



Universidad
Zaragoza

Trabajo Fin de Grado

Control de movimiento de ejes electromecánicos
mediante un autómatas programable.

Motion control of electromechanical axes by
means of a programmable logic controller.

Autor

Sergio Muñoz Martínez

Director

Ramón Piedrafita Moreno

Grado en Ingeniería electrónica y automática.

Escuela de Ingeniería y Arquitectura de Zaragoza.

Año académico 2022-2023.

Resumen.

El objetivo de este trabajo ha sido el control de ejes electromecánicos, utilizando para ello un autómatas programable o PLC.

Para la realización del trabajo, se han empleado las 2 maquetas siguientes:

1. En primer lugar, la maqueta de un disco de inercia acoplado a un motor LEGO NXT, con el fin de poner en práctica los conocimientos adquiridos sobre el software de control TIA Portal,
2. Y, en segundo lugar, la maqueta de un ascensor, accionada también por otro motor LEGO NXT, al objeto de profundizar en los conocimientos de TIA Portal.

Se comenzó el trabajo buscando información relacionada con las dos maquetas para poder conocer los requisitos técnicos de las mismas, como la tensión y la corriente requerida por el motor o los parámetros técnicos del autómatas.

Una vez adquiridos los conocimientos sobre ambas maquetas, se continuó buscando información sobre TIA Portal en internet.

Y, por último, alcanzados todos los conocimientos necesarios, se pasó a la parte importante de este trabajo: la creación de los proyectos de TIA Portal donde, a petición del Director de este trabajo, se han creado 2 proyectos más con el objetivo de que los alumnos de la asignatura **Automatización Industrial** pudieran utilizarlo durante el curso 2022-2023. Los proyectos creados son los siguientes:

- Proyecto 1. Control de posición angular directamente mediante instrucciones Motion Control. En este proyecto la programación se realizó directamente sobre las instrucciones de la librería Motion Control.
- Proyecto 2. Control de posición angular mediante programación en lenguaje SFC. En este proyecto el control lo realiza un programa en lenguaje SFC que controla las instrucciones Motion_Control.
- Proyecto 3. Control de posición lineal mediante programación en lenguaje SFC. Igual que el proyecto 2 pero usando un eje lineal.
- Proyecto 4. Control de posición lineal mediante generación de código de control con simulink. En este proyecto el código de control se creó en simulink y se ha exportado a TIA Portal.

A continuación, se configuró una pantalla HMI para cada proyecto que permita al usuario realizar el control de forma correcta.

Paralelamente al desarrollo de los proyectos de TIA Portal, y con los conocimientos adquiridos sobre las maquetas, se creó una placa electrónica para adaptar las señales electrónicas entre las maquetas y el autómatas.

Por último, se ha comprobó el correcto funcionamiento del conjunto autómatas, placa adaptadora y maquetas.

Abstract.

The aim of this work was the control of electromechanical axis using a PLC.

The next 2 models have been used to drive the work:

1. Firstly, a model consisting of flywheel couple to a LEGO NXT motor to put into practice the knowledge acquired about the software TIA Portal
2. Secondly, a model consisting of a lift moved by another LEGO NXT motor to consolidate the knowledge of TIA Portal.

To begin with, any information related to both models was sought to learn about the technical requirements like voltage and current required by the NXT motor, or the technical parameters of the PLC.

Once the required knowledge about the two models and the PLC was acquired, the search for further information about TIA Portal continued.

And finally, with the relevant knowledge obtained from this information, the most important part of the work was started - the creation of the projects of TIA Portal. Furthermore, at the request of the Director of this work, two more projects have been created to allow the students of the subject **Automatizaci3n Industrial** to use them for the current academic year 2022/2023.

The projects created are:

- Project 1. Angular position control using Motion Control instructions. In this project the programming was made directly with the Motion Control Instructions.
- Project 2. Angular position control using SFC programming. In this project the programming was made by a SFC program that controls the Motion Control Instructions.
- Project 3. Linear position control using SFC programming. Same as the project 2 but using a linear axis.
- Project 4 Linear position control using Simulink's control code generation. In this project, the control code was made in Simulink and then, exported to TIA Portal.

In the next step, an HMI screen for every project was configured to allow the user to monitor the control properly.

In parallel with the performance of the projects in TIA Portal - and with the knowledge acquired about the models and the PLC- an electronic board was created in order to adapt the electronic signals between the models and the PLC.

Finally, it was verified that the PLC, the electronic board and the model worked correctly combined

Contenido

<i>Resumen</i>	2
<i>Abstract</i>	3
<i>1. Presentación del trabajo</i>	14
1.1. Objetivo del trabajo.....	14
1.2. Bases del trabajo.....	15
1.3. Desarrollo del trabajo.....	15
1.4. Partes de las que consta la memoria.....	17
1.5. Finalidad del trabajo.....	17
<i>2. Elementos que componen el trabajo</i>	18
2.1. Elementos de control y supervisión.....	18
2.1.1. Autómata SIMATIC S7-1512C-1 PN.....	19
2.1.2. HMI TP700 Confort.....	21
2.1.3. Software TIA Portal.....	22
2.2. Elementos de acción.....	22
2.2.1. Motor Lego NXT.....	23
2.2.2. Disco de inercia.....	29
2.2.3. Sensor inductivo.....	30
2.2.4. Ascensor.....	31
<i>3. Desarrollo del programa de control del autómata</i>	34
3.1. Proyecto 2. Control de posición angular mediante programación en SFC.....	35
3.1.1. Creación del proyecto.....	35
3.1.2. Configuración del encóder.....	36
3.1.3. Configuración de los generadores PWM.....	37
3.1.4. Configuración del objeto tecnológico.....	38
3.1.5. Creación del programa de control.....	46
3.2. Proyecto 4. Control de posición lineal mediante generación de código de control con Simulink.....	58
3.2.1. Cambios en la configuración del encóder.....	58
3.2.2. Cambios en el objeto tecnológico.....	59
3.2.3. Cambio en la función Control del motor.....	63
3.2.4. Generación de código de control con Simulink.....	63
<i>4. Configuración de la pantalla de control</i>	78
4.1. Creación y conexión de la pantalla.....	78
4.2. Imágenes de las que consta la pantalla de operador HMI.....	80
4.2.1. Imagen Raiz.....	80
4.2.2. Imagen Intro control.....	81

4.2.3.	Imagen Diagnóstico.	82
4.2.4.	Imagen Control del eje.....	83
4.2.5.	Imagen Posicionamiento absoluto.....	84
4.2.6.	Imagen Posicionamiento relativo.....	85
4.3.	Pantallas del resto de proyectos.	86
5.	<i>Desarrollo de la placa electrónica adaptadora de niveles.</i>	87
5.1.	Consideraciones previas.....	87
5.2.	Elección de componentes.....	88
5.3.	Diseño del circuito.....	90
5.4.	Diseño de la placa y vista 3D.	93
5.5.	Conexión automática – Placa – Maquetas.....	94
6.	<i>Pruebas realizadas</i>	95
6.1.	Prueba de la electrónica.....	95
6.2.	Prueba conjunta de todos los elementos.....	95
7.	<i>Conclusiones.</i>	98
7.1.	Dificultades.....	98
7.2.	Desarrollo del proyecto.....	98
7.3.	Ampliaciones y mejoras futuras.....	98
8.	<i>Glosario.</i>	100
9.	<i>Bibliografía.</i>	101
Apéndice A.	<i>Introducción a TIA Portal.</i>	103
A.1.	Creación del proyecto y configuración inicial.	103
A.2.	Interfaz de trabajo.....	105
A.3.	Añadir entradas y salidas y variables de memoria globales.....	106
A.4.	Configurar los elementos de la CPU	107
A.5.	Objetos tecnológicos.	108
A.6.	Creación del programa.....	109
A.6.1.	Bloque de organización (OB)	110
A.6.2.	Bloque de datos (DB).....	110
A.6.3.	Bloques de Funciones (FB).....	111
A.6.4.	Funciones (FC).....	111
A.7.	Interfaz de trabajo del programa.	112
Apéndice B.	<i>Obtención de los parámetros dinámicos de ambas maquetas.</i>	116
B.1.	Dinámica del motor en la maqueta del disco de inercia.....	116
B.1.1.	Rango de velocidades.	116
B.1.2.	Velocidad nominal.....	118
B.1.3.	Velocidades de aproximación y de referenciado.....	119

B.1.4.	Tiempos de aceleración y deceleración y valores de la aceleración y deceleración nominales.	119
B.1.5.	Tiempos de aceleración y deceleración y valores de la aceleración y deceleración máximos.	120
B.1.6.	Tiempo de deceleración de parada de emergencia y valor de la deceleración. 121	
B.1.7.	Jerk nominal y Jerk máximo y tiempos de redondeo.	122
B.2.	Dinámica del motor en la maqueta del ascensor.	122
B.2.1.	Rango de velocidades.	122
B.2.2.	Velocidad nominal.	125
B.2.3.	Tiempos de aceleración y deceleración y valores de la aceleración y deceleración nominales.	126
B.2.4.	Velocidades de referenciado y aproximación.	127
B.2.5.	Intervalo de alturas excluido para la parada en la planta 1.	128
B.2.6.	Tiempos de aceleración y deceleración y valores de la aceleración y deceleración máximos.	129
B.2.7.	Tiempo de deceleración de parada de emergencia y valor de la deceleración. 130	
B.2.8.	Tiempos de redondeo nominal y máximo.	131
B.3.	Tabla resumen con los valores de la variables.	131
<i>Apéndice C.</i>	<i>Cálculo de los elementos de la placa adaptadora de niveles de tensión.</i>	<i>132</i>
C.1.	Cálculo de resistencias pull-up para el correcto funcionamiento del autómeta.	132
C.2.	Cálculo de resistencias del filtro RC para el correcto funcionamiento del autómeta. 134	
C.3.	Cálculo de resistencias limitadoras de 24 V de las bases de los transistores.	135
C.4.	Cálculo de resistencias limitadoras de los LED.	137
C.5.	Cálculo de condensadores de los filtros RC.	138
<i>Apéndice D.</i>	<i>Esquema completo de la placa adaptadora de niveles de tensión y PCB.</i>	<i>140</i>
<i>Apéndice E.</i>	<i>Diagrama de Gantt.</i>	<i>146</i>
Figura 1.1.	Resumen del proyecto. [4], [10].	14
Figura 1.2.	Interfaz de usuario del programa TIA Portal.	15
Figura 1.3.	Interfaz de usuario del programa DipTrace.	16
Figura 2.1.	Autómeta SIMATIC S7-1512C-1 PN. [10]	18

Figura 2.2. Pantalla HMI TP700 Comfort. [4].....	18
Figura 2.3. Posición de la CPU en el autómeta. [3]	19
Figura 2.4. Posición del módulo de e/s analógicas en el autómeta. [3]	20
Figura 2.5. Posición del módulo de e/s analógicas en el autómeta. [3]	20
Figura 2.6. Maqueta del disco de inercia.	23
Figura 2.7. Maqueta del Ascensor.....	23
Figura 2.8. Motor LEGO NXT. [6]	23
Figura 2.9. Situación interna de los componentes del motor NXT. [6]	23
Figura 2.10. Modelo del motor en lazo abierto. [5].....	24
Figura 2.11. Modelo del motor para simulación. [5].....	24
Figura 2.12. Ubicación del motor eléctrico dentro del motor NXT. [6].....	25
Figura 2.13. Detalle de la caja reductora. [6].....	26
Figura 2.14. Ubicación de la caja reductora. [6].....	26
Figura 2.15. Detalle del encóder del motor NXT. [6].....	26
Figura 2.16. Ubicación del encóder en el motor NXT. [6].....	28
Figura 2.17. Detalle del conector del motor NXT.	28
Figura 2.18. Momentos de inercia de un disco plano. [11].....	29
Figura 2.19. Disco de inercia.	29
Figura 2.20. Sensor inductivo de la maqueta del disco de inercia. [9]	30
Figura 2.21. Placa adaptadora de tensión del sensor inductivo.	30
Figura 2.22. Polea del ascensor.	31
Figura 2.23. Diagrama del sistema mecánico del ascensor.....	31
Figura 2.24. Conector DB25 de la maqueta del ascensor.	33
Figura 2.25. Detalle del conjunto formado por el Final de carrera, el pulsador y el LED	33
Figura 3.1. Maqueta del disco de inercia utilizada en los proyectos 1 y 2.....	34
Figura 3.2. Maqueta del ascensor utilizada en los proyectos 3 y 4.	34
Figura 3.3. Pantalla de creación del nuevo proyecto.....	35
Figura 3.4. Menú de selección de CPU.	35
Figura 3.5. Apartado de activación del contador rápido.	36
Figura 3.6. Apartado de parametrización de la configuración del encóder.	36
Figura 3.7. Apartado de parametrización de la conexión del encóder.....	37
Figura 3.8. Apartado de parametrización de las direcciones de entrada y salida del encóder.	37
Figura 3.9. Apartado de activación del generador PWM.....	37
Figura 3.10. Apartados de configuración del generador PWM.	38
Figura 3.11. Configuración de los parámetros básicos del objeto tecnológico.	39
Figura 3.12. Apartados de selección del accionamiento y del encóder del objeto.	39
Figura 3.13. Apartado de configuración del accionamiento del objeto.....	39
Figura 3.14. Apartado de configuración del encóder del objeto.	40
Figura 3.15. Apartado de configuración mecánica del objeto.	40
Figura 3.16. Apartado de configuración del preajuste de dinámica del objeto tecnológico. ...	41
Figura 3.17. Apartado de configuración del preajuste de dinámica del objeto tecnológico. ...	41
Figura 3.18. Apartado de configuración de los límites de movimiento del objeto tecnológico.	42
Figura 3.19. Apartado de configuración de los límites dinámicos del objeto tecnológico.....	42
Figura 3.20. Apartado de configuración del referenciado activo del objeto tecnológico.	43
Figura 3.21. Apartado de configuración de la vigilancia de posicionamiento del objeto tecnológico.....	44
Figura 3.22. Apartado de configuración del error de seguimiento del objeto tecnológico	44

Figura 3.23. Apartado de configuración de la señal de velocidad cero del objeto tecnológico.	45
Figura 3.24. Apartado de configuración de la regulación de posición del objeto tecnológico.	45
Figura 3.25. Secuencia de generación de la señal PWM. [3]	46
Figura 3.26. Segmento 1 de la función conversor PWM.	47
Figura 3.27. Segmento 2 de la función conversor PWM.	47
Figura 3.28. Segmento 3 de la función conversor PWM.	48
Figura 3.29. Segmento 4 de la función conversor PWM.	48
Figura 3.30. Segmento 5 de la función conversor PWM.	49
Figura 3.31. Segmento 6 de la función conversor PWM.	49
Figura 3.32. Instrucción MC_RESET del bloque de control del eje.	50
Figura 3.33. Instrucción MC_HOME del bloque de control del eje.	50
Figura 3.34. Instrucción MC_POWER del bloque de control del eje.	50
Figura 3.35. Instrucción MC_MOVEJOG del bloque de control del movimiento.	51
Figura 3.36. Instrucción MC_HALT del bloque de control del movimiento.	51
Figura 3.37. Instrucción MC_MOVEABSOLUTE del bloque de control de movimiento.	51
Figura 3.38. Instrucción MC_MOVERELATIVE del bloque de control de movimiento.	52
Figura 3.39. Red SFC de la función control de la maqueta.	52
Figura 3.40. Etapa de reposo y transición de comienzo de la función control de la maqueta.	52
Figura 3.41. Etapa y transición de reinicio de la función control de la maqueta.	53
Figura 3.42. Etapa y transición de habilitación del eje de la función control de la maqueta.	53
Figura 3.43. Etapa y transición de posicionamiento y referenciado de la función control de la maqueta.	54
Figura 3.44. Etapa de espera y ramas de elección de movimiento de la función control de la maqueta.	54
Figura 3.45. Etapa de habilitación y transición del movimiento absoluto de la función control de la maqueta.	55
Figura 3.46. Etapa y transición de realización del movimiento absoluto de la función control de la maqueta.	56
Figura 3.47. Etapa y transición que activan el movimiento manual en la función control de la maqueta.	56
Figura 3.48. Etapa y transición que ejecuta el movimiento manual hacia la izquierda de la función control de la maqueta.	57
Figura 3.49. Etapa y transición de detención del eje en la función control de la maqueta.	57
Figura 3.50. Cambios realizados en la configuración del encóder.	58
Figura 3.51. Cambios realizados en la configuración de las unidades de medida del objeto tecnológico.	59
Figura 3.52. Cambios en la velocidad de giro de referencia del objeto tecnológico.	59
Figura 3.53. Cambios en la mecánica del accionamiento del objeto tecnológico.	59
Figura 3.54. Cambios realizados en los valores nominales de la dinámica del objeto tecnológico.	60
Figura 3.55. Cambios en la detección de los límites de posición del objeto tecnológico.	61
Figura 3.56. Cambios en los límites dinámicos del objeto tecnológico.	61
Figura 3.57. Cambios en el referenciado del objeto tecnológico.	62
Figura 3.58. Cambio en la vigilancia de posicionamiento del objeto tecnológico.	62
Figura 3.59. Cambio en la ventana de velocidad cero del objeto tecnológico.	63
Figura 3.60. Proyecto creado para exportar el bloque de Asignación de variables.	64
Figura 3.61. Proyecto creado para exportar el bloque de Asignación del SetPoint.	64
Figura 3.62. StateFlow de la función Asignación de variables.	65

Figura 3.63. Estado Asignación_Variables del StateFlow de la función Asignación de variables.....	66
Figura 3.64. Estados que reinician las llamadas del StateFlow de la función Asignación de variables.....	67
Figura 3.65. StateFlow de la función Asignación del SetPoint.	67
Figura 3.66. Estado Espera del StateFlow de la función Asignación del SetPoint.	68
Figura 3.67. Estados Reinicio y Habilitación del StateFlow de la función Asignación del SetPoint.....	68
Figura 3.68. Estado Posicionamiento del StateFlow de la función Asignación del SetPoint.	69
Figura 3.69. Estado Selección_Modo del StateFlow de la función Asignación del SetPoint.	69
Figura 3.70. Estado PosicionadoMemoria del StateFlow de la función Asignación del SetPoint.	69
Figura 3.71. Estado SubirMemoria del StateFlow de la función Asignación del SetPoint.....	70
Figura 3.72. Estado ParadoPlantaMemoria del StateFlow de la función Asignación del SetPoint.....	70
Figura 3.73. Estado BajarMemoria del StateFlow de la función Asignación del SetPoint.	71
Figura 3.74. Estado ParadaPlantaMemoria1 del StateFlow de la función Asignación del SetPoint.....	71
Figura 3.75. Estado PosicionadoNoMemoria del StateFlow de la función Asignación del SetPoint.....	72
Figura 3.76. Estado MovimientoNoMemoria del StateFlow de la función Asignación del SetPoint.....	72
Figura 3.77. Estado ParadoPlantaNoMemoria del StateFlow de la función Asignación del SetPoint.....	73
Figura 3.78. Lista de fabricantes y lenguajes de PLCs admitidos por MATLAB.	73
Figura 3.79. Menú de configuración de los parámetros de exportación de MATLAB.	74
Figura 3.80. Ubicación de la opción de importar fuentes externas en TIA Portal.	75
Figura 3.81. Ubicación de la opción generar bloques a partir de fuente en TIA Portal.	75
Figura 3.82. Cambios realizados en la programación tras ser importada a TIA Portal.	76
Figura 3.83. Conexión entre ambas funciones en la función control del ascensor en TIA Portal.	77
Figura 4.1. Asistente de configuración de la pantalla HMI.	79
Figura 4.2. Pantalla de introducción al control.	80
Figura 4.3. Pantalla de elección de movimiento.....	81
Figura 4.4. Pantalla de diagnóstico del sistema.....	82
Figura 4.5. Pantalla de control del eje.....	83
Figura 4.6. Pantalla de control del movimiento absoluto.....	84
Figura 4.7. Pantalla de control del movimiento relativo.....	85
Figura 4.8. Pantalla de control del proyecto 3.....	86
Figura 4.9. Pantalla de control del proyecto 2.....	86
Figura 4.10. Pantalla de control del proyecto 4.....	86
Figura 5.1. Pinout del chip ULN2003AD. [8].....	88
Figura 5.2. Pinout del chip LB1836M. [17].....	88
Figura 5.3. Pinout del chip SN74AC14D. [18].	88
Figura 5.4. Pico de intensidad de arranque del motor NXT.	88
Figura 5.5. Intensidad nominal del motor NXT.....	88
Figura 5.6. Aplicación típica de la fuente conmutada LM78_09-2.0R. [19].....	89
Figura 5.7. Pinouts de los chips LM7805CV y L7810ACD2T. [20].....	89
Figura 5.8. Conector DB9. [22].....	89
Figura 5.9. Conector DB25. [21].....	89

Figura 5.10. Conector lego NXT RJ12. [7].....	89
Figura 5.11. Esquema de adaptación de las señales entre el autómatas y el motor NXT.....	90
Figura 5.12. Esquema de adaptación entre las señales del encóder y el autómatas.	90
Figura 5.13. Esquema de adaptación entre las señales de los pulsadores y el autómatas.	91
Figura 5.14. Estructura que adapta las señales entre los finales de carrera y el autómatas.....	92
Figura 5.15. Estructura que adapta las señales entre el autómatas y los LEDs.....	92
Figura 5.16. Visión 3D de la placa electrónica.	93
Figura 5.17. Esquema de conexión entre la placa y las maquetas.	94
Figura 6.1. Detalle del chip 74AC14D sin los pines 6 y 12.	95
Figura 6.2. Grafica que muestra la referencia de posición, la posición actual, la referencia de velocidad y la velocidad actual.	96
Figura 6.3. Gráfica donde se observa la velocidad actual y la referencia de velocidad juntas.	96
Figura A.1. Pantalla inicial de TIA Portal.	103
Figura A.2. Pantalla de creación de proyecto.....	104
Figura A.3. Pantalla inicial del proyecto.....	104
Figura A.4. Interfaz de trabajo de TIA Portal.	105
Figura A.5. División de la interfaz de trabajo de TIA Portal.....	105
Figura A.6. Ubicación de la vista de variables en TIA Portal.....	107
Figura A.7. Ubicación de los dispositivos en TIA Portal.	108
Figura A.8. Ubicación de los objetos tecnológicos en TIA Portal.	109
Figura A.9. Tipos de bloques de programación.	109
Figura A.10. Ubicación de los bloques de programa en TIA Portal.....	112
Figura A.11. Interfaz de trabajo de una función en TIA Portal.	112
Figura A.12. Interfaz de trabajo de una función en TIA Portal dividida en secciones.	113
Figura A.13. Lista de variables de una función en TIA Portal.	113
Figura A.14. Utilidades básicas para crear una función en TIA Portal.....	114
Figura A.15. Segmento de trabajo de una función en TIA Portal.	114
Figura A.16. Sección de trabajo de una función en TIA portal.	115
Figura A.17. Librería de instrucciones de TIA Portal.	115
Figura C.1. Circuito simplificado de la estructura que adapta las señales entre los finales de carrera y el autómatas.....	133
Figura C.2. Esquema del circuito que compone cada par Darlington del integrado ULN2003AD. [8].....	136
Tabla 2.1. Tabla de parámetros de la CPU del autómatas. [3].....	19
Tabla 2.2. Características de la pantalla HMI. [4]	22
Tabla 2.3. Parámetros y características del motor NXT. [5].....	24
Tabla 2.4. Relación de transmisión de los engranajes de la caja reductora. [6]	26
Tabla 2.5. Parte eléctrica de la maqueta del ascensor.....	32
Tabla B.1. Mediciones de frecuencia para la maqueta del disco.	117
Tabla B.2. Mediciones de frecuencia para la maqueta del ascensor.	124
Tabla B.3 Tabla resumen con todos los valores calculados de ambas maquetas.	132
Ecuación 2.1. Ecuación que determina el número de vueltas que gira el motor.	27
Ecuación 2.2. Ecuación que determina el número de vueltas que gira el encóder.	27

Ecuación 2.3. Ecuación que determina la resolución en pulsos del encóder.....	27
Ecuación 2.4. Ecuación que determina la resolución en grados del encóder.	28
Ecuación 2.5. Ecuación que determina el valor del momento de inercia del disco.	29
Ecuación 2.6. Ecuación que determina la longitud de la polea.	32
Ecuación 6.1. Ecuación que determina el número de lecturas por vuelta del encóder.	97
Ecuación B.1. Ecuación que determina la velocidad angular del disco de inercia en vueltas por minuto.....	116
Ecuación B.2. Ecuación que determina la velocidad angular del disco de inercia en grados por segundo.	117
Ecuación B.3. Ecuación que determina el valor numérico de la velocidad máxima del disco de inercia en vueltas por minuto.	117
Ecuación B.4. Ecuación que determina el valor numérico de la velocidad máxima del disco de inercia en grados por segundo.	117
Ecuación B.5. Ecuación que determina el valor numérico de la velocidad mínima del disco de inercia en grados por segundo.	118
Ecuación B.6. Ecuación que determina la velocidad angular nominal del disco de inercia en grados por segundo.	118
Ecuación B.7. Ecuación que determina el valor de la velocidad angular del disco de inercia en grados por segundo.	118
Ecuación B.8. Ecuación que determina la velocidad angular nominal del disco de inercia en vueltas por minuto.....	118
Ecuación B.9. Ecuación que determina el valor de la velocidad angular del disco de inercia en vueltas por minuto.....	118
Ecuación B.10. Ecuación que determina la aceleración angular nominal del disco de inercia.	119
Ecuación B.11. Ecuación que determina la deceleración angular nominal del disco de inercia.	119
Ecuación B.12. Ecuación que determina el valor de la aceleración angular nominal del disco de inercia.	120
Ecuación B.13. Ecuación que determina el valor de la deceleración angular nominal del disco de inercia.	120
Ecuación B.14. Ecuación que determina la aceleración angular máxima del disco de inercia.	120
Ecuación B.15. Ecuación que determina la deceleración angular máxima del disco de inercia.	121
Ecuación B.16. Ecuación que determina el valor de la aceleración angular máxima del disco de inercia.	121
Ecuación B.17. Ecuación que determina el valor de la deceleración angular máxima del disco de inercia.	121
Ecuación B.18. Ecuación que determina la deceleración angular de emergencia del disco de inercia.....	121
Ecuación B.19. Ecuación que determina el valor de la deceleración angular de emergencia del disco de inercia.	122
Ecuación B.20. Ecuación que determina la velocidad angular del motor del ascensor en vueltas por segundo.....	123
Ecuación B.21. Ecuación que determina la velocidad angular del motor del ascensor en vueltas por minuto.....	123
Ecuación B.22. Ecuación que determina la velocidad del ascensor en milímetros por segundo.	123
Ecuación B.23. Ecuación que determina la velocidad angular del motor del ascensor en vueltas por minuto.....	123

Ecuación B.24. Ecuación que determina la velocidad angular del ascensor en milímetros por segundo.	124
Ecuación B.25. Ecuación que determina el valor numérico de la velocidad angular máxima del motor del ascensor en vueltas por minuto.	124
Ecuación B.26. Ecuación que determina el valor numérico de la velocidad máxima ascensor en milímetros por segundo.	124
Ecuación B.27. Ecuación que determina el valor numérico de la velocidad mínima del ascensor en milímetros por segundo.	125
Ecuación B.28. Ecuación que determina la velocidad nominal del ascensor en milímetros por segundo.	125
Ecuación B.29. Ecuación que determina el valor de la velocidad nominal del ascensor en milímetros por segundo.	125
Ecuación B.30. Ecuación que determina la velocidad angular nominal del motor del ascensor en vueltas por minuto.	125
Ecuación B.31. Ecuación que determina el valor de la velocidad angular nominal del motor del ascensor en vueltas por minuto.	126
Ecuación B.32. Ecuación que determina el valor de la aceleración nominal del ascensor.	126
Ecuación B.33. Ecuación que determina la aceleración nominal del ascensor.	126
Ecuación B.34. Ecuación que determina el valor de la aceleración nominal del ascensor.	126
Ecuación B.35. Ecuación que determina el valor de la deceleración nominal del ascensor.	126
Ecuación B.36. Ecuación que determina la distancia recorrida por el ascensor durante la deceleración.	127
Ecuación B.37. Ecuación que determina el tiempo de deceleración del ascensor.	127
Ecuación B.38. Ecuación simplificada que determina la distancia recorrida por el ascensor durante la deceleración.	128
Ecuación B.39. Ecuación que determina el valor de la velocidad de referenciado máxima del ascensor.	128
Ecuación B.40. Ecuación que determina el valor de la velocidad de referenciado máxima del ascensor.	128
Ecuación B.41. Ecuación que determina el valor de la distancia de deceleración del ascensor.	129
Ecuación B.42. Ecuación que determina la altura mínima del intervalo de parada restringido del ascensor.	129
Ecuación B.43. Ecuación que determina la altura máxima del intervalo de parada restringido del ascensor.	129
Ecuación B.44. Ecuación que determina el valor de la altura mínima del intervalo de parada restringido del ascensor.	129
Ecuación B.45. Ecuación que determina el valor de la altura máxima del intervalo de parada restringido del ascensor.	129
Ecuación B.46. Ecuación que determina la aceleración máxima del ascensor.	130
Ecuación B.47. Ecuación que determina la deceleración máxima del ascensor.	130
Ecuación B.48. Ecuación que determina el valor de la aceleración máxima del ascensor.	130
Ecuación B.49. Ecuación que determina el valor de la deceleración máxima del ascensor.	130
Ecuación B.50. Ecuación que determina la deceleración de emergencia del ascensor.	131
Ecuación B.51. Ecuación que determina el valor de la deceleración de emergencia del ascensor.	131
Ecuación C.1. Ecuación que determina el balance de corrientes en el nodo.	133
Ecuación C.2. Ecuación que determina la resistencia máxima de pull-up.	133
Ecuación C.3. Ecuación que determina el valor máximo de la resistencia de pull-up.	133
Ecuación C.4. Ecuación que determina la potencia máxima que puede disipar la resistencia de pull-up.	134

Ecuación C.5. Ecuación que determina la mínima resistencia de pull-up.	134
Ecuación C.6. Ecuación que determina el valor mínimo de la resistencia de pull-up.....	134
Ecuación C.7. Ecuación que determina el equilibrio de corrientes en el nodo.	135
Ecuación C.8. Ecuación que determina la máxima resistencia del filtro RC.	135
Ecuación C.9. Ecuación que determina el valor máximo de la resistencia del filtro RC.	135
Ecuación C.10. Ecuación que determina la corriente de base del par Darlington.	136
Ecuación C.11. Ecuación que determina el valor de la corriente de base del par Darlington.	136
Ecuación C.12. Ecuación que determina la nueva resistencia de base del par Darlington. .	136
Ecuación C.13. Ecuación que determina el valor de la nueva resistencia de base del par Darlington.....	137
Ecuación C.14. Ecuación que determina la corriente que pasa por el LED.	137
Ecuación C.15. Ecuación que determina el valor de la corriente que pasa por el LED verde alimentado a 3,3 V.	137
Ecuación C.16. Ecuación que determina el valor de la corriente que pasa por el LED rojo alimentado a 3,3 V.	138
Ecuación C.17. Ecuación que determina la nueva resistencia limitadora de los LED.	138
Ecuación C.18. Ecuación que determina el valor de la nueva resistencia limitadora para el LED verde.....	138
Ecuación C.19. Ecuación que determina el valor de la nueva resistencia limitadora para el LED rojo.	138
Ecuación C.20. Ecuación que determina la constante de tiempo de un sistema de primer orden.	139
Ecuación C.21. Ecuación que determina el valor de la constante de tiempo del filtro RC....	139
Ecuación C.22. Ecuación que determina la constante del sistema de un filtro RC.	139
Ecuación C.23. Ecuación que determina el valor del condensador de un filtro RC.....	139
Ecuación C.24. Ecuación que determina el valor del condensador del filtro RC.	139

1. Presentación del trabajo.

En este capítulo se van a presentar los principales aspectos del trabajo como son el objetivo, las bases, el desarrollo, las partes de la memoria y la finalidad.

1.1. Objetivo del trabajo.

El objetivo de este trabajo ha sido el control de movimiento de ejes electromecánicos mediante un autómata programable. Además, para que un operador pueda controlar el sistema, también se ha incorporado una pantalla HMI. Los ejes electromecánicos que se han controlado son:

- En primer lugar, una maqueta formada por un motor lego NXT al que se le ha acoplado un disco de inercia.
- En segundo lugar, una maqueta de un ascensor, accionada, también, por un motor lego NXT.

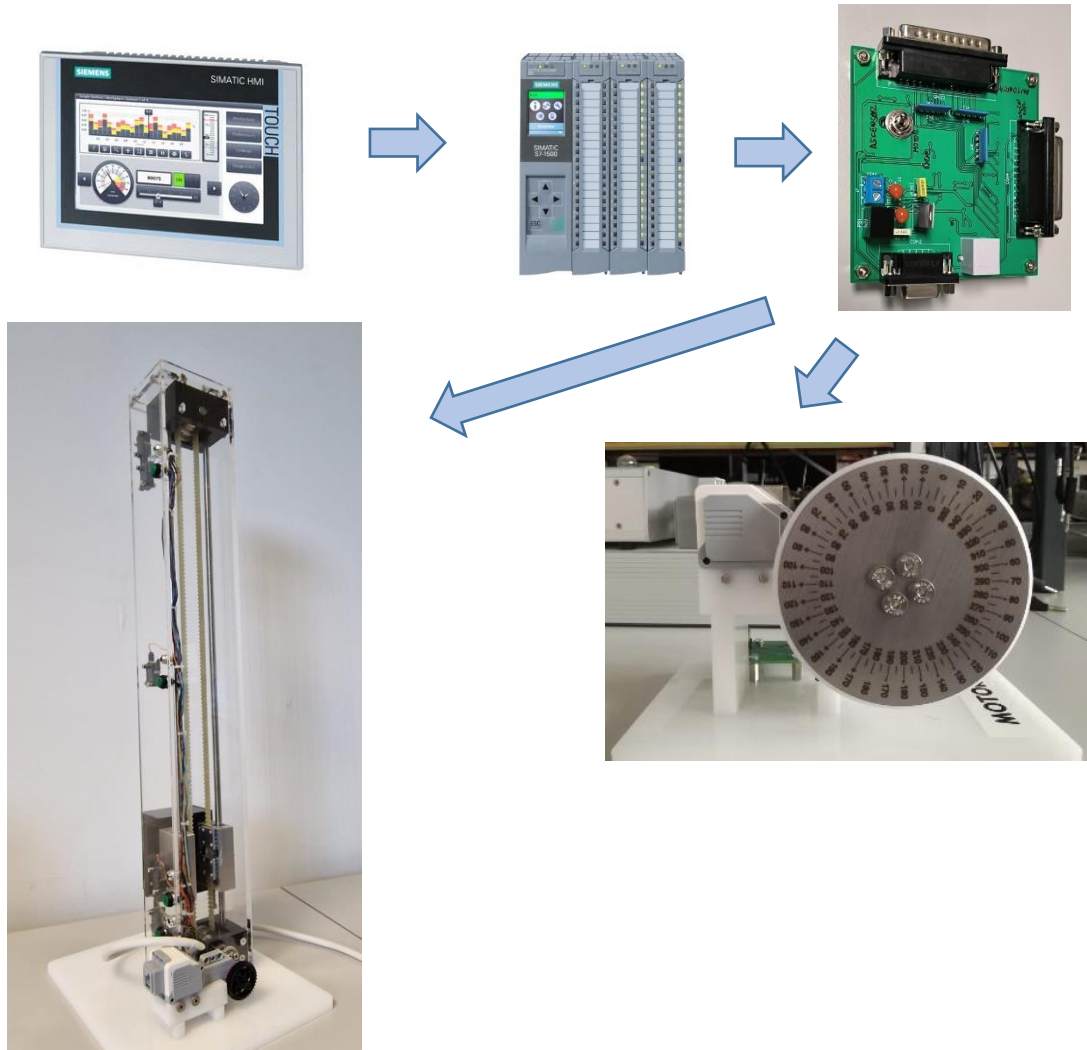


Figura 1.1. Resumen del proyecto. [4], [10]

1.2. Bases del trabajo.

Este trabajo ha tomado como base otros 2 trabajos de fin de grado:

- En primer lugar, está el TFG realizado por Alan Malo [1] cuyo trabajo era muy similar a este y consistía, entre otras cosas, en utilizar este mismo autómatas para controlar la maqueta del disco de inercia, pero, debido a la pandemia de coronavirus, este trabajo tuvo que ser realizado solo en simulación sin llegar a ser probado sobre el autómatas real.
- En segundo lugar, está el TFG de Miguel Sesma [2] que consistía en el control de la maqueta del ascensor mediante Simulink y del que se ha utilizado parte del control de la maqueta, adaptándolo a los elementos disponibles en TIA Portal.

1.3. Desarrollo del trabajo.

Una vez que se ha tenido claro el objetivo del trabajo se ha elaborado un plan para la realización de dicho trabajo.

Inicialmente, se ha buscado la información necesaria sobre los elementos utilizados en el trabajo como son los parámetros técnicos del autómatas o los de las maquetas.

Cuando se ha reunido la información necesaria, se ha analizado dicha información en busca de problemas que pudieran surgir a la hora de desarrollar el programa de control o la placa electrónica de adaptación de niveles de tensión.

Una vez identificados y resueltos los problemas, se han adquirido los conocimientos necesarios en el manejo del software que permite programar el autómatas. En este caso se ha usado un autómatas del fabricante alemán Siemens y las herramientas de desarrollo que proporciona, como es TIA Portal.

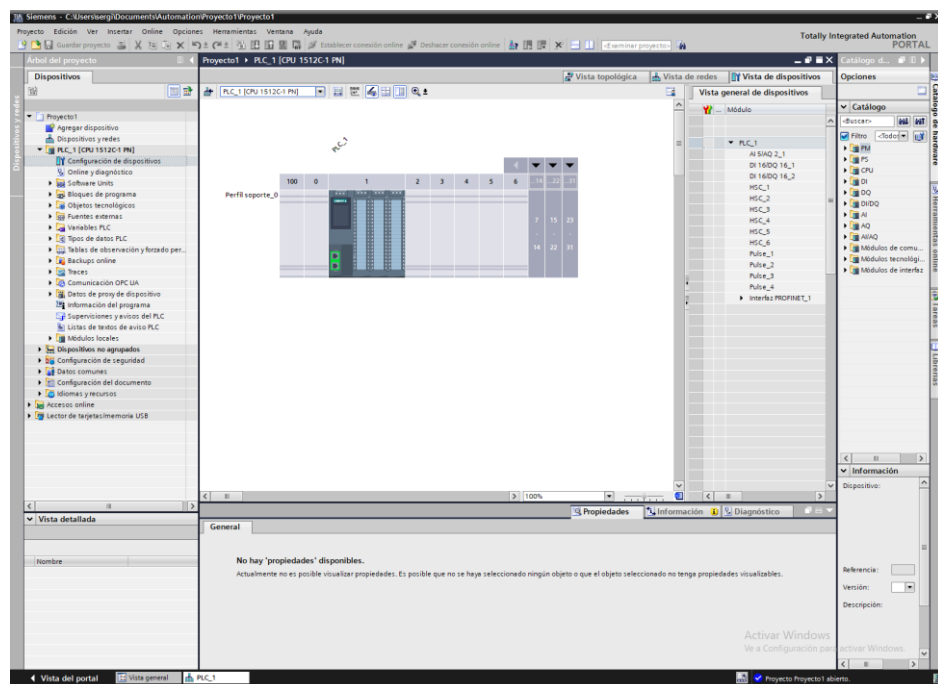


Figura 1.2. Interfaz de usuario del programa TIA Portal.

Tras haber adquirido los conocimientos necesarios para el correcto uso del software TIA Portal, se ha procedido a diseñar el programa de control del autómeta y la pantalla del operador que permite al usuario realizar el control de las maquetas. Posteriormente se ha verificado el programa haciendo uso del simulador que TIA Portal lleva incorporado.

Paralelamente al desarrollo del programa en TIA Portal, se ha diseñado la placa de adaptación de niveles de tensión entre el autómeta y las maquetas. Para ello, se ha dispuesto de varios programas que permiten realizar esta acción y que han sido utilizados alguna vez en el desarrollo de los estudios como son EasyEDA, DipTrace o EagleCAD. En este caso se ha usado DipTrace.

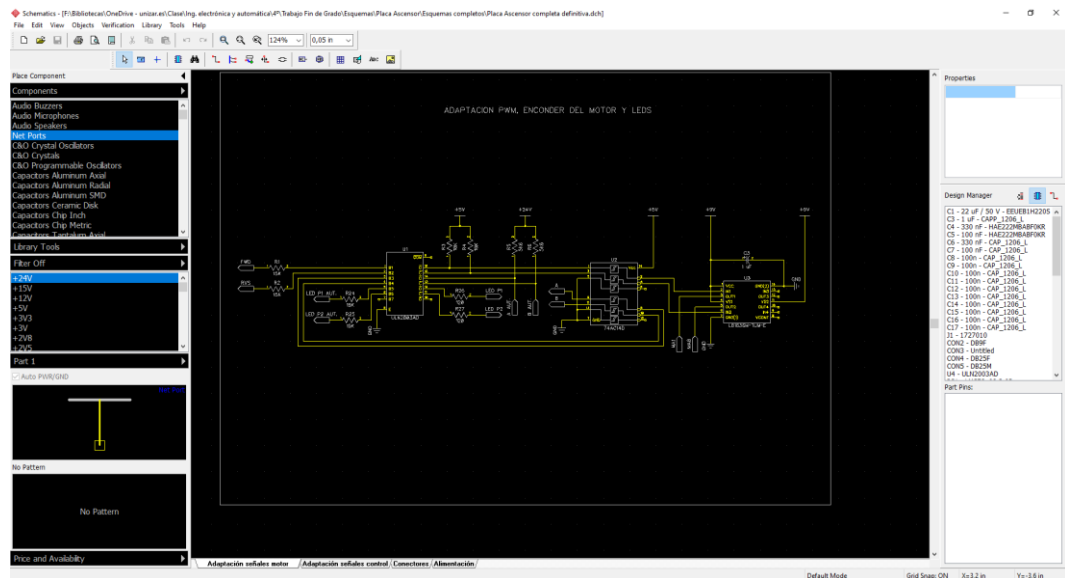


Figura 1.3. Interfaz de usuario del programa DipTrace.

Una vez que la placa está diseñada, se ha verificado en el laboratorio y en conjunto con las maquetas y el autómeta, el correcto funcionamiento. Cuando el resultado ha sido satisfactorio, se ha mandado fabricar dicha placa.

Tras haber recibido la placa fabricada, se ha realizado una última comprobación por si hubiera habido algún fallo en la fase de fabricación, externa a este proyecto.

Por último, se ha elaborado la presente memoria sobre el trabajo realizado.

En resumen, las etapas de desarrollo de este proyecto han sido:

1. Etapa 1: Búsqueda de información.
2. Etapa 2: Identificación y resolución de problemas que puedan surgir.
3. Etapa 3: Adquisición de conocimientos sobre el software de programación del autómeta.
4. Etapa 4: Desarrollo del programa del autómeta.
5. Etapa 5: Diseño de la pantalla del operador.
6. Etapa 6: Desarrollo de la placa electrónica.
7. Etapa 7: Verificación completa del trabajo.
8. Etapa 8: Elaboración de la memoria.

1.4. Partes de las que consta la memoria.

La memoria del trabajo consta de las siguientes partes:

- Capítulo 1 – Presentación del trabajo. Presentación de los elementos básicos del trabajo y de la memoria como el objetivo, el desarrollo o la finalidad del trabajo.
- Capítulo 2 – Elementos que componen el trabajo. Presentación y caracterización de los elementos de los que se compone el trabajo.
- Capítulo 3 – Desarrollo del programa de control del autómeta. Desarrollo de los programas de control para cada maqueta, incluyendo la creación del proyecto, la configuración de la CPU del autómeta o la escritura del programa de control.
- Capítulo 4 – Desarrollo de la pantalla de operador. Desarrollo de la pantalla que permite al operador controlar ambas maquetas sin necesidad de un ordenador.
- Capítulo 5 – Desarrollo de la placa electrónica de adaptación de niveles. Desarrollo de la placa que permite adaptar los niveles de tensión entre las maquetas y el autómeta y viceversa.
- Capítulo 6 – Pruebas realizadas. Realización de diversas pruebas para garantizar el correcto control de las maquetas y en cuyo caso analizar los errores posibles en el control.
- Capítulo 7 – Conclusiones. Presentación de las conclusiones y posibilidad de desarrollo futuro del presente trabajo.

1.5. Finalidad del trabajo.

Este trabajo ha servido como aplicación práctica de los conocimientos adquiridos durante la etapa formativa, especialmente los conocimientos adquiridos, tanto de automatización industrial y sistemas automáticos como de electrónica analógica. Además, dado que hasta ahora se utilizaban autómetas fabricados por Schneider Electric este proyecto también ha servido para ampliar los conocimientos sobre autómetas industriales de distintos fabricantes.

2. Elementos que componen el trabajo.

En este capítulo se van a presentar y caracterizar los elementos que componen el trabajo, es decir, el autómata, el software de programación y las maquetas. Este capítulo se corresponde con las etapas 1, 2 y 3 del desarrollo del trabajo.

Este capítulo podría dividirse en dos partes:

- Control: formada por el autómata, la pantalla de operador y el software de control.
- Acción: formada por las maquetas.

2.1. Elementos de control y supervisión.

Los elementos de control utilizados en este trabajo se encuentran a continuación:

Con respecto al autómata se ha utilizado un SIMATIC S7-1512C-1 PN, un autómata compacto para aplicaciones medias con módulos de entradas y salidas integrados. Este autómata requiere una versión de TIA PORTAL mínima v15.

Con respecto a la pantalla del operador se ha utilizado una pantalla HMI TP700 CONFORT, una pantalla panorámica TFT de 7" de tamaño. Esta pantalla requiere una versión de winCC / TIA Portal mínima v11.

Debido a los requerimientos de versión necesarios, tanto por la CPU como la pantalla, se ha utilizado la versión de TIA PORTAL v15.1.



Figura 2.1. Autómata SIMATIC S7-1512C-1 PN. [10]



Figura 2.2. Pantalla HMI TP700 Comfort. [4]

2.1.1. Autómata SIMATIC S7-1512C-1 PN.

Como se ha mencionado antes, el autómata incluye la CPU y los módulos de entrada y salida. En esta sección se van a presentar las características principales tanto de la CPU como de los módulos de entradas y salidas. A parte, se van a presentar las características de funciones que incluye el autómata y que son necesarias para realizar el control del sistema como son los contadores rápidos HSC, que han sido necesarios para el encoder o generadores de impulsos PTO / PWM que han sido necesarios para controlar la velocidad del motor.

2.1.1.1. CPU.



Figura 2.3. Posición de la CPU en el autómata. [3]

En la Tabla 2.1, se muestran las características más relevantes de la CPU.

CARACTERÍSTICAS MÁS RELEVANTES DE LA CPU	
Modelo	CPU 1512C-1 PN (6ES7512-1CK01-0AB0)
Tensión de alimentación	24 V DC (19,2 V – 28,8 V DC)
Intensidad de alimentación	0,8 A
Módulo de e/s digitales incluido	SÍ
Módulo de e/s analógicas incluido	SÍ
Versión mínima de STEP 7	V14
Memoria de trabajo para programa	250 KB
Memoria de trabajo para datos	1 MB
Memoria de carga enchufable	32 GB
Bloques CPU	DB, FB, FC, OB
Número máximo de bloques CPU	2000
Contadores rápidos (HSC)	SÍ, 6
Generadores de impulsos (PTO/PWM)	SÍ, 4
Interfaces soportadas	Profinet, Ethernet
Objetos tecnológicos soportados	Motion control, PIDs, Contador rápido

Tabla 2.1. Tabla de parámetros de la CPU del autómata. [3]

2.1.1.2. Módulos de entradas y salidas.

Aparte de la CPU, el autómata contiene 3 módulos de entradas y salidas, los cuales son:

- 2 módulos de entradas y salidas digitales.
- 1 módulo de entradas y salidas analógicas.



Figura 2.5. Posición del módulo de e/s analógicas en el autómata. [3]



Figura 2.4. Posición del módulo de e/s analógicas en el autómata. [3]

Los parámetros más relevantes de los módulos de entradas y salidas digitales, que son los únicos utilizados en el trabajo son los siguientes:

- 32 entradas y 32 salidas.
- El rango de tensiones de la entrada es entre 11 y 30 V para el valor de "1" siendo 24 V la tensión nominal y de entre -3 y 5 V para el valor "0" siendo 0 V el valor nominal.
- Intensidad típica de entrada de 2,5 mA.
- Tensión mínima de salida para valor "1" de 23,2 V siendo 24 V el valor nominal.
- Tensión máxima de salida para valor "0" de 1 V siendo 0 V el valor nominal.
- La frecuencia de salida máxima es de 100 kHz con salida High Speed y de 10 kHz con salida normal.
- La corriente máxima por salida es de 0,5 A con salida normal y de 0,1 A con salida High Speed.

2.1.1.3. Otras características.

En esta sección se incluyen elementos que incorpora el autómata y que han sido necesarios para realizar el control del motor como son los contadores rápidos HSC o los generadores de impulsos PTO / PWM.

2.1.1.3.1. Contadores rápidos HSC.

La utilidad que presentan los contadores rápidos o HSC es realizar conteos de variables a una velocidad superior a la que los bloques OB (que depende de la velocidad de la CPU) pueden ejecutarse. En el caso de conteos a velocidades menores existen otras herramientas que no involucran estos contadores. Típicamente, estos contadores rápidos son utilizados para entradas de encóder, como el que se ha usado en el trabajo para poder medir la velocidad del motor.

Estos contadores tienen las siguientes características:

- 2 entradas digitales por cada contador rápido para posibles funciones extra.
- 1 salida digital para reacción rápida a la lectura del contador.
- Rango de conteo de 32 bits.
- Alarmas de diagnóstico y proceso parametrizables.
- Soportan reparametrización mientras el autómatas está en estado RUN.

Y soporta los siguientes encóder:

1. Encóder incrementales de 24 V.
2. Encóder de impulsos de 24 V con y sin señal de sentido
3. Encóder de impulsos de 24 V respectivamente para impulsos adelante / atrás.

2.1.1.3.2. Generadores de impulsos PWM / PTO

El autómatas también contiene 4 generadores de impulsos cuya función principal es generar un tren de pulsos, ya sea PWM o PTO.

Los modos de funcionamiento admitidos son:

1. Modulación PWM
2. Salida de frecuencia
3. PTO, los cuales pueden ser:
 - 3.1. Impulso A y sentido B.
 - 3.2. Contador ascendente A y descendente B.
 - 3.3. A y B desfasados, tanto simple como cuádruple.

2.1.2. HMI TP700 Confort.

Las características más relevantes de la pantalla HMI se pueden encontrar en la Tabla 2.2.

CARACTERÍSTICAS MÁS RELEVANTES DE LA PANTALLA HMI	
Modelo	TP700 Confort (6AV2124-0GC01-0AX0)
Tamaño	7"
Resolución	800x480 píxeles
Número de colores	16.777.216
Tensión de alimentación	24 V DC (19,2 V a 28,8 V DC)
Corriente de alimentación	0,5 A
Pantalla táctil	Sí
Teclado en pantalla	Sí

Memoria Flash	Sí
Memoria RAM	Sí
Memoria disponible para usuario	12 MB
Reloj de hardware (Reloj de tiempo real)	Sí
Reloj por software	Sí
Zumbador de aviso	Sí
N.º de interfaces ethernet	1 (2 puertos / switch)
N.º de interfaces RS 485	1 combinada con RS 422
N.º de interfaces USB	2 (USB 2.0)
N.º de slots para tarjeta SD	2
Protección IP frontal	IP65
Protección IP lado posterior	IP20

Tabla 2.2. Características de la pantalla HMI. [4]

2.1.3. Software TIA Portal.

Dentro del paquete TIA Portal, en su versión 15.1, se incluyen los siguientes paquetes:

- Simatic Step 7 Basic. Este es el paquete principal con el que se ha trabajado y es el que permite configurar y programar el autómatas.
- Simatic WINCC Basic. Este paquete permite la configuración y utilización de paneles de operador HMI y sistemas de visualización.
- PLCSIM. Este paquete contiene el entorno de simulación que permite simular el funcionamiento del autómatas.

En el Apéndice A puede consultarse información básica acerca de este programa.

2.2. Elementos de acción.

Una vez conocida la parte de control, se va a presentar la parte sobre la que se ha realizado el control, es decir, las maquetas.

Por un lado, está la primera maqueta, el motor con el disco de inercia acoplado, que está formada por:

- Motor Lego NXT.
- Disco acoplado al eje rotacional del bloque que hace de disco de inercia.
- Sensor de proximidad inductivo para generar la señal N del encoder.

Por otro lado, está la segunda maqueta, el ascensor, que está formada por:

- Motor Lego NXT.
- Ascensor.

Como ambas maquetas comparten el motor Lego NXT, se va a comenzar la caracterización por este elemento.

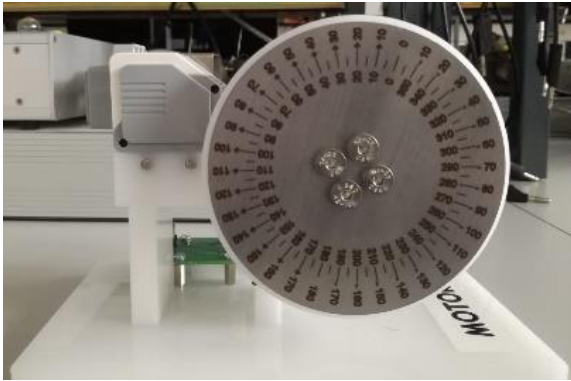


Figura 2.6. Maqueta del disco de inercia.

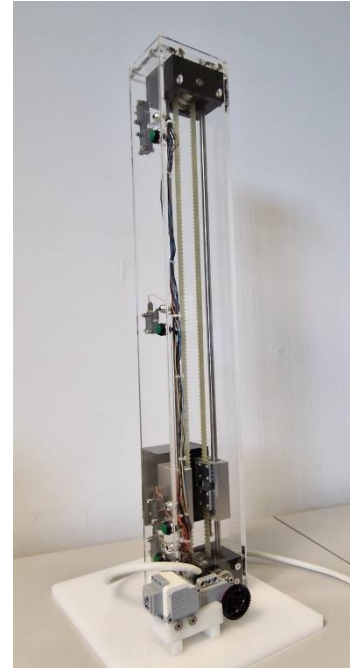


Figura 2.7. Maqueta del Ascensor.

2.2.1. Motor Lego NXT.

El motor Lego NXT forma parte de la colección Mindstorms de la marca LEGO y constituye, junto al autómata, una de las partes más importantes de este trabajo ya que es el elemento sobre el que se va a realizar el control. Sus principales parámetros y características pueden consultarse en la Tabla 2.3.



Figura 2.8. Motor LEGO NXT. [6]

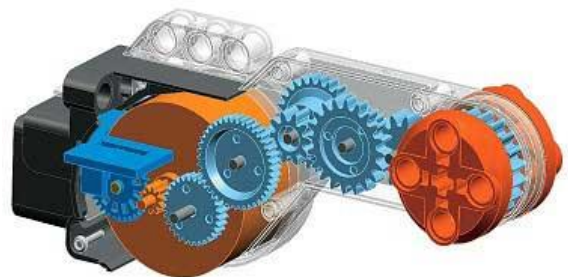


Figura 2.9. Situación interna de los componentes del motor NXT. [6]

PARÁMETROS Y CARACTERÍSTICAS DEL MOTOR NXT	
PARÁMETROS EXPERIMENTALES	
Resistencia R_a	2,27 Ω
Inductancia L_a	0,0047 H
Constante de torque del motor K_i	0,25 N * m/A
Constante contraelectromotriz K_b	0,25 V * s / rad
Coeficiente de fricción viscosa B_m	0,00131 N * m / s
Constante mecánica de tiempo τ_m	0,081 s
Inercia del motor y tren de engranajes J_m	0,00233 kg * m ²
CARACTERÍSTICAS	
Peso	80 g
Tensión nominal	9 V DC
Velocidad de rotación sin carga	170 rpm
Corriente sin carga	60 mA
Torque o inercia	50 N * cm

Tabla 2.3. Parámetros y características del motor NXT. [5]

Otro dato que es de vital importancia conocer cuando se realiza el control de posición de un motor, aunque no en este trabajo dado que el propio autómatas se encarga de identificar el sistema y realizar el control, es conocer el modelo electromecánico del motor. En la Figura 2.10 se puede apreciar el modelo en lazo abierto de un motor DC y en la Figura 2.11 el modelo de simulación, también en lazo abierto, realizados, ambos modelos, utilizando la herramienta SIMULINK de MATLAB.

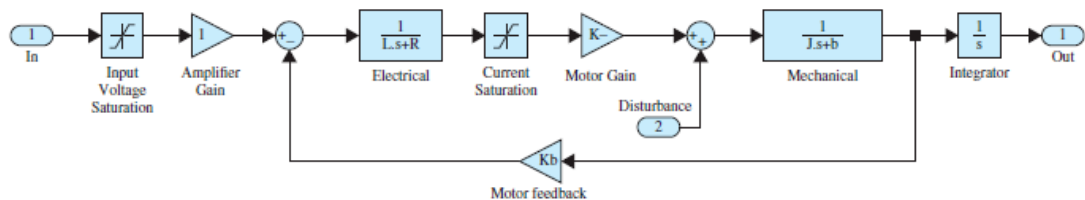


Figura 2.10. Modelo del motor en lazo abierto. [5]

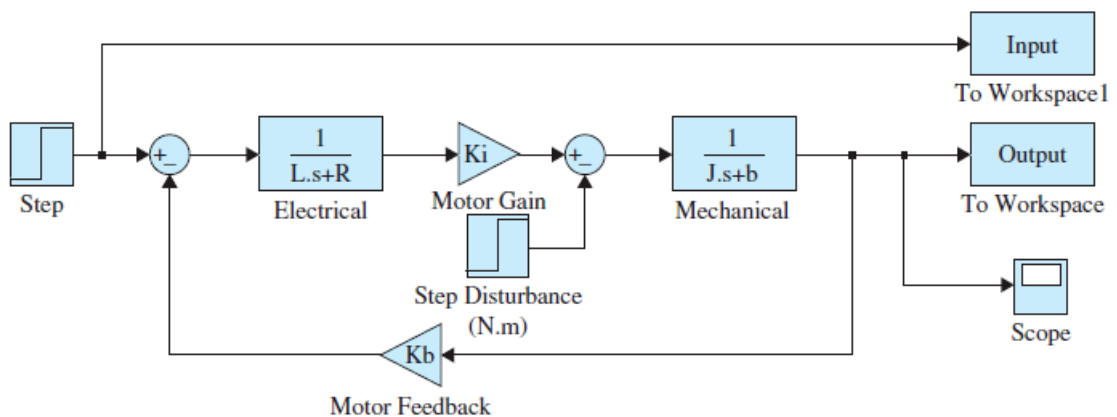


Figura 2.11. Modelo del motor para simulación. [5]

Como se aprecia, en ambos modelos, el sistema base está formado por los siguientes bloques:

1. Bloque de inercia eléctrica.
2. Ganancia del motor.
3. Un sumatorio al que se le añaden las perturbaciones que puede tener el motor que, en el caso del modelo de simulación, vienen dadas por un escalón.
4. Bloque de inercia mecánica.
5. Realimentación con la constante contraelectromotriz.

A parte, en el caso del modelo de simulación se añaden unos bloques de visualización, scope, para ver la evolución y la respuesta del sistema frente al escalón unitario de entrada, mientras que, en el modelo real, se añaden limitaciones de voltaje y corriente, así como un integrador para obtener la posición, al contrario que en el modelo de simulación que lo que se obtiene es la velocidad del motor.

Una vez conocidas las características generales del motor Lego NXT, es importante conocer la composición interna de dicho bloque. El motor Lego NXT viene constituido por los siguientes elementos:

- Motor eléctrico.
- Encoder.
- Caja reductora.
- Conector de alimentación del motor NXT.

2.2.1.1. Motor eléctrico.

El motor es un simple motor de corriente continua de 9 V a cuyo eje se encuentra acoplada una rueda dentada con 10 dientes.

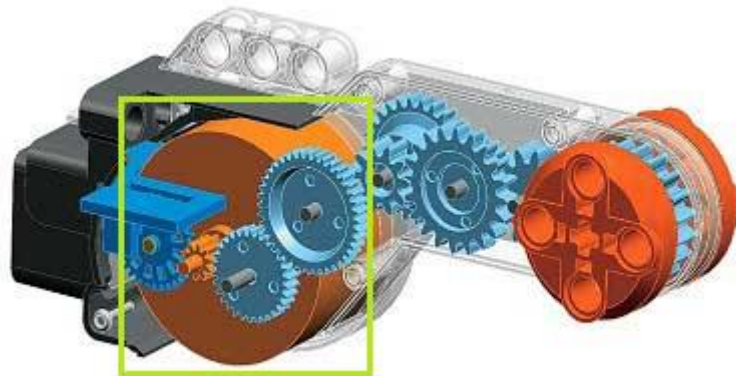


Figura 2.12. Ubicación del motor eléctrico dentro del motor NXT. [6]

2.2.1.2. Caja reductora.

La caja reductora contiene una serie de engranajes que aumentan el par que es capaz de entregar el motor, a costa de reducir la velocidad de giro del eje de salida del motor NXT. La caja posee 4 ejes de reducción, que vienen recogidos junto al número de dientes y su relación de transmisión en la Tabla 2.4.

RELACIÓN DE TRANSMISION EJE MOTOR – EJE SALIDA MOTOR NXT		
Número de eje	Relación de los engranajes	Relación de reducción
Eje 1	10:30:40	1:4
Eje 2	9:27	1:3
Eje 3	10:20	1:2
Eje 4	10:13:20	1:2
Relación total		1:48

Tabla 2.4. Relación de transmisión de los engranajes de la caja reductora. [6]

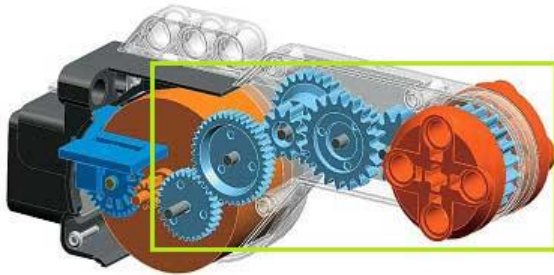


Figura 2.14. Ubicación de la caja reductora. [6]

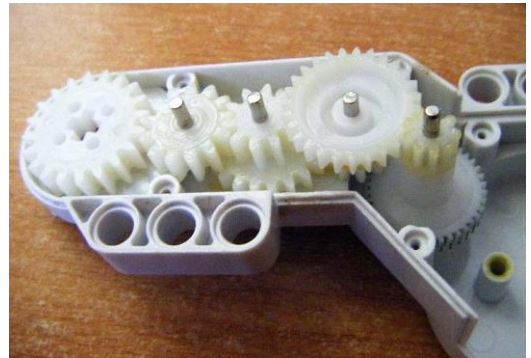


Figura 2.13. Detalle de la caja reductora. [6]

2.2.1.3. Encóder.

El encóder consiste en un engranaje de 32 dientes con 12 ventanas situado entre una foto barrera electrónica, acoplado al engranaje de 10 dientes del eje. Algunas características del encóder son:

- Trabaja a 5 V.
- Relación de 10:32 entre el encóder y el eje del motor.
- Es de tipo incremental.
- No posee señal N de referencia de posición.
- Resolución de 12 pulsos por vuelta, o lo que es lo mismo, de 30° por pulso.
- Admite evaluación cuadrática de las señales de pulsos.

Un factor que hay que tener en cuenta es que la resolución y la relación de transmisión se refieren al eje del motor eléctrico, y no al de salida



Figura 2.15. Detalle del encóder del motor NXT. [6]

del motor Lego NXT, que es el que interesa en este trabajo. Por tanto, es necesario adaptar la resolución a la del eje de salida del motor Lego NXT, que, dado que se conoce la relación entre el eje del motor y el eje de salida del motor Lego NXT, se puede hacer de la siguiente manera:

Sabiendo que las relaciones de transmisión motor - encoder y motor - salida son 10:32 y 1:48, el número de vueltas que gira el encóder por cada vuelta del eje de salida viene dado por las ecuaciones Ecuación 2.1 y Ecuación 2.2.

$$\theta_{eje\ motor}[vueltas] = \theta_{eje\ salida}[vueltas] * RT_{m \rightarrow s}^{-1} = 1 * \left(\frac{1}{48}\right)^{-1} = 48\ vueltas$$

Ecuación 2.1. Ecuación que determina el número de vueltas que gira el motor.

$$\theta_{encóder}[vueltas] = \theta_{eje\ motor}[vueltas] * RT_{m \rightarrow e} = 48 * \left(\frac{10}{32}\right) = 15\ vueltas$$

Ecuación 2.2. Ecuación que determina el número de vueltas que gira el encóder.

Donde $\theta_{eje\ motor}$ es el número de vueltas que ha girado el eje del motor, $\theta_{eje\ salida}$ es el número de vueltas que ha girado el eje de salida del motor NXT, $\theta_{encóder}$ es el número de vueltas que ha girado el encóder, $RT_{m \rightarrow s}$ es la relación de transmisión entre el eje del motor y el eje de salida del motor lego NXT y $RT_{m \rightarrow e}$ es la relación de transmisión entre el eje del motor y el encóder.

Por tanto, cuando el eje de salida ha dado una vuelta, la rueda del encoder ha dado 15 vueltas.

Si se tiene en cuenta que el encoder tiene 12 ventanas, es decir, que por cada vuelta se producen 12 pulsos, el número de pulsos del encoder por cada vuelta del eje de salida del motor Lego NXT viene dado por la Ecuación 2.3.

$$N_{encóder} \left[\frac{pulsos}{vuelta} \right] = \theta_{encóder}[vueltas] * N_{p/v} \left[\frac{pulsos}{vuelta} \right] = 15 * 12 = 180 \frac{pulsos}{vuelta}$$

Ecuación 2.3. Ecuación que determina la resolución en pulsos del encóder.

Donde $N_{encóder}$ es el número de pulsos del encóder por cada vuelta del eje de salida del motor Lego NXT, $\theta_{encóder}$ es el número de vueltas que ha girado el encóder por vuelta del eje de salida del motor NXT y $N_{p/v}$ es el número de pulsos por vuelta del encóder.

Debido a que se puede evaluar cuadráticamente el número de pulsos, este asciende hasta los $720 \frac{pulsos}{vuelta}$.

Otro dato importante es conocer la resolución del encóder en grados por pulso, ya que este valor determinará el error de posición. Puede calcularse mediante la Ecuación 2.4.

$$R_{\text{encóder}} \left[\frac{\text{º}}{\text{pulso}} \right] = \frac{360\text{º}}{N_{\text{encóder}} \left[\frac{\text{pulsos}}{\text{vuelta}} \right]} = \frac{360}{720} = 0,5 \frac{\text{º}}{\text{pulso}}$$

Ecuación 2.4. Ecuación que determina la resolución en grados del encóder.

Donde $R_{\text{encóder}}$ es la resolución del encóder y $N_{\text{encóder}}$ es el número de pulsos del encóder.

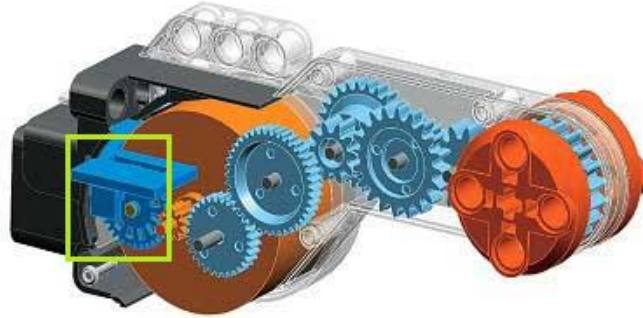


Figura 2.16. Ubicación del encóder en el motor NXT. [6]

2.2.1.4. Conector de alimentación.

Por último, el conector RJ12 que contiene las conexiones eléctricas del motor, y la alimentación y las salidas del encoder, como se puede ver en la Figura 2.17. El conector contiene las siguientes conexiones:

1. Señal A del encóder (Amarillo).
2. Señal B del encóder (Azul).
3. + 5 V (Verde).
4. GND (Rojo).
5. Señal 1 de potencia del motor (Blanco).
6. Señal 2 de potencia del motor (Negro).



Figura 2.17. Detalle del conector del motor NXT.

2.2.2. Disco de inercia.

Para permitir la visualización del ángulo girado por el motor y para hacer las funciones de carga, acoplado al eje de salida del motor va montado un disco, Figura 2.19, cuyas características son las siguientes:

- Fabricado en poliacetal (POM), un tipo de termoplástico cristalino de alta rigidez.
- 100 mm de diámetro.
- 15 mm de ancho.
- 161,4 gramos de peso.
- Posee una escala graduada de 10 en 10 grados para visualizar la posición angular respecto de la referencia.

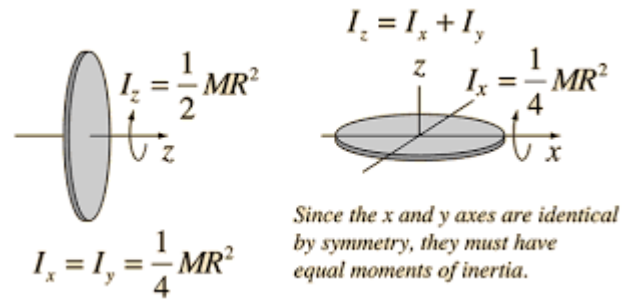


Figura 2.18. Momentos de inercia de un disco plano. [11]

Un detalle muy importante es conocer la inercia que posee el disco. Como el disco gira en el eje Z, es decir, el perpendicular a su superficie, tal como se ve en la Figura 2.18, la Ecuación 2.5 da el valor de su momento de inercia.

$$I_z [kg \cdot m^2] = \frac{1}{2} * M [kg] * R^2 [m] = \frac{1}{2} * (161,4 * 10^{-3}) * \left(\frac{100}{2} * 10^{-3}\right)^2 = 2,0175 * 10^{-4} kg * m^2$$

Ecuación 2.5. Ecuación que determina el valor del momento de inercia del disco.

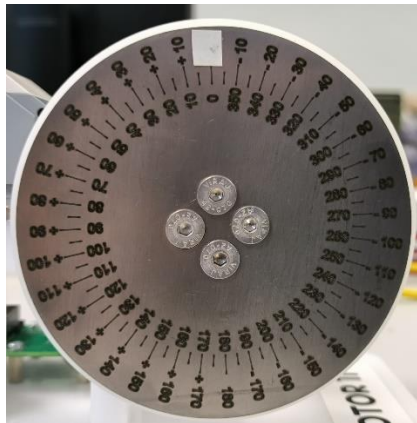


Figura 2.19. Disco de inercia.

2.2.3. Sensor inductivo.

Debido que el encóder no posee marca de referencia N, y es necesario tenerla para referenciar correctamente el autómeta, la maqueta incluye un sensor inductivo para realizar esta tarea. Las características más importantes de este sensor son:

- Se alimenta a 5 V. No admite otra tensión.
- Requiere 30 mA.
- Detecta metales a una distancia de hasta 4 mm.
- Es de tipo NPN, es decir, salida normalmente abierta.
- Protección frente polvo y agua IP67.
- Tiempo de respuesta menor de 1 ms.
- Es capaz de operar en un rango de temperaturas de -20 °C hasta 65 °C.



Figura 2.20. Sensor inductivo de la maqueta del disco de inercia. [9]

Para su conexión se dispone de 3 hilos:

- Señal (Negro).
- + 5 V (marrón).
- GND (Azul).

Como el sensor se alimenta a 5 V, la maqueta posee un circuito que convierte la señal de 5 V que sale del sensor, en una de 24 V que puede ser usada, por ejemplo, por el autómeta, Figura 2.21.

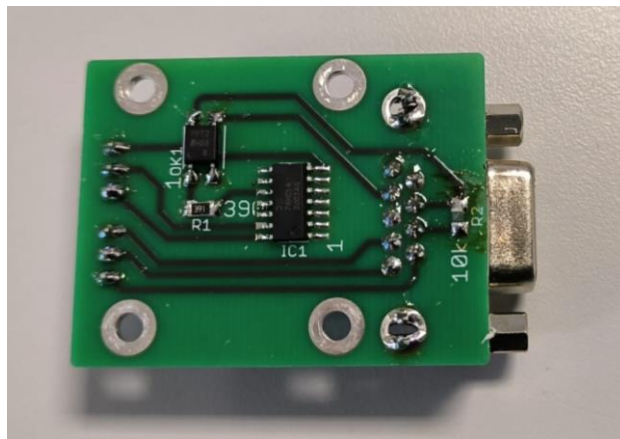


Figura 2.21. Placa adaptadora de tensión del sensor inductivo.

2.2.4. Ascensor.

Esta maqueta representa un pequeño ascensor de 3 plantas (2 pisos + la planta baja). La maqueta se compone de una base, una torre que actúa como soporte y una cabina que, a excepción de las guías de la cabina que son metálicas, están hechas de policarbonato.

Para poder controlarlo, cada piso lleva asociados un pulsador, que representaría el botón de llamada de ese piso, y un final de carrera, cuya utilidad es la de indicar que se ha alcanzado el piso correspondiente.

La maqueta del ascensor puede dividirse en dos partes:

- Por un lado, la parte mecánica
- Por otro lado, la parte eléctrica.

2.2.4.1. Parte mecánica

Esta parte se encarga de convertir el movimiento angular del motor en movimiento lineal que permite mover la cabina del ascensor. Como puede verse en la Figura 2.23, el sistema mecánico del ascensor está formado por:

- r1. Es una rueda dentada de 8 dientes acoplada al eje del motor NXT.
- r2. Es una rueda dentada de 24 dientes conectada a la polea del ascensor y engranada con r1.
- p1. Es una rueda de polea dentada con las siguientes características:
 - 21 mm de ancho.
 - El paso de la polea, es decir, la distancia entre los dientes, es de 5 mm.
 - Tiene 16 dientes.
 - La correa de la polea es de perfil B con un ancho primitivo de 14 mm.
 - Diámetro primitivo de 25,47 mm.
- p2. Es idéntica a p1.



Figura 2.22. Polea del ascensor.

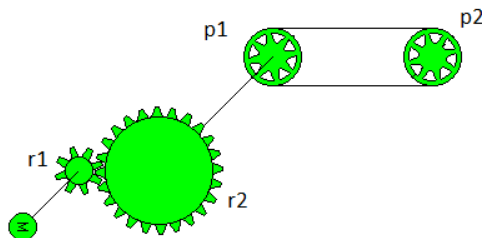


Figura 2.23. Diagrama del sistema mecánico del ascensor.

Para realizar correctamente el control de posición del ascensor, es necesario conocer el avance de la polea, es decir, cuanta distancia avanza la correa de la polea, o en este caso la

cabina del ascensor, cuando la rueda de la polea da una vuelta, valor que es igual a la longitud de la rueda de la polea y puede calcularse mediante la Ecuación 2.6.

$$L_{ascensor} = D_p * \pi = 25,47 * \pi = 80,01 \text{ mm} \sim 80 \text{ mm}$$

Ecuación 2.6. Ecuación que determina la longitud de la polea.

Donde D_p es el diámetro primitivo de la rueda de la polea.

2.2.4.2. Parte eléctrica

Esta parte está formada por todo el sistema eléctrico que posee la maqueta y que se necesita para su control. Aquí se incluyen los siguientes elementos:

- Finales de carrera.
- Pulsadores.
- LEDs.

En la Tabla 2.5 se recoge la información sobre estos elementos.

PARTE ELÉCTRICA DE LA MAQUETA DEL ASCENSOR – SENSORES E INDICADORES		
FINALES DE CARRERA		
FCB	NC	Detector planta 0
FC1	NC	Detector planta 1
FC2	NC	Detector planta 2
FC Inf	NC	Detector límite inferior
FC Sup	NC	Detector límite superior
PULSADORES		
PB	NO	Llamada planta 0
P1	NO	Llamada planta 1
P2	NO	Llamada planta 2
PI	NO	Función auxiliar 1
PF	NO	Función auxiliar 2
LEDs		
ROJO	Cátodo	Indicador de emergencia
LEDPB	Cátodo	Indicador planta 0
LEDP1	Cátodo	Indicador planta 1
LEDP2	Cátodo	Indicador planta 2

Tabla 2.5. Parte eléctrica de la maqueta del ascensor.

Para poder conectar la maqueta al autómatas, se dispone de una manguera con un conector DB25 Macho, Figura 2.24, que incluye las conexiones de los elementos mencionados anteriormente, así como las correspondientes líneas de alimentación y de masa para alimentar dichos elementos.

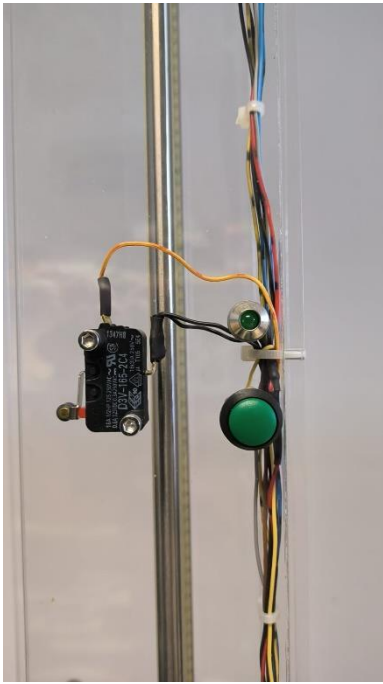


Figura 2.25. Detalle del conjunto formado por el Final de carrera, el pulsador y el LED

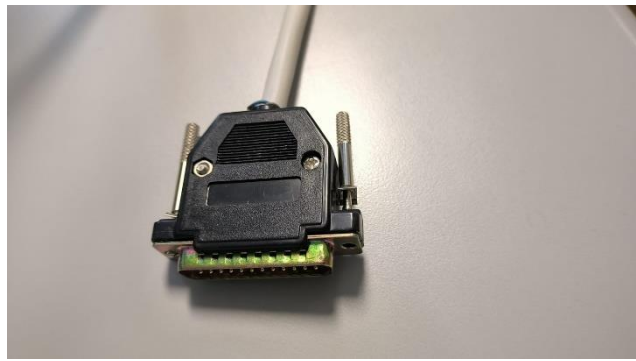


Figura 2.24. Conector DB25 de la maqueta del ascensor.

3. Desarrollo del programa de control del autómeta.

En este capítulo se va a mostrar y a desarrollar cómo ha sido el proceso de elaboración del programa de control del autómeta. Este capítulo se corresponde con la etapa 4 del desarrollo del trabajo.

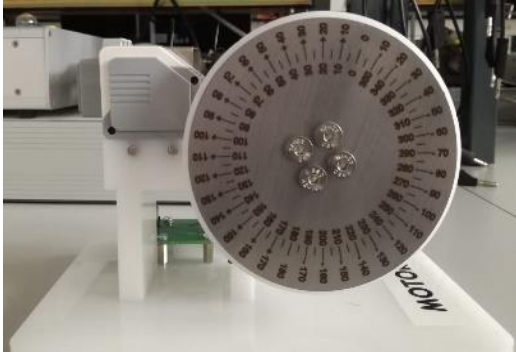


Figura 3.1. Maqueta del disco de inercia utilizada en los proyectos 1 y 2.

Para empezar, debido a que en el proyecto se han utilizado 2 maquetas con movimientos muy diferentes, rotativo en el caso de la maqueta del disco y lineal en el caso de la maqueta del ascensor, ha sido necesario crear un proyecto diferente para cada una.

Con el objetivo de poner en práctica los conocimientos adquiridos sobre las instrucciones Motion Control, los objetos tecnológicos y sobre los bloques de programación, se ha elaborado un pequeño programa, que contiene una función en lenguaje LD, que incluye las instrucciones Motion Control necesarias para controlar el movimiento de la maqueta del disco y que, a su vez, son controladas por el usuario a través de la pantalla HMI.

Posteriormente, una vez el programa funciona se ha ampliado el proyecto, añadiendo otra función, esta vez en lenguaje SFC (Graph), para realizar el control de posición, tanto de la maqueta del disco, como del ascensor, con la diferencia de que esta vez el usuario controla la función en lenguaje SFC y esta, a la función que contiene las instrucciones Motion Control. Una vez comprobado que ambos programas funcionan, se ha realizado la programación completa de la maqueta del ascensor, pero esta vez haciendo uso del lenguaje ST.

Por tanto, este trabajo consta de 4 proyectos distintos:

- Proyecto 1. Control de posición angular directamente mediante instrucciones Motion Control.
- Proyecto 2. Control de posición angular mediante programación en lenguaje SFC.
- Proyecto 3. Control de posición lineal mediante programación en lenguaje SFC.
- Proyecto 4. Control de posición lineal mediante programación en lenguaje ST.



Figura 3.2. Maqueta del ascensor utilizada en los proyectos 3 y 4.

De estos 4 proyectos, los más representativos del trabajo son el proyecto 2 y el proyecto 4, por lo que van a ser los que se expliquen a continuación.

Cabe destacar que tanto el proyecto 2 como el proyecto 3 han sido utilizados en el curso 2022-2023 en las prácticas de la asignatura de Automatización Industrial, impartida por el profesor Ramón Piedrafita.

Para empezar la secuencia para crear el programa ha sido:

1. Crear el proyecto.
2. Elegir la CPU.
3. Añadir variables del sistema.
4. Configurar los elementos de la CPU.
5. Añadir y configurar el objeto tecnológico.
6. Crear el programa de control.

Como el proyecto 2 es una ampliación del 1 y el proyecto 4 es una mejora del 3, solamente se van a explicar estos dos proyectos.

3.1. Proyecto 2. Control de posición angular mediante programación en SFC.

3.1.1. Creación del proyecto.

El primer paso ha sido crear el proyecto. El nombre elegido para el proyecto ha sido Proyecto_2_TFG_Motor y la versión la v15.1 ya que es la única instalada en los equipos del laboratorio.

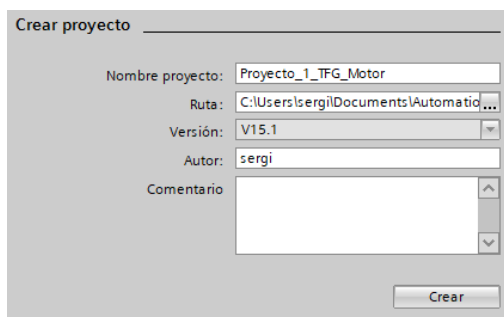


Figura 3.3. Pantalla de creación del nuevo proyecto.

Después de haber creado el proyecto, se ha elegido la CPU con la que se va a trabajar. Como se ha mencionado antes la CPU es la S7-1512C-1 PN cuyo número de modelo es 6ES7512-1CK01-0AB0. También se ha seleccionado la versión más alta disponible, la 2.6.

Una vez que se ha creado y configurado inicialmente el proyecto, se continua con la configuración de las entradas, salidas y variables de memoria del proyecto.

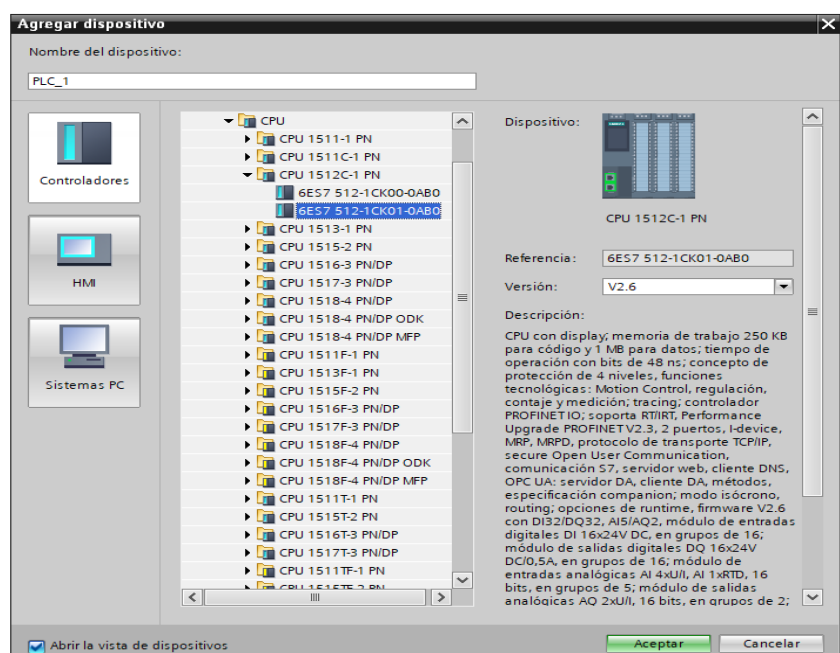


Figura 3.4. Menú de selección de CPU.

3.1.2. Configuración del encóder.

Antes de poder configurar un objeto tecnológico, es necesario configurar un par de aspectos como el encóder o los generadores de PWM. En este apartado se muestra como se ha configurado el encóder que necesita el objeto tecnológico para obtener la posición real del motor de las maquetas. La configuración tomada es la siguiente:

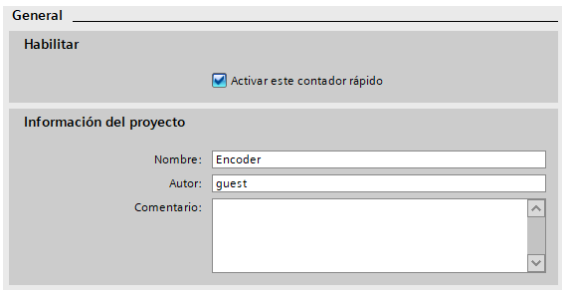


Figura 3.5. Apartado de activación del contador rápido.

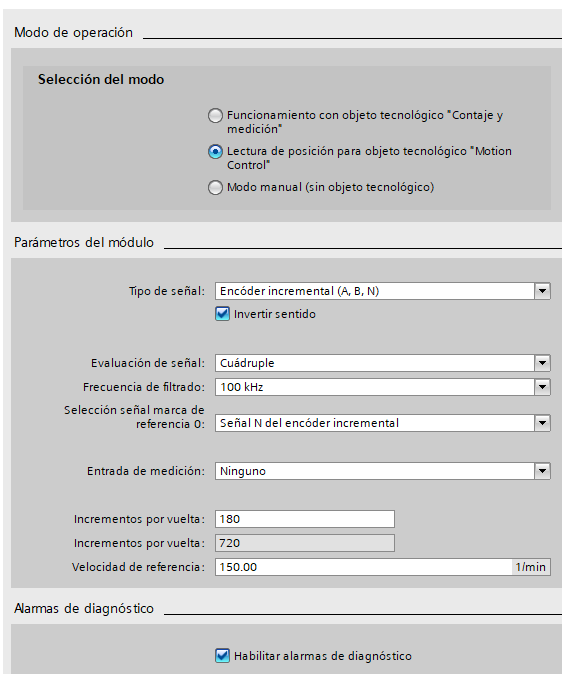


Figura 3.6. Apartado de parametrización de la configuración del encóder.

Respecto a los incrementos por vuelta, como se menciona en el capítulo 2, el encóder es de 180 pulsos por vuelta, por tanto, se elige este valor.

Y finalmente como velocidad de referencia se elige 150 rpm, tal y como se ha calculado en el Apéndice B respecto a la velocidad máxima posible del motor.

También se ha activado la casilla de habilitar alarmas de diagnóstico para facilitar la resolución de los posibles errores que puedan existir.

El resto de las opciones, como las entradas y salidas hardware o las direcciones de E/S del encóder se han dejado por defecto.

El primer paso, dentro del submenú general, ha sido habilitarlo. También, para facilitar su identificación se le ha dado el nombre de encóder.

Una vez habilitado, y dentro del submenú canal 0, se le ha dado el modo de funcionamiento "Lectura de posición para objeto tecnológico "Motion Control".

A partir de aquí se ha parametrizado el encóder de las maquetas. Como tipo de señal se ha elegido Encóder incremental (A, B, N) ya que, aunque el encóder real no dispone de posición N, el inductivo cumple esta función, como se menciona en el capítulo 2. A parte, se ha seleccionado la casilla Invertir sentido, ya que durante las pruebas se encontró que había un error en la conexión del encóder y era necesario invertir ambas señales.

Para tener más precisión en el control se ha elegido la opción cuádruple en el apartado de evaluación de señal.

La frecuencia de filtrado se ha dejado por defecto.

Como selección de marca de referencia 0, se ha elegido la propia señal N del encóder incremental.

En este caso, interesa que el encóder este activo siempre, por lo tanto, la entrada de medición se parametriza en ninguno.

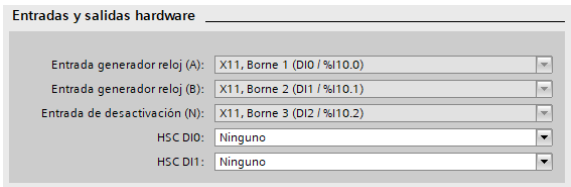


Figura 3.7. Apartado de parametrización de la conexión del encóder.

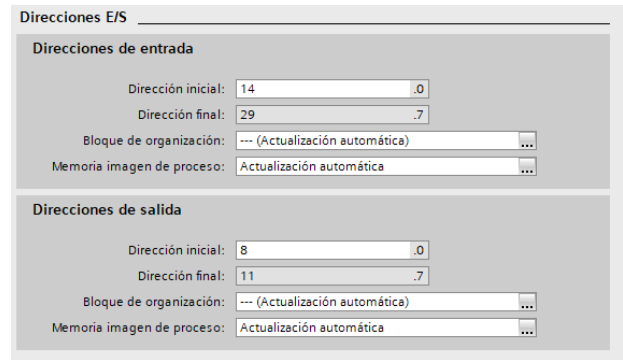


Figura 3.8. Apartado de parametrización de las direcciones de entrada y salida del encóder.

3.1.3. Configuración de los generadores PWM.

Como la referencia de tensión que da el objeto tecnológico es analógica y está comprendida entre -10 V y + 10 V y dado que el motor trabaja con señales PWM unipolares de 9 V, ha sido necesario transformar la referencia del objeto tecnológico en señales PWM. Para ello, lo primero es activar dichos generadores. A diferencia del encóder donde solo se necesitaba 1, aquí se necesitan 2. La configuración que se ha tomado para los generadores es la siguiente:

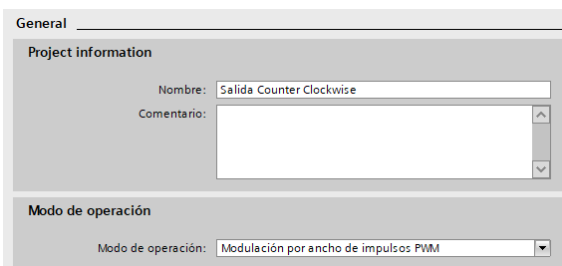


Figura 3.9. Apartado de activación del generador PWM.

El primer paso es habilitarlos y darles la función que van a tener, por lo que en modo de operación se elige la opción Modulación por ancho de pulsos PWM. También se les da un nombre para ayudar a su identificación, Salida Counter Clockwise para el primero y Salida Clockwise para el segundo.

El resto de los parámetros son comunes a ambos, por lo que solamente se van a explicar en uno de ellos.

Como reacción a STOP de la CPU, se ha elegido que se aplique un valor sustitutivo y que este valor sea 0, ya que interesa que cuando la CPU se detenga, la maqueta también lo haga.

Al igual que con el encóder se ha marcado la casilla activar alarmas de diagnóstico para facilitar en caso de que se produzca un error, su identificación y solución.

Como salida del generador se ha dejado la que viene por defecto, pero se ha activado la casilla High-Speed, que aumenta la frecuencia que permite sacar la salida, dado que se ha decidido trabajar con una frecuencia de 25 kHz y la salida normal solo permite 10 kHz pero, a cambio, se reduce la corriente que puede sacar el generador, aunque en este trabajo no es un problema ya que la potencia que le llega al motor lo hace a través de un controlador.

Como parámetros, se ha elegido que el formato de salida sea de 1/100, para tener el valor de duty en tanto por ciento, que la duración mínima sea de 0 microsegundos para poder tener un duty del 0% y que la duración del periodo sea de 40 microsegundos para obtener la frecuencia de 25 kHz, antes mencionada.

Las direcciones de E/S se han dejado por defecto.

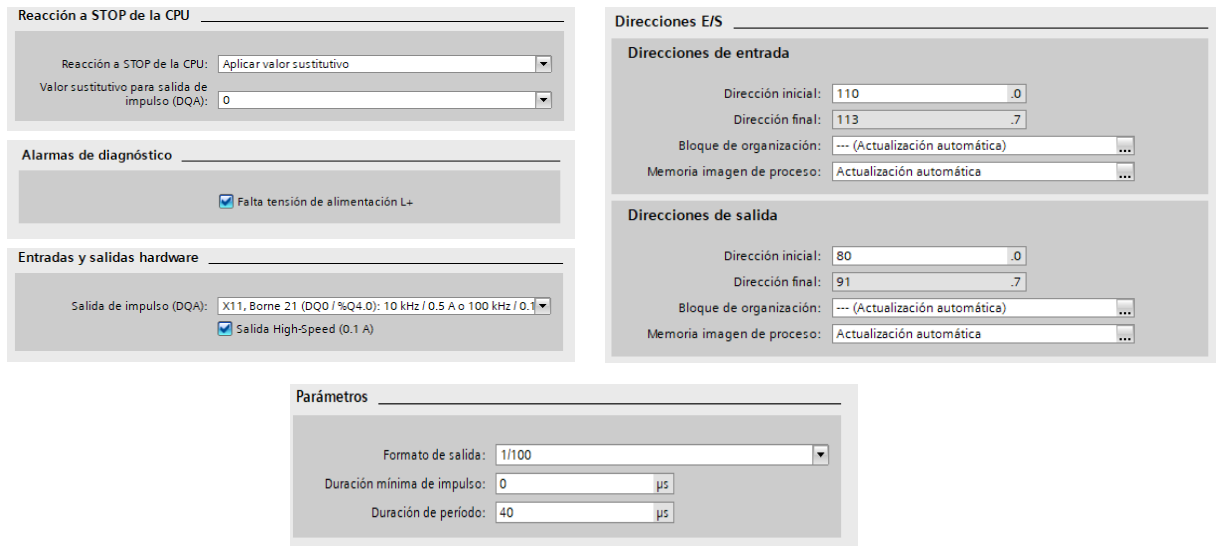


Figura 3.10. Apartados de configuración del generador PWM.

3.1.4. Configuración del objeto tecnológico.

Una vez configurados el encóder y los generadores PWM, ya se puede configurar el objeto tecnológico. De todas las opciones posibles, que se pueden consultar en el Apéndice A, se ha elegido, como objeto tecnológico, un eje de posición ya que se está realizando un control de posición y se le ha dado el nombre de Eje_Motor.

Una vez creado el objeto, se procede a su configuración. Este, al igual que con el encóder y los generadores PWM se divide en submenús, y cada submenú se divide en varios apartados. A continuación, se muestra la configuración de cada submenú.

3.1.4.1. Parámetros básicos.

Dentro de este submenú, en el parámetro tipo de eje, se ha elegido un tipo de eje rotativo ya que, en esta maqueta, se está haciendo un control de posición angular. Respecto a las unidades de medida se han elegido:

- $^{\circ}$ para la posición.
- $^{\circ}/s$ para la velocidad.
- Nm para el par.

Por último, las opciones operación modulo y simulación se han dejado desactivadas ya que no son necesarias.

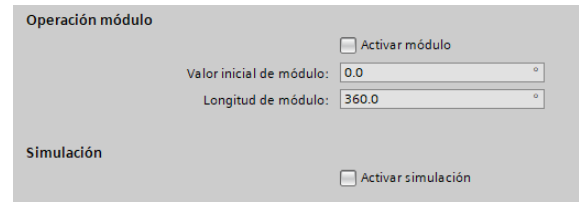
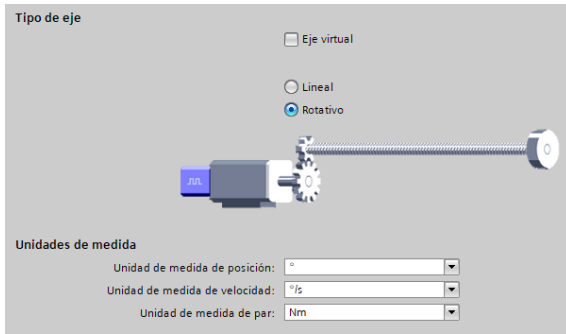


Figura 3.11. Configuración de los parámetros básicos del objeto tecnológico.

3.1.4.2. Interfaz de hardware.

Dentro de este submenú, en el apartado accionamiento, se ha elegido conexión analógica del motor como accionamiento mediante la variable *Acción_Motor* de tipo Int, indicado en el campo salida analógica. No se han utilizado ni salida de habilitación ni entrada de disponibilidad.

En el siguiente apartado, el de configuración del encóder, se ha elegido encóder como conexión de datos y se ha seleccionado el encóder que se ha configurado anteriormente.

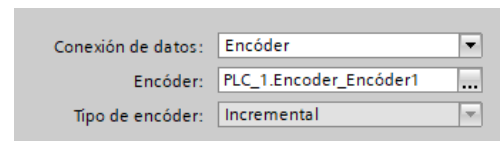
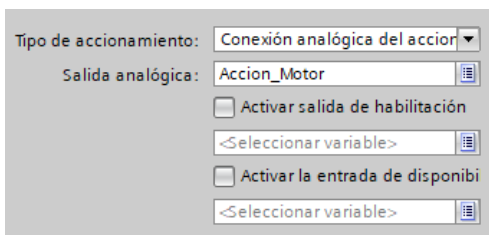


Figura 3.12. Apartados de selección del accionamiento y del encóder del objeto.

En el tercer apartado intercambio de datos con accionamiento, el único parámetro que se ha configurado es la velocidad de giro de referencia, que se ha establecido en 128,20 rpm, tal y como se ha calculado en el Apéndice B.

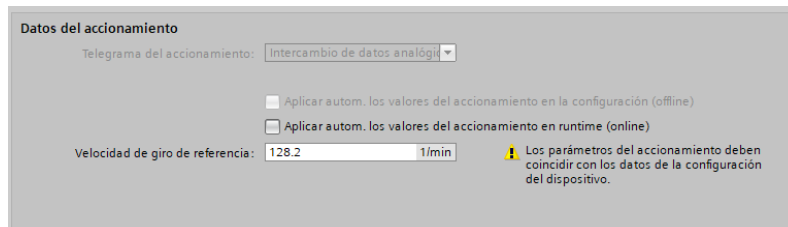


Figura 3.13. Apartado de configuración del accionamiento del objeto.

Por último, en el apartado de intercambio de datos con el encóder, el telegrama del encóder se ha dejado por defecto en telegrama 83 y se ha marcado la opción intercambio de datos automático para valores del encóder (offline).

Figura 3.14. Apartado de configuración del encóder del objeto.

3.1.4.3. Parámetros Avanzados.

En el submenú parámetros avanzados y dentro del apartado mecánica, se ha configurado que el modo de montaje del encóder sea en el eje del motor y que la relación, especificada mediante el parámetro reductor de carga, sea de 1:1.

Figura 3.15. Apartado de configuración mecánica del objeto.

Respecto al preajuste de dinámica, es decir, los valores nominales que seguirá el motor cuando realice el movimiento, se han tomado los valores nominales para esta maqueta calculados en el Apéndice B. Estos son:

- Velocidad: 769,23 ^o/s.
- Tiempo de aceleración: 0,75 s.
- Tiempo de deceleración: 0,75 s.
- Tiempo de redondeo: 0,1 s.

El resto de los parámetros son calculados automáticamente por TIA Portal al introducir estos valores.

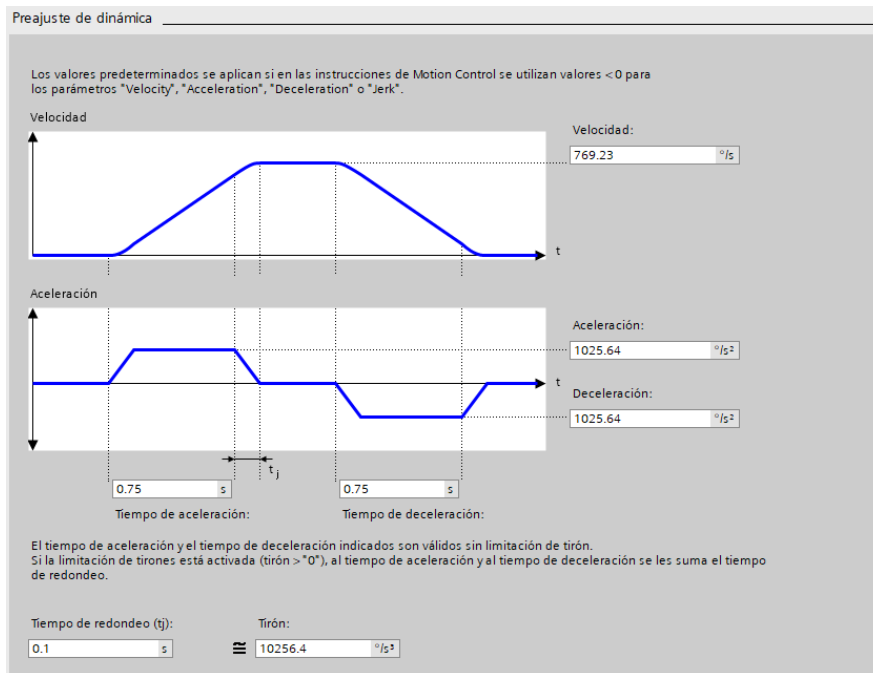


Figura 3.16. Apartado de configuración del preajuste de dinámica del objeto tecnológico.

En el apartado parada de emergencia, se ha tomado el valor del tiempo calculado en el Apéndice B, que es de 0,25 s. El resto de los parámetros son calculados automáticamente por TIA Portal.

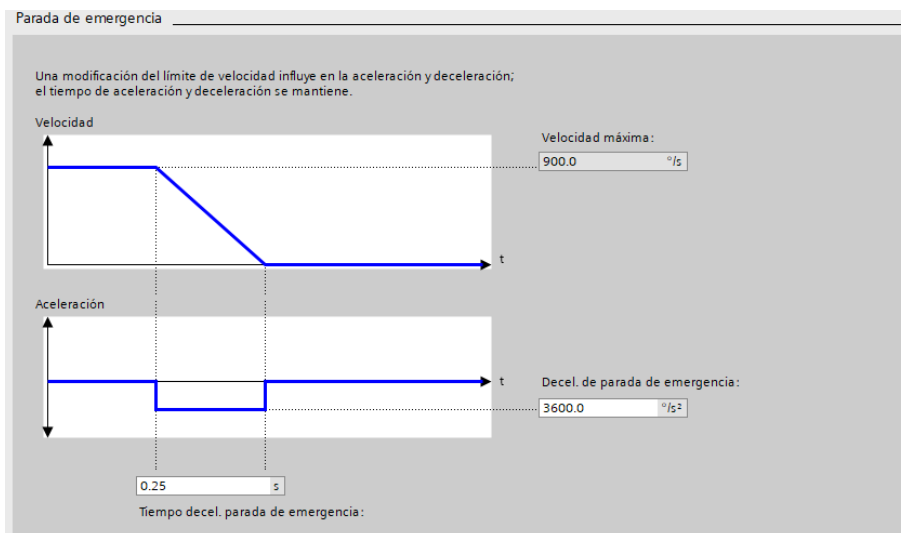


Figura 3.17. Apartado de configuración del preajuste de dinámica del objeto tecnológico.

El siguiente apartado, son los límites de posición, que en esta maqueta no se han utilizado y se han dejado desactivados.

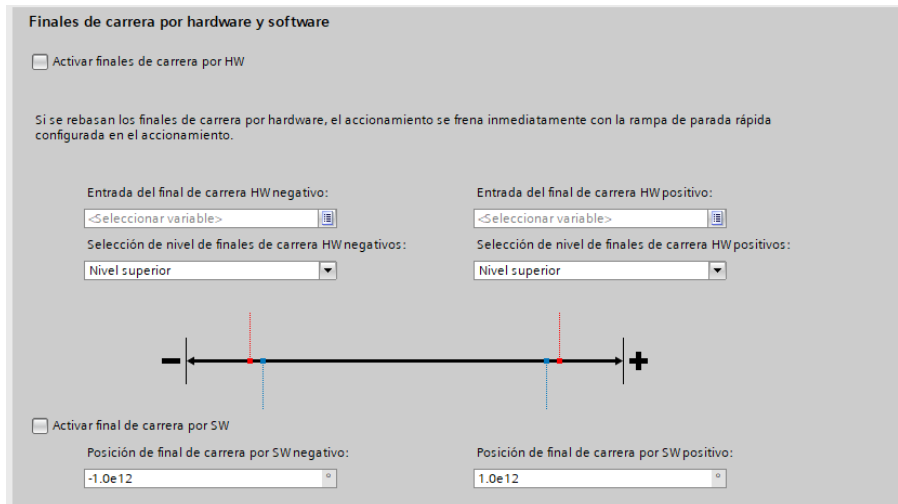


Figura 3.18. Apartado de configuración de los límites de movimiento del objeto tecnológico.

En el apartado límites dinámicos y al igual que con los valores nominales y de parada de emergencia, se han introducido los valores correspondientes calculados en el Apéndice B. Estos son:

- Velocidad máxima: 900 °/s.
- Tiempo de aceleración: 0,25 s.
- Tiempo de deceleración: 0,25 s.
- Tiempo de redondeo: 0,05 s.

Al igual que en los apartados anteriores, el resto de los parámetros son calculados automáticamente por TIA Portal.

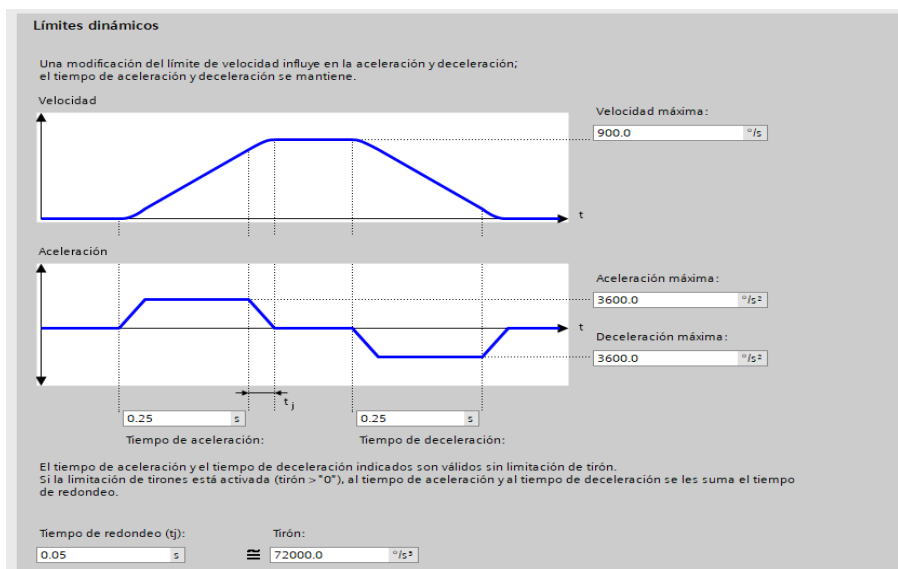


Figura 3.19. Apartado de configuración de los límites dinámicos del objeto tecnológico.

El apartado detección de tope fijo se ha dejado con la configuración por defecto que da TIA Portal.

Dentro del apartado referenciado, se ha dejado todo como viene por defecto en TIA Portal, salvo que en el referenciado activo, se han introducido los valores de velocidad calculados en el Apéndice B. Estos son:

- Velocidad de aproximación: 120 ^o/s.
- Velocidad de referenciado: 100 ^o/s.

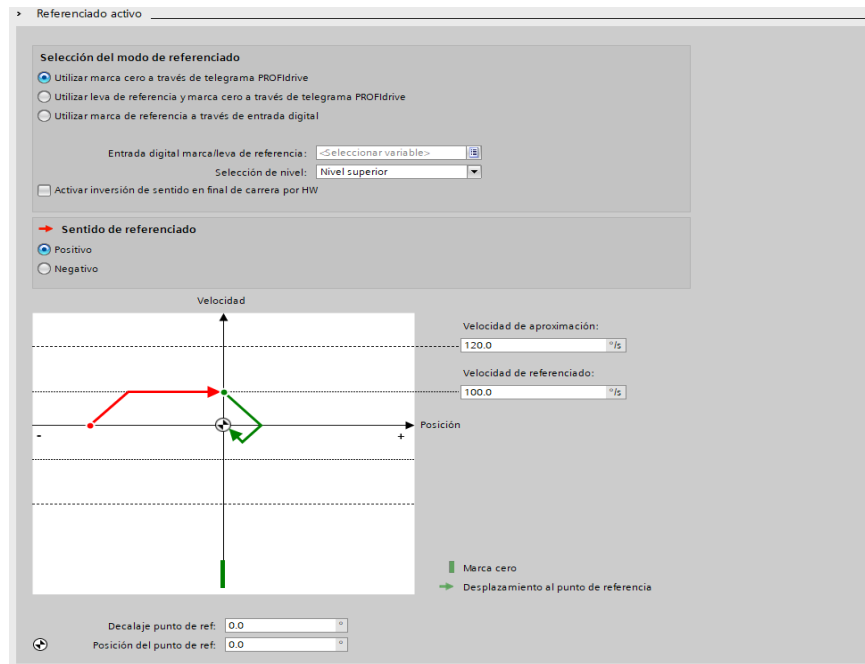


Figura 3.20. Apartado de configuración del referenciado activo del objeto tecnológico.

En el siguiente apartado, vigilancia de posicionamiento, se ha decidido que la ventana de posicionamiento sea de 10^o ya que, debido a la precisión del encóder, en las pruebas saltaba algún error de posicionamiento y este es el valor mínimo para que no aparezca dicho error. Respecto a los tiempos de tolerancia y mínimo de permanencia en la ventana de posicionamiento, se ha decidido que estos sean de 10 segundos para darle tiempo al motor a que se detenga por completo y evitar la aparición de errores.

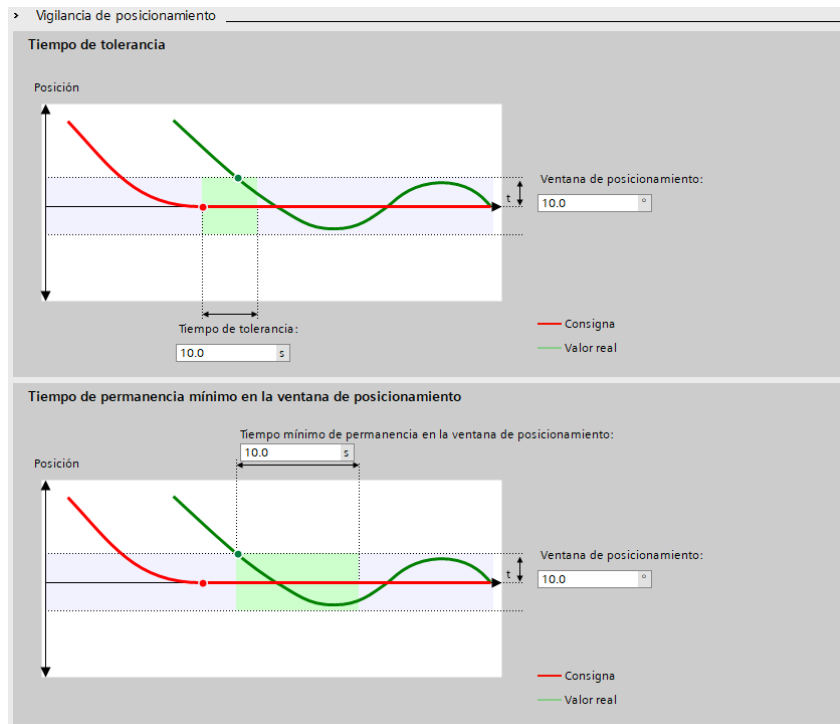


Figura 3.21. Apartado de configuración de la vigilancia de posicionamiento del objeto tecnológico.

El apartado error de seguimiento se ha dejado desactivado ya que, en las pruebas realizadas, generaba un error y su utilización tampoco aportaba nada al trabajo.

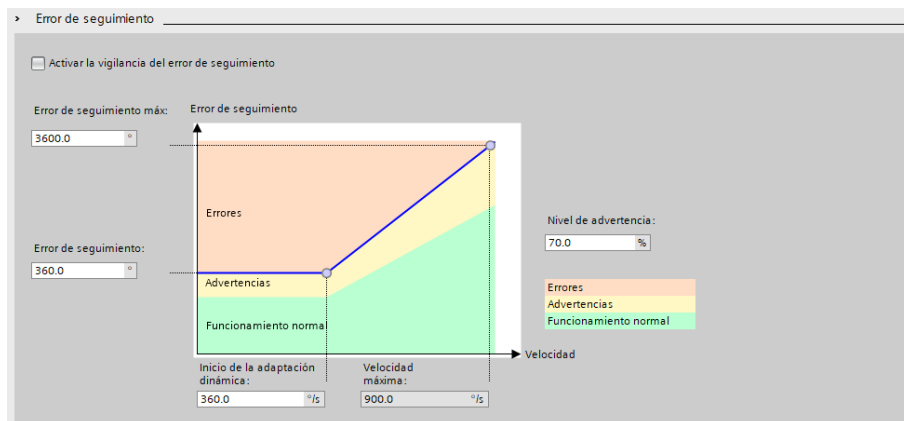


Figura 3.22. Apartado de configuración del error de seguimiento del objeto tecnológico

En el penúltimo apartado, señal de velocidad cero, se ha decidido que la ventana de velocidad cero sea de 75 y el tiempo de permanencia sea de 2 segundos. Estos valores se han elegido arbitrariamente y durante las pruebas realizadas se ha encontrado que no generaban ningún tipo de problema.

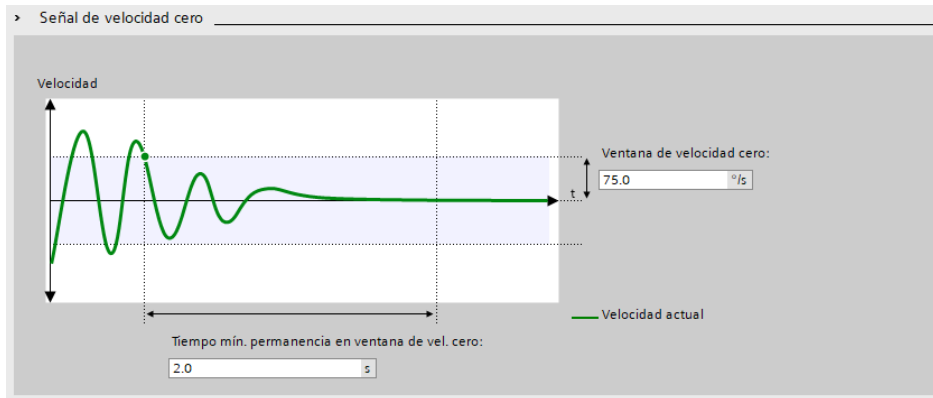


Figura 3.23. Apartado de configuración de la señal de velocidad cero del objeto tecnológico.

Por último, en el apartado lazo de regulación, los parámetros precontrol y tiempo sustitutivo se han dejado por defecto como vienen en TIA Portal. En cambio, el parámetro ganancia se ha ajustado mediante un ensayo de prueba y error y se ha determinado que el valor óptimo es 40. El parámetro Dynamic Servo Control (DSC) se ha dejado en regulación de posición en el PLC ya que no es posible hacerlo en el accionamiento.

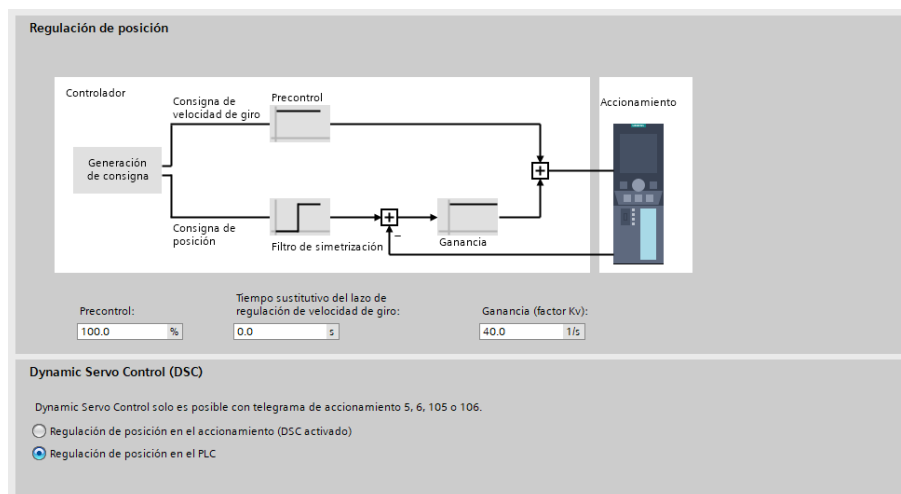


Figura 3.24. Apartado de configuración de la regulación de posición del objeto tecnológico.

3.1.5. Creación del programa de control.

Una vez creado y configurado el objeto tecnológico, ya se puede crear el programa de control que controlará dicho objeto. En este proyecto se han creado tres funciones, las cuales, son las siguientes:

- Función conversor PWM. El objetivo de esta función es convertir la señal de referencia analógica que genera el objeto tecnológico en los duty correspondientes a cada generador PWM para poder ser utilizada en el motor.
- Función control del motor. El objetivo de esta función es controlar el objeto tecnológico mediante las instrucciones Motion Control.
- Función control de la maqueta. El objetivo de esta función es controlar el movimiento de la maqueta

A continuación, se muestra el desarrollo de cada función.

3.1.5.1. Función conversor PWM.

El objetivo de esta función es transformar la salida analógica del objeto tecnológico, a 2 señales PWM, una por sentido, para realizar el control de la maquetas.

Como se ha mencionado en el capítulo 2, información sobre el autómata, el control del generador PWM se hace mediante bits y flags. Según el manual de producto del autómata, la generación de la señal PWM se hace de la siguiente manera:

- El primer paso es transferir el control de la salida PWM al autómata, ajustando el valor de TM_CTRL_DQ a "1".
- El segundo paso es habilitar la salida del generador PWM, ajustando el valor SW_ENABLE a "1".
- El tercer paso es introducir el valor de duty mediante OUTPUT_VALUE.

De esta manera, ya se estaría generando una señal PWM con el duty requerido.

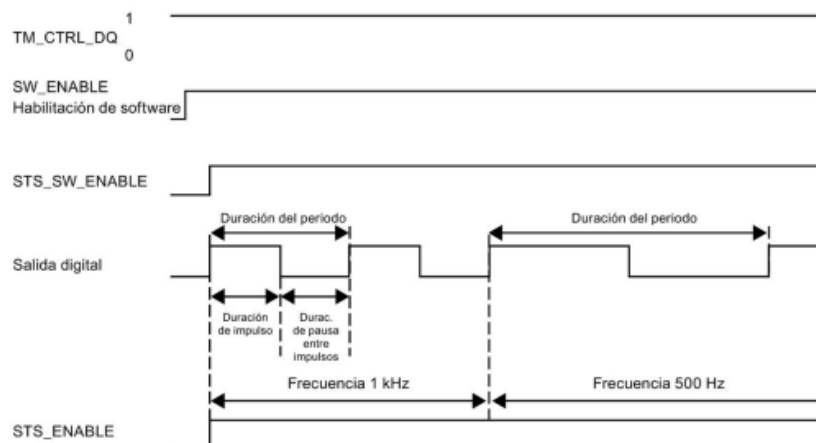


Figura 3.25. Secuencia de generación de la señal PWM. [3]

Una vez que conocida la forma en la que se genera la señal PWM, se continúa creando la función. En este caso, la función se ha creado en un bloque FC ya que no es necesario

almacenar ningún tipo de información. La secuencia de funcionamiento de la función es la siguiente:

Inicialmente, se limitan los valores, ya que el rango de la variable analógica que da el objeto tecnológico es de $[-32768, 32767]$, que correspondería al rango de tensión $[-11,85 \text{ V}, 11,85 \text{ V}]$ y el rango de trabajo es $[-27648, 27648]$ o en tensión $[-10 \text{ V}, 10 \text{ V}]$. Esto se realiza

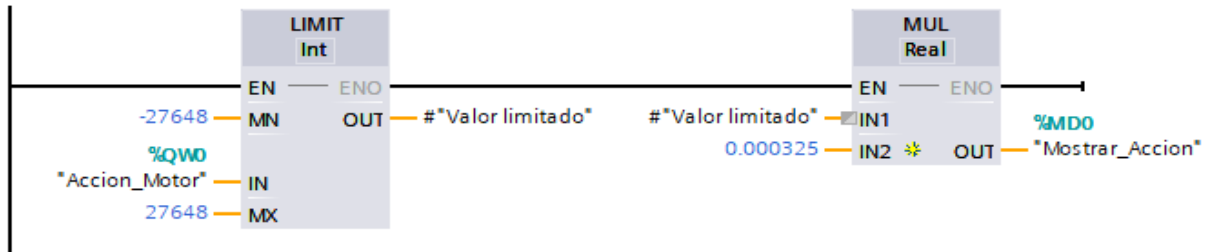


Figura 3.26. Segmento 1 de la función conversor PWM.

usando la instrucción LIMIT donde tanto la variable de entrada, *Acción_Motor*, como la de salida, *Valor limitado* son de tipo Int. A continuación, se encuentra una instrucción MULT para convertir *Valor limitado* en el rango $[-10, 10]$ para que pueda ser mostrado en la pantalla HMI. La variable de salida de esta instrucción es *Mostrar_Accion*, que es una variable global.

Una vez que se han limitado los valores al rango de trabajo, se normalizan, ya que se requiere que estén en el rango $[0, 100]$, mediante la instrucción NORM_X, en la cual se le especifica que la variable de entrada, *Valor limitado* es de tipo Int y la de salida, *Valor normalizado*, es de tipo Real.

Después de la instrucción NORM_X, se encuentra la instrucción SCALE_X donde los valores se escalan del rango $[0,1]$ al rango $[-1,1]$. A esta instrucción se le especifica que tanto la entrada, *Valor normalizado*, como la salida, *Valor escalado*, son del tipo Real.

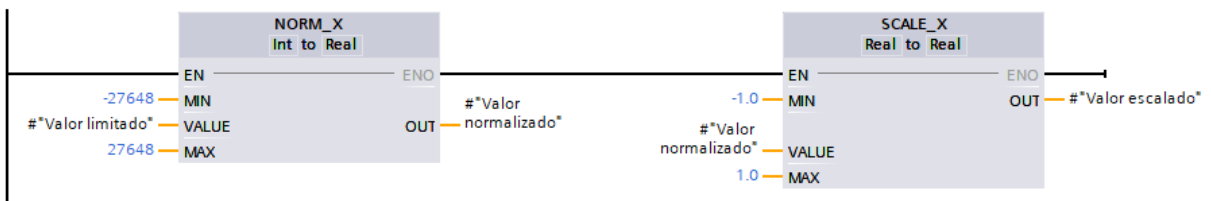


Figura 3.27. Segmento 2 de la función conversor PWM.

Cuando ya se tiene la salida escalada, estos valores se diferencian entre positivos y negativos para aplicarlos a las correspondientes señales PWM. Si el valor es menor que 0, la acción se aplica en la salida del PWM Counter-Clockwise, mientras que, si es positivo o 0, se aplica en la salida PWM clockwise. A parte de hacer la comparación, se necesita volver a escalar el valor de la variable *Valor escalado*, a un valor entre 0 y 100, que es el duty aplicado a cada generador PWM.

Para los valores positivos de la variable *Valor escalado*, simplemente se escalan del rango $[0, 1]$ al rango $[0, 100]$ usando, de nuevo, la instrucción SCALE_X, pero esta vez pasando de una variable tipo Real a una variable tipo UInt. El valor escalado se guarda en la variable *Duty Clockwise*.

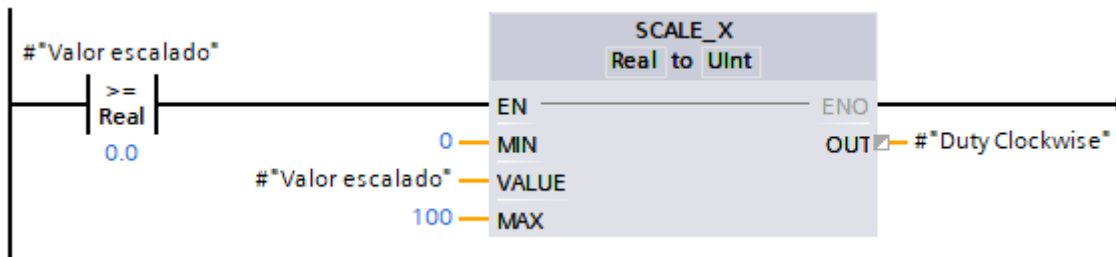


Figura 3.28. Segmento 3 de la función conversor PWM.

Para los valores negativos de la variable *Valor escalado*, primero se necesita aplicarle el valor absoluto a la variable, puesto que, si simplemente se escala, el valor queda invertido, es decir, cuando la variable *Valor escalado* vale -1, que sería el duty máximo, la señal escalada vale 0, que es el duty mínimo. Por ello, se utiliza la instrucción ABS, donde el valor se guarda en la variable *Valor absoluto*. Posteriormente, al igual que en el caso de la variable *Duty Clockwise*, se utiliza una instrucción SCALE_X en el que se escala la variable *Valor absoluto* al rango [0, 100]. El valor escalado se guarda en la variable *Duty Counter-Clockwise*.

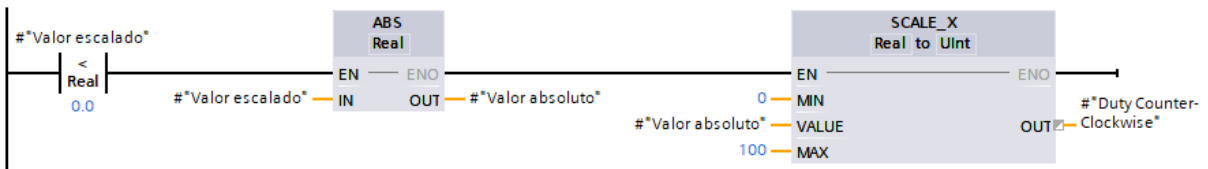


Figura 3.29. Segmento 4 de la función conversor PWM.

Una vez obtenido el valor del duty de cada generador PWM, solamente queda activar dichos generadores para que saquen la señal. Antes de generar la señal, es necesario diferenciar cuando activar un generador u otro. Al igual que antes, esto se hace diferenciando el valor de la señal *Valor escalado* según sea positivo o negativo. A parte, existe la variable *Habilitacion_PWM* para que los generadores sean activados mientras el Objeto tecnológico esté habilitado. Esta variable es necesaria ya que en las pruebas realizadas se ha observado que, durante la parada de emergencia, la salida del generador que se encontraba activo se bloqueaba a un valor del 100% sin detener la maqueta.

Para activar la salida del generador, se activan los bits de habilitación y control de cada generador PWM y se pasa el duty a cada generador, tal y como se ha indicado anteriormente.

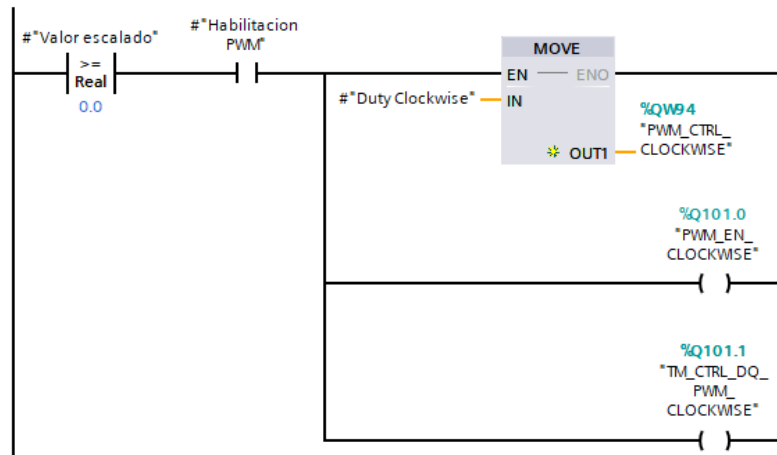


Figura 3.30. Segmento 5 de la función conversor PWM.

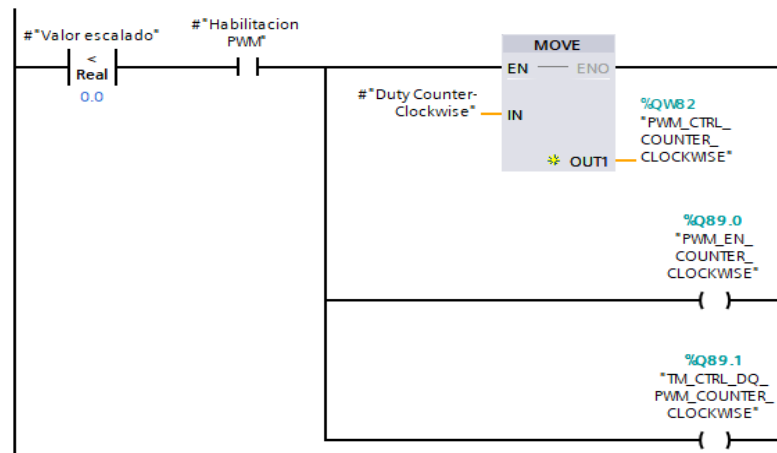


Figura 3.31. Segmento 6 de la función conversor PWM.

3.1.5.2. Función control del motor.

Esta función es la que se encarga de controlar el objeto tecnológico mediante las instrucciones Motion Control. A diferencia de la función anterior, esta función se ha creado en un bloque FB, para tener el DB correspondiente al que acceder para poder ser controlada según se requiera por otras funciones.

Las instrucciones pueden dividirse en 2 bloques:

1. Bloque de control del eje.
2. Bloque de control del movimiento.

3.1.5.2.1. Bloque de control del eje.

En este bloque se encuentran las instrucciones que habilitan, reinician y posicionan el eje correctamente para poder realizar los movimientos. Las instrucciones que constituyen este bloque son:

- La primera instrucción es MC_Power, que se encarga de habilitar y deshabilitar el eje mediante la variable *Habilitar_Eje*. Además, se muestra el estado del eje mediante la variable *Eje_Habilitado* y los posibles errores en el eje mediante las variables *Error_Eje* y *Error_Eje_ID*.
- La siguiente instrucción es la instrucción MC_Reset que se encarga de reiniciar el eje en caso de que hubiera algún tipo de error, ya que estos bloquean y deshabilitan el eje. También pueden acusarse solamente las alarmas tecnológicas. Con la variable *Tipo_Reinicio* se elige la tarea a realizar y con la variable *Ejecutar_Reinicio* se realiza.
- La última instrucción de este bloque es MC_Home, que se encarga de posicionar el eje en la posición de referencia establecida. Esta acción se realiza con la variable *Posicionar_Eje*.

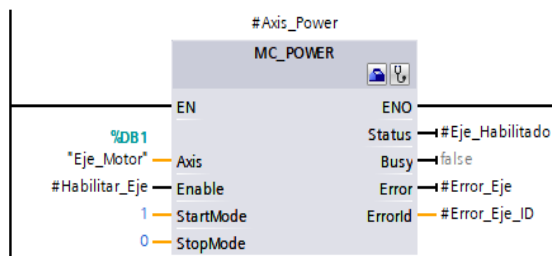


Figura 3.34. Instrucción MC_POWER del bloque de control del eje.

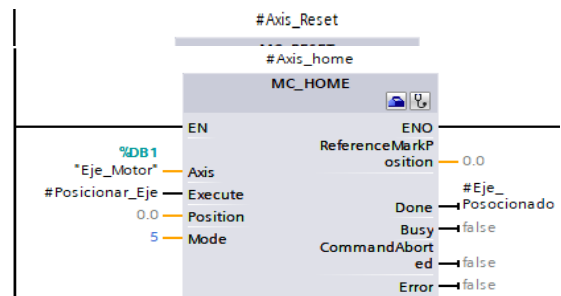


Figura 3.32. Instrucción MC_RESET del bloque de control del eje.

3.1.5.2.2. Bloque de control de movimiento.

En este bloque se encuentran las instrucciones que permiten realizar los movimientos del eje. Estas instrucciones son:

- La primera instrucción es MC_MoveJog que permite mover el eje de forma manual con las variables *Izquierda* y *Derecha*. Para habilitar esta instrucción se necesita que la variable que controla el movimiento manual esté habilitada y que las que controlan los movimientos absolutos y relativos estén desactivadas.
- La segunda instrucción, MC_Halt es complementaria de la anterior y se encarga de detener el eje mediante la variable *Parar_Eje*. Las condiciones para habilitar esta instrucción son las mismas que en la instrucción anterior.
- La tercera instrucción es MC_MoveAbsolute, que se encarga de realizar un movimiento absoluto a la posición indicada con la variable *Referencia_Posicion* cuando se activa se activa la variable *Realizar_Movimiento_Absoluto*. La instrucción se habilita cuando la variable *Habilitar_Movimiento_Absoluto* está activa y *Habilitar_Movimiento_Relativo* y *Habilitar_Movimiento_Manual* desactivadas.
- Por último, la cuarta instrucción es MC_MoveRelative, que realiza un movimiento relativo de la distancia indicada por la variable *Referencia_Posicion* cuando se activa la variable *Realizar_Movimiento_Absoluto*. Para que se habilite la instrucción, la variable *Habilitar_Movimiento_Relativo* debe estar activa, mientras

que *Habilitar_Movimiento_Absoluto* y *Habilitar_Movimiento_Manual* deben estar desactivadas.

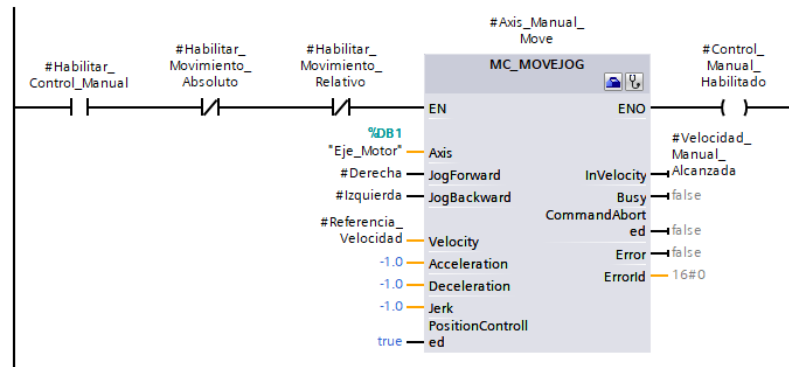


Figura 3.35. Instrucción MC_MOVEJOG del bloque de control del movimiento.

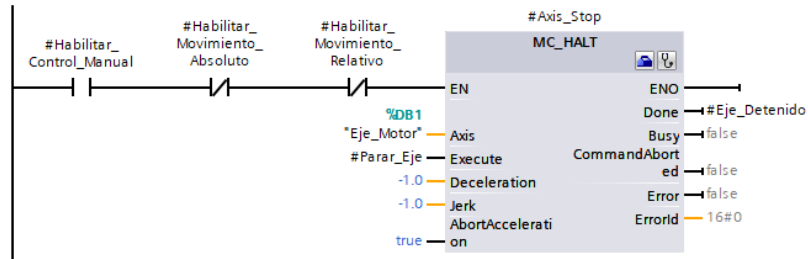


Figura 3.36. Instrucción MC_HALT del bloque de control del movimiento.

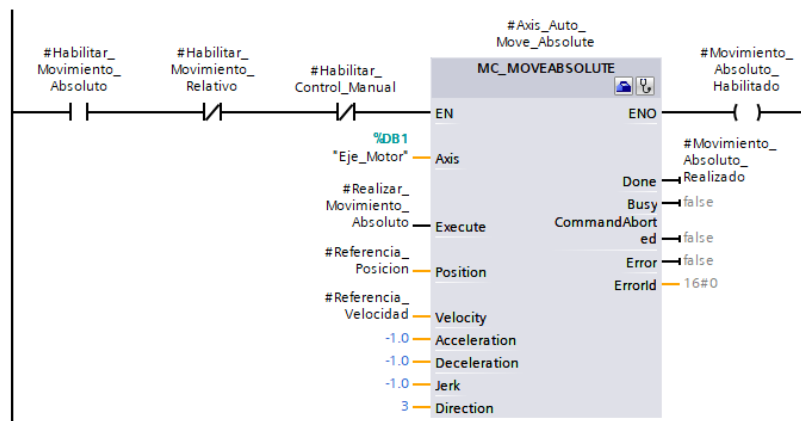


Figura 3.37. Instrucción MC_MOVEABSOLUTE del bloque de control de movimiento.

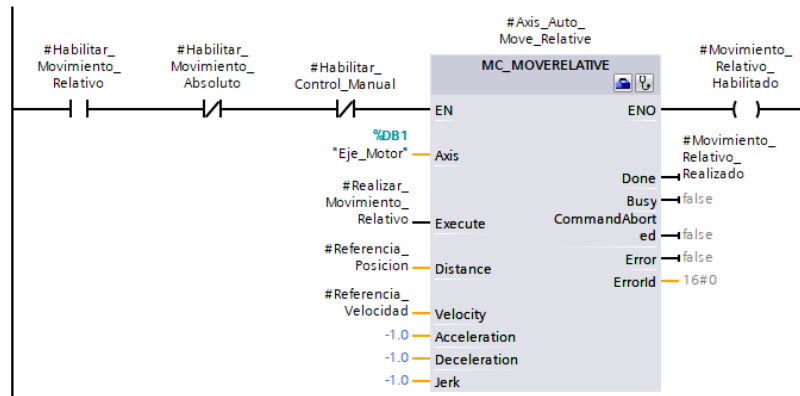


Figura 3.38. Instrucción MC_MOVERELATIVE del bloque de control de movimiento.

1.1.5.3. Función control de la maqueta.

Esta función es la que se encarga de controlar los movimientos que realiza la maqueta y, al igual que con la función control del motor, esta función se ha creado en un bloque FB, ya que es el único tipo de bloque que admite programación en SFC. La secuencia de funcionamiento es la siguiente:

Para empezar, la primera etapa, la etapa S1, sirve como reposo hasta que el usuario decida empezar con el control. Para avanzar de esta etapa a la siguiente debe activarse la transición T1, start. En dicha transición se encuentra la variable de memoria Start.

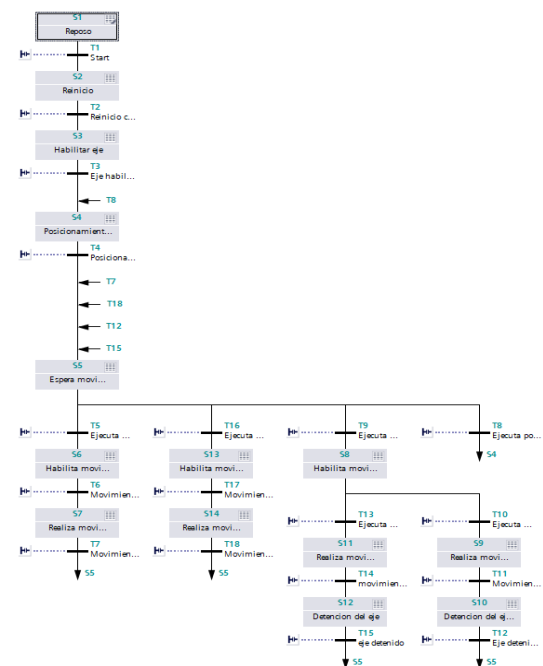


Figura 3.39. Red SFC de la función control de la maqueta.

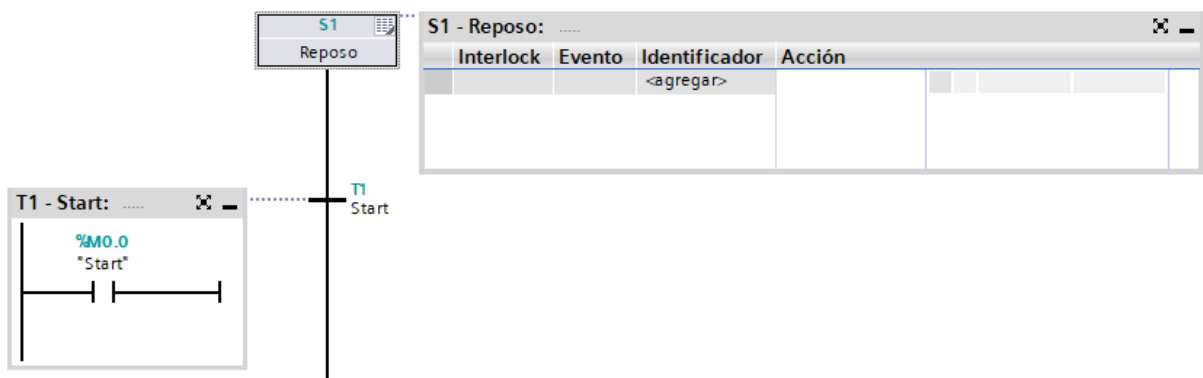


Figura 3.40. Etapa de reposo y transición de comienzo de la función control de la maqueta.

La siguiente etapa, la etapa S2, contiene el reinicio del objeto tecnológico mediante la variable *Ejecutar_Reinicio*. Esta variable tiene identificador N, lo que quiere decir es que solo está activa mientras la etapa lo esté. Para avanzar a la siguiente etapa, es necesario que la transición T2, Reinicio Completado, esté activa, hecho que se logra con la variable *Eje_Reiniciado*.

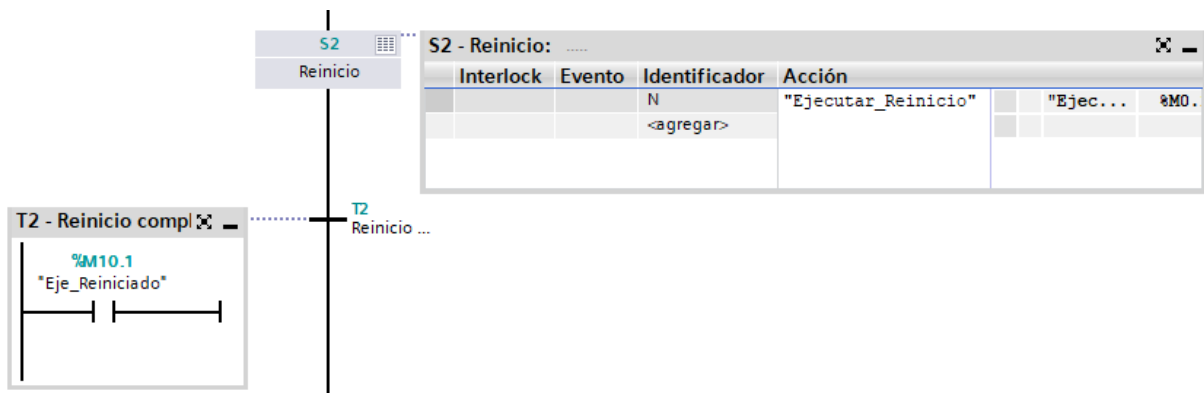


Figura 3.41. Etapa y transición de reinicio de la función control de la maqueta.

Después de la etapa de reinicio, viene la etapa en la que se habilita el objeto tecnológico, la etapa S3, con la variable *Habilitar_Eje*. Esta variable tiene identificador S, es decir, que activa la variable independientemente de si la etapa está activa o no. Una vez que la variable *Eje_Habilitado* está activa, se avanza a la siguiente etapa, la de posicionamiento en origen mediante la transición T3.

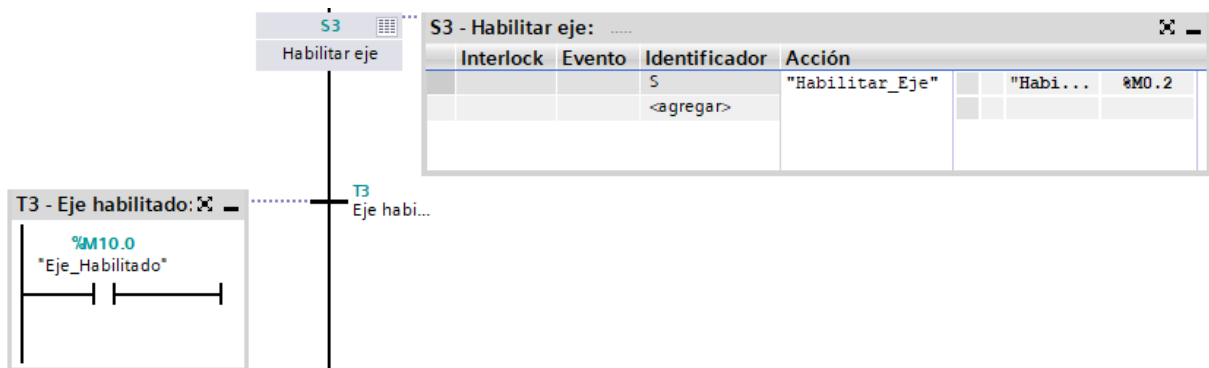


Figura 3.42. Etapa y transición de habilitación del eje de la función control de la maqueta.

En esta etapa, la S4, se le da la orden al objeto tecnológico, de posicionarse en la posición de referencia, que es detectada a través de la señal N del inductivo. Esta orden se le da mediante la variable *Ejecutar_Posicionamiento* con identificador N. Cuando la variable *Eje_Referenciado*, que se localiza en la transición T4, se activa, se pasa a la siguiente etapa, la de esperar un movimiento.

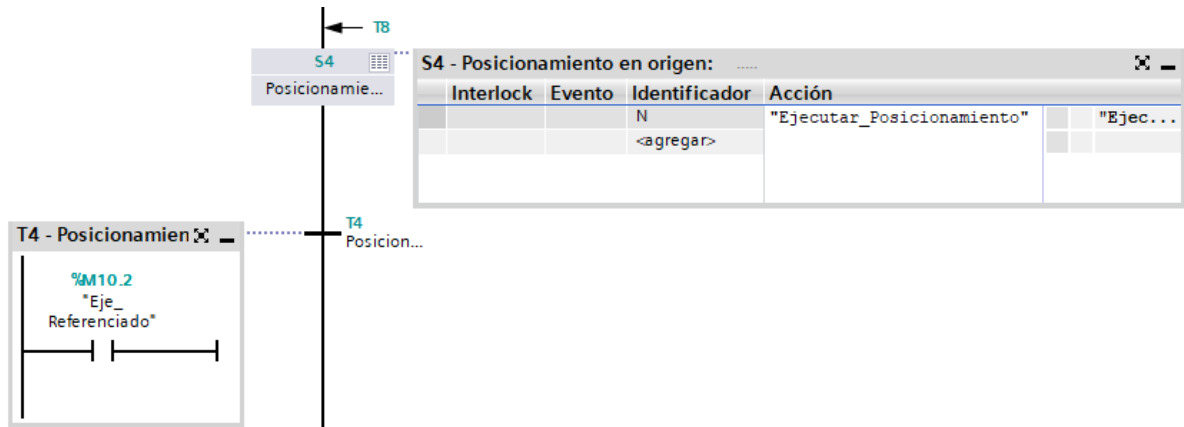


Figura 3.43. Etapa y transición de posicionamiento y referenciado de la función control de la maqueta.

Esta etapa, la S5, lo único que hace es desactivar la habilitación de los movimientos una vez que estos han terminado mediante el identificador R, y esperar a que llegue una nueva orden de movimiento. Aquí la cadena de control se ramifica en 4 ramas distintas:

- Una rama para el movimiento absoluto.
- Una rama para el movimiento relativo.
- Una rama para el movimiento manual.
- Una rama para volver a posicionar en origen.

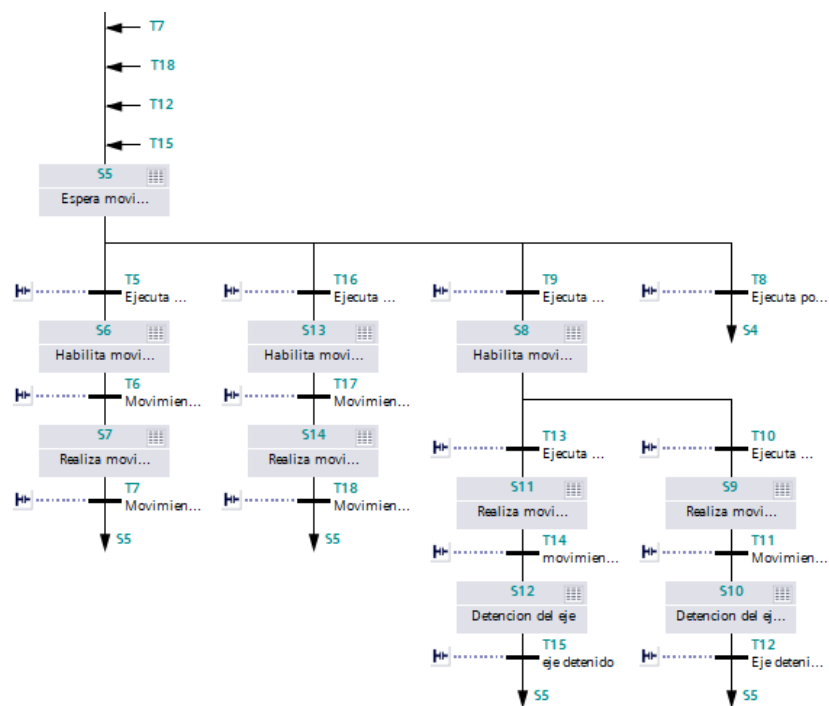


Figura 3.44. Etapa de espera y ramas de elección de movimiento de la función control de la maqueta.

Las ramas se activan cuando el usuario activa, a través de la pantalla HMI, las variables correspondientes que son:

- *Realiza_Movimiento_Absoluto*, que se localiza en la transición T5.
- *Realiza_Movimiento_Relativo*, que se localiza en la transición T16.

- *Realiza_Movimiento_Manual*, que se localiza en la transición T9.
- *Realiza_Posicionamiento*, que se localiza en la transición T8.

Si el usuario activa el movimiento absoluto, la ejecución del programa salta a la etapa de habilitar el movimiento absoluto, que es la etapa S6, a través de la variable *Habilitar_Movimiento_Absoluto*, con identificador S, y donde también se establece una referencia de posición, que en este caso es fija y no puede ser modificada por el usuario. Una vez que la instrucción del movimiento absoluto se ha habilitado, la variable *Movimiento_Absoluto_Habilitado*, situada en la transición T6, permite a la ejecución del programa pasar a la etapa de realización del movimiento absoluto.

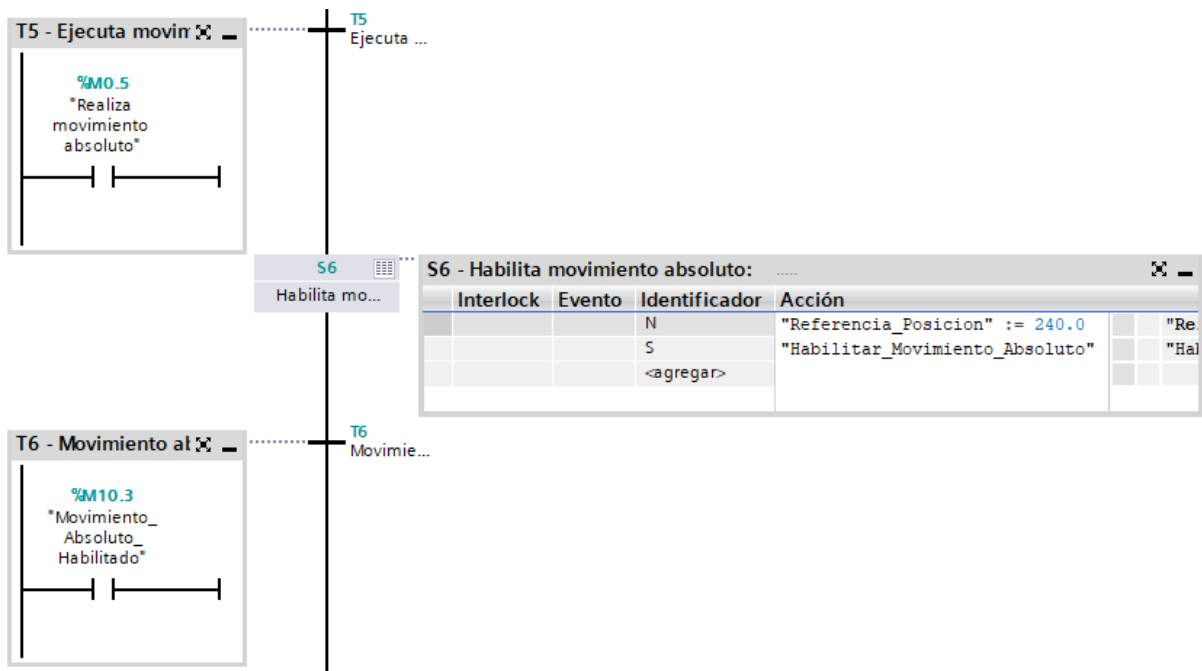


Figura 3.45. Etapa de habilitación y transición del movimiento absoluto de la función control de la maqueta.

En esta etapa, la S7, es donde se realiza directamente el movimiento, que se activa con la variable *Comenzar_Movimiento_Absoluto* cuyo identificador es el N. Una vez que el movimiento se ha realizado correctamente, se activa *Movimiento_Absoluto_Realizado*, situada en la transición T7, y la ejecución del programa vuelve a la etapa S5, la de esperar movimiento, desactivando el movimiento absoluto.

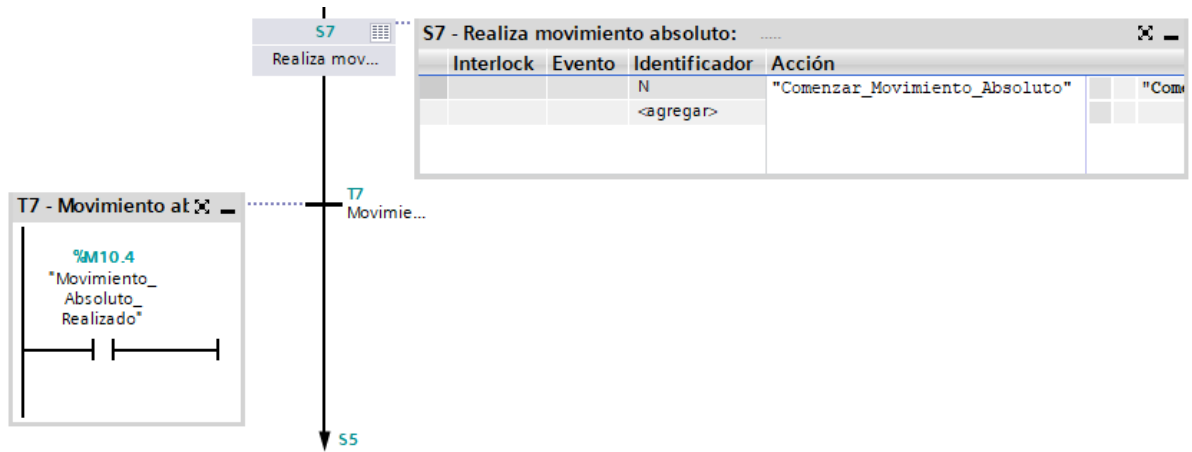


Figura 3.46. Etapa y transición de realización del movimiento absoluto de la función control de la maqueta.

En el caso que el usuario active el movimiento relativo, la cadena de ejecución es idéntica a la del movimiento absoluto, pero con las variables referidas al movimiento relativo, como puede verse en la Figura 3.47.

Si el usuario activa el movimiento manual, la cosa cambia. La parte relativa a la habilitación del movimiento es similar a la de los otros movimientos, pero el resto de la rama cambia.

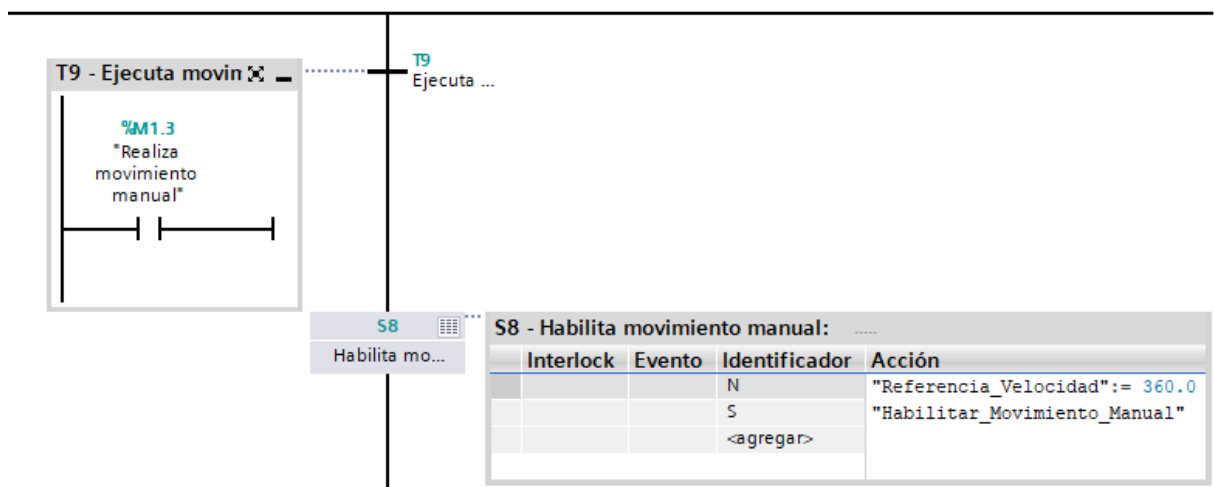


Figura 3.47. Etapa y transición que activan el movimiento manual en la función control de la maqueta.

Esta, se divide en otras 2 ramas, como se ve en la figura 3.44, una para cada sentido. Una vez que el usuario decide el sentido de giro del eje mediante las variables *Realiza_Movimiento_Izquierda* o *Realiza_Movimiento_Derecha*, que activan las transiciones T13 y T10 respectivamente, la ejecución salta a la etapa correspondiente a cada sentido, S11 para el giro hacia la izquierda y S9 para el giro a la derecha, donde se activan las variables que controlan la dirección de giro, *izquierda* y *Derecha*. Ambas variables tienen el identificador N.

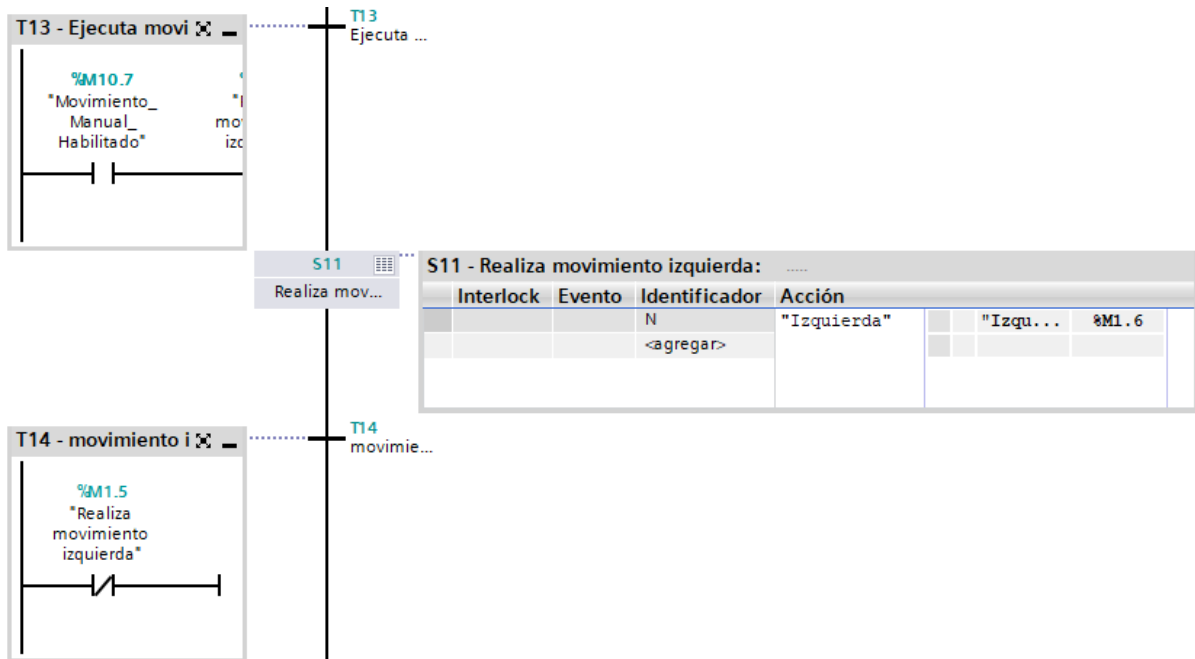


Figura 3.48. Etapa y transición que ejecuta el movimiento manual hacia la izquierda de la función control de la maqueta.

Una vez que el usuario decide detener el eje, estas ramas saltan a la etapa de frenado del eje, que son las etapas S10 y S12. Cuando el eje se detiene, la variable *Eje_Detenido* se activa, la ejecución del programa salta a través de la transición T12 o T15, según el sentido de giro, a la etapa de espera de movimiento.

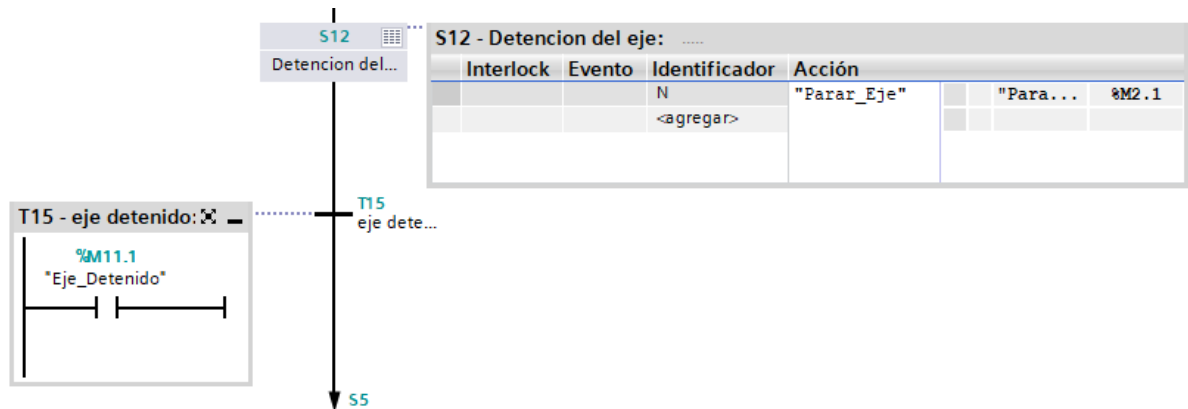


Figura 3.49. Etapa y transición de detención del eje en la función control de la maqueta.

Por último, el usuario puede volver a posicionar y referenciar el eje en origen, mediante la variable *Realiza_Posicionamiento*, tal y como puede verse en la figura 3.44. Cuando se activa esta variable la transición T8 permite que la ejecución del programa salte a la etapa de posicionamiento.

La función control de la maqueta del proyecto 3 es muy similar a esta. La parte de habilitación y configuración del eje se mantiene mientras que la parte donde se realizan los movimientos cambia a 3 movimientos absolutos, 1 por cada piso.

3.2. Proyecto 4. Control de posición lineal mediante generación de código de control con Simulink

En este proyecto, dado que la maqueta cambia, ha sido necesario hacer una serie de cambios en el proyecto de TIA Portal. Los cambios necesarios son los siguientes:

- Cambios en la configuración del encóder. Dado que en esta maqueta el encóder no dispone de ninguna señal de referencia, ha sido necesario establecer que no hay marca de referencia. A parte, se ha tenido que cambiar también la velocidad máxima de la maqueta.
- Cambios en la configuración del objeto tecnológico. Aunque el tipo de eje es el mismo porque se sigue haciendo un control de posición, en este caso, la posición ya no es angular, sino que es lineal. A parte, como el encóder no dispone de marca de referencia, ha habido que darle como referencia, una variable de entrada.
- Cambios en el programa de control. La función conversor PWM se ha mantenido, pero ha sido necesario realizar cambios en las funciones Control del motor y Control de la maqueta.

Esta última función ha sido adaptada del TFG de Miguel Sesma utilizando Simulink y exportada a TIA Portal y representa la parte más importante del presente trabajo, demostrar la versatilidad que tiene simulink para modelar y diseñar controles para sistemas muy diversos, en este caso, PLCs, y poder exportar el control diseñado a sus suites de diseño correspondientes.

A continuación, se muestran los cambios realizados en el objeto tecnológico y en la nueva función que controla el movimiento de la maqueta.

3.2.1. Cambios en la configuración del encóder.

El primer cambio que se ha hecho en la configuración del encóder es en el submenú canal 0, se ha cambiado el parámetro selección señal marca de referencia 0, de señal N del encóder incremental a ninguno.

El segundo cambio es relativo a la velocidad máxima del proyecto. En este caso, el ascensor va algo más lento que la maqueta del disco debido a la mayor masa del motor y, por tanto, mayor es el par necesario para mover la maqueta. Como se ha calculado en el Apéndice B, la velocidad máxima pasa de 150 rpm a 133,33 rpm.

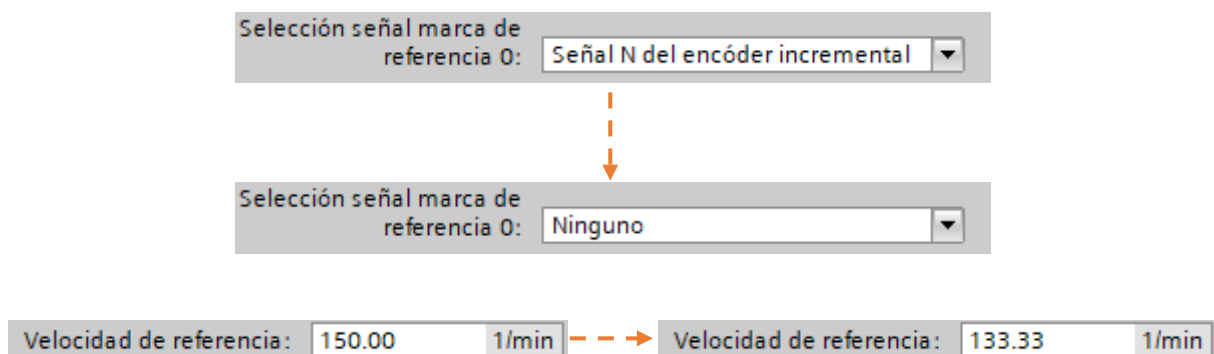


Figura 3.50. Cambios realizados en la configuración del encóder.

3.2.2. Cambios en el objeto tecnológico.

Lógicamente, el primer cambio es el tipo de eje, que pasa de rotativo a lineal. Debido a este cambio las unidades también cambian, siendo esta vez:

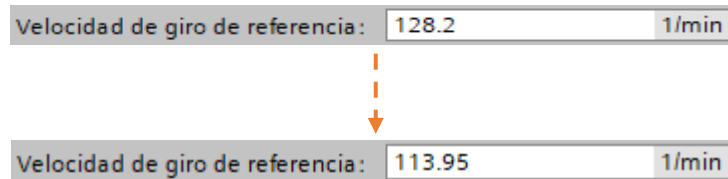
- mm para la posición.
- mm/s para la velocidad.
- Nm para el par.
- N para la fuerza.



The image shows two screenshots of a software interface for configuring measurement units. The left screenshot shows the initial configuration: 'Unidad de medida de posición: °', 'Unidad de medida de velocidad: °/s', and 'Unidad de medida de par: Nm'. An orange dashed arrow points to the right screenshot, which shows the updated configuration: 'Unidad de medida de posición: mm', 'Unidad de medida de velocidad: mm/s', and 'Unidad de medida de fuerza: N'.

Figura 3.51. Cambios realizados en la configuración de las unidades de medida del objeto tecnológico.

En la parte de Interfaz del hardware el único cambio que se ha realizado es en el apartado intercambio de datos con el accionamiento, se ha cambiado el parámetro velocidad de giro de referencia de $128,2\ rpm$ a $113,95\ rpm$.



The image shows two screenshots of a software interface for configuring the reference rotation speed. The top screenshot shows 'Velocidad de giro de referencia: 128.2 1/min'. An orange dashed arrow points down to the bottom screenshot, which shows 'Velocidad de giro de referencia: 113.95 1/min'.

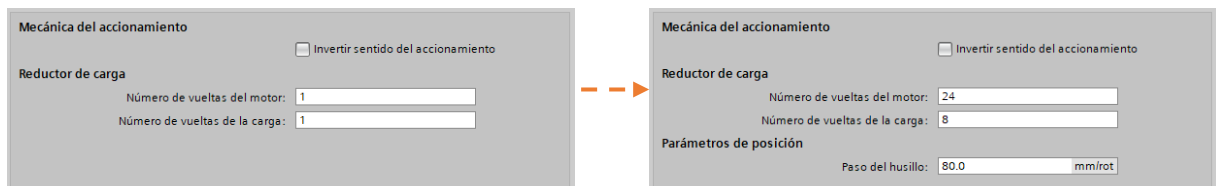
Figura 3.52. Cambios en la velocidad de giro de referencia del objeto tecnológico.

El siguiente cambio llega en el submenú parámetros avanzados, dentro del apartado de mecánica del accionamiento. Aquí, dado que la carga (la cabina del ascensor), ya no está acoplada directamente al eje de salida del motor, hay que especificar la relación de transmisión entre ambos, siendo esta:

- Número de vueltas del motor: 24.
- Número de vueltas de la carga: 8.

También se puede simplificar a 3:1, pero se ha decidido dejarlo así para que los alumnos puedan identificar correctamente a que engranaje hace referencia cada parámetro.

A parte, se ha añadido un parámetro nuevo que antes no estaba, el paso del husillo, que, como se ha mencionado en el capítulo 2, es de $80\ mm/vuelta$.



The image shows two screenshots of a software interface for configuring the drive mechanism. The left screenshot shows 'Número de vueltas del motor: 1' and 'Número de vueltas de la carga: 1'. The right screenshot shows 'Número de vueltas del motor: 24', 'Número de vueltas de la carga: 8', and a new parameter 'Paso del husillo: 80.0 mm/rot'.

Figura 3.53. Cambios en la mecánica del accionamiento del objeto tecnológico.

Dado que se ha cambiado el tipo de eje, la parte relativa a la dinámica ha cambiado también. Para el apartado preajuste de dinámica, se han tomado los siguientes valores, calculados en el Apéndice B:

- Velocidad: 50,65 mm/s.
- Tiempo de aceleración: 1,1 s.
- Tiempo de deceleración: 1,25 s.
- Tiempo de redondeo: 0,1 s. Este valor es común con la otra maqueta.

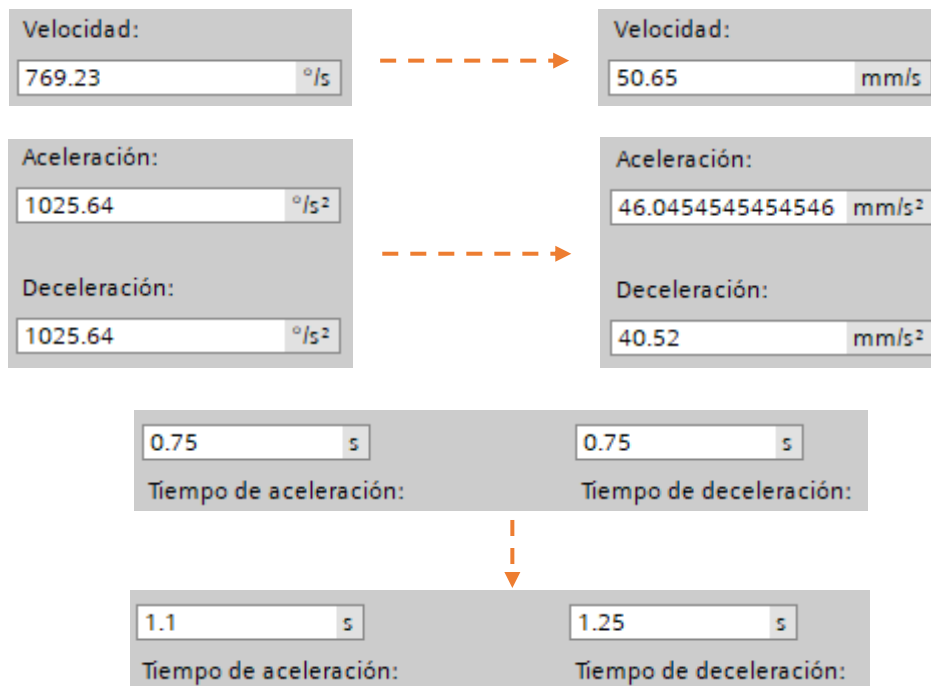


Figura 3.54. Cambios realizados en los valores nominales de la dinámica del objeto tecnológico.

A diferencia de la otra maqueta, aquí sí que se han activado los finales de carrera por hardware ya que en esta maqueta sí que se encuentra limitado el movimiento. Para ello, se han activado los finales de carrera por hardware y se ha establecido FCInf como final de carrera por hardware negativo y FCSup como final de carrera por hardware positivo. Por último, se ha configurado que el nivel de ambos sensores sea superior ya que TIA Portal establece que la señal que indica que se ha alcanzado el límite inferior se activa como False, mientras que la que indica el nivel superior se activa como True. Inicialmente el nivel el final de carrera negativo se tenía configurado como límite inferior, y esto generaba muchos problemas cuando el ascensor alcanzaba este sensor durante el posicionamiento.

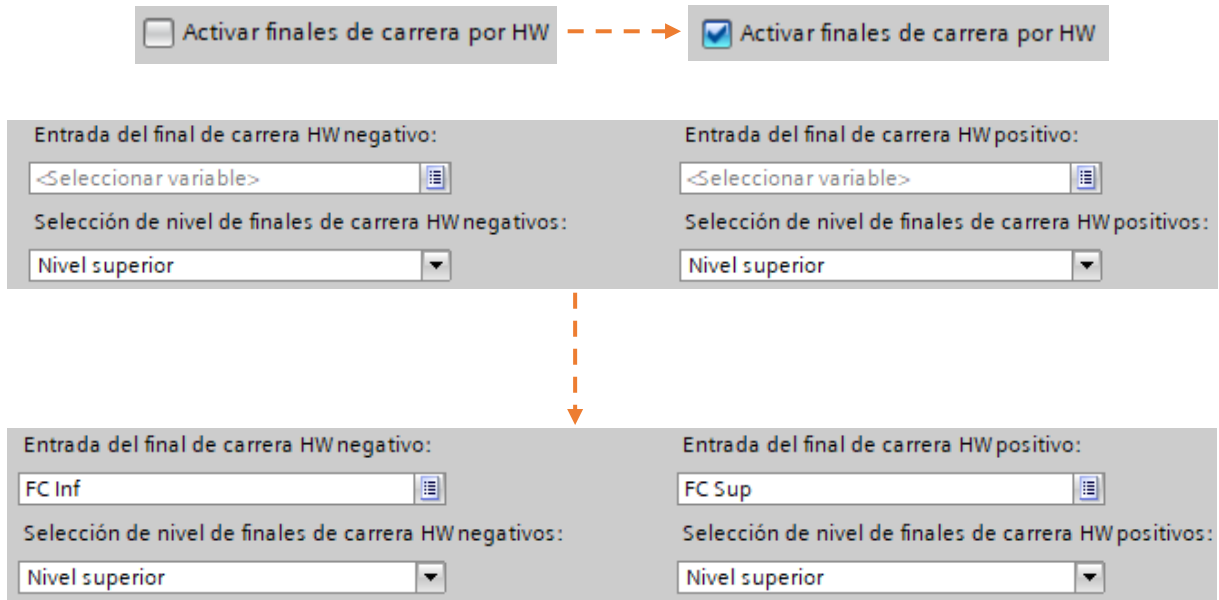


Figura 3.55. Cambios en la detección de los límites de posición del objeto tecnológico.

Respecto a los límites dinámicos, se han configurado con los valores calculados en el Apéndice B:

- Velocidad máxima: 59,26 °/s.
- Tiempo de aceleración: 0,25 s.
- Tiempo de deceleración: 0,25 s.
- Tiempo de redondeo: 0,05 s.

Los valores de los tiempos de aceleración, deceleración y redondeo son los mismos que para la otra maqueta.

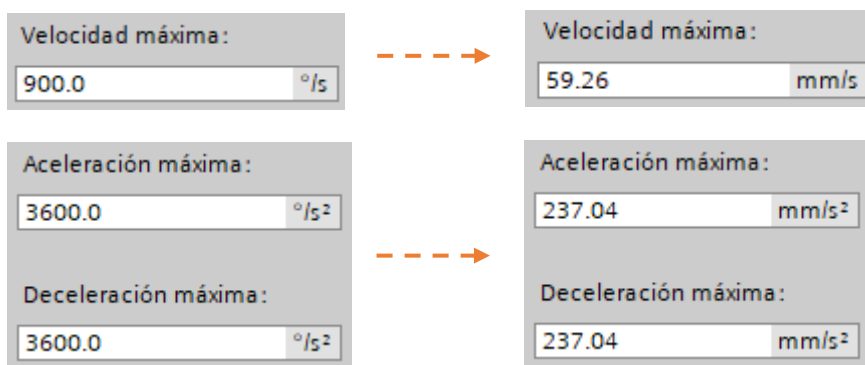


Figura 3.56. Cambios en los límites dinámicos del objeto tecnológico.

Respecto al referenciado, se ha cambiado el modo de referenciado debido a la falta de señal N del encóder. A diferencia de la otra maqueta donde en el referenciado se usaba la marca cero a través de telegrama PROFIdrive, aquí se usa una marca de referencia a través de una entrada digital, en este caso FCB. A esta entrada se le asigna un nivel superior y se le activa la inversión de sentido en final de carrera por hardware. Respecto a los sentidos de aproximación y referenciado se ha decidido que sea negativo, ya que, de esta manera, el ascensor bajará. Se ha decidido que el ascensor baje durante el posicionamiento ya que es más probable que se encuentre por encima de la marca de referencia, situada en la parte baja

del ascensor, que por debajo. Como velocidades de referenciado y aproximación se han tomado las calculadas en el Apéndice B, 22,5 mm/s para ambas.

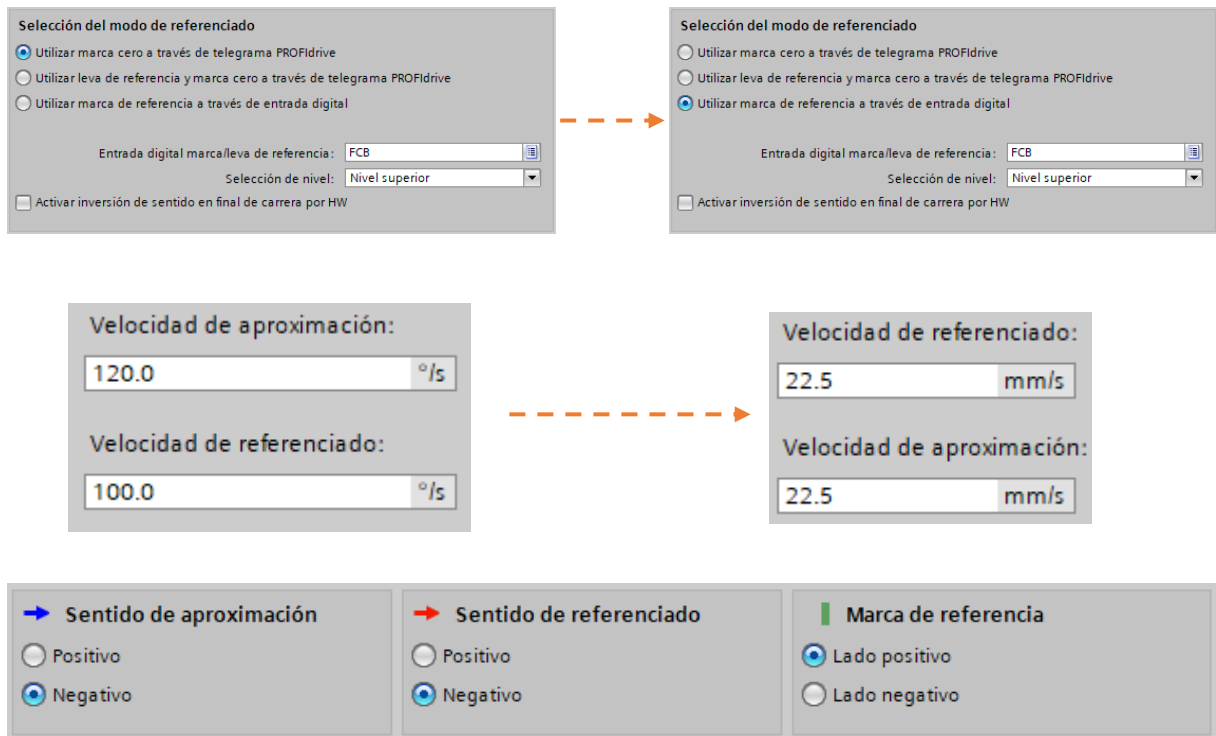


Figura 3.57. Cambios en el referenciado del objeto tecnológico.

En el siguiente apartado, la vigilancia de posición se ha tomado 1 mm como ventana de posicionamiento y se ha mantenido los 10 segundos como tiempo de tolerancia y de tiempo mínimo de permanencia, para darle tiempo al ascensor a que se detenga por completo. En esta maqueta, la posición es más precisa que en la maqueta del disco, por lo que se puede disminuir la ventana de posicionamiento. Se ha estado probando esta configuración y no ha dado ningún problema.



Figura 3.58. Cambio en la vigilancia de posicionamiento del objeto tecnológico.

En el penúltimo apartado, la señal de velocidad cero, se ha decidido que la ventana de velocidad cero sea de 10 mm/s y que el tiempo de permanencia siga siendo de 2 segundos. Estos valores han sido elegidos arbitrariamente y, al igual que con la otra maqueta, durante las pruebas realizadas se ha encontrado que no generaban ningún error.

El resto de los parámetros que no han sido mencionados, se han mantenido entre la configuración de una maqueta y otra.



Figura 3.59. Cambio en la ventana de velocidad cero del objeto tecnológico.

3.2.3. Cambio en la función Control del motor.

El primer cambio ha consistido en la eliminación de las instrucciones MC_MoveJog, MC_Halt y MC_MoveRelative y sus variables asociadas ya que en este proyecto solamente se ha necesitado realizar movimientos absolutos.

El siguiente cambio que se ha realizado en esta función ha sido respecto a la velocidad de la maqueta. Mientras que, en la maqueta del disco de inercia, la velocidad puede ser controlada por el usuario, en esta maqueta la velocidad es la nominal que se ha calculado en el Apéndice B e introducido en la configuración del objeto tecnológico. Por tanto, se ha cambiado el valor de la velocidad en la instrucción MC_MoveAbsolute de la variable *Referencia_Velocidad* al valor fijo -1.0 para que la instrucción tome el valor nominal de velocidad en la realización del movimiento.

3.2.4. Generación de código de control con Simulink.

En este proyecto, la maqueta puede trabajar de dos formas distintas, con memoria o sin memoria.

En el modo memoria, el usuario puede seleccionar varios pisos y el ascensor hará parada en ellos, pero siempre de forma lineal. Es decir, si el usuario llama al piso 2, luego al 0 y luego al 1, el ascensor para, si es posible, primero en el piso 1, luego continúa al piso 2 y luego baja al piso 0, parando antes en el piso 1 si no lo ha hecho en la subida.

En el modo sin memoria, el ascensor solamente realiza un movimiento. Si el usuario llama al piso 2, el ascensor únicamente sube al piso 2 y el usuario no puede llamar a otro piso hasta que el ascensor no se detenga en el piso 2.

Como se ha mencionado en la introducción de este capítulo, esta función ha sido desarrollada íntegramente en Simulink a partir del TFG de Miguel Sesma y exportada a TIA Portal para ser integrada en el proyecto.

La función está dividida en otras 2 funciones:

- Asignación de Variables.
- Asignación del SetPoint.

Para trabajar mejor, se han copiado los bloques Asignación de Variables y Asignación del SetPoint a un proyecto nuevo en blanco y se han editado por separado, Figura 3.60 y Figura 3.61 respectivamente.

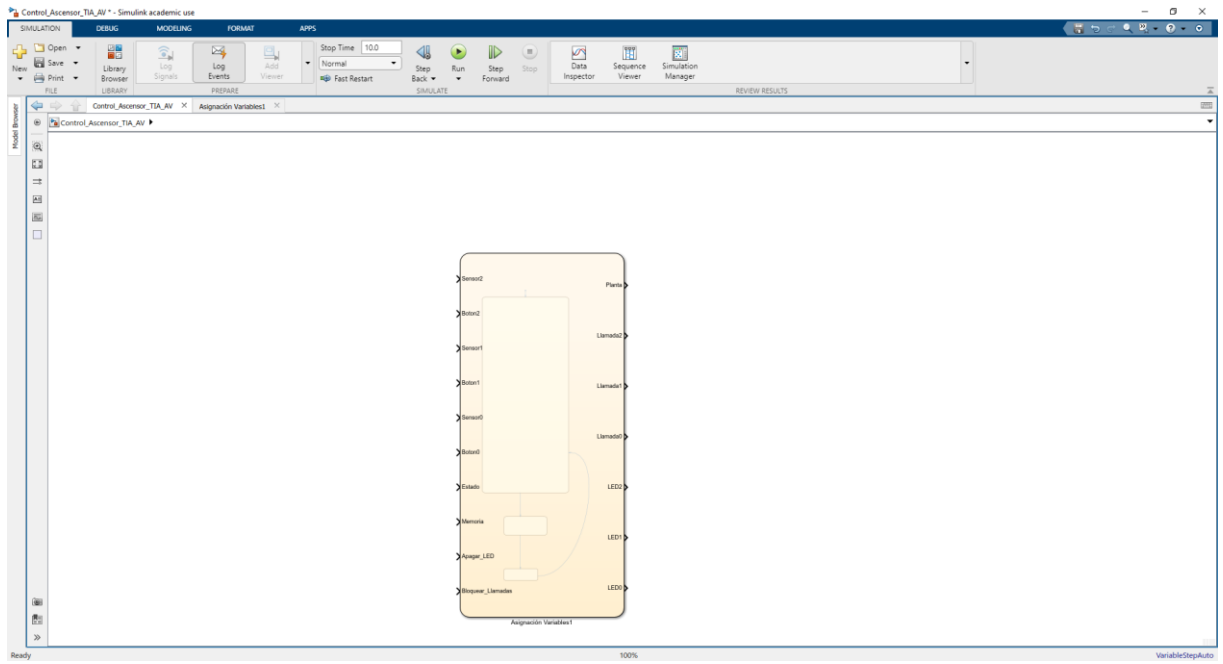


Figura 3.60. Proyecto creado para exportar el bloque de Asignación de variables.

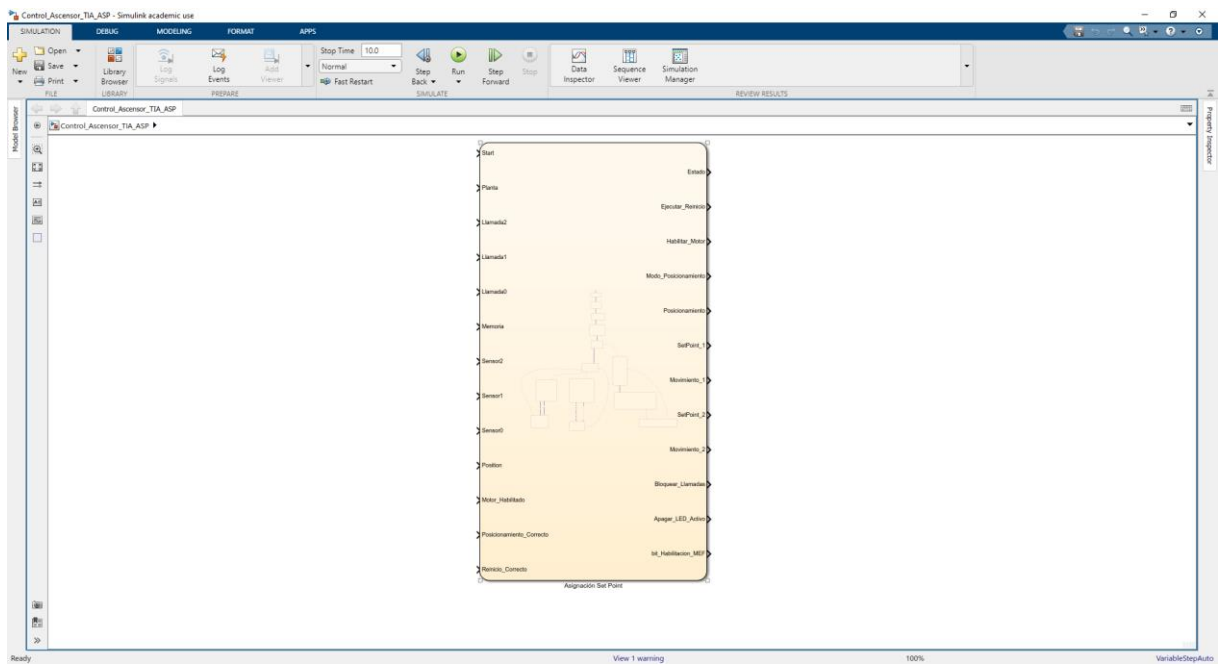


Figura 3.61. Proyecto creado para exportar el bloque de Asignación del SetPoint.

3.2.4.1. Función Asignación de Variables.

Esta función, Figura 3.62, sirve de función auxiliar a la función Asignación del SetPoint y está formada por una máquina de estados que se encarga de controlar las llamadas de la maqueta. Las llamadas son las peticiones que realiza el usuario para que el ascensor se posicione al piso requerido.

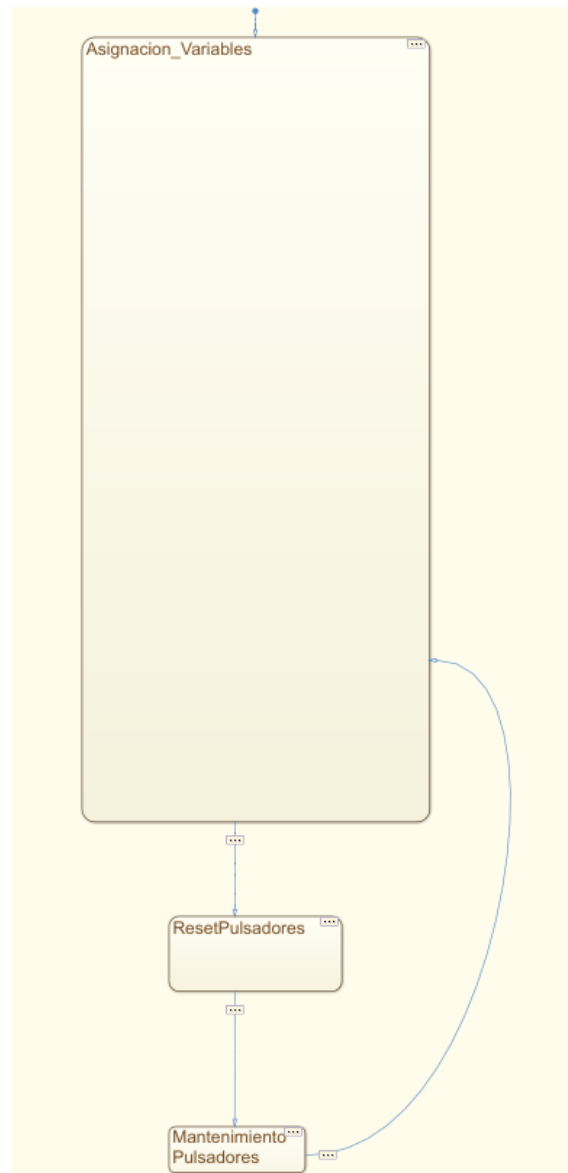


Figura 3.62. StateFlow de la función Asignación de variables.

Para empezar, Figura 3.63, la habilitación de la máquina de estados se encuentra controlada por la función Asignación del SetPoint mediante la variable *bit habilitación MEF AV*. Cuando la máquina de estados se habilita, se reinician las llamadas para evitar posibles errores.

A continuación, cuando uno de los finales de carrera se activa, mediante las variables *sensorX* donde X es el número de piso, se asigna el valor correspondiente al piso a la variable *planta*. Además, si la máquina de estados de la función Asignación del SetPoint, indicado por

la variable *estado*, se encuentra en el estado 7, 9 o 12, que son los estados ParadoPlantaMemoria, ParadoPlantaMemoria1 y ParadoPlantaNoMemoria respectivamente, se desactiva la llamada correspondiente al piso y, dependiendo de si se encuentra encendido y si se le da la orden mediante la variable *Apagar_LED* se apaga también el LED correspondiente a ese piso. Por último, si el modo de trabajo del ascensor es sin memoria, se bloquea la posibilidad de activar de nuevo la variable llamada correspondiente a ese piso hasta que el ascensor cambie de piso.

Después, y si las llamadas no están bloqueadas mediante las variables *Bloquear_Llamada_X* o *Bloquear_Llamadas*, se activan la llamada y el LED correspondiente a cada piso cuando el usuario activa el botón correspondiente de ese piso.

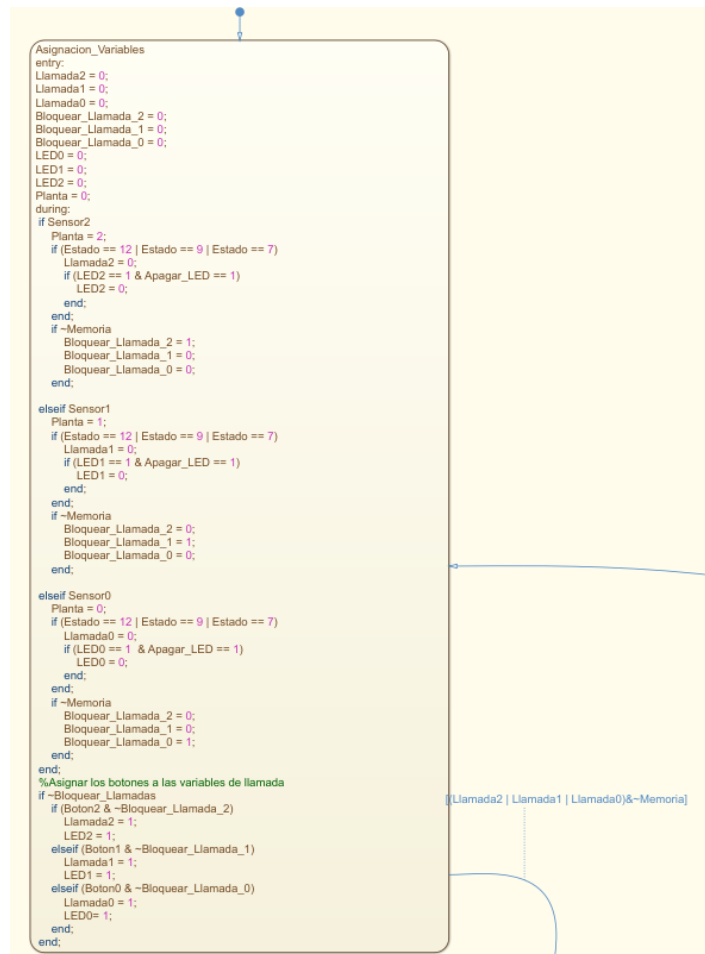


Figura 3.63. Estado *Asignación_Variables* del StateFlow de la función *Asignación de variables*.

Finalmente, si el modo de trabajo del ascensor es sin memoria, Figura 3.64, se reinician las llamadas pasados unos segundos.

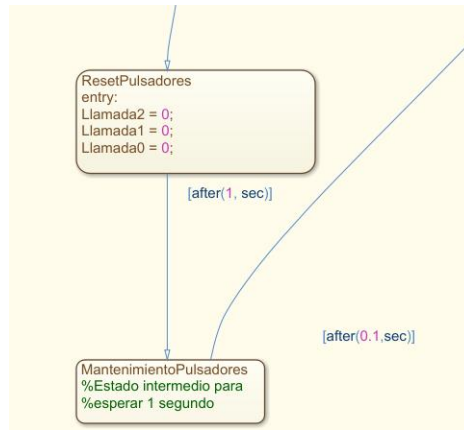


Figura 3.64. Estados que reinician las llamadas del StateFlow de la función Asignación de variables.

3.2.4.2. Función Asignación del SetPoint

Esta es la función que controla el movimiento de la maqueta y, al igual que la función Asignación de variables, está formada por una máquina de estados, Figura 3.65.

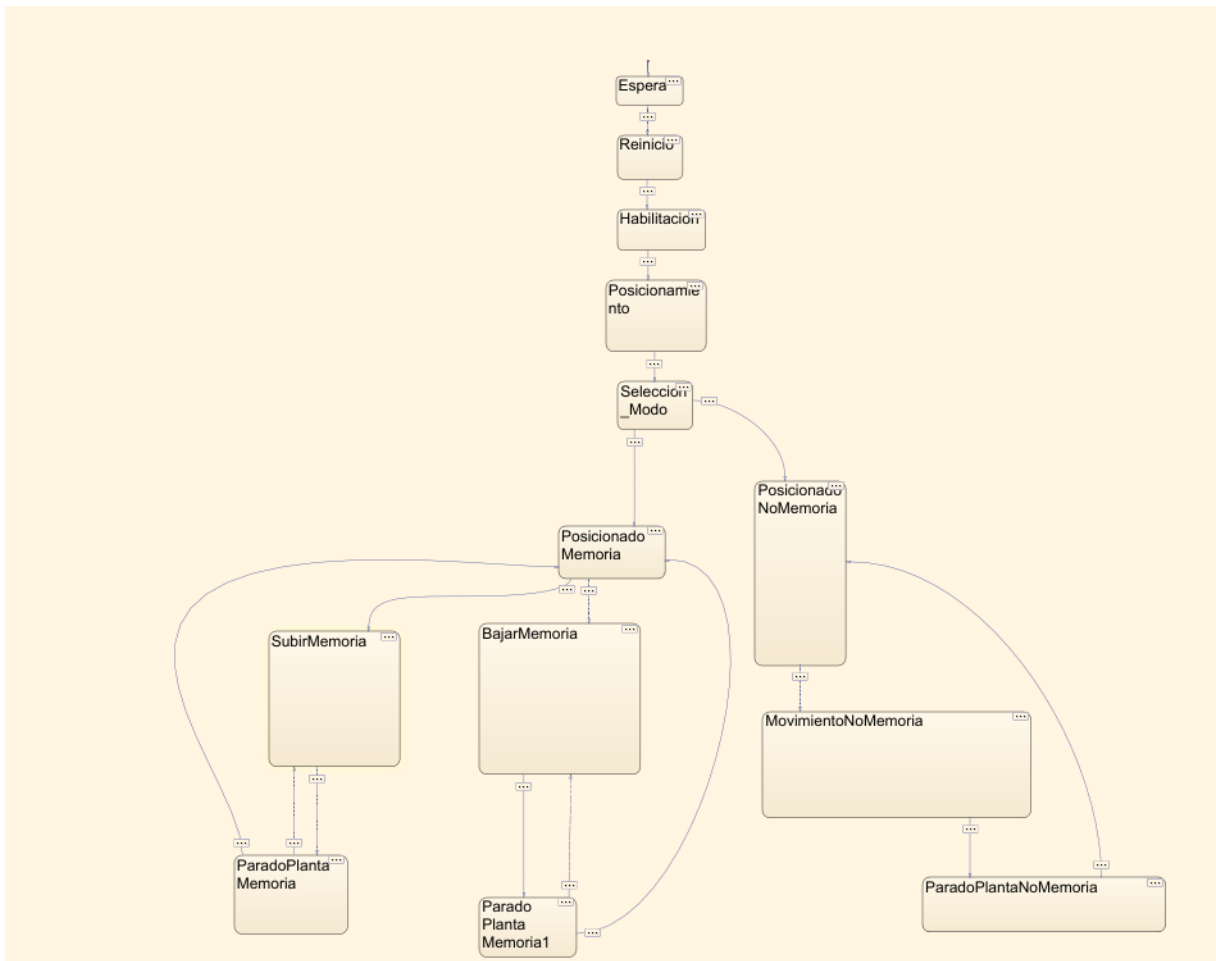


Figura 3.65. StateFlow de la función Asignación del SetPoint.

Inicialmente, la máquina se encuentra en el estado Reposo, Figura 3.66. hasta que el usuario decida activar la máquina mediante la variable *Start*.

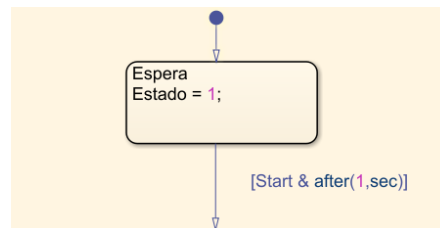


Figura 3.66. Estado Espera del StateFlow de la función Asignación del SetPoint.

Cuando la maquina se inicia, se pasa al estado Reinicio, Figura 3.67, donde se reinicia el objeto tecnológico con *Ejecutar_Reinicio*, por si hubiera habido algún error anterior. Tras recibir la confirmación de que el reinicio se ha realizado correctamente, *Reinicio_Correcto*, la máquina pasa al estado Habilitación, donde se habilita, el eje con *Habilitar_Motor*.

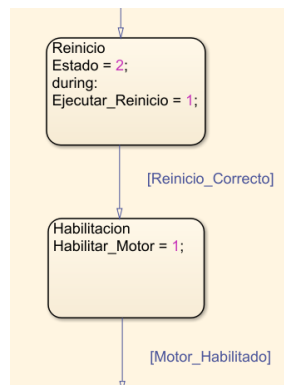


Figura 3.67. Estados Reinicio y Habilitación del StateFlow de la función Asignación del SetPoint.

Una vez que se activa *Motor_Habilitado*, significa que el eje ya está habilitado y listo para moverse, por lo que se posiciona en la posición de referencia, Figura 3.68, que se encuentra en el piso 0, activando *Posicionamiento*. Si el ascensor ya se encontraba en el piso 0, se establece *Modo_Posicionamiento* a 0 mientras que, si el ascensor no se encuentra a la altura de sensor 0, *Modo_posicionamiento* vale 5. La diferencia entre el modo 0 y el modo 5, es que en el modo 0 simplemente se ajusta *posición* a 0,0 independientemente de donde se encuentre el eje, sin realizar ningún tipo de movimiento, mientras que en el modo 5 el eje realiza primero un movimiento hasta la posición de referencia indicada en la configuración del objeto tecnológico y luego establece *posición* a 0,0.

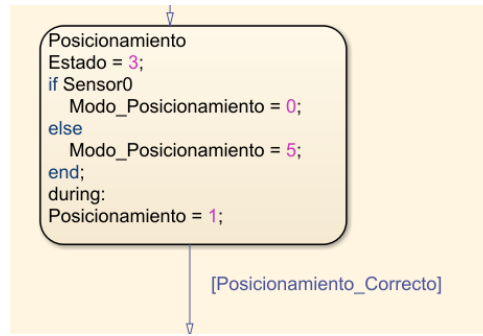


Figura 3.68. Estado Posicionamiento del StateFlow de la función Asignación del SetPoint.

Si el eje se ha posicionado correctamente, indicado con *Posicionamiento_Correcto*, la máquina de estados pasa al estado de decisión, el estado Selección_Modo, Figura 3.69, donde el usuario dispone de un tiempo, *Tiempo_Decision*, para elegir el modo de funcionamiento, con memoria o sin memoria. Si pasado ese tiempo el usuario no ha elegido, el programa salta automáticamente al modo sin memoria, el estado PosicionadoNoMemoria. Si, por el contrario, el usuario activa el modo memoria, el programa pasa al siguiente estado, el estado PosicionadoMemoria. Tras haber seleccionado el modo, se habilita la máquina de estados de la función asignación de variables mediante la variable *Bit_Habilitacion_MEF_AV*.

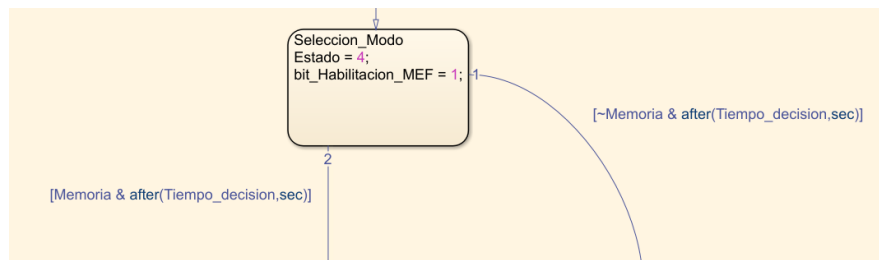


Figura 3.69. Estado Selección_Modo del StateFlow de la función Asignación del SetPoint.

El estado PosicionadoMemoria, Figura 3.70 es el estado de espera del modo memoria. Cuando llega una petición de movimiento a través de las llamadas, la ejecución del programa salta al estado SubirMemoria o al estado BajarMemoria, según el tipo de movimiento que se requiera.

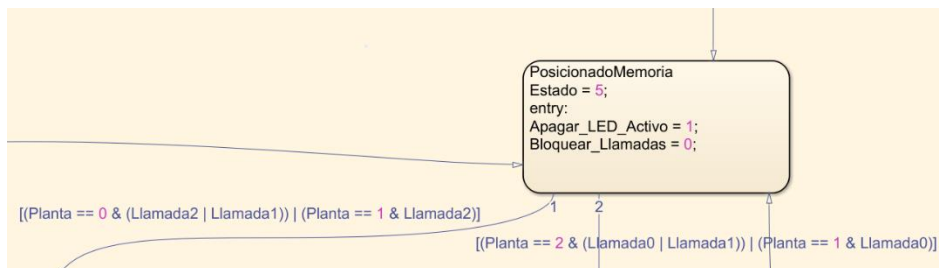


Figura 3.70. Estado PosicionadoMemoria del StateFlow de la función Asignación del SetPoint.

El estado SubirMemoria, Figura 3.71, es donde el ascensor realiza un movimiento ascendente con memoria. Si el piso seleccionado es el 1, indicado a través de *Llamada 1*, se establece *Referencia_Posicion* a 240,0, se desactiva *movimiento_1*, se activa *movimiento_2*

y se establece *PObjetivo* a 1. Por el contrario, si el piso seleccionado es el 2, indicado a través de *Llamada 2*, se establece *Referencia_Posicion* a 480,0, se activa *movimiento_1*, se desactiva *movimiento_2* y se establece *PObjetivo* a 2.

Si mientras el ascensor está subiendo al piso 2 el usuario selecciona el piso 1, y *posición* vale menos que 208,33, se ajustan las variables anteriores para realizar una parada en el piso 1, pero manteniendo *llamada 2* activa. Si, por el contrario, el ascensor ya ha rebasado esa posición o se le llama desde el piso 0, continúa hasta el piso 2, manteniendo las llamadas correspondientes activas.

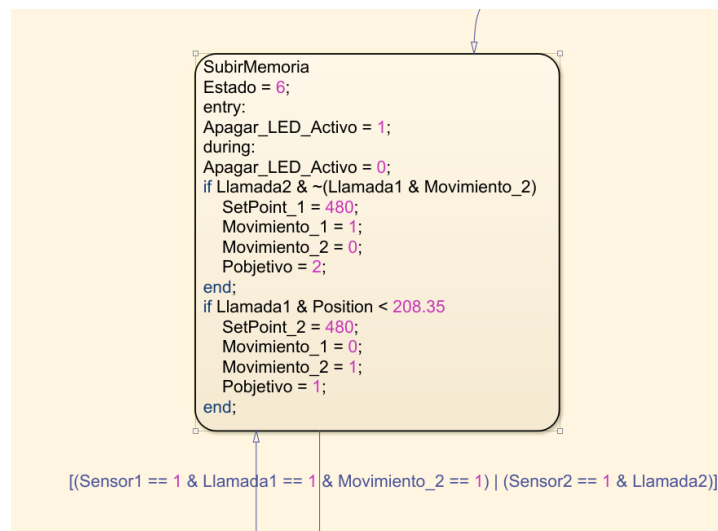


Figura 3.71. Estado *SubirMemoria* del StateFlow de la función *Asignación del SetPoint*.

Una vez que el ascensor alcanza el piso objetivo, la máquina de estados pasa al estado *ParadoPlantaMemoria*, Figura 3.72, donde se realiza una espera, cuya duración viene dada por *Tiempo_Parada*. Si el ascensor tiene ordenes de subir activas, la maquina salta de nuevo al estado 6. Si, por el contrario, ya ha terminado con las ordenes de subir la máquina salta al estado *PosicionadoMemoria*.

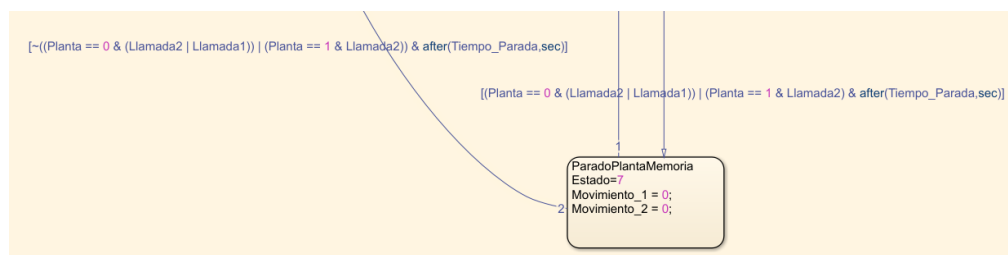


Figura 3.72. Estado *ParadoPlantaMemoria* del StateFlow de la función *Asignación del SetPoint*.

El estado *BajarMemoria*, Figura 3.73, es similar al estado *SubirMemoria* y es el estado donde el ascensor realiza un movimiento descendiente con memoria. Si el piso seleccionado es el 0, indicado a través de *llamada 0*, se establece *Referencia_Posicion* a 0,0, se activa *movimiento_1*, se desactiva *movimiento_2* y se establece *PObjetivo* a 0. Por el contrario, si el piso seleccionado es el 1, se establece *Referencia_Posicion* a 240,0, se desactiva *movimiento_1*, se activa *movimiento_2* y se establece *PObjetivo* a 1.

Al igual que en el estado SubirMemoria, si mientras el ascensor está bajando al piso 0 el usuario selecciona el piso 1, y *posición* vale más que 271,67, se ajustan las variables anteriores para realizar una parada en el piso 1, pero manteniendo *llamada 0* activa. Si, por el contrario, el ascensor ya ha rebasado esa posición o se le llama desde el piso 2, continúa hasta el piso 0, manteniendo las llamadas correspondientes activas.

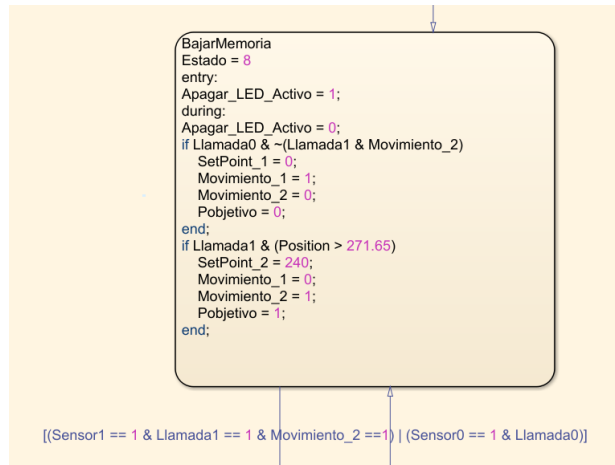


Figura 3.73. Estado BajarMemoria del StateFlow de la función Asignación del SetPoint.

Cuando el ascensor ha alcanzado el piso objetivo, la máquina de estados pasa al estado ParadoPlantaMemoria1, Figura 3.74, donde se realiza una espera, con una duración indicada por *Tiempo_Parada*. Si el ascensor tiene ordenes de bajar activas, la maquina salta de nuevo al estado 8. Si, por el contrario, ya ha terminado con las ordenes de subir la máquina salta al estado 5.

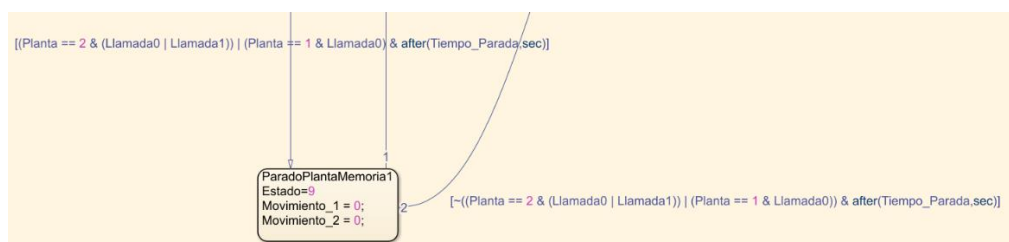


Figura 3.74. Estado ParadoPlantaMemoria1 del StateFlow de la función Asignación del SetPoint.

En el estado PosicionadoNoMemoria, Figura 3.75, se preparan los movimientos sin memoria del ascensor. En este caso, como solamente se puede realizar un movimiento, no existe una distinción de estados según la dirección del ascensor.

Cuando se selecciona un piso, se ajustan las variables *Referencia_Posicion* y *PObjetivo* al piso correspondiente y se bloquean las llamadas.

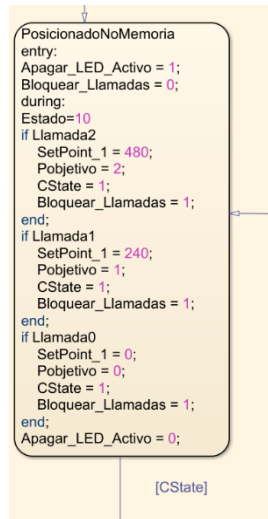


Figura 3.75. Estado PosicionadoNoMemoria del StateFlow de la función Asignación del SetPoint.

Tras haber ajustado las variables para realizar al movimiento, la máquina pasa al estado MovimientoNoMemoria, Figura 3.76, donde se activa *Movimiento_1* para comenzar el movimiento. Cuando el movimiento termina al activarse el sensor correspondiente de cada piso, *Movimiento_1* se desactiva y la máquina pasa al estado ParadoPlantaNoMemoria.

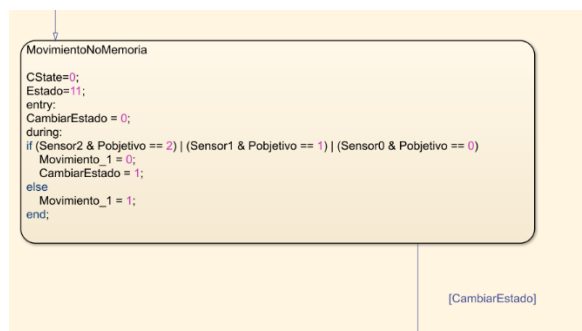


Figura 3.76. Estado MovimientoNoMemoria del StateFlow de la función Asignación del SetPoint.

La función del estado ParadaPlantaNoMemoria, Figura 3.77, es similar a la de los estados ParadoPlantaMemoria y ParadoPlantaMemoria1 del modo con memoria, simplemente es una espera, cuya duración viene dada por *Tiempo_Parada*. Además, pasado el tiempo de espera se apaga el LED correspondiente al piso en el que se ha detenido el ascensor con *Apagar_LED* y el programa salta de nuevo al estado PosicionadoNoMemoria, esperando que se le indique un nuevo movimiento.

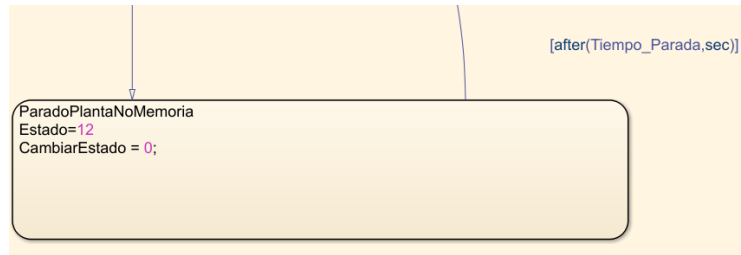


Figura 3.77. Estado ParadoPlantaNoMemoria del StateFlow de la función Asignación del SetPoint.

3.2.4.3. Exportado de las funciones a TIA Portal.

Una vez que se han editado las funciones para adaptarlas a este trabajo, se han exportado para ser utilizadas en TIA Portal.

Antes de exportarlas ha sido necesario ajustar los parámetros para poder exportar a TIA Portal, ya que Simulink permite exportar el código creado a C/C++ y a PLCs, aceptando esta última categoría diversos fabricantes como Siemens, Rockwell o Phoenix Contact, como puede verse en la Figura 3.78. Para poder configurar las opciones de exportado es necesario ir a la siguiente ruta:

Clic derecho sobre el bloque >> PLC Code >> Options.

Dentro de esta ruta aparece el cuadro de configuración del exportado del bloque, Figura 3.79. En el parámetro Target IDE se ha elegido la opción Siemens SIMATIC Step 7. Además, en el parámetro Target IDE Path se ha elegido la ruta donde se va a guardar el archivo exportado.

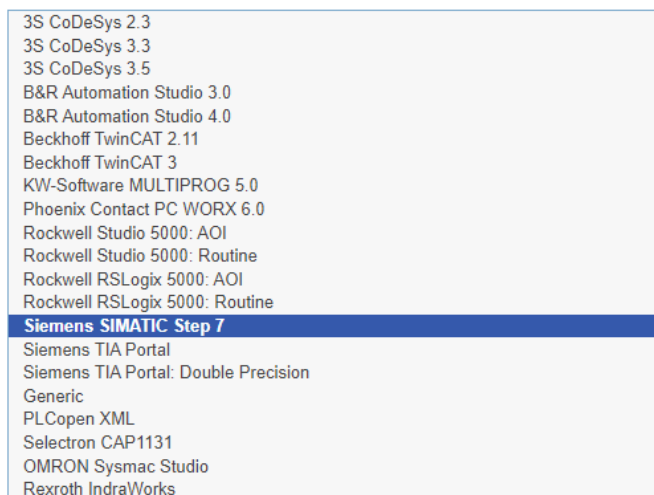


Figura 3.78. Lista de fabricantes y lenguajes de PLCs admitidos por MATLAB.

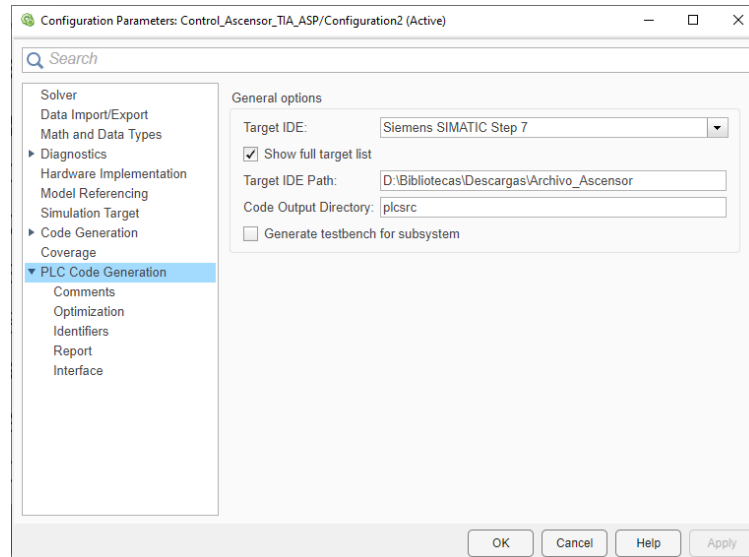


Figura 3.79. Menú de configuración de los parámetros de exportación de MATLAB.

Antes de exportar se ha realizado una prueba para comprobar la compatibilidad del bloque con TIA Portal. Esto se hace en la siguiente ruta:

Clic derecho sobre el bloque >> PLC Code >> Check subsystem compatibility.

Una vez que ha pasado la prueba, ya se puede exportar. Para hacer esto hay que ir a la siguiente ruta:

Click derecho sobre el bloque >> PLC Code >> Generate code for subsystem.

Cuando se ha exportado, se crea un fichero con extensión .scl en la dirección introducida en el cuadro de opciones.

Tras haber repetido este proceso con ambos bloques, se obtienen los siguientes ficheros:

- Control_Ascensor_TIA_AV.scl
- Control_Ascensor_TIA_ASP.scl

El siguiente paso es importar estos ficheros a TIA Portal. Dentro del proyecto 4, en el árbol del proyecto situado a la izquierda y dentro del apartado del PLC, aparece la opción, fuentes externas que permite importar ficheros con código externo a TIA Portal. Tras haber importado los ficheros anteriores, todavía no están disponibles para ser utilizados ya que es necesario generarlos. Esto se hace de la siguiente manera:

Clic derecho sobre el fichero >> Generar bloques a partir de fuente.

Cuando se han generado los bloques de ambos ficheros, aparecen los bloques FB que contiene las siguientes funciones:

- Control_Ascensor_TIA_AV.
- Control_Ascensor_TIA_ASP.
- PLC_CODE_TIMER.

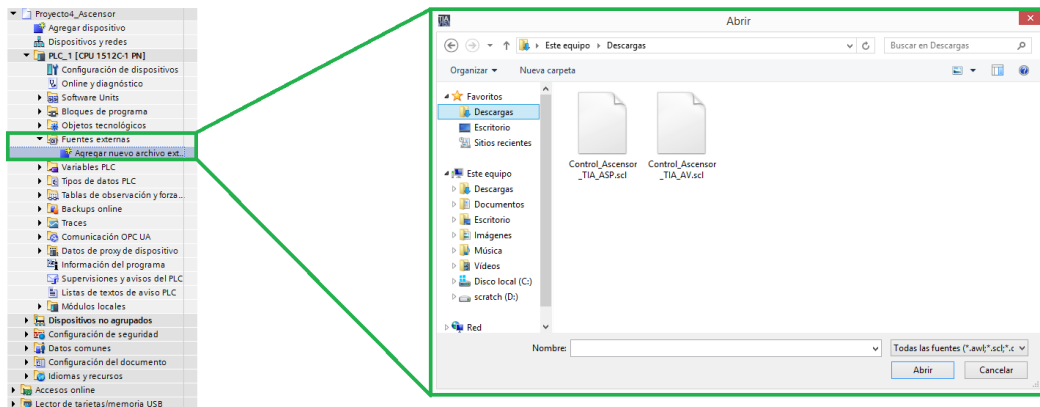


Figura 3.80. Ubicación de la opción de importar fuentes externas en TIA Portal.

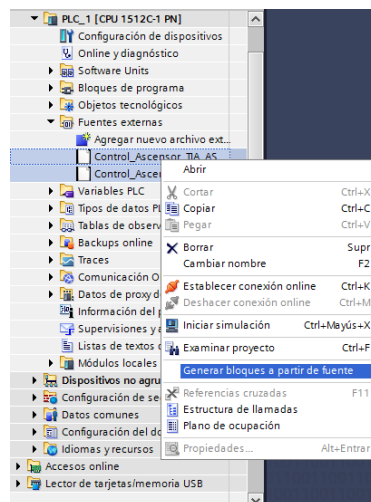


Figura 3.81. Ubicación de la opción generar bloques a partir de fuente en TIA Portal.

Las dos primeras funciones son las creadas anteriormente mientras que la última, PLC_CONDE_TIMER, se ha creado automáticamente y contiene los temporizadores utilizados en las funciones anteriores.

Tras haber creado los bloques se han revisado para comprobar que el proceso de exportado de Simulink es correcto.

Para empezar, las variables las importa como tipo Int, por lo que ha sido necesario cambiarlas cada una a su respectivo tipo. También se les ha dado un valor inicial predefinido.

El siguiente cambio, es que Simulink, en el proceso de exportado, ha añadido una variable de habilitación inicial que no estaba en el diseño, *ssMethodType*. Esta variable es de entrada, por lo que está controlada por otras funciones y cuando está desactivada, el resto de las variables, tanto internas, como de salida se ponen a 0. Cuando se activa, se inicia la función.

Esta variable se ha editado, tanto en la función asignación de Variables como en Asignación del SetPoint, cambiándole el nombre a *Reset* y se ha invertido su funcionamiento, para que reinicie las variables cuando está activada y que habilite la máquina de estados cuando está desactivada.

Tras esta variable, se encuentra la variable *c_is_active_c1_Asign*, que no tiene gran utilidad, ya que, si la variable está desactivada, se activa ella misma. Esta variable lo que hace

es habilitar o deshabilitar la máquina de estados. Esta variable debido a su falta de utilidad se ha eliminado de ambas funciones.

El último cambio que se ha realizado ha consistido en reordenar la máquina de estados para dejarla lo más parecida posible a la máquina de estados original de Simulink. Este cambio ha sido más estético que funcional, para ayudar a entender más fácilmente el funcionamiento de la maquina en caso de sea necesaria su revisión.

```

1 CASE #ssMethodType OF
2   0:
3
4     #temporalCounter_i1(timerAction := 1,
5       maxTime := 0);
6     #temporalCounter_i2(timerAction := 1,
7       maxTime := 0);
8     #temporalCounter_i1(timerAction := 1,
9       maxTime := 0);
10    #temporalCounter_i2(timerAction := 1,
11      maxTime := 0);
12    #temporalCounter_i1(timerAction := 1,
13      maxTime := 0);
14    #temporalCounter_i2(timerAction := 1,
15      maxTime := 0);
16    #c_is_active_c4_Asign := 0;
17    #is_c4_Asignaci := 0;
18    #Pobjetivo := 0.0;
19    #CState := 0.0;
20    #CambiarEstado := 0.0;
21    #Estado := 0.0;
22    #Ejecutar_Reinicio := 0.0;
23    #Habilitar_Motor := 0.0;
24    #Modo_Posicionamiento := 0.0;
25    #Posicionamiento := 0.0;
26    #SetPoint_1 := 0.0;
27    #Movimiento_1 := 0.0;
28    #SetPoint_2 := 0.0;
29    #Movimiento_2 := 0.0;
30    #Bloquear_Llamadas := 0.0;
31    #Apagar_LED_Activo := 0.0;
32    #bit_Habilitacion_MEF := 0.0;
33  1:
34
35    IF #c_is_active_c4_Asign = 0 THEN
36      #c_is_active_c4_Asign := 1;
37
1 IF #Reset THEN
2
3
4     #temporalCounter_i1(timerAction := 1,
5       maxTime := 0);
6     #temporalCounter_i2(timerAction := 1,
7       maxTime := 0);
8     #Habilitacion_MEF := FALSE;
9     #Estado_MEF := 1;
10    #Pobjetivo := 0;
11    #CState := FALSE;
12    #CambiarEstado := FALSE;
13    #Posicionamiento := FALSE;
14    #Habilitar_Motor := FALSE;
15    #SetPoint_1 := 0.0;
16    #SetPoint_2 := 0.0;
17    #Estado := 1;
18    #Movimiento_1 := FALSE;
19    #Movimiento_2 := FALSE;
20    #"Bloqueo Planta 1" := TRUE;
21    #Ejecutar_Reinicio := FALSE;
22    #"bit habilitacion MEF AV" := FALSE;
23    #"Apagar LED Activo" := FALSE;
24    #"Bloquear Llamadas" := TRUE;
25  ELSE
26
27    IF (NOT #Habilitacion_MEF) THEN
28      #Habilitacion_MEF := TRUE;
29    ELSE
30
31      CASE #Estado_MEF OF
32        1:
33          #Estado := 1;
34
35      IF #Start THEN
36        #temporalCounter_i1(timerAction := 2,

```

Figura 3.82. Cambios realizados en la programación tras ser importada a TIA Portal.

3.2.4.4. Composición de la función control de la maqueta.

Dentro de la función control de la maqueta se han insertado las dos funciones mencionadas anteriormente y se han interconectado con sus respectivas variables. Las variables que conectan ambas funciones, como *Planta*, *Estado* o *Apagar_LED* se han creado como variables estéticas, mientras que las variables que conectan las funciones con la función control del motor o con las variables globales se han creado como variables de entrada y salida.

Finalmente, la función control de la maqueta se ha insertado en el bloque OB Main y se han conectado sus variables de entrada y de salida con las variables correspondientes, Figura 3.83.

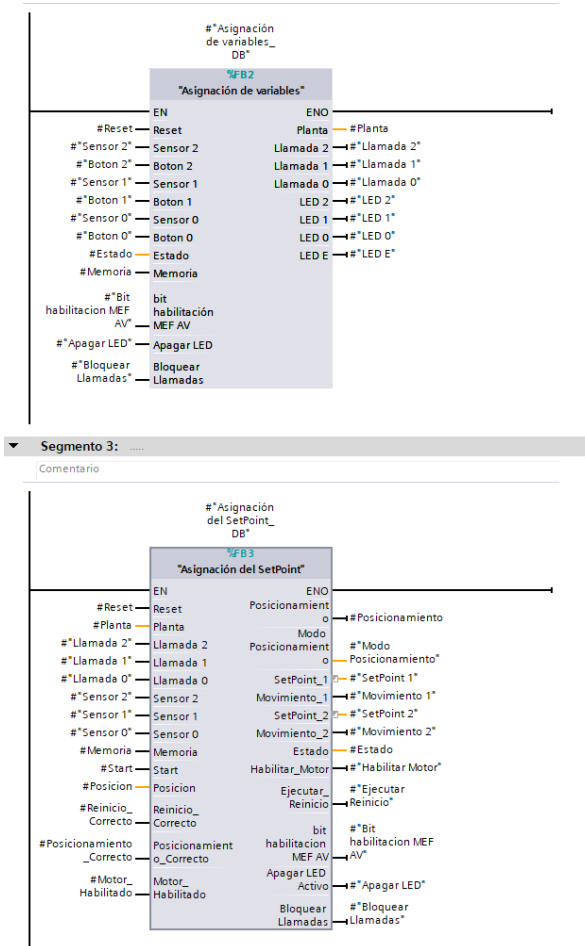


Figura 3.83. Conexión entre ambas funciones en la función control del ascensor en TIA Portal.

4. Configuración de la pantalla de control.

En este capítulo se va a presentar y caracterizar los detalles del diseño de las imágenes que constituyen las pantallas de operador. Este capítulo se corresponde con la etapa 5 del desarrollo del trabajo.

Una vez que se han realizado y verificado los programas de control de cada proyecto es necesario incluir el dispositivo HMI en los proyectos ya que, hasta este momento, la única forma de controlar las maquetas, salvo en el caso del proyecto 4 que puede realizarse mediante la botonera de la maqueta, es mediante un ordenador con el software TIA Portal instalado con el que poder cambiar manualmente el valor de las variables.

De forma similar a como se ha elaborado el programa de control de cada proyecto se ha seguido una secuencia para la elaboración de la pantalla HMI la cual, es la siguiente:

1. Elección del dispositivo HMI.
2. Configuración del dispositivo HMI.
3. Creación de las pantallas de control del programa.
4. Asignación de las variables del PLC a los elementos de la pantalla.

En este caso se va a explicar solamente la pantalla del proyecto 1, ya que todas han derivado de él.

4.1. Creación y conexión de la pantalla.

De forma similar a la selección del autómatas se ha elegido la pantalla HMI que se ha utilizado. Como se ha mencionado antes, en este trabajo se ha utilizado la pantalla HMI TP700 Confort.

Tras seleccionarla, aparece el asistente de configuración, Figura 4.1 donde, entre otras cosas, se realiza la conexión con la CPU del autómatas, la estructura de imágenes o se elige el encabezado. En este proyecto, se han añadido las siguientes imágenes:

- Imagen raíz.
- Control Eje.
- Diagnostico.
- Intro control.
- Modo manual.
- Posicionamiento absoluto.
- Posicionamiento relativo.

Una vez que se ha añadido la pantalla, aparece la imagen raíz, que será la imagen inicial cuando se encienda el dispositivo HMI.

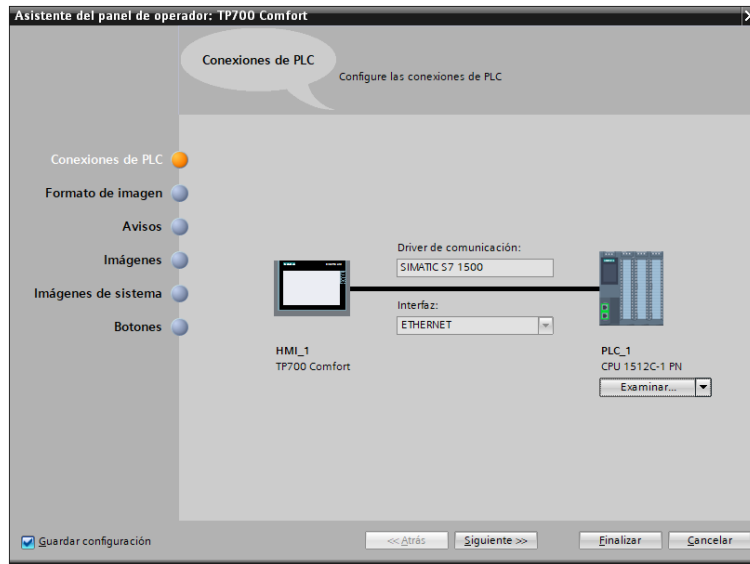


Figura 4.1. Asistente de configuración de la pantalla HMI.

4.2. Imágenes de las que consta la pantalla de operador HMI.

En esta sección se van a presentar las imágenes de las que consta la pantalla de operador y que han sido mencionadas anteriormente.

4.2.1. Imagen Raiz.

Esta pantalla es la primera que aparece cuando se enciende el dispositivo HMI, de ahí el nombre de raíz, y su finalidad es servir de introducción al usuario sobre el control que va a realizar. Dentro de esta imagen existen dos botones para ir a la pantalla de diagnóstico o a la de introducción al control.

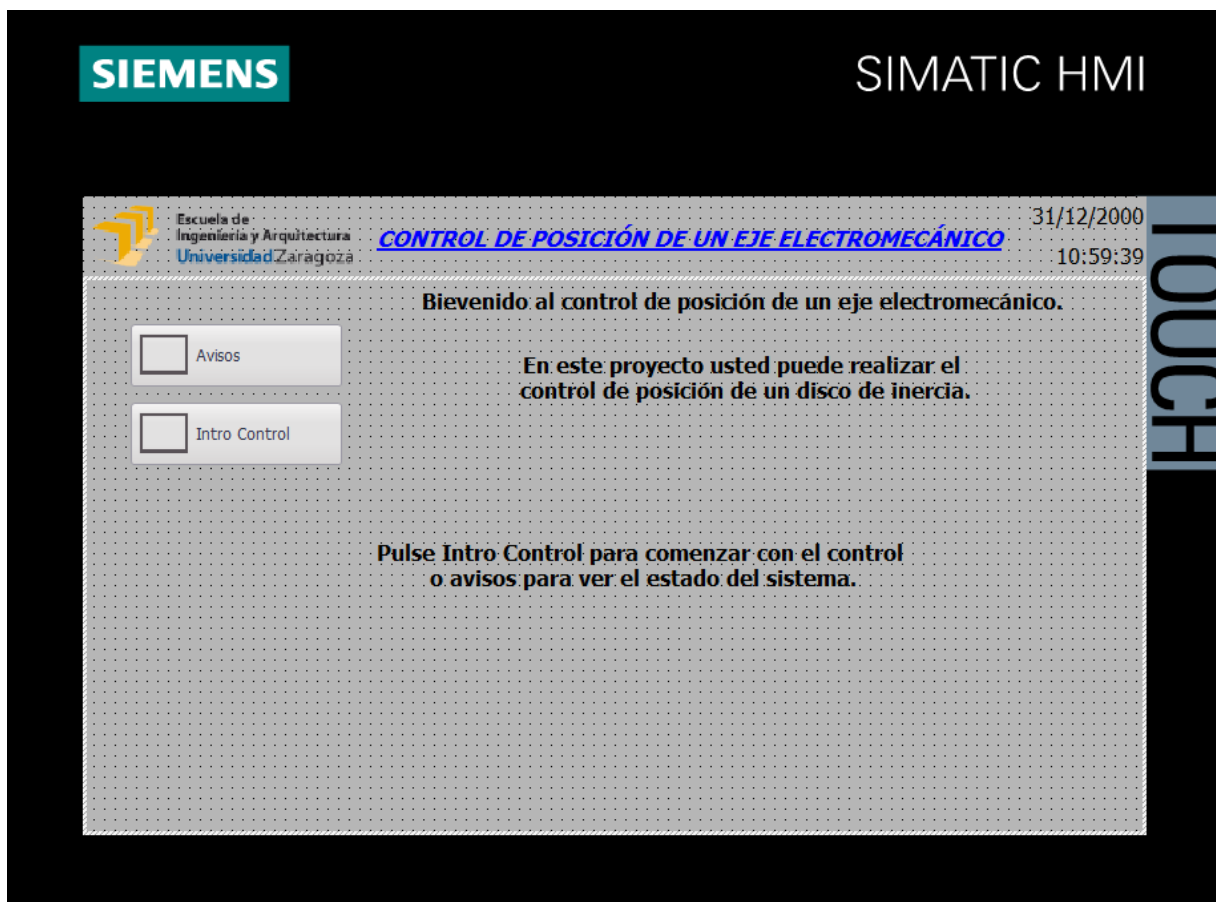


Figura 4.2. Pantalla de introducción al control.

4.2.2. Imagen Intro control.

La finalidad de esta imagen es permitir al usuario seleccionar el tipo de movimiento que quiera realizar:

- Modo manual.
- Movimiento absoluto.
- Movimiento Relativo.

Además, se incluye una breve descripción de cada tipo de movimiento.

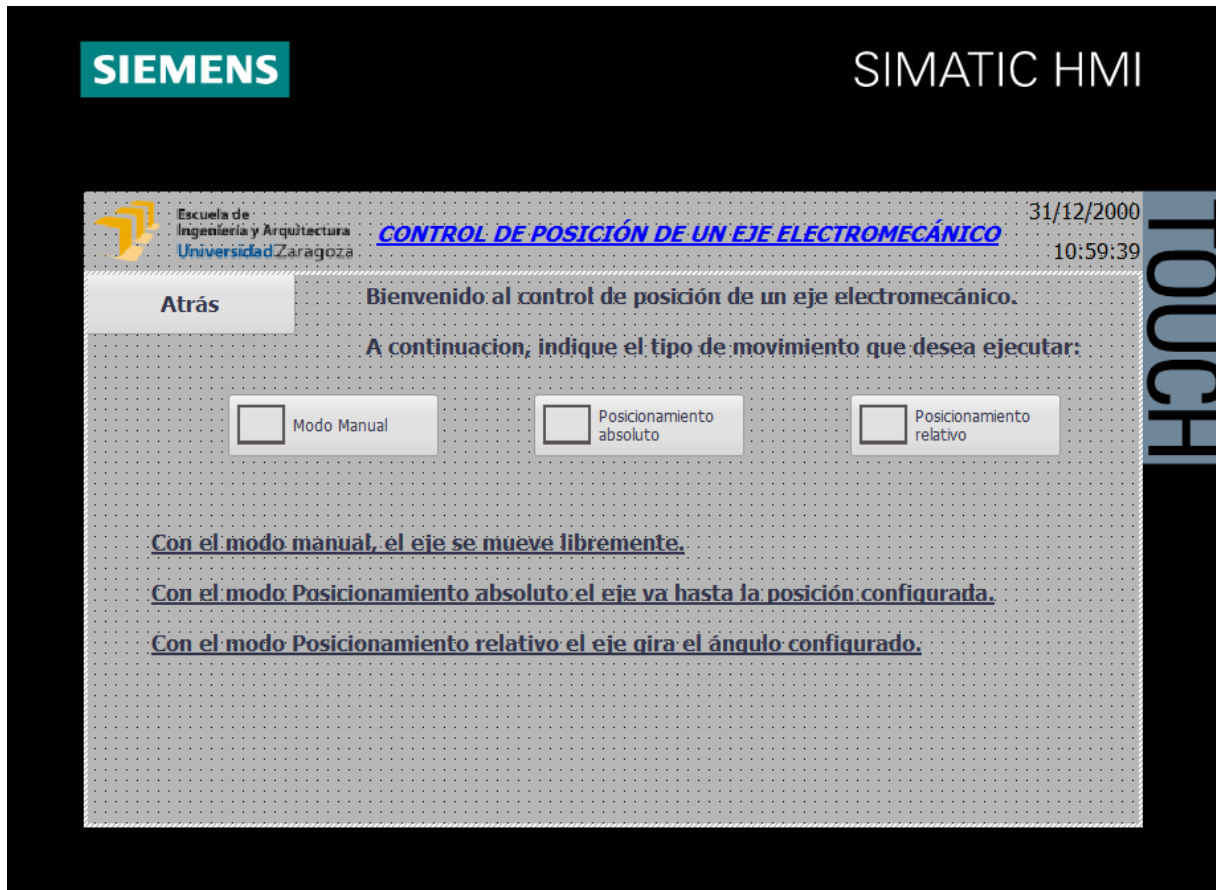


Figura 4.3. Pantalla de elección de movimiento.

4.2.3. Imagen Diagnóstico.

La finalidad de esta imagen es servir como diagnóstico del estado de la pantalla HMI y del autómeta.

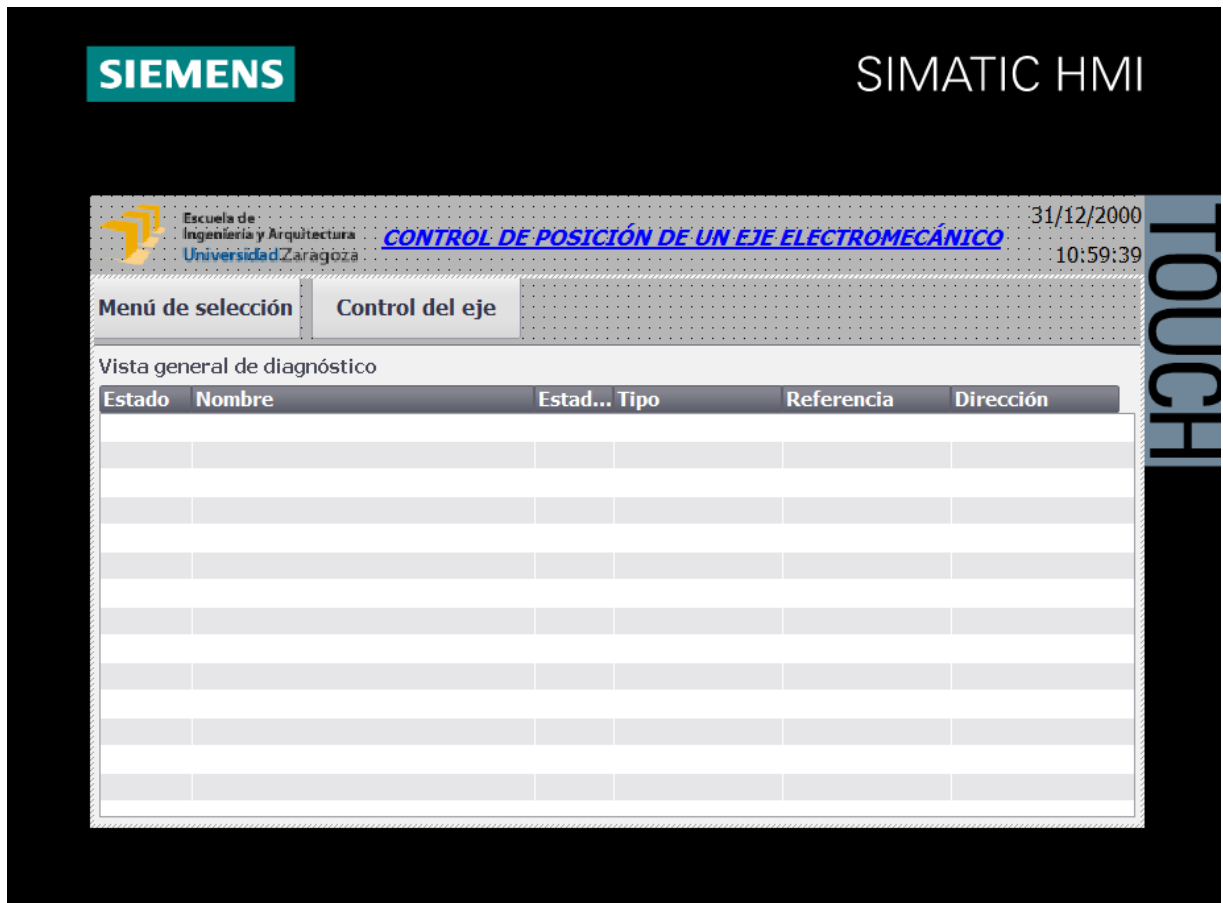


Figura 4.4. Pantalla de diagnóstico del sistema.

4.2.4. Imagen Control del eje.

La finalidad de esta imagen es que el usuario pueda controlar el eje, es decir, habilitarlo, reiniciarlo o posicionarlo. Esta imagen consta de los siguientes elementos:

Un botón que permite volver al menú de selección del tipo de movimiento y otro que permite ir a la imagen de diagnóstico.

1. Un conmutador ON/OFF que permite habilitar o deshabilitar el eje.
2. Un conmutador RST/ACK que permite seleccionar entre reiniciar el eje o confirmar los avisos.
3. Un botón que permite posicionar al eje en la referencia de posición.
4. Un botón que permite realizar el reinicio del eje o confirmar los avisos, según se haya elegido en el conmutador.
5. Un LED indicador del estado del eje, si está habilitado o no.
6. Un LED para indicar que el eje esta referenciado.
7. Un LED que indica si hay algún error en el eje.
8. Un cuadro de texto que indica el código de error para facilitar su resolución.
9. Un LED que indica que el conmutador está en posición de reinicio del eje y otro que indica que el conmutador está en posición de confirmar avisos.
10. Un LED que indica cuando el eje ha sido reiniciado o se han confirmado los avisos.
11. Un LED que indica cuando se ha alcanzado la velocidad de referencia en el modo manual y otro que indica cuando se ha detenido el eje en el modo manual.
12. Una gráfica que indica el porcentaje del duty aplicado a cada generador PWM, así como la acción en voltios aplicada por el autómata.

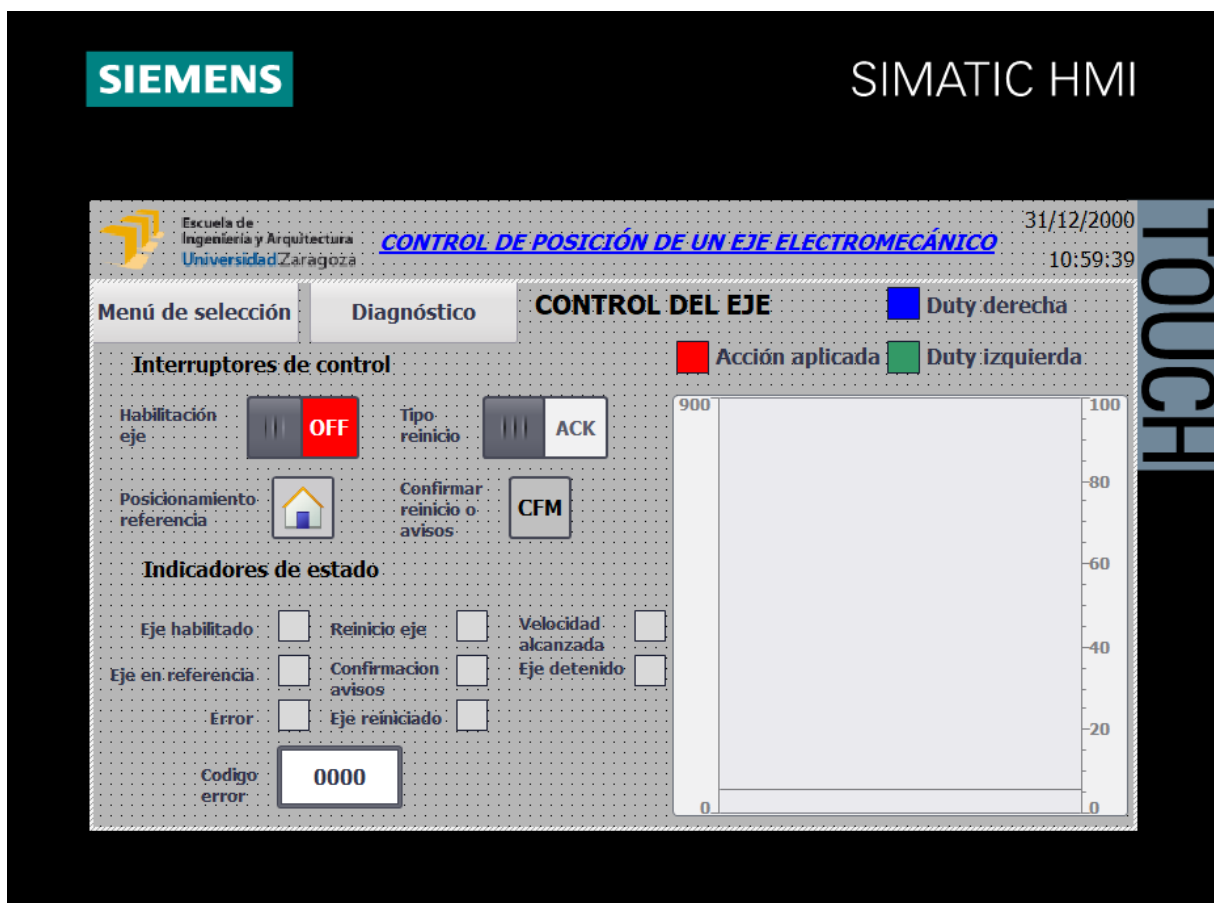


Figura 4.5. Pantalla de control del eje.

4.2.5. Imagen Posicionamiento absoluto.

La finalidad de esta imagen es permitir al usuario realizar un movimiento absoluto hasta la posición indicada. Esta pantalla es muy similar a la del modo manual y consta de los siguientes elementos:

1. Un botón que permite volver al menú de selección del tipo de movimiento y otro que permite ir a la imagen de control de eje.
2. Un conmutador ON/OFF que permite activar el movimiento absoluto a la vez que se desactivan el movimiento relativo y el modo manual.
3. Un cuadro de texto que permite al usuario introducir el valor deseado de velocidad a la cual gira el disco de inercia de la maqueta.
4. Un cuadro de texto que permite al usuario introducir el valor objetivo de la posición hasta la cual se moverá el disco de inercia de la maqueta.
5. Un botón para comenzar el movimiento.
6. Un botón de parada de emergencia, que frena el eje, desactiva el modo de control e inhabilita y bloquea el eje de forma instantánea.
7. Tres cuadros de texto que indican la referencia de velocidad, la consigna generada por el autómatas y la velocidad actual.
8. Dos cuadros de texto que indican la posición objetivo y la actual.
9. Una gráfica que permite visualizar tanto la velocidad actual como la posición actual, además de las referencias de velocidad y de posición.

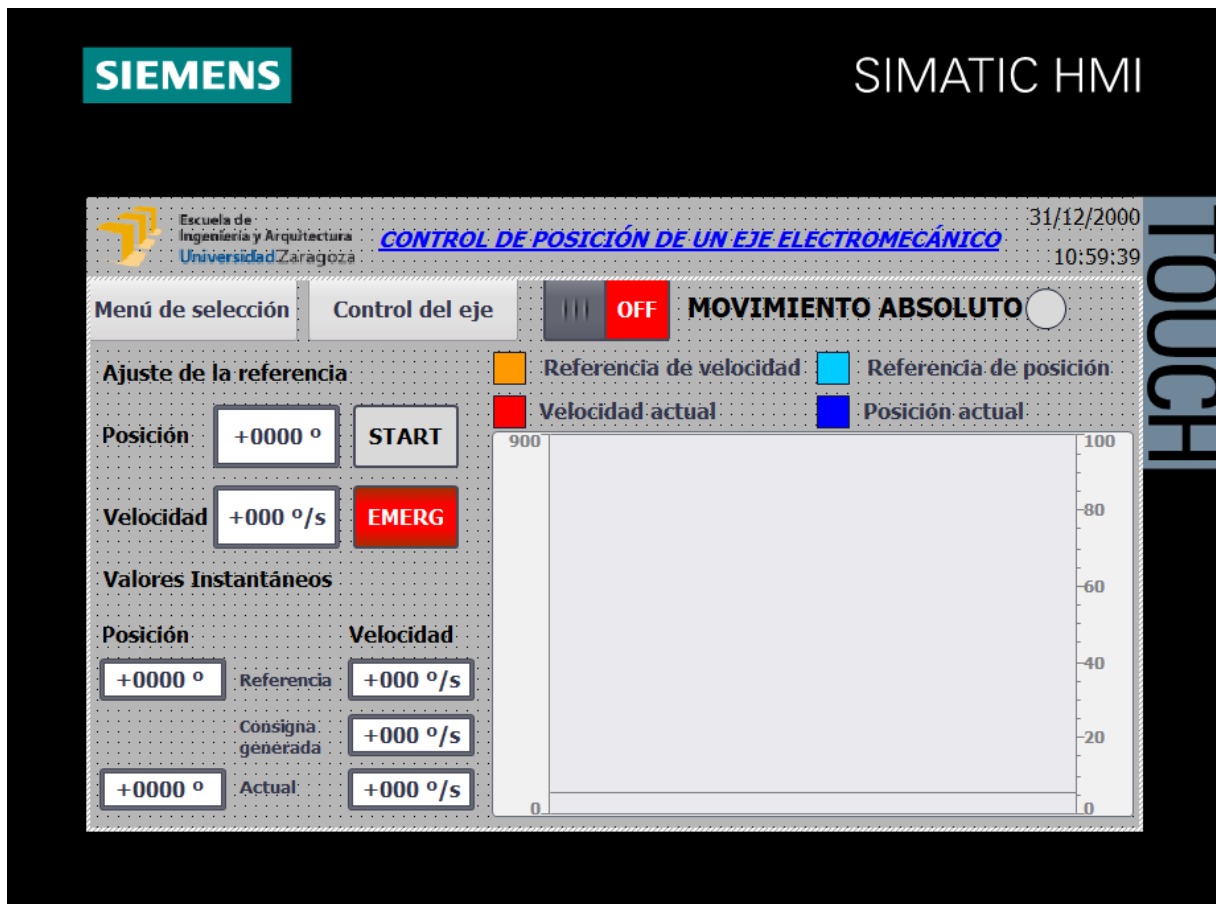


Figura 4.6. Pantalla de control del movimiento absoluto.

4.2.6. Imagen Posicionamiento relativo.

La finalidad de esta imagen es permitir al usuario realizar un movimiento relativo con la distancia indicada. Esta pantalla es idéntica a la del movimiento absoluto con las siguientes particularidades:

- El conmutador ON/OFF activa el movimiento relativo y desactiva el movimiento absoluto y el control manual.
- El botón Start realiza el movimiento relativo.

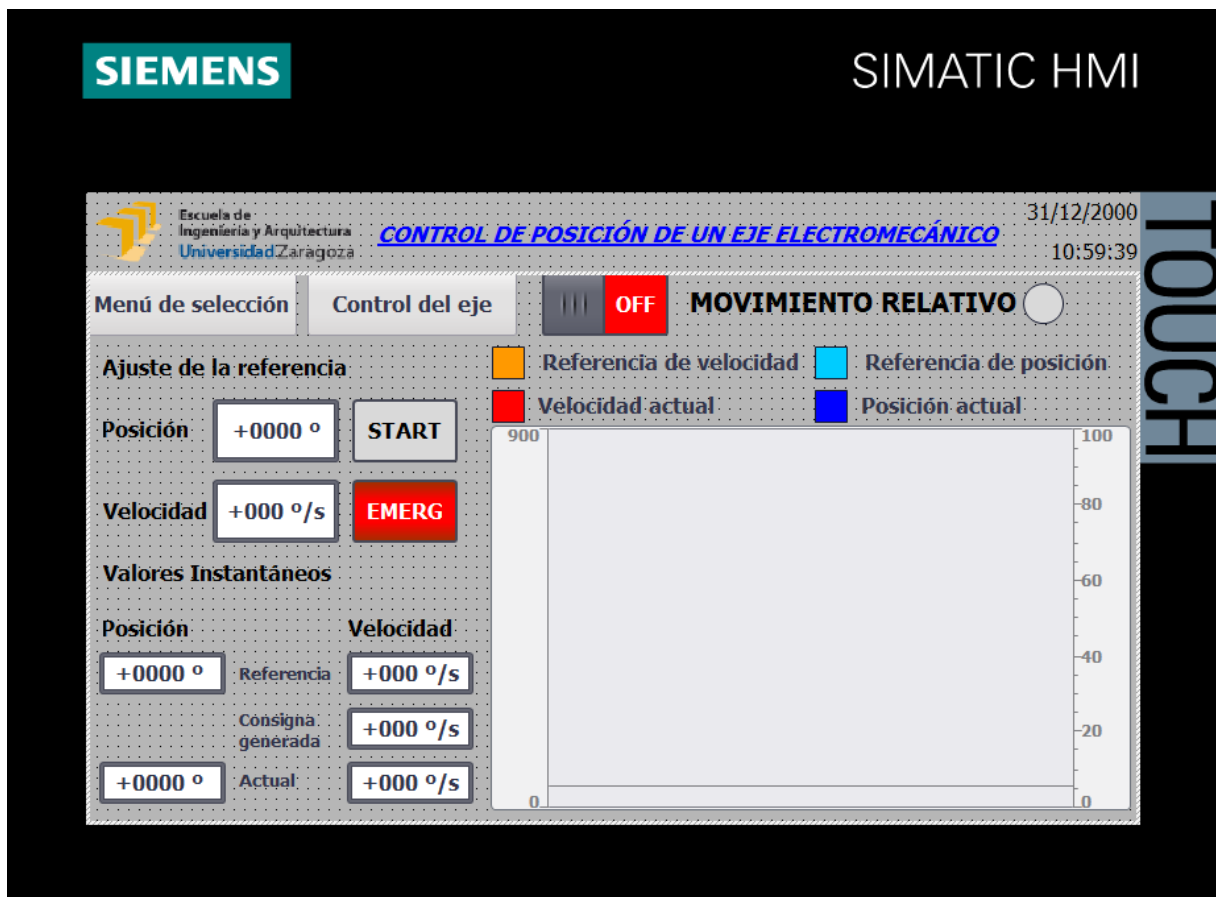


Figura 4.7. Pantalla de control del movimiento relativo.

4.3. Pantallas del resto de proyectos.

Las pantallas de los proyectos 2 y 3 incluyen simplemente los elementos básicos para el control de las maquetas mencionados anteriormente como son el conmutador ON/OFF que habilita el eje el cuadro de texto para elegir la referencia de posición de los movimientos. En el caso del proyecto 3 se incluyen también los botones para llamar al ascensor desde los diferentes pisos.

En el caso del proyecto 4, la pantalla es similar a la del 1, pero incorporando solo la pantalla del movimiento absoluto de entre las 3 imágenes de movimiento del proyecto 1, donde la selección de la referencia ha sido sustituida por los botones de los pisos correspondientes y la indicación de la posición se muestra en tiempo real sobre una figura con la forma del ascensor.

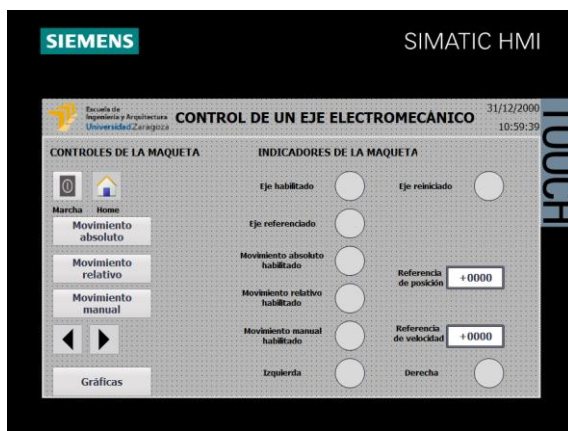


Figura 4.9. Pantalla de control del proyecto 2.

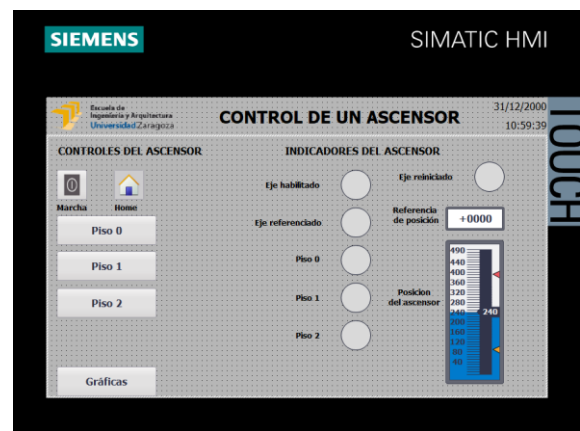


Figura 4.8. Pantalla de control del proyecto 3.

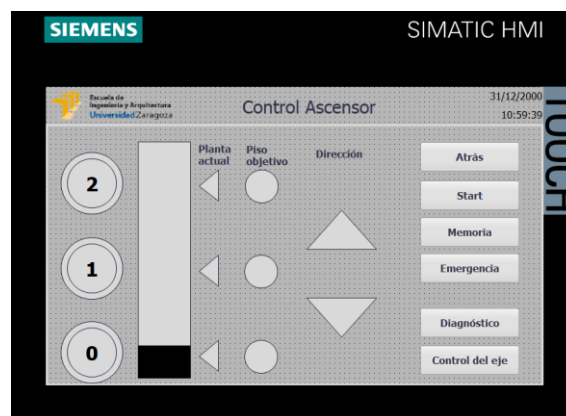


Figura 4.10. Pantalla de control del proyecto 4.

5. Desarrollo de la placa electrónica adaptadora de niveles.

En este capítulo se va a presentar el proceso de desarrollo de la placa adaptadora de niveles para poder conectar las maquetas con el autómatas. Este capítulo se corresponde con la etapa 6 del proyecto.

Como se ha visto en el capítulo 2, las maquetas funcionan con una variedad de tensiones que van desde los 5 V hasta los 10 V, mientras que el autómatas trabaja con 24 V nominales. Para resolver estos problemas, se ha diseñado y construido una placa adaptadora de niveles de tensión para convertir los 5 V, por ejemplo, del encóder de la maqueta del disco de inercia, a los 24 V de la entrada del autómatas, o los 24 V de las salidas del autómatas a los 5 V de los LEDs de la maqueta del ascensor.

5.1. Consideraciones previas.

Antes de diseñar la placa, es necesario conocer las señales utilizadas en el trabajo y sus rangos de tensión, las cuales, son los siguientes:

- 1) Autómatas:
 - a) Salidas PWM: rango de tensión de 0 a 24 V.
 - b) Salidas LEDs del ascensor: rango de tensión de 0 a 24 V.
 - c) Entrada de señal del inductivo: rango de tensión de 0 a 24 V.
 - d) Entradas del encóder: rango de tensión de 0 a 24 V.
 - e) Entrada del encóder externo: rango de tensión de 0 a 10 V.
 - f) Entradas pulsadores del ascensor: rango de tensión de 0 a 24 V.
 - g) Entradas finales de carrera del ascensor: rango de tensión de 0 a 24 V.

- 2) Maqueta del disco de inercia:
 - a) Entradas PWM: rango de tensión de 0 a 9 V.
 - b) Salidas del encóder: rango de tensión de 0 a 5 V.
 - c) Salida de señal del inductivo: rango de tensión de 0 a 24 V.
 - d) Salida del encóder externo: rango de tensión de 0 a 10 V.

- 3) Ascensor:
 - a) Entradas PWM: rango de tensión de 0 a 9 V.
 - b) Salidas del encóder: rango de tensión de 0 a 5 V.
 - c) Entrada LEDs: rango de tensión de 0 a 3,3 V.
 - d) Salida pulsadores: sin rango definido.
 - e) Salida finales de carrera: sin rango definido.

Las señales PWM y del encóder, son las mismas para ambas maquetas ya que los motores NXT de las maquetas son idénticos. Por otro lado, el rango de los pulsadores y los finales de carrera no está definido puesto que al ser interruptores y finales de carrera no tienen unos requisitos de tensión.

5.2. Elección de componentes.

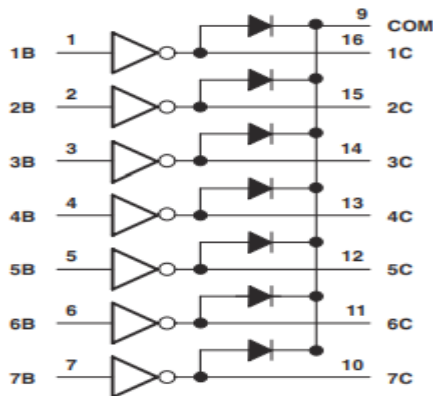


Figura 5.1. Pinout del chip ULN2003AD. [8]

En la adaptación de los rangos de tensión de las señales y teniendo en cuenta que, salvo la señal del encóder externo que puede conectarse directamente al autómeta, las señales son digitales, se ha utilizado una estructura compuesta por un transistor funcionando en modo saturación. Como las señales son de baja potencia, de unos pocos mW , y de baja frecuencia, como mucho 10 kHz , se han utilizado arrays de transistores para realizar la conversión de voltaje, tanto de 24 V a 5 V , como de 5 V a 24 V . El circuito integrado que se ha elegido para realizar esta tarea es el ULN2003AD, que contiene un array de 7 transistores con emisor común y con la base preparada para trabajar directamente a 5 V .

También, en el caso de las señales de 5 V que requieren de una inversión de polaridad, para mantener la polaridad original ya que los transistores la invierten, y de un filtrado para eliminar los ruidos, como las señales del encóder, se ha utilizado un array de inversores Trigger – Schmitt, el circuito integrado 74AC14D, que contiene 6 inversores y trabaja, como mucho, a 7 V .

Para las señales PWM, estas no pueden conectarse directamente entre los inversores y el motor, ya que este tiene unos requisitos de corriente de $0,181\text{ A}$ nominales y de $0,653\text{ A}$ en el pico de arranque, tal y como se muestra en las figuras, Figura 5.4 y Figura 5.5 los inversores solamente pueden dar 50 mA como máximo. A parte, el motor requiere 9 V y el valor máximo de tensión permitido en los inversores es de 7 V . Por tanto, ha sido necesario incluir un controlador para motores DC que sea capaz de trabajar con 9 V y dar hasta 1 A de corriente. El controlador elegido ha sido el LB1836M, con capacidad para controlar tensiones de hasta $10,5\text{ V}$ y corrientes de hasta 1 A .

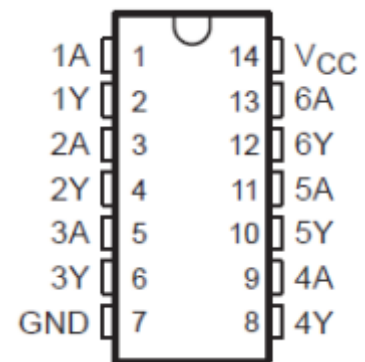
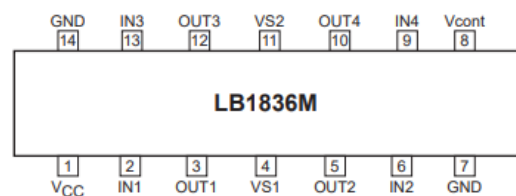


Figura 5.3. Pinout del chip SN74AC14D. [18].



Note) Ground both GND pins.

Figura 5.2. Pinout del chip LB1836M. [17]

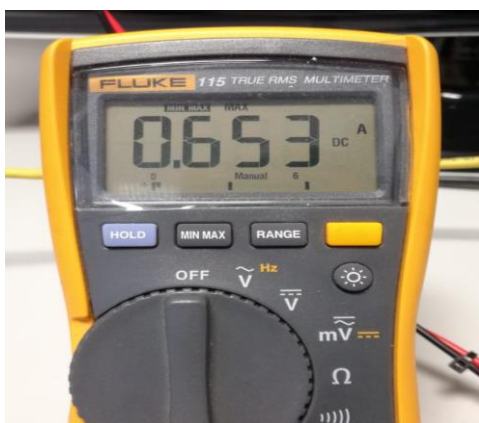


Figura 5.4. Pico de intensidad de arranque del motor NXT.

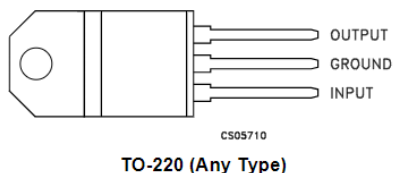


Figura 5.5. Intensidad nominal del motor NXT.

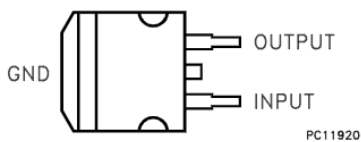


Figura 5.6. Aplicación típica de la fuente conmutada LM78_09-2.0R. [19]

Respecto a las obtención de las tensiones de trabajo, se ha decidido que, por eficiencia, la placa se alimente a 24 V DC de una fuente externa y de esta se obtengan el resto de las tensiones de 10, 9 y 5 V. Para obtener el bus de 9 V, que es el que alimenta al motor NXT y, por tanto, el bus que más energía requiere, se ha decidido utilizar el integrado LM78_09-2.0R, que es una fuente conmutada y encapsulada de tipo Buck, para mantener la máxima eficiencia posible, que se encarga de reducir la tensión de 24 V a 9 V, siendo capaz de dar una corriente de 2 A.



TO-220 (Any Type)



D²PAK (Any Type)

Figura 5.7. Pinouts de los chips LM7805CV y L7810ACD2T. [20]

En el caso de los buses de 10 y 5 V, debido a su escaso requerimiento de energía, se han utilizado reguladores lineales, el LM7805CV para obtener el bus de 5 V y el L7810ACD2T para obtener el de 10V. El bus de 10 V se obtiene a partir del bus de 24 V y el de 5 V a partir del bus de 9 V.

Respecto a los conectores, se ha utilizado un conector DB25 Macho para agrupar todas las señales que conectan la placa con el autómata mientras que para las señales entre la placa y la maqueta del ascensor se ha utilizado un conector DB25 Hembra. Como ambos conectores son DB25, se ha decidido que uno fuese macho y otro hembra para evitar errores en la conexión de ambos.

Para las señales de potencia del motor y las señales del encóder, se ha utilizado el mismo conector LEGO derivado del RJ12 que se ha visto en el capítulo 2. En el caso de las señales del inductivo y del encóder externo se ha utilizado un conector DB9. Por último, para la alimentación de la placa con 24 V se ha utilizado un borne de dos vías.

Finalmente se han incluido también filtros en las señales de los botones y finales de carrera de la maqueta del ascensor para evitar ruidos que pudieran afectar al correcto funcionamiento del programa.

Los valores calculados de los componentes pueden consultarse en el Apéndice C.



Figura 5.10. Conector lego NXT RJ12. [7]



Figura 5.8. Conector DB9. [22]



Figura 5.9. Conector DB25. [21]

5.3. Diseño del circuito.

Como las señales del motor NXT son comunes a ambas maquetas, son las primeras señales cuya adaptación se va a explicar, empezando por las señales PWM.

Inicialmente, las señales PWM que salen del autómatas, entre 0 y 24 V, llegan a la placa a través de los pines 12 y 13 del conector DB25 macho, donde se reducen, mediante un par de transistores del integrado ULN2003AD, a una tensión de entre 0 y 5 V. Estos transistores también invierten la señal por lo que las señales pasan por un par de inversores del chip 74AC14D. Con la tensión y la fase adecuadas, pasan al LB1836M de donde salen hacia el motor, a través de los pines 1 y 4 del conector RJ12, con el rango de tensión de 0 a 9 V y con la corriente necesaria. La Figura 5.11 muestra la estructura que adapta dichas señales

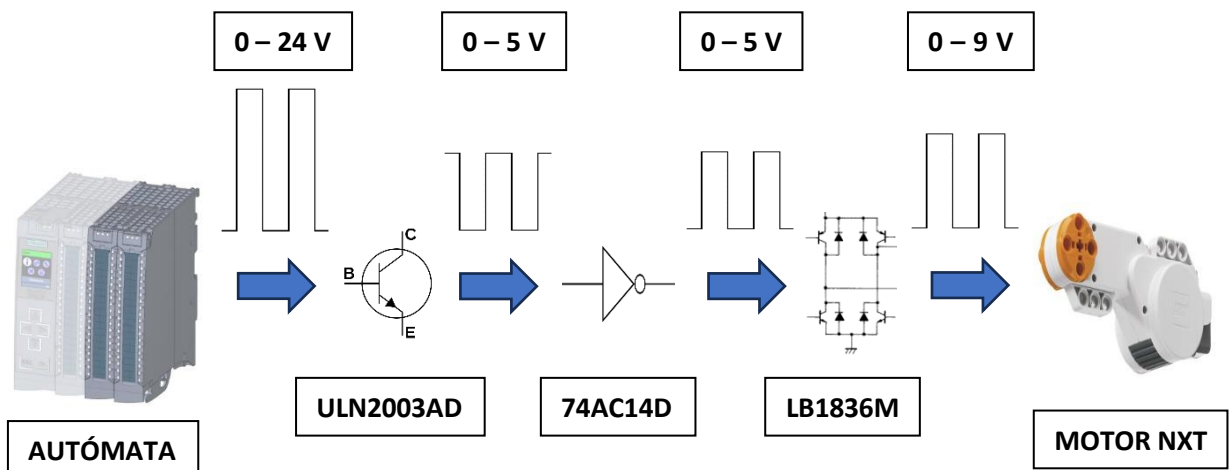


Figura 5.11. Esquema de adaptación de las señales entre el autómatas y el motor NXT.

Después de las señales PWM van las señales A y B del encóder, que van desde el motor NXT hasta el autómatas. En este caso, las señales entran en la placa a través de los pines 3 y 6 del conector RJ12 y llegan hasta un par de inversores del integrado 74AC14D, donde son invertidas para que al convertirlas al rango 0-24V mediante un par de transistores del integrado ULN2003AD, salgan con la polaridad correcta. Tras haber sido adaptadas al rango de trabajo del autómatas, las señales salen de la placa, hacia el autómatas, por el conector DB25 macho. La Figura 5.12 muestra la estructura que adapta dichas señales.

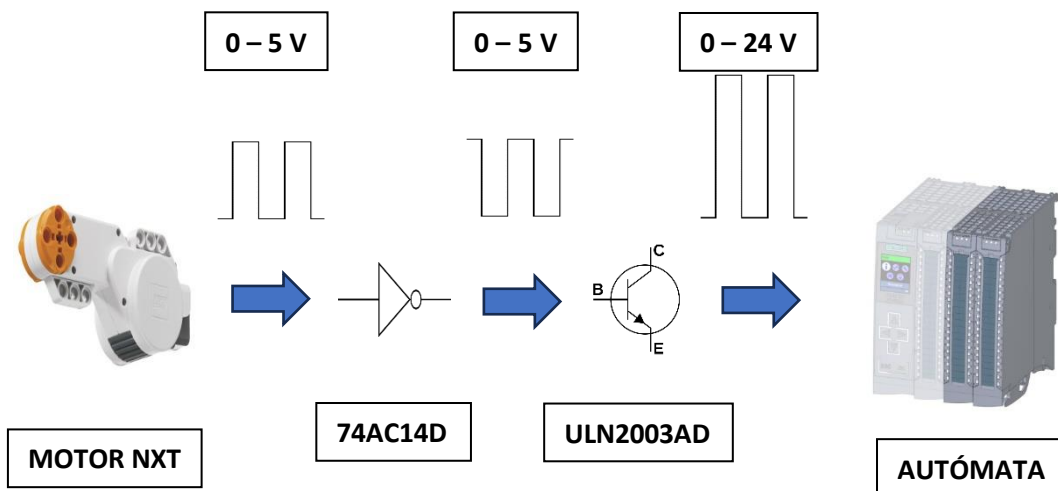


Figura 5.12. Esquema de adaptación entre las señales del encóder y el autómatas.

En el caso de las señales de la maqueta del disco de inercia, la señal del inductivo y del encóder externo, la adaptación se produce en la propia maqueta por lo que llegan a la placa adaptadora a través de los pines 1 y 8 del conector DB9 y salen directamente hacia el autómatas por los pines 20 y 21 del conector DB25 macho.

En el caso de las señales de los pulsadores de la maqueta del ascensor, que son NC, estas llegan a la placa a través de los pines 2, 3, 4, 5 y 6 del conector DB25 hembra, donde se filtran mediante un filtro RC para evitar comportamientos raros de la señal y posteriormente se invierten con transistores del integrado ULN2003AD para que una pulsación de estos botones implique un "1" lógico en el autómatas. Al igual que el resto de las señales que van al autómatas, estas salen por los pines 2, 3, 4, 5 y 6 del conector DB25 macho. A diferencia de las señales del encóder, aquí no se utilizan inversores ya que el rango de tensiones es demasiado alto para el chip 74AC14D. La Figura 5.13 muestra la estructura que adapta dichas señales.

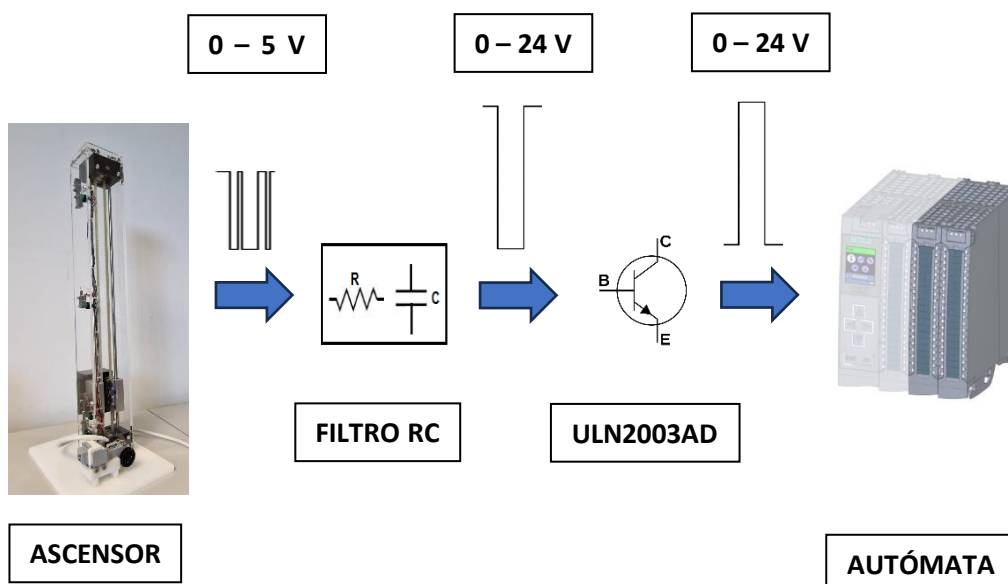


Figura 5.13. Esquema de adaptación entre las señales de los pulsadores y el autómatas.

La adaptación de las señales de los finales de carrera de la maqueta del ascensor es muy similar a la de los pulsadores. La única diferencia es que estas no se invierten porque los finales de carrera son NO, entonces su activación provoca un "1" lógico. Como con las señales de los pulsadores, estas llegan a la placa por los pines 7, 8, 9, 23 y 24 del conector DB25 Hembra y salen hacia el autómatas por los pines 7, 8, 9, 23 y 24 del conector DB25 macho. La Figura 5.14 muestra la estructura que adapta dichas señales.

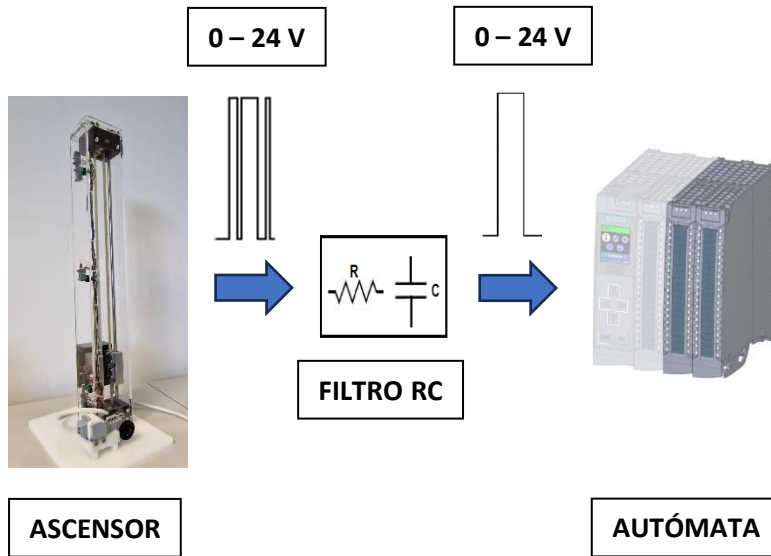


Figura 5.14. Estructura que adapta las señales entre los finales de carrera y el automático.

Finalmente, dado que los LEDs de la maqueta funcionan con lógica negativa, y el automático con lógica positiva, es necesario invertir la señal que llega del automático, a través de los pines 14, 15, 16, 17 del conector DB25 Macho, mediante transistores. Como los LEDs funcionan a 5 V se ha incluido una resistencia limitadora. Tras haber adaptado la señal, sale hacia la maqueta a través de los pines 16, 17, 18 y 19 del conector DB25 hembra. La Figura 5.15 muestra la estructura que adapta dichas señales.

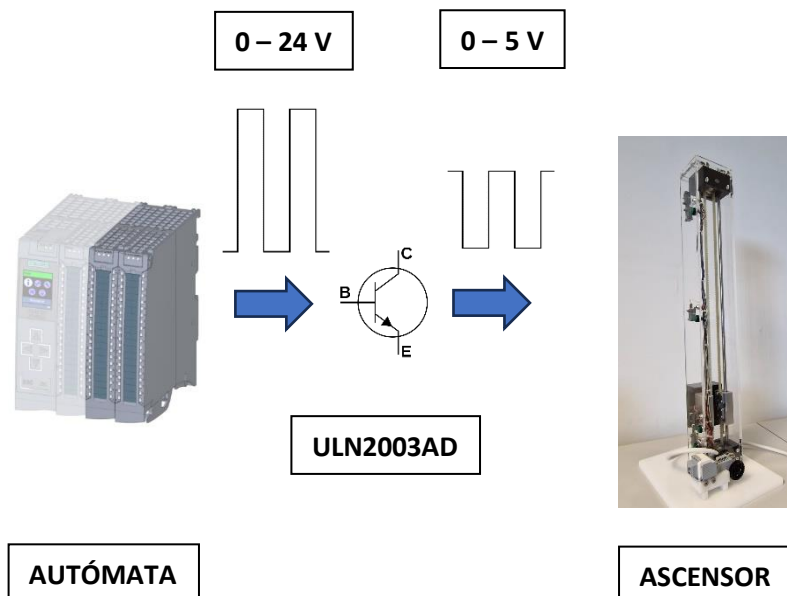


Figura 5.15. Estructura que adapta las señales entre el automático y los LEDs.

5.4. Diseño de la placa y vista 3D.

Cuando se ha verificado que el circuito diseñado en el apartado anterior funciona, se ha diseñado la placa PCB. Al igual que para diseñar el circuito, se ha utilizado el software DipTrace.

Debido al escaso número de componentes y, por tanto, de conexiones, se ha utilizado una placa de 2 capas, nombradas como inferior y superior. En la capa inferior se han colocado los componentes SMD mientras que en la superior se han colocado los componentes THD. Una vez colocados los componentes, se han enrutado las pistas a mano. A parte se han añadido planos de tierra, tanto en la cara superior como en la inferior para que sirva de apantallamiento contra el ruido electrónico.

Por último, se ha realizado el diseño 3D de la placa y se han creado los ficheros GERBER necesarios para su fabricación.

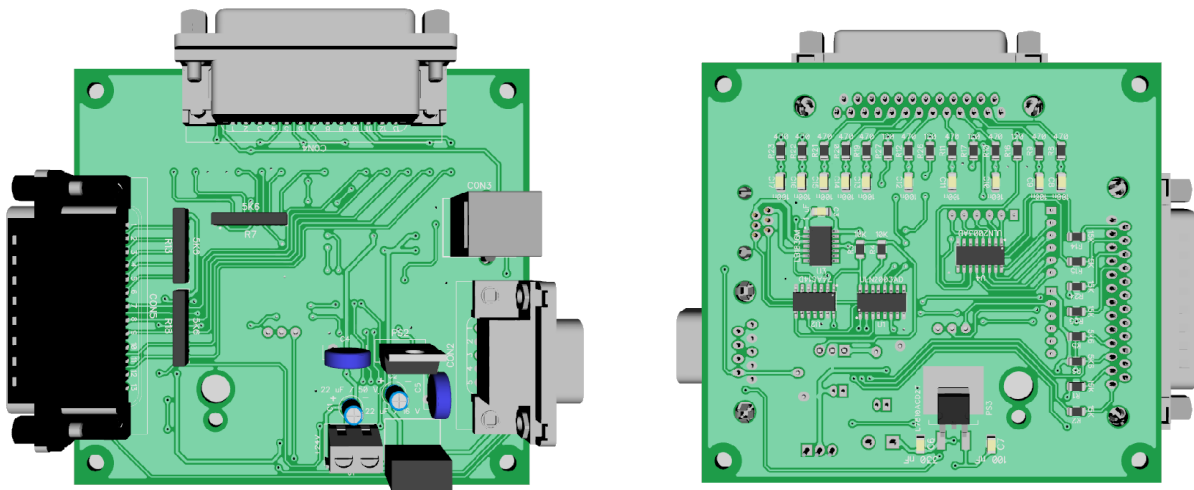


Figura 5.16. Visión 3D de la placa electrónica.

5.5. Conexión automática – Placa – Maquetas.

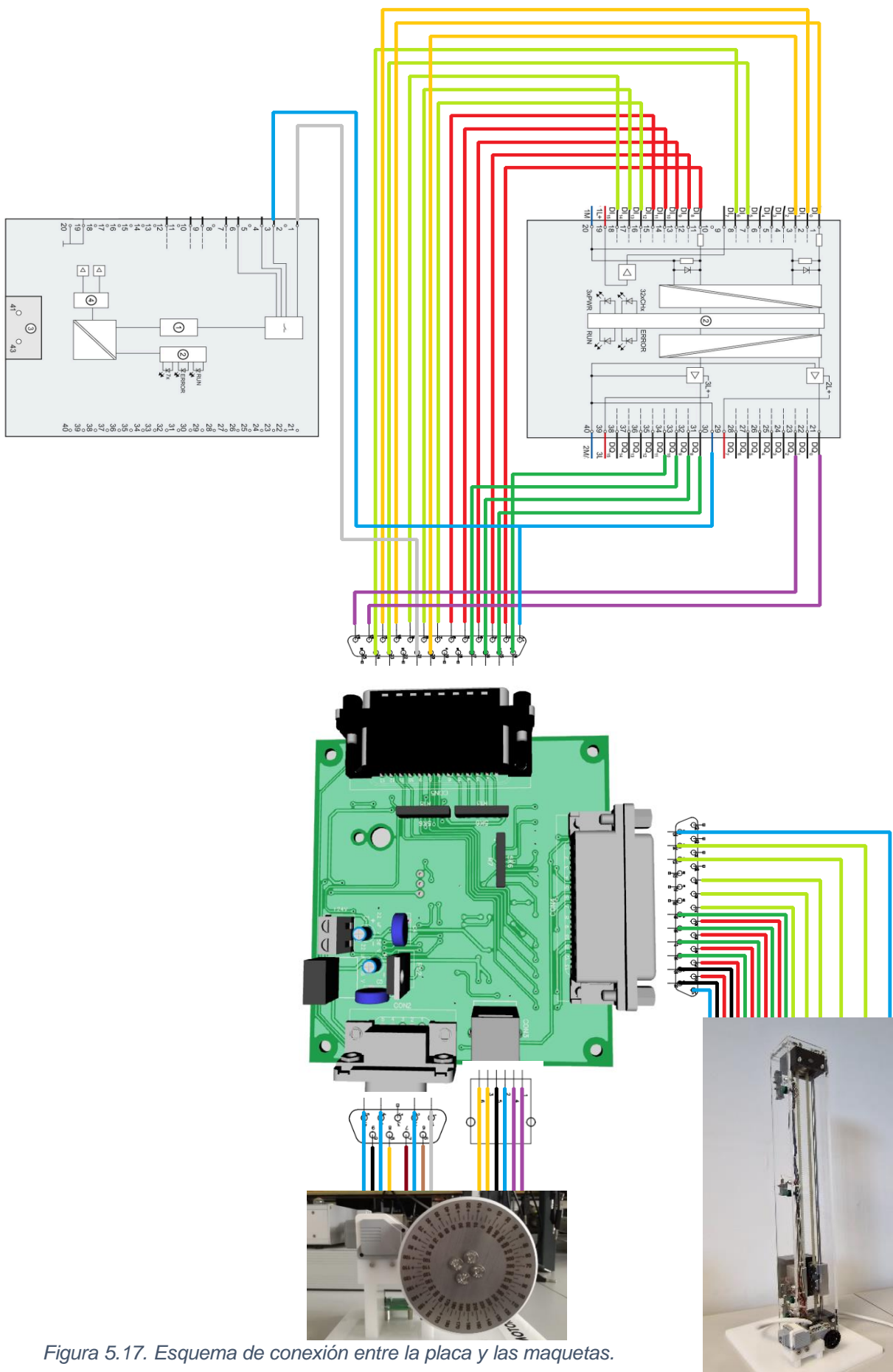


Figura 5.17. Esquema de conexión entre la placa y las maquetas.

6. Pruebas realizadas.

En este capítulo se va a presentar algunas de las pruebas realizadas tras haber finalizado la parte de diseño del trabajo. Este capítulo se corresponde con la etapa 7 del trabajo.

De las múltiples pruebas que se han realizado para verificar el funcionamiento de todas las partes del trabajo, tanto de componentes individuales, como, por ejemplo, simulaciones de los programas de control, como de todo el trabajo completo, se van a destacar las siguientes:

- Prueba de la electrónica del trabajo. – En esta prueba se ha comprobado que tanto la placa electrónica como las entradas y salidas del autómata funcionan correctamente.
- Prueba conjunta de todos los elementos del trabajo. – En esta prueba se ha comprobado que todos los elementos del trabajo funcionan correctamente.

6.1. Prueba de la electrónica.

En esta prueba se ha verificado el correcto funcionamiento de la placa electrónica, es decir, que no existan cortocircuitos y que la adaptación de las señales sea correcta.

Inicialmente, en esta prueba se ha detectado un cortocircuito en el bus de 5 V. El cortocircuito se ha encontrado en el integrado 74AC14D donde los pines 6 y 12 estaban unidos a las pistas de masa. Como este error es un defecto de fabricación de las placas la solución adoptada ha sido cortar los pines 6 y 12 del integrado, Figura 6.1.

Una vez eliminado el cortocircuito se ha comprobado como algunas señales no llegaban de las maquetas al autómata y viceversa. Tras revisar las placas con el osciloscopio se ha detectado que algunos componentes no estaban correctamente soldados.

Finalmente, tras revisar las soldaduras y rehacer algunas, la placa PCB funcionaba correctamente.



Figura 6.1. Detalle del chip 74AC14D sin los pines 6 y 12.

6.2. Prueba conjunta de todos los elementos.

Esta prueba se ha realizado en los 4 proyectos creados y ha servido para lo siguiente:

- Por un lado, ajustar algunos parámetros de la configuración del objeto tecnológico como la constante proporcional del regulador o los márgenes de tolerancia del movimiento. Estos ajustes se han realizado mediante prueba y error.
- Por otro lado, una vez que los parámetros están correctamente configurados, comprobar el correcto funcionamiento de cada proyecto. En esta prueba se ha comprobado, tanto el error de posición como el de velocidad. Además, se ha comprobado que los tiempos, tanto de aceleración como de deceleración sean correctos.

Es precisamente en esta última prueba, donde se ha comprobado que el autómatas es capaz de regular correctamente la posición, pero no la velocidad, figuras Figura 6.2 Figura 6.3. El error consiste en que el regulador no puede mantener la velocidad de forma lineal, sino que la velocidad va a saltos. Por ejemplo, en el caso de los proyectos del disco de inercia, la velocidad produce saltos de $125^\circ / s$ y nunca se mantiene en el valor establecido en la referencia.

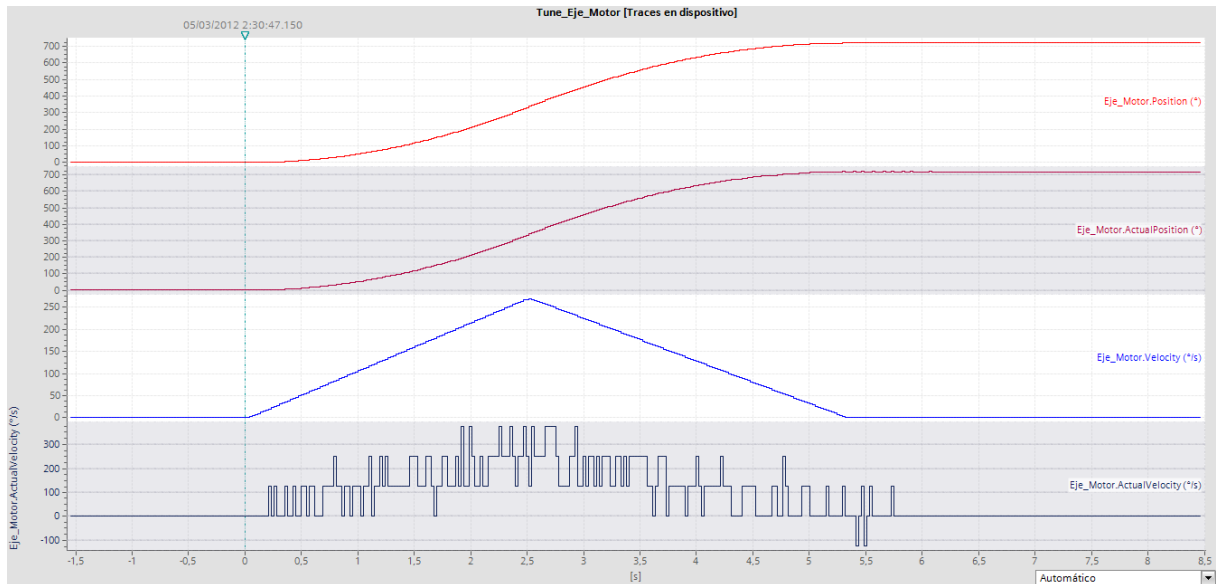


Figura 6.2. Gráfica que muestra la referencia de posición, la posición actual, la referencia de velocidad y la velocidad actual.

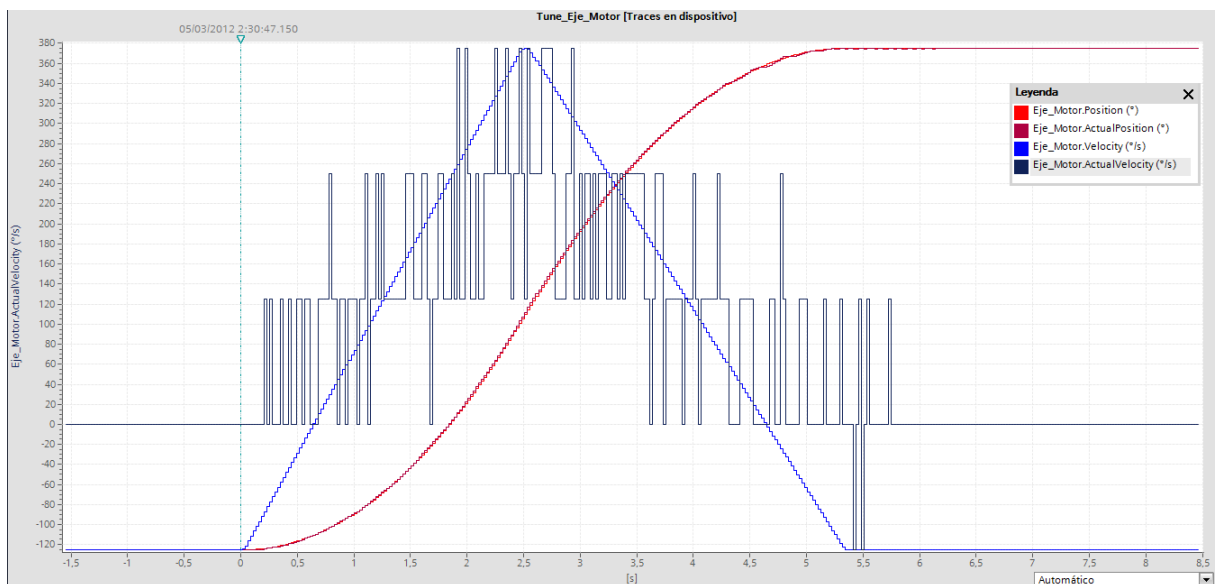


Figura 6.3. Gráfica donde se observa la velocidad actual y la referencia de velocidad juntas.

Teniendo en cuenta que la resolución del encóder es de $0,5^\circ / pulso$, el error en pulsos es de $250 pulsos / s$. La razón de este error es la siguiente:

Como el encóder es de $720 pulsos / vuelta$, la cantidad de veces que se lee el valor del número de pulsos del encóder se calcula mediante la Ecuación 6.1.

$$N_l = \frac{N_{encóder}}{e_p} = \frac{720}{250} = 2,88 \frac{\text{lecturas}}{\text{vuelta}}$$

Ecuación 6.1. Ecuación que determina el número de lecturas por vuelta del encóder.

Donde N_l es el número de lecturas por vuelta, $N_{encóder}$ es la resolución del encoder y e_p es el error de posición.

Como el autómata solo puede leer el valor del número de pulsos del encóder un número entero de veces por vuelta, unas veces lee 2 veces el valor mientras que otras veces lo lee 3 veces. Como el número de lecturas es aleatorio, es debido a esto la aparición de esos saltos de velocidad.

Para solucionar el error se ha probado a bajar la velocidad con la que el autómata calcula los valores de los reguladores, que por defecto es de 250 veces por segundo, para tener un número mayor de lecturas entre cada cálculo, consiguiendo mejores resultados respecto a la regulación de velocidad, pero empeorando mucho la regulación de posición, que es el principal objetivo de este trabajo, por lo que finalmente se ha dejado la velocidad por defecto.

Por tanto, salvo una mejora del encóder, resulta imposible solucionar este problema.

7. Conclusiones.

Con este trabajo se ha conseguido ampliar los conocimientos existentes sobre autómatas programables y sobre las capacidades que tiene Simulink para realizar proyectos de distinta índole como ha sido este caso.

Estos conocimientos se espera que repercutan facilitando el ingreso al mundo laboral debido a la importancia de la automatización industrial hoy en día y al uso tan extendido de los autómatas de Siemens en las industrias a nivel mundial.

7.1. Dificultades.

La única dificultad que destacar ha sido la falta de componentes electrónicos debido a la crisis de los semiconductores derivada de la pandemia de Covid-19 sufrida desde 2020, que ha provocado retrasos en la elaboración de la placa PCB.

7.2. Desarrollo del proyecto

Respecto a la adquisición de conocimientos sobre las maquetas o TIA Portal, esta no ha conllevado un gran esfuerzo debido a la gran cantidad de material existente en internet, como páginas especializadas en autómatas de Siemens, videotutoriales o incluso al propio sistema de ayuda de TIA Portal.

A la hora de desarrollar los proyectos en TIA Portal, se han encontrado algunos problemas, para los cuales se ha tenido que recurrir al foro oficial de Siemens, donde es de agradecer la rapidez y la profesionalidad con la que han sido resueltos los problemas. También se ha intentado realizar los proyectos de la forma eficientemente posible, consumiendo la menor cantidad posible de recursos del autómatata.

Respecto a la pantalla HMI, se ha intentado realizar de forma que cualquier persona pueda realizar el control, incluso sin tener conocimientos, facilitando esta tarea mediante indicadores, dándole al usuario datos como la velocidad o la posición actual o incluso los códigos de error que puedan generarse.

Finalmente, la placa electrónica se ha tratado de hacer lo más compacta y eficiente posible.

7.3. Ampliaciones y mejoras futuras.

Como posible mejora del proyecto, se podría sustituir tanto el motor lego NXT, como el encóder, por sus equivalentes industriales con lo que se conseguiría mayor precisión y estabilidad a la hora de realizar los movimientos. También podría añadirse, al igual que se ha hecho con los autómatas de Schneider Electric, una estación de trabajo utilizando estos autómatas y motores industriales que puede sustituir o complementar a la maqueta del ascensor.

Otra idea posible es realizar los proyectos usando diferentes lenguajes de programación de autómatas, como por ejemplo el proyecto 2, destinado a utilizarse en prácticas, sustituir el

lenguaje SFC por el lenguaje ST permitiendo a los alumnos adquirir conocimientos de diversos lenguajes.

Al igual que con los proyectos del disco de inercia, también es posible realizar el control del ascensor utilizando diferentes lenguajes de programación de autómatas como LD o realizando una programación similar a la del proyecto 4 mediante SFC.

Finalmente, podría desarrollarse un programa de control para el ascensor utilizando un eje de velocidad en vez del eje de posición utilizado en este trabajo.

8. Glosario.

Abreviatura	Significado
PLC	Programable Logic Computer (Automata programable)
TIA Portal	Totatly Integrated Automation Portal (Portal de automatización totalmente integrado)
CPU	Central Processing Unit (Unidad central de procesamiento)
DB	Data Block (Bloque de datos)
FC	Function (Función)
FB	Function Block (Bloque de función)
OB	Organization Block (Bloque de organización)
E/S	Entradas y salidas
PTO	Pulse Train Output (Salida en tren de pulsos)
PWM	Pulse Width Modulation (Modulación por ancho de impulsos)
HSC	High Speed Counter (Contador rapido)
HMI	Human Machine Interface (Interfaz hombre maquina)
DC	Direct Current (Corriente continua)
AC	Altern Current (Corriente alterna)
RPM	Revoluciones por minuto
GND	Ground (Tierra)
FC Inf	Final de carrera inferior
FCB	Final de carrera piso 0
FC1	Final de carrera piso 1
FC2	Final de carrera piso 2
FC Sup	Final de carrera superior
PI	Pulsador auxiliar I
PF	Pulsador auxiliar F
PB	Pulsador piso 0
P1	Pulsador piso 1
P2	Pulsador piso 2
NO	Normally Opened (Normalmente abierto)
NC	Normally Closed (Normalmente cerrado)
SFC	Sequential Function Char
LD	Ladder Diagram (Diagrama de contactos)
ST	Structured Text (Texto estructurado)
PCB	Printed Circuit Board (Placa de circuito impreso)
GERBER	Formato que permite la fabricación de las placas PCB.

9. Bibliografía.

- [1] A. Malo González, «Control de posición de ejes electromecánicos mediante autómatas programables,» 2020.
- [2] M. Sesma Castán, «Control de una maqueta de un ascensor en tiempo real mediante Simulink,» 2022.
- [3] Siemens AG, «Simatic S7 CPU 1512C-1 PN (6ES7512-1CK01-0AB0): Manual de producto.,» 2021.
- [4] Siemens AG, «Siemens, catálogo de paneles de mando,» [En línea]. Available: <https://mall.industry.siemens.com/mall/es/es/Catalog/Product/6AV2124-0GC01-0AX0>.
- [5] F. G. y. B. C. Kuo, Appendix D: Control Lab, Automatic Control Systems, 10th ed., McGraw-Hill Education, 2017.
- [6] R. W. K. K. D. B. R. Philohome, «Philohome.com,» [En línea]. Available: <https://www.philohome.com/nxtmotor/nxtmotor.htm>.
- [7] Adafruit, «RJ12 Jack Connectors - EV3/NXT LEGO Compatible,» [En línea]. Available: <https://www.adafruit.com/product/4434>.
- [8] Texas Instruments, «ULN2003AD Datasheet.,» 2022.
- [9] DFROBOT, «Metal Proximity Switch,» [En línea]. Available: <https://wiki.dfrobot.com/5V%20Metal%20Proximity%20Switch%20SKU%3A%20FIT0658>.
- [10] Siemens AG, «Siemens, catalogo de dispositivos de control,» [En línea]. Available: <https://mall.industry.siemens.com/mall/es/WW/Catalog/Product/6ES7512-1CK01-0AB0>.
- [11] R. N. M Olmo, «Momento de Inercia: Disco Delgado,» [En línea]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbasees/tdisc.html>.
- [12] Siemens AG, «S7-1500 Motion Control V4.0 en el TIA Portal V15: Manual de funciones,» 2017.
- [13] D. Madeo, Dirección, *ESERCIZIO: Parte1 Configurazione PWM su Siemens S7 1512C con ingresso Analogico simulato Tia Portal*. [Película]. 2022.
- [14] D. Madeo, Dirección, *ESERCIZIO: Parte2 Configurazione PWM su Siemens S7 1512C con ingresso Analogico simulato Tia Portal*. [Película]. 2022.
- [15] U. P. d. V. -. UPV, Dirección, *Organización de un proyecto en TIA Portal V15 1 | | UPV*. [Película]. 2022.
- [16] infoPLC, «El contador rápido para encoder “HSC” del S7-1200,» [En línea]. Available: <https://www.infopl.net/descargas/103-siemens/automatas/s7-1200/1852-el-contador-r%C3%A1pido-%E2%80%9Chsc%E2%80%9D-del-s7-1200>.
- [17] ON Semiconductor, «LB1836M Datahseet,» 2013.
- [18] Texas Instruments, «SN74AC14D Datasheet,» 2023.

- [19] GAPTEC, «LM78_09-2.0R Datasheet,» 2022.
- [20] STMicroelectronics, «LM7805CV Datasheet,» 2004.
- [21] Mouser, «Conector DB25 Amphenol Communication Solutions L77SDB25SVFC309,» [En línea]. Available: https://www.mouser.es/ProductDetail/Amphenol-Commercial-Products/L77SDB25SVFC309?qs=mq7kV%2Fq8lk5U43DEAWoAlg%3D%3D&mgh=1&vip=1&gclid=CjwKCAjwxaanBhBQEiwA84TVXKJo5OA0VQCEjLI2H7FFm3_EX0On7diPLI3TzywF5sIN3Ky3QgNjNxoCDc4QAvD_BwE.
- [22] Mouser, «Conector DB9 TE-Connectivity 2301838-2,» [En línea]. Available: https://www.mouser.es/ProductDetail/TE-Connectivity/2301838-2?qs=YCa%2FAAYMW00TbWYD0LSiFQ%3D%3D&mgh=1&vip=1&gclid=CjwKCAjwxaanBhBQEiwA84TVXFTmXj2HGEYvURoFByMPdJrKy9ADg5nQpbwJIFk10uGSE0QpTnLvEhoCk8UQAvD_BwE.
- [23] Siemens AG, «Sistema de ayuda de TIA Portal».

Apéndice A. Introducción a TIA Portal.

En este apéndice se va a realizar una breve presentación e introducción al software TIA Portal, en su versión V15.1.

Para comenzar, la secuencia para crear un programa podría ser:

1. Crear el proyecto.
2. Elegir la CPU.
3. Añadir entradas y salidas.
4. Configurar los elementos de la CPU.
5. Añadir y configurar el objeto tecnológico.
6. Crear el programa de control.

A.1. Creación del proyecto y configuración inicial.

Al ejecutar TIA Portal, lo primero que aparece es la pantalla de selección de proyecto, Figura A.1, donde se puede crear un nuevo proyecto, abrir uno existente o migrar un proyecto. También se puede consultar los paquetes instalados, la ayuda o cambiar el idioma de la interfaz. Se incluye también un tutorial sobre el uso del programa.

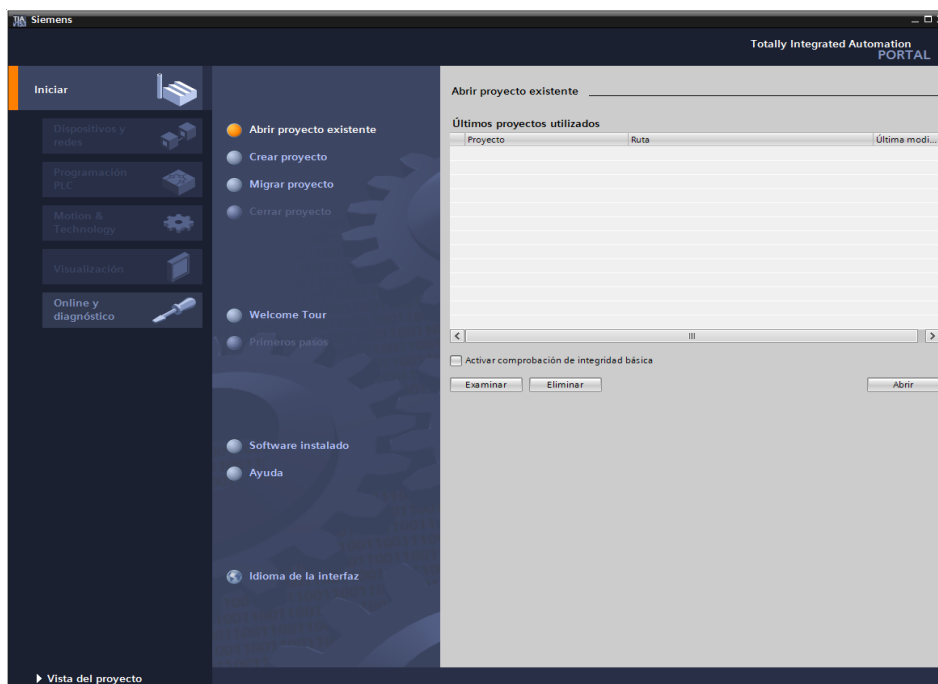


Figura A.1. Pantalla inicial de TIA Portal.

Para crear un nuevo proyecto simplemente es necesario ir a la pestaña crear proyecto, donde aparecerá el asistente para crearlo, Figura A.2. En él, se puede escoger nombre del proyecto, dirección donde guardarlo, el autor, un comentario acerca del proyecto y, en el caso de haber más de una versión de TIA Portal instalada, se puede elegir la versión.

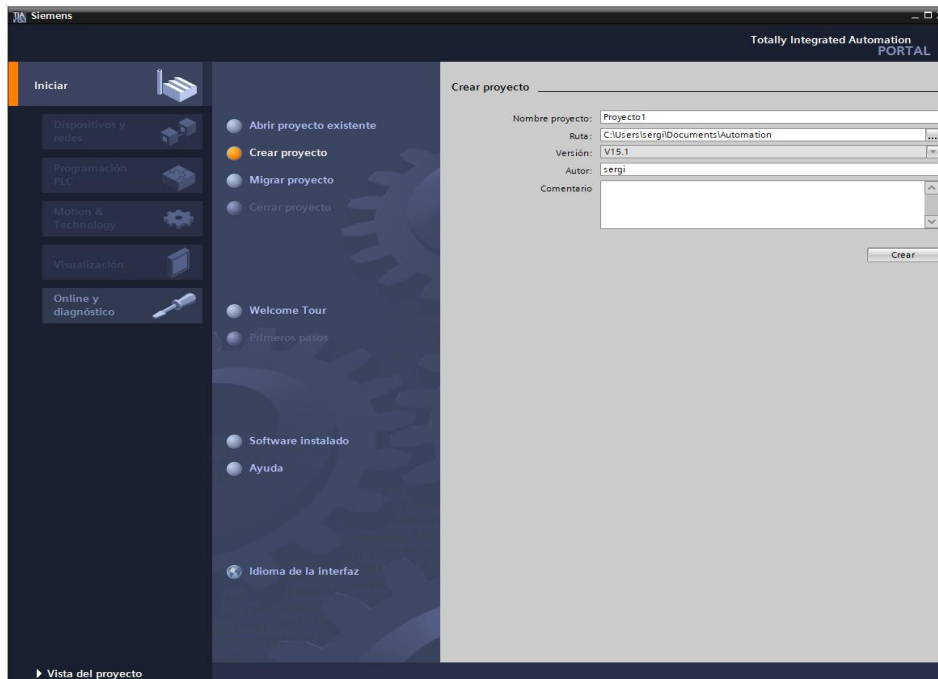


Figura A.2. Pantalla de creación de proyecto.

Una vez se abre o se crea el proyecto, aparece la pantalla inicial del proyecto, Figura A.3. En esta pantalla se puede configurar nuevos dispositivos como una nueva CPU o una pantalla HMI. También se puede escribir el programa para el PLC, crear un nuevo objeto tecnológico o configurar la pantalla HMI. Tener al menos una CPU es necesario para crear un objeto tecnológico o un programa.

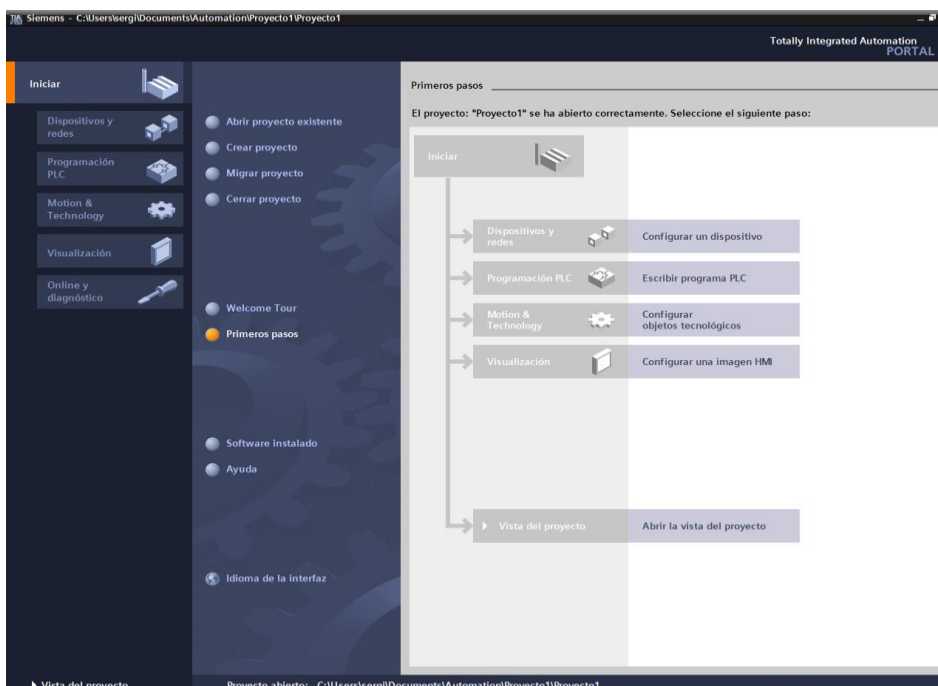


Figura A.3. Pantalla inicial del proyecto.

A.2. Interfaz de trabajo.

Una vez que se ha creado el proyecto y elegido la CPU, aparece la interfaz de trabajo, la cual puede verse en la Figura A.4.

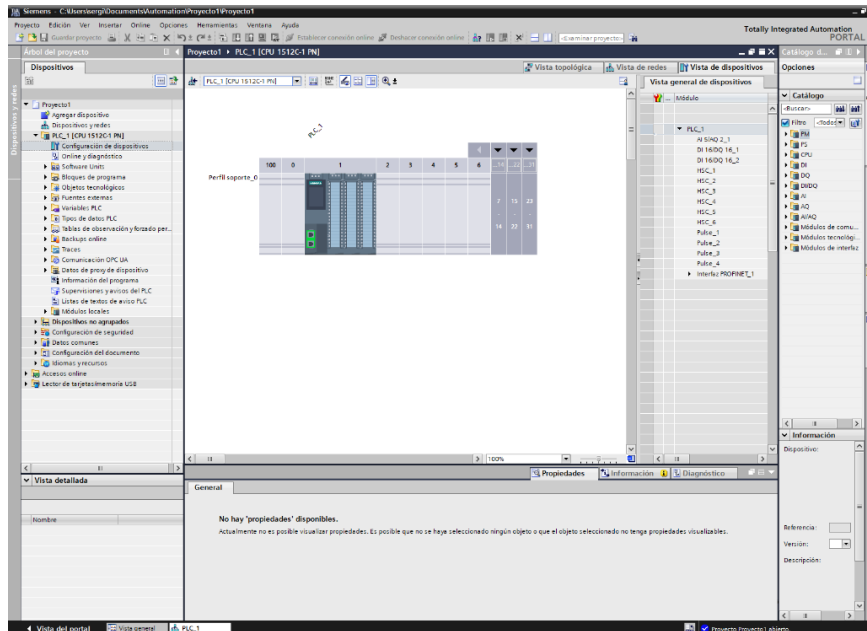


Figura A.4. Interfaz de trabajo de TIA Portal.

La interfaz de trabajo se podría dividir en varias zonas de interés, como puede verse en la Figura A.5.

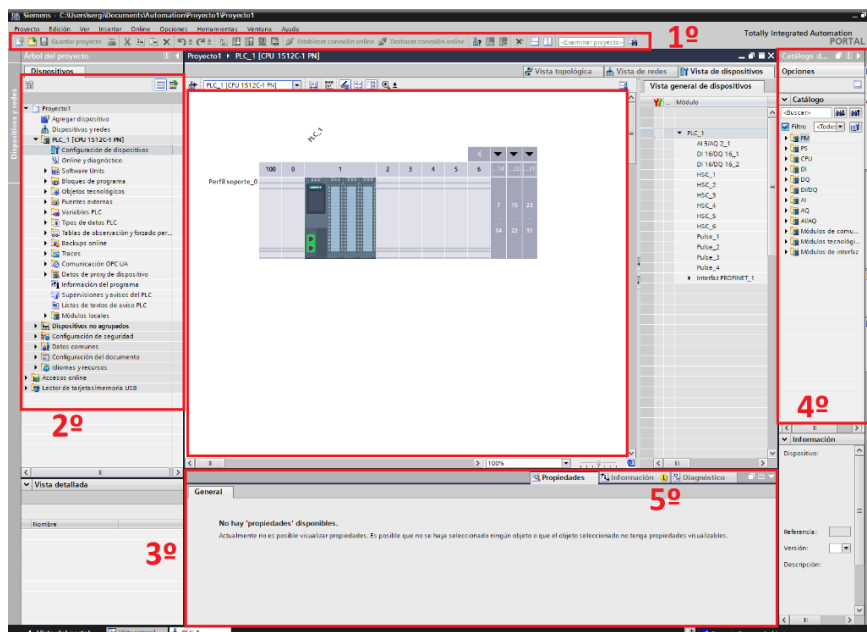


Figura A.5. División de la interfaz de trabajo de TIA Portal.

La primera zona es la barra de herramientas que, al igual que en la mayoría de los programas informáticos, contiene tareas que son de gran utilidad a la hora de trabajar con el programa. Las de mayor utilidad son:

- Crear, abrir o guardar el proyecto.
- Cortar, copiar, pegar, deshacer o rehacer.
- Transferir programa al autómatas y desde el autómatas o abrir la simulación.
- Establecer o deshacer conexión online.
- Arrancar o parar la CPU.

La segunda zona es una de las más importantes y utilizadas. En ella se encuentran los accesos a los distintos elementos de un proyecto como la gestión de los dispositivos, y dentro de esta, la gestión de variables o de objetos tecnológicos dentro de la CPU o la configuración de las distintas pantallas de operador dentro del HMI.

La tercera zona contiene los menús de propiedades, donde pueden configurarse las propiedades de algunos elementos, por ejemplo, los contadores rápidos o la interfaz PROFINet, información, donde se puede ver las advertencias y errores que se producen en los elementos y operaciones de TIA Portal y diagnóstico, donde se puede ver el estado de los dispositivos.

La cuarta zona contiene las librerías donde se encuentran las diferentes instrucciones que se usan, por ejemplo, a la hora de crear una función.

La quinta y última zona es la zona más importante de TIA Portal, la zona de trabajo. En esta zona es donde se va a realizar cualquier acción dentro de TIA Portal, como la configuración de los objetos tecnológicos o la creación del programa de control.

A.3. Añadir entradas y salidas y variables de memoria globales.

Una vez ya se tiene seleccionada la CPU y antes de configurar algunos elementos de dicha CPU, como, por ejemplo, los contadores rápidos o los generadores de pulsos, es necesario configurar las entradas y salidas del autómatas, ya que tanto los contadores rápidos como los generadores de pulsos pueden requerir de algunas entradas auxiliares. Para configurar las variables de entrada o salida y las variables globales es necesario ir a la siguiente sección mostrada en la Figura A.6.

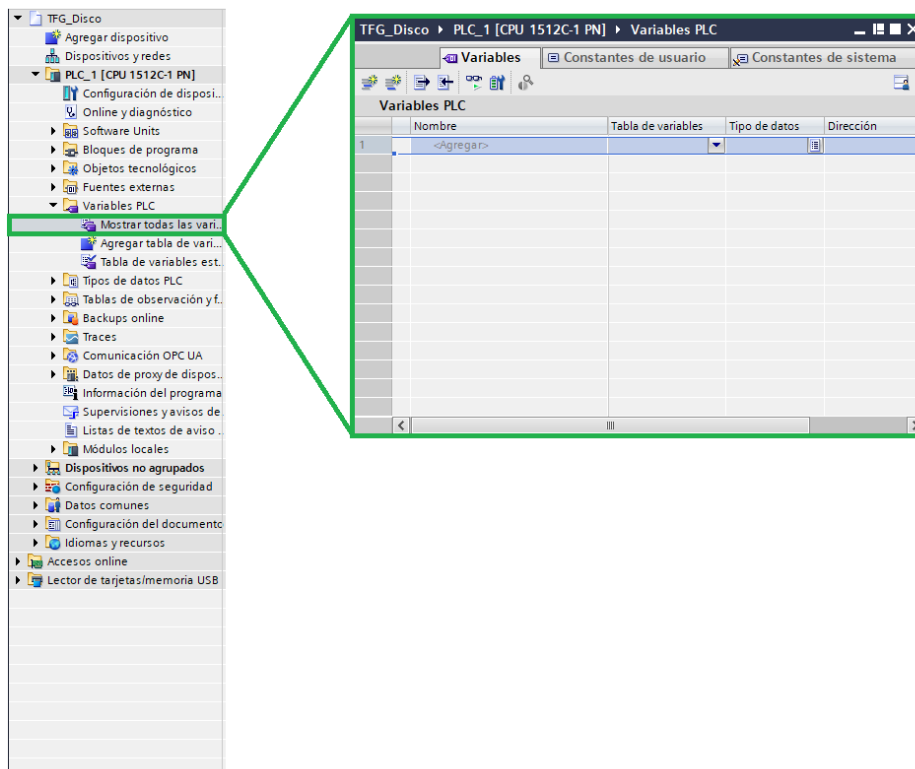


Figura A.6. Ubicación de la vista de variables en TIA Portal.

Para declarar la variable basta con hacer click en agregar y seleccionar el nombre, el tipo de dato y la dirección. Dependiendo de la dirección introducida, la variable será de entrada, de salida o variable global siendo:

- I para variable de entrada.
- Q para variable de salida.
- M para variable de memoria global.

A.4. Configurar los elementos de la CPU

Una vez ya se tienen las entradas y salidas necesarias del autómata configuradas, ya se pueden configurar los contadores rápidos y los generadores de pulsos. Para ello, hay que ir a la siguiente dirección:

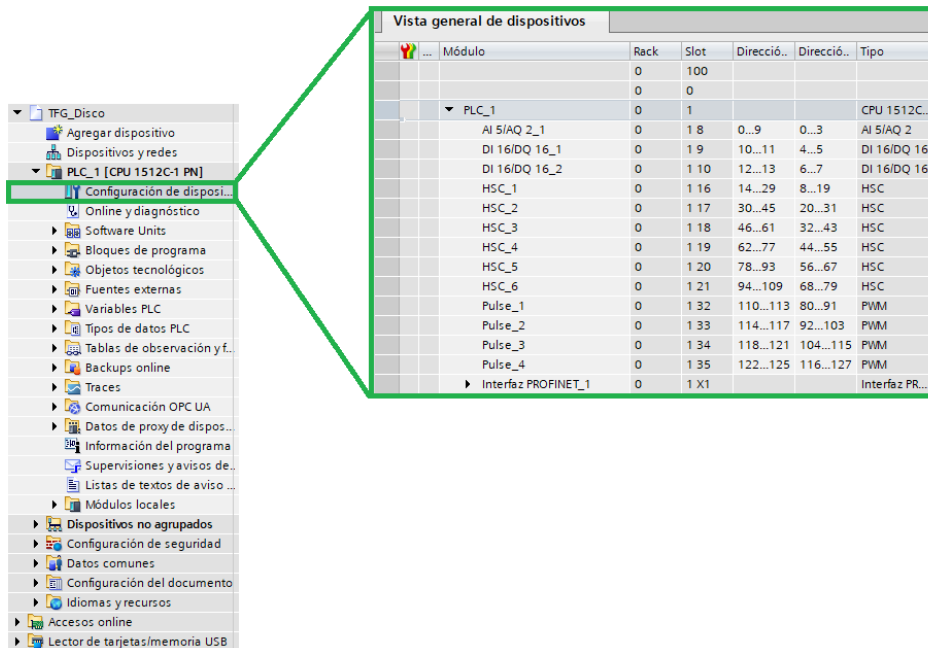


Figura A.7. Ubicación de los dispositivos en TIA Portal.

Si se desea configurar un contador rapido, se elige uno de los 6 HSC que tiene incorporado el autómeta. En cambio, si se desea configurar un generador PWM se selecciona uno de los 4 Pulse que incorpora el autómeta. La forma para configurar un contador rapido o un generador de pulsos se explica en el siguiente capitulo.

Por otro lado, si se desea configurar la dirección IP del autómeta dentro de una red es necesario abrir la interfaz PROFINET.

A.5. Objetos tecnológicos.

Una vez se han añadido las entradas y salidas y las variables globales y configurado la CPU ya se pueden crear los objetos tecnológicos necesarios. Según Siemens, los objetos tecnológicos sirven para simplificar la programación de las tareas tecnológicas. Las tareas tecnológicas son las tareas sobre las que va a actuar el autómeta como por ejemplo el movimiento de un motor o de un pistón, regulación de algún tipo de variable, por ejemplo, temperatura, si el autómeta se usa como controlador PID.

Dentro de los objetos tecnológicos se pueden encontrar 4 tipos:

- Motion Control. Estos objetos sirven para controlar accionamientos físicos, como pueden ser un motor o un pistón, si se usan ejes de velocidad o posición, o una leva o un árbol de levas.
- PIDs. Estos objetos actúan, como su nombre indica, como controladores PID.
- Simatic Ident. Estos objetos sirven para controlar sistemas de identificación, como escáneres de códigos de barras.
- Contaje y medidas. Estos objetos sirven para controlar sistemas de conteo como contadores de alta velocidad o encóder absolutos externos a los que contiene el autómeta.

Para crear un objeto tecnológico es necesario ir a la sección siguiente:

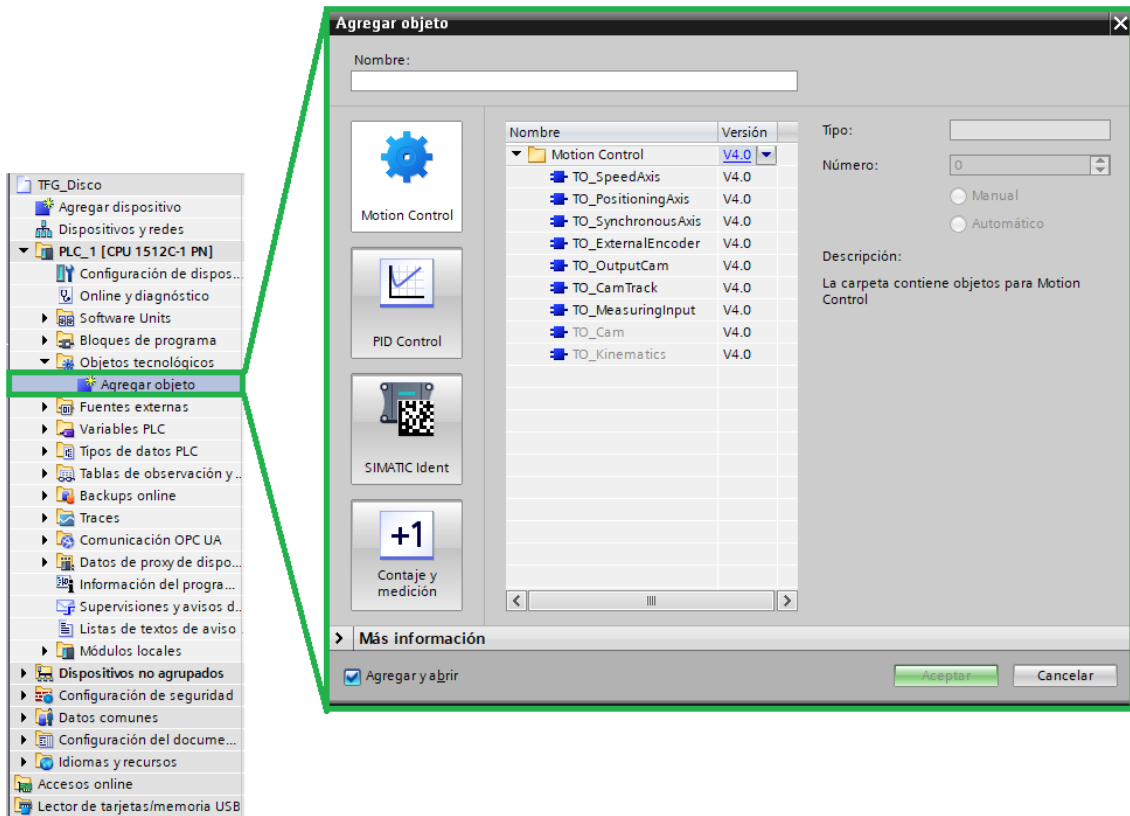


Figura A.8. Ubicación de los objetos tecnológicos en TIA Portal.

Una vez seleccionado el objeto tecnológico, aparece la pantalla de configuración de dicho objeto y ya se puede configurar.

A.6. Creación del programa.



Figura A.9. Tipos de bloques de programación.

Cuando ya se tiene configurado un Objeto tecnológico, se pasa a programar el control del PLC sobre dicho objeto. La programación del autómatas mediante TIA Portal se realiza mediante bloques. Existen 4 tipos de bloques:

- Bloques de organización (OB).
- Bloque de Función (FB).
- Función (FC).
- Bloque de datos (DB).

A.6.1. Bloque de organización (OB)

El primer tipo de bloque y quizás el más importante es el bloque de organización o bloque OB. En la ayuda de TIA Portal se puede encontrar más información acerca de cada bloque. En el caso de los bloques OB la definición que da Siemens es:

Los bloques de organización (OBs) constituyen la interfaz entre el sistema operativo y el programa de usuario y son llamados por el sistema operativo.

En otras palabras, los bloques OB son los bloques principales de ejecución del programa. Dentro de los bloques OB existen varios tipos, aunque en el trabajo se han utilizado únicamente 3:

- Main (OB1).
- MC-Servo (OB91).
- MC- MC-Interpolator (OB92).

El OB1 o Main es el bloque más importante a la hora de elaborar el programa de control ya que es el principal bloque de ejecución, desde él se llaman al resto de las funciones que componen el programa de control. Para que una función, ya sea FC o FB sea utilizada en el programa, debe estar en el OB1. En el caso que se utilice una función FB, cuando se arrastra al OB1, este crea un bloque de datos o DB asociado a la función correspondiente.

EL siguiente bloque OB es el bloque OB91 o MC-Servo. Este bloque va asociado a un objeto tecnológico y se crea junto a él. Este bloque se encarga de calcular el regulador de posición.

El último bloque OB es el bloque OB92 o MC-Interpolator. Este bloque, al igual que el OB91, va asociado a un objeto tecnológico y se crea junto a él. Este bloque se encarga de evaluar las instrucciones de Motion Control, que se verán más adelante, generar las consignas, en este caso posición y velocidad y como función de vigilancia sobre el objeto tecnológico.

Estos dos bloques, el OB91 y el OB92, están protegidos (con protección de know-how), es decir, no se puede ni ver ni escribir el código de los programas. Además, se ejecutan en una relación de 1:1 aunque el bloque OB91 se ejecuta siempre antes que el OB92. El único ajuste que se les puede cambiar es el periodo de ejecución, que por defecto es de 4 ms.

A.6.2. Bloque de datos (DB)

Antes de explicar los bloques FB y FC es necesario conocer el funcionamiento de los bloques DB, ya que los bloques FB contienen este tipo de bloques. Según Siemens la función de los bloques de datos es:

“Al contrario que los bloques lógicos, los bloques de datos no contienen instrucciones, sino que sirven para almacenar datos de usuario.”

Como dice Siemens, los bloques DB son bloques que sirven como memoria para almacenar variables y pueden ser leídos o escritos por el resto de los bloques del programa. De este tipo de bloques existen 3 distintos:

- Instancia individual.
- Multi-instancia.
- Instancia de parámetro.

La diferencia que existe entre los 3 tipos es que los de instancia individual se generan individualmente y cada función o instrucción tiene el suyo independientemente del resto mientras que los de Multi-instancia van asociados al bloque de la función principal que contiene a las funciones o instrucciones. Los de instancia de parámetro son independientes de la función y se le pasa a la función como variable de entrada / salida.

A.6.3. Bloques de Funciones (FB)

Los siguientes bloques son los bloques FB, que es donde se escribe el programa de control del autómatas. La definición que da Siemens de cada uno de estos bloques es la siguiente:

*“Los bloques de función (FB) son bloques lógicos que depositan sus variables de entrada, salida y de entrada/salida, así como las variables estáticas, de forma permanente en bloques de datos de instancia, de modo que **continúan disponibles tras la ejecución del bloque.**”*

La información de estos bloques, al estar asociada con un DB, no se pierde al finalizar el ciclo de ejecución, sino que es permanente.

En estos bloques se pueden seleccionar los lenguajes de programación:

- ST, que en el autómatas corresponde al lenguaje SCL.
- LD, que en el autómatas corresponde al lenguaje KOP.
- SFC, que en el autómatas corresponde al lenguaje Graph.

Estos bloques no permiten devolver ningún valor al finalizar la ejecución de dicho bloque.

A.6.4. Funciones (FC)

Los últimos bloques son los bloques FC que, al igual que con los bloques FB, es donde se escribe el programa de control del autómatas. La definición que da Siemens de cada uno de estos bloques es la siguiente:

*“Las funciones (FC) son bloques lógicos sin memoria. **No poseen una memoria de datos que permita almacenar valores de parámetros de bloque.**”*

La principal diferencia con el bloque FB es que el bloque FB posee un bloque DB asociado donde guarda los datos mientras que el bloque FC no. Por tanto, una vez ejecutada la función del bloque FC, la información que posee se borra. Si se desea que un bloque FC guarde datos, se le debe asignar manualmente un bloque DB.

En estos bloques se pueden seleccionar los lenguajes de programación:

- ST, que en el autómatas corresponde al lenguaje SCL.
- LD, que en el autómatas corresponde al lenguaje KOP.

Estos bloques permiten devolver un valor al finalizar la ejecución de dicho bloque.

Una vez conocido cada bloque ya se puede escribir el programa de control. Para crear una nueva función es necesario hacer lo siguiente:

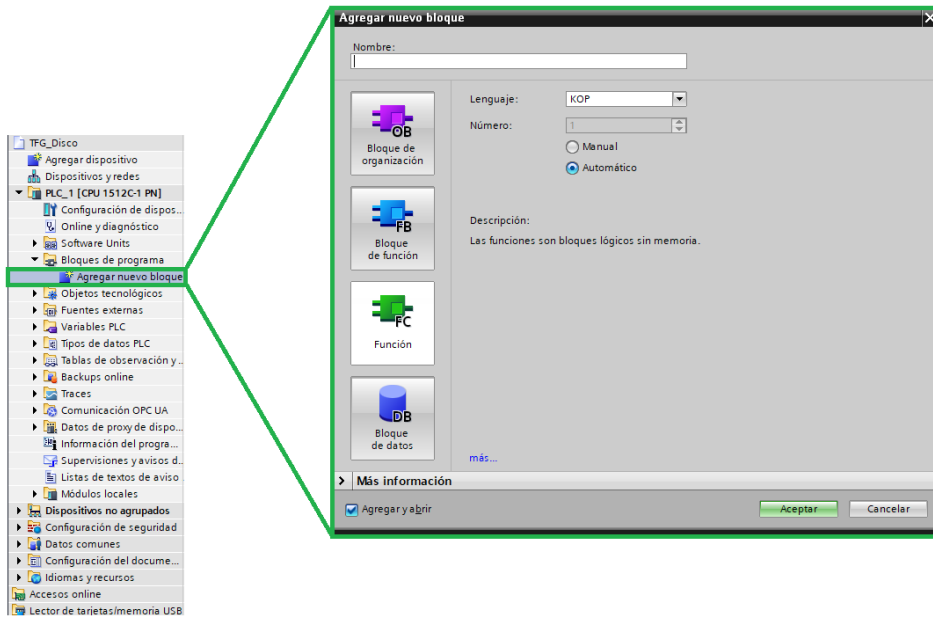


Figura A.10. Ubicación de los bloques de programa en TIA Portal.

A.7. Interfaz de trabajo del programa.

Una vez creada la función, la interfaz de trabajo es la siguiente:

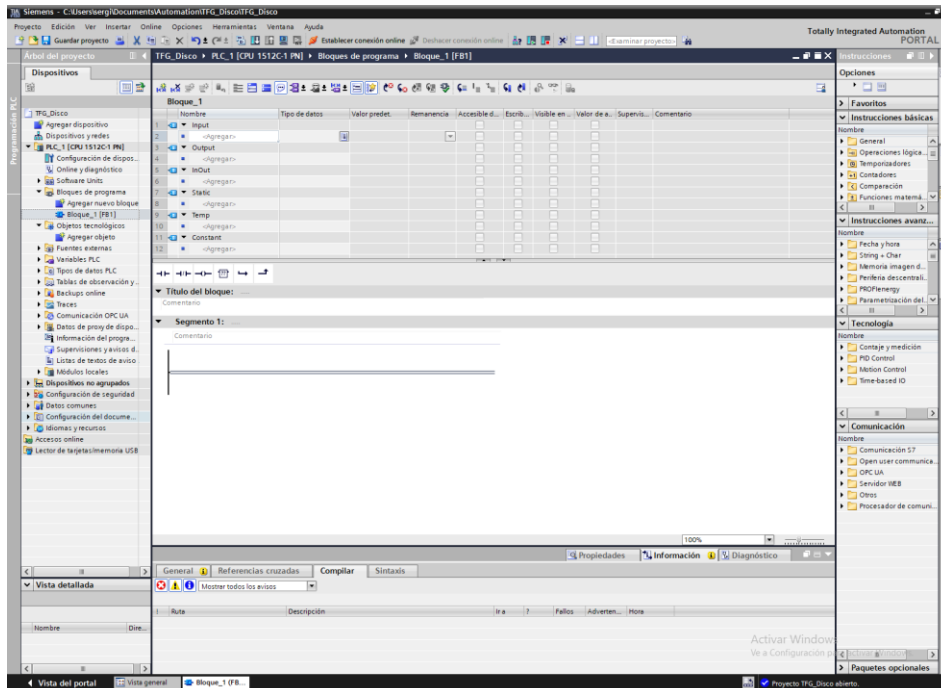


Figura A.11. Interfaz de trabajo de una función en TIA Portal.

Dentro de la interfaz de trabajo se destacan 4 zonas de utilidad, las cuales, son las siguientes:

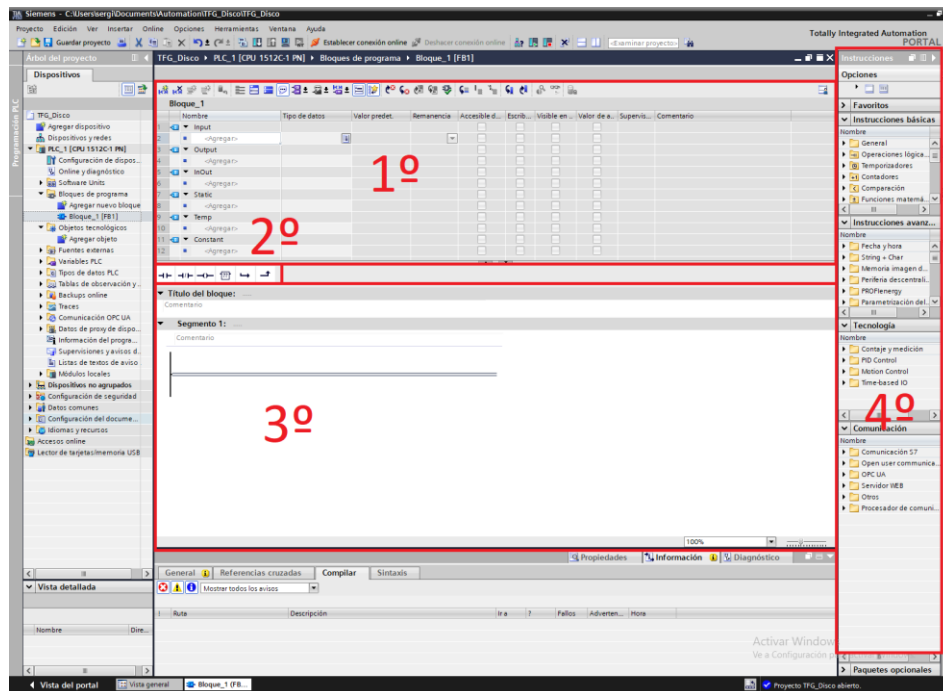


Figura A.12. Interfaz de trabajo de una función en TIA Portal dividida en secciones.

1. La primera zona corresponde a la configuración de las variables locales que se van a utilizar dentro del bloque. Estas variables pueden ser:
 - Input. Estas son las variables de entrada de datos al bloque.
 - Output. Estas son las variables de salida de datos del bloque.
 - InOut. Estas variables pueden usarse como variables de entrada y de salida.
 - Static. Estas variables no están disponibles en los bloques FC al carecer de bloque DB incorporado y almacenan los datos de forma indefinida.
 - Temp. Estas variables solo almacenan los datos mientras dure el ciclo de ejecución. Una vez terminado el ciclo la variable vuelve a su valor predefinido.
 - Constant. Esto no es una variable, sino que, como su nombre indica, son constantes que no se pueden modificar.
 - Return. Esta variable solo está disponible en los bloques FC y hace que la función devuelva un valor determinado.

Bloque_1										
	Nombre	Tipo de datos	Valor predet.	Remanencia	Accesible d...	Escrib...	Visible en ..	Valor de a..	Supervis...	Comentario
1	Input									
2	<Agregar>									
3	Output									
4	<Agregar>									
5	InOut									
6	<Agregar>									
7	Static									
8	<Agregar>									
9	Temp									
10	<Agregar>									
11	Constant									
12	<Agregar>									

Figura A.13. Lista de variables de una función en TIA Portal.

Para configurar una nueva variable basta con hacer click en agregar en el tipo de variable que se quiera crear y elegir el nombre, el tipo de datos y el valor predeterminado si fuese necesario.

2. La segunda zona corresponde a la de funciones básicas para realizar el programa, las cuales, son:
 - Contacto normalmente abierto. Este contacto solo se activa cuando el estado de la variable, de tipo booleana, que lo activa es true.
 - Contacto normalmente cerrado. Este contacto solo se activa cuando el estado de la variable, de tipo booleana, que lo activa es false.
 - Bobina. Activa la variable, de tipo booleana, asignada.
 - Cuadro vacío. Este cuadro permite añadir una función de la librería de instrucciones simplemente buscándola por su nombre.
 - Abrir rama. Como su nombre indica permite abrir una nueva rama de programación.
 - Cerrar rama. Como su nombre indica permite cerrar una rama de programación.



Figura A.14. Utilidades básicas para crear una función en TIA Portal.

3. La tercera zona se corresponde con el área de programación. Esta área se encuentra dividida en segmentos independientes donde a cada uno se le puede añadir un título y una descripción. En cada segmento, se pueden añadir los elementos antes mencionados, o elementos de la librería de instrucciones.

A parte, se pueden añadir segmentos con lenguajes de programación distintos al definido en la función.



Figura A.15. Segmento de trabajo de una función en TIA Portal.

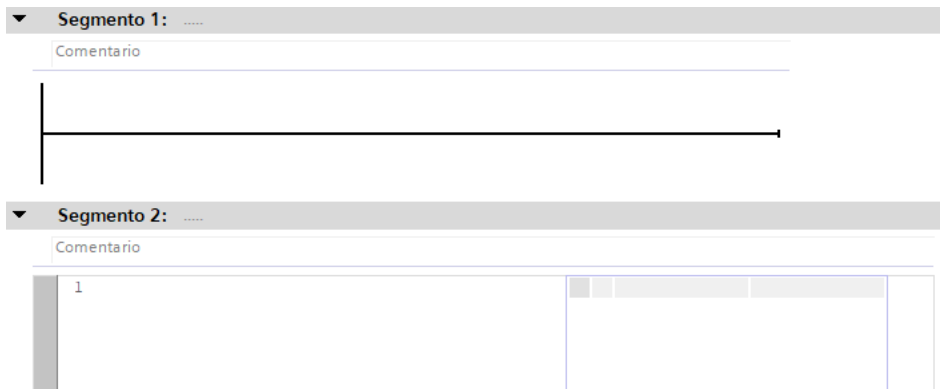


Figura A.16. Sección de trabajo de una función en TIA portal.

4. La cuarta zona se corresponde con la librería de instrucciones. Estas librerías son:
- Instrucciones básicas. En esta librería se encuentran las instrucciones más utilizadas a la hora de escribir el programa como operaciones lógicas, temporizadores, contadores o comparadores.
 - Instrucciones avanzadas. En esta librería se encuentran instrucciones como fecha y hora, alarmas o diagnóstico.
 - Tecnología. En esta librería se encuentran instrucciones relacionadas con el control de los objetos tecnológicos como PID Control o Motion Control. Esta última es la que se va a utilizar para el control de las maquetas.
 - Comunicación. En esta librería se encuentran las instrucciones que permiten al autómatas controlar los protocolos de conexión.

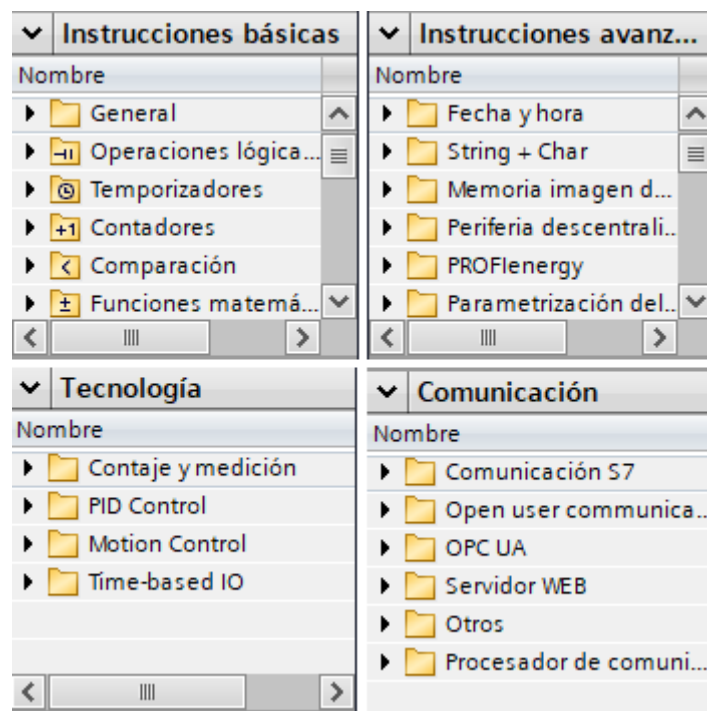


Figura A.17. Librería de instrucciones de TIA Portal.

Con todo esto, ya se podría crear el programa de control de ambas maquetas.

Apéndice B. Obtención de los parámetros dinámicos de ambas maquetas.

En este apéndice, se va a mostrar cómo se han obtenido algunas de los parámetros necesarios para realizar correctamente la configuración de la dinámica de los objetos tecnológicos que representan los motores de ambas maquetas. Para ello, se han llevado a cabo una serie de pruebas en ambas maquetas y se han obtenido unos datos que, posteriormente, han servido para calcular dichos parámetros. Algunos de los parámetros calculados han sido:

- Velocidad máxima.
- Velocidad mínima.
- Velocidad nominal.
- Tiempo de aceleración o aceleración.
- Tiempo de deceleración o deceleración.
- Tirón.

B.1. Dinámica del motor en la maqueta del disco de inercia.

B.1.1. Rango de velocidades.

Para poder obtener la velocidad nominal con la que se va a mover el disco o la de referencia con la que se va a posicionar el disco es necesario obtener primero el rango de velocidades que permite el bloque NXT.

La forma de obtener esta velocidad ha sido la siguiente:

Para obtener la velocidad máxima, se conecta el bloque NXT de la maqueta al nivel de tensión máxima con la que trabaja el motor, que es de 9 V y que se corresponde con un valor de duty del 100%. Una vez que el motor está girando, se lee, mediante el osciloscopio, la frecuencia de la señal N del inductivo ya que con cada pulso significa que el disco ha dado una vuelta.

El proceso para obtener la velocidad mínima es similar, se va aplicando una tensión desde 0 V hasta que el motor comienza a moverse, y se mantiene en movimiento sin detenerse. En este caso la tensión ha sido de 2 V, que es el mínimo nivel de tensión con el que el motor se ha mantenido girando en ambos sentidos incluso con perturbaciones.

Una vez se tiene la frecuencia de los pulsos del inductivo, es necesario transformar estas frecuencias en los valores de velocidad máxima y mínima de la siguiente manera:

Los valores, tanto de la velocidad máxima como de la mínima, pueden calcularse mediante la Ecuación B.1.

$$\omega \left[\frac{\text{vueltas}}{\text{min}} \right] = f \left[\frac{\text{vueltas}}{\text{s}} \right] * 60 \left[\frac{\text{s}}{\text{min}} \right]$$

Ecuación B.1. Ecuación que determina la velocidad angular del disco de inercia en vueltas por minuto.

Donde ω es la velocidad angular del eje de salida del bloque NXT y f es la frecuencia de los pulsos del sensor inductivo.

También es necesario para la correcta configuración del objeto tecnológico obtener, mediante la Ecuación B.2, el valor de la velocidad máxima en $\frac{^\circ}{s}$.

$$\omega \left[\frac{^\circ}{s} \right] = f \left[\frac{\text{vueltas}}{s} \right] * 360 \left[\frac{^\circ}{\text{vuelta}} \right]$$

Ecuación B.2. Ecuación que determina la velocidad angular del disco de inercia en grados por segundo.

Una vez que se conocen las ecuaciones se han efectuado 5 pruebas por cada sentido de giro para determinar la velocidad, tanto para la velocidad mínima como para la máxima. En el caso de la velocidad máxima, las frecuencias han coincidido para ambos sentidos. En el caso de la velocidad mínima se ha tomado la mínima que se ha obtenido. Los resultados se encuentran en la Tabla B.1.

MEDICIONES DE FRECUENCIA OBTENIDAS - DISCO		
Número de medición	Frecuencia (Vmax)	Frecuencia (Vmin)
1	2,5 Hz	290,7 mHz
2	2,78 Hz	280,9 mHz
3	2,22 Hz	274,7 mHz
4	2,5 Hz	274,7 mHz
5	2,5 Hz	263,2 mHz
Media	2,5 Hz	276,84 mHz

Tabla B.1. Mediciones de frecuencia para la maqueta del disco.

Conocida la frecuencia de los pulsos del inductivo ya se pueden obtener las velocidades máxima, en $\frac{\text{vueltas}}{\text{min}}$, Ecuación B.3, y en $\frac{^\circ}{s}$, Ecuación B.4, y mínima, en $\frac{^\circ}{s}$, Ecuación B.5.

$$\omega_{max} \left[\frac{\text{vueltas}}{\text{min}} \right] = f \left[\frac{\text{vueltas}}{s} \right] * 60 \left[\frac{s}{\text{min}} \right] = 2,5 * 60 = 150 \frac{\text{vueltas}}{\text{min}}$$

Ecuación B.3. Ecuación que determina el valor numérico de la velocidad máxima del disco de inercia en vueltas por minuto.

$$\omega_{max} \left[\frac{^\circ}{s} \right] = f_{max} \left[\frac{\text{vueltas}}{s} \right] * 360 \left[\frac{^\circ}{\text{vuelta}} \right] = 2,5 * 360 = 900 \frac{^\circ}{s}$$

Ecuación B.4. Ecuación que determina el valor numérico de la velocidad máxima del disco de inercia en grados por segundo.

$$\omega_{min} \left[\frac{^\circ}{s} \right] = f_{min} \left[\frac{\text{vueltas}}{s} \right] * 360 \left[\frac{^\circ}{\text{vuelta}} \right] = (276,84 * 10^{-3}) * 360 = 99,66 \frac{^\circ}{s}$$

Ecuación B.5. Ecuación que determina el valor numérico de la velocidad mínima del disco de inercia en grados por segundo.

Por tanto, la velocidad máxima del disco es de $900 \frac{\circ}{s}$ o $150 \frac{vueltas}{min}$ y la mínima es de $99,66 \frac{\circ}{s}$.

B.1.2. Velocidad nominal.

Inicialmente, se escogió el valor de la velocidad máxima como velocidad nominal, pero tras varios avisos de adaptación de la dinámica y la aparición de la alarma tecnológica 110 por parte del autómat, se buscó información sobre esta alarma en la ayuda del programa y es lo siguiente:

“En caso de conexión analógica del accionamiento, corrija el valor de referencia en el accionamiento y en la configuración del objeto tecnológico a "<TO>.Actor.MaxSpeed" / 1.17”

Por tanto, la velocidad nominal tiene que ser 1,17 veces más pequeña que la velocidad máxima, Ecuación B.6, por lo que se procede a calcularla en la Ecuación B.7.

$$\omega_{nom} \left[\frac{\circ}{s} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{1,17}$$

Ecuación B.6. Ecuación que determina la velocidad angular nominal del disco de inercia en grados por segundo.

$$\omega_{nom} \left[\frac{\circ}{s} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{1,17} = \frac{900}{1,17} = 769,23 \frac{\circ}{s}$$

Ecuación B.7. Ecuación que determina el valor de la velocidad angular del disco de inercia en grados por segundo.

También es necesario para la correcta configuración del objeto tecnológico la velocidad del eje de salida del bloque NXT en $\frac{vueltas}{min}$, por lo que puede adaptarse la ecuación B.6 para obtener el valor en $\frac{vueltas}{min}$, la Ecuación B.8. El valor numérico resultante se obtiene en la Ecuación B.9.

$$\omega_{nom} \left[\frac{vueltas}{min} \right] = \frac{\omega_{max} \left[\frac{vueltas}{min} \right]}{1,17}$$

Ecuación B.8. Ecuación que determina la velocidad angular nominal del disco de inercia en vueltas por minuto.

$$\omega_{nom} \left[\frac{vueltas}{min} \right] = \frac{\omega_{max} \left[\frac{vueltas}{min} \right]}{1,17} = \frac{150}{1,17} = 128,20 \frac{vueltas}{min}$$

Ecuación B.9. Ecuación que determina el valor de la velocidad angular del disco de inercia en vueltas por minuto.

Por tanto, las velocidades nominales son $769,23 \frac{\circ}{s}$ para el disco y $128,20 \frac{vueltas}{min}$ para el eje de salida del bloque NXT.

B.1.3. Velocidades de aproximación y de referenciado.

Una vez que se han calculado las velocidades máximas y mínimas se puede elegir las velocidades de referenciado y de aproximación. En este caso se ha decidido que la velocidad de aproximación sea de $100 \frac{\circ}{s}$ y la de referenciado de $120 \frac{\circ}{s}$. Ambas velocidades están dentro del rango y al ser bajas, permiten posicionar el eje con precisión.

B.1.4. Tiempos de aceleración y deceleración y valores de la aceleración y deceleración nominales.

Dado que TIA Portal permite elegir el valor de la aceleración y, por tanto, el propio programa calcula el tiempo de aceleración o viceversa, se ha decidido elegir el tiempo de aceleración y que TIA Portal calcule el valor de la aceleración. Las consideraciones que se han tenido en cuenta para elegir los tiempos son las siguientes:

- Para no forzar ni a la electrónica que controla al motor ni al propio motor, tanto el tiempo de aceleración como el de deceleración se han elegido lo suficientemente altos para que los cambios de referencia de velocidad se comporten como una rampa y no como un escalón.
- El tiempo de deceleración se ha elegido alto para obtener mayor precisión en la detención, es decir, que el error de posición sea lo más pequeño posible.
- Como al comienzo del movimiento no se requiere ningún tipo de precisión el tiempo de aceleración se ha elegido algo más pequeño que el de deceleración.

Finalmente, los tiempos elegidos son:

- 0,75 segundos para la aceleración.
- 0,75 segundos para la deceleración.

Una vez conocidos los tiempos, se puede calcular tanto la aceleración como la deceleración mediante las ecuaciones Ecuación B.10 Ecuación B.11 respectivamente y sus respectivos valores numéricos en las ecuaciones Ecuación B.12 Ecuación B.13.

$$\alpha_{a_{nom}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{nom} \left[\frac{\circ}{s} \right]}{t_{a_{nom}} [s]}$$

Ecuación B.10. Ecuación que determina la aceleración angular nominal del disco de inercia.

$$\alpha_{d_{nom}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{nom} \left[\frac{\circ}{s} \right]}{t_{d_{nom}} [s]}$$

Ecuación B.11. Ecuación que determina la deceleración angular nominal del disco de inercia.

Donde $\alpha_{a_{nom}}$ es el valor de la aceleración angular nominal, $t_{a_{nom}}$ es el tiempo de aceleración nominal, $\alpha_{d_{nom}}$ es el valor de la deceleración angular nominal y $t_{d_{nom}}$ el tiempo de deceleración nominal.

$$\alpha_{a_{nom}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{nom} \left[\frac{\circ}{s} \right]}{t_{a_{nom}} [s]} = \frac{769,23}{0,75} = 1025,64 \frac{\circ}{s^2}$$

Ecuación B.12. Ecuación que determina el valor de la aceleración angular nominal del disco de inercia.

$$\alpha_{d_{nom}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{nom} \left[\frac{\circ}{s} \right]}{t_{d_{nom}} [s]} = \frac{769,23}{0,75} = 1025,64 \frac{\circ}{s^2}$$

Ecuación B.13. Ecuación que determina el valor de la deceleración angular nominal del disco de inercia.

Por tanto, el tiempo de aceleración nominal es de 0,75 s y el valor de la aceleración angular nominal es de $1025,64 \frac{\circ}{s^2}$, mientras que el tiempo de deceleración nominal es de 0,75 s y el valor de la deceleración angular nominal es de $1025,64 \frac{\circ}{s^2}$.

B.1.5. Tiempos de aceleración y deceleración y valores de la aceleración y deceleración máximos.

Al igual que con los valores nominales, se han tomado unos valores concretos basados en unas consideraciones previas similares a las de los valores nominales:

- En este caso, como estos valores no van a ser usados por norma general, como en la parada de emergencia, se puede elegir un valor más bajo que fuerce algo más al motor.
- A diferencia de con los valores nominales, no se necesita precisión en la parada ni en un supuesto comienzo del movimiento.

Por tanto, los valores elegidos han sido:

- 0,25 segundos para el tiempo de deceleración.
- 0,25 segundos para el tiempo de aceleración.

Como se ha hecho con los valores nominales, se puede calcular el valor de la aceleración y deceleración máximas mediante las ecuaciones Ecuación B.14 Ecuación B.15 respectivamente y sus respectivos valores en las ecuaciones Ecuación B.16 Ecuación B.17.

$$\alpha_{a_{max}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{t_{a_{min}} [s]}$$

Ecuación B.14. Ecuación que determina la aceleración angular máxima del disco de inercia.

$$\alpha_{d_{max}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{t_{d_{min}} [s]}$$

Ecuación B.15. Ecuación que determina la deceleración angular máxima del disco de inercia.

Donde $\alpha_{a_{max}}$ es el valor de la aceleración angular máxima, $t_{a_{max}}$ es el tiempo de aceleración máxima, $\alpha_{d_{max}}$ es el valor de la deceleración angular máxima y $t_{d_{max}}$ el tiempo de deceleración máxima.

$$\alpha_{a_{max}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{t_{a_{max}} [s]} = \frac{900}{0,25} = 3600 \frac{\circ}{s^2}$$

Ecuación B.16. Ecuación que determina el valor de la aceleración angular máxima del disco de inercia.

$$\alpha_{d_{max}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{t_{d_{max}} [s]} = \frac{900}{0,25} = 3600 \frac{\circ}{s^2}$$

Ecuación B.17. Ecuación que determina el valor de la deceleración angular máxima del disco de inercia.

Por tanto, el tiempo de aceleración nominal es de 0,25 s y el valor de la aceleración angular nominal es de $3600 \frac{\circ}{s^2}$, mientras que el tiempo de deceleración nominal es de 0,25 s y el valor de la deceleración angular nominal es de $3600 \frac{\circ}{s^2}$.

B.1.6. Tiempo de deceleración de parada de emergencia y valor de la deceleración.

Este valor es usado por el autómatas para detener el motor en caso de una parada de emergencia y puede ser superior al valor de deceleración máximo. Al igual que con los valores de aceleración nominales y máximo, puede elegirse este valor o bien, el tiempo que le cuesta realizar la parada. Este valor ha sido elegido arbitrariamente, aunque siempre menor al valor máximo. El valor elegido ha sido de 0,1 segundos.

Como se ha hecho con los valores máximos y nominales, se va a calcular el valor de la deceleración de emergencia, Ecuación B.18 y su valor, Ecuación B.19.

$$\alpha_{d_{emerg}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{t_{d_{emerg}} [s]}$$

Ecuación B.18. Ecuación que determina la deceleración angular de emergencia del disco de inercia.

Donde $\alpha_{d_{emerg}}$ es el valor de la deceleración de emergencia y $t_{d_{emerg}}$ es el tiempo de la deceleración de emergencia.

$$\alpha_{d_{emerg}} \left[\frac{\circ}{s^2} \right] = \frac{\omega_{max} \left[\frac{\circ}{s} \right]}{t_{d_{emerg}} [s]} = \frac{900}{0,1} = 9000 \frac{\circ}{s^2}$$

Ecuación B.19. Ecuación que determina el valor de la deceleración angular de emergencia del disco de inercia.

Por tanto, el valor de la deceleración de emergencia es de $9000 \frac{\circ}{s^2}$ y el tiempo es de 0,1 segundos.

B.1.7. Jerk nominal y Jerk máximo y tiempos de redondeo.

El jerk es la rapidez con la que cambia la aceleración y, por tanto, el tiempo de redondeo es el tiempo que le cuesta a la aceleración pasar de 0,0 a un valor determinado.

A diferencia de los tiempos de aceleración y deceleración, este valor se ha elegido arbitrariamente sin un criterio establecido.

Los valores elegidos son:

- 0,1 segundos para el tiempo de redondeo nominal.
- 0,05 segundos para el tiempo de redondeo máximo.

B.2. Dinámica del motor en la maqueta del ascensor.

B.2.1. Rango de velocidades.

Al igual que con la maqueta del disco es necesario conocer las velocidades máximas y mínimas para poder obtener el resto de los parámetros dinámicos, como la velocidad de referenciado o las aceleraciones.

El proceso para obtener las velocidades máximas y mínimas es muy similar al de la maqueta del disco, solo que esta vez, al no disponer de sensor inductivo, es necesario leer la frecuencia de los pulsos del encoder mediante el osciloscopio. El procedimiento para obtener dichas velocidades es el siguiente:

El primer paso es, mediante la Ecuación B.20, convertir la frecuencia de los pulsos del encóder a velocidad.

$$\omega \left[\frac{\text{vueltas}}{s} \right] = \frac{f \left[\frac{\text{pulsos}}{s} \right]}{n \left[\frac{\text{pulsos}}{\text{vuelta}} \right]}$$

Ecuación B.20. Ecuación que determina la velocidad angular del motor del ascensor en vueltas por segundo.

Donde ω es la velocidad angular del eje de salida del bloque NXT, f es la frecuencia de los pulsos del encóder y n es el número de pulsos por vuelta del encóder.

Otra forma de expresar la velocidad angular, y que es necesaria para configurar correctamente el encóder en TIA Portal, es en revoluciones por minuto y se puede calcular mediante la Ecuación B.21.

$$\omega \left[\frac{\text{vueltas}}{\text{min}} \right] = \omega \left[\frac{\text{vueltas}}{s} \right] * 60 \left[\frac{s}{\text{min}} \right]$$

Ecuación B.21. Ecuación que determina la velocidad angular del motor del ascensor en vueltas por minuto.

Una vez conocidas las fórmulas que determinan la velocidad angular en revoluciones por segundo y por minuto del eje de salida del bloque NXT ya se puede conocer la velocidad lineal del ascensor mediante la Ecuación B.22.

$$v \left[\frac{\text{mm}}{s} \right] = \omega \left[\frac{\text{vueltas}}{s} \right] * \frac{Z_m}{Z_a} * r \left[\frac{\text{mm}}{\text{vuelta}} \right]$$

Ecuación B.22. Ecuación que determina la velocidad del ascensor en milímetros por segundo.

Donde v es la velocidad del ascensor, ω es la velocidad del eje de salida del bloque NXT, Z_m el número de dientes del engranaje acoplado al eje de salida del bloque NXT, Z_a el número de dientes del engranaje acoplado a la polea del ascensor y r el avance de la polea.

Para simplificar los cálculos, pueden agruparse las ecuaciones Ecuación B.20 Ecuación B.21 en la Ecuación B.23 para obtener ω en revoluciones por minuto a partir de la frecuencia del encóder medida y las ecuaciones Ecuación B.20 Ecuación B.22 en la Ecuación B.24 para obtener la velocidad v en mm/s a partir de la frecuencia del encóder medida y del diámetro primitivo de la rueda de la polea.

$$\omega \left[\frac{\text{vueltas}}{\text{min}} \right] = \frac{f \left[\frac{\text{pulsos}}{s} \right]}{n \left[\frac{\text{pulsos}}{\text{vuelta}} \right]} * 60 \left[\frac{s}{\text{min}} \right]$$

Ecuación B.23. Ecuación que determina la velocidad angular del motor del ascensor en vueltas por minuto.

$$v \left[\frac{mm}{s} \right] = \frac{f \left[\frac{pulsos}{s} \right]}{n \left[\frac{pulsos}{vuelta} \right]} * \frac{Z_m}{Z_a} * r \left[\frac{mm}{vuelta} \right]$$

Ecuación B.24. Ecuación que determina la velocidad angular del ascensor en milímetros por segundo.

Una vez se tienen todas las ecuaciones necesarias se realizan las pruebas para obtener los valores. Se han realizado 5 mediciones por cada sentido, cuyos resultados se recogen en la Tabla B.2.

MEDICIONES DE FRECUENCIA OBTENIDAS - ASCENSOR		
Número de medición	Frecuencia (Vmax)	Frecuencia (Vmin)
1	400 Hz	100 Hz
2	400 Hz	125 Hz
3	375 Hz	125 Hz
4	400 Hz	125 Hz
5	425 Hz	150 Hz
Media	400 Hz	125 Hz

Tabla B.2. Mediciones de frecuencia para la maqueta del ascensor.

Una vez conocidas tanto las ecuaciones como los valores necesarios ya se puede obtener el valor de velocidad máxima del eje de salida del bloque NXT en $\frac{vueltas}{min}$, Ecuación B.25, y del ascensor en mm/s , Ecuación B.26, y de la velocidad mínima del ascensor en mm/s , Ecuación B.27, a partir de las ecuaciones Ecuación B.23 Ecuación B.24.

$$\omega_{max} \left[\frac{vueltas}{min} \right] = \frac{f_{max} \left[\frac{pulsos}{s} \right]}{n \left[\frac{pulsos}{vuelta} \right]} * 60 \left[\frac{s}{min} \right] = \frac{400}{180} * 60 = 133,33 \frac{vueltas}{min}$$

Ecuación B.25. Ecuación que determina el valor numérico de la velocidad angular máxima del motor del ascensor en vueltas por minuto.

$$v_{max} \left[\frac{mm}{s} \right] = \frac{f_{max} \left[\frac{pulsos}{s} \right]}{n \left[\frac{pulsos}{vuelta} \right]} * \frac{Z_m}{Z_a} * r \left[\frac{mm}{vuelta} \right] = \frac{400}{180} * \frac{8}{24} * 80 = 59,26 \frac{mm}{s}$$

Ecuación B.26. Ecuación que determina el valor numérico de la velocidad máxima ascensor en milímetros por segundo.

$$v_{min} \left[\frac{mm}{s} \right] = \frac{f_{min} \left[\frac{pulsos}{s} \right]}{n \left[\frac{pulsos}{vuelta} \right]} * \frac{Z_m}{Z_a} * r \left[\frac{mm}{vuelta} \right] = \frac{125}{180} * \frac{8}{24} * 80 = 18,52 \frac{mm}{s}$$

Ecuación B.27. Ecuación que determina el valor numérico de la velocidad mínima del ascensor en milímetros por segundo.

Por tanto, la velocidad máxima del ascensor es de $59,26 \frac{mm}{s}$ o $133,33 \frac{vueltas}{min}$ y la mínima es de $18,52 \frac{g}{s}$.

B.2.2. Velocidad nominal.

Como se ha visto en el cálculo de la velocidad nominal para la maqueta del disco, esta tiene que ser 1,17 veces menor a la velocidad máxima, Ecuación B.28, por lo que se procede a calcularla en la Ecuación B.29.

$$v_{nom} \left[\frac{mm}{s} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{1,17}$$

Ecuación B.28. Ecuación que determina la velocidad nominal del ascensor en milímetros por segundo.

$$v_{nom} \left[\frac{mm}{s} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{1,17} = \frac{59,26}{1,17} = 50,65 \frac{mm}{s}$$

Ecuación B.29. Ecuación que determina el valor de la velocidad nominal del ascensor en milímetros por segundo.

Dado que la velocidad obtenida se encuentra en el rango antes mencionado, se puede dar por válida.

También es necesario para la correcta configuración del objeto tecnológico la velocidad del eje de salida del bloque NXT en $\frac{vueltas}{min}$, por lo que puede adaptarse la ecuación B.28 para obtener el valor en $\frac{vueltas}{min}$, la Ecuación B.30. El valor numérico resultante se obtiene en la Ecuación B.31.

$$\omega_{nom} \left[\frac{vueltas}{min} \right] = \frac{\omega_{max} \left[\frac{vueltas}{min} \right]}{1,17}$$

Ecuación B.30. Ecuación que determina la velocidad angular nominal del motor del ascensor en vueltas por minuto.

$$\omega_{nom} \left[\frac{vueltas}{min} \right] = \frac{\omega_{max} \left[\frac{vueltas}{min} \right]}{1,17} = \frac{133,33}{1,17} = 113,95 \frac{vueltas}{min}$$

Ecuación B.31. Ecuación que determina el valor de la velocidad angular nominal del motor del ascensor en vueltas por minuto.

Por tanto, la velocidad nominal del ascensor es de $50,65 \frac{mm}{s}$ y la nominal del eje de salida del bloque NXT es de $113,95 \frac{vueltas}{min}$.

B.2.3. Tiempos de aceleración y deceleración y valores de la aceleración y deceleración nominales.

Al igual que con la maqueta del disco es necesario seleccionar los tiempos de aceleración y deceleración, aunque en este caso en $\frac{mm}{s}$. Las consideraciones para elegir estos tiempos son las mismas que con la otra maqueta. Los tiempos elegidos en este caso son:

- 1,1 segundos para la aceleración.
- 1,25 segundos para la deceleración.

Una vez conocidos los tiempos, se puede calcular tanto la aceleración como la deceleración mediante las ecuaciones Ecuación B.32 y Ecuación B.33 y sus respectivos valores numéricos en las ecuaciones Ecuación B.34 y Ecuación B.35.

$$a_a \left[\frac{mm}{s^2} \right] = \frac{v_{nom} \left[\frac{mm}{s} \right]}{t_a [s]}$$

Ecuación B.32. Ecuación que determina el valor de la aceleración nominal del ascensor

$$a_d \left[\frac{mm}{s^2} \right] = \frac{v_{nom} \left[\frac{mm}{s} \right]}{t_d [s]}$$

Ecuación B.33. Ecuación que determina la aceleración nominal del ascensor.

Donde a_a es el valor de la aceleración, t_a es el tiempo de aceleración, a_d es el valor de la deceleración y t_d el tiempo de deceleración.

$$a_a \left[\frac{mm}{s^2} \right] = \frac{v_{nom} \left[\frac{mm}{s} \right]}{t_a [s]} = \frac{50,65}{1,1} = 46,04 \frac{mm}{s^2}$$

Ecuación B.34. Ecuación que determina el valor de la aceleración nominal del ascensor.

$$a_d \left[\frac{mm}{s^2} \right] = \frac{v_{nom} \left[\frac{mm}{s} \right]}{t_d [s]} = \frac{50,65}{1,25} = 40,52 \frac{mm}{s^2}$$

Ecuación B.35. Ecuación que determina el valor de la deceleración nominal del ascensor.

Por tanto, el tiempo de aceleración es de 1,1 s y el valor de la aceleración es de $46,04 \frac{mm}{s^2}$, mientras que el tiempo de deceleración es de 1,25 s y el valor de la deceleración es de $40,52 \frac{mm}{s^2}$.

B.2.4. Velocidades de referenciado y aproximación.

Una vez que se han calculado los tiempos de aceleración y deceleración, se puede calcular la velocidad con la que se produce el referenciado del ascensor, ya que cuando el eje se detiene cuando alcanza el punto de referencia, lo hace con estos valores.

Es importante este dato ya que, a diferencia de la maqueta del disco, que el recorrido es libre, aquí el espacio está limitado a la distancia entre los finales de carrera superior e inferior y si se acciona alguno de estos sensores se produce un error en el autómatas cuya única solución es el reiniciado del autómatas.

Como lo más probable es que el ascensor se encuentre por encima del punto de referencia, el posicionamiento se hace hacia abajo, por tanto, el sensor que puede tocar es el del límite inferior, que se encuentra a una distancia de 17 mm del sensor de referencia. Con este valor y con el de la deceleración se puede calcular la velocidad de referenciado máxima de la siguiente mediante la Ecuación B.36.

$$e_{max}[mm] = e_0[mm] + v_{refmax} \left[\frac{mm}{s} \right] * t_d[s] - \frac{1}{2} * a_d \left[\frac{mm}{s^2} \right] * (t_d[s])^2$$

Ecuación B.36. Ecuación que determina la distancia recorrida por el ascensor durante la deceleración.

Donde v_{refmax} es la velocidad de referenciado máxima posible, e_{max} es la distancia entre el sensor del límite de posición y el sensor de referencia, e_0 es la distancia inicial, a_d es el valor de la deceleración de referenciado y t_d es el tiempo de detención de referenciado.

Dado que se desconoce el valor de t_d , la Ecuación B.36 tiene 2 incógnitas, por tanto, para resolverla es necesario establecer alguna relación entre t_d y v_{refmax} tal y como se muestra en la Ecuación B.37.

$$t_d[s] = \frac{v_{refmax} \left[\frac{mm}{s} \right]}{a_d \left[\frac{mm}{s^2} \right]}$$

Ecuación B.37. Ecuación que determina el tiempo de deceleración del ascensor.

Por tanto, introduciendo la Ecuación B.37 en la Ecuación B.36 se obtiene la Ecuación B.38 que permite calcular el valor de la velocidad de referenciado máxima.

$$e_{max}[mm] = e_0[mm] + \frac{\left(v_{refmax} \left[\frac{mm}{s} \right] \right)^2}{2 * a_d \left[\frac{mm}{s^2} \right]}$$

Ecuación B.38. Ecuación simplificada que determina la distancia recorrida por el ascensor durante la deceleración.

Ahora, ya se puede despejar el valor de $v_{ref_{max}}$ mediante la Ecuación B.39.

$$v_{ref_{max}} \left[\frac{mm}{s} \right] = \sqrt{2 * (e_{max}[mm] - e_0[mm]) * a_d \left[\frac{mm}{s^2} \right]}$$

Ecuación B.39. Ecuación que determina el valor de la velocidad de referenciado máxima del ascensor.

Una vez conocida la ecuación se sustituye cada término por su valor correspondiente y se resuelve en la Ecuación B.40. Estos valores son:

- $e_{max} = 17 \text{ mm}$.
- $e_0 = 0 \text{ mm}$.
- $a_d = 40,52 \frac{mm}{s^2}$.

$$v_{ref_{max}} \left[\frac{mm}{s} \right] = \sqrt{2 * (17 - 0) * 40,52} = 37,12 \frac{mm}{s}$$

Ecuación B.40. Ecuación que determina el valor de la velocidad de referenciado máxima del ascensor.

Por tanto, la velocidad de referenciado máxima es de $37,12 \frac{mm}{s}$ que, como se aprecia, está dentro del rango de velocidades que permite el ascensor. Para garantizar mayor precisión a la hora de detenerse ya que la ventana en la que la cabina activa el sensor de referencia es de unos 5 mm , es necesario que la velocidad sea lo más baja posible, por lo que se ha elegido una velocidad de referenciado y de aproximación de $22,5 \frac{mm}{s}$.

B.2.5. Intervalo de alturas excluido para la parada en la planta 1.

Para el correcto funcionamiento del ascensor en el modo memoria es necesario introducir una limitación de espacio para que, si se llama al ascensor en la planta 1, este pueda detenerse de forma suave con el valor de la deceleración configurada, Ecuación B.35. Para ello, particularizando la Ecuación B.38, es decir, cambiando $v_{ref_{max}}$ por v_{nom} se calcula la distancia recorrida durante el proceso de frenado, Ecuación B.41. Para calcularla, se tienen los siguientes valores:

- $a_d = 40,52 \frac{mm}{s^2}$.
- $e_0 = 0 \text{ mm}$.
- $v_{nom} = 50,65 \frac{mm}{s}$.

$$e_{dec}[mm] = 0 + \frac{50,65^2}{2 * 40,52} = 31,65 \text{ mm}$$

Ecuación B.41. Ecuación que determina el valor de la distancia de deceleración del ascensor.

Por tanto, sabiendo que el primer piso del ascensor se encuentra a una altura de 240 mm el intervalo restringido viene dado por las ecuaciones Ecuación B.42 Ecuación B.43 y su valor numérico por las ecuaciones Ecuación B.44 Ecuación B.45.

$$h_{min}[mm] = h_{piso1}[mm] - e_{dec}[mm]$$

Ecuación B.42. Ecuación que determina la altura mínima del intervalo de parada restringido del ascensor.

$$h_{max}[mm] = h_{piso1}[mm] + e_{dec}[mm]$$

Ecuación B.43. Ecuación que determina la altura máxima del intervalo de parada restringido del ascensor.

Donde h_{min} es la altura mínima del intervalo restringido, h_{piso1} es la altura del piso 1, e_{dec} , es el espacio recorrido durante la deceleración y h_{max} es la altura máxima del intervalo restringido.

$$h_{min}[mm] = 240 - 31,65 = 208,35 \text{ mm}$$

Ecuación B.44. Ecuación que determina el valor de la altura mínima del intervalo de parada restringido del ascensor.

$$h_{max}[mm] = 240 + 31,65 = 271,65 \text{ mm}$$

Ecuación B.45. Ecuación que determina el valor de la altura máxima del intervalo de parada restringido del ascensor.

Por tanto, el intervalo restringido dentro del cual el ascensor no se detendrá si se realiza la llamada es [208,35, 271,65] mm.

B.2.6. Tiempos de aceleración y deceleración y valores de la aceleración y deceleración máximos.

En este caso se han tomado los mismos valores que con la maqueta del disco. Los valores elegidos son:

- 0,25 segundos para el tiempo de deceleración.
- 0,25 segundos para el tiempo de aceleración.

Como se ha hecho con los valores nominales, se puede calcular el valor de la aceleración y deceleración máximas mediante las ecuaciones Ecuación B.46 Ecuación B.47 respectivamente y sus respectivos valores en las ecuaciones Ecuación B.48 Ecuación B.49.

$$a_{a_{max}} \left[\frac{mm}{s^2} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{t_{a_{max}} [s]}$$

Ecuación B.46. Ecuación que determina la aceleración máxima del ascensor.

$$a_{d_{max}} \left[\frac{mm}{s^2} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{t_{d_{max}} [s]}$$

Ecuación B.47. Ecuación que determina la deceleración máxima del ascensor.

Donde $\alpha_{a_{max}}$ es el valor de la aceleración angular máxima, $t_{a_{max}}$ es el tiempo de aceleración máxima, $\alpha_{d_{max}}$ es el valor de la deceleración angular máxima y $t_{d_{max}}$ el tiempo de deceleración máxima.

$$a_{a_{max}} \left[\frac{mm}{s^2} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{t_{a_{max}} [s]} = \frac{59,26}{0,25} = 237,04 \frac{mm}{s^2}$$

Ecuación B.48. Ecuación que determina el valor de la aceleración máxima del ascensor.

$$a_{d_{max}} \left[\frac{mm}{s^2} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{t_{d_{max}} [s]} = \frac{59,26}{0,25} = 237,04 \frac{mm}{s^2}$$

Ecuación B.49. Ecuación que determina el valor de la deceleración máxima del ascensor.

Por tanto, el tiempo de aceleración nominal es de 0,25 s y el valor de la aceleración angular nominal es de $237,04 \frac{mm}{s^2}$, mientras que el tiempo de deceleración nominal es de 0,25 s y el valor de la deceleración angular nominal es de $237,04 \frac{mm}{s^2}$.

B.2.7. Tiempo de deceleración de parada de emergencia y valor de la deceleración.

Al igual que con los valores de aceleración y deceleración máximos, se ha tomado el mismo tiempo que para la maqueta del disco, que es de 0,1 segundos.

Como se ha hecho con los valores máximos y nominales, se va a calcular el valor de la deceleración de emergencia, Ecuación B.50 y su valor, Ecuación B.51.

$$a_{emerg} \left[\frac{mm}{s^2} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{t_{d_{emerg}} [s]}$$

Ecuación B.50. Ecuación que determina la deceleración de emergencia del ascensor.

Donde $a_{d_{emerg}}$ es el valor de la deceleración de emergencia y $t_{d_{emerg}}$ es el tiempo de la deceleración de emergencia.

$$a_{d_{emerg}} \left[\frac{mm}{s^2} \right] = \frac{v_{max} \left[\frac{mm}{s} \right]}{t_{d_{emerg}} [s]} = \frac{59,26}{0,1} = 592,6 \frac{mm}{s^2}$$

Ecuación B.51. Ecuación que determina el valor de la deceleración de emergencia del ascensor.

Por tanto, el valor de la deceleración de emergencia es de $592,6 \frac{mm}{s^2}$ y el tiempo es de 0,1 segundos.

B.2.8. Tiempos de redondeo nominal y máximo.

En este caso, también se han elegido los mismos valores que para la maqueta del disco.

Los valores elegidos son:

- 0,1 segundos para el tiempo de redondeo nominal.
- 0,05 segundos para el tiempo de redondeo máximo.

B.3. Tabla resumen con los valores de la variables.

Finalmente, a modo de resumen y para agrupar datos, se incluye la tabla B.3 donde aparecen todos los datos calculados.

PARÁMETROS DINÁMICOS DE LAS MAQUETAS	
MAQUETA DEL DISCO	
Velocidad máxima	900 ° / s o 150 vueltas / min
Velocidad mínima	99,66 ° / s
Velocidad nominal	769,23 ° / s o 128,20 vueltas / min
Velocidad de referenciado	100 ° / s
Velocidad de aproximación	120 ° / s
Aceleración máxima	3600 ° / s ²
Deceleración máxima	3600 ° / s ²
Tiempo de aceleración máxima	0,25 s
Tiempo de deceleración máxima	0,25 s
Aceleración nominal	1025,64 ° / s ²
Deceleración nominal	1025,64 ° / s ²

Tiempo de aceleración nominal	0,75 s
Tiempo de deceleración nominal	0,75 s
Deceleración de emergencia	9000 ^e / s ²
Tiempo de deceleración de emergencia	0,1 s
Tiempo de redondeo máximo	0,05 s
Tiempo de redondeo nominal	0,1 s
MAQUETA DEL ASCENSOR	
Velocidad máxima	59,26 mm / s o 133,33 vueltas / min
Velocidad mínima	18,52 mm / s
Velocidad nominal	50,65 mm / s o 113,95 vueltas / min
Velocidad de referenciado	22,5 mm / s
Velocidad de aproximación	22,5 mm / s
Altura mínima del intervalo restringido para llamar al piso 1	
Altura máxima del intervalo restringido para llamar al piso 1	
Aceleración máxima	237,04 mm / s ²
Deceleración máxima	237,04 mm / s ²
Tiempo de aceleración máxima	0,25 s
Tiempo de deceleración máxima	0,25 s
Aceleración nominal	46,04 mm / s ²
Deceleración nominal	40,52 mm / s ²
Tiempo de aceleración nominal	1,1 s
Tiempo de deceleración nominal	1,25 s
Deceleración de emergencia	592,6 mm / s ²
Tiempo de deceleración de emergencia	0,1 s
Tiempo de redondeo máximo	0,05 s
Tiempo de redondeo nominal	0,1 s

Tabla B.3 Tabla resumen con todos los valores calculados de ambas maquetas.

Apéndice C. Cálculo de los elementos de la placa adaptadora de niveles de tensión.

C.1. Cálculo de resistencias pull-up para el correcto funcionamiento del autómatas.

Dado que la tensión que entiende el autómatas en sus bornes es de entre 11 y 30 V para tener un "1" lógico y la corriente de al menos 2 mA, según se ha visto en diversas pruebas realizadas, ha sido necesario calcular la resistencia de pull-up para que, junto a la corriente que demanda el autómatas, se obtenga una tensión en este intervalo.

Partiendo de la estructura que adapta los finales de carrera normalmente cerrados y considerando, dado que se está trabajando con corriente continua, el condensador con impedancia infinita, el circuito se simplifica tal y como aparece en la Figura C.1.

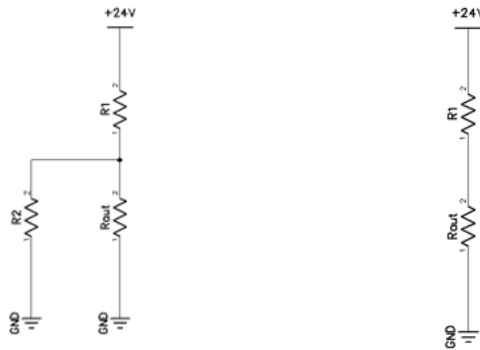


Figura C.1. Circuito simplificado de la estructura que adapta las señales entre los finales de carrera y el autómata.

Tomando la estructura de la derecha y analizando el circuito por el método de los nodos se plantea la Ecuación C.1.

$$\frac{V_{DD} - V_{"1" \min}}{R_{1 \max}} = I_{AUT}$$

Ecuación C.1. Ecuación que determina el balance de corrientes en el nodo.

Donde V_{DD} es el valor de la tensión de alimentación, $V_{"1" \min}$ es el valor mínimo de tensión que escribe un "1" lógico en el autómata, R_1 es el valor de la resistencia de pull-up e I_{AUT} es el valor de la corriente para escribir un "1" lógico en el autómata.

Transformando la ecuación anterior se obtiene la Ecuación C.2, que permite obtener el valor máximo de la resistencia R_1 .

$$\frac{V_{DD} - V_{"1" \min}}{I_{AUT}} = R_{1 \max}$$

Ecuación C.2. Ecuación que determina la resistencia máxima de pull-up.

En la ecuación anterior, se sustituyen las variables por sus respectivos valores y se obtiene la Ecuación C.3 que da el valor máximo de la resistencia de pull-up.

$$R_{1 \max} = \frac{24 - 11}{0,002} = 6,5 \text{ k}\Omega$$

Ecuación C.3. Ecuación que determina el valor máximo de la resistencia de pull-up.

Debido a que las resistencias van empaquetadas en un array de resistencias, es necesario calcular el valor mínimo de la resistencia de pull-up para garantizar que la potencia disipada por el array se encuentre por debajo del máximo permitido. Para ello, se usa la Ecuación C.4.

$$N_R * \frac{(V_{DD} - V_{n1}^{min})^2}{R_{1min}} = P_{max}$$

Ecuación C.4. Ecuación que determina la potencia máxima que puede disipar la resistencia de pull-up.

Donde N_R es el número de resistencias que componen el array, V_{DD} es el valor de la tensión de alimentación, V_{n1}^{min} es el valor mínimo de tensión que escribe un "1" lógico en el autómata, R_1 es el valor de la resistencia de pull-up y P_{max} es la potencia máxima que soporta el array.

Transformando la ecuación anterior se obtiene la Ecuación C.5, que permite obtener el valor mínimo de la resistencia R_1 .

$$\frac{N_R * (V_{DD} - V_{n1}^{min})^2}{P_{max}} = R_{1min}$$

Ecuación C.5. Ecuación que determina la mínima resistencia de pull-up.

En la ecuación anterior, se sustituyen las variables por sus respectivos valores y se obtiene la Ecuación C.6 que da el valor mínimo de la resistencia de pull-up.

$$R_{1min} = \frac{5 * (24 - 11)^2}{0,25} = 3,38 \text{ k}\Omega$$

Ecuación C.6. Ecuación que determina el valor mínimo de la resistencia de pull-up.

Finalmente, el rango disponible para elegir R_1 es de entre 3,38 k Ω y 6,5 k Ω . En este caso se ha elegido el valor de las resistencias de 5,6 k Ω , para minimizar las pérdidas disipadas por calor.

En el caso de los finales de carrera normalmente abiertos, la forma de calcular la resistencia de pull-up es la misma, porque, al fin y al cabo, la parte del circuito entre el transistor y el autómata, que es donde se encuentra la resistencia de pull-up, es la misma que para los finales de carrera normalmente cerrados. Por tanto, se ha utilizado el mismo valor de resistencia de pull-up para este caso.

En el caso del inductivo de la maqueta del disco de inercia, existía el problema de que la adaptación interna del inductivo que llevaba no era capaz de dar la tensión suficiente para el autómata, por lo que ha sido necesario sustituir la resistencia de 10 k Ω por una de 2,2 k Ω , ya que, en este caso, no hay problema con la disipación de energía y generación de calor.

C.2. Cálculo de resistencias del filtro RC para el correcto funcionamiento del autómata.

El cálculo anterior garantiza que la tensión que ve el autómata esté dentro del rango válido para interpretarlo como un "1" lógico, pero no garantiza lo mismo para el "0" lógico. Este valor viene garantizado por la resistencia del filtro RC. Por ello, tomando la estructura de la izquierda que se ve en la Figura C.1, y analizando el circuito por el método de los nodos se plantea la Ecuación C.7.

$$\frac{V_{DD} - V_{"0"_{max}}}{R_1} = \frac{V_{"0"_{max}}}{R_{2_{max}}} + I_{AUT}$$

Ecuación C.7. Ecuación que determina el equilibrio de corrientes en el nodo.

Donde V_{DD} es el valor de la tensión de alimentación, $V_{"0"_{max}}$ es el valor máximo de tensión que escribe un "0" lógico en el autómata, R_1 es el valor de la resistencia de pull-up, R_2 es la resistencia del filtro RC e I_{AUT} es el valor de la corriente para escribir un "1" lógico en el autómata.

Transformando la ecuación anterior se obtiene la Ecuación C.8, que permite obtener el valor máximo de la resistencia R_2 .

$$\frac{V_{"0"_{max}}}{\left(\frac{V_{DD} - V_{"0"_{max}}}{R_1} - I_{AUT}\right)} = R_{2_{max}}$$

Ecuación C.8. Ecuación que determina la máxima resistencia del filtro RC.

En la ecuación anterior, se sustituyen las variables por sus respectivos valores y se obtiene la Ecuación C.9 que da el valor máximo de la resistencia RC.

$$R_{2_{max}} = \frac{5}{\left(\frac{24 - 5}{5.600} - 0,002\right)} = 3,58 \text{ k}\Omega$$

Ecuación C.9. Ecuación que determina el valor máximo de la resistencia del filtro RC.

Como no existe un valor mínimo, el rango disponible para R_2 es de entre 0 y 3,58 k Ω . En este caso se ha elegido un valor de 470 Ω para R_2 .

C.3. Cálculo de resistencias limitadoras de 24 V de las bases de los transistores.

Dado que los bloques inversores que contiene el integrado ULN2003AD están diseñados para trabajar directamente con 5 V y en algunos casos se conectan a 24 V, es necesario conectar una resistencia a la entrada de la base para proteger los transistores de la corriente

excesiva. Para ello, es necesario tener en cuenta el diagrama de bloques que aparece en la hoja de características del integrado, Figura C.2.

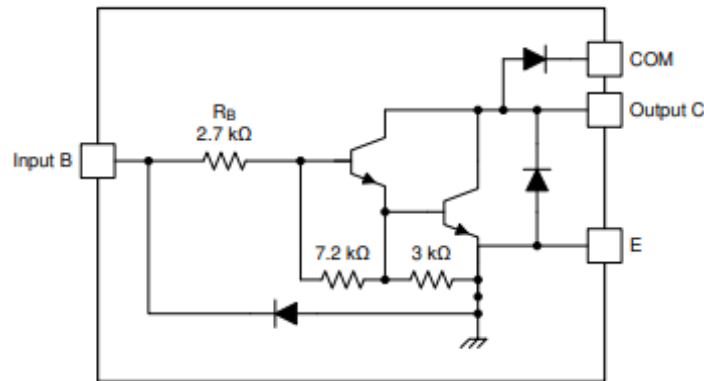


Figura C.2. Esquema del circuito que compone cada par Darlington del integrado ULN2003AD. [8]

Sabiendo que en la base del primer transistor la tensión es 2 veces V_{BE} y que en la entrada B, la tensión es de 5 V, la corriente que circula puede calcularse mediante la Ecuación C.10.

$$\frac{V_{CC} - 2 * V_{BE}}{R_B} = I_B$$

Ecuación C.10. Ecuación que determina la corriente de base del par Darlington.

Donde V_{CC} es la tensión de alimentación, V_{BE} es la caída de tensión entre la base y el emisor del transistor, R_B es la resistencia limitadora de la base e I_B es la corriente de base

Dado que para un transistor bipolar de silicio V_{BE} es aproximadamente 0,7 V, sustituyendo en la ecuación anterior por los valores de trabajo se obtiene la Ecuación C.11, que permite obtener el valor de la corriente I_B .

$$I_B = \frac{5 - 2 * 0,7}{2700} = 0,0013 \text{ A} = 1,3 \text{ mA}$$

Ecuación C.11. Ecuación que determina el valor de la corriente de base del par Darlington.

Si se desea mantener la misma corriente para 24 V, la nueva resistencia de base puede calcularse con la Ecuación C.12.

$$\frac{V_{DD} - V_{CC}}{I_B} = R_{B2}$$

Ecuación C.12. Ecuación que determina la nueva resistencia de base del par Darlington.

Donde V_{DD} es la nueva tensión de base, V_{CC} es la tensión de base de diseño, I_B es la corriente de base y R_{B2} es la nueva resistencia limitadora.

Si en la ecuación anterior se sustituye cada variable por su valor numérico se obtiene la Ecuación C.13, que da como resultado el valor de la resistencia R_{B2} .

$$R_{B2} = \frac{24 - 5}{0,0013} = 14.615,4 \Omega$$

Ecuación C.13. Ecuación que determina el valor de la nueva resistencia de base del par Darlington.

Como este valor de resistencia no existe comercialmente, el valor más aproximado son 15 k Ω . Por tanto, para proteger la base de los transistores de 24 V se ha incluido una resistencia de 15 k Ω .

C.4. Cálculo de resistencias limitadoras de los LED.

Como los LEDs de la maqueta del ascensor van conectados a una tensión superior a la que la resistencia limitadora de corriente fue diseñada, se ha incluido otra resistencia limitadora.

Antes, es necesario tener en cuenta unos detalles. La caída de tensión mínima de un LED verde es de 1,9 V mientras que la de un led rojo es de 1,63 V. Otro parámetro que es importante conocer es la corriente máxima que pasaría si se conectase a la tensión de alimentación original. Esta corriente puede calcularse con la Ecuación C.14.

$$\frac{V_{CC} - V_{LED_MIN}}{R_{LED}} = I_{LED_MAX}$$

Ecuación C.14. Ecuación que determina la corriente que pasa por el LED.

Donde V_{CC} es la tensión de alimentación original, V_{LED_MIN} es la caída de tensión mínima del LED para encenderse, R_{LED} es la resistencia limitadora de la corriente del LED e I_{LED_MAX} es la corriente máxima del LED.

Si en la ecuación anterior se sustituye cada variable con su valor correspondiente puede calcularse la corriente máxima para el LED verde, Ecuación C.15 y para el LED rojo, Ecuación C.16.

$$I_{LED_MAX_VERDE} = \frac{3,3 - 1,9}{120} = 0,0116 A = 11,6 mA$$

Ecuación C.15. Ecuación que determina el valor de la corriente que pasa por el LED verde alimentado a 3,3 V.

$$I_{LED_MAX_ROJO} = \frac{3,3 - 1,63}{120} = 0,0139 \text{ A} = 13,9 \text{ mA}$$

Ecuación C.16. Ecuación que determina el valor de la corriente que pasa por el LED rojo alimentado a 3,3 V.

Entonces, teniendo en cuenta estos valores puede calcularse la nueva resistencia mediante la Ecuación C.17.

$$\frac{V_{CC} - V_{LED_MIN}}{I_{LED}} - R_{LED} = R_{NEW}$$

Ecuación C.17. Ecuación que determina la nueva resistencia limitadora de los LED.

Donde V_{CC} es la tensión de alimentación nueva, V_{LED_MIN} es la tensión mínima para encender el LED, I_{LED} es la corriente del LED, R_{LED} es la resistencia original del LED y R_{NEW} es el valor de la nueva resistencia limitadora.

Si en la ecuación anterior se sustituye cada variable con su valor correspondiente puede calcularse la resistencia para el LED verde, Ecuación C.18, y para el LED rojo, Ecuación C.19.

$$R_{NEW_VERDE} = \frac{5 - 1,9}{0,0116} - 120 = 147,24 \Omega$$

Ecuación C.18. Ecuación que determina el valor de la nueva resistencia limitadora para el LED verde.

$$R_{NEW_ROJO} = \frac{5 - 1,63}{0,0139} - 120 = 122,44 \Omega$$

Ecuación C.19. Ecuación que determina el valor de la nueva resistencia limitadora para el LED rojo.

Como ambos dan un valor muy próximo, se ha decidido utilizar una resistencia de 120 Ω para ambos, ya que es el valor comercial más cercano.

Aunque el valor es ligeramente inferior a lo que debería, esto no representa un problema, ya que la corriente máxima de estos LEDs suele rondar los 20 mA y esta diferencia solo aumenta la corriente que circula por ellos algunos nA.

C.5. Cálculo de condensadores de los filtros RC.

Previamente se ha calculado la resistencia del filtro RC por lo que ahora es necesario calcular la capacidad del condensador para el filtro RC. Lo primero es conocer la constante de tiempo del sistema a partir del tiempo de respuesta del pulsador que se requiere que sea lo más rápido posible. Se ha estimado que para no deformar demasiado la señal debido al condensador el tiempo de respuesta sea de 1,5 ms. El valor de τ puede calcularse mediante la Ecuación C.20.

$$\tau = \frac{t}{3}$$

Ecuación C.20. Ecuación que determina la constante de tiempo de un sistema de primer orden.

Donde τ es la constante del tiempo del sistema y t es el tiempo de respuesta del sistema.

Si en esta ecuación, se sustituyen los valores, se obtiene la Ecuación C.21, que da como resultado el valor de la constante de tiempo.

$$\tau = \frac{1,5}{3} = 0,5 \text{ ms}$$

Ecuación C.21. Ecuación que determina el valor de la constante de tiempo del filtro RC.

Como este circuito se trata de un circuito RC, la constante de tiempo puede calcularse también mediante la Ecuación C.22.

$$\tau = R * C$$

Ecuación C.22. Ecuación que determina la constante del sistema de un filtro RC.

Donde τ es el valor de la constante de tiempo del sistema, R es el valor de la resistencia del sistema RC y C la capacidad del condensador del sistema RC.

De la ecuación anterior, puede despejarse el valor de C , mediante la Ecuación C.23.

$$C = \frac{\tau}{R}$$

Ecuación C.23. Ecuación que determina el valor del condensador de un filtro RC.

Finalmente, sustituyendo las variables por sus respectivos valores en la Ecuación C.23Ecuación C.24, se obtiene como resultado la Ecuación C.24, que da como resultado el valor de C buscado.

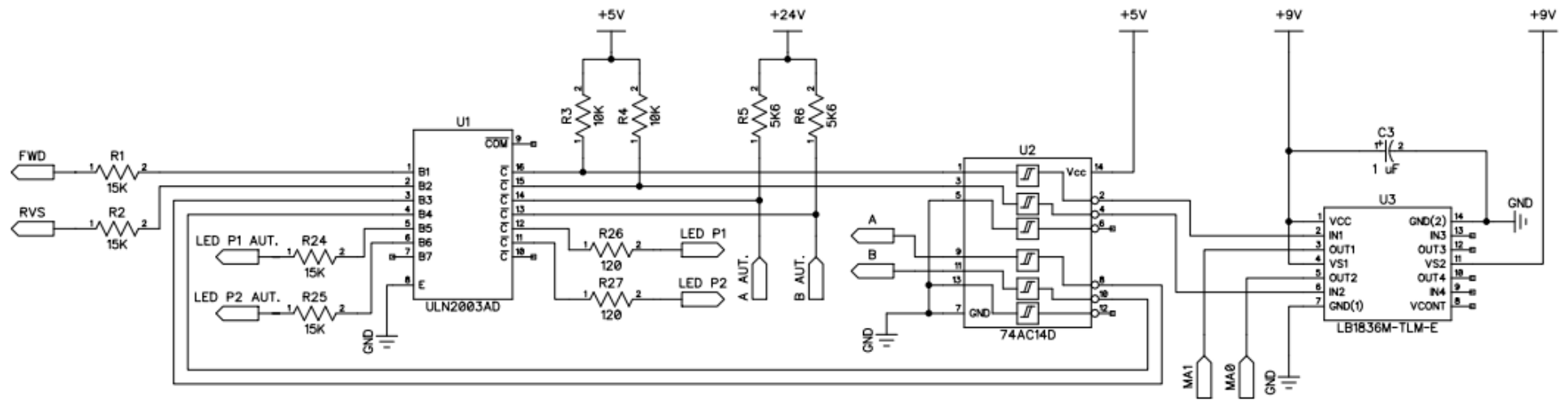
$$C = \frac{0,5}{470} \cong 1 \mu F$$

Ecuación C.24. Ecuación que determina el valor del condensador del filtro RC.

Por tanto, el valor de los condensadores de los filtros RC será de $1 \mu F$.

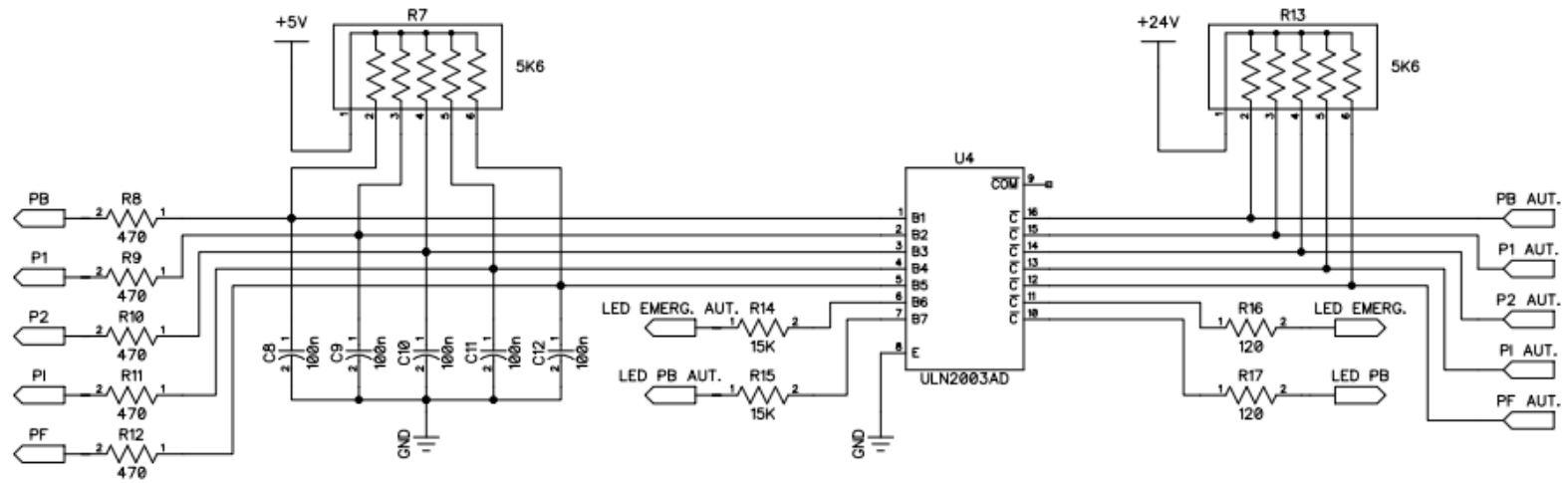
Apéndice D. Esquema completo de la placa adaptadora de niveles de tensión y PCB.

ADAPTACION PWM, ENCONDER DEL MOTOR Y LEDS

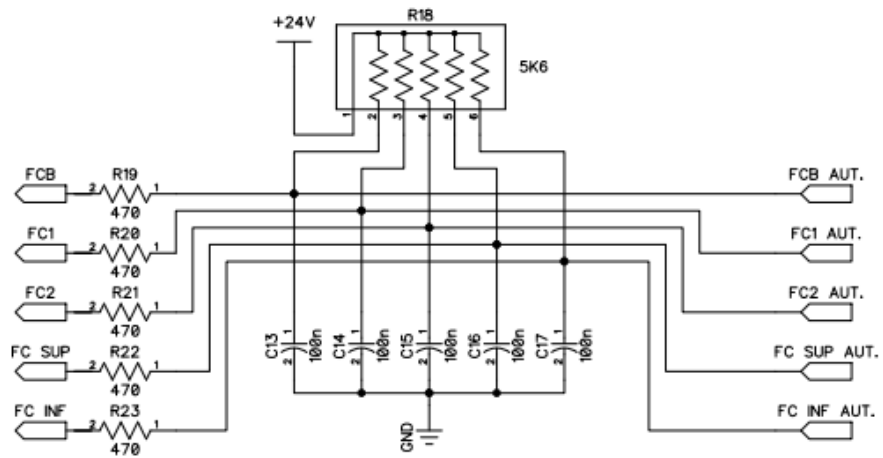


ADAPTACION PULSADORES, FINALES DE CARRERA Y LEDS

Adaptacion pulsadores y parte de LEDS

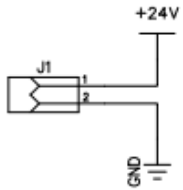


Adaptacion finales de carrera

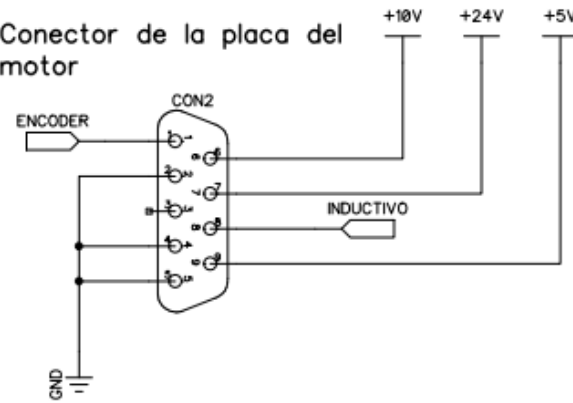


CONECTORES E INTERRUPTORES DE LA PLACA

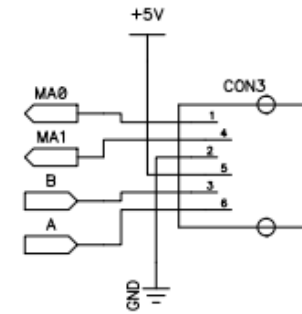
Conector de alimentacion



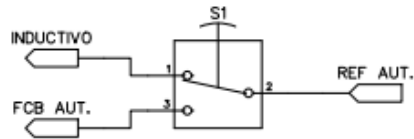
Conector de la placa del motor



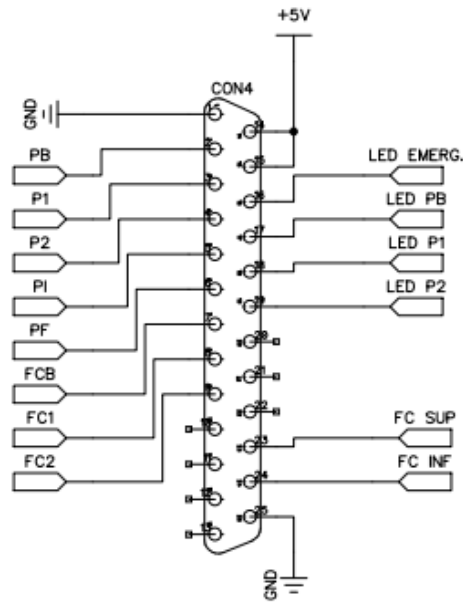
Conector del motor



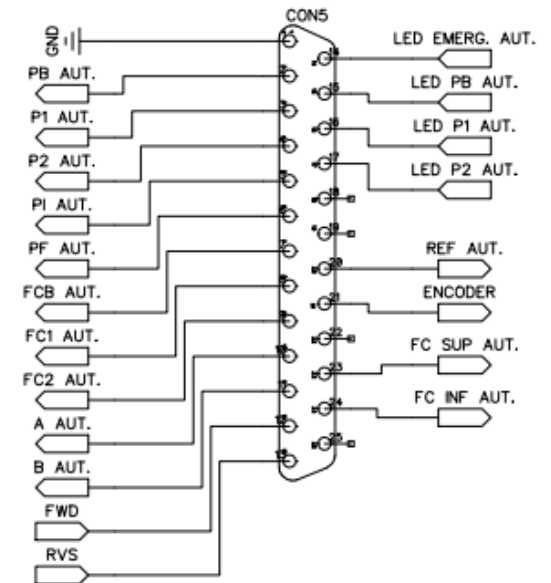
Conmutador referencia de posicion



Conector de la maqueta a la placa (DB25H)

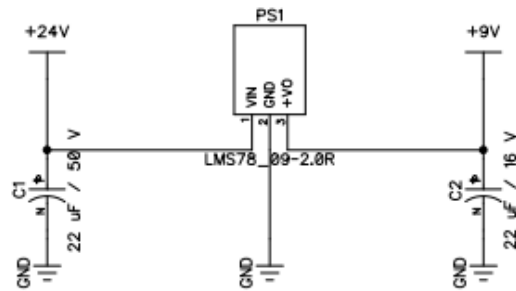


Conector de la placa al automata (DB25M)

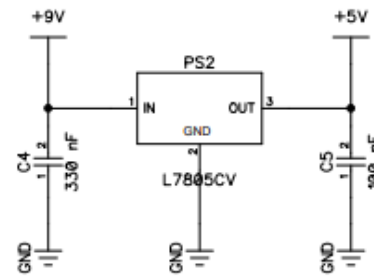


FUENTES DE ALIMENTACION

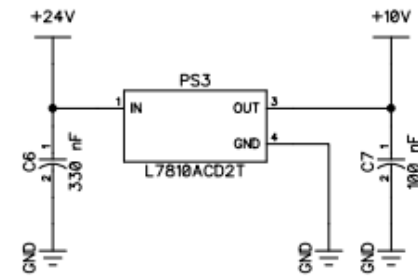
Alimentacion del motor (+9V)

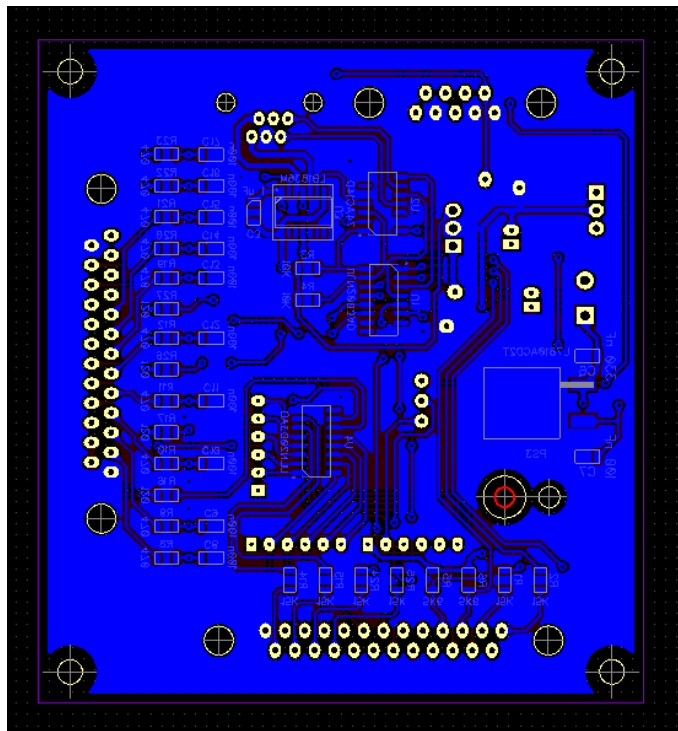
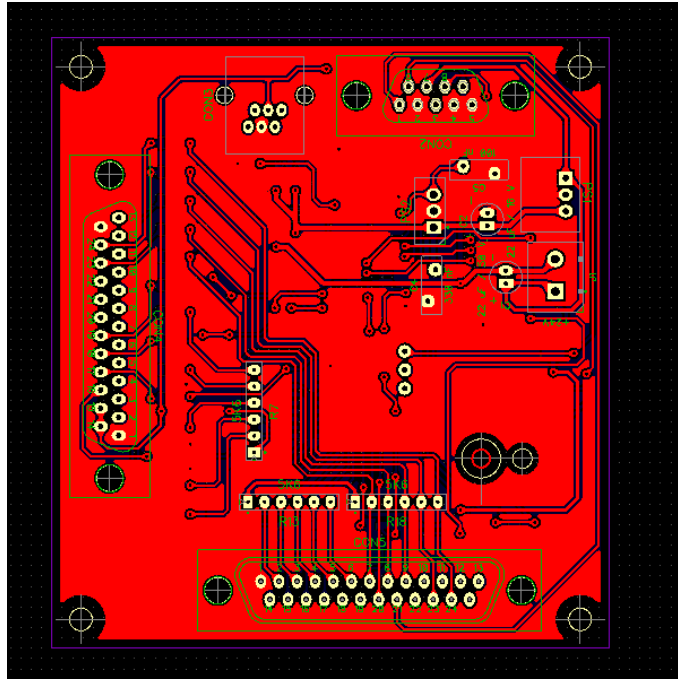


Alimentacion de senales (+5V)



(Opcional) Alimentacion encoder (+10V)





Apéndice E. Diagrama de Gantt.

Fecha de inicio	Duracion	Fecha de final	Tarea
25/02/2022	327	18/01/2023	desarrollo placa electrónica
25/02/2022	7	04/03/2022	reunir información de la maquetas
25/02/2022	7	04/03/2022	reunir información del autómata
04/03/2022	245	04/11/2022	desarrollo esquema del circuito
11/03/2022	59	09/05/2022	elección de componentes
13/10/2022	22	04/11/2022	comprobación del circuito
04/11/2022	11	15/11/2022	desarrollo PCB
15/11/2022	57	11/01/2023	fabricación y montaje PCB
16/01/2023	2	18/01/2023	comprobación de la placa con las maquetas
25/02/2022	444	15/05/2023	desarrollo TIA Portal
25/02/2022	187	31/08/2022	búsqueda información TIA Portal
01/09/2022	62	02/11/2022	Desarrollo proyecto 1
03/11/2022	193	15/05/2023	Desarrollo proyecto 4
17/04/2023	8	25/04/2023	Desarrollo proyecto 2
17/04/2023	8	25/04/2023	Desarrollo proyecto 3

15/05/2023	108	31/08/2023	Desarrollo memoria trabajo
------------	-----	------------	----------------------------

Diagrama de Gantt del trabajo

