



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**PROYECTO DE AUTOMATIZACIÓN DE UNA
ESTACIÓN MULTIPROCESO Y UN BRAZO
MANIPULADOR MEDIANTE AUTÓMATAS
SIEMENS S7-1200 Y SCHNEIDER M241
CON MONITORIZACIÓN E INTERACCIÓN
A TRAVÉS DE UNA HMI OMRON**

AUTOR: Juan Carlos García Hernández

TUTOR: Raúl Simarro Fernández

Curso Académico: 2017-18

RESUMEN

En el presente proyecto, se ha desarrollado el diseño y la posterior implementación de la automatización de un conjunto formado por una estación multiproceso y un brazo manipulador, incluyendo la incorporación de un sistema *HMI* para la interacción y la monitorización de diversas señales del sistema.

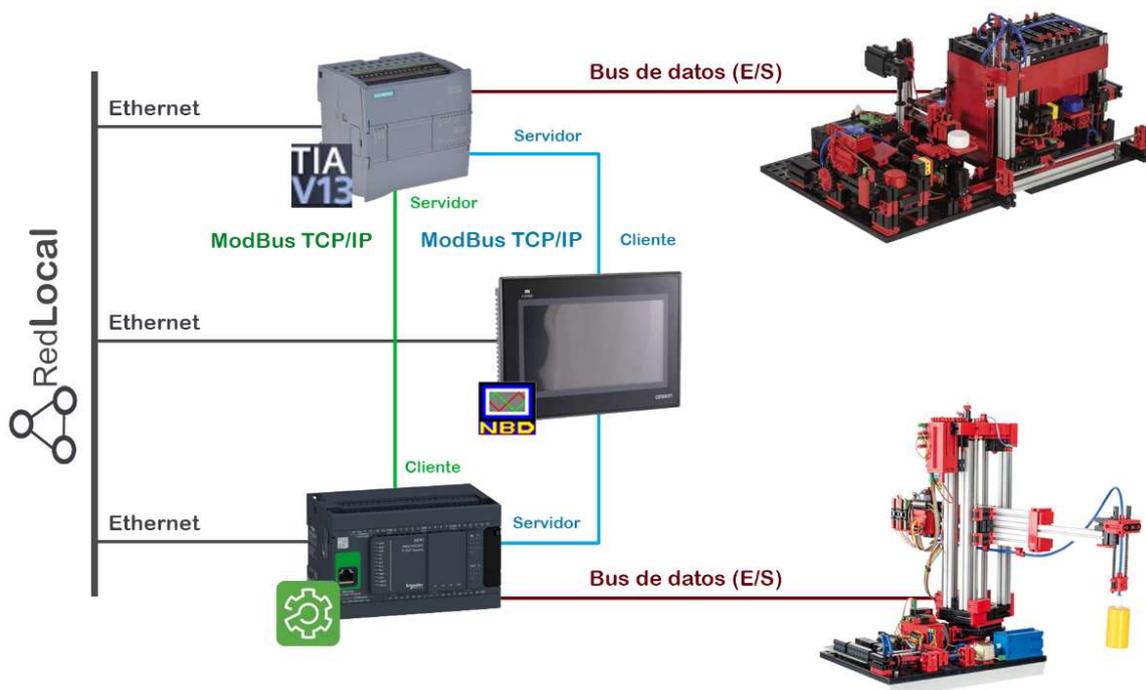
Para diseñar el sistema se ha utilizado la herramienta *GRAF CET (Grphe Fonctionnel de Commande Etape Transition)*, la cual se ha desarrollado utilizando el direccionamiento de variables y descripción del funcionamiento ofrecida por el cliente, quien ofreció dos maquetas que simulan el proceso real: *Fischer Technik (Multi Processing Station with Oven 24V)* para la estación multiproceso, y *Fischer Technik (Vacuum Gripper Robot 24V)* para el brazo manipulador.

El diseño se ha implementado sobre los autómatas *Siemens S7-1200* y *Schneider M241* a través de la programación y posterior transferencia de los programas mediante los softwares *Tia Portal* y *So Machine* respectivamente.

El diseño y la implementación del sistema *HMI* se ha realizado mediante el software *NB-Designer*, debido a que, el dispositivo elegido para realizar esta función ha sido *Omron NB7W*.

Por último, se ha dotado al sistema de la capacidad de comunicación mediante el protocolo *ModBus TCP/IP*, existiendo así una transferencia entre todos los dispositivos que forman el sistema.

Palabras clave: automatización, diseño, *GRAF CET*, *FischerTechnik*, implementación, *Siemens*, *Schneider*, *Omron*, comunicación, *ModBus*.



RESUM

En el present projecte, s'ha desenvolupat el disseny i la posterior implementació de l'automatització d'un conjunt format per una estació multiprocés i un braç manipulador, incloent la incorporació d'un sistema HMI per a la interacció i la monitorització de diversos senyals del sistema.

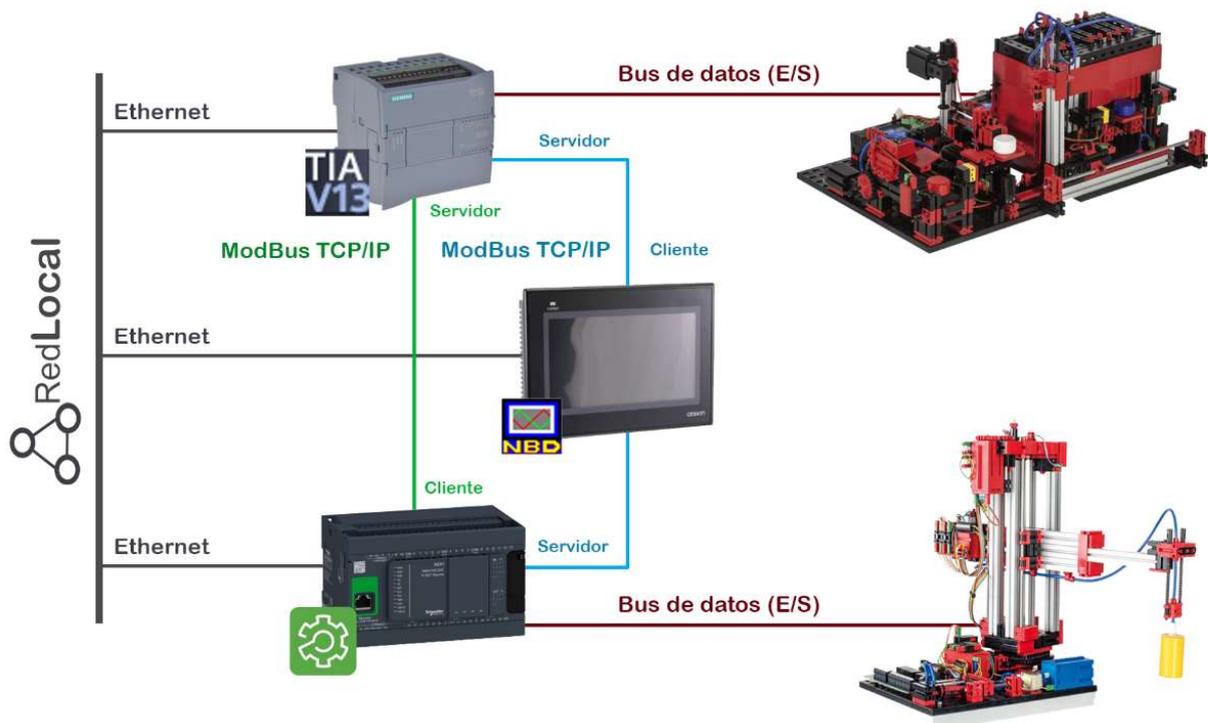
Per dissenyar el sistema s'ha utilitzat l'eina *GRAF CET* (*graphè fonctionnel de commande Etape Transition*), la qual s'ha desenvolupat utilitzant el direccionament de variables i descripció del funcionament oferta pel client, que va oferir dues maquetes que simulen el procés real: *Fischer Technik* (*Multi Processing station with Oven 24V*) per a l'estació multi-procés, i *Fischer Technik* (*Vacuum Gripper Robot 24V*) per al braç manipulador.

El disseny s'ha implementat sobre els autòmats *Siemens S7-1200* i *Schneider M241* a través de la programació i posterior transferència dels programes mitjançant els softwares *Tia Portal* i *So Machine* respectivament.

El disseny i la implementació del sistema HMI s'ha realitzat mitjançant el software *NB-Designer*, pel fet que, el dispositiu triat per a realitzar aquesta funció ha estat *Omron NB7W*.

Finalment, s'ha dotat al sistema de la capacitat de comunicació mitjançant el protocol *ModBus TCP / IP*, existint així una transferència entre tots els dispositius que formen el sistema.

Paraules clau: automatització, disseny, *GRAF CET*, *Fischer Technik*, implementació, *Siemens*, *Schneider*, *Omron*, comunicació, *ModBus*.



ABSTRACT

In the present project, the design and subsequent implementation of the automation of a set consisting of a multiprocessing station and a manipulator arm has been developed, including the incorporation of an *HMI* system for the interaction and monitoring of various system signals.

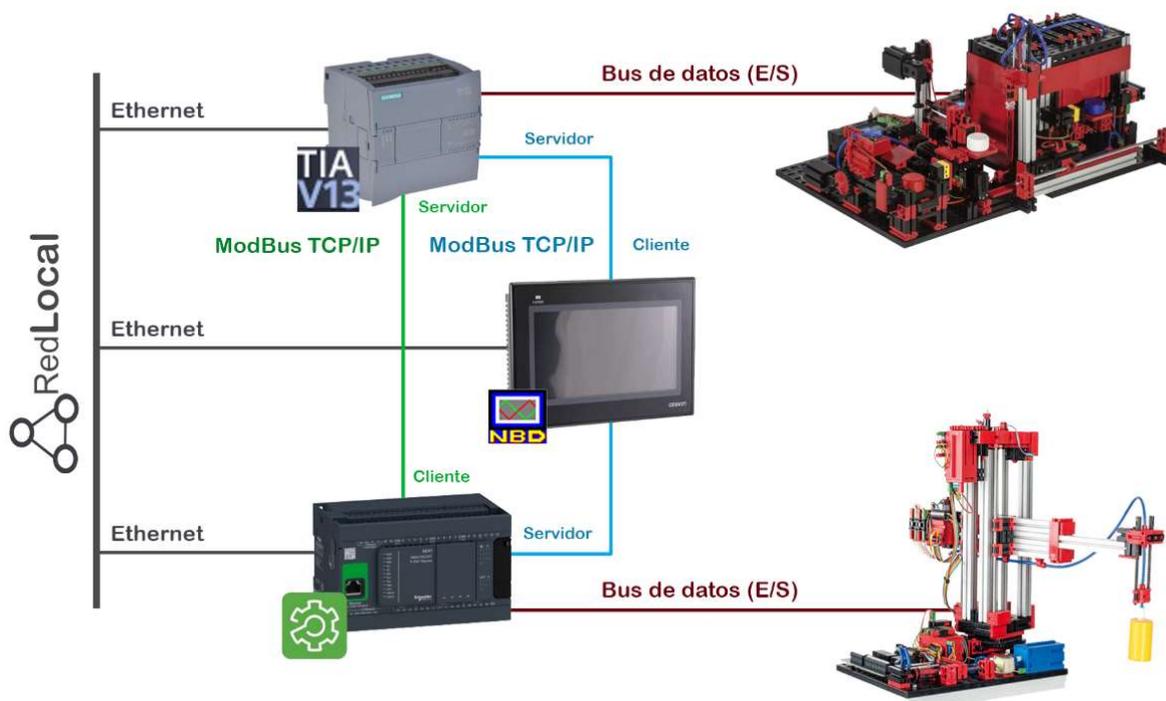
GRAF CET (Grphe Fonctionnel of Commande Etape Transition) tool has been used for the automatic design, which was developed using the variable addressing and description of the operation offered by the client, who offered two models that simulate the real process: *Fischer Technik (Multi Processing Station with Oven 24V)* for the multiprocessing station, and *Fischer Technik (Vacuum Gripper Robot 24V)* for the manipulator arm.

The design has been implemented on the *Siemens S7-1200* and *Schneider M241* PLCs through the programming and subsequent transfer of the programs through the *Tia Portal* and *So Machine* software respectively.

The design and implementation of the *HMI* system has been carried out using the *NB-Designer* software, because the device chosen to perform this function has been *Omron NB7W*.

Finally, the system has been equipped with the communication capability *vía ModBus TCP / IP* protocol, thus there is a transfer between all the devices that make up the system.

Keywords: automation, design, *GRAF CET*, *FischerTechnik*, implementation, *Siemens*, *Schneider*, *Omron*, communication, *ModBus*.



DOCUMENTOS CONTENIDOS EN EL PROYECTO

- MEMORIA
- PRESUPUESTO
- ANEXO I
- MANUAL DE USUARIO
- PLANOS

Índice del documento MEMORIA

1. OBJETO DEL PROYECTO	5
2. ESTUDIO DE NECESIDADES, FACTORES A CONSIDERAR: LIMITACIONES Y CONDICIONANTES ...	6
3. PLANTEAMIENTO DE SOLUCIONES ALTERNATIVAS Y JUSTIFICACIÓN DE LA SOLUCIÓN ADOPTADA	7
4. FUNCIONAMIENTO Y DESCRIPCIÓN DEL SISTEMA	9
4.1. Estación multiproceso	9
4.2. Brazo-ventosa.....	11
4.3. HMI (Human Machine Interface)	12
4.4. Distribución en planta	13
5. DESARROLLO DE LA SOLUCIÓN	14
5.1. Diseño del sistema automático	14
5.1.1. Seguridad.....	16
5.1.2. Modos de funcionamiento	17
5.1.3. Inicialización	19
5.1.4. Funcionamiento.....	20
5.2. Implementación del sistema automático.....	24
5.2.1. Implementación en Siemens S7-1200.....	24
5.2.2. Implementación en Schneider M241	34
5.3. Implementación de la interfaz de usuario	45
5.4. Implementación de la comunicación (ModBus TCP/IP)	55

5.4.1.	Implementación en Siemens S7-1200	56
5.4.2.	Implementación en Schneider M241	57
5.4.3.	Implementación en Omron NB7W	58
5.4.4.	Direccionamiento ModBus	59
6.	PUESTA EN MARCHA	65
7.	CONCLUSIONES	66
8.	BIBLIOGRAFÍA	67

Índice del documento PRESUPUESTO

1.	CUADRO Nº1: MANO DE OBRA	73
2.	CUADRO Nº2: MATERIALES Y MAQUINARIA	73
3.	CUADRO Nº3: PRECIOS UNITARIOS	74
4.	CUADRO Nº4: PRECIOS DESCOMPUESTOS	74
5.	PRESUPUESTO BASE DE LICITACIÓN	76

Índice del documento ANEXO I

1.	VARIABLES SIEMENS S7-1200	81
1.1.	Entradas y salidas (E/S)	81
1.2.	Registros	82
1.2.1.	Seguridad	82
1.2.2.	Modos de funcionamiento	83
1.2.3.	Estación multiproceso	84
1.2.3.1.	Alimentador	84
1.2.3.2.	Manipulador	85
1.2.3.3.	Mesa	86
1.2.3.4.	Cinta	86
1.2.3.5.	Modo manual	87
2.	VARIABLES SCHNEIDER M241	88

2.1.	Entradas y salidas (E/S).....	88
2.2.	Registros.....	89
2.2.1.	Brazo Ventosa.....	89
2.2.2.	Configuración cajas.....	89
2.2.3.	Gestión paro.....	90
2.2.4.	Modo manual.....	90
2.2.5.	Matrices de coordenadas.....	90
2.2.5.1.	CE2x2.....	90
2.2.5.2.	CE3x3.....	90
2.2.5.3.	CS2x2.....	91
2.2.5.4.	CS3x3.....	91

Índice del documento MANUAL DE USUARIO

1.	DESCRIPCIÓN Y FUNCIONALIDADES DE LAS PANTALLAS PRINCIPALES.....	97
1.1.	Pantalla <i>Inicio</i>	97
1.1.1.	Funcionamiento modo automático.....	98
1.2.	Pantalla <i>Información de la Producción</i>	99
1.2.1.	Detención de errores.....	100
1.3.	Pantalla <i>Modo manual</i>	101
1.3.1.	Funcionamiento en Modo manual.....	102
1.4.	Pantalla <i>Configuración</i>	103
1.5.	Pantalla <i>Tiempos</i>	104
1.5.1.	Configuración de una variable temporal.....	105
1.6.	Pantalla <i>Configuración cajas</i>	105
1.7.	Pantalla <i>Log In</i>	106
1.7.1.	Inicio de sesión.....	107
2.	EMERGENCIA.....	108
3.	PARADA DEL SISTEMA.....	111

Índice del documento PLANOS

1.	PLANO DISEÑO DEL SISTEMA AUTOMÁTICO.....	117
-----------	---	------------



MEMORIA

JUAN CARLOS GARCÍA HERNÁNDEZ



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





Índice

1. OBJETO DEL PROYECTO	5
2. ESTUDIO DE NECESIDADES, FACTORES A CONSIDERAR: LIMITACIONES Y CONDICIONANTES	6
3. PLANTEAMIENTO DE SOLUCIONES ALTERNATIVAS Y JUSTIFICACIÓN DE LA SOLUCIÓN ADOPTADA	7
4. FUNCIONAMIENTO Y DESCRIPCIÓN DEL SISTEMA	9
4.1. Estación multiproceso	9
4.2. Brazo-ventosa.....	11
4.3. HMI (Human Machine Interface)	12
4.4. Distribución en planta	13
5. DESARROLLO DE LA SOLUCIÓN	14
5.1. Diseño del sistema automático	14
5.1.1. Seguridad.....	16
5.1.2. Modos de funcionamiento	17
5.1.3. Inicialización	19
5.1.4. Funcionamiento.....	20
5.2. Implementación del sistema automático.....	24
5.2.1. Implementación en Siemens S7-1200.....	24
5.2.2. Implementación en Schneider M241	34
5.3. Implementación de la interfaz de usuario	45
5.4. Implementación de la comunicación (ModBus TCP/IP)	55
5.4.1. Implementación en Siemens S7-1200.....	56
5.4.2. Implementación en Schneider M241	57
5.4.3. Implementación en Omron NB7W	58
5.4.4. Direccionamiento ModBus.....	59
6. PUESTA EN MARCHA.....	65
7. CONCLUSIONES.....	66
8. BIBLIOGRAFÍA	67



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





1. OBJETO DEL PROYECTO

El objeto del presente proyecto se centra en el diseño y la implementación de un sistema automático, el cual incorporará un sistema de visualización y configuración mediante una interfaz de usuario.

La línea de producción a automatizar constará de una estación multiproceso encargada de procesar piezas proporcionadas a través de cajas de distinto tamaño, utilizando procesos térmicos (horneado) y mecánicos (serrado), contando también con elementos encargados del transporte de dichas piezas, para posteriormente, introducir las piezas de nuevo en cajas de cartón de distinto tamaño.

También se deberá automatizar el elemento encargado de transportar las piezas desde las cajas de entrada hasta la posición inicial de la línea de producción, y de transportar las piezas de la posición final de la línea de producción hasta las cajas de salida.

Se establece que el sistema automático sea jerárquico, existiendo niveles superiores que ejerzan acciones sobre niveles inferiores pudiendo así implementar sistemas de seguridad en el primer nivel, un sistema de modos de funcionamiento en el segundo nivel, y, por último, un sistema de inicialización y funcionamiento como tercer nivel.



2. ESTUDIO DE NECESIDADES, FACTORES A CONSIDERAR: LIMITACIONES Y CONDICIONANTES

En el presente proyecto, se impusieron una serie de limitaciones y condicionantes que dieron lugar a una serie de restricciones que han acotado la solución final. Las limitaciones y condicionantes se han clasificado en dos grupos claramente diferenciados.

- *Limitaciones y condicionantes que nos dictan las normas.*
- *Limitaciones y condicionantes que nos dicta el encargo.*

Limitaciones y condicionantes que nos dictan las normas.

Las normas que se han tenido que cumplir durante el desarrollo del sistema han sido:

- **IEC 61131-3:** Parte 3 del estándar internacional IEC 61131 para Controladores Lógicos Programables (PLC)
- **UNE-EN 60848:2013:** Lenguaje de especificación GRAFCET para diagramas funcionales secuenciales.
- **UNE-EN ISO 13849-1:2008:** Seguridad de las máquinas. Partes de los sistemas de mando relativas a la seguridad. Parte 1: Principios generales para el diseño.
- **UNE-EN 61158-1:2014:** Redes de comunicaciones industriales. Especificación de Fieldbus. Parte 1: Generalidades y guía para la serie de normas IEC 61158 e IEC 61784.
-

Limitaciones y condicionantes que nos dicta el encargo.

Se imponen basándose en el encargo realizado por el cliente, y acotan el diseño.

En cuanto a la automatización de los procesos, el cliente ofreció dos maquetas de la marca *Fischer Technik*, las cuales simulan de forma adecuada las tareas que realizará el proceso real, por lo tanto, se tiene la primera restricción, diseñar el sistema automático específicamente para las maquetas utilizando el direccionamiento de variables que aparece en el *Anexo I*.

Por otro lado, en cuanto a diseño de la interfaz de usuario, el cliente expuso una serie de características a incorporar las cuales se exponen a continuación:

- Seleccionar entre diversos modos de funcionamiento.
- Monitorizar señales del sistema.
- Modificar parámetros como tiempos y tamaño de cajas de entrada y de salida.
- Restringir funciones dependiendo del usuario.
- Bloquear el sistema en caso de emergencia.



3. PLANTEAMIENTO DE SOLUCIONES ALTERNATIVAS Y JUSTIFICACIÓN DE LA SOLUCIÓN ADOPTADA

Realizar un estudio exhaustivo sobre las distintas soluciones alternativas que se podrían haber elegido para resolver el objeto de un proyecto, sería sumamente importante si el proyecto a resolver fuera de carácter profesional, y, por lo tanto, es obvio que la solución adoptada finalmente, sería la óptima, y se hubiera obtenido comparando soluciones, basándose en aspectos como la facilidad de implementación, calidad, coste, etc.

Por el contrario, el presente proyecto es de carácter educativo, por lo tanto, se va a justificar la elección de la solución adoptada mediante un enfoque educativo.

La primera elección fue decidir si el diseño de nuestro sistema se iba a realizar para la posterior implementación en lógica programada o en lógica cableada. La elección escogida era obvia, ya que, actualmente, la industria está en proceso de evolución hacia lo que ya se conoce como la *Industria 4.0*, en la cual, en un futuro, todos los medios de producción estarán automatizados y recibirán constantemente información que hará que varíe su forma de actuar. Por lo tanto, bajo una perspectiva estudiantil, se decide optar por la lógica programada, ya que será la base de esta nueva Industria.

Una vez elegido el tipo de tecnología a emplear, fue necesario decidir qué hardware se iba a emplear. De nuevo, si se hubieran comparado las alternativas mediante un enfoque profesional, se habría elegido la solución en la cual se utilizase el menor hardware posible con la mayor compatibilidad posible en aspectos de comunicación entre los diversos componentes, por ejemplo, utilizar dos autómatas de la marca Siemens y utilizar una comunicación de tipo *PROFINET*. Sin embargo, bajo una perspectiva estudiantil, se decide optar por 3 elementos de marcas distintas y utilizar el protocolo de comunicación *ModBus TCP/IP* para la comunicación del sistema.

A continuación, se muestra una lista del Hardware empleado y su posterior descripción y justificación la cual aparece en la *Tabla 3.1*.

- *PLC Siemens SIMATIC S7-1200 con módulo de ampliación de entradas y salidas.*
- *PLC Schneider MODICON M241*
- *HMI Omron NB7W-TW01B*

Nombre	Imagen identificativa	Software para programarlos	Lenguajes de programación
Siemens SIMATIC S7-1200		<i>Tia Portal</i>	<ul style="list-style-type: none"> - <i>KOP (Ladder)</i> - <i>SL (Texto Estructurado)</i>
Schneider MODICON M241		<i>SoMachine</i>	<ul style="list-style-type: none"> - <i>LD (Ladder)</i> - <i>ST (Texto Estructurado)</i> - <i>SFC</i>
Omron NB7W- TW01B		<i>NB-Designer</i>	<ul style="list-style-type: none"> - <i>Lenguaje gráfico mediante objetos funcionales por capas</i>

Tabla 3.1. Tabla resumen del hardware escogido

Como se puede observar, en este apartado no se centra en las características técnicas, ya que para esto ya se tienen los manuales de los correspondientes dispositivos (*S7-1200, Manual del sistema*), (*Modicon M241 Logic Controller, Guía de programación*), (*NB-SERIES, STEUP MANUAL*), en los cuales se explican detalladamente todas las características técnicas del hardware empleado en este proyecto.



4. FUNCIONAMIENTO Y DESCRIPCIÓN DEL SISTEMA

Como ya se expuso en el *capítulo 2*, el cliente acota la solución proporcionando las maquetas a escala reducida del proceso y los requerimientos que debería incorporar la HMI. En este apartado, se procede a introducir dichos componentes y explicar brevemente el funcionamiento de cada uno en el sistema.

4.1. Estación multiproceso

Para la simulación de la línea de producción el cliente ofrece la maqueta *Fischer Technik (Multi Processing Station with Oven 24V)* mostrada en la *Figura 4.1* a la cual se identificó como: estación multiproceso.

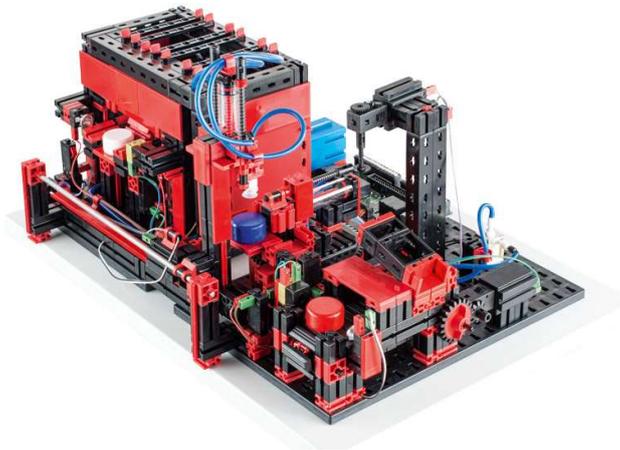


Figura 4.1. Imagen de la maqueta Fischer Technik (Multi Processing Station with Oven 24V). Recuperado de <https://www.fischertechnik.de/>

Como se puede observar en la figura, y como su propio nombre indica, dicha maqueta incorpora múltiples procesos los cuales se encargarán de simular el funcionamiento de la implementación final del sistema. Para poder explicar correctamente el funcionamiento, y con el objetivo de que en apartados posteriores se pueda referenciar con claridad a ciertos sectores de la estación, se ha representado toda la información necesaria en la *Tabla 4.1*, la cual muestra la localización, identificación, y un breve resumen del funcionamiento de los distintos sectores que se han elegido para desarrollar el funcionamiento del sistema y que han sido útiles para simplificar el proceso posterior de diseño del automatismo.

Nota: En la memoria aparecerán continuamente referencias a las variables del sistema, todas estas variables aparecen expuestas y descritas en el *Anexo I*, por lo tanto, se omitirá cualquier explicación adicional sobre éstas ya que se podrán entender en su totalidad todos los conceptos complementando ambos documentos, *Memoria* y *Anexo I*.

La estación multiproceso incorpora un gran número de actuadores y sensores, por lo tanto, se ha elegido el autómata *Siemens S7-1200* para la automatización de esta maqueta, ya que este incorpora un módulo de ampliación de entradas y salidas y podremos acceder a todas las variables de la estación multiproceso.

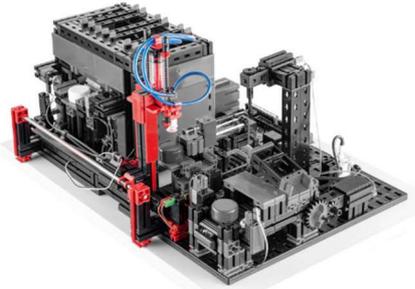
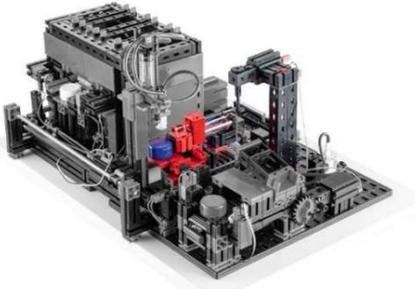
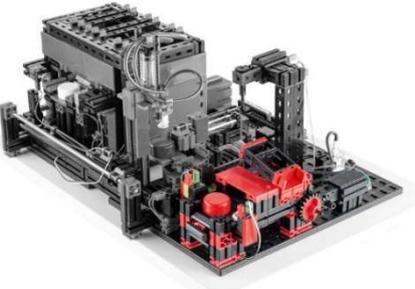
Localización	Identificación	Funcionamiento
	<p align="center"><i>Alimentador</i></p>	<p>Es el encargado de recibir las piezas a procesar.</p> <p>Una vez recibidas, las introducirá al horno y cuando éste termine, las extraerá y esperará a que el “Manipulador” las recoja.</p>
	<p align="center"><i>Manipulador</i></p>	<p>Es el encargado de transportar las piezas que han finalizado el proceso de horneado hacia la “Mesa”.</p> <p>Para ello cuenta con un sistema neumático con ventosa el cual provocará un vacío que se encargará de sostener la pieza en el recorrido.</p>
	<p align="center"><i>Mesa</i></p>	<p>Es la encargada de transportar mediante un movimiento giratorio la pieza desde la posición final del “Manipulador” hasta el inicio de la “Cinta” mediante un empujador neumático.</p> <p>También realizará el proceso mecánico de serrado.</p>
	<p align="center"><i>Cinta</i></p>	<p>Es la encargada de transportar la pieza desde la posición inicial, donde el empujador ha dejado la pieza, hasta la posición final, donde el “Brazo-ventosa” se encargará de recoger la pieza y depositarla en la correspondiente caja.</p>

Tabla 4.1. Identificación y descripción de los distintos sectores de la estación multiproceso.



4.2. Brazo-ventosa

Para la simulación del elemento encargado de introducir y extraer las piezas a la estación multiproceso, el cliente ofrece la maqueta *Fischer Technik (Vacuum Gripper Robot 24V)* mostrada en la *Figura 4.2*, al cual se le identificó como: brazo-ventosa.

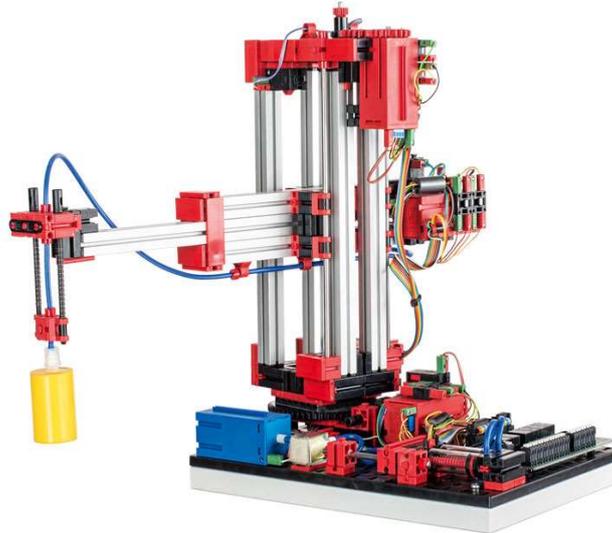


Figura 4.2. Fischer Technik (Vacuum Gripper Robot 24V). Recuperado de <https://www.fischertechnik.de/>

El brazo-ventosa incorpora tres motores con encóder, los cuales lo dotan de capacidad de movimiento e identificación de la posición en las tres direcciones (vertical, horizontal y giro). Por lo tanto, el brazo-ventosa, consta de tres grados de libertad, ya que, indicando el número de pulsos en cada movimiento, se puede acudir a una coordenada en el espacio.

Como ya se ha descrito en anteriores apartados, su funcionamiento consiste en extraer las piezas de las cajas de entrada para introducirlas en el alimentador, o extraer las piezas de la cinta para introducirlas en las cajas de salida.

Anteriormente se asignó el autómata *Siemens S7-1200* para ser el encargado de la automatización de la estación multiproceso. En este caso, asignamos el autómata *Schneider M241* ya que ofrece una programación más sencilla para la gestión de pulsos como se verá en el *apartado 5.2.2*.



4.3. HMI (Human Machine Interface)

Como su nombre indica, una HMI (Human Interfaz Machine), proporciona una interfaz, en este caso gráfica, con la cual poder interactuar con nuestro sistema. Como ya se ha visto en el *capítulo 2*, en él aparecen las características que impuso el cliente en la definición de la solución, las cuales se han analizado para dar lugar al árbol funcional de la *Figura 4.3*, en el cual aparecen las funcionalidades que incorporara la HMI, incluyendo también en éste principios de solución adoptados.

En el documento adjunto, “Manual de usuario” se incorpora un manual detallado para comprender el funcionamiento de la HMI en su totalidad.

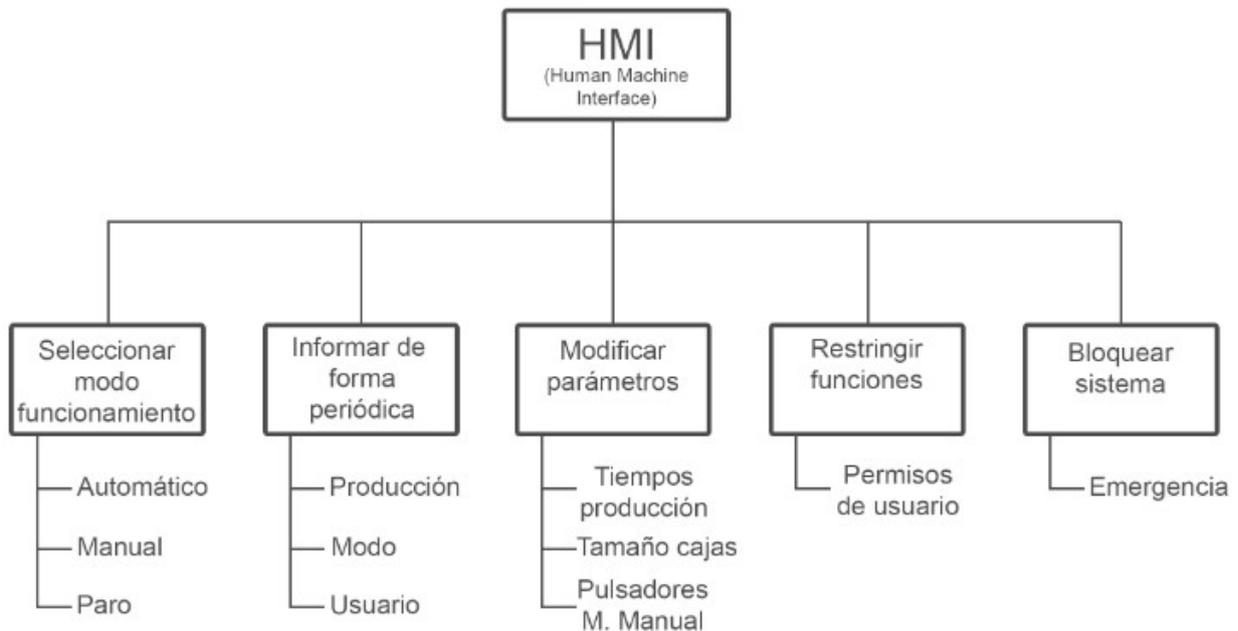


Figura 4.3. Árbol funcional generado a partir de las peticiones del cliente.

4.4. Distribución en planta

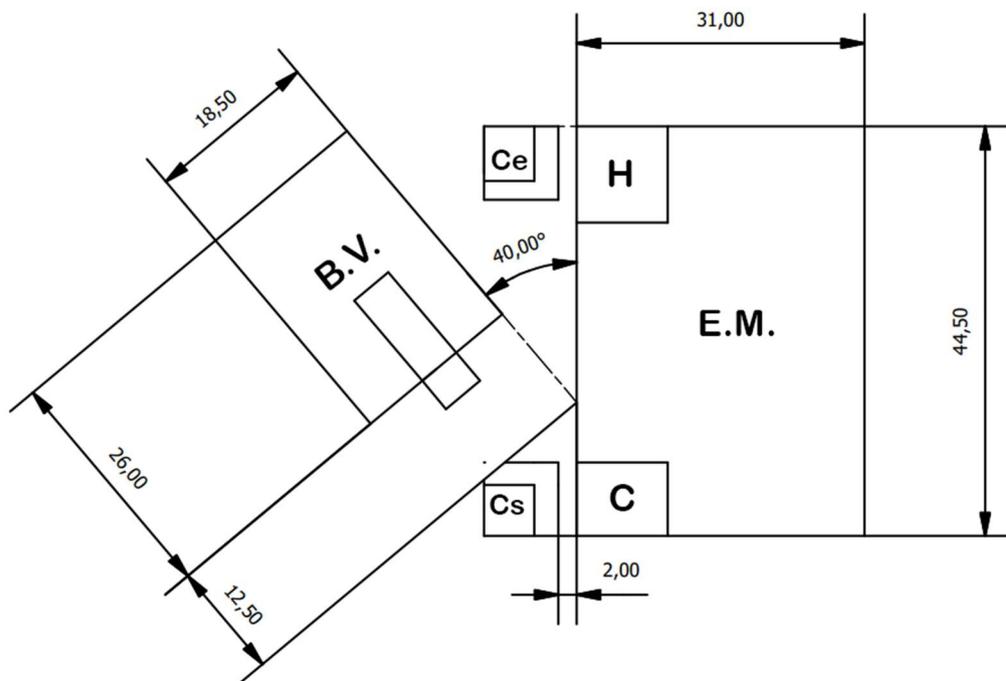


Figura 4.4. Plano de distribución en planta de los elementos que conforman el sistema

Como se ha introducido en el apartado 4.2, el Brazo-Ventosa incorpora tres encóders que emitirán unos pulsos que nos servirán posteriormente para indicarle al brazo-ventosa dónde tiene que ir.

Se ha realizado un estudio para colocar los elementos del sistema con la finalidad de, por un lado, ser accesibles por el brazo ventosa y que no haya posiciones a las que no se puedan acceder físicamente, y por otro lado, si se sigue la distribución expuesta, los pulsos insertados en la programación se cumplirán en el proceso real, ya que, como se ha comentado, la maqueta es una escala reducida del proceso real del cliente, por lo tanto bastaría con multiplicar por el factor de escala las distancias que separan los elementos para colocarlos en la situación correcta.

La distribución elegida se muestra en la Figura 4.4 en forma de plano donde hay que aclarar que:

- Las medidas del plano son en cm.
- Los identificadores que aparecen son:
 1. **E.M.:** Estación Multiproceso
 2. **B.V.:** Brazo Ventosa
 3. **Ce:** Caja de entrada
 4. **Cs:** Caja de salida
 5. **H:** Sector Horno
 6. **C:** Sector Cinta



5. DESARROLLO DE LA SOLUCIÓN

5.1. Diseño del sistema automático

Para la solución del diseño, se escogido la herramienta *GRAF CET* (*Grphe Fonctionnel de Commande Etape Transition*), esta herramienta es una de las más potentes en el campo del desarrollo de sistemas automáticos, ya que, además de poder diseñar el funcionamiento secuencial de un sistema automático, se pueden emplear herramientas para establecer relaciones jerárquicas entre varios *GRAF CET*'s que serán útiles para incorporar sistemas de seguridad y modos de funcionamiento como se verá a continuación. Todos los *GRAF CET* representados cumplen la norma (UNE-EN 60848:2013).

Es necesario añadir, que a pesar de tener un *GRAF CET* correctamente definido, posteriormente se ha tenido que implementar en los correspondientes autómatas como se verá en el *apartado 5.2*.

Los *GRAF CET*'s diseñados se encuentran adjuntos en el documento *Planos*, en la *Figura 5.1* se muestra un pequeño esquema en el cual aparece los *GRAF CET*'S parciales generados y la jerarquía utilizada para el sistema en conjunto.

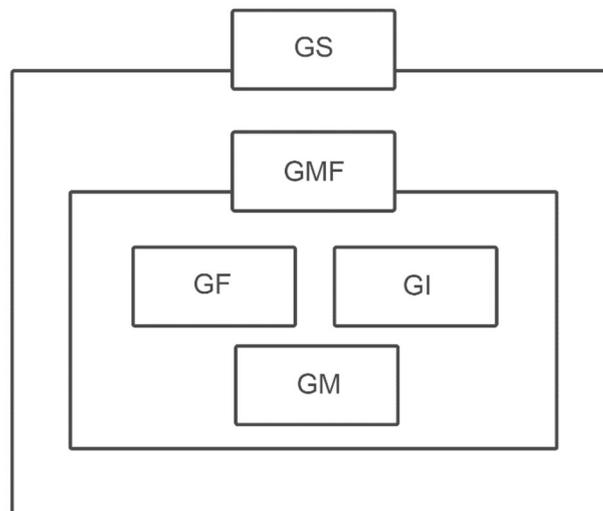


Figura 5.1. Esquema jerárquico de los *GRAF CET*'s parciales del sistema

Como se puede observar en el plano, cada *GRAF CET* parcial a su vez contiene unos *GRAF CET*'s conexos, en la *Tabla 5.1*, se resume el esquema anterior con un nivel superior de detalle ya que se incorporan los *GRAF CET*'s conexos y se asignan niveles de seguridad a los distintos *GRAF CET*'s parciales.



GRAF CET Parcial	GRAF CET Conexos	Nivel Jerárquico
GS (Seguridad)	Seguridad	0
GMF (Modos de funcionamiento)	Modos de funcionamiento	1
GF (Funcionamiento)	Funcionamiento Alimentador Funcionamiento Manipulador Funcionamiento Mesa Funcionamiento Cinta Funcionamiento Brazo Ventosa Funcionamiento Asignador Coordenadas	2
GI (Iniciación)	Iniciación Alimentador Iniciación Manipulador Iniciación Mesa Iniciación Brazo Ventosa	2
GM (Manual)	Modo Manual	2

Table 5.1. Nivel jerárquico y GRAFCET conexos de los distintos GRAFCET parciales

Para no alargar y hacer demasiado extensa la memoria, se va a explicar con claridad y detalle los GRAFCET's parciales de Seguridad y Modos de funcionamiento, sin embargo, los GRAFCET's de Iniciación y Funcionamiento serán nombrados y explicados con menos profundidad, exceptuando los GRAFCET's conexos de Brazo-ventosa y Asignador ya que el nivel de complejidad es superior a los anteriores.

Nota: En el GRAFCET aparecen los subíndices "e" y "b" y se utilizan para distinguir si la variable proviene de la Estación multiproceso (e) o el Brazo-ventosa (b), en el Anexo I no aparece expuesto de esta manera ya que como se ha visto, la programación de cada elemento se produce con autómatas diferentes.



5.1.1. Seguridad

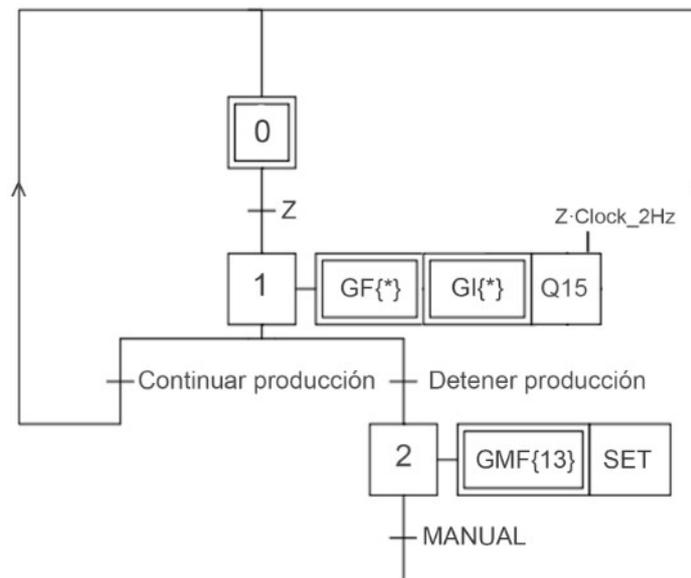


Figura 5.2. GRAFCET parcial de Seguridad (GS)

En la *Figura 5.2* se puede observar el *GRAFCET* parcial de Seguridad, es el más importante ya que su objetivo es el de bloquear el sistema en el caso de que “Z” se active.

Se ha denominado “Z” a la variable de Emergencia, la cual depende de tres variables: *Z_Seta*, *Z_Cable* y *Z_Puerta*, mediante la siguiente condición:

$$Z = Z_{Seta} \text{ OR } Z_{Cable} \text{ OR } Z_{puerta}$$

Como se puede observar en el *GRAFCET* la “etapa 0” será activada al arrancar el sistema, y solamente cambiará de estado cuando “Z” se active. Se interpreta claramente que entre las acciones de la “etapa 1” se encuentran dos forzados, uno que se encarga de congelar el funcionamiento automático del sistema, y el otro que se encarga de congelar la etapa de inicialización, ya que, aunque esta última se encuentra activada solamente al principio del modo automático, y es un periodo corto en comparación comparado con el funcionamiento normal, no por ello se debe descartar su bloqueo.

Por otro lado, también aparece una acción encargada de activar una señal luminosa alternante, en este caso con una alternancia de 2 Hz y que sólo estará activada cuando “Z” se encuentre activa.

Es muy importante aclarar que como bien muestra el *GRAFCET*, la “etapa 1” no finaliza cuando se desactiva “Z”, esto es debido a que se le ha incorporado al sistema la posibilidad de elegir si se quiere continuar la producción o se quiere detener la producción.



Por un lado, si se seleccionase “Continuar producción” se volvería a activar la “etapa 0” y por lo tanto el sistema seguiría funcionando con normalidad porque ya no se estaría congelando su funcionamiento.

Si por el contrario se eligiese “Detener producción”, se activaría la “etapa 2” que lleva ligada dos acciones:

- Un forzado que activará la “etapa 13” del *GRAFSET Modos de Funcionamiento*, que si se observa el plano que aparece en el documento *Planos*, se interpreta que esta etapa encapsula al *GRAFSET* parcial “GM”, por lo tanto, el sistema entrará en modo manual.
- Una activación de la variable “SET”, que activará todas las variables, denominadas variables de seguridad, que pudiesen poner en peligro elementos materiales y provocar daños incluso humanos, por ejemplo: ventosas que tengan piezas en ese momento y por lo tanto éstas se desprendan, o la puerta del horno que se pueda cerrar estando el alimentador debajo.

5.1.2. Modos de funcionamiento

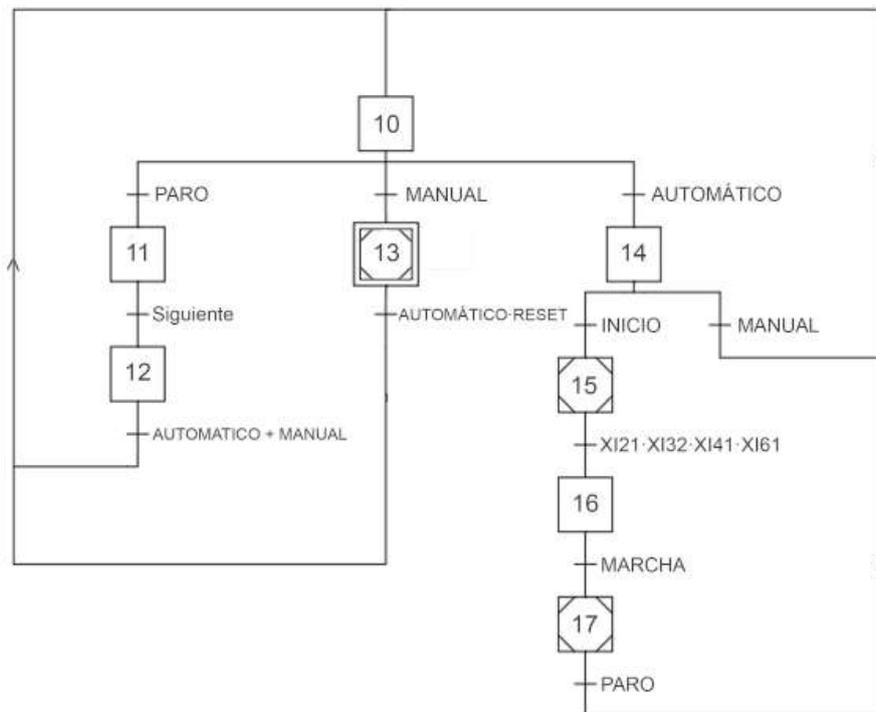


Figura 5.3. GRAFCET parcial de Modos de funcionamiento (GMF)



En la *Figura 5.3* se puede observar el *GRAFSET Modos de funcionamiento*, es el encargado de gestionar en todo momento si el sistema se encuentra en el modo manual, automático, o parado y pendiente de una solicitud.

Como se puede apreciar en la figura no aparece ninguna acción ligada a las etapas, esto es debido a que las propias etapas se utilizan para identificar en qué situación se encuentra el sistema, esta información será utilizada por los distintos dispositivos para saber qué funciones realizar como veremos en apartados posteriores.

En cuanto al funcionamiento, la “etapa 13” será activada al iniciarse el sistema a pesar de que la “etapa 10” pueda parecer más importante ya que en ella divergen tres ramas las cuales definen los tres modos de funcionamiento. Esto es debido a que en caso de que se hubiera producido una emergencia y las variables expuestas en el apartado de seguridad aún siguiesen activadas debido a la activación de la variable “SET”, el sistema arrancararía en la “etapa 13”, y no se podrá avanzar hasta que no se modifiquen las variables de seguridad (condición “RESET”).

Una vez explicado por qué se inicia en la “etapa 13” en vez de la “etapa 10”, se procede a explicar las tres ramas que divergen de la etapa 10:

- 1ª Rama (Sistema Parado): Viene activada por la variable “PARO”, esta variable se activa cuando el funcionamiento automático se ha detenido por completo, en el *apartado 5.2.2*.
- 2ª Rama (Modo Manual): Viene activada por la variable “MANUAL” la cual se activa mediante la HMI, y también cuando en el *GRAFSET Emergencia* se activa la “etapa 2”, como se ha visto en el apartado anterior.
Como se ha comentado anteriormente, se puede apreciar que la “etapa 13” encapsula al *GRAFSET* parcial “GM” y, por lo tanto, cuando nos encontremos en esta etapa se podrá manipular cualquier acción siempre y cuando las condiciones ligadas a éstas lo permitan.
Como se ha explicado anteriormente, no se podrá desactivar la etapa que encapsula el modo manual “etapa 13” hasta que no se active la variable “RESET” que indica que se han desactivado las variables de seguridad.
- 3ª Rama (Modo Automático): Viene activada por la variable “AUTOMATICO” la cual será activada desde la HMI.

Una vez activada la “etapa 14”, se proporciona la opción de volver al modo manual, pero en el caso de activar la variable “INICIO” se activará la “etapa 15” la cual encapsula al *GRAFSET Inicialización* y el sistema comenzará a inicializarse. Una vez inicializado, se activará la “etapa 16” la cual se utilizará para indicar al operario que se puede empezar la automatización, ésta se llevará a cabo mediante la activación del *GRAFSET Funcionamiento* el cual viene encapsulado por la “etapa 17” tras activarse la variable “MARCHA”, y por último, finalizará el proceso de automatización cuando se le ordene parar mediante la HMI y se hayan cumplido las condiciones que se han comentado con anterioridad en la *1ª Rama (Sistema Parado)*. Como se puede observar, el sistema está perfectamente conectado, ya que al acabar el modo automático se accede a la rama de sistema parado, y al finalizar esta, se le da al usuario la opción de elegir si quiere volver a entrar al modo automático o quiere iniciar un modo manual.



5.1.3. Inicialización

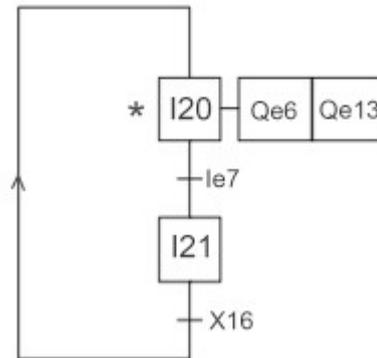


Figura 5.4. GRAFCET conexo Inicialización Alimentador

En la Figura 5.4 se muestra el GRAFCET Inicialización Alimentador, que junto con los otros tres GRAFCET's conexos de inicialización forman el GRAFCET parcial "G1", el funcionamiento de los distintos GRAFCET's conexos es idéntico, cuando la "etapa 15" se active, se activarán las etapas iniciales y comenzará el proceso de inicialización.

La existencia de un proceso de inicialización es esencial, ya que, antes de la automatización se parte de un estado inicial idóneo para que no aparezca ningún problema durante el funcionamiento.

En la Tabla 5.2 aparece la finalidad del conjunto de acciones que ejecuta cada GRAFCET:

GRAFCET Conexo	Finalidad
Inicialización Alimentador	Subirá la puerta mientras activa el motor que trasladará el alimentador hacia fuera
Inicialización Manipulador	Conducirá al manipulador a una posición cercana al alimentador configurable mediante TRI (Tiempo Retorno Inicialización)
Inicialización Mesa	Activará el giro antihorario para conducir a la mesa a la posición en la que el manipulador le entregará la pieza
Inicialización Brazo Ventosa	Activará todos los motores en sentido de recogida para situar al brazo-ventosa en una posición conocida (0,0,0)

Tabla 5.2. Finalidad de los distintos GRAFCET conexos de "G1"



5.1.4. Funcionamiento

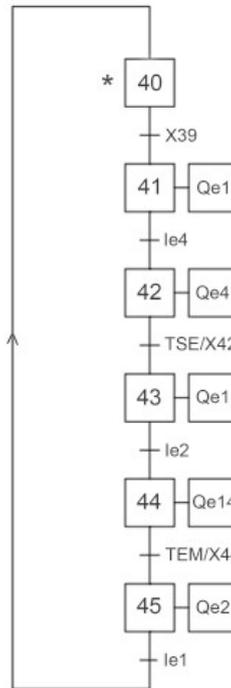


Figura 5.5. GRAFCET conexo Funcionamiento Mesa

Como se explicó anteriormente, el *GRAFCET Funcionamiento* contiene los *GRAFCET's* conexos encargados del funcionamiento de los distintos elementos del sistema.

En la *Figura 5.5* aparece el *GRAFCET* conexo en el cual se plasma la lógica a seguir por el sector “Mesa”. El mecanismo de activación es idéntico al del modo de inicialización, pero en este caso es la “etapa “17” la que encapsula los *GRAFCET's* conexos

Como ya se ha nombrado con anterioridad, en el plano que se encuentra en el documento *Planos*, se expone y se describe el funcionamiento global de los *GRAFCET's* diseñados, por lo tanto, para reducir la memoria, se ha decidido no exponer todos los *GRAFCET's* conexos de “GF”.

Sin embargo, como se ha comentado anteriormente, es necesario explicar con un nivel superior de detalle el *GRAFCET Brazo Ventosa* y el *GRAFCET Asignador*.

En la *Figura 5.6* aparece el *GRAFCET* que dicta el funcionamiento del brazo-ventosa, como se verá a continuación, para el funcionamiento correcto del brazo es imprescindible la existencia del *GRAFCET Asignador de coordenadas*, ya que el presente *GRAFCET* solamente define el tipo de movimiento que debe de seguir, no a dónde tiene que ir. Dicho esto, el *GRAFCET* mostrado se puede descomponer en una serie de funciones por etapas que son las siguientes:

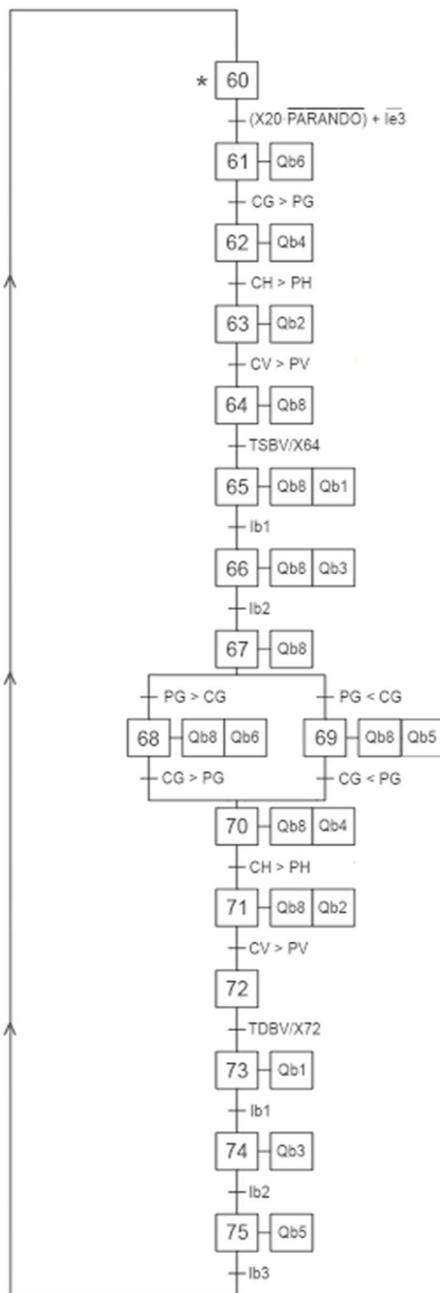


Figura 5.6. GRAFCET conexo Funcionamiento Brazo-Ventosa

Espera (etapa 60): Etapa de espera de una solicitud, el brazo-ventosa se encuentra en espera hasta que el alimentador solicita una pieza de entrada, o la cinta solicita al brazo que se puede retirar una pieza acabada.

Recogida (etapas 61-64): El brazo acude a la posición solicitada y activa la succión durante un tiempo “TSBV (Tiempo Succión Brazo Ventosa) para recoger la pieza. Las posiciones solicitadas en esta función podrían ser:

Caja Entrada: En caso de que el Alimentador solicite la pieza.

Cinta: en caso de que ésta contenga una pieza a retirar.

Entrega (etapas 65-72): El brazo acude a la segunda posición solicitada siguiendo un orden de movimientos, y desactiva la succión durante un tiempo “TDBV” para entregar la pieza. Las posiciones en esta función podrían ser:

Alimentador: En caso de que el Alimentador solicite la pieza.

Caja de Salida: En caso de haber retirado la pieza de la cinta.

Inicialización (etapas 73-75): El brazo-ventosa realiza una inicialización para así volver a asignar los valores iniciales a los pulsos del encóder y no provocar errores de posición.

Los movimientos que realiza el brazo se producen activando los motores descritos en el apartado 4.2, dichos motores permanecen activados hasta que se cumplen las transiciones en las cuales se establece que los pulsos transmitidos por el encóder sobrepasan un límite establecido que depende de las funciones.

Como se ha comentado anteriormente los movimientos siempre son realizados en un orden determinado para garantizar que los diversos elementos móviles no colisionen. Dicho orden está ligado a la acción que se esté realizando. Por ejemplo, en las acciones que se tenga que acudir a una posición de una altura mayor a una posición de altura menor se seguirá un orden de: giro en primer lugar, horizontal en segundo lugar, y vertical como último movimiento ya que no tendría sentido realizar el giro cuando se encuentra expandida la articulación horizontal y estamos en bajas alturas. Pero, por otro lado, en las acciones que se realizan para transportar la pieza desde una posición hasta otra de la misma altura, en primer lugar, se realizará el siguiente orden: vertical en primer lugar, horizontal en segundo lugar, y

giro en último lugar. Y una vez haya recogido su posición, se seguirá el orden producido de una altura superior a una inferior de nuevo.

Una vez descrito el funcionamiento del brazo-ventosa, se procede a explicar el *GRAFSET Asignador de Coordenadas* que proporcionará la posición a la que deberá ir el brazo en todo momento. Dicho *GRAFSET* se muestra en la *Figura 5.7*.

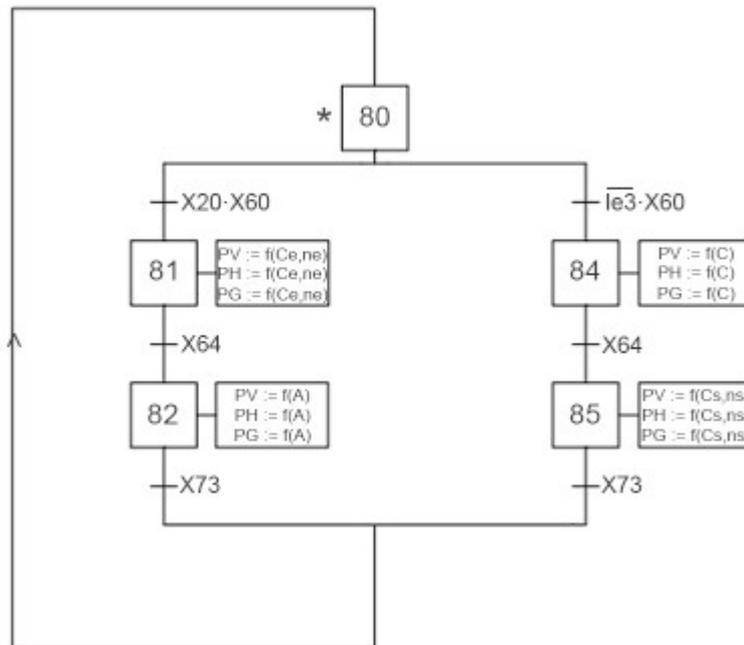


Figura 5.7. GRAFCET conexo Funcionamiento Asignador de Coordenadas

Como se ha explicado anteriormente en el *GRAFSET* que definía el funcionamiento del brazo-ventosa, las posiciones deben variar en función de qué sector lo llamaba, pero a su vez, las posiciones tendrán que variar en función de qué pieza se está procesando y qué tamaño de caja se tiene seleccionado.

El funcionamiento consiste en la espera en la etapa 80 hasta que ocurre una de las dos condiciones de la divergencia generada.

- **1ª Rama:** Si el alimentador solicita una pieza ("X20") y el brazo-ventosa se encuentra en espera ("X60"), se activa la "etapa 81", y automáticamente se asignan los pulsos para que el brazo-ventosa acuda a la posición correcta en su función de recogida. Una vez acabada la función de recogida ("X64"), se activa la "etapa 82" y se asignan unos nuevos pulsos para la función entrega para que una vez finalizada esta función ("X73") se vuelva a la etapa de espera.



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.



- 2º Rama: Si la cinta solicita la recogida de una pieza (NOT "le") y el brazo-ventosa se encuentra en espera, se realizarán de nuevo unas asignaciones de pulsos, pero en este caso, teniendo en cuenta que las posiciones serán las de la cinta y caja de salida.

En el *Anexo I*, se muestran los pulsos obtenidos para las distintas posiciones.



5.2. Implementación del sistema automático

Una vez diseñado el sistema, el siguiente paso será implementarlo, en el presente apartado se va a proceder a explicar este proceso teniendo en cuenta que la metodología a emplear será distinta para cada autómata, ya que, como se ha visto cada uno utiliza un software distinto y se podrán utilizar unos u otros lenguajes de programación.

5.2.1. Implementación en Siemens S7-1200

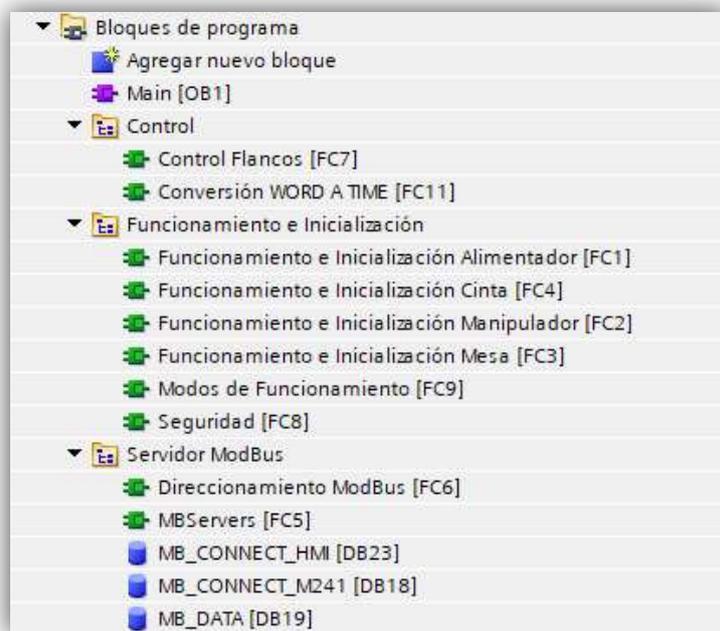


Figura 5.8. Estructura del programa generado en Tia Portal V13

En la *Figura 5.8* se representa la estructura programa en el software *Tia Portal V13*, como se puede apreciar, existen tres grupos diferenciados:

- Control: Agrupa los programas utilizados para realizar acciones secundarias que son necesarias para que el sistema funcione correctamente, estos son:
 - *Control Flancos*: Como se verá a continuación, esta función detectará los flancos de subida de las etapas encapsulantes.



- *Conversión WORD a TIME:* Se encargará de convertir las palabras (*WORD*) transferidas por la HMI y convertirlas a variables de tipo tiempo (*TIME*) para poder asignarlas a los retardos.
- Funcionamiento e Inicialización: Agrupa la implementación de los *GRAF CET's* de funcionamiento e inicialización de los sectores representados en la *Tabla 4.1*, así como la representación de los *GRAF CET's* de mayor nivel jerárquico: Seguridad y Modos de Funcionamiento.
- Servidor ModBus: Se encargará de gestionar las comunicaciones ModBus con los distintos dispositivos del proyecto, dicha estructura se explicará detalladamente en el *apartado 5.4*.

Para implementar el sistema automático en el autómata *Siemens S7-1200*, es necesario obtener las ecuaciones lógicas a partir de los *GRAF CET's*, ya que no se dispone de la posibilidad de programar en SFC lo cual sería lo más lógico si se parte de un diseño mediante *GRAF CET*. Una vez obtenidas las ecuaciones, éstas serán fáciles de implementar en lenguaje *KOP*.

Se va a desarrollar un pequeño ejemplo con el cual se entenderá la dinámica general de esta transformación, para este ejemplo se ha decidido generar las ecuaciones lógicas del *GRAF CET Seguridad* que ya se mostró en la *Figura 5.2*.

La condición de activación S_n de una etapa X_n es la siguiente:

$$S_n = X_{n-1} \cdot T_{n-1}$$

Ecuación 5.1

Donde: X_{n-1} = Etapa anterior.

T_{n-1} = Condición de transición de la etapa anterior.

La condición para la etapa X_n esté activa es:

$$X_n = S_n + S_{n+1} \cdot X_n$$

Ecuación 5.2

Donde: S_n = Condición de activación de la etapa.

S_{n+1} = Condición de activación de la siguiente etapa.



La condición para que la salida Q_n esté activa es:

$$Q_n = c \cdot X_n$$

Ecuación 5.3

Donde: c = Condición.

X_n = Etapa donde se produce la acción.

Aplicando los conceptos anteriores basándose en el *GRAFSET Seguridad* se tienen las siguientes ecuaciones e implementación en bloque "Seguridad (FC8)" en Tia Portal V13.

Ecuaciones de activación

$$S_0 = X_2 \cdot MANUAL + X_1 \cdot Continuar producción + C_i$$

$$S_1 = X_0 \cdot Z$$

$$S_2 = X_1 \cdot Detener producción$$

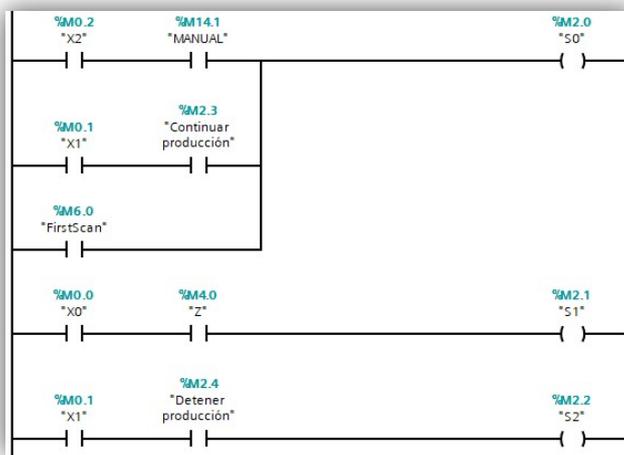


Figura 5.9. Ecuaciones de activación en lenguaje KOP implementadas en Tia Portal



Ecuaciones de estado

$$X_0 = S_0 + \bar{S}_1 \cdot X_0$$

$$X_1 = S_1 + \bar{S}_0 \cdot \bar{S}_2 \cdot X_1$$

$$X_2 = S_2 + \bar{S}_0 \cdot X_2$$

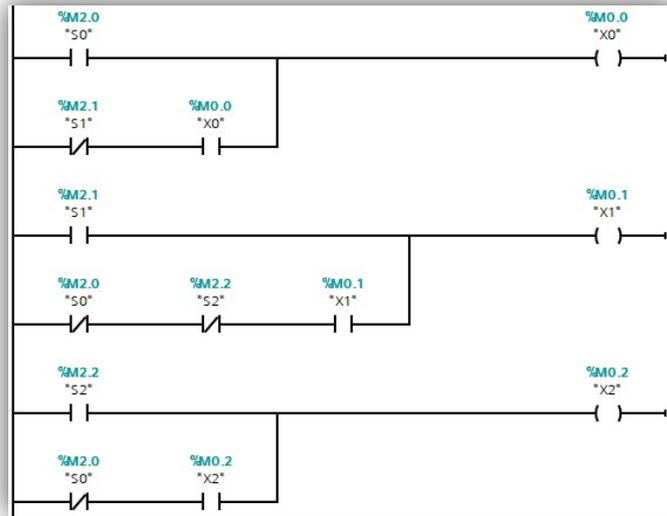


Figura 5.10. Ecuaciones de estado en lenguaje KOP implementadas en Tia Portal

Ecuaciones de salida

$$Q15 = X_1 \cdot Z \cdot Clock_2HzSET = X_2$$

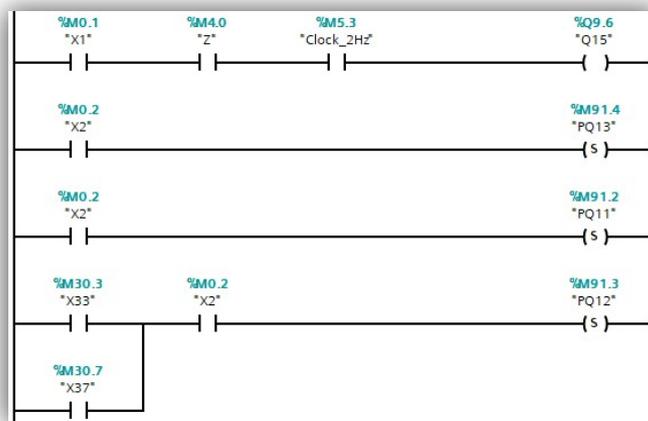


Figure 5.11. Ecuaciones de salida en lenguaje KOP implementadas en Tia Portal



Como se puede observar, es relativamente sencillo implementar el funcionamiento básico de un *GRAFSET* en lenguaje de contactos, y aunque en este caso se muestra solamente la implementación del *GRAFSET seguridad* en el bloque “Seguridad (FC8)”, los demás *GRAFSET*’s siguen la misma dinámica y se encuentran implementados en sus respectivos bloques de “Funcionamiento e Inicialización”.

A pesar de haber obtenido la implementación del funcionamiento básico, en los *GRAFSET*’s diseñados aparecen diversas funciones que se deben implementar, y que no se han explicado anteriormente, éstas son:

- Encapsulado
- Forzado
- Temporizadores.
- Conversiones *WORD* a *Time*

Encapsulado

$$X_{20} = (S_{20} + \overline{S_{21}} \cdot X_{20}) \cdot X_{17} + \uparrow X_{17}$$

La ecuación de estado anterior corresponde a la “etapa 20” del *GRAFSET Funcionamiento (GF)*, la cual ha sido la elegida para explicar la implementación del encapsulado en Tia Portal V13.

Se puede apreciar que la “etapa 20” se activará cuando haya un flanco de subida de la variable “X17” que representa la “etapa 17” del *GRAFSET Modos de funcionamiento (GMF)* la cual a su vez desactivará todas las etapas del *GRAFSET* conexas en el que se encuentra la “etapa 20” si no se encuentra activada.

Para la implementación de los encapsulados se ha creado un bloque de función en Tia Portal denominado “Control Flancos [FC7]” el cuál será el encargado de detectar los flancos de subida de las etapas encapsulantes.

Dicho bloque, mostrado en la *Figura 5.12*, lleva incorporado las funciones *R_TRIG* que nos proporciona Tia Portal para detectar flancos, dicha función incorpora una entrada (CLK), en la cual se tiene que añadir la variable que queremos analizar, y la salida (Q), la cual se activará cuando se produzca un flanco de subida.

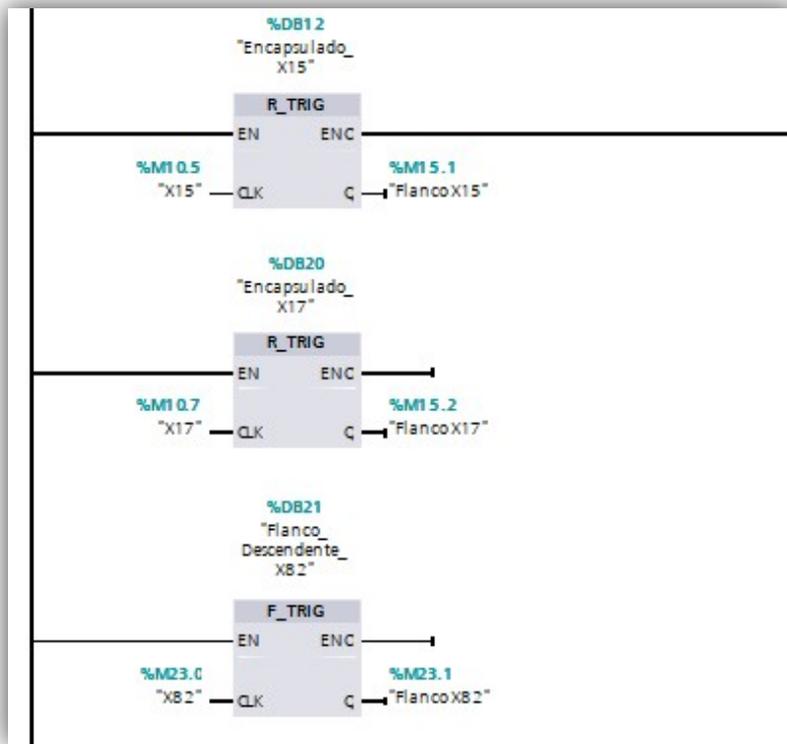


Figura 5.12. Bloque de función "Control Flancos [FC7]"

Por lo tanto, una vez que el bloque "Control de flancos [FC7]" nos proporciona los flancos de las variables la implementación de las ecuaciones de estado de las etapas encapsuladas en lenguaje KOP son relativamente sencillas de generar como muestra la Figura 5.13.

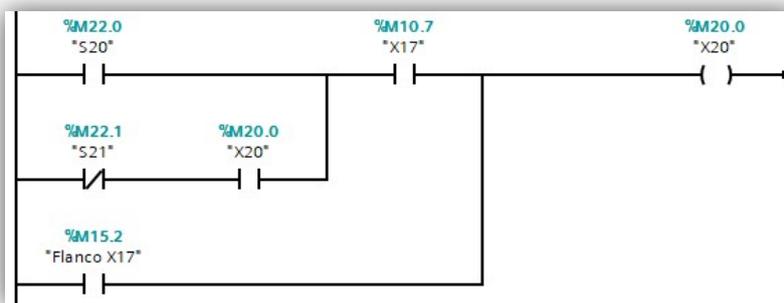


Figura 5.13. Ecuación de estado de la etapa 20 en lenguaje KOP implementada en el bloque "Funcionamiento e Inicialización Alimentador [FC1]"



Forzado

En los GRAFCET's diseñados aparecen dos tipos de forzado, uno que paralizará a otro GRAFCET parcial, y otro que impondrá la activación de una etapa en otro GRAFCET parcial.

Para resolver el forzado que "congela", simplemente se añade una condición a las salidas que hacen que evolucione el GRAFCET a forzar, para que se produzcan siempre y cuando no nos encontremos en la etapa que se produce el forzado, por lo tanto, si no se acciona ninguna de estas salidas el GRAFCET no evolucionará debido a que no se activarán las transiciones.

En el párrafo anterior se ha descrito que la condición se tenía que añadir en las acciones que hiciesen evolucionar el GRAFCET, esta aclaración es necesaria, ya que, si se añadiese la congelación a todas las acciones, existirían algunas como por ejemplo la succión en las ventosas, las cuales no hacen evolucionar el sistema y si no se produjesen se caerían las piezas si estas estuvieran en dichos elementos en el momento del forzado.

En la Figura 5.14 se puede observar un ejemplo gráfico del tipo de forzado explicado hasta ahora y que muestra claramente cómo la salida "Q5" sí que incorpora la condición de no estar en la etapa 1, y sin embargo la salida "Q13" no incorpora tal condición.

Por otro lado, para resolver la situación que provoca el otro tipo de forzado, simplemente se modifican las ecuaciones como se muestra a continuación para que cuando se active la etapa que produce el forzado ("etapa 2"), se active la etapa forzada ("etapa 13") y se desactiven las demás etapas a las que no va dirigida el forzado. La implementación en TIA PORTAL se muestra en la Figura 5.15.

$$X_{13} = (S_{13} + \overline{S_{10}} \cdot X_{13}) + X_2$$

$$X_{12} = (S_{12} + \overline{S_{10}} \cdot X_{12}) + \overline{X_2}$$

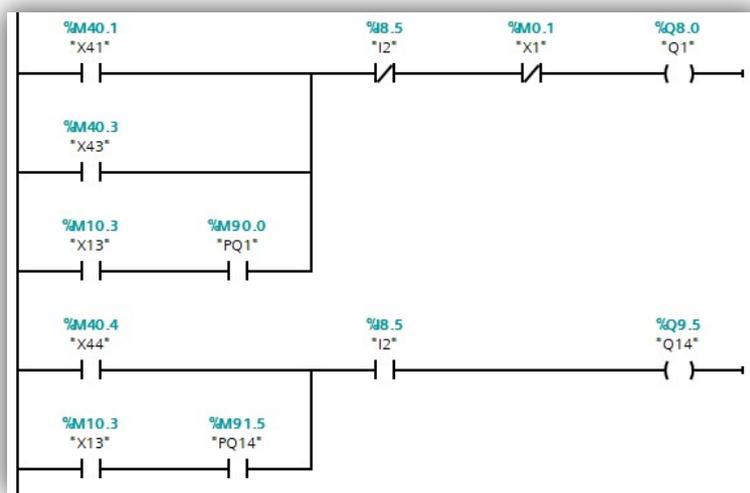


Figura 5.14. Ecuaciones de salida de las acciones "Q1" y "Q14" en lenguaje KOP implementadas en el bloque "Funcionamiento e Inicialización Mesa [FC3]"

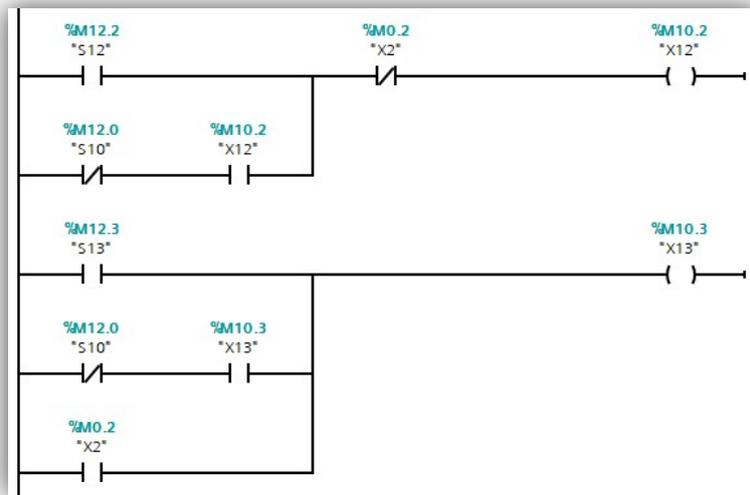


Figura 5.15. Ecuaciones de estado de la “etapa 12” y “etapa 13” en lenguaje KOP implementadas en el bloque “Modos de Funcionamiento [FC9]”

Temporizadores

Como bien se ha nombrado en apartados anteriores, se ha dotado al sistema la capacidad de incorporar una serie de temporizadores en ciertas transiciones los cuales son introducidos mediante la HMI, se va a proceder a explicar implementar dichos retardos en lenguaje KOP. Será en el siguiente punto, donde se explicará detalladamente la conversión entre la variable transferida por la HMI y el autómatas.

Por seguridad, se han añadido retardos en ciertas salidas para que algunos elementos mecánicos no tengan la posibilidad de golpear a otros, como, por ejemplo, pudiera ocurrir con el alimentador y la puerta del horno ya que ambas acciones son inmediatas y no se dispone de sensores capaces de informar que la puerta ya está completamente subida.

Se ha utilizado la función *TON* (*STEP 7, Manual del Sistema*) incorporada en *Tia Portal*, en la *Figura 5.16*, se muestra un ejemplo de retardo en la salida “Q5”, la cual se verá retardada un tiempo dado por la entrada (PT), que en este caso se le asigna la variable “Retardo Entrada”.

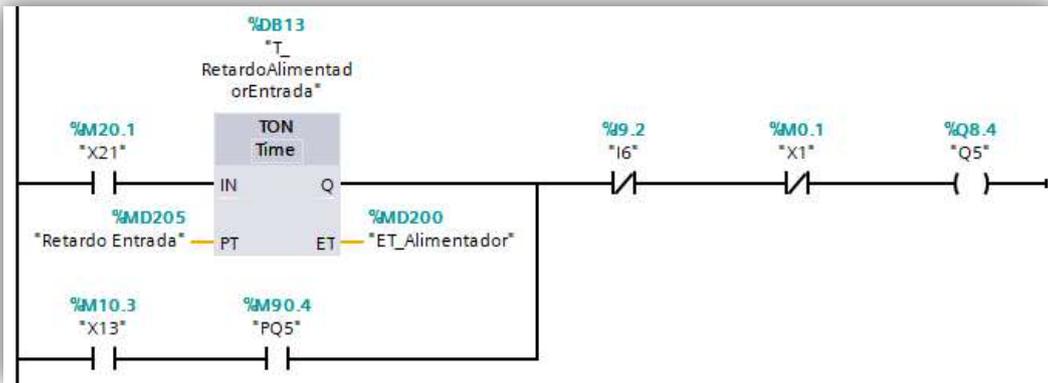


Figura 5.16. Ecuación de salida de la acción "Q5" implementada en el bloque "Funcionamiento e Inicialización Alimentador [FC1]"

Conversiones WORD a TIME

En la introducción del presente apartado, se introdujo el bloque de función "Conversión WORD a TIME [FC11]", el cual tiene un papel muy importante ya que sin este bloque no sería posible la implementación de la configuración de tiempos mediante la HMI.

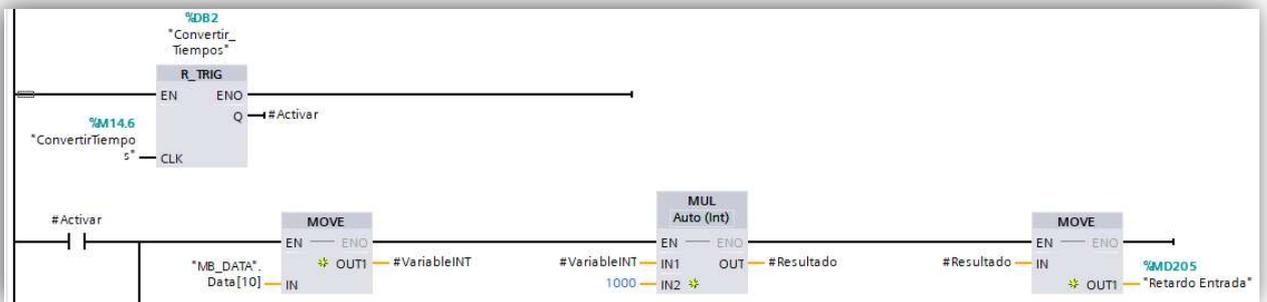


Figura 5.17. Detalle del bloque de función "Conversión WORD a TIME [FC11]"

En la Figura 5.17, se muestra un fragmento del programa implementado. El funcionamiento es el siguiente:

- 1) En primer lugar, se detecta el flanco de una variable proporcionada por la HMI, mediante el ya visto bloque R_TRIG, la cual nos indicará que se ha producido una modificación en alguna de las variables temporales.



- 2) Una vez producido el flanco, se observa que se activará la variable “activar” durante un ciclo del autómata, la cual activará un total de trece operaciones en paralelo que son idénticas a la primera mostrada en la figura, y que contienen los siguientes bloques:
- a) Bloque MOVE: Se encargará de mover la variable de tipo WORD recibida por la HMI a una variable auxiliar de tipo INT, declarada como “#VariableINT” la cual irá modificando su valor en cada rama en paralelo. *(STEP 7, Manual del Sistema)*
 - b) Bloque MUL: Se encargará de multiplicar la variable anterior por el valor entero 1000 y asignar el resultado a otra variable auxiliar de tipo INT declarada como “#Resultado”, la cual, como ocurre con “#VariableINT”, esta se irá modificando, dependiendo de en qué rama estemos para así evitar declarar trece variables auxiliares. La operación se realiza debido a que, si introdujésemos el valor directamente en la variable temporal, éste aparecería en ms. *(STEP 7, Manual del Sistema)*
 - c) Bloque MOVE: Se encargará de mover la variable “#Resultado” a la variable temporal correspondiente que será la entrada (PT) de los bloques TON como se ha visto en el apartado anterior.

Nota: Se ha incorporado el sistema de llamada mediante la variable #Activar para no emplear recursos del PLC en realizar operaciones que, aunque no sobrecargarían mucho el sistema, no es necesario calcularlas en todos los ciclos.



5.2.2. Implementación en Schneider M241

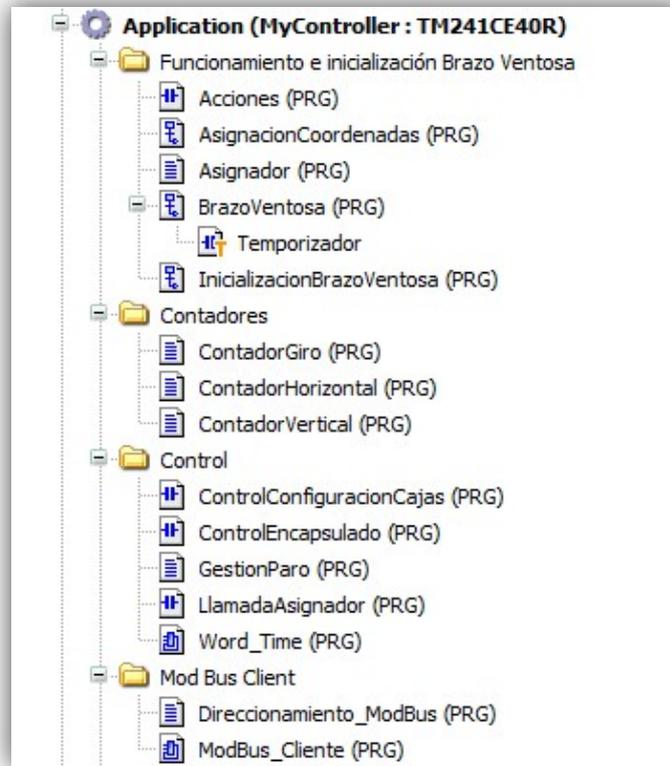


Figura 5.18. Estructura de programa generada en SoMachine

En la *Figura 5.18* se muestra la estructura del programa realizado en *SoMachine*, como se puede apreciar, es muy similar a la estructura empleada en *Tia Portal*, ya que se tiene un grupo el cual se encarga del funcionamiento y la inicialización, un grupo de control el cual se encarga de realizar acciones secundarias las cuales son necesarias para el correcto funcionamiento del programa, un grupo en el cual se insertan los elementos necesarios para la comunicación ModBus la cual se explicará en el *apartado 5.4.2*. La única diferencia, viene dada por la existencia de un cuarto grupo, “Contadores”, el cuál será el encargado de gestionar los pulsos emitidos por los encóders que lleva incorporado el brazo-ventosa.

Como se ha visto en la *Tabla 3.1* el Schneider M241 sí que admite la programación en lenguaje SFC, la cual emplearemos debido a su sencillez para implementar los *GRAFSET's* al autómatas.

Para poder explicar la aplicación del funcionamiento del proceso, antes es necesario explicar con detalle cómo se ha recogido la información de los pulsos del encóder.

Como se ha comentado anteriormente, el grupo “Contadores” es el encargado de realizar dicha gestión. Dicho grupo se compone de tres programas los cuales están programados en ST (Texto



Estructurado). El código de la gestión de pulsos del movimiento vertical se muestra en la *Figura 5.19*, y se ejecutará cada vez que se produzca un pulso.

Como se puede apreciar en el código, este asigna el valor “0” cuando el final de carrera del respectivo movimiento se activa (Condición IF), esto es así ya que se han elegido como origen de cada eje los finales de carrera de cada movimiento.

Cuando la condición descrita en el párrafo anterior no se cumpla, (Condición ELSE), aparecen de nuevo tres condiciones:

- 1º Condición IF: Si está activado el motor que hace que se desplace en dirección vertical hacia abajo suma un pulso a la cuenta vertical.
- 2º Condición IF: Si está activado el motor que hace que el brazo se desplace en la dirección vertical hacia arriba resta un pulso a la cuenta vertical.
- 3º Condición IF: Si el brazo supera o no unos ciertos pulsos, los cuales se han establecido como máximos para que el sistema funcione correctamente y no haya daños mecánicos, se active una variable la cual será la encargada de comunicar a la HMI que se ha alcanzado el máximo y no se puede avanzar más.

Las condiciones anteriores se han explicado para el movimiento vertical, los códigos que gobiernan los demás movimientos son idénticos, pero con sus respectivos movimientos y finales de carrera.

```
IF Input_I1=TRUE THEN
    CuentaVertical:=0;
ELSE
    IF Output_Q2=TRUE THEN
        CuentaVertical:=CuentaVertical+1;
    END_IF;

    IF Output_Q1=TRUE THEN
        CuentaVertical:=CuentaVertical-1;
    END_IF;
    IF CuentaVertical > MaxCuentaVertical THEN
        AvisoVertical := TRUE;
    ELSE
        AvisoVertical := FALSE;
    END_IF
END_IF;
```

Figura 5.19. Código implementado en el programa “ContadorVertical”



Como se ha explicado anteriormente, el código se tendrá que ejecutar cuando se produzca un pulso en el encóder. Para conseguir este propósito, se ha creado una tarea, mostrada en la *Figura 5.20*, la cual ejecutará el programa “Contadores” cuando la variable “I0” se ejecute.

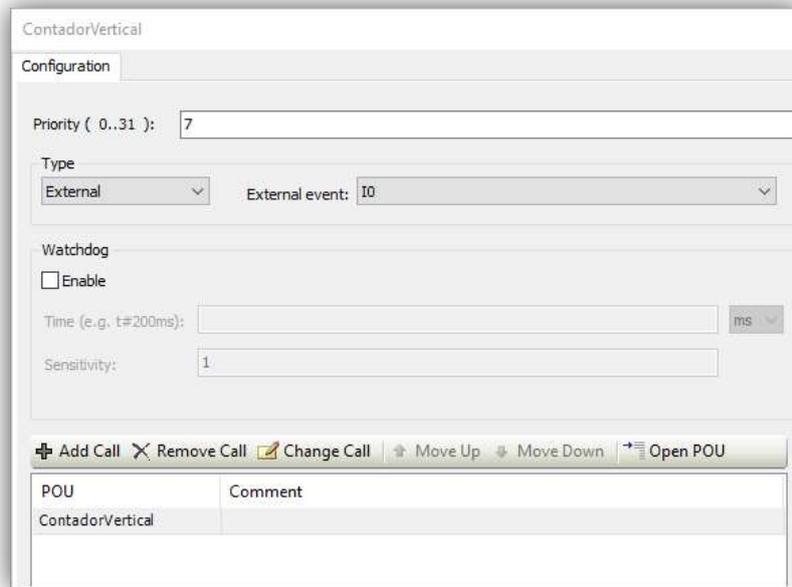


Figura 5.20. Configuración de la tarea encargada de ejecutar el programa “ContadorVertical”

Anteriormente se configuraron las variables que nos proporcionarán los pulsos de los encóders, “I0”, “I2” e “I3” como variables de evento y asignarles un filtro de muestreo de 0.002 ms para no perder ningún pulso (véase *Figura 5.21*)

Parameter	Type	Value	Default Value	Unit
Inputs Parameters				
I0				
Event	Enumeration of BYTE	Rising edge	No	
Bounce	Enumeration of BYTE	0.002	0.000	ms
I1				
Event	Enumeration of BYTE	Rising edge	No	
Bounce	Enumeration of BYTE	0.002	0.000	ms
I2				
Event	Enumeration of BYTE	Rising edge	No	
Bounce	Enumeration of BYTE	0.002	0.000	ms

Figura 5.21. Detalle de la configuración aplicada a las variables “I0”, “I1” e “I2”



Una vez explicado cómo se obtienen administran los pulsos de los encóders se puede proceder a explicar la implementación del brazo ventosa. Como se ha mostrado en apartados anteriores, el funcionamiento completo del brazo-ventosa lo componen los *GRAF CET's* conexos: "brazo-ventosa", "asignador de coordenadas" y su respectivo *GRAF CET* conexo de inicialización.

La metodología empleada para la implementación del funcionamiento básico de los *GRAF CET's* ha sido la siguiente:

Se han creado programas base en lenguaje "SFC" en los cuales se han introducido la información que nos proporcionaban los *GRAF CET's*, denominando a una "etapa X" de un *GRAF CET*, "EX" e introduciendo las ecuaciones lógicas de las transiciones. En la *Figura 5.22* aparece un fragmento del programa "BrazoVentosa", en concreto las funciones de espera (E60) y recogida (E61-E64).

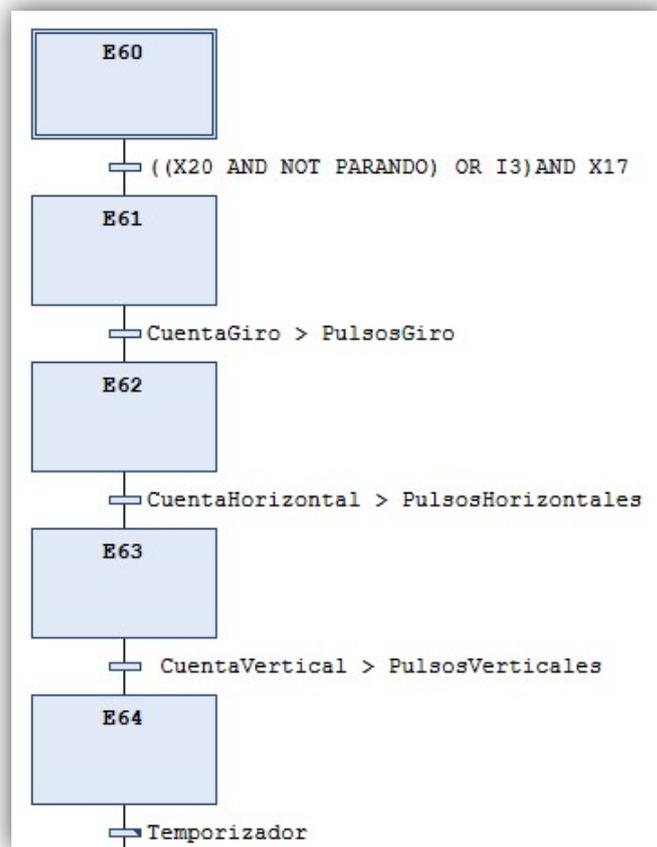


Figura 5.22. Detalle del programa "BrazoVentosa" implementado en SoMachine

Una vez creados los programas base, se ha creado un programa en lenguaje LD en el cual aparecen las ecuaciones lógicas de activación de las salidas. Esto es posible debido a que SoMachine nos permite



conocer cuándo una etapa está activa, para conocer dicha información se debe escribir la siguiente ruta:

$$\text{Etapa } X = \text{"Nombre del POU (SFC)}. \text{"Nombre de la etapa"}.x$$

En la *Figura 5.23* aparece un fragmento del POU “Acciones” el cual muestra las ecuaciones de activación de la salida “Output_Q1”. (Como se puede apreciar, se utiliza la nomenclatura explicada anteriormente para referirse a las distintas etapas que deben activar dicha salida.)

Se puede observar que como ya ocurría en la implementación en el Siemens S7-1200, se incorporan de nuevo condiciones adicionales que en un funcionamiento teóricamente perfecto no deberían aparecer, pero sin embargo es necesario añadir en caso de seguridad para que no haya accidentes. En la figura por ejemplo se muestra que la salida “Output_Q1” (motor movimiento vertical hacia arriba) sólo se activará en caso de que la entrada “Input_I1” (final de carrera) no se encuentre activa.

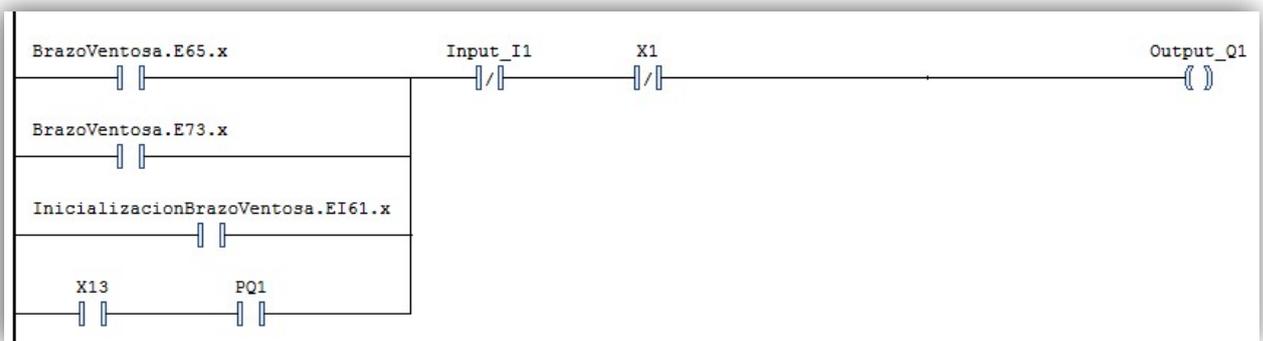


Figura 5.23. Ecuación de la salida “Output_Q1” implementada en el programa “Acciones”

Una vez descrito el funcionamiento básico de funcionamiento, es necesario explicar cómo se han implementado funciones más complejas, como ya se hizo en la implementación del Siemens S7-1200, las cuales vuelven a ser:

- Asignación de coordenadas y gestión de paro.
- Encapsulado.
- Forzado.
- Temporizadores.
- Conversiones WORD a TIME.



Asignación de coordenadas y gestión de paro

Como se vio en el apartado 5.1.4₂ el GRAFCET Asignador de coordenadas es el encargado de indicarle al brazo ventosa a qué posiciones deberá ir en cualquier momento.

También se explicó que el sistema debería incorporar un sistema de paro el cual detendría el proceso cuando la caja de salida se llenase.

La implementación de dichos requisitos se ha obtenido gracias a dos programas que aparecen representados en la Figura 5.24 y que son los siguientes:

- Asignador: Se trata de un programa en lenguaje (ST) que depende íntegramente del programa “AsignacionCoordenadas”, el cual es simplemente una implementación sencilla del GRAFCET “Asignador de coordenadas” que se consigue siguiendo la metodología explicada anteriormente.

Consta de una serie de condiciones (IF), las cuales se activarán cuando las etapas correspondientes del programa “AsignaciónCoordenadas” se activen. En la figura se observa claramente como las dos primeras condiciones asignan los pulsos del alimentador y de la cinta, que como se vio en el diseño, no dependen ni de la configuración de cajas, ni del número de pieza a procesar.

En cuanto a la última condición, esta muestra una complejidad superior ya que se encarga de asignar los pulsos dependiendo de la configuración de caja seleccionada y el número de pieza a procesar, y de también reiniciar la cuenta de piezas y sumar una caja producida o consumida cuando se alcance el valor máximo de cada caja. Como se puede apreciar en la figura, los pulsos se encuentran almacenados en matrices las cuales se exponen con mayor claridad en el Anexo I.

- GestiónParo: Programa en lenguaje ST de corta extensión que se encargará de gestionar la parada del sistema, el funcionamiento es el siguiente:

Cuando desde la HMI se ordene parar el sistema se activará la variable “PARAR”, una vez activada, el autómata comprobará en cada ciclo si se cumple la siguiente condición:

$$diferencia + (nps - 1) = MaxCaja$$

Donde: $diferencia = npe - nps$

$MaxCaja = \text{Máximo de piezas a introducir en una caja}$

La condición anterior indica que el número de piezas que falta por introducir en la caja de salida es igual a el número de piezas que se encuentran actualmente procesándose.



Cuando se cumpla dicha condición, la variable "PARANDO" se activará, y el brazo-ventosa dejará de introducir piezas (véase de nuevo la Figura 5.22) y será entonces, cuando en el programa asignador se llene la caja, cuando se active la variable "PARO", que será la variable que indicará que el sistema se ha parado por completo.

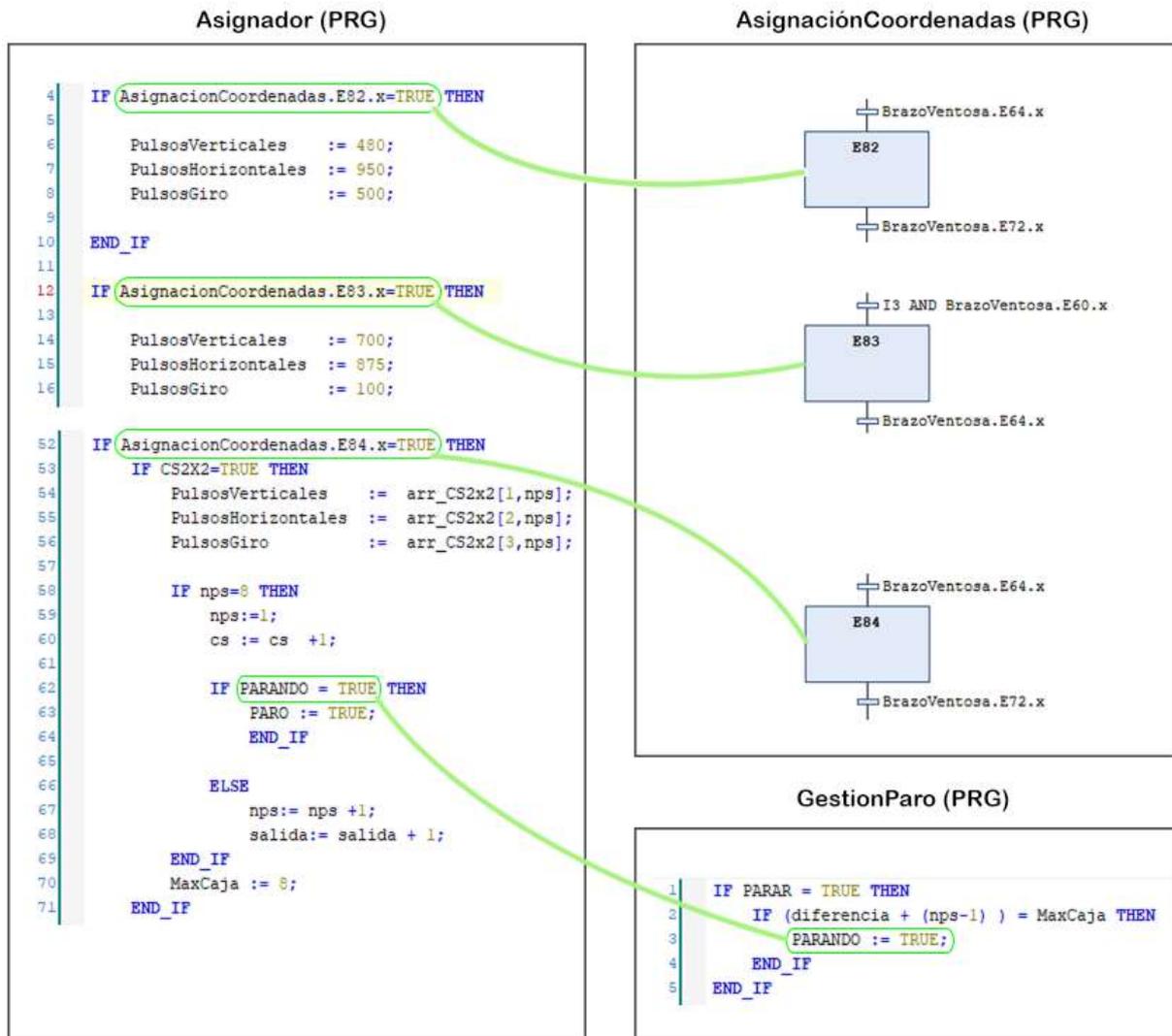


Figura 5.24. Esquema de interacción entre los programas "Asignador", "AsignaciónCoordenadas" y "GestionParo"



Encapsulado

Como se puede apreciar en el plano contenido en el documento *Planos*, las etapas de los *GRAF CET*'s encapsulados que se ejecutarán cuando la etapa del *GRAF CET* encapsulante se active, son las etapas iniciales. Se ha optado por utilizar la variable “*SFCInit*” ofrecida SoMachine para que, una vez activada, la etapa inicial de un programa en lenguaje SFC se active.

Para utilizar la variable “*SFCInit*” ha sido necesario activarla previamente en el submenú “Configuración del Proyecto” en la pestaña SFC, y una vez activada, se declaró como variable en el GVL.

Una vez declarada, se accede a la variable “*SFCInit*” de un programa en SFC escribiendo:

Nombre del programa.SFCInit

En la *Figura 5.25* se muestra el programa “InicializaciónBrazoVentosa”, el cual está programado en lenguaje SFC y representa la implementación del *GRAF CET* conexo de la inicialización del brazo-ventosa en el *GRAF CET* parcial (GI). Si se compara la implementación con el *GRAF CET* citado, se puede apreciar que tiene una etapa adicional. Esta etapa incorpora una transición con la condición de activación de la etapa encapsulante (“*X13*”), lo cual provocará que hasta que no se active la etapa encapsulante no se activará la siguiente etapa y por lo tanto no se activará ninguna acción.

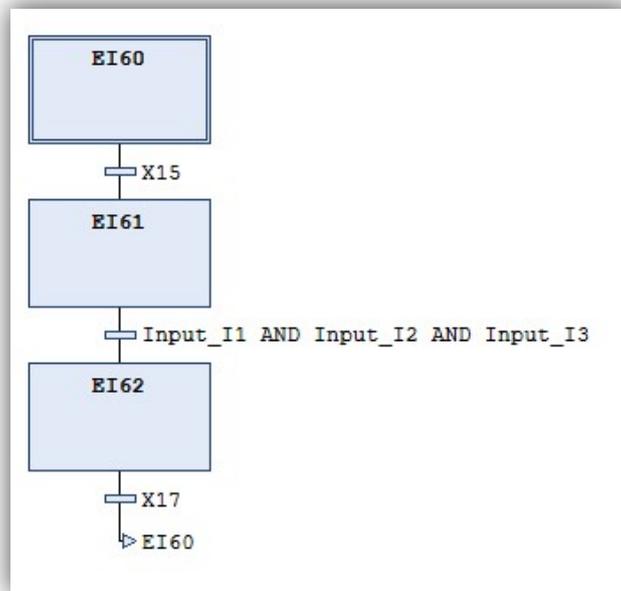


Figura 5.25. Programa “InicializaciónBrazoVentosa” implementado en SoMachine



La etapa adicional, solo se añade a los *GRAFSET's* en los que su etapa inicial, incorpora algunas acciones. Por ejemplo, la implementación del *GRAFSET* conexo brazo-ventosa mostrado en la *Figura 5.22*, el cual está contenido en el *GRAFSET* parcial (*GF*), no incorpora ninguna etapa adicional, pero lo que sí incorpora es la condición de activar la segunda etapa solo si nos encontramos en la “etapa 17” la cual es la encapsulante de dicho *GRAFSET*.

Una vez modificada la estructura, se necesita un sistema de reinicio, ya que, si éste no se incorporara, cuando finalizara la etapa encapsulante el programa seguiría ejecutándose. Por este motivo, se ha implementado un programa en (LD) “ControlEncapsulado”, mostrado en la *Figura 5.26*, en el cual se han introducido funciones *F_TRIG*, proporcionadas por *SoMachine* para la detección de flancos descendentes de la variable (CLK), que en nuestro caso serán las variables encapsuladas. Al detectar el flanco descendente se activa la señal (Q) durante un ciclo, que en nuestro caso es la variable “*SFCInit*” explicada anteriormente de cada programa en lenguaje *SFC* que implementa un *GRAFSET* Encapsulado.

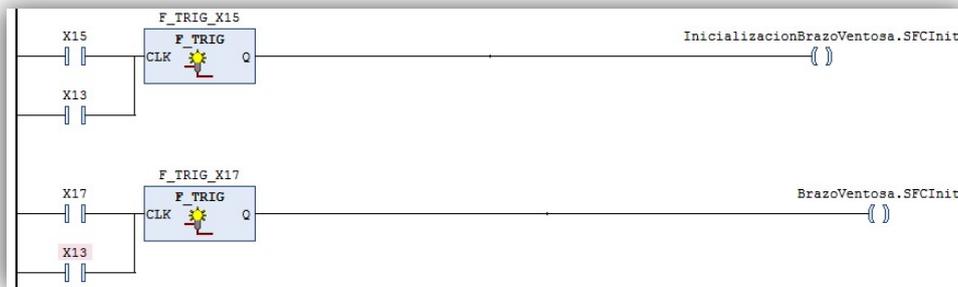


Figura 5.26. Programa “ControlEncapsulado” implementado en *SoMachine*

Forzado

La implementación de dicha función en *SoMachine* es bastante similar a cómo se ha implementado en *Tia Portal*, pero en este caso, como el único forzado que aparece es el de congelar el sistema, ha sido mucho más rápida.

De nuevo, se tiene que para congelar el sistema basta con poner una condición negada de la etapa que fuerza a los otros *GRAFSET's* a todas las acciones que hacen evolucionar el sistema para que no se cumpla ninguna transición y por lo tanto se quede “congelado” en la etapa donde se encuentre en ese momento.

Como se muestra en la *Figura 5.23* mostrada anteriormente, se puede apreciar cómo se ha añadido la negación de “*X1*” a la salida “*Output_Q1*”.

Se recuerda nuevamente que solamente se han condicionado aquellas acciones que hagan evolucionar el sistema y en las que no se corra ningún peligro, tanto humano como material, ya que, por ejemplo,

aparece la variable “Output_Q8” que se encarga de activar o desactivar la succión del brazo-ventosa, y es obvio que dicha variable no se tiene que condicionar.

Temporizadores

Existen dos temporizadores que se pueden configurar en el sistema, éstos son: “T_Succión” y “T_Despegue”.

Estos temporizadores se han introducido con el fin de dotar al ingeniero con el poder de configurarlos vía HMI para que la ventosa agarre y despegue las piezas correctamente.

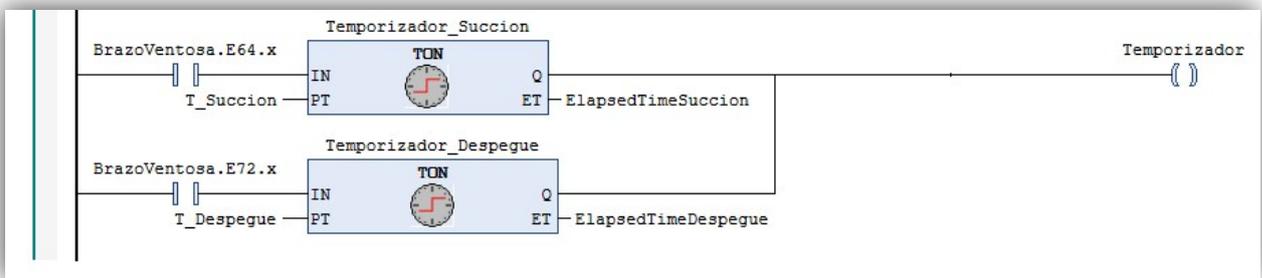


Figura 5.27. Programa de tipo transición “Temporizador” incorporado en el programa “BrazoVentosa”

Para su implementación en SoMachine se ha creado un nuevo programa de tipo transición dentro del POU “BrazoVentosa”, el cual aparece en la Figura 5.27.

El funcionamiento de este tipo de bloques es sencillo, se introducen las ecuaciones lógicas de la transición, en este caso, dos bloques TON (*Modicon M241 Logic Controller, Guía de programación*), los cuales funcionan exactamente igual que en TIA Portal, y una bobina con el nombre del programa de tipo transición creado, y una vez se cumplan las condiciones y se active la bobina, se activará la transición y se activará la siguiente etapa.

En la Figura 5.22, se puede ver cómo aparece el nombre de “Temporizador” en una transición.



Conversiones WORD a TIME

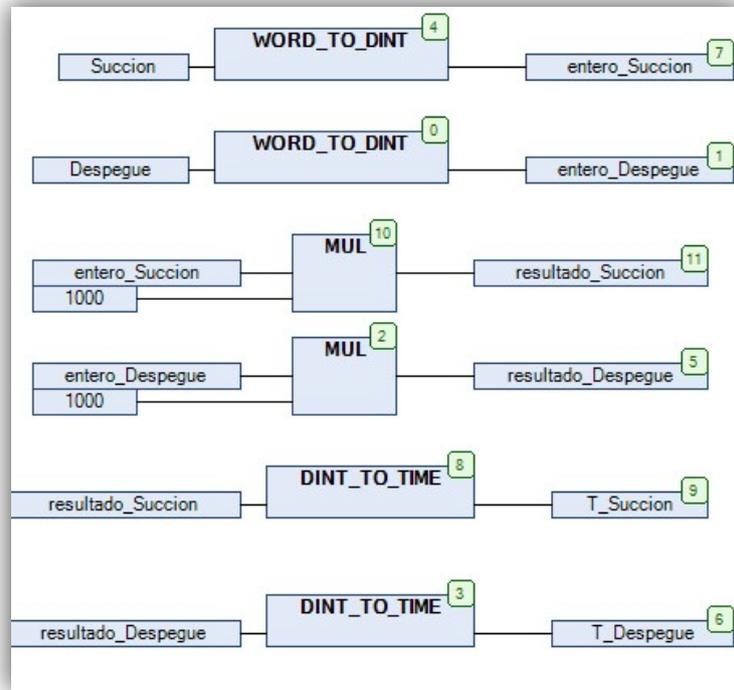


Figura 5.28. Programa “Word_Time” implementado en SoMachine

En la Figura 5.28, aparece el programa “Word_Time”. Se puede observar viendo la figura que el funcionamiento es idéntico a la conversión que se implementó en Tia Portal, la única diferencia es que, en este caso, se tiene que convertir solamente dos variables temporales, por lo tanto, se prescinde de un sistema de llamada.



5.3. Implementación de la interfaz de usuario

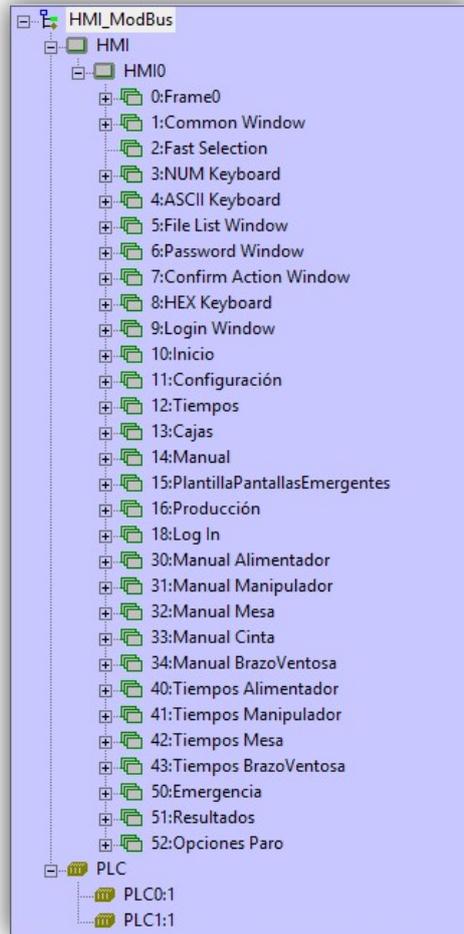


Figura 5.29. Estructura de programa generada en NB-Designer

En el presente apartado se explicarán los objetos funcionales y configuraciones utilizados en el proyecto, que han servido para poder lograr una interfaz sencilla, intuitiva, agradable, pero a su vez, muy potente. El funcionamiento de la HMI a nivel de usuario se expone en el documento *Manual de usuario*. (NB-Designer, OPERATION MANUAL)

En la *Figura 5.29* se muestra el árbol del proyecto que se ha generado en NB-Designer al finalizar la programación.

Aclarar, que la explicación de la configuración de la comunicación ModBus y el direccionamiento de variables que se utilizará en los objetos funcionales se detallará en el *apartado 5.4.4*, por lo tanto, en la explicación solamente se nombrará la variable a la que se quiere hacer referencia.

Los objetos funcionales que se van a desarrollar son los siguientes:



- Botón (SB), Lámpara (BL) e Interruptor de bit (SW).
- Ventana Directa (DW).
- Botón (SWD), Lámpara (WL) e Interruptor (MSW) de palabras.
- Ventana Indirecta (IDW).
- Visualizador de Alarmas (AD).
- Entradas (NI, TI) y Visualizadores (ND, TD) numéricos y de texto.
- Gráfico de tendencias (TR).
- Botón de Función (FK).
- Permisos de usuarios.

Botón (SB), Lámpara (BL) e Interruptor de bit (SW)

Estas funciones permiten activar o desactivar un bit, y mostrar información de una dirección de lectura y escritura idéntica, representar información dependiendo un valor de un bit, o combinar estas dos acciones con direcciones de lectura y escritura distintas (interruptor de bit).

Por ejemplo, en la *Figura 5.30* aparece el aspecto que tiene una parte de la interfaz de la pantalla “Manual Mesa” la cual se compone de una serie de botones y lámparas que harán el control muy intuitivo.

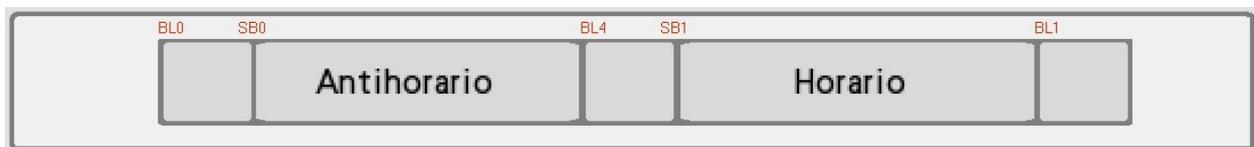


Figura 5.30. Detalle de la pantalla “Manual Mesa”

En la *Tabla 5.3*, se muestra una descripción del funcionamiento de cada elemento.

Identificador	Variable lectura	Variable escritura	Descripción
BLO	“I1”	-	Cambiará de estado si la variable “I1” (final de carrera mesa en manipulador) se activa
SB0	“PQ2”	“PQ2”	Activará la variable “PQ2” (Pulsador manual para salida Q2) y cambiará de estado mientras éste esté activo.



BL4	"I4"	-	Cambiará de estado si la variable "I4" (final de carrera mesa en sierra) se activa.
SB1	"PQ1"	"PQ1"	Activará la variable "PQ1" (Pulsador manual para salida Q1) y cambiará de estado mientras éste esté activo.
BL0	"I2"	-	Cambiará de estado si la variable "I2" (final de carrera mesa en cinta) se activa.

Tabla 5.3. Descripción de funcionamiento de los elementos "BL0", "SB0", "BL4", "SB1" y "BL0" de la pantalla "Manual Mesa"

Para dotar al sistema de una mayor capacidad de interacción con el usuario, se han utilizado la opción "Mostrar según condición → Referencia indirecta" la cual permite mostrar un objeto o no dependiendo de si una variable se encuentra activada o contiene un valor en concreto.

La configuración anterior, se ha utilizado con el objetivo de poder mostrar toda la información de un conjunto de elementos, ya que NB-Designer funciona con un sistema de capas, y capas superiores ocultan a capas inferiores, por lo tanto, es necesario aplicar condiciones de visualización (C.V) para que esto no ocurra.

En la *Figura 5.31*, aparece un ejemplo de la configuración comentada, la configuración hace referencia al elemento "BL3 (Botón de Inicio)", y la condición a la dirección "2.04", que como se verá en el apartado de direccionamiento se trata de la variable "X15", por lo tanto, cuando esta etapa se encuentre activa, el elemento se mostrará y funcionará normalmente.

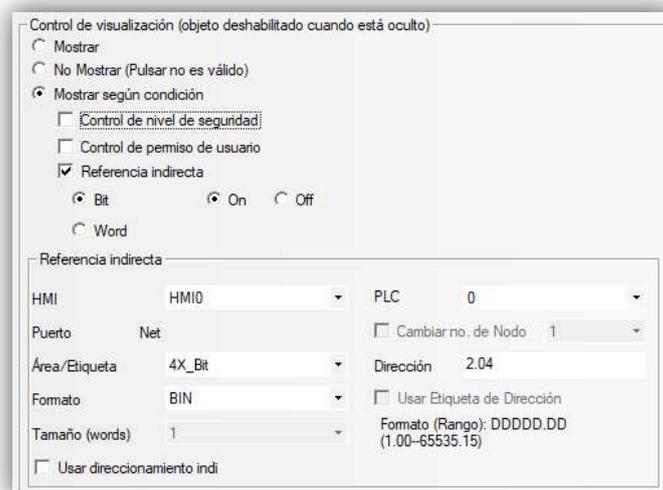


Figura 5.31. Configuración de visualización del elemento "Botón de Inicio"



Ventana Directa (DW)

Esta función se encarga de mostrar una pantalla emergente la cual se selecciona dependiendo del valor de un bit. La presente función se ha utilizado en diversas ocasiones en el proyecto, las cuales vienen listadas en la *Tabla 5.4*, se puede apreciar, que todas aparecen en la pantalla “Inicio”, esto es debido a que es la que más elementos de control tiene y con la que más se interactúa con el funcionamiento del sistema.

Pantalla	Identificador	Variable de activación	Descripción
Inicio	DW0	“LB0”	Mostrará la pantalla “Log In” cuando el bit interno LB0 se active.
	DW1	“X1”	Mostrará la pantalla “Emergencia” mientras la variable “X1” (etapa 1 del GRAFCET Emergencia) esté activa.
	DW2	“X11”	Mostrará la pantalla “Resultados” mientras la variable “X12” (etapa 12 del GRAFCET MF) esté activa.
	DW4	“X12”	Mostrará la pantalla “Opciones Paro” mientras la variable X13 se encuentre activa.
	DW5	“AvisoReset”	Mostrará una pequeña pantalla que mostrará un mensaje de texto que avisará que se tiene que reiniciar las variables de seguridad para continuar.

Tabla 5.4. Descripción y variables de activación de las distintas ventanas directas del proyecto

Botón (SWD), Lámpara (WL) e Interruptor (MSW) de palabras

El funcionamiento de estas funciones es similar a las funciones de “Botón (SB), Lámpara (BL) e Interruptor de bit (SW)”, pero en este caso, éstas manipulan o muestran información de palabras en vez de bits.

Por ejemplo, para el correcto funcionamiento de la función que se explicará en la siguiente función (Ventana Indirecta), ha sido necesario escribir ciertos valores en palabras locales “LW”, para ello, se ha implementado un tipo de menú, el cual se muestra en la *Figura 5.32*, que se compone de un conjunto de botones y lámparas.

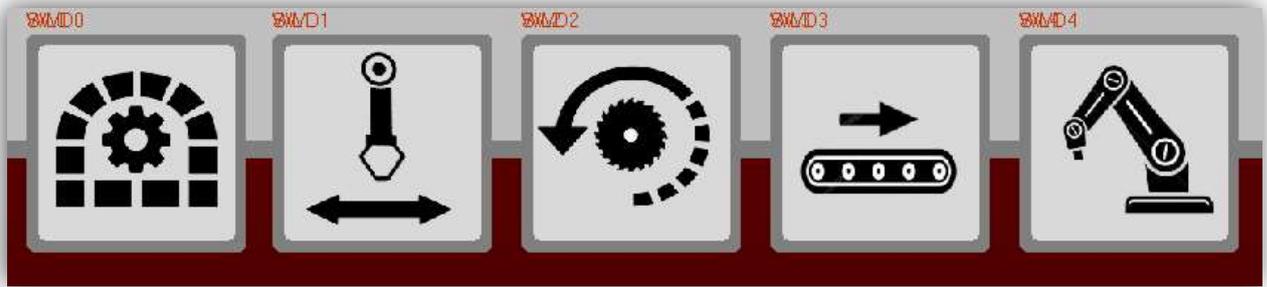


Figura 5.32. Menú implementado en la pantalla “Modo manual”

Como el funcionamiento de los elementos del menú es idéntico, se procede a explicar el conjunto botón-lámpara de la izquierda, el cual se encargará de escribir el valor que se encargará de mostrar el modo manual del alimentador.

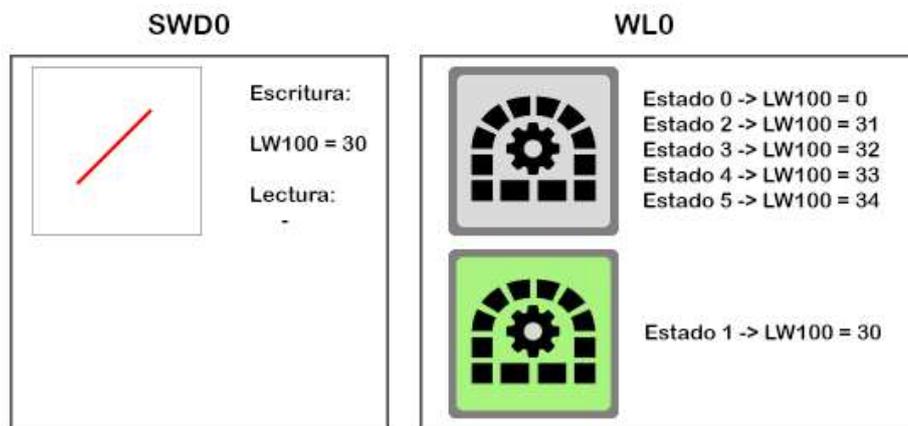


Figura 5.33. Esquema de información de las propiedades del botón “SWD0” y la lámpara “WL”

La Figura 5.33 muestra la descomposición del conjunto comentado, a la izquierda se muestra el botón “SWD0”, el cual no tiene gráficos (será transparente), al pulsar dicho botón, éste añadirá el valor “30” a la palabra “LW100”. Por otro lado, a la derecha se muestra la lámpara “WLO”, la cual se mostrará en su estado 1 cuando el botón anterior se pulse, es decir, cuando “LW100” valga 30, la cual nos proporcionará el mismo icono, pero en este caso, con un tono verde el cual transmite la información de que, como se verá en el apartado posterior, nos encontramos en el modo manual del alimentador. Se puede apreciar, que, para los otros valores de la palabra, la lámpara muestra un tono grisáceo que indica que está apagada y no los encontramos en el modo manual del sector al que representa el icono.



Ventana Indirecta (IDW)

Al igual que en la función “Ventana Directa” se mostraba una pantalla creada con antelación basándonos en el valor de un Bit, la presente función es similar, pero con la diferencia de que en esta ocasión la información se recoge del valor de una palabra a la cual se le escribirá el número de la pantalla que se quiere mostrar. En la *Tabla 5.5*, aparecen las Ventanas Indirectas que se han empleado en el proyecto.

Pantalla	Identificador	Descripción
Manual	IDW0	Será la encargada de mostrar las distintas pantallas del modo manual de cada sector de nuestro sistema.
Tiempos	IDW0	Será la encargada de mostrar las pantallas que contienen las entradas de texto para cada sector de nuestro sistema.

Tabla 5.5. Descripción de las ventanas indirectas del proyecto

Para programar esta función se ha trabajado de nuevo con variables internas de la HMI, ya que no se requiere ninguna información del autómatas. A continuación, se detalla el proceso que se ha seguido para la programación de las ventanas indirectas del proyecto.

Ventana Indirecta IDW0 (Manual)

En primer lugar, se arrastró la función “Ventana Indirecta” a la pantalla de trabajo y se modificó su tamaño para que cuadrara a la perfección con las pantallas que emergerán de ésta.

En segundo y último lugar, se introduce la palabra que registrará en todo momento la pantalla indirecta, y que a la que se escribirá para indicarle qué pantalla deberá mostrar, en este caso “LW100” a la cual escriben los botones del menú como se ha explicado en el apartado anterior.

Una vez configurada la ventana indirecta y el menú explicado en el subapartado anterior, se habrá dotado al sistema de una interfaz limpia e intuitiva la cual mostrará el modo manual de cada sector dependiendo del botón que se pulse. Ya que, si por ejemplo se pulsase el icono que muestra un brazo-ventosa, el botón escribirá el valor 34 a la palabra “LW100”, y la ventana indirecta provocará que emerja la pantalla 34 “Manual Brazo Ventosa”.

Ventana Indirecta IDW0 (Tiempos)

La base del funcionamiento es idéntica a la ventana indirecta “IDW0” (Manual) pero en este caso se ha optado por utilizar interruptores (MSW) ya que incorporan la opción de representar la información en una lista desplegable y dotan al sistema de una mayor limpieza y claridad.



En la *Figura 5.34*, aparece representado el esquema a seguir para implementar el sistema elegido para la configuración de variables temporales de nuestro sistema.

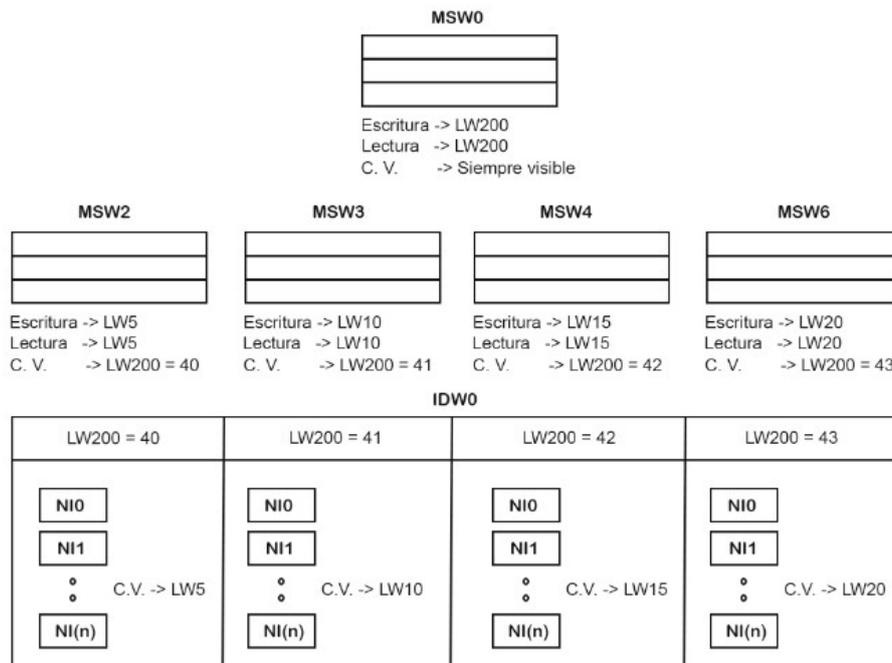


Figura 5.34. Esquema de escritura, lectura, y C.V. (Condiciones de visualización) de los elementos que forman la lógica de representación en la modificación de tiempos

Para conseguir el funcionamiento deseado, se ha incorporado de nuevo la configuración de “Mostrar según condición → Referencia indirecta”.

El funcionamiento es el siguiente:

Se tiene un interruptor de palabra el cual será el encargado de escribir en la palabra “LW200” el número de identificación de la pantalla que se mostrará mediante la ventana indirecta “IDW0” y el que servirá para condicionar qué interruptor se mostrará, ya que como se puede observar, llevan ligados una condición de visualización (C.V.) que depende de “LW200”. Una vez se haya mostrado el interruptor de palabra del sector a modificar, se mostrarán las entradas y visualizadores de texto, que, como se verá a continuación, serán los encargados de escribir y mostrar valores numéricos en la dirección de memoria indicada, y de nuevo nos encontramos de que los distintos elementos funcionales numéricos están condicionados visualmente con el valor de las palabras que mostrarán los interruptores más específicos.

[Visualizador de Alarmas \(AD\).](#)

Esta función será la encargada de mostrar en todo momento las alarmas que se encontrarán activas.

Para que esta función funciona correctamente, se ha tenido que configurar previamente en “Configuración de Alarmas”. Las direcciones de las alarmas apuntan al autómata Siemens S7-1200, éste es el encargado de la gestión de Seguridad (véase *Figura 5.35*).

No.	Tipo	HMI	Información d...	Dirección	Alarma	Contenido
0	0	0	HMI0:PLC0:1	4X_Bit:3.01	On	Seta
1	0	0	HMI0:PLC0:1	4X_Bit:3.02	On	Puerta
2	0	0	HMI0:PLC0:1	4X_Bit:3.03	On	Cable

Figura 5.35. Configuración de Alarmas introducida en NB-Designer

Para implementar el visualizador, simplemente se arrastró (AD1) a una región de la pantalla “Emergencia” la cual, como se ha visto en el subapartado “Ventana Directa” emergerá cuando “X1”, se active.

La *Figura 5.36*, muestra la región de la pantalla donde aparece la función desarrollada comparándola entre el aspecto que tiene en el área de trabajo, y el aspecto que tiene en una simulación offline cuando “Z_Seta” se activa.

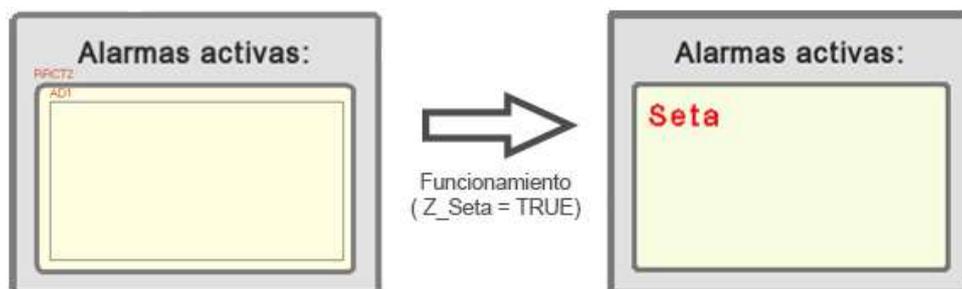


Figura 5.36. Comportamiento de la función “AD1” cuando la variable Z_Seta e activa

Entradas (NI, TI) y Visualizadores (ND, TD) numéricos y de texto.

Mediante las presentes funciones, se ha podido dotar al sistema de la capacidad de introducir los valores deseados a las palabras a modificar, y mostrar el valor a tiempo real de éstas.

La implementación de las funciones es simple, y se consigue arrastrando la función a la ventana de trabajo y configurar los parámetros con el fin de direccionar los valores introducidos a las variables correspondientes en cada autómata para que éstos las transformen de la manera adecuada para introducirlos correctamente a los bloques correspondientes como se ha explicado en los programas de conversión Word a Time explicados en la implementación.



Gráfico de tendencias (TR)

Como se puede apreciar en el documento “Manual de Usuario”, la HMI lleva incorporada una pantalla de información a tiempo real de la producción, esto se ha conseguido mediante un gráfico de tendencias (TR) el cual nos mostrará el cambio de una palabra (eje y) respecto al tiempo de muestreo (eje x), que en este caso será 5 segundos.

La implementación es bastante sencilla, se consiguió arrastrando la función “Gráfico de tendencias” al área de trabajo y configurarla de manera de poder monitorizar la variable “npe”.

Botón de Función (FK)

A pesar de las múltiples acciones que nos permite realizar esta función, se ha utilizado en nuestro proyecto para navegar por las distintas pantallas y tener un control completo de las pantallas que estarán ejecutándose en todo momento.

Por ejemplo, en la *Figura 5.37*, aparece la configuración del “FK Producción”, al aplicar dicha configuración, cuando se pulse el FK se abrirá la Pantalla “Producción”, pero si no se incorpora un FK que cierre la pantalla, la pantalla de producción se mantendrá ejecutándose en segundo plano y podría consumir demasiados recursos, esta es la razón por la que se han incorporado FK para cerrar las pantallas cuyas configuraciones aparecen en la *Figura 5.38* y que en la pantalla aparecen como botones con la etiqueta “Volver”.

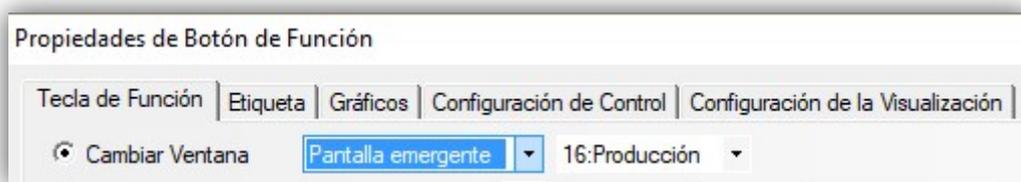


Figura 5.37. Configuración de la Tecla de función “FK Producción”

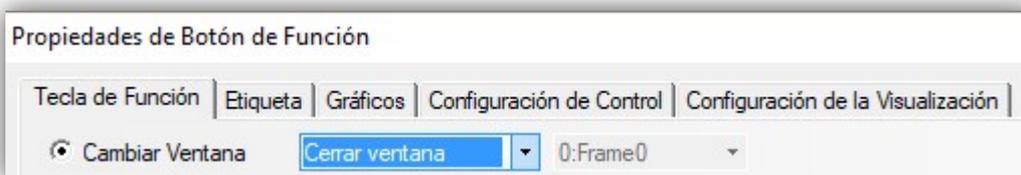


Figura 5.38. Configuración de la Tecla de función “Volver”



Permisos de usuarios

Mediante esta configuración, se ha dotado al sistema de la posibilidad de restringir algunas acciones a los usuarios que no tienen permisos suficientes.

Para ello, se ha introducido en las propiedades de la HMI la información de los usuarios, la cual se muestra en la siguiente tabla:

Usuario	Duración sesión (min)	Permisos
Ingeniero	5	Cuadro de mando Modo manual Configuración Alarmas
Operario	Indefinido	Cuadro de mando Modo manual.

Tabla 5.6. Información sobre los permisos y duración de sesión de las dos cuentas de usuario de nuestro proyecto

Como se puede apreciar, se el usuario “Ingeniero” tiene acceso a todas las pantallas de la HMI y su tiempo de sesión es limitado, para que en caso de que no cierre sesión, ésta se cierre por seguridad. Por otro lado, el usuario “Operario”, solo tiene acceso al cuadro de mando y al modo manual, por lo tanto, no podrá decidir si continuar produciendo en el caso de que una alarma salte, y no podrá cambiar la configuración del sistema.

Un ejemplo de cómo implementar la presente configuración se muestra en la *Figura 5.39*, en ella aparece la configuración del elemento (FK) que abriría la pantalla de Modo Manual, en concreto la pestaña de “Configuración de control”.

Se aprecia que se ha activado que la configuración del táctil habilitado sea condicional, mediante control de permisos de usuario, y el permiso será “Modo Manual”. Por lo tanto, se podrá acceder al modo manual siempre y cuando alguno de los dos usuarios comentados haya iniciado sesión.

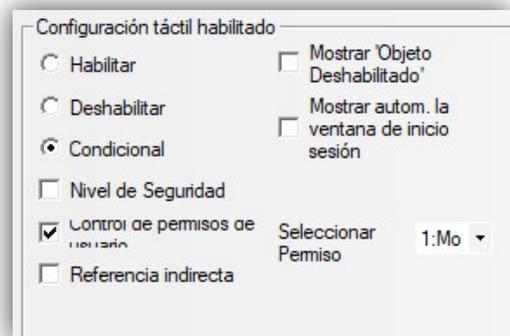


Figura 5.39. Configuración de habilitar táctil del elemento “FK Manual”

5.4. Implementación de la comunicación (ModBus TCP/IP)

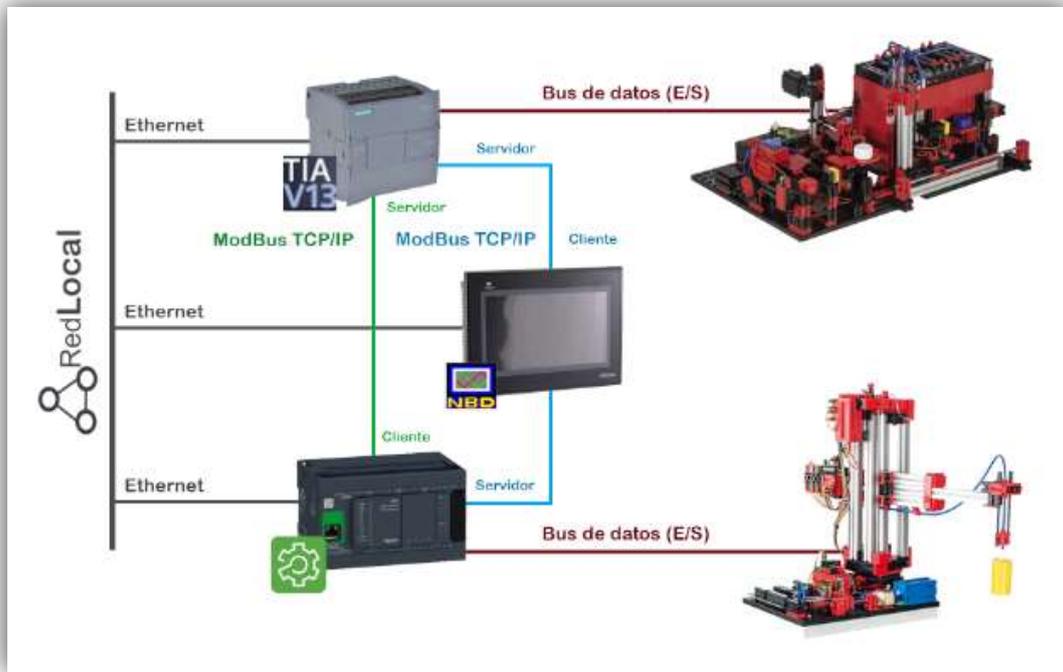


Figura 5.40. Esquema de comunicación entre los distintos dispositivos del sistema

Como se introdujo en el *capítulo 3* el sistema iba a realizar la comunicación mediante el protocolo de comunicación ModBus TCP/IP, pero no se describieron las comunicaciones que deberían existir.

En la *Figura 5.40* aparecen resumidas y aunque esta las muestra de forma intuitiva, se procede a exponerlas:

- Comunicación ModBus TCP/IP (Azul): En esta comunicación, la HMI será el cliente y será la que solicite información a los servidores que serán los autómatas, por lo tanto, se podría decir que es la comunicación incorporada para poder dotar al sistema de la HMI.
- Comunicación ModBus TCP/IP (Verde): En esta comunicación, no entra en juego la HMI. Por un lado, se tiene al autómata Schneider M241 como Cliente y por otro, al autómata Siemens S7-1200 como servidor. Esta comunicación se ha implementado para poder compartir variables entre ambos autómatas.



5.4.1. Implementación en Siemens S7-1200

En la *Figura 5.8* se mostraba la existencia de un grupo que contenía los bloques de programa necesarios para llevar a cabo la comunicación ModBus. Como se ha comentado anteriormente, el autómata S7_1200 adopta el rol de servidor en ambas conexiones de nuestra red, Tia Portal nos ofrece un bloque dentro de su librería denominado “MB_SERVER” el cual nos proporcionará dicha característica.

En la *Figura 5.41* se muestra el contenido del bloque “MB_SERVERS” del grupo “Servidor ModBus”.

Como se puede apreciar, los bloques necesitan unos parámetros de entrada, de los cuales se destacan dos debidos a su importancia en este apartado:

- **MB_HOLD_REG:** En este parámetro se introduce un DB el cual contiene un vector, que en nuestro caso es de tipo WORD, en el cual se escribirán o se leerán los datos con previa solicitud de los clientes. El mecanismo de direccionamiento para acceder a cada palabra de dicha matriz se explicará en el *subapartado 5.4.4*.
- **CONNECT:** En este parámetro se deben introducir la información de la comunicación, en nuestro caso en dos DB que contienen dos parámetros de tipo “TCON_IP_v4”, los cuales se han configurado para que el sistema funcione correctamente con una comunicación de tipo ModBus TCP/IP. La única diferencia es la dirección IP del cliente que va a solicitar acceso.

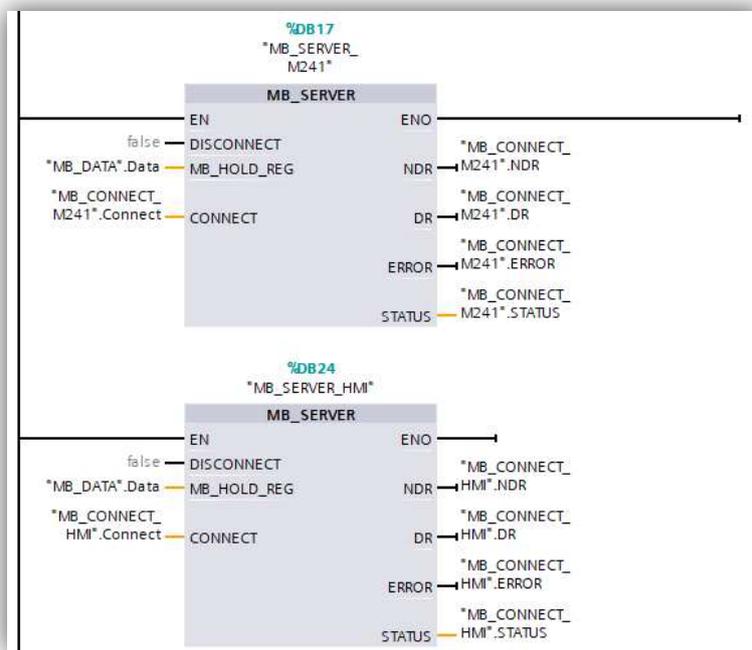


Figura 5.41. Bloque de función MBServers [FC5] implementado en Tia Portal



5.4.2. Implementación en Schneider M241

Como se ha visto anteriormente, el autómata Schneider M241 toma el rol de servidor en una conexión y el rol de cliente en otra conexión. La ventaja de tomar el rol de servidor es que éste viene aplicado por defecto y no es necesario configurar ninguna opción ni añadir ningún programa adicional.

En cuanto al rol de cliente, este se ha conseguido introduciendo las funciones mostradas en la *Figura 5.42*.

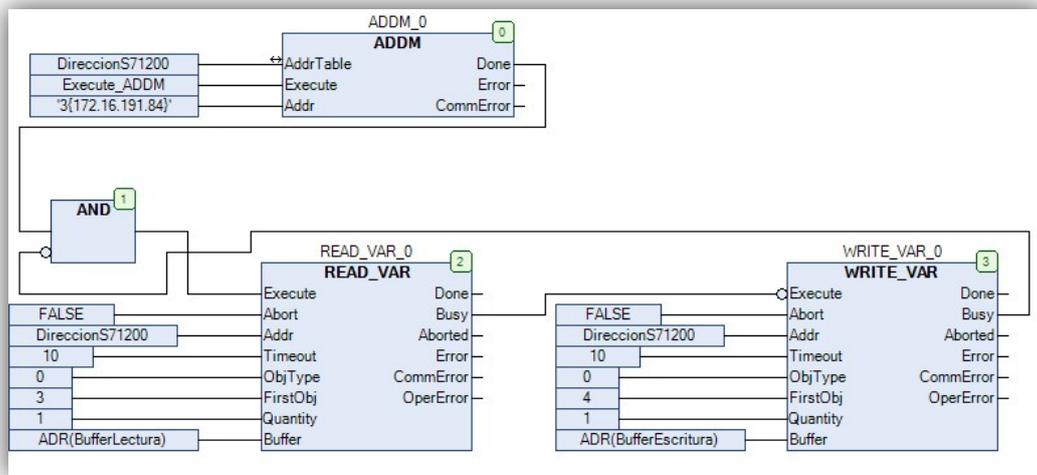


Figura 5.42. Programa “ModBus_Cliente” implementado en SoMachine

El programa funciona de la siguiente manera, el bloque ADDM es el encargado de realizar un direccionamiento al dispositivo, en este caso el autómata S7_1200. Cuando este finaliza se ejecutan los bloques READ_VAR y WRITE_VAR, a los cuales se le introduce la dirección de destino y matriz de almacenamiento de lectura y escritura como veremos en el *subapartado 5.4.4*.

Cabe destacar que como se aprecia, la escritura se produce cuando no está ocupado la lectura, y para más seguridad, se incorpora una condición AND la cuál comienza un nuevo ciclo de lectura/escritura cuando ya no se encuentra ocupado el bloque de escritura y el direccionamiento es correcto.

5.4.3. Implementación en Omron NB7W

Como se ha visto en el árbol del proyecto, la HMI Omron NB7W funciona como cliente en la conexión que participa. Para configurarla correctamente se añadieron en el área de trabajo del árbol del proyecto dos objetos de tipo PLC denominados “ModBus Slave” que hacen referencia al Siemens S7-1200 y al Schneider M241 (véase *Figura 5.43*) para que posteriormente se introdujeran las direcciones IP correspondientes a cada dispositivo y se configurase como puerto de comunicación el 502, que es el utilizado para las conexiones ModBus como bien se muestra en la *Figura 5.44*.

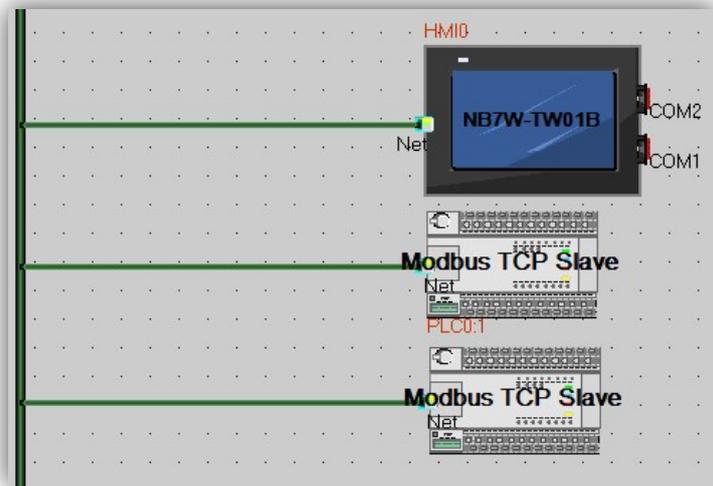


Figura 5.43. Aspecto del área de trabajo de la ventana de conexiones en NB-Designer

Dispositivo	Dirección IP	Puerto	Protocolo	Maestro/E...	No. de N...	No. de P
HMI0	172.16.191.53	502	Modbus TCP	M		
PLC0	172.16.191.84	502	Modbus TCP Slave	S	1	
PLC1	172.16.191.112	502	Modbus TCP Slave	S	1	

Figura 5.44. Configuración de las comunicaciones en NB-Designer



5.4.4. Direccionamiento ModBus

En el apartado anterior se ha explicado la configuración y los programas a añadir en los distintos softwares para que se produzca una comunicación ModBus entre los distintos elementos.

El presente apartado por el contrario se centra en el cómo acceder a cada una de las variables utilizadas en la comunicación, ya que la metodología empleada es completamente distinta para cada elemento.

En la siguiente lista aparecen las distintas solicitudes entre los dispositivos que aparecen en nuestro proyecto:

- *Omron (Cliente) -> Schneider M241 (Servidor)*
- *Omron (Cliente) -> Siemens S7 1200 (Servidor)*
- *Schneider M241 (Cliente) -> Siemens S7 1200 (Servidor)*

Omron (Cliente) -> Schneider M241 (Servidor)

En la *Tabla 5.7*, aparecen las referencias a utilizar y las funciones que realiza cada una de ellas.

Referencia	Acción
0xxxx	Leer/Escribir Salidas discretas
1xxxx	Leer Entradas discretas
3xxxx	Leer registros de entrada externos
4xxxx	Leer/Escribir Registros (16-bits)

Tabla 5.7. Acción que produce cada referencia en la comunicación ModBus TCP/IP

Por otro lado, en la Figura 5.45, aparece un cuadro resumen de cómo acceder a las distintas salidas (QX), entradas (IX), y registros (M) del Schneider M241.



0X (Bit)	%QX (Bit)	1X (Bit)	%IX (Bit)	4X_Bit	Controller Addressing			4X (Word)
1	%QX0.0	1	%IX0.0	1.07 ... 1.00	%MX0.7...%MX0.0	%MB0	%MW0	1
2	%QX0.1	2	%IX0.1	1.15 ... 1.08	%MX1.7...%MX1.0	%MB1		2
3	%QX0.2	3	%IX0.2	2.07 ... 2.0	%MX2.7...%MX2.0	%MB2	%MW1	3
...	2.15 ... 2.08	%MX3.7...%MX3.0	%MB3		4
8	%QX0.7	8	%IX0.7	...	%MX4.7...%MX4.0	%MB4	%MW2	5
9	%QX1.0	9	%IX1.0	...	%MX5.7...%MX5.0	%MB5		6
10	%QX1.1	10	%IX1.1	...	%MX6.7...%MX6.0	%MB6	%MW3	7
...	%MX7.7...%MX7.0	%MB7		8

Figura 5.45. Tabla resumen de direccionamiento ModBus para Siemens S7-1200 (CFP-UPV)

Por ejemplo, en la Figura 5.45, se muestra una descomposición detallada de los elementos de la pantalla “Manual Brazo Ventosa”, si se observa el Anexo I, en el cual aparecen las direcciones de memoria en las cuales tiene almacenado el autómata las variables de interacción con los botones se tiene que:

$Input_I1 \rightarrow IX0.4$

$PQ1 \rightarrow MX1.1$

$CuentaVertical \rightarrow MW20$

Se puede confirmar, que se cumple la metodología extraída de la Figura 5.45 para direccionar variables en ModBus.



Figura 5.45. Esquema de direccionamiento de los elementos “BLO”, “SB0” y “ND0” contenidos en la pantalla “Modo Manual” en el sector del Brazo-ventosa

[Omron \(Cliente\) -> Siemens S7 1200 \(Servidor\)](#)

La metodología empleada en este caso es bastante similar a la del caso anterior en cuanto a entradas y salidas digitales, pero dista bastante si nos referimos a registros.



En el subapartado 5.4.1., se introdujo el parámetro “MB_HOLD_REG”, el cual contenía un DB al que se le introducía un vector de tipo WORD. En el caso de nuestro proyecto el DB es “MB_DATA [DB19]” y se muestra en la *Figura 5.46*.

La diferencia radica en que, en este caso, el direccionamiento ModBus va destinado a las palabras de dicho vector, no a las direcciones de memoria que habíamos empleado para programar todo el proyecto, las cuales como se ha comentado anteriormente aparecen representadas en el *Anexo I*.

MB_DATA				
	Nombre	Tipo de datos	Valor de arranq...	Remanen...
1	Static			<input type="checkbox"/>
2	Data	Array[0..50] of Word		<input checked="" type="checkbox"/>
3	Data[0]	Word	16#0	<input checked="" type="checkbox"/>
4	Data[1]	Word	16#0	<input checked="" type="checkbox"/>
5	Data[2]	Word	16#0	<input checked="" type="checkbox"/>
6	Data[3]	Word	16#0	<input checked="" type="checkbox"/>
7	Data[4]	Word	16#0	<input checked="" type="checkbox"/>
8	Data[5]	Word	16#0	<input checked="" type="checkbox"/>
9	Data[6]	Word	16#0	<input checked="" type="checkbox"/>

Figura 5.46. Detalle del parámetro “MB_DATA [DB19]” implementado en Tia Portal

Como se puede observar, contiene 50 componentes de tipo WORD, los cuales serán escritos y leídos siguiendo la metodología expuesta en la siguiente tabla.

Modbus address	Absolute address	Symbolic name
40001	DB1.DBW0	"HoldingRegister".Read_HoldingRegister[0]
40002	DB1.DBW2	"HoldingRegister".Read_HoldingRegister[1]
40003	DB1.DBW4	"HoldingRegister".Read_HoldingRegister[2]
40004	DB1.DBW6	"HoldingRegister".Read_HoldingRegister[3]
40005	DB1.DBW8	"HoldingRegister".Read_HoldingRegister[4]

Figura 5.47. Tabla de direccionamiento para el vector “Data” introducido en el parámetro “MB_DATA [DB19]” (CFP-UPV)

La escritura y lectura de palabras será sencilla ya que, por ejemplo, para acceder a la palabra “Data[0]” se asignará la siguiente dirección:

$$4X\ 1 \rightarrow Data[0]$$

La escritura y lectura de bits, es menos intuitiva, pero quedará clara una vez explicada.

La *Figura 5.48* muestra un ejemplo de cómo se han direccionado registros en forma de bit. En la figura aparecen dos elementos incluidos en la pantalla “Manual Manipulador”, se puede observar que como se ha comentado anteriormente, el direccionamiento de entradas y salidas es idéntico en el apartado anterior, la única diferencia es que en este ejemplo se ha utilizado una salida para mostrar una información, ya que no existía un sensor que informase de que el manipulador se encuentra arriba.



Figura 5.48. Esquema de direccionamiento de los elementos “BL0” y “SB0” contenidos en la pantalla “Modo manual” en el sector Manipulador

A la derecha se puede observar que para acceder a la variable PQ12 se ha escrito la dirección “4X_Bit 1.11” la cual nos indica que está leyendo la información de la palabra “Data[0]”, pero en concreto, el bit número 11.

Por lo tanto, surge el problema de tener que asignar los valores de las palabras del parámetro “MB_DATA [DB19]” a las variables de nuestro sistema.

Para solucionar el problema, se ha implementado el programa “Direccionamiento Modbus [FC6]”, el cual se muestra descompuesto y reducido en la *Figura 5.49*.

El funcionamiento del programa consiste en asignar el valor del parámetro “MB_DATA [DB19]” a las variables del sistema en el caso de que se esté escribiendo al PLC, o de asignar el valor de las variables al parámetro en el caso de que se esté leyendo del PLC.

Se puede observar, que al principio del programa aparece una condición IF la cual indica que se desactiven algunos bits cuando arranca el sistema. Esto es debido a que se ha tenido que activar la opción de remanencia para todo el vector, ya que se quería que las variables temporales no se desactivasen al arrancar el sistema, y, por lo tanto, al no poder activar la remanencia solo en una serie de elementos se tuvo que activar para todos éstos provocando que en algunos casos se quedasen variables activadas al apagar el sistema las cuales deberían de haberse reseteado en un funcionamiento normal del sistema.



```
1 IF "FirstScan" = TRUE THEN
2     "MB_DATA".Data[1].%X9 := FALSE;
3     "MB_DATA".Data[1].%X10 := FALSE;
4     "MB_DATA".Data[2].%X4 := FALSE;
5     "MB_DATA".Data[2].%X5 := FALSE;
6 END_IF;

9 // Direcciones de escritura HMI ( PLC --> HMI)
10
11 "MB_DATA".Data[2].%X0 := "Z";
12 "MB_DATA".Data[2].%X1 := "Z_Seta";
13 "MB_DATA".Data[2].%X2 := "Z_Puerta";
14 "MB_DATA".Data[2].%X3 := "Z_Cable";
29 // Direcciones de lectura HMI ( HMI --> PLC)
30
31 "Continuar producción" := "MB_DATA".Data[2].%X4;
32 "Detener producción" := "MB_DATA".Data[2].%X5;
33 "Siguiete" := "MB_DATA".Data[1].%X14;
58 //Direcciones de escritura M241 ( S7_1200 -> M241)
59
60 "MB_DATA".Data[3].%X0 := "X1";
61 "MB_DATA".Data[3].%X1 := "X2";
62 "MB_DATA".Data[3].%X2 := "X13";
63 "MB_DATA".Data[3].%X3 := "X15";
70 //Direcciones de lectura M241 ( M241 -> S7_1200)
71
72 "XI61" := "MB_DATA".Data[4].%X0;
73 "VentosaBrazo" := "MB_DATA".Data[4].%X1;
74 "X82" := "MB_DATA".Data[4].%X2;
75 "PARO" := "MB_DATA".Data[4].%X3;
```

Figura 5.49. Descomposición del bloque de función "Direccionamiento ModBus" implementado en Tia Portal

[Schneider M241 \(Cliente\) -> Siemens S7 1200 \(Servidor\)](#)

La metodología empleada para realizar este funcionamiento es bastante similar a la anterior en cuanto al servidor, ya que se escribirá en el parámetro "MB_DATA" y se gestionará de la misma manera mediante el programa "Direccionamiento ModBus" que como se ha comentado, aparece en la *Figura 5.49*.

Por lo tanto, se procede a explicar, cómo el cliente (Schneider M241) escribe y lee los datos gestionados por el servidor. Como se introdujo en el *subapartado 5.4.1*, en los bloques "READ_VAR" y "WRITE_VAR" aparecen unas entradas las cuales se muestran en la *Tabla 5.8* junto con una descripción:



Entrada	Descripción
ObjType	Indica el tipo de objeto a leer, 0 para Registros
FirstObj	Indica la dirección del primer objeto a leer, si se pone un valor X, corresponderá a la palabra Data[X]
Quantity	Indica el número de palabras que se leerán a partir de la primera.
Buffer	Indica la dirección de la matriz que va a almacenar la información.

Tabla 5.8. Descripción de los parámetros de entrada de los bloques de comunicación "READ_VAR" y "WRITE_VAR" que contiene SoMachine

Combinando la Figura 5.42 y la información de la Tabla 5.8, es obvio que la configuración del proyecto ha sido:

- Para la lectura se ha direccionado solamente la palabra "Data[3]", y la información se almacenará en la matriz "BufferLectura".
- Para la escritura, se ha direccionado solamente la palabra "Data[4]", y la información a escribir se encuentra en la matriz "BufferEscritura".

En la Figura 5.50, se muestra una región de el "GVL", donde se puede apreciar que se han declarado las dos matrices comentadas anteriormente, y justo debajo aparecen las variables que compondrán dichas matrices.

```
//Variables de lectura Siemens S71200

BufferLectura AT %MW60: ARRAY[0..0] OF WORD;

X1      AT %MX120.0: BOOL;
X2      AT %MX120.1: BOOL;
X13     AT %MX120.2: BOOL;
X15     AT %MX120.3: BOOL;
X17     AT %MX120.4: BOOL;
PARAR   AT %MX120.5: BOOL;
X20     AT %MX120.6: BOOL;
I3      AT %MX120.7: BOOL;
X0      AT %MX121.0: BOOL;

//Variables escritura Siemens S71200

BufferEscritura AT %MW65: ARRAY[0..0] OF WORD;

XI61    AT %MX130.0  : BOOL;
Ventosa AT %MX130.1  : BOOL;
X82     AT %MX130.2  : BOOL;
PARO    AT %MX130.3  : BOOL;
```

Figura 5.50. Detalle de la declaración de las matrices "BufferLectura" y "BufferEscritura" en el GVL de SoMachine



6. PUESTA EN MARCHA

En el presente apartado se exponen las condiciones necesarias para que la puesta en marcha del sistema diseñado no cause ningún problema. Estas condiciones son las siguientes:

- 1) Cablear los autómatas siguiendo el direccionamiento de variables de *entradas y salidas (E/S)* que se expone en el *Anexo I*.
- 2) Comprobar que los distintos dispositivos del sistema tienen cargados sus respectivos programas, los cuales son:
 - *Siemens S7-1200: Estacion Multiproceso.ap13*
 - *Schneider M241: Brazo Ventosa.project*
 - *Omron NB7W: HMI_ModBus.nbp*

Nota: Para llevar a cabo dicha comprobación, seguir las instrucciones proporcionadas en los correspondientes manuales de usuario: (*S7-1200, Manual del sistema*), (*Modicon M241 Logic Controller, Guía de programación*) y (*NB-SERIES, STEUP MANUAL*).

- 3) Al existir diversas comunicaciones entre los dispositivos, es necesario ponerlos en estado *"Run"* en un orden determinado:
 - 1) Arrancar el Siemens S7-1200, ya que éste es servidor en todas las comunicaciones y debe de estar listo cuando un cliente quiera solicitarle información.
 - 2) Arrancar el Schneider M241, ya que este actúa como servidor de la HMI.
 - 3) Arrancar la HMI.

Una vez seguidas las pautas anteriores el sistema debería funcionar con normalidad y no debería existir ningún problema entre las comunicaciones existentes.



7. CONCLUSIONES

Una vez finalizado el proyecto, se puede afirmar que se han cumplido con todos y cada uno de los objetivos que se tenían al inicio.

En primer lugar, se han adquirido un gran número de conocimientos en cuanto a la programación de autómatas se refiere, ya que se han adquirido conocimientos en hasta tres lenguajes de programación: Contactos (*KOP, LD*), texto estructurado (*SCL, ST*) y *SFC*. Pero sin duda, lo importante ha sido conocerlos y haber desarrollado conocimiento sobre cuándo es más eficiente aplicar un lenguaje u otro.

En segundo lugar, se han implementado hasta dos comunicaciones utilizando para ello el protocolo de comunicación ModBus, el cual es uno de los más utilizados en la industria, adquiriendo por lo tanto unos conocimientos muy valiosos, ya que se ha visto que a pesar de que la implementación en cada dispositivo es distinta, el modo de direccionar es bastante similar, y por lo tanto, tras realizar el presente proyecto se estaría en disposición de comunicarse con cualquier dispositivo industrial que fuese compatible con dicho protocolo.

En tercer lugar, se ha creado un sistema HMI muy potente, debido a que se ha podido cumplir perfectamente las peticiones del cliente, las cuales eran diversas, y requerían bastantes conocimientos de informática industrial para poder implementarlas correctamente. Cabe destacar que, en adición a la potencia se suma un diseño bastante innovador, el cuál ha sido posible crear gracias a la incorporación de iconos, que han sido diseñados mediante programas de diseño gráfico, implementación de menús y sistemas, que hacen que cualquier función del sistema sea accesible de una manera intuitiva y rápida. Por lo tanto, se concluye que la unión entre ingeniería y diseño gráfico puede dar lugar a herramientas muy potentes que pueden dotar a los sistemas automáticos de grandes funcionalidades que pueden llegar a ser muy intuitivas.

Por lo tanto, se extrae la conclusión final de que la unión de los conocimientos anteriores, han dado lugar a una evolución personal, la cual me será útil en un futuro para ejercer como profesional en el campo de la automática.



8. BIBLIOGRAFÍA

SIEMENS SIMATIC S7 Controlador programable S7-1200, Manual del sistema (11/2009). Obtenido de: <https://www.siemens.com>

SIEMENS SIMATIC STEP 7 Basic V13 SP1, Manual del Sistema (12/2014). Obtenido de: <https://support.industry.siemens.com>

Schneider Modicon M241 Logic Controller, Guía de programación (11/2014). Obtenido de: <https://www.schneider-electric.com>

Schneider SoMachine, Guía de programación (04/2014). Obtenido de: <https://www.schneider-electric.com>

Omron NB-Designer, OPERATION MANUAL (2011). Obtenido de: <http://www.omron.com.au>

Omron NB-SERIES, STEUP MANUAL (2011). Obtenido de: <http://www.omron.com.au>

Simarro Fernández, Raúl (2016-2017). Apuntes de Laboratorio de Automatización y Control. Seminario de Automatización. València: UPV.

CFP-UPV (2017). Apuntes de Diploma de especialización Instalación, configuración y programación de sistemas de automatización industrial.

AENOR, (2013). "Norma UNE-EN 60848:2013 Lenguaje de especificación GRAFCET para diagramas funcionales secuenciales."



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





PRESUPUESTO

JUAN CARLOS GARCÍA HERNÁNDEZ



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





Índice

1. CUADRO Nº1: MANO DE OBRA	73
2. CUADRO Nº2: MATERIALES Y MAQUINARIA	73
3. CUADRO Nº3: PRECIOS UNITARIOS	74
4. CUADRO Nº4: PRECIOS DESCOMPUESTOS	74
5. PRESUPUESTO BASE DE LICITACIÓN	76



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





1. CUADRO Nº1: MANO DE OBRA

Código	Unidad	Descripción	Salario (€/h)
MO.GITI	h	Graduado en Ingeniería en Tecnologías Industriales	20

2. CUADRO Nº2: MATERIALES Y MAQUINARIA

Para los presentes materiales y maquinarias se han utilizado las amortizaciones expuestas a continuación:

$$\text{Amortización MM.PSV} = \frac{\text{base de amortización(€)}}{\text{vida útil(h)}} = \frac{800 - 0}{5\text{años} \cdot \frac{1700\text{h}}{1\text{año}}} = 0.09 \text{ €/h}$$

$$\text{Amortización MT.TP13} = \frac{\text{base de amortización(€)}}{\text{vida útil(h)}} = \frac{1800 - 0}{3\text{años} \cdot \frac{1700\text{h}}{1\text{año}}} = 0.35 \text{ €/h}$$

$$\text{Amortización MT.PCS6} = \frac{\text{base de amortización(€)}}{\text{vida útil(h)}} = \frac{24.19 - 0}{1\text{mes} \cdot \frac{140\text{h}}{1\text{mes}}} = 0.17 \text{ €/h}$$

Código	Unidad	Descripción	Precio (€/Unidad)
MT.ASS7	ud	SIMATIC S7-1200 CPU 1214C AC/DC/RLY 6ES7214-1BG40-0XB0	287.27
MT.MAS	ud	SM 1223 DI16/DQ16 x relé, 6ES7 223-1PL32-0XB0	208.58
MT.ASM	ud	Modicon M241 TM241CE40R	430.59
MT.ONB	ud	Omron NB7W-TW01B	903.52
MT.TP13	h	Software Tia Portal V13	0.35
MT.SSM	h	Software So Machine	0
MT.ONBD	h	Software NB-Designer	0
MT.PCS6	h	Software Photoshop CS6	0.17
MM.PSV	h	Ordenador portátil SONY VAIO	0.09



3. CUADRO Nº3: PRECIOS UNITARIOS

<i>Código</i>	<i>Unidad</i>	<i>Descripción</i>	<i>Precio (€/ud)</i>
UD.001	ud	Diseño del sistema automático	800.9
UD.002	ud	Implementación del sistema automático en Siemens S7-1200	1500.28
UD.003	ud	Implementación del sistema automático en Schneider M241	1033.29
UD.004	ud	Desarrollo de la interfaz y la programación del sistema HMI	2114.87
UD.005	ud	Elaboración de la memoria, anexo y manual de usuario	401.8

4. CUADRO Nº4: PRECIOS DESCOMPUESTOS

<i>Código</i>	<i>Unidad</i>	<i>Descripción</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Importe (€)</i>
UD.001	ud	Diseño del sistema automático		800.9	
<i>MO.GITI</i>	h	Graduado en Ingeniería en Tecnologías Industriales	40	20	800
<i>MM.PSV</i>	h	Ordenador portátil SONY VAIO	10	0.09	0.9

<i>Código</i>	<i>Unidad</i>	<i>Descripción</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Importe (€)</i>
UD.002	ud	Implementación del sistema automático en Siemens S7-1200		1500.28	
<i>MO.GITI</i>	h	Graduado en Ingeniería en Tecnologías Industriales	50	20	1000
<i>MT.ASS7</i>	ud	SIMATIC S7-1200 CPU 1214C AC/DC/RLY 6ES7214-1BG40-0XB0	1	287.27	287.2
<i>MT.MAS</i>	ud	SM 1223 DI16/DQ16 x relé, 6ES7 223-1PL32-0XB0	1	208.58	208.58
<i>MM.PSV</i>	h	Ordenador portátil SONY VAIO	50	0.09	4.5



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.



<i>Código</i>	<i>Unidad</i>	<i>Descripción</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Importe (€)</i>
UD.003	ud	Implementación del sistema automático en Schneider M241		1033.29	
<i>MO.GITI</i>	h	Graduado en Ingeniería en Tecnologías Industriales	30	20	600
<i>MT.ASM</i>	ud	Modicon M241 TM241CE40R	1	430.59	430.59
<i>MT.SSM</i>	h	Software So Machine	1	0	0
<i>MM.PSV</i>	h	Ordenador portátil SONY VAIO	30	0.09	2.7

<i>Código</i>	<i>Unidad</i>	<i>Descripción</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Importe (€)</i>
UD.004	ud	Desarrollo de la interfaz y la programación del sistema HMI		2114.87	
<i>MO.GITI</i>	h	Graduado en Ingeniería en Tecnologías Industriales	60	20	1200
<i>MT.ONB</i>	ud	Omron NB7W-TW01B	1	903.52	903.52
<i>MT.SSM</i>	h	Software NB-Designer	25	0	0
<i>MT.PCS6</i>	h	Software Photoshop CS6	35	0.17	5.95
<i>MM.PSV</i>	h	Ordenador portátil SONY VAIO	60	0.09	5.4

<i>Código</i>	<i>Unidad</i>	<i>Descripción</i>	<i>Cantidad</i>	<i>Precio</i>	<i>Importe (€)</i>
UD.005	ud	Elaboración de la memoria, anexo y manual de usuario		401.8	
<i>MO.GITI</i>	h	Graduado en Ingeniería en Tecnologías Industriales	20	20	400
<i>MM.PSV</i>	h	Ordenador portátil SONY VAIO	20	0.09	1.8



5. PRESUPUESTO BASE DE LICITACIÓN

Código	Unidad	Descripción	Cantidad	Precio (€/Unidad)	Importe (€)
UD.001	ud	Diseño del sistema automático	1	800.9	800.90
UD.002	ud	Implementación del sistema automático en Siemens S7-1200	1	1500.28	1500.28
UD.003	ud	Implementación del sistema automático en Schneider M241	1	1033.29	1033.29
UD.004	ud	Desarrollo de la interfaz y la programación del sistema HMI	1	2114.87	2114.87
UD.005	ud	Elaboración de la memoria, anexo y manual de usuario	1	401.8	401.8
		<i>Presupuesto de Ejecución Material (PEM)</i>			5851.14
		<i>Gastos Generales (0.15)</i>			877.67
		<i>Beneficio Industrial (0.06)</i>			351.07
		<i>Presupuesto de Ejecución por Contrata (PEC)</i>			7079.88
		<i>IVA (21%)</i>			1486.77
		<i>Presupuesto base de licitación</i>			8566.65 €

Asciende el presente presupuesto a la expresada cantidad de:

OCHO MIL QUINIENTOS SESENTA Y SEIS EUROS CON SESEINTA Y CINCO CÉNTIMOS



ANEXO I

Variables del Sistema.

JUAN CARLOS GARCÍA HERNÁNDEZ



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





Índice

1. VARIABLES SIEMENS S7-1200	81
1.1. Entradas y salidas (E/S).....	81
1.2. Registros	82
1.2.1. Seguridad	82
1.2.2. Modos de funcionamiento	83
1.2.3. Estación multiproceso	84
1.2.3.1. Alimentador	84
1.2.3.2. Manipulador	85
1.2.3.3. Mesa	86
1.2.3.4. Cinta.....	86
1.2.3.5. Modo manual	87
2. VARIABLES SCHNEIDER M241	88
2.1. Entradas y salidas (E/S).....	88
2.2. Registros	89
2.2.1. Brazo Ventosa.....	89
2.2.2. Configuración cajas.....	89
2.2.3. Gestión paro	90
2.2.4. Modo manual	90
2.2.5. Matrices de coordenadas	90
2.2.5.1. CE2x2	90
2.2.5.2. CE3x3	90
2.2.5.3. CS2x2	91
2.2.5.4. CS3x3	91



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





1. VARIABLES SIEMENS S7-1200

1.1. Entradas y salidas (E/S)

<i>Nombre</i>	<i>Dirección</i>	<i>Descripción</i>
I1	%I8.4	Final de carrera que indica que la mesa está en el manipulador
I2	%I8.5	Final de carrera que indica que la mesa está en la cinta
I3	%I8.6	Sensor óptico que indica que la pieza está al final de la cinta
I4	%I8.7	Final de carrera que indica que la mesa está en la sierra
I5	%I8.0	Final de carrera que indica que el manipulador está en la mesa
I6	%I9.2	Final de carrera que indica que el alimentador está dentro del horno
I7	%I8.1	Final de carrera que indica que el alimentador está fuera del horno
I8	%I9.3	Final de carrera que indica que el manipulador está en el horno
I9	%I8.2	Sensor óptico que indica que la pieza está en el alimentador

<i>Nombre</i>	<i>Dirección</i>	<i>Descripción</i>
Q1	%Q8.0	Motor movimiento giratorio mesa sentido horario
Q2	%Q8.1	Motor movimiento giratorio mesa sentido antihorario
Q3	%Q8.2	Motor movimiento de avance de la cinta
Q4	%Q8.3	Motor sierra
Q5	%Q8.4	Motor movimiento hacia dentro del alimentador
Q6	%Q8.5	Motor movimiento hacia fuera del alimentador
Q7	%Q8.6	Motor movimiento hacia el horno del manipulador
Q8	%Q8.7	Motor movimiento hacia la mesa del manipulador
Q9	%Q9.0	Habilitar señales de entrada del proceso
Q10	%Q9.1	Habilitar compresor
Q11	%Q9.2	Habilitar la succión de la ventosa del manipulador
Q12	%Q9.3	Activar electroválvula para que el manipulador baje
Q13	%Q9.4	Activar electroválvula para que la puerta suba
Q14	%Q9.5	Activar electroválvula para empujar la pieza
Q15	%Q9.6	Activar luz horno



1.2. Registros

1.2.1. Seguridad

<i>Nombre</i>	<i>Tipo</i>	<i>Dirección</i>	<i>Descripción</i>
<i>X0</i>	Bool	%M0.0	Estado etapa 0
<i>X1</i>	Bool	%M0.1	Estado etapa 1
<i>X2</i>	Bool	%M0.2	Estado etapa 2
<i>S0</i>	Bool	%M2.0	Activación etapa 0
<i>S1</i>	Bool	%M2.1	Activación etapa 1
<i>S2</i>	Bool	%M02.2	Activación etapa 2
<i>Z</i>	Bool	%M4.0	Variable general de emergencia
<i>Continuar producción</i>	Bool	%M2.3	Opción continuar producción
<i>Detener producción</i>	Bool	%M2.4	Opción detener producción
<i>Z_Seta</i>	Bool	%M4.1	Variable específica de emergencia (seta)
<i>Z_Puerta</i>	Bool	%M4.2	Variable específica de emergencia (puerta)
<i>Z_Cable</i>	Bool	%M4.3	Variable específica de emergencia (cable)
<i>VentosaBrazo</i>	Bool	%M4.4	Variable que indica el estado de la ventosa del brazo



1.2.2. Modos de funcionamiento

<i>Nombre</i>	<i>Tipo</i>	<i>Dirección</i>	<i>Descripción</i>
X10	Bool	%M10.0	Estado etapa 10
X11	Bool	%M10.1	Estado etapa 11
X12	Bool	%M10.2	Estado etapa 12
X13	Bool	%M10.3	Estado etapa 13
X14	Bool	%M10.4	Estado etapa 14
X15	Bool	%M10.5	Estado etapa 15
X16	Bool	%M10.6	Estado etapa 16
X17	Bool	%M10.7	Estado etapa 17
S10	Bool	%M12.0	Activación etapa 10
S11	Bool	%M12.1	Activación etapa 11
S12	Bool	%M12.2	Activación etapa 12
S13	Bool	%M12.3	Activación etapa 13
S14	Bool	%M12.4	Activación etapa 14
S15	Bool	%M12.5	Activación etapa 15
S16	Bool	%M12.6	Activación etapa 16
S17	Bool	%M12.7	Activación etapa 17
PARO	Bool	%M14.0	Transición de sistema parado
MANUAL	Bool	%M14.1	Variable que activa el modo manual
AUTOMATICO	Bool	%M14.2	Variable que activa el modo automático
INICIO	Bool	%M14.3	Variable que activa la inicialización
MARCHA	Bool	%M14.4	Variable que inicia la producción automática
Siguiente	Bool	%M14.5	Variable asignada al botón Siguiente
Convertir Tiempos	Bool	%M14.6	Variable que activa el programa "Conversión WORD a TIME"
Flanco X15	Bool	%M15.1	Variable asignada al flanco de subida de X15
Flanco X17	Bool	%M15.2	Variable asignada al flanco de subida de X17
PARAR	Bool	%M15.3	Variable para indicar que se quiere parar el sistema
RESET	Bool	%M15.4	Variable que indica que las variables de emergencia están reseteadas
Visualizador Automático	Bool	%M15.6	Variable que indica que el sistema se encuentra en el modo automático
Visualizador Manual	Bool	%M15.7	Variable que indica que el sistema se encuentra en el modo manual
Aviso Reset	Bool	%M16.1	Variable que indica a la HMI que se tiene que se tienen que resetear las variables de emergencia



1.2.3. Estación multiproceso

1.2.3.1. Alimentador

<i>Nombre</i>	<i>Tipo</i>	<i>Dirección</i>	<i>Descripción</i>
X20	Bool	%M20.0	Estado etapa 20
X21	Bool	%M20.1	Estado etapa 21
X22	Bool	%M20.2	Estado etapa 22
X23	Bool	%M20.3	Estado etapa 23
X24	Bool	%M20.4	Estado etapa 24
S20	Bool	%M22.0	Activación etapa 20
S21	Bool	%M22.1	Activación etapa 21
S22	Bool	%M22.2	Activación etapa 22
S23	Bool	%M22.3	Activación etapa 23
S24	Bool	%M22.4	Activación etapa 24
FIN HORNO	Bool	%M22.7	Variable que indica que finaliza el proceso de hornear
ET_ Alimentador	Time	%MD200	Variable para almacenar el tiempo transcurrido en los bloques TON del sector Alimentador
XI20	Bool	%M24.0	Estado etapa I20
XI21	Bool	%M24.1	Estado etapa I21
SI20	Bool	%M26.0	Activación etapa I20
SI21	Bool	%M26.1	Activación etapa I21
X82	Bool	%M23.0	Activación etapa 82
Retardo Entrada	Time	%MD205	Variable temporal del retardo de entrada del alimentador.
Retardo Salida	Time	%MD210	Variable temporal del retardo de salida del Alimentador.
Retardo Inicio	Time	%MD215	Variable temporal del retardo de salida en la inicialización del Alimentador.
Flanco X82	Bool	%M23.1	Variable asignada al flanco de subida de X82



1.2.3.2. Manipulador

Nombre	Tipo	Dirección	Descripción
X30	Bool	%M30.0	Estado etapa 30
X31	Bool	%M30.1	Estado etapa 31
X32	Bool	%M30.2	Estado etapa 32
X33	Bool	%M30.3	Estado etapa 33
X34	Bool	%M30.4	Estado etapa 34
X35	Bool	%M30.5	Estado etapa 35
X36	Bool	%M30.6	Estado etapa 36
X37	Bool	%M30.7	Estado etapa 37
X38	Bool	%M31.0	Estado etapa 38
X39	Bool	%M31.1	Estado etapa 39
S30	Bool	%M32.0	Activación etapa 30
S31	Bool	%M32.1	Activación etapa 31
S32	Bool	%M32.2	Activación etapa 32
S33	Bool	%M32.3	Activación etapa 33
S34	Bool	%M32.4	Activación etapa 34
S35	Bool	%M32.5	Activación etapa 35
S36	Bool	%M32.6	Activación etapa 36
S37	Bool	%M32.7	Activación etapa 37
S38	Bool	%M33.0	Activación etapa 38
S39	Bool	%M33.1	Activación etapa 39
ET_	Time	%MD300	Variable para almacenar el tiempo transcurrido en los bloques TON del sector Manipulador
Manipulador			
XI30	Bool	%M34.0	Estado etapa I30
XI31	Bool	%M34.1	Estado etapa I30
XI32	Bool	%M34.2	Activación etapa I30
SI30	Bool	%M36.0	Activación etapa I30
SI31	Bool	%M36.1	Activación etapa I30
SI32	Bool	%M36.2	Activación etapa I30
Retorno	Time	%MD305	Variable temporal que contiene el tiempo de retorno del alimentador.
Bajada Recogida	Time	%MD310	Variable temporal que contiene el tiempo de bajada en la recogida del Manipulador
Succión Recogida	Time	%MD315	Variable temporal que contiene el tiempo de succión en la recogida del Manipulador
Subida Recogida	Time	%MD320	Variable temporal que contiene el tiempo de subida en la recogida del Manipulador
Bajada Entrega	Time	%MD325	Variable temporal que contiene el tiempo de bajada en la entrega del Manipulador
Despegue Entrega	Time	%MD330	Variable temporal que contiene el tiempo de despegue en la entrega del Manipulador
Subida Entrega	Time	%MD335	Variable temporal que contiene el tiempo de subida en la entrega del Manipulador
Retorno Inicio	Time	%MD340	Variable temporal que contiene el tiempo de retorno del alimentador en la inicialización.



1.2.3.3. Mesa

Nombre	Tipo	Dirección	Descripción
X40	Bool	%M40.0	Estado etapa 40
X41	Bool	%M40.1	Estado etapa 41
X42	Bool	%M40.2	Estado etapa 42
X43	Bool	%M40.3	Estado etapa 43
X44	Bool	%M40.4	Estado etapa 44
X45	Bool	%M40.5	Estado etapa 45
S40	Bool	%M42.0	Activación etapa 40
S41	Bool	%M42.1	Activación etapa 41
S42	Bool	%M42.2	Activación etapa 42
S43	Bool	%M42.3	Activación etapa 43
S44	Bool	%M42.4	Activación etapa 44
S45	Bool	%M42.5	Activación etapa 45
ET_Mesa	Time	%MD400	Variable para almacenar el tiempo transcurrido en los bloques TON del sector Mesa
XI40	Bool	%M44.0	Estado etapa I40
XI41	Bool	%M44.1	Estado etapa I41
SI40	Bool	%M46.0	Activación etapa I40
SI41	Bool	%M46.1	Activación etapa I41
Serrado	Time	%MD405	Variable temporal que contiene el tiempo de serrado
Empuje	Time	%MD410	Variable temporal que contiene el tiempo de empuje

1.2.3.4. Cinta

Nombre	Tipo	Dirección	Descripción
X50	Bool	%M50.0	Estado etapa 50
X51	Bool	%M50.1	Estado etapa 51
S50	Bool	%M52.0	Estado etapa I50
S51	Bool	%M52.1	Estado etapa I51



1.2.3.5. Modo manual

<i>Nombre</i>	<i>Tipo</i>	<i>Dirección</i>	<i>Descripción</i>
<i>PQ1</i>	Bool	%M90.0	Variable asignada al pulsador de la HMI que activa <i>Output_Q1</i>
<i>PQ2</i>	Bool	%M90.1	Variable asignada al pulsador de la HMI que activa <i>Output_Q2</i>
<i>PQ3</i>	Bool	%M90.2	Variable asignada al pulsador de la HMI que activa <i>Output_Q3</i>
<i>PQ4</i>	Bool	%M90.3	Variable asignada al pulsador de la HMI que activa <i>Output_Q4</i>
<i>PQ5</i>	Bool	%M90.4	Variable asignada al pulsador de la HMI que activa <i>Output_Q5</i>
<i>PQ6</i>	Bool	%M90.5	Variable asignada al pulsador de la HMI que activa <i>Output_Q6</i>
<i>PQ7</i>	Bool	%M90.6	Variable asignada al pulsador de la HMI que activa <i>Output_Q6</i>
<i>PQ8</i>	Bool	%M90.7	Variable asignada al pulsador de la HMI que activa <i>Output_Q8</i>
<i>PQ9</i>	Bool	%M91.0	Variable asignada al pulsador de la HMI que activa <i>Output_Q1</i>
<i>PQ10</i>	Bool	%M91.1	Variable asignada al pulsador de la HMI que activa <i>Output_Q2</i>
<i>PQ11</i>	Bool	%M91.2	Variable asignada al pulsador de la HMI que activa <i>Output_Q3</i>
<i>PQ12</i>	Bool	%M91.3	Variable asignada al pulsador de la HMI que activa <i>Output_Q4</i>
<i>PQ13</i>	Bool	%M91.4	Variable asignada al pulsador de la HMI que activa <i>Output_Q5</i>
<i>PQ14</i>	Bool	%M91.5	Variable asignada al pulsador de la HMI que activa <i>Output_Q6</i>
<i>PQ15</i>	Bool	%M91.6	Variable asignada al pulsador de la HMI que activa <i>Output_Q6</i>



2. VARIABLES SCHNEIDER M241

2.1. Entradas y salidas (E/S)

<i>Nombre</i>	<i>Dirección</i>	<i>Descripción</i>
<i>Input_B1</i>	%IX0.0	Final de carrera de la referencia vertical
<i>Input_B3</i>	%IX0.1	Final de carrera de la referencia horizontal
<i>Input_B5</i>	%IX0.2	Final de carrera de la referencia giro
<i>Input_I1</i>	%IX0.4	Pulsos encóder movimiento vertical
<i>Input_I2</i>	%IX0.5	Pulsos encóder movimiento horizontal
<i>Input_I3</i>	%IX0.6	Pulsos encóder movimiento giratorio

<i>Nombre</i>	<i>Dirección</i>	<i>Descripción</i>
<i>Output_Q1</i>	%QX0.4	Motor movimiento vertical hacia arriba
<i>Output_Q2</i>	%QX0.5	Motor movimiento vertical hacia abajo
<i>Output_Q3</i>	%QX0.6	Motor movimiento retroceso horizontal
<i>Output_Q4</i>	%QX0.7	Motor movimiento avance horizontal
<i>Output_Q5</i>	%QX1.0	Motor movimiento giro horario
<i>Output_Q6</i>	%QX1.1	Motor movimiento giro antihorario
<i>Output_Q7</i>	%QX1.2	Habilitar compresor
<i>Output_Q8</i>	%QX1.3	Succión ventosa
<i>Output_Q9</i>	%QX1.4	Habilitar señales de entrada del proceso



2.2. Registros

2.2.1. Brazo Ventosa

<i>Nombre</i>	<i>Tipo</i>	<i>Dirección</i>	<i>Descripción</i>
<i>CuentaVertical</i>	Int	%MW20	Variable que almacena los pulsos del encóder “vertical”
<i>CuentaHorizontal</i>	Int	%MW25	Variable que almacena los pulsos del encóder “horizontal”
<i>CuentaGiro</i>	Int	%MW30	Variable que almacena los pulsos del encóder “giro”
<i>MaxCuentaVertical</i>	Int		Límite del movimiento vertical expresado en pulsos
<i>MaxCuentaHorizontal</i>	Int		Límite del movimiento horizontal expresado en pulsos
<i>MaxCuentaGiro</i>	Int		Límite del movimiento giratorio expresado en pulsos
<i>AvisoVertical</i>	Bool	%MX2.0	Se activa cuando se alcanza el límite de pulsos vertical
<i>AvisoHorizontal</i>	Bool	%MX2.1	Se activa cuando se alcanza el límite de pulsos horizontal
<i>AvisoGiro</i>	Bool	%MX2.2	Se activa cuando se alcanza el límite de pulsos giro
<i>PulsosVerticales</i>	Int		Variable que contiene los pulsos verticales introducidos por el asignador
<i>PulsosHorizontales</i>	Int		Variable que contiene los pulsos horizontales introducidos por el asignador
<i>PulsosGiro</i>	Int		Variable que contiene los pulsos verticales introducidos por el asignador
<i>T_Succion</i>	Time		Variable temporal que contiene el tiempo de succión
<i>T_Despegue</i>	Time		Variable temporal que contiene el tiempo de despegue

2.2.2. Configuración cajas

<i>Nombre</i>	<i>Tipo</i>	<i>Dirección</i>	<i>Descripción</i>
<i>B_CE2x2</i>	Bool	%MX0.0	Variable que activará la configuración CE2x2
<i>B_CE3x3</i>	Bool	%MX0.1	Variable que activará la configuración CE3x3
<i>B_CS2x2</i>	Bool	%MX0.2	Variable que activará la configuración CS2x2
<i>B_CS3x3</i>	Bool	%MX0.3	Variable que activará la configuración CS3x3
<i>CE2x2</i>	Bool	%MX0.4	Variable de estado configuración CE2x2
<i>CE3x3</i>	Bool	%MX0.5	Variable de estado configuración CE3x3
<i>CS2x2</i>	Bool	%MX0.6	Variable de estado configuración CS2x2
<i>CS3x3</i>	Bool	%MX0.7	Variable de estado configuración CS3x3
<i>npe</i>	Int		Número de pieza de entrada a recoger
<i>nps</i>	Int		Número de pieza de salida a entregar
<i>ce</i>	Int	%MW45	Cajas de entrada procesadas
<i>cs</i>	Int	%MW50	Cajas de salida producidas
<i>MaxCaja</i>	Int		Número máximo de piezas en una caja



2.2.3. Gestión paro

Nombre	Tipo	Dirección	Descripción
entrada	Int		Variable que almacena el valor de piezas introducidas
salida	Int	%MW55	Variable que almacena el valor de las piezas producidas
diferencia	Int		Variable que almacena el valor de la diferencia entre <i>entrada</i> y <i>salida</i>
PARANDO	Bool		Variable de aviso para no introducir más piezas

2.2.4. Modo manual

Nombre	Tipo	Dirección	Descripción
PQ1	Bool	%MX1.0	Variable asignada al pulsador de la HMI que activa <i>Output_Q1</i>
PQ2	Bool	%MX1.1	Variable asignada al pulsador de la HMI que activa <i>Output_Q2</i>
PQ3	Bool	%MX1.2	Variable asignada al pulsador de la HMI que activa <i>Output_Q3</i>
PQ4	Bool	%MX1.3	Variable asignada al pulsador de la HMI que activa <i>Output_Q4</i>
PQ5	Bool	%MX1.4	Variable asignada al pulsador de la HMI que activa <i>Output_Q5</i>
PQ6	Bool	%MX1.5	Variable asignada al pulsador de la HMI que activa <i>Output_Q6</i>
PQ8	Bool	%MX1.6	Variable asignada al pulsador de la HMI que activa <i>Output_Q8</i>

2.2.5. Matrices de coordenadas

2.2.5.1. CE2x2

	npe=1	npe=2	npe=3	npe=4	npe=5	npe=6	npe=7	npe=8
Pulsos Verticales	700	700	700	700	650	650	650	650
Pulsos Horizontales	400	405	420	425	400	405	420	425
Pulsos Giro	700	690	700	690	700	690	700	690

2.2.5.2. CE3x3

	npe=1	npe=2	npe=3	npe=4	npe=5	npe=6	npe=7	npe=8
Pulsos Verticales	700	700	700	700	700	700	700	700
Pulsos Horizontales	400	405	410	420	425	430	440	445
Pulsos Giro	700	690	680	700	690	680	700	690

	npe=9	npe=10	npe=11	npe=12	npe=13	npe=14	npe=15	npe=16
Pulsos Verticales	700	650	650	650	650	650	650	650
Pulsos Horizontales	450	400	405	410	420	425	430	440
Pulsos Giro	680	700	690	680	700	690	680	700



	<i>npe=17</i>	<i>npe=18</i>
<i>Pulsos Verticales</i>	650	650
<i>Pulsos Horizontales</i>	445	450
<i>Pulsos Giro</i>	690	680

2.2.5.3. CS2x2

	<i>npe=1</i>	<i>npe=2</i>	<i>npe=3</i>	<i>npe=4</i>	<i>npe=5</i>	<i>npe=6</i>	<i>npe=7</i>	<i>npe=8</i>
<i>Pulsos Verticales</i>	700	700	700	700	650	650	650	650
<i>Pulsos Horizontales</i>	100	105	120	125	100	105	120	115
<i>Pulsos Giro</i>	30	20	30	20	30	20	30	20

2.2.5.4. CS3x3

	<i>npe=1</i>	<i>npe=2</i>	<i>npe=3</i>	<i>npe=4</i>	<i>npe=5</i>	<i>npe=6</i>	<i>npe=7</i>	<i>npe=8</i>
<i>Pulsos Verticales</i>	700	700	700	700	700	700	700	700
<i>Pulsos Horizontales</i>	100	105	110	120	125	130	140	145
<i>Pulsos Giro</i>	30	20	10	30	20	10	30	20

	<i>npe=9</i>	<i>npe=10</i>	<i>npe=11</i>	<i>npe=12</i>	<i>npe=13</i>	<i>npe=14</i>	<i>npe=15</i>	<i>npe=16</i>
<i>Pulsos Verticales</i>	700	650	650	650	650	650	650	650
<i>Pulsos Horizontales</i>	150	100	105	110	120	125	130	140
<i>Pulsos Giro</i>	10	30	20	10	30	20	10	30

	<i>npe=17</i>	<i>npe=18</i>
<i>Pulsos Verticales</i>	650	650
<i>Pulsos Horizontales</i>	20	10
<i>Pulsos Giro</i>	700	690



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





MANUAL DE USUARIO

OMRON NB7W

JUAN CARLOS GARCÍA HERNÁNDEZ



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





Índice

1. DESCRIPCIÓN Y FUNCIONALIDADES DE LAS PANTALLAS PRINCIPALES	97
1.1. Pantalla <i>Inicio</i>	97
1.1.1. Funcionamiento modo automático.....	98
1.2. Pantalla <i>Información de la Producción</i>	99
1.2.1. Detención de errores.....	100
1.3. Pantalla <i>Modo manual</i>	101
1.3.1. Funcionamiento en Modo manual.....	102
1.4. Pantalla <i>Configuración</i>	103
1.5. Pantalla <i>Tiempos</i>	104
1.5.1. Configuración de una variable temporal.....	105
1.6. Pantalla <i>Configuración cajas</i>	105
1.7. Pantalla <i>Log In</i>	106
1.7.1. Inicio de sesión	107
2. EMERGENCIA.....	108
3. PARADA DEL SISTEMA	111



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.



1. DESCRIPCIÓN Y FUNCIONALIDADES DE LAS PANTALLAS PRINCIPALES

1.1. Pantalla *Inicio*



Figura 1.1. Pantalla Inicio

La pantalla *Inicio*, mostrada en la *Figura 1.1*, es la más importante de la HMI, ya que, permite acceder a todas las pantallas del sistema, cambiar entre el modo automático y manual y manipular las variables que servirán para interactuar con el sistema cuando el modo automático esté activado.

Será la primera en aparecer cuando se conecte la HMI, y para poder interactuar con los distintos elementos, se deberá iniciar sesión (véase *subapartado 1.7.1*).

A continuación, en la *Tabla 1.1*, se muestran los distintos elementos separados por las regiones delimitadas en la pantalla y la acción que realiza cada uno.



Región	Elemento	Acción
1	Botón Automático	Seleccionar modo automático
	Botón Manual	Seleccionar modo manual
2	Botón Marcha	Iniciar funcionamiento automático
	Botón Paro	Parar funcionamiento automático
	Botón Inicio	Inicializar el proceso
3	FK Producción	Mostrar la pantalla <i>Información de la Producción</i>
	FK Manual	Mostrar la pantalla <i>Modo manual</i>
	FK Ajustes	Mostrar la pantalla <i>Configuración</i>
4	FK Log In	Mostrar la pantalla <i>Log in</i>
5	Reloj	Informa de la fecha y hora actual

Tabla 1.1. Elementos contenidos en la pantalla Inicio

1.1.1.1. Funcionamiento modo automático

Como se ha comentado anteriormente, la pantalla *Inicio* nos proporciona control sobre el modo automático. Para iniciar la producción automática hay que seguir los siguientes pasos:

- 1)** Seleccionar el modo automático pulsando el “*Botón Automático*”.
Nota: Podría darse el caso de que no se permita activar el modo automático debido a que algunas variables se encuentran activadas (véase *capítulo 2*).
- 2)** Una vez seleccionado el modo automático el “*Botón Inicio*” parpadeará indicando que se tiene que pulsar para inicializar el sistema.
- 3)** Una vez pulsado el “*Botón Inicio*”, se podrá observar cómo el sistema se ha inicializado, y el “*Botón Marcha*” parpadeará indicando que ya se puede empezar la producción automática.
- 4)** Al pulsar el “*Botón Marcha*”, el sistema se encontrará en el modo de producción automática y no finalizará hasta pulsar el “*Botón Paro*” el cual se iluminará indicando que la opción de paro se ha enviado correctamente.
- 5)** El sistema parará cuando la caja de salida se llene y se mostrarán las pantallas del modo paro las cuales se explicarán en el *capítulo 3*.

1.2. Pantalla *Información de la Producción*



Figura 1.2. Pantalla Información de la Producción

La presente pantalla, mostrada en la *Figura 1.2*, dota a la HMI de una tecnología SCADA para obtener información actualizada de la producción.

En la *Tabla 1.2*, se muestran los elementos que aparecen en la pantalla.

Región	Elemento	Acción
1	Gráfico	Muestra la información de la producción actualizada en un gráfico
2	V.N cajas consumidas	Informa de las cajas consumidas
	V.N. cajas producidas	Informa de las cajas producidas
	V.N. piezas producidas	Informa de las piezas producidas

Tabla 1.2. Elementos contenidos en la pantalla Información de la Producción



1.2.1. Detención de errores

La pantalla *Información de la Producción* ofrece la posibilidad de detectar si existe algún error en el sistema gracias al “Gráfico” descrito en la *Tabla 1.2*.

La detección de un error de producción se observaría en un decrecimiento o de la pendiente de la gráfica, para ello, el operario debe conocer con anterioridad la pendiente que tendría dicha gráfica en un funcionamiento automático sin errores.

1.3. Pantalla *Modo manual*



Figura 1.3. Pantalla Modo manual

La pantalla *Modo manual*, mostrada en la *Figura 1.3*, será la encargada de mostrar las pantallas de los modos manuales de los distintos sectores del sistema, los cuales se describirán en el *subapartado 1.3.1*.

En la *Tabla 1.3* se muestran los elementos que aparecen en la pantalla.

Región	Elemento	Acción
1	Ventana indirecta	Representa el modo manual de los distintos sectores
2	Botón Alimentador	Muestra el modo manual del alimentador
	Botón Manipulador	Muestra el modo manual del manipulador
	Botón Mesa	Muestra el modo manual de la mesa
	Botón Cinta	Muestra el modo manual de la cinta
	Botón Brazo	Muestra el modo manual del brazo

Tabla 1.3. Elementos contenidos en la pantalla Modo manual

1.3.1. Funcionamiento en Modo manual

- 1) Seleccionar el modo manual mediante el “Botón Manual”.
- 2) Pulsar el “FK (Function key) Manual” para acceder a la pantalla *Modo manual*.

Al seguir los dos pasos anteriores aparecerá la pantalla *Modo manual* y se tendrá acceso a todos los botones que permiten activar manualmente las variables del sistema las cuales aparecen resumidas en la *Figura 1.4*.

	<p align="center">Alimentador</p> <p>Adentro : Introducir alimentador en el horno Afuera : Extraer alimentador del horno Puerta : Subir puerta Horno : Activar horno</p>
	<p align="center">Manipulador</p> <p>Arriba : Subir el manipulador Abajo : Bajar el manipulador Ventosa : Activar la ventosa Horno : Desplazar el alimentador hacia el horno Mesa : Desplazar el alimentador hacia la mesa</p>
	<p align="center">Mesa</p> <p>Antihorario : Activar movimiento antihorario Horario : Activar movimiento horario Empuje : Activar empujador Sierra : Activar motor sierra</p>
	<p align="center">Cinta</p> <p>Avance : Activar avance cinta</p>
	<p align="center">Brazo Ventosa</p> <p>Vert- / Vert+ : Subir y bajar verticalmente Hor- / Hor+ : Extraer y recoger horizontalmente Giro- / Giro+ : Girar brazo ventosa Ventosa : Activar ventosa</p>

Figura 1.4. Botones y sus acciones de los distintos sectores en el modo manual

1.4. Pantalla Configuración

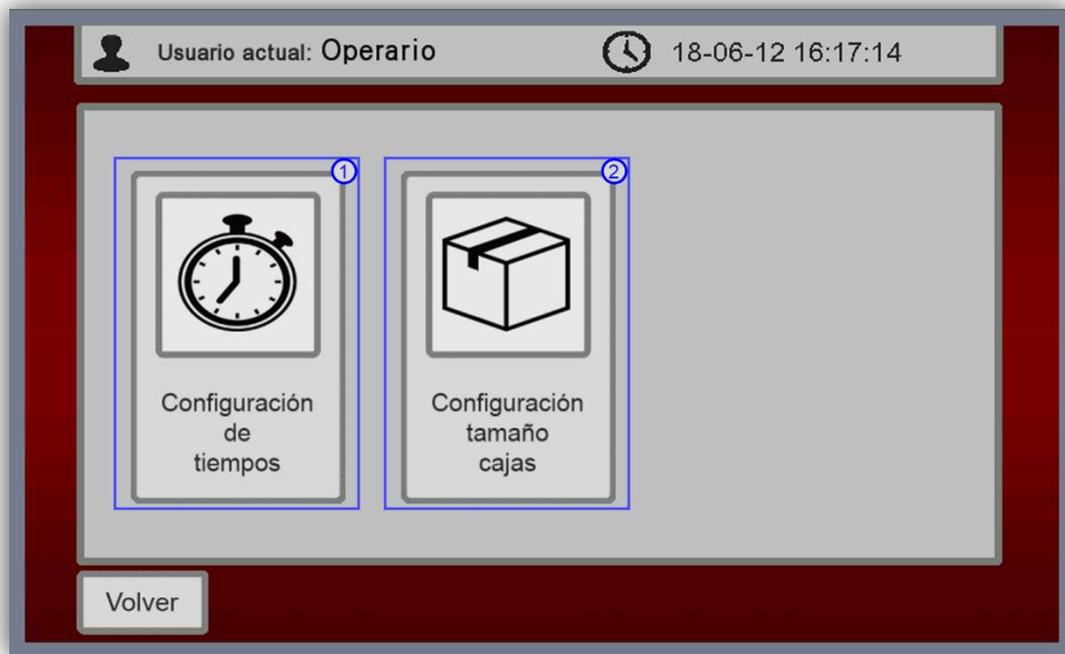


Figura 1.5. Pantalla “Configuración”

La pantalla *Configuración*, mostrada en la *Figura 1.5* es la encargada de proporcionar el acceso a las pantallas *Tiempos* y *Configuración cajas* las cuales son las encargadas de modificar los parámetros que afectan al funcionamiento del sistema.

A continuación, en la *Tabla 1.4* se muestran los elementos que aparecen en la pantalla.

<i>Región</i>	<i>Elemento</i>	<i>Acción</i>
1	FK tiempos	Muestra la pantalla <i>Tiempos</i>
2	FK cajas	Muestra la pantalla <i>Configuración cajas</i>

Tabla 1.4. Elementos contenidos en la pantalla *Configuración*

1.5. Pantalla *Tiempos*

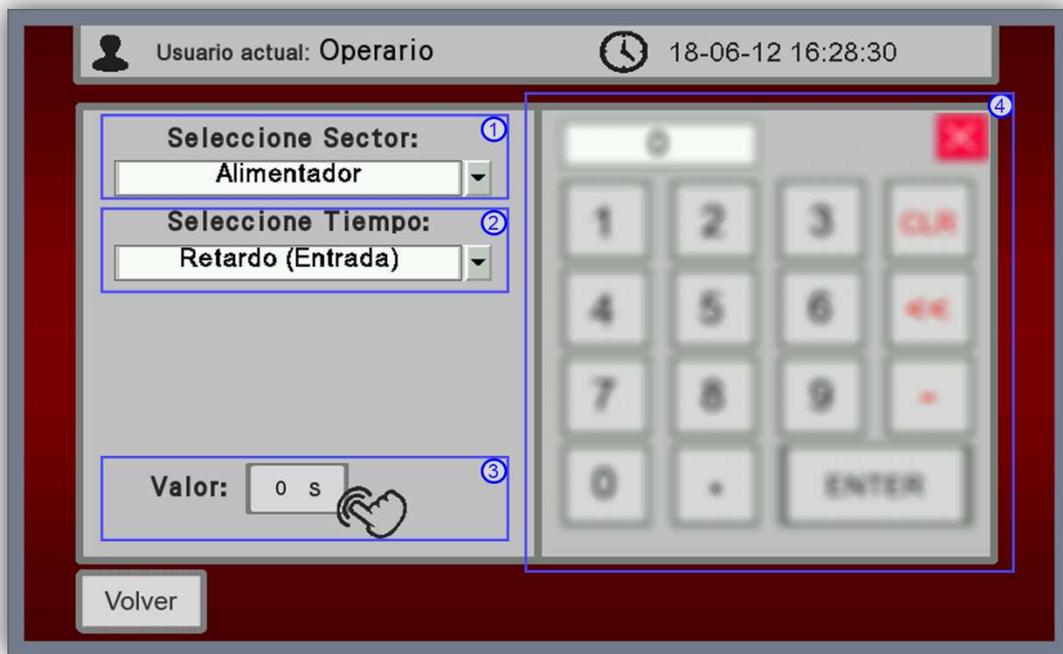


Figura 1.6. Pantalla *Tiempos*

La pantalla *Tiempos*, mostrada en la Figura 1.6 es la encargada de mostrar los elementos necesarios para poder modificar el valor de las variables temporales que aparecen en el sistema.

A continuación, en la Tabla 1.5, se muestran los elementos que aparecen en la pantalla.

<i>Región</i>	<i>Elemento</i>	<i>Acción</i>
1	Lista de sectores	Muestra los sectores en forma de lista
2	Lista de tiempos	Muestra los tiempos del sector elegido
3	Botón tiempos	Activa el teclado numérico
4	Teclado numérico	Introducción del valor del tiempo modificado

Tabla 1.5. Elementos contenidos en la pantalla *Tiempos*

1.5.1. Configuración de una variable temporal.

Para configurar una variable temporal, se deben seguir los siguientes pasos:

- 1) Iniciar como ingeniero para tener los permisos necesarios.
- 2) Pulsar “FK Configuración” para acceder a la pantalla *Configuración* y posteriormente pulsar “FK tiempos”.
- 3) Seleccionar el sector y el tiempo a modificar mediante la “Lista de sectores” y la “Lista de tiempos”.
- 4) Pulsar el “Botón tiempos” para desenfocar el “Teclado numérico”.
- 5) Introducir el valor deseado y pulsar *ENTER*.

1.6. Pantalla *Configuración cajas*

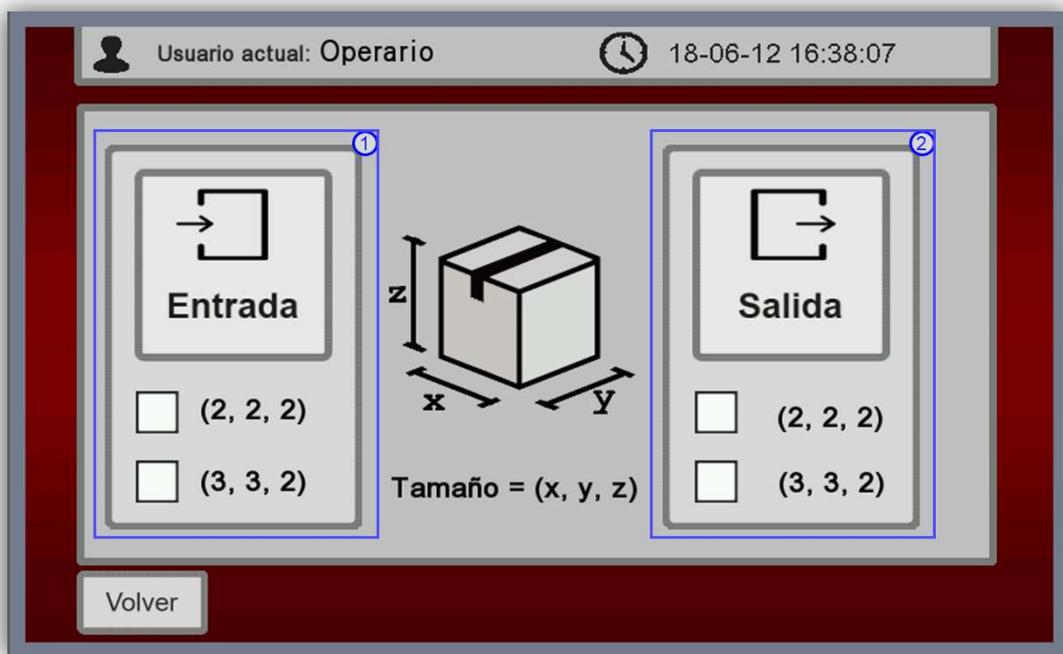


Figura 1.7. Pantalla *Configuración cajas*

La pantalla *Configuración cajas*, mostrada en la *Figura 1.7* será la encargada de mostrar los botones y lamparas necesarias para cambiar la configuración y saber qué configuración tenemos seleccionada actualmente, el funcionamiento es tan intuitivo y simple que se ha obviado una explicación detallada de la misma, ya que solo bastará con pulsar el botón de las dos configuraciones disponibles.

En la *Tabla 1.6*, se muestran los elementos que aparecen en la pantalla.

<i>Región</i>	<i>Elemento</i>	<i>Acción</i>
1	Botón (2,2,2) entrada	Activa la configuración (2,2,2) para las cajas de entrada
	Botón (3,3,2) entrada	Activa la configuración (3,3,2) para las cajas de entrada
2	Botón (2,2,2) salida	Activa la configuración (2,2,2) para las cajas de salida
	Botón (3,3,2) salida	Activa la configuración (3,3,2) para las cajas de salida

Tabla 1.6. Elementos contenidos en la pantalla Configuración cajas

1.7. Pantalla Log In



Figura 1.9. Pantalla Log In

La pantalla de *Log In*, mostrada en la *Figura 1.9*, incorpora los elementos necesarios para poder realizar la función de iniciar sesión. En la *Tabla 1.7*, se muestran los elementos que aparecen en la pantalla.



Región	Elemento	Acción
1	I.T Usuario	Al pulsarlo, aparecerá el teclado alfanumérico para introducir el nombre de usuario.
	I.T. Contraseña	Al pulsarlo, aparecerá el teclado alfanumérico para introducir la contraseña de usuario
	Botón Log In	Activa la acción de Log In
	Botón Log Out	Activa la acción de Log Out
2	Teclado alfanumérico	Introducción de los caracteres que pulse el usuario.

Tabla 1.7. Elementos contenidos en la pantalla Log In

1.7.1. Inicio de sesión

Para realizar un inicio de sesión, se tendrán que seguir los siguientes pasos:

- 1)** Pulsar “FK Log In” el cual mostrará la pantalla *Log In*.
- 2)** Pulsar el “I.T. (introducción de texto) usuario” para que aparezca el teclado alfanumérico e introducir el nombre de usuario, y, una vez introducido, pulsar *ENTER*.
- 3)** Pulsar el “I.T. contraseña” para que aparezca el teclado alfanumérico y procediendo como en el punto anterior, introducir la contraseña
- 4)** Una vez rellenado todos los campos, apretar el “Botón Log In” para iniciar la sesión.

Si todo ha ido correctamente aparecerá el nombre de usuario actual en la *región 4* de la pantalla *Inicio*.

El procedimiento para cerrar sesión es simple, consiste en apretar el “Botón Log Out” tras acceder a la pantalla *Log In*.



2. EMERGENCIA

Se ha incorporado un sistema de gestión de alarmas a la HMI el cual se representará cuando se produzca una emergencia, es decir, se active una alarma.

La encargada de gestionar las alarmas se denomina pantalla *Emergencia*.

La pantalla *Emergencia* emergerá automáticamente cuando se detecte una alarma en el sistema. Ésta consta de dos estados claramente diferenciados los cuales se exponen a continuación.

Estado 1, alarmas activadas:

La apariencia de la pantalla “Emergencia” en esta situación se muestra en la *Figura 2.1*.

En esta situación la pantalla muestra las alarmas activas, y se restringe completamente la decisión de seguir produciendo, o de parar la producción.

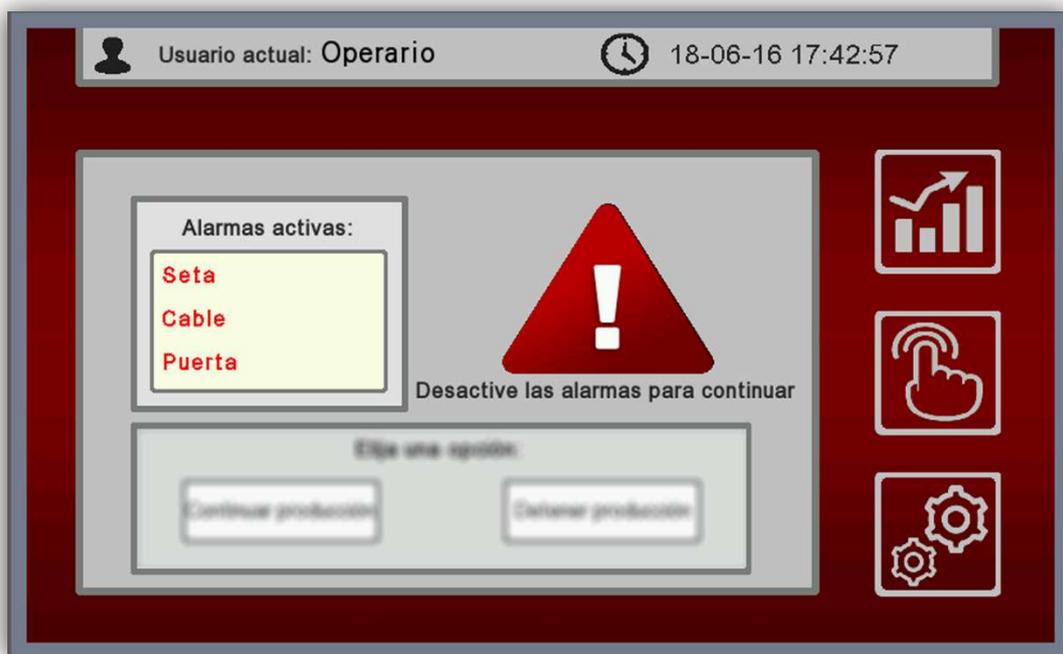


Figura 2.1. Pantalla Emergencia en estado 1, alarmas activadas

Estado 2, alarmas desactivadas:

La apariencia de la pantalla *Emergencia* en este estado se muestra en la *Figura 2.2*.

Como se puede observar, no aparecerá ninguna alarma activa y el icono de peligro se habrá puesto en verde, lo que significará que se han rearmado todas las alarmas del sistema.

En esta situación la pantalla desenfoca el submenú para elegir si se quiere continuar producción, o detenerla.

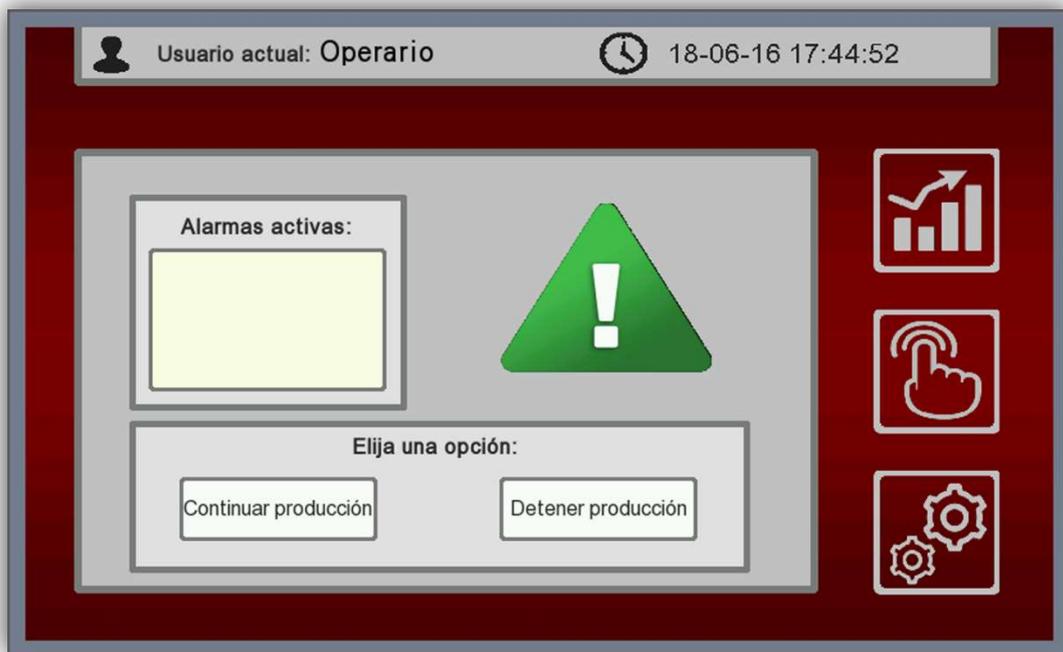


Figura 2.2. Pantalla Emergencia en estado 2, alarmas desactivadas

Por lo tanto, en esta situación se tienen dos caminos diferenciados, los cuales serían:

- Continuar producción: Si el ingeniero eligiese esta opción, será porque habrá decidido que a pesar de haberse producido al menos una alarma, el sistema no ha sido dañado y por lo tanto se podrá seguir produciendo con total normalidad.
- Detener producción: Si por el contrario el ingeniero eligiese esta opción, será porque habrá considerado que el fallo producido ha dañado el sistema y por lo tanto no se podrá seguir produciendo. Al pulsar esta opción se forzarán las identificadas como variables de seguridad, las cuales se muestran en la *Tabla 2.1*, y se entrará automáticamente en el modo manual.

Nota: No será posible cambiar al modo automático a no ser que se desactiven las variables de seguridad, por lo tanto, el ingeniero deberá entrar en el modo manual (véase *subapartado 1.3.1*), y desactivarlas.



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.



Variable	Motivo del forzado
Ventosa Brazo	Evitar la caída de piezas que pudieran estar agarradas
Ventosa Manipulador	Evitar la caída de piezas que pudieran estar agarradas
Puerta Horno	Evitar colisión de puerta con el <i>Alimentador</i>
Bajada Manipulador	Evitar daños superiores si la alarma proviene de éste



3. PARADA DEL SISTEMA

El comportamiento de la HMI cuando se produce una parada del sistema se basa en mostrar dos pantallas que vienen explicadas a continuación.

Se recuerda que el sistema se parará cuando el operario pulse el “Botón Paro”, y se cumpla la condición de que la caja de salida se haya llenado y ya no queden piezas por procesar.

Pantalla2 Resultados

La pantalla se muestra en la *Figura 3.1*.

Muestra un resumen de los resultados obtenidos en el funcionamiento automático e incorpora un botón el cual se pulsará para acceder a la siguiente pantalla.

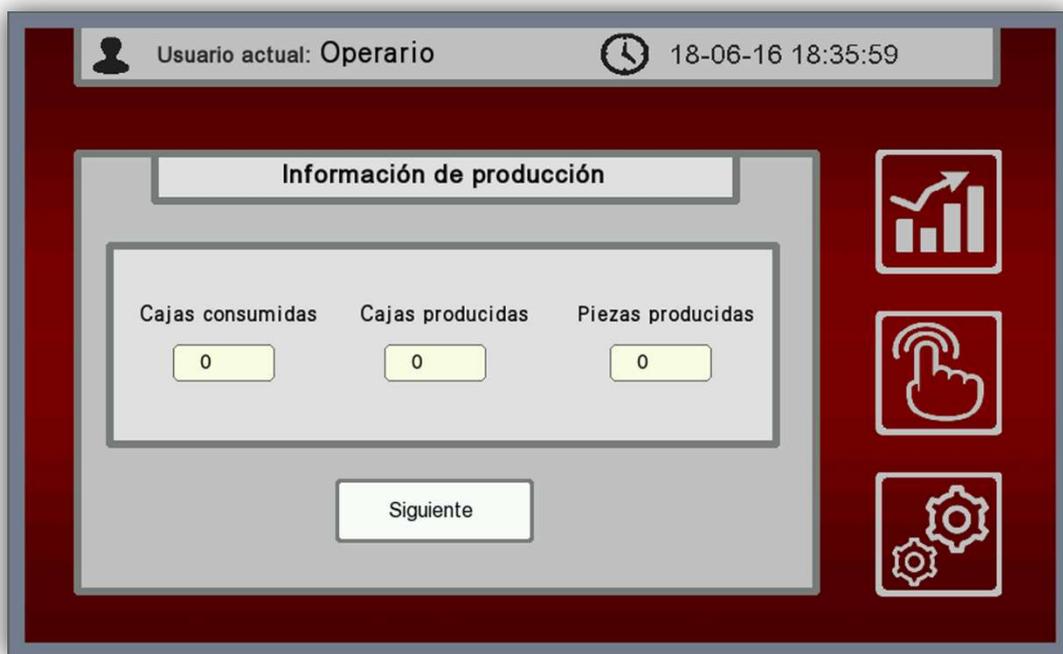


Figura 3.1. Pantalla Resultados



Pantalla Opciones del modo paro

La pantalla se muestra en la *Figura 3.2*.

La pantalla contiene dos botones que se pulsarán para acceder al modo que se desee oportuno tras haber parado el sistema.

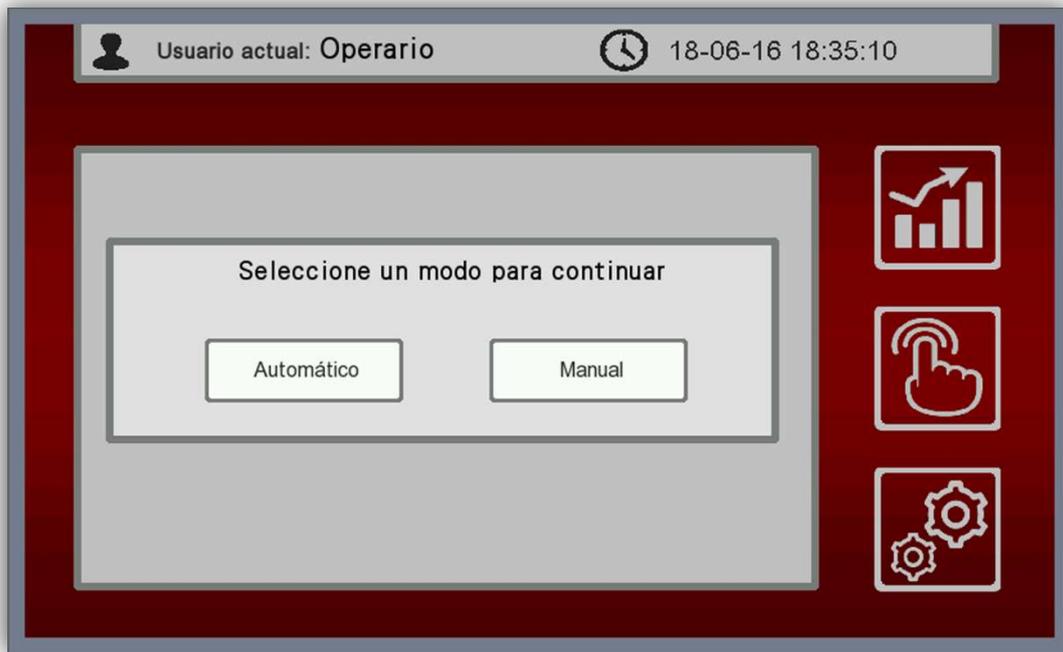


Figura 3.2. Pantalla de "Opciones" del modo paro



PLANOS

JUAN CARLOS GARCÍA HERNÁNDEZ



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.





Índice

1. PLANO DISEÑO DEL SISTEMA AUTOMÁTICO.....	117
--	------------



Proyecto de automatización de una estación multiproceso y un brazo manipulador mediante autómatas Siemens S7-1200 y Schneider M241 con monitorización e interacción a través de una HMI Omron.



