

Proyecto Fin de Carrera

Ingeniería de las Tecnologías Industriales

Servomotor Industrial. Puesta en Servicio y Supervisión a través de un Autómata Programable

Autor: Antonio Calvo Cid

Tutor: Luis Fernando Castaño Castaño

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022



Proyecto Fin de Carrera
Ingeniería de las Tecnologías Industriales

Servomotor Industrial. Puesta en Servicio y Supervisión a través de un Autómata Programable

Autor:

Antonio Calvo Cid

Tutor:

Luis Fernando Castaño Castaño

Doctor Ingeniero Industrial

Dpto. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022

Proyecto Fin de Carrera: Servomotor Industrial. Puesta en Servicio y Supervisión a través de un Autómata Programable

Autor: Antonio Calvo Cid

Tutor: Luis Fernando Castaño Castaño

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

Agradecimientos

Se acaban al fin estos cuatro años de duro esfuerzo y sacrificio. Se cierra una etapa pero otra nueva comienza, por lo que nada me haría más feliz que agradecer a los que me han ayudado a llegar hasta aquí y que seguro que lo harán en lo que está por llegar.

A mis padres, por interesarse siempre en mí. A mis hermanas, que han sabido comprenderme y animarme. A mis amigos, por ser la ruta de escape en momentos de agobio y estrés.

Por último, agradecer también a mi tutor, Luis Fernando Castaño Castaño, por la ayuda recibida desde el primer momento.

Antonio Calvo Cid

Sevilla, 2022

Resumen

De entre todas las formas de control que se conocen en la actualidad, el control de servomotores es la opción que permite mayor precisión de movimiento. Es por ello por lo que la industria aumenta la demanda de estos dispositivos dejando atrás en ciertas aplicaciones a los motores paso a paso, por ejemplo.

Este proyecto se centrará en cómo poner en funcionamiento un servomotor, concretamente de la familia BSH0552T01A1A, que se empleará para el movimiento de una cinta y que se conectará al driver Lexium 32M.

Una vez conformado el servoaccionamiento, se hará un desarrollo sobre el uso del software SoMove, de Schneider Electric, para modificar tanto el modo de funcionamiento así como cualquier parámetro del servomotor. Se profundizará en todas las funcionalidades que ese software ofrece y se realizarán unos ejercicios prácticos centrados en este software con la finalidad de conseguir un determinado movimiento del servomotor.

Una vez familiarizado con SoMove, se desarrollarán otra serie de ejercicios prácticos, esta vez centradas en la ejecución de funciones destinadas para servomotores desarrolladas en Unity Pro, por lo que el control del movimiento pasará a estar en manos de un autómata. Se explicarán todos los aspectos necesarios para una completa comprensión de cada ejercicio como son configuración de parámetros, conexionado y modo de empleo de estas librerías.

Abstract

Among all the forms of control that we know, servomotor control is the option that allows greater precision of movement. That is why the industry increases the demand for these devices, leaving stepper motors behind in certain applications.

In this project we will focus on how to start up a servomotor, specifically from the BSH0552T01A1A family, which will be used to move a tape, and which will be connected to the Lexium 32M driver.

Forming the servo drive, we will rely on the SoMove software, from Schneider Electric, to modify both the operating mode as well as any parameter of the servomotor. All the functionalities that this software can provide will be delved into and finally practical exercises will be carried out to achieve a certain movement of the servomotor.

Once familiar with SoMove, another series of practical exercises are developed by executing functions intended for servomotors and developed in Unity Pro, so that in this type of exercise the movement control will pass to it in the hands of an automaton. All aspects such as parameter configuration, connection and how to use these libraries will be explained.

Índice

Agradecimientos	7
Resumen	9
Abstract	11
Índice	12
Índice de Tablas	14
Índice de Figuras	15
1 Introducción	1
1.1 <i>Objetivos del proyecto</i>	2
1.2 <i>Procedimiento</i>	3
2 Servoaccionamiento con Lexium 32M	11
2.1 <i>Servo accionamiento</i>	11
2.1.1 <i>Servomotor</i>	11
2.1.2 <i>Servo drive</i>	12
2.1.3 <i>Conexionado</i>	13
2.2 <i>Primeros pasos con Lexium 32M</i>	14
2.2.1 <i>Interfaz HMI</i>	14
2.2.2. <i>Diagrama de estados</i>	15
2.2.3. <i>Errores de funcionamiento</i>	16
2.2.4. <i>Modos de funcionamiento.</i>	17
3 Autómata programable	28
3.1 <i>Configuraciones en Unity Pro</i>	28
3.2 <i>Conexión y comunicación</i>	29
3.3 <i>Unidades de programación</i>	32
3.4 <i>Modos de ejecución</i>	34
4 SoMove	35
4.1 <i>Primeros pasos en SoMove</i>	35
4.1.1 <i>Selección de dispositivo</i>	35
4.1.2 <i>Interfaz de SoMove</i>	37
4.1.3 <i>Conexión y modos de funcionamiento</i>	40
5 Unidades prácticas. Uso del servomotor mediante señales externas	42
5.1 <i>Ejemplo introductorio I</i>	42
5.1.1 <i>Enunciado</i>	42
5.1.2 <i>Solución propuesta</i>	42
5.2 <i>Ejemplo introductorio II</i>	43
5.2.2 <i>Enunciado</i>	43

5.2.3	Solución propuesta	43
5.3	<i>Unidades prácticas. Implementación de librerías.</i>	45
5.3.2	Unidad práctica I	46
5.3.3	Ejemplo práctico II	52
5.3.4	Ejemplo práctico III	54
5.3.5	Ejemplo práctico IV	55
6	Anexos	58
6.0	<i>Introducción</i>	58
6.1	<i>Ejemplo práctico I</i>	58
6.2	<i>Ejercicio práctico II</i>	60
6.3	<i>Ejemplo práctico III</i>	63
6.4	<i>Ejemplo práctico IV</i>	66
	Referencias	69

ÍNDICE DE TABLAS

Tabla 2-1. Características del servomotor BSH0551T01A2A.	11
Tabla 2-2. Canales del servo drive Lexium 32M.	13
Tabla 2-3. LEDs de estados del LXM32M.	15
Tabla 2-4. LEDs de nivel del LXM32M.	15
Tabla 2-5. Estados de funcionamiento.	16
Tabla 2-6. Clases de error.	17
Tabla 2-7. Métodos homing.	26
Tabla 6-1. Entradas. Ejemplo práctico I.	58
Tabla 6-2. Salidas. Ejemplo práctico I.	58
Tabla 6-3. Variables internas. Ejemplo práctico I.	58
Tabla 6-4. Entradas. Ejemplo práctico II.	60
Tabla 6-5. Salidas. Ejemplo práctico II.	60
Tabla 6-6. Entradas/Salidas. Ejercicio práctico II.	60
Tabla 6-7. Variables internas. Ejemplo práctico II.	60
Tabla 6-8. Entradas. Ejemplo práctico III.	63
Tabla 6-9. Salidas. Ejemplo práctico III.	63
Tabla 6-10. Entradas/Salidas. Ejercicio práctico III.	63
Tabla 6-11. Variables internas. Ejemplo práctico III.	64
Tabla 6-12. Entradas. Ejercicio práctico IV.	66
Tabla 6-13. Salidas. Ejercicio práctico IV.	66
Tabla 6-14. Entradas/Salidas. Ejercicio práctico IV.	66
Tabla 6-15. Variables privadas. Ejercicio práctico IV.	66

ÍNDICE DE FIGURAS

Figura 1-1. Variador de frecuencia.	1
Figura 1-2. Motor paso a paso.	1
Figura 1-3. Funcionamiento motor paso a paso.	2
Figura 1-4. Servomotor.	2
Figura 2-1. Servomotor BSH0551T01A2A.	11
Figura 2-2. Lexium 32M. Placa de características.	12
Figura 2-3. Canales del servo drive Lexium 32M.	12
Figura 2-4. Conexión general entre servomotor y LXM32M.	13
Figura 2-5. Conexiones CN10 y CN11.	14
Figura 2-6. Conectores M23.	14
Figura 2-7. Interfaz HMI del LXM32M.	14
Figura 2-8. Display del LXM32M.	15
Figura 2-9. Diagrama de estados de funcionamiento.	16
Figura 2-10. Modo Jog. Movimiento continuo.	18
Figura 2-11. Modo Jog. Movimiento paso a paso.	18
Figura 2-12. Parámetros modo Jog.	19
Figura 2-13. Parámetros modo Electronic Gear.	20
Figura 2-14. Parámetros Torque Profile.	21
Figura 2-15. Parámetros Profile Velocity.	22
Figura 2-16. Parámetros Profile Position.	23
Figura 2-17. Interpolated Position.	23
Figura 2-18. Homing con final de carrera.	24
Figura 2-19. Homing con interruptor de referencia.	25
Figura 2-20. Homing con pulso índice.	25
Figura 2-21. Homing con establecimiento de medida.	25
Figura 2-22. Parámetros y secuencia modo homing.	26
Figura 3-1. Modicon M340.	28
Figura 3-2. Selección del autómatas programable.	29

Figura 3-3. Cable USB-Mini B.	29
Figura 3-4. Desplegable PLC.	30
Figura 3-5. Establecer dirección mediante USB con el autómata.	30
Figura 3-6. Establecer dirección mediante TCP-IP con el autómata.	30
Figura 3-7. Ventana BUS PLC.	31
Figura 3-8. Ventana de configuración del canal Ethernet.	31
Figura 3-9. Ventana de configuración de la IP del PLC.	31
Figura 3-10. Creación de bloques DBF.	32
Figura 3-11. Lenguaje de programación ST.	32
Figura 3-12. Lenguaje de programación LD.	33
Figura 3-13. Lenguaje de programación FBD.	33
Figura 3-14. Lenguaje de programación IL.	33
Figura 3-15. Llamada a función DFB.	34
Figura 3-16. Selector de modos de ejecución.	34
Figura 3-17. Botón de conexión con PLC.	34
Figura 3-18. Botones de transferencia de proyectos.	34
Figura 3-19. Botón de ejecución del proyecto.	34
Figura 4-1. Cable Modbus - USB.	35
Figura 4-2. Seleccionar dispositivo.	35
Figura 4-4-3. Crear topología.	36
Figura 4-4. Ventana Principal de SoMove.	36
Figura 4-5. Ventana "Lista de parámetros".	37
Figura 4-6. Reiniciar variador.	38
Figura 4-7. Ventana "Visualización".	38
Figura 4-8. Ventana "Grabación".	39
Figura 4-4-9. Ventana "Tuning".	39
Figura 4-10. Ventana "My Dashboard".	40
Figura 4-11. Barra de herramientas.	40
Figura 4-12. Ventana Connect.	40
Figura 4-13. Procedimiento de activación de la etapa de potencia.	41
Figura 4-14. Barra de modos de funcionamiento.	41
Figura 5-1. Panel de mandos.	42
Figura 5-2. Conexionado. Ejemplo I.	43
Figura 5-3. Configuración de parámetros. Ejemplo I.	43
Figura 5-4. Conexionado. Ejemplo II.	44
Figura 5-5. Configuración de parámetros. Ejemplo II.	45
Figura 5-6. Señales de control de un motor paso a paso.	45
Figura 5-7. Introducción de un nuevo dispositivo.	46
Figura 5-8. Selección del módulo EHC0200.	47
Figura 5-9. Configuración del canal 0.	47

Figura 5-10. Configuración del canal 1.	48
Figura 5-11. Estructura del canal 0.	48
Figura 5-12. Estructura del canal 1.	49
Figura 5-13. Estructura de estados.	49
Figura 5-14. Parámetros PTI.	50
Figura 5-15. Parámetros Electronic Gear. Ejercicio práctico I.	50
Figura 5-16. Conexión de salidas del EHC200 con CN5.	50
Figura 5-17. Conexión EHC0200.	51
Figura 5-18. Conexión general. Ejemplo práctico I.	51
Figura 5-19. Interfaz HMI. Ejemplo práctico I.	51
Figura 5-20. Configuración PTO. Ejemplo práctico II.	52
Figura 5-21. Conexión del Canal 1.	53
Figura 5-22. Conexión general. Ejemplo práctico II.	53
Figura 5-23. Interfaz HMI. Ejemplo práctico II.	53
Figura 5-24. Conexión Canal 6. Ejemplo práctico III.	54
Figura 5-25. Conexión general. Ejemplo práctico III.	55
Figura 5-26. Interfaz HMI. Ejemplo práctico III.	55
Figura 5-27. GearRatio vs Avance.	56
Figura 5-28. Lista de parámetros. Configuración de ejes.	56
Figura 5-29. Interfaz HMI. Ejemplo práctico IV.	57
Figura 6-1. Programación de la función. Ejemplo práctico I.	59
Figura 6-2. Programación de la función. Ejemplo práctico II.	63
Figura 6-3. Programación de la función. Ejemplo práctico III.	66
Figura 6-4. Programación de la función. Ejemplo práctico IV.	68

1 INTRODUCCIÓN

Con el paso del tiempo la tecnología ha ido avanzando y con ella la automatización de procesos industriales hasta tal punto que se requieren motores para prácticamente cualquier aplicación. En estos procesos se requiere de eficacia, precisión además de que estos dispositivos puedan ser controlados en velocidad, posición y par.

Para cumplir estos requisitos se puede aplicar distintos dispositivos:

- **Motor controlado por variador de velocidad.** Un variador de velocidad es un equipo empleado para controlar la velocidad y par de un motor variando la frecuencia y el voltaje de la entrada del mismo, por lo que no son empleados para control en posición.



Figura 1-1. Variador de frecuencia.

- **Motor paso a paso:** Los motores paso a paso se ayudan de un driver que permite convertir una señal pulsante en un avance de determinados grados del motor. Este avance también es configurable hasta cierto punto desde el driver, que también maneja la dirección de giro del motor. Son empleados en su mayoría para aplicaciones de control de posición.

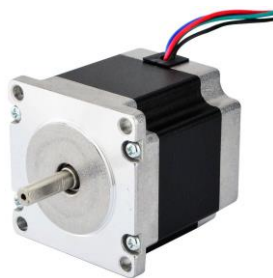


Figura 1-2. Motor paso a paso.

En cuanto a su funcionamiento, el motor paso a paso consta de un rotor y un estator con varios devanados los cuales se excitan para conseguir la rotación del rotor como se aprecia en la siguiente imagen:

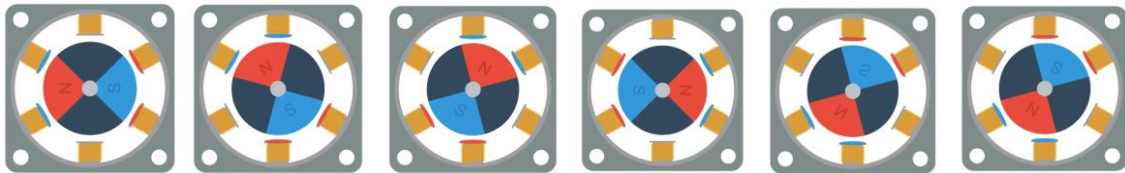


Figura 1-3. Funcionamiento motor paso a paso.

- **Servomotor:** Mientras que los dos grupos de motores anteriores trabajaban a priori con un bucle abierto sin realimentación, el servomotor se caracteriza por este mismo factor ya que lleva integrado un encoder, un dispositivo que convierte el movimiento del servomotor en una señal eléctrica que puede ser leída por un sistema de control, pudiendo ser utilizados para determinar la posición y velocidad del mismo. En la siguiente figura se pueden apreciar las partes de un servomotor:

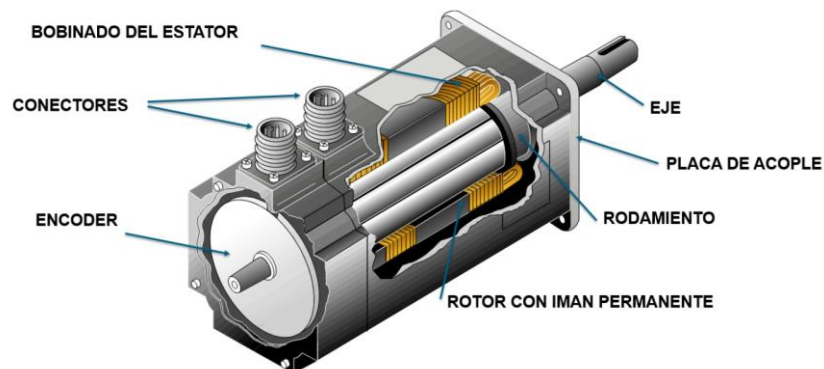


Figura 1-4. Servomotor.

A continuación se hará una breve comparativa entre los diferentes tipos de control de un motor:

En cuanto a los variadores de frecuencia, estos se emplean en operaciones en las que se precisa una determinada rotación por minuto como pueden ser en ruedas, ventiladores, bombas, etc. Por otro lado, la elección entre un motor paso a paso y un servomotor es algo más compleja. Los motores paso a paso son más baratos que los servomotores ya que no incluyen realimentación de posición lo que hace que sean menos precisos que los servomotores. Sin embargo, los motores paso a paso se emplean en operaciones que requieren bajas velocidades ya que es la zona de trabajo en la que presentan un mayor par, reservándose a velocidades altas el uso de servomotores. Estos procurarán siempre y cuando puedan completar el movimiento deseado si la carga a la que están sometidos no es excesivamente alta ya que tienen la capacidad de generar un par elevado incluso a altas velocidades. Además, el movimiento de los motores paso a paso implica poca aceleración y un ruido elevado mientras que el servomotor se caracteriza por unos movimientos más suaves y con alta aceleración y poco ruido.

1.1 Objetivos del proyecto

Este proyecto está realizado con una finalidad didáctica. Está destinado a alumnos que aunque inicialmente no tengan conocimiento sobre este tipo de motores puedan leer este documento y saber a qué se enfrentan y saber cómo manipularlos.

Aquella persona que empiece desde cero a tratar los servomotores se encontrará con una documentación por parte del fabricante bastante extensa y que dificulta en gran medida el progreso. Esto afecta sobre todo a alumnos que simplemente quieren realizar una serie de prácticas simples con este tipo de equipos.

En el desarrollo de este proyecto, el alumno afianzará conceptos en Unity Pro, de Schneider Electric, en el que se desarrollarán librerías para manejar el servomotor, así como conocimientos en conexiones del servomotor, cuáles son los límites que ofrece el servoaccionamiento empleado en este proyecto, así como aprender el entorno de otro programa como es SoMove, también de Schneider Electric, para la puesta en marcha.

1.2 Procedimiento

En primer lugar, se dará a conocer al alumno qué componentes son necesarios para afrontar este proyecto así como los modelos de cada equipo requerido. Se hará hincapié en un principio al apartado físico, cómo son el controlador Lexium y el servomotor, su conexionado y cómo manejar el controlador desde su panel HMI, es decir, sin ninguna aplicación externa, terminando por explicar los modos de funcionamiento que ofrece el modelo Lexium 32M así como los parámetros que entran en juego en cada uno de dichos modos.

Una vez se ha afianzado estos conceptos, se pasará al manejo de SoMove, con el que se modificarán los valores de parámetros comentados anteriormente sin necesidad de usar la interfaz HMI, se ejecutarán movimientos del servomotor, y se realizará un barrido de todas las funcionalidades que ofrece este software.

Por último, se recordará y aprenderá algunas funcionalidades de Unity Pro a nivel básico y se introducirán los ejercicios prácticos. En estos, se aportará al alumno un funcionamiento determinado del servomotor y será este el que deba aprender a modificar parámetros, configurar la conexión entre dispositivos, etc. Al final de cada apartado se describe una solución detallada de cómo conseguir el objetivo fijado, además del código de las funciones empleadas en los ejercicios.

2 SERVOACCIONAMIENTO CON LEXIUM 32M

Este capítulo está destinado a la descripción del servoaccionamiento con el Lexium 32M presente en este proyecto así como del servomotor. Se estudiará su conexionado, características y prestaciones de una forma general, especificando en un capítulo posterior el uso de un programa, SoMove, que se emplea para su configuración.

2.1 Servo accionamiento

De forma general, se pueden distinguir dos partes bien diferenciadas que conforman el servoaccionamiento, que son el servomotor y el servo drive.

2.1.1 Servomotor

Se define de una forma simple que un servomotor es un motor eléctrico, existiendo tanto de corriente continua como de alterna, que permite ser controlado para una satisfacer una determinada aplicación. Dependiendo de a qué aplicación se someta, el control se puede realizar en posición, en velocidad o en par con la ventaja de que son sistemas muy precisos. En este proyecto se usa el modelo BSH0551T01A2A de *Schneider Electric* [1]



Figura 2-1. Servomotor BSH0551T01A2A.

A continuación se presenta una tabla con los datos técnicos generales:

Característica	Valor
Número pares de polos	3
Par nominal, M_N	0,5 Nm
Par de parada continua, M_o	0,5 Nm
Corriente nominal, I_N	1,1 A _{rms}
Corriente de parada continua, I_o	1,4 Arms
Corriente máxima, $I_{m\acute{a}x}$	5,4 Arms
Velocidad nominal, n_N	8000 rpm
Velocidad máxima, $n_{m\acute{a}x}$	9000 rpm
Potencia nominal, P_N	0,4 kW

Tabla 2-1. Características del servomotor BSH0551T01A2A.

Este servomotor cuenta además con un encoder que mide la posición y velocidad del eje del servomotor. En concreto, es un encoder SKS35 Singleturn, midiendo un valor absoluto en el transcurso de una revolución que una vez completada, continúa contando a partir de este valor de forma incremental.

2.1.2 Servo drive

El servo drive es un dispositivo electrónico que controla la corriente de alimentación de los bobinados que integran el servomotor en función del tipo de control que se esté aplicando en ese momento y que, como se dijo anteriormente, podía ser control en posición, velocidad o par. Para este proyecto se usa el modelo **LMX32MU90M2** de *Schneider Electric* [2].



Figura 2-2. Lexium 32M. Placa de características.

A continuación se presentan una tabla con los diferentes canales que presenta el servo drive Lexium 32M:

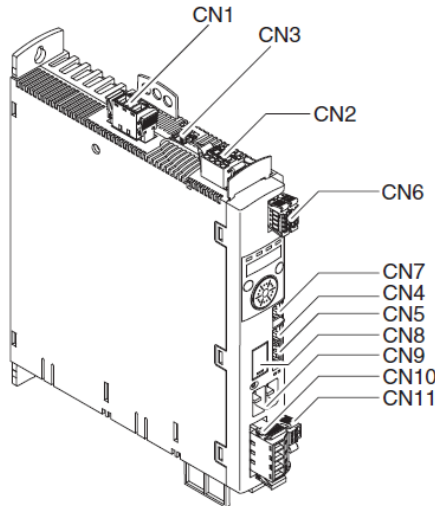


Figura 2-3. Canales del servo drive Lexium 32M.

Canal	Definición
CN1	Conexión de red (alimentación de la etapa de potencia)
CN2	Conexión: <ul style="list-style-type: none"> • Alimentación de control, 24V • Función de seguridad STO
CN3	Conexión para el encoder del motor
CN4	CAN in
CN5	CAN out
CN6	Entradas y salidas <ul style="list-style-type: none"> • 4 entradas digitales configurables • 2 salidas digitales configurables
CN7	Modbus (interfaz puesta en marcha)
CN8	Conexión para resistencia de frenado externa
CN9	Conexión para unión de bus DC
CN10	Conexión para fases del motor
CN11	Conexión para freno de parada del motor

Tabla 2-2. Canales del servo drive Lexium 32M.

En la tabla 2-2 se indica que las entradas y salidas digitales del LXM32M son configurables y para ello se usará el software SoMove de *Schneider Electric*. Las configuraciones de estas dependerán de las prestaciones que se requieran en la aplicación para la que se va a hacer uso, por lo que su explicación se dejará para los apartados posteriores.

2.1.3 Conexionado

En la siguiente figura se muestra de forma general el conexionado entre el servomotor y el servo drive:

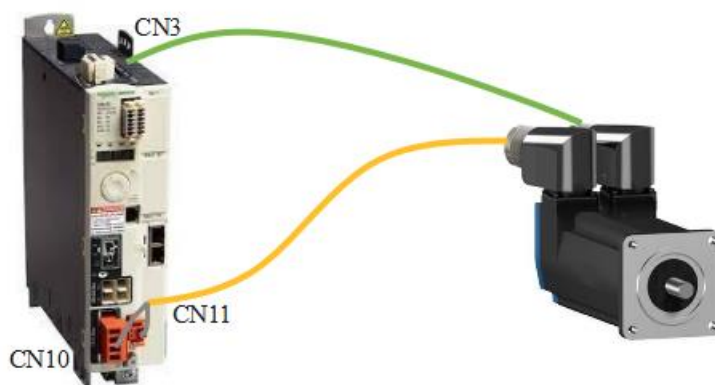


Figura 2-4. Conexionado general entre servomotor y LXM32M.

El servomotor se conecta a tres puertos del servo drive. Por un lado, se puede distinguir la conexión de potencia para la alimentación del motor, representándose en la figura 2-4 con una conexión naranja que se conecta los canales 10 y 11. En concreto, el canal 10 es una salida trifásica con sus tres fases más tierra, U-V-W-PE mientras que el canal 11 se emplea para la conexión del freno de parada¹. La otra conexión, de color verde según la figura 2-4 y conectada al canal CN3, sirve para recibir la señal del encoder por parte del motor.

¹ El freno de parada en un motor tiene la función de mantener la posición del motor con la etapa de potencia desactivada. No es una función de seguridad ni un freno de servicio.

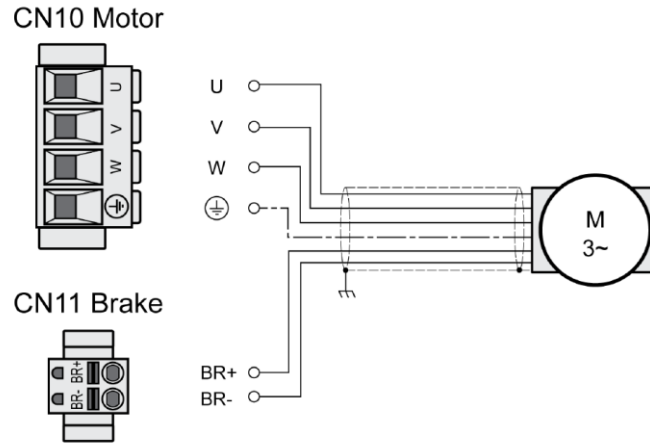


Figura 2-5. Conexiones CN10 y CN11.

En ambas conexiones del motor se usan conectores M23 de 8 polos para la fase de potencia (CN10 y CN11) y de 12 polos para la conexión del canal CN3. Desde el lado del Lexium 32M se cuenta con la presencia de conectores desnudos para conectar las fases de alimentación del motor mientras que para la lectura del encoder del canal CN3 se usa una conexión RJ45.



Figura 2-6. Conectores M23.

2.2 Primeros pasos con Lexium 32M

2.2.1 Interfaz HMI

El servo drive LXM32M ofrece la posibilidad de realizar ciertas operaciones sobre el servomotor mediante una interfaz HMI integrada, además de modificar parámetros y mostrar códigos de error.

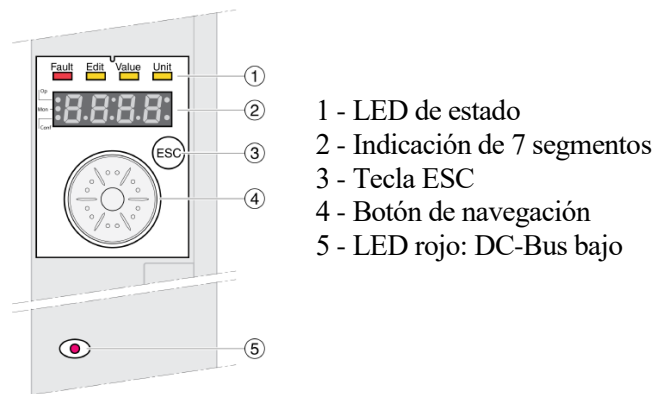


Figura 2-7. Interfaz HMI del LXM32M.

Es mediante el LED de estado y el display de 7 segmentos con 4 dígitos con los que se muestran los estados del equipo, parámetros y códigos de error. Con el botón de navegación pueden seleccionarse diferentes menús y modificar los parámetros de funcionamiento. La tecla ESC sirve para salir de pantallas de modificación de parámetros y otros menús.

A continuación se muestra una figura centrada en display y se indicara cada uno de los elementos y sus estados posibles:

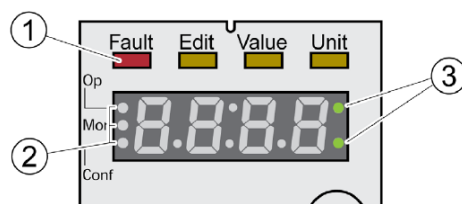


Figura 2-8. Display del LXM32M.

Usando como referencia la figura 2-8, se encuentra [2]:

1 - **Cuatro LEDs de estado.** Se estructura según la siguiente tabla:

Fault	Edit	Value	Unit	Significado
Iluminado en rojo				Estado de funcionamiento Fault
	Iluminado en amarillo	Iluminado en amarillo		El valor del parámetro puede editarse
		Iluminado en amarillo		Valor del parámetro
			Iluminado en amarillo	Unidad del parámetro seleccionado

Tabla 2-3. LEDs de estados del LXM32M.

2 - **Tres LEDs de nivel.** Se emplea para identificar los niveles de menú según la siguiente tabla[]

LED	Significado
Op	Funcionamiento
Mon	Informaciones de estado
Conf	Configuración

Tabla 2-4. LEDs de nivel del LXM32M.

3 - **Puntos parpadeantes.** Avisan de un error de clase 0.

2.2.2. Diagrama de estados

Una vez descritos los elementos físicos del servo drive y cómo este se maneja mediante el botón de navegación se comentarán los estados de funcionamiento del servo drive. Para alcanzar los modos de funcionamiento y poner en marcha el motor es necesario pasar previamente por unos estados de funcionamiento. Estos se reflejan en la siguiente figura:

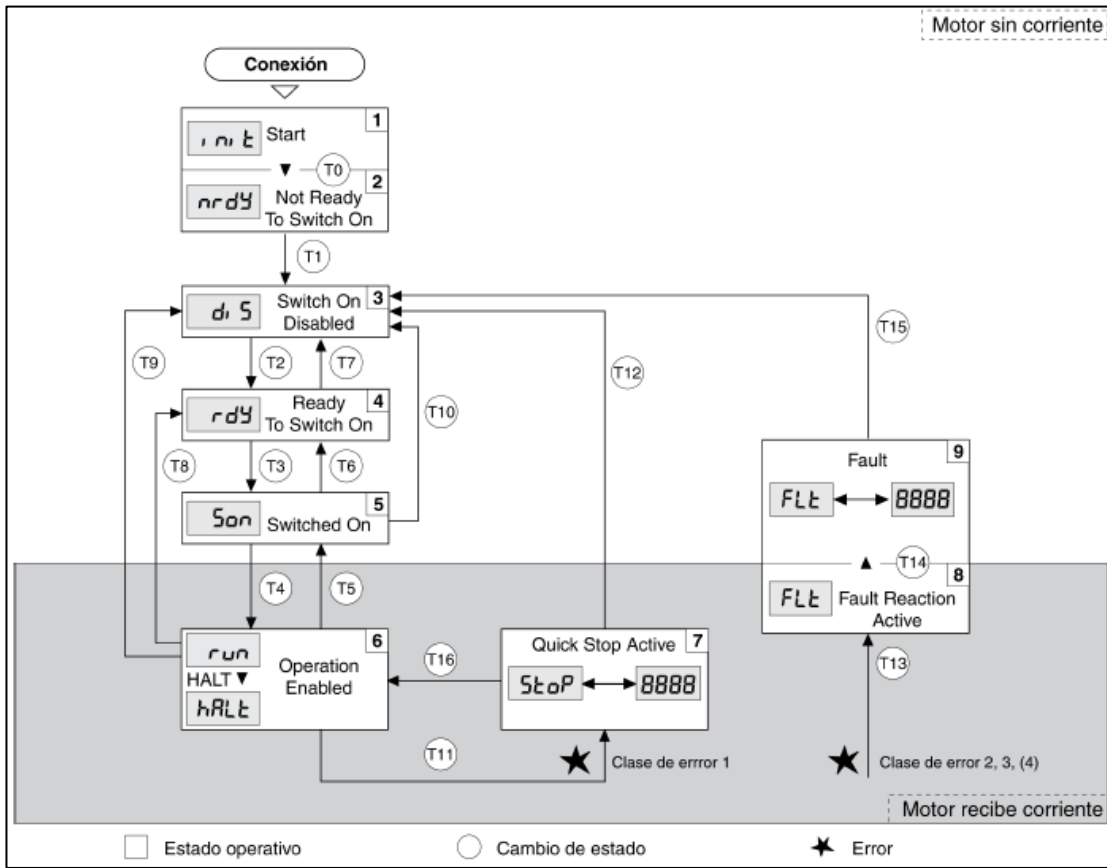


Figura 2-9. Diagrama de estados de funcionamiento.

Los estados de funcionamiento son:

Estado operacional	Designación
1 Start	Se inicia la electrónica
2 Not Ready to Switch On	La etapa de potencia no está lista para la conexión
3 Switch On Disabled	No se puede activar la etapa de potencia
4 Ready To Switch On	La etapa de potencia está lista para la conexión
5 Switched On	Se conecta la etapa de potencia
6 Operation Enabled	Se conecta la etapa de potencia. El modo de funcionamiento ajustado está activo
7 Quick Stp Active	“Quick Stop” se está ejecutando
8 Fault Reaction Active	Se ejecuta la reacción de error
9 Fault	Reacción de error finalizada. Se desactiva la etapa de potencia

Tabla 2-5. Estados de funcionamiento.

2.2.3. Errores de funcionamiento

A su vez, los errores que se producen se dividen en diferentes clases dependiendo de la importancia de dicho error. Para ello, en la siguiente table se especifican las clases, la consecuencia de que este se produzca y como eliminarlo[2]:

Clase de error	Transición de estado	Error response	Reinicio de un mensaje de error
0	-	No se interrumpe el movimiento	Función "Fault Reset"
1	T11	Detener el movimiento con "Quick Stop"	Función "Fault Reset"
2	T13, T14	Detener el movimiento con "Quick Stop" y desactivar la etapa de potencia durante la parada del motor	Función "Fault Reset"
3	T13, T14	Desactivar de inmediato la etapa de potencia sin detener antes el movimiento	Función "Fault Reset"
4	T13, T14	Desactivar de inmediato la etapa de potencia sin detener antes el movimiento	Desconexión y reconexión

Tabla 2-6. Clases de error.

2.2.4. Modos de funcionamiento.

Los modos de funcionamientos son accesibles siempre y cuando se haya realizado una buena transición entre los estados de la figura 2-9. Por ello, establecer un contexto, esta fase empezaría cuando el motor está energizado y el display de la interfaz HMI marque "Run".

El servo drive tiene 8 modos de funcionamiento[2]:

- Modo Jog
- Modo Electronic Gear
- Modo Profile Torque
- Modo Profile Velocity
- Modo Profile Position
- Modo Interpolated Position
- Modo Homing
- Modo Motion Sequence

2.2.4.1. Modo Jog

El modo de funcionamiento **Jog** es un modo de movimiento manual en el que se efectúa un movimiento en la dirección establecida. Este modo se puede llevar a cabo mediante 2 métodos diferentes:

- **Movimiento continuo:** Si se activa la señal de movimiento de motor en cualquier sentido de giro, se efectúa un movimiento en dicha dirección.

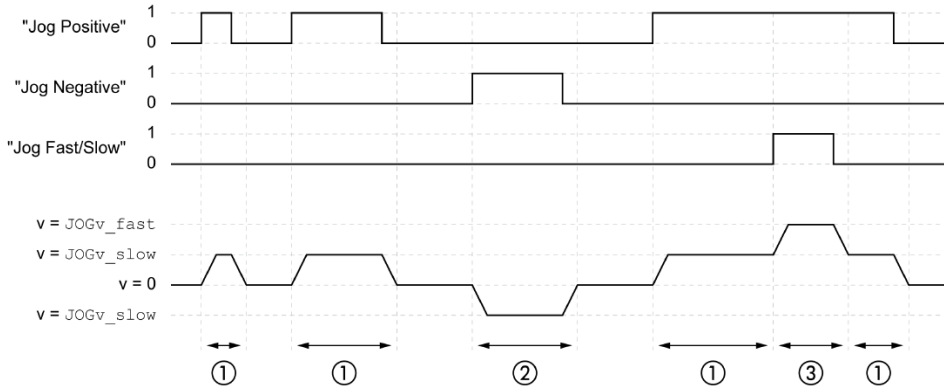


Figura 2-10. Modo Jog. Movimiento continuo.

La numeración de la figura 2-10 hace referencia:

1. Movimiento lento en dirección positiva
2. Movimiento lento en dirección negativa
3. Movimiento rápido en dirección positiva

Las señales “Jog Positive”, “Jog Negative” y “Jog Fast/Slow” se activan mediante los parámetros *JOGactivate_Bit0*, *JOGactivate_Bit1* y *JOGactivate_Bit2*, respectivamente.

- **Movimiento paso a paso:** Si se activa la señal de movimiento de motor en cualquier sentido de giro, el motor gira una cantidad definida previamente por el usuario, mediante el parámetro *JOGstep*, y tras esto el motor un tiempo definido mediante *JOGtime*. En caso de que la señal hacia la misma dirección siga activada, el movimiento continua de forma indefinida hasta que se deje de recibir esta señal.

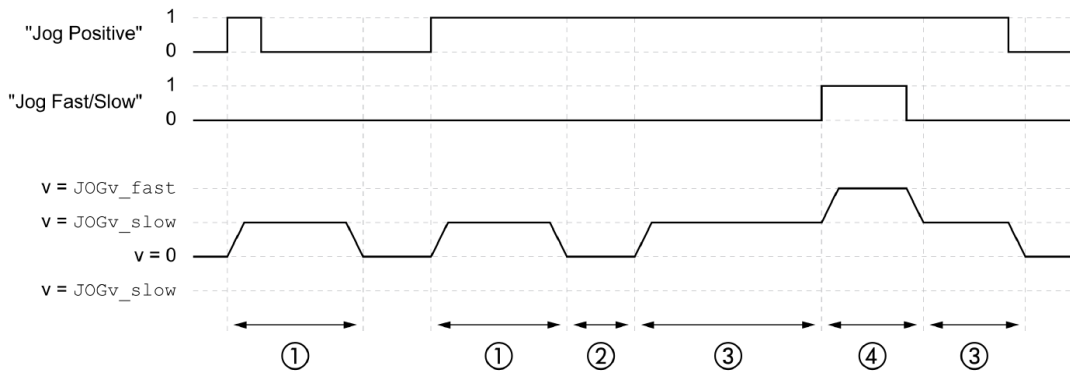


Figura 2-11. Modo Jog. Movimiento paso a paso.

La numeración de la figura 2-11 hace referencia:

1. Movimiento lento con una cantidad parametrizable, *JOGstep*, en dirección positiva
2. Tiempo de espera *JOGtime*
3. Movimiento lento continuo en dirección positiva
4. Movimiento rápido continuo en dirección positiva

A modo de resumen, se muestra en la siguiente figura los parámetros ajustables en este modo de funcionamiento. Los únicos parámetros que no han sido comentados son *RAMP_v_acc*, *RAMP_v_dec* y *RAMP_v_max*, que se encargan de definir el perfil de velocidad de movimiento.

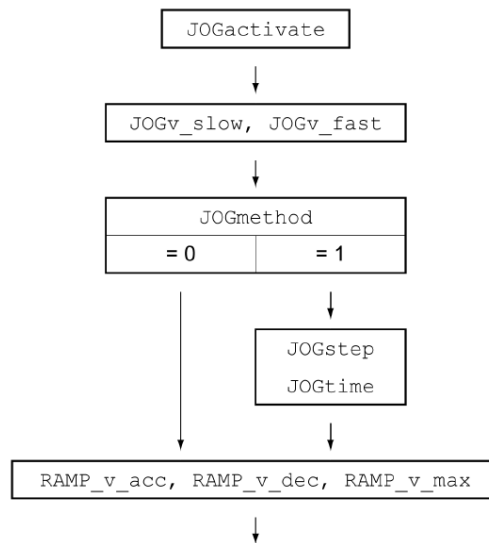


Figura 2-12. Parámetros modo Jog.

2.2.4.2. Electronic Gear

En el modo engranaje electrónico o **Electronic Gear**, se realiza un movimiento en función de señales, configurables y externas, que recibe el servo drive. Estas señales pueden ser del tipo A/B, señales P/D o señales CW/CCW y se seleccionan mediante el parámetro *PTI_signal_type*. El movimiento puede llevarse a cabo mediante 3 métodos diferentes, seleccionables mediante *GEARreference*:

- **Sincronización de posición sin movimiento de compensación:** El movimiento se realiza en función de las señales externas de tal forma que estas no se tengan en cuenta durante una interrupción, ya sea por una parada o por un error de clase 1.
- **Sincronización de posición con movimiento de compensación:** El movimiento se realiza en función de las señales externas y a diferencia del método anterior, se tienen en cuenta las señales externas y se compensan. Si se selecciona este modo, con el parámetro *GEARposChgMode* se ajusta cómo deben tratarse las modificaciones en la posición del motor y en las señales piloto con la etapa de potencia desactivada.
- **Sincronización de velocidad:** Se realiza un movimiento de velocidad síncrona con respecto a las señales externas.

En los métodos “Sincronización de posición sin movimiento de compensación” y “Sincronización de posición con movimiento de compensación” se puede realizar un movimiento offset con el que se puede realizar un movimiento de una cantidad parametrizable de posiciones de motor mediante la activación de una señal independiente de las externas. La posición offset se ajusta mediante *OFSp_RelPos1* y *OFSp_RelPos2*.

Es necesario configurar también el parámetro *GearRatio*, que viene a ser una relación de transmisión de tal forma que a cada pulso de señal externa que reciba el servo drive el motor lo traduce en un número de posiciones de motor.

$$GearRatio = \frac{\text{Incrementos de motor}}{\text{Incrementos de la referencia}} = \frac{\text{Num. del factor de engranaje}}{\text{Denom. del factor de engranaje}}$$

Para dar valor a este parámetro se necesitan otros como son *GEARselect*, *GEARnum*, *GEARdenom*, *GEARnum2* y *GEARdenom2*.

A modo de resumen, se muestra en el siguiente gráfico los parámetros ajustables en este modo de funcionamiento.

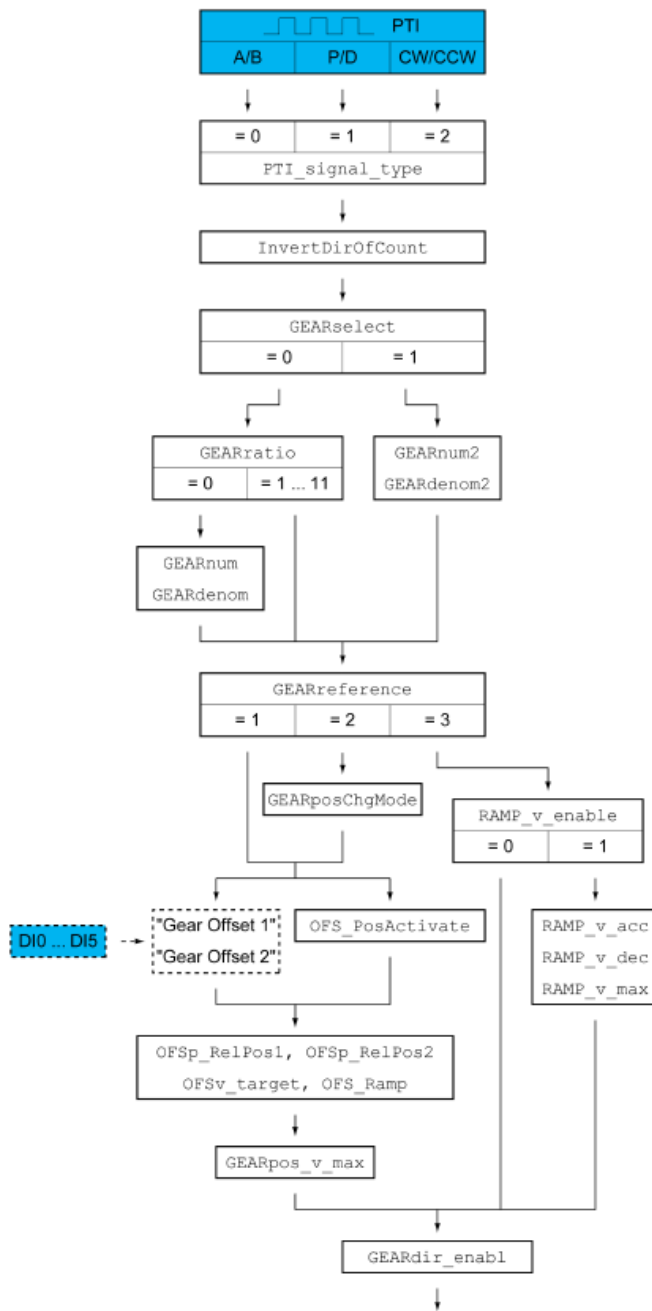


Figura 2-13. Parámetros modo Electronic Gear.

Los únicos parámetros que no han sido comentados son $RAMP_v_acc$, $RAMP_v_dec$ y $RAMP_v_max$, que se encargan de definir el perfil de velocidad de movimiento, además de $GEARpos_v_max$, que limita la velocidad en el modo sincronización de posición y $GEARdir_enabl$, que limita el movimiento a una única dirección en caso de estar activada.

2.2.4.3. Modo Profile Torque

En el modo de funcionamiento **Profile Torque** se realiza un movimiento con la finalidad de alcanzar una determinada referencia de par fijado bien mediante entradas analógicas, a través de parámetros ($PTiq_target$), o a través del puerto PTI. Esta selección se lleva a cabo dependiendo del valor seleccionado en $PTiq_reference$.

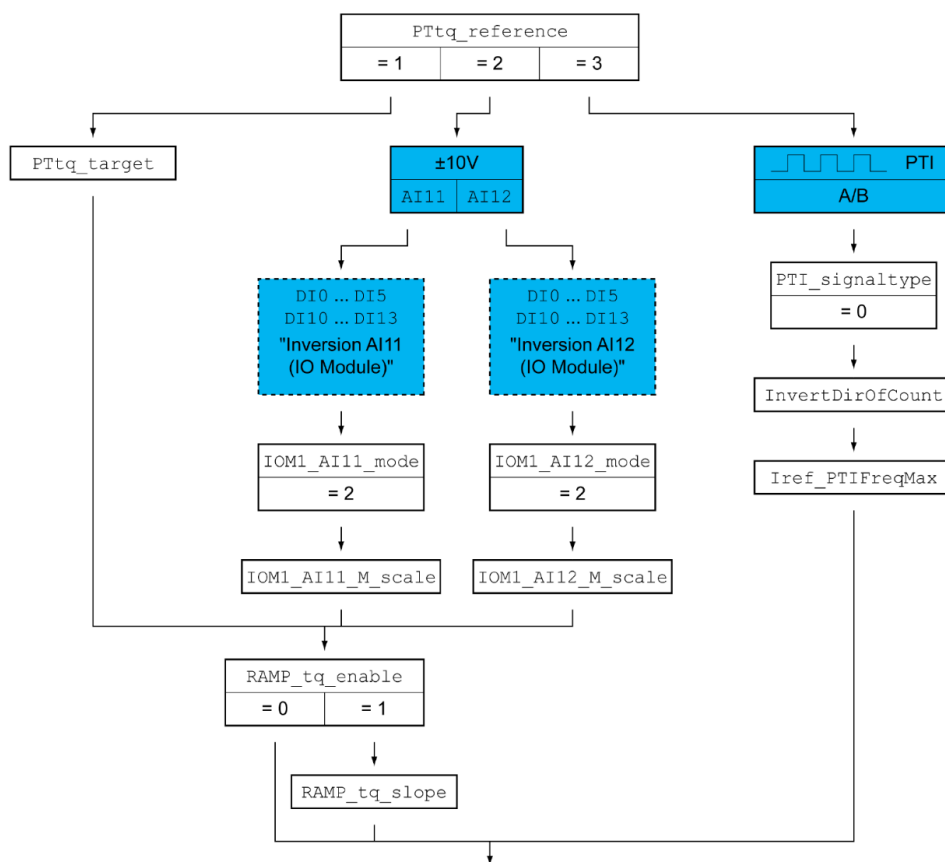


Figura 2-14. Parámetros Torque Profile.

Comentar otros parámetros como son:

- *IOM1_AIIX_mode*: Parámetros de ajuste del modo de utilización de las entradas de señales digitales.
- *IOM1_AIIX_scale*: Parámetros de ajuste del par de destino para un valor de tensión de 10 V.
- *RAMP_tq_enable* y *RAMP_tq_slope*: Parámetros de ajuste del perfil que permiten la activación del perfil de movimiento y definir la pendiente del perfil de movimiento, respectivamente.
- *I_ref_PTIFreqMax*: Parámetro que define la corriente de referencia a través de la interfaz PTI.

2.2.4.4. Modo Profile Velocity

En el modo de funcionamiento **Profile Velocity** se realiza un movimiento con la finalidad de alcanzar una determinada referencia de velocidad fijado bien mediante entradas analógicas o bien mediante el parámetro *PVv_target*.

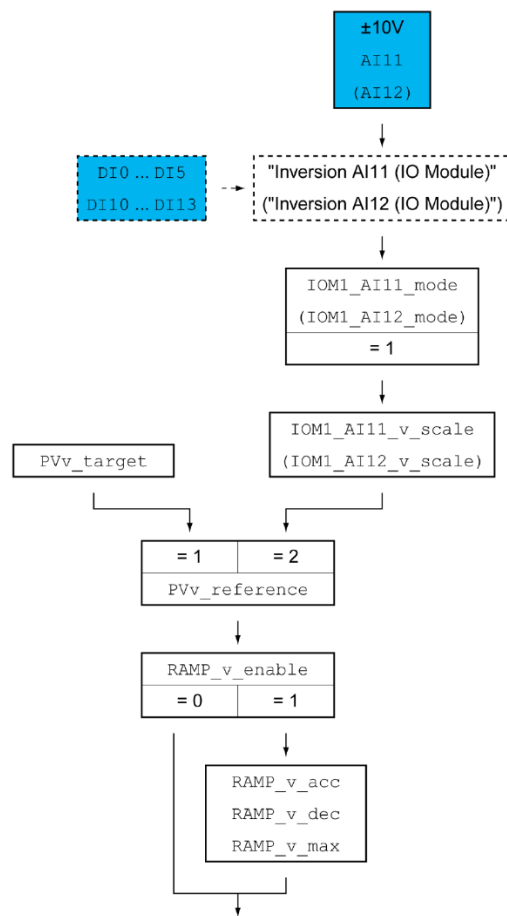


Figura 2-15. Parámetros Profile Velocity.

Comentar otros parámetros como son:

- *IOM1_AIIX_mode*: Parámetros de ajuste del modo de utilización de las entradas de señales digitales.
- *IOM1_AIIX_scale*: Parámetros de ajuste del par de destino para un valor de tensión de 10 V.
- *RAMP_v_acc*, *RAMP_v_dec* y *RAMP_v_max*: Parámetros de ajuste del perfil de movimiento.

2.2.4.5. Modo Profile Position

En el modo de funcionamiento **Profile Position** se realiza un movimiento a posición destino definida por *PPp_target* llegando a una velocidad destino, definida por *PPv_target*. El movimiento se puede llevar a cabo mediante dos métodos diferentes:

- **Movimiento relativo**: Se emplea para realizar un movimiento hacia una posición se toma como referencia la posición actual del motor.
- **Movimiento absoluto**: Se emplea para realizar un movimiento hacia una posición se toma como referencia el punto cero, por lo que este debe estar definido previamente mediante el modo de funcionamiento homing.

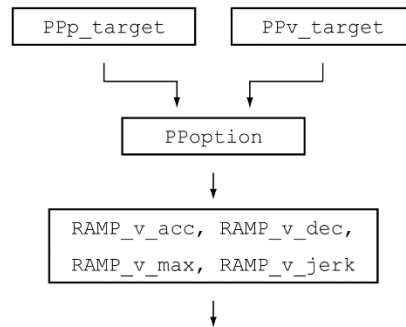


Figura 2-16. Parámetros Profile Position.

Comentar otros parámetros como son:

- *PPoption*: Parámetro de ajuste del método a aplicar (relativo o absoluto).
- *IOMI_AIX_scale*: Parámetros de ajuste del par de destino para un valor de tensión de 10 V.
- *RAMP_v_acc*, *RAMP_v_dec*, *RAMP_v_max* y *RAMP_v_jerk*: Parámetros de ajuste del perfil de movimiento.

2.2.4.6. Modo Interpolated Position

El modo de funcionamiento **Interpolated Position** ejecuta un movimiento a posiciones de referencia preestablecidas cíclicamente. El movimiento hacia las posiciones de referencia se inicia con la señal SYNC, y se sigue una interpolación fina cada 250 μ s.

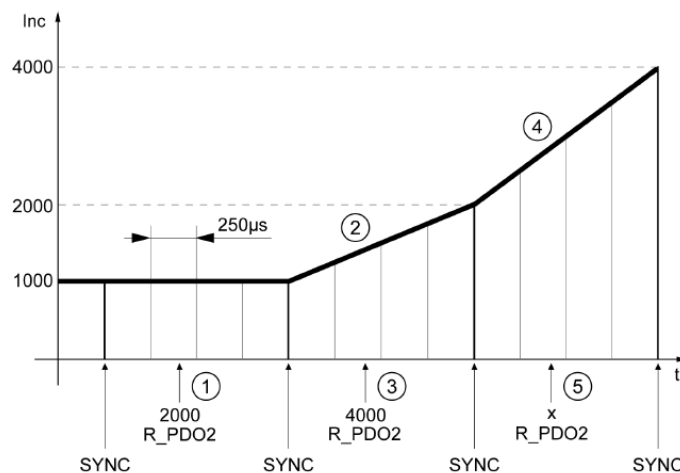


Figura 2-17. Interpolated Position.

La numeración de la figura 2-17 hace referencia:

1. Transferencia de la primera posición de referencia
2. Movimiento hasta la primera posición de referencia
3. Transferencia de la segunda posición de referencia
4. Movimiento hasta la segunda posición de referencia
5. Transferencia de la siguiente posición de referencia

Para este modo de funcionamiento se debe activar el mecanismo de sincronización a través del parámetro *SyncMechStart=2*.

2.2.4.7. Modo Homing

El modo de funcionamiento **Homing** permite a través de un movimiento crear referencias de posición, es decir, asignar a una determinada posición de motor (posición mecánica) un origen de movimiento o posición cero.

Se pueden aplicar diferentes métodos para obtener dicho origen:

1. **Movimiento de referencia a un final de carrera:** Este método consiste en realizar un movimiento hacia un final de carrera. Existen dos posibilidades, que el final de carrera se encuentre en un extremo u otro (final de carrera positivo o negativo) lo que definirá el sentido del movimiento para activarlo, en la figura inferior casos A y B. Cuando este se active, el motor se detendrá y realizará un movimiento en sentido contrario hasta que este deje de estar activado, tras lo cual, volverá a realizar un movimiento en el mismo sentido que el último realizado y de una determinada distancia definida por $HMdis$, tras el cual el motor para definitivamente.

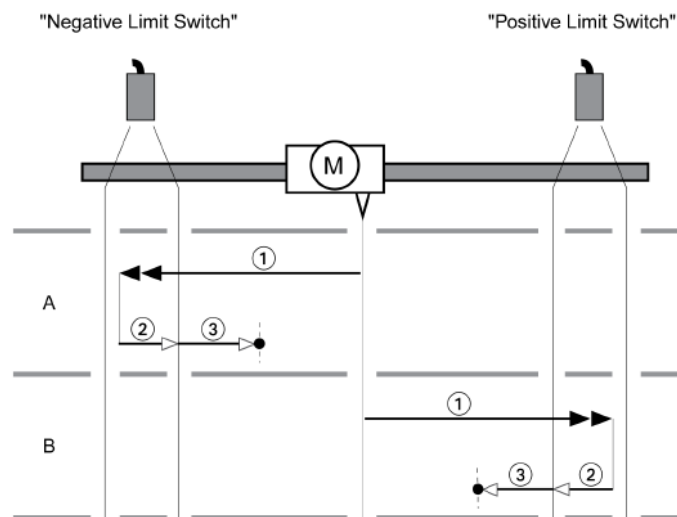


Figura 2-18. Homing con final de carrera.

1. Movimiento a un final de carrera con velocidad HMv
 2. Movimiento al punto de conmutación a velocidad HMv_{out}
 3. Movimiento a la distancia $HMdis$ a velocidad HMv_{out}
2. **Movimiento de referencia al interruptor de referencia:** Este método consiste en realizar un movimiento hacia un interruptor de referencia. Existen también dos posibilidades, como en el caso anterior, en función de la localización del interruptor de referencia, definiendo así los sentidos del movimiento. Cuando este se active, el motor se detendrá y realizará un movimiento hasta que este deje de estar activado, tras lo cual, volverá a realizar otro movimiento de una determinada distancia definida por $HMdis$, tras el cual el motor para definitivamente. En este caso, como el interruptor de referencia no es tan restrictivo como el final de carrera que hace la función de un límite mecánico, la dirección de movimiento de estos dos últimos movimientos depende del tipo (A, B, C y D) seleccionado.

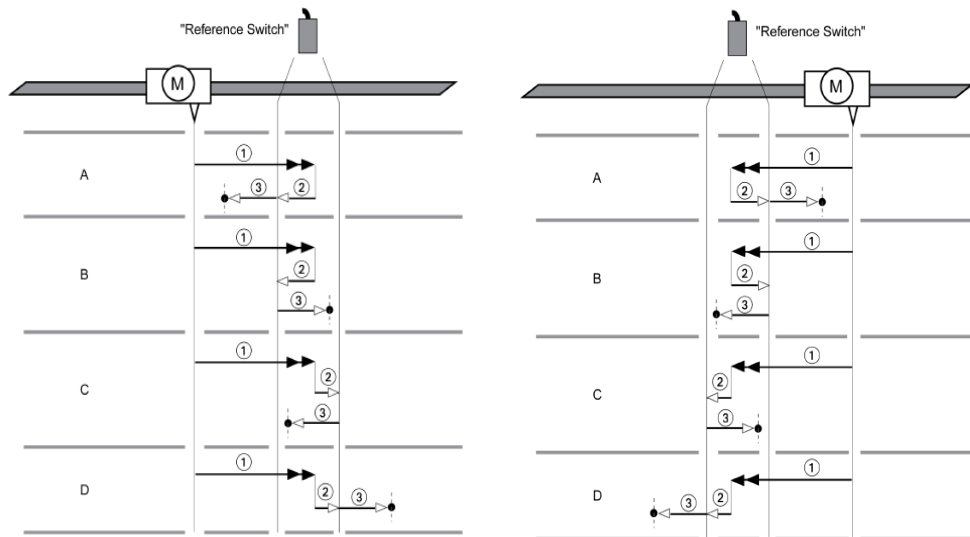


Figura 2-19. Homing con interruptor de referencia.

La numeración de la figura 2-19 hace referencia:

1. Movimiento a un final de carrera con velocidad HMv
 2. Movimiento al punto de conmutación a velocidad HMv_{out}
 3. Movimiento a la distancia $HMdis$ a velocidad HMv_{out}
3. **Movimiento de referencia en el pulso índice:** Este método consiste en realizar un movimiento al pulso índice desde la posición actual del motor hasta el siguiente punto índice del encoder. Una vez alcanzada, esta posición será la posición de referencia. El sentido de giro del motor dependerá del método seleccionado.

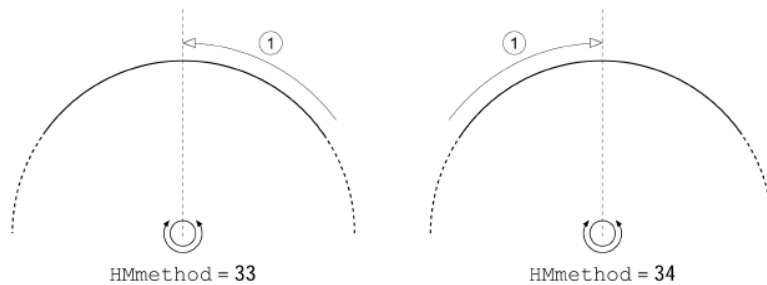


Figura 2-20. Homing con pulso índice.

4. **Establecimiento de medida:** Con este método, se sustituye la posición actual del motor por una posición establecido mediante el parámetro HMp_{setP} .

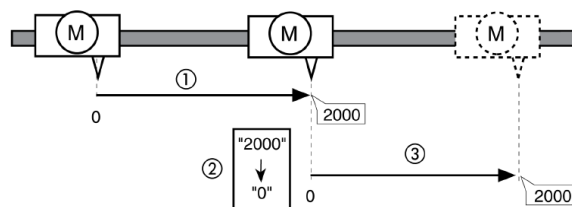


Figura 2-21. Homing con establecimiento de medida.

La numeración de la figura 2-21 hace referencia:

1. El motor se posiciona en la posición de motor 2000
2. Se activa el establecimiento de medida, con $HMp_{setP}=0$
3. Se realiza un nuevo movimiento a la posición de motor 2000

La siguiente figura recoge la configuración de parámetros del modo Homing:

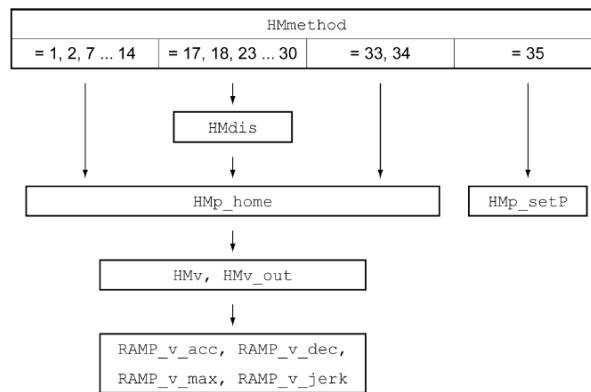


Figura 2-22. Parámetros y secuencia modo homing.

A modo de resumen, se ofrece la siguiente tabla para aclarar todo los tipos de homing de los que dispone el servo drive LXM32M seleccionado mediante el parámetro *HMmethod*:

Id.	Método de Homing
1	LIMN con pulso índice
2	LIMP con pulso índice
7	REF+ con pulso índice, inv., exterior
8	REF+ con pulso índice, inv., interior
9	REF+ con pulso índice, no inv., interior
10	REF+ con pulso índice, no inv., exterior
11	REF- con pulso índice, inv., exterior
12	REF- con pulso índice, inv., interior
13	REF- con pulso índice, no inv., interior
14	REF- con pulso índice, no inv., exterior
17	LIMN
18	LIMP
23	REF+ inv., exterior
24	REF+ inv., interior
25	REF+ no inv., interior
26	REF+ no inv., exterior
27	REF- inv., exterior
28	REF- inv., interior
29	REF- no inv., interior
30	REF- no inv., exterior
33	Pulso índice, dirección negativa
34	Pulso índice dirección positiva
35	Establecimiento de medida

Tabla auxiliar

- **REF+:** Movimiento de búsqueda en dirección positiva
- **REF-:** Movimiento de búsqueda en dirección negativa
- **Inv.:** Invertir la dirección del interruptor
- **No Inv.:** No invertir la dirección del interruptor
- **Exterior:** Distancia pulso índice fuera del interruptor
- **Interior:** Distancia pulso índice dentro del interruptor

Tabla 2-7. Métodos homing.

2.2.4.8. Modo Motion Sequence

El modo de funcionamiento **Motion Sequence** consiste en la realización de movimientos acorde a unos registros de datos parametrizables. En un registro de datos personalizable se encuentra la información respectiva al movimiento como pueden ser el tipo de movimiento, posición de destino, velocidad de destino, etc. Por otro lado, este registro también podría contener, ya que depende de la configuración, información respectiva a la secuenciación de movimiento, de tal manera que tras finalizar un movimiento, se inicie otro movimiento contenido en otro registro de datos, formando una secuencia.

En este modo de funcionamiento se encuentran varios tipos de registros de datos:

- Movimiento a un valor de posición determinado (movimiento absoluto, relativo o aditivo)
- Movimiento a velocidad determinada
- Homing
- Repetición de una secuencia determinada
- Movimiento Electronic Gear
- Modificar valores de parámetros.

3 AUTÓMATA PROGRAMABLE

En este apartado se describirán los primeros pasos en la iniciación del uso del software Unity Pro, de *Schneider Electric*. Se centrará en los aspectos requeridos en el proyecto como la definición del autómata del que se hace uso en el proyecto, en la creación de bloques funcionales, tipos de lenguajes disponibles, así como de los tipos de conexiones entre PC-PLC.

3.1 Configuraciones en Unity Pro

La configuración del autómata empleado en el proyecto se realiza mediante el software **Unity Pro**. Esta configuración inicialmente se refiere a la implementación del PLC real en el programa, por lo que deberá ser acorde al autómata físico o de lo contrario el programa impedirá trabajar con dicha configuración. A continuación se describirán los pasos para su elaboración teniendo en cuenta que se va a emplear un autómata M340.



Figura 3-1. Modicon M340.

Pasos a realizar en Unity Pro:

1. Una vez abierto el software Unity Pro, clicar en “Generar un nuevo proyecto”.
2. Seleccionar el PLC con el que se llevará a cabo el proyecto. En este caso se emplea el modelo M340 BXM P34 20102 y un bastidor de 4 slots. Se mostrará en las figuras siguientes el proceso de selección.

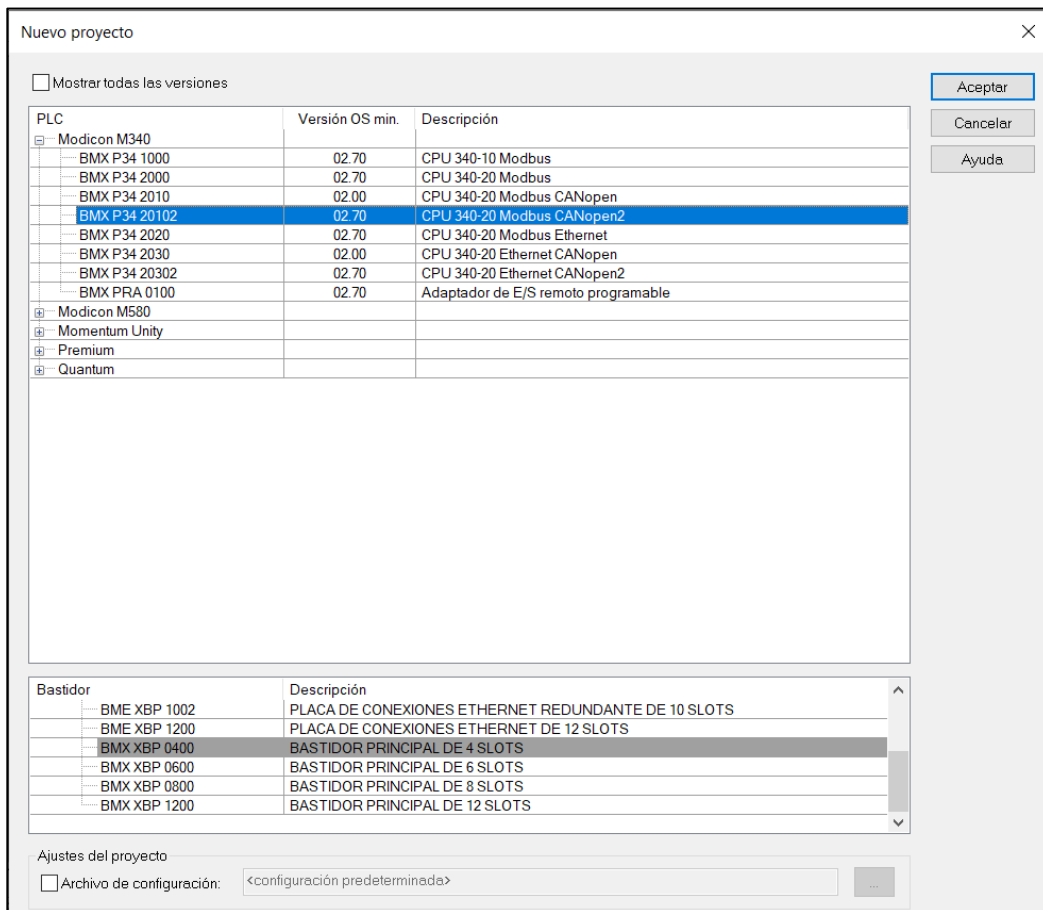


Figura 3-2. Selección del autómata programable.

Inicialmente se mantendrá esta configuración básica para que en capítulos posteriores donde se vean la implementación de librerías y será en esas donde se añadan los módulos necesarios.

3.2 Conexión y comunicación

La conexión entre PLC-PC se puede hacer mediante:

- **USB 2.0:** Con esta conexión se conectará el extremo USB al PC mientras que se conectará el extremo Mini-B al PLC.



Figura 3-3. Cable USB-Mini B.

Para configurar este modo de conexión en el software Unity Pro se realizarán los siguientes pasos:

1. Seleccionar “PLC” y luego “Establecer dirección...”

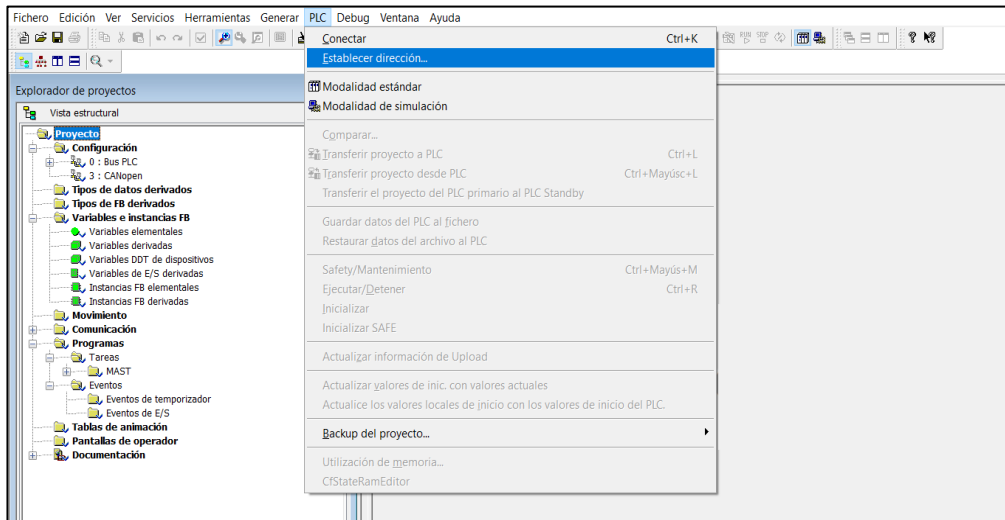


Figura 3-4. Desplegable PLC.

2. Seleccionar en el apartado “Dirección” la opción “SYS” y en el apartado “Medios” la opción USB. Por último, aceptar.

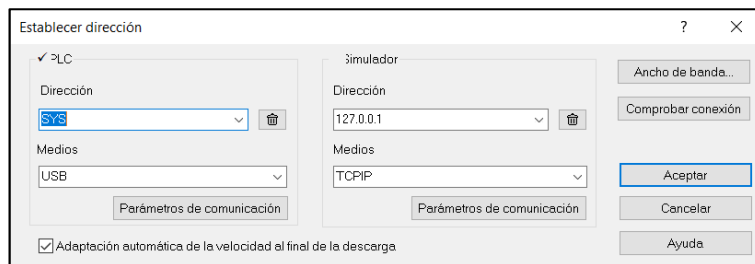


Figura 3-5. Establecer dirección mediante USB con el autómata.

- **Modelo TCP-IP:** Este modo de conexión se realiza mediante un cable Ethernet que conecta PLC-PC y depende de si la CPU empleada permite dicho modo de conexión. En nuestro caso, el modelo P34 20102 no presenta canal Ethernet, por lo que no será posible la conexión por este canal. Aun así, se explicará, haciendo uso de la tarjeta P34 2020, cómo se realizaría pues el proyecto se puede realizar con otro modelo de autómata y la mayoría de estos sí que contienen canal ethernet. Para ello es necesario seguir los siguientes pasos:
 1. Seleccionar “PLC” y luego “Establecer dirección...” al igual que en la figura 3-1.
 2. Establecer en el apartado “Dirección” la IP del PLC y en el apartado “Medios” el método “TCP/IP”. En este caso es necesario que PLC y PC se encuentren en la misma subred para que se pueda establecer la conexión por lo que es posible que se tenga que modificar la configuración ethernet del ordenador.



Figura 3-6. Establecer dirección mediante TCP-IP con el autómata.

3. Por último, hay que configurar el canal Ethernet del M340. Para ello, primero dirigirse a la ventana “Bus PLC” desde la carpeta lateral “Configuración” y se hace doble clic sobre el canal ethernet de la CPU, como se muestra en la siguiente imagen:

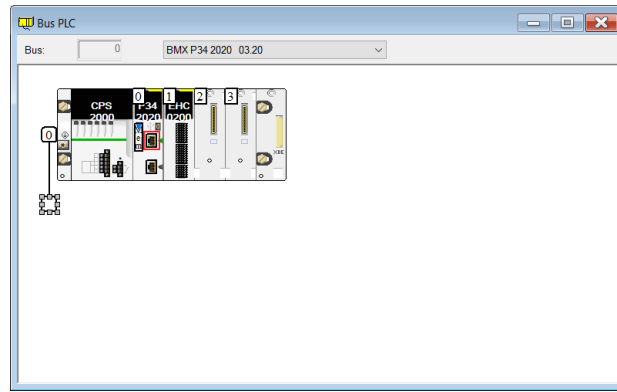


Figura 3-7. Ventana BUS PLC.

Al pulsar sobre este se abrirá una ventana de configuración del canal ethernet y se realizará la siguiente modificación:

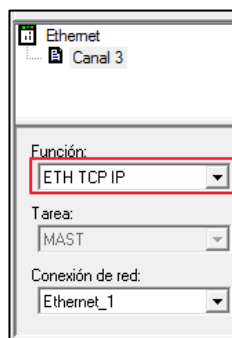


Figura 3-8. Ventana de configuración del canal Ethernet.

Luego, abrir los ajustes ethernet desde la carpeta Comunicación\Redes\Ethernet_1, que desplegará la ventana de la figura 3-7 donde se debe establecer la dirección IP del PLC.

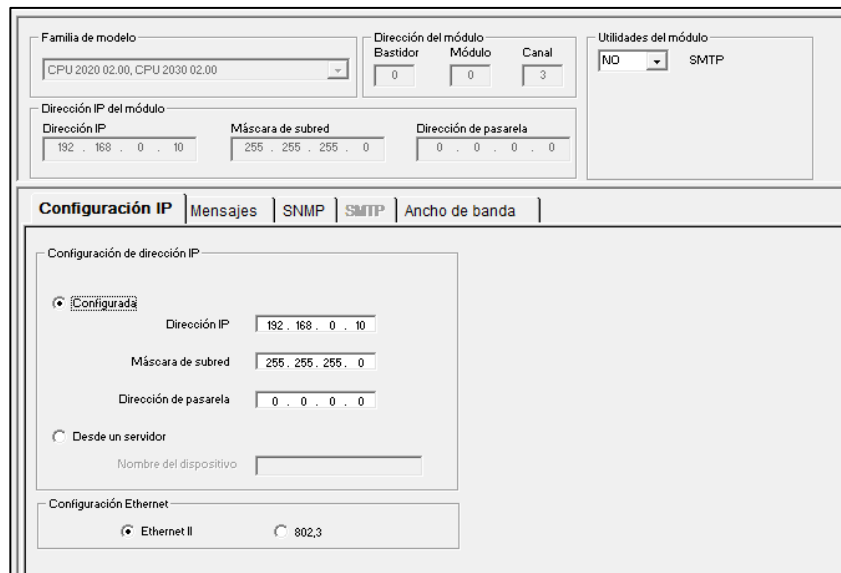


Figura 3-9. Ventana de configuración de la IP del PLC.

3.3 Unidades de programación

En este apartado se presenta como se crean las unidades de programación que se emplean en este proyecto y como ejecutarlos [3].

El código de este proyecto se ha desarrollado sobre unidades de programación POU, o en este caso para este software, unidades DFB. Estos bloques se crean declarando en las entradas, salidas y variables internas que se empleará en el bloque DFB. Para crear uno y declarar estas variables se realizarán los siguientes pasos:

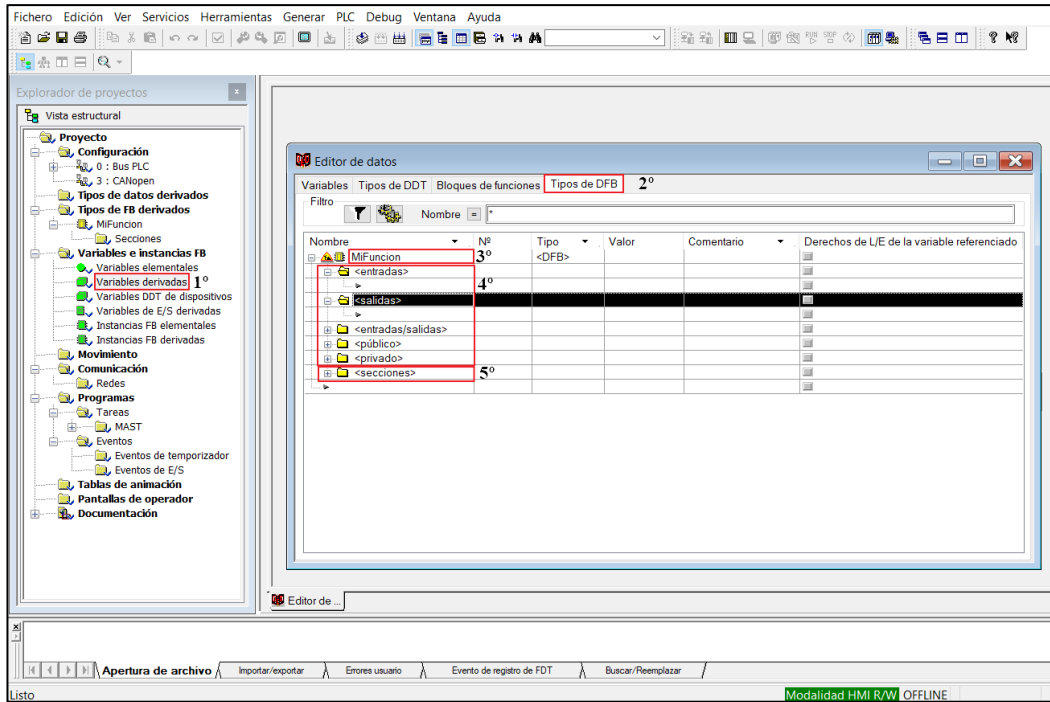


Figura 3-10. Creación de bloques DFB.

- 1- Haciendo doble clic derecho en el recuadro número 1 se desplegará la ventana “Editor de datos” que se encuentra a la derecha de la imagen.
 1. Se selecciona la sección “Tipos de DFB” en la que se declaran los bloques funcionales.
 2. Se aporta un nombre al bloque.
 3. Una vez definido el nombre, el programa permitirá definir las variables de entrada, salida y entrada/salida del bloque, así como variables internas.
 4. En el apartado “Secciones” se creará el código del bloque. Si se añade una sección se debe especificar en que lenguaje se desarrollará, pudiendo distinguir:
 - a. **ST**: El lenguaje de programación ST o texto estructurado permite crear código mediante un editor de texto.

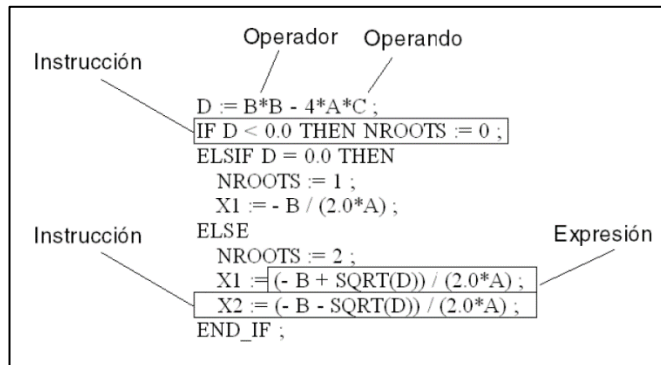


Figura 3-11. Lenguaje de programación ST.

- b. **LD:** El editor LD permite la programación gráfica de diagramas empleando bobinas y contactores además de otros bloques de funciones.

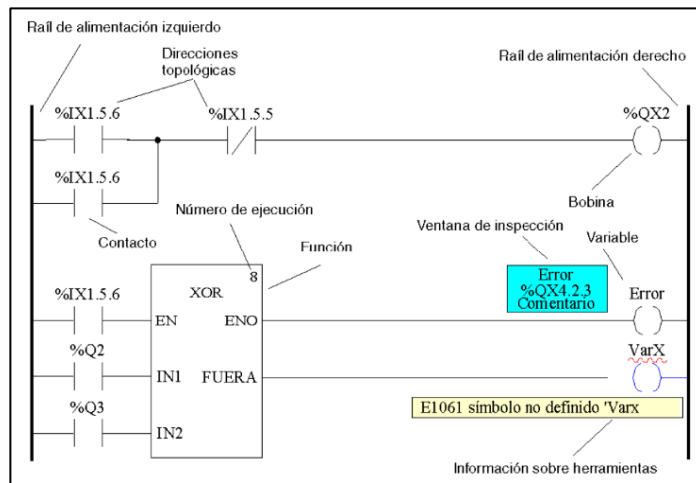


Figura 3-12. Lenguaje de programación LD.

- c. **FBD:** El editor FBD permite la programación gráfica de bloques funcionales.

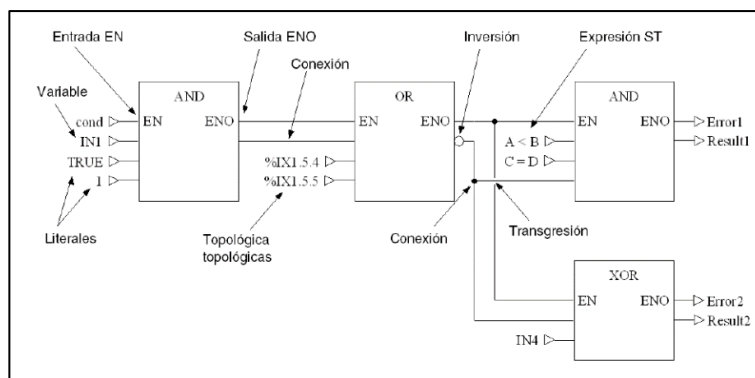


Figura 3-13. Lenguaje de programación FBD.

- d. **IL:** El lenguaje IL permite ejecutar una lista de instrucciones, de ahí su nombre:

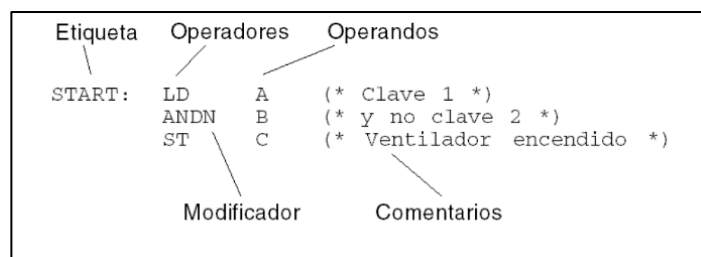


Figura 3-14. Lenguaje de programación IL.

Una vez definido el lenguaje, aparecerá una hoja en blanco para la programación de dicho bloque. Cuando se tiene terminado el bloque funcional, solo faltaría ejecutarlo desde otra unidad de programa. En este caso, como modo de ejemplo, se va a realizar mediante LD.

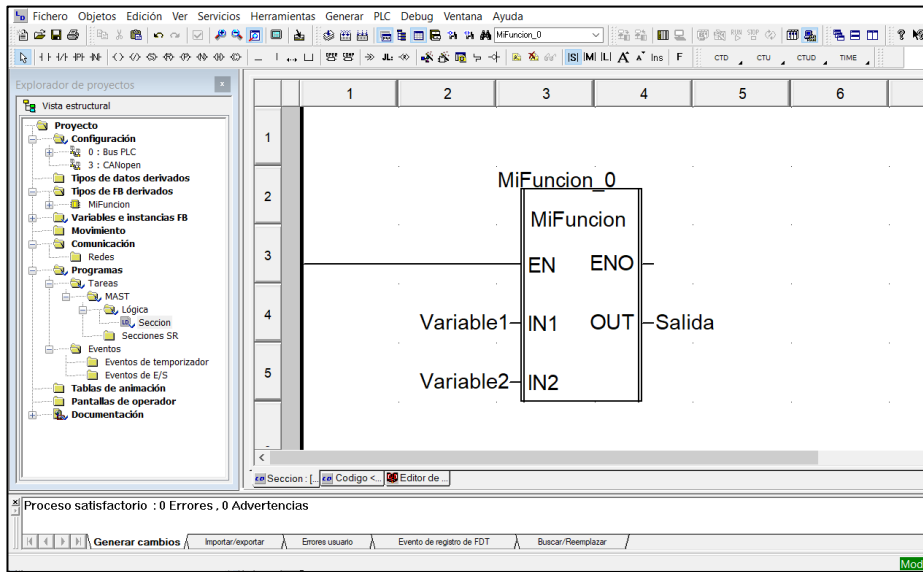


Figura 3-15. Llamada a función DFB.

Solo se ejecutará el código que se añade en las sección “MAST”, por lo que si se realiza un bloque y este no es llamado en esta sección, su código no se ejecutará. Si se desea consultar algunos detalles del lenguaje u otros aspectos de este software puede consultar la referencia [3].

3.4 Modos de ejecución

Unity pro tiene la ventaja de que permite probar el funcionamiento de cualquier programa en modo de simulación, permitiendo detectar posibles errores de código antes de ejecutarlo en el PLC. Por otro lado, se cuenta el modo de ejecución estándar, donde se carga el programa desarrollado en el PLC. Referidos a la figura 3-14, se presentan el modo de ejecución estándar y simulación, respectivamente:



Figura 3-16. Selector de modos de ejecución.

Por otro lado, si se quiere ejecutar un programa se debe pulsar en el siguiente botón, cuya función es la de establecer la conexión entre PC-PLC, este último físico o no dependiendo del modo de ejecución que esté seleccionado:



Figura 3-17. Botón de conexión con PLC.

Una vez establecida la conexión el programa permite, respecto a la figura 3-16, tanto transferir el proyecto al PLC como recibir el proyecto cargado en el PLC mediante los dos botones señalados en la siguiente imagen:



Figura 3-18. Botones de transferencia de proyectos.

En nuestro caso, se selecciona el primero para transferir el proyecto al PLC. Tras esto, el PLC comienza una fase de compilación y regeneración del programa que una vez ha finalizado permite ejecutarlo mediante el siguiente botón:



Figura 3-19. Botón de ejecución del proyecto.

4 SoMove

En este capítulo se presenta el software SoMove de *Schneider Electric*. Este programa permitirá hacer cambios en valores de parámetros, realizar un muestreo de algunos de los parámetros del servomotor para más tarde graficarlo, así como activar el movimiento del servomotor en cualquier modo de funcionamiento.

Recordar que estas operaciones pueden ser realizadas desde la interfaz HMI. Sin embargo, esta operación resulta ser bastante engorrosa y es por ello por lo que se prefiere usar una interfaz como la que ofrece SoMove.

4.1 Primeros pasos en SoMove

4.1.1 Selección de dispositivo

La conexión entre software y servo drive se realiza por protocolo Modbus mediante un cable Modbus-USB. Se procederá a conectar el extremo USB al PC en el que se disponga el SoMove y el extremo del conector Ethernet al Lexium 32M.



Figura 4-1. Cable Modbus - USB.

Una vez conectados PC y servo drive se procederá a ejecutar el software SoMove, en el que se comenzará creando un nuevo proyecto. Tras esto, SoMove solicitará que se defina el dispositivo del que se va a hacer uso mediante la siguiente pantalla:

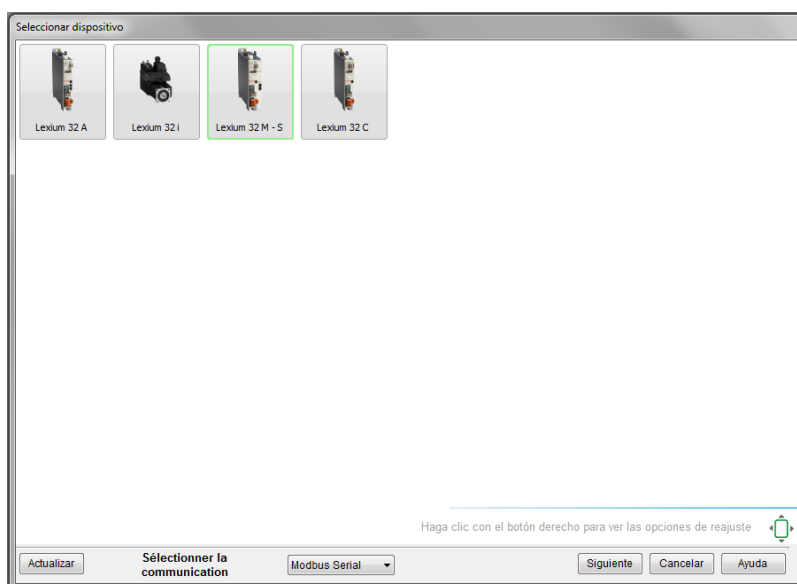


Figura 4-2. Seleccionar dispositivo.

En nuestro caso se selecciona el dispositivo “Lexium 32M – S” y se pulsa en “Siguiente”, con lo que aparecerá una nueva pantalla para crear la topología en la que se definirá tanto el modelo del servo drive como del servomotor:

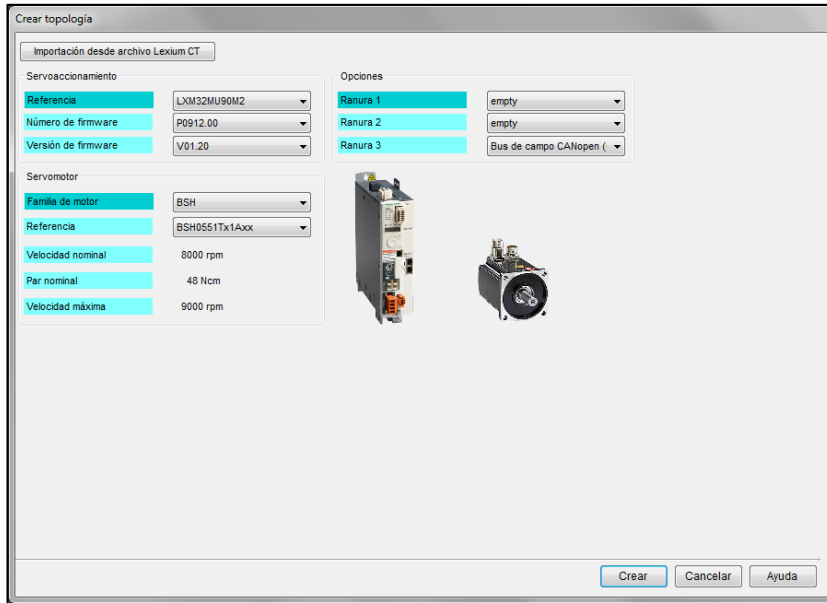


Figura 4-4-3. Crear topología.

En cuanto al servoaccionamiento se elige la referencia “LXM32MU90M2” y en cuanto al servomotor la familia BSH con referencia “BSH0551Tx1Axx”. Seguidamente, se pulsará en “Crear” con lo que aparecerá la ventana principal del SoMove:

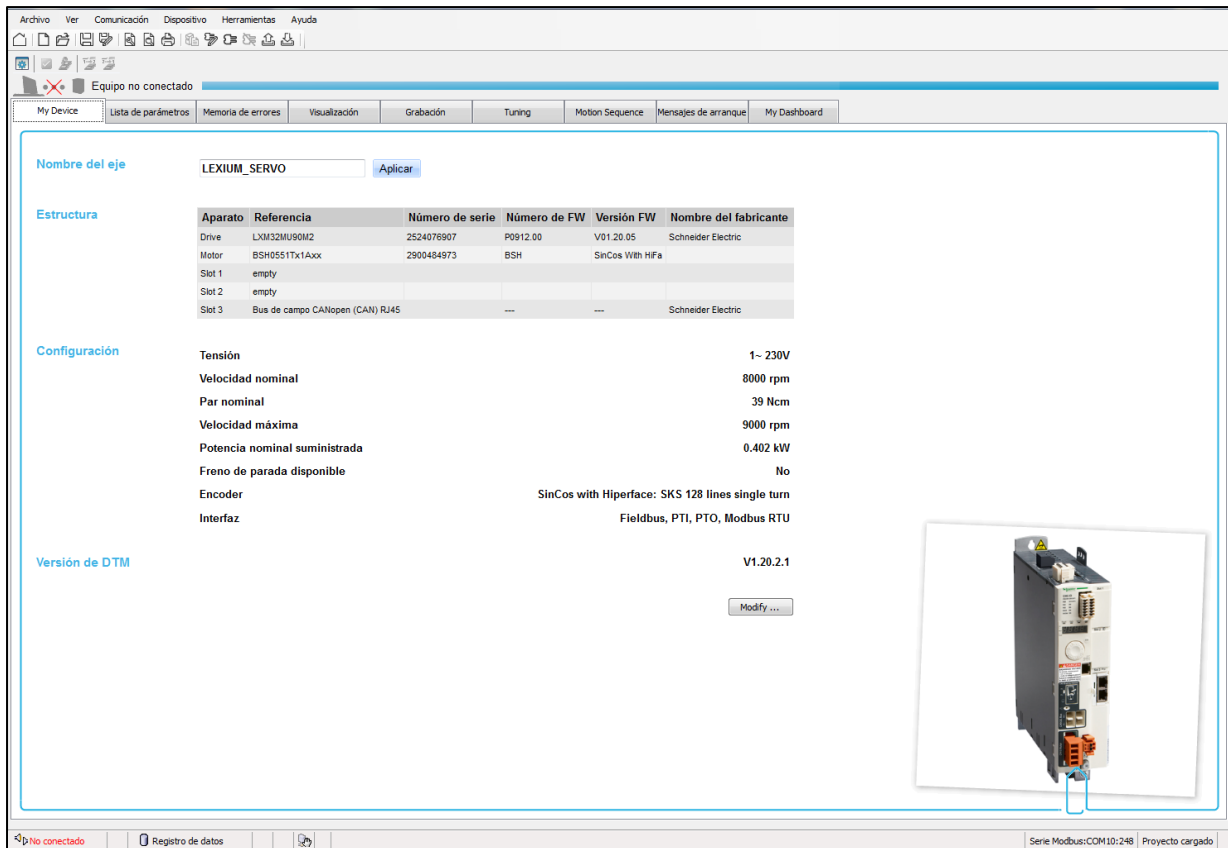


Figura 4-4. Ventana Principal de SoMove.

4.1.2 Interfaz de SoMove

Desde esta ventana se puede acceder a todas las prestaciones que ofrece este software. Se distinguen varias pestañas:

1. Ventana **“My Device”**. En esta ventana se podrá encontrar información tanto de la estructura del servo drive como de las características técnicas del servomotor.
2. Ventana **“Lista de Parámetros”**. Desde esta ventana se podrá modificar cualquier parámetro de funcionamiento del servomotor. Esta ventana es la más importante pues en los capítulos siguientes se centrará en modificar parámetros para alcanzar modos de funcionamiento determinados. Si se pulsa sobre esta ventana aparece la siguiente clasificación:

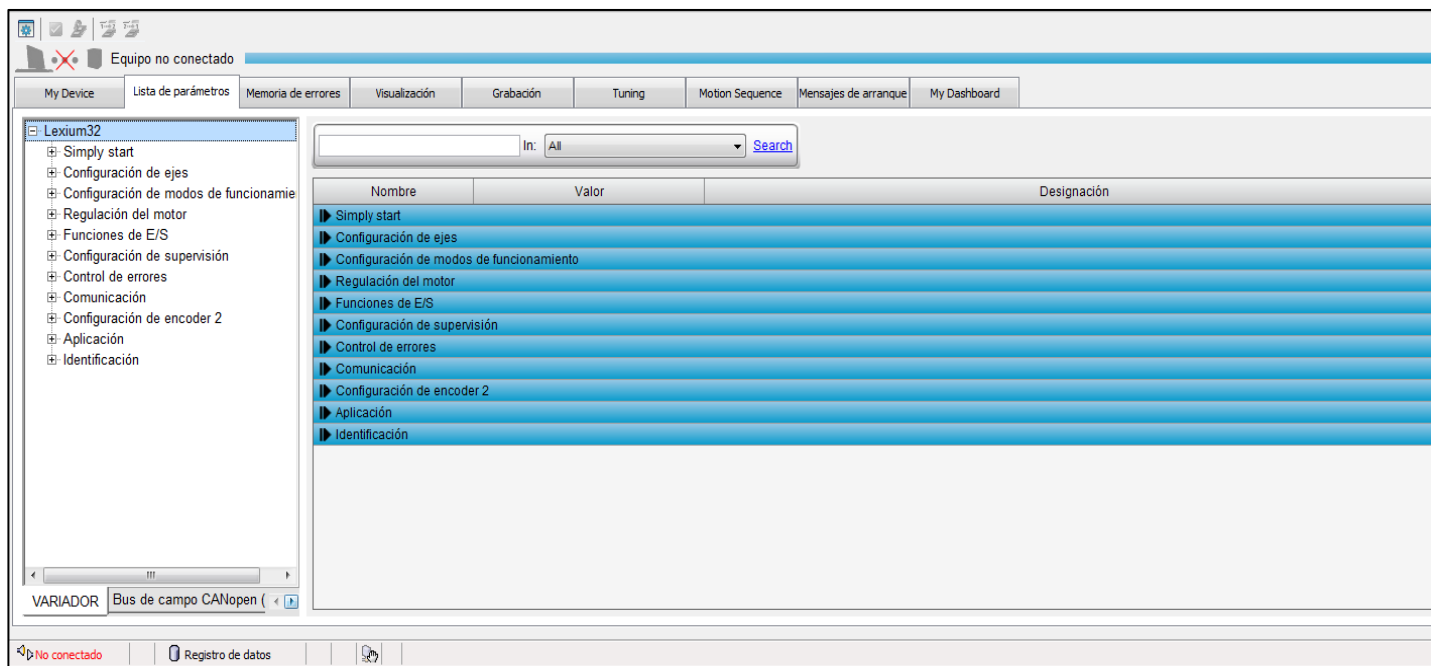


Figura 4-5. Ventana "Lista de parámetros".

Si cualquier parámetro de esta lista es modificado, la lista debe enviarse al servo drive para hacer efectivo los cambios de parámetros tras lo cual, el servo drive debe reiniciarse. Se muestra a continuación una imagen en la que se refleja estas acciones.

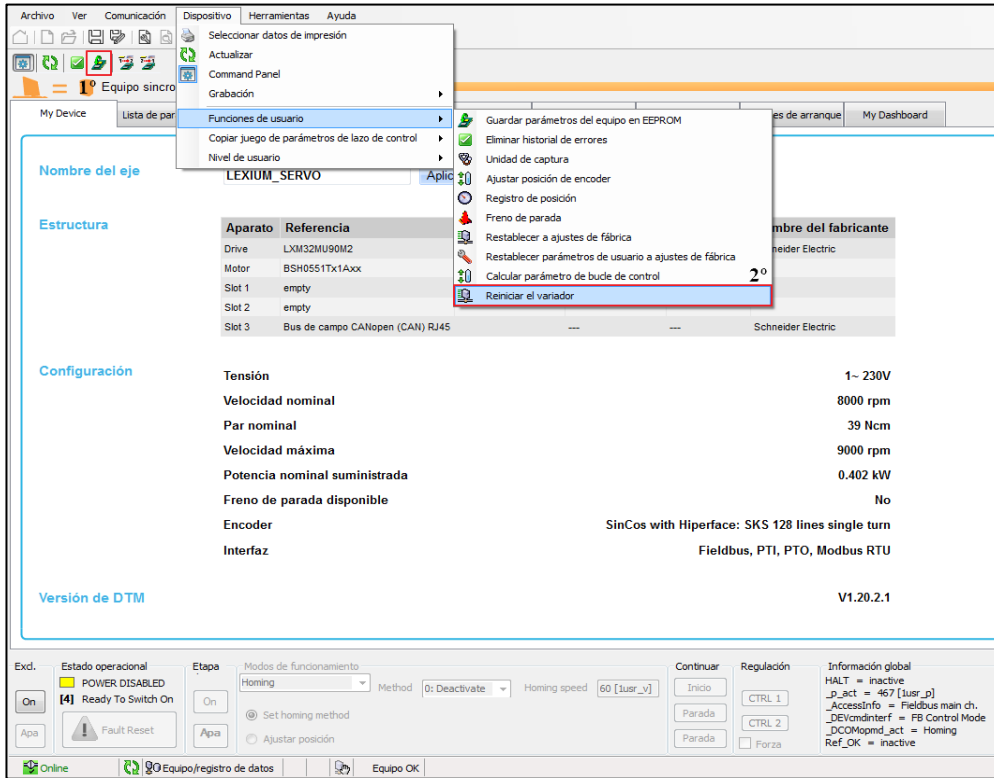


Figura 4-6. Reiniciar variador.

3. Ventana “**Memoria de errores**”. Con esta ventana se definen y clasifican los errores que se produzcan tanto en el servo drive como fallos en el movimiento en el servomotor.
4. Ventana “**Visualización**”. En esta ventana se puede encontrar información a tiempo real de las magnitudes físicas del motor tales como par aportado, posición actual, velocidad, carga en el motor, etc. La visualización se realiza mediante displays tales como la siguiente figura:

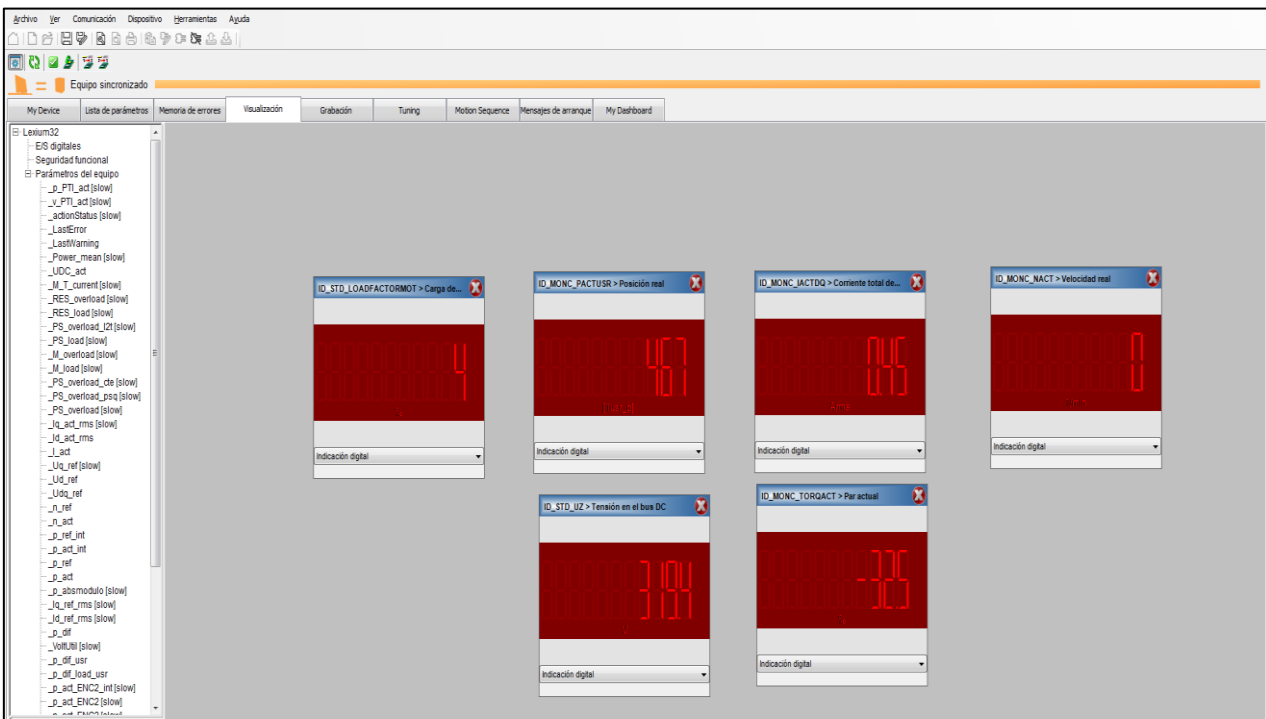


Figura 4-7. Ventana "Visualización".

5. Ventana “**Grabación**”. Mientras que la ventana “Visualización” permite ver a tiempo real algunas magnitudes físicas del motor, la ventana “Grabación” realiza un histórico de estas y las grafica.

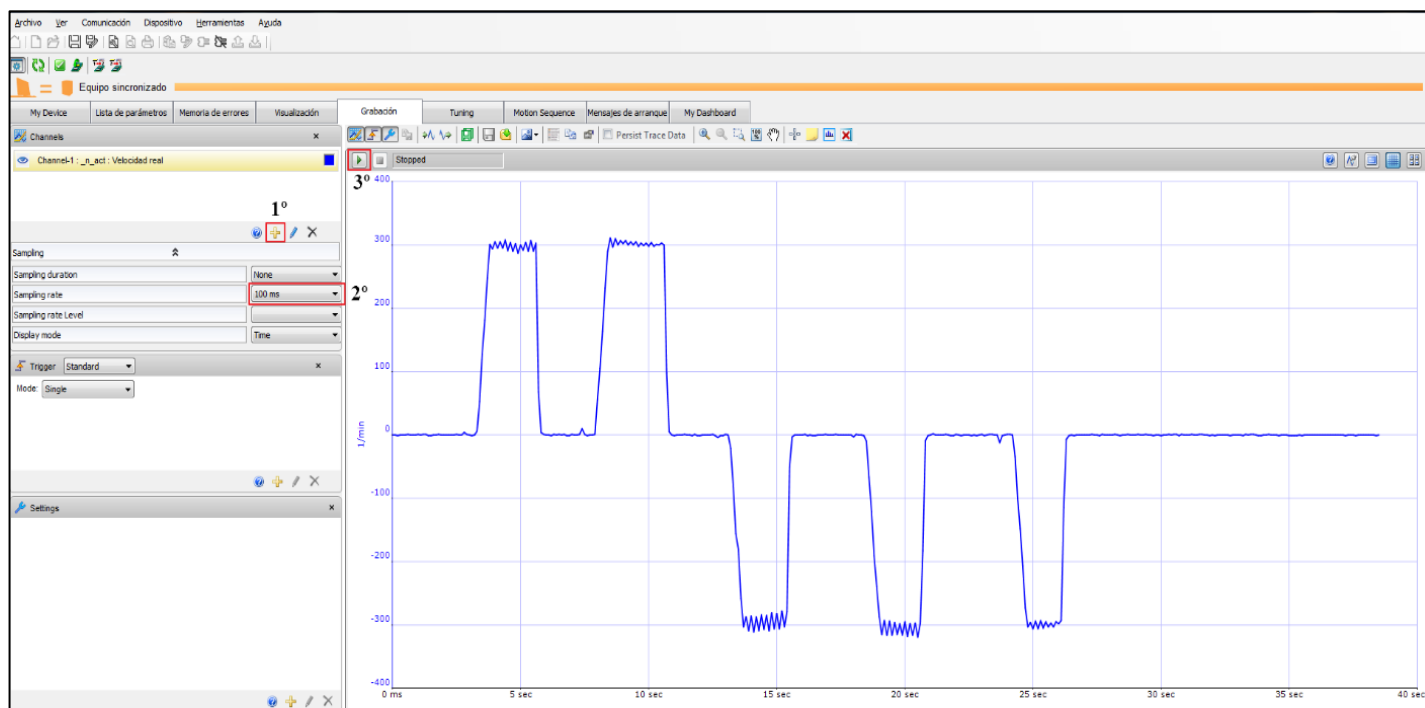


Figura 4-8. Ventana "Grabación".

Si se desea representar alguna magnitud usando SoMove se deberán seguir los siguientes pasos:

- **Paso 1:** Se pulsa en el botón indicado. Se abrirá una ventana de selección de parámetros y se elegirá la magnitud deseada.
 - **Paso 2:** Se establece un tiempo de muestreo de 100 ms, aunque este valor puede cambiarse en función de la precisión que se necesite.
 - **Paso 3:** Se pulsa el botón de inicio y se espera a que se genere la gráfica.
6. Ventana “**Tuning**”. Mediante esta ventana se ajustan diferentes parámetros pero esta vez respecto al controlador del servo drive permitiendo modificarlos en tres modos distintos, “Easy Tuning” o modo automático, “Confort Tuning” o modo semiautomático y “Expert Tuning” o modo manual.

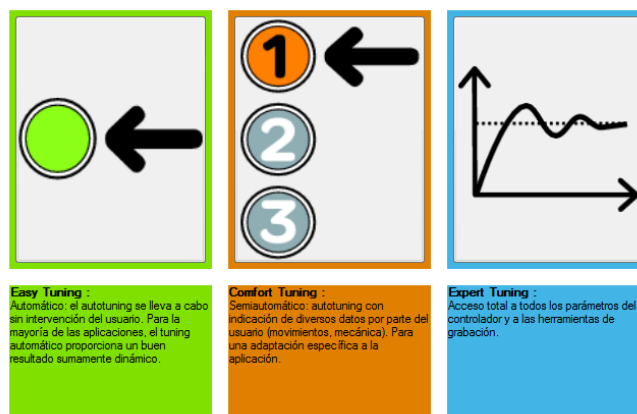


Figura 4-4-9. Ventana "Tuning".

7. Ventana “**Motion Sequence**”. Esta ventana permite añadir registros de datos que se usarán en el modo de funcionamiento “Motion Sequence”.
8. Ventana “**Mensajes de arranque**”. En esta ventana aparecen mensajes de reinicio del variador. Esto ocurre cuando se cambian los valores por defecto de los parámetros de funcionamiento.

9. Ventana **“My Dashboard”**. En esta ventana se muestran los valores de algunas variables, pudiendo elegir cual se quiere visualizar. Además, muestra los valores de las entradas y salidas digitales además de parámetros de control.

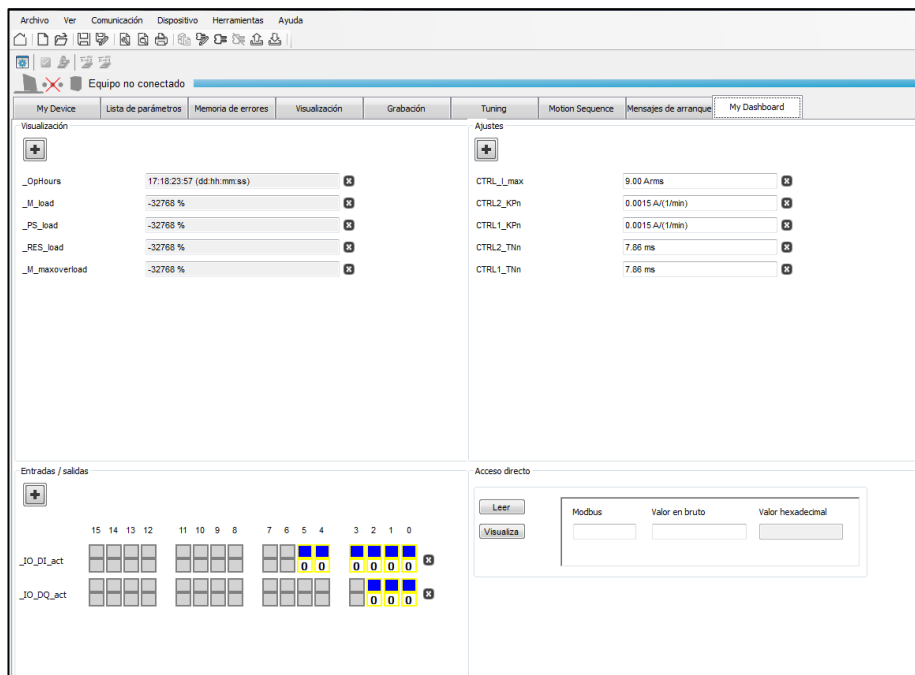


Figura 4-10. Ventana "My Dashboard".

4.1.3 Conexión y modos de funcionamiento

Una vez conocida la interfaz de SoMove, se procederá a detallar cómo se conecta este programa con el servo drive y cómo realizar algunos movimientos. Si tanto la elección del modelo como la topología fueron correctas, se podrá establecer conexión con el servo drive haciendo uso del botón **“Conectar a dispositivo”** como se muestra en la siguiente imagen.



Figura 4-11. Barra de herramientas.

Tras pulsar este botón, SoMove dará la opción de cargar un programa desde el PC hasta el servo drive o de recuperar el último programa que tuviera cargado el servo drive al PC. En nuestro caso, se elegirá la opción **“Almacenar en el dispositivo y conectar”**.

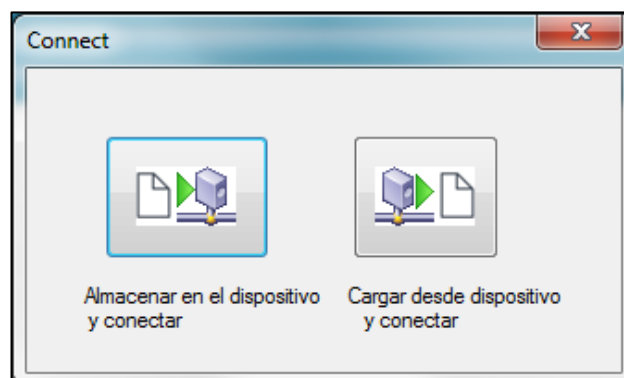


Figura 4-12. Ventana Connect.

Una vez establecida la conexión entre el SoMove y el PC, se aprecia que la interfaz cambia de color a un tono naranja, que indica que se ha conectado. En este modo hay que tener en cuenta que no todos los parámetros disponibles en la ventana “Lista de parámetros” se podrán modificar, por lo que es conveniente modificar parámetros para posteriormente conectar el dispositivo.

Una vez cargado el archivo, el programa está listo para poner en movimiento el servomotor. Para ello se debe activar la etapa de potencia siguiendo el siguiente procedimiento (1º y 2º):

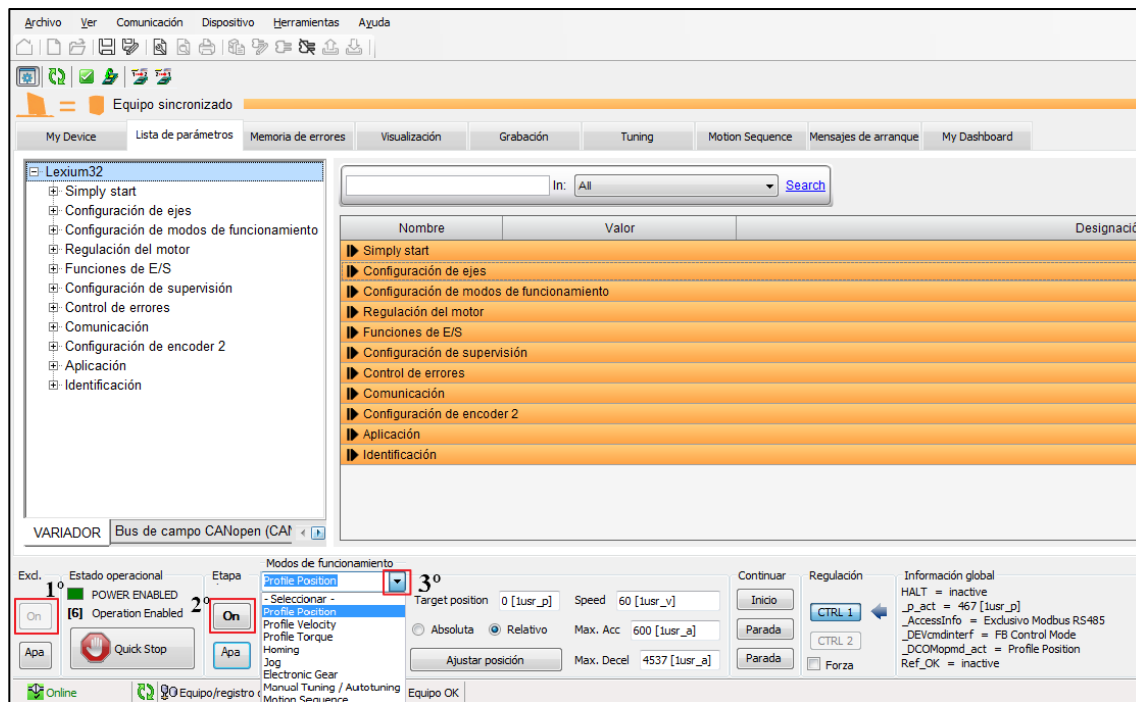


Figura 4-13. Procedimiento de activación de la etapa de potencia.

Una vez activa, se procede a activar cualquier modo de funcionamiento pulsando en el desplegable de la figura superior (3º).

Todos los modos de funcionamiento en modo de control local, es decir, de forma manual desde la pantalla de la figura 4-14 tienen una misma composición, presentando algunos parámetros en la zona derecha y por otro lado una botonera que inicia y finaliza el modo de movimiento mediante los botones “Inicio” y “Parada” respectivamente.

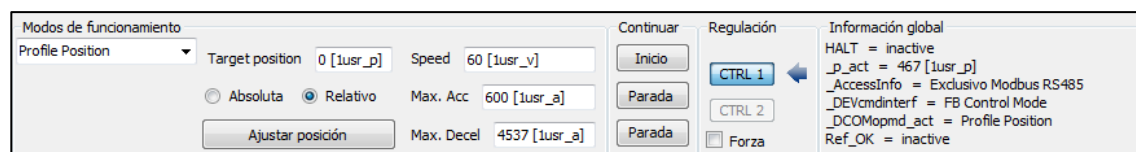


Figura 4-14. Barra de modos de funcionamiento.

Además, en todos los modos de funcionamiento se presenta información global del servomotor en el que se visualiza posición, modo de funcionamiento, así como otros datos generales.

5 UNIDADES PRÁCTICAS. USO DEL SERVOMOTOR MEDIANTE SEÑALES EXTERNAS

El capítulo que comienza está destinado a alumnos poca o ninguna formación previa en softwares como SoMove o Unity Pro. Por ello, este será desarrollado de forma didáctica, explicando todos los pasos necesarios para conseguir la implementación de librerías desarrolladas en Unity Pro, que posteriormente se expondrán en otro capítulo explicando cómo se han realizado, así como sus entradas y sus salidas. La finalidad de estas librerías es lograr el funcionamiento del servomotor de mediante señales piloto externas a partir de una interfaz HMI básica realizada en Unity Pro, apoyándose también en SoMove.

Los primeros subcapítulos se destinarán a la familiarización por parte del alumno del software SoMove, en concreto, con la Ventana “Lista de parámetros” así como con la modificación de parámetros desde la interfaz HMI con la que cuenta el servo drive Lexium 32M.

5.1 Ejemplo introductorio I

5.1.1 Enunciado

En este subcapítulo se expondrá un ejemplo de funcionamiento del servomotor haciendo uso de un panel de mandos. En concreto, se usará únicamente un botón giratorio tal y como se indica en la siguiente figura:



Figura 5-1. Panel de mandos.

El objetivo es emplear el botón giratorio seleccionada en la figura 5-1, que es solidario al eje de un encoder para obtener las fases A/B, que serán transmitidas al Lexium y que el servomotor replique este movimiento. El alumno deberá ser capaz de saber con la información ya expuesta en los capítulos anteriores qué parámetros deben ser modificados y cómo debe realizarse la conexión teniendo en cuenta que el panel ya se encontraría alimentado y que el panel de mandos presenta una conexión de salida tipo rj45.

5.1.2 Solución propuesta

5.1.2.1 Conexionado

En cuanto al conexionado, únicamente es necesario conectar la salida del panel de mandos a la entrada PTI, CN5, del Lexium como aparece en la siguiente figura:



Figura 5-2. Conexión. Ejemplo I.

5.1.2.2 Configuración de parámetros y modo de funcionamiento

Funciones de E/S				
▶ Entradas digitales				
▶ Salidas digitales				
▼ PTI				
PTI_signal_type	A/B Signals	Tipo de señal piloto para la interfaz PTI		
PTI_pulse_filter	0.25 µs	Tiempo de filtrado para señales de entrada de la interfaz PTI	0.00 µs	13.00 µs
InvertDirOfCount	Inversion On	Inversión de la dirección de conteo en la interfaz PTI		
OFSp_RelPos1	0 Inc	Posición offset relativa 1 para movimiento offset	-2147483648 Inc	2147483647 Inc
OFSp_RelPos2	0 Inc	Posición offset relativa 2 para movimiento offset	-2147483648 Inc	2147483647 Inc

Figura 5-3. Configuración de parámetros. Ejemplo I.

Para ejecutar un correcto funcionamiento se debe configurar el parámetro *PTI_signal_type* a *A/B Signals* mientras que se deja a libre disposición el valor del parámetro *InvertDirOfCount*.

Por otro lado, se debe configurar el modo de funcionamiento estableciéndolo en modo Electronic Gear. En cuanto al parámetro *GearRatio* se deja al alumno la configuración de este parámetro para observar cómo reacciona el servomotor ante un cambio en el valor de este parámetro. Aun así, si se desea que por cada vuelta de la botón rotatorio del panel de mandos el motor de una vuelta, se deberá configurar el parámetro *GearRatio* en un valor de 28.

5.2 Ejemplo introductorio II

5.2.2 Enunciado

En este ejemplo se contará con la presencia de dos servoaccionamientos, es decir, dos servomotores y dos drive Lexium 32M. Se pretenderá configurar los dos motores de tal forma que al accionar un motor en un modo de funcionamiento determinado, el otro realice un seguimiento del movimiento del anterior. Además se pretenderá que el alumno practique modificando parámetros desde el software SoMove así como desde las entradas físicas de las que se disponen en el driver mediante la pantalla HMI. Además, se deberá conocer cómo se deben establecer las conexiones entre los drivers para que la operación que se pretende abordar se realice de manera correcta.

5.2.3 Solución propuesta

5.2.3.1 Conexión

En cuanto al conexionado, es necesario conectar la salida PTO (CN4) del Lexium maestro a la entrada PTI (CN5) del Lexium esclavo como se muestra en la siguiente figura:

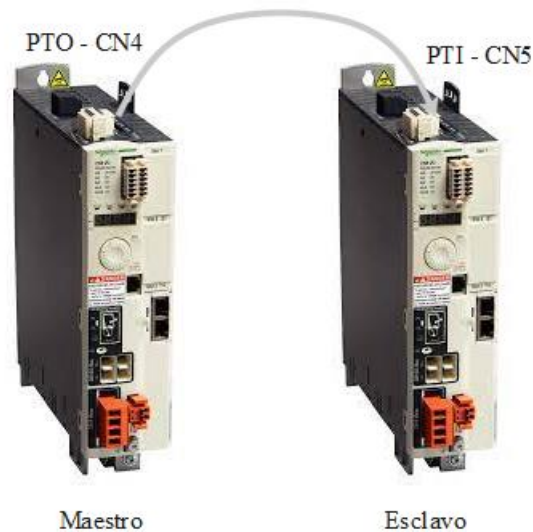


Figura 5-4. Conexionado. Ejemplo II.

5.2.3.2 Configuración de parámetros y modo de funcionamiento

En este caso, se configurará el driver esclavo desde SoMove mientras que el driver maestro se configurará de forma física, por lo que para poner en funcionamiento todo el sistema habrá que configurar también el modo de funcionamiento desde estas entradas físicas.

Configuración del driver maestro

Por un lado se modifican los parámetros relacionados con la salida PTO mientras que por otro se configurará el modo de funcionamiento, que se ha elegido el modo Jog por su simplicidad. Para ello, será necesario realizar con el menú HMI:

- *PTO_mode*: ConF → ACG → PtoN
- Modo Jog: oP → JoG → JGSt. En este último, girando el botón de navegación se puede cambiar entre 4 tipos de movimiento:
 - **JG-** : Movimiento lento en dirección positiva
 - **JG=** : Movimiento rápido en dirección positiva
 - **-JG** : Movimiento lento en dirección negativa
 - **=JG** : Movimiento rápido en dirección negativa

El movimiento se inicia pulsando el botón de navegación.

Configuración del driver esclavo

Estas modificaciones se realizan desde SoMove por lo que la configuración de parámetros es la siguiente:

Nombre	Valor	Designación	Valor mín.	Valor máx.
▼ Simply start				
▶ Configuración básica				
▼ Electronic Gear				
GEARratio	Gear Factor	Selección de la relación de transmisión		
GEARnum	35	Numerador del factor de engranaje	-2147483648	2147483647
GEARdenom	1	Denominador del factor de engranaje	1	2147483647
GEARnum2	1	Numerador de la relación de transmisión número 2	-2147483648	2147483647
GEARdenom2	1	Denominador de la relación de transmisión número 2	1	2147483647
▶ Configuración de ejes				
▶ Configuración de modos de funcionamiento				
▶ Regulación del motor				
▼ Funciones de E/S				
▶ Entradas digitales				
▶ Salidas digitales				
▼ PTI				
PTI_signal_type	A/B Signals	Tipo de señal piloto para la interfaz PTI		
PTI_pulse_filter	0.25 µs	Tiempo de filtrado para señales de entrada de la interfaz PTI	0.00 µs	13.00 µs
InvertDirOfCount	Inversion Off	Inversión de la dirección de conteo en la interfaz PTI		
OFSp_RelPos1	0 Inc	Posición offset relativa 1 para movimiento offset	-2147483648 Inc	2147483647 Inc
OFSp_RelPos2	0 Inc	Posición offset relativa 2 para movimiento offset	-2147483648 Inc	2147483647 Inc

Figura 5-5. Configuración de parámetros. Ejemplo II.

Para ejecutar un correcto funcionamiento se debe configurar el parámetro *PTI signal type* a *A/B Signals* mientras que se deja a libre disposición el valor del parámetro *InvertDirOfCount*, de la misma forma en la que se realice en el ejemplo I.

Por otro lado, se debe configurar el modo de funcionamiento estableciéndolo en modo Electronic Gear. En cuanto al parámetro *GearRatio* se deja al alumno la configuración de este parámetro para observar cómo reacciona el servomotor ante un cambio en el valor de este parámetro. Aun así, si se desea que por cada vuelta de la ruleta del panel de mandos el motor de una vuelta, se deberá configurar el parámetro *GearRatio* en un valor de 35.

Se pueden realizar otras prácticas consultando la siguiente referencia [4].

5.3 Unidades prácticas. Implementación de librerías.

En este apartado se introduce el autómata programable para la implementación de las librerías desarrolladas en Unity Pro. El alumno, siguiendo unos pasos detallados en cuanto a la configuración de Unity Pro se refiere, deberá ser capaz de configurar los parámetros, seleccionar el modo de funcionamiento adecuado en SoMove para un correcto funcionamiento de las funciones así como saber que función es la más adecuada en cada situación una vez se expongan todas las unidades prácticas.

Se realizarán 4 prácticas diferentes junto con el autómata programable, cada una con una librería diferente centradas en la utilización del servomotor de la misma forma en la que se utilizaría un motor paso a paso. Recordar que para accionar el movimiento de un motor paso a paso se alimenta al driver de este con tres señales:

1. **PUL**: Señal de tren de pulsos continuo.
2. **DIR**: Señal de dirección de movimiento.
3. **EN**: Señal de habilitación. En este caso, esta señal no será necesaria.

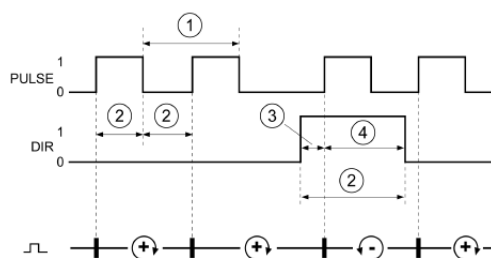


Figura 5-6. Señales de control de un motor paso a paso.

Para ello se seguirá el protocolo de comunicación maestro-esclavo donde un M340 junto con una tarjeta rápida EHC200 serán el grupo maestro, mientras que se configurará el servomotor como esclavo siguiendo las consignas del PLC.

5.3.2 Unidad práctica I

5.3.2.1 Enunciado

En esta práctica el alumno controlará el servomotor mediante una señal de trenes de pulsos procedentes de una tarjeta rápida EHC200 [5]. Con esta librería, el programa dará una serie de pulsos determinados realizando así un control de posición o un seguidor de referencias en pulsos. Este control se realizará en bucle abierto pues no hay ningún elemento que realmente e informe del progreso del motor, es decir, no se sabe a ciencia cierta si el motor se está moviendo pues el conteo de pulsos lo realizará la propia tarjeta EHC200 y se obvia la información disponible desde SoMove.

En este ejercicio el alumno aprenderá a configurar las tarjetas que se requieren para implementación de esta librería.

5.3.2.2 Solución

Configuración en Unity Pro

Suponiendo que se siguen los pasos del apartado 3 en el que se añadía el modelo de autómatas, ahora se debe añadir la tarjeta rápida en la ranura número 1, ya que físicamente se encuentra en dicha ranura. Debe mantenerse la igualdad entre los módulos añadidos en Unity Pro con respecto a los físicos. Para ello se realizan los siguientes pasos:

1. Añadir la tarjeta rápida en la ranura número 1, ya que físicamente se encuentra en dicha ranura. Debe mantenerse la igualdad entre los módulos del programa con los físicos. Para ello se realiza lo siguiente:

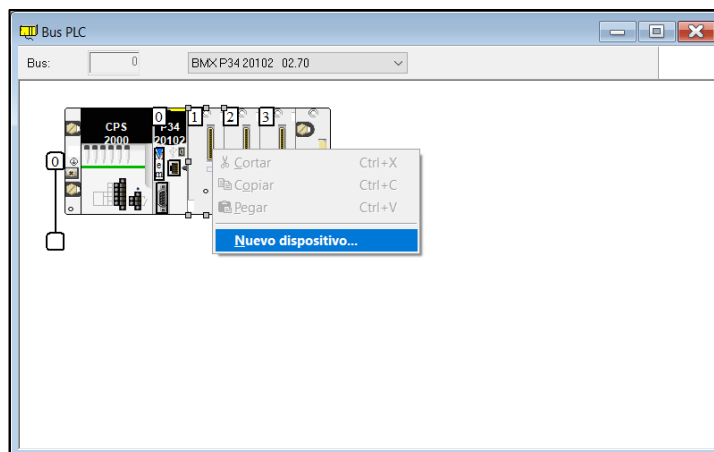


Figura 5-7. Introducción de un nuevo dispositivo.

2. Seleccionar el módulo de conteo BMX EHC0200:

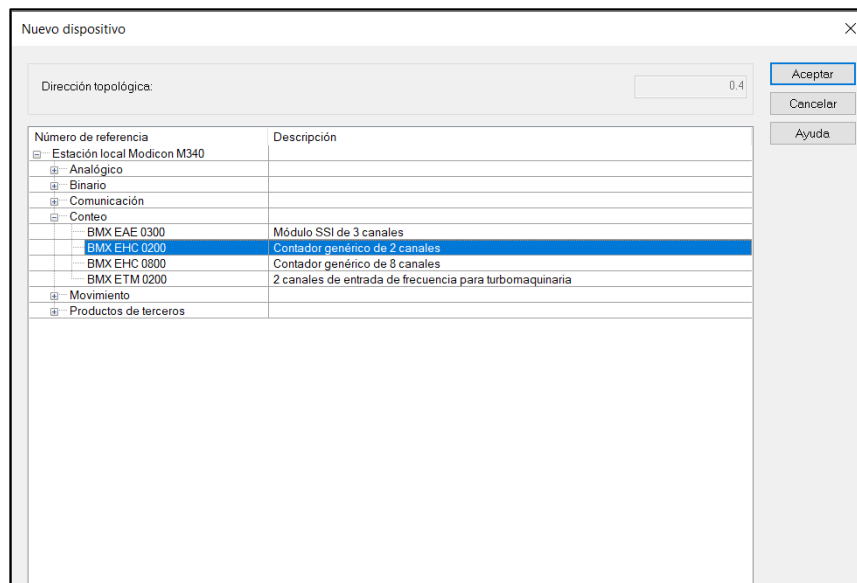


Figura 5-8. Selección del módulo EHC0200.

3. Por último, configurar la tarjeta rápida EHC200 de manera correcta:

- a. Se configura el canal 0 en modo PWM en la tarjeta EHC200. Para ello se le asigna la función 'Modo de modulación de ancho de pulsos'.

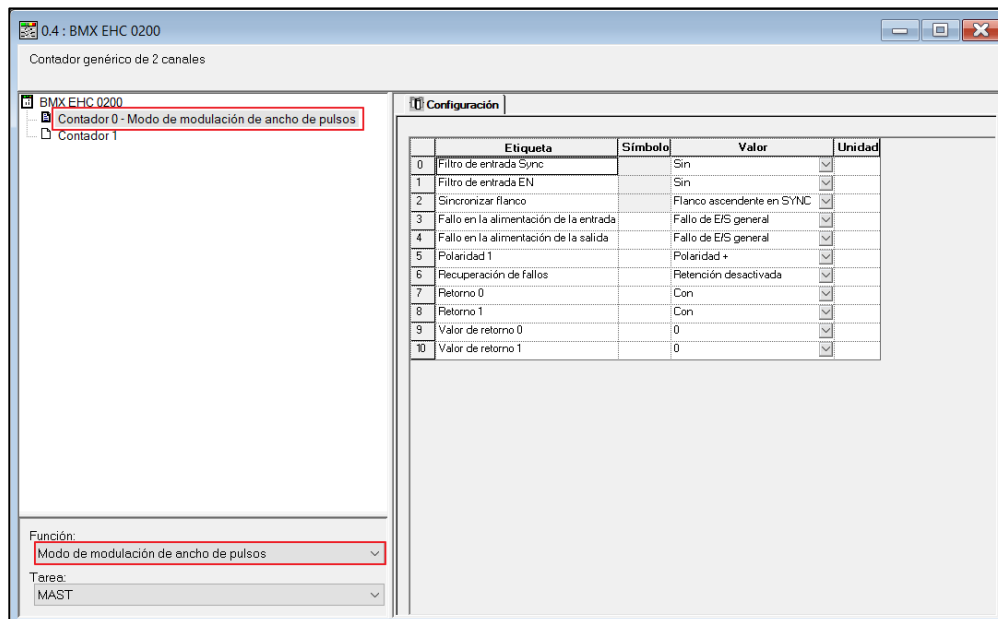


Figura 5-9. Configuración del canal 0.

- b. Se configura el canal 1 con la función ‘Modo contador libre de 32 bits’ y se configuran las etiquetas marcadas con la función que aparece en la siguiente figura:

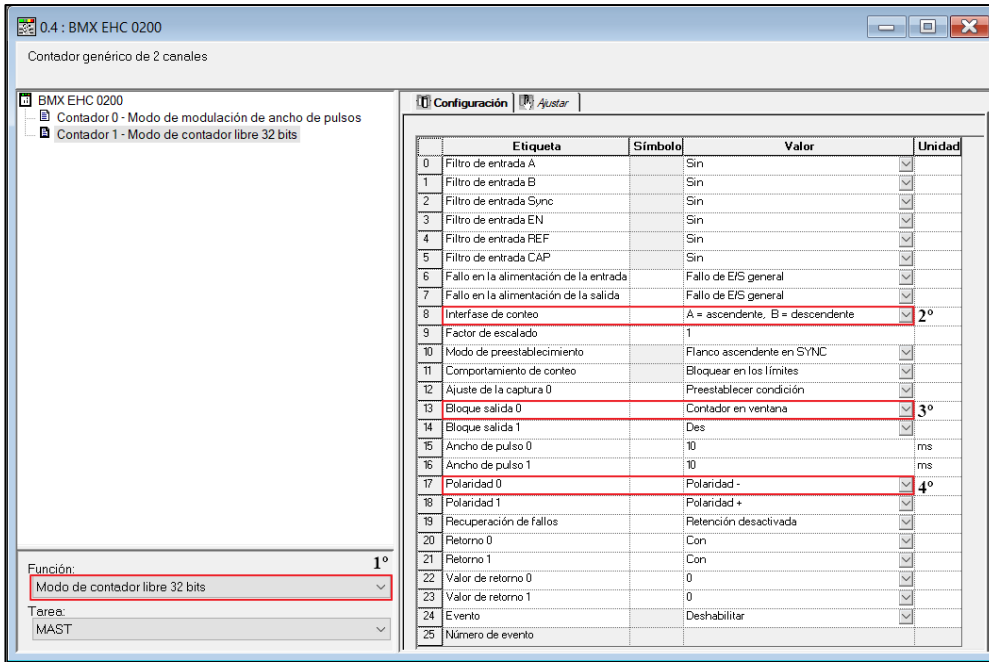


Figura 5-10. Configuración del canal 1.

- c. Definir la variable estructura de manejo del canal 0:

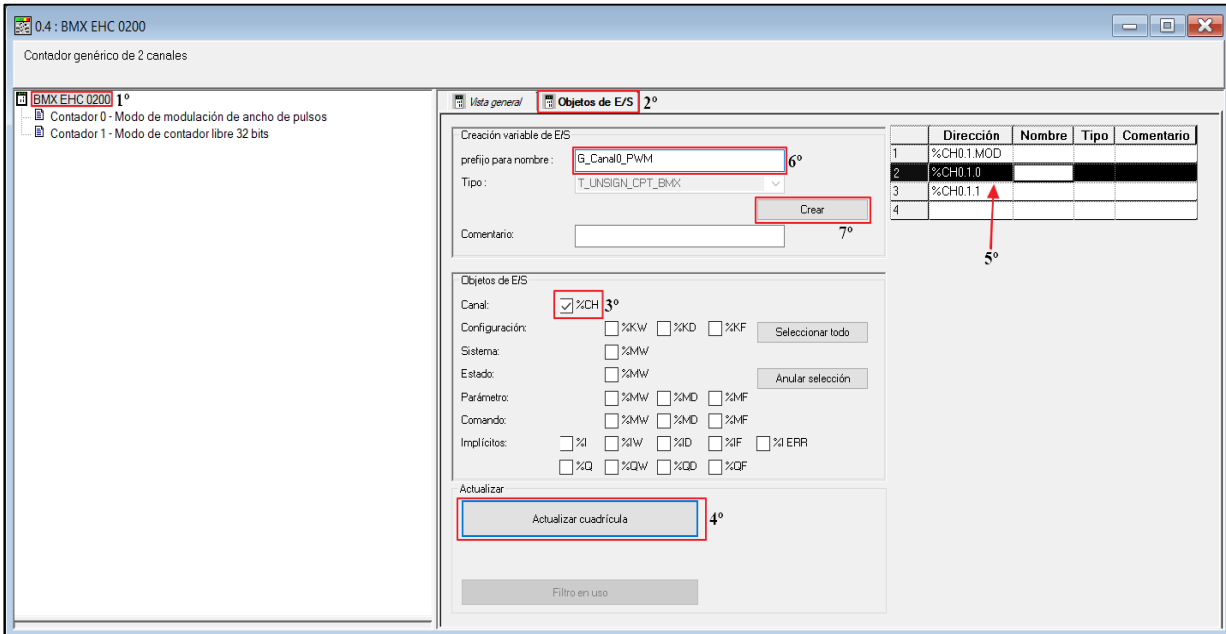


Figura 5-11. Estructura del canal 0.

d. Definir la variable estructura de manejo del canal 1:

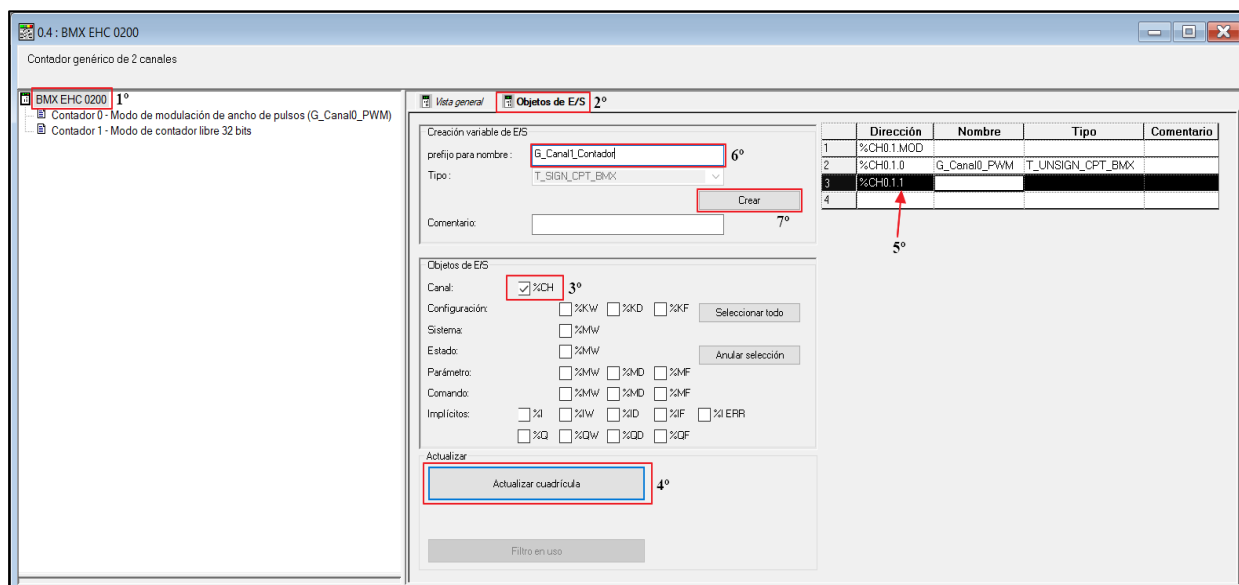


Figura 5-12. Estructura del canal 1.

e. Definir la variable estructura de 'Estado' de la tarjeta:

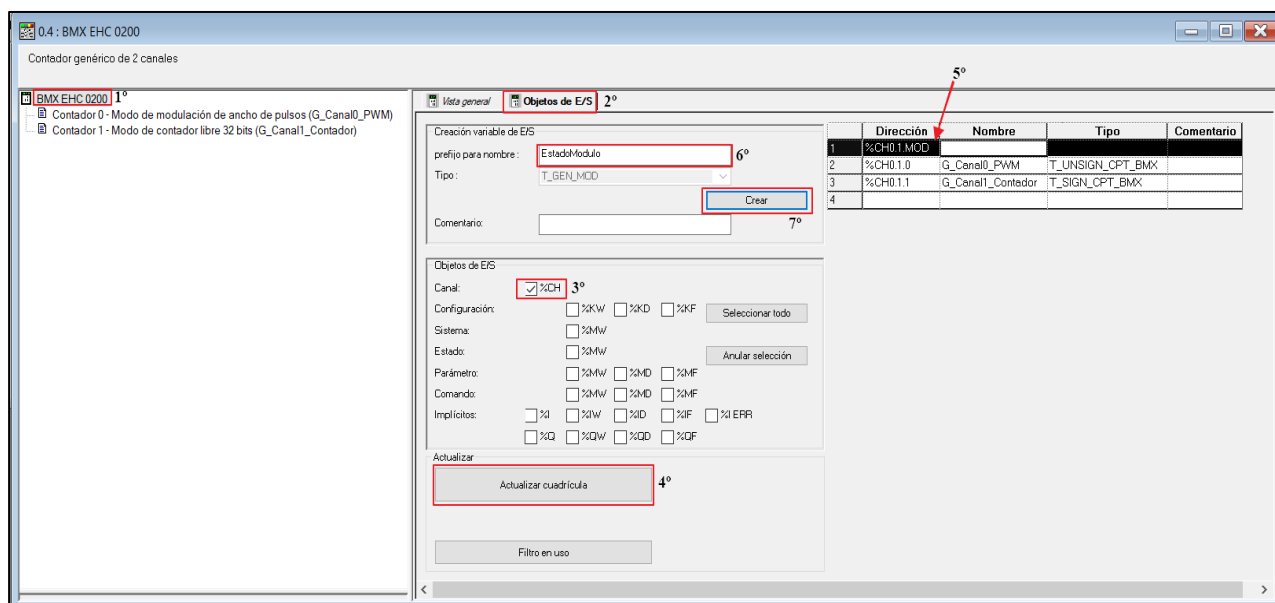


Figura 5-13. Estructura de estados.

Llegados a este punto ya se tiene completada la configuración del PLC y de la tarjeta rápida.

Configuración en SoMove

El primer paso será seleccionar el modo de funcionamiento correcto, siendo en este caso “Electronic Gear”. Una vez seleccionado este modo, el movimiento del servomotor queda gobernado por una serie de señales externas configurables dependiendo del tipo (según configuración) con las que se alimentará al Lexium 32M mediante la entrada PTI (CN5). El parámetro *PTI_signal_type* elige el tipo de señal con la que se opera entre las que se encuentran A/B, P/D y CW/CCW.

PTI					
PTI_signal_type	P/D Signals	Tipo de señal piloto para la interfaz PTI			1284
PTI_pulse_filter	0.25 µs	Tiempo de filtrado para señales de entrada de la interfaz PTI	0.00 µs	13.00 µs	1374
InvertDirOfCount	Inversion On	Inversión de la dirección de contaje en la interfaz PTI			2062
OFSp_RelPos1	0 Inc	Posición offset relativa 1 para movimiento offset	-2147483648 Inc	2147483647 Inc	10000
OFSp_RelPos2	0 Inc	Posición offset relativa 2 para movimiento offset	-2147483648 Inc	2147483647 Inc	10004

Figura 5-14. Parámetros PTI.

Por tanto, se configurará el campo *PTI_signal_type*=P/D. Además, en caso de ser necesaria una inversión de la dirección del movimiento, se empleará el parámetro *InvertDirOfCount*.

Tras esto, se debe ajustar la relación de transmisión mediante el parámetro *GearRatio*. A este parámetro se le otorga un valor de 200. Además, en la siguiente figura se muestra el valor de todos los parámetros que entran en juego en el modo Electronic Gear:

Electronic Gear					
IO_GEARmethod	Position Synchronization Immediate	Modo de procesamiento para el modo de funcionamiento Electronic Gear			1326
GEARratio	200	Selección de la relación de transmisión			9740
GEARjerklim	Apagado	Activación de la limitación de tirones			9742
GEARpos_v_max	0 [1usr_v]	Limitación de la velocidad para el método de sincronización de posición	0 [1usr_v]	2147483647 [1usr_v]	9746
GEARnum	200	Numerador del factor de engranaje	-2147483648	2147483647	9736
GEARdenom	1	Denominador del factor de engranaje	1	2147483647	9734
GEARnum2	1	Numerador de la relación de transmisión número 2	-2147483648	2147483647	9754
GEARdenom2	1	Denominador de la relación de transmisión número 2	1	2147483647	9752
GEARdir_enabl	Both	Dirección de movimiento activada para modo de funcionamiento Electronic Gear (engranaje electrónico)			9738
InvertDirOfCount	Inversion On	Inversión de la dirección de contaje en la interfaz PTI			2062
GEARposChgMode	Apagado	Tratamiento de las modificaciones de posición estando desactivada la etapa de potencia			9750
OFSp_RelPos1	0 Inc	Posición offset relativa 1 para movimiento offset	-2147483648 Inc	2147483647 Inc	10000
OFSp_RelPos2	0 Inc	Posición offset relativa 2 para movimiento offset	-2147483648 Inc	2147483647 Inc	10004
OFSp_target	60 [1usr_v]	Velocidad de destino para movimiento offset	1 [1usr_v]	2147483647 [1usr_v]	9992
OFSp_Ramp	600 [1usr_a]	Aceleración y deceleración para movimiento offset	1 [1usr_a]	2147483647 [1usr_a]	9996

Figura 5-15. Parámetros Electronic Gear. Ejercicio práctico I.

Conexionado

La comunicación entre PLC-Servo drive se realiza mediante los canales PTI (CN5), que recibe las consignas de pulso y dirección provenientes de la tarjeta rápida EHC200(, y por el canal PTO (CN4), que emite la entrada PTI desde el servo drive también a la tarjeta rápida.)

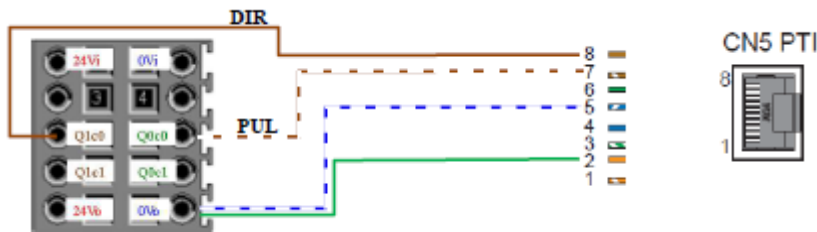


Figura 5-16. Conexión de salidas del EHC200 con CN5.

Por otro lado, el conexionado del canal 0, generador de pulsos, y canal 1, contador de pulsos, se muestra en la siguiente figura:

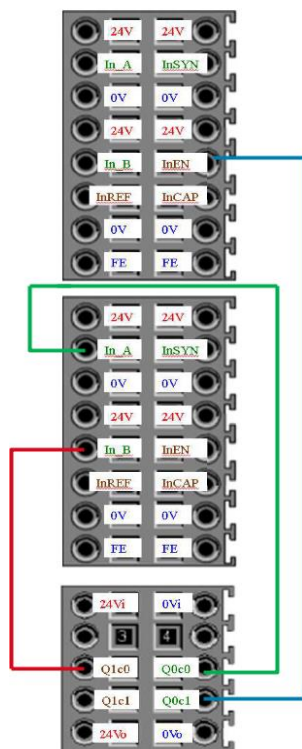


Figura 5-17. Conexión EHC0200.

En la siguiente figura se muestra el conexionado general en el que se incluyen el resto de elementos externos:

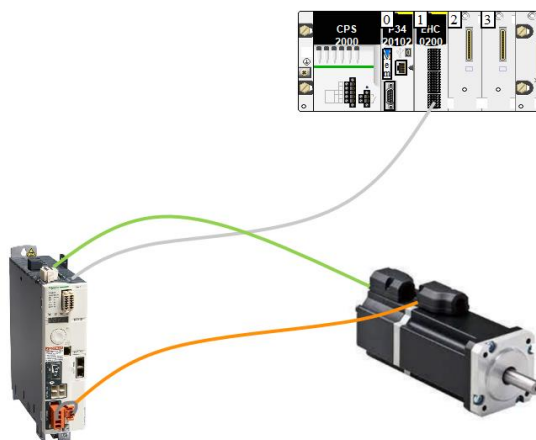


Figura 5-18. Conexión general. Ejemplo práctico I.

Uso de la librería desde Unity Pro

El software Unity Pro permite crear interfaces HMI. En este proyecto se utilizarán para establecer el valor de las entradas de las funciones desarrolladas así como ver las salidas que proporcionan estas. En este caso, para la función del ejemplo práctico I se encontrará con una interfaz como la siguiente:

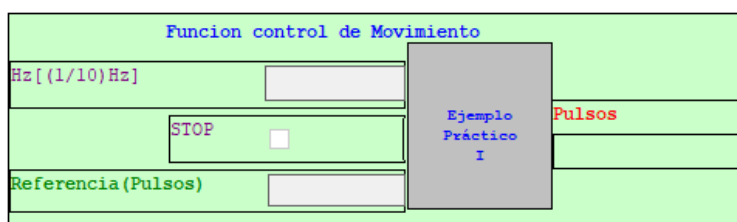


Figura 5-19. Interfaz HMI. Ejemplo práctico I.

De esta forma, se dará valor a la frecuencia de la señal de pulsos, que se introducirá en décimas de Hz, así como el valor de la referencia de pulsos que se quiere aportar al servomotor. Se cuenta también con una entrada de STOP que se utilizará para detener el motor. Como salida se observa el contador de pulsos de la función en la casilla ‘Pulsos’.

En el apartado 6 se expondrá el código en LD de la función de este ejemplo.

5.3.3 Ejemplo práctico II

En este ejemplo el alumno volverá a controlar el servomotor mediante señales externas de pulso y dirección pero en este caso se contará con una entrada de señal A/B de realimentación proporcionada por el encoder del servomotor. Lo que se consigue usando un encoder es que el PLC no se rija por los pulsos enviados sino que lo haga por los pulsos de encoder recibidos, conociendo gracias a estos si el servomotor está realizando un movimiento y en qué dirección lo está realizando.

El programa constará de dos partes, una de ellas tendrá la finalidad de controlar el servomotor en modo seguidor de referencias y la otra será una secuencia para encontrar una posición Home o posición origen del movimiento el cual comenzará mediante una señal de entrada digital proveniente de un sensor capacitivo.

5.3.3.1 Solución propuesta

Configuración en Unity Pro

Dado que no se requiere de otra tarjeta externa para esta aplicación, se mantendrá la configuración de Unity Pro tal y como se realizó en el apartado 5.3.2.2.

Configuración en SoMove

En cuanto al modo de funcionamiento se seleccionará el modo ‘Electronic Gear’. Por otro lado, los parámetros respectivos a la entrada PTI se mantienen con la misma configuración que en el ejemplo práctico anterior, es decir, se mantiene *PTI_mode=P/D signal*. Por otro lado, en este ejemplo se mantiene el valor de *GearRatio* a 200.

En este caso, el PLC se realimenta de la posición del motor mediante el uso de un encoder. En este caso, es el encoder del servomotor el que realiza dicha labor. Esta configuración es la referida a la salida PTO (CN4) y se realiza mediante el parámetro *PTO_mode*, que permite la realimentación de la posición según distintos tipos de lecturas de encoder (modos 1,2,4,5,6). Por tanto, hay que configurar este parámetro como *PTO_mode=4150 EncInc*, como se muestra en la siguiente figura:

▼ PTO					
ESIM_scale	4150 EncInc	Resolución de la simulación de encoder	8 EncInc	65535 EncInc	1322
PTO_mode	PTI Signal	Modo de utilización de la interfaz PTO			1342
ESIM_HighResolution	0 EncInc	Simulación de encoder: resolución alta	0 EncInc	268431360 EncInc	1380

Figura 5-20. Configuración PTO. Ejemplo práctico II.

Conexionado

En este caso, al contar con realimentación habrá que utilizar tanto el canal PTI, por el que se recibirán los pulsos, como el PTO, por el que se enviará la realimentación de la posición. Dado que las salidas que proporciona el canal PTO son de 5V y las entradas a la tarjeta EHC200 son de 24V, es necesaria una adaptación de la señal. La siguiente ilustración refleja el conexionado necesario para suplir este efecto. Además, como se ha comentado anteriormente, el programa se ayuda en la fase de homing de un sensor de presencia que debe estar alimentado a 24V y estar cableado directamente con la entrada IN_REF de la tarjeta EHC200.

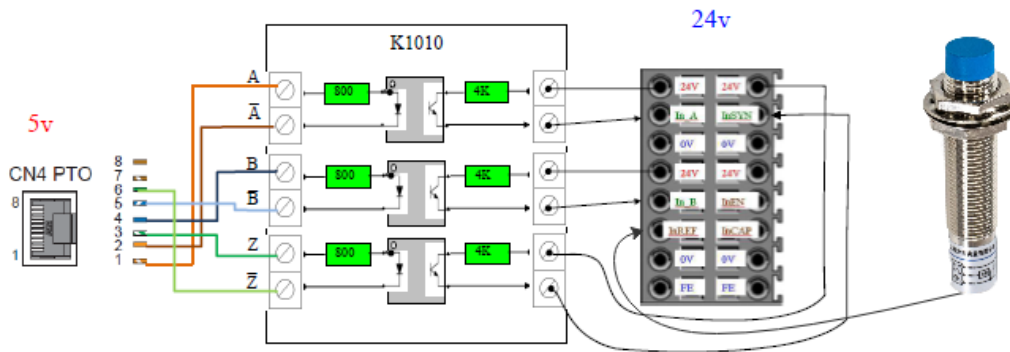


Figura 5-21. Conexión del Canal 1.

A continuación se expone de una imagen del conexionado general:

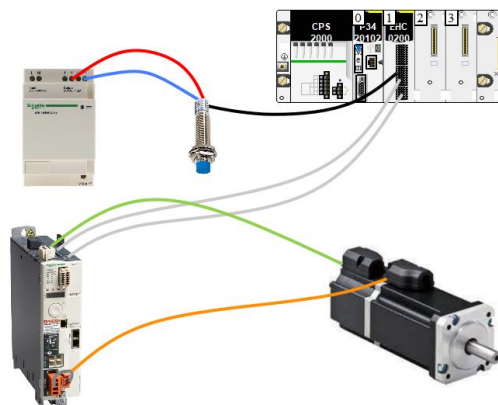


Figura 5-22. Conexión general. Ejemplo práctico II.

Uso de la librería desde Unity Pro

De nuevo en Unity Pro se encontrará con una interfaz HMI para modificar los parámetros de la función. En concreto, en el ejemplo práctico II esta interfaz es la que se muestra en la siguiente figura:

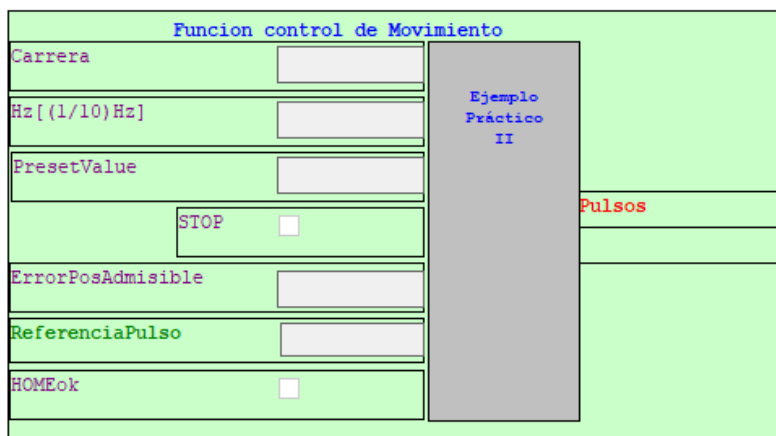


Figura 5-23. Interfaz HMI. Ejemplo práctico II.

En este caso, se podrán modificar los valores tanto de la carrera, de la frecuencia de la señal de pulsos, del 'Preset Value' o valor de preselección tras un realizar un home, del error de posición admisible y de la referencia de pulsos. Por otro lado, se cuenta con unas casillas de verificación para realizar una parada de emergencia, en caso de activar la casilla 'STOP' o de realizar un recalibrado, en caso de activar la casilla 'HomeOk'. En cuanto a la salida, se puede observar el valor del contador de pulsos en la casilla 'Pulsos'.

En el apartado 6 se expondrá el código de la función en LD de este ejemplo.

5.3.4 Ejemplo práctico III

5.3.4.1 Enunciado

Este ejemplo es muy parecido al ejemplo práctico II y se ha realizado para tener una mayor amplitud con respecto a lo que se puede realizar con el servo driver. De esta forma, el programa realizará el mismo funcionamiento, es decir, seguirá contando con el modo seguidor de referencias usando también la señal del encoder y por otro lado de un modo homing. Este último es el que va a cambiar ligeramente con respecto al ejemplo II, pues lo que se pretende es que la librería desarrollada se mejore y pase a detectar errores en el posicionado, concretamente en el caso de que el PLC esté mandando pulsos y el dispositivo no realice ningún movimiento. En dicho caso, el programa mandará un mensaje con la finalidad de que se realice un recalibrado, en este caso, de realizar un home al equipo.

De esta forma, el home se hará de forma manual mediante SoMove por lo que en este caso no se conectará el sensor a la entrada IN_REF de la tarjeta rápida EHC0200 sino que se conecte al servo drive pues es quien realizará la fase de homing.

5.3.4.2 Solución adoptada

Configuración en Unity Pro

La configuración en Unity Pro se mantendrá del mismo modo que se tuvo en el ejemplo práctico II.

Configuración en SoMove

En este caso, como el proceso de recalibrado se realizará desde SoMove se tendrá que ir modificando el modo de funcionamiento. De esta manera, el proceso de empleo de esta función será la siguiente:

1. Se establecerá el modo de funcionamiento 'Electronic Gear' con el parámetro *GearRatio* con valor 200. Por otra parte, la función ya estaría lista para su funcionamiento una vez se introduzcan las consignas en la interfaz HMI de Unity Pro.
2. En el caso de que el sistema se descalibre se encenderá el led 'Homing Necesario' de la interfaz HMI de Unity Pro. Para comenzar el proceso de recalibrado se seguirá la siguiente secuencia:
 - a. Activar la casilla Home, en Unity Pro. Con esta casilla activada, se avisa a la función de que va a comenzar un proceso de recalibrado.
 - b. Seleccionar en SoMove el modo de funcionamiento 'Homing' y se seleccionará el modo de homing número 27.
 - c. Tras finalizar el proceso de recalibrado, la casilla se reseteará automáticamente y ya de nuevo se podrá reanudar el funcionamiento que se tenía en el apartado a.

Conexionado

En este caso, se desconectará el sensor de la entrada IN_REF como se tenía en el ejemplo práctico II y pasará a conectarse al canal 6 del servo drive, como se muestra en la siguiente figura:

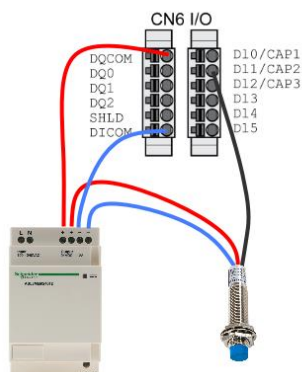


Figura 5-24. Conexionado Canal 6. Ejemplo práctico III.

En las siguientes dos imágenes se ilustran como es el conexionado general de la maqueta tanto en el modo 1 como en el modo 2, respectivamente:

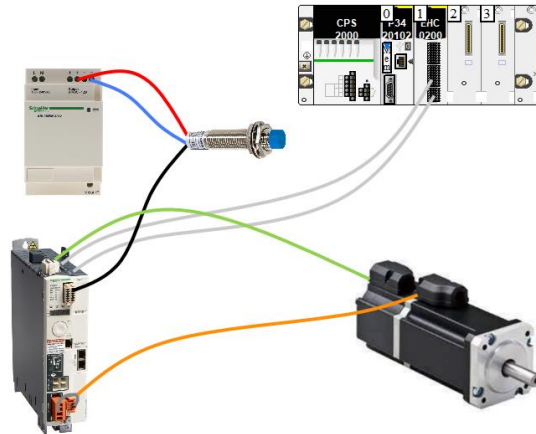


Figura 5-25. Conexionado general. Ejemplo práctico III.

Uso de la librería desde Unity Pro

De nuevo en Unity Pro se encontrará con una interfaz HMI para modificar los parámetros de la función. En concreto, en el ejemplo práctico II esta interfaz es la que se muestra en la siguiente figura:



Figura 5-26. Interfaz HMI. Ejemplo práctico III.

En este caso, se podrán modificar los valores tanto de la carrera, de la frecuencia de la señal de pulsos, del 'Preset Value' o valor de preselección tras un realizar un home, del error de posición admisible y de la referencia de pulsos. Por otro lado se cuenta con unas casillas de verificación para realizar una parada de emergencia, en caso de activar la casilla 'STOP' o de realizar un recalibrado, en caso de activar la casilla 'HomeOk'. En cuanto a las salidas, se puede observar el valor del contador de pulsos en la casilla 'Pulsos' así como un led de aviso de error, en cuyo caso se deberá hacer un recalibrado o 'Home' para volver al funcionamiento.

En el apartado 6 se expondrá el código en LD de la función de este ejemplo.

5.3.5 Ejemplo práctico IV

5.3.5.1 Enunciado

En las prácticas anteriores se habían desarrollado librerías de posicionado de un servomotor, trabajando en todo momento en unidades de pulsos. Aunque el valor de la posición del servomotor puede ser obtenida en todo momento desde SoMove, en esta práctica se introducirán consignas en unidades de posición del servomotor desde Unity Pro, por lo que la posición que aporta SoMove y la posición que aporta librería desarrollada en este apartado deberán coincidir prácticamente en todo momento.

En cuanto al modo de funcionamiento, se seguirá manteniendo la pareja de movimientos de seguidor de referencias y modo homing tal y como se ha realizado en el ejercicio práctico III.

5.3.5.2 Solución adoptada

Para desarrollar esta librería se han necesitado realizar varias pruebas para cuantificar el valor del parámetro ‘Avance’ para diferentes valores del parámetro *GearRatio*. Para ello, se realizaron tres pruebas, que son los tres puntos de la gráfica inferior, consistentes en dar una revolución completa al motor, anotando cuantos pulsos se habían requerido para un valor determinado de *GearRatio* y cuantas posiciones se habían avanzado con la lectura de posición por parte de SoMove. Este proceso método de obtener los datos hace que no sea totalmente exacto pero es bastante parecido al que se obtiene con SoMove.

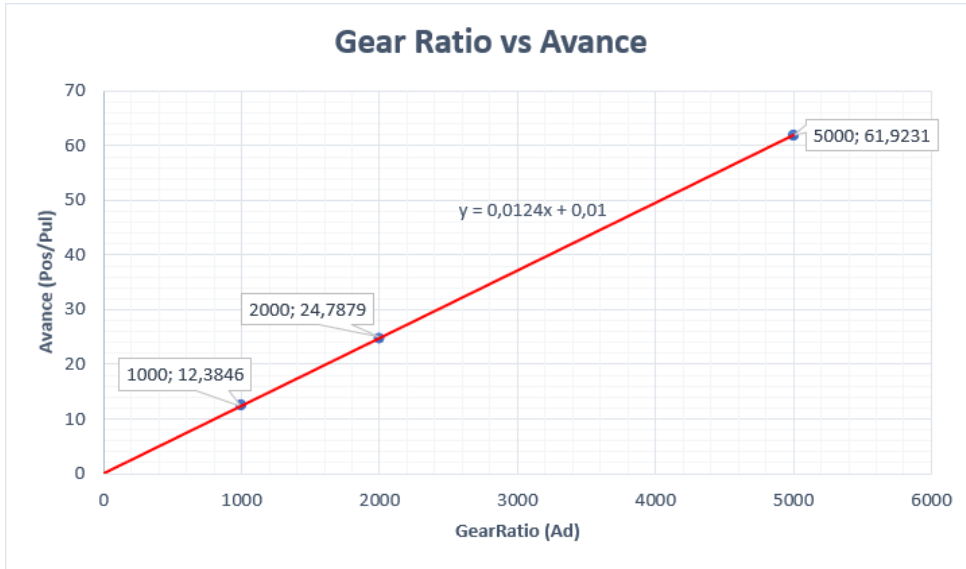


Figura 5-27. GearRatio vs Avance.

De esta forma, al obtener la recta se puede obtener el avance del motor mediante el valor de *GearRatio*.

Configuración en Unity Pro

La configuración de Unity Pro se deberá configurar como en el ejemplo práctico III.

Configuración en SoMove

Desde SoMove es posible configurar qué valor de recorrido sea el recorrido máximo en unidades de posición del servomotor mediante la siguiente lista de parámetros

Configuración de ejes				
Modulo				
MOD_Enable	Modulo On	Activación de Modulo		
MOD_AbsDirection	Shortest Distance	Dirección del movimiento absoluto con Modulo		
MOD_AbsMultiRng	Multiple Ranges Off	Rangos múltiples para movimiento absoluto con Modulo		
MOD_Min	0 [1usr_p]	Posición mínima del rango Modulo	-2147483648 [1usr_p]	2147483647 [1usr_p]
MOD_Max	16500 [1usr_p]	Posición máxima del rango Modulo	-2147483648 [1usr_p]	2147483647 [1usr_p]

Figura 5-28. Lista de parámetros. Configuración de ejes.

De esta forma, modificando los parámetros *MOD_Min* y *MOD_Max* se consigue un rango de funcionamiento de la posición del servomotor. Una vez se sobrepase el valor *MOD_Max* se reiniciará la lectura de posición, volviendo al valor *MOD_Min* y viceversa.

Con respecto al resto de parámetros, estos se deberán configurar como en el ejemplo práctico III.

Conexionado

En cuanto al conexionado, se mantendrá la misma configuración que en el ejercicio práctico I, pues en este no se habían introducido aún el uso del encoder.

Uso de la librería desde Unity Pro

De nuevo en Unity Pro se encuentra con una interfaz HMI para modificar los parámetros de la función. En concreto, en el ejemplo práctico II esta interfaz es la que se muestra en la siguiente figura:

Funcion control de Movimiento	
Carrera	<input type="text"/>
GearRatio	<input type="text"/>
HzMAX [(1/10)Hz]	<input type="text"/>
STOP	<input type="checkbox"/>
HOMEok	<input type="checkbox"/>
ErrorPosAdmisible	<input type="text"/>
ReferenciaPos	<input type="text"/>

Ejemplo Practico IV

Homing Necesario

Posicion

Figura 5-29. Interfaz HMI. Ejemplo práctico IV.

En este caso, se podrán modificar los valores tanto de la carrera, del avance por pulso por parte del motor, de la frecuencia de la señal de pulsos y de la referencia de pulsos. Por otro lado, se cuenta con unas casillas de verificación para realizar una parada de emergencia, en caso de activar la casilla 'STOP' o de realizar un recalibrado, en caso de activar la casilla 'HomeOk'. En cuanto a las salidas, se podrá observar el valor del contador de pulsos en la casilla 'Pulsos'.

6 ANEXOS

6.0 Introducción

En este apartado se procederá a presentar las funciones creadas para los ejemplos prácticos desde el I hasta el IV. De esta manera, se presentará primero una tabla en la que se definirán las entradas, salidas y variables internas que se han necesitado para realizar el programa seguido de la propia función desde Control Expert.

6.1 Ejemplo práctico I

Entradas:

Nombre	Tipo	Comentario
Hz	DINT	Frecuencia en décimas de Hz, de generación de pulsos
STOP	EBOOL	Parada de emergencia
ReferenciaPulso	DINT	Referencia de posición en pulsos

Tabla 6-1. Entradas. Ejemplo práctico I.

Salidas:

Nombre	Tipo	Comentario
Pulsos	DINT	Contador de pulsos emitidos

Tabla 6-2. Salidas. Ejemplo práctico I.

Variables internas:

Nombre	Tipo	Comentario
Posicion	DINT	Posición en pulsos
Aux_RefPosicion	DINT	Referencia de posición en pulsos
Referencia	DINT	Referencia de posición en pulsos
HzUDint	UDINT	Hz en UDint

Tabla 6-3. Variables internas. Ejemplo práctico I.

Código de la función:

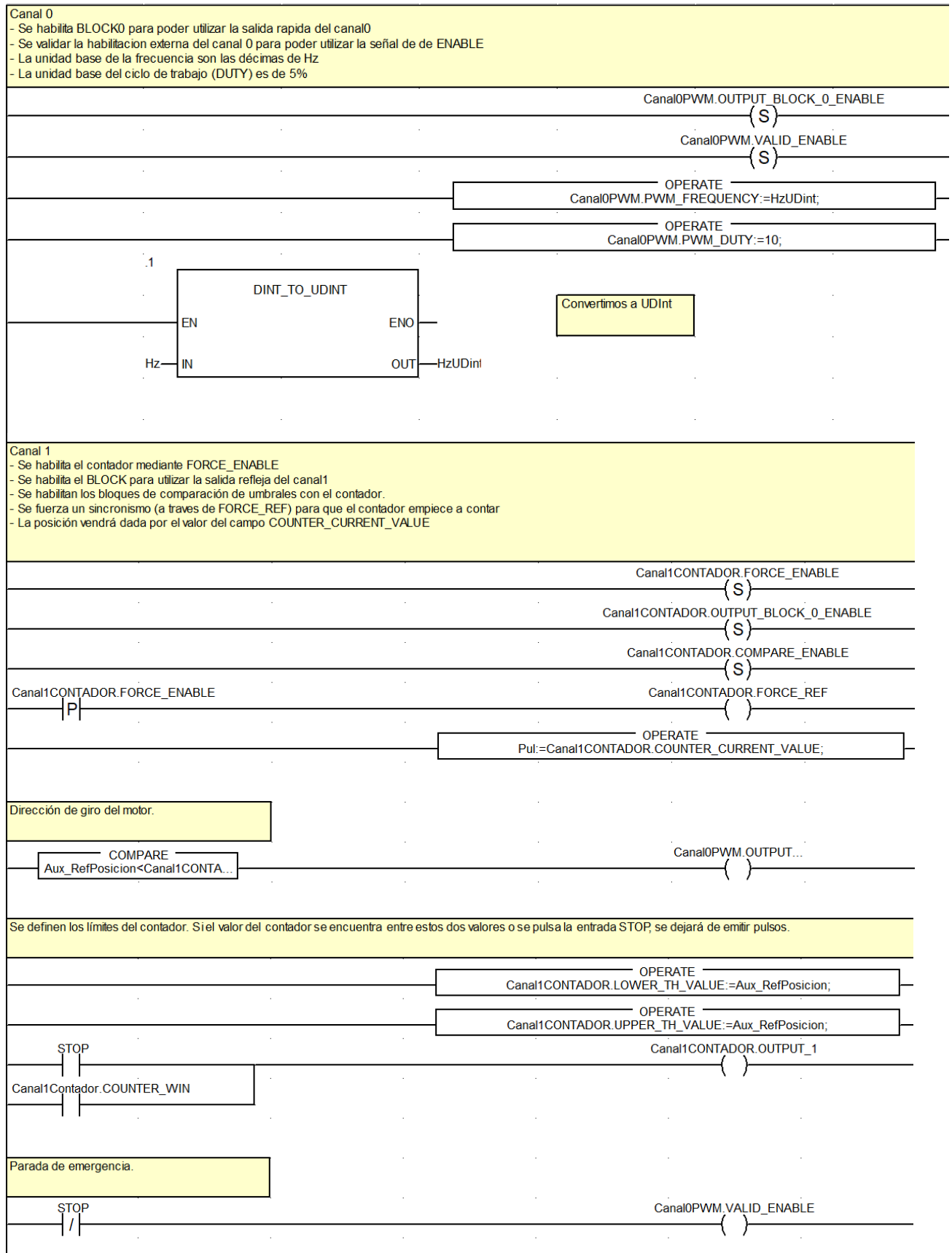


Figura 6-1. Programación de la función. Ejemplo práctico I.

6.2 Ejercicio práctico II

Entradas:

Nombre	Tipo	Comentario
Carrera	DINT	Recorrido ²
Hz	DINT	Frecuencia en décimas de Hz, de generación de pulsos
Preset Value	DINT	Valor cargado tras finalizar el Homing. Se establecerá en 0
STOP	EBOOL	Parada de emergencia
ErrorPosAdmisible	DINT	Error en posición admisible en el modo seguidor de referencias
ReferenciaPulso	DINT	Referencia de posición en pulsos

Tabla 6-4. Entradas. Ejemplo práctico II.

Salidas:

Nombre	Tipo	Comentario
Pulsos	DINT	Contador de pulsos emitidos

Tabla 6-5. Salidas. Ejemplo práctico II.

Entradas/Salidas:

Nombre	Tipo	Comentario
HomeOk	EBOOL	Entrada que inicia el proceso de Homing

Tabla 6-6. Entradas/Salidas. Ejercicio práctico II.

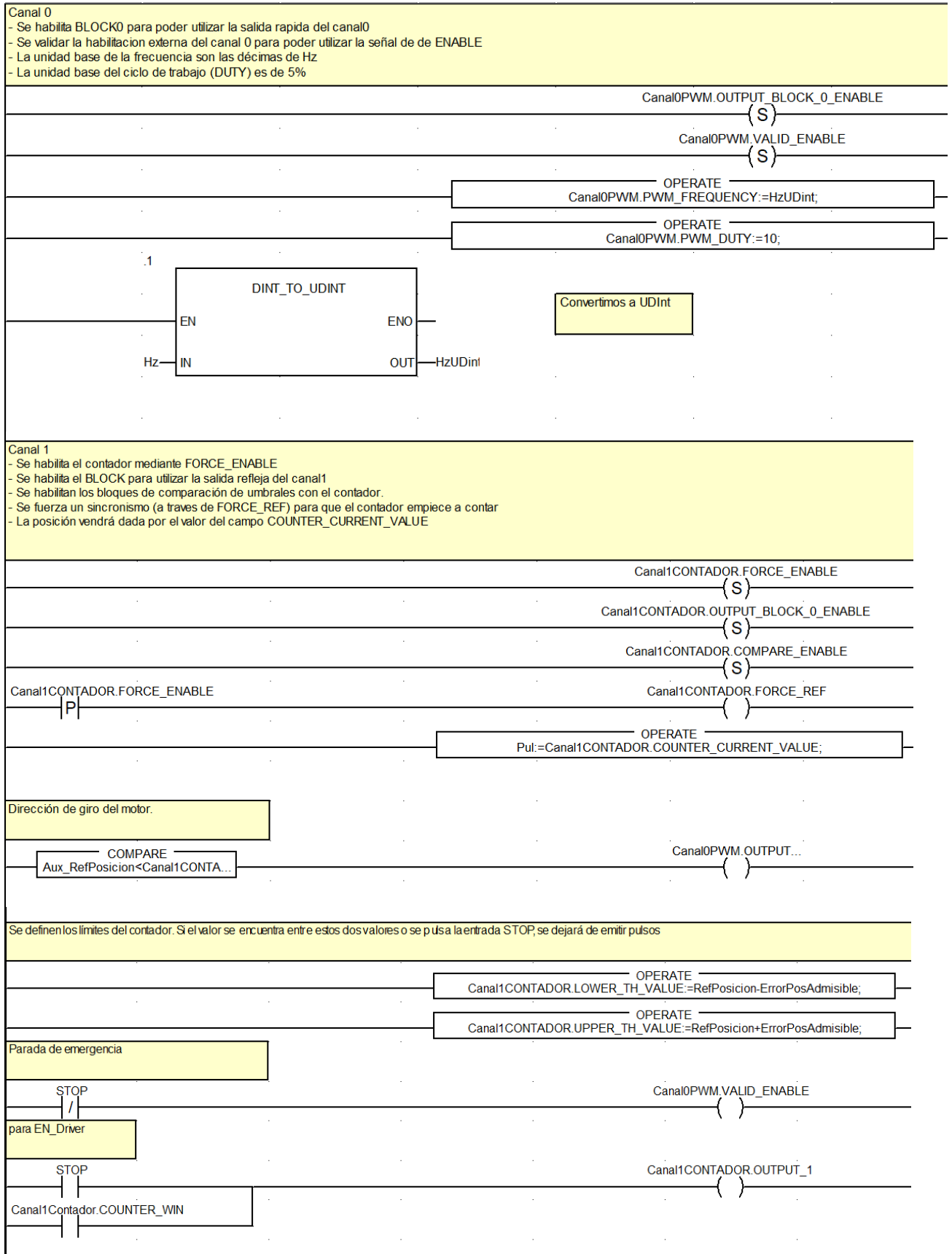
Variables internas:

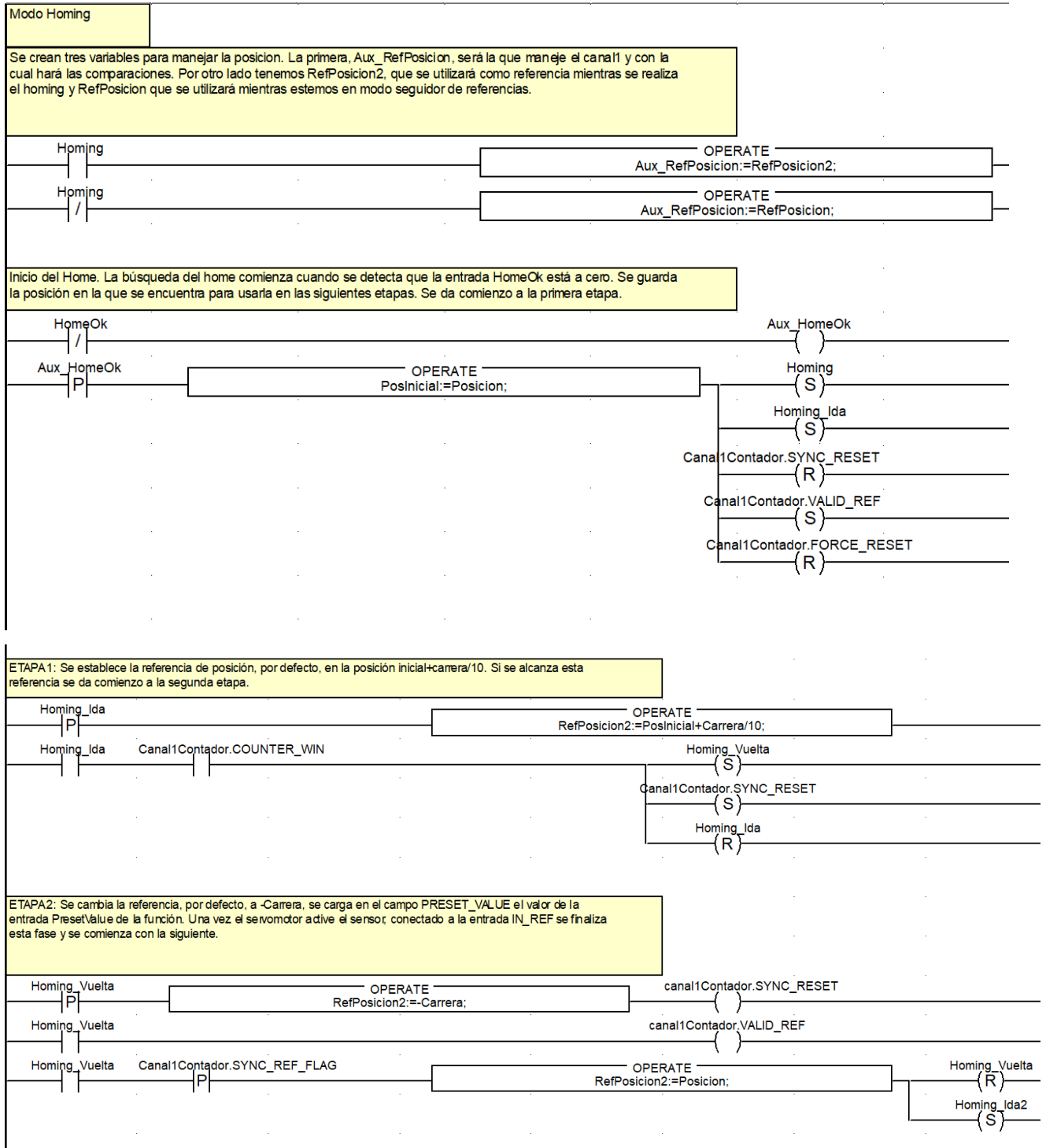
Nombre	Tipo	Comentario
Posicion	DINT	Posición en pulsos
Aux_RefPosicion	DINT	Referencia de posición en pulsos
Aux_HomeOk	EBOOL	Variable auxiliar del modo Homing
RefPosicion	DINT	Referencia de posición auxiliar
RefPosicion2	DINT	Referencia de posición auxiliar
Homing_Ida	EBOOL	Señalización etapa 2
Homing_Vuelta	EBOOL	Señalización etapa 2
Homing_Ida2	EBOOL	Señalización etapa 3
PosInicial	DINT	Posición inicial al hacer Home
HzUDint	UDINT	Frecuencia del tipo UDINT

Tabla 6-7. Variables internas. Ejemplo práctico II.

Código de la función:

² Al no contar con finales de carrera este parámetro carece de sentido en el modo seguidor de referencia. Por otro lado su valor es necesario para el modo Homing.





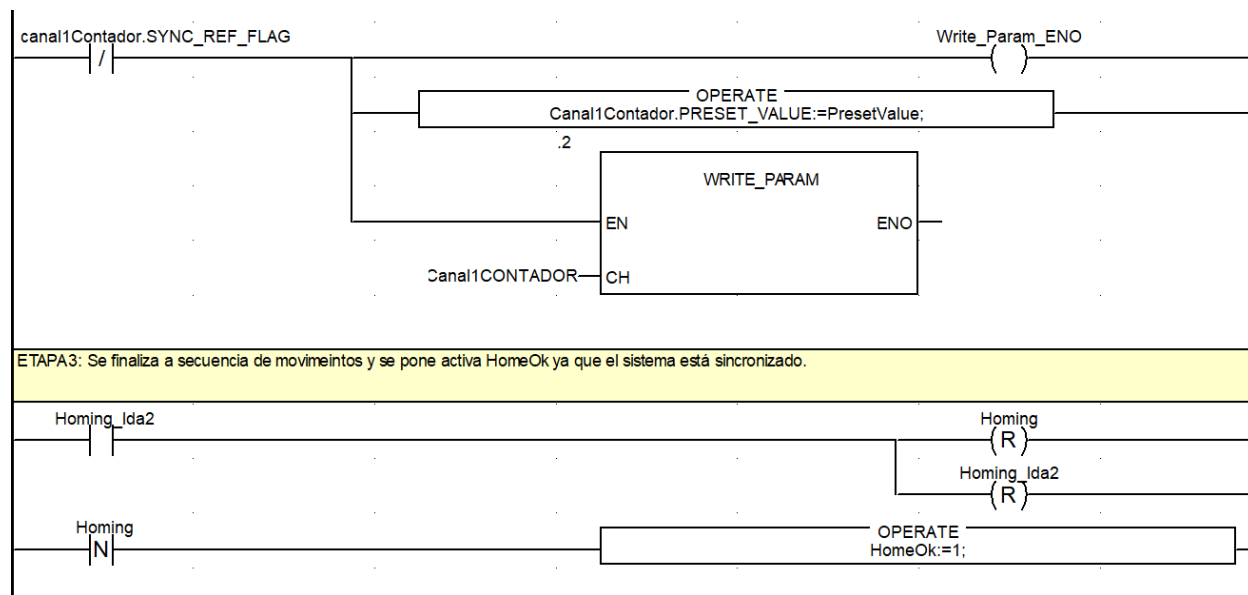


Figura 6-2. Programación de la función. Ejemplo práctico II.

6.3 Ejemplo práctico III

Entradas:

Nombre	Tipo	Comentario
Carrera	DINT	Recorrido ³
Hz	DINT	Frecuencia en décimas de Hz, de generación de pulsos
Preset Value	DINT	Valor cargado tras finalizar el Homing. Se dejará en 0
STOP	EBOOL	Parada de emergencia
ErrorPosAdmisible	DINT	Error en posición admisible en el modo seguidor de referencias
ReferenciaPulso	DINT	Referencia de posición en pulsos

Tabla 6-8. Entradas. Ejemplo práctico III.

Salidas:

Nombre	Tipo	Comentario
Pulsos	DINT	Contador de pulsos emitidos
Homing_Necesario	EBOOL	Led indicador de sistema descalibrado

Tabla 6-9. Salidas. Ejemplo práctico III.

Entradas/Salidas:

Nombre	Tipo	Comentario
HomeOk	EBOOL	Entrada que inicia el proceso de Homing

Tabla 6-10. Entradas/Salidas. Ejercicio práctico III.

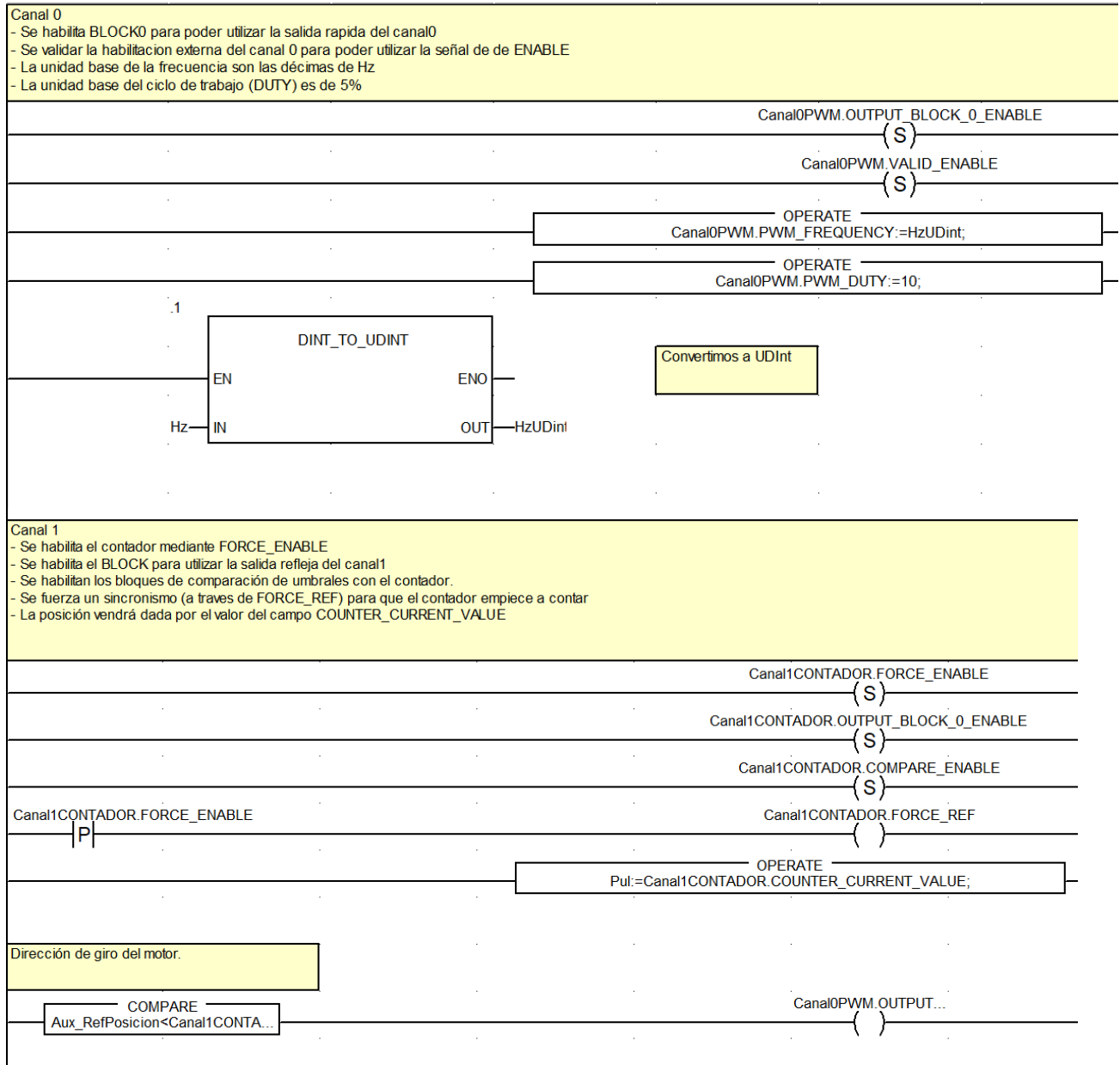
³ Al no contar con finales de carrera este parámetro carece de sentido en el modo seguidor de referencia. Por otro lado su valor es necesario para el modo Homing.

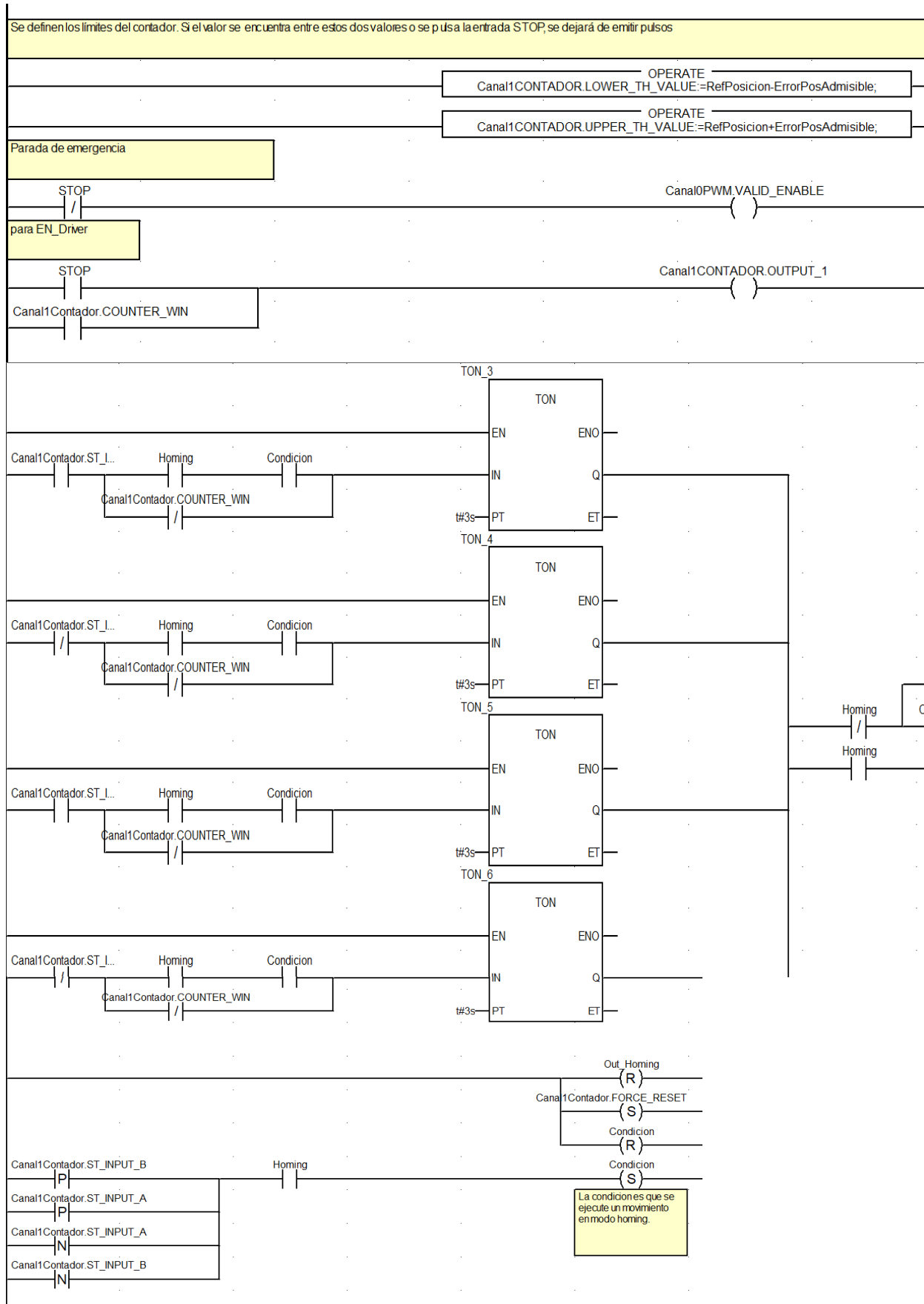
Variables internas:

Nombre	Tipo	Comentario
Reseteo		Señalización de reseteo del contador
HzUDint	UDINT	Frecuencia del tipo UDINT

Tabla 6-11. Variables internas. Ejemplo práctico III.

Código de la función:





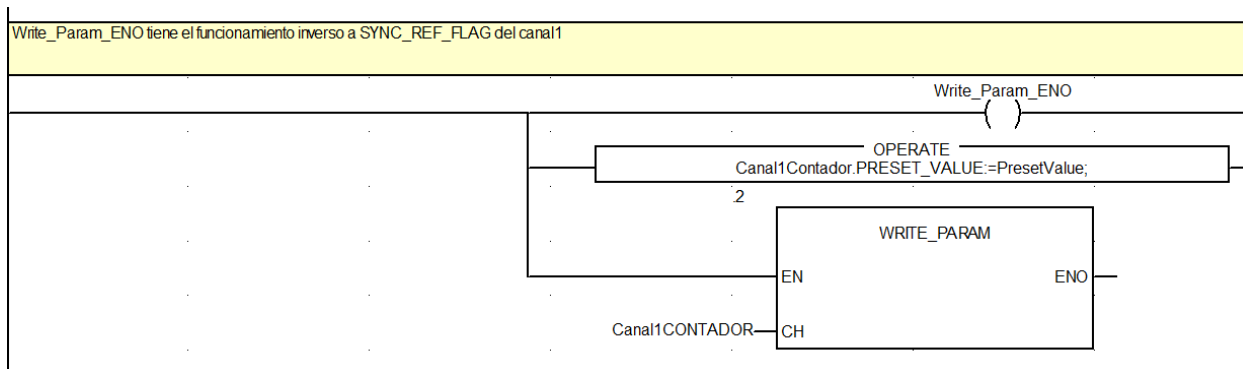


Figura 6-3. Programación de la función. Ejemplo práctico III.

6.4 Ejemplo práctico IV

Entradas:

Nombre	Tipo	Comentario
Carrera	DINT	Recorrido ⁴
Hz	DINT	Frecuencia en décimas de Hz, de generación de pulsos
GearRatio	DINT	Parámetro <i>GearRatio</i> de SoMove
STOP	EBOOL	Parada de emergencia
ErrorPosAdmisible	DINT	Error en posición admisible en el modo seguidor de referencias
ReferenciaPos	DINT	Referencia de posición en pulsos

Tabla 6-12. Entradas. Ejercicio práctico IV.

Salidas:

Nombre	Tipo	Comentario
Pulsos	DINT	Posición del motor
Homing_Necesario	EBOOL	Led indicador de sistema descalibrado

Tabla 6-13. Salidas. Ejercicio práctico IV.

Entradas/Salidas:

Nombre	Tipo	Comentario
HomeOk	EBOOL	Entrada que inicia el proceso de Homing

Tabla 6-14. Entradas/Salidas. Ejercicio práctico IV.

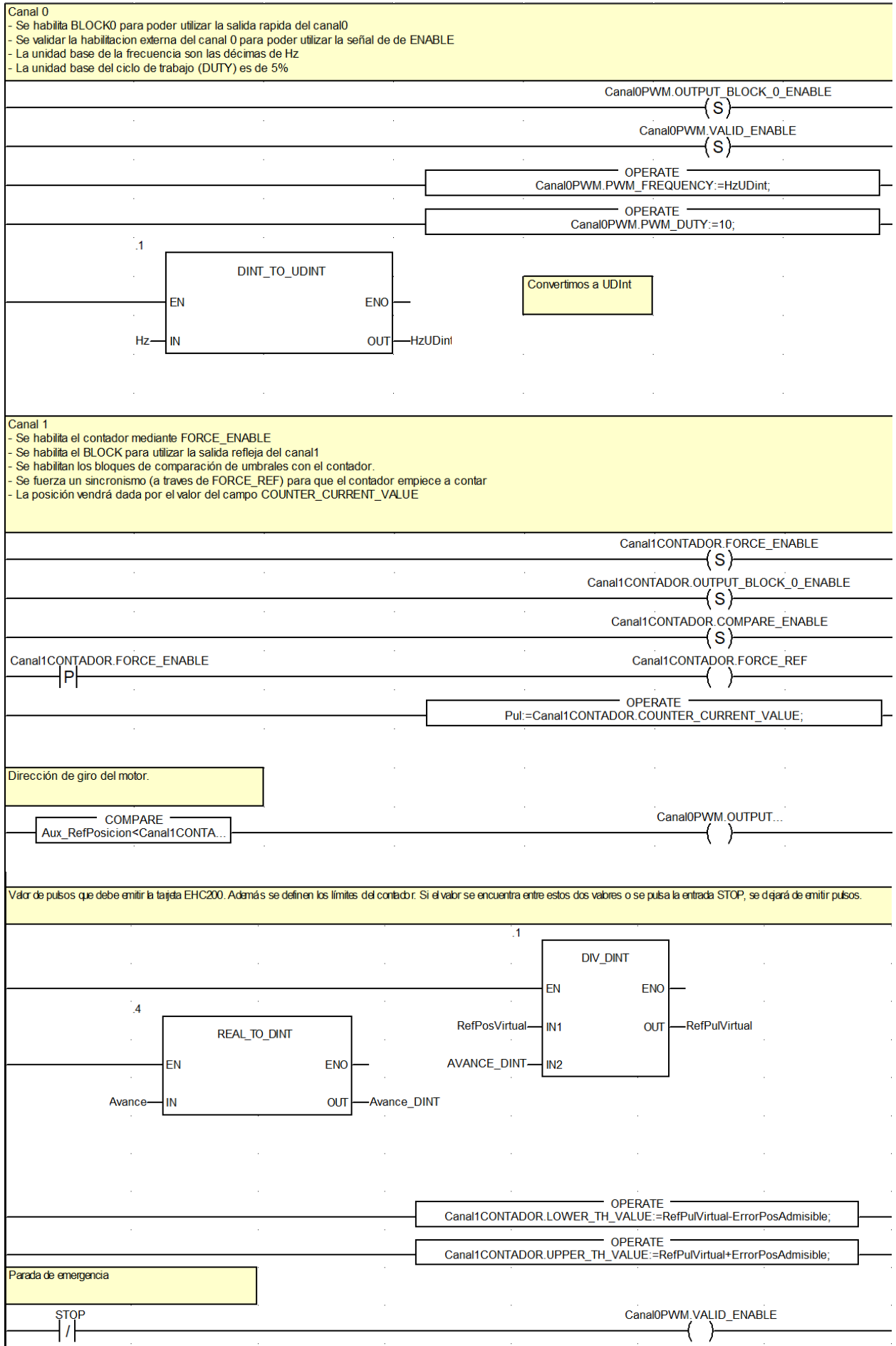
VARIABLES INTERNAS:

Nombre	Tipo	Comentario
Reseteo		Señalización de reseteo del contador
HzUDint	UDINT	Frecuencia del tipo UDINT

Tabla 6-15. Variables privadas. Ejercicio práctico IV.

Código de la función:

⁴ Al no contar con finales de carrera este parámetro carece de sentido en el modo seguidor de referencia. Por otro lado su valor es necesario para el modo Homing.



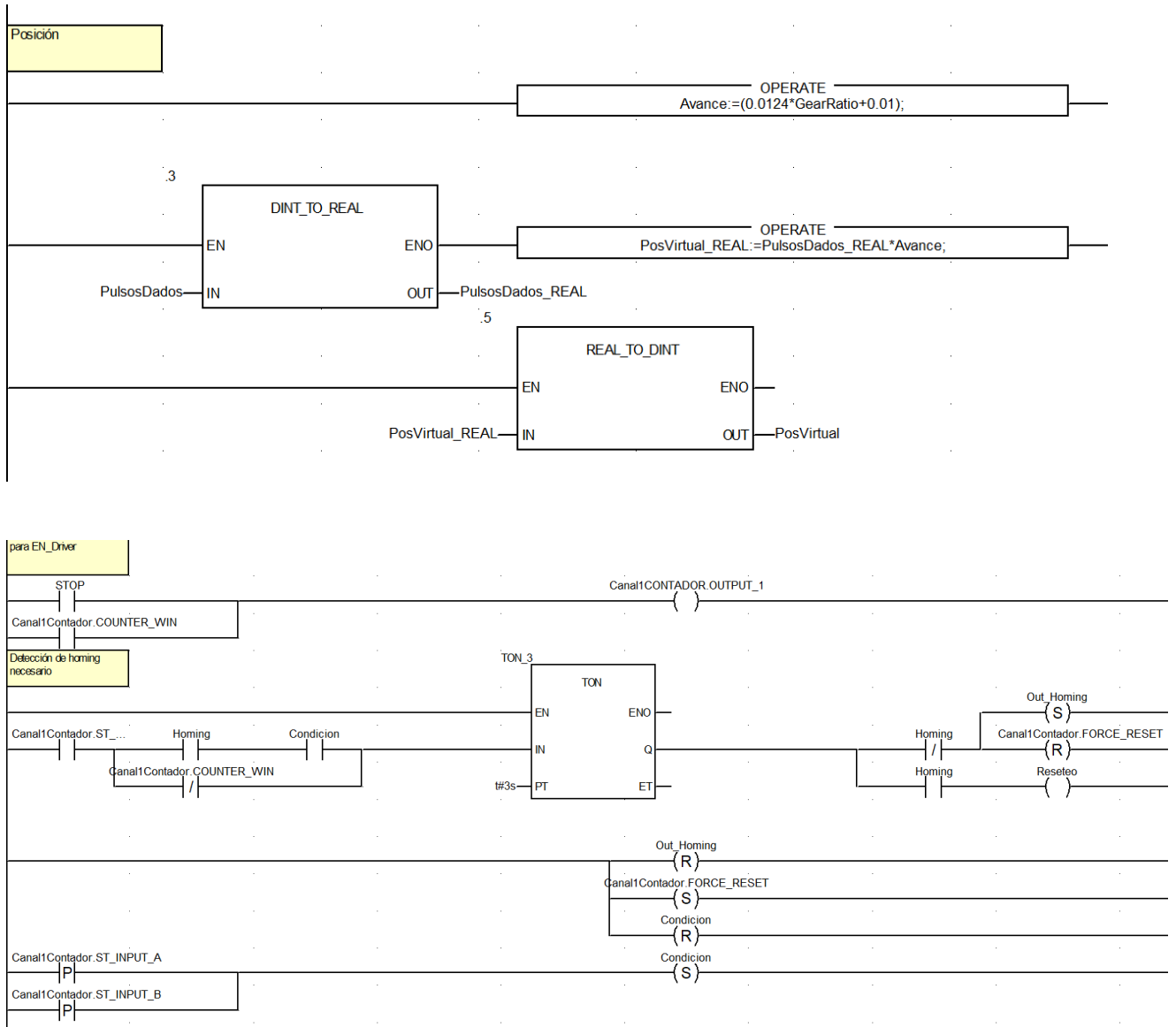


Figura 6-4.Programación de la función. Ejemplo práctico IV.

REFERENCIAS

- [1] Schneider Electric, LXM32M BSH Servomotor, *Manual del motor*, 2017.
- [2] Schneider Electric, LXM32M Servoaccionamiento AC, *Manual del producto*, 2018.
- [3] Schneider Electric, Unity Pro, *Manual de referencia*, 2010.
- [4] Schneider Electric, Servodriver Lexium 32M, *Manual de formación*, 2013.
- [5] Schneider Electric, Módulos Modicon x80_BMXEHC0200, 2022.

