



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

**Puesta en marcha y control de un horno
industrial usando un autómeta
programable y una pantalla HMI.**

Autor:

Matilla Buenaposada, Omar Florentino

Tutor:

Moya de la Torre, Eduardo Julio

Departamento de Ingeniería en

Sistemas y Automática

Valladolid, Febrero de 2022.

Agradecimientos

A mi tutor, Eduardo Julio Moya de la Torre, por ayudarme a crecer.

A la empresa Mahle Electronics, por darme la oportunidad y la confianza de realizar este proyecto.

A mis profesores y compañeros que me han acompañado y ayudado durante toda la carrera.

A mi familia, pareja y amigos por estar siempre presentes tanto en los buenos como en los malos momentos.

Resumen

El presente trabajo de fin de grado consiste principalmente en la puesta en marcha de un horno industrial para placas electrónicas, como biosensores y aforadores.

Dicho trabajo surge de las prácticas propuestas por la empresa MAHLE Electronics S.L.

El trabajo parte de un horno industrial inutilizable, ya que previamente trabajaba con un autómata obsoleto en el mercado, de modo que el horno se actualizó a nivel de hardware. Los cambios más significativos fueron, el cambio del autómata que ahora se trata de un PLC S7-1200 de la marca Siemens, y también para el movimiento de la cadena que se hace uso de un variador de frecuencia de la serie G120C, también de la marca Siemens.

Finalmente se crea una interfaz hombre-máquina con la que el operario pueda controlar el horno a la perfección a la vez que supervisar sus distintos parámetros.

Palabras clave

Control lógico programable, interfaz hombre-máquina, regulador PID, automatización, horno.

Abstract

The present final degree Project consists mainly of setting up an industrial oven for electronic boards, such as biosensor and gauges.

This work arises from the practices proposed by the company MAHLE Electronics S.L.

The work starts from an unusable industrial oven, since previously it worked with an obsolete automaton on the market, so the oven was updated at the hardware level. The most significant changes were, the change of the automaton that now is a PLC S7-1200 of the Siemens brand, and also for the movement of the chain that uses a frequency inverter of the G120C series, also of the Siemens brand.

Finally, a man-machine interface is created with which the operator can control the oven to perfection while supervising various parameters.

Keywords

Program logic control, human-machine interface, PID regulator, automation, oven.

Índice

1. Introducción y objetivos.....	15
1.1 Introducción	15
1.2. Objetivos	16
1.3. Estructura de la memoria	17
2. Propuesta de funcionamiento del horno.....	19
3. Descripción de los equipos y elementos.....	21
3.1. Equipos de control.....	21
3.1.1 Autómata programable	21
3.1.2. Módulo de entradas analógicas.....	22
3.1.3 Módulo de salidas digitales.....	23
3.2 Equipos de comunicación	23
3.2.1 Switch.....	24
3.3. Variador de frecuencia	24
3.4. Horno.....	25
3.5. Equipos de alimentación	26
3.6 Otros elementos	27
4. Desarrollo de la programación	31
4.1 Agregar dispositivos	31
4.1.1. PLC	31
4.1.2. Módulos	32
4.1.3. CSM 1277	34
4.1.4. Convertidor G120C	34
4.1.5 Estación PC.....	35
4.2 Configuración vistas	35
4.3 Puesta en marcha G120C	37
4.4 Programación PLC	38
4.4.1 Introducción de los distintos bloques.....	38
4.4.2 Lenguaje de programación.....	40
4.5. Desarrollo del código.....	42
4.5.1. Puesta en marcha	43

4.5.2 Control temperatura V1	46
4.5.2.1 Configuración bloque regulador PID.....	47
4.5.2.2 Puesta en marcha PID_Compact.....	50
4.5.3 Control temperatura V2	51
4.5.4. Estabilización.....	52
4.5.5. Alarmas	53
4.5.6. Gestión de alarmas.....	57
4.5.7. Programación automática	58
4.5.8. Warnings.....	63
4.5.9. Motor.....	65
4.5.10. Apagado horno y señalización	67
4.5.11. Mensajes HMI	73
5. Desarrollo de la interfaz hombre-máquina	75
5.1 Descripción interfaz hombre-máquina.....	75
5.2. Creación interfaz hombre-máquina.....	76
5.3. Configuración interfaz hombre-máquina	79
5.3.1. Configuración botones, interruptores y recuadros E/S.....	79
5.3.2. Configuración elementos gráficos, figuras y cuadros de texto	82
5.3.3. Creación de receta	85
5.3.4 Visor de avisos	87
6. Manual de usuario.....	89
6.1. Introducción	89
6.2. Encendido del horno	89
6.3. Estados del horno.....	95
6.4. Posibles errores y soluciones	97
7. Conclusiones.....	99
8. Bibliografía	101
Anexos.....	103
Anexo 1: Bloques de programa	103
Anexo 2: Variables PLC y HMI.....	165

Índice de figuras

Ilustración 1. PLC S7-1200 con CPU 1214C DC/DC/DC	22
Ilustración 2. Módulo de entradas analógicas SM 1231 TC.....	22
Ilustración 3. Módulo de salidas digitales SM 1222 DC	23
Ilustración 4. CSM 1277 SIMATIC NET	24
Ilustración 5. Convertidor G120C	24
Ilustración 6. Termopar tipo K.....	25
Ilustración 7. Horno industrial THERMATECH	25
Ilustración 8. Fuente de alimentación PM 1207	26
Ilustración 9. Fuente de alimentación LOGO! Power.....	26
Ilustración 10. Interruptor automático magnetotérmico.....	27
Ilustración 11. Transformador.	27
Ilustración 12. Relés.	28
Ilustración 13. Fusibles	28
Ilustración 14. Contactor principal.....	29
Ilustración 15. Borneras de conexión.....	29
Ilustración 16. Agregar PLC 1214C	32
Ilustración 17. Agregar módulo entradas analógicas.....	33
Ilustración 18. Agregar módulo de salidas digitales.....	33
Ilustración 19. Agregar CSM 1277	34
Ilustración 20. Agregar G120C.....	34
Ilustración 21. Módulo de comunicación para la estación PC	35
Ilustración 22. Vista topológica del proyecto	36
Ilustración 23. Vista de redes del proyecto.....	36
Ilustración 24. Puesta en servicio G120C.....	37
Ilustración 25. Funcionamiento OB1.....	39
Ilustración 26. Bloque de función.....	39
Ilustración 27. Función	39
Ilustración 28. Bloque de datos	40
Ilustración 29. Ejemplo lenguaje FUP.....	40
Ilustración 30. Ejemplo lenguaje SCL.....	41
Ilustración 31. Código AWL.....	41
Ilustración 32. Ejemplo lenguaje KOP	42
Ilustración 33. Bloques de programa	43
Ilustración 34. Activación contactor principal.....	44
Ilustración 35. Ejemplo realimentación bobina	44
Ilustración 36. Reset contactor principal	44
Ilustración 37. Segmento 2 de Puesta en Marcha	45
Ilustración 38. Segmento 3 de Puesta en Marcha	45
Ilustración 39. Segmento 4 de Puesta en Marcha	46
Ilustración 40. Activación del regulador PID	46

Ilustración 41. Activación resistencia de calentamiento zona 1	47
Ilustración 42. Bloqueo PIDs opuestos	47
Ilustración 43. Bloque PID_Compact.....	48
Ilustración 44. Configurar PID_Compact.....	49
Ilustración 45. Apartados configuración bloque PID_Compact.....	49
Ilustración 46. Online, medición y modo de ajuste PID.	50
Ilustración 47. Optimización fina PID.	50
Ilustración 48. Carga de parámetros PID.....	51
Ilustración 49. Estabilización zona 1.....	52
Ilustración 50. Horno estable e inicio de setup. OB Alarmas	53
Ilustración 51. Alarmas zona 1	54
Ilustración 52. Trozo de código de alarma rotura de resistencia	55
Ilustración 53. Envío alarma error crítico	56
Ilustración 54. Envío alarma velocidad de cadena para visor de avisos interfaz HMI.....	56
Ilustración 55. Gestión de alarmas.....	57
Ilustración 56. Definición de errores	58
Ilustración 57. Ubicación bloque RD_LOC_T.....	59
Ilustración 58. Variable tipo DTL.....	59
Ilustración 59. Bloque RD_LOC_T.....	60
Ilustración 60. Desglose Domingo y Lunes	60
Ilustración 61. Activación modo automático.....	61
Ilustración 62. Programación automática Domingo rama de encendido	62
Ilustración 63. Programación automática Domingo rama de apagado	63
Ilustración 64. Definición warning zona 1.....	64
Ilustración 65. Envío warning para visor de avisos de la interfaz HMI.....	64
Ilustración 66. Paquetes opcionales Sinamics.....	65
Ilustración 67. Bloque SinaSpeed para controlar el G120C.....	65
Ilustración 68. Conversión velocidad de la cadena para entrada SinaSpeed.....	67
Ilustración 69. Conversión velocidad de cadena a cm por minuto	67
Ilustración 70. Encendido color verde de la columna de señalización	68
Ilustración 71. Encendido color rojo de la columna de señalización	68
Ilustración 72. Activación marcas de ciclo.....	69
Ilustración 73. Encendido color naranja de la columna de señalización	70
Ilustración 74. Variable para animación color naranja de los cuadros de zona de la interfaz HMI.....	70
Ilustración 75. Variable para animación color verde de los cuadros de zona de la interfaz HMI	71
Ilustración 76. Variable para animación color rojo de los cuadros de zona de la interfaz HMI	71
Ilustración 77. Variable para animación color rojo del cuadro de velocidad de la cadena en la interfaz HMI	71

Ilustración 78. Variable para animación color verde del cuadro de velocidad de la cadena en la interfaz HMI	72
Ilustración 79. Correspondencia de color del cuadro zona con la barra de temperaturas.....	72
Ilustración 80. No hay presencia de aire.....	73
Ilustración 81. No hay presencia de agua en el sensor 1.....	73
Ilustración 82. Programación automática desactivada	73
Ilustración 83. Pantallas HMI de Siemens.	75
Ilustración 84. WinCC RT Advanced – HMI	76
Ilustración 85. Nuevo árbol de proyecto tras agregar dispositivo HMI.	77
Ilustración 86. Agregar y diseñar imagen.....	77
Ilustración 87. Imagen HMI - Pantalla principal.....	78
Ilustración 88. Imagen HMI - Programación automática.	79
Ilustración 89. HMI - Propiedades Botón.	80
Ilustración 90. HMI - Eventos del botón de programación automática.....	80
Ilustración 91. HMI - Funciones para los eventos.	81
Ilustración 92. HMI - Vincular interruptor a variable PLC.....	81
Ilustración 93. HMI - Termómetro con zonas definidas.	82
Ilustración 94. HMI - Variables definidas del elemento barra.	82
Ilustración 95. HMI - Apartado animaciones.....	83
Ilustración 96. HMI - Estado Zona 1	83
Ilustración 97. HMI - Apariencia rectángulo zona 1.	84
Ilustración 98. HMI - Visibilidad rectángulo zona 1	84
Ilustración 99. Receta.....	85
Ilustración 100. Registro Perfil 850°C.....	85
Ilustración 101. Botón Crear/Cargar Programa.....	86
Ilustración 102. Visor de recetas	86
Ilustración 103. Avisos de bit.....	87
Ilustración 104. Manual de usuario. Campana 850 °C	89
Ilustración 105. Manual de usuario. Sensores.....	90
Ilustración 106. Manual de usuario. Botón Crear/Cargar Programa	91
Ilustración 107. Manual de usuario. Ventana de perfiles.....	91
Ilustración 108. Manual de usuario. Activar cadena.....	92
Ilustración 109. Manual de usuario. Botón de calentamiento.	92
Ilustración 110. Manual de usuario. Horno en modo setup	93
Ilustración 111. Manual de usuario. Horno estable.....	94
Ilustración 112. Manual de usuario. Programación automática.....	95
Ilustración 113. Manual de usuario. Acuse de alarmas	96
Ilustración 114. Manual de usuario. Error del PLC.....	97
Ilustración 115. Manual de usuario. Error con carga de parámetros	98

1. Introducción y objetivos

1.1 Introducción

El presente trabajo de fin de grado se trata de la puesta en marcha de un horno industrial, el cual se encuentra localizado en la empresa MAHLE Electronics S.L, del parque tecnológico de Boecillo (Valladolid).

La función de este horno industrial es seguir un perfil de temperaturas concreto en función de la placa electrónica con la que se esté trabajando en ese instante. En el horno principalmente se introducen dos tipos de placas electrónicas: biosensores y aforadores. Se hace con el fin de que la pasta que recubre la placa adquiera unas propiedades eléctricas determinadas, siendo este un paso crucial para que el producto final funcione correctamente.

Los biosensores son dispositivos analíticos conformados por un elemento biológico de reconocimiento asociado a un mecanismo de detección e interpretación de la señal obtenida de la interacción entre el analito y el dispositivo analítico, por ejemplo, existen biosensores con los que medir el nivel de glucosa en sangre.

Los aforadores sirven para medir la capacidad de un recipiente o el caudal de un fluido, por ejemplo, se usan para medir la cantidad de gasolina que hay en el depósito de un coche.

Un buen seguimiento de cualquier perfil de temperaturas es una parte esencial del proceso por lo que dicho horno se actualizó a nivel de hardware. Se cambió principalmente el autómatas con el que trabajaba y se añadió un variador de frecuencia con el que gobernar el motor encargado de mover la cadena. También tuvieron lugar otros cambios: gran parte de la instalación eléctrica se modificó, se hizo uso de otros sensores para medir la temperatura, se añadió una columna de señalización junto con un buffer, y algún otro detalle más.

Debido a los diversos cambios que sufrió el horno, nació todo el proyecto para la puesta en marcha del horno en la que se realiza la programación de un PLC Siemens S7-1200. Esto se lleva a cabo para alcanzar otro nivel de la eficacia y la eficiencia del horno a la hora de seguir los procesos de calentamiento.

Hoy en día existen infinidad de procesos automatizados y controlados mediante autómatas programables como va a ser el caso de nuestro horno. El software más conocido y el que nos ofrece un amplio abanico de posibilidades es el TIA Portal de Siemens. Con este llevaremos a cabo la programación completa del S7-1200 que es uno de los PLCs con mayor relación calidad-

precio e ideal para este tipo de procesos. También existen otros modelos como el S7-300, S7-400 y S7-1500.

Una vez se completa la programación del PLC con la que controlar el horno y gestionar tanto las advertencias como las alarmas a la perfección, se procede a crear una interfaz HMI. Esto es algo muy común en la industria ya que gracias a este tipo de interfaces podemos tener un control total sobre el proceso además de poder visualizar todo tipo de parámetros en tiempo real dentro del mismo.

Para la creación de la interfaz se utiliza nuevamente TIA Portal en conjunto con WinCC Runtime Advanced, un complemento de Siemens que nos ayudará a la creación de la interfaz.

Finalmente se diseña un manual de usuario del horno para que toda persona que no esté familiarizada con él adquiera unos conocimientos básicos acerca de este para saber encender o apagar el horno, realizar una programación automática o crear y modificar los distintos perfiles de temperatura.

1.2. Objetivos

En este apartado se van a dar a conocer los objetivos del trabajo de fin de grado, así como la motivación de cada uno de ellos. Es una parte muy importante de todo el trabajo realizado y que se ha estudiado en conjunto con la empresa, ya que es necesario tener los objetivos bien definidos desde un principio y contar con el apoyo y la supervisión por parte de la empresa.

Para definir estos objetivos se planificaron dos reuniones semanales con los jefes de la empresa, con el fin de ir revisándolos y actualizándolos si fuera necesario.

Al finalizar el trabajo el horno debe trabajar diariamente con normalidad, ya que al comienzo el horno no se encontraba operativo por la actualización hardware que había sufrido. Los objetivos principales fueron los siguientes:

- 1. Funcionamiento del horno:** Análisis en conjunto con la empresa del funcionamiento del horno y de los posibles modos de funcionamiento que pudieran ser implementados.
- 2. Descripción de los elementos y equipos utilizados para realizar el control:** Entender cada elemento y equipo de la actualización hardware, así como su motivación para poder desarrollar con más facilidad la programación.

3. Programación del autómatas: El horno está gobernado por un autómatas programable con el que gestionaremos todas las entradas y salidas del proceso. Como entradas podemos tener válvulas de aire, de agua, sensores de temperatura, etc. Como salidas tendremos el movimiento de la cadena, el calentamiento del horno, entre otras. A lo largo de todo el proyecto se ha ido definiendo dentro de este apartado toda la gestión de las alarmas, de la señalización, la programación automática, las condiciones de encendido y apagado del horno, etc.

4. Creación de una interfaz HMI: Una vez el horno esta correctamente programado, debemos de poder tener una correcta comunicación con este. Esto lo vamos a conseguir creando una interfaz hombre-máquina que permitirá la visualización y el acceso a las distintas variables. Gracias a esta interfaz buscamos conseguir un acceso rápido a las variables tanto para escribirlas como para leerlas.

Una vez completados los objetivos principales podíamos centrarnos en una serie de objetivos más secundarios, pero también muy importantes:

- **Realizar ensayos con diferentes perfiles de temperatura:** Importante comprobar que funciona correctamente la programación y que los PIDs de los que hablaremos más adelante, han quedado correctamente configurados.
- **Detallar el uso del horno:** Para ello realizamos un manual de usuario donde explicamos el funcionamiento del horno, así como la solución a posibles errores que se han ido dando a lo largo del proceso.

1.3. Estructura de la memoria

A continuación, se muestra la estructura que conforma el Trabajo Fin de Grado.

Capítulo 1. Introducción y objetivos.

Dedicado a hacer una breve introducción sobre el trabajo que se va a realizar, así como los objetivos principales y secundarios que se van a establecer acerca del horno, la programación del autómatas y la creación de una pantalla HMI.

Capítulo 2. Propuesta de funcionamiento del horno

Breve explicación sobre cómo puede funcionar el horno y que funcionalidades puede tener.

Capítulo 3. Descripción de los equipos y elementos.

Dedicado a describir brevemente los componentes que conforman el horno industrial, en especial los necesarios para la programación del horno.

Capítulo 4. Desarrollo de la programación.

Dedicado a explicar detalladamente la programación que se ha realizado para el funcionamiento del horno.

Capítulo 5. Desarrollo de la interfaz hombre – máquina.

Dedicado a explicar la creación de nuestra interfaz HMI.

Capítulo 6. Manual de usuario

Dedicado a realizar un manual de usuario para facilitar la labor del operario en donde se recoge como hacer uso del horno, así como los posibles fallos y soluciones.

Capítulo 7. Conclusiones.

En este capítulo quedan reflejadas las conclusiones finales tras la realización del proyecto, así como posibles líneas futuras.

Capítulo 8. Bibliografía.

En este capítulo se recogen todas las fuentes utilizadas para la realización de este proyecto.

ANEXOS**ANEXO 1. Bloques de programa**

Código desarrollado con el que funciona correctamente nuestro horno industrial.

ANEXO 2. Variables PLC y HMI

Variables creadas tanto en el autómata como en el HMI para poder llevar a cabo toda la programación del código.

2. Propuesta de funcionamiento del horno

Para poder dar comienzo al proyecto, y para más adelante poder desarrollar la programación, primero debíamos analizar qué puede hacer el horno y cómo conseguirlo.

1. Control del perfil de temperaturas:

El horno dispone de una cadena de 40 centímetros de ancho en la cual se colocarán las placas electrónicas en filas de tres en tres. Se colocarán en la boca del horno, conocida como zona de carga. Una vez ahí, con la cadena en movimiento entrarán al horno y comenzarán su proceso de calentamiento, siguiendo un perfil determinado de temperaturas que nosotros tendremos que dejar previamente establecido.

El calentamiento se produce en el interior del horno, en el cual se encuentran las resistencias de calentamiento y una mufla que mantendrá el calor y protegerá a los distintos elementos del horno.

Las resistencias de calentamiento las controlaremos mediante un control PID que realizaremos con el autómat. Para ello uno de los elementos implicados será un módulo de entrada de señales analógicas, donde recogeremos los datos de temperatura para poder realizar este control.

Una vez realizado el control de temperatura, habrá que gestionar la velocidad de la cadena. Esto se realizará gracias a un variador G120C en conjunto con nuestro autómat.

Combinando la temperatura y la velocidad conseguiremos el perfil de temperatura deseado. Las placas completarán su proceso con éxito y llegarán a la boca de salida.

2. Seguridad:

El horno debe de contar con un sistema de seguridad que pueda proporcionar la seguridad tanto de los operarios como de los elementos del equipo y del proceso.

En caso de superar determinadas temperaturas debemos gestionar una baliza de señalización e incluso el apagado del horno. También no podremos dar comienzo al proceso de calentamiento en caso de no tener unos requisitos previos de seguridad como disponer de agua y aire para la refrigeración del horno.

El control de seguridad se realizará con el autómat, gestionando las señales entrantes y con la capacidad de actuar sobre contactores para cortar la tensión o sobre electroválvulas para abrir o cerrar el paso del agua.

3. Modos de funcionamiento:

Plantearémos un modo de funcionamiento en el que el encendido del horno sea manual, donde tendremos que realizar unas comprobaciones para comenzar el proceso. Otro modo de funcionamiento será en modo automático donde el horno podrá comenzar dicho proceso o terminarlo de forma automática en cualquier día y hora de la semana.

En paralelo tendremos también un modo de funcionamiento a máxima potencia de calentamiento y otro en donde la potencia será ligeramente inferior con el fin de reducir costes de energía.

3. Descripción de los equipos y elementos

En este apartado vamos a hacer una breve descripción de los distintos equipos y elementos que conforman el horno industrial con el que vamos a trabajar. Se dividirán en distintos grupos para tener un poco más claro sus utilidades. Tendremos equipos de control, de comunicación, de alimentación, y “otros”, donde se incluirán los elementos básicos de la instalación.

3.1. Equipos de control

Es el grupo de elementos que se va a encargar de leer, gestionar y controlar todas las entradas y salidas del sistema. Como entradas tendremos sensores de temperatura, válvulas de refrigeración, etc. Como salidas habrá resistencias, accionamientos, relés y muchos otros.

3.1.1 Autómata programable

El autómata programable o PLC es el que realice todo el proceso, control de temperaturas, gestión de alarmas, velocidad de la cadena, comunicación con la interfaz HMI, etc.

Nuestro PLC se trata de un S7-1200 de la marca Siemens. Este nos ofrece la capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización. Pertenece a la gama de controladores lógicos programables más estandarizada de Siemens, que recibe el nombre de SIMATIC S7, además, esta también incluye los modelos S7-300, S7-400 y S7-1500.

La CPU es una 1214C DC/DC/DC, con referencia 6ES7214-1AG31-0XB0. Esta dispone de 14 entradas digitales, 10 salidas digitales y 2 entradas analógicas. Tiene una tensión de alimentación que varía entre 20,4 y 28,8V DC. A su vez, tiene disponible un puerto PROFINET para la programación y comunicación. Gracias a 100 kB de memoria para guardado de datos y programa, esta CPU es óptima en todas las industrias en múltiples tareas de control.



Ilustración 1. PLC S7-1200 con CPU 1214C DC/DC/DC

3.1.2. Módulo de entradas analógicas

Se trata del módulo de señales de termopar SM 1231 de la marca Siemens. Es necesario para poder hacer la medición de temperatura de cada zona del horno. El módulo mide el valor de la tensión conectada a sus entradas. Este valor puede ser tanto la temperatura de un termopar como voltios.

- Si se trata de voltios, el valor máximo del rango nominal serán 27648 decimales.
- Si se trata de temperatura, el valor se expresará en grados multiplicados por diez (p. ej. 25,3 grados se expresarán como 253 decimales). Es por ello que, a lo largo del código, la temperatura que recibimos la dividimos entre diez.



Ilustración 2. Módulo de entradas analógicas SM 1231 TC

3.1.3 Módulo de salidas digitales

Se trata del módulo de salidas digitales SM 1222 DC de la marca Siemens con el que podemos añadir 16 salidas digitales. Tiene salidas con transistor con una corriente de salida de 0,5A a 24V DC.

Se usa por la necesidad de tratar más entradas digitales como las correspondientes al semáforo de señalización que utilizaremos, o como las pertenecientes a las válvulas aire y agua.



Ilustración 3. Módulo de salidas digitales SM 1222 DC

3.2 Equipos de comunicación

Son los encargados de comunicar, coordinar y dirigir todas las operaciones. Se rigen mediante unos protocolos de comunicación de redes. En la industria actual se pueden ver protocolos como 'PROFINET', 'Industrial Ethernet', 'PROFIBUS' y 'AS-i' entre otros.

'PROFINET' que está basado en el estándar ETHERNET, es uno de los más utilizados dentro de la automatización, por lo que también ha sido el protocolo utilizado durante este proyecto.

3.2.1 Switch

Este módulo de comunicación, en concreto el CSM 1277 de Siemens, cuenta con 4 puertos no gestionables con una velocidad de transmisión de 10/100 Mbit/s. El switch se alimenta con 24V DC, y es perfecto para la conexión ethernet del S7-1200 con otros 3 componentes.



Ilustración 4. CSM 1277 SIMATIC NET

3.3. Variador de frecuencia

El variador de frecuencia con el que se trabaja es un SINAMICS G120C, miembro de la familia de accionamientos SINAMICS. Es la primera elección para soluciones de automatización innovadoras y con proyección de futuro. Tiene un manejo y una funcionalidad uniformes gracias a una plataforma común de hardware y software, además de un alto grado de flexibilidad y capacidad de combinación. Se suele utilizar tanto para baja como para media tensión.



Ilustración 5. Convertidor G120C

3.4. Horno

Trabajamos con un horno industrial de la marca THERMATECH. Este dispone de 8 zonas de calentamiento, controladas por unas resistencias. El valor de la temperatura se recoge con 8 termopares tipo k como el que vemos en la Ilustración [6]. Mediante el control de las resistencias y el seguimiento de los termopares podremos seguir una u otra curva de temperatura según el perfil requerido por una u otra placa electrónica.



Ilustración 6. Termopar tipo K

El aspecto del horno industrial con el que se ha trabajado es el siguiente:



Ilustración 7. Horno industrial THERMATECH

3.5. Equipos de alimentación

Son los encargados de alimentar con los valores adecuados a los distintos componentes del horno (p.ej. la fuente de alimentación de la Ilustración [8] sería la encargada de alimentar el PLC y el Switch de comunicación entre otros).



Ilustración 8. Fuente de alimentación PM 1207

También tenemos otras fuentes como la LOGO Power con referencia 6EP1 332-1SH41, que es una fuente de alimentación regulada con entrada 100...240V AC y salida 24V DC/2,5 A.



Ilustración 9. Fuente de alimentación LOGO! Power.

3.6 Otros elementos

En la instalación también encontramos otro tipo de elementos que son muy comunes y básicos dentro de las instalaciones eléctricas, pero no por ello menos importantes.

- **Magnetotérmicos**

Son unos dispositivos capaces de interrumpir la corriente eléctrica de un circuito cuando esta sobrepasa ciertos valores máximos. Su función es proteger el resto de la instalación frente a sobrecargas y cortocircuitos.



Ilustración 10. Interruptor automático magnetotérmico.

- **Transformadores**

Son elementos que permiten aumentar o disminuir la tensión de un circuito eléctrico de corriente alterna, manteniendo la potencia.



Ilustración 11. Transformador.

- **Relés**

Son dispositivos electromagnéticos que funcionan como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.



Ilustración 12. Relés.

- **Fusibles**

Son dispositivos constituidos por un soporte adecuado y un filamento o lámina de metal o aleación de bajo punto de fusión que se intercala en un punto determinado de una instalación eléctrica para que se funda cuando la intensidad de corriente supere un determinado valor que pudiera hacer peligrar la integridad del resto de elementos de la instalación.



Ilustración 13. Fusibles

- **Contactores**

Son aparatos eléctricos de mando a distancia que pueden habilitar o cortar el flujo de corriente de un circuito eléctrico.



Ilustración 14. Contactor principal.

- **Borneras de conexión**

Son elementos de conexión cuya función principal es garantizar la seguridad y asegurar las conexiones del conductor eléctrico y mecánico.

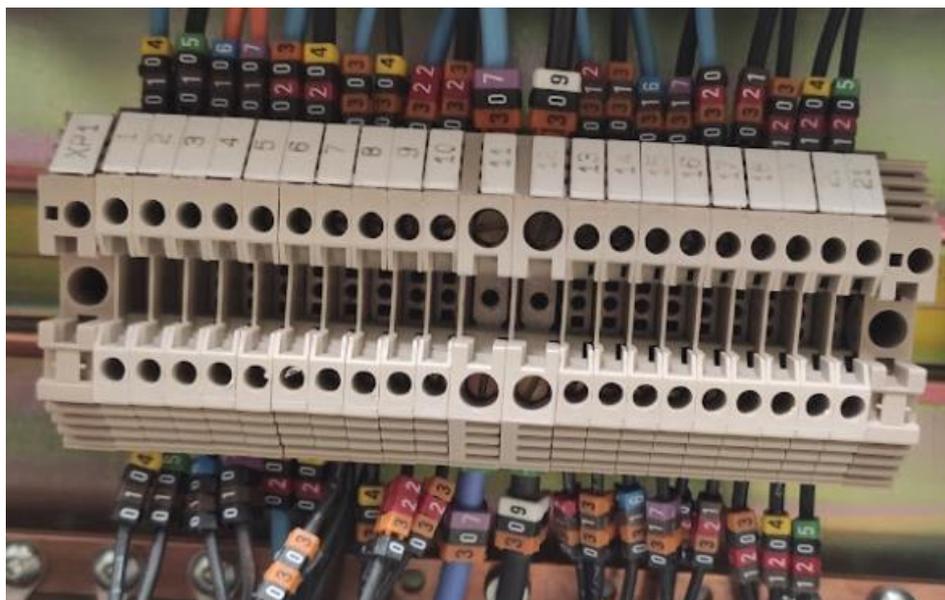


Ilustración 15. Borneras de conexión.

4. Desarrollo de la programación

Para la programación del autómata, de la pantalla HMI y del variador de frecuencia, se va a utilizar el software de Siemens TIA PORTAL, con los siguientes complementos:

- **Totally Integrated Automation Portal.** Versión V16.
- **WinCC Runtime Start.** Versión V16.

Las principales ventajas que nos ofrece este software es que nos permite configurar o programar PLCs, desarrollar interfaces HMI o sistemas de supervisión de adquisición de datos.

Dividiremos la explicación acerca de la programación según se ha ido realizando.

4.1 Agregar dispositivos

Para poder dar paso al desarrollo de la programación, previamente debemos agregar todos los dispositivos implicados en el proceso.

4.1.1. PLC

Para agregar nuestro autómata programable debemos:

1. Ir a agregar un dispositivo.
2. Ir a controladores.
3. Ir a SIMATIC S7-1200.
4. Seleccionar la CPU correspondiente y la versión de firmware.

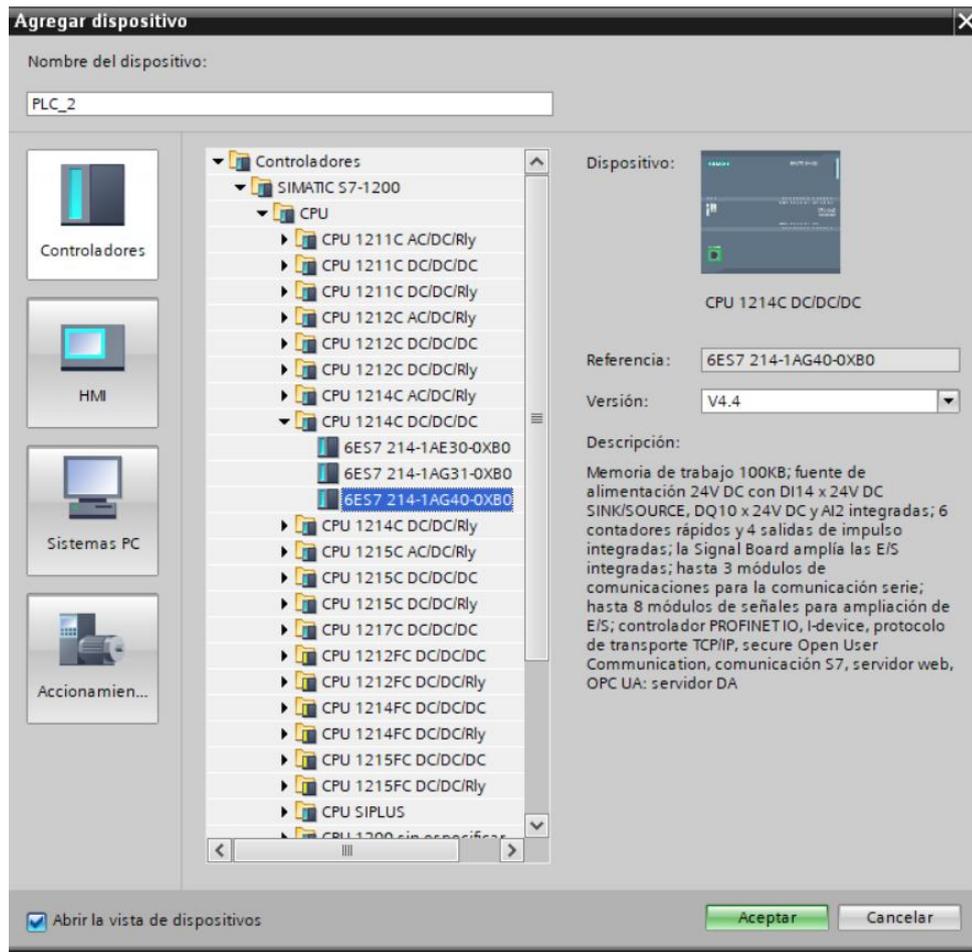


Ilustración 16. Agregar PLC 1214C

Otra forma con la que agregar el dispositivo a nuestro proyecto, sería conectarnos on-line y ver los dispositivos accesibles que hay en ese momento. Mediante la IP seleccionaríamos nuestro PLC y lo agregaríamos a nuestro árbol de proyecto.

4.1.2. Módulos

Para agregar los módulos utilizados en el proyecto debemos de encontrarnos en la vista de dispositivos e ir al catálogo hardware. Finalmente seleccionamos el módulo y arrastramos al siguiente rack disponible.

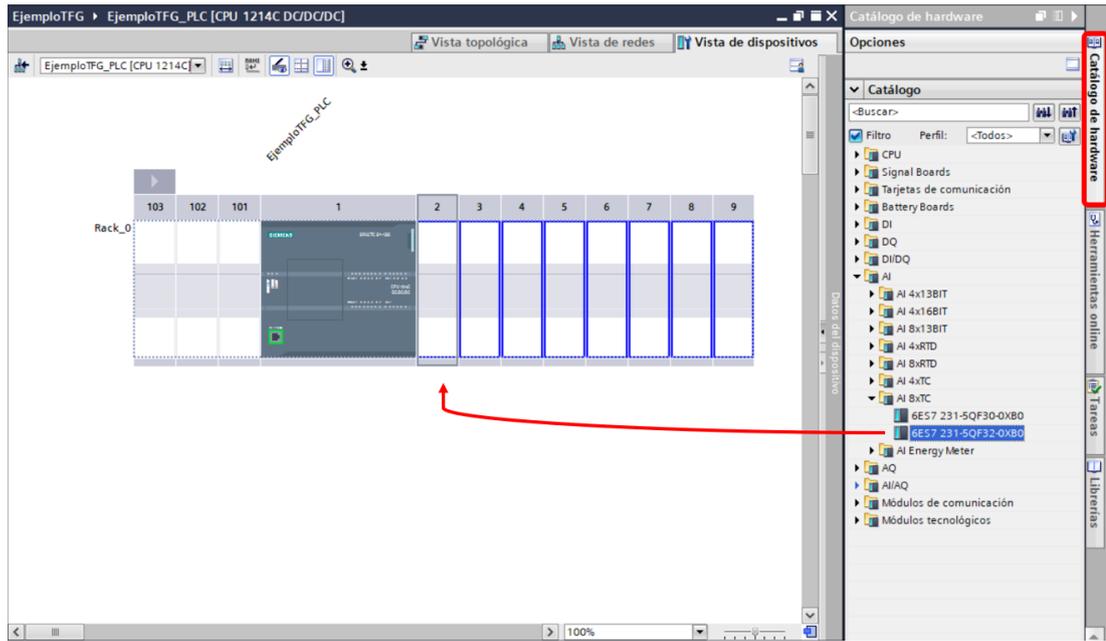


Ilustración 17. Agregar módulo entradas analógicas

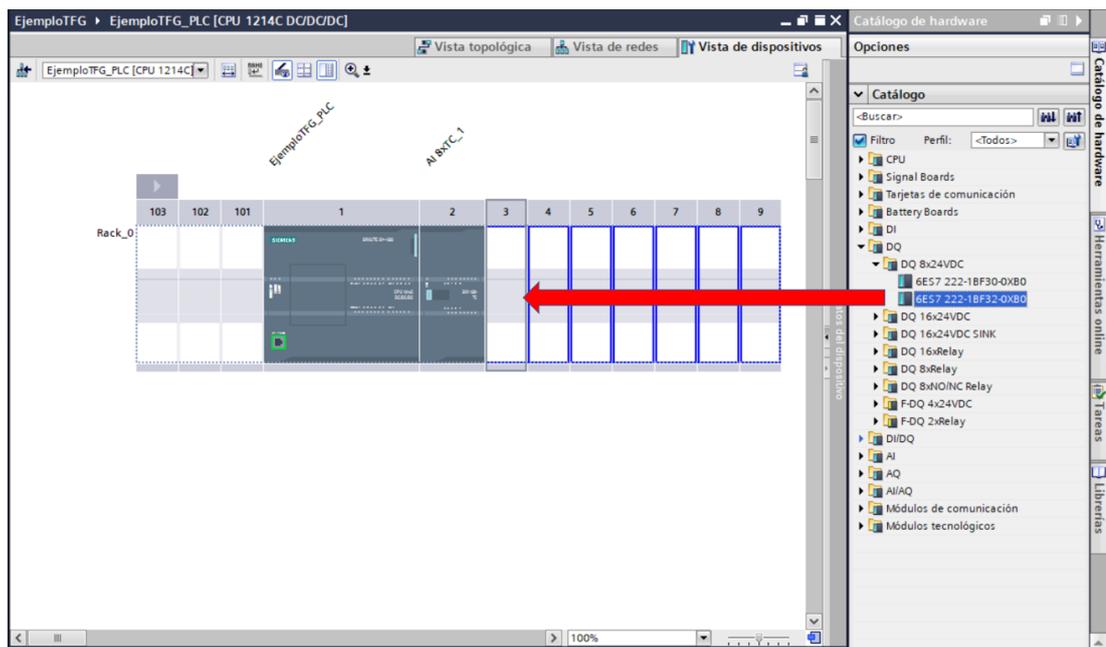


Ilustración 18. Agregar módulo de salidas digitales

4.1.3. CSM 1277

Para agregar este dispositivo, tratándose de un switch debemos situarnos en la vista topológica del proyecto, y nuevamente accediendo al catálogo hardware buscamos en el apartado de componentes de red. Finalmente arrastramos el objeto a la vista topológica.

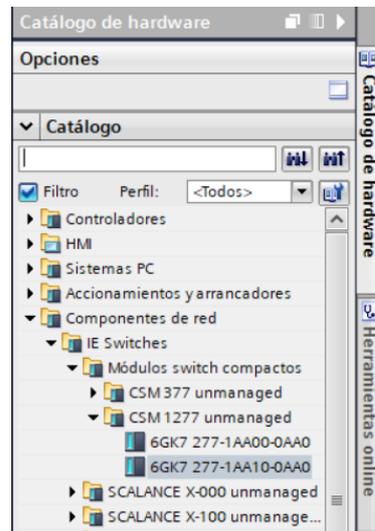


Ilustración 19. Agregar CSM 1277

4.1.4. Convertidor G120C

Para agregar nuestro convertidor, seguimos el mismo procedimiento que para los módulos o el switch. Buscaremos nuestro modelo de convertidor en el catálogo hardware.

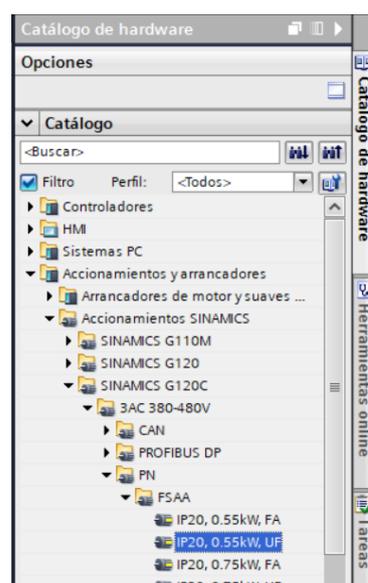


Ilustración 20. Agregar G120C

4.1.5 Estación PC

Agregamos este dispositivo por lo que ya hemos comentado anteriormente, no disponemos de una pantalla física HMI de Siemens. Se detalla el procedimiento en el apartado 4.2.

Para poder conectar esta estación con el resto dispositivos debemos añadirle un módulo de comunicación.

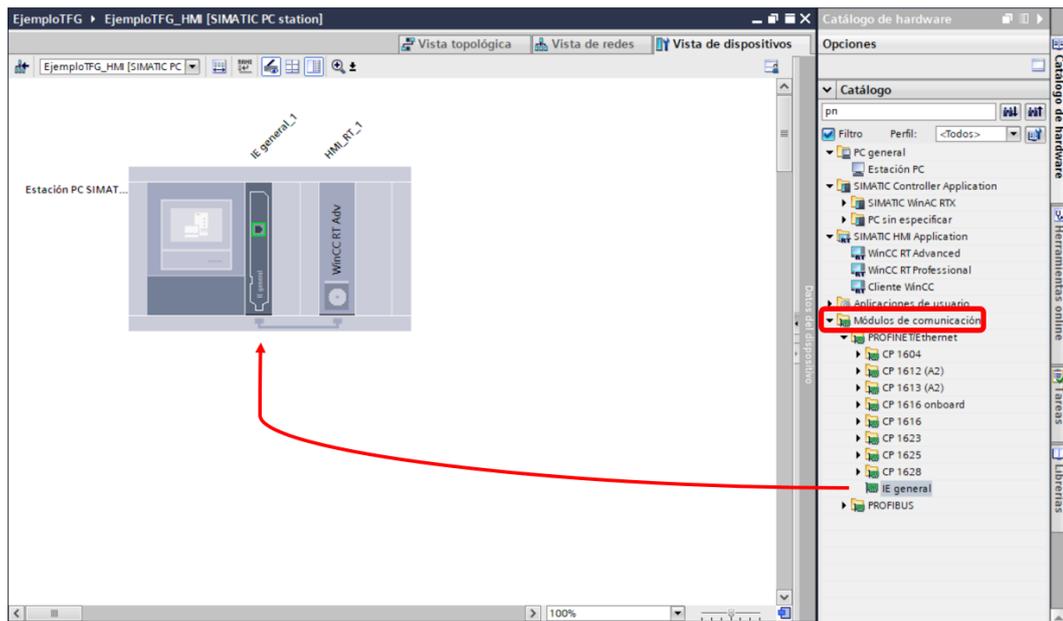


Ilustración 21. Módulo de comunicación para la estación PC

4.2 Configuración de las vistas

Una vez tenemos todos los dispositivos agregados, debemos de indicar al TIA Portal las conexiones que habrá entre estos dispositivos. Para ello usamos las vistas.

- **Vista topológica:** Es una de las tres áreas de trabajo del editor de hardware y redes. Se pueden realizar las siguientes tareas:
 1. Mostrar la topología de Ethernet.
 2. Configurar la topología de Ethernet.
 3. Determinar y minimizar diferencias entre la topología prevista y la topología real.

A continuación, se muestra la vista topológica del proyecto:

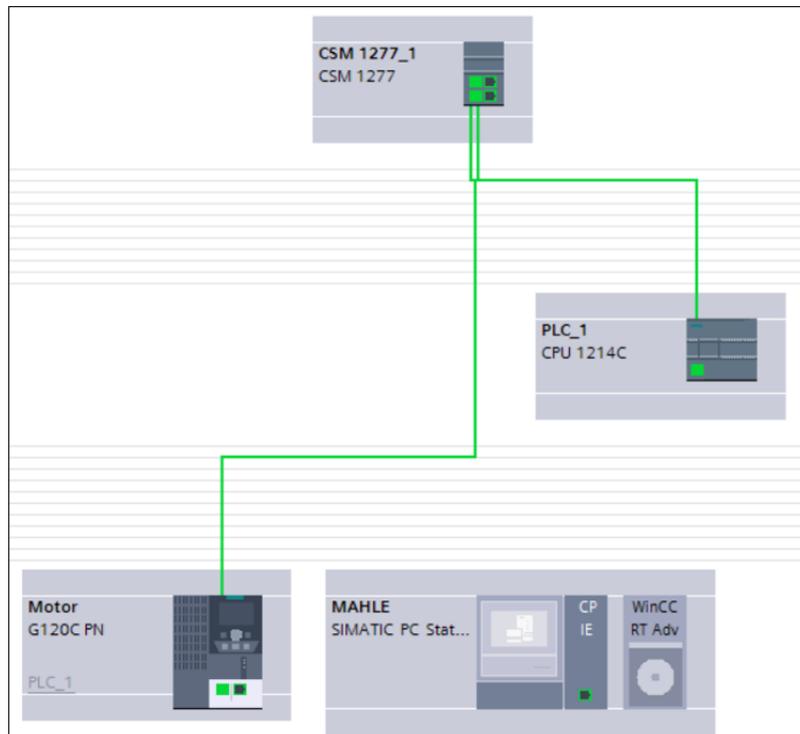


Ilustración 22. Vista topológica del proyecto

- **Vista de redes:** Nos permite ver todas las conexiones Profinet a la vez. Podemos apreciar nuestra configuración de la vista de redes del proyecto en la siguiente Ilustración.

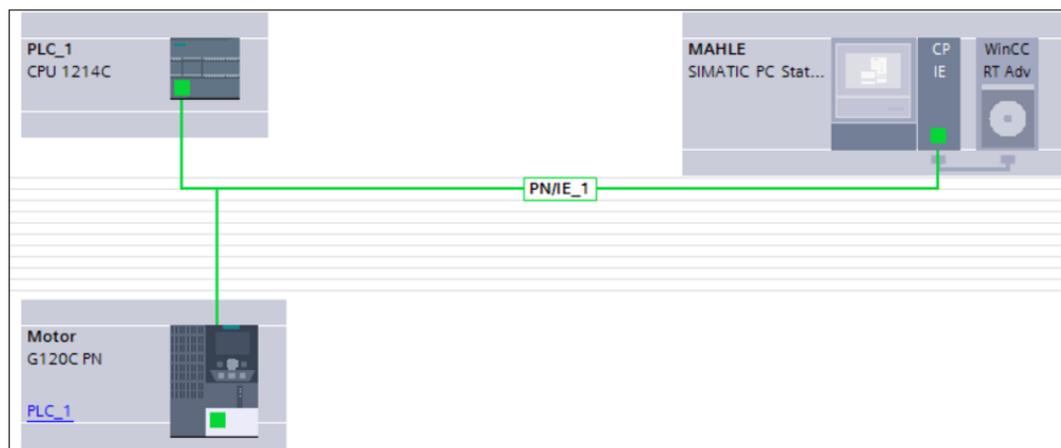


Ilustración 23. Vista de redes del proyecto

Es importante realizar correctamente tanto la vista topológica como la vista de redes, ya que si hay errores no conseguiremos comunicar correctamente los dispositivos entre sí. Debemos de conocer como están cableados físicamente los dispositivos.

4.3 Puesta en marcha G120C

El convertidor G120C es un equipo con la capacidad de controlar un motor asíncrono trifásico. Está diseñado para el montaje en instalaciones eléctricas o máquinas. Además, está homologado para la utilización en redes industriales del ámbito industrial y terciario. El uso en redes públicas requiere de medidas suplementarias.

Se podría comenzar con la programación del autómatas una vez están agregados todos los dispositivos, pero como medida de comprobación se realiza una puesta en marcha del G120C. Con ello vamos a comprobar que el motor esta correctamente comunicado y sigue las instrucciones que le indiquemos a la perfección.

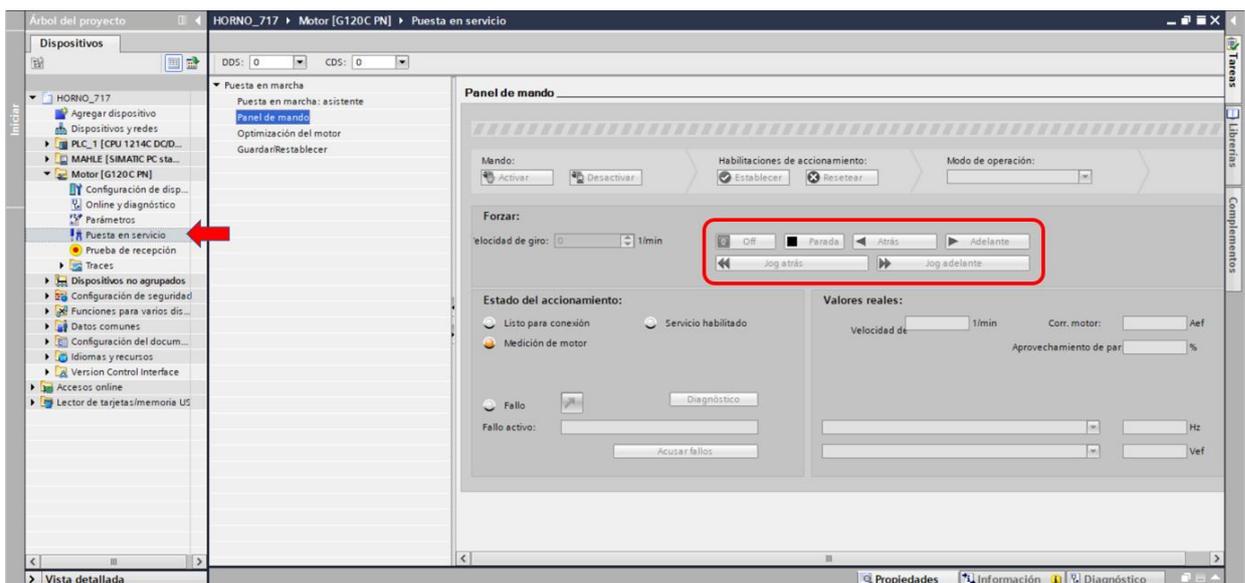


Ilustración 24. Puesta en servicio G120C

Como vemos en la Ilustración [24], tenemos una opción de “puesta en servicio”. Con ella podemos realizar movimientos del motor en sentido horario o sentido antihorario con un clic en los botones “atrás” y “adelante” respectivamente. También se pueden realizar estos movimientos mientras el botón este pulsado con “Jog atrás” y “Jog adelante”.

Una vez comprobamos que el motor se mueve correctamente, es interesante fijarse en la velocidad de giro del motor. Esta función resultó muy

útil ya que el operario, a través de la interfaz HMI, debe ser capaz de encender, apagar y controlar la velocidad de la cadena. Dependiendo de la velocidad de la cadena, el perfil de temperatura será uno u otro, por ello es un parámetro muy importante en este proceso.

Finalmente, con la puesta en marcha del G120C, podemos dar paso a la programación del autómatas.

4.4 Programación PLC

En este apartado se va a hacer una pequeña introducción a la programación de PLCs con TIA Portal, además de detallar el código implementado para este horno industrial.

Ya que no es un proyecto muy extenso, ni va a ser modificado por otro ingeniero, se ha seguido una programación completa mediante bloques de organización "OB" y mediante bloques de datos "DB".

En las grandes empresas donde tengan su planta automatizada con PLCs, todo el código tiene que seguir unas pautas que hagan que este sea un código limpio o *clean code*. Esto se debe a que la planta con el paso del tiempo realizará mejoras o ampliaciones, y puede que el ingeniero que tenga que programar todo ello no sea el mismo, por lo que tener un código bien organizado y estructurado, con comentarios, y en general, muy limpio, lo agradecerá la empresa y el ingeniero en especial.

Para tener un código bien organizado y estructurado se hacen uso de todos los bloques disponibles en TIA Portal. A pesar de solo haber utilizado dos de ellos, se va a hacer una pequeña explicación también sobre los dos restantes.

4.4.1 Introducción de los distintos bloques

El software de TIA Portal nos ofrece la posibilidad de usar cuatro tipos de bloques para la programación del autómatas. Estos son los siguientes:

- **OB (bloque de organización):** Permite estructurar el programa del autómatas sirviendo de interfaz entre el sistema operativo y el programa de usuario. El OB de ciclo contiene el programa principal pudiendo llegar a tener más de un OB de ciclo en el programa de usuario.

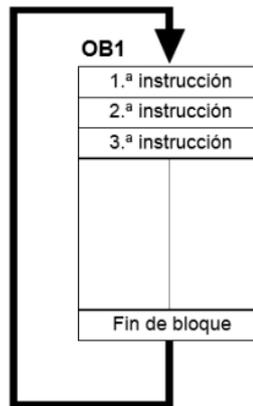


Ilustración 25. Funcionamiento OB1

- **FB (bloque de función):** Necesita un área de memoria asignada para cada llamada (instancia). Al llamar a un FB se le puede asignar un bloque de datos (DB) como bloque de datos instancia. A los datos de este DB de instancia se accede a través de las variables del FB. Si se llama varias veces a un FB, se le deben asignar distintas áreas de memoria. En un FB también pueden ser llamados otros FB y FC.



Ilustración 26. Bloque de función

- **FC (función):** Similar a un bloque de función pero una FC no tiene ningún área de memoria asignada. Los datos locales de una función se pierden tras ejecutar la función.



Ilustración 27. Función

- **DB (bloque de datos):** Los DB se utilizan para proporcionar espacio de memoria para las variables de datos. Existen dos tipos de bloques de

datos. DB globales, en los que todos los OB, FB y FC pueden leer datos almacenados o incluso escribir datos en los DB; y DB de instancia, que están asignados a un FB determinado.



Ilustración 28. Bloque de datos

4.4.2 Lenguaje de programación

TIA Portal dispone de cuatro lenguajes de programación con los que trabajar. Estos son FUP, SCL, AWL y KOP. Se puede trabajar con uno de ellos solo, o bien, ir combinándolos según el segmento que haya que programar.

- **FUP (lenguaje mediante funciones).**

Es un lenguaje gráfico que permite programar elementos que aparecen como bloques para ser cableados entre sí de forma análoga al esquema de un circuito, muy cómodo para técnicos habituados a trabajar con circuitos de puertas lógicas.

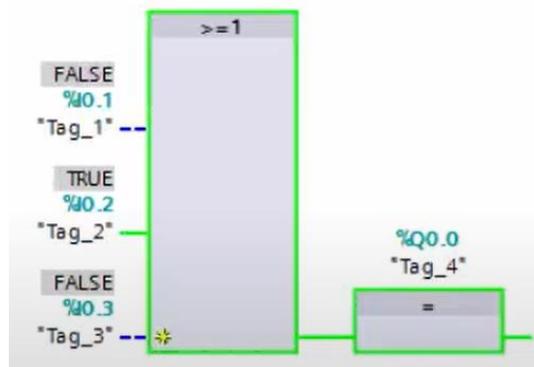


Ilustración 29. Ejemplo lenguaje FUP

- **SCL (lenguaje de control estructurado).**

Es un lenguaje de programación de alto nivel orientado a Pascal que posibilita una programación estructurada.

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)
-------	------------------	--------------------	-------------------	---------

```

1
2 #i:=2;
3 #j:=5;
4 FOR #i:=2 TO 8 DO
5   %EW(#i) := 0;
6   %AB(#i) := %MB(#i);
7   %AW(#i+1) := %MW(#i+1);
8 END_FOR;
9 #Var_Bool := %MX(IDX:=#i, BIT:=#j);
10 %EX(IDX:=#i, BIT:=#j) := %MX(IDX:=#i, BIT:=#j);

```

Ilustración 30. Ejemplo lenguaje SCL

- **AWL.**

Es conocido como el lenguaje ensamblador de Siemens, se trata de un conjunto de nemónicos y operandos que componen instrucciones traducibles a código máquina.

Es un lenguaje potente y compacto que permite implementar programas de control complejos. Es el que está más cerca de la máquina y por tanto no está pensado para expertos en instalaciones y dispositivos de control, sino para personas con formación en informática y electrónica digital.

Las instrucciones básicas de este lenguaje son las que vemos a continuación:

- U: Como una puerta AND.
- UN: Como una puerta NAND (Negación de AND).
- O: Puerta OR.
- ON: Puerta OR negada.
- X: Puerta XOR.
- XN: Puerta XOR negada.
- =: Asignación.

A continuación, vemos un ejemplo de código en AWL:

```

U(
O E32.0
O E32.1
)
U(
ON E32.0
ON E32.1
)
UN E32.2
= A 33.0

```

Ilustración 31. Código AWL

- **KOP (diagrama de contactos).**

Lenguaje de programación de tipo escalera o ladder. Tiene un gran parecido con los esquemas eléctricos. Es el lenguaje que se ha utilizado durante todo el proyecto.

Principalmente se basan en contactos normalmente abiertos, normalmente cerrados y bobinas.



Ilustración 32. Ejemplo lenguaje KOP

Además de estos cuatro lenguajes con los que se puede ir conformando nuestro código, existen librerías de bloques dentro de TIA Portal que nos dan la posibilidad de realizar operaciones binarias, temporizadores, bloques de 'motion control' con los que realizar el control de accionamientos, y una infinidad de bloques más.

4.5. Desarrollo del código

Antes de comenzar con la programación del horno, había que tener clara la idea que nos transmitía la empresa acerca de su funcionamiento. Los requisitos que ha de cumplir el horno son los siguientes:

1. **Seguridad:** El autómatas debe de saber gestionar correctamente cualquier imprevisto que pueda ocurrir en el horno, por lo que el horno tiene que ser totalmente seguro para asegurar el bienestar de la infraestructura y del personal.
2. **Precisión:** Para que el producto final salga bien y tenga la mayor calidad posible, el horno debe ser capaz de seguir a la perfección los perfiles de temperatura.
3. **Sencillez:** Tiene que ser fácil e intuitivo para el operario. Este apartado se trabaja en conjunto con la interfaz HMI.
4. **Eficiencia:** El horno tiene que ser capaz de trabajar lo más eficientemente posible. Esto quiere decir que siga correctamente los perfiles de temperatura y no desperdiciemos energía.

Se va a ver en detalle todos los bloques de organización programados y su motivación. Cada bloque tendrá su DB de instancia correspondiente donde estarán muchas de las variables. La estructura que tiene el proyecto es la que vemos en la Ilustración [33].



Ilustración 33. Bloques de programa

4.5.1. Puesta en marcha

Este OB está compuesto por cuatro segmentos con los que tratamos de gestionar el encendido y apagado del horno. En el primer segmento se gestiona el encendido del horno. Este dispone de un contactor principal que al ser activado alimentará al horno de la red eléctrica. Este contactor principal está conectado a la salida “%Q1.0” del autómeta.

Las condiciones para activar el contactor deben cumplirse todas a la vez como si fuera una puerta AND, y son las siguientes:

1. La variable “2. Control de temperatura. PM1” debe estar activa. Esta variable estará vinculada al botón de encendido del horno en la interfaz HMI.
2. La entrada “%I0.0” este activa. Corresponde con otro contactor que está conectado a la entrada del autómeta.
3. La salida “%Q12.3” este activa. Es necesaria porque antes de encender el horno debemos de tener activada la salida de aire.

- La variable “1. Puesta en marcha. PresenciaAguaOK” que nos indicará que por el sistema de refrigeración del horno está circulando agua.

En cualquier instante que se cumplan estas condiciones, el contactor principal se activará con una bobina de “SET”. Esto quiere decir que estará activada permanentemente, aunque las condiciones anteriores se dejen de cumplir.

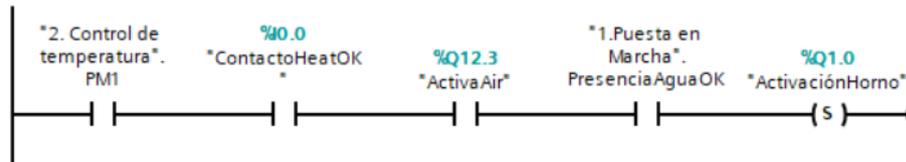


Ilustración 34. Activación contactor principal

Para entender mejor la bobina “SET” vemos en la siguiente Ilustración que es lo mismo que una realimentación de dicha variable.

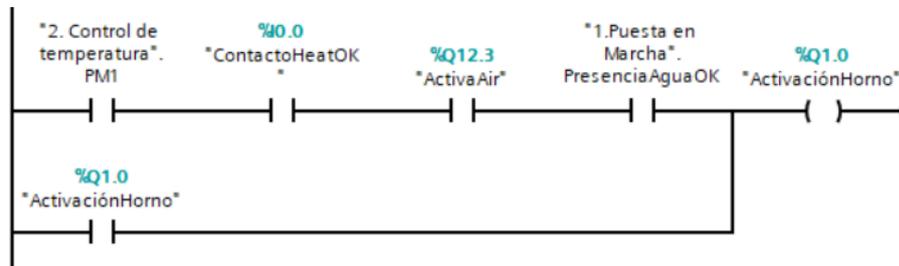


Ilustración 35. Ejemplo realimentación bobina

Para desactivar el contactor principal deberá cumplirse que la variable asociada al botón de encendido del horno en la interfaz HMI este apagado. Es decir, cuando la variable “2. Control de temperatura. PM1” este desactivada. Por seguridad, todas las variables serán desactivadas con la bobina “RESET” que resetea su valor a cero.

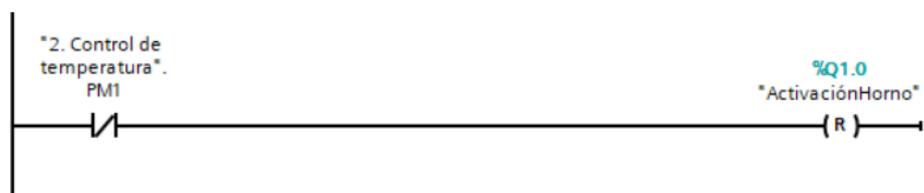


Ilustración 36. Reset contactor principal

En el segundo segmento se gestiona una señalización de que el horno está preparado para calentar. Si el contactor principal esta activado, lo guardaremos en la variable "2. Control de temperatura. PM_OK11". Esto se hace para no tener que llamar múltiples veces a una salida del autómatas y por tener un código más claro.

Y si el contactor principal esta desactivado, reseteamos el modo "SETUP" del horno. Este modo corresponde al periodo en que el horno está a la temperatura correcta y está disponible para trabajar.

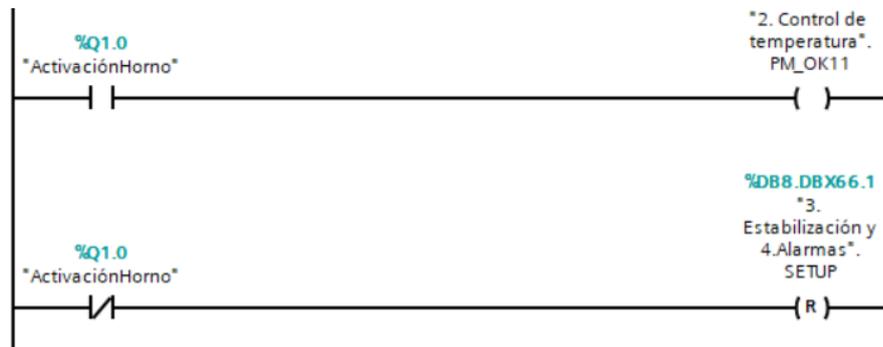


Ilustración 37. Segmento 2 de Puesta en Marcha

El tercer segmento activa la salida de aire "%Q12.3" cuando la entrada "%I0.4" que corresponde con una válvula de aire, este activa. Es importante la gestión del aire ya que durante el proceso de calentamiento que sufren las placas electrónicas, estas sueltan residuos.

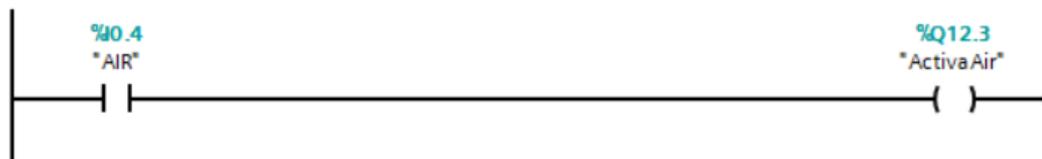


Ilustración 38. Segmento 3 de Puesta en Marcha

El cuarto segmento gestiona una variable que nos indica la presencia de agua. Tendremos presencia de agua o refrigeración cuando las entradas "%I0.6" y "%I0.7" estén activas. Estas entradas son unos sensores de nivel de agua que van directamente conectadas al autómatas.

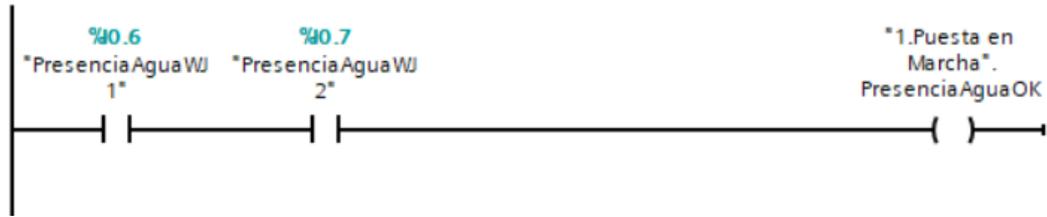


Ilustración 39. Segmento 4 de Puesta en Marcha

4.5.2 Control temperatura V1

En este OB se gestiona el calentamiento en modo potencia de las ocho zonas de las que dispone el horno para calentar y seguir un perfil de temperaturas.

En la interfaz HMI vamos a tener la opción de escoger un modo de potencia en el que el horno va a alcanzar la temperatura deseada en un menor tiempo o un modo económico en el que tardará más tiempo.

Para la gestión de cada zona se utiliza un objeto tecnológico que nos proporciona TIA Portal, un regulador PID. Es un mecanismo de control que a través de un lazo de retroalimentación permite regular velocidad, presión y temperatura entre otras variables. El regulador PID calcula la diferencia entre nuestra variable real y la variable deseada gracias a tres parámetros distintos: el proporcional, el integral y el derivativo.

El bloque PID trabajará cuando tengamos activado el modo potencia y el modo eco este desactivado. También debemos tener activado el contactor principal.

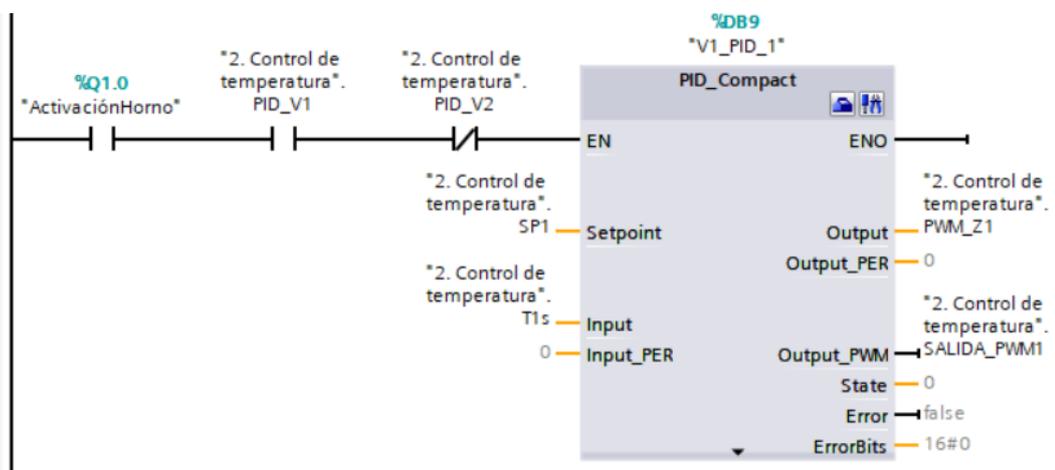


Ilustración 40. Activación del regulador PID

Puede ocurrir que por error se quede trabajando el regulador PID, por eso la salida configurada del regulador no es directamente la salida de los relés que controlan las resistencias de calentamiento.

Las resistencias se calentarán cuando el regulador PID este trabajando, cuando el contactor principal este activo, cuando este activado el modo potencia y no el modo eco, y cuando la temperatura real no haya superado la temperatura establecida como límite superior.

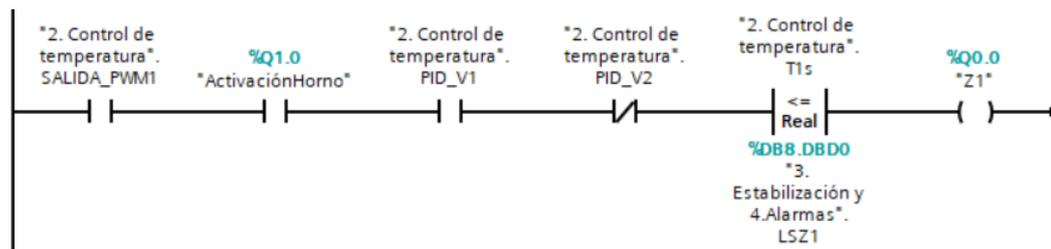


Ilustración 41. Activación resistencia de calentamiento zona 1

Para las siete zonas restantes se procede de la misma manera. Finalmente tenemos un segmento en el que por seguridad gestionamos los modos de potencia y eco. Cuando un modo este activado, el otro no podrá estarlo y viceversa.

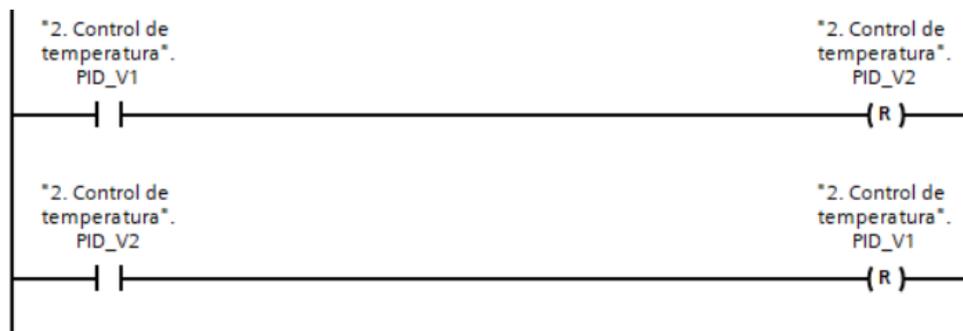


Ilustración 42. Bloqueo PIDs opuestos

4.5.2.1 Configuración bloque regulador PID

Seguir de forma precisa el perfil de temperaturas es una tarea crucial que determinará la calidad del producto final, por ello es muy importante realizar una configuración muy precisa de todos los bloques PID. En concreto, de dieciséis bloques PID, uno por cada zona del horno, tanto en modo potencia como en modo eco.

En primer lugar, tenemos que entender las entradas y las salidas que tiene el bloque “PID_Compact” en TIA Portal.

- **Entradas:**

1. **Setpoint:** Es el valor que vamos a establecer como referencia. El valor que queremos alcanzar y mantener. Puede ser presión, velocidad, temperatura, etc. dependiendo de la magnitud que queramos seguir.
2. **Input:** Será nuestro valor del proceso, en nuestro caso la temperatura de cada zona.
3. **Input_PER:** Es nuestro valor de proceso si procede directamente desde una entrada analógica y no queremos hacer un escalado, sino que dejamos que el bloque “PID_Compact” haga su propio escalado. No la vamos a utilizar.

- **Salidas:**

1. **Output:** Salida del PID.
2. **Output_PER:** Salida del PID ya escalada para una salida analógica.
3. **Output_PWM:** Salida digital PWM. Controlará los relés de estado sólido, que a su vez, estos controlarán las resistencias de calentamiento.

Además de estas entradas y salidas, también tenemos otras para establecer límites, ver el estado del PID, activar modo automático, etc. pero que no haremos uso de ellas. En la siguiente Ilustración podemos apreciar todas:

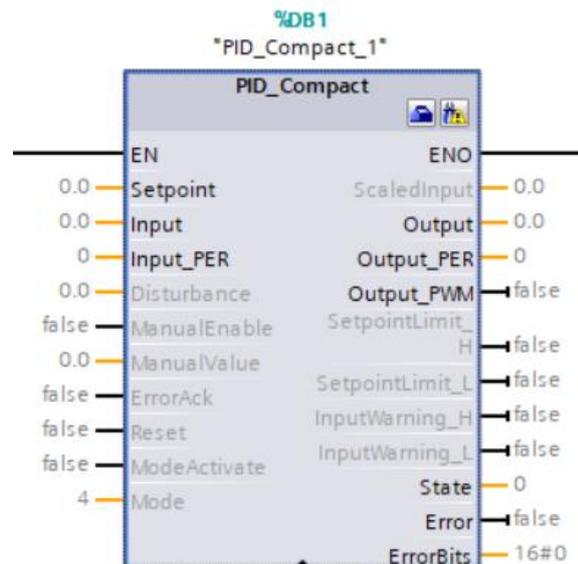


Ilustración 43. Bloque PID_Compact

En segundo lugar, una vez que entendemos las entradas y las salidas del bloque y las seleccionamos, pasamos a la configuración del bloque.

Para ello tenemos que hacer clic en el icono que viene señalado de rojo en la Ilustración [44]. El icono señalado de verde se utilizará más adelante para la puesta en marcha.

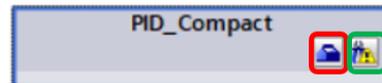


Ilustración 44. Configurar PID_Compact

Dentro de la configuración del bloque tenemos una serie de apartados que vemos en la siguiente Ilustración.



Ilustración 45. Apartados configuración bloque PID_Compact

En 'tipo de regulación' debemos de indicar que la variable a regular es la temperatura y se encuentra en grados centígrados- También activamos el modo automático en caso de rearrancar la CPU. Esto lo hacemos para que en caso de un corte eléctrico o que apaguemos la CPU, los bloques PID comiencen a trabajar automáticamente y no manualmente.

En 'Parámetros de entrada/salida' indicamos que la entrada es digital, y la salida es PWM. En ningún caso estamos haciendo uso de entradas o salidas analógicas.

A continuación, establecemos unos límites dentro del apartado de 'Límites del valor real'.

En 'Límites del valor de salida' establecemos el valor superior con el que trabajará el PWM, esto va a marcar la diferencia a la hora de configurar los PIDs

del apartado de potencia o del apartado eco. También nos da la posibilidad de establecer un comportamiento en caso de error. En nuestro caso, mientras dure el error, el valor de salida será '0.0'.

Finalmente tenemos 'Parámetros PID' donde podemos introducir los que queramos o calculemos, pero TIA Portal nos proporciona una puesta en marcha que nos los calculará automáticamente.

4.5.2.2 Puesta en marcha PID_Compact

Para la puesta en marcha del bloque PID_Compact iremos al símbolo señalado de verde en la Ilustración [44]. Una vez abierto el menú de puesta en servicio hemos de activar la vista online en el icono superior izquierdo que podemos ver en la Ilustración [46].



Ilustración 46. Online, medición y modo de ajuste PID.

Seleccionaremos un tiempo de muestreo acorde a nuestras necesidades y comenzaremos realizando una optimización inicial. Esta determinará el comportamiento del proceso a un escalón del valor de salida y busca el punto de inflexión. Los parámetros PID óptimos se calculan a partir de la pendiente máxima y el tiempo muerto del sistema regulado. Para obtener los mejores parámetros PID, debe efectuarse una optimización inicial y una optimización fina.

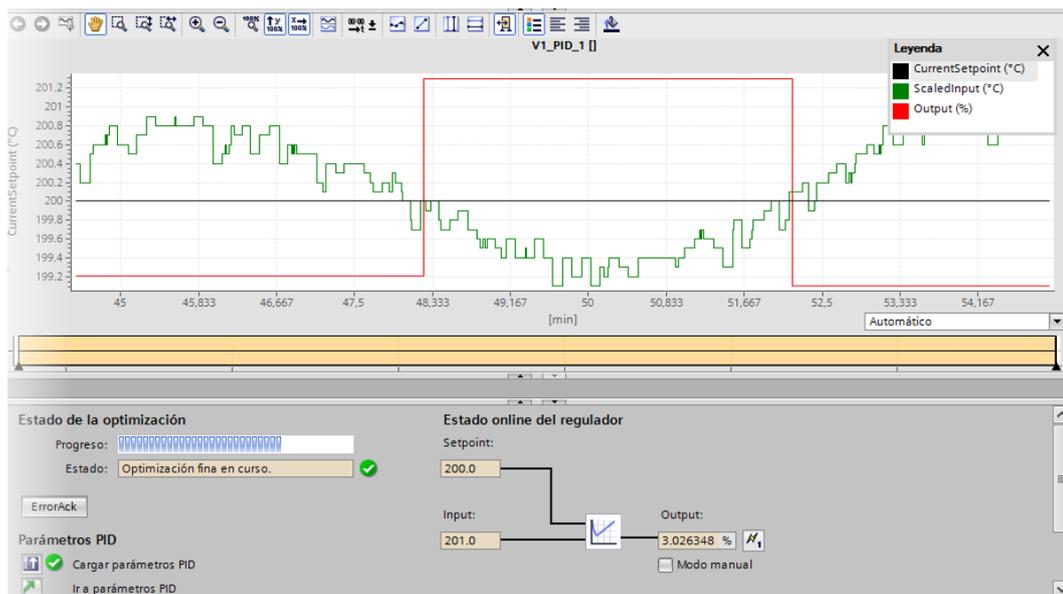


Ilustración 47. Optimización fina PID.

Tras finalizar varias pruebas de optimización con diferentes valores de PID, y diferentes valores de temperatura, conseguimos una parametrización que funciona a la perfección con nuestro sistema. Finalmente debemos de cargar estos parámetros al autómatas como vemos a continuación.

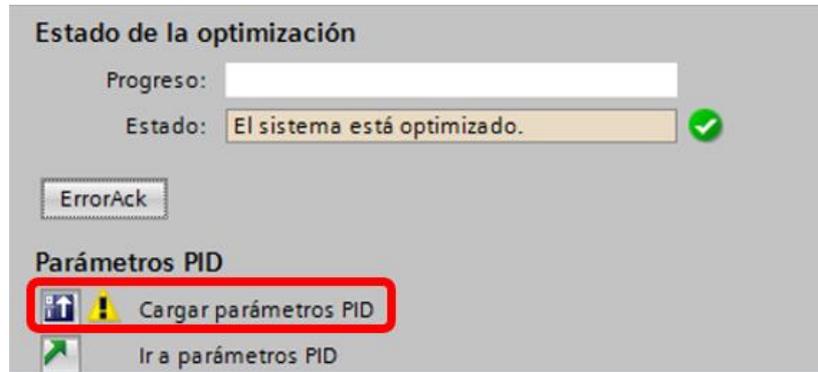


Ilustración 48. Carga de parámetros PID.

Esta fue una de las tareas más complicadas ya que por la arquitectura del horno, tanto la zona 1 (zona de entrada) y zona 8 (zona de salida), costó mucho que estabilizaran su temperatura con el regulador PID. Esto se debía a ser las zonas con más contacto con el exterior, y también, eran zonas que según los perfiles de temperatura que debíamos de seguir, los valores de temperatura no eran muy altos en comparación con otras zonas, por lo que, aunque quisiéramos regular el valor en 300 °C, por contaminación térmica de la siguiente zona ascendíamos a valores de 550 °C.

A pesar de este inconveniente, tras todas las pruebas realizadas se demostró que los perfiles de temperatura que podíamos conseguir con la programación y con el horno eran más que válidos.

4.5.3 Control temperatura V2

Esta OB es prácticamente igual que la anterior, con la única diferencia de la limitación del valor de salida del bloque PID. Como se trata de un modo eco, la salida estará limitada a un valor de 90%, donde calentará más lentamente utilizando menos potencia. Lo ideal sería poder bajar más este porcentaje, pero por la arquitectura y antigüedad del horno ha sido imposible ya que bajándolo por ejemplo a un 80% el horno tardaría en calentar casi 24 horas o incluso no podría llegar a tan altas temperaturas con esa limitación.

4.5.4. Estabilización

El objetivo de este OB es la estabilización de la temperatura en cada zona. Con esta estabilización, es decir, que las temperaturas se muevan entre unos valores predefinidos, podremos empezar a trabajar con el horno. Esta estabilización, junto con la velocidad de la cadena, nos asegura que el perfil de temperaturas se siga correctamente.

Para ello hacemos uso de las instrucciones básicas que nos brinda TIA Portal, en concreto, de las funciones de comparación.

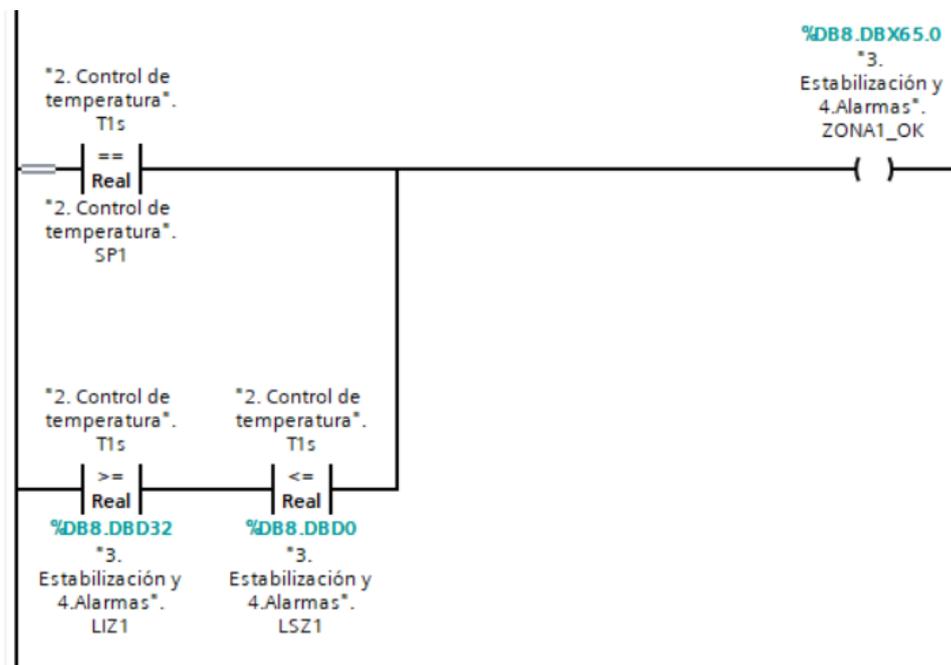


Ilustración 49. Estabilización zona 1

Se proponen dos ramas en paralelo que nos informen sobre el estado de la zona. Una rama compara la temperatura actual con la temperatura de referencia, si esto se cumple la zona 1 estará estable. La otra condición de estabilidad es que la temperatura actual se encuentre entre los límites inferior y superior de temperatura previamente definidos. Estos límites se corresponden con los límites de alarma y no con los límites de advertencia, ya que con los de advertencia si que podemos seguir trabajando con el horno, pero con precaución de no rebasar el resto de los límites.

Este procedimiento se repite para las ocho zonas del horno y se puede comprobar en el apartado de anexos.

4.5.5. Alarmas

En este OB además de definir las alarmas de cada zona, también se define la estabilidad del horno y el inicio del setup. Llamamos setup cuando decimos que el horno está listo para trabajar y solo se tiene que mantener entre determinadas temperaturas. Cuando el horno está comenzando a calentar e intentando alcanzar las referencias, no estamos en setup.

Veamos algunos de los segmentos de esta OB:

- **Segmento 1: Horno estable e inicio de setup.**

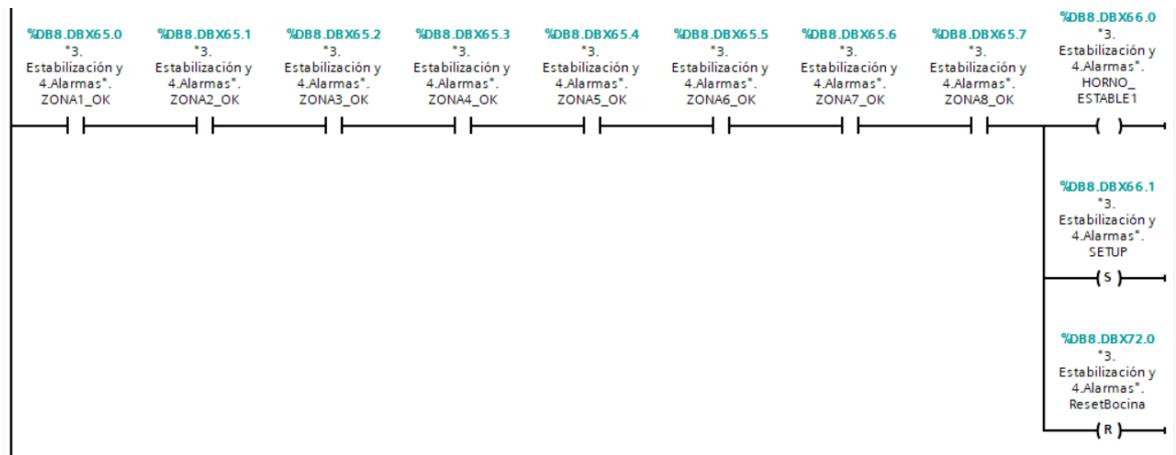


Ilustración 50. Horno estable e inicio de setup. OB Alarmas

Este segmento está vinculado con la OB anterior, ya que cuando estén estabilizadas las ocho zonas y tengamos el “OK”, diremos que el horno está estable. También indicamos que hemos alcanzado el setup. Esto nos servirá de ayuda para la visualización en la interfaz HMI y para poder diferenciar dos situaciones.

Cuando el horno está calentando todavía no ha alcanzado la temperatura de referencia, y tampoco los límites inferiores de temperatura definidos. Según la lógica de programación que hemos seguido, esto daría lugar a una alarma, ya que la temperatura es inferior a la de los límites, pero esta alarma sería incorrecta ya que el horno aún está comenzando a calentar, de modo que se crea este modo de setup. Cuando el horno este estable y este en modo setup, si las temperaturas son inferiores a los límites, esta vez sí tendremos alguna alarma.

Por ultimo y por seguridad, se resetea la variable “3. Estabilización y 4. Alarmas. ResetBocina”, que corresponde a un botón de emergencia para la bocina en la interfaz HMI.

- **Segmento 2: Alarma superior e inferior de zona 1.**

Como hemos comentado anteriormente, tendremos alarma cuando la temperatura actual supere al límite superior o inferior, pero será necesario estar en el modo setup y que tengamos el horno encendido. Esto es porque una vez terminada la jornada laboral, y el horno se apague, el autómatas leerá las temperaturas igualmente pero el botón de encendido del horno en la interfaz HMI estará deshabilitado, por ello, aunque se rebasen estos límites, no existirá ninguna alarma.

Cuando se cumplen todas estas condiciones de alarma, se activa la variable “5. Definición de avisos y alarmas. ALARMAS1. %X0”. Esta variable es de tipo “Word”, es una palabra de dos bytes. Como vemos en la siguiente Ilustración, el byte 0 corresponde con las alarmas y el byte 1 corresponde con los acuses. Esto se hace así porque el primer byte será el que lance el aviso de alarma en la interfaz HMI y el segundo byte será el que acusará las alarmas.

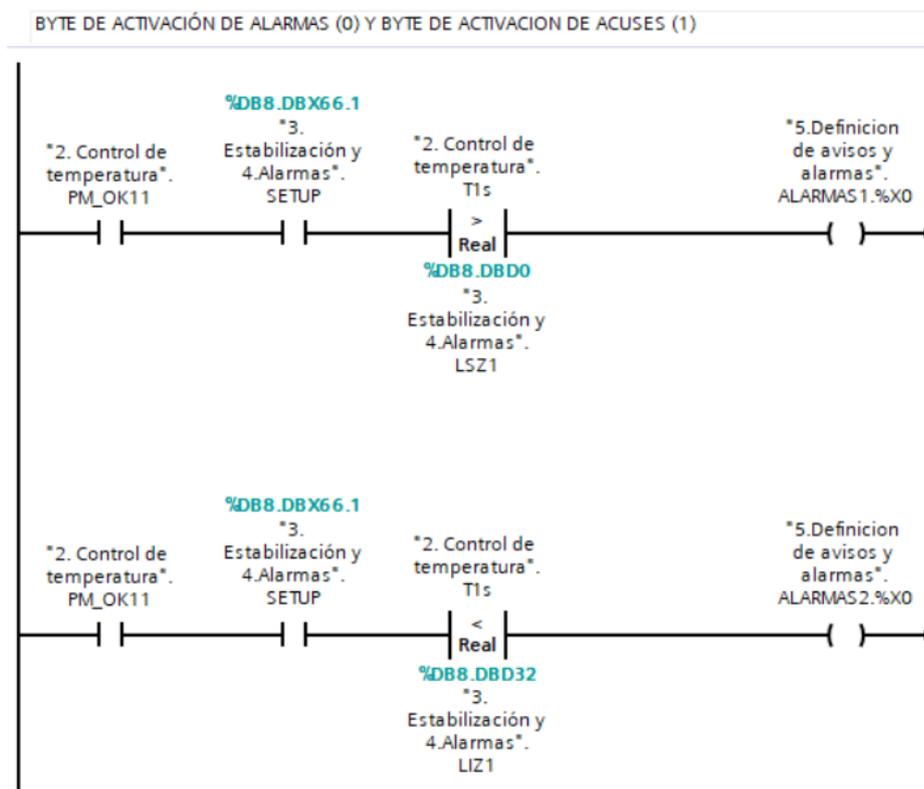


Ilustración 51. Alarmas zona 1

- **Segmento 10: Alarma rotura de resistencia o contaminación térmica**

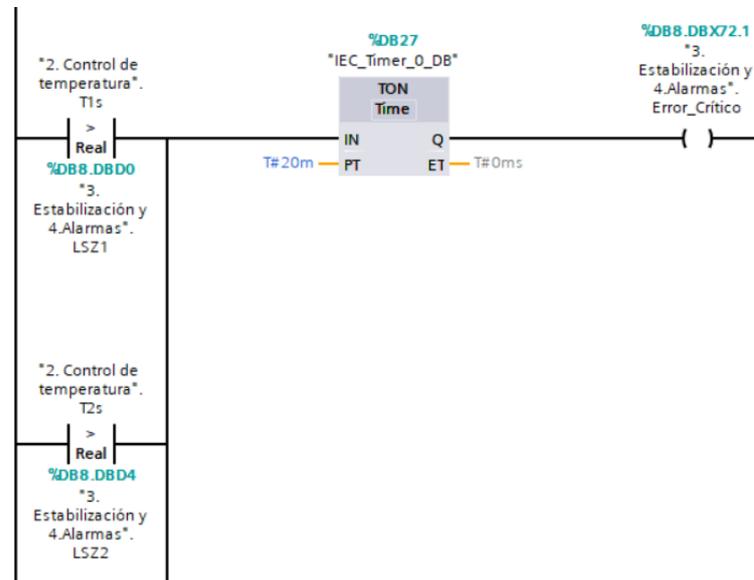


Ilustración 52. Trozo de código de alarma rotura de resistencia

En este segmento se trata de programar dos posibles situaciones que puedan llegar a darse en el horno.

La primera es que se rompa una resistencia o un relé de estado sólido se quede "clavado" y no pare de calentar. La segunda es que mediante contaminación térmica de una zona a otra del horno, cualquiera de estas pueda llegar a rebasar sus límites en exceso. Ambas situaciones provocarían un aumento de temperatura indeseado que tratamos de identificar con un temporizador con retardo a la conexión (TON). Su funcionamiento se basa en que el temporizador se activa cuando alguna de las señales que tenga a la entrada pase a valor uno.

Como vemos en la Ilustración [52], cuando la temperatura de la zona 1 supere el límite superior de la zona 1, la entrada del temporizador estará activa y comenzará una cuenta de veinte minutos que es lo que hemos establecido. Cuando se alcancen estos veinte minutos, la salida se activará. Esta es una variable que nos indicará si tenemos o no un error crítico.

Se muestra solo una parte del código para que se vea con mayor claridad, ya que el resto de código es similar, con las comparaciones del resto de zonas en paralelo.

- **Segmento 11: Envío error crítico mediante variable Word para el HMI**

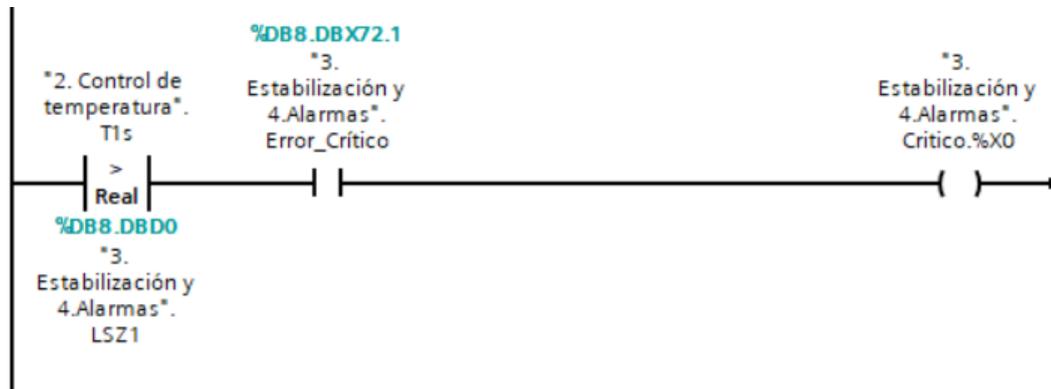


Ilustración 53. Envío alarma error crítico

Con este segmento se busca almacenar mediante una variable de tipo word, la existencia de un error crítico en cualquiera de las zonas. Las direcciones de cada bit de la variable word, serán variables de disparo para que aparezcan como avisos dichas alarmas en la interfaz HMI.

- **Segmento 12: Alarma cadena**

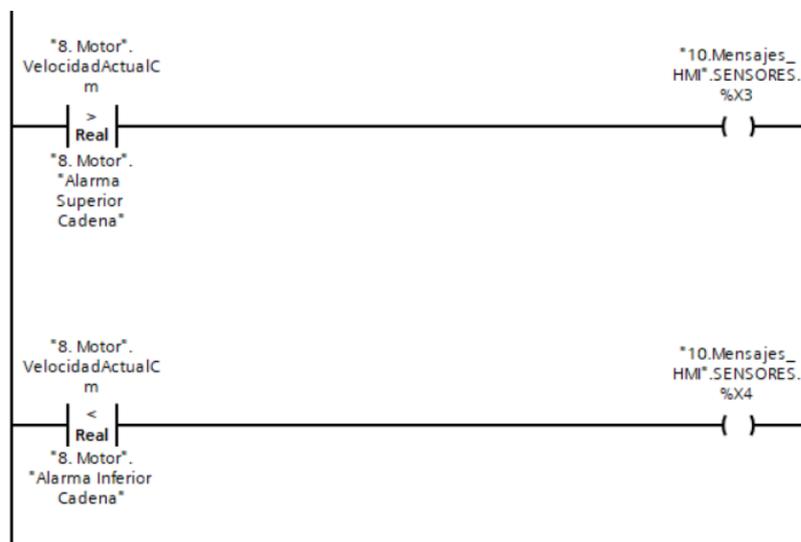


Ilustración 54. Envío alarma velocidad de cadena para visor de avisos interfaz HMI

Para finalizar creamos una alarma para la velocidad de la cadena. Si aumenta demasiado y supera un determinado valor que será introducido por el operario a través de la interfaz HMI, o disminuye superando otro valor

determinado, guardaremos esta información en la variable tipo word 'SENSORES', en sus bits 4 y 5. Esta información será variable de disparo para que aparezca el aviso en el visor de avisos de la interfaz HMI.

4.5.6. Gestión de alarmas

Para esta OB el objetivo es la gestión de los distintos errores que hemos definido, además también se incluyen dentro de esta OB, unos segmentos donde se realiza la transformación de la temperatura de cada zona que explicaremos más adelante.

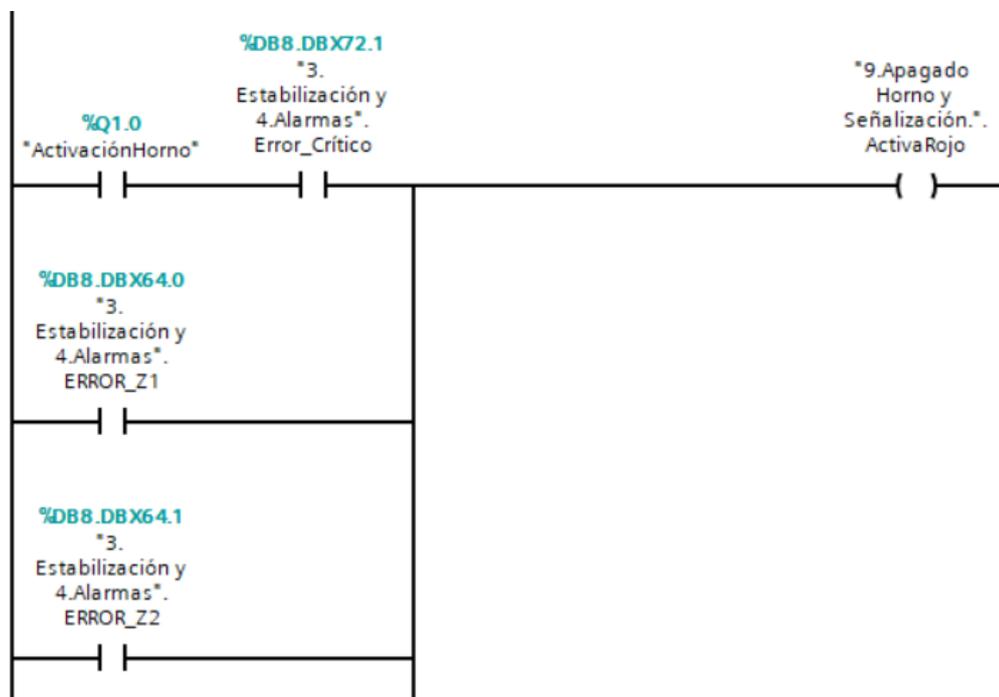


Ilustración 55. Gestión de alarmas

Para este segmento de código como vemos en la Ilustración [55], tratamos de gestionar los errores, de modo que si tenemos cualquier error en alguna zona, o tenemos un error crítico mientras el horno esta encendido, activaremos la señal roja.



Ilustración 56. Definición de errores

En la Ilustración [56] se encuentra definido lo que sería un error. Las variables "ALARMAS1" y "ALARMAS2" son de tipo Word, y en sus ocho primeros bits tienen almacenado si hay algún error. En el ejemplo de la Ilustración, si la temperatura de la zona 1 supera los límites de alarma superiores, la variable "ALARMAS1" almacenará en su primer bit un uno lógico. Esto activará el error en zona 1. Si por el contrario, la temperatura supera los límites inferiores, la variable que actuará será "ALARMAS2". Es decir, una almacenará las alarmas de temperatura superior y la otra de temperatura inferior.

4.5.7. Programación automática

En este OB se busca programar un modo automático, en el que el operario mediante la interfaz HMI, pueda elegir la hora de encendido y la hora de apagado del horno, además del modo de funcionamiento, que sería el modo de potencia o el modo eco.

Para ello, en primer lugar, el autómatas necesita saber el día y la hora actuales. Esto se va a conseguir gracias al bloque que nos proporciona TIA Portal conocido como "RD_LOC_T". Lo encontraremos en el apartado de instrucciones avanzadas, dentro de la carpeta de fecha y hora como vemos en la Ilustración siguiente.

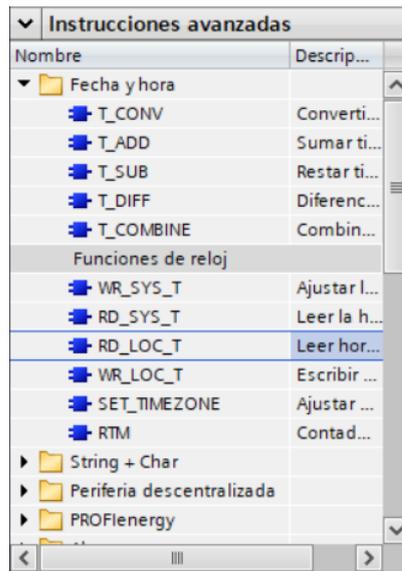


Ilustración 57. Ubicación bloque RD_LOC_T

Para poder llamar a este bloque en un OB, previamente debemos de crear un bloque de datos DB donde guardaremos nuestras variables de tiempo, de tipo DTL en TIA Portal. Una variable tipo DTL desglosa el año, el mes, el día, el día de la semana, la hora, los minutos, los segundos y los nanosegundos. Esto lo vemos en la variable que hemos creado para capturar los tiempos de la CPU.

Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...
▼ Static				<input type="checkbox"/>
■ ▼ Tiempos_CPU	DTL	0.0	DTL#1970-01-01-4	<input checked="" type="checkbox"/>
■ YEAR	UInt	0.0	1970	<input checked="" type="checkbox"/>
■ MONTH	USInt	2.0	1	<input checked="" type="checkbox"/>
■ DAY	USInt	3.0	1	<input checked="" type="checkbox"/>
■ WEEKDAY	USInt	4.0	5	<input checked="" type="checkbox"/>
■ HOUR	USInt	5.0	0	<input checked="" type="checkbox"/>
■ MINUTE	USInt	6.0	0	<input checked="" type="checkbox"/>
■ SECOND	USInt	7.0	0	<input checked="" type="checkbox"/>
■ NANOSECOND	UDInt	8.0	0	<input checked="" type="checkbox"/>

Ilustración 58. Variable tipo DTL

Para la configuración del bloque 'RD_LOC_T' debemos saber que la primera patilla de salida nos devuelve el estado del bloque, es decir, nos devolverá un valor que nos indique si la ejecución se ha realizado sin errores o qué tipo de error ha ocurrido. Este valor lo guardaremos en 'STATUS_RELOJ'.

La segunda patilla de salida nos almacenará el día y la hora en una variable de tiempo, en nuestro caso en 'TIEMPOS_CPU' de tipo DTL.

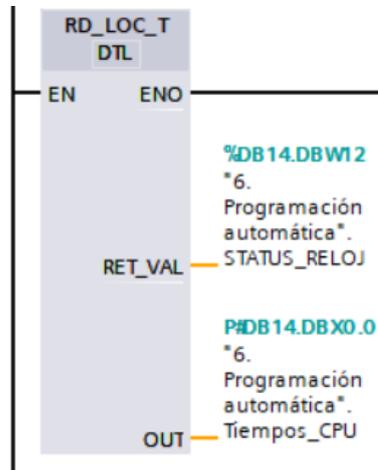


Ilustración 59. Bloque RD_LOC_T

Una vez hemos capturado el día y la hora, vamos a desglosar los días de la semana. Para ello usamos los bloques de comparación, donde vamos a comparar el día de la semana, almacenado en la variable DTL 'TIEMPOS_CPU', con un número del 1 al 7. En TIA Portal el primer día de la semana, el número 1, corresponde con el Domingo, el número 2 con el Lunes y así sucesivamente. Tras hacer la comparación, si se cumple, guardaremos un 1 lógico en nuestra variable de tipo bool correspondiente con el día de la semana como vemos en la Ilustración [60].

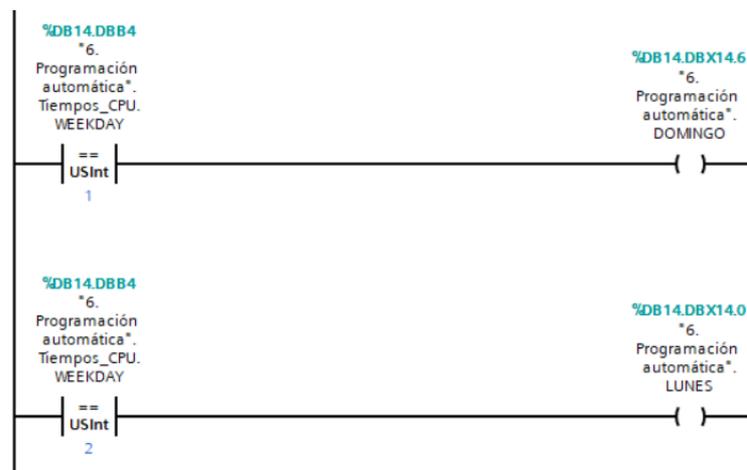


Ilustración 60. Desglose Domingo y Lunes

Tras hacer el desglose de los días de la semana, creamos un segmento con el que el autómatas sabrá si estamos en modo automático o no.



Ilustración 61. Activación modo automático

La variable 'ENCENDIDO_AUTOMÁTICO' corresponde con el interruptor de modo automático de la interfaz HMI. Si el interruptor está activado, el autómatas guardará un 1 lógico en la variable 'ON'.

Finalmente programamos el encendido del horno y el modo de funcionamiento para todos los días de la semana. Por ejemplo, para el Domingo, si el operario deja pulsado el modo automático, la variable 'ON' tendrá un 1, y cuando la CPU indique el día 1 de la semana, la variable 'DOMINGO' tendrá un 1 también. A partir de ahí, tendremos dos ramas, la de encendido y la de apagado.

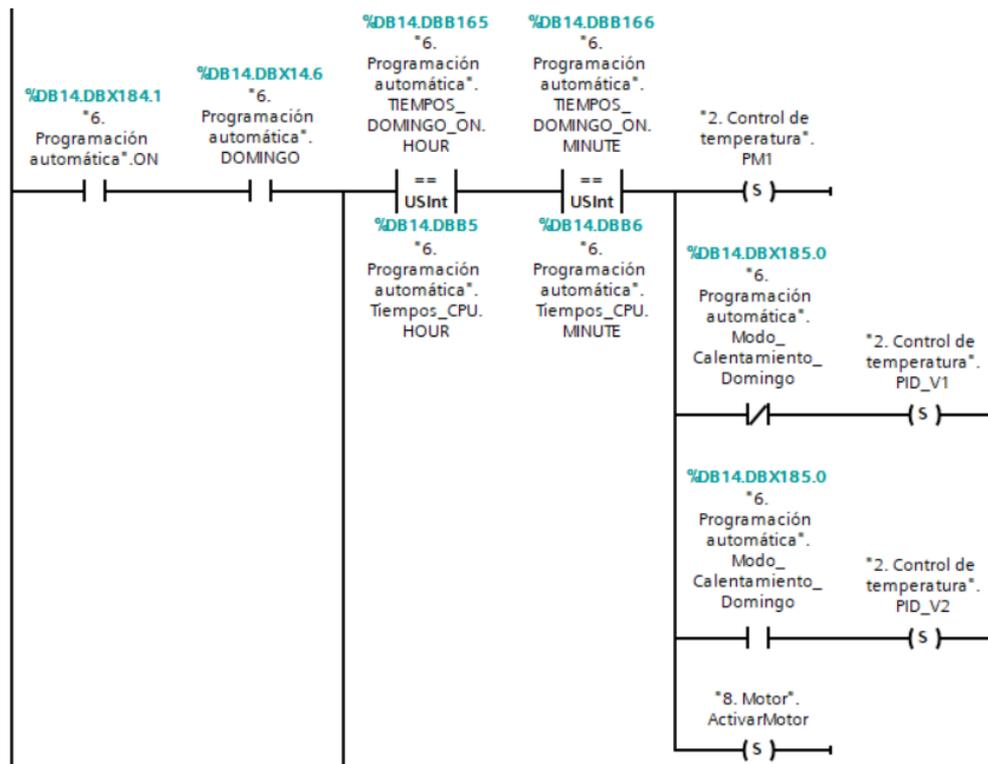


Ilustración 62. Programación automática Domingo rama de encendido

Si la hora y los minutos que lee la CPU, coinciden con los valores que hemos guardado en 'DOMINGO_ON. HOUR' y 'DOMINGO_ON. MINUTE' a través de la interfaz HMI, encenderemos el horno con 'PM1', la cadena comenzará a girar con 'ActivarMotor' y dependiendo del modo que tengamos configurado para ese día, se activará 'PID_V1' o 'PID_V2' que corresponden con modo potencia y modo eco respectivamente.

En la rama de apagado simplemente quedamos programado el apagado del horno y no de la cadena. Esto se debe a que normalmente la cadena quedará activada, salvo un apagado manual del operario, para que mediante su movimiento y del flujo de aire, se disipe mejor el calor cuando el horno no esté en funcionamiento. Además, también esto es una buena práctica ya que si ocurre cualquier tipo de problema mientras el horno está cargado de placas electrónicas y la cadena deja de moverse, estas placas recibirán otro perfil de temperaturas y terminarán por salir defectuosas.

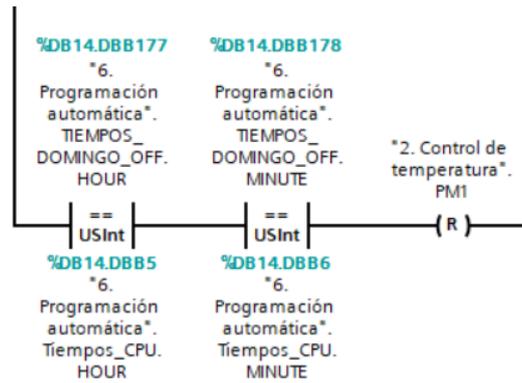


Ilustración 63. Programación automática Domingo rama de apagado

El resto de los días se programan de la misma manera en segmentos diferentes.

4.5.8. Warnings

Para la programación del OB 'Warnings' primero debemos de definir estas advertencias. Una advertencia significa que el horno no está trabajando en su zona estable, pero está muy cercano a ella, donde ha superado unos valores de temperatura. Aun así, se puede seguir trabajando, pero con una pequeña supervisión ya que nos encontramos más cerca de la zona de alarmas.

La definición de estas advertencias se basa en que solo pueden existir mientras que el horno se encuentre encendido y en modo setup. Cumpliendo estas dos condiciones, para encontrarnos con un warning nos debemos de encontrar en una zona de temperaturas determinada. Esta será la que nosotros establezcamos al cargar el programa de calentamiento. Por ejemplo, el warning superior de la zona 1 que vemos en la Ilustración [64], será la temperatura comprendida entre la temperatura límite de warning superior y la temperatura límite de alarma superior.

Cuando se hayan cumplido todas estas condiciones, guardaremos el warning en una variable de tipo word llamada 'WARNING1'. Dependiendo de que el warning sea superior o inferior, y dependiendo de la zona, este se guardará en un bit u otro de dicha variable word.

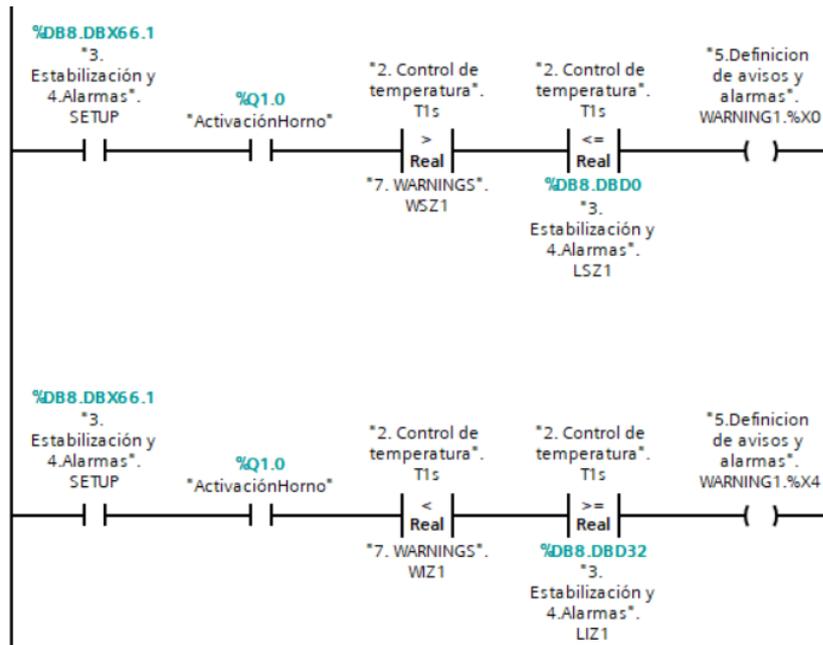


Ilustración 64. Definición warning zona 1

Una vez definidos todos los warnings de todas las zonas, solo queda agrupar los warning superior e inferior con sus zonas respectivas para una mejor gestión de estos.

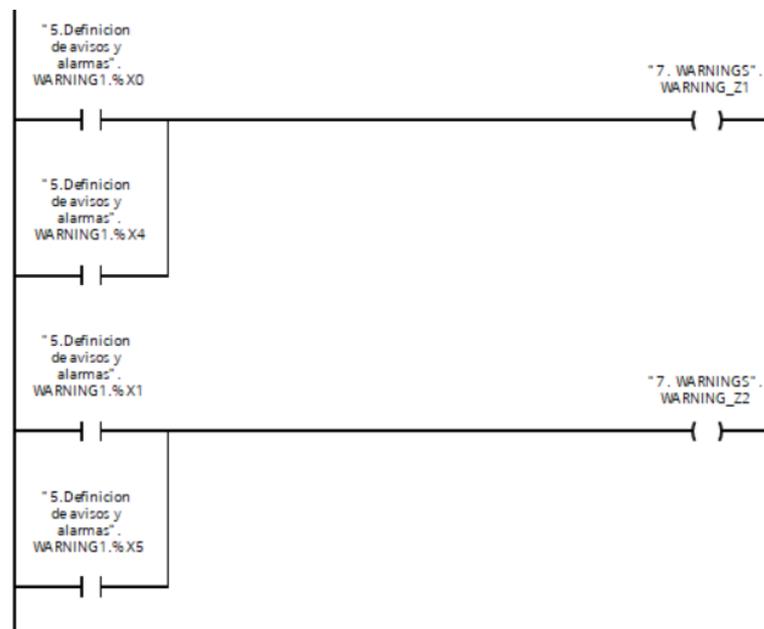


Ilustración 65. Envío warning para visor de avisos de la interfaz HMI

4.5.9. Motor

Para el control del variador de frecuencia Sinamics G120C, vamos a hacer uso de un bloque que nos ofrece TIA Portal, el bloque 'SinaSpeed'. Este bloque sirve para el control del eje con regulación de velocidad procedente del telegrama estándar 1. Lo vamos a encontrar en la zona de instrucciones, dentro de paquetes opcionales en la carpeta de Sinamics como vemos en la siguiente Ilustración.



Ilustración 66. Paquetes opcionales Sinamics

El bloque 'SinaSpeed' debemos de agregarlo al segmento que queramos y automáticamente se creará un DB de instancia para este bloque, donde se almacenarán todas sus variables. A continuación, vamos a ver todas sus patillas de entrada y de salida.

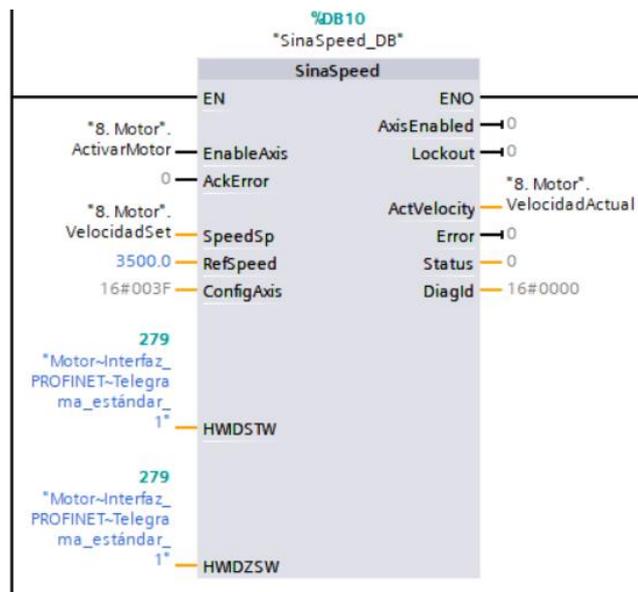


Ilustración 67. Bloque SinaSpeed para controlar el G120C

- **Entradas:**
 - EnableAxis: Con ella habilitamos o no el movimiento del eje.
 - AckError: Para resetear en caso de fallo.
 - SpeedSp: Sería la consigna de velocidad.
 - RefSpeed: Corresponde con la velocidad de referencia que configuramos al variador.
 - ConfigAxis: Viene por defecto después de configurar el variador.
 - HWDSTW: Para indicar el identificador del telegrama.
 - HWDZSW: Para indicar el identificador del telegrama
- **Salidas:**
 - AxisEnabled: Para saber si el eje ha sido habilitado.
 - Lockout: Para saber si el eje se encuentra bloqueado.
 - ActVelocity: Velocidad actual del eje.
 - Error: Código de error con el que sabremos si ha habido error y cual.
 - Status: Estado del bloque.
 - DiagId: Id diagnóstico.

Una vez conocemos todas sus patillas y para qué sirven solo debemos de asignar las variables que vayamos a utilizar y asignar el estándar de comunicación que se vaya a utilizar. Las variables 'Activa Motor', 'VelocidadSet' y 'VelocidaAct' estarán vinculadas con la interfaz HMI y serán introducidas por el operario.

Finalmente, ya que los parámetros de velocidad del bloque 'SinaSpeed' son vueltas, y lo que debemos de ver por pantalla en la interfaz HMI es la velocidad en centímetros por minuto, programamos dos segmentos con los que haremos la conversión de estos valores de velocidad.

Para hacer el cálculo de este factor de conversión se podía realizar teóricamente con el número de dientes de los engranajes y algún dato más, pero que personalmente desconocía. De modo que para hallar el factor de conversión hice una prueba práctica donde realicé los siguientes pasos:

1. **Indicar** al bloque 'SinaSpeed' una velocidad set point.
2. **Colocar** un pequeño objeto encima de la cadena, a la entrada del horno, pero con este apagado,
3. **Realizar** una marca de inicio y comenzar a medir el desplazamiento que tiene el objeto en un minuto.
4. Tras hallar la marca de fin después de un minuto, **comprobar** nuevamente un minuto las marcas de fin e inicio.
5. **Repetir** estos pasos cinco veces, pero con diferentes valores de set point.

6. Tras anotar los valores de set point y su correspondiente velocidad en centímetros por minuto, **hallar** un factor de conversión entre estos valores.

Por último, el factor de conversión hallado corresponde con el número 141. El número de vueltas del bloque 'SinaSpeed' dividido entre 141, nos dará la velocidad en centímetros por minuto, y esta, si es multiplicada por 141 nos dará la velocidad correcta para introducir en el bloque.

A continuación, se ilustra como recibimos la velocidad en centímetros por minuto de la interfaz HMI y la transformamos a una velocidad válida para el bloque. También se ilustra el proceso contrario respectivamente.

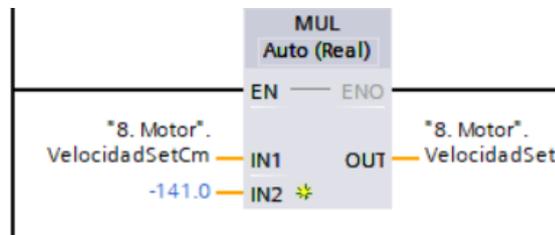


Ilustración 68. Conversión velocidad de la cadena para entrada SinaSpeed

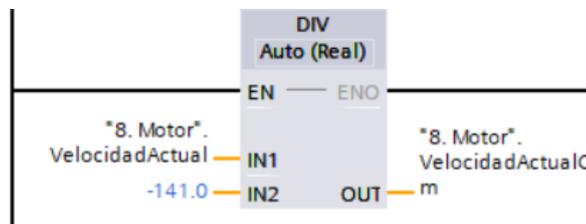


Ilustración 69. Conversión velocidad de cadena a cm por minuto

4.5.10. Apagado horno y señalización

En este OB se va a gestionar el control de la columna de señalización que lleva consigo el horno, además de la señalización dentro de la interfaz HMI. La columna de señalización además de incorporar los tres colores de un semáforo, verde, naranja y rojo. También incorpora un zumbador.

En primer lugar, definimos la señalización del color verde. Las condiciones para que este color se encienda son:

- No solaparse con ningún otro color.
- El horno este encendido, tengamos la salida 'Q1.0' activada.

- El horno este en modo estable, es decir, que se las temperaturas estén en sus rangos y se pueda trabajar con él.

De este modo el operario al ver el color verde en la columna de señalización sabrá que puede comenzar a cargar el horno de placas electrónicas.

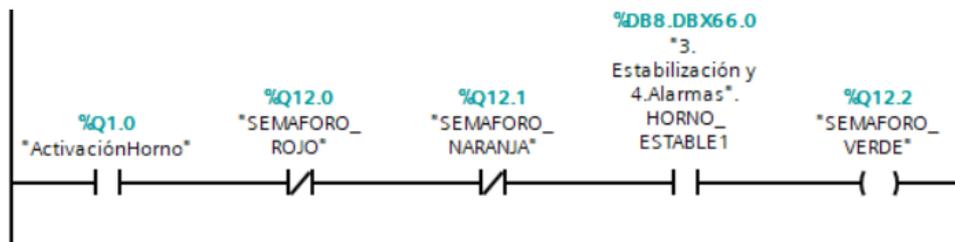


Ilustración 70. Encendido color verde de la columna de señalización

En segundo lugar, programamos el color rojo. Este será el más restrictivo y se activará cuando tengamos algún tipo de alarma. También definimos que, si el horno se encuentra en estado de alarma durante veinte minutos o más, el horno se apague. Normalmente el operario detectará antes de veinte minutos el error, acusándolo y solucionando, pero si por el motivo que sea esto no ocurre, el horno se apagará para evitar errores mayores.

Se utiliza un temporizador con retardo a la conexión para el apagado del horno.

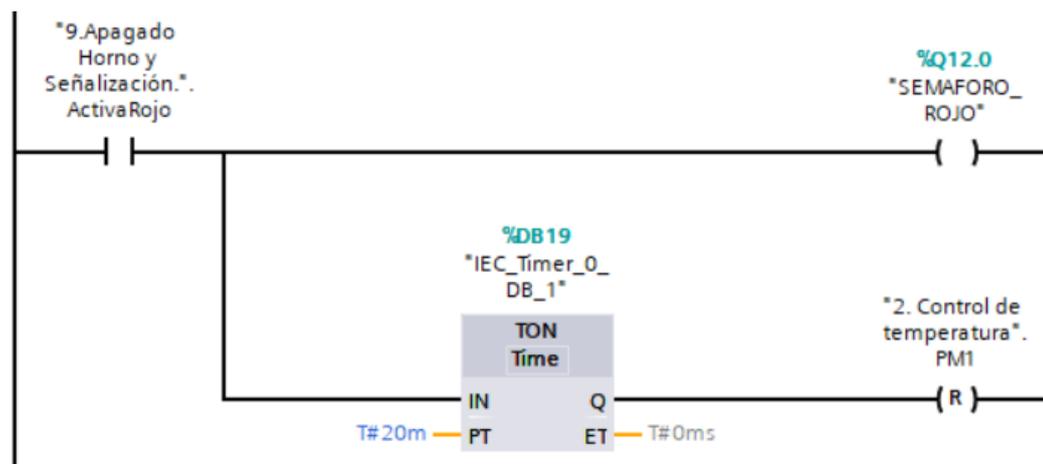


Ilustración 71. Encendido color rojo de la columna de señalización

En tercer lugar, procedemos con el color naranja. Este color está destinado para señalar cualquier tipo de warning o para indicar que el horno está encendido y calentando, pero todavía no se puede trabajar con él. Esta segunda opción que comentamos se señalará con el color naranja y un parpadeo de un segundo de frecuencia. Para lograr esto previamente hemos de activar las marcas de ciclo.

Para ello solo debemos de acudir a la vista de dispositivos, y seleccionando nuestro PLC entraremos en sus propiedades. Finalmente vamos a la sección de 'Marcas de sistema y de ciclo' y activamos las marcas de ciclo. Podemos establecerlas en el byte que nosotros queramos siempre que esté libre. En nuestro caso utilizamos la marca de ciclo 300 y como vemos en la Ilustración siguiente, se nos indica en que bit esta cada marca con su determinada frecuencia.

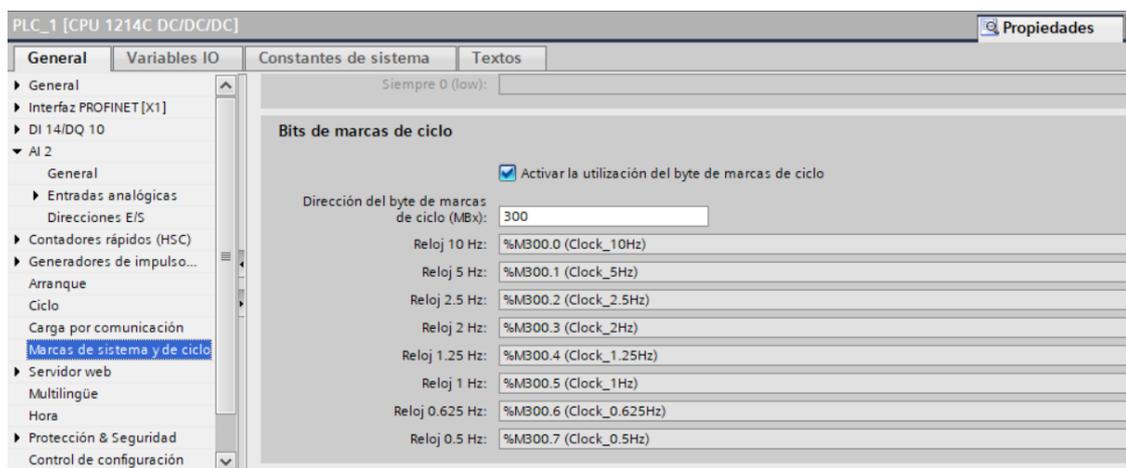


Ilustración 72. Activación marcas de ciclo

Existe la posibilidad de que el color verde y el color naranja estén activados simultáneamente. Esto ocurre cuando el horno está trabajando en modo estable, pero nos encontramos con un warning en la zona 1 por ejemplo. En la Ilustración [73] vemos el código realizado para el color naranja.

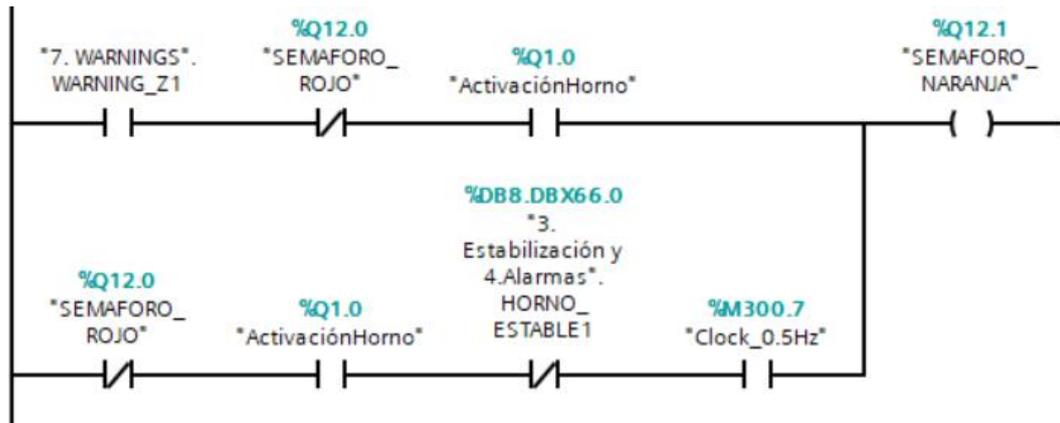


Ilustración 73. Encendido color naranja de la columna de señalización

Como habíamos dicho, el color rojo es el más restrictivo, por ello si tenemos el color rojo activado por cualquier alarma, los warning no se verán reflejados en la columna de señalización porque son menos importantes. Estos avisos quedarán reflejados en el visor de avisos de la interfaz HMI.

Por último, se crean unas variables que serán utilizadas para lograr las animaciones dentro de la interfaz HMI.

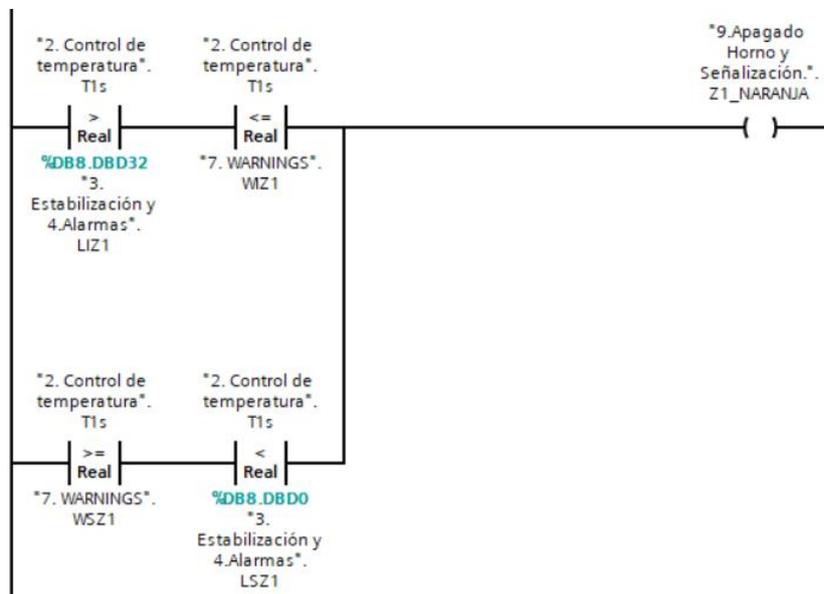


Ilustración 74. Variable para animación color naranja de los cuadros de zona de la interfaz HMI

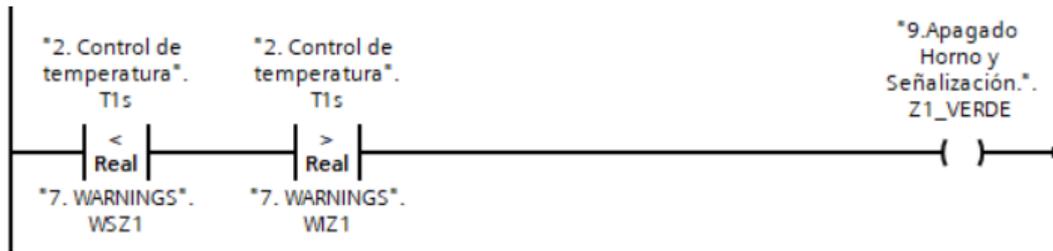


Ilustración 75. Variable para animación color verde de los cuadros de zona de la interfaz HMI

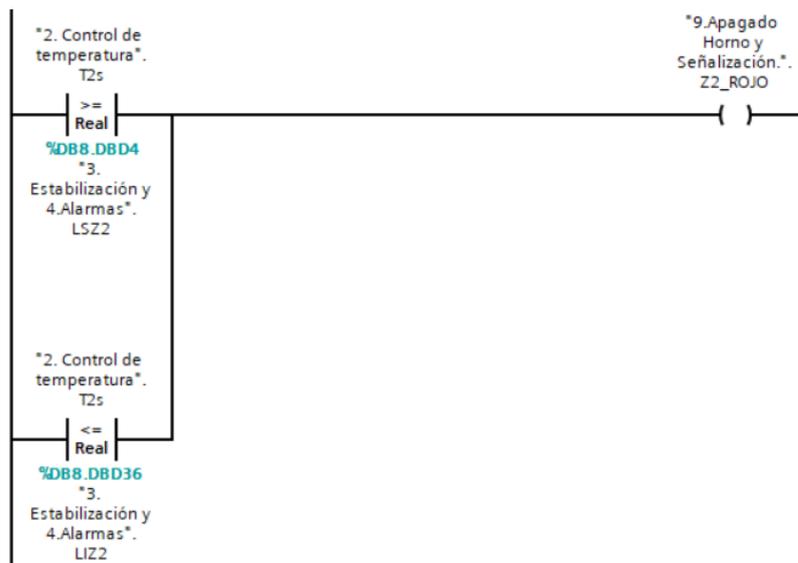


Ilustración 76. Variable para animación color rojo de los cuadros de zona de la interfaz HMI

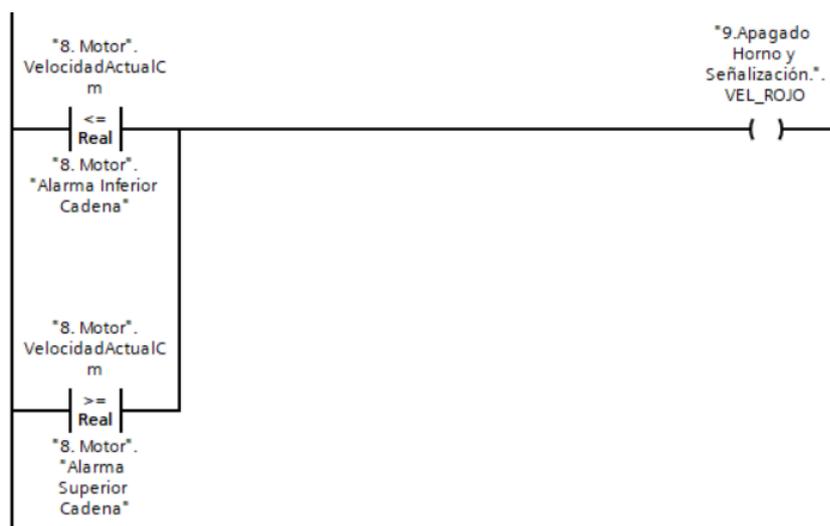


Ilustración 77. Variable para animación color rojo del cuadro de velocidad de la cadena en la interfaz HMI

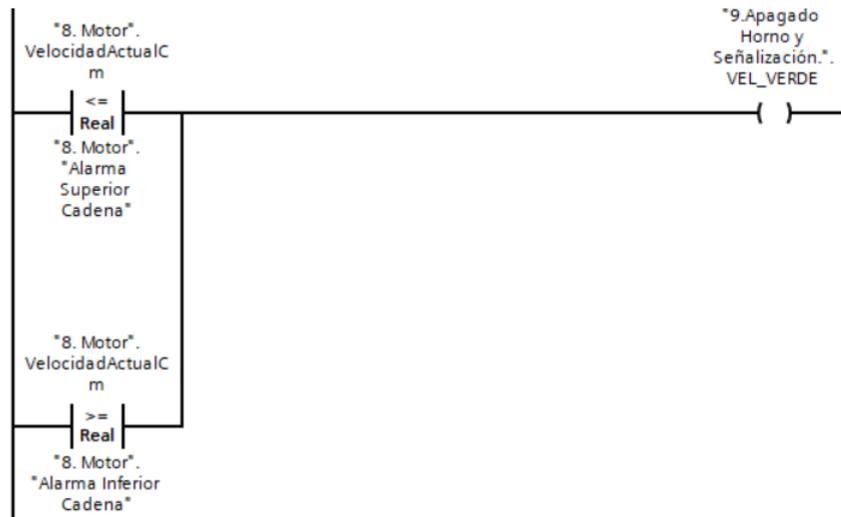


Ilustración 78. Variable para animación color verde del cuadro de velocidad de la cadena en la interfaz HMI

Las barras de temperatura de cada zona indican en qué zona se encuentra la temperatura, zona estable, alarma o warning, y buscamos que los cuadros de cada zona indiquen el mismo color dependiendo de la zona. Con estas variables conseguimos una mayor cantidad de información dentro de la interfaz HMI. En la siguiente Ilustración podemos ver un ejemplo de lo explicado.

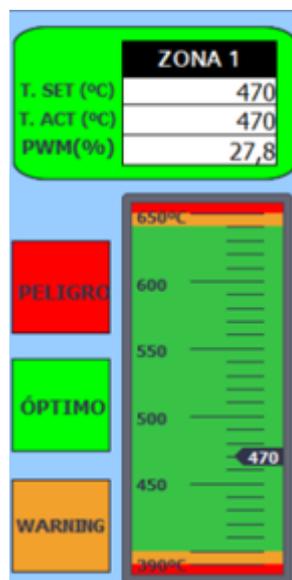


Ilustración 79. Correspondencia de color del cuadro zona con la barra de temperaturas

4.5.11. Mensajes HMI

Este OB está destinado a configurar los distintos avisos que serán mostrados en la interfaz HMI. El procedimiento es sencillo, establecemos las condiciones que veamos oportunas, y cuando se cumplan enviamos un 1 lógico a un bit. Ese bit será luego un bit de disparo que muestre por pantalla el aviso.

Esto es solo es la recogida de información que utilizará la interfaz HMI. Para que luego estos avisos se vean reflejados queda por configurarlos en apartados posteriores.

A continuación, vemos tres ejemplos:

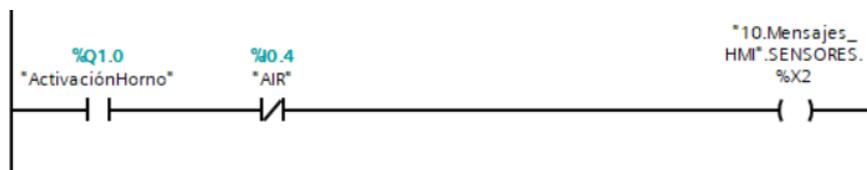


Ilustración 80. No hay presencia de aire

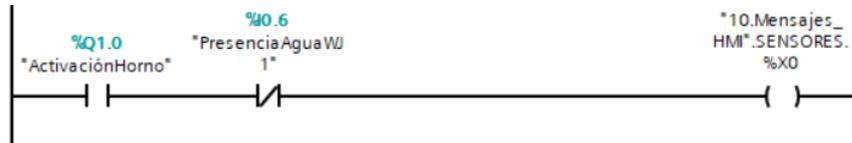


Ilustración 81. No hay presencia de agua en el sensor 1



Ilustración 82. Programación automática desactivada

5. Desarrollo de la interfaz hombre-máquina

En este apartado se va a explicar qué es una interfaz hombre-máquina, así como su funcionamiento. A su vez, se va a detallar el procedimiento que se ha seguido para diseñar y programar la interfaz HMI para la empresa.



Ilustración 83. Pantallas HMI de Siemens.

5.1 Descripción interfaz hombre-máquina

Una interfaz hombre-máquina, más conocida como HMI por sus siglas en inglés, es la interfaz entre la máquina procesadora y el operario. En esencia, es como un panel de control del operario. Es la herramienta principal que los operarios y los supervisores de línea utilizan para coordinar y controlar procesos y máquinas industriales y de fabricación. La interfaz HMI traduce complejas variables de proceso en información utilizable y procesable.

En nuestro caso, la máquina procesadora es el PLC S7-1200 y el objetivo final es mostrar información en tiempo real, como puede ser la temperatura de cada zona del horno o la velocidad de la cadena. Esta información debe de darse de forma fácilmente comprensible para facilitar la labor del operario.

5.2. Creación interfaz hombre-máquina

Continuaremos con el ejemplo anteriormente creado en el apartado 4 para realizar una explicación paso a paso de la creación de una interfaz HMI.

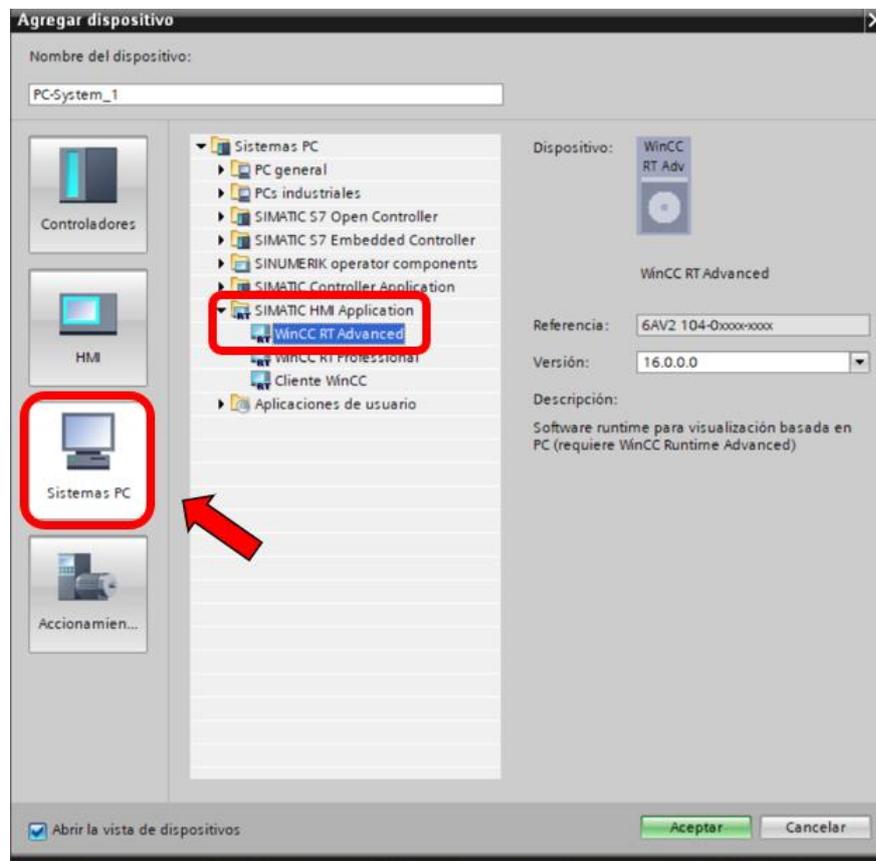


Ilustración 84. WinCC RT Advanced – HMI

En primer lugar, debemos de agregar un dispositivo, el que queramos, desde una pantalla KTP700 Basic de 7” (que la encontraremos dentro de “HMI”), o una pantalla HMI simulada con WinCC Runtime Advanced, que la encontraremos en “Sistemas PC”. Como podemos ver en la Ilustración [84], nuestro caso es el segundo ya que no disponíamos de una pantalla física y decidimos incorporar la interfaz en la pantalla del PC.

En segundo lugar, tras incorporar este nuevo dispositivo a nuestro proyecto, nos encontraremos algo como la Ilustración [85]. En el nuevo árbol de proyecto, tenemos “Simatic PC Station”, y al abrir esa pestaña nos encontramos dentro con la pantalla HMI con “WinCC RT Advanced”. Una vez hay, podemos comenzar a diseñar nuestra interfaz.

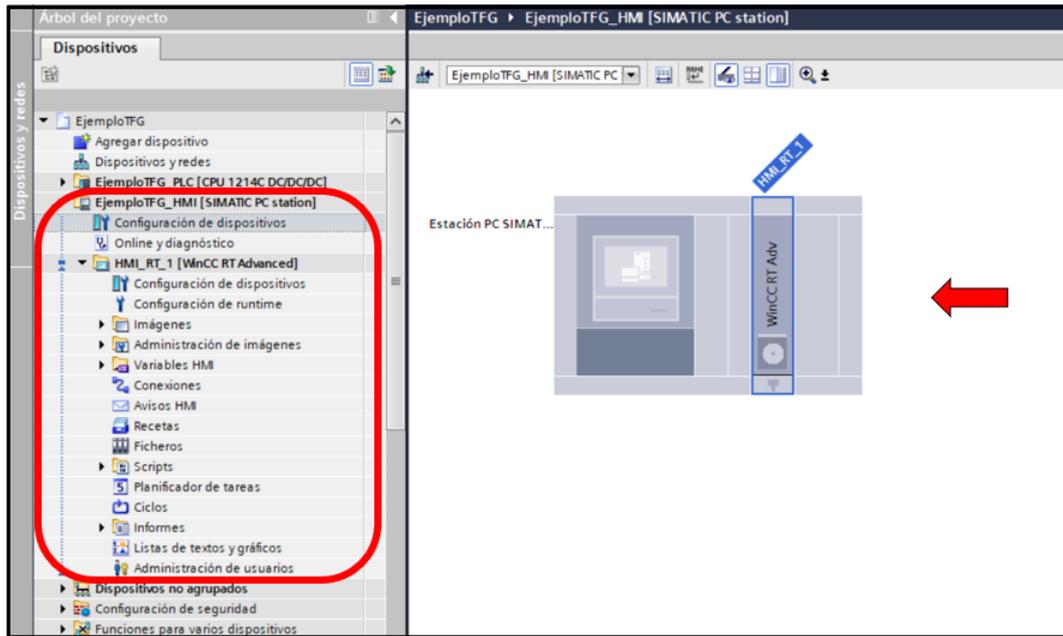


Ilustración 85. Nuevo árbol de proyecto tras agregar dispositivo HMI.

En tercer lugar, entramos en la pestaña “Imágenes” y agregamos una. Podemos agregar todas las que queramos y serán las que aparezcan en la interfaz.

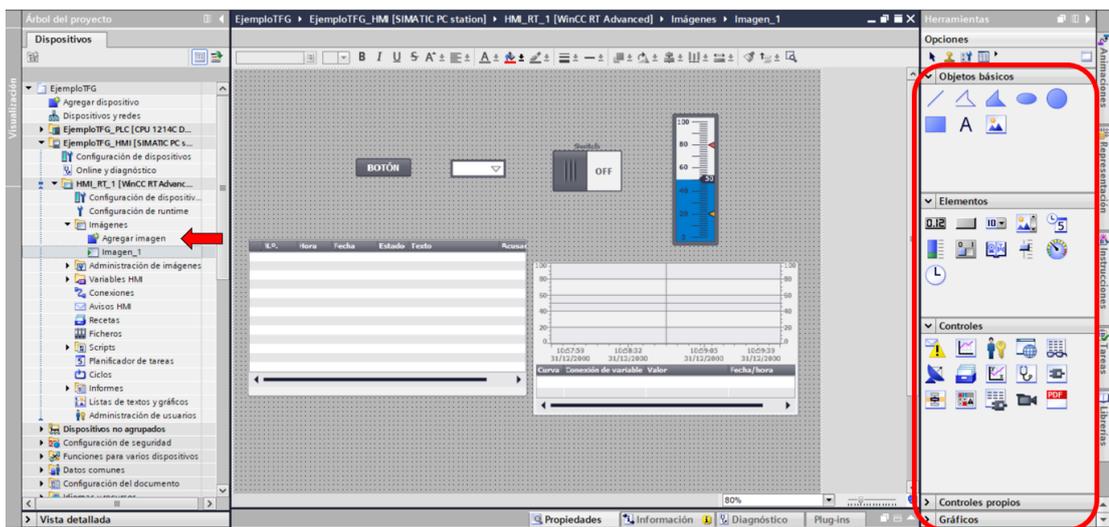


Ilustración 86. Agregar y diseñar imagen.

Una vez nos encontramos aquí, solo queda diseñar la imagen gracias a las herramientas que nos proporciona TIA Portal como vemos a la derecha de la Ilustración [86]. Tenemos múltiples opciones:

- Añadir figuras como cuadrados, circunferencias, rectas, etc.
- Añadir botones.
- Añadir interruptores.
- Añadir recuadros de entrada, de salida o de entrada/salida.
- Añadir visor de avisos.
- Añadir gráficas.

Principalmente haremos el uso de estas, pero hay infinidad de opciones más.

En las siguientes Ilustraciones se va a mostrar el diseño final de nuestra interfaz, con todos los botones, recuadros, gráficas, etc., programados.

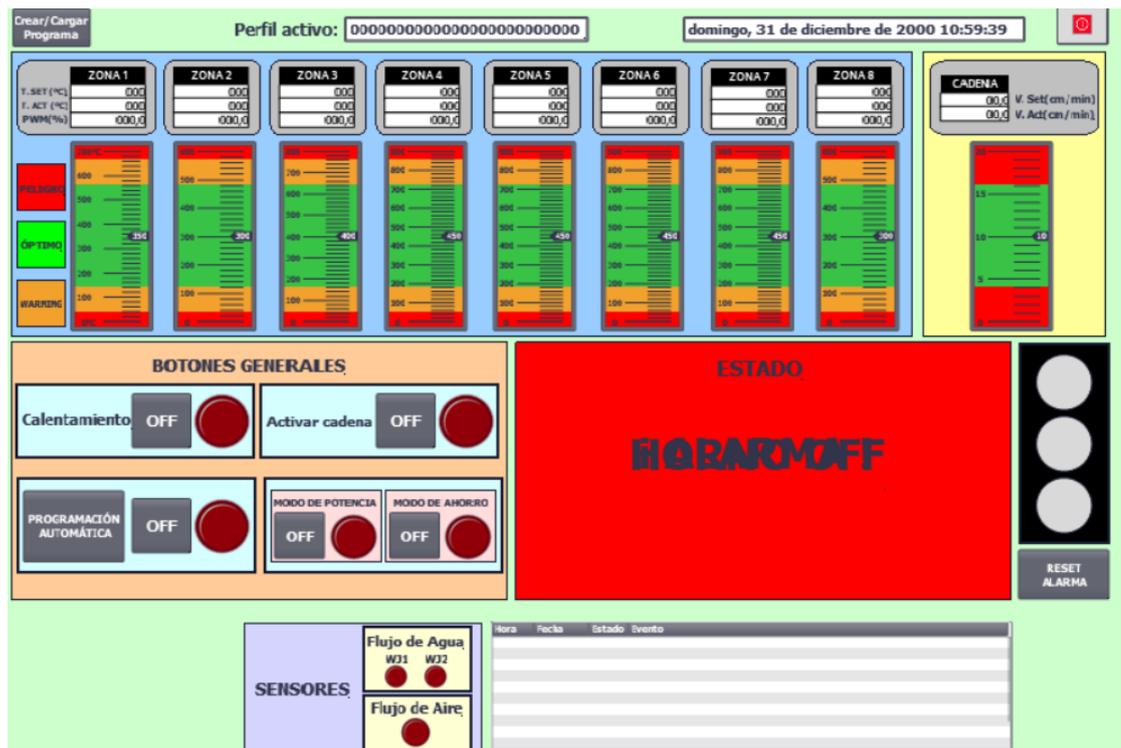


Ilustración 87. Imagen HMI - Pantalla principal

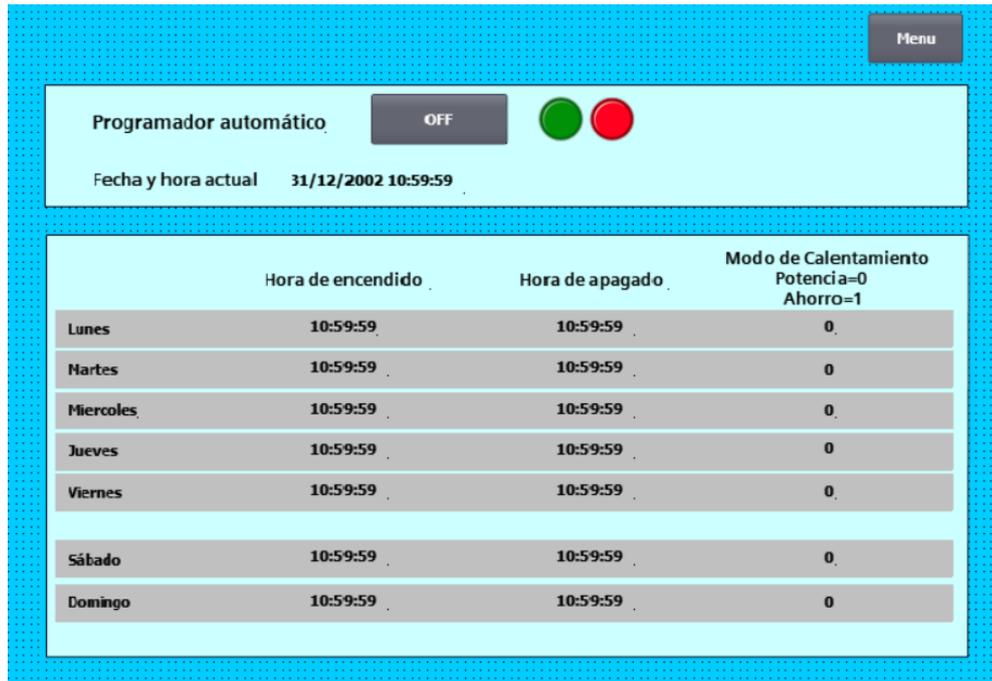


Ilustración 88. Imagen HMI - Programación automática.

Una vez creadas todas las imágenes que tendrá la interfaz, el último paso será programar la interfaz.

5.3. Configuración interfaz hombre-máquina

En este apartado vamos a detallar la configuración que se ha seguido para todos los botones, gráficas, recuadros, etc. También veremos cómo crear una “Receta”, para qué sirve, y por qué es necesaria en nuestro proyecto.

5.3.1. Configuración botones, interruptores y recuadros E/S

- **Botones:** Se van a utilizar para poder “viajar” entre imágenes, para abrir “imágenes emergentes” que explicaremos más adelante, o para cerrar la aplicación de Runtime.

Como vemos en la siguiente Ilustración, en el apartado propiedades del botón podemos cambiar el texto que muestra, su apariencia, tamaño, formato, etc.

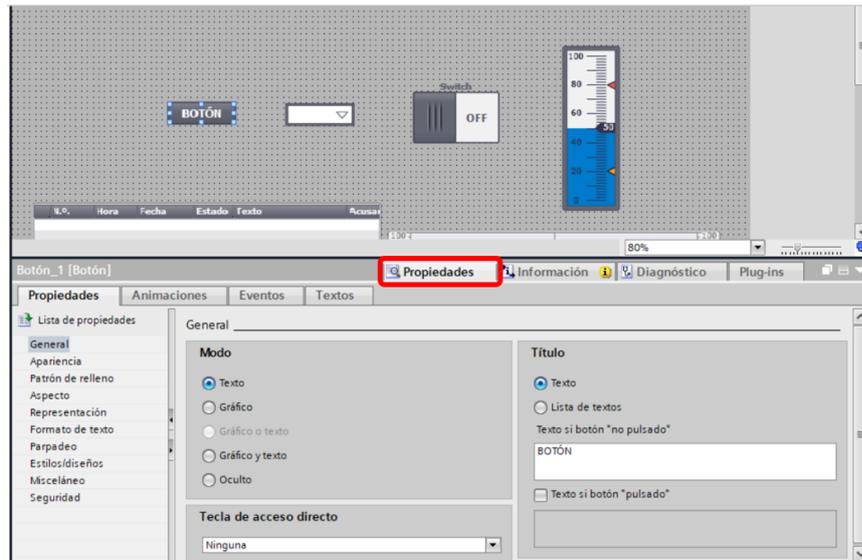


Ilustración 89. HMI - Propiedades Botón.

El principal uso que le vamos a dar a los botones lo encontramos en el apartado “Eventos”, donde podemos realizar una acción en función de lo que hagamos con el botón.

Por ejemplo, en nuestra interfaz tenemos configurado el botón de “Programación automática” de la siguiente manera:

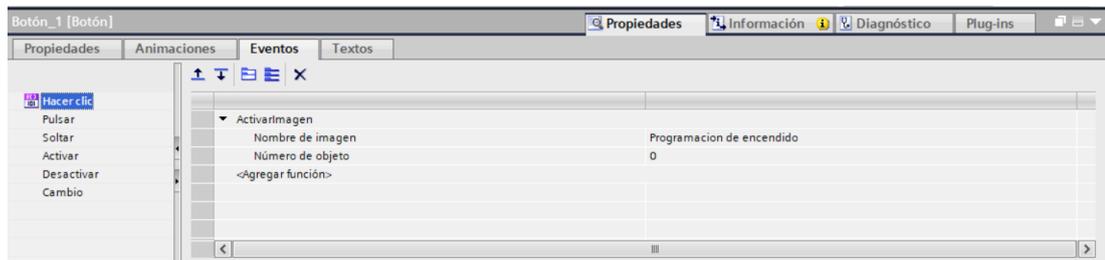


Ilustración 90. HMI - Eventos del botón de programación automática.

Indicamos que al “hacer clic” en el botón, se active una imagen, en este caso la imagen de “Programación de encendido”.

Tenemos infinidad de funciones que podemos añadir como vemos en la Ilustración [91]. Además, se pueden agregar varias funciones a la vez sin ningún problema.



Ilustración 91. HMI - Funciones para los eventos.

- **Interruptores:** Se van a utilizar para activar o desactivar variables.

Del mismo modo que con los botones, tenemos un apartado de propiedades con las mismas utilidades.

Lo fundamental para estos interruptores será vincularlos a una variable del PLC que activaremos o desactivaremos según el estado del interruptor. Como vemos en la Ilustración [92], tenemos el interruptor vinculado a la variable “Programación automática_ENCENDIDO_AUTOMATICO”.

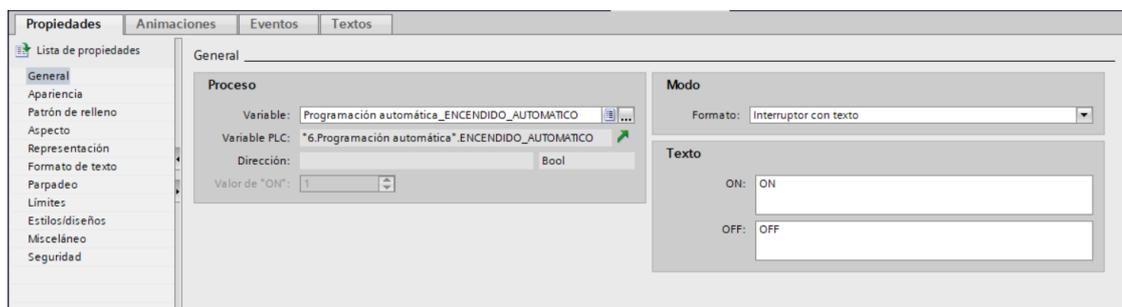


Ilustración 92. HMI - Vincular interruptor a variable PLC.

A su vez, la variable del PLC tiene que estar vinculada con la variable correspondiente dentro del HMI. Para ello debemos de crear estas variables antes de todo en “Variables HMI” o podemos ir las agregando poco a poco mientras configuramos los interruptores. Hay que recalcar la importancia de que la variable PLC y la variable HMI estén vinculadas correctamente, ya que si cogemos por ejemplo la variable PLC “Encendido_Horno” y la vinculamos con la variable HMI “Temperatura_Zona8”, no tendrá ningún sentido y dará lugar a errores.

- **Recuadros de entrada/salida:** Se van a utilizar para leer o mostrar datos.

5.3.2. Configuración elementos gráficos, figuras y cuadros de texto

- **Elementos gráficos:** Con estos queremos simular un termómetro en el que podamos ver en tiempo real la temperatura de cada zona, así como los valores de exceso o defecto de temperatura.

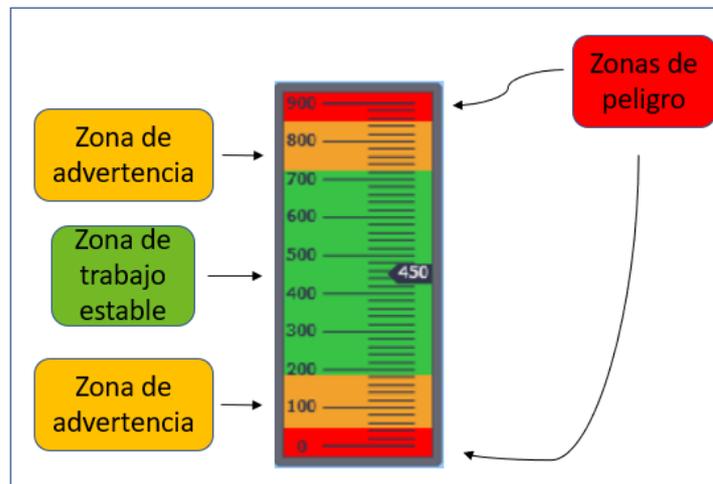


Ilustración 93. HMI - Termómetro con zonas definidas.

Para ello, escogeremos el elemento gráfico “Barra” y entraremos en sus propiedades, en donde seleccionaremos la variable que queremos visualizar, en este caso la temperatura de cada zona, y también definiremos sus límites como vemos en la siguiente Ilustración.

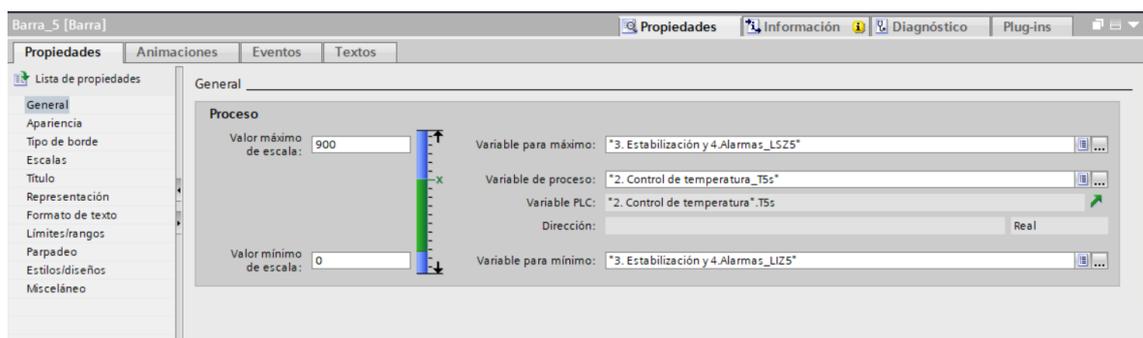


Ilustración 94. HMI - Variables definidas del elemento barra.

- **Figuras:** Se han utilizado para hacer la interfaz visualmente más estética y para delimitar zonas. También se hace uso del apartado “animaciones” como vemos en la siguiente Ilustración.

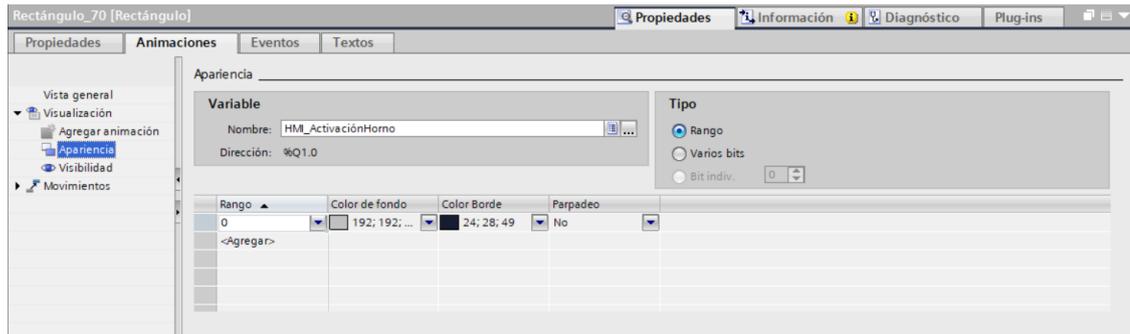


Ilustración 95. HMI - Apartado animaciones

Es una herramienta con la que podemos cambiar la apariencia, la visibilidad e incluso darle movimiento al objeto, que en este caso es un cuadrado gris que vemos a continuación.

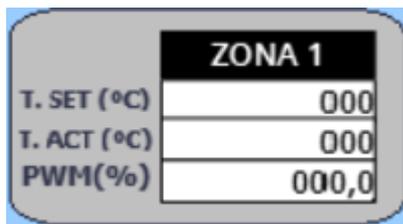


Ilustración 96. HMI - Estado Zona 1

La dirección de la empresa quería una interfaz con mucha información, y en especial quería que estos recuadros, además de indicar la temperatura que queremos, la temperatura actual y el valor del PWM, cambiaran de color en consonancia con el termómetro, es decir, si la temperatura está en el intervalo de advertencia, lo vemos rápidamente en el termómetro, pero también queremos ver que el recuadro este de color anaranjado.

Para ello le damos una apariencia en función de una variable. Cuando la temperatura sea adecuada para trabajar, el termómetro estará en la zona verde, y esto activará la variable "Z1_Verde". Al activarse, el recuadro cambiará al color verde, y al desactivarse al color gris.

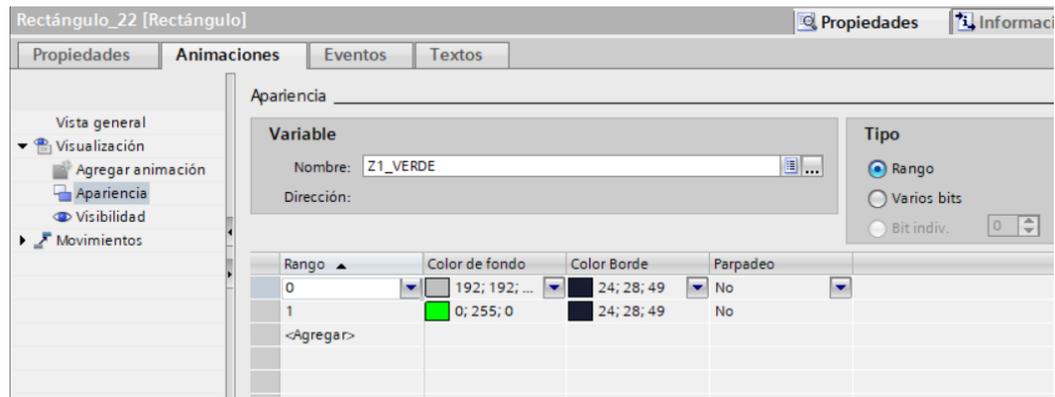


Ilustración 97. HMI - Apariencia rectángulo zona 1.

Del mismo modo para la zona naranja y zona roja, aunque para ello se han hecho copias de estos recuadros y superpuesto unos encima de otros, ya que cada recuadro está vinculado a una sola variable.

Finalmente, se añade visibilidad a cada recuadro para que solo se vea cuando necesitamos. Esto se debe a que, si nos encontramos en zona verde se tendrá que ver el recuadro verde, pero al superponer unos encima de otros este no se vería. Por lo que, cuando la zona verde este activa solo será visible dicho recuadro.

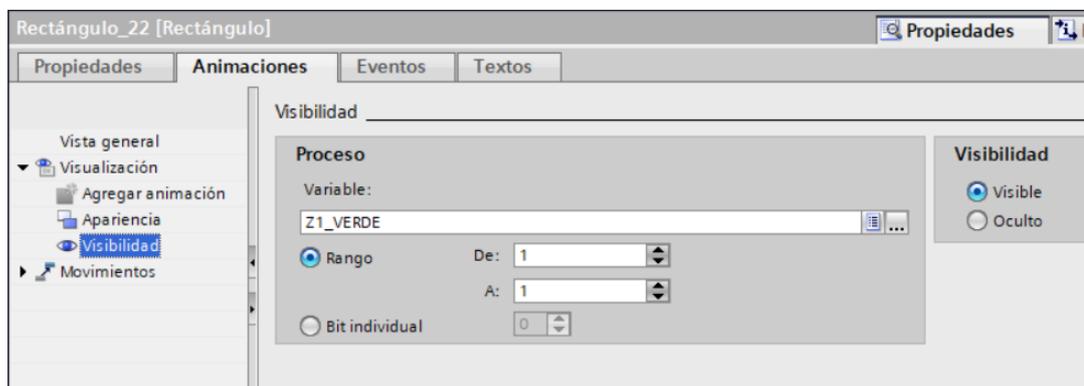


Ilustración 98. HMI - Visibilidad rectángulo zona 1

5.3.3. Creación de receta

Una receta es un conjunto de elementos, que tienen nombre, nombre de visualización, una variable asignada y un tipo de dato entre otras cosas. Con ello podremos definir las diferentes curvas de temperatura que el horno tenga que utilizar.

Recetas									
...	Nombre	Nombre de visualizac...	Número	Versión	Ruta	Tipo	Número máximo de re...	Tipo de comunicación	Tooltip
	Horno0717	Receta_1	1	26/08/2020 10:...	C:\Recipes	Limitado	5000	Variables	
<Agregar>									

Elementos									
...	Nombre	Nombre de visualizac...	Variable	Tipo de datos	Longitud d...	Valor predetermi...	Valor mínimo	Valor máximo	Decima
	Zona1	Zona1—...	*2. Control de te...	Real	4	0	-3,402823466E+38	3,402823466E+38	0
	Alarma Superior Z1	Alarma Superior Z1	*3. Estabilización...	Real	4	0	-3,402823466E+38	3,402823466E+38	0
	Advertencia Superior...	Advertencia Superior...	W5Z1	Real	4	0	-3,402823466E+38	3,402823466E+38	0
	Advertencia Inferior Z1	Advertencia Inferior Z1	WZ1	Real	4	0	-3,402823466E+38	3,402823466E+38	0
	Alarma Inferior Z1	Alarma Inferior Z1	*3. Estabilización...	Real	4	0	-3,402823466E+38	3,402823466E+38	0

Ilustración 99. Receta

En primer lugar, debemos crear una receta e ir añadiendo sus distintos elementos. Para nuestro caso, decidimos que los elementos serán cinco por cada zona y tres más para la cadena. “Zona 1” corresponde con la temperatura que queremos que alcance dicha zona. “Alarma Superior Z1” y “Alarma Inferior Z2” son las temperaturas que, si son superadas o no son alcanzadas respectivamente, saltarán las alarmas. “Advertencia Superior Z1” y “Advertencia Inferior Z2” son las temperaturas que marcarán la zona de advertencia, donde se podrá seguir trabajando con el horno, pero habrá que tener cuidado. En cuanto a la cadena, tenemos la velocidad que queremos que alcance la cadena, y sus límites superior e inferior.

Una vez tenemos todos los elementos definidos, en el apartado registro debemos ir asignando valores en función de la curva de temperatura que necesitemos. Con esto logramos que el operario en la pantalla HMI tenga un “menú” donde estén guardados todos los perfiles y se pueda elegir uno u otro en función de las necesidades.

Elementos								
Registros								
...	Nombre	Nombre de visualizac...	Número	Zona1	Alarma Superior Z1	Advertencia Superi...	Advertencia Inferio...	Alarma Inferior Z1
	Perfil 850	Perfil 850	1	520	540	530	510	500
<Agregar>								

Ilustración 100. Registro Perfil 850°C

Para finalizar, creamos una imagen emergente que contenga un visor de recetas. Esta imagen emergente estará vinculada a un botón de la pantalla principal.

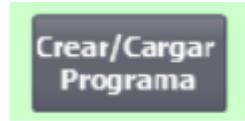


Ilustración 101. Botón Crear/Cargar Programa

El visor de recetas nos permitirá visualizar los perfiles de temperatura ya guardados, así como crear otros o modificarlos, y al igual que podemos cargar dichos parámetros dentro del PLC, también podremos descargar los parámetros que ya tenga el PLC en caso de perder el perfil de temperatura o por cualquier otro motivo.

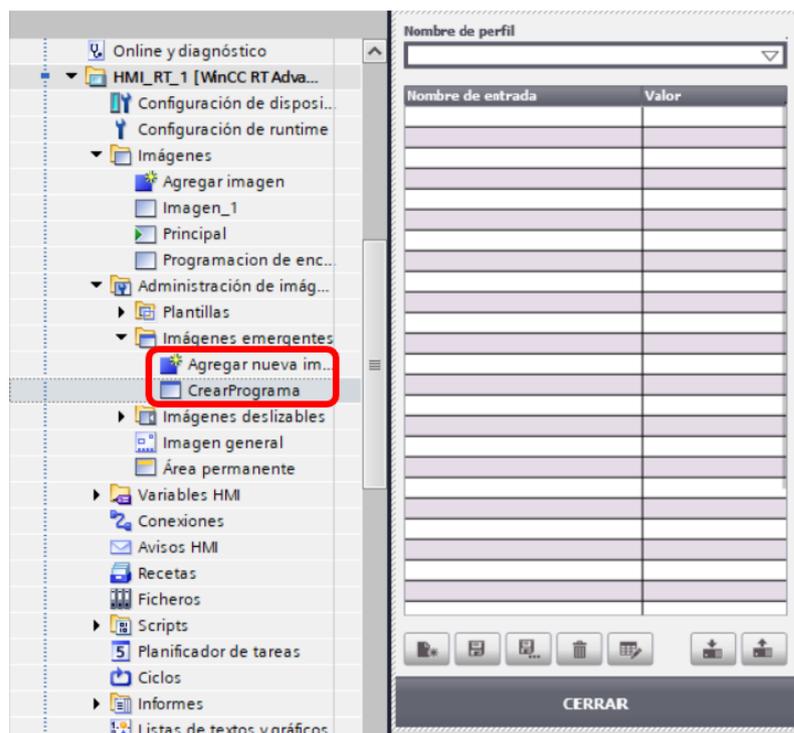


Ilustración 102. Visor de recetas

5.3.4 Visor de avisos

Los avisos son una parte fundamental de la interfaz HMI ya que nos proporcionan información adicional de lo que está ocurriendo en el proceso. Cualquier pulsación de botón, cambio de parámetros, advertencia o alarma quedará registrada en el sistema, tanto el día como la hora exacta.

Durante el proyecto, este visor de avisos fue muy útil. Por ejemplo, el horno debía quedar encendido durante todo el día, y al llegar al día siguiente el horno estaba apagado. Esto se debía a que por las noches la bomba de agua se desactivaba y no llegaba agua para la refrigeración del horno, de modo que este se apagaba automáticamente como había quedado programado. Gracias al visor de avisos, se descubrió que el fallo era la falta de agua y gracias a la hora se supo que era el corte automático de las bombas de agua a una determinada hora de la noche.

Para la creación del visor de avisos debemos añadir el objeto “Visor de avisos” en cualquier imagen del HMI, en nuestro caso, en la pantalla principal. Luego debemos declarar todos los avisos como vemos en la siguiente Ilustración.

ID	Nombre	Texto de aviso	Categoría	Variable de di...	Bit de ...	Dirección de ...	Variable de ac...	Bit de ...	Dirección de acuse HMI	Informe
1	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z1	Errors	ALARMAS 1	0	%DB12.DBX1.0	ACUSE 1	0	%DB12.DBX9.0	<input type="checkbox"/>
2	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z2	Errors	ALARMAS 1	1	%DB12.DBX1.1	ACUSE 1	1	%DB12.DBX9.1	<input type="checkbox"/>
3	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z3	Errors	ALARMAS 1	2	%DB12.DBX1.2	ACUSE 1	2	%DB12.DBX9.2	<input type="checkbox"/>
4	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z4	Errors	ALARMAS 1	3	%DB12.DBX1.3	ACUSE 1	3	%DB12.DBX9.3	<input type="checkbox"/>
5	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z5	Errors	ALARMAS 1	4	%DB12.DBX1.4	ACUSE 1	4	%DB12.DBX9.4	<input type="checkbox"/>
6	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z6	Errors	ALARMAS 1	5	%DB12.DBX1.5	ACUSE 1	5	%DB12.DBX9.5	<input type="checkbox"/>
7	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z7	Errors	ALARMAS 1	6	%DB12.DBX1.6	ACUSE 1	6	%DB12.DBX9.6	<input type="checkbox"/>
8	ERROR: TEMPERATU...	ERROR: TEMPERATURA ALTA EN Z8	Errors	ALARMAS 1	7	%DB12.DBX1.7	ACUSE 1	7	%DB12.DBX9.7	<input type="checkbox"/>
9	ERROR: TEMPERATU...	ERROR: TEMPERATURA BAJA EN Z1	Errors	ALARMAS 2	0	%DB12.DBX3.0	ACUSE 2	0	%DB12.DBX11.0	<input type="checkbox"/>
10	ERROR: TEMPERATU...	ERROR: TEMPERATURA BAJA EN Z2	Errors	ALARMAS 2	1	%DB12.DBX3.1	ACUSE 2	1	%DB12.DBX11.1	<input type="checkbox"/>

Ilustración 103. Avisos de bit

Es muy importante elegir bien tanto la variable de disparo como la dirección de disparo. En esa dirección estará la información necesaria para mostrar o no el aviso. También es importante, en el caso de necesitarlo, escoger bien la variable de acuse y su dirección. Esto servirá para resetear dicho aviso o programar otro tipo de acción.

6. Manual de usuario

6.1. Introducción

El siguiente manual de usuario pertenece a la última actualización de hardware y software del horno industrial de la marca 'THERMATECH', perteneciente a la empresa 'MAHLE Electronics' con ubicación en el Parque Tecnológico de Boecillo (Valladolid).

Este horno industrial trabaja principalmente con placas electrónicas, en concreto con biosensores y aforadores. Tiene como objetivo seguir un perfil de temperaturas con forma de campana con un pico máximo de 850 °C. A continuación, podemos ver un ejemplo de dicho perfil.

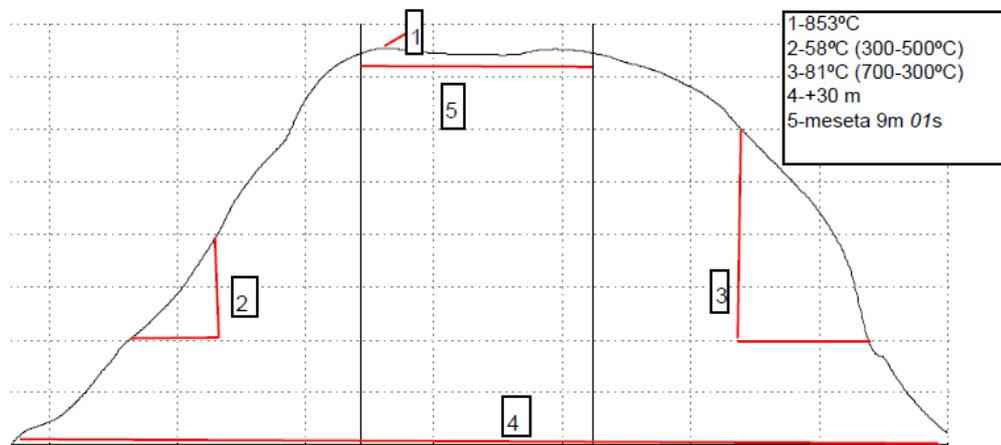


Ilustración 104. Manual de usuario. Campana 850 °C

El horno trabaja con un autómata programable de la marca 'SIEMENS', en concreto con el S7-1200 con CPU 1214C DC/DC/DC. Se ha realizado una programación completa desde cero junto con una interfaz hombre-máquina desde la que realizar todas las operaciones.

En los siguientes apartados podremos ver detalladamente el funcionamiento de este horno.

6.2. Encendido del horno

El objetivo de este apartado es explicar con detalle cómo encender correctamente el horno en sus dos modos de funcionamiento.

a. Manual.

El operario en ese mismo instante trata de encender el horno a través de los botones de la interfaz HMI. Para encender el horno manualmente previamente el operario debe de tener en cuenta una serie de pasos a seguir.

- **Paso N.º 1: Verificar los sensores de agua y aire.**

El operario debe acudir a verificar que tanto la válvula de aire, como la llave de paso del agua se encuentren abiertas. Esto es necesario para que el horno tenga una correcta refrigeración.

Para una mayor seguridad, comprobar nuevamente el estado de estos elementos, pero en la pantalla HMI. Para que el horno pueda encenderse deben de estar todos los sensores activados.



Ilustración 105. Manual de usuario. Sensores

En caso de abrir la válvula de aire y la llave de paso del agua, y no contar con agua y/o aire, debemos acudir a la sala de mantenimiento situada en la planta baja. Si no tenemos aire en el circuito, el problema será del compresor de aire que no estará en funcionamiento o ha tenido algún otro problema con el que deberíamos de llamar a un supervisor.

Por otro lado, si no tenemos agua en el circuito debemos de acudir a verificar la bomba de agua. Esta puede encontrarse apagada, ya que está programada para que se apague a determinadas horas o puede que tenga algún error. También puede ser que la bomba esté operativa pero las llaves de paso de la planta baja estén cerradas.

- **Paso N.º 2: Cargar programa y activar cadena correctamente.**

Para proceder a cargar un programa debemos acudir al botón superior izquierda de 'Crear/Cargar Programa' como podemos ver a continuación:

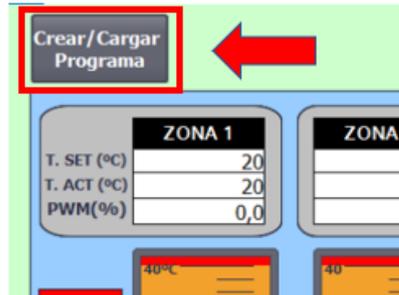


Ilustración 106. Manual de usuario. Botón Crear/Cargar Programa

Tras hacer clic en él, nos aparecerá una ventana con los tipos de perfiles que queremos cargar y con sus diferentes parámetros. Estos se podrán modificar bajo la supervisión de un superior.

Nombre de perfil	
DATA_5	
Nombre de entrada	Valor
-----Zona1-----	20
Alarma Superior Z1	40
Advertencia Superior Z1	19
Advertencia Inferior Z1	18
Alarma Inferior Z1	17
-----Zona2-----	20
Alarma Superior Z2	40
Advertencia Superior Z2	20
Advertencia Inferior Z2	18
Alarma Inferior Z2	17
-----Zona3-----	20
Alarma Superior Z3	40
Advertencia Superior Z3	20
Advertencia Inferior Z3	18
Alarma Inferior Z3	17
-----Zona4-----	20
Alarma Superior Z4	40
Advertencia Superior Z4	20
Advertencia Inferior Z4	18
Alarma Inferior Z4	15
-----Zona5-----	20
Alarma Superior Z5	40
Advertencia Superior Z5	30
Advertencia Inferior Z5	18
Alarma Inferior Z5	15
-----Zona6-----	20
Alarma Superior Z6	40

CERRAR

Ilustración 107. Manual de usuario. Ventana de perfiles

La ventana anterior tiene una serie de botones en la parte inferior cuyas funciones son, de izquierda a derecha, “Nuevo perfil”, “Guardar”, “Guardar como...”, “Borrar”, “Cambiar nombre”, “Cargar al dispositivo” y “Cargar del dispositivo”.

Cuando cargamos al dispositivo, introducimos al PLC los parámetros que nosotros hayamos elegido, y cuando cargamos del dispositivo, cargamos en nuestra ventana los parámetros que ya teníamos guardados en el PLC.

Finalmente, tras comprobar que los parámetros de temperatura y parámetros de velocidad de la cadena son correctos, hacemos clic en el botón de “Activar Cadena”. Una vez esta activada, este botón conmuta a “ON” y el indicador de al lado estará de color verde.



Ilustración 108. Manual de usuario. Activar cadena

- **Paso N.º 3: Calentamiento del horno.**

Para dar comienzo al calentamiento del horno, debemos hacer clic en “Calentamiento”. Del mismo modo que para la cadena, el botón conmutará a “ON” y el indicador pasará a color verde.



Ilustración 109. Manual de usuario. Botón de calentamiento.

Según realizamos este procedimiento, escucharemos como se activa el contactor principal, lo que indica que hay tensión para proceder a calentar. En el caso de no escuchar dicho contactor, no habrá tensión porque tendrá lugar un error, ya sea de los sensores o cualquier otro componente. En este caso debemos de acudir a un supervisor.

Tras iniciar el calentamiento siempre nos encontraremos en modo “Setup”, lo que quiere decir que el horno está calentando, pero todavía no está listo para que se le introduzcan placas. Esto lo podemos ver en la Ilustración [110].

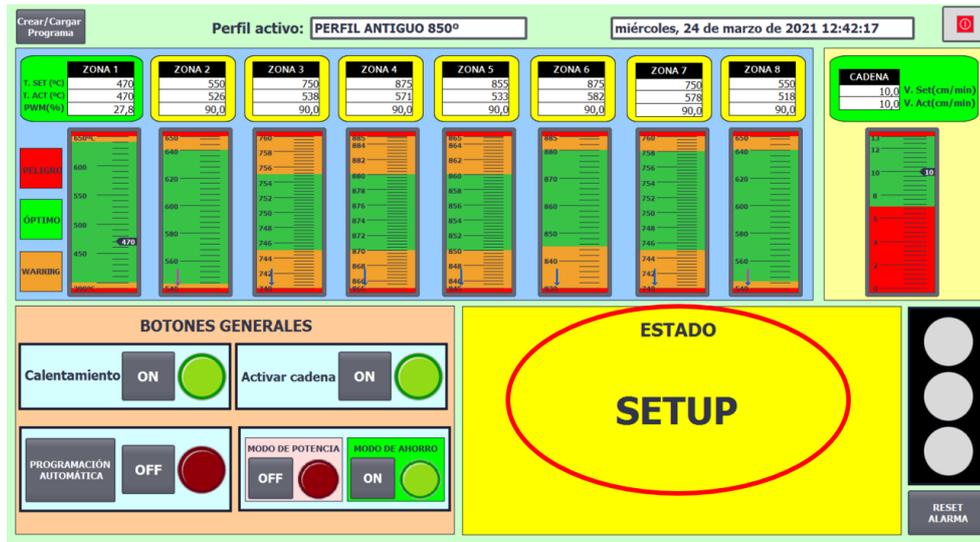


Ilustración 110. Manual de usuario. Horno en modo setup

- **Paso N.º 4: Setup y horno estable.**

El horno se encuentra calentando y todavía no ha entrado en la zona estable. Los indicadores de cada zona permanecerán en amarillo mientras calientan hasta que entren en su zona estable previamente configurada. Una vez van alcanzando las zonas la estabilidad, dicho recuadro pasará a verde, indicando que esa zona está lista para trabajar.

El horno en su totalidad estará en zona estable cuando todas las zonas del horno, es decir las ocho zonas de calentamiento del horno estén estabilizadas al mismo tiempo. Si por ejemplo tenemos siete zonas estables y una que todavía no ha entrado, no podemos comenzar a trabajar con el horno ya que no seguiríamos correctamente el perfil de temperaturas.

Una vez el horno este en zona estable, lo visualizaremos en pantalla como podemos ver en la siguiente Ilustración.



Ilustración 111. Manual de usuario. Horno estable

De aquí en adelante solo se pueden barajar tres opciones:

1. Horno estable.
2. Warning de una o varias zonas.
3. Alarma de una o varias zonas.

b. Automático.

Se trata de un encendido programado para una determinada hora, con el objetivo de que el horno comience a calentar en horario no laboral, para que al comienzo de la jornada el horno esté disponible para trabajar. Es el encendido más habitual. Lo que debemos de tener en cuenta es que a la hora que tengamos programado el encendido, debemos tener habilitada la entrada de agua y de aire. También debemos tener correctamente cargados los parámetros.

En el caso de no disponer de agua y/o aire en el momento de encendido, el horno no comenzará a calentar y tendremos que hacerlo manualmente siguiendo todos los pasos.

Para trabajar con el modo automático debemos de hacer clic en 'Programación automática' y configurar tanto los horarios de encendido y de apagado, como el modo de calentamiento con el que comenzará el horno.

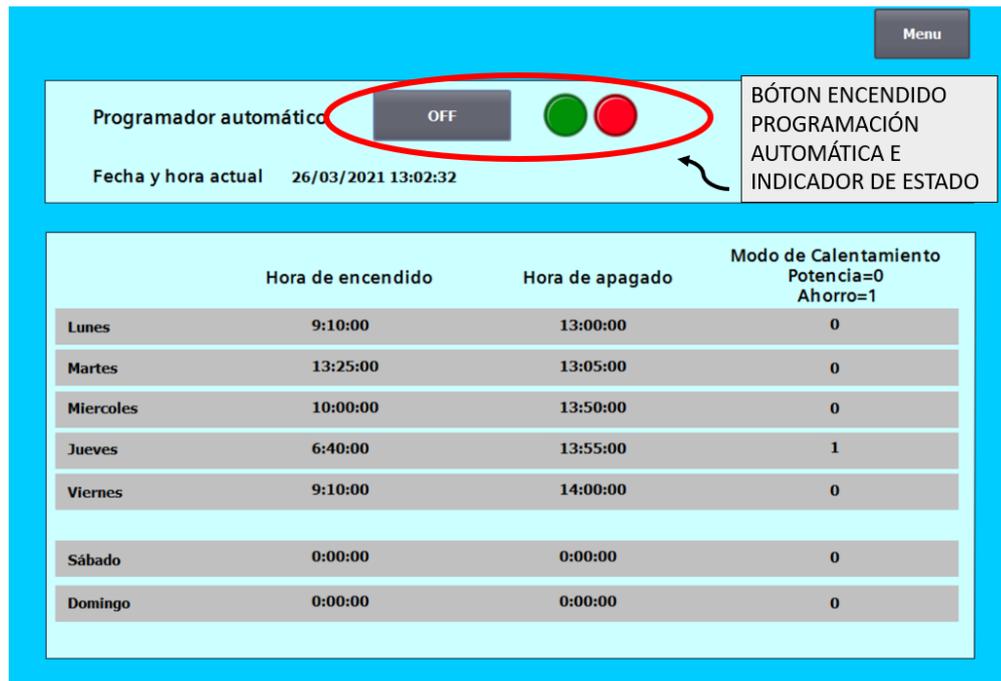


Ilustración 112. Manual de usuario. Programación automática

6.3. Estados del horno

En este apartado vamos a distinguir los tres estados en los que podemos encontrar el horno.

a. Estable.

Estado en el que podemos comenzar a trabajar con el horno y no necesita ningún tipo de atención. Para que pueda existir algún warning o alarma, previamente el horno tiene que encontrarse estable. En caso de no haber alcanzado esa zona de estabilidad, no encontraremos warnings o alarmas de temperatura salvo que se exceda los límites superiores de alarma preestablecidos.

b. Warning.

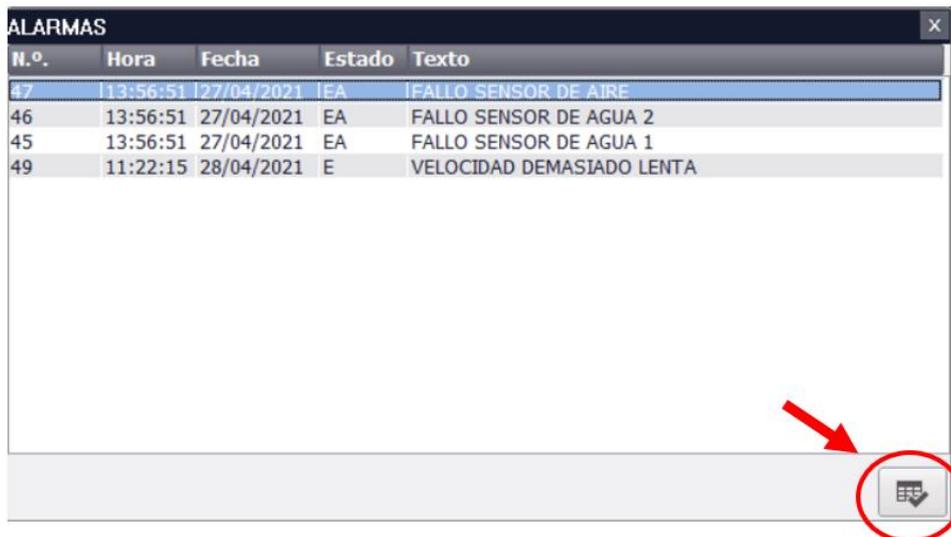
Una o varias zonas del horno han entrado en la zona naranja, en la zona warning. Se puede seguir trabajando con el horno exactamente igual ya que la variación del perfil de temperaturas es mínima y la calidad del producto final no se ve afectada. Lo único a tener en cuenta es que se debe de estar pendientes de estos cambios ya que es más probable que se excedan las temperaturas.

c. Alarma.

El horno ha detectado que una o varias zonas han excedido los límites, ya sea superior o inferior. Se puede seguir trabajando en un breve periodo de tiempo siempre y cuando se solucione rápidamente dicha alarma y no haya sido excesiva. En caso de una temperatura excesiva, o un error crítico, durante 20 minutos o más, el horno dejará de calentar automáticamente. La cadena continuará en movimiento para poder salvar la mayor cantidad de material posible.

En caso de alarma debemos de acusar todos los errores en el visor de alarmas. Si por error hemos cerrado el visor de alarmas, pulsaremos “RESET ALARMAS”.

Tras acusar errores, o hacer clic en este botón, la bocina dejará de sonar, pero la columna de señalización seguirá en color rojo hasta solucionar dicho fallo.



N.º	Hora	Fecha	Estado	Texto
47	13:56:51	27/04/2021	EA	FALLO SENSOR DE AIRE
46	13:56:51	27/04/2021	EA	FALLO SENSOR DE AGUA 2
45	13:56:51	27/04/2021	EA	FALLO SENSOR DE AGUA 1
49	11:22:15	28/04/2021	E	VELOCIDAD DEMASIADO LENTA

Ilustración 113. Manual de usuario. Acuse de alarmas

En el estado de la alarma podemos encontrar:

- E: Significa que ha aparecido el error.
- EA: Aparecido y acusado.
- S: Desaparecido.

6.4. Posibles errores y soluciones

En este apartado se detallan tres de los errores menos frecuentes y cómo solucionarlos.

- **Error de comunicación con el horno.**

Puede ocurrir por una desconexión del PLC, de la red eléctrica o por diversos factores más. Lo identificamos cuando queremos poner en marcha la cadena y no responde a pesar de haber introducido correctamente los parámetros y haber seguido todos los pasos.

Para comprobarlo podemos ver dentro del horno el estado del PLC. Lo encontraremos en estado "ERROR" como podemos ver a continuación.



Ilustración 114. Manual de usuario. Error del PLC

Para solucionar el error debemos de quitar la tensión al contactor principal y ponerla nuevamente. Esperaremos un minuto para que se conecten PLC y PC.

- **Error al cargar parámetros.**

Puede ocurrir accidentalmente al introducir cualquier parámetro y que los valores dados no tengan sentido, por ejemplo:

- Advertencia Superior: 100°C
- Advertencia Inferior: 120°C

Como el primer parámetro se entiende que debe de ser de un valor mayor que el segundo, visualizaremos la barra de zona de la siguiente manera.



Ilustración 115. Manual de usuario. Error con carga de parámetros

El orden correcto de los parámetros, de mayor a menor temperatura sería el siguiente

Alarma Superior > Advertencia Superior > Zona > Advertencia Inferior > Alarma Inferior

En el momento que observemos algo similar a la Ilustración [115], sabemos que se trata de este fallo y lo solucionaremos cargando correctamente los parámetros.

- **Alarma tras cargar nuevos parámetros.**

Hay veces que el horno está trabajando en la zona estable y necesitamos cambiar algún parámetro. Puede ocurrir que salte una alarma en determinados casos, por ejemplo: si tenemos en la zona 1 una temperatura actual de 500°C, y cambiamos el parámetro de alarma inferior, y este es superior a la temperatura actual, el PLC lo detectará como un error de temperatura baja, por lo que saltará la alarma.

Para solucionarlo simplemente acusamos el error cuando nos aparezca en el visor de avisos o haciendo clic en 'RESET ALARMA'.

7. Conclusiones

El objetivo de este proyecto principalmente era dejar operativo el horno industrial para su uso a diario además de la creación de una pantalla HMI para el control de este.

El estudio y el trabajo a diario de los equipos y componentes que forman el horno me ha ayudado a una búsqueda más eficiente de documentación, y también me ha aportado una experiencia mucho más real y cercana que la adquirida durante mi formación. Esto se debe al hecho de tener un acceso total a la instalación, donde poder comprobar cableado, señales, observar y analizar detenidamente el funcionamiento y el comportamiento de los equipos, regular válvulas, cambio y/o reparación de componentes, etc.

En cuanto a la programación, se han conseguido con éxito todos los puntos propuestos por la empresa. Se ha conseguido un producto final de mayor calidad debido al seguimiento con mayor exactitud de los perfiles de temperatura. Esto se ha logrado gracias al nuevo autómatas, al control PID, a la regulación de la cadena, etc. También se ha probado y logrado una correcta seguridad del horno a la hora de trabajar.

En la parte de programación he ganado fluidez a la hora de manejar el software, así como agilidad mental para pensar el proceso y estructurar los pasos de forma correcta.

Con el desarrollo de la interfaz HMI, hemos logrado una pantalla de gran utilidad y muy intuitiva, algo muy importante dentro de la industria y de los procesos para poder tener una monitorización y un mejor control de datos, entradas, salidas, etc.

Acerca de interfaz HMI con TIA PORTAL no tenía nada de conocimiento y por ello es la parte que más me ha gustado realizar. Se han invertido muchas horas en leer documentación y ver cursos acerca de HMI, por lo que es el conocimiento más grande y enriquecedor que me llevo de este proyecto.

Tras la consecución de estos objetivos con éxito, me he dado cuenta de que, durante mi formación en la universidad, he adquirido una serie de conocimientos muy útiles acerca de autómatas y del software TIA PORTAL que me han ayudado a llevar este proyecto de una manera más sencilla, pero que aun así, la realidad es otra. La posibilidad de poder trabajar con una instalación real en solitario me ha hecho profundizar más en el software de Siemens, en donde mi conocimiento y mi capacidad de autoaprendizaje se ha elevado a otro nivel. También he podido conseguir una mayor confianza en mí mismo, además de experimentar reuniones de empresa en donde la comunicación juega un papel fundamental.

En cuanto al proyecto en general, se pueden plantear una serie de líneas futuras. Se puede trabajar en una nueva versión de la interfaz HMI para monitorizar más datos, así como tener más control sobre el horno. Para ello se pueden añadir otras válvulas, sensores de presión del aire, añadir otros módulos al autómeta, balizas de señalización, etc. Además, se puede añadir esta nueva interfaz trabajándola sobre una pantalla física HMI de Siemens, donde realizar todo su cableado y conexión al autómeta.

Otra línea futura también puede ser llevar a cabo la automatización del proceso de carga del horno, en donde las placas llegarían a través de una cinta procedentes de un cargador o molde, y posteriormente mediante un sistema de ventosas, coger las placas electrónicas y depositarlas en la cadena del horno.

A nivel personal, siendo la automatización algo que me apasiona, tras realizar este proyecto he seguido adquiriendo conocimientos mediante cursos acerca del software de TIA Portal y me he dado cuenta de la importancia de la estructuración y la claridad que ha de tener el código mientras uno está programando. Era algo que ya sabía, pero no lo he aplicado de la mejor forma posible a este proyecto, ya que es un trabajo muy eficiente y funciona correctamente, pero en el futuro si otro ingeniero trata de continuar con la programación, inicialmente le costará entender la estructura y el funcionamiento de cada apartado. Es por ello que me llevo conmigo una serie de pautas a seguir para proyectos futuros, en las que haré uso de carpetas, FBs, FCs, crear variables, etc. con el fin de que sea un código mucho más limpio y legible. Todo el proyecto ha supuesto un gran reto para mí y a su vez, una experiencia muy gratificante, donde he conseguido llevar a cabo todos los objetivos impuestos por la empresa, quedando está muy satisfecha. Ha sido lo más parecido a lo que yo me espero de trabajar como automatista, en donde se tiene que realizar la programación y la puesta en marcha de un proceso, en este caso la puesta en marcha de un horno industrial. El cliente sería la empresa MAHLE Electronics y yo sería el ingeniero encargado de toda la programación y puesta a punto.

Como conclusión final, gracias a todas las horas de esfuerzo y dedicación que hay detrás de este trabajo, he logrado afianzar algunos de los conocimientos adquiridos durante la carrera, y aumentado muchos otros, ya que he tenido la oportunidad de aplicarlos en un caso práctico que podría encontrarme a lo largo de mi vida laboral.

8. Bibliografía

<https://docs.rs-online.com/4ed5/0900766b81397276.pdf> (Último acceso 5/03/2021)

https://www.tecnical.cat/PDF/SIEMENS/MECATRONICA/manual_SINAMICS_G120C.pdf (Último acceso 23/03/2021)

<https://masvoltaje.com/simatic-s7-1200/1207-simatic-s7-1200-modulo-de-entradas-digitales-sm-1221-16-ed-24v-dc-6940408100404.html> (Último acceso 15/02/2021)

https://bookdown.org/alberto_brunete/intro_automatica/lista-de-instrucciones.html (Último acceso 26/05/2021)

https://cache.industry.siemens.com/dl/files/036/108210036/att_916495/v1/s71500_pid_control_function_manual_esES_es-ES.pdf (Último acceso 24/03/2021)

https://cache.industry.siemens.com/dl/files/727/109485727/att_926980/v1/109485727_G120_CU240E2PN_at_S7_1200_SINA_SPEED_v10_DOCU_en.pdf (Último acceso 16/03/2021)

<https://media.automation24.com/datasheet/es/6ES72221BF320XB0.pdf> (Último acceso 8/02/2021)

https://media.automation24.com/manual/es/91696622_s71200_system_manual_es-ES_es-ES.pdf (Último acceso 11/02/2021)

https://cache.industry.siemens.com/dl/files/840/91379840/att_110005/v1/WinCC_Basic_V13_esES_es-ES.pdf (Último acceso 5/05/2021)

Anexos

Anexo 1: Bloques de programa

En este anexo se encuentran todos los bloques de programa del proyecto, así como todos los segmentos y comentarios adicionales.

Por bloques de programa entendemos todos los bloques OB y DB utilizados.

Totally Integrated Automation Portal		
--------------------------------------	--	--

HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

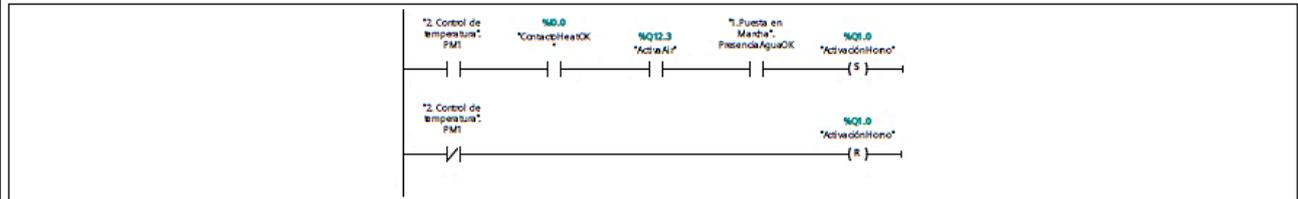
1. Puesta en Marcha [OB126]

1. Puesta en Marcha Propiedades							
General							
Nombre	1. Puesta en Marcha	Número	126	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

1. Puesta en Marcha			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

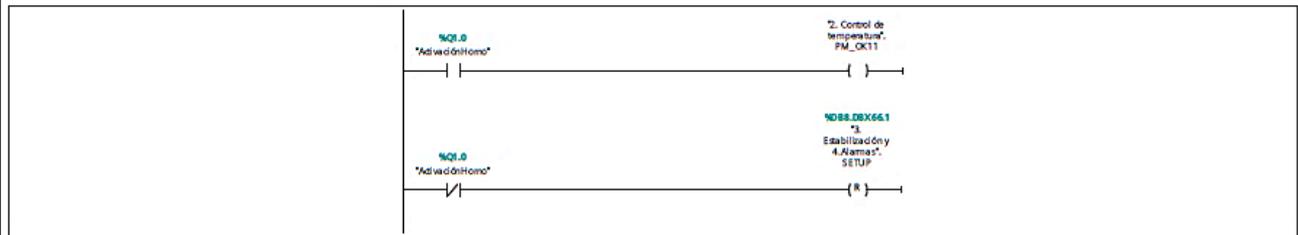
Segmento 1: Activación y desactivación de los contactos

Activación del contactor principal

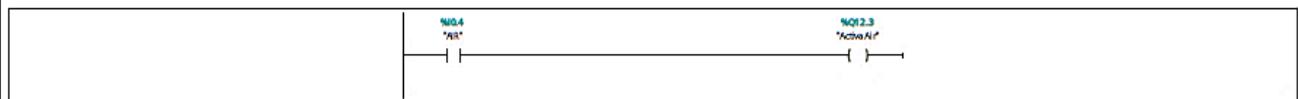


Segmento 2: Activación PM_OK11

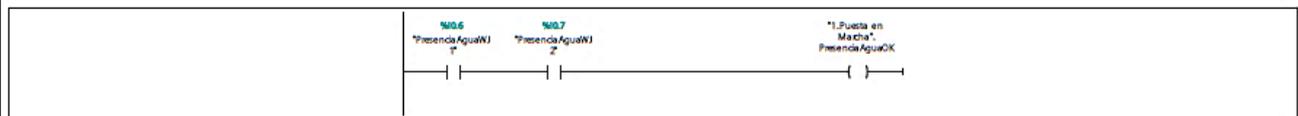
Señalización de que el horno está preparado para calentar.



Segmento 3: Activar Aire



Segmento 4: Presencia Agua



Totally Integrated Automation Portal		
--------------------------------------	--	--

HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

2. Control temperatura V1 [OB32]

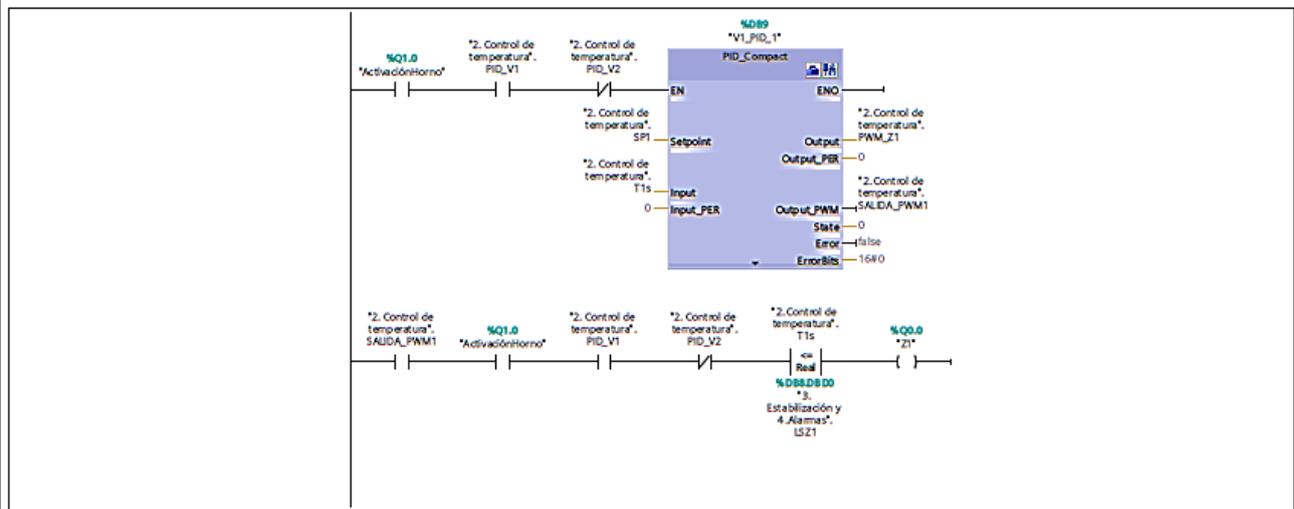
2. Control temperatura V1 Propiedades

General							
Nombre	2. Control temperatura V1	Número	32	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

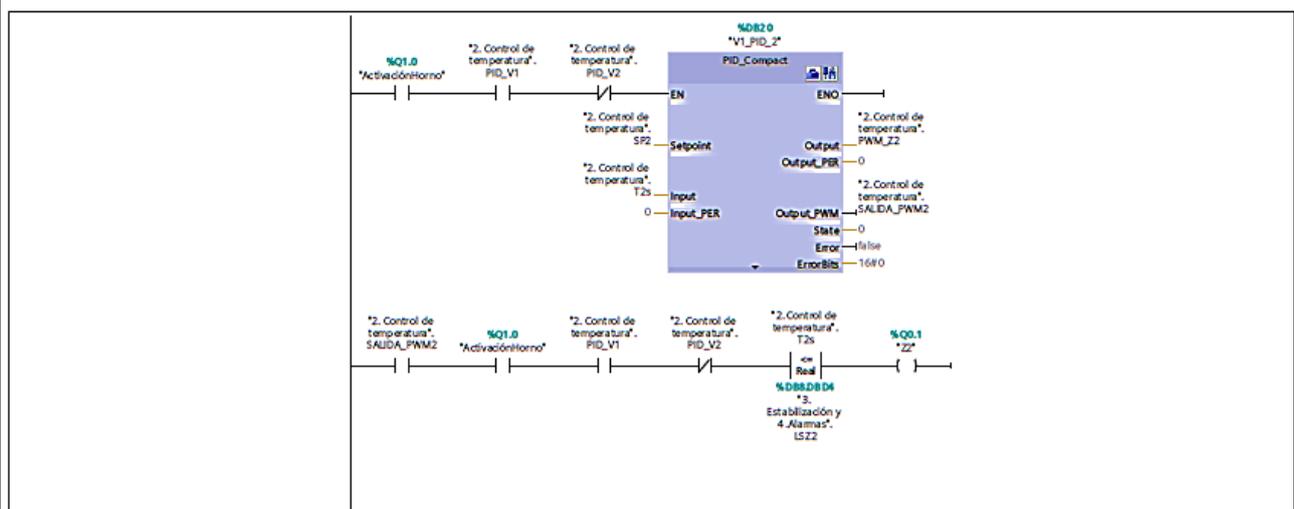
2. Control temperatura V1

Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
Temp			
Constant			

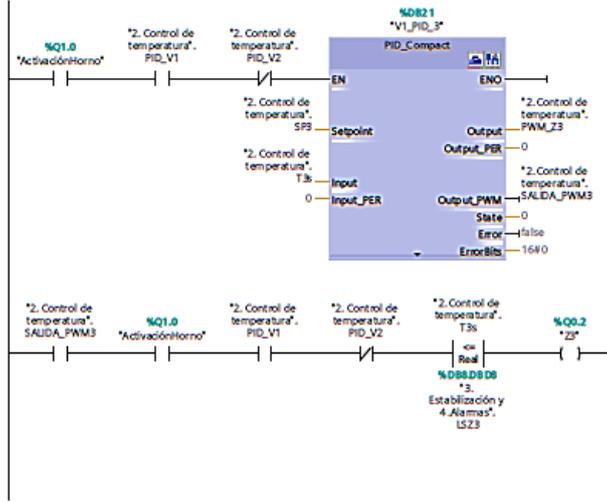
Segmento 1: PWM ZONA 1



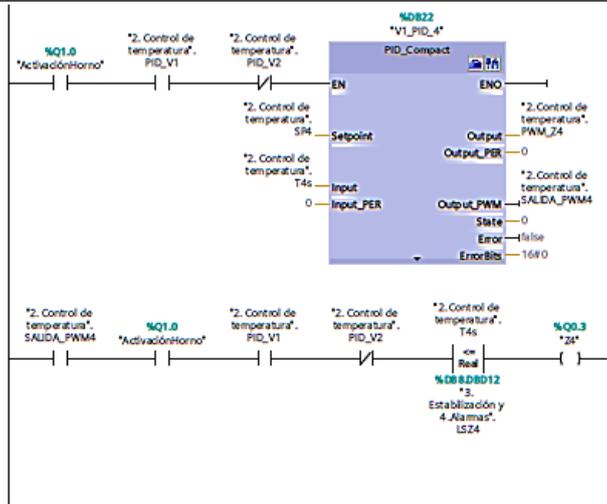
Segmento 2: PWM ZONA 2



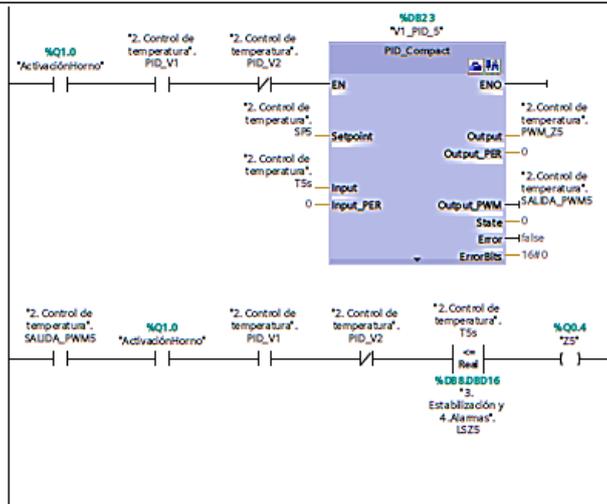
Segmento 3: PWM ZONA 3



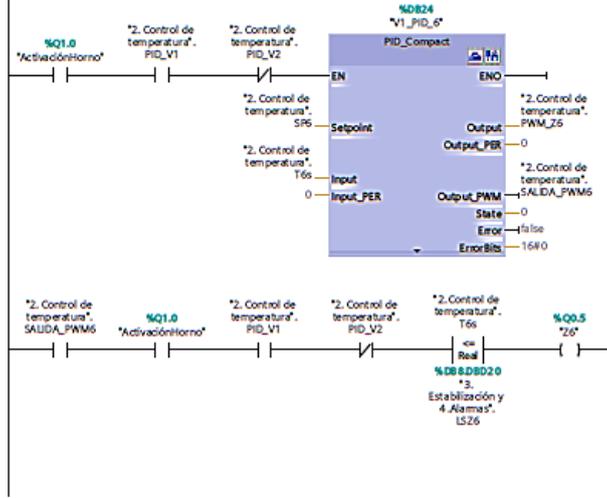
Segmento 4: PWM ZONA 4



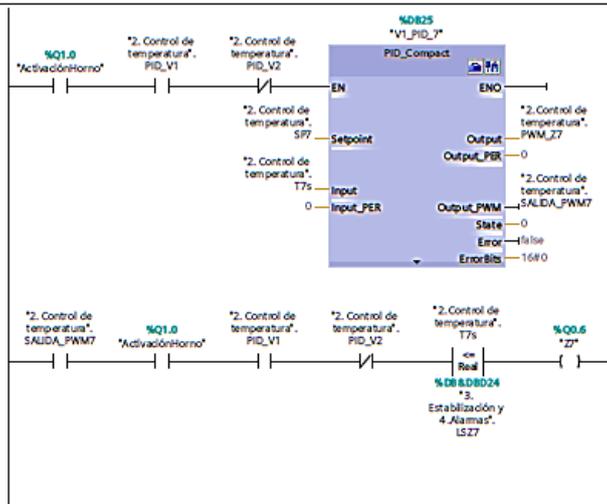
Segmento 5: PWM ZONA 5



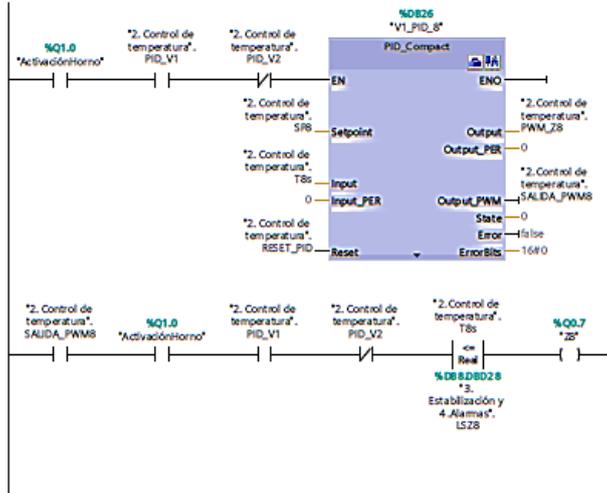
Segmento 6: PWM ZONA 6



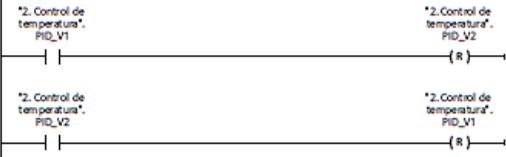
Segmento 7: PWM ZONA 7



Segmento 8: PWM ZONA 8



Segmento 9: BLOQUEO PID

Totally Integrated Automation Portal		
		

Totally Integrated Automation Portal		
--------------------------------------	--	--

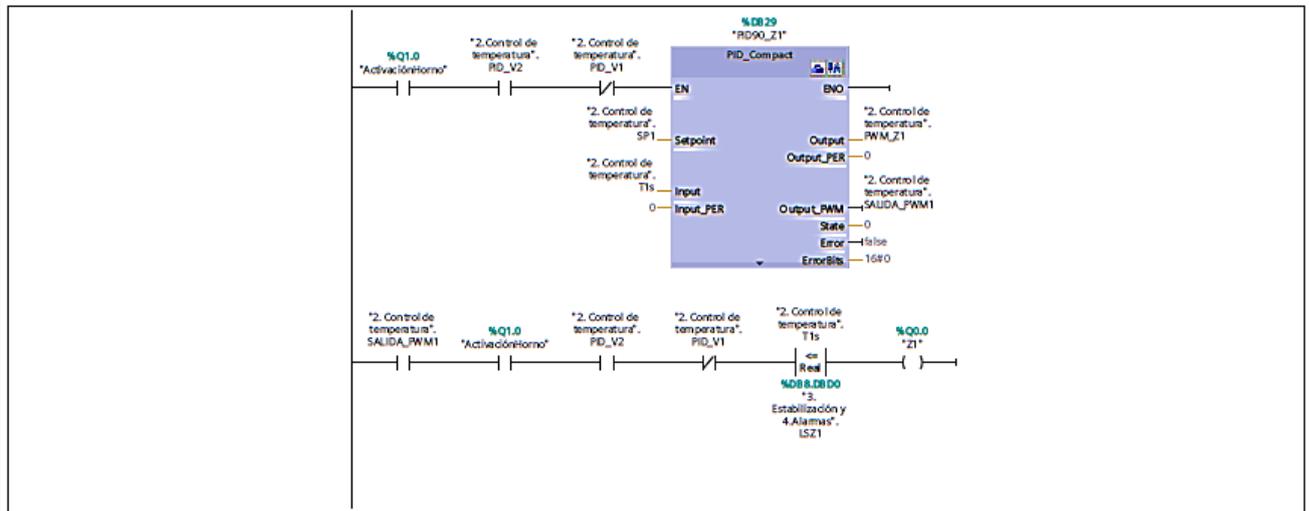
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

2.Control temperatura V2 [OB30]

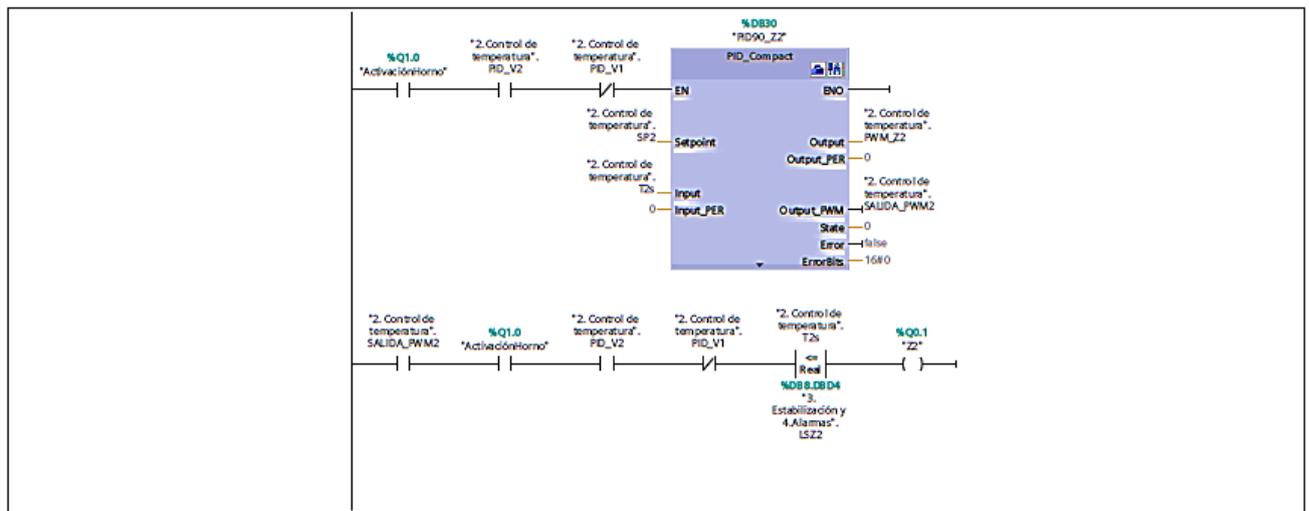
2.Control temperatura V2 Propiedades							
General							
Nombre	2.Control temperatura V2	Número	30	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

2.Control temperatura V2			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
Temp			
Constant			

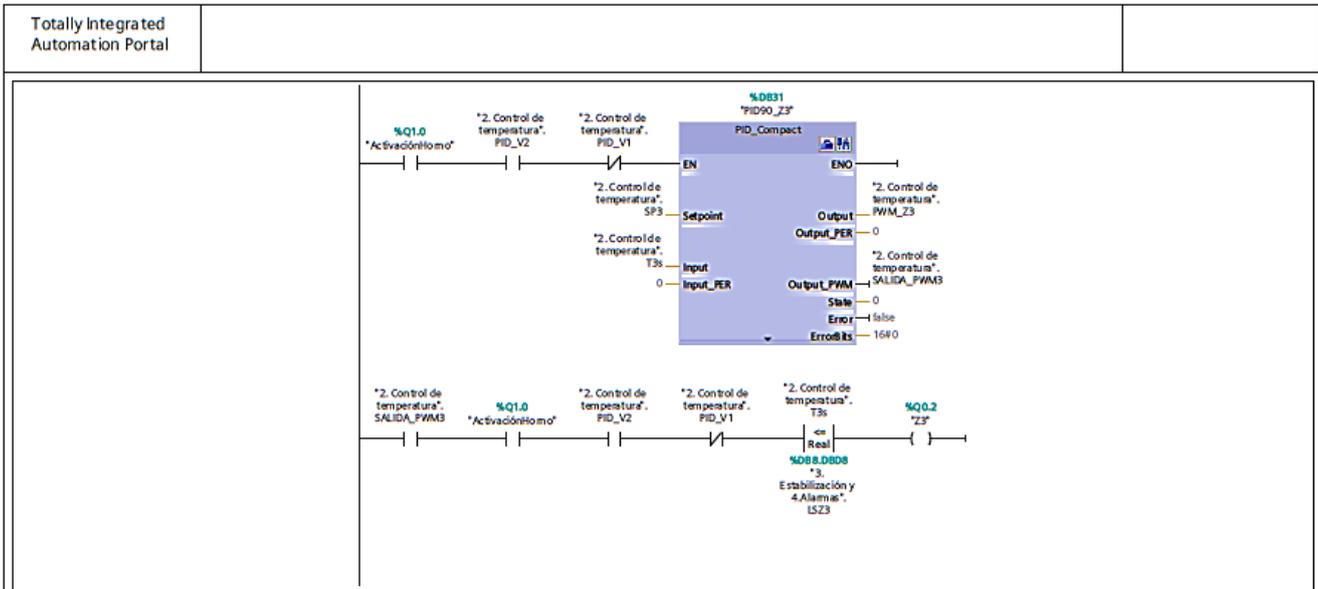
Segmento 1: PWM ZONA 1



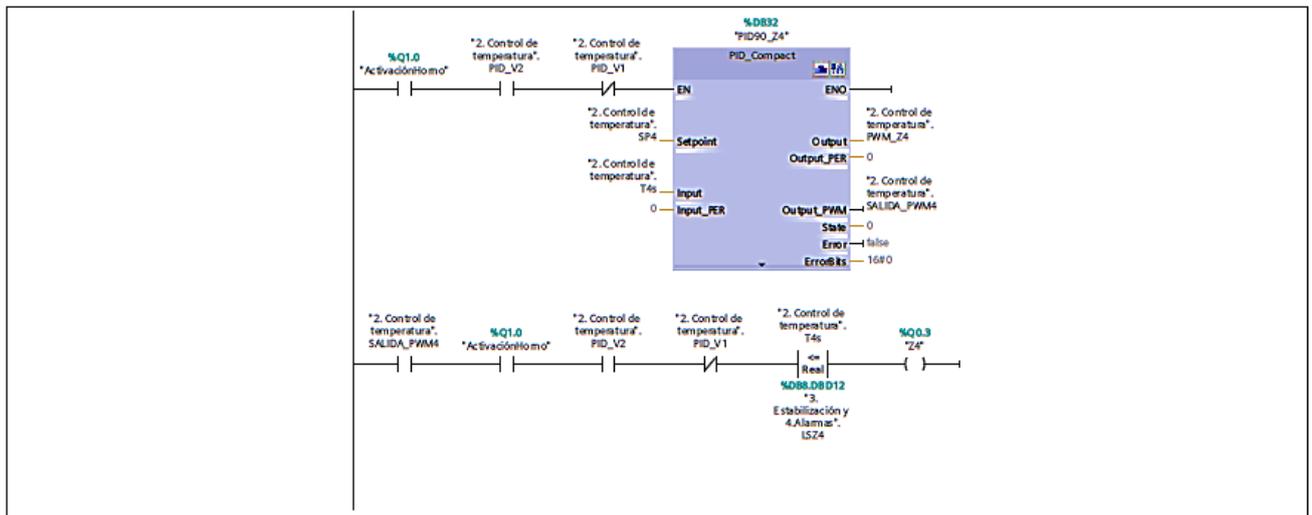
Segmento 2: PWM ZONA 2



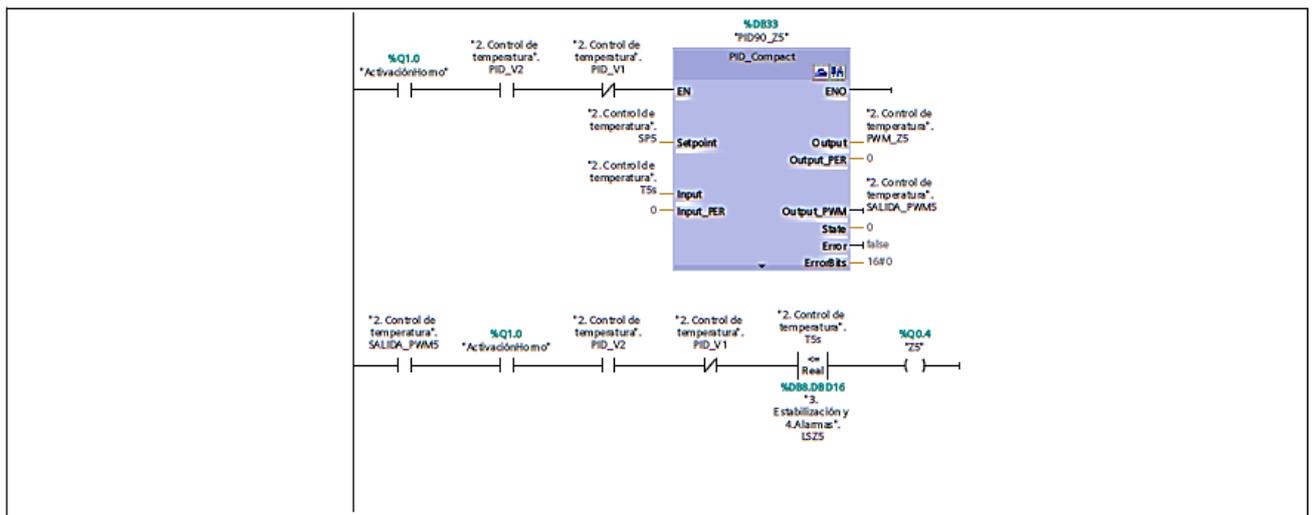
Segmento 3: PWM ZONA 3



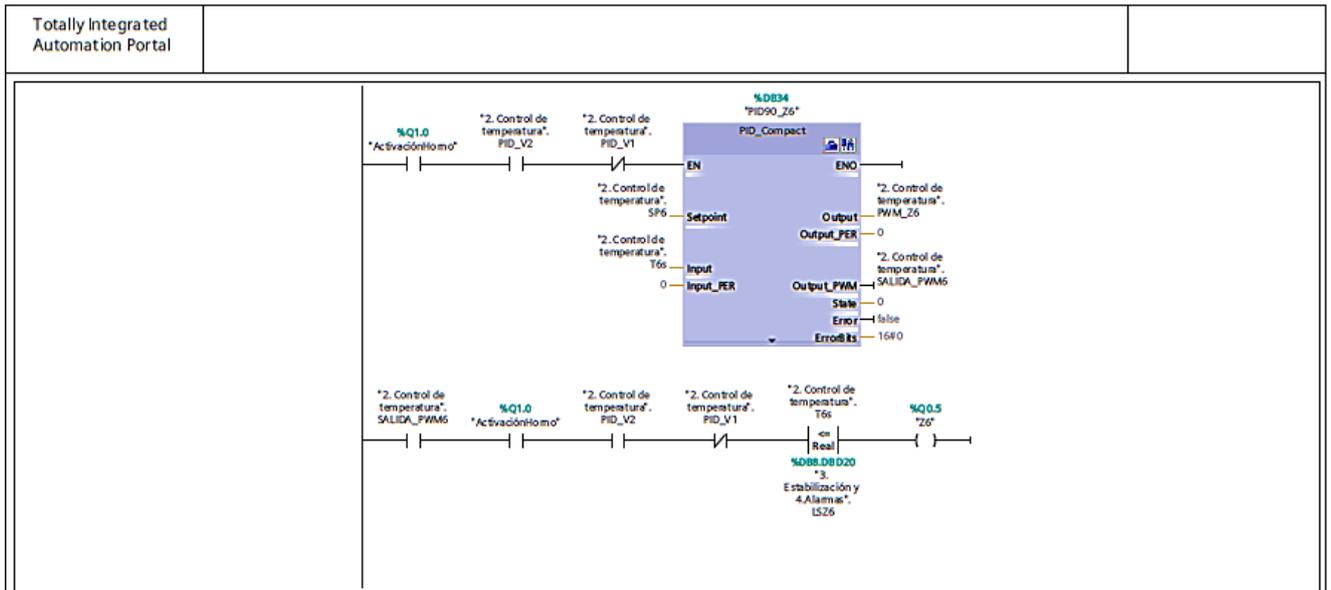
Segmento 4: PWM ZONA 4



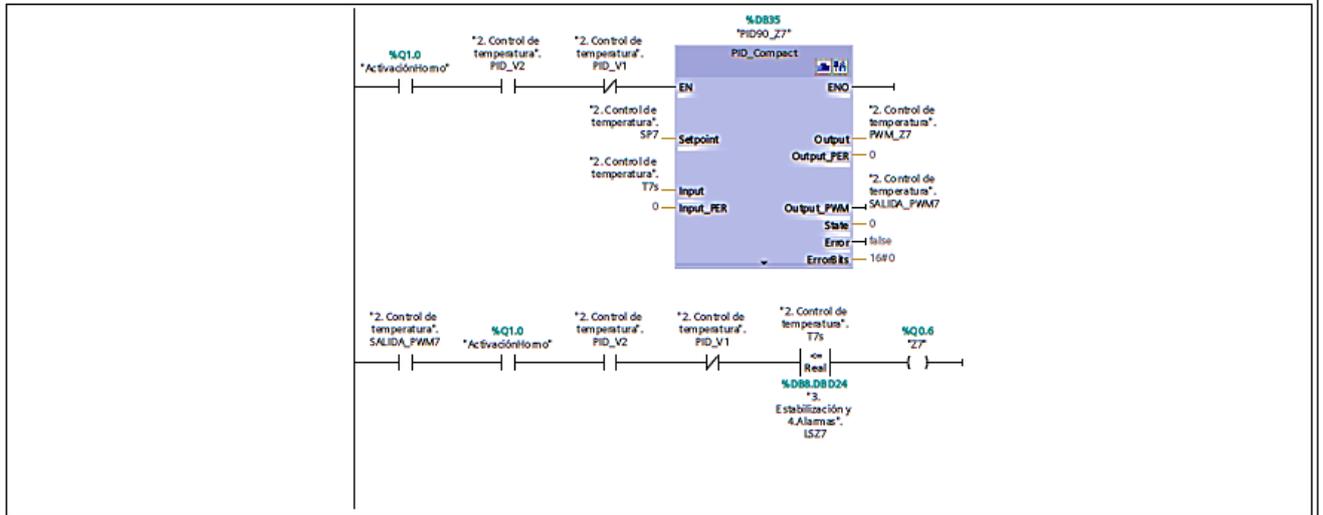
Segmento 5: PWM ZONA 5



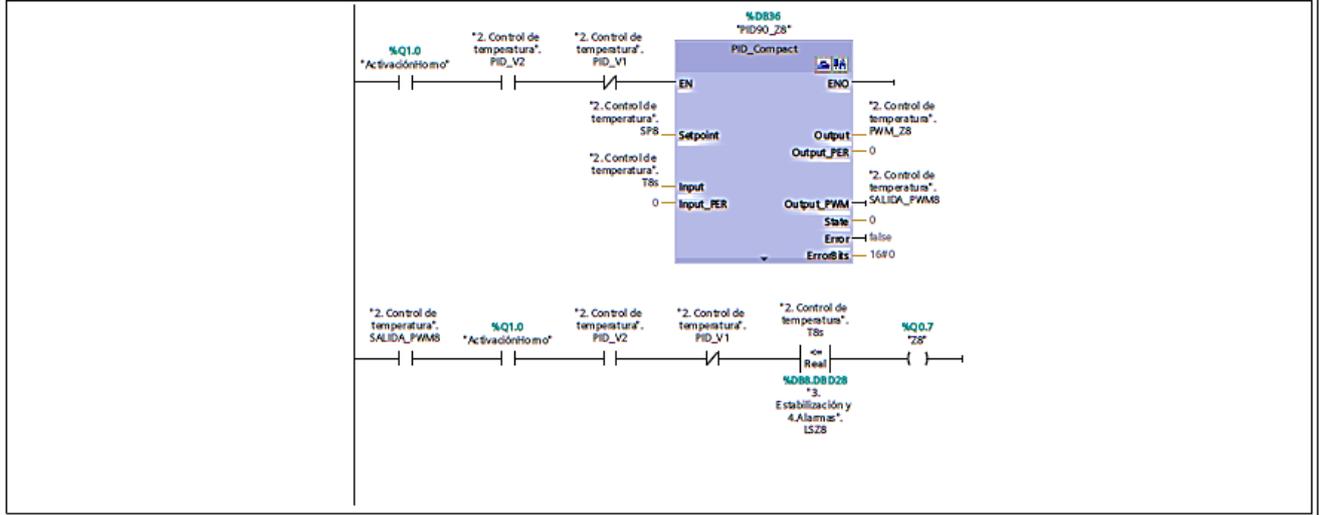
Segmento 6: PWM ZONA 6



Segmento 7: PWM ZONA 7

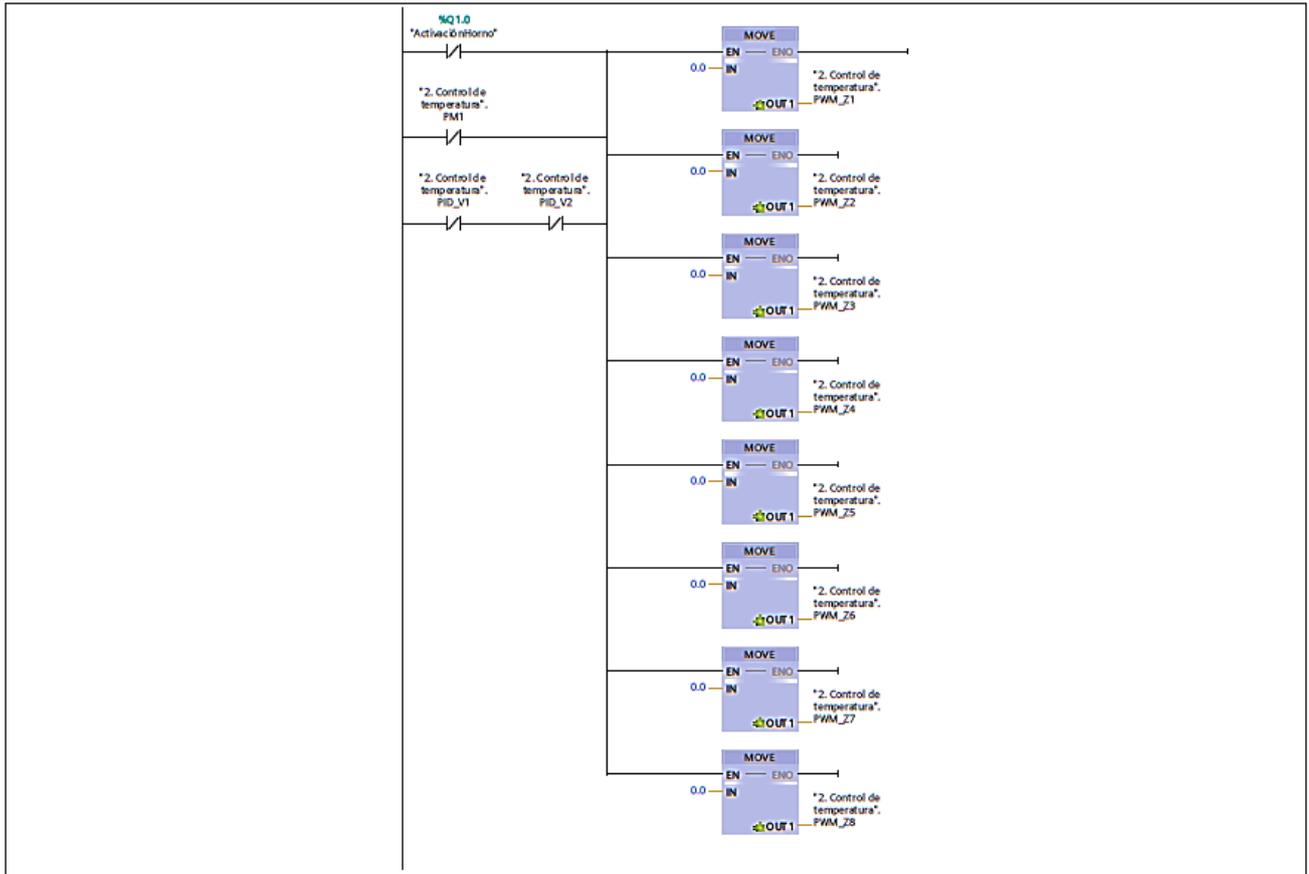


Segmento 8: PWM ZONA 8



Segmento 9:

SIRVE PARA PONER 0.0 EN LA SALIDA DE LOS PWM PARA CUANDO NO TENAMOS TENSIÓN O NO ESTEMOS CALENTANDO, PARA QUE A NIVEL VISUAL, EN EL HMI APAREZCA 0.0 EN EL PWM Y NO DAR LUGAR A MALENTENDIDOS, YA QUE SI QUITAMOS DE CALENTAR, NO CALENTARÁ PERO SE QUEDARÍA GUARDADO EL ÚLTIMO VALOR DEL PWM Y PUEDE DAR LUGAR A PENSAR QUE SI ESTA CALENTANDO.



Totally Integrated Automation Portal	
--------------------------------------	--

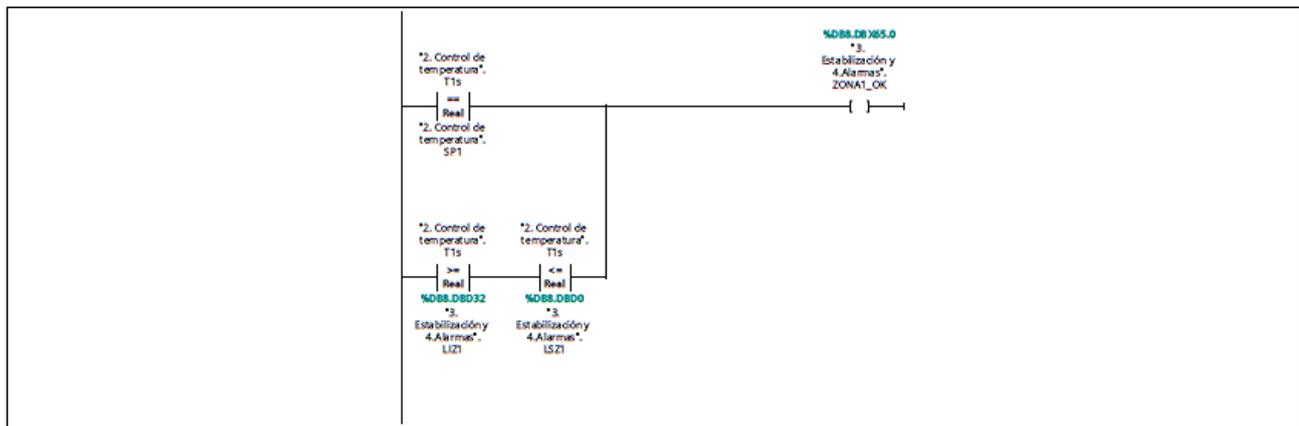
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

3.Estabilizacion [OB123]

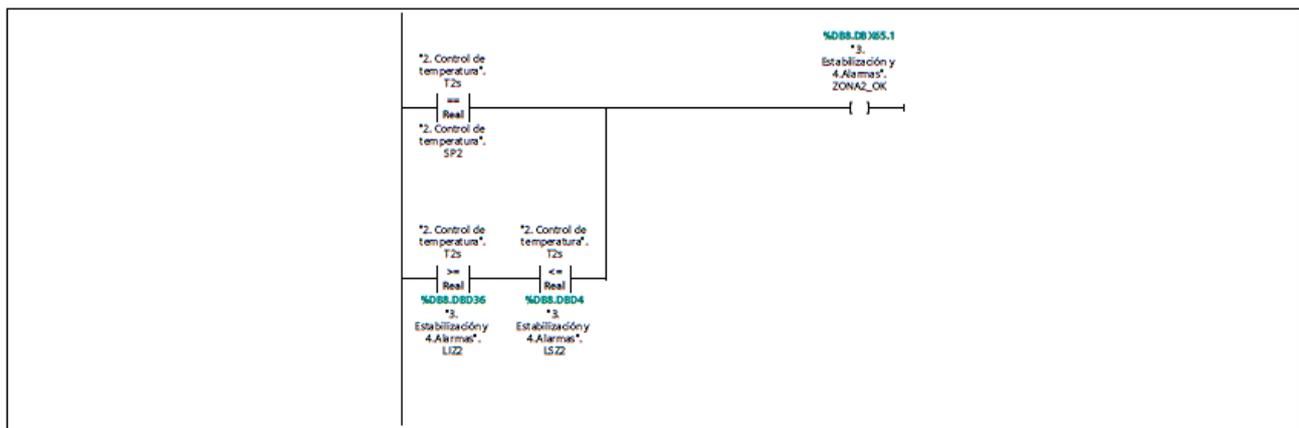
3.Estabilizacion Propiedades							
General							
Nombre	3.Estabilizacion	Número	123	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

3.Estabilizacion			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Segmento 1: Z1_OK



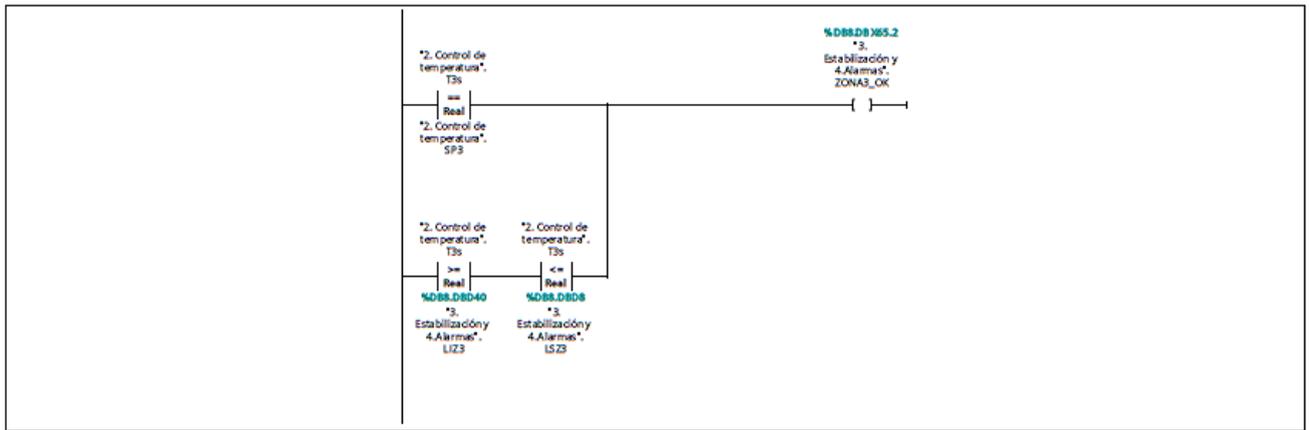
Segmento 2: Z2_OK



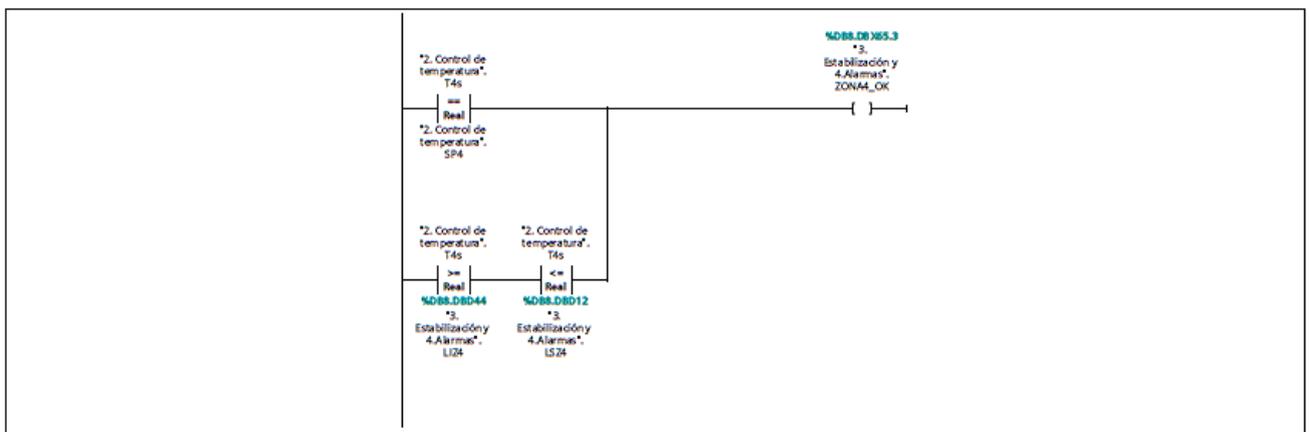
Segmento 3: Z3_OK

--	--

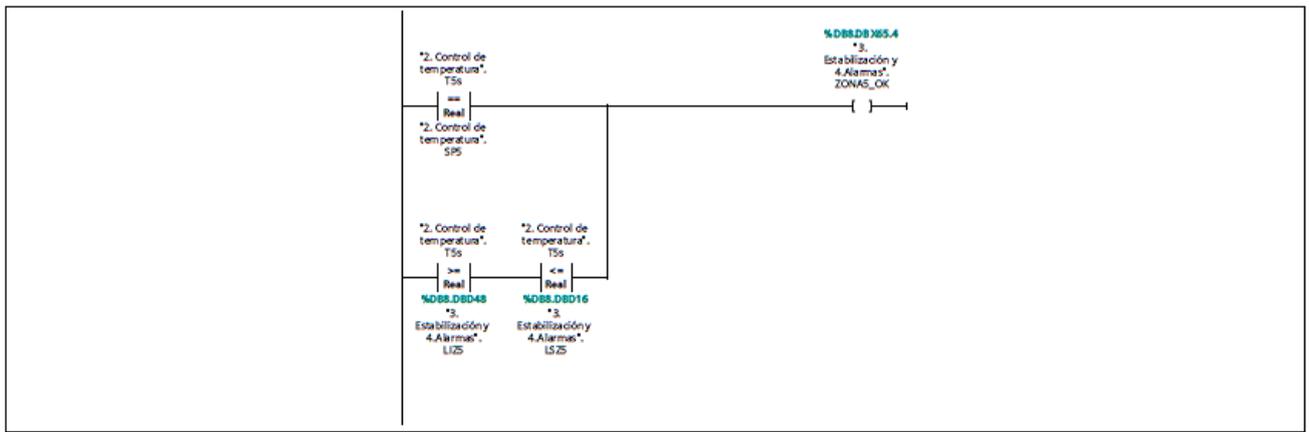
Totally Integrated Automation Portal



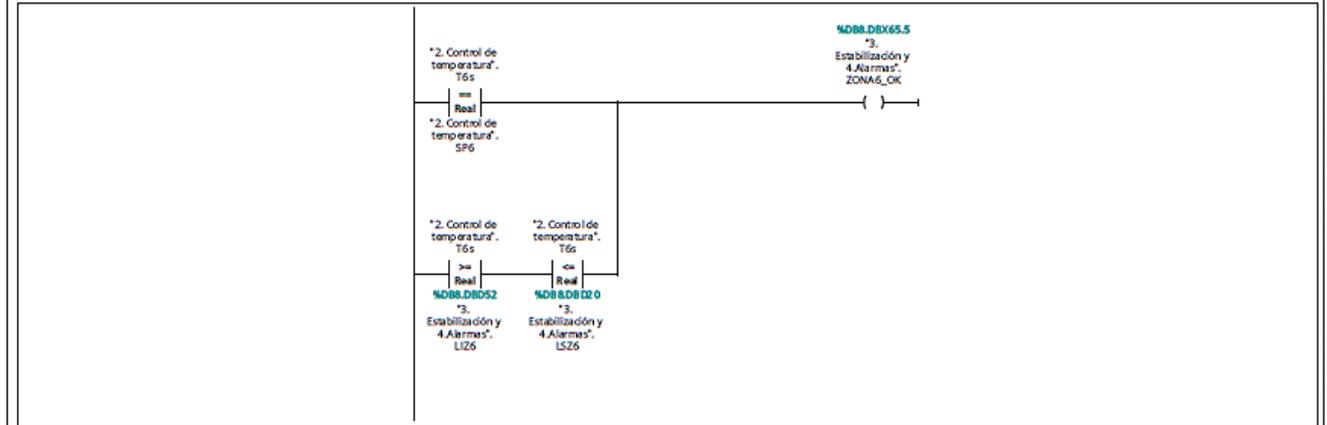
Segmento 4: Z4_OK



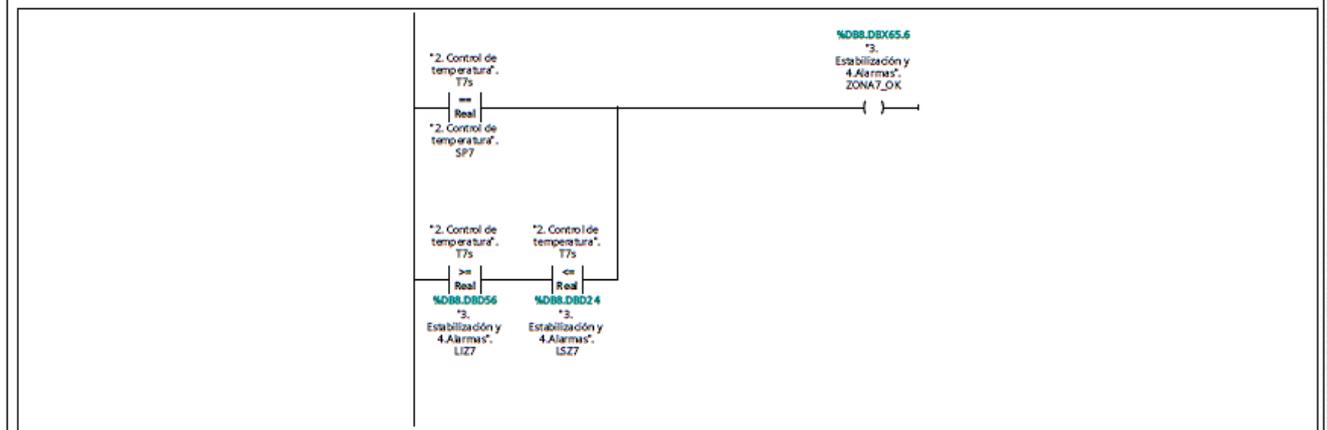
Segmento 5: Z5_OK



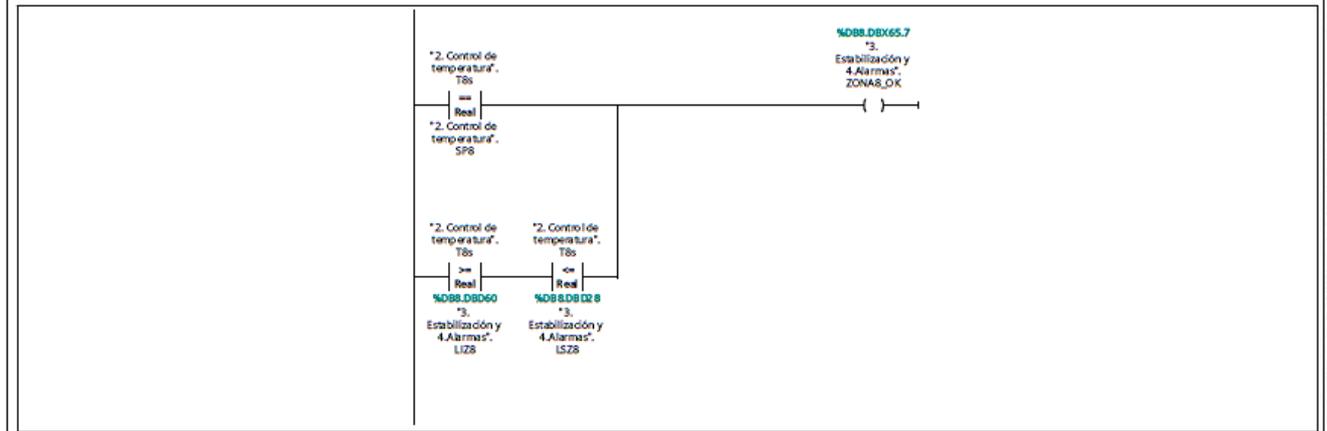
Segmento 6: Z6_OK



Segmento 7: Z7_OK



Segmento 8: Z8_OK



Totally Integrated Automation Portal	
--------------------------------------	--

HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

4.Alarmas [OB125]

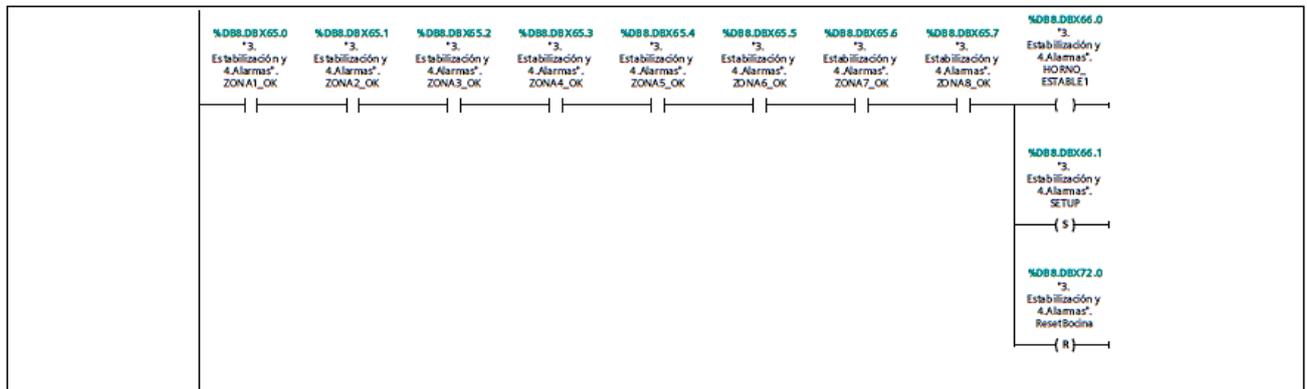
4.Alarmas Propiedades

General							
Nombre	4.Alarmas	Número	125	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cyde)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

4.Alarmas

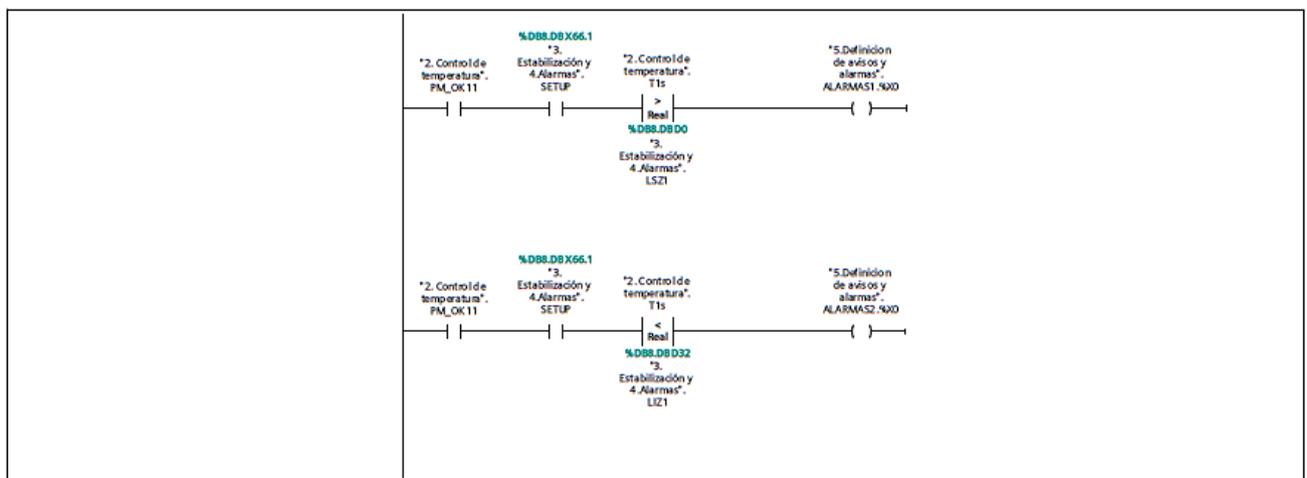
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Segmento 1: HORNO ESTABLE E INICIO SETUP



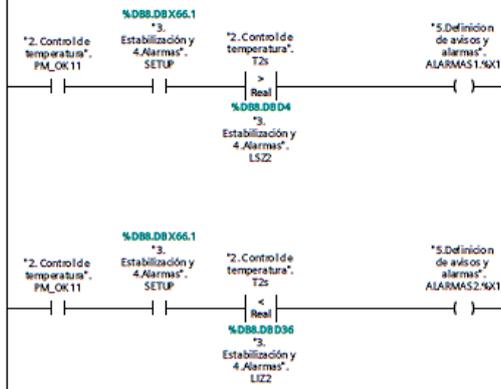
Segmento 2: ALARMA SUPERIOR E INFERIOR Z1

BYTE DE ACTIVACIÓN DE ALARMAS (0) Y BYTE DE ACTIVACION DE ACUSES (1)

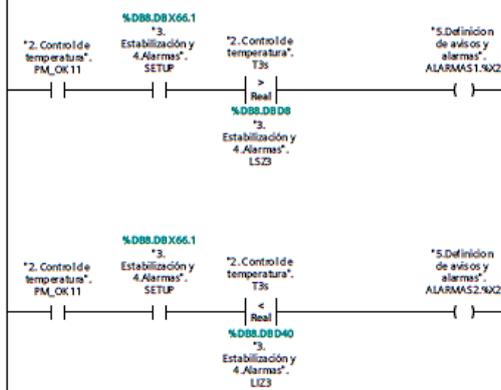


Segmento 3: ALARMA SUPERIOR E INFERIOR Z2

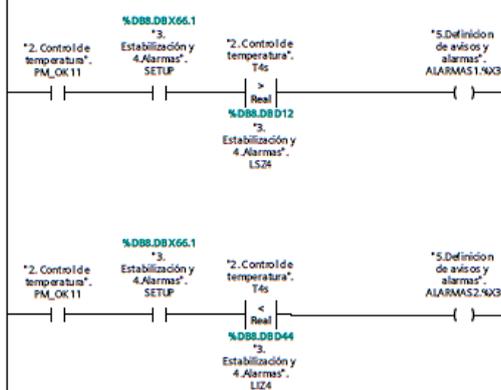
Totally Integrated Automation Portal



Segmento 4: ALARMA SUPERIOR E INFERIOR Z3

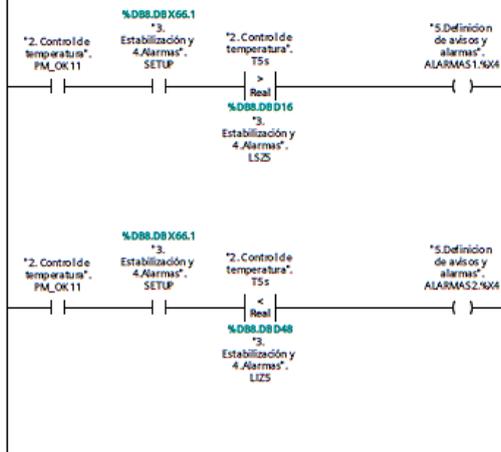


Segmento 5: ALARMA SUPERIOR E INFERIOR Z4

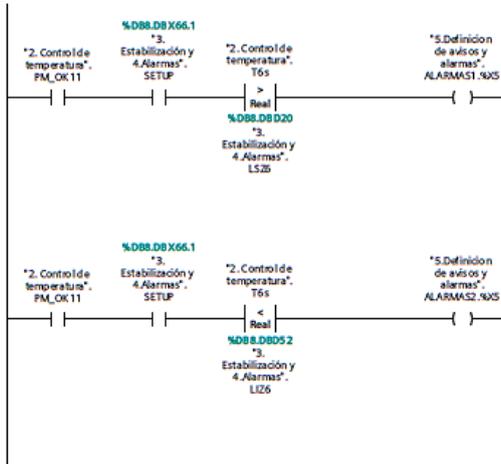


Segmento 6: ALARMA SUPERIOR E INFERIOR Z5

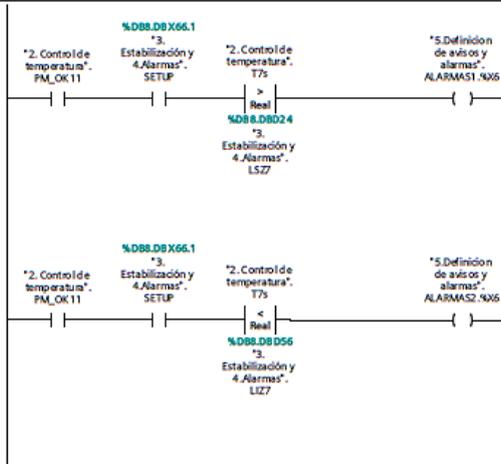
Totally Integrated Automation Portal



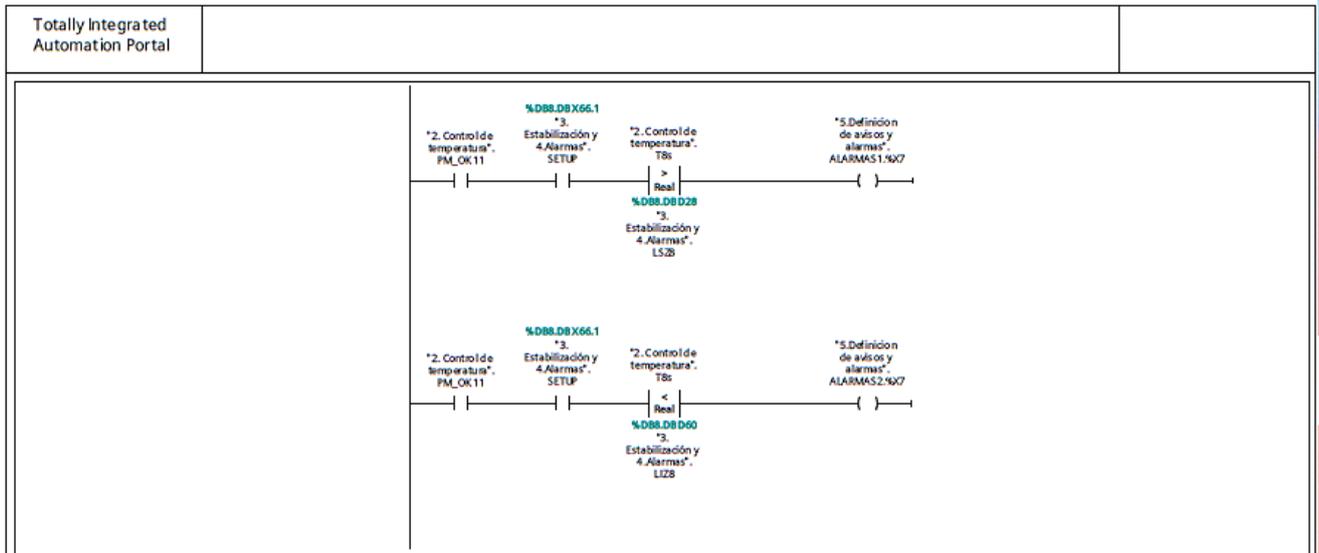
Segmento 7: ALARMA SUPERIOR E INFERIOR Z6



Segmento 8: ALARMA SUPERIOR E INFERIOR Z7

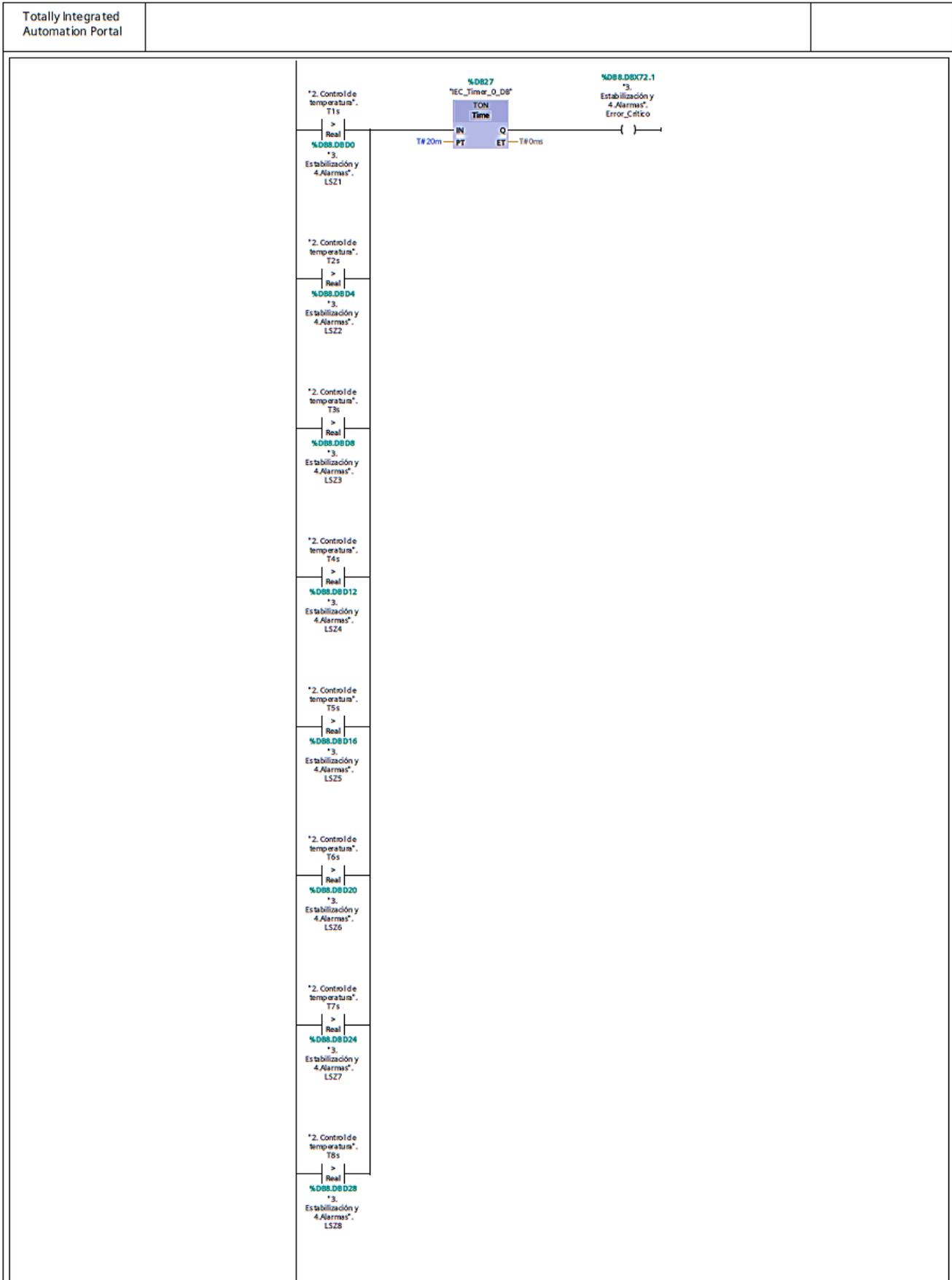


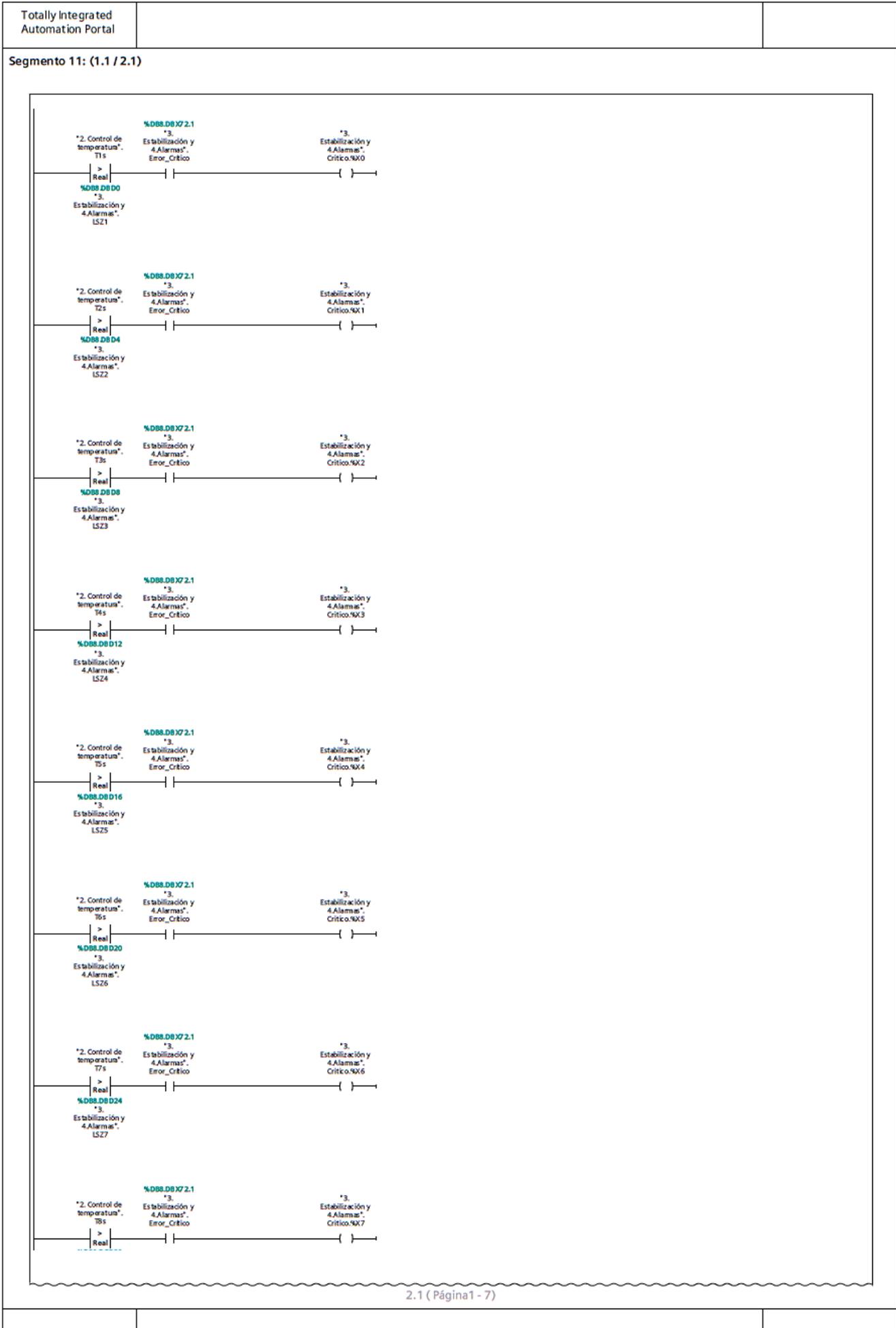
Segmento 9: ALARMA SUPERIOR E INFERIOR Z8



Segmento 10: ALARMA ROTURA DE RESISTENCIA O CONTAMINACIÓN TÉRMICA

En caso de que una zona exceda su temperatura superior límite, ya sea en SETUP o no, porque se ha roto una resistencia, se ha quedado un relé pegado o se ha contaminado una zona. Cortamos la tensión a los 15 minutos.





Totally Integrated Automation Portal		
Segmento 11: (2.1 / 2.1)		
<p style="text-align: right;">1.1 (Página1 - 6)</p> <div style="border: 1px solid black; padding: 10px;"><p>%DB8.DBD28 "3. Estabilización y 4.Alarme". LSZ8</p></div>		

Totally Integrated Automation Portal		
--------------------------------------	--	--

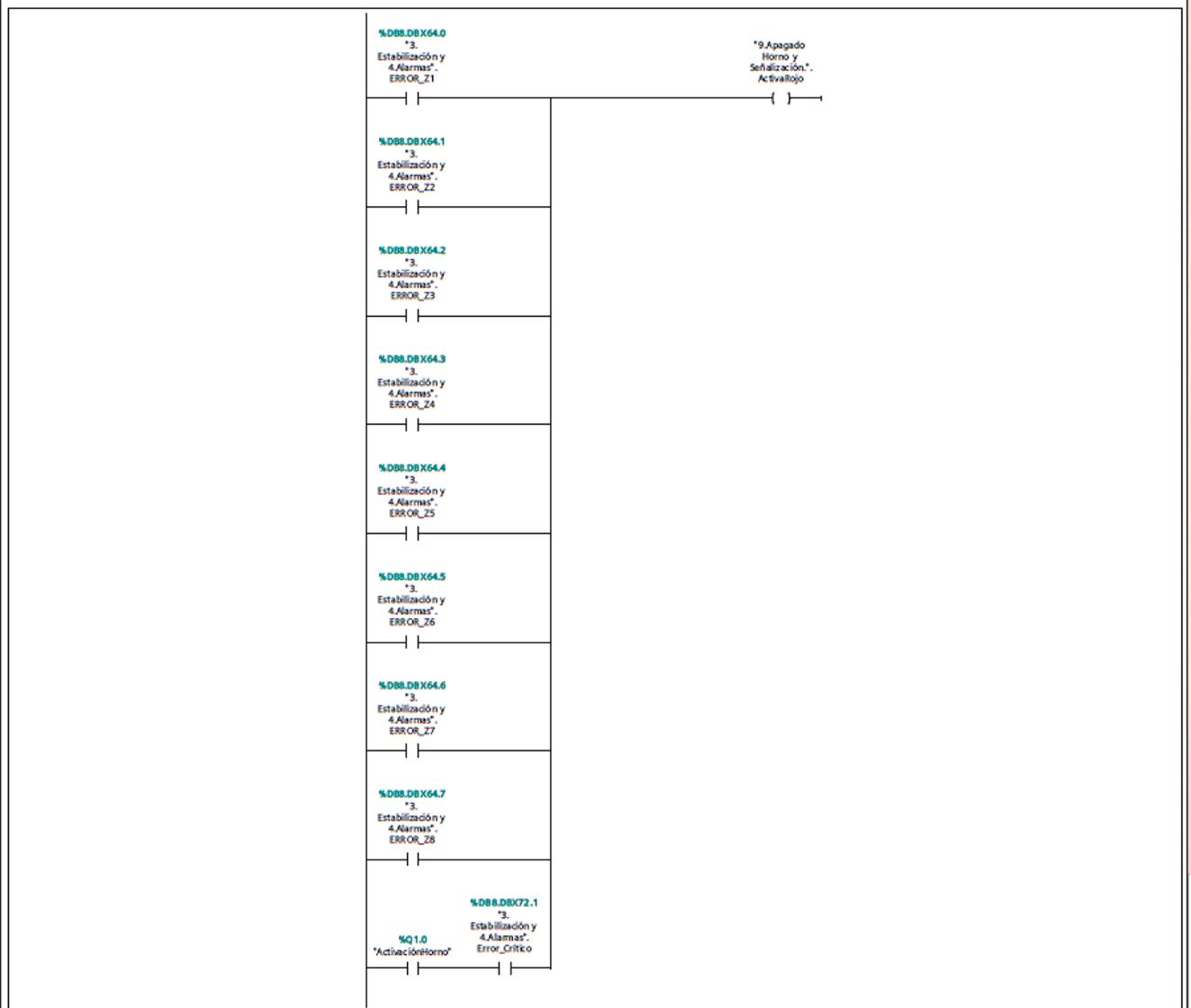
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

5.Gestion de alarmas [OB124]

5.Gestion de alarmas Propiedades							
General							
Nombre	5.Gestion de alarmas	Número	124	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

5.Gestión de alarmas			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Inicia_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Segmento 1: SI HAY ERROR, SEMÁFORO ROJO, BOCINA Y SI CONTINUA ERROR 30 MIN, QUITAMOS TENSIÓN.



Segmento 2: SI ACUSAMOS EN LA PANTALLA DE ALARMAS HMI, APAGAMOS BOCINA Y RESET SETUP.

<p>*S.Definicion de avisos y alarmas*. ACUSE1.%X0</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X1</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X2</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X3</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X4</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X5</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X6</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE1.%X7</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X0</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X1</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X2</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X3</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X4</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X5</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X6</p> <p> </p> <p>*S.Definicion de avisos y alarmas*. ACUSE2.%X7</p> <p> </p>	<p>----- (S) -----</p> <p>----- (R) -----</p> <p>----- (R) -----</p> <p>----- (R) -----</p>	<p>%DBB.DBX7.2.0 *3. Estabilización y 4.Alarmas*. Rese#Bocina</p> <p>%DBB.DBX66.1 *3. Estabilización y 4.Alarmas*. SETUP</p> <p>%DBB.DBX7.2.1 *3. Estabilización y 4.Alarmas*. Error_Critico</p>
---	---	---

Segmento 3: HORNO ESTABLE, RESETEAMOS ACUSES.

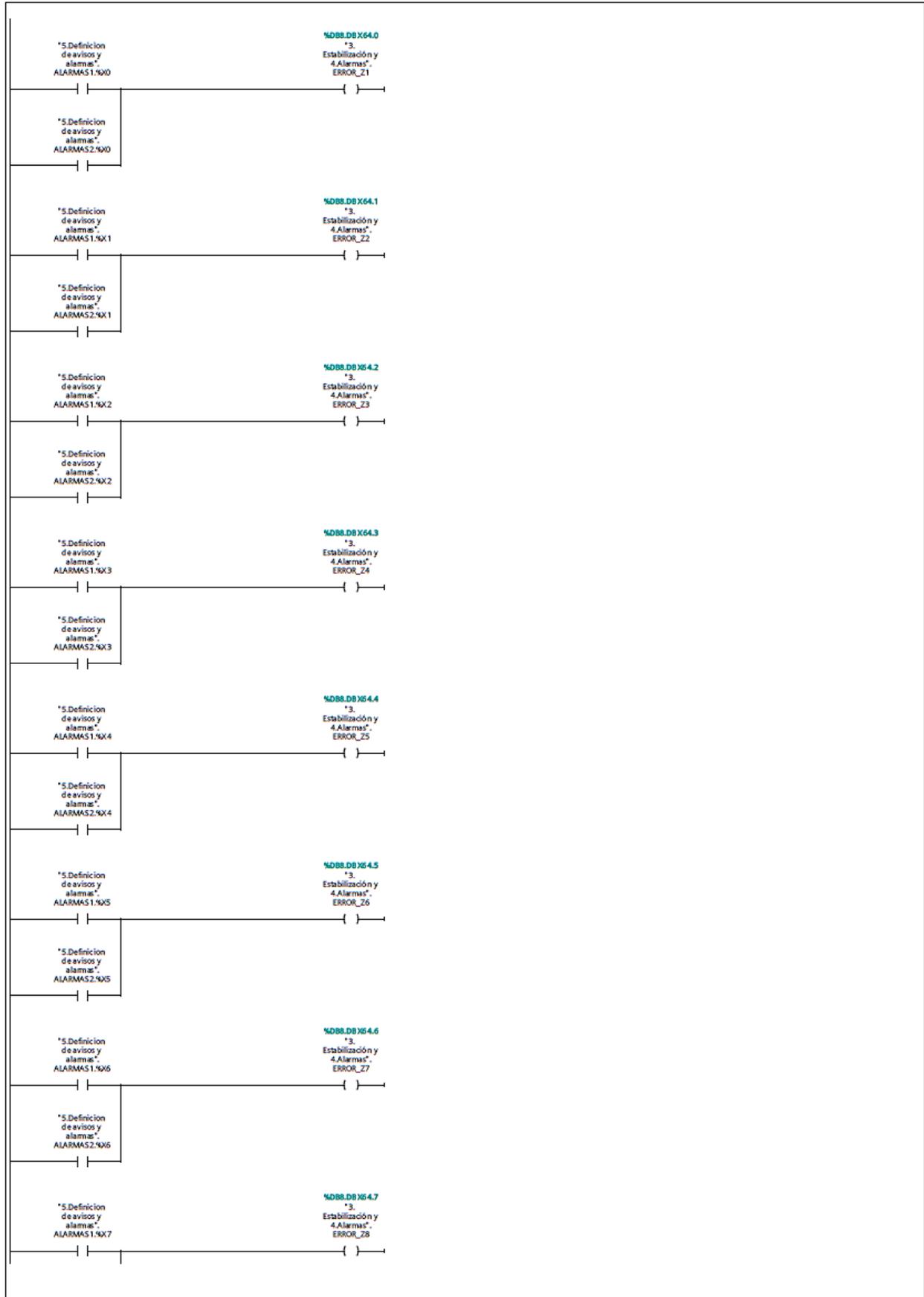
	<p>%DBB.DBX6.0 *3. Estabilización y 4.Alarmas"- HORNO_ ESTABLE1</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X0 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X1 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X2 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X3 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X4 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X5 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X6 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE1.%X7 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X0 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X1 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X2 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X3 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X4 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X5 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X6 {R}</p> <p>"S.Definición de avisos y alarmas"- ACUSE2.%X7 {R}</p>
--	---

Segmento 4: SI TENEMOS ALARMA X, TENEMOS ERROR EN ZONA X.

ALARMA 1 == SUPERIOR
ALARMA 2 == INFERIOR

Totally Integrated Automation Portal

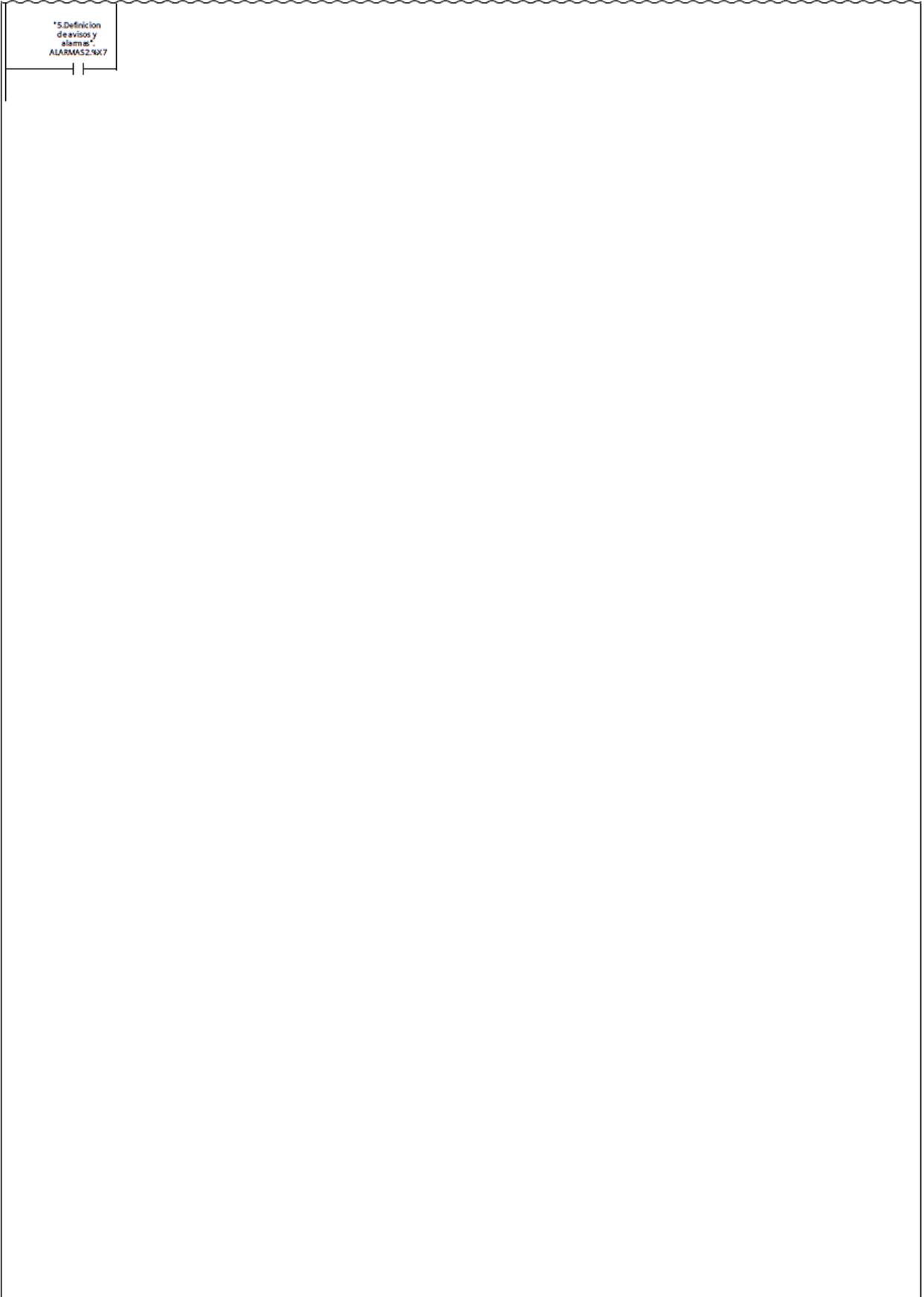
Segmento 4: SI TENEMOS ALARMA X, TENEMOS ERROR EN ZONA X. (1.1 / 2.1)



Totally Integrated Automation Portal		
--------------------------------------	--	--

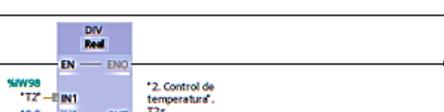
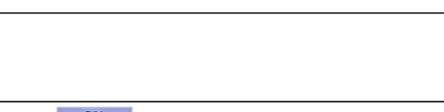
Segmento 4: SI TENEMOS ALARMA X, TENEMOS ERROR EN ZONA X. (2.1 / 2.1)

1.1 (Página1 - 4)

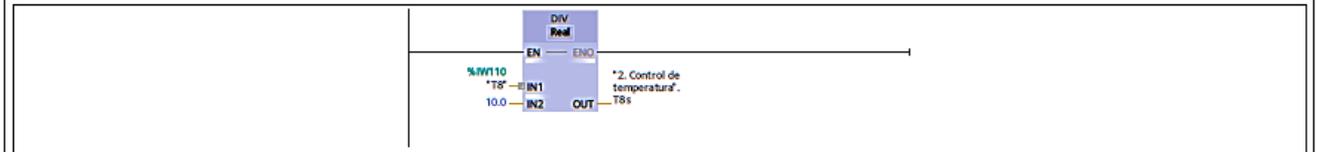


--	--	--

<p>Totally Integrated Automation Portal</p>	<p>Segmento 5: HORNO ESTABLE, RESET ALARMAS.</p> <table border="1"> <tr> <td data-bbox="571 295 667 1966"> <p>%DBB.DBX6.0 *3 Estabilización y 4.Alarmas". HORNO_ ESTABLE1</p> </td> <td data-bbox="667 295 1072 1966"> <table border="1"> <tr> <td data-bbox="986 324 1066 385"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X0</p> </td> <td data-bbox="1008 385 1066 407"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 430 1066 490"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X0</p> </td> <td data-bbox="1008 490 1066 512"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 535 1066 595"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X1</p> </td> <td data-bbox="1008 595 1066 618"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 640 1066 701"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X1</p> </td> <td data-bbox="1008 701 1066 723"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 723 1066 784"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X2</p> </td> <td data-bbox="1008 784 1066 806"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 828 1066 889"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X2</p> </td> <td data-bbox="1008 889 1066 911"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 934 1066 994"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X3</p> </td> <td data-bbox="1008 994 1066 1016"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1039 1066 1099"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X3</p> </td> <td data-bbox="1008 1099 1066 1122"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1144 1066 1205"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X4</p> </td> <td data-bbox="1008 1205 1066 1227"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1227 1066 1288"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X4</p> </td> <td data-bbox="1008 1288 1066 1310"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1310 1066 1370"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X5</p> </td> <td data-bbox="1008 1370 1066 1393"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1393 1066 1453"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X5</p> </td> <td data-bbox="1008 1453 1066 1476"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1476 1066 1536"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X6</p> </td> <td data-bbox="1008 1536 1066 1559"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1559 1066 1619"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X6</p> </td> <td data-bbox="1008 1619 1066 1641"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1641 1066 1702"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X7</p> </td> <td data-bbox="1008 1702 1066 1724"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1724 1066 1785"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X7</p> </td> <td data-bbox="1008 1785 1066 1807"> <p>{ R }</p> </td> </tr> </table> </td> </tr> </table>	<p>%DBB.DBX6.0 *3 Estabilización y 4.Alarmas". HORNO_ ESTABLE1</p>	<table border="1"> <tr> <td data-bbox="986 324 1066 385"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X0</p> </td> <td data-bbox="1008 385 1066 407"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 430 1066 490"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X0</p> </td> <td data-bbox="1008 490 1066 512"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 535 1066 595"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X1</p> </td> <td data-bbox="1008 595 1066 618"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 640 1066 701"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X1</p> </td> <td data-bbox="1008 701 1066 723"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 723 1066 784"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X2</p> </td> <td data-bbox="1008 784 1066 806"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 828 1066 889"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X2</p> </td> <td data-bbox="1008 889 1066 911"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 934 1066 994"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X3</p> </td> <td data-bbox="1008 994 1066 1016"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1039 1066 1099"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X3</p> </td> <td data-bbox="1008 1099 1066 1122"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1144 1066 1205"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X4</p> </td> <td data-bbox="1008 1205 1066 1227"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1227 1066 1288"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X4</p> </td> <td data-bbox="1008 1288 1066 1310"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1310 1066 1370"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X5</p> </td> <td data-bbox="1008 1370 1066 1393"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1393 1066 1453"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X5</p> </td> <td data-bbox="1008 1453 1066 1476"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1476 1066 1536"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X6</p> </td> <td data-bbox="1008 1536 1066 1559"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1559 1066 1619"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X6</p> </td> <td data-bbox="1008 1619 1066 1641"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1641 1066 1702"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X7</p> </td> <td data-bbox="1008 1702 1066 1724"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1724 1066 1785"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X7</p> </td> <td data-bbox="1008 1785 1066 1807"> <p>{ R }</p> </td> </tr> </table>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X0</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X0</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X1</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X1</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X2</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X2</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X3</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X3</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X4</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X4</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X5</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X5</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X6</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X6</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X7</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X7</p>	<p>{ R }</p>
<p>%DBB.DBX6.0 *3 Estabilización y 4.Alarmas". HORNO_ ESTABLE1</p>	<table border="1"> <tr> <td data-bbox="986 324 1066 385"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X0</p> </td> <td data-bbox="1008 385 1066 407"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 430 1066 490"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X0</p> </td> <td data-bbox="1008 490 1066 512"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 535 1066 595"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X1</p> </td> <td data-bbox="1008 595 1066 618"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 640 1066 701"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X1</p> </td> <td data-bbox="1008 701 1066 723"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 723 1066 784"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X2</p> </td> <td data-bbox="1008 784 1066 806"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 828 1066 889"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X2</p> </td> <td data-bbox="1008 889 1066 911"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 934 1066 994"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X3</p> </td> <td data-bbox="1008 994 1066 1016"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1039 1066 1099"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X3</p> </td> <td data-bbox="1008 1099 1066 1122"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1144 1066 1205"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X4</p> </td> <td data-bbox="1008 1205 1066 1227"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1227 1066 1288"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X4</p> </td> <td data-bbox="1008 1288 1066 1310"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1310 1066 1370"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X5</p> </td> <td data-bbox="1008 1370 1066 1393"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1393 1066 1453"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X5</p> </td> <td data-bbox="1008 1453 1066 1476"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1476 1066 1536"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X6</p> </td> <td data-bbox="1008 1536 1066 1559"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1559 1066 1619"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X6</p> </td> <td data-bbox="1008 1619 1066 1641"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1641 1066 1702"> <p>"5_Definición de avisos y alarmas". ALARMAS1.%X7</p> </td> <td data-bbox="1008 1702 1066 1724"> <p>{ R }</p> </td> </tr> <tr> <td data-bbox="986 1724 1066 1785"> <p>"5_Definición de avisos y alarmas". ALARMAS2.%X7</p> </td> <td data-bbox="1008 1785 1066 1807"> <p>{ R }</p> </td> </tr> </table>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X0</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X0</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X1</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X1</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X2</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X2</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X3</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X3</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X4</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X4</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X5</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X5</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X6</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X6</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS1.%X7</p>	<p>{ R }</p>	<p>"5_Definición de avisos y alarmas". ALARMAS2.%X7</p>	<p>{ R }</p>		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X0</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X0</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X1</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X1</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X2</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X2</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X3</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X3</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X4</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X4</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X5</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X5</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X6</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X6</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS1.%X7</p>	<p>{ R }</p>																																		
<p>"5_Definición de avisos y alarmas". ALARMAS2.%X7</p>	<p>{ R }</p>																																		

<p>Totally Integrated Automation Portal</p>		
<p>Segmento 6: TRANSFORMACIÓN TEMP. Z1</p>		
<p>CONTROL DEL PID DE LA PRIMERA ZONA</p>		
		
<p>Segmento 7: TRANSFORMACIÓN TEMP. Z2</p>		
<p>CONTROL DEL PID DE LA SEGUNDA ZONA</p>		
		
<p>Segmento 8: TRANSFORMACIÓN TEMP. Z3</p>		
<p>CONTROL DEL PID DE LA TERCERA ZONA</p>		
		
<p>Segmento 9: TRANSFORMACIÓN TEMP. Z4</p>		
<p>CONTROL DEL PID DE LA CUARTA ZONA</p>		
		
<p>Segmento 10: TRANSFORMACIÓN TEMP. Z5</p>		
		
<p>Segmento 11: TRANSFORMACIÓN TEMP. Z6</p>		
		
<p>Segmento 12: TRANSFORMACIÓN TEMP. Z7</p>		
		
<p>Segmento 13: TRANSFORMACIÓN TEMP. Z8</p>		

Totally Integrated Automation Portal		
--------------------------------------	--	--



--	--	--

Totally Integrated Automation Portal		
--------------------------------------	--	--

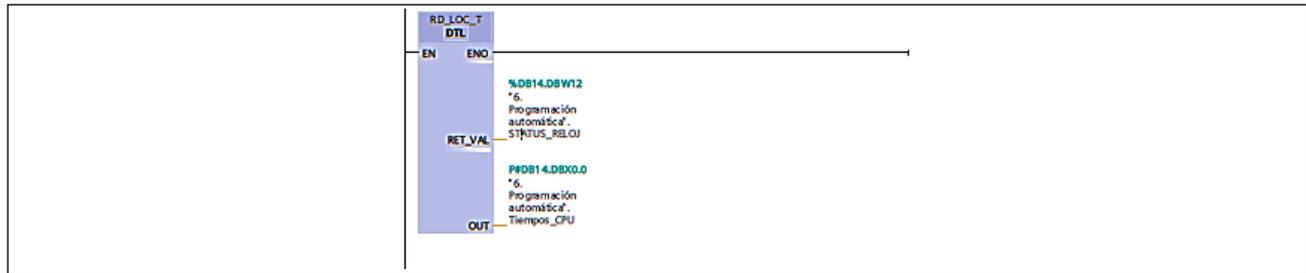
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

6.programacion automatica [OB128]

6.programacion automatica Propiedades							
General							
Nombre	6.programacion automatica	Número	128	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

6.programacion automatica			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

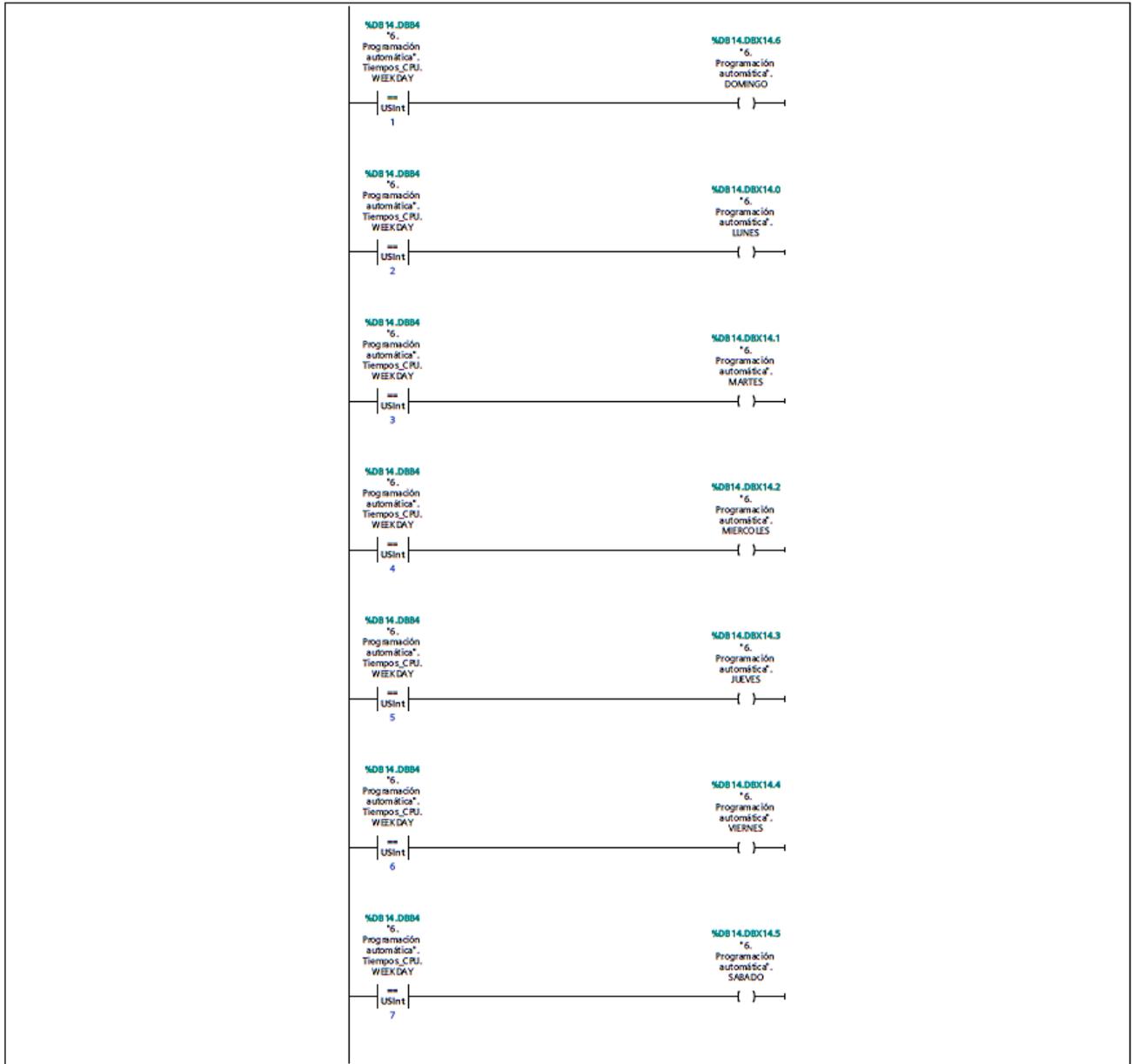
Segmento 1: RD_LOC_T



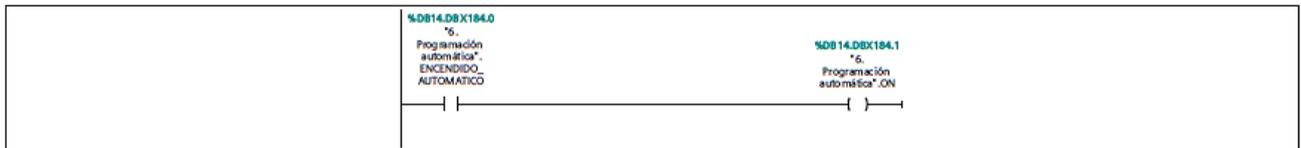
Segmento 2: DIA DE LA SEMANA

AQUÍ SE DESGLOSA EN QUE DIA DE LA SEMANA ESTAMOS

Totally Integrated Automation Portal

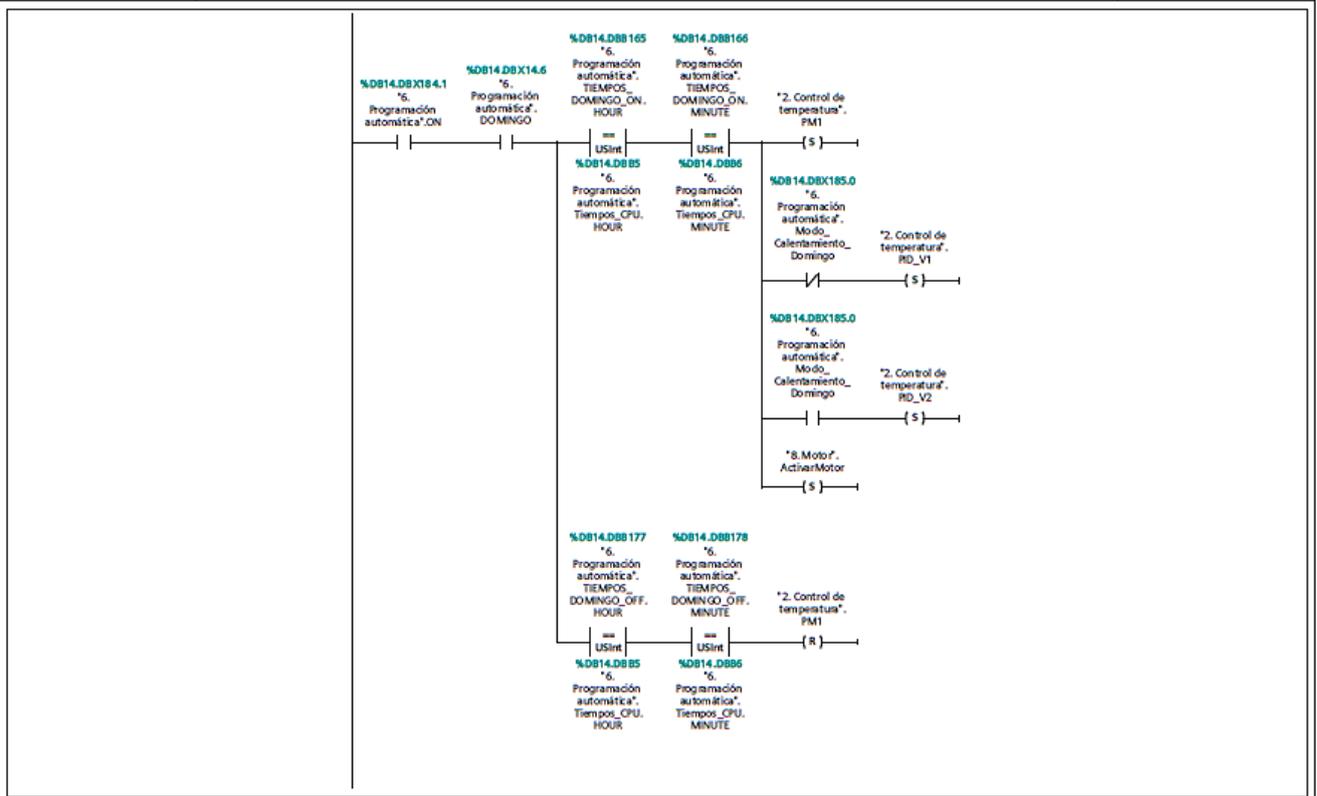


Segmento 3: ACTIVACIÓN ENCENDIDO AUTOMÁTICO



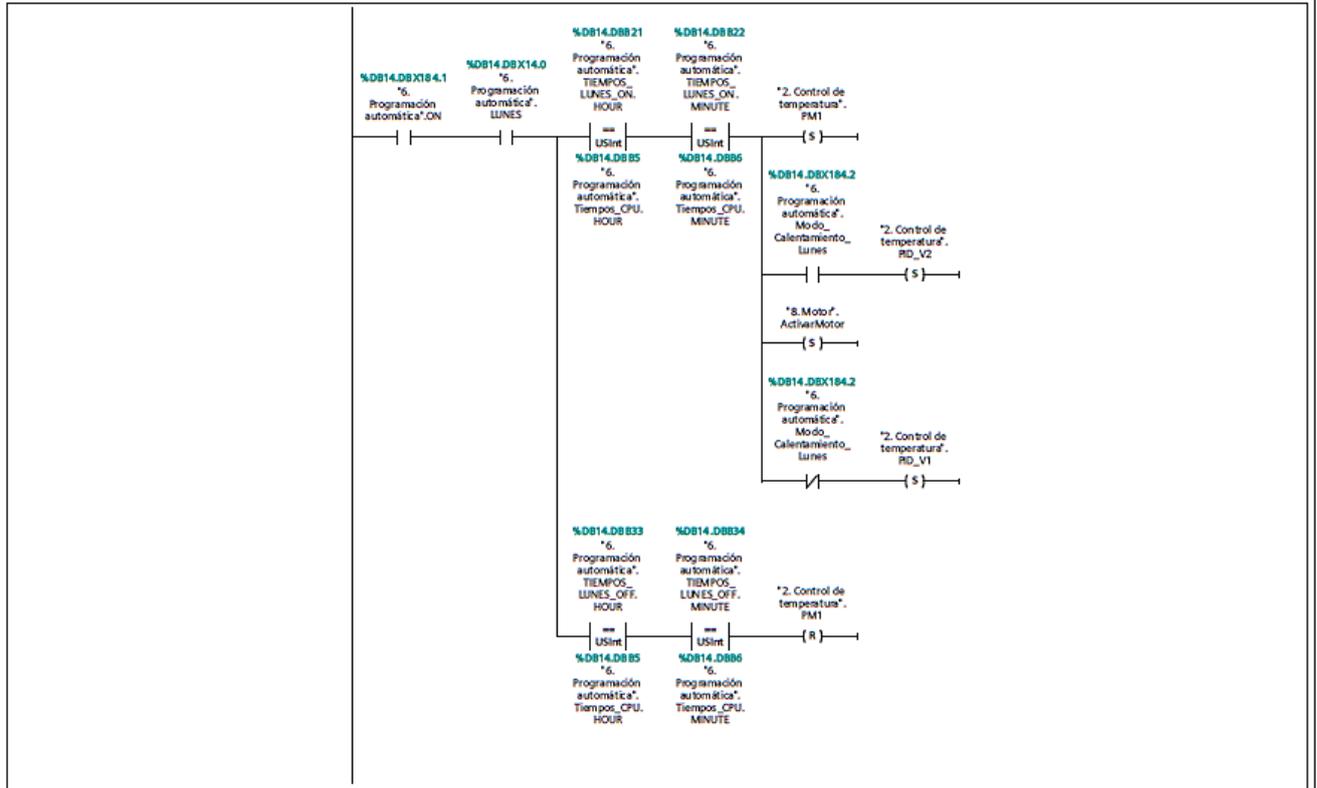
Segmento 4: DOMINGO

SE ACTIVA SIEMPRE Y CUANDO SEA DOMINGO



Segmento 5: LUNES

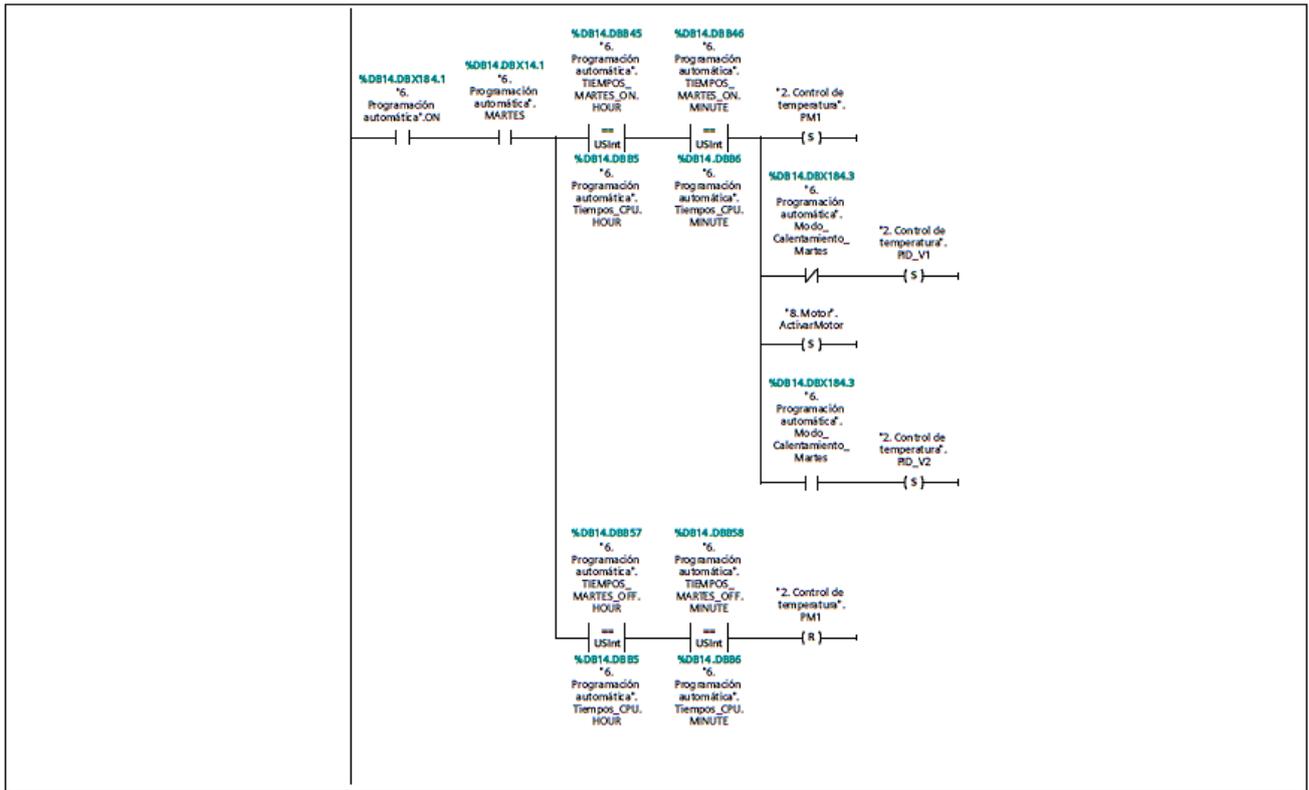
SE ACTIVA SIEMPRE Y CUANDO SEA LUNES



Segmento 6: MARTES

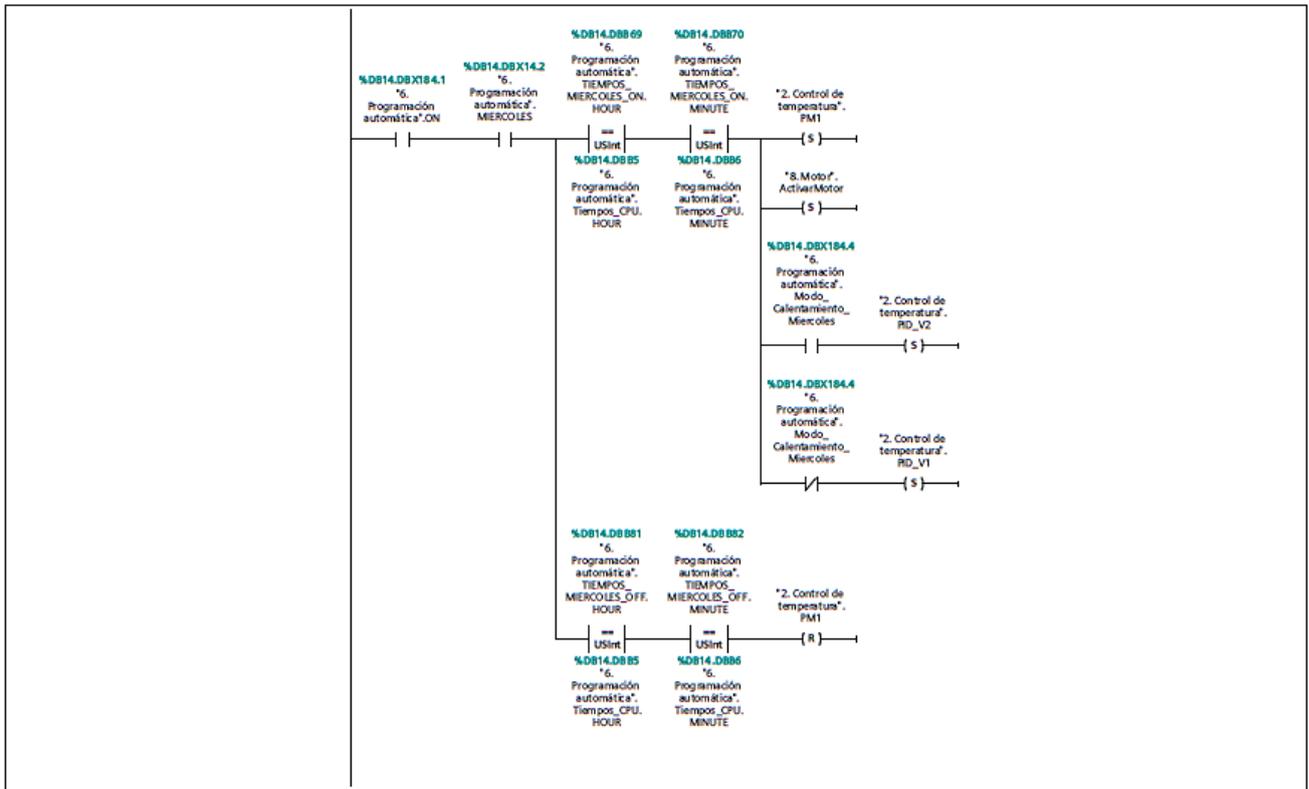
SE ACTIVA SIEMPRE Y CUANDO SEA MARTES

Totally Integrated Automation Portal



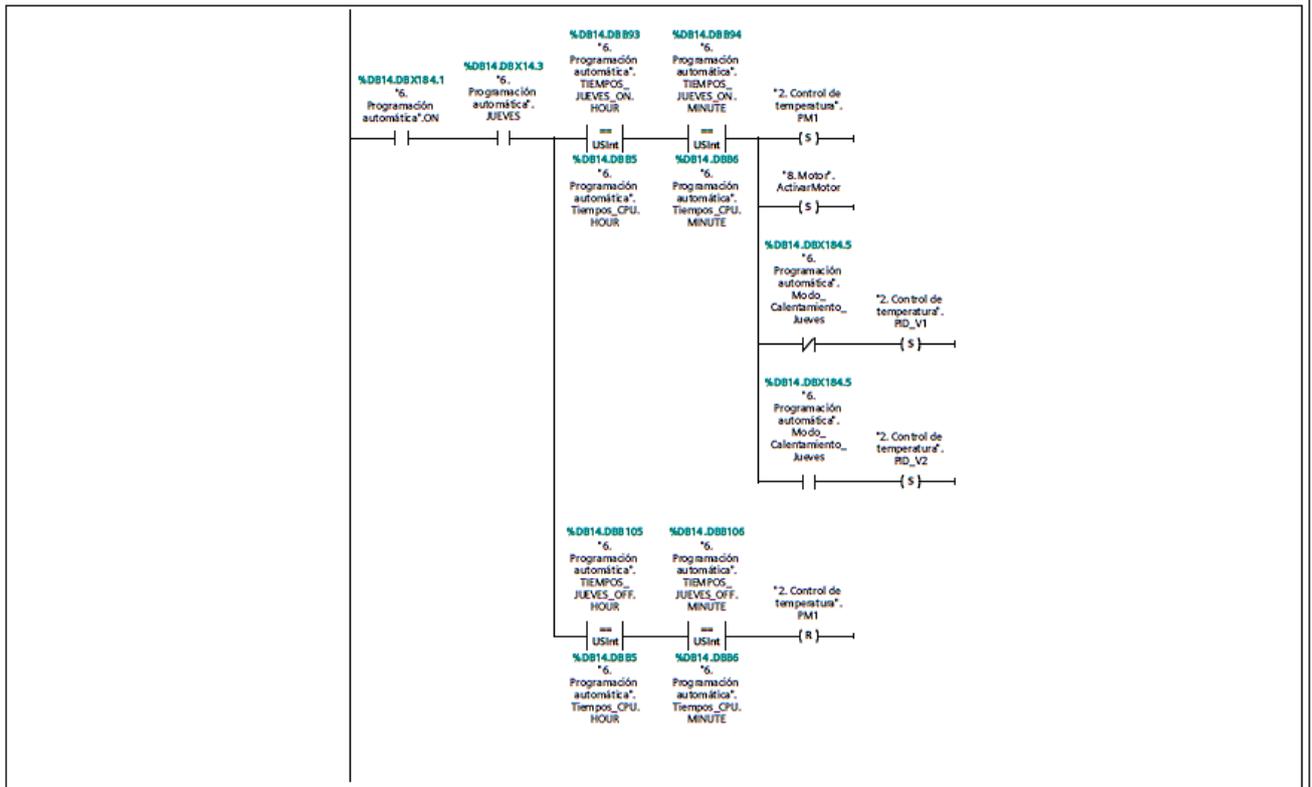
Segmento 7: MIERCOLES

SE ACTIVA SIEMPRE Y CUANDO SEA MIERCOLES



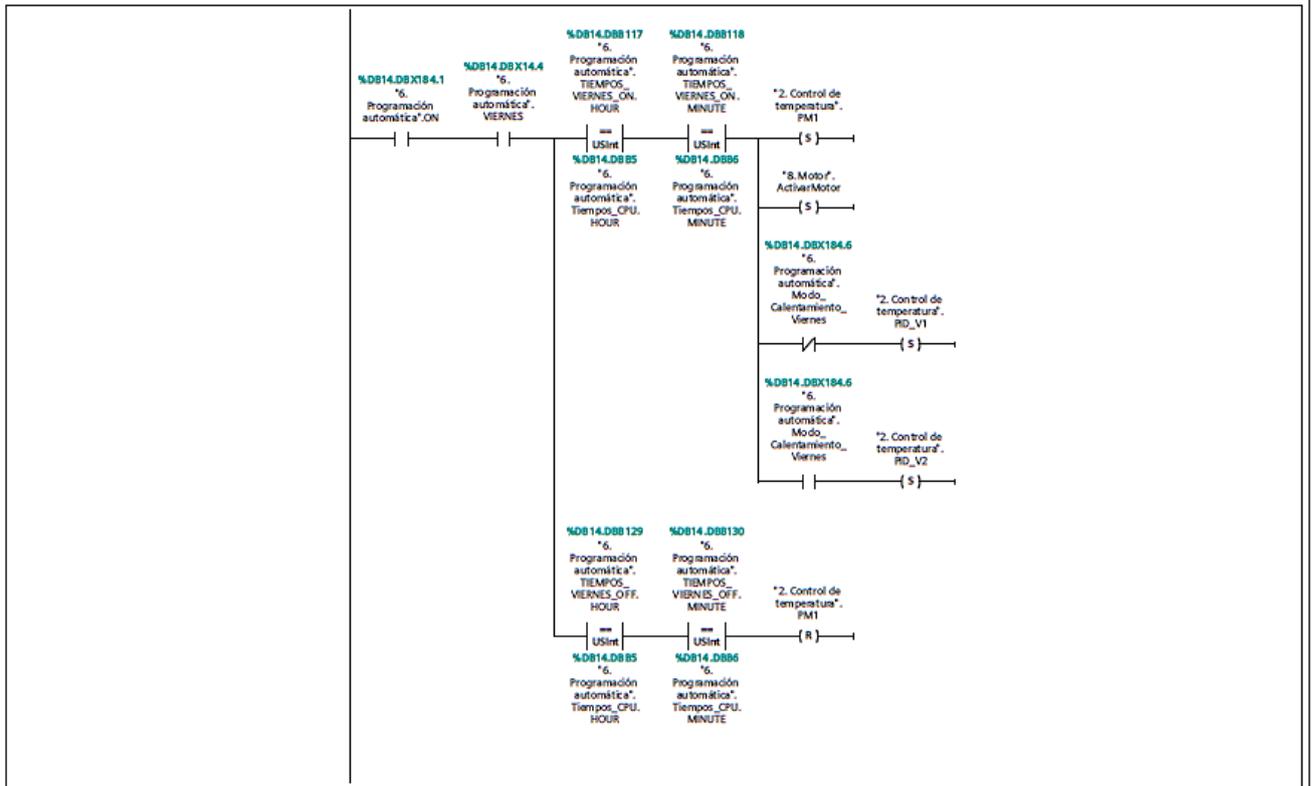
Segmento 8: JUEVES

SE ACTIVA CUANDO ES JUEVES



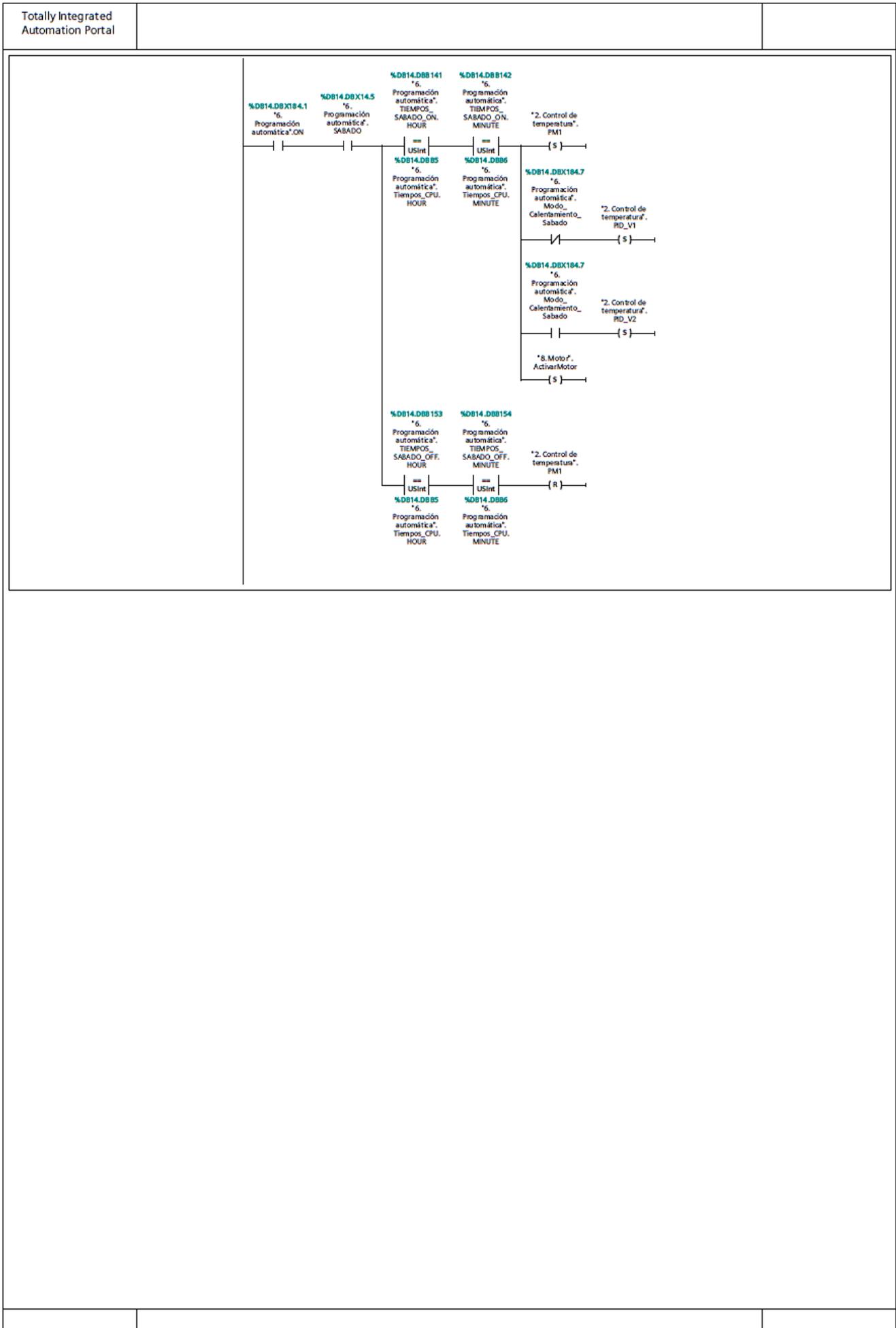
Segmento 9: VIERNES

SE ACTIVA CUANDO ES VIERNES



Segmento 10: SABADO

SE ACTIVA CUANDO ES SABADO



Totally Integrated Automation Portal		
--------------------------------------	--	--

HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

7.Warnings [OB1]

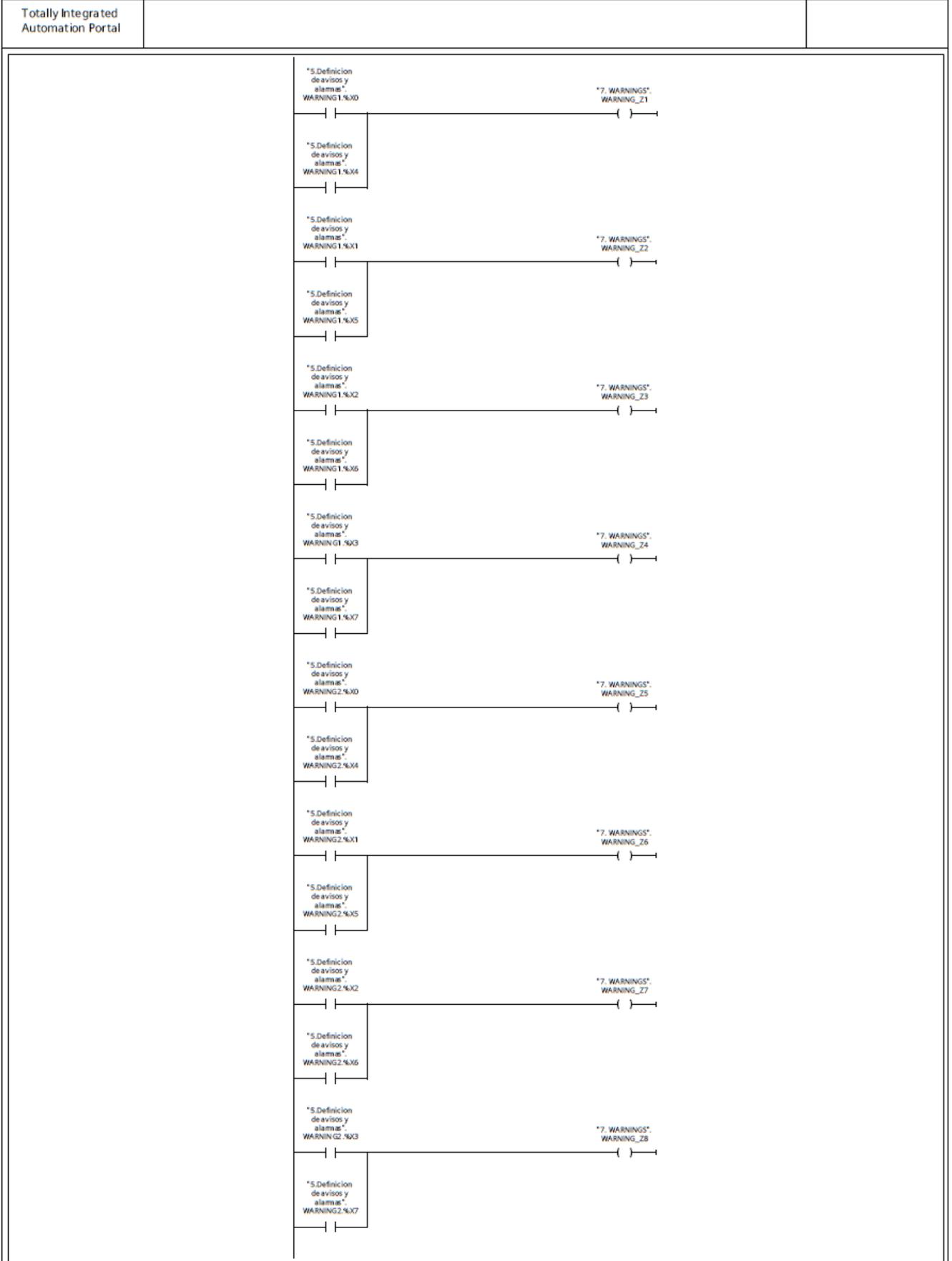
7.Warnings Propiedades							
General							
Nombre	7.Warnings	Número	1	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cy- de)"	Autor		Comentario		Familia	
Versión	0.1	ID personaliza- do					

7.Warnings			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Segmento 1: ACTIVA WARNING

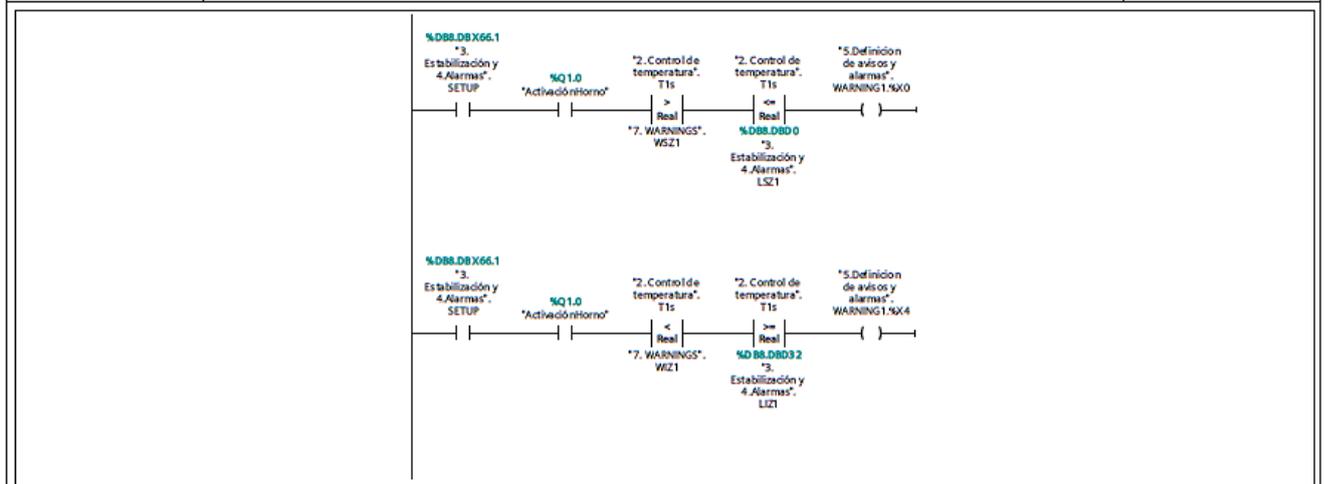
SI TENEMOS ADVERTENCIA EN CUALQUIER ZONA, ACTIVAMOS WARNING DE LA ZONA DE ADVERTENCIA.

--	--	--

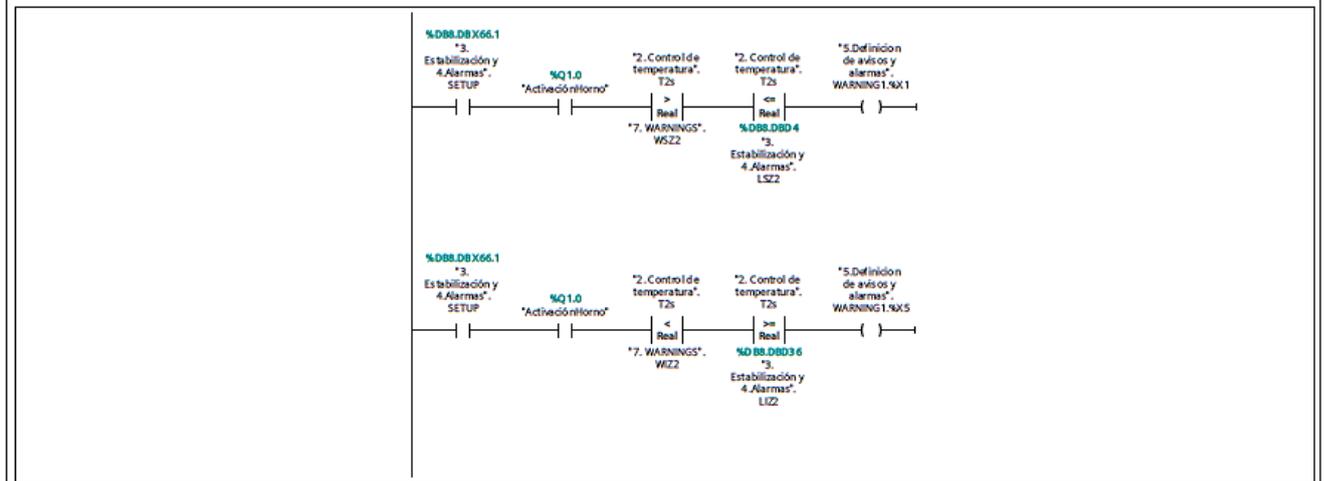


Segmento 2: ZONA 1
 SI ESTAMOS EN EL SETUP Y TENEMOS UNA TEMPERATURA MENOR QUE LA ALARMA SUPERIOR Y MAYOR QUE LA ADVERTENCIA SUPERIOR, TENEMOS UN WARNING

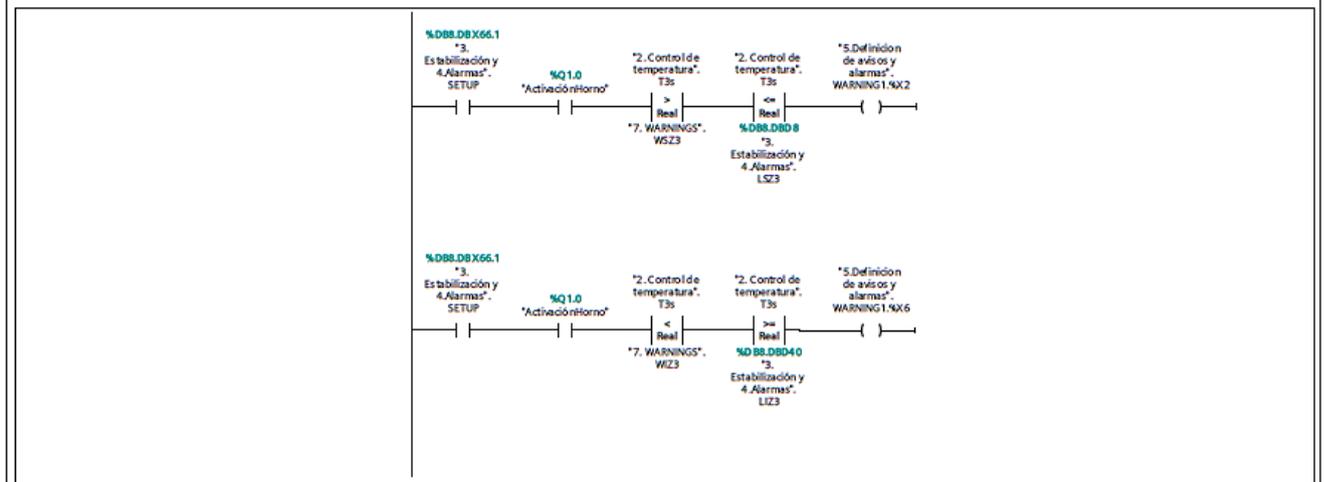
Totally Integrated Automation Portal		
--------------------------------------	--	--



Segmento 3: ZONA 2



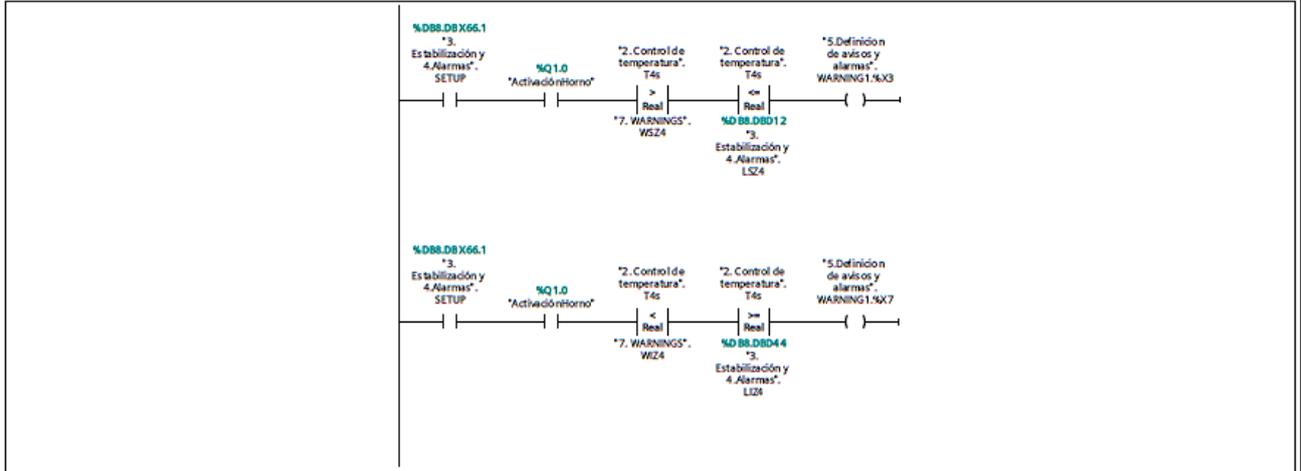
Segmento 4: ZONA 3



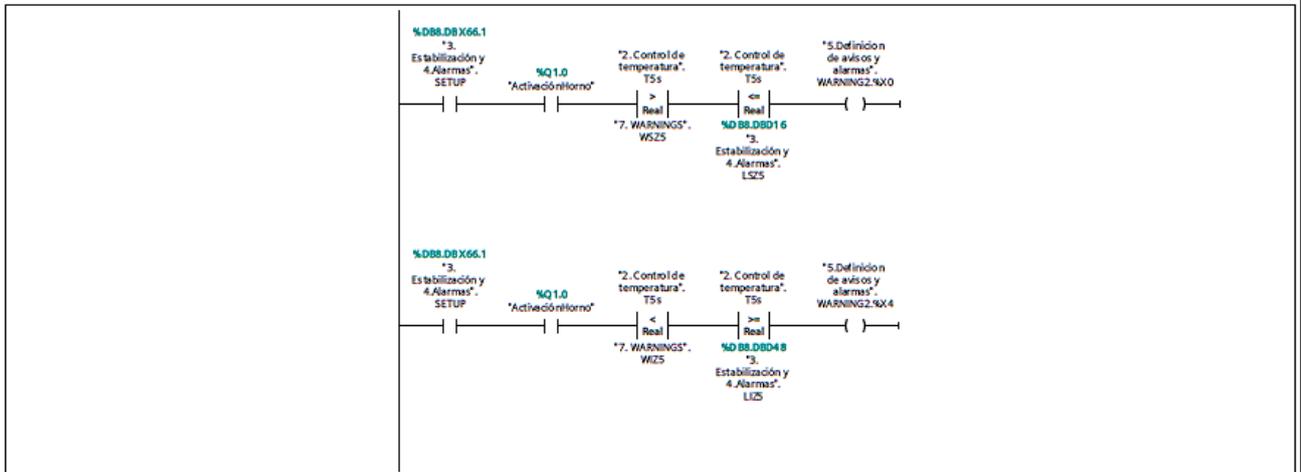
Segmento 5: ZONA 4

--	--	--

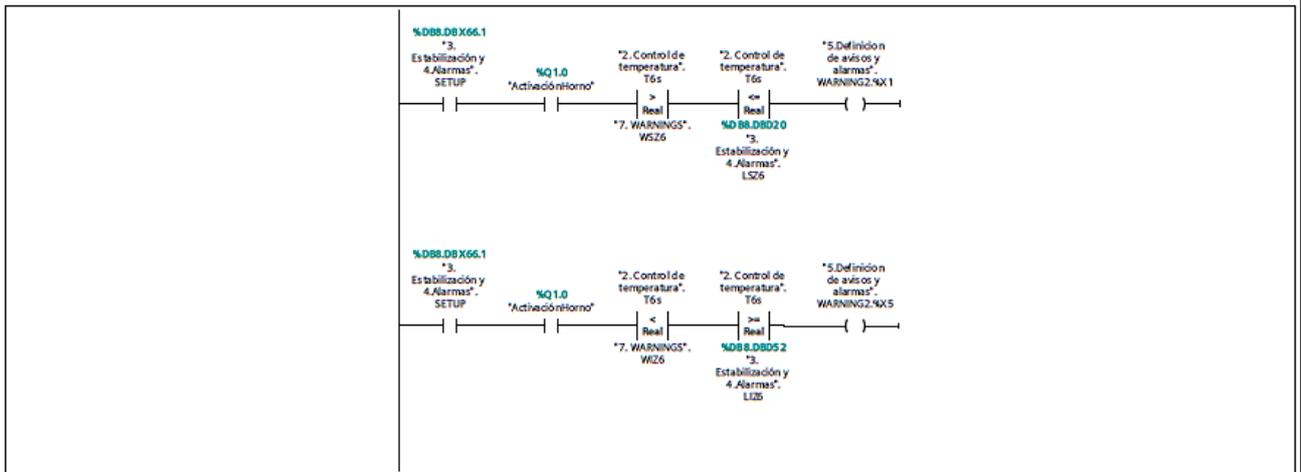
Totally Integrated Automation Portal



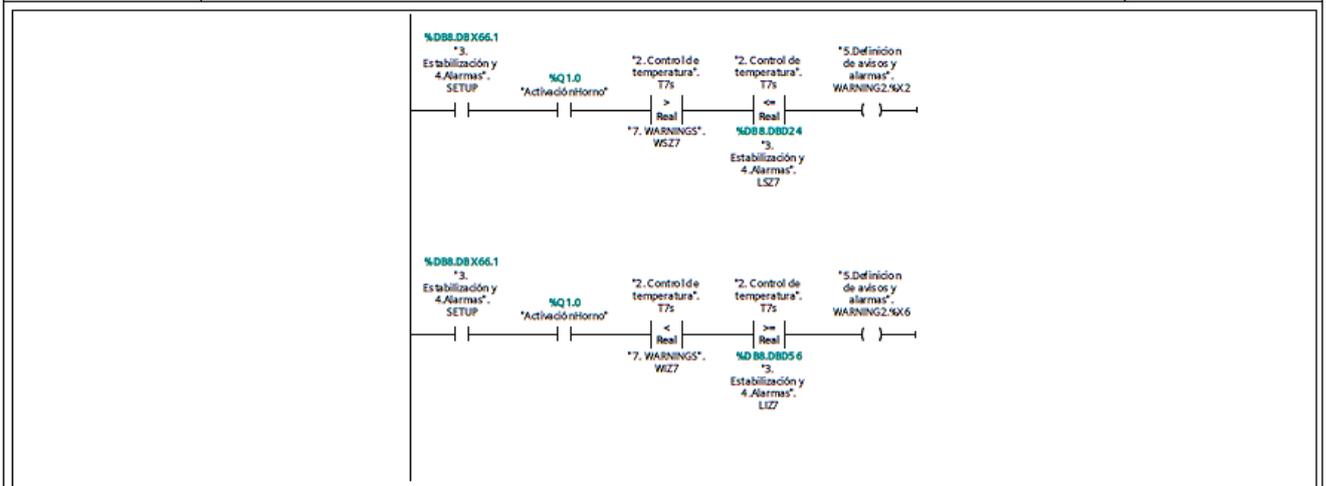
Segmento 6: ZONA 5



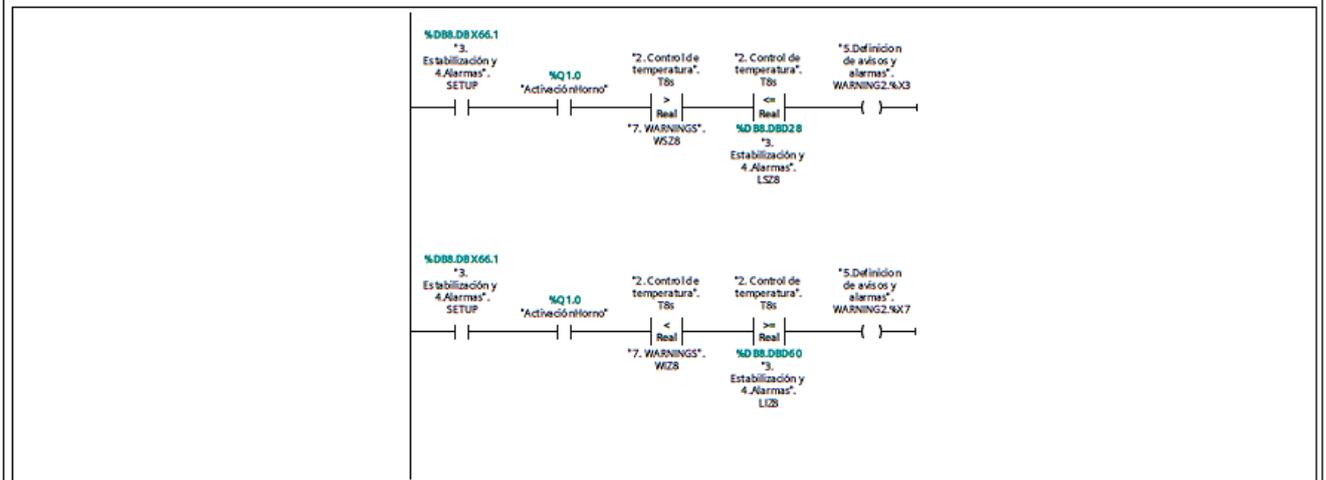
Segmento 7: ZONA 6



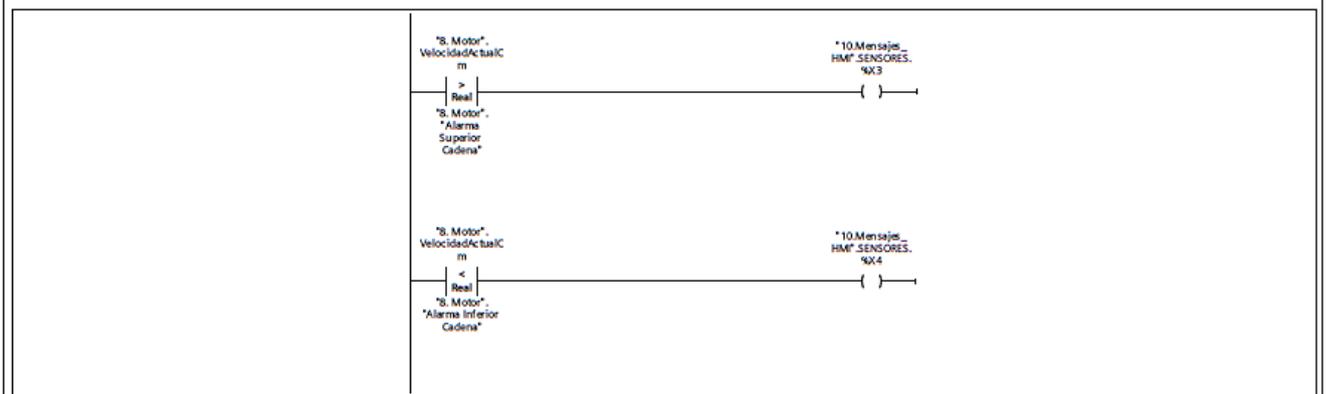
Segmento 8: ZONA 7



Segmento 9: ZONA 8



Segmento 10: CADENA



Totally Integrated Automation Portal		
--------------------------------------	--	--

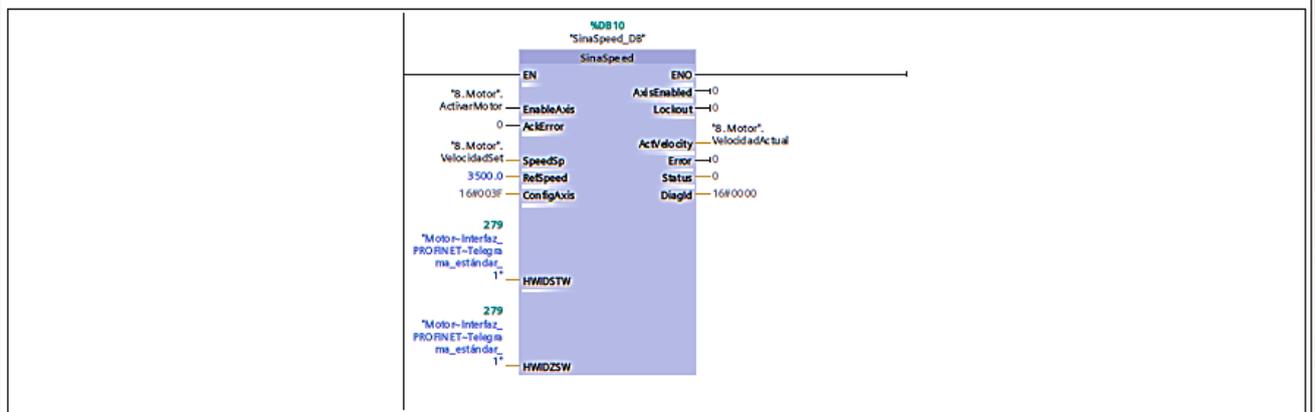
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

8.Motor [OB127]

8.Motor Propiedades							
General							
Nombre	8.Motor	Número	127	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

8.Motor			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
▼ Temp			
Posición_Pieza_Cadena_Int	Int		
Constant			

Segmento 1: BLOQUE SINASPEED QUE CONTROLA EL G120 PN



Segmento 2: PARÁMETRO PARA SETPOINT DE SINASPEED

TRAS CALCULAR LA VELOCIDAD EN CM/MIN DE LA CADENA EN FUNCIÓN DEL PARÁMETRO SETPOINT QUE PONEMOS EN EL BLOQUE SINASPEED, COMPROBAMOS QUE PARA TRANSFORMAR DE CM/MIN QUE INTRODUCIREMOS POR PANTALLA, TENEMOS QUE MULTIPLICAR POR -140.0 PARA CONSEGUIR EL SENTIDO DE GIRO CORRECTO Y EL PARÁMETRO ADECUADO PARA QUE DICHO BLOQUE NOS DE LA VELOCIDAD EN CM/MIN DESEADA.



Segmento 3: TRANSFORMACIÓN DE VARIABLE SALIDA SINASPEED A VELOCIDAD CM/MIN.

OPERACIÓN IDÉNTICA PERO A LA INVERSA PARA SABER QUE LA VELOCIDAD QUE TENEMOS ACTUALMENTE EN CM/MIN.



Totally Integrated Automation Portal	
--------------------------------------	--

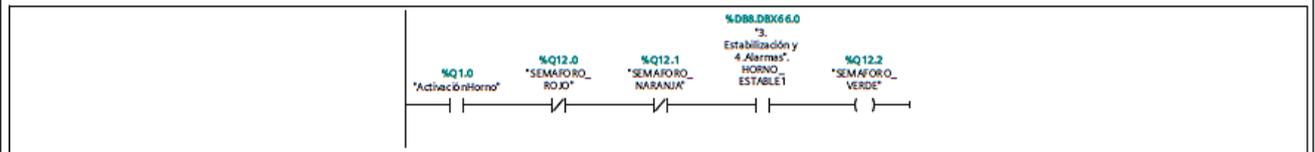
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

9. Apagado Horno y Señalización. [OB31]

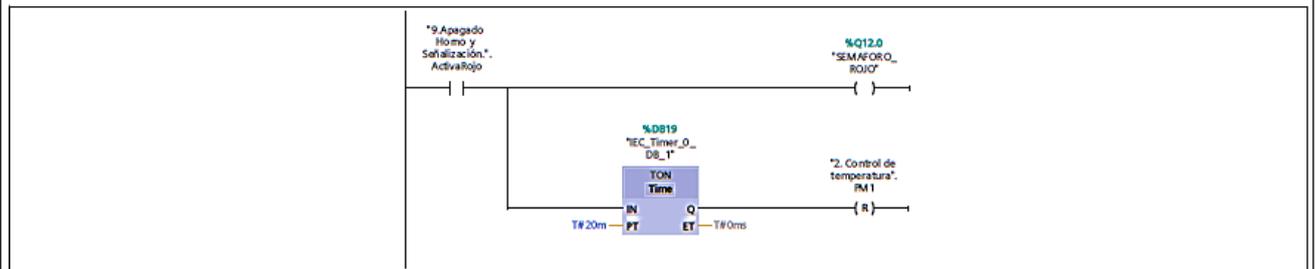
9. Apagado Horno y Señalización. Propiedades							
General							
Nombre	9. Apagado Horno y Señalización.	Número	31	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

9. Apagado Horno y Señalización.			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
▼ Temp			
bocina	Bool		
rojo	Bool		
algo	Bool		
Constant			

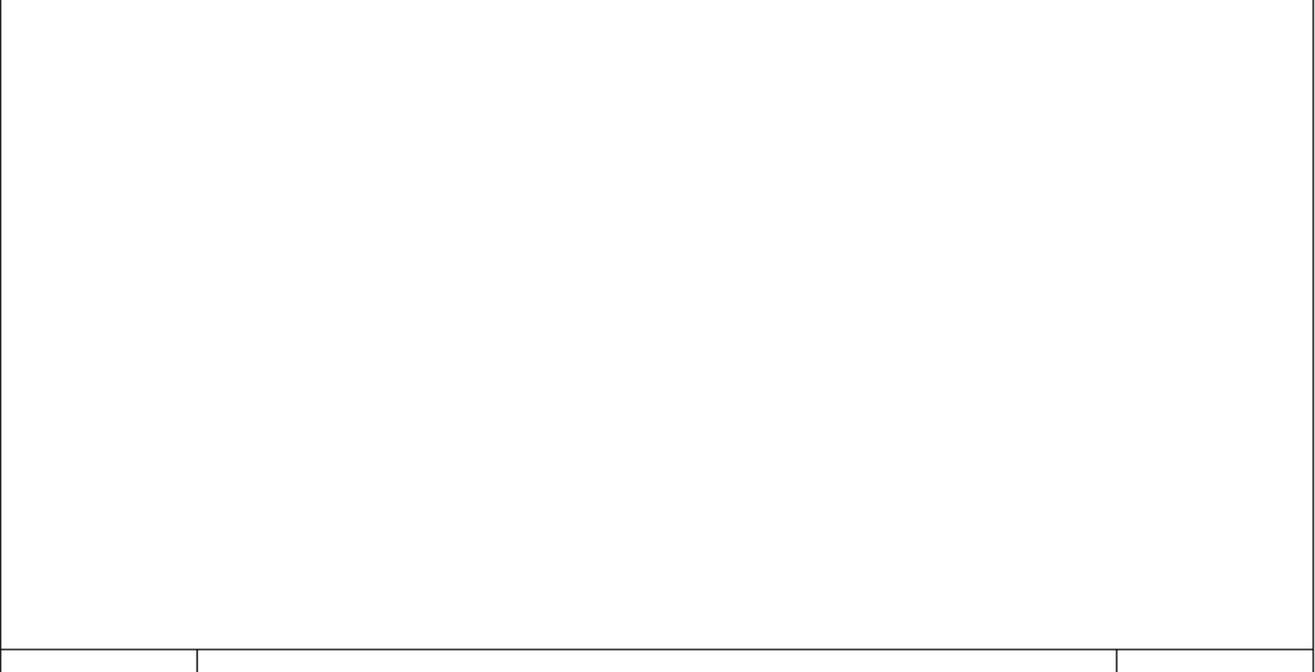
Segmento 1: SEMÁFORO VERDE

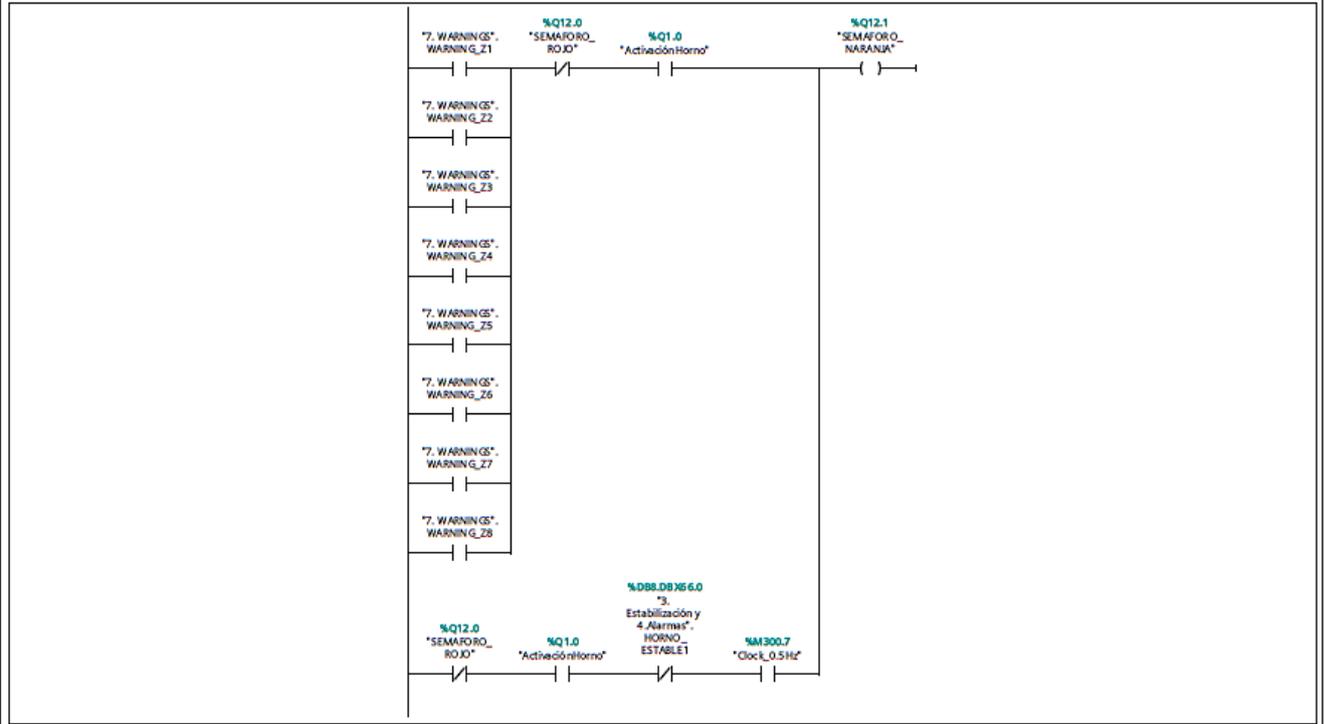


Segmento 2: SEMÁFORO ROJO



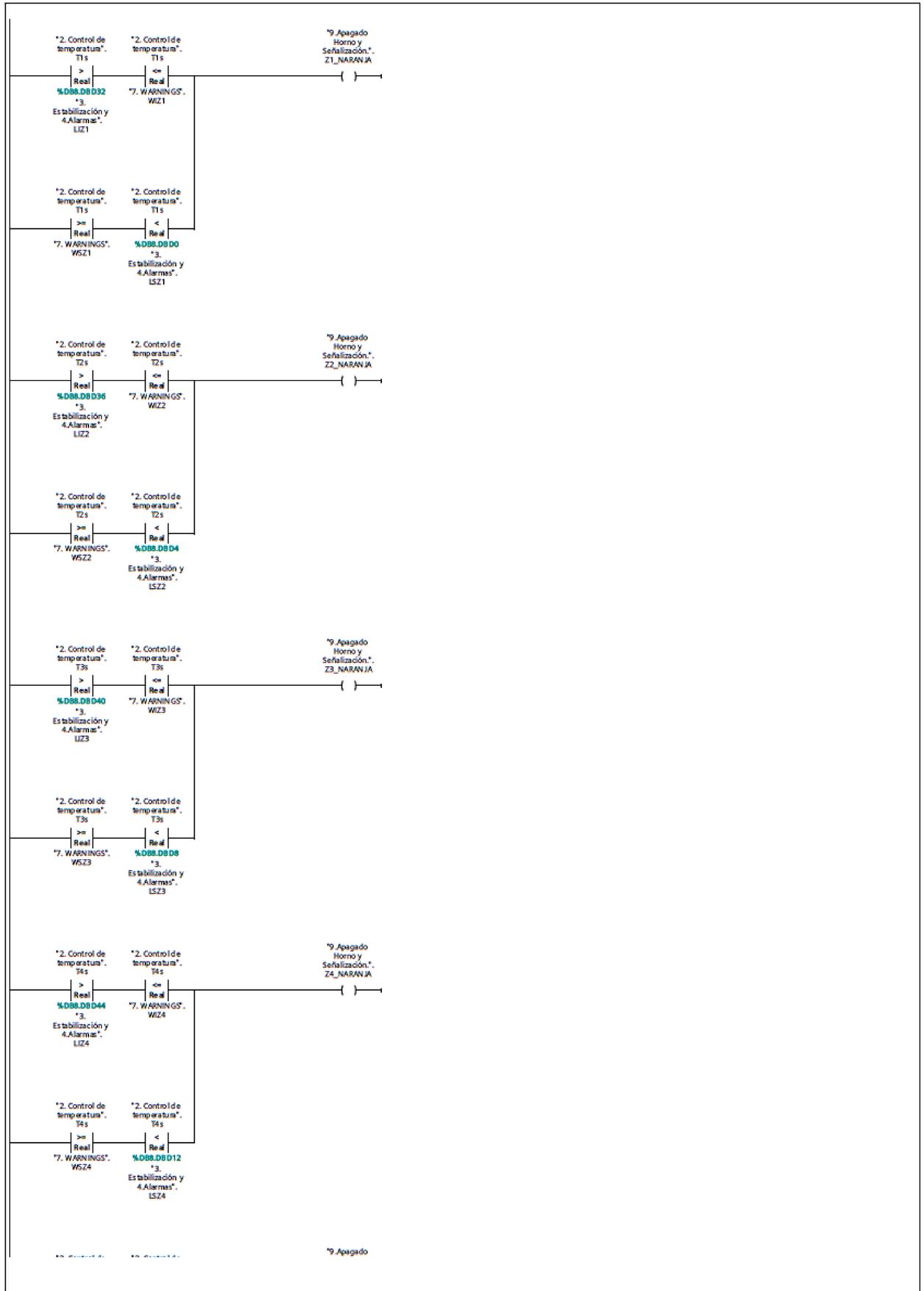
Segmento 3: SEMAFORO NARANJA





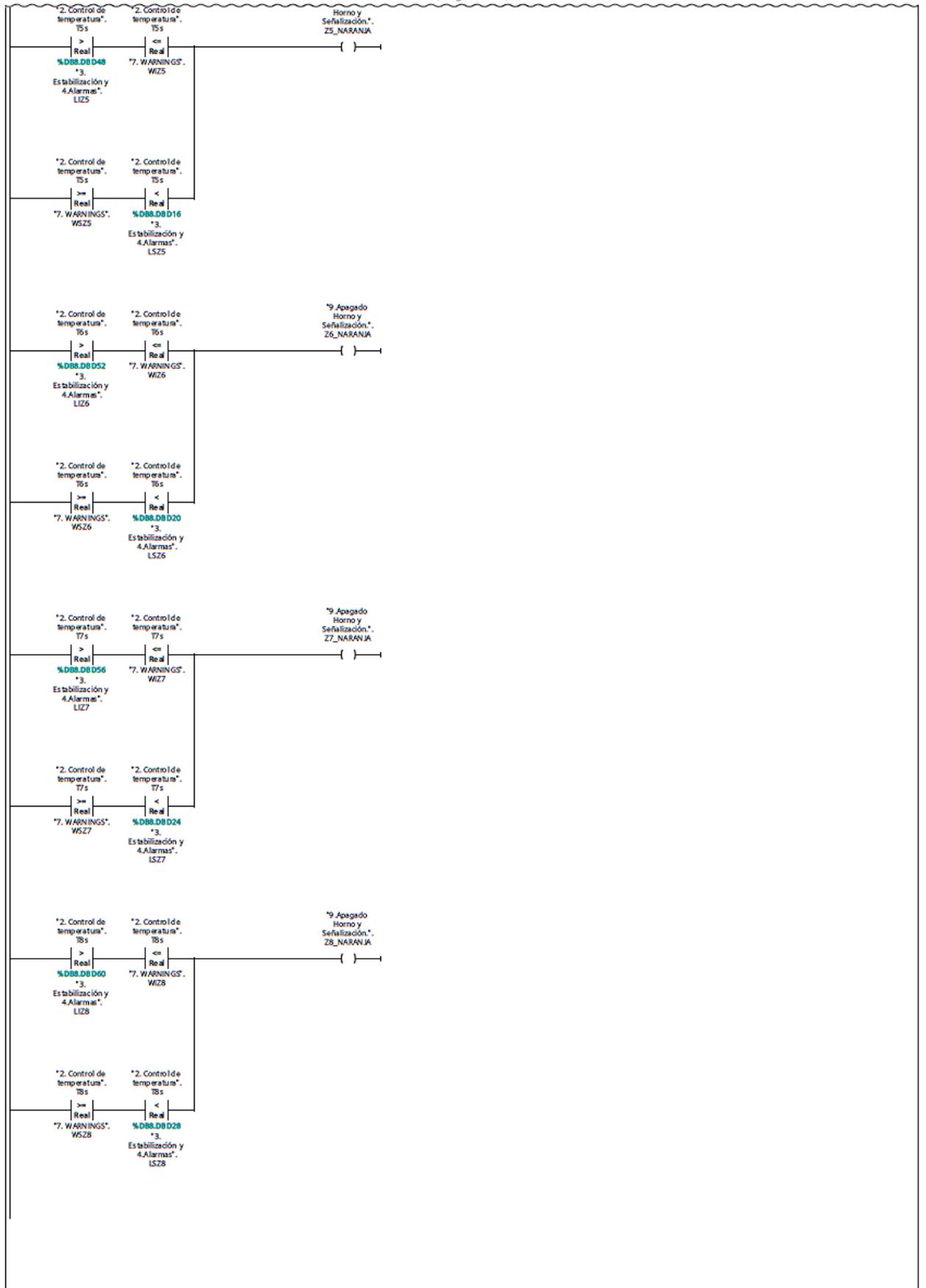
Segmento 4: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA NARANJA

Segmento 4: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA NARANJA (1.1 / 2.1)

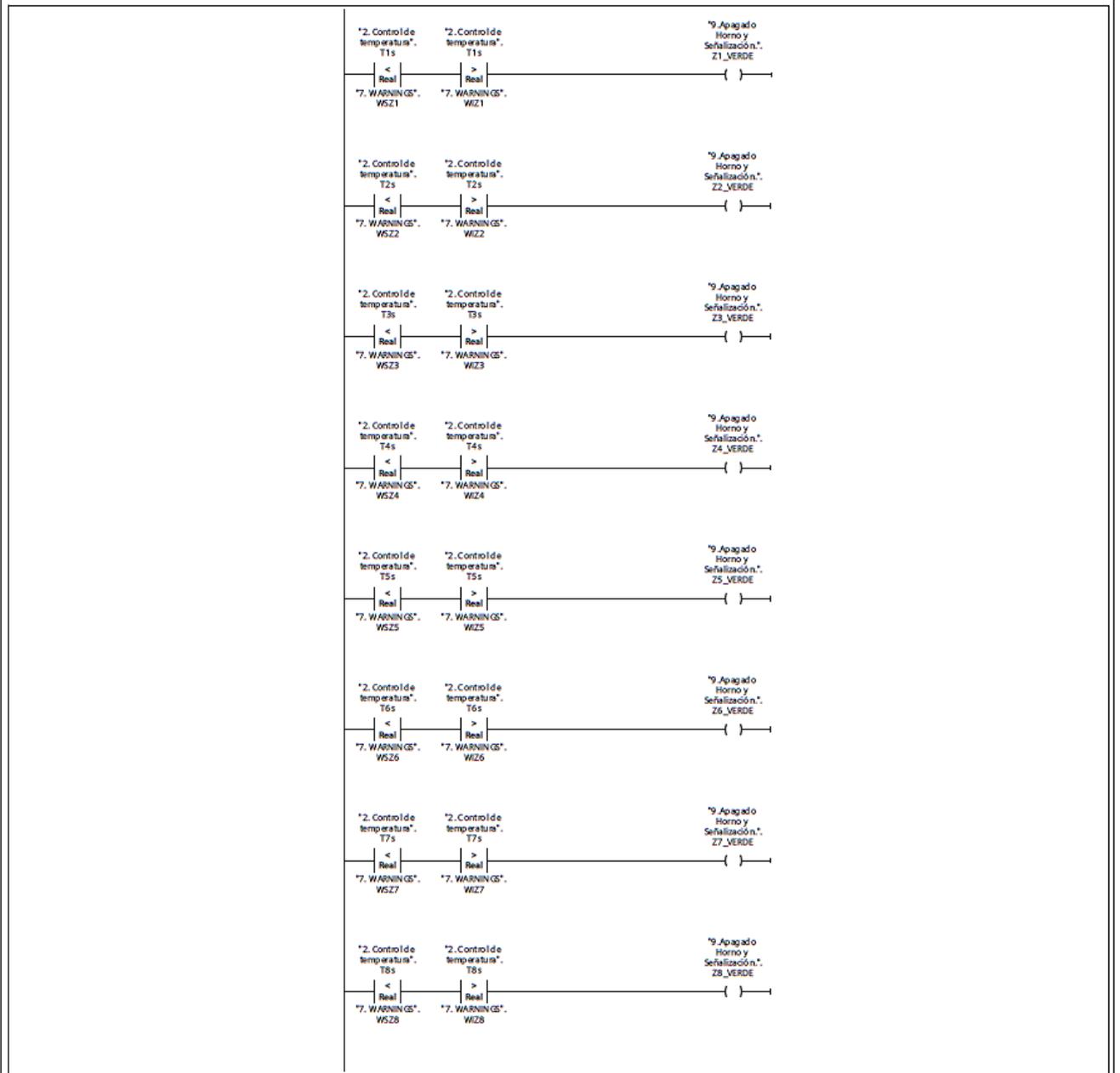


Segmento 4: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA NARANJA (2.1 / 2.1)

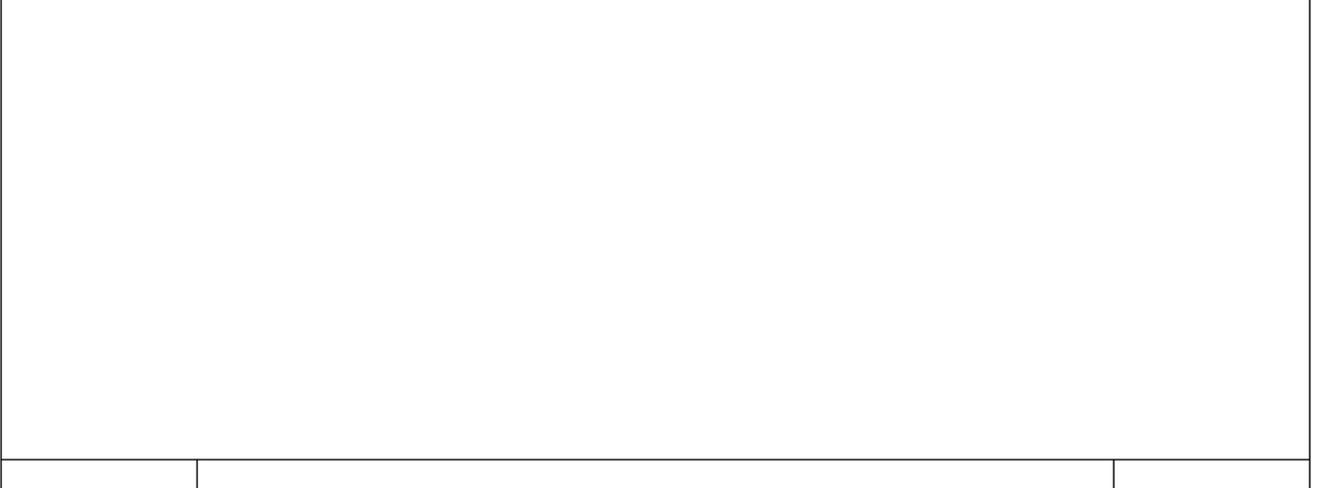
1.1 (Página1 - 3)



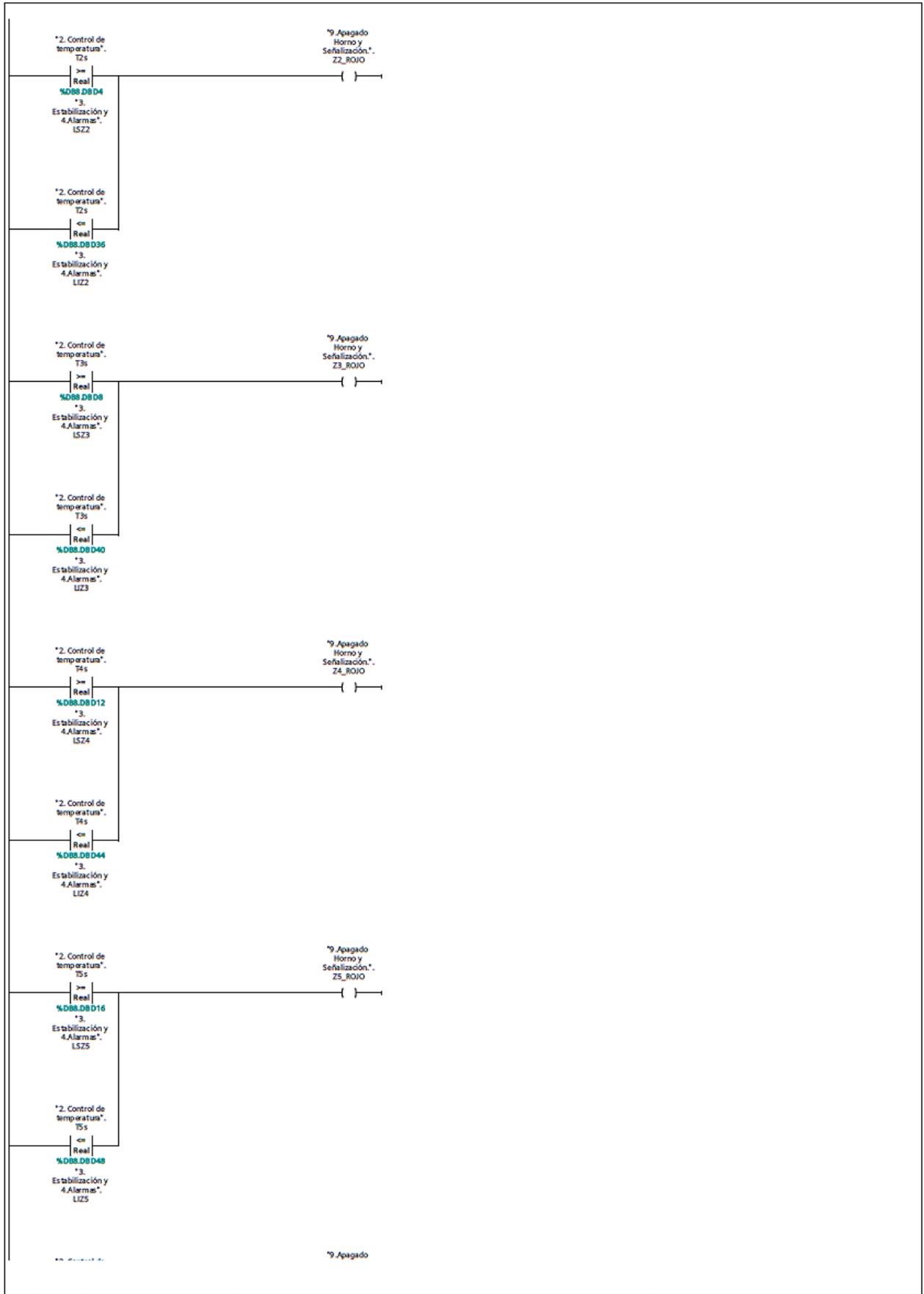
Segmento 5: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA VERDE



Segmento 6: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA ROJA

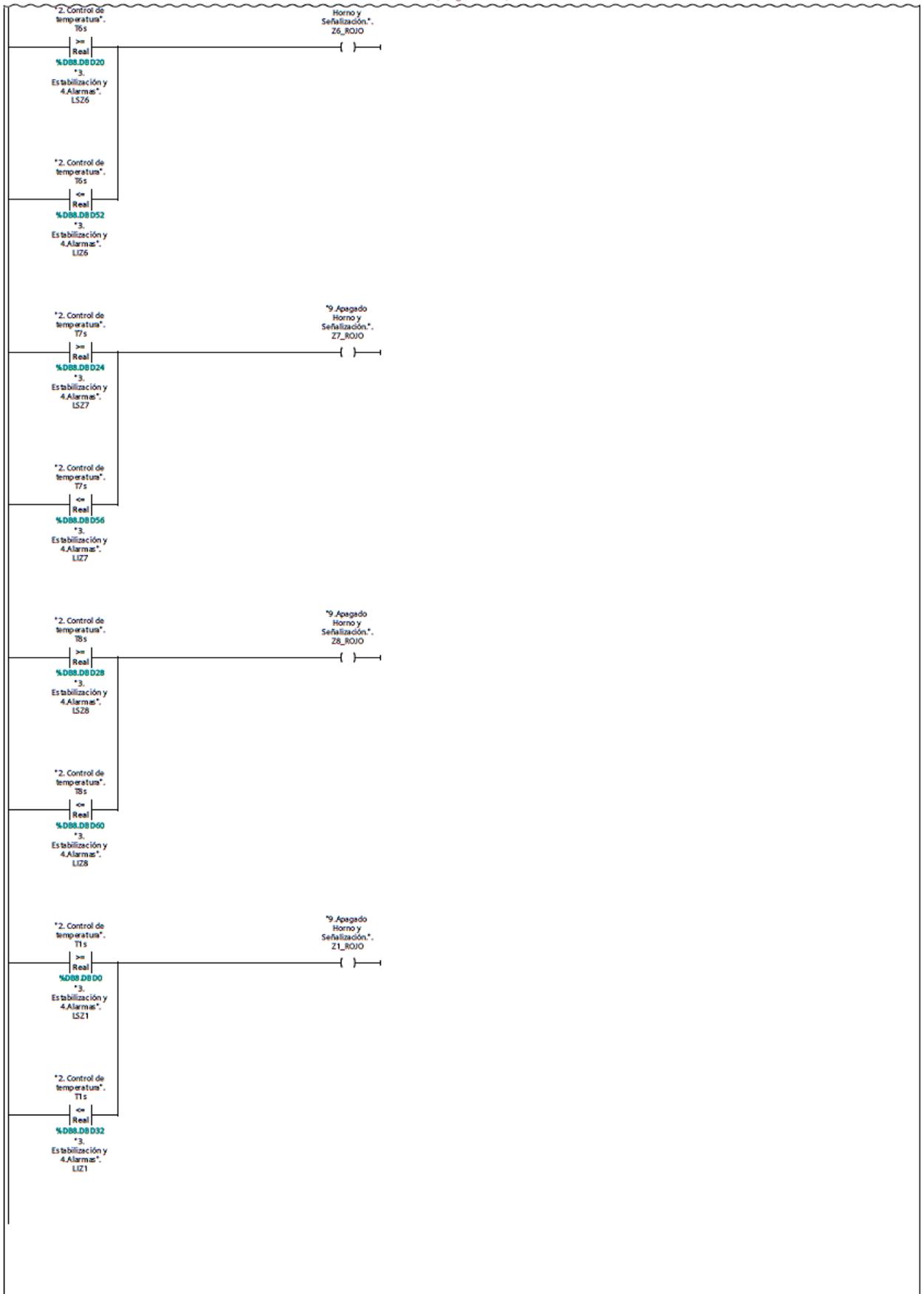


Segmento 6: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA ROJA (1.1 / 2.1)

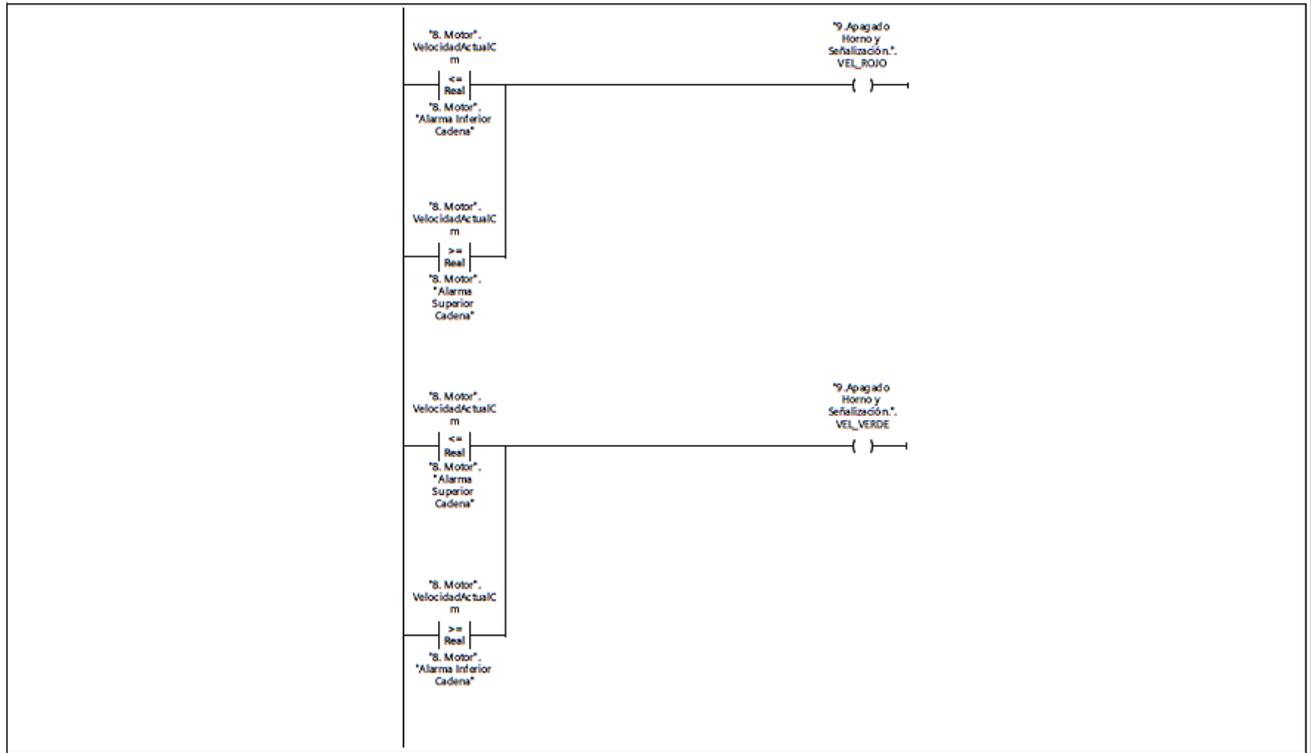


Segmento 6: CÓDIGO PARA SEÑALIZACIÓN TERMÓMETROS HMI ZONA ROJA (2.1 / 2.1)

1.1 (Página1 - 6)



Segmento 7: COLORES CADENA



Totally Integrated Automation Portal	
--------------------------------------	--

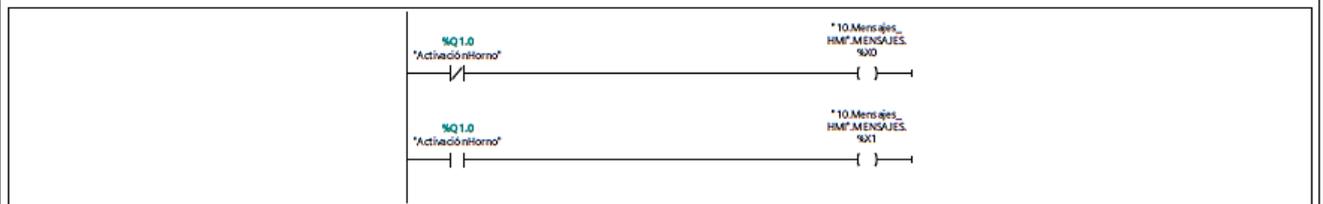
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC] / Bloques de programa

10. Mensajes_HMI [OB130]

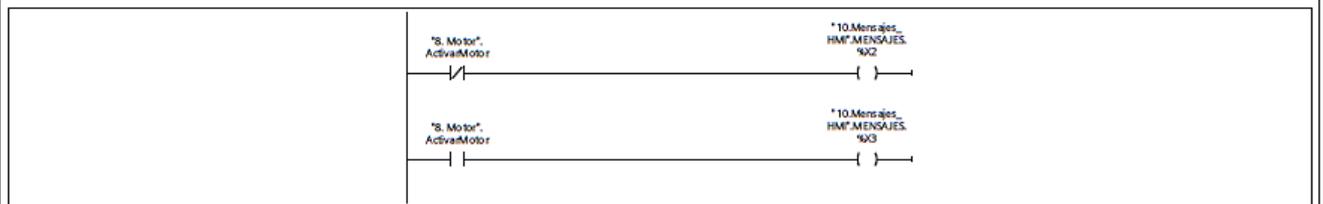
10. Mensajes_HMI Propiedades							
General							
Nombre	10. Mensajes_HMI	Número	130	Tipo	OB	Idioma	KOP
Numeración	Automático						
Información							
Título	"Main Program Sweep (Cycle)"	Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

10. Mensajes_HMI			
Nombre	Tipo de datos	Valor predet.	Comentario
▼ Input			
Inicia_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

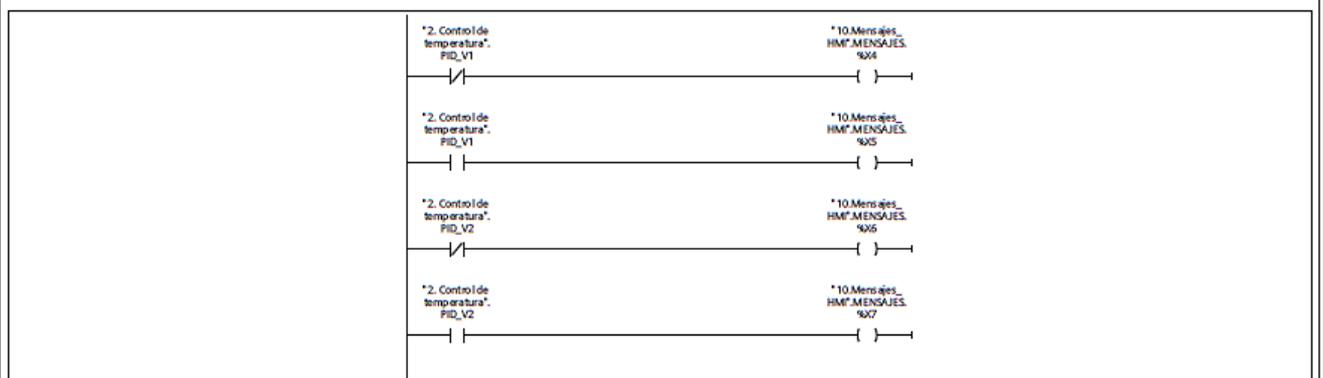
Segmento 1: MENSAJES CONTACTOR



Segmento 2: MENSAJES CADENA



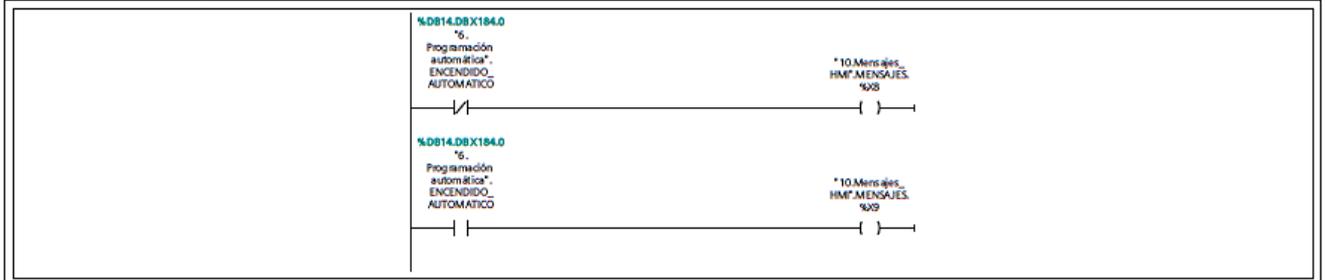
Segmento 3: MENSAJES MODOS DE POTENCIA



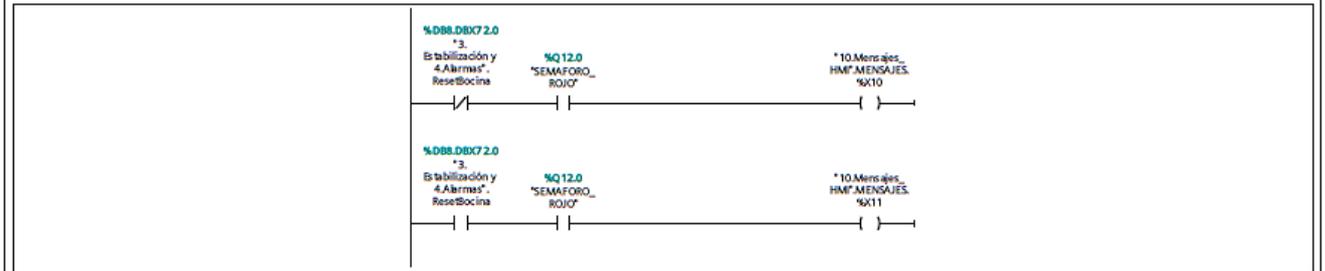
Segmento 4: MENSAJES PROGRAMACIÓN AUTOMÁTICA

--	--

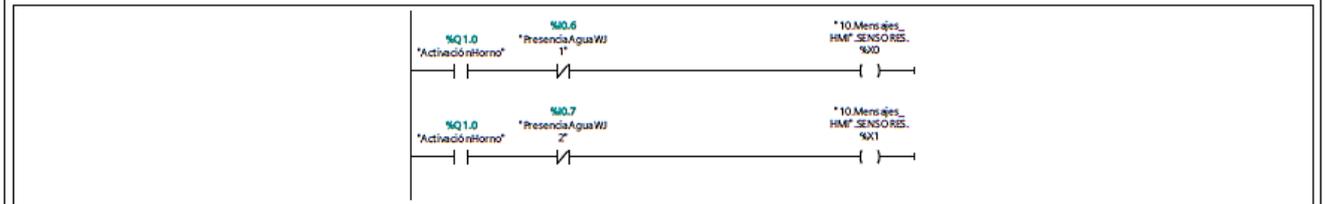
Totally Integrated Automation Portal		
--------------------------------------	--	--



Segmento 5: MENSAJES BOCINA



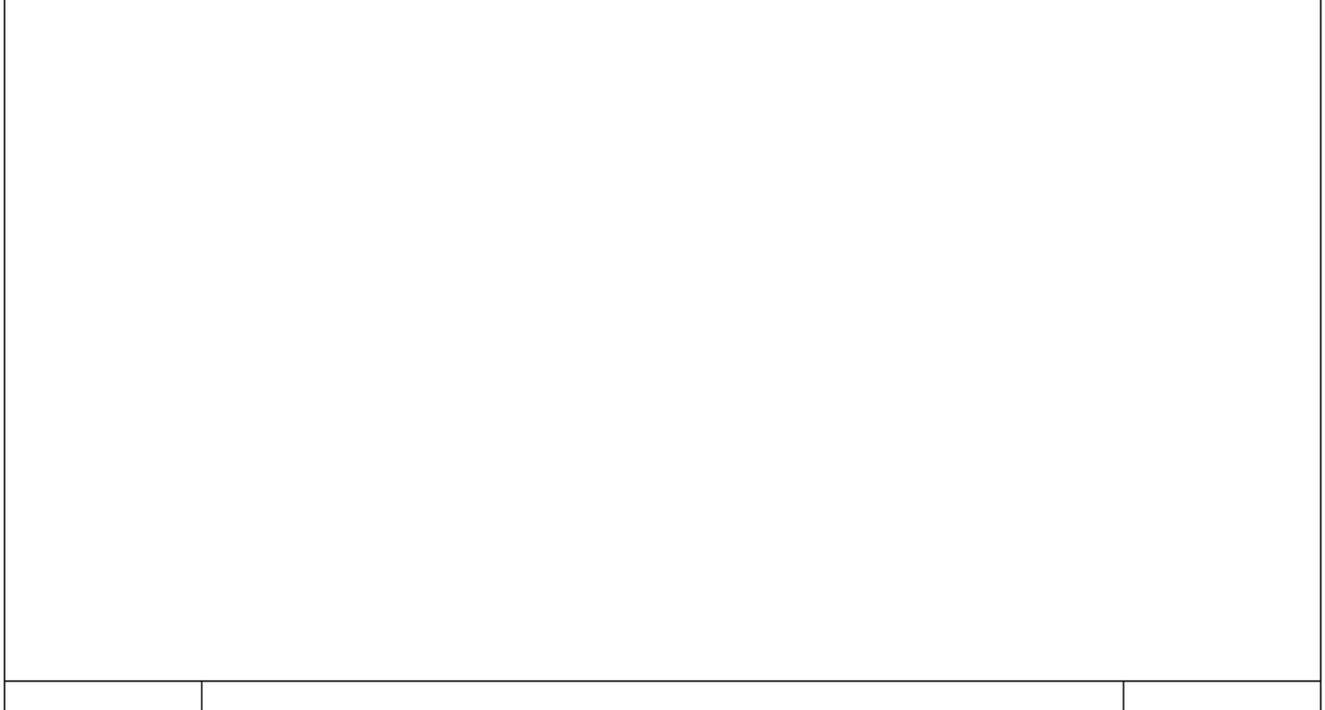
Segmento 6: MENSAJES FALLO SENSOR AGUA



Segmento 7: MENSAJES FALLO SENSOR AIRE



Segmento 8: ZONAS_OK



Totally Integrated Automation Portal		
	<p>*2. Control de temperatura*. T1s <= >= Real %DB8.DBD0 "3. Estabilización y 4.Alarmas". LSZ1</p>	<p>*2. Control de temperatura*. T1s >= <= Real %DB8.DBD32 "3. Estabilización y 4.Alarmas". UZ1</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X0</p>
	<p>*2. Control de temperatura*. T2s <= >= Real %DB8.DBD4 "3. Estabilización y 4.Alarmas". LSZ2</p>	<p>*2. Control de temperatura*. T2s >= <= Real %DB8.DBD36 "3. Estabilización y 4.Alarmas". UZ2</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X1</p>
	<p>*2. Control de temperatura*. T3s <= >= Real %DB8.DBD8 "3. Estabilización y 4.Alarmas". LSZ3</p>	<p>*2. Control de temperatura*. T3s >= <= Real %DB8.DBD40 "3. Estabilización y 4.Alarmas". UZ3</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X2</p>
	<p>*2. Control de temperatura*. T4s <= >= Real %DB8.DBD12 "3. Estabilización y 4.Alarmas". LSZ4</p>	<p>*2. Control de temperatura*. T4s >= <= Real %DB8.DBD44 "3. Estabilización y 4.Alarmas". UZ4</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X3</p>
	<p>*2. Control de temperatura*. T5s <= >= Real %DB8.DBD16 "3. Estabilización y 4.Alarmas". LSZ5</p>	<p>*2. Control de temperatura*. T5s >= <= Real %DB8.DBD48 "3. Estabilización y 4.Alarmas". UZ5</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X4</p>
	<p>*2. Control de temperatura*. T6s <= >= Real %DB8.DBD20 "3. Estabilización y 4.Alarmas". LSZ6</p>	<p>*2. Control de temperatura*. T6s >= <= Real %DB8.DBD52 "3. Estabilización y 4.Alarmas". UZ6</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X5</p>
	<p>*2. Control de temperatura*. T7s <= >= Real %DB8.DBD24 "3. Estabilización y 4.Alarmas". LSZ7</p>	<p>*2. Control de temperatura*. T7s >= <= Real %DB8.DBD56 "3. Estabilización y 4.Alarmas". UZ7</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X6</p>
	<p>*2. Control de temperatura*. T8s <= >= Real %DB8.DBD28 "3. Estabilización y 4.Alarmas". LSZ8</p>	<p>*2. Control de temperatura*. T8s >= <= Real %DB8.DBD60 "3. Estabilización y 4.Alarmas". UZ8</p> <p>*10.Mensajes_ HMF.ZON AS_ OK%X7</p>

Totally Integrated Automation Portal									
1.Puesta en Marcha [DB17]									
1.Puesta en Marcha Propiedades									
General									
Nombre	1.Puesta en Marcha	Número	17	Tipo	DB	Idioma	DB		
Numeración	Automático								
Información									
Título		Autor		Comentario		Familia			
Versión	0.1	ID personalizado							
Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
SAMPLE_TIME	Dint	0	False	True	True	True	False		
TIEMPO_ACTIVO	Dint	0	False	True	True	True	False		
RESET	Bool	false	False	True	True	True	False		
SALIDA	Bool	false	False	True	True	True	False		
PresenciaAguaOK	Bool	false	False	True	True	True	False		

Totally Integrated Automation Portal									
2. Control de temperatura [DB7]									
2. Control de temperatura Propiedades									
General									
Nombre	2. Control de temperatura	Número	7	Tipo	DB	Idioma	DB		
Numeración	Automático								
Información									
Título				Autor			Comentario		
Versión	0.1	ID personalizado							
Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
PM1	Bool	false	False	True	True	True	False		
PE_1	Bool	false	False	True	True	True	False		
SP5	Real	0.0	True	True	True	True	False		
SP6	Real	0.0	True	True	True	True	False		
SP7	Real	0.0	True	True	True	True	False		
SP8	Real	0.0	True	True	True	True	False		
T5s	Real	0.0	False	True	True	True	False		
T6s	Real	0.0	False	True	True	True	False		
T7s	Real	0.0	False	True	True	True	False		
T8s	Real	0.0	False	True	True	True	False		
SALIDA_PWM5	Bool	false	False	True	True	True	False		
SALIDA_PWM6	Bool	false	False	True	True	True	False		
SALIDA_PWM7	Bool	false	False	True	True	True	False		
SALIDA_PWM8	Bool	false	False	True	True	True	False		
PE	Bool	false	False	True	True	True	False		
SP1	Real	0.0	True	True	True	True	False		
SP2	Real	0.0	True	True	True	True	False		
SP3	Real	0.0	True	True	True	True	False		
SP4	Real	0.0	True	True	True	True	False		
T1s	Real	0.0	False	True	True	True	False		
T2s	Real	0.0	False	True	True	True	False		
T3s	Real	0.0	False	True	True	True	False		
T4s	Real	0.0	False	True	True	True	False		
SALIDA_PWM1	Bool	false	False	True	True	True	False		
SALIDA_PWM2	Bool	false	False	True	True	True	False		
SALIDA_PWM3	Bool	false	False	True	True	True	False		
SALIDA_PWM4	Bool	false	False	True	True	True	False		
PM_OK11	Bool	false	False	True	True	True	False		Activación de la señal de luz en verde HEAT HMI
HECHO	Bool	false	False	True	True	True	False		
Prueba	Bool	false	False	True	True	True	False		
STATUS	Word	16#0	False	True	True	True	False		
ERROR1	Bool	false	False	True	True	True	False		
Nombre_Perfil	WString	WSTRING#"	True	True	True	True	False		
PID_V1	Bool	false	True	True	True	True	False		
PID_V2	Bool	false	True	True	True	True	False		
PWM_Z1	Real	0.0	False	True	True	True	False		
PWM_Z2	Real	0.0	False	True	True	True	False		
PWM_Z3	Real	0.0	False	True	True	True	False		
PWM_Z4	Real	0.0	False	True	True	True	False		
PWM_Z5	Real	0.0	False	True	True	True	False		
PWM_Z6	Real	0.0	False	True	True	True	False		
PWM_Z7	Real	0.0	False	True	True	True	False		
PWM_Z8	Real	0.0	False	True	True	True	False		
RESET_PID	Bool	false	False	True	True	True	False		

Totally Integrated Automation Portal										
3. Estabilización y 4.Alarmas [DB8]										
3. Estabilización y 4.Alarmas Propiedades										
General										
Nombre	3. Estabilización y 4. Alarmas	Número	8	Tipo	DB	Idioma	DB			
Numeración	Automático									
Información										
Título				Autor				Comentario		
Versión	0.1	ID personalizado								
Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
LSZ1	Real	0.0	0.0	False	True	True	True	False		
LSZ2	Real	4.0	0.0	False	True	True	True	False		
LSZ3	Real	8.0	0.0	False	True	True	True	False		
LSZ4	Real	12.0	0.0	False	True	True	True	False		
LSZ5	Real	16.0	0.0	False	True	True	True	False		
LSZ6	Real	20.0	0.0	False	True	True	True	False		
LSZ7	Real	24.0	0.0	False	True	True	True	False		
LSZ8	Real	28.0	0.0	False	True	True	True	False		
LIZ1	Real	32.0	0.0	False	True	True	True	False		
LIZ2	Real	36.0	0.0	False	True	True	True	False		
LIZ3	Real	40.0	0.0	False	True	True	True	False		
LIZ4	Real	44.0	0.0	False	True	True	True	False		
LIZ5	Real	48.0	0.0	False	True	True	True	False		
LIZ6	Real	52.0	0.0	False	True	True	True	False		
LIZ7	Real	56.0	0.0	False	True	True	True	False		
LIZ8	Real	60.0	0.0	False	True	True	True	False		
ERROR_Z1	Bool	64.0	false	False	True	True	True	False		
ERROR_Z2	Bool	64.1	false	False	True	True	True	False		
ERROR_Z3	Bool	64.2	false	False	True	True	True	False		
ERROR_Z4	Bool	64.3	false	False	True	True	True	False		
ERROR_Z5	Bool	64.4	false	False	True	True	True	False		
ERROR_Z6	Bool	64.5	false	False	True	True	True	False		
ERROR_Z7	Bool	64.6	false	False	True	True	True	False		
ERROR_Z8	Bool	64.7	false	False	True	True	True	False		
ZONA1_OK	Bool	65.0	false	False	True	True	True	False		
ZONA2_OK	Bool	65.1	false	False	True	True	True	False		
ZONA3_OK	Bool	65.2	false	False	True	True	True	False		
ZONA4_OK	Bool	65.3	false	False	True	True	True	False		
ZONA5_OK	Bool	65.4	false	False	True	True	True	False		
ZONA6_OK	Bool	65.5	false	False	True	True	True	False		
ZONA7_OK	Bool	65.6	false	False	True	True	True	False		
ZONA8_OK	Bool	65.7	false	False	True	True	True	False		
HORNO_ESTABLE1	Bool	66.0	false	False	True	True	True	False		
SETUP	Bool	66.1	false	False	True	True	True	False		
T_ON2	Time	68.0	T#1h30m	False	True	True	True	False		
ResetBocina	Bool	72.0	false	False	True	True	True	False		
Error_Crítico	Bool	72.1	false	False	True	True	True	False		
Crítico	Word	74.0	16#0	False	True	True	True	False		

Totally Integrated Automation Portal											
5. Definición de avisos y alarmas [DB12]											
5. Definición de avisos y alarmas Propiedades											
General											
Nombre	5. Definición de avisos y alarmas	Número	12	Tipo	DB	Idioma	DB				
Numeración	Automático										
Información											
Título				Autor				Comentario			
Versión	0.1	ID personalizado									
Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario	
▼ Static											
ALARMAS1	Word	0.0	16#0	False	True	True	True	False		BYTE DE ACTIVACIÓN DE ALARMAS (0) Y BYTE DE ACTIVACION DE ACUSES (1)	
ALARMAS2	Word	2.0	16#0	False	True	True	True	False		BYTE DE ACUSE DEL PLC (3) Y BYTE DE ACUSE DEL HMI (2)	
WARNING1	Word	4.0	16#0	False	True	True	True	False			
WARNING2	Word	6.0	16#0	False	True	True	True	False			
ACUSE1	Word	8.0	16#0	False	True	True	True	False			
ACUSE2	Word	10.0	16#0	False	True	True	True	False			
AcuseCadena	Bool	12.0	false	False	True	True	True	False			

Totally Integrated Automation Portal									
7. WARNINGS [DB6]									
7. WARNINGS Propiedades									
General									
Nombre	7. WARNINGS	Número	6	Tipo	DB	Idioma	DB		
Numeración	Automático								
Información									
Título		Autor		Comentario		Familia			
Versión	0.1	ID personalizado							
Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
WSZ1	Real	0.0	True	True	True	True	False		
WSZ2	Real	0.0	True	True	True	True	False		
WSZ3	Real	0.0	True	True	True	True	False		
WSZ4	Real	0.0	True	True	True	True	False		
WSZ5	Real	0.0	True	True	True	True	False		
WSZ6	Real	0.0	True	True	True	True	False		
WSZ7	Real	0.0	True	True	True	True	False		
WSZ8	Real	0.0	True	True	True	True	False		
WIZ1	Real	0.0	True	True	True	True	False		
WIZ2	Real	0.0	True	True	True	True	False		
WIZ3	Real	0.0	True	True	True	True	False		
WIZ4	Real	0.0	True	True	True	True	False		
WIZ5	Real	0.0	True	True	True	True	False		
WIZ6	Real	0.0	True	True	True	True	False		
WIZ7	Real	0.0	True	True	True	True	False		
WIZ8	Real	0.0	True	True	True	True	False		
WARNING_Z1	Bool	false	False	True	True	True	False		
WARNING_Z2	Bool	false	False	True	True	True	False		
WARNING_Z3	Bool	false	False	True	True	True	False		
WARNING_Z4	Bool	false	False	True	True	True	False		
WARNING_Z5	Bool	false	False	True	True	True	False		
WARNING_Z6	Bool	false	False	True	True	True	False		
WARNING_Z7	Bool	false	False	True	True	True	False		
WARNING_Z8	Bool	false	False	True	True	True	False		

Totally Integrated Automation Portal	
--------------------------------------	--

6.Programación automática [DB14]

6.Programación automática Propiedades

General							
Nombre	6.Programación automática	Número	14	Tipo	DB	Idioma	DB
Numeración	Automático						

Información

Título		Autor		Comentario		Familia	
Versión	0.1	ID personalizado					

Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HM/OPC UA/Web API	Escribible desde HM/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
▼ Tiempos_CPU	DTL	0.0	DTL#1970-01-01-00:00:00	True	True	True	True	True		
YEAR	UInt	0.0	1970	True	True	True	True	True		
MONTH	UInt	2.0	1	True	True	True	True	True		
DAY	UInt	3.0	1	True	True	True	True	True		
WEEKDAY	UInt	4.0	5	True	True	True	True	True		
HOUR	UInt	5.0	0	True	True	True	True	True		
MINUTE	UInt	6.0	0	True	True	True	True	True		
SECOND	UInt	7.0	0	True	True	True	True	True		
NANOSECOND	UInt	8.0	0	True	True	True	True	True		
STATUS_RELOJ	Word	12.0	16#0	True	True	True	True	False		
LUNES	Bool	14.0	false	True	True	True	True	False		
MARTES	Bool	14.1	false	True	True	True	True	False		
MIERCOLES	Bool	14.2	false	True	True	True	True	False		
JUEVES	Bool	14.3	false	True	True	True	True	False		
VIERNES	Bool	14.4	false	True	True	True	True	False		
SABADO	Bool	14.5	false	True	True	True	True	False		
DOMINGO	Bool	14.6	false	True	True	True	True	False		
▼ TIEMPOS_LUNES_ON	DTL	16.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	16.0	1970	True	True	True	True	False		
MONTH	UInt	18.0	1	True	True	True	True	False		
DAY	UInt	19.0	1	True	True	True	True	False		
WEEKDAY	UInt	20.0	5	True	True	True	True	False		
HOUR	UInt	21.0	0	True	True	True	True	False		
MINUTE	UInt	22.0	0	True	True	True	True	False		
SECOND	UInt	23.0	0	True	True	True	True	False		
NANOSECOND	UInt	24.0	0	True	True	True	True	False		
▼ TIEMPOS_LUNES_OFF	DTL	28.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	28.0	1970	True	True	True	True	False		
MONTH	UInt	30.0	1	True	True	True	True	False		
DAY	UInt	31.0	1	True	True	True	True	False		
WEEKDAY	UInt	32.0	5	True	True	True	True	False		
HOUR	UInt	33.0	0	True	True	True	True	False		
MINUTE	UInt	34.0	0	True	True	True	True	False		
SECOND	UInt	35.0	0	True	True	True	True	False		
NANOSECOND	UInt	36.0	0	True	True	True	True	False		
▼ TIEMPOS_MARTES_ON	DTL	40.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	40.0	1970	True	True	True	True	False		
MONTH	UInt	42.0	1	True	True	True	True	False		
DAY	UInt	43.0	1	True	True	True	True	False		
WEEKDAY	UInt	44.0	5	True	True	True	True	False		
HOUR	UInt	45.0	0	True	True	True	True	False		
MINUTE	UInt	46.0	0	True	True	True	True	False		
SECOND	UInt	47.0	0	True	True	True	True	False		
NANOSECOND	UInt	48.0	0	True	True	True	True	False		
▼ TIEMPOS_MARTES_OFF	DTL	52.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	52.0	1970	True	True	True	True	False		
MONTH	UInt	54.0	1	True	True	True	True	False		
DAY	UInt	55.0	1	True	True	True	True	False		
WEEKDAY	UInt	56.0	5	True	True	True	True	False		
HOUR	UInt	57.0	0	True	True	True	True	False		
MINUTE	UInt	58.0	0	True	True	True	True	False		
SECOND	UInt	59.0	0	True	True	True	True	False		
NANOSECOND	UInt	60.0	0	True	True	True	True	False		
▼ TIEMPOS_MIERCOLES_ON	DTL	64.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	64.0	1970	True	True	True	True	False		
MONTH	UInt	66.0	1	True	True	True	True	False		
DAY	UInt	67.0	1	True	True	True	True	False		

Totally Integrated Automation Portal										
Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
WEEKDAY	USInt	68.0	5	True	True	True	True	False		
HOUR	USInt	69.0	0	True	True	True	True	False		
MINUTE	USInt	70.0	0	True	True	True	True	False		
SECOND	USInt	71.0	0	True	True	True	True	False		
NANOSECOND	UDInt	72.0	0	True	True	True	True	False		
▼ TIEMPOS_MIERCOLES_OFF	DTL	76.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	76.0	1970	True	True	True	True	False		
MONTH	USInt	78.0	1	True	True	True	True	False		
DAY	USInt	79.0	1	True	True	True	True	False		
WEEKDAY	USInt	80.0	5	True	True	True	True	False		
HOUR	USInt	81.0	0	True	True	True	True	False		
MINUTE	USInt	82.0	0	True	True	True	True	False		
SECOND	USInt	83.0	0	True	True	True	True	False		
NANOSECOND	UDInt	84.0	0	True	True	True	True	False		
▼ TIEMPOS_JUEVES_ON	DTL	88.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	88.0	1970	True	True	True	True	False		
MONTH	USInt	90.0	1	True	True	True	True	False		
DAY	USInt	91.0	1	True	True	True	True	False		
WEEKDAY	USInt	92.0	5	True	True	True	True	False		
HOUR	USInt	93.0	0	True	True	True	True	False		
MINUTE	USInt	94.0	0	True	True	True	True	False		
SECOND	USInt	95.0	0	True	True	True	True	False		
NANOSECOND	UDInt	96.0	0	True	True	True	True	False		
▼ TIEMPOS_JUEVES_OFF	DTL	100.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	100.0	1970	True	True	True	True	False		
MONTH	USInt	102.0	1	True	True	True	True	False		
DAY	USInt	103.0	1	True	True	True	True	False		
WEEKDAY	USInt	104.0	5	True	True	True	True	False		
HOUR	USInt	105.0	0	True	True	True	True	False		
MINUTE	USInt	106.0	0	True	True	True	True	False		
SECOND	USInt	107.0	0	True	True	True	True	False		
NANOSECOND	UDInt	108.0	0	True	True	True	True	False		
▼ TIEMPOS_VIERNES_ON	DTL	112.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	112.0	1970	True	True	True	True	False		
MONTH	USInt	114.0	1	True	True	True	True	False		
DAY	USInt	115.0	1	True	True	True	True	False		
WEEKDAY	USInt	116.0	5	True	True	True	True	False		
HOUR	USInt	117.0	0	True	True	True	True	False		
MINUTE	USInt	118.0	0	True	True	True	True	False		
SECOND	USInt	119.0	0	True	True	True	True	False		
NANOSECOND	UDInt	120.0	0	True	True	True	True	False		
▼ TIEMPOS_VIERNES_OFF	DTL	124.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	124.0	1970	True	True	True	True	False		
MONTH	USInt	126.0	1	True	True	True	True	False		
DAY	USInt	127.0	1	True	True	True	True	False		
WEEKDAY	USInt	128.0	5	True	True	True	True	False		
HOUR	USInt	129.0	0	True	True	True	True	False		
MINUTE	USInt	130.0	0	True	True	True	True	False		
SECOND	USInt	131.0	0	True	True	True	True	False		
NANOSECOND	UDInt	132.0	0	True	True	True	True	False		
▼ TIEMPOS_SABADO_ON	DTL	136.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	136.0	1970	True	True	True	True	False		
MONTH	USInt	138.0	1	True	True	True	True	False		
DAY	USInt	139.0	1	True	True	True	True	False		
WEEKDAY	USInt	140.0	5	True	True	True	True	False		
HOUR	USInt	141.0	0	True	True	True	True	False		
MINUTE	USInt	142.0	0	True	True	True	True	False		
SECOND	USInt	143.0	0	True	True	True	True	False		
NANOSECOND	UDInt	144.0	0	True	True	True	True	False		
▼ TIEMPOS_SABADO_OFF	DTL	148.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	148.0	1970	True	True	True	True	False		
MONTH	USInt	150.0	1	True	True	True	True	False		
DAY	USInt	151.0	1	True	True	True	True	False		
WEEKDAY	USInt	152.0	5	True	True	True	True	False		
HOUR	USInt	153.0	0	True	True	True	True	False		
MINUTE	USInt	154.0	0	True	True	True	True	False		
SECOND	USInt	155.0	0	True	True	True	True	False		
NANOSECOND	UDInt	156.0	0	True	True	True	True	False		
▼ TIEMPOS_DOMINGO_ON	DTL	160.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		

Totally Integrated Automation Portal										
Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
YEAR	UInt	160.0	1970	True	True	True	True	False		
MONTH	USInt	162.0	1	True	True	True	True	False		
DAY	USInt	163.0	1	True	True	True	True	False		
WEEKDAY	USInt	164.0	5	True	True	True	True	False		
HOUR	USInt	165.0	0	True	True	True	True	False		
MINUTE	USInt	166.0	0	True	True	True	True	False		
SECOND	USInt	167.0	0	True	True	True	True	False		
NANOSECOND	UDInt	168.0	0	True	True	True	True	False		
▼ TIEMPOS_DOMINGO_OFF	DTL	172.0	DTL#1970-01-01-00:00:00	True	True	True	True	False		
YEAR	UInt	172.0	1970	True	True	True	True	False		
MONTH	USInt	174.0	1	True	True	True	True	False		
DAY	USInt	175.0	1	True	True	True	True	False		
WEEKDAY	USInt	176.0	5	True	True	True	True	False		
HOUR	USInt	177.0	0	True	True	True	True	False		
MINUTE	USInt	178.0	0	True	True	True	True	False		
SECOND	USInt	179.0	0	True	True	True	True	False		
NANOSECOND	UDInt	180.0	0	True	True	True	True	False		
ENCENDIDO_AUTOMATICO	Bool	184.0	false	True	True	True	True	False		
ON	Bool	184.1	false	True	True	True	True	False		
Modo_Calentamiento_Lunes	Bool	184.2	false	True	True	True	True	False		
Modo_Calentamiento_Martes	Bool	184.3	false	True	True	True	True	False		
Modo_Calentamiento_Miercoles	Bool	184.4	false	True	True	True	True	False		
Modo_Calentamiento_Jueves	Bool	184.5	false	True	True	True	True	False		
Modo_Calentamiento_Viernes	Bool	184.6	false	True	True	True	True	False		
Modo_Calentamiento_Sabado	Bool	184.7	false	True	True	True	True	False		
Modo_Calentamiento_Domingo	Bool	185.0	false	True	True	True	True	False		

Totally Integrated Automation Portal									
8. Motor [DB11]									
8. Motor Propiedades									
General									
Nombre	8. Motor	Número	11	Tipo	DB	Idioma	DB		
Numeración	Automático								
Información									
Título		Autor		Comentario		Familia			
Versión	0.1	ID personalizado							
Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
ActivarMotor	Bool	false	False	True	True	True	False		
AcuseErrorMotor	Bool	false	False	True	True	True	False		
VelocidadSet	Real	0.0	True	True	True	True	False		
VelocidadActual	Real	0.0	False	True	True	True	False		
VelocidadActualCm	Real	0.0	False	True	True	True	False		
VelocidadSetCm	Real	0.0	True	True	True	True	False		
Alarma Superior Cadena	Real	0.0	True	True	True	True	False		
Alarma Inferior Cadena	Real	0.0	True	True	True	True	False		

Totally Integrated Automation Portal									
9.Apagado Horno y Señalización. [DB5]									
9.Apagado Horno y Señalización. Propiedades									
General									
Nombre	9.Apagado Horno y Señalización.	Número	5	Tipo	DB	Idioma	DB		
Numeración	Automático								
Información									
Título		Autor		Comentario		Familia			
Versión	0.1	ID personalizado							
Nombre	Tipo de datos	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static									
Z1_NARANJA	Bool	false	False	True	True	True	False		
Z2_NARANJA	Bool	false	False	True	True	True	False		
Z3_NARANJA	Bool	false	False	True	True	True	False		
Z4_NARANJA	Bool	false	False	True	True	True	False		
Z5_NARANJA	Bool	false	False	True	True	True	False		
Z6_NARANJA	Bool	false	False	True	True	True	False		
Z7_NARANJA	Bool	false	False	True	True	True	False		
Z8_NARANJA	Bool	false	False	True	True	True	False		
Z1_VERDE	Bool	false	False	True	True	True	False		
Z2_VERDE	Bool	false	False	True	True	True	False		
Z3_VERDE	Bool	false	False	True	True	True	False		
Z4_VERDE	Bool	false	False	True	True	True	False		
Z5_VERDE	Bool	false	False	True	True	True	False		
Z6_VERDE	Bool	false	False	True	True	True	False		
Z7_VERDE	Bool	false	False	True	True	True	False		
Z8_VERDE	Bool	false	False	True	True	True	False		
Z1_ROJO	Bool	false	False	True	True	True	False		
Z2_ROJO	Bool	false	False	True	True	True	False		
Z3_ROJO	Bool	false	False	True	True	True	False		
Z4_ROJO	Bool	false	False	True	True	True	False		
Z5_ROJO	Bool	false	False	True	True	True	False		
Z6_ROJO	Bool	false	False	True	True	True	False		
Z7_ROJO	Bool	false	False	True	True	True	False		
Z8_ROJO	Bool	false	False	True	True	True	False		
VEL_VERDE	Bool	false	False	True	True	True	False		
VEL_ROJO	Bool	false	False	True	True	True	False		
ActivaRojo	Bool	false	False	True	True	True	False		

Totally Integrated Automation Portal										
10.Mensajes_HMI [DB28]										
10.Mensajes_HMI Propiedades										
General										
Nombre	10.Mensajes_HMI	Número	28	Tipo	DB	Idioma	DB			
Numeración	Automático									
Información										
Título		Autor		Comentario		Familia				
Versión	0.1	ID personalizado								
Nombre	Tipo de datos	Offset	Valor de arranque	Remanencia	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Valor de ajuste	Supervisión	Comentario
▼ Static										
MENSAJES	Word	0.0	16#0	False	True	True	True	False		
SENSORES	Word	2.0	16#0	False	True	True	True	False		
ZONAS_OK	Word	4.0	16#0	False	True	True	True	False		

Anexo 2: Variables PLC y HMI

En este anexo se encuentran todas las variables definidas durante el proyecto tanto dentro del PLC como para la interfaz HMI.

Totally Integrated Automation Portal									
HORNO_717 / PLC_1 [CPU 1214C DC/DC/DC]									
Variables PLC									
Nombre	Tipo de datos	Dirección	Remanenda	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Supervisión	Comentario	
T1	Int	%IW96	False	True	True	True		ENTRADA DE TEMPERATURA Z1	
T2	Int	%IW98	False	True	True	True		ENTRADA DE TEMPERATURA Z2	
T3	Int	%IW100	False	True	True	True		ENTRADA TEMPERATURA Z3	
T4	Int	%IW102	False	True	True	True		ENTRADA TEMPERATURA Z4	
frq	DInt	%ID1000	False	True	True	True			
T5	Int	%IW104	False	True	True	True		ENTRADA TEMPERATURA Z5	
T6	Int	%IW106	False	True	True	True		ENTRADA TEMPERATURA Z6	
T7	Int	%IW108	False	True	True	True		ENTRADA TEMPERATURA Z7	
ContactoHeatOK	Bool	%IO.0	False	True	True	True			
velocidad	Bool	%IO.1	False	True	True	True			
T8	Int	%IW110	False	True	True	True			
AIR	Bool	%IO.4	False	True	True	True			
WATER	Bool	%IO.5	False	True	True	True			
PresenciaAguaWJ1	Bool	%IO.6	False	True	True	True			
PresenciaAguaWJ2	Bool	%IO.7	False	True	True	True			
Z1	Bool	%Q0.0	False	True	True	True		PWM ZONA 1	
Z2	Bool	%Q0.1	False	True	True	True		PWM ZONA 2	
Z3	Bool	%Q0.2	False	True	True	True		PWM ZONA 3	
Z4	Bool	%Q0.3	False	True	True	True		PWM ZONA 4	
Z5	Bool	%Q0.4	False	True	True	True		PWM ZONA 5	
Z6	Bool	%Q0.5	False	True	True	True		PWM ZONA 6	
Z7	Bool	%Q0.6	False	True	True	True		PWM ZONA 7	
ActivaciónHomo	Bool	%Q1.0	False	True	True	True			
US	Bool	%Q1.1	False	True	True	True			
Z8	Bool	%Q0.7	False	True	True	True			
ActivaAir	Bool	%Q12.3	False	True	True	True			
Activa Waer	Bool	%Q12.4	False	True	True	True			
BOCINA	Bool	%Q12.5	False	True	True	True			
SEMAFORO_ROJO	Bool	%Q12.0	False	True	True	True			
SEMAFORO_NARANJA	Bool	%Q12.1	False	True	True	True			
SEMAFORO_VERDE	Bool	%Q12.2	False	True	True	True			
STATUS1	DWord	%MD1	False	True	True	True		ESTADO DEL CONTADOR	
System_Byte	Byte	%MB1	False	True	True	True			
FirstScan	Bool	%M1.0	False	True	True	True			
DiagStatusUpdate	Bool	%M1.1	False	True	True	True			
AlwaysTRUE	Bool	%M1.2	False	True	True	True			
AlwaysFALSE	Bool	%M1.3	False	True	True	True			
Clock_Byte	Byte	%M300	False	True	True	True			
Clock_10Hz	Bool	%M300.0	False	True	True	True			
Clock_5Hz	Bool	%M300.1	False	True	True	True			
Clock_2.5Hz	Bool	%M300.2	False	True	True	True			
Clock_2Hz	Bool	%M300.3	False	True	True	True			
Clock_1.25Hz	Bool	%M300.4	False	True	True	True			
Clock_1Hz	Bool	%M300.5	False	True	True	True			
Clock_0.625Hz	Bool	%M300.6	False	True	True	True			
Clock_0.5Hz	Bool	%M300.7	False	True	True	True			
AVISOS	Word	%MW1	False	True	True	True			
ACUSE_DE_AVISOS	Word	%MW6	False	True	True	True			
ControlPID	Word	%MW22	False	True	True	True			
STATUS2	Word	%MW24	False	True	True	True			
Clock_Byte(1)	Word	%MW26	False	True	True	True			
Clock_10Hz(1)	Word	%MW28	False	True	True	True			
Clock_5Hz(1)	Word	%MW30	False	True	True	True			
Clock_2.5Hz(1)	Word	%MW32	False	True	True	True			
Clock_2Hz(1)	Word	%MW34	False	True	True	True			
Clock_1.25Hz(1)	Word	%MW36	False	True	True	True			

Totally Integrated Automation Portal									
Nombre	Tipo de datos	Dirección	Remanenda	Accesible desde HMI/OPC UA/Web API	Escribible desde HMI/OPC UA/Web API	Visible en HMI Engineering	Supervisión	Comentario	
 Clock_1Hz(1)	Word	%MW38	False	True	True	True			
 Clock_0.625Hz(1)	Word	%MW40	False	True	True	True			
 Clock_0.5Hz(1)	Word	%MW42	False	True	True	True			
 Var_Semáforo	Int	%MW12	False	True	True	True			
 System_Byte(1)	Byte	%MB1	 False	True	True	True			