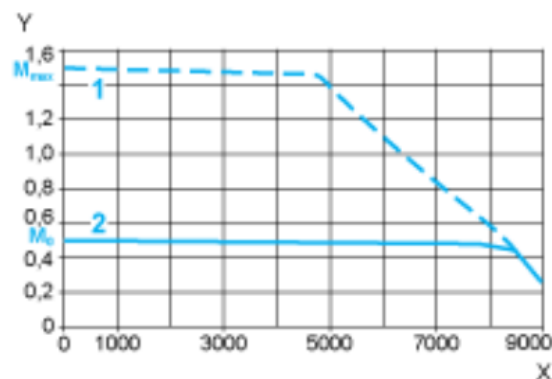


Figura 3: Curva par/velocidad del servomotor.

En donde, X es la velocidad en rpm e Y es el par en $N \cdot m$ 1 es el par de pico y 2 es el par continuo. M_{max} ($N \cdot m$) representa el valor de par de pico en parada, mientras que M_0 ($N \cdot m$) representa el valor de par continuo en parada.

El servodrive que vamos a utilizar es de la familia Lexium 32, en particular el modelo LXM32MU45M2, el cuál aparece en la figura 4. Los Lexium 32 vienen de serie con entradas y salidas lógicas y analógicas que se pueden configurar de forma óptima para su aplicación. También cuenta con una interfaz de control que facilita el acceso a los diferentes niveles de la arquitectura.



Figura 4: Servodrive LXM32MU45M2.

Tiene una interfaz de control que opera a través de un tren de pulsos. Cuenta con un puerto combinado CANopen/CANmotion para mejorar el rendimiento del sistema de control. También se puede conectar a las principales redes de comunicación y buses a través de varias tarjetas de comunicación.

Presenta disponibles los siguientes protocolos: DeviceNet, EtherNet/IP y PROFIBUS DP V1.

La serie Lexium 32 forma parte del sistema de seguridad del sistema de control, ya que cuenta con una función de desconexión de par segura (STO) que evita el reinicio accidental del servomotor.

Esta función corresponde al nivel SIL2 de la norma IEC/EN 61508 para instalaciones eléctricas y la norma IEC/EN 61800-1 para sistemas de accionamiento de potencia.

Un servomotor es básicamente un tipo de motor eléctrico de alta potencia cuyo sistema convierte la electricidad en un movimiento controlado con precisión. El servomotor se considera esclavo del servodrive porque comunican y ejecutan comandos (posición, velocidad, par) sobre su eje.

Por lo tanto, el funcionamiento óptimo de los servomotores se basa en esta comunicación bidireccional persistente, que también verifica si el movimiento es el deseado.

El servodrive es un dispositivo inteligente que interpreta las señales capturadas por el controlador. Luego interactúa con el servomotor para lograr el movimiento deseado de manera altamente precisa y repetible.

Un servo es un actuador giratorio o lineal que proporciona funciones que los motores normales no pueden, como el control preciso de la posición angular, la aceleración y la velocidad de un eje. Básicamente, consta de un motor eléctrico con un sistema de retroalimentación integrado (encoder) que funciona como un sistema de lazo cerrado, como se muestra en la figura 5.

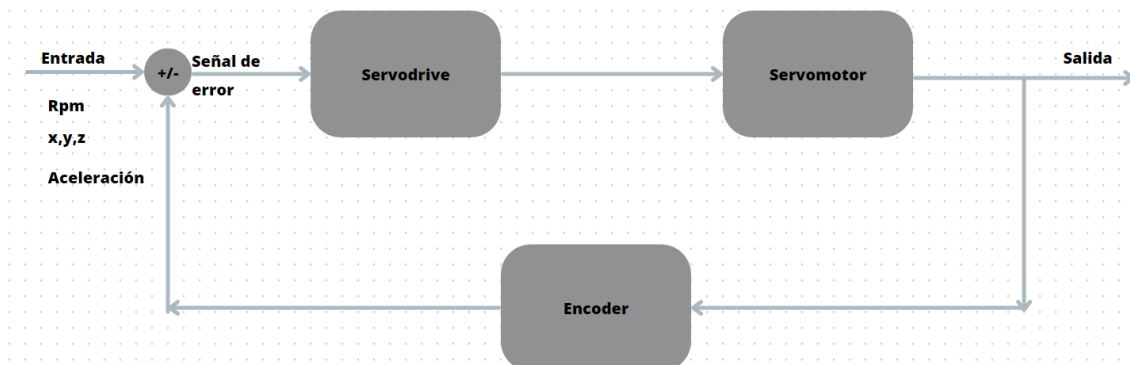


Figura 5: Lazo cerrado del servomotor.

Las señales se procesan en el servodrive, que actúa como fuente al servomotor y este obedece la señal del servodrive traduciéndola en velocidad, aceleración y posición. En la salida, se va a leer la señal para compararla con la señal de entrada, y si hay una diferencia entre estas señales, se genera una ecuación de error que es procesada por el servodrive para corregir e igualar la señal de salida con la de entrada.

La función de la retroalimentación (encoder) es informarle al servodrive la posición mecánica del rotor para ver qué bobinas hay que alimentar del estator y conseguir así la interacción magnética.

La alimentación del servodrive que acompaña al servomotor es trifásica. Hay que tener en cuenta que una fuente de alimentación trifásica cuenta con tres corrientes monofásicas de la misma frecuencia y amplitud, pero que se alternan para alcanzar un valor máximo. Estas corrientes mantienen una diferencia de fase equilibrada de 120 grados entre ellas como aparece en la figura 6.

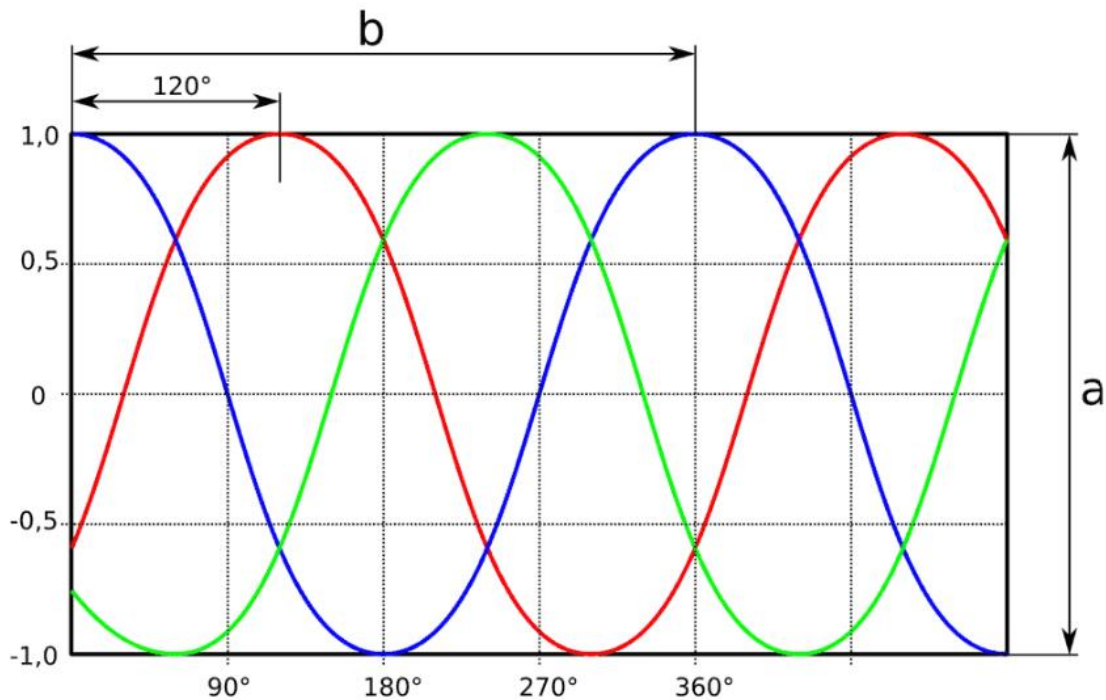


Figura 6: Corrientes de fase del servomotor.

Según los principios anteriores, la generación de movimiento por un dispositivo trifásico requiere cambios entre fases para alimentar correctamente los servomotores. Este proceso de cambio entre fases individuales se llama conmutación. Para conseguir el movimiento deseado, el servodrive decide qué fases deben activarse y los momentos exactos en los que deben activarse.

Por lo tanto, un cambio en la tensión asegura la continuidad del movimiento y por ende el par proporcionado por el servomotor depende directamente de este cambio de corriente.

La conmutación está directamente relacionada con la configuración interna de un servomotor, ya que la dirección del campo magnético en las bobinas se invierte cuando cambia la corriente, y la interacción de este campo magnético con el rotor crea movimiento.

Los servomotores no contienen segmentos de conmutación que cambien la corriente entre los devanados para producir un campo magnético giratorio. Más bien, necesitamos una señal generada por un codificador o, por lo general, un sensor de efecto Hall.

Estas señales proporcionan al servodrive información sobre la posición del rotor, que el controlador utiliza para alterar las corrientes que fluyen a través de los devanados para producir un campo magnético giratorio. En la figura 7 se muestra la conmutación del servomotor.

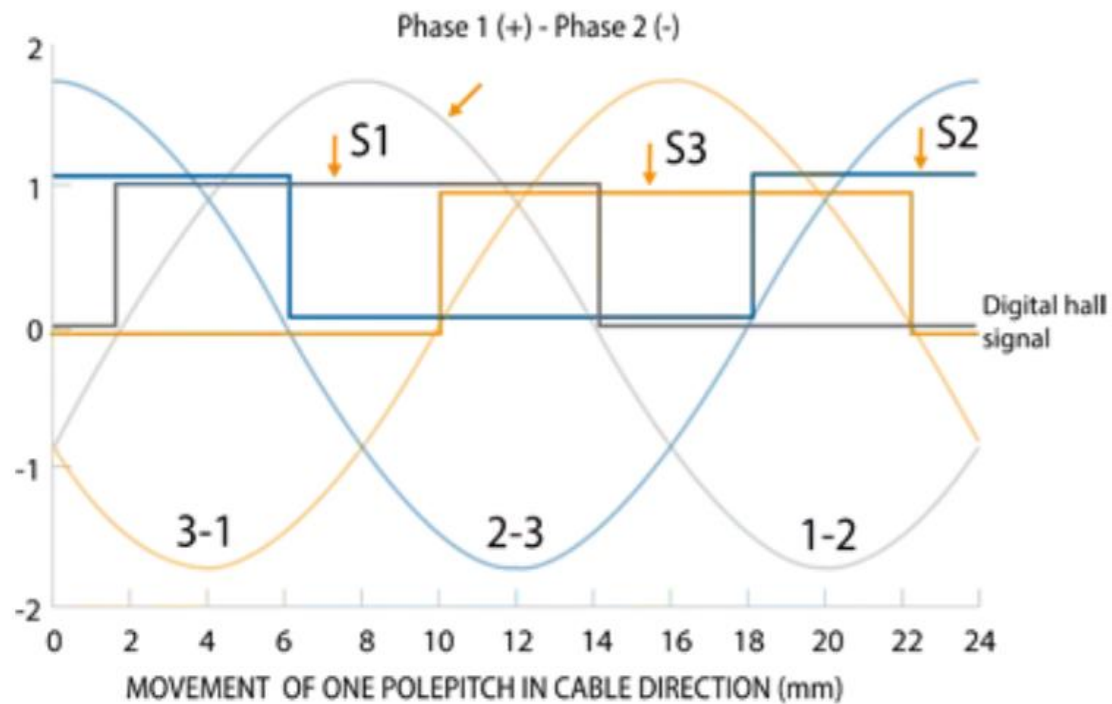


Figura 7: Conmutación del servomotor.

En otras palabras, para finalizar, los servomotores no tienen interruptores mecánicos, por lo que se requieren dispositivos electrónicos de conmutación y retroalimentación para controlar el movimiento.

2.1.2. Autómata programable

Los controladores lógicos Modicon M241 incluyen funciones de control de movimiento y velocidad. Este proyecto en particular es TM241CEC24R, el cuál aparece en la figura 8. Posee un puerto Ethernet integrado que proporciona servicios de servidor web y FTP.



Figura 8: Controlador TM241CEC24R.

Las funciones integradas del controlador son las siguientes:

- Puerto serie Modbus.
- puerto USB para programación.
- bus de campo CANopen para arquitecturas distribuidas.
- Funciones avanzadas de control de movimiento (contador de alta velocidad y salida de tren de pulsos para control de servomotor).

Todos los controladores M241 tienen un interruptor Run/Stop y también tienen una ranura para tarjeta de memoria SD. El controlador lógico M241 contiene hasta cinco puertos de comunicación: Ethernet con función servidor web, CANopen, dos puertos serie y un puerto USB de programación.

Machine Expert es una solución de software para que los fabricantes de máquinas construyan, configuren y pongan en marcha una máquina completa que incluya lógica, control de movimiento, HMI y automatización en un único entorno de software, dentro de la red. Ofrece numerosas funciones de mantenimiento y supervisión, incluidos todos los idiomas IEC 61131-3, configurador de campo, depuración y diagnóstico experto y visualización web.

Los lenguajes de programación que se pueden utilizar en ellos son:

- IL (lista de instrucciones).
- LD (diagrama de contactos).
- SFC (grafcet).
- ST (texto estructurado).
- FBD o CFC (bloques de función).

2.1.3. Interfaz hombre-máquina (HMI)

Los operadores juegan un papel importante en la función de interacción hombre-máquina. Con base en los datos obtenidos, se deben tomar medidas para garantizar el correcto funcionamiento de las máquinas y equipos sin comprometer la seguridad o la disponibilidad. Por ello, es importante que la calidad del diseño de las interfaces y la interactividad asegure que los operadores puedan actuar con seguridad en todo momento.

Los paneles Magelis, en este caso la referencia HMISTU855, el cuál aparece en la figura 9 y un módulo trasero HMIS5T [5], cuentan con tecnologías de información y comunicación optimizadas, tales como:

- Comunicación de alto nivel (Ethernet incorporado, Multilink, Web y servidor FTP, correo electrónico).
- Almacenamiento de datos externo (tarjeta de memoria SD y memoria USB) para guardar datos de producción y protección de aplicaciones.
- Administración de periféricos, impresoras, lectores de código de barras, etc.



Figura 9: Pantalla táctil HMISTU855.

El modelo de la pantalla es el HMISTU855, que posee un tipo de pantalla a color QVGA TFT, una resolución de pantalla de 5,7 pulgadas y 320 x240 píxeles QVGA, una aplicación de 96 MB Flash EPROM, tarjeta SD de 4GB ampliable, conexión asíncrona de serie: RS 232C (COM1) y RS 485 (COM2) y puertos USB (1 puerto host A y un puerto mini B).

2.2. Componentes secundarios

En esta sección vamos a definir los componentes secundarios, que, aunque los llamemos de esta manera, son igual de importantes para conseguir el funcionamiento de este proyecto.

2.2.1. Fuente de Alimentación

Estamos ante una fuente de alimentación cuya referencia es ABL8MEM24012 [6] (figura 10), que en estos momentos está obsoleta y su referencia equivalente es la ABLM1A24012 [7], que posee las mismas características. Esta fuente de alimentación controlada por modo conmutado (SMPS9), proporciona alimentación de corriente continua. Posee una entrada de 100-240 VCA, una corriente de salida de 1,2 A a 24 VCC y una potencia nominal de 30 W. Es ideal para proporcionar la potencia de corriente continua requerida en los equipos de automatización utilizados en aplicaciones industriales.



Figura 10: Fuente de alimentación ABL8MEM24012.

Donde más se utilizan son para aplicaciones comerciales y residenciales, cintas transportadoras, embalaje y control, alimentos y bebidas, etc. Además, al tener un ancho de 54 mm, se puede montar fácilmente en cualquier sitio de un cuadro eléctrico o similar.

2.2.2. Interruptor automático

Este producto cuya referencia es A9N21554 [8] (figura 11), es un disyuntor (MCB) de bajo voltaje. Se trata de un interruptor automático de polo más neutro, corriente nominal de 3ª y una curva de disparo C. Nos va a permitir proteger nuestro sistema contra posibles sobrecargas o cortocircuitos. El poder de corte en cortocircuito aumenta a 10 kA a 230 VAC de acuerdo con la norma IEC 60947-2.



Figura 11: Interruptor automático A9N21554.

Este interruptor cumple con el modelo industrial EN/IEC 60898-1 y el modelo residencial EN/IEC 60947-2. Tiene una frecuencia de operación de 50 Hz o 60 Hz y debido a su pequeño tamaño se puede montar en un riel DIN para una instalación modular.

2.2.3. Router inalámbrico

Este aparato cuya referencia es TL-WR802 [9] (figura 12), nos va a proporcionar Wi-Fi, posee una tasa de transferencia de 300 Mbps y es ideal para casa y para viajar debido a su tamaño de bolsillo.



Figura 12: Router inalámbrico TL-WR802.

En nuestro caso va a ser de vital importancia para que el usuario pueda observar el servidor web desde cualquier lugar en el que pueda estar conectado a la Wi-Fi proporcionada por este aparato. Es ideal para que en una empresa no tengan que ir a la máquina en cuestión para saber si está funcionando correctamente o no. De esta manera, el usuario podrá saber el estado de la máquina desde su despacho o lugar de trabajo.

2.2.4. Conmutador TCP/IP Ethernet

Este producto forma parte de la familia ConneXium de dispositivos de conectividad Ethernet. Este conmutador TCP/IP que muestro en la figura 13, tiene cinco puertos para conexiones de cable de cobre. Su referencia es TCSESU053FN [10], pero se encuentra obsoleta y su referencia equivalente es MCSESU053FN0 [11].



Figura 13: Conmutador TCP/IP TCSESU053FN0.

Se trata de un conmutador de red con una tensión de alimentación de 24 V y un consumo de energía de 2.2 W. Este producto ofrece una forma inteligente y flexible de integrar soluciones Ethernet para cualquier empresa. En este caso, nos va a permitir tener todos nuestros componentes del proyecto interrelacionados entre sí, para que se puedan comunicar unos con otros.

3. Comunicación de los componentes de la máquina

La comunicación entre los componentes de esta máquina es la comunicación Ethernet, también conocida como el estándar IEEE 802.3. Es un estándar de transmisión de datos para redes de área local basado en el principio de que todos los dispositivos en una red Ethernet están conectados a la misma línea de comunicación por cable circular (figura 14).

Todos los dispositivos en una red Ethernet están conectados a la misma línea de transmisión y se comunican a través de un protocolo llamado CSMA/CD. Esto significa que es un protocolo de acceso de monitoreo de detección de portador y detección de colisión.



Figura 14: Comunicación de dispositivos.

Este protocolo permite que cada dispositivo transmita a través del cable en cualquier momento sin priorización entre dispositivos. Esta comunicación se realiza de forma sencilla y antes de enviarla, cada equipo se asegura de que no haya comunicación en la línea.

Se produce una colisión cuando dos dispositivos transmiten al mismo tiempo, es decir, cuando hay varias tramas de datos en el cable al mismo tiempo. En ese caso, los dos equipos dejan de comunicarse y esperan una cantidad de tiempo aleatoria. Puede enviarla tan pronto como su primer envío exceda el límite de tiempo.

Este principio se basa en algunas restricciones, como que los paquetes de datos deben tener un tamaño máximo y que debe de haber un tiempo de espera requerido entre dos envíos. El tiempo de espera depende de la frecuencia de las colisiones.

La topología está organizada en forma de estrella alrededor de un conmutador. Examina las direcciones de origen y destino de los mensajes y crea una tabla que muestra qué computadoras están conectadas a qué puertos de conmutación. Si el conmutador reconoce el puerto receptor, el mensaje solo se enviará al puerto apropiado, dejando los otros puertos libres para otras transmisiones que puedan ocurrir al mismo tiempo.

4. Arquitectura del sistema

La arquitectura se basa en el PLC M241 que se comunica con los elementos de campo como son el servodrive Lexium 32 y el terminal HMI de Magelis a través de la red Ethernet. Esta red tiene un punto de conexión, que también está conectado a Ethernet, creando un acceso Wi-Fi, lo que permite acceder a la máquina desde una tableta o teléfono inteligente. Esta arquitectura se muestra en la figura 15.



Figura 15: Arquitectura del proyecto.

También se puede acceder al servidor web personalizado del PLC desde un PC utilizando un navegador estándar, o si conectamos el ordenador al conmutador mediante un cable Ethernet.

5. Solución alternativa

Hay varios factores a considerar al elegir el tipo de red industrial a utilizar en el proyecto. Una primera aproximación es tomar los dos enfoques principales siguientes:

- Cantidad de información enviada
- Tiempo de respuesta deseado

Esto permite el posicionamiento de las redes líderes en el mercado las cuales podemos observar en la figura 16.

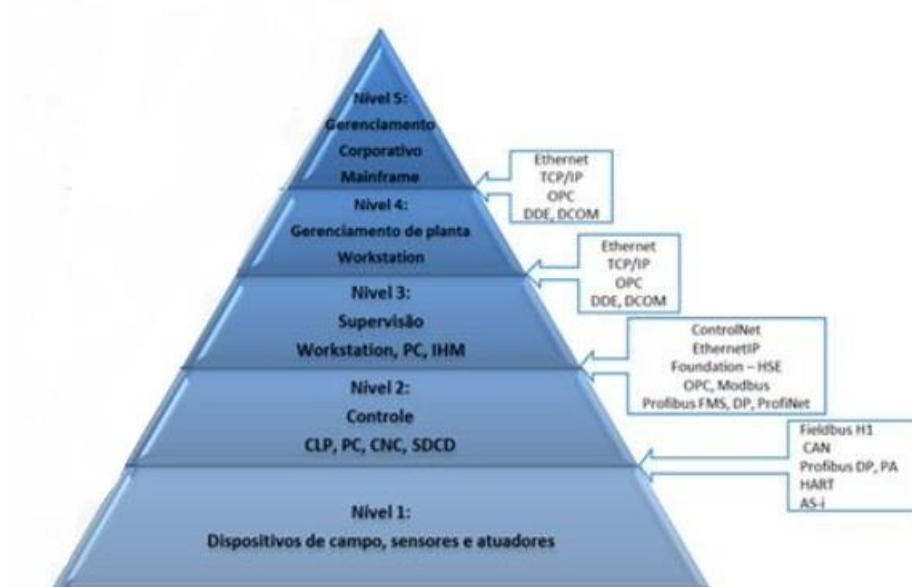


Figura 16: Pirámide de las redes de comunicación.

Como ya he nombrado, para este Trabajo Fin de Grado he utilizado una comunicación completamente Ethernet. Y es que, con el predominio de Ethernet en los negocios e Internet, esta red se ha convertido en un estándar de comunicación.

Ayuda a reducir los costos de conexión y mejorar el rendimiento, la confiabilidad y la funcionalidad. Su velocidad no ralentiza sus aplicaciones y su arquitectura facilita la modernización. Los productos y el software son compatibles, por lo que el sistema se caracteriza por su durabilidad.

Por estas razones, es muy utilizado, especialmente en los campos de negocios, la ofimática e Internet, y se prefiere a otras opciones debido a sus claras ventajas sobre las redes propietarias.

No obstante, otra solución totalmente válida y eficiente podría haber sido realizar una comunicación CANopen para del servodrive Lexium 32, quedando una estructura tal y como se muestra en la figura 17.



Figura 17: Arquitectura de la solución alternativa.

CANopen es una comunicación de bus de campo diseñada para aplicaciones automotrices. Durante muchos años, esta red ha demostrado ser flexible y confiable en diversas aplicaciones, como equipos médicos, trenes, ascensores, diversas máquinas y plantas industriales.

CANopen está integrado como base para los servodrive Lexium 32, lo que ofrece ahorros económicos con respecto a otros buses de campo y es la principal razón por la cual esta alternativa también sería una buena idea. Además, el bus es fácil de instalar y no requiere de conexiones equipotenciales entre dispositivos, como es el caso de muchos buses de campo.

Además del ahorro de utilizar este bus, también tiene estas ventajas:

- Permite que las máquinas y los sistemas funcionen con precisión incluso en atmósferas muy perturbadas debido a su alta inmunidad a las interferencias electromagnéticas.
- Cuando se transmiten datos, el sistema genera un mensaje y automáticamente toma en cuenta su prioridad. Se elimina la pérdida de telegramas debido a problemas de colisión y se evita la pérdida de tiempo de espera hasta que la red vuelva a estar inactiva.
- Permite una transmisión de datos sin interrupciones. Esta es una de las razones por las que CANopen se usa en dispositivos médicos que requieren redes altamente confiables.
- Perder el tiempo es también perder el dinero, por lo que está diseñado para minimizar estas pérdidas. Posee una buena detección de errores y buenos mecanismos de corrección. Es por eso, que es una de las redes más confiables para máquinas y sistemas.

Las principales razones para utilizar este tipo de red son su rendimiento y flexibilidad para adaptarse exactamente a las necesidades de su aplicación.

Finalmente he elegido la conexión mediante una red de comunicación completa mediante Ethernet, debido a que para aplicaciones que requieren una alta precisión, sincronización y que incorporan control, Ethernet es una mejor opción que CANopen.

6. Programación del control

La programación de la máquina lijadora se ha llevado a cabo mediante el programa Machine Expert de Schneider Electric. La secuencia principal del modo de trabajo de la máquina se ha programado en lenguaje grafcet (SFC). Cada estado de esta secuencia principal de trabajo se ha programado en lenguaje de diagramas de bloques funcionales (FBD). Además, también se ha programado una serie de avisos y averías, estas en lenguaje de diagrama de contactos (LD).

Como complemento adicional, también se ha programado un reloj para que el usuario pueda observar la fecha y la hora y que cuando se active alguna alarma o aviso, pueda ver así, en qué momento exacto ha ocurrido. Cabe destacar que esta parte ha sido totalmente opcional, puesto que no cumple ninguna función relevante en el sistema.

6.1. Secuencia principal

La secuencia principal (figura 18) realiza el control de los modos de maniobra de la máquina. Cuenta con un estado inicial, un estado de modo de selección de operación y los estados de trabajo de la máquina, los cuales son el automático y el manual.

Al pulsar comenzar en el control de mando, la máquina entraría en el modo de selección. A continuación, al pulsar automático o manual, entraría en esos modos de operación respectivamente. Y si pulsamos exit_manual o exit_automático dependiendo en el modo donde la máquina esté trabajando, volveríamos al estado de modo de selección.

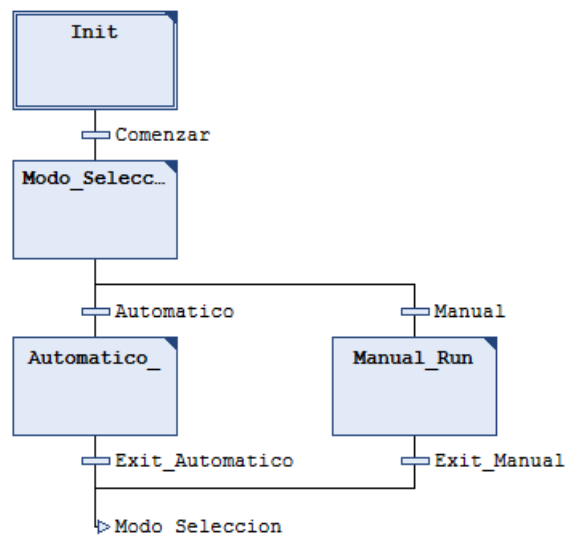


Figura 18: Secuencia principal.

6.2. Estados de la secuencia principal

Como he comentado, cada estado de la secuencia principal de la máquina posee sus propias funciones operativas, todas ellas programadas en lenguaje de diagrama de bloques funcionales (FBD).

6.2.1. Estado inicial

En el estado inicial (figura 19) encontramos el bloque de potencia, que se encuentra a 0, un bloque de reset y cuatro bloques que nos van a permitir observar por pantalla tanto la velocidad como la posición actual de nuestro servomotor.

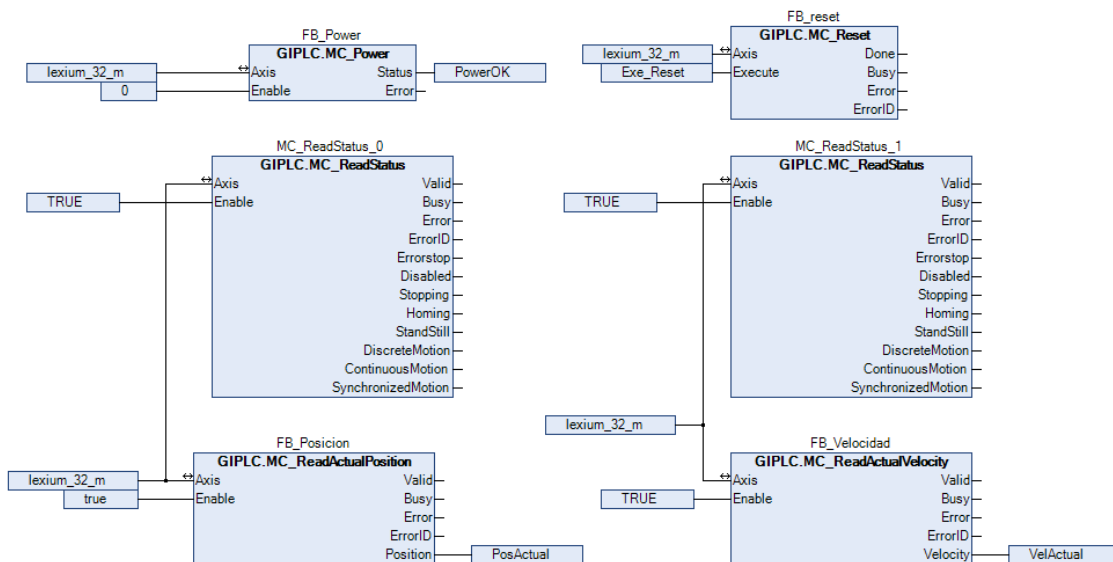


Figura 19: Estado inicial.

6.2.2. Estado modo selección

En el estado del modo de selección (figura 20) tenemos además de los bloques anteriores un bloque de homing para que cada vez que estemos en este modo podamos mandar a nuestra máquina a una posición inicial. Además de ello, podemos observar que en este estado ya sí que va a haber tensión en nuestro servomotor para que así la máquina pueda realizar el homing.

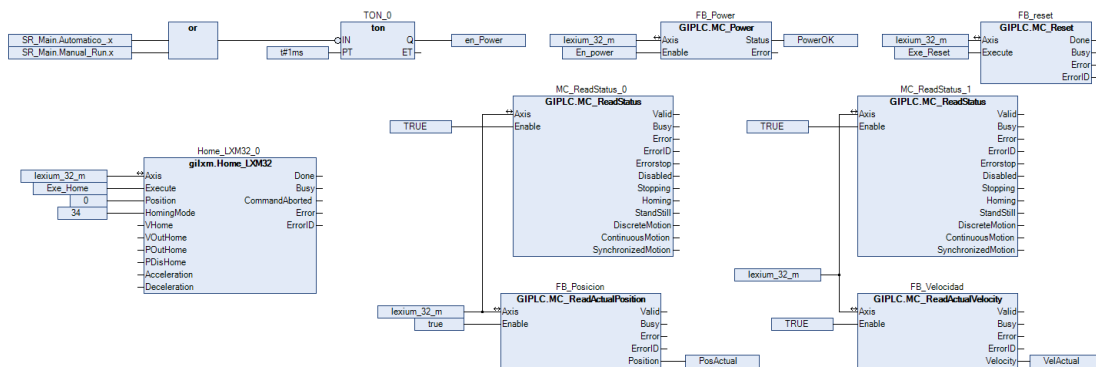


Figura 20: Estado del modo de selección.

6.2.3. Estado automático

En el estado automático (figura 21) tenemos la automatización de nuestra máquina para cuando esté trabajando de manera permanente. Tenemos dos bloques de movimiento absoluto, que poseen como entrada tanto la posición como la velocidad a la que queremos que vaya a esa posición. Entre medio de estos bloques tenemos unos bloques de tiempo que son el tiempo que queremos que nuestra máquina lijadora esté trabajando en esa posición. También podemos apreciar un bloque AND, añadido para conseguir que al negar la señal de salida del segundo bloque de tiempo y con la señal de inicio activada, el proceso automático no pare y pueda continuar.

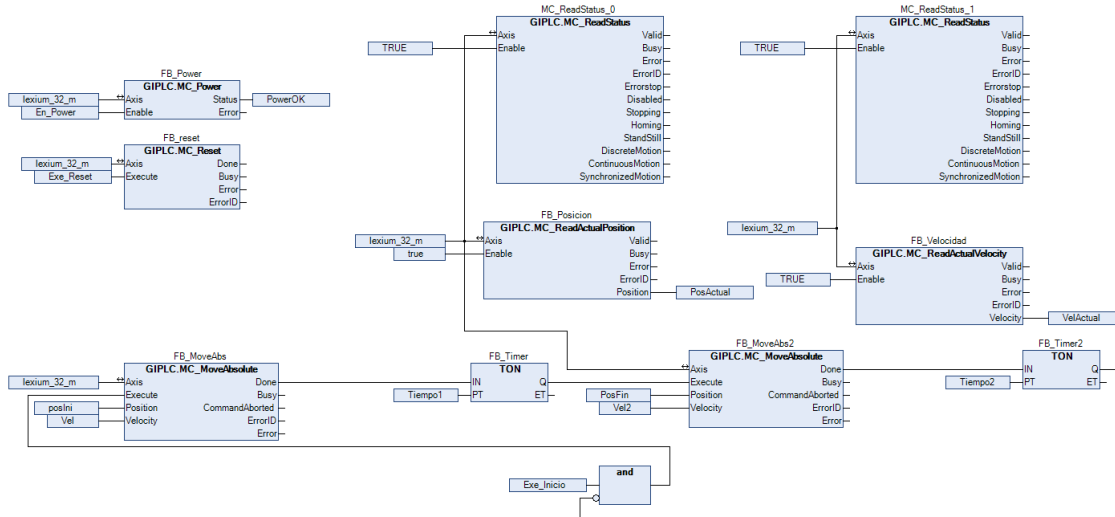


Figura 21: Estado del modo automático.

6.2.4. Estado manual

En el estado manual (figura 22) podemos apreciar que se encuentran dos bloques de movimiento relativo. Ambos tienen como entrada la misma velocidad y una distancia opuesta, para que así a través de uno de los bloques podamos avanzar y con el otro podamos retroceder.

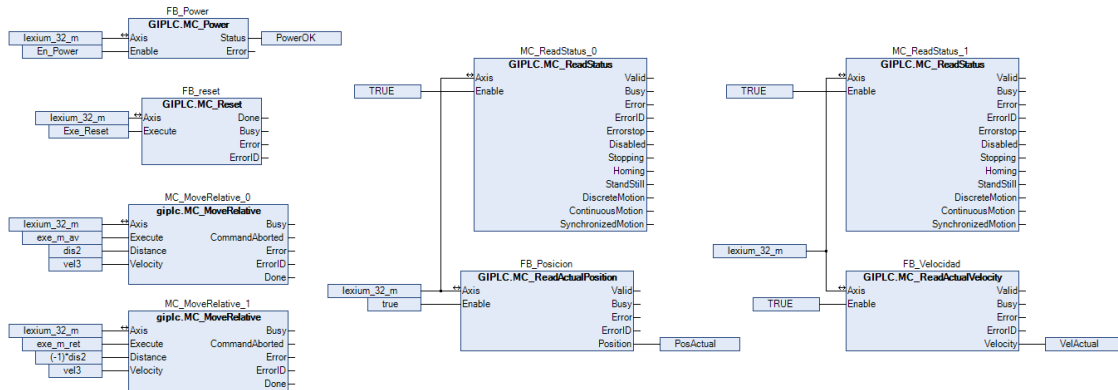


Figura 22: Estado del modo manual.

6.3. Averías y avisos

En cuanto a este apartado, he programado una serie de averías y avisos según la posición y la velocidad a la que está trabajando el servomotor de nuestra máquina por temas de seguridad, como muestro en la figura 23.

Cuando la velocidad sea superior a 3000 mm/s o inferior a -3000 mm/s tendremos un aviso y cuando sea superior a 3500 mm/s o inferior a -3500 mm/s tendremos una avería. Por otro lado, cuando la posición sea superior a 90000 mm tendremos un aviso y cuando sea superior a 95000 tendremos una avería.

Además, también podemos apreciar cómo he programado un reset de las mismas para poder reiniciarlas desde la pantalla y que cuando ocurran las averías la máquina vaya al estado inicial y por lo cual deje de estar en tensión.

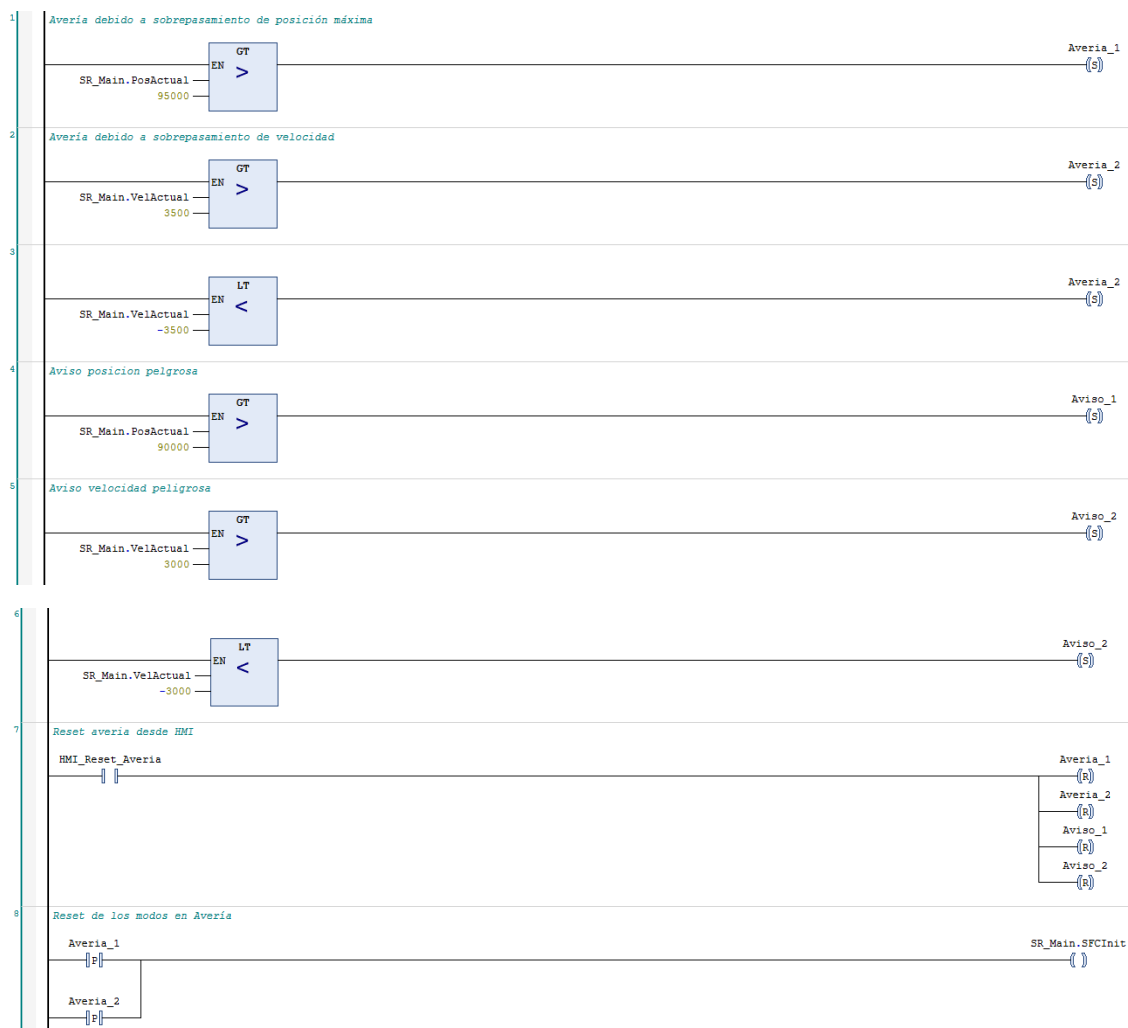


Figura 23: Averías y avisos.

Además, también he añadido un contador que nos mostrará el tiempo que lleva en tensión nuestra máquina, como muestro en la figura 24.

Mediante el bloque blink generamos una señal con un flanco ascendente de medio segundo y uno descendente de otro medio segundo, regulada por el contacto de tensión de la máquina. Estos flancos ascendentes ocurren cada segundo, activando la función ADD y agregándolos a la variable Maq_seg, que especifica la cantidad de segundos que la máquina ha estado funcionando.

Después lo que se hace es comparar estos segundos para que cada vez que lleguen a 60 ponga los segundos a 0 y sumar uno a la variable Maq_min. Este proceso se repite para obtener las horas.

Al final he añadido un reseteo del contador para poder reiniciarlo desde la pantalla.

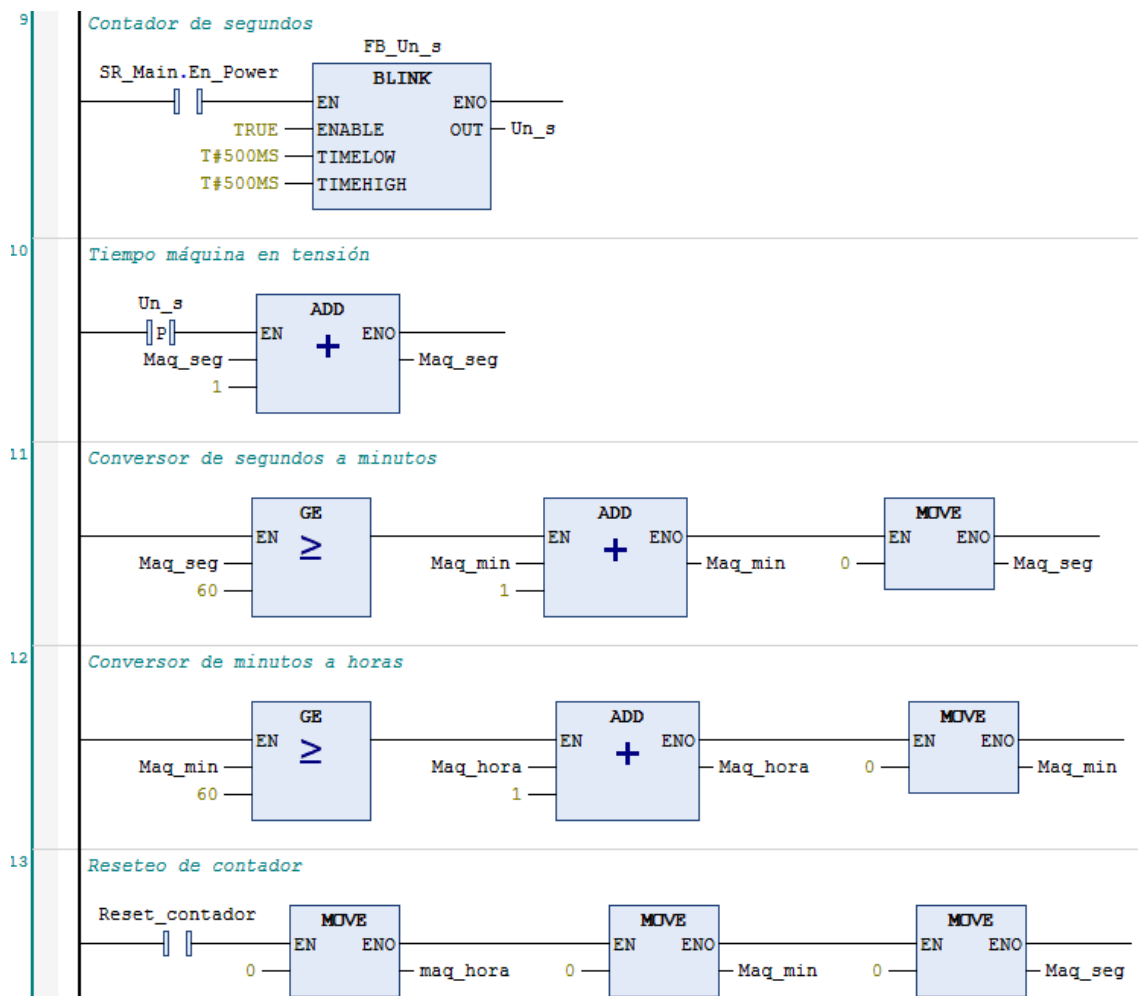


Figura 24: Contador de tiempo en tensión.

6.4. Reloj

Como ya he mencionado, he añadido un reloj de manera adicional (figura 25), puesto que no cumple ninguna función relevante en el sistema automático, pero que he visto necesario para que el usuario pueda ver en todo momento la fecha y la hora y que además cuando haya alguna avería o aviso se pueda saber el momento exacto en el que ha ocurrido.

Para ello en Machine Expert se programa una conversión de tiempo del PLC que está en formato UTC, que es un principal estándar de tiempo obtenido a partir del Tiempo Atómico Internacional por el cual el mundo regula los relojes y el tiempo, para sacar el año, mes, día, hora, minutos, segundos.

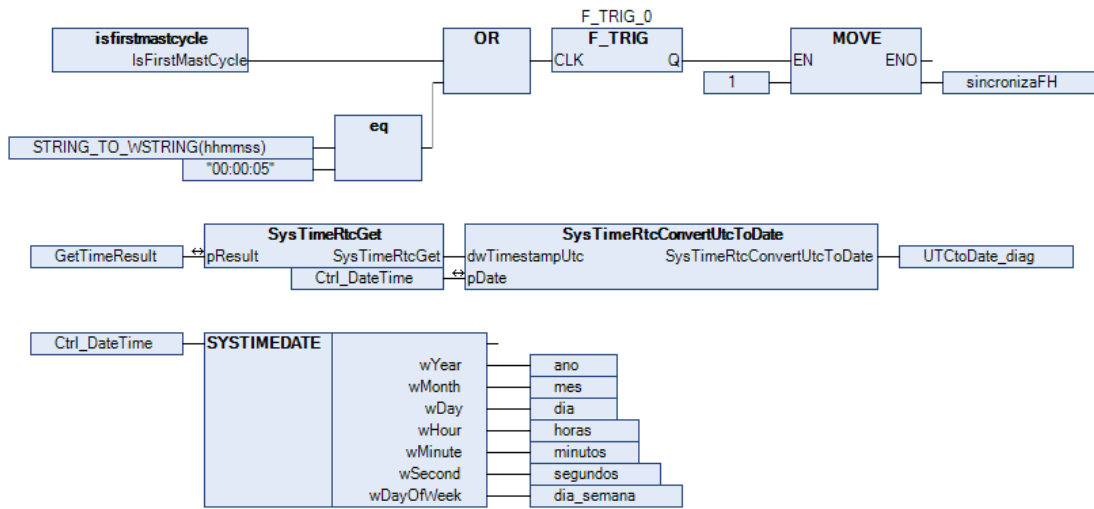


Figura 25: Reloj.

7. Programación del HMI

HMI es la interfaz que establece la conexión entre el operador y la máquina, diseñada para ser fácil e intuitiva de usar.

Consta de 6 paneles (pantalla de inicio, modo de operación, ajustes automáticos, ajustes manuales, supervisión del servomotor y registro de alarmas). Todos ellos poseen la fecha, la hora y el logotipo de la Escuela Técnica Industrial de Araiz Suministros Eléctricos. Además, desde todos ellos podemos acceder directamente a los paneles de pantalla de inicio, modo de operación, supervisión del servomotor y registro de alarmas.

7.1. Pantalla de Inicio

Como podemos apreciar en la figura 26, es un panel a través del que podemos acceder a todos los paneles principales del proyecto y además es el panel desde el cual daremos tensión al servomotor.



Figura 26: Pantalla de Inicio.

7.2. Pantalla modo de operación

Desde esta pantalla (figura 27), el usuario podrá hacer un homing dándole al botón de home para conseguir así que la lija comience en la posición 0. También, podrá elegir y abandonar el modo de operación de la máquina, que como ya sabemos es el modo manual y el automático. Y acceder a los ajustes de los diferentes modos de funcionamiento.



Figura 27: Pantalla de modo de operación.

7.3. Pantalla ajustes manuales

En esta pantalla (figura 28) podemos visualizar también tanto la posición como la velocidad actual y podremos avanzar o retroceder una distancia impuesta por pantalla a la velocidad que queramos.

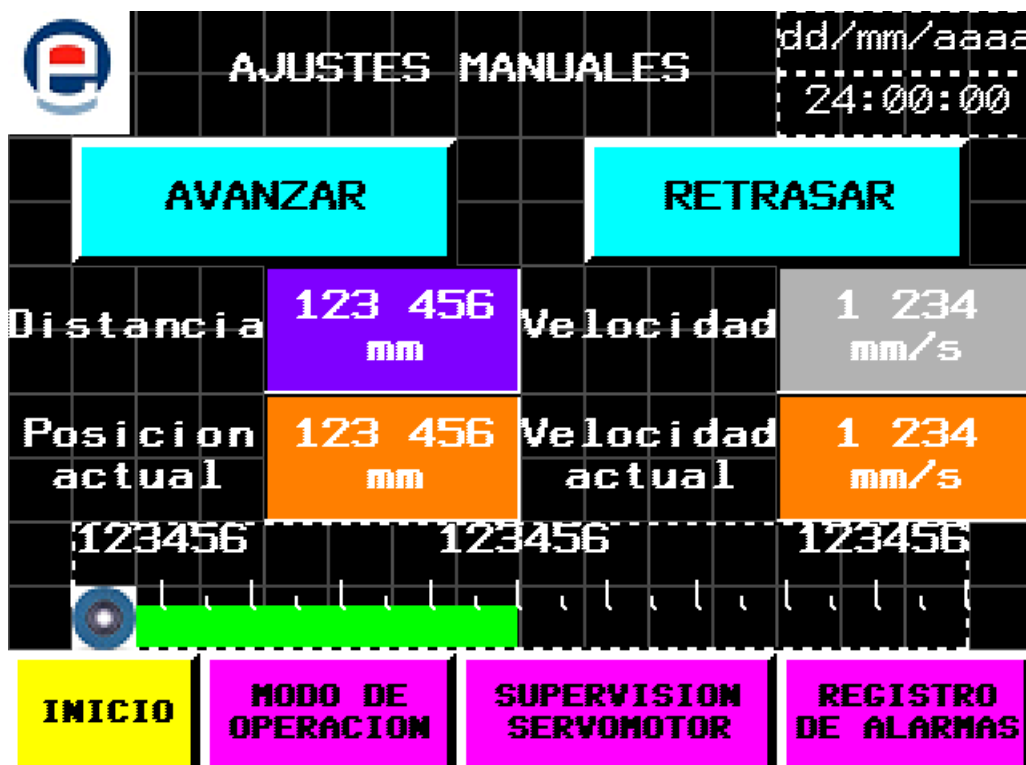


Figura 28: Pantalla de ajustes manuales.

7.4. Pantalla ajustes automáticos

En esta pantalla (figura 29) podemos observar los diferentes ajustes modificables, que son las dos posiciones en las que queremos trabajar con la lija y las velocidades a las cuales queremos ir a cada una de estas posiciones. También podemos modificar el tiempo que queremos que esté trabajando la lija en cada una de esas posiciones.

Como podemos observar también podremos ver tanto la posición actual como la velocidad actual de la lija, así como una barra para tener en cuenta visualmente la posición de esta.

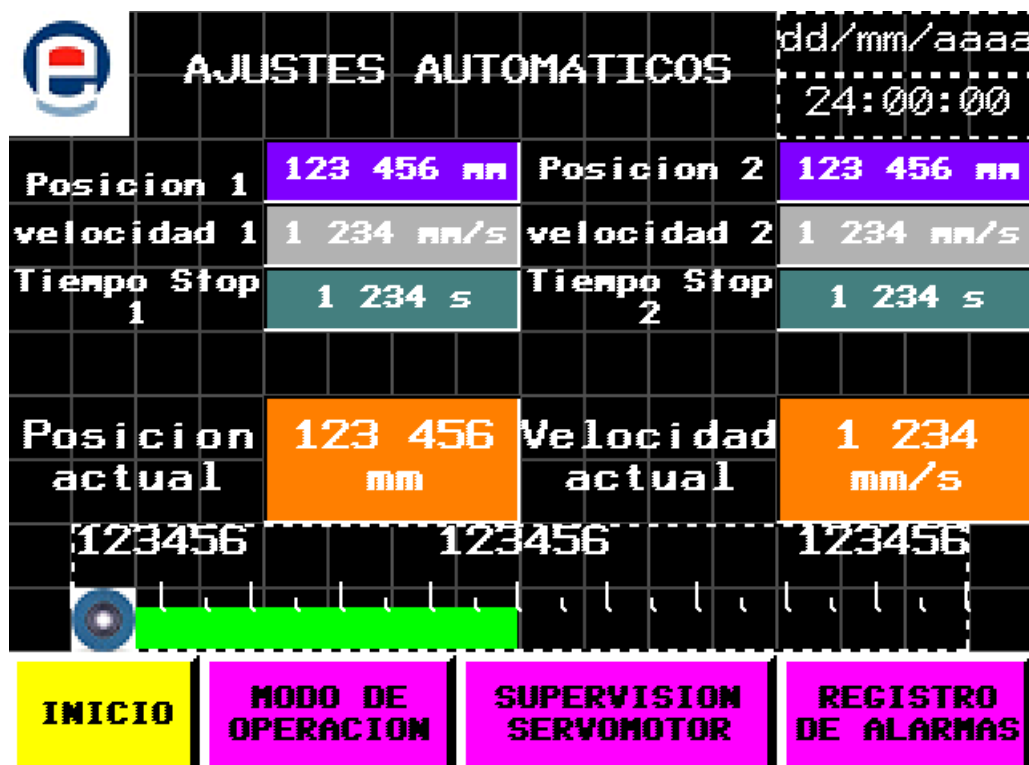


Figura 29: Pantalla de ajustes automáticos.

7.5. Pantalla de supervisión del servomotor

En esta pantalla (figura 30) podemos supervisar tanto la posición en la que se encuentra nuestra herramienta como la velocidad a la que se está moviendo.

Además, podemos observar el tiempo de tensión total que lleva nuestra máquina, el cuál podemos resetearlo mediante el botón de reset tiempo.

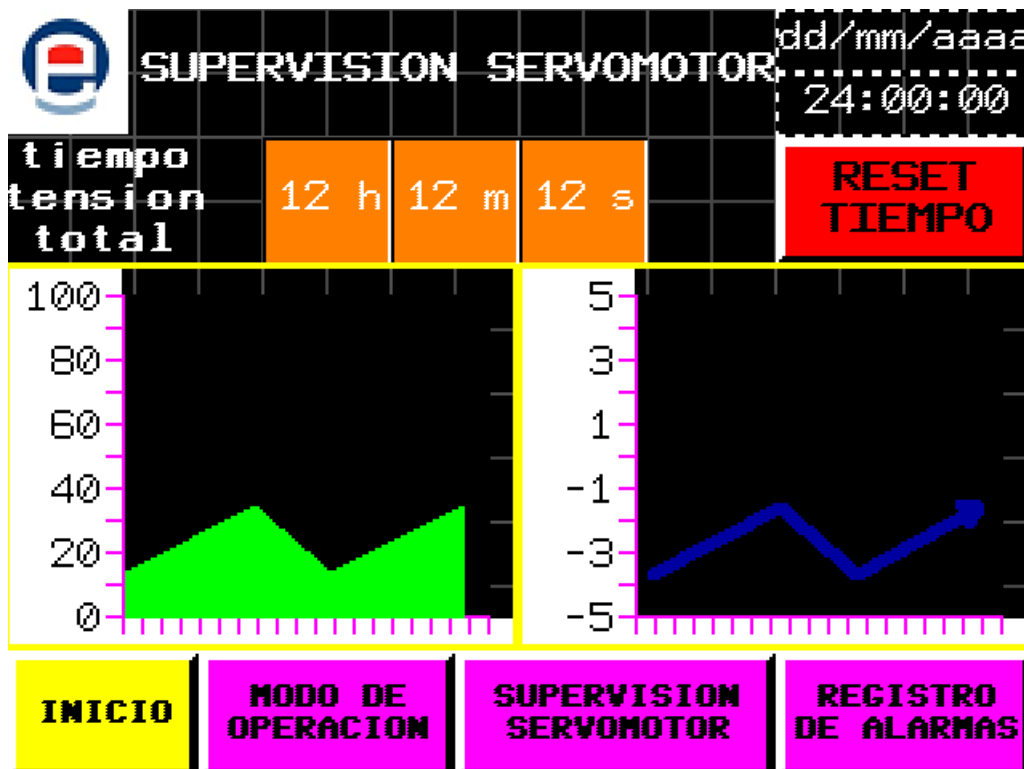


Figura 30: Pantalla de supervisión del servomotor.

7.6. Pantalla de registro de alarmas

En esta pantalla (figura 31) podemos visualizar tanto las averías como los avisos que hemos definido con anterioridad. Nos mostrará un mensaje del motivo de la avería o aviso y la hora exacta a la cual ha ocurrido.

El color rojo muestra que la avería o aviso está activa, el color amarillo expresa que en la alarma hemos reconocido la avería o aviso y al darle al botón de reset avería el mensaje se pondrá en color verde expresando que la avería ha sido arreglada.

También se pueden identificar, seleccionándolas y presionando los botones de la parte superior izquierda, para identificar los errores.



Figura 31: Pantalla de registro de alarmas.

8. Visualizador Web

También he diseñado un visualizador web para la máquina lijadora en este proyecto, que proporciona información básica sobre la lija para que pueda ser monitoreada fácilmente en todo momento.

El servidor web permite el control y seguimiento remoto de determinadas funciones de la máquina lijadora conectando el router al PLC y accediendo desde un ordenador o teléfono móvil. Esta es una herramienta útil, ya que permite a los usuarios averiguar de manera remota qué salió mal con el sistema automatizado en caso de avería, o controlar si la máquina está siempre en funcionamiento.

El servidor web presenta el siguiente aspecto mostrado en la figura 32.

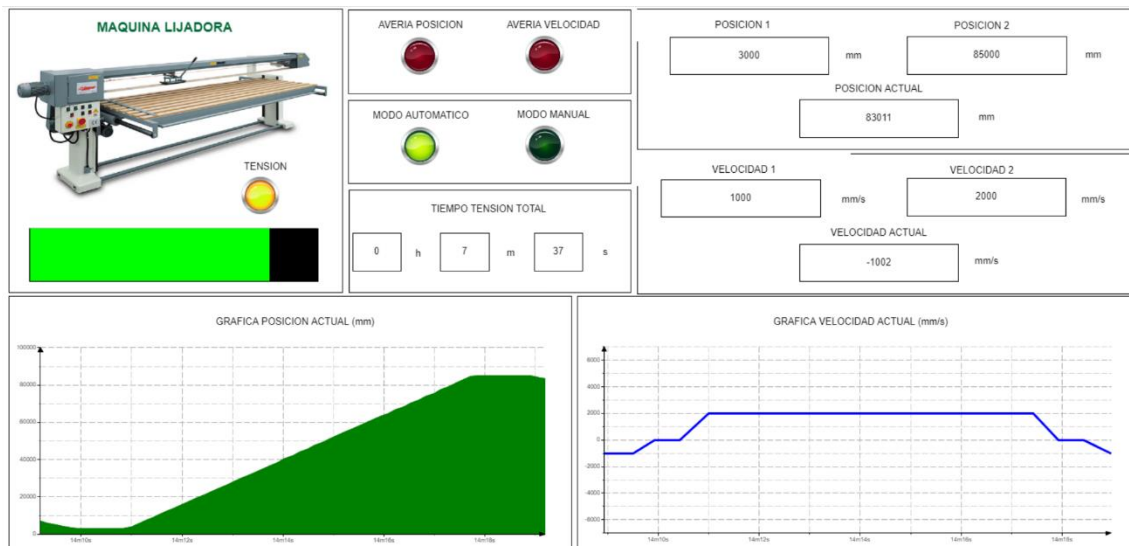


Figura 32: Pantalla del visualizador web.

Cuenta con 5 pilotos en los que se encuentra en todo momento el modo en el que se encuentra la máquina, tensión, manual, automático, avería de posición o avería de velocidad. También aparece una barra verde para poder visualizar la posición en la que se encuentra la lija de la máquina.

En esta pantalla, también podemos visualizar el tiempo en tensión que lleva la máquina y los parámetros de posiciones y velocidades del modo automático. A pesar de que el objetivo principal de esta pantalla es la supervisión, en esta parte también se pueden modificar los parámetros de posiciones y velocidades.

Por último, podemos visualizar desde aquí de manera más legible que en la pantalla las gráficas de posición y velocidad actuales.

9. Legislación

Las leyes que hay que tener en cuenta al realizar un proyecto como este son las siguientes:

- Directiva D.C. de baja tensión 2014/35/UE.
- Directiva D.C de máquinas 2006/42/CEE.
- Directiva D.C de compatibilidad electromagnética 2014/30/UE.
- Seguridad de espacio libre mínimo para evitar la colisión de partes del cuerpo EN 349:1993.
- Seguridad de máquinas, prevención y protección contra incendios EN 13478:2001.
- Seguridad del equipo eléctrico de la máquina EN 60204-1:1997.
- Accionamiento eléctrico de velocidad variable EN 61800-5-2:2007.
- Partes de un sistema de control relacionado con la seguridad EN ISO 13849-1:2006.
- Seguridad terminología básica, principios generales de diseño EN ISO 12100-1:2004.
- Parada de emergencia, principio de diseño EN ISO 13850:2006.

10. Presupuesto

En este punto, el costo total incurrido para completar el proyecto se establece como presupuesto. El presupuesto solo se desglosa en hardware y mano de obra del proyecto que hemos realizado.

Hay que comentar que en Araiz Suministros Eléctricos no realizamos la mano de obra, pero hablaríamos con nuestros electricistas de confianza para ofrecer un presupuesto del montaje de los aparatos.

También he de comentar que en Araiz Suministros Eléctricos aportamos un servicio técnico totalmente gratuito. Esto es algo que nos diferencia con la mayoría de las empresas y es por esto por lo que el presupuesto solo se basa en el hardware y la mano de obra.

Tanto el software como el servicio técnico posterior si el cliente tuviera alguna duda o cambio de este es totalmente gratuito.

10.1. Mano de obra

A continuación, se detallan los costos involucrados en la construcción del proyecto (tabla 1). Para la ejecución detallada del presupuesto en este apartado, hay que tener en cuenta el salario de los ingenieros de montaje, con un salario por hora de 15 euros.

Tareas	horas	h/€	precio total (€)
Instalación	7	15	105
Montaje	10	15	150
Comprobación	2	15	30
TOTAL			285
TOTAL (+IVA)			344,85

TABLA 1: PRECIO DE LA MANO DE OBRA.

10.2. Hardware

A continuación, se muestran los materiales que podemos proporcionar desde Araiz Suministros Eléctricos para construir la máquina lijadora (figura 33).



www.araiz.es
CR Poligono La Paz o/Estocolmo, parcela 24B
44195 - Teruel
Tfno. 978 60 13 61; Fax 978 60 78 42
teruel@araiz.es
Cif: A50079961

OFERTA VENTA N° 3059247 TER

Fecha Oferta: 21/08/2023

S/Referencia:

Att.

Página: 1

Muy Sr.(s) nuestro(s).

En atención a su amable solicitud, nos complace enviarle(s) nuestra mejor oferta, para el suministro de los siguientes materiales, quedando a su entera disposición para cualquier consulta que necesiten al respecto.

Esperando que la presente oferta merezca su aprobación, aprovechamos la ocasión para saludarles atentamente.

Descripción	Cantidad	Precio	Uds/PVP	Dtos.	Plazo Entrega	Importe
BSH0551P01A2A MOTOR 0,5Nm IP40 LISO ST ACODAD<	1,00	721,080	1	25,00		540,81
LXM32MU90M2 U.ITGDA.LXM32 MODUL.9A PICO 1F 230	1,00	894,390	1	25,00		670,79
TM241CEC24R CPU AC 14E/10S RELE ETHERNET CANOPEN	1,00	502,500	1	37,00		316,58
HMISTU855 TOUCH PANEL SCREEN .	1,00	649,640	1	39,00		396,28
HMIS5T Modulo Trasero A:118mm P:40.1mm	1,00	311,870	1	39,00		190,24
ABLM1A24012 Fuente Alimentacion Conmu. Modular 1,2A 24Vdc 30W	1,00	140,880	1	62,00		53,53
A9N21554 Int. Automatico IDPN N 1P+N 3A Curva C 10kA	1,00	10,922	1	62,00		4,15
TL-WR802N Router de Red, Inalámbrico, Nano, Tamaño de Bolsillo, 300 Mbps, 5VDC, Enchufe USB	1,00	20,900	1	35,00		13,59
MCSESU053FND Switch No Gestionado 5 Bocas 100TXRJ45	1,00	145,750	1	35,00		94,74

OBSERVACIONES:

**TOTAL OFERTA
2.280,71 EUR**
**% IVA
21,00**
**Importe IVA
478,95**
Total Oferta Venta (con I.V.A.) 2.759,66 EUR

EN CASO DE PEDIDO ROGAMOS
CITEN EL NUMERO DE OFERTA

Un Cordial Saludo Ivan Garbayo

Figura 33: Precios del hardware.

10.3. Coste total

Por lo tanto, el presupuesto total para el proyecto es el siguiente (tabla 2):

Concepto	Coste (€)
Hardware	2280,71
Mano de obra	285
TOTAL	2565,71
TOTAL (+IVA)	3104,5091

TABLA 2: PRECIO TOTAL.

11. Conclusiones

En este trabajo de fin de grado he intentado aplicar algunos de los conocimientos adquiridos durante mi estancia en el grado universitario de Ingeniería Electrónica y Automática.

Realizar el trabajo descrito en este documento me ha dado la oportunidad de conocer de primera mano cómo se lleva a cabo realmente un proyecto de ingeniería, desde la etapa de análisis hasta el diseño y una parte de la implementación.

He podido obtener algunos de los componentes del equipo que necesitaba y he podido profundizar en el conocimiento de los temas cubiertos en el proyecto.

Este Trabajo Fin de Grado ha mejorado mis habilidades de programación tanto en Ladder como en diagramas de bloques funcionales. También he podido profundizar mis conocimientos sobre redes de comunicación industriales y buses de campo.

Además, particularmente en las primeras etapas de un proyecto, es importante desarrollar una mejor comprensión de cómo funcionan las máquinas herramientas de este tipo para aumentar la eficiencia y reducir los costos a largo plazo de una empresa, comprendiendo así la importancia de automatizar los procesos industriales.

También he comprendido temas como la seguridad en el trabajo cuando se trata del uso de este tipo de máquinas y la importancia de utilizar señales y protocolos adecuados tanto en el manejo como en la instalación de los equipos para evitar accidentes.

Por lo tanto, se puede concluir que, al completar este trabajo, he mejorado mis conocimientos y habilidades como ingeniero y he actuado como una ventana para las tareas que realizan los ingenieros.

12. Bibliografía

https://bookdown.org/alberto_brunete/intro_automatica/automatizacionindustrial.html (Fundamentos teóricos). Fecha de la consulta: 17/06/2023.

<https://www.cursosaula21.com/que-es-un-servomotor/> (Información general de un servomotor). Fecha de la consulta: 02/07/2023.

<https://urany.net/blog/conoce-el-funcionamiento-de-los-servomotores> (Información específica de servomotores). Fecha de la consulta: 03/06/2023.

<https://en.wikipedia.org/wiki/EtherNet/IP> (Información sobre Ethernet IP). Fecha de la consulta: 10/07/2023.

<https://es.wikipedia.org/wiki/CANopen> (Información sobre CANopen). Fecha de la consulta: 10/07/2023.

<https://www.se.com/es/es/product/TM241CEC24R/modicon-m241-controlador-m241-24-es-rel%C3%A9-ethernet-c/> (Catálogo PLC). Fecha de la consulta: 20/07/2023.

<https://www.tecnical.cat/PDF/SCHNEIDER/PLC/M241/EIO000000143503M241.pdf> (Manual M241). Fecha de la consulta: 20/07/2023.

<https://www.se.com/es/es/product/HMISTU855/magelis-sto-stu-terminal-d22-5-7-color-qvga-eth/> (Catálogo pantalla). Fecha de la consulta: 20/07/2023.

<https://www.tecnical.cat/PDF/SCHNEIDER/HMI/EIO000000061805MAGELIS.pdf> (Manual Magelis). Fecha de la consulta: 20/07/2023.

<https://www.se.com/es/es/product/LXM32MU45M2/motion-servo-drive-lexium-32tensi%C3%B3n-de-alimentaci%C3%B3n-de-una-fase-115-230v-0-15-0-3kw/> (Catálogo servodrive). Fecha de la consulta: 20/07/2023.

https://www.tecnical.cat/PDF/SCHNEIDER/MECATRONICA/Lexium_32.pdf (Manual Lexium 32). Fecha de la consulta: 20/07/2023.

<https://www.se.com/es/es/product/BSH0551P01A1A/servo-motor-bsh-iec-055-mm-05-nm-4000-rpm-eje-liso-sin-freno-doble-conector-recto-encoder-abs-st-128b-sincos-ip54/> (Catálogo servomotor). Fecha de la consulta: 20/07/2023.

http://www.insht.es/InshtWeb/Contenidos/Normativa/ColeccionesRelacionadas/ContenidosRelacionados/TaxNormativa4_1/GuiaUEMaquinasDir%2006_42_ESP.pdf (directivas relativas a las máquinas). Fecha de la consulta: 25/07/2023.

<https://www.se.com/es/es/product/ABL8MEM24012/fuente-de-alimentaci%C3%B3n-conmutada-regulada-1-o-2-fases-100-240-v-ac-24-v-1-2-a/> (Catálogo fuente de alimentación). Fecha de la consulta: 10/08/2023.

<https://www.se.com/il/en/product/A9N21554/idpn-circuit-breaker-idpn-n-1p+-n-3a-c-curve/> (Catálogo del interruptor automático). Fecha de la consulta: 10/08/2023.



<https://www.tp-link.com/es/home-networking/wifi-router/tl-wr802n/> (Catálogo router inalámbrico). Fecha de la consulta: 10/08/2023.

<https://www.se.com/il/en/product/TCSESU053FN0/connexium-unmanaged-switch-5-ports-for-copper/> (Catálogo switch). Fecha de la consulta: 10/08/2023.

13. Anexos

13.1. Machine Expert

13.1.1. Variables de la secuencia principal

```

1  PROGRAM SR_Main
2  VAR
3      // Variables del lenguaje SFC
4      Manual: BOOL;
5      Comenzar: BOOL;
6      Automatico: BOOL;
7      Exit_Automatico: BOOL;
8      Exit_Manual: BOOL;
9
10     // Variables comunes
11     En_Power: BOOL;
12     PowerOK: BOOL;
13     Exe_Reset: BOOL;
14     PosActual: DINT;
15     VelActual: DINT;
16     FB_Power: GIPLC.MC_Power;
17     FB_Reset: GIPLC.MC_Reset;
18     FB_Posicion: GIPLC.MC_ReadActualPosition;
19     FB_Velocidad: GIPLC.MC_ReadActualVelocity;
20     MC_ReadStatus_0: GIPLC.MC_ReadStatus;
21     MC_ReadStatus_1: GIPLC.MC_ReadStatus;
22
23     // Variables del Modo Seleccion
24     Exe_Home: BOOL;
25     Home_LXM32_0: gilxm.Home_LXM32;
26     TON_0: ton;
27
28     // Variables del Automático_Run
29     PosIni: DINT :=15000;
30     PosFin: DINT :=20000;
31     Vel: DINT := 200;
32     vel2: DINT := 100;
33     Exe_Inicio: BOOL;
34     Tiempo1: TIME := T#2S;
35     Tiempo2: TIME := T#4S;
36     FB_MoveAbs: GIPLC.MC_MoveAbsolute;
37     FB_MoveAbs2: GIPLC.MC_MoveAbsolute;
38     FB_Timer2: TON;
39     FB_Timer: TON;
40
41     // Variables Manual_Run
42     exe_m_av: BOOL;
43     exe_m_ret: BOOL;
44     vel3: DINT :=500;
45     dis2: DINT := 10000;
46     MC_MoveRelative_0: giplc.MC_MoveRelative;
47     MC_MoveRelative_1: giplc.MC_MoveRelative;
48
49 END_VAR

```

```
52 | //Declaración para poder utilizar el estado Init en Alarmas
53 | VAR_INPUT
54 | SFCinit: BOOL;
55 | END_VAR
```

13.1.2. Variables de avisos y alarmas

```
1 | PROGRAM AlarmasyContadorTension
2 | VAR
3 |     HMI_Reset_Averia: BOOL;
4 |     Averia_1: BOOL;
5 |     Averia_2: BOOL;
6 |     Aviso_1: BOOL;
7 |     Aviso_2: BOOL;
8 |     fb_Un_s: BLINK;
9 |     Maq_seg: INT;
10 |    Maq_min: INT;
11 |    Maq_hora: INT;
12 |    Reset_contador: BOOL;
13 |    Maq_tens_seg: INT;
14 |    Un_s: BOOL;
15 | END_VAR
```

13.1.3. Programación de lectura del reloj

Para poder observar por pantalla la fecha y la hora de manera legible he tomado las variables numéricas de hora, minutos, segundos, día y mes convirtiéndolas en cadenas de caracteres formateadas.

Esto se ha conseguido utilizando la función CONCAT. Para la visualización del reloj, si la variable es menor de 10 entonces agregamos un '0' o ':0', de lo contrario agregamos '' o ':' según nos convenga para que la hora sea legible.

En el caso de la fecha hemos realizado el mismo criterio, pero evidentemente con '0' o '/0' para las variables menores que 10 y '' o '/' para lo contrario.

```

3  IF horas<10 THEN
4      hh:= CONCAT ('0',DINT_TO_STRING(horas));
5      ELSE
6      hh:= CONCAT ('',DINT_TO_STRING(horas));
7  END_IF
8
9  IF minutos<10 THEN
10     mm:= CONCAT (':0',DINT_TO_STRING(minutos));
11     ELSE
12     mm:= CONCAT (':',DINT_TO_STRING(minutos));
13 END_IF
14
15 IF segundos<10 THEN
16     ss:= CONCAT (':0',DINT_TO_STRING(segundos));
17     ELSE
18     ss:= CONCAT (':',DINT_TO_STRING(segundos));
19 END_IF
20
21 mmss:= CONCAT (mm,ss);
22 hhmmss:= CONCAT (hh,mmss);
23
24
25 IF dia<10 THEN
26     dd:= CONCAT ('0',DINT_TO_STRING(dia));
27     ELSE
28     dd:= CONCAT ('',DINT_TO_STRING(dia));
29 END_IF
30
31 IF mes<10 THEN
32     nn:= CONCAT ('/0',DINT_TO_STRING(mes));
33     ELSE
34     nn:= CONCAT ('/',DINT_TO_STRING(mes));
35 END_IF
36
37 ddnn:= CONCAT (dd,nn);
38 aaaa:= CONCAT ('/',DINT_TO_STRING(ano));
39 ddmmaaaa:= CONCAT (ddnn,aaaa);

```

13.1.4. Variable de consigue hora

```

1  PROGRAM consigue_hora
2  VAR
3      F_TRIG_0: F_TRIG;
4  END_VAR
5

```

13.1.5. Variables del reloj

```
1  VAR GLOBAL
2
3  sincronizaFH: BOOL;
4
5  //VARIABLES DE TIEMPO
6  GetTimeResult: UDINT;
7  UTCtoDate_diag: UDINT;
8  Ctrl_DateTime: SYSTIMEDATE;
9
10 ano AT %mw100: UINT;
11 mes AT %MW101: UINT;
12 dia AT %MW102: UINT;
13 horas AT %MW103: UINT;
14 minutos AT %MW104: UINT;
15 segundos AT %MW105: UINT;
16 dia_semana: UINT;
17
18 hhmmss: STRING (8);
19 ddmmaaaa: STRING(10);
20
21 END VAR
```

13.2. Vijeo Designer

13.2.1. Script reloj

Script

```
//Puesta en hora

int hour_PLC,min_PLC,sec_PLC;

hour_PLC=_SoM.MyController.Application.GVL_Hora.horas.getIntValue();
min_PLC=_SoM.MyController.Application.GVL_Hora.minutos.getIntValue();
sec_PLC=_SoM.MyController.Application.GVL_Hora.segundos.getIntValue();

Sys.setTime(hour_PLC,min_PLC,sec_PLC);

//Puesta en fecha

int ano_PLC,mes_PLC,dia_PLC;

ano_PLC=_SoM.MyController.Application.GVL_Hora.ano.getIntValue();
mes_PLC=_SoM.MyController.Application.GVL_Hora.mes.getIntValue();
dia_PLC=_SoM.MyController.Application.GVL_Hora.dia.getIntValue();

Sys.setDate(ano_PLC,mes_PLC,dia_PLC);
```

13.2.2. Declaración del grupo de alarmas

	Variable	Grupo de alarm...	Origen de datos	Dirección de dispositivo	Mensaje
1	_SoM.MyController...	AlarmGroup1	Externo	Application.AlarmasyContadorTension.Averia_1	Pos. inadmisible <timestamp:h:mm:ss a dd/MM/yyyy>
2	_SoM.MyController...	AlarmGroup1	Externo	Application.AlarmasyContadorTension.Averia_2	Vel. inadmisible <timestamp:h:mm:ss a dd/MM/yyyy>
3	_SoM.MyController...	AlarmGroup1	Externo	Application.AlarmasyContadorTension.Aviso_1	Pos. peligrosa <timestamp:h:mm:ss a dd/MM/yyyy>
4	_SoM.MyController...	AlarmGroup1	Externo	Application.AlarmasyContadorTension.Aviso_2	Vel. peligrosa <timestamp:h:mm:ss a dd/MM/yyyy>

13.2.3. Declaración de registro de datos

	Variable	Grupo de registros	Origen de datos	Dirección de dispositivo
1	_SoM.MyController.Application.SR_Main.VelActual	GrupoDeRegistros02	Externo	Application.SR_Main.VelActual
	Variable	Grupo de registros	Origen de datos	Dirección de dispositivo
1	_SoM.MyController.Application.SR_Main.PosActual	LoggingGroup01	Externo	Application.SR_Main.PosActual

13.2.4. Declaración de variables

Nombre	Tipo de datos	Origen de datos	Grupo de escaneo	Dirección de dispositivo	Grupo de alarm...	Grupo de regist...
SoM						
MyController						
Application						
AlarmasyContad						
Averia_1	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.Averia_1	AlarmGroup1	Ninguno
Averia_2	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.Averia_2	AlarmGroup1	Ninguno
Aviso_1	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.Aviso_1	AlarmGroup1	Ninguno
Aviso_2	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.Aviso_2	AlarmGroup1	Ninguno
fb_Un_s	Structure	Externo	SOM_MyController			
HMI_Reset_Avi	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.HMI_Reset_Averia	Desactivado	Ninguno
Maq_hora	INT	Externo	SOM_MyController	Application.AlarmasyContadorTension.Maq_hora	Desactivado	Ninguno
Maq_min	INT	Externo	SOM_MyController	Application.AlarmasyContadorTension.Maq_min	Desactivado	Ninguno
Maq_seg	INT	Externo	SOM_MyController	Application.AlarmasyContadorTension.Maq_seg	Desactivado	Ninguno
Maq_tens_seg	INT	Externo	SOM_MyController	Application.AlarmasyContadorTension.Maq_tens_seg	Desactivado	Ninguno
Reset_contado	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.Reset_contador	Desactivado	Ninguno
Un_s	BOOL	Externo	SOM_MyController	Application.AlarmasyContadorTension.Un_s	Desactivado	Ninguno

GVL_Hora						
ano	UINT	Externo	SOM_MyController	Application.GVL_Hora.ano	Desactivado	Ninguno
ddmmaaaa	STRING	Externo	SOM_MyController	Application.GVL_Hora.ddmmaaaa		Ninguno
dia	UINT	Externo	SOM_MyController	Application.GVL_Hora.dia	Desactivado	Ninguno
hhmmss	STRING	Externo	SOM_MyController	Application.GVL_Hora.hhmmss		Ninguno
horas	UINT	Externo	SOM_MyController	Application.GVL_Hora.horas	Desactivado	Ninguno
mes	UINT	Externo	SOM_MyController	Application.GVL_Hora.mes	Desactivado	Ninguno
minutos	UINT	Externo	SOM_MyController	Application.GVL_Hora.minutos	Desactivado	Ninguno
segundos	UINT	Externo	SOM_MyController	Application.GVL_Hora.segundos	Desactivado	Ninguno

Nombre	Tipo de datos	Origen de datos	Grupo de escaneo	Dirección de dispositivo	Grupo de alarm...	Grupo de regist...
SoM						
MyController						
Application						
AlarmasyContad						
GVL_Hora						
SR_Main						
Automatico	BOOL	Externo	SOM_MyController	Application.SR_Main.Automatico	Desactivado	Ninguno
Automatico_	Structure	Externo	SOM_MyController			
Comenzar	BOOL	Externo	SOM_MyController	Application.SR_Main.Comenzar	Desactivado	Ninguno
dis2	DINT	Externo	SOM_MyController	Application.SR_Main.dis2	Desactivado	Ninguno
En_Power	BOOL	Externo	SOM_MyController	Application.SR_Main.En_Power	Desactivado	Ninguno
Exe_Home	BOOL	Externo	SOM_MyController	Application.SR_Main.Exe_Home	Desactivado	Ninguno
Exe_Inicio	BOOL	Externo	SOM_MyController	Application.SR_Main.Exe_Inicio	Desactivado	Ninguno
exe_m_av	BOOL	Externo	SOM_MyController	Application.SR_Main.exe_m_av	Desactivado	Ninguno
exe_m_ret	BOOL	Externo	SOM_MyController	Application.SR_Main.exe_m_ret	Desactivado	Ninguno
Exe_Reset	BOOL	Externo	SOM_MyController	Application.SR_Main.Exe_Reset	Desactivado	Ninguno
Exit_Automati	BOOL	Externo	SOM_MyController	Application.SR_Main.Exit_Automatico	Desactivado	Ninguno
Exit_Manual	BOOL	Externo	SOM_MyController	Application.SR_Main.Exit_Manual	Desactivado	Ninguno
FB_MoveAbs	Structure	Externo	SOM_MyController			
FB_MoveAbs2	Structure	Externo	SOM_MyController			
FB_Posicion	Structure	Externo	SOM_MyController			
FB_Power	Structure	Externo	SOM_MyController			
FB_Reset	Structure	Externo	SOM_MyController			
FB_Timer	Structure	Externo	SOM_MyController			
FB_Timer2	Structure	Externo	SOM_MyController			
FB_Velocidad	Structure	Externo	SOM_MyController			

Home_LXM32	Structure	Externo	SOM_MyController			
Init	Structure	Externo	SOM_MyController			
Manual	BOOL	Externo	SOM_MyController	Application.SR_Main.Manual	Desactivado	Ninguno
Manual_Run	Structure	Externo	SOM_MyController			
MC_MoveRela	Structure	Externo	SOM_MyController			
MC_MoveRela	Structure	Externo	SOM_MyController			
MC_ReadStatu	Structure	Externo	SOM_MyController			
MC_ReadStatu	Structure	Externo	SOM_MyController			
Modo_Seleccic	Structure	Externo	SOM_MyController			
PosActual	DINT	Externo	SOM_MyController	Application.SR_Main.PosActual	Desactivado	LoggingGroup01
PosFin	DINT	Externo	SOM_MyController	Application.SR_Main.PosFin	Desactivado	Ninguno
PosIni	DINT	Externo	SOM_MyController	Application.SR_Main.PosIni	Desactivado	Ninguno
PowerOK	BOOL	Externo	SOM_MyController	Application.SR_Main.PowerOK	Desactivado	Ninguno
SFCinit	BOOL	Externo	SOM_MyController	Application.SR_Main.SFCinit	Desactivado	Ninguno
Tiempo1	UDINT	Externo	SOM_MyController	Application.SR_Main.Tiempo1	Desactivado	Ninguno
Tiempo2	UDINT	Externo	SOM_MyController	Application.SR_Main.Tiempo2	Desactivado	Ninguno
TON_0	Structure	Externo	SOM_MyController			
Vel	DINT	Externo	SOM_MyController	Application.SR_Main.Vel	Desactivado	Ninguno
vel2	DINT	Externo	SOM_MyController	Application.SR_Main.vel2	Desactivado	Ninguno
vel3	DINT	Externo	SOM_MyController	Application.SR_Main.vel3	Desactivado	Ninguno
VelActual	DINT	Externo	SOM_MyController	Application.SR_Main.VelActual	Desactivado	GrupoDeRegistr...

13.2.5. Implementación en gráficas de supervisión del servomotor

En las gráficas en la pantalla de la supervisión del servomotor he implementado una función para que cambien a color rojo indicando peligro de velocidad y posición cuando se superan los 90000 mm o los ± 3000 mm/s.

Canal1	Habilitado
Variable	_SoM.MyController.Application.SR_Main.PosActual
Formato de visualización	Relleno
Línea de base	Desactivado
Marca	Desactivado
Color	
Color de primer plano	 (0,255,0)
Color de fondo	 (0,0,0)
Diseño	 1: Sólido
Fuera de rango	Habilitado
Mínimo	0
Máximo	90000
Color de primer plano	 (255,0,0)
Color de fondo	 (0,0,0)
Diseño	 1: Sólido



[-] Canal1	Habilitado
[-] Variable	_SoM.MyController.Application.SR_Main.VelActual
[-] Formato de visualización	Línea
[-] Línea de base	Desactivado
[-] Marca	Desactivado
[+] Color	
[-] Fuera de rango	Habilitado
[-] Mínimo	-3000
[-] Máximo	3000
[-] Color de línea	 (255,0,0)
[-] Estilo de línea	 0: Sólido
[-] Ancho de línea	 4
[-] Color de fondo	 (0,0,0)