UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

PROGRAMACIÓN Y PUESTA EN MARCHA DE TABLERO DE PRÁCTICAS PARA CONTROL DE SERVOMOTOR CONECTADO A EJE LINEAL

TRABAJO FINAL DEL

Grado en Ingeniería Eléctrica

REALIZADO POR

Miguel Domínguez Belloch

TUTORIZADO POR

Manuel Pineda Sánchez

Juan Pérez Cruz

FECHA: **Valencia, Septiembre, 2019**

Agradecimientos

Agradecer todo el apoyo recibido por parte de mi familia, pareja y amigos.

Especialmente a mis padres, Miguel y Conchi,

por todos los sacrificios que han realizado para que llegará este momento

Resumen

En el presente trabajo se lleva a cabo la puesta en marcha de un tablero, perteneciente al departamento de ingeniería eléctrica (DIE), para el control de un servomotor, el cual se acoplará a una bancada. Esta bancada será nuestro sistema de eje lineal. Con esto, podremos simular una línea de montaje.

Los equipos de los que disponemos son de la empresa japonesa Omron, tanto el autómata programable, el servo accionamiento, la placa de conexiones y el servomotor. Todos estos equipos son cedidos por el departamento para la realización de este proyecto.

Para la programación del autómata, se hará uso del programa CX-Programmer, propiedad de Omron. Para el acceso de este programa, el departamento permite el uso de sus ordenadores, los cuales cuentan con esta herramienta de programación.

La metodología de trabajo trata primero de una revisión completa de los equipos, cerciorándonos que el conjunto al completo se encuentra en condiciones de uso. Tras ello, una correcta configuración del servo-pack o variador de frecuencia, para controlar el servomotor mediante la botonera física que incorpora el tablero. Por último, el funcionamiento y control del servomotor, pero llevado a cabo mediante la programación de un autómata.

Resum

En el present treball es duu a terme la posada en marxa d'un tauler, pertanyent al departament d'enginyeria elèctrica (DIE), per al control d'un servomotor, el qual s'acoblarà a una bancada. Aquesta bancada serà el nostre sistema d'eix lineal. Amb això, podrem simular una línia de muntatge.

Els equips dels quals disposem són de l'empresa japonesa Omron, tant l'autòmat programable, el servo accionament, la placa de connexions i el servomotor. Tots aquests equips són cedits pel departament per a la realització d'aquest projecte.

Per a la programació de l'autòmat, es farà ús del programa CX-programmer, propietat d'Omron. Per a l'accés d'aquest programa, el departament permet l'ús dels seus ordinadors, els quals compten amb aquesta eina de programació.

La metodologia de treball tracta primer d'una revisió completa dels equips, cerciorant-nos que el conjunt al complet es troba en condicions d'ús. Després d'això, una correcta configuració del servo-pack o variador de freqüència, per a controlar el servomotor mitjançant la botonera física que incorpora el tauler. Finalment, el funcionament i control del servomotor, però portat acabe mitjançant la programació d'un autòmat.

Abstract

In this project will turn on a table, which is owned by the electrical engineering department (DIE), to control a servomotor that will be linked to a linear axis. With this idea, we may simulate an assembly line.

All the stuff that we have are from the Japanese brand that is called Omron. The programmable automaton (PLC), the frequency drive, the connection board and the servomotor. All this stuff is given by the department to make this project possible.

To program the PLC, it will use the CX-programmer from Omron. To access to this tool, the department allows to use its computers, which have this programmer tool.

The work methodology starts with the check out of all the components, to look after that everything is in condition to use. After that, a correct configuration of the servo-pack or frequency drive, to control the servomotor with physical controls that is part of the table. Finally, the operation and control of the servomotor, although with the programming of an automaton.

Memoria

Índice

1. Objetivo

El objetivo de este proyecto será la puesta en marcha del tablero que el departamento de ingeniería para el control de un servomotor, el cual se acoplará a un Sistema de engranajes para el movimiento de lineal de una plataforma. Simularemos una línea de montaje controlada por un autómata.

Se realizará un Scada, con el objetivo de permitir a un operario controlar desde un ordenador el sistema entero desde un lugar remoto. Además, se llevará a cabo la realización de una pantalla con la cual queremos simular el control de la línea de montaje a pie de línea por otro operario.

2. Introducción

Como indica el título del presente proyecto, el equipo a utilizar para la acción de movimiento es un servomotor síncrono. La elección de este tipo de motor se debe a la precisión que tiene con respecto a otro tipo de motor eléctrico que podemos encontrar en el mercado, como son los motores asíncronos de inducción.

Los servomotores son dispositivos de accionamientos que permiten un control preciso tanto en su utilización en control de velocidad, posición y par. Gracias a esta característica que presentan, sus aplicaciones en la industria son aquellas que presentan la necesidad de alta precisión, como puede ser una línea de mecanizado, donde la necesidad de la colocación de la pieza en distintas posiciones exactas, que serán aquellas estaciones en donde se realizarán todas las acciones para obtener el producto final, hace necesario la utilización de este tipo de accionamiento. Esta idea es la que se quiere simular en el proyecto, acoplando el servomotor a un eje lineal, con el cual el movimiento giratorio del motor, lo convertimos en un movimiento lineal con la utilización de un sistema de engranaje con cadena. Esta acción se repite infinitas veces, a muy alta velocidad, lo que para el servomotor no es ningún inconveniente. Sin embargo, la repetición constante y a alta velocidad, sería algo que un variador de frecuencia no podría asumir.

El servomotor, además, tiene la ventaja de poder mantener el par desde velocidad nula, hasta la velocidad nominal.

Otra idea a tener en cuenta es que los servomotores trabajan en bucle cerrado, lo que les permite medir en todo momento la posición del eje. Para ello, los servomotores tienen incorporados un encoder.

Ilustración 1. Servomotor Omron

Los enconders ópticos son sensores que permiten detectar el movimiento de rotación de un eje. La idea de su funcionamiento es la de convertir una magnitud (posición lineal y angular), en una señal digital. Son colocados solidariamente al eje el cual se quiere conocer su posición. Se utiliza luz para conocer sus medidas. Detectando la presencia o ausencia de luz a través del disco, se determina la posición.

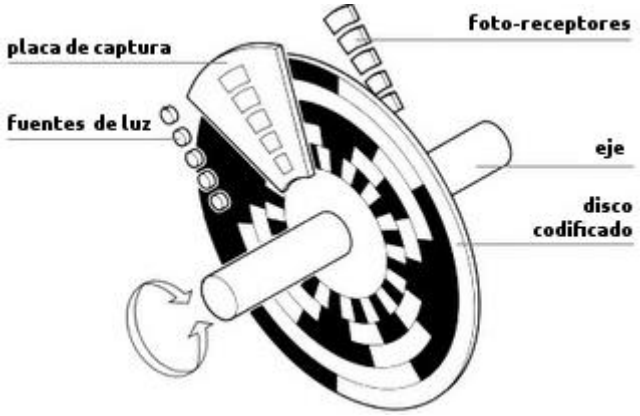


Ilustración 2. Esquema encoder

Los servomotores también se encuentran acompañados de un servo drive o servo accionamiento, los cuales se encargan de mandar los valores correspondientes de magnitudes tales como tensión o intensidad al motor para que estos funcionen acorde a la señal de movimiento que se ha programado.

El servo accionamiento puede controlar el par, la velocidad o la posición. Por ello, podemos realizar distintos tipos de control, eligiendo el más adecuado a la aplicación para la que se quiera realizar.

Mediante el uso de sensores, el motor manda señales al servo driver. Según la señal, el servo driver variará las magnitudes para que el motor trabaje según la consigna que desea el usuario. También es el encargado de bloquear el servomotor en caso de una señal anómala que pudiera dañarlo, como puede ser una situación de “overspeed”.

Ilustración 3. Servo accionamientos

Como hemos comentado previamente, el servo accionamiento recibe unas señales, las cuáles son la consigna que ha programado el usuario para que el motor trabaje para la aplicación que se desea. Estas señales que se mandan provienen de un autómata programable que trabaja según el programa que se ha diseñado para la realización de una acción.

Los autómatas programables son sistemas electrónicos diseñados para un uso industrial, los cuáles están dotados de memoria para el almacenamiento del programa de control diseñado por un individuo y para el almacenamiento de datos del programa que serán la interfaz del operario y el proceso.

La aparición de los autómatas data de la década de los 60, cuando se buscaban nuevas tecnologías electrónicas con el objetivo de reemplazar los sistemas de control basados en circuitos eléctricos con relés, interruptores y otros componentes comúnmente utilizados para el control de los sistemas de lógica combinacional cableada. Además, el uso de esta tecnología dotaría de mayor versatilidad a los procesos, el tiempo de ciclos se vería reducido y, en especial, la adaptación a los cambios sería mejor que los sistemas que había por aquel entonces.

El primer autómata fue creado por la empresa Modicon (**MO**dular **DI**gital **CON**troller), actualmente perteneciente a Schneider Electric. Su nombre fue Modicon 084, debido a que fue el proyecto número 84 realizado por Bedford Associates group, fundadora de Modicon. Esta propuesta fue la ganadora en 1968 del concurso organizado por GM Hydramatic, perteneciente a General Motors, con el objetivo que se comentaba en el párrafo anterior, el reemplazo electrónico de los sistemas cableados de relés de la empresa que organizó el concurso.

Ilustración 4. El 084 con sus creadores

Los PLC trabajan de forma cíclica, repitiendo infinitas veces la misma acción: Lee entradas, ejecuta programa y escribe salidas.

Los autómatas tienen distintas entradas y salidas que permiten que estén conectados con el proceso exterior. Las E/S (I/O) pueden ser de tipo analógico o digital, de forma compacta o modular, concentradas o distribuidas. Este tipo de sistema posee diferentes funciones tales como son las funciones lógicas, de temporización, contaje, etc.

Para la programación de los autómatas, en el mercado se encuentran distintos software propiedad de distintas marcas. Por lo general, un autómata de una marca será programado con el software que haya desarrollada esta misma. Esto lo podemos encontrar en marcas como Siemens con el TIA Portal, Schneider Electric con el Zelio Soft u Omron con su CX-programmer.

La programación se puede realizar en distintos lenguajes. La normativa IEC 61131-3, define 5 lenguajes de programación de PLC:

- Diagrama de Escalera (Ladder o Contactos)
- Diagrama de Bloques (Function Block Diagram)
- Grafcet (Structure Function Chart)
- Lista Instrucciones (Instructions List)
- Texto Estructurado (Structure List)

Fuera de esta normativa, se pueden encontrar otros lenguajes para la programación de autómatas, como puede ser el lenguaje CFC (Continuous Function Chart).

La utilización de los PLC se da fundamentalmente en aquellas aplicaciones en donde es necesario un proceso de maniobra, control, señalización, etc. Por ello, el campo de aplicación de estos sistemas es muy amplio. Desde procesos de fabricación industrial en cualquier campo hasta procesos de control de instalaciones. Incluimos también el ámbito de la docencia, en donde podemos simular distintas situaciones gracias a estos aparatos, como podría ser el control de silos en una fábrica.

Con respecto a la empresa Omron, la cual la gran mayoría del equipo que disponemos son productos suyos, podemos comentar que es una empresa japonesa fundada en 1933 por Kazuma Tatesi. Son los encargados de desarrollar el primer relé temporizado para su uso en máquinas de rayos X en un hospital de Osaka.

La empresa tiene su sede en Kioto, Japón y su negocio es principalmente la fabricación y venta de componentes, equipos y sistemas de automatización industrial. Sin embargo, la empresa es más conocida por sus productos en el campo de la salud, como son los termómetros digitales, monitores de presión arterial, nebulizadores, etc.

A lo largo de su historia, destaca que la empresa desarrolló la primera puerta automatizada del mundo y el primer cajero automático del mundo.

Ilustración 5. Logo de la empresa Omron

3. Equipos

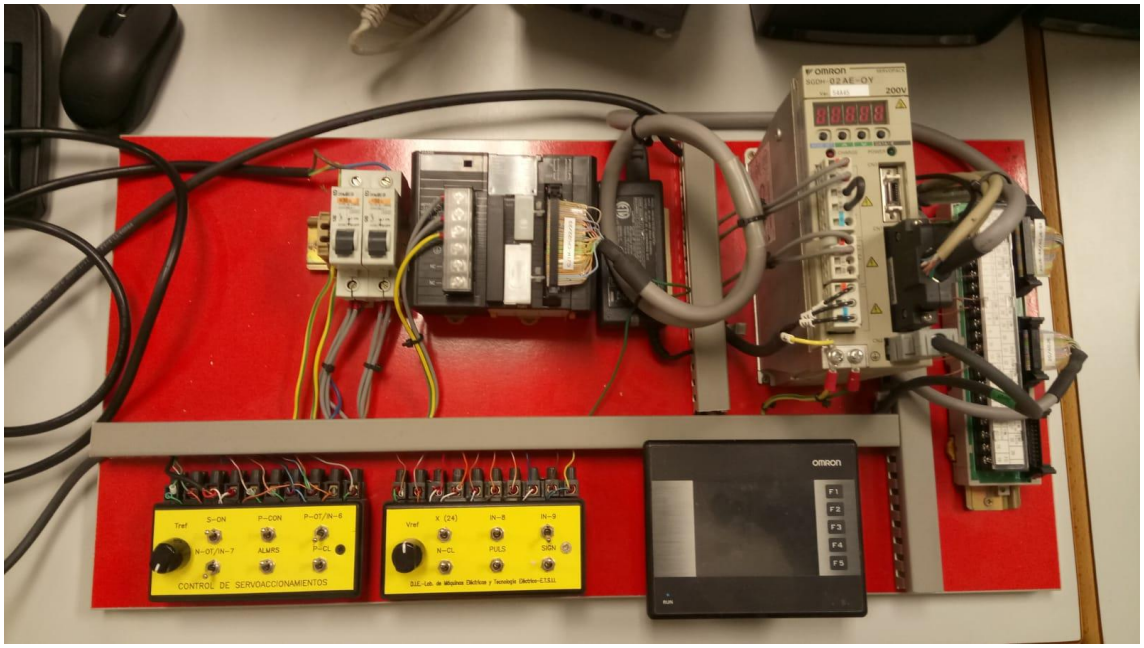


Ilustración 6. Tablero de trabajo

3.1 Servo drive

Como se ha comentado en el apartado anterior, el servo drive es el encargado del control del servomotor para que este actué conforme requiere el proceso y de protegerlo, puesto que se encuentra en todo momento en comunicación con el motor para que este no resulte dañado, mediante el corte de alimentación al motor. Además, según la señal que haya recibido el servo accionamiento para el corte de tensión al servomotor, este indica un tipo de error para que la persona que trabaja con los equipos conozca el origen de fallo.

Para nuestra aplicación, el servo accionamiento a utilizar es el Omron SGDH-02AE-OY, de alimentación monofásica a una tensión de 200 V.

Ilustración 7. Omron SGDH-02AE-OY

El nombre que recibe el servo accionamiento es un código con el cual podemos conocer sus características como vamos a mostrar a continuación:

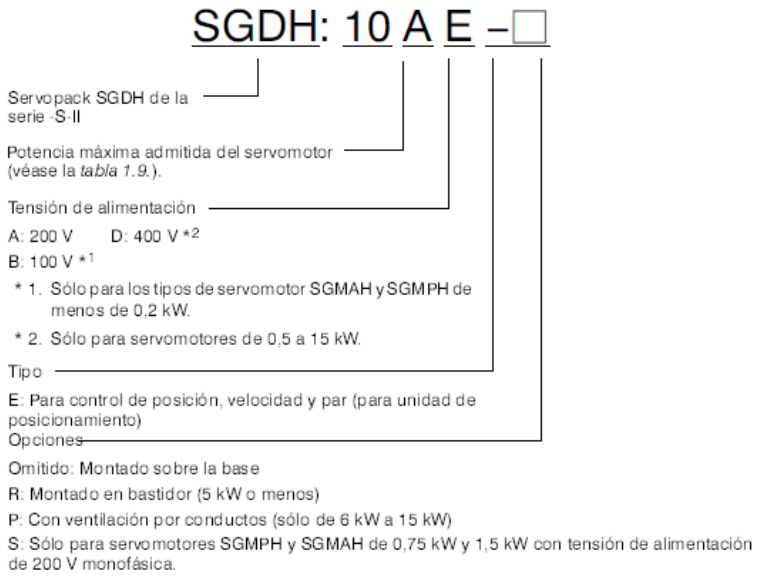


Ilustración 8. Explicación código servo drive

Símbolo de potencia máxima admitida del servomotor	Potencia (en kW)
A3	0.03
A5	0.05
01	0.10
02	0.20
04	0.40
05	0.45
08	0.75
10	1.0
15	1.5
20	2.0
30	3.0
50	5.0
60	6.0
75	7.5
1A	11
1E	15

Tabla 1. Información extraída de la tabla 1.9 del manual de usuario de SGMAH/SGDH con la correspondencia de código del servo drive y la potencia máxima admitida del servomotor

Por tanto, con la información que se detallan en la ilustración 7 y en la ilustración 8, podemos obtener las características del SGDH-02AE-OY:

- **SGDH:** Servo pack SGDH de la serie -S-II.
- **02:** La potencia máxima admitida con del servomotor será de 0,2 kW.
- **A:** 200 V tensión de alimentación.
- **E:** Para control de posición, velocidad y par (para unidad de posicionamiento).
- **-:** Montado sobre la base.

Después de analizar el código del nombre del servo accionamiento, vamos a mostrar la placa de características de este y comentar la información que aparece indicada.

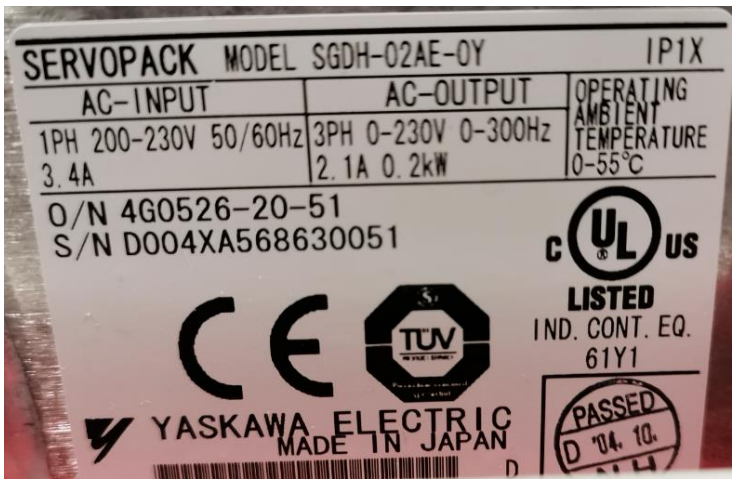


Ilustración 9. Placa características Servo drive

Podemos observar en la parte superior de la placa del modelo, el número de la serie mediante la indicación S/N.

Con respecto a la alimentación, y como se ha comentado previamente, será monofásica con valor de tensión de 200 a 230 V y con frecuencias tanto de 50 Hz como de 60 Hz.

La salida que genera este variador es de valores de tensión de 0 a 230 V, con frecuencias que se encuentran entre 0 a 300 Hz.

Su nivel de protección IP es de IP1X, como vemos en la parte superior derecha de la imagen. Esta indicación nos dice que:

- **1:** Protección contra el polvo en donde el elemento de prueba (esfera de <50 mm de diámetro) no debe entrar por completo
- **0:** No tiene protección contra el agua.

Como también observamos, con el indicador CE, nuestro equipo cumple con la conformidad europea, en materia de requisitos técnicos y legales.

3.2 Servomotor

Al igual que en el caso del servo accionamiento, el servomotor es de la marca Omron. Más concretamente, se nos ha permitido trabajar con el modelo Omron SGMAH-02AAA61D-OY. Como se ha comentado en apartados anteriores, el uso de un servomotor síncrono se debe a la necesidad de una alta precisión para la acción que vamos a desempeñar.

Tal y como sucede con el servo drive, el nombre que recibe el motor se debe a que es un código que nos informa de las características de este, para así poder escoger de un catálogo aquel que mejor se acople a las necesidades de la tarea a desempeñar.

A continuación, mostramos con más detalle el significado de los códigos:

SGMAH - 01 A 1 A 6 S D - OY

Sigma-II servo motor type
SGMAH: Super high power rate type
SGMPH: Cube type
SGMGH: High-speed feed type
SGMSH: Super high power rate type
SGMUH: High speed type

Capacity (kW)

Code	SGMAH 3000 min⁻¹	SGMPH 3000 min⁻¹	SGMGH 1500 min⁻¹	SGMSH 3000 min⁻¹	SGMUH 6000 min⁻¹
A3	0.03				
A5	0.05				
01	0.1	0.1			
02	0.2	0.2			
03	0.3				
04	0.4	0.4			
05			0.45		
06					
07	0.65				
08	0.75	0.75			
09			0.85		
10				1.0	1.0
12					
13			1.3		
15		1.5		1.5	1.5
20			1.8	2.0	
22					
30			2.9	3.0	3.0
32					
40				4.0	4.0
44			4.4		
50				5.0	
55			5.5		
60					
75			7.5		
1A			11		
1E			15		

Voltage
A: 230 V
D: 400 V

Connector specifications

Blank	No option
D	Hypertac connector (SGMAH,SGMPH)

Brake, oil seal specifications

1	No brake, no oil/dust seal
S	Oil seal
B	90 V brake
C	24 V brake
D	Oil seal + 90 DC brake
E	Oil seal + 24 VDC brake
F	Dust seal
G	Dust seal + 90 VDC brake
H	Dust seal + 24 VDC brake

Shaft end specifications

Code	Shaft end	Type SGMAH	SGMPH	SGMGH	SGMSH	SGMUH
2	Straight, no key	○	○	○	○	
4	Straight, key	○	○			
6	Straight, key, tapped	●	●	●	●	●
8	Straight, tapped	○	○			

●: Standard ○: Option

Design procedure:
A: Standard
E: SGMPH (IP67)
F: SGMAH (prepared for oil seal mounting)

Serial encoder specifications

Code	Encoder	Type SGMAH	SGMPH	SGMGH	SGMSH	SGMUH
1	16-bit absolute	○	○			
2	17-bit absolute			○	○	
A	13-bit incremental	●	●			
B	16-bit incremental	○	○			
C	17-bit incremental			●	●	●

●: Standard ○: Option

Ilustración 10. Explicación códigos servomotores de la serie SGMAH de Omron

Tras observar la información que no muestra la ilustración 8, podemos deducir las características de nuestro servomotor SGMAH-02AAA61D-OY:

- **SGMAH:** Servomotor Sigma-II de tipo muy alta potencia.
- **02:** Potencia de 0,2 kW.
- **A:** Tensión de 230 V.
- **A:** Encoder incremental de 13 bit de resolución.
- **A:** Procedimiento de diseño estándar.
- **6:** Especificación final del eje de tipo recto, chaveta y muñón.
- **1:** No freno.
- **D:** Conector de tipo Hypertac.

Después de analizar el código de nombre de nuestro servomotor, vamos a mostrar la placa de características de este y comentar la información que nos muestra esta.

Ilustración 11. Placa de características del servomotor

De la placa de características del servomotor podemos comentar que la potencia nominal del motor es de 200 W, que su corriente de salida es de 2,1 A y que debe alimentarse a una tensión de 200 V. Con respecto a su par nominal, podemos saber que es de 0,637 N·m y que la velocidad nominal de este será de 3000 rpm.

Con respecto estas dos últimas características, podemos comentar también la curva que las relaciones y el efecto que hay cuando se sobrepasa la velocidad nominal del servomotor.

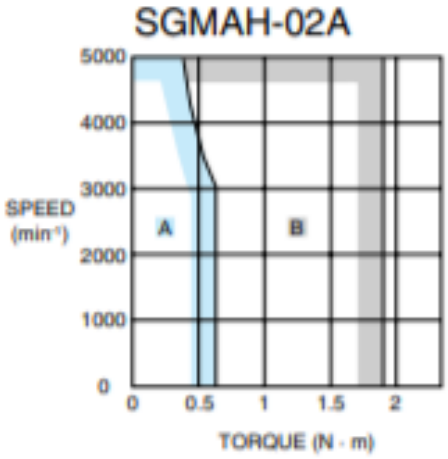


Ilustración 12. Curva característica Par-Velocidad de la serie SGMAH-02A

Fijándonos únicamente en la curva A de la ilustración 10, la cual llama el catálogo Omron como *"Zona de trabajo continuo"*, observamos que el par se mantiene siempre constante hasta la velocidad nominal, como se había comentado previamente, puesto que está es una característica de los servomotores. Sin embargo, cuando sobrepasamos esta velocidad nominal, el servomotor puede seguir trabajando, pero con una reducción del par que a medida que se va aumentando la velocidad. Observamos que la velocidad máxima de nuestro servomotor es de 5000 rpm y que cuando se llega a esta, el par llega a valor nulo. Por último, hacer también hincapié en el hecho que cuando la velocidad es nula, el par que tenemos es el nominal, característica que no se obtiene con todos los motores.

3.3 Autómata

Para nuestro proyecto, trabajamos con el Omron SYSMAC CJ1M CPU21. Las CPUs CJ1M son PLC con E/S incorporadas y de tamaño reducido, capaces de trabajar a alta velocidad. Incorpora la funcionalidad de refresco inmediato de las E/S en mitad de un ciclo cuando se ejecute una instrucción para ello.

Incorpora diez entradas que pueden trabajar con filtro para la eliminación de ruido que contamine la señal. Para ello, se debe modificar la constante de tiempo de entrada, que va desde los 0ms (sin filtro) a los 32ms.

Otra peculiaridad es la posibilidad de conectar un encoder rotativo a una entrada incorporada para admitir entradas de contador de alta velocidad.

Con respecto a las salidas, podemos contar con la generación de salidas de impulsos. La relación ON/OFF que es posible emitir desde las salidas permite llevar a cabo el posicionamiento o el control de velocidad con el uso de un servo controlador que sea capaz de aceptar las entradas de impulsos.

Es posible que estas salidas de impulsos, mediante su correcta configuración, que sean de sentido horario (CW), sentido anti horario (CCW) o salida de impulsos + dirección. Más adelante se comprobará que para nuestro caso se utilizó la opción de salida de horario y salida anti horaria. Además, el servo accionamiento SGDH-02AE-OY es capaz de trabajar con este tipo de

señales, las cuales podemos considerar como uno de los puntos más importante del presente proyecto.

Nos permite también la posibilidad de cambio de control durante una operación de trabajo, lo que podría permitir una colocación más exacta de una pieza en una estación o la búsqueda de origen para el inicio de un proceso de una forma más precisa. Sin embargo, otra incorporación que cuenta nuestro autómata y que está ligada con la idea de origen, es la opción que incorpora de búsqueda de origen. Si ejecutamos esta instrucción, la cual necesita el uso de varias E/S, como son los sensores límites de carrera, proximidad de origen, la señal de posicionamiento finalizado…, obtenemos nuestra posición de origen para comenzar a trabajar y a partir de ahí, podemos conocer en todo momento donde se encuentra la pieza de trabajo y con totalidad precisión.

Además, sobre la idea de este sistema de búsqueda de origen, se puede realizar la operación de vuelta al origen cuando finalizamos nuestra tarea, permitiendo volver al origen establecido.

Para la programación del autómata se hizo uso del programa CX-Programmer, el software de programación para toda la serie de autómatas de la empresa Omron. Más adelante, se explicará como se ha realizado la programación para que el autómata CJ1M CPU21 pudiera realizar la tarea que se requería para este proyecto.

El método de comunicación utilizado con el PLC es el *SYSMAC WAY,* el cual se trata de un sencillo formato de comunicaciones entre un ordenador y un PLC. Este protocolo de comunicación es propio de la empresa OMRON.

Este protocolo de comunicación puede utilizarse de varias maneras:

- **Local:** PLC conectado directamente al ordenador mediante el puerto serie.
- **Puente:** PLC en zona remota y la conexión es directa con un PLC el cual está conectado al PLC remoto por una red.
- **Modem:** PLC en zona remota y se hace uso de un modem telefónico para la conexión.
- **Modem Puente:** PLC en zona remota y se utiliza un modem telefónico para enlazar con la red donde se encuentra el PLC de destino.

La comunicación de autómata con PC la realizaremos con un cable tipo COM (RS232), desde el puerto serie para la transferencia del programa que debe ejecutar de forma cíclica al autómata. Por lo que podemos deducir que, en nuestro caso, la opción de comunicación es la primera que se indica previamente sobre el protocolo *SYSMAC WAY*.

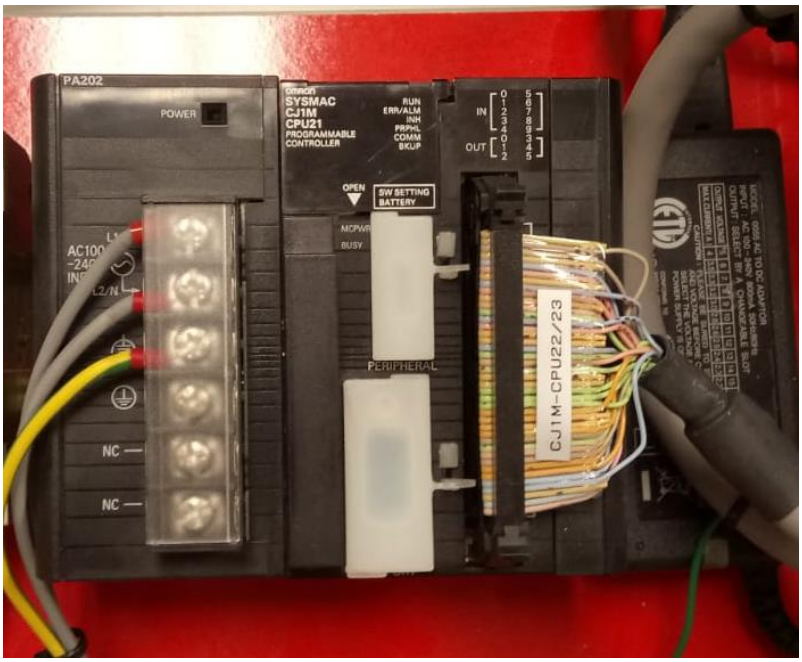


Ilustración 13. Autómata programable CJ1M CPU21 con su fuente de alimentación y el conector

3.4 Placa de conexiones

Para poder transmitir las señales desde al autómata CJ1 CPU21 al servo drive SGDH-02AE-OY, necesitaremos el uso de una placa de conexiones, la cual también será un producto de la empresa OMRON y en concreto se usa la placa de conexiones o interfaz pasiva XW2B-40J6-9A.

La peculiaridad de este tipo de placa es que permite la conexión de dos servos accionamientos a la vez, mandando a cada una el tren de pulsos correspondiente que mande el autómata programable. Este caso lo podríamos usar en cualquier máquina que trabaje con sistema de dos ejes. Para nuestra aplicación, únicamente utilizaremos una de las conexiones que ofrece este modelo de placa de conexiones.

A continuación, mostramos un esquema del método de conexión que tiene este tipo de equipos:

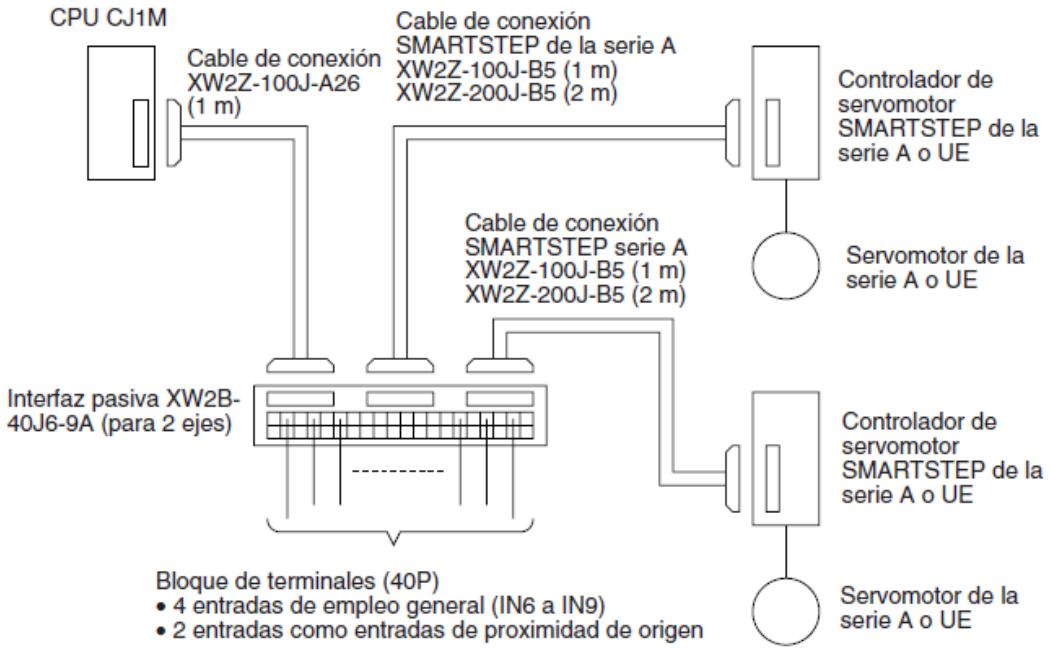


Ilustración 14. Conexión de placa de conexiones XW2B-40J6-9A con dos servos drivers

Para la conexión de los pines de la placa de conexiones con los distintos elementos, se usarán conectores de cable plano MIL como mostramos a continuación con las siguientes imágenes:

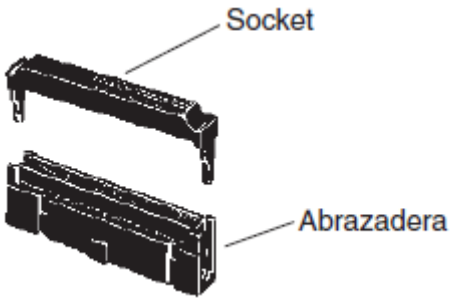


Ilustración 15. Dibujo del sistema de conectores de cable plano MIL

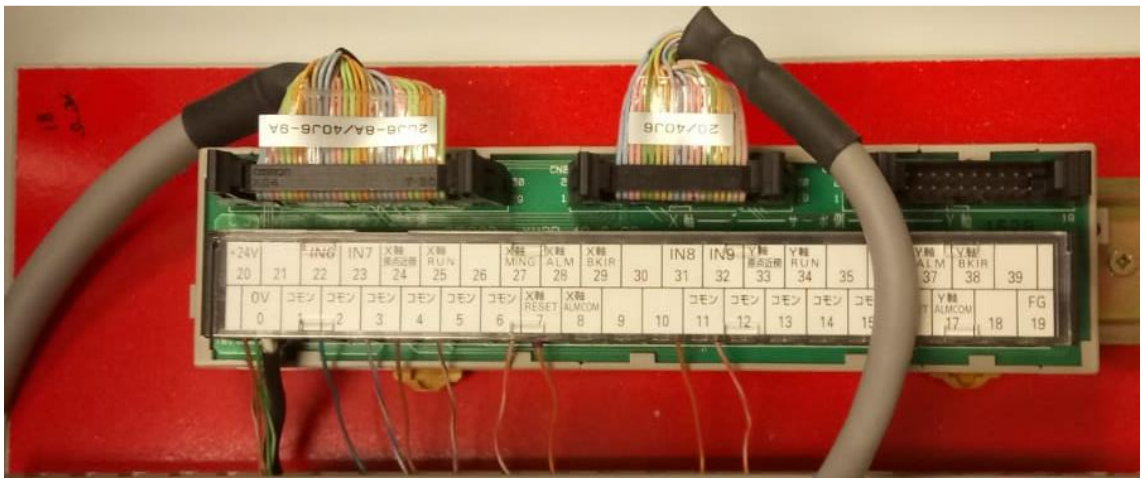


Ilustración 16. Detalle de placa de conexiones

Como vemos, la conexión de los sensores se realiza directamente a la interfaz pasiva, en donde van distintas señales como son los finales de carrera, la alimentación del servomotor, etc. Para ello, hay unas conexiones internas en esta placa, que son las que podemos observar en la próxima ilustración.

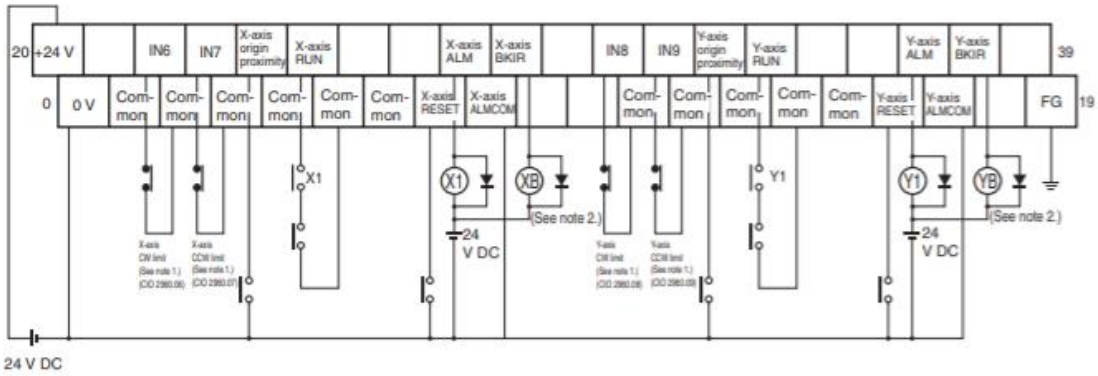


Ilustración 17. Esquema interno de la placa de conexiones XW2B-40J6-9A

4. Servicio de Prueba

En este apartado explicaremos el servicio de prueba para el correcto funcionamiento del tablero. Debemos tener en cuenta que estos equipos se han encontrado en desuso en un periodo de aproximadamente 5 años, por lo que habrá que hacer una primera revisión del estado en el que se encuentra y con la ayuda de distintos manuales, comprobar que todas las conexiones y funcionamientos son correctos.

- Cableado

A continuación, se muestra una imagen obtenida del manual de usuario de SGMAH/SGDH de la serie ∑-||, en la que marca las referencias de los componentes del servo-pack de nuestro equipo:

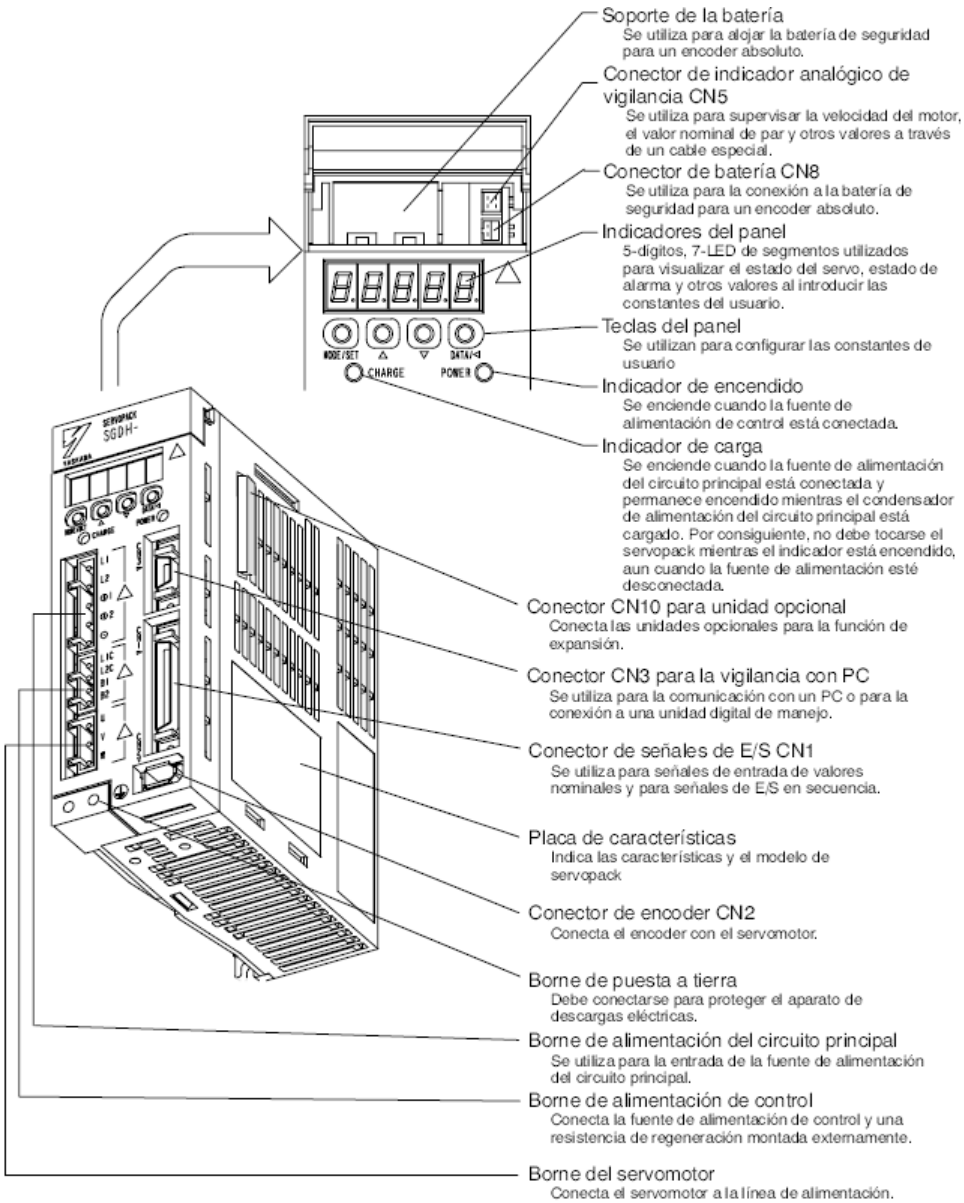


Ilustración 18. Referencias de componentes de servo-pack

Para nuestro caso, no se encuentran en uso el soporte de la batería, el CN5 ni el conector CN8.

A continuación, mostramos una imagen extraída del manual de usuario de SGMAH/SGDH de la serie ∑-||, en la que muestra el conexionado que debe tener el servo-pack a la hora de trabajar con los equipos periféricos:

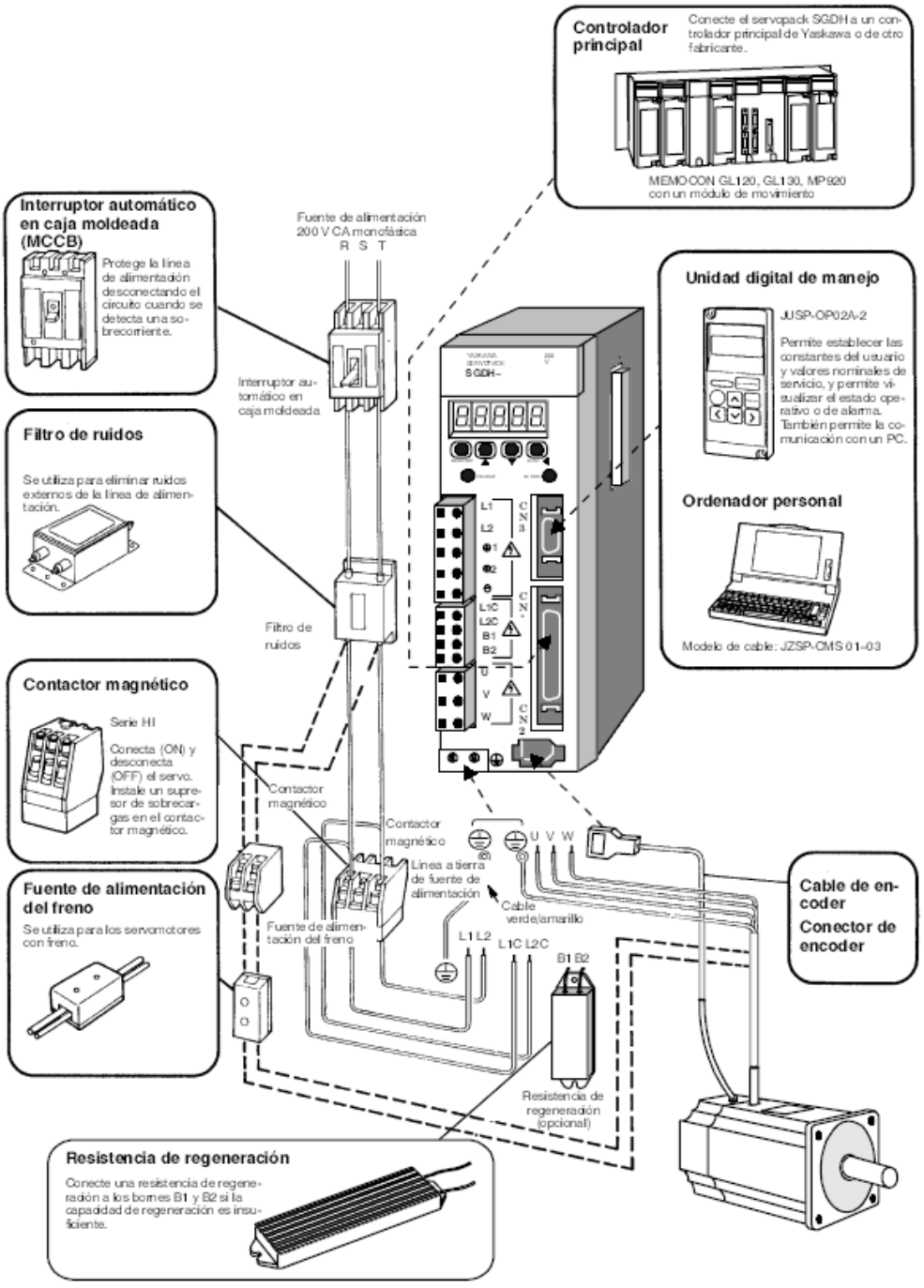


Ilustración 19. Ejemplo de conexionado obtenido desde el manual de usuario de SGMAH/SGDH de la serie ∑-||

Seguidamente mostramos una imagen del servo-pack y como se encuentra cableado:

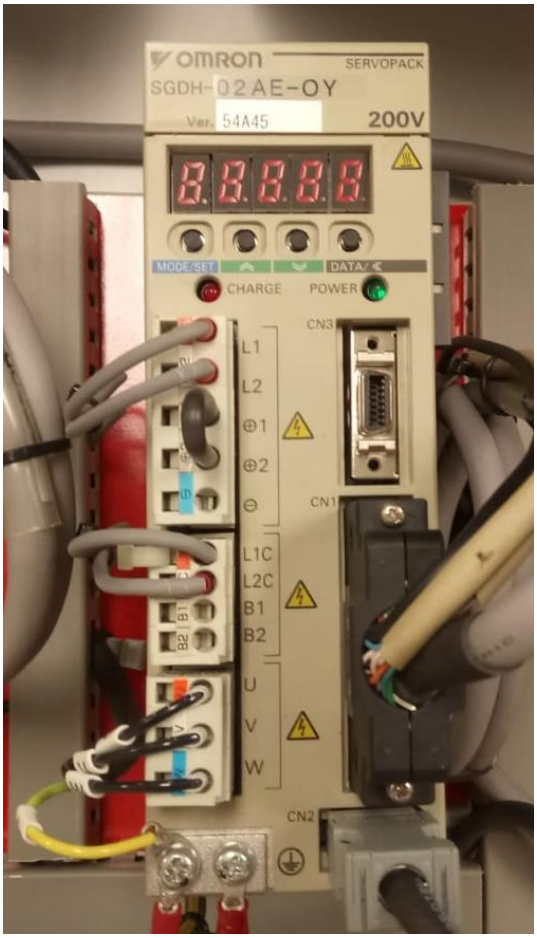


Ilustración 20. Servo-pack OMRON SGDH-02AE-OY

- Bornes del circuito principal:

Nombre	Símbolo	Tensión Circuito Principal (V)	Potencia Máxima admitida por el servomotor (kW)	Descripción	Estado
Borne de entrada de CA del circuito principal	L1,L2	200	De 0.03 a 0.4	Monofásico de 200 a 230 V CA +10 %, -15 % (50 / 60 Hz)	Conectado
Borne de conexión con el servomotor	U, V, W	-	-	Establece una conexión con el servomotor	Conectado

Borne de entrada de alimentación de control	L1C, L2C	200	De 0.03 a 15.0	Monofásico de 200 a 230 V CA +10 %, -15 % (50 / 60 Hz)	Conectado
Borne de pusta a tierra	⏚	-	-	Conectado con los bornes de puesta a tierra del motor y de la alimentación	Conectado
Borne de resistencia de regeneración externa	B1, B2	200	De 0.03 a 0.4	Normalmente no se conecta. Conecte una resistencia de regeneración si la capacidad de regeneración es insuficiente	No conectado
Conexión de la reactancia de CC para la contramedida de onda armónica de la fuente de alimentación	⊕1, ⊕2	200	De 0.03 a 0.5	Normalmente se realiza un cortocircuito entre estos bornes. Si es necesaria una contramedida para armónicos, conectar una reactancia de CC entre los bornes	Cortocircuitado

Tabla 2. Bornas de conexión del servo accionamiento

- Servicio de Prueba

Para la puesta en marcha del servomotor controlado por el servo-pack, se realizó el servicio de prueba que indica el manual.

Para la comprobación de las conexiones, se realizó el servicio de prueba que especifica el apartado 4 del manual de usuario de SGMAH/SGDH de la serie ∑-||. Se decidió realizar el servicio de prueba sin carga.

Primero se conectó el cableado de potencia y control al servomotor. Tras ello, y tal como indicaba el manual, se desconectó el conector del CN1 del servo-pack SGDH-02AE-OY, puesto que no era necesario su uso para la prueba. A continuación, se conectó la alimentación del tablero.

Tras la puesta en marcha, en el display del servo-pack pudimos observar que había dos señales, las cuales se alternaban durante periodos aproximados de 1 segundo. Estas dos señales eran “Pot” y “not”. Se verificó que está era la visualización normal que debía mostrar

el servo-pack. Al no aparecer la señal de alarma, se verificó que el circuito de alimentación era el correcto y se encontraba en buen estado.

A continuación, verificamos que el modelo del motor era el correcto. Esto lo debíamos hacer mediante el uso de la constante de usuario Fn011, como indica el manual. Para navegar en el servo-pack por sus distintas variables, se utiliza la unidad de manejo que incorpora esté. Tras ir a la constante Fn011, se mantiene durante 1 segundo el botón DATA/SHIFT para entrar en la constante. A continuación, indicamos los códigos que se observaron y los distintos significados que tienen estos según lo indicado en el manual de usuario en su apartado 7.2.6 "Comprobación del modelo de motor":

- F.0100: 01 indica la tensión con la que trabaja el servomotor. En este caso significa que la tensión es de 200 V CA ó 280 V CC.
  El 00 indica el modelo del servomotor, que en este caso es un SGMAH.
- P.0020: Indica la potencia x10. En este caso la potencia del motor era 200W.
- E.0013: 00 indicaba el tipo de encoder. En este caso era un encoder de tipo incremental. 13 nos muestra la resolución de este. Su resolución es de 13 bits.
- Y0000: Indica el orden de modificación del servo-pack. En este caso era el "Y00".

Para poder visualizar cada uno de los anteriores códigos dentro de la constante Fn011, debíamos apretar la tecla MODE/SET de la unidad de manejo de nuestro equipo. A continuación, mantenemos la tecla DATA/SHIFT durante 1 segundo y volvíamos a salir al menú que nos permite cambiar la constante de usuario que se necesita en cada momento.

El siguiente paso que se realizó fue la utilización del modo JOG que nos permite el servo-pack para comprobar el control del servomotor mediante el funcionamiento de la unidad de manejo que incorpora el equipo SGDH-02AE-OY. Para ello, se selecciona Fn002. Cuando entramos en este modo, observamos que el código que muestra el display del servo-pack es "-.-.JoG". Cuando pulsamos la tecla MODE/SET, configuramos el estado del motor como *Conectado*, y el display mostraba ".-.JoG". Así, observamos que al apretar las teclas que tienen las flechas arriba o abajo, el eje del servomotor gira en un sentido u otro. Con la flecha arriba, el sentido de giro es anti-horario y con la flecha abajo, el sentido es horario.

La velocidad de giro del servomotor en esta prueba es de 500 rpm, puesto que esta es la velocidad estándar que tiene el parámetro Pn304. Para comprobar el correcto funcionamiento, se decidió variar el valor de este parámetro a 250 rpm. Tras ello, volvimos a la constante Fn002 y pusimos de nuevo en marcha el modo JOG. Observamos que la velocidad de giro se había reducido y así confirmar que el estado de ambos equipos era el correcto. Para esta prueba, se siguió en todo momento las indicaciones del manual de usuario de los equipos que disponemos de la marca OMRON.

A continuación, se conectaron las líneas de señales al CN1 del servopack, para seguir con el servicio de prueba del servomotor sin carga.

El siguiente paso consistía en la comprobación del cableado de señales de entrada mediante el uso del modo de vigilancia que tiene el servopack. Para ello, debemos ir al parámetro Un005. Conectamos y desconectamos cada una de las señales mediante la botonera física y observamos cómo cambian el estado de los LED de vigilancia del panel. Seguidamente,

adjuntamos una captura del manual de usuario de SGMAH/SGDH de la serie ∑-||, en la que nos indicará que variable corresponde a cada uno de los indicadores LED que conforman el display en este modo:

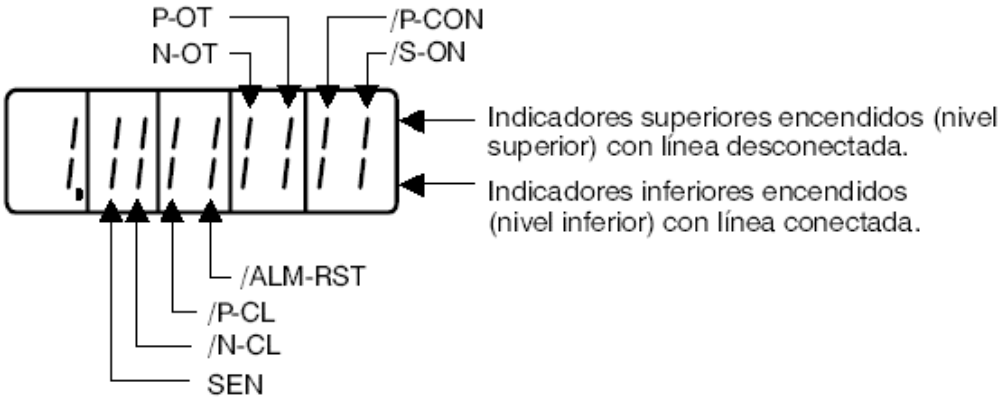


Ilustración 21. Captura manual de los indicadores LED de señales de entrada

Observamos que cada uno de los indicadores cambian de estado según el accionamiento de la botonera para cada uno de los parámetros, a excepción de la señal SEN, la cual no está cableada con la botonera física.

El siguiente punto del servicio de prueba sin carga, era el procedimiento operativo del en el modo de control de velocidad. Para ello, lo que nos indica el manual es que debemos configurar el parámetro Pn000.1 como 0, para asa estar en modo de control de velocidad. Lo que debíamos hacer era mandar una señal de tensión de referencia (V-REF, CN1-5). Para ello, hicimos uso de potenciómetro. Para visualizar la velocidad real el servomotor, debemos ir al parámetro Un000 del modo de vigilancia del servo-pack. Para conocer la velocidad nominal de entrada que recibe el servomotor, vamos a la constante Un001.

Para observar el funcionamiento de nuestro equipo, activamos la señal S-ON mediante la botonera física. Según la posición del potenciómetro, la señal que observamos en el parámetro Un001, puede ser de valor positivo o negativo. Esto nos indica el sentido de giro que lleva el motor. Para accionar el sentido horario, debemos tener la referencia de velocidad que nos transmite el potenciómetro en valor positivo y además, tener activo la señal P-OT. Para que el eje gire en sentido contrario, la señal que recibe de tensión de referencia nuestro servomotor, tendrá que tener un símbolo negativo, y a la vez, tener accionado la señal N-OT. Observamos que nuestro servomotor gira en ambos sentidos sin ningún problema.

En caso de cambiar la referencia de señal del sentido de giro, debemos modificar el parámetro Pn000.0 . Mientras que si queremos modificar la amplificación de la señal de entrada de velocidad nominal, debemos ir a la constante Pn300. Para nuestra prueba de servicio, los valores fueron:

- Pn000.0: 0.
- Pn300: 1000.

Tras la prueba del control de velocidad, a continuación, se realizó la de control de posición. Para ello, lo primero que tuvimos que modificar fue el valor de la constante Pn000.1 por el valor 1. Cuando cambiamos este parámetro, debemos de volver a alimentar el servo-pack para que funcione el nuevo tipo de control que hemos seleccionado.

Para el funcionamiento en modo de control de posición, deberemos introducir un impulso de velocidad lenta. Para el servicio de pruebo, el PULS lo mandamos desde la propia botonera, alternando el estado de esta entrada. En todo momento, debemos tener activa la señal S-ON para el funcionamiento del servomotor. Otra entrada que debe recibir el servo-pack para este funcionamiento, es la señal SIGN, con la cual indicaremos el sentido de giro que deseamos. De nuevo, para el servicio de prueba, esta señal la mandamos desde la botonera. En caso de querer cambiar la referencia de sentido de giro, deberíamos cambiar el valor de la constante Pn000.0, la cual puede ser de valor 0 ó 1.

Para el caso de la relación electrónica, y así poder cambiar el ángulo de giro por pulso que realiza el servomotor, debemos modificar los parámetros Pn202 y Pn203. En el caso de la prueba de servicio, se realizó con los valores predeterminados que tiene estas constantes, los cuales son:

- Pn202: 4.
- Pn203: 1.

Con esto, observamos un movimiento muy reducido por parte del servomotor. La configuración de estos dos parámetros para el correcto funcionamiento de la bancada, se detallará más adelante, puesto que se debe calcular la relación que hay entre la distancia que se quiere recorrer por cada pulso que se le quiere mandar al servomotor.

Con esto, finaliza el servicio de prueba sin carga del servo-pack SGDH-02AE-OY y el servomotor SGMAH-02AAA61D-OY.

5. Configuración servo drive

En el presente apartado explicaremos la configuración de las distintas variables del servo drive OMRON SGDH-02AE-OY para su correcto funcionamiento para la aplicación de línea de montaje que se desea simular como objetivo en este proyecto fin de grado.

Debemos tener en cuenta que una correcta configuración es indispensable, puesto que las aplicaciones en donde se puede llevar a cabo el uso de estos equipos son muy extensa, tanto el tipo de control que se quisiera realizar (velocidad, par, posición o posible combinación entre ellos).

Lo primero que vamos a mostrar es una tabla con las señales de usuario que tiene el servo accionamiento, en donde se especifica, por bloques, el tipo de variable que es cada uno. La información de la tabla ha sido extraída del manual del servo accionamiento.

Tipo	**N° de constante de usuario**	**Descripción**
Constante de selección de función	De Pn000 a Pn003	Selecciona las funciones básicas y de aplicación como el tipo de control o el modelo de detención en caso de alarma
Amplificación de servo y otras constantes	De Pn100 a Pn123	Establecen valores numéricos como las ganancias de bucle de posición y de velocidad
Constantes de control de posición	De Pn200 a Pn208	Establecen los parámetros de control de posición como el formato de entrada del impulso nominal y la relación de transmisión
Constantes de control de velocidad	De Pn300 a Pn308	Establecen los parámetros de control de velocidad como la amplificación de entrada nominal y el tiempo de desaceleración de arranque suave
Constantes de control de par	De Pn400 a Pn409	Establece los parámetros de control de par como la amplificación nominal de par y los límites de par a izquierdas y derechas
Constantes de secuencia	De Pn500 a Pn512	Establecen las condiciones de salida para todas las señales de secuencia y los cambios de asignación y selección de señales E/S
Otras	De Pn600 a Pn601	Especifican la capacidad de una resistencia de

		regeneración externa y constantes reservadas
Ejecución de funciones auxiliares	De Fn000 a Fn014	Ejecutan funciones auxiliares como el funcionamiento en el modo paso a paso
Modos de vigilancia	De Un000 a Un00D	Permiten supervisar los valores nominales del par y la velocidad, así como comprobar el estado de la E/S

Tabla 3. Señales de usuario del servo accionamiento

Como podemos observar con el último caso de las constantes de modo vigilancia, la codificación de estas variables es de tipo hexadecimal.

No todas las variables han sido modificadas, y a continuación iremos indicando los cambios que se han realizado y el motivo de porque han tenido que recibir un valor distinto al que tienen en estado predeterminado.

5.1 Selección de función

Como se ha visto previamente, en este subapartado se tratarán aquellas variables que van desde la Pn000 a la Pn003. Únicamente haremos hincapié en aquellas que han debido de ser cambiadas.

- Pn000.0: Selección del sentido.

Si esta variable tiene el valor 0, la rotación a derechas se define como rotación antihoraria (CCW).

Si, por el contrario, el valor de esta es 1, la rotación a derechas sería especificada como rotación con sentido horario (CW).

Cuando hablamos de la variable Pn000.0, nos referimos a que debemos modificar dentro de Pn000, el 0 que se encuentra más hacia a la derecha y asignarle el valor que deseemos. Si habláramos del valor de Pn000.1, nos referíamos al 0 que se encuentra en medio y Pn000.2, el de la izquierda del todo. Esto nos permite gran posibilidad de configuraciones y sin la necesidad de crear un gran número de variables.

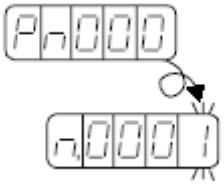


Ilustración 22. Explicación modificación de variable Pn000.0

Para nuestro caso, el valor que le dimos a la variable fue el de 0.

- Pn000.1: Selección de modo de control.

Para nuestro caso, como necesitábamos un control de posición para nuestra aplicación, el valor que debíamos asignarle a esta variable debía ser el "1". Sin embargo, otra opción que tenemos es la posibilidad de cambiar los tipos de controles con la modificación del estado del botón P-CON.

A continuación, mostramos una tabla especificando los distintos valores que puede tener esta variable y a su vez, los distintos controles que nos permite el servo driver.

La información que se muestra a continuación ha sido extraída directamente del manual de usuario del variador Omron.

Configuración Pn000.1	Modo de control
0	Control de velocidad (valor nominal analógico)
1	Control de posición (valor nominal de secuencia de impulsos)
2	Control de par (valor nominal analógico)
3	Selección de control de velocidad mediante entrada de contactos (referencia de contactos)
4	Selección de velocidad mediante entrada de contactos (referencia de contactos) ↔ Control de velocidad (valor nominal analógico)
5	Selección de velocidad mediante entrada de contactos (referencia de contactos) ↔ Control de posición (valor nominal de secuencia de impulsos)
6	Selección de control de velocidad mediante entrada por contactos (referencia de contacto) ↔ Control de par (valor nominal analógico)
7	Control de posición (valor nominal de secuencia de impulsos) ↔ Control de velocidad (valor nominal analógico)
8	Control de posición (valor nominal de secuencia de impulsos) ↔ Control de par (valor nominal analógico)
9	Control de par (valor nominal analógico) ↔ Control de velocidad (valor nominal analógico)
A	Control de velocidad (valor nominal analógico) ↔ Control de bloqueo en cero
B	Control de posición (valor nominal de secuencia de impulsos) ↔ Control de posición (inhibición)

Tabla 4. Modos de control a configurar en la constante de usuario Pn000.1

De los distintos modos de control con los que nos permite trabajar el servo accionamiento SGDH-02AE-OY, debemos seleccionar aquellos que nos permitan un control de posición. En nuestro caso seleccionaremos el de control de posición único (Pn000.1 = 1), pero debemos comentar que cuando se realiza el control mediante la botonera que incorpora nuestro tablero, la utilización de la configuración en valor 7, nos permite mover el servomotor con uno de los potenciómetros en caso de que sea necesario y según el valor que le estemos dando a la entrada analógica de la velocidad (según el signo de la señal que estemos mandando), nos permitirá que

el giro del motor sea de sentido horario o antihorario. A su vez, según la posición en el que se encuentre el contacto denominado P-CON, podemos seleccionar el control de posición o velocidad. Esta opción es interesante en el caso de emergencia, en donde es necesario el movimiento del motor mediante el control de un operario de una forma inmediata por posible daño de la instalación, de la pieza que se está manipulando o que alguna persona se encuentre en una situación de riesgo y se debe cambiar la posición de la plataforma de una forma inmediata.

- Pn002.2: Uso de encoder absoluto.

En esta variable seleccionaremos como deseamos que trabaje el encoder que se encuentra en el servomotor, si de forma absoluta o de forma incremental. La diferencia entre trabajar de una forma u otra es el hecho que, al trabajar de forma absoluta, siempre tendremos el mismo punto 0 y conoceremos la posición exacta de donde se encuentre nuestra pieza, aunque tengamos, por ejemplo, un corte de alimentación, debido a que a la codificación del encoder. Si trabajamos de forma incremental, la posición 0 se selecciona arbitrariamente y esta puede modificarse, por lo que, en caso de finalizar la tarea de trabajo o corte de alimentación, no sabemos exactamente donde se encuentra la pieza deberíamos seleccionar nosotros mismo el punto 0 al inicio de cada jornada para continuar el trabajo, lo que nos llevaría una pérdida de tiempo que de la otra forma se podría evitar.

Por tanto, el valor que debemos darle a esta variable será “0” para trabajar de forma absoluta. En caso de trabajar de forma incremental, simplemente le daríamos el valor de “1”.

5.2 Control de posición

En este subapartado, veremos aquellas variables que deberán ser modificadas para el correcto funcionamiento de nuestra aplicación. Como hemos observado previamente, las variables que están relacionadas con el control de posición son aquellas que se encuentran entre los valores Pn200 a Pn208. De nuevo, haremos únicamente hincapié en aquellas que debieron ser modificadas y el motivo de este cambio.

- Pn200.0: Formato de impulsos nominal.

La parametrización de estas variables nos permitirá seleccionar como deseamos que sea la entrada de impulsos que recibe el variador, que posteriormente será enviado al servomotor para los movimientos que debe realizar este último. Debemos tener en cuenta que las señales que se mandan al variador provienen de los canales que se emiten desde la placa de conexiones y que esta emite tanto del autómata o la botonera.

Este parámetro nos permite que las señales de tren de pulsos se manden desde el mismo canal (CN1-7) y que sea otro canal que, según su estado, 0 o 1, indique el sentido en el que debe rotar el servomotor (CN1-11). Esta configuración no es la que en nuestro caso deseamos, puesto que el funcionamiento del sistema se realizará desde autómata.

Por tanto, para nuestro trabajo necesitaremos que dos trenes de impulsos, uno que será para que la plataforma gire en sentido horario, la cual vendrá desde el canal CN1-11, mientras que

en sentido antihorario se mandará desde el canal CN1-7. Así, según la señal de salida que mande el autómata, conseguiremos una rotación hacia un sentido o hacia otro.

A continuación, mostramos las explicaciones que se dan en el manual de usuario del servo drive sobre el parámetro Pn200.0:

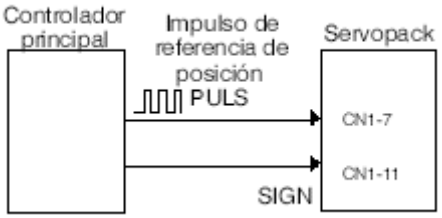


Ilustración 23. Funcionamiento básico del tren de pulsos

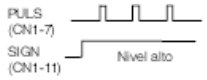
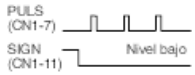
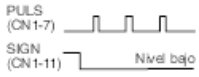
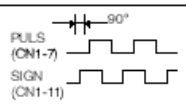
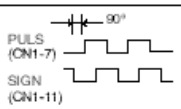
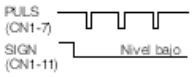
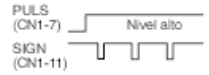
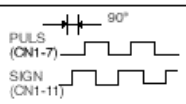
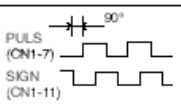
Constante de usuario Pn200.0	Formato de impulso nominal	Multiplicador de impulsos de entrada	Operador lógico	Valor nominal de rotación a derechas	Valor nominal de rotación a izquierdas
0	Secuencia de impulsos signo +	–	Operador lógico positivo		
1	Impulso en el sentido del reloj + impulso en el sentido opuesto al reloj	–			
2	Secuencia de impulsos bifásica con diferencial de fase 90°	×1			
3		×2			
4		×4			
5	Secuencia de impulsos signo +	–	Operador lógico negativo		
6	Impulso en el sentido del reloj + impulso en el sentido opuesto al reloj	–			
7	Secuencia de impulsos bifásica con diferencial de fase 90°	×1			
8		×2			
9		×4			

Ilustración 24. Distintas parametrizaciones de la variable Pn200.0

Tras lo que hemos observado en sobre este parámetro y las necesidades de nuestro proyecto, la configuración que debe tener Pn200.0 es el "1", para poder tener los dos trenes de pulsos independientes como hemos comentado previamente, siendo uno el que haga que el servomotor gire en sentido horario y el otro tren en sentido opuesto.

- Pn201: Divisor de PG.

Este parámetro establece la cantidad de impulsos para las señales de salida PG enviadas externamente.

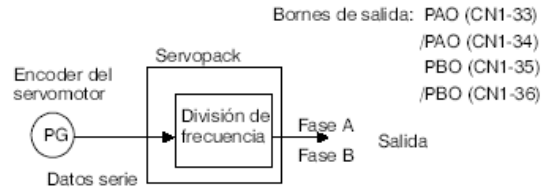


Ilustración 25. Explicación del divisor de PG

Los impulsos del encoder del servomotor (PG) se dividen por el número predeterminado de impulsos antes de enviarlos a la salida.

El número de impulsos de salida por revolución se configura con esta constante. Esto lo medimos mediante impulsos/revolución.

A continuación, mostramos los distintos valores que se asocian según el encoder y servomotor con el que trabajemos.

Especificaciones de encoder y modelo de servomotor	**Resolución en bits**	**Número de impulsos de encoder por revolución (imp/rev)**	**Rango de configuración**
A	13	2048	De 16 a 2048
B, 1	16	16384	De 16 a 16384
C, 2	17		

Tabla 5. Valores para configurar el parámetro Pn201 según servomotor de trabajo y necesidad

Para esta constante de usuario debemos tener en cuenta para que se lleve a cabo cualquier cambio que se realice, deberá ir acompañado de una realimentación.

En nuestro caso, al disponer de un servomotor SGMAH-02AAA61D-OY, el encoder es de resolución de 13 bits, por tanto, el valor que le debemos asignar de imp/rev se encontrará en el intervalo de 16 a 2048. En caso de superar el valor máximo que se le puede asignar para este tipo de motor, el variador automáticamente detectará que esto no se cumple, otorgando un valor automático de 2048. Con esta opción que tiene el servo pack evitamos posibles errores que se comentan a la hora de la configuración debidos a posibles despistes o por desconocimiento con el tipo máquinas con la que se están trabajando.

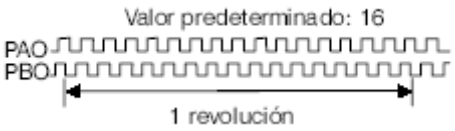


Ilustración 26. Ejemplo de configuración

- Pn202: Relación de transmisión electrónica (numerador).
- Pn203: Relación de transmisión electrónica (denominador).

Estos dos parámetros están directamente relacionados, puesto que estos valores con los cuales conseguimos de forma ajustada que el servomotor desplace la plataforma que tenemos en el eje lineal para nuestro caso.

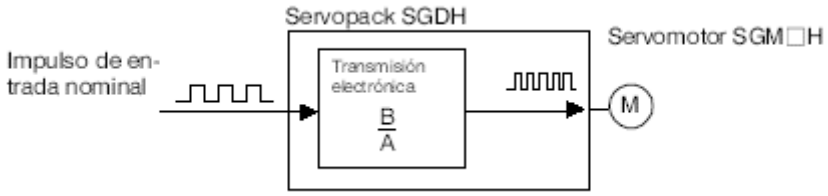


Ilustración 27. Explicación de funcionamiento de la transmisión electrónica

Como observamos en la ilustración 24, el Pn202 lo conocemos como "B", según el manual. Mientras, el parámetro Pn203 lo denomina "A".

Debemos saber que los valores que se le puede poner a cada uno de estos parámetros van de 1 a 65535 y que debemos tener especial cuidado en el valor que les otorgamos, puesto que en caso de que la división que se realiza para indicar el paso que debe realizar el servomotor sea muy elevado, podemos provocar que el variador de frecuencia bloquee la conexión con el motor por riesgo de "overspeed" y entrar en modo protección.

$$Electronic\ gear\ ratio\ \left(\frac{B}{A}\right) = \frac{Pn202}{Pn203}$$

$$B = [(número\ de\ impulsos\ ) \cdot 4] \cdot [velocidad\ del\ motor]$$

$$A = [Unidades\ de\ valor\ nominal\ (distancia\ de\ trayecto\ por\ revolución\ de\ árbol\ de\ carga)] \cdot [velocidad\ de\ revoluciones\ de\ carga\ del\ árbol]$$

Las tres anteriores ecuaciones que se muestran se han extraído directamente del manual del servo accionamiento.

Debemos tener en cuenta que en ningún caso el valor que se le otorgue al parámetro "A" (Pn203) puede ser superior al valor que se le dote al parámetro "B" (Pn203).

Parámetro	Rango de configuración	Configuración básica de fábrica
Pn202	De 1 a 65535	4
Pn203	De 1 a 65535	1

Tabla 6. Valores para parametrizar las constantes Pn202 y Pn203

Las variables que hemos vistos para trabajar en control de posición están todas relacionadas, tal y como vamos a observar en el próximo diagrama de bloques, en donde se muestra los puntos de unión que tienen y como es el funcionamiento de este tipo de control. La imagen que se muestra en la siguiente página ha sido extraída directamente del manual de usuario del servo pack de Omron.

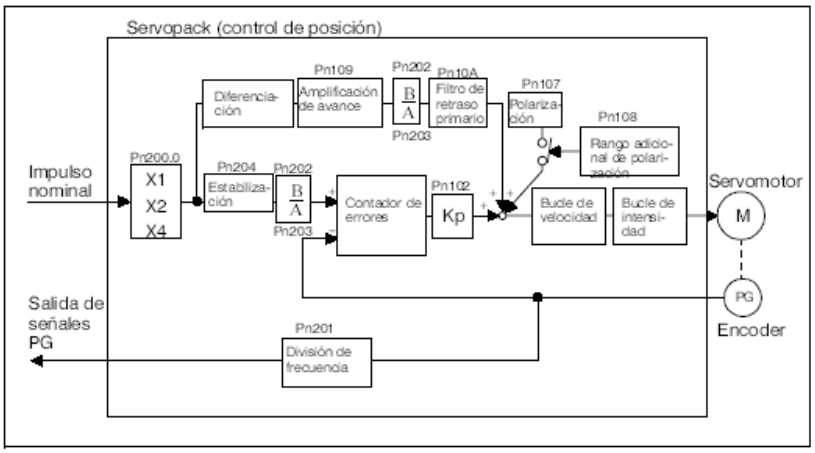


Ilustración 28. Diagrama de bloques de control para el control de posición

6. Programación Autómata

En el presente apartado del proyecto explicaremos la programación del autómata con el que trabajamos, el Omron SYSMAC CJ1M CPU21. Como bien se ha comentado en apartados previos, la herramienta a utilizar es el software de programación CX-Programmer en su versión 7.2. Dicha herramienta de programación es de propiedad de Omron para la programación de sus productos.

Debido a que para el uso de este programa es necesario una licencia de pago, el departamento de accionamientos nos permitió el uso de sus ordenadores del aula, los cuales disponen de dicho programa.

La idea es que el programa realice una primera búsqueda del origen, acción que la serie CJ1M puede realizar con el uso de una acción que podemos configurar desde el CX-Programmer, mandar un tren de pulsos con unos pulsos fijos, pues suponemos que las distancias que hay entre estación y estación son las mismas, temporizadores simular que la pieza se encuentra en una estación con el tiempo necesario para la realización del trabajo que necesite la materia prima, contadores para saber en todo momento en que zona se encuentra la pieza y otro para conocer el número de piezas que llevamos y una acción de vuelta a origen continuar con la siguiente pieza o volver a empezar con la siguiente tira de piezas.

Por último, se añadirá las acciones necesarias para un control manual de tipo JOG, para poder trabajar con nuestro sistema en un modo manual, donde se podrá elegir el sentido de giro del motor y la velocidad que queremos a la que vaya este.

A continuación, desarrollaremos este apartado explicando como se trabaja en el CX-Programmer, como configuramos distintas variables que se necesitaban para nuestra aplicación y el proyecto que se realizó para nuestra aplicación.

6.1 Inicio CX-Programmer

Como cualquier otro programa instalado en una máquina con sistema operativo Windows, el acceso al programa y apertura de este es simplemente buscarlo en el menú de herramientas de Windows. Sin embargo, para nuestro caso, hicimos uso del acceso directo que disponía el ordenador del aula del departamento que se nos prestó para el desarrollo de este trabajo.

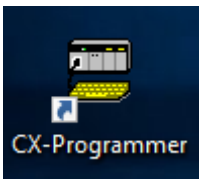


Ilustración 29. Icono CX-Programmer

Tras abrir el programa, nos encontramos con que debemos crear un programa nuevo. Para ello, en la esquina de arriba izquierda, seleccionamos el icono de un folio para crear un nuevo proyecto. Seguidamente, se nos abrirá una ventana en la que nos indicará el autómata con el que se quiere trabajar y el tipo de comunicación que se va a emplear.

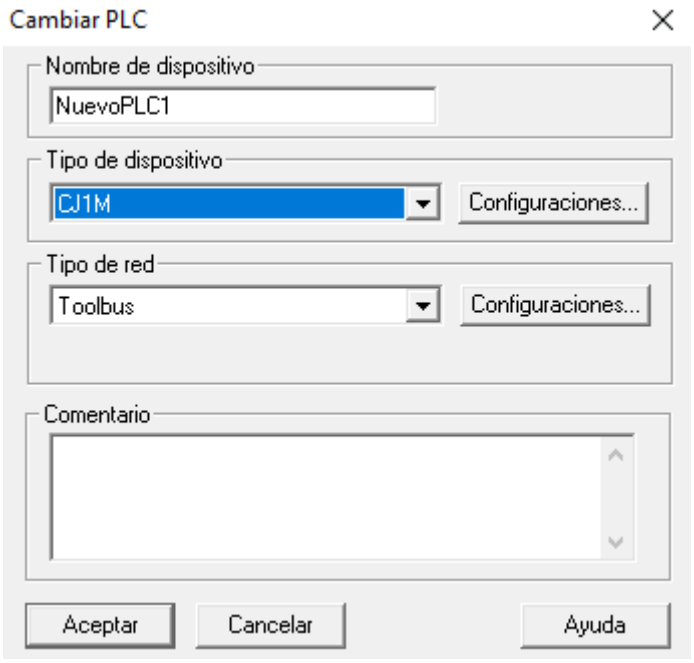


Ilustración 30. Ventana de selección de dispositivo y tipo de red

Como observamos en la figura 29, y tal como es en nuestro caso, seleccionamos el tipo de dispositivo con el cual contamos. Puesto que en nuestro caso es un autómata Omron SYSMAC CJ1M CPU21, en el tipo de dispositivo seleccionamos "CJ1M". Para el caso de escoger la CPU, debemos entrar en la opción de configuraciones que hay en tipo de dispositivo, y a continuación se nos abrirá la siguiente ventana en donde elegiremos la opción de CPU21:

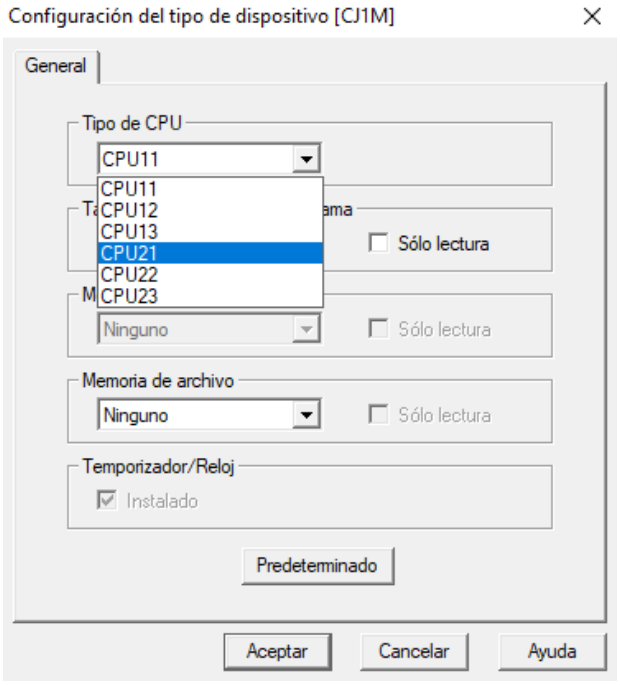


Ilustración 31. Selección de CPU del autómata

Tras esto, como hemos comentado con anterioridad, debemos seleccionar el tipo de comunicación. Para nuestra aplicación seleccionaremos el protocolo SYSMAC WAY, el cual es un tipo de comunicación de propiedad de Omron.

El SYSMAC WAY se caracteriza por ser un protocolo sencillo entre un ordenador y un PLC. La conexión se puede realizar mediante una interfaz RS232, en el caso de que utilicemos un PLC individual, y en caso de que utilicemos múltiples PLC para la tarea que se quiera realizar, será necesario el RS422.

Los usos que se puede dar a este protocolo de comunicación van desde un uso local, cuando se encuentran el PC y el PLC conectados directamente desde el puerto de la computadora. Otra situación es uso de forma puente, en donde al igual que en el caso del local, se encuentran un PC y un PLC conectados por un puerto. Sin embargo, la diferencia es que en este caso se encuentra en la existencia de otro PLC remoto que está conectado mediante una red al PLC que se ha mencionado anteriormente.

También se podría usar en tipo modem, en donde el PLC se encuentran en una ubicación remota y para su conexión se puede utilizar con un modem telefónico.

Por último, se encuentra la opción de trabajar en forma modem puente, en donde se encuentra un PLC remoto y se utiliza un modem para la comunicación de datos mediante una red en donde está el PLC destino.

Para nuestro caso, utilizaremos este protocolo de forma local, pero podemos tener en cuenta el uso de este protocolo para en caso de su utilización en otras aplicaciones en donde la ubicación de los elementos se encuentre en lugares distintos. Además de este protocolo, también se encuentran el tipo Ethernet, Modbus, Toolbus, etc.

Por tanto, la ventana de configuración de inicio para nuestro proyecto deberá ser la que se muestra a continuación en la ilustración 31:

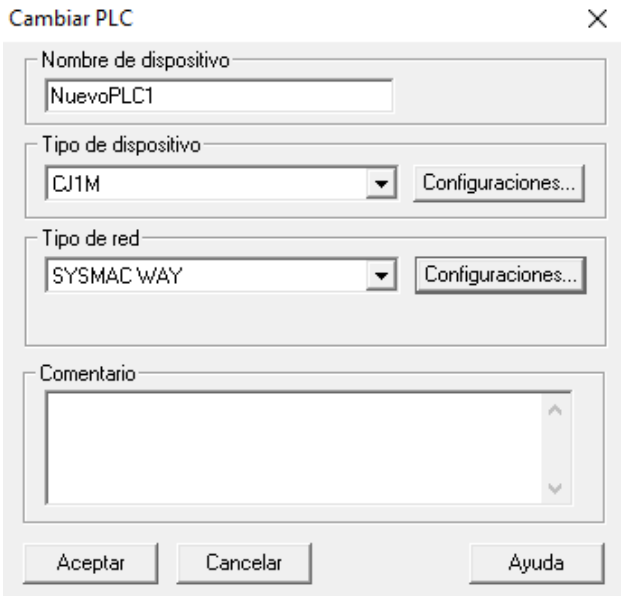


Ilustración 32. Configuración inicial del programa finalizada

6.2 Programación

Tras realizar la configuración inicial, ya nos encontramos en disposición de crear nuestro programa para que el autómata lo realice de forma cíclica.

Lo primero es conocer el entorno con el que trabajamos en CX-Supervisor y el tipo de programación con la que vamos a trabajar. Sobre esta última idea, he de comentar que nuestro trabajo se realizará en "Ladder" o diagrama de contactos o lenguaje en escalera.

El lenguaje de tipo Ladder es aquel que se basa en los esquemas de control de conmutación clásicos y que siguen una lógica matemática. Este tipo de lenguaje se encuentra estandarizado en la IEC 61131-3. Se trata de un tipo de lenguaje que podemos considerar "sencillo". La CPU lee de izquierda a derecha y de arriba a abajo. Por tanto, las entradas o contactos se encuentran a la izquierda del diagrama, mientras que las salidas o bobinas son aquellas que podemos observar en la derecha de este tipo de lenguaje.

Cuando hemos comentado ante que el control de conmutación sigue una lógica matemática nos referimos a que responde a si se encuentra en "1" o estado activo, o si por lo contrario la señal que reciben es un "0" o estado inactivo.

Para nuestro proyecto, a la hora de la programación, hemos utilizado los contactos clásicos de este tipo de lenguaje, que son los contactos normalmente abiertos, cerrados y la utilización de algún flanco positivo o de subida. Estos son los que se encuentran a la izquierda cuando realizamos la programación y, por tanto, son las entradas que queremos y necesitamos.

Ilustración 33. Tipos de contactos en lenguaje ladder

- Contacto normalmente abierto: Su estado se encuentra en 0 cuando está abierto y cuando este se encuentra activo, el valor que recibe es 1.
- Contacto normalmente cerrado: Se encuentra en cortocircuito cuando su valor es 0 y, por tanto, esta entrada se encuentra activa. Cuando su estado lógico cambia a 1, la entrada se encuentra en circuito abierto. La utilidad de este tipo de contactos es para la detección de fallos, como es el caso de los relés térmicos.
- Flanco positivo: Se activa en el momento que el estado lógico de un contacto pasa de 0 a 1.
- Flanco negativo: Se activa en el momento que el estado lógico de un contacto pasa de 1 a 0.

Después de observar los tipos de entradas que tenemos, debemos comentar un idea básica que habla sobre la ubicación y combinación que podemos realizar con las entradas, puesto que la activación de una salida, en la gran mayoría de casos, no viene únicamente con la activación exclusiva de una entrada, sino que puede venir por la activación de dos entradas o más a la vez,

o por la activación de alguna entrada que la programemos. A la primera idea es lo que conocemos como "AND" y a la segunda como "OR".

Como ya hemos comentado, en el caso del "AND", deben encontrarse todos los contactos activos para poder activar la salida. Se representarán las salidas en la misma fila y al final de la línea, se encontrará la salida que se quiere activar.

Ilustración 34. Ejemplo contactos en combinación "AND"

Por el otro lado, nos encontramos con la combinación "OR", en donde se representan las entradas en posición vertical entre ellas. Para este caso, cualquiera de las entradas que se encuentre en cortocircuito, activará la salida a las que se encuentran conectadas.

Ilustración 35. Ejemplo contactos en combinación "OR"

También, existe la posibilidad de combinar contactos en estado "AND" y en estado "OR" según lo que queramos que realice el PLC.

Tras la información mostrada sobre las entradas básicas que nos encontramos, ahora debemos hablar un poco de las salidas o bobinas que encontramos en la programación por diagrama de contactos.

Al igual que en el caso de las entradas, nos encontramos con cuatro tipos de bobinas o salidas. Estas se encuentran a la derecha a la hora de realizar el programa y son las salidas que va a tener nuestro programa.

Ilustración 36. Tipos de bobinas en lenguaje ladder

- Bobina normalmente abierta: Se encuentra normalmente en estado 0 y cuando se active pasará a estado lógico de 1.

- Bobina normalmente cerrada: Se encuentra normalmente en estado 1 y cuando se activa pasa a encontrarse en estado 0.
- Enclavamiento de bobina o Set: Al activarse pondrá a nivel lógico 1 la bobina y aunque las condiciones de entrada cambien, la bobina se quedará en este este estado.
- Reset de bobina: Al activarse pondrá a nivel lógico 0 la bobina y aunque las condiciones de entrada cambien, la bobina se quedará en este estado.

Después de la explicación dada sobre las entradas y salidas que existente, comentar que también usaremos funciones básicas que se pueden utilizar a la hora de la programación de autómatas con cualquier software a parte del CX-Programmer, como puede ser el TIA Portal de la empresa Siemens. Estas funciones a las que nos referimos son temporizadores (TON, TOFF), contadores, SR (Set-Reset), etc. La utilización de estos en nuestro programa se explicará más adelante.

Además de estas acciones, se escogió este tipo de PLC por incorporar una función en especial que nos es indispensable para la labor que queremos realizar. La acción a la que nos estamos refiriendo es la búsqueda de origen. Como hemos comentado en el párrafo anterior, esta se explicará más adelante y con más detalle, incluyendo la configuración de funcionamiento seleccionada de esta función.

A continuación, comenzamos la explicación del entorno y uso del programa CX-Programmer y como se ha realizado el programa que utilizaremos para nuestra aplicación.

Tras realizar la primera configuración y principal para la realización del programa, observamos en la siguiente imagen, como está diseñado este programa:

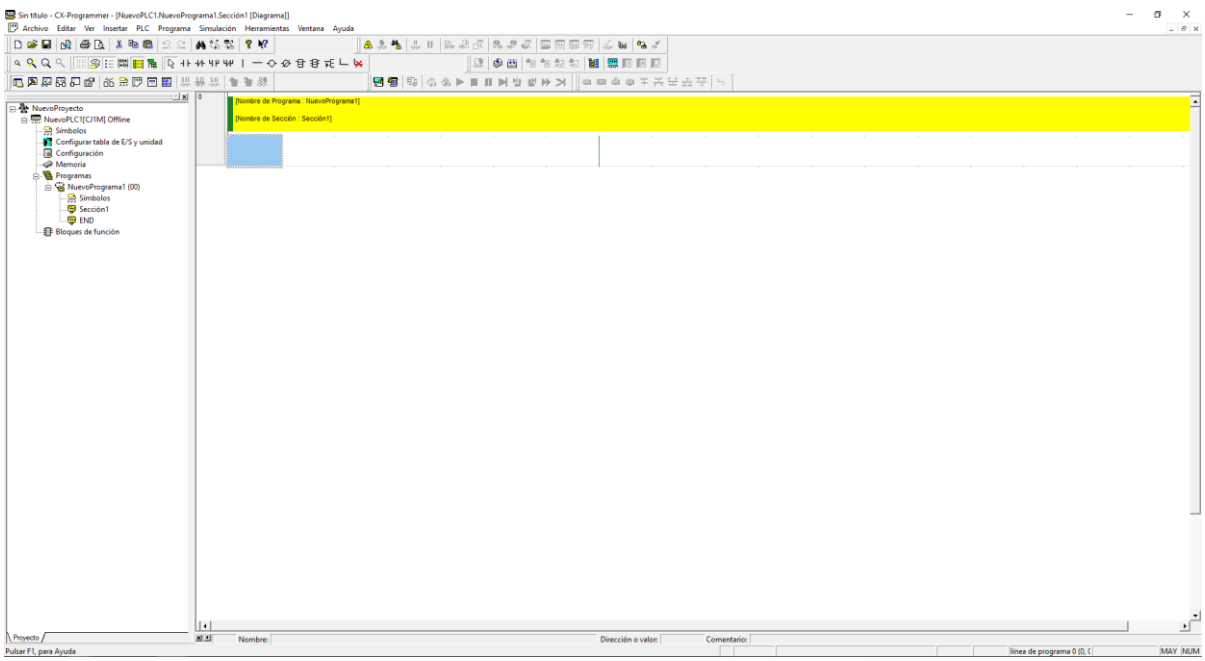


Ilustración 37. Entorno del CX-Programmer

Como podemos apreciar, la zona en la que diseñaremos nuestro programa es la que se encuentra debajo de la línea horizontal de color amarillo, todo lo que observamos en blanco. El rectángulo de color azul que observamos es en donde colocaremos el elemento que necesitaremos.

Para la colocación de cualquier elemento, podemos realizarla de varias formas. La más sencillas que nos encontramos es la de clicar en la barra superior de herramientas el elemento que queremos añadir y seleccionar la casilla en donde queremos colocarlo. Por otro lado, y de forma más rápida, podemos usar el teclado del ordenador, puesto que cada elemento tiene asignada un carácter para esta forma de trabajo, como son:

- Contacto: “C”
- Contacto cerrado: “/”
- Línea vertical: “Ctrl + Abajo”
- Línea horizontal: “Ctrl + Derecha”
- Bobina: “O”
- Bobina cerrada: ”Q”
- Instrucción: “I”

Por tanto, y tal como hemos explicado con anterioridad, a la izquierda de la página de programación tendremos los contactos y a la derecha tanto las bobinas como las instrucciones que necesitemos en cada fila. No es necesario colocar todas las líneas que conectan la entrada y la salida, simplemente con colocar una y haciendo doble click con el botón izquierdo del ratón, el programa automáticamente interpreta que esa línea del programa se ha finalizado y coloca todas las líneas en las casillas para conectar la parte izquierda con la derecha sin la necesidad de que el usuario tenga que escribir por sí mismo todas las líneas desde un lado y otro. De esta forma, conseguimos reducir tiempo.

Ahora, explicaremos las instrucciones que utilizamos para que el sistema realice el trabajo que queremos que este realice. Para ello, hicimos usos de las siguientes funciones:

- KEEP (011) o SR: Esta instrucción básica hace que todo lo que coloquemos que ataca desde la línea que va a la parte superior del bloque active la variable a la que está asociada esta función y lo que va a la parte inferior de esta, desactive o resetee la variable.
  La utilización de esta instrucción se lleva a cabo para activar la variable que inicia la búsqueda del origen.

- TIM o TON: Para este caso, el temporizador comenzará a contar cuando la entrada a la que se encuentre asociada se active. Pasado el tiempo a la que el temporizador este programado, activará la salida a la que se asocie. Debemos tener en cuenta que en el CX-Programmer debemos asignar un número a cada temporizador, puesto que la variable que activa será un contacto con nombre T0 y Número del temporizador.
  Usamos los temporizadores para el inicio, pues queremos que no se ponga en funcionamiento el servomotor nada más encuentra el origen, sino que espere un pequeño tiempo antes de ponerse en marcha. También lo utilizamos para el tiempo en el que se debe quedar la pieza en cada etapa.

- CNT o Contador Decreciente: Cada vez que entra la entrada a la que se encuentra conectado el contador, resta una unidad al valor máximo que se le coloca al contador. Como en el caso de los temporizadores, cada contador tendrá un número, para así poder utilizar el contacto al que está asociado. El nombre será C0 y Número del contador. Esto hace que cada vez haya un valor actual distinto y así poder conocer la pieza de trabajo que es la que se encuentra en producción y donde se encuentra esta, puesto que también lo utilizamos para conocer la etapa en la que se encuentra la pieza.

- Comparadores: Se usará cuando se cumpla la lógica matemática cuando comparamos dos valores.
  Para nuestro caso, hicimos uso de un comparador de inferioridad, para que cada vez que la etapa actual fuera inferior que el número de etapas a realizar, sumáramos uno al valor de etapa actual y así conocer en qué etapa se encuentra la pieza.
  Usamos también igualadores, para poder cambiar el tiempo de etapa según sea el valor de etapa actual, así podemos seleccionar independientemente el tiempo de etapa que queramos para cada una.
  Por último, usamos también un comparador de superioridad, para que, en caso de seleccionar una velocidad demasiado elevada, nos active una señal de error en la pantalla y también pare el sistema al instante para evitar posibles daños.

- MOV(21) o Move: Esta acción la utilizamos para cambiar el valor de alguna memoria. Cuando ataca la entrada a este bloque, automáticamente el valor de la variable al que este asociado cambia por el que hayamos preindicado.
  Esta instrucción la utilizamos en el caso que hemos comentado anteriormente de los tiempos de etapa, para que el tiempo de etapa actual corresponda con el tiempo de etapa según la etapa en la que nos encontremos.
  También la utilizaremos para poner a valor unitario el número de etapa cuando volvamos al origen tras haber realizado un ciclo y también en el caso de poner a la unidad el valor de pieza actual, tras haber completado el número de piezas que se debían realizar.

Estas son las acciones básicas que realizamos con nuestro programa. Además de estas, también hemos usado un número de instrucciones especiales que tiene nuestro PLC y el CX-Programmer, las cuales son fundamentales para el correcto funcionamiento de nuestra aplicación. Y son las que vamos a comentar seguidamente.
Como hemos comentado, la utilización de estos equipos nos permite el uso de la instrucción de búsqueda de origen, algo indispensable pues para la aplicación que debemos simular será siempre la primera acción que se deberá llevar a cabo antes de comenzar con la producción, pues debemos suponer que la maquinaría al llegar, puede encontrarse la plataforma en donde se ubican las piezas de trabajo en cualquier tramo de la cinta y que la ubicación de las estaciones de trabajo se encuentran siempre a una distancia con respecto este punto de origen.
Deberemos de diferenciar dos modos de funcionamiento de esta acción, pues en caso de configurarla con el número 0 en el apartado de selección de modos de esta instrucción, realizaremos una búsqueda al origen, ubicando donde se encuentra este. Sin embargo, en caso de configurar la selección de modo con el número 1, se realiza una vuelta al origen, que

utilizamos siempre que terminamos de trabajar con una pieza y para trabajar con la siguiente o cuando hemos terminado definitivamente y hacemos que vuelva la plataforma a su sitio.

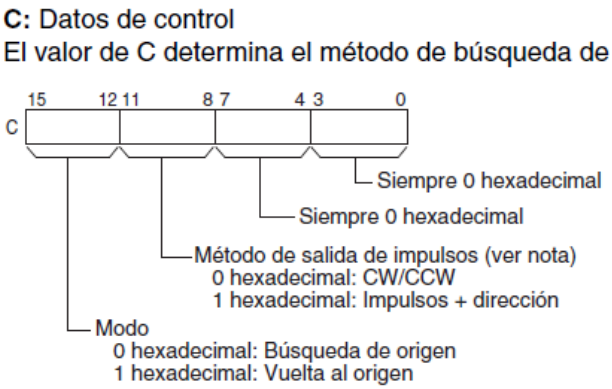


Ilustración 38. Datos de control para configurar la instrucción ORG (889)

Ahora trataremos con más detalle la búsqueda de origen, pues el Omron SYSMAC CJ1M CPU21 nos permite seleccionar entre distintas formas para la obtención del punto origen. Estas distintas formas se diferencian en cómo queremos que interprete el autómata la detección de la búsqueda de origen y también como queremos que actúe en caso de detectar un límite de carrera.

Cuando hablamos de la interpretación que queremos que realice el autómata cuando detecta la señal de proximidad, nos referimos así el servomotor se debe desacelerar cuando la señal de proximidad pasa de estar en estado activo a inactivo y entonces establecer el origen, o si, por lo contrario, queremos que sea el punto 0 cuando esta se encuentra en estado activo. Esto se lleva a cabo con la selección de la inversión de señal del origen. También podemos trabajar sin la utilización de la señal de proximidad al origen y trabajar directamente con el sensor de origen.

Por otro lado, también tenemos la posibilidad de seleccionar como queremos que actúe el servomotor en caso de llegar a uno de los finales de carrera. Lo primero es configurar en qué sentido queremos que gire el motor al principio. En caso de detectar un límite de carrera, podemos hacer que esta señal haga que el sentido de la marcha se cambie, para así seguir buscando el origen, o, por otro lado, que cuando se detecte esta señal de final de carrera, el motor se frene automáticamente.

Para nuestro caso, haremos uso del método de inversión 1, el cual como hemos explicado, es aquel que no tiene la necesidad de un sensor de proximidad al origen que provoque una desaceleración en el servomotor. Con respecto a la acción que debe llevar a cabo nuestro sistema en caso de una detección de final de carrera, el motor haremos que cambie su sentido de giro para poder obtener el punto de origen sin necesidad de un paro en nuestra actividad de trabajo durante la búsqueda del origen.

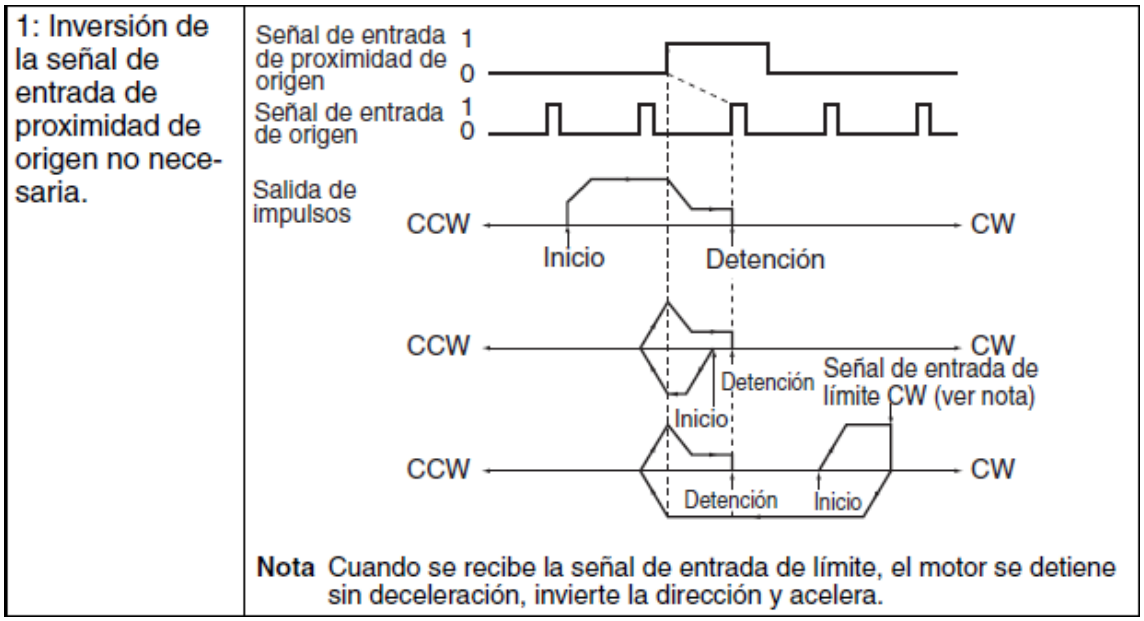


Ilustración 39. Explicación método seleccionado de búsqueda de origen

Para seleccionar el método de origen, en CX-Programmer, debemos ir a la izquierda de la pantalla, en donde encontramos la ventana que podemos observar la opción de "Configuración". Cuando esta se selecciona, automáticamente se nos abrirá una nueva ventana en donde deberemos dirigirnos directamente a la opción de "Pulse Output 0", salida de pulsos 0, que es nuestro caso la única que vamos a utilizar, pese a que la tarjeta pasiva nos permite utilizar dos trenes de pulsos, pero para nuestra aplicación, únicamente nos hace falta un tren de pulsos porque trabajamos en un único eje.

A continuación, mostramos una imagen obtenida directamente del programa en donde mostramos la configuración que hemos seleccionado para el correcto funcionamiento de nuestro proyecto:

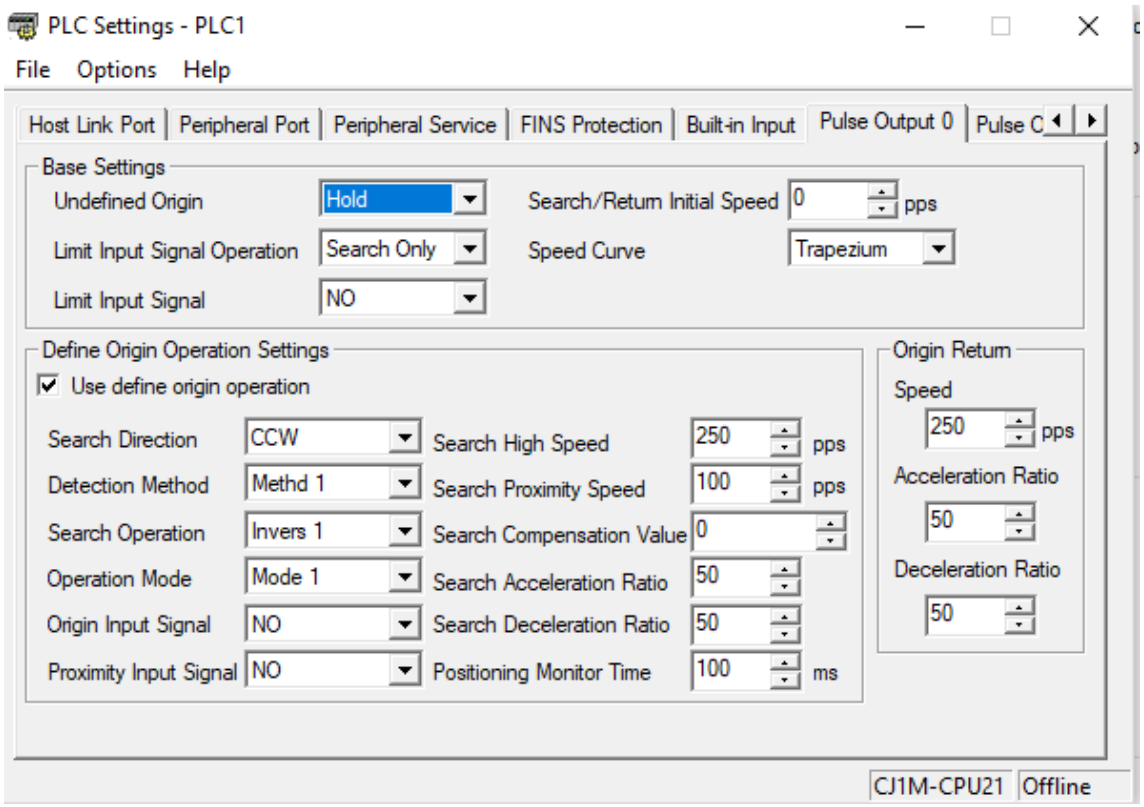


Ilustración 40. Configuración de búsqueda de origen en CX-Programmer

Como podemos observar en la figura 39, debemos también configurar valores como son las velocidades, medidas en pps (pulsos por segundo). Debemos tener en cuenta la naturaleza de los sensores que vamos a utilizar, que como observamos, todos ellos son normalmente abiertos (NO). Además, desde esta ventana también es en donde configuramos los valores que incumben a la instrucción de vuelta al origen, en la parte derecha de la imagen.

Tras comentar la utilización de la función ORG (889), comentaremos la instrucción de tren de pulsos que utilizamos a la hora de la programación de nuestro proyecto.

Deberemos tener en cuenta que la instrucción de tren de pulsos, PULS (886), debe ir acompañada de la instrucción de velocidad, SPED (885), o por la de aceleración que en el CX-Programmer es designada con el nombre ACC (888). La decisión de acompañar el tren de pulsos con una u otra será según las necesidades que tengamos, si vemos por ejemplo que nuestra aplicación es necesario ser precavidos por la posibilidad de dañar la pieza que se transporta y que no debe haber movimientos bruscos en la línea de montaje, se acompañaría el tren de pulsos con aceleración.

Para nuestro caso, trabajaremos con la instrucción de pulsos en forma relativa, pues consideramos que la distancia que hay entre estaciones siempre es la misma y como este tipo de funcionamiento hace que cada vez se vaya sumando los pulsos que son necesarios, después la función de vuelta al origen guarda esta cantidad para volver al punto de partida de todo el proceso para seguir con el trabajo.

Con respecto a la elección escogida de la instrucción que acompaña al tren de pulsos, consideramos la necesidad de que las piezas pueden considerarse como elementos frágiles y que los movimientos bruscos que pudiera provocar la ausencia de aceleración dañarían el elemento de manipulación. Además, trabajará como configuración independiente el bloque de aceleración, pues como indica el manual de usuaria, este modo hace que la función termine cuando se termine el tren de pulsos y para inmediatamente el accionamiento del motor. Sin embargo, en caso de estar configurado en modo continua, hasta que no haya una entrada que pare el tren de pulsos, el motor sigue en funcionamiento hasta la orden de paro.

Por último, hay que comentar que la utilización del bloque de velocidad se lleva a cabo cuando trabajamos de forma manual en el modo JOG, pero en este caso sin el acompañamiento de un tren de pulsos, pues en este caso nosotros accionamos el motor indicando a la velocidad que queremos que el motor gire. Este es otro modo de uso de SPED (885).

Para poder utilizar todas las instrucciones, debemos de hacer uso de distintas memorias para transmitir, por ejemplo, el número de pulsos que queremos que se lleven a cabo entre estación y estación para trasladar la pieza.

Las memorias en CX-Programmer se manipulan en la ventana de “memorias”, en la parte izquierda de la pantalla y en donde se nos abrirá una nueva ventana. Para nuestro caso, utilizamos principalmente las memorias denominadas “D”, en donde colocábamos los valores en formato hexadecimal.

Por otro lado, tenemos la ventana de símbolos, en la cual definimos las direcciones de las entradas y salidas, direcciones auxiliares que necesitamos para algunas acciones, como puede ser la activación de errores según la negligencia que haya ocurrido durante el tiempo de trabajo. Como en el caso anterior, esto se encuentra en la parte izquierda de la pantalla y seleccionamos la opción de “símbolos”, la que abarca todo el programa.

Para finalizar, debemos mandar el programa al autómata. Para ello debemos ir a la parte superior de la ventana de herramientas y seleccionar el símbolo con forma de triángulo y de color amarillo, que simula las señales de alta tensión. Tras esto, un poco más hacia la derecha, encontramos la opción de transferir programa a PLC. La seleccionamos y comenzará la transmisión de programa al autómata. Tras indicarnos el programa que se ha transferido todo correctamente, estamos en situación de poder trabajar. Se recomienda que trabajemos en modo de supervisión, el cual activaremos desde el símbolo de unas gafas, pues esto nos permite visualizar como se encuentra el funcionamiento del programa, como cambian de estado cada variable e incluso modificarlas desde el propio ordenador.

Con esto, terminamos el apartado de programación del autómata. El programa entero se encuentra en el anexo del presente proyecto, junto a la dirección de las distintas memorias que se han utilizado para poder diseñar un programa de trabajo adecuado.

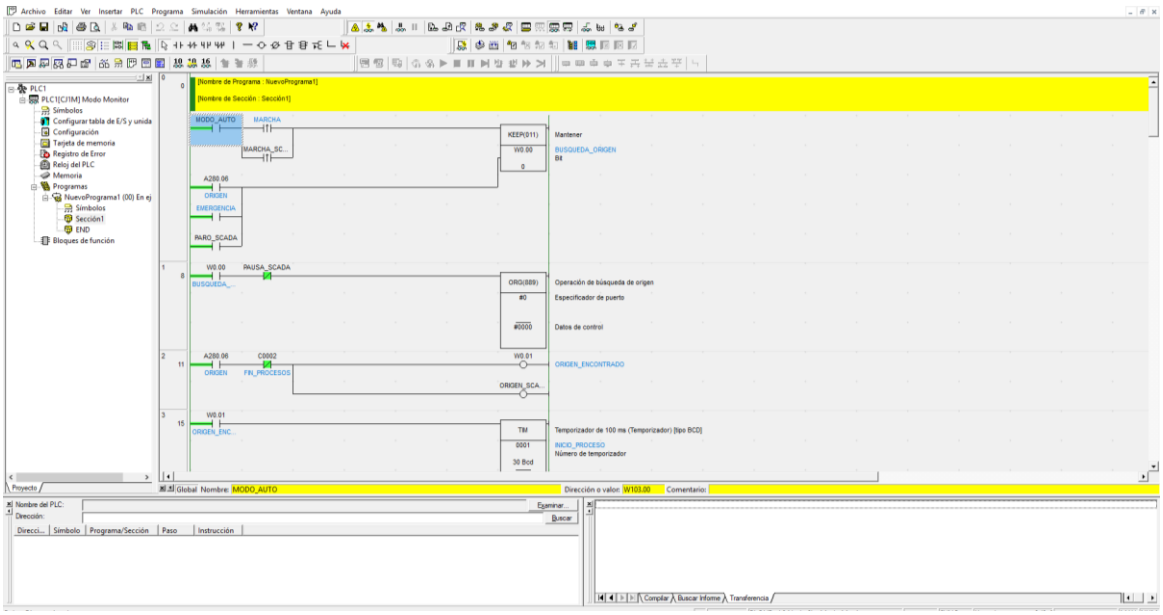


Ilustración 41. Ilustración con el programa en funcionamiento durante una prueba

7. Realización de Scada

En el presente apartado, trataremos las ideas que están relacionadas con la creación y diseño de una interfaz scada. Se debe tener en consideración que la importancia que tiene la creación de un scada es muy alta, pues será, por lo general, la herramienta que utilizarán los operarios en el día a día y en donde podrán monitorizar todo el funcionamiento de la línea de montaje. Dentro de lo que conocemos como automatización industrial, esta herramienta de trabajo se ubica en el nivel de supervisión.

Los sistemas Scada (Supervisory Control and Data Acquisition) son aplicaciones software de control de la producción, que se comunican con los distintos dispositivos de campo y con el que se controla desde la computadora o pantalla de operador, el proceso de forma automática.

Para el diseño de una pantalla, existen varios programas que nos permiten la creación de estas. En muchos casos, los programas son propiedad de la misma empresa que ha fabricado el autómata que hemos trabajado, como puede ser el TIA Portal de la empresa Siemens. Sin embargo, y como ya conocemos, en nuestro proyecto hemos empleado equipos de la empresa Omron y, por lo tanto, haremos uso de su programa de diseño, el CX-Supervisor.

Como se ha mencionado previamente, la herramienta scada nos permite realizar las siguientes acciones:

- Adquisición de datos: Procesamiento, recogida y almacenamiento de la información.
- Supervisión: Supervisar desde un monitor todas las variables de control.
- Control: Modificar la evolución del proceso, bien sobre los reguladores, consignas, etc. o directamente sobre las salidas.

Seguidamente, vamos a desarrollar la realización de nuestros scada, pues uno de los objetivos que teníamos a cumplir era crear un scada que nos permitiera controlar el sistema sin ser de la empresa Omron y que fuera gratuito el programa en el que lo diseñamos. Pero primero explicaremos la creación del scada realizado con la herramienta CX-Supervisor.

7.1 Scada en CX-Supervisor

En esta parte del apartado siete, trataremos las ideas que engloban la creación de una pantalla de monitorización con la herramienta de diseño CX-Supervisor. El scada que vamos a crear trabajará de forma local, es decir, teniendo que estar conectado directamente el autómata de control de la línea de mecanizado con la computadora en la que los operarios podrán controlar y visualizar los datos del trabajo que se está llevando a cabo. Al igual que con el CX-Programmer, para la creación de la pantalla, hicimos uso de uno de los ordenadores de la aula del departamento, el cual disponía del programa de diseño.

Abrimos el programa y nos encontramos que debemos crear es un nuevo proyecto. Para ello, debemos seleccionar en la parte superior izquierda de la pantalla, el botón donde podemos leer “archivo”. Tras esto, se nos abrirá una ventana en donde indicaremos el nombre del proyecto y la dirección en donde queremos que se guarde, algo básico en la mayoría de los programas que funcionan en un sistema operativo Windows y que no lleva una gran dificultad.

A continuación, tras realizar la creación del proyecto, volvemos a seleccionar el botón de “archivo” y esta vez, seleccionamos “nueva página”. Automáticamente en el entorno del CX-

Supervisor, se nos mostrará una venta con fondo gris que en realidad es la base de nuestra futura pantalla de control y en la cual trabajaremos. Es de mencionar, que también se pueden crear varias pantallas en un mismo proyecto, para tareas que necesites más de una pantalla de visualización de datos, aunque no es el caso en el presente proyecto. Como ya hemos comentado, el color predeterminado de la base de la pantalla es el gris, pero en caso de querer cambiar este, deberemos volver a seleccionar el botón de "archivo" y entrar en la opción de "propiedades de página…". Todo esto se lleva a cabo siempre que se encuentre preseleccionada la venta de la página en donde vamos a diseñar, no confundir con la selección predeterminada del programa. En propiedades de página, a parte de modificar el fondo de la pantalla, podemos darle un nombre a la pantalla, lo que nos resultará más sencillo a la hora de trabaja con ella en caso de trabajar con varias pantallas a la vez. También aquí le podemos dar el tamaño que queramos y donde queremos que aparezca la pantalla cuando la estemos utilizando. En nuestro caso, la opción más lógica era crear una pantalla que ocupara todo el monitor.

Después de realizar estos primeros ajustes de la base de la pantalla, nos encontramos en la situación de poder crear un punto o variable. Para ello, deberemos ir a la parte superior de la ventana de herramientas de CX-Supervisor y seleccionar el botón de "editor de puntos", y a continuación se nos abrirá una ventana en la que podremos crear los puntos que necesitemos.

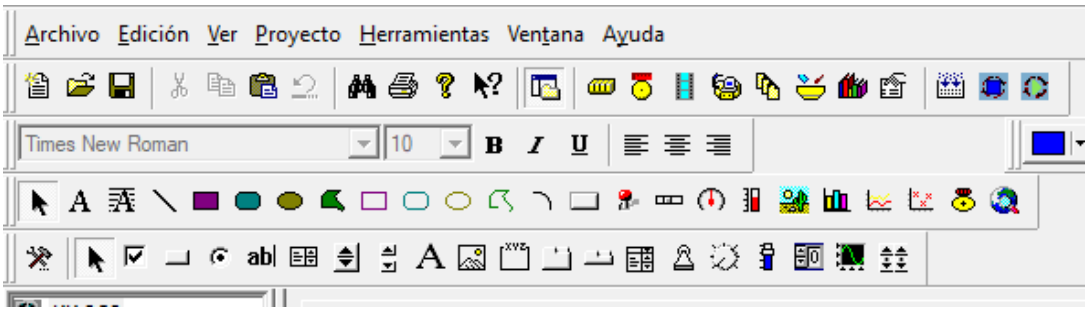


Ilustración 42. Vista de la barra de herramientas del CX-Supervisor

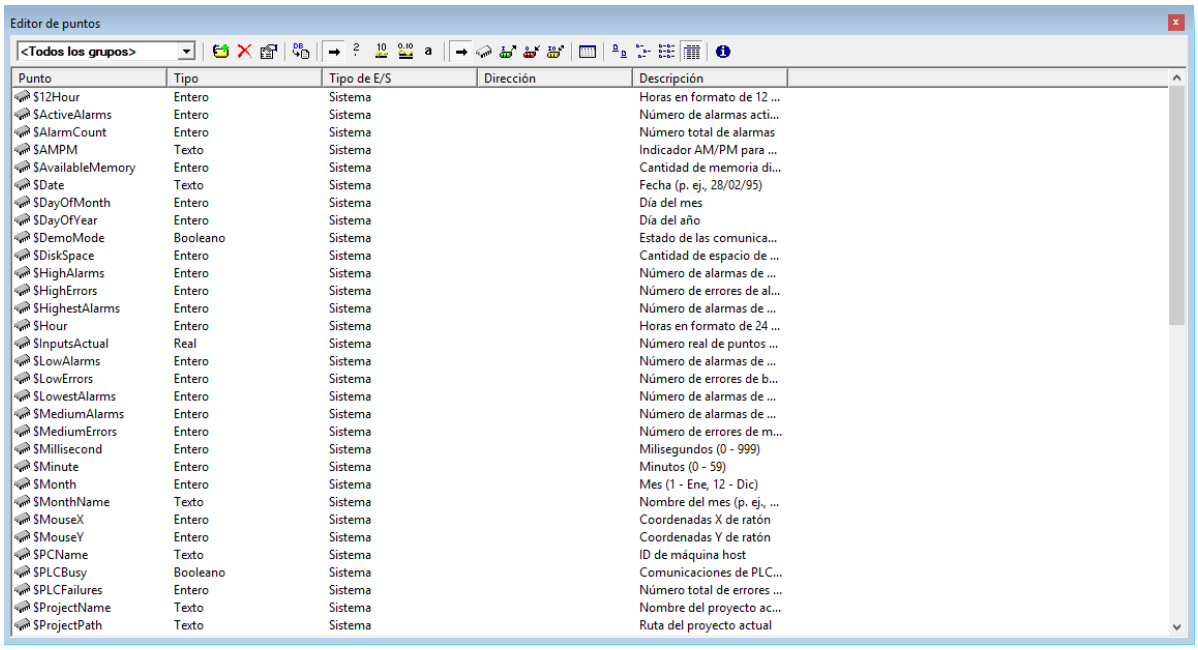


Ilustración 43. Venta de puntos en CX-Supervisor

Cuando nos encontramos dentro de la ventana de puntos, en la parte superior de esta, seleccionamos el icono que se encuentra más hacia el izquierda, el cual nos permitirá añadir la variable que necesitemos. Cuando esta acción se realice, le deberemos dar nombre a esta y asignar el tipo de variables que se trata, tal puede ser booleana, entera, etc. Para ello, seleccionaremos con doble click del botón izquierdo del ratón y automáticamente, se nos mostrará una pestaña en la que podremos realizar la configuración de estas. En esta ventana, además, podremos seleccionar la frecuencia de actualización de la variable, que va desde los cincuenta milisegundos, hasta el segundo. Debemos tener en cuenta que si al crear un gran número de puntos y con una frecuencia muy baja, podríamos tener ciertos problemas con la actualización de estas en el autómata.

A continuación, mostramos una imagen de esta ventana que hemos descrito en el párrafo anterior:

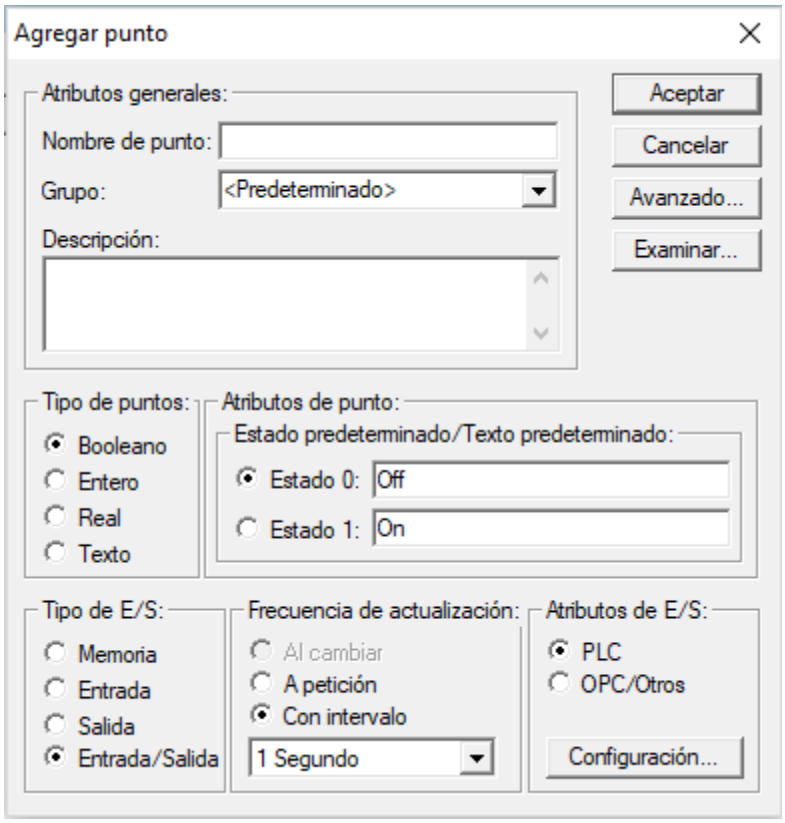


Ilustración 44. Ventana de configuración de punto

Como podemos apreciar en la ilustración 44, en la parte inferior derecha de la pestaña de "Agregar punto", se encuentra el atributo de E/S del que trata la variable. Como podemos observar, nos permite la opción de trabajar directamente con el PLC, o en su caso, trabaja con un servidor OPC o de otro tipo. Esta última idea, la de trabajar con un servidor OPC, es algo que desarrollaremos más adelante, pues para el caso de trabajar con una herramienta directamente del fabricante, es más lógico trabajar de forma local, pues no supone ningún beneficio trabajar con un servidor con este programa si el PLC con el que se está trabajando es de la propia empresa Omron. Por lo tanto, lo que debemos realizar seguidamente es seleccionar el botón de "Configuración" y así entrar en la ventana en la que seleccionaremos el PLC con el que vamos a trabajar y la dirección de la variable a la que se corresponde en el autómata.

Para añadir un PLC, debemos irnos a la ventana principal y seleccionar el icono de "Configurar dispositivo". Ahí, seleccionamos el tipo de dispositivo con el cual vamos a trabajar y la configuración de red que tiene este para la comunicación de datos y estado de variables. Este procedimiento debe ser realizado correctamente, pues en caso de no realizarse, cualquier trabajo de diseño de la pantalla no serviría de nada sin poder atacar al autómata en cuestión.

Con esto, tenemos creado los puntos o variables que necesitamos para nuestro scada y los cuales podremos asociar a distintos botones que añadiremos en nuestra pantalla.

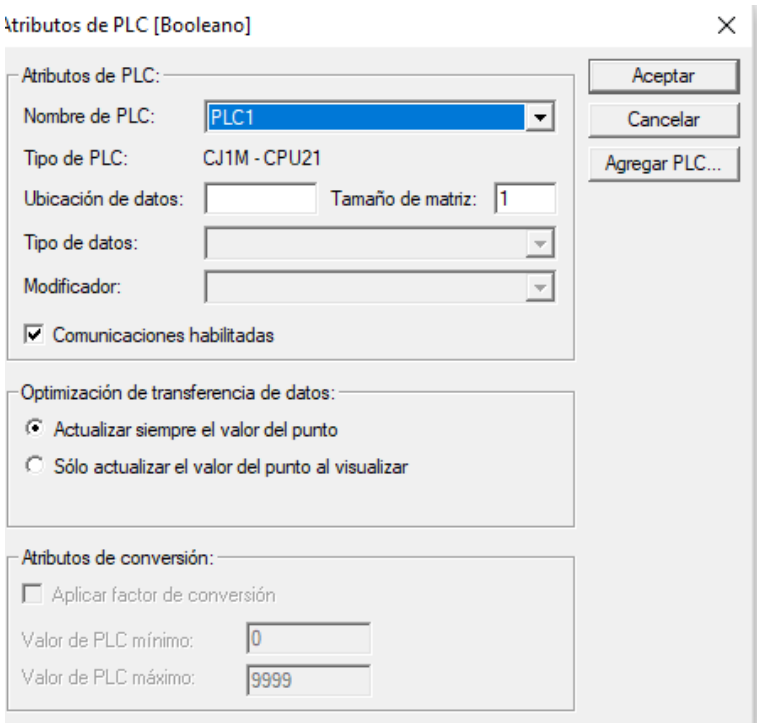


Ilustración 45. Ventana de Atributos de PLC

El siguiente paso que debemos realizar es la de añadir elementos a nuestro scada para así poder realizar las acciones que queramos y necesitemos para mostrar toda la información y poder permitir también el control que queremos que se lleve a cabo en la pantalla. Para ello, deberemos regresar al entorno principal del programa e ir las filas tres y cuarto de la ventana superior de herramientas. Tal y como podemos observar en la ilustración 42, en la tercera fila será aquella que utilizaremos para añadir las palabras que veamos oportunas como información añadida para los operarios que trabajen con esta. Con respecto a la fila cuatro, en esta se encuentran las opciones de añadir figuras, botones, medidores numéricos, etc.

Sobre la última idea que hemos comentado, crearemos distintos botones para accionar distintas variables, textos para que nos muestren el valor de algunas variables, e incluso que nos permitan modificarlas en la pantalla, como pueden ser el número de piezas o etapas.

Como hemos dicho previamente, para crear botones, seleccionamos en la cuarta fila y deberemos escoger la opción de “Botones de activación/desactivación”. Tras esto, nos posamos en la base de la pantalla y le damos el tamaño que veamos oportuno. Para poder editar los botones, lo seleccionamos y nos abrirá una ventana en la que podremos darle la configuración que queramos, el diseño que nos parezca más correcto y la variable a la que queremos que se encuentre asociado.

A modo de ejemplo, mostramos una imagen en la que se puede observar cómo es la ventana de personalización de los botones en CX-Supervisor.

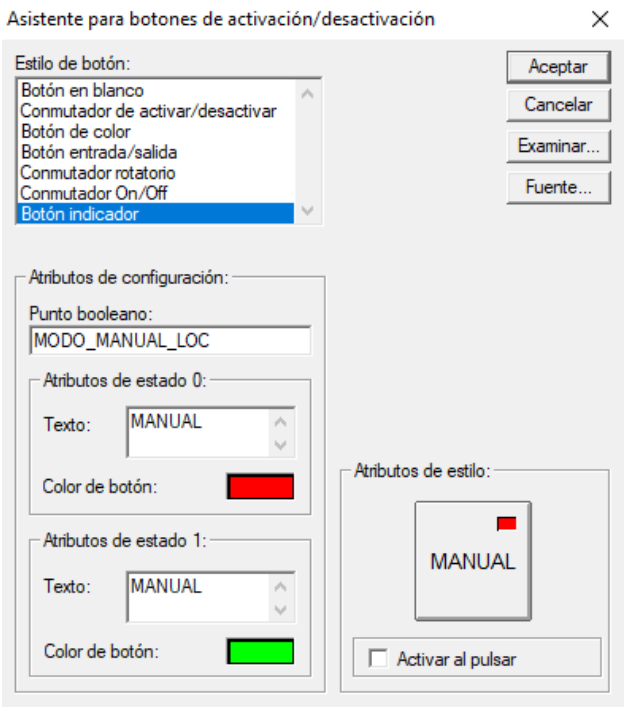


Ilustración 46. Ventana de "Asistente para botones de activación/desactivación" de CX-Supervisor

Pero además de realizar botones así, también los podemos realizar con figuras, como hemos hecho con los botones en el modo manual, los cuales son dos triángulos en el que indicamos el sentido que queremos que gire el servomotor. Incluso, las figuras nos permiten que sean visibles en caso de que alguna variables del sistema este en el estado que nosotros queramos, como vamos a mostrar a continuación:

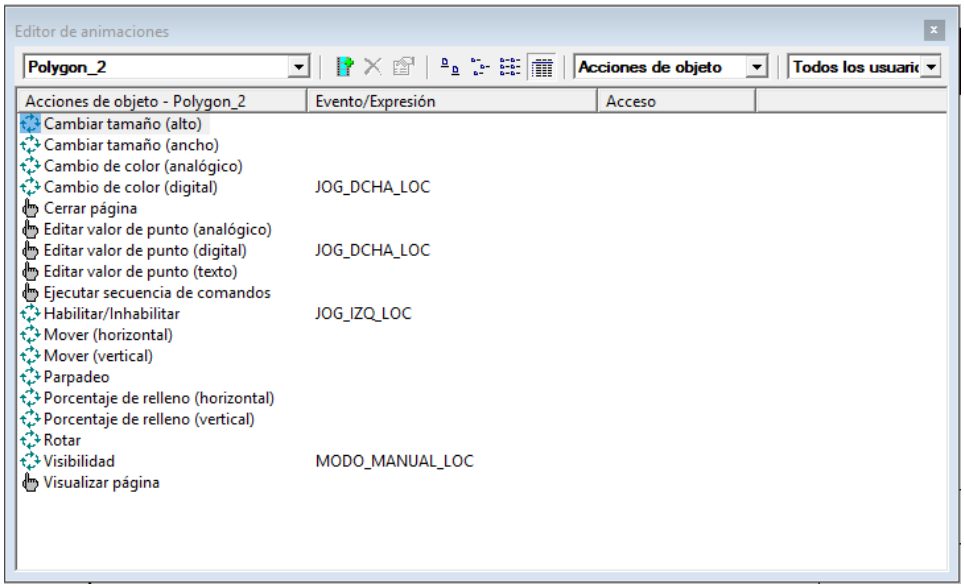


Ilustración 47. Ventana de editor de animación de una figura en funcionamiento como botón

Como podemos apreciar en la ilustración 47, se trata de la configuración que indica el sentido derecho en modo JOG. Lo primero es hacer que cambie de color para saber que está girando hacia el lado derecho y saber que la hemos seleccionado. Lo que debemos hacer seguidamente, es el cambiar el estado de la variable para que pasa de estado "0" a estado activo o "1". Como observamos, esto se realiza en la fila que nos indica "Editar valor de punto (digital)". Después, por motivos de seguridad y así poder evitar cualquier posible daño en el motor, inhabilitamos este botón en caso de que este seleccionado el botón del otro sentido de la marcha. Por último, y para poder hacer la pantalla más clara, haremos visible todo lo relacionado con el modo cuando se encuentre activado el modo manual de funcionamiento.

El siguiente elemento que añadiremos a nuestro scada serán las señales luminosas para indicarnos el estado de distintas variables que veamos que deben ser oportunas el conocer en cada momento como se encuentran, como pueden ser los indicadores del sentido de giro del motor o la indicación de que nos encontramos en el origen. Para ello, haremos uso de las figuras, y más concretamente de los círculos. Al seleccionarlo, le daremos el tamaño que veamos oportuno, como hacemos con el resto de los objetos que añadimos. En este caso, es más sencillo que cuando hacemos una figura como botón, pues simplemente lo que haremos es que cambie el color de la figura cuando cambie a estado activo la variables al que queremos que se encuentre asociado. Como detalle, no es necesario ir haciendo figuras individualmente y siempre teniendo que ir a la barra de herramientas, pues el CX-Supervisor nos permite copiar las figuras, y así también conseguimos que todas las señales sean iguales.

A modo de ejemplo, mostramos en la siguientes ilustración la configuración de la señales luminosas:

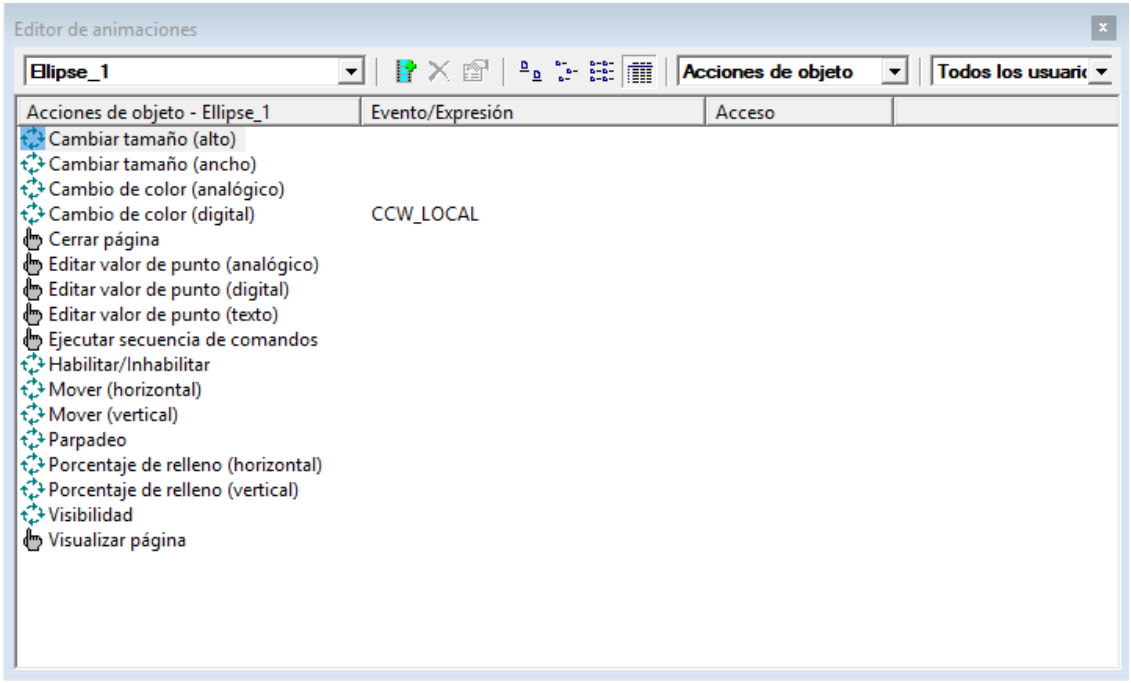


Ilustración 48. Editor de animaciones de una señal luminosa de sentido de giro del servomotor

La última idea que nos falta por comentar es la de incorporar textos en la pantalla. Debemos diferenciar el texto que es simplemente informativo y el cual nos permitirá dar el valor a ciertas variables.

Con respecto a la primera de las opciones de texto que vamos a incorporar, tenemos que ir a la tercera fila de la barra de herramientas y seleccionar el tipo de letra que queremos, y como en el caso de los programas de edición de texto, nos aparecerá en la base de la pantalla el indicador vertical para escribir. Tras escribir, colocaremos el texto en aquella zona donde queramos que aparezca. Podemos hacer que el texto sea visible o no según el estado de una variable, de la misma forma que hemos explicado en el caso previo de las figuras del indicador del sentido del JOG. Esto lo haremos para los textos de "JOG", que se activará en modo manual, y en el caso de "ETAPAS" cuando el modo automático se encuentre activo.

El siguiente tipo de texto que nos queda por explicar es aquel que nos permite cambiar el valor de alguna variable. Para ello, podemos escribir un texto normal, para indicar la variable de la que se trata, pero deberemos añadir "#", tantos como cifras tenga el valor. Esto lo haremos para el caso de indicar el número de etapas y de piezas. Además, el CX-Supervisor nos permite poner valores que son modificados por una operación matemática. Esto lo utilizaremos para el caso del tiempo, pues en la pantalla no nos permite colocar un valor con decimales y para no confundir al usuario, lo que hacemos es que el operario coloque una cifra falseada, pues hacemos que parezca que está colocando un valor con decimales, pero en realidad estamos mandando un valor entero que es dividido por diez para que el valor de la variable que le llega al autómata sea la que de verdad queremos.

Para una mejor explicación, adjuntamos una captura de como realizamos uno de los tiempos.

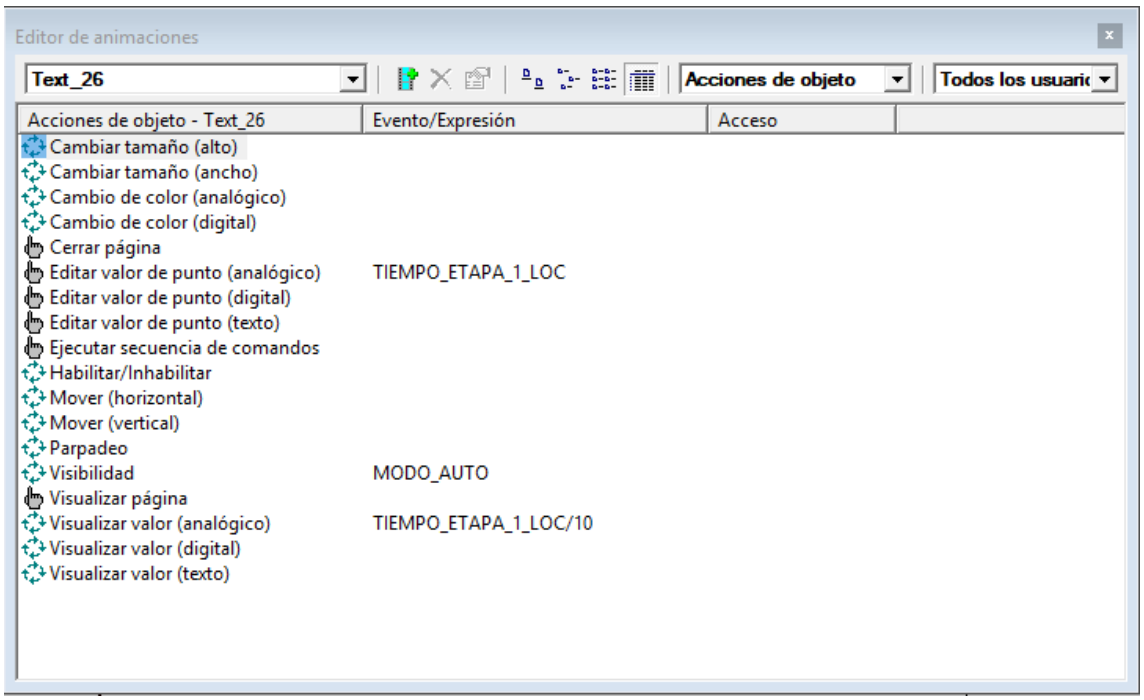


Ilustración 49. Editor de animaciones de una variable de tiempo

Además de estas ideas, podemos hacer que un texto se ilumine en caso de que una variable tenga un valor concreto. Esto lo realizaremos con los números que nos indican las etapas. Para ello, haremos que cambien de color en el caso de que el valor de etapa actual sea igual al valor del número que queremos que se ilumine. El CX-Supervisor nos permite poner funciones de comparación para obtener esto.

Como último detalle a nuestra pantalla, incorporamos un botón de salida de la pantalla. Esto lo realizaremos seleccionando un botón normal, pero que cuando lo configuramos, en la ventana de "Editor de animaciones", iremos a la fila que nos indica "Ejecutar secuencia de comando". Entonces, se nos abrirá una pestaña nueva en la que deberemos donde nos indique "Especial" y en la opción de "General", escoger la opción de "Shutdown". De esta forma, ya hemos añadido la última idea que necesitábamos

Con esto, finalizamos el apartado de la creación del scada con la herramienta de CX-Supervisor y el funcionamiento en modo local del scada. El diseño y explicación del funcionamiento del scada se encuentra más adelante en la parte del anexo.

7.2 Scada en funcionamiento con OPC

Tras haber observado el subapartado anterior y ver que se trabaja en forma local, nos propusimos en mitad del proyecto en realizar una mejora al proyecto y que nos permitiera además la posibilidad de crear un scada con una herramienta de diseño que no fuera propiedad de una de las empresa propietarias de software de programación y de creación de autómatas. El objetivo que nos propusimos con esto fue utilizar un programa gratuito que nos permitiera diseñar un scada correcto y que, además, fuera posible utilizar con cualquier otro autómata para que, en caso de que en el futuro se cambiará el autómata que realiza el control de la línea, pudiéramos seguir utilizando la misma pantalla y así abaratar posibles costes en el futuro. Para ello, el tutor propuso la idea de la utilización de un servidor OPC.

El OPC (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

El objetivo del OPC es crear una arquitectura genérica cliente/servidor, con la robustez y rapidez requerida en entornos industriales, la cual es ofrecida a cualquier desarrollador para acabar con los sistemas propietarios.

El método definido por OPC, facilita el intercambio de datos en forma estandarizada y simple en aplicaciones de control y automatización, entre los dispositivos y sistemas de campo y las aplicaciones de supervisión, administrativas y de oficina. Es decir, OPC simplifica la interfaz entre componentes de automatización de distintos fabricantes, con programas y aplicaciones tales como los sistemas administrativos y de visualización.

Con estas especificaciones, el diseño de un paquete scada, cuya comunicación se realizará con servidor OPC, no necesita disponer de drivers para los numerosos equipos industriales posibles.

Una de las páginas más importantes para conocer el funcionamiento de los servidores OPC y poder utilizar productos que nos permitan trabajar con esta herramienta, es la de la empresa "Matrikon".

Ilustración 50. Explicación funcionamiento OPC por Matrikon

Para nuestro caso, para crear el servidor que vamos a utilizar, haremos uso de la herramienta CX-Server OPC, en la cual deberemos añadir las variables que hemos creado previamente en la programación del autómata con la herramienta CX-Programmer. Debemos tener en cuenta que las variables deben estar con la opción de funcionamiento con OPC. Además, en el CX-Programmer, le añadiremos el servidor, indicando la dirección en la que se encuentra este, que en nuestro caso se encontrará dentro del mismo ordenador.

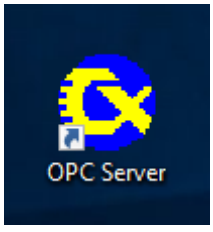


Ilustración 51. Icono en Windows del CX-Server OPC

A continuación, desarrollaremos el proceso de creación del servidor que necesitamos y que más adelante utilizaremos definitivamente.

Cuando seleccionamos el icono de OPC Server, se nos abre automáticamente una ventana en la que se nos indica que se puede controlar el servidor desde el icono que nos aparecerá automáticamente abajo a la derecha en la barra de tareas de Windows.

Tras observar que aparece el icono azul en la barra de tareas, lo seleccionamos con botón derecho del ratón y seleccionamos la opción de "Seleccionar proyecto de CX-Server…".

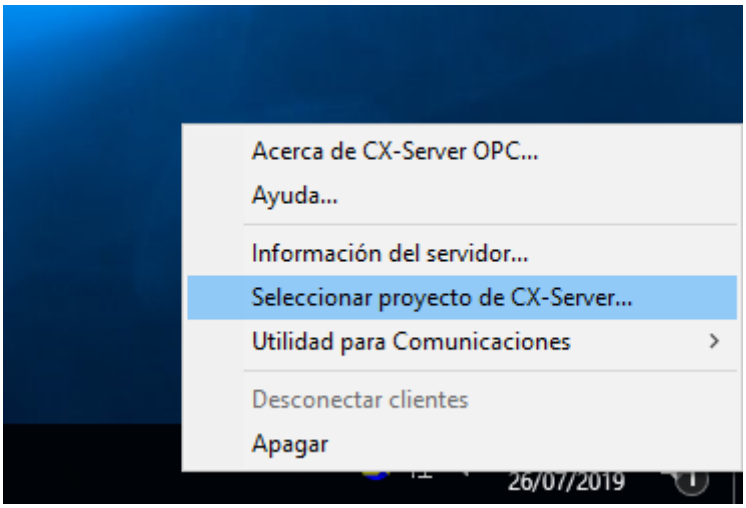


Ilustración 52. Pestaña de opciones del CX-Server

En ella, emergerá una venta en la que podremos crear el servidor en la dirección que queremos que este se encuentre o seleccionar el servidor en caso de que ya exista. Esta acción es muy simple, puesto que es como cualquier otro programa en el sistema operativo Windows, cuando queremos guardar algún documento y nos indica que seleccionemos la ruta en donde se ubica.

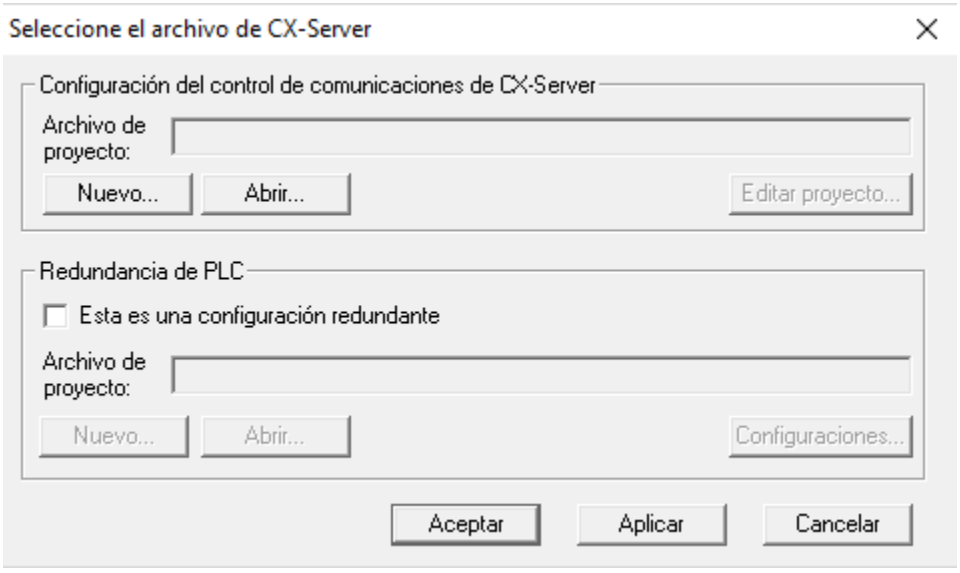


Ilustración 53. Venta de creación del servidor en CX-Server

Tras realizar esta acción, deberemos seleccionar la opción de “Editar proyecto…” y así poder acceder a la ventana en donde podremos añadir el autómata que queremos que comparta el estado de las variables para la comunicación entre scada y PLC.

Por tanto, añadimos el PLC al servidor con la ayuda de las acciones que se encuentran en la barra de herramientas de la parte superior de esta ventana y añadimos los puntos que tenemos en el programa principal. Deberemos cerciorarnos de que las direcciones que añadimos son las mismas para que se pueda trabajar correctamente con el servidor. Tras añadir cada una de las variables con la dirección que le corresponde, ya hemos finalizado con la creación del servidor.

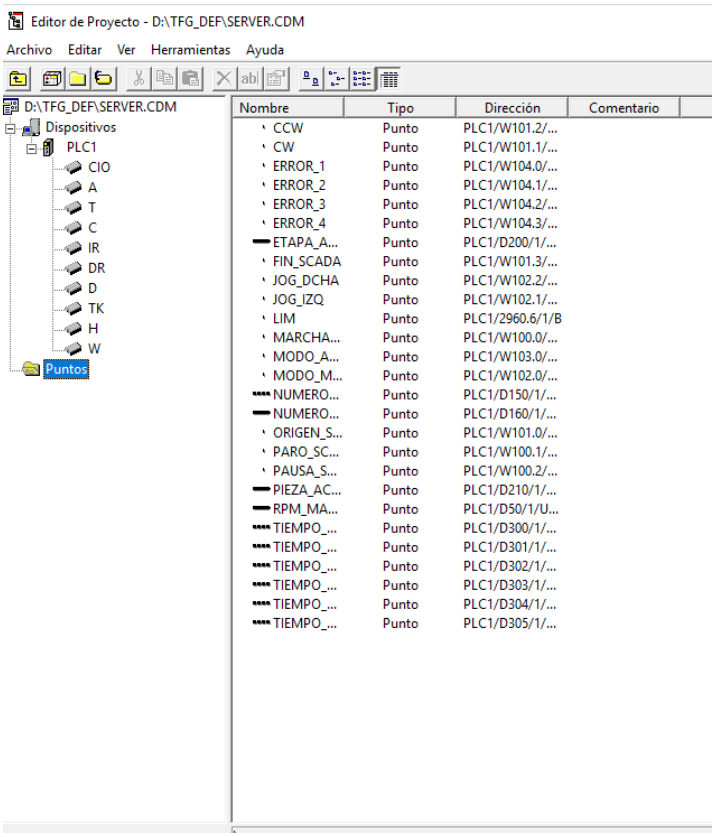


Ilustración 54. Ventana con los puntos del PLC creados en el servidor OPC

Como recomendación para el correcto funcionamiento del servidor, la dirección donde se encuentre este, no debe ser muy larga.

Hemos comentado previamente que con la herramienta del CX-Supervisor, se podía trabajar con servidores OPC. Sin embargo, en este caso no obtendríamos ninguna ventaja pues tiene más sentido que su funcionamiento sea de comunicación directa con el PLC al ser ambos de Omron y permitir esta comunicación, y que como hemos comentado previamente, el objetivo que debíamos cumplir con la utilización de servidores OPC es la de crear un scada con una herramienta gratuita y que sea compatible. Después de hacer varias pruebas y observar la incompatibilidad que había con trabajar con los servidores OPC con Windows 10 con distintos programas de edición que nos permitía crear con versiones anteriores de Windows, se decidió utilizar la herramienta Visual Basic, la cual es gratuita y se encuentra dentro del paquete que

ofrece Windows en Visual Studio 2019, de descarga gratuita en la red. Con ello, ya teníamos la herramienta en la que podemos trabajar nuestro scada genérico.

Tal y como se indica en la página web de Microsoft, Visual Basic está diseñado para crear de manera productiva aplicaciones con seguridad de tipos orientadas a objetos. Visual Basic permite a los desarrolladores establecer como destino dispositivos móviles, web y Windows. Al igual que todos los lenguajes que tienen como destino Microsoft .NET Framework, los programas escritos en Visual Basic se benefician de la seguridad y la interoperabilidad entre lenguajes.

Además, como dato adicional, Visual Basic es la herramienta de programación que nos permite crear macros en Excel.

Como hemos comentado hace unos instantes, Visual Basic se encuentra dentro de Visual Studio, como unos de las herramientas de programación que incluye y que, a continuación, mostraremos como hemos sido capaces de crear nuestro scada con esta herramienta.

Como otros programas de Windows, seleccionamos Visual Studio, que se encuentra en el escritorio del ordenador con el que nos han permitido trabajar en la aula del departamento, y el cual, en mitad del proceso de realización del presente proyecto, le tuvieron que instalar este programa para que pudiéramos utilizar esta herramienta. Cuando nos encontremos dentro de la aplicación, deberemos seleccionar el tipo de proyecto que vamos a crear. Para nuestro caso, seleccionamos la opción de "Biblioteca de controles de Windows Forms (.NET Framework)", en la que podemos observar que se encuentra el Visual Basic

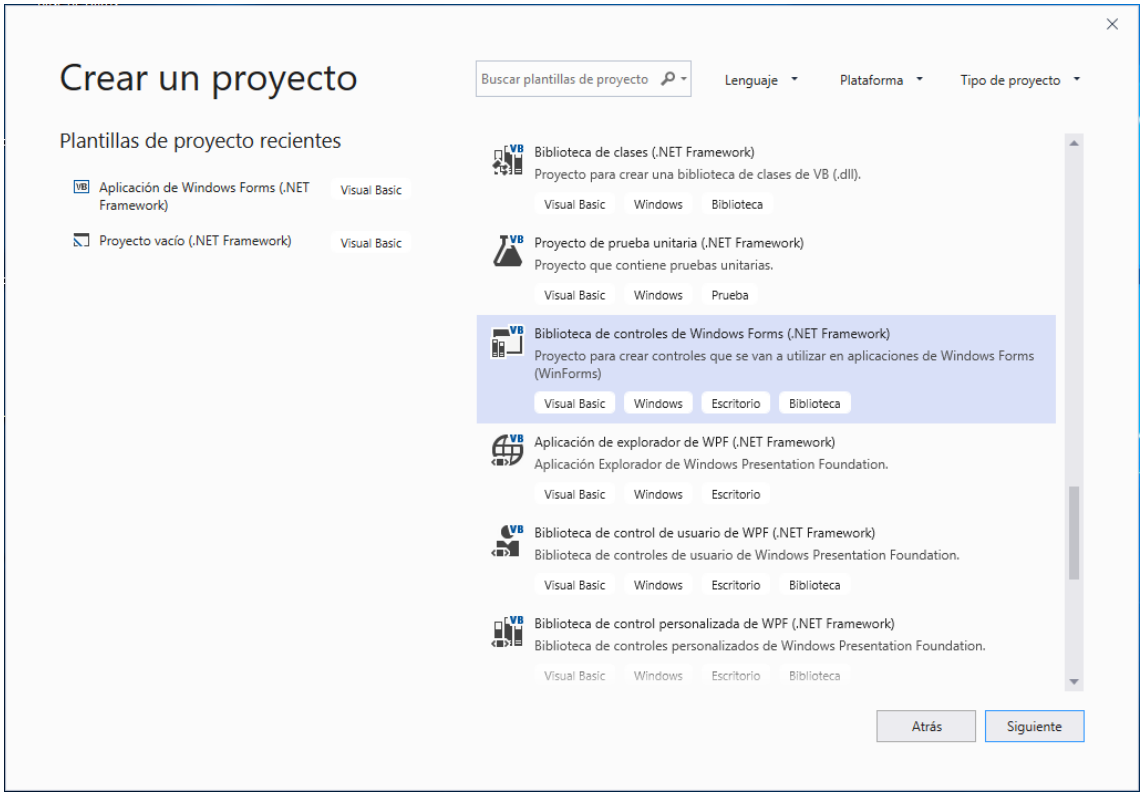


Ilustración 55. Ventana de crear un proyecto en Visual Studio

Después de haber seleccionado el tipo de proyecto, se nos abre el entorno del Visual Basic en donde podemos comenzar a diseñar nuestra otra pantalla.

Observaremos que como el caso del CX-Supervisor, nos aparecerá un fondo en el cual trabajaremos y que le podremos dar el tamaño que nosotros deseemos desde la esquina inferior derecha en el Visual Studio. También nos permite realizar cambios de color de fondo directamente desde la pantalla principal sin necesidad de tener que abrir una ventana nueva de edición.

Sin embargo, este programa es genérico y no tiene de forma predeterminada las herramientas con las que vamos a trabajar para diseñar nuestro scada. Por ello, deberemos añadirlas tal y como vamos a explicar a continuación.

Para incorporar las herramientas que necesitamos, deberemos ir a la parte superior de la ventana y seleccionar en el texto de "Proyecto", el cual se desplegara y en él, deberemos ir a la opción de "Agregar referencia…". Entonces emergerá una ventana nueva en la que deberemos seleccionar todos aquellos elementos con los que queramos trabajar para poder llevar a cabo el diseño de la pantalla. En nuestro caso seleccionamos todas aquellas herramientas que tenían el nombre de Omron. Esto se hace muy sencillo gracias a que todas las herramientas que podemos seleccionar se encuentran en orden alfabético.

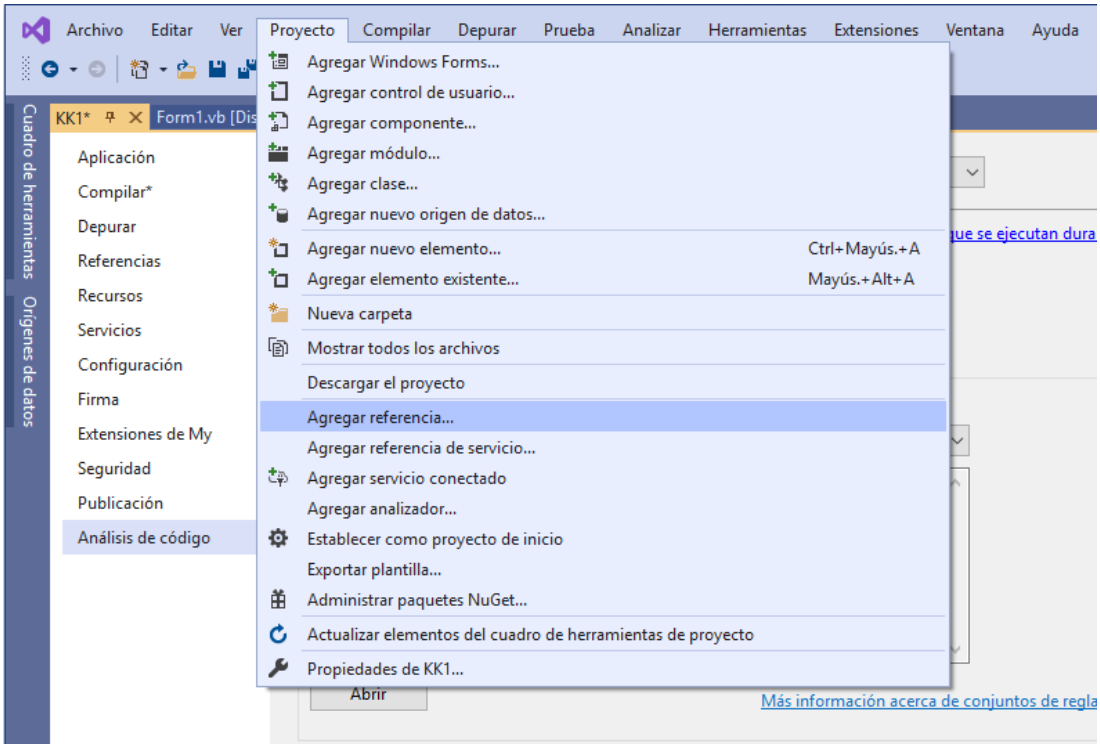


Ilustración 56. Despliegue en Visual Basic de la opción del Proyecto

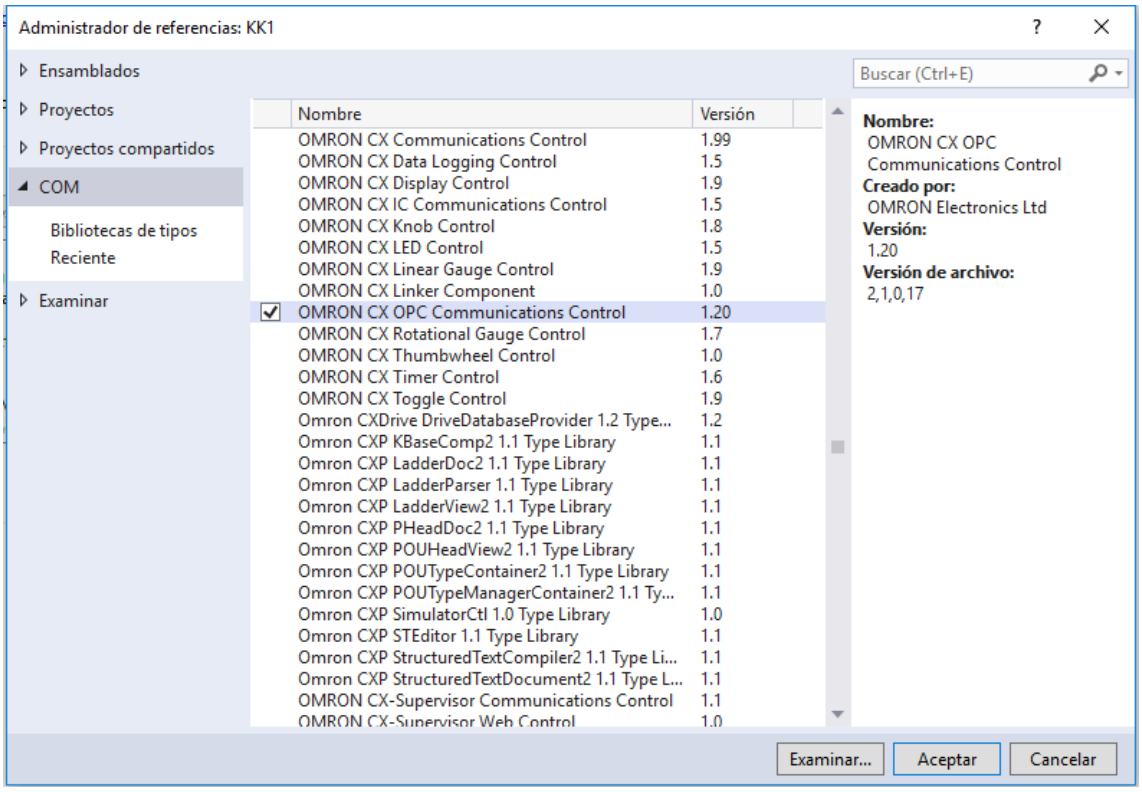


Ilustración 57. Ventana de Administrador de referencias en Visual Basic

Tras haber añadido todos los elementos, deberemos ir al cuadro de herramientas e ir a la opción de “Elegir elementos”. Esto se encuentra en la parte izquierda de la pantalla cuando vamos al entorno principal del programa. Ahí se encuentran todos los elementos, tal y como vamos a mostrar a continuación, en donde podremos observar en la ilustración 58 cuando aún no se han añadidos los elementos de Omron, mientras que la siguiente ilustración ya podremos observar que estos elementos se han añadido como hemos explicado con anterioridad.

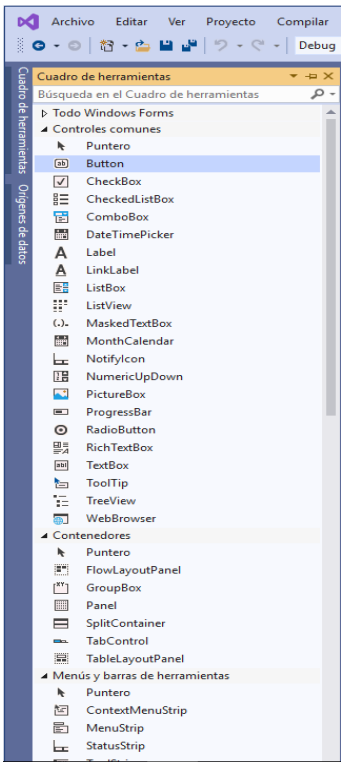


Ilustración 58. Cuadro de herramientas sin añadir los elementos de Omron

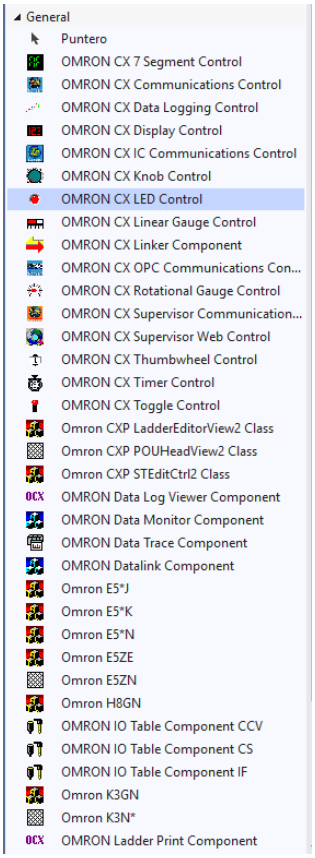


Ilustración 59. Cuadro de herramientas con elementos de Omron

Tras tener a nuestra disposición las herramientas de Omron, comenzamos con el diseño de la pantalla. Lo primero que deberemos realizar es incorporar el elemento que nos permita conectarnos al servidor OPC que hemos creado previamente. El elemento que debemos seleccionar es el de "Omron CX Communications Control". Tras seleccionarlo, lo colocaremos en la pantalla de diseño para poder asociarle la dirección del servidor. Tener en cuenta que cuando la pantalla se encuentra en estado "RUN", este símbolo desaparece, pues únicamente se puede visualizar cuando nos encontramos en modo edición. Para una mejor explicación, mostramos una captura en la que podremos observar que, al seleccionar el elemento, se nos desplegará un menú en donde deberemos seleccionar la opción de "Propiedades".

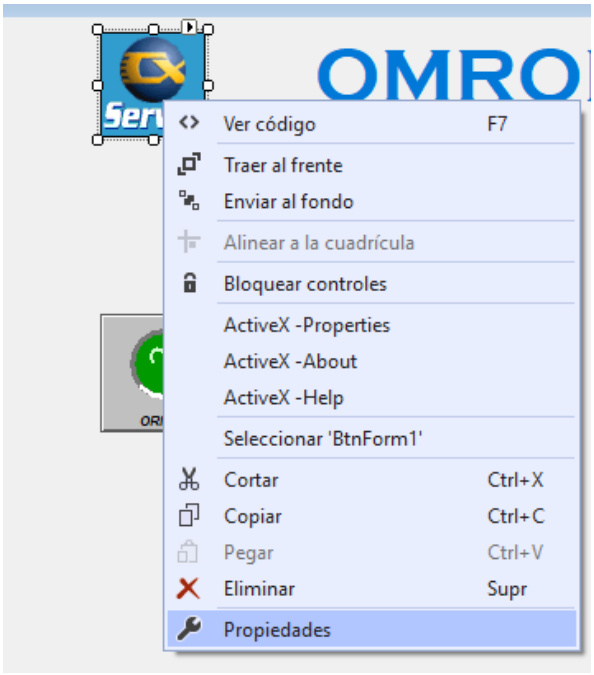


Ilustración 60. Menú desplegado del Omron CX OPC Communications Control

Como se ha comentado, en este elemento es donde seleccionaremos el servidor con el cual queremos trabajar para la comunicación de datos con el autómata programable. Esto nos permitirá, que el autómata no tendrá que estar conectado de forma local con el ordenador en donde se encuentre la pantalla con la que trabajen los operarios, sino que se encontrarán en espacios separados. Para seleccionar la dirección, nos aparecerá una pestaña en la que simplemente indicaremos el servidor al que queremos conectarnos. Como podremos observar en la figura siguiente, esta ventana es igual a la que nos aparece cuando trabajamos con el OPC Server para activar el servidor.

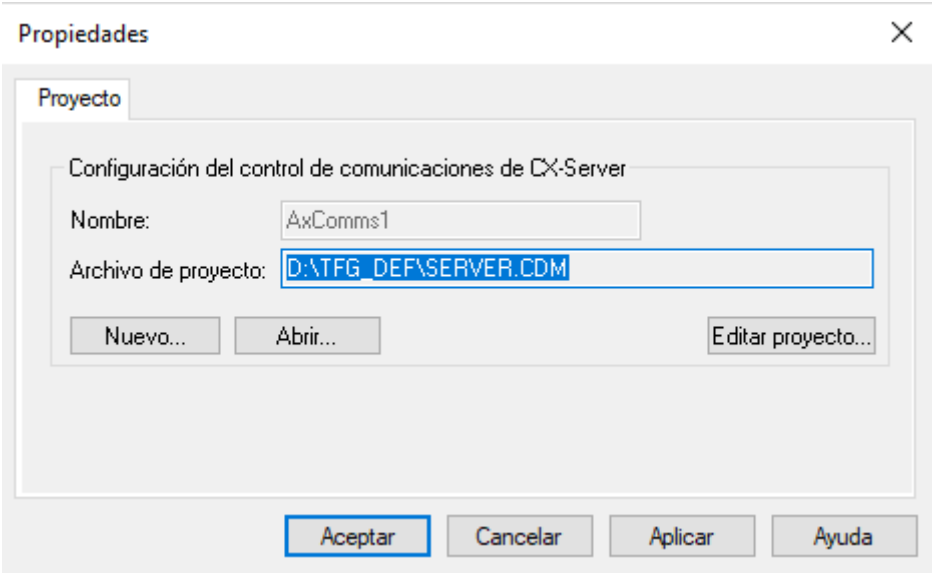


Ilustración 61. Ventana de propiedades del Omron CX OPC Communications Control

Tras haber añadido este primer elemento, ahora, como en el caso de la pantalla que hemos diseñado en CX-Supervisor, comenzaremos a añadir todos aquellos elementos que necesitemos para realizar un scada con el que se pueda controlar el servomotor.

Los primeros elementos que añadiremos serán las señales luminosas, que deberemos seleccionar en el cuadro de herramientas, la opción de "Omron CX LED Control". Al seleccionar esta opción, le damos el tamaño que nosotros deseemos. Cuando abrimos las propiedades de las señales luminosas, podemos observar que se nos permite añadir el nombre que queremos que tenga, sin necesidad de utilizar un texto externo, y pudiéndolo ubicar en donde sea más correcto. Esto se encuentra todo en opciones generales. Para cambiar el estilo y darle la imagen y color que queramos que tenga este elemento, deberemos ir a la opción de "Estilo".

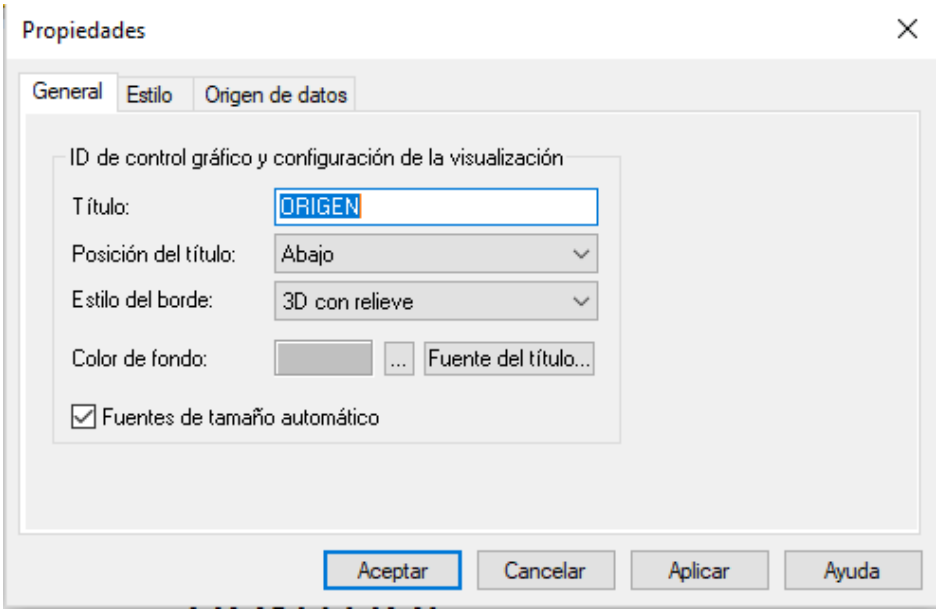


Ilustración 62. Ventana general de las propiedades de una señal luminosa en Visual Basic

Lo último que nos quedará será, con respecto crear una señal luminosa, añadir a la variable a la que queremos que este asociada para poder visualizar el estado de esta. Para ello, dentro de la propia ventana de propiedades, seleccionamos la opción de "Origen de datos". En este punto, nos mostrará el servidor que hemos configurado previamente, y de ahí podremos seleccionar el PLC con el que queremos compartir datos. Por último, seleccionamos el elemento que queremos visualizar, sin necesidad de poner la memoria donde se encuentra, pues al seleccionar el PLC del servidor, nos aparecerá el listado de los puntos que hay creados en él.

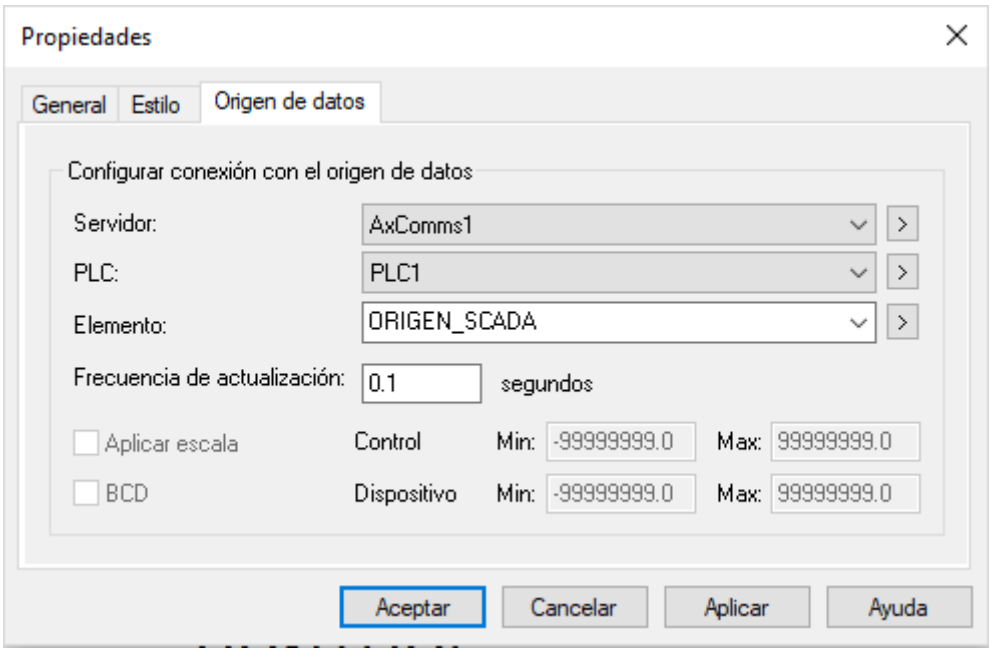


Ilustración 63. Ventana de origen de datos de las propiedades de una señal luminosa en Visual Basic

El siguiente elemento que añadiremos en nuestro futuro scada serán los botones, con los que podremos seleccionar el modo de funcionamiento con el que queremos trabajar y también los controles de marcha, pausa y paro. Al igual que los otros elementos, nos iremos al cuadro de herramientas y seleccionaremos para este caso la opción de "Omron CX Toggle Control". Lo añadimos a la pantalla principal y accedemos a su ventana de propiedades en donde, y al igual que en el caso de las señales luminosas, tendremos las tres mismas ventanas de general, estilo y origen de datos, donde realizaremos las mismas acciones que en el anterior.

En la ventana de general, añadiremos el texto que queremos que tenga cada botón para su fácil identificación en el scada. En estilo, seleccionaremos el diseño que veamos que es el más adecuado para nuestra aplicación. Y, por último, añadiremos el punto del servidor que queremos atacar con el correspondiente botón para cambiar su estado.

Para un mejor entendimiento, mostraremos en la tres siguientes imágenes, correspondientes a las números 63, 64 y 65, un ejemplo de cómo hemos realizado uno de estos botones, el cual se corresponde al que activa el modo manual de funcionamiento de nuestro sistema.

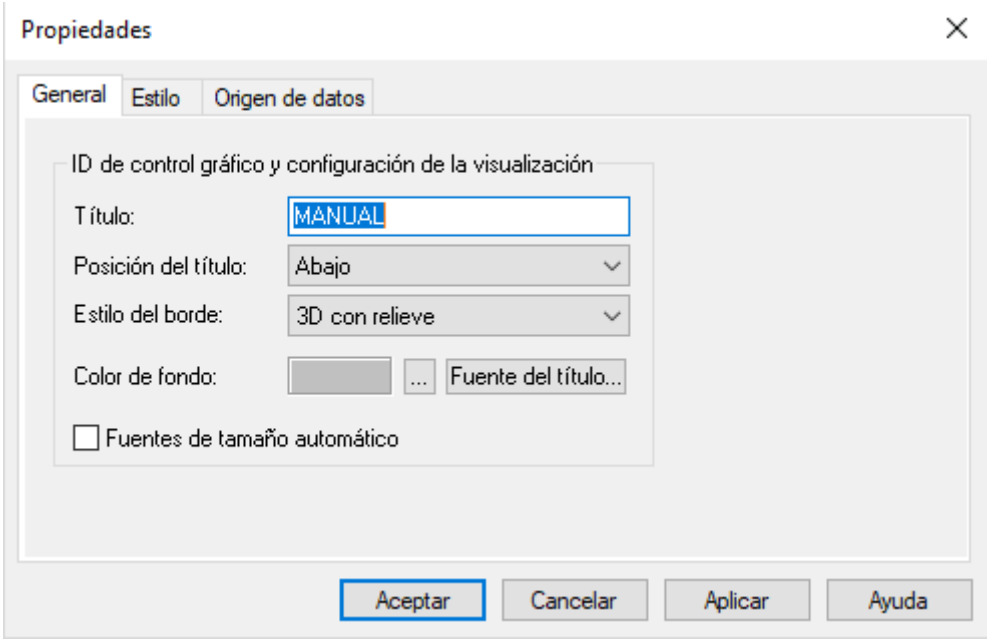


Ilustración 64. Ventana general de las propiedades de un botón en Visual Basic

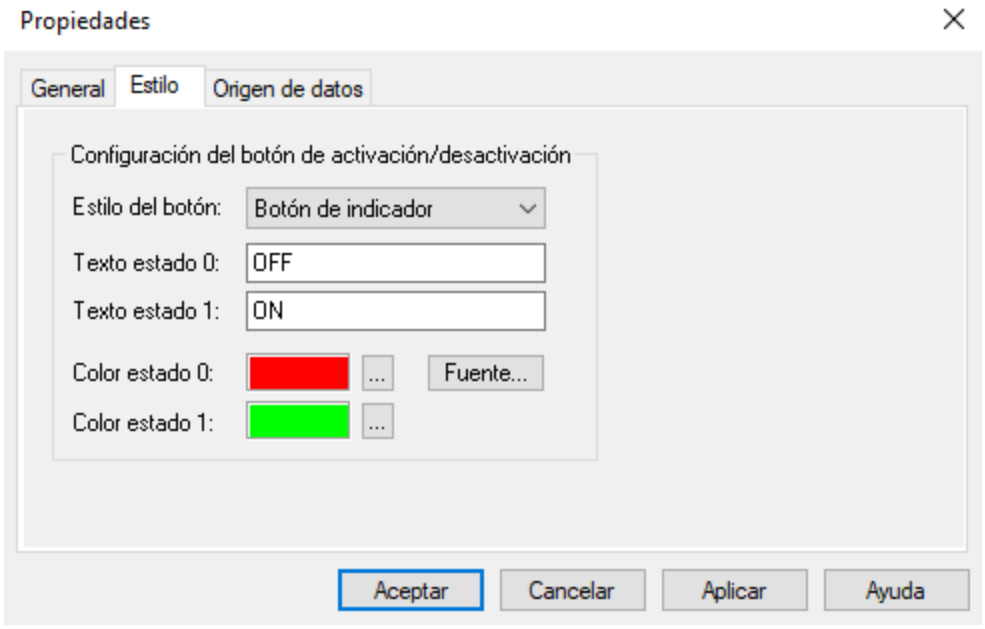


Ilustración 65. Ventana de estilo de las propiedades de un botón en Visual Basic

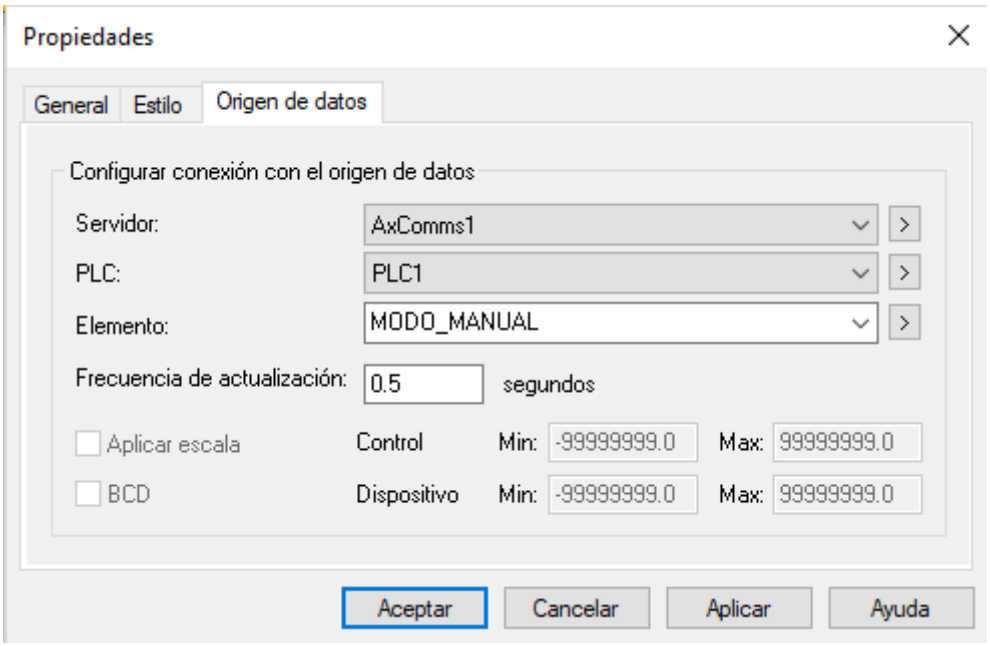


Ilustración 66. Ventana de origen de datos de las propiedades de un botón en Visual Basic

Para continuar con los elemento a incorporar en nuestra pantalla, hablaremos de los elementos para cambiar los valores numéricos de las variables enteras. En ello, trabajaremos tanto el número de etapas y piezas que queremos que se trabajen, como los tiempos por etapas del funcionamiento en modo automático.

Al igual que en las otras variables, todo esto lo seleccionaremos desde el cuadro de herramientas, incorporándolo a la ventana principal del diseño de la pantalla. Cuando hayamos seleccionado el tamaño que queramos que tenga, deberemos ir a la opción de propiedades y dotar a este elemento de las propiedades que queramos que tenga.

Para la explicación de este elemento, haremos uso del ejemplo realizado para modificar el número de etapas que queremos que tenga nuestro proceso de trabajo.

Lo primero, ir a la ventana de estilo que queremos que tenga nuestra instrucción. Hay que comentar que esta es exactamente igual a los elementos que hemos visto previamente con los otros elementos que conformarán nuestra pantalla.

Con respecto a la ventana de estilo, esta si varía con los elementos anteriores, pues para este caso, deberemos indicar el valor predeterminado de la variable, su número máximo y su número mínimo, el tipo de dato con el que estamos trabajando y el número de dígitos que queremos que nos muestre. También, nos permite seleccionar como queremos variar los número, con flechas o con botones de “+/- “.

Para el caso de seleccionar la variable con la que queremos trabajar, la ventana es exactamente la misma que la que nos encontramos en los demás elementos.

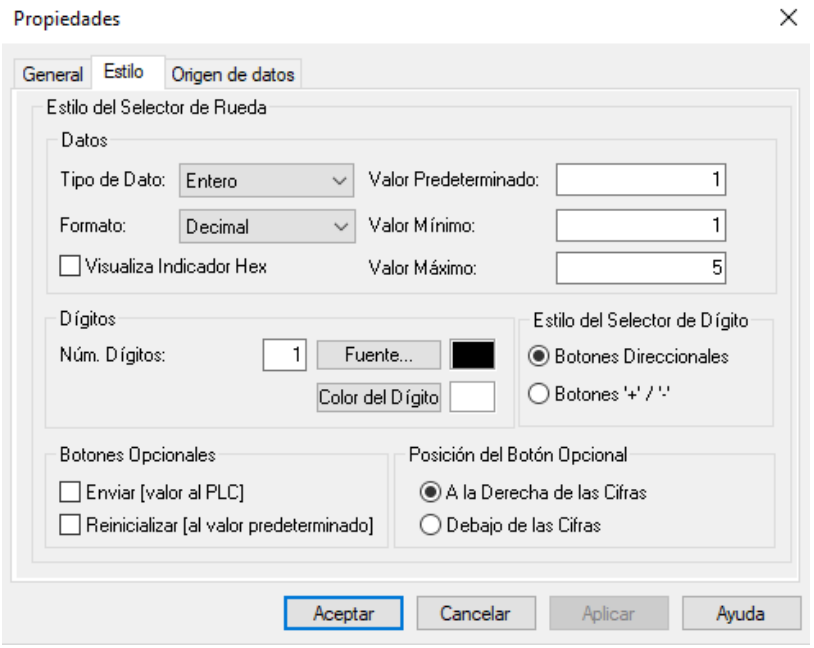


Ilustración 67. Ventana de estilo de las propiedades para cambiar el valor de una variable numérica en Visual Basic

Este tipo de elementos también lo utilizaremos para mandar la velocidad que queremos que gire el servomotor cuando trabajamos en modo JOG, aunque añadiremos la opción del botón "Enviar" por motivos de seguridad.

Otro caso especial en donde observamos que hay un cambio en la ventana de estilo, es la herramienta de visualización numérica que utilizamos para conocer en que etapa y pieza se encuentra en cada momento el sistema. Como observaremos a continuación, en ella seleccionaremos el tipo y formato en el que queremos visualizar este valor, además de los valores máximos y mínimos que pueden tener estos.

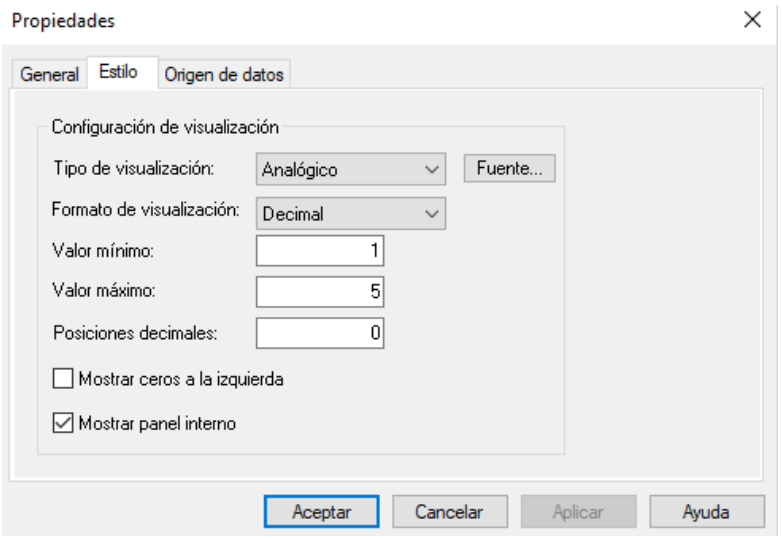


Ilustración 68. Ventana de estilo de las propiedades del elemento visualizador del número de etapa en Visual Basic

Por último, como hemos realizado previamente con el anterior scada que hemos diseñado, añadiremos un botón de salido cuando nos encontremos en modo “RUN” y el cual añadiendo un código que haga para el modo run y haga un “Shutdown” de la pantalla.

Con esto, finalizamos el apartado de realización del scada y nos disponemos a acoplar el servomotor al eje lineal para la demostración, habiendo realizado previamente las pruebas oportunas sin conectarnos al eje lineal.

8. Montaje

En este apartado explicaremos la realización del montaje entre el banco o eje lineal y el servomotor Omron, conectado al tablero.

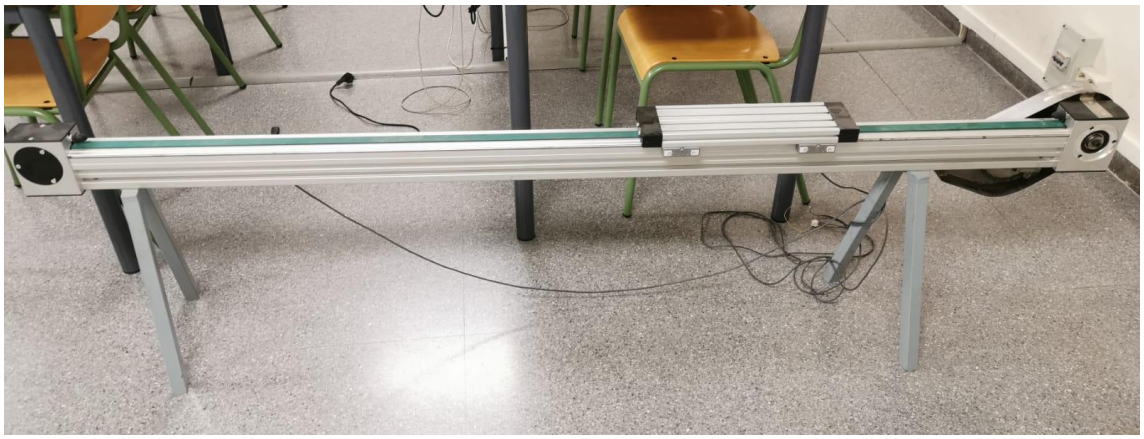


Ilustración 69. Bancada preparada para comenzar con el montaje

Debemos comentar que, a la hora de realizar la asociación, debido a que la transmisión se realiza mediante un tren de engranajes y cadena, el servomotor que debemos acoplar debe tener acoplado una rosca con piñones. Más concretamente, la rosca que utilizaremos será de 12 piñones.

Para la asociación del servomotor a la bancada, es una tarea que no lleva una excesiva complicación, pues el servomotor se encuentra atornillado a una base rectangular que dispone de dos orificios para colocar los tornillos. Con el uso de unos topes que colocamos en la bancada, atornillamos el motor con su base a estos topes. Debemos tener en consideración que debemos colocar los piñones en la cadena que queda suelta en el sistema de engranajes que utilizamos para transmitir el movimiento circular que realiza el motor, en movimiento lineal para mover la base que se encuentra en la parte superior. Tras atornillar y colocar en ubicación correcta el servomotor, para que los piñones se encuentren dentro de la cadena, hemos finalizado esta parte del montaje. A continuación, debemos conectar los sensores a nuestro tablero para su correcto funcionamiento.

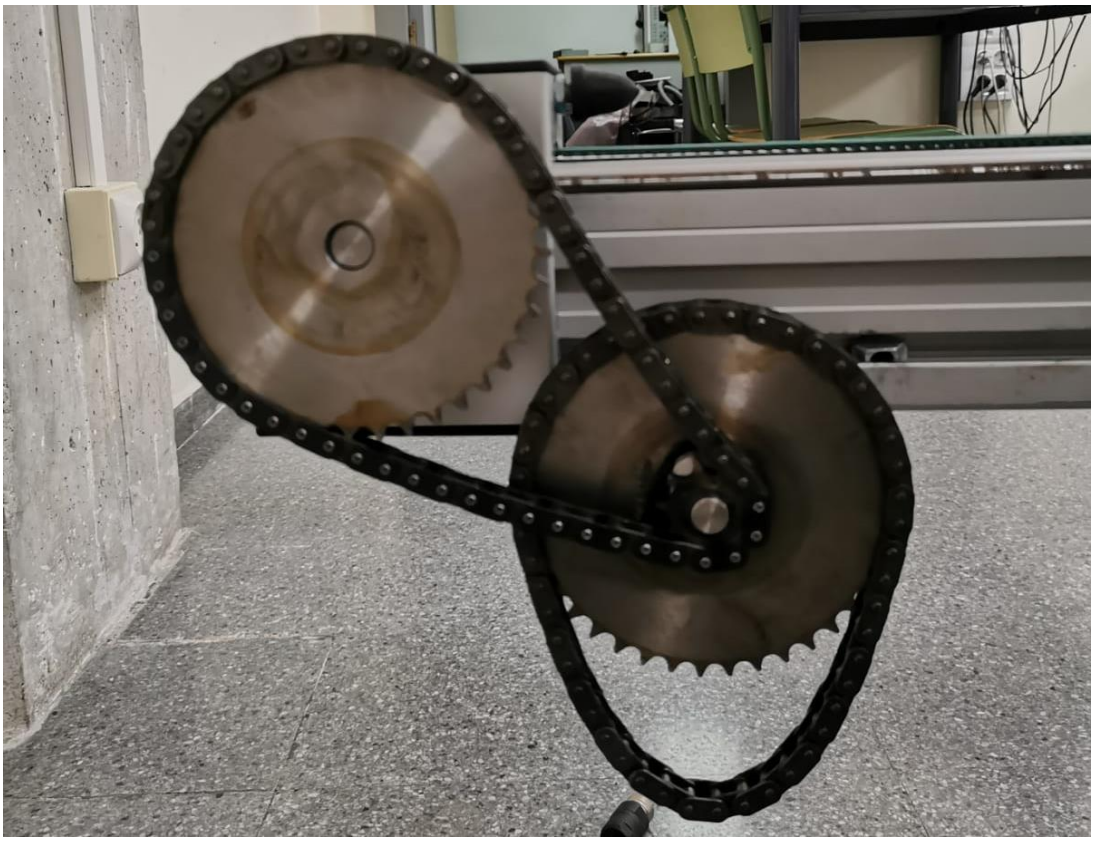


Ilustración 70. Sistema de engranajes sin tener acoplado el servomotor

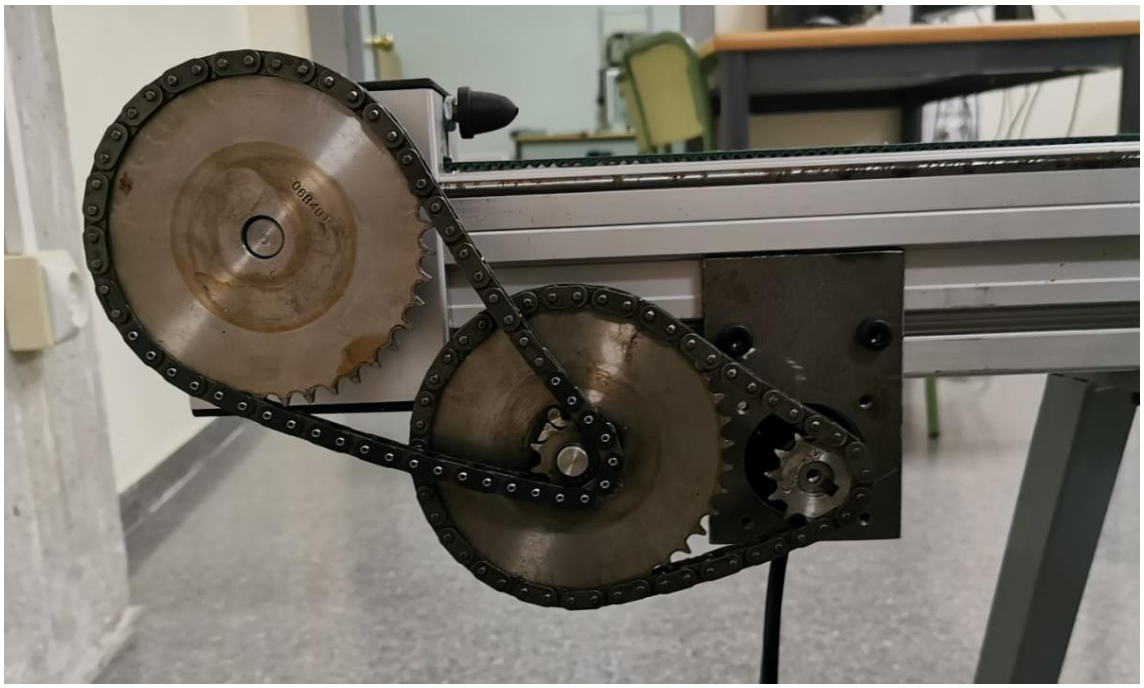


Ilustración 71. Sistema de engranajes junto con el servomotor

Los sensores a utilizar para nuestra demostración son sensores ópticos, que debemos conectarlos a la interfaz pasiva o placa de conexiones XW2B-40J6-9A. Los sensores se encuentran colocados en la bancada en una placa que está atornillada. Nos encontramos con tres sensores: Dos sensores de final de carrera y uno que será el que nos indique la señal de origen.

Ilustración 72. Sensor inactivo

Para su colocación, los finales de carrera irán en las bornas 22 (IN6), el cual será el de final de carrera de izquierdas y 23 (IN7), que se trata del final de carrera de derechas. Mientras, el sensor de indicador de origen se encontrará en la borna 24 (X). Tener en cuenta que en estas bornas ya se encuentran algunos botones de la botonera que hemos utilizado previamente cuando hemos estado en el periodo de diseño y hemos simulado el funcionamiento de nuestra aplicación. Los cables de las botoneras nos las desconectaremos, puesto que podemos utilizarlos en caso de cualquier emergencia o revisión del sistema. El color del cable que debe ir a estas bornas por parte de los sensores son los de color negro, que serán los que transmitan la señal. Por otro lado, se encuentran los cables de color marrón y azul, que deberán ir a las bornas de alimentación, las cuales son las que tiene los números 20 (+24V) y la 0 (0V).

Para las conexiones es simplemente abrir la tapa de la placa de conexiones, desentornillas las bornas en donde debemos conectarlas y posteriormente volverlas a atornillar. Tener en cuenta que esta manipulación se debe hacer en ausencia de alimentación.

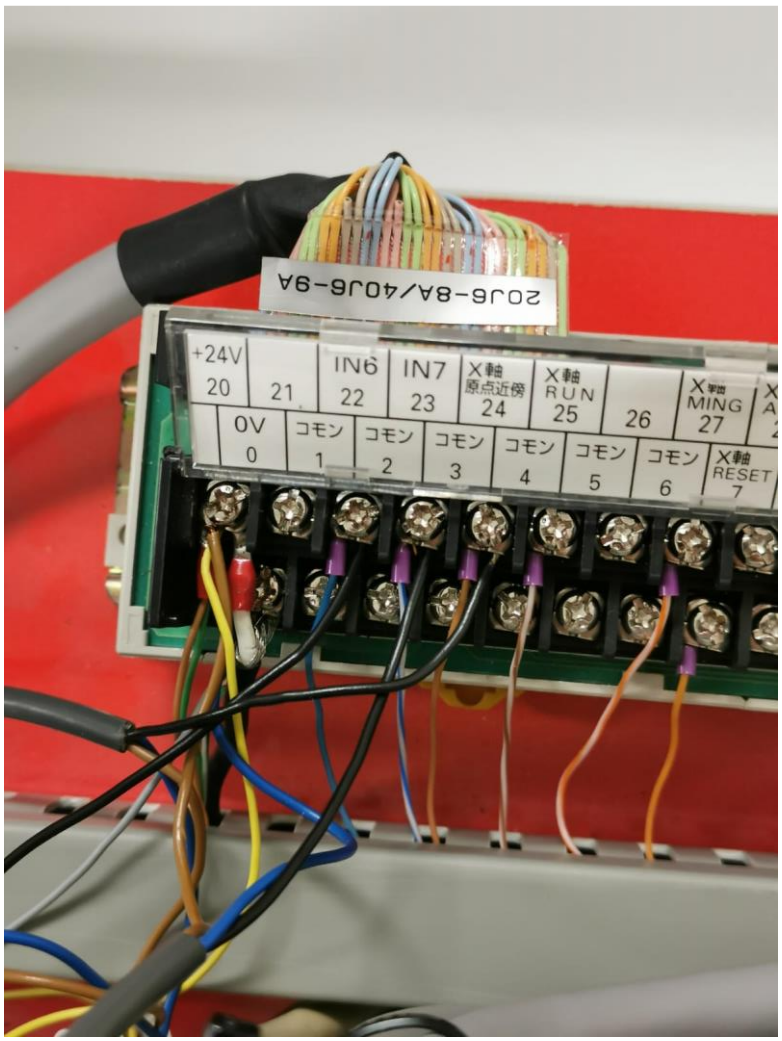


Ilustración 73. Conexiones de los sensores en la interfaz pasiva

Tras la conexión, estamos en la situación de realizar nuestra simulación y comprobar que todo funciona de forma correcta.

Observamos que la plataforma que se encuentra en la parte superior tiene atornillada una plaquita de metal que será la que haga activarse los sensores ópticos de los que disponemos. Los sensores tienen en su parte inferior una señal luminosa, lo que nos permite darnos cuenta de que si se encuentra activo o no.

Ilustración 74. Sensor en estado activo

Finalmente, y tras ver el funcionamiento en el eje lineal, podemos sentirnos satisfechos del funcionamiento general del sistema, pues realiza la acción que estábamos deseando.

9. Conclusiones

Tras la realización del todo el proyecto, podemos comentar que el resultado obtenido es satisfactorio, puesto que hemos sido capaces de poner en funcionamiento un equipo formado por varios elementos que se encontraban en desuso hace varios años y que simplemente han necesitado ser modificadas sus configuraciones para poder realizar la tarea que queríamos que se llevará a cabo, el diseño de una línea de montaje, obviamente con ciertas limitaciones como el número de etapas máximas y piezas programadas.

Hicimos funcionar el sistema con la botonera física, comprobando el correcto funcionamiento de todos los dispositivos, aunque sabíamos que el hecho de trabajar con esta botonera, en la vida real, no es el sistema más eficiente y queríamos que nuestro sistema estuviera automatizado.

Hemos sido capaces de trabajar con el software de programación de la empresa Omron, que no habíamos tocado previamente, al igual que el programa para diseñar la pantalla, CX-Supervisor.

Sin embargo, la mayor particularidad que tiene el proyecto es el haber trabajado con servidor OPC, con el objetivo de conocer esta tecnología y ser capaces de poder hacer un scada que no fuera de software propio de una empresa del sector de la automatización como es el CX-Supervisor como es el Visual Basic dentro del paquete que ofrece Microsoft en Visual Studio y que es de acceso libre, por lo que nos ahorramos los costes con respecto el uso de la licencia del programa, al igual del hecho de un posible cambio en el futuro del autómata que se utiliza para controlar todo el proceso, puesto que al usar un servidor OPC, si el nuevo autómata fuera de una empresa que no fuera Omron, debería funcionar con el mismo Scada sin la necesidad de realiza un nuevo scada, lo que en el ámbito industrial podría provocar una paralización en la línea de producción por el hecho de necesitar diseñar un nuevo scada que se pudiera utilizar.

Por último y tras varios ensayos previos sin ningún tipo de conexión al eje lineal, el cual utilizaríamos para demostrar el correcto funcionamiento de nuestro trabajo, se hizo el correcto acople del servomotor al eje lineal, con las conexiones pertinentes de los sensores en donde se materializado todo el trabajo que se llevo a cabo en los últimos meses.

Nos ha permitido poner en práctica varios conocimientos que hemos ido aprendiendo en el grado, sobre todo conocimientos derivados de las tres asignaturas de la mención de accionamientos eléctricos y operación remota.

10. Presupuesto

En este apartado desarrollaremos el coste de cada elemento que compone nuestro sistema y que ha hecho falta para la realización de este trabajo. Debemos tener en cuenta que todos los elementos han sido cedidos por el departamento del DIE para hacer posible la realización de este proyecto.

Material	**Ud**	**Cantidad**	**Coste (€)/ Ud**	**Costes (€)/Total**
PLC Omron SYSMAC CJ1M CPU21	Ud	1	359	359
Servopack Omron SGDH-02 AE-OY	Ud	1	899	899
Placa de conexiones Omron XW2B-40J6-9A	Ud	1	159,14	159,14
Sservomotor SGMAH-02AAA61D-OY	Ud	1	1500	1500
Eje lineal	Ud	1	750	750
Sensor Fotoeléctrico	Ud	3	30	90
Interruptor unipolar	Ud	12	0,59	7,08
Interruptor automático	Ud	2	10,7	21,4
Licencia Software	Ud	1	2194,94	2194,94
Materiales varios	-	-	50	50
Ingeniería	h	100	25	2500
		Total (Sin IVA):		8530,56
		Beneficio Industrial (%6)		511,83
		Total (Sin IVA) + BI		9042,39
		Total + 21% IVA		**10941,30**

11. Pliego de condiciones

En este apartado, mostraremos el pliego de condiciones que se comento al principio del proyecto, cuando surgió la idea de realizarlo en el mes de Febrero del año 2019, en el que los profesores del departamento de ingeniería eléctrica, Don Manuel Pineda Sánchez y Don Juan Pérez Cruz, y el alumno del grado en ingeniería eléctrica, Miguel Domínguez Belloch, se pusieron de acuerdo para realizar el presente proyecto y en donde comentaron una ruta de trabajo, especificando paso por paso los progresos que se debían llevar a cabo y lo que se debía realizar para obtener el resultado esperado.

Se deberá llevar la puesta en marcha y funcionamiento del tablero que prestará el departamento de ingeniería eléctrica, el cual está formado por un autómata programable de la empresa Omron, el SYSMAC CJ1M CPU21; un servopack SGDH-02 AE-OY, también de la empresa Omron y una placa de conexiones de Omron con designación XW2B-40J6-9A. En el tablero, además, se encuentra una botonera física que deberá ser revisada, para comprobar que todos los interruptores que la conforman se encuentran en un estado óptimo de funcionamiento y que las conexiones que se hay entre los distintos equipos están en condiciones de poder trabajar con ellas.

A continuación, se proporciona al alumno un servomotor Omron SGMAH-02AAA61D-OY, con el cual trabajará para llevar a cabo la correcta demostración de que todo se encuentra en perfectas condiciones y que los equipos se encuentran en condiciones de poder trabajar con ellos. Antes de acoplar el servomotor al eje lineal, se deberán realizar pruebas de funcionamiento sin estar el motor acoplado para evitar posibles daños en el material.

Se realizará un funcionamiento con la botonera, previamente revisada, tanto trabajando con un control de tipo velocidad, sin exceder los límites del servomotor para evitar daños, como un control de posición.

El siguiente paso a realizar será realizar un programa con la herramienta de programación de la empresa Omron CX-Programmer en su versión 7.2, la cual tiene instalada todas las computadoras del aula del departamento, y en donde se deberá llevar a cabo el funcionamiento de forma automatizada. El lenguaje de programación a utilizar será el de tipo “Ladder”. En el programa se deberá llevar a cabo la posibilidad de trabajar en dos modos:

- Modo manual: Consistirá en un control de velocidad de tipo JOG, en el cual el motor podrá girar en un sentido u otro y a velocidades que serán posibles de seleccionar de forma externa.
- Modo automático: Consistirá en la realización de un control de posición, haciendo que el motor se mueva con un mismo tren de pulsos, la cantidad de veces que se haya programado y simulando una cadena de montaje, en donde se desplaza una pieza por varias estaciones o etapas. Lo primero a realizar en este modo de funcionamiento será la búsqueda de un punto de referencia 0 o punto de origen, el cual, a partir de ahí, se podrá dar comienzo a la línea de montaje.
  Se podrán realizar un máximo de 5 etapas por ciclo, y con un máximo de 5 ciclos, en donde cada etapa podrá ser personalizada su duración de tiempo, el cual se escogerá de forma externa desde una interfaz scada.

Deberá haber un botón de paro, que, en caso de necesidad, corte directamente el movimiento del motor y cuando este se retire, ponga el sistema en funcionamiento desde el inicio, volviendo a la búsqueda del punto origen.

Se deberá tener en cuenta la posibilidad de errores, los cuales harán parar automáticamente las acciones del PLC y paran el sistema al instante. Estos errores serán la activación de los dos sensores de final de carrera, sobre pasar la velocidad de 3000 rpm por parte del motor, que los dos modos de funcionamiento se encuentren activos al mismo tiempo y en el caso de estar en el modo manual, se seleccionen los dos sentidos de giro simultáneamente, pues esto podría provocar daños en los equipos.

Habrá que realizar una interfaz scada para supervisar el funcionamiento del sistema, en donde podrá seleccionar el modo de trabajo que se quiere llevar a cabo. Se integrarán señales luminosas en la que nos indicarán el punto de origen, el sentido de motor a izquierdas y a derechas y otra señal luminosa de indicación de fin cuando se iluminará cuando se haya acabado de realizar un ciclo de trabajo. Incorporará también la opción de seleccionar el número de etapas que queremos que se realicen y las piezas que queremos que se hagan en una batida, al igual que la posibilidad de programar los tiempos que debe haber por cada etapa.

Como añadido, se quiere que el sistema trabaje con un servidor OPC y con un scada que sea realizado en un programa genérico, sin la necesidad de pagar licencia por ello, y que permita el correcto funcionamiento, tal y como dejaría un programa de diseño de interfaces de una empresa propia de la industria de automatización, para abaratar costes y pensando que en el futuro, en caso de recambio del autómata de la casa Omron, se cambiará por otro de otra empresa y que fuera compatible su uso con servidores OPC, funcionará con el scada que ya se ha realizado.

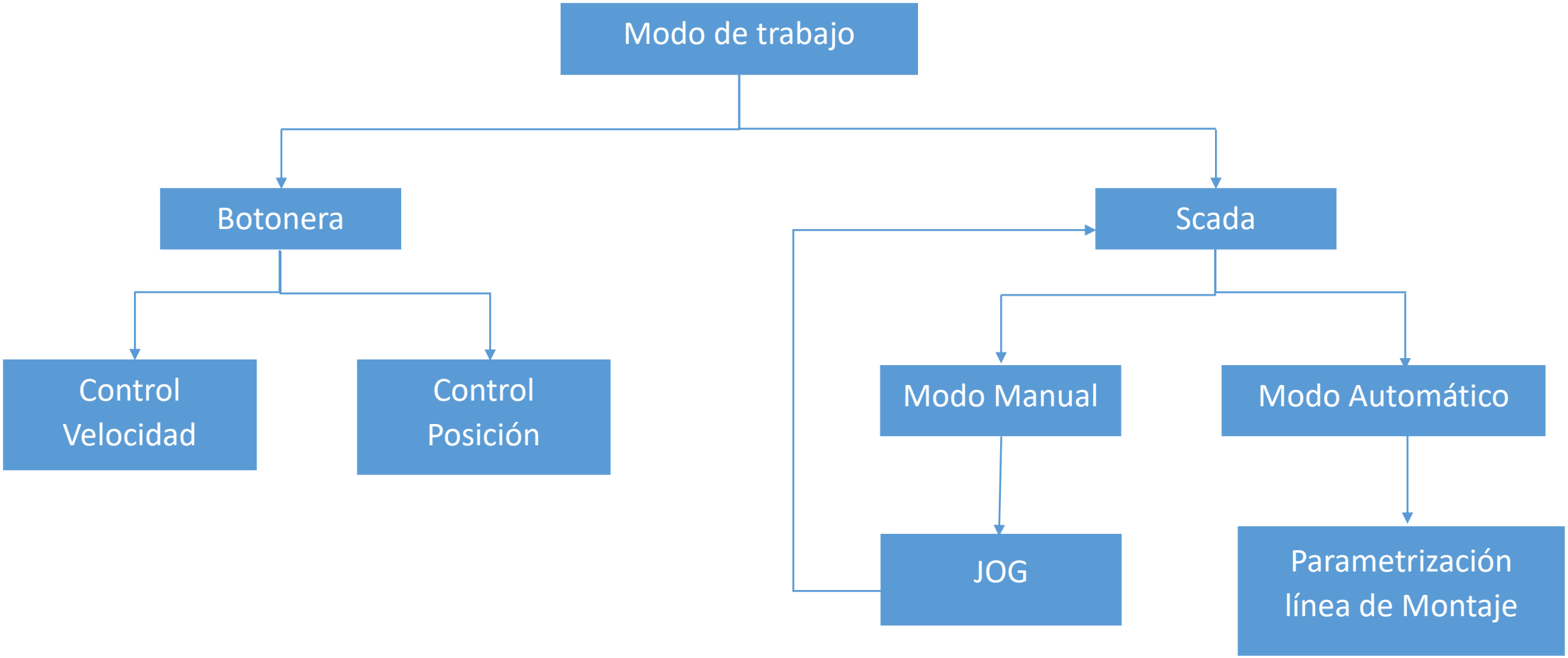
Por último, se acoplará el servomotor a la bancada o eje lineal, realizando la instalación que sea necesaria para el correcto funcionamiento de los elementos que lo comprenden y poder realizar una demostración del que trabajo que se ha realizado, ha sido el correcto.

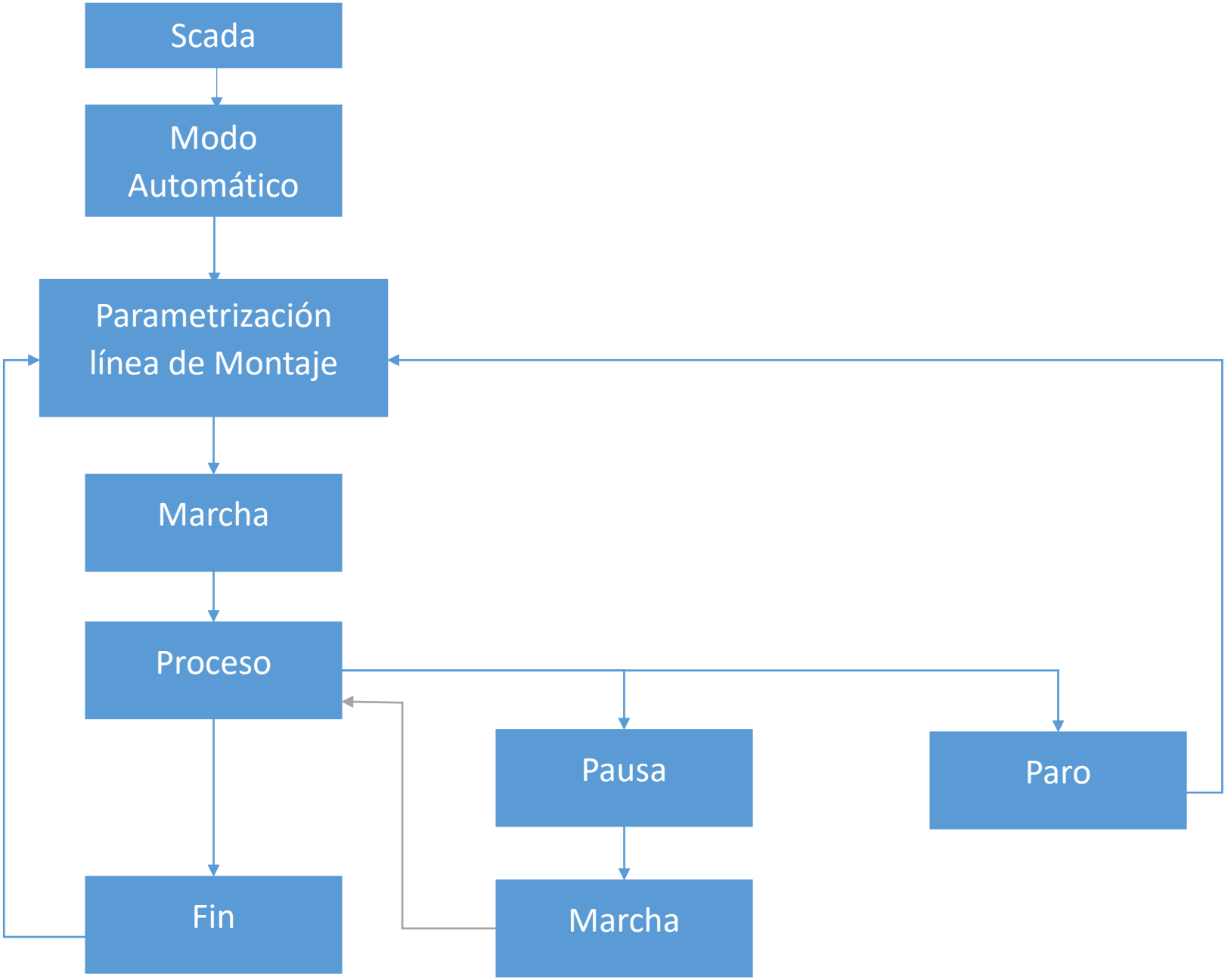
Anexo

Índice

1. Flujograma

En el primer punto que conforma nuestro anexo, mostraremos un flujograma a continuación, en donde se muestra de una forma esquemática el funcionamiento que queremos que tenga nuestro sistema.


Modo de trabajo
Botonera
Scada
Control Velocidad
Control Posición
Modo Manual
Modo Automático
JOG
Parametrización línea de Montaje

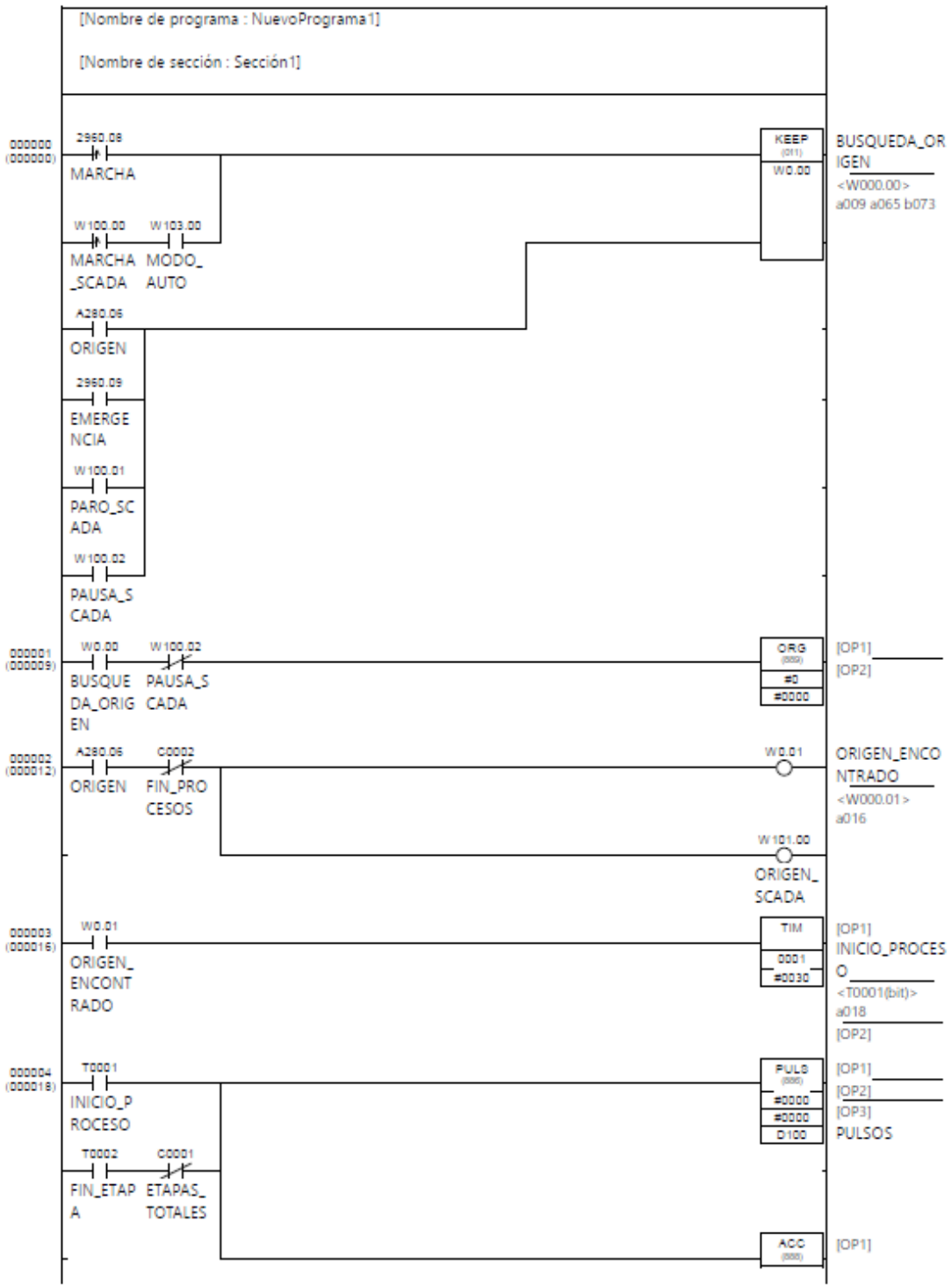

Scada
Modo Automático
Parametrización línea de Montaje
Marcha
Proceso
Pausa
Paro
Fin
Marcha

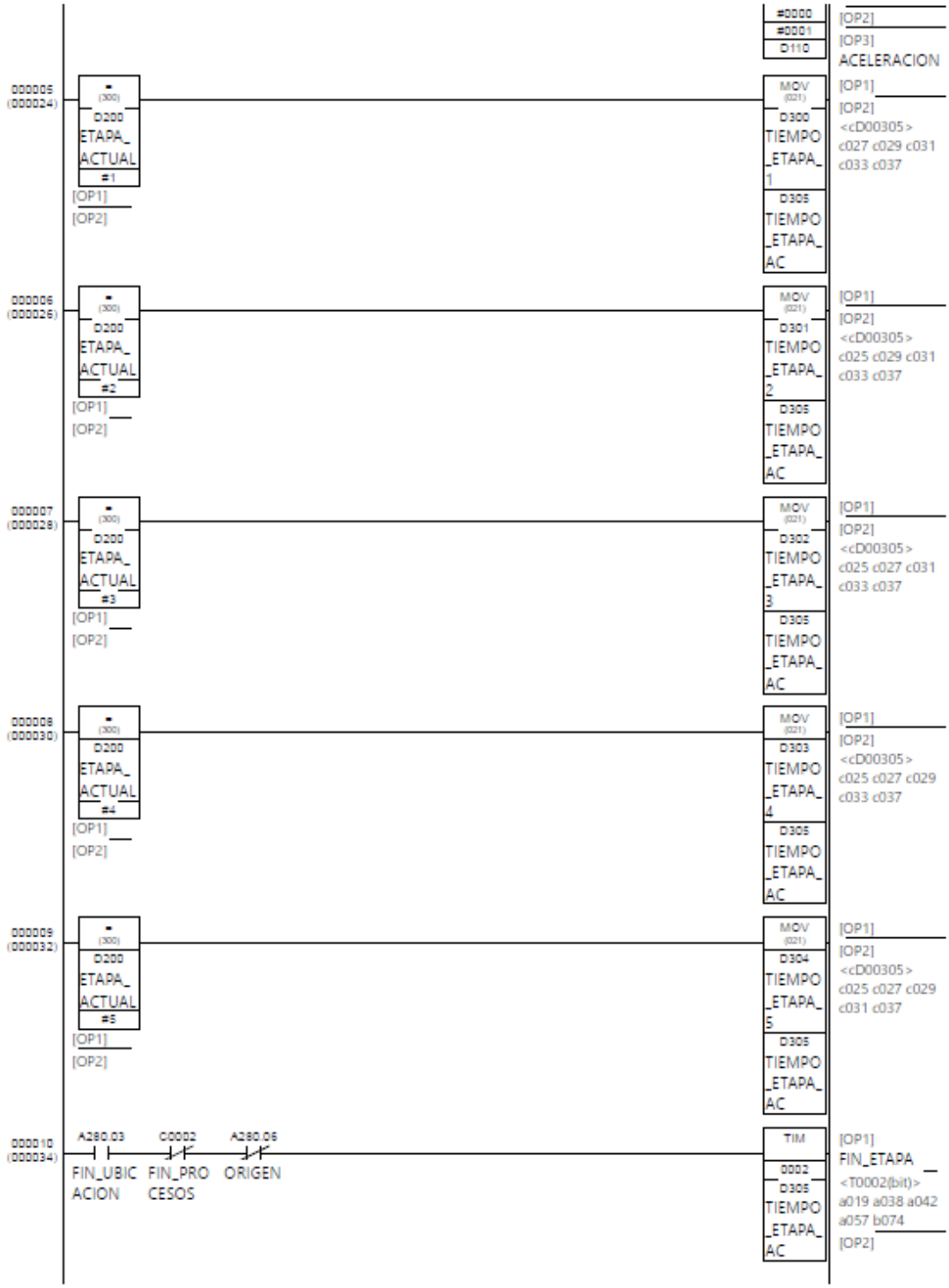
2. Programa

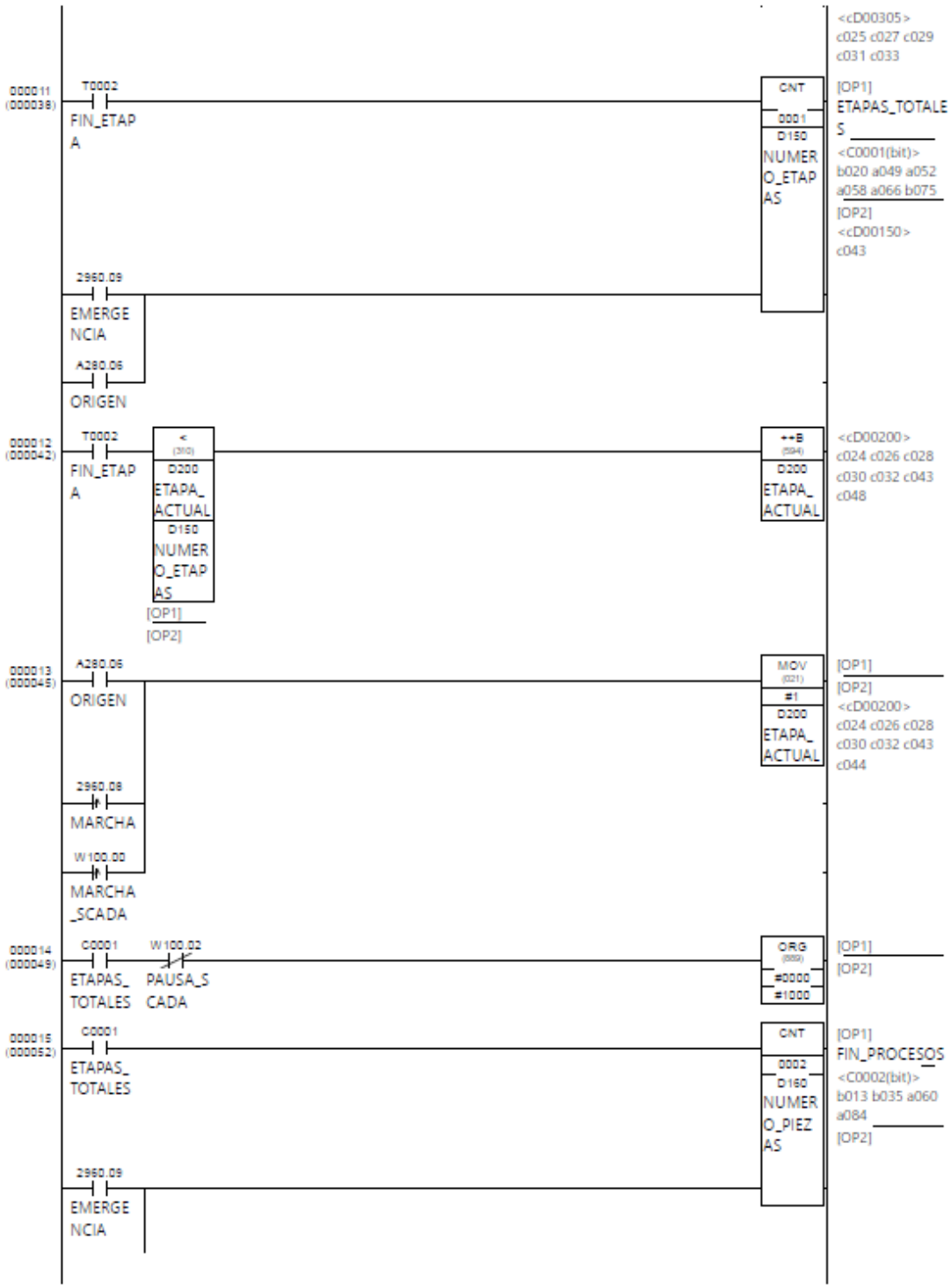
En el este punto del anexo, veremos el programa que se ha realizado con el CX-Programmer para el funcionamiento del sistema de trabajo de la línea de producción que queremos simular. Como hemos comentado previamente en la memoria, el lenguaje es de tipo Ladder o escalera, como es más comúnmente conocido.

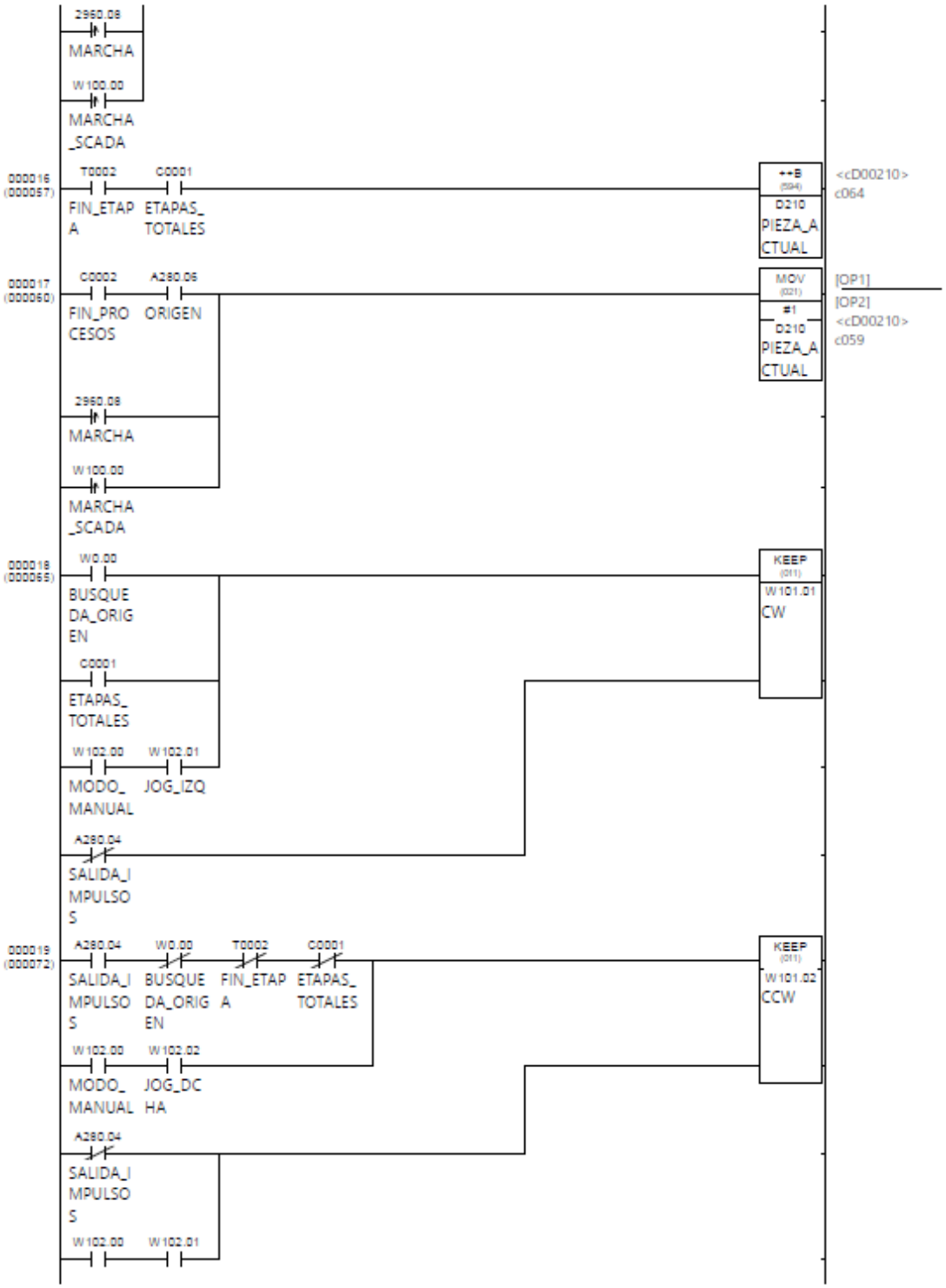
A continuación, mostramos el programa completo y posteriormente explicaremos cual es el funcionamiento de cada parte de cada línea que lo conforma.

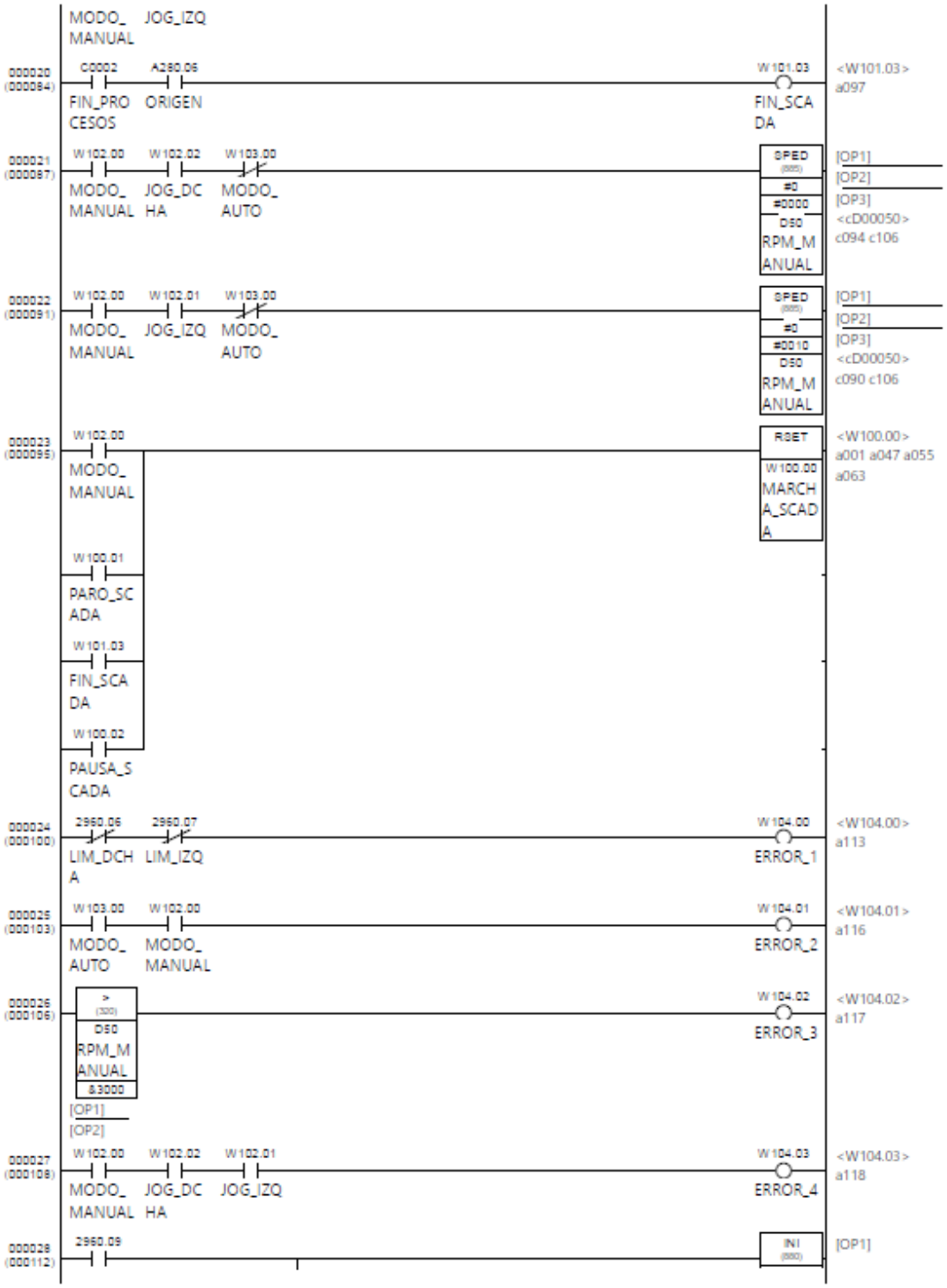
La obtención del programa en este formato se ha realizado con la opción del CX-Programmer de imprimir, puesto que permite pasar el programa en un formato PDF. Sin embargo, debido a que la exportación de PDF a Word no ha salido como esperábamos, hemos debido realizar capturas del PDF, teniendo el resultado que se puede apreciar a partir de la próxima página del proyecto.

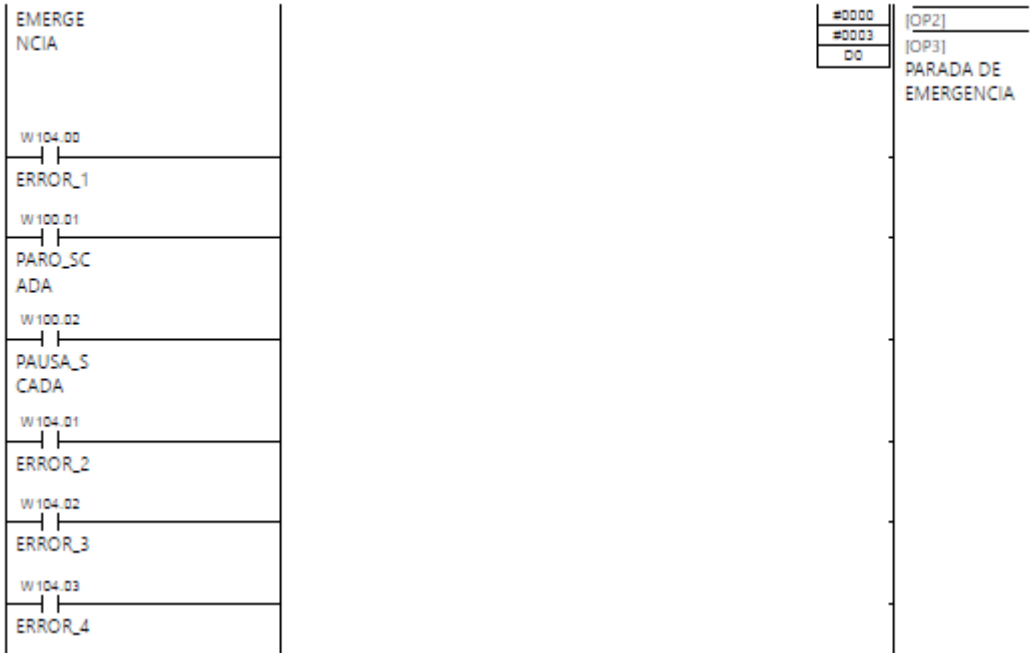

[Nombre de programa : NuevoPrograma1]
[Nombre de sección : Sección1]
000000 (000000)
2960.08
MARCHA
W100.00
MARCHA_SCADA
W103.00
MODO_AUTO
KEEP (011)
W0.00
BUSQUEDA_ORIGEN
<W000.00>
a009 a065 b073
A280.06
ORIGEN
2960.09
EMERGENCIA
W100.01
PARO_SCADA
W100.02
PAUSA_SCADA
000001 (000009)
W0.00
BUSQUEDA_ORIGEN
W100.02
PAUSA_SCADA
ORG (055)
#0
#0000
[OP1]
[OP2]
000002 (000012)
A280.06
ORIGEN
C0002
FIN_PROCESOS
W0.01
ORIGEN_ENCONTRADO
<W000.01>
a016
W101.00
ORIGEN_SCADA
000003 (000016)
W0.01
ORIGEN_ENCONTRADO
TIM (550)
0001
#0030
[OP1]
INICIO_PROCESO
<T0001(bit)>
a018
[OP2]
000004 (000018)
T0001
INICIO_PROCESO
T0002
FIN_ETAPA
C0001
ETAPAS_TOTALES
PULS (886)
#0000
#0000
D100
[OP1]
[OP2]
[OP3]
PULSOS
ACC (888)
[OP1]


#0000
#0001
D110
[OP2]
[OP3]
ACELERACION
000005
(000024)
(300)
D200
ETAPA_ACTUAL
#1
[OP1]
[OP2]
MOV (021)
D300
TIEMPO_ETAPA_1
D305
TIEMPO_ETAPA_AC
[OP1]
[OP2]
<cD00305>
c027 c029 c031
c033 c037
000006
(000026)
#2
D301
TIEMPO_ETAPA_2
c025 c029 c031
c033 c037
000007
(000028)
#3
D302
TIEMPO_ETAPA_3
c025 c027 c031
c033 c037
000008
(000030)
#4
D303
TIEMPO_ETAPA_4
c025 c027 c029
c033 c037
000009
(000032)
#5
D304
TIEMPO_ETAPA_5
c025 c027 c029
c031 c037
000010
(000034)
A280.03
FIN_UBICACION
C0002
FIN_PROCESOS
A280.06
ORIGEN
TIM
0002
FIN_ETAPA
<T0002(bit)>
a019 a038 a042
a057 b074


000011 (000038)
T0002
FIN_ETAPA
CNT
0001
D150
NUMERO_ETAPAS
[OP1]
ETAPAS_TOTALES
<C0001(bit)> b020 a049 a052 a058 a066 b075
[OP2]
<cD00150> c043
<cD00305> c025 c027 c029 c031 c033
2960.09
EMERGENCIA
A280.06
ORIGEN
000012 (000042)
T0002
FIN_ETAPA
<
(310)
D200
ETAPA_ACTUAL
D150
NUMERO_ETAPAS
[OP1]
[OP2]
+B
(594)
D200
ETAPA_ACTUAL
<cD00200> c024 c026 c028 c030 c032 c043 c048
000013 (000045)
A280.06
ORIGEN
2960.08
MARCHA
W100.00
MARCHA_SCADA
MOV
(021)
#1
D200
ETAPA_ACTUAL
[OP1]
[OP2]
<cD00200> c024 c026 c028 c030 c032 c043 c044
000014 (000049)
C0001
ETAPAS_TOTALES
W100.02
PAUSA_SCADA
ORG
(800)
#0000
#1000
[OP1]
[OP2]
000015 (000052)
C0001
ETAPAS_TOTALES
CNT
0002
D160
NUMERO_PIEZAS
[OP1]
FIN_PROCESOS
<C0002(bit)> b013 b035 a060 a084
[OP2]
2960.09
EMERGENCIA


2960.08
MARCHA
W100.00
MARCHA_SCADA
000016 (000057)
T0002
FIN_ETAPA
C0001
ETAPAS_TOTALES
++B (594)
D210
PIEZA_ACTUAL
<cD00210>
c064
000017 (000060)
C0002
FIN_PROCESOS
A280.06
ORIGEN
MOV (021)
#1
D210
PIEZA_ACTUAL
[OP1]
[OP2]
<cD00210>
c059
2960.08
MARCHA
W100.00
MARCHA_SCADA
000018 (000065)
W0.00
BUSQUEDA_ORIGEN
KEEP (011)
W101.01
CW
C0001
ETAPAS_TOTALES
W102.00
MODO_MANUAL
W102.01
JOG_IZQ
A280.04
SALIDA_IMPULSOS
000019 (000072)
A280.04
SALIDA_IMPULSOS
W0.00
BUSQUEDA_ORIGEN
T0002
FIN_ETAPA
C0001
ETAPAS_TOTALES
KEEP (011)
W101.02
CCW
W102.00
MODO_MANUAL
W102.02
JOG_DCHA
A280.04
SALIDA_IMPULSOS
W102.00
W102.01


MODO_ MANUAL
JOG_IZQ
000020 (000084)
C0002 FIN_PRO CESOS
A280.06 ORIGEN
W101.03 FIN_SCA DA
<W101.03> a097
000021 (000087)
W102.00 MODO_ MANUAL
W102.02 JOG_DC HA
W103.00 MODO_ AUTO
SPED
#0
#0000
D50 RPM_M ANUAL
[OP1]
[OP2]
[OP3]
<cD00050> c094 c106
000022 (000091)
W102.00 MODO_ MANUAL
W102.01 JOG_IZQ
W103.00 MODO_ AUTO
SPED
#0
#0010
D50 RPM_M ANUAL
[OP1]
[OP2]
[OP3]
<cD00050> c090 c106
000023 (000095)
W102.00 MODO_ MANUAL
RSET
W100.00 MARCH A_SCAD A
<W100.00> a001 a047 a055 a063
W100.01 PARO_SC ADA
W101.03 FIN_SCA DA
W100.02 PAUSA_S CADA
000024 (000100)
2960.06 LIM_DCH A
2960.07 LIM_IZQ
W104.00 ERROR_1
<W104.00> a113
000025 (000103)
W103.00 MODO_ AUTO
W102.00 MODO_ MANUAL
W104.01 ERROR_2
<W104.01> a116
000026 (000106)
> (320)
D50 RPM_M ANUAL
#3000
[OP1]
[OP2]
W104.02 ERROR_3
<W104.02> a117
000027 (000108)
W102.00 MODO_ MANUAL
W102.02 JOG_DC HA
W102.01 JOG_IZQ
W104.03 ERROR_4
<W104.03> a118
000028 (000112)
2960.09
INI
[OP1]



Comentando el funcionamiento del programa, iniciamos con la búsqueda del origen, la cual empezará con una función SR o KEEP como se denomina en el software de programación de Omron, la cual en caso de activar la marcha física desde la botonera o la desde la pantalla, siempre y cuando esta se encuentre con el modo automático activa, activará una memoria auxiliar, que denominamos BUSQUEDA_ORIGEN que activará la función de ORG. Como observamos, la memoria auxiliar se resetea en caso de encontrar el origen, o en caso de haber una emergencia, seleccionar paro o pausa, y de esta forma pararíamos la búsqueda del origen.

En la siguiente parte, cuando nos encontramos en el origen y siempre y cuando no nos encontremos con que hemos terminado el proceso, se activa otra memoria auxiliar, que activará un temporizador (TIM en CX-Programmer). Además, activaremos una memoria que utilizamos como la señal luminosa en el scada para indicar que nos encontramos en el origen.

Cuando termina el primer temporizador, el cual dura 3 segundos, se activa el tren de pulsos que nos llevará a cada una de las estaciones. También, en caso de que acabemos una etapa y siempre y cuando no se hayan realizado todas las etapas que hemos programado, volverá activarse el tren de pulsos que acciona el motor. Para nuestro proyecto, la distancia que hay entre estación y estación, será siempre la misma.

En la parte de los comparadores, tenemos como intención ir cambiando el tiempo de etapa según sea el valor de la etapa actual. Como en nuestro caso, había un número máximo de 5 etapas, esta solución se puede utilizar. Para el caso de un mayor número de etapa, habría que utilizar otro método óptimo y que haga que el programa funcione de una misma forma. Para cambiar el valor del tiempo de etapa actual, hacemos uso de la instrucción MOV.

El siguiente paso que realiza el programa es la activación de un segundo temporizador, el cual es el que hace que las etapas se activen. Este se activará cuando se produzca el final de ubicación

de la pieza y siempre y cuando no se haya acabado el proceso y no sea cuando nos encontramos en el punto de origen. Con esto activaremos el temporizador 2.

Con la activación de la variable que genera el temporizador 2, la cual es la de final de etapa, vamos a un contador que utilizamos para ir aumentando de unidad en unidad el valor del número de etapa actual. Como se observa, el contador se resetea, haciendo que el valor de etapa actual vuelva a la unidad en caso de haber una emergencia o se active el punto de origen.

Después usamos otro contador, aunque es distinto (++B), que hace que aumente el valor de etapa actual, que esta variable es la que utilizamos para mostrar el valor de la etapa actual en la pantalla de control y monitorización, siempre y cuando la etapa actual tenga un valor inferior al numero de etapa que hayamos programado. En caso de encontrar el origen, o seleccionar marcha, este valor pasa a ser la unidad.

En la siguiente fila, vemos que cuando se ha cumplido el contador 1 del número de etapas que hemos seleccionado, y siempre y cuando no tengamos activo el botón de pausa del scada, se activa la instrucción de vuelta al origen.

Como en el caso de las etapas, también tenemos que utilizar contador para el número de piezas, el cual cuenta cada vez que la variable del contador 1, que nos indica que hemos realizado todas las etapas que hemos programado, se activa. El contador se resetea cuando entra una emergencia o ponemos en marcha el proceso.

De la misma forma que con anterioridad, usamos una memoria para mandar el valor de la pieza actual al scada, el cual aumenta cuando se ha producido el fin de la última, que lo hacemos con la activación de la variable de fin de etapa y la variable de etapas totales. Para que el valor vuelva a la unidad, se debe estar en el final del proceso y llegar al origen, o con la activación de las variables de marcha, tanto la física como la de pantalla.

Los siguientes bloques los utilizamos para las señalizaciones luminosas. En primer lugar, tenemos varios SR para indicar los sentidos de giro que realiza el motor. La señal de giro de derecha se produce siempre que estamos en la búsqueda del origen y cuando retornamos a este. También se encuentra aquí el hecho que se ilumina la señal cuando trabajamos hacia este sentido en modo manual. Del mismo modo, tenemos el indicador del sentido a izquierda, el cual se activa cuando se está produciendo una salida de impulsos, siempre y cuando esta no sea la de búsqueda o retorno al origen o en caso de que estemos trabajando en modo manual y queramos que el motor gire hacia este sentido. En ambos casos, para resetear esta variable auxiliar, será cuando no se produzca una salida de pulsos, pues suponemos que el motor se encuentra estático en ese momento.

La siguiente acción es aquella que utilizamos para activar la señal luminosa de fin del proceso, que se activa cuando nos encontramos en el origen y el proceso se ha finalizado.

El siguiente bloque trata del modo manual o JOG, que es un control de velocidad. En él, siempre que tengamos activo el modo manual e indiquemos el sentido de giro que queramos, y no esté el modo automático activo, entra la instrucción de SPED. La configuramos en cada caso para que gire en un sentido u otro.

La siguiente instrucción, la utilizamos para únicamente resetear la marcha del scada, que será en caso de activar el modo manual, el paro, cuando se finalice el proceso o se active la pausa de la pantalla.

Los siguientes bloques son para la señalización de los diferentes errores que hemos tenido en cuenta en nuestro proyecto, que como podemos observar son la activación de los dos límites de carrera en la misma situación, la activación de los dos modos de funcionamiento, superar la programación de velocidad en modo manual y activar los dos sentido de giro en el modo manual.

Por último, tenemos la instrucción de paro del sistema que se denomina INI en CX-Programmer, la cual se activa cuando se produce una emergencia, se activa cualquiera de los errores o seleccionamos la pausa o el paro.

3. Funcionamiento con botonera

Como ya hemos comentado con anterioridad, antes de colocar el servomotor acoplado al eje lineal, se decidió realizar varias pruebas de funcionamiento con el uso de la botonera indicando con esta los finales de carrera, la indicación de origen, etc.

Ilustración 75. Imagen de la botonera

Para realizar los dos tipos de controles que queremos realizar con la botonera, deberemos colocar en el servo drive, en la variable Pn000.1 el valor 7. Con esto, según la posición del botón "P-CON", conseguiremos que se trabaje con control de posición o con control de velocidad.

Comenzamos explicando el control de velocidad, por lo que lo primero que haremos será activar la alimentación hacia el servomotor con el botón "S-ON" y teniendo los límites desconectados. Para hacer girar el motor, movemos el potenciómetro de izquierda a derecha, empezando con el potenciómetro en una posición intermedia que indicará que nos encontramos en velocidad de 0 rpm. Según el sentido que giremos el potenciómetro, haremos que el motor gire a un lado u otro de la marcha. Si simulamos la llegada a un límite, lo cual se hace tanto con el botón "P-OT/ IN-6", como con el "N-OT/IN-7" , según el giro del motor, este se frenará en seco.

Cuando trabajamos en modo de posición, debemos cambiar la posición del "P-CON", y como hemos configurado que el tren de pulsos en el servopack entra de forma CW y CCW, utilizaremos los botones "PULS" para ir en sentido CW y el botón "SIGN" para en sentido CCW. De la misma forma que en el anterior control, en caso de simular una llegada al límite, el motor se para automáticamente.

Cuando trabajemos para hacer la simulación del funcionamiento de la línea de montaje, usaremos el botón "X (24)" como indicador de punto de origen, el botón "IN-8" como marcha y el botón "IN-9" como paro.

En caso de que el servopack entre en modo de seguridad por que ha detectado este que ha habido un problema, para quitar la señal de error y poder volver a trabajar, activaremos y desactivaremos el botón que indica "ALMRS".

Con esto finaliza la explicación del funcionamiento con la botonera que hemos realizado en el presente proyecto.

4. Pantalla en funcionamiento local

En este apartado mostraremos la pantalla que realizamos con el programa de la empresa Omron CX-Supervisor y que nos ha servido para trabajar en modo local con el autómata SYSMAC CJ1M CPU21.

A continuación, se muestra una imagen completa del pantalla que se ha diseñado.

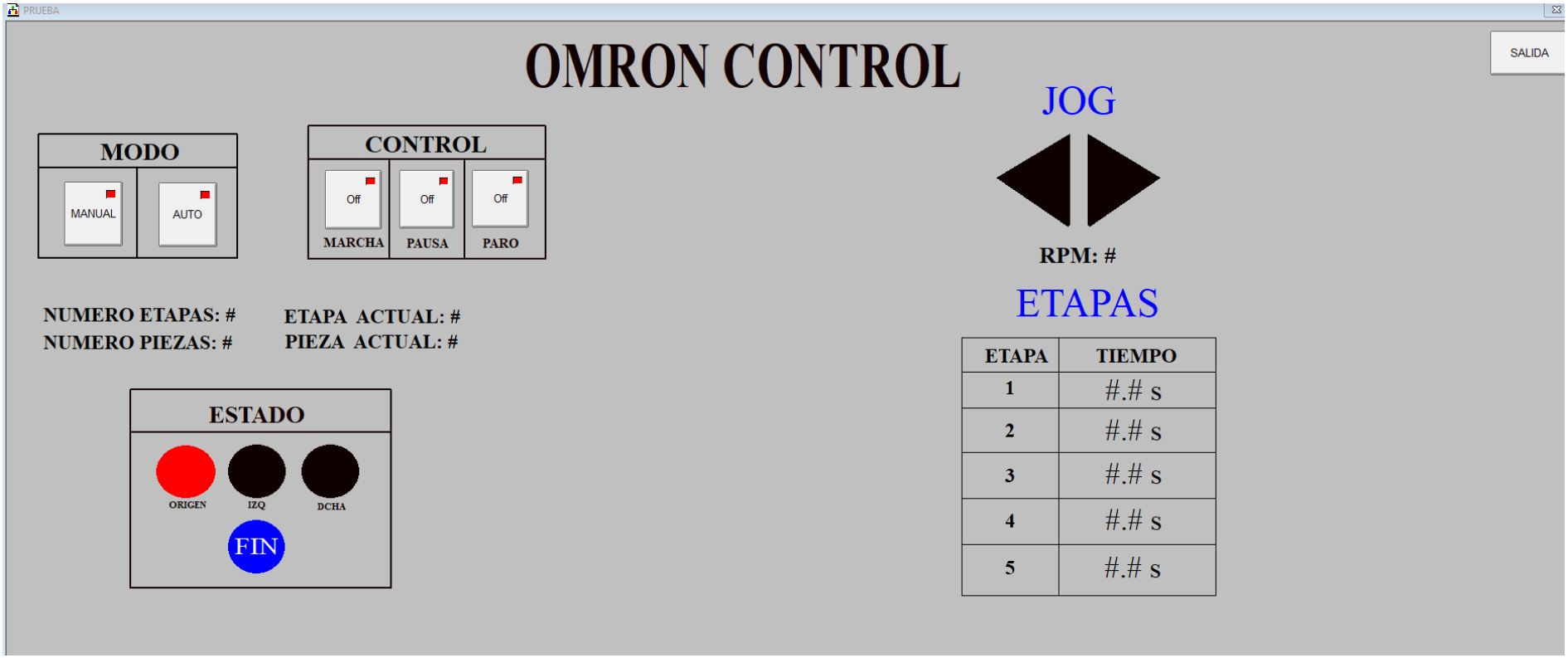


Ilustración 76. Scada diseñado con el CX-Supervisor

Comenzaremos la explicación del funcionamiento de este primer scada que diseñamos para una conexión directa con el PLC con el modo manual. Cuando seleccionamos este modo, emerge el texto en el visualizamos la palabra “JOG”, al igual que las dos flechas que indican el sentido de giro que queremos que vaya el servomotor y también en donde podemos escribir a la velocidad a la que queremos que vaya este. Para indicar a la velocidad que queremos que gire, cuando seleccionamos el valor que nos aparezca en estado run, emergerá una ventana en la que podremos escribir la velocidad. En caso de poner una velocidad que no se encuentre entre los intervalos de 0 y 3000 rpm, el programa nos indicará un error y nos pedirá que cambiemos el valor por uno que se encuentre en el intervalo. A la hora de seleccionar el sentido que queremos que gire el motor, seleccionamos con el cursor la flecha y automáticamente esta se nos activará. Cuando una flecha se encuentre activa, la otra se encontrará en estado de inhibición y por tanto no podremos seleccionarla. Esto es un método de seguridad que también hemos querido incorporar.

Con respecto al modo de funcionamiento automático, cuando lo seleccionemos, emergerán tanto los botones de control, como la zona en la que podemos modificar los valores de los tiempos de etapa. El número que nos indica en qué etapa nos encontramos, se irá iluminando según en qué etapa se encuentre en ese mismo momento la pieza en la línea de trabajo. La indicación del tiempo de etapa es exactamente igual que el que realizamos con la indicación de la velocidad a la que queremos que vaya el servomotor en modo manual. Al mismo tiempo, según el sentido de la marcha del motor, indicará en la zona de estado, se encenderá un indicador u otro. Estas son de color negro cuando se encuentran desactivadas y cambian a un color verde cuando su estado es activo. De la zona de estado, también tenemos los indicadores de punto de origen, que se activará cada vez que nos ubiquemos en él, la cual es de color rojo cuando se encuentra desactivada y de color verde cuando nos encontramos en este punto; y el de fin, que se trata de una luz azul con un letrero en blanco con la palabra “FIN” en su interior. En la parte superior del estado, nos encontramos en la zona en la que indicamos cuantas etapas y número de piezas queremos. Para indicarlas, de la misma forma que hemos explicado previamente, pues debemos seleccionar el valor y poner un resultado que se encuentre dentro del rango que hemos configurado, el cual ha sido de entre 1 y 5 para ambas variables. También podremos monitorizar en todo momento en que etapa y pieza nos encontramos durante el proceso. Para finalizar la parte del modo automático, tenemos la parte de control, en la que podemos diferenciar los tres botones que la componen: “Marcha”, “Pausa” y “Paro”. La primera de ellas es para poner en marcha el sistema, siempre y cuando este se encuentre en situación de poder ponerse en funcionamiento. Para las otras dos, ambas paran el sistema en el momento, aunque las diferencia el hecho de que pausa para, pero permite retomar el funcionamiento en donde nos hemos quedado, mientras que el caso de paro pone todo a cero y se debe volver a empezar con la búsqueda de origen.

Como observamos arriba a la derecha, tenemos el botón de salida que nos permitirá abandonar la pantalla cuando nos encontremos en modo “RUN” y nos sea necesario tener que pararla.

5. Pantalla en funcionamiento OPC

Este scada que hemos realizado con la aplicación gratuita de Visual Basic, tal como hemos explicado en la parte de la memoria, su funcionamiento se realiza con la utilización de un servidor OPC. La idea principal que debemos tener en cuenta es el hecho de que la herramienta Visual Basic es gratuita y que nos permite un mismo funcionamiento que con la que hemos creado con CX-Supervisor. Otra ventaja que nos ofrece, y como hemos repetido con anterioridad, es que, en el futuro, este scada podrá utilizarse con otros autómatas fuera que sean de otras marcas, permitiendo crear una pantalla de monitorización y control genérico para PLC que trabajen con OPC.

Seguidamente, mostramos la pantalla que hemos diseñado con Visual Basic.

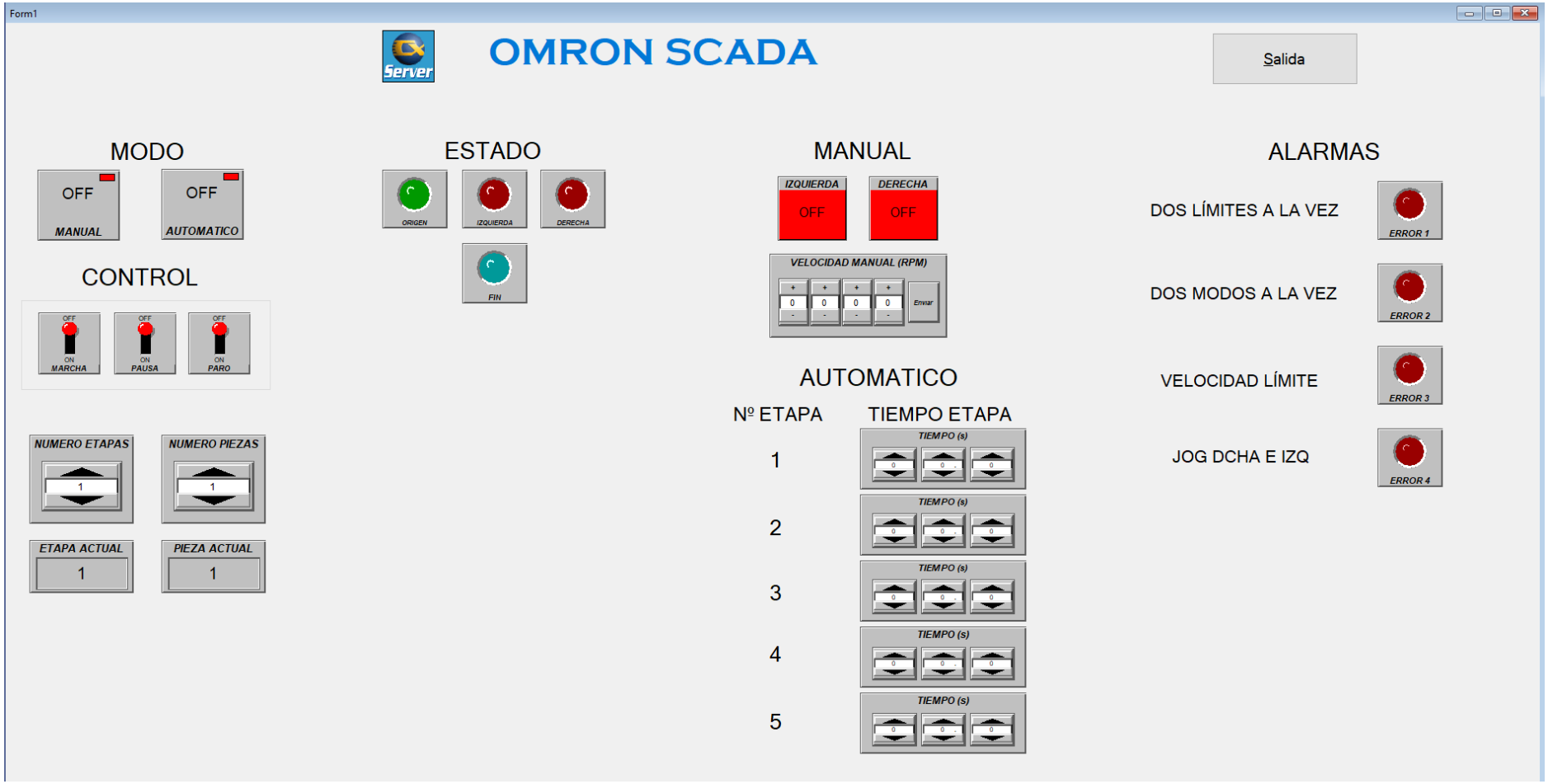


Ilustración 77. Scada en Visual Basic

Como podemos observar, la pantalla que hemos diseñado es parecida a la que hemos diseñado con anterioridad para que trabajará en modo local. Sin embargo, debemos tener en cuenta que al ser un software genérico Visual Basic, hay acciones que no nos permite realizar como en el caso del CX-Supervisor. Una de las diferencias es el hecho de no haber ocultado partes concretas de la pantalla según el modo en el que nos encontremos. Otro es como podemos seleccionar el número de etapas, de piezas, tiempos y velocidad en modo manual, pues como podemos observar, lo deberemos realizar con botones que incorporan unas flechas en donde podemos aumentar o disminuir el valor. En el caso de la velocidad en modo manual, además se ha añadido un botón que indica “Enviar”, por motivos de seguridad, para que, en caso de un valor erróneo, el sistema nos indique con un letrero que ese valor no es el adecuado. Para las otras variables, al estar sus límites muy próximos, 1 y 5, simplemente cuando lleguemos alguno de estos valores, pese a que vayamos a intentar aumentar o disminuir, según el límite que nos encontremos, el valor no variara. Un añadido que hemos realizado en este scada han sido los indicadores de errores, que son señales luminosas que se activan según el error que se haya producido y así averiguar con mayor facilidad, cual ha sido el problema que ha parado nuestro sistema. Para este caso, tampoco se iluminan los números de las etapas según nos encontremos. Sin embargo, este scada nos permite realizar el mismo funcionamiento que con el otro tipo.

6. Cálculo de paso por pulso

En este apartado mostraremos el paso por pulso que hemos realizado para nuestro sistema. Para ello, debemos tener en cuenta que tal como aparece en la memoria, en el apartado de configuración del servo drive, debemos parametrizar las variables Pn202 y Pn203.

Debemos cumplir con la siguiente norma:

$$2048 \geq Pn202 \geq Pn203 \geq 1$$

Como la resolución de nuestro encoder es de 13 bits y queremos que el paso de nuestro servomotor por pulso sea de 10°, deberemos realizar el siguiente cálculo:

$$\frac{Pn202}{Pn203} = \frac{2^{13}}{N} \cdot Unidad = \frac{8192}{Distancia} \cdot Paso$$

$$\frac{Pn202}{Pn203} = \frac{8192}{360} \cdot 10 = 227.5556$$

Como debemos darles un valor entero a las dos variables, si le damos el valor máximo a Pn202, 2048, obtenemos que el valor siguiente para Pn203:

$$Pn203 = \frac{Pn202}{227.5556} = \frac{2048}{227.5556} = 9$$

Por tanto, para conseguir que el servomotor gire 10° por cada pulso que nosotros le mandemos, los valores de las dos constantes deben ser 2048 y 9 respectivamente.

7. Bibliografía

En esta parte, mostraremos todos aquellos sitios en donde hemos obtenido información para la realización del presente proyecto.

- Manual autómata Omron SYSMAC CJ1M CPU21.

http://www.infoplc.net/files/descargas/omron/infoPLC_net_CJ1M_CPU2x%20_IO_Incorporadas.pdf

- Manual del usuario. Diseño y Mantenimiento del servomotor SGMAH y del servopack SGDH.

file:///C:/Users/Miguel/Downloads/Espanol_Diseno_Y_mantenimiento(SIS-S800-32-2).pdf

- Matrikon OPC

https://www.matrikonopc.es/index.aspx

- Poliformat

https://poliformat.upv.es/portal/site/GRA_12022_2018

https://poliformat.upv.es/portal/site/GRA_12028_2018

https://poliformat.upv.es/portal/site/GRA_12022_2018

- YouTube

https://www.youtube.com/watch?v=hGV9-kx_d_U

https://www.youtube.com/user/cavanilles2009

Índice Imágenes