



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

PROYECTO DE AUTOMATIZACIÓN Y  
MONITORIZACIÓN DE UN PROCESO DE  
TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER  
Y COMUNICACIÓN OPC-UA

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

AUTOR/A: Ferrer Campos, Carlos

Tutor/a: Pitarch Pérez, José Luis

Cotutor/a: Simarro Fernández, Raúl

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERIA  
INDUSTRIAL VALÈNCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**PROYECTO DE AUTOMATIZACIÓN Y  
MONITORIZACIÓN DE UN PROCESO DE  
TRATAMIENTO TÉRMICO MEDIANTE PLCs  
SCHNEIDER Y COMUNICACIÓN OPC-UA**

AUTOR: CARLOS FERRER CAMPOS

TUTOR: JOSÉ LUÍS PITARCH PÉREZ

COTUTOR: RAÚL SIMARRO FERNÁNDEZ

**Curso Académico: 2021-2022**



## **AGRADECIMIENTOS**

A mi tutor y cotutor, José Luís Pitarch y Raúl Simarro por guiarme hasta este punto. Por resolver todas las dudas que me han ido surgiendo por el camino y ayudarme a desarrollarme académicamente al máximo.

A mi familia, por estar ahí siempre y apoyarme en momentos en los que ni siquiera yo veía el camino a seguir.

A mi pareja y mis amigos por creer en mí y ayudarme a seguir día a día.

Y por último a la universidad y el departamento, por poner a mi disposición todo el conocimiento y material necesario para haber llegado hasta aquí.

## RESUMEN

El objetivo del trabajo es el de diseñar e implementar un sistema que automatice y digitalice una estación donde se realiza un tratamiento térmico y el postprocesado de ciertas piezas. La implementación se realiza utilizando autómatas M241 y TSX-Premium de Schneider y se demuestra con unos prototipos a escala de la estación y un brazo manipulador encargado del traslado de las piezas.

Para emular el proceso térmico se ha utilizado una célula Peltier controlada mediante PID. Una cámara Kinect se utiliza junto con un programa de LabView a modo de sensor de visión artificial para seleccionar el tratamiento térmico a llevar a cabo en función del color de la pieza.

Se ha utilizado además la base de datos InfluxDB para almacenar los datos del proceso térmico y ofrecer una interfaz personalizable para consultar los datos, tanto del histórico como en tiempo real.

La comunicación entre el PLC que controla el proceso, el PC con el software de detección de color y la base de datos se realiza mediante un servidor OPC-UA, permitiendo la gestión y supervisión en tiempo real de todo el proceso.

El HMI se aloja en un servidor web y puede accederse a él a través de cualquier dispositivo conectado a la misma red, por ejemplo, una tableta.

**Palabras Clave:** Control PID, Comunicación, OPC-UA, PLC, Servidor Web, Schneider, LabView, Proceso Industrial, Base de Datos, InfluxDB.

## **ABSTRACT**

The goal of this project is to design and implement a system that automatizes and digitalizes a thermal treatment process in a station where the pieces are also postprocessed. The implementation has been done using the Schneider PLCs, M241 and TSX-Premium and it is demonstrated with a scaled prototypes of the station and a robotic arm which transfer the pieces.

The termic process has been emulated by a Peltier cell controled through PID. One Kinect camera is used with a LabView program as artiifiical-vision sensor to select the right termic treatment that will be performed in function of the pieces color.

The proposed solution also includes the real-time database InfluxDB, to store the process data and to provide a customizable interface that allows consulting historic and real-time mesurement.

Comunication between the PLC client, the PC with the colour detection software and the data base is done via OPC-UA server, allowing real-time management and supervision of the entire process.

There's a web server hosting the HMI and it can be accesed through any devices connected to the same network, like a tablet.

**Keywords:** PID Control, Comunication, OPC-UA, PLC, Web Server, Schneider, LabView, Industrial Process, Data Base, InfluxDB.

## RESUM

L'objectiu del treball és dissenyar i implementar un sistema que automatitzi i digitalitzi una estació on es realitza un tractament tèrmic i el postprocessat de certes peces. La implementació es realitza utilitzant autòmats M241 i TSX-Premium de Schneider i es demostrarà amb uns prototips a escala de l'estació i un braç manipulador encarregat del trasllat de les peces.

Per emular el procés tèrmic s'ha utilitzat una cèl.lula Peltier controlada mitjançant un PID. Una càmera Kinect s'utilitza junt amb un programa de LabView a mode de sensor de visió artificial per seleccionar el tractament tèrmic a dur a terme en funció del color de la peça.

S'ha utilitzat també la base de dades, InfluxDB, per emmagatzemar les dades del procés tèrmic i oferir una interfície personalitzable per consultar les dades, tant de històric com en temps real.

La comunicació entre el PLC que controla el procés, el PC amb el software de detecció de color i la base de dades es realitza mitjançant un servidor OPC-UA, permetint així la gestió i supervisió en temps real de tot el procés.

El HMI s'allotja en un servidor web i pot accedir-se a ell per mitjà de qualsevol dispositiu connectat a a mateixa xarxa, per exemple, una tableta.

**Paraules Clau:** Control PID, Comunicació, OPC-UA, PLC, Servidor Web, Schneider, LabView, Procés Industrial, Base de Dades, InfluxDB.

**PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA**

# ÍNDICE DEL PROYECTO

## DOCUMENTOS DEL PROYECTO

- MEMORIA
- ANEXO I: DISEÑO DEL AUTOMATISMO
- ANEXO II: DISEÑO DEL PID
- ANEXO III: INSTALACIÓN Y CONFIGURACIÓN INFLUXDB
- PRESUPUESTO

## ÍNDICE DE LA MEMORIA

<b>CAPÍTULO 1: Introducción</b> .....	2
1.1 Objetivo .....	2
1.2 Antecedentes .....	3
1.3 Justificación de las alternativas .....	3
1.3.1 Proceso .....	4
1.3.2 PLCs .....	4
1.3.3 Sensor de Visión .....	5
1.3.4 Tratamiento térmico .....	6
1.3.5 Base de Datos y Servidor OPC-UA .....	6
<b>CAPÍTULO 2: Solución Adoptada</b> .....	7
2.1 Descripción de los medios utilizados .....	7
2.1.1 Maquetas FischerTechnik .....	7
2.1.2 PLCs .....	11
2.1.3 Cámara Kinect .....	12
2.1.4 Célula Peltier .....	12
2.2 Funcionamiento del sistema .....	18
2.2.1 Estructura de subsistemas .....	18

2.2.2 Modos de funcionamiento .....	19
2.3 Manual del programador .....	23
2.3.1 Programación PLCs .....	23
2.3.2 LabView .....	43
2.3.3 Base de Datos .....	50
<b>CAPÍTULO 3: Manual del usuario .....</b>	<b>58</b>
3.1 Inicialización del sistema .....	58
3.2 Manipulación del sistema .....	59
<b>CAPÍTULO 4: Conclusión .....</b>	<b>65</b>
<b>CAPÍTULO 5: Bibliografía .....</b>	<b>66</b>

## ÍNDICE DEL PRESUPUESTO

<b>1 ELABORACIÓN DEL PRESUPUESTO .....</b>	<b>2</b>
<b>2 CUADRO DE MANO DE OBRA .....</b>	<b>2</b>
<b>3 CUADRO DE MATERIALES .....</b>	<b>3</b>
<b>4 CUADRO DE MAQUINARIA .....</b>	<b>3</b>
<b>5 CUADROS DE PRECIOS UNITARIOS .....</b>	<b>4</b>
<b>6 CUADROS DE PRECIOS DESCOMPUESTOS .....</b>	<b>5</b>
<b>7 PRESUPUESTO DE EJECUCIÓN MATERIAL .....</b>	<b>7</b>
<b>8 PRESUPUESTO BASE DE LICITACIÓN .....</b>	<b>7</b>



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERIA  
INDUSTRIAL VALÈNCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**PROYECTO DE AUTOMATIZACIÓN,  
MONITORIZACIÓN Y DIGITALIZACIÓN DE UN  
PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y  
COMUNICACIÓN OPC-UA**

**MEMORIA**

AUTOR: CARLOS FERRER CAMPOS

TUTOR: JOSÉ LUÍS PITARCH PÉREZ

COTUTOR: RAÚL SIMARRO FERNÁNDEZ

**Curso Académico: 2021-2022**

## CAPÍTULO 1: Introducción

### 1.1 Objetivo

El objetivo de este proyecto es el de automatizar y digitalizar un proceso de tratamiento térmico. El proceso en cuestión consta de tres partes, una zona donde el operario, u otro sistema que escapa al alcance de este proyecto, coloca la pieza a tratar y esta es clasificada, un horno donde se realizará el tratamiento térmico y una estación de mecanizado donde se postprocesará la pieza en cuestión. Paralelamente un brazo manipulador trasladará las piezas de la zona de recogida al horno, y de la fase final de mecanizado a su lugar de almacenamiento.

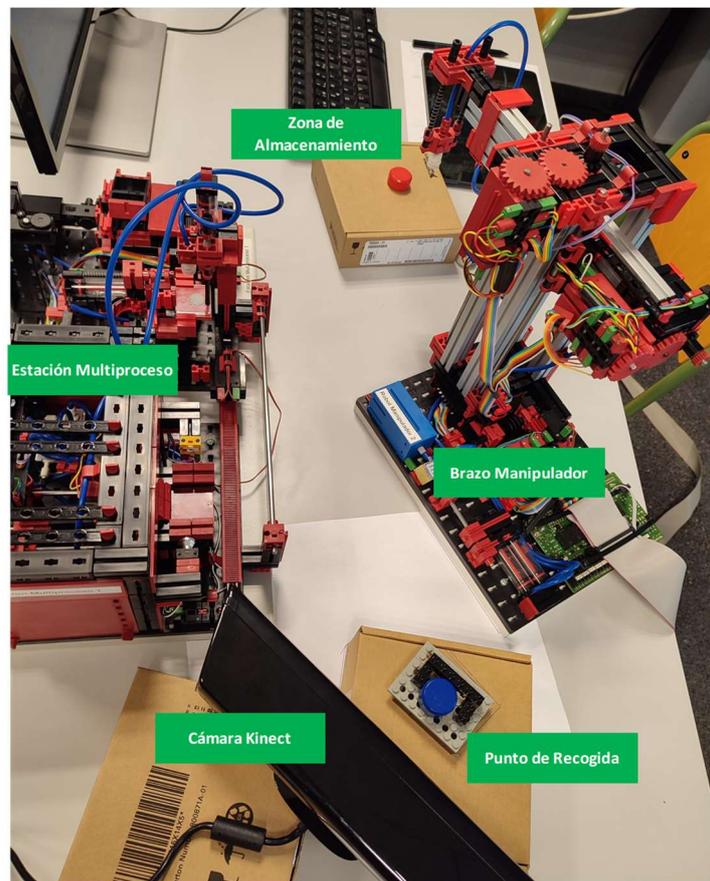


Imagen 1: Montaje experimental. Fuente: Elaboración propia.

Dada la naturaleza del proceso y las necesidades de la planta donde se aplicará el sistema, las especificaciones deberían ser las siguientes:

- ❖ Existencia de un sistema que clasifique y transmita la información sobre el tratamiento a aplicar/aplicado.
- ❖ Sistema de seguridad que asegure la compatibilidad entre los procesos de traslado y limite los riesgos que puedan darse.

- ❖ Para asegurar la fluidez del proceso se ha limitado el tiempo que debe tardar el horno en calentarse a 15 segundos y se hará compatible la simultaneidad del proceso de fresado y el tratamiento térmico.
- ❖ Una interfaz sencilla y robusta para poder gobernar el proceso.
- ❖ Una base de datos que permita consultar los datos para buscar posibles mejoras o fallos en el proceso.

Para ello se debe desarrollar:

- Un automatismo capaz de procesar la información de los sensores y realizar las acciones pertinentes para asegurar la correcta secuenciación del proceso automático.
- La instalación de un sensor inteligente que permita determinar el color de la pieza a tratar y seleccionar el tratamiento térmico que se le realizará.
- Un protocolo de comunicaciones común y estandarizado.
- Un sistema de control capaz de procesar la pieza de acuerdo con las especificaciones.
- Un sistema de monitorización y almacenamiento de información en tiempo real.

## 1.2 Antecedentes

Este proyecto nace de la necesidad que tenemos los seres humanos de optimizar nuestro tiempo, de buscar la mejora continua y la optimización de nuestros procesos haciéndolos más rápidos, sencillos y menos peligrosos.

Es en este marco en el que nace la industria 3.0, esta se caracteriza por el comienzo de la automatización de los procesos, basada en el uso de tecnologías como la electrónica, la informática, la microelectrónica y nos lleva a la producción automatizada. (SystemHaus , 2021)

No obstante, la continua evolución de la tecnología y de los procesos nos lleva nuevamente a dar un paso más e integrar esos automatismos y procesos físicos al mundo digital donde puedan almacenarse, gestionarse y compartirse datos en tiempo real; aplicarse técnicas de Inteligencia Artificial o facilitar la comunicación entre sistemas. Esta es la revolución de la industria 4.0.

En el presente proyecto aparecen técnicas ya utilizadas en la industria 3.0 como la programación de autómatas o sensores de visión basados en el espectro electromagnético, por tanto, se ha querido avanzar hacia esta industria 4.0 incluyendo sistemas que gestionen y comuniquen los datos generados por estos procesos, una base de datos versátil y un interfaz robusto que pueda ser soportado por diversos dispositivos.

## 1.3 Justificación de las alternativas

De acuerdo con las necesidades y especificaciones expresadas en los puntos anteriores las opciones para desarrollar el proyecto son muy variadas.

### 1.3.1 Proceso

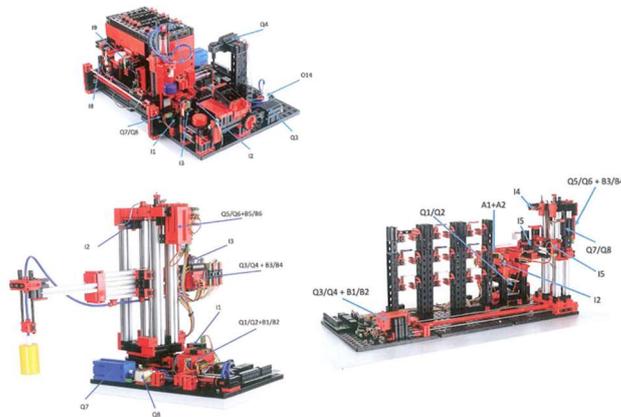
La primera decisión que debía tomarse era la de que proceso recrear y como recrearlo, podría hacerse bien mediante software de simulación, o bien utilizando las maquetas de que dispone el Departamento de Ingeniería de Sistemas y Automática.

El software en cuestión sería el FactoryIO, con el cual pueden realizarse simulaciones de procesos productivos creando una planta desde cero y automatizándola con simulación PLC.



*Imagen 2: Simulación en FactoryIO. Fuente: Trabajo Automatización de Procesos, Raúl Simarro (2022).*

Las maquetas disponibles son las de la firma alemana fischertechnik, algunos de los ejemplos son la estación multiproceso, el brazo manipulador o el almacén elevado automatizado.



*Imagen 3: Maquetas Fischertechnik. Fuente: Procesos FischerTechnik, Raúl Simarro(2022).*

### 1.3.2 PLCs

La función de gestionar los diferentes sensores y actuadores de las maquetas recae sobre el Controlador Lógico Programable, en adelante PLC. Algunos de los PLCs disponibles en el laboratorio y con los cuales se había tenido una experiencia previa o se conocían ciertas características de ellos que los hacían interesantes son:

- **TSX-Premium**

PLC de la marca Schneider, ya había sido utilizado a nivel básico en asignaturas de cursos anteriores como Tecnología Automática o la optativa de Laboratorio de Automatización y Control. El hecho de conocer el software de programación previamente y el conocimiento de la función para compartir memoria con otros PLCs fácilmente lo hacía una alternativa bastante interesante.



Imagen 4: TSX-Premium de Schneider

- **M241**

PLC también de la marca Schneider, este PLC se programa con un software basado en Codesys, una herramienta de programación que se adapta al estándar **IEC 61131-3** y que es compatible con más de 1000 dispositivos de más de 400 fabricantes (Elektronika, Larraioz, s.f.). Además, este software tiene la opción de crear un HMI muy visual y que puede accederse vía servidor web. Por todo esto, el M241 se ve como una buena opción en cuanto a utilidad y valor didáctico para alguien que está iniciándose en la programación de automatismos.



Imagen 5: M241 de Schneider

### 1.3.3 Sensor de Visión

Una vez seleccionado el proceso se debía elegir como complementarlo, la posibilidad de poder realizar diferentes procesos en función de alguna característica de las piezas era muy llamativo, y en cuanto a los medios existían dos opciones:

- **Pista de clasificación con reconocimiento de color FischerTechnik**

Maqueta de FischerTechnik con un sensor de color que permite clasificar las piezas.

Esta idea aporta complejidad sin aportar ninguna novedad al proyecto, simplemente tendría que ampliarse el programa ejecutado por el PLC y probablemente traería incompatibilidades con el rango de trabajo del brazo, no obstante, era una opción que estaba disponible.

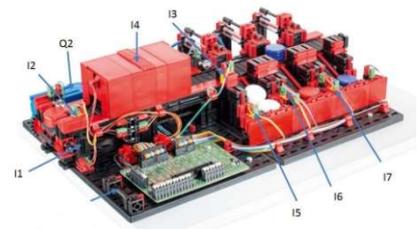


Imagen 6: Pista de clasificación. Fuente: Procesos FischerTechnik, LAC

- **Cámara Kinect + software LabView**

El entorno de programación gráfica LabView, permite desarrollar aplicaciones muy interesantes para el campo de la ingeniería, son cada día más las empresas que incorporan software creado con este software de National Instruments. En concreto una

de sus bibliotecas nos permite utilizar una cámara Kinect para distintos cometidos, uno de ellos detectar el espectro electromagnético de una zona localizada de la imagen captada, y con ello puede distinguir colores creando la aplicación adecuada.



Imagen 7: Cámara Kinect

#### 1.3.4 Tratamiento térmico

Para asimilar el proceso lo máximo posible a la realidad el funcionamiento del horno debía ir ligado a algún sistema que emulara el incremento de temperatura y que además pudiera controlarse. En el departamento existían dos posibles opciones:

- **Horno**  
Sistema formado por una resistencia acoplada a un radiador, un sensor de temperatura modelo SEM104/TC y la correspondiente etapa de potencia y tarjeta de adquisición de datos. Se ha utilizado en la asignatura de Tecnología Automática y constituiría una muy buena aproximación del proceso a emular.
- **Célula Peltier**  
Este tipo de células aprovecha el efecto termoeléctrico para enfriar una de sus caras mientras que la otra se calentaría, el prototipo del departamento además cuenta con una resistencia acoplada a la cara caliente y un ventilador para disipar el calor generado rápidamente. Este sistema también fue utilizado en la optativa de Laboratorio de Automatización y Control. Pese a que la función original de la célula Peltier no es la de calentar sino la de enfriar podemos jugar con las entradas para crear el efecto contrario, con el horno inactivo la temperatura se estabilizará a un valor, y se “calentará” cuando este se active.

#### 1.3.5 Base de Datos y Servidor OPC-UA

Con el avance del proyecto y con la mirada puesta en la industria 4.0 y el progreso del ámbito tecnológico en el que se enmarca el trabajo parecía indispensable añadir una base de datos avanzada para la gestión de la información tanto durante el procesado como en instantes posteriores. Así es como se llegó hasta InfluxDB, una base de datos fácil de utilizar que ofrece interfaces personalizables, tratamiento de los datos, creación de alarmas y un fácil acceso en toda la red. Para transmitir los datos desde el PLC había diversas opciones como Modbus TCP/IP o un servidor OPC-UA, en un primer momento se utilizaba un servidor OPC-DA para comunicar el programa de LabView con el PLC con lo que se decidió sustituir este servidor por uno OPC-UA creado por el PLC M241 que sirviera para conectar servidor, autómatas y base de datos

## CAPÍTULO 2: Solución Adoptada

Antes de entrar en detalle es esencial comprender el funcionamiento esperado del sistema, así como los elementos que conforman cada subsistema y la misión que cumple cada uno de ellos. Este es por tanto el objeto de este capítulo y no el de explicar cómo se ha implementado cada solución, lo cual se analizará en el Capítulo 3.

### 2.1 Descripción de los medios utilizados

Después de tener en cuenta todas las posibilidades disponibles, finalmente los medios seleccionados han sido los siguientes:

#### 2.1.1 Maquetas FischerTechnik

Se ha preferido emular el proceso mediante las maquetas FischerTechnik por considerarse una alternativa bastante cercana a la aplicación en un sistema real a escala.

Las maquetas que se utilizan son la estación multiproceso y el brazo manipulador. En la estación multiproceso se puede ejecutar un proceso secuencial en el cual después de depositar una pieza en la bandeja del horno, esta entra, simula el proceso de horneado, y sale. Después de este proceso un manipulador integrado en la maqueta recoge la pieza y la lleva a la mesa giratoria. Una vez aquí la mesa puede girar para ejecutar el mecanizado de la pieza o para continuar hasta la posición final, donde la pieza es empujada y llevada mediante una cinta a la zona de recogida. El brazo manipulador se trata de un brazo robótico con movilidad en tres ejes, tiene un rango de movimiento de 140 mm en el eje horizontal, de 120 mm en el vertical y de 270º respecto del eje vertical (Fischertechnik, 2022). Dispone de una ventosa accionada mediante neumática para recoger piezas.

En adelante se analizarán las señales de entrada y los actuadores de las maquetas utilizadas.

#### Entradas

Las entradas son aquellas variables activadas o controladas por los sensores existentes en la maqueta y son esenciales para proporcionar al usuario información sobre el proceso y permitir un control eficiente del mismo.

En la estación multiproceso pueden localizarse las entradas siguientes:

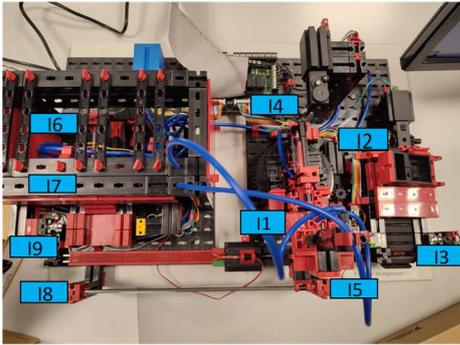


Imagen 8: Entradas estación multiproceso.

Fuente: Elaboración propia.

	Variable	Descripción
Entradas	I1	FC Mesa en Manipulador
	I2	FC Mesa en Cinta
	I3	Pieza Acabada
	I4	FC Mesa en Sierra
	I5	FC Manipulador en Mesa
	I6	FC Alimentador dentro
	I7	FC Alimentador fuera
	I8	FC Manipulador en Horno
	I9	Pieza en Alimentador

Tabla 1: Entradas estación multiproceso.

Fuente: Elaboración propia.

Mientras que en el brazo manipulador se localizan las siguientes:

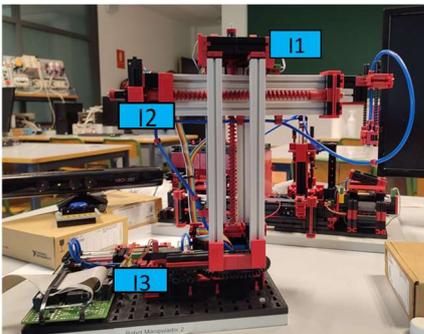


Imagen 9: Entradas brazo manipulador. Fuente: Elaboración propia.

	Variable	Descripción
Entradas	B1	Encoder Vertical
	B3	Encoder Horizontal
	B5	Encoder Giratorio
	I1	FC Vertical
	I2	FC Horizontal
	I3	FC Giro

Tabla 2: Entradas brazo manipulador.

Fuente: Elaboración propia.

Se diferencian 3 tipos de sensores de entrada, la función de todos ellos es la de determinar la posición del dispositivo controlado:

- Finales de Carrera: Disponen de un pulsador, cuando este sea presionado el valor de la variable binaria asociada será de 1.
- Sensor de Luz: Conjunto formado por una bombilla y un sensor de luz, el sensor estará activo siempre que la luz de la bombilla incida en él.
- Contador Pulsos: Dispositivos asociados a un encoder, incrementando su valor por cada tres pulsos eléctricos del encoder. De esta manera se transforma una señal digital temporal a un valor discretizado que permite la localización y calibración de las coordenadas del brazo.

Las entradas asociadas a Finales de Carrera en la estación multiproceso son la I1, I2, I4, I5, I6, I7 e I8 y determinan si la bandeja del horno, el manipulador y la mesa de fresado se encuentran en sus posiciones extremas. Las posiciones extremas de la bandeja son dentro o fuera del horno, las del manipulador son en el horno o en la mesa de fresado y las de la mesa de fresado serán en el manipulador, en la fresa o en la cinta.

Las entradas asociadas a Sensores de Luz en la estación multiproceso son la I9 y la I3. El sensor I3 se desactivará cuando haya una pieza terminada al final de la cinta mientras que el I9 simplemente reflejará un flanco de bajada cuando una pieza sea depositada en la bandeja, esto

se debe a que el sensor no detecta cuando la pieza está depositada, sino que esta interrumpe el paso de luz a su paso.

Por lo que respecta al brazo los Finales de Carrera están relacionados con las variables I1, I2 e I3 las cuales se activarán cuando el brazo esté en su posición de reposo. Esta posición se corresponde con el brazo arriba, retraído y con la orientación que se observa en la imagen 9.

Esta posición es importante porque cuando el brazo está en la posición de reposo de alguno de los 3 ejes en que se mueve, el encoder del motor asociado a dicho movimiento reinicia su valor a 0.

Así pues, los encoders están asociados a las variables B1, B3 y B5. Estos ofrecen información sobre la posición del brazo en el eje correspondiente respecto de la posición de reposo. B1 y B3 están limitados a un valor de 1000 mientras que B5 lo está a un valor de 750.

Salidas

Las salidas son las acciones y procesos que se realizan en las maquetas, estas están controladas por sus respectivos actuadores, los cuales serán activados mediante una variable que puede ser analógica o digital. Los actuadores de la estación multiproceso están alimentados a 9V (Fischertechnik, 2022) mientras que los del brazo lo están a 24V (Fischertechnik, 2022)

En la estación multiproceso pueden localizarse las salidas siguientes:

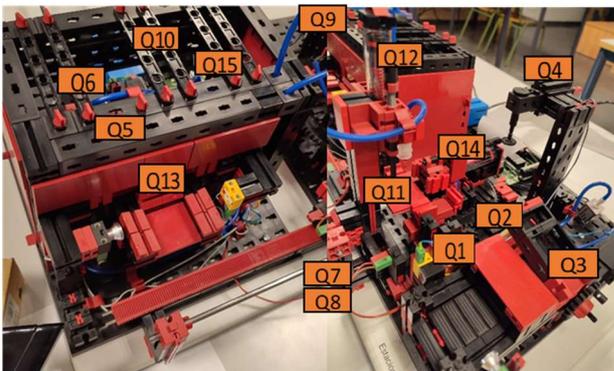


Imagen 10: Salidas estación multiproceso. Fuente: Elaboración propia.

Variable	Descripción
Q1	Giro Mesa Horario
Q2	Giro Mesa Antihorario
Q3	Avance Cinta
Q4	Sierra
Q5	Alimentador dentro
Q6	Alimentador fuera
Q7	Manipulador hacia Horno
Q8	Manipulador hacia Mesa
Q9	Habilitar Señales
Q10	Compresor
Q11	Ventosa
Q12	Manipulador Abajo
Q13	Puerta Arriba
Q14	Empujador Mesa
Q15	Luz Horno

Tabla 3: Salidas estación multiproceso. Fuente: Elaboración propia.

Mientras que en el brazo manipulador se localizan las siguientes:

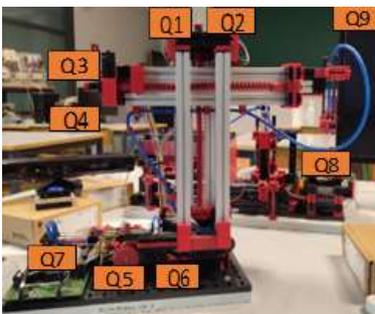


Imagen 11: Salidas brazo manipulador. Fuente: Elaboración propia.

Variable	Descripción
Q1	Vertical Arriba
Q2	Vertical Abajo
Q3	Retroceso Horizontal
Q4	Avance Horizontal
Q5	Giro Horario
Q6	Giro Antihorario
Q7	Compresor
Q8	Ventosa
Q9	Habilitar Señales

Tabla 4: Salidas brazo manipulador. Fuente: Elaboración propia.

Los actuadores disponibles se clasifican en neumáticos, mecánicos y eléctricos. La variable Q9 se utiliza en ambas maquetas para habilitar sus respectivos sensores y actuadores.

La estación multiproceso tiene diversos actuadores mecánicos que se utilizan para mover los diferentes dispositivos, el motor activado por Q1 y Q2 es el responsable del movimiento rotatorio de la mesa en los dos sentidos; los motores accionados por Q3, Q5 y Q6; Q7 y Q8 generan el movimiento lineal de la cinta, del alimentador del horno y del manipulador, respectivamente. Por último, Q4 genera el movimiento rotatorio de la sierra.

En cuanto a la neumática, la variable Q10 activa el compresor, lo cual permita que el resto de las actuaciones neumáticas sean posibles. La ventosa que permite la cogida de las piezas se activa mediante Q11 mientras que Q12, Q13 y Q14 activan los cilindros de simple efecto que hacen descender el manipulador, abren la puerta y empujan la pieza de la mesa, respectivamente.

Por último, Q15 enciende una bombilla que simula el encendido del horno.

El brazo también tiene motores que generan un movimiento rotatorio, es el caso del motor activado por Q5 y Q6, los cuales se encargan la rotación del brazo entorno al eje vertical; las variables Q1 y Q2, Q3 y Q4 activan los motores que generan el movimiento vertical y horizontal respectivamente.

En último lugar, la neumática del brazo se reduce simplemente a la ventosa que permite el traslado de las piezas, activada por Q8. Al igual que en la estación multiproceso, para activar la neumática será necesario habilitar el compresor, esto se hará con la variable Q7.

### Calibración Brazo

Para el correcto posicionamiento de las maquetas es preciso asegurarse de que el brazo manipulador es capaz de llegar a los puntos de depósito y recogida de las piezas. Esto se ha conseguido mediante un sencillo programa que permite accionar los actuadores del brazo entre sus valores extremos y que muestra el valor de las variables vinculadas a los encoders, gracias a ellas pueden programarse acciones condicionadas a la posición del brazo.



Imagen 12: Pantalla calibración brazo.

Teniendo en cuenta que tanto para calibrar como durante el funcionamiento del brazo se tendrán que empezar todos los movimientos desde su posición inicial y hacerlos de una vez, para no descuadrar los pulsos generados por el sensor, las posiciones calibradas han sido las siguientes:



Los autómatas están programados con el software Unity Pro M y SoMachine los cuales como podemos comprobar en el manual de referencia del Unity Pro (2009) y en el manual de programación de SoMachine(2018) están basados en el estándar **IEC 61131-3** gracias al cual podemos programarlos de manera similar utilizando lenguajes como el de Bloques de función secuenciales (SFC) o el de Diagrama de Contactos (LD).

### 2.1.3 Cámara Kinect

Finalmente se adopta la solución de la cámara Kinect junto con el software LabView, con el objetivo de adquirir conocimientos sobre este software, que previsiblemente iba a ser utilizado para otro de los procesos.

La cámara Kinect es un dispositivo habitualmente utilizado en videojuegos de realidad aumentada, con una resolución de 640×480 de alto color y que permite una frecuencia de actualización de la imagen de hasta 30fps, cuenta además con una cámara RGB y un sensor de profundidad.

Tiene un campo de visión horizontal de 57º, un campo de visión vertical de 43º y una capacidad física de inclinación de (-27, +27) º (FayerWayer, 2010).

La cámara RGB es la parte fundamental de este dispositivo en el proyecto, ya que, junto a un software adecuado, permite identificar las longitudes de onda del espectro visible de una zona concreta (Point of Interest). Esto será utilizado para identificar el color de la pieza a tratar y decidir el tratamiento que recibirá en función de este.

### 2.1.4 Célula Peltier

La célula Peltier es el dispositivo seleccionado para llevar a cabo la emulación del proceso térmico, pese a que podría parecer más lógico seleccionar el horno por su similitud con el proceso que se quiere imitar, el comportamiento de la Peltier cuando su cara fría se calienta también es muy similar al real, con la diferencia de la velocidad. Mientras el proceso del horno puede tardar varios minutos en estabilizarse, el de la Peltier lo hace en cuestión de segundos, con lo que se adapta mejor al proyecto por agilizar las pruebas de funcionamiento del sistema completo. Se ha decidido integrar el control de la temperatura en el programa de LabView.

“Las células Peltier son dispositivos que permiten generar un enfriamiento mediante el efecto termoeléctrico o Peltier. Físicamente son de reducido tamaño y en este caso tienen forma de paralelepípedo achatado (ver figura adjunta). Internamente están constituidos por parejas de semiconductores de tipo P y de tipo N conectados eléctricamente en serie, que se encuentran encerradas entre dos superficies cerámicas” (José Vicente Salcedo, Manuales de prototipos).

Para poder aplicar su funcionamiento al proyecto se ha de complementar con un termopar tipo k para sensar la temperatura de la cara fría, junto con el conversor conveniente, en este caso de 4-20mA.

Para identificar el proceso se utiliza una Tarjeta de adquisición de datos, en este caso una USB-6001 de National Instruments, para que esta procese la información del termopar, se acopla una

resistencia de precisión 500Ω, convirtiendo la corriente del convertidor en un rango de tensiones (2,10) V.

Como el rango de trabajo de la peltier es de (-30,90)° se ha realizado la conversión de V→°C para conocer la temperatura que está detectando el sensor en función de la tensión generada.

$$\begin{aligned} 90^{\circ}\text{C} &\rightarrow 20\text{mA} \rightarrow 10\text{V} \\ -30^{\circ}\text{C} &\rightarrow 4\text{mA} \rightarrow 2\text{V} \end{aligned}$$

$$T(^{\circ}\text{C}) = -60 + 15 * V$$

Ecuación 1: Conversión T°Peltier en función de la tensión.

La peltier permite una tensión máxima de funcionamiento de 8V, en este caso se alimentará mediante una fuente de tensión controlable, que trabajará en un rango de tensiones de 2V, que serán amplificados hasta 6V. Se ha realizado la conversión entre ΔU → V 0 sabiendo que el valor de actuación mínimo sobre la Peltier y 2 el máximo.

$$V = 0.02 * \Delta U$$

Ecuación 2: Conversión tensión Peltier en función de la acción de control.

Para garantizar la integridad de todo el sistema se utilizan un radiador y un ventilador para garantizar la disipación del calor generado, este ventilador está actuado con un rango de tensiones de 10V, también se ha realizado la conversión entre ΔU → V.

$$V = 0.1 * \Delta U$$

Ecuación 3: Conversión tensión ventilador en función de la acción de control.

La conexión de todos los sistemas anteriormente mencionados se realizará de la siguiente manera:

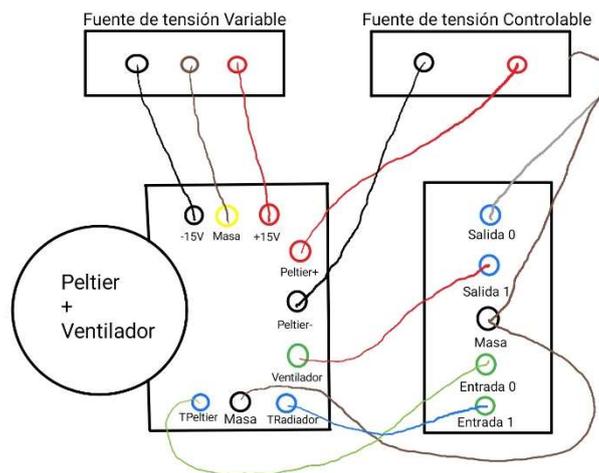


Imagen 16: Esquema conexión Peltier. Fuente: Elaboración propia.

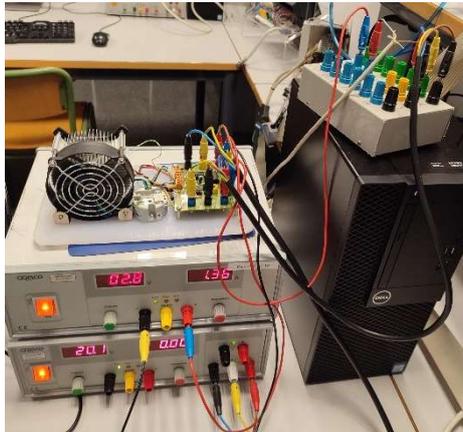


Imagen 17: Montaje experimental - célula Peltier. Fuente: Elaboración propia.

Puede observarse que en la tarjeta de adquisición de datos se dispondrá de la lectura de la temperatura de la cara fría y de la cara caliente en las entradas 0 y 1 respectivamente, y podrá actuarse sobre la Peltier y el ventilador mediante las salidas 0 y 1 respectivamente, para la identificación del proceso además puede disponerse de la lectura de la temperatura ambiente conectando la única conexión que queda libre en el circuito impreso a cualquiera de las entradas de la tarjeta de adquisición de datos.

En el proyecto que nos ocupa la célula Peltier se utiliza para simular el proceso térmico, enfriando la célula hasta una temperatura que represente la situación de reposo, para después calentarla a la temperatura de referencia, en este sistema estos procesos de calentamiento y enfriamiento ante una perturbación de tipo escalón siguen un modelo de primer orden sin retardo ( $\theta = 0$ ).

$$G(s) = \frac{K}{\tau \cdot s + 1}$$

Ecuación 4: Modelo proceso primer orden

Para aplicar el proceso al sistema completo es necesario conocer sus características para conseguir un control adecuado que asegure las especificaciones deseadas. Para ello se ha identificado el proceso utilizando las herramientas proporcionadas por los softwares LabView y Matlab.

El primer paso para la identificación es el de conseguir datos experimentales, esto se ha hecho implementando un programa en LabView que permita controlar las acciones sobre la célula Peltier y el ventilador, y monitorizar y almacenar las lecturas del termopar sobre la cara fría de la célula.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

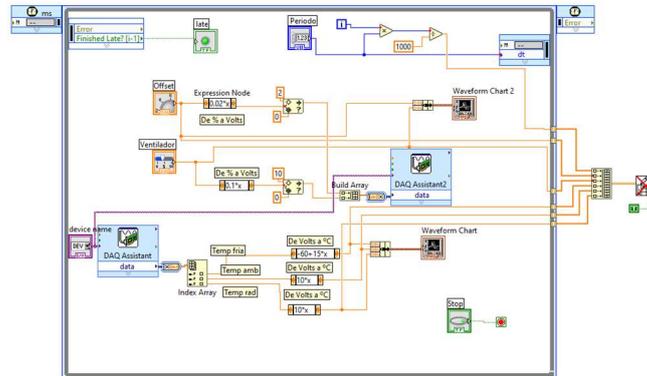


Imagen 18: Programa identificación proceso.

En dicho programa se han implementado las conversiones descritas en las ecuaciones 1, 2 y 3 y mediante las bibliotecas disponibles se ha habilitado la lectura y escritura a través de la tarjeta de adquisición de datos. Adicionalmente, se registran las temperaturas de la cara caliente de la Peltier y la temperatura ambiente, como simples indicadores de que el proceso funciona correctamente.

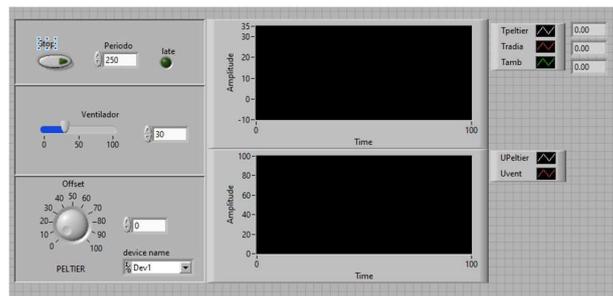


Imagen 19: Interfaz programa identificación proceso.

En la imagen anterior se observa el interfaz con el cual se interactúa para llevar a cabo la identificación. Es importante la configuración de periodo ya que se quiere modelar un sistema continuo con lecturas discretas. Se ha decidido tomar las muestras con un periodo de 100ms<sup>2</sup>. Después de configurar el nombre del dispositivo utilizado, la tarjeta de adquisición de datos, podemos configurar las variables controladas e iniciar la identificación.

El ventilador se mantendrá constante a un valor del 30%, mientras que la actuación sobre la Peltier se irá variando hasta conseguir una temperatura estable alrededor de los 0°C<sup>3</sup>. Una vez estabilizado el sistema se realizarán saltos de entre (-10,-5) %, volviendo siempre a la temperatura de reposo, 0°C, tal y como hará el proceso una vez integrado en el sistema. Una vez terminado el experimento, se presiona el botón de Stop para darlo por finalizado y el programa devuelve un fichero con los datos registrados.

<sup>2</sup> En el lazo abierto el tiempo de establecimiento ante un cambio de referencia de tipo escalón en los rangos estudiados es de entre 10 y 15 segundos, la selección del periodo se ha hecho teniendo en cuenta intervalos de al menos  $T_e/100$ .

<sup>3</sup> Por una experiencia previa se sabe que el proceso tiene una ganancia variable y que un buen rango de trabajo es entre los 2°C y los 10°C, igualmente se procedió a la identificación experimental.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

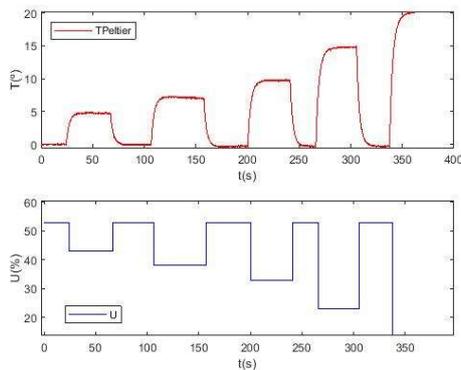


Imagen 20: Datos Experimentales. Fuente: Elaboración propia.

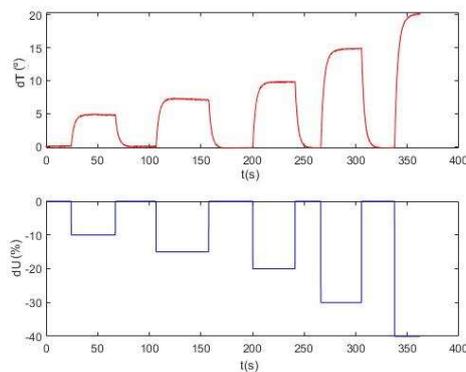


Imagen 21: Datos experimentales normalizados. Fuente: Elaboración propia.

Una vez comprobados y normalizados los datos se procede a la identificación propiamente dicha, se utiliza la herramienta de Matlab ident, incluida en el Toolbox de identificación de sistemas el cual proporciona a Matlab y Simulink funciones y bloques que permiten la construcción de modelos matemáticos de sistemas dinámicos a partir de los datos de entrada y de salida (MathWorks, 2022).

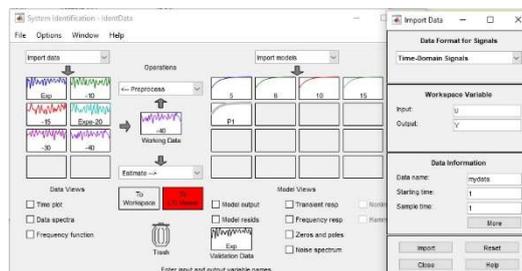


Imagen 22: Ident

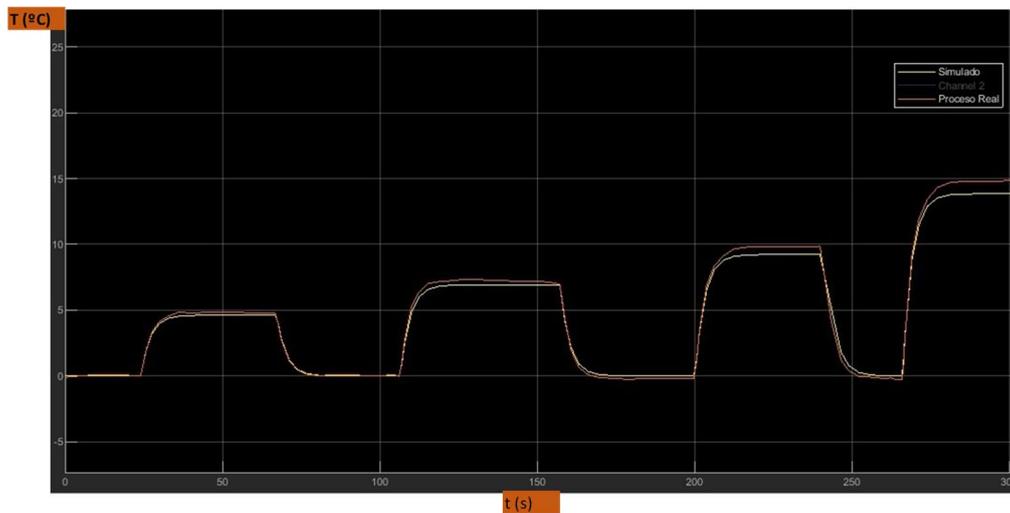
A partir de ella e introduciendo los datos normalizados de la variación de la variable controlada y de la variable de control, se consiguen los modelos para cada uno de los diferentes saltos quedando comprobado que el proceso posee una ganancia variable que hace variar la respuesta del proceso dependiendo del rango de trabajo. Esta ganancia es bastante estable hasta el salto de 0°C a 10°C, rango de interés por ser en el que se realizará el proceso, se procede a realizar

otro experimento en el cual todos los saltos se produzcan en el rango límite, de 0°C a 10°C por ser este el supuesto más crítico en la aplicación práctica, unas temperaturas demasiado elevadas podrían dañar la pieza.

Después de repetir el proceso, comparar los diferentes resultados y seleccionar el que más se ajusta al proceso real, se llega a la expresión que modela el proceso.

$$G(s) = \frac{-0,46156}{2,9208 \cdot s + 1}$$

*Ecuación 5: Proceso Modelado*



*Imagen 23: Validación del modelo mediante simulación Simulink.*

El diseño del PID para controlar la temperatura se analizará en otro apartado más adelante.

## 2.2 Funcionamiento del sistema

Una vez entendida la naturaleza de los medios que componen el sistema, se procede a explicar el funcionamiento general del conjunto.

### 2.2.1 Estructura de subsistemas



Imagen 24: Esquema conexión subsistemas. Fuente: Elaboración propia.

El papel fundamental de la estructura recae en el PLC M241, es el encargado de gestionar los modos de funcionamiento, el proceso de la estación, la creación del HMI junto con su visualizador web y de un servidor OPC-UA que sirve para conectar el resto de los elementos. Son varias las variables que deben ser intercambiadas entre el M241 y el programa de LabVIEW para identificar el proceso y realizar el control PID, esto se consigue gracias al servidor OPC-UA creado por el autómat. InfluxDB es una base de datos que permite la captación de datos en tiempo real por distintas vías, la elegida ha sido mediante telegraf, un agente escrito en GO que permite adaptar los datos de diversas fuentes al formato requerido por InfluxDB. El autómat TSX Premium está fuera del servidor OPC, pero recibe la información necesaria mediante el uso de la memoria compartida a través de una comunicación Modbus TCP/IP donde el ejerce como cliente y el M241 como servidor.

Para la colocación física de los elementos del sistema no se han realizado planos sino una distribución en planta a partir de la calibración previa.

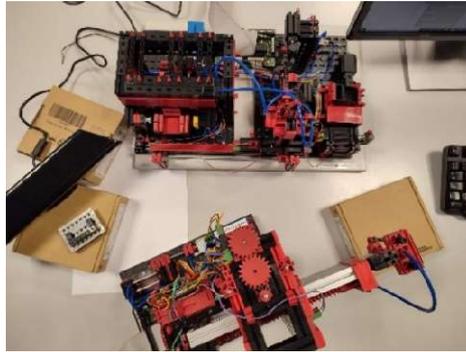


Imagen 25: Vista en planta del montaje. Fuente: Elaboración propia.

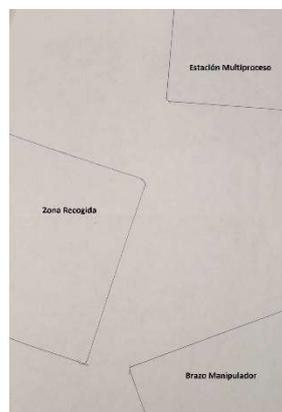


Imagen 26: Distribución en planta. Fuente: Elaboración propia.

La caja con el punto de recogida lleva una marca con la situación de este, y la cámara se colocará previo al proceso según las condiciones de iluminación para enfocarla óptimamente.

### 2.2.2 Modos de funcionamiento

Todo sistema automatizado requiere de una estructura de modos de funcionamiento que faciliten al operario no solo el control del proceso en sí, sino también de la alternancia correcta y segura entre diferentes tareas como pueden ser paradas a mitad de ciclo, paradas para el mantenimiento, el manejo manual del proceso o una parada de emergencia. Al igual que la herramienta del GRAFCET (*Grphe Fonctionnel de Commande Etape-Transition*) se convierte en 1982 en estándar AFNOR (*Agencia Nacional Francesa para la Estandarización*) (Ponsa Asensio & Vilanova Arbós, 2005) con el objetivo de normalizar y mecanizar el diseño de automatismos, nació la necesidad de unificar el proceso de diseño de la estructura de control de los distintos modos de funcionamiento dado que este proceso puede acabar siendo bastante complejo y en ocasiones difícil de interpretar por aquellos ajenos al diseño del proceso. De esta necesidad se crea la guía GEMMA (*Guide d'Etude des Modes de Marches et d'Arrets*) regulada por la norma nacional francesa UTE C 03-191 en 1993 (Ponsa Asensio & Vilanova Arbós, 2005).

La GEMMA divide los tipos de secuencias en 3, procedimientos de parada, de funcionamiento y de emergencia.

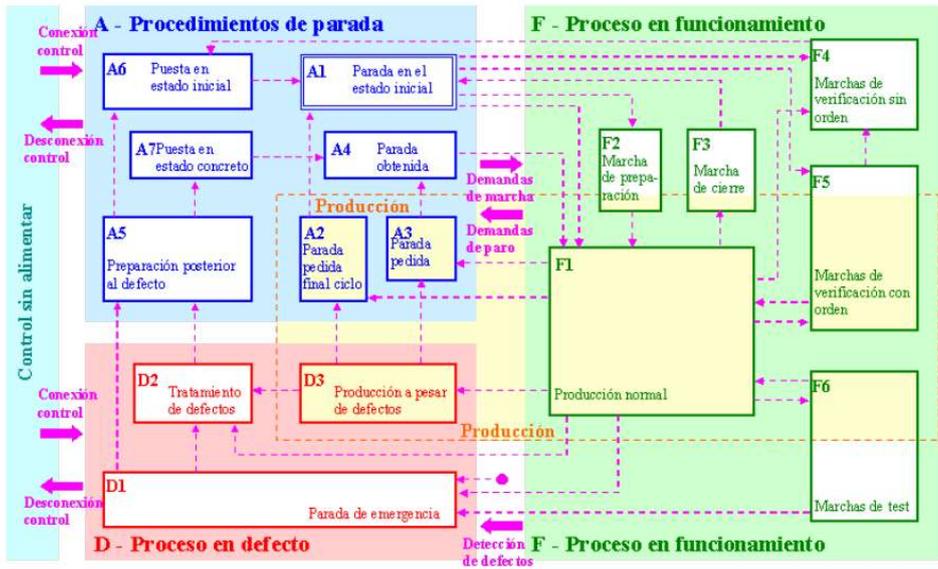


Imagen 27: Gráfico guía GEMMA. Fuente: Esquemas Guía GEMMA

Dependiendo de la complejidad del sistema se emplearán todos los bloques o únicamente los necesarios para su correcto funcionamiento, en el caso que nos ocupa la estructura GEMMA utilizada ha sido la siguiente:

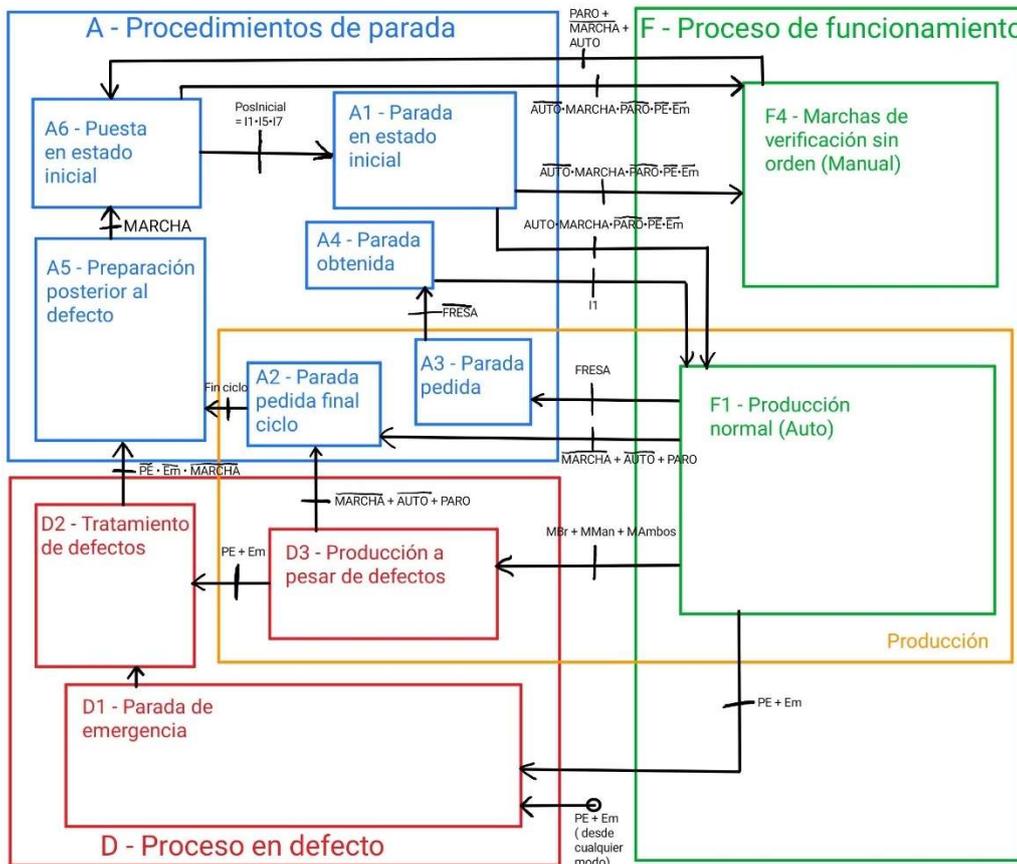


Imagen 28: Esquema guía GEMMA. Fuente: Elaboración propia.

El sistema se conectará siempre y cuando el interruptor de MARCHA este activo y existen dos modos de desconectarlo, desactivando MARCHA, con lo cual el sistema se desconectará una vez todos los procesos activos hayan terminado, o desde el estado de emergencia, solventados los problemas que hayan hecho saltar las alarmas y desconectando MARCHA y el pulsador con enclavamiento PE.

Los estados A6 y A1 se corresponden con la puesta del sistema en la posición de reposo, y se pasará por ellos cada vez que este se conecte, bien desde 0 o bien después de una desconexión debida a un defecto, también pasaremos por ellos antes de reconectar el modo automático, bien vengamos del modo manual o del semiautomático. La posición de reposo es la que podemos observar en la imagen 25, el brazo tocando los tres finales de carrera, el manipulador y la mesa de mecanizado en su posición común, el horno cerrado y apagado y la bandeja fuera.

El resto de los estados se encargan de gestionar los distintos modos de funcionamiento:

- **Producción Normal o modo automático:** Es el modo de funcionamiento principal en el cual se desarrollará en procesado de las piezas en la estación multiproceso, empezará con el operario colocando la pieza en el punto de recogida, donde será identificado según su color y asignado a un proceso, seguidamente el brazo recogerá esta pieza, siempre y cuando el horno esté disponible y no haya una pieza terminada. El brazo depositará la pieza en la bandeja del horno donde será tratada una vez el horno alcance la temperatura correspondiente. Una vez horneada, el manipulador de la estación llevará la pieza a la mesa de mecanizado donde será procesada y depositada en la cinta transportadora. Una vez en la posición final, el brazo recibe la orden de recogerla, teniendo este proceso prioridad respecto de recoger una pieza nueva. La pieza será colocada en una zona u otra en función del tratamiento realizado. Este modo se identifica con el procedimiento F1 y con el A2, el cual corresponde con el estado que garantiza que ninguna otra acción será realizada hasta que los procesos activos hayan terminado. Se accede a él con el sistema en marcha y en la situación de reposo y moviendo el selector a la posición "AUTO".
  - **FRESA:** Dentro del modo automático cabe la posibilidad de tener que parar la estación de mecanizado y de retirar el manipulador de la zona por seguridad, bien por mantenimiento o bien para intercambiar la herramienta. Es por ello que si se activa la señal FRESA el proceso productivo se pausará en cuanto sea posible (si hay una pieza siendo mecanizada, el proceso terminará y no se admitirá un nuevo ciclo) y una vez el operario haya terminado la operación correspondiente, desactivará la señal y el proceso se reanudará en el punto en el que se quedó (si había una pieza en el horno esta saldrá una vez terminado el proceso y esperará a que se reanude el sistema). Este submodo se corresponde con los procedimientos A3 y A4.
- **Modo Manual:** Este modo permite al operario manipular a su voluntad cualquier elemento del sistema, incluida la temperatura del horno. Es un modo muy necesario debido a que puede haber necesidades imprevistas que precisen de la supervisión y control directos del operario, puede utilizarse en tareas de mantenimiento, para hacer comprobaciones de funcionamiento o para recalibrar el sistema. Se corresponde con el

procedimiento F4 y se accede a él mientras el sistema esté en funcionamiento y desactivando el modo Auto, para pasar al modo Man. Debe tenerse en cuenta que antes de entrar en funcionamiento este modo deberán terminar todos los procesos activos.

- **Modo semiautomático:** Este modo se ha considerado como un proceso de defecto en el cual no se detiene la producción, su función es la de, en caso de que el operario detecte una avería, algún peligro externo o pueda darse una situación delicada; este pueda tomar los mandos del manipulador, del brazo o de ambos para hacerse cargo personalmente del problema sin que este afecte a la cadena de producción. Se accede a él pulsando alguno de los interruptores disponibles para este modo, y solo con el sistema en modo "AUTO". Al interrumpir el proceso automático el operario deberá tener constancia de que proceso está realizándose a las piezas. Esto puede hacerlo fácilmente antes de tomar los mandos consultando los indicadores luminosos del HMI o comprobando la base de datos. Este modo se corresponde con el procedimiento D3 y al igual que el funcionamiento normal, al finalizar el proceso, se deberá pasar por A2 para asegurar que todos los procesos han terminado satisfactoriamente, de lo contrario, si se detecta alguna incidencia se pasará al modo de emergencia y se parará el funcionamiento completo.
  
- **Modo de emergencia:** Por último, el modo de emergencia es el que garantiza que en caso de avería o problema de fuerza mayor el proceso entero se podrá paralizar instantáneamente. Como en el proceso intervienen manipuladores que trasladarán piezas a una cierta velocidad no debe permitirse que al interrumpir el funcionamiento estas caigan al vacío, pudiendo estas ser dañadas o resultar un riesgo para los operarios, es por ello por lo que, aunque no lo estuvieran en un principio, se activan los compresores y las ventosas de ambos manipuladores. Además, para evitar que la pieza que está siendo tratada térmicamente se dañe, el horno abre sus puertas, se apaga y se extrae la bandeja, será decisión del operario si retirarla o terminar el proceso manualmente más adelante, en función de la gravedad del problema y de la importancia del producto. Respecto al resto de variables, pueden ser manipuladas con libertad como si del modo manual se tratara, con las excepciones mencionadas, las succiones estarán activas con lo que no podrán recogerse nuevas piezas y no podrán introducirse piezas en el horno. Los procedimientos implicados son el D1, D2 y A5. Se accede a él desde cualquiera de los otros procesos descritos, y se hará cuando se pulse el pulsador con enclavamiento PE, o el propio sistema detecte un error y active la señal Em. Para evitar cualquier error a la hora de salir del modo de emergencia se requerirá que tanto el pulsador PE como el interruptor de MARCHA queden liberados, además de extinguirse la señal Em.

## 2.3 Manual del programador

Ya se ha visto cual debe ser el funcionamiento del sistema planteado, así como sus componentes y especificaciones. En este capítulo se entrará al detalle de cómo se ha hecho, enseñando las configuraciones utilizadas para cada subsistema. Si se desea conocer más en profundidad sobre el automatismo se dispone de la explicación de su proceso de diseño en el ANEXO I.

### 2.3.1 Programación PLCs

#### Arquitectura y Variables

Para entender el funcionamiento es primordial entender donde disponemos de las variables y como accedemos a ellas dado que las mecánicas pueden cambiar de un software a otro.

Tanto el software SoMachine como el UnityPro M no precisan que se defina la dirección donde se almacenará la nueva variable creada, sino que el mismo gestiona cual es la ubicación óptima para ella. Además, en ambos podemos declarar variables tanto a nivel global como a nivel local.

La principal diferencia la encontramos en el direccionamiento de memoria, en ambos existe el tamaño de dirección %MW (Word) pero la subdivisión de este en cada uno de los dos PLCs es diferente, mientras que en el TSX Premium, una dirección %MW está formada por 16 bits; en el M241 esa misma dirección %MW está formada por 2 bytes de 8 bits, la diferencia puede parecer trivial pero debe tenerse en cuenta si se quiere trabajar utilizando la memoria compartida, dado que en este caso sí que será necesario direccionar las variables comunes y deberemos conocer las equivalencias entre los bits de ambos autómatas.

DIRECCIONAMIENTO DE MEMORIA				
		TSX Premium	M241	
Palabras Dobles	Palabras	Bits	Bytes	Bits
%MDO	%MW0	%MW0.0	%MBO	%MX0.0
		%MW0.1		%MX0.1
		%MW0.2		%MX0.2
		%MW0.3		%MX0.3
		%MW0.4		%MX0.4
		%MW0.5		%MX0.5
		%MW0.6		%MX0.6
		%MW0.7		%MX0.7
	%MW0.8	%MB1	%MX1.0	
	%MW0.9		%MX1.1	
	%MW0.10		%MX1.2	
	%MW0.11		%MX1.3	
	%MW0.12		%MX1.4	
	%MW0.13		%MX1.5	
	%MW0.14		%MX1.6	
%MW0.15	%MX1.7			
%MDO	%MW1	%MW1.0	%MB2	%MX2.0
		%MW1.1		%MX2.1
		%MW1.2		%MX2.2
		%MW1.3		%MX2.3
		%MW1.4		%MX2.4
		%MW1.5		%MX2.5
		%MW1.6		%MX2.6
	%MW1.7	%MX2.7		
	%MW1.8	%MB3	%MX3.0	
	%MW1.9		%MX3.1	
	%MW1.10		%MX3.2	
	%MW1.11		%MX3.3	
	%MW1.12		%MX3.4	
	%MW1.13		%MX3.5	
	%MW1.14		%MX3.6	
%MW1.15	%MX3.7			

Tabla 5: Direccionamiento de memoria. Fuente: Elaboración propia.

**PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA**

Como se verá en el apartado dedicado a la comunicación, desde el TSX puede accederse a cualquier casilla de memoria del otro PLC para realizar una acción de lectura o escritura, copiando o compartiendo de una de sus propias casillas la información correspondiente. Para facilitar la localización de las variables estas se han colocado en el bit equivalente siguiendo la estructura de la tabla 5. De esta forma si quisiéramos compartir una variable en el bit %MW1.8 del TSX está aparecerá en el bit %MX3.0 del M241<sup>4</sup>.

Pensando en esta arquitectura se ha diseñado el almacenado de variables en ambos autómatas.

M241 TM241CE40R				TSX PREMIUM		
Variable	Casilla de memoria	Descripción	Tipo	Variable	Casilla de memoria	Tipo
PiezaNueva	%MX14.0	Interna	BOOL	PiezaNueva	%MW7.0	BOOL
Rojo	%MX14.1	Interna	BOOL			
Azul	%MX14.2	Interna	BOOL			
Blanco	%MX14.3	Interna	BOOL			
PiezaEnHorno	%MX14.4	Interna	BOOL	PiezaEnHorno	%MW7.4	BOOL
HRojo	%MX14.5	Interna	BOOL			
HAzul	%MX14.6	Interna	BOOL			
HBlanco	%MX14.7	Interna	BOOL			
PiezaEnManipulador	%MX15.0	Interna	BOOL			
MRojo	%MX15.1	Interna	BOOL			
MAzul	%MX15.2	Interna	BOOL			
MBlanco	%MX15.3	Interna	BOOL			
PiezaEnFresado	%MX15.4	Interna	BOOL			
FRojo	%MX15.5	Interna	BOOL			
FAzul	%MX15.6	Interna	BOOL			
Fblanco	%MX15.7	Interna	BOOL			
PiezaAcabada	%MX16.0	Interna	BOOL	PiezaAcabada	%MW8.0	BOOL
ARojo	%MX16.1	Interna	BOOL	ARojo	%MW8.1	BOOL
AAzul	%MX16.2	Interna	BOOL	AAzul	%MW8.2	BOOL
ABlanco	%MX16.3	Interna	BOOL	ABlanco	%MW8.3	BOOL
MARCHA	%MX16.4	Interna	BOOL	MARCHA	%MW8.4	BOOL
AUTO	%MX16.5	Interna	BOOL			
PARO	%MX16.6	Interna	BOOL			
FRESA	%MX16.7	Interna	BOOL	FRESA	%MW8.7	BOOL
PE	%MX17.0	Interna	BOOL			
MBr	%MX17.1	Interna	BOOL	MBr	%MW8.9	BOOL
MMan	%MX17.2	Interna	BOOL			
Mambos	%MX17.3	Interna	BOOL			
Em	%MX17.4	Interna	BOOL			
VArriba	%MX17.5	Interna	BOOL	VArriba	%MW8.13	BOOL
VAbajo	%MX17.6	Interna	BOOL	VAbajo	%MW8.14	BOOL
RHorizontal	%MX17.7	Interna	BOOL	RHorizontal	%MW8.15	BOOL
AHorizontal	%MX18.0	Interna	BOOL	AHorizontal	%MW9.0	BOOL
Giro Horario	%MX18.1	Interna	BOOL	Giro Horario	%MW9.1	BOOL
Giro Antihorario	%MX18.2	Interna	BOOL	Giro Antihorario	%MW9.2	BOOL
Compresor	%MX18.3	Interna	BOOL	Compresor	%MW9.3	BOOL
Ventosa	%MX18.4	Interna	BOOL	Ventosa	%MW9.4	BOOL
P60	%MX18.5	Interna	BOOL	P60	%MW9.5	BOOL
P61	%MX18.6	Interna	BOOL	P61	%MW9.6	BOOL
P62	%MX18.7	Interna	BOOL	P62	%MW9.7	BOOL
P63	%MX19.0	Interna	BOOL	P63	%MW9.8	BOOL
P64	%MX19.1	Interna	BOOL	P64	%MW9.9	BOOL
P66	%MX19.2	Interna	BOOL	P66	%MW9.10	BOOL
X124 (pieza colocada)	%MX24.0	Interna	BOOL	X124 (pieza colocada)	%MW12.0	BOOL

Tabla 6: Variables compartidas. Fuente: Elaboración propia.

En la tabla anterior aparecen todas las variables internas que han sido direccionadas por ser susceptibles de ser compartidas, como veremos en el siguiente apartado todas las variables entre los bits %MX14.0 y %MX19.2 están siendo leídas por el TSX Premium y por tanto también las tiene en memoria, sin embargo, se han mostrado únicamente aquellas que realmente son utilizadas por el TSX para llevar a cabo sus procesos.

<sup>4</sup> Según la comunicación realizada, véase en el siguiente apartado.

Aquellas variables que únicamente están relacionadas con procesos internos o que, en el caso del M241, están destinadas a la comunicación con el servidor OPC-UA no tienen la necesidad de ser direccionadas ni de ser declaradas como globales.

En el caso del TSX Premium, las únicas variables adicionales a las ya vistas han sido las vinculadas a transiciones, han sido declaradas variables globales sin direccionamiento concreto.

En cuanto al M241, aquellas variables destinadas a transiciones o a la gestión del HMI se han declarado como locales en sus respectivas tareas, mientras que el resto de las variables internas sí que han sido grabadas en la lista de locales, pero sin direccionamiento fijo.

Variable	Casilla de memoria	Descripción	Tipo
SemiMan		Interna	BOOL
Pulsador		Interna	BOOL
F1		Interna	BOOL
F2		Interna	BOOL
F3		Interna	BOOL
S1		Interna	BOOL
S2		Interna	BOOL
S3		Interna	BOOL
S4		Interna	BOOL
S5		Interna	BOOL
S6		Interna	BOOL
S7		Interna	BOOL
S8		Interna	BOOL
S9		Interna	BOOL
S10		Interna	BOOL
S11		Interna	BOOL
S12		Interna	BOOL
S13		Interna	BOOL
S14		Interna	BOOL
S15		Interna	BOOL
OPCAzul		OPC	BOOL
OPCRojo		OPC	BOOL
OPCBlanco		OPC	BOOL
OPCHAzul		OPC	BOOL
OPCHRojo		OPC	BOOL
OPCHBlanco		OPC	BOOL
TemperaturaC		OPC	REAL
Tref		OPC	REAL
Uman		OPC	REAL
U		OPC	REAL
REstable		OPC	BOOL
HornoAuto		OPC	BOOL
RDB		OPC	REAL
ADB		OPC	REAL
BDB		OPC	REAL

Tabla 7: Variables Internas M241. Fuente: Elaboración propia.

Por último, la asignación de las entradas y salidas del autómatas se ha hecho siguiendo las instrucciones del personal del laboratorio, en función del cableado de las maquetas.

**PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA**

M241 TM241CE40R			
Variable	Casilla de memoria	Descripción	
Entradas	I1	%IX0.4	Mesa en Manipulador
	I2	%IX0.5	Mesa en Cinta
	I3	%IX0.6	Pieza Acabada
	I4	%IX0.7	Mesa en Sierra
	I5	%IX0.0	Manipulador en Mesa
	I6	%IX1.2	Alimentador dentro
	I7	%IX0.1	Alimentador fuera
	I8	%IX1.3	Manipulador en Horno
	I9	%IX0.2	Pieza en Alimentador
Salidas	Q1	%QX0.4	Giro Mesa Horario
	Q2	%QX0.5	Giro Mesa Antihorario
	Q3	%QX0.6	Avance Cinta
	Q4	%QX0.7	Sierra
	Q5	%QX1.0	Alimentador dentro
	Q6	%QX1.1	Alimentador fuera
	Q7	%QX1.2	Manipulador hacia Horno
	Q8	%QX1.3	Manipulador hacia Mesa
	Q9	%QX1.4	Habilitar Señales
	Q10	%QX1.5	Compresor
	Q11	%QX1.6	Ventosa
	Q12	%QX1.7	Manipulador Abajo
	Q13	%QX0.0	Puerta Arriba
	Q14	%QX0.1	Empujador Mesa
	Q15	%QX0.2	Luz Horno

Tabla 8: Entradas y salidas M241. Fuente: Procesos FischerTechnik, LAC.

TSX PREMIUM			
Variable	Casilla de memoria	Descripción	
Entradas	B1	%IO.2.0	Encoder Vertical
	B3	%IO.2.1	Encoder Horizontal
	B5	%IO.2.2	Encoder Giratorio
	I1	%IO.2.4	FC Vertical
	I2	%IO.2.5	FC Horizontal
	I3	%IO.2.6	FC Giro
Salidas	Q1	%QO.2.16	Vertical Arriba
	Q2	%QO.2.17	Vertical Abajo
	Q3	%QO.2.18	Retroceso Horizontal
	Q4	%QO.2.19	Avance Horizontal
	Q5	%QO.2.20	Giro Horario
	Q6	%QO.2.21	Giro Antihorario
	Q7	%QO.2.22	Compresor
	Q8	%QO.2.23	Ventosa
	Q9	%QO.2.24	Habilitar Señales

Tabla 9: Entradas y salidas TSX. Fuente: Procesos FischerTechnik, LAC

En cuanto a la funcionalidad de las variables, las almacenadas entre %MX14.0 y %MX16.3 se utilizan para almacenar información sobre donde está la pieza y que tipo es, cuando se expliquen los programas, estas serán las variables gestionadas por la tarea “lógica color” del M241.

Las variables entre %MX16.4 y %MX17.4 son las que se utilizan para los cambios entre modos de funcionamiento. Aquellas que quedan entre 17.5 y 18.4 y el resto de las variables internas se utilizan directamente para el control manual del proceso. Las variables restantes se utilizan para intercambiar información entre ambos de PLCs sobre las etapas en funcionamiento importantes, en el caso de las direccionadas; y las asociadas a la comunicación OPC son aquellas que provienen o se utilizan en el programa de LabView para la identificación del proceso o el control PID y se explicarán más adelante.

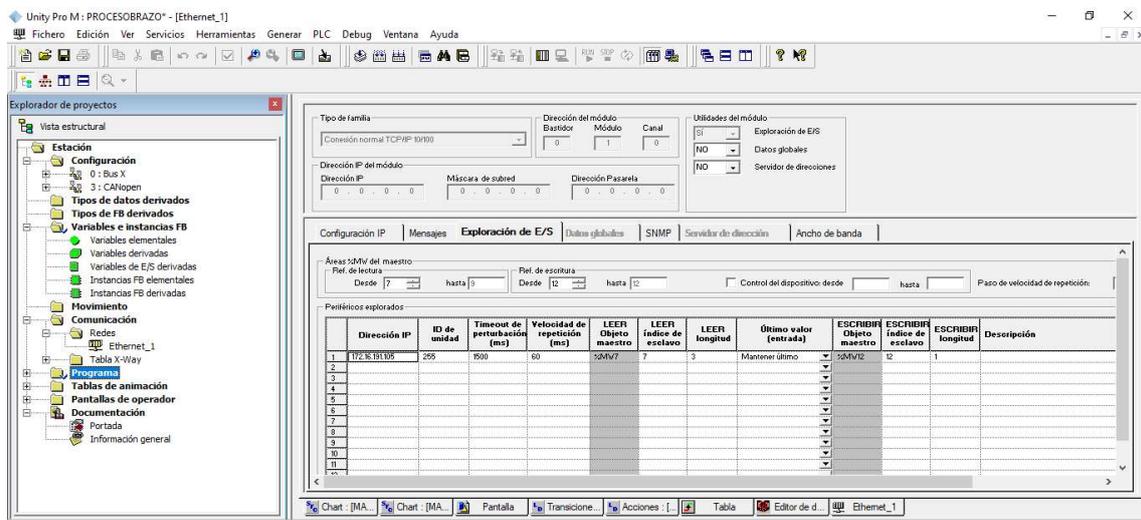
### Comunicación

La comunicación entre ambos PLCs una vez entendida la relación entre sus direccionamientos resulta muy sencilla. Se ha utilizado la memoria compartida mediante la conexión Modbus TCP/IP dado que ambos PLCs implementan dicho protocolo de forma nativa. Se ha utilizado el TSX Premium como PLC cliente, las variables que este necesita las leerá en los direccionamientos del %MX14.0 al %MX19.2 del M241, es decir, tendrá que leer de %MW7 a %MW9; en estas variables el brazo recibe información sobre en qué estado se encuentra el proceso del horno para poder actuar en consecuencia, así como las órdenes sobre sus actuadores para funcionar

**PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA**

en el modo manual. Hay una única variable que el horno necesita del brazo, está es la X124, la etapa en que el brazo ha colocado una pieza terminada en la zona de almacenaje; esta será escrita en el bit %MX24.0, es decir se escribirá la palabra %MW12.

Esta comunicación se configura fácilmente desde el mismo programa del UnityPro M, abriendo el apartado Comunicación del proyecto y dirigiéndose a Redes→Ethernet\_1, se abre una ventana con diversas configuraciones de las cuales la que interesa es la de Exploración de E/S, allí se activa la Exploración E/S y se configura la IP del PLC servidor, en este caso el 172.16.191.105 y las casillas de memoria a leer y escribir en este caso se lee desde la 7 hasta la 9 y se escribe sobre la 12.



*Imagen 29: Comunicación Modbus TCP/IP. Fuente: Elaboración propia.*

El otro medio de comunicación utilizado es el servidor OPC-UA. Este es creado por el mismo PLC M241, que pone las variables anteriormente mencionadas para que otros dispositivos se conecten a él mediante la dirección IP del autómatas y el puerto seleccionado, en este caso: "opc.tcp://172.16.191.105:4840"

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

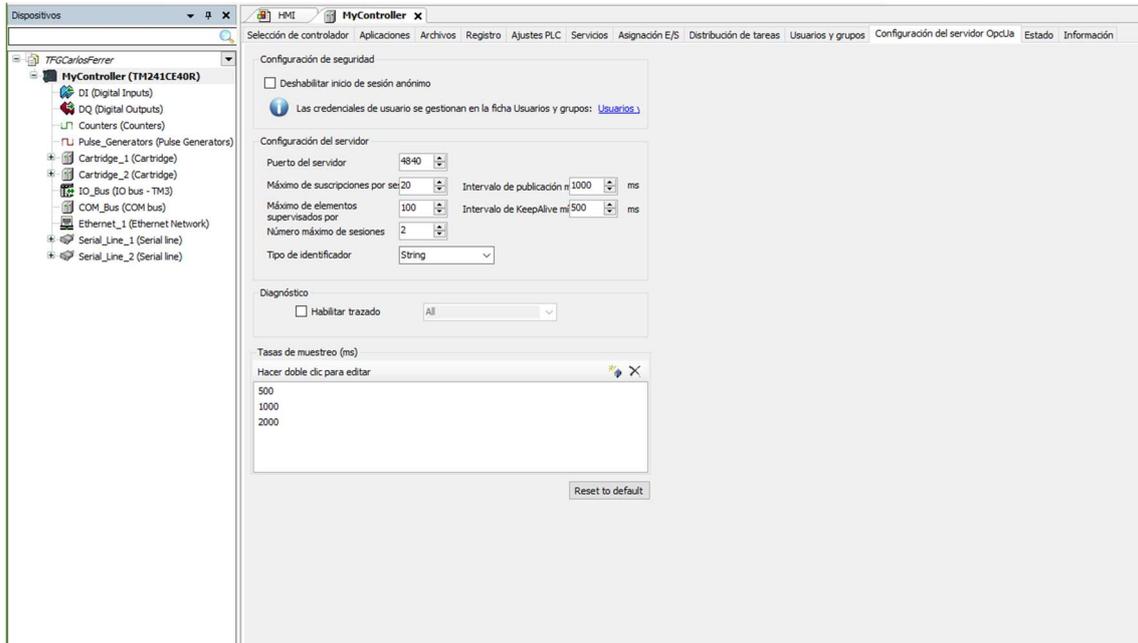


Imagen 30: Configuración servidor OPC-UA. Fuente: Elaboración propia.

Puede configurarse el servidor OPC-UA desde la ventana Dispositivo→MyController→Configuración del servidor OpcUa. Para el proyecto que nos ocupa se ha mantenido el puerto 4840 que se asigna por defecto para esta conexión, así como los intervalos de publicación. El número máximo de sesiones es un componente a tener en cuenta, en este caso únicamente se conectarán dos subsistemas al servidor, la base de datos y el programa de LabView, pero si se quisiera conectar alguno más, como por ejemplo un cliente rápido, este número debería aumentarse para evitar el colapso del servidor.

Las variables están habilitadas tanto para lectura como para escritura, el caso particular de cada variable se estudiará en el apartado correspondiente a cada subsistema.

#### Estación – SoMachine

La programación del PLC M241 se ha realizado mediante el software SoMachine proporcionado por el fabricante y basado en Codesys, el cual permite programar en los lenguajes recogidos en el estándar **IEC 61131-3**, en concreto se han utilizado los lenguajes SFC, LD y ST (texto estructurado).

Las tareas a realizar por este autómatas serán no solo gestionar el funcionamiento de la estación multiproceso, sino también gestionar las etapas del programa de control del proceso del brazo, así como la información que se comparte con él. En él también se programan los bloques necesarios para definir las variables internas, controlar la comunicación mediante el servidor OPC-UA y gestionar el HMI.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

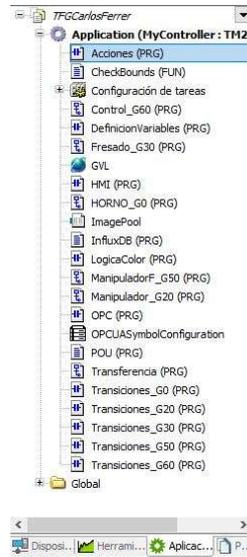


Imagen 31: Estructura de tareas.

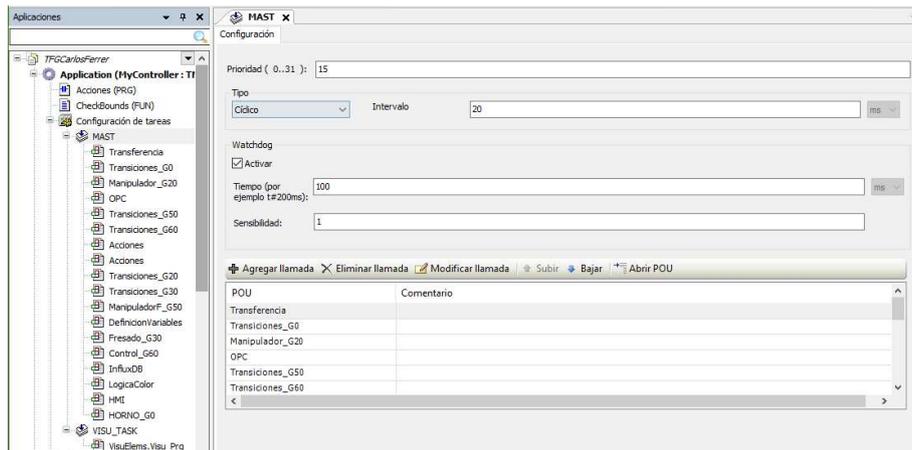


Imagen 32: Configuración tareas.

Puede observarse que la programación se ha hecho mediante bloques, también llamados POU ("Program Organization Unit"), y para que el programa los ejecute correctamente deben integrarse en una o varias tareas que se encargarán de ejecutarlos cíclicamente o cuando se active cierto evento. En este caso tenemos dos tareas, la primera y con más prioridad es aquella que se encarga del propio funcionamiento del sistema y la segunda, que se encargará de ejecutar el HMI y compartirlo en el servidor web, esta tarea tiene asignada una prioridad menor para asegurar primero el funcionamiento del sistema. Ambas tareas están configuradas para ejecutarse cíclicamente.

Como podemos observar en el ANEXO I para facilitar la comprensión y la programación del automatismo, se ha dividido el proceso en diversos subsistemas que funcionan con una secuencia prácticamente lineal y que se coordinan entre ellos gracias al subsistema llamado "Control" y la lógica del color que transmite la información sobre qué proceso y en qué estado se encuentra.

Estos subsistemas se han diseñado mediante el método GRAFCET por lo que el entramado de etapas de cada uno se ha programado mediante programas en lenguaje SFC, un lenguaje basado en la metodología GRAFCET. Para activar las acciones que se deben dar en cada etapa, así como las transiciones más complejas se han empleado programas en lenguaje LD. Para la activación de las variables internas se han utilizado programas en lenguaje de diagramas de bloques o texto estructurado según convenga.

El programa más interesante para analizar es “Control\_G60”, es el que se encarga de gestionar los modos de funcionamiento, siguiendo la guía GEMMA se ha creado un estado para cada uno de los procesos explicados allí.

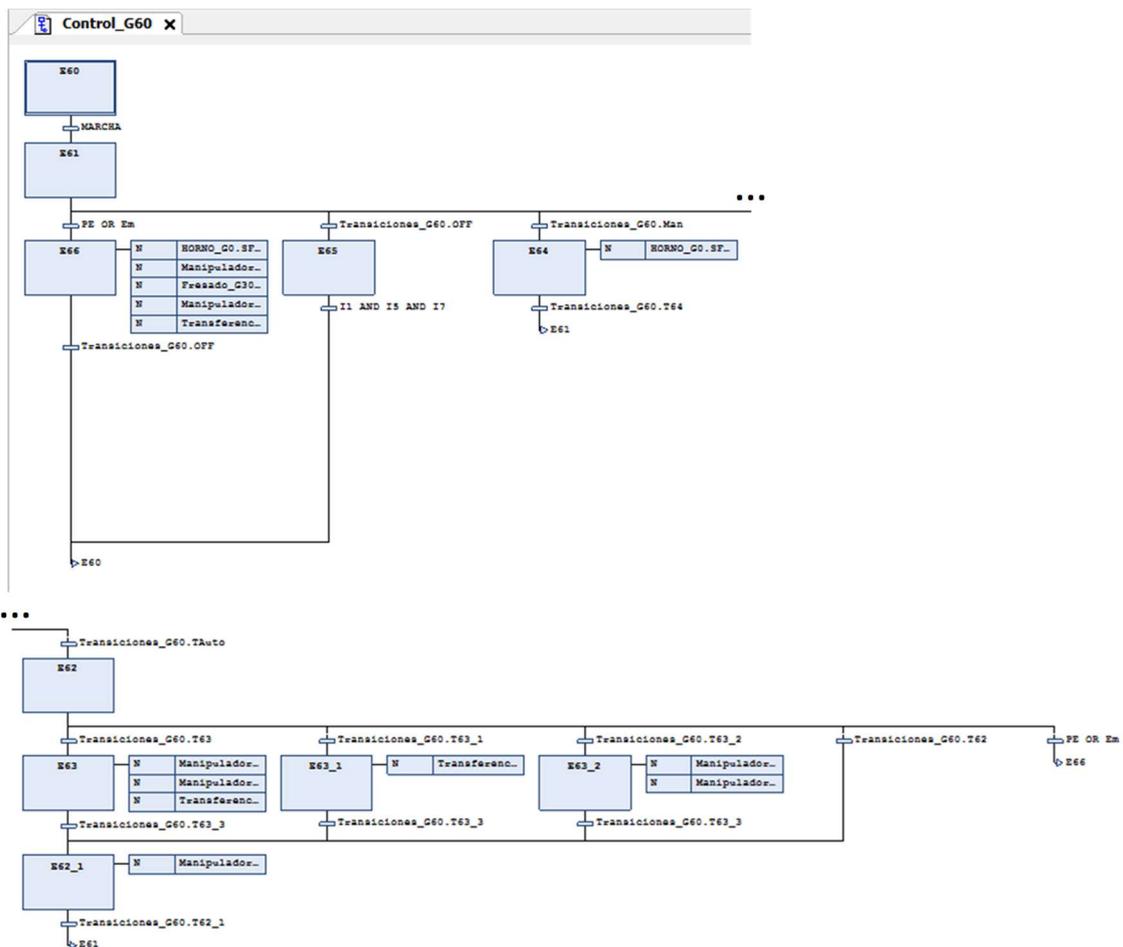


Imagen 33: POU Control\_G60.

La etapa E60 representa el sistema en desconexión, esperando que se active el interruptor MARCHA. Una vez esto ocurre se pasa la etapa E61, que se corresponde con la puesta en marcha inicial, como se verá en el POU “Acciones”, en ella se activan aquellas señales para llevar al sistema al estado inicial.

De la etapa E61 cuelgan los 4 diferentes escenarios que pueden plantearse una vez iniciado el sistema, con el interruptor PARO desactivado se podrá acceder al estado E62 (modo automático

o funcionamiento normal) y E64 (modo manual), según la posición del selector AUTO/MANUAL. Si el interruptor PARO está conectado no saldremos de este estado, a no ser que este se desconecte desactivando MARCHA (estado E65) o que se declare una emergencia (estado E66). Debe tenerse en cuenta que la prioridad máxima la tiene el estado de emergencia, E66, seguido del de desconexión, E65 y por último los dos modos de funcionamiento.

En el estado de emergencia se forzarán las variables Q6, Q10, Q11 y Q13 para asegurar que no se deja caer ninguna pieza y que el horno se desactiva, además se reiniciarán el resto de los programas a su etapa inicial mediante la variable interna del PLC SFCReset, la cual debe activarse expresamente en Proyecto→Configuración del Proyecto→SFC, esta variable reinicia tanto los pasos como las acciones de un programa SFC y ejecuta las acciones asociadas a este inmediatamente.

En el estado de desconexión se ejecutan las acciones necesarias para devolver el sistema al estado de reposo al igual que en E61. En el modo manual se habilita el forzado manual de las acciones de salida y se fuerza el programa del horno para evitar posibles incompatibilidades.

El modo automático da pie a que se ejecuten el resto de los subsistemas con el funcionamiento explicado, dentro de él podrá realizarse el mantenimiento mediante la variable FRESA. De este modo puede pasarse al control manual del brazo, el manipulador o ambos; este funcionamiento se corresponde con las etapas E63\_1, E63\_2 y E64 respectivamente, teniendo prioridad el control simultáneo. Una vez se salga de él o de alguno de los tres modos secundarios se deberá pasar por la etapa 62\_1 que permite ejecutarse los sistemas en activo, pero no permite que pasen a una nueva tarea. De cualquiera de estos cuatro estados se sale mediante la activación del estado de emergencia.

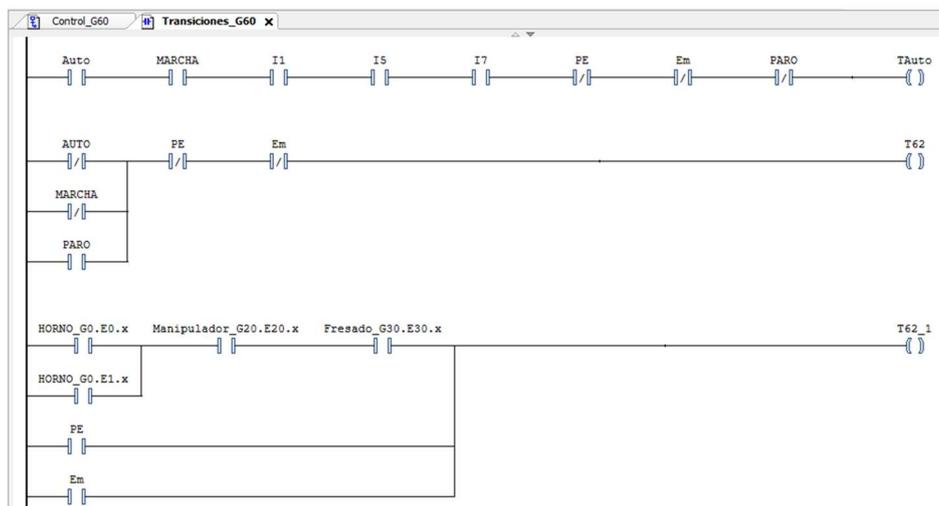


Imagen 34: fragmento POU Transiciones\_G60

En cuanto al resto de programas SFC, su programación es trivial si se ha entendido el diseño propuesto en el ANEXO I, puede observarse el ejemplo del programa "Horno\_G0". Este programa es puramente secuencial con la excepción de las etapas relacionadas con el proceso térmico, a las cuales se accederá en función del color en proceso. En este programa se ven algunas de las programaciones básicas utilizadas para el resto de los programas:

temporizadores, expresiones lógicas en función del estado del resto de procesos y variables que se activan desde el HMI para garantizar la continuidad del proceso en caso de funcionamiento semiautomático, como por ejemplo la variable “Pulsador”.

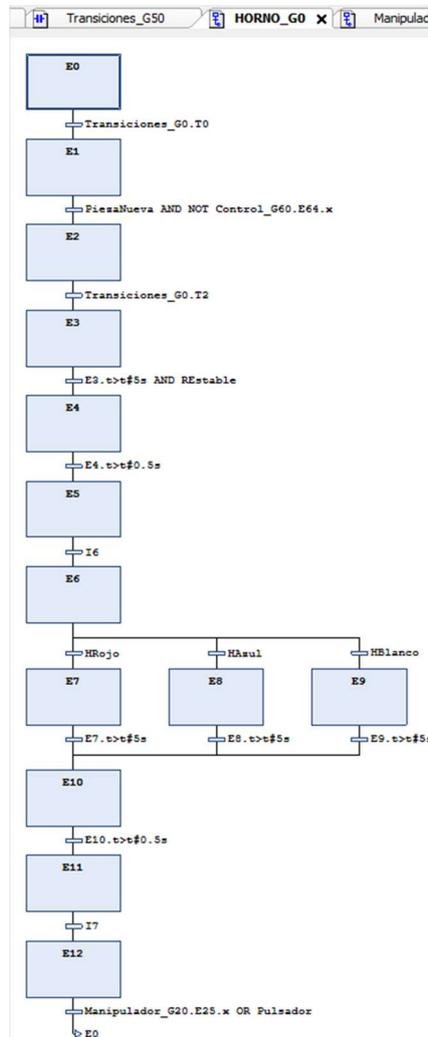


Imagen 35: POU Horno\_GO

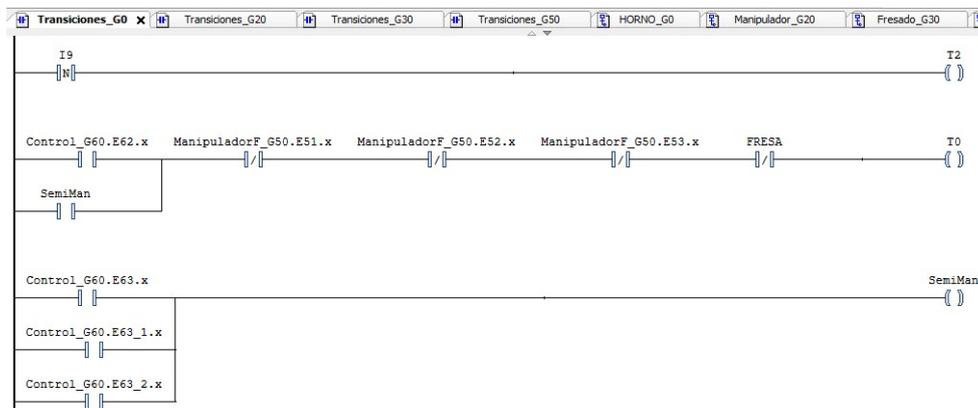


Imagen 36: fragmento POU Transiciones\_G0

Nótese que puede accederse a variables internas del programa como son “paso.x” o “paso.t” con las cuales podemos hacer referencia a si una etapa determinada está activa o cuanto tiempo lo ha estado, también se observa la nomenclatura a seguir para hacer referencia a variables o etapas de otro POU, primero se escribe el nombre del POU donde se encuentra la variable seguido de un punto, para luego hacer referencia a la variable en sí, de esta manera pueden también utilizarse variables locales en otros POUs.

Van a analizarse ahora el resto de POUs, programados como diagramas de contactos y texto estructurado, que se utilizan para activar las salidas y el resto de las variables del sistema.

### ACCIONES

Este POU se utiliza para activar las salidas en las etapas correspondientes y en el modo manual activadas desde el HMI, y para activar las variables utilizadas en el programa del brazo para pasar de un estado a otro del proceso de control.



Imagen 37: fragmento POU Acciones - Variables Control Brazo.

A continuación, se muestra la programación para la correcta activación de las salidas de algunas de ellas, ninguna de ellas se conectará a no ser que la variable de activación Q9 lo esté. Esto ocurrirá cuando se conecte el sistema, es decir cuando la etapa E60 está desconectada. Además, se observa la programación realizada para el modo manual, aparte de en las etapas correspondientes en el automatismo, también activará la salida la combinación del estado de emergencia y la variable “Sx” vinculada a los pulsadores e interruptores presentes en el HMI. Si la acción correspondiente tiene algún sensor o condición que deba limitar su movimiento<sup>5</sup>, esta también se añade en serie con el resto de las condiciones.

<sup>5</sup> Ej: Q1 (giro mesa horario) limitado por I2 (mesa en cinta).

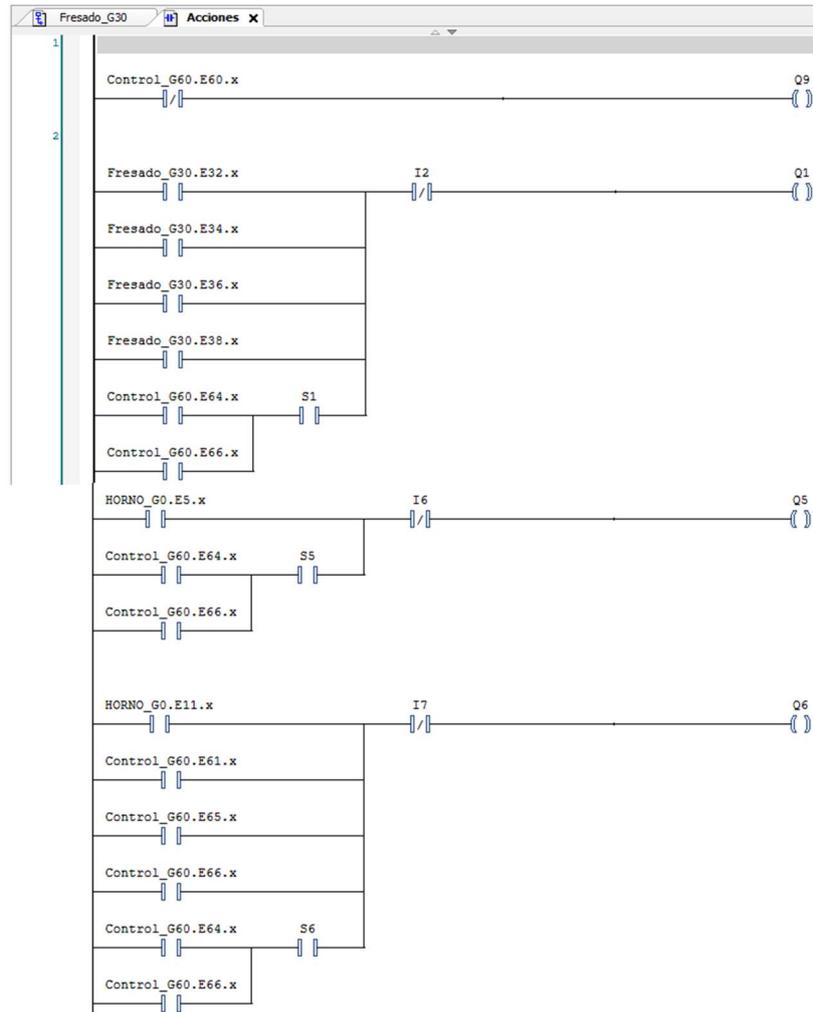


Imagen 38: fragmento POU Acciones – Activación.

### Definición Variables

En este POU se ha realizado la activación de ciertas variables que se utilizan para condicionar transiciones en el resto de los subsistemas y que dan información sobre el estado del sistema en funcionamiento automático. Su programación es trivial y las expresiones correspondientes están detalladas al final del ANEXO I.

### HMI

En la programación del HMI se utilizan ciertas variables para enseñar o esconder las ventanas en función del modo de funcionamiento activo algunas de estas variables se han programado en este POU.

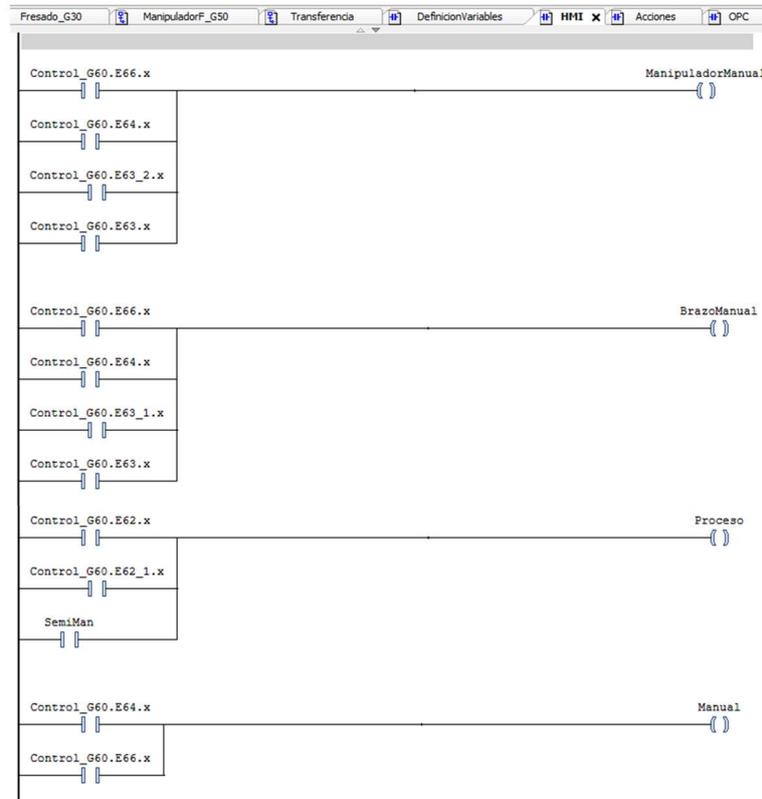


Imagen 39: POU HMI

### InfluxDB

En este programa en texto estructurado simplemente se implementa que cuando la variable de color de cada pieza en el horno está activa, la variable asociada coje el valor de 100. Esta variable asociada está compartida con la base de datos y se utiliza para crear un indicador bastante visual que sube su valor a 100 cuando la pieza de su color está siendo horneada y lo baja a 0 cuando no.

### OPC

En este POU se definen ciertas variables que son activadas en función de información proveniente del servidor, y otras que deben definirse antes de ser comunicadas con el servidor.

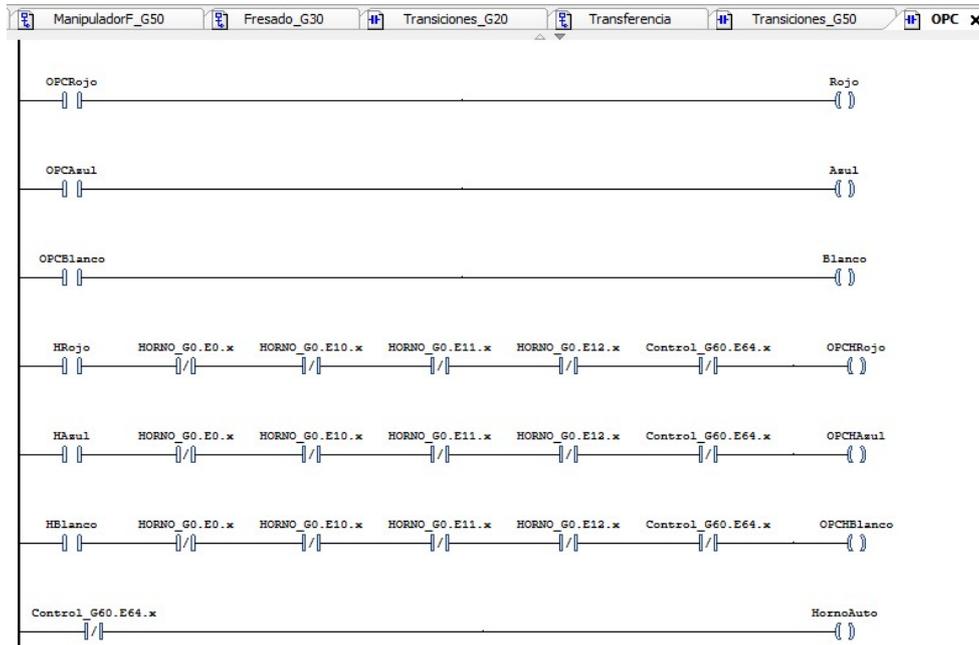


Imagen 40: POU OPC

Las variables OPCRojo/Azul/Blanco se activan cuando el programa de detección de color detecta la frecuencia correspondiente en el punto de interés, esta información es copiada en las variables del mismo nombre sin el prefijo OPC, que se utilizan en el horno para iniciar la detección del proceso, pero también en el brazo, con lo que deben ser compartidas y para ello almacenadas en la casilla de memoria correspondiente; podría haberse unido este proceso en una sola variable que se compartiera tanto en el servidor OPC como mediante la conexión MODBUS pero de esta manera queda más claro de donde proviene cada variable y en qué proceso interviene.

El proceso térmico debe comenzar nada más se detecte la pieza, por ello se define la variable OPCHRojo/Azul/Blanco, que se activarán cuando se detecte una pieza y hasta que esta sea tratada, estas variables serán leídas por el programa de LabView por ser necesarias para el proceso de control.

Por último, Horno Auto sirve para indicar cuando el proceso no está en modo manual, esta variable es precisada por el programa de control de temperatura para determinar cuando la selección de temperatura será manual y cuando automática, según el color de la pieza.

### Lógica Color

La “Lógica de Color” utilizada en este programa es el elemento vertebrador del sistema, gracias a ella puede conocerse en que procesos hay una pieza y de qué tipo es esta, gracias a ello pueden programarse las diferencias entre programas en cada proceso y los indicadores luminosos del HMI, que gracias a estar vinculados a estas variables muestran de una manera gráfica el estado del sistema.

Como se ha explicado en el ANEXO I las “variables de color” se han dividido según los subsistemas existentes, existen las variables Rojo, Azul y Blanco; que se activan cuando los

respectivos colores son detectados por la cámara Kinect y el programa asociado. Posteriormente existen tres variables llamadas como estas, pero con un prefijo delante que indica el proceso al que están vinculadas, H para el horno, M para el manipulador, F para el mecanizado y A para el brazo. El valor de estas variables se reinicia en la etapa inicial de que cada proceso y se activan cuando el proceso anterior traspasa los valores correspondientes, existe por tanto en cada proceso una etapa común entre procesos contiguos en la cual se comparte el color de la pieza tratada y a tratar.

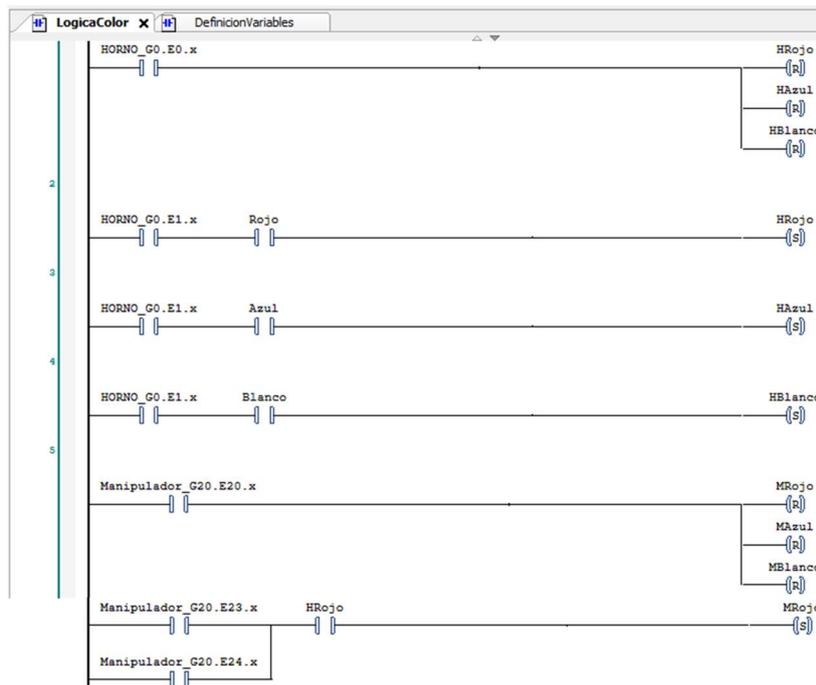


Imagen 41: fragmento POU LogicaColor

#### Brazo – Unity Pro M

La programación del PLC TSX-Premium se ha realizado mediante el software Unity Pro M proporcionado por el fabricante y basado al igual que el SoMachine en el estándar IEC 61131-3, en este caso únicamente se ha utilizado el lenguaje SFC y el LD.

Dado que este autómata únicamente controla el proceso del brazo, el programa contenido en él será mucho más sencillo que el mostrado para el M241. Estará compuesto únicamente por dos secciones SFC, una para el control del proceso, al igual que en el M241 y otra con las etapas del proceso automático; y por dos secciones LD, una para activar las acciones, y otra para las transiciones. También se ha programado en este PLC la comunicación mediante memoria compartida, en la cual este actúa como cliente, la configuración de esta ya está descrita en el punto 3.1.2. Dichas secciones se agrupan dentro de una misma tarea de ejecución cíclica.

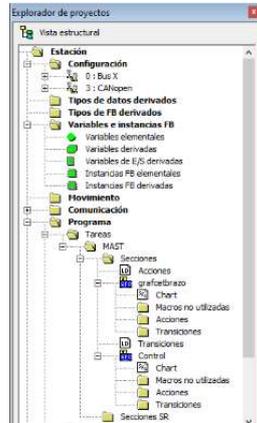


Imagen 42: Estructura Tareas

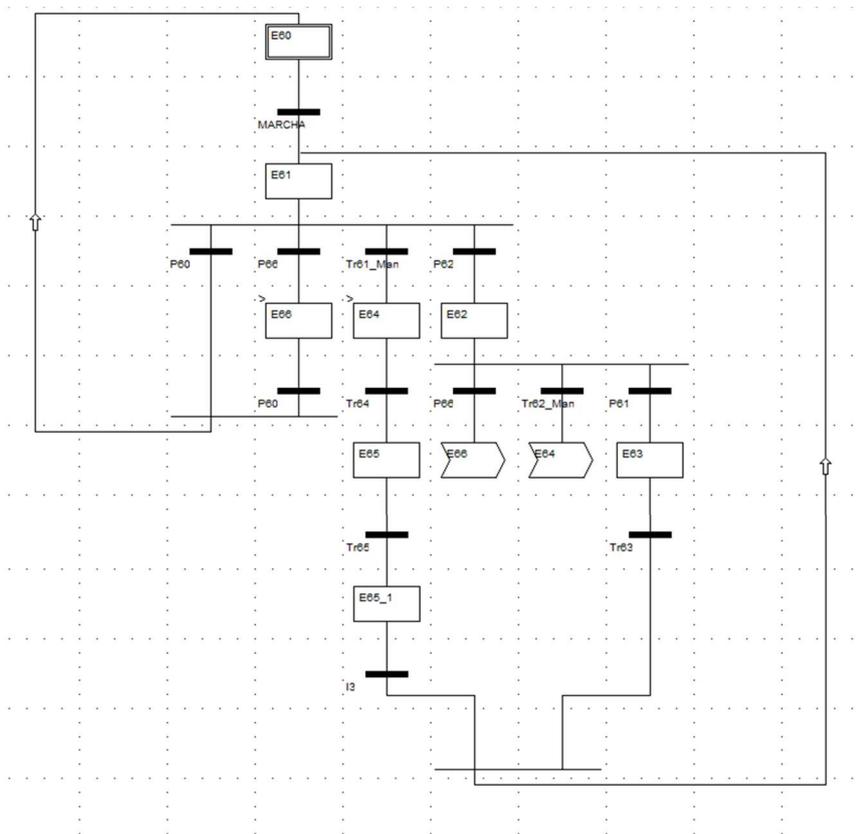


Imagen 43: Sección Control

Debe tenerse en cuenta que las variables P6x se corresponden con los estados de la tarea de control de M241, en función de estas funcionará la sección dedicada al control en este PLC. El flujo del proceso automático está programado tal y como se muestra en el ANEXO I, una vez el brazo se ha llevado a la posición inicial se queda en espera hasta que se activa la rama de recogida de la pieza en la posición inicial o en la final y después se ejecuta el proceso secuencialmente. Como novedad se tienen las transiciones en las cuales la condición es la llegada del brazo a la posición calibrada, ha de tenerse en cuenta el sentido del movimiento a la

hora de crear esta condición si el movimiento va desde la posición inicial hacia una extrema la condición a aplicar será del estilo “>=” mientras que si el sentido es hacia la posición de reposo la condición será la contraria, “<=”.

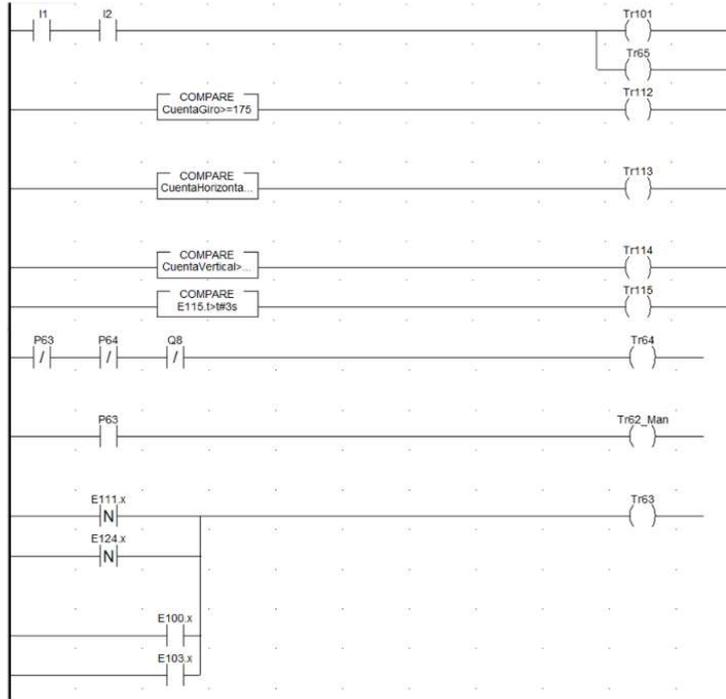


Imagen 44: Ejemplo transiciones sección grafcetzbrazo

Respecto a la programación de las salidas se ha hecho en el LD “Acciones” y en el mismo SFC como acciones asociadas a ciertas etapas, esto puede hacerse en este software dado que no crea incompatibilidades, como si pasaba con el SoMachine. También se ha incorporado el bloque de función SFCNTRL, el cual sirve para el control de las secuencias de ejecución. En este caso se ha utilizado para forzar la sección del proceso a su etapa inicial cuando se active la etapa de emergencia.

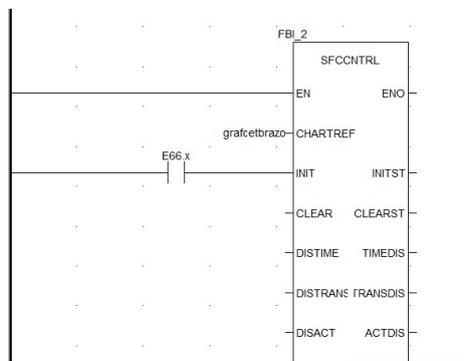


Imagen 45: Bloque SFCNTRL

Las variables utilizadas para que el programa sepa donde se encuentra la pieza y que proceso se le efectuará son: Pieza Nueva, Horno Ocupado, Pieza Acabada, para determinar su posición; y AROJO/Azul/Blanco, las cuales estarán activas desde que la pieza llega a la posición final en la estación, hasta que es depositada en la zona de almacenamiento.

### HMI

La programación del HMI se ha hecho con el SoMachine dado que este permite crear una visualización en un servidor web que puede ser ejecutada desde el navegador de cualquier dispositivo conectado a la misma red que el PLC.

Para la composición de este se ha querido conseguir un interfaz que sea intuitivo y que pueda mostrar la secuencia seguida por el proceso. Para ello se ha creado una barra de comandos en la parte superior de este, fija y con los controles para la puesta en marcha del sistema y la selección de los modos de funcionamiento. En ella tenemos el interruptor de MARCHA y el de PARO, el pulsador con enclavamiento para las emergencias, el selector entre modo Automático y Manual y los interruptores para los modos semiautomáticos y el de mantenimiento (FRESA).

En el centro del HMI tendremos una barra indicadora que nos mostrará en todo momento la temperatura del horno, a través de la variable TemperaturaC. El resto de los elementos mostrados por pantalla no serán fijos, sino que, aprovechando una de las configuraciones proporcionadas por el programa, se ocultarán o mostrarán en función de las necesidades del usuario.

Puede configurarse la visibilidad de los elementos desde Propiedades→Variables de estado→Invisibilidad, allí se colocará la variable que queremos que active el elemento, negada, ya que la función se utiliza para hacer invisibles los elementos cuando esta variable está activa y se requiere el efecto contrario.

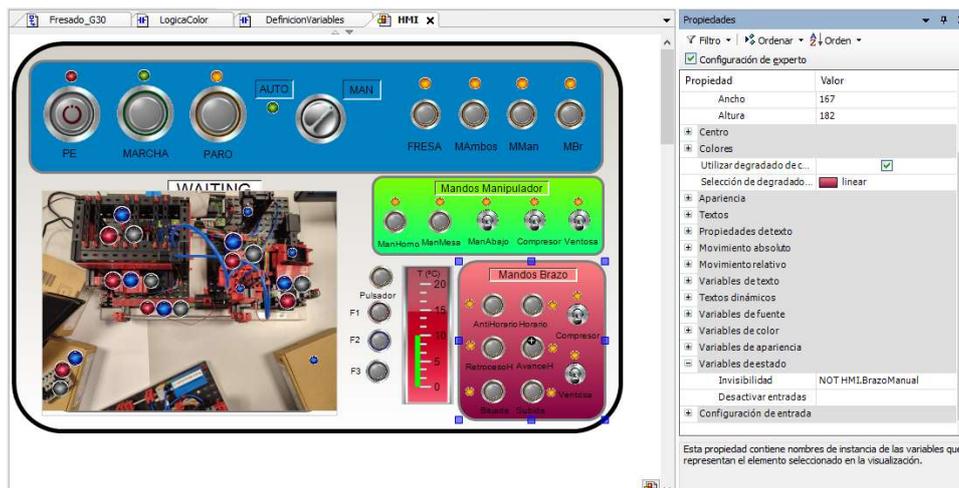


Imagen 46: Configuración invisibilidad HMI.

Las variables definidas en el LD "HMI" tienen precisamente esta función, la de arbitrar cuando se mostrará cada sección.

Con el sistema desconectado, MARCHA inactiva, únicamente se mostrará la barra de comandos fija y el indicador de temperatura.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

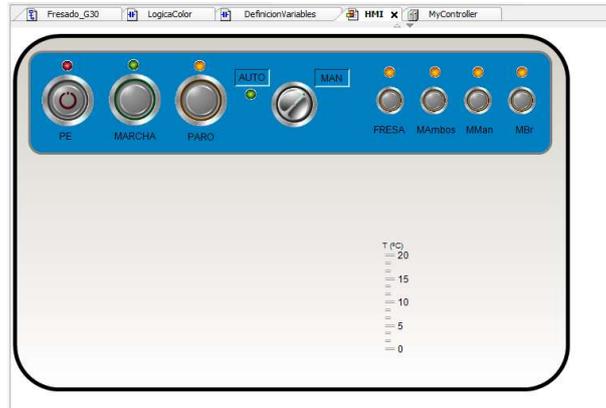


Imagen 47: HMI sistema desconectado.

Una vez se conecte MARCHA y mientras el sistema se prepara podrá observarse un mensaje que indica la necesidad de esperar hasta que se entre en el modo automático o el manual.



Imagen 48: HMI sistema en espera.

Una vez empiece el sistema a funcionar en modo automático, en el sector izquierdo se mostrará una imagen de la planta con indicadores luminosos que indican qué pieza está siendo procesada y en qué punto del proceso se encuentra, estos indicadores luminosos están vinculados a las variables explicadas en la sección “Lógica de Color”.

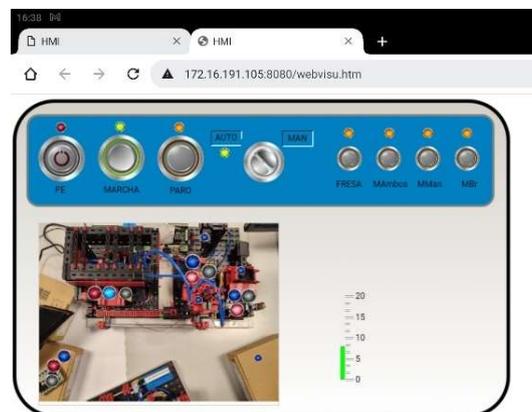


Imagen 49: HMI servidor web, proceso automático.

Los controles de los diferentes elementos que componen el proceso se mostrarán cuando se active su manejo en modo manual, el brazo y el manipulador se mostrarán además de en el funcionamiento manual, con el modo semiautomático.



Imagen 50: HMI modo semiautomático.

Otra opción que nos ofrece el SoMachine y que se aprovecha en el modo de emergencia, es el cambio de color de los elementos cuando se activa un “estado de alarma”, el cual se activa con una variable configurable.



Imagen 51: HMI configuración fondo alarma.

En el manual del usuario se entrará más a fondo en los distintos modos de funcionamiento que se han configurado.

Por último, se debe configurar el servidor web para poder acceder al HMI desde el navegador de cualquier dispositivo conectado a la misma red que el PLC. Desde herramientas se accede al Gestor de Visualización y se crea una nueva visualización web.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA



Imagen 52: Creación Visualización Web.

Una vez creada debe configurarse, se introduce el nombre del HMI a mostrar y el nombre de la web. También debe especificarse el tamaño del cliente, en este caso se ha dejado el tamaño por defecto, es decir, en 1280x1024 para que la representación no salga distorsionada.

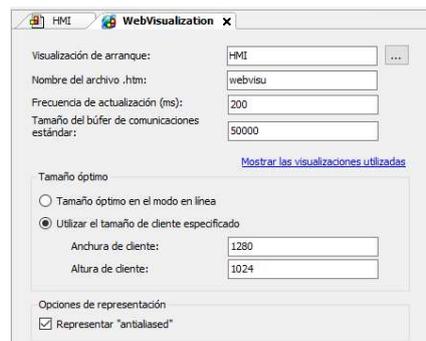


Imagen 53: Configuración Visualización Web

### 2.3.2 LabView

LabView es un entorno de programación gráfica que se utiliza para desarrollar sistemas de pruebas automatizadas de investigación, validación y producción (National Instruments, 2022). Es un software que se utiliza cada vez más para el desarrollo de aplicaciones con el objetivo de identificar y controlar procesos industriales, su amplia variedad de bibliotecas y su interfaz sencillo e intuitivo lo hace un programa muy útil para aplicaciones como la que ocupa el presente proyecto.

Se ha desarrollado un programa que mediante la Kinect identifica el color de la pieza depositada en el punto de recogida y que además controla el proceso térmico mediante la implementación de un PID. Todo ello se complementa con la biblioteca del OPC-UA toolkit, que permite comunicar el programa con el servidor; y con bloques de funciones que completan el funcionamiento. Todo ello se enmarca dentro de un bucle temporizado<sup>6</sup> que repite cíclicamente todas las funciones insertadas en él, dado que la misión de este programa es la de detectar imágenes y controlar un proceso continuo mediante mediciones discretas, lo ideal sería reducir este periodo al máximo, no obstante, la gran cantidad de bloques de funciones implementados

<sup>6</sup> Este bucle temporizado podrá identificarse como el recuadro azul del cual se observa su parte superior junto con los parámetros de configuración de este, como el periodo de cada ciclo (dt).

retrasa la ejecución del programa entero, haciendo que deba encontrarse un compromiso entre la velocidad de cálculo y el desempeño del sistema, en el cual se ejecuten todos los bloques pero se repita la secuencia lo más rápido posible, este compromiso se ha encontrado en los 250ms.

### Detección Color

La parte de la detección del color se ha hecho utilizando las bibliotecas del Kinesthesia Toolkit, los tres primeros bloques de la imagen 54, situados fuera del bucle temporizado, son los responsables de la configuración de la cámara, esta configuración se puede ver en la interfaz, aunque el único parámetro sobre el que se actúa es el ángulo de inclinación de la cámara, se ha configurado a  $-20^\circ$ . El primer bloque dentro del bucle temporizado es el encargado de la selección del punto de interés, de él se saca un indicador para mostrar lo que ve la cámara en el interfaz y se lleva esta imagen a otro bloque el cual detecta las diferentes frecuencias del espectro electromagnético presentes en el punto de interés. De este bloque se saca otro indicador donde se observa el valor de cada una de estas frecuencias, experimentalmente se ha comprobado que los armónicos que son interesantes son el 11, el 1 y el 15, correspondiendo a los colores Azul, Rojo y Blanco. Esta información queda almacenada en un vector de tamaño  $1 \times 16$ , del cual se extrae la información deseada utilizando la función "index array" la cual permite extraer el valor deseado de un vector. Después de pasar las tres variables por un filtro que determina el rango valores en el que se está detectando dicho color, se activa la variable correspondiente, estas se muestran en el interfaz como indicadores luminosos del color correspondiente. Estas variables son las que se escriben en OPCRojo/Azul/Blanco y que son compartidas mediante el servidor OPC-UA.

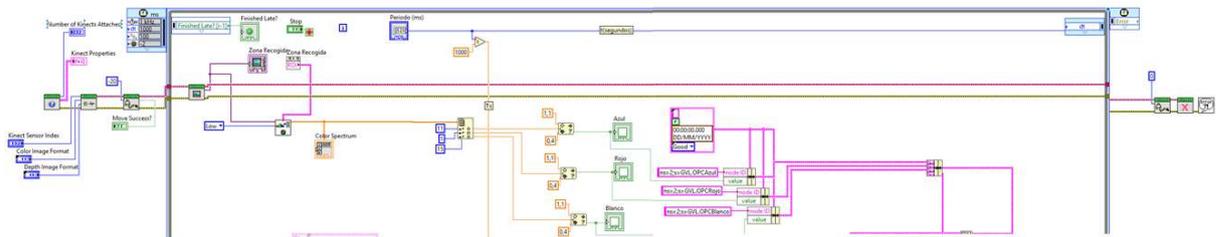


Imagen 54: LabView detección color.

### Control temperatura

Para realizar el control de la temperatura se ha implementado un control PID, esto es mediante un algoritmo que modifica la acción controlada en función de las necesidades del sistema. El proceso que se trabaja es bastante sencillo, de primer orden y sin retardo, además no se tienen en cuenta perturbaciones de forma explícita en la ley de control, lo que provocaría utilizar una estructura de control más compleja, así como la necesidad de medir dichas perturbaciones. Como únicamente se tiene en cuenta la variación de la variable controlada se utilizará un bucle realimentado simple.

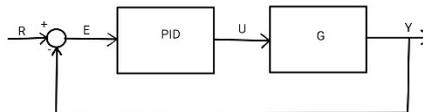


Imagen 55: Estructura de Control. Fuente: Elaboración propia.

El funcionamiento es sencillo, la salida del sistema se realimenta para restarse a la referencia, el valor deseado de esta, y dar un error, que será la variable de entrada del PID. Este controlador se encarga de proporcionar la acción de control (U) necesaria para conseguir el valor deseado, esto lo hace aplicando tres acciones: la acción P o proporcional, que actúa en función del error presente; la acción I o integral, que actúa en función del error acumulado; y la acción D o derivada, que actúa en función del error futuro, según la pendiente que este muestra. La unión de estas tres acciones es lo que se conoce como controlador PID.

$$U(t) = k_c \cdot [e(t) + \frac{1}{T_i} \cdot \int_0^t (e(t)dt) + T_d \cdot \frac{de(t)}{dt}]$$

*Ecuación 6: Algoritmo PID continuo.*

Pueden identificarse los tres parámetros principales utilizados para configurar el PID:  $K_c$ ,  $T_i$  y  $T_d$ . No obstante, esta expresión solo es válida para un dominio del tiempo continuo, y al implementarse dentro del bucle temporizado del LabView, la realimentación no será continua, sino que se repetirá cada periodo  $dT$  de tiempo. Por ello debe realizarse una discretización de la expresión anterior, siendo  $k$  un instante de tiempo y por tanto  $(k-1)$  y  $(k+1)$  los instantes inmediatamente anterior y posterior:

$$U_p(k) = K_c \cdot E(k)$$

*Ecuación 7: Discretización acción proporcional.*

$$U_i(k) = U_i(k - 1) + \frac{K_c \cdot T_s}{T_i} \cdot E(k)$$

*Ecuación 8: Discretización acción integral.*

$$U_d(k) = K_c \cdot T_d \cdot \frac{E(k) - E(k - 1)}{T_s}$$

*Ecuación 9: Discretización acción derivada.*

Puede observarse que la discretización se ha hecho a partir de la definición inicial de cada acción, la acción proporcional está directamente ligada al error presente, la acción integral se obtiene añadiendo el valor del error ponderado del último ciclo a la acción del ciclo anterior, y la acción derivada es directamente proporcional a la pendiente y por tanto a la tendencia que lleva el error. Estas tres acciones son las que se han implementado en el bloque de funciones dedicado al PID. La acción derivada es sensible al ruido dado que este puede distorsionar la tendencia entre dos instantes de tiempo, no obstante, en el caso que trabajamos no es preocupante dado que no existe acción derivada como se comprobará más adelante. Lo mismo pasa para ganancias elevadas y la acción proporcional o variaciones altas en la acción de control, que disparen el error y por tanto la acción integral.

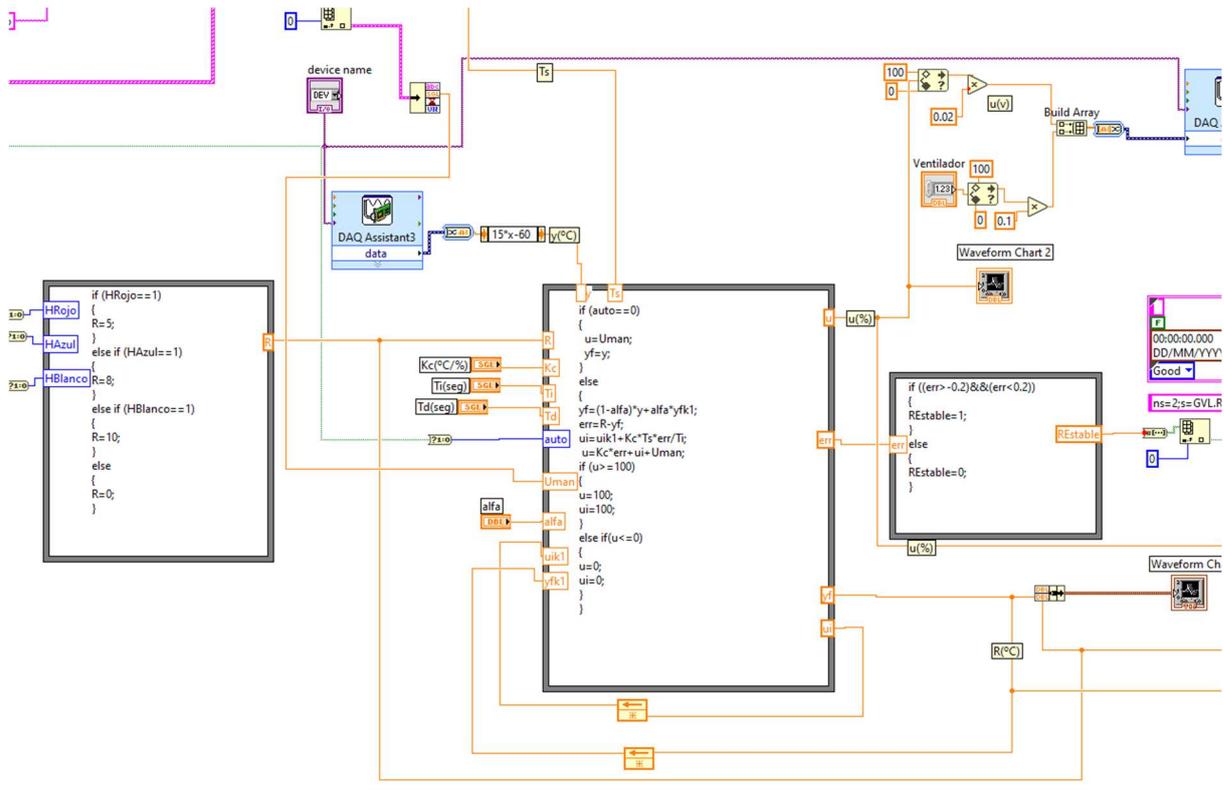


Imagen 56: Control PID mediante LabView

En primer lugar, se analizarán las variables de entrada utilizadas por el programa. El bloque izquierdo se utiliza para asignar el valor de la referencia, para ello precisa de las variables que indican que el proceso del horno está en marcha y qué tratamiento se realizará, es decir OPCHRojo/Azul/Blanco, estas variables se obtienen del servidor OPC-UA como veremos más adelante. De este bloque pues se genera el valor de la referencia R, que será entrada del bloque central, donde se ejecuta el control PID. Las entradas de este bloque serán, la temperatura de la Peltier (variable controlada, Y), obtenida a partir de la TAD; el periodo del programa en segundos (Ts), los parámetros de configuración del PID se ajustarán manualmente en el interfaz; la acción de control en caso de manejarse manualmente, la ganancia del filtro y los valores realimentados del ciclo anterior. Las salidas serán, la acción de control resultante, que se escribirá en la entrada correspondiente de la TAD junto con el valor de la acción del ventilador<sup>7</sup>, el error; que se utilizará en el bloque siguiente para determinar cuando el sistema se ha estabilizado; la temperatura resultante después del filtrado, utilizada para su monitorización en el interfaz del LabView, en el HMI y en la base de datos; y la acción integral que se realimenta.

En la imagen 57 se observa cómo se ha implementado el control PID mediante bloques de funciones programables, pueden observarse las expresiones de la acción proporcional e integral explicadas anteriormente en las líneas 9, 10 y 11 del bloque de funciones central; en las cuales también se ha tenido en cuenta que en el sistema puede haberse programado previamente un valor de U manualmente.

<sup>7</sup> Esta acción será de valor constante y se configurará antes de arrancar el programa.

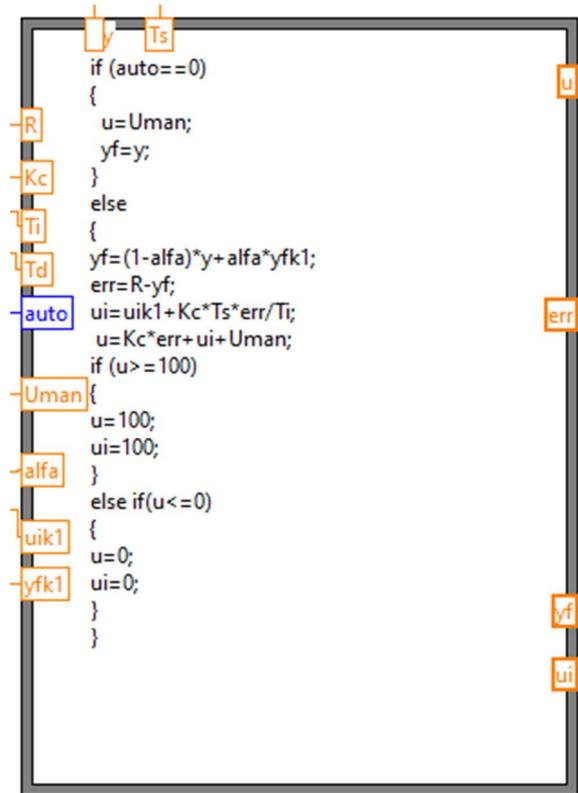


Imagen 57: Bloque de funciones - PID

Las primeras cinco líneas se encargan de ajustar el valor de la acción de control manualmente cuando la variable HornoAuto está desactivada, es decir, cuando el programa del PLC se encuentra en el modo manual, de lo contrario se ejecutará el bucle de control, y la referencia se establecerá en un valor u otro según la pieza presente en el horno, este valor se establece en el bloque de funciones de la izquierda y las variables utilizadas son OPCHRojo/Azul/Blanco, variables que indican qué pieza hay procesándose en el horno, si no existe pieza a procesar, el horno se apaga, estableciendo la referencia a 0°C.

Volviendo al bloque de funciones central, pude observarse que justo antes de la programación del PID se ha añadido una línea. Esta línea se corresponde con un filtro que pondera la salida proporcionada por la TAD y la realimentada, con el objetivo de eliminar el ruido generado a la salida, el valor de este filtro se puede ajustar desde el interfaz, con valores entre 0 y 1, y experimentalmente se ha comprobado que en este caso el valor óptimo es 0,7.

$$Y_f(k) = (1-\alpha) \cdot Y(k) + \alpha \cdot Y_f(k-1)$$

Ecuación 10: Filtro de la salida.

Otro problema que está solucionado en este bloque de funciones es el de la saturación de la acción de control, el cual se produce cuando se dispara la acción integral, siendo incapaz el algoritmo de controlarla. Para evitar este problema, se ha configurado el llamado mecanismo de antiwindup, el cual básicamente consiste en desactivar la integración cuando la acción de control sature hacia arriba ( $U > 100$ ) o hacia abajo ( $U < 100$ ), y en fijar el valor de esta acción en el

valor límite, es decir, 0 o 100. De esta manera se consigue que, en caso de saturación, el mismo algoritmo se regule y consiga estabilizarse.

El último bloque simplemente activa la variable Restable cuando la temperatura se encuentra en un rango de (-0'2, +0'2) °C respecto de la referencia, esta variable es la que informará al proceso que el horno está preparado para introducir la pieza en él.

#### *Comunicación OPC-UA y TAD*

Todos los procesos explicados anteriormente no tienen ningún sentido si el sistema no recibe información sobre el proceso real o si el resultado del procesado de los datos no va a ninguna parte. Esta es la función del servidor y de la tarjeta de adquisición de datos.

La tarjeta de adquisición de datos puede configurarse fácilmente mediante el bloque seleccionado (DAQ assistant) donde simplemente debe indicarse si estamos leyendo o escribiendo datos, de que entrada o salida proceden estos y las unidades de estas. El modo de lectura o escritura será bajo demanda, y se hará automáticamente en cada ciclo.

La implementación del servidor precisa de la biblioteca OPC-UA toolkit, de ella se saca el bloque de configuración, del cual se precisa de la entrada del protocolo de seguridad, que en este caso será ninguno; y de la entrada del endpoint del servidor, este se introducirá en el interfaz. A este bloque se le unirán en serie todos los bloques de lectura y escritura que sean necesarios en serie. Para cerrar el funcionamiento debe incluirse el bloque de desconexión junto con otro que avise en caso de producirse algún error durante el proceso.

En cuanto a la lectura de datos, se inserta la función correspondiente la cual puede configurarse para cualquier tipo de entrada, en este caso hemos creado una para las variables booleanas y otra para la entrada analógica, las variables "float" no existen en el M241, estas están declaradas como variables reales que el servidor transforma en "float" al recibirlas. El siguiente paso es el de introducir las variables a leer, esto se hace creando un vector de tipo "string", como se configuró al crear el servidor, que contenga los identificadores de las variables deseadas. Tanto los identificadores como el tipo de las variables pueden consultarse mediante un cliente rápido como puede ser el UAExpert, proporcionado por la empresa Unified Automation. La salida de estos bloques de lectura es un cluster que contiene cuatro variables: el identificador de la variable, su valor, el instante en que se ha leído (timestamp) y el estado de la variable. Si se tiene más de una variable la salida será un vector de clusters con esta información, para asegurarse del correcto funcionamiento se genera un indicador donde consultar estos valores. Para utilizar el valor de las variables primero se debe seleccionar el cluster deseado dentro del vector, y después sacar la variable deseada del cluster, el valor. Las variables leídas son, las booleanas, HornoAuto, OPCHRojo/Azul/Blanco y la variable real, UMan.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

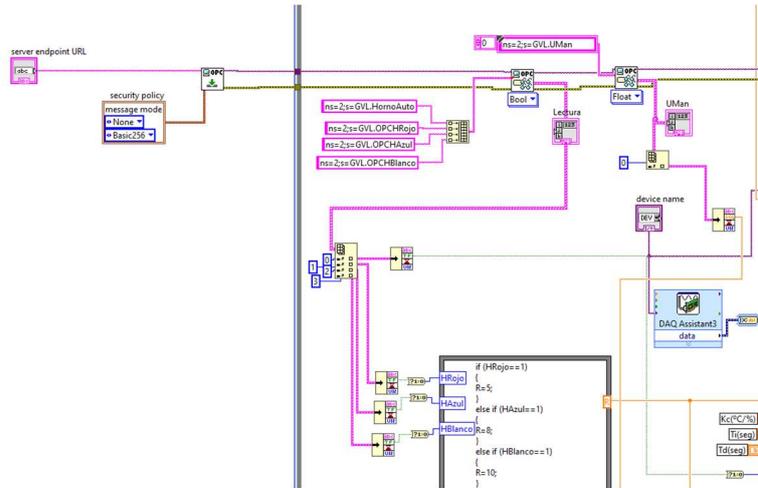


Imagen 58: Lectura y configuración inicial OPC-UA.

El proceso de escritura de datos en el servidor es similar, la mayor diferencia es que ahora la entrada de la función también tendrá forma de cluster, para configurar correctamente la entrada debe generarse desde el bloque una constante, la cual tendrá la forma del cluster necesario. Esta estructura debe copiarse y ponerse como entrada en la función de creación de cluster, para darle el formato requerido a este, una vez se tenga la función correctamente configurada se introduce el identificador de la variable a escribir y su valor, si se da la escritura simultanea de varias variables como es el caso, deben unirse en un solo vector. Las variables a escribir son las booleanas que marcan el color detectado por la Kinect y la variable Restable, mientras que las reales son la acción de control, la temperatura de la Peltier y la Referencia.

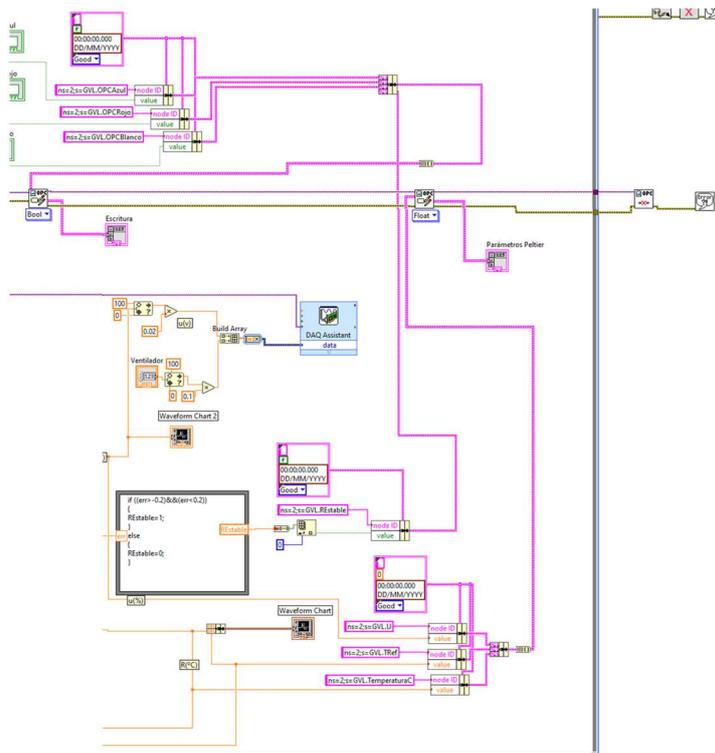


Imagen 59: Escritura datos OPC-UA

### 2.3.3 Base de Datos

Para el registro y monitorización de los datos se ha empleado la base de datos InfluxDB, el proceso de instalación y configuración inicial se ha explicado en el ANEXO III, en este apartado se explicará cómo se ha configurado el Dashboard para la visualización de los datos y la comunicación entre el servidor y la base de datos mediante telegraf.

Las variables utilizadas para la monitorización desde la base de datos han ido:

- TemperaturaHorno: Identificada como T en la base de datos, se mostrará junto con la referencia para poder visualizar las fluctuaciones de temperatura del proceso térmico.
- Referencia: Identificada como TRef en la base de datos, esta variable es necesaria para identificar el proceso que se está realizando en cada momento.
- Acción de Control: Identificada como U en la base de datos, esta variable se mostrará justo debajo de la gráfica con la temperatura para conocer el esfuerzo que realiza el sistema para conseguir la temperatura objetivo, también es útil para saber si el sistema satura u oscila demasiado en algún momento.
- Indicadores del Color: Identificados como Rojo, Azul y Blanco en la base de datos, están configurados como variable real, aunque en la práctica funcionan como una variable binaria. Su valor puede ser 0 o 100, en función de si el color correspondiente está siendo procesado o no. Se utilizan para crear unos indicadores que informen sobre la pieza que está siendo procesada.

Empezando por la comunicación, esta se configurará desde el fichero telegraf.conf, el cual se encuentra en la ruta donde esté instalada la aplicación Telegraf. Este fichero contiene muchos Plugin de entrada y salida para configurar la conexión con la plataforma desde fuentes muy variadas, el primer paso que debe darse es editar el intervalo de recolección de datos. Inicialmente este se configura para unos 10s, y dado el proceso tratado el muestreo debería hacerse lo más rápido posible, esto es cada segundo.

```
telegraf: Bloc de notas
Archivo Edición Formato Ver Ayuda
# Telegraf Configuration
#
# Telegraf is entirely plugin driven. All metrics are gathered from the
# declared inputs, and sent to the declared outputs.
#
# Plugins must be declared in here to be active.
# To deactivate a plugin, comment out the name and any variables.
#
# Use 'telegraf -config telegraf.conf -test' to see what metrics a config
# file would generate.
#
# Environment variables can be used anywhere in this config file, simply surround
# them with ${}. For strings the variable must be within quotes (ie, "${STR_VAR}"),
# for numbers and booleans they should be plain (ie, ${INT_VAR}, ${BOOL_VAR})

# Global tags can be specified here in key="value" format.
[global_tags]
# dc = "us-east-1" # will tag all metrics with dc=us-east-1
# rack = "1a"
## Environment variables can be used as tags, and throughout the config file
# user = "$USER"

# Configuration for telegraf agent
[agent]
## Default data collection interval for all inputs
interval = "1s"
## Rounds collection interval to 'interval'
## ie, if interval="10s" then always collect on :00, :10, :20, etc.
round_interval = true
```

Imagen 60: Configuración intervalo muestreo.

El siguiente paso es el de configurar las entradas y las salidas a la plataforma telegraf, esto es desde el servidor y hacia la base de datos.

Comenzando por la salida hacia la base de datos, se debe buscar dentro de “OUTPUT PLUGINS” el que está referido a la plataforma influxdb\_v2 debe andarse con ojo dado que todavía es posible configurar el fichero para conectarse a la versión anterior influxdb, de hacerse en este último la configuración, la comunicación no funcionará. Dentro de outputs.influxdb\_v2 se encuentran los campos a rellenar comentados, junto con instrucciones con cómo proceder a la configuración. Después de descomentar las líneas a utilizar, se rellenan los datos pedidos.

- URLs: Dirección donde se aloja la base de datos, esta es el host local, en el puerto 8086.
- Token: InfluxDB utiliza un mecanismo de seguridad llamado token, el cual actúa a modo de llave para que solo personal autorizado pueda acceder a ciertos datos, este token será conocido solo por aquel que lo ha configurado, y contiene información sobre si está habilitada la lectura y la escritura de ciertos campos, en este caso el token da acceso completo.
- Organización y bucket: Estos son los campos que indican donde se almacenarán los datos, estos pueden crearse desde la web de la base de datos y han sido creados durante la configuración inicial.

```
!
# # Configuration for sending metrics to InfluxDB
[[outputs.influxdb_v2]]
  urls = ["http://127.0.0.1:8086"]
  token = "1CuwYwpMzpwBB0xkHm-J_SAWQfsv83Fts65KbHjpIL2-XmUjE3hlyHHsec9YCVfK0Qrm9oSECDQ5IQ_sFF40Q=="
  organization = "UPV"
  bucket = "Monitorizacion"
# ## The URLs of the InfluxDB cluster nodes.
# ##
# ## Multiple URLs can be specified for a single cluster, only ONE of the
# ## urls will be written to each interval.
# ## ex: urls = ["https://us-west-2-1.aws.cloud2.influxdata.com"]
# urls = ["http://127.0.0.1:8086"]
#
# ## Token for authentication.
# token = ""
#
# ## Organization is the name of the organization you wish to write to; must exist.
# organization = ""
#
# ## Destination bucket to write into.
# bucket = ""
#
```

*Imagen 61: Configuración salidas telegraf.*

Por último, se configuran las entradas desde el servidor OPC-UA. Ahora hay que dirigirse al apartado “INPUT PLUGINS”, y en el plugin dedicado al servidor, el “inputs.opcu”; se configuran los parámetros correspondientes.

- Name: Se le da el nombre que se desee al conjunto de datos.
- Endpoint: En este campo debe introducirse la dirección del servidor OPC-UA, “opc.tco://(IP PLC):4840”
- Connect\_timeout y request\_timeout: Debe descomentarse y puede dejarse con los valores por defecto, 10s y 5s.

- Security policy y security mode: Estos dos parámetros dependen del servidor, al no haberse configurado ningún protocolo de seguridad su valor será “none”.
- Auth\_method: Este campo sirve para indicar el usuario con el cual se accede, en este caso se ha configurado el servidor para permitir la conexión en modo anónimo.
- Timestamp: Este parámetro se usa para definir la referencia temporal que se utilizará para hacer la conexión. La configuración “gather” utilizará la misma timestamp de los datos de entrada, es decir, la del servidor OPC-UA.
- [inputs.opcuagroup]: En este campo deberán rellenarse los datos a introducir, estos tendrán que organizarse tal y como indican las instrucciones existentes en el archivo .conf. En name, namespace y identifier\_type introduciremos el nombre del conjunto de datos que aparecerá en la base de datos, el número correspondiente al nodo y el tipo de identificador, en este caso string. En el espacio “nodes” deberán introducirse todas las variables, name será el nombre que reciban en la base de datos y identifier el nombre que tienen en el servidor. Los parámetros namespace y identifier\_type deben coger el mismo valor que en los campos anteriores, puede consultarse esta información mediante un cliente rápido del servidor.

```

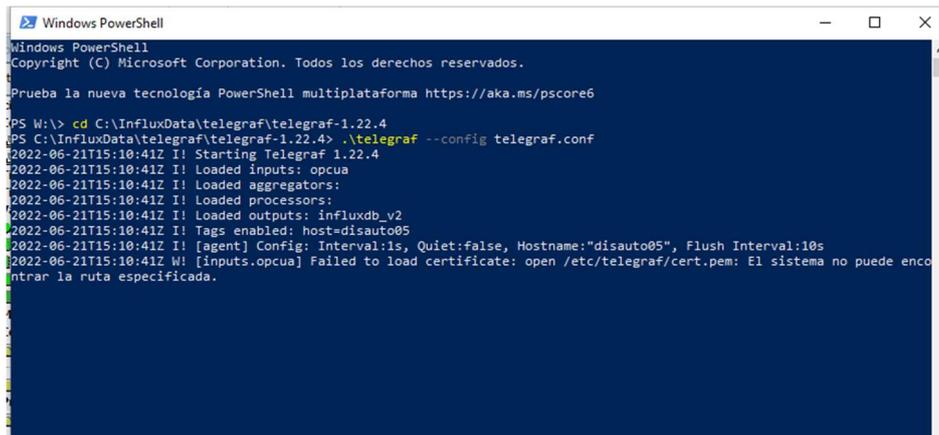
## Retrieve data from OPCUA devices
[[inputs.opcuagroup]]
  name = "Monitorizacion"
  # OPC UA Endpoint URL
  endpoint = "opc.tcp://172.16.191.105:4840/"
  ## Maximum time allowed to establish a connect to the endpoint.
  connect_timeout = "10s"
  ## Maximum time allowed for a request over the established connection.
  request_timeout = "5s"
  ## Security policy, one of "None", "Basic128Rsa15", "Basic256",
  ## "Basic256Sha256", or "auto"
  security_policy = "None"
  ## Security mode, one of "None", "Sign", "SignAndEncrypt", or "auto"
  security_mode = "None"
  auth_method = "Anonymous"
  timestamp = "gather"
#
## Node ID configuration
## name - field name to use in the output
## namespace - OPC UA namespace of the node (integer value 0 thru 3)
## identifier_type - OPC UA ID type (s=string, i=numeric, g=guid, b=opaque)
## identifier - OPC UA ID (tag as shown in opcuagroup browser)
## Example:
## {name="ProductUri", namespace="0", identifier_type="i", identifier="2262"}
# nodes = [
#   {name="", namespace="", identifier_type="", identifier=""},
#   {name="", namespace="", identifier_type="", identifier=""},
# ]
[[inputs.opcuagroup]]
#Group Metric name. Overrides the top level name. If unset, the
#top level name is used.
name = "Peltier"
namespace = "2"
identifier_type = "s"
nodes = [
  {name="Rojo", namespace="2", identifier_type="s", identifier="GVL.RDB"},
  {name="Azul", namespace="2", identifier_type="s", identifier="GVL.ADB"},
  {name="Blanco", namespace="2", identifier_type="s", identifier="GVL.BDB"},
  {name="T", namespace="2", identifier_type="s", identifier="GVL.TemperaturaC"},
  {name="TRef", namespace="2", identifier_type="s", identifier="GVL.TRef"},
  {name="U", namespace="2", identifier_type="s", identifier="GVL.U"},
]

```

Imagen 62: Configuración entradas telegraf.

Con estos parámetros la comunicación quedaría configurada y solo quedaría iniciarla. Esto se hace manualmente mediante el cliente de PowerShell, primero debe direccionarse al cliente a la carpeta donde está el archivo .conf guardado allí (cd “C:\InfluxData\telegraf\telegraf-1.22.4”),

y luego ejecutar telegraf allí (telegraf --config telegraf.conf). Haciendo esto la comunicación queda activada y puede empezar a configurarse el Dashboard y la consulta de datos. Puede que al ejecutarlo y dado que no se ha configurado ningún certificado ni sistema de seguridad, aparezca alguna alerta, esto no obstaculizará el proceso. También puede ejecutarse automáticamente configurando un servicio de Windows para tal efecto.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS W:\> cd C:\InfluxData\telegraf\telegraf-1.22.4
PS C:\InfluxData\telegraf\telegraf-1.22.4> .\telegraf --config telegraf.conf
2022-06-21T15:10:41Z I! Starting Telegraf 1.22.4
2022-06-21T15:10:41Z I! Loaded inputs: opcu
2022-06-21T15:10:41Z I! Loaded aggregators:
2022-06-21T15:10:41Z I! Loaded processors:
2022-06-21T15:10:41Z I! Loaded outputs: influxdb_v2
2022-06-21T15:10:41Z I! Tags enabled: host=disauto05
2022-06-21T15:10:41Z I! [agent] Config: Interval:1s, Quiet:false, Hostname:"disauto05", Flush Interval:10s
2022-06-21T15:10:41Z W! [inputs.opcu] Failed to load certificate: open /etc/telegraf/cert.pem: El sistema no puede encontrar la ruta especificada.
```

Imagen 63: Inicio comunicación telegraf.

El siguiente paso es abrir la base de datos y empezar a monitorizar el proceso o programar el DashBoard si es la primera vez que se ejecuta el proceso, a continuación, se explica el proceso de configuración del DashBoard utilizado.

Primero que nada, se inicia el InfluxDB, desde el buscador se entra a la dirección del host local con el puerto 8086: <http://localhost:8086>

Allí se deberá iniciar sesión ya que teóricamente ya se ha realizado el proceso de registro durante la instalación y configuración inicial vista en el ANEXO III.

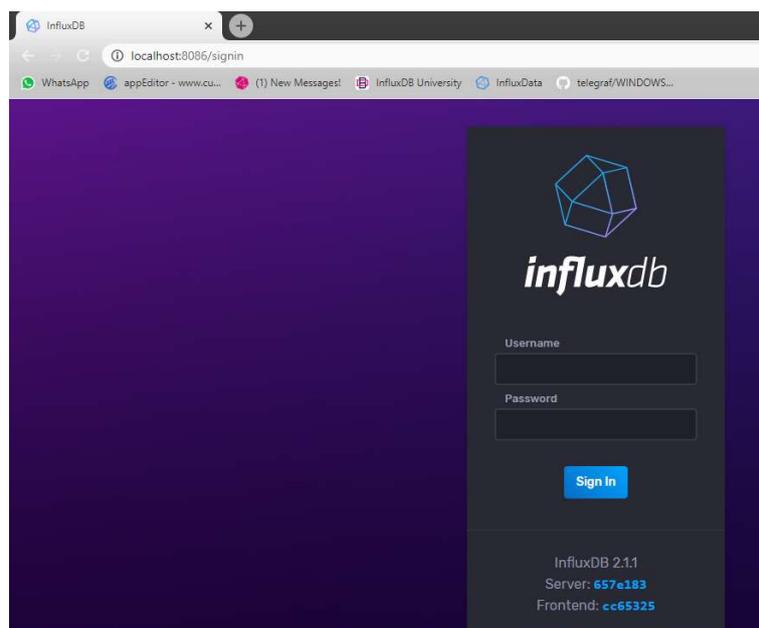


Imagen 64: Inicio de sesión InfluxDB.



Al crear la ventana nueva aparecerá la pantalla de configuración donde en la parte inferior se verán los diferentes paquetes de datos, donde se seleccionará el deseado, en este caso:

- Bucket: Monitorización.
- Measurement: Peltier.
- Data: El que corresponda, para la primera gráfica se selecciona la temperatura, T y la referencia.

En la parte superior hay una ventana donde puede elegirse el formato en que se mostrarán los datos seleccionados, en este caso, será un gráfico. También se le da aquí el nombre a la ventana.

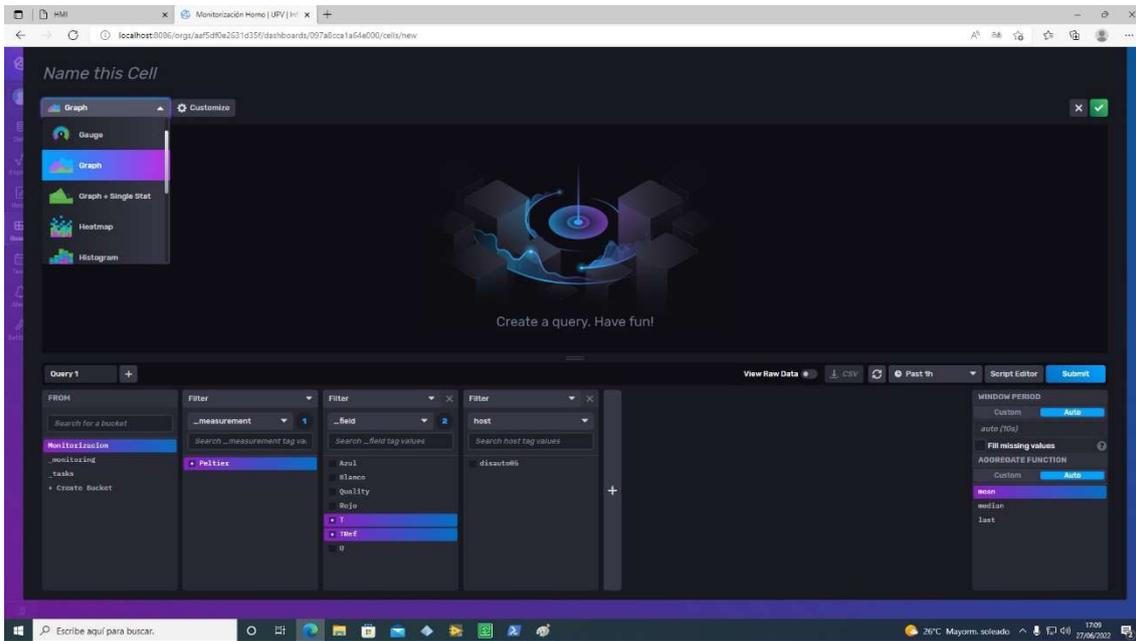


Imagen 67: InfluxBD – Selección datos.

Para darle el formato deseado se entrará en la ventana customize donde se podrá elegir el tipo de gráfico, definir la leyenda y darles formato a los ejes. Se escribirá el nombre de la variable mostrada en el eje Y, la leyenda en la parte inferior del gráfico y el tipo de gráfico será un gráfico lineal con interpolación suave. Aprovechando el ejemplo de la imagen anterior, el rango de temperaturas en que se mueve el sistema es entre 0º y 10ºC con lo que se mostrará un rango entre -0,5º y 12ºC. Pulsando “submit” se mostrarán los datos por pantalla facilitando la tarea de configuración. Cuando se tengan todos los parámetros configurados se aprieta el botón de validación y se coloca la ventana creada donde se desee y al tamaño conveniente.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

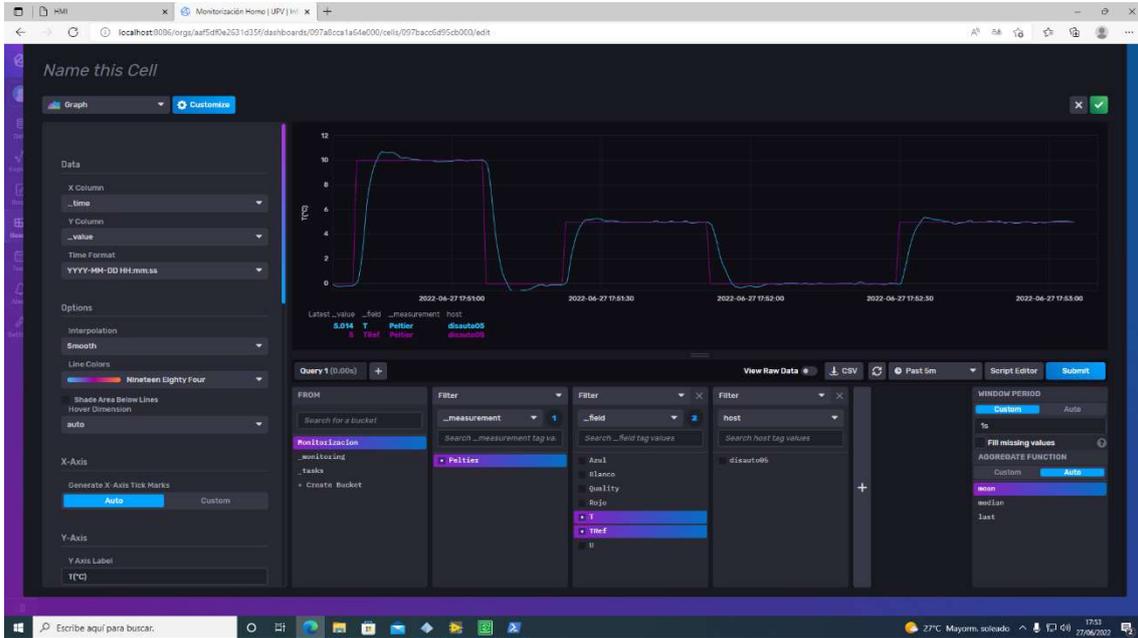


Imagen 68: InfluxDB - customize Tdgraph1.

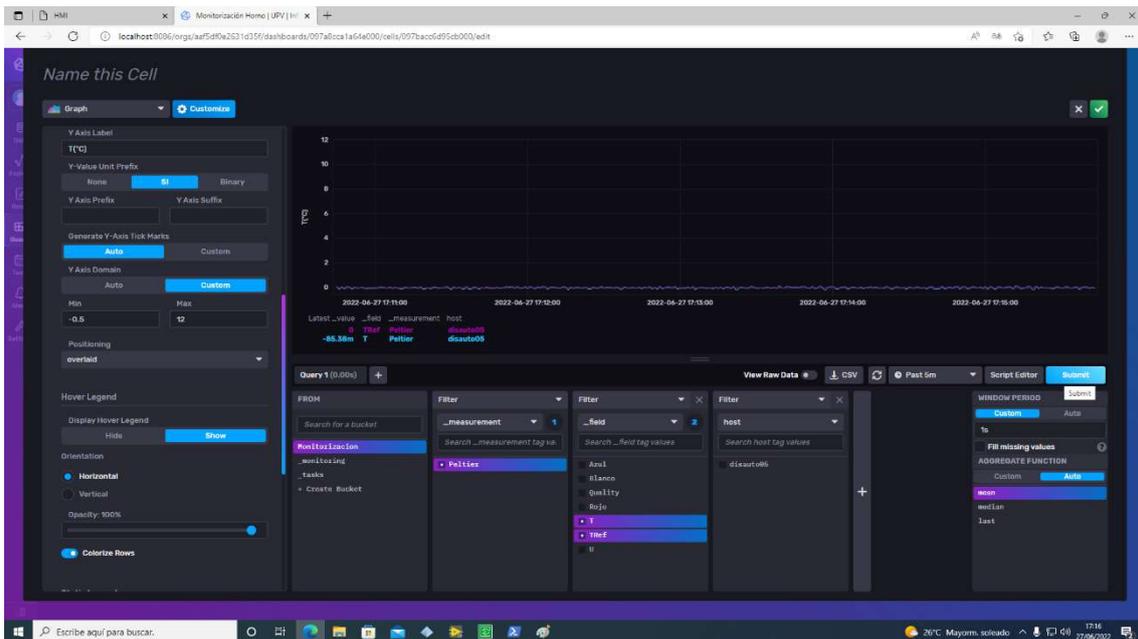


Imagen 69: InfluxDB - customize Tdgraph2.

El proceso de configuración de la acción de control es similar, dado que también se utilizará una gráfica lineal con interpolación suave, como la variable se expresa en porcentaje el rango esta vez será de 0 a 100.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

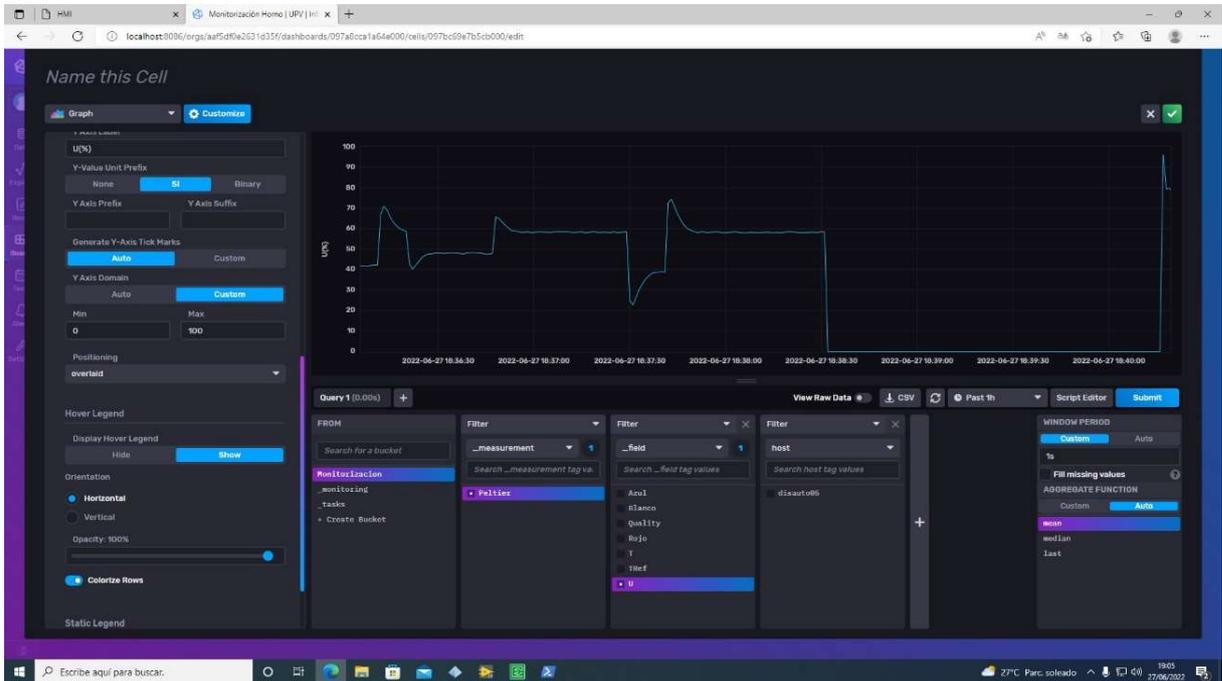


Imagen 70: InfluxDB – customize Acción de control.

Por último, para el indicador dedicado al color, se utilizará un medidor circular. Este tomará su valor máximo cuando la pieza del color correspondiente esté siendo procesada y cuando no se mantendrá a 0. El tipo de indicador a mostrar es el “Gauge” y se configura su valor mínimo a 0 y su valor máximo a 100.

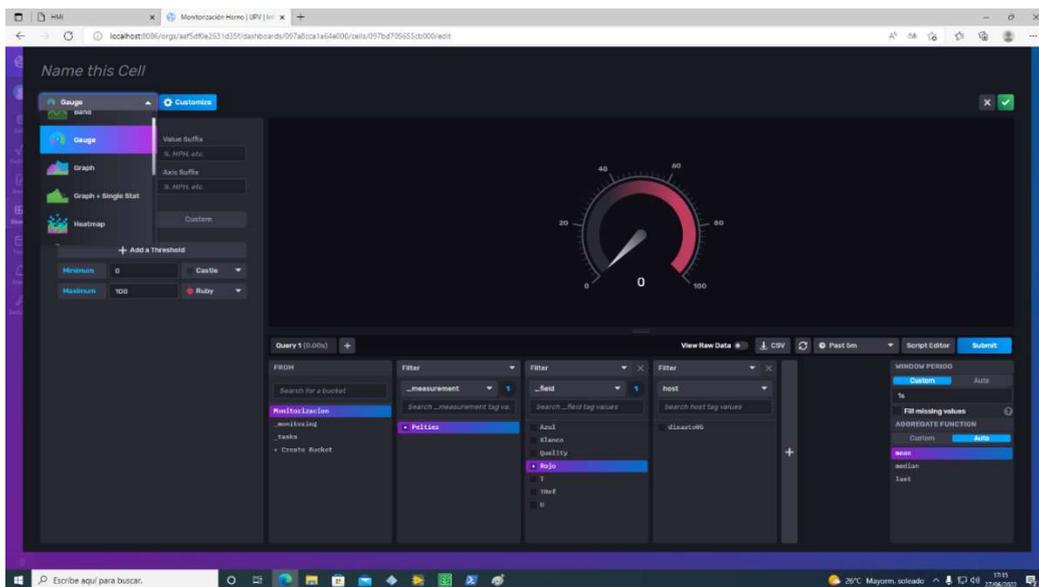


Imagen 71: InfluxDB - customize indicador Rojo.

Esta es la configuración adoptada para el Dashboard del proyecto, aunque InfluxDB ofrece multitud de opciones que deben ser tomadas en cuenta.

## CAPÍTULO 3: Manual del usuario

En este capítulo se va a describir la puesta en marcha y manipulación del sistema para que cualquier persona que conozca el proceso sea capaz de gobernarlo. Los parámetros introducidos en la inicialización pueden variar si se desea cambiar el PID debido a un cambio en el proceso o de sus condiciones, si se cambia la IP de los PLC utilizados o en definitiva si se produce algún cambio de circunstancia que obligue a alterarlos. Estos cambios serán ejecutados previsiblemente por un ingeniero de control o de procesos, y será el quién indique los nuevos parámetros a configurar. Para el proceso y sistema considerado, se dan por válidos los datos proporcionados en cuanto a direcciones y parámetros del PID, estos se justifican en el ANEXO II.

### 3.1 Inicialización del sistema

Partiendo de la hipótesis de que los elementos están correctamente colocados según la distribución en planta de la imagen 25 y que los PLC, la célula Peltier y la Kinect están debidamente conectados; se procede a explicar la puesta en marcha del sistema completo.

Para empezar y en caso de que no se haya hecho previamente, se inician los programas de los respectivos PLCs, se transfieren y se ponen en modo Run. Mientras que no se conecte MARCHA el proceso no arrancará, sin embargo, es necesario empezar iniciando los dos programas para que el M241 cree el servidor OPC.



Imagen 72: Inicio Unity Pro M.



Imagen 73: Inicio SoMachine.

Una vez iniciados los PLCs, se ejecuta el programa de LabView y se configuran los parámetros presentes en el interfaz.

- PLC:  $K_c = -2,8165$ ;  $T_i = 2,1906$ ;  $T_d = 0$ .
- Alfa = 0,7.
- DeviceName = Dev1.
- Ventilador = 30%.
- Server Endpoint URL = `opc.tcp://172.16.191.105:4840`
- Periodo = 250ms.

Con los parámetros configurados puede presionarse la flecha que inicia el programa. Tardará unos segundos en responder, si todo ha funcionado correctamente la Kinect se inclinará hacia debajo y empezará a mostrar imágenes en directo. Si la cámara no detecta o encuadra correctamente las piezas, se reubicará donde la imagen sea la deseada. Desde este interfaz pueden observarse los parámetros de configuración de la kinect, la imagen que esta ve, el color

que detecta el programa junto con el valor de cada una de las 16 frecuencias y una gráfica con los datos de la temperatura y la referencia.

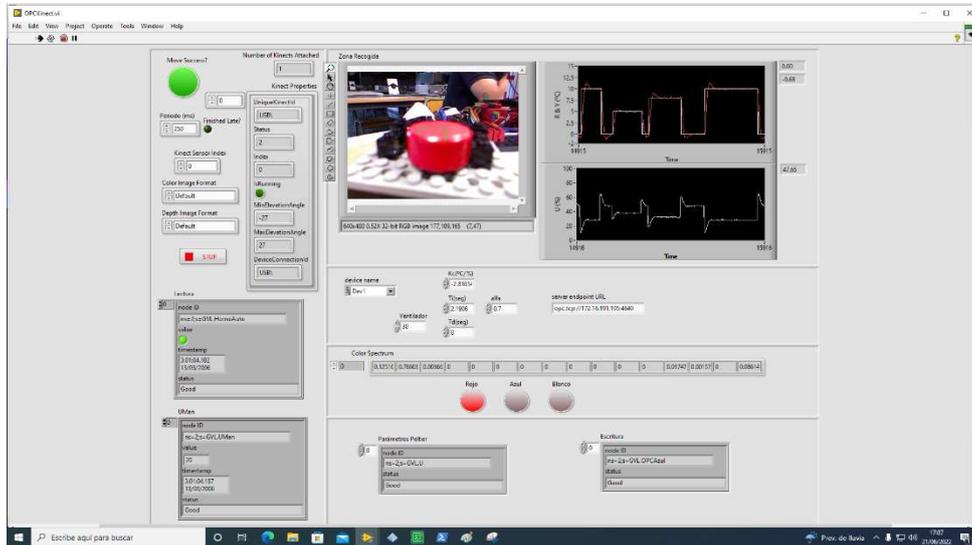


Imagen 74: Captura Interfaz LabView

Por último, y para almacenar la sesión de trabajo, se inicia la conexión con la base de datos tal y como se ha explicado en el apartado 3.3 y si se quiere monitorizar el proceso en tiempo real se accede a la web <http://localhost:8086/>, se inicia sesión con el usuario y contraseña definidos y se abre el Dashboard.

### 3.2 Manipulación del sistema

Partiendo de la situación a la que se ha llegado en el apartado anterior, ya se tienen todos los subsistemas en marcha, a no ser que quiera comprobarse algún aspecto técnico del programa, el interfaz del LabView y el Unity Pro M podrán minimizarse y no serán necesarios para la manipulación del sistema.

Se ha creado un visualizador Web desde el M241 para poder acceder al HMI desde cualquier dispositivo, por lo que el manejo podrá hacerse tanto desde el PC conectado al PLC como desde algún dispositivo portátil del que disponga el operario. La dirección web a la que se debe acceder es "(IP PLC):8080/webvisu.htm". Sea cual sea la elección, para iniciar el proceso se seleccionará el modo de funcionamiento deseado, automático o manual, y se pulsará el interruptor MARCHA.

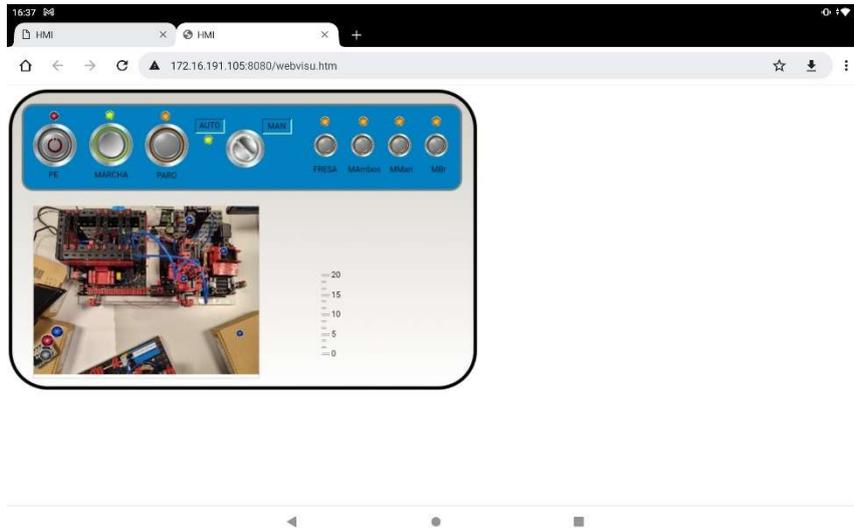


Imagen 75: HMI inicial modo auto.

Una vez iniciado el sistema en modo automático, este esperará a que el usuario coloque una pieza en el punto de recogida, donde será reconocido su color, iluminando el indicador correspondiente e iniciando la secuencia. El brazo mecánico recogerá esta pieza, momento en que se verá apagarse el indicador que informaba de la existencia de una pieza en el punto de recogida, y se iluminará y aparecerá el de pieza en la bandeja del horno mientras este se calienta. Una vez el horno se ha equilibrado a la temperatura de referencia, la pieza entra y aparecen y se iluminan los indicadores pertinentes. La señalización mediante los indicadores irá marcando la posición de la pieza durante toda la secuencia hasta que esta sea depositada en la zona de almacenamiento.

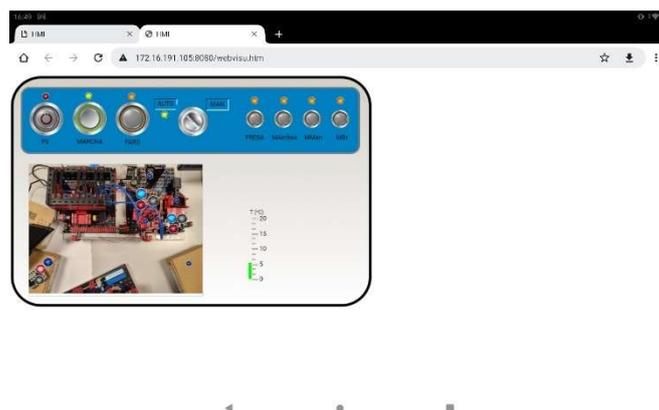


Imagen 76: HMI durante funcionamiento automático.

El usuario podrá detener el proceso, ejecutar el estado de alarma o cambiar el modo de funcionamiento en cualquier momento desde la barra de comandos fija superior.



Imagen 77: HMI - Barra de comandos fija.

Si desde el modo automático se desea activar el modo manual o parar el sistema debe cambiarse la posición del selector o pulsar el interruptor de PARO y el proceso se detendrá en cuanto acabe el proceso activo. Si alguna pieza ha quedado en la estación de mecanizado o en la bandeja del horno esta debe retirarse antes de reiniciar el proceso en automático.



Imagen 78: HMI - modo manual.

El modo de mantenimiento y cambio de herramienta se activa mediante el interruptor FRESA, y esperará a que el manipulador este libre y el brazo manipulador este quieto, en caso de estar dejando una pieza nueva en el horno. Como se ha dicho este modo está pensado para realizar el mantenimiento de la estación o realizar el cambio de herramientas, por lo que se retira el manipulador por seguridad y se mantiene el proceso activo en pausa, reanudándose en el punto en qué quedo una vez se desactiva el interruptor.

Los interruptores MAmbos, MMan y MBr deben utilizarse por personal cualificado y con el proceso real delante, no puede hacerse desde una oficina, por lo que se elimina la visualización del proceso cuando estos se activan. Su función es la de pasar los mandos del manipulador, del brazo o de ambos a modo manual por las necesidades del momento. En este modo semiautomático el proceso continúa bajo la supervisión del operario, aunque no se podrá volver al modo automático sin primero realizar una parada del sistema y la desactivación de la señal que activó el modo semiautomático.

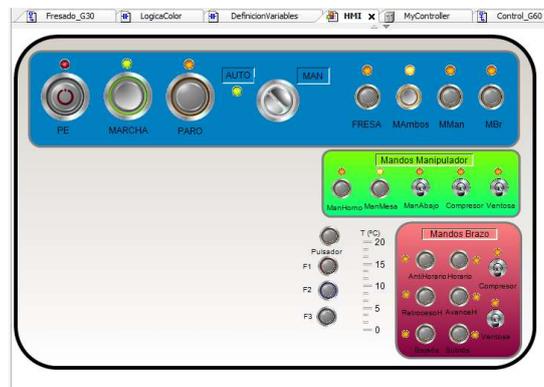


Imagen 79: HMI - modo semiautomático.

Con la excepción del modo manual, en el que se supone que el operario está supervisando y controlando directamente el funcionamiento del proceso, se accede al modo de emergencia desde cualquier otro estado simplemente pulsando el pulsador con enclavamiento PE. En este modo el fondo del HMI se volverá rojo en señal de alerta y permitirá manejar manualmente el proceso con la excepción de aquellas salidas que estén forzadas, como es el caso del horno o de las ventosas de los manipuladores.

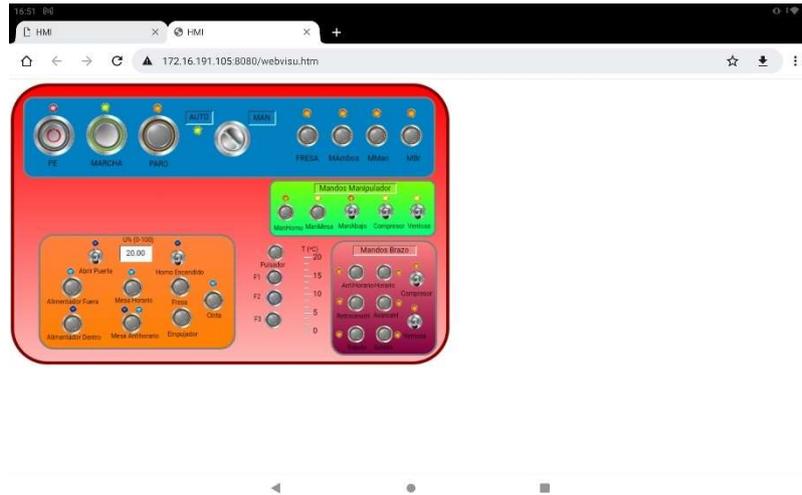


Imagen 80: HMI - modo de emergencia.

Para salir del modo de emergencia, después de solucionar los problemas existentes, se deben desconectar tanto el pulsador con enclavamiento PE, como el interruptor de MARCHA.

Por último, si en cualquier momento se quiere detener definitivamente el proceso debe desactivarse el interruptor de MARCHA, si el proceso está en modo manual, este se apagará inmediatamente; si está en modo de emergencia seguirá en el hasta que se desconecte PE y si está en modo automático o semiautomático, se esperará a que terminen los procesos activos para luego desconectarse, tal y como pasaría activando PARO. La principal diferencia entre PARO y desconectar MARCHA, es que PARO lleva al sistema a un estado de espera mientras que la desconexión de MARCHA conlleva la desconexión completa del sistema.

Una vez desconectado el sistema puede procederse al apagado del resto de sistemas, prestando especialmente atención a la desconexión del LabView, esta debe hacerse desde el botón STOP presente en su interfaz, de lo contrario, la cámara Kinect no volverá a su posición de reposo.

En cualquier momento, durante el procesado o posteriormente, puede consultarse la base de datos donde veremos una gráfica que muestra la evolución de la temperatura junto con la referencia y justo debajo otra con la evolución de la acción de control. Justo al lado de estas se observan tres indicadores con forma de potenciómetro y con el color de la pieza a la que representan, los cuales señalarán su valor máximo cuando la pieza de su color este siendo

procesada. Gracias a esto, y junto a los indicadores del HMI, puede comprobarse en tiempo real si el sistema funciona como debería, ajustando la referencia a la pieza correspondiente.

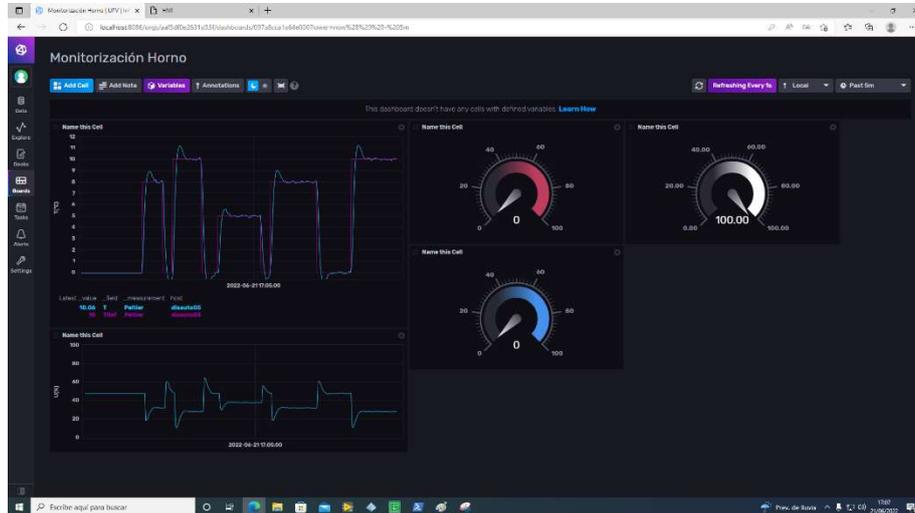


Imagen 81: InfluxDB - Interfaz Base de datos.

Arriba a la derecha pueden configurarse parámetros relacionados con los instantes mostrados y el muestreo en tiempo real. Este último puede activarse o desactivarse, para este proceso lo recomendable es activarlo con una frecuencia de muestreo de 1s, el tiempo mínimo. Para observar con detalle el proceso también es recomendable observar los últimos 5 minutos, aunque si lo que se desea es consultar el histórico, desde la configuración del tiempo mostrado, puede seleccionarse la opción de personalizar el rango a visualizar, y seleccionar el o los días que interesen, así como también las horas a mostrar.

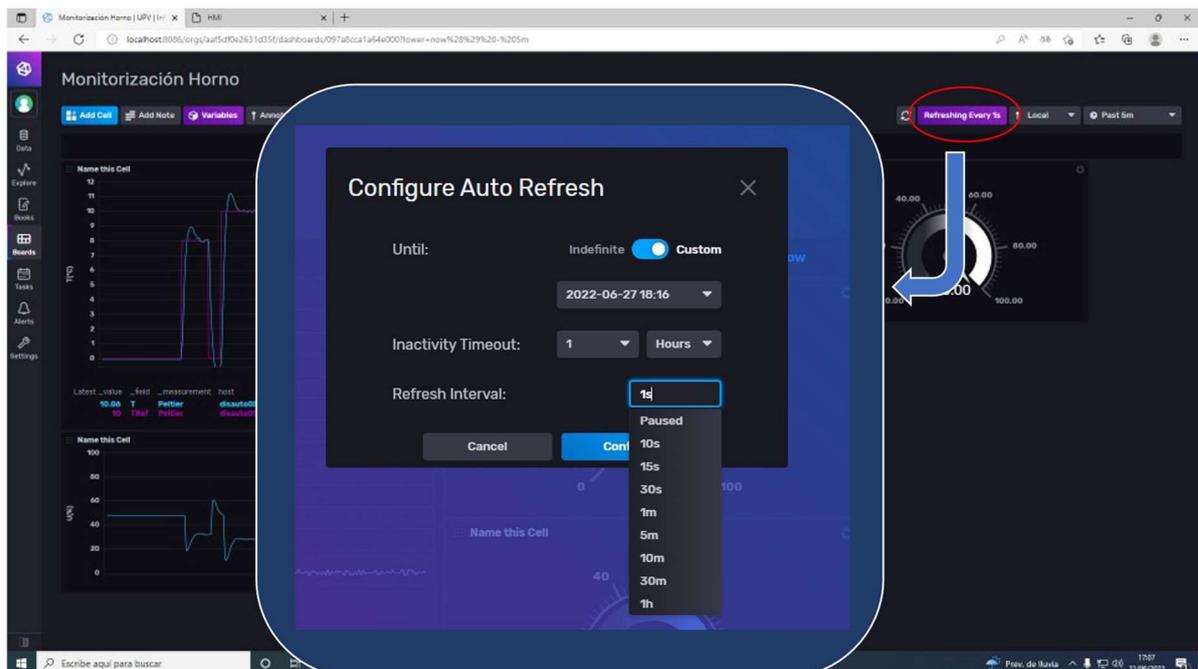


Imagen 82: InfluxDB – Monitorización en tiempo real.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

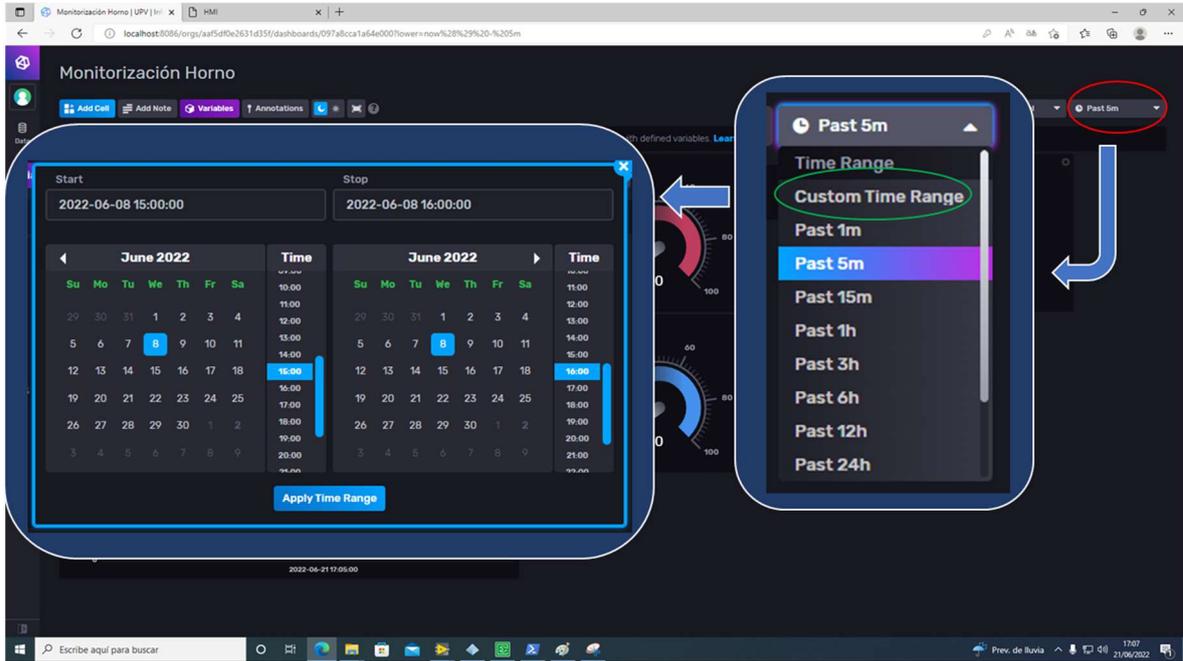


Imagen 83: InfluxDB - Visualizar rango personalizado.

## CAPÍTULO 4: Conclusión

El objetivo del trabajo era diseñar e implementar un sistema que automatice, controle y digitalice una estación donde se realiza un tratamiento térmico y el postprocesado de las piezas. Para ello se ha supuesto el que debería ser el funcionamiento normal de la estación y a partir de él y con la ayuda de elementos como el programa con la identificación de color y control PID, el servidor OPC-UA y la base de datos, InfluxDB; se ha desarrollado una estructura de procesos que ha permitido implementar el sistema con bastante robustez.

Las constantes pruebas realizadas en el laboratorio han dado como fruto la perfecta coordinación entre cada uno de los subsistemas para realizar una tarea perfectamente escalable a una industria real que maneja parámetros similares.

Merece la pena profundizar en elementos como el servidor OPC, que con una configuración mínima permite la conexión directa entre sistemas de características muy diferentes; o la base de datos InfluxDB, la cual permite el almacenamiento de datos basados en series temporales de manera muy optimizada y permitiendo ser almacenados tanto en un servidor local como en la nube, tiene la posibilidad además de crear interfaces de consulta muy visuales. Otra de las características que hace a InfluxDB una plataforma muy potente es la posibilidad que ofrece de programar el borrado de los datos automáticamente cada cierto periodo de tiempo, con el objetivo de no sobresaturar la memoria con datos obsoletos y optimizar aún más su funcionamiento.

Desde el punto de vista práctico del proyecto, las mayores dificultades a la hora de hacer funcionar el sistema correctamente han sido:

- La correcta calibración de la posición de los equipos. El funcionamiento del generador de pulsos es muy sensible y con la mínima perturbación el funcionamiento real puede ser muy diferente del deseado. La solución es tener paciencia y realizar los movimientos de una sola pasada.
- Conseguir que la aplicación LabView asegure el funcionamiento adecuado del controlador PID. La carga de funciones en el programa es alta y se requiere la máxima velocidad posible con lo que para asegurar un buen funcionamiento se ha tenido que optimizar el programa al máximo y aumentar el periodo del bucle temporizado. Esto ha pasado por cargar en el mismo programa la detección del color y el control PID, con el peso añadido y necesario de las comunicaciones con el servidor OPC-UA. En un futuro, de replicarse el proyecto, la elección del sistema para implementar el PID sería desde el PLC, ahorrando al programa de LabView no solo el control PID sino también la comunicación de las variables de este con el servidor OPC-UA, aumentando la eficiencia del sistema en su conjunto .
- La integración de la base de datos en el proyecto. Aunque una vez se conocen todos los detalles de la plataforma el desarrollo de esta es bastante sencillo, aprender el funcionamiento de esta, partiendo de la base de conocimiento sobre programación en GO y bases de datos que se tenía al principio del proyecto; ha sido todo un reto que ha dado sus resultados.

La combinación de subsistemas utilizada hace que las posibilidades de ampliación del proyecto sean bastante interesantes, algunas de las propuestas son:

- Profundizar en las posibilidades que ofrece InfluxDB. Lo mostrado en el proyecto es solo una muestra de todo lo que puede conseguirse con esta plataforma, puede ampliarse el proyecto configurando alarmas si se detecta un pico de temperatura o incluyendo el tratamiento de los datos desde la misma plataforma.
- Ampliación de los modos de funcionamiento existentes.
- Configuración de la señal Em, aunque se ha implementado en el programa como señal de activación del modo de emergencia, no se ha definido. Esto es porque no se han considerado situaciones de peligro o de incompatibilidades en el funcionamiento que no pudieran solucionarse rediseñando las transiciones y haciendo más robusto el programa. Sin embargo, en un sistema real con sensores que informen sobre el estado de los actuadores y de otros parámetros, sería obligatorio tener una variable como esta vinculada a estos posibles defectos.

## CAPÍTULO 5: Bibliografía

economia 3. (s.f.). *economia3.com*. Obtenido de <https://economia3.com/2016/10/13/86041-industria-3-0/#:~:text=A%20mediados%20del%20siglo%20pasado,lleva%20a%20la%20producci%C3%B3n%20automatizada>.

Elektronika, Larraioz. (s.f.). *larraioz*. Obtenido de <https://larraioz.com/codesys>

FayerWayer. (2010). *FayweWayer.com*. Obtenido de <https://www.fayerwayer.com/2010/06/especificaciones-tecnicas-de-kinect/>

Fischertechnik. (2022). <https://www.fischertechnik.de>. Obtenido de [https://www.fischertechnik.de/-/media/fischertechnik/fite/service/elearning/simulieren/doku-9v/es/536627-multiestacin\\_de\\_procesamiento\\_con\\_horno\\_de\\_coccin\\_9v.ashx](https://www.fischertechnik.de/-/media/fischertechnik/fite/service/elearning/simulieren/doku-9v/es/536627-multiestacin_de_procesamiento_con_horno_de_coccin_9v.ashx)

Fischertechnik. (2022). <https://www.prodel.es>. Obtenido de <https://www.prodel.es/producto/fischertechnik/manipulador-aspiracion-vacio/>

Fischertechnik. (2022). <https://www.prodel.es/producto/fischertechnik/manipulador-aspiracion-vacio/>.

MathWorks. (2022). *es.mathworks.com*. Obtenido de <https://es.mathworks.com/help/ident/>

National Instruments. (2022). *NI*. Obtenido de <https://www.ni.com/es-es/shop/labview.html>

- Ponsa Asensio, P., & Vilanova Arbós, R. (2005). Automatización de procesos . *Automatización de procesos* . Barcelona, Cataluña, España: Edicions UPC.
- Salcedo, J. V. (s.f.). Manuales de Prototipos. *Manual Nº 8: Célula Peltier*.
- Schneider. (2009). [http://origin-faq.pro-face.com/resources/sites/PROFACE/content/live/FAQS/28000/FA28426/es\\_ES/Unity%20v41%20-%20Manual%20de%20referencia.pdf](http://origin-faq.pro-face.com/resources/sites/PROFACE/content/live/FAQS/28000/FA28426/es_ES/Unity%20v41%20-%20Manual%20de%20referencia.pdf).
- Schneider. (2018). <https://www.se.com/es/es/download/document/EIO0000000071/>.
- Simarro, R. (2022). Procesos FischerTechnik.
- Simarro, R. (2022). Trabajo de Automatización y control.
- SystemHaus . (2021). *systemhaus.com*. Obtenido de <https://www.systemhaus.com.br/es/blog/categoria/15/detalhe/77/-C2-BFcual-es-la-diferencia-entre-industria-3-0-e-industria-4-0-3F->
- SystemHaus. (s.f.). *SystemHaus.com*. Obtenido de <https://www.systemhaus.com.br/es/blog/categoria/15/detalhe/77/-C2-BFcual-es-la-diferencia-entre-industria-3-0-e-industria-4-0-3F->
- Toledo, F. J. (2009). Obtenido de <http://fjrivass.com>



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERIA  
INDUSTRIAL VALÈNCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**PROYECTO DE AUTOMATIZACIÓN,  
MONITORIZACIÓN Y DIGITALIZACIÓN DE UN  
PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y  
COMUNICACIÓN OPC-UA**

**ANEXO I: DISEÑO DEL AUTOMATISMO**

AUTOR: CARLOS FERRER CAMPOS

TUTOR: JOSÉ LUÍS PITARCH PÉREZ

COTUTOR: RAÚL SIMARRO FERNÁNDEZ

**Curso Académico: 2021-2022**

## 1 Introducción

En este anexo se detalla el proceso seguido para diseñar el automatismo. Para ello se han seguido dos estándares cuya finalidad es precisamente la de unificar esta primera etapa de diseño. Estos estándares son la metodología GRAFCET y la guía GEMMA.

## 2 Diseño

En primer lugar, se ha diseñado un GRAFCET de control para cada uno de los procesos basándose en el esquema creado a partir de la guía GEMMA ya descrita en el apartado 2.2.2. La idea es crear un diagrama en el cual cada estado se corresponda con un modo de funcionamiento de los definidos mediante la guía y las transiciones entre ellas sean las condiciones reflejadas en la guía GEMMA, descrita en la imagen 28.

El diagrama de control aplicado en la estación multiproceso es el que gobernará todo el sistema, el del brazo será uno paralelo que funcionará según el principal. Los distintos modos de funcionamiento que se materializan en etapas son:

- El modo de desconexión, el sistema está desconectado y no se realizará ninguna acción hasta que se active la MARCHA, se corresponde con la etapa 60.
- El modo de puesta y parada en estado inicial, etapa 61.
- El modo de parada pedida a final de ciclo, etapa 65 si es para la desconexión total o 62\_1 si es para salir del modo automático o semiautomático.
- El modo de parada de emergencia, correspondiendo con la etapa 66.
- El modo de funcionamiento manual, etapa 64.
- El modo de funcionamiento automático, etapa 62.
- Los modos de funcionamiento semiautomático, considerados modos de producción con defectos, en las etapas 63, 63\_1 y 63\_2.

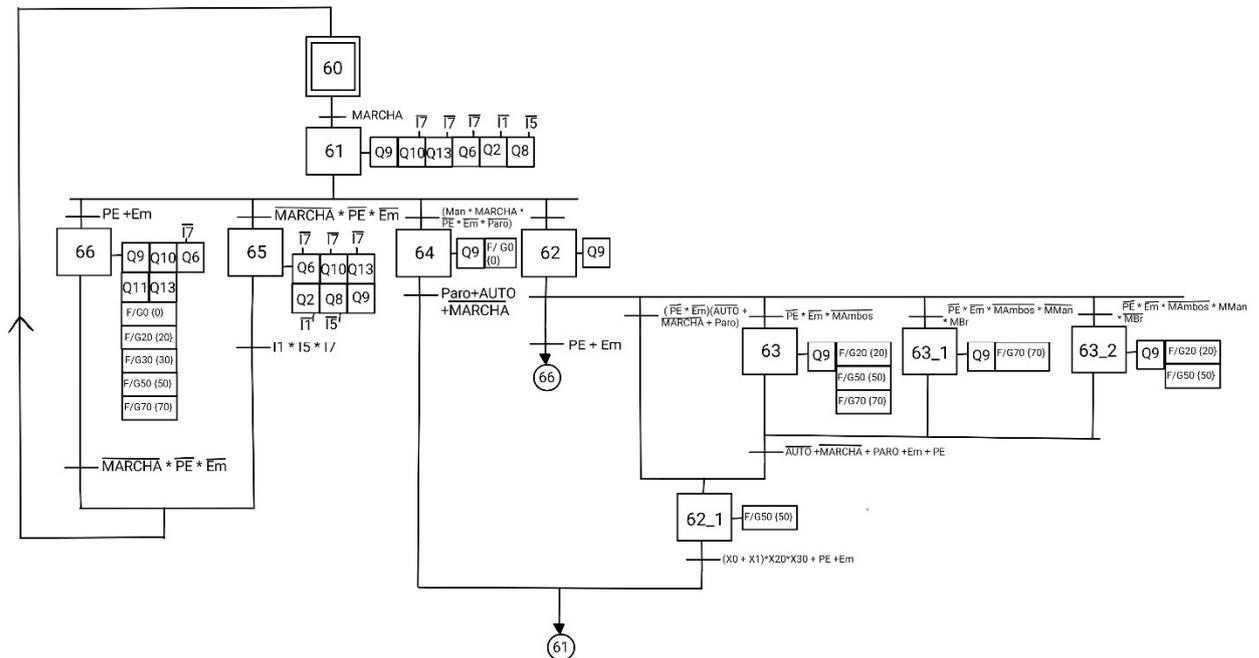


Imagen 84: GRAFCET Control\_G60.

En la etapa 61 y 65 se activan las acciones necesarias para llevar al sistema a la posición inicial, la descrita en el apartado 2.2.2. En la 66, se han fuerzan el resto de los procesos a su etapa inicial, se activa la succión de los manipuladores para evitar accidentes, y se apaga y abre el horno. Durante el proceso manual se ha decidido forzar el primer Grafcet para evitar incompatibilidades. La etapa 62 sirve para habilitar el inicio de algunos de los procesos como se verá más tarde, es la condición principal para que el automatismo de comienzo. Los modos semiautomáticos fuerzan aquellos grafkets relacionados con su funcionamiento, esto es el G70, donde se da la transferencia del color para el brazo; y G20 y G50 para el manipulador, ya que son los grafkets encargados de su control. La etapa 62\_1 garantiza que todos los subprocesos activos han finalizado antes de dejar paso al sistema a seguir con su funcionamiento, se fuerza G50 (grafcet con la secuencia de mantenimiento) porque se entiende que al parar el sistema o salir del modo automático este proceso también debería finalizar, aunque esté activo.

Se han creado las variables P6x para guardar la información de la etapa activa y compartirla con el PLC del brazo, para gobernar su diagrama de control.

Las variables P60, P61, P62 y P66 se activan cuando lo hacen las etapas X60, X61, X62 y X66 respectivamente y por tanto se utilizan como condición para llegar a estas mismas etapas en el Grafcet del brazo, se ha mantenido la misma numeración para las etapas para que sea fácil de entender el paralelismo entre ambos grafkets. De los modos semiautomáticos al PLC del brazo realmente solo le interesa saber cuándo este pasa a estar en modo manual por tanto la variable P63 se activará con X63 y X63\_1 y llevará a la secuencia desde el modo automático hasta el manual, este mismo modo manual podrá ser accedido cuando este se active en el PLC de la estación, esto es con la etapa X64 y X62\_1.

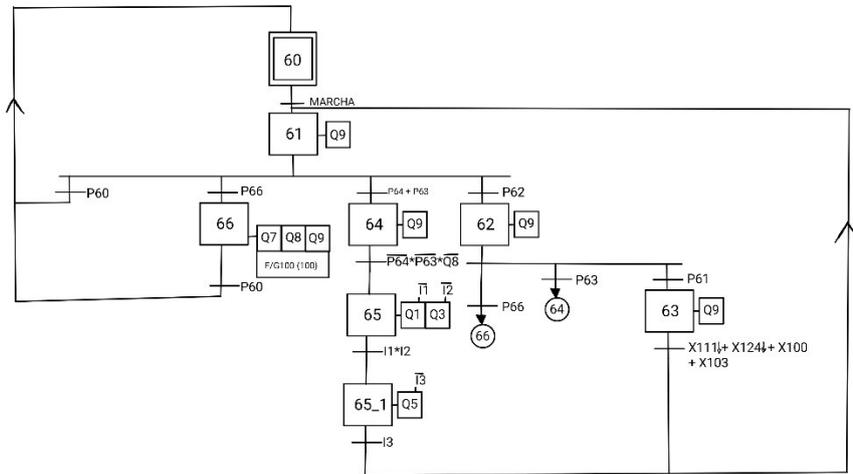


Imagen 85: GRAFCET Control\_G60'.

De esta forma se ha conseguido un grafcet de control paralelo para el brazo y con los mismos modos:

- El modo de desconexión se corresponde con la etapa 60.
- En este caso la etapa 61 es simplemente una etapa de espera, la puesta en el estado inicial se incluye en el programa del modo automático.
- El modo de parada pedida a final de ciclo, con la etapa 63, donde se esperará que se termine el proceso activo.
- El modo de parada de emergencia, correspondiendo con la etapa 66.
- El modo de funcionamiento manual que en este caso también sirve para el modo semiautomático, etapa 64.
- El modo de funcionamiento automático, etapa 62.
- Una puesta en el estado inicial a la salida del modo manual, en las etapas 65 y 65\_1.

Durante el funcionamiento automático los diferentes subsistemas tienen un funcionamiento secuencial por lo que se han diseñado también mediante el uso del GRAFCET. En estos se ha incluido la lógica utilizada para compartir el color de la pieza tratada de un proceso a otro.

Una vez dentro del funcionamiento automático, el primer proceso que se activa es el del horno. Como en todos los diagramas, la etapa inicial sirve para reiniciar las variables que identifican el color de la pieza y para esperar a que se cumplan las condiciones necesarias para empezar el proceso. El inicio del proceso del horno se activa mientras el modo automático está activo y el mantenimiento no esté en marcha, en la etapa 1 se esperará a que la Kinect detecte una pieza, y cuando esto ocurra se activará la variable correspondiente a su color (HRojo/Azul/Blanco), hasta que esto pase todavía existe la posibilidad de pasar al modo manual. Una vez detectada la pieza, el color es compartido con el programa del control de temperatura el cual enciende el horno y lo ajusta a la temperatura indicada según el proceso, paralelamente el brazo recoge la

pieza y la deja en la bandeja, etapa 3. Una vez la temperatura se ha estabilizado, el horno introduce la pieza y realiza el tratamiento el tiempo indicado, en este caso se ha asignado el mismo tiempo a las tres piezas, pero el automatismo se ha diseñado para que este pueda variar en función del proceso, ya que para cada color se tiene una etapa de espera distinta a la cual se le puede configurar el tiempo de espera mediante la transición de salida, etapas 7, 8 y 9. Una vez hecho esto la pieza se extrae y se espera a que el manipulador la haya recogido, etapa 12. En esta última etapa también se produce la transmisión de la información sobre el color de la pieza. Una vez la pieza haya sido recogida, el proceso queda libre y puede recibirse otra pieza para ser procesada. Otra opción para liberarlo es activar la señal Pulsador, esto se utiliza durante el modo semiautomático dado que aquí la secuencia del manipulador se paraliza.

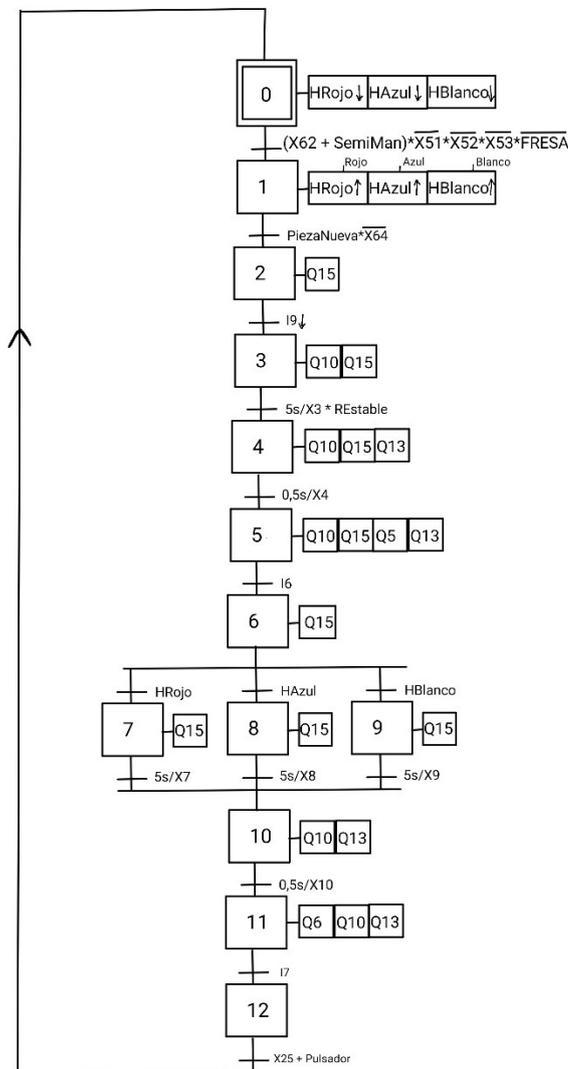


Imagen 86: GRAFCET Horno\_G0.

El siguiente proceso tal y como se ha dicho es el del manipulador, este empieza cuando la pieza ha terminado el tratamiento térmico y la bandeja sale, etapa 10. En este momento el manipulador se dirige a por la pieza, el modo de mantenimiento puede ser activado durante este movimiento, haciendo que el manipulador quede detenido en el horno hasta que este finalice, pero si se recoge la pieza, el mantenimiento deberá esperar a que esta sea depositada y tratada por la estación de mecanizado. La transferencia de información sobre el color se produce en las etapas de recogida, 23 y 24. Una vez la pieza ha sido recogida y siempre y cuando la estación de mecanizado esté disponible, el horno queda liberado para demandar otra pieza y el manipulador transporta la pieza a la estación de mecanizado. La última etapa también es de espera, y en ella se transmite la información al proceso de mecanizado, este punto también se utiliza como final de proceso en caso de que se demande una parada del sistema, quedando la pieza en la mesa de la estación y teniendo que ser retirada o procesada manualmente.

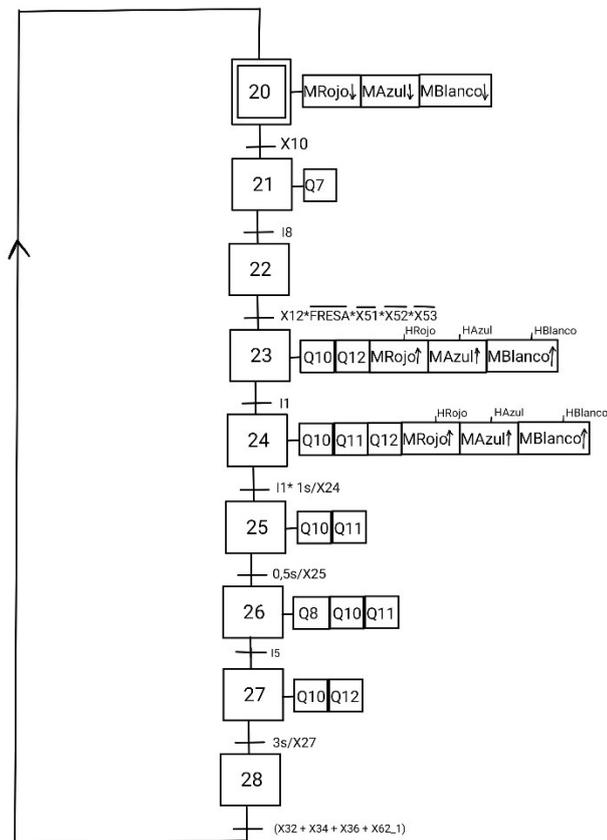


Imagen 87: GRAFCET Manipulador\_G20.

El uso del manipulador también está gobernado por otro GRAFCET, el encargado de la secuencia de mantenimiento la cual puede darse durante el funcionamiento automático. Esta secuencia básicamente consiste en trasladar el manipulador al horno para permitir que la estación de mecanizado pueda recibir un mantenimiento, debido a la posición que ocupará el manipulador no debe permitirse que el brazo lleve una pieza al horno, provocando un choque de estos dos,

por ello, si el horno está esperando a que el brazo transporte una pieza, etapas 1 y 2, no se permitirá el movimiento del manipulador hasta que el brazo haya sido retirado, etapa 3. Se han resuelto las incompatibilidades entre los graficets G20 y G50 asegurando que G50 puede activarse siempre y cuando G20 no esté manipulando una pieza, y permitiendo que si se ha pausado el proceso de recogida de una pieza, que este se pueda reanudar. Para evitar incompatibilidades también, este diagrama se reinicia con la demanda de parada del sistema.

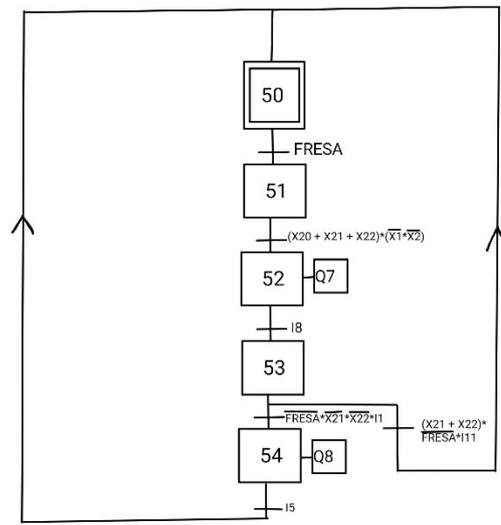


Imagen 88: GRAFCET ManipuladorF\_G50.

El último proceso es el de la estación de mecanizado, la secuencia comienza cuando el manipulador deposita la pieza, etapa 28, en caso de estar en funcionamiento semiautomático el operario deberá indicar que puede comenzar el proceso mediante la activación de Pulsador, en este momento se recibe la información referente al color y en función de él se mecanizará la pieza el tiempo estipulado, en caso de modo semiautomático, el operario también debe indicar qué proceso se ha de realizar. Cuando la información ha sido comunicada, flanco de bajada de la etapa 31, el manipulador queda liberado. Cuando está claro el proceso a llevar a cabo la mesa gira hasta la herramienta, se realiza el mecanizado, y la mesa sigue hasta la cinta. En esta posición se activa el empujador y la cinta para dar salida a la pieza que esperará al final de esta a ser recogida por el brazo o por un operario si el modo manual o la parada del sistema se activaron mientras la pieza era procesada. Una vez la pieza ha sido liberada y la mesa está en su posición inicial, a la que volverá después de expulsar la pieza, la estación queda libre para procesar otra pieza.

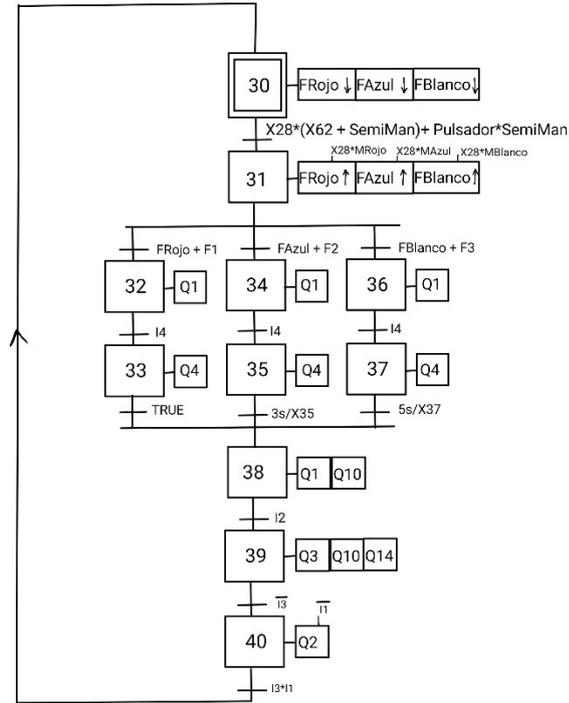


Imagen 89: GRAFCET Fresado\_G30.

En este caso debe transmitirse la información al brazo, esto se hará mediante las variables compartidas AROJO/Azul/Blanco y se ha diseñado un grafcet dedicado a esta transferencia para no bloquear el proceso de la estación. Es una secuencia con tan solo dos estados, el inicial reinicia los valores de las variables de color, una vez se activa la etapa final de la estación, la etapa 40, se activa la transferencia y se mantiene el valor activado hasta que el brazo ha depositado la pieza, en la etapa 124. En caso de encontrarse en modo semiautomático, llegada a esta etapa deberá indicarse mediante el pulsador correspondiente que color está siendo procesado, ya que la transferencia de la información del color se interrumpe con la interrupción de la secuencia del manipulador. Como esta etapa se da en el otro autómatas, se ha asociado una variable compartida que se activa con esta etapa, la variable es X124.

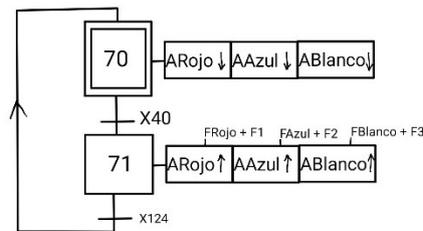


Imagen 90: GRAFCET Transferencia\_G70.

Por último, debe hablarse de la secuencia del brazo. Esta se divide en dos ramas bien diferenciadas. Después de llevarse el brazo a su posición inicial, paso obligatorio para reiniciar los contadores de pulsos, se abre la posibilidad de recoger una pieza nueva y llevarla al horno o de recoger una pieza ya finalizada y llevarla a la zona de almacenamiento. La última rama tendrá prioridad para evitar la congestión del sistema. El funcionamiento de las dos ramas es esencialmente el mismo, el brazo se dirige a la posición de recogida, calibrada como se ha explicado en el apartado 2.1.2, activa la succión para recoger la pieza y la lleva hasta la posición final, la bandeja del horno o la zona de almacenamiento. Acto seguido volverá a la posición inicial y quedará a la espera de una pieza en alguna de las dos zonas. Para la rama de la pieza inicial se tiene en cuenta también que no esté activado el modo de mantenimiento ni el horno ocupado.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
 MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

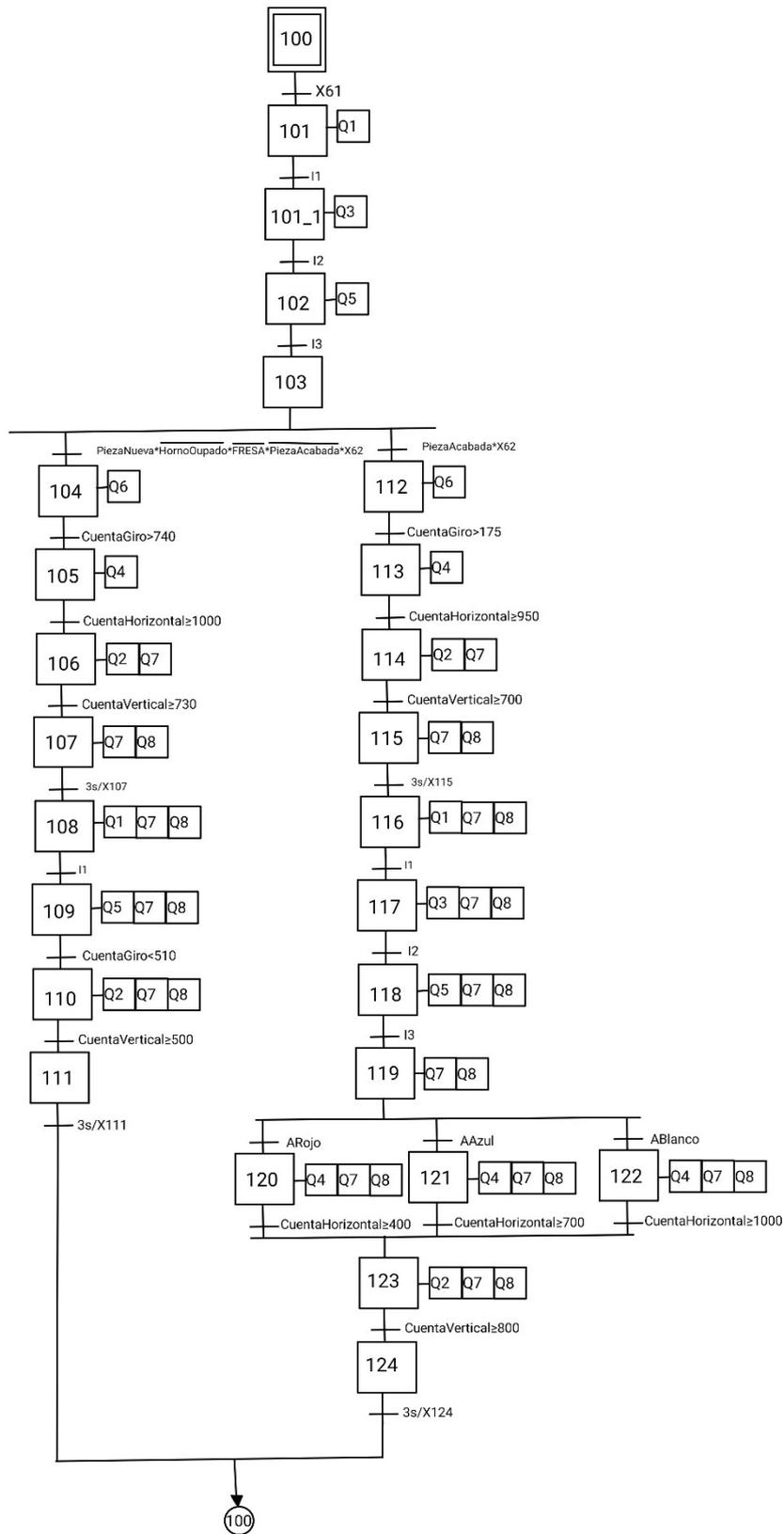


Imagen 91: GRAFCET BrazoManipulador\_G100.

Los procesos realizados según el color se resumen en la temperatura a la que se calienta el horno y los tiempos de permanencia en el horno y de mecanizado. Estos parámetros pueden cambiarse en el programa de LabView y en el programa del PLC para ajustarlos a las necesidades correspondientes. En el presente proyecto los parámetros seleccionados han sido:

- Un tiempo de permanencia en el horno de 5 segundos para todas las piezas.
- Una temperatura de referencia de 5º, 8º y 10ºC para los colores Rojo, Azul y Blanco respectivamente
- Un tiempo de mecanizado de 3 y 5 segundos para las piezas Azul y Blanca, mientras que a la Roja no se le realiza ningún mecanizado.



*Imagen 92: Piezas proceso.*

Las variables utilizadas para controlar el proceso han sido:

- PiezaNueva = Rojo + Azul + Blanco
- PiezaEnHorno =  $\overline{X0} * \overline{X1} * \overline{X2}$
- PiezaEnManipulador =  $\overline{X20} * \overline{X21} * \overline{X22}$
- PiezaEnFresado =  $\overline{X30}$
- PiezaAcabada =  $\overline{X40}$
- X124 = X124(etapa)



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERIA  
INDUSTRIAL VALÈNCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**PROYECTO DE AUTOMATIZACIÓN,  
MONITORIZACIÓN Y DIGITALIZACIÓN DE UN  
PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y  
COMUNICACIÓN OPC-UA**

**ANEXO II: DISEÑO DEL PID**

AUTOR: CARLOS FERRER CAMPOS

TUTOR: JOSÉ LUÍS PITARCH PÉREZ

COTUTOR: RAÚL SIMARRO FERNÁNDEZ

**Curso Académico: 2021-2022**

## 1 Introducción

En la memoria ha sido explicado el proceso de modelado del proceso térmico involucrado en el proyecto. Para el control del proceso no basta con esto, sino que debe desarrollarse un bucle de control tal y como se explica en el apartado 3.2.2. El desarrollo del controlador utilizado se detalla en este capítulo.

## 2 Planteamiento de las especificaciones

Según los criterios expuestos en los objetivos del proyecto, el sistema controlado debe cumplir que:

- ❖ Para asegurar la fluidez del proceso se ha limitado el tiempo que debe tardar la célula Peltier en calentarse a 15 segundos y se hará compatible la simultaneidad del proceso de fresado y el tratamiento térmico.
- ❖ Dada la potencial criticidad de los tratamientos a aplicar, es preferible un comportamiento sobreamortiguado de la variable controlada para asegurar la eficiencia del proceso.

El modelo determinado es uno de primer orden y sin retardo, por ello ha decidido aplicarse un método experimental para el cual se reúnen todas las condiciones necesarias.

El método utilizado ha sido el método Ciancone, en él se recogieron correlaciones tanto para cambios en la referencia como en la perturbación; en este caso solo será necesario el uso de las gráficas y tablas vinculadas al cambio de referencia, que es el aspecto que se considera más relevante dado el tipo de proceso. Según el tema 8 del libro Sintonización de Controladores PID de F. Javier Rivas (2009) las hipótesis en las que se basa el método Ciancone son las siguientes:

- ❖ Errores de (-25, +25) % en los parámetros del modelo.
- ❖ Modelo simple de primer orden con retardo.

El modelo definido cumple estas tres premisas, dado que las diversas pruebas realizadas pueden asegurar el margen de error del 50%, el modelo es simple sin retardo ( $\theta = 0$ ) y al no tener retardo y según este mismo método, no será necesaria la acción derivada. El método sin embargo no sirve para asegurar un tiempo de establecimiento o una sobreoscilación concretas, por lo que se deberá comprobar experimentalmente si el modelo cumple o debe realizarse algún ajuste.

### 3 Desarrollo de la solución

El método Ciancone utiliza un parámetro calculado a partir del modelo para determinar los valores de los parámetros del PID. Recordando la expresión del modelo:

$$G(s) = \frac{-0,46156}{2,9208 \cdot s + 1}$$

Ecuación 11: Modelo experimental.

Los parámetros principales de este modelo son:

- $K = -0.46156$
- $\tau = 2.9208$
- $\theta = 0$

A partir de ellos se calcula la fracción de retardo, valor a partir del cual se determinan el resto de los parámetros:

$$f_r = \frac{\theta}{\tau + \theta}$$

Ecuación 12: fracción de retardo.

En este caso  $f_r = 0$ .

Según Ciancone, para cambios en la consigna los valores según la ganancia del PID son:

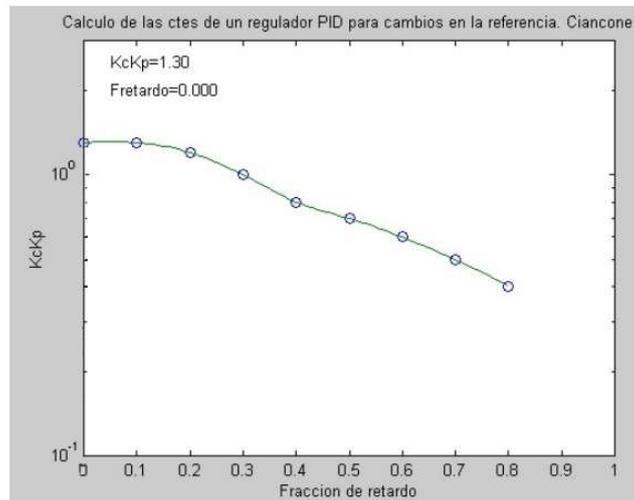


Imagen 93: Ciancone - Cálculo de la ganancia. Fuente: Seminario de Control, LAC.

Como la fracción de retardo es nulo, el producto entre la ganancia del sistema y del controlador debe ser 1,3.

$$K_c = 1,3/K_p$$

Ecuación 13: Determinación de la ganancia.

De esta manera se calcula que  $K_c = -2,81654$ .

Según Ciancone, para cambios en la consigna, los valores según el tiempo integral del PID son:

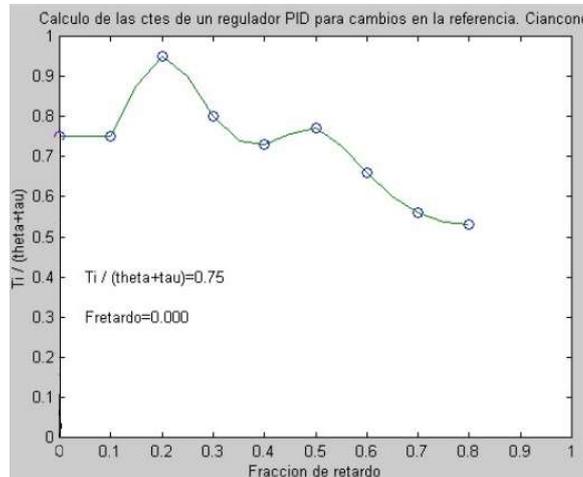


Imagen 94: Ciancone - Cálculo del tiempo integral. Fuente: Seminario de Control, LAC.

Como la fracción de retardo es nula y theta también, simplemente debe despejarse  $T_i$  para determinar su valor.

$$T_i = 0,75 \cdot \tau$$

Ecuación 14: Determinación del tiempo integral.

De esta manera se calcula que  $T_i = 2,1906$ .

Según Ciancone, para cambios en la consigna, los valores según el tiempo derivado del PID son:

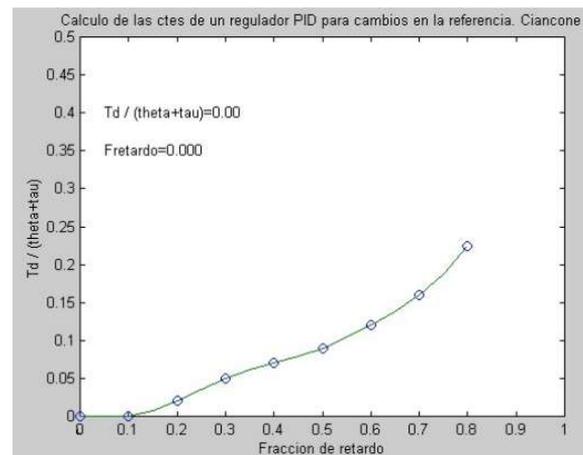


Imagen 95: Ciancone - Cálculo del tiempo derivado. Fuente: Seminario de Control, LAC.

Es trivial observar que, para una fracción de retardo nula, no existirá acción derivada,  $T_d = 0$ .

De esta forma se ha conseguido un modelo PI a partir de un método empírico, no obstante, debe contrastarse que este cumple las restricciones impuestas para validarlo.

La validación se ha hecho mediante un programa de LabView que únicamente se encargaba del control de la temperatura, tal y como se ha visto en el apartado 3.2.2 pero donde todas las entradas podían configurarse manualmente, y los resultados eran almacenados en un fichero al terminar el proceso, tal y como se hizo para la identificación del proceso. En este programa se tienen en cuenta tal y como se ha visto en el apartado 3.2.2 el mecanismo antiwindup y el filtro de la salida. También se ha utilizado el simulink para comprobar que el comportamiento real se asemejaba al teórico, se ha programado el bucle de control y se ha simulado el proceso, mostrando la salida de este junto con los datos reales obtenidos en el experimento hecho con el LabView. Los resultados son los siguientes:

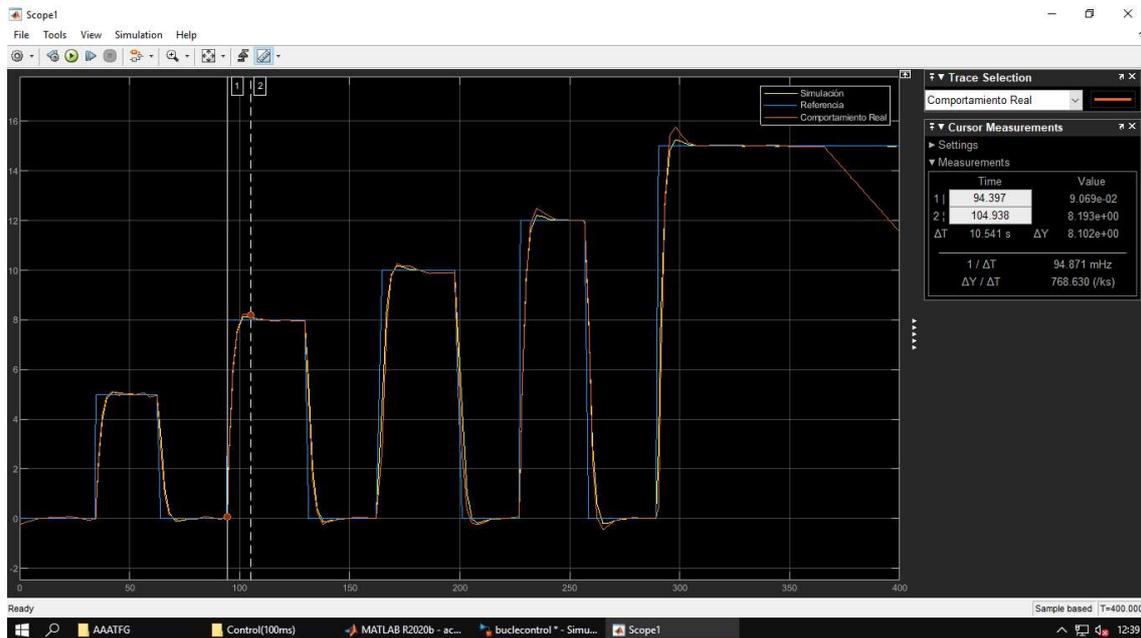


Imagen 96: Validación PI - R = 8°C.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

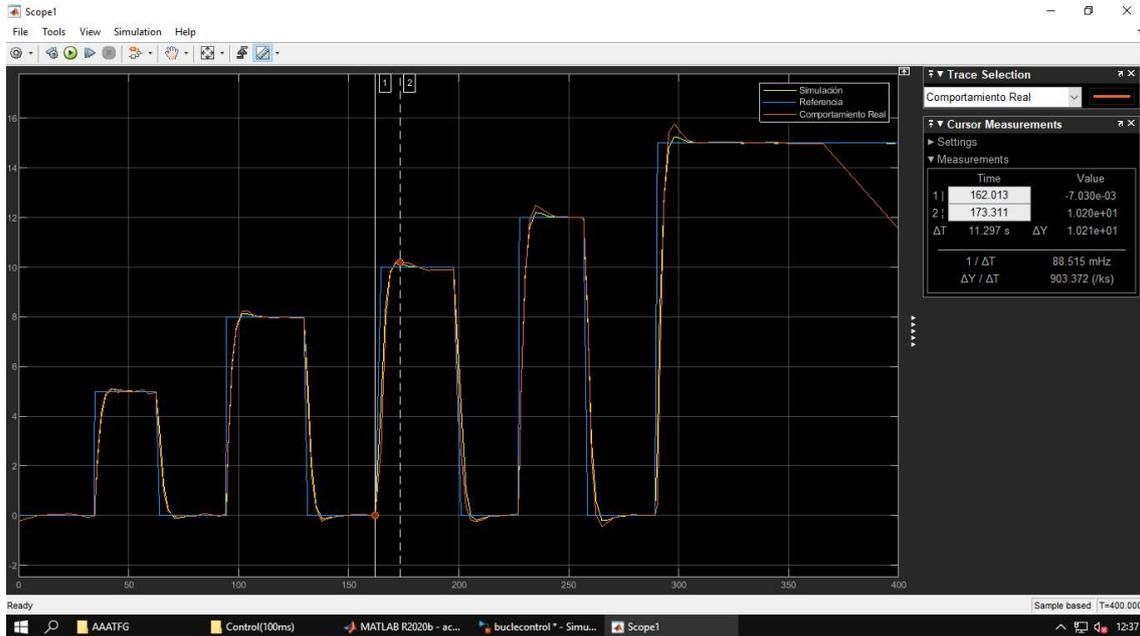


Imagen 97: Validación PI - R = 10°C.

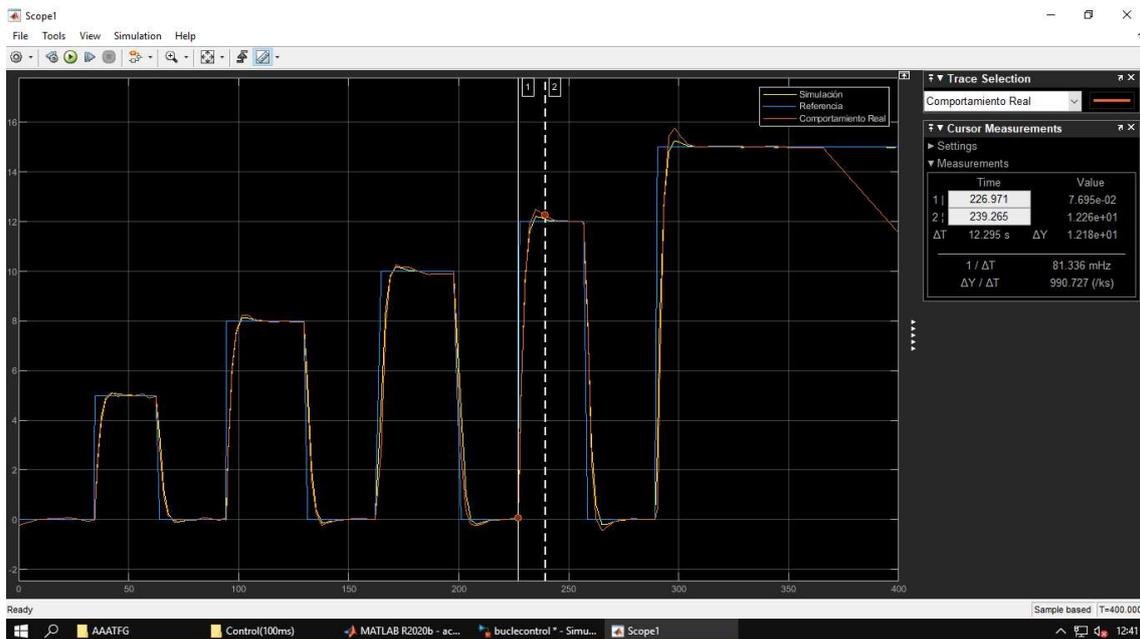


Imagen 98: Validación PI - R = 12°C.

Los resultados de la validación por tanto son:

	Testablecimiento (s)	Sobreoscilación (%)
R = 8°C	10,541	<20
R = 10°C	11,298	<20
R = 12°C	12,294	<20

Tabla 10: Resultados validación experimental.

Puede concluirse que, basándose en las restricciones impuestas, el controlador diseñado con datos empíricos cumple para los puntos de funcionamiento estudiados. Puede también observarse que tal y como se explicó al principio del apartado 1.3.4 de la memoria, el modelo posee ganancia variable, con lo que a medida que nos alejamos de punto de funcionamiento considerado ( $R = 10^{\circ}\text{C}$ ), el control funciona peor, observándose cada vez una sobreoscilación mayor y un mayor tiempo de establecimiento. Si se quisiera trabajar en otro punto de funcionamiento, a una temperatura mayor, debería volver a calcularse el modelo para ajustar la ganancia y repetir el procedimiento de diseño descrito para el PID.



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERIA  
INDUSTRIAL VALÈNCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**PROYECTO DE AUTOMATIZACIÓN,  
MONITORIZACIÓN Y DIGITALIZACIÓN DE UN  
PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y  
COMUNICACIÓN OPC-UA**

**ANEXO III: INSTALACIÓN Y CONFIGURACIÓN  
INFLUXDB**

AUTOR: CARLOS FERRER CAMPOS

TUTOR: JOSÉ LUÍS PITARCH PÉREZ

COTUTOR: RAÚL SIMARRO FERNÁNDEZ

**Curso Académico: 2021-2022**

## 1 Introducción

En el presente anexo va a tratarse el proceso seguido para instalar la base de datos InfluxDB y la plataforma de enlace Telegraf. Así como su configuración genérica para su implementación en cualquier proyecto. Debe tenerse en cuenta que para realizar la mayoría de las acciones que van a relatarse, es necesario ejecutar el PC en modo administrador.

## 2 Instalación

El primer paso es el de instalar InfluxDB y Telegraf, esto puede hacerse desde el portal de InfluxData. Allí se seleccionará la última versión y el sistema operativo correspondiente y se ejecutará la URL en la consola de PowerShell. La dirección es la siguiente:

<https://portal.influxdata.com/downloads/>

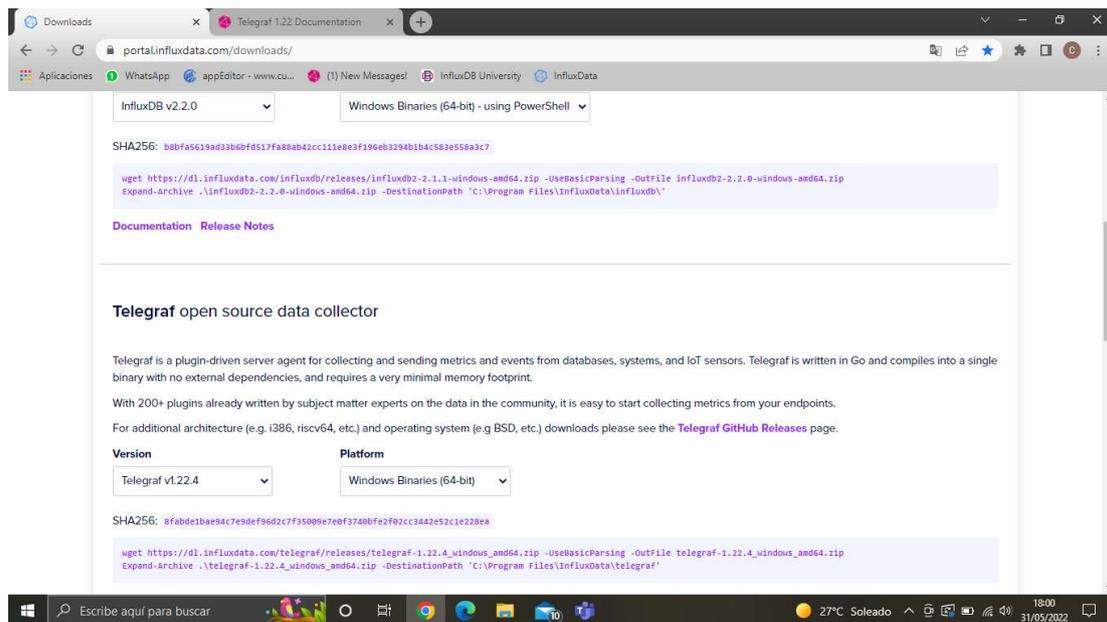


Imagen 99: Instalación InfluxDB y Telegraf.

Más adelante será necesario crear un servicio de InfluxDB, para hacerlo se ha utilizado el software NSSM el cual puede descargarse en la dirección:

<https://nssm.cc/download>

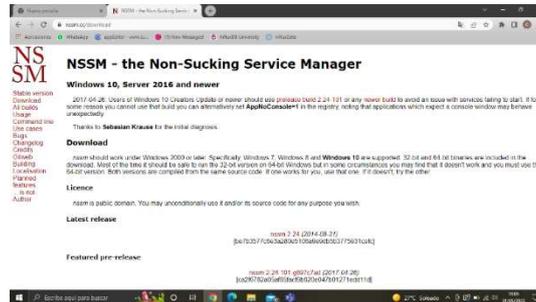


Imagen 100: Instalación NSSM.

Una vez descargado se ha de guardar el ejecutable en la carpeta InfluxData creada con la instalación del InfluxDB y Telegraf. La ruta de acceso suele ser: "C:\Program Files\InfluxData"

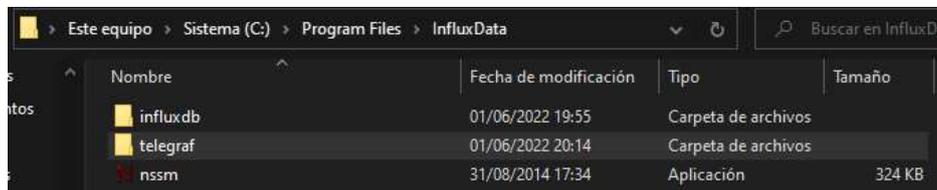


Imagen 101: Carpeta InfluxData.

### 3 Configuración de la base de datos

El primer paso para la configuración de la base de datos es la creación del servicio de InfluxDB, esto se puede hacer utilizando la aplicación NSSM instalada y la consola PowerShell:

```
PS C:\Windows\system32> cd "C:\Program Files\InfluxData"
PS C:\Program Files\InfluxData> .\nssm.exe install Influxdb
Service "Influxdb" installed successfully!
PS C:\Program Files\InfluxData> .\nssm.exe start InfluxDB
Influxdb: Unexpected status SERVICE_START_PENDING in response to START control.
PS C:\Program Files\InfluxData>
```

Imagen 102: Creación servicio InfluxDB.

Una vez ejecutado el código anterior aparecerá una ventana emergente creada por NSSM solicitando el archivo ejecutable del influxDB y la ruta de este.

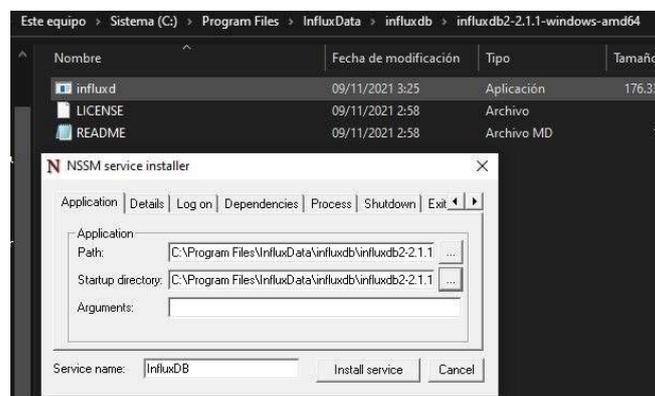


Imagen 103: Creación servicio InfluxDB.

Para activar el servicio hay que dirigirse a la aplicación de Windows “servicios” y desde allí buscar el servicio creado y activarlo, este paso solo deberá realizarse esta vez, el servicio se mantendrá activo a no ser que se lo desactive.

Con el servicio activado ya puede accederse a la base de datos, esta se encuentra en la URL:

<http://localhost:8086/>

Si es la primera vez que se abre InfluxDB desde el PC utilizado deberá hacerse un registro, definiendo un usuario y una contraseña, de lo contrario deberá iniciarse sesión con el usuario creado anteriormente.

Una vez dentro se empieza la configuración necesaria para comunicar el proyecto desarrollado, hay que dirigirse a la ventana de inicio y crear una organización junto con un bucket, en InfluxDB se llama “bucket” a la estructura que almacena los datos.

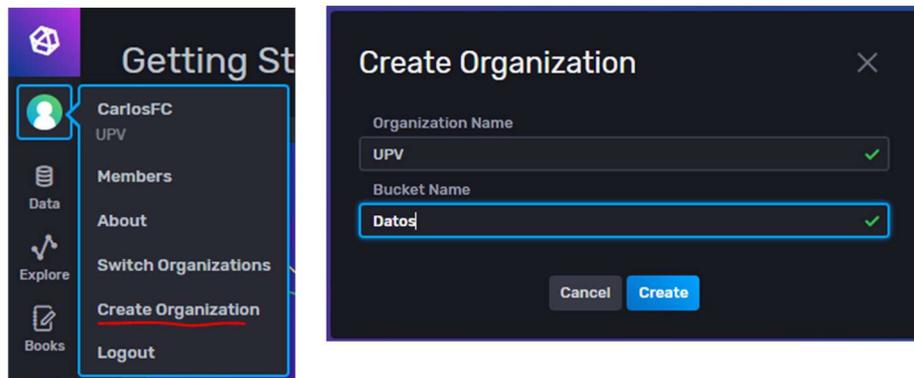


Imagen 104: Creación del bucket.

En el caso aplicado al proyecto el bucket debería llamar Monitorizacion para coincidir con el nombre configurado en el archivo de telegraf, la organización también es UPV, dentro de una misma organización puede haber diferentes buckets, y dentro de estos puede haber distintos “measurements”, que no son más que los diferentes conjuntos de datos introducidos desde una o varias fuentes.

Otro factor a tener en cuenta es el acceso a la base de datos por parte de Telegraf, para que la plataforma pueda acceder necesitará un Token, una especie de clave que da acceso a la lectura y escritura de variables en la base de datos, en este caso se creará un Token con acceso completo.

Para crear el token se accede a la pestaña Data→API tokens→Generate API Token, se generará un token con acceso completo, se lo nombrará como se desee y se copiará la clave de este que permitirá el acceso a la base de datos recién configurada.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

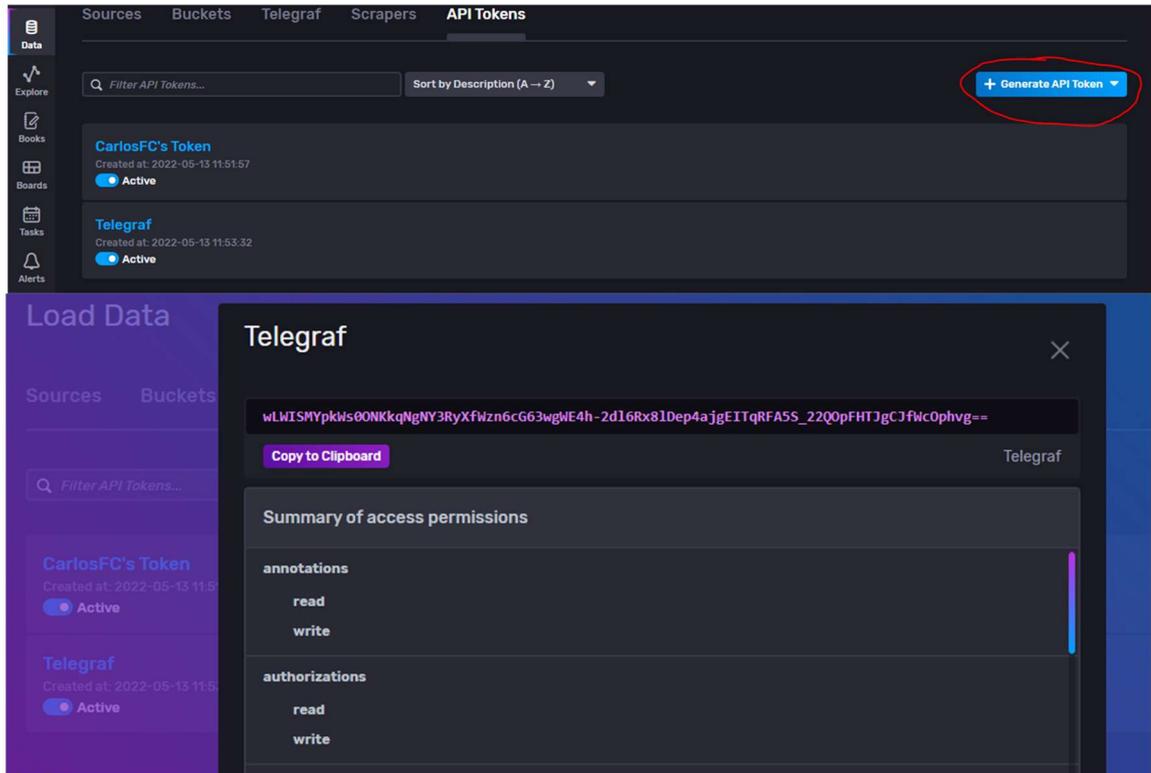


Imagen 105: Generación Token

En este caso el token tampoco coincide con el generado para la conexión hecha en el proyecto, este fue:

“lCuwYwpMzpWBB0xkHm-J\_SAWQfsv83Fts65KbHjplL2-XmUjE3hlyHHsec9YCVfk0Qrm9oSEDCQSIQ\_sFF4OQ==”

Aunque lo importante es comprender la generación y funcionamiento del token dado que al utilizar la base de datos desde otro PC se deberá generar un token diferente.

Siguiendo estos pasos debería obtenerse la información y medios suficientes como para configurar la comunicación tal y como se explica en el apartado 3.3 de la memoria.

**TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

**PROYECTO DE AUTOMATIZACIÓN,  
MONITORIZACIÓN Y DIGITALIZACIÓN DE UN  
PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y  
COMUNICACIÓN OPC-UA**

**PRESUPUESTO**

AUTOR: CARLOS FERRER CAMPOS

TUTOR: JOSÉ LUÍS PITARCH PÉREZ

COTUTOR: RAÚL SIMARRO FERNÁNDEZ

**Curso Académico: 2021-2022**

## 1 ELABORACIÓN DEL PRESUPUESTO

Para elaborar el presupuesto, este se ha dividido en dos capítulos. Estos, representan las dos fases en que se divide el proyecto. La primera, es la fase experimental en la cual se diseña, implementa y prueba el automatismo, así como se recaba la información y los datos necesarios para la redacción posterior del proyecto. La segunda, se corresponde con la fase de redacción del proyecto y la preparación de la documentación.

Para llevar a cabo el proyecto se considera que han colaborado dos ingenieros industriales con diferentes funciones por lo que la remuneración será diferente. El primero ejerce el papel de ingeniero de procesos, dedicado en exclusiva a la primera fase, es decir, al desarrollo e implementación del automatismo; y el segundo el de ingeniero de proyectos, el cual se ha encargado de la segunda fase del proyecto, pero también ha intervenido en ciertos momentos en la fase de diseño, como por ejemplo en el estudio de las alternativas y en la fase final de implementación para recabar la información necesaria. La suma total de las horas trabajadas entre ambos son las 300h que se ha tardado en desarrollar este proyecto.

## 2 CUADRO DE MANO DE OBRA

Para la ejecución del proyecto se han contratado dos ingenieros industriales, uno ejerciendo de ingeniero de procesos que cobra promedio 30.080€ anuales y otro de proyectos que cobra promedio 26.320€ anuales, para el cálculo de cuanto cobra cada uno por horas se ha tenido en cuenta que 2022 ha tenido 249 días hábiles descontando sábados, domingos y festivos. Se considera también que el ingeniero tiene derecho a dos semanas de vacaciones anualmente con lo que tendría 235 días laborables. Esto hace un total de 1880 horas trabajadas anualmente con lo que cada uno recibe 16€/h y 14€/h respectivamente.

		PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022		Pág.: 1
		CUADRO DE MANO DE OBRA		Ref.: PRESUPUESTO TFG
				06/22
Nº	Descripción	Importe		
		Precio (Euros)	Cantidad (Horas)	Total (Euros)
1	Ingeniero de automática o procesos	16,00	200,000 h	3.200,00
2	Ingeniero de Proyectos	14,00	100,000 h	1.400,00
			Importe total:	4.600,00

Tabla 11: Cuadro de Mano de Obra.

### 3 CUADRO DE MATERIALES

Se ha considerado como material utilizado las licencias software utilizadas para programar el automatismo, solo se tiene en cuenta el tiempo utilizado para desarrollar el proyecto, además no se incluyen en el presupuesto los softwares gratuitos empleados, estos son el SoMachine y InfluxDB.

Software	Licencia (€/año)	Coste Horario (€/h)
Unity Pro M	1.186,84	0,14
SoMAchine	0	0
LabView(Completo)	1.875	0,21
InfluxDB	0	0
Matlab+Simulink	2.000	0,23
Office	69	0,01

Tabla 12: Coste software.

PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022		Pág.: 2		
CUADRO DE MATERIALES		Ref.: PRESUPUESTO TFG		
		06/22		
Nº	Descripción	Importe		
		Precio (Euros)	Cantidad Empleada	Total (Euros)
1	Unity Pro M	0,14	30,000 h	4,20
2	Licencia LabView Completa	0,21	60,000 h	12,60
3	Licencia del software Matlab y su extensión Simulink	0,23	35,000 h	8,05
			Importe total:	24,85

Tabla 13: Cuadro de Materiales.

### 4 CUADRO DE MAQUINARIA

Se ha considerado como gasto de maquinaria todo aquel hardware utilizado tanto para diseñar y programar el automatismo, como para llevarlo a cabo. Mientras que la maquinaria utilizada para diseñar y programar el automatismo se tiene en cuenta solo en las horas en que ha sido utilizado, se supone que aquellos elementos que componen el sistema son una inversión y pasarán a ser propiedad del cliente con lo que se tiene en cuenta su coste total.

Componente	Precio (€)	Vida útil (años)	Horas de uso (h/día)	Coste Horario(€/h)
PC	1.000	8	8	0,04
Portátil	1.200	6	5	0,11
Tableta	750	4	5	0,10
TSX-Premium	1.100,00	-	-	-
M241	350	-	-	-
Kinect	25	-	-	-
Horno Multiproceso	600	-	-	-
Brazo Manipulador	545	-	-	-
TAD+Peltier	1.000	-	-	-

Tabla 14: Coste Hardware.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
 MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

		PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022		Pág.: 3
		CUADRO DE MAQUINARIA		Ref.: PRESUPUESTO TFG
				06/22
Nº	Descripción	Importe		
		Precio (Euros)	Cantidad	Total (Euros)
1	Ordenador Portátil HP-Laptop 15-bslxx	0,11	90,000 h	9,90
2	Tableta Lenovo Tab P11 Pro	0,10	90,000 h	9,00
3	Ordenador de sobremesa	0,04	140,000 h	5,60
4	PLC TSX-Premium:TSX PS7 2634M	1.100,00	1,000 u	1.100,00
5	PLC M241 TM241CE40R	350,00	1,000 u	350,00
6	Cámara kinect	25,00	1,000 u	25,00
7	Maqueta FischerTechnik de una estación multiproceso	600,00	1,000 u	600,00
8	Maqueta FischerTechnik de Brazo Manipulador	545,00	1,000 u	545,00
9	Célula Peltier, circuito electrónico integrado, sensores y TAD USB-6001 de National Instruments.	1.000,00	1,000 u	1.000,00
			Importe total:	3.644,50

Tabla 15: Cuadro de Maquinaria.

## 5 CUADROS DE PRECIOS UNITARIOS

		PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022		Pág.: 8					
		MEDICIONES Y PRESUPUESTOS		Ref.: PRESUPUESTO TFG					
		DISEÑO E IMPLEMENTACIÓN DEL AUTOMATISMO		06/22					
Nº Orden	Descripción de las unidades de obra	Uds.	Latitud	Longitud	Altura	Subtotal	Medición	Precio	Importe
<b>1</b>	<b>01 DISEÑO E IMPLEMENTACIÓN DEL AUTOMATISMO</b>								
1.1	U Estudio Alternativas								
01.01	Estudio previo de las Alternativas								
Total partida 1.1 .....							1,00	394,23	394,23
1.2	U Diseño Automatismo								
01.02	Fase de Diseño del Automatismo								
Total partida 1.2 .....							1,00	663,32	663,32
1.3	U Programación Automatismo								
01.03	Programación del Automatismo								
Total partida 1.3 .....							1,00	499,96	499,96
1.4	U Desarrollo y Programación del Programa de LabView								
01.04	Desarrollo y Programación del Programa de LabView								
Total partida 1.4 .....							1,00	1.012,54	1.012,54
1.5	U Desarrollo e Implementación Base de Datos								
01.05	Desarrollo e Implementación Base de Datos								
Total partida 1.5 .....							1,00	330,42	330,42
1.6	U Puesta en Marcha del sistema								
01.06	Puesta en Marcha del sistema								
Total partida 1.6 .....							1,00	4.451,87	4.451,87
<b>Total 01 Diseño e Implementación del automatismo .....</b>									<b>7.352,34</b>

Tabla 16: Cuadro de precios unitarios 1.

PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
 MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA

		PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022					Pág.: 9			
		MEDICIONES Y PRESUPUESTOS					Ref.: PRESUPUESTO TFG			
		PREPARACIÓN Y REDACCIÓN DE LA DOCUMENTACIÓN					06/22			
Nº Orden	Descripción de las unidades de obra	Uds.	Latitud	Longitud	Altura	Subtotal	Medición	Precio	Importe	
<b>2</b>	<b>02 PREPARACIÓN Y REDACCIÓN DE LA DOCUMENTACIÓN</b>									
2.1	U Preparación y redacción de la documentación									
02.01	Preparación y redacción de la documentación									
	Total partida 2.1 .....						1,00	1.165,08	1.165,08	
	<b>Total 02 Preparación y Redacción de la Documentación .....</b>								<b>1.165,08</b>	

Tabla 17: Cuadro de precios unitarios 2.

## 6 CUADROS DE PRECIOS DESCOMPUESTOS

		PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022					Pág.: 6			
		CUADRO DE PRECIOS DESCOMPUESTOS Y CUADRO DE PRECIOS Nº 1 Y Nº 2					Ref.: PRESUPUESTO ...			
		DISEÑO E IMPLEMENTACIÓN DEL AUTOMATISMO					06/22			
Nº Actividad	Código	Ud	Descripción			Rendimiento	Precio	Importe		
<b>1</b>	<b>01</b>		<b>DISEÑO E IMPLEMENTACIÓN DEL AUTOMATISMO</b>							
1.1	01.01	u	<b>ESTUDIO ALTERNATIVAS</b>							
			Estudio previo de las Alternativas							
	MO.01	h	Ingeniero de automática			15,000	16,00	240,00		
	Maq01	h	Ordensador Portátil			25,000	0,11	2,75		
	MO.02	h	Ingeniero de Proyectos			10,000	14,00	140,00		
	3,000	%	Costes indirectos				382,75	11,48		
			Clase: Mano de obra					380,00		
			Clase: Maquinaria					2,75		
			Clase: 3 % Costes indirectos					11,48		
			<b>Coste total</b>					<b>394,23</b>		
			TRESCIENTOS NOVENTA Y CUATRO EUROS CON VEINTITRES CÉNTIMOS							
1.2	01.02	u	<b>DISEÑO AUTOMATISMO</b>							
			Fase de Diseño del Automatismo							
	MO.01	h	Ingeniero de automática			40,000	16,00	640,00		
	Maq02	h	Tableta Lenovo Tab P11 Pro			40,000	0,10	4,00		
	3,000	%	Costes indirectos				644,00	19,32		
			Clase: Mano de obra					640,00		
			Clase: Maquinaria					4,00		
			Clase: 3 % Costes indirectos					19,32		
			<b>Coste total</b>					<b>663,32</b>		
			SEISCIENTOS SESENTA Y TRES EUROS CON TREINTA Y DOS CÉNTIMOS							
1.3	01.03	u	<b>PROGRAMACIÓN AUTOMATISMO</b>							
			Programación del Automatismo							
	MO.01	h	Ingeniero de automática			30,000	16,00	480,00		
	Maq03	h	Ordenador de Sobremesa			30,000	0,04	1,20		
	S01	h	Unity Pro M			30,000	0,14	4,20		
	3,000	%	Costes indirectos				485,40	14,56		
			Clase: Mano de obra					480,00		
			Clase: Maquinaria					1,20		
			Clase: Materiales					4,20		
			Clase: 3 % Costes indirectos					14,56		
			<b>Coste total</b>					<b>499,96</b>		
			CUATROCIENTOS NOVENTA Y NUEVE EUROS CON NOVENTA Y SEIS CÉNTIMOS							

Tabla 18: Cuadro de precios descompuestos 1.

**PROYECTO DE AUTOMATIZACIÓN, MONITORIZACIÓN Y DIGITALIZACIÓN DE UN PROCESO DE TRATAMIENTO TÉRMICO  
MEDIANTE PLCs SCHNEIDER Y COMUNICACIÓN OPC-UA**

Nº Actividad	Código	Ud	Descripción	Rendimiento	Precio	Importe
1.4	01.04	u	<b>DESARROLLO Y PROGRAMACIÓN DEL PROGRAMA DE LABVIEW</b>			
			Desarrollo y Programación del Programa de LabView			
	MO.01	h	Ingeniero de automática	60,000	16,00	960,00
	Maq03	h	Ordenador de Sobremesa	60,000	0,04	2,40
	S03	h	Licencia LabView Completa	60,000	0,21	12,60
	S05	h	Licencia Matlab + Simulink	35,000	0,23	8,05
	3,000	%	Costes indirectos		983,05	29,49
			Clase: Mano de obra			960,00
			Clase: Maquinaria			2,40
			Clase: Materiales			20,65
			Clase: 3 % Costes indirectos			29,49
			<b>Coste total</b>			<b>1.012,54</b>
			MIL DOCE EUROS CON CINCUENTA Y CUATRO CÉNTIMOS			
1.5	01.05	u	<b>DESARROLLO E IMPLEMENTACIÓN BASE DE DATOS</b>			
			Desarrollo e Implementación Base de Datos			
	MO.01	h	Ingeniero de automática	20,000	16,00	320,00
	Maq03	h	Ordenador de Sobremesa	20,000	0,04	0,80
	3,000	%	Costes indirectos		320,80	9,62
			Clase: Mano de obra			320,00
			Clase: Maquinaria			0,80
			Clase: 3 % Costes indirectos			9,62
			<b>Coste total</b>			<b>330,42</b>
			TRESCIENTOS TREINTA EUROS CON CUARENTA Y DOS CÉNTIMOS			
1.6	01.06	u	<b>PUESTA EN MARCHA DEL SISTEMA</b>			
			Puesta en Marcha del sistema			
	MO.01	h	Ingeniero de automática	35,000	16,00	560,00
	MO.02	h	Ingeniero de Proyectos	10,000	14,00	140,00
	Maq02	h	Tableta Lenovo Tab P11 Pro	10,000	0,10	1,00
	Maq03	h	Ordenador de Sobremesa	30,000	0,04	1,20
	Maq04	u	TSX-Premium	1,000	1.821,79	1.821,79
	Maq05	u	M241	1,000	1.071,00	1.071,00
	Maq06	u	Cámara Kinect	1,000	25,00	25,00
	Maq07	u	Estación Multiproceso	1,000	600,00	600,00
	Maq08	u	Brazo Manipulador	1,000	545,00	545,00
	Maq09	u	Célula Peltier + TAD	1,000	1.000,00	1.000,00
	3,000	%	Costes indirectos		5.764,99	172,95
			Clase: Mano de obra			700,00
			Clase: Maquinaria			5.064,99
			Clase: 3 % Costes indirectos			172,95
			<b>Coste total</b>			<b>5.937,94</b>
			CINCO MIL NOVECIENTOS TREINTA Y SIETE EUROS CON NOVENTA Y CUATRO CÉNTIMOS			
<b>2</b>	<b>02</b>		<b>PREPARACIÓN Y REDACCIÓN DE LA DOCUMENTACIÓN</b>			
2.1	02.01	u	<b>PREPARACIÓN Y REDACCIÓN DE LA DOCUMENTACIÓN</b>			
			Preparación y redacción de la documentación			
	MO.02	h	Ingeniero de Proyectos	80,000	14,00	1.120,00
	Maq01	h	Ordenador Portátil	65,000	0,11	7,15
	Maq02	h	Tableta Lenovo Tab P11 Pro	40,000	0,10	4,00
	3,000	%	Costes indirectos		1.131,15	33,93
			Clase: Mano de obra			1.120,00
			Clase: Maquinaria			11,15
			Clase: 3 % Costes indirectos			33,93
			<b>Coste total</b>			<b>1.165,08</b>
			MIL CIENTO SESENTA Y CINCO EUROS CON OCHO CÉNTIMOS			

Tabla 19: Cuadro de precios descompuestos 2.

## 7 PRESUPUESTO DE EJECUCIÓN MATERIAL

	PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022	Pág.: 10
	MEDICIONES Y PRESUPUESTOS	Ref.: PRESUPUESTO TFG
	RESUMEN DE CAPÍTULOS	06/22

### Presupuesto de ejecución material

1 Diseño e Implementación del automatismo	7.352,34
2 Preparación y Redacción de la Documentación	1.165,08
<b>Total .....</b>	<b>8.517,42</b>

Asciende el presupuesto de ejecución material a la expresada cantidad de OCHO MIL QUINIENTOS DIECISIETE EUROS CON CUARENTA Y DOS CÉNTIMOS.

Tabla 20: Presupuesto de ejecución material.

## 8 PRESUPUESTO BASE DE LICITACIÓN

	PRESUPUESTO PARA EL TFG DE CARLOS FERRER, AÑO 2022	Pág.: 11
	RESUMEN DE PRESUPUESTO	Ref.: PRESUPUESTO TFG
	RESUMEN DE CAPÍTULOS	06/22

Nº Orden	Código	Descripción de los capítulos	Importe	%
1	01	Diseño e Implementación del automatismo	7.352,34	86,32
2	02	Preparación y Redacción de la Documentación	1.165,08	13,68
<b>PRESUPUESTO DE EJECUCIÓN MATERIAL .....</b>			<b>8.517,42</b>	
13% Gastos Generales.....			1.107,26	
6% Beneficio Industrial.....			511,05	
<b>PRESUPUESTO .....</b>			<b>10.135,73</b>	
21% IVA.....			2.128,50	
<b>PRESUPUESTO + IVA .....</b>			<b>12.264,23</b>	

Suma el presente presupuesto más IVA la cantidad de:  
 DOCE MIL DOSCIENTOS SESENTA Y CUATRO EUROS

Tabla 21: Presupuesto base de licitación.