



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Curso Académico:

RESUMEN

En el trabajo de fin de grado “PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON HORNO MEDIANTE AUTÓMATA SIEMENS S7 1214C Y SCADA EN WINCC” se desarrolla la automatización de un sistema de procesado de piezas mediante un autómata programable. También se realiza un SCADA con el que se supervisa dicho sistema y se observa de manera instantánea el desarrollo del proceso que se lleva a cabo.

Para automatizar el sistema se han utilizado herramientas de programación avanzadas, permitiendo el uso de varios modos de funcionamiento: manual y automático.

El uso manual permite un control centralizado, desde el ordenador, a través de un panel de control. En cuanto al uso automático, se ha programado de manera que el proceso solo requiera de la alimentación y extracción de piezas externamente.

Se ha dotado al proceso de mecanismos de seguridad que mejoran la programación y la hacen más segura.

La programación se ha realizado de manera estructurada, permitiendo cambiar de modo de funcionamiento sin dificultad.

Para la interacción entre usuario y maquina se ha diseñado un SCADA sencillo, muy intuitivo. Este panel dispone de representación visual de toda la estación, así como de todos sus sensores y actuadores, permitiendo saber en todo momento que subproceso de la planta multiproceso se está ejecutando.

Palabras Clave: programación, automatización, SCADA, control, PLC, Siemens, WinCC.

RESUM

En el treball de fi de grau “PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON HORNO MEDIANTE AUTÓMATA SIEMENS S7 1214C Y SCADA EN WINCC” es desenvolupa l'automatització d'un sistema de processat de peces mitjançant un autòmat programable. També es realitza un SCADA amb el que es supervisa el mencionat sistema i s'observa de manera instantània el desenvolupament del procés que es du a terme.

Per automatitzar el sistema s'han utilitzat ferramentes de programació avançades, permetent el us de diversos modes de funcionament: manual i automàtic.

El us manual permet un control centralitzat, des de l'ordinador, a través d'un panel de control. En el que respecta a l'ús automàtic, s'ha programat fent que el procés soles necessite de l'alimentació i extracció de peces externament.

S'ha dotat al procés de mecanismes de seguretat que milloren la programació i la fan més segura.

La programació s'ha realitzat de manera estructurada, permetent canviar de mode de funcionament sense dificultat.

Per a la interacció entre usuari i màquina s'ha dissenyat un SCADA senzill, molt intuïtiu. Aquest panel disposa de representació visual de tota l'estació, així com de tots els seus sensors y actuadors, permetent esbrinar en tot moment quin sub-procés de la planta multiprocessos està executant-se.

Paraules clau: programació, automatització, SCADA, control, PLC, Siemens, WinCC.

ABSTRACT

In the final degree project "PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON HORNO MEDIANTE AUTÓMATA SIEMENS S7 1214C Y SCADA EN WINCC" is developed an automatic piece processing system by means of a programmable automaton. There is also a SCADA which the system is supervised and with the development of the process is observed in an instant.

To automate the system are used an advanced programming tools, allowing the use of two operating modes: manual and automatic.

The manual control allows a centralized control, from the computer, through a control panel. As for the automatic control, it has been programmed that the process only requires the feeding and extraction of pieces externally.

The process has been provided with security mechanisms that improve programming and make it more secure.

The programming has been done in a structured way, allowing change of operation mode without difficulty.

For the interaction between user and machine a very intuitive SCADA has been designed. This visual panel represents the entire station, as well as all its sensors and actuators. The SCADA allows to know which sub-process of the plant is running.

Keywords: programming, automation, SCADA, control, PLC, Siemens, WinCC.

AGRADECIMIENTOS

Quiero agradecer este trabajo a mi familia y sobre todo a mi novia, por aguantarme en mis momentos de desvarío y apoyarme en todo momento.

A mis amigos, porque los ratos a su lado me han dado fuerza para continuar.

Y, por supuesto, a mi tutor y a todo el personal del laboratorio, por la ayuda que me han brindado en cada momento.

A todos, gracias.

ÍNDICE

ÍNDICE DE LA MEMORIA

1. INTRODUCCIÓN GENERAL.....	9
1.1. Introducción	9
1.2. Objetivo del trabajo.....	10
1.3. Presentación del trabajo.....	10
1.4. Organización de la memoria	13
2. CAPÍTULO 1. Programación.....	14
1.1. Definición general de funcionamiento	15
2.2. Detalle de funcionamiento	15
2.2.1. Variables del proceso.....	15
2.2.2. Proceso horno.....	17
2.2.3. Proceso brazo manipulador	18
2.2.4. Proceso mesa giratoria y cinta	19
2.3. Descripción del proceso de Automatización.....	20
2.3.1. Direccionamiento de variables	21
2.3.1.1. Asignación de entradas.....	21
2.3.1.2. Asignación de salidas.....	22
2.3.1.3. Uso de marcas del sistema	22
2.3.1.4. Bloques de Datos (DB)	24
2.3.2. Automatización mediante GRAFCET	25
2.3.3. Conversión de GRAFCET a LADDER (KOP)	29
2.3.4. Modos de funcionamiento.....	31
2.3.4.1. Modo Automático.....	31
2.3.4.2. Modo manual	32
2.4. Medidas de seguridad	34
2.4.1. Devolver proceso al estado inicial	34
2.4.2. Parada de emergencia	34

3. CAPÍTULO 2. El autómata SIEMENS	35
3.1. Características S7-1214C SIEMENS	35
3.2. Software STEP 7 TIA Portal V13	37
3.2.1. Conexión entre TIA Portal y S7-1214C	37
3.2.2. Bloques de programa. Bloque Main, funciones y DB's	38
4. CAPÍTULO 3. Panel de Control SCADA.....	42
4.1. Software WinCC RT Advanced	42
4.2. Descripción del Panel de Control.....	43
4.2.1. Plantilla	44
4.2.2. Imagen Principal	44
4.2.3. Imagen Modo Automático.....	45
4.2.4. Imagen Modo Manual	46
5. CONCLUSIÓN.....	48

ÍNDICE DEL PRESUPUESTO

1.INTRODUCCIÓN	50
2.COSTES MATERIALES	51
3.COSTES MANO DE OBRA	52
4.COSTES GENERALES.....	52
5.MARGEN DE BENEFICIO.....	53
6.COSTE FINAL	53

ÍNDICE DEL PLIEGO DE CONDICIONES

1.INTRODUCCIÓN	55
2.PLIEGO DE CONDICIONES GENERALES	56
3.PLIEGO DE CONDICIONES PARTICULARES.....	56
3.1. CONDICIONES DE ÍNDOLE TÉCNICA	56
3.1.1. Especificaciones de material y equipos	56
3.1.2. Especificaciones de ejecución	57

ÍNDICE ANEXO I

1.SEGMENTO 1	59
2.SEGMENTO 2	61
3.SEGMENTO 3	62
4.SEGMENTO 4	63
5.SEGMENTO 5	64
6.SEGMENTO 6	65
7.SEGMENTO 7	69
8.SEGMENTO 8	70
9.SEGMENTO 9	71
10.SEGMENTO 10.....	72

MEMORIA

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON
HORNO MEDIANTE AUTÓMATA SIEMENS S7-1214C Y SCADA EN
WINCC

Alfonso Serrano Gallego
Tutor: Jose Vicente Salcedo

1. INTRODUCCIÓN GENERAL

1.1. Introducción

En la actualidad la gran mayoría de los procesos industriales son susceptibles de automatizarse y de ser monitorizados para tener un seguimiento de la labor que desempeñan y para comprobar su correcto funcionamiento.

Es posible que esta tendencia creciente nos lleve a pensar que la automatización tenga como objetivo reducir costes de personal, es decir, sustituir trabajadores por maquinaria que realice su mismo trabajo. Nada más lejos de la realidad, automatizar procesos no tiene como finalidad reemplazar mano de obra, pues tiene su fin en la mejora de la calidad y seguridad de los trabajadores y evitando así la mano de obra en tareas que sean extremadamente peligrosas.

Pero la automatización no sería como hoy la conocemos de no ser por unos elementos clave, los autómatas o PLC's. Sin ellos no se tendría un control tan rápido y preciso de los procesos industriales. El continuo avance de esta tecnología nos permite automatizar cada vez sistemas más complejos y complicados.

Otro aspecto relevante es la posibilidad de conexión que tienen los PLC's. Es posible establecer redes donde unos autómatas controlen a otros, que varios controlen un mismo proceso o incluso controlar procesos desde kilómetros de distancia. Es algo común el uso de internet para establecer conexiones de este tipo.

Cabe añadir que la automatización no solo es exclusiva del sector industrial, pero si es el sector donde tiene más sentido aplicarla, ya que optimizar recursos, espacio y costes es algo primordial para garantizar su éxito en el sector.

Es de esta idea de la cual surge la motivación de este trabajo de automatización. Siendo una buena prueba de que automatizar un proceso nos ayuda a mejorar la eficiencia del mismo y así favorecer al sector industrial.

Este proyecto se va a realizar sobre una maqueta de una estación multiproceso con horno, es decir, que tiene un carácter meramente académico, haciendo ver las virtudes y complicaciones que tiene la automatización y como de potente puede llegar a ser. Cabe añadir que el proceso es susceptible de escalarse a un proceso real, con las debidas modificaciones.

Además, se pretende hacer ver que, aparte de mostrar cómo se elabora todo el proceso de automatización, quede constancia de la existencia de herramientas para monitorizar dicho proceso y poder tener todo el control que se desee sobre este. Todo dependerá de la programación y de a qué nivel de control se desee llegar.

1.2. Objetivo del trabajo

Este trabajo se enfoca en realizar la programación de una estación multiproceso con horno *FischerTechnik*, funcionando de forma automática y autónoma utilizando la tecnología de los PLC's, permitiendo un control instantáneo y eficiente de todo el sistema.

Se elaborará también un panel de control por ordenador (SCADA) para monitorizar el proceso y tener información de las variables que lo gobiernan. Este constará de diferentes modos de funcionamiento para tener un manejo más amplio de todo el sistema.

La programación del autómatas que gobierna el proceso se realizará mediante el software de Siemens STEP 7 TIA Portal, este se utiliza para programar la familia de autómatas SIMATIC-S7 de Siemens. Todos los detalles de la programación se verán en el apartado "2.3. Descripción del proceso de automatización" así como las características del software utilizado se explican en "CAPITULO 2".

Por otro lado, la elaboración del SCADA para controlar el proceso se realizará con el software *WinCC RT Advanced*, el cual se utilizará a través del ya mencionado TIA Portal. Para más información ver el apartado "CAPITULO 3. Panel de control SCADA".

Más concretamente, este trabajo trata (teniendo presente que se trata de una maqueta de laboratorio) que las piezas avancen desde dentro del horno hasta una mesa de procesado mediante un brazo manipulador. Una vez procesadas las piezas en la mesa estas se evacuan de la estación por medio de una cinta.

Este proyecto es susceptible de utilizarse en conjunto con otras maquetas de la familia *FischerTechnik*, de forma que todas ellas interconectadas con sus respectivos autómatas que las gestionen formen un proceso aun mayor y más complejo. En este trabajo solamente nos centraremos en la mencionada estación multiproceso con horno, por ser una de las más completas a la hora de programarla.

El panel de control servirá para gestionar los modos de funcionamiento, el modo automático y el modo manual. Con esto se obtiene un sistema más versátil y adaptable a cada requerimiento.

- Modo Automático: el sistema se ejecuta de manera automática, disponiendo de una parada de emergencia para evitar posibles situaciones de peligro.
- Modo Manual: el usuario tendrá todo el control del sistema a través del panel de control. Se podrá verificar el correcto funcionamiento de cada actuador.

1.3. Presentación del trabajo

El trabajo que seguidamente se detallará tiene como finalidad última mostrar al lector lo potente y versátil que puede llegar a ser la automatización de un proceso industrial. Además, se mostrará cómo se puede controlar y monitorizar dicho proceso mediante un pequeño panel de control o SCADA.

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON HORNO MEDIANTE AUTÓMATA SIEMENS S7 1214C Y SCADA EN WINCC

La maqueta de la que se dispone en el laboratorio es una construcción de la marca *FischerTechnik*, marca alemana de kits mecánicos y eléctricos muy utilizados para la educación. La maqueta consta de numerosos sensores y actuadores, además de un pequeño compresor para activar diferentes actuadores que requieren de aire comprimido.

La maqueta está conectada con el PLC mediante un bus paralelo de 25 pines, que vincula las entradas y salidas del autómatas con las diferentes variables de la maqueta del laboratorio.

Cabe mencionar que al tratarse de una maqueta la etapa de potencia que hay del PLC a esta es ínfima o prácticamente inexistente. Si este proyecto se extrapolase a uno real esta etapa no sería despreciable en absoluto, siendo de una crucial importancia.

La automatización se lleva a cabo mediante el ya mencionado PLC SIMATIC-S7 de Siemens, en concreto se trata de un autómatas S7-1214C (de la familia S7-1200). La programación se descarga desde un PC mediante conexión de área de red local, en concreto la red LAN de la Universidad Politécnica de Valencia. Se conectan tanto el PC como el autómatas a dicha red y mediante el software TIA Portal se establece la conexión entre ambos. Para obtener información más detallada de la conexión lean el apartado “3.2.1. Conexión entre TIA Portal y S7-1214C”

Durante el funcionamiento, es el PLC el que gestiona toda la información que le llega del proceso, leyendo la información de los sensores (bits de entrada) y activando los actuadores (bits de salida) en función de la programación realizada. El PLC también se encarga de las instrucciones que le llegan desde el panel de control, teniendo en cuenta los modos de

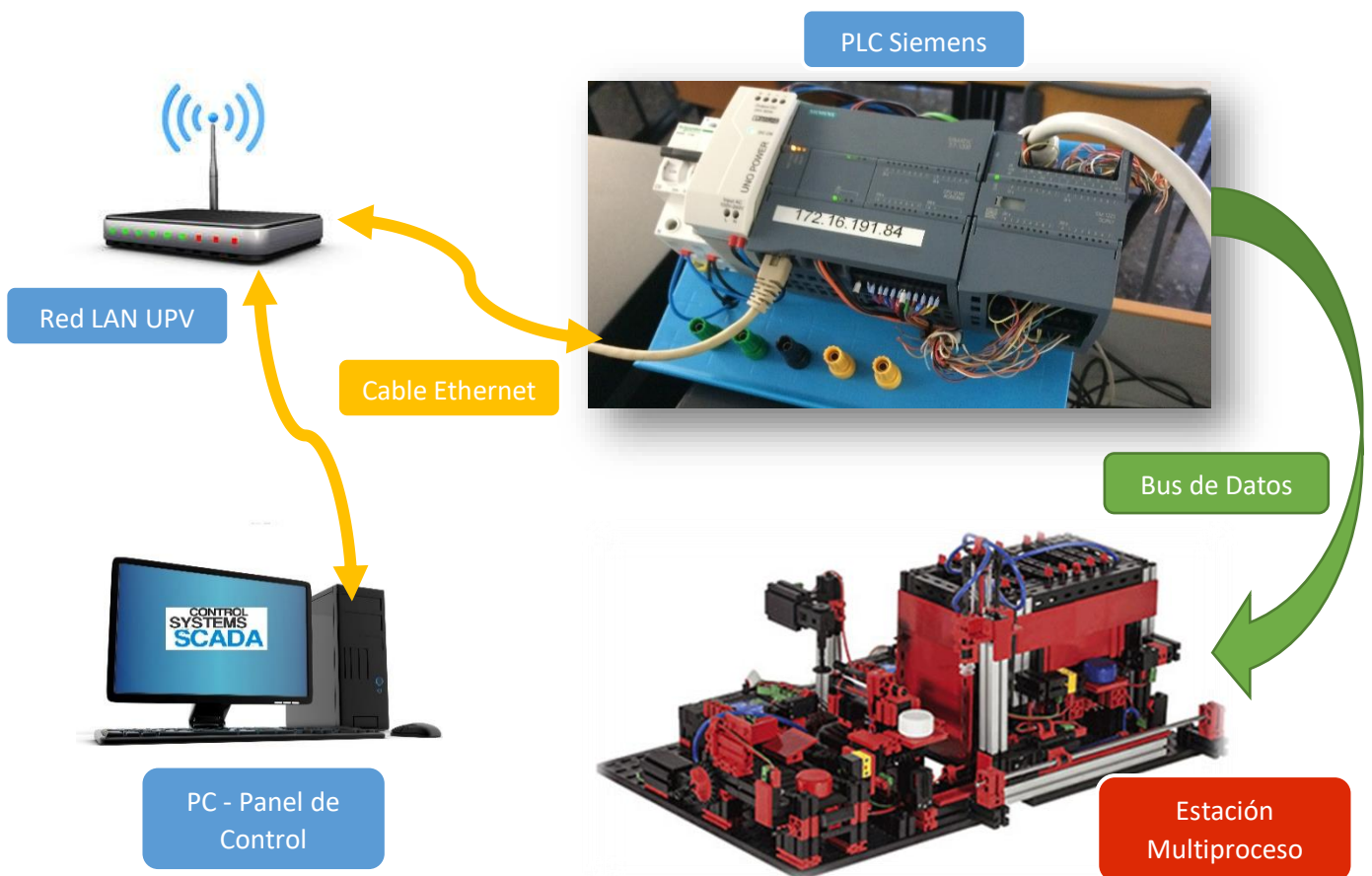


Fig. 1 Esquema conexiones Panel de Control – PLC – Estación Multiproceso

funcionamiento que el usuario quiera activar. Toda esta información es enviada tanto a la maqueta como al SCADA desde el autómatas.

Hemos mencionado ya los diferentes modos de funcionamiento que posee el proceso. Al añadir estos modos se consigue una programación más amplia y versátil, capaz de solventar los diferentes imprevistos del sistema.

- **Modo Automático:** Se trata del modo principal, proceso que se seguirá cuando esté en su funcionamiento normal. Este se activará cuando un usuario u otro proceso deposite una pieza en la alimentación del horno y en el panel de control se active este modo, a partir de este momento todo el proceso se seguirá de manera completamente automática, finalizando cuando se retire la pieza del final de la cinta.
- **Modo Manual:** Se trata de un modo accesible desde el panel de control que permite verificar el correcto funcionamiento de todo el proceso, teniendo acceso a todos los actuadores que tiene la maqueta y disponiendo de un control absoluto de todos ellos.

Estos dos modos de funcionamiento están disponibles únicamente en el SCADA, desde el cual se dispone de una pantalla principal que permite activar solamente uno de los dos. Ambos modos no pueden estar funcionando al mismo tiempo, como era de esperar.

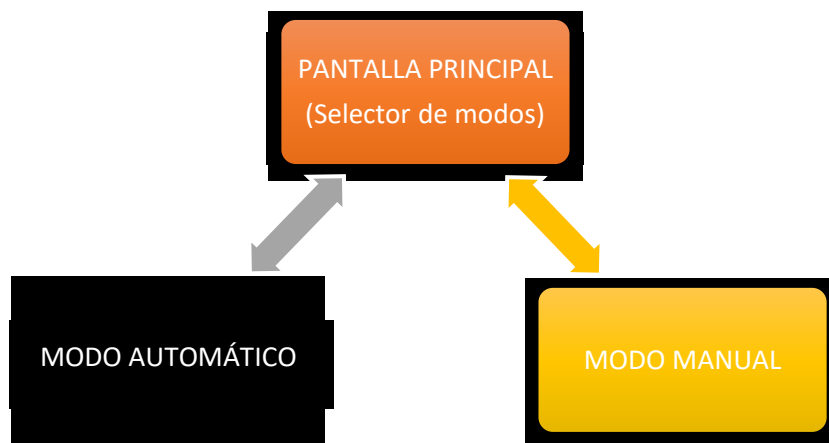


Fig. 2 Esquema control desde SCADA

Como hemos dicho todo este proceso está basado en una maqueta de laboratorio, sin embargo todo el proceso de programación y de generación del panel de control es perfectamente escalable a un proceso real en cualquier industria.

1.4. Organización de la memoria

La memoria del presente trabajo se estructurará en diferentes capítulos, separando en cada uno de ellos diferentes aspectos clave de la realización de este.

Cada capítulo abordará diferentes aspectos del trabajo y los métodos utilizados, así como las herramientas utilizadas y porque se ha optado por elegirlos.

En total la memoria consta de tres capítulos:

- **CAPÍTULO 1: Programación**
Abordaremos las técnicas de programación utilizadas. Se verá con detalle cual es la función de la estación multiproceso y como se ha abordado su programación. Se presentarán todas las variables de entrada/salida, marcas del sistema y bloques de datos necesarias para la programación.
- **CAPÍTULO 2: El autómata Siemens**
Hablaemos de cómo funciona nuestro autómata y que particularidades tiene. Se detallará el software de programación utilizado y sus potenciales ventajas.
- **CAPÍTULO 3: Panel de Control SCADA**
Veremos cómo se ha planteado el panel de control que gestiona al proceso, los elementos que lo componen y se explicará, con gran detalle, la función de cada modo de funcionamiento y todos los contenidos de estos.

Finalmente se expondrán las conclusiones de todo el trabajo realizado y que se puede extraer de este.

2. CAPÍTULO 1. Programación

El propósito principal de la programación es crear programas que cumplan un propósito deseado. Para que pueda ser interpretado y ejecutado, el programa debe estar escrito en un lenguaje de programación.

Existen diversos lenguajes que se utilizan a la hora de programar, cada uno de ellos tienen ciertas ventajas o características que los hacen idóneos para determinadas tareas en particular.

En el entorno de programación de autómatas en el que nos encontramos, el lenguaje de programación viene condicionado por el PLC que dispongamos. Principalmente los autómatas usan uno o varios de los siguientes lenguajes de programación:

- Texto Estructurado (ST)
- Esquema Básico de Funciones (FBD)
- Esquema Secuencial de Funciones (SFC)
- Esquema o Diagrama de Contactos (LD)

En nuestro caso, los lenguajes de programación que hemos utilizado son principalmente dos: diagrama de contactos o *Ladder* (LD) y esquema secuencial de funciones (SFC), también conocido este último como Grafcet.

El lenguaje de programación de diagrama de contactos o *Ladder* (también conocido como lógica de escalera) es uno de los más utilizados para la programación de PLC's. Se basa en los diagramas eléctricos, el motivo es que fue uno de los primeros lenguajes utilizados y surgió como la transformación desde los sistemas de control realizados con relés.

El lenguaje de diagrama Grafcet, o SFC (*Section Function Chart*) es un lenguaje gráfico, que funciona a base de ramificaciones. Se divide en etapas, donde se realizan acciones mientras la programación se encuentre en la etapa asociada. Además, según el programa con el que se realiza la programación, también se puede retrasar el comienzo de una acción en una etapa, retrasar la finalización una vez se ha salido de ésta, mantener una acción activa durante un periodo de tiempo concreto, etc. Entre etapa y etapa hay transiciones que deben cumplirse para que el paso de una a otra se cumpla.

El lenguaje Grafcet permite programar, tanto las acciones como las transiciones, con otros tipos de lenguajes. La flexibilidad de este tipo de lenguaje le permite ser realmente útil y eficaz en la programación de tareas con gran número de condiciones, en tareas que rigen procesos de varios subprocesos y en tareas de control.

1.1. Definición general de funcionamiento

Para el correcto funcionamiento de la estación de multiproceso *Fischertechnik* se ha decidido que esta se gobierne exclusivamente desde el panel de control configurado, pudiendo decidir que tipo de modo de funcionamiento se desea utilizar y recibiendo información detallada de las variables del proceso.

Como en este capítulo nos vamos a centrar en los aspectos relacionados con la programación, primero será conveniente introducir como es el funcionamiento de la maqueta, viendo paso a paso cuáles son sus funciones.

Cabe añadir que, como se explicará más adelante, la maqueta se ha estructurado en tres bloques principales. El motivo es mejorar y hacer más eficiente la programación de todo el proceso.

Por este motivo, en los siguientes apartados se procederá a comentar con profundidad cada uno de estos bloques.

Todo este proceso de subdivisión del proceso cobra aún más sentido cuando la programación se ha basado en el lenguaje SFC (Grafcet). Así, cada uno de los bloques en los que se ha dividido la maqueta se entienden como diferentes arboles Grafcet. Son pues tareas secuenciales que se encuentran interconectados entre sí.

2.2. Detalle de funcionamiento

Como hemos comentado la maqueta se ha estructurado en tres bloques bien diferenciados. Estos bloques serán en adelante los siguientes:

- Proceso horno
- Proceso brazo manipulador
- Proceso mesa giratoria y cinta

Estos bloques tienen una programación interconectada entre sí, es decir, un proceso solo se habilitará si se cumplen unas condiciones en las variables correspondientes. Esto se ha conseguido por el tipo de lenguaje SFC utilizado. Todo esto se podrá ver detalladamente en el apartado "2.3. Descripción del proceso de automatización".

2.2.1. Variables del proceso

Para poder tener un control del proceso es necesario recibir información de la estación y enviar las instrucciones a esta. Para poder realizar esto es necesario la existencia de variables de entrada y salida.

Todas las variables de entrada/salida que se disponen son de tipo booleano, es decir, son bits con dos estados posibles: 0 o 1.

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON HORNO MEDIANTE
AUTÓMATA SIEMENS S7 1214C Y SCADA EN WINCC

A continuación, se expondrán las entradas y salidas digitales del sistema. Nos limitaremos a nombrar su función y el nombre que se le ha asignado a cada una de ellas. El direccionamiento de estas variables se puede observar en el apartado “2.3.1. Direccionamiento de variables”.

Entradas

NOMBRE VARIABLE	DESCRIPCIÓN
I1	Final de carrera de la mesa en el brazo manipulador
I2	Final de carrera de la mesa en la cinta
I3	Final de carrera de la pieza en el final de la cinta
I4	Final de carrera de la mesa en la sierra
I5	Final de carrera del brazo manipulador en la mesa
I6	Final de carrera del alimentador del horno dentro
I7	Final de carrera del alimentador del horno fuera
I8	Final de carrera del brazo manipulador en el horno
I9	Final de carrera de la pieza en el alimentador

Tabla 1 Variables de entrada

Salidas

NOMBRE VARIABLE	DESCRIPCIÓN
Q1	Motor para giro de la mesa en sentido horario
Q2	Motor para giro de la mesa en sentido anti horario
Q3	Motor para avance de la cinta
Q4	Motor para el movimiento de la sierra
Q5	Motor para mover alimentador dentro del horno
Q6	Motor para mover alimentador fuera del horno
Q7	Motor para movimiento del brazo manipulador hacia el horno
Q8	Motor para movimiento del brazo manipulador hacia la mesa giratoria
Q9	Habilitar las señales de entrada del proceso
Q10	Habilitar el compresor
Q11	Habilitar la succión de la ventosa del brazo manipulador
Q12	Habilitar la bajada del brazo manipulador
Q13	Habilitar la apertura de puerta a la entrada del horno
Q14	Habilitar el empujador de la mesa giratoria
Q15	Habilitar la luz del horno

Tabla 2 Variables de salida

Estas variables son las que tiene físicamente la maqueta de la estación de multiproceso. A la hora de programar, se han utilizado muchas otras variables auxiliares al proceso. Todas estas variables son las que denominaremos como Marcas del sistema, variables internas del autómata que nos serán de gran utilidad. En los apartados que se ven más adelante queda constancia de todo esto.

2.2.2. Proceso horno

Este proceso se ejecuta en primer lugar, siempre y cuando nos encontremos en modo automático. El alimentador se encuentra inicialmente fuera del horno, tal y como se aprecia en la Fig. 3.

Una vez se deposita la pieza en el alimentador se abre la compuerta, accionada mediante aire comprimido por el compresor de la maqueta. El alimentador entra dentro del horno hasta que se activa el final de carrera de dentro "I6". Acto seguido se cierra la compuerta y el horno se pone en funcionamiento. Para visualizar que el horno está en marcha se dispone de una señal luminosa, accionada de manera intermitente cada un segundo. Transcurrido un tiempo la compuerta se abre de nuevo y se extrae la pieza mediante el alimentador. Este se detiene al activarse el final de carrera "I7".

Durante este proceso el brazo manipulado ya se ha dispuesto justo enfrente del alimentador, esperando a que termine el proceso del horno para poder recoger la pieza y transportarla al siguiente módulo.

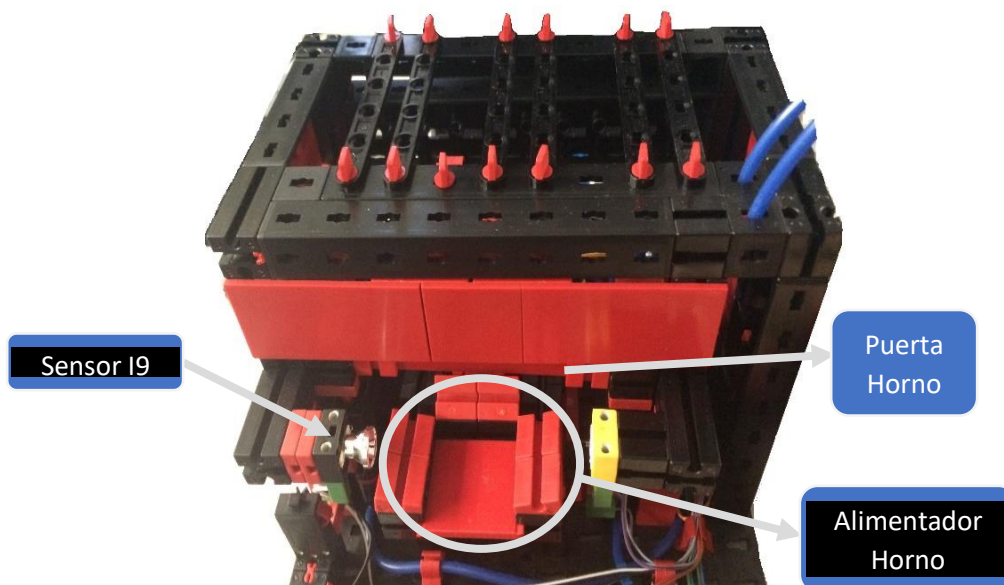


Fig. 3 Detalle del proceso Horno

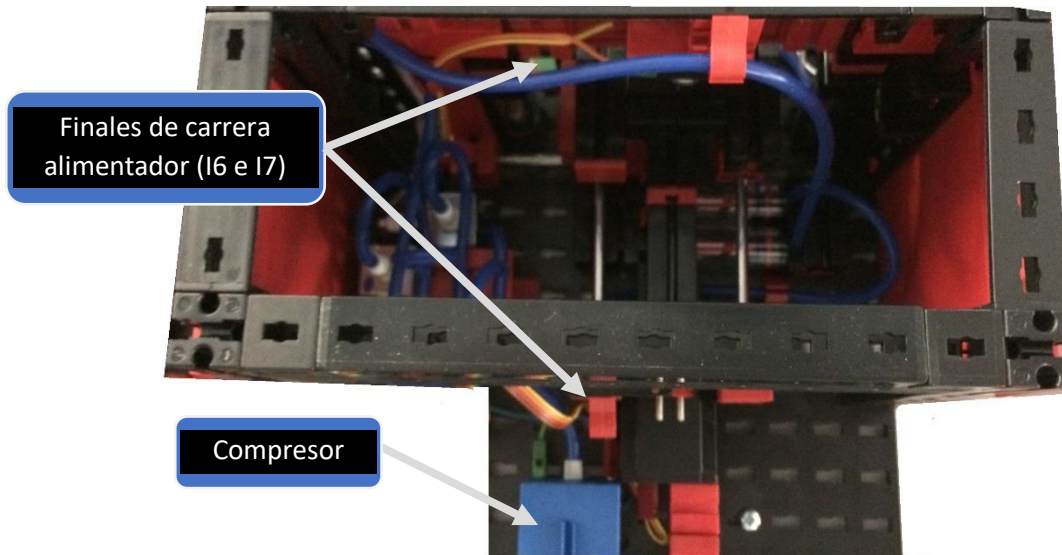


Fig. 4 Detalle del proceso horno. Visto desde arriba

2.2.3. Proceso brazo manipulador

Este proceso se ejecuta seguidamente al del horno, aunque estrictamente se inicia poco antes de que acabe dicho proceso. El objetivo de esto es agilizar el proceso, haciéndolo más eficiente sin comprometer la integridad del mismo.

El brazo inicialmente se encuentra enfrente de la mesa giratoria, posición mostrada en la *Fig. 4*. Una vez la pieza está dentro del horno el brazo avanza hacia allí, parándose al accionar el final de carrera "18". Una vez la pieza se extrae del horno, el brazo manipulador baja y la succiona con la ventosa que posee. Activada la succión, el brazo se eleva y avanza hacia la mesa giratoria, parando al accionarse el final de carrera "15". La succión se mantiene durante todo el movimiento del brazo manipulador hacía la mesa giratoria.

Al activarse el final de carrera del manipulador en la mesa giratoria "15", se activa de nuevo la bajada del brazo y se desactiva acto seguido la succión de la ventosa, depositando la pieza en el receptor de la mesa giratoria.

Hay que mencionar que el brazo manipulador se ha programado para que avance cuando una pieza se encuentre en el proceso del horno, es decir, la activación del manipulador está gestionada por el proceso del horno. Todo esto se detalla en el apartado "2.3. Descripción del proceso de automatización".

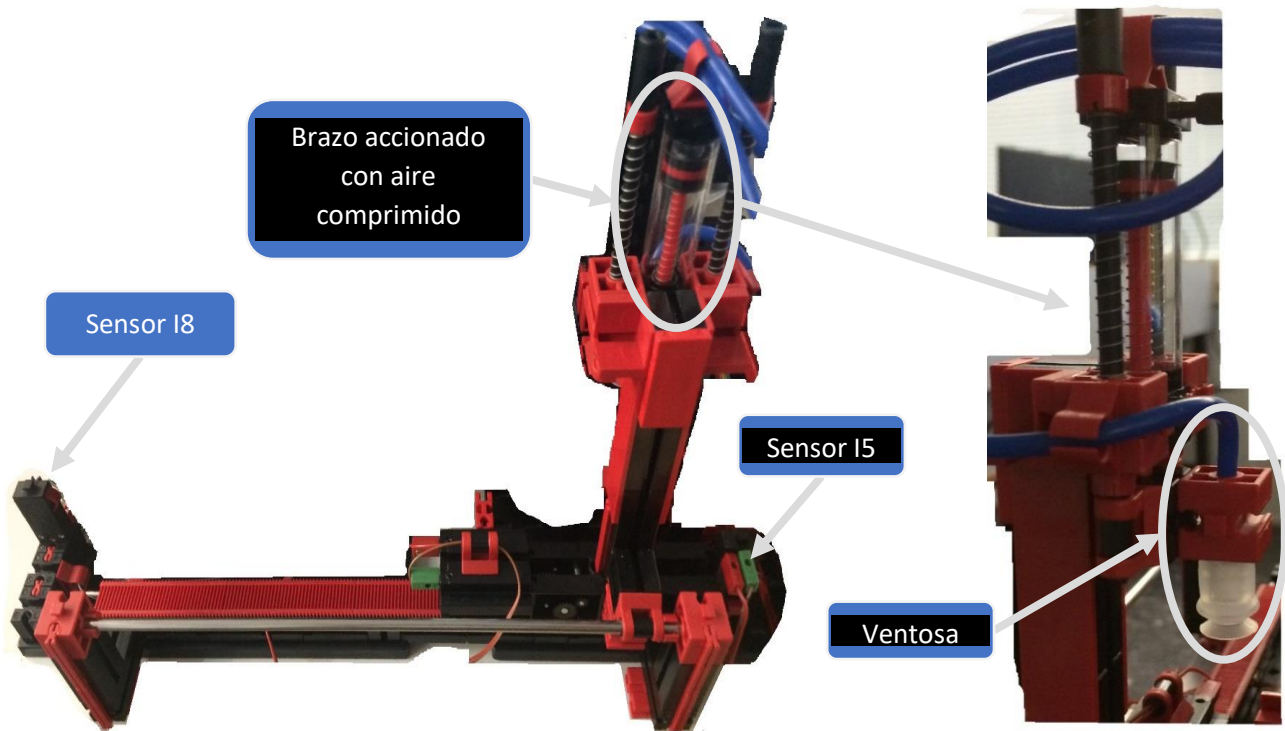


Fig. 5 Detalle proceso brazo manipulador. Ampliación del brazo con ventosa

2.2.4. Proceso mesa giratoria y cinta

Este último proceso comienza cuando han finalizado las tareas del brazo manipulador, iniciándose las acciones de la mesa giratoria cuando se deposite la pieza en el receptor de esta. Como se puede apreciar en la Fig. 6, el estado inicial de la mesa es el que se aprecia (con "11" activado).

Al depositarse la pieza en el receptor de la mesa se procede al giro, en sentido horario, hasta activar el final de carrera que la mesa dispone en la zona de la sierra (sensor I4). Cuando se encuentra aquí, se activa el motor que acciona la sierra durante un espacio de tiempo (unos 3 segundos). Una vez finalizado el serrado, la mesa vuelve a avanzar en sentido horario hasta llegar al final de carrera en la cinta (sensor "12"). Activado este final de carrera se acciona el empujador, accionado por aire comprimido, que lanza la pieza desde la mesa giratoria a la cinta. El empujador es activado durante un intervalo de tiempo muy corto, aproximadamente 0.3 segundos.

En el momento en que la pieza es conducida a la cinta esta se activa, procediendo a evacuar la pieza ya mecanizada. Cuando el final de carrera de la cinta se desactiva (sensor normalmente activo, "13") la cinta detiene su avance. Durante el proceso de evacuación de la pieza, la mesa giratoria se devuelve a su estado inicial, esperando a que el brazo manipulador deposite otra pieza en el receptor.

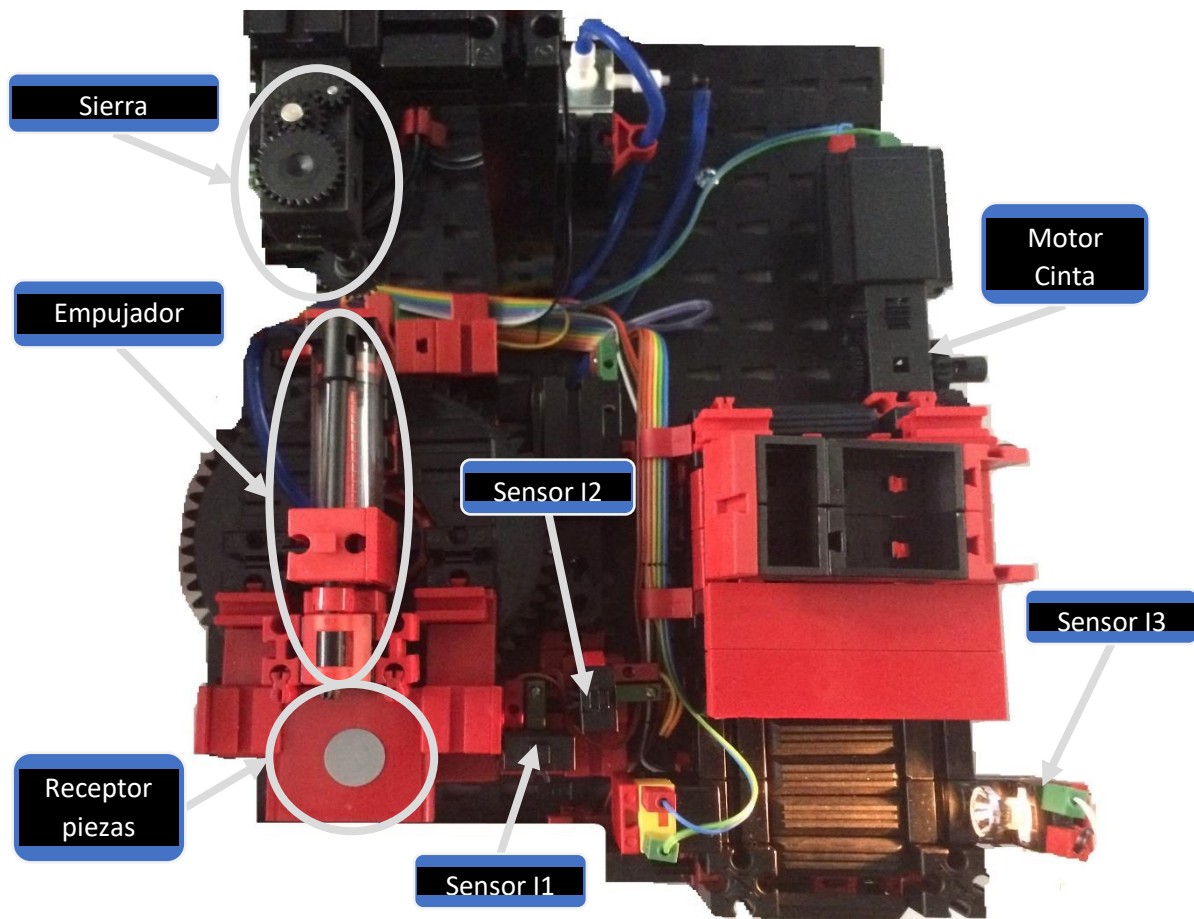


Fig. 6 Detalle proceso mesa giratoria y cinta

2.3. Descripción del proceso de Automatización

Como ya hemos comentado, la programación de todo el proceso se ha realizado en dos tipos de lenguaje, esquema secuencial de funciones o GRAFCET (SFC) y diagrama de contactos o *Ladder* (LD).

En esencia, todo el proceso de automatización se ha programado en GRAFCET, disponiéndolo en diferentes etapas secuenciales en las que se activan las acciones deseadas y de transiciones que, de ser flanqueadas, cambian de etapa al proceso.

El proceso se ha subdividido en bloques. Esta subdivisión se debe a que cada uno de estos bloques está programado como un GRAFCET, existiendo una interconexión entre ellos. Podríamos decir que existe una jerarquía en la programación de todo el proceso, donde este se inicia en el proceso del horno, le sigue el proceso del brazo manipulador y por último el proceso de la mesa giratoria y cinta.

2.3.1. Direccionamiento de variables

Para poder entender claramente todo el proceso es necesario que primero se presenten todas aquellas variables que toman partido en el proceso de automatización. Todas las variables que participan en este proyecto son booleanas, es decir, solamente tendrán dos estados: *True* o *False*.

En anteriores apartados ya se ha comentado la existencia de entradas y salidas del proceso, pero no se ha mencionado la dirección de dichas variables ni como el autómata las reconoce. Aquí se expondrán que direcciones toman todas y cada una de estas variables y como se clasifican.

Además, este proceso requiere de muchas otras variables auxiliares que faciliten su programación, a estas variables se les conoce como marcas de sistema.

En el software de programación TIA Portal se han creado diferentes tablas de variables, estas tablas se han creado a nivel global, es decir, todo el programa tiene acceso a estas. Por otro lado, tenemos las variables asociadas a temporizadores y contadores, estas se crean en bloques de datos separados que se vinculan a ellos.

Toda esta gestión de variables desde el editor de proyectos de TIA Portal se expone con más detalle en el apartado “3.2.2. Bloques de programa. Bloque Main, funciones y DB’s”.

2.3.1.1. Asignación de entradas

En la imagen siguiente se muestran las variables de entrada disponibles en el proceso, con su respectivo direccionamiento para ser reconocidas por el autómata.

	Nombre ▲	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	I1	Bool	%I8.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera mesa en manipulador
2	I2	Bool	%I8.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera mesa en cinta
3	I3	Bool	%I8.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera pieza final cinta
4	I4	Bool	%I8.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera pieza en sierra
5	I5	Bool	%I8.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera manipulador en mesa
6	I6	Bool	%I9.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera manipulador horno dentro
7	I7	Bool	%I8.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera manipulador horno fuera
8	I8	Bool	%I9.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera manipulador en horno
9	I9	Bool	%I8.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Final de carrera pieza en alimentador

Fig. 7 Tabla de variables de entrada

En el panel de arriba de la figura vemos que estas variables se han declarado en la Tabla de variables Horno [70].

2.3.1.2. Asignación de salidas

En la siguiente imagen se representan las salidas del proceso y su direccionamiento en el autómatas de Siemens.

	Nombre	Tipo de datos	Dirección ▲	Rema...	Visibl...	Acces...	Comentario
10	Q1	Bool	%Q8.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor giro mesa horario
11	Q2	Bool	%Q8.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor giro mesa antihorario
12	Q3	Bool	%Q8.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor movimiento avance cinta
13	Q4	Bool	%Q8.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor movimiento sierra
14	Q5	Bool	%Q8.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor alimentador horno dentro
15	Q6	Bool	%Q8.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor alimentador horno fuera
16	Q7	Bool	%Q8.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor mov. manipulador hacia horno
17	Q8	Bool	%Q8.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Motor mov. manipulador hacia mesa girat..
18	Q9	Bool	%Q9.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Habilitar señales entrada proceso
19	Q10	Bool	%Q9.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Habilitar compresor
20	Q11	Bool	%Q9.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Habilitar succión ventosa
21	Q12	Bool	%Q9.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Movimiento manipulador abajo
22	Q13	Bool	%Q9.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Movimiento puerta arriba
23	Q14	Bool	%Q9.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Movimiento empujador mesa giratoria
24	Q15	Bool	%Q9.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Habilitar luz horno

Fig. 8 Tabla de variables de salida

Como se puede observar en la figura, las variables de salida también están declaradas en la Tabla de variables Horno [70].

2.3.1.3. Uso de marcas del sistema

Para la programación del proyecto son necesarias muchas otras variables además de las entradas y salidas. Estas variables son las llamadas marcas del sistema, declaradas a nivel global en tablas de variables.

En cuanto a las marcas utilizadas podemos clasificarlas de la manera que sigue:

- Marcas de Estados

	Nombre	Tipo de datos	Dirección ▲	Rema...	Visibl...	Acces...	Comentario
25	X0	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	X1	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	X2	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	X3	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	X4	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	X5	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	X10	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	X11	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	X12	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	X13	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	X14	Bool	%M1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	X15	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37	X16	Bool	%M1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38	X20	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39	X21	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	X22	Bool	%M2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41	X23	Bool	%M2.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42	X24	Bool	%M2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43	X25	Bool	%M2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
44	X30	Bool	%M3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
45	X31	Bool	%M3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
46	X32	Bool	%M3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Fig. 9 Tabla de variables: marcas de estados

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON HORNO MEDIANTE AUTÓMATA SIEMENS S7 1214C Y SCADA EN WINCC

Son las marcas que representan las variables asociadas a los estados de cada Grafcet. Reciben un nombre y una dirección.

También están declaradas en la Tabla de variables Horno [70].

- Marcas de transiciones

Representan las marcas asociadas a las transiciones de cada Grafcet, el nombre que tienen asignado representa la etapa que desactivan si se flanquea dicha transición.

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
47	TR0	Bool	%M5.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
48	TR1	Bool	%M5.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
49	TR2	Bool	%M5.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
50	TR3	Bool	%M5.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
51	TR4	Bool	%M5.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
52	TR0-1	Bool	%M5.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
53	TR5	Bool	%M5.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
54	TR10	Bool	%M6.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
55	TR11	Bool	%M6.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
56	TR12	Bool	%M6.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
57	TR13	Bool	%M6.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
58	TR14	Bool	%M6.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
59	TR15	Bool	%M6.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
60	TR16	Bool	%M6.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
61	TR20	Bool	%M7.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
62	TR21	Bool	%M7.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
63	TR22	Bool	%M7.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
64	TR23	Bool	%M7.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
65	TR24	Bool	%M7.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
66	TR25	Bool	%M7.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
67	TR30	Bool	%M8.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
68	TR30-1	Bool	%M8.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
69	TR31	Bool	%M8.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
70	TR32	Bool	%M8.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Fig. 10 Tabla de variables: marcas de transiciones

Al igual que las marcas de estados están declaradas en la Tabla de variables Horno [70].

- Marcas de SCADA

Estas variables se han creado expresamente para el panel de control, siendo variables que se comunican tanto con el SCADA como con el PLC.

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	Modo_Auto	Bool	%M60.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Modo_Manu	Bool	%M60.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Parada_Emergencia	Bool	%M60.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Acc_Q1	Bool	%M60.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Acc_Q2	Bool	%M60.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Acc_Q3	Bool	%M60.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Acc_Q4	Bool	%M60.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Acc_Q5	Bool	%M60.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Acc_Q6	Bool	%M61.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Acc_Q7	Bool	%M61.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Acc_Q8	Bool	%M61.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Acc_Q11	Bool	%M61.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Acc_Q12	Bool	%M61.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Acc_Q13	Bool	%M61.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Acc_Q14	Bool	%M61.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Acc_Q15	Bool	%M61.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Fig. 11 Tabla de variables: marcas de SCADA

Como puede apreciarse en la figura estas marcas pertenecen a la Tabla de variables SCADA [16]. En el Capítulo 3 se hablará sobre otra tabla de variables necesaria para que el SCADA tenga control sobre las variables deseadas.

- Otras marcas de interés
Desde TIA Portal se pueden configurar un tipo de marcas especiales (marcas de sistema y de ciclo), para habilitarlas se debe de configurar las direcciones que van a ocupar.

	Nombre	Tipo de datos	Dirección	Rema...	Visibl...	Acces...	Comentario
1	System_Byte	Byte	%MB100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	FirstScan	Bool	%M100.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	DiagStatusUpdate	Bool	%M100.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Always TRUE	Bool	%M100.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Always FALSE	Bool	%M100.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Clock_Byte	Byte	%MB200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Clock_10Hz	Bool	%M200.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Clock_5Hz	Bool	%M200.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Clock_2.5Hz	Bool	%M200.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Clock_2Hz	Bool	%M200.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Clock_1.25Hz	Bool	%M200.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Clock_1Hz	Bool	%M200.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Clock_0.625Hz	Bool	%M200.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Clock_0.5Hz	Bool	%M200.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Flanqueo_1	Bool	%M10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Fig. 12 Tabla de variables: otras marcas de interés

Como se puede apreciar en la imagen estas marcas tienen particularidades especiales. En nuestro proyecto únicamente se han utilizado “FirstScan”, que se habilita cuando el PLC pasa a RUN, y “Clock_1Hz”, que se activa cíclicamente cada un segundo. La variable “Flanqueo_1” va asociada a “I9”, por ser una variable de flanco descendente.

Hay que mencionar que estas variables están declaradas en la Tabla de variables estándar [43].

2.3.1.4. Bloques de Datos (DB)

Como hemos comentado, en los bloques de datos se almacenan las variables asociadas a temporizadores y contadores. Estas variables solo toman lugar cuando, desde la programación, se activan dichos temporizadores o contadores.

Para la programación de nuestro proceso se han utilizado numerosos temporizadores, algunos para establecer que una acción o varias estén activas durante un tiempo establecido. Otros simplemente nos permiten retrasar el inicio o el final de un proceso.

Para conocer las variables de interés de cada uno de los DB’s se recomienda leer el apartado “3.2.2. Bloques de programa. Bloque Main, funciones y DB’s”.

2.3.2. Automatización mediante GRAFCET

La programación que se ha seguido para automatizar los diferentes bloques es la de esquema secuencial de funciones (SFC). Se mostrará la estructura de cada uno de los Grafquets para ver con detalle cómo funciona cada bloque, como se han realizado las conexiones entre procesos y ver claramente la jerarquía de bloques.

Las variables que aparecen en los esquemas de cada bloque son mayormente las entradas y salidas físicas del sistema. No obstante, también aparecen variables asociadas a las etapas de cada proceso, así como temporizadores en transiciones y asociados a acciones. Todas estas variables se consideran variables auxiliares que, pese a no ser físicamente del proceso, son casi tan importantes como estas. Todas ellas han sido introducidas en el apartado anterior.

Hay que añadir que inicialmente todos los Grafquets se encuentran en su etapa inicial (etapa con doble borde).

- **Grafquet Horno**

Este esquema podría denominarse como el inicial, en él se inicia todo el proceso y cuando se producen los flaqueos de ciertas etapas inicia la acción de otros Grafquets.

No vamos a entrar en detalle del funcionamiento que tiene la maqueta al programar este Grafquet, pues ya se ha explicado anteriormente. Sin embargo, si cabe aclarar y explicar diferentes aspectos de su programación.

Claramente el Grafquet se inicia en la etapa "X0", a partir de esta hay una ramificación en O excluyente (solamente se puede cumplir una de ambas). Si este pasa a la etapa "X5" se activará la programación de otro Grafquet denominado Estado Inicial. No entraremos en detalle de su funcionamiento ya que se va a explicar en el siguiente punto.

Si, por el contrario, es la etapa "X6" la que se activa se procederá al funcionamiento normal, ya descrito del proceso del horno.

Cuando se alcanza la etapa "X4" el proceso horno ha finalizado, pero como se puede observar existe otra transición para devolver este proceso a su estado inicial. Esta transición está asociada a una de las etapas del Grafquet del brazo manipulador. El motivo de esto es sencillo, mientras el brazo no acabe con su tarea no será posible alimentar al horno con más piezas.

Se puede observar que la etapa "X0" y todas las demás de este Grafquet tienen siempre dos acciones activas, estas son "Q9" y "Q10". Volviendo a la tabla de variables de la página 14 podemos ver que estas salidas se corresponden con habilitar las señales de entrada del proceso y el compresor respectivamente.

Esto se ha realizado así para que estén siempre activas, ya que es necesario poder leer siempre las entradas y nos ahorra tiempo y energía mantener el compresor siempre activo, no arrancándolo y parándolo a cada momento que sea necesario.

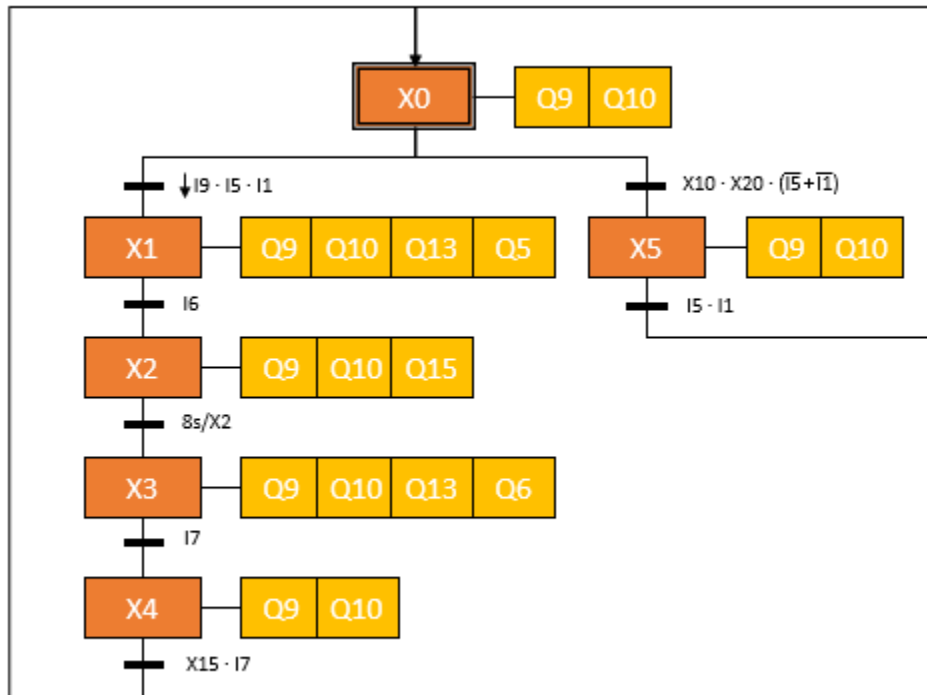


Fig. 13 Grafset del proceso horno

- **Grafset Estado Inicial**

Como se ha mencionado en el punto anterior este Grafset se inicia cuando en el proceso del horno se da la transición que activa la etapa "X5". El flanqueo de dicha transición está sujeto a que todos los Grafcets se encuentren en su etapa inicial y que, adicionalmente, no se encuentren activos los sensores "I1" o "I5".

La finalidad de este Grafset no es otra que devolver la maqueta a su estado inicial, es decir, que el brazo manipulador y la mesa giratoria estén activando el sensor I5 (manipulador enfrente de la mesa giratoria) y "I1" (mesa enfrente del manipulador) respectivamente.

Este proceso es útil para casos en que se activa una parada de emergencia, ya que en caso de resetear el sistema todos los Grafcets activan su etapa inicial. Por tanto, si el brazo o la mesa no se encuentran en su posición inicial estos vuelven a ella nada más arrancar el proceso.

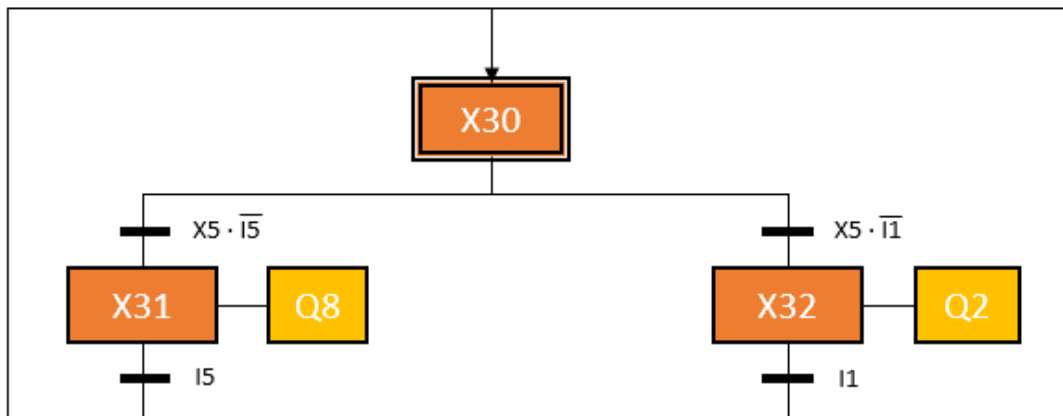


Fig. 14 Grafcet del proceso devolver estación al estado inicial

Adicionalmente se ha añadido el modo manual que también puede solventar este problema. Sin embargo, no está de más tener mecanismos de seguridad que nos agilicen la tarea en caso de situaciones inesperadas, por eso se ha decidido mantener este Grafcet en la programación.

- **Grafcet brazo manipulador**

De manera análoga a lo mencionado en el proceso del horno se van a detallar los aspectos clave del funcionamiento del Grafcet, no se va a describir el proceso que realiza el brazo manipulador, pues ya se ha explicado con detalle anteriormente.

Inicialmente el Grafcet se encuentra en la etapa "X10", tal y como se aprecia en la imagen. Para que el proceso se inicie se debe de cumplir la primera transición y, como se puede ver, está vinculada a la segunda etapa del Grafcet horno. Por tanto, es aquí donde se relacionan ambos Grafcets.

En esencia, para que el brazo manipulador inicie su tarea la pieza debe de estar ya dentro del horno.

Los temporizadores que aparecen en las transiciones sirven para marcar el tiempo que el brazo manipulador se mantiene bajado. Por otro lado, el tiempo asociado a la acción "Q11" en la etapa "X15" solamente sirve para que la succión de la ventosa finalice antes de que suba el brazo, evitando que la pieza se despeje del receptor de la mesa.

Como ya sucedía en el proceso del horno hay una etapa ("X16") que sirve de espera para devolver al Grafcet a su etapa inicial. Para ello se debe cumplir que se active la etapa "X22", correspondiente al Grafcet de la mesa giratoria. Esta transición marca que ya se ha iniciado el siguiente proceso y que el brazo puede volver a su etapa inicial, esperando a ser llamado de nuevo.

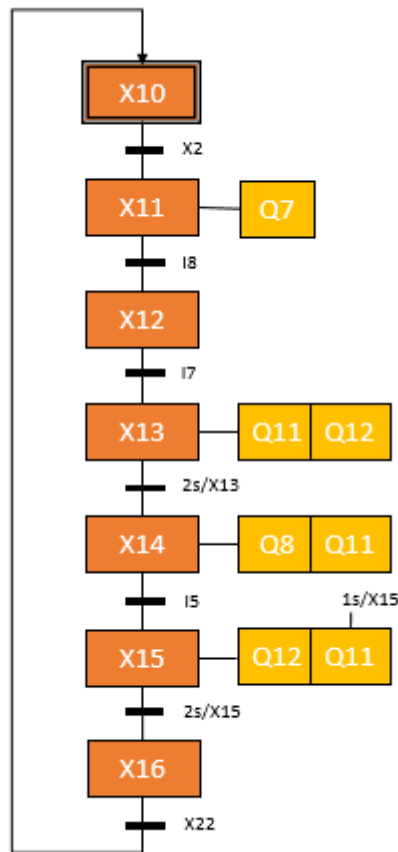


Fig. 15 Grafcet del proceso brazo manipulador

- **Grafcet mesa giratoria y cinta**

Con un rápido vistazo a la imagen se puede determinar que la etapa inicial del Grafcet es la "X20". Como ya sucedía en los anteriores procesos, la transición para que se inicien las tareas de la mesa y cinta está relacionada con una etapa del Grafcet anterior, en concreto con el proceso del brazo manipulador.

Para iniciar este proceso es necesario que el anterior haya finalizado, esa es la finalidad de que aparezca la etapa "X16" en la primera transición. También aparecen los sensores I1 e "I3", verificando que la mesa este en su posición inicial y que ninguna pieza se encuentre en el final de la cinta respectivamente.

En la etapa "X22", la acción condicionada al sensor "I4" no es más que una comprobación de seguridad para la sierra. Así, mientras la mesa esté parada en el lugar correcto la sierra puede girar el tiempo requerido.

Los temporizadores funcionan de la misma manera que se ha comentado en el proceso anterior. El tiempo asociado a la acción "Q14" se debe a que el empujador solo se activa durante un breve periodo.

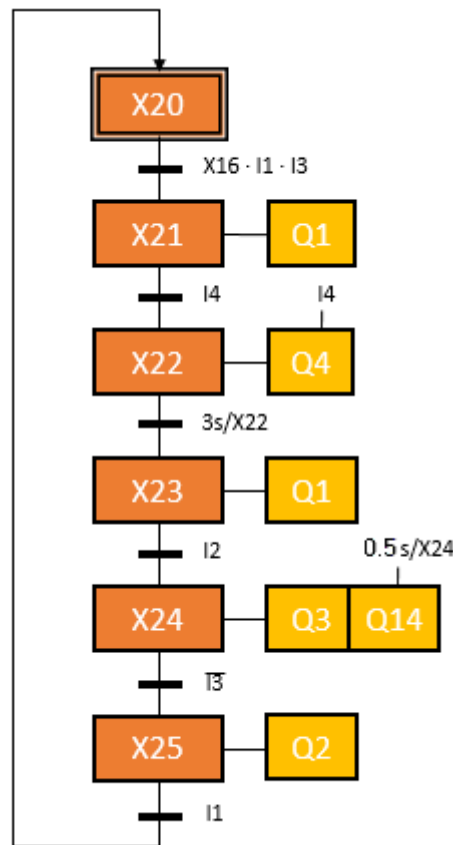


Fig. 16 Grafcet del proceso mesa giratoria y cinta

2.3.3. Conversión de GRAFCET a LADDER (KOP)

Ya se ha visto como se ha programado el proceso en lenguaje de programación Grafcet. La elección de este tipo de programación nos permite realizar una gran infinidad de tareas que con otros lenguajes resultaría más complicado.

Pero la elección de este tipo de programación tiene un fin aún más específico. El entorno de programación del autómatas que disponemos no permite su programación en esquema secuencial de funciones, esto es un problema pues toda la programación se ha basado en lenguaje tipo Grafcet. Es aquí donde la versatilidad y capacidad de este lenguaje nos sirve para adaptarlo a un lenguaje que el PLC si sea capaz de ejecutar.

Se ha optado por pasar del lenguaje Grafcet al de esquema de contactos o *Ladder* (también conocido como KOP). Ya se ha comentado que este tipo de lenguaje está basado en esquemas eléctricos. Para la conversión de toda la programación se seguirán unos pasos muy simples que facilitarán esta tarea.

1) Dar nombre a las etapas del proceso y las transiciones

En este primer paso se añadirán variables de bits de marcas para cada uno de los estados y transiciones. Al hacer esto se tiene control y constancia de todas las etapas y transiciones que hay en el proceso.

Para ver todas las variables añadidas ver el apartado “2.3.1. Direccionamiento de variables” del presente capítulo.

2) Programar la etapa inicial

Durante el primer ciclo de ejecución del programa se inician aquellas etapas que son iniciales, en nuestro caso las etapas “X0”, “X10”, “X20” y “X30” se activarían mediante un *Set*. El resto de variables asociadas a las etapas se colocan en modo *Reset*. En el Anexo I Segmento 1 se puede ver esta programación.

Para poder realizar la activación inicial en el programa se tendrán que habilitar unas marcas con características especiales. La marca utilizada es “FirstScan”, esta variable se activa cada vez que el autómata cambia al modo RUN (modo de ejecución del proceso).

Todas estas variables están explicadas en el apartado “2.3.1. Direccionamiento de variables”.

3) Programar flanqueo de transiciones

Hay que tener en cuenta que para que una transición se active se debe de cumplir dos cosas: que la etapa inmediatamente anterior este activa y que sean ciertas las instrucciones asociadas a la transición.

En el Anexo I Segmentos 2, 3, 4 y 5 pueden verse este flanqueo de transiciones con detalle.

4) Programar activación y desactivación de las etapas

Una etapa se activa si se cumple la transición inmediatamente anterior. Por el contrario, una etapa se desactiva si se cumple la transición que tiene inmediatamente delante.

Para poder ver todas las activaciones y desactivaciones de etapas ver Anexo I Segmento 6.

5) Programar acciones asociadas a etapas

Casi todas las etapas tienen asociadas a ellas unas acciones determinadas. Por tanto, mientras una etapa este activa las salidas asociadas a ella estarán activas también.

Si estas salidas están sujetas a otros parámetros (como ocurre con la acción Q14 de la etapa X24) tendrán asociados temporizadores o sensores que habilitarán la salida un cierto tiempo o bajo ciertas condiciones.

Para tener más información ver Anexo I Segmentos 7, 8, 9 y 10.

Al realizar esta conversión se consigue aumentar el potencial de la programación realizada; por un lado, se mantienen las características del lenguaje de programación Grafcet y, por otro, se consigue añadir las ventajas que tiene el programar en KOP.

Programar en KOP permite tener un uso sencillo de temporizadores, retardos a la conexión o desconexión, contadores, funciones matemáticas, comparadores, conversión de datos, etc. Algo muy ventajoso en nuestra programación.

2.3.4. Modos de funcionamiento

Hemos mencionado que existen dos modos de funcionamiento: el modo automático, que controla las variables del proceso y lo ejecuta de forma automática y el modo manual, que nos permite tener un completo control sobre las acciones de la estación de multiproceso.

Como es obvio ambos métodos no pueden estar activos al mismo tiempo. Por este motivo en la programación en KOP (descrita en el apartado anterior) se han añadido bits de marcas del sistema, creadas para controlar el proceso desde el Panel de Control, para poder adaptar el proceso a los diferentes modos de funcionamiento.

Seguidamente detallaremos los cambios introducidos en la programación en KOP para que sea posible su control desde el SCADA.

2.3.4.1. Modo Automático

Lo primero que tenemos que tener en cuenta es que el proceso no se inicie hasta que el modo automático no esté seleccionado desde el SCADA. Para poder iniciar el modo automático se tiene que habilitar la señal "Modo_Auto", este se ha habilitado en la primera transición del proceso del Horno (Ver Fig.17).

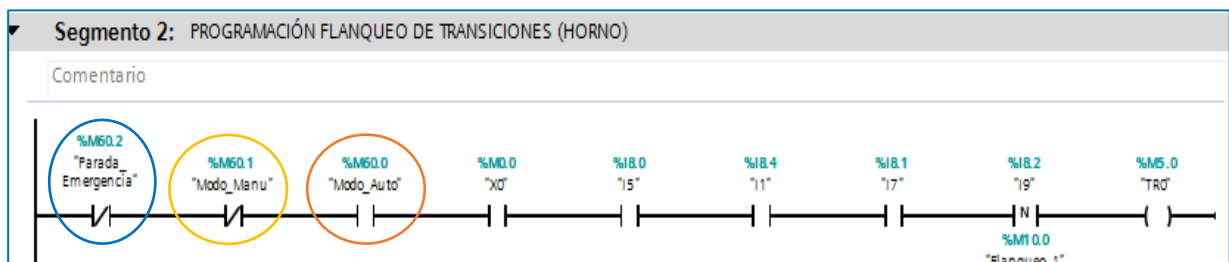


Fig. 17 Ejemplo de programación con uso de bits de modos de funcionamiento y parada de emergencia

Todos los detalles relacionados con esta programación se pueden ver en el Anexo I (Ver Fig. 17 Fig. 18 y Fig. 19).

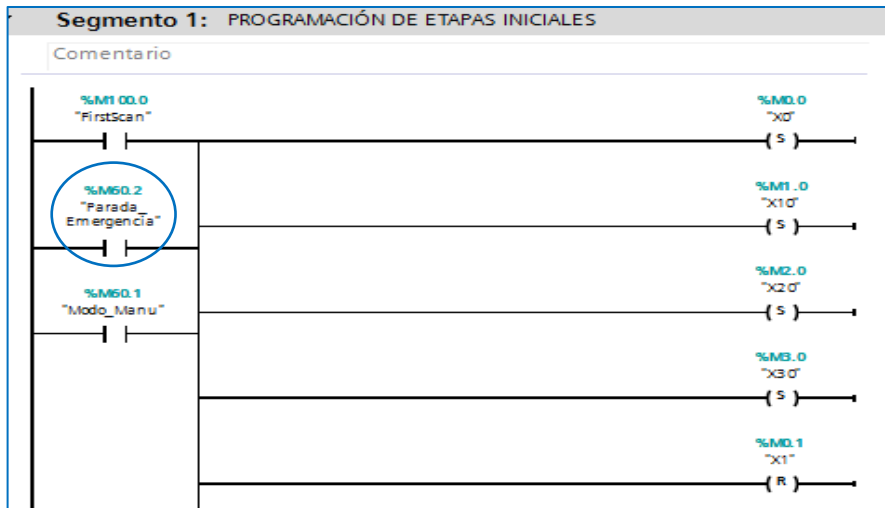


Fig. 19 Ejemplo de programación, bits que activan las etapas iniciales y desactivan el resto

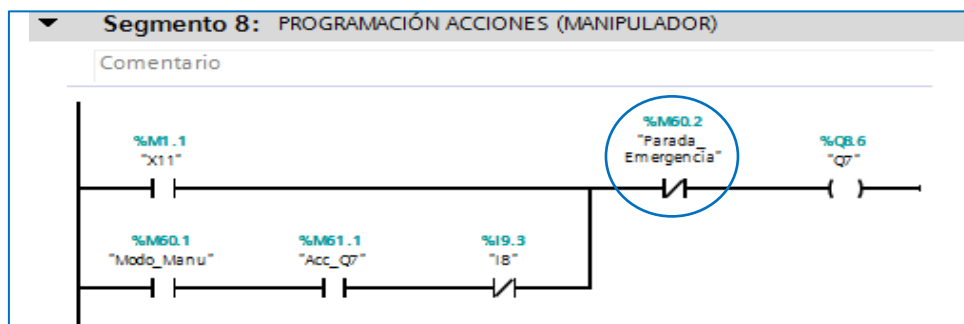


Fig. 18 Ejemplo de programación de desactivación de una acción por parada de emergencia

2.3.4.2. Modo manual

En el modo manual nos interesa que el proceso permanezca en las etapas iniciales, es decir, que no se produzca las primeras transiciones de cada Grafcet. Por esto en la programación se ha añadido una variable "Modo_Manu" que se activa cuando se selecciona el modo manual en el panel de control (Ver Fig.20 y Fig.21).

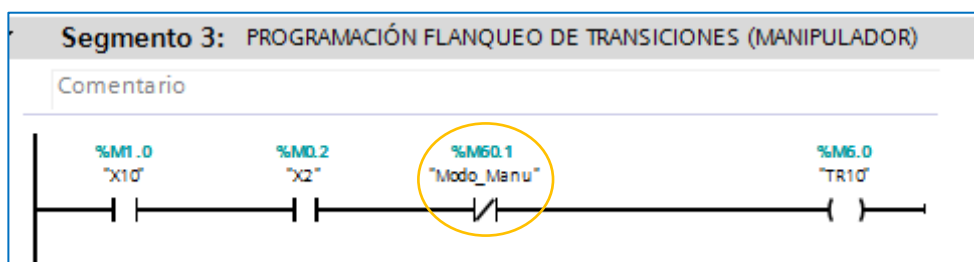


Fig. 20 Ejemplo de programación de no flanqueo de transición por activación de modo manual

Para ver con más detalle la función de esta variable ver Anexo I.

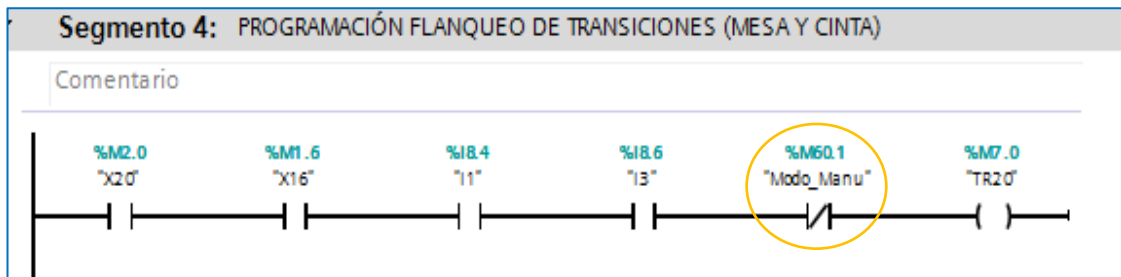


Fig. 21 Ejemplo de programación de no flanqueo de transiciones en modo manual

Hemos visto, en el apartado de las variables del sistema, que se ha creado una tabla de variables para gestionar el SCADA. En el modo manual se han creado unas marcas, activables a través del panel de control, que controlan todas las salidas del sistema. De nuevo estas marcas solo podrán ser activadas si la variable "Modo_Manu" esta activada.

En la Fig. 22 se pueden apreciar ejemplos de cómo funciona la activación de las acciones del proceso. La activación de algunas de estas salidas depende de ciertos sensores. Por ejemplo, "Q7" se mantiene activo mientras "I8" no se ponga a 1 y este activo "Acc_Q7".

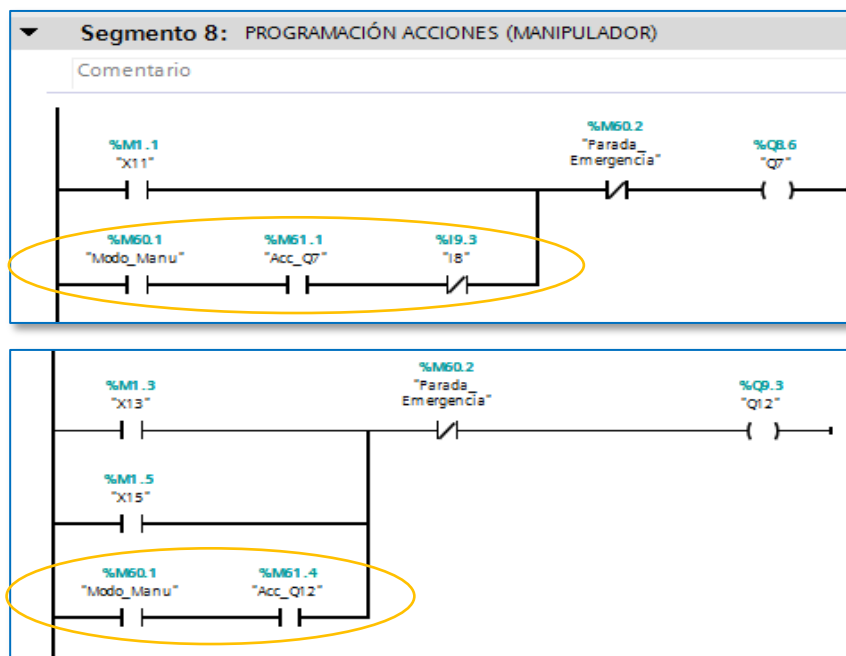


Fig. 22 Activación de acciones desde SCADA con modo manual ejecutado

En el Anexo I se exponen con detalle todos los casos en que ocurre esto.

2.4. Medidas de seguridad

Un proceso industrial nunca está exento de tener imprevistos, cualquier tarea que se lleve a cabo de forma automática requiere de medidas para garantizar la seguridad personal y material.

Es por este motivo que en cualquier proceso de esta índole encontramos acciones que detienen total o parcialmente el sistema. Las setas de emergencia a pie de maquina o en los paneles de control son un elemento indispensable cuando se trata de automatizar un proyecto.

2.4.1. Devolver proceso al estado inicial

Ya se ha comentado en el apartado “2.3.2. Automatización mediante GRAFCET” la existencia de un proceso que devuelve la maqueta a su estado inicial. Esta tarea está muy relacionada con el paro de emergencia, pues si sucede que se interrumpe el proceso por una situación de peligro se devolverá la estación a su situación de inicio.

Pese a que el modo manual puede llegar a tener una función similar, tener mecanismos que agilicen la vuelta a la marcha de un proceso siempre tiene cabida.

Para tener más información de esta programación ver el apartado “2.3.2. Automatización mediante GRAFCET”.

2.4.2. Parada de emergencia

En nuestro caso, no disponemos de un interruptor físico que pare el proceso, pero si hemos añadido un interruptor de emergencia en el SCADA que habilita esta parada. Como es de suponer esta seta de emergencia solo está presente en el modo automático, pues en el modo manual no es necesaria.



Este pulsador de emergencia detiene completamente todo el proceso automático de la planta multiproceso. Está asociado a la variable “Parada_Emergencia”. Si este bit se pone a 1 para las acciones del sistema y devuelve el proceso a sus etapas iniciales. En el Anexo I se puede apreciar mejor cómo afecta al programa la activación de esta variable.

Fig. 23
Seta de emergencia del
SCADA

3. CAPÍTULO 2. El autómata SIEMENS

Hay muchos y muy variados tipos de autómatas en el mercado, de marcas tan conocidas como Omron, Schneider Electric o Siemens. Además, cada uno de estos fabricantes dispone de diferentes modelos, adaptados a cada situación y procesos.

El proceso de selección de un PLC no debe de tomarse a la ligera, hay que tener en cuenta muchos aspectos a la hora de adquirir uno. Normalmente, la principal distinción radica en el coste de cada uno de estos.

En nuestro caso esto no sucede, pues este proyecto se centra en automatizar un proceso con un PLC previamente seleccionado, un autómata de la marca Siemens. La ventaja de esta marca es que disponen de su propio software para la programación, el SIMATIC STEP 7 Basic TIA Portal, un potente software con una interfaz muy intuitiva y fácil de comprender.

En los siguientes apartados pasaremos a exponer las características del autómata disponible en el laboratorio, con el que se ha programado la estación multiproceso con horno. Hablaremos también de los módulos compatibles utilizados y de las conexiones de que se dispone. Adicionalmente hablaremos sobre el software de Siemens y de la conexión realizada con el autómata.

3.1. Características S7-1214C SIEMENS

No vamos a profundizar mucho en las características técnicas de nuestro PLC, pues no es objeto de este proyecto. Para conocer con detalle todo lo referente a este dispositivo se recomienda la lectura del manual de sistema de Siemens sobre el Controlador programable S7-1200, disponible a través de la página web www.siemens.com.

Aun así, conocer bien el dispositivo utilizado no está de más. En la siguiente tabla se detallan algunas de las características básicas de nuestro controlador.

Función		CPU 1214C
Referencia		6ES7 214-1BG40-0XB0
Versión firmware		V4.0
Dimensiones físicas (mm)		110 x 100 x 75
Memoria de usuario	Trabajo	100 KB
	Carga	4 MB
Entradas/salidas integradas	Digital	14 entradas/10 salidas
	Analógico	2 entradas
Memoria imagen del proceso entrada/salida		1024/1024 bytes
Área de marcas (M)		8192 bytes
Ampliación con módulo de señales (SM)		8
Memory Card (Opcional)		SIMATIC Memory Card
Puerto de comunicación Ethernet PROFINET		1
Velocidad de ejecución booleana		0.08 µs/instrucción

Tabla 3 Características de la CPU s7 1214C

Una de las ventajas de los autómatas Siemens es su capacidad de añadir módulos adicionales en función de las prestaciones que se requieran. Existen un gran número de módulos con diferentes finalidades.

Los SM, o módulos de señales, sirven para añadir más entradas y salidas físicas al sistema, idóneos para procesos muy grandes. También se tienen módulos de comunicación, diseñados para establecer conexiones con otros dispositivos o procesos. Por último, se tienen los *Signal Board* (SB) que se acoplan en la parte central del controlador, tienen diferentes funciones, como añadir entradas analógicas o digitales al proceso.

En nuestro controlador S7-1214C se han añadido un par de módulos adicionales. Estos módulos son dos: una *Signal Board* (SB) analógica y un módulo de señales (SM) digital. Este último es el que se ha utilizado para conectar las entradas y salidas del proceso con el autómata.

- **Signal Board (SB 1231) de una entrada analógica**

Modelo	SB 1231 AI 1 x 12 bits
Referencia	6ES7 231-4HA30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	35 gramos
Número de entradas	1
Tipo	Tensión o intensidad (diferencial)
Rango	± 10 V, ± 5 V, $\pm 2,5$ o de 0 a 20 mA
Resolución	11 bits + bit de signo

Tabla 4 Características del SB de una entrada analógica acoplada al PLC

- **Módulo de señales (SM 1221) digital**

Modelo	SM 1221 DI 16 x 24 V DC
Referencia	6ES7 221-1BH32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75
Peso	210 gramos
Número de entradas	16
Tensión nominal	24 V DC a 4 mA, nominal
Consumo de corriente (24 V DC)	4 mA/entrada utilizada

Tabla 5 Características del módulo de señales digital acoplado al PLC

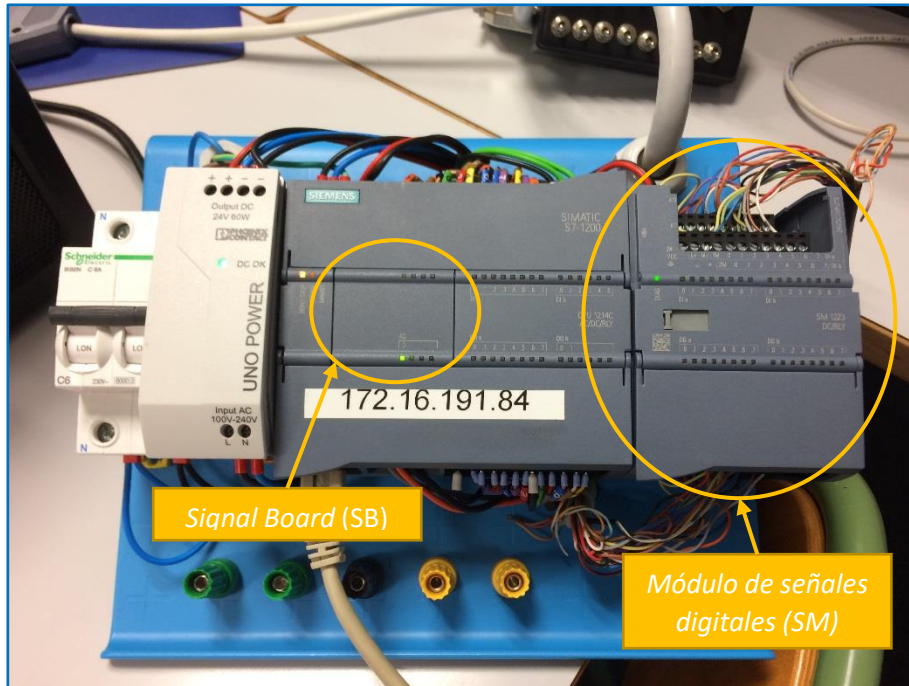


Fig. 24 Detalle del PLC y de los módulos acoplados a este

3.2. Software STEP 7 TIA Portal V13

Como ya hemos comentado, Siemens tiene su propio software de programación para sus controladores. Este software es el STEP 7 TIA Portal, un programa muy versátil y amigable.

Se trata de un software ampliamente conocido y utilizado, sobre todo en la industria alemana. Sus constantes mejoras lo han hecho un programa robusto y fácil de entender, actualmente ya se está comercializando la versión V14 del SIMATIC STEP 7.

Para la programación de nuestro PLC se ha utilizado el software STEP 7 TIA Portal V13. Este programa dispone de tres lenguajes de programación reconocidos por la norma DIN EN 61131-3:

- FUP (*Funktionsplan*) – Diagrama de funciones
- KOP (*Kontaktplan englisch*) – Ladder (LD) o Esquema de contactos
- AWL (*Anweisungsliste englisch*) – Texto estructurado (STL)

Como ya hemos reiterado durante este documento la programación de nuestro proceso se ha realizado en esquema de contactos (KOP).

3.2.1. Conexión entre TIA Portal y S7-1214C

La principal ventaja de trabajar con un software desarrollado por la misma marca que el autómata es su facilidad de comunicación. Al inicio de este documento se ha comentado que para conectar el PLC con el ordenador, y por tanto con el programa, se ha utilizado la red LAN de los laboratorios del Departamento de Ingeniería de Sistemas y Automática (DISA).

Es necesario definir de manera exacta el hardware del autómata dentro del programa TIA Portal. De no ser así la comunicación entre software y hardware no se podrá llevar a cabo. También hay que definir la referencia exacta y la versión de firmware de todos los componentes del PLC.

La comunicación entre el autómata y el PC requiere de una IP, tanto para el controlador como para el router al que se conectan. La IP del autómata ya había sido definida previamente a la realización del presente proyecto. Sin embargo, para establecer comunicación entre el programa y el PLC es necesario que la IP del router utilizado se encuentre en el mismo dominio de red.

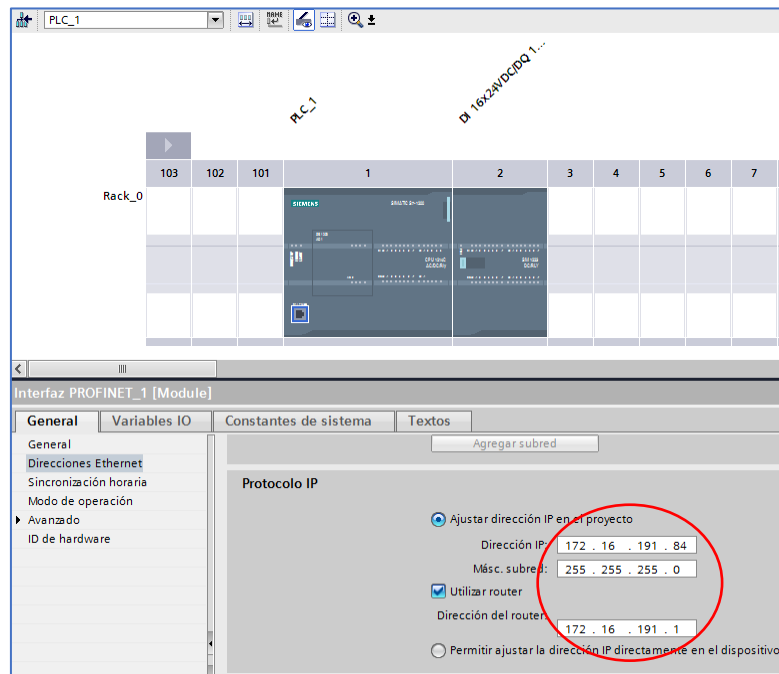


Fig. 25 Comunicación entre el PLC y el programa mediante red LAN

Establecer conexión entre el programa y el PLC nos sirve para mucho más que únicamente transferir el programa. Es posible tener un cierto control de las variables del sistema mediante la creación de tablas de observación y forzado. Estas tablas permiten observar la activación y desactivación de variables digitales, el valor que tienen las diferentes entradas o salidas analógicas o, incluso, observar las variables de un DB (temporizadores, contadores...).

Pero además de estas tablas de observación existe la opción de que se pueda seguir el funcionamiento del proceso desde los bloques de programa, habilitando la conexión online. Esto permite ver cómo funciona el programa de una forma muy visual, permitiendo la comprobación de todo el proceso.

3.2.2. Bloques de programa. Bloque Main, funciones y DB's

En el Anexo I se ha añadido toda la programación realizada, lo que no se ha mencionada es que esta programación está incluida en una función dentro del bloque de programa "Main [OB1]". El bloque "Main [OB1]" es el programa principal y es de carácter cíclico, es decir, se ejecuta continuamente.

Dado que se ha realizado una programación particular, pasando de un lenguaje de programación a otro compatible con el PLC, todo el programa está incluido en el bloque de función "Programa Horno [FC1]".

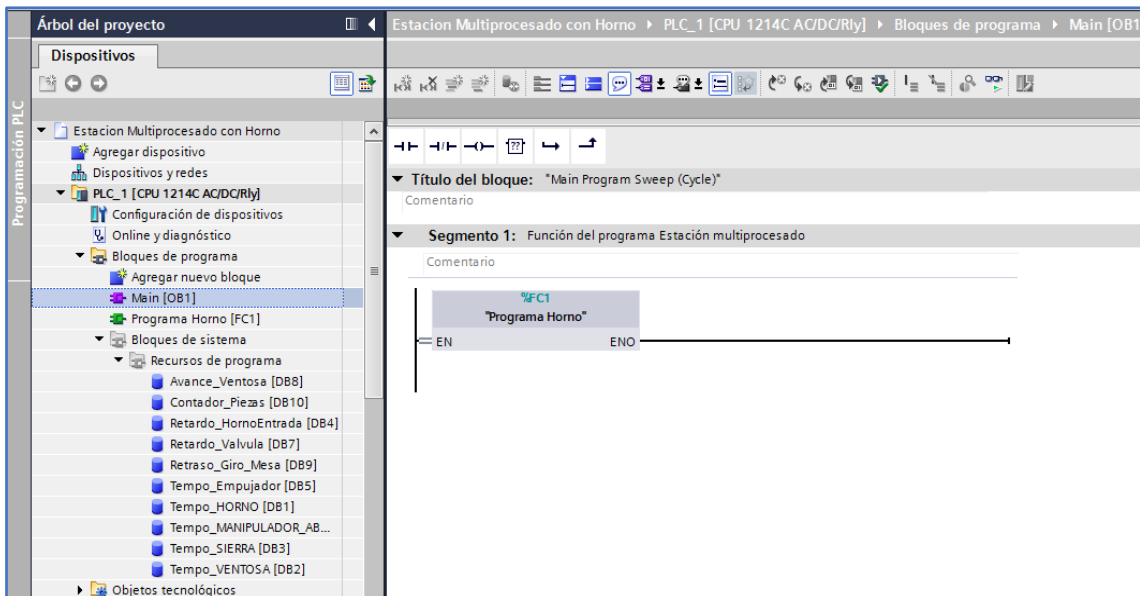


Fig. 26 Árbol de proyecto y contenido de la función Main [OB1]

Dentro de este "Programa Horno [FC1]" esta toda la programación contenida en el Anexo I. Sin embargo, a su vez este programa tiene llamadas a otras funciones, son los temporizadores y contadores que ya hemos mencionado anteriormente. Estos están asociados a los bloques de datos (DB) que el programa genera para gestionar las variables que los constituyen. A la izquierda de la Fig. 26, en el Árbol del Proyecto, se puede observar el desplegable con todos estos bloques de programa.

En nuestro proyecto se han añadido temporizadores, retardos y contadores. Cada uno de estos tiene una finalidad que se va a ir detallando más adelante.

Para una mejor comprensión del funcionamiento de estos bloques se van a exponer los dos tipos de temporizadores usados en el programa y el contador.

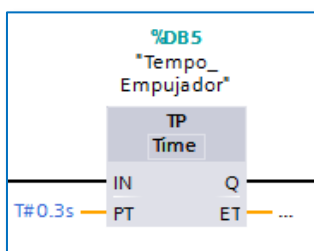


Fig. 28 Ejemplo de temporizador TF

El temporizador TON (de retardo a la conexión) activa la salida "Q" transcurrido el tiempo marcado en "PT" siempre que la señal "IN" este a 1. La variable "ET" es del tipo *time* y marca en todo momento la cuenta del temporizador. Ver Fig. 27.

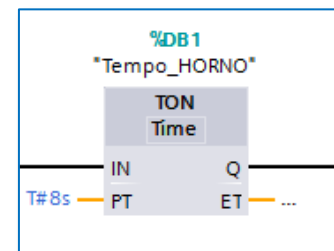


Fig. 27 Ejemplo de temporizador TON

El otro temporizador utilizado en la programación es el TP (de impulso), que activa la salida “Q” durante un tiempo marcado por “PT” siempre que se produzca en la señal “IN” un flanco de subida. La variable “ET” es del tipo *time* y marca en todo momento la cuenta del temporizador. Ver Fig. 28.

Por último, tenemos el contador CTU (contador ascendente), cada vez que se activa la señal “CU” la variable “CV” incrementa en 1 su valor. Una vez “CV” alcanza el valor marcado en “PV” la función activa la salida “Q”. Si se activa la variable “R” se resetea la cuenta y devuelve a 0 el valor de “CV”. Ver Fig. 29.

Por tanto, la lista de todos los bloques de datos utilizados y su función en la que sigue:

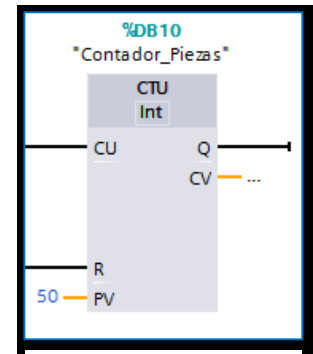


Fig. 29 Ejemplo de contador ascendente CTU

- **“Tempo_HORNO” [DB1]**
Este temporizador TON sirve para establecer el tiempo que la pieza debe permanecer en el horno. El tiempo de horneado se ha marcado en 8s.
- **“Tempo_VENTOSA” [DB2]**
Es el tiempo (TON) que se debe de cumplir para que se produzca la transición “TR15”, es decir, el tiempo que el brazo manipulador permanece bajado.
- **“Tempo_SIERRA” [DB3]**
Es el tiempo (TON) que la sierra permanece girando, marca también la transición “TR22”.
- **“Retardo_HornoEntrada” [DB4]**
Es un pequeño retardo (TON) que se produce al colocar la pieza en el alimentador del horno para evitar que esta no se coloque muy precipitadamente.
- **“Tempo_Empujador” [DB5]**
Es un pequeño temporizador TP asociado a la acción del empujador “Q14” para activarla muy brevemente.
- **“Tempo_MANIPULADOR_ABAJO” [DB6]**
Es el tiempo (TON) que se debe de cumplir para que se produzca la transición “TR13”, es decir, el tiempo que el brazo manipulador permanece bajado. Muy similar al temporizador “Tempo_VENTOSA”.
- **“Retardo_Valvula” [DB7]**
Es un temporizador TON que retrasa la succión de la ventosa (“Q11”) para que la pieza se puede succionar correctamente una vez que el brazo haya bajado.
- **“Avance_Ventosa” [DB8]**
Es un temporizador TP que mantiene activa la succión de la ventosa cuando el brazo baja, evitando que la pieza quede suelta antes de que el manipulador la deposite en la mesa giratoria.

- **“Retraso_Giro_Mesa” [DB9]**
Es un pequeño retardo (TON) de 1s al inicio del giro horario de la mesa para dejar tiempo a que el brazo deposite la pieza correctamente.
- **“Contador_Piezas” [DB10]**
Se trata de un contador ascendente CTU que cuenta las piezas acabadas. Se ha marcado un límite de piezas ficticio (PV = 50). Se resetea al apagar el PLC.

Si queda alguna duda del funcionamiento de estos bloques de programa se puede consultar el Anexo I.

4. CAPÍTULO 3. Panel de Control SCADA

El término SCADA se suele referir a un sistema, normalmente centralizado, que supervisa y controla un proceso completo o una parte del mismo.

Cualquier proceso automático industrial precisa de mecanismos de control y supervisión del proceso. Es inviable que cualquier tarea gobernada por un autómatas no se encuentre monitorizada de alguna manera.

En este sentido los paneles de control mediante HMI o PC han supuesto un avance y comodidad muy notables. Por supuesto que el control de procesos in situ también es necesario, pero poder tener control sobre cualquier tarea a distancia nunca está de más, sobre todo para garantizar la seguridad en planta.

Existen muchos y muy diversos programas que nos permiten realizar paneles de control. Cada fabricante de PLC suele recomendar un software para elaborar los SCADA, incluso algunos desarrollan sus propios programas para garantizar que no exista problemas de conexión con sus autómatas y los HMI o paneles en PC.

En nuestro caso, al disponer de un autómatas de Siemens, se nos da este último caso. Siemens suele trabajar con el software de pago WinCC para realizar sus paneles de control, además no solo hay una versión de este programa. En función del tipo de control que se vaya a realizar nos encontramos con diferentes versiones de WinCC.

Para nuestras condiciones de control, donde utilizaremos una monitorización “monopuesto” hemos elegido la versión WinCC RT Advanced.

4.1. Software WinCC RT Advanced

Desde la página web de Siemens encontramos la siguiente descripción de este software:

"SIMATIC WinCC RT Advanced es la solución HMI basada en PC para sistemas de un solo usuario directamente en la máquina".

<http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-runtime/Pages/default.aspx>

Como ya hemos mencionado, este software es adecuado para realizar paneles de control en procesos “monopuesto”, es decir, procesos que solo tengan un panel de control desde el que se supervise todo. Justo este es nuestro caso.

La principal ventaja y motivo de elección de este software es que es completamente compatible con TIA Portal. Es posible ejecutarlo desde el programa donde hemos realizado toda la programación del proceso, solo hay que establecer la conexión con el PLC desde el panel de Dispositivos de TIA Portal.

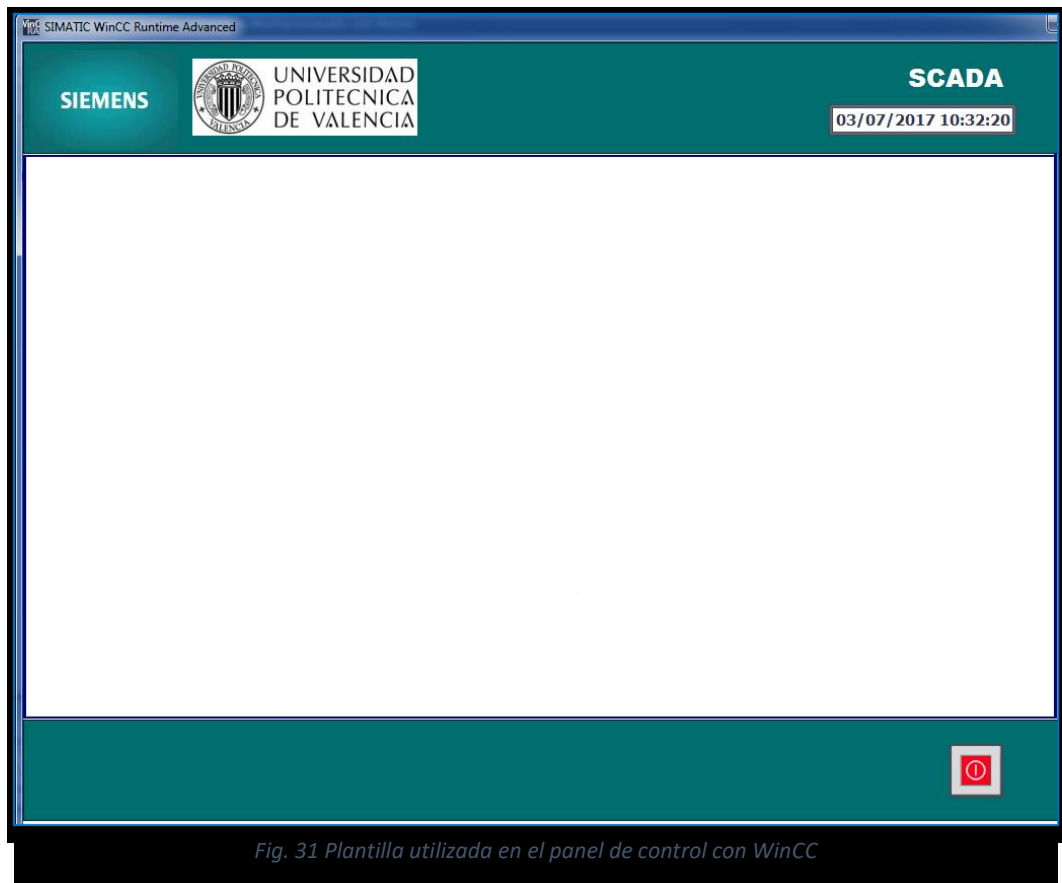
En adelante se expondrá con detalle todo el proceso de diseño del panel de control, así como que variables se han asociado a los diferentes elementos que lo componen, complementando lo ya visto en el apartado “2.3.3. Modos de funcionamiento”.

4.2.1. Plantilla

Se ha generado una plantilla que se va a utilizar en todas las imágenes del proyecto. La plantilla es una forma de añadir elementos que no van a variar entre imágenes dentro del panel de control.

La plantilla que hemos generado se puede ver en la *Fig.31*, consta de dos márgenes, superior e inferior. En el margen superior se han añadido logotipos de Siemens y de la UPV, también disponemos de fecha y hora a la izquierda de este margen.

En el margen inferior solamente se ha implementado un botón de apagado del Runtime. En los dos modos de funcionamiento se han añadido más elementos a este margen, pero serán descritos en los apartados correspondientes.



4.2.2. Imagen Principal

La imagen principal es en la que se inicial el panel de control. Como hemos mencionado, en ella podemos ver la plantilla que se ha explicado en el anterior apartado.

Básicamente esta pantalla nos permite seleccionar en que modos de funcionamiento queremos trabajar. Permitiendo el paso a modo manual o automático al pulsar alguno de los botones que hay disponibles a la derecha de la Fig.32. Se ha añadido una imagen de la estación para saber que proceso es el que estamos controlando.

Para poder realizar esto se han vinculado a ambos pulsadores las variables "Modo_Auto" y "Modo_Manu". El funcionamiento de ambas esta explicado en el apartado "2.3.3. Modos de funcionamiento".

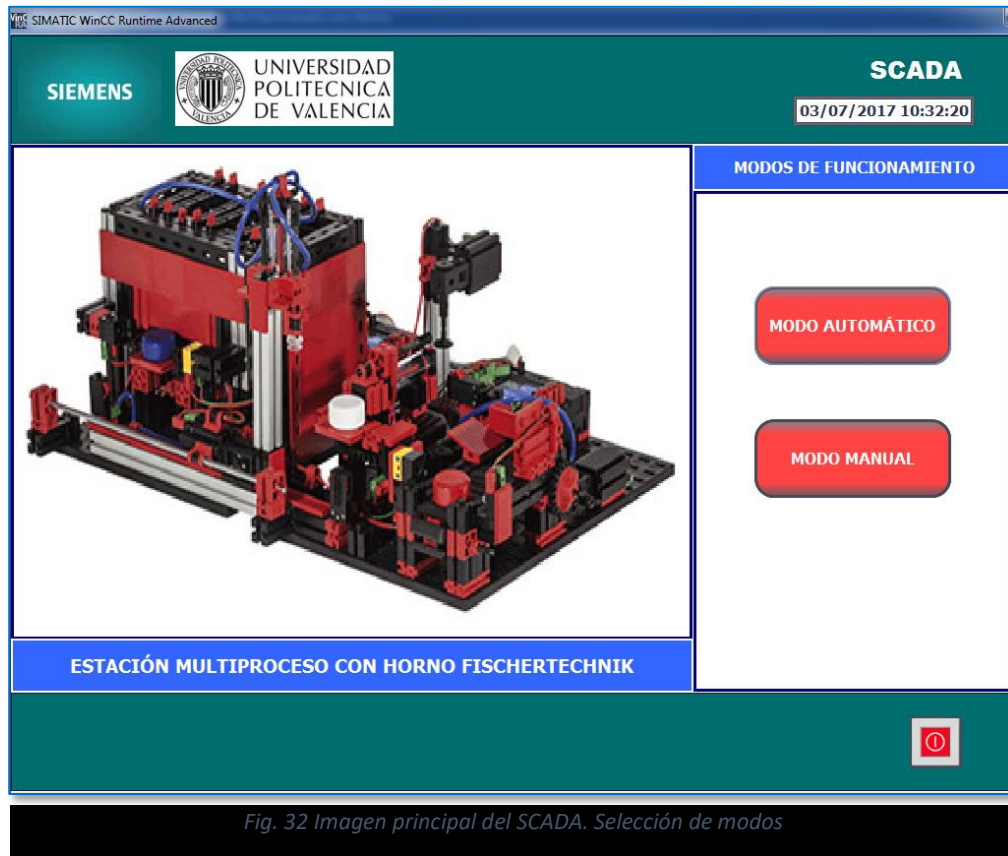


Fig. 32 Imagen principal del SCADA. Selección de modos

4.2.3. Imagen Modo Automático

Como bien se ha explicado anteriormente este modo realiza todo el proceso de manera autónoma. Para supervisar todo este proceso se ha realizado una tabla con las variables de entrada/salida de la estación, estas se han asociado a unos elementos luminosos en el panel que se activan cuando la entrada o la salida se accionan.

Se ha añadido, con objetivo de clarificar el sistema, una imagen aérea de la estación de multiprocesado. En esta fotografía se han colocado diferentes elementos luminosos, permitiéndonos indicar por donde va avanzando la pieza en cada momento o que acciones están activas a cada instante.

Además, se han colocado unos elementos de salida que indican el tiempo, en milisegundos, que la pieza permanece dentro del horno y en el proceso de serrado. Estos elementos se han vinculado con las variables "PT" de los temporizadores de cada una de estas tareas. También se

ha añadido un visualizador que nos permite contar las piezas que han sido extraídas de la estación, este se ha vinculado con la variable "CV" del contador ascendente.

En la esquina inferior izquierda se puede apreciar el pulsador de emergencia asociado a la variable "Parada_Emergencia". Este pulsador con enclavamiento detiene el proceso y las acciones que se estén ejecutando en ese momento. En el apartado "2.4.2. Parada de emergencia" hay más información al respecto.

Por último, tenemos que mencionar que el botón colocado a la izquierda del margen inferior de la plantilla sirve para volver a la imagen principal. Se ha configurado para que este botón de retroceso solo funcione cuando el proceso este parado, es decir, no haya ninguna pieza transitando por la estación. Al pulsarlo se deshabilita la variable "Modo_Auto" y por tanto el proceso queda bloqueado.

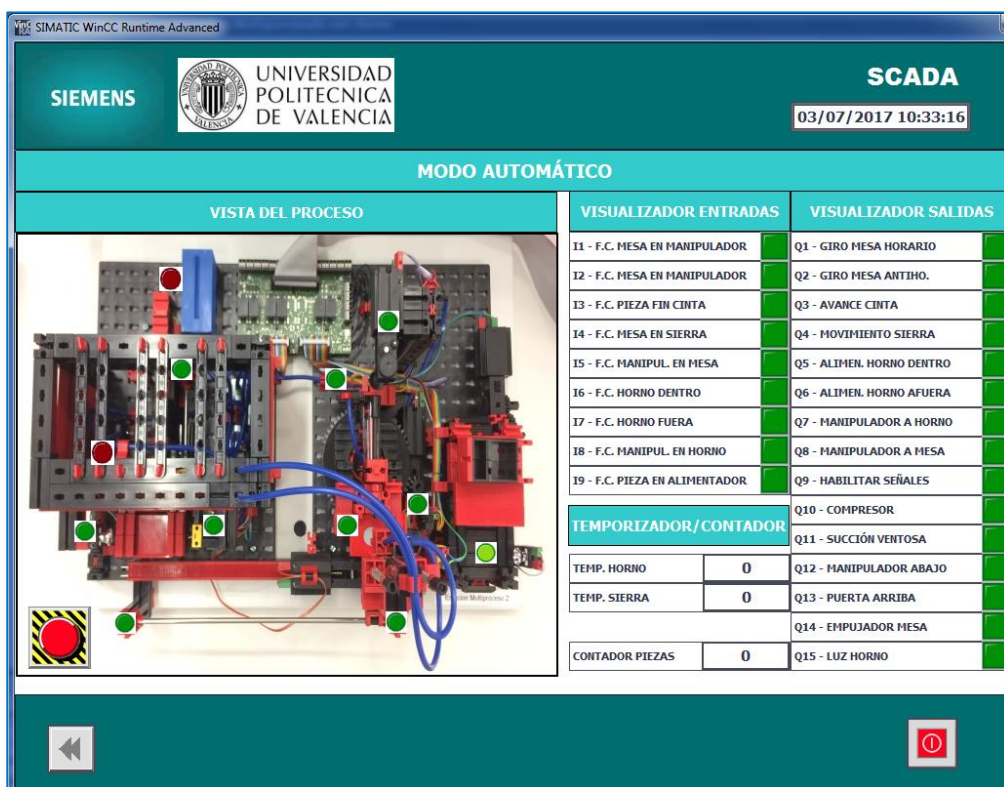


Fig. 33 Imagen del modo automático. Supervisión del proceso

4.2.4. Imagen Modo Manual

En la imagen del modo manual vemos que se ha compartimentado cada bloque en diferentes cajetines: uno para las acciones del horno, otro para las del brazo manipulador y otro para los actuadores de la mesa giratoria y cinta.

Para representar las diferentes acciones de cada proceso se han utilizado pulsadores e interruptores que mostraran claramente la función que realizan al pulsarlos.

Se han añadido interruptores para los casos en que la acción convenía que quedara enclavada. Es el caso del interruptor que acciona la subida de la puerta del horno, por ejemplo. También

hay pulsadores enclavados, como el de la succión de la ventosa. Este último tiene asociada luminosidad para saber cuándo se activa la succión.

Para la acción que habilita el avance de la cinta se han añadido sensores luminosos que nos indican si la cinta está en marcha, led verde. El led de color rojo nos indica que se ha encontrado algún objeto al final de la cinta, en este caso el pulsador no accionará el avance. El pulsador luz del horno también tiene asociado un led rojo que muestra el parpadeo de dicha luz.

El resto de pulsadores mantienen la salida activa cuando se pulsan los botones y dejan de hacerlo al soltarlos.

Como puede verse en la imagen también se ha añadido un botón de retroceso que nos devuelve a la imagen principal, es decir, al selector de modos. Al igual que en el modo automático este botón deshabilita la variable "Modo_Manu".

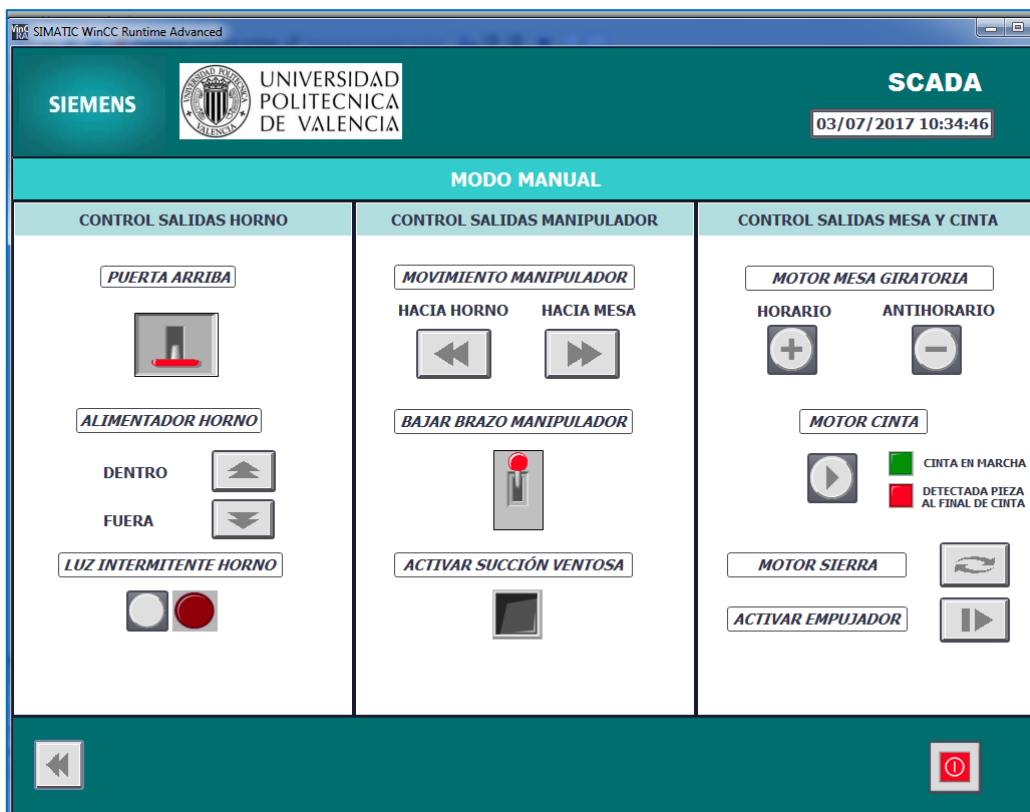


Fig. 34 Imagen del modo manual. Control de las salidas del proceso

En el Anexo I se pueden observar cómo se han programado las variables que el SCADA utiliza. No obstante, en el Capítulo 2 también se ha expuesto como funcionan estas variables.

5. CONCLUSIÓN

El proyecto ha alcanzado los objetivos marcados inicialmente. Se ha desarrollado un programa completo y sencillo, fácil de entender. Teniendo en cuenta que el autómata utilizado tiene restricciones a la hora de programar y que se ha utilizado un lenguaje que no se había visto hasta el momento se ha resuelto el problema de manera muy satisfactoria.

El resultado final del panel de control también es satisfactorio, ha cumplido con los objetivos que se había marcado. Se ha desarrollado un SCADA sencillo y fácil de comprender, capaz de supervisar y manipular todo el proceso.

A nivel personal me siento realizado con mi trabajo, he trabajado de una manera diferente, marcándome retos que he conseguido superar. A lo largo de este trabajo he adquirido aptitudes que me servirán para mi futuro, tanto académico como profesional.

En definitiva, puedo decir que me enorgullezco de mi trabajo y del resultado que se ha obtenido al final.



PRESUPUESTO

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON
HORNO MEDIANTE AUTÓMATA SIEMENS S7-1214C Y SCADA EN
WINCC

12.16.191.84

Alfonso Serrano Gallego
Tutor: Jose Vicente Salcedo

1. INTRODUCCIÓN

En este documento se muestran los costes del proyecto. Se analizará económicamente las partes que componen todo el trabajo.

Los costes aquí expuestos se clasifican según su naturaleza. De esta manera distinguimos entre costes materiales, costes de mano de obra y costes generales. Al ser parte de un trabajo académico los costes generales no se van a contabilizar con detalle.

Como en todo presupuesto de carácter industrial se añadirá al coste un porcentaje del margen de beneficio, luego se tendrán en cuenta los impuestos para elaborar el coste final.

Este documento solamente abarca el coste económico de la ejecución del trabajo. No requiere de estudios previos para garantizar la viabilidad del proyecto.

En lo que respecta a los costes de material cabe añadir que, pese a cuantificarse en el documento, su coste total será nulo. Debido a que ya se disponía de todo el material necesario en el laboratorio para llevar a cabo el proyecto.

2. COSTES MATERIALES

En este apartado se tienen en cuenta los costes de los elementos utilizados en el desarrollo del proyecto, cuantificados por el precio de venta que proporciona cada fabricante en el mercado.

Los elementos se van a cuantificar de manera general, es decir, no se va a ver en detalle cada pieza que compone los materiales utilizados.

En este apartado también se tendrán en cuenta el coste de las licencias de los programas utilizados.

TIPO	ELEMENTO	CANTIDAD (unidades)	PRECIO	IMPORTE ¹
Maquinaria	Multiestación de procesamiento 24V de <i>Fischertechnik</i>	1	632.23€	0€
Control	PLC S7-1214C de Siemens Ref. 6ES7214-1BG40-0XB0	1	288.09€	0€
	Siemens SB 1231 AE Ref. 6ES7231-4HA30-0XB0	1	60.48€	0€
	Siemens SM 1221 DC Ref. 6ES7221-1BH32-0XB0	1	166.62€	0€
	Ordenador del laboratorio	1	-	0€
Licencias	STEP 7 TIA Portal V13	1	427.43€	0€
	WinCC RT Advanced	1	2303.99€	0€
Componentes	Cable de red ethernet de 3m	2	6.10€	0€
	Bus de datos plano de 25 pines	1	12.00€	0€
	Conector hembra de 25 pines	1	1.89€	0€
SUBTOTAL				0€

Tabla 6 Presupuesto de material

¹ Para los elementos ya adquiridos el importe será de 0€

3. COSTES MANO DE OBRA

Para obtener el coste de la mano de obra se ha dividido el tiempo de trabajo en diferentes tareas. Estos costes se refieren al autor de este proyecto, con la titulación obtenida tras la finalización de este trabajo. Por este motivo, se considera un trabajador con la titulación de graduado en ingeniería tecnológica industrial.

El precio por hora de un ingeniero técnico industrial está sujeto a muchos factores, en nuestro caso vamos a hacer una estimación de unos 20€/h.

TAREA	HORAS (h)	PRECIO UNITARIO (€/h)	IMPORTE
Estudio previo	40	20	800€
Programación	160	20	3200€
Diseño del control	80	20	1600€
Pruebas y comprobaciones	20	20	400€
SUBTOTAL			6000€

4. COSTES GENERALES

Los costes generales corresponden a gastos derivados de los proyectos que se llevan a cabo en una empresa. La luz, el agua, las instalaciones o el gas son incluidos como este tipo de costes.

Este coste se obtiene como un porcentaje de la suma de los costes de material y de mano de obra. El porcentaje varía mucho en función de la empresa donde se realice el proyecto.

Se va a estimar un porcentaje de costes generales del 13% sobre los costes anteriores.

La suma de los costes por material y de mano de obra quedará:

$$0€ + 6000€ = 6000€$$

Ecuación 1

Quedando los costes de fabricación tras aplicar el porcentaje de costes generales:

$$6000€ * 1,13 = 6780€$$

Ecuación 2

5. MARGEN DE BENEFICIO

El beneficio industrial es un porcentaje aplicado sobre el coste total de fabricación del proyecto.

Representa la parte de beneficio que se lleva la empresa o espera llevarse. Este porcentaje varía mucho en función del tipo de empresa e industria al que se aplica el proyecto. En nuestro caso, aplicaremos un 6%, porcentaje que suele aplicar la Administración Pública.

$$6780\text{€} * 1.06 = 7186.80\text{€}$$

Ecuación 3

6. COSTE FINAL

Por último, se deben aplicar el porcentaje de impuestos que corresponda. Par nuestro caso aplicaremos el IVA, que actualmente tiene un valor del 21% sobre el coste total, obteniendo el coste final del proyecto.

$$7186.80\text{€} * 1.21 = 8696.03\text{€}$$

Ecuación 4

PRESUPUESTO FINAL DEL PROYECTO: 8696.03 €

Ocho mil seiscientos noventa y seis euros con tres céntimos.



PLIEGO DE CONDICIONES

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON
HORNO MEDIANTE AUTÓMATA SIEMENS S7 1214C Y SCADA EN
WINCC

Alfonso Serrano Gallego
Tutor: Jose Vicente Salcedo

1. INTRODUCCIÓN

En este documento se establecen las condiciones que deben de cumplirse para que el presente proyecto pueda ejecutarse de manera satisfactoria.

Se hablará sobre el pliego de condiciones generales y particulares.

Dada la finalidad de este proyecto en las condiciones particulares del pliego solamente se abarcarán las de índole técnica, dividiéndola a su vez en dos subapartados.

1) Especificaciones de material y equipos:

Se incluirán las características de los materiales utilizados, haciendo referencia a la norma y los reglamentos.

2) Especificaciones de ejecución:

Se incluirán los procedimientos llevados a cabo a lo largo de este proyecto.

2. PLIEGO DE CONDICIONES GENERALES

Para la ejecución de este proyecto se deberán seguir las especificaciones técnicas descritas en la memoria y anexos, debiendo aplicar los requerimientos descritos en el presente documento.

Los elementos normalizados deben ser los mencionados en la memoria, cumpliendo con las propiedades y características expuestas.

3. PLIEGO DE CONDICIONES PARTICULARES

3.1. CONDICIONES DE ÍNDOLE TÉCNICA

3.1.1. Especificaciones de material y equipos

Normativa general

Las instalaciones que se requieren para este proyecto son principalmente componentes eléctricos. Se encuentran conectados a la red de corriente alterna, 230V y 50Hz. Por lo tanto, se rigen por el Reglamento Electrotécnico de Baja Tensión (RBT) y todas sus Instrucciones Técnicas Complementarias.

Se deberán considerar las siguientes normas UNE:

ITC-BT-18 Instalaciones de puesta a tierra

ITC-BT-19 Instalaciones interiores o receptoras. Prescripciones generales

ITC-BT-20 Instalaciones interiores o receptoras. Sistemas de instalación

ITC-BT-21 Instalaciones interiores o receptoras. Tubos y canales protectoras

ITC-BT-22 Instalaciones interiores o receptoras. Protección contra sobreintensidades

ITC-BT-23 Instalaciones interiores o receptoras. Protección contra sobretensiones

ITC-BT-24 Instalaciones interiores o receptoras. Protección contra los contactos directos e indirectos

ITC-BT-51 Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios

UNE 20 514 1M: reglas de seguridad para aparatos electrónicos y aparatos con ellos relacionados de uso doméstico o uno general análogo conectado a una red de energía.

Ordenador o PC

El ordenador utilizado deberá disponer de al menos estos elementos: CPU, monitor, ratón, teclado y conexión ethernet. Adicionalmente puede tener entrada de puerto USB.

Las características técnicas mínimas deben satisfacer: procesador Core™ i5-3320M 3.3 GHz o similar, 8 GB al menos de memoria principal, disco duro 300 GB SSD, display de pantalla ancha (1920 x 1080) y sistema operativo Windows 7 o superior.

Autómata Programable

Se utilizará el PLC Siemens S7-1214C AC/DC/Rly compatible con el software STEP 7 TIA Portal. El autómata esta alimentado mediante una fuente de alimentación de corriente alterna de 230V conectada a la red, aunque también permite la alimentación a 120V en alterna.

Debe cumplir la norma DIN IEC 60068-2-6 de funcionamiento en servicio.

3.1.2. Especificaciones de ejecución

Automatización

Antes de finalizar la automatización (también durante la realización de la misma) se deben realizar comprobaciones de las respuestas que tiene la planta multiproceso, realizando las oportunas modificaciones si es necesario.

Control

El procedimiento siguiente será la comprobación del panel de control en los diferentes modos de funcionamiento. Una vez se haya comprobado que todo está bien vinculado, que responde como debería y que la supervisión y control se llevan a cabo sin problemas se iniciarán las pruebas finales.

Ensayos

Con el montaje y el panel de control en funcionamiento se procederá a realizar ensayos, abordando situaciones en que no se siga el funcionamiento normal. Así se podrá comprobar que se han tenido en cuenta la gran mayoría de sucesos en la programación.

Pruebas finales

Se realizarán pruebas de los modos de funcionamiento. Cuando se superen estas pruebas el proyecto se dará por finalizado.

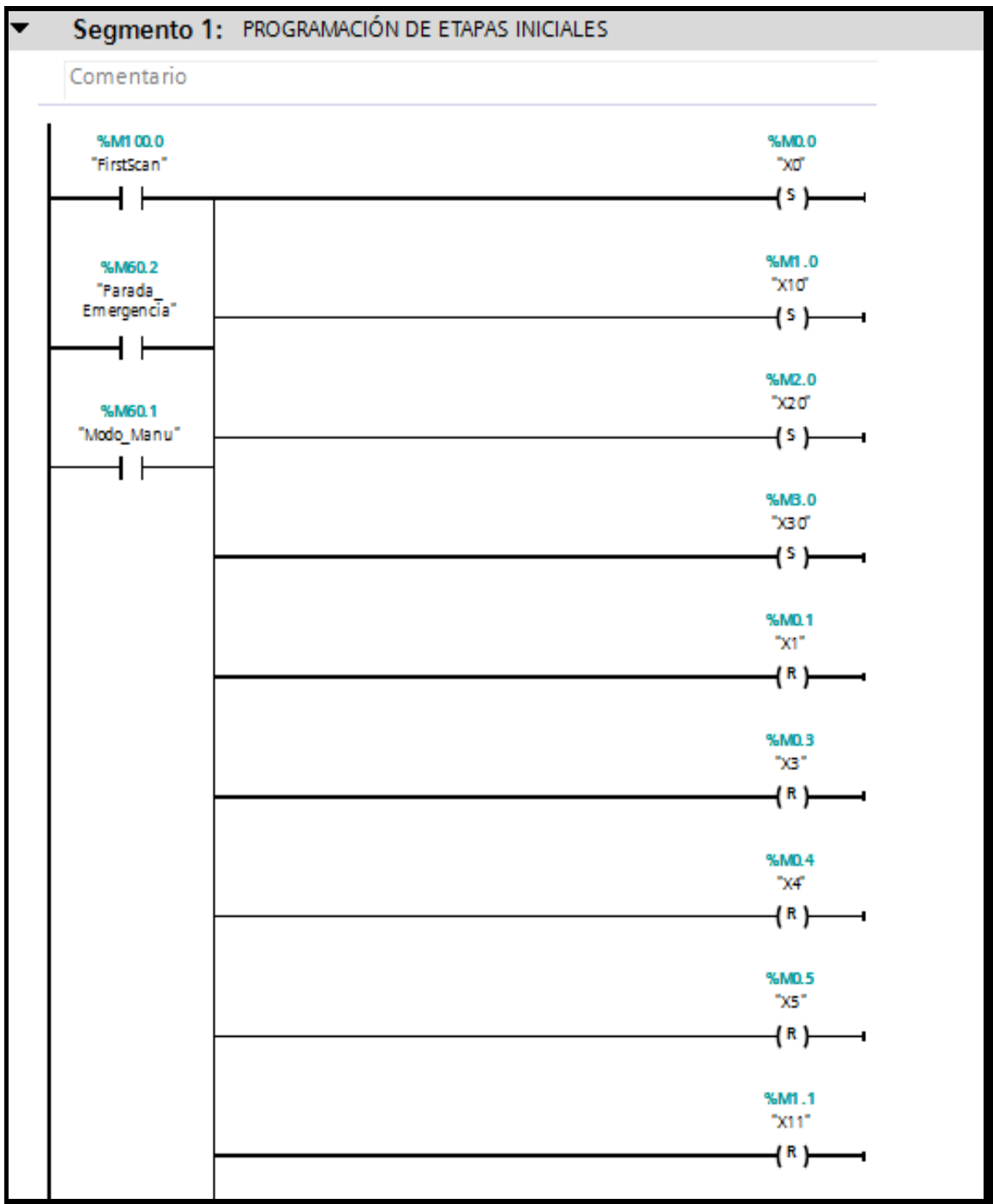


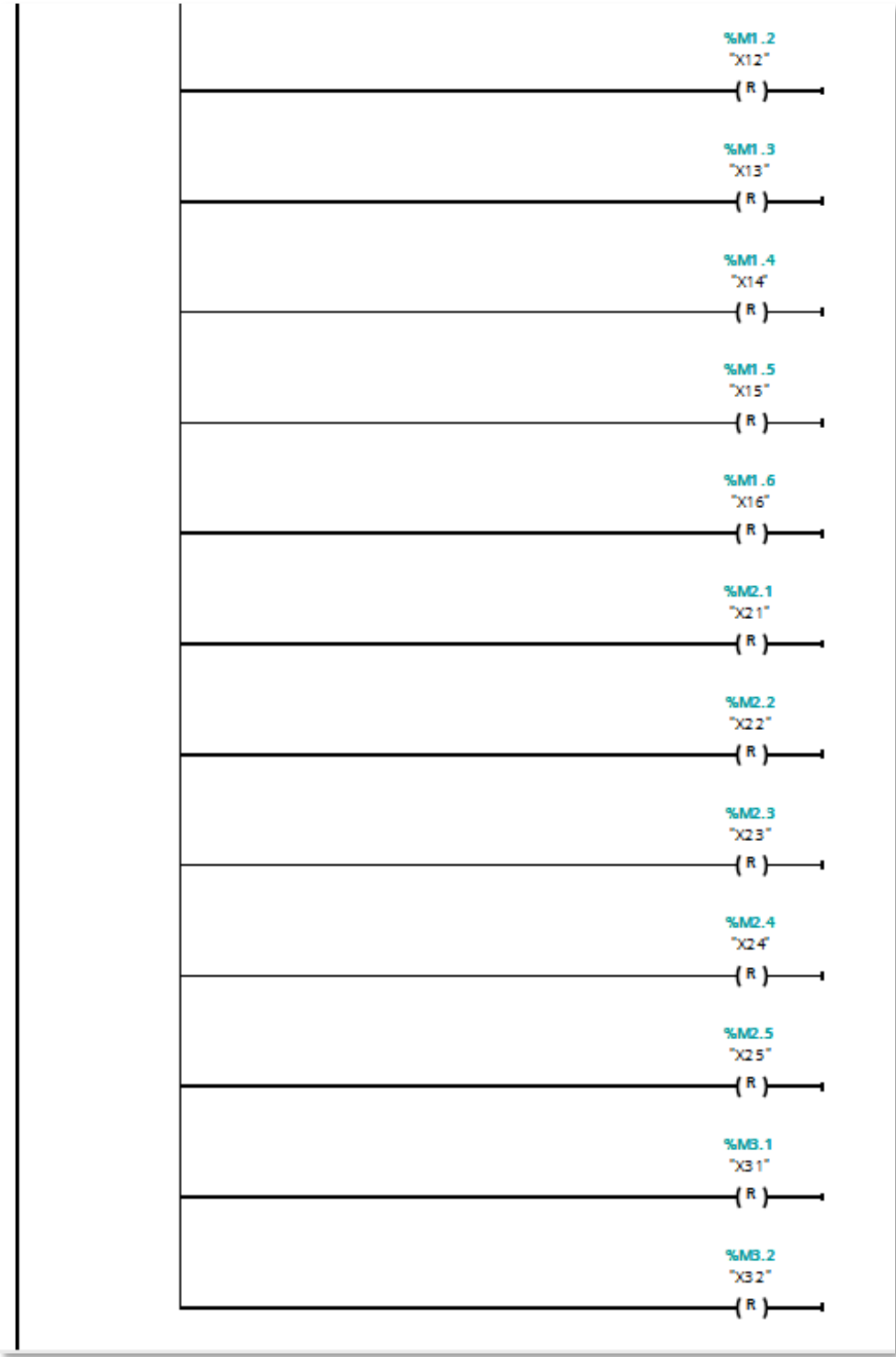
ANEXO I. PROGRAMAS

PROYECTO DE AUTOMATIZACIÓN DE PLANTA MULTIPROCESO CON
HORNO MEDIANTE AUTÓMATA SIEMENS S7-1214C Y SCADA EN
WINCC

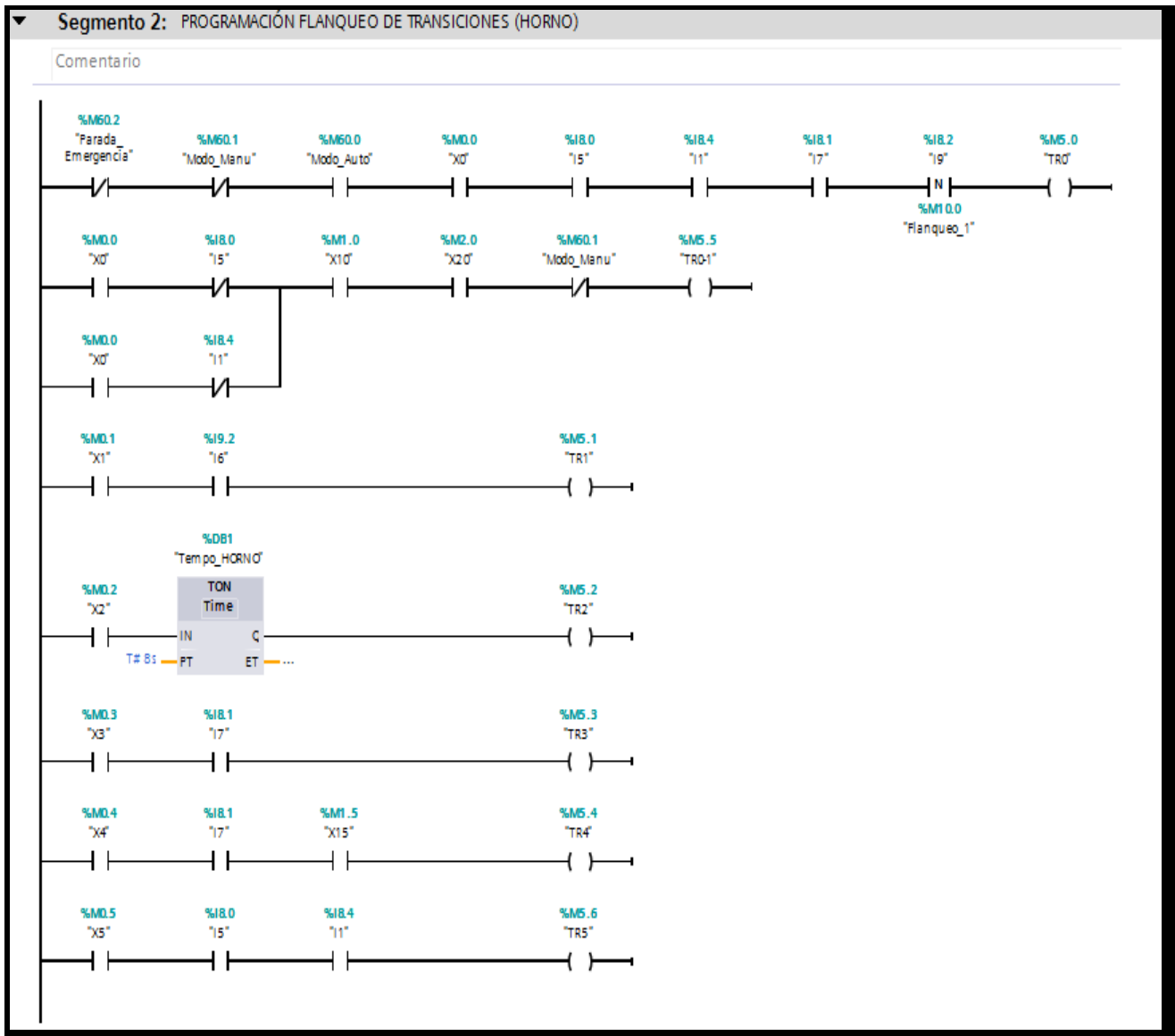
Alfonso Serrano Gallego
Tutor: Jose Vicente Salcedo

1. SEGMENTO 1

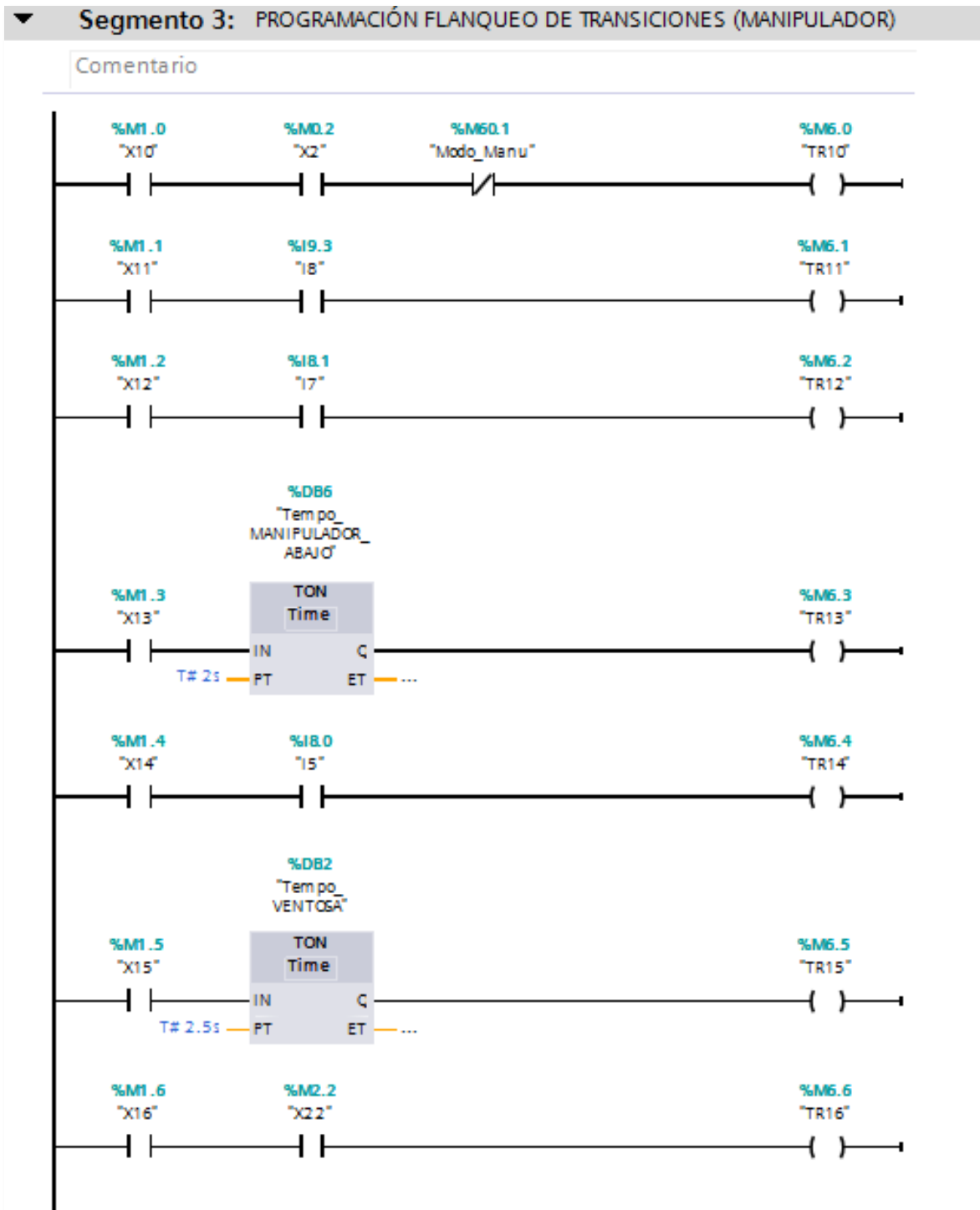




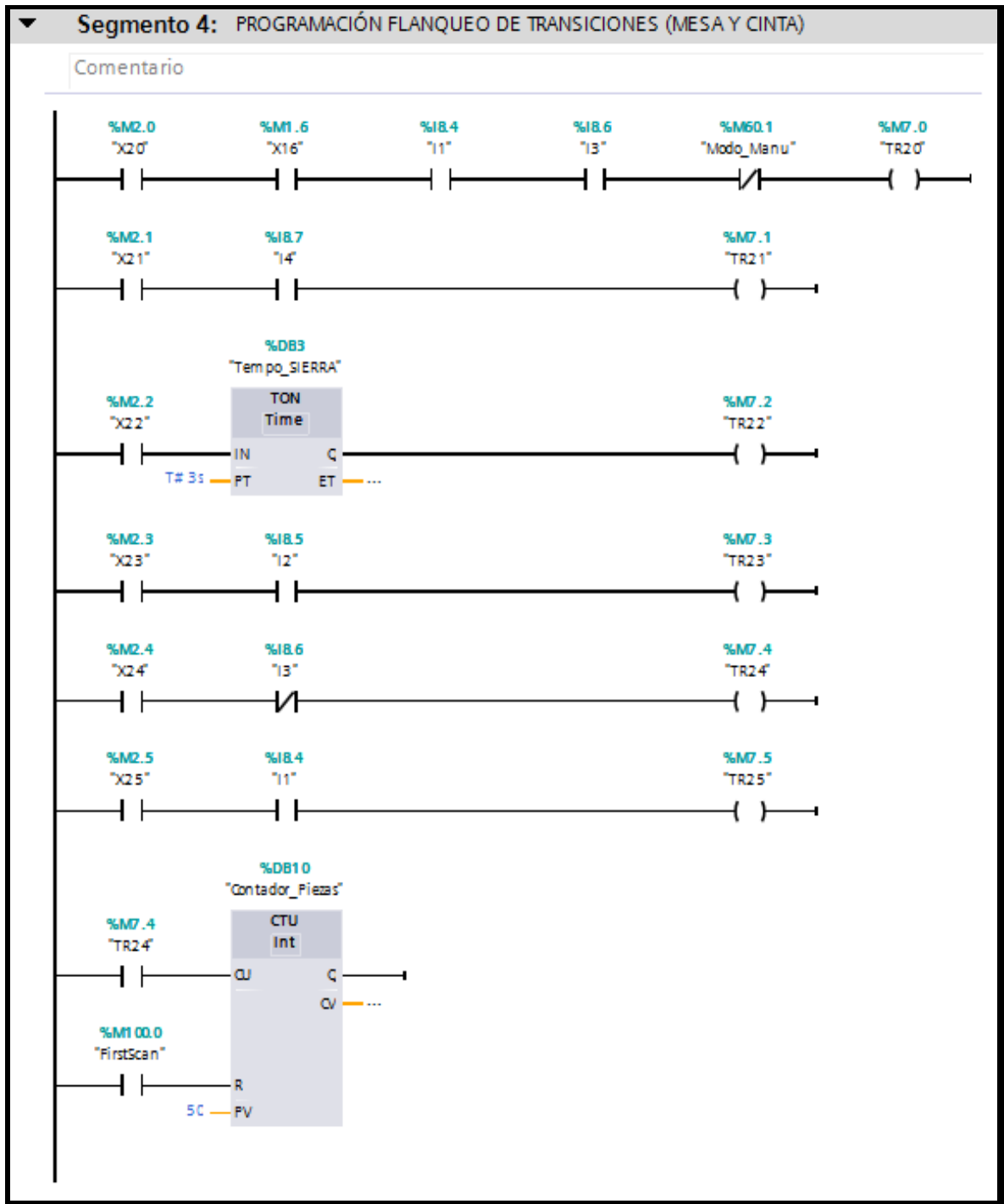
2. SEGMENTO 2



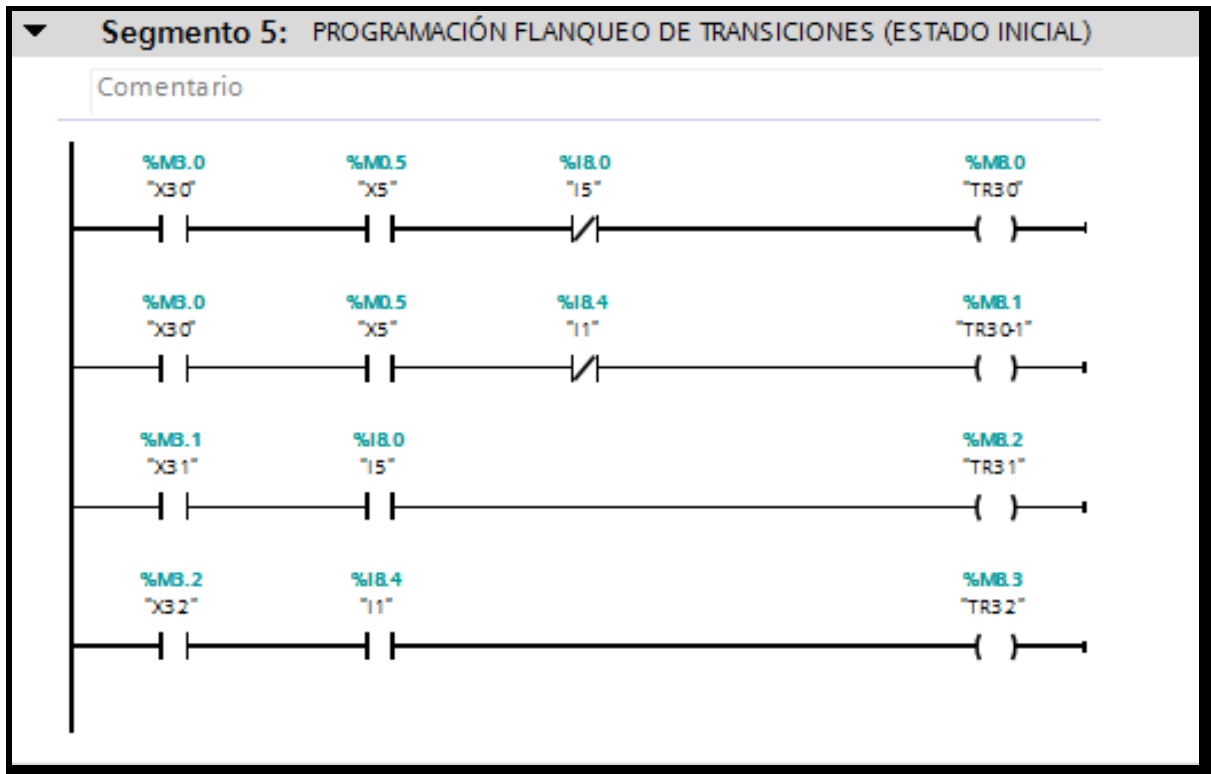
3. SEGMENTO 3



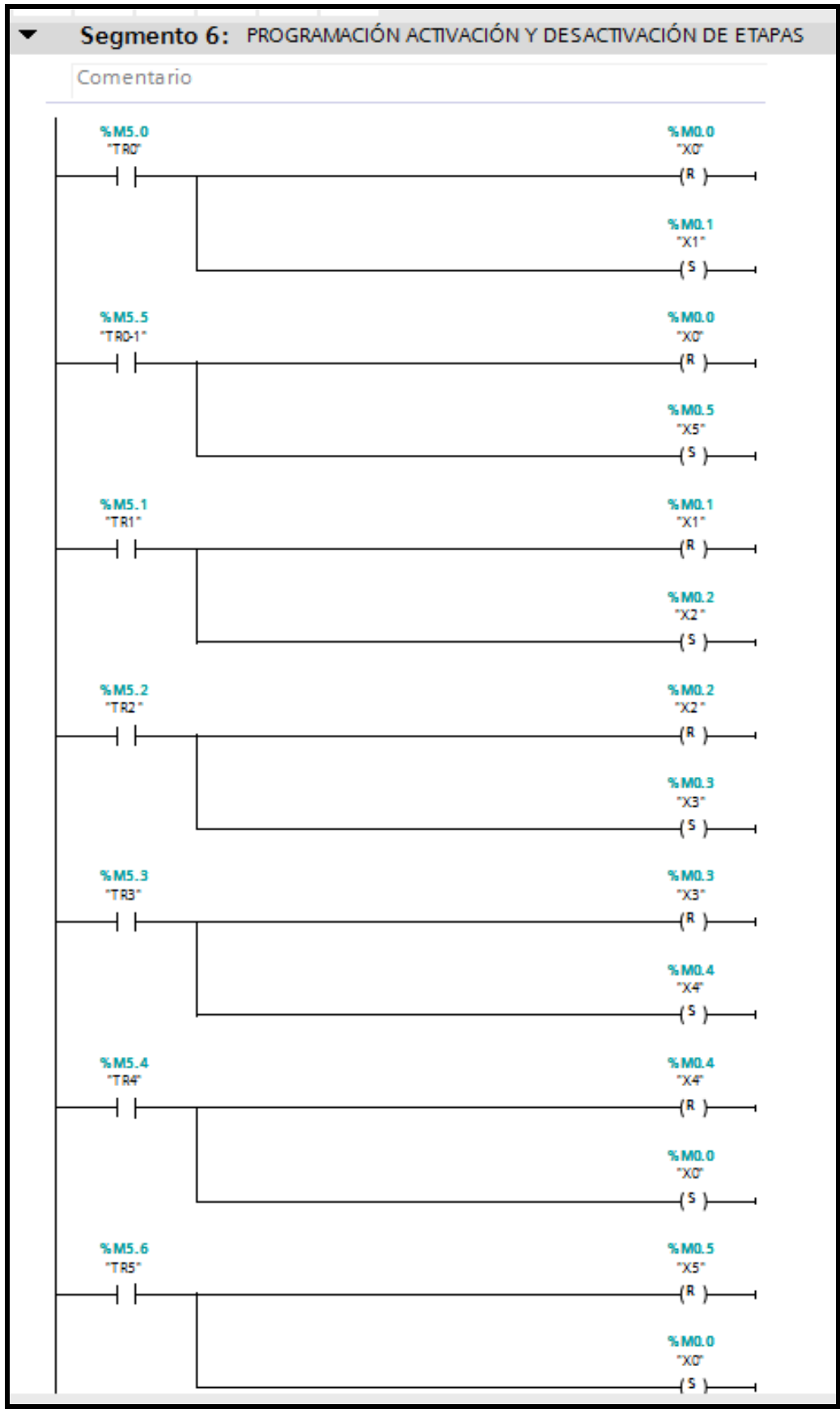
4. SEGMENTO 4

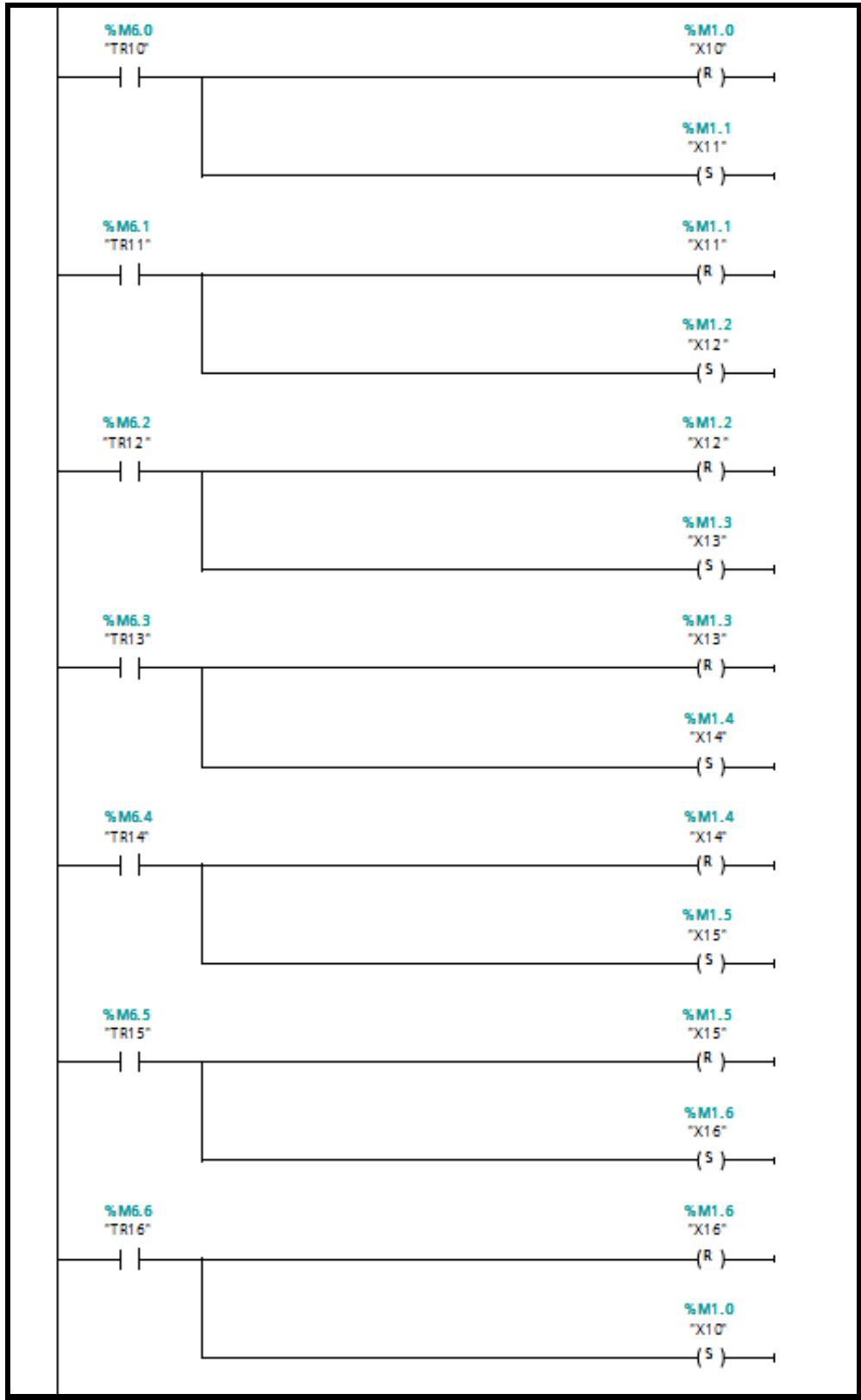


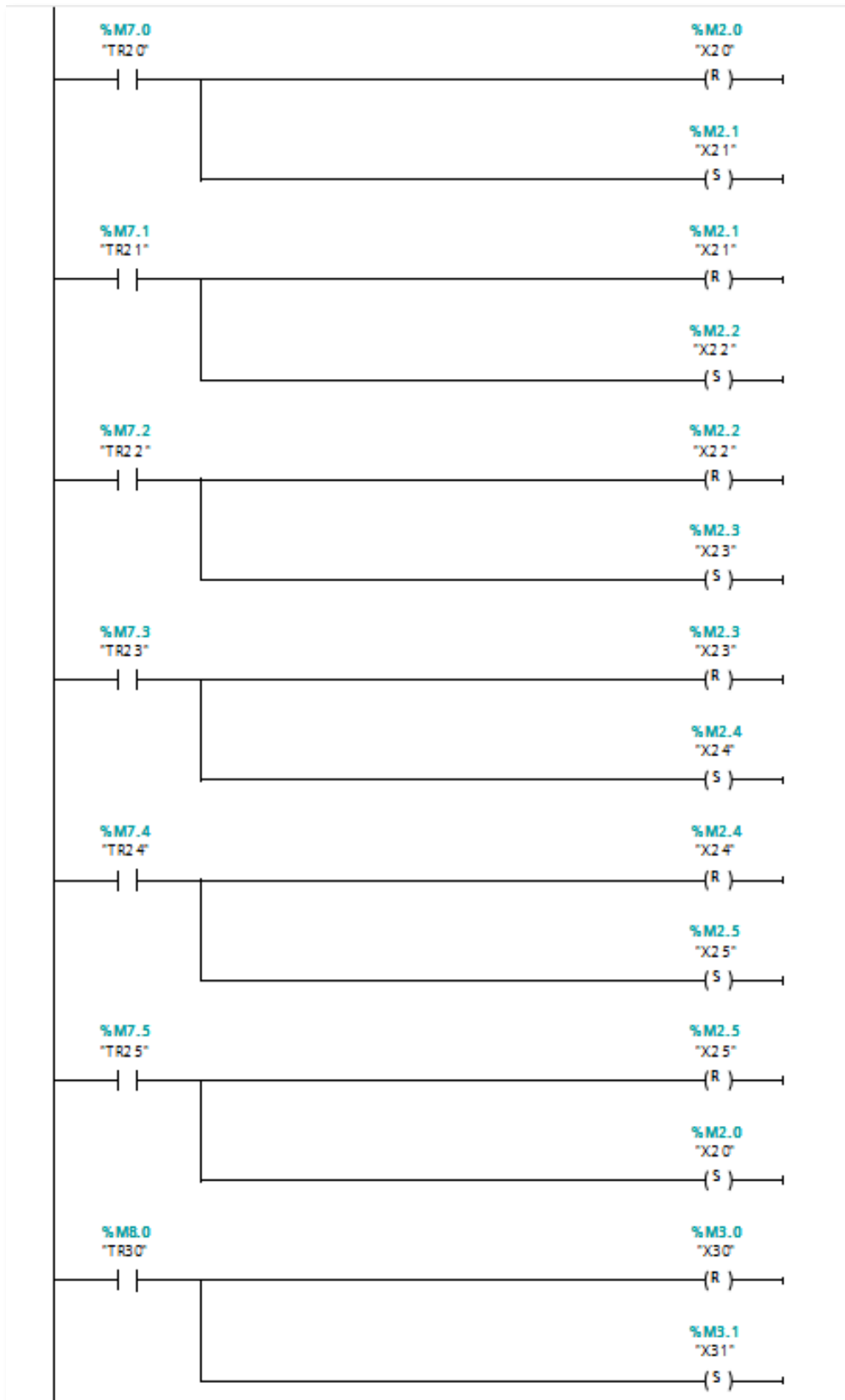
5. SEGMENTO 5

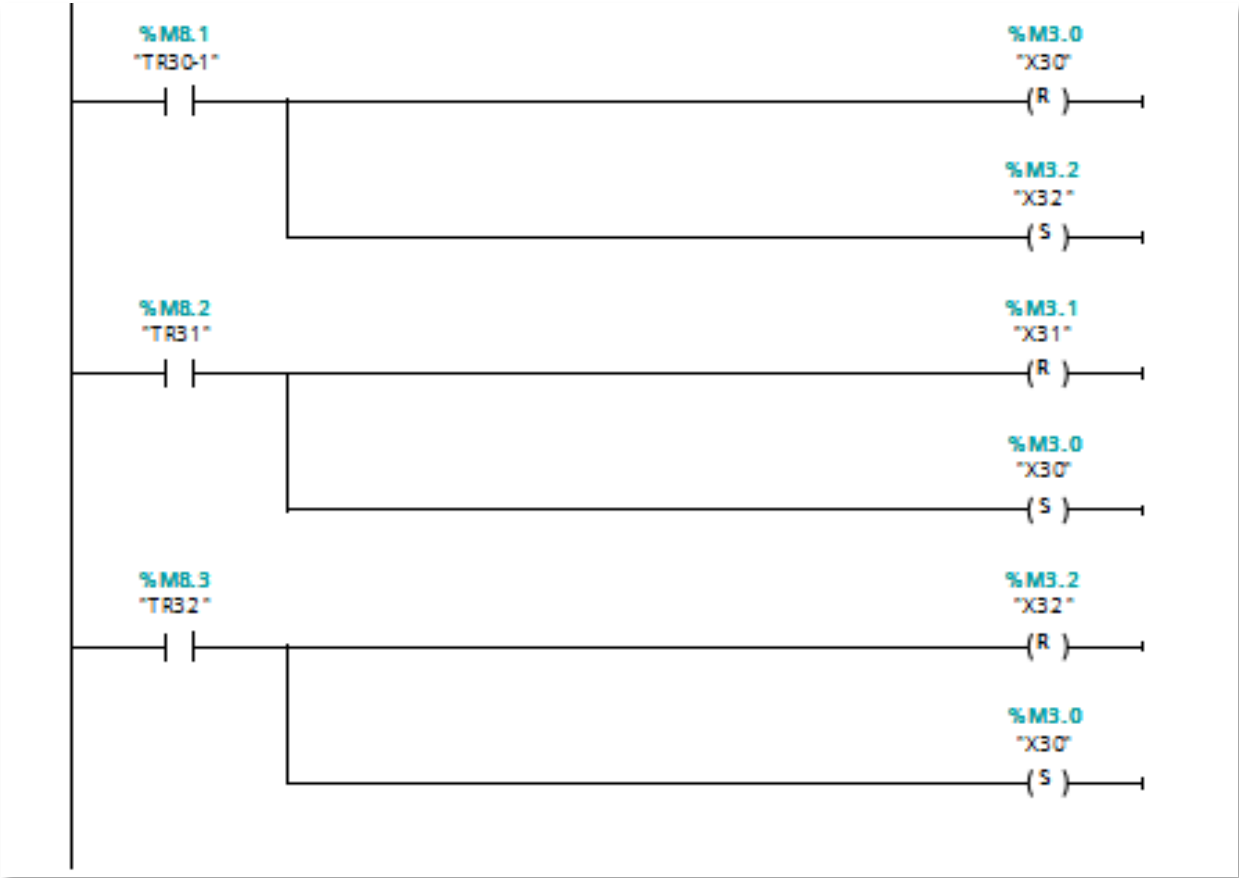


6. SEGMENTO 6

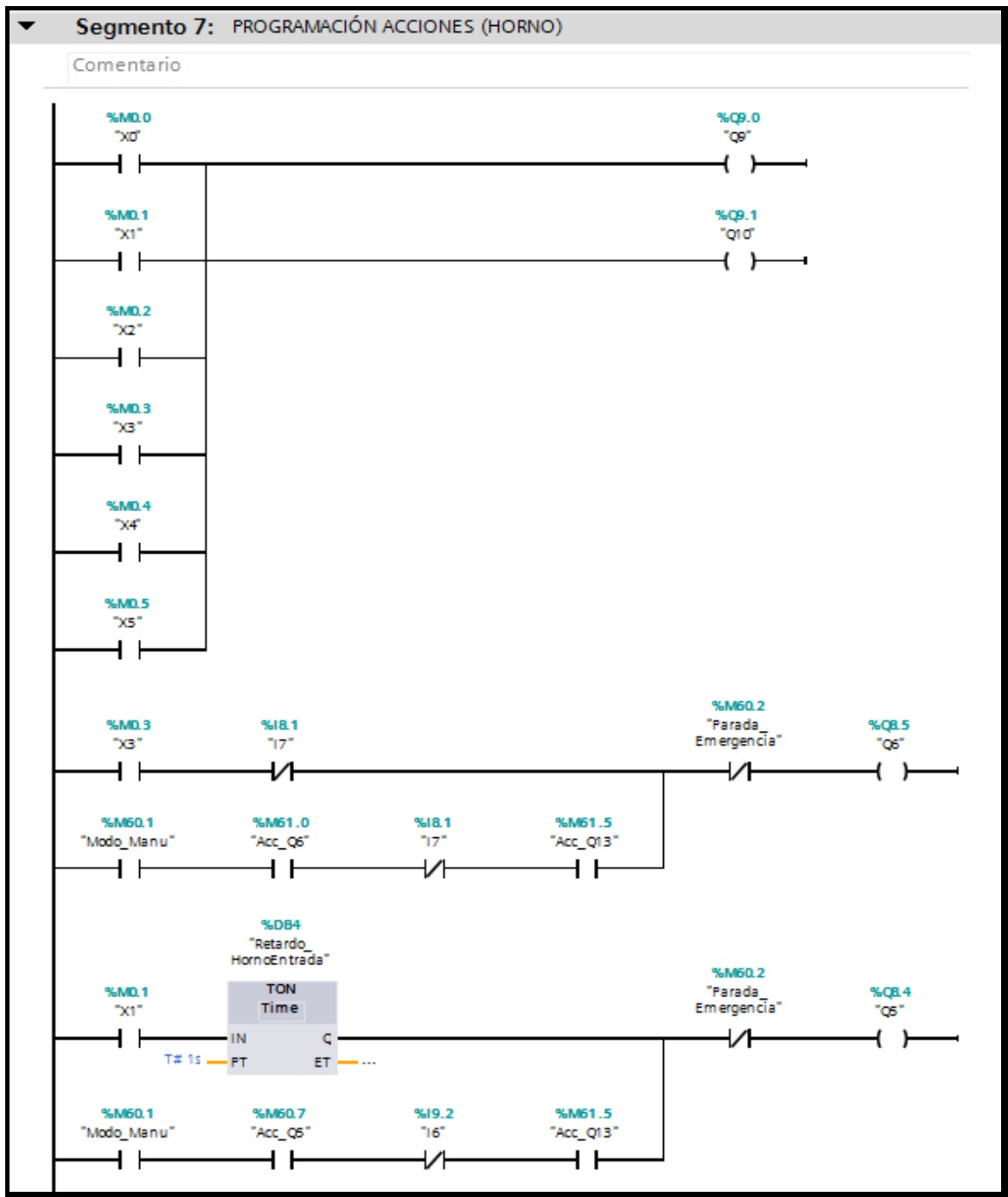


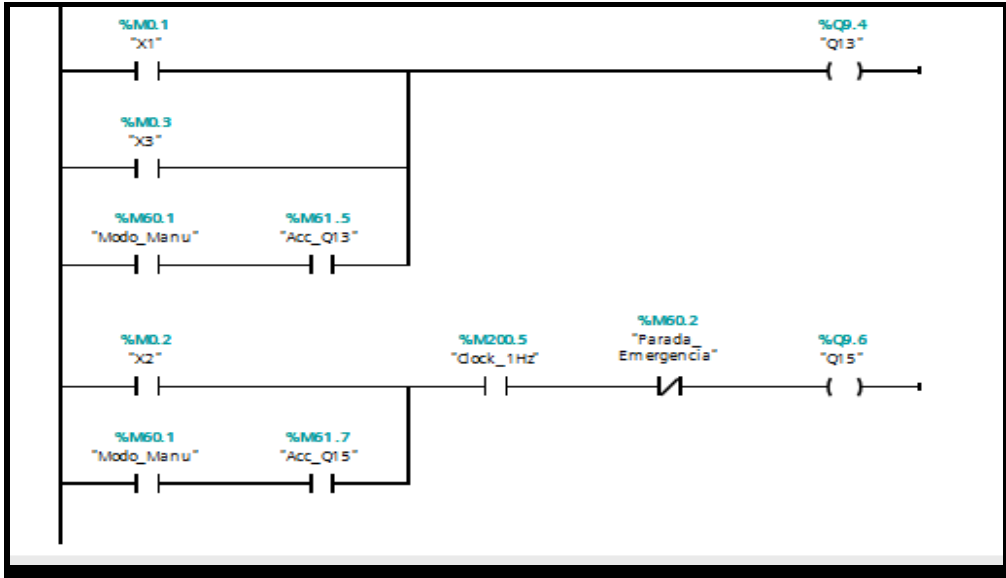




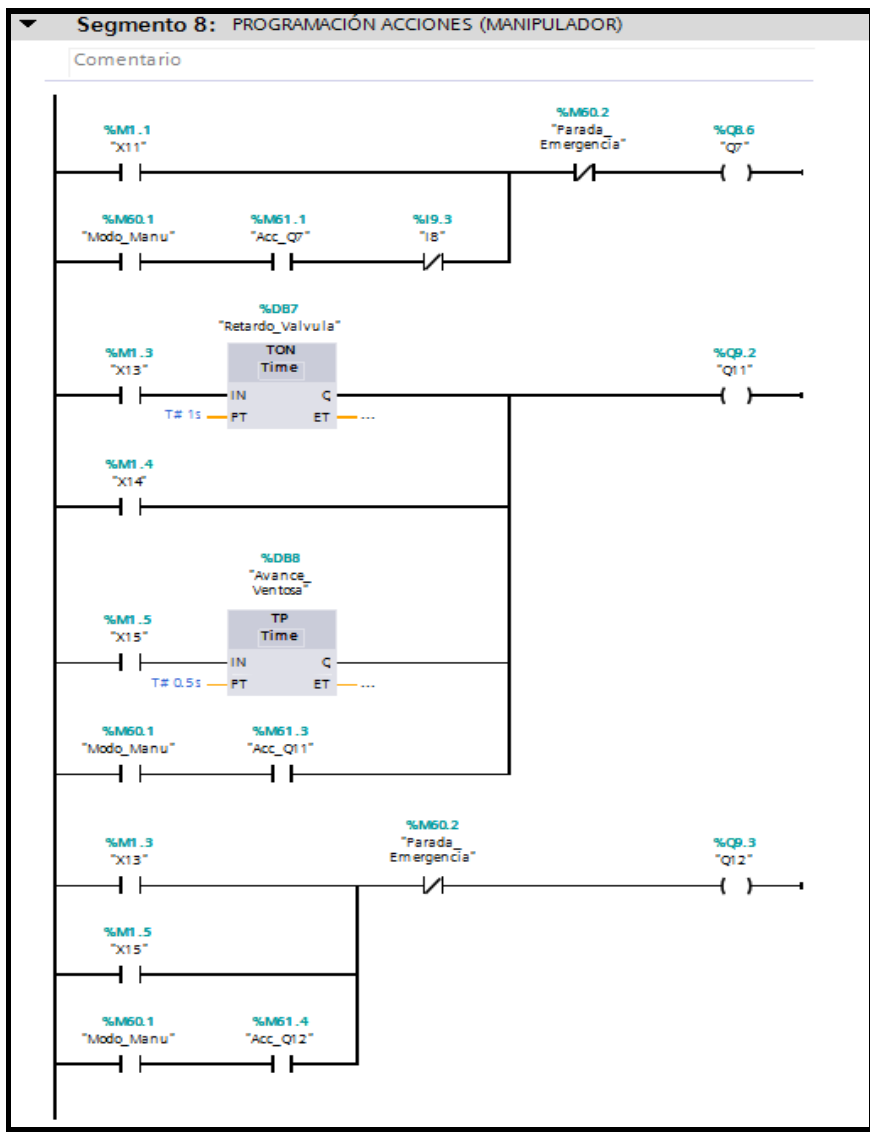


7. SEGMENTO 7

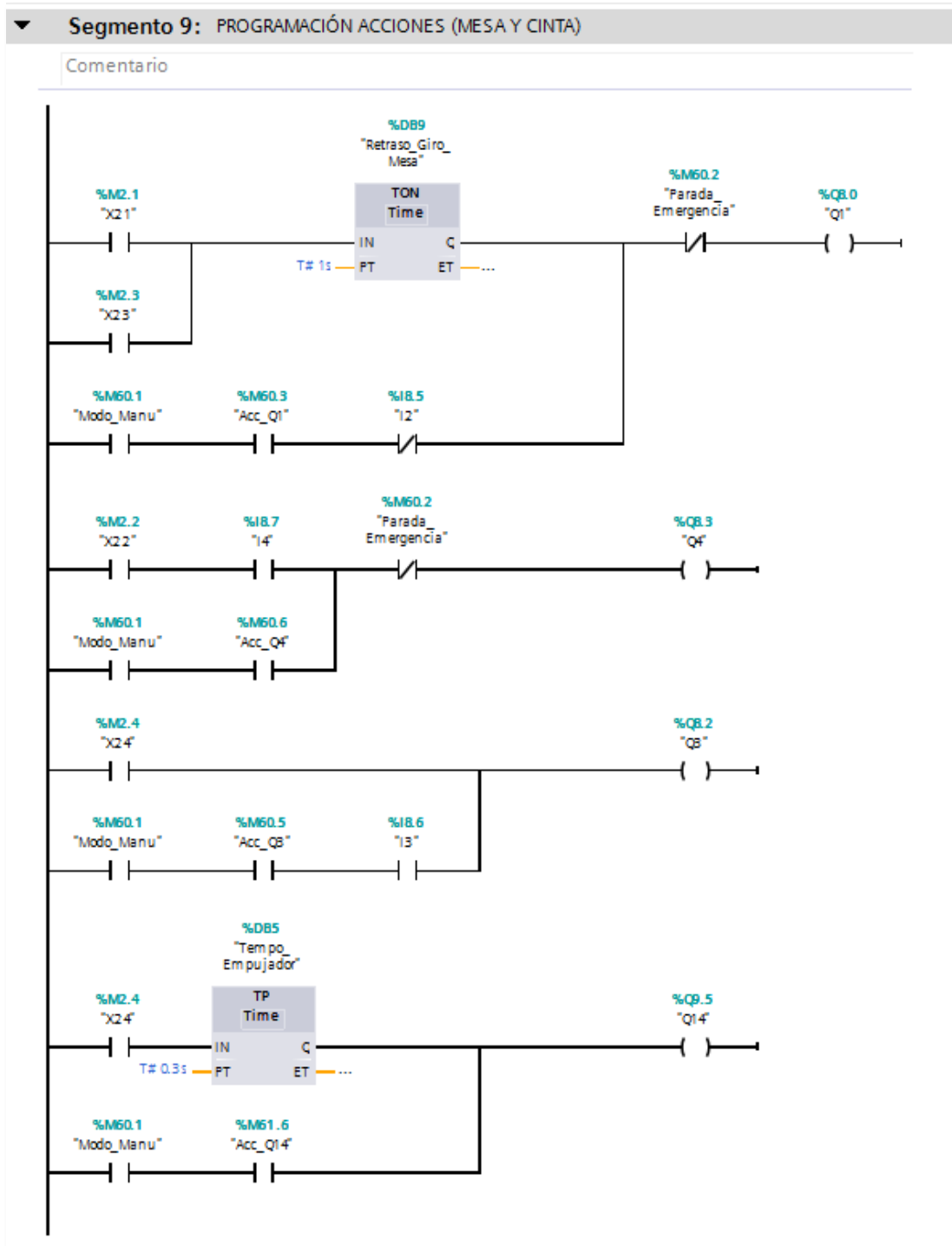




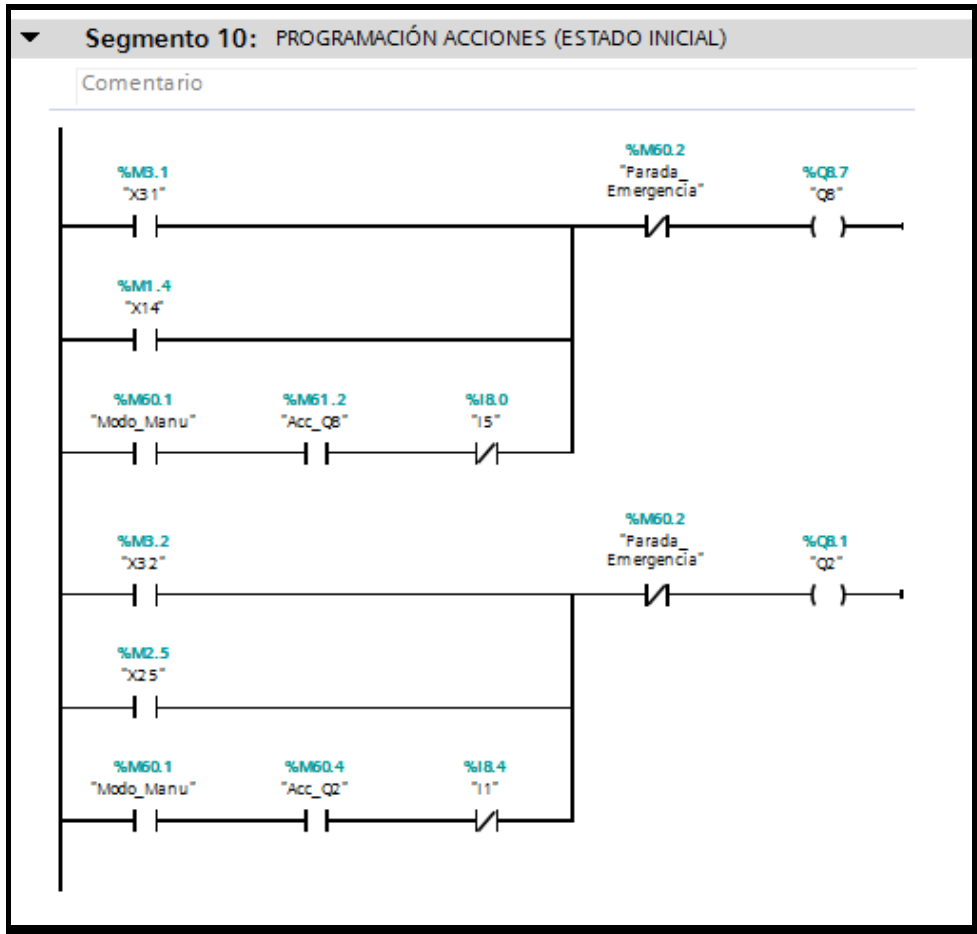
8. SEGMENTO 8



9. SEGMENTO 9



10. SEGMENTO 10



REFERENCIAS

<https://support.industry.siemens.com/cs/start?lc=es-ES>

http://www.fisica.ru/dfmg/teacher/archivos/progr_grafcet.pdf, autor desconocido, visitada el 05/2017.

<https://emacsan.wikispaces.com/file/view/conexion+wincc+advance+con+s7-1200.pdf>, autor Jose María Hurtado Torres, I.E.S. Himilice-Linares.

<http://w3.siemens.com/mcms/automation-software/en/tia-portal-software/wincc-tia-portal/wincc-tia-portal-runtime/Pages/default.aspx>

Manuales de usuario de SIEMENS: “SIMATIC S7 Controlador programable S7-1200 Manual de sistema” publicado el 06/2015 y “SIMATIC S7-1200 *Getting Started* del S7-1200 (primeros pasos)” publicado el 11/2009.

Proyecto final de grado “PROYECTO DE AUTOMATIZACIÓN Y CONTROL MEDIANTE SCADA DE UN SISTEMA DE ALMACENAMIENTO Y RECUPERACIÓN INDUSTRIAL DE PIEZAS”, autor Jose Cabot Almela, publicado en el curso 2014-15.