



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**PROYECTO DE AUTOMATIZACIÓN DE UN
PROTOTIPO DE LABORATORIO MEDIANTE
AUTÓMATAS SIEMENS S7 1200 Y
SCHNEIDER M241, Y PANTALLA HMI
TP-2507-E.**

AUTOR: DANIEL MORENO GARRIDO

TUTOR: JOSÉ VICENTE SALCEDO ROMERO DE ÁVILA

Curso Académico: 2018-19

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Agradecimientos

Quiero aprovechar este espacio para agradecer la ayuda de mi tutor José Vicente y de mi padre, sin ellos no habría sido posible este proyecto. También agradecer a Carmen por ayudarme a redactar este documento. Y por último agradecer a mi familia y a mis amigos por apoyarme durante el grado y hacérmelo más ameno.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Resumen

En este proyecto se lleva a cabo la automatización coordinada de dos subprocesos industriales que suponen dos fases consecutivas en la fabricación de un producto. Estos subprocesos están representados por dos prototipos de laboratorio, uno simulando un proceso de estampado y otro representando un proceso de fresado y taladrado. Para ello se utilizan los autómatas Siemens S7 1200 para el proceso de estampación y Schneider Modicon M241 para el otro proceso. También se cuenta con una pantalla HMI TP-2507-E de la marca HCFA con la que se realiza una aplicación SCADA para la monitorización y control de ambos procesos. Además, la tableta HMI es utilizada para coordinar los subprocesos mediante la programación de una macro.

Palabras clave: Automatización, programación, procesos industriales, autómatas, pantalla HMI, SoMachine, TIA Portal, HCTDesigner, Grafcet, Diagrama de contactos, SCADA.

Resum

En aquest projecte es duu a terme l'automatització coordinada de dues subprocessos industrials que suposen dues fases consecutives en la fabricació d'un producte. Aquests subprocessos estan representats per dos prototips de laboratori, un simulant un procés d'estampat i un altre representant un procés de fresat i trepat. Per a això s'utilitzen els autòmats Siemens S7 1200 per al procés d'estampació i Schneider Modicon M241 per a l'altre procés. També es compta amb una pantalla HMI TP-2507-E de la marca HCFA amb la qual es realitza una aplicació SCADA per al monitoratge i control de tots dos processos. A més, la tauleta HMI és utilitzada per a coordinar els subprocessos mitjançant la programació d'una macro.

Paraules clau: Automatització, programació, processos industrials, autòmats, pantalla HMI, SoMachine, TIA Portal, HCTDesigner, Grafcet, Diagrama de contactes, SCADA.

Abstract

This project carries out the automation of two coordinated industrial subprocesses that involve two consecutive phases in the manufacture of a product. These subprocesses are represented by two lab prototypes, one simulating a stamping process and the other representing a milling and drilling process. For this purpose, a Siemens S7 1200 PLC is used for the stamping process and a Schneider Modicon M241 PLC is used for the other process. There is also a TP-2507-E HMI display of the HCFA brand with which a SCADA application is made for monitoring and control of both processes. In addition, the HMI tablet is used to coordinate both subprocesses by programming a macro.

Keywords: Automation, programming, industrial processes, automatons, HMI display, SoMachine, TIA Portal, HCTDesigner, Grafcet, Contact diagram, SCADA.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Índice del proyecto

1. Memoria descriptiva
2. Pliego de condiciones
3. Presupuesto
4. Anexos

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

1- MEMORIA DESCRIPTIVA

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Índice de la memoria descriptiva

1. Introducción	1
2. Objetivos	2
3. Conceptos previos	3
3.1 Autómatas	3
3.2 Pantalla HMI	8
3.3 Lenguajes de programación	9
3.4 Variables bool y variables Word	13
4. Materiales	14
4.1 Modicon M241	14
4.2 SIMATIC S7-1200	16
4.3 TP2507-E	17
4.4 SoMachine	18
4.5 TIA Portal	19
4.6 HCTDesigner 2	19
4.7 Prototipo de estampadora	19
4.8 Prototipo de proceso de fresado y taladrado	20
4.9 Fuente de alimentación	21
5. Desarrollo del proyecto	21
5.1 Diseño proceso de la estampadora	21
5.1.1 Funcionamiento del proceso	22
5.1.2 Iniciar nuevo proyecto en TIA Portal	23
5.1.3 Programa de la estampadora	27
5.1.4 Transferencia del programa al autómatas	32
5.2 Diseño de proceso de fresado y taladrado	33
5.2.1 Funcionamiento del proceso	33
5.2.2 Iniciar nuevo proyecto en SoMachine	36
5.2.3 Programa del prototipo de fresado y taladrado	40
5.2.4 Transferencia del programa al autómatas	44
5.3 Diseño de la aplicación SCADA	44
5.3.1 Inicio de nuevo proyecto en HCTDesigner 2	45
5.3.2 Diseño aplicación para prototipo estampadora	45
5.3.3 Diseño aplicación para prototipo fresado y taladrado	47
5.3.4 Otras prestaciones de la aplicación HMI	50
5.3.5 Transferencia de la aplicación a la pantalla HMI	54

5.4 Sincronización de ambos procesos.....	55
5.4.1 Creación de macro en la pantalla HMI.....	56
5.4.2 Programación de la macro.....	56
6. Conclusiones	61
7. Bibliografía	62

Índice de ilustraciones

Ilustración 1. Montaje final	2
Ilustración 2. Representación simplificada de un proceso automatizado	3
Ilustración 3 . Ejemplo de autómatas modular de la marca Allen-Bradley	4
Ilustración 4. Ejemplo de autómatas compacto de la marca Omron.....	4
Ilustración 5. Representación esquematizada de un PLC.....	6
Ilustración 6. Ejemplo de microcontrolador (Espressif Esp8266).....	7
Ilustración 7. Pantalla HMI de la marca Siemens (fuente: Catálogo de Siemens).....	8
Ilustración 8. Tipos de contactos en un diagrama ladder	9
Ilustración 9. Tipos de bobinas en diagrama ladder	9
Ilustración 10. Conexiones de un temporizador TON.....	10
Ilustración 11. Conexiones de un contador CTU	10
Ilustración 12. Componentes de un graficet	11
Ilustración 13. Graficet con una bifurcación en OR.....	12
Ilustración 14. Graficet con bifurcación en AND	13
Ilustración 15. Autómatas TM241CE40R instalado en el laboratorio de automatización del DISA	15
Ilustración 16. S7-1200 CPU 1214C instalado en el laboratorio de automatización del DISA	17
Ilustración 17. Pantalla HMI TP2507-E.....	18
Ilustración 18. Prototipo de máquina estampadora	20
Ilustración 19. Prototipo de proceso de fresado y taladrado	20
Ilustración 20. Fuente de alimentación Promax FAC-363B	21
Ilustración 21. Pantalla de inicio en TIA Portal V13	24
Ilustración 22. Ventana de vista de proyecto en TIA Portal.....	24
Ilustración 23. Selección de CPU en TIA Portal.....	25
Ilustración 24. Vista de proyecto.....	26
Ilustración 25. Ventana de dispositivos y redes	26
Ilustración 26. Ejemplo de Graficet para funciones lógicas	27
Ilustración 27. Graficet prototipo estampadora	29
Ilustración 28. Graficet de emergencia de la estampadora.....	29
Ilustración 29. Variables del PLC en TIA Portal	32
Ilustración 30. Menú de carga de programa en TIA Portal	32
Ilustración 31. Actuadores del prototipo de fresado y taladrado.....	33
Ilustración 32. Ventana de inicio de SoMachine.....	36
Ilustración 33. Ventana de configuración de proyecto de SoMachine	37
Ilustración 34. Selección de dispositivo	37
Ilustración 35. Ventanas de entradas y salidas digitales en SoMachine.....	38
Ilustración 36. Creación de un nuevo POU	38
Ilustración 37. Menú de configuración de un nuevo POU	39
Ilustración 38. Selección de módulos de E/S analógicas	39
Ilustración 39. Asignación de IP al PLC	40
Ilustración 40. creación de una transición en un POU	40
Ilustración 41. POU Cintai	42
Ilustración 42. Parte del código del POU Acciones	42
Ilustración 43. Declaración de variables globales en SoMachine	43
Ilustración 44. Declaración de variables locales en SoMachine	43
Ilustración 45. Condición de accionamiento del modo manual	43
Ilustración 46. Graficet de emergencia.....	44
Ilustración 47. Configuración de la IP de la pantalla HMI.....	45
Ilustración 48. Ventana de control del proceso de estampación	46

Ilustración 49. Barra de herramientas de HCTDesigner 2	46
Ilustración 50. Asignación de dirección de memoria a un interruptor	47
Ilustración 51. Ventana de control del proceso de fresado y taladrado.....	48
Ilustración 52. Ventana de control manual del proceso de fresado y taladrado	48
Ilustración 53. Como activar la lógica negativa en HTCDesigner 2.....	49
Ilustración 54. Selección de registro de alarmas y eventos	50
Ilustración 55. Configuración de la tabla de registro de alarmas y eventos	51
Ilustración 56. Ventana de configuración de una alarma	52
Ilustración 57. Registro de alarmas	52
Ilustración 58. Creación de niveles de usuario.....	53
Ilustración 59. Cargar aplicación en HMI.....	54
Ilustración 60. Selección de pantalla HMI a través de su dirección IP	54
Ilustración 61. Esquema de la sincronización de ambos procesos	55
Ilustración 62. Crear nueva macro	56
Ilustración 63. Creación de una variable dentro de una macro	58
Ilustración 64. Código de la macro	59
Ilustración 65. Contador del programa de la estampadora.....	60
Ilustración 66. Contador del proceso de fresado y taladrado	60

Índice de tablas

Tabla 1. Organización de las variables bool, byte y word en la memoria del PLC.....	14
Tabla 2. Características del TM241CE40R (fuente: Guía de programación Modicon M241 2014) 15	15
Tabla 3. Resumen características C1214C AC/DC/Rly	16
Tabla 4. Diferencias entre las versiones del modelo TP2507 (fuente: Catálogo de HCFA)	17
Tabla 5. Entradas y salidas del proceso de estampado	22
Tabla 6. Variables del prototipo de fresado y taladrado.....	35
Tabla 7. Direccionamiento de variables para M241 en HCTDesigner 2	50

1. Introducción

La automatización consiste en la agrupación de elementos y procesos informáticos, mecánicos y eléctricos que prácticamente sin la intervención del ser humano mejoran un proceso industrial, aunque esta también puede aplicarse a otros ámbitos como la agricultura, el control inteligente de casas (domótica), control de tráfico en las ciudades, etc.

Puede entenderse como la mejora de la mecanización; la realización de tareas usando herramientas, pero con intervención humana. Sin embargo, mediante la automatización los seres humanos son sustituidos por autómatas programables.

A lo largo de la historia el ser humano siempre ha creado herramientas que le facilitan la realización de tareas, desde la invención de la rueda en la Prehistoria hasta la creación de la máquina de vapor en la Revolución Industrial. Fue precisamente en esta época durante la que debido a la introducción de los primeros motores se creó la necesidad de inventar sistemas de control tales como reguladores de temperatura y/o presión.

En el siglo XX la automatización ganó protagonismo en la industria, siendo lo que se ha conocido más tarde como la Segunda Revolución Industrial. La automatización que había existido a pequeña escala comenzó a aplicarse cada vez más gracias a los avances en la computación digital.

Actualmente, los avances en informática y robótica han potenciado el alcance de la automatización provocando que esté presente en la mayoría de los sectores industriales. Su éxito se debe, sobre todo, a la reducción de tiempos y costes de producción, además de aumentar el volumen de producción. Pese a tener un coste elevado en su instalación inicial, es una inversión rentable en el largo plazo.

Esto ha permitido que los productos sean progresivamente más asequibles haciendo posible que el público mayoritario tenga acceso a ellos. Sin contar que ha supuesto una mayor seguridad para los trabajadores en las fábricas.

Estamos cerca del siguiente gran paso en la automatización: la introducción de la inteligencia artificial. Hasta ahora los autómatas solo pueden realizar tareas programadas siendo todavía necesaria la intervención humana. En cambio, en el futuro éstos podrán aprender de sus errores y llegar a tener una planta de producción carente de presencia humana.

No obstante, puede suponer la desaparición de puestos de trabajo, principalmente aquellos que requieren de baja formación o cualificación. Por otro lado, también es una oportunidad para crear nuevos puestos de trabajo e incluso reducir la contaminación gracias a la mejor optimización de los procesos industriales.

2. Objetivos

El objetivo principal de este trabajo consiste en llevar a cabo el control y automatización de dos prototipos de laboratorio que simulan dos subprocesos coordinados; una estampadora y un proceso de fresado y taladrado, que se detallarán más adelante.

Para ello se cuenta con dos autómatas programables: el Modicon M241 de Schneider y el S7 1200 de Siemens. También se dispone de una tableta HMI TP2507-E de HCFA a la que se conectan ambos autómatas mediante una conexión a través de ethernet. Además, la tableta HMI servirá para llevar a cabo la coordinación entre ambos autómatas.

Para la consecución de este objetivo, se definen los siguientes subobjetivos:

1. Familiarización con los entornos de programación SoMachine de Schneider, TIA Portal de Siemens y HCTDesigner 2 para el control de la pantalla HMI.
2. Asignación de autómatas para cada uno de los prototipos.
 - a. El autómatas M241 se designa para el proceso de fresado y taladrado.
 - b. El autómatas S7 1200 se designa para el proceso de estampación.
3. Programación de ambos procesos y diseño de una aplicación SCADA en la pantalla HMI para la supervisión de ambos procesos.

Se usará la misma aplicación en ambos autómatas para mayor comodidad a la hora de realizar el control de los procesos.
4. Coordinación de ambos procesos usando la pantalla HMI para la comunicación entre ambos autómatas.
 - a. Asignación de variables de comunicación en los programas de los autómatas.
 - b. Creación de macro en la pantalla para realizar el intercambio de información.



Ilustración 1. Montaje final

3. Conceptos previos

3.1 Autómatas

Un PLC o autómata programable es una computadora digital utilizada en la industria, que recibe datos de diversos dispositivos como pueden ser sensores o botones que generan señales de entrada al PLC. Con estos datos y siguiendo las instrucciones de un programa que ha sido previamente cargado en su memoria, calcula y envía unas determinadas señales de salida que sirven para poner en funcionamiento los distintos actuadores a los que está conectado y que conforman el proceso industrial; como pueden ser motores, pistones o válvulas y así controlar este proceso industrial de forma automática.

Un PLC solo puede funcionar y controlar un proceso si previamente se ha introducido en su memoria un programa. Para poder hacer esto se debe utilizar un software específico para cada una de las marcas fabricantes de autómatas programables.

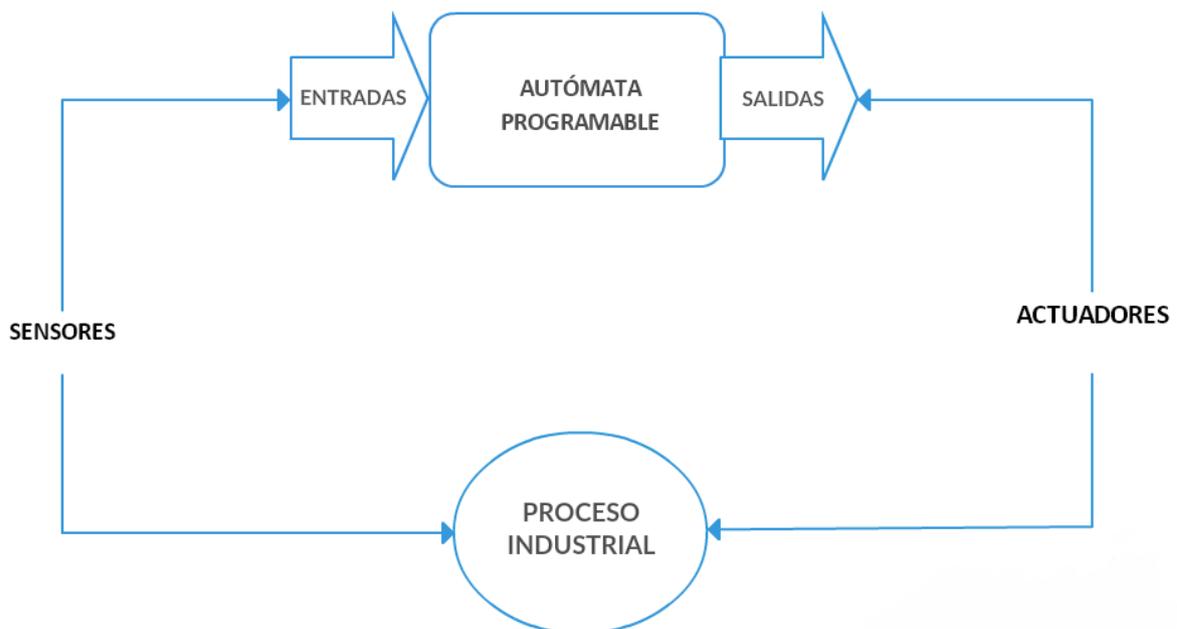


Ilustración 2. Representación simplificada de un proceso automatizado

Un PLC está compuesto por varios componentes y dependiendo de cómo estén dispuestos, podemos hablar de autómatas compactos o autómatas modulares.

- Los compactos llevan integrados todos sus componentes en una misma carcasa o bastidor, aunque pueden tener un compartimento para poder ampliar su número de entradas y salidas. Estos PLC son los más baratos y pequeños del mercado.

- Los modulares pueden ser configurados como uno quiera, sus componentes están separados y se puede armar un PLC distintito dependiendo de las necesidades del proceso industrial. Estos autómatas son los más caros y también ocupan más espacio, pero pueden tener mayor número de entradas y salidas, además de mayor capacidad de procesamiento.

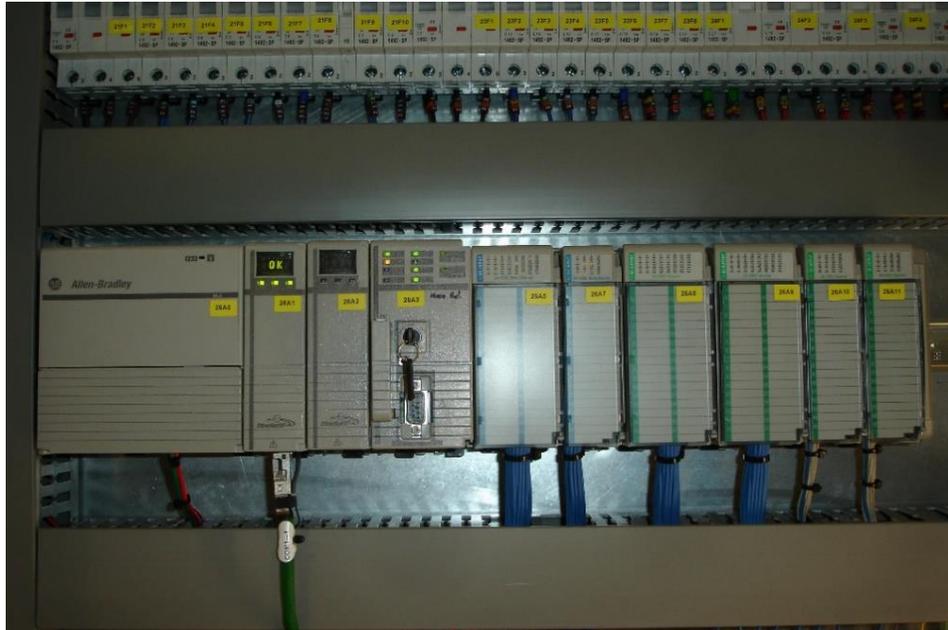


Ilustración 3 . Ejemplo de autómata modular de la marca Allen-Bradley

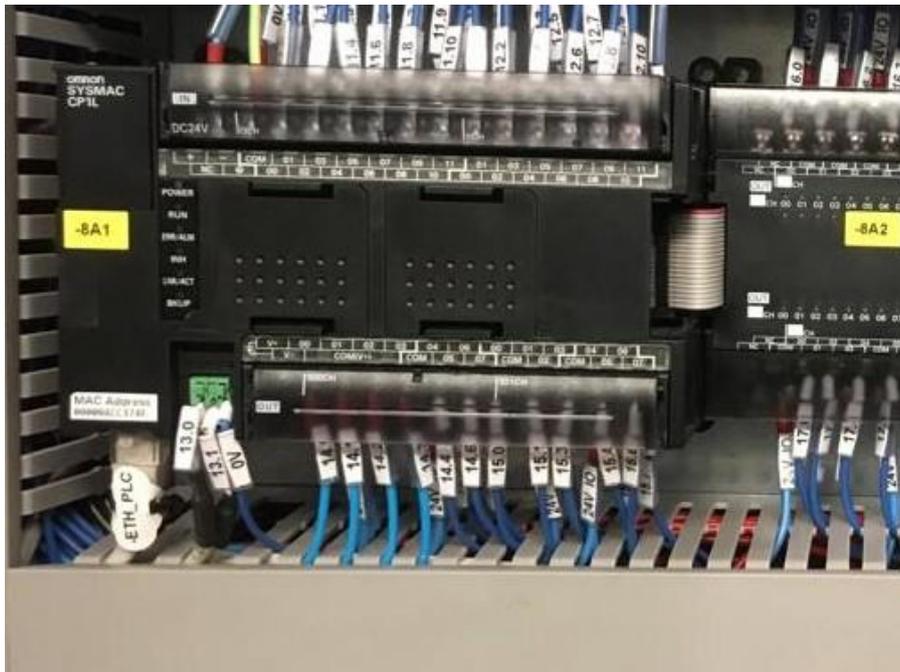


Ilustración 4. Ejemplo de autómata compacto de la marca Omron

A continuación, se describirán cada una de las partes que componen un PLC:

- Fuente de alimentación: es la encargada de proporcionar la energía eléctrica a todo el sistema para que éste pueda funcionar.
- Módulo de memoria: esta parte del autómatas es la que se ocupa de almacenar la información que éste recibe del exterior, ya sea de los módulos de entrada y salida o desde un PC a la hora de introducir el programa en él; o sea, las instrucciones que debe realizar el PLC. La capacidad de memoria del PLC suele medirse en bytes ya que no suelen ser memorias muy grandes. Hay dos tipos de memoria:
 - La memoria de datos: es el sitio donde se guardan los datos necesarios para la ejecución del programa, así como aquellos necesarios para su ejecución. Aquí es donde se guardan los datos recibidos y enviados al módulo de entradas y salidas.
 - La memoria de usuario: es la encargada de guardar el programa que ha diseñado el usuario. También se puede llamar memoria de programa. Esta memoria puede ser de varios tipos, aunque normalmente se suelen utilizar las de tipo RAM o EEPROM; la diferencia entre estas dos está en que la RAM necesitaría la incorporación de una pila al PLC porque cuando no recibe corriente eléctrica sus datos se borran, mientras que esto en la EEPROM no pasa.

Aparte de estas memorias, algunos autómatas tienen la posibilidad de ampliar su capacidad de almacenamiento mediante el uso de tarjetas SD.

- CPU: este dispositivo es el que se ocupa de la ejecución del programa llevando a cabo cada una de las instrucciones diseñadas. También se encarga de controlar la comunicación entre las distintas partes del PLC para que todo funcione correctamente.
- Módulo de entradas y salidas: tiene dos partes, el módulo de entradas y el módulo de salidas.
 - Módulo de entradas: es donde la información entra del exterior al PLC. Las entradas pueden ser digitales o analógicas; las digitales solo tienen dos estados posibles:
 - Encendido (1)
 - Apagado (0)

Cuando llega tensión al conector del módulo de entrada, esto se interpreta como un 1 y cuando no llega tensión se interpreta como un 0. Aunque en la práctica se suelen establecer unos rasgos de tensión para cada estado.

Las analógicas recogen datos que pueden variar de forma continua, como una presión o una temperatura. A estas entradas llega una determinada tensión o corriente proporcional al valor que está midiendo el sensor; y el autómatas se encarga de interpretar este valor como una magnitud. Por ejemplo, si es una temperatura pasaría esta tensión a grados centígrados.

- Módulo de salidas: en este módulo pasa lo mismo que lo mencionado anteriormente con los de entradas, puede haber salidas digitales o analógicas. Las digitales están divididas en:

- Salidas de relé: son las más usuales y se pueden conectar a ellas tanto cargas de corriente alterna como de corriente continua. Su tiempo de accionamiento es algo lento y suele estar alrededor de los 10 milisegundos.
- Salidas de transistor: solo pueden operar con cargas de corriente continua de baja potencia. Por esto suelen usarse para activar relés que activen cargas con más potencia, sin embargo, tienen la ventaja de que sus tiempos de accionamiento son muy rápidos y tienen mayor vida útil que las salidas de relé.
- Salidas por triac: Un triac es un triodo para corriente alterna de la familia de los tiristores. Este tipo de salidas son parecidas a los transistores y están hechas de semiconductores, pero a diferencia de los transistores, estas pueden manejar cargas de corriente alterna.

Por último, las salidas analógicas son las que permiten que el valor numérico de una variable continua se convierta en una tensión o una corriente proporcional a este valor. Por lo tanto, estas salidas se utilizan para controlar magnitudes continuas como, por ejemplo, una temperatura.

- Puerto de comunicación: el último de los componentes y el que se encarga de realizar la comunicación entre el autómatas y otros dispositivos periféricos a él, como pueden ser otros autómatas o tabletas HMI.

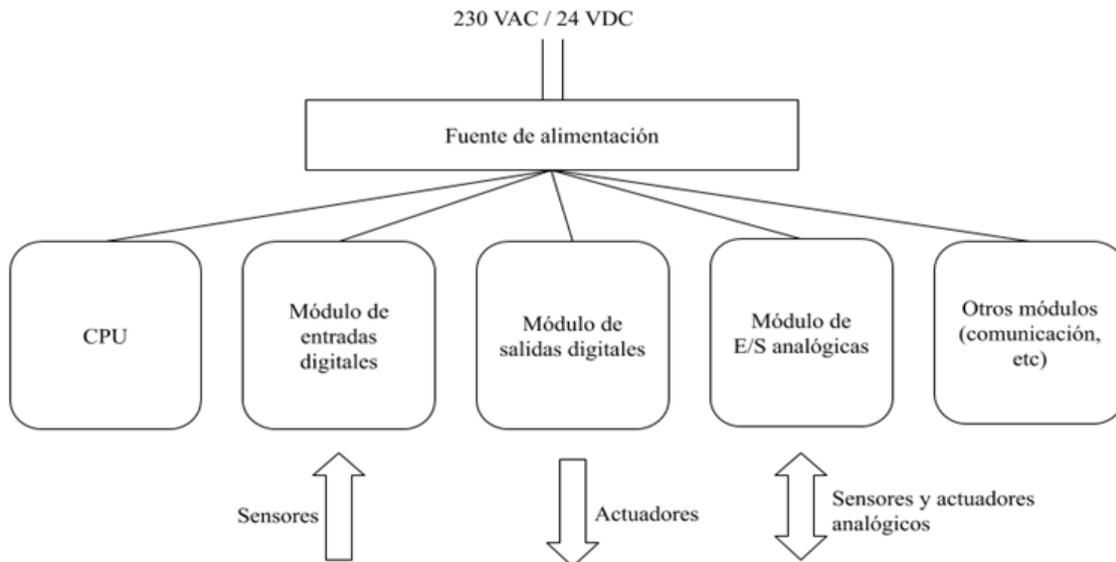


Ilustración 5. Representación esquematizada de un PLC

Conviene también hablar de las alternativas que hay actualmente al control mediante autómatas programables. Estas pueden ser las siguientes:

- **Lógica cableada:** La más utilizada hasta la aparición de los PLC. Consiste en realizar el control de un proceso mediante el uso de circuitos eléctricos cableados usando relés, relés temporizados, válvulas y otros elementos dependiendo del proceso a automatizar. Su principal desventaja es que para realizar cualquier cambio en su programación hay que cambiar el cableado y los componentes para añadir una nueva función.
- **Tarjeta de adquisición de datos (DAQ):** Otra alternativa es usar un PC al que se conecta una tarjeta de adquisición de datos. Su principal ventaja es la mayor capacidad de procesado y de memoria de un PC frente a un PLC, sin embargo, el precio del conjunto PC+DAQ es mucho más elevado que el precio de un PLC.
- **Microcontroladores:** Son circuitos integrados que pueden ser programados. Son capaces de llevar a cabo las instrucciones que han sido guardadas en su memoria. Están compuestos por una unidad central de procesamiento, una memoria y unidades para la entrada y salida de información. Sus ventajas con respecto a los PLC son que son mucho más pequeños que un autómata programable por lo que ocupa menos espacio y son más baratos que un PLC. Por otro lado, son menos resistentes que los autómatas y no tienen la capacidad de procesamiento de un PLC por lo que se deben utilizar en sistemas más sencillos.

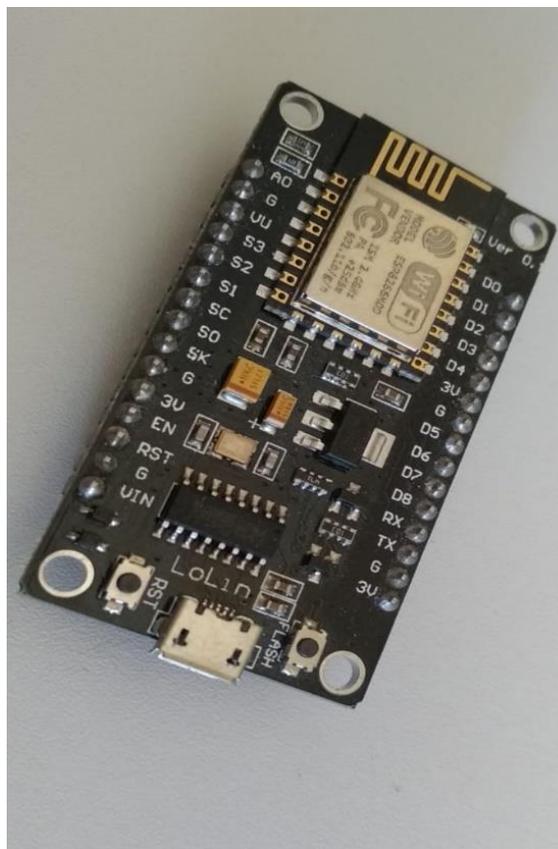


Ilustración 6. Ejemplo de microcontrolador (Espressif Esp8266)

3.2 Pantalla HMI

Una interfaz hombre-máquina (HMI por sus siglas en inglés) es un dispositivo utilizado como sistema de control de un proceso industrial. La HMI ayuda a los operarios que trabajan en el proceso a interactuar con las distintas máquinas que pueden componerlo.

Una pantalla HMI es básicamente un panel de control, normalmente táctil, que puede realizar diversas funciones como: representar gráficamente información recibida desde el proceso industrial (temperaturas, presiones, contadores, etc.), permitir manejarlo fácilmente transmitiendo órdenes a los autómatas que controlan el sistema o representar el estado en el que se encuentra el proceso en un determinado momento; por ejemplo, indicando los sensores que están activos o informando de fallos en las máquinas.

Los operarios tradicionalmente interactuaban con el proceso mediante botoneras y ahora se usan las HMI como complemento a estas porque determinadas funciones como la parada de emergencia se sigue realizando con un botón o seta de emergencia. Además, en procesos industriales muy sencillos no es rentable instalar una tableta HMI siendo preferible una botonera tradicional.

Es interesante remarcar la diferencia entre HMI y SCADA ya que a veces se suele pensar que son lo mismo.

Un sistema SCADA es un Sistema de Supervisión, Control y Adquisición de datos que se comunica con uno o varios elementos que forman parte del sistema de producción como pueden ser los autómatas, los sistemas de adquisición de datos o la telemetría. Es el sistema que controla la comunicación hombre-máquina monitorizando y conectando los dispositivos.

Las pantallas HMI son utilizadas para la monitorización de los sistemas SCADA, es decir, no están al mismo nivel, sino que uno forma parte del conjunto del otro: las HMI son un elemento de un sistema SCADA.



Ilustración 7. Pantalla HMI de la marca Siemens (fuente: Catálogo de Siemens)

3.3 Lenguajes de programación

Para programar un PLC se pueden utilizar varios tipos de lenguaje de programación dependiendo del software utilizado y aquí se describirán solo los que son de interés para este proyecto puesto que han sido los lenguajes utilizados:

- Diagrama de contactos o LD: También conocido como lenguaje ladder o diagrama de escalera. Es el lenguaje de programación de autómatas más usado porque es muy sencillo adaptarse a él y es muy fácil de usar y entender. Este tipo de lenguaje es el que más se parece a la lógica cableada, de ahí su extendido uso. Tiene dos ramas verticales, una a la izquierda que representa al positivo de la fuente y una a la derecha que representa al negativo. Entre estas dos ramas se insertan todos los elementos que componen el programa. Sus dos componentes esenciales son los contactos y las bobinas.
 - Contactos: Hay varios tipos de contactos; pueden ser normalmente abierto, normalmente cerrado, de flanco positivo o de flanco negativo. Cada uno se usa dependiendo de los requisitos del programa. Los contactos son las condiciones que se tienen que dar para activar una bobina y suelen estar asociados a las entradas del autómatas (sensores).
 - Bobinas: Al igual que pasa con los contactos, hay varios tipos de bobina; pueden ser bobinas normalmente abiertas, normalmente cerradas, un set (fijar un bit a 1) o un reset (fijar un bit a 0).

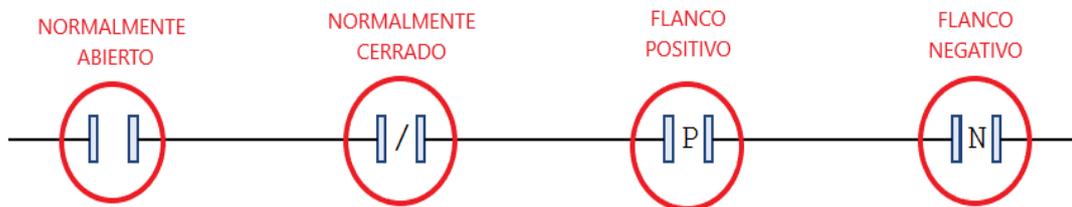


Ilustración 8. Tipos de contactos en un diagrama ladder



Ilustración 9. Tipos de bobinas en diagrama ladder

Tanto los contactos como las bobinas tienen que llevar direcciones de memoria asociados a ellos, por ejemplo, una bobina puede estar asociada a uno de los pines del módulo de salidas del PLC o puede estar asociada con una dirección de la memoria interna de este.

Este lenguaje tiene muchos más elementos disponibles, los más importantes y utilizados en este proyecto son los temporizadores y los contadores.

- Los temporizadores pueden ser de tres tipos: TON, cuando la entrada se pone a 1 se tarda x segundos en poner la salida a 1, TOF, cuando la entrada se pone a 0 tarda x segundos en poner la salida a 0 y por último los TP que son los temporizadores de pulso. Estos generan un pulso durante x segundos cuando la entrada pasa a 1. Los temporizadores que se usan en este proyecto son los TON.
- Los contadores, como los temporizadores, están divididos en tres tipos. El CTU es un contador creciente, cuando recibe un pulso en su entrada incrementa en uno el contador y cuando se alcanza una determinada cifra activa la salida. El CTD es decreciente. Funciona parecido al CTU, pero cuando recibe un pulso en su entrada en vez de sumarle uno a la cuenta lo que hace es restarle y cuando la cuenta llega a 0 su salida se activa. Por último, el CTUD que es una combinación de los anteriores. Dependiendo de si se activa su patilla de Up o Down suma o resta una unidad a la cuenta. Los contadores que se han utilizado en este proyecto son los CTU.

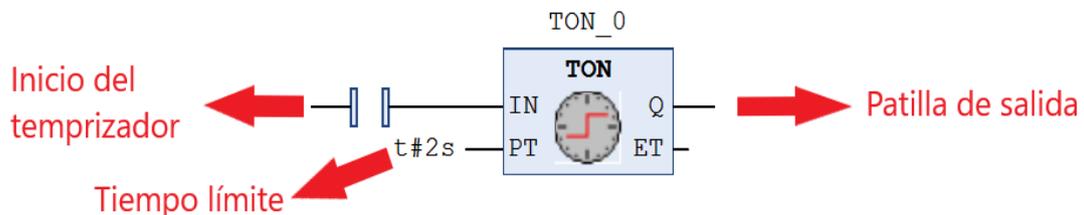


Ilustración 10. Conexiones de un temporizador TON

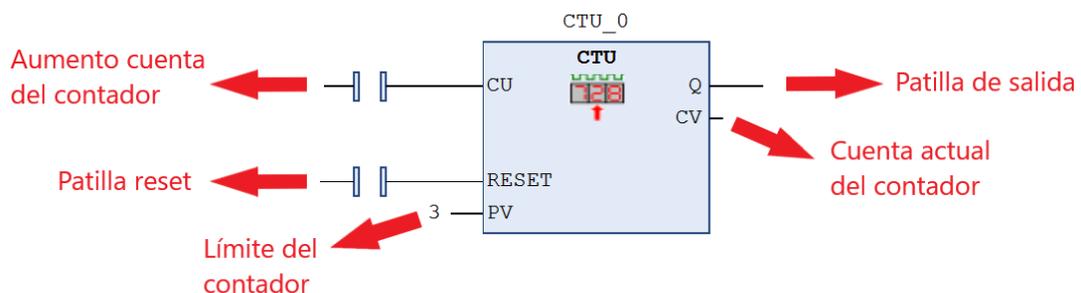


Ilustración 11. Conexiones de un contador CTU

- Grafcet o SFC: Es un método de programación de sistemas secuenciales que ofrece de forma sencilla, eficiente y estructurada la posibilidad de diseñar automatismos. En este lenguaje de programación se representa gráficamente el orden de las operaciones que debe realizar el automatismo. Su evolución se representa mediante unos estados o etapas que se activan al cumplirse unas determinadas condiciones. Los elementos que componen un Grafcet son los siguientes:
 - Etapa: Se usan para saber en qué estado se encuentra el automatismo y puede estar activa o desactivada. Las etapas normales se representan mediante un recuadro sencillo con un número en su interior que identifica dicha etapa. Todas las etapas se representan igual excepto las etapas iniciales que están representadas mediante un recuadro doble.
 - Acción asociada: Es la acción que se realizara mientras la etapa a la que está asociada este activa. Estas acciones suelen activar las salidas del autómatata que activan los distintos actuadores que tiene el proceso industrial, como pueden ser motores o válvulas. Las acciones pueden estar condicionadas, por ejemplo, que solo se activen durante un tiempo determinado o solo si una variable esta activada.
 - Transición: Es la condición o condiciones que hacen al automatismo avanzar a su siguiente etapa. Puede ser la activación de una de las entradas del PLC o condiciones asociadas a temporizadores o contadores. Se representan mediante una línea perpendicular a la línea de enlace entre dos etapas y se pone a su lado (derecha o izquierda) la condición necesaria para que se active la transición.
 - Líneas de enlace: Son las líneas que unen un estado con otro e indican el orden de ejecución de los estados.

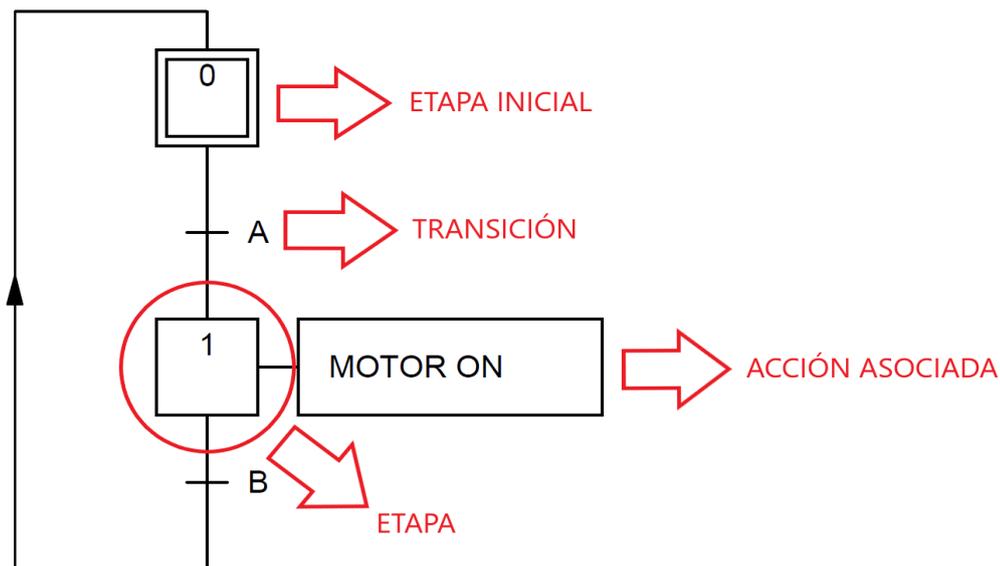


Ilustración 12. Componentes de un grafcet

Los Grafcet pueden tener sus elementos estructurados de varias formas.

- Puede estructurarse mediante una secuencia única (*Ilustración 12*), en este caso habrá solo una única rama donde las etapas estarán una seguida de la otra.
- También puede estructurarse como una bifurcación OR. En este caso se pueden poner varias ramas en paralelo, pero solamente se puede activar una de estas ramas. Además, habrá que asegurarse de que no se activen las condiciones de transición de varias ramas a la vez. El método más común para evitar la activación de dos ramas en OR a la vez es dar prioridad a una de las dos para que cuando se den las condiciones de ambas al mismo tiempo solo se pueda activar la rama con prioridad.
- Otra forma es disponerlos también en una bifurcación AND en la que varias ramas están activadas al mismo tiempo trabajando en paralelo.
- Por último, se pueden realizar bucles de repetición mientras una condición este activada haciendo que unos determinados estados se repitan y por tanto se repitan también sus acciones asociadas.

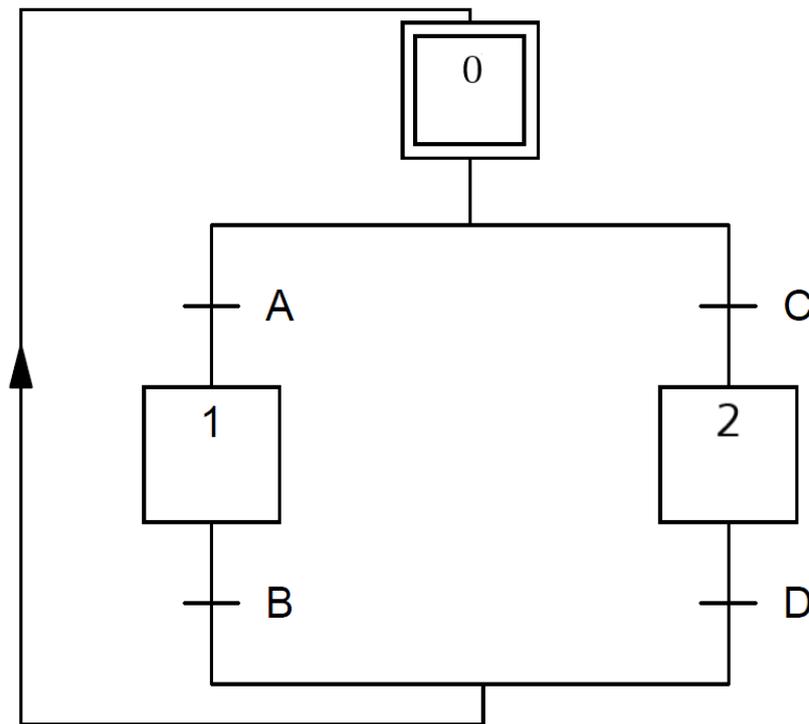


Ilustración 13. Grafcet con una bifurcación en OR

almacenar valores numéricos o para crear vectores. La memoria de un autómata quedaría organizada como aparece en la Tabla 1.

Direccionamiento bool	Direccionamiento byte	Direccionamiento word
%M0.0 a %M0.7	%MB0	%MW0
%M1.0 a %M1.7	%MB1	
%M2.0 a %M2.7	%MB2	%MW1
%M3.0 a %M3.7	%MB3	

Tabla 1. Organización de las variables bool, byte y word en la memoria del PLC

Como se ve los bits se nombran por el byte al que pertenecen y a posición que ocupan en él (un valor comprendido entre 0 y 7 para los 8 bits que componen un byte). En la tabla anterior solo se han mostrado las primeras posiciones de la memoria, se continuaría con el byte 4, el byte 5 y así sucesivamente.

4. Materiales

4.1 Modicon M241

Los Modicon M241 son una gama de controladores lógicos compactos de la marca Schneider. Hay varios tipos de autómatas programables en esta serie de controladores y se diferencian por el número de entradas y salidas de las que dispone el PLC, pudiendo elegir entre 24 o 40. Otra característica que los diferencia, es el tipo de salida que tiene instalado pudiendo elegir en este caso entre salidas de relé y transistor o solamente salidas de transistor. La diferencia entre ambos es que los de salida de transistor solamente pueden proporcionar una tensión continua mientras que los de relé y transistor puedan maniobrar con cargas tanto en alterna como en continua. Las características de los controladores pertenecientes a esta gama de productos pueden encontrarse en el documento “Modicon M241 - Guía de Programación (EIO000000143503M241)” proporcionado por Schneider.

El controlador lógico utilizado en este trabajo es el TM241CE40R, cuyas características se detallan en el documento “EcoStruxure™ Controlador lógico Modicon M241_TM241CE40R” proporcionado por Schneider a través de su página web. Como su nombre indica, este PLC dispone de 40 entradas y salidas que se distribuyen de la siguiente forma:

- 24 entradas digitales con una tensión de entrada de 24V en continua. El autómata interpretará una tensión superior o igual a 15V como un 1 lógico y una tensión menor o igual a 5V como un 0 lógico.
- 16 salidas digitales divididas en 12 salidas de relé y 4 salidas de transistor o salidas rápidas. La tensión de las salidas de relé es de 5V a 125V si se trata de corriente continua y de 5V a 250V si se trata de corriente alterna, los límites de tensión en estas salidas es 125V si es continua y 277V si es alterna. Con respecto a las salidas de transistor tienen una tensión de salida de 24V solo en corriente continua, pero pueden soportar hasta 30V.

El TM241CE40R debe alimentarse con una tensión alterna comprendida entre los 100V y 240V, aunque sus tensiones de alimentación límite son 85V como mínimo y 264V como máximo.

En cuanto a memoria, dispone de una memoria RAM de 64MB de los cuales 8MB están destinados a la ejecución del programa y una memoria flash de 128MB que sirve para almacenar el programa en caso de corte de electricidad. Además, los PLC de la serie M241 tienen una ranura para tarjetas SD para ampliar la memoria, proporcionándole una capacidad de almacenamiento extra de hasta 16GB.

Este controlador lógico puede operar en un rango de temperatura ambiente comprendido entre los -10°C y los 55°C si se encuentra instalado en posición horizontal y entre los -10°C y los 50°C si está instalado verticalmente.

Para la comunicación dispone de un puerto Ethernet, un puerto de programación vía USB y dos puertos para comunicación de línea serie.

El PLC del laboratorio tiene además instalados unos módulos de entradas y salidas analógicas, el TMC4AI2 de entradas analógicas y el TMC4AQ2 de salidas analógicas.

Referencia	Entradas digitales	Salidas digitales	Puertos de comunicación	Tipo de terminal	Fuente de alimentación
TM241CE40R	16 entradas normales ⁽¹⁾ 8 entradas rápidas (contadores) ⁽²⁾	12 salidas de relé de 2 A 4 salidas rápidas de común positivo (generadores de pulsos) ⁽³⁾	2 puertos de línea serie 1 puerto de programación USB 1 puerto Ethernet	Bloques de terminales de tornillo extraíbles	De 100 a 240 V CA
<p>(1) Las entradas normales tienen una frecuencia máxima de 5 kHz. (2) Las entradas rápidas se pueden utilizar como entradas normales o como entradas rápidas para funciones de conteo o de evento. (3) Las salidas de transistor rápidas pueden usarse como salidas de transistor normales o como salidas rápidas para el generador de pulsos (FG/PTO/ PWM) o para funciones de conteo (HSC).</p>					

Tabla 2. Características del TM241CE40R (fuente: Guía de programación Modicon M241 2014)



Ilustración 15. Autómata TM241CE40R instalado en el laboratorio de automatización del DISA

4.2 SIMATIC S7-1200

La serie Simatic S7-1200 es una gama de autómatas programables compactos de la marca Siemens. Estos PLC disponen de una alta capacidad de procesamiento con cálculo de 64 bits. Tienen integrados entradas analógicas. La serie S7-1200 dispone de 5 tipos de CPU diferentes:

- CPU 1211C
- CPU 1212C
- CPU 1214C
- CPU 1215C
- CPU 1217C

Todas, a excepción de la CPU 1211C, pueden ampliar sus entradas y salidas digitales y analógicas mediante módulos, además de poder añadirles módulos de comunicación.

La CPU utilizada en este proyecto es la CPU 1214C. Este PLC tiene tres versiones diferentes:

- DC/DC/DC: Se alimenta con 24V en continua y sus salidas solo proporcionan tensión continua.
- DC/DC/Rly: Se alimenta también con 24V en continua pero sus salidas son de tipo relé, por lo que soporta alterna y continua.
- AC/DC/Rly: Es la utilizada en este proyecto y se describirá más detalladamente a continuación.

La CPU 1214C AC/DC/Rly se alimenta con una tensión alterna comprendida entre los 120V y los 240V, pero soporta un mínimo de 85V y un máximo de 264V. Dispone de 14 entradas digitales con una tensión de entrada de 24V en continua (como las otras CPU). El PLC considera 0 lógico a una tensión menor o igual a 5V y 1 lógico a una superior o igual a 15V.

En cuanto a salidas digitales dispone de 10 salidas de relé que soportan una potencia de 30W en continua y 200W en alterna.

Tiene integradas 2 entradas analógicas con un rango de 0V a 10V y en cuanto a memoria tiene integrados una memoria de trabajo de 75KB y una memoria de carga de 4MB.

El rango de temperatura ambiente en el que puede trabajar es entre -20°C y 60°C.

Referencia	Alimentación	Entradas	Salidas
CPU C1214 AC/DC/Rly 6ES7-214-1BG40-0XB0	120-240VAC	14 digitales 2 analógicas	10 salidas relé

Tabla 3. Resumen características C1214C AC/DC/Rly



Ilustración 16. S7-1200 CPU 1214C instalado en el laboratorio de automatización del DISA

4.3 TP2507-E

La TP2507-E es una pantalla HMI perteneciente a la serie TP2000 de la marca HCFA, la cual tiene tres modelos de HMI diferentes: la TP2510, la TP2507 y la TP2504, cada modelo tiene versiones diferentes. El modelo utilizado (TP207) tiene tres versiones cuyas diferencias se muestran en la Tabla 4.

Parameter	TL2507	TP2507	TP2507-E
Color display	24-bit	24-bit	24-bit
Ethernet	None	None	Provided
SD CARD	None	Provided	Provided
USB HOST	Provided	Provided	Provided
USB DEVICE	Provided	Provided	Provided
COM1/COM3	Provided	Provided	Provided
COM2/COM4	None	Provided	Provided
Isolated power supply	Non-isolated power Surge protection	Isolated power Surge protection	Isolated power Surge protection

Tabla 4. Diferencias entre las versiones del modelo TP2507 (fuente: Catálogo de HCFA)

Esta HMI se alimenta con una tensión de 24V en continua, aunque admite un rango comprendido entre 18V y 28V DC, tiene un consumo inferior a 7W y su resolución es de 800x480.

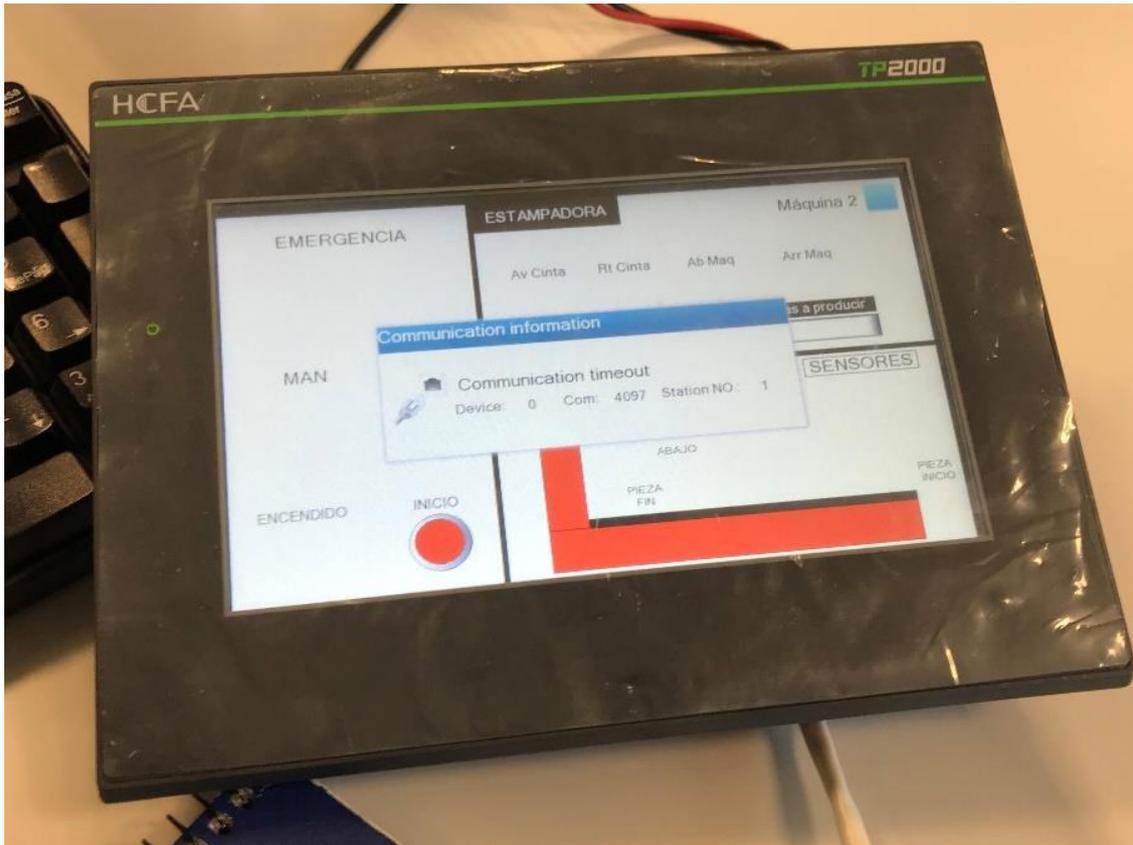


Ilustración 17. Pantalla HMI TP2507-E

4.4 SoMachine

SoMachine es el software distribuido por Schneider para la programación de sus controladores lógicos, pantallas HMI y otros de sus dispositivos. Proporciona un entorno para la configuración y programación de sus productos de forma sencilla. Funciona mediante la creación de lo que llaman POU, o unidad de organización de programa, que es donde se realiza el diseño del automatismo a instalar en el dispositivo. La versión utilizada ha sido la V4.1 aunque hay otras versiones disponibles como lo V4.3. Los lenguajes de programación disponibles en SoMachine son los siguientes:

- Diagrama de función continua (CFC)
- Grafcet (SFC)
- Editor de texto estructurado (ST)
- Diagrama de bloques funcionales (FBD)
- Diagrama de contactos (LD)
- Lista de instrucciones (IL)

Como se ha dicho en el apartado 3.3 Lenguajes de programación, los únicos lenguajes utilizados en este proyecto han sido el SFC y el LD.

4.5 TIA Portal

El TIA Portal es el software que se ha utilizado para la configuración y diseño del automatismo del PLC S7-1200 de Siemens. Este es el software que proporciona Siemens para la programación y configuración de sus dispositivos. Con este software se pueden programar los controladores S7-1200, S7-1500, S7-300 y S7-400. Tiene tres tipos de lenguaje de programación:

- Diagrama de contactos (KOP)
- FUP: Este lenguaje es parecido a utilizar puertas lógicas electrónicas (funciones AND, OR, etc)
- Texto estructurado

La versión de TIA Portal que se ha usado para este proyecto ha sido la V13.

4.6 HTCDesigner 2

El HTCDesigner 2 es el programa que se ha utilizado para configurar y diseñar la aplicación de la tableta HMI que controla ambos procesos. Este software lo distribuye HCFA de forma gratuita a través de su página web. Gracias a este programa es muy sencillo diseñar y programar la aplicación HMI mediante la creación de elementos gráficos que actúan como botones e indicadores de luz que se pueden vincular con la memoria de un PLC. El uso de este software se explicará más detalladamente más adelante.

4.7 Prototipo de estampadora

La maqueta utilizada en el proyecto simula un proceso de transporte y estampado de una pieza. La maqueta está fabricada por la marca Fischertechnik y se alimenta con 24V DC.

Los componentes que forman parte del prototipo son los siguientes:

- 2 motores de 24V DC (Uno para mover la cinta y otro para mover la estampadora)
- 2 finales de carrera (Para indicar la posición de la maquina estampadora)
- 2 fototransistores
- 2 barreras de luz LED (Usadas para indicar el final y principio de la cinta transportadora. Se usan unto con los fototransistores).

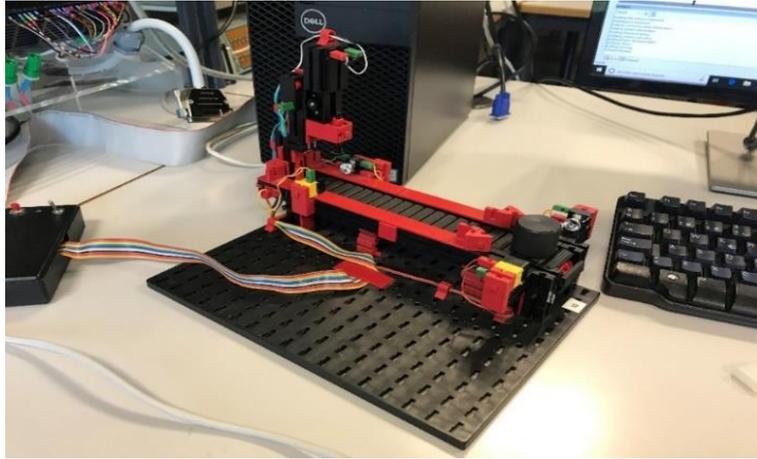


Ilustración 18. Prototipo de máquina estamadora

4.8 Prototipo de proceso de fresado y taladrado

Este prototipo simula un proceso en el que se lleva una pieza mediante unas cintas dispuestas en forma de U a unos puestos de fresado y taladrado. Esta maqueta está fabricada también por Fischertechnik y se alimenta con 24V en continua como la maqueta de la estamadora. Sus componentes son los siguientes:

- 8 motores de 24VDC (son los encargados de realizar el movimiento de las cintas, los empujadores, la fresadora y la taladradora)
- 4 finales de carrera (Se usan para detectar la posición inicial y final de los empujadores)
- 5 fototransistores
- 5 barreras de luz LED (Utilizados para conocer la posición de la pieza junto con los fototransistores)



Ilustración 19. Prototipo de proceso de fresado y taladrado

4.9 Fuente de alimentación

La fuente de alimentación utilizada es una Promax FAC-363B que estaba disponible en el laboratorio de automatización del DISA. Se ha utilizado para suministrarle los 24V en continua a la tableta HMI.

Este producto dispone de tres fuentes de alimentación separadas e independientes. Una suministra una tensión continua ajustable entre los 0V y los 30V. Esta es la que se ha utilizado en el proyecto. La segunda fuente de alimentación de la que dispone es una fuente doble fija de -15V, 0V y +15V. La última de las tres, también fija, suministra 5V.

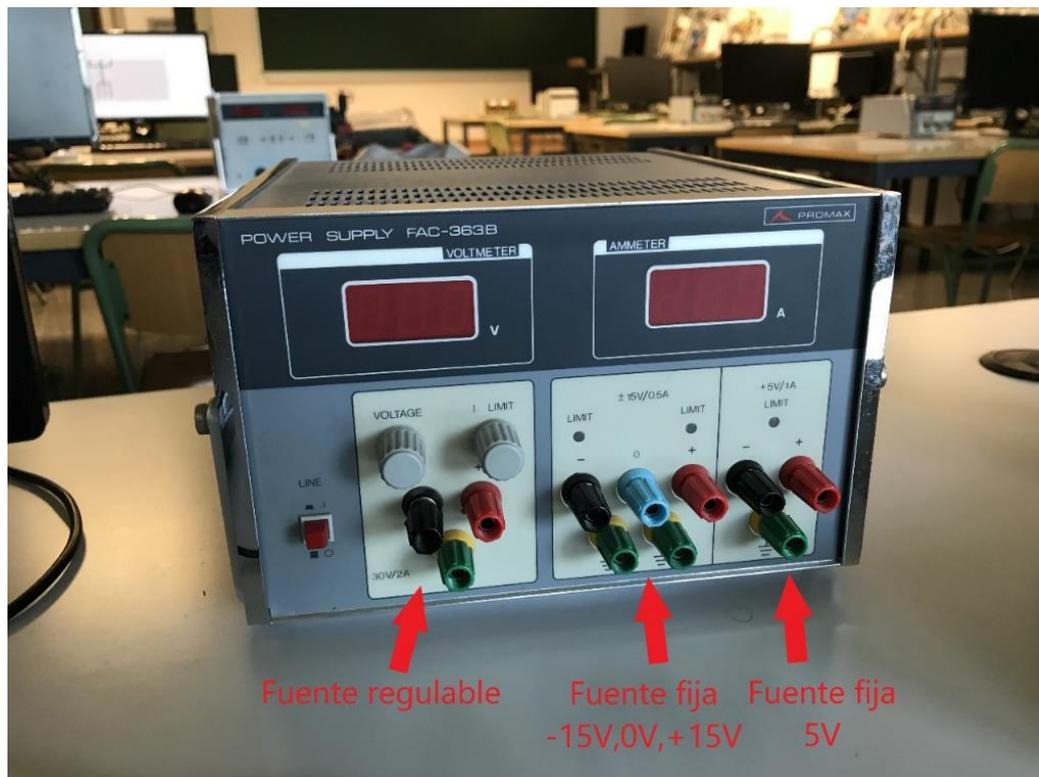


Ilustración 20. Fuente de alimentación Promax FAC-363B

5. Desarrollo del proyecto

5.1 Diseño proceso de la estampadora

El prototipo de la máquina estampadora estará controlado por el autómatas S7-1200 como se ha dicho en el apartado 2. Objetivos y su programa está diseñado con el software TIA Portal V13 utilizando lenguaje Ladder.

5.1.1 Funcionamiento del proceso

Primero, hay que mencionar todas las entradas y salidas de las que dispone este prototipo, así como las direcciones de memoria del controlador asociadas a estas entradas y salidas. También se indican las variables de las que dispone el programa para su funcionamiento. Estas se detallan en la

Tabla 5. Entradas y salidas del proceso de estampado

ENTRADAS	
Nombre de la variable	Dirección de memoria
<i>Encendido</i>	%M2.4
<i>Inicio</i>	%M2.5
<i>Manual</i>	%M4.5
<i>Emergencia</i>	%M2.6
<i>AvCinta</i>	%M4.6
<i>RtCinta</i>	%M4.7
<i>Ab Máquina</i>	%M5.0
<i>Arr Máquina</i>	%M5.1
<i>S1-INI</i>	%I0.1
<i>S2-FIN</i>	%I0.2
<i>ARRIBA</i>	%I0.3
<i>ABAJO</i>	%I0.4
SALIDAS	
<i>DIRCINTA</i>	%Q0.0
<i>CINTA-ON</i>	%Q0.1
<i>DIRMAQUINA</i>	%Q0.2
<i>MAQUINA-ON</i>	%Q0.3
<i>POWER</i>	%Q0.4
<i>Cuenta</i>	%MW6

Tabla 5. Entradas y salidas del proceso de estampado

En el autómatas, la nomenclatura %I se refiere a una dirección de memoria asociada a una entrada del PLC y %Q a una dirección asociada a una salida del PLC. Por otro lado, la nomenclatura %M se refiere a una posición de la memoria interna del autómatas que no lleva asociada ninguna entrada o salida. Las variables %M se podrán manipular a través de la HMI.

El proceso dispondrá de dos modos de funcionamiento, manual y automático, a parte de un modo de parada de emergencia.

Por lo tanto, el funcionamiento de este prototipo en modo automático es el siguiente:

1. Al activar la variable *Encendido* se activará la salida *POWER*. Esta salida hará que se enciendan todos los sensores del proceso por lo que deberá estar activada en todo momento salvo cuando se quiera apagar la máquina.

2. Cuando se coloque una pieza al principio de la cinta transportadora se activará la entrada *SI-INI*. Si la entrada manual esta desactivada (modo automático activado) al pulsar inicio empezara el proceso.
3. Tras haber pulsado inicio se esperará 3 segundos simulando que este es el tiempo que necesita la estampadora para calentarse. Tras estos 3 segundos la cinta avanzará activando las salidas *CINTA-ON* y *DIRCINTA* hasta que se active *S2-FIN* que indica que se ha llegado al final de la cinta transportadora.
4. La máquina estampadora bajará (*MAQUINA-ON* activado) hasta que se active *ABAJO* y una vez así esperará 4 segundos.
5. Pasados los 4 segundos la máquina estampadora subirá hacia arriba (*MAQUINA-ON* y *DIRMAQUINA* activados) hasta que se active *ARRIBA*.
6. Una vez la máquina ya este arriba se activará la cinta transportadora (*CINTA-ON*) hasta el principio de la cinta. La cinta parará cuando se active *SI-INI*. Se permanecerá en este estado hasta que se retire la pieza producida y dependiendo de si *Encendido* esta activado o no se apagará el prototipo o se esperará para fabricar una pieza nueva.

El modo manual sirve para accionar manualmente la cinta transportadora o la máquina estampadora. Se activará cuando la variable *Manual* esté activada. Con la entrada *AvCinta* la cinta transportadora se moverá hacia adelante, con *RtCinta* la cinta se moverá hacia atrás, con la variable *Ab Máquina* la máquina estampadora se moverá hacia abajo y finalmente, con la entrada *Arr Máquina* la maquina estampadora se moverá hacia arriba.

Por último, el modo de emergencia servirá para paralizar todo el proceso de producción. Lo que hará es evitar el movimiento de cualquiera de los actuadores de los que dispone el prototipo, bloqueándolos tanto si se encuentra en modo automático como si se encuentra en modo manual.

5.1.2 Iniciar nuevo proyecto en TIA Portal

Como se ha dicho anteriormente, el programa de este prototipo ha sido diseñado con el software TIA Portal de Siemens en su versión V13. En este apartado se describirá como iniciar un nuevo proyecto en este programa y como se realiza la configuración inicial para empezar a diseñar un automatismo.

Lo primero que hay que hacer al iniciar el TIA Portal es seleccionar *Crear nuevo proyecto* como se muestra en la Ilustración 21. Pantalla de inicio en TIA Portal V13

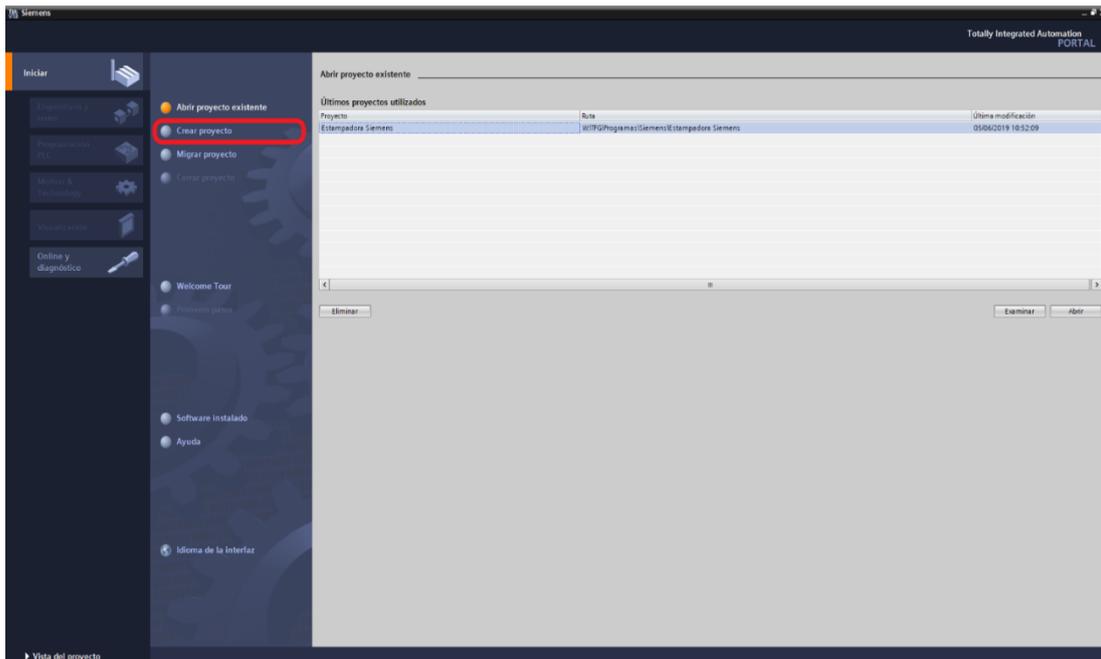


Ilustración 21. Pantalla de inicio en TIA Portal V13

Una vez que se ha seleccionado *Crear proyecto* aparecerá una ventana en la que se debe introducir el nombre que se le quiere dar al nuevo proyecto y se pulsa en *Crear*. En este caso al proyecto se le ha llamado Estampadora Siemens.

Una vez se ha creado el nuevo proyecto se abrirá la ventana que se representa en la Ilustración 22. En esta ventana hay dos elementos principales que son los que deberán seleccionarse: La opción *Configurar un dispositivo* (marcada con un 1 en la imagen) y la opción *Abrir la vista de proyecto* (marcada con un 2).

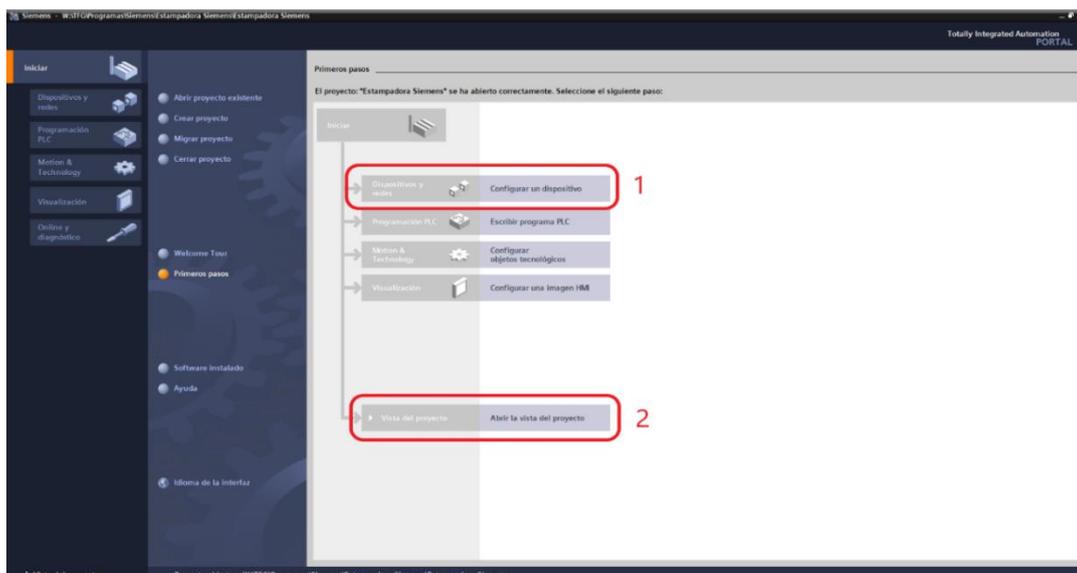


Ilustración 22. Ventana de vista de proyecto en TIA Portal

Cuando se ha creado un nuevo proyecto primero hay que seleccionar *Configurar un dispositivo* para incluir el dispositivo con en que estamos trabajando, en este caso la CPU 1214C AC/DC/Rly. Cuando se seleccione esa opción se abrirá la ventana que está representada en la Ilustración 23. Selección de CPU en TIA Portal en la que se deberá clicar en el apartado *Agregar dispositivo* y una vez seleccionado este menú se deberá ir a *Controladores>SIMATIC S7-1200>CPU 1214C AC/DC/Rly>6ES7-214-1BG40-0XB0*.

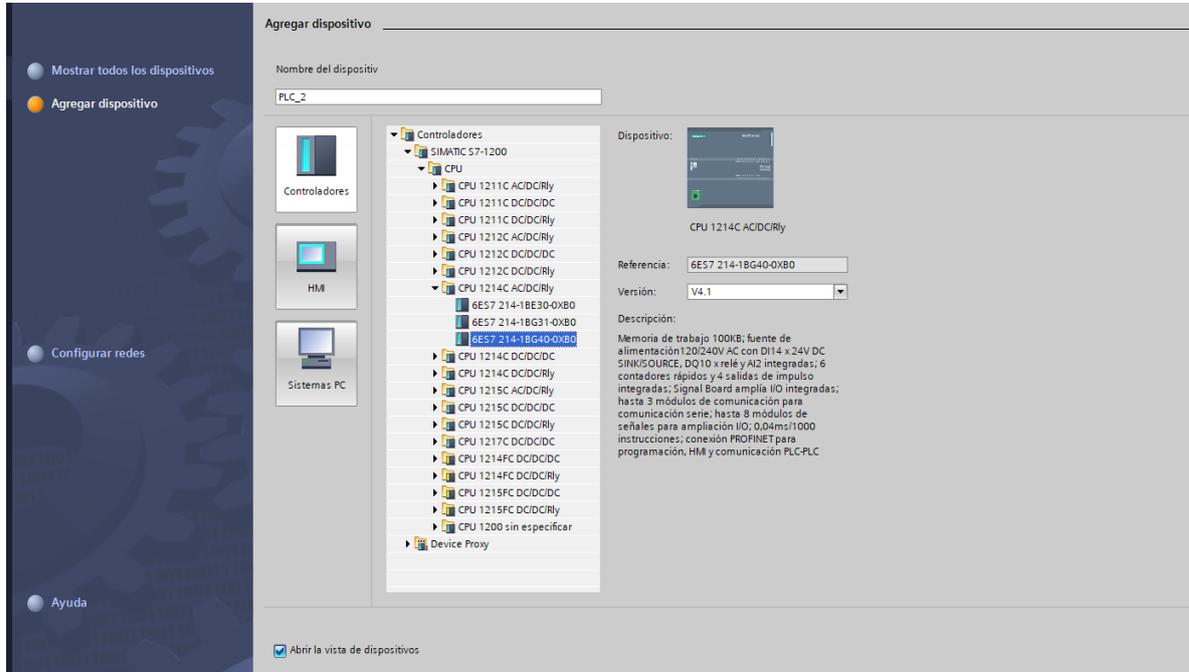


Ilustración 23. Selección de CPU en TIA Portal

Tras haber hecho esto volvemos a la ventana de la Ilustración 22 y ahora se selecciona *Abrir la vista de proyecto*. Así se entrará en la vista de proyecto que sirve para configurar el autómata y para escribir el código que se quiere introducir en este (Ilustración 24. Vista de proyecto). Lo primero que habría que hacer es introducir la dirección IP de nuestro controlador lógico para que más tarde no de fallos a la hora de intentar introducir el programa en él. Para ello debemos ir al menú *Dispositivos y redes* y se abrirá la ventana que se muestra en la Ilustración 25. Ventana de dispositivos y redes. Se verá una representación del autómata, se debe clicar en la representación de la clavija para el cable RJ-45 de ethernet (marcado con un círculo rojo en la Ilustración 25) y así se accederá directamente a la parte de configuración de las direcciones de ethernet. Aquí se deberá introducir la dirección IP de nuestro autómata y habrá que seleccionar la opción *Utilizar router* en la que se deberá poner la misma dirección IP del autómata, pero cambiando el último número por un 1 (como se ve también en la Ilustración 25). Una vez hecho esto ya no habrá problemas a la hora de conectar el PC con el PLC.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

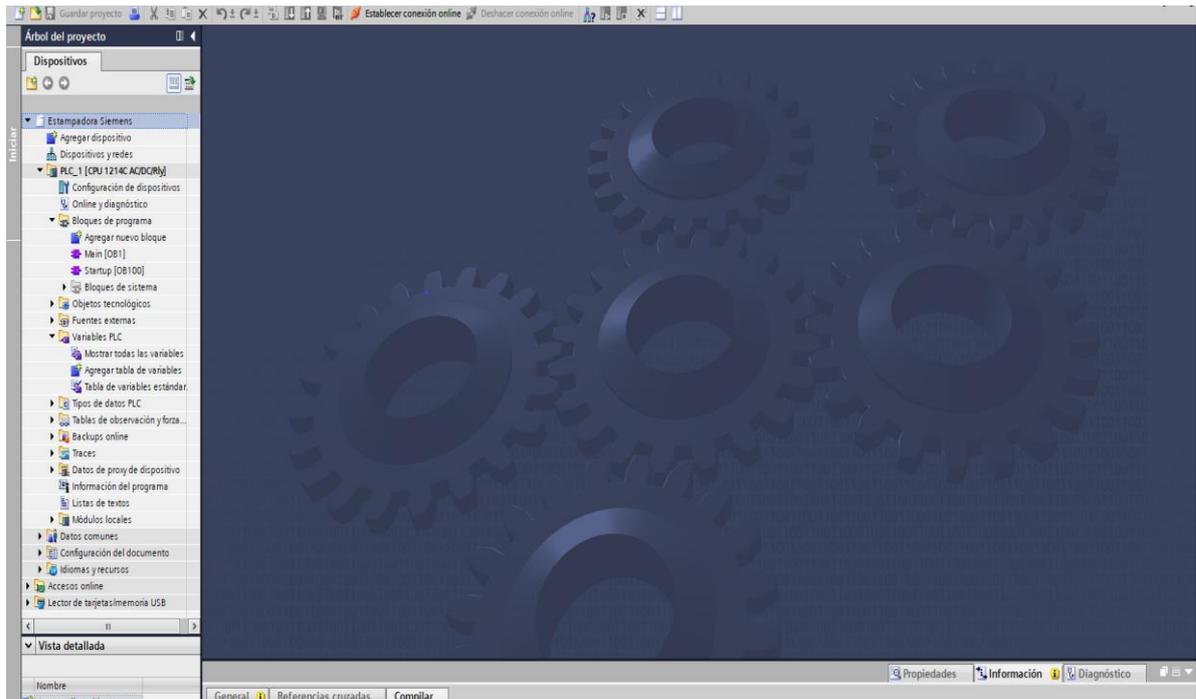


Ilustración 24. Vista de proyecto

Tras la configuración inicial del proyecto (selección de dispositivo e introducción de la dirección IP) ya se puede empezar a programar. En TIA Portal el código debe introducirse en los llamados bloques de programa. Hay varios tipos de estos, por ejemplo, un bloque que solo se pone en marcha una vez al iniciar el PLC (llamado OB100). Para la programación del prototipo de la estampadora solo se han utilizado el bloque Main y el bloque Starup (OB100).

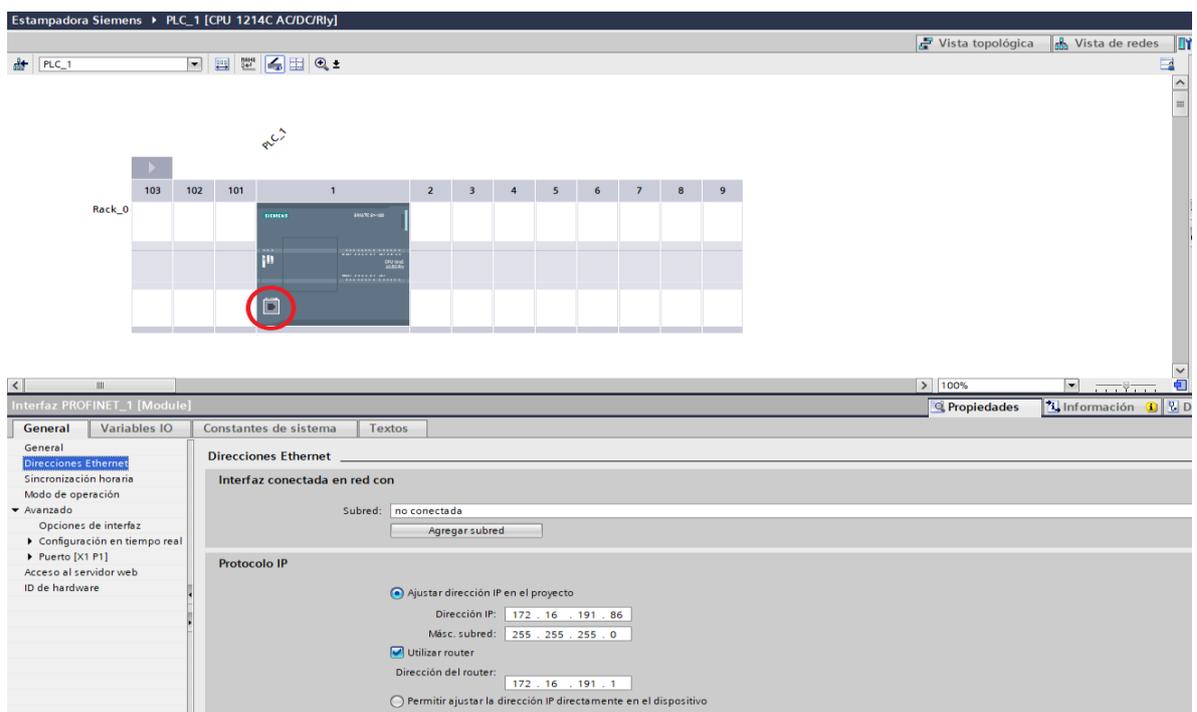


Ilustración 25. Ventana de dispositivos y redes

5.1.3 Programa de la estampadora

Como se ha dicho anteriormente, para diseñar este programa se ha realizado primero el diseño a mano usando el lenguaje de Grafcet ya que es más sencillo de programar y se ha pasado después a diagrama de contactos usando las funciones lógicas de los Grafcet.

Las funciones lógicas son una manera de representar un programa de un automatismo mediante una serie de ecuaciones. Hay tres tipos de ecuaciones:

- Ecuaciones de activación: Son las ecuaciones con las que se representa la transición entre una etapa y otra del Grafcet. Son las condiciones que se tienen que dar para que se active la etapa a la que está asociada. Su expresión es $S_n = X_{n-1} * T_{n-1}$ donde X es la ecuación de estado de la etapa anterior y T las condiciones de activación.
- Ecuaciones de estado: Son las ecuaciones que indican cuando una etapa esta activada y cuando no. Su forma es $X_n = S_n + \overline{S_{n+1}} * X_n$ donde X_n es la etapa que se quiere activar, S_n su ecuación de activación y S_{n+1} la ecuación de activación de la etapa siguiente.
- Ecuaciones de salida: Estas sirven para indicar cuando se tiene que activar las acciones que están asociadas a una etapa del Grafcet.

Para entender mejor como funciona cada una se pondrá un ejemplo sencillo a continuación, el Grafcet representado en la Ilustración 26.

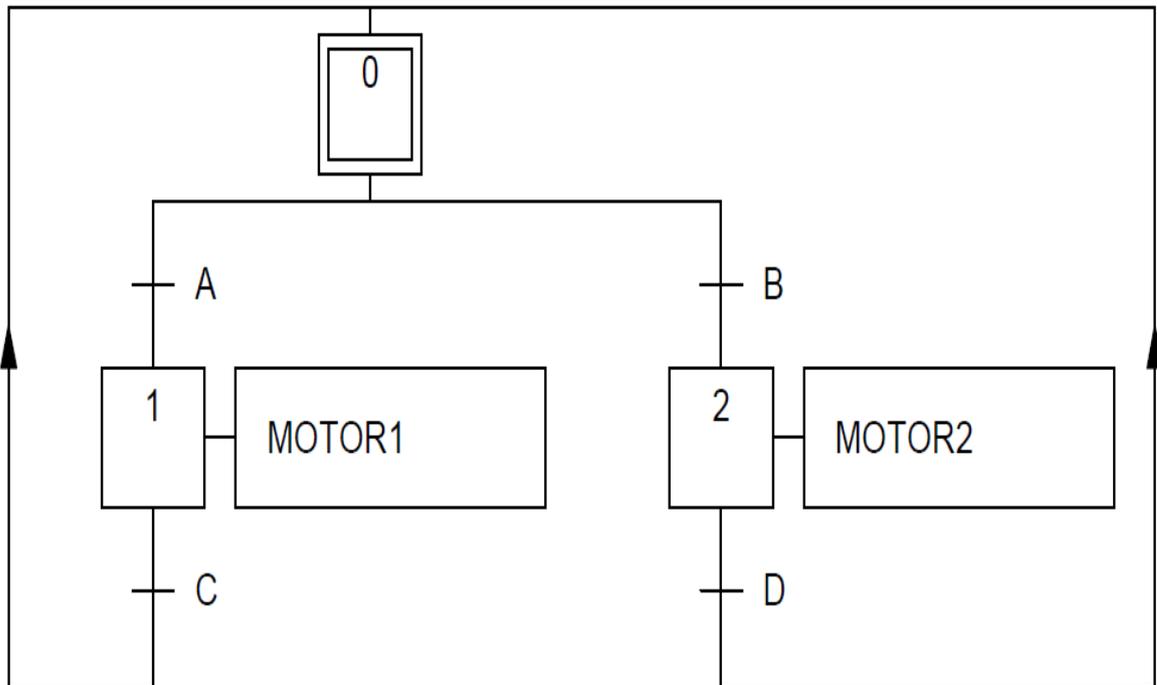


Ilustración 26. Ejemplo de Grafcet para funciones lógicas

Primero se escribirán las ecuaciones de activación del Grafcet representado en la Ilustración 26.

$$S_0 = X_1 * C + X_2 * D + C_i$$

$$S_1 = X_0 * A$$

$$S_2 = X_0 * B$$

C_i es el bit de inicio que se pone a 1 solo cuando se arranca el autómatas para que pueda iniciarse el programa. En TIA Portal se realiza mediante un bloque de programa Startup (OB100).

Segundo, las ecuaciones de estado de este Grafcet serían:

$$X_0 = S_0 + \bar{S}_1 * \bar{S}_2 * X_0$$

$$X_1 = S_1 + \bar{S}_0 * X_1$$

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

Las etapas (representadas con X) solo estarán activas cuando se dé su ecuación de activación y no se active la ecuación de activación de las etapas que tiene a continuación esta.

Por último, las ecuaciones de salida serían:

$$MOTOR1 = X_1$$

$$MOTOR2 = X_2$$

La salida MOTOR1 solo estará activada mientras se esté dando X_1 y lo mismo pasa con MOTOR2 y X_2 . Si la salida MOTOR1 se activara con más etapas solo habría que sumar la ecuación de estado correspondiente a dicha etapa.

Con las funciones lógicas es muy sencillo pasar de Grafcet a diagrama de contactos. Lo que habría que hacer en TIA Portal es declarar variables internas (en la zona de memoria %M) que representen las ecuaciones de estado y las de activación, esto es las X y las S. Después se irían poniendo todas las ecuaciones de activación y las de estado de todas las etapas que tenga el Grafcet teniendo en cuenta que las multiplicaciones son poner contactos en serie y las sumas ponerlos en paralelo.

Volviendo al programa de la estampadora, el Grafcet de su modo manual y automático sería el representado en la Ilustración 27, que se ajusta al funcionamiento descrito en el apartado 5.1.1 Funcionamiento del proceso.

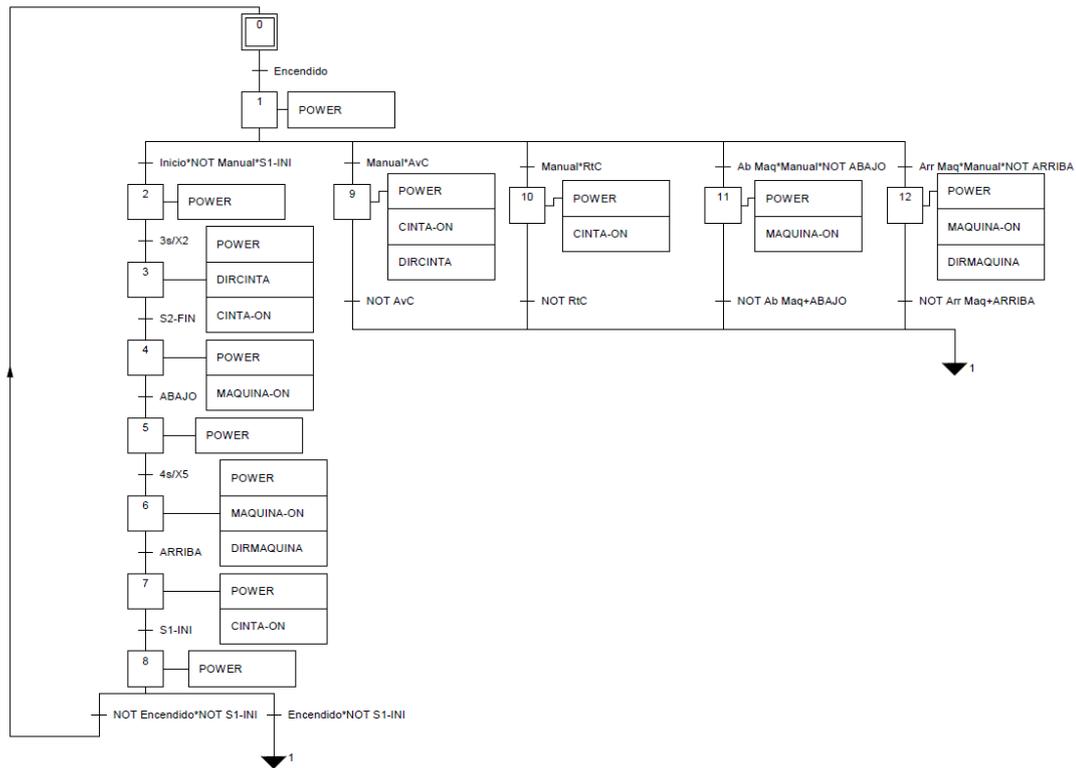


Ilustración 27. Grafcet prototipo estampadora

También hay un Grafcet de emergencia que es el que se usa para entrar en el modo de parada de emergencia que está representado en la Ilustración 28. En este programa, para salir del modo de emergencia se debe desactivar la entrada Emergencia y debe activarse el modo manual. Además, para asegurarse de que la máquina estampadora va a estar posicionada arriba el Grafcet de emergencia tiene un estado que sube automáticamente esta máquina hasta que está arriba y entonces se sale completamente del modo de emergencia. También se ha incorporado un contador de piezas producidas que cuenta cuando se activa X8.

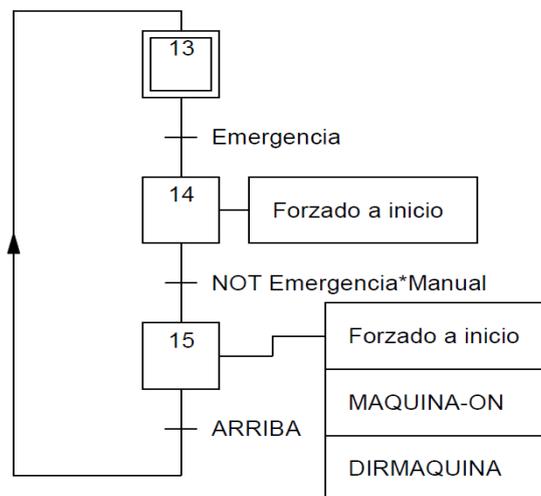


Ilustración 28. Grafcet de emergencia de la estampadora

A continuación, están escritas todas las funciones lógicas diseñadas para este proceso que se corresponden con los Grafcet mostrados en Ilustración 27 y en Ilustración 28, pudiendo verse el programa completo en lenguaje Ladder en el apartado 1. Programa TIA Portal de los anexos.

- Ecuaciones de activación

$$S_0 = X_8 * \overline{Encendido} * \overline{S1 - INI} + C_i$$

$$S_1 = X_0 * Encendido + X_8 * Encendido * \overline{S1 - INI} + X_9 * \overline{AvC} + X_{10} * \overline{RtC} \\ + X_{11} * (\overline{Ab Maq} + ABAJO) + X_{11} * (\overline{Arr Maq} + ARRIBA)$$

$$S_2 = X_1 * S1 - INI * Inicio$$

$$S_3 = X_2 * 3s/X_2$$

$$S_4 = X_3 * S2 - FIN$$

$$S_5 = X_4 * ABAJO$$

$$S_6 = X_5 * 4s/X_5$$

$$S_7 = X_6 * ARRIBA$$

$$S_8 = X_7 * S1 - INI$$

$$S_9 = X_1 * Manual * AvC$$

$$S_{10} = X_1 * Manual * RtC$$

$$S_{11} = X_1 * Manual * Ab Maq * \overline{ABAJO}$$

$$S_{12} = X_1 * Manual * Arr Maq * \overline{ARRIBA}$$

$$S_{13} = X_{15} * ARRIBA$$

$$S_{14} = X_{13} * Emergencia$$

$$S_{15} = X_{14} * \overline{Emergencia} * Manual$$

- Ecuaciones de estado

$$X_0 = S_0 + \overline{S_1} * X_0 + X_{14} + X_{15}$$

$$X_1 = (S_1 + \overline{S_2} * \overline{S_9} * \overline{S_{10}} * \overline{S_{11}} * \overline{S_{12}} * X_1) * \overline{X_{14}} * \overline{X_{15}}$$

$$X_2 = (S_2 + \bar{S}_3 * X_2) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_3 = (S_3 + \bar{S}_4 * X_3) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_4 = (S_4 + \bar{S}_5 * X_4) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_5 = (S_5 + \bar{S}_6 * X_5) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_6 = (S_6 + \bar{S}_7 * X_6) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_7 = (S_7 + \bar{S}_8 * X_7) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_8 = (S_8 + \bar{S}_0 * \bar{S}_1 * X_8) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_9 = (S_9 + \bar{S}_1 * X_9) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_{10} = (S_{10} + \bar{S}_1 * X_{10}) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_{11} = (S_{11} + \bar{S}_1 * X_{11}) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_{12} = (S_{12} + \bar{S}_1 * X_{12}) * \bar{X}_{14} * \bar{X}_{15}$$

$$X_{13} = S_{13} + \bar{S}_{14} * X_{13}$$

$$X_{14} = S_{14} + \bar{S}_{15} * X_{14}$$

$$X_{15} = S_{15} + \bar{S}_{13} * X_{15}$$

Como se puede ver en las ecuaciones de estado, las etapas X₁₃, X₁₄ y X₁₅ son las que condicionan todo el proceso al ser los estados del Grafcet de seguridad.

- Ecuaciones de salida

$$POWER = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} + X_{11} + X_{12}$$

$$CINTA - ON = X_3 + X_7 + X_9 + X_{10}$$

$$DIRCINTA = X_3 + X_9$$

$$MAQUINA - ON = X_4 + X_6 + X_{11} + X_{12}$$

$$DIRMAQUINA = X_6 + X_{12}$$

Todas estas variables (las X y las S) deberán ser declaradas junto con todas las demás variables del proceso. Para declarar variables en TIA Portal se debe ir a la opción *Mostrar*

todas las variables y aquí aparecerá una tabla con todas las variables del programa. Aquí pulsando en una celda se puede introducir una nueva variable, indicando su nombre, el tipo de variable (bool o word por ejemplo) y la dirección de memoria en la que estará la variable.

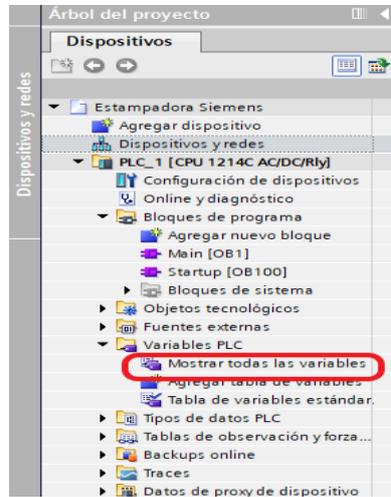


Ilustración 29. Variables del PLC en TIA Portal

5.1.4 Transferencia del programa al autómata

Para terminar, lo último que habría que hacer una vez se ha configurado el autómata y se diseñado el programa sería introducir este en la memoria del PLC. Para ello se debe ir al menú de carga al que se accede pulsando en el icono marcado con un 1 en la Ilustración 30. Al pulsar aparecerá la ventana de carga de programa mostrada también en la Ilustración 30.

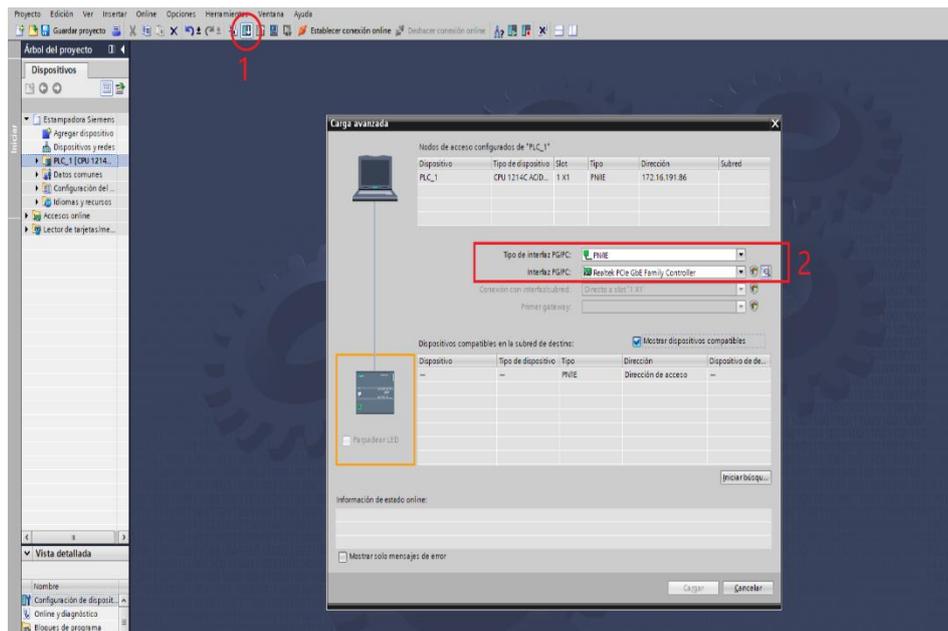


Ilustración 30. Menú de carga de programa en TIA Portal

Aquí se deberá elegir la opción PN/IE en el desplegable de *Tipo de interfaz PG/PC* y en el desplegable de abajo *Interfaz PG/PC* se debe elegir la opción Realtek PCIe GbE Family Controller.

Tras haber seleccionado todas estas opciones se debe pulsar en *Iniciar búsqueda* tras lo cual se buscará si hay algún dispositivo con la dirección IP que le hemos asignado al nuevo proyecto. Cuando lo encuentre el botón de cargar se podrá pulsar y se podrá cargar el programa en el autómata. Después se selecciona *Establecer conexión online* y se conectará al autómata y deberá seleccionarse la opción *Run*, situada a la derecha del botón anterior.

5.2 Diseño de proceso de fresado y taladrado

Para este prototipo se ha seleccionado el controlador lógico TM241CE40R de Schneider. Se ha utilizado el programa SoMachine en su versión V4.1 para el diseño del automatismo y la configuración del autómata.

5.2.1 Funcionamiento del proceso

Para empezar, la maqueta dispone de varios actuadores. Estos son los siguientes:

- 2 empujadores
- 4 cintas de transporte
- 2 motores para mover la fresadora y la taladradora

Estos han sido nombrados de una forma específica en el proyecto. Esta nomenclatura se puede ver en la Ilustración 31. Actuadores del prototipo de fresado y taladrado.

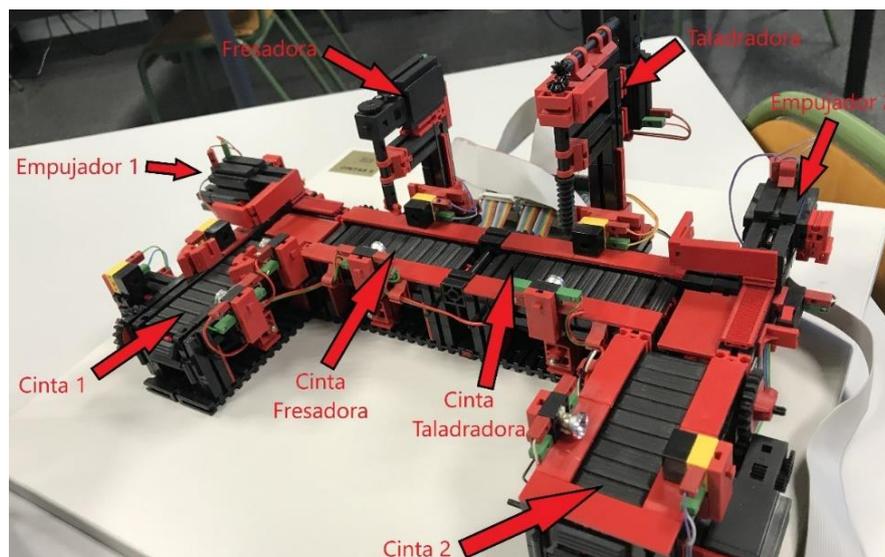


Ilustración 31. Actuadores del prototipo de fresado y taladrado

Las variables que han sido utilizadas para la programación del automatismo de este prototipo son las mostradas en la

Tabla 6. Variables del prototipo de fresado y taladrado, tanto las variables asociadas a entradas y salidas físicas (variables %IX y variables %QX) como las asociadas a la memoria interna del autómatas (variables %MX).

Hay que tener en cuenta que los sensores de posición de la pieza funcionan con lógica negada, es decir, cuando se desactive indicará que la pieza está en esa posición, por lo que cuando ese sensor está a 0 es cuando ha detectado la presencia de la pieza.

El proceso, al igual que el del prototipo de estampación, dispone de dos modos de funcionamiento, el manual y el automático, además de un modo de parada de emergencia y de un contador para llevar la cuenta de las piezas que han sido producidas en el proceso.

ENTRADAS		
Nombre de la variable	Dirección de memoria	Función
Encendido	%MX0.2	Encender los sensores
Manual	%MX0.1	Cambiar entre manual y automático
Emergencia	%MX0.0	Activar parada de emergencia
Inicio	%MX0.3	Iniciar proceso automático
AvC1	%MX0.4	Activar cinta 1 en manual
AvC2	%MX0.5	Activar cinta 2 en manual
AvCF	%MX0.6	Activar cinta fresadora en manual
AvCT	%MX0.7	Activar cinta taladradora en manual
Fon	%MX1.0	Activar fresadora en manual
TaOn	%MX1.1	Activar taladradora en manual
AvEm1	%MX1.2	Empujador 1 hacia adelante en manual
RtEm1	%MX1.3	Empujador 1 hacia atrás en manual
AvEm2	%MX1.4	Empujador 2 hacia adelante en manual
RtEm2	%MX1.5	Empujador 2 hacia atrás en manual
FCF1	%IX0.0	Empujador 1 delante
FCT1	%IX0.1	Empujador 1 atrás
FCF2	%IX0.2	Empujador 2 delante
FCT2	%IX0.3	Empujador 2 detrás
FinalCinta1	%IX0.4	Pieza en final de cinta 1
PFresadora	%IX0.5	Pieza en fresadora
InicioCinta1	%IX0.6	Pieza en inicio de cinta 1
PTaladradora	%IX0.7	Pieza en taladradora
FinSalida	%IX1.0	Pieza en final cinta 2
SALIDAS		
AvEmp1	%QX0.4	Mover empujador 1 hacia adelante
RtEmp1	%QX0.5	Mover empujador 1 hacia atrás
AvEmp2	%QX0.6	Mover empujador 2 hacia adelante
RtEmp2	%QX0.7	Mover empujador 2 hacia atrás
Cinta1On	%QX1.0	Cinta 1 en marcha
CintaFresadoraOn	%QX1.1	Cinta fresadora en marcha
Fresadora	%QX1.2	Fresadora en marcha
CintaTaladradoraOn	%QX1.3	Cinta taladradora en marcha

Taladradora	%QX1.4	Taladradora en marcha
CintaSalida	%QX1.5	Cinta 2 en marcha
POWER	%QX1.6	Encender sensores
Cuenta	%MW3	Contar el número de piezas producidas

Tabla 6. Variables del prototipo de fresado y taladrado

El proceso en modo automático deberá realizar las siguientes acciones:

1. Se pulsará el botón *Encendido* y se encenderán todos los sensores del prototipo mediante la activación de la salida *POWER*.
2. Si la variable *Manual* esta desactivada y se coloca una pieza al principio de la cinta 1 (se desactiva *InicioCinta1*), al pulsar *Inicio* se activará la salida *Cinta1On* y la cinta 1 avanzará hasta llegar al empujador 1 (cuando se desactiva la variable *FinalCinta1*).
3. El empujador 1 avanzará hacia adelante empujando la pieza hasta que se active *FCF1*, entonces el empujador empezará a retroceder hasta que se active *FCT1* y al mismo tiempo se activará la salida *CintaFresadoraOn* y la cinta de la fresadora se pondrá en marcha.
4. Cuando la pieza llegue a *PFresadora* la cinta parará y se activara *Fresadora* durante 4 segundos. Tras este tiempo la cinta volverá a ponerse en marcha, esta vez durante 2 segundos para que dé tiempo a transferir la pieza a la cinta de la taladradora. Al mismo tiempo que se activa la cinta de la fresadora, se activará también la cinta de la taladradora para facilitar la transferencia de la pieza.
5. Cuando la pieza llegue a la posición de la taladradora (*PTaladradora*) esta se activará durante 4 segundos al igual que la fresadora. Transcurridos los 4 segundos la cinta de la taladradora volverá a encenderse para mover la pieza hasta el empujador 2. La cinta permanecerá encendida durante 2 segundos para asegurar que la pieza sale de la cinta, ya que no hay un sensor que indique el final de esta.
6. El empujador 2 avanzará hacia adelante transfiriendo la pieza a la cinta 2, la cinta donde termina el proceso. Cuando se active *FCF2* el empujador 2 parará y empezará a retroceder hasta que se active *FCT2* y al mismo tiempo que retrocede se pondrá en marcha la cinta 2.
7. La cinta 2 parará cuando se desactive *FinSalida*, entonces se esperará en este estado hasta que se retire la pieza.

Este proceso deberá admitir más de una pieza al mismo tiempo. En cuanto la cinta 1 quede disponible se debe poder iniciar el proceso para otra pieza diferente.

Además del modo automático, el proceso también tiene un modo de funcionamiento manual como se ha dicho anteriormente. Este modo sirve para accionar manualmente cualquiera de los actuadores del prototipo.

También está el modo de parada de emergencia que bloquea todos los movimientos del proceso excepto los del modo manual para permitir mover los actuadores si es necesario.

5.2.2 Iniciar nuevo proyecto en SoMachine

En este apartado se explicará cómo se debe iniciar un proyecto nuevo en el software SoMachine en su versión V4.1, aunque en la versión más reciente (la V4.3) el procedimiento es el mismo.

Al iniciar el SoMachine aparece la ventana mostrada en Ilustración 32. En ella se debe seleccionar la opción *Nuevo proyecto* para comenzar con el proyecto que se quiere diseñar.

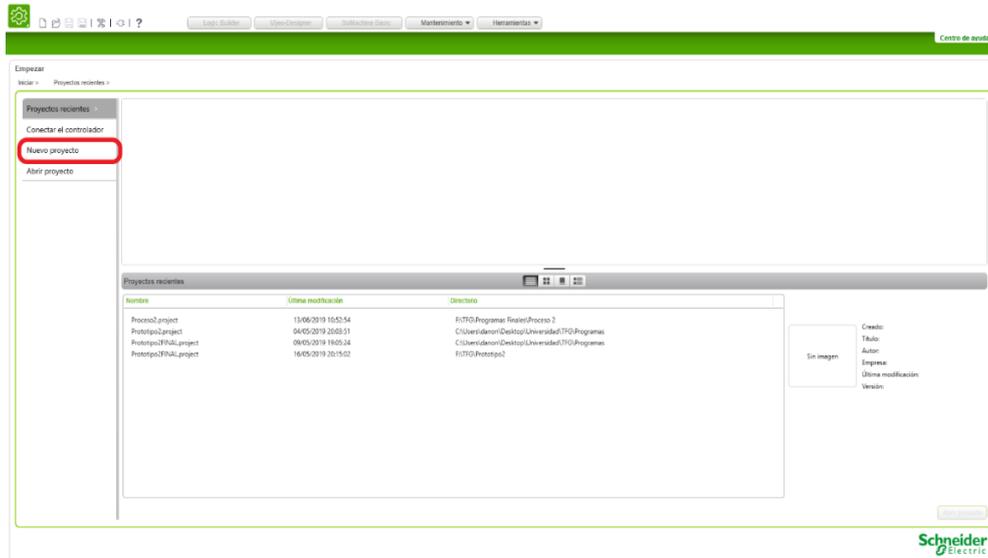


Ilustración 32. Ventana de inicio de SoMachine

Una vez seleccionada esa opción se debe seleccionar la opción *Proyecto vacío* y se abrirá una ventana en la que se debe introducir el nombre del nuevo proyecto. Al proyecto del prototipo de fresado y taladrado se le ha dado el nombre de *Proceso 2* en este trabajo. Una vez se haya introducido el nombre se pulsa en *Aceptar* y aparecerá el menú de configuración del proyecto que se muestra en la Ilustración 33. Ventana de configuración de proyecto de SoMachine. Aquí se tiene que seleccionar primero la opción *Añadir y eliminar dispositivos* marcada con un 1 en la ilustración. Aquí se debe seleccionar el dispositivo que se usa en el proyecto, en este caso el controlador lógico TM241CE40R, como se muestra en la Ilustración 34. Selección de dispositivo y una vez seleccionado se pulsa en la flecha que va hacia la derecha para que se cargue el dispositivo en el proyecto.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

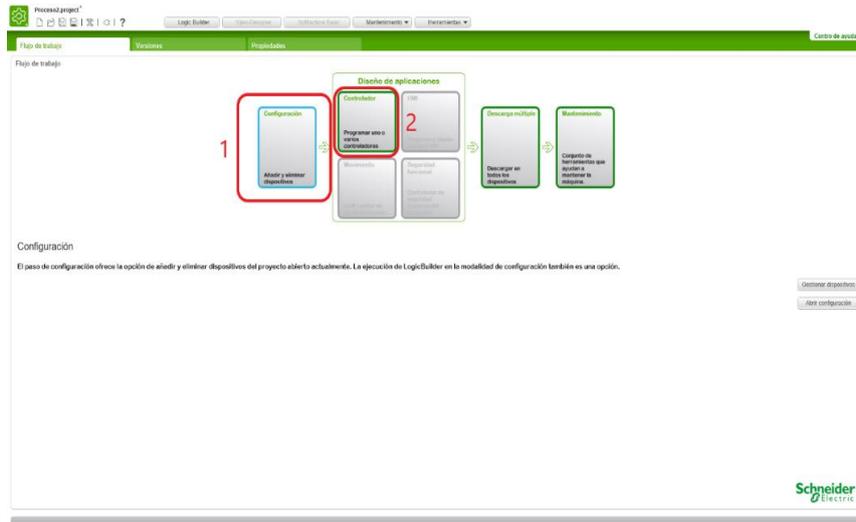


Ilustración 33. Ventana de configuración de proyecto de SoMachine

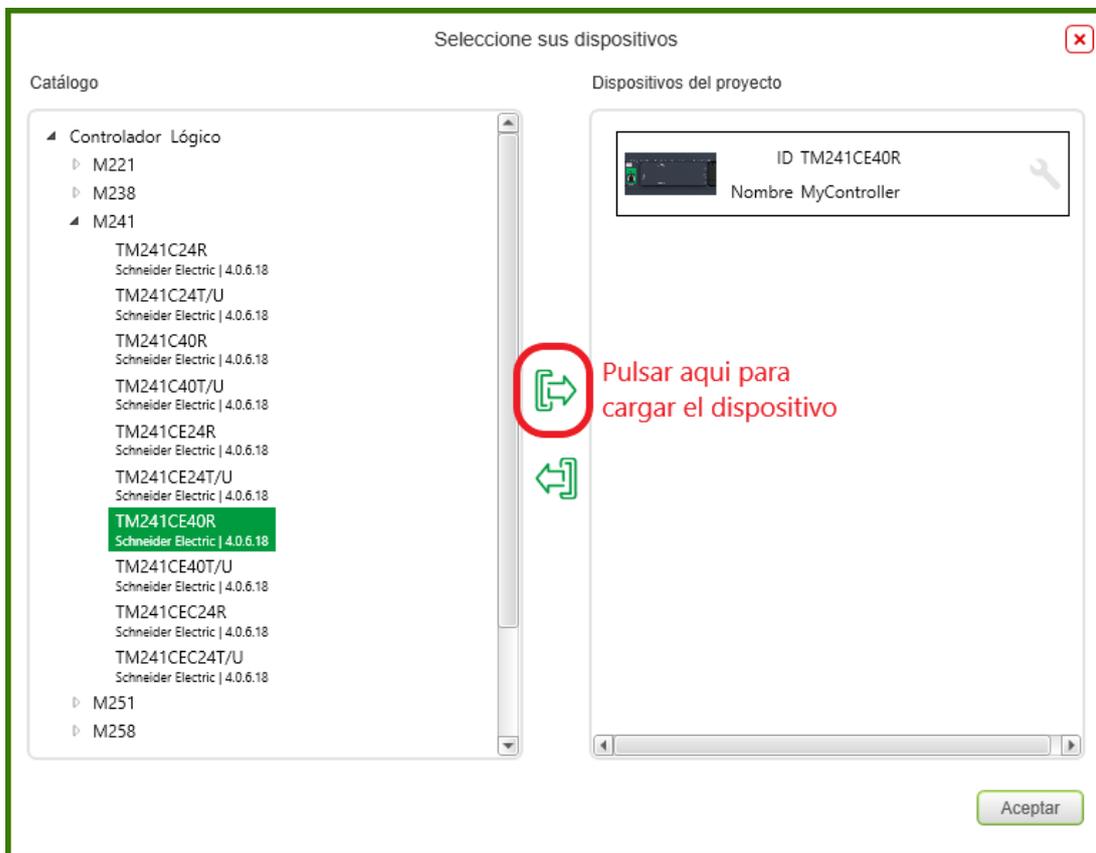


Ilustración 34. Selección de dispositivo

Tras la selección del dispositivo se vuelve a la ventana de configuración de proyecto y ahora se selecciona la opción *Programar uno o varios controladores* marcada con un 2 en la Ilustración 33. Así se abrirá el proyecto que se ha creado, apareciendo la ventana que se representa en la Ilustración 35 . Aquí es donde se debe escribir el código del programa. Es recomendable poner nombre a las entradas y salidas del autómata para facilitar la programación de este. Para ello en la pestaña inferior *Dispositivos* se selecciona DI para

cambiar el nombre de las entradas y DQ para cambiar el nombre de las salidas. Hay que tener en cuenta que para poder nombrar las salidas se debe borrar primero el nombre que hay en la casilla marcada en la Ilustración 35 con el rótulo *Borrar nombre de aquí*.

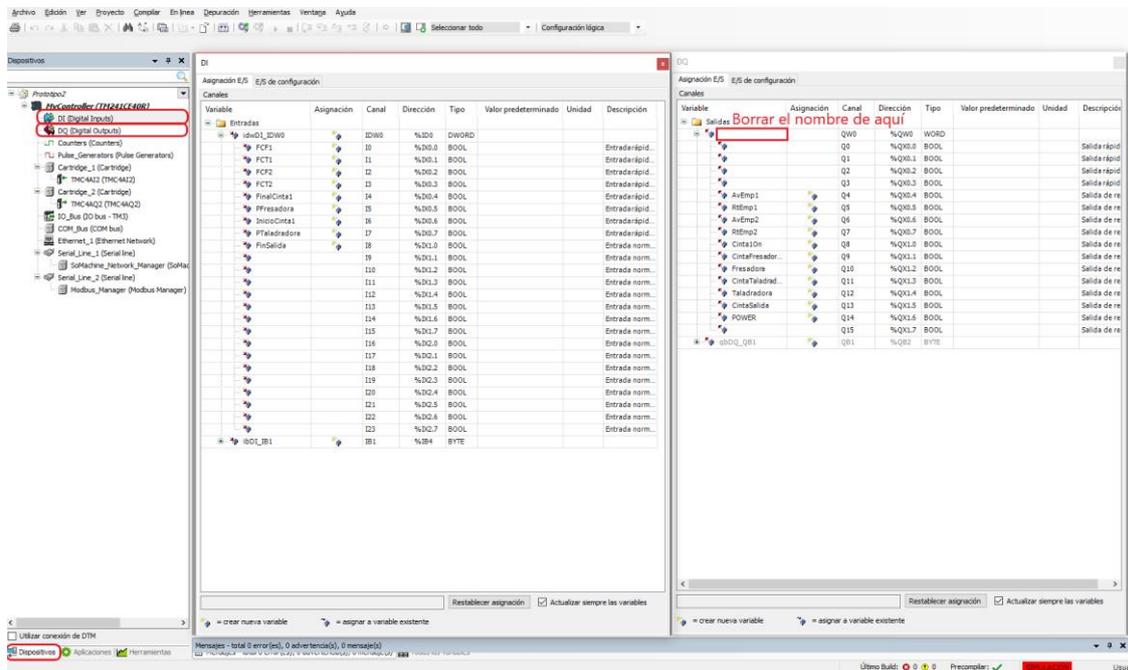


Ilustración 35. Ventanas de entradas y salidas digitales en SoMachine

En la pestaña inferior de *Aplicaciones* se entra en el menú donde se deben realizar los programas del automatismo. En este software a estos programas los llama POU y se pueden crear tantos como sean necesarios (el único límite es la memoria del PLC). Para crear un nuevo POU se selecciona la opción *Application* (hay que clicar en el símbolo verde con un + para crear seleccionar esta opción) y se abrirá un desplegable en el que se debe seleccionar POU (Ilustración 36).

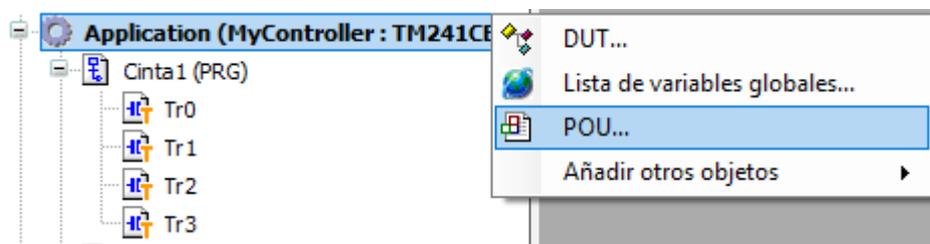


Ilustración 36. Creación de un nuevo POU

Después se elige el nombre del nuevo POU y se selecciona en qué tipo de lenguaje entre los disponibles en SoMachine se quiere programar. En este trabajo como se ha dicho anteriormente solo se han utilizado los lenguajes LD y SFC para el diseño de os programas de este prototipo.

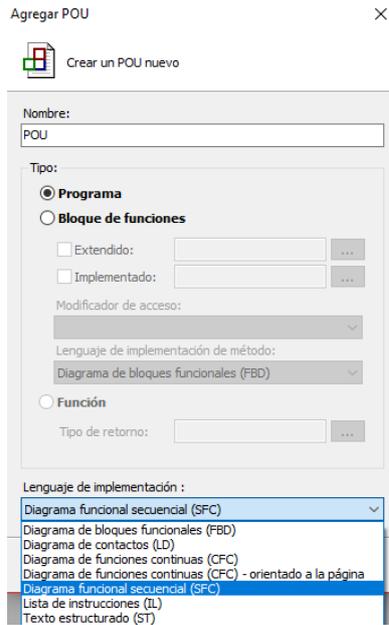


Ilustración 37. Menú de configuración de un nuevo POU

El PLC que se ha utilizado para realizar este proyecto tiene incorporados unos módulos de entradas y salidas analógicas (como se ha dicho en el apartado 4.1 Modicon M241) por lo que deben añadirse en el proyecto de SoMachine. Para ello se debe ir a la pestaña *Dispositivos* y en *Cartridge 1* y *Cartridge 2* se deben incluir los módulos correspondientes, en este caso el TMC4AI2 en se selecciona en *Cartridge 1* y el TMC4AQ2 se elige para el *Cartridge 2*. Con esto se termina de hacer la configuración del nuevo proyecto en SoMachine.

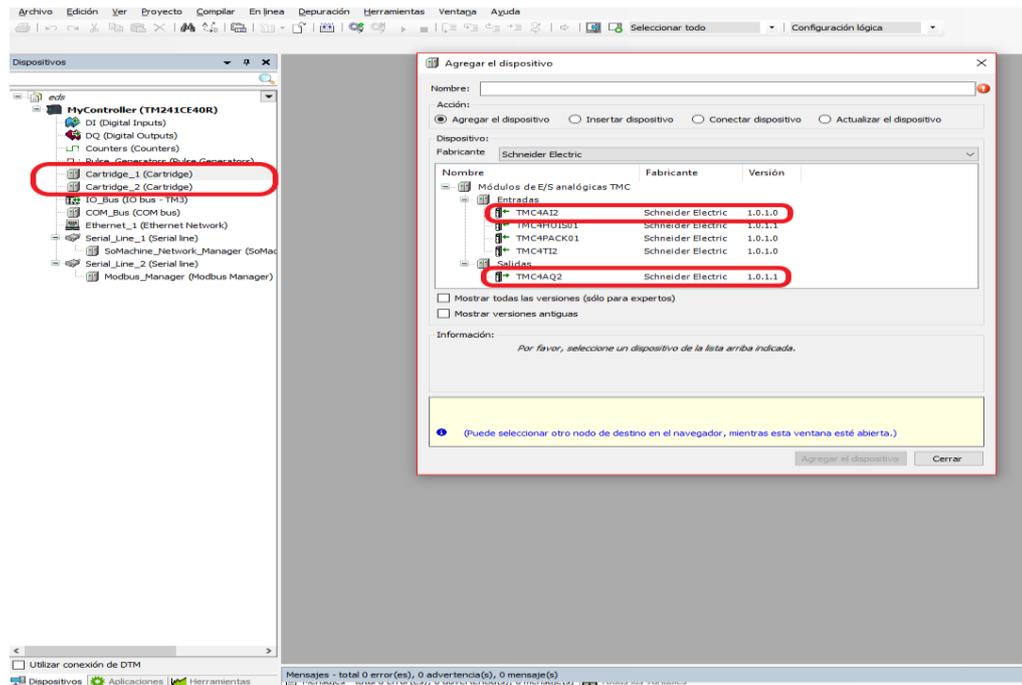


Ilustración 38. Selección de módulos de E/S analógicas

Después deberá introducirse la IP del PLC utilizado en el proyecto. Para ello en la pestaña *Dispositivos>Ethernet_1* se abre la ventana de la Ilustración 39.

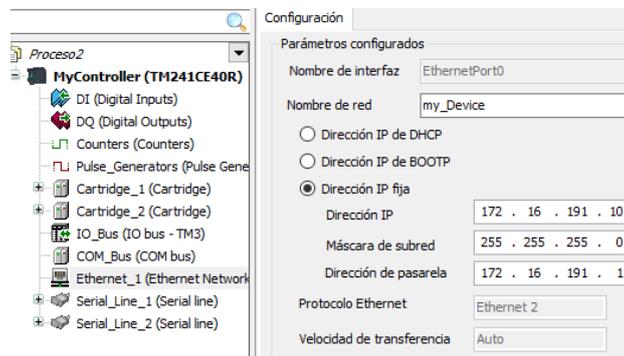


Ilustración 39. Asignación de IP al PLC

5.2.3 Programa del prototipo de fresado y taladrado

Para realizar este programa se han utilizado los lenguajes Grafcet o SFC y el diagrama de contactos o LD. Principalmente el diagrama de contactos se ha utilizado para configurar las transiciones del Grafcet. Las transiciones se pueden añadir en SoMachine añadiéndolas en los POU como se indica en la Ilustración 40. Se configuran como si fuera un POU, se le pone un nombre y se elige en que lenguaje se quiere programar, aunque en las transiciones no hay tantos tipos de lenguaje de programación disponibles.

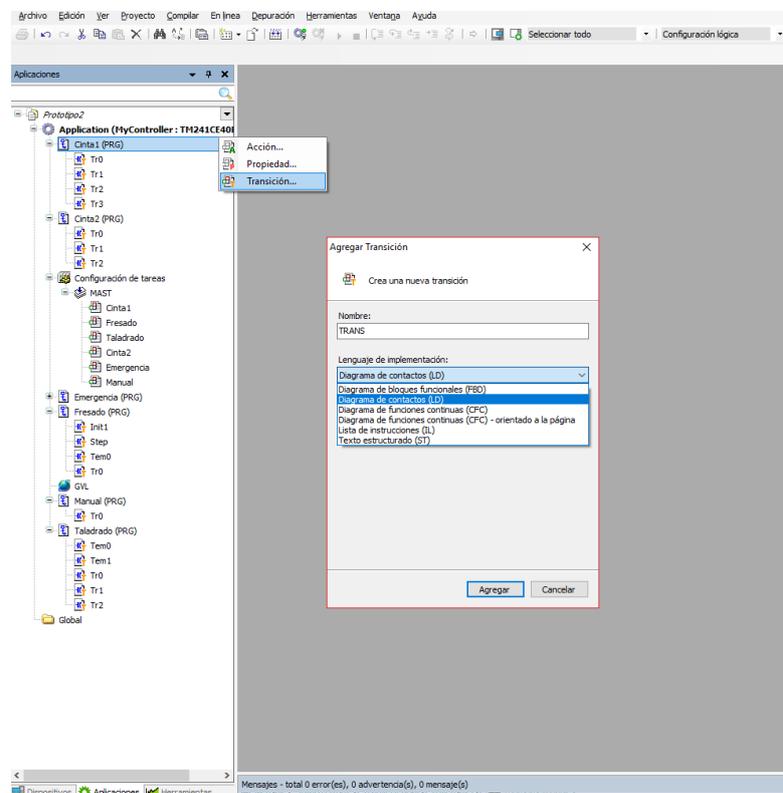


Ilustración 40. creación de una transición en un POU

Para poder realizar la admisión de más de una pieza en el proceso al mismo tiempo se ha optado por realizar el programa mediante varios POU programados en SFC y LD. Primero se va a detallar el funcionamiento del proceso en modo automático. Este modo dispone de cuatro graficets separados (Cinta1, Fresado, Taladrado y Cinta2) que actúan de un modo sincronizado. Para que actúen conjuntamente se ha indicado como condición de una transición de una etapa a otra la activación de otra etapa de otro Graficet distinto. Para poder hacer esto hay que usar la nomenclatura *Nombre del Pou.Etapa.x*, por lo que, si se quiere poner como condición la activación del Step2 del POU Cinta1, habría que escribir *Cinta1.Step2.x* y esta variable se pondría a 1 cuando se activa esa etapa. Cada uno de los Graficet controlará una de las cintas de transporte del prototipo, por los que los procesos realizados por cada uno son los descritos a continuación:

- Cinta1: Se encarga del movimiento de la cinta 1, así como del movimiento del empujador 1. Este Graficet es el que controla también el encendido y apagado de la máquina por los que dependiendo de si está el botón *Encendido* en 1 o en 0 se admitirán más piezas o no, por eso este Graficet es considerado como el Graficet principal del proceso. Si esta *Encendido* activado volverá a su etapa 1(Step0) y si no volverá a su estado inicial (Init) no se admitirán más piezas.
- Taladrado: Este Graficet controla la cinta de la fresadora y la puesta en marcha de la fresadora también. Esperará a que el POU Cinta1 empiece a retroceder el empujador 1 para comenzar a ejecutar las ordenes incluidas en él. Cuando halla transferido la pieza a la cinta de taladrado volverá a su estado inicial.
- Taladrado: Aquí se controla la cinta de la taladradora y la taladradora, así como el movimiento del empujador 2. Este Graficet empezará a ejecutarse cuando el POU Fresado termine de realizar el fresado de la pieza. Cuando el empujador 2 halla retrocedido completamente volverá a su estado inicial.
- Cinta2: Empieza a ejecutarse cuando El POU Taladrado a colocado la pieza en la cinta 2 mediante el empujador 2. Este POU moverá la cinta hasta que la pieza llegue al final y esperará a que la retire un operario antes de volver a su estado inicial.

Como puede haber más de una pieza produciéndose para evitar la transferencia de una pieza a un Graficet que todavía no ha acabado sus órdenes se ha puesto como condición antes de empezar la transferencia de la pieza que este activado el Init (estado inicial) del Graficet que viene a continuación, esto es, el programa espera a que el siguiente Graficet este libre para enviarle la pieza. Se puede ver esto en el ejemplo que representa el Graficet de la cinta 1 esperando a Init del Graficet de la fresadora para transferir la pieza.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

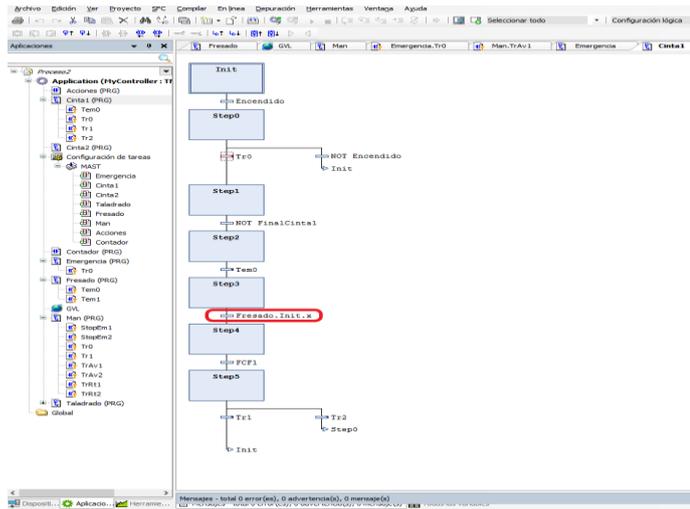


Ilustración 41. POU Cinta1

Cabe destacar que, como se puede ver en la Ilustración 41 no hay acciones asociadas a las etapas. Esto es porque durante la programación de estos POU se incluyeron en un principio las salidas como acciones en cada una de las etapas de los POU existentes, pero esto provocaba que una misma salida estuviera puesta a 1 y a 0 al mismo tiempo y el prototipo no se ponía en marcha o no funcionaba correctamente. Para solucionar esto se optó por crear un POU en lenguaje LD en el que solo se pone una bobina por salida y en paralelo todos los contactos que hacen referencia a las etapas de todos los Grafcet del proyecto. Así se consigue que solo haya una salida para cada salida digital física y se soluciona el problema mencionado anteriormente.

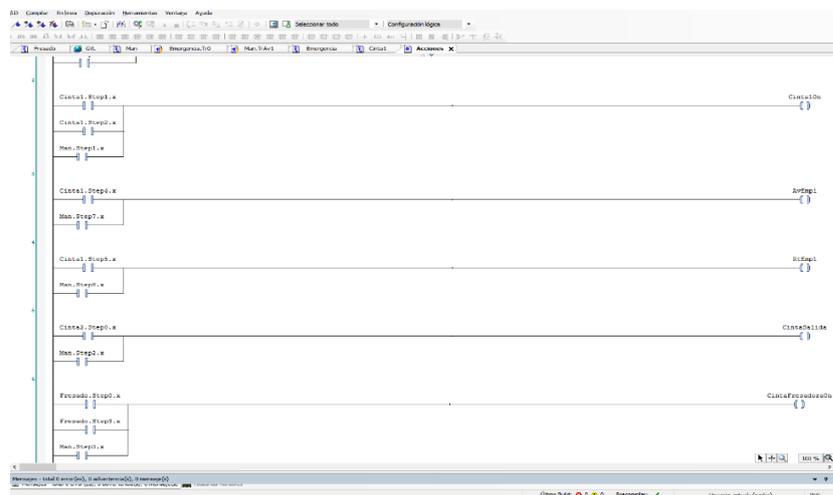


Ilustración 42. Parte del código del POU Acciones

También es importante mencionar como es la declaración de variables en este software. Se puede declarar variables globales o locales. Como este proyecto dispone de varios POU diferentes es interesante declarar todas las variables comunes a todos como variables

globales. Para ello se sigue la ruta *Aplicaciones>GVL*. Las variables globales se declaran como se muestra en la Ilustración 43.

```
1 VAR_GLOBAL
2   Emergencia AT %MX0.0: BOOL;
3   Manual AT %MX0.1: BOOL;
4   Encendido AT %MX0.2: BOOL;
5 END_VAR
```

Ilustración 43. Declaración de variables globales en SoMachine

Las variables locales se declaran dentro del POU correspondiente. Pueden declararse de dos formas distintas: escribiendo el nombre de la variable directamente, entonces SoMachine detectará que esa variable no existe y preguntará si se quiere declarar (opción 1) o pulsando en las flechas que aparecen en la parte superior dentro del POU e introducir la variable como en el caso de las variables globales (opción 2).

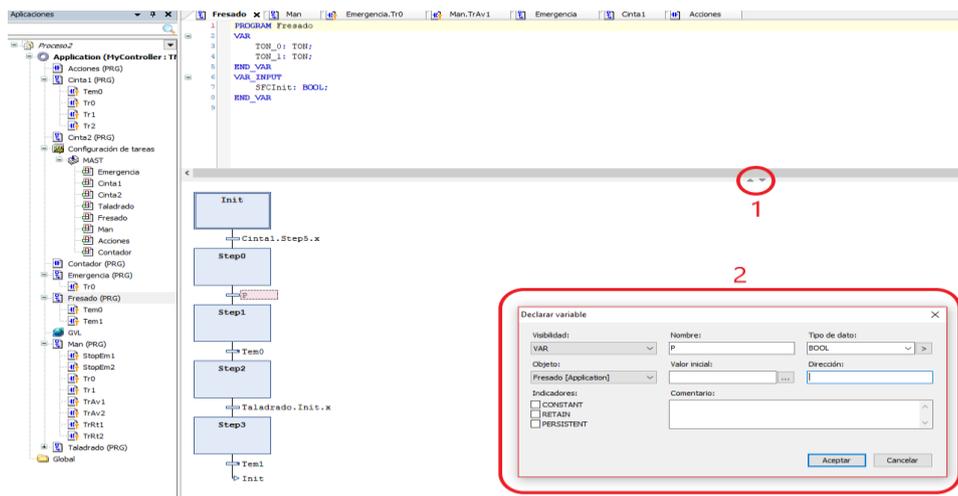


Ilustración 44. Declaración de variables locales en SoMachine

A parte del funcionamiento automático está el funcionamiento manual realizado mediante el POU Manual. Este POU dejará de estar en su estado inicial solo si se activa al mismo tiempo *Encendido* y *Manual* y no se está ejecutando el modo automático. Estas condiciones se pueden ver más fácilmente en la Ilustración 45. Una vez en el modo manual se pueden activar manualmente cada una de las cintas, los empujadores, la fresadora o la taladradora.

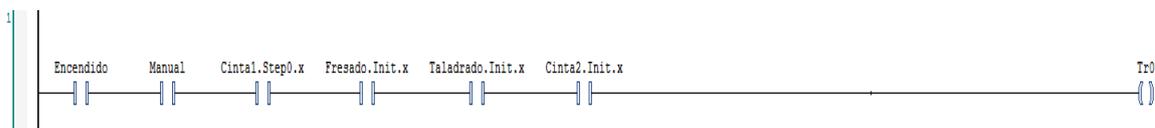


Ilustración 45. Condición de accionamiento del modo manual

Para controlar la seguridad del proceso está el POU Emergencia con un Grafcet que paraliza los movimientos del modo automático, pero no para los movimientos del modo manual. Para poder realizar el bloqueo de los POU del modo automático se ha usado las variables SFCInit que fuerzan los Grafcet al estado inicial. Estas variables se declaran como una variable local de cada POU usando la nomenclatura SFCInit: BOOL; y después deben activarse haciendo clic derecho en el nombre del POU, después se selecciona *Propiedades* y en la ventana que se abre se selecciona *Configuración SFC* y se deshabilita la opción *Utilizar los valores predeterminados*. Tras esto se marca la opción *SFCInit* y después se selecciona *Aceptar*. Después en el Grafcet de emergencia se ponen estas variables como acciones, como se muestra en la Ilustración 46.

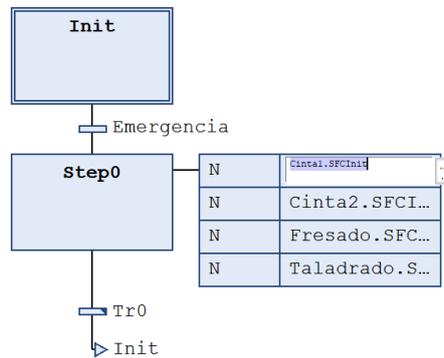


Ilustración 46. Grafcet de emergencia

Todos los POU de este proyecto se pueden ver en el apartado 2. Programa SoMachine de los anexos.

5.2.4 Transferencia del programa al autómatas

Una vez terminado el programa debe cargarse en el PLC. Para ello, en la pestaña *Dispositivos* se selecciona la opción *MyController* y se selecciona el controlador que corresponda con la dirección IP del PLC utilizado. Tras esto se clicca en el icono de cargar y se pulsa Alt+F cuando lo pida y así se carga el programa en el autómatas. Tras esto se clicca en el botón con forma de *Play* y se pone el autómatas en modo Run.

5.3 Diseño de la aplicación SCADA

La pantalla HMI es utilizada para el control simultaneo de los dos procesos de producción simulados en este proyecto. A esta aplicación se le han añadido varias funciones, las más importantes son las de control de los procesos y monitorización de los sensores activados en los prototipos. También dispone de un registro de eventos que detecta cuando se ha activado la parada de emergencia y cuando se ha activado el modo manual, así como la posibilidad de crear un nivel de usuario para restringir el acceso a algunas ventanas de la aplicación. Para la programación de la aplicación se ha usado el software HCTDesigner 2 suministrado por HCFA para la programación de sus tabletas HMI.

5.3.1 Inicio de nuevo proyecto en HCTDesigner 2

Cuando se abre este programa se abrirá directamente el espacio para trabajar en el diseño de la aplicación, pero primero se debe abrir un nuevo proyecto. Para ello se debe seguir la ruta *File>Crear nuevo proyecto* y se abrirá una ventana para configurar la tableta HMI. Primero se debe elegir el nombre del proyecto y donde se va a guardar. Después se escogerán las características de la HMI, que debe ser *Tamaño del LCD 7 pulgadas*, *Resolución de la pantalla 800x480* y *Tipo de dispositivo HMI TP2507/TP2510*. Después se debe configurar la dirección IP como se muestra en la Ilustración 47. Este paso es muy importante ya que si no se introduce correctamente la dirección IP de la pantalla, cuando se cargue el proyecto en esta dará error, pero además no dejará comunicarse con la pantalla a través de ethernet, aunque se cambie la dirección IP y se deberá conectar a la pantalla a través de USB para cambiarle la dirección IP. Y por último se debe asociar un PLC eligiendo la opción *PLC Ethernet* y pulsando en *Añadir*, aquí se debe introducir la IP del autómata y seleccionar la marca y modelo (Siemens S7-1200 Network y Schneider Modbus TCP). Tras esto se debe pulsar en *Confirmar* y ya se puede empezar a trabajar.

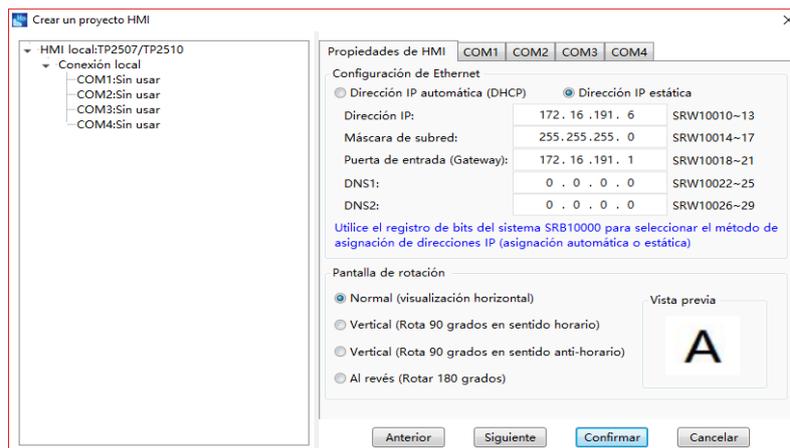


Ilustración 47. Configuración de la IP de la pantalla HMI

5.3.2 Diseño aplicación para prototipo estampadora

La ventana de esta aplicación llamada proceso 1 sirve para interactuar con el prototipo de la estampadora. Desde aquí se puede activar la emergencia, cambiar entre modo manual y automático, iniciar la producción de una pieza y activar los actuadores en modo manual, además se puede ver el número de piezas producidas y cambiar entre las demás ventanas del proceso. Esta ventana se puede ver en la Ilustración 48, con todos los botones rotulados con su función.

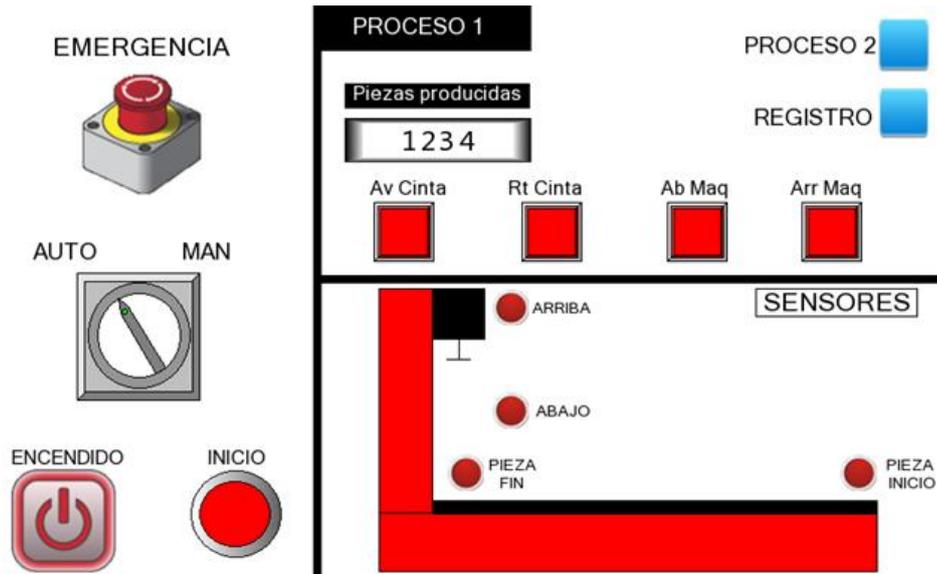


Ilustración 48. Ventana de control del proceso de estampación

Los distintos tipos de elementos que se han introducido en esta ventana son los siguiente:

1. Texto: Usado para escribir los rótulos. Se puede ajustar el estilo de letra, tamaño, color, etc.
2. Bit de configuración: Utilizado para hacer pulsadores. Pulsando en la flecha de al lado se abren más opciones de pulsador. Con esta opción se han creado el botón inicio (eligiendo un pulso de encendido de 0,2s) y los botones de cambio de ventana (seleccionando operación de ventana) con los que se puede cambiar de ventana o cerrarlas. Todos los botones para cambiar de ventana que se han utilizado en este proyecto tienen la forma de un rectángulo azul.
3. Bit indicador de luz: Para crear indicadores de luz. Estos solo son de lectura, no pueden escribir un bit o palabra (16 bits), solo pueden leer una variable. Así se han hecho los indicadores de los sensores.
4. Visualizador de valor numérico: Usado para el contador (piezas producidas). Solo puede leer palabras, no puede asociarse con la dirección de un bit.
5. Bit de cambio: Usado para hacer interruptores. Con esta opción se han creado los botones de emergencia, manual, encendido y los controles manuales (botones situados encima de la zona de los sensores). Para que sea un interruptor debe elegirse la opción *Inverso* en la ventana que se abre en la pestaña *Interruptor de palanca*.

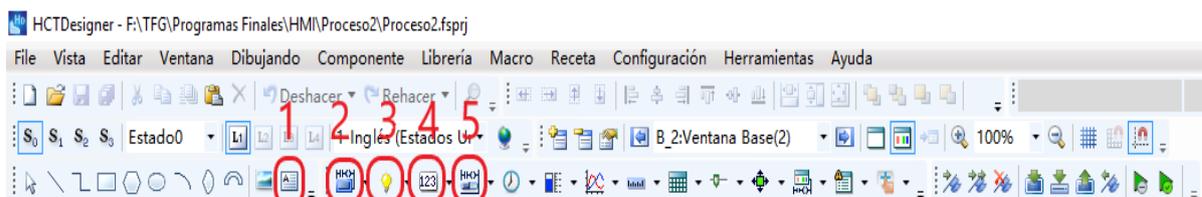


Ilustración 49. Barra de herramientas de HCTDesigner 2

Para que todos los pulsadores e indicadores funcionen correctamente deben asociarse con una dirección de la memoria del autómat. Para ello en la ventana mostrada en la Ilustración 50, que se abre al seleccionar un elemento, se debe elegir en *Dispositivo* el controlador de Siemens, en el desplegable de *Tipo de dirección* se selecciona a que parte de la memoria del PLC se quiere acceder (Entradas I, salidas Q o memoria interna M) y finalmente en *Dirección* se escoge la dirección de memoria a la que se accede. Como dirección debe escogerse la dirección de memoria del autómat en la que se ha guardado la variable a la que se quiere acceder (por ejemplo, para acceder al botón de emergencia habría que escribir M.B2.6 ya que esta variable está en la posición de memoria %M2.6). Además, en la pestaña de gráficos se puede cambiar la imagen que representa al elemento que se ha creado, pudiendo importar imágenes nuevas de una librería que trae incorporada el HCTDeigner 2.

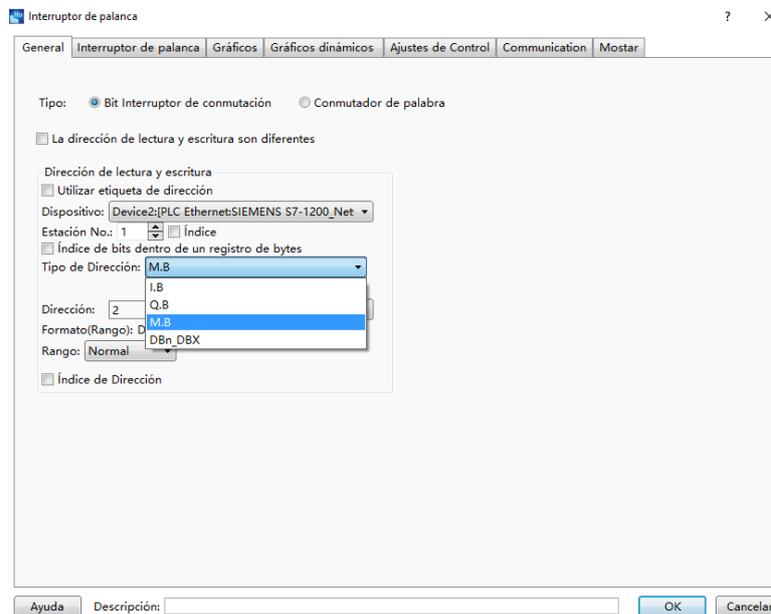


Ilustración 50. Asignación de dirección de memoria a un interruptor

También en los símbolos que aparecen al lado de *B_2V:entana Base(2)* en la Ilustración 49 se pueden crear, borrar y editar las ventanas creadas. En propiedades se pueden ajustar los tamaños de las ventanas y los colores. Para dibujar la silueta de la estampadora, así como las líneas de separación y los rectángulos de los rótulos se han usado las herramientas de dibujo situadas en la parte inferior izquierda de la barra de herramientas, con las que se pueden dibujar varias figuras geométricas. Si se hace clic derecho encima de una de estas figuras y se selecciona *Ajustar propiedades* se puede cambiar el color y la forma de la figura.

5.3.3 Diseño aplicación para prototipo fresado y taladrado

Para diseñar la ventana de control del proceso de fresado y taladrado se han utilizado los mismos elementos descritos en el apartado anterior para diseñar el control del prototipo de la estampadora. La ventana de control de este proceso puede verse en la Ilustración 51. La configuración de los elementos es muy parecida a la ventana anterior, disponiendo de un botón de parada de emergencia, selector de modo manual o automático, botón de encendido

de la maquina y pulsador de inicio del proceso automático. También dispone de un contador para ver el número de piezas que ha producido el proceso.

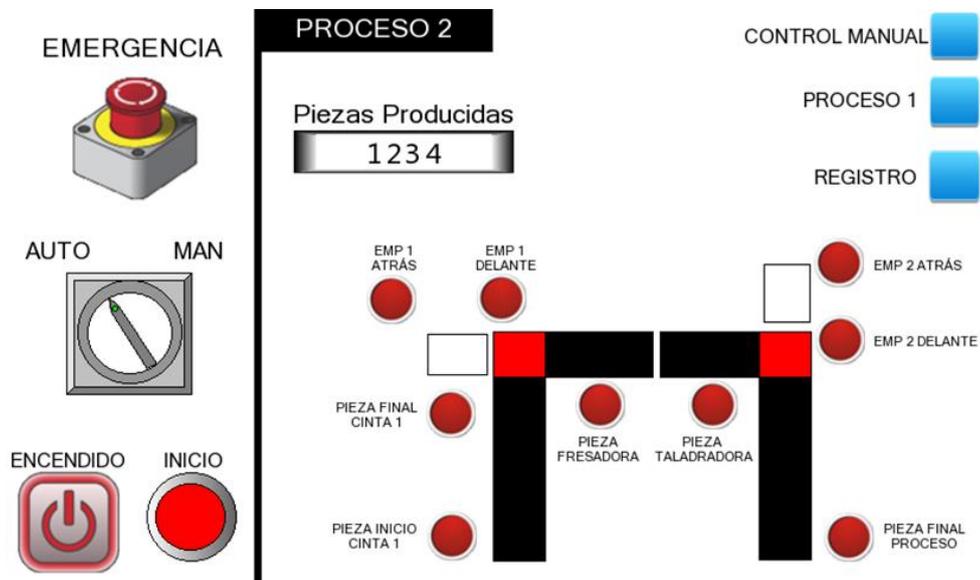


Ilustración 51. Ventana de control del proceso de fresado y taladrado

Una diferencia con respecto a la ventana del otro proceso es que el control manual ha tenido que realizarse en una ventana a parte a la que se accede en el botón *Control Manual* situado en la esquina superior derecha junto con todos los botones para realizar cambios de ventana. La ventana de control manual tiene el aspecto que se ve en la Ilustración 52. Se sale de esta ventana pulsando en la cruz de la esquina superior derecha (ese botón se hace con otra operación de ventana eligiendo la opción *Volver a la ventana anterior*).

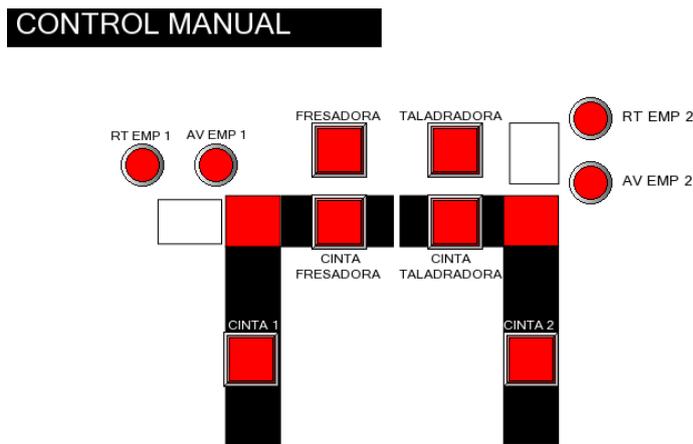


Ilustración 52. Ventana de control manual del proceso de fresado y taladrado

Para poder visualizar cuando una ha activado los sensores se debe activar en los sensores que se activan con la barrera LED la lógica negativa, ya que cuando se pone a 0 es cuando

hay una pieza en ese sensor. Los demás sensores (finales de carrera) siguen usando lógica positiva. En la Ilustración 53 se puede ver como activar la lógica negativa.

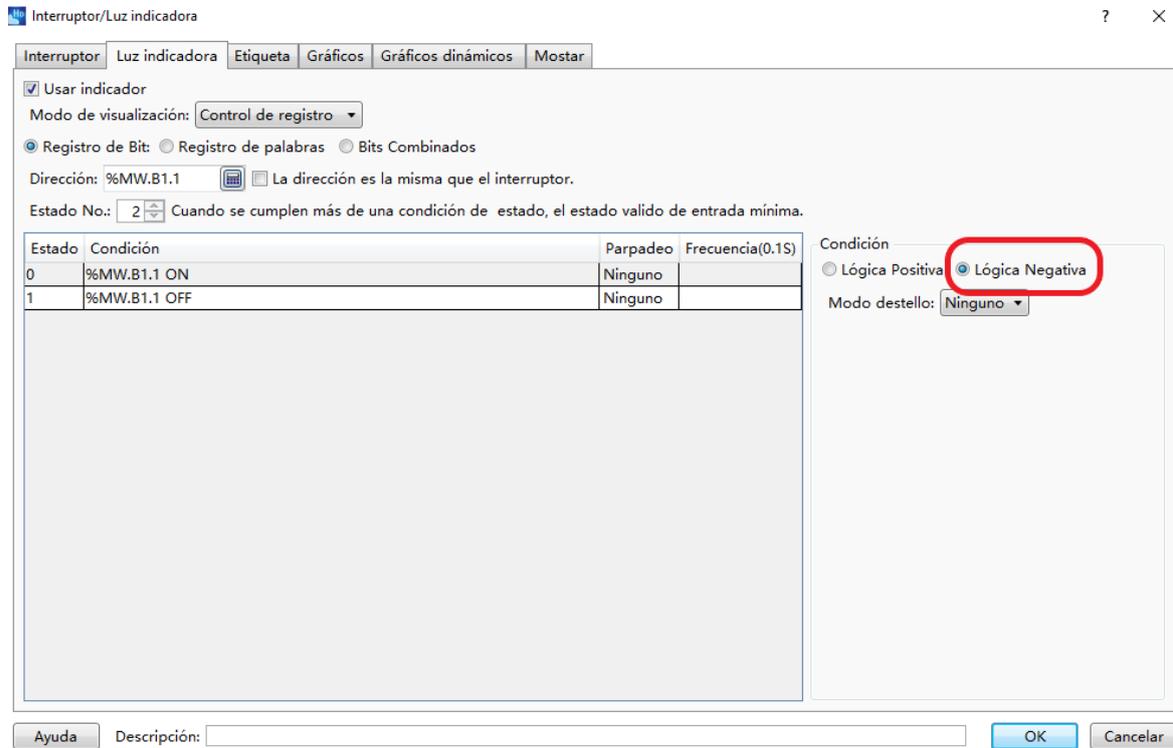


Ilustración 53. Como activar la lógica negativa en HCTDesigner 2

A la hora de asociar los elementos a la memoria del autómata, es diferente asociar direcciones de memoria para el S7-1200, donde basta con indicar la dirección de memoria directamente (por ejemplo, si se quiere asociar con un elemento la dirección %M1.1, en HCTDesigner 2 se debe introducir la dirección 1.1 para seleccionar este bit), sin embargo en el M241 no se pueden seleccionar las zonas de memoria reservadas a entradas y salidas físicas, esto quiere decir que no se pueden seleccionar las posiciones de memoria %IX ni %QX, solo se puede elegir la zona de memoria %MX, por eso en el programa de este prototipo se han tenido que copiar todas las variables asociadas a los sensores (entradas al PLC) en variables %MX para poder visualizarlas en la pantalla HMI. Sin embargo, hay que tener cuidado ya que en el programa HCTDesigner 2 se seleccionan los bits dentro de una palabra (conjunto de 16 bits) por lo que las direcciones en los bits quedan se cómo indica en la

Tabla 7. Por lo tanto, si se quiere acceder a la posición %MX1.6 en HCTDesigner 2 se debe seleccionar la dirección %MW0.14. Para elegir variables de tipo bool, o sea un bit específico dentro de la variable word se debe marcar la opción *Índice de bits dentro de un registro de bytes*.

Dirección en el PLC	Dirección en pantalla HMI	Word correspondiente
%MX0.0 a %MX0.7	%MW0.1 a %MW0.7	%MW0
%MX1.0 a %MX1.7	%MW0.8 a %MW0.15	
%MX2.0 a %MX2.7	%MW1.0 a %MW1.7	%MW1
%MX3.0 a %MX3.7	%MW1.8 a %MW1.15	
%MX4.0 a %MX4.7	%MW2.0 a %MW2.7	%MW2
%MX5.0 a %MX5.7	%MW2.8 a %MW2.15	
%MX6.0 a %MX6.7	%MW3.0 a %MW3.7	%MW3
%MX7.0 a %MX7.7	%MW3.8 a %MW3.15	

Tabla 7. Direccionamiento de variables para M241 en HCTDesigner 2

El dibujo que se ha usado para representar de forma simplificada el prototipo de este proceso se ha realizado también con las herramientas de dibujo de HCTDesigner 2.

5.3.4 Otras prestaciones de la aplicación HMI

A parte de la monitorización y control de ambos procesos, esta aplicación para la pantalla HMI también tiene otras prestaciones. Estas son las que se enumeran a continuación:

1. Registro de alarmas de los procesos
2. Nivel de usuario

La más importante de las dos prestaciones es el registro de alarmas que permitirá conocer que ha pasado en el proceso a lo largo del tiempo. Para llevarlo a cabo se ha creado otra ventana en la que se ha introducido un registro de eventos y alarmas. Para eso se debe seleccionar la opción que aparece en la Ilustración 54 y aparecerá un menú en el que se puede introducir todos los datos que el usuario quiere que aparezcan en la tabla de registro de alarmas y eventos. Antes de hacer esta tabla hay que crear las alarmas que se quiere que aparezcan en este registro. Para ello, siguiendo la ruta *Configuración>Ajustes de sistema>Alarma y evento*, se pueden introducir todos los eventos y alarmas necesarios.

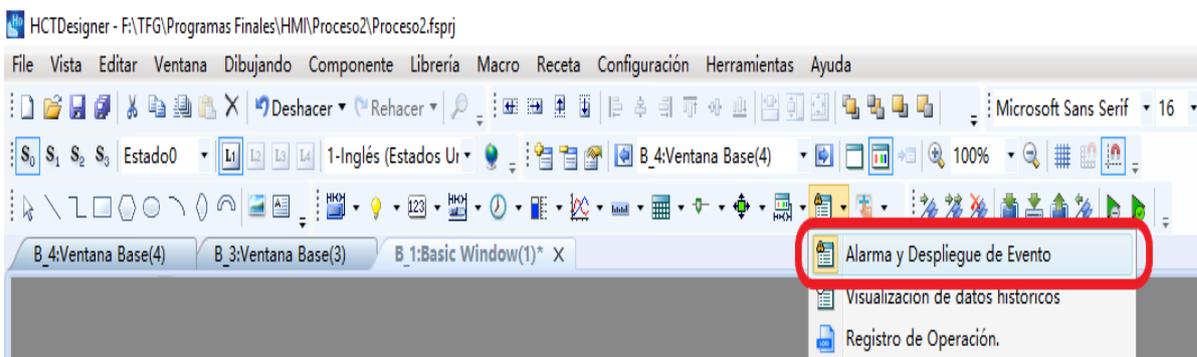


Ilustración 54. Selección de registro de alarmas y eventos

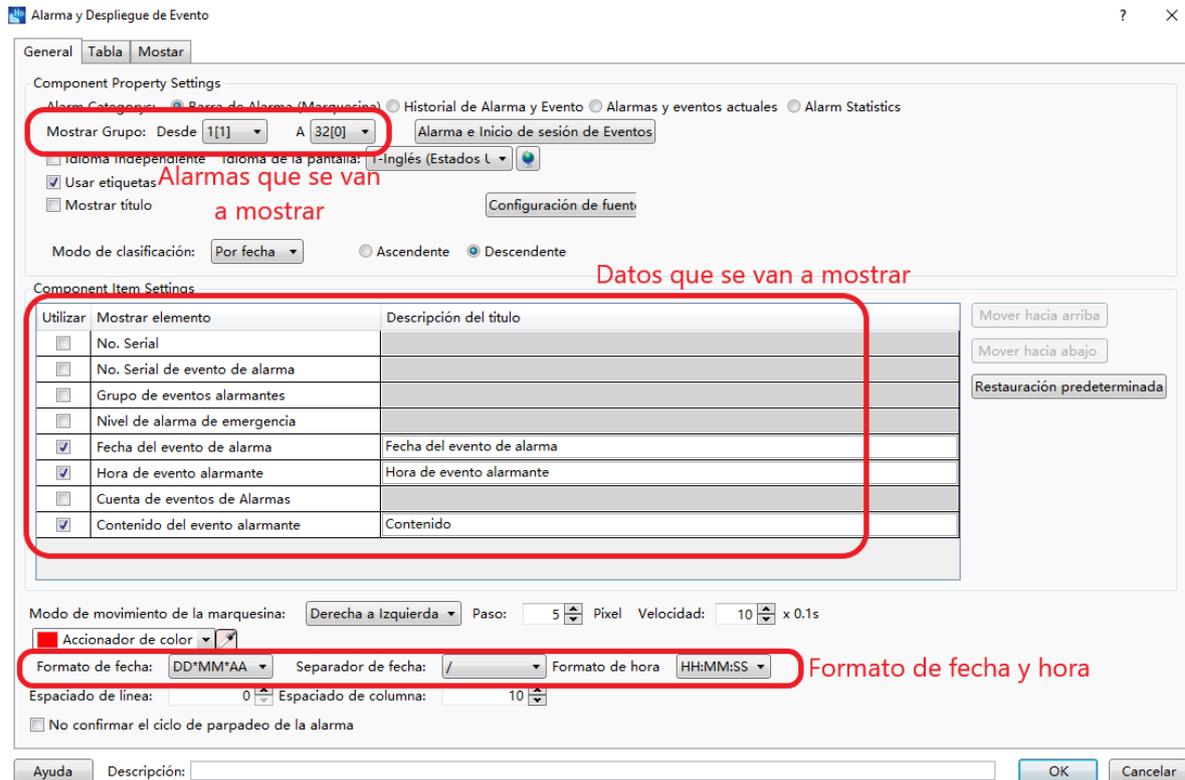


Ilustración 55. Configuración de la tabla de registro de alarmas y eventos

Al seguir esa ruta, se abrirá una ventana en la que aparecen todas las alarmas y eventos creados en el proyecto. Para introducir una nueva habrá que seleccionar *Crear* y se abrirá una ventana en la que se debe seleccionar la id del grupo (es el número que identifica a esa alarma y sirve para seleccionar cual aparece en la tabla de registro), la condición de activación de la alarma (por ejemplo, cuando se pulsa un botón), la descripción de la alarma (este mensaje aparecerá en la tabla de registro para saber que alarma es la que se ha activado) y, opcionalmente, se puede elegir si salta una ventana emergente o se activa un sonido al activarse esta alarma. En el caso de la alarma de parada de emergencia se ha creado una ventana emergente que da instrucciones de cómo salir del modo de emergencia cuando este se ha activado. La aplicación HMI creada para este proyecto dispone de cuatro alarmas, dos para cada emergencia de cada proceso y otras dos para cuando se pasa de modo automático a manual también en cada proceso. La información dada en la ventana de registro es la descripción de la alarma (que debe usarse para indicar que es lo que significa esa alarma) que aparece en la columna *Contenido* y la fecha y la hora en la que se ha producido la alarma.

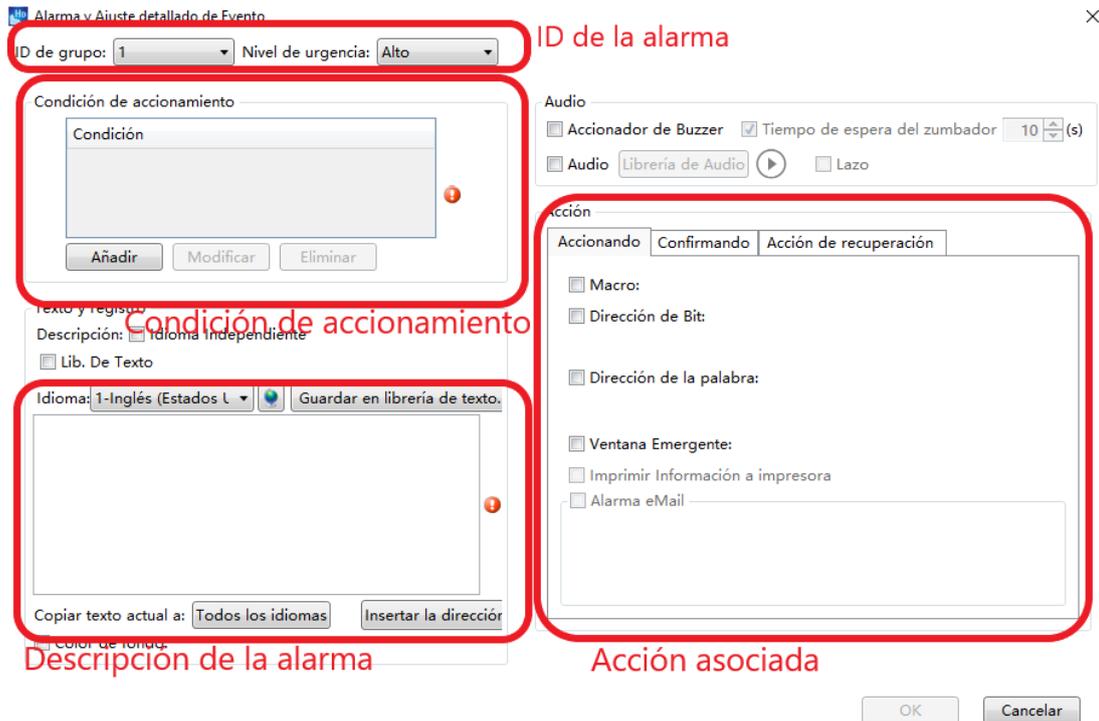


Ilustración 56. Ventana de configuración de una alarma

Fecha del evento de alarma	Hora de evento alarmante	Contenido
15/06/19	00:28:50	Modo manual activado en proceso 1.
15/06/19	00:28:47	Emergencia. Parada de proceso 1.
15/06/19	00:28:44	Modo manual activado en proceso 1.
15/06/19	00:28:36	Modo manual activado en proceso 2.
15/06/19	00:28:32	Emergencia. Parada de proceso 2.
15/06/19	00:28:26	Modo manual activado en proceso 2.

Ilustración 57. Registro de alarmas

El nivel de usuario sirve para restringir algunas opciones dentro de la aplicación de la pantalla. Para este proyecto esta opción ha sido utilizada para restringir el acceso a la ventana de registro de alarmas y eventos. Para esto se pueden crear varios niveles de usuario cada uno con su propia contraseña y se puede seleccionar a que nivel de usuario corresponde una ventana, por lo que al intentar acceder a ella se pedirá la contraseña correspondiente al nivel de usuario seleccionado para esa ventana.

Para crear niveles de usuario se debe seguir la ruta *Configuración>Ajustes de sistema>Nivel de usuario* y aquí aparecerá una tabla con los niveles de usuario de los que dispone el proyecto, así como sus contraseñas correspondientes. Para crear otro nivel de usuario se debe seleccionar el desplegable *Nivel de cuenta de usuario* y ahí se selecciona cuantos niveles se quieren tener en la aplicación (pudiendo crear hasta 16 niveles de usuario distintos) y en la tabla aparecerán todos los que hay. En la columna *Contraseña predefinida* se puede cambiar la contraseña de los niveles. Al crear una nueva ventana o ajustar las propiedades de una existente se puede elegir a que nivel de usuario es accesible esa ventana. Todas las ventanas están disponibles para el nivel de usuario 0 (sin contraseña) de forma predeterminada.

En la aplicación HMI que se ha creado para el control de los procesos de este proyecto se han introducido solo dos niveles de usuario. Uno es el nivel 0, el cual no tiene contraseña. Es el nivel de usuario predeterminado al iniciar la aplicación, con este nivel se puede acceder a la mayoría de las ventanas de la aplicación. El otro nivel sería el 1, este nivel sí que tiene una contraseña que sería conocida solo por el encargado del proceso y con ella podría acceder a la ventana del registro de alarmas y eventos. Así se restringe el acceso al registro a cualquier persona no autorizada

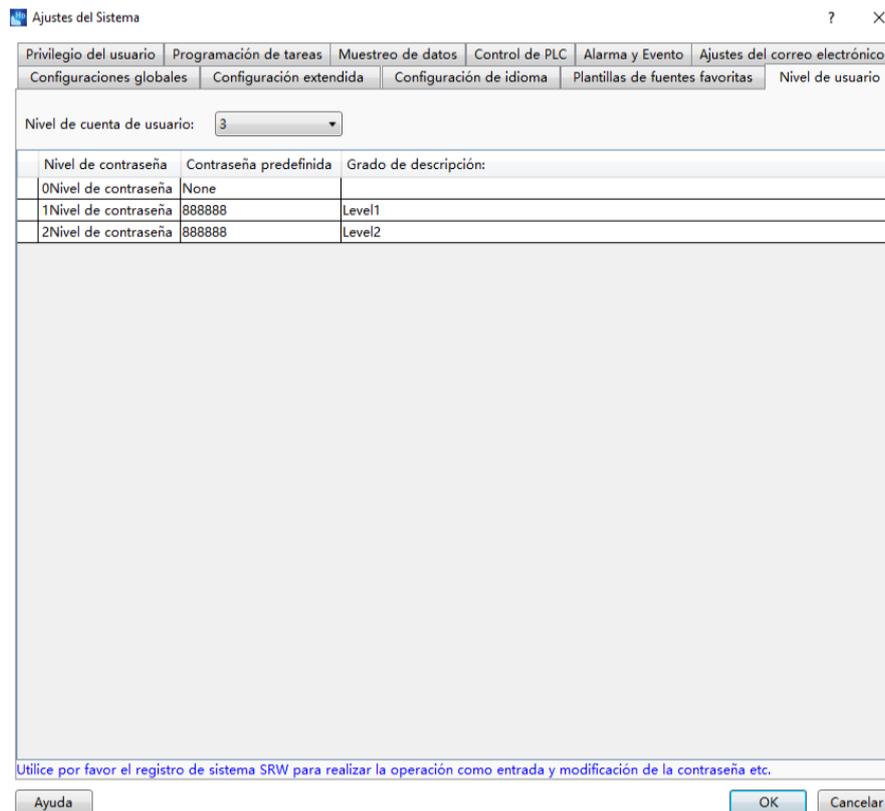


Ilustración 58. Creación de niveles de usuario

5.3.5 Transferencia de la aplicación a la pantalla HMI

Una vez que se ha terminado de diseñar toda la aplicación HMI llega el momento de transferirla a la pantalla. Para ello, en la barra de herramientas se debe pulsar en la opción de *Descargar todo* (marcada con un rectángulo rojo en Ilustración 59). También está la opción de *Cargar* que sirve para cargar en el PC una aplicación que está instalada en la pantalla HMI.

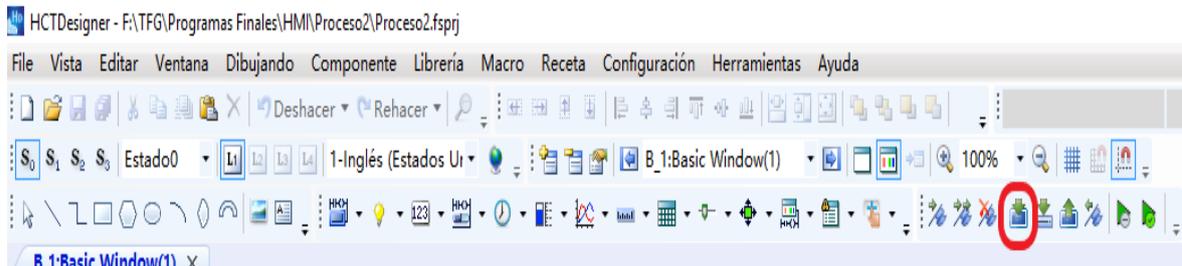


Ilustración 59. Cargar aplicación en HMI

Antes de realizar la transferencia de la aplicación a la pantalla HMI se debe conectar esta a la fuente de alimentación para que esta se encienda. La fuente de alimentación se enciende y se fija el voltaje regulable a 24V. Tras esto se conectan los cables de alimentación que trae la pantalla HMI a la fuente de alimentación, uniendo el cable rojo con el conector rojo y el cable negro con el conector negro.

Una vez seleccionada la opción de *Descargar todo* se debe buscar la tableta HMI en la que se quiere cargar el proyecto mediante su dirección IP. Se marca la opción *Ethernet*, se introduce la IP de la tableta y se pulsa en *Scan*. Tras esto se debe esperar a que la tableta HMI sea encontrada, cuando la encuentre se selecciona y se pulsa en *Aceptar*. Ahora ya solo queda pulsar en *Download* y el proyecto se cargará en la tableta HMI seleccionada.

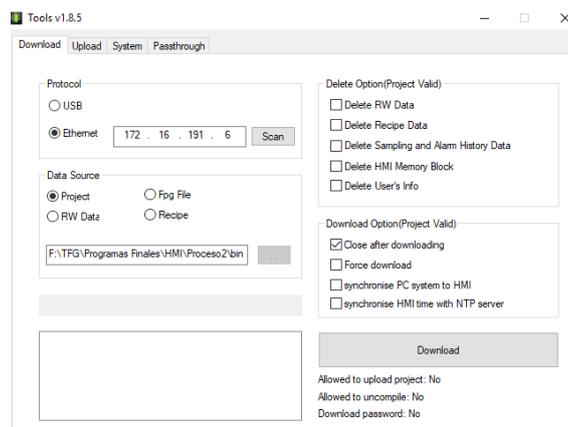


Ilustración 60. Selección de pantalla HMI a través de su dirección IP

5.4 Sincronización de ambos procesos

Para terminar con la descripción del desarrollo del proyecto, quedaría detallar como se ha realizado la sincronización de ambos procesos. En este proyecto se ha considerado que todas las piezas que han pasado por un proceso deben pasar por el otro obligatoriamente y el segundo proceso no se puede aceptar una pieza que no haya estado en el otro previamente. El orden que se ha escogido es el siguiente:

Todas las piezas se someterán primero al proceso de fresado y taladrado para posteriormente someterse al proceso de estampación. La transferencia de las piezas de una maquina a otra se realizará de forma manual, considerando que un encargado recogerá las piezas cuando hayan pasado por todo el proceso de fresado y taladrado y las llevará hasta la máquina estampadora para que se mecanicen aquí. Sin embargo, las piezas que se han retirado del primer proceso no tienen por qué llevarse inmediatamente a someterse al proceso siguiente, sino que se ha considerado que existe un almacén de piezas intermedio en el que se pueden guardar las piezas que se han producido en la primera máquina para llevarlas posteriormente al otro proceso cuando sea necesario o cuando la máquina estampadora esté disponible, ya que el proceso de fresado y taladrado al admitir más de una pieza al mismo tiempo puede mecanizar las piezas más rápido que el proceso de estampado, por lo que si no existiera un almacenaje intermedio de las piezas los dos proyectos no podrían sincronizarse.

La sincronización de los dos prototipos se ha realizado a través de la tableta HMI, mediante cables RJ-45 se han conectado todos los dispositivos (M241, S7-1200 y pantalla HMI) comunicándose mediante ethernet. En la aplicación de la HMI se han añadido los dos PLC con sus respectivas direcciones IP tal como se ha detallado en el apartado 5.3.1 Inicio de nuevo proyecto en HCTDesigner 2. Es importante que a la hora de cargar el programa en la pantalla estén conectados entre sí todos los dispositivos (los dos autómatas conectados a la HMI) porque si no aparecerá un mensaje de error de comunicación y no se podrá trabajar con la pantalla. Para realizar la sincronización de ambos procesos se ha utilizado una macro en la pantalla HMI. Este proceso se detallará más adelante.

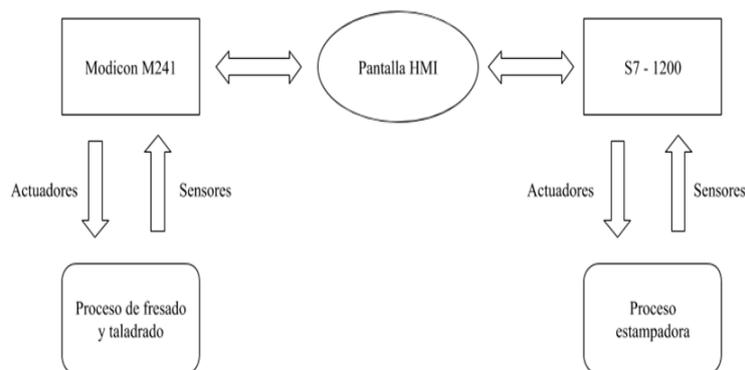


Ilustración 61. Esquema de la sincronización de ambos procesos

5.4.1 Creación de macro en la pantalla HMI

Primero se explicará que es una macro y después como se crea una en HCTDesigner 2. En una aplicación HMI se introducen elementos gráficos que sirven para dar órdenes sencillas a los autómatas, como activar un bit o transmitirle un valor numérico a través de una variable de tipo word, pero estas acciones no son un código de instrucciones programado.

Para poder realizar un programa dentro de una pantalla HMI que realice instrucciones más complejas que las realizadas por los elementos gráficos que se configuran en una ventana se deben usar las macros. Las macros son trozos de código informático en texto estructurado que permite diseñar un pequeño programa dentro de la pantalla. En HCTDesigner 2 está basado en lenguaje C, aunque tiene un asistente sencillo para poder realizar la programación de forma fácil. Para crear una macro nueva en HCTDesigner 2 se debe seguir la ruta siguiente: *Macro*>*Common Macro*>*Crear macro*, la otra opción de *Editar macro* sirve para modificar las macros que ya han sido creadas anteriormente.

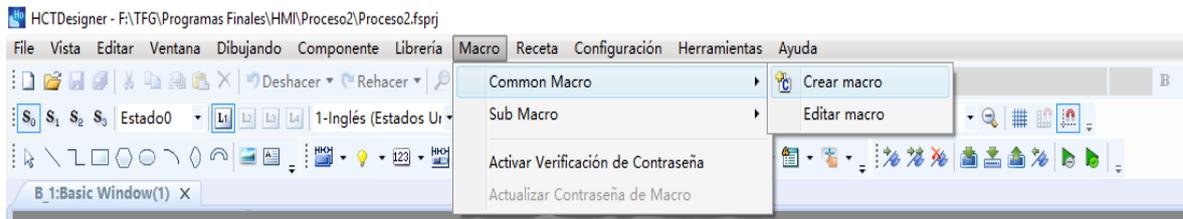


Ilustración 62. Crear nueva macro

5.4.2 Programación de la macro

Para sincronizar estos dos procesos lo que se debe hacer es llevar la cuenta de las piezas que se han realizado en cada uno de los procesos y compararlas. Para realizar esto se ha pensado en dos métodos diferentes:

1. Añadir un contador CTUD en el programa de SoMachine (proceso de fresado y taladrado) que será el encargado de comparar las cuentas de los dos procesos. Para ello se creará una variable para permitir iniciar el proceso en el autómata de la estampadora, que solo se activará cuando la cuenta del contador CTUD sea mayor de 0. Por otro lado, el autómata de la estampadora también tendrá que enviar una señal de que ha terminado de procesar una pieza para que el contador del otro proceso reste uno a la cuenta general. Cuando finalice una pieza en el proceso de fresado y taladrado se sumará uno en el contados CTUD, así si la cuenta del contador es mayor de 0 quiere decir que se han producido más piezas en el primer proceso que en el segundo y se debe permitir iniciar el segundo proceso. Por lo tanto, la macro de la pantalla HMI solo se utiliza como medio para comunicar ambos controladores lógicos pues lo único que haría en este método es leer y escribir variables de un autómata a otro.
2. El otro método es utilizar la macro de la pantalla HMI para realizar la comparación de los valores de la cuenta de piezas de cada proceso en vez de usar un contador en

uno de los programas de los autómatas. Para ello se le pasa a la macro desde los autómatas los valores de los contadores de piezas que tenían programados inicialmente para mostrar el número de piezas realizadas en la aplicación de la HMI y dentro de la macro se realiza una resta de estos dos valores, si la resta sale positiva, es decir, mayor de 0 quiere decir que hay piezas que se han procesado en el proceso primero pero no han pasado aún por el segundo proceso, por lo que se debe permitir iniciar el proceso de estampado al pulsar inicio y si la resta sale igual a 0 quiere decir que todas las piezas que se han procesado en el prototipo de la fresadora y la taladradora también han sido procesadas en el prototipo de la estampadora. Para poder indicarle al proceso de la estampadora que puede iniciar la producción de una pieza se debe crear una variable en su programa para que habilite esto.

La opción que se ha escogido de las dos anteriores es la segunda, ya que es más sencillo de aplicar este ya que no se tiene que modificar demasiado los códigos de los procesos porque no hay que introducir un nuevo contador como en el método primero y con una simple resta se puede llevar a cabo la sincronización.

Además, durante la programación de la macro se intentó llevar a cabo el primer método como opción para sincronizar ambos procesos. Casi todo funcionaba correctamente, sin embargo, cuando se necesitaba escribir una variable que fuera desde el autómatas de la estampadora al autómatas del otro proceso para indicar que se había acabado de producir una pieza en la estampadora y se debía restar al valor del contador esto no funcionaba correctamente, ya que no se consiguió hacer que se activara esta variable.

A continuación, va a detallarse como se ha realizado la programación de la macro del segundo método para la sincronización de ambos procesos.

Lo primero que se debe hacer es declarar las variables que se van a utilizar en la macro. Estas tienen que estar asociadas con una dirección de memoria bien de los autómatas o bien de la propia memoria interna de la pantalla HMI. En esta macro solo se van a utilizar variables asociadas a direcciones de la memoria de los autómatas.

Declarar una variable en una macro en el HCTDesigner 2 es muy sencillo. Se debe seleccionar la opción *Crear* situada en la parte inferior izquierda y se desplegará la ventana que aparece en la Ilustración 63 donde se introduce el nombre de la variable y la dirección de memoria con la que está asociada. En este caso se deben declarar las siguientes variables:

- 2 variables de tipo word que servirán para registrar el número de piezas producidas por cada proceso. Para el proceso de fresado y taladrado se crea la variable Cuenta2 que se asocia con la dirección %MW3 del autómatas de Schneider. Para el proceso de estampación se crea la variable Cuenta1 asociada a la dirección %MW6 del autómatas de Siemens.
- Una variable de tipo bool que servirá para permitir el inicio del proceso automático de la estampadora. Esta variable, denominada como Permite1 en la macro, está asociada con una variable que se ha creado en el programa de la estampadora llamada Q situada en la dirección de memoria %M5.2.

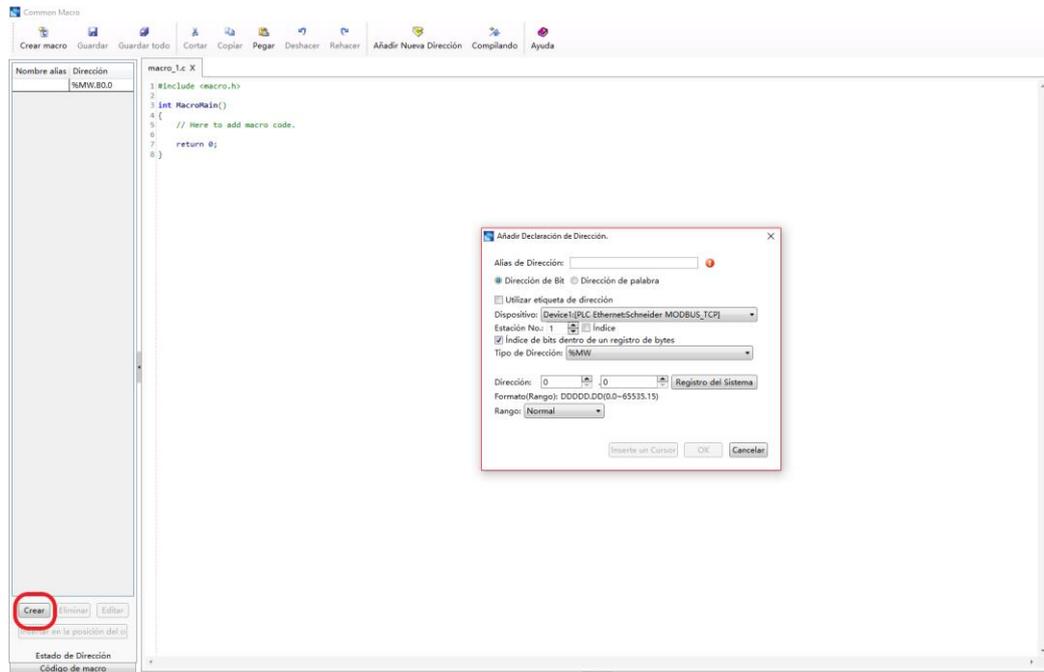


Ilustración 63. Creación de una variable dentro de una macro

El código programado y sus variables quedarían como se puede observar en la Ilustración 64. Primero se crean las variables X e Y que servirán para guardar dentro del código los valores guardados en Cuenta2 y Cuenta3 mediante el comando GetWord. Este comando funciona como se describe a continuación: Primero se introduce la variable que tiene la dirección que se quiere leer, en este caso Cuenta1 y Cuenta2, y el segundo valor a introducir es un Address Offset que sirve para moverse a través de direcciones de memoria como si fuera un vector. Por ejemplo, Cuenta2 está en el MW3, si en vez de 0 se introdujera 1 leería la dirección MW4 y no la MW3. Una vez que se han leído los valores de Cuenta1 y Cuenta2 se realiza la comparación mediante un if else. Si X menos Y es mayor de 0, es decir, si se han producido más piezas en el proceso de fresado y taladrado que en el proceso de estampación, la variable Permite1 se pondrá a 1 y si la resta no es mayor de 0 se escribirá un 0 en Permite1. Para escribir en la dirección de memoria de Permite1 se ha usado el comando SetBit. Este comando funciona igual que el comando GetWord, se debe introducir la dirección del bit a través de la variable de la macro (Permite1 en este caso), luego se introduce el address offset que en este caso es 0 también y por último se debe introducir que valor se quiere escribir en esa dirección de memoria. En este caso al ser un bit se debe introducir 0 o 1. En este caso, al contrario que en la función de lectura, el comando devuelve un bit de control de error para saber si se ha realizado correctamente la escritura. Este bit se guarda en las variables llamadas err y E aunque no han sido utilizadas para nada más, solo se han creado porque este comando obliga a ello.

Nombre alias	Dirección
Cuenta2	%MW3
Permite1	M.B5.2
Cuenta1	MW6
Reinicio	%MW.B1.7

```
Sincronizaci... X
1 #include <macro.h>
2
3 int MacroMain()
4 {
5     // Here to add macro code.
6     unsigned short X = GetWord(@Cuenta2@,0);
7     unsigned short Y = GetWord(@Cuenta1@,0);
8     if((X-Y)>0)
9     {
10        int err = SetBit(@Permite1@,0,1);
11        int E = SetBit(@Reinicio@,0,0);
12    }
13    else
14    {
15        int err = SetBit(@Permite1@,0,0);
16        int E = SetBit(@Reinicio@,0,1);
17    }
18 }
```

Ilustración 64. Código de la macro

Como puede verse en el código de la macro se ha creado una variable llamada Reinicio que se pone a 0 cuando la diferencia del número de piezas producidas en cada proceso es distinto de 0 y se pone a 1 cuando es igual. Los contadores se reinician solo cuando se apaga la máquina, esto es cuando se activa el estado inicial. Esta variable ha sido creada para permitir el reinicio de los contadores de los autómatas solo si la cuenta de piezas es igual en ambos procesos y así evitar que al apagar uno de los dos procesos la cuenta se reinicie y bloquee la macro.

Por ejemplo, si se han producido 4 piezas en el primer proceso y 2 en el segundo, si el primer proceso se apaga el número de piezas en uno sería 0 y en el otro 2 y la resta saldría negativa por lo que entraría a ejecutar las instrucciones del else y no permitiría iniciar un nuevo proceso, aunque en realidad faltan 2 piezas por producirse, o por el contrario si se apaga el segundo proceso este se reiniciaría y permitiría producir más piezas en este proceso de las que debe producir realmente.

Solo se ha creado una variable de reinicio para el programa del M241 ya que en el S7-1200 se ha aprovechado la variable Permite1, nombrada como Q en el programa del autómata, para permitir el reinicio del contador usando lógica negada.

Por otro lado, mientras se comprobaba el funcionamiento final de los dos procesos sincronizados se detectó otro fallo. Este es que al pulsar el botón de emergencia mientras se está produciendo una pieza en uno de los procesos cuando los contadores tienen el mismo número de piezas contado (ya que solo aumentan su cuenta cuando la pieza llega al final del proceso) se permite el reinicio y el contador del proceso en el que se produce la emergencia se pone a 0 produciéndose el fallo descrito antes y no permitiría terminar de procesar las piezas. Por lo tanto, se han añadido más condiciones para el reinicio de los contadores de ambos autómatas, que son las siguientes:

- Que no esté el interruptor que acciona el modo de emergencia accionado. Así si se activa el modo de emergencia cuando la cuenta de ambos contadores es la misma no pasaría nada al no reiniciarse.
- También se ha añadido una condición extra, aunque esta es solo asegurar que los contadores se van a reiniciar solo cuando la máquina esta apagada. Esta condición se incluye solo para mantener la cuenta de las piezas tras el accionamiento del modo de emergencia porque al salir de la emergencia, si los contadores tienen guardado el mismo número se habilita el reinicio y al estar todos los Grafcet en su etapa inicial el contador se reinicia, pudiendo volver a surgir el problema anteriormente descrito. Por eso solo se habilita el reset de los contadores si esta desactivado el interruptor Encendido.

En el caso del contador en el programa de la fresadora y la taladradora se ha añadido también las condiciones de que todos los Grafcet del proceso automático estén en su etapa inicial para asegurar que solo se reinicia el contador cuando la máquina está completamente parada.

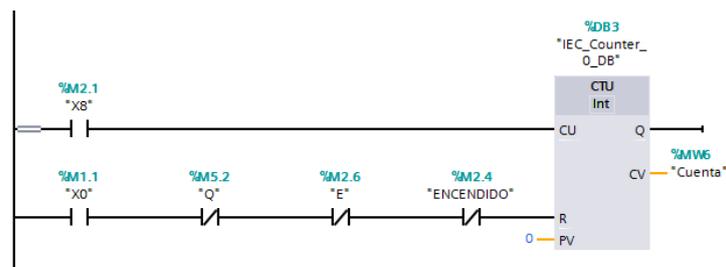


Ilustración 65. Contador del programa de la estampadora

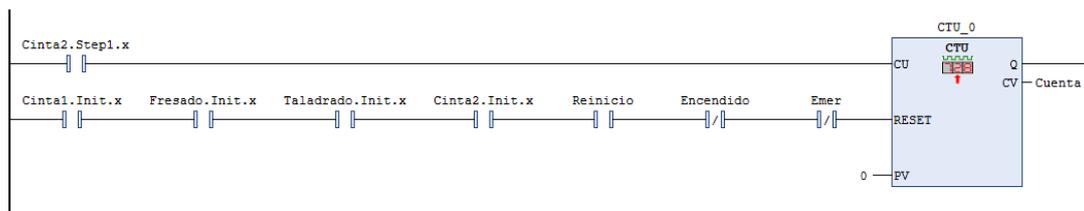


Ilustración 66. Contador del proceso de fresado y taladrado

Y con esto se puede dar por finalizada la descripción de la parte práctica de este proyecto. Una vez se ha diseñado todo se ha realizado el montaje final y la puesta en marcha de todos los equipos comprobando que funciona todo correctamente y que los dos procesos se sincronizan.

6. Conclusiones

Para la realización del proyecto y la consecución de los objetivos propuestos se han aplicado conocimientos de automatización industrial adquiridos durante el grado universitario y, además, se han ampliado éstos. Antes de la realización del trabajo, el alumno solo había tenido contacto con el lenguaje de programación Grafcet, sin embargo, para la realización de este proyecto se necesitaba aprender a utilizar lenguajes como diagramas de contactos o diseño de aplicaciones HMI.

Este proyecto ha servido para conocer cómo funcionan los sistemas de automatización realmente, utilizando unos softwares muy potentes tales como TIA Portal y SoMachine. Debido a su orientación profesional, los conocimientos adquiridos suponen una herramienta esencial en la futura vida laboral del estudiante.

Mientras se llevaba a cabo este proyecto, se han podido ver las dificultades que conlleva la puesta en marcha de un proceso automatizado. La principal dificultad observada son los fallos en el código programado de los autómatas, debido a que el planteamiento inicial de su funcionamiento es muy complicado de llevar a la práctica. Otra dificultad observada ha sido la sincronización de ambos procesos porque la comunicación entre los autómatas a través de la HMI era complicada de llevar a cabo sin errores.

Este proyecto ha supuesto la oportunidad de conocer la automatización y las capacidades necesarias para poder trabajar en un futuro dentro de este sector. Se considera interesante la capacidad de resolución de problemas y la prevención de irregularidades que puedan darse durante la vida del automatismo.

7. Bibliografía

- Automatización Industrial. (s.f). Lenguaje de contactos. *Formación para la Industria 4.0*. Recuperado de <https://automatismoindustrial.com/automatas/g-1-introduccion-automatas-programables-plcs/d-9-4-lenguaje-de-contactos/>
- Cursos de Ingeniería Industrial. (5 de septiembre de 2018). Diferencias SCADA y HMI. *Vester Industrial Training Center (VITC)*. Recuperado de <https://vestertraining.com/diferencias-scada-hmi/>
- Electrin. (2016). Entradas y salidas de un PLC. *Electrinblog*. Recuperado de <https://electrinblog.wordpress.com/2016/04/25/post-3/>
- Escuela Politécnica Nacional. (2017). Fuente de alimentación FAC-363B. *StuDocu*. Recuperado de <https://www.studocu.com/es/document/escuela-politecnica-nacional/hidraulica/otros/fuente-promax-fac363b/1118190/view>
- García, Borja y Martín-Peñasco, Jaime. (11 de junio de 2018). Inteligencia artificial y automatización, el futuro del ser humano en peligro. *Blogthinking.com*. Recuperado de <https://blogthinkbig.com/automatizacion-del-empleo>
- Genia. (s.f). *Resumen sobre Grafcet*. Recuperado de <http://isa.uniovi.es/genia/spanish/publicaciones/grafcet.pdf>
- HCFA. (s.f). TP2507. *HCFA.com – Controllers*. Recuperado de <http://www.hcfa.com.cn/home/content/controldetails/id/27.html>
- HCFA. (s.f). *HCTDesigner SOFTWARE MANUAL.pdf*
- InfoPLC. (16 de diciembre de 2017). Curso autómatas – Lenguaje Ladder (LD). *InfoPLC – Automatización industrial, robótica e industria 4.0*. Recuperado de <http://www.infoplcn.net/documentacion/5-automatas/2704-curso-automatas-lenguaje-ladder>
- Logicbus. (s.f). Que es la automatización. *Logicbus*. Recuperado de <https://www.logicbus.com.mx/automatizacion.php>
- Mecafenix, Frank. (16 de enero de 2018). ¿Qué es y para qué sirve un PLC? *Ingeniería Mecafenix*. Recuperado de <https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>
- Merino, María y Pérez Porto, Julián. (2017) Definición de automatización. *Definición.de*. Recuperado de <https://definicion.de/automatizacion/>
- Ribas, Jose. (s.f). Lógica cableada y lógica programada. *DissenyProducte*. Recuperado de <https://dissenyproducte.blogspot.com/2010/12/logica-cableada-y-logica-programada.html>
- S2E. (s.f). Diferencia entre SCADA y HMI. *S2E – Ingeniería Electrónica*. Recuperado de <http://s2e.es/diferencia-entre-scada-y-hmi/>
- Schneider Electric. (Diciembre 2015). *Catálogo – Controladores Lógicos Modicon M241.pdf*
- Schneider Electric. (s.f). *Controlador Lógico Modicon M241_TM241CE40R.pdf*

Schneider Electric. (2014). *Modicon M241 – Guía de Programación (EIO000000143503M241).pdf*

Schneider Electric. (Abril 2017). *Modicon M241: Funciones y variables del sistema (EIO0000001441.03).pdf*

Schneider Electric. (2015). *SoMachine – Guía de Programación (EIO0000000071.12).pdf*.

Schneider Electric. (s.f). *SoMachine: Manual de formación.pdf*

Siemens. (Junio 2015). *S7: Controlador programable S7-1200 - Manual de sistema.pdf*

Siemens. (Noviembre 2009). *S7-1200: Getting Started del S7-1200.pdf*

Sy Corvo, Helmut. (s.f). Automatización industrial: historia, características y tipos. *Lifeder.com*. Recuperado de <https://www.lifeder.com/automatizacion-industrial/>

Wonderware Iberia. (s.f) Interfaz Hombre-Máquina (HMI). *Wonderware.es*. Recuperado de <https://www.wonderware.es/hmi-scada/que-es-hmi/>

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

2. PLIEGO DE CONDICIONES

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Índice del pliego de condiciones

1. Objeto y alcance del proyecto	1
2. Descripción del montaje.....	1
3. Características del material empleado.....	2
4. Normativa.....	2
5. Manual de la aplicación HMI.....	3

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

1. Objeto y alcance del proyecto

En este documento se recogen las condiciones técnicas y legislativas a cumplir para la correcta realización de este proyecto de forma segura y eficiente, además de los pasos a seguir para llevar a cabo el mismo.

El objetivo de este proyecto es realizar la automatización de dos subprocesos industriales coordinados entre sí y simulados mediante dos prototipos. Para ello se ha utilizado los autómatas Siemens S7 1200 y Schneider M241 También se ha empleado para la monitorización y control de los subprocesos una aplicación SCADA en una pantalla HMI TP-2507-E.

2. Descripción del montaje

Para la correcta realización y puesta en marcha de este proyecto hay que tener en cuenta que debe ser realizado y llevado a cabo por personal cualificado con los conocimientos necesarios para poder resolver cualquier problema que pueda surgir durante su montaje o funcionamiento y que este capacitado para utilizar los softwares necesarios para este proyecto. Se debe proceder siguiendo los siguientes pasos:

1. Evaluación de las maquetas proporcionadas por el cliente, conocer todas sus entradas y salidas y los elementos de los que están compuestas
2. Analizar y comprender los procesos a automatizar entendiendo las especificaciones requeridas por estos.
3. Montaje de los autómatas
4. Programación de cada uno de los procesos utilizando el PLC y el software asignados a cada uno.
5. Evaluación de errores de los automatismos programados
6. Diseño de aplicación para pantalla HMI
7. Evaluación de funcionamiento de la aplicación para cada uno de los procesos y autómatas.
8. Programación de la macro para sincronizar ambos procesos.
9. Conexión de todos los dispositivos mediante cables Ethernet.
10. Evaluación del funcionamiento de la sincronización.
11. Evaluación global del proyecto.

Para el montaje del automatismo del proceso de estampación se utilizará el PLC Siemens S7 1200 programado mediante TIA Portal V13. Para el montaje del proceso de fresado y taladrado se utilizará el PLC Schneider Modicon M241 programado mediante SoMachine V4,1 y para el control y sincronización de ambos procesos la pantalla HMI HCFA TP-2507-E programada mediante HCTDesigner 2.

3. Características del material empleado

Las características técnicas de los materiales empleados están descritas en el apartado 4. Materiales del documento de la memoria descriptiva.

Para el correcto funcionamiento de los procesos automatizados en este proyecto se deben utilizar exclusivamente los materiales expuestos a lo largo de este, ya que si se cambia alguno de los componentes por otro distinto no se garantiza el funcionamiento correcto de los procesos. En cuanto a recambios, no debe ser difícil encontrarlos al tratarse de materiales muy utilizados en el sector como son los autómatas de Siemens o Schneider.

En caso de mala utilización de cualquiera de los materiales expuestos, la responsabilidad únicamente caerá en el responsable del uso incorrecto de ese material.

Con respecto al ordenador utilizado, debe cumplir unos requisitos mínimos para poder usar los softwares de este producto, siendo estos los siguientes:

- Para el TIA Portal V13:
 - Windows 7 o 8.1
 - Intel® Core™ i5-3320M 3,3 GHz o superior
 - 8GB de RAM
- Para el SoMachine V4.1:
 - Windows 7 o 8.1
 - Intel® Core™ 2 DUO o equivalente
 - 3GB de RAM

4. Normativa

A continuación, se detallan todas las normas y legislaciones que se deben cumplir para la correcta realización de este proyecto:

- Norma IEC-1131, en la que se estandarizan todos los lenguajes de programación de autómatas para evitar incompatibilidades entre ellos.
- Reglamento Electrotécnico de Baja Tensión (REBT), especialmente la Instrucción Técnica Complementaria ITC-BT-51 sobre instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios.
- Estándar internacional IEEE 802.3, en el que se estandarizan las redes basadas en Ethernet.
- Manual de TIA Portal V13
- Manual de SoMachine V4.1
- Manual de HCTDesigner
- Catálogo de especificaciones técnicas de cada material empleado.

5. Manual de la aplicación HMI

En este apartado se explica cómo hacer uso de la aplicación HMI programada. La pantalla que aparecerá por defecto al arrancar la HMI será la del llamado proceso 2 (fresado y taladrado). Para cambiar de ventana se deben usar los botones situados en la esquina superior derecha. Cada botón tiene el nombre de la ventana a la que dirige. Hay tres ventanas principales disponibles: El proceso 1, el proceso 2 y el registro de eventos y emergencias. Para acceder a este último se requiere una contraseña previamente introducida mediante el HCTDesigner 2. Esta contraseña debe elegirla el cliente. A continuación, se describen las opciones de cada una de estas ventanas:

- Proceso 1: En la parte izquierda se dispone de todas las opciones de control del proceso, pudiendo cambiar entre modo manual y automático, iniciar el modo de parada de emergencia, iniciar el proceso automático o encender o apagar la máquina. En la parte derecha aparece un esquema de la máquina indicando los sensores activados en cada momento y también cuenta con un contador de piezas producidas en modo automático. En esta parte, entre el contador de piezas y el esquema de la máquina, se encuentran los controles manuales, los cuales solo se pueden utilizar con el modo manual activado.
- Proceso 2: En la parte izquierda dispone de los mismos botones que el proceso 1, la diferencia está en que para acceder al control manual se debe ir a la ventana *Modo Manual*. Para salir de la ventana de control manual se pulsa en la cruz de la esquina superior derecha. En la parte derecha estaría el esquema de la máquina con los sensores y el contador de piezas.
- Registro: En esta ventana aparece un listado de los eventos y alarmas de los procesos (emergencias y activación de modo manual) junto con la fecha y la hora de activación. Para salir de esta ventana se debe pulsar en la cruz de arriba a la derecha.

Cuando se active el modo de emergencia se deberá activar el modo manual y desactivar el botón de emergencia para poder salir de la parada del proceso. Aparecerá una ventana emergente que indica que se ha activado la parada de emergencia y explica cómo salir de ella. Para cerrar esta ventana se pulsa en la cruz de la esquina superior derecha.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

3. PRESUPUESTO

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Índice del presupuesto

1. Hardware	1
2. Software	2
3. Mano de obra.....	2
4. Presupuesto total	3

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

En este documento se desglosa el presupuesto que engloba todos los elementos utilizados para la realización del proyecto; incluyendo tanto materiales como softwares, autómatas y la pantalla HMI.

Dentro del presupuesto se contabiliza la mano de obra utilizada para la correcta consecución del trabajo y su puesta en marcha.

1. Hardware

Los precios aquí demostrados para los elementos de hardware son estimados. En el caso de los prototipos de laboratorio se considera que los suministra el cliente. En el caso del PC, no se le puede cobrar su precio integro al cliente ya que este hardware se puede utilizar para otros proyectos, para ello se procede a calcular su precio por hora. Fijando como precio medio de los ordenadores del mercado que cumplen los requisitos de los softwares utilizados en 550€ y que se considera una amortización en 3 años, sería una amortización de 183,3€/año. Considerando que el año tiene alrededor de 220 días laborables y que la jornada máxima de trabajo son 8h diarias da un total de 1760h anuales en las que se usa el ordenador, por lo que su precio por hora son 0,1€/h.

HARDWARE			
Descripción	Cantidad	Precio / unidad	Importe
Siemens CPU 1214C AC/DC/Rly	1	342,41€	342,41€
Schneider TM241CE40R	1	381,51€	381,51€
HCFA TP-2507-E	1	223,96€	223,96€
Prototipo estampadora Fischertechnik	1	0€	0€
Prototipo fresado y taladrado Fischertechnik	1	0€	0€
Fuente de alimentación Promax FAC-363B	1	110€	110€
Cable Ethernet	3	9,5€	28,5€
PC	200h	0,1€/h	20€

Importe bruto	IVA 21%	Importe total
1106,38€	232,34€	1338,72€

2. Software

Los precios aquí demostrados para los elementos de software son estimados. Para el caso del TIA Portal sucede lo mismo que con el PC, en vez de cobrar el precio total de la licencia al cliente se cobra un precio proporcional por horas de uso. El precio de la licencia del Step7 Basic es de 373,89€ y considerando una amortización en 3 años como con el PC, utilizando las 1760h de trabajo anuales el precio por hora es de 0,07€/h. Los softwares de SoMachine y HCTDesigner 2 son gratuitos.

SOFTWARE			
Descripción	Cantidad	Precio / unidad	Importe
Licencia TIA Portal-Step7 Basic	60h	0,07€/h	4,2€
Licencia SoMachine V4.1	1	0€	0€
Licencia HCTDesigner 2	1	0€	0€

Importe bruto	IVA 21%	Importe total
4,2€	0,88€	5,08€

3. Mano de obra

Como sueldo medio de un ingeniero técnico industrial se toman 25€/h.

INGENIERO TÉCNICO INDUSTRIAL			
Descripción	Cantidad	Precio / unidad	Importe
Análisis	20h	25€/h	500€
Diseño de los automatismos	90h	25€/h	2250€
Evaluación	20h	25€/h	500€
Redacción del documento	70h	25€/h	1750€

Importe bruto	IVA 21%	Importe total
5000€	1050€	6050€

4. Presupuesto total

Aquí se muestran los costes totales del hardware, el software y la mano de obra, así como el presupuesto final para la realización de este proyecto.

PRESUPUESTO TOTAL	
Descripción	Importe
Hardware	1338,72€
Software	5,08€
Mano de obra	6050€
TOTAL	7393,8€

El presupuesto total es de siete mil trescientos noventa y tres con ochenta euros.

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

4. ANEXOS

Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

Índice de anexos

1. Programa TIA Portal	1
1.1 Variables del programa	1
1.2 Contador	2
1.3 Ecuaciones de activación	2
1.4 Ecuaciones de estado.....	4
1.5 Ecuaciones de salida.....	6
1.6 Startup	8
2. Programa SoMachine	8
2.1 Cinta1	8
2.2 Fresado	10
2.3 Taladrado.....	11
2.4 Cinta2	12
2.5 Emergencia.....	12
2.6 Man	13
2.7 Contador.....	14
2.8 Acciones.....	15

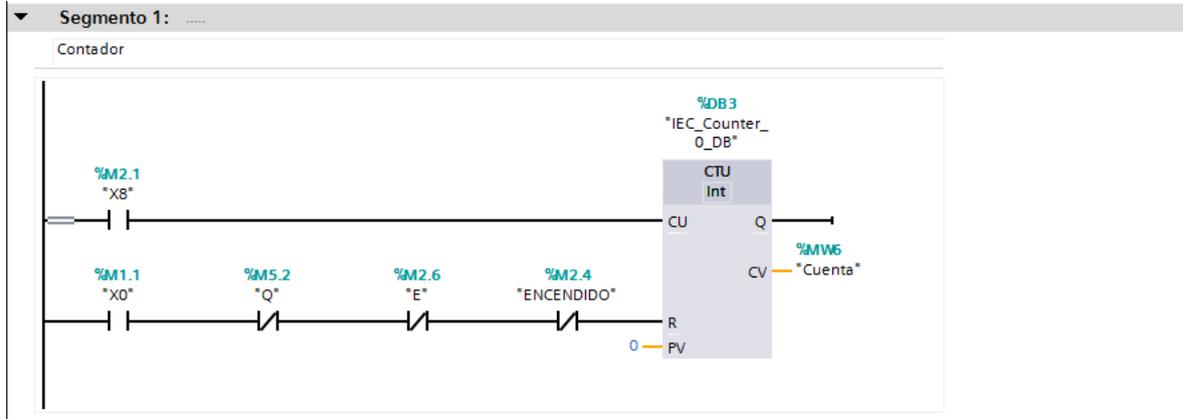
Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

1. Programa TIA Portal

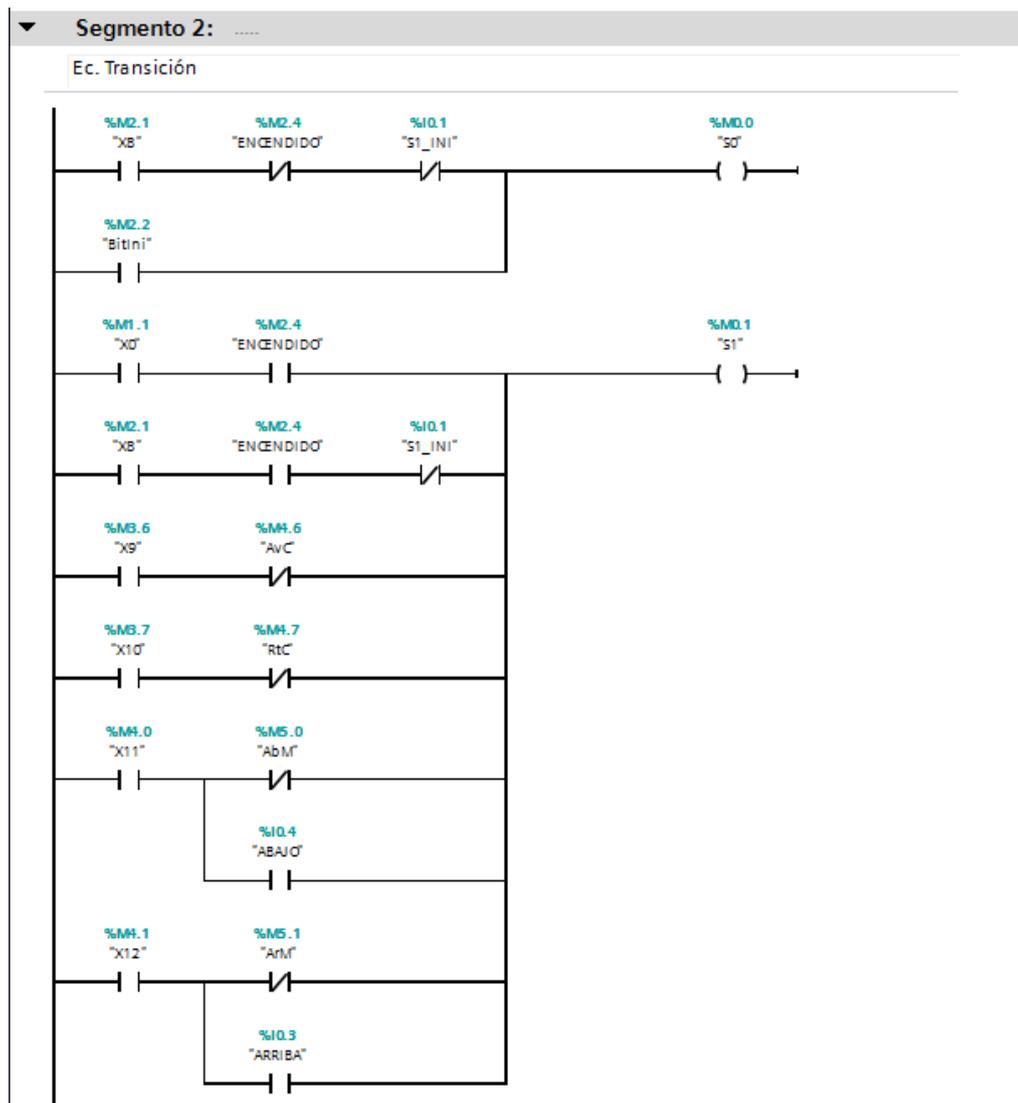
1.1 Variables del programa

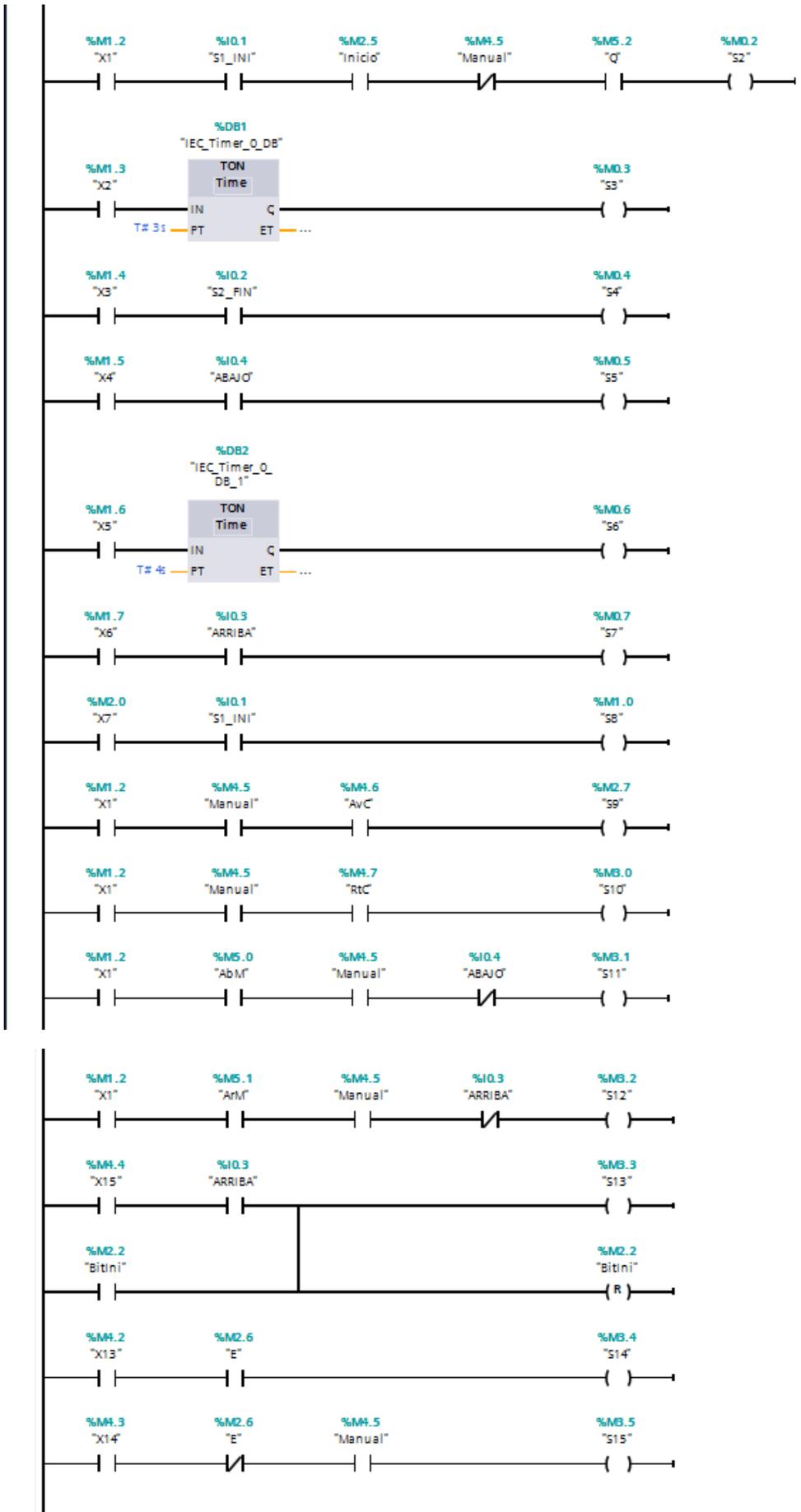
Variables PLC							
	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Visibl...	Acces...
1	S0	Tabla de variabl...	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	S1	Tabla de variables e..	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	S2	Tabla de variables e..	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	S3	Tabla de variables e..	Bool	%M0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	S4	Tabla de variables e..	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	S5	Tabla de variables e..	Bool	%M0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	S6	Tabla de variables e..	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	S7	Tabla de variables e..	Bool	%M0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	S8	Tabla de variables e..	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	X0	Tabla de variables e..	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	X1	Tabla de variables e..	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	X2	Tabla de variables e..	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	X3	Tabla de variables e..	Bool	%M1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	X4	Tabla de variables e..	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	X5	Tabla de variables e..	Bool	%M1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	X6	Tabla de variables e..	Bool	%M1.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	X7	Tabla de variables e..	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	X8	Tabla de variables e..	Bool	%M2.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	Inicio	Tabla de variables e..	Bool	%M2.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	S1_INI	Tabla de variables e..	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	S2_FIN	Tabla de variables e..	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	ARRIBA	Tabla de variables e..	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	ABAJO	Tabla de variables e..	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24	ENCENDIDO	Tabla de variables e..	Bool	%M2.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25	DIRCINTA	Tabla de variables e..	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	CINTA-ON	Tabla de variables e..	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	DIRMAQUINA	Tabla de variables e..	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	MAQUINA-ON	Tabla de variables e..	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
29	POWER	Tabla de variables e..	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
30	Cuenta	Tabla de variables e..	Int	%MW6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
31	BitIni	Tabla de variables e..	Bool	%M2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32	E	Tabla de variables e..	Bool	%M2.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33	S9	Tabla de variables e..	Bool	%M2.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34	S10	Tabla de variables e..	Bool	%M3.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35	S11	Tabla de variables e..	Bool	%M3.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36	S12	Tabla de variables e..	Bool	%M3.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37	S13	Tabla de variables e..	Bool	%M3.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
38	S14	Tabla de variables e..	Bool	%M3.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
39	S15	Tabla de variables e..	Bool	%M3.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
40	X9	Tabla de variables e..	Bool	%M3.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
41	X10	Tabla de variables e..	Bool	%M3.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
42	X11	Tabla de variables e..	Bool	%M4.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
43	X12	Tabla de variables e..	Bool	%M4.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
44	X13	Tabla de variables e..	Bool	%M4.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
45	X14	Tabla de variables e..	Bool	%M4.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
46	X15	Tabla de variables e..	Bool	%M4.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
47	Manual	Tabla de variables e..	Bool	%M4.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
48	AvC	Tabla de variables e..	Bool	%M4.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
49	RtC	Tabla de variables e..	Bool	%M4.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
50	AbM	Tabla de variables e..	Bool	%M5.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
51	ArM	Tabla de variables e..	Bool	%M5.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
52	Q	Tabla de variables e..	Bool	%M5.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
53	Reini	Tabla de variables e..	Bool	%M5.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
54	Valor	Tabla de variables e..	Int	%MW7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
55	<Agregar>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

1.2 Contador

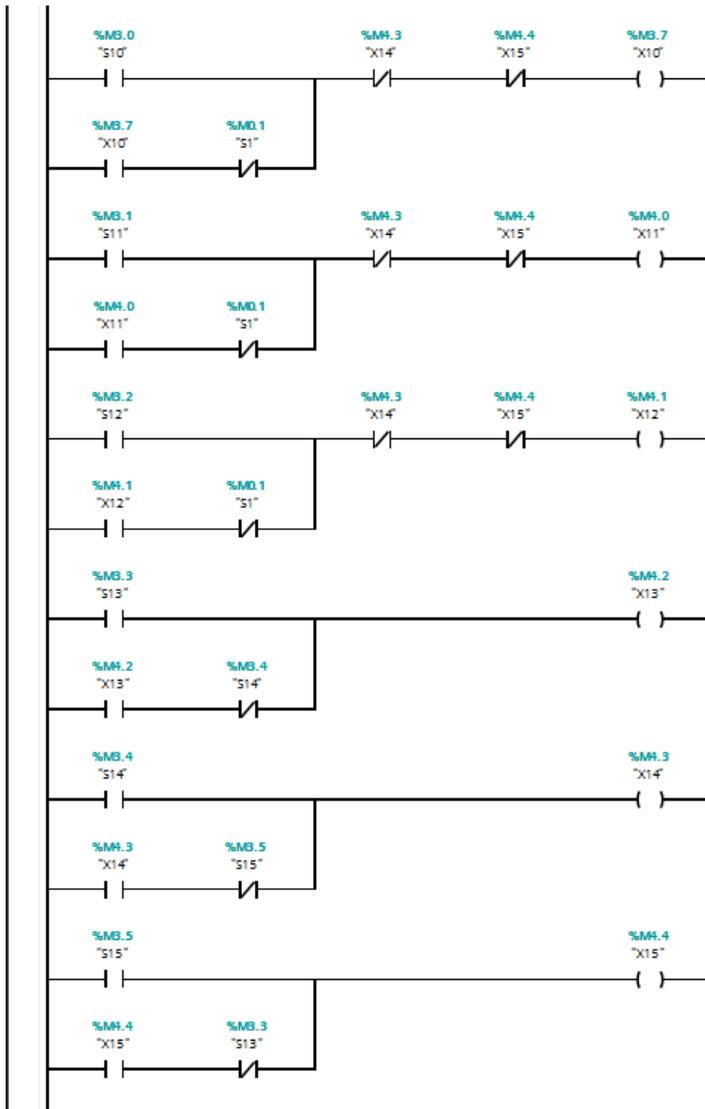


1.3 Ecuaciones de activación

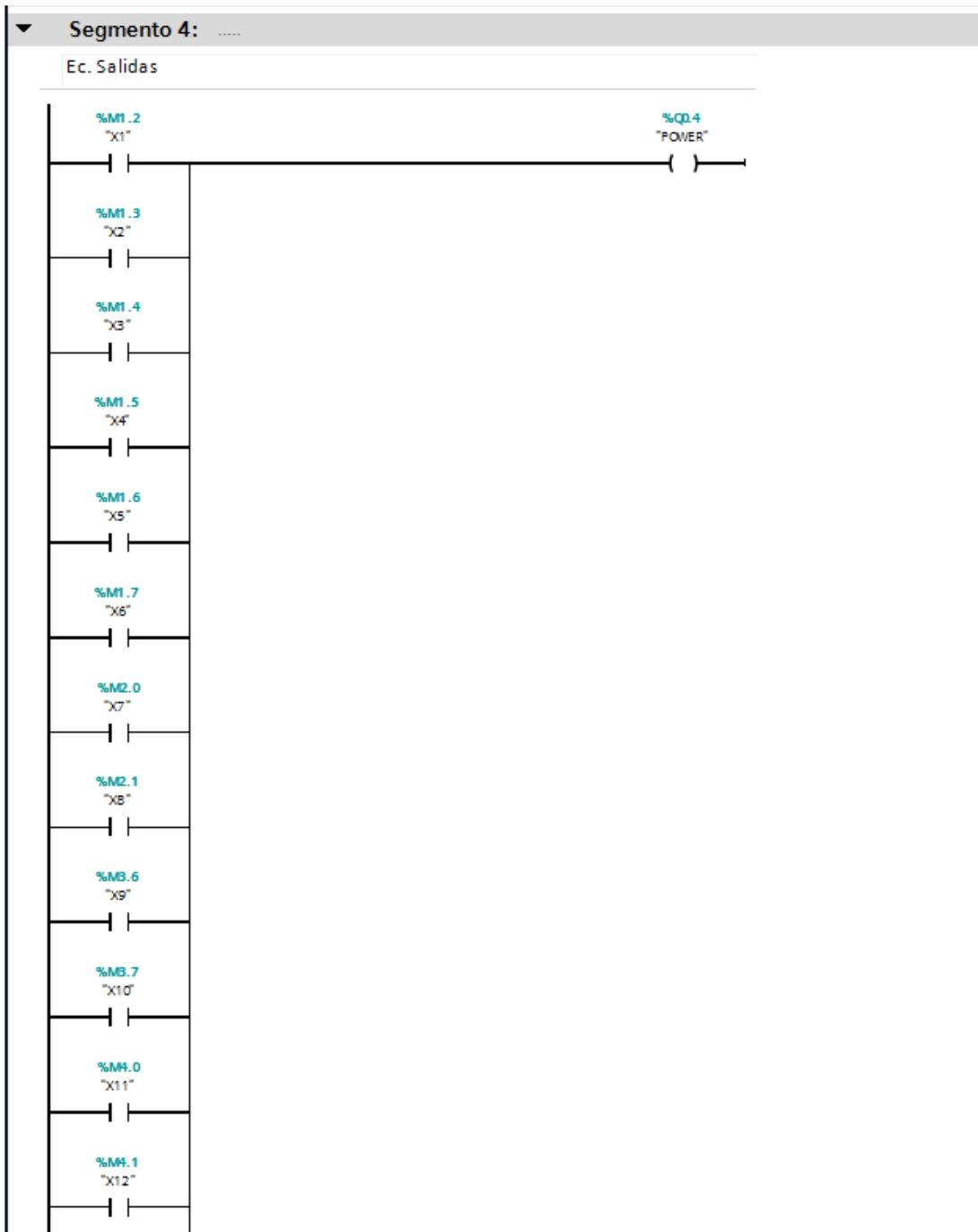


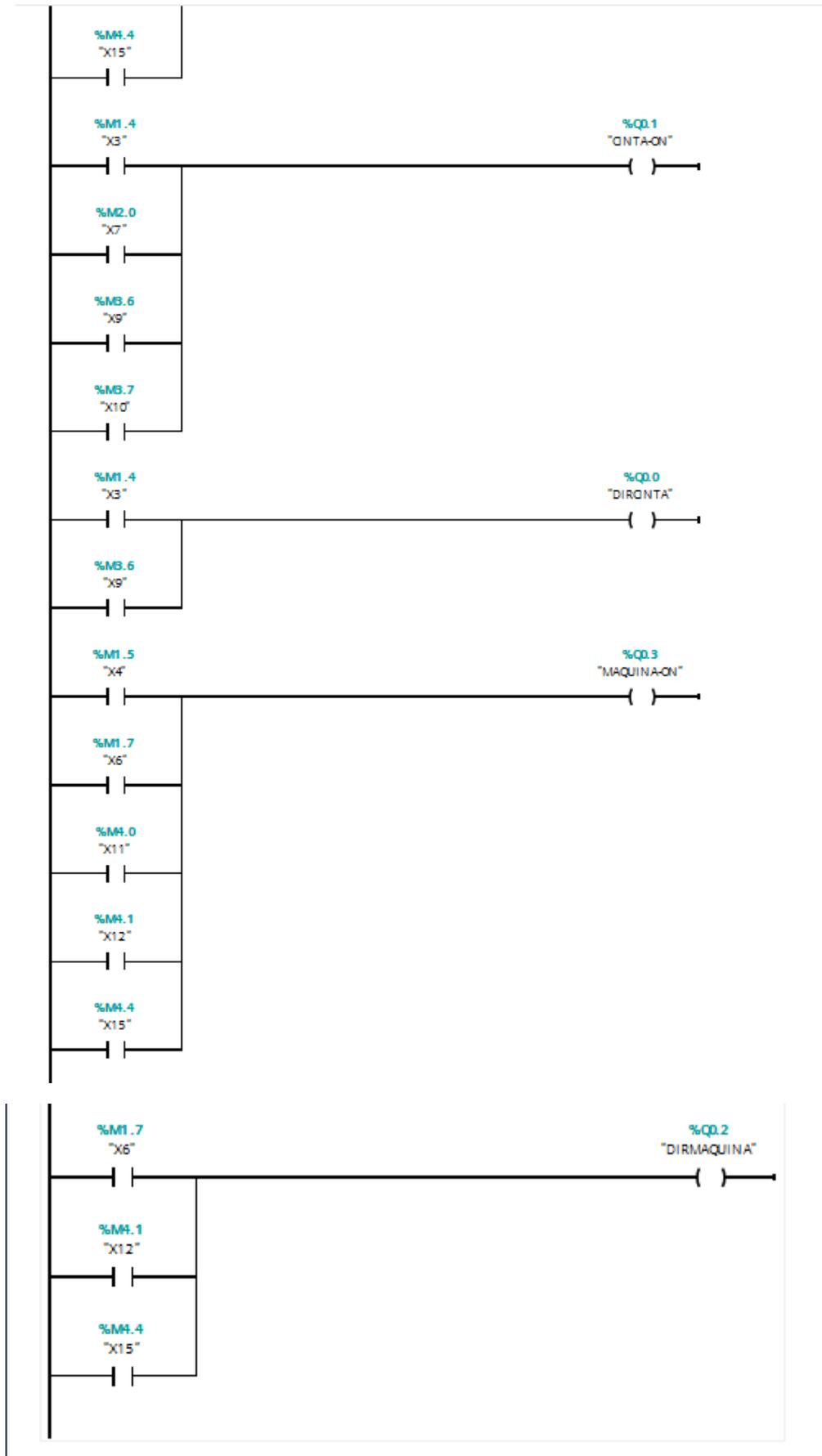


Proyecto de automatización de un prototipo de laboratorio mediante autómatas Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

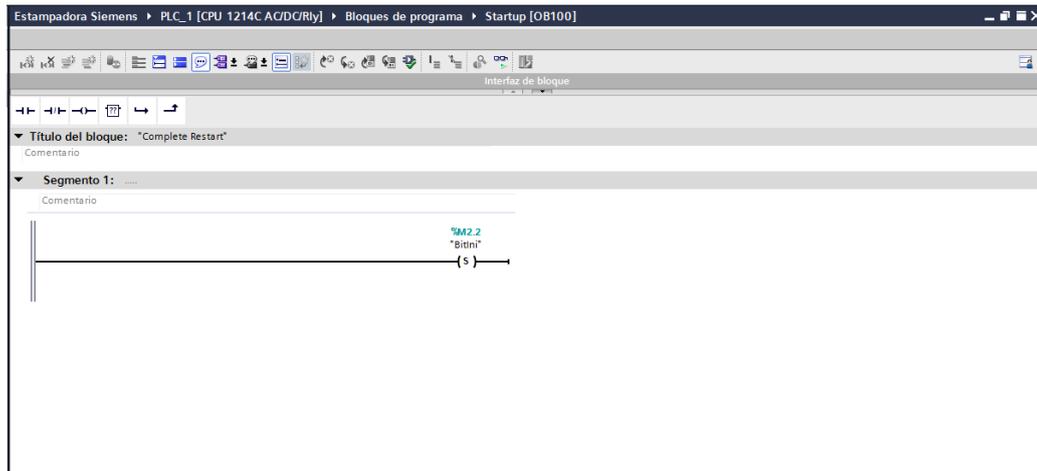


1.5 Ecuaciones de salida



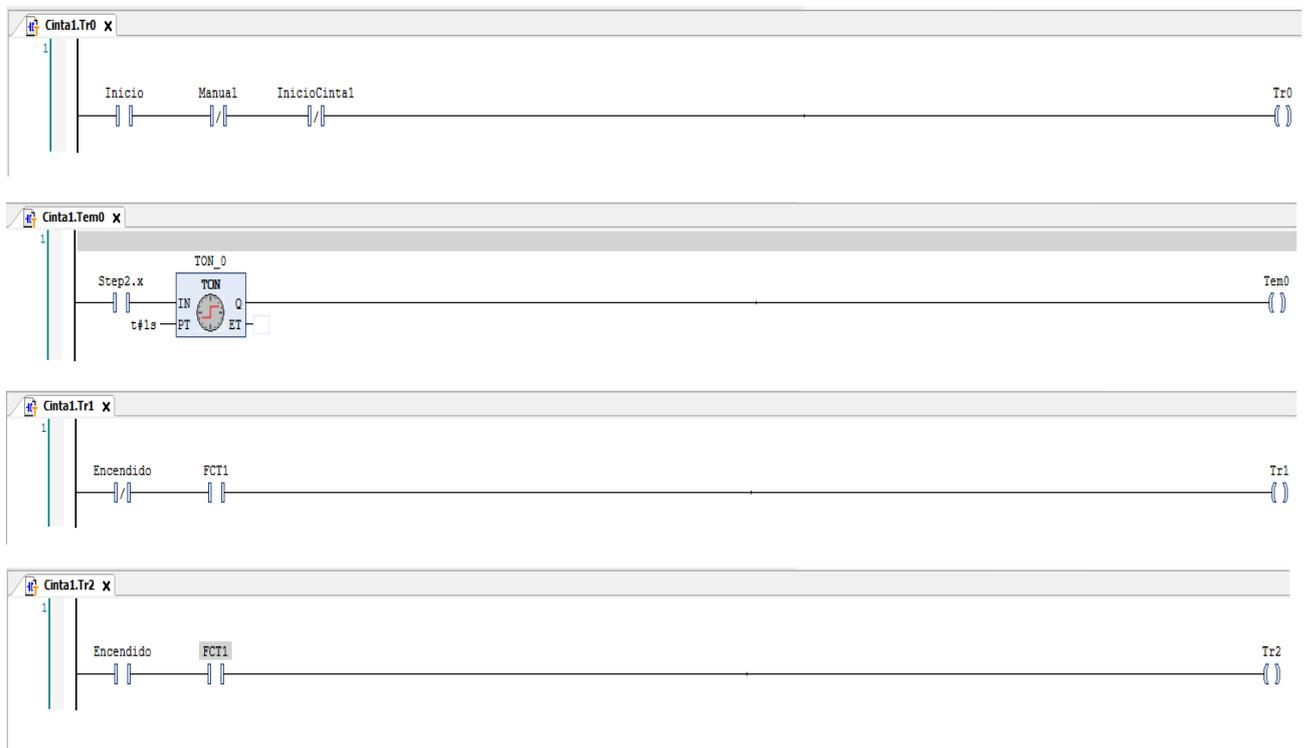


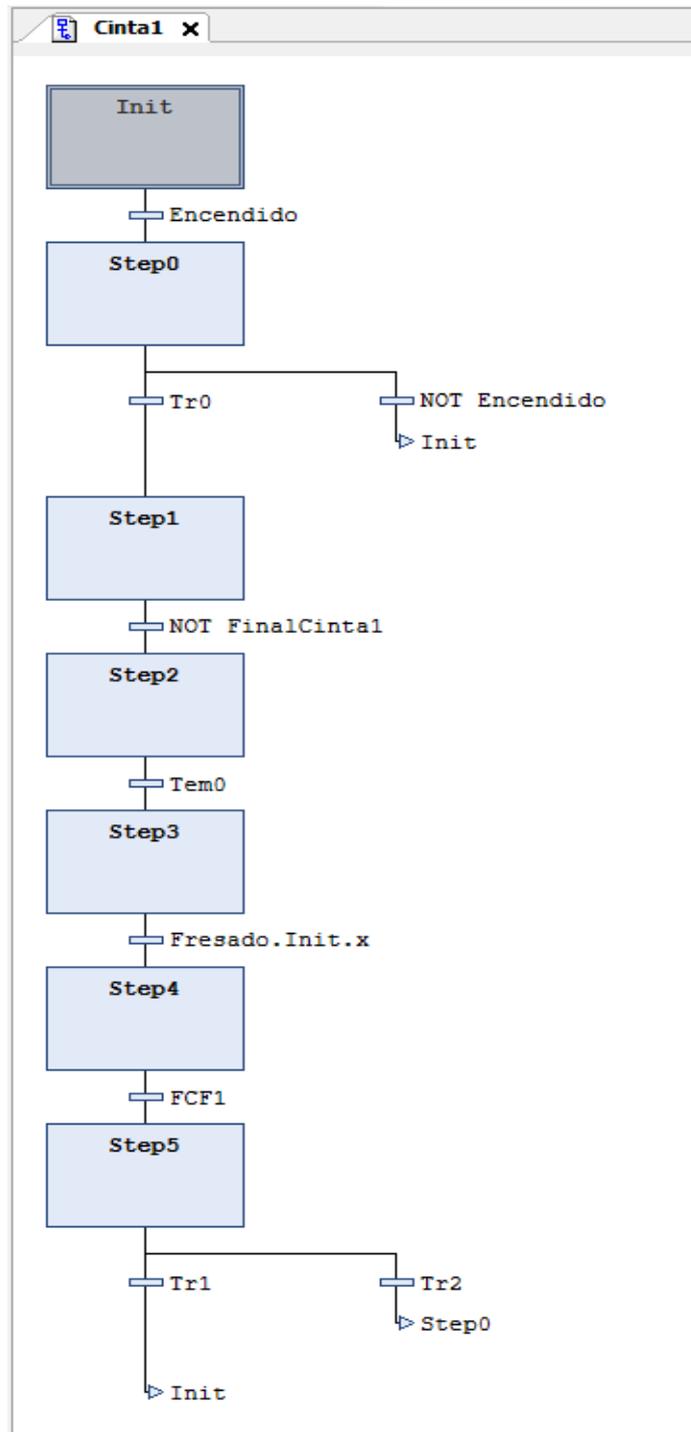
1.6 Startup



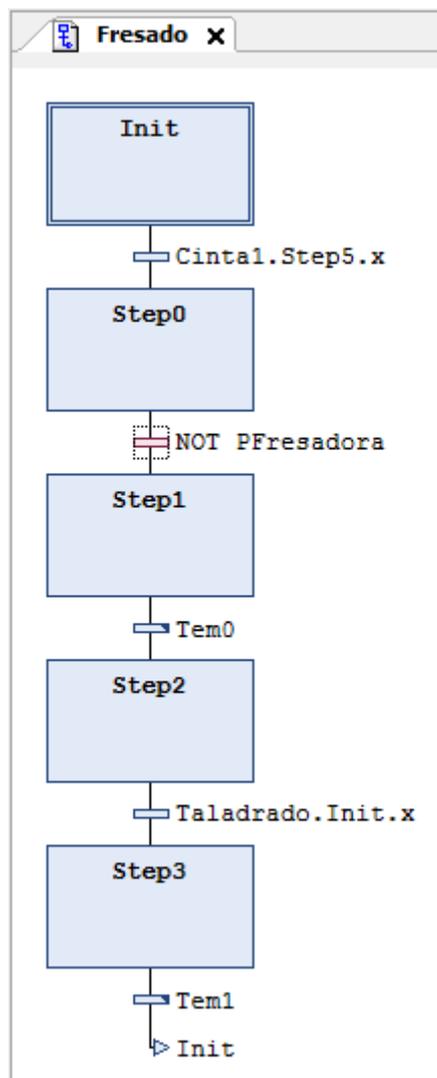
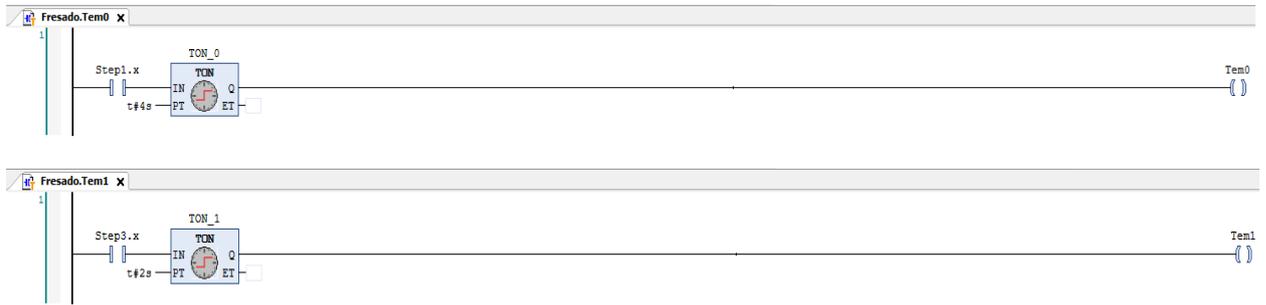
2. Programa SoMachine

2.1 Cinta1

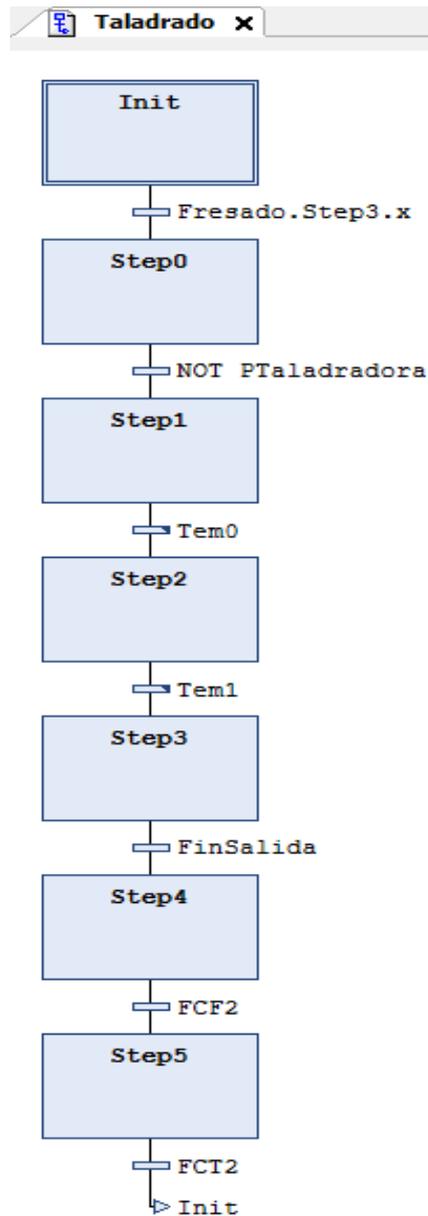
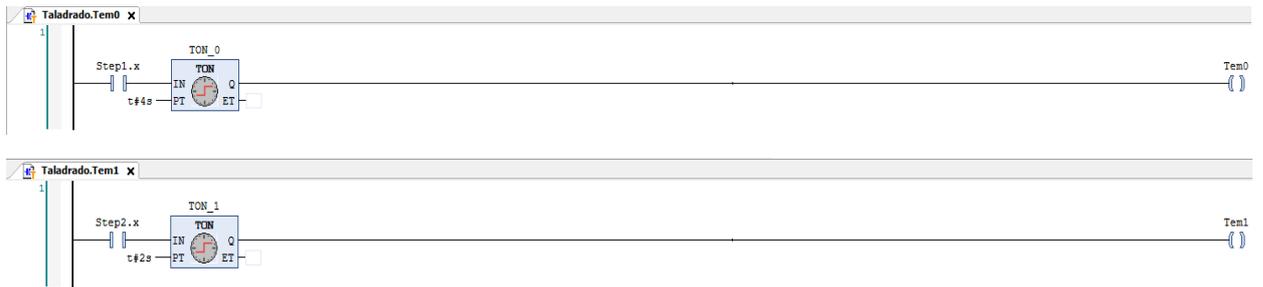




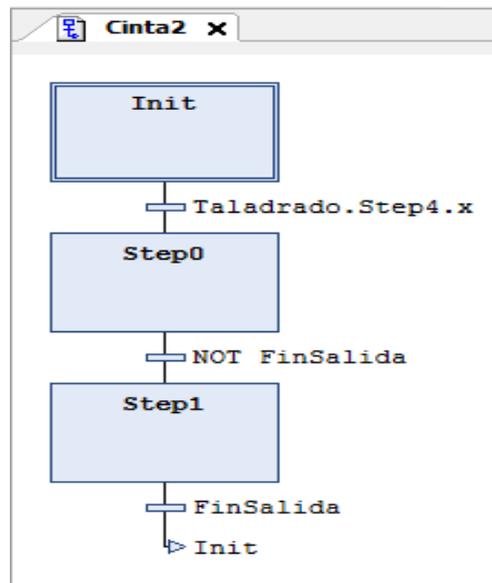
2.2 Fresado



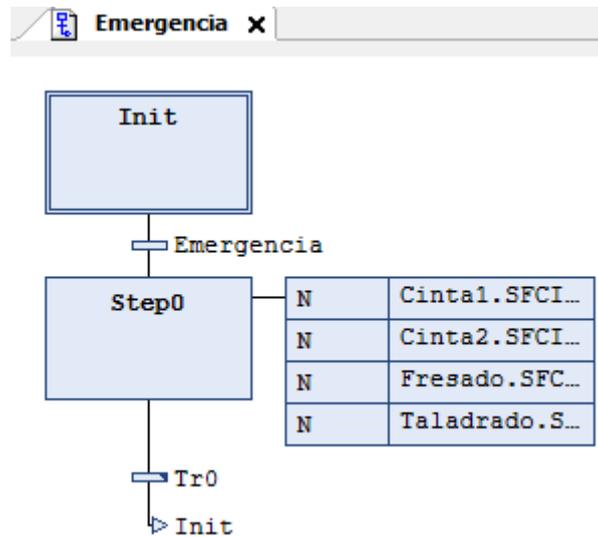
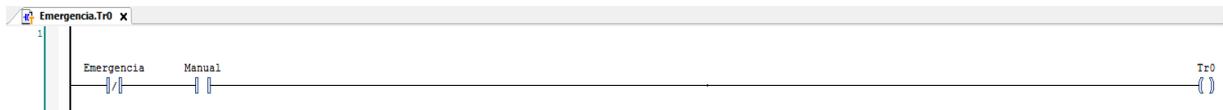
2.3 Taladrado



2.4 Cinta2



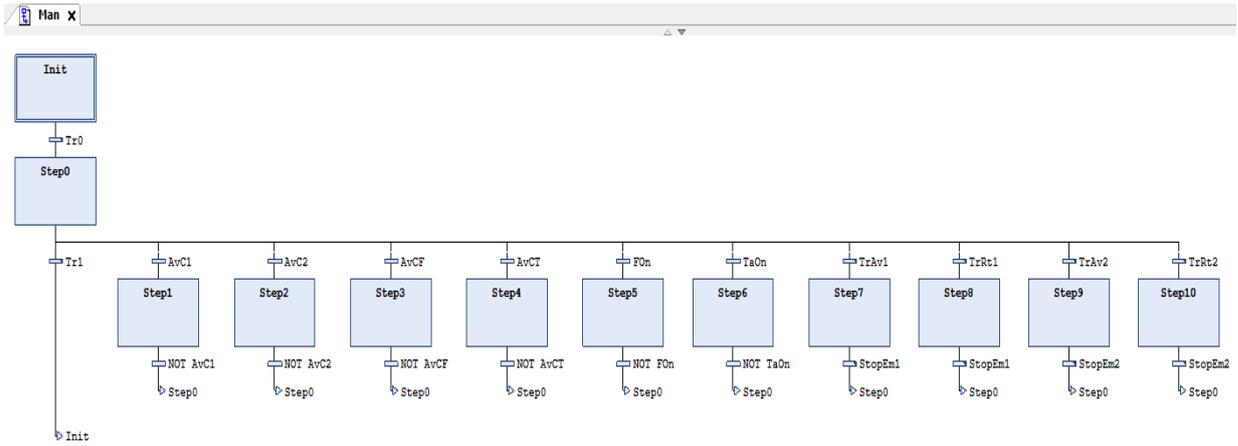
2.5 Emergencia



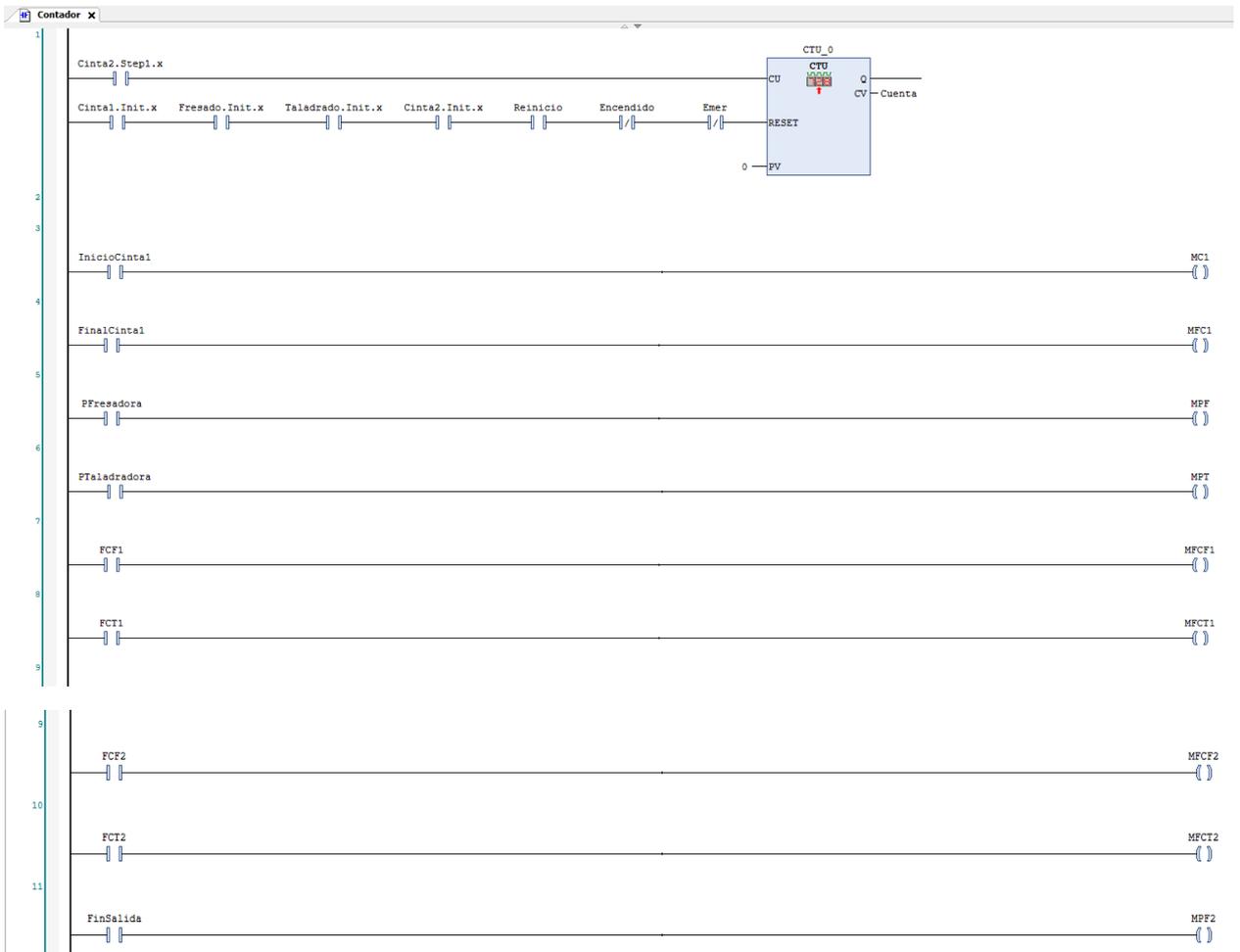
2.6 Man



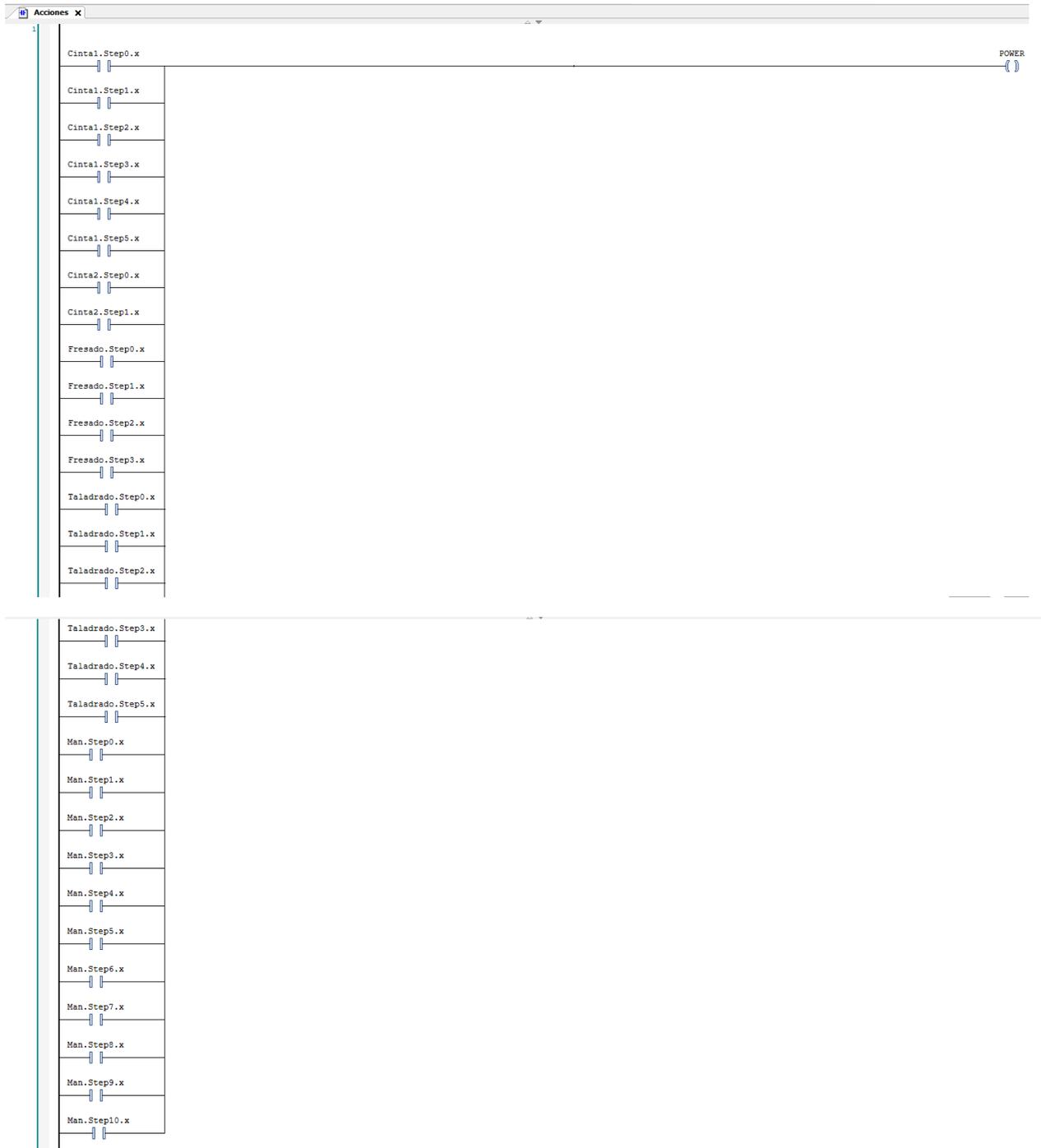
Proyecto de automatización de un prototipo de laboratorio mediante autómatas Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.



2.7 Contador



2.8 Acciones



Proyecto de automatización de un prototipo de laboratorio mediante autómatas
Siemens S7 1200 y Schneider M241, y pantalla HMI TP-2507-E.

